Qiping Chu
Bob Mulder
Daniel Choukroun
Erik-Jan van Kampen
Coen de Visser
Gertjan Looye Editors

ADVANCES IN Aerospace Guidance, Navigation and Control

Selected Papers
of the Second CEAS Specialist Conference
on Guidance,
Navigation and Control

Springer

CEAS

Advances in Aerospace Guidance, Navigation and Control

Qiping Chu · Bob Mulder · Daniel Choukroun
Erik-Jan van Kampen · Coen de Visser
Gertjan Looye
Editors

Advances in Aerospace Guidance, Navigation and Control

Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control

Springer

Editors

Qiping Chu
Faculty of Aerospace Engineering
TU Delft
Delft
The Netherlands

Erik-Jan van Kampen
Faculty of Aerospace Engineering
TU Delft
Delft
The Netherlands

Bob Mulder
Faculty of Aerospace Engineering
TU Delft
Delft
The Netherlands

Coen de Visser
Faculty of Aerospace Engineering
TU Delft
Delft
The Netherlands

Daniel Choukroun
Faculty of Aerospace Engineering
TU Delft
Delft
The Netherlands

Gertjan Looye
German Aerospace Center (DLR)
Robotics and Mechatronics Center
System Dynamics and Control
Wessling
Germany

ISBN 978-3-642-38252-9 ISBN 978-3-642-38253-6 (eBook)
DOI 10.1007/978-3-642-38253-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013937540

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

About the Book

For the 2^{nd} CEAS Specialist Conference on Guidance, Navigation and Control the International Program Committee conducted a formal review process. Each paper was reviewed in compliance with good journal practice by at least two independent and anonymous reviewers. The papers published in this book were selected based on the results and recommendations from the reviewers.

The papers in the book are assigned to four technical tracks of the conference: Guidance and Control, Navigation and Estimation, Atmospheric Applications and Space Applications.

The members of the International Program Committee are:

Daniel Alazard	ISAE, France
Mark Balas	University of Wyoming, USA
Henk Blom	NLR, The Netherlands
Jan Breeman	NLR, The Netherlands
John Crassidis	University at Buffalo, USA
Jörg Dittrich	DLR, Germany
Chris Edwards	University of Leicester, UK
Alexej Efremov	Moscow Aviation Institute, Russia
Patrick Fabiani	ONERA, France
Walter Fichter	Universität Stuttgart, Germany
Luisella Giulicchi	ESA/ESTEC, The Netherlands
Martin Hagström	Swedish Defense Research Agency, Sweden
Florian Holzapfel	Technische Universität München, Germany
Eric Johnson	Georgia Institute of Technology, USA
Gertjan Looye	DLR, Germany
Karl Heinz Kienitz	Instituto Tecnologico de Aeronautica, Brazil
Philipp Kraemer	Eurocopter, Germany
Marco Lovera	Politecnico di Milano, Italy
Robert Luckner	Berlin Technical University, Germany

Felix Mora-Camino	ENAC, France
Janusz Narkiewicz	Warsaw University, Poland
Guillermo Ortega	ESA, Netherlands
Stephan Theil	DLR, Germany
Andrzej Tomczyk	Rzeszów University of Technology, Poland
Michel Verhaegen	TU Delft, The Netherlands
Martin Weiss	TNO, The Netherlands
Ali Zolghadri	Université Bordeaux 1, France

Preface

Following the successful 1st CEAS (Council of European Aerospace Societies) Specialist Conference on Guidance, Navigation and Control (CEAS EuroGNC) held in Munich, Germany in 2011, Delft University of Technology happily accepted the invitation of organizing the 2nd CEAS EuroGNC in Delft, The Netherlands in 2013. Starting with the lessons learnt from the CEAS EuroGNC 2011, the EuroGNC 2013 conference was realised by a local organising committee, chaired by Bob Mulder and consisting of Daniel Choukroun, Qiping Chu, Erik-Jan van Kampen, Coen de Visser and Bertine Markus. The EuroGNC 2013 which took place in Delft, The Netherlands, on April 10-12, 2013, fundamentally owed its success to its International Program Committee: a selected group of eminent scientists and engineers who were crucial in setting the high standards of the conference technical program, either by soliciting, authoring, or reviewing and selecting the final proceedings papers. The book you are reading now is an additional outcome of this committee's work: a selection of papers presented at the EuroGNC 2013.

The goal of the conference is to promote new advances in aerospace GNC theory and technologies for enhancing safety, survivability, efficiency, performance, autonomy and intelligence of aerospace systems using on-board sensing and computing systems. In modern guidance, navigation and control, digital computers were already applied in the mid 60s for the Apollo program. The Apollo Primary Guidance, Navigation and Control System (Apollo PGNCS) included even a Kalman filter for optimally estimating the position from on-board measurements using the Apollo Guidance Computer (AGC) that was introduced in 1966. Space technologies have always been a drive for innovations in civil and military aeronautical applications. In fact the AGC was used in the first experimental digital Fly-By-Wire (FBW) system installed into an F-8 Crusader to demonstrate the practicality of digital computer driven FBW in 1972. The results led to a series of applications in military aircraft at the time including the Space Shuttle digital FBW in the 1980s. In civil aviation, Europe took a next revolutionary step by introducing digital FBW flight control in the Airbus A320 in 1987. Boeing later on joined this development for their B777. In guidance and navigation, by applying digital computers as well, equally impressive steps forward were achieved. Precise navigation using the Global Positioning System (GPS) with millimetre accuracy for space applications and centimetre accuracy for aeronautical applications is one of the most striking and challenging examples. The efforts from aerospace GNC scientists and engineers have made current GNC systems more accurate and robust than ever before, not only due to the introduction of new hardware but also of theoretical advancements in GNC algorithms. As digital computers become more and more powerful, scientists and engineers in aerospace GNC have virtually unlimited opportunities to respond to new challenges dictated by the higher and higher requirements from aerospace industries.

A great push for new developments in GNC are the ever higher safety and sustainability requirements in aviation. Impressive progress was made in new research fields such as sensor and actuator fault detection and diagnosis, reconfigurable and fault tolerant flight control, online safe flight envelop prediction and protection, online global aerodynamic model identification, online global optimization and flight upset recovery. All of these challenges depend on new online solutions from on-board computing systems. Scientists and engineers in GNC have been developing model based, sensor based as well as knowledge based approaches aiming for highly robust, adaptive, nonlinear, intelligent and autonomous GNC systems. Although the papers presented at the conference and selected in this book could not possibly cover all of the present challenges in the GNC field, many of them have indeed been addressed and a wealth of new ideas, solutions and results were proposed and presented.

GNC scientists and engineers in Europe benefit from their long history in mathematics and physics and their associated education systems through centuries. Many theoretical developments in GNC have found their common foundations based on theories developed in Europe far before the emergence of GNC problems. Although the European countries have their own research styles and foci , European Framework Programmes for Research and Technological Development, and the Group for Aeronautical Research and Technologies in EURope GARTEUR have brought all EU nations together in sharing their knowledge and experience. The GNC community in Europe benefits also from international cooperation with the United States, the Russian Federation and the BRIC countries where aerospace is a booming business.

The organization of the CEAS EuroGNC 2013 would have been impossible without the strong support of many people and communities. On behalf of the Local Organization Committee of CEAS EuroGNC 2013, we would like to take the opportunity to thank all contributors to the conference. These contributors are: Council of European Aerospace Societies CEAS, the organizers of the first CEAS EuroGNC 2011 in particular also DGLR, the Faculty of Aerospace Engineering of Delft University of Technology, the European Conference on Aerospace Sciences EUCASS, the American Institute of

Aeronautics and Astronautics AIAA, the Institute of Electrical and Electronic Engineers IEEE, the European Collaborative Dissemination of Aeronautical research and applications E-CAero, all members of the CEAS EuroGNC 2013 International Program Committee, all reviewers of technical papers, the ‘Nederlandse Vereniging voor Luchtvaarttechniek’ NVvL, the ‘Nederlandse Vereniging voor Ruimtevaart’ NVR, the Delft University of Technology, and the City of Delft.

Last but not least, we are delighted to acknowledge the prominent contribution of the conference Secretary, Ms. Bertine Markus, to the success of the conference.

The papers in the book are divided into four parts based on the four technical tracks of the conference: Guidance and Control, Navigation and Estimation, Atmospheric Applications and Space Applications. Some papers from invited sessions and papers from the graduate student competition have also been selected.

The Editors
Delft, The Netherlands
April, 2013

Editorial Committee

Editors-in-Chief

Dr. Qiping Chu

Dr. Qiping Chu obtained his Ph.D degree in aerospace engineering from the Faculty of Aerospace Engineering, TU Delft, The Netherlands in 1987. He was appointed as a research associate professor at the Centre for Space Science and Applications Research, Chinese Academy of Sciences from 1987 to 1991 and re-joined the Faculty of Aerospace Engineering, TU Delft since October 1991 as an assistant professor. Dr. Chu is presently an associate professor and the head of Aerospace Guidance, Navigation and Control Cluster at the Section of Control and Simulation, Faculty of Aerospace Engineering, TU Delft.

The research field of Dr. Chu is ranged from adaptive control, nonlinear control, robust control, and intelligent control to precise integrated navigation, system model identification and nonlinear optimisation for aerospace vehicles. He has authored and co-authored more than 200 journal, conference and book chapter publications in the related field. He has been the responsible teacher for courses on Aircraft Advanced Flight Control and Spacecraft Attitude Dynamics and Control. Dr. Chu was the designer of the attitude control system for the third Dutch satellite Sloshsat launched successfully in February 2005 and has been the supervisor for more than 15 Ph.D students since 2000.

Prof. emeritus dr ir J.A. (Bob) Mulder

Prof. emeritus dr ir J.A. (Bob) Mulder followed an MSc track of the Faculty of Aerospace Engineering of Delft University of Technology preparing for a career as experimental pilot. He received his license as airline transport pilot in 1967 and the MSc degree in 1968, his doctors degree in 1986. He was appointed chair of the Control and Simulation Division in the Faculty of Aerospace Engineering in 1989. He founded the Institute for research in Simulation, Motion and Navigation Technologies (SIMONA) and the institute for Aerospace Software and Technologies (ASTI). He is (co)author of more than 200 conference and journal papers on subjects ranging from human-machine interface design, cybernetics and pharmacokinetics to identification, estimation and advanced flight control. He is captain (rtd) on the Boeing 757-200 and Boeing 767-300 and has logged more than 9000 flight hours.

Associate Editors

Dr. Daniel Choukroun

Dr. Daniel Choukroun received the B.Sc. (summa cum laude), M.Sc., and Ph.D. degrees in 1997, 2000, and 2003, respectively, from The Technion – Israel Institute of Technology, Faculty of Aerospace Engineering. He also received the title Engineer under Instruction from the Ecole Nationale de l'Aviation Civile, France, in 1994.

From 1998 to 2003, he was a teaching and research assistant in the field of automatic control at The Technion—Israel Institute of Technology. From 2003 to 2006, he has been a postdoctoral fellow and a lecturer at the University of California at Los Angeles in the Department of Mechanical and Aerospace Engineering. From 2006 to 2010, he has been with the Department of Mechanical Engineering at Ben-Gurion University of the Negev, Israel. In 2010, he joined the Faculty of Aerospace Engineering at Delft University of Technology, The Netherlands, where he is currently Assistant Professor in Space Systems Engineering.

Dr. Choukroun received the Miriam and Aaron Gutwirth Special Excellency Award for achievement in research from The Technion—Israel Institute of Technology. His research interests are in optimal estimation and control theory with applications to aerospace systems. He is a member of the AIAA Technical Committee on Guidance Navigation and Control and Associate Editor in GNC for the IEEE Transactions in Aerospace and Electronic Systems.

Dr.ir. Erik-Jan van Kampen

Dr.ir. Erik-Jan van Kampen obtained his BSc-degree in Aerospace Engineering at Delft University of Technology in 2004, his MSc-degree in Control and Simulation in 2006, and a PhD-degree in Aerospace Engineering in 2010. The topic of the PhD-research was interval optimization and its application to aerospace problems. The PhD research was performed at the Aerospace Software and Technologies Institute in The Netherlands.

He is currently employed as assistant professor at the Control and Simulation division at Delft University of Technology, where he is giving courses on automatic flight control system design and advanced flight control methods. Research interests are: Knowledge based control (reinforcement learning), global nonlinear optimization methods and interval analysis.

Dr.ir. Coen de Visser

Dr.ir. Coen de Visser obtained his BSc-degree in Aerospace Engineering at Delft University of Technology in 2006, his MSc-degree in Control and Simulation in 2007, and a PhD-degree in Aerospace Engineering in 2011. The topic of the PhD-research was the development of a new method for system identification using multivariate splines.

From 2011 till 2012, he was a post-doctoral fellow at the Delft Center for Systems and Control where he worked on novel methods for distributed system identification.

Currently, he is an assistant professor at the Control and Simulation Division of the Faculty of Aerospace Engineering at Delft University of Technology, where he gives courses on aircraft system identification. His research interests are: system identification, multivariate spline theory, flight dynamics, distributed optimization, and nonlinear control theory.

Dr.ir. Gertjan Looye

Gertjan Looye graduated from the Delft University of Technology, Faculty of Aerospace Engineering in 1996 and received his Ph.D. from the same university in 2008. His main research interests are modern flight control design techniques, flight dynamics modelling, flight loads, and aeroelasticity. Applications include nonlinear flight control of transport, high-performance, and unmanned aircraft, flight loads analysis of elastic aircraft, and on-ground control of transport aircraft. Gertjan has been a research engineer at the German Aerospace Center (DLR) since 1997. He currently is acting head of the Aircraft Systems Dynamics department at the Institute of System Dynamics and Control, which is part of the Robotics and Mechatronics Center in Oberpfaffenhofen.

Contents

Part I: Guidance and Control

Part II: Estimation and Navigation

Part III: Atmospheric Applications

Part IV: Space Applications

Part I
Guidance and Control

Linear Parameter Varying Control of an Agile Missile Model Based on the Induced L_2-norm Framework

Raziye Tekin and Harald Pfifer

Abstract. This paper deals with the application of a linear parameter varying (LPV) controller synthesis for a modern air defense missile model. The model represents a challenging control problem due to the wide operation range. First, an LPV model of the missile is constructed via a novel approach of function substitution. Then, an LPV controller is designed based on the induced L_2-norm framework. A mixed sensitivity weighting scheme is applied to specify the performance requirements. In order to fulfill various time and frequency domain criteria, a multiobjective optimization is used to tune the weighting functions of the mixed sensitivity weighting scheme. Finally, the robustness and performance of the controller is evaluated by nonlinear simulations.

1 Introduction

Tactical missiles operate over a large flight envelope. Moreover, they need to be able to perform rapid maneuvers leading to fast variations in the flight conditions. Hence, a major requirement on the control system is to be able to retain good performance despite these fast varying parameters. An autopilot designed on a set of linear models over different flight conditions and ad hoc scheduling seems unsuitable to fulfill this demand. Such a design always assumes sufficiently slow parameter variations. In contrast to this classical approach, the LPV framework introduced in [1] can directly deal with fast varying scheduling parameters.

In this paper, first a brief theoretical background is given including the derivation of LPV systems and the controller synthesis in the induced L_2-norm framework. Then, the notions of generalized plant and mixed sensitivity weighting schemes are introduced. Afterwards, an LPV model is obtained for the considered nonlinear

Raziye Tekin · Harald Pfifer
Institute of System Dynamics and Control,
German Aerospace Center - DLR, Muenchner Str. 20, 82234 Wessling, Germany
e-mail: {raziye.tekin,harald.pfifer}@dlr.de

missile based on a function substitution proposed in [2]. A comparison of classical Jacobian linearization approach to obtain LPV model and function substitution is given to show the effectiveness of the latter method. A state feedback LPV controller is designed for the function substitution based LPV model using the methods proposed in [1] to track the normal acceleration commands. The controller objectives are defined in the frequency and time domain. In order to fulfill the requirements, the weightings are parameterized related to the system dynamics regarding the closed loop behavior of the LPV system. A multiobjective optimization is used to tune the parameters of the weightings. Finally, the performance and robustness of the resulting LPV controller is assessed using nonlinear simulations.

2 Theoretical Background

In this section, linear parameter systems and methods to obtain LPV systems are introduced. Then, a brief overview of the concept of generalized plant is presented. Finally, the solution to the state feedback LPV synthesis in the L_2-norm framework is given.

2.1 LPV Systems

LPV systems are defined as systems which are linear in $[x^T \ u^T]^T$ but nonlinear in some exogenous time varying parameters $\rho(t) : \mathscr{R}^+ \to \mathscr{P}$ as shown in Eq. 1.

$$\begin{bmatrix} \dot{x} \\ \hline y \end{bmatrix} = \left[\begin{array}{c|c} A(\rho(t)) & B(\rho(t)) \\ \hline C(\rho(t)) & D(\rho(t)) \end{array}\right] \begin{bmatrix} x \\ \hline u \end{bmatrix} \tag{1}$$

For most physical applications, the parameter variation rate is bounded i.e. $\dot{\rho}(t) : \mathscr{R}^+ \to \dot{\mathscr{P}}$ with

$$\dot{\mathscr{P}} := \{q \in \mathscr{R}^{n_p} \mid |q_i| < \nu_i \ , \ i = 1, \dots, n_p\} \tag{2}$$

It shall be pointed out that an LPV system reduces to an LTI (linear time invariant) system if ρ is constant and it reduces to an LTV (linear time varying) system when ρ is along a predefined trajectory. In contrast to LTV systems, the parameter trajectory is not now a priori but assuming to be online measurable for LPV systems. Hence, the following synthesis is not performed along a trajectory $\rho(t)$ with $\dot{\rho}(t)$ but over the corresponding parameter spaces represented by $p \in \mathscr{P}$ with $q \in \dot{\mathscr{P}}$ as given in Eq. 3.

$$\begin{bmatrix} \dot{x} \\ \hline y \end{bmatrix} = \left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right] \begin{bmatrix} x \\ \hline u \end{bmatrix} \tag{3}$$

2.1.1 Derivation of LPV Systems

There are various methods in literature to obtain LPV systems from nonlinear systems such as Jacobian linearization, function substitution and state transformation.

Detailed information about derivation of LPV systems can be found in [3]. In this section, only the two methods which are applied in this study are introduced.

Jacobian Linearization: This method is the most common methodology used to obtain LPV systems. It requires trimming at a set of equilibrium points and linearizing at equilibrium points which must span the operation space of the system. There are various examples using Jacobian linearization to derive LPV systems and design LPV controllers, e.g [4]. While it is the most widespread, it is generally not possible to acquire the transient behavior of the nonlinear system. See [5] for the shortcomings of this approach.

Function Substitution: This approach has been first presented in [6] and further enhanced in [2] and [3]. Unlike the Jacobian linearization approach, this method does not rely on a set of equilibrium points. It is essentially only an analytic transformation of the nonlinear differential equations. Hence, it is possible to capture the transient behavior of a nonlinear system well. Details of the chosen analytic transformation for the missile model are presented in Section 3.1.

2.2 LPV Controller Synthesis

In the presented LPV controller synthesis, the requirements of the closed loop system are specified using induced L_2-norm (i.e. input/output gain) performance objectives. For more detailed description of the method see [1].

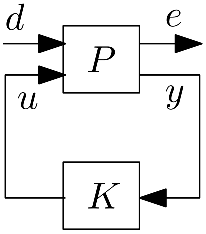


Fig. 1 Generalized plant

In Fig. 1, a generalized plant is presented where P represents the LPV plant including weightings and K the controller. The synthesis problem is to find a controller that minimizes the closed loop induced L_2 gain γ from the performance inputs d to performance outputs e as described below:

$$\min_{K} \|\mathscr{F}_l(P,K)\|_{\mathscr{L}_2 \to \mathscr{L}_2} \tag{4}$$

s.t. $\mathscr{F}_l(P,K)$ is stable for all admissible trajectories $\rho(t)$ with bounded parameter variation rate $\dot{\rho}(t)$. $\mathscr{F}_l$ describes the lower fractional transformation, as seen in Fig. 1.

The synthesis problem Eq. 4 can be solved by applying a generalized version of the Bounded Real Lemma for LPV systems. The Bounded Real Lemma provides an upper bound on the induced L_2-norm of a given LPV system for the bounded parameter variation rate $\dot{\rho}(t)$. In order to shorten the following notation, a differential operator $\partial X(p,q)$ is introduced as in [7]. For continuously differentiable $X(p)$, the differential operator $\partial X(p,q)$ is defined as $\partial X(p,q)=\sum_i^{n_p} \frac{\partial X(p)}{\partial p_i} q_i$. With this choice of $\partial X(p,q)$ represents the first time derivative of $X(\rho(t))$ for any trajectory $\rho(t)$.

Bounded Real Lemma: Let P be an LPV system of the form Eq. 3 then P is exponentially stable and $\|P\|_{\mathscr{L}_2 \to \mathscr{L}_2} < \gamma$ for all $\forall (p,q) \in \mathscr{P} \times \dot{\mathscr{P}}$, if $\exists X(p) > 0$ such that $\forall (p,q) \in \mathscr{P} \times \dot{\mathscr{P}}$

$$\begin{bmatrix} A(p)X^T(p) + X(p)A^T(p) + \partial X(p,q) & X(p)B(p) \\ B^T(p)X(p) & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} C^T(p) \\ D^T(p) \end{bmatrix} \begin{bmatrix} C(p) & D(p) \end{bmatrix} < 0 \tag{5}$$

2.2.1 State Feedback Synthesis

With the help of the Bounded Real Lemma, the synthesis problem defined in Eq. 4 can be turned into a semidefinite program. In this study, only the state feedback problem is considered, as the missile example belongs to this class. In this case, the generalized plant can be written as Eq. 6.

$$\begin{bmatrix} \dot{x} \\ e_1 \\ e_2 \\ y \end{bmatrix} = \begin{bmatrix} A(p) & B_1(p) & B_2(p) \\ C_{11}(p) & 0 & 0 \\ C_{12}(p) & 0 & I \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \tag{6}$$

Introducing new variables $R(p) = \gamma^{-2} X(p)$, $\partial R(p,q) = -\gamma^{-2} X^{-1} \partial X(p,q) X^{-1}(p)$ and $\hat{A}(p) = A(p) - B_2(p)C_{12}(p)$, and applying the Bounded Real Lemma on the closed loop $F_l(P,K)$, the controller synthesis problem Eq. 4 becomes:

$$\min_{R(p)} \gamma, \text{ s.t. } \exists (p,q) \in \mathscr{P} \times \dot{\mathscr{P}}$$

$$R(p) > 0$$

$$\begin{bmatrix} R(p)\hat{A}(p)^T + \hat{A}(p)R(p) - \partial R(p,q) - B_2(p)B_2^T(p) & R(p)C_{11}^T(p) & \gamma^{-1}B_1(p) \\ C_{11}(p)R(p) & -I & 0 \\ \gamma^{-1}B_1(p)^T & 0 & -I \end{bmatrix} < 0 \tag{7}$$

It is not possible to solve Eq. 7 over the whole function space of $R(p)$. Hence, $R(p)$ has to be restricted to a finite dimensional space. The function $R(p)$ is defined by a set of basis function $g(p)$ of the form:

$$R(p) = \sum_j g_j(p) R_j \tag{8}$$

Note that the constraints in Eq. 7 are actually infinite dimensional due to their dependency on p and q. Since q enters in Eq. 7 only affinely and the set $\dot{\mathscr{P}}$ is a polytype, it is sufficient to check the constraints on the vertices of $\dot{\mathscr{P}}$. To deal with dependency on p, a grid over P is generated and the constraints are only checked on this grid. This approach is common in literature, see for example [1]. Finally, the state feedback control law is given by

$$K(p) = -(B_2^T(p)R^{-1}(p) + C_{12}(p)), \tag{9}$$

where the controller $K(p)$ is a continuous matrix function of p.

2.2.2 Generalized LPV Plant

The mixed sensitivity weighting scheme is a well known and common concept to define the controller objectives in the induced L_2-norm framework [8]. The generalized plant which is the plant augmented by the weighting functions is depicted in Fig. 2. In this weighting scheme, the closed loop sensitivity function $S = (I+GK)^{-1}$ and KS are shaped by W_y and W_u respectively. Due to the choice of the performance input d as seen in Fig. 2, the sensitivity function S and KS are weighted by the plant.

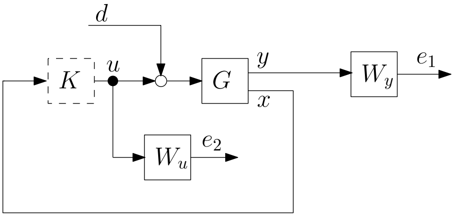


Fig. 2 Structure of mixed sensitivity scheme for generalized plant

The transfer functions of W_u and W_y are given in Eq. 10 where ω_{ro} is the desired roll-off frequency of the controller input, k_u the low frequency gain of W_u, ω_b is the desired minimum bandwidth of the closed system and k_y is high frequency gain of W_y to penalize the overshoot which are further going to be optimized for LPV controller design.

$$W_u = 100k_u \frac{s+\omega_{ro}}{s+100\omega_{ro}}, \quad W_y = k_y \frac{s+\omega_b}{s+0.005} \tag{10}$$

3 LPV Controller Design for the Missile Model

In the following section, the design process of the LPV controller is described for longitudinal motion of a missile. First, an LPV model of the nonlinear missile dynamics is derived by function substitution. Jacobian linearized and function substitution based LPV models are compared with nonlinear simulation. A mixed sensitivity weighting scheme is applied for the controller design. After defining the structure of the weighting functions given in Section 2.2.2, they are optimized. Then, an LPV controller is designed based on the approach described in Section 2.2.

3.1 LPV Modeling

The model and aerodynamic database is taken from [9] which includes all flight regime including boost phase with thrust vector control. However, in this paper, only the post boost phase is studied. The body axes longitudinal motion of the tail controlled air defense missile, extracting flexible and high angle of attacks phenomenas, can be described by the following differential equations:

$$\dot{w} = \frac{1}{m}(F_z - qu), \;\; \dot{q} = \frac{M}{I_y}, \tag{11}$$

where m is the missile mass, I_y the inertia in principal axes, q the pitch rate, α the angle of attack and u the longitudinal velocity. Since in this study only the post boost phase is studied, mass, inertia and center of gravity are constant and the gravitational force is neglected. All the considered forces and moments are aerodynamic. The related aerodynamic forces and moments are modeled as:

$$\begin{aligned} F_z &= Q(Ma,h)SC_z(Ma,\alpha,\delta_e) \\ M &= Q(Ma,h)SlC_m(Ma,\alpha,\delta_e) \end{aligned} \tag{12}$$

Here, h is the altitude and Q, S, Ma, δ_e and l denote dynamic pressure, reference surface, Mach number, elevator deflection and reference length, respectively. The aerodynamic coefficients can be decomposed into the following structure:

$$\begin{aligned} C_Z &= C_{Z0}(Ma,\alpha) + C_{Z\delta_e}(Ma,\alpha)\delta_e \\ C_M &= C_{M0}(Ma,\alpha) + C_{Mq}(Ma,\alpha)lq/(2V) + C_{M\delta_e}(Ma,\alpha)\delta_e \end{aligned} \tag{13}$$

where V is absolute velocity and $C_{Z\delta_e}$ and $C_{M\delta_e}$ are the derivatives of the aerodynamic coefficients with respect to the elevator deflection angle. Note that in the aerodynamic data set of [9] C_z and C_M are actually not affine in δ_e. Still, this assumption is well justifiable as they are almost affine.

Using Eq. 12, Eq. 13 and $u = V\cos\alpha$, the nonlinear missile dynamics can be written as

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & V\cos\alpha \\ 0 & \frac{QSl^2}{2VI_y}C_{Mq}(Ma,\alpha) \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \underbrace{QS\begin{bmatrix} \frac{1}{m}C_{Z0}(Ma,\alpha) \\ \frac{l}{I_y}C_{M0}(Ma,\alpha) \end{bmatrix}}_{\hat{f}(Ma,\alpha)} + \begin{bmatrix} \frac{QS}{m}C_{Z\delta_e}(Ma,\alpha) \\ \frac{QSl}{I_y}C_{M,\delta_e}(Ma,\alpha) \end{bmatrix} \delta_e \tag{14}$$

The system Eq. 14 is already almost in the form of an LPV model. The only remaining step is a function substitution of $\hat{f}(Ma,\alpha)$, such that $\hat{f}(Ma,\alpha) = A_1(Ma,\alpha)w$. For this purpose the method proposed in [2] is used. It exploits the relation $w = V\sin\alpha$ by introducing

$$C_{Zbar} = \begin{cases} 0, & \text{if } \alpha = 0 \\ C_{Z0}/(\sin\alpha), & \text{otherwise} \end{cases}, \quad C_{Mbar} = \begin{cases} 0, & \text{if } \alpha = 0 \\ C_{M0}/(\sin\alpha), & \text{otherwise} \end{cases} \tag{15}$$

With Eq. 15 the missile dynamics can be written in LPV form with the parameter vector $[Ma\ \alpha]$ as

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} QSC_{zbar}/mV & Vcos(\alpha) \\ QSlC_{mbar}/I_yV & QSl^2C_{mq}/2I_yV \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} QSC_{z\delta_e}/m \\ QSlC_{m\delta_e}/I_y \end{bmatrix} \delta_e \tag{16}$$

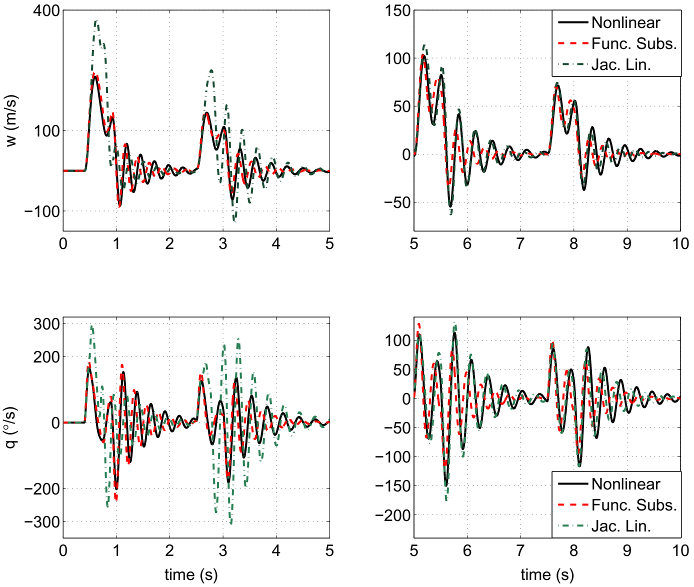


Fig. 3 Comparison of linearly scheduled Jacobian linearized and function substitution based LPV systems

A comparison between a Jacobian linearized and a function substitution based LPV system is presented in Fig. 3. In the simulation, the velocity is decreasing from Mach 2.5 to Mach 1 and the angle of attack range is between 0 and 20 degrees. The Jacobian linearized LPV system seems better at the low angle of attack regime. The worse accuracy of the function substitution based model in this region most likely stems from the affine approximation of the dependence of the aerodynamic coefficients on the elevator deflection angle. This assumption is not required in the

Jacobian linearization case. Although Jacobian linearized LPV system has this advantage, when the whole flight regime including the high angle of attack regime is examined, it can be said that the function based LPV system is the better approximation. Hence, the controller design will be applied onto the function substitution based LPV system.

3.2 Controller Design

Based on the function substitution LPV model described in the previous section, an induced L_2-norm LPV controller is designed for the nonlinear missile model. The weighting scheme introduced in Section 2.2.2 is used to specify the performance requirements of the controller. The major aim of the controller is to provide good tracking performance in the acceleration (a_n). First of all, a state transformation is applied on the system, in order to turn the synthesis into a state feedback problem. The new states are the acceleration (a_n) and the pitch rate q which can both be measured by an inertial measurement unit (IMU). The controller synthesis is conducted using 64 grid points to cover the flight envelope which are:

- $\alpha = [0:3:21]^\circ$
- $Ma = [1 \;\; 1.2 \;\; 1.3 \;\; 1.4 \;\; 1.5 \;\; 1.6 \;\; 2 \;\; 2.4]$

It shall be emphasized that the Mach number gridding needs to be finer in the transonic region, where the system dynamics are changing critically. The parameter variation rates are bounded by $\left|\dot{Ma}\right| \leq 0.2(\text{Ma/s})$ and $|\dot{\alpha}| \leq 100(^\circ/\text{s})$. These values are chosen with respect to the flight envelope of the missile [10].

The structure of the parameter dependent R is chosen as:

$$R(Ma,\alpha) = R_0 + Ma_i R_1 + Ma_i^2 R_2 + \alpha_i R_3 + \alpha_i^2 R_4$$

This choice of $R(Ma,\alpha)$ is motivated by using a few simple basis functions to keep the computational burden of the synthesis low.

3.3 Optimization of the Weighting Functions

Finding suitable weighting functions is not an easy task. In order to handle this problem, the weighting functions are parameterized and optimized over all the grid points. The roll-off frequency (w_{ro}) and the low frequency gain (k_u) of W_u are parameterized with respect to the most dominant flight parameter, namely the velocity which has a large impact on the dynamics.

$$\omega_{ro} = w_u(1) + w_u(2)Ma_i, \quad k_u = w_u(3) + w_u(4)Ma_i \tag{17}$$

The corresponding control effort penalty is depicted in Fig. 4. As seen in the Fig. 4, as the Mach number increases, the control effort is penalized more because control power is higher than at low Mach numbers, i.e. the missile needs less elevator deflection angle to achieve the same accelerations at high Mach numbers.

The gain and bandwidth of W_y is not only parameterized with respect to the Mach number but also the angle of attack. Moreover, the minimum bandwidth of the closed system is parameterized as a function of the natural frequency of the missiles short period dynamics (ω_n). With this parameterization less bandwidth is demanded at low velocities, where the open loop system is slower than at high velocities, see Fig. 4. Similarly, the tracking performance is less penalized at higher angle of attacks, because the open loop bandwidth is lower in comparison to lower angle of attacks.

$$\omega_b = \omega_n(Ma,\alpha)w_y(1), \quad k_y = w_y(2) + w_y(3)Ma_i - w_y(4)\alpha_i \tag{18}$$

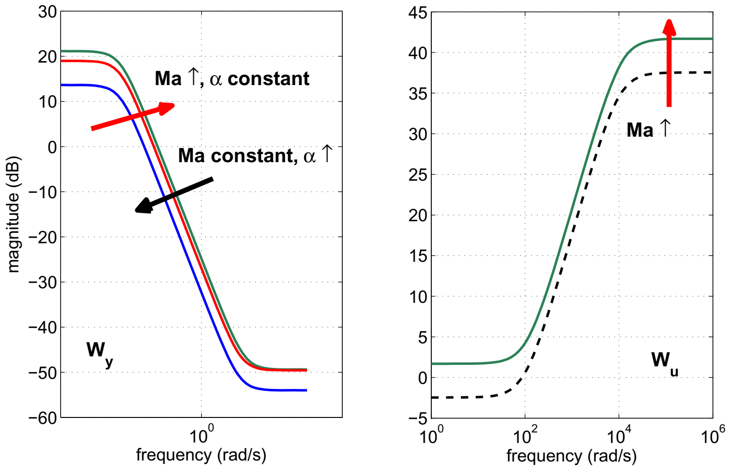


Fig. 4 Performance penalty W_y and control effort W_u

With the weighting functions W_u and W_y only a general trend can be specified for the controller objectives, as shown in Fig. 4. Detailed performance and robustness requirements are usually not given in terms of input/output gains but as a mixture of frequency and time domain criteria. For the present missile model these requirements are defined in Table 1. Note that the demanded settling and rise time depend on the Mach number to better exploit the capabilities of the missile. To ensure that the controller is robust, classical gain and phase margin requirement are used.

Using these criteria (Table 1), an optimization problem is specified. The tuners of the optimization are the free parameters in the weighting functions, see Eq. 17 and Eq. 18. The aim of the optimization is to find suitable weighting functions that minimize rise and settling time while not violating any of the specified constraints. The problem is solved for each point of the synthesis grid using the Matlab optimization environment MOPS [11]. At each grid point, it successfully surpasses the demand values and satisfies the constraints. Table 2 presents the optimized tuners, w_u and w_y, respectively.

Table 1 Optimization parameters

Inequality Constraints	Value	Demands	Value
Phase Margin	$50-90°$	Settling Time	$0.5/Ma$ s
Gain Margin	10-20 dB	Rise Time	$0.2/Ma$ s
Undershoot	$\leq 20\%$		
Overshoot	$\leq 10\%$		
Gamma Value	[1 2.9]		

Table 2 Optimization results of weighting tuners

W_y	Value	W_u	Value
$w_y(1)$	0.55	$w_u(1)$	90
$w_y(2)$	0.1	$w_u(2)$	9.125
$w_y(3)$	0.995	$w_u(3)$	0.4237
$w_y(4)$	0.018	$w_u(4)$	0.33

4 Nonlinear Simulations

To assess the robustness performance of the design, $\pm 10\%$ uncertainty are considered in mass, center of gravity, inertia, aerodynamic forces and moments. Nonlinear simulations are performed for every combination of minimum and maximum values of these uncertainties. This results in 512 simulations.

The simulation results are presented in Fig. 5 and Fig. 6. During the simulation, step signals in the reference acceleration are applied between 250-50 m/s^2 depending on the velocity, Fig. 5. All of the nominal results are given in bold lines. Overall, the degradation of the performance due to the considered uncertainties is very low.

The scheduling parameters are illustrated in Fig. 6(a). During the simulation the angle of attack is changing from 0 to 16 degrees. The velocity is decreasing from $Ma = 2.5$ to $Ma = 1$. In addition, the altitude is also changing from 2500 to 4000 meters. Note that the controller is not scheduled with respect to these altitude changes. The control surface deflections are depicted in Fig. 6(b). As seen in the figure, the deflection angles are between ± 18 degrees which is within the limits of the considered actuator system. In the transonic region, the control effort increases drastically for the uncertain cases in comparison to the nominal simulation. In the worst case 15 degrees of deflection have to be applied instead of 5 degrees to achieve the same performance. However, 15 degrees are still within the limits of the actuator system. The difference in the rest of the flight is not very noticeable.

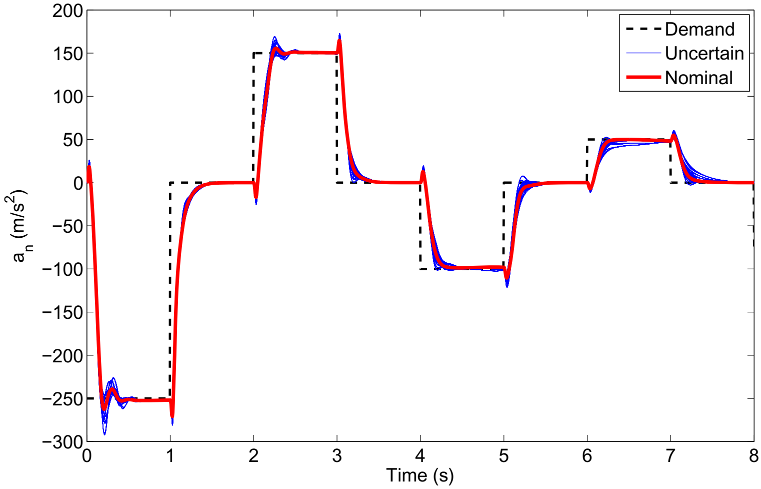
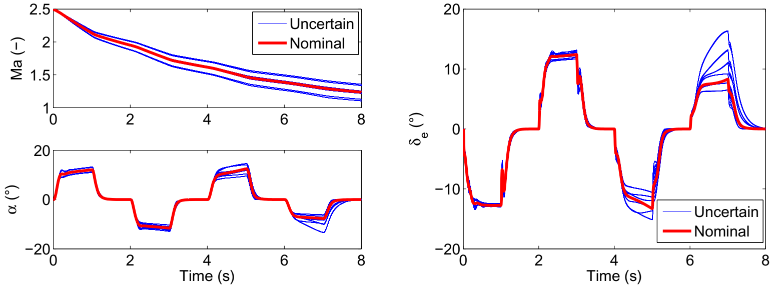


Fig. 5 Tracking performance of the LPV controller



(a) Scheduling parameters (b) Elevator deflection

Fig. 6 Time histories of the nonlinear simulations

5 Discussion and Conclusion

An LPV controller synthesis for a tail controlled missile is presented in this paper. First, the nonlinear missile dynamics are brought into an LPV form using the innovative method of function substitution, see [2]. It is shown that the resulting LPV system captures the behavior of the nonlinear missile over wide operation conditions better than a classical LPV modeling approach based Jacobian linearization. Second, a state feedback LPV controller in the induced L_2-norm framework (see [1]) is designed for the LPV model. A multiobjective optimization problem is used to tune the weighting functions for the controller design. Robust performance simulations are conducted with $\pm 10\%$ uncertainty on all relevant flight parameters. The results of these nonlinear simulations show that the proposed controller synthesis method is capable of providing the required level of performance for the missile model. In the future, the complete flight envelope shall be covered by the LPV controller including the very challenging boost phase.

References

1. Wu, F.: Control of Linear Parameter Varying Systems. PhD thesis, University of California (1995)
2. Pfifer, H.: Quasi-lpv model of a ndi-controlled missile based on function substitution. In: AIAA Guidance and Control Conference (2012)
3. Marcos, A., Balas, G.: Development of linear-parameter-varying models for aircraft. Journal of Guidance, Control and Dynamics 27 (2004)
4. Pfifer, H., Hecker, S.: Lpv controller synthesis for a generic missile model. In: IEEE Conference on Control Applications (2010)
5. Leith, D.J., Leithead, W.: Comments on the prevalance of linear parameter varying systems. Technical report, Department Electronic and Electrical Engineering, University of Strathclyde, Glasgow (1999)
6. Wey, T.: Applications of linear parameter varying control theory. Master's thesis, University of California at Berkeley (1997)
7. Scherer, C., Weiland, S.: Linear matrix inequalities in control. Lecture notes, Delft University of Technology (2005)
8. Skogestad, S., Postlethwaite, I.: Multivariable Feedback Control. John Wiley and Sons (2005)
9. Tekin, R., Atesoglu, O., Leblebicioglu, K.: Modeling and vertical launch analysis of an aero- and thrust vectored surface to air missile. In: AIAA Atmospheric Flight Mechanics Conference (2010)
10. Tekin, R.: Design, modeling, guidance and control of a vertical launch surface to air missile. Master's thesis, Middle East Technical University (2010)
11. Joos, H.D., Bals, J., Looye, G., Schnepper, K., Varga, A.: A multi-objective optimization-based software environment for control systems design. In: Proceedings of the IEEE International Symposium on Computer Aided Control System Design (2002)

Improving the Performance of an Actuator Control Scheme during Saturation

Chang How Lo, Hyo-Sang Shin, Antonios Tsourdos, and Seung-Hwan Kim

Abstract. This paper first introduces a new control scheme for a four fin missile actuation system. Exiting missile autopilot systems generally compute aileron, elevation, and rudder commands since these three variables dominantly influence the roll, pitch, and yaw motion of the vehicle. These commands are distributed to four fin deflection commands and fin controller actuates the fins to track the defection command. The performance of such control schemes can be significantly degraded when fin actuators are saturated due to certain physical constraints, such as voltage, current, or slew rate limit. This paper analytically proves that the proposed control scheme mitigates this problem, so it outperforms the conventional control scheme in the tracking performance if an actuator is saturated. Without any actuator saturation, the performance of the proposed scheme is also proved to be equivalent to that of a conventional actuator scheme. Numerical simulations verify the superiority of the proposed scheme and the theoretical analysis.

1 Introduction

The performance of the actuation system plays a decisive role in determining the performance of the flight control system, especially for a highly manoeuvrable air vehicles [7]. The vehicles are generally controlled by fins of which defection produces aerodynamic force. However, in classical autopilot design, the autopilot produces virtual roll, pitch and yaw moment demands [2] instead of physical fin deflection commands. These moment demands are then ‘mixed’or allocated by an

Chang How Lo · Hyo-Sang Shin · Antonios Tsourdos
Cranfield University, College Road, Cranfield, Bedfordshire,
MK43 0AL, United Kingdom
e-mail: {c.lo,h.shin,a.tsourdos}@cranfield.ac.uk

Seung-Hwan Kim
ADD (Agency for Defense Development), Daejeon, Korea
e-mail: shadd@hanmail.net

control allocation algorithm to generate individual actuator commands. The actuator commands will typically be tracked by their individual actuator controllers. Common problems faced in the actuators are that they may be saturated due to their physical constraints such as voltage, current, or slew rate limit. These problems can result in the significant performance degradation of the flight control system, or worse still, destablize the entire system.

Many approaches have been proposed to tackle this problem. One way is to treat this as a control allocation problem, and methods such as redistributed pesudo inverse [8], dynamic control allocation [5], and direct allocation [4] have been devised to optimally handle these actuator constraints. However, many control allocation methods do not track the actuator's actual performance, and thus possible deviations from the desired autopilot virtual commands can happen whenever there are unexpected degradations in the physical actuator due to faults, or disturbances.

In the design aspect, numerous control methodologies have been researched on in designing control schemes to handle actuator saturations and constraints with considerations on the stability, domain of attraction, and performance of actuation system. Extensive reviews and design methodologies of them can be found in Ref. [1] and Ref. [6]. In a typical conventional actuator scheme, each actuator has a dedicated controller to track its assigned command from a outer control loop. Practical implementations of such schemes are commonly found; an example is Ref. [9].

However, many of these methodologies aim at the design of a single actuator, and not as an actuation system. This may not fully exploit the analytical redundancies found in many systems to improve on the performance, or enhance its robustness to faults.

In our previous study [7], a new alternative actuator control scheme for a four tail fin controlled missile is proposed to alleviate the performance degradation results from actuator saturation. To utilize the analytically redundant actuator, the proposed scheme regulates the error in the virtual moment space rather than the physical fin deflection space. This alternative approach contrasts with ideas from Ref. [10], where linear in-line filters are used to exploit the actuator redundancy space to alleviate input rate and magnitude saturations.

This paper extends on our previous work in Ref. [7] to theoretically analyse the performance of the proposed actuator control system, which presented simulation results without formal proof. The main aim of this analysis is to analytically show that the proposed scheme outperforms the conventional actuator control scheme under actuator saturation. The performance index for the analysis is defined as the magnitude of the actuator tracking error, since the smaller the tracking error is, the better the performance is. From the analysis, in unsaturated operation region, it is proved that the performance of the proposed scheme is equivalent to that of a conventional scheme. The superior performance of the proposed scheme is also proved when one actuator is saturated. Simulation results verifying the analysis is then shown.

The organization of the rest of the paper is as follows. A description of the conventional and proposed control schemes are presented in Section 2. The main results of the paper are presented in Section 3, where the two control schemes are compared analytically. In Section 4, the simulation setup is described, and simulation results verifying the analysis are shown. Finally, the paper is concluded in Section 5.

2 Description of Control Scheme

2.1 Conventional Control Scheme

Figure 1 shows a conventional actuator control architecture for a four tail fin controlled missile. A classical autopilot outputs virtual command $\boldsymbol{\delta_{mc}}$ in the roll, pitch and yaw moment space to the actuators. These commands are then allocated or mixed to individual actuator's controller for tracking by their local controller. The response achieved from actuator controllers in the moment space can be found by deallocating the individual actuator response as a measure of the control performance.

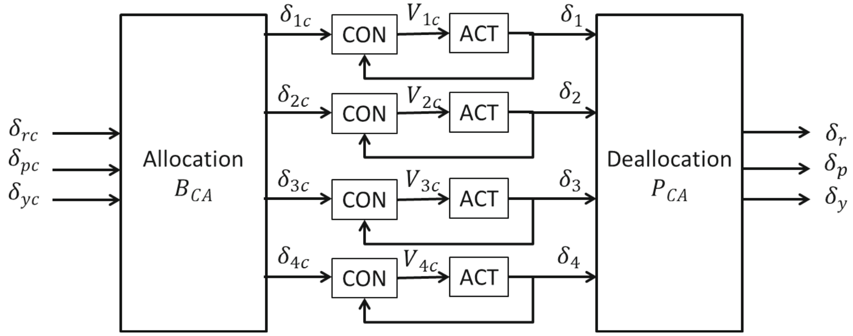


Fig. 1 Conventional Actuator Control Architecture

The closed loop transfer function for the conventional actuator control scheme for the autopilot demands is

$$\begin{aligned}\mathbf{G_o(s)} &= \frac{\boldsymbol{\delta_m}\mathbf{(s)}}{\boldsymbol{\delta_{mc}}\mathbf{(s)}} \\ &= \mathbf{P_{ca}}\left(\frac{\mathbf{K\,K_s\,G}}{\mathbf{I+K\,K_s\,G}}\right)\mathbf{B_{ca}}\end{aligned} \tag{1}$$

where $\boldsymbol{\delta_{mc}} = \begin{bmatrix} \delta_{rc} & \delta_{pc} & \delta_{yc} \end{bmatrix}^T$ is the virtual control demand from the autopilot,$\boldsymbol{\delta_m} = \begin{bmatrix} \delta_r & \delta_p & \delta_y \end{bmatrix}^T$ is the resultant response from the actuators in the virtual space, $\mathbf{K} = diag(K_i)$ is the actuator controller. The i^{th} actuator is described by two terms: a linear actuator model G_i, which is preceded by a physical voltage constraint K_{is}. The four actuators can be combined mathmatically as $\mathbf{G} = diag(G_i)$ and $\mathbf{K_s} = diag(K_{is})$.

K_{is} is the standard saturation function to describe the physical voltage input constraint, V_c. In frequency domain, the nonlinear saturation function can be represented as

$$K_{is} = 1, \qquad |V_c| \leq V_{max} \tag{2}$$

$$0 < K_{is} < 1, \qquad |V_c| > V_{max} \tag{3}$$

It is assumed there is no unstable pole zero cancellation in the system. For a four tail fin missile under consideration here, the control allocation matrix is

$$\delta = \mathbf{B_{ca}}\,\delta_{\mathbf{m}}$$
$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \delta_r \\ \delta_p \\ \delta_y \end{bmatrix} \tag{4}$$

and the deallocation matrix $\mathbf{P_{ca}}$ being

$$\mathbf{P_{ca}} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \tag{5}$$

with

$$\mathbf{P_{ca}B_{ca}} = \mathbf{I} \tag{6}$$

or $\mathbf{P_{ca}}$ being the pseudo inverse solution of $\mathbf{B_{ca}}$. The control allocation matrix is obtained by considering the resultant torque generated by each actuator's position from the aerodynamics point of view.

2.2 Proposed Control Scheme

Figure 2 shows the control scheme first proposed in our previous study [7]. Here, the actuator control scheme regulates the tracking error in the virtual moment space space before control allocation. This contrasts with the conventional control scheme regulating the physical tracking error of each individual actuator. The closed loop transfer function of the proposed scheme can be derived as

$$\mathbf{G_n}(\mathbf{s}) = \frac{\mathbf{K\,P_{ca}\,K_s\,G\,B_{ca}}}{\mathbf{I} + \mathbf{K\,P_{ca}\,K_s\,G\,B_{ca}}} \tag{7}$$

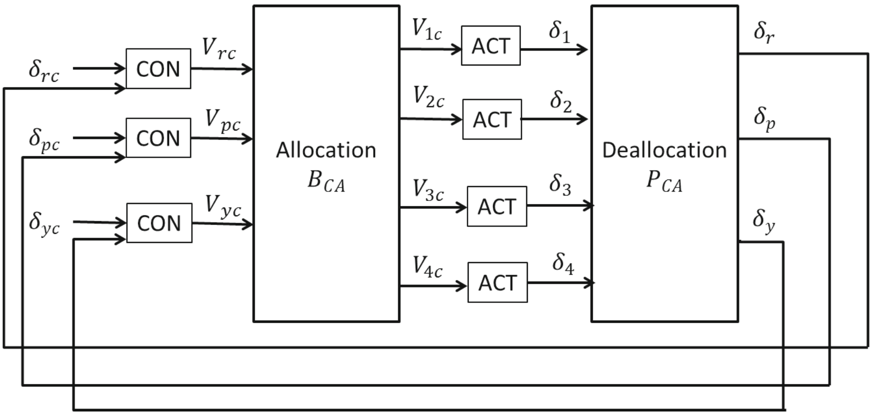


Fig. 2 Proposed Actuator Control Architecture

3 Comparison of Control Schemes

The comparison of the two control schemes is divided into two cases: linear (non-saturating) case, and when one actuator is saturated.

3.1 Non Saturation Case

Replacing K_{is} with unity gain in Equations 1 and 7, and assuming the actuators and their controllers are similar $G_i = G$ and $K_i = K$, it can be verified that both control schemes can be reduced to the following linear transfer function:

$$\frac{\delta_{\mathbf{m}}(\mathbf{s})}{\delta_{\mathbf{mc}}(\mathbf{s})} = \frac{GK}{GK+1}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The significance of this case is to show the two control schemes are equivalent during nominal operation, and proof that the improvement in performance of the proposed scheme shown later is not due to increase in gain. In practice, this may reduce the amount of design analysis needed to convert from the proposed scheme for any existing actuator scheme.

3.2 One Actuator Saturation Case

Next, the analysis is extended to the case when one actuator is saturated during operation. First, assume Actuator Number 4 is saturated. This can be represented by

$$\mathbf{K_s} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & K_{4s} \end{bmatrix} \tag{9}$$

where $K_{4s} < 1$. Substituting Equation 9 into Equation 7 for the proposed scheme, and rearranging with Equation 5 reveals

$$\begin{aligned}\delta_r &= \left(\frac{GK}{GK+1}\right)\delta_{rc} \\ &\quad + \left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\begin{bmatrix}1\\-1\\-1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix} \\ &= \left(\frac{GK}{GK+1}\right)\delta_{rc} \\ &\quad + \frac{1}{4}\left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\delta_{4c}\end{aligned} \tag{10}$$

$$\begin{aligned}\delta_p &= \left(\frac{GK}{GK+1}\right)\delta_{pc} \\ &\quad + \left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix} \\ &= \left(\frac{GK}{GK+1}\right)\delta_{pc} \\ &\quad - \frac{1}{4}\left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\delta_{4c}\end{aligned} \tag{11}$$

$$\begin{aligned}\delta_y &= \left(\frac{GK}{GK+1}\right)\delta_{yc} \\ &\quad + \left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix} \\ &= \left(\frac{GK}{GK+1}\right)\delta_{yc} \\ &\quad - \frac{1}{4}\left(\frac{K_{4s}-1}{K\left(G+3GK_{4s}\right)+4}\right)\delta_{4c}\end{aligned} \tag{12}$$

Equations 10 to 12 show the effect of actuator saturation. The first term on the right hand side of these equations is the nominal unsaturated performance of the actuator, while the second term is the additional dynamics introduced by the saturated actuator. For the conventional scheme, similar relationships can be obtained by manipulating Equation 1 in the same manner to obtain Equations 13 to 15:

$$
\begin{aligned}
\delta_r &= \left(\frac{GK}{GK+1}\right)\delta_{rc} \\
&\quad + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}1\\-1\\-1\end{bmatrix}^T \begin{bmatrix}\delta_{rc}\\ \delta_{pc}\\ \delta_{yc}\end{bmatrix} \\
&= \left(\frac{GK}{GK+1}\right)\delta_{rc} \\
&\quad + \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
\delta_p &= \left(\frac{GK}{GK+1}\right)\delta_{pc} \\
&\quad + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T \begin{bmatrix}\delta_{rc}\\ \delta_{pc}\\ \delta_{yc}\end{bmatrix} \\
&= \left(\frac{GK}{GK+1}\right)\delta_{pc} \\
&\quad - \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
\delta_y &= \left(\frac{GK}{GK+1}\right)\delta_{yc} \\
&\quad + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T \begin{bmatrix}\delta_{rc}\\ \delta_{pc}\\ \delta_{yc}\end{bmatrix} \\
&= \left(\frac{GK}{GK+1}\right)\delta_{yc} \\
&\quad - \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}
\end{aligned} \tag{15}
$$

Similarly, the effects of actuator 4's saturation can be accounted for the conventional actuator scheme. The second term on the right hand side of Equations 13 to 15 is the detrimental contribution by the saturation.

Comparing between the proposed and conventional schemes, it can be seen the numerator of the saturation dynamics for both schemes are the same at $(K_{4s}-1)$ from Equations 10 to 15. Now, comparing the denominator of conventional and proposed scheme, it can be seen that

$$(4GKK_{4s}+4) < (GK+3GKK_{4s}+4) \tag{16}$$

by noting that $K_{4s} < 1$ during actuator saturation. This implies the magnitude of the effect caused by the saturation term for the conventional scheme will be bigger than the proposed scheme. The alleviation in actuator saturation in the proposed scheme in this case is due to the availability of actuator redundancy, which enables the controller to increase the commands of the other non-saturated actuators.

The same results are obtained when saturations occur with other actuators in the control scheme. For the single actuator saturation case, the proposed control scheme's results in Equations 10 to 12 can be generalised to

$$\delta_{\mathbf{mc}} = \frac{GK}{GK+1}\left(\mathbf{I} + \mathbf{P_{ca}}\mathbf{D}\mathbf{B_{ca}}\right)\delta_{\mathbf{m}} \tag{17}$$

with the appropriate actuator being saturated, and the other K_{is} being equal to 1. For the proposed actuator scheme, $\mathbf{D}$ is defined as

$$\mathbf{D} = diag\left[4\left(\frac{K_{is}-1}{KG+3GKK_{is}+4}\right)\right] \tag{18}$$

Similarly, the corresponding single actuator saturation case using conventional actuator control scheme's results, described in Equations 13 to 15, can be generalised to Equation 17, with $\mathbf{D}$ defined as

$$\mathbf{D} = diag\left[4\left(\frac{K_{is}-1}{4GKK_{is}+4}\right)\right] \tag{19}$$

The generalizations here assumes the structure of $\mathbf{B_{ca}}$ and $\mathbf{P_{ca}}$, and their relationship described in Equations 4 to 6. Overall, one can see that the proposed scheme is superior to the conventional scheme, as the magnitude of $\mathbf{D}$ in Equation 18 for the proposed scheme is smaller than the magnitude of $\mathbf{D}$ in Equation 19 for the conventional scheme. Next, simulations results are presented.

4 Simulation

4.1 Simulation Model

The actuator plant used in the simulation is described in [7], which is shown in Figure 3. The plant is a typical DC motor with gearing, and explained in Ref. [3]. The parameters used in are detailed in Table 1.

A cascaded two loop controller shown in Figure 4 is used to provide good control performance of the actuator plant from Ref. [7]. The control law is

$$V_c(t) = K_p\left(\delta_c(t) - \delta(t)\right) - K_d\left(\dot{\delta}(t)\right) \tag{20}$$

where δ_{ci} is the commanded deflection angle, and $K_p = 6$ and $K_d = 0.02$ are gains for the control law. The reference command used in this simulation is a sinusoidal input

Table 1 Actuator plant parameters used in the simulation

Parameters	Values	Parameters	Values
K_T	0.303125	K_B	$5.7333x10^{-4}$
L	0.35×10^{-3}	H	0
R	0.933	V_{lim}	28
J	8.5354×10^{-7}	N	274
B	2.0835×10^{-6}		

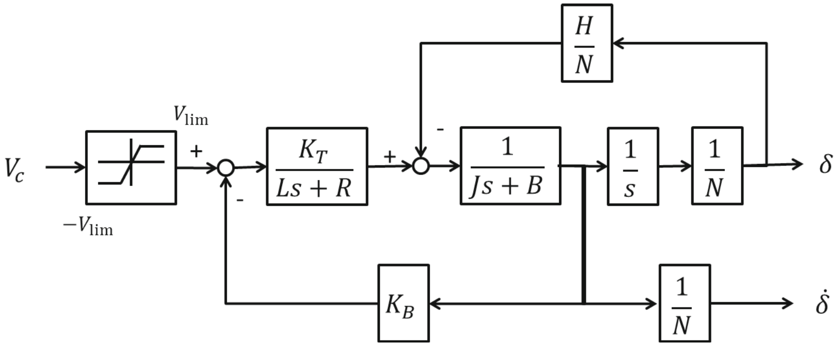


Fig. 3 Actuator plant model used in the simulation

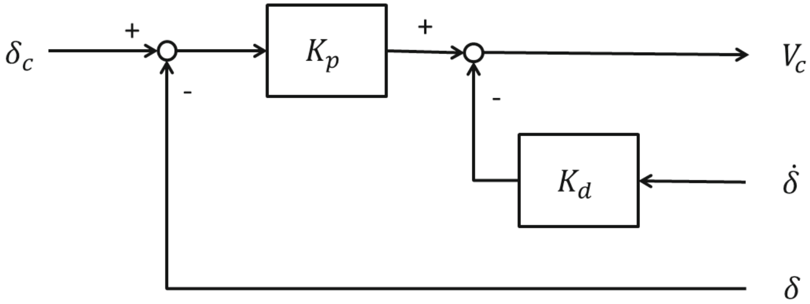


Fig. 4 Controller used for controlling the actuator plant used in the simulation

at 0.3 rad/sec as input virtual commands for all 3 virtual demands. This continuously changing reference input allows for visualisation of the differences between the two control schemes.

4.2 Simulation Results

The unsaturated simulation results are first shown in Figures 5 and 6. The two control schemes perform exactly the same as expected from theoretical analysis presented earlier.

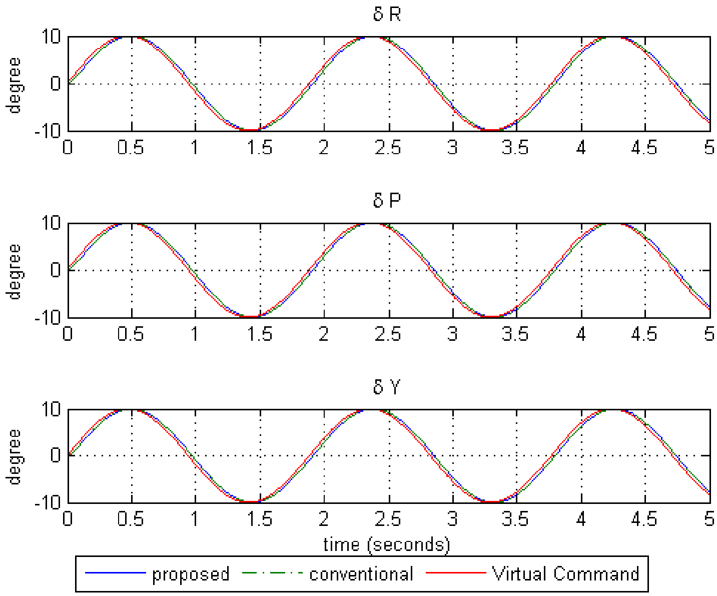


Fig. 5 Simulation results showing the 2 control schemes having exact responses when there is no actuator saturation events. The blue line is the proposed scheme, green dotted line is the conventional scheme, and red line is the virtual command.

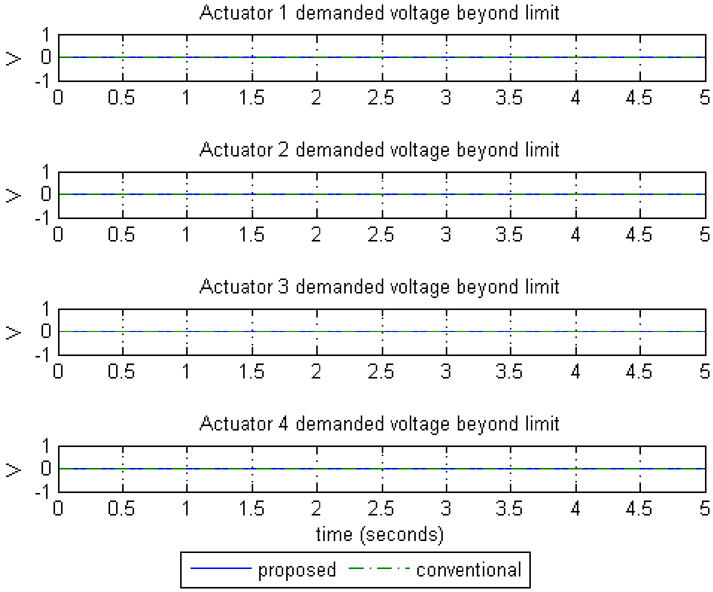


Fig. 6 Simulation plots showing no saturation results showing the 2 control schemes having exact responses when there is no actuator saturation events. The blue line is the proposed scheme, green dotted line is the conventional scheme, and red line is the virtual command.

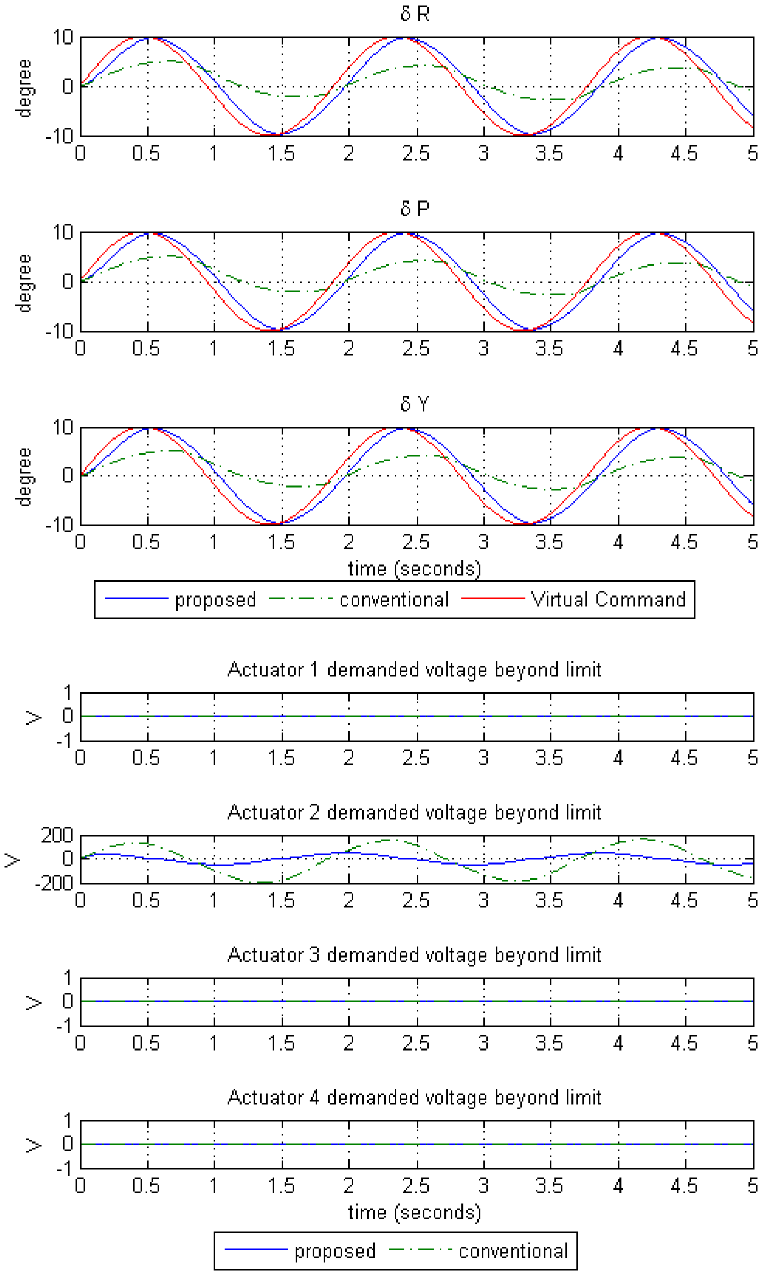


Fig. 7 Simulation results for 1 actuator saturation case. The blue line is the proposed scheme, green dotted line is the conventional scheme, and red line is the virtual command.

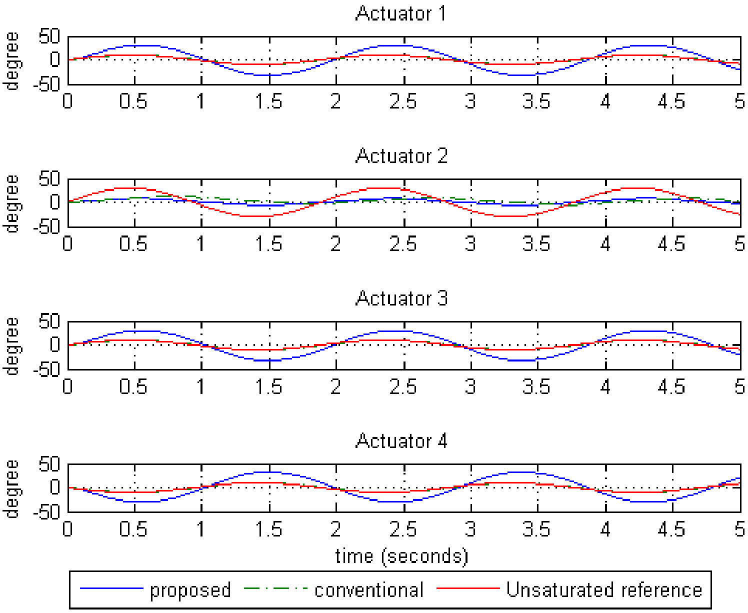


Fig. 8 Individual actuator responses for 1 actuator saturation case. The blue line is the proposed scheme, and green dotted line is the conventional scheme. The red line shows the unsaturated reference signal from previous simulation in comparison.

To simulate the effect of actuator saturation, the maximum voltage input level for Actuator Number 2 is reduced by 90 percent of its original value. This can also be thought of the actuator having a fault to reduce its effectiveness.

Figure 7 shows the simulation results with Actuator Number 2 voltage range reduced. The proposed scheme is able to track the virtual demand relatively well for all three virtual demands. In contrast, the conventional scheme shown with green dotted line exhibits poor tracking consistent across the three virtual demands. Figure 8 shows the individual actuator responses for the same simulation. It can be seen the proposed scheme is able to exploit the available actuator capability when actuator 2 is saturated to improve the performance over the conventional scheme. This also implies the need for the remaining healthy unsaturated actuators to have remaining actuator margins in physical constraints for the proposed scheme to exploit.

5 Conclusion

Analytical analysis of a proposed actuator control scheme over conventional control scheme is presented. By directly regulating tracking error in the moment space, the change in the control variables over conventional actuator control schemes improves the tracking performance during actuator saturation when there is available actuator redundancy available. The proposed scheme is shown to be superior in performance when one actuator is saturated with smaller magnitude in the error dynamics. This is due to its ability to utilize the available redundancy to reduce the performance degradation. Simulations verified the theoretical analysis.

References

1. Bernstein, D., Michel, A.: A chronological bibliography on saturating actuators. International Journal of Robust and Nonlinear Control 5(5), 375–380 (1995)
2. Blakelock, J.: Automatic control of aircraft and missiles. Wiley-Interscience (1991)
3. Dorf, R., Bishop, R.: Modern control systemss. Prentice Hall (2001)
4. Durham, W.: Constrained control allocation. Journal of Guidance, Control, and Dynamics 16(4), 717–725 (1993)
5. Harkegard, O.: Dynamic control allocation using constrained quadratic programming. Journal of Guidance Control and Dynamics 27(6), 1028–1034 (2004)
6. Hu, T., Lin, Z., Chen, B.: An analysis and design method for linear systems subject to actuator saturation and disturbance. Automatica 38(2), 351–359 (2002)
7. Kim, S., Lee, Y., Tahk, M.: New structure for an aerodynamic fin control system for tail fin-controlled stt missiles. Journal of Aerospace Engineering 24(4) (2011)
8. Oppenheimer, M., Doman, D., Bolender, M.: Control allocation for over-actuated systems. In: 14th Mediterranean Conference on Control and Automation, MED 2006, pp. 1–6. IEEE (2006)
9. Yount, L., Todd, J.: Flight control module with integrated spoiler actuator control electronics, uS Patent 6,561,463 (May 13, 2003)
10. Zaccarian, L.: Dynamic allocation for input redundant control systems. Automatica 45(6), 1431–1438 (2009)

Concurrent Learning Adaptive Model Predictive Control

Girish Chowdhary, Maximilian Mühlegg, Jonathan P. How,
and Florian Holzapfel

Abstract. A concurrent learning adaptive-optimal control architecture for aerospace systems with fast dynamics is presented. Exponential convergence properties of concurrent learning adaptive controllers are leveraged to guarantee a verifiable learning rate while guaranteeing stability in presence of significant modeling uncertainty. The architecture switches to online-learned model based Model Predictive Control after an online automatic switch gauges the confidence in parameter estimates. Feedback linearization is used to reduce a nonlinear system to an idealized linear system for which an optimal feasible solution can be found online. It is shown that the states of the adaptively feedback linearized system stay bounded around those of the idealized linear system, and sufficient conditions for asymptotic convergence of the states are presented. Theoretical results and numerical simulations on a wing-rock problem with fast dynamics establish the effectiveness of the architecture.

1 Introduction

Model based optimal control of dynamical systems is a well studied topic. For example, one of the most commonly used techniques for linear and nonlinear systems with constraints is model predictive control (see e.g. [4, 30, 20]). While this technique has been heavily studied and implemented for slower industrial processes, only in the past decade enough computational power has become available to enable online optimization for fast system dynamics typical of aerospace applications (some relevant demonstrations are in [14, 15, 13, 32, 33, 5]). MPC depends on a

Girish Chowdhary · Jonathan P. How
Massachusetts Institute of Technology
e-mail: {girishc,jhow}@mit.edu

Maximilian Mühlegg · Florian Holzapfel
Technische Universität München
e-mail: {maximilian.muehlegg,florian.holzapfel}@tum.de

dynamic predictive model of the system. However, unaccounted modeling errors and dynamic variations in any real world scenario often result in an *a-priori* generated model of a system becoming obsolete or inaccurate. In such cases, the stability and performance of an MPC approach cannot be guaranteed, especially if the underlying dynamics are nonlinear [31]. One way to deal with this is to estimate parameters of the dynamic model online, and then generate optimal controllers at each time step assuming that the estimated model at that time step is the correct one. This results in an indirect adaptive control approach that uses the principle of certainty equivalence (see e.g. [19, 2]). The benefit of this indirect-adaptive MPC approach is that it allows for a way to incorporate learning in the MPC framework. However, the main drawback of this method is that it is difficult to guarantee stability, especially during parameter estimation transient phases. This is one major challenge in synthesizing algorithms for online adaptive-optimal control [25].

Several authors have made key contributions to implementing such adaptive MPC architectures. Fukushima et al. used the comparison principle to develop adaptive MPC for linear systems [17]. Adetola et al. considered adaptive MPC of linearly parameterized nonlinear systems and showed that one way to guarantee stability is to ensure that the initial parameter errors are within certain bounds [1]. Aswani et al. explored and experimented in flight with the notion of safe-MPC by guaranteeing that control inputs are selected such that the system evolution is constrained to (approximations of) invariant reachable sets. Their work has clearly demonstrated that adaptive MPC can indeed result in improved flight performance through flight testing. However, they used an EKF for parameter estimation, which is known to not guarantee predictable and quantifiable learning rates under general operating conditions, and concentrate on linear dynamical systems [5, 3]. In general, while significant progress has been made in adaptive MPC, the results tend to be conservative, as the presence of learning transients prevent a general non-conservative solution to be formed.

On the other hand, adaptive control is one of the most well studied areas in control systems theory. In adaptive control algorithms and techniques are developed for dealing with modeling uncertainties and disturbances. Direct adaptive control methods directly modify the system input to account for modeling uncertainties. In a certain light, these techniques could be viewed as model-free, in the sense that they do not focus on learning the system model, but rather on suppressing the uncertainty pointwise-in-time to minimize the instantaneous tracking error. Direct adaptive controllers can guarantee stability, even during harsh transients, however, they do not offer any long-term improvement due to model learning unless the system states are persistently exciting (PE; see e.g. [6]). Furthermore, it is difficult to generate optimal solutions in presence of input and state constraints with direct adaptive control architectures.

Adaptive control literature also consists of hybrid-direct-indirect control architectures. For example, Duarte and Narendra, Lavretsky, and Chowdhary and Johnson have shown that modifying direct adaptive controllers such that they focus also on learning the uncertainty improves performance (see e.g. [12, 24, 9]). The power of these techniques is that they can handle harsh learning transients,guarantee learning of unknown model parameters subject to conditions on the system trajectories, and guarantee system stability during the learning. It is natural therefore, to hypothesize that adaptive-optimal control algorithms can be devised that use provable hybrid adaptive control techniques to guarantee stability in the learning phase and then switch automatically to model-based optimal control algorithms (e.g. MPC) after sufficient confidence in estimated parameters has been gauged online. One such architecture is proposed in this paper and displayed in Figure 1. The main challenges in developing such an architecture include guaranteeing a verifiable learning rate for the uncertainty estimation such that the uncertainty is approximated in finite time before the architecture switches to the online learned model-based optimal controller, guaranteeing stability before and after the switch, and guaranteeing that the architecture can switch back to the adaptive controller if ideal parameters of the system change.

In this paper, we present a Concurrent Learning based adaptive-optimal Model Predictive Controller (CL-MPC) to address these challenges. Our architecture leverages the CL algorithm of Chowdhary and Johnson [9, 8], which guarantees simultaneous system stability and exponential convergence to the ideal parameters without requiring persistency of excitation. This allows us to guarantee verifiable convergence rates. A online metric is developed to initiate a switch to MPC. Learning continues while the system is in MPC using a variant of the CL algorithm, and it is shown that exponential convergence of parameters can be guaranteed if the basis of the uncertainty is known. Furthermore, using a feedback linearization approach we show that a feedback linearizable nonlinear system can be transformed into a a linear system for which an optimal feasible MPC solution can be formulated in presence of constraints. This greatly helps in ensuring feasibility of obtaining an optimal solution for aerospace systems with fast dynamics, as one need only to solve the MPC problem for the ideal feedback linearized system. It is also shown that the actual feedback linearized system's solution is mean square exponentially bounded around the ideal system, and sufficient conditions are provided to guarantee asymptotic convergence to the ideal solution. The presented architecture is validated through simulation on a wing-rock dynamics system. The results show significant improvement over an adaptive-only approach in presence of significant modeling uncertainty.

2 Approximate Model Inversion Based Model Reference Adaptive Control

Let $x(t) \in D_x \subset \mathbb{R}^n$, $\delta(t) \in D_\delta \subset \mathbb{R}^l$, and consider the following multiple-input nonlinear uncertain dynamical system

$$\dot{x}(t) = f(x(t), \delta(t)). \tag{1}$$

The unknown function $f(\cdot)$ is assumed to be globally Lipschitz and the control input δ is assumed to be bounded and piecewise continuous, so as to ensure the

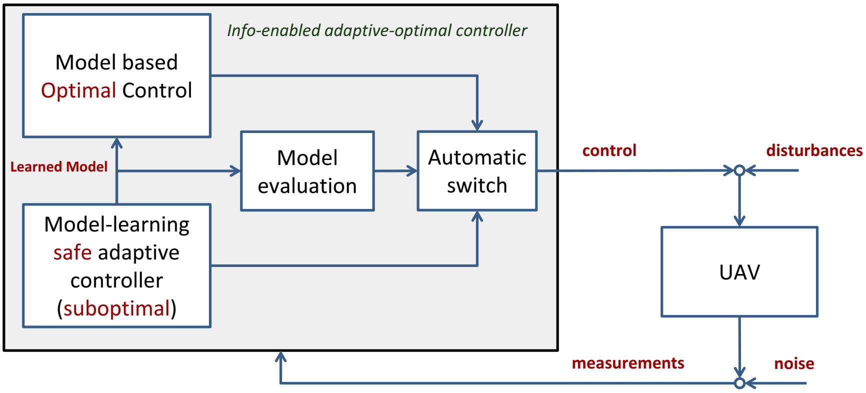

Info-enabled adaptive-optimal controller
Model based Optimal Control
Learned Model
Model evaluation
Automatic switch
Model-learning safe adaptive controller (suboptimal)
control
disturbances
UAV
measurements
noise

Fig. 1 An adaptive-optimal control architecture. A learning-focused adaptive controller guarantees stability while learning uncertain system parameters. Once sufficient confidence has been gauged online in the estimated parameters, the architecture switches to using an online model-based controller, such as MPC. The resulting switched adaptive-optimal controller is guaranteed to be stable without being conservative about initial parameter errors.

existence and uniqueness of the solution to (1). Furthermore, it is assumed that an admissible control input exists that drives the system from any initial condition in D_x to a neighborhood of any arbitrary point in D_x in finite time. It is further assumed that $l \leq n$ (while restrictive for overactuated systems, this assumption can be relaxed through the design of appropriate control assignment [16]).

The Approximate Model Inversion based MRAC approach used here feedback linearizes the system (1) by finding a pseudo-control input $\nu(t) \in \mathbb{R}^l$ that achieves a desired acceleration. If the exact plant model in equation (1) is known and invertible, the required control input to achieve the desired acceleration is computable by inverting the plant dynamics. However, since this usually is not the case, an approximate inversion model $\hat{f}(x,\delta)$ is employed. The inversion model is chosen to be invertible w.r.t. δ; the operator $\hat{f}^{-1} : \mathbb{R}^{n+l} \to \mathbb{R}^l$ is assumed to exist and assign for every unique element of $\mathbb{R}^{n+l}$ a unique element of $\mathbb{R}^l$.

The following assumption guarantees invertibility of $\hat{f}(x,\delta)$ w.r.t. δ [21].

Assumption 1. $\frac{\partial \hat{f}(x,\delta)}{\partial \delta}$ is continuous w.r.t δ and nonsingular over $D_x \times D_\delta$.

Given a desired pseudo-control input $\nu \in \mathbb{R}^l$ a control command δ can be found by approximate dynamic inversion:

$$\delta = \hat{f}^{-1}(x,\nu). \tag{2}$$

Let $z = (x,\delta)$ for brevity. The use of an approximate model results in a modeling error Δ for the system,

$$\Delta(z) = f(z) - \hat{f}(z). \tag{3}$$

It should be noted that if the control assignment function (the mapping between control inputs to states) is known and invertible with respect to δ, then an inversion model exists such that the modeling error is not dependent on the control input δ.

The modeling uncertainty can be assumed to be represented using a linear combination of basis functions. The basis functions can often be designed based on knowledge of the system dynamics (see e.g. [37, 9]). Alternatively, universally approximating bases, such as Gaussian radial basis functions, can be used ([23]). In either case, letting the basis be represented by $\phi(z) \in \mathbb{R}^m$, we assume the existence of an ideal weight matrix $W^* \in \mathbb{R}^{m \times l}$ such that

$$\Delta(z) = W^{*^T}\phi(z) + \eta(z), \tag{4}$$

where the representation error $\eta_{\text{sup}} = \sup_{z \in D_x} \|\bar{\eta}(z)\|$ is bounded over D_x.

A designer chosen reference model is used to characterize the desired response of the system

$$\dot{x}_{rm} = f_{rm}(x_{rm}, r), \tag{5}$$

where $f_{rm}(\cdot)$ denote the reference model dynamics, which are assumed to be continuously differentiable in x_{rm} for all $x_{rm} \in D_x \subset \mathbb{R}^n$. The reference command $r(t)$ is assumed to be bounded and piecewise continuous, furthermore, $f_{rm}(\cdot)$ is assumed to be such that x_{rm} is bounded for a bounded reference input.

Define the tracking error to be $e(t) = x_{rm}(t) - x(t)$, and the pseudo-control input ν to be

$$\nu = \nu_{rm} + \nu_{pd} - \nu_{ad}, \tag{6}$$

consisting of a linear feedback term $\nu_{pd} = Ke$ with $K \in \mathbb{R}^{l \times n}$; a linear feedforward term $\nu_{rm} = \dot{x}_{rm}$; and an adaptive term $\nu_{ad}(z)$. Since Δ is a function of ν_{ad} as per equation (3), and ν_{ad} needs to be designed to cancel Δ, the following assumption needs to be satisfied:

Assumption 2. The existence and uniqueness of a fixed-point solution to $\nu_{ad} = \Delta(\cdot, \nu_{ad})$ is assumed.

Sufficient conditions for satisfying this assumption are available in [40, 21]. Assumption 2 implicitly requires the sign of the control effectiveness matrix to be known ([21]).

Using equation (3) and the pseudo-control (6) the tracking error dynamics can be written as

$$\dot{e} = Ae + B[\nu_{ad}(z) - \Delta(z)], \tag{7}$$

where the state space model (A, B) is in canonical form with the eigenvalues of A assigned by ν_{pd}. The baseline full state feedback controller ν_{pd} is chosen to make A Hurwitz. Hence, for any positive definite matrix $Q \in \mathbb{R}^{n \times n}$, a positive definite solution $P \in \mathbb{R}^{n \times n}$ exists for the Lyapunov equation

$$0 = A^T P + PA + Q. \tag{8}$$

The adaptive controller framework described above guarantees that the tracking error is uniformly bounded if the following well known gradient based update laws that minimize a cost on the instantaneous tracking error $e^T e$ are used:

$$\dot{W}(t) = -\Gamma_W \phi(z(t)) e^T(t) PB. \tag{9}$$

However, this adaptive law guarantees that the adaptive parameters (W) stay bounded within a neighborhood of the ideal parameters (W^*) only if the regressor vector $\phi(z)$ is PE (see e.g. [35, 29, 28, 22]). Note that even when $\phi(z)$ is PE, $e(t) \nrightarrow 0$ uniformly only if $\sup_{z \in D_x} \|\bar{\eta}(z)\| = 0$. Therefore, this adaptive law cannot be used within the proposed framework as there is no guarantee that the weights will converge to their true values and an (approximate) representation of the system uncertainty will be learned. This has been a major reason why MRAC-MPC switching systems cannot be formulated easily.

Fortunately, it is possible to incorporate long term learning in the MRAC framework by ensuring that the adaptive law learns the modeling uncertainty by incorporating additional information [24, 12, 9]. It was shown in [8, 9] that for linearly parameterized uncertainties the requirement on persistency of excitation can be relaxed if online recorded data is used concurrently with instantaneous data for adaptation. In particular, for a linearly parameterized representations of the uncertainty, the following theorem can be proven [8, 9, 10]:

Theorem 1. *Consider the system given by (1), with the inverse law (2), and the reference model of (5). Assume that the uncertainty is linearly parameterizable using an appropriate set of bases over a compact domain D_x. For each online recorded data point i, let $\varepsilon_i(t) = W^T(t)\phi(x_i, \delta_i) - \hat{\Delta}(x_i, \delta_i)$, with $\hat{\Delta}(x_i, \delta_i) = \hat{\dot{x}}_i - \nu(x_i, \delta_i)$, where $\hat{\dot{x}}_i$ is the bounded estimate of $\dot{x}_i$, and consider the following update law*

$$\dot{W} = -\Gamma_W \phi(z) e^T PB - \frac{1}{p} \sum_{j=1}^{p} \Gamma_{W_b} \phi(x_i, \delta_i) \varepsilon_j^T, \tag{10}$$

where $\Gamma_{W_b} > 0$ is the learning rate for training on online recorded data. Let $Z = [\phi(z_1), ..., \phi(z_p)]$ and assume that rank$(Z) = m$. Furthermore, let B_α be the largest compact ball in D_x with radius α, and assume $x(0) \in B_\alpha$, define $\delta = \frac{2\|PB\|\bar{\eta}}{\lambda_{\min}(Q)} + \frac{p\bar{\eta}\sqrt{l}}{\lambda_{\min}(\Omega)}$ with $\Omega = ZZ^T$, and assume that D_x is sufficiently large such that $m_{rm} = \alpha - \delta$ is a positive scalar. If the states x_{rm} of the bounded input bounded output reference model of (5) remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m_{rm}\}$ for all $t \geq 0$ then the tracking error e and the weight error $\tilde{W} = W - W^*$ are mean-squared exponentially uniformly ultimately bounded. Furthermore, if the representation is exact over D_x, that is if $\sup_{z \in D_x} \|\bar{\eta}(z)\| = 0$, then the tracking error and weight error converge exponentially fast to a compact ball around the origin for arbitrary initial conditions, with the rate of convergence directly proportional to the minimum singular value of the history stack matrix Z.

Remark 1. The size of the compact ball around the origin where the weight and tracking error converge is dependent on the representation error $\bar{\eta}$ and the estimation error $\breve{\varepsilon} = \max_i \|\dot{x}_i - \hat{\dot{x}}_i\|$. The former can be reduced by choosing appropriate number of RBFs across the operating domain, and the latter can be reduced by an appropriate implementation of a fixed point smoother. A fixed point smoother uses data before and after a data point is recorded to form very accurate estimates of $\hat{\dot{x}}_i$ using a forward-backward Kalman filter [18, 11]. Note that $\hat{\dot{x}}(t)$ is not needed at the *current* time instant t, which is a much more restrictive requirement. Therefore, an appropriate implementation of a fixed point smoother alleviates the time-delay often observed in estimating $\hat{\dot{x}}(t)$ with forward Kalman filter (or a low pass filter) only.

Remark 2. The history stack matrix $Z = [\phi(z_1), ..., \phi(z_p)]$ is not a buffer of the last p states. It can be updated online by including data points that are of significant interest over the course of operation. Theoretically, convergence is guaranteed as soon as the history stack becomes full ranked. New data points could replace existing data points once the history stack reaches a pre-determined size. It was shown in [10] that the rate of convergence of the tracking and weight error is directly proportional to the minimum singular value of Z. This provides a useful metric to determine which data points are most useful for improving convergence. Consequently, an algorithm for adding points that improve the minimum singular value of Z for the case of linearly parameterizable uncertainty was presented there.

Remark 3. The main limitation of the linearly parameterized RBF NN representation of the uncertainty is that the RBF centers need to be preallocated over an estimated compact domain of operation D_x. Therefore, if the system evolves outside of D_x all benefits of using adaptive control are lost. This can be addressed by evolving the RBF basis to reflect the current domain of operation. A reproducing kernel Hilbert space approach for accomplishing this was presented in [23]. However, when the basis is fixed, in order for the adaptive laws above to hold, the reference model and the exogenous reference commands should be constrained such that the desired trajectory does not leave the domain over which the neural network approximation is valid. Ensuring that the state remains within a given compact set implies an upper bound on the adaptation gain (see for example Remark 2 of Theorem-1 in [39]).

3 Feedback Linearization for MPC

The key enabling factor for the proposed switching CL-MPC architecture presented here is the guaranteed convergence property of CL-MRAC as established in Theorem 1. Once the approximation of the uncertainty is good enough the system shall change to the new MPC structure. Therefore a decision algorithm is implemented which tests for

$$\|x\| \neq 0 \ \text{ and } \ \|\hat{\dot{x}} - \nu - \nu_{ad}\| \leq \varepsilon_{\text{tol}}, \tag{11}$$

where $\varepsilon_{\text{tol}} \geq 0$ represents a tolerated approximation error. Note that due to Theorem 1 it can be shown that this guarantees an upper bound on $\tilde{W}(t_\sigma)$, where t_σ is a switching time. Note further that other automatic-switching algorithms, including those that approximate the switching surface probabilistically, are possible and expected to be investigated in our future work.

Once the weights converged to a neighborhood around their optimal values, as determined by the test in (11), the system switches to the model-based optimal controller. In this mode, the plant does not track a reference model but use the complete available control authority $\nu_{avail} \in \mathbb{R}^l$ after feedback linearization to track commands optimally. For the case that the system switched to the model-based optimal controller and with regard to equation (6) redefine the pseudo-control ν to be

$$\nu = \nu_{fb} + K_B \nu_{avail} - \nu_{ad}, \tag{12}$$

consisting of the linear feedback term $\nu_{fb} = K_M x$ with $K_M \in \mathbb{R}^{n\times n}$; a feedforward part $K_B \nu_{avail}$ with $K_B \in \mathbb{R}^{l\times l}$; and the adaptive part ν_{ad}. Let $B_m = BK_B$, then the feedback linearized system becomes

$$\dot{x}(t) = A_m x + B_m \nu_{avail} + B(\Delta - \nu_{ad}), \tag{13}$$

where the state space model (A_m, B_m) is in canonical form with the eigenvalues of A_m assigned by ν_{fb}. Choose the gains such that if $\Delta - \nu_{ad} = 0$, a unique solution to (13) exists and B_m satisfies assumption 1. Furthermore, the resulting matrices (A_m, B_m) need to be chosen such that a feasible optimal solution to the system (13) is known; one possibility is to choose (A_m, B_m) equal to the reference model, which was used during the exclusively adaptive case. The available control authority ν_{avail} is dynamically constrained by the physical maximum and minimum control allowed $\nu_{min/max}$, minus the adaptive part (ν_{ad}) of the pseudo control which is needed to cancel the uncertainty and the part (ν_{fb}) which is required in order for the feedback linearized system to recover the dynamics in 13. For each element of ν_{fb} we have

$$K_B^{-1}(\nu_{min} + \nu_{ad} - \nu_{fb}) \leq \nu_{avail} \leq K_B^{-1}(\nu_{max} + \nu_{ad} - \nu_{fb}). \tag{14}$$

Using equation (4), the last term in equation (13) is

$$\|\Delta(z) - \nu_{ad}(z)\| \leq \|\tilde{W}\| \|\phi(z)\| + \eta_{\text{sup}}. \tag{15}$$

Let $\beta(z) = \Delta(z) - \nu_{ad}$. The feedback linearized system can be written as

$$\dot{x}(t) = A_m x + B_m \nu_{avail} + B\beta(z) \tag{16}$$

Let t_σ be a time instant when the control architecture switches to using MPC. Due to Theorem 1 it follows that $\tilde{W}(t)$ approaches a neighborhood of zero exponentially fast, furthermore, since the algorithm switches to the optimal controller (MPC) only when $\|\dot{x} - \nu + \nu_{ad}\| \leq \varepsilon_{\text{tol}}$ and $\|x\| \neq 0$, it follows that $\|\tilde{W}(t_\sigma)\|$ is small. Leveraging this fact, MPC design is performed on the ideal feedback linearized system with states $\bar{x}(t)$ given by

$$\dot{\bar{x}}(t) = A_m \bar{x}(t) + B_m \nu_{avail}(t), \tag{17}$$

assuming $\beta(z) = 0$.

3.1 Stability

Let $[t_\sigma, t_{\sigma+1}]$ be a finite interval where the algorithm has switched to using the optimal model based controller (e.g. MPC). It is clear that when the algorithm switches, although $\beta(z)$ is likely to be very small, it will probably not be zero. In this case, the question arises as to whether learning should continue or not. Since any possible initial transients have already passed, there seems no reason to continue to learn. In fact, such an approach can be thought equivalent to an assumption on allowable initial parameter error $\tilde{W}(0)$ for a non-switching based MPC [17, 1, 5]. One approach therefore, could be to continue to learn using a smaller learning-rate Γ and using estimates of model error only (not using also the tracking error as was the case in Theorem 1). The following lemma characterizes that in this case, a concurrent learning gradient descent law guarantees that the feedback linearization error $\beta(z)$ is exponentially bounded. To facilitate the analysis, it is assumed that a noise free estimate of $\dot{x}_i$ for all online recorded data points i is available. This assumption can be relaxed to yield mean squared exponential ultimate boundedness of $\tilde{W}$ instead of mean square exponential stability [27].

Lemma 1. *Consider the model error given by (3), $\varepsilon_i(t)$ as defined in (10) for the recorded data points, and the following gradient descent law*

$$\dot{W} = -\Gamma \sum \phi(x_i, \delta_i)\varepsilon_j^T). \tag{18}$$

Assume also that the history stack is full ranked, that is rank$(Z) = m$, then the parameter error is exponentially bounded as $\|\tilde{W}(t)\| \le \exp^{-c_1 t}\|\tilde{W}(t_\sigma)\|$ for some $c_1 > 0$ dependent on Z and the parameter error $\tilde{W}(t_\sigma)$ at the instant the algorithm switches to model based optimal control. Furthermore,
$\beta(z(t)) \le \exp^{-c_1 t}\|\tilde{W}(t_\sigma)\|\phi(z(t)) + \eta_{\sup}$ for all $t \in [t_\sigma, t_{i+1}]$.

Proof. Consider the quadratic function given by $V(\tilde{W}) = \frac{1}{2}\tilde{W}(t)^T\Gamma^{-1}\tilde{W}(t)$, and note that $V(0) = 0$ and $V(\tilde{W}) > 0 \ \forall \ \tilde{W} \neq 0$, hence $V(\tilde{W})$ is a Lyapunov function candidate. Since $V(\tilde{W})$ is quadratic, letting $\lambda_{\min}(.)$ and $\lambda_{\max}(.)$ denote the operators that return the minimum and maximum eigenvalue of a matrix, we have: $\lambda_{\min}(\Gamma^{-1})\|\tilde{W}\|^2 \le V(\tilde{W}) \le \lambda_{\max}(\Gamma^{-1})\|\tilde{W}\|^2$. Differentiating the Lyapunov candidate with respect to time along the trajectories of (18) we have

$$\dot{V}(\tilde{W}(t)) \le -\tilde{W}(t)^T[\sum_{j=1}^{p}\phi(x_j)\phi^T(x_j)]\tilde{W}(t). \tag{19}$$

Let $\Omega = \sum_{j=1}^{p}\Phi(x_j)\Phi^T(x_j)$ and note that since $\phi(x(t))\phi^T(x(t)) \ge 0 \, \forall\phi(x(t))$, $\lambda_{min}(\Omega)$ > 0. Then it follows that

$$\dot{V}(\tilde{W}) \le -\lambda_{\min}(\Omega)\|\tilde{W}\|^2 \le -\frac{\lambda_{\min}(\Omega)}{\lambda_{\max}(\Gamma^{-1})}V(\tilde{W}). \tag{20}$$

Let $c_1 = \frac{\lambda_{\min}(\Omega)}{\lambda_{\max}(\Gamma^{-1})}$, then $\|\tilde{W}(t)\| \leq \exp^{-c_1 t}\|\tilde{W}(t_\sigma)\|$. It follows from the definition of $\beta(z)$ in (16) that for all $t \in [t_\sigma, t_{i+1}]$

$$\beta(z(t)) \leq \exp^{-c_1 t}\|\tilde{W}(t_\sigma)\|\phi(z(t)) + \eta_{\sup}. \tag{21}$$

□

The next theorem shows that $\tilde{x} = x - \bar{x}$, the difference between the ideal feedback linearized system (16) and the true feedback linearized system (17) is bounded. Therefore, applying the stabilizing feasible solution of the ideal system (17) to the true system (16) guarantees boundedness of the true system states.

Theorem 2. *Consider the true feedback linearized system in (16) and Lemma 1. Assume that a feasible optimal solution ν^*_{avail} exists for the ideal feedback linearized system of (17). Then, the states of the true system with the control ν^*_{avail} are uniformly ultimately bounded around those of the ideal system, and approach asymptotically a compact set that is a function of the representation error $\eta_{\sup}$ over every switching interval $[t_\sigma, t_{\sigma+1}]$ where MPC control is active.*

Proof. Note that

$$\dot{\tilde{x}} = A_m\tilde{x} + B\beta(z). \tag{22}$$

Let $V(\tilde{x}) = \frac{1}{2}\tilde{x}^T P_m \tilde{x}$, where P_m is the positive definite solution to $0 = Q_m + A_m^T P_m + P_m A_m$ for a positive definite Q_m, guaranteed to exist due to the feedback ν_{fb}, which is chosen such that A_m in (13) is Hurwitz. Hence $V(\tilde{W})$ is a radially unbounded quadratic Lyapunov function candidate with: $\lambda_{\min}(\Gamma^{-1})\|\tilde{W}\|^2 \leq V(\tilde{W}) \leq \lambda_{\max}(\Gamma^{-1})\|\tilde{W}\|^2$. It follows therefore that

$$\dot{V}(\tilde{x}) \leq -\tilde{x}^T Q_m \tilde{x} + \tilde{x}^T P B\beta(z). \tag{23}$$

Applying Lemma 1 we have

$$\dot{V}(\tilde{x}) \leq -\lambda_{\min}(Q_m)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P_m B\|(\exp^{-c_1 t}\|\tilde{W}(t_\sigma)\|\phi(z(t))\| + \eta_{\sup}). \tag{24}$$

Let $c_2 = \|P_m B\|\|\tilde{W}(t_\sigma)\|$, and noting that the m basis functions are bounded by $\|\phi(.)\| \leq c_3$, we have

$$\dot{V}(\tilde{x}) \leq -\lambda_{\min}(Q_m)\|\tilde{x}\|^2 + \|\tilde{x}\|(mc_2c_3\exp^{-c_1 t} + \eta_{\sup}). \tag{25}$$

Therefore, outside of the compact set $\|\tilde{x}\| \geq \frac{mc_2c_3\exp^{-c_1 t} + \eta_{\sup}}{\lambda_{\min}Q_m}$, $\dot{V}(\tilde{x}) \leq 0$. Therefore $\tilde{x}$ is uniformly ultimately bounded and approaches asymptotically the set $\|\tilde{x}\| \geq \frac{\eta_{\sup}}{\lambda_{\min}Q_m}$.

□

Corollary 1. *Assume that Theorem 2 holds and that an exact representation exists such that $\eta_{\sup} = 0$ in (4), then, the states of the true feedback linearized system asymptotically approach the states of the ideal feedback linearized system over every switching interval $[t_\sigma, t_{\sigma+1}]$ where MPC control is active.*

Proof. The proof follows by noting that (25) becomes

$$\dot{V}(\tilde{x}) \leq -\lambda_{\min}(Q_m)\|\tilde{x}\|^2 + \|\tilde{x}\|(mc_2c_3 \exp^{-c_1 t}), \tag{26}$$

hence, $V \to 0$ as $t \to \infty$.

□

4 Model Predictive Control

For the implementation of the MPC a discrete model of the feedback linearized system in equation (17) is formulated:

$$\bar{x}(k+1) = \bar{A}_m \bar{x}(k) + \bar{B}_m v_{avail}(k), \tag{27}$$

where $\bar{A}_m$ and $\bar{B}_m$ denote the discretized versions of the respective matrices in equation (17). Let $\Delta v_{avail}(k+1) = v_{avail}(k+1) - v_{avail}(k)$ be a future incremental control. The optimal control trajectory is captured by a sequence of incremental control signals:

$$\Delta U = \begin{bmatrix} \Delta v_{avail}(k) \\ \Delta v_{avail}(k+1) \\ \vdots \\ \Delta v_{avail}(k+N_c-1) \end{bmatrix}, \tag{28}$$

where N_c denotes the control horizon. Within the prediction horizon $N_p \geq N_c$ the MPC drives the state of the system $\bar{x}(k)$ onto the desired reference signal $r(k)$ by minimization of a quadratic cost function. Define the following matrices:

$$F = \begin{bmatrix} \bar{A}_m \\ \bar{A}_m^2 \\ \vdots \\ \bar{A}_m^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} \bar{B}_m & 0 & \ldots\ldots & 0 \\ \bar{A}_m\bar{B}_m & \bar{B}_m & 0 \ldots & 0 \\ \vdots & & & \\ \bar{A}_m^{N_p-1}\bar{B}_m & \bar{A}_m^{N_p-2}\bar{B}_m & \ldots\ldots & \bar{A}_m^{N_p-N_c}\bar{B}_m \end{bmatrix}, \tag{29}$$

where $F \in \mathbb{R}^{n \cdot N_p \times n}$ and $\Phi \in \mathbb{R}^{N_p \cdot n \times N_c \cdot n_s}$. Let $\Delta\bar{x}(k+1) = \bar{x}(k+1) - \bar{x}(k)$. Then the vector containing the predicted states $X \in \mathbb{R}^{n \cdot N_p}$ within the prediction horizon can be built by

$$X = F\Delta\bar{x}(k) + \Phi\Delta U. \tag{30}$$

In the MPC framework constraints can be formulated for the input and the states. The goal is to formulate the constraints dependent on the incremental control ΔU. For the control input we have

$$v_{avail,min} \leq M_1 v(k-1) + M_2 \Delta U \leq v_{avail,max}, \tag{31}$$

where $M_1 = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \in \mathbb{R}^{Nc \cdot ns \times ns}$ and $M_2 = \begin{bmatrix} I\, 0 \ldots \ldots 0 \\ I\; I \;\; 0 \;\ldots 0 \\ \vdots \\ I \quad \ldots \quad I \end{bmatrix} \in \mathbb{R}^{ns \cdot N_c \times ns \cdot N_c}$. The input constraints are placed by the vectors $v_{avail,min}$ and $v_{avail,max}$, each consisting of N_c elements of the minimum and maximum control input. Equation (31) can also be expressed in matrix form:

$$\begin{bmatrix} -M_2 \\ M_2 \end{bmatrix} \Delta U \leq \begin{bmatrix} -v_{avail,min} + M_1 v_{avail}(k-1) \\ v_{avail,max} - M_1 v_{avail}(k-1) \end{bmatrix}. \tag{32}$$

Similarly, constraints on the states can be placed by

$$X_{min} \leq F\Delta\bar{x}(k_i) + \Phi\Delta U \leq X_{max}. \tag{33}$$

Similar to equation (31), X_{min} and X_{min} are vectors containing the lower and upper constraints for the states. Written in matrix form we have

$$\begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -X_{min} + F\Delta\bar{x}(k) \\ X_{max} - F\Delta\bar{x}(k) \end{bmatrix}. \tag{34}$$

There exists a functional relationship between the predicted system state and the incremental control input Δv_{avail}. Using hard constraints on input and the states simultaneously can cause constraint conflicts. Introducing a slack variable $s > 0$ and relaxing the constraints $X_{min/max}$ solves this problem. Let $R_S \in \mathbb{R}^{n \cdot N_p}$ be a vector containing the reference command $r(k)$ with $R_s = [1, ..., 1]r(k)$ and define the following quadratic cost function, which reflects the control objective:

$$J = (R_s - X)^T \bar{Q}(R_s - X) + \Delta U^T \bar{R} \Delta U. \tag{35}$$

Here $\bar{R}$ and $\bar{Q}$ denote positive definite diagonal matrices. Inserting equation (30) into equation (35) the problem of model predictive control is finding the control sequence ΔU which minimizes the cost function

$$\begin{aligned} J = {} & (R_s - F\bar{x}(k))^T \bar{Q}(R_s - F\bar{x}(k)) - 2\Delta U^T \Phi^T \bar{Q}(R_s - F\Delta\bar{x}(k)) \\ & + \Delta U^T (\Phi^T \bar{Q}\Phi + \bar{R})\Delta U, \end{aligned} \tag{36}$$

subject to the inequality constraint

$$\begin{bmatrix} -M_2 \\ M_2 \\ -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -v_{avail,min} + M_1 v_{avail}(k-1) \\ v_{avail,max} - M_1 v_{avail}(k-1) \\ -X_{min} + F\Delta\bar{x}(k) \\ X_{max} - F\Delta\bar{x}(k) \end{bmatrix}. \tag{37}$$

Note that since A_m is known a-priori, it may be possible to solve a significant portion of this problem off-line to create the optimal value-function which can be directly used on-line for an approximate optimal solution.

5 Trajectory Tracking in the Presence of Wing Rock Dynamics

Modern highly swept-back or delta wing fighter aircraft are susceptible to lightly damped oscillations in roll angle known as “Wing Rock”. Wing rock often occurs at low speeds and at high angle of attack, conditions commonly encountered when the aircraft is landing (see [34] for a detailed discussion of the wing rock phenomena). Hence, precision control of the aircraft in the presence of wing rock dynamics is critical in order to ensure safe landing. In this section we use concurrent learning adaptive control and the proposed MPC framework to track a sequence of roll commands in the presence of wing rock dynamics. Let ϕ denote the roll attitude of an aircraft, p denote the roll rate and δ_a denote the aileron control input . Then a model for wing rock dynamics is ([26]):

$$\dot{\phi} = p \tag{38}$$

$$\dot{p} = \delta_a + \Delta(x), \tag{39}$$

where $\Delta(x) = W_0 + W_1\phi + W_2 p + W_3|\phi|p + W_4|p|p + W_5\phi^3$. The parameters for wing rock motion are partly adapted from [36, 38, 7]; they are $W_1 = 6.2314, W_2 = 2.1918, W_3 = -0.6245, W_4 = 0.0095, W_5 = 0.0214$. In addition to these parameters, a trim error is introduced by setting $W_0 = 0.8$. A simple inversion model has the form $\nu = \delta_a$. The linear part of the control law is given by $\nu_{pd} = -4\phi - 2p$ for the exclusive adaptive as well as the MPC part of the control framework of Figure 1. These values are chosen as they result in good baseline control performance without exciting high-gain oscillations. Furthermore, in the MPC part the feedforward gain is chosen to be $K_B = 4$. The reference model is chosen to be of second order with natural frequency of $2\,rad/sec$ and a damping ratio of 0.5. This choice results in reasonably smooth trajectories without large transients and without exceeding the constraints when baseline or CL-MRAC controllers are used. The learning rate is set to $\Gamma_W = 6$ for both the instantaneous update and the update based on stored data.

For the concurrent learning adaptive controller only points which increase the rank of the history stack are considered for storage. As long as the history stack does not contain at least as many linearly independent data points as the dimension of the regressor vector, a σ-modification term with gain $\kappa = 0.01$ is added to the update law. Once the history stack is full, an algorithm is employed which increases its minimum singular value ([10]).

The simulation runs for a total of $60\,s$ with a time step of $0.01\,s$. The reference signal $r(t)$ is comprised out of several step inputs. The first two steps start at $5\,s$ and $15\,s$, each having an amplitude of 30° and lasting $5\,s$. The next two step inputs occur after $25\,s$ and $35\,s$, each having an amplitude of 45° and also lasting $5\,s$. After $50\,s$ more aggressive commands shall be tracked. Therefore consecutive steps with an amplitude of -45° or 45° are commanded, alternating every $3\,s$.

Figure 2 shows the performance of the proposed control architecture. During the first step the plant states still deviate from the reference model significantly. However, the tracking performance increases quickly, the plant tracks the reference model at the second step nearly perfectly. After about $30\,s$ the switching condition is met and the system automatically switches to the MPC part of the control framework. It can be observed that the performance increases drastically. This is attributed to the fact that the CL-MPC architecture leverages available control authority fully while simultaneously ensuring that the constraint on the roll rate is not violated.

Figure 3 shows the evolution of the adaptive weights. As soon as the history stack meets the rank condition after about $6s$ the weights start to converge to their optimal values, thus increasing the tracking performance significantly. At the switch to the MPC framework the weights have already nearly converged to their optimal weights. Still, the resulting parameter error is further reduced by the CL update law of Lemma 1 which learns only on stored data.

Figure 4 shows the trajectory of the system in the phase plane during the simulation. It can be seen that, once the MPC is switched on, the region the states reside in increases drastically. This is attributed to the fact that the full available control authority is used, thus increasing the roll rate in transient phases. In addition, despite the increased performance, the chosen state constraints on the roll rate are not violated.

Finally, Figure 5 shows the control input. As long as only the adaptive controller is used, the available control authority is not completely leveraged. Once the MPC is switched on, the complete available control authority is used. Additionally, the constraints placed on the input are not violated.

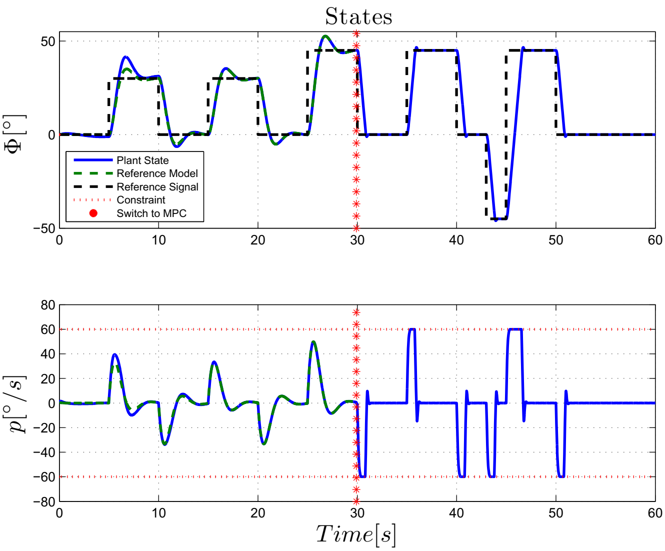


Fig. 2 Performance of the proposed control architecture. At the beginning of the simulation a distinct deviation tracking error is observed. Due to the concurrent learning adaptive controller the performance increases drastically over time. After the switch to the MPC framework, instead of tracking the suboptimal reference model, the plant tracks the command optimally. The constraints on the roll rate are not violated.

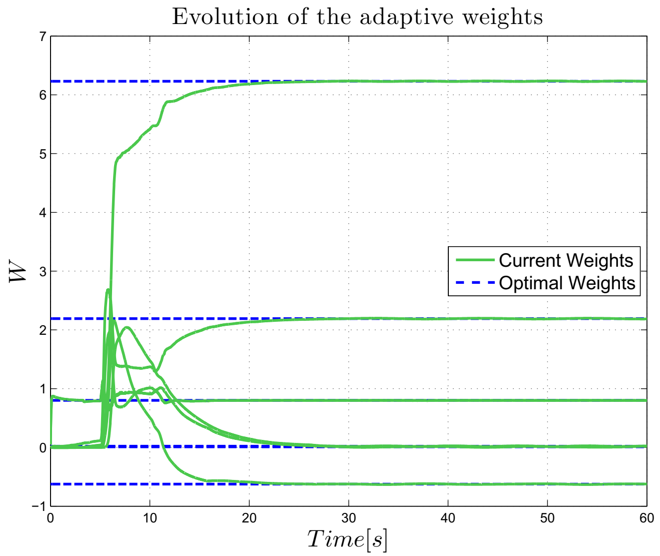


Fig. 3 Evolution of the adaptive weights. Once the history stack meets a condition of linearly independence on the stored data the weights start to converge to their optimal values. Even after the switch to the MPC framework learning based on stored data continues using the algorithm in Lemma 1 and the parameter error is further reduced.

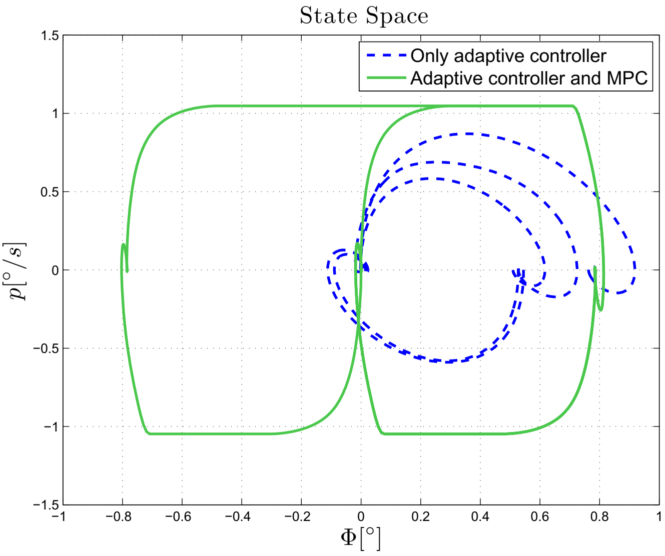


Fig. 4 Phase portrait of the system trajectory. Once the controller switches to the MPC framework the region in which the states evolve is drastically increased as the controller can execute optimal commands w.r.t. the constraints. In addition, MPC ensures that aggressive commands can be tracked without violating the constraints on the roll rate.

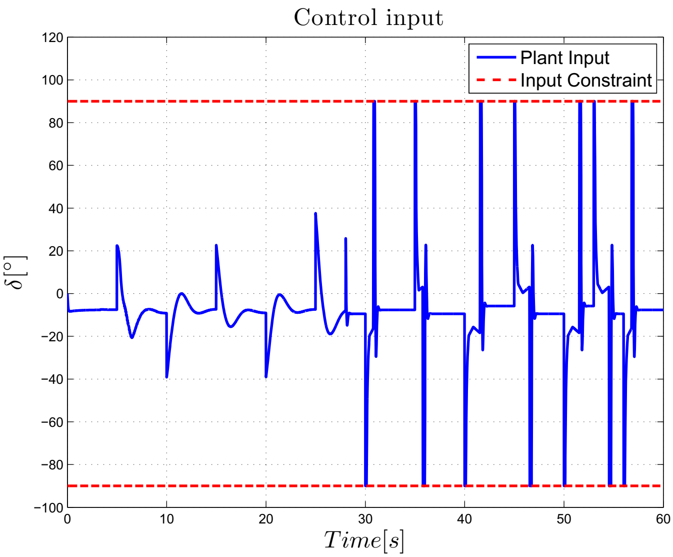


Fig. 5 Control input with constraints. In the beginning of the simulation the controller is concerned with letting the uncertain system behave like the reference model. For this, only a part of the available control authority is used as a conservative reference model is used to ensure constraints are not vioalated. Once the controller switches to the MPC framework the complete control authority is leveraged without violating input constraints.

6 Conclusion

Initial transients often observed during online learning can result in undesirable performance of (receding horizon) online optimal control architectures such as Model Predictive Control. This could make it difficult to implement adaptive MPC on aerospace systems that have fast dynamics. We proposed an adaptive-optimal control architecture in which a concurrent learning adaptive controller is used to guarantee system stability while parameters are adaptively learned. The online-learned model is used to feedback linearize the system and transform its behavior to an ideal feedback-linearized system for which a feasible optimal MPC can be formulated. The MPC takes over after an online metric has gauged sufficient confidence in the learned parameters. It was shown that the states of the feedback linearized system stay exponentially mean square bounded around those of the ideal system, and sufficient conditions were provided to guarantee asymptotic convergence. Simulation results were presented on a wing-rock dynamics system with fast dynamics. These results establish the feasibility of the CL-MPC architecture. Furthermore, these results indicate that learning in adaptive controllers can be used to improve the performance of the system.

Acknowledgements. This research was supported in part by ONR MURI Grant N000141110688.

References

1. Adetola, V., DeHaan, D., Guay, M.: Adaptive model predictive control for constrained nonlinear systems. Systems and Control Letters 58(5), 320–326 (2009)
2. Aström, K.J., Wittenmark, B.: Adaptive Control, 2nd edn. Addison-Weseley, Readings (1995)
3. Aswani, A., Sastry, S.S., Humberto, G., Tomlin, C.: Provably safe and robust learning-based model predictive control. arxiv arXiv:1107.2487v2, UC Berkley (2011)
4. Bemporad, A., Morari, M.: Robust model predictive control: A survey. In: Garulli, A., Tesi, A. (eds.) Robustness in Identification and Control. LNCIS, vol. 245, pp. 207–226. Springer, Heidelberg (1999)
5. Bouffard, P., Aswani, A., Tomlin, C.: Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 279–284 (May 2012)
6. Boyd, S., Sastry, S.: Necessary and sufficient conditions for parameter convergence in adaptive control. Automatica 22(6), 629–639 (1986)
7. Calise, A., Yucelen, T.: Enforcing a linear constraint in adaptive control: A kalman filter optimization approach. In: Proceedings of the AIAA Guidance Navigation and Control Conference, held at Chicago, IL (2009)
8. Chowdhary, G.: Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation. PhD thesis, Georgia Institute of Technology, Atlanta, GA (2010)
9. Chowdhary, G., Johnson, E.N.: Concurrent learning for convergence in adaptive control without persistency of excitation. In: 49th IEEE Conference on Decision and Control, pp. 3674–3679 (2010)
10. Chowdhary, G., Johnson, E.N.: A singular value maximizing data recording algorithm for concurrent learning. In: American Control Conference, San Francisco, CA (June 2011)
11. Chowdhary, G., Johnson, E.N.: Theory and flight test validation of a concurrent learning adaptive controller. Journal of Guidance Control and Dynamics 34(2), 592–607 (2011)
12. Duarte, M.A., Narendra, K.S.: Combined direct and indirect approach to adaptive control. IEEE Transactions on Automatic Control 34(10), 1071–1075 (1989)
13. Dunbar, W.: Distributed receding horizon control of dynamically coupled nonlinear systems. IEEE Transactions on Robotics 52, 1249–1263 (2007)
14. Dunbar, W.B., Murray, R.: Model predictive control of coordinated multi-vehicle formations. In: IEEE Conference on Decision and Control (CDC), pp. 4631–4636 (2002)
15. Dunbar, W.B., Murray, R.M.: Receding horizon control of multi-vehicle formation: A distributed implementation. In: IEEE Conference on Decision and Control, CDC (2004)
16. Durham, W.: Constrained control allocation. AIAA Journal of Guidance, Control, and Dynamics 16, 717–772 (1993)
17. Fukushima, H., Kim, T.-H., Sugie, T.: Adaptive model predictive control for a class of constrained linear systems based on the comparison model. Automatica 43(2), 301–308 (2007)
18. Gelb, A.: Applied Optimal Estimation. MIT Press, Cambridge (1974)

19. Ioannou, P., Sun, J.: Theory and design of robust direct and indirect adaptive-control schemes. International Journal of Control 47(3), 775–813 (1988)
20. Kerrigan, E.C.: Robust Constraint Satisfaction Invariant Sets and Predictive Control. PhD thesis, University of Cambridge, Department of Engineering (November 2000)
21. Kim, N.: Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control. PhD thesis, Georgia Institute of Technology, Atlanta Ga (2003)
22. Kim, Y.H., Lewis, F.L.: High-Level Feedback Control with Neural Networks. Robotics and Intelligent Systems, vol. 21. World Scientific, Singapore (1998)
23. Kingravi, H.A., Chowdhary, G., Vela, P.A., Johnson, E.N.: Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control. IEEE Transactions on Neural Networks and Learning Systems 23(7), 1130–1141 (2012)
24. Lavretsky, E.: Combined/composite model reference adaptive control. IEEE Transactions on Automatic Control 54(11), 2692–2697 (2009)
25. Mayne, D.Q.: Nonlinear Model Predictive Control: Challenges and Opportunities. In: Nonlinear Model Predictive Control. Progress in Systems and Control Theory, vol. 26, pp. 23–44. Birkhäuser, Basel-Boston (2000)
26. Monahemi, M.M., Krstic, M.: Control of wingrock motion using adaptive feedback linearization. Journal of Guidance Control and Dynamics 19(4), 905–912 (1996)
27. Mühlegg, M., Chowdhary, G., Johnson, E.: Concurrent learning adaptive control of linear systems with noisy measurements. In: AIAA Guidance, Navigation, and Control Conference (Augest 2012)
28. Nardi, F.: Neural Network based Adaptive Algorithms for Nonlinear Control. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332 (November 2000)
29. Narendra, K.S.: Neural networks for control theory and practice. Proceedings of the IEEE 84(10), 1385–1406 (1996)
30. Nicolao, G.D., Magi, L., Scattolini, R.: Stability and Robustness of Nonlinear Receding Horizon Control. In: Nonlinear Model Predictive Control. Progress in Systems and Control Theory, vol. 26, pp. 3–22. Birkhäuser, Basel-Boston (2000)
31. Rawlings, J.B.: Tutorial overview of model predictive control. IEEE Control Systems Magazine 20(3), 38–52 (2000)
32. Richards, A., How, J.P.: Decentralized model predictive control of cooperating UAVs. In: IEEE Conference on Decision and Control (CDC), Paradise Island, Bahamas, pp. 4286–4291 (December 2004)
33. Richards, A.G.: Robust Constrained Model Predictive Control. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA (February 2005)
34. Saad, A.A.: Simulation and analysis of wing rock physics for a generic fighter model with three degrees of freedom. PhD thesis, Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, Dayton, Ohio (2000)
35. Sanner, R.M., Slotine, J.-J.E.: Gaussian networks for direct adaptive control. IEEE Transactions on Neural Networks 3(6), 837–863 (1992)
36. Singh, S.N., Yim, W., Wells, W.R.: Direct adaptive control of wing rock motion of slender delta wings. Journal of Guidance Control and Dynamics 18(1), 25–30 (1995)
37. Tao, G.: Adaptive Control Design and Analysis. Wiley, New York (2003)

38. Volyanskyy, K., Calise, A.: An error minimization method in adaptive control. In: Proceedings of AIAA Guidance Navigation and Control Conference (2006)
39. Yucelen, T., Calise, A.: Kalman filter modification in adaptive control. Journal of Guidance, Control, and Dynamics 33(2), 426–439 (2010)
40. Zhang, T., Ge, S.S., Hang, C.C.: Direct adaptive control of non-affine nonlinear systems using multilayer neural networks. In: Proceedings of the 1998 American Control Conference, vol. 1, pp. 515–519 (June 1998)

Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays – Part I

James P. Nelson, Mark J. Balas, and Richard S. Erwin

Abstract. In this paper, we develop a Direct Model Reference Adaptive Tracking Controller for mildly non-linear systems with unknown time varying input delays. This controller can also reject bounded disturbances of known waveform but unknown amplitude, e.g. steps or sinusoids. In this paper a robustness result is developed for DMRAC of mildly non-linear systems with unknown small constant or time varying input delays using the concept of undelayed ideal trajectories. We will show that the adaptively controlled system is globally stable, but the adaptive tracking error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood can be achieved as a result of the control design. A simple example will be provided to illustrate this adaptive control method. The proof of the corollary for the application and further examples are provided in the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part II.

1 Introduction

Time delay affects many engineering, physics and biological systems [1]-[5]. These manuscripts present a firm motivation for the study of time delay systems and a brief overview of the different control approaches commonly used when delays are present. In this overview the open problem of control via the delay and constructive use of the delayed inputs is presented [5]. Further,

James Nelson · Mark J. Balas
University of Wyoming Department of Electrical and Computer Engineering,
1000 E. Universtiy Ave. Laramie, WY 82073
e-mail: {nelsonpj,mbalas}@uwyo.edu

Richard S. Erwin
Air Force Research Laboratory,
Kirtland AFB, NM, 87117, USA

many control systems suffer from unknown delays [6]-[7]. Often these are introduced via systems controlled through a network, e.g in [7].

In previous work [8]-[11] direct model reference adaptive control (DMRAC) and disturbance rejection with very low order adaptive gain laws for MIMO systems was accomplished. Feuntes and Balas developed an ultimate bounded-ness theorem for DMRAC in [11]. When systems are subjected to an unknown internal delay, the adaptive control theory can be modified to handle this situation [12]. However, delays appearing in the inputs or outputs of systems seem to cause more system sensitivity to the delay. A robustness result for the Direct Adaptive Control (DAC) or input delay systems was developed in [13]. A robustness result for the DMRAC of linear systems with "small" input/output delays was developed in [14] using the concept of un-delayed ideal trajectories for the development of the adaptive error system. Using the concept of un-delayed ideal trajectories and this "small-ness" assumption the results of [13] can achieved for the DMRAC of mildly non-linear systems. We will show that the adaptively controlled system is globally stable, but the adaptive error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood can be achieved as a result of the control design. A simple example will be provided to illustrate this adaptive control method. The proof of the corollary for the application and further examples are provided in the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part II.

2 Development of the Adaptive Error System Using "Undelayed Ideal Trajectories"

Our Mildly Non-Linear Plant with Unknown Delay will be modelled by the following mildly non-linear system with an input delay term and an external persistent disturbance:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t-\tau(t)) + \Gamma u_D(t) + f(x) \\ y(t) = Cx(t)\,;\, x(0) = x_0 \end{cases} \tag{1}$$

where the plant state, x(t), is an N-dimensional vector with M-dimensional control input vector, u(t), and M-dimensional sensor output vector, y(t), i.e. the plant is square. The delay $\tau(t) > 0$ is time varying and unknown. The disturbance input vector $u_D(t)$ is N_D-dimensional and will be thought to come from the following Disturbance Generator:

$$\begin{cases} u_D = \Theta\, z_D \\ \dot{z}_D = F\, z_D\,;\, z_D(0) = z_0 \end{cases} \tag{2}$$

The objective of control in this paper will be to cause the output $y(t)$ of the plant to asymptotically track the output $y_m\,(t)$ of an un-delayed Reference Model:

$$\begin{cases} \dot{x}_m = A_m x_m + B_m u_m + f_m(x_m) \\ \quad y_m = C_m x_m;\; x_m(0) = x_0^m \end{cases} \tag{3}$$

where the reference model state $x_m\,(t)$ is an N_m-dimensional vector with reference model output $y_m(t)$ having the same dimension as the plant output $y(t)$. In general, the plant and reference models need not have the same dimensions. The excitation of the reference model is accomplished via the vector $u_m\,(t)$ which is generated by:

$$\dot{u}_m = F_m u_m;\; u_m(0) = u_0^m \tag{4}$$

The reference model parameters (A_m, B_m, C_m, F_m) will be completely known. We define the output error vector:

$$e_y \equiv y - y_m \underset{t\to\infty}{\longrightarrow} 0 \tag{5}$$

and this control objective will be accomplished by an Adaptive Control Law of the form:

$$u = G_m x_m + G_u u_m + G_e e_y + G_D \varphi_D \tag{6}$$

2.1 Ideal Trajectories

We define the "Un-Delayed Ideal Trajectories" for (1) in the following way:

$$\begin{cases} x_* = S_{11}^* x_m + S_{12}^* u_m + S_{13}^* z_D \\ u_* = S_{21}^* x_m + S_{22}^* u_m + S_{23}^* z_D \end{cases} \tag{7}$$

where the ideal trajectory $x_*\,(t)$ is generated by the ideal control $u_*\,(t)$ from

$$\begin{cases} \dot{x}_* = A x_* + B u_* + \Gamma u_D + f(x_*) \\ \qquad\qquad y_* = C x_* = y_m \end{cases}. \tag{8}$$

If such ideal trajectories exist, they will be linear combinations of the reference model state and input (3) and they will produce exact output tracking in a delay-free plant (8).

By substitution of (7) into (8) using (3)-(4), we obtain the Model Matching Conditions:

$$A S_{11}^* + B S_{21}^* = S_{11}^* A_m \tag{9}$$

$$A S_{12}^* + B S_{22}^* = S_{12}^* F_m + S_{11}^* B_m \tag{10}$$

$$C S_{11}^* = C_m \tag{11}$$

$$CS_{12}^{*}=0 \tag{12}$$

$$AS_{13}^{*}+BS_{23}^{*}+\varGamma\varTheta=S_{13}^{*}F \tag{13}$$

$$BS_{23}^{*}=0 \tag{14}$$

$$CS_{13}^{*}=0 \tag{15}$$

$$f(S_{11}^{*}x_m+S_{12}^{*}u_m+S_{13}^{*}z_D)=S_{11}^{*}f_m x_m \tag{16}$$

These conditions (9)-(16) are necessary and sufficient conditions for the existence of the ideal trajectories in the form of (7).

2.2 Fixed Gain Controller

In this section only we will assume that all parameters $(A,B,C,\varGamma,\varTheta,F)$ are known, as well as the solutions to the Model Matching Conditions (9)-(16). This section will help to explain the development of the adaptive scheme; it is not meant to be used in place of such a scheme. We define the state tracking error:

$$e_*\equiv x-x_* \tag{17}$$

and, from (5) and (8) , we obtain

$$e_y\equiv y-y_m=y-y_*=Cx-Cx_*=Ce_*. \tag{18}$$

Furthermore, from (1) and (8) , we have

$$\begin{cases} e_*\equiv x-x_* \\ \varDelta u\equiv u-u_* \\ e_y=y-y_* \\ \varDelta f\equiv f(x)-f(x_*) \end{cases} \tag{19}$$
$$\Rightarrow\begin{cases} \dot{e}_*=Ae_*+B(u(t-\tau(t))-u_*)+\varDelta f \\ \varDelta u\equiv u(t-\tau(t))-u_*=u(t-\tau(t))-u+u-u_* \\ e_y=Ce_* \end{cases}$$

We define a Fixed Gain Controller:

$$u=(S_{21}^{*}x_m+S_{22}^{*}u_m+S_{23}^{*}L\varphi_D)+G_e^{*}e_y=u_*+G_e^{*}e_y. \tag{20}$$

From (19) and (20), we have

$$\begin{cases} \dot{e}_* = A_C e_* + B(u(t-\tau(t)) - u) + \Delta f \\ A_C \equiv A + BG_e^* C \\ \Delta f \equiv f(x) - f(x_*) \end{cases} . \tag{21}$$

The above can be summarized as:

If (A, B, C) is output feedback stabilize-able with a gain G_e^* and the delay equation (21) is stable, then the fixed gain controller (20) will produce local output tracking, i.e.:

$$\lim_{t\to\infty} e_y < R_* \tag{22}$$

Note that output feedback stabilization can be accomplished when

$$M + P + N_D > N \tag{23}$$

and (A, B, C) is controllable and observable; see [14]. Since (23) does not require detailed knowledge of the parameter matrices, this suggests that an adaptive control scheme might be possible.

2.3 The Adaptive Controller

The form of our adaptive controller remains (6). In this we must develop the gain adaptation laws to make asymptotic output tracking possible. We form

$$\begin{cases} \Delta G_u \equiv G_u - S_{22}^* \\ \Delta G_m \equiv G_m - S_{21}^* \\ \Delta G_e \equiv G_e - G_e^* \\ \Delta G_D \equiv G_D - S_{23}^* L \end{cases} \tag{24}$$

where the starred gains come from (9)-(16) and (20). Now, from (6), and (20),

$$\Delta u = u - u_* = \Delta G_u u_m + \Delta G_m x_m + (G_e^* + \Delta G_e) e_y + \Delta G_D \varphi_D \tag{25}$$

Then, via (18) and (25), with appropriate definitions, we have

$$\begin{aligned} \dot{e}_* &= Ae_* + B(u(t-\tau(t)) - u) + B\Delta u + \Delta f \\ &= (A + BG_e^* C)\, e_* + B(u(t-\tau(t)) - u) + B \begin{bmatrix} \Delta G_u & \Delta G_m & \Delta G_e & \Delta G_D \end{bmatrix} \eta + \Delta f \\ &= A_C e_* + B(u(t-\tau(t)) - u) + B\Delta G\eta + \Delta f \end{aligned} \tag{26}$$

where,

$$\eta \equiv [u_m^T x_m^T e_y^T \varphi_D^T]^T$$

is the vector of known available signals.

We combine (18) and (26) to obtain the Tracking Error System:

$$\left\{ \begin{array}{c} \dot{e}_* = A_C e_* + B(u(t-\tau(t)) - u) + B\Delta G\eta + \Delta f \\ e_y = C e_* \end{array} \right. . \tag{27}$$

Now we specify the Adaptive Gain Laws:

$$\dot{G} = -e_y \eta^T H - aG(t) \tag{28}$$

where

$$H \equiv diag[h_{11}, h_{22}, h_{33}, h_{44}] > 0$$

is an arbitrary, diagonal, positive definite matrix. This yields

$$\left\{ \begin{array}{l} \dot{G}_u = -e_y u_m^T h_{11} - aG_u(t) \\ \dot{G}_m = -e_y x_m^T h_{22} - aG_m(t) \\ \dot{G}_e = -e_y e_y^T h_{33} - aG_e(t) \\ \dot{G}_D = -e_y \varphi_D^T h_{44} - aG_D(t) \end{array} \right. \tag{29}$$

3 Robustness of the Adaptive Error System

Our closed loop Adaptive Error System becomes (27) with the above adaptive gain laws (29)

$$\left\{ \begin{array}{c} \dot{e}_* = A_C e_* + B(u(t-\tau(t)) - u) + B\Delta G\eta + \Delta f \\ e_y = C e_* \\ \Delta\dot{G} = \dot{G} = -e_y \eta^T H - aG(t) \end{array} \right. . \tag{30}$$

With the development of the above adaptive error system, recall the theorem developed in [13]

Theorem: Consider the nonlinear, coupled system of differential equations,

$$\left\{ \begin{array}{c} \dot{e} = A_c e + f(e) + B\left(G(t) - G^*\right) z + \nu + f(x) \\ e_y = Ce \\ \dot{G}(t) = -e_y z^{\mathrm{T}} \gamma - aG(t) \end{array} \right. . \tag{31}$$

where G* is any constant matrix and is any positive definite constant matrix, each of appropriate dimension. Assume the following:

1. the delay-free linear part (A_c, B, C) is SPR (see [15]),
2. $\exists M_g > 0 \ni \sqrt{tr(G^* G^{*T})} \leq M_G$
3. $\exists M_v > 0 \ni \sup\limits_{t \geq 0} \|\nu(t)\| \leq M_\nu$
4. $\exists a > 0 \ni a \leq \frac{\beta}{2p_{\min}}; \beta \equiv q_{\min} - 2\mu_f p_{\max} > 0$where p_{min}, p_{max} are the minimum and maximum eigenvalues of P and q_{min} is the minimum eigenvalue of Q with respect to the Kalman-Yacubovich equations,

5. the positive definite matrix γ satisfies

$$tr(\gamma^{-1}) \le \left(\frac{M_\nu}{aM_G}\right)^2$$

6. the nonlinear term f(x) is Lipshitz continuous at 0, i.e.

$$\|f(x)\| \le \mu_f \|x\|$$

with

$$\mu_f < \frac{q_{\min}}{2p_{\max}}$$

Then the gain matrix, G(t), is bounded, and the state, e(t) exponentially with rate approaches the ball of radius

$$R_* \equiv \frac{\left(1+\sqrt{p_{\max}}\right)}{a\sqrt{p_{\min}}} M_\nu$$

We can obtain a corollary of the above theorem for the adaptive error system (30) with the following assumptions:

We will say that the unknown time varying delay $\tau(t)$ is small when

$$\begin{cases} |\tau(t)| \le \tau_* < \infty \\ \|u(t)-u(t-\tau(t))\| \leqslant M(\tau_*) \underset{t_*\to 0}{\to} 0 \end{cases} \tag{32}$$

the above system must have output tracking to a neighborhood:

$$e_y \underset{t\to\infty}{\to} R_* \tag{33}$$

The adaptive controller will have the form:

$$\begin{cases} \dot{G}_u = -e_y u_m^T h_{11} - aG_u(t) \\ \dot{G}_m = -e_y x_m^T h_{22} - aG_m(t) \\ \dot{G}_e = -e_y e_y^T h_{33} - aG_e(t) \\ \dot{G}_D = -e_y \varphi_D^T h_{44} - aG_D(t) \end{cases} . \tag{34}$$

Using the above, we have the following corollary about the corresponding direct adaptive control strategy for the adaptive error system in 30:

Corollary: Assume the following:

1. There exists a gain, G_e^*such that the triple $(A_C \equiv A + BG_e^*C, B, C)$ is SPR (this is known to be equivalent to $CB > 0$ and the open loop transfer function

$$P(s) \equiv C(sI-A)^{-1}B \tag{35}$$

is minimum phase),

2. (32) is satisfied
3. Span(Γ) $\subseteq$ Span(B)

with a positive constants, then the output y exponentially approaches a neighborhood with radius proportional to the magnitude of the disturbance, v, for sufficiently small a and γ_i. Furthermore, each adaptive gain matrix is bounded.

This corollary provides a control law that is robust with respect to persistent disturbances and, exponentially with rate e^{-at}, produces:

$$\overline{\lim_{\tau\to\infty}} \|e(t)\| \leqslant \frac{\left(1+\sqrt{p_{\max}}\right)}{a\sqrt{p_{\min}}} \|B\|\, M(\tau) \underset{t\to 0}{\to} 0.$$

The Proof of the Corollary is provided in the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part II.

4 Simulation and Results

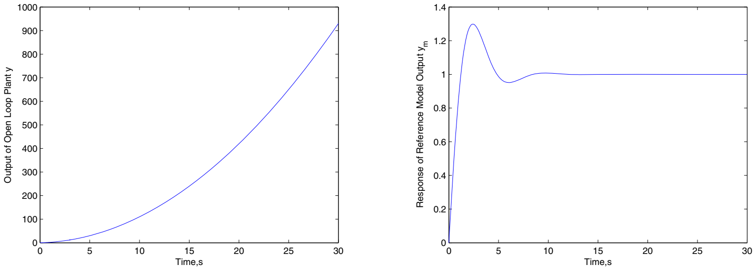
We will illustrate the above robust adaptive controller on the following plant:

$$\begin{cases} \dot{x} = \underbrace{\begin{bmatrix} x_2 \\ 0.3*\sin(x_1) \end{bmatrix}}_{A(x)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u(t-\tau) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\Gamma} u_D \\ y = \underbrace{\begin{bmatrix} 1 \; 0.1 \end{bmatrix}}_{C} x \end{cases} \tag{36}$$

We use step disturbances to provide simulation results for various small time varying values of delay $\tau(t)$. An adequate reference model must be developed for output tracking. The open loop output response to a step disturbance of magnitude 1 can be seen in Fig. 1(a). The desired reference model output, $y_m(t)$, for the closed loop reference model linear plant and lead controller to a step disturbance of magnitude 1 can be seen in Fig. 1(b). This reference model output was created by designing a lead controller to stabilize the plant and achieve the desired temporal response characteristics. Further simulations to illustrate this adaptive control method are provided in the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part II.

4.1 Step Disturbances

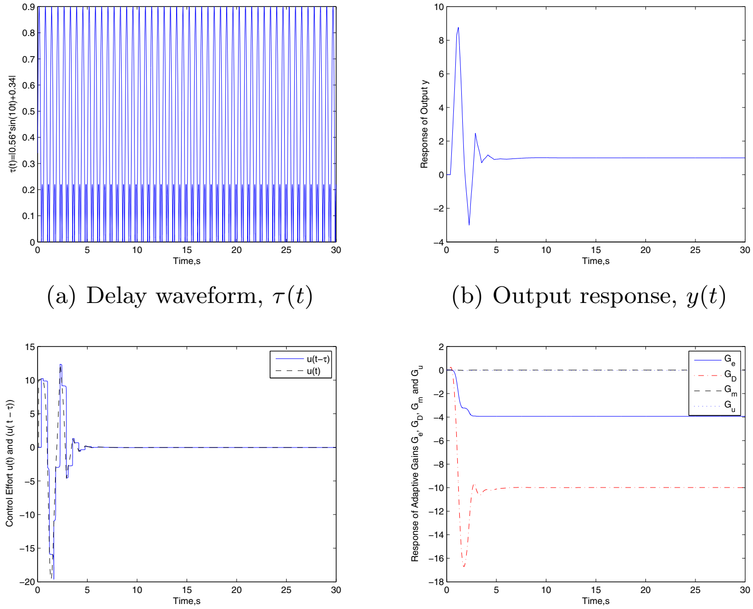
The waveform of time varying delay $\tau(t) = |0.56*sin(10t)+0.34|(s)$ is shown in Fig. 2(a). The response to a step disturbance of magnitude 10 of the output response, y(t), control effort u(t), and the adaptive gains for the input delay time, $\tau(t) = |0.56*sin(10t)+0.34|(s)$ are shown in Fig. 2(b), 2(c) and 2(d)



(a) Output response, $y(t)$

(b) Output response, $y_m(t)$

Fig. 1 Output response, (a) $y(t)$, for the open loop plant and (b) $y_m(t)$, for the closed loop reference model plant and lead controller to a step disturbance of magnitude 1



(a) Delay waveform, $\tau(t)$

(b) Output response, $y(t)$

(c) Control Effort, u(t)

(d) Response of adaptive gains

Fig. 2 (a)Delay waveform, $\tau(t)$, (b) Output response, $y(t)$, (c) Control Effort, u(t) and (d) Response of adaptive gains for a step disturbance of magnitude 10 and $\tau(t) = |0.56 * sin(10t) + 0.34|$(s)

respectively. This simulation has shown that the adaptive controller can force a simple midly-nonlinear plant to adequately track a linear reference model. The adaptive controller can operate in the presence of “small” constant and time varying delays without any knowledge of the delay.

5 Conclusions

In this paper, we developed a Direct Model Reference Adaptive Tracking Controller for mildly non-linear systems with unknown time varying input delays. This controller can also reject bounded disturbances of known wave form but unknown amplitude, e.g. steps or sinusoids. In this paper a robustness result was developed for DMRAC of mildly non-linear systems with unknown small constant or time varying input delays using the concept of un-delayed ideal trajectories. We showed that the adaptively controlled system is globally stable, but the adaptive tracking error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood can be achieved as a result of the control design. A simple example was provided to illustrate this adaptive control method. The proof of the corollary for the application and further examples are provided in the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part II.

References

1. Sipahi, R., Niculescu, S.-I., Abdallah, C.T., Michiels, W., Gu, K.: Stability and stabalization of systems with time delay. IEEE Control Systems Magazine 31(1), 38–65 (2001)
2. Gu, K., Kharitonov, V.L., Chen, J.: Stability of Time Delay Systems. Bikhauser (2003)
3. Niculescu, S.-I.: Delay Effects on Stability. Springer (2001)
4. Luna, J.: Time delay systems and applications on sattelite control, University of New Mexico/ARFL Space Vechicles, Tech. Rep.
5. Richard, J.-P.: Time delay systems: an overview of some recent advances and open problems. Science Direct Automatica 39, 1667–1694 (2003)
6. Krstic, M.: Compensation of Infinite-Dimensional Actuator and Sensor Dynamics: Nonlinear Delay-Adaptive Systems. IEEE Control Systems Magazine, 22–41 (February 2010)
7. Seuret, A.: Networked Control Under Synchronization Errors. In: Proc. of American Control Conference, Seattle, WA (2008)
8. Balas, M., Erwin, R.S., Fuentes, R.: Adaptive control of persistent disturbances for aerospace structures. In: AIAA GNC, Denver (2000)
9. Fuentes, R., Balas, M.: Direct Adaptive Rejection of Persistent Disturbances. Journal of Mathematical Analysis and Applications 251, 28–39 (2000)
10. Fuentes, R., Balas, M.: Disturbance accommodation for a class of tracking control systems. In: AIAA GNC, Denver, Colorado (2000)
11. Fuentes, R., Balas, M.: Robust Model Reference Adaptive Control with Disturbance Rejection. In: Proc. ACC (2002)
12. Balas, M., Gajendar, S., Robertson, L.: Adaptive Tracking Control of Linear Systems with Unknown Delays and Persistent Disturbances (or Who You Callin Retarded?). In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL (Augest 2009)

13. Balas, M., Nelson, J., Gajendar, S., Robertson, L.: Robust Adaptive Control of Nonlinear Systems with Input/ Output Delays. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, OR (August 2011)
14. Nelson, J., Balas, M., Erwin, R.: Direct Model Reference Adaptive Control of Linear Systems with Unknown Time Varying Input/Output Delays. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Minneapolis, MN (August 2012)
15. Wen, J.: Time domain and frequency domain conditions for strict positive realness. IEEE Trans. Automat. Contr. 33(10), 988–992 (1988)

Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays – Part II

James P. Nelson, Mark J. Balas, and Richard S. Erwin

Abstract. In this paper, a proof for the corollary developed for the Direct Model Reference Adaptive Tracking Control of mildly non-linear systems with unknown time varying input delays found in Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays - Part I is completed. The adaptive error system was developed for the DMRAC of mildly non-linear systems with unknown small constant or time varying input delays using the concept of un-delayed ideal trajectories. We will show that the adaptively controlled system is globally stable, but the adaptive tracking error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood can be achieved as a result of the control design. A simple example will be provided to illustrate this adaptive control method.

1 Introduction

This paper is the companion to Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays - Part I. The introduction and some of the theoretical development will be restated so it can be read as a stand alone paper. Time delay affects many engineering, physics and biological systems [1]-[5]. These manuscripts present a firm motivation for the study of time delay systems and a brief overview of the different control approaches commonly used when delays are present. In this overview the

James Nelson · Mark J. Balas
University of Wyoming,
Department of Electrical and Computer Engineering,
1000 E. Universtiy Ave. Laramie, WY 82073
e-mail: {nelsonpj,mbalas}@uwyo.edu

Richard S. Erwin
Air Force Research Laboratory, Kirtland AFB, NM, 87117, USA

open problem of control via the delay and constructive use of the delayed inputs is presented [5].Further, many control systems suffer from unknown delays [6]-[7]. Often these are introduced via systems controlled through a network, e.g in [7].

In previous work [8]-[11] direct model reference adaptive control (DMRAC) and disturbance rejection with very low order adaptive gain laws for MIMO systems was accomplished. Feuntes and Balas developed an ultimate bounded-ness theorem for DMRAC in [11]. When systems are subjected to an unknown internal delay, the adaptive control theory can be modified to handle this situation [12]. However, delays appearing in the inputs or outputs of systems seem to cause more system sensitivity to the delay. A robustness result for the Direct Adaptive Control (DAC) or input delay systems was developed in [13]. A robustness result for the DMRAC of linear systems with "small" input/output delays was developed in [14] using the concept of undelayed ideal trajectories for the development of the adaptive error system. Using the concept of un-delayed ideal trajectories and this "small-ness" assumption the results of [13] can achieved for the DMRAC of mildly non-linear systems. We will show that the adaptively controlled system is globally stable, but the adaptive error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood can be achieved as a result of the control design.

2 Robustness of the Adaptive Error System

In the paper: Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delay - Part I the concept of "undelayed ideal trajectories" was used to develop the adaptive error system:

$$\left\{\begin{array}{c} \dot{e}_* = A_C e_* + B(u(t-\tau(t)) - u) + B\Delta G\eta + \Delta f \\ e_y = Ce_* \\ \Delta\dot{G} = \dot{G} = -e_y\eta^T H - aG(t) \end{array}\right. . \tag{1}$$

Recall the theorem developed in [13]

Theorem: Consider the nonlinear, coupled system of differential equations,

$$\left\{\begin{array}{c} \dot{e} = A_c e + f(e) + B\left(G(t) - G^*\right) z + \nu + f(x) \\ e_y = Ce \\ \dot{G}(t) = -e_y z^{\mathrm{T}}\gamma - aG(t) \end{array}\right. . \tag{2}$$

where G* is any constant matrix and is any positive definite constant matrix, each of appropriate dimension. Assume the following:

1. the delay-free linear part (A_c, B, C) is SPR (see [15]),
2. $\exists M_g\, 0 \ni \sqrt{tr(G^* G^{*T})} \leq M_G$

3. $\exists M_v\ 0 \ni \sup_{t\geq 0} \|\nu(t)\| \leq M_\nu$
4. $\exists a\ 0 \ni a \leq \frac{\beta}{2p_{\min}}; \beta \equiv q_{\min} - 2\mu_f p_{\max} > 0$where p_{min}, p_{max} are the minimum and maximum eigenvalues of P and q_{min} is the minimum eigenvalue of Q with respect to the Kalman-Yacubovich equations,
5. the positive definite matrix γ satisfies

$$tr(\gamma^{-1}) \leq \left(\frac{M_\nu}{aM_G}\right)^2$$

6. the nonlinear term f(x) is Lipshitz continuous at 0, i.e.

$$\|f(x)\| \leq \mu_f \|x\|$$

with

$$\mu_f < \frac{q_{\min}}{2p_{\max}}$$

Then the gain matrix, G(t), is bounded, and the state, e(t) exponentially with rate approaches the ball of radius

$$R_* \equiv \frac{\left(1+\sqrt{p_{\max}}\right)}{a\sqrt{p_{\min}}} M_\nu$$

We can obtain a corollary of the above theorem for the adaptive error system in (1) with the following assumptions:

We will say that the unknown time varying delay $\tau(t)$ is small when

$$\begin{cases} |\tau(t)| \leq \tau_* < \infty \\ \|u(t) - u(t-\tau(t))\| \leqslant M(\tau_*) \underset{t_* \to 0}{\to} 0 \end{cases} \tag{3}$$

the above system must have output tracking to a neighborhood:

$$e_y \underset{t\to\infty}{\to} R_* \tag{4}$$

The adaptive controller will have the form:

$$\begin{cases} \dot{G}_u = -e_y u_m^T h_{11} - aG_u(t) \\ \dot{G}_m = -e_y x_m^T h_{22} - aG_m(t) \\ \dot{G}_e = -e_y e_y^T h_{33} - aG_e(t) \\ \dot{G}_D = -e_y \varphi_D^T h_{44} - aG_D(t) \end{cases} . \tag{5}$$

Using the above, we have the following corollary about the corresponding direct adaptive control strategy the adaptive error system in 1:

Corollary: Assume the following:

1. There exists a gain, G_e^*such that the triple $(A_C \equiv A + BG_e^* C, B, C)$ is SPR (this is known to be equivalent to $CB > 0$ and the open loop transfer function
$$P(s) \equiv C(sI - A)^{-1}B \tag{6}$$
is minimum phase),
2. (3) is satisfied
3. Span($\mathit{\Gamma}$) $\subseteq$ Span(B)

with a positive constants, then the output y exponentially approaches a neighborhood with radius proportional to the magnitude of the disturbance, v, for sufficiently small a and γ_i. Furthermore, each adaptive gain matrix is bounded.

This corollary provides a control law that is robust with respect to persistent disturbances and, exponentially with rate e^{-at}, produces:

$$\overline{\lim_{\tau \to \infty}} \|e(t)\| \leqslant \frac{\left(1 + \sqrt{p_{\max}}\right)}{a\sqrt{p_{\min}}} \|B\| \, M(\tau) \underset{t \to 0}{\to} 0.$$

Proof

We form the Energy Storage Functions:

$$V = \frac{1}{2} e^{\mathrm{T}} P e + \frac{1}{2} \mathrm{tr} \left[\mathit{\Delta} G \gamma^{-1} \mathit{\Delta} G^{\mathrm{T}} \right] \tag{7}$$

where $tr\, Q \equiv \sum\limits_{i=1}^{N} q_{ii}$ and $P > 0$ is the solution of the following pair of equations:

$$\begin{cases} A_c^T P + P A_c = -Q < 0 \\ \qquad PB = C^T \end{cases} \tag{8}$$

These equations are usually known as the Kalman-Yacubovic Conditions. The existence of a symmetric positive definite solution of (8) is known to be equivalent to the following condition:

$$T_c(s) \equiv C(sI - A_c)^{-1}B \tag{9}$$

strict positive realness (SPR). $T_C(s)$ (SPR) means, for some $\sigma > 0$,

$$\mathrm{Re}\, T_C(-\sigma + j\omega) \geqslant 0 \tag{10}$$

for all ω real. When the open-loop system (A, B, C) can be made SPR by output feedback $A_C \equiv A + BG_e^* C$, we say the open-loop system is almost strictly positive real (ASPR). This is known to be equivalent to $CB > 0$ and the open-loop $T(s) \equiv C(sI - A)^{-1} B$ being minimum phase, i.e. all transmission zeros stable; for example, see [16]. If we calculate the derivatives along the trajectories of (7), we have, using (1), that

$$\dot{V} = e^{\mathrm{T}} P A_c \, e + e^T P B w + e^T P \Delta f + \mathrm{tr}\left[\Delta \dot{G} \gamma^{-1} \Delta G^{\mathrm{T}}\right] + \nu^{\mathrm{T}} P e;$$

where

$$w \equiv \Delta G \eta$$

and

$$v \equiv B(u(t - \tau(t)) - u).$$

Invoking the equalities in the definition of SPR and substituting into the last expression, we get

$$\begin{cases} \dot{V} = -\frac{1}{2} e^{\mathrm{T}} Q e + \langle e_y, w \rangle + e^T P \Delta f - a \cdot \mathrm{tr}\left[G \gamma^{-1} \Delta G^{\mathrm{T}}\right] - \underbrace{tr(e_y z^T \Delta G^T)}_{\langle e_y, w \rangle} + \nu^{\mathrm{T}} P e \\ \qquad \leq -\frac{1}{2} (\underbrace{q_{\min} - 2\mu_f p_{\max}}_{\beta}) \, \|e\|^2 - a \cdot \mathrm{tr}\left[(\Delta G + G^*) \gamma^{-1} \Delta G^{\mathrm{T}}\right] + \nu^{\mathrm{T}} P e \\ \qquad \leq -(\frac{1}{2} \beta \|e\|^2 + a \cdot \mathrm{tr}\left[\Delta G \gamma^{-1} \Delta G^{\mathrm{T}}\right]) + a \cdot \left|\mathrm{tr}\left[G^* \gamma^{-1} \Delta G^{\mathrm{T}}\right]\right| + \left|v^{\mathrm{T}} P e\right| \\ \qquad \leq -(\frac{\beta}{2 p_{\min}} e^T P e + 2a \bullet \frac{1}{2} \mathrm{tr}\left[\Delta G \gamma^{-1} \Delta G^{\mathrm{T}}\right]) + a \cdot \left|\mathrm{tr}\left[G^* \gamma^{-1} \Delta G^{\mathrm{T}}\right]\right| + \left|v^{\mathrm{T}} P e\right| \\ \qquad \leq -2aV + a \cdot \left|\mathrm{tr}\left[G^* \gamma^{-1} \Delta G^{\mathrm{T}}\right]\right| + \left|v^{\mathrm{T}} P e\right| \end{cases} .$$

Now, using the Cauchy-Schwartz Inequality

$$\left|\mathrm{tr}\left[G^* \gamma^{-1} \Delta G^{\mathrm{T}}\right]\right| \leq \|G^*\|_2 \|\Delta G\|_2$$

and

$$\left|v^{\mathrm{T}} P e\right| \leq \left\|P^{\frac{1}{2}} \nu\right\| \, \left\|P^{\frac{1}{2}} e\right\| = \sqrt{v^T P \nu} \bullet \sqrt{e^T P e}$$

We will say that the unknown delay $\tau(t)$ is small when (3) is satisfied so,

$$\|\nu\| \equiv \|B\| \, \|u(t) - u(t - \tau(t))\| \leqslant \|B\| \, M(\tau).$$

We have

$$\begin{aligned} &\dot{V} + 2aV \leqslant a \cdot \|G^*\|_2 \|\Delta G\|_2 + \sqrt{p_{\max}} \, \|\nu\| \, \sqrt{e^T P e} \\ &\leqslant a \cdot \|G^*\|_2 \|\Delta G\|_2 + (\sqrt{p_{\max}} \|B\| \, M(\tau)) \sqrt{e^T P e} \\ &\leqslant (a \|G^*\|_2 + \sqrt{p_{\max}} \|B\| \, M(\tau)) \sqrt{2} \underbrace{[\frac{1}{2} e^T P e + \frac{1}{2} \, \|\Delta G\|_2^2]^{\frac{1}{2}}}_{V^{\frac{1}{2}}} \\ &\therefore \frac{\dot{V} + 2aV}{V^{\frac{1}{2}}} \leqslant (a \|G^*\|_2 + \sqrt{p_{\max}} \|B\| \, M(\tau)) \sqrt{2} \qquad . \end{aligned}$$

Now, using the identitytr $[ABC] = \mathrm{tr}\,[CAB]$,

$$\begin{aligned}&\|G^*\|_2 \equiv [tr(G^*\gamma^{-1}(G^*)^T)]^{\frac{1}{2}} = [tr((G^*)^T G^*\gamma^{-1})]^{\frac{1}{2}}\\ &\leq [(tr((G^*)^T G^*(G^*)^T G^*)^{\frac{1}{2}}(tr(\gamma^{-1}\gamma^{-1})^{\frac{1}{2}}]^{\frac{1}{2}}\\ &= [tr(G^*(G^*)^T)]^{\frac{1}{2}}[tr\gamma^{-1}]^{\frac{1}{2}} \leq \tfrac{\|B\|M(\tau)}{aM_G} \bullet M_G = \tfrac{\|B\|M(\tau)}{a}\end{aligned}$$

$$\Rightarrow \frac{\dot{V}+2aV}{V^{\frac{1}{2}}} \leq (1+\sqrt{p_{\max}})\|B\|\,M(\tau))\sqrt{2}, \tag{11}$$

from

$$\frac{d}{dt}(2e^{at}V^{\frac{1}{2}}) = e^{at}\frac{\dot{V}+2aV}{V^{\frac{1}{2}}} \leq e^{at}(1+\sqrt{p_{\max}})\|B\|\,M(\tau))\sqrt{2}.$$

Integrating this expression we have:

$$e^{at}V(t)^{1/2} - V(0)^{1/2} \leq \frac{\left(1+\sqrt{p_{\max}}\right)\,\|B\|\,M(\tau)}{a}\left(e^{at}-1\right)$$

$$\therefore V(t)^{1/2} \leq V(0)^{1/2}e^{-at} + \frac{\left(1+\sqrt{p_{\max}}\right)\,\|B\|\,M(\tau)}{a}\left(1-e^{-at}\right) \tag{12}$$

The function V is a norm function of the state e(t) and matrix G(t): so, since $V^{\frac{1}{2}}$ is bounded for all t, then e(t) and G(t) are bounded. We also have the following inequality:

$$\sqrt{p_{\min}}\,\|e(t)\| \leq V(t)^{1/2}.$$

Substitution of this into (12) gives us an exponential bound on state $e(\tau)$:

$$\|e(t)\| \leq \frac{e^{-at}}{\sqrt{p_{\min}}}V(0)^{1/2} + \frac{\left(1+\sqrt{p_{\max}}\right)\,\|B\|\,M(\tau)}{a\sqrt{p_{\min}}}\left(1-e^{-at}\right) \tag{13}$$

Taking the limit superior of (13), we have

$$\overline{\lim_{\tau\to\infty}}\,\|e(t)\| \leq \frac{\left(1+\sqrt{p_{\max}}\right)}{a\sqrt{p_{\min}}}\|B\|\,M(\tau) \equiv R_* \tag{14}$$

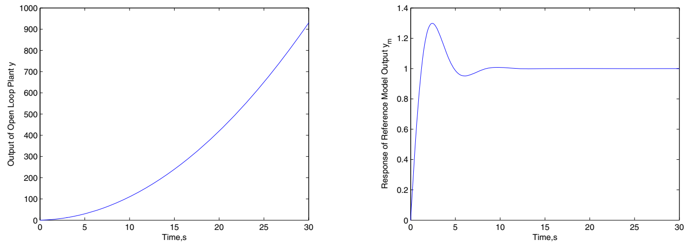
#

3 Simulation and Results

We will illustrate the above robust adaptive controller on the following plant:

$$\begin{cases}\dot{x} = \underbrace{\begin{bmatrix} x_2 \\ 0.3*\sin(x_1)\end{bmatrix}}_{A(x)} + \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{B}u(t-\tau) + \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\mathit{\Gamma}}u_D \\ y = \underbrace{\begin{bmatrix}1 \; 0.1\end{bmatrix}}_{C}x\end{cases} \tag{15}$$

We use step disturbances to provide simulation results for various small time varying values of delay $\tau(t)$. An adequate reference model must be developed for output tracking. The open loop output response to a step disturbance of magnitude 1 can be seen in Fig. 1(a). The desired reference model output, $y_m(t)$, for the closed loop reference model linear plant and lead controller to a step disturbance of magnitude 1 can be seen in Fig. 1(b). This reference model output was created by designing a lead controller to stabilize the plant and achieve the desired temporal response characteristics.



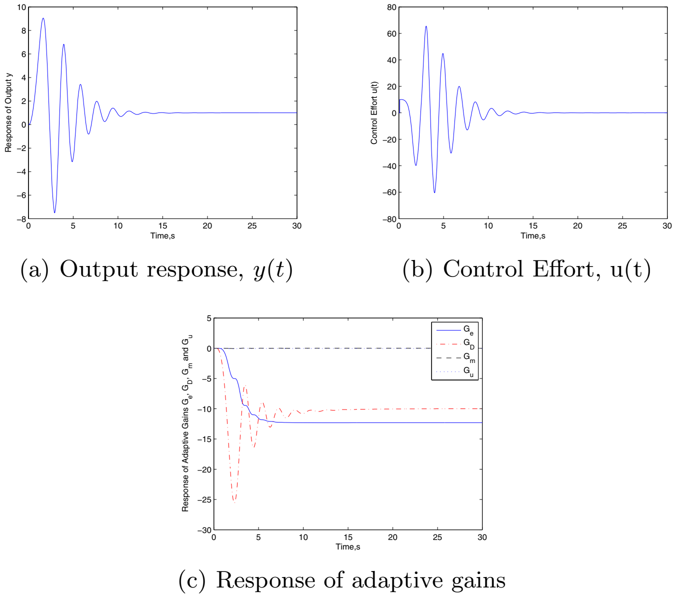
(a) Output response, $y(t)$

(b) Output response, $y_m(t)$

Fig. 1 Output response, (a) $y(t)$, for the open loop plant and (b) $y_m(t)$, for the closed loop reference model plant and lead controller to a step disturbance of magnitude 1

3.1 Step Disturbances

The response to a step disturbance of magnitude 10 of the output response, y(t), control effort u(t), and the adaptive gains for no input delay are shown in Fig. 2(a), 2(b) and 2(c) respectively.The response to a step disturbance of magnitude 10 of the output response, y(t), control effort u(t), and the adaptive gains for the input delay time, $\tau = 0.09s$ are shown in Fig. 3(a), 3(b) and 3(c) respectively. The response to a step disturbance of magnitude 10 of the output response, y(t), control effort u(t), and the adaptive gains for the input delay time, $\tau = 0.115s$ are shown in Fig. 4(a), 4(b) and 4(c) respectively. It can be seen that the adaptive error system adequately tracks the desired reference model output for the delay free system and the "small" delay case. As the constant delay grows, the adaptive system still tracks the desired reference model output, albeit with poor temporal characteristics. The waveform of time varying delay $\tau(t) = |0.56 * sin(10t) + 0.34|(s)$ is shown in Fig. 5(a). The response to a step disturbance of magnitude 10 of the output response, y(t), control effort u(t), and the adaptive gains for the input delay time, $\tau(t) = |0.56 * sin(10t) + 0.34|(s)$ are shown in Fig. 5(b), 5(c) and 5(d) respectively. This simulation has shown that the adaptive controller can force a simple midly-nonlinear plant to adequately track a linear reference model. The adaptive controller can operate in the presence of "small" constant and time varying delays without any knowledge of the delay.

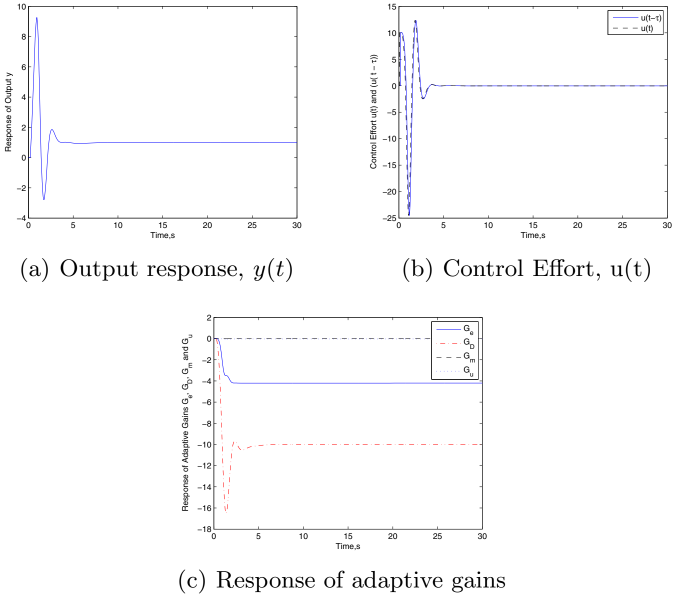


(a) Output response, $y(t)$

(b) Control Effort, u(t)

(c) Response of adaptive gains

Fig. 2 (a) Output response, $y(t)$, (b) Control Effort, u(t) and (c) Response of adaptive gains for a step disturbance of magnitude 10 and no delay

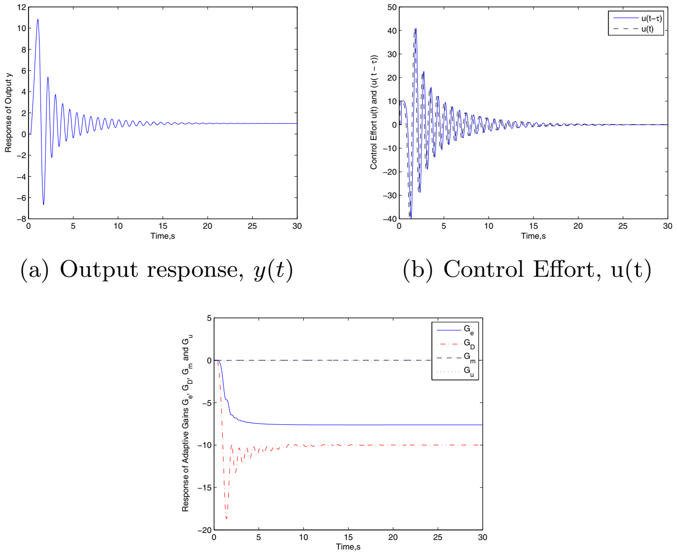


(a) Output response, $y(t)$

(b) Control Effort, u(t)

(c) Response of adaptive gains

Fig. 3 (a) Output response, $y(t)$, (b) Control Effort, u(t) and (c) Response of adaptive gains for a step disturbance of magnitude 10 and $\tau = 0.09s$

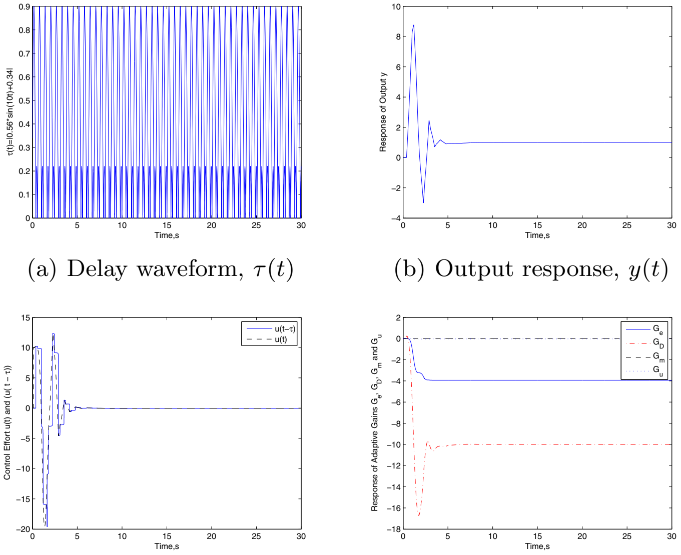


(a) Output response, $y(t)$

(b) Control Effort, u(t)

(c) Response of adaptive gains

Fig. 4 (a) Output response, $y(t)$, (b) Control Effort, u(t) and (c) Response of adaptive gains for a step disturbance of magnitude 10 and $\tau = 0.155s$



(a) Delay waveform, $\tau(t)$

(b) Output response, $y(t)$

(c) Control Effort, u(t)

(d) Response of adaptive gains

Fig. 5 (a)Delay waveform, $\tau(t)$, (b) Output response, $y(t)$, (c) Control Effort, u(t) and (d) Response of adaptive gains for a step disturbance of magnitude 10 and $\tau(t) = |0.56 * sin(10t) + 0.34|$(s)

4 Conclusions

In this paper, a proof for the corollary developed for the Direct Model Reference Adaptive Tracking Control of mildly non-linear systems with unknown time varying input delays found in Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays - Part I was completed. The adaptive error system was developed for the DMRAC of mildly non-linear systems with unknown small constant or time varying input delays using the concept of un-delayed ideal trajectories. It has been shown that the adaptively controlled system is globally stable, but the adaptive tracking error is no longer guaranteed to approach the origin. However, exponential convergence to a neighborhood is achieved as a result of the control design. A simple example was provided to illustrate this adaptive control method.

References

1. Sipahi, R., Niculescu, S.-I., Abdallah, C.T., Michiels, W., Gu, K.: Stability and stabalization of systems with time delay. IEEE Control Systems Magazine 31(1), 38–65 (2001)
2. Gu, K., Kharitonov, V.L., Chen, J.: Stability of Time Delay Systems. Bikhauser (2003)
3. Niculescu, S.-I.: Delay Effects on Stability. Springer (2001)
4. Luna, J.: Time delay systems and applications on sattelite control, University of New Mexico/ARFL Space Vechicles, Tech. Rep.
5. Richard, J.-P.: Time delay systems: an overview of some recent advances and open problems. Science Direct Automatica 39, 1667–1694 (2003)
6. Krstic, M.: Compensation of Infinite-Dimensional Actuator and Sensor Dynamics: Nonlinear Delay-Adaptive Systems. IEEE Control Systems Magazine, 22–41 (February 2010)
7. Seuret, A.: Networked Control Under Synchronization Errors. In: Proc. of American Control Conference, Seattle, WA (2008)
8. Balas, M., Erwin, R.S., Fuentes, R.: Adaptive control of persistent disturbances for aerospace structures. In: AIAA GNC, Denver (2000)
9. Fuentes, R., Balas, M.: Direct Adaptive Rejection of Persistent Disturbances. Journal of Mathematical Analysis and Applications 251, 28–39 (2000)
10. Fuentes, R., Balas, M.: Disturbance accommodation for a class of tracking control systems. In: AIAA GNC, Denver, Colorado (2000)
11. Fuentes, R., Balas, M.: Robust Model Reference Adaptive Control with Disturbance Rejection. In: Proc. ACC (2002)
12. Balas, M., Gajendar, S., Robertson, L.: Adaptive Tracking Control of Linear Systems with Unknown Delays and Persistent Disturbances (or Who You Callin Retarded?). In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL (August 2009)
13. Balas, M., Nelson, J., Gajendar, S., Robertson, L.: Robust Adaptive Control of Nonlinear Systems with Input/ Output Delays. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, OR (August 2011)

14. Nelson, J., Balas, M., Erwin, R.: Direct Model Reference Adaptive Control of Linear Systems with Unknown Time Varying Input/Output Delays. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Minneapolis, MN (August 2012)
15. Wen, J.: Time domain and frequency domain conditions for strict positive realness. IEEE Trans. Automat. Contr. 33(10), 988–992 (1988)
16. Balas, M., Fuentes, R.: A Non-orthogonal Projection Approach to Characterization of Almost Positive Real Systems with an Application to Adaptive Control. In: Proceedings of American Control Conference, Boston (2004)

Flight Control Algorithms for a Vertical Launch Air Defense Missile

Raziye Tekin, Ozgur Atesoglu, and Kemal Leblebicioglu

Abstract. The necessity of high maneuverability and vertical launching require thrust vector control additional to aerodynamic control. That hybrid usage of aerodynamic and thrust vectoring controls effectively increases the agility of the missile against air defense threats. This requirement and the rapidly changing dynamics of this type of missiles renders the guidance and control design critical. However, the findings suggest that classical guidance and control design approaches are still valuable to apply and can have successful performance within the effective flight envelope. It is very rare that a study concerns from detailed dynamics and analysis of the dynamics covering flight mission and algorithms. In this study, together with the modeling of the agile dynamics of a vertical launch surface to air missile and the corresponding thrust forces and moments depending on linear supersonic theory, the application of the flight control algorithms are presented. Two classic linear autopilot structures are studied. During autopilot design process, an additional term related to short period dynamics of boost phase is proposed and the drastic effect of this term is shown. In addition to control algorithms, guidance algorithms are also defined to fulfill the mission of the missile. Body pursuit algorithm is applied for rapid turnover maneuver and midcourse guidance. Proportional navigation guidance is chosen for terminal phase. In addition, an alternative maneuvering technique is proposed to reduce further side slip angle during vertical flight.

Raziye Tekin
Institute of Robotics and Mechatronics,
German Aerospace Center - DLR, Muenchner Str. 20,
82234 Wessling, Germany
e-mail: raziye.tekin@dlr.de

Ozgur Atesoglu
Mechanical Engineering, Middle East Technical University,
06531, Ankara, Turkey

Kemal Leblebicioglu
Electrical and Electronics Engineering,
Middle East Technical University, 06531, Ankara, Turkey

1 Introduction

This paper presents the practical application of guidance and control methods for a vertical launch surface to air missile (VLSAM). Through the paper, the challenging dynamics of the mentioned missile is briefly presented with the modeling of the thrust vector control forces and moments, that are formed by the jet vane deflections, originating from the linear supersonic theory. There are many advanced control methods for such a rapidly varying dynamics. However, the authors of this papers advocate from a practical point of view that developing advanced control techniques should be an option only after classical control techniques have been proven to be inadequate. Hence, classical control techniques and ad-hoc gain scheduling is applied for the VLSAM. The autopilot design is pursued separately for the mid-course and terminal guidance phases of the flight. Angle autopilots are designed for the mid-course, including the rapid turnover maneuver and acceleration autopilot for terminal guidance phase respectively. The gains of the autopilot are scheduled with respect to time during boost phase and Mach for the post boost phase. The performance of the autopilots are analyzed within nonlinear simulation.

The effect of the axial acceleration during the boost phase is emphasized. Recent studies covering boost phase do not present such a term. The effect is illustrated within the simulation results regarding the comparison of schedules linearized system with the nonlinear system. The second important issue is to define a constant hybrid control ratio that interconnects the thrust vector with aerodynamic control; a singular value analysis of the linear control influence is conducted for that reason.

In addition, autopilots are integrated with body pursuit and proportional navigation guidance (PNG) guidance schemes. The overall guidance and control design is tested for a defense maneuver to defeat an approaching target. In literature, a study regarding the initial roll maneuver for interceptors based on fuzzy guidance has been found [1]. Here an alternative maneuver based on a basic approach: initial roll command generation to minimize side slip angle during vertical flight is studied which is very practical and efficient to implement.

2 Modeling of the Vertical Launch Missile Dynamics

Dynamic modeling of the VLSAM is carried out by implementing the well known Newton-Euler equations with rigid body assumption. The VLSAM, analyzed in this paper, is axi-symmetric and has a blunt nose. It is a tail controlled missile and uses both the aerodynamic tail fins and jet vanes. Two main coordinate systems as the body coordinate system (B) and the earth fixed inertial coordinate system (E) are defined and the equations of motion are derived with respect to them. The origin of the body axis system is assumed to be at the final center of gravity location after burnout. Also, since the propellant of the missile is burning throughout the flight, the mass, inertia and the position of center of gravity are formulized as a function of thrust and total impulse values and included in the model. Hence, since the thrust and impulse are modeled as a function of time, the mass, inertia and center of gravity position of the missile change as a function of time. Hence, the translational and rotational motion of the missile can be written as Eq. 1. Detailed information about the dynamics and aerodynamics can be found in [2].

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} rv - qw + (F_{Ax} + F_{Gx} + F_{Tx})/m \\ pw - ru + (F_{Ay} + F_{Gy} + F_{Ty})/m \\ qu - pv + F_{Az} + (F_{Gz} + F_{Tz})/m \\ (M_{Ax} + M_{Tx})/I_{xx} \\ (M_{Ay} + M_{Ty} - pr(I_{xx} - I_{yy}))/I_{yy} \\ (M_{Az} + M_{Tz} - pq(I_{xx} - I_{yy}))/I_{yy} \end{bmatrix} \tag{1}$$

Here, $\bar{F}^{(B)}_{A(x,y,z)}$, $\bar{F}^{(B)}_{T(x,y,z)}$ and $\bar{F}^{(B)}_{G(x,y,z)}$ are the Cartesian components of the aerodynamic, thrust and gravity forces. $\bar{M}^{(B)}_{A(x,y,z)}$, $\bar{M}^{(B)}_{T(x,y,z)}$ and are the Cartesian components of the aerodynamic and thrust moments. u, v, w are the components of missile velocity in body coordinates, p, q, r are the missile body rates, m is mass. As, the VLSAM in this study is axi-symmetric and has cruciform geometry, $I_{yy} = I_{zz}$.

As for the calculation of the rates of the Euler angles, to avoid the singularity when the pitch angle is equal to $\mp 90^\circ$, direction cosine matrix (DCM) formulation is rather preferred to the Euler angle formulation to avoid the singularity problem. Although the quaternion formulation is computationally more efficient, it is not chosen because the DCM is more practical to apply and interpret physically.

2.1 Aerodynamic Forces and Moments

The aerodynamic forces/moments are functions of dynamic pressure (Q_d), missile reference area (S) and the aerodynamic force coefficients, i.e. C_i 's. Hence the aerodynamic force vector in matrix representation can be written:

$$\begin{aligned} \bar{F}^{(B)}_{A} &= \begin{bmatrix} F_{Ax} \; F_{Ay} \; F_{Az} \end{bmatrix} = Q_d S \begin{bmatrix} C_x \; C_y \; C_z \end{bmatrix} \\ \bar{M}^{(B)}_{A} &= Q_d S \begin{bmatrix} C_l \; C_m \; C_n \end{bmatrix} + \begin{bmatrix} 0 \\ (x_{cgref} - x_c(t))F_{Az} \\ (x_{cgref} - x_c(t))F_{Ay} \end{bmatrix} \end{aligned} \tag{2}$$

The aerodynamic force/moment coefficients as nonlinear functions of flight variables; $Ma, \alpha, \beta, \delta_a, \delta_e, \delta_r$ where Ma is the Mach number, α is the angle of attack and β is the angle of sideslip, δ_a, δ_r, δ_r are the aileron, elevator, rudder deflections of aerodynamic control surfaces. x_{cref} is the final, i.e. after the propellant burn-out, and $x_c(t)$ is the instantaneous position of the center of gravity, Q_d is the dynamic pressure, S is reference surface. the VLSAM has high angle of attack flight regime, so that an aerodynamic database is created including ±90 degrees of angle of attack and cross coupling terms. The details of the aerodynamic modeling and analysis of the VLSAM can be found in [2].

2.2 Thrust Forces and Moments

The thrust forces and moments, are generated by deflecting the thrust vector by the jet vanes located at the nozzle exit of the missile. The magnitude of thrust force (T), duration of the boost phase, the time to reach the maximum thrust level and the geometrical properties of the jet vanes are the critical parameters for thrust vectoring. These parameters are adapted to achieve desired maneuvering capabilities for the VLSAM [3]. The maximum thrust vectoring forces and moments are dependent on the maximum jet vane deflection angle and vane characteristics such as thickness (t_k), chord (c) and thrust motor characteristics as nozzle exit pressure (P) and flow velocity (M_∞). To determine the lift and drag forces, linear supersonic theory is directly applied. The area of the jet vane is not changing or may be neglected. There are studies that include the jet vane erosion phenomenon which effects the lift and drag forces created by vane deflections, [4], [5], [6] and [7].

$$L = \frac{PC_L \gamma M_\infty^2 S_{jv}}{2}, \;\; D = \frac{PC_D \gamma M_\infty^2 S_{jv}}{2} \tag{3}$$

Here, lift and drag force coefficients (C_L and C_D) are functions of deflection of jet vanes (δ_{JV}), nozzle exit pressure, thickness and chord of the jet vanes:

$$C_L = \frac{4\delta_{JV}}{\sqrt{M_\infty^2 - 1}}, \;\; C_D = \frac{4}{\sqrt{M_\infty^2 - 1}}(\delta_{JV}^2 + (\frac{t_k}{c})^2) \tag{4}$$

Within the scope of this paper, the area of the jet vanes are assumed to be fixed. Linear supersonic theory may sometimes overestimate the lift and drag forces, and it has to verified with 3D computational fluid dynamics analysis and experiments. However, it is known that the accuracy of linearized theory is high when jet vanes are located at enough distance to with respect to each other and outside the nozzle [8]. Also, the dynamics between the jet vane deflection and total thrust deflection is taken as unity, because of the high the inflow rate [9]. The moments created by the deflection of the jet vanes are calculated by using the forces and the moment arms which are the distance between the nozzle exit diameter and jet vane center of pressure and the distance between the jet vane center of gravity and missile center of gravity.

3 Autopilot Model

There are studies on different algorithms for the control techniques applied at high angle of attack flight regimes. They can be mentioned as the robust control design for IRIS-T [10], the sliding mode controller [11], the adaptive control [12] and some nonlinear control strategies comparison with classical control [13]. As mentioned earlier, the aim of this study is to investigate the applicability of the classical control design techniques on that challenging VLSAM dynamics and identify the possible advantages and disadvantages. Other control techniques may then come into picture to defeat the discrepancies that the classical control design techniques cannot handle. In what follows, it will be demonstrated that autopilots designed with classical control techniques can meet the mission requirements of the VLSAM.

3.1 Linearized Systems

The state space matrix calculated with the Jacobian linearization of the nonlinear missile dynamics, at different instants of the vertical flight, can be written in the following form to express the control efficiency originating from the control surfaces separately:

$$\Delta\dot{\bar{x}} = \hat{A}\Delta\bar{x} + \hat{B}_A\bar{u}_A + \hat{B}\bar{u}_T \tag{5}$$

where $\bar{u}_A = [\delta_{Aa}\ \delta_{Ae}\ \delta_{Ar}]$ and $\bar{u}_T = [\delta_{Ta}\ \delta_{Te}\ \delta_{Tr}]$, and $\hat{B}_A$, $\hat{B}_T$ are the control matrices for the aerodynamic and thrust vector control. Here, $\Delta\bar{x}$ =$[\Delta U\ \alpha\ \beta\ p\ q\ r\ \phi\ \Delta\ \theta]$ is in general form. The eigenvalues and system matrices of vertical flight at different velocities and altitudes before and after burnout are given in [14].

To apply linear control techniques, nonlinear missile dynamics is decoupled into three simplified representations of the overall motion since there is no clear distiction between lateral and longitudinal dynamics because of its axi-symmetry property. The pitch plane state space equations can be formulated as:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}Z_\alpha - \dot{u}/U & 1+Z_q\\ M_\alpha & M_q\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}Z_{A_{\delta_e}} & Z_{T_{\delta_e}}\\ M_{A_{\delta_e}} & M_{T_{\delta_e}}\end{bmatrix}\begin{bmatrix}\delta_{Ae}\\ \delta_{Te}\end{bmatrix} \tag{6}$$

where $M_\alpha = \frac{Q_d Sd}{I_{yy}} C_{m_\alpha}$, $Z_\alpha = \frac{Q_d S}{mU} C_{z\alpha}$, $M_q = \frac{Q_d Sd^2}{2UI_{yy}} C_{m_q}$, $Z_q = \frac{Q_d Sd}{2U^2 m} C_{z_q}$, $M_{A_{\delta_e}} = \frac{Q_d Sd}{I_{yy}} C_{mA_{\delta_e}}$, $Z_{A_{\delta_e}} = \frac{Q_d S}{mU} C_{zA_{\delta_e}}$, $M_{T_{\delta_e}} = \frac{T_z l_x}{I_{yy}}$, and $Z_{T_{\delta_e}} = \frac{T_z}{mU}$ including aerodynamic derivatives e.g. C_{m_α}, C_{z_α} etc.

3.1.1 Enhanced Short Period Approximation

The addition of the term $\dot{u}/U$ as seen in Eq. 6 enhances short period approximation [15]. Conventional short period approximation assumes that the directional velocity component of the air vehicle is constant ($\dot{u} = 0$), however the missile under study, especially accelerates rapidly in the boost phase. Thus, the $\dot{u}/U$ becomes significant especially at the beginning of the vertical climb.

In Fig. 1, the normal acceleration time histories for a given longitudinal control input is presented to show the comparison between linearized and nonlinear models. Here, the linearized systems are scheduled with respect to time in the boost phase. As it is seen from the figure, the linear system behavior is drastically separated from the nonlinear system behaviour especially where $\dot{u}/U$ is high, i.e. at the beginning of the boost phase. Thus, as for the acceleration autopilot design, it is crucial to add $\dot{u}/U$ compensation to the short period approximation when the missile velocity is comparably low. Otherwise, the normal acceleration controller, designed without $\dot{u}/U$ compensation, may show either inadequate performance or unstable behavior [15].

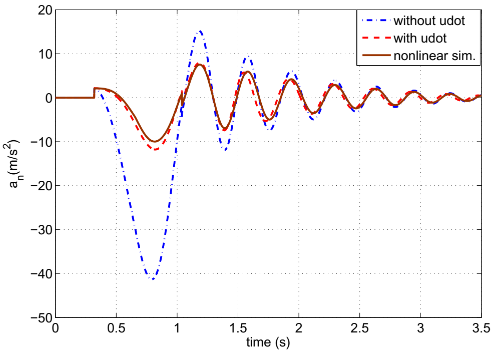


Fig. 1 Effect of missile axial acceleration in linearized missile acceleration dynamics

3.2 *Thrust Vector-Aerodynamic Control Effectiveness*

Control surfaces of such missiles are generally actuated using the same servo actuator. The challenging design problem is to set the ”hybrid control ratio” which is directly related with the desired control capability. This ratio has to be considered together with the mission requirements and the control effectiveness. A static control effectiveness analysis is conducted for that purpose. As expected, at low velocities aero control is less effective than TVC and becomes powerful as the speed increases. However, it looses control efficiency at high angle of attack values and also at relatively high altitudes. The control effectiveness analysis of the aerodynamic and thrust vectoring controls for the VLSAM at different altitudes is shown in Fig. 2. Here, the singular values of the $\hat{B}$ matrices are computed. TVC efficiency stays nearly constant. This is an expected result that only the total mass of the missile

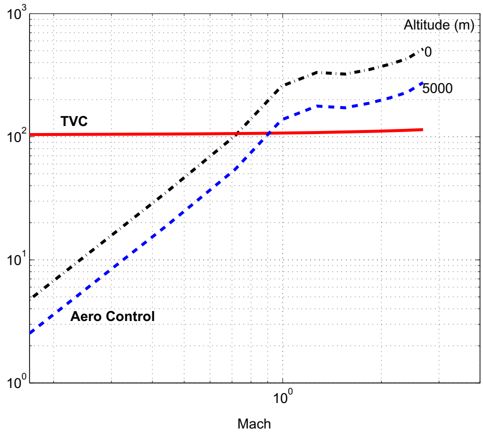
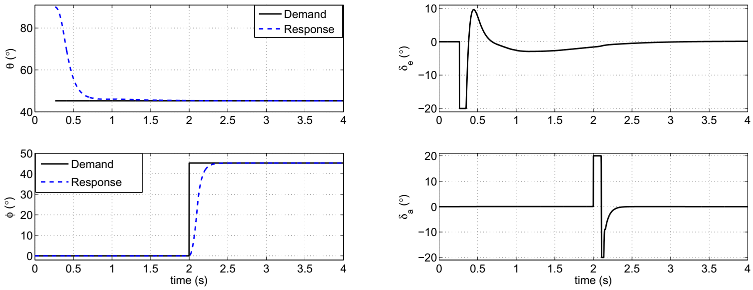


Fig. 2 Maximum singular value of B matrix

is decreasing while thrust level is nearly constant and it does not effect the TVC efficiency much. Further, the aerodynamic control becomes more effective than TVC after 0.7 or 0.9 Mach depending on the altitude. In order to have maximum maneuverability, the control allocation between the two control schemes is distributed as 1, but a detailed analysis and optimization study is done for this purpose [14].

3.3 Autopilot Simulations

In order to be used for the mid-course guidance, autopilot is designed to operate on roll and pitch angles [16]. In that phase, autopilot gains are time scheduled. The performance of autopilot for pitch/roll angle reference commands and deflection histories of aileron and elevator deflections are presented in Fig. 3(a) and Fig. 3(b). Time histories of critical parameters are also depicted in Fig. 4. Although large angle difference (45°) are demanded for both the roll and pitch attitudes, autopilot results are quite successful considering high variation of parameters such as α $(0-30^\circ)$, speed (0-3 Ma) and drastic increase in dynamic pressure.



(a) Angle autopilot response histories (b) Elevator and aileron deflections

Fig. 3 Angle autopilot nonlinear simulations

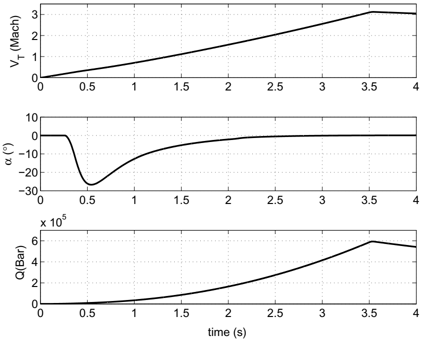
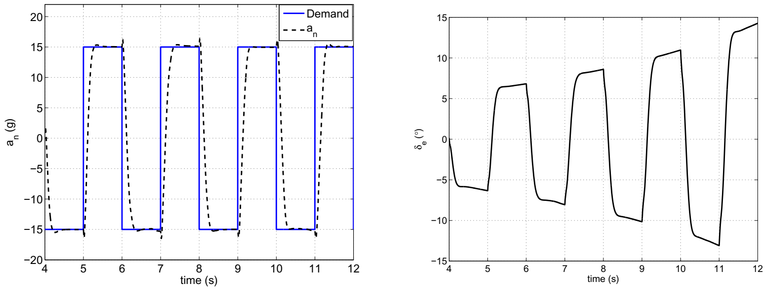


Fig. 4 Velocity, angle of attack and dynamic pressure histories

For the terminal guidance, acceleration autopilot is designed using the 3-loop acceleration autopilot scheme [13]. Autopilot gains are scheduled with respect to Mach. The performance of the autopilot to square command input of 15g is illustrated in Fig. 5(a) and the elevator deflection in Fig. 5(b). Since the speed is decreasing and demand stays constant in magnitude (15g), control commands increase to compensate this kinetic energy loss.



(a) Acceleration autopilot response

(b) Elevator deflection

Fig. 5 Tracking performance of acceleration autopilot

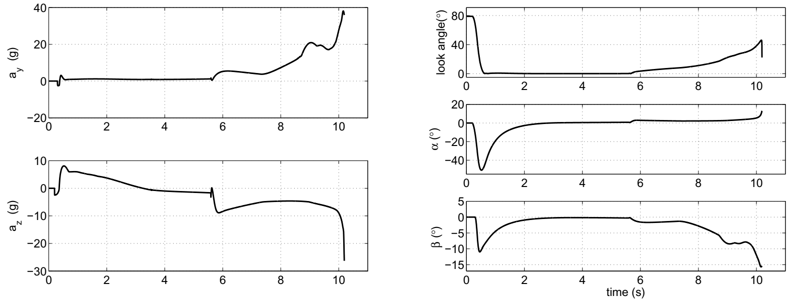
4 Guidance Algorithms

One of the main advantages of vertical launching is the engagement capability to the targets in all possible directions that brings the necessity to direct the missile to the plane of motion of target as soon as possible. Here, body pursuit guidance is a possible candidate, and simple to apply, to align the missile body axis to the line of sight. This procedure tries to minimize the look angle and effectively increases the target detection possibility. The aim of the body pursuit guidance algorithm is to produce reference body angle commands to be processed in the previously designed angle command controller. As for the terminal guidance after the boost phase, the conventional PNG guidance methodology is chosen for its proven performance and ease of application.

In this study, the design of the guidance algorithms are divided into two phases as the mid-course and the terminal guidance. The mid-course guidance starts in the launch phase and operates until the hand-over to the terminal guidance phase. The switching condition from mid-course to terminal guidance generally depends on the current state of the missile and the target, trajectory constraints and the target detection sensor, i.e. seeker, properties. A simple switching condition is defined and set to occur when the lock-on range is less than 5 km and the field of view is less than 3 degrees. Intercept condition is defined based on the achievable minimum value of the closing distance as 1 m.

4.1 Guidance and Control Simulations

The proposed flight control algorithms are implemented in a defense scenario on a closing target that has a velocity of 1 Mach and starts a pull-up maneuver with 7g when the target to missile range becomes less than 3 km. The initial position of the target with respect to the missile is $p_T = [10,\ 2, -2]$ km. In Fig. 6(a), the acceleration time histories are presented. Fig. 6(b) illustrates time histories of the look angle, angle of attack, sideslip angle of the missile. Look angle is decreasing from 80° to 0 during the mid-course guidance phase. Whenever the guidance algorithm switches to PNG, look angle starts increasing again (the behavior of the look angle at the end is not a numerical but dynamical trend). More maneuvers on different type of targets are simulated, see [14]. In all of the target types, flight control algorithms are succesfull. Besides, the flight envelope has to be clarified and the overall success of the flight control algorithms has to be examined.



(a) Acceleration histories (b) Look angle, angle of attack and sideslip

Fig. 6 Engagement simulations

5 Turnover Strategy

In the previous section, the turnover strategy used in mid-course guidance was skid-to-turn strategy in which missile does not roll and yaw/pitch channel commands are applied together to maneuver the missile towards the desired direction. Here, the proposed turn-over strategy is actually a mixed ascend that turnover maneuver composed of an initial roll followed by skid-to-turn (Fig. 7). That kind of maneuvering is also used in the mid-course guidance of air to air missiles [17] and [18]. [1] discusses a turn over strategy implemented with back to turn and roll maneuvers, however a direct comparison of skid-to-turn and skid-to-turn with initial roll maneuver, and their advantages and disadvantages are not explicitly conducted.

In this turn-over strategy, the missile has an initial roll maneuver and then starts turning towards the target. The objective of the initial roll maneuver is to align the pitch plane of motion of the body of the missile to the same vertical plane with the target. Thus, after the initial roll maneuver that aligns the missile's pitch plane of motion, a maneuver in that single plane is required to head on towards the target.

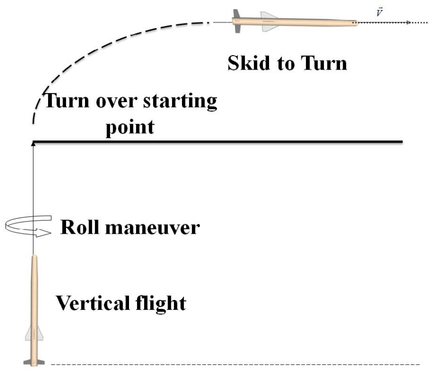


Fig. 7 Turnover with initial roll maneuver

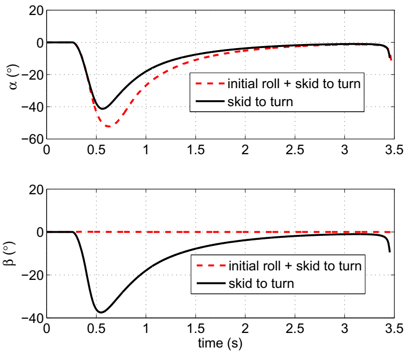


Fig. 8 Angle of attack and sideslip histories

This brings the advantage of flying with less control effort and reduces the side slip angle values. In order to analyze the differences between standard turn-over and the turnover with initial roll maneuver, they are implemented for different simulations for the same static target which is at $p_T = [1,\ 1,\ -0.2]$ km with respect to the missile. This necessitates a roll angle command of 45 degrees. Upon executing the roll maneuver, the engagement will become planar; so, only pitching control can be used to capture the target. Fig. 8 shows angle of attack and side slip angle time histories compared for the standard skid-to-turn and skid-to-turn with initial roll maneuvers. With the proposed maneuver, yaw maneuver is not required to head on the target. However, as a draw back, it necessitates higher angle of attack than the standard skid-to-turn maneuver as also seen in Fig. 8.

6 Discussion and Conclusion

The paper illustrates practical flight control algorithms for an advanced missile, the VLSAM. First, six degrees of freedom flight dynamics is briefly introduced. An enhancing linearization term for the short period approximation which has not directly mentioned in literature is proposed. Its drastic effect to similarity of scheduled linearized systems and nonlinear dynamics is shown. Two different autopilots, angle and acceleration, are designed using classical control techniques for the flight envelope of the VLSAM in co-operation with guidance algorithms requirements. Aerodynamic and thrust vector control capabilities are blended in 1 : 1 ratio to have more agility. The scheduled autopilots demonstrate satisfying performance in a highly nonlinear, rapidly parameter and time varying environment which is a promising start up for design process of industrial applications.

There are also advanced guidance techniques for agile missiles, but the analysis and numerical results show that a body pursuit guidance for midcourse guidance phase and proportional guidance for terminal guidance can be directly applied to the VLSAM. Moreover, a turnover maneuver, which is the initial roll maneuver, is also accomplished for midcourse guidance phase that decreases maneuver requirements in lateral direction. To sum up, classical approaches are still applicable for such an agile system. In order to start up a design, from industrial point of view, the flight control algorithms which are applied in this paper can be implemented easily and effectively. For further studies, maneuvers and autopilots are going to be optimized in order to maximize total energy and increase flight time. Regarding the optimization results, advanced flight control algorithms may be considered.

References

1. Lin, C.L., Lin, Y.P., Wang, T.L.: A fuzzy guidance law for vertical launh interceptors. Journal of Guidance, Control, and Dynamics 17(8), 914–923 (2009)
2. Tekin, R., Atesoglu, O., Leblebicioglu, K.: Modeling and vertical launch analysis of an aero- and thrust vectored surface to air missile. In: AIAA Atmospheric Flight Mechanics Conference (2010)
3. Tekin, R., Atesoglu, O., Leblebicioglu, K.: Vertical launch phase and effective turnover maneuver analysis of an air defence missile. In: 5. Defence Technologies, Ankara, Turkey (2010)
4. I-Sharkawy, A.: Experimental studies on thrust vector control in solid propellant rockets. In: 13th AIAA/SAE Propulsion Conference (1977)
5. Giragosian, P.A.: Theoretical and experimental aerodynamic correlation of jet vane control effectiveness. In: AIAA Atmospheric Flight Mechanics Conference (1981)
6. Harrisson, V., deChamplin, A., Kretschmer, D., Farinaccio, R., Stowe, R.A.: Force measurement evaluating erosion effects on jet vanes for a thrust vector control system. In: 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit (2003)
7. Orbekk, E.: Thrust vector model and solid rocket motor firing validations. In: 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit (2008)
8. Orbekk, E.: Novel tvc system utilizing guide vanes with jet flap's into a high efficiency compact nozzle. In: AIAA Atmospheric Flight Mechanics Conference (2005)

9. Sung, G., Hwang, Y.S.: Thrust-vector characteristics of jet vanes arranged in x-formation within a shroud. Journal of Propulsion and Power 20(3) (2004)
10. Buschek, H., McDaniel, M.: Design and flight test of a robust autopilot for iris-t air to air missile. Control Engineering Practice 11, 551–558 (2003)
11. Thukral, A., Innocenti, M.: A sliding mode missile pitch autopilot synthesis for high angle of attack maneuvering. IEEE Transactions on Control Systems Technology 6(3), 359–371 (1998)
12. Song, C.: Adaptive Control of Linear Time Varying Systems with Applications to Missile Autopilots. PhD thesis, University of California (1989)
13. Devaud, E., Siguerdidjane, H., Font, S.: Some control strategies for a high angle of attack missile autopilot. Control Engineering Practice 8, 885–892 (2000)
14. Tekin, R.: Design, modeling, guidance and control of a vertical launch surface to air missile. Master's thesis, Middle East Technical University (2010)
15. Erer, K.S., Tekin, R.: An investigation on the effects of forward speed change on lateral missile dynamics. In: National Automatic Control Conference (2010)
16. Solis, R.: An analysis of the vertical launch phase of a missile concept. In: AIAA Aerospace Sciences Meeting (1983)
17. Menon, P.K., Ohlmeyer, E.: Integrated design of an agile missile guidance and control systems. In: Proceedings of the 7th Mediterranean Conference on Control and Automation (1999)
18. Wise, K.A., Broy, D.J.: Agile missile dynamics and control. Journal of Guidance, Control, and Dynamics 21(3), 441–449 (1998)

LFT Model Generation via ℓ_1-Regularized Least Squares

Harald Pfifer and Simon Hecker

Abstract. The paper presents a general approach to approximate a nonlinear system by a linear fractional representation (LFR), which is suitable for LFT-based robust stability analysis and control design. In a first step, the nonlinear system will be transformed into a quasi linear parameter varying (LPV) system. In the second step, the nonlinear dependencies in the quasi-LPV, which are not rational in the parameters, are approximated using polynomial fitting based on ℓ_1-regularized least squares. Using this approach an almost Pareto front between the accuracy and complexity of the resulting LFR can be efficiently obtained. The effectiveness of the proposed method is demonstrated by applying it to a nonlinear missile model of industrial complexity.

1 Introduction

Linear fractional transformations (LFTs) can be considered a standard form for many modern robust control methods. In literature, a plethora of algorithms based on LFTs exist for analysis or synthesis, see e.g. [1]. In general, control problems are dealing with nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x,p,u) \\ y &= g(x,p,u), \end{aligned} \tag{1}$$

Harald Pfifer
Institute of System Dynamics and Control,
German Aerospace Center - DLR,
Muenchner Str. 20, 82234 Wessling, Germany
e-mail: harald.pfifer@dlr.de

Simon Hecker
University of Applied Sciences Munich,
Lothstr. 64, 80335 Muenchen, Germany
e-mail: hecker@ee.hm.edu

where $x \in X \subset \mathbb{R}^{n_s}$ is the state vector, $y \in \mathbb{R}^{n_y}$ the output vector and $u \in \mathbb{R}^{n_u}$ the input vector. In addition the system can depend on a parameter vector $p \in \Pi$. In order to apply modern robust control methods on them, an efficient approach to approximate (1) by a linear fractional representation (LFR) needs to be available.

As an intermediate step on the way of obtaining an LFR of (1), the system is first transformed into an LPV system of the form

$$\begin{aligned} \dot{x} &= A(\delta)x + B(\delta)u \\ y &= C(\delta)x + D(\delta)u. \end{aligned} \tag{2}$$

In (2), δ can not only consist of the parameter vector p but also includes state-dependent nonlinearities, i.e. $\delta \in (X \times \Pi) \subset \mathbb{R}^{n_\delta}$, see [2]. In the latter case, the system is called quasi-LPV. Various techniques have been proposed in literature to perform the transformation of (1) into an quasi-LPV system (see for example [3, 4]).

If the quasi-LPV system (2) depends only rationally on δ, the transformation into a linear fractional representation (LFR) of the form

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_1 x + D_{11} w + D_{12} u \\ y &= C_2 x + D_{21} w + D_{22} u \\ w &= \Delta(\delta) z \\ \Delta &= \operatorname{diag}(\delta_1 I_{s_1}, \ldots, \delta_{n_\delta} I_{s_{n_\delta}}) \end{aligned} \tag{3}$$

is straightforward.

Many sophisticated methods have been proposed in literature to obtain low order LFRs of a given LPV system, see [5] and the references therein. Usually, three steps are applied in the transformation process. First, a symbolic preprocessing of the LPV model is performed. Second, the actual transformation is conducted via object oriented LFT realization. Finally, numerical order reduction can be utilized to further reduce the order of the resulting LFR.

In several cases one may directly derive an analytic quasi-LPV (2) suitable for transforming into an LFR from a nonlinear system (1) via symbolic calculations. However, especially in aeronautical applications the models usually include highly nonlinear functions (neural networks, tables) or may only be given for a discrete set of conditions (linear aeroelastic models). In such cases the quasi-LPV model obtained via function substitution cannot be directly transformed into an LFR. The highly nonlinear functions or the discrete set of conditions have to be approximated by rational functions first.

It is largely an open question, how a function approximation is obtained, which is suitable for transformation of (2) into an LFR. Note that the minimal achievable LFR order depends mainly on the complexity (order of rational or polynomial approximations) and the structure of (2). In [6], we proposed a method to generate optimal LFT models achieving a good accuracy while keeping the complexity low. It involves directly minimizing a weighted sum of the LFR order and some error metric via a complex non-convex optimization. While this procedure provides good results as shown in [7] and [8], it is rather cumbersome and time consuming. The major contribution of this paper is a convex relaxation of the optimization of [6], which allows to efficiently obtain an (almost) Pareto front representing the compromise between the LFR order and accuracy.

In Section 2, the general problem formulation is stated. In Section 3, an overview over ℓ_1-regularized least squares and its application to polynomial fitting will be presented. Then the LFR generation problem is reformulated in the ℓ_1-regularized least squares framework in Section 4. Finally, in Section 5 the proposed algorithms are applied to a nonlinear missile model of industrial complexity with a nonlinear dynamic inversion based controller. It is shown that the approach allows to transform the highly nonlinear system into an LFR of sufficient accuracy, which still possesses a complexity suitable for performing LFT based stability analysis.

2 Problem Statement

The starting point of the LFT model generation is a quasi-LPV model (2), which can for instance be obtained via a function substitution technique as introduced by [3]. It is assumed that (2) does not depend rationally on the parameter vector δ. More precisely, there are $\{s_i(\delta)\}_1^p$ elements in (2) which need to be approximated by a rational/polynomial function, in order to transform (2) into an LFR. In a typical aerospace application the set $\{s_i(\delta)\}_1^p$ would contain for example the aerodynamic forces and moments coefficients. In the present work due to its simplicity only polynomial functions will be considered to approximate the original functions $s_i(\delta)$.

The first step is to generate a grid of values $s_{i,k}$ for each function s_i at a set of pre-specified parameter values. The value $s_{i,k}$ represents the i^{th} function evaluated at the k^{th} point and δ_k is the corresponding parameter vector. For each index value k an LTI system with transfer function $G_k = C_k(sI - A_k)^{-1}B_k + D_k$ can be built by evaluating the quasi-LPV model (2) at $\delta = \delta_k$.

The goal is now to calculate a polynomial approximation of the elements s_i, such that (2) can be transformed into an LFR of low complexity. It should, however, still represent the original nonlinear system (1) adequately. The problem can be conceptually described by

$$\min_{G_{lfr} \in \mathscr{S}_{lfr}} d(G_{lfr}, \{G_k\}_1^m) + w\, c(G_{lfr}), \tag{4}$$

where $d(.,.)$ is the notation of distance or model error between the approximate model G_{lfr} which is restricted to the class of LFT-based LPV models $\mathscr{S}_{lfr}$, see (3), and the grid point LPV model $\{G_k\}_1^m$. In addition, $c(.)$ describes the complexity of the resulting LFR and w is a weighting factor to balance complexity and accuracy.

As a measurement of the LFR's quality the ν-gap metric, as specified by [9], between the $\{G_k\}_1^m$ and the LFR evaluated at $\{\delta_k\}_1^m$ is applied. The ν-gap metric can take values between zero and one with zero meaning that two plants match closely and one that they are far apart. In general, any system norm can be used, e.g. $\mathscr{H}_2$-norm. The ν-gap metric has a decisive advantage over other system norms, though, as an error measurement for LPV model generation: It is also defined for unstable systems. Since in many practical cases the plant may be at least partially unstable in the admissible parameter set, special care has to be taken when choosing other system norms.

The complexity of the resulting LFR is estimated by the lower bound on its order as defined in [5], which is computationally faster than using the actual achievable order. For a given linear parametric model $S(\delta)$ with $\delta \in \mathbb{R}^{n_\delta}$ the lower bound can be calculated as follows: Substitute all but one parameter δ_i with random values and compute a minimal order, one parametric LFR with order m_i. Note, that for single parametric systems one can always calculate a minimal order LFR. Repeat this procedure for all parameters. Finally, the lower bound is given by $m_{LB} = \sum_{i=1}^{n_\delta} m_i$.

3 Polynomial Approximation by ℓ_1-Regularized Least Squares

The algorithm for finding polynomial approximations of the single matrix elements is based on a regularized least squares fitting. In the following $x_{j,k}$ denotes the numerical values of the j^{th} parameter at the k^{th} grid point, y is a vector including the m grid point values of an element s_i and b is a vector including the polynomial coefficients. In a first step, a matrix X will be built, which considers all possible bases for a multivariable polynomial of a given order evaluated at the m grid points.

In function approximation, it is often desired to choose from a set of potential bases one subset, which offers the best approximation of all subsets of the same cardinality, see [10]. The aim is to find a solution to the least squares problem with a sparse coefficient vector b, i.e with a small cardinality card(b), which is defined as the number of nonzero elements in the vector b. Considering a coefficient vector $b \in \mathbb{R}^n$ and $k < n$, it can be described as

$$\min_b \|Xb - y\|_2^2 \,, \text{ s.t. } \operatorname{card}(b) \leq k. \tag{5}$$

As shown in [10] this is a hard combinatorial problem. However, there exists a good heuristics to approximate (5), which is called ℓ_1-regularized least squares. In (6) $\|b\|_1 = \sum_{i=1}^n |b_i|$ is the ℓ_1-norm and λ the regularization parameter.

$$\min_b \|Xb - y\|_2^2 + \lambda \|b\|_1 \tag{6}$$

For a given λ this problem is actually convex and can be easily solved. By performing a sweep over λ a Pareto front is obtained, which presents the trade-off between card(b) and the residual $\|Xb - y\|_2$. Various techniques exist in literature to solve the convex problem (6), e.g. specialized interior-point methods as proposed in [11].

4 Procedure for the Generation of LFT Models

A weighted form of (6) can be used to simultaneously approximate all $\{s_i(\delta)\}_1^p$ by polynomials.

$$\min_b \sum_{i=1}^{p} w_{1,i} \|X\beta_i - y_i\|_2^2 + \lambda w_2^T |b|, \tag{7}$$

where b is a vector containing the polynomial coefficients over all elements that should be approximated. The vectors $\{\beta_i\}_1^p$ consist of those coefficients which are used in the approximation of the i^{th} element. The weightings w_1 and w_2 will be used later on to bring (7) closer to the original problem of minimizing the LFR order and an error metric between the LFR and the quasi-LPV.

Since all elements and especially all polynomial coefficients over all elements are considered in the cost function, scaling them is important. In the presented approach, the data is normalized so that each column of X and y has unit length and zero mean [12].

$$\begin{aligned} &\sum_{i=1}^{m} y_i = 0 \quad &&\sum_{i=1}^{m} X_{i,j} = 0 \\ &\sum_{i=1}^{m} y_i^2 = 1 \quad &&\sum_{i=1}^{m} X_{i,j}^2 = 1 \;\text{ for }\; j = 1,\ldots,n \end{aligned} \tag{8}$$

Note that making use of orthogonal polynomials as e.g. Chebyshev polynomials and the standardization (8) results in X being orthonormal, i.e. $X^T X = I$. This simplifies solving the ℓ_1-regularized problem (7).

The regularization parameter λ in (7) is employed as a weighting between the accuracy and the complexity of the polynomial approximation as described in Section 3. The Pareto front between card(b) and the polynomial fit serves as an approximation of the trade-off between the LFR order and the accuracy of the LFR model.

The starting value of λ is is chosen in accordance with the following proposition.

Proposition 1. *Setting λ in (7) to*

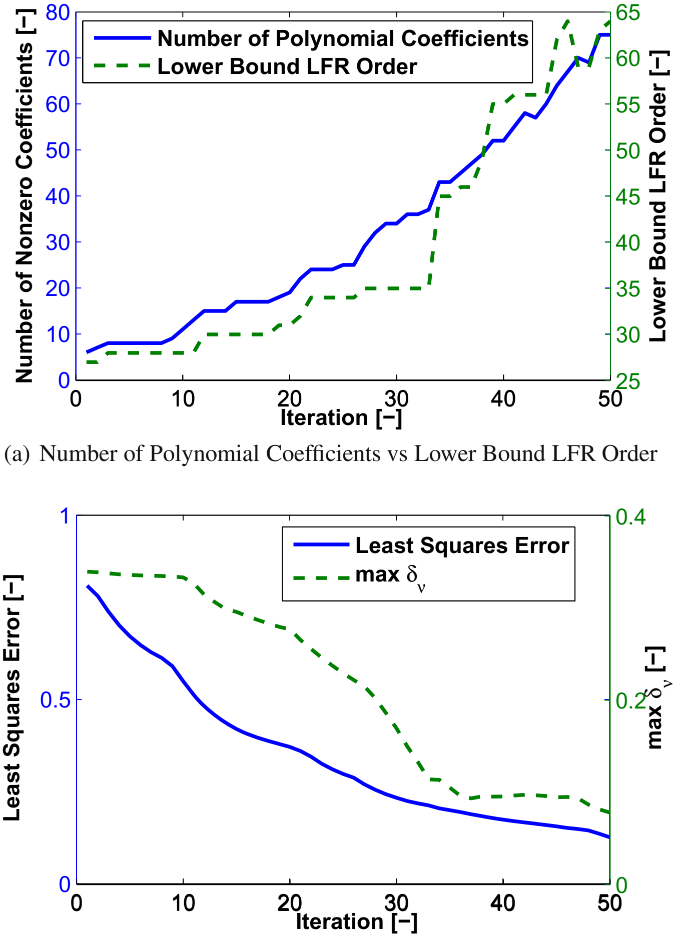
$$\lambda_0 = 2 \max_i \left(w_{1,i} \max_j \left(\frac{|(X^T y_i)_j|}{w_{2,j}} \right) \right) \tag{9}$$

yields a constant approximation, i.e. $b = 0$.

For the proof the reader is referred to [11]. Incrementally decreasing λ will result in steadily better fits with higher card(b).

An advantage of this approach is that a lot of computations can be done upfront and reused at each iteration. Additionally, the solution of the last iteration is used as the starting point of the new iteration. Hence, performing a sweep over λ is computationally comparably cheap.

To demonstrate the validity of the convex approximation some brief results of the polynomial approximation used in the missile model with nonlinear dynamic



(a) Number of Polynomial Coefficients vs Lower Bound LFR Order

(b) Least Squares Residual vs Maximum ν-Gap Metric

Fig. 1 Results of ℓ_1-Regularized Least Squares Problem for LFR Model Generation

inversion controller are presented (see Section 5 for more details). The elements that require approximation in the missile model are the aerodynamic force and moment coefficients.

In Fig. 1, the difference between the cost function of the convex problem (7) and the lower bound of the LFR order as well as the maximum ν-gap metric $\delta_{\nu,max}$ is presented. As shown in the upper figure the number of polynomial coefficients and the lower bound LFR order m_{LB} follow a similar trend. However, it can be seen that it is possible to increase the cardinality of the coefficient vector b without increasing m_{LB}.

A likewise statement can be made about the relation between the sum of the residuals $\sum_i \|Xb_i - y_i\|_2^2$ and the maximum ν-gap metric between the LFR and the grid-point LPV model. Minimizing the residual in general seems to also lower $\delta_{\nu,max}$. Still, it has to be kept in mind that this is only a heuristics and no direct relation between the residual and some system metric can be established.

4.1 Weighting of the Elements

The quality of the LPV approximation is only accounted for in the weighted sum of the polynomial approximation errors in a least squares sense. Since not all elements s_i have the same significance for the LPV model, equally weighting them would not reflect the major aim to find an LFR of good accuracy and low order. Hence, for each element s_i a so-called influence coefficient IC_i is determined.

An element s_i has a low influence coefficient if its variation among the set of grid point models does not significantly influence the transfer matrix of the frozen models in terms of a specified error metric. For each s_i a set of transfer matrices $\{G_{k_i}\}_1^m$ is generated, which is equal to the set $\{G_k\}_1^m$ except that for s_i the mean value of the m grid point values y_i is chosen. Finally, the influence coefficient IC_i of s_i is defined as

$$IC_i = \max_k(\delta_\nu(G_k, G_{k_i})),\ k = 1, \ldots, m, \tag{10}$$

where δ_ν denotes the ν-gap metric between G_k and G_{k_i}.

The IC can be directly used as weighting w_1 in the convex approach. Hence, the algorithm is biased towards minimizing the approximation errors of elements with a high IC.

In order to show the advantage of using the influence coefficient as weighting w_1, a comparison between a weighted sum and weighting each elements equally is made. The example uses again the data from the missile model. In Fig. 2 the maximum error in terms of the ν-gap metric $\delta_{\nu,max}$ is shown over the lower bound on the LFR order for a sweep over the regularization parameter λ. The case, where each approximation error is weighted equally, is depicted by the circles. The crosses represent the results of the approximation with using the influence coefficients of the elements as weightings w_1 in (7). As can be seen in the figure, the weighted ℓ_1-regularized problem yields much better results for most λ in comparison to the unweighted problem. Note, that in both cases not each point is actually Pareto optimal in terms of ν-gap and LFR order.

In addition to its usage to weight the elements, the influence coefficients are also used to determine which elements can be considered constant in the proposed approach. Is the influence coefficient of an element below a specified threshold, the respective element will be approximated by its constant mean value over all grid points.

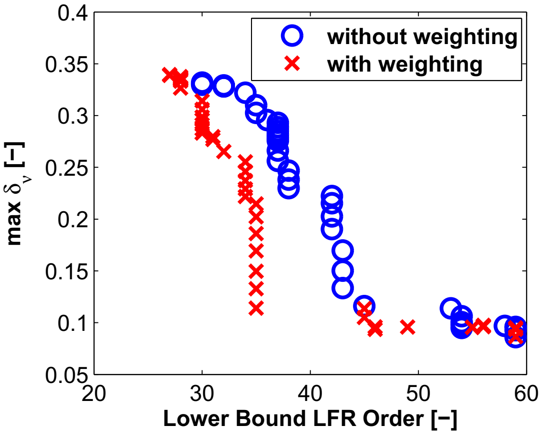


Fig. 2 Effect of Using Influence Coefficients as Weightings in the Optimization

4.2 Weighting of the Polynomial Coefficients

Instead of minimizing the lower bound LFR order a weighted ℓ_1-norm over all possible polynomial coefficients is used. The weighting w_2 shall be chosen such as to penalize polynomial bases which would lead to higher order LFRs. A good heuristics for choosing w_2 is to set $w_{2,j}$ to the degree of the respective polynomial basis $g_j(x)$. For instance considering the monomial basis $g_j(x) = x_1^2 x_2^2$, the corresponding weighting factor $w_{2,j}$ would be four. This also coincides with the minimum achievable LFR order of g_j.

5 Example: Missile Model

The example is based on a nonlinear model of a modern air defense missile. The missile is in a cruciform configuration with four fins at the tail. It is axis symmetric with a slender body. A controller based on nonlinear dynamic inversion has been designed for the missile model within [13]. The aim of this work is to obtain an LFR of the closed loop missile, which is suitable for modern LFT based robust stability analysis.

The mathematical model has six states, namely the velocities in y- and z-direction v and w respectively, the roll rate p, pitch rate q, yaw rate r and the bank angle around the velocity vector Φ_V. The inputs are the deflections angles in aileron ξ, elevator η and rudder ξ. As outputs the accelerations in y- and z-direction a_y and a_z, Φ_V and the angular velocities $\Omega = [p,q,r]^T$ are available. The parameter vector δ consists of the Mach number Ma, the angle of attack α and the side slip angle β.

In the following, the general differential equations of momentum and angular momentum are given in the body-fixed coordinate system. F and M represent the external forces and moments respectively acting on the missile and I_T the inertial tensor.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m}\sum F - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I_T^{-1} \left(\sum M - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(I_T \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) \tag{12}$$

Note that the acceleration in x-direction $\dot{u}$ has been neglected in the quasi-LPV model, as the control has no influence on them.

In the following equations the C_i are the aerodynamic coefficients which are nonlinear functions of δ. In addition, ρ is the air density, $V = Ma\, a$ is the absolute velocity with a being the speed of sound. The other parameters are the reference area S_{ref}, reference length l_{ref}, mass m and moments of inertia I_{xx}, I_{yy} and I_{zz}. For the sake of brevity the auxiliary variables $K_1 = \rho V S_{ref}/(2m)$, $K_2 = \rho V S_{ref} l_{ref}/2$ and $K_3 = \rho V S_{ref} l_{ref}^2/4$ are introduced.

In the quasi-LPV model the forces and moments will be described by their respective dimensionless coefficients, which is common practice in aerospace. These aerodynamic coefficients are nonlinear functions of the parameter vector δ and are only available as discrete table data.

$$\begin{aligned} \begin{bmatrix} F_Y \\ F_Z \end{bmatrix} &= mK_1V \begin{bmatrix} C_{Y0}(\delta) + C_{Y\zeta}(\delta)\zeta \\ C_{Z0}(\delta) + C_{Z\eta}(\delta)\eta \end{bmatrix} \\ \begin{bmatrix} L \\ M \\ N \end{bmatrix} &= K_2V \begin{bmatrix} C_{L0}(\delta) + C_{L\xi}(\delta)\xi \\ C_{M0}(\delta) + C_{M\eta}(\delta)\eta \\ C_{N0}(\delta) + C_{N\zeta}(\delta)\zeta \end{bmatrix} + K_3 \begin{bmatrix} C_{Lp}(\delta)p \\ C_{Mq}(\delta)q \\ C_{Nr}(\delta)r \end{bmatrix} \end{aligned} \tag{13}$$

The state equations are given in (16), where $A_1(\delta)$ is obtained via a function substitution of the aerodynamic coefficients. The following substitutions have been used:

$$\bar{C}_{i0} = \begin{cases} 0, & \text{if } \beta = 0 \\ C_{i0}/\sin\beta, & \text{otherwise} \end{cases} \quad \text{for } i = Y, L, N \tag{14}$$

$$\bar{C}_{i0} = \begin{cases} 0, & \text{if } \alpha = 0 \\ C_{i0}/(\sin\alpha\cos\beta), & \text{otherwise} \end{cases} \quad \text{for } i = Z, M \tag{15}$$

$$\begin{bmatrix} \dot{v} \\ \dot{w} \\ \dot{\Phi}_V \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = A(\delta) \begin{bmatrix} v \\ w \\ \Phi_V \\ p \\ q \\ r \end{bmatrix} + B(\delta) \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \tag{16}$$

$$A(\delta) = \begin{bmatrix} K_1\bar{C}_{Y0} & 0 & 0 & V\sin\alpha\cos\beta & 0 & -V\cos\alpha\cos\beta \\ 0 & K_1\bar{C}_{Z0} & 0 & -V\sin\beta & V\cos\alpha\cos\beta & 0 \\ 0 & 0 & 0 & \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ K_2/I_{xx}\bar{C}_{L0} & 0 & 0 & K_3/I_{xx}C_{Lp} & 0 & 0 \\ 0 & K_2/I_{yy}\bar{C}_{M0} & 0 & 0 & K_3/I_{yy}C_{Mq} & 0 \\ K_2/I_{zz}\bar{C}_{N0} & 0 & 0 & 0 & 0 & K_3/I_{zz}C_{Nr} \end{bmatrix} \tag{17}$$

$$B(\delta) = \begin{bmatrix} 0 & 0 & K_1VC_{Y\zeta} \\ 0 & K_1VC_{Z\eta} & 0 \\ 0 & 0 & 0 \\ K_2V/I_{xx}C_{L\xi} & 0 & 0 \\ 0 & K_2V/I_{yy}C_{M\eta} & 0 \\ 0 & 0 & K_2V/I_{zz}C_{N\zeta} \end{bmatrix} \tag{18}$$

Note that as a requirement for the application of the inversion based control method, the system has to be minimum phase. This is not the case when considering the accelerations at the center of gravity. As a remedy, in [13], it has been proposed to use accelerations a_y and a_z at a virtual point P instead. If the point P is sufficiently far in front of the center of gravity the system is minimum phase. The output equations for the accelerations at P can be written as

$$a_y = \frac{\bar{q}S_{ref}}{m}C_Y + \dot{r}x_{gp} \qquad a_z = \frac{\bar{q}S_{ref}}{m}C_Z - \dot{q}x_{gp} \tag{19}$$

with x_{gp} being the distance between P and the center of gravity. Using the same function substitution (19) can be written in a suitable form to fit into the quasi-LPV framework. The equations for the remaining outputs, namely Φ_V and Ω, are easily obtained, as both Φ_V and Ω are states of the system.

In addition to the quasi-LPV parameters defined above also uncertainties in the aerodynamic data are considered in the model. For the moment and control surface coefficients (δC_L, δC_M, δC_N and δC_{ctrl}) these are ±20 percent and for the force coefficients (δC_Y and δC_Z) ±5 percent.

The controller is a standard nonlinear dynamic inversion based one. It is separated into three parts: the inversion of the rotational dynamics Ω, the inversion of the outer dynamics a_z, a_y and Φ_V and a reference model, which is only used in the feed forward path. A classical PI-controller has been developed for the inverted plant. A detailed description of the controller is found in [13].

5.1 Generation of the LFR Model

In order to transform (16) into an LFR the aerodynamic coefficients need to be approximated by polynomials. The results of this ℓ_1-regularized least squares fitting are shown in Fig. 3. An approximate Pareto front between the lower bound of the LFR order and the approximation error in terms of the ν-gap metric for the plant can be seen. Both the maximum error (dashed red line) and the mean error (solid blue

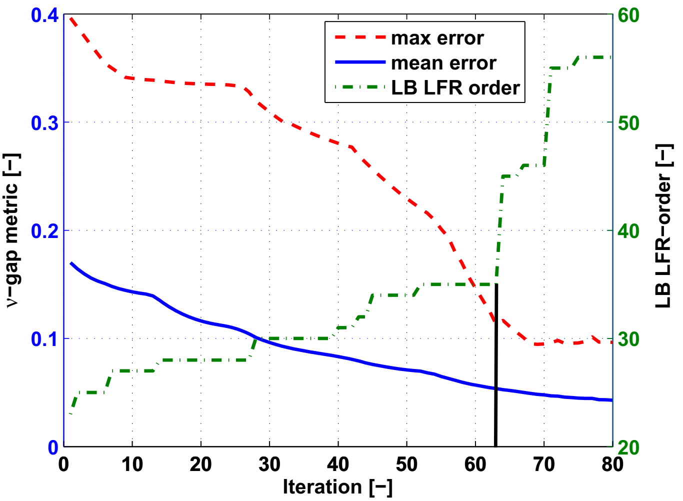


Fig. 3 Results of the ℓ_1-regularized Least Squares Fitting

line) over all grid points are provided. The black vertical line designates the chosen iteration for the LFR generation. This iteration has a maximum error $\delta_{\nu,max} = 0.11$, a mean error $\delta_{\nu,mean} = 0.054$ and a lower bound of LFR order of $m_{LB} = 35$.

The nonlinear dynamic inversion controller is already in LPV form. The same polynomial approximations for the aerodynamic coefficients can be used for it. The trigonometric functions in (17) can simply be approximated by a Taylor series expansion and truncation after a sufficient high order.

At this point, the closed loop of the missile benchmark is available as a symbolic description of an LPV model in its general form (2). It only depends rationally on the parameter vector δ. This is a requirement for transforming the system into an LFR. By employing the sophisticated techniques of [14] the resulting closed loop LFR has a dimension of 65, with the Δ-Block having the following structure:

$$\Delta = \mathrm{diag}(MaI_{25\times 25}, \alpha I_{19\times 19}, \beta I_{11\times 11}, \delta C_y, \delta C_z, \delta C_l, \delta C_m, \delta C_n, \delta C_{ctrl} I_{5\times 5}) \quad (20)$$

5.2 Model Assessment

In order to show that the closed loop LFT system still closely matches the original nonlinear system, a Monte-Carlo simulation is conducted. To estimate the error, which has been introduced due to the various approximation steps, the LFT system is run in a parallel setup with the fully nonlinear model. The error is then measured in terms of the maximum of a relative $\mathscr{L}_2$-norm over a finite time horizon, which is defined as

$$error = \max_i \frac{\left(\int_{t0}^{t1} (y_{nls,i} - y_{lfr,i})^2 dt\right)^{0.5}}{\left(\int_{t0}^{t1} y_{nls,i}^2 dt\right)^{0.5}} \tag{21}$$

with $y_{nls} = [\Phi_{V,nls}, a_{z,nls}, a_{y,nls}]^T$ and $y_{lfr} = [\Phi_{V,lfr}, a_{z,lfr}, a_{y,lfr}]^T$.

Simultaneous sinusoidal sweeps in all three command channels are applied as input signals for the nonlinear simulation. The amplitude of the signals are 10°, 20m/s^2 and 10m/s^2 for the Φ_V-, a_z- and a_y-channel respectively. The frequencies from the sinusoidal sweeps range from 0.1 Hz to 10 Hz.

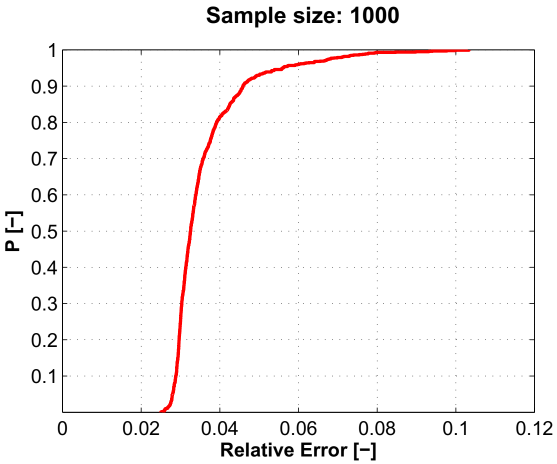


Fig. 4 Statistical Results of the Monte-Carlo Simulation

The only parameters considered in the Monte-Carlo simulation are Ma, α and β. All parameters corresponding to model uncertainty are set to their respective maximum values. The results of the Monte-Carlo run with 1000 samples are shown in Fig. 4 in form of the cumulative distribution function (CDF). The CDF gives the percentage of simulation runs which are less than a specified error. The samples are uniformly distributed over the considered flight envelope spanned by $Ma = [0.9, 4.4]$, $\alpha = [0, 25]^\circ$ and $\beta = [0, 10]^\circ$. In the CDF, it is seen that in 90 percent of the cases the error is less than 5 percent. The worst case found in the Monte-Carlo run is 10.2 percent.

The time history of the worst case found in the Monte-Carlo simulation is shown in Fig. 5, i.e. the error is 10.2 percent. It can be seen in the figure that the LFR model still matches the original nonlinear model well.

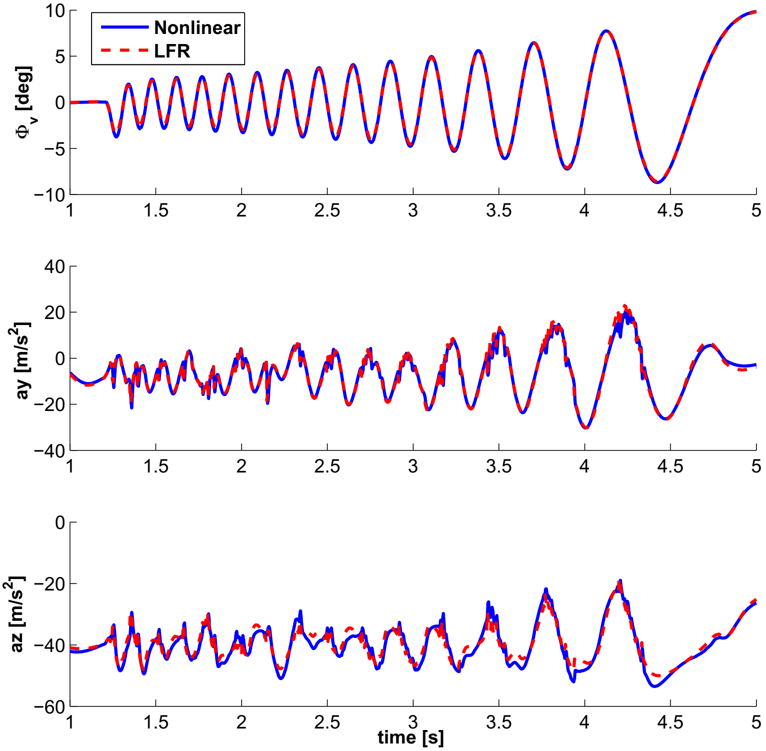


Fig. 5 Comparison Simulation between Nonlinear Model and LFR

6 Conclusion

A very general algorithm for generating LFT models has been developed, which can be applied to arbitrary nonlinear systems, as long as the system behavior can be accurately described/approximated with polynomial or rational parametric state-space systems. In order to efficiently generate LFT models, a convex relaxation has been proposed. It is very time efficient and can compute an almost Pareto front between the LFR order and accuracy.

In the present work, this algorithm has been successfully applied to an industrial benchmark problem. LFRs of high accuracy and reasonable order could be generated for a highly nonlinear missile model. The quality of the LFRs has been assessed using Monte-Carlo simulations.

In the future, it is contemplated incorporating the approximation error made during the polynomial fitting as a dynamic unstructured uncertainty. Methods as the ones described in [15] can likely be adopted for this purpose. Such mixed parametric dynamic uncertainty models might be better suited for controller synthesis purpose.

References

1. Zhou, K., Doyle, J.C.: Essentials of Robust Control. Prentice Hall, New Jersey (1998)
2. Leith, D.J., Leithead, W.E.: Survey of gain-scheduling analysis and design. International Journal of Control 73(11), 1001–1025 (2000)
3. Tan, W.: Applications of linear parameter varying control theory. Master's thesis, University of California at Berkeley (1997)
4. Marcos, A., Balas, G.: Development of linear-parameter-varying models for aircraft. Journal of Guidance, Control and Dynamics 27, 218–228 (2004)
5. Hecker, S.: Generation of low order LFT Representations for Robust Control Applications. VDI Verlag (2007)
6. Pfifer, H., Hecker, S.: Generation of optimal linear parametric models for LFT-based robust stability analysis and control design. In: Proceedings of IEEE Conference on Decision and Control (2008)
7. Pfifer, H., Hecker, S., Michalka, G.: Lft-based stability analysis of generic guided missile. In: Proceedings of the AIAA Guidance, Navigation and Control Conference (2009)
8. Hecker, S., Pfifer, H.: Generation of LPV Models and LFRs for a Nonlinear Aircraft Model. In: Varga, A., Hansson, A., Puyou, G. (eds.) Optimization Based Clearance of Flight Control Laws. LNCIS, vol. 416, pp. 39–57. Springer, Heidelberg (2012)
9. Vinnicombe, G.: Frequency domain uncertainty and the graph topology. IEEE Transactions on Automatic Control 38 (1993)
10. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
11. Kim, S., Koh, K., Lustig, M., Boyd, S., Gorinevsky, G.: An interior-point method for large scale ℓ_1-regularized least squares. IEEE Journal on Selected Topics in Signal Processing (2007)
12. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics 32 (2004)
13. Peter, F., Leitao, M., Holzapfel, F.: Adaptive augmentation of a new baseline control architecture for tail-controlled missiles using a nonlinear reference model. In: Proceedings of the AIAA Guidance, Navigation and Control Conference (2012)
14. Hecker, S., Varga, A., Magni, J.: Enhanced LFR-toolbox for Matlab. Aerospace Science and Technology 9 (2005)
15. Hindi, H., Seong, C.-Y., Boyd, S.: Computing optimal uncertainty models from frequency domain data. In: Proceedings of Conference on Decision and Control (2002)

An Impulsive Input Approach to Short Time Convergent Control for Linear Systems

Martin Weiss and Yuri Shtessel

Abstract. The paper considers the problem of bringing the state of a controllable linear system to the origin in a very short time. It takes the approach of considering an "ideal" control input consisting of a linear combination of the Dirac delta function and its derivatives that realizes this goal instantaneously. Three schemes are introduced to approximate the impulsive input with physically realizable functions: a smooth approximation with compact support, a Gaussian function approximation and a step approximation. It is shown using a numerical example that all approximations work reasonably well, with the Gaussian approximation providing slightly worse results. It is also shown that a direct approach to obtain a state nulling input by solving an integral equation runs quicker into numerical problems than the impulsive input approach as the convergence time decreases. Finally, an application to an orbital rendez-vous problem is presented.

1 Introduction and Motivation

The interest in impulsive control theory has steadily increased over the past few years with many new books and articles being added to an already impressive list. Without any ambition to be exhaustive we may cite here books like [1, 2, 3, 4] and numerous journal and conference contributions such as [5, 6, 7]. The idea of using the delta distribution and its derivatives in control synthesis is not new. This approach seems to be first considered in [8]. Another work that takes a similar approach is [9]. More recent publications such as [7] have even extended the problem to the case of linear descriptor systems. In the paper [5] a dynamic programming

Martin Weiss
TNO Organization, POB 45, 2280 AA Rijswijk, Netherlands
e-mail: martin.weiss@tno.nl

Yuri Shtessel
University of Alabama in Huntsville, Huntsville, Alabama 35899, USA
e-mail: shtessel@eng.uah.edu

approach is proposed for optimal impulsive control laws that turn out to be linear combinations of delta and delta derivatives. Another recent contribution [6] uses delta and delta derivative state feedback for the adaptive stabilization of a second order nonlinear system.

There are numerous practical situations in which impulsive control is not just an option, but the only solution to achieve the required performance. In general, this is the case in all situations that large deviations from equilibrium need to be corrected in very short time. An example in this direction are the reaction control systems for steering and attitude control of space vehicles. For instance, see the works [10], [6] and [11]. For exo-atmospheric missiles, reaction control based on solid fuel rocket thrusters is an attractive solution, but they are not throttable and deliver a large impulse during a short time period. The action of small thrusters on the missile can largely be approximated by an impulsive signal.

Let us illustrate the approach of this paper using a simple example of a perturbed double-integrator

$$\begin{aligned} \dot{x}_1 &= x_{2,} \\ \dot{x}_2 &= f(t) + u, \end{aligned} \tag{1}$$

with a control input u and an unknown disturbance $f(t)$. This may be the model of a high precision positioning system of a point mass, with x_1, the position, and x_2, the velocity. Typically a fine positioning control system will use high accuracy sensors and actuators to ensure that the disturbing force is effectively rejected. However, high accuracy sensors and actuators typically have a limited range. The system is designed in such a way that, most of the time, the disturbance will not bring the system out of the range of the fine positioning control system, except if a peak in the disturbing force occurs. These occurrences may be rare, but a fine positioning control system will be poorly equipped to deal with these situations, so an additional system needs to be in place.

In this situation it is important to restore as quickly as possible the system to the neighborhood of the origin, so that the fine control system can take it over. This is in essence the problem that we consider in this paper.

As the occurrence of peaks in the disturbance is assumed to be a rare event, we are not very much concerned with limitations in the energy necessary to perform the correction. We will even allow for impulsive inputs. For example, an impulsive input of the form

$$u = x_2(0)\delta_0 + x_1(0)\dot{\delta}_0$$

brings the double integrator (1) instantaneously to the origin. Of course, this input needs to be practically implemented, and we are examining this problem in this paper too. We limit ourselves here to the open loop control problem, leaving the feedback control for future work.

One of the main problems in impulsive control consists is in the approximation of the delta function and its derivatives. The most popular approximation is one based on the Gaussian bell function as used in [9] and [7]. In this paper, we compare this approximation with two other approximations that have the advantage of having finite support. The main goal and the contribution of this paper is to present a theoretical and practical study of these approximations of the delta function and its derivatives and to compare these approximations with a direct approach to bring the state of the system to the origin that is based on a solution of an integral Volterra equation. We will see that the Gaussian function approximation provides slightly worse results in practice, whereas the direct approach runs into serious numerical problems as the convergence time decreases.

The structure of the paper is as follows. Next section formulates the problem that we consider in this paper: nulling the state of the system in very short time using an impulsive type of input. Section 3 presents the main theoretical contribution of the paper, deriving the formula's for the ideal impulsive input, as well as offering three different solutions to approximate the impulsive inputs with practically implementable signals. These solutions are compared to a conventional solution both qualitatively, as on a numerical example in Section 4. In Section 5, the proposed techniques are illustrated on a satellite rendezvous problem. Finally, Section 6 presents some concluding remarks and ideas for future work.

2 Problem Formulation

Consider the following linear system with n states and m inputs:

$$\dot{x} = Ax + Bu, \; x(0) = x_0. \tag{2}$$

The problem that we consider in this paper can broadly be formulated as: *Find an input signal u that brings the state to the origin in a short time*. Of course, the problem formulated in this way has very many solutions. In fact, it has many ways to approach it.

One of the well-known approaches is the Minimum Time Optimal Control Problem, often used as an application of the Optimum Principle of Pontriaghin. In this case, it is assumed that there is a bound on the magnitude of the control signal u and the problem is to find an admissible input u that brings the state in the origin in minimum time.

This is not however the approach that we take here. In fact, we will not put any bound on the magnitude of u, but we will rather fix a time interval within which the state should be nulled. By making this time interval arbitrarily short, we hope to achieve the stated objective. Even under this formulation of the problem, there are infinitely many solutions. The most straightforward approach, that we will call the *direct approach*, consists of determining an input that solves the integral equation of Volterra-type

$$0 = \exp(A\varepsilon)x_0 + \int_0^{\varepsilon} \exp(A(\varepsilon - s))Bu(s)ds, \tag{3}$$

that can be deduced directly by imposing $x(\varepsilon) = 0$ and using the variations of constants formula for (2). If the system (2) is controllable (an obviously necessary condition for the problem to have a solution at all), there are infinitely many solutions of this equation that can be generated in the following way: Let $Q(\cdot)$ be an $m \times m$-matrix valued function such that

$$W_\varepsilon = \int_0^\varepsilon \exp(A(\varepsilon - s))BQ(s)B^T \exp(A^T(\varepsilon - s))ds \tag{4}$$

is invertible. Then

$$u(t) = -Q(t)B^T \exp(A^T(\varepsilon - t))W_\varepsilon^{-1} \exp(A\varepsilon)x_0 \tag{5}$$

is a solution of (3), as can be easily verified. In this way, we constructed an entire set of solutions for the equation (3). Each of them represent an input function that brings the state of (2) in the origin at time ε.

The approach that we introduce in this paper and that we call the *impulsive input approach* is based on starting with an input u that brings the state in the origin instantaneously. Of course, such an input signal necessarily has an impulsive character. In fact, it is a sum of Dirac delta derivatives. By approximating the Dirac delta derivatives, we can determine practical input signals that "almost" bring the state in the origin. We will actually present two different systematic ways of determining an approximation for the impulsive input.

3 The Impulsive Input Approach

As explained before, we are looking for an input of the form

$$u(t) = \sum_{k=0}^{n-1} \delta_\varepsilon^{(k)}(t)\alpha_k, \tag{6}$$

where $\delta_\varepsilon^{(k)}$ are the generalized derivatives of the Dirac-delta distribution centered in $\varepsilon > 0$, defined (see e.g. [12, Sec. 2.2]) as

$$\int \delta_\varepsilon^{(k)}(t)\phi(t)dt = (-1)^k \phi^{(k)}(\varepsilon),$$

for any test function ϕ, and α_k are vectors of dimension m, that need to be determined. By substituting (6) in the variation-of-constants formula for system (2), we have

$$x(t) = \exp(At)x_0 + \sum_{k=0}^{n-1} [\int_0^t \exp(A(t-s))B\delta_\varepsilon^{(k)}(s)ds]\alpha_k.$$

Using familiar properties of derivatives of the Dirac-delta distribution, this is equivalent to

$$x(t) = \exp(At)x_0 + \sum_{k=0}^{n-1} \exp(A(t-\varepsilon))A^k B\alpha_k,$$

for $t \geq \varepsilon$. Requiring that $x(\varepsilon) = 0$, the coefficients α_k need to satisfy

$$\begin{bmatrix} B\ AB\ \ldots\ A^{n-1}B \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} = -\exp(A\varepsilon)x_0. \tag{7}$$

Notice that the matrix in the left hand side of (7) is the controllability matrix of the pair (A,B). If system (2) is controllable, (7) has a least one solution for every x_0 and every ε. For each such solution, the input (6) will make $x(\varepsilon) = 0$.

Obviously, to make such an approach practical, it is necessary to approximate the impulsive input with a regular input. Fortunately, this is possible. In fact, there are infinitely many ways to do this. We will propose here two types of approximations: using smooth functions (one with bounded support and one using the Gaussian function) and using step functions.

3.1 A C^∞ Approximation with Bounded Support

Consider the following kernel function

$$\omega_h(t) = \begin{cases} \frac{1}{\kappa h} e^{\frac{t^2}{t^2-h^2}}, & |t| < h, \\ 0 & |t| \geq h, \end{cases}$$

where $\kappa = \int_{-1}^{1} e^{\frac{t^2}{t^2-1}} dt$ is a normalization factor, and $h > 0$ is arbitrary. It is well known that the functions ω_h are C^∞ smooth, and as $h \to 0$, these functions approximate in a special sense the Dirac-delta distribution (i.e. they weakly converge to the delta distribution, see e.g. [13, pag. 13 and following]).

We propose to replace the input (6) by

$$u_h(t) = \sum_{k=0}^{n-1} \omega_h^{(k)}(t-\varepsilon)\alpha_k. \tag{8}$$

For any $\varepsilon > h > 0$, this function is smooth and is null everywhere outside the interval $[\varepsilon - h, \varepsilon + h]$. Due to the approximation property of the kernel function, it is to be expected that the state response will come close to the origin for $t \geq h + \varepsilon$.

Proposition 1. *Let $x_h(\cdot)$ denote the solution of (2) for $u = u_h$, for some positive h. Then*

$$\lim_{h \to 0} x_h(h+\varepsilon) = 0.$$

Proof. Introducing (8) into

$$\begin{aligned} x_h(h+\varepsilon) &= \exp(A(h+\varepsilon))x_0 \\ &\quad + \int_0^{h+\varepsilon} \exp(A(h+\varepsilon-s))Bu_h(s)ds, \end{aligned}$$

and using the formula

$$\int_0^{h+\varepsilon} \exp(A(h+\varepsilon-\tau))B\alpha\omega_h^{(k)}(\tau-\varepsilon)d\tau =$$
$$\int_0^{h+\varepsilon} \exp(A(h+\varepsilon-\tau))A^k B\alpha\omega_h(\tau-\varepsilon)d\tau,$$

that can be proven by using integration by parts and induction, we obtain

$$x_h(h+\varepsilon) = \exp(A(h+\varepsilon))x_0$$
$$+\int_0^{h+\varepsilon} \exp(A(h+\varepsilon-\tau))\sum_{k=0}^{n-1} A^k B\alpha_k\omega_h(\tau-\varepsilon)d\tau.$$

By substituting here formula (7) we obtain

$$\begin{aligned} x_h(h+\varepsilon) &= \exp(A(h+\varepsilon))x_0 \\ &\quad - \int_0^{h+\varepsilon} \exp(A(h+2\varepsilon-\tau))x_0\omega_h(\tau-\varepsilon)d\tau. \end{aligned}$$

Because of the finite support of the function ω_h, the limits of integration in the previous expression can be extended to the entire axis. Now, using the fact that, for every continuous function $\phi(t)$

$$\lim_{h\to 0}\int_{-\infty}^{\infty} \phi(t-\tau)\omega_h(\tau)d\tau = \phi(t),$$

we can take directly the limit in the last expression of $x_h(h+\varepsilon)$ and the assertion is readily proved.

3.2 A Gaussian Function Approximation

The approximation of the impulsive input by the Gaussian function approximation was proposed and studied in quite a few references (e.g. [8, 7]). In this case, the Dirac delta function is approximated as

$$\Phi_h(t) = \frac{1}{\sqrt{2\pi h}}e^{\frac{t^2}{2h^2}}.$$

Although, this is a C^∞ function, unlike the previous approximation, this function does not have compact support. A similar result as Proposition 1 holds for this approximation, but we will not state here since this approximation was extensively studied in the literature.

3.3 A Piecewise-Constant Function Approximation

The function

$$\delta_h^{[0]}(t) = \begin{cases} \frac{1}{2h}, & |t| \le h \\ 0, & rest. \end{cases} \tag{9}$$

is clearly a piecewise constant approximation of the Dirac delta function. The first order derivative of the delta function can be approximated by the following "symmetric finite difference" relation

$$\delta_h^{[1]} = \frac{\delta_{\frac{h}{2}}^{[0]}(t+\frac{h}{2}) - \delta_{\frac{h}{2}}^{[0]}(t-\frac{h}{2})}{h} = \begin{cases} \frac{1}{h^2}, & -h \le t \le 0, \\ -\frac{1}{h^2}, & 0 < t \le h, \\ 0, & rest. \end{cases} \tag{10}$$

Notice that the support of $\delta_h^{[1]}$ as defined above, just as the support of $\delta_h^{[0]}$ is $[-h,h]$. Approximations of the higher order derivatives are defined iteratively as

$$\delta_h^{[k]}(t) = \frac{\delta_{\frac{h}{2}}^{[k-1]}(t+\frac{h}{2}) - \delta_{\frac{h}{2}}^{[k-1]}(t-\frac{h}{2})}{h}, \tag{11}$$

for all $k \ge 1$.

We propose to replace the input (6) by

$$u_h(t) = \sum_{k=0}^{n-1} \delta_h^{[k]}(t-\varepsilon)\alpha_k. \tag{12}$$

Just as the approximation (8), for any $\varepsilon \ge h > 0$, this function is null everywhere outside the interval $[\varepsilon - h, \varepsilon + h]$. The next result shows that this input is also bringing the state close to the origin for $t \ge h + \varepsilon$ for h small enough.

Proposition 2. *Let $x_h(\cdot)$ denote the solution of (2) for $u = u_h$, for some positive h. Then*

$$\lim_{h\to 0} x_h(h+\varepsilon) = 0.$$

Proof. Introducing (12) into

$$\begin{aligned} x_h(h+\varepsilon) &= \exp(A(h+\varepsilon))x_0 + \\ &\quad + \int_0^{h+\varepsilon} \exp(A(h+\varepsilon-s))Bu_h(s)ds, \end{aligned}$$

while taking into account that the support of u_h is $[h+\varepsilon, \varepsilon - h]$,

$$\begin{aligned} x_h(h+\varepsilon) &= \exp(A(h+\varepsilon))x_0 \\ &\quad + \sum_{k=0}^{n-1} \alpha_k \int_{-h+\varepsilon}^{h+\varepsilon} e^{A(h+\varepsilon-s)} B \delta_h^{[k]}(s-\varepsilon)ds, \end{aligned}$$

and after a simple change of variable

$$x_h(h+\varepsilon) = e^{A(h+\varepsilon)}x_0 + \sum_{k=0}^{n-1} \alpha_k \int_{-h}^{h} e^{A(h-s)}B\delta_h^{[k]}(s)ds.$$

From the last relation and from (7), it is clear that the assertion is proved if we show that

$$\lim_{h\to 0}\int_{-h}^{h} \exp(A(h-s))B\delta_h^{[k]}(s)ds = A^kB. \tag{13}$$

We prove this relation by induction. First of all, for $k = 0$, the relation

$$\lim_{h\to 0}\frac{1}{2h}\int_{-h}^{h} \exp(A(h-s))Bds = B$$

follows from the properties of the matrix exponential.

Let us denote by

$$I_k(h) = \int_{-h}^{h} \exp(A(h-s))B\delta_h^{[k]}(s)ds. \tag{14}$$

By hypothesis

$$\lim_{h\to 0} I_{k-1}(h) = A^{k-1}B.$$

Using the recursive definition (11), we can write

$$I_k(h) = \frac{1}{h}\int_{-h}^{h} e^{A(h-s)}B[\delta_{\frac{h}{2}}^{[k-1]}(s+\frac{h}{2}) - \delta_{\frac{h}{2}}^{[k-1]}(s-\frac{h}{2})]ds.$$

Taking into account that the support of $\delta_{\frac{h}{2}}^{[k-1]}$ is $[-\frac{h}{2},\frac{h}{2}]$, we have

$$\begin{aligned} I_k(h) &= \frac{1}{h}[\int_{-h}^{0} \exp(A(h-s))B\delta_{\frac{h}{2}}^{[k-1]}(s+\frac{h}{2})ds \\ &\quad - \int_{0}^{h} \exp(A(h-s))B\delta_{\frac{h}{2}}^{[k-1]}(s-\frac{h}{2})ds], \end{aligned}$$

and translating the variable in each integral,

$$\begin{aligned} I_k(h) &= \frac{1}{h}[\int_{-\frac{h}{2}}^{\frac{h}{2}} \exp(A(\frac{3h}{2}-s))B\delta_{\frac{h}{2}}^{[k-1]}(s)ds \\ &\quad - \int_{-\frac{h}{2}}^{\frac{h}{2}} \exp(A(\frac{h}{2}-s))B\delta_{\frac{h}{2}}^{[k-1]}(s)ds], \end{aligned}$$

that is

$$\begin{aligned}
I_k(h) &= \frac{1}{h}[\exp(Ah)-I]\int_{-\frac{h}{2}}^{\frac{h}{2}} e^{A(\frac{h}{2}-s)}B\delta_{\frac{h}{2}}^{[k-1]}(s)ds \\
&= \frac{1}{h}[\exp(Ah)-I]I_{k-1}(\frac{h}{2}).
\end{aligned}$$

Since

$$\lim_{h\to 0}\frac{1}{h}[\exp(Ah)-I]=A,$$

it is clear that

$$\lim_{h\to 0} I_k(h) = A^k B,$$

and the proof is completed.

4 Comparing the Impulsive Input Approach with the Direct Approach

The approach based on approximating the impulsive input requires very simple computations, and they are independent of the convergence time h. Formula (12) can be computed by solving a linear equation (7) and using analytic functions such as the function ω_h and its derivatives that can be even computed off-line. Similarly, the approximation using piecewise constant function and the approximation using Gaussian functions can easily be implemented using appropriate data computed off-line. The piecewise constant approximation may be easier to implement in a practical situation, and presents a definite advantage that it allows a good estimate of the maximum value of the control input. On the other hand, formula (5) requires numerical integration to compute W_h and then matrix exponentials are also required.

However, formula (12) is only an approximate solution of the state nulling problem, whereas formula (5) is an exact solution. Also, any choice of $Q(t)$ that keeps W_h invertible generates a solution to the state nulling problem. Notice that equation (7) may also have an infinity of solutions in case $m > 1$, that is, if there are more than a single input.

So far the qualitative analysis. We tested the various approximations of the impulsive control as well as the direct approach on a few numerical examples. Due to space limitation, we present only a single example of a third order system with one input:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 3 & -2 & 1 \\ 3 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, x_0 = \begin{bmatrix} 30 \\ -20 \\ 42 \end{bmatrix}.$$

The results for the first, impulsive input approach are represented in Figures 1 and 2 for the case of the C^{∞} approximation and in Figures 3 and 4 for the case of the piecewise constant approximation. The results for the Gaussian approximation are represented in Figures 5 and 6. Notice that the Gaussian approximation provided worse results than the other two approximations especially as the convergence time decreases. We have seen this for all the examples considered, and it is probably due to the fact that the Gaussian function has no compact support, but we do not have at this time a rigorous argument to explain this observation.

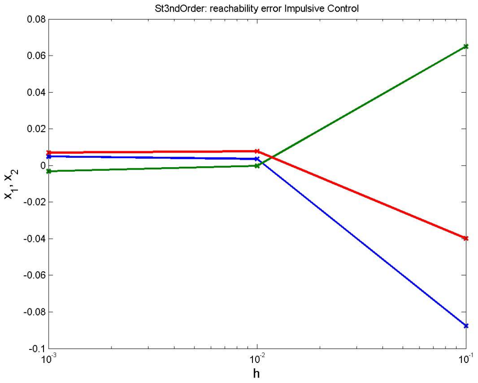


Fig. 1 State value after applying the impulsive input approach with C^{∞}approximation for different times h

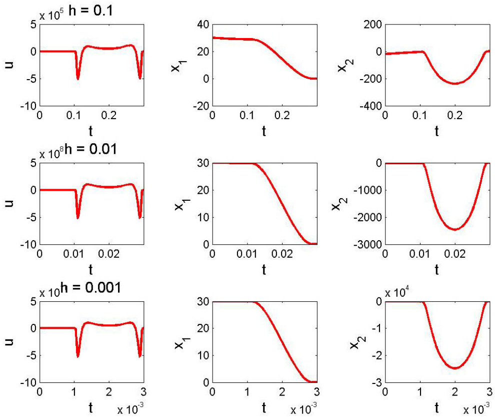


Fig. 2 Input and state response using the impulsive input approach with C^{∞}approximation for different times h

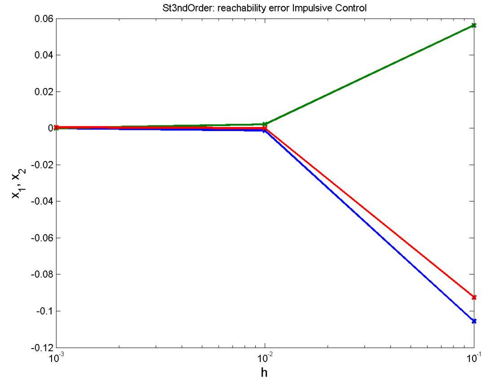


Fig. 3 State value after applying the impulsive input approach with piecewise constant approximation for different times *h*

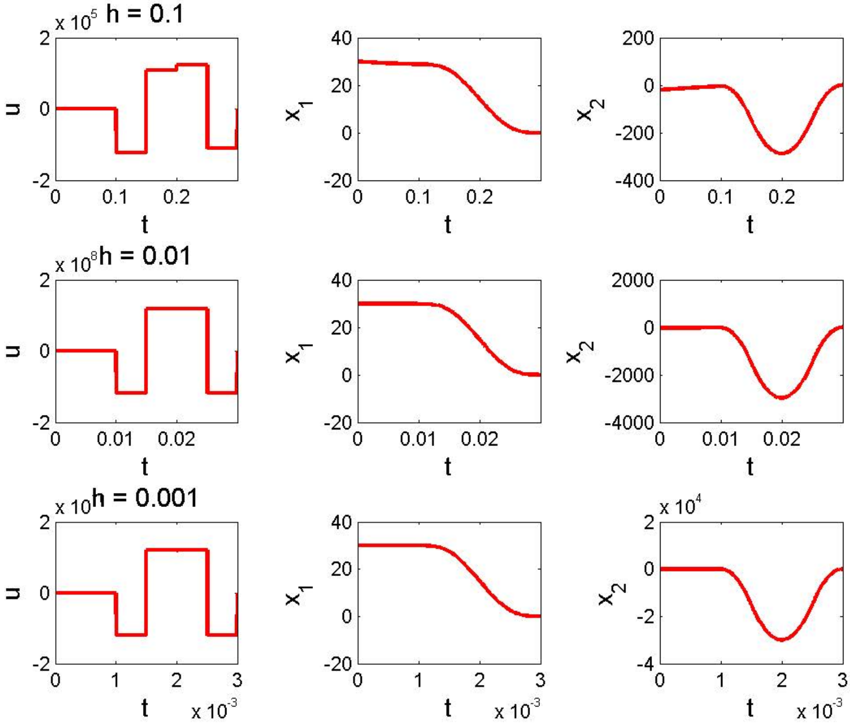


Fig. 4 Input and state response using the impulsive input approach with piecewise constant approximation for different times *h*

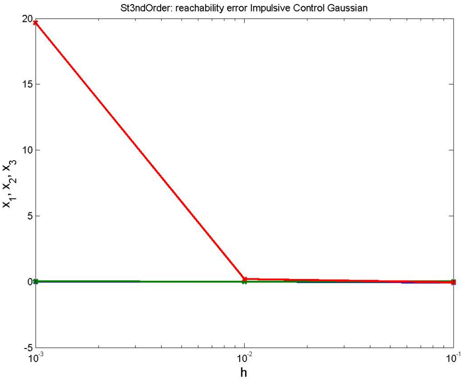


Fig. 5 State value after applying the impulsive input approach with Gaussian approximation for different times *h*

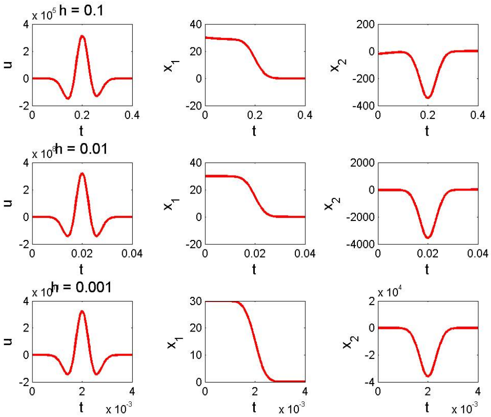


Fig. 6 Input and state response using the impulsive input approach with Gaussian approximation for different times *h*

Applying the direct approach on the same example, we obtain the results represented in Figures 7 and 8. This is much worse than expected. Actually, the state is effectively brought close to the origin only for the case of $h = 0.1$, in which case the state coordinates are of the order 0.01. However, this is not visible in Figure 7 due to the very bad performance for the other two values of h.

Analyzing the cause of the failure of the direct approach in this case, we notice that formula (5) involves the inverse of the matrix W_h, which even if invertible for all $h > 0$ in case that the pair (A, B) is controllable, may actually be quite poorly

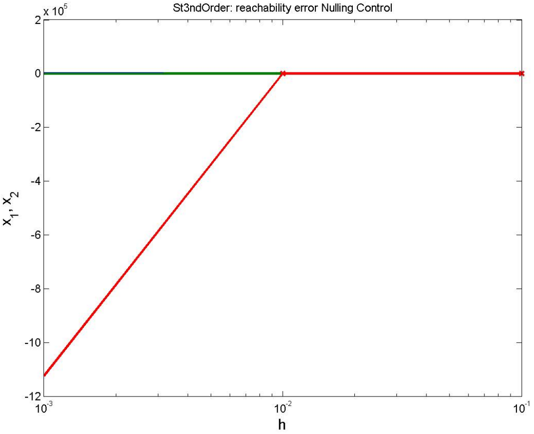


Fig. 7 State value after applying the nulling input for different times h

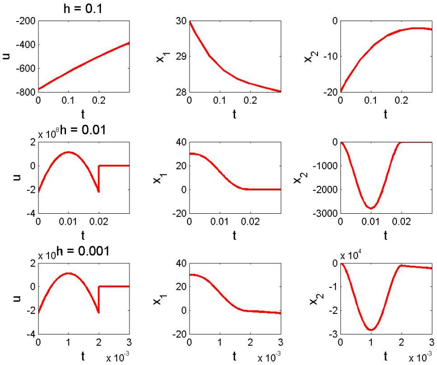


Fig. 8 Input and state response using the direct approach

conditioned. In this case, the computation of the nulling input will be challenging. Let us consider

$$\tilde{W}_h = \int_0^h \exp(-As)BB^T \exp(-A^T s))ds,$$

which is related to W_h by the relation

$$W_h = \exp(Ah)\tilde{W}_h \exp(A^T h).$$

Therefore, inverting W_h is just as difficult as inverting $\tilde{W}_h$. Figure 9 represents the condition number of $\tilde{W}_h$, that is defined as the ratio of the largest and the smallest singular value. It is well known that a large value of the condition number is indicating that the matrix is badly conditioned numerically, and it is easy to see that for

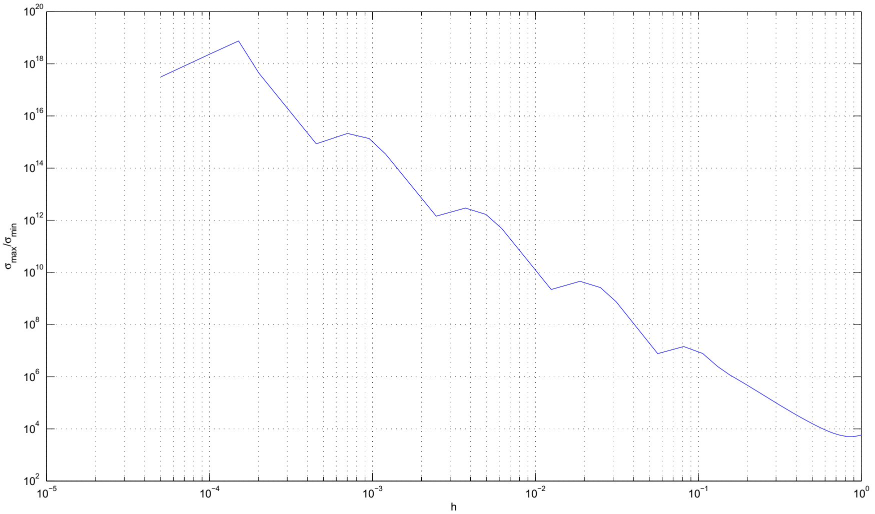


Fig. 9 Numerical condition number of $\tilde{W}_h$ as a function of h

$h = 0.01$, the condition number is 10^{10}, whereas for $h = 0.001$, the condition number is around 10^{15}. This explains the failure of the direct approach for these values of h, whereas the impulsive input approach is clearly not affected by this issue.

The phenomenon illustrated in this example is generic. Even in the two dimensional case, we show that the condition number of W_h tends to infinity as h decreases to zero. Indeed let

$$A = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

It is easy to compute the expression of W_h in this case explicitly as

$$\begin{aligned} W_h &= \begin{bmatrix} w_1(h) & w_{12}(h) \\ w_{12}(h) & w_2(h) \end{bmatrix} \\ &= \begin{bmatrix} \frac{e^{2a_1h}-1}{2a_1}b_1^2 & \frac{e^{(a_1+a_2)h}-1}{a_1+a_2}b_1b_2 \\ \frac{e^{(a_1+a_2)h}-1}{a_1+a_2}b_1b_2 & \frac{e^{2a_2h}-1}{2a_2}b_2^2 \end{bmatrix}, \end{aligned} \tag{15}$$

where can see that, as $h \to 0$

$$\frac{w_1(h)}{h} \to b_1^2, \; \frac{w_2(h)}{h} \to b_2^2, \; \frac{w_{12}(h)}{h} \to b_1b_2.$$

The two eigenvalues of W_h are

$$\lambda_1(h) = \frac{1}{2}(w_1 + w_2 + \sqrt{(w_1 - w_2)^2 + 4w_{12}^2}),$$

$$\lambda_2(h) = \frac{1}{2}(w_1 + w_2 - \sqrt{(w_1 - w_2)^2 + 4w_{12}^2}).$$

Using repeatedly the fact that $\lim_{h\to 0} \frac{e^{ah}-1}{ah} = 1$, and the expressions in (15), we readily deduce that

$$\lim_{h\to 0} \frac{\lambda_1(h)}{h} = b_1^2 + b_2^2,$$

which is not zero unless the system is uncontrollable. On the other hand

$$\lim_{h\to 0} \frac{\lambda_2(h)}{h} = 0.$$

Consequently

$$\lim_{h\to 0} \frac{\lambda_1(h)}{\lambda_2(h)} = \infty,$$

which shows that W_h becomes badly conditioned as h becomes small. It is very likely that this result holds true for the higher dimensional case, but it is already clear that the example presented in this section is not isolated in the sense that the impulsive control approach is better suited than the direct approach to drive the state quickly to the origin.

5 Application to an Orbital Rendezvous Problem

We consider the linearized model for orbital rendezvous that is well-known as the Clohessy-Wiltshire equations [14] that expresses the relative motion of a chasing spacecraft in the coordinate system fixed to the target spacecraft, as represented in Figure 10,

$$\begin{aligned} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x &= u_x, \\ \ddot{y} + 2\omega\dot{x} &= u_y, \\ \ddot{z} + \omega^2 z &= u_z, \end{aligned} \tag{16}$$

where ω is the orbital rate, x, y and z are the components of the relative displacement between chasing spacecraft and the target, and u_x, u_y and u_z are the components of the thrust acceleration of the chasing spacecraft. If we denote by $X = \begin{bmatrix} x \; y \; z \; \dot{x} \; \dot{y} \; \dot{z} \end{bmatrix}^T$ the state space vector of this model, and by $U = \begin{bmatrix} u_x \; u_y \; u_z \end{bmatrix}^T$ the input vector, the motion equations (16) can be written as

$$\frac{d}{dt}X = AX + BU, \tag{17}$$

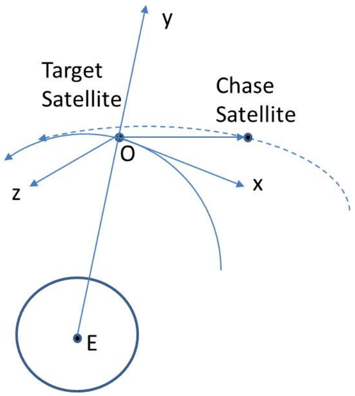


Fig. 10 Schematic representation of the satellite rendez-vous problem

where

$$A = \begin{bmatrix} & O_3 & & & I_3 & \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} O_3 \\ I_3 \end{bmatrix}.$$

Clearly, the rendezvous problem can be formulated as bringing the initial state $X(t_o)$ to a final state $X(t_f) = \begin{bmatrix} x_f \ y_f \ z_f \ 0 \ 0 \ 0 \end{bmatrix}^T$, where x_f, y_f and z_f are the final relative displacements between the two spacecrafts. We seek inputs of the form

$$U(t) = \delta(t-t_i)\alpha_o + \delta'(t-t_i)\alpha_1,$$

where α_o and α_1 are constant vectors in $\mathbb{R}^3$. The state at final time t_f is

$$X(t_f) = e^{A(t_f-t_o)}X(t_o) + e^{A(t_f-t_i)}B\alpha_o + e^{A(t_f-t_i)}AB\alpha_1.$$

If the initial and the final states are known, this relation can readily be solved for the impulse coefficients

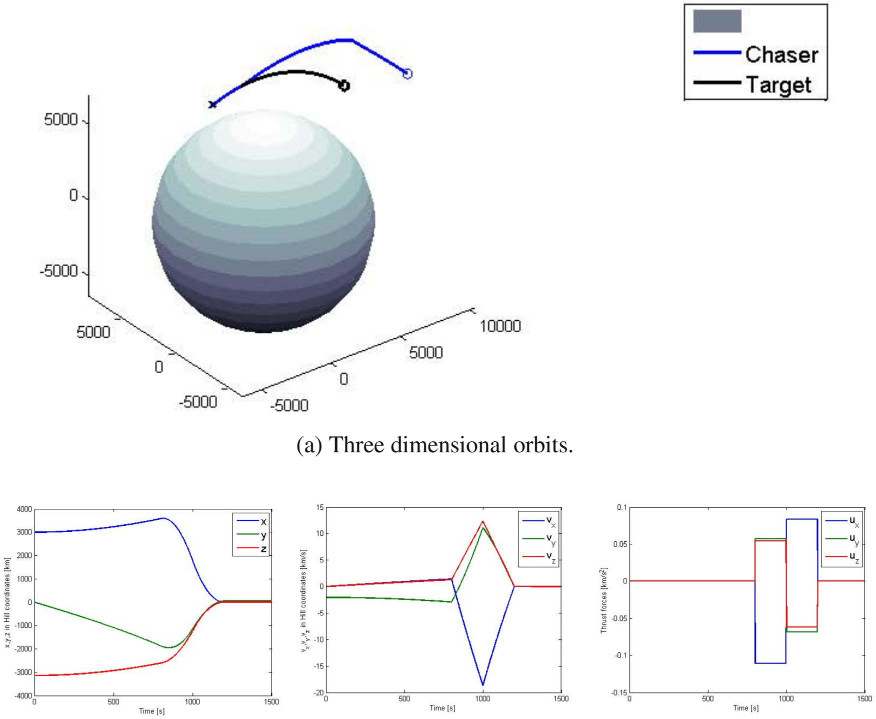
$$\begin{bmatrix} \alpha_1 \\ \alpha_o \end{bmatrix} = \begin{bmatrix} AB \ B \end{bmatrix}^{-1} [e^{A(t_i-t_f)}X(t_f) - e^{A(t_i-t_o)}X(t_o)].$$

This expression can be used to give an analytic expression for the impulse coefficients if we notice that

$$\begin{bmatrix} AB \ B \end{bmatrix}^{-1} = \begin{bmatrix} I_3 & O_3 \\ -A_o & I_3 \end{bmatrix},$$

Table 1 Numerical values of the parameters for the rendez-vous problem

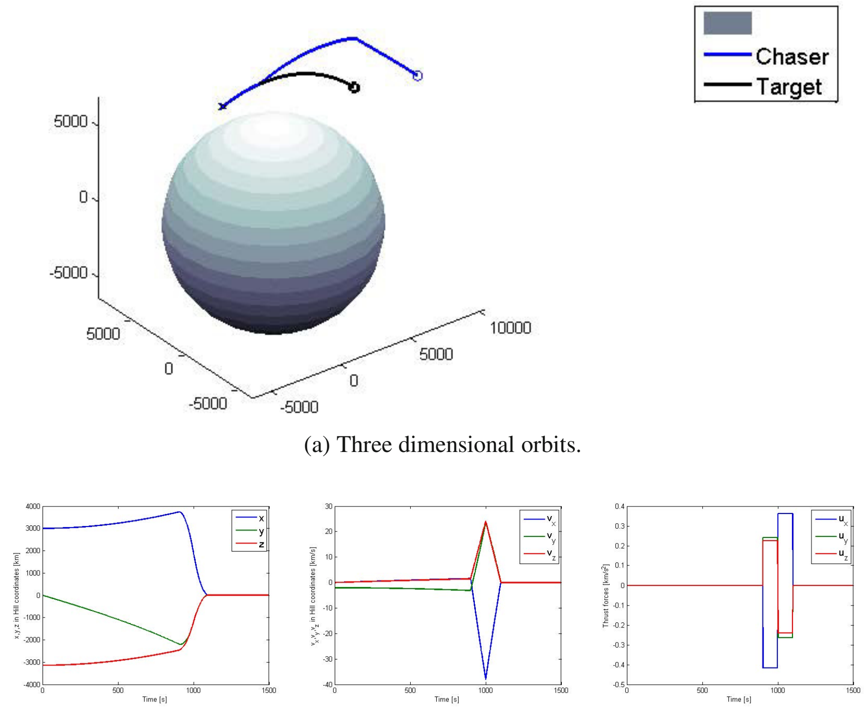
Parameter	**Symbol**	**Value**
Major semi-axis target	R_T	9000 km
Orbital inclination target		-50 deg
Orbital rate of the target	$\omega = \sqrt{\frac{\mu}{R_T^3}}$	$7.39444\ 10^{-4}$ rad/s
Initial semi-axis chaser	R_I	12000 km
Orbital inclination chaser		-30 deg
Final time	t_f	1500 s
Impulse application time	t_i	1000 s
Rendez-vous position (Hill coordinates)	x_f, y_f, z_f	$10, 10, 10$ km
Time step for impulse approximation	h	200, 100, 20 s



(a) Three dimensional orbits.

(b) Positions, velocities and thrust accelerations of chaser in Hill coordinates.

Fig. 11 Simulation results for the case of pulse width $h = 200$ s



(a) Three dimensional orbits.

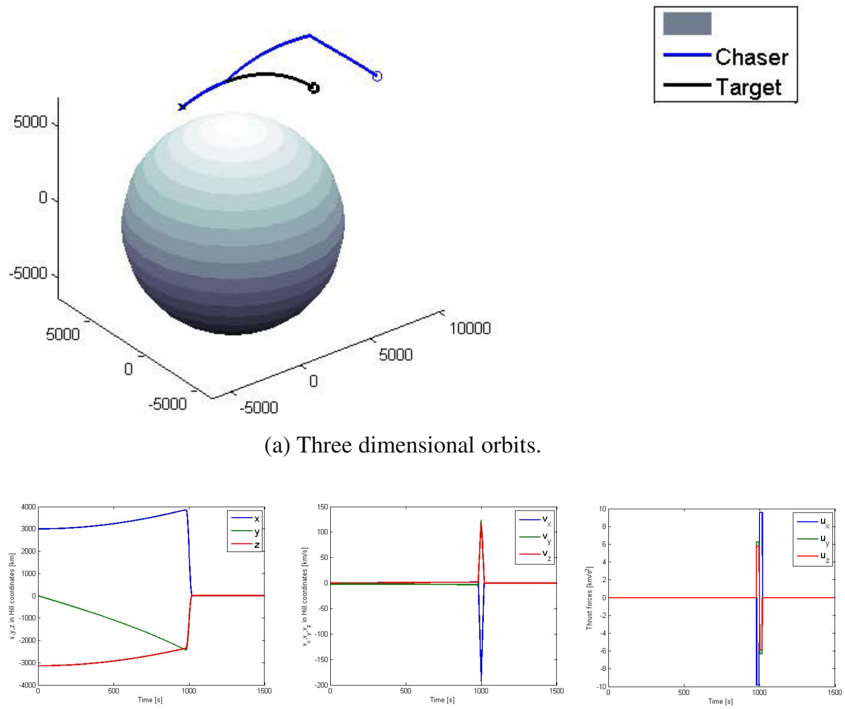
(b) Positions, velocities and thrust accelerations of chaser in Hill coordinates.

Fig. 12 Simulation results for the case of pulse width $h = 100$ s

with $A_o = \begin{bmatrix} 0 & 2\omega & 0 \\ -2\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and that

$$e^{A\tau} = \begin{bmatrix} 2-\cos\omega\tau & 0 & 0 & \frac{\sin\omega\tau}{\omega} & \frac{2(1-\cos\omega\tau)}{\omega} & 0 \\ 6(\sin\omega\tau-\omega\tau) & 1 & 0 & \frac{2(\cos\omega\tau-1)}{\omega} & \frac{4\sin\omega\tau-3\omega\tau}{\omega} & 0 \\ 0 & 0 & \cos\omega\tau & 0 & 0 & \frac{\sin\omega\tau}{\omega} \\ 3\omega\sin\omega\tau & 0 & 0 & \cos\omega\tau & 2\sin\omega\tau & 0 \\ 6(\cos\omega\tau-1) & 0 & 0 & -2\sin\omega\tau & 4\cos\omega\tau-3 & 0 \\ 0 & 0 & -\omega\sin\omega\tau & 0 & 0 & \cos\omega\tau \end{bmatrix}.$$

By applying these relations and the approximations schemes proposed here, it is possible to devise efficient online algorithms for computing the steering thrust for solving the rendezvous problem. For illustration purposes, we consider a numerical example with the parameter values given in Table 1. Only the piecewise constant approximation approach is considered as it is better suited for the case of solid fuel thrusters. The time h was succesively varied from 200 s, 100 s, and 20 s. The results



(a) Three dimensional orbits.

(b) Positions, velocities and thrust accelerations of chaser in Hill coordinates.

Fig. 13 Simulation results for the case of pulse width $h = 20$ s

of the simulations are represented in Figure 11, Figure 12, and respectively Figure 13. As expected, the necessary thrust acceleration level is increasing as h becomes smaller. However, the vectors α_o and α_1 are independent of h. It is therefore easy to determine, using our approach, a minimum h that is compatible with the maximum achievable thrust acceleration.

The proposed impulsive control technique, combined with a robust feedback control, including traditional and higher order sliding mode control algorithms (see e.g. [15, 16]) can also be applied to the satellite formation control problem considered in [17]. However, such a closed loop implementation will be the subject of future work.

6 Conclusions and Way Forward

An impulsive input approach to the problem of driving the state of a linear system to the origin in very short time is studied in this work. The control input was derived as a linear combination of the Dirac delta function and its derivatives. Subsequently, two approximation schemes were proposed for approximating the impulsive input and theoretical results were proven to confirm their validity. Using a numerical example, we have shown that a direct approach to obtain a nulling input by solving an integral equation runs into numerical problems for short time intervals, whereas the solutions obtained by the impulsive input approach are not affected. For the second order case, we showed that the numerical problems are generic and not particular to the chosen example. Another observation is that the approximation using the Gauss function may give poor results due to the unbounded support, although this seems to be the approximation most studied in the literature.

Future work will concentrate on combining the proposed approach with robust feedback control, including adaptive output feedback sliding mode estimation and control.

References

1. Yang, T.: Impulsive control theory. LNCIS, vol. 272. Springer, Heidelberg (2001)
2. Miller, B., Rubinovitch, E.: Impulsive Control in Continuous and Discrete- Continuous Systems. Kluwer Academic Publishers, Amsterdam (2002)
3. Li, Z., Soh, Y., Wen, C.: Switched and impulsive systems: analysis, design and applications. LNCIS, vol. 313. Springer, Heidelberg (2005)
4. Haddad, W., Chellaboina, V., Nersesov, S.: Impulsive and hybrid dynamical systems: stability, dissipativity, and control. Princeton series in applied mathematics. Princeton University Press (2006)
5. Kurzhanski, A., Daryin, A.: Dynamic programming for impulse controls. Annual Reviews in Control 32(2), 213–227 (2008)
6. Glumineau, A., Shtessel, Y., Plestan, F.: Impulsive-sliding mode adaptive control of second order system. Preprints of the 18th IFAC World Congress, Milano, Italy, August 28-September 2, pp. 5389–5394 (2011)
7. Karageorgos, A.D., Pantelous, A.A., Kalogeropoulos, G.I.: Transferring instantly the state of higher-order linear descriptor (regular) differential systems using impulsive inputs. J. Control Sci. Eng. 2009, 6:1–6:29 (2009)
8. Gupta, S.C., Hasdorff, L.: Changing the state of a linear system by use of normal function and its derivatives. Journal of Electronics and Control 14(3), 351–359 (1963)
9. Karcanias, N., Kouvaritakis, B.: Zero time adjustment of initial conditions and its relationship to controllability subspaces. International Journal of Control 29(5), 749–765 (1979)
10. Rosello, A.: A vehicle health monitoring system for space shuttle reaction control system during reentry. M.Sc. Thesis, MIT (May 1995)
11. Li, H.-Y., Luo, Y.-Z., Tang, G.-J.: Optimal multi-objective linearized impulsive rendezvous under uncertainty. Acta Astronautica 66(3-4), 439–445 (2010)
12. Gel'fand, I., Shilov, G.: Generalized functions. Academic Press (1964)
13. Sobolev, S.: Applications of functional analysis in mathematical physics. Translations of Mathematical Monographs, vol. 7. American Mathematical Society (1963)
14. Clohessy, W.H., Wiltshire, R.: Terminal guidance system for satellite rendezvous. Journal of Aerospace Science 27(9), 653–658 (1960)

15. Edwards, C., Spurgeon, S.: Sliding Mode Control: Theory And Applications. Taylor & Francis systems and control book series. Taylor & Francis (1998)
16. Levant, A.: Higher-order sliding modes, differentiation and output-feedback control. International Journal of Control 76(9-10), 924–941 (2003)
17. Massey, T., Shtessel, Y.: Continuous traditional and high order sliding modes for satellite formation control. AIAA Journal on Guidance, Control, and Dynamics 28(4), 826–831 (2005)

Model Formulation of Pursuit Problem with Two Pursuers and One Evader

Sergey S. Kumkov, Stéphane Le Ménec, and Valerii S. Patsko

Abstract. We study a model differential zero-sum game, which can be regarded as an idealized variant of the final stage of a space pursuit, in which two pursuing objects and one evader are involved. Results of numeric constructions of level sets of the value function for qualitatively different cases of the game parameters and results of simulation of optimal motions are presented.

1 Introduction and Problem Formulation

1) In the paper, a model differential zero-sum game with two pursuers and one evader is studied. Three inertial objects moves in the straight line. The dynamics descriptions for pursuers P_1 and P_2 are

$$\begin{aligned}
&\ddot{z}_{P_1} = a_{P_1}, && \ddot{z}_{P_2} = a_{P_2},\\
&\dot{a}_{P_1} = (u_1 - a_{P_1})/l_{P_1}, && \dot{a}_{P_2} = (u_2 - a_{P_2})/l_{P_2},\\
&|u_1| \le \mu_1, && |u_2| \le \mu_2,\\
&a_{P_1}(t_0) = 0, && a_{P_2}(t_0) = 0.
\end{aligned} \tag{1}$$

Sergey S. Kumkov · Valerii S. Patsko
Institute of Mathematics and Mechanics UrB RAS,
S.Kovalevskaya str., 16, Ekaterinburg, 620990, Russia
Ural Federal University,
Turgenev str., 4, 620075, Russia
e-mail: sskumk@gmail.com, patsko@imm.uran.ru

Stéphane Le Ménec
EADS / MBDA France, 1 Avenue Reaumur,
92358 Le Plessis Robinson Cedex, France
e-mail: stephane.le-menec@mbda-systems.com

Here, z_{P_1} and z_{P_2} are the geometric coordinates of the pursuers; a_{P_1} and a_{P_2} are their accelerations generated by the controls u_1 and u_2. The time constants l_{P_1} and l_{P_2} define how fast the controls affect the systems.

The dynamics of the evader E is similar:

$$\ddot{z}_E = a_E, \quad \dot{a}_E = (v - a_E)/l_E, \quad |v| \le \nu, \quad a_E(t_0) = 0. \tag{2}$$

Let us fix some instants T_1 and T_2. At the instant T_1, the miss of the first pursuer with respect to the evader is computed, and at the instant T_2, the miss of the second one is calculated:

$$r_{P_1,E}(T_1) = |z_E(T_1) - z_{P_1}(T_1)|, \quad r_{P_2,E}(T_2) = |z_E(T_2) - z_{P_2}(T_2)|. \tag{3}$$

Assume that the pursuers act in coordination. This means that we can join them into one player P (which will be called the *first player*). This player governs the vector control $u = (u_1, u_2)$. The evader is regarded as the *second player*. The resultant miss is computed by the following formula:

$$\varphi = \min\{r_{P_1,E}(T_1),\ r_{P_2,E}(T_2)\}. \tag{4}$$

At any instant t, both players know exact values of all state coordinates z_{P_1}, $\dot{z}_{P_1}$, a_{P_1}, z_{P_2}, $\dot{z}_{P_2}$, a_{P_2}, z_E, $\dot{z}_E$, a_E. The vector composed of these components is denoted by z. The first player choosing its feedback control minimizes the miss φ, the second one maximizes it.

Relations (1)–(4) define a standard antagonistic differential game. One needs to construct the value function $(t, z) \mapsto \mathcal{V}(t, z)$ of this game and optimal (or quasioptimal) strategies of the players.

2) Up to now, there are a lot of publications dealing with differential games where one group of objects pursues another group; concerning games with linear dynamics see, for example, works [1,4,6,11,12]. The problem under consideration has two pursuers and one evader. So, from the point of view of number of objects, it is the simplest one. On the other hand, strict mathematical studies of problems "group-on-group" usually include quite strong assumptions onto the dynamics of objects, dimension of the state vector, and conditions of termination. Unlike, this paper considers the problem without any assumptions of these types.

3) Let us describe a practical problem, whose reasonable simplification gives the model game (1)–(4). Suppose that two pursuing objects attack the evading one with high velocities. They can be rockets or aircrafts in the horizontal plane (Fig. 1). A nominal motion of the first pursuer is chosen such that at the instant T_1 the exact capture occurs. In the same way, a nominal motion of the second pursuer is chosen (the capture is at the instant T_2). But indeed, the real positions of the objects differ from the nominal ones. Moreover, the evader using its control can change its trajectory but not essentially, without sharp turns. Coordinated efforts of the pursuers are computed during the

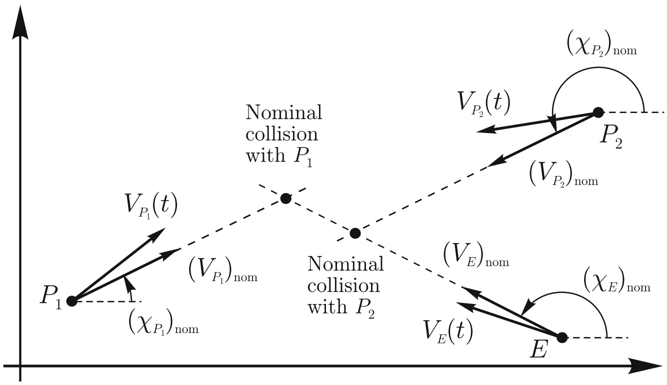

$(\chi_{P_2})_{\mathrm{nom}}$
$V_{P_2}(t)$
P_2
$(V_{P_2})_{\mathrm{nom}}$
Nominal collision with P_1
$V_{P_1}(t)$
$(V_{P_1})_{\mathrm{nom}}$
P_1
$(\chi_{P_1})_{\mathrm{nom}}$
Nominal collision with P_2
$(V_E)_{\mathrm{nom}}$
$(\chi_E)_{\mathrm{nom}}$
$V_E(t)$
E

Fig. 1 Scheme of the nominal motions in the pursuit problem with weak-maneuvering objects

process by the feedback method to minimize the resultant miss, which is the minimum of the distances at the instants T_1 and T_2 from the first and second pursuers, respectively, to the evader.

Assume that we can choose a line (in Fig. 1, it is a horizontal line) such that the major components of velocities of all three objects are directed along it. Then, the misses at the instants T_1 and T_2, can be computed along a direction orthogonal to such a line ignoring difference of positions along this line.

The passage from the original non-linear dynamics to a dynamics, which is linearized with respect to the nominal motions, gives [13, 14] the problem under consideration.

2 Passage to Two-Dimensional Differential Game

At first, let us pass to the relative geometric coordinates

$$y_1 = z_E - z_{P_1}, \quad y_2 = z_E - z_{P_2} \tag{5}$$

in dynamics (1), (2), and payoff function (4). After this, we have the following notations:

$$\begin{array}{ll}
\ddot{y}_1 = a_E - a_{P_1}, & \ddot{y}_2 = a_E - a_{P_2}, \\
\dot{a}_{P_1} = (u_1 - a_{P_1})/l_{P_1}, & \dot{a}_{P_2} = (u_2 - a_{P_2})/l_{P_2}, \\
\dot{a}_E = (v - a_E)/l_{P_1}, & |u_2| \le \mu_2, \\
|u_1| \le \mu_1,\ |v| \le \nu, & \varphi = \min\{|y_1(T_1)|,\ |y_2(T_2)|\}.
\end{array} \tag{6}$$

State variables of system (6) are y_1, $\dot{y}_1$, a_{P_1}, y_2, $\dot{y}_2$, a_{P_2}, a_E; u_1 and u_2 are controls of the first player; v is the control of the second one. The payoff function φ depends on the coordinate y_1 at the instant T_1 and on the coordinate y_2 at the instant T_2.

A standard approach to study linear differential games with fixed terminal instant and payoff function depending on some target coordinates of the state vector at the terminal instant is to pass to new state coordinates (see, for example, [7, 8]) that can be treated as values of the target coordinates forecasted to the terminal instant under zero controls. Often, these coordinates are called the *zero effort miss coordinates* [13, 14]. In our case, we have two instants T_1 and T_2, but coordinates computed at these instants are independent; namely, at the instant T_1, we should take into account $y_1(T_1)$ only, and at the instant T_2, we use the value $y_2(T_2)$. This fact allows us to use the mentioned approach when solving the differential game (6). With that, we pass to new state coordinates x_1 and x_2, where $x_1(t)$ is the value of y_1 forecasted to the instant T_1 and $x_2(t)$ is the value of y_2 forecasted to the instant T_2.

The forecasted values are computed by formula

$$x_i = y_i + \dot{y}_i \tau_i - a_{P_i} l_{P_i}^2 h(\tau_i / l_{P_i}) + a_E l_E^2 h(\tau_i / l_E), \quad i = 1, 2. \tag{7}$$

Here, x_i, y_i, $\dot{y}_i$, a_{P_i}, and a_E depend on t; $\tau_i = T_i - t$. Function h is described by the relation $h(\alpha) = e^{-\alpha} + \alpha - 1$. Emphasize that the values τ_1 and τ_2 are connected to each other by the relation $\tau_1 - \tau_2 = \text{const} = T_1 - T_2$. It is very important that $x_i(T_i) = y_i(T_i)$. Let $X(t, z)$ be a two-dimensional vector composed of the variables x_1, x_2 defined by formulae (5), (7).

The dynamics in the new coordinates x_1, x_2 is the following [9]:

$$\begin{aligned} \dot{x}_1 &= -l_{P_1} h(\tau_1 / l_{P_1}) u_1 + l_E h(\tau_1 / l_E) v, \quad |u_1| \le \mu_1, \ |u_2| \le \mu_2, \\ \dot{x}_2 &= -l_{P_2} h(\tau_2 / l_{P_2}) u_2 + l_E h(\tau_2 / l_E) v, \quad |v| \le \nu. \end{aligned} \tag{8}$$

The payoff function is $\varphi\big(x_1(T_1), x_2(T_2)\big) = \min\{|x_1(T_1)|, \ |x_2(T_2)|\}$.

The first player governs the controls u_1, u_2 and minimizes the payoff φ; the second one has the control v and maximizes φ.

Note that the control u_1 (u_2) affects only the horizontal (vertical) component $\dot{x}_1$ ($\dot{x}_2$) of the velocity vector $\dot{x} = (\dot{x}_1, \dot{x}_2)^{\mathrm{T}}$. When $T_1 = T_2$, the second summand in dynamics (8) is the same for $\dot{x}_1$ and $\dot{x}_2$. Thus, the component of the velocity vector $\dot{x}$ depending on the second player control is directed at any instant t along the bisectrix of the first and third quadrants of the plane x_1, x_2. When $v = +\nu$, the angle between the axis x_1 and the velocity vector of the second player is $45°$; when $v = -\nu$, the angle is $225°$. This property simplifies the dynamics in comparison with the case $T_1 \ne T_2$.

Let $x = (x_1, x_2)^{\mathrm{T}}$ and $V(t, x)$ be the value of the value function of game (8) at the position (t, x). From general results of the differential game theory, it follows that $\mathcal{V}(t, z) = V\big(t, X(t, z)\big)$. This relation allows one to compute the value function of the original game (1)–(4) using the value function for game (8).

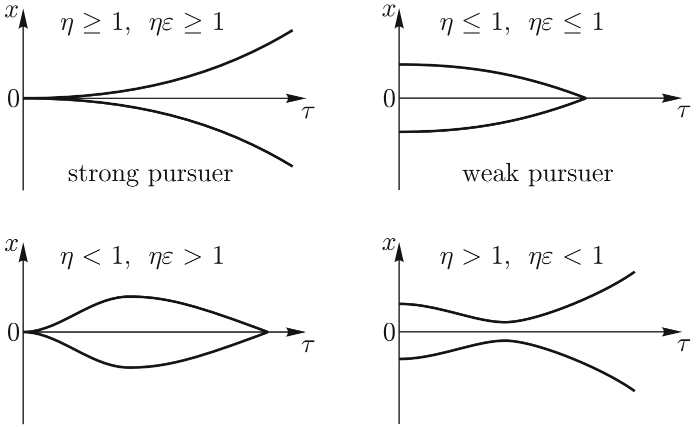

x
$\eta \geq 1,\ \eta\varepsilon \geq 1$
0
τ
strong pursuer
x
$\eta \leq 1,\ \eta\varepsilon \leq 1$
0
τ
weak pursuer
x
$\eta < 1,\ \eta\varepsilon > 1$
0
τ
x
$\eta > 1,\ \eta\varepsilon < 1$
0
τ

Fig. 2 Variants of the solvability set evolution in an individual game

For any $c \geq 0$, a level set (a Lebesgue set) $W_c = \big\{(t, x) : V(t, x) \leq c\big\}$ of the value function in game (8) can be treated as the solvability set for the considered game with the result not greater than c, that is, for a differential game with dynamics (8) and the terminal set

$$M_c = \big\{(t, x) : t = T_1,\ |x_1| \leq c\big\} \bigcup \big\{(t, x) : t = T_2,\ |x_2| \leq c\big\}.$$

When $c = 0$, one has the situation of the exact capture. The exact capture means equality to zero, at least, one of $x_1(T_1)$ and $x_2(T_2)$. Let $W_c(t) = \{x : (t, x) \in W_c\}$ be the time section (t-section) of the set W_c at the instant t. Similarly, let $M_c(t)$ for $t = T_1$ and $t = T_2$ be the t-section of the set M_c at the instant t.

Comparing dynamics capabilities of each of pursuers P_1 and P_2 and the evader E, one can introduce the parameters [9, 14] $\eta_i = \mu_i/\nu$, $\varepsilon_i = l_E/l_{P_i}$, $i = 1$, 2. They define the shape of the solvability sets in the individual games P_1–E and P_2–E. Namely, depending on values of η_i and $\eta_i\varepsilon_i$ (which are not equal to 1 simultaneously), there are 4 cases [14] of the solvability set evolution (see Fig. 2):

- expansion in the backward time (a strong pursuer);
- contraction in the backward time (a weak pursuer);
- expansion until some backward time instant and further contraction;
- contraction until some backward time instant and further expansion (if the solvability set still has not broken).

Respectively, given combinations of pursuers' capabilities in individual games and durations T_1, T_2 (equal/different), there are significant number of variants for the problem with two pursuers and one evader.

The ideology of solving the game used by us is the following. Choose the parameters η_i, ε_i, and, also, the instants T_i, $i = 1$, 2; then, using some fine grid for values of c, we compute the level sets W_c of the value function. After that, we can build optimal or quasioptimal strategies of the first and second players.

Nowadays, different workgroups suggested many algorithms for numeric solution of differential games of quite general type (see, for example, [2,3,5, 10,15]). Problem (8) has the second order on the state variable and can be rewritten as

$$\dot{x} = \mathcal{D}_1(t)u_1 + \mathcal{D}_2(t)u_2 + \mathcal{E}(t)v, \qquad |u_1| \le \mu_1,\ |u_2| \le \mu_2,\ |v| \le \nu. \tag{9}$$

Here, $x = (x_1, x_2)^{\mathrm{T}}$; vectors $\mathcal{D}_1(t)$, $\mathcal{D}_2(t)$, and $\mathcal{E}(t)$ look like

$$\mathcal{D}_1(t) = \bigl(-l_{P_1}h((T_1 - t)/l_{P_1})^{\mathrm{T}},\, 0\bigr), \qquad \mathcal{D}_2(t) = \bigl(0,\, -l_{P_2}h((T_2 - t)/l_{P_2})\bigr)^{\mathrm{T}},$$
$$\mathcal{E}(t) = \bigl(l_E h((T_1 - t)/l_E),\, l_E h((T_2 - t)/l_E)\bigr)^{\mathrm{T}}.$$

The control of the first player has two independent components u_1 and u_2. The vector $\mathcal{D}_1(t)$ ($\mathcal{D}_2(t)$) is directed along the horizontal (vertical) axis. The second player's control v is scalar. When $T_1 = T_2$, the angle between the axis x_1 and the vector $\mathcal{E}(t)$ equals 45°; when $T_1 \ne T_2$, the angle changes in time.

Due to peculiarity of our problem, we use special methods for constructing level sets of the value function.

3 Maximal Stable Bridge: Control with Discrimination

A level set W_c of the value function V is a maximal stable bridge (MSB) breaking on the terminal set M_c [7,8].

Let $T_1 = T_2$. Denote $T_f = T_1$. Using the concept of MSB from [7,8], we can say that W_c is the set maximal by inclusion in the space $t \le T_f$, x such that $W_c(T_f) = M_c(T_f)$ and the *stability* property holds: for any position $(t_*, x_*) \in W_c(t_*)$, $t_* < T_f$, any instant $t^* > t_*$, $t^* \le T_f$, any constant control v of the second player, which obeys the constraint $|v| \le \nu$, there is a measurable control $t \to \bigl(u_1(t), u_2(t)\bigr)$ of the first player, $t \in [t_*, t^*)$, $|u_1(t)| \le \mu_1$, $|u_2(t)| \le \mu_2$, guiding system (8) from the state x_* to the set $W_c(t^*)$ at the instant t^*.

The stability property assumes a discrimination of the second player by the first one: the choice of the first player's control in the interval $[t_*, t^*)$ is made after the second player announces his control in this interval.

It is known (see [7,8]) that any MSB is close. The set $W_c'(t) = \mathrm{cl}\bigl(R^2 \setminus W_c(t)\bigr)$ (the symbol cl denotes the operation of closure) is the time section of MSB W_c' for the second player at the instant t. The bridge terminates at the instant T_f on the set $M_c'(T_f) = \mathrm{cl}\bigl(R^2 \setminus M_c(T_f)\bigr)$. If the initial position of system (8) is in W_c' and if the first player is discriminated by the second one, then the second player is able to guide the motion to the set $M_c'(T_f)$ at the instant T_f. Thus, $\partial W_c = \partial W_c'$. It is proved that for any initial position $(t_0, x_0) \in \partial W_c$, the value c is the best guaranteed result for the first (second) player in the class of feedback controls.

Due to symmetry of dynamics (8) and the set $W_c(T_f)$ with respect to the origin, one gets that for any $t \leq T_f$ the time section $W_c(t)$ is symmetric also.

If $T_1 \neq T_2$, then there is no any appreciable complication in constructing MSBs for the problem considered in this paper in comparison with the case $T_1 = T_2$. Indeed, let $T_1 > T_2$. Then in the interval $(T_2, T_1]$ in (8), we take into account only the dynamics of the variable x_1 when building the bridge W_c backwardly from the instant T_1. With that, the terminal set at the instant T_1 is taken as $M_c(T_1) = \{(x_1, x_2) : |x_1| \leq c\}$. When the constructions are made up to the instant T_2, we add the set $M_c(T_2)$, that is, we take

$$W_c(T_2) = W_c(T_2 + 0) \bigcup \{(x_1, x_2) : |x_2| \leq c\},$$

and further constructions are made on the basis of this set.

So, our tool for finding a level set of the value function in game (8) corresponding to a number c is the backward procedure for constructing a MSB with the terminal set M_c. Presence of an idealized element (the discrimination of the opponent) allowed us to create effective numeric methods for backward construction of MSBs.

The solvability set with the index equal to c in the individual game $P1$–E ($P2$–E) is MSB built in the coordinates t, x_1 (t, x_2) and terminating at the instant T_1 (T_2) on the set $|x_1| \leq c$ ($|x_2| \leq c$). Its t-section, if it is non-empty, is a segment in the axis x_1 (x_2) symmetric with respect to the origin. In the plane x_1, x_2, this segment corresponds to a vertical (horizontal) strip of the same width near the axis x_2 (x_1). It is evident that when $t \leq T_1$ ($t \leq T_2$), such a strip is contained in the section $W_c(t)$ of MSB W_c of game (8) with the terminal set M_c.

4 Results of Numeric Constructions of Maximal Stable Bridges

Case of Strong Pursuers. In the case of two strong pursuers, the t-sections of MSBs in individual games $P1$–E and $P2$–E grow with increasing of the backward time. This gives that for any $c \geq 0$ and any $t \leq \bar{t} = \min\{T_1, T_2\}$ the set $W_c(t)$ includes a cross near the axes x_1, x_2, which expands with decreasing t.

Let us give results of constructing t-sections $W_c(t)$ for the following values of the game parameters: $\mu_1 = 2$, $\mu_2 = 3$, $\nu = 1$, $l_{P_1} = 1/2$, $l_{P_2} = 1/0.857$, $l_E = 1$.

Equal terminal instants. Let $T_1 = T_2 = 6$. Fig. 3 shows results of constructing the set W_0 (that is, with $c = 0$). In the figure, one can see several time sections $W_0(t)$ of this set. The bridge has a quite simple structure. At the initial instant $\tau = 0$ of the backward time (when $t = 6$), its section coincides with the target set, which is the union of two coordinate axes. Further, at the instants $t = 4$, 2, 0, the cross thickens, and two triangles are added to it. The widths of the vertical and horizontal parts of the cross correspond to sizes of MSBs in the individual games with the first and second pursuers. These triangles are located in the II and IV quadrants (where the signs of x_1 and x_2 are different, in other words, when the evader is between the pursuers). They give the zone where the exact capture is possible only under collective actions of both pursuers.

Time sections $W_c(t)$ of other bridges W_c, $c > 0$, have a shape similar to $W_0(t)$.

Different terminal instants. Let $T_1 = 7$, $T_2 = 5$. Results of constructing the set W_0 are given in Fig. 4. When $t < 5$, time sections $W_0(t)$ grow both horizontally and vertically; two additional triangles appear, but in this case they are curvilinear. In Fig. 5, the set W_0 is shown in the three-dimensional space t, x_1, x_2.

The given results are typical for the case of strong pursuers. When $T_1 = T_2$, the sets $W_c(t)$ can be described analytically. This was done in paper [9]. Also, there the case $T_1 \neq T_2$ was studied. But for it, only an upper approximation of the sets $W_c(t)$ was obtained.

Case of Weak Pursuers. Since in the case of weak pursuers the t-sections of MSBs in individual games $P1$–E and $P2$–E contract with growth of the backward time and become empty at some instant, the set $W_c(t)$ for any $c \geq 0$ with decreasing of t loses infinite sizes along axes x_1 and x_2.

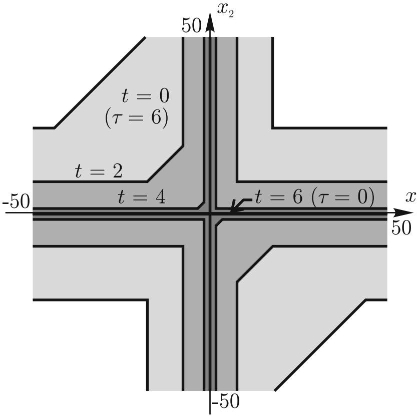


Fig. 3 Two strong pursuers, equal terminal instants: time sections of the maximal stable bridge W_0

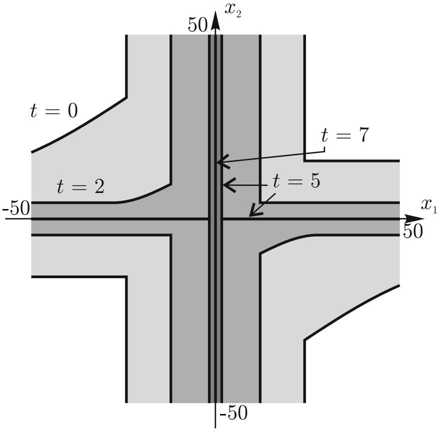


Fig. 4 Two strong pursuers, different terminal instants: time sections of the maximal stable bridge W_0

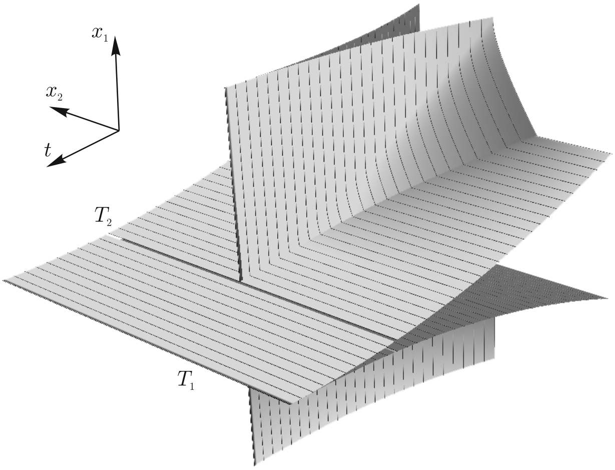


Fig. 5 Strong pursuers, different terminal instants: 3D-view of the set W_0

The most surprising fact discovered during the numeric study was that the connected set $W_c(t)$ with decreasing of t loses connectedness and disjoins into two separate parts.

Take the parameters $\mu_1 = 0.9$, $\mu_2 = 0.8$, $\nu = 1$, $l_{P_1} = l_{P_2} = 1/0.7$, $l_E = 1$. Let us show results for the case of different terminal instants only: $T_1 = 9$, $T_2 = 7$. Since in this variant the evader is more maneuverable than the pursuers, the first player cannot guarantee the exact capture.

The set W_c in the space t, x_1, x_2 for $c = 2.0$ is shown in Fig. 6. During evolution of the sections $W_{2.0}(t)$ in t, they change their structure at some instants. These places are marked by drops in the constructed surface of the set.

One Strong and One Weak Pursuers. Let us take the following parameters: $\mu_1 = 2$, $\mu_2 = 1$, $\nu = 1$, $l_{P_1} = 1/2$, $l_{P_2} = 1/0.3$, $l_E = 1$. Now the evader is more maneuverable than the second pursuer, and an exact capture by this pursuer is unavailable. Assume $T_1 = 5$, $T_2 = 7$.

In Fig. 7, a three-dimensional view of MSB $W_{5.0}$ is shown. The part along the axis x_1 of its time section $W_{5.0}(t)$ contracts with decreasing of τ, and breaks further. The part along the axis x_2 grows. After breaking the individual MSB $P2$–E (and respective collapse of the part of the cross along the axis x_1), there is a strip along the axis x_2 only with two additional parts determined by the joint actions of both pursuers.

Varying Advantage of Pursuers. Consider a variant when both pursuers P_1 and P_2 are equal, with that at the beginning of the backward time, the bridges in the individual games contract and further expand. Choose the game parameters in such a way that for some c the section $W_c(t)$ of MSB W_c

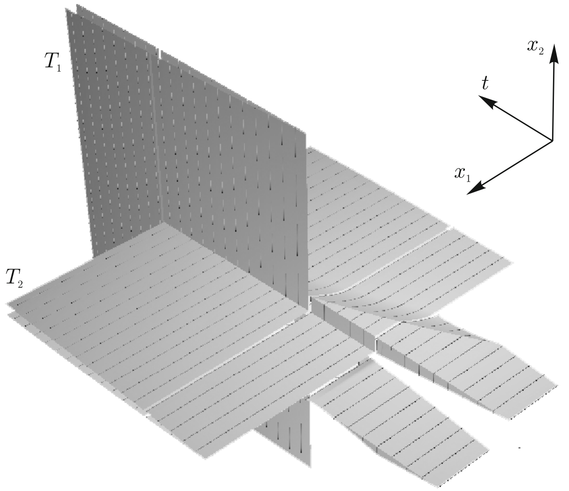


Fig. 6 Two weak pursuers, different terminal instants: 3D-view of the set $W_{2.0}$

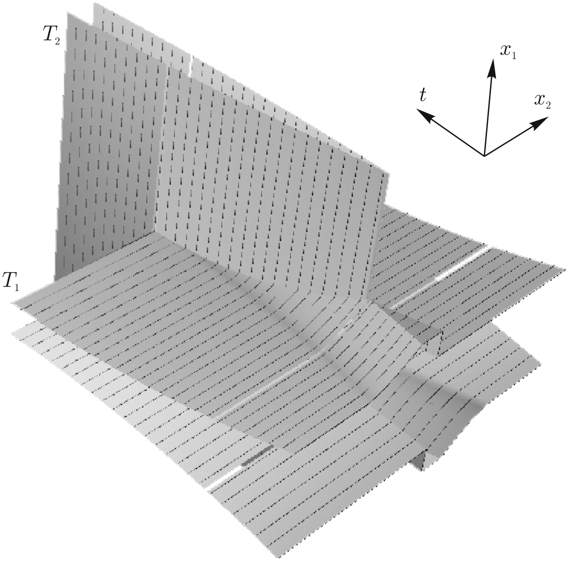


Fig. 7 One strong and one weak pursuers, different termination instants: 3D-view of the set $W_{5.0}$

with decreasing of t disjoins into two parts, which join back with further decreasing of t.

Parameters of the game are $\mu_1 = \mu_2 = 1.5$, $\nu = 1$, $l_{P_1} = l_{P_2} = 1/0.25$, $l_E = 1$. Termination instants are equal: $T_1 = T_2 = 15$.

A three-dimensional view of MSB $W_{1.315}$ is shown in Fig. 8.

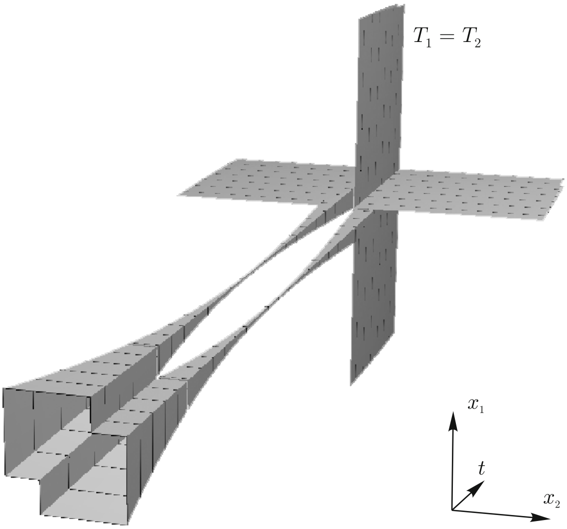


Fig. 8 Varying advantage of the pursuers, equal termination instants: 3D-view of the maximal stable bridge $W_{1.315}$

5 Control on the Basis of Switching Lines

A control based on the switching lines assumes separation of the state space x_1, x_2 to some cells at instants from some grid in time. In each cell, every scalar control keeps some extreme value. The time grid should contain intants, when a player chooses its control in a discrete scheme. Under a discrete control scheme [7,8] with the step $\varDelta$, a control chosen at the instant t_s is kept until the instant $t_{s+1} = t_s + \varDelta$. At the position $\bigl(t_{s+1}, x(t_{s+1})\bigr)$, a new control value is chosen, etc.

1) In the game under consideration, the first player has two scalar controls u_1, u_2, which are bounded by the inequalities $|u_1| \leq \mu_1$, $|u_2| \leq \mu_2$. The component of the velocity of system (9), which is affected by the control u_1, is connected to the vector $\mathcal{D}_1(t)$ and is horizontal in our case. The component corresponding to the control u_2 is connected to the vector $\mathcal{D}_2(t)$ and is directed vertically.

To separate the plane x_1, x_2 into parts, in which the control u_1 takes one of the extreme values $u_1 = +\mu_1$ or $u_1 = -\mu_1$, we study the change of the value function at the instant t in lines parallel to the vector $\mathcal{D}_1(t)$, that is, in horizontal lines.

In the problem that we investigate, the following property is true (except situations of varying advantage of the pursuers) for each horizontal line. The restriction of the value function $V(t, \cdot)$ to a horizontal line is a function having only one interval of local minimum, which is either a point, or a segment, or the entire line. With that, the restriction grows when the argument goes from the interval of minimum.

Considering an arbitrary horizontal line, we can gather the points of minimum of the restriction of the value function to this line. We take an arbitrary point from such an interval of minimum as a point for the switching line of the control u_1. Taking points from all horizontal lines in such a way, we obtain a switching line $\varPi_1(t)$ separating the plane x_1, x_2 into two parts. In the part, where the vector $\mathcal{D}_1(t)$ is directed from the switching line, we define the control u_1^* equal to $-\mu_1$, and in the another part, it is equal to $+\mu_1$. During numeric constructions, the switching line $\varPi_1(t)$ is built on the basis of some number (quite great, but finite) of time sections $W_{c_j}(t)$ of the level sets of the value function for some collection $\{c_j\}$ of values of the parameter c.

In the same way using corresponding objects, the switching line $\varPi_2(t)$ can be built for the control u_2.

The control of the first player based on the switching lines $\varPi_1(t)$ and $\varPi_2(t)$, we call *quasioptimal* because we assume that in the switching lines, the control u_1 (u_2) is taken arbitrary from the interval $[-\mu_1, +\mu_1]$ ($[-\mu_2, +\mu_2]$). For the cases of "strong" and "weak" pursuers, it can be proved that such a choice is optimal indeed. But for the case of varying advantage of the pursuers, it is possible that for some small neighborhood of the switching lines we need some additional information about the value function. The authors have not studied this question yet.

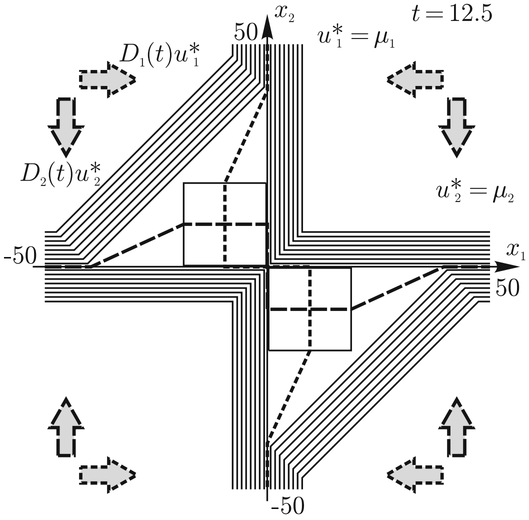


Fig. 9 The case of varying advantage of the pursuers. The typical picture of the switching lines for the first player; the dark green line is for the control u_1, the light green one is for the control u_2.

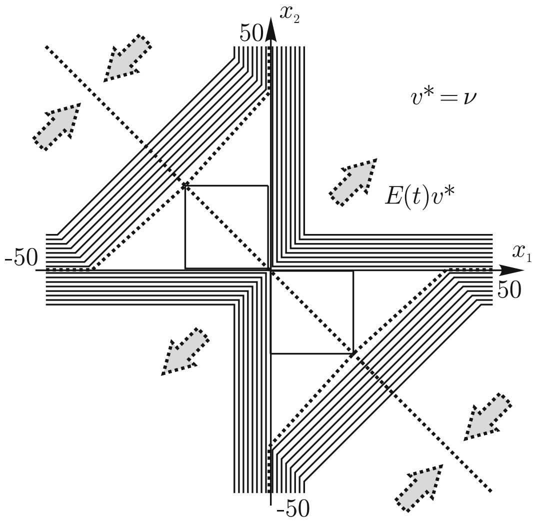


Fig. 10 The case of varying advantage of the pursuers. The typical picture of the switching lines for the second player for the same instant $t = 12.5$ as in Fig. 9.

Fig. 9 shows the typical picture of the time sections $W_c(t)$ of the level sets and switching lines $\Pi_1(t)$ and $\Pi_2(t)$ for the case of varying advantage of the pursuers.

Emphasize once more that the switching lines depend on time t, and the choice of the control is defined by the current state position of the system with respect to the corresponding switching line. The vectors $\mathcal{D}_1(t)$ and $\mathcal{D}_2(t)$ are used. Drawing a ray from the point $x(t)$ with the directing vector $\mathcal{D}_i(t)$, one can decide whether it crosses the switching line $\Pi_i(t)$. If it does not, then $u_i^*\big(t, x(t)\big) = -\mu_i$, if it crosses, then $u_i^*\big(t, x(t)\big) = +\mu_i$.

Thus, to organize computations of the discrete control scheme of the first player, we should keep in memory of the computer a collection of the switching lines in some time grid.

2) The direction of the action of the second player's scalar control v is defined by the vector $\mathcal{E}(t)$. Its direction is constant in the case $T_1 = T_2$ and changes in time if $T_1 \neq T_2$. When constructing the switching lines for the second player, we analyze points of local maxima and minima of restrictions of the value function to lines parallel to the vector $\mathcal{E}(t)$. For each of these lines, the collection of all points of minima and maxima can consists, generally speaking, of several intervals. Nevertheless, their number is small. This allows us to take corresponding points from them and to constitute some lines, which separate the plane x_1, x_2 into parts, in which the control v keeps one of its extreme values $-\nu$ or $+\nu$.

To construct $v^*\big(t, x(t)\big)$, we use the vector $\mathcal{E}(t)$. Compute how many times (even or odd) a ray with the beginning at the point $x(t)$ and the directing vector $\mathcal{E}(t)$ crosses the second player switching lines. If the number of crosses is even (absence of crosses means that the number equals zero and is even), then we take $v^*\bigl(t, x(t)\bigr) = +\nu$; otherwise, $v^*\bigl(t, x(t)\bigr) = -\nu$.

The typical picture of the switching lines of the second player is given in Fig. 10 for the case of varying advantage of the pursuers. Here, one can see 6 domains of constancy of the second player's control v. Direction of its action are shown by arrows. In the lines, which are composed of points of local maxima of the value function, the control can be taken arbitrary from the interval $[-\nu, +\nu]$. But in the lines consisting of the point of local minima, from the theoretic point of view, only extreme values $-\nu$ and $+\nu$ are allowed, which push the system from the switching line. Due to errors of numeric construction of the swiching lines, this way of control can lead to a motion in a sliding regime along the switching line (that changes in time). Such a motion can be unoptimal from the point of view of the second player. Assuming this situation to be almost impossible, we regard the suggested method of the second player's control as a quasioptimal one.

6 Optimal Motion Simulation Results

Let the pursuers P_1, P_2, and the evader E move in the plane. This plane is called the *original geometric space*. At the initial instant t_0, velocities of all objects are parallel to the horizontal axis and sufficiently larger than the possible changes of the lateral velocity components. The components of object velocities, which are parallel to the horizontal axis, are constant. Magnitudes of these components are such that the rendezvous of the objects P_1 and E happens at the instant T_1, and the objects P_2 and E encounter at the instant T_2. The dynamics of lateral motion is described by relations (1), (2); the resultant miss is given by formula (4).

The initial lateral velocities and accelerations are assumed to be zero:

$$\dot z^0_{P_1} = \dot z^0_{P_2} = \dot z^0_E = 0, \quad a^0_{P_1} = a^0_{P_2} = a^0_E = 0.$$

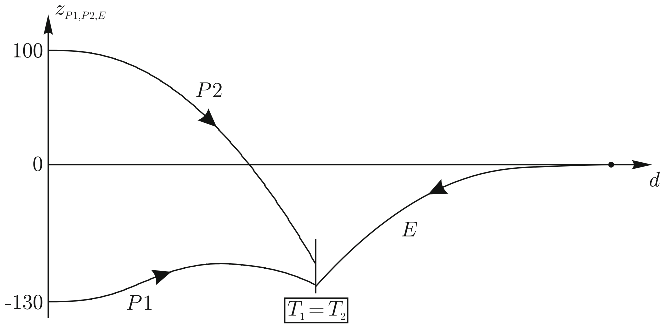


Fig. 11 Optimal trajectories in the case of varying advantage of the pursuers; large initial deviations

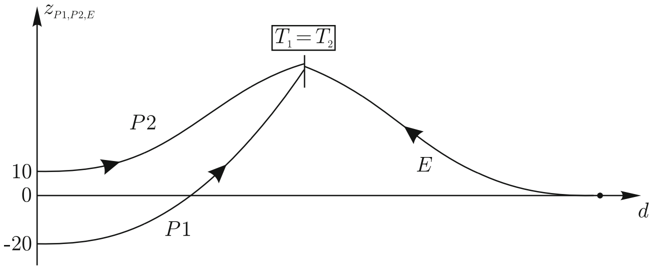


Fig. 12 Optimal trajectories in the case of varying advantage of the pursuers; small initial deviations

The simulation is made for the following parametes of the game:

$$\mu_1 = \mu_2 = 1.1,\ \nu = 1,\ l_{P_1} = l_{P_2} = 1/0.6,\ l_E = 1,\ T_1 = T_2 = 20.$$

The parameters are such that the pursuers can achieve a higher acceleration than the evader, but they are more inertial, that is, the achievement of the extreme acceleration lasts longer than the evader's one. We have

$$\eta_i = \mu_i/\nu = 1.1 > 1,\ \eta_i\varepsilon_i = \eta_i \cdot \frac{l_E}{l_{P_i}} = 1.1 \cdot 0.6 = 0.66 < 1,\ i = 1,\ 2.$$

So, we consider the case of varying advantage of the pursuers. In this situaton, the exact capture is not guaranteed.

In Figs. 11 and 12, the horizontal axis is denoted by the symbol d. The coordinate d shows the longitudinal position of the objects. Controls of the objects affect the vertical (lateral) coordinate.

Fig. 11 shows the optimal trajectories of the objects for the following initial positions at the instant $t_0 = 0$:

$$z^0_{P_1} = -130,\ z^0_{P_2} = 100,\ z^0_E = 0.$$

The initial deviations are so large that the second pursuer (the upper one) is unable reach the evader, even applying its extremal control. But the first pursuer (the lower one) has a quite small miss, which, nevertheless, is still non-zero.

In Fig. 12, the optimal trajectories are given for the initial positions

$$z^0_{P_1} = -20,\ z^0_{P_2} = 10,\ z^0_E = 0.$$

Now, both pursuers have small terminal misses, but they are non-zero due to the advantage of the evader at the final stage of the pursuit. Note that the evader is just in the middle between the pursuers at the instant $T_1 = T_2$: such a position provides the maximal possible payoff value for him.

7 Conclusion

For a model zero-sum differential game with two pursuing ond one evading objects, a numeric solution is obtained: the level sets of the value function, quasioptimal strategies on the basis of switching lines, simulation of motions using the suggested strategies. A complete investigation of the problem can be made because the original formulation allows an equivalent presentation with two-dimensional state vector in the plane of coordinates of one-dimensional forecasted misses (zero-effort miss coordinates). Similar problems are much harder if the miss between each pursuer and the evader are computed in a two-dimensional geometric space.

Acknowledgements. This work was supported by Program of Presidium RAS "Dynamic Systems and Control Theory" under financial support of UrB RAS (project No.12-*П*-1-1002) and also by the Russian Foundation for Basic Research under grant no.12-01-00537.

References

1. Blagodatskih, A.I., Petrov, N.N.: Conflict Interaction Controlled Objects Groups. Udmurt State University, Izhevsk (2009) (in Russian)
2. Botkin, N.D., Hoffmann, K.H., Turova, V.L.: Stable numerical schemes for solving Hamilton–Jacobi–Bellman–Isaacs equations. SIAM Journal on Scientific Computing 33(2), 992–1007 (2011)
3. Cardaliaguet, P., Quincampoix, M., Saint-Pierre, P.: Set-valued numerical analysis for optimal control and differential games. In: Bardi, M., Raghavan, T.E., Parthasarathy, T. (eds.) Annals of the International Society of Dynamic Games. Stochastic and Differential Games — Theory and Numerical Methods, vol. 4, pp. 177–247. Birkhauser, Boston (1999)
4. Chikrii, A.A.: Conflict-Controlled Processes. Mathematics and its Applications, vol. 405. Kluwer Academic Publishers Group, Dordrecht (1997)
5. Cristiani, E., Falcone, M.: Fully-discrete schemes for the value function of pursuit-evasion games with state constraints. In: Bernhard, P., Gaitsgory, V., Pourtallier, O. (eds.) Annals of the International Society of Dynamic Games. Advances in Dynamic Games and Applications, vol. 10, pp. 177–206. Birkhauser, Boston (2009)
6. Grigorenko, N.L.: The problem of pursuit by several objects. In: Hämäläinen, R.P., Ehtamo, H.K. (eds.) Differential Games — Developments in Modelling and Computation (Espoo, 1990). LNCIS, vol. 156, pp. 71–80. Springer, Berlin (1991)
7. Krasovskii, N.N., Subbotin, A.I.: Positional Differential Games. Nauka, Moscow (1974) (in Russian)
8. Krasovskii, N.N., Subbotin, A.I.: Game-Theoretical Control Problems. Springer, New York (1988)

9. Le Ménec, S.: Linear differential game with two pursuers and one evader. In: Breton, M., Szajowski, K. (eds.) Annals of the International Society of Dynamic Games. Advances in Dynamic Games. Theory, Applications, and Numerical Methods for Differential and Stochastic Games, vol. 11, pp. 209–226. Birkhauser, Boston (2011)
10. Mitchell, I.: Application of level set methods to control and reachability problems in continuous and hybrid systems. Ph.D. thesis, Stanford University (2002)
11. Petrosjan, L.A.: Differential Games of Pursuit. Leningrad University, Leningrad (1977) (in Russian)
12. Pschenichnyi, B.N.: Simple pursuit by several objects. Kibernetika 3, 145–146 (1976) (in Russian)
13. Shima, T., Shinar, J.: Time-varying linear pursuit-evasion game models with bounded controls. Journal of Guidance, Control and Dynamics 25(3), 425–432 (2002)
14. Shinar, J., Shima, T.: Non-orthodox guidance law development approach for intercepting maneuvering targets. Journal of Guidance, Control, and Dynamics 25(4), 658–666 (2002)
15. Taras'ev, A.M., Tokmantsev, T.B., Uspenskii, A.A., Ushakov, V.N.: On procedures for constructing solutions in differential games on a finite interval of time. Journal of Mathematical Sciences 139(5), 6954–6975 (2006)

Nonlinear Output-Feedback H_∞ Control for Spacecraft Attitude Control

Alon Capua, Nadav Berman*, Amir Shapiro, and Daniel Choukroun

Abstract. In this paper, a novel computational scheme is proposed in order to solve the output-feedback H_∞ control problem for a class of nonlinear systems with polynomial vector field. By converting the resulting Hamilton-Jacobi inequalities from rational forms to their equivalent polynomial forms, we overcome the non-convex nature and numerical difficulty. Using quadratic Lyapunov functions, both the state-feedback and output-feedback problems are reformulated as semi-definite optimization conditions, while locally tractable solutions can be obtained through sum of squares (SOS) programming. A numerical example shows that the proposed computational scheme results in a better disturbance attenuation closed-loop system, as compared to standard methods, by using classical quadratic Lyapunov functions. The novel methodology is applied in order to develop a robust spacecraft attitude regulator.

1 Introduction

In the past decade, there has been substantial interest in H_∞ control of nonlinear systems [23, 8]. Interpreting nonlinear H_∞ control in terms of dissipativity and

Alon Capua · Nadav Berman · Amir Shapiro
Department of Mechanical Engineering,
Ben Gurion University of the Negev, Beer Sheva , Israel
e-mail: {aloncap,ashapiro}@post.bgu.ac.il

Daniel Choukroun
Department of Mechanical Engineering,
Ben Gurion University of the Negev,
Beer Sheva , Israel, on a leave from Delft University of Technology,
Faculty of Aerospace Engineering, Delft, Netherlands
e-mail: d.choukroun@tudelft.nl

* Nadav Berman passed away on April 21, 2012. Professor Berman was a very active member of the Israeli community of control scientists he was an admired figure and an exceptional supervisor who we dearly miss.

differential game [2, 24] where the solution has been related to an appropriate Hamilton-Jacobi inequality. For hyperbolic nonlinear systems whose linearized plant is stabilizable, the solution of the Hamilton-Jacobi inequality was characterized by an invariant manifold of Hamiltonian vector fields using differential geometric theory [23, 24].

In linear systems, it is well known that the Hamilton-Jacobi partial differential inequality reduces to the Riccati inequality, which can be solved easily by efficient numerical algorithms. However, in the nonlinear case, there is no systematic numerical algorithm currently available for the solution of this partial differential inequality. Therefore, the key difficulty of nonlinear H_∞ control theory is the solvability of the Hamilton-Jacobi inequality. To this end, various approaches have been proposed to solve the Hamilton-Jacobi inequality numerically. One of the suggested methods is a Taylor series expansion of the storage function [10, 29], in an iterative fashion, provided that the linearized model of the nonlinear system has a solution. However, a numerically efficient solution remains an unsolved issue [1].

Isidori [8] showed, that the solution to the output-feedback control problem is determined by a pair of coupled Hamilton-Jacobi inequalities. Parallel to linear H_∞ control theory, a separation principle was also established under a detectability hypothesis [9]. Obviously, there are major advantages of the output-feedback problem for continuous-time nonlinear systems over linear systems [1], despite the fact that the output-feedback problem for nonlinear systems has not been studied as widely as for linear systems. Although there are studies of the static output-feedback for nonlinear systems, the dynamic output-feedback for nonlinear systems was studied much less; one of the reasons is the non-complex structure rather than the dynamic output-feedback case. In addition, it preserves the controllers structure, based on the physical intuition from the actual system. Yet, the dynamic output-feedback results in high order controllers [8] which are more accurate. The dynamic output-feedback problem has been investigated while parameterized as a nonlinear fractional transformation on locally contractive and stable nonlinear operators [12]. A solution based on allowing nonhyperbolic equilibria for the Hamiltonian systems associated with the two Hamilton-Jacobi-Isaacs equations: the state-feedback and, respectively, output-injection design problems are presented in [8, 25]. However, the solutions from these approaches do not have a closed form and therefore may not converge to an analytic solution, due to their non-convex nature.

A recent computational relaxation based on the sum of squares (SOS) decomposition for multivariable polynomials and semidefinite programming [16, 4] provides potentially effective ways for the analysis and synthesis of nonlinear systems. In nonlinear system design, the verification of the non-negativity of the Lyapunov conditions is a complex task. However, the new computationally tractable analysis methodology provides a new way of searching for SOS decomposition to relax the original problem. This crucial property of the SOS based methodology finds applications successfully in several nonlinear control problems. For example, the stability analysis and synthesis problem have been studied in [19, 3, 18, 27] for nonlinear systems. In [31] local stability analysis was considered, and the region of attraction inner-bound enlargement problem was presented for polynomial systems with uncertain dynamics. A semidefinite programming approach based on state dependent inequalities is proposed in [17] to obtain global stability and performance objective by using quadratic Lyapunov functions.

As a result, a convex parametrization of the nonlinear H_∞ control problem was derived in [13] based on a pair of positive definite matrix functions. Prempain [21] formulated the $\mathscr{L}_2$-gain analysis problem for polynomial nonlinear systems as a convex state-dependent LMI, which can be recasted as a SOS optimization problem. This approach was shown promising to overcome the numerical difficulty in solving the Hamilton-Jacobi inequality and provides an analytic solution at the same time. Wei *et al.* [32] proposed an iterative method based on SOS programming [18], [7] to solve a special case of the state-feedback H_∞ control problem. As a powerful and promising technique, SOS programming has also been applied to solve nonlinear analysis [15], [28] and stabilization [17], [20] problems. The main advantages of SOS decomposition are the resulting computational tractability and the algorithmic characteristics of the solution procedure [16]. This could help to provide coherent methodology of synthesizing Lyapunov functions for nonlinear systems. In addition, the importance of SOS technique also lies in its ability to provide tractable relaxations for many difficult optimization problems, such as the nonlinear output-feedback H_∞ controller.

Motivated by all of these developments, we propose a computational scheme for solving the nonlinear dynamic output-feedback design problems for a class of affine nonlinear systems. Moreover, the resulting output feedback controller will be constructed to achieve closed-loop stability as well as $\mathscr{L}_2$-gain performance. Specifically, we use polynomial type Lyapunov functions to convert the original Hamilton-Jacobi inequalities into linear matrix inequalities for polynomial nonlinear systems. As a result, the numerical difficulty in solving the nonlinear H_∞ output-feedback problem is overcome, and the output-feedback controllers and Lyapunov functions are constructed in an efficient computational manner.

Spacecraft attitude control is a critical function in any space mission. The development of nonlinear spacecraft attitude control algorithms has been following many paths over the last four decades, from Lyapunov-based regulator [14], nonlinear adaptive control [22], dynamic inversion [5], optimization [26], model predictive control [11], to sliding mode control, State-Dependent-Riccati-Equation (SDRE) control [6], and H_∞ control [30]. Applying the proposed novel methodology, a robust attitude controller will be developed in the final manuscript under the assumptions of rigid body dynamics, three-axis control authority, and full state information. Using the quaternion of rotation and the angular velocity vector as state variables yields a polynomial structure of the dynamical model, enabling the novel H_∞ control design. Particular attention will be given to the quaternion properties, i.e., non-uniqueness with regard to attitude and norm unity.

2 Sum of Squares

A basic problem that appears in many areas of mathematics is that of checking global non-negativity of a function of several variables. In particular, the problem is to establish equivalent conditions or a procedure for checking the validity of:

$$F(x_1,\dots,x_n) \geq 0, \quad \forall x_1,\dots,x_n \in \mathbb{R} \tag{1}$$

A polynomial $F(x) \in \mathbb{R}[x]$ is said to be nonnegative or positive semidefinite (PSD) if $F(x) \geq 0 \; \forall \; x \in \mathbb{R}^n$. Clearly, a necessary condition for a polynomial to be PSD is that its total degree be even. We say that $F(x)$ is sum of squares (SOS), if there exist polynomials $f_1(x), \dots f_m(x)$ such that:

$$F(x) = \sum_{i=1}^{m} f_i^2(x) \tag{2}$$

It is clear that $F(x)$ being SOS implies that $F(x)$ is PSD. We define a function $q : \mathbb{R}^n \to \mathbb{R}$ as a *monomial* if:

$$q(x) = c_a x^a, \quad x \in \mathbb{R}^n, c_a \in \mathbb{R}, a \in \mathbb{N}^n \tag{3}$$

such that $q(x) = c_a \left(x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}\right)$. Defining a function $p = \sum_{i=1}^{r} q_i(x)$ to be polynomial if it is a sum of monomials $q_1, q_2, \dots, q_r$:$\mathbb{R}^n \to \mathbb{R}$ with finite degree. The largest degree of the monomials $q_1, q_2, \dots, q_r$ is defined to be the degree of p. A set of polynomials $p : \mathbb{R}^n \to \mathbb{R}$ is denoted by $\mathscr{P}$, where the polynomial with the largest degree defines the degree of the family $\mathscr{P}$. We define $x^{\{d\}} \in \mathbb{R}^{\sigma(n,d)}$ with $x \in \mathbb{R}^n$ as a *vector of monomials* for the polynomials in $\mathscr{P}$ of degree d, as a basis of $\mathscr{P}$, where $\sigma(n,d)$ is defined as, $\sigma(n,d) = \frac{(n+d-1)!}{(n-1)!d!}$ in n scalar variables. The basic idea of the method is the following: express the given polynomial as a quadratic form in some new variables $x^{\{d\}}$. These new variables are the original x ones, plus all the monomials of degree less than or equal to $\frac{d}{2}$, given by the different products of the x variables, where d is the degree of the polynomial. Therefore, $F(x)$ can be represented as:

$$F(x) = x^{\{d\}^T} Q x^{\{d\}} \tag{4}$$

where Q is a constant matrix called the Gramian matrix, not necessarily unique. The following representation is also called the square matrix representation (SMR). If in the representation above Q is positive semidefinite, then $F(x)$ is also positive semidefinite. Notice that in the case of quadratic forms, for instance, the two conditions (nonegativity and sum of squares) are equivalent. The problem of checking if a given polynomial may be written as a sum of squares can be solved via convex optimization, in particular semidefinite programming. SOSTOOLS a free, third party MATLAB toolbox provides a way of finding sum of squares, over an affine family of polynomials. For instance, it can be used in the computation of Lyapunov functions for proving stability of nonlinear systems.

3 The Nonlinear H_∞ Problem

Considering the following system where the plant is represented by an affine causal state space system defined on a smooth n-dimensional manifold $\mathscr{X} \subseteq \mathbb{R}^n$ in local coordinates $x = (x_1, \dots, x_n)$:

$$\Sigma: \begin{array}{l} \dot{x} = f(x) + g_1(x)w + g_2(x)u \\ y = x \\ z = h_1(x) + k_{12}(x)u, \quad z \in \mathbb{R}^s \end{array} \tag{5}$$

with two sets of inputs u and w and two sets of outputs y and z. Where $x \in \mathscr{X}$ is the state vector, $u \in \mathscr{U} \subseteq \mathbb{R}^p$ is the p-dimensional control input, which belongs to the set of admissible controls $\mathscr{U}$, $w \in \mathscr{W}$ is the disturbance signal, which belongs to the set $\mathscr{W} \subset \mathscr{L}_2([t_0, \infty), \mathbb{R}^r)$ of admissible disturbances. The output $y \in \mathbb{R}^n$ is the states vector of the system which is measured directly, and $z \in \mathbb{R}^s$ is the output to be controlled. The functions $f: \mathscr{X} \to C^\infty(\mathscr{X})$, $g_1: \mathscr{X} \to \mathscr{M}^{n\times r}(\mathscr{X})$, $g_2: \mathscr{X} \to \mathscr{M}^{n\times p}(\mathscr{X})$, $h_1: \mathscr{X} \to \mathbb{R}^s$, and $k_{12}: \mathscr{X} \to \mathscr{M}^{s\times p}(\mathscr{X})$ are assumed to be real C^∞-functions of x. The H_∞ control problem, is described as finding a controller $K(x)$ which produces a control input such that in the closed-loop configuration satisfies,

$$\int_0^\infty ||z(t)||^2 dt \leq \gamma^2 \left[||x_0||^2 + \int_0^\infty ||w(t)||^2 dt \right], \qquad \forall w \in \mathscr{L}_2 \tag{6}$$

then we can say that the closed loop system has an $\mathscr{L}_2$-gain$\leq \gamma$. Furthermore, the closed-loop system should be stable.

A state-space system Σ is said to be dissipative with respect to the supply rate s if there exists a function $S: X \to \mathbb{R}^+$, called the storage function, such that for all $x_0 \in \mathscr{X}$, all $t_1 \geq t_0$, and all disturbances $w \in \mathscr{L}_2$.

$$S(x(t_1)) \leq S(x(t_0)) + \int_{t_0}^{t_1} s(w(t), z(t)) dt \tag{7}$$

The latter inequality is called the dissipation inequality. It expresses the fact that the "stored energy" $S(x(t_1))$ of Σ at any future time t_1 is, at most, equal to the sum of the stored energy $S(x(t_0))$ at the present time t_0 and the total externally supplied energy is, $\int_{t_0}^{t_1} s(w(t), z(t)) dt$, during the time interval $[t_0, t_1]$. Hence, there can be no internal "creation of energy", only internal dissipation of energy is possible.

By choosing a supply rate:

$$s(w, z) = \frac{1}{2}\gamma^2 \|w\|^2 - \frac{1}{2} \|z\|^2, \quad \gamma \geq 0 \tag{8}$$

Σ is dissipative with respect to this supply rate if and only if there exists $S \geq 0$ such that for all $t_1 \geq t_0$, $x(t_0)$ and u valid the following:

$$\frac{1}{2} \int_{t_0}^{t_1} \left(\gamma^2 \|w\|^2 - \|z\|^2 \right) dt \geq S(x(t_1)) - S(x(t_0)) \tag{9}$$

It follows that the system Σ has $\mathscr{L}_2$-gain $\geq \gamma$ if it is dissipative with respect to the supply rate $s(w,z) = \frac{1}{2}\left(\gamma^2 \|w\|^2 - \|z\|^2\right)$.

We will consider a storage functions S as C^1 functions. By letting $t_1 \to t_0$ we see that (7) is equivalent to:

$$S_x \dot{x} \leq s(w, z(x,u)), \quad \forall x, u \tag{10}$$

with $S_x(x)$ denoting the vector of the partial derivatives $S_x(x) = \left(\frac{\partial S}{\partial x_1}(x), ..., \frac{\partial S}{\partial x_n}(x)\right)$. Furthermore, one can establish a direct link between dissipativity and Lyapunov stability. Assume now that $x^* \in \mathscr{X}$ is a strict local minimum of S. Then x^*, is a stable equilibrium of the unforced system $\dot{x} = f(x)$, i.e. $w = 0, u = 0$, with Lyapunov function $V(x) = S(x) - S(x^*) \geq 0$, for x around x^* [24]. According to (10) we can write for the above system as the following dissipation inequality:

$$V_x\left(f(x) + g_1(x)w + g_2(x)u\right) - \frac{1}{2}\gamma^2 \|w\|^2 + \frac{1}{2}\|z(x,u)\|^2 \leq 0 \tag{11}$$

maximizing with respect to w results in $w^* = \frac{1}{\gamma^2} g_1^T V_x^T$ while minimizing with respect to u results in $u^* = -g_2^T V_x^T$. Substituting these into the above inequality and assuming that $h_1(x)^T k_{12}(x) = 0$, yields the Hamilton Jacobi inequality (HJI):

$$\begin{aligned} HJI: \ V_x f(x) + \frac{V_x}{2}\left(\frac{1}{\gamma^2} g_1(x) g_1(x)^T - g_2(x) k_{12}(x)^T k_{12}(x) g_2(x)^T\right) V_x^T \\ + \frac{1}{2} h_1(x)^T h_1(x) \leq 0 \end{aligned} \tag{12}$$

which needs to be satisfied for all $x \in \mathscr{X}$. Thus, if exists a $V \geq 0$ which satisfies the latter inequality, then it is said that Σ has an $\mathscr{L}_2$-gain $\leq \gamma$. Therefore, sufficient condition for a system to have $\mathscr{L}_2$-gain is the existence of a controller $u(x) = K(x)$ which renders a dissipative closed loop system. By taking $t_0 = 0$ and assuming that $V(x(0)) \leq \gamma^2 \|x(0)\|^2$ then the dissiptivity implies that $\mathscr{L}_2$-gain $\leq \gamma$.

3.1 Sum of Square Based Nonlinear H_∞ State-Feedback

Consider the following input-affine nonlinear time invariant system which is in a state dependent linear-like representation:

$$\begin{aligned} \dot{x} &= A(x)x^{\{d\}} + B_1(x)w + B_2(x)u \\ z &= C_1(x)x^{\{d\}} + D_{12}(x)u \\ y &= x \end{aligned} \tag{13}$$

where $x^{\{d\}}$ is an $N \times 1$ vector of monomials in x satisfying the following

Assumption 1. $x^{\{d\}} = 0$ *iff* $x = 0$

Remark 1. It should be noted that, given $f(x), h_1(x) \in \mathscr{P}^n$, the representation $f(x) = A(x)x^{\{d\}}$ and $h_1(x) = C_1(x)x^{\{d\}}$ is highly non-unique. Notice that for any $E(x)$ with $E(x)x^{\{d\}} = 0$, $A(x) + E(x)$ can also be used as a representation for $f(x)$. A special case of the representation corresponds to $x^{\{d\}} = x$, while $x^{\{d\}}$ can be selected to contain all the monomials in $f(x)$, i.e. $A(x)$ becomes a constant matrix.

Let $M(x)$ be a $N \times n$ polynomial matrix whose $(i,j)^{th}$ entry are given by

$$M_{ij} = \frac{\partial x_i^{\{d\}}}{\partial x_j}, \quad i = 1,\ldots,N, \; j = 1,\ldots,n \tag{14}$$

Assumption 2. $C_1^T(x)D_{12}(x) = 0$ *and* $R_2(x) = D_{12}^T(x)D_{12}(x) > 0$

Theorem 1. *Consider system (13), if exists $X = X^T > 0$ and $Y(x)$ such that the following linear matrix inequality is satisfied while minimizing γ*

$$\begin{bmatrix} \begin{array}{c} Y^T(x)B_2^T(x)M^T + M(x)B_2(x)Y(x) \\ +XA^T(x)M^T + M(x)A(x)X \end{array} & M(x)B_1(x) & Y^T(x) & XC_1^T(x) \\ * & -\gamma^2 I & 0 & 0 \\ * & * & -R_2(x) & 0 \\ * & * & * & -I \end{bmatrix} \leq 0, \tag{15}$$

then the control law $u = K(x)x^{\{d\}}$ stabilizes the system and achieves the H_∞ performance $\|z(x)\|_2 \leq \gamma \|w(x)\|_2$ with

$$K(x) = Y(x)X^{-1} \tag{16}$$

where $*$ indicates symmetric entries in a symmetric matrix.

Proof. Considering the closed-loop system of (13), a storage function $V(x) = x^{\{d\}T}(x)Px^{\{d\}}$ and controller matrix (16), then, according to the dissipation inequality (7) we obtain,

$$\begin{aligned} & A^T(x)M^TP + P(x)M(x)A(x) + PY^T(x)B_2^T(x)M^TP + P(x)M(x)B_2(x)Y(x)P \\ & + \frac{1}{\gamma^2}PM(x)B_1(x)B_1^T(x)M^TP + C_1^T(x)C_1(x) + PY^T(x)D_{12}^T(x)D_{12}(x)Y(x)P \leq 0 \end{aligned} \tag{17}$$

multiplying both sides by $X = P^{-1}$, and applying the schur complement, then with the zero initial condition, the system is stable and the H_∞ performance is achieved as $\|z(x)\|_2 \leq \gamma \|w(x)\|_2$ with (16). □

4 Nonlinear H_∞ Output-Feedback

For the output-feedback suboptimal H_∞ control problem one wants to construct, if possible, for a given attenuation level $\hat{\gamma} \geq 0$ an output-feedback controller. We begin by synthesizing a dynamic observer-based controller using the output measurements. As before we consider an affine causal state space system defined on a smooth n-dimensional manifold $\mathscr{X} \subseteq \mathbb{R}^n$ in local coordinates $x = (x_1, \dots, x_n)$:

$$\begin{aligned} \dot{x} &= f(x) + g_1(x)w_1 + g_2(x)u \\ z &= h_1(x) + k_{12}(x)u \\ y &= h_2(x) + k_{21}(x)w_2 \end{aligned} \tag{18}$$

the output $y \in \mathscr{Y} \subset \mathbb{R}^m$ is the measured output of the system, $h_2 : \mathscr{X} \to \mathbb{R}^m$ and $k_{21} : \mathscr{X} \to \mathscr{M}^{m\times r}(\mathscr{X})$ are assumed to be real C^∞-functions of x. The estimator and control law are modeled as

$$\begin{aligned} &\dot{\xi} = f(\xi) + g_1(\xi)w_1 + g_2(\xi)u + G(\xi)[y - h_2(\xi) - k_{21}(\xi)w_2] \\ &u = \alpha_2(\xi), \ \ \alpha_2(0) = 0 \end{aligned} \tag{19}$$

Substituting into the observer the optimal control law $u^* = \alpha_2(\xi)$, obtained from the state-feedback problem and the worst disturbance $w_2^* = \alpha_1(\xi)$, obtained as well from the state-feedback problem . Results in the following matrix formed dynamical equations

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{X}} = \underbrace{\begin{bmatrix} f(x) + g_1(x)\alpha_1(x) + \cdots \\ G(\xi)\,[h_2(x) + k_{21}(x)\alpha_1(x) - h_2(\xi) - k_{21}(\xi)\alpha_1(\xi)] + \cdots \end{bmatrix}}_{F(x,\xi)} \\ \begin{matrix} +g_2(x)\alpha_2(\xi) \\ +f(\xi) + g_1(\xi)\alpha_1(\xi) + g_2(\xi)\alpha_2(\xi) \end{matrix} \Bigg] + \underbrace{\begin{bmatrix} g_1(x) \\ G(\xi)k_{21}(x) \end{bmatrix}}_{g(x,\xi)} (w - \alpha_1(x)) \end{aligned} \tag{20}$$

Similar to the case of the state-feedback, dissipativity results in,

$$V_x\dot{x} + \|z\|^2 - \gamma^2\|w\|^2 = HJI + \|u - \alpha_2(x)\|^2_{R_2(x)} - \gamma^2\|w - \alpha_1(x)\|^2 \tag{21}$$

where the latter inequality can be written as,

$$V_x(f(x) + g_1(x)w + g_2(x)\alpha_2(\xi)) + \|z\|^2 - \gamma^2\|w\|^2 \leq \|v\|^2_{R_2(x)} - \gamma^2\|r\|^2 \tag{22}$$

where, $v = u - \alpha_2(x)$, $R_2(x) = k_{12}^T(x)k_{12}(x)$ and $r(x) = w - \alpha_1(x)$. Implementing the above supply rate such that the $\mathscr{L}_2$-gain will be sustained for the nonnegative C^1 storage function $W(X)$ yields,

$$W_X[F(X) + g(X)r] \leq \hat{\gamma}^2\|r\|^2 - \|v\|^2_{R_2(x)}\,. \tag{23}$$

While, substituting the essential supremum of $r(x)$ into the (23) results in the Hamilton Jacobi inequality for the unified system,

$$W_X F(X) + \frac{1}{4\gamma^2} W_X g(X) g^T(X) W_X^T + v^T(X) R_2(x) v(X) \leq 0. \tag{24}$$

This approach has essentially two disadvantages. The Hamilton Jacobi has twice as many independent variables as that of the state-feedback Hamilton Jacobi. The second disadvantage is the fact that the inequality is not convex since $G(\xi)$ is a design parameter. An alternative set of sufficient conditions for the solution of the problem are proposed in order to solve the problem of disturbance attenuation via measurement-feedback. The solution is based on an additional Hamilton Jacobi inequality which has the same number of independent variables as the Hamilton Jacobi inequality for the state-feedback problem. Assuming $W(X) = Q(x - \xi)$ we have,

$$\begin{aligned} HJI_q: \quad & Q_x[\hat{f}(x) - G(x)\hat{h}(x)] + \alpha_2^T(\xi) R_2(x) \alpha_2(\xi) \\ & + \frac{1}{4\hat{\gamma}^2} Q_x[g_1(x) - G(x)k_{21}(x)][g_1(x) - G(x)k_{21}(x)]^T Q_x^T \leq 0 \end{aligned} \tag{25}$$

where, $\hat{f}(x) = f(x) + g_1(x)\alpha_1(x)$, $\hat{h}(x) = h_2(x) + k_{21}(x)\alpha_1(x)$.

By completion to square of the HJI_q we obtain,

$$Q_x f_0(x) + \frac{1}{4\hat{\gamma}^2} Q_x g_0(x) g_0^T(x) Q_x^T + T_0 \leq 0 \tag{26}$$

where,

$$\begin{aligned} f_0(x) &= \hat{f}(x) - g_1(x) k_{21}^T R_1^{-1}(x) \hat{h}(x) \\ T_0(x) &= \alpha_2^T(x) R_2(x) \alpha_2(x) - \gamma^2 \hat{h}^T(x) R_1^{-1}(x) \hat{h}(x) \\ g_0(x) &= g_1(x)[I - k_{21}^T R_1^{-1}(x) k_{21}(x)] \\ R_1(x) &= k_{21}^T(x) k_{21}(x) \end{aligned} \tag{27}$$

This is valid if and only if

$$Q_x G(x) = [2\hat{\gamma}^2 \hat{h}^T(x) + Q_x g_1(x) k_{21}^T(x)] R_1^{-1}(x) \tag{28}$$

so that,

$$G(x) = \left(2\hat{\gamma}^2 L(x) + g_1(x) k_{21}^T(x)\right) R_1^{-1}(x) \tag{29}$$

if and only if Q satisfies $\hat{h}^T(x) = Q_x L(x)$, for some matrix $L(x)$ of smooth function of x.

4.1 Sum of Square Based Nonlinear H_∞ Output-Feedback

Consider the following input-affine nonlinear time invariant system which is in the state dependent linear-like representation:

$$\begin{aligned} \dot{x} &= A(x)x^{\{d\}} + B_1(x)w_1 + B_2(x)u \\ z &= C_1(x)x^{\{d\}} + D_{12}(x)u \\ y &= C_2(x)x^{\{d\}} + D_{21}(x)w_2 \end{aligned} \tag{30}$$

where the dynamics of the estimator describes as,

$$\dot{\xi} = A(\xi)\xi^{\{d\}} + B_1(\xi)w_1 + B_2(\xi)u + G(\xi)[y - C_2(\xi)\xi^{\{d\}} - D_{21}(\xi)w_2] \tag{31}$$

Assumption 3. *The system matrices are such that* $R_1(x) = D_{21}^T(x)D_{21}(x) > 0$ *and* $D_{21} : \mathscr{X} \to \mathscr{M}^{m\times m}(\mathscr{X})$, $\mathscr{W} \subset \mathscr{L}_2([t_0,\infty),\mathbb{R}^m)$ *or* $D_{21} : \mathscr{X} \to \mathscr{M}^{m}(\mathscr{X})$, $\mathscr{W} \subset \mathscr{L}_2([t_0,\infty),\mathbb{R})$.

Theorem 2. *Consider system (30), if exists* $T = T^T > 0$ *, such that the following linear matrix inequality is satisfied while minimizing* $\hat{\gamma}$

$$\begin{bmatrix} \begin{matrix} A_0^T(x)M^T(x)T + TM(x)A_0(x) \\ -\gamma^2 C_2^T(x)R_1^{-1}(x)C_2(x) \end{matrix} & PMB_2(x) & PMB_1(x)D_{21}^T(x) & TM(x)\hat{B}_1(x) \\ * & -R_2(x) & 0 & 0 \\ * & * & -\gamma^2 R_1(x) & 0 \\ * & * & * & -\hat{\gamma}^2 I \end{bmatrix} \leq 0 \tag{32}$$

then the measurement-feedback nonlinear H_∞ control problem for the system is solvable with the controller (16), (31) iff G(.) is selected as

$$G(x) = \left(2\hat{\gamma}^2 L(x) + B_1(x)D_{21}^T(x)\right) R_1^{-1}(x) \tag{33}$$

for some $n \times m$ smooth C^2 matrix function $L(x)$ which satisfies the condition

$$(M^T(x)T^{-1}x^{\{d\}} + x^{\{d\}T}T^{-1}M(x))L(x) = \hat{C}^T(x) \tag{34}$$

Where P, γ are obtained from the solution of the state-feedback problem (13), and $A_0(x)$, $\hat{B}_1(x)$, $\hat{C}^T(x)$ are defined as,

$$\begin{aligned} A_0(x) &= A(x) + \frac{1}{\gamma^2}B_1(x)B_1^T(x)M^T P - B_1(x)D_{21}^T(x)R_1^{-1}(x)\Big(C_2(x) + \frac{1}{\gamma^2}D_{21}(x)B_1^T(x)M^T P\Big) \\ \hat{B}_1(x) &= B_1(x)[I - D_{21}^T(x)R_1^{-1}(x)D_{21}(x)] \\ \hat{C}(x) &= \big(C_2(x) + \frac{1}{\gamma^2}D_{21}(x)B_1^T(x)P\big)x^{\{d\}} \end{aligned} \tag{35}$$

Proof. Suppose exists a negative definite function $\mathfrak{S}(x)$ for each nonzero x,

$$\begin{aligned}\mathfrak{S}(x) =& Q_x[A(x)x^{\{d\}} + B_1(x)\alpha_1(x) - G(x)\Big(C_2(x)x^{\{d\}} + D_{21}(x)w\Big)] + \alpha_2^T(x)R_2(x)\alpha_2(x)\\ &+ \frac{1}{4\hat{\gamma}^2}Q_x[B_1(x) - G(x)D_{21}(x)][B_1(x) - G(x)D_{21}(x)]^T Q_x^T \le 0\end{aligned} \tag{36}$$

such that its Hessian matrix $\frac{\partial^2 \mathfrak{S}(x)}{\partial x^2}$ is nonsingular, where $Q(x)$ is a C^3 positive-definite function $Q : N_1 \subset \mathscr{X} \to \mathbb{R}_+$ locally defined in a neighborhood N_1 of $x = 0$, and vanishing at $x = 0$. In order for Q to satisfy *HJIq* (25), i.e. $Q(x-\xi) = W(X)$, is to proof that a function $\mathfrak{K}(x,\xi)$ is non-positive, for

$$\begin{aligned}\mathfrak{K}(x,\xi) =& W_X F(X) + v^T(X)R_2(x)v(X) + \frac{1}{4\hat{\gamma}^2}W_X g(X)g^T(X)W_X^T\\ =& \left[W_x(X)\ W_\xi(X)\right] F(X) + h^{eT}(X)R_2(x)h^e(X)\\ &+ \frac{1}{4\hat{\gamma}^2}\left[W_x(X)\ W_\xi(X)\right]\begin{bmatrix} g_1(x)g_1^T(x) & 0\\ 0 & G(\xi)R_1(x)G^T(\xi)\end{bmatrix}\begin{bmatrix} W_x^T(X)\\ W_\xi^T(X)\end{bmatrix}.\end{aligned} \tag{37}$$

By setting $e = x - \xi$ and defining

$$\mathfrak{F}(e,\xi) = \mathfrak{K}(x,\xi)\Big|_{x=e+\xi} \tag{38}$$

then by a second order Taylor expansion we obtain,

$$\mathfrak{F}(e,\xi) \approx \mathfrak{F}(0,\xi) + e^T\frac{\partial \mathfrak{F}(e,\xi)}{\partial e}\Big|_{e=0} + e^T\frac{\partial^2 \mathfrak{F}(e,\xi)}{\partial e^2}\Big|_{e=0} e \tag{39}$$

It can be shown that,

$$\mathfrak{F}(0,\xi) = \frac{\partial \mathfrak{F}(e,\xi)}{\partial e}\Big|_{e=0} = 0 \tag{40}$$

and that

$$\frac{\partial^2 \mathfrak{F}(e,\xi)}{\partial e^2}\Big|_{e,\xi=0} = \frac{\partial^2 \mathfrak{S}(x)}{\partial x^2}\Big|_{x=0}. \tag{41}$$

Since we set $\mathfrak{S}(x)$ to be non-positive we obtain that

$$\frac{\partial^2 \mathfrak{S}(x)}{\partial x^2}\Big|_{x=0} < 0 \tag{42}$$

which results in $\mathfrak{F}(e,\xi)$ being non-positive in the neighbourhood of $(e,\xi) = (0,0)$. Thus the function $Q(x-\xi)$ satisfies *HJIq* (25). By completion of the squares it can be shown that the function $\mathfrak{S}(x)$ satisfies the following inequality,

$$\begin{aligned}\mathfrak{S}(x) \geq & Q_x\left[A(x)x^{\{d\}}+B_1(x)\alpha_1(x)-B_1(x)D_{21}^T(x)R_1^{-1}(x)\Big(C_2(x)x^{\{d\}}+D_{21}(x)w\Big)\right]\\ & +\frac{1}{4\hat{\gamma}^2}Q_xB_1(x)\left[I-D_{21}^T(x)R_1^{-1}(x)D_{21}(x)\right]B_1^T(x)Q_x^T+\alpha_2^T(x)R_2(x)\alpha_2(x)\\ & -\hat{\gamma}^2\Big(C_2(x)x^{\{d\}}+D_{21}(x)w\Big)^TR_1^{-1}(x)\Big(C_2(x)x^{\{d\}}+D_{21}(x)w\Big)\end{aligned} \tag{43}$$

The latter inequality becomes an equality when,

$$Q_xG(x)=\left[2\hat{\gamma}^2\Big(C_2(x)x^{\{d\}}+D_{21}(x)w\Big)^T+Q_xB_1(x)D_{21}^T(x)\right]R_1^{-1}(x) \tag{44}$$

As a result we can conclude that in order for $\mathfrak{S}(x)$ to be non-positive, it is suffices to assume that the right hand side of inequality (43), which does not contain $G(x)$, is negative for each nonzero x. The right hand side of (43) can be written as,

$$Q_xA_0(x)+\frac{1}{4\hat{\gamma}^2}Q_xB_0(x)B_0^T(x)Q_x^T+\hat{T}_0(x)\leq 0 \tag{45}$$

where $A_0(x)$, $B_0(x)$ and $\hat{T}_0(x)$ are similarly defined in (27) . Assuming that $Q=x^{\{d\}T}(x)T^{-1}x^{\{d\}}$, and by the use of the schur complement we obtain (32) □

Remark 2. It seems that the latter result is true for $G(x)$ and not for $G(\xi)$, although it can be easily shown that $G(x)$ and $G(\xi)$ are dual. This is done proving that

$$Q(e)=Q(x-\xi)=Q(\xi-x)=Q(-e) \tag{46}$$

i.e. (45), (33) can be written for ξ and not x. Thus, to show that $W(X)=Q(\xi-x)$ satisfies the Hamilton Jacobi inequality (24), is to show that the function $\mathfrak{K}(\xi,x)$ is non-positive. Therefore, similar to the proof which was presented before, by setting $e=-e$ and defining

$$\mathfrak{F}(-e,x)=\mathfrak{K}(\xi,x)\Big|_{\xi=-e+x} \tag{47}$$

results in $\mathfrak{F}(-e,x)$ being non-positive in the neighbourhood of $(-e,x)=(0,0)$. Thus the function $Q(\xi-x)$ satisfies the HJI (24).

If we conclude, in order to solve the H_∞ control via output-feedback with the use of SOS, the following convex optimization problems needs to be solved, for the state-feedback

$$\begin{aligned}\text{minimize} \quad & \gamma \quad \forall\,\zeta\\ \text{subject to} \quad & V(x) \quad \in \text{SOS}\\ & -\zeta^THJI\zeta \quad \in \text{SOS}\end{aligned} \tag{48}$$

and for the output measurement-feedback

$$\begin{array}{rll} \text{minimize} & \hat{\gamma} & \forall\, \zeta \\ \text{subject to} & Q(x) & \in \text{SOS} \\ & -\zeta^T HJI_q \zeta & \in \text{SOS} \end{array} \tag{49}$$

Thus, in order to implement the algorithm, ones needs to compute:

- The state-feedback problem (48), which result in $K(x)$, P and γ.
- The output-feedback problem (49) for $K(x)$, P and γ, which result in T and $\hat{\gamma}$.
- Compute a suitable matrix $L(x)$ which satisfies (34).
- Compute the estimators dynamic gain $G(x)$ from (33).
- Solve the estimator dynamics (31) for $w_1 = w_2 = \frac{1}{\gamma^2} B_1^T(\xi) P \xi^{\{d\}}$.

The following example will present the advantages of the use of SOS over the traditional solution; where by the use of SOS, the acceptable domain of suitable Lyapunov functions is much larger.

Example 1. Considering the following non linear system:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -0.01 - 0.1x_1^2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.8 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ 1 + 0.13x_1^2 \end{bmatrix} u \\ \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} 0.6 & 0.3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1.61 & 0 \\ 0 & 1.38 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \end{aligned} \tag{50}$$

Solving the output-feedback H_∞ problem for a *second* order Lyapunov function yields $\hat{\gamma} = 1.55$ and storage function,

$$Q(x) = 1.53x_1^2 + 1.3x_1x_2 + 1.62x_2^2$$

while the solution of output-feedback H_∞ problem for a *fourth* order Lyapunov function yields $\hat{\gamma} = 1.02$ and a storage function,

$$Q(x) = 0.38x_1^4 + 1.87x_1^2 + 1.93x_1x_2 + 1.13x_2^2$$

The above example reveals the advantages of the use of SOS, where a better disturbance attenuation closed-loop system is achieved.

5 Spacecraft Attitude Control

Consider a rigid body spacecraft which rotates around its center of mass under the influence of control and perturbations torques. Let $\mathscr{B}$ denote a spacecraft body frame, i.e., a Cartesian coordinates frame with the origin at the center of mass. Let $\mathscr{R}$ denote the Earth Centered Earth Inertial reference frame (ECEI). Let $\mathbf{q}$ denote the quaternion of rotation from $\mathscr{R}$ to $\mathscr{B}$, with vector part $\mathbf{e}$ and scalar part q [33, p. 758], and ω denote the angular velocity of $\mathscr{B}$ with respect to $\mathscr{R}$ expressed in $\mathscr{B}$. The rotational dynamics and kinematics of the rigid body spacecraft are governed by the following differential equations [33, Chap. 16]

$$\frac{d}{dt}\begin{bmatrix}\omega \\ \mathbf{e} \\ q\end{bmatrix} = \begin{bmatrix}-J^{-1}[\omega\times]J\omega \\ \frac{1}{2}\left(qI_3 + [\mathbf{e}\times]\right)\omega \\ -\frac{1}{2}\mathbf{e}^T\omega\end{bmatrix} + \begin{bmatrix}J^{-1} \\ \mathbf{0}_{4\times 3}\end{bmatrix}\mathbf{T}_b \tag{51}$$

where J denotes the spacecraft tensor of inertia matrix in $\mathscr{B}$, $[\omega\times]$ denotes the cross-product matrix related to ω, and $\mathbf{T}_b$ is the vector of total external torques applied to the spacecraft, i.e.

$$\mathbf{T}_b = \mathbf{u}_b + \mathbf{w}_b \tag{52}$$

where $\mathbf{u}_b$ denote the 3×1 vector of control torques and $\mathbf{w}_b$ denote the 3×1 vector of disturbance torques. It is assumed that the Attitude Control System is equipped with a triad of three orthogonal reaction wheels, providing full control authority in all axes. The perturbation torques, modeled via $\mathbf{w}_b$, typically include the gravity gradient torque, the aerodynamic torque, a residual magnetic torque, and the solar pressure torque. Equation (51) is re-written as follows

$$\dot{\mathbf{x}} = f(\mathbf{x}) + G\mathbf{u}_b + G\mathbf{w}_b \tag{53}$$

where $\mathbf{x} \triangleq \{\omega, \mathbf{q}\}$. Notice that $f(\mathbf{x})$ is a polynomial function of the state variables. Also notice that f, G are C^k with $k \geq 2$, and that the unforced system has two equilibrium points:

$$(\omega, \mathbf{e}, q) = (\mathbf{0}, \mathbf{0}, \pm 1) \tag{54}$$

where both $\mathbf{q}_{1,2} = (0,0,0,\pm 1)$ correspond to the null attitude. It is assumed that the Attitude Determination and Control System is equipped with a suite of sensors that guarantee full observability of the state, such that $\mathbf{q}$ and ω can be estimated. As a first step, before applying more realistic assumptions, it is further assumed that the estimation errors can be neglected, i.e., that there is full state information. The attitude control objectives consist in globally stabilizing the system state Eq. (53) with respect to the equilibrium point $(\mathbf{0}, \mathbf{0}, 1)$, while attenuating the influence of the exogenous inputs $\mathbf{w}_b$ on the system dynamics.

5.1 Spacecraft Attitude Control Simulation

Considering the rotational dynamics and kinematics of a rigid body spacecraft by the differential equations governed in (51). The disturbance torque, w_b, is simulated as the sum of a torque caused by a impact collision and the aerodynamic drag. The impact collision, which is caused by the impact of a 1 gr particle, is described as a impulse function of 1.5 Nm with a duration of 0.1 sec. The particle hits the spacecraft at a velocity of 10 km/s and at a distance of 1 cm from the center of mass. The aerodynamic drag disturbance torque will be modeled as a first order Markov process, which has the worst case magnitude for approximately 5% of the orbital period,

$$\begin{aligned} &T_{d+1} = \alpha_d \mathscr{N} + (1-\alpha_d)T_d \\ &\text{If } \ \|T_d\| > 1.89 \cdot 10^{-3}\text{Nm} \\ &\text{Then } \ T_d = T_d \frac{2.89 \cdot 10^{-3}}{\|T_d\|} \end{aligned} \tag{55}$$

where $\alpha_d = 1/6000$ is a filter constant, which determines the speed of the random walk, where a smaller value means a smaller speed. The variable $\mathscr{N}$ is a Gaussian white noise with a standard deviation of $0.75 \cdot 10^{-5}$ Nm. The initial value T_d is chosen as a random unit vector with a magnitude of $2.89 \cdot 10^{-5}$ Nm.

The measurement noise w_n is described as a finite energy Gaussian white noise. Let $\sigma_\omega(t)$ denote the time-varying variance intensity of the gyro's angler velocity measurements, which are equal to $0.25 \cdot 10^{-3}$, and $\sigma_{\mathbf{q}}(t)$ as a time-varying variance intensity of the line-of-sight quaternion's measurements noise, which is equal to $0.25 \cdot 10^{-4}$. The inertia matrix which was chosen is similar to a typical microsatellite system and is equal to

$$J = \begin{bmatrix} 0.06 & 1 \cdot 10^{-3} & 6 \cdot 10^{-4} \\ 1 \cdot 10^{-3} & 0.05 & 5 \cdot 10^{-4} \\ 6 \cdot 10^{-4} & 5 \cdot 10^{-4} & 0.015 \end{bmatrix} \text{kgm}^2. \tag{56}$$

In order to use a quadratic Lyapunov function such that $x^{\{d\}} = x$ and such that the equilibrium vector is $\begin{bmatrix}\mathbf{0}_{1\times3} \ \mathbf{0}_{1\times3} \ 0\end{bmatrix}^T$ and not $\begin{bmatrix}\mathbf{0}_{1\times3} \ \mathbf{0}_{1\times3} \ 1\end{bmatrix}^T$ we shall perform a change in variables, i.e $\tilde{q} \triangleq q - 1$ which result in the following tracking error dynamic system,

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix}\omega \\ \tilde{\mathbf{e}} \\ \tilde{q}\end{bmatrix} &= \begin{bmatrix} -J^{-1}[\omega\times]J & \mathbf{0}_{3\times4} \\ 0.5 \cdot \begin{bmatrix}\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times1}\end{bmatrix} & 0.5 \cdot \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix}\omega \\ \tilde{\mathbf{e}} \\ \tilde{q}\end{bmatrix} + \begin{bmatrix}\mathbf{1}_{3\times1} \\ \mathbf{0}_{4\times1}\end{bmatrix} w_b + \begin{bmatrix}J^{-1} \\ \mathbf{0}_{4\times3}\end{bmatrix}\mathbf{u}_b \\ z = \begin{bmatrix}\mathbf{u}_b \\ \tilde{\mathbf{e}} \\ \tilde{q}\end{bmatrix} &= C_1 \begin{bmatrix}\omega \\ \tilde{\mathbf{e}} \\ \tilde{q}\end{bmatrix} + D_{12}\mathbf{u}_b \\ y = \mathbf{I}_{7\times7} \begin{bmatrix}\omega \\ \tilde{\mathbf{e}} \\ \tilde{q}\end{bmatrix} &+ \begin{bmatrix} \frac{\sigma_{\mathbf{q}}}{\sigma_\omega}{}_{3\times1} \\ \sigma_{\mathbf{q}3\times1} \end{bmatrix} w_n \end{aligned} \tag{57}$$

where $\mathbf{1}$ denotes a vector of ones. The measurements of $\begin{bmatrix}\tilde{\mathbf{e}} \ \tilde{q}\end{bmatrix}^T$ are obtained from the line-of-sight quaternion's measurements $\begin{bmatrix}\mathbf{e} \ q\end{bmatrix}^T$. Several simulation were

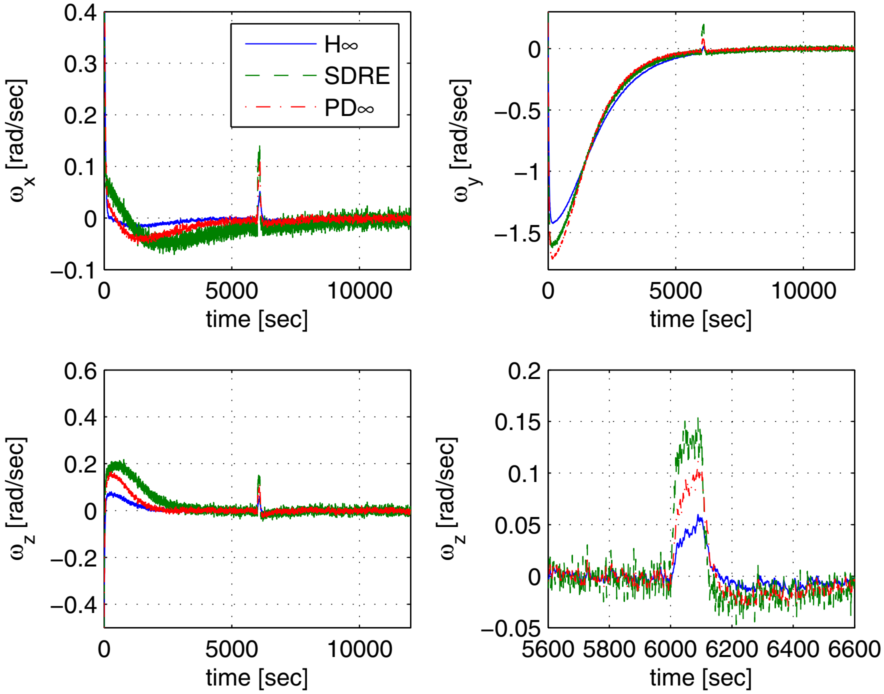


Fig. 1 Closed-loop angular velocities based on three different controllers. All three controllers use the nonlinear H_∞ estimator. It can be clearly seen that the H_∞ controller achieves a better disturbance attenuation closed-loop system.

performed, in order to measure the performance of the H_∞ output-feedback controller which was derived. The attenuation level which was obtained from the semi-definite optimization problem was $\gamma = 0.08$ and $\hat{\gamma} = 4.9$, where the matrix L was chosen by minimizing its Euclidean norm, while satisfying (34). In addition the Lyapunov functions which where obtained are,

$$
\begin{aligned}
V(\mathbf{x}) =& 114.2q^2 + 114.26e_1^2 + 114.23e_2^2 + 0.014qe_3 + 114.2e_3^2 + 4.07q\omega_1 \\
&+ 0.1e_3\omega_1 + 0.07\omega_1^2 + 0.11q\omega_2 + 0.0801e_3\omega_2 + 0.03\omega_1\omega_2 + 0.13\omega_2^2 \\
&+ e_2\left(0.14q + 0.0409e_3 + 0.11\omega_1 + 0.067\omega_2 + 0.06\omega_3\right) \\
&+ e_1\left(0.09q + 0.099e_2 + 0.11e_3 + 0.122\omega_1 + 0.15\omega_2 + 0.11\omega_3\right) \\
&+ 0.13q\omega_3 + 0.053e_3\omega_3 + 0.09\omega_1\omega_3 + 0.08\omega_2\omega_3 + 0.08\omega_3^2
\end{aligned} \tag{58}
$$

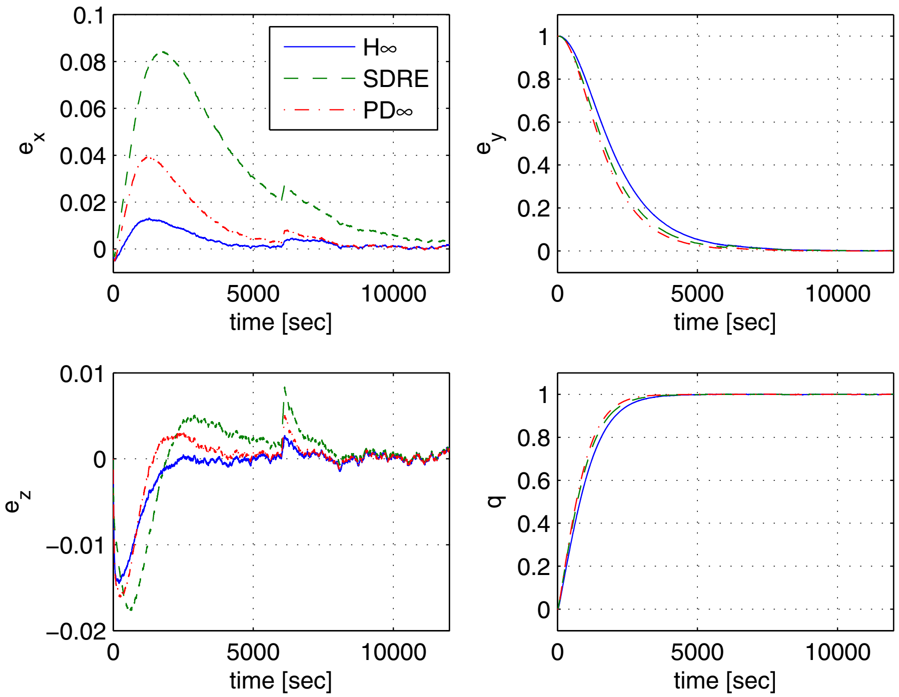


Fig. 2 Closed-loop attitude quaternion time histories for three different controllers. All three controllers use the nonlinear H_∞ estimator.

$$\begin{aligned} Q(\mathbf{x}) =& 15.61q^2 + 14.57e_1^2 + 15.55e_2^2 + 0.05qe_3 + 14.56e_3^2 + 0.12q\omega_1 \\ &+ 0.15e_3\omega_1 + 0.01\omega_1^2 + 0.064q\omega_2 + 0.07e_3\omega_2 + 0.08\omega_1\omega_2 + 0.02\omega_2^2 \\ &+ e_2\left(0.11q + 0.14e_3 + 0.81\omega_1 + 0.05\omega_2 + 0.042\omega_3\right) \\ &+ e_1\left(0.19q + 0.14e_2 + 0.14e_3 + 0.09\omega_1 + 0.13\omega_2 + 0.12\omega_3\right) + 0.11q\omega_3 \\ &+ 0.92e_3\omega_3 + 0.11\omega_1\omega_3 + 0.06\omega_2\omega_3 + 0.07\omega_3^2 \end{aligned} \tag{59}$$

It is of great interest to compare the H_∞ performance with a standard proportional controller [34, 14], and with an optimal nonlinear control law, for example the state dependent Riccati equality (SDRE) controller [6]. The proportional controller, PD_∞, was derived based on the H_∞ controller. While, in both cases the SDRE and the proportional controller used the H_∞ estimator. The initial conditions for the simulations were considered as $\left[1\ 0.5\ -0.5\ 0\ 1\ 0\ 0\right]^T$. An extended Kalman filter (EKF) was implemented as well for the SDRE controller, but was not capable to cope with the disturbances and as a result did not converge. It can be seen from Fig. 1 and Fig. 2 that the H_∞ controller achieves a better disturbance attenuation closed-loop system

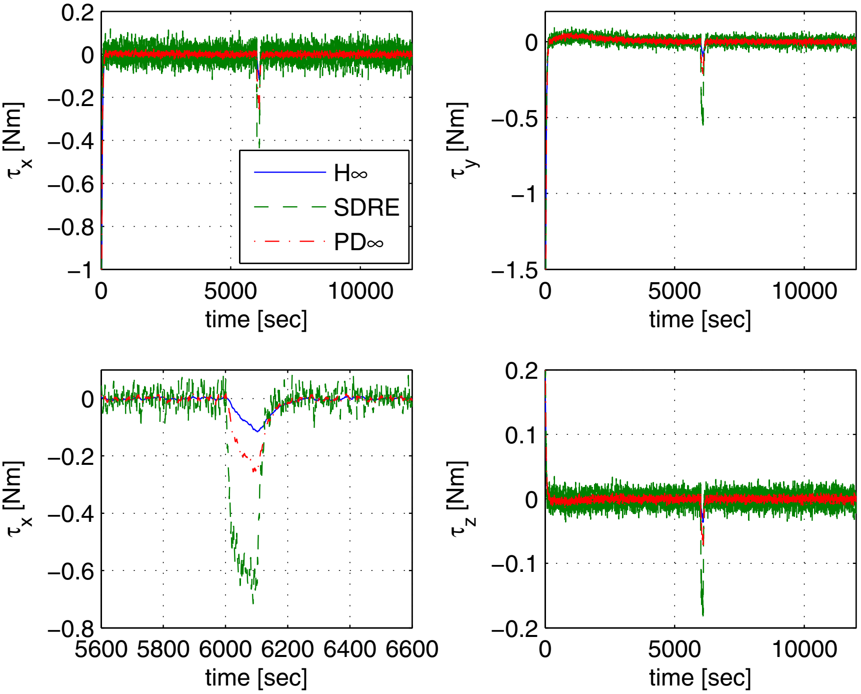


Fig. 3 Control signals time histories obtained from three different controllers

than the SDRE and the proportional controllers. Moreover, the measurement noise is better attenuated, and the control effort is reduced, despite the fact that they are both based on the H_∞ estimator.

6 Conclusions

A novel computational scheme was presented in order to solve the output-feedback H_∞ control problem for a class of nonlinear systems with polynomial vector field. By converting the resulting Hamilton-Jacobi inequalities from rational forms to their equivalent polynomial forms, we overcome the non-convex nature and numerical difficulty. Using quadratic Lyapunov functions, both the state-feedback and output-feedback problems were reformulated as semi-definite optimization conditions, while locally tractable solutions were obtained through sum of squares (SOS) programming. A numerical example and a spacecraft attitude control simulation showed that the proposed computational scheme result in a better disturbance attenuation closed-loop system, and more robust, while compared to standard methods.

References

1. Aliyu, M.: Nonlinear H_∞ Control, Hamiltonian Systems and Hamilton-Jacobi Equations. CRC Press (2011)
2. Basar, T., Bernhard, P.: H_∞ Optimal Control and Related Minimax Design Problems. Birkhäuser (1991)
3. Chesi, G.: LMI techniques for optimization over polynomials in control: A survey. IEEE Transactions on Automatic Control 55, 2500–2510 (2010)
4. Chesi, G., Tesi, A., Vicino, A., Genesio, R.: On convexification of some minimum distance problems. In: Proceeding 5th Europe Control Conference, Karlsruhe, Germany (1999)
5. Chu, Q.P., Acquatella, P., Van Kampen, E.J., Falkena, W.: Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Paper AIAA-4623 (2012)
6. Haim, L., Choukroun, D.: Optimized state-dependent riccati equation method for spacecraft attitude estimation and control. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Paper AIAA-4557 (2012)
7. Hol, C., Scherer, C.: Sum of squares relaxations for polynomial semi-definite programming. In: Proc. Symp. Mathematical Theory of Networks and Systems (2004)
8. Isidori, A., Astolfi, A.: Disturbance attenuation and H_∞-control via measurement feedback in nonlinear systems. IEEE Transactions on Automatic Control 37, 1283–1293 (1992)
9. Isidori, A., Kang, W.: H_∞ control via measurement feedback for general nonlinear systems. IEEE Transactions on Automatic Control 40, 466–472 (1995)
10. Kang, W., De, P., Isidori, A.: Flight control in a windshear via nonlinear H_∞ methods. In: Proceedings of the 31st IEEE Conference on Decision and Control, vol. 1, pp. 1135–1142 (1992)
11. Kim, J., Crassidis, J.: Spacecraft attitude control using approximate receding-horizon model-error control synthesis. Journal of Guidance, Control, and Dynamics 29 (2006)
12. Lu, W.M., Doyle, J.: H_∞ control of nonlinear systems via output feedback: a class of controllers. In: Proceedings of the 32nd IEEE Conference on Decision and Control (1993)
13. Lu, W.M., Doyle, J.C.: H_∞ control of nonlinear systems: a convex characterization. IEEE Transactions on Automatic Control 40, 1668–1675 (1995)
14. Mortensen, R.E.: A globally stable linear attitude regulator. International Journal of Control 8(3), 297–302 (1968), `http://www.tandfonline.com/doi/abs/10.1080/00207176808905679`, doi:10.1080/00207176808905679
15. Papachristodoulou, A., Prajna, S.: On the construction of Lyapunov functions using the sum of squares decomposition. In: Proceeding the 41st IEEE Conference on Decision and Control, pp. 3482–3487 (2002)
16. Parrillo, P.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Inst. Technol., Pasadena, CA (2000)
17. Prajina, S., Parrilo, P.A., Rantzer, A.: Nonlinear control sysnthesis by convex optimization. IEEE Transactions on Automatic Control 49, 310–314 (2004)
18. Prajna, S., Papachristodoulou, A., Seiler, P., Parrilo, P.: SOSTOOLS: Sum of squares optimization toolbox for matlab (2004), `http://www.cds.caltech.edu/sostools`
19. Prajna, S., Papachristodoulou, A., Seiler, P., Parrilo, P.A.: SOSTOOLS: Control applications and new developments. In: Proceeding of IEEE Conf. on CACSD (2005)

20. Prajna, S., Papachristodoulou, A., Wu, F.: Nonlinear control synthesis by sum of squares optimization a lyapunov-based approach. In: Proceeding the Asian Control Conference (2004)
21. Prempain, E.: An application of the sum of squares decomposition to the L_2-gain computation for a class of nonlinear systems. In: Proceeding of the 44th IEEE Conference on Decision and Control, the European Control Conference Sevile, Spain, pp. 6865–6868 (2005)
22. Paytner, S.J., Bishop, R.: Adaptive nonlinear attitude control and momentum mangeement of spacecraft. Journal of Guidance, Control, and Dynamics 20 (1997)
23. van der Schaft, A.: On a state space approach to nonlinear H_∞ control. Systems & Control Letters 16, 1–8 (1991)
24. van der Schaft, A.: L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control. IEEE Transactions on Automatic Control 37, 770–784 (1992)
25. van der Schaft, A.: L_2 - Gain and Passivity Techniques in Nonlinear Control, London (2000)
26. Schaub, H., Junkins, J., Robinett, R.: New penalty functions and optimal control formulation for spacecraft attitude control problems. Journal of Guidance, Control, and Dynamics 20 (1997)
27. Scherer, C.: Multiobjective H_2 / H_∞ control. IEEE Transactions on Automatic Control 40, 1054–1062 (1995)
28. Tan, W., Packard, A.: Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. IEEE Transactions on Automatic Control 53, 565–571 (2008)
29. Tsiotras, P., Corless, M., Rotea, M.A.: An L_2 disturbance attenuation solution to the nonlinear benchmark problem. International Journal of Robust and Nonlinear Control 8, 311–330 (1998)
30. Kang, W.: Nonlinear h_∞ control and its application to rigid spacecraft. IEEE Transactions on Automatic Control 40(7) (1995)
31. Tan, W.: Nonlinear control analysis and synthesis using sum of squares programming. Ph.D. thesis, University of California, Berkeley (2006)
32. Wei, X., Liu, L.: H_∞ control for a class of nonlinear system based on l_2 gain computation. In: Proceeding of the European Control Conference (2007)
33. Wertz, J.R.: Spacecraft Attitude Determination and Control. Kluwer Academic, Dordrecht (1978)
34. Wie, B., Weiss, H., Arapostathis, A.: Quarternion feedback regulator for spacecraft eigenaxis rotations. Journal of Guidance, Control, and Dynamics 12(3), 375–380 (1989)

Part II
Estimation and Navigation

Rotorcraft System Identification: An Integrated Time-Frequency Domain Approach

Marco Bergamasco and Marco Lovera

Abstract. The problem of rotorcraft system identification is considered and a novel, two step technique is proposed, which combines the advantages of time domain and frequency domain methods. In the first step, the identification of a black-box model using a subspace model identification method is carried out, using a technique which can deal with data generated under feedback; subsequently, in the second step, *a-priori* information on the model structure is enforced in the identified model using an $\mathscr{H}_\infty$ model matching method. A simulation study is used to illustrate the proposed approach.

1 Introduction

The problem of system identification of helicopter aeromechanics has been studied extensively in the last few decades, as identification has been known for a long time as a viable approach to the derivation of control-oriented dynamic models in the rotorcraft field (see for example the recent books [21, 12] and the references therein). Model accuracy is becoming more and more important, as progressively stringent requirements are being imposed on rotorcraft control systems: as the required control bandwidth increases, accurate models become a vital part of the design problem.

In the system identification literature, on the other hand, one of the main novelties of the last two decades has been the development of the so-called Subspace Model Identification (SMI) methods (see for example the books [22, 25]), which have proven extremely successful in dealing with the estimation of state space models for Multiple-Inputs Multiple-Outputs (MIMO) systems. Surprisingly enough, even though SMI can be effectively exploited in dealing with MIMO modelling problems, until recently these methods have received limited attention from the rotorcraft community, with the partial exception of some contributions such as [24, 7, 16]). SMI methods are particularly well suited for rotorcraft problems, for a number of

Marco Bergamasco · Marco Lovera
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
e-mail: {bergamasco,lovera}@elet.polimi.it

reasons. First of all, the subspace approach can deal in a very natural way with MIMO problems; in addition, all the operations performed by subspace algorithms can be implemented with numerically stable and efficient tools from numerical linear algebra. Finally, information from separate data sets (such as generated during different experiments on the system, *i.e.*, different test flights) can be merged in a very simple way into a single state space model. Recently, see [15], the interest in SMI for helicopter model identification has been somewhat revived and the performance of subspace methods has been demonstrated on flight test data. However, so far only methods and tools which go back 10 to 15 years in the SMI literature (such as the MOESP algorithm of [23] and the bootstrap-based method for uncertainty analysis of [8]) have been considered. Therefore, the further potential benefits offered by the latest developments in the field have not been fully exploited. Among other things, present-day approaches can provide:

- unbiased model estimates from data generated during closed-loop operation, as is frequently the case in experiments for rotorcraft identification (see, *e.g.*, [9, 11]);
- the possibility to quantify model uncertainty using analytical expressions for the variance of the estimates instead of relying on computational statistics (see [9]);
- the direct estimation of continuous-time models from (possibly non-uniformly) sampled input-output data (see [6] and the references therein).

Some preliminary results in the application of continuous-time SMI to the rotorcraft problem have been presented in [5].

The only, well known, downside of the SMI approach to state space model identification, on the other hand, is the impossibility to impose a fixed basis to the state space representation. This, in turn, implies that it is hard to impose a parameterisation to the state space matrices in this framework, and therefore recovering physically-motivated models is a challenging problem. This, to date, prevents the successful application of SMI methods to the problem of initialising iterative methods for the identification of structured state space representations and constitutes a major stumbling block for the application of such methods in communities in which physically motivated models represent the current practice.

In this paper the problem of bridging the gap between ”unstructured” models obtained using SMI and structured ones deriving from flight mechanics is addressed as an input-output model matching one, in terms of the $\mathscr{H}_\infty$ norm of the difference between the two models (see also [3]). The solution of the problem is then computed using recent results in non-smooth optimisation techniques, see [1], which yield effective computational tools (see [10]).

In view of the above discussion, this paper has the following objectives. First, a set of methods suitable for time-domain, continuous-time identification of rotorcraft dynamics using SMI is presented. The proposed technique can deal with data generated in closed-loop operation as it does not require restrictive assumptions in this sense. Subsequently, a frequency-domain $\mathscr{H}_\infty$ approach to the problem of deriving a structured model from the unstructured one is proposed. Finally, the achievable model accuracy is illustrated by means of simulation results for a full-scale helicopter.

The paper is organised as follows. In Section 2 the problem statement is given and some definitions are provided. Section 3 provides a summary of the proposed two-step approach. Finally, some simulation results are presented in Section 4 to illustrate the performance of the proposed method.

2 Problem Statement and Preliminaries

Consider the linear, time-invariant continuous-time system

$$\mathscr{M}_s(\lambda) : \begin{cases} \dot{x}(t) &= A(\lambda)x(t) + B(\lambda)u(t) + w(t),\ x(0) = x_0 \\ y(t) &= C(\lambda)x(t) + D(\lambda)u(t) + v(t) \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are, respectively, the state, input and output vectors and $w \in \mathbb{R}^n$ and $v \in \mathbb{R}^p$ are the process and the measurement noise, respectively, with covariance given by

$$E\left\{ \begin{bmatrix} w(t_1) \\ v(t_1) \end{bmatrix} \begin{bmatrix} w(t_2) \\ v(t_2) \end{bmatrix}^T \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta(t_2 - t_1).$$

The system matrices $A(\lambda)$, $B(\lambda)$, $C(\lambda)$, and $D(\lambda)$ are dependent on the constant parameter vector $\lambda \in \mathbb{R}^{n_\lambda}$ such that $(A(\lambda), C(\lambda))$ is observable and $(A(\lambda), [B(\lambda), Q^{1/2}])$ is controllable.

Assume now that a dataset $\{u(t_i), y(t_i)\}$, $i \in [1, N]$ of sampled input/output data (possibly associated with a non equidistant sequence of sampling instants) obtained from system (1) is available. Then, the problem is to provide an estimate of the parameter λ on the basis of the available data. Note that unlike most identification techniques, in this setting incorrelation between u and w, v is not required, so that this approach is viable also for systems operating under feedback.

In the following Sections a number of definitions will be used, which are summarised hereafter for the sake of clarity (see, *e.g.*, [26, 13, 17, 2] for further details).

Definition 1. (**Laguerre basis**) Let $\mathscr{L}_2(0,\infty)$ denote the space of square integrable and Lebesgue measurable functions of time $0 < t < \infty$. Consider the first order all-pass (inner) transfer function

$$w(s) = \frac{s-a}{s+a}, \tag{2}$$

$a > 0$. $w(s)$ generates the family of Laguerre filters, defined as

$$\mathscr{L}_i(s) = w^i(s)\mathscr{L}_0(s) = \sqrt{2a}\frac{(s-a)^i}{(s+a)^{i+1}}. \tag{3}$$

Denote with $\ell_i(t)$ the impulse response of the i-th Laguerre filter. Then, it can be shown that the set

$$\{\ell_0, \ell_1, \ldots, \ell_i, \ldots\} \tag{4}$$

is an orthonormal basis of $\mathscr{L}_2(0,\infty)$, *i.e.*, all signals in $\mathscr{L}_2(0,\infty)$ can be represented by means of the set of their projections on the Laguerre basis.

Definition 2. (**$\mathscr{H}_\infty$ norm**) Consider an asymptotically stable, linear time-invariant system with transfer function $G(s)$. Then the $\mathscr{H}_\infty$ norm of the system is defined as

$$\|G\|_\infty = \sup_{\alpha>0} \left\{ \sup_{\omega} \bar{\sigma}\left(G(\alpha + j\omega)\right) \right\} = \sup_{\omega} \bar{\sigma}\left(G(j\omega)\right), \tag{5}$$

where $\bar{\sigma}$ is the maximum singular value.

Identifiability is an important issue in system identification problems; for the purpose of this study we adopt the following definitions:

Definition 3. (**Local identifiability**) Let $\lambda^o \in \Lambda \subset \mathbb{R}^{n_\lambda}$, the model structure is said to be locally identifiable in λ^o if $\forall \lambda_1, \lambda_2$ in the neighborhood of λ^o it holds that

$$\mathscr{M}_s(\lambda_1) = \mathscr{M}_s(\lambda_2) \Rightarrow \lambda_1 = \lambda_2.$$

Definition 4. (**Global identifiability**) The model structure $\mathscr{M}_s(\lambda)$ is said to be globally identifiable if it is locally identifiable $\forall \lambda \in \Lambda$, *i.e.*, over the entire parameter space.

In the following the model structure $\mathscr{M}_s(\lambda)$ is considered globally identifiable.

3 An Integrated Time-Frequency Domain Approach

The problem formulated in the previous Section can be faced using a two-steps approach: in the first step a black-box model is identified using a continuous-time SMI method, which can deal with data generated under feedback but generates an ”unstructured” model; in the subsequent step *a-priori* information on the model structure is enforced in the model using an $\mathscr{H}_\infty$ model matching method.

In Section 2 the gray-box model $\mathscr{M}_s(\lambda)$ was introduced, while a generic ”unstructured” black-box model $\mathscr{M}_{ns}$ can be described as the linear time-invariant system

$$\mathscr{M}_{ns} : \begin{cases} \dot{x}(t) = \hat{A}x(t) + \hat{B}u(t) + w(t), \; x(0) = x_0 \\ y(t) = \hat{C}x(t) + \hat{D}u(t) + v(t) \end{cases} \tag{6}$$

where x, u, y, w, and v are defined as in Section 2. The system matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ have been estimated from a dataset $\{u(t_i), y(t_i)\}$, $i \in [1,N]$ of sampled input/output data using the continuous-time predictor-based subspace model identification algorithm introduced in the Section 3.1. Suppose $\mathscr{M}_{ns}$ belonging to the same model structure of $\mathscr{M}_s(\lambda)$, and that (1) and (6) describe the same system with different state space basis. Therefore the problem becomes to provide estimates of λ such that the input-output behaviors of $\mathscr{M}_{ns}$ and $\mathscr{M}_s(\lambda)$ are equivalent under some criterion, and it is faced using an $\mathscr{H}_\infty$ approach described in Section 3.2.

3.1 Continuous-Time Predictor-Based Subspace Model Identification

3.1.1 From Continuous-Time to Discrete-Time Using Laguerre Projections

The main issue in the application of subspace model identification methods to continuous-time model identification is the need of computing the high order derivatives of input-output measurements arising from the continuous-time data equation. This problem can be faced using a method, based on the results first presented in [19, 17], and further expanded in [14, 18], that transforms a continuous-time system and signals to their discrete-time representations. First note that under the assumptions stated in the previous section, (6) can be written in innovation form as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \tag{7}$$

and it is possible to apply the results of [19] to derive a discrete-time equivalent model, as follows. Note that the notation $\hat{(\cdot)}$ has been dropped for clarity. Consider the first order inner function $w(s)$ defined in (2) and apply to the input u, the output y and the innovation e of (7) the transformations

$$\begin{aligned} \tilde{u}(k) &= \int_0^\infty \ell_k(t)u(t)dt \\ \tilde{y}(k) &= \int_0^\infty \ell_k(t)y(t)dt \\ \tilde{e}(k) &= \int_0^\infty \ell_k(t)e(t)dt, \end{aligned} \tag{8}$$

where $\tilde{u}(k) \in \mathbb{R}^m$, $\tilde{e}(k) \in \mathbb{R}^p$ and $\tilde{y}(k) \in \mathbb{R}^p$. Then (see [19] for details) the transformed system has the state space representation

$$\begin{aligned} \xi(k+1) &= A_o\xi(k) + B_o\tilde{u}(k) + K_o\tilde{e}(k),\ \xi(0) = 0 \\ \tilde{y}(k) &= C_o\xi(k) + D_o\tilde{u}(k) + \tilde{e}(k) \end{aligned} \tag{9}$$

where the state space matrices are given by

$$\begin{aligned} A_o &= (A - aI)^{-1}(A + aI) \\ B_o &= \sqrt{2a}(A - aI)^{-1}B \\ K_o &= \sqrt{2a}(I - C(A - aI)^{-1}K)^{-1}(A - aI)^{-1}K \\ C_o &= -\sqrt{2a}C(A - aI)^{-1} \\ D_o &= D - C(A - aI)^{-1}B. \end{aligned} \tag{10}$$

It is worth to underline that in this context k is not a time index, but refers to the projection of the signals onto the k-th basis function.

3.1.2 Predictor-Based Subspace Model Identification

In this Section a summary of the continuous-time PBSID algorithm proposed in [4, 6], called CT-PBSID$_o$, is provided, and its implemention is discussed. More precisely, starting from system (7), a sketch of the derivation of a PBSID-like approach to the estimation of the state space matrices A_o, B_o, C_o, D_o, K_o is presented. Considering the sequence of sampling instants t_i, $i = 1,\dots,N$, the input u, the output y and the innovation e of (7) are subjected to the transformations

$$\begin{aligned}\tilde{u}_i(k) &= \int_0^\infty \ell_k(\tau)u(t_i+\tau)d\tau \\ \tilde{y}_i(k) &= \int_0^\infty \ell_k(\tau)y(t_i+\tau)d\tau \\ \tilde{e}_i(k) &= \int_0^\infty \ell_k(\tau)e(t_i+\tau)d\tau\end{aligned} \tag{11}$$

(or to the equivalent ones derived from (8)), where $\tilde{u}_i(k) \in \mathbb{R}^m$, $\tilde{e}_i(k) \in \mathbb{R}^p$ and $\tilde{y}_i(k) \in \mathbb{R}^p$. Then (see [19] for details) the transformed system has the state space representation

$$\begin{aligned}\xi_i(k+1) &= A_o\xi_i(k) + B_o\tilde{u}_i(k) + K_o\tilde{e}_i(k),\ \xi_i(0) = x(t_i) \\ \tilde{y}_i(k) &= C_o\xi_i(k) + D_o\tilde{u}_i(k) + \tilde{e}_i(k)\end{aligned} \tag{12}$$

where the state space matrices are given by (10).

Letting now

$$\tilde{z}_i(k) = \left[\tilde{u}_i^T(k)\ \tilde{y}_i^T(k)\right]^T$$

and

$$\begin{aligned}\bar{A}_o &= A_o - K_oC_o \\ \bar{B}_o &= B_o - K_oD_o \\ \widetilde{B}_o &= \left[\bar{B}_o\ K_o\right],\end{aligned}$$

system (12) can be written in predictor form as

$$\begin{aligned}\xi_i(k+1) &= \bar{A}_o\xi_i(k) + \widetilde{B}_o\tilde{z}_i(k),\ \xi_i(0) = x(t_i) \\ \tilde{y}_i(k) &= C_o\xi_i(k) + D_o\tilde{u}_i(k) + \tilde{e}_i(k),\end{aligned} \tag{13}$$

to which the PBSID$_{opt}$ algorithm, summarised hereafter, can be applied to compute estimates of the state space matrices A_o, B_o, C_o, D_o, K_o. To this purpose note that iterating $p-1$ times the projection operation (i.e., propagating $p-1$ forward in the index k the first of equations (13), where p is the so-called past window length) one gets

$$\begin{aligned}
\xi_i(k+2) &= \bar{A}_o^2 \xi_i(k) + \left[\bar{A}_o \widetilde{B}_o \ \widetilde{B}_o\right] \begin{bmatrix} \tilde{z}_i(k) \\ \tilde{z}_i(k+1) \end{bmatrix} \\
&\vdots \\
\xi_i(k+p) &= \bar{A}_o^p \xi_i(k) + \mathscr{K}^p Z_i^{0,p-1}
\end{aligned} \tag{14}$$

where

$$\mathscr{K}^p = \left[\bar{A}_o^{p-1} \widetilde{B}_0 \dots \widetilde{B}_o\right] \tag{15}$$

is the extended controllability matrix of the system in the transformed domain and

$$Z_i^{0,p-1} = \begin{bmatrix} \tilde{z}_i(k) \\ \vdots \\ \tilde{z}_i(k+p-1) \end{bmatrix}.$$

Under the considered assumptions, $\bar{A}_o$ has all the eigenvalues inside the open unit circle, so the term $\bar{A}_o^p \xi_i(k)$ is negligible for sufficiently large values of p and we have that

$$\xi_i(k+p) \simeq \mathscr{K}^p Z_i^{0,p-1}.$$

As a consequence, the input-output behaviour of the system is approximately given by

$$\begin{aligned}
\tilde{y}_i(k+p) &\simeq C_o \mathscr{K}^p Z_i^{0,p-1} + D_o \tilde{u}_i(k+p) + \tilde{e}_i(k+p) \\
&\vdots \\
\tilde{y}_i(k+p+f) &\simeq C_o \mathscr{K}^p Z_i^{f,p+f-1} + D_o \tilde{u}_i(k+p+f) + \\
&\quad + \tilde{e}_i(k+p+f),
\end{aligned} \tag{16}$$

so that introducing the vector notation

$$\begin{aligned}
Y_i^{p,f} &= \left[\tilde{y}_i(k+p) \ \tilde{y}_i(k+p+1) \dots \tilde{y}_i(k+p+f)\right] \\
U_i^{p,f} &= \left[\tilde{u}_i(k+p) \ \tilde{u}_i(k+p+1) \dots \tilde{u}_i(k+p+f)\right] \\
E_i^{p,f} &= \left[\tilde{e}_i(k+p) \ \tilde{e}_i(k+p+1) \dots \tilde{e}_i(k+p+f)\right] \\
\Xi_i^{p,f} &= \left[\xi_i(k+p) \ \xi_i(k+p+1) \dots \xi_i(k+p+f)\right] \\
\bar{Z}_i^{p,f} &= \left[Z_i^{0,p-1} \ Z_i^{1,p} \dots Z_i^{f,p+f-1}\right]
\end{aligned} \tag{17}$$

equations (14) and (16) can be rewritten as

$$\begin{aligned}
\Xi_i^{p,f} &\simeq \mathscr{K}^p \bar{Z}_i^{p,f} \\
Y_i^{p,f} &\simeq C_o \mathscr{K}^p \bar{Z}_i^{p,f} + D_o U_i^{p,f} + E_i^{p,f}.
\end{aligned} \tag{18}$$

Considering now the entire dataset for $i = 1, \ldots, N$, the data matrices become

$$\begin{aligned} Y^{p,f} = [&\tilde{y}_1(k+p) \ldots \tilde{y}_N(k+p) \ldots \\ &\tilde{y}_1(k+p+f) \ldots \tilde{y}_N(k+p+f)], \end{aligned} \tag{19}$$

and similarly for $U_i^{p,f}$, $E_i^{p,f}$, $\Xi_i^{p,f}$ and $\bar{Z}_i^{p,f}$. The data equations (18), in turn, are given by

$$\begin{aligned} \Xi^{p,f} &\simeq \mathscr{K}^p \bar{Z}^{p,f} \\ Y^{p,f} &\simeq C_o \mathscr{K}^p \bar{Z}^{p,f} + D_o U^{p,f} + E^{p,f}. \end{aligned} \tag{20}$$

From this point on, the algorithm can be developed along the lines of the discrete-time PBSID$_{opt}$ method, i.e., by carrying out the following steps. Considering $p = f$, estimates for the matrices $C_o \mathscr{K}^p$ and D_o are first computed by solving the least-squares problem

$$\min_{C_o \mathscr{K}^p, D_o} \| Y^{p,p} - C_o \mathscr{K}^p \bar{Z}^{p,p} - D_o U^{p,p} \|_F, \tag{21}$$

where by $\| \cdot \|_F$ we denote the Frobenius norm of a matrix. Defining now the extended observability matrix Γ^p as

$$\Gamma^p = \begin{bmatrix} C_o \\ C_o \bar{A}_o \\ \vdots \\ C_o \bar{A}_o^{p-1} \end{bmatrix} \tag{22}$$

and noting that the product of Γ^p and $\mathscr{K}^p$ can be written as

$$\Gamma^p \mathscr{K}^p \simeq \begin{bmatrix} C_o \bar{A}^{p-1} \widetilde{B}_o & \ldots & C_o \widetilde{B}_o \\ 0 & \ldots & C_o \bar{A} \widetilde{B}_o \\ \vdots & & \\ 0 & \ldots & C_o \bar{A}^{p-1} \widetilde{B}_o \end{bmatrix}, \tag{23}$$

such product can be computed using the estimate $\widehat{C_o \mathscr{K}^p}$ of $C_o \mathscr{K}^p$ obtained by solving the least squares problem (21).

Recalling now that

$$\Xi^{p,p} \simeq \mathscr{K}^p \bar{Z}^{p,p} \tag{24}$$

it also holds that

$$\Gamma^p \Xi^{p,p} \simeq \Gamma^p \mathscr{K}^p \bar{Z}^{p,p}. \tag{25}$$

Therefore, computing the singular value decomposition

$$\Gamma^p \mathscr{K}^p \bar{Z}^{p,p} = U \Sigma V^T \tag{26}$$

an estimate of the state sequence can be obtained as

$$\widehat{\Xi}^{p,p} = \Sigma_n^{1/2} V_n^T = \Sigma_n^{-1/2} U_n^T \Gamma^p \mathscr{K}^p \bar{Z}^{p,p}, \tag{27}$$

from which, in turn, an estimate of C_o can be computed by solving the least squares problem

$$\min_{C_o} \|Y^{p,p} - \widehat{D}_o U^{p,p} - C_o \widehat{\Xi}^{p,p}\|_F. \tag{28}$$

The final steps consist of the estimation of the innovation data matrix $E^{p,p}$

$$E^{p,p} = Y^{p,p} - \widehat{C}_o \widehat{\Xi}^{p,p} - \widehat{D}_o U^{p,p} \tag{29}$$

and of the entire set of the state space matrices for the system in the transformed domain, which can be obtained by solving the least squares problem

$$\min_{A_o, B_o, K_o} \|\widehat{\Xi}^{p+1,p} - A_o \widehat{\Xi}^{p,p-1} - B_o U^{p,p-1} - K_o E^{p,p-1}\|_F. \tag{30}$$

The state space matrices of the original continuous-time system can then be retrieved by inverting the (bilinear) transformations (10).

3.2 From Unstructured to Structured Models with an $\mathscr{H}_\infty$ Approach

Suppose that the linear continuous-time time-invariant system $\mathscr{M}_{ns}$ has been estimated from a dataset of sampled input/output data using the CT-PBSID$_o$ algorithm presented in the previous Section. Consider now the model class $\mathscr{M}_s(\lambda)$ introduced in Section 1. $\mathscr{M}_{ns}$ and $\mathscr{M}_s(\lambda)$ should have the same input-output behavior. This problem can be faced in a computationally effective way by defining the input-output operators associated with $\mathscr{M}_{ns}$ and $\mathscr{M}_s(\lambda)$ and seeking the values of the parameters corresponding to the solution of the optimisation problem

$$\lambda^\star = \arg\min_{\lambda} \|\mathscr{M}_{ns} - \mathscr{M}_s(\lambda)\| \tag{31}$$

for a suitably chosen norm. In the linear time-invariant case, the input-output operators can be represented as the transfer functions $\hat{G}_{ns}(s)$ and $G_s(s;\lambda)$ and the $\mathscr{H}_\infty$ norm is considered, so that the model matching problem can be recast as

$$\lambda^\star = \arg\min_{\lambda} \|\hat{G}_{ns}(s) - G_s(s;\lambda)\|_\infty. \tag{32}$$

Note that the open-loop dynamics of a helicopter is unstable in most flight conditions and so the $\mathscr{H}_\infty$ norm is undefined. In this case the eigenvalues of $\mathscr{M}_s(\lambda)$ and $\mathscr{M}_{ns}$ are shifted on the real axis by a suitable value μ as follows

$$\tilde{G}_s(s;\lambda) = C(\lambda)((s-\mu)I - A(\lambda))^{-1}B(\lambda) + D(\lambda) \tag{33}$$

$$\tilde{G}_{ns}(s) = \hat{C}((s-\mu)I - \hat{A})^{-1}\hat{B} + \hat{D}, \tag{34}$$

where μ is chosen such that all eigenvalues of $\mathscr{M}_{ns}$ have negative real part. Then the model matching problem is reformulated as

$$\lambda^\star = \arg\min_{\lambda} \|\tilde{G}_{ns}(s) - \tilde{G}_s(s;\lambda)\|_\infty. \tag{35}$$

As mentioned in the Introduction, this is a non-convex, non-smooth optimisation problem, which has been studied extensively in recent years in the framework of the fixed-structured controller design problem and for which reliable computational tools (see [10]) are presently available.

4 Simulation Study: Model Identification for the BO-105 Helicopter

The simulation example considered in this paper is based on the BO-105 helicopter. Possibly it is the most studied helicopter in the rotorcraft system identification literature. The BO-105 is a light, twin-engine, multi-purpose utility helicopter.

It is considered in forward flight at 80 knots, a flight condition which corresponds to unstable dynamics, with the aim of demonstrating the identification of a nine-DOF state-space model with test data extracted from a simulator based on the nine-DOF model from [20]. As described in the cited reference, the model includes the classical six-DOF and some additional states to account for some additional effects, namely:

- the BO-105 exhibits highly coupled body-roll and rotor-flapping responses; their interaction is represented in the model with a dynamic equation that describes the flapping dynamics using the cyclic controls.
- A second order dipole is appended to the model of roll rate response to lateral stick in order to account for the effect of lead-lag rotor dynamics.

Therefore, the simulator includes a nine-DOF linear model including the six-DOF quasi steady dynamics, the flapping equations and the lead-lag dynamics modelled with a complex dipole. Delays at the input of the model are also taken into account in the simulation, though they are not estimated. The state vector and the trim values are

$$x = \begin{bmatrix} u \; v \; w \; p \; q \; r \; \phi \;\; \theta \;\; a_{1s} \; b_{1s} \; x_1 \; x_2 \end{bmatrix}$$

and, respectively,

$$u_0 = 40\,\text{m/s}, \quad v_0 = 3\,\text{m/s}, \quad w_0 = -5\,\text{m/s}, \quad \phi_0 = 0\,\text{rad}, \quad \theta_0 = 0\,\text{rad}. \tag{36}$$

In details, the state vector includes the longitudinal flapping a_{1s}, the lateral flapping b_{1s} and two state variables x_1 and x_2, coming from the lead-lag dynamics complex dipole. The corresponding equations of motion are

$$
\begin{aligned}
\dot{u} &= X_u u + X_w w + X_p p + (X_q - w_0)q + v_0 r - g\theta + X_{\delta_{lon}}\delta_{lon} + X_{\delta_{col}}\delta_{col} \\
\dot{v} &= Y_v v + Y_w w + (Y_p + w_0)p + Y_q q + (Y_r - u_0)r - g\phi + Y_{\delta_{lat}}\delta_{lat} + Y_{\delta_{col}}\delta_{col} \\
\dot{w} &= Z_u u + Z_w w + (Z_p - v_0)p + u_0 q + Z_{\delta_{col}}\delta_{col} \\
\dot{p} &= L_u u + L_v v + L_w w + L_q q + L_{\delta_{b_{1s}}} b_{1s} + L_{\delta_{lon}}\delta_{lon} + L_{\delta_{ped}}\delta_{ped} + L_{\delta_{col}}\delta_{col} \\
\dot{q} &= M_v v + M_w w + M_p p + M_r r + M_{\delta_{a_{1s}}} a_{1s} + M_{\delta_{ped}}\delta_{ped} + M_{\delta_{col}}\delta_{col} \\
\dot{r} &= N_v v + N_w w + N_q q + N_r r + N_{\delta_{lon}}\delta_{lon} + N_{\delta_{lat}}\delta_{lat} + N_{\delta_{ped}}\delta_{ped} + N_{\delta_{col}}\delta_{col} \\
\dot{\phi} &= p \\
\dot{\theta} &= q \\
\dot{a}_{1s} &= -q - \frac{1}{\tau_f} a_{1s} + \frac{K_{a_{1s}}}{\tau_f}\delta_{lon}, \\
\dot{b}_{1s} &= -p - \frac{1}{\tau_f} b_{1s} + \frac{K_{b_{1s}}}{\tau_f}\delta_{lat} + K_{x_1} x_1 + K_{x_2} x_2 \\
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= C_1 x_1 + C_2 x_2 + \delta_{lat}.
\end{aligned}
\tag{37}
$$

Finally, the output vector is

$$
y = \begin{bmatrix} u \ v \ w \ p \ q \ r \ a_x \ a_y \ a_z \ \phi \ \theta \end{bmatrix},
$$

where

$$
\begin{aligned}
a_x &= \dot{u} + w_0 q - v_0 r + g\theta \\
a_y &= \dot{v} - w_0 p + u_0 r - g\phi \\
a_z &= \dot{w} + v_0 p - u_0 q,
\end{aligned}
$$

i.e., the state variables related to quasi-steady dynamics and the linear accelerations are measured. Considering (37), λ contains the stability derivatives, the control derivatives, the flapping and lead-lag rotor dynamics parameters, for a total of 47 parameters.

The identification experiment is performed in closed-loop because of the instability of the model, with the helicopter operating under feedback from an LQG controller tuned in order to maintain the helicopter close enough to trim to justify the identification of a linear model. In the experiment, additive perturbations have been applied to the input variables (δ_{lat},δ_{lon},δ_{ped},δ_{col}) computed by the controller; in particular, all the channels have been excited in the same experiment with pseudorandom binary signals with a duration of 60 s and a dwell time of 0.8 s. The perturbation of the control inputs has a 1% amplitude and the sampling time is 0.008s. For the purpose of the present preliminary study, measurement noise has not been included in the simulated data. The parameters of the algorithm presented in the previous Section have been chosen as $p = 40$ and $a = 45$. The obtained results are illustrated in Table 1.

Table 1 Comparison between simulator and black-box identified eigenvalues

	Simulator				Identified Model (CT-PBSID$_o$)			
	Real	Imag	Omega	Zeta	Real	Imag	Omega	Zeta
Pitch phugoid	0.119	0.278	0.302	-0.394	0.119	0.278	0.302	-0.394
Dutch roll	-0.571	2.546	2.609	0.219	-0.571	2.546	2.609	0.219
Roll/flapping	-9.904	7.740	12.569	0.788	-9.901	7.7399	12.568	0.788
Lead-Lag	-0.868	15.567	15.592	0.0557	-0.867	15.566	15.590	0.0556
Spiral	-0.0510				-0.0507			
Pitch$_1$	-0.448				-0.448			
Pitch$_2$	-5.843				-5.844			
Long. flapping	-15.930				-15.901			

As can be seen from the Table, the CT-PBSID$_o$ algorithm is able to identify the dynamics of the system with a slight loss of accuracy at high frequency.

The study in the reconstruction of the above described structured state-space representation has been carried out by applying the approach presented in Section 3 to estimate the relevant parameters. In order to evaluate the performance of the proposed method the relative estimation error is defined as follows

$$\lambda_{err} = \frac{\lambda^0 - \bar{\lambda}}{\lambda^0}, \tag{38}$$

where $\hat{\lambda}$ and λ^0 are respectively the estimated and the actual value of the parameter λ. In this example the relative errors of the estimated physical parameters in (37) are below 0.03%. It is clear from Table 2, where the eigenvalues of the real system and the identified gray-box model are shown, that using *a-priori* information, *i.e.*, exploting the model structure, the estimation accuracy increases.

Table 2 Comparison between simulator and gray-box identified eigenvalues

	Simulator				Identified Model (Gray-box)			
	Real	Imag	Omega	Zeta	Real	Imag	Omega	Zeta
Pitch phugoid	0.119	0.278	0.302	-0.394	0.119	0.278	0.302	-0.394
Dutch roll	-0.571	2.546	2.609	0.219	-0.571	2.546	2.609	0.219
Roll/flapping	-9.904	7.740	12.569	0.788	-9.903	7.740	12.568	0.788
Lead-Lag	-0.868	15.567	15.592	0.0557	-0.868	15.566	15.590	0.557
Spiral	-0.0510				-0.0507			
Pitch$_1$	-0.448				-0.448			
Pitch$_2$	-5.843				-5.843			
Long. flapping	-15.930				-15.929			

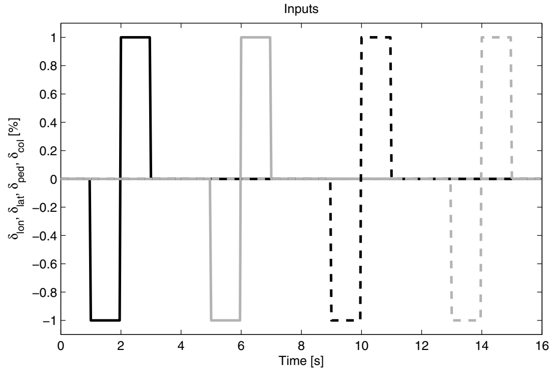


Fig. 1 Doublet input signal used for model validation

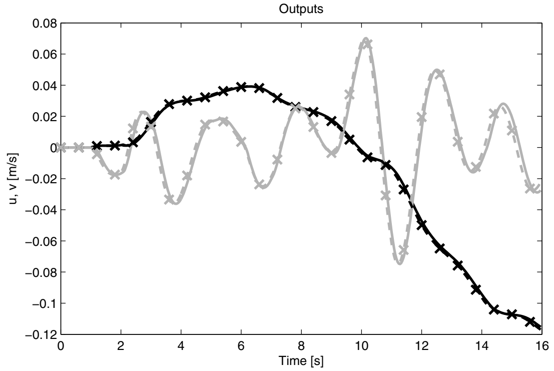


Fig. 2 Doublet output signals (real: solid line; black-box: dashed line; gray-box: cross)

Table 3 Relative errors norm

Output	$RMS_{CT-PBSID_o}$	$RMS_{Gray-Box}$
u	0.0013	0.0013
v	0.0044	0.0044
w	0.0026	0.0026
p	0.0002	0.0002
q	0.0003	0.0003
r	0.0003	0.0003
a_x	0.0013	0.0013
a_y	0.0017	0.0017
a_z	0.0077	0.0077
ϕ	0.0001	0.0001
θ	0.0001	0.0001

Finally, a time-domain validation of the identified models has been also carried out, by measuring the simulation accuracy of the models in response to a doublet input signal on each input channel. The input sequence used in the validation experiment is illustrated in Figure 1, while the time history for two of the outputs (*u* and *w*) is presented in Figure 2. Again, even though the open-loop system is unstable, the simulated outputs obtained from the identified models (dashed lines: black-box; cross: gray-box) match very well the ones computed from the nine-DOF model (solid lines).

In quantitative terms, considering the root mean square error, defined as

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(i)-\hat{y}(i))^2}, \tag{39}$$

where y is the real output and $\hat{y}$ is the estimated one, its value is below 0.01 on all the considered output variables as shown in Table 3. Note that most of the error is due the unestimated input delays, as can be seen in Figure 2.

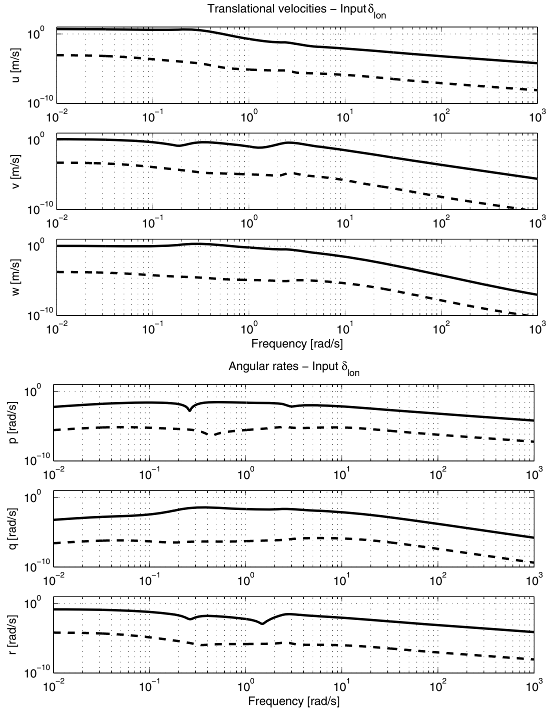


Fig. 3 Frequency response from longitudinal input to linear (top) and angular (bottom) velocities. (real: solid line; error: dashed line)

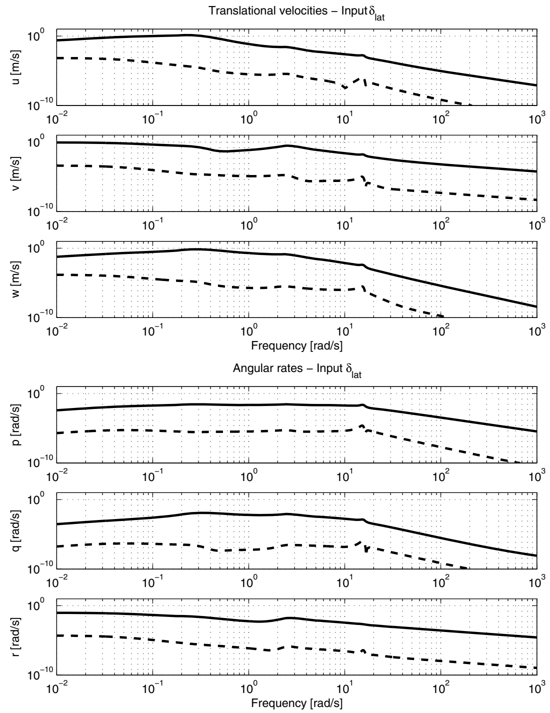


Fig. 4 Frequency response from lateral cyclic input to linear (top) and angular (bottom) velocities. (real: solid line; error: dashed line)

Finally, in Figures 3-14 the magnitude of the frequency response of the error transfer function defined as

$$E_s(s) = G_s(s;\lambda^0) - G_s(s;\hat{\lambda})$$

is shown, where $G(s;\lambda^0)$ is the true transfer function of the BO-105 model and $G_s(s;\hat{\lambda})$ is the gray-box estimated one. As can be seen from the figures, the magnitude of the error frequency response is always several orders of magnitude smaller than the one for the true transfer function.

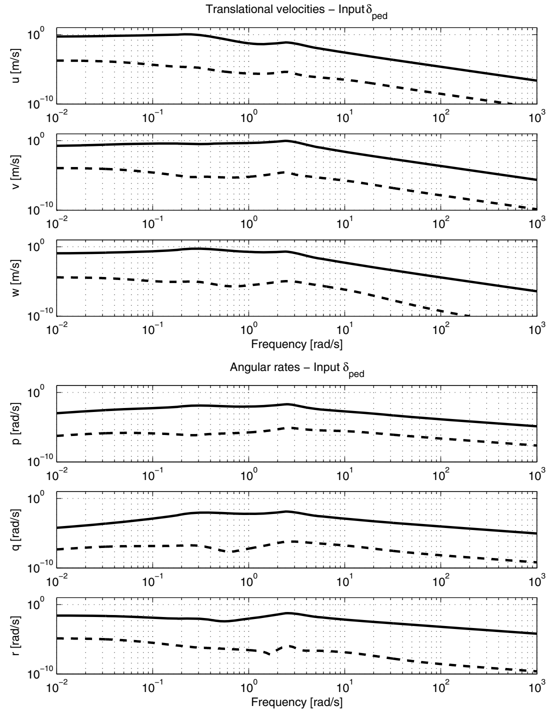


Fig. 5 Frequency response from pedal cyclic input to linear (top) and angular (bottom) velocities. (real: solid line; error: dashed line)

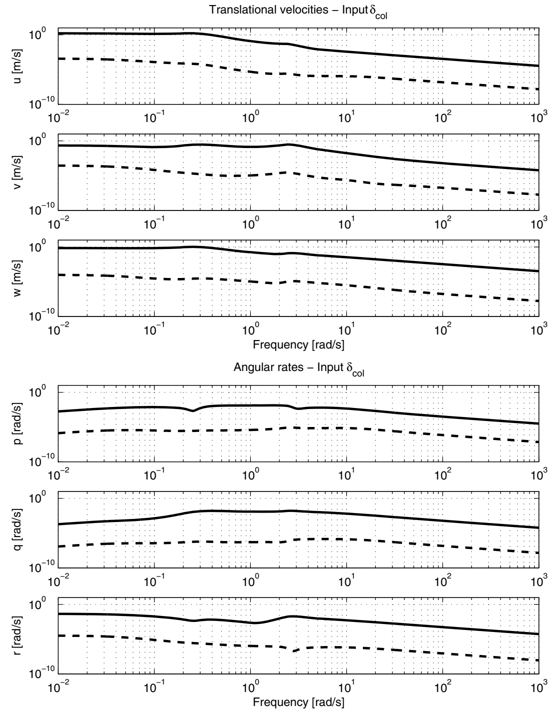


Fig. 6 Frequency response from collective input to linear (top) and angular (bottom) velocities. (real: solid line; error: dashed line)

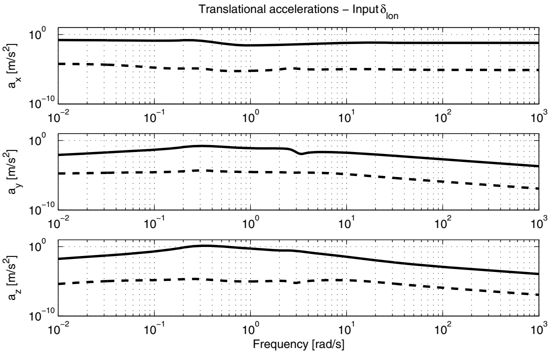


Fig. 7 Frequency response from longitudinal input to linear accelerations. (real: solid line; error: dashed line)

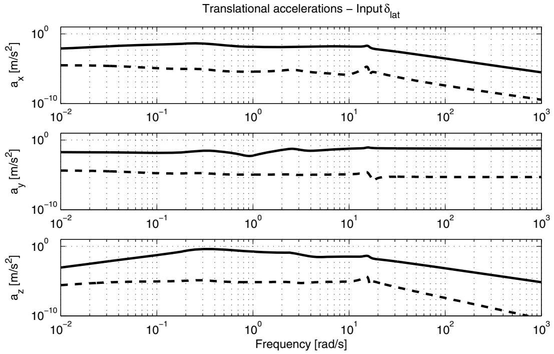


Fig. 8 Frequency response from lateral cyclic input to linear accelerations. (real: solid line; error: dashed line)

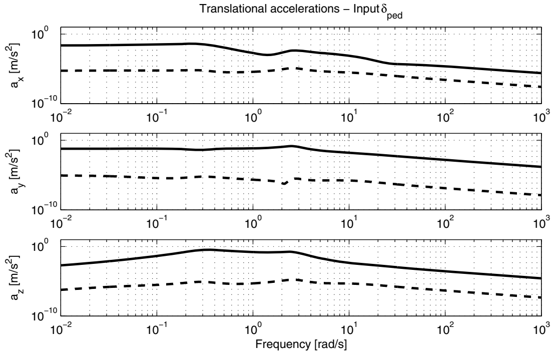


Fig. 9 Frequency response from pedal cyclic input to linear accelerations. (real: solid line; error: dashed line)

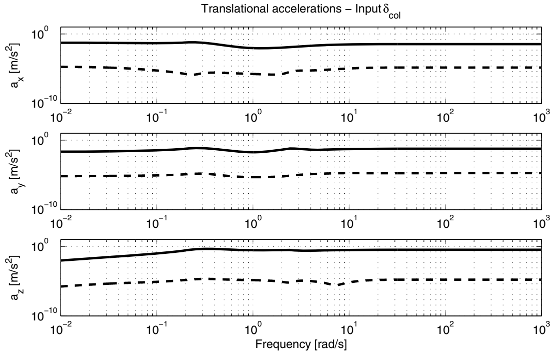


Fig. 10 Frequency response from collective input to linear accelerations. (real: solid line; error: dashed line)

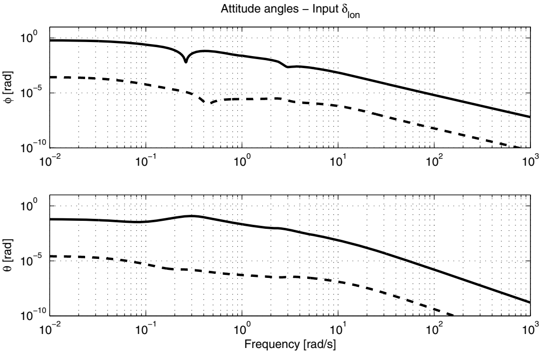


Fig. 11 Frequency response from longitudinal input to attitude angles. (real: solid line; error: dashed line)

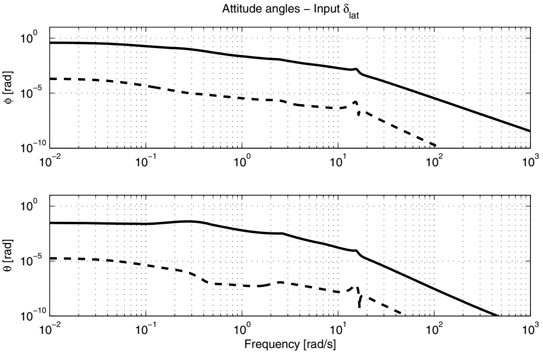


Fig. 12 Frequency response from lateral cyclic input to attitude angles. (real: solid line; error: dashed line)

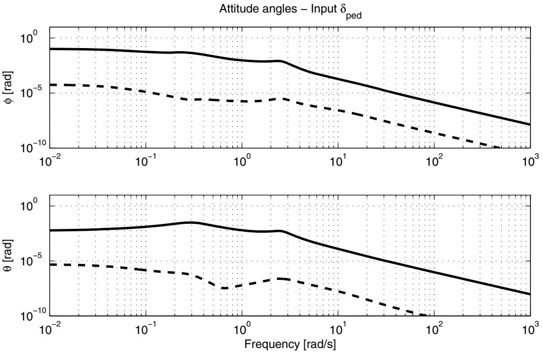


Fig. 13 Frequency response from pedal cyclic input to attitude angles. (real: solid line; error: dashed line)

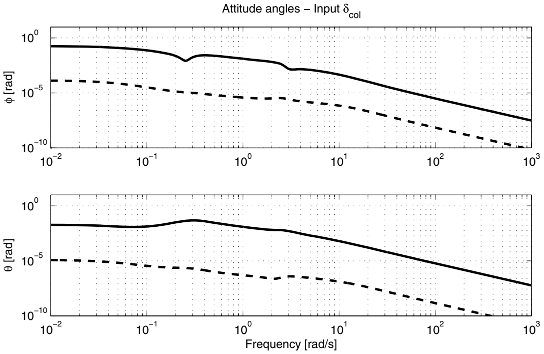


Fig. 14 Frequency response from collective input to attitude angles. (real: solid line; error: dashed line)

5 Concluding Remarks

The problem of rotorcraft system identification has been considered and a two step technique combining the advantages of time domain and frequency domain methods has been proposed. A simulation study based on a model of the BO-105 helicopter has been used to illustrate the proposed approach. Simulation results show that the proposed schemes are viable for rotorcraft applications and can deal successfully with data generated during closed-loop experiments. Future work will focus on the analysis of the impact on the solution of (35) of an identified model that has been obtained under noisy conditions.

References

1. Apkarian, P., Noll, D.: Nonsmooth H_∞ synthesis. IEEE Transactions on Automatic Control 51(1), 71–86 (1996)
2. Bergamasco, M.: Continuous-time model identification with applications to rotorcraft dynamics. PhD thesis, Politecnico di Milano (2012)
3. Bergamasco, M., Lovera, M.: Recovering structured models from unstructured ones: an H_∞ approach (submitted)
4. Bergamasco, M., Lovera, M.: Continuous-time subspace identification in closed-loop. In: 19th International Symposium on Mathematical Theory of Networks and Systems, Budapest, Hungary (2010)
5. Bergamasco, M., Lovera, M.: Continuous-time predictor-based subspace identification for helicopter dynamics. In: 37th European Rotorcraft Forum, Gallarate, Italy (2011)
6. Bergamasco, M., Lovera, M.: Continuous-time predictor-based subspace identification using Laguerre filters. IET Control Theory and Applications 5(7), 856–867 (2011); Special issue on Continuous-time Model Identification
7. Bittanti, S., Lovera, M.: Identification of linear models for a hovering helicopter rotor. In: Proceedings of the 11th IFAC Symposium on system identification, Fukuoka, Japan (1997)

8. Bittanti, S., Lovera, M.: Bootstrap-based estimates of uncertainty in subspace identification methods. Automatica 36(11), 1605–1615 (2000)
9. Chiuso, A., Picci, G.: Consistency analysis of certain closed-loop subspace identification methods. Automatica 41(3), 377–391 (2005)
10. Gahinet, P., Apkarian, P.: Decentralized and fixed-structure H_∞ control in matlab. In: 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, USA (2011)
11. Huang, B., Ding, S.X., Qin, S.J.: Closed-loop subspace identification: an orthogonal projection approach. Journal of Process Control 15(1), 53–66 (2005)
12. Jategaonkar, R.: Flight Vehicle System Identification. AIAA (2006)
13. Johansson, R., Verhaegen, M., Chou, C.T.: Stochastic theory of continuous-time state-space identification. IEEE Transactions on Signal Processing 47(1), 41–51 (1999)
14. Kinoshita, Y., Ohta, Y.: Continuous-time system identification using compactly supported filter kernels generated from Laguerre basis functions. In: 49th IEEE Conference on Decision and Control, Atlanta, USA (2010)
15. Li, P., Postlethwaite, I.: Subspace and bootstrap-based techniques for helicopter model identification. Journal of the American Helicopter Society 56(1), 012002 (2011)
16. Lovera, M.: Identification of MIMO state space models for helicopter dynamics. In: 13th IFAC Symposium on System Identification, Rotterdam, The Nederlands (2003)
17. Ohta, Y.: Realization of input-output maps using generalized orthonormal basis functions. Systems & Control Letters 22(6), 437–444 (2005)
18. Ohta, Y.: System transformation of unstable systems induced by a shift-invariant subspace. In: 50th IEEE Conference on Systems and Control (2011) (submitted)
19. Ohta, Y., Kawai, T.: Continuous-time subspace system identification using generalized orthonormal basis functions. In: 16th International Symposium on Mathematical Theory of Networks and Systems, Leuven, Belgium (2004)
20. Tischler, M., Cauffman, M.: Frequency-response method for rotorcraft system identification: Flight applications to BO-105 coupled rotor/fuselage dynamics. Journal of the American Helicopter Society 37(3), 3–17 (1992)
21. Tischler, M., Remple, R.: Aircraft And Rotorcraft System Identification: Engineering Methods With Flight-test Examples. AIAA (2006)
22. Van Overschee, P., De Moor, B.: Subspace identification: theory, implementation, application. Kluwer Academic Publishers (1996)
23. Verhaegen, M.: Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. Automatica 30(1), 61–74 (1994)
24. Verhaegen, M., Varga, A.: Some experience with the MOESP class of subspace model identification methods in identifying the BO105 helicopter. Technical Report TR R165-94, DLR (1994)
25. Verhaegen, M., Verdult, V.: Filtering and System Identification: A Least Squares Approach. Cambridge University Press (2007)
26. Zhou, K., Doyle, J., Glover, K.: Robust and optimal control. Prentice Hall (1996)

Air Data Sensor Fault Detection Using Kinematic Relations

Laurens van Eykeren and Qiping Chu

Abstract. This paper presents a Fault Detection and Isolation (FDI) method for Air Data Sensors (ADS) of aircraft. In the most general case, fault detection of these sensors on modern aircraft is performed by a logic that selects one of, or combines three redundant measurements. Such a method is compliant with current airworthiness regulations. However, in the framework of the global aircraft optimization for future and upcoming aircraft, it could be required, e.g. to extend the availability of sensor measurements. So, an improvement of the state of practice could be useful. Introducing a form of analytical redundancy of these measurements can increase the fault detection performance and result in a weight saving of the aircraft because there is no necessity anymore to increase the number of sensors. Furthermore, the analytical redundancy can contribute to the structural design optimization. The analytical redundancy in this method is introduced using an adaptive form of the Extended Kalman Filter (EKF). This EKF uses the kinematic relations of the aircraft and makes a state reconstruction from the available measurements possible. From this estimated state, an estimated output is calculated and compared to the measurements. Through observing a metric derived from the innovation of the Extended Kalman Filter (EKF), the performance of each of the redundant sensors is monitored. This metric is then used to automatically isolate the failing sensors.

1 Introduction

In this paper a newly developed architecture for Air Data Sensors (ADS) monitoring is proposed. The method deals with the Fault Detection and Isolation (FDI) of measurements required for the Electronic Flight Control System (EFCS) of aircraft and is part of the work performed for the Advanced Fault Diagnosis for Sustainable Flight Guidance and Control (ADDSAFE) project. The goal of the

Laurens van Eykeren · Qiping Chu
Delft University of Technology, Kluyverweg 1, 2629HS Delft, Netherlands
e-mail: {l.vaneykeren,q.p.chu}@tudelft.nl

ADDSAFE project is to research and develop model-based Fault Detection and Diagnosis (FDD) methods for aircraft flight control systems, mainly sensor and actuator malfunctions [12]. Furthermore, the ADDSAFE project aims at closing the gap between the academic field of research of FDD and the practical application of these methods in industry.

1.1 Motivation

In Fig. 1 an overview is given of the typical architecture of EFCS of an aircraft. As can be noticed, one way of how faults can be introduced in the control loop is by sensor faults, indicated as Air Data and Inertial Reference System (ADIRS) faults in the figure. Faulty measurements which are fed back to the flight control laws can create unwanted control signals, leading e.g. to higher loads on the aircraft structure. For that reason, the aircraft structures are designed to withstand these unwanted loads up to a level at which it is guaranteed that the faults can be detected and appropriate actions can be taken.

However, for upcoming and future aircraft one important aspect is the structural design optimization. This can lead to a substantial decrease in the weight of the aircraft, which again leads to an increase in the aircraft's performance, including a decrease in fuel consumption, a decrease of produced noise and an increased range. Furthermore, these advantages also satisfy the newer societal imperatives toward an environmentally friendlier aircraft.

Sensor fault detection for flight parameter measurements, like e.g. air data and inertial measurements in modern aircraft is generally achieved through the use of typically three redundant measurement units (e.g. Air Data and Inertial Reference Units (ADIRUs) [22]). Through a decision logic, also called consolidation process, the correct measurement is selected and used by the EFCS [10],[22].

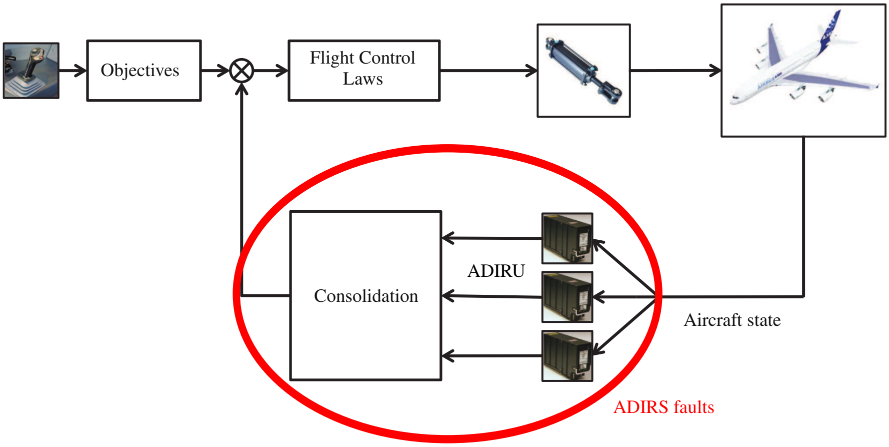


Fig. 1 Flight control architecture of an aircraft

Improving the FDD performance of the aircraft's EFCS allows to optimize the aircraft structural design and performance, resulting in a lower operating cost and decreased environmental impact [13], as explained above.

Another motivation for the development of analytical redundancy for aircraft parameter measurements is to extend the availability of the sensor measurements. Instead of adding one or several new sensors, the option of adding a "virtual" sensor, i.e. analytical redundancy, gives the advantage no additional weight is required. This results again in the same advantages as described in the previous paragraph.

These two main reasons indicate the need to create new advanced FDD methods and to close the gap between academic research and industrial application.

1.2 Antecedents and Main Contribution

In this work a model-based FDD approach is presented for the fault detection of Air Data Sensors (ADS), in particular applied to the angle-of-attack measurements. Different methods have been investigated for mitigating the effects of failing ADS, such as: signal based diagnosis [16],[9], alternative sensing methods which are fault tolerant [4], robust fault detection approaches [11], finding ways to operate without traditional ADS[6]. Other solutions for the problem of ADIRS monitoring dealing with oscillatory faults are presented in [3],[2].

In this paper a method is introduced based on the general kinematic relations of aircraft. By relating different available measurements in the ADIRU, it becomes possible to perform FDD of the ADS. For this purpose, an adaptive modification of the EKF is applied to the kinematic equations. The Kalman Filter (KF) and its numerous modifications have been used in the field of aerospace engineering since it was developed in the 1960s [14]. In this way, the EKF has also been used for sensor fault detection [7].

The EKF was originally formulated for state estimation of dynamic systems when the dynamics and measurement equations are nonlinear, but linearizable [17] and has been widely used for sensor monitoring and fusion techniques [1]. The method that will be proposed here directly builds on this principle, i.e., using the redundant measurements available form the multiple ADIRUs the state of the aircraft is reconstructed by means of a adaptive version of an EKF which was first introduced by [19]. [15] proposes sensor fault detection by evaluating the innovation sequence of the filter. This information can furthermore be used to fuse the measurements in such a way that failing sensors are detected and isolated.

1.3 Structure of the Paper

In the next section the FDD problem to solve is introduced, giving the system description and the fault scenarios. In Section 3 the proposed FDD method is described and in Section 4 the simulation results are presented. The paper ends with a conclusion in Section 5.

2 Problem Definition

2.1 System Definition

One of the key elements of this method is the use of the kinematic equations that describe the aircraft's behavior, i.e., the state reconstruction is achieved using the measurements of the Inertial Reference Unit (IRU). When the load factors and the rotational rates are used as inputs to the EKF, the state of the aircraft can be reconstructed ([5]). The big advantage of this approach lies in the following three points:

1. The method developed is valid over the whole flight envelope of the aircraft. This means that no special measures need to be taken such as gain scheduling, etc.
2. Secondly, the method can be applied to any aircraft, without large modifications (except for the location of the sensors). So the developed method is general for aircraft.
3. The method is insensitive to other types of faults, e.g. actuator faults, control surface jamming, etc.

The aircraft kinematics can be represented by the following nonlinear system:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t)) + \mathbf{G}(\mathbf{x}(t))\,(\mathbf{u}(t)+\mathbf{w}(t))\\ \mathbf{z}(t) &= \mathbf{C}\mathbf{x}(t)\\ \mathbf{z}_m(t) &= \mathbf{C}\mathbf{x}(t)+\mathbf{v}(t)\end{aligned} \tag{1}$$

Where $\mathbf{x}$ represents the state of the system, $\mathbf{u}$ the input and $\mathbf{z}$ the measurable output of the system. $\mathbf{w}(t)$ and $\mathbf{v}(t)$ represent Gaussian white noise sequences and are the measurement noise of respectively the measured input and output. In this particular case, the state description can be reduced to a five state system, and these states are measurable:

$$\mathbf{x} = \begin{bmatrix} V_{TAS} \ \alpha \ \beta \ \phi \ \theta \end{bmatrix}^\top \tag{2}$$

$$\mathbf{u} = \begin{bmatrix} A_x \ A_y \ A_z \ p \ q \ r \end{bmatrix}^\top \tag{3}$$

$$\mathbf{z} = \begin{bmatrix} V_{TAS} \ \alpha \ \beta \ \phi \ \theta \end{bmatrix}^\top \tag{4}$$

Where V_{TAS} is the true airspeed, α the angle-of-attack, β the side-slip angle, ϕ the roll angle and θ the pitch angle. A_x, A_y, and A_z are the accelerations at the center of gravity, p, q, and r the rotational rates. Note that a transformation is necessary to convert the measured load factors at the IRU to accelerations at the center of gravity. Furthermore, note that $\mathbf{C} = \mathbf{I}$. In fact, both the inputs to this system and the outputs are measured from the aircraft and can be assumed available in the EFCS for each modern aircraft. Although the position of an aircraft can be considered a part of the state, it is not required for the purpose of fault detection of the ADS. Having only these five states, will decreases the computational load of the proposed method. Furthermore, no wind influences are accounted for in this work. However, according to [20] it is possible to estimate the wind, giving a more precise estimate of the state of the aircraft if necessary, at the cost of a larger state vector and so increased computational load.

According to [8], the kinematic state update equations can be described by:

$$\begin{aligned}
\dot{V} &= g\,(-\sin\theta\cos\alpha\cos\beta + \sin\phi\cos\theta\sin\beta + \cos\phi\cos\theta\sin\alpha\cos\beta) \\
&\qquad + A_x\cos\alpha\cos\beta + A_y\sin\beta + A_z\sin\alpha\cos\beta \qquad (5) \\
\dot{\alpha} &= \frac{g}{V\cos\beta}\,(\cos\phi\cos\theta\cos\alpha + \sin\theta\sin\alpha) \\
&\qquad + \frac{1}{V\cos\beta}\,(A_z\cos\alpha - A_x\sin\alpha) + q - \tan\beta\,(p\cos\alpha + r\sin\alpha) \qquad (6) \\
\dot{\beta} &= \frac{g}{V}\,(\sin\theta\cos\alpha\sin\beta + \sin\phi\cos\theta\cos\beta - \cos\phi\cos\theta\sin\alpha\sin\beta) \\
&\qquad + \frac{1}{V}\,(-A_x\cos\alpha\sin\beta + A_y\cos\beta - A_z\sin\alpha\sin\beta) + p\sin\alpha - r\cos\alpha \qquad (7) \\
\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \qquad (8) \\
\dot{\theta} &= q\cos\phi - r\sin\phi \qquad (9)
\end{aligned}$$

which defines $\mathbf{f}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ in Eq. (1).

2.2 Fault Scenario

The definition for the fault scenario follows from the ADDSAFE project [12]. All faults investigated in this paper are related to the measurement of the angle-of-attack α, however the method developed can be extended to the monitoring of the measurements of the true airspeed V_{TAS} and the side-slip angle β, without losing generality. Different types of faults are considered, such as oscillating faults, runaway faults and increased noise faults, of which examples are shown in Fig. 2. Note that in this graph, and all other graphs in this paper, all values are normalized to the operational range of the measurements. Furthermore, also the time axis will be normalized for each simulation.

Each of the different type of faults can occur on one or simultaneously on two sensors. Whereas the fault detection of the case of only one failing sensor is a trivial task, the fault detection when two sensors fail at the same time is less obvious without incorporating any kind of analytical redundancy. An overview of the different faults investigated and there amplitudes is shown in Table 1.

3 FDD Approach

The general idea of the approach taken here is to fuse the redundant measurements based on the quality of the measurement. This is achieved by filtering the available measurements using an EKF and comparing the state estimates with the redundant measurements based on a so-called "*R*-adaptation" [21], which will be explained in Section 3.2.

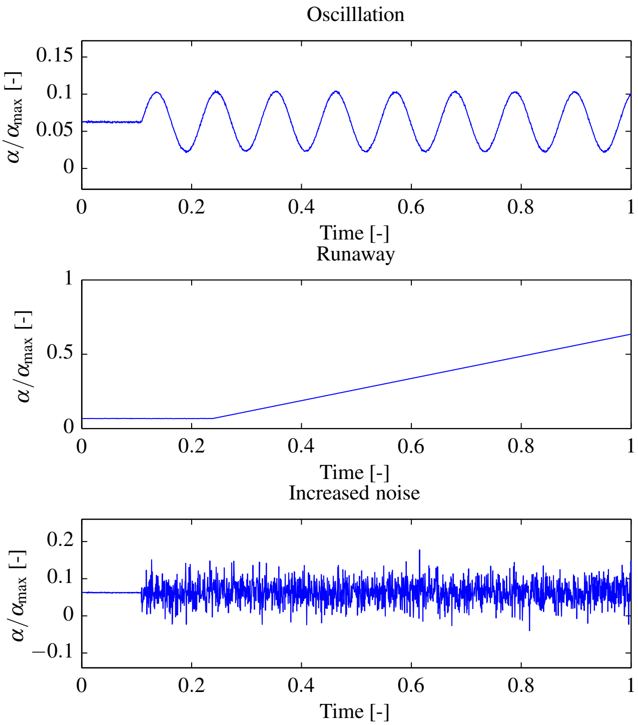


Fig. 2 Different types of faults

Table 1 Fault Amplitudes (for the runaway, the rate is expressed as a percentage of $\alpha_{\max}$, for the extra noise the standard deviation of the noise as a percentage of $\alpha_{\max}$)

Scenario	Fault type	Amplitude (% of $\alpha_{\max}$)
1	Oscillation 1 sensor	4
2	Oscillation 2 sensors	11
3	Runaway slow 2 sensors	9
4	Runaway fast 2 sensors	33
5	Extra noise 1 sensor	2
6	Extra noise 2 sensors	14

For this purpose, first the basic principles of the EKF are briefly explained, which is essential in understanding the method. Then the sensor monitoring algorithm is addressed, which is used to perform the FDD.

3.1 Extended Kalman Filter

The standard EKF exists of two main steps. The first step can be called the prediction of the estimated mean of the state of the system, and uses the system dynamic equations. Also the covariance of the estimate is predicted. This step can be represented by:

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \int_{t_{k-1}}^{t_k} \left[\mathbf{f}(\hat{\mathbf{x}}(\tau)) + \mathbf{G}(\hat{\mathbf{x}}(\tau))\mathbf{u}_m(\tau)\right] \mathrm{d}\tau \tag{10}$$

$$\mathbf{P}_{k|k-1} = \Phi_k \mathbf{P}_{k-1|k-1} \Phi_k^\top + \mathbf{Q}_d \tag{11}$$

Where $\hat{\mathbf{x}}_{k|k-1}$ is the estimated state at time $t = t_k$, knowing the measurement at time $t = t_{k-1}$. $\mathbf{u}_m(t)$ represents the measured input to the system. The matrix $\mathbf{P}_{k-1|k-1}$ represents the covariance matrix of the estimated state at time $t = t_{k-1}$. The matrix Φ_k is the discretized version of the Jacobian matrix $\mathbf{F}_k$, both defined as follows:

$$\Phi_k = e^{\mathbf{F}_k \Delta t} = \sum_n^\infty \frac{\mathbf{F}_k^n (\Delta t)^n}{n!} \tag{12}$$

$$\text{with: } \mathbf{F}_k = \left.\frac{\partial\left(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \tag{13}$$

and from [18] we can approximate $\mathbf{Q}_d$ as:

$$\mathbf{Q}_d(k) = \Gamma_k \mathbf{Q} \Gamma_k^\top \tag{14}$$

$$\text{with: } \Gamma_k = \left(\int_{k-1}^{k} \Phi_k \Delta t\right) \mathbf{G}(\hat{\mathbf{x}}_{k|k}) \tag{15}$$

where $\mathbf{Q} = E\left[\mathbf{w}(t)\mathbf{w}^\top(t)\right]$ represents the input noise covariance matrix.

The second step is the measurement update. It is represented by:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^\top \left(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}\right)^{-1} \tag{16}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_m - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\right) \tag{17}$$

$$\mathbf{P}_{k|k} = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right] \mathbf{P}_{k|k-1} \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]^\top + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^\top \tag{18}$$

Where $\mathbf{K}$ is the Kalman gain, $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \mathbf{I}$, and $\mathbf{R} = E\left[\mathbf{v}(t)\mathbf{v}^\top(t)\right]$ the measurement noise covariance matrix. Furthermore, from these equations we can define the innovation as $\mathbf{z}_{m_k} - \hat{\mathbf{z}}_k$ and the innovation covariance matrix as:

$$\mathbf{V}_{e_k} = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R} \tag{19}$$

This standard EKF can be applied to the system described in Section 2.1 with the triple measurement of the angle-of-attack α augmented in the measurement vector. Note that is was chosen for this approach in favor of a dedicated filter for each ADIRU, as to reduce the computational load. In this case, a simulation was chosen with a double runaway fault, i.e., sensor 1 and 2 experienced the same runaway fault at $t = 0.03$. In Fig. 3 the result of the estimated angle-of-attack ($\hat{\alpha}$) can be seen compared to the three different measurements. As can be noticed, the estimated value of the angle-of-attack is in between the measured values. This is logical, as the assigned variances to the different sensors, through the matrix **R**, are equal. Therefore, each measurement of the same variable is equally weighted by the filter. From the figure it is clear that it cannot be decided on this information which sensor is failing, and which sensor is providing a correct value. In this we find a motivation to modify the algorithm such that FDI becomes possible by monitoring the performance of the sensors.

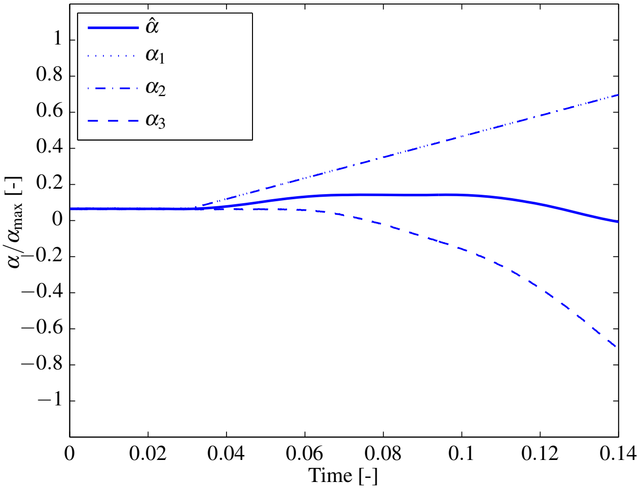


Fig. 3 $\hat{\alpha}$ compared with measurements for runaway fault of sensor 1 and 2 (equal fault value), regular EKF

3.2 Adaptive Fusion

Instead of using all redundant measurements as separate observations in an adaptive EKF [23], here is chosen to fuse the redundant measurements based on their performance. For this, a certain metric is introduced which represents the performance (fault-free/fault) of the sensor.

The theoretical innovation covariance of the EKF is represented by (19). This value can also be estimated online:

$$\hat{\mathbf{V}}_{e_k} = \frac{1}{N} \sum_{i=k-N+1}^{k} (\mathbf{z}_{m_i} - \hat{\mathbf{z}}_i)(\mathbf{z}_{m_i} - \hat{\mathbf{z}}_i)^{\top} \tag{20}$$

where N represents the moving window width. In the case a sensor fails, for each diagonal value, the estimated value will exceed the theoretical value, i.e.:

$$\hat{V}_{e_{k_{(i,i)}}} \geq \left[\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\top}\right]_{(i,i)} + R_{(i,i)} \tag{21}$$

where the subscript $\ldots_{(i,i)}$ means the value on the ith diagonal. Now one can introduce the diagonal scale factor matrix $\mathbf{S}(k)$ ([21]) such that:

$$\hat{V}_{e_{k_{(i,i)}}} = \left[\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\top}\right]_{(i,i)} + S_{k_{(i,i)}} R_{(i,i)} \tag{22}$$

And therefore the values of $\mathbf{S}(k)$ can be calculated as:

$$S_{k_{(i,i)}} = \left(\hat{V}_{e_{k_{(i,i)}}} - \left[\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\top}\right]_{(i,i)}\right) R^{-1}_{(i,i)} \tag{23}$$

In the fault free case, the matrix $\mathbf{S}(k)$ will approximate the unity matrix $\mathbf{I}$, in a faulty case, the diagonal value related to the failing sensor will increase and become bigger than 1. To perform the fusion of the redundant measurements, the scale factors are calculated for the different sensors. Then a weighted average of the three measurements is taken using the reciprocals of $S(k)$ as weights:

$$\alpha_c = \frac{1}{\sum_{i=1}^{3} \frac{1}{S^*_{\alpha_i}}} \sum_{i=1}^{3} \frac{1}{S^*_{\alpha_i}} \alpha_i \tag{24}$$

As can be noticed, in the case one or two sensors give a bad measurement, the related value of $S(k)$ will increase and the faulty measurement will be given a lower weight. As will be shown in the results, the scale factor of the faulty measurement is much larger than 1, so $S(k) \gg 1$, and as such the faulty measurements will have almost no influence on α_c. The detection and isolation signal will then be based on whether the value of $S(k)$ will exceed a preset threshold. Another approach would be to set a threshold, and disregard any measurement with a scale factor above this threshold.

Note that the method as presented here is limited to the detection of measurement faults related to the signals in $\mathbf{z}$, as defined in Eq. (4). It is assumed that the input measurements $\mathbf{u}$ are fault-free.

4 Simulation Results

The method described above is applied to the system described in Section 2.1. Simulations were run on the ADDSAFE benchmark. Two main tuning parameters are required to be determined for the application of the filter. First of all, there is the time window N, over which the estimate of the innovation covariance is calculated. This parameter depends on the system dynamics and the required detection performance. However, setting this parameter is trivial, and is done by trial and error. The second tuning parameter is the threshold T introduced in the previous paragraph. This parameter can be set based on the amplitude of the residuals in fault-free cases. Although the matrices $\mathbf{Q}$ and $\mathbf{R}$ can be considered as tuning variables, they are related to the performance of the sensors measuring the input and output vectors $\mathbf{u}$ and

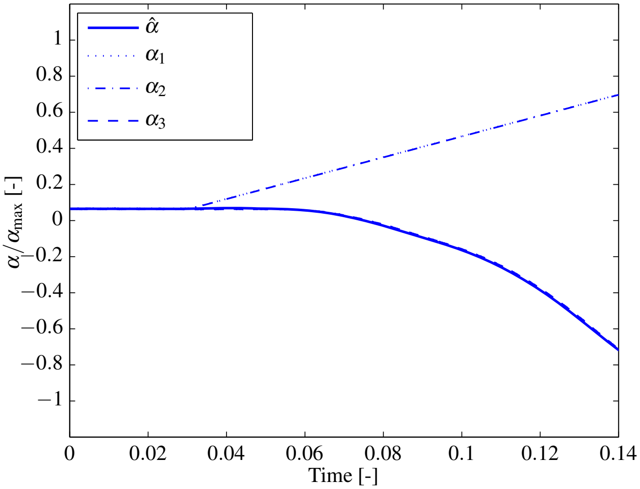


Fig. 4 $\hat{\alpha}$ compared with measurements for runaway fault of sensor 1 and 2 (equal fault value), EKF with adaptive fusion

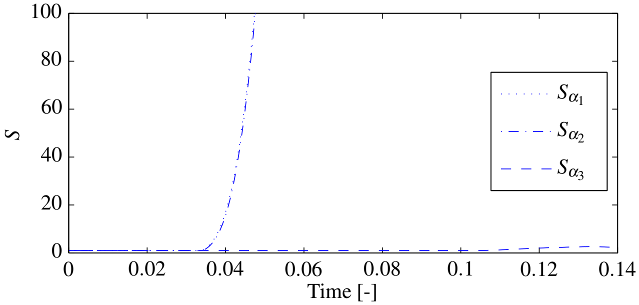


Fig. 5 S_α for the different sensors for runaway fault of sensor 1 and 2 (equal fault value), EKF with adaptive fusion

z. Therefore, both matrices should be based on the real sensor performances which are considered to be known.

First, the method was applied to the same simulation as shown in Fig. 3. The result is shown in Fig. 4. As can be noticed, the estimated $\hat{\alpha}$ now follows the correct measurement α_3 and α_1 and α_2 are discarded. Fig. 5 shows the values of S_{α_i}, $i = 1,2,3$. The scale factors related to sensor 1 and 2 clearly show an increase in value after the fault occurred. Other typical results for the different scenarios described in Table 1 are shown in Figs. 6, 7, 8 and 9.

A simulation campaign was set-up to test the proposed FDI method. This campaign existed of fault-free simulations in which different maneuvers were performed

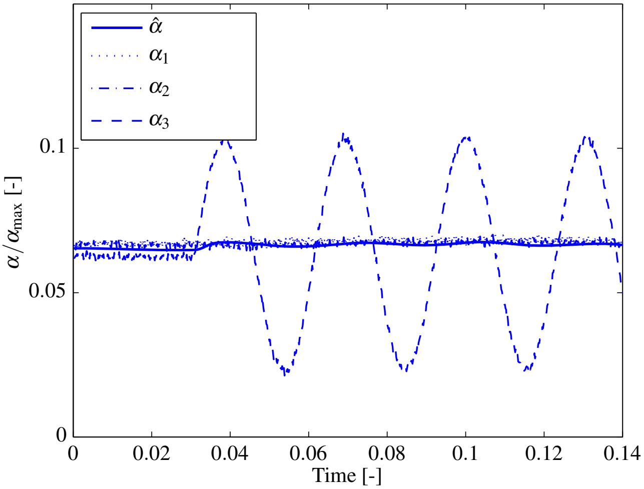


Fig. 6 $\hat{\alpha}$ compared with measurements for oscillatory fault of sensor 3, EKF with adaptive fusion

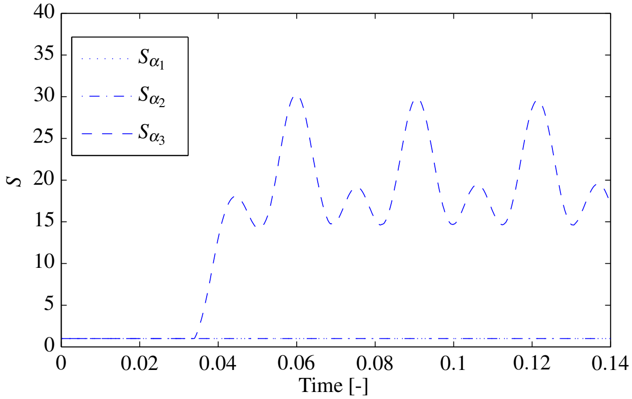


Fig. 7 S_α for the different sensors for oscillatory fault of sensor 3, EKF with adaptive fusion

to test the false alarm rate. These simulations included sudden pitch up maneuvers with high angle-of-attack attitudes and lateral maneuvers including substantial side slipping of the aircraft. The introduction of faults occurred during simulations of the cruise condition of the aircraft, for the faults presented in Table 1.

The simulation campaign involved changing the following parameters of the simulations: the flight parameters (altitude, velocity), geometric parameters (mass, position of center of gravity), uncertainties in the measurements (mass, velocity, center of gravity, altitude) and the aerodynamic coefficients. All these parameters were

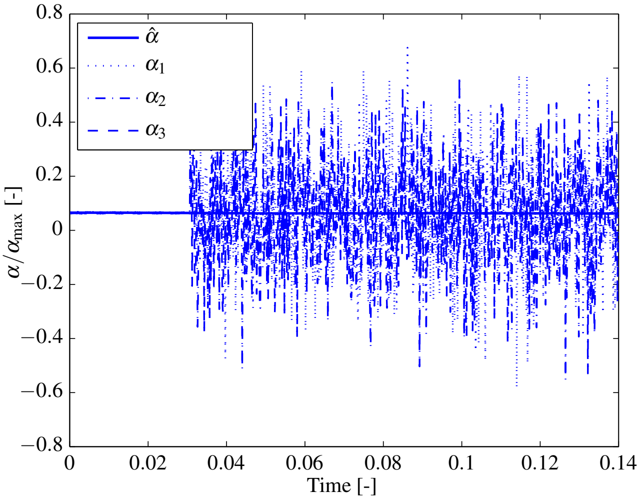


Fig. 8 $\hat{\alpha}$ compared with measurements for "noise" fault of sensor 1 and 2, EKF with adaptive fusion

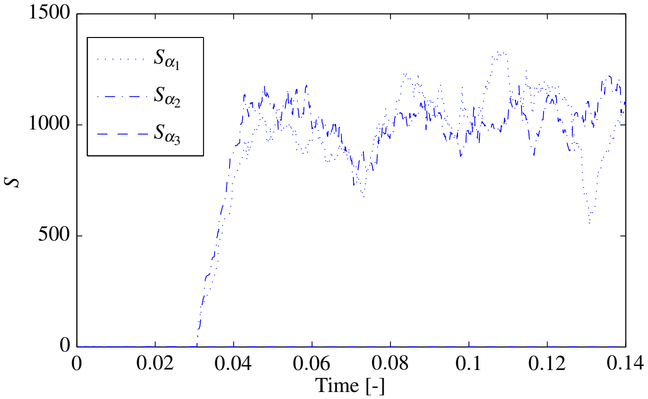


Fig. 9 S_α for the different sensors for "noise" fault of sensor 1 and 2, EKF with adaptive fusion

adjusted in two different ways: using the extreme values (all uncertainties on the minimum or maximum value at the same time) and using Monte Carlo simulations. In this way, a large part of the flight envelope of the aircraft was covered.

In Table 2 an overview is given of the detection performance of these simulations, consisting of 252 simulations (152 parametric variations and 100 Monte Carlo variations) for each scenario. In this table, "DTP" stands for "Detection Time Performance" and is expressed in function of the maximal allowed detection time for that

Table 2 Summary of obtained simulation results

#	Detection (%)	DTP mean	DTP max	DTP min
1	100	0.0063	0.0068	0.0062
2	100	0.16	0.18	0.14
3	100	0.41	0.95	0.32
4	100	0.16	0.20	0.13
5	100	0.0035	0.0042	0.0030
8	100	0.06	0.06	0.06

type of fault. As can be noticed, for the different fault scenarios considered, a 100% fault detection performance was achieved, i.e., no missed detections and no late detections. The low values for scenario 1 and 5 are due to different maximum allowed detection times. It can be noted that the absolute detection times for both one or two sensors failing were in the same magnitude order, i.e., the absolute detection time is not influenced by the amount of sensors (one or two) that are failing. Furthermore, no false alarms were obtained during the simulation of fault-free maneuvers. Here only results are presented for the detection of faults in the angle-of-attack sensors. However, it should be noted that this FDI method can be extended to the other variables in the measurement vector $\mathbf{z}$ without losing any functionality. As these types of faults can be detected and accounted for by the same methodology, it can be stated that the fault detection of one specific variable is insensitive to other faults. The detection of measurement faults of the variables in the input vector $\mathbf{u}$, being the accelerations and rotational rates, is not considered in this work, and so these measurements are assumed to be fault-free, i.e., FDI for these measurements is covered by a different methodology.

5 Conclusion

This paper presented an algorithm based on an adaptive modification to the EKF that is capable of providing mathematical redundancy for the purpose of sensor fault detection. The main advantages of this method are the independence from the dynamics of the aircraft and it's low tuning complexity. In fact, the only aircraft specific knowledge required is the exact location of the IRU and the sensor performance characteristics. Because only kinematic and no dynamic (forces and moments) relations are used, no special measures need to be taken to make to method valid over the whole flight envelope of the aircraft. This results in a very low tuning complexity, limited to setting a time window and one threshold. Furthermore, it should be noted that this method can be extended to other air data measurements, which will be investigated in future work.

References

1. Allerton, D.J., Jia, H.: A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems. Journal of Navigation 58(3), 405 (2005), doi:10.1017/S0373463305003383
2. Berdjag, D., Cieslak, J., Zolghadri, A.: Fault diagnosis and monitoring of oscillatory failure case in aircraft inertial system. Control Engineering Practice 20(12), 1410–1425 (2012), http://www.sciencedirect.com/science/article/pii/S0967066112001682
3. Berdjag, D., Zolghadri, A., Cieslak, J., Goupil, P.: Fault detection and isolation for redundant aircraft sensors. In: 2010 Conference on Control and Fault-Tolerant Systems (SysTol), vol. 1, pp. 137–142. Ieee, Nice (2010), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5675993 , doi:10.1109/SYSTOL.2010.5675993
4. Cervia, F., Denti, E., Galatolo, R., Schettini, F.: Air Data Computation in Fly-By-Wire Flight-Control Systems. Journal of Aircraft 43(2), 450–455 (2006), http://arc.aiaa.org/doi/abs/10.2514/1.16270, doi:10.2514/1.16270
5. Chu, Q.P., Mulder, J.A., Sridhar, J.K.: Decomposition of Aircraft State and Parameter Estimation Problems. In: Proceedings of the 10th IFAC Symposium on System Identification, vol. 3, pp. 61–66. Danish Automation Society, Copenhagen (1994)
6. Colgren, R., Frye, M., Olson, W.: A Proposed System Architecture for Estimation of Angle-Of-Attack and Sideslip Angle. In: AIAA Guidance, Navigation and Control Conference, pp. 743–750. AIAA (1999)
7. Del Gobbo, D., Napolitano, M., Famouri, P., Innocenti, M.: Experimental application of extended Kalman filtering for sensor validation. IEEE Transactions on Control Systems Technology 9(2), 376–380 (2001), doi:10.1109/87.911389
8. Duke, E.L., Antoniewicz, R.F., Krambeer, K.D.: Derivation and Definition of a Linear Aircraft model. Tech. rep., NASA (1988)
9. Eubank, R.D., Atkins, E.M., Ogura, S.: Fault Detection and Fail-Safe Operation with a Multiple-Redundancy Air-Data System. In: AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, pp. 1–14 (August 2010)
10. Favre, C.: Fly-by-wire for commercial aircraft: the Airbus experience. International Journal of Control 59(1), 139–157 (1994)
11. Freeman, P., Seiler, P., Balas, G.J.: Robust Fault Detection for Commercial Transport Air Data Probes. In: 18th IFAC World Congress. IFAC, Milan (2011)
12. Goupil, P., Marcos, A.: Advanced Diagnosis for Sustainable Flight Guidance and Control: The European ADDSAFE Project. SAE Technical Paper 2011-01-2804 (2011), doi:10.4271/2011-01-2804
13. Goupil, P., Marcos, A.: Industrial benchmarking and evaluation of ADDSAFE FDD designs. In: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Mexico City, Mexico (2012)
14. Grewal, M.S., Andrews, A.P.: Applications of Kalman Filtering in Aerospace 1960 to the Present. IEEE Control Systems Magazine 30(3), 69–78 (2010)
15. Hajiyev, C.: Testing the covariance matrix of the innovation sequence with sensor / actuator fault detection applications. International Journal of Control and Signal Processing 24(9), 717–730 (2010), doi:10.1002/acs.1160
16. Houck, D., Atlas, L.: Air Data Sensor Failure Detection. In: 17th Digital Avionics Systems Conference. AIAA/IEEE/SAE (May 1979, 1998)
17. Jazwinski, A.H.: Stochastic Processes and Filtering Theory. Academic Press, New York (1970)

18. Lombaerts, T.J.J.: Fault Tolerant Flight Control, A Physical Model Approach. Phd thesis, Delft University of Technology (2010)
19. Mehra, R.K., Peschon, J.: An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems. Automatica 7, 637–640 (1971)
20. Mulder, J.A., Chu, Q.P., Sridhar, J.K., Breeman, J.H., Laban, M.: Non-linear aircraft flight path reconstruction review and new advances. Progress in Aerospace Sciences 35(7), 673–726 (1999)
21. Soken, H.E., Hajiyev, C.: Fault Tolerant Attitude Estimation for Pico Satellites Using Robust Adaptive UKF. In: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes. IFAC (2012)
22. Traverse, P., Lacaze, I., Souyris, J.: Airbus fly-by-wire: A total approach to dependability. In: Proceedings of the 18th IFIP World Computer Congress, Toulouse, France, pp. 191–212 (2004)
23. Van Eykeren, L., Chu, Q.P., Mulder, J.A.: Sensor Fault Detection and Isolation using Adaptive Extended Kalman Filter. In: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp. 1155–1160. IFAC, Mexico City (2012)

A Multiple-Observer Scheme for Fault Detection, Isolation and Recovery of Satellite Thrusters

Antoine Abauzit and Julien Marzat

Abstract. The method proposed in this paper aims at automatically detecting, isolating and identifying faults on actuators of a satellite model and also aims at automatically reconfiguring the reference input once the fault has been isolated. The method uses two sliding mode observers to detect and reconstruct the fault. A cusum test on the output of the detection observer triggers a bank of Unknown Input Observers in order to isolate the faulty actuator. The reference input is automatically reconfigured in order to pre-compensate the fault, which makes the satellite capable of fulfilling its mission with the desired performances and good precision. Monte Carlo analysis, based on performance criteria, is carried out to assess the performance of the strategy. The combination of these different types of filters might provide better detection, isolation and identification capabilities than a single filter that would be forced to achieve a trade-off between fast detection and accurate estimation.

1 Introduction

During the last decades, fault detection, isolation and recovery (FDIR) has met a growing interest in the scientific community. The higher levels of automation expected from modern systems require a higher reliability. Hardware redundancy is usually employed to achieve this reliability yet it implies added complexity and higher costs. For satellite systems, hardware redundancy is particularly cumbersome since each actuator should be built several times, and the cumulated mass leads to a much higher launch cost. Furthermore, it is almost impossible to repair

Antoine Abauzit · Julien Marzat
ONERA – The French Aerospace Lab,
F-91123 Palaiseau, France
e-mail: {antoine.abauzit,julien.marzat}@onera.fr

the satellite if a fault occurs after launch. Therefore, methods avoiding hardware redundancy and making it possible for the satellite to fulfil its mission in spite of a fault are highly recommended.

By using a mathematical model of the system, model-based fault diagnosis (also called *analytical redundancy*) can be able to detect and isolate faults on actuators or sensors. Thus, the reliability of the system can rely less heavily on hardware redundancy and more on software efficiency.

Different model-based methods have been studied for satellite models: parity space [10], neural-network [8], parameter estimation techniques [4], observer-based techniques [1], bank of Kalman filters [12] and techniques based on unknown input observer (UIO) [6].

When a fault occurs, the system has to be able to carry on its mission. A Fault-Tolerant Control System (FTCS) is a closed-loop system that has the ability to tolerate faults without threatening its performances or its stability. The reader may refer to the excellent bibliographical review [13], which explains the existing approaches on this topic.

The method presented in this paper aims at automatically detecting a fault on an actuator of a satellite model, isolate it, and reconfigure the control input in order to carry on the mission. This allows the assessment of the performance of a whole FDIR loop on a realistic aerospace model, which is seldom addressed in the literature. The detection and the reconstruction of the fault are achieved via sliding mode observers like in [3]. The obtained residuals are analyzed by cusum tests in a decision-making scheme. Once the fault is detected, a bank of UIOs isolates the actuator on which the fault occurred. An interesting feature of the proposed method is to make use of observers with different dynamics for the detection and estimation of faults to escape from the classical trade-off between reliable detection (few false alarms) and fast estimation. Finally, the reference input is modified in order to compensate for the effect of the fault like in [9]. This way, the dynamics of the feedback laws remain unchanged and a good precision can be achieved, without the need to reconfigure entirely the controller. The performance of the fault diagnosis method is evaluated with Monte-Carlo (MC) simulations.

2 Satellite Modelling

2.1 Satellite Dynamics

The vehicle considered is a deep-space satellite with 12 thrusters (organized in 4 sets), similar to the one presented in [12]. The state of the satellite is described in Eq. (1). $\mathbf{p}$ is the inertial position, $\mathbf{v}$, the inertial velocity, $\mathbf{q}$ the quaternion describing the rotation from the inertial frame to the body frame, and $\boldsymbol{\omega}$ the angular velocity of the satellite.

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix} \tag{1}$$

The sensitivity matrix $\mathbf{B_F}$ represents the force due to each thruster input. The j^{th} column of $\mathbf{B_F}$ refers to the direction of the j^{th} thrust of the satellite, $\mathbf{d}_\text{j}$, described in the body frame.

$$\mathbf{B_F} = [\mathbf{d}_1 \;\; \mathbf{d}_2 \;\; \cdots \;\; \mathbf{d}_{12}] \tag{2}$$

$$\mathbf{B_F} = \begin{bmatrix} 1 & 1 & 1 & 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{3}}{2} & 0 & 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix}$$

The input from the thrusters is defined by the vector **u**. Thereby the input of the i^{th} thruster is the i^{th} component of the vector, u_i. It is a positive scalar value between 0 and 100 N.

In the end, the net force of all the thrusters, in the body frame, simply is:

$$\mathbf{F} = \mathbf{B_F}\mathbf{u} = \sum_{i=1}^{12} \mathbf{d}_i u_i \tag{3}$$

The dynamics of the inertial position and the inertial velocity are described in Eq. (4):

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \frac{1}{m}\mathbf{R}_{\text{B}\to\text{I}}\mathbf{B_F}\mathbf{u} \end{aligned} \tag{4}$$

where $\mathbf{R}_{\text{B}\to\text{I}}$ is the rotation matrix that turns a vector described in the body frame into a vector described in the inertial frame. The mass of the vehicle is denoted by *m*.

The rotational dynamics are generally described by [7]:

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\mathbf{q} \tag{5}$$

$$\boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{6}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\mathbf{B}_{\mathrm{T}}\mathbf{u} - \mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{7}$$

The inertia of the satellite in the body frame is defined as **I**. The sensitivity matrix $\mathbf{B_T}$ represents the torque of each thruster input. The j^{th} column of $\mathbf{B_T}$ refers to the direction of the torque due to the j^{th} thruster in the body frame.

$$\mathbf{B_T} = [\mathbf{d}_{\mathrm{T1}} \ \ \mathbf{d}_{\mathrm{T2}} \ \cdots \ \mathbf{d}_{\mathrm{T12}}] \tag{8}$$

with $\mathbf{d}_{\mathrm{Ti}} = \mathbf{GA}_{\mathrm{i}} \times \mathbf{d}_{\mathrm{i}}$ where A_i is the point where the thrust applies and G is the center of mass of the satellite.

Table 1 Description of the parameters of the satellite model (from [12])

Parameter	Value	Unit
Mass	879	[kg]
Ixx	2787	[kg.m²]
Iyy	2836	[kg.m²]
Izz	2266	[kg.m²]

2.2 Control Allocation

Two state feedback laws are used to control the state of the satellite. The first one is dedicated to the control of the position and the velocity while the second one is for the attitude and the angular velocity. The outputs of the linear controller and the attitude controller are respectively linear and angular accelerations.

A nonlinear iterative control allocation procedure [11] is used to compute the reference thrust for each actuator, in order to respect the commands coming from the feedback laws. The output of the allocator is a desired thrust level for each actuator, which is assumed to be achieved instantaneously.

2.3 Measurement Model

Star trackers assumed to be faultless measure the attitude of the satellite relative to the inertial frame. Let **q** be the actual attitude of the satellite. The measurement of the star trackers is corrupted by a rotation error $\mathbf{q_{err}}$. The measured quaternion $\mathbf{q_m}$ is given by the quaternion product:

$$\mathbf{q}_{\mathrm{m}} = \mathbf{q}_{\mathrm{err}} \otimes \mathbf{q} \tag{9}$$

For the measurement of the angular velocity, the quaternion between the previous measured quaternion $\mathbf{q}_{\mathbf{p}}$ and the current measured quaternion is first computed:

$$\boldsymbol{\delta}\mathbf{q} = \mathbf{q}_{\mathrm{p}}^{-1} \otimes \mathbf{q}_{\mathrm{m}} \tag{10}$$

The attitude variation is:

$$\boldsymbol{\delta}\mathbf{q} = \begin{bmatrix} \cos\dfrac{\varphi}{2} \\ \sin\dfrac{\varphi}{2}\mathbf{e} \end{bmatrix} \tag{11}$$

where **e** is a unit vector that gives the axis of the rotation. In the end, we have an angular increment of φ around the axis **e** between two steps.

The angular velocity estimation is then:

$$\boldsymbol{\omega}_{\mathbf{est}} = \frac{\varphi}{\Delta t}\mathbf{e} \tag{12}$$

where Δt is the time step of the control system. In the following, we assume that the measurement of the angular velocity is given by $\boldsymbol{\omega}_{\mathbf{mes}} = \boldsymbol{\omega}_{\mathbf{est}}$. It is of course worth mentioning that more refined methods may lead to a better estimation of the angular velocity, yet it seemed to be a sufficient modelling level to assess the interest of the proposed method.

3 FDIR Methodology

The objective of the proposed method is to quickly detect, then isolate, and finally compensate a fault on a thruster when it happens. Since two thrusters can have the same force direction but different torque directions, we assume that it is easier to get explicit residuals from the angular velocity than from the linear velocity. If two thrusters have the same force and torque directions, it is necessary to study the linear velocity too in order to determine the sign of the fault and isolate the faulty actuator.

Two sliding mode observers are used to detect and to reconstruct the fault signals. The design of these observers is based on [3]. This kind of non-linear observer generates an output estimate $\hat{\mathbf{y}}$ and a state estimate $\hat{\mathbf{x}}$ such that the estimation error converges to zero in finite time. In [3], this observer is written in the form:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} - \mathbf{G}_\mathbf{l}\mathbf{e}_\mathbf{y} + \mathbf{G}_\mathbf{n}\mathbf{v} \tag{13}$$

where $\mathbf{e}_\mathbf{y} = \hat{\mathbf{y}} - \mathbf{y}$ is the output error, **v** is a non-linear switched function of the output error and $\mathbf{G}_\mathbf{l}$ and $\mathbf{G}_\mathbf{n}$ are gain matrices. It is shown that once the sliding motion $\mathbf{e}_\mathbf{y} = \mathbf{0}$ and $\dot{\mathbf{e}}_\mathbf{y} = \mathbf{0}$ is attained, it becomes possible to estimate actuator or sensor faults from the *equivalent output injection* signal required to maintain sliding motion.

The two observers only differ in their tuning since opposite properties are required for them: one has to provide explicit residuals while the other has to quickly reconstruct the disturbance.

Once a detection flag has been raised, a bank of Unknown Input Observers (UIO) is used to isolate the faulty thruster as in [6].

All these observers only reconstruct the angular velocity of the satellite from $\boldsymbol{\omega}_{\mathbf{mes}}$ and the satellite model. The computational cost is then reduced, since there is no need to process the entire state vector.

Figure 1 depicts the interaction between the satellite dynamics and the on-board algorithms including navigation, state feedback, the control allocation procedure, the FDI functions and the reconfiguration of the input command.

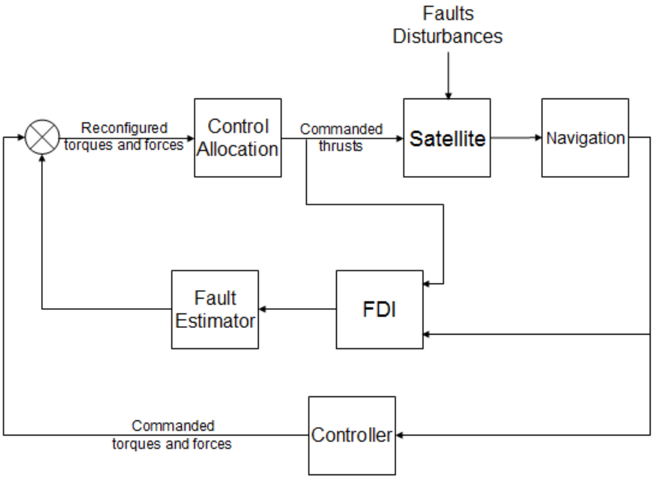


Fig. 1 Control system and FDIR methodology

Figure 2 is a detailed illustration of the FDI block from the previous diagram. It depicts the faulty actuator isolation process and the estimation of the torque disturbance due to the fault.

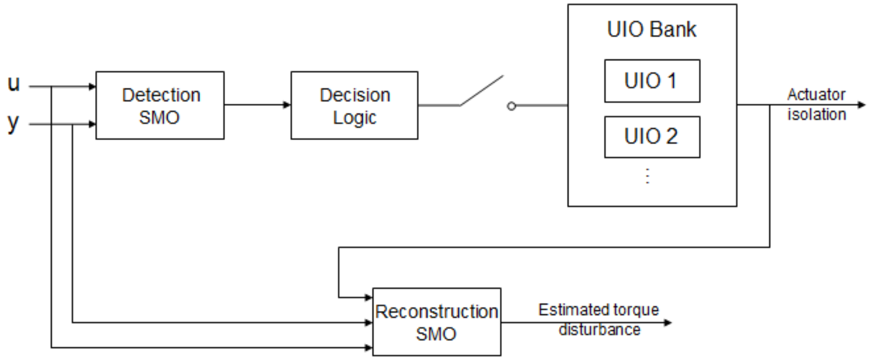


Fig. 2 Detailed FDI scheme

3.1 Detection Observer

In order to decide if a fault happened or not, a detection observer is designed. For computational complexity, the detection observer is the same as the reconstruction observer described further, but with a different tuning.

The dynamics of this observer are chosen to be slow, as a result, when a fault occurs, the error between the measured angular velocity and the output of the observer quickly increases.

3.2 A Cusum Test to Trigger the Isolation Process

The decision is based on a cusum test [2] on the angular velocity estimation error. The mean error of this state is supposed to be small when no fault occurs. If a fault happens, the mean value of the estimation error evolves. When a threshold is crossed on the cusum statistics, the test indicates that an actuator may be faulty. It is then time to isolate the fault.

3.3 Fault Isolation: A Bank of UIOs

For real-time applicability, the UIOs are triggered only when the decision criterion indicates that a fault has occurred. Even if only the angular velocity is estimated, keeping the UIOs switched off before the fault is detected seems to be a good strategy, regarding the number of actuators of the satellite studied here.

The isolation of the actuator is based on a bank of observers like in [6]. Unknown Input observers have been chosen here because of their decoupling capabilities. Once again, the angular velocity measurement is used by the observers, but this time for isolation.

For each thruster, an UIO is designed. The tuning is the same for all the actuators. Each UIO is such that it can fully reconstruct the angular velocity with all the inputs but one. As a result, the UIO dedicated to the faulty actuator will not be affected by the fault while all the others will be.

A UIO can be written in the form:

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{Fz} + \mathbf{TBu} + \mathbf{Ly} \\ \hat{\mathbf{x}} &= \mathbf{z} + \mathbf{Hy}\end{aligned} \tag{14}$$

For the state to be observable despite an unknown input, the design of the UIO has to respect some constraints.

In our case, the output equation of the state space representation simply is:

$$\mathbf{y} = \mathbf{Cx} = \mathbf{x}, \text{ i.e. } \mathbf{C} = \mathbf{I}_3 \tag{15}$$

Thus, for each UIO, the matrix **H** is such than $\mathbf{T} = \mathbf{I}_3\text{-}\mathbf{H}$ is orthogonal to the direction of the "missing" thruster, i.e.:

$$(\mathbf{I}_3\text{-}\mathbf{H})\mathbf{d}_{\mathrm{Ti}} = \mathbf{0} \tag{16}$$

Here, the matrix **F** is very simple:

$$\mathbf{F} = \text{-}\mathbf{K} \tag{17}$$

where **K** is a feedback matrix designed by pole placement. This matrix should be stable in order to ensure the convergence of the estimation error $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$.

Finally, the last condition to ensure the convergence is to design the matrix **L** as follows:

$$\mathbf{L} = \mathbf{K}(\mathbf{I}_3\text{-}\mathbf{H}) \tag{18}$$

Since we are not trying to reconstruct the fault here – we just want to isolate the faulty actuator – a classical observer design is proposed for the UIOs.

The UIO bank has to indicate clearly which actuator is faulty. Since the UIO dedicated to the faulty actuator is insensitive to the fault, the residuals of the other UIOs have to be very sensitive to input errors. The tuning of the UIO is such that the error estimation rapidly increases when the fault appears, while the residuals on the faulty UIO stay small. One drawback of this method is that even if an UIO can fully reconstruct the state, it can take time to have a small estimation error in the direction of the "missing" actuator.

To overcome this drawback, the residuals on the angular velocity are described in a specific frame for each UIO: the rotation matrix is such that the first axis of the new frame is the direction of the torque of the dedicated actuator. This means that the transient phase only happens on the first component while the others are quickly fully reconstructed. This is closely related to projections used in the context of parity space techniques for enhancing sensitivity to faults.

As a result, the residuals on the first component of each UIO will always converge to 0, with or without fault. On the other hand, the residuals on components 2 and 3 of all the UIO but the faulty one will increase when the fault appears.

It is now possible to study the norm of the residuals on components 2 and 3. The UIO that minimizes the norm of the residual on components 2 and 3 indicates which actuator is faulty.

3.4 Disambiguation Process

The UIOs use the direction of the torque due to the thrusters to isolate the faulty actuator. However, among the 12 thrusters of the satellite, two pairs of actuators have opposite directions making a disambiguation process necessary if the fault happens on one of these thrusters. We have:

$$\begin{aligned} \mathbf{d}_{\mathrm{T3}} &= -\mathbf{d}_{\mathrm{T1}} \\ \mathbf{d}_{\mathrm{T4}} &= -\mathbf{d}_{\mathrm{T2}} \end{aligned} \tag{19}$$

As a result, isolating the faulty thruster between actuators 1 and 3 or 2 and 4 requires the knowledge of the sign of the fault.

These four thrusters have the same force direction that is, described in the body frame:

$$\mathbf{d}_1 = \mathbf{d}_2 = \mathbf{d}_3 = \mathbf{d}_4 \tag{20}$$

The sign of the fault can be determined by comparing the commanded linear acceleration (i.e. the output of the control law on position and velocity) with the estimate of the linear acceleration computed from velocity measurement. Once the sign is known, the fault is easily isolated.

3.5 Reconstruction Observer

The design of this observer is based on the sliding mode observer proposed by [3] where the sliding motion is maintained even in the presence of faults. An estimate of the fault can be computed thanks to the *equivalent output injection* signal that maintains the sliding motion.

The reconstruction observer proposed here only deals with the angular velocity of the satellite. The state estimation follows the dynamics:

$$\dot{\hat{\boldsymbol{\omega}}} = \mathbf{I}^{-1}\mathbf{B}_{\mathrm{T}}\mathbf{u} - \mathbf{I}^{-1}\mathbf{y} \times \mathbf{I}\mathbf{y} + \mathbf{W}\mathbf{e}_{\mathrm{y}} + \mathbf{v} \tag{21}$$

where **W** is a stable design matrix and the vector **v** is defined as:

$$\mathbf{v} = -M \arctan \mathbf{e}_{\mathrm{y}} \text{ with } M > 0 \tag{22}$$

When an additive fault $\mathbf{\Delta u}$ occurs, the dynamics of the angular velocity become:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\mathbf{B}_{\mathbf{T}}(\mathbf{u} + \mathbf{\Delta u}) - \mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{23}$$

The state estimation error is defined as $\mathbf{e}_{\mathbf{y}} = \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}$, so its dynamics are:

$$\dot{\mathbf{e}}_{\mathbf{y}} = \mathbf{W}\mathbf{e}_{\mathbf{y}} + \mathbf{v} - \mathbf{I}^{-1}\mathbf{B}_{\mathbf{T}}\mathbf{\Delta u} - \mathbf{I}^{-1}\mathbf{y} \times \mathbf{I}\mathbf{y} + \mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{24}$$

It has been shown by [3] that during the sliding motion, $\mathbf{e}_{\mathbf{y}} = \mathbf{0}$ and $\dot{\mathbf{e}}_{\mathbf{y}} = \mathbf{0}$, so if no fault happens, $\mathbf{v} \to 0$ and if a fault occurs $\mathbf{v} \to \mathbf{I}^{-1}\mathbf{B}_{\mathbf{T}}\mathbf{\Delta u}$.

Thanks to the reconstruction observer, it is possible to compute an estimation of the torque disturbance due to the fault. If the fault has been correctly isolated, it is now possible to estimate it.

3.6 Reconfiguration of the Input Command

Once that the faulty actuator has been isolated and that the torque disturbance due to the fault has been reconstructed, it is possible to compute the value of the fault on the actuator thanks to the method of least squares.

Let $\mathbf{F}_{\mathbf{pert}}$ and $\mathbf{T}_{\mathbf{pert}}$ be the estimation of the disturbance due to the actuator fault on the force and the torque respectively, and let $\mathbf{F}_{\mathbf{co}}$ and $\mathbf{T}_{\mathbf{co}}$ the output of the control laws.

Once that a fault is detected, the input of the allocator becomes:

$$\begin{aligned} \mathbf{F}_{\text{alloc}} &= \mathbf{F}_{\text{co}} - \mathbf{F}_{\text{pert}} \\ \mathbf{T}_{\text{alloc}} &= \mathbf{T}_{\text{co}} - \mathbf{T}_{\text{pert}} \end{aligned} \tag{25}$$

The disturbance is directly pre-compensated in the control laws; therefore, even if the disturbance is not perfectly rejected because of the model errors, the "actual" commands $\mathbf{F}_{\mathbf{co}}$ and $\mathbf{T}_{\mathbf{co}}$ can be respected.

4 Simulations

4.1 An Example

The proposed method has been applied to the satellite model described in Section 2.1. The method is first illustrated on one example. For our scenario, the faulty actuator is the number 12. Figure 3 shows the residuals of the detection observer. The residuals clearly increase after the occurrence of the fault. The cusum test on the residuals of the detection observer triggers the bank of UIO.

The UIO dedicated to the 12th actuator is insensitive to faults on this actuator. Its residuals remain close to zero, despite the model errors that we considered here, while the residuals of the 11th UIO quickly increase (Figure 4). The UIO that minimizes the residuals on components 2 and 3 indicates which actuator is the faulty one. The faulty actuator is automatically detected and isolated (Figure 5). The amplitude of the fault is estimated (Figure 6) then pre-compensated by the reconfiguration of the reference input.

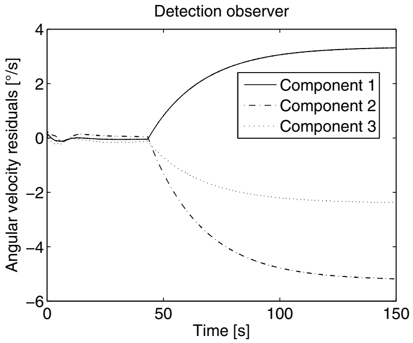


Fig. 3 Output of the detection observer

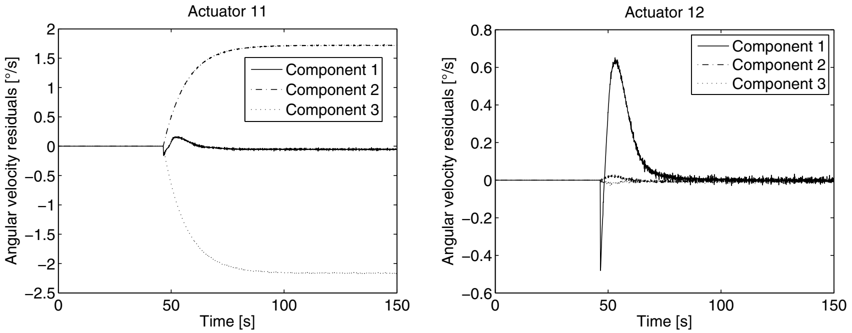
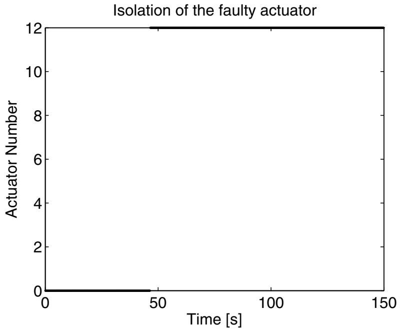


Fig. 4 Residuals of the 11th and the 12th UIOs



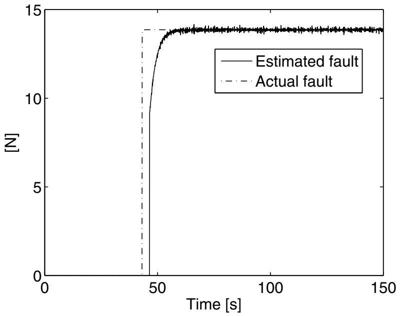


Fig. 5 Index of the faulty actuator

Fig. 6 Fault estimation

4.2 Influence of the Tuning of the SMOs

In order to highlight the interest of using different tunings for the detection and the reconstruction observers, the previous example is run again twice. The difference of tuning between the detection and the reconstruction observers lies in the scalar parameter M from equation 22. Let M_D be the parameter for the detection observer and M_R the parameter for the reconstruction observer.

In the tuning presented above, the mixed tuning, the parameter M_R is equal to ten times the parameter M_D since the dynamics of the detection are chosen to be slow while fast dynamics are required to estimate the fault quickly. The tuning where the detection and the reconstruction observers both have slow dynamics is referred as case 1 in the table below while the tuning where both have fast dynamics is referred as case 2.

Table 2 SMOs parameter M tuning

	M_D	M_R
Case 1: slow tuning	M_1	M_1
Case 2: fast tuning	10 M_1	10 M_1
Mixed tuning	M_1	10 M_1

Since the detection observer is the same for the case 1 and the mixed tuning, the detection happens at the same date. However, in case 1, the fault estimation converges much slower to the actual fault (Figure 7) because of the slow dynamics of the reconstruction observer. As a result, the input command can not be reconfigured efficiently.

For the case 2, the fault estimation is as fast as with the mixed tuning since the reconstruction observers are the same. Unfortunately, the detection time is longer than with the mixed tuning, by 2.3 s on this example. Indeed, once the fault has appeared, the residuals of the detection observer do not grow as fast as with the

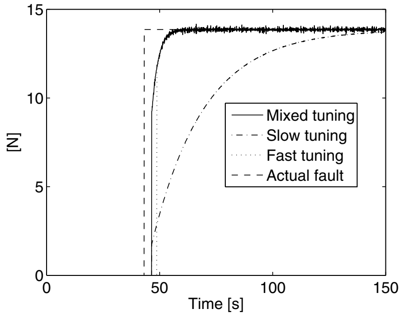


Fig. 7 Fault estimation for the three tunings

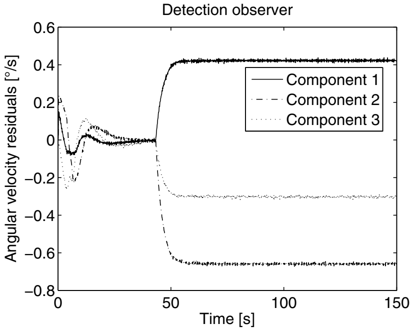


Fig. 8 Output of the detection observer for the fast tuning (case 2)

mixed tuning. As a result, it takes more time to cross the threshold of the cusum test making the isolation process triggered later. Furthermore, since the final values of the residuals are smaller, it becomes harder to detect small faults (Figure 8 compared to Figure 3) and the missed fault rate would be higher.

The tuning of the cusum test could be adapted to the fast tuning of the detection observer, but there is a risk that the residuals disturbed by the model errors would trigger the detection observer, especially in the first seconds of flight, resulting in an increased false alarm rate.

In the end, the use of two SMOs, one with slow dynamics dedicated to the detection, and one with fast dynamics dedicated to the fault estimation makes it possible to circumvent the usually required trade-off between the two tunings.

4.3 Performance Evaluation

In order to evaluate the efficiency of the method and its robustness to model errors, some performance indices should be defined. The fault isolation time indicates the performance of the method while the missed fault rate and the false alarm rate allow the evaluation of its robustness.

The fault isolation time is the difference between the moment the isolation is definitively done (the actuator indicator remains constant until the end of the scenario) and the moment the fault occurs.

The false alarm rate is the number of times a fault is detected while all the actuators are still healthy divided by the total number of scenarios.

The missed fault rate is the number of time a fault was not detected at the end of the scenario divided by the total number of scenarios.

To compute these indices, Monte-Carlo simulation tests have been carried out. For each simulation run, parameters of the model such as the position of the center of mass, the mass, the inertia, and the direction of the thrusters are altered. The value of the fault, the time of its appearance, and the index of the faulty actuator are randomly chosen.

All the model alterations follow a uniform distribution with bounds as described in Table 3.

Table 3 Summary table of the altered parameters of the satellite and their distribution

Parameter	Unit	Lower bound – Upper bound
Mass	[kg]	-10; +10
Inertia (by axis)	[kg.m²]	-10; +10
Center of mass (by axis)	[m]	-0.02; +0.02
Thruster direction (by actuator)	[°]	-0.2; +0.2

All the fault parameters follow a uniform distribution with bounds as described in Table 4. For each scenario, after the occurrence time, a bias appears on the faulty actuator until the end of the simulation.

Table 4 Summary table of the fault parameters

Parameter	Unit	Lower bound – Upper bound
Value	[N]	-30; +30
Actuator	-	1; 12
Occurrence time	[s]	0; 100

A number of 1000 MC simulation tests have been carried out. The false alarm rate and the missed fault rate can be directly computed. It appears that the appropriate tuning of the method allows avoiding false alarms since the false alarm rate is equal to zero. On the other hand, 49 faults were not detected among the 1000 scenarios. This means that we have a missed fault rate of 4.9%.

It might be possible to get a lower missed fault rate with different tuning of the cusum test but it could lead to a higher false alarm rate. It should however be noted that the values of the non-detected faults are small and have thus a very small effect on the system dynamics.

Figure 9 presents the detection time for the different fault values met in the 1000 scenarios. It clearly appears that the detection time depends on the value of the fault: the more the fault is important, the faster it is isolated. The smallest faults – less than 1.5N – are not detected.

Figure 10 shows the required delay for the fault isolation once it is detected. It appears that 50 % of the faults are isolated less than 0.2s after detection, and that 88 % are isolated within the second that follows the detection. In some cases - small fault, high model errors - the isolation time can be more important, however, the isolation is always achieved in the end.

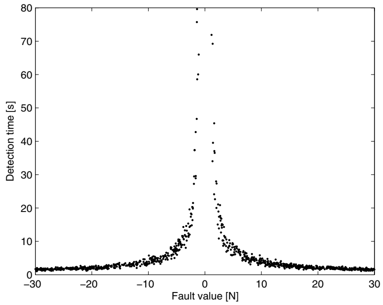


Fig. 9 Detection time versus fault

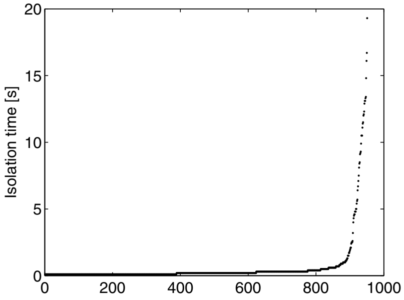


Fig. 10 Isolation time after detection

5 Conclusion

This paper has described a method to detect, isolate and compensate an actuator fault in a satellite system. The main challenge is to detect as soon as possible the fault and to reconstruct it rapidly among all the possible thrusters. The proposed method differs from the usual solutions by the use of two observers, one for detection, and one for reconstruction. The detection and the reconstruction observers use the same kind of sliding mode observer but with different tunings. The isolation of the faulty actuator is performed with a bank of UIO that is triggered by the detection observer. Each of the UIO residuals are described in a specific frame bound to the dedicated actuator in order to get more "readable" residuals for an easier isolation. Once a fault is isolated and reconstructed, the input command is reconfigured in order to pre-compensate the disturbances due to the fault.

The proposed method allows to successfully detect and isolate faults in most cases. Only the smallest faults are not detected but such faults lead to small disturbances, so the integrity of the system is not threatened. An optimal tuning thanks to a minimax optimization could allow to reduce the missed fault alarm without deteriorating the false alarm rate, taking into account the sources of uncertainty [5].

References

1. Alwi, H., Edwards, C., Marcos, A.: FDI for a Mars orbiting satellite based on a sliding mode observer scheme. In: Proc. of the IEEE Conference on Control and Fault-Tolerant Systems, Nice, France, pp. 143–148 (2010)
2. Basseville, M., Nikiforov, V.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Englewood Cliffs (1993)
3. Edwards, C., Spurgeon, S.K., Patton, R.J.: Sliding mode observers for fault detection and isolation. Automatica 36(4), 541–553 (2000)
4. Jian, T., Khorasani, K., Tafazoli, S.: Parameter estimation-based fault detection, isolation and recovery for nonlinear satellite models. IEEE Transactions on Control System Technology 16(4), 799–808 (2008)

5. Marzat, J., Walter, E., Damongeot, F., Piet-Lahanier, H.: Robust automatic tuning of diag-nosis methods via an efficient use of costly simulations. In: Proc. of the 16th IFAC Symposium on System Identification, Brussels, Belgium, pp. 398–403 (2012)
6. Patton, R.J., Uppal, F.J., Simani, S., Polle, B.: Robust FDI applied to thruster faults of a satellite system. Control Engineering Practice 18(9), 1093–1109 (2009)
7. Sidi, M.J.: Spacecraft dynamics and control: a practical engineering approach. Cambridge University Press (1997)
8. Talebi, H.A., Khorasani, K., Tafazoli, S.: A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem. IEEE Transactions on Neural Networks 20(1), 45–60 (2009)
9. Theilliol, D., Join, C., Zhang, Y.: Actuator Fault Tolerant Control Design Based on a Reconfigurable Reference Input. Journal of Applied Mathematics and computer Science 18(4), 553–560 (2008)
10. Varga, A.: Monitoring actuator failures for a large transport aircraft – the nominal case. In: Proc. of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain, pp. 627–632 (2009)
11. Wie, B.: Space Vehicle Dynamics and Control, Reston, Virginia. AIAA Educational Series (1997)
12. Williamson, W.R., Speyer, J.L., Dang, V.T., Sharp, J.: Fault Detection and Isolation for Deep Space Satellites. Journal of Guidance, Control and Dynamics 32(5), 1570–1584 (2009)
13. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control 32(2), 229–252 (2008)

A Spherical Coordinate Parametrization for an In-Orbit Bearings-Only Navigation Filter

Jonathan Grzymisch, Walter Fichter, Massimo Casasco, and Damiana Losa

Abstract. In-orbit rendezvous is a key enabling technology for many space missions. Implementing it employing only bearing measurements would simplify the relative navigation hardware currently required, increasing robustness and reliability by reducing complexity, launch mass and cost. The problem of bearings-only navigation has been studied intensively by the Naval and Military communities. Several authors have proposed that a polar or spherical coordinate parametrization of the underlying dynamics produces a more robust navigation filter due to the inherent de-coupling of the observable and un-observable states. Nevertheless, the complexity of this problem increases significantly when the underlying dynamics follow those of relative orbital motion. This paper develops a spherical coordinate parametrization of the linearized relative orbital motion equations for elliptical orbits and uses an approximation of these equations for circular orbits to develop an Extended Kalman Filter (EKF) for bearings-only navigation. The resulting filter is compared to its equivalent based on the well known Hill Equations in cartesian coordinates via a Monte Carlo analysis for a given reference trajectory. Simulations show that a spherical coordinate based EKF can perform better than its cartesian coordinate counterpart in terms of long-term stability tracking of the reference trajectory, with little additional computational effort.

Jonathan Grzymisch · Walter Fichter
Institute of Flight Mechanics and Control,
University of Stuttgart, Germany
e-mail: {jonathan.grzymisch,walter.fichter}@ifr.uni-stuttgart.de

Massimo Casasco
European Space Technology Centre, European Space Agency, The Netherlands
e-mail: massimo.casasco@esa.int

Damiana Losa
Research and Technology Department, Thales Alenia Space, France
e-mail: damiana.losa@thalesaleniaspace.com

1 Introduction

In-orbit rendezvous is a key enabling technology for many space missions. Without a technology allowing a chaser spacecraft to reach a target with high accuracy and low collision risk, we could never achieve missions such as in-orbit assembly of large structures (ISS); planetary exploration and return (Apollo and Mars Sample Return); in-orbit servicing, refuelling and inspection (ATV, DEOS) and Active de-orbiting (OTV).

While this topic has been widely researched and there exists significant heritage, it currently requires complex and/or bulky hardware to measure the relative range between the chaser spacecraft and the target [1]. This is especially the case with un-cooperative targets at long distances, as the power requirements and complexity of the range sensors increases exponentially without dedicated hardware on the target [2]. There have been may efforts to simplify the required hardware for rendezvous [1]. Nevertheless, most solutions only work at short ranges of less than a few kilometres, such as those involving visual cameras using stereo vision (triangulation along a well-known baseline) or estimation of the range from the relative size of the target.

For these reasons there is a strong motivation to develop algorithms to perform in-orbit rendezvous without requiring a direct measurement of the range between the chaser and the target. Bearing measurements to an un-cooperative target are easier to obtain, especially at long range, without the need of heavy or complex hardware, for example by using a single optical camera to measure Line of Sight to the target. Therefore, this would not only enable rendezvous missions with low launch mass and cost, but would also provide a back-up strategy for contingency cases in missions employing more advance sensors.

The problem of bearings-only navigation has been studied intensively by the Naval and Military communities with applications to ship navigation and missile guidance assuming a constantly moving and non-maneuvering target [3–5]. In particular, there have been several studies claiming that using a polar or spherical coordinate parametrization of the equations of motion to construct an EKF for bearings-only navigation naturally decouples the un-observable states (range) from the observable ones (angles). This prevents covariance matrix ill-conditioning and filter instability, resulting in a more robust and unbiased filter [6, 7].

Nonetheless, the assumptions employed in the Naval and Military literature do not apply to the in-orbit problem due to the complexity of orbital dynamics. Only a few authors have treated the in-orbit bearings-only navigation problem [1] and to the authors' knowledge, none have attempted to test the potential advantages of a spherical coordinate parametrization for a bearings-only navigation filter for in-orbit rendezvous. This paper tackles this specific problem and for this purpose develops a spherical coordinate parametrization for the equations of linearized relative orbital motion for eccentric and circular orbits. The latter are equivalent to the well known Hill Equations.

2 Spherical Relative Motion Equations

In order to derive the equations of relative motion in Spherical coordinates, the definition of the relative position vector in spherical coordinates (see Appendix) is used in conjunction with the vectorial equation of relative motion (16) derived in the Appendix. Solving for the desired accelerations in Spherical coordinates $\ddot{r}$, $\ddot{\theta}$ and $\ddot{\phi}$, the final expressions for the Spherical Equations of Relative Motion emerge and are shown in Equations 1 to 3. These equations are valid for elliptical orbits.

$$\begin{aligned}\ddot{r} = {} & a_r + \dot{\phi}^2 r + r\omega^2 + 2kr\omega^{\frac{3}{2}} + r\dot{\theta}^2 c_\phi^2 - r\omega^2 c_\phi^2 + r\omega^2 c_\phi^2 c_\theta^2 - 3kr\omega^{\frac{3}{2}} c_\phi^2 \dots \\ & + 2\dot{\phi} r\omega c_\theta + 2r\dot{\theta}\omega s_\theta c_\phi s_\phi\end{aligned} \tag{1}$$

$$\ddot{\theta} = \frac{a_\theta - 2\dot{r}\omega s_\phi s_\theta}{r c_\phi} + \frac{2\dot{\phi}\dot{\theta} s_\phi - \dot{\omega} s_\phi s_\theta}{c_\phi} - \frac{2\dot{r}\dot{\theta}}{r} - c_\theta s_\theta \omega^2 - 2\dot{\phi} s_\theta \omega \tag{2}$$

$$\begin{aligned}\ddot{\phi} = {} & -\dot{\omega} c_\theta - \dot{\theta}^2 s_\phi c_\phi + \omega^2 s_\phi c_\phi s_\theta^2 + \frac{a_\phi - 2\dot{\phi}\dot{r} - 2\dot{r}\omega c_\theta}{r} + 3k\omega^{\frac{3}{2}} s_\phi c_\phi \dots \\ & + 2\dot{\theta}\omega s_\theta c_\phi^2\end{aligned} \tag{3}$$

Here $c_i = \cos(i)$ and $s_i = \sin(i)$ for $i = \phi$ or θ are used to simplify the notation. Substituting the assumptions of constant orbital rate $\omega = \sqrt{\mu/r_t^3}$ into Equations 1 to 3, yields the Equations for Relative Motion in Spherical Coordinates for Circular Orbits, shown in Equations 4 to 6. These are equivalent to the Hill Equations cited in many references including [2], also shown in the Appendix.

$$\ddot{r} = a_r + r\dot{\phi}^2 + 2r\dot{\phi}\omega c_\theta + r\dot{\theta}^2 c_\phi^2 + 2r s_\phi s_\theta \dot{\theta}\omega c_\phi + r\omega^2 \left(c_\phi^2 c_\theta^2 - 4c_\phi^2 + 3\right) \tag{4}$$

$$\ddot{\theta} = \frac{a_\theta - 2\dot{r}\omega s_\phi s_\theta}{r c_\phi} + \frac{2\dot{\phi}\dot{\theta} s_\phi}{c_\phi} - \frac{2\dot{r}\dot{\theta}}{r} - c_\theta s_\theta \omega^2 - 2\dot{\phi} s_\theta \omega \tag{5}$$

$$\ddot{\phi} = \frac{a_\phi - 2\dot{\phi}\dot{r} - 2\dot{r}\omega c_\theta}{r} + \omega^2 s_\phi c_\phi \left(s_\theta^2 + 3\right) - \dot{\theta}^2 s_\phi c_\phi + 2\dot{\theta}\omega s_\theta c_\phi^2 \tag{6}$$

3 Validation of the Spherical Equations

In order to validate the equations of relative motion for circular orbits in Spherical coordinates, a comparison of equations 4 to 6 with the Hill Equations (see Appendix) was performed via numerical simulations. Various test orbits were propagated from a known initial condition using both sets of equations. The the resulting trajectories were then compared to verify the equivalence of the equations. The propagation were carried out using the MATLAB® *ODE*45 solver. In addition, for some test cases the trajectories were validated using the linearized solution to the Hill Equations, the Clohessy-Wiltshire (CW) equations [2]. Table 1 lists the trajectories tested as well as the resulting position and velocity errors between the Spherical and Cartesian trajectories, over the whole simulation period. The errors for all the trajectories tested were within numerical integration errors in the order of 1×10^{-6} meters. An example of the validation trajectories, corresponding to the last row of Table 1, is shown in Figure 1 and the corresponding errors in Figure 2.

Table 1 Spherical relative motion equations validation results

Test Description (Compared with)	IC [m & m/s] ($\mathbf{x}^c = [x\, y\, z\, \dot{x}\, \dot{y}\, \dot{z}]^T$)	$\lvert\lvert\mathbf{x}^c(t) - \mathbf{x}^s(t)\rvert\rvert$ Pos [m]	 Vel [m/s]
Arbitrary IC (CW)	[100 10 150 0.01 0.1 0.2]	8.00e-08	4.77e-11
Arbitrary IC (Hill)	[100 10 150 0.01 0.1 0.2]	7.23e-08	3.86e-11
Different Altitude (CW)	[0 0 150 0.2617994 0 0]	2.31e-08	1.26e-11
Release at Z-Direction (CW)	[0 0 150 0 0 0]	7.32e-08	8.04e-11
Release at Y-Direction (CW)	[0 10 0 0 0 0]	3.31e-10	1.78e-09
Initial Velocity on X-Direction (CW)	[ε^* 0 0 0.01 0 0]	7.01e-09	6.07e-12
Initial Velocity Out of Plane (CW)	[0 ε^* 0 0 0.1 0]	9.51e-10	3.57e-10
Forced Motion with Accelerations** (Hill)	[1000 10 15 0.01 0.01 0.02]	3.84e-05	3.37e-08

*Note: ε = 2.22e-16 is used to avoid the numerical singularity at $\mathbf{r} = [0\,0\,0]^T$ m

**Constant acceleration used throughout the simulation: $\mathbf{a} = [1\,2\,3]^T$ m/s^2

4 In-Orbit Bearings-Only Navigation Filters

Two discrete Extended Kalman filters were formulated according to [8], using the Cartesian and Spherical equations of motion. The propagation step within the filters was implemented using a fourth order Runge-Kutta (RK4) integration of the state equations, shown below.

The state vectors for the navigation filters in Cartesian $\mathbf{x}^c$ and Spherical $\mathbf{x}^s$ coordinates are defined as in Equations 23 and 24 respectively. The state equations for each filter, $\mathbf{f}^c(\mathbf{x}^c)$ and $\mathbf{f}^s(\mathbf{x}^s)$, were formulated from the relative motion equations for circular orbits in Cartesian (18-20) and Spherical (4-6) coordinates respectively. These are shown in Equations 7 and 8 below.

$$\dot{\mathbf{x}}^c = \mathbf{f}^c(\mathbf{x}^c) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 2\dot{z}\omega + a_x \\ -\omega^2 y + a_y \\ 3\omega^2 z - 2\dot{x}\omega + a_z \end{bmatrix} \tag{7}$$

$$\dot{\mathbf{x}}^s = \mathbf{f}^s(\mathbf{x}^s) = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{\phi} \\ a_r + r\dot{\phi}^2 + 2r\dot{\phi}\omega c_\theta + r\dot{\theta}^2 c_\phi^2 + 2r s_\phi s_\theta \dot{\theta}\omega c_\phi + r\omega^2\left(c_\phi^2 c_\theta^2 - 4c_\phi^2 + 3\right) \\ \frac{a_\theta - 2\dot{r}\omega s_\phi s_\theta}{r c_\phi} + \frac{2\dot{\phi}\dot{\theta}s_\phi}{c_\phi} - \frac{2\dot{r}\dot{\theta}}{r} - c_\theta s_\theta \omega^2 - 2\dot{\phi}s_\theta\omega \\ \frac{a_\phi - 2\dot{\phi}\dot{r} - 2\dot{r}\omega c_\theta}{r} + \omega^2 s_\phi c_\phi\left(s_\theta^2 + 3\right) - \dot{\theta}^2 s_\phi c_\phi + 2\dot{\theta}\omega s_\theta c_\phi^2 \end{bmatrix} \tag{8}$$

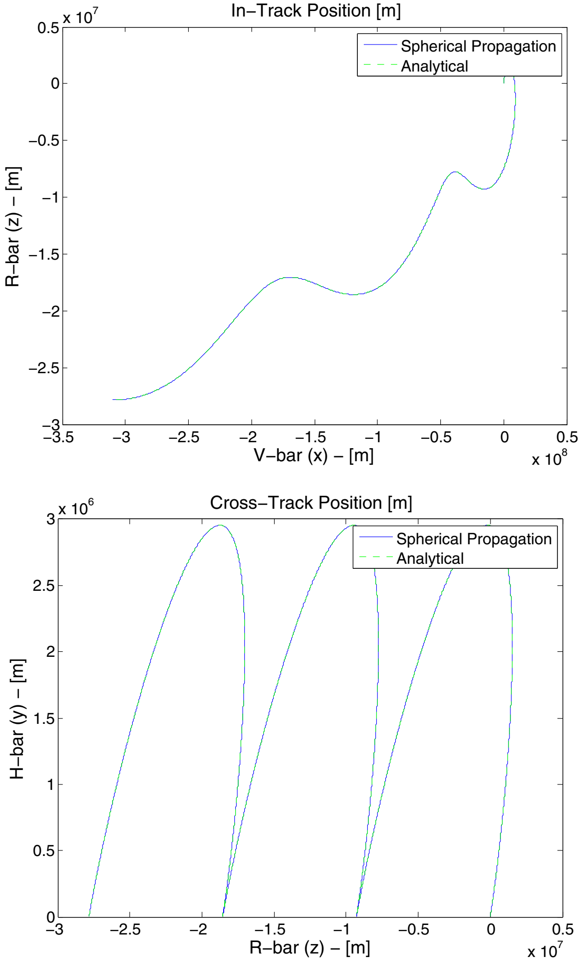


Fig. 1 Validation Example Trajectory for the Spherical Relative Motion Equations

Finally, the angles-only measurement equations for each filter were developed from the geometric relations in Figure 10. These are shown in Equation 9 and 10 for the Cartesian and Spherical filters respectively.

$$\mathbf{h}^c(\mathbf{x}^c) = \begin{bmatrix} \arctan(y/x) \\ \arcsin\left(z/\sqrt{x^2+y^2+z^2}\right) \end{bmatrix} \tag{9}$$

$$\mathbf{h}^s(\mathbf{x}^s) = \begin{bmatrix} \theta \\ \phi \end{bmatrix} \tag{10}$$

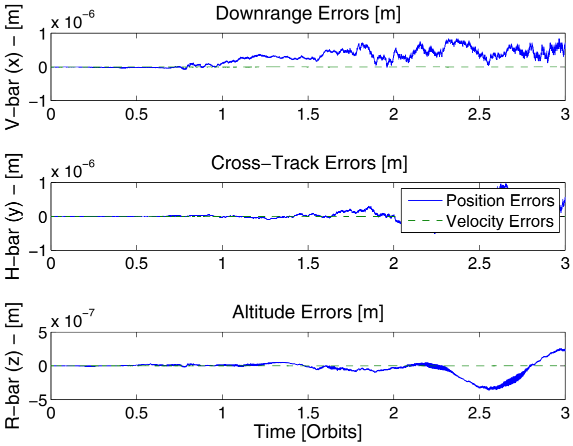


Fig. 2 Validation Errors for the Example Trajectory

4.1 Filter Stability Measure

Even though the stability of Extended Kalman Filters can not be guaranteed, EKFs are attractive since their performance is near-optimal when the estimation errors are small and the non-linearities are tightly cone bounded [9]. Several references including [9] suggest that the performance of an EKF depends heavily on the coordinate system used to formulate the filter. In order to aid in the selection of a suitable coordinate system, Weiss and Moore [9] provide a "stability measure" test based on a bound on the decay rate of a Lyapunov function. According to this test, the larger the value of the "stability measure" μ_s in Equation 11, the more stable the system will be over the range of state estimates. Here, **R** is the noise covariance matrix for the measurements.

$$\mu_s = \mathbf{h}^T \mathbf{R}^{-1} \mathbf{h} - [\mathbf{H}_k \mathbf{x} - \mathbf{h}]^T \mathbf{R}^{-1} [\mathbf{H}_k \mathbf{x} - \mathbf{h}] \tag{11}$$

$$\text{where} \quad \mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}^T} \right|_{\mathbf{x}=\mathbf{x}_k}$$

Applying this stability measure to both the cartesian and spherical systems by substituting the corresponding measurement equations from 9 and 10 into Equation 11, we obtain the following measures for each system, shown in equations 12 and 13.

$$\mu_s^c = 0 \tag{12}$$

$$\mu_s^s = (\mathbf{h}^s)^T \mathbf{R}^{-1} \mathbf{h}^s = \frac{\phi^2}{R_\phi} + \frac{\theta^2}{R_\theta} \tag{13}$$

Since $\mu_s^s > 0$ for any real value of $\mathbf{h}^s$, the stability measure test proposes that the spherical coordinate formulation is a better candidate for an EKF than its cartesian counterpart. Note that this is not in any way a guarantee of stability of the EKF. It is only an indicator that in the linear case, or when the filter is operating near the linearization set-point, the rate of decay of ”energy” in the system for the spherical filter formulation would be positive and larger than the cartesian one. Therefore, this supports the results of the simulations that are presented in the next Section.

5 Navigation Filter Simulations

A Simulink® simulation was prepared where both the Cartesian and the Spherical filters were used to estimate a trajectory propagated using the Hill equations. The simulation model is shown in Figure 3.

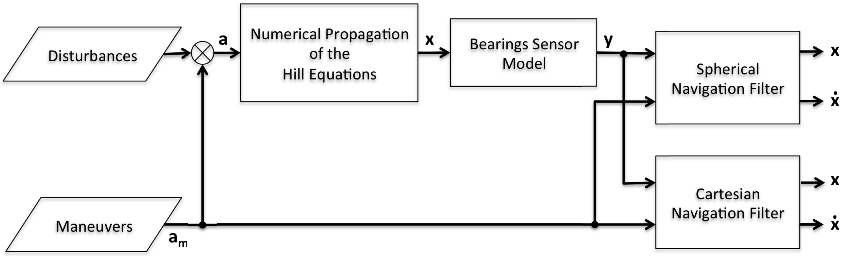

Disturbances
a
Numerical Propagation of the Hill Equations
x
Bearings Sensor Model
y
Spherical Navigation Filter
x
ẋ
Maneuvers
a_m
Cartesian Navigation Filter
x
ẋ

Fig. 3 Simulation Block Diagram for Navigation Filters Comparison

The parameters and tunings used to initialize the simulations of the navigation filters are summarized in Table 2. It is important to note that the same tunings and initialization parameters were used for both filters by converting the cartesian quantities into spherical ones, as explained in the Appendix.

5.1 Observability of the Bearings-Only Problem

The problem of estimating position and velocity from only angle measurements is known to have reduced observability depending on the relative motion between the satellites [10]. In addition, the range along the LOS direction is known to be not observable in the bearings-only problem, unless a maneuver is executed in a suitable direction [10].

In order to illustrate this point as well as to validate the functionality of the filters, a ‘noiseless’ simulation was performed. All sensor noises and un-modelled disturbances in Table 2 were set to zero. The reference trajectory was generated by propagating the following initial condition for the position and velocity $\mathbf{x_0} = [x y z \dot{x} \dot{y} \dot{z}]^T = [10000\ 5\ 1\ 0.2\ 0.1\ -2]^T$. As it can be seen on Figure 4 and 5, both filters can track the reference trajectory fairly well.

Table 2 Filter Comparison Simulation Parameters

Parameter	**Description**	**Simulator Value**	**Units**
Environment			
Orbital Period (T)	90 min	54000	[s]
Fly-around IC ($\mathbf{x}_0^c$)	$[x y z \dot{x} \dot{y} \dot{z}]$	[10000 5 1 0 0 -2]	[m & m/s]
Unmodelled Acceleration	100m/rev 3σ	([100/T^2 100/T^2 100/T^2]/3)^2	[m/s]
Time Step (T_s)	10 sec	10	[s]
Sensor			
Measurement Noise	1 mrad/axis 3σ	([1e-3 1e-3]/3)^2	[rad]
Measurement Bias	None	[0 0]	[rad]
Filters			
Spread of Initial Errors	StDev 1σ	[300 30 300 0.03 0.3 0.3]/3	[m & m/s]
Initial Covariance	300m, 0.3 m/s 3σ	diag(([300 300 300 0.3 0.3 0.3]/3)^2)	[m & m/s]
Sensor Noise Covariance	1 mrad/axis 3σ	diag(([1e^{-3} 1e^{-3}]/3)^2)	[rad]
Plant Noise Covariance	100 m/rev 3σ	diag(([0 0 0 100 100 100]/3T)^2)/T_s	[m & m/s]
Orbital Rate (ω)	$2\pi/T$	0.0012	[rad/s]
Time Step (T_s)	10 sec	10	[s]

The filter covariance is also shown in Figure 4 for some points along the trajectory. These ellipses, which are the filter's estimate of its own errors, correspond to the 3σ values from the error covariance matrices, scaled by a factor of 3 for easy visualization. The covariances are very similar for both filters since they employ the same tunning of their Plant and Sensor noise covariance matrices. Finally, it can be seen that the uncertainty is always greater in the LOS direction to the target at [0,0], as there is no observability in this direction when no maneuvers are performed.

5.2 *Filter Comparison Monte Carlo*

One hundred Monte Carlo Simulations were run in order to test the performance of the filters over a range of different initialization errors employing noisy measurements and subject to disturbances in the reference trajectory. In order to provide a high dynamic relative motion that aids filter convergence, the relative initial conditions were chosen to yield a 'football' or 'fly-around' periodic trajectory. Table 2 in section 5 summarizes all the parameters used to initialize the Monte Carlo.

In addition, a maneuver was performed mid way through the simulation in order to show how the filters gain observability in the range direction. This impulsive maneuver was executed after two orbits via an acceleration pulse $\mathbf{a}_{man} = [0\,0.005\ -0.005]^T$ lasting 10 seconds, resulting in a total delta-V of 0.1 m/s in the y and $-z$ directions. Figure 6 shows the resulting trajectory estimates by both filters, along with the reference trajectories for each of the Monte Carlo simulations. Note the trajectory change due to the maneuver after two orbits.

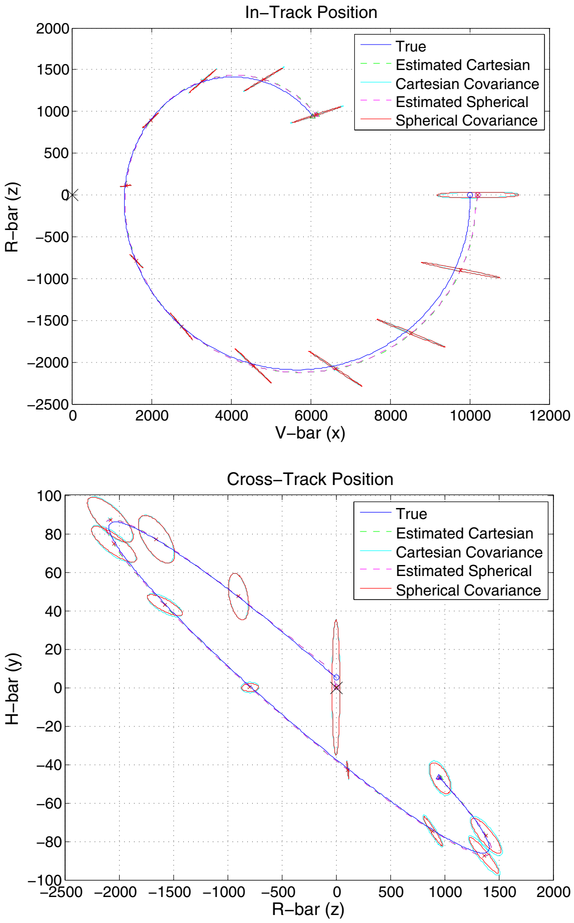


Fig. 4 Filter Comparison Trajectory

Statistical dispersions were computed from the trajectory errors of each simulation in order to evaluate the 1σ performance of each filter. These are shown in Figure 7 along with their percentage of the range to the target. This last measure is very useful in spacecraft rendezvous since, as a rule of thumb, a relative position estimate of around 1% of the range is required to achieve impulsive rendezvous [2].

Finally, the run-time of each filter was analyzed during the Monte Carlo simulations. On a 1.8 GHz computer, the cartesian filter demanded on average 8.3×10^{-4} seconds per call (propagation + update), while the spherical one demanded 9.1×10^{-4} seconds per call. That is only about a 10% increase in average CPU run time for the spherical filter.

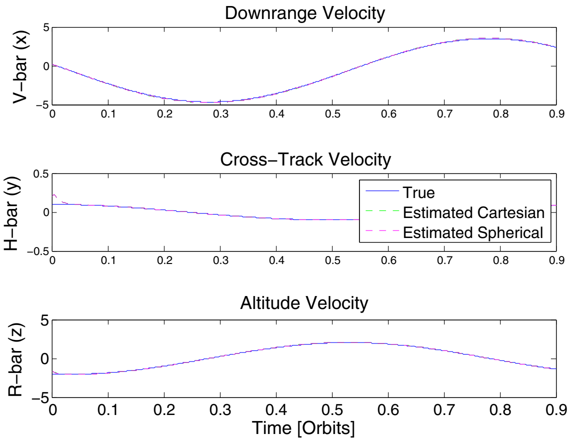


Fig. 5 Filter Comparison Velocities

5.3 Filter Comparison Results

The Monte Carlo analysis shows that for the reference trajectory chosen, the spherical filter statistically outperforms the cartesian one. This can be seen in the resulting navigation dispersions shown in Figure 7. Note that both solutions slowly diverge due to the inherent reduced observability of the bearings-only navigation problem. However, when a maneuver is performed in a suitable direction [10], it provides the necessary observability to reduce the estimated error in the LOS direction. Both filters take advantage of this and reduce their total estimation uncertainties when the maneuver is performed.

In any case, the spherical filter tracks the reference trajectory with superiority when no maneuvers are performed, diverging at a much slower rate. Conceptually, this is due to the fact that in contrast to the cartesian filter, the spherical filter separates the observable (angles) and un-observable (range) states. Essentially, it only needs to estimate the range and range-rate, as the other four states are directly the measurements and their derivatives. On the other hand, the cartesian filter needs to estimate all six states (position and velocities) from measurements that are non-linearly related to its states, resulting in lower performance. Mathematically, this is readily explained by noting where the filters employ key linearizations of the underlying equations. The Extended Kalman filter relies on a linearization of the measurement equation in order to calculate the Kalman Gain that is used to apply the measurement update. In contrast to the spherical filter where this equation is already linear, the measurement equation in the cartesian filter is highly non-linear (Refer to Equations 9 and 10). Thus, the linearization required in the cartesian filter results in a slightly less accurate measurement update. In addition, the linearized measurement equation is also used in the update of the filter covariance matrix, introducing further inaccuracies. Therefore, even though both filters rely on a linearized state

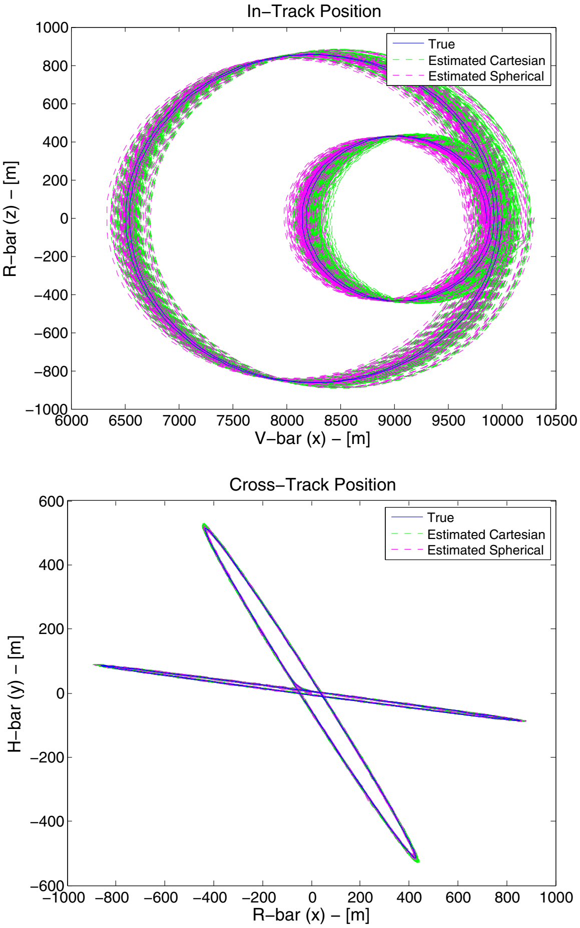


Fig. 6 Monte Carlo Trajectories

transition matrix to propagate their covariance matrix, the cartesian filter also relies on a linearized measurement equation. This introduces additional inaccuracies compared to the the spherical filter, where the measurement equation is already linear. Hence, decoupling the observable and un-observable states results in a simple measurement equation which reduces the linearization inaccuracies in the filter.

Several other simulations on top of the Monte-Carlo analysis were performed during the characterization of the spherical filter implementation, which are not shown here due to space constraints. Nevertheless, the general observation was that the Spherical filter implementation was found to be more robust than the Cartesian one in terms of changes to its tunning parameters as well as changes to the measurement update frequency.

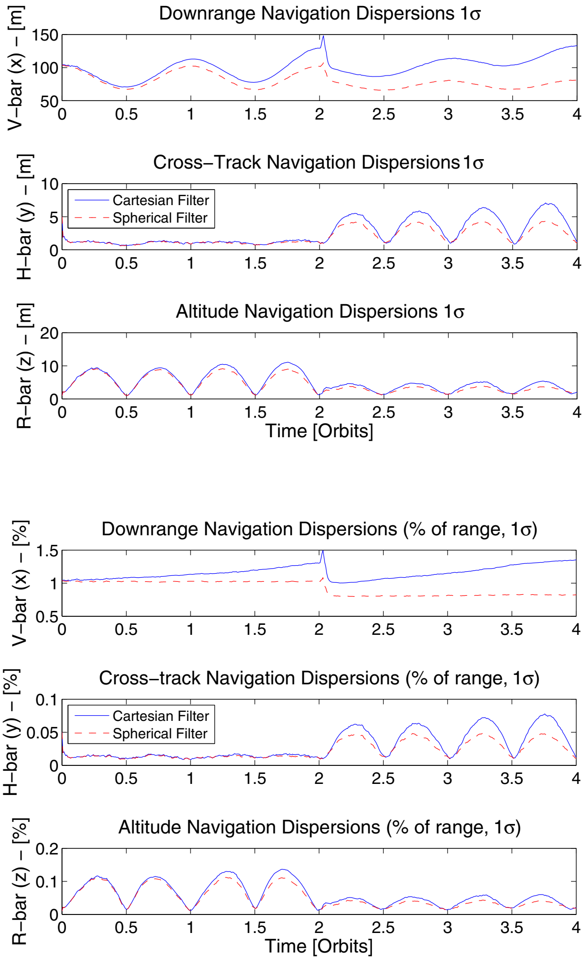


Fig. 7 Monte Carlo Trajectory Dispersions

Finally, as mentioned in section 5.2, the increase in performance from the spherical filter only comes with a small increase of about a 10% in CPU run-time.

6 Conclusions

In this paper, the linearized equations of relative orbital motion were derived in spherical coordinates and a new in-orbit bearings-only navigation filter was implemented using these equations.

This work shows that a spherical coordinate based filter can perform better for the bearings-only in-orbit navigation problem than a traditional Cartesian implementation. A more in depth characterization of the robustness of the spherical filter implementation with respect to different measurement errors, update frequencies and and filter tunings is required. Nevertheless, these results show the potential advantages in performance and robustness that can result from the use of a coordinate system parameterization that acquires the measurements as its own states. This results in simple measurement equations, essentially shifting the non-linearities inside the EKF from the measurement update, where linearizations are heavily relied upon, to the propagation, where the full state equations can be partly employed. This was shown in the construction of the spherical filter.

Even though this filter implementation implies more development effort due to the complex and longer equations required to model the relative motion dynamics, there is little additional on-line computational effort required to perform the actual trajectory estimation. This makes the spherical filter a very interesting robust alternative for an on-board implementation.

Acknowledgements. The authors would like to thank the European Space Agency (TEC-ECN Section at ESTEC) and Thales Alenia Space France (Research and Technology/Science and Observation within the R&D department) for providing the funding that made this research possible, through the ESA Networking/Partnering Initiative (NPI) program. In addition, we would like to thank André Posh from the iFR at the University of Stuttgart for the discussions on Extended Kalman Filters and their performance.

Appendix

In-Orbit Relative Motion Background

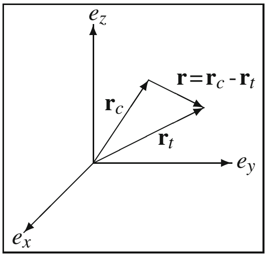
A generic derivation of the in-orbit relative motion equations is presented here. A more detailed derivation, but focused on the Cartesian coordinate parametrization of the equations, can be found in [2] and [11].

Relative Motion in the Inertial Frame

Consider the general scenario of two point mass spacecraft subject to the effects of a central Spherical gravity field and other external accelerations. Their geometry is defined in Figure 8, where the spacecraft are denominated as a Target and a Chaser with position vectors $\mathbf{r}_t$ and $\mathbf{r}_c$ respectively. In inertial space, the relative acceleration is directly the second time derivative of the relative position vector $\ddot{\mathbf{r}}$.

The motion of each of these spacecraft can be described by Newton's Second Law $\mathbf{F}_i = m\ddot{\mathbf{r}}_i$, where $\mathbf{F}_i$ must include all external forces for each vehicle. Considering only the influence of a central gravity force given by Newton's law of Gravitation as well as control thrust accelerations from the chaser vehicle $\mathbf{a}_{ext} = \mathbf{F}_{ext}/m_c$, the linearized differential equation for relative motion in the inertial frame is:

Fig. 8 Definition of the chaser, target and relative position vectors in the inertial frame



$$\ddot{\mathbf{r}} = \left.\frac{d\mathbf{f}_{\mathbf{g}}(\mathbf{r}_i)}{d\mathbf{r}_i}\right|_{\mathbf{r}_i=\mathbf{r}_t} \mathbf{r} + \mathbf{a}_{ext} \tag{14}$$

where the jacobian of $\mathbf{f}_{\mathbf{g}}(\mathbf{r}_i)$ comes from the linearization by Taylor expansion of the gravitational force of the chaser around the target location, which expressed with respect to a generic vector $\mathbf{r}_i = [x_i\ y_i\ z_i]^T$ is as follows:

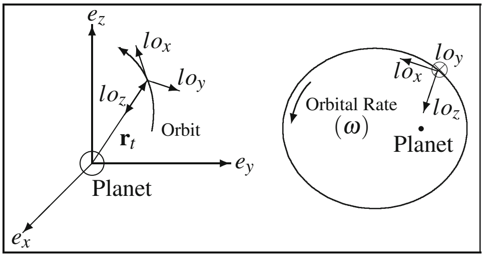
$$\frac{d\mathbf{f}_{\mathbf{g}}(\mathbf{r}_i)}{d\mathbf{r}_i} = \frac{\mu}{r_i^3}\begin{bmatrix} \frac{3x_i^2}{r_i^2} - 1 & \frac{3x_iy_i}{r_i^2} & \frac{3x_iz_i}{r_i^2} \\ \frac{3x_iy_i}{r_i^2} & \frac{3y_i^2}{r_i^2} - 1 & \frac{3y_iz_i}{r_i^2} \\ \frac{3x_iz_i}{r_i^2} & \frac{3y_iz_i}{r_i^2} & \frac{3z_i^2}{r_i^2} - 1 \end{bmatrix} \tag{15}$$

where $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$.

Relative Motion in the Local Orbital Frame

The Local Orbital Frame ($\mathfrak{F}^{\mathbf{lo}}$), fixed to the orbital motion of the target spacecraft, is centered at the target position $\mathbf{r}_t$ and rotates with respect to the inertial frame at a rate $\boldsymbol{\omega}$ equal to the instantaneous orbital rate of the target, as depicted in Figure 9. Here, the z-axis always points towards the center of the orbit; the y-axis is in the opposite direction of the angular momentum and the x-axis completes the triad.

Fig. 9 Definition of the local orbital frame $(\mathfrak{F}^{\mathbf{lo}})$



In order to obtain the equations of relative motion in the $\mathfrak{F}^{\mathbf{lo}}$ frame, Equation 14 can be expressed in this frame by using the second derivative of the relative position vector $\mathbf{r}$ in the rotating frame:

$$\ddot{\mathbf{r}}^{lo} + 2\boldsymbol{\omega} \times \dot{\mathbf{r}}^{lo} + \dot{\boldsymbol{\omega}} \times \mathbf{r}^{lo} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}^{lo} - \left.\frac{d\mathbf{f_g}(\mathbf{r}_i)}{d\mathbf{r}_i}\right|_{\mathbf{r}_i = \mathbf{r}_t^{lo}} \mathbf{r}^{lo} = \mathbf{a}_{ext}^{lo} \qquad (16)$$

In addition, from the definition of the $\mathfrak{F}^{\mathbf{lo}}$ frame, the following assumptions apply to Equation 16:

$$\mathbf{r_t^{lo}} = \begin{bmatrix} 0 \\ 0 \\ -r_t \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} 0 \\ -\omega \\ 0 \end{bmatrix}, \quad h = {r_t}^2\omega \quad \text{and} \quad k = \frac{\mu}{h^{\frac{3}{2}}} \qquad (17)$$

where the constant h is the orbital momentum for the planar orbital motion of the target spacecraft and the constant k is defined as done in [11] in order to remove the orbital radius r_t from the equations.

The Hill Equations

The Hill equations, shown below, can be obtained by substituting into Equation 16 the assumptions related to a cartesian position vector $\mathbf{r} = [x\ y\ z]^T$ along with the assumptions for circular orbits described in Section 2.

$$\ddot{x} = a_x + 2\,\omega\,\dot{z} \qquad (18)$$

$$\ddot{y} = a_y - \omega^2\,y \qquad (19)$$

$$\ddot{z} = 3\,z\,\omega^2 - 2\,\dot{x}\,\omega + a_z \qquad (20)$$

Definition of the Spherical Coordinates

In the $\mathfrak{F}^{\mathbf{lo}}$ frame, we can define the relative position vector in terms of the Spherical coordinates r, θ and ϕ as shown in Figure 10.

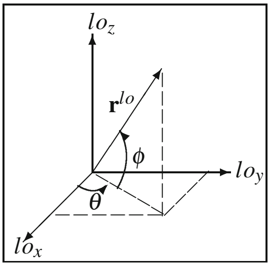


Fig. 10 Definition of the Relative Position Vector in terms of the Spherical Coordinates in the $\mathfrak{F}^{\mathbf{lo}}$ frame

The relative position vector in terms of these Spherical coordinates is:

$$\mathbf{r}^{lo} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} rc_\phi c_\theta \\ rc_\phi s_\theta \\ rs_\phi \end{bmatrix} \tag{21}$$

where $c_i = \cos(i)$ and $s_i = \sin(i)$ for $i = \phi$ or θ is used to simplify the notation. It is also useful to express the external accelerations $\mathbf{a}^{lo}_{ext}$ in terms of the spherical variables $\mathbf{a}^{lo}_{sph} = \left[a_r \; a_\theta \; a_\phi\right]^T$:

$$\mathbf{a}^{lo}_{ext} = \mathbf{C_{sc}}^T \mathbf{a}^{lo}_{sph} \quad \text{where} \quad \mathbf{C_{sc}} = \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta & s_\phi \\ -s_\theta & c_\theta & 0 \\ -c_\theta s_\phi & -s_\phi s_\theta & c_\phi \end{bmatrix} \tag{22}$$

Coordinate Transformations

In order to compare the Spherical coordinate results with the Cartesian ones, the following coordinate transformations were defined. These were used to compute the equivalent initial conditions in Spherical coordinates as well as to translate the resulting trajectories to Cartesian coordinates. Each position-velocity vector expressed in Cartesian $\mathbf{x}^c_i$ or Spherical coordinates $\mathbf{x}^s_i$, can be transformed back and forth between the coordinate systems by using the relationships in Equations 23 and 24. These relations were obtained from the geometric definitions of Figure 10 as well as their time derivatives.

$$\mathbf{x}^c_i = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \equiv \mathbf{F}_{cs}(\mathbf{x}^s_i) = \begin{bmatrix} rc_\phi c_\theta \\ rc_\phi s_\theta \\ rs_\phi \\ \dot{r}c_\phi c_\theta - r\dot{\theta}c_\phi s_\theta - \dot{\phi}rc_\theta s_\phi \\ \dot{r}c_\phi s_\theta - \dot{\phi}rs_\phi s_\theta + r\dot{\theta}c_\phi c_\theta \\ \dot{r}s_\phi + \dot{\phi}rc_\phi \end{bmatrix} \tag{23}$$

$$\mathbf{x}^s_i = \begin{bmatrix} r \\ \theta \\ \phi \\ \dot{r} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \equiv \mathbf{F}_{sc}(\mathbf{x}^c_i) = \begin{bmatrix} \sqrt{x^2+y^2+z^2} \\ \arctan\left(\frac{y}{x}\right) \\ \arcsin\left(z/\sqrt{x^2+y^2+z^2}\right) \\ \frac{x\dot{x}+y\dot{y}+z\dot{z}}{\sqrt{x^2+y^2+z^2}} \\ \frac{x\dot{y}-\dot{x}y}{x^2+y^2} \\ \frac{\dot{z}x^2-\dot{x}zx+\dot{z}y^2-\dot{y}zy}{\sqrt{1-\frac{z^2}{x^2+y^2+z^2}}\left(x^2+y^2+z^2\right)^{\frac{3}{2}}} \end{bmatrix} \tag{24}$$

Note that the singularity arising when x and y are *zero* in the calculation of θ and $\dot{\theta}$ can be resolved by using the *atan*2 function. The velocities $\dot{r}$, $\dot{\theta}$ and $\dot{\phi}$ are then solved for using the expressions in Equation 23.

The covariance matrices can be transformed back and forth between the coordinate systems via a similarity transformation [12]. For example, to convert a Cartesian covariance matrix $\mathbf{P}_i^c$ into a spherical one $\mathbf{P}_i^s$, the transformation is as follows:

$$\mathbf{P}_i^s = \mathbf{M}_{sc}\mathbf{P}_i^c\mathbf{M}_{sc}^T \quad \text{where} \quad \mathbf{M}_{sc} = \left.\frac{d\mathbf{F}_{sc}(\mathbf{x}^c)}{d\mathbf{x}^c}\right|_{\mathbf{x}^c=\mathbf{x}_i^c}$$

where the matrix $\mathbf{M}_{sc}$ is the Jacobian of the transformation function $\mathbf{F}_{sc}$ in Equation 24 with respect to the cartesian coordinates $\mathbf{x}^c = [x\, y\, z\, \dot{x}\, \dot{y}\, \dot{z}]$, evaluated at the corresponding position-velocity point $\mathbf{x}_i^c$ where the covariance matrix is sampled. The inverse transformation is constructed in a similar way, using the transformation function $\mathbf{F}_{cs}$ in Equation 23.

Finally, the accelerations can be transformed back and forth betwen the cartesian $[a_x\, a_y\, a_z]$ and spherical $[a_r\, a_\theta\, a_\phi]$ representations via the simple rotation matrix defined in Equation 22.

References

1. Woffinden, D.C.: Angles-Only Navigation for Autonomous Orbital Rendezvous. Ph.D. thesis, Utah State University (2008)
2. Fehse, W.: Automated Rendezvous and Docking of Spacecraft. Cambridge Aerospace Series. Cambridge University Press (2003)
3. Petridis, V.: A method for bearings-only velocity and position estimation. IEEE Transactions on Automatic Control 26(2), 488–493 (1981)
4. Hammel, S.E., Liu, I.P.T., Hilliard, J.E., Gong, K.E.: Optimal observer motion for localization with bearing measurements. Computers Math. Applic. 18(1-3), 171–180 (1989)
5. Oshman, Y., Davidson, P.: Optimization of Observer Trajectories for Bearings-Only Target Localization. IEEE Transactions on Aerospace And Electronic Systems 35(3) (1999)
6. Aidala, V.J., Hammel, S.E.: Utilization of Modified Polar Coordinates for Bearings-Only Tracking. IEEE Transactions on Automacit Control AC-28(3), 283–294 (1983)
7. Mallick, M., Arulampalam, S., Mihaylova, L., Yan, Y.: Angle-only Filtering in 3D using Modified Spherical and Log Spherical Coordinates. In: Proceedings of the 14th International Conference on Information Fusion, Chicago, USA, pp. 1905–1912 (2011)
8. Gelb, A.: Applied Optimal Estimation. MIT Press (1974)
9. Weiss, H., Moore, J.B.: Improved Extended Kalman Filter Design for Passive Tracking. IEEE Transactions on Automatic Control AC-25(4) (1980)
10. Woffinden, D.C., Geller, D.K.: Optimal Orbital Rendezvous Maneuvering for Angles-Only Navigation. Journal of Guidance, Control, and Dynamics 32(4), 1382–1387 (2009)
11. Yamanaka, K., Ankersen, F.: New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit. Journal of Guidance, Control, and Dynamics 25(1), 60–66 (2002)
12. Vallado, D.A.: Covariance Transformations for Satellite Flight Dynamics Operations. In: AAS / AIAA Astrodynamics Specialist Conference, Big Sky, Montana, pp. 1–35 (2003)

A New Substitution Based Recursive B-Splines Method for Aerodynamic Model Identification

Liguo Sun, Coen de Visser, and Qiping Chu

Abstract. A new substitution based (SB) recursive identification method, using multivariate simplex B-splines (MVSBs), has been developed for the purpose of reducing the computational time in updating the spline B-coefficients. Once the structure selected, the recursive identification problem using the MVSBs turns to be a constrained recursive identification problem. In the proposed approach, the constrained identification problem is converted into an unconstrained problem through a transformation using the orthonormal bases of the kernel space associated with the constraint equations. The main advantage of this algorithm is that the required computational time is greatly reduced due to the fact that the scale of the identification problem, as well as the scale of the global covariance matrix, is reduced by the transformation. For validation purpose, the SB-RMVSBs algorithm has been applied to approximate a wind tunnel data set of the F-16 fighter aircraft. Compared with the batch MVSBs method and the equality constrained recursive least squares (ECRLS) MVSBs method, the computational load of the proposed SB-RMVSBs method is much lower than that of the batch type method while it is comparable to that of the ECRLS-MVSBs method. Moreover, the higher the continuity order is, the less computational time the SB-RMVSBs method requires compared with the ECRLS-MVSBs method.

1 Introduction

The control performance of a model-based automatic control system, like for example the adaptive nonlinear dynamic inversion (ANDI) flight control system [4, 7] and the module based adaptive backstepping flight control system [7], heavily relies on the accuracy of the object model that is identified in real-time. Recently, de Visser et al. [11] proposed a novel batch type identification method using multivariate simplex B-splines. Comparing with the ordinary polynomial basis (OPB) based

Liguo Sun · Coen de Visser · Qiping Chu
Delft University of Technology, Delft, The Netherlands, 2600GB
e-mail: {L.sun,c.c.devisser,q.p.chu}@tudelft.nl

method, this simplex spline basis (SSB) based method can provide a relatively more stable basis and enjoys a higher approximation power owing to the fact that multiple local modules are identified instead of identifying a single overall model[10]. Another main merit of the multivariate simplex B-splines (MVSBs) is that they are capable of using the scattered dataset as training data. This is a property that the multivariate sensor product splines method does not have [11].

Later, de Visser and Chu et al. [12] developed an equality constrained recursive least squares (ECRLS) based MVSBs identification method after combining the linear regression formulation of the spline bases from [11] with the recursive least squares identification method from [15]. The recursive identification method presented in [15] can convert a constrained identification problem into a free-of-constraint identification problem . In this recursive identification method, the constrained recursive identification process is circumvented by merely injecting the equality constraint information into the general least square solution calculated using an initial training data collection.

However, in order to enable the real-time aerodynamic model identification, it is still necessary to reduce the computational load of the recursive MVSBs method. This paper is aimed at providing a more effective recursive identification method than the ECRLS-MVSBs method developed in [12]. The new method should enjoy a much lower computational load than the batch MVSBs, and have a lower computational load than the ECRLS-MVSBs method. In this paper, a new substitution based multivariate simplex B-splines (SB-MVSBs) method is developed. The kernel-space bases based transformation can greatly cut down the computational time required by the SB-MVSBs method.

This paper is structured as follows. The preliminaries on the multivariate simplex B-splines are introduced in section 2. The SB-MVSBs method is developed in section 3. In section 4, the proposed SB-RMVSBs method is applied to a wind tunnel data set of the F-16 fighter aircraft, and the selection of the spline function structure is investigated. Subsequently, the proposed method is compared with both the batch method and the ECRLS-MVSBs method in section 5. Finally, this paper is concluded by section 6.

2 Preliminaries on Multivariate Simplex B-Splines

The basic principles for simplex splines are briefly introduced in this section. Without this introduction, the formulation of the SB-MVSBs method will be incomplete.

2.1 Simplex and Barycentric Coordinates

Let t be an n-simplex formed by the convex hull of its $n+1$ non-degenerate vertices $(v_0, v_1, ..., v_n) \subset \mathbb{R}^n$. The normalized barycentric coordinates of some evaluation point $\mathbf{x} \in \mathbb{R}^n$ with respect to simplex t are defined as

$$b(\mathbf{x}) := (b_0, b_1, ..., b_n) \in \mathbb{R}^{n+1}, \ \ \mathbf{x} \in \mathbb{R}^n \tag{1}$$

which follows from the following implicit relation:

$$\mathbf{x} = \sum_{i=0}^{n} b_i v_i, \ \sum_{i=0}^{n} b_i = 1 \tag{2}$$

2.2 Triangulations of Simplices

The approximation power of the multivariate simplex spline is partly determined by the structure of the triangulation. A triangulation $\mathscr{T}$ is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathscr{T} := \bigcup_{i=1}^{J} t_i, \, t_i \cap t_j \in \{\varnothing, \tilde{t}\}, \forall t_i, t_j \in \mathscr{T} \tag{3}$$

with the edge simplex $\tilde{t}$ a k-simplex with $0 \leq k \leq n-1$. High quality triangulations can be obtained using constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented by Shewchuk [8].

2.3 Basis Functions of the Simplex B-splines

According to [3] and [11], the Bernstein basis polynomial $B^d_\kappa(b(\mathbf{x}))$ of degree d in terms of the barycentric coordinates $b(\mathbf{x}) = (b_0, b_1, ..., b_n)$ from Eq. (2) is defined as:

$$B^d_\kappa(b(\mathbf{x})) := \begin{cases} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{4}$$

where $\kappa = (\kappa_0, \kappa_1, ..., \kappa_n) \in N^{n+1}$ is a *multi-index* with the following properties: $\kappa! = \kappa_0!\kappa_1!...\kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + ... + \kappa_n$. In Eq. (4) we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} ... b_n^{\kappa_n}$. Given that $|\kappa| = d$, the total number of valid permutations of the *multi-index* κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!} \tag{5}$$

In [2], it was proved that any polynomial $p(b)$ of degree d on a simplex t can therefore be written as a linear combination of $\hat{d}$ basis polynomials in what is known as the B-form as follows:

$$p^t(b(\mathbf{x})) := \begin{cases} \sum_{|\kappa|=d} c^t_\kappa B^d_\kappa(b(\mathbf{x})) & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{6}$$

with c^t_κ the B-coefficients which uniquely determines $p^t(b(\mathbf{x}))$, where the superscript 't' indicates that p is defined on the simplex 't'. The total number of basis function terms is equal to $\hat{d}$, which is the total number of valid permutations of κ.

2.4 Vector Formulations of the B-Form

As introduced in [12], the vector formulation, according to Eq. (6), for a B-form polynomial $p(b(\mathbf{x}))$ in barycentric $\mathbb{R}^{n+1}$ has the following expression:

$$p^t(\mathbf{x}) := \begin{cases} \mathbf{B}_t^d(b(\mathbf{x})) \cdot \mathbf{c}^t & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases}, \tag{7}$$

with $b(\mathbf{x})$ the barycentric coordinates of the Cartesian $\mathbf{x}$. The row vector $\mathbf{B}_t^d(b(\mathbf{x}))$ in Eq. (7) is constructed from individual basis polynomials which are sorted lexicographically[12].

The simplex B-spline function $s_d^m(b(\mathbf{x}))$ of degree d and continuity order m, defined on a triangulation consisting of J simplices, is defined as follows:

$$s_d^m(\mathbf{x}) := \mathbf{B}^d(b(\mathbf{x})) \cdot \mathbf{c} \quad \in \mathbb{R}, \tag{8}$$

with $\mathbf{B}^d(b(\mathbf{x}))$ the global vector of basis polynomials which has the following full expression:

$$\mathbf{B}^d(b(\mathbf{x})) := [\,\mathbf{B}_{t_1}^d(b(\mathbf{x}))\ \mathbf{B}_{t_2}^d(b(\mathbf{x}))\ \cdots\ \mathbf{B}_{t_J}^d(b(\mathbf{x}))\,] \ \in \mathbb{R}^{1\times J\cdot\hat{d}} \tag{9}$$

Note that according to Eq. (7) we have $\mathbf{B}_{t_j}^d(b(\mathbf{x})) = 0$ for all evaluation locations $\mathbf{x}$ that are located outside of the triangle t_j. This results in that $\mathbf{B}^d$ is a sparse row vector.

The global vector of B-coefficients $\mathbf{c}$ in Eq. (8) has the following formulation:

$$\mathbf{c} := \left[\mathbf{c}^{t_1\top}\ \mathbf{c}^{t_2\top}\ \cdots\ \mathbf{c}^{t_J\top}\right]^\top \ \in \mathbb{R}^{J\cdot\hat{d}\times 1} \tag{10}$$

with each $\mathbf{c}^{t_j}$ a per-simplex vector of lexicographically sorted B-coefficients.

For a single observation on y we have:

$$f = \mathbf{B}^d(b(\mathbf{x}))\mathbf{c} + \varepsilon \tag{11}$$

with ε the residue. Then, for all the N observations, we have the following well-known formulation:

$$\mathbf{f} = \mathbf{X}(b(\mathbf{x}))\mathbf{c} + \xi \in \mathbb{R}^{N\times 1} \tag{12}$$

with $\mathbf{X}(b(\mathbf{x})) \in \mathbb{R}^{N\times J\cdot\hat{d}}$ a collection matrix of the row vector $\mathbf{B}^d$ from Eq. (9), and $\xi = [\varepsilon_1, \varepsilon_2, \ldots \varepsilon_N]^T$ the residue vector. For writing convenience, $\mathbf{X}(b(\mathbf{x}))$ will be written as $\mathbf{X}$ in the remainder of this paper.

2.5 Global Continuity Constraints

The formulation for the continuity conditions from [1] and [3] is used:

$$\mathbf{c}^{t_i}_{(\kappa_0,\ldots,\kappa_{n-1},m)} = \sum_{|\gamma|=m} \mathbf{c}^{t_j}_{(\kappa_0,\ldots,\kappa_{n-1},0)+\gamma} B^m_\gamma(\mathbf{v}),\ \ 0 \le m \le r \tag{13}$$

with $\mathbf{v}$ the Bernstein coordinates of the vertex which only belongs to the i^{th} simplex, $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_n)$ a multi-index independent of κ, $|(\kappa_0, \ldots, \kappa_{n-1}, m) + \gamma| = d$. t_i, t_j denote the i-th and j-th simplices separately.

Eventually, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H} \cdot \mathbf{c} = 0 \tag{14}$$

with $\mathbf{H} \in \mathbb{R}^{(E \cdot R) \times (J \cdot \hat{d})}$ the smoothness matrix [11], R is the number of continuity conditions per edge. E is the number of edges in the specified triangulation. If all the simplices' surfaces connect smoothly on the edges within the whole triangulation, we call the simplex splines globally continuous. Global continuity is determined by Eq. (13) and Eq. (14).

2.6 Spline Function Space and a Polynomial Function Space

In this paper, we use a new type of definition of polynomial function space:

$$P_d(n) := \{p_k(\mathbf{x}) : p_k|_{\mathbf{x}} \in \mathbb{P}_k,\ \forall \mathbf{x} \in \mathbb{R}^n\ and\ \forall k \le d\} \tag{15}$$

with $\mathbf{x}$ the input vector, $\mathbb{P}_k$ the space of polynomials of degree k.

We use the following definition of the spline space, which is a modified form of the definition given by Lai et al. in [3]:

$$S^m_d(n) := \{s^m_d(\mathbf{x}) \in C^m : s^m_d|_{\mathbf{x}} \in \mathbb{P}_d,\ \forall \mathbf{x} \in \mathbb{R}^n\} \tag{16}$$

with $\mathbb{P}_d$ the space of polynomials of degree d, and n the dimension of function inputs.

Note that, the former represents the ordinary polynomial function bases with the order up to d. For example, if we select $\mathbf{x} = [x,\ y]^T$, then $P_2(2) := c_1 + c_2 x + c_4 y + c_3 x^2 + c_6 xy + c_5 y^2$ with x and y two elements of $\mathbf{x}$.

3 Transformation Based Recursive Identification Method

The kernel space information of the equality constraint matrix $\mathbf{H}$, formulated in Eq. (14), has been utilized to transform the constrained recursive identification problem into a free-of-constraint recursive identification problem.

3.1 Transformation of Constraints

Once the triangulation and the spline function structure are chosen, the equality constraints have the property that they are time invariant and known a priori. In this case, a straightforward substitution method can be applied to remove the constraints in each recursion step.

Following from Eq. (8), the original constrained recursive identification problem has the following expression:

$$f = \mathbf{B} \cdot \mathbf{c} + \varepsilon \tag{17}$$

$$\text{s.t. } \mathbf{H} \cdot \mathbf{c} = 0 \tag{18}$$

Assume that the singular value decomposition (SVD) result of $\mathbf{H}$ is as follows:

$$\mathbf{H}_{n\times m} = \mathbf{V}_{n\times n} \begin{bmatrix} \Sigma_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)} \end{bmatrix} \mathbf{U}_{m\times m}^{\top} \tag{19}$$

where $\sum = \operatorname{diag}\left(\sigma_1\ ,\ \cdots\ \sigma_r\right)$ is the diagonal vector of all singular values , $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and r is the rank of $\mathbf{H}$. $\mathbf{V} = \left[\mathbf{V}_1\ \mathbf{V}_2\right]$ is an n_{th} order orthogonal matrix, $\mathbf{V}_1$ is an n by r matrix. $\mathbf{U} = \left[\mathbf{U}_1\ \mathbf{U}_2\right]$ is a m_{th}-order orthogonal matrix, $\mathbf{U}_1$ is an m by r matrix. Because $\mathbf{c} \in null\left(\mathbf{H}\right)$, one feasible general solution for the homogeneous equation Eq. (18) is:

$$\mathbf{c} = \mathbf{U}_2\mathbf{y} \tag{20}$$

where the column vectors of $\mathbf{U}_2$ form an orthonormal basis of $null\left(\mathbf{H}\right)$ [14, 5]. $\mathbf{y}$ is a column vector which needs to be calculated (identified) later, and its length is $m-r$. The feasibility of the above mentioned conversion will be proved later in theorem 1.

By introducing this general solution into Eq. (17), we get the following formation:

$$f = \mathbf{B}\mathbf{U}_2\mathbf{y} + \varepsilon \tag{21}$$

with $\mathbf{U}_2$ a basis for $null\left(\mathbf{H}\right)$. Since Eq. (21) only represents an unconstrained identification problem, a regular recursive least squares identification method becomes capable to solve it. In order to obtain the final unknown parameters (B-coefficients), we only need to substitute the identified vector $\mathbf{y}$ into Eq. (20). The computational flow chart is concluded as follows.

Algorithm 1

step.1 determine the triangulation $\mathscr{T}$, calculate the smoothness matrix $\mathbf{H}$, and carry out the SVD according to Eq. (19) to get $\mathbf{U}_2$.

step.2 calculate the spline basis vector according to Eq. (9).

step.3 identify the unknown vector $\mathbf{y}$ contained by Eq. (21) using regular recursive least squares method.

step.4 reconstruct the B-coefficient vector $\mathbf{c}$ from the vector $\mathbf{y}$ using Eq. (20). Return to step.2 if a new data is available.

Theorem 1: Optimal Approximation
Given $\mathbf{y}$ the unique and optimal least square estimation vector of problem Eq. (21), $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ is the optimal least squares solution of the constrained problem Eq. (17).

Proof
Given $\mathbf{U}_2$ derived from Eq. (19), columns of matrix $\mathbf{U}_2$ constitute orthonormal bases for the kernel space of $\mathbf{H}$. Therefore, we have $\mathbf{H}\mathbf{U}_2 = 0$. Hence, we can get $\mathbf{H}\mathbf{U}_2 \cdot \mathbf{y} = 0$. Because $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ as shown in Eq. (20), we can get $\mathbf{H} \cdot \mathbf{c} = 0$. The equality constraints $\mathbf{H} \cdot \mathbf{c} = 0$ are satisfied during parameter estimation.

Because Eq. (17) and Eq. (21) hold, we have

$$\mathbf{f} - \mathbf{X} \cdot \mathbf{c} = \xi = \mathbf{f} - \mathbf{X}\mathbf{U}_2\mathbf{y} \tag{22}$$

We define the cost function of the least square problem as $C(\mathbf{c}) = \min_{\mathbf{c}} \xi^T \xi$, where $\mathbf{c}$ is the vector to estimate. As $\mathbf{y}$ is the optimal and unique least square solution of problem 21, we assume that it leads to a minimum residual vector ξ_d, so the minimum cost function value can be written as $C(\mathbf{y}) = \xi_d^T \xi_d$. Because the two problems described by Eq. (17) Eq. (18) and Eq. (21) are identical systems in view of the output approximation, we can get the following result: $C(\mathbf{c}) = C(\mathbf{y}) = \xi_d^T \xi_d$ from Eq. (22). □

3.2 Remarks

Note that, according to Eq. (21), the proposed recursive identification method has cut down the scale of the original identification problem by multiplying the regression data matrix by $\mathbf{U}_2$ from the right hand side.

There exist some similarities between the SB-MVSBs method and the orthogonal least squares based identification method presented in [9]. In theory, the singular value decomposition allows to reduce the structure of the aerodynamic model. By keeping all (non-zero) singular values, the SB-MVSBs method has removed the dependent columns in the data matrix. However, it is not reasonable to cut out the smallest singular values and further reduce the scale of the model because the constraints are originally added to the unknown parameters rather than to the regression data matrix.

4 Validation Using Wind Tunnel Data of the F-16 Fighter Aircraft

4.1 F-16 Aerodynamic Model Structure

According to the F-16 aerodynamic wind tunnel data presented in [6], the following structure is a good option for X-direction aerodynamic force (moment) coefficient:

$$\begin{aligned} F_x\left(\alpha,\beta,\delta_e,\delta_{lef},\frac{q\bar{c}}{V}\right) =& f_1\left(\alpha,\beta,\delta_e\right)+f_2\left(\alpha,\beta\right)\cdot\delta_{lef} \\ &+f_3\left(\alpha\right)\cdot\frac{q\bar{c}}{V}+f_4\left(\alpha\right)\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef} \end{aligned} \tag{23}$$

Note that the engine thrust is assumed to be constant and its related term is removed from Eq. (23). According to Eq. (23), once the q, V and δ_{lef} are fixed, we can derive the following linear regression formulation for a three dimensional MVSBs function.

$$S(x) = \mathbf{B}\cdot\mathbf{c} \tag{24}$$

where **B** is the B-form spline vector calculated using Eq. (9).

According to de Visser [13], the global continuity matrix **H** for the three dimensional MVSBs function should be calculated using Eq. (13).

In the simulation, an aerodynamic model of the F-16 aircraft was identified using simulated flight test data generated with a nonlinear F-16 simulator based on a NASA wind tunnel dataset [6]. The training inputs of the simulated flight test dataset were obtained by generating 20,000 uniformly distributed inputs within their own valid regions. The inputs of the test dataset containing 4331 points are produced by the grids determined by α and β. The system output were calculated through the high resolution interpolation from the wind tunnel data provided by [6] with $\delta_{lef} = 1^\circ, V = 600\,ft/s, q = 0.1\,rad/s, \bar{c} = 11.32\,m$. Moreover, the model outputs of the aerodynamic model is contaminated artificially by adding a white noise with a magnitude of 1% (relative to its maximum and minimum value).

4.2 Cross Validation Results in Determining the Structure

In the numerical simulation, we have chosen the MVSBs function to have only one three dimensional sub-function. The notation $S_d^m(n)$ from Sec. 2 has been used, and the overall spline function becomes the following expression:
$S(x) = S_d^m(n)$, where $n = 3$, while d, m are kept undetermined. The partitioning vector of α is $[-20\ 10\ 40]$. The partitioning vector of β is $[-25\ 25]$. The partitioning vector of δ_e is $[-20\ 20]$. In order to enhance the approximation ability of this algorithm, all the inputs are normalized into the closed range of $[0\ 1]$. In order to select a suitable structure for the spline model of C_m (i.e. the nondimensional pitch moment coefficient), the effects of the structural parameters (i.e. d and m) will be investigated. To demonstrate the approximation power of the SB-MVSBs method, we compared it with the batch MVSBs method.

Fig. 1 shows the root mean squared errors (RMSE) of the fitting outputs (C_m) using the ordinary polynomial basis (OPB) based recursive least squares identification method.

Fig. 2(a) and Fig. 2(b) show the RMSE of the training data set using the batch MVSBs method and the proposed SB-MVSBs method respectively. Comparing these two figures, it has been found that the SB-MVSBs method enjoys the same level of approximation power as that of the batch MVSBs.

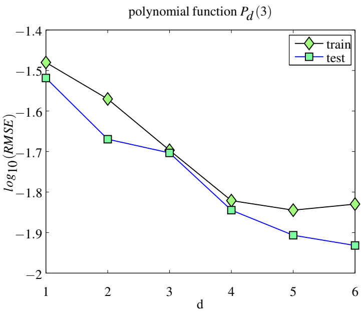
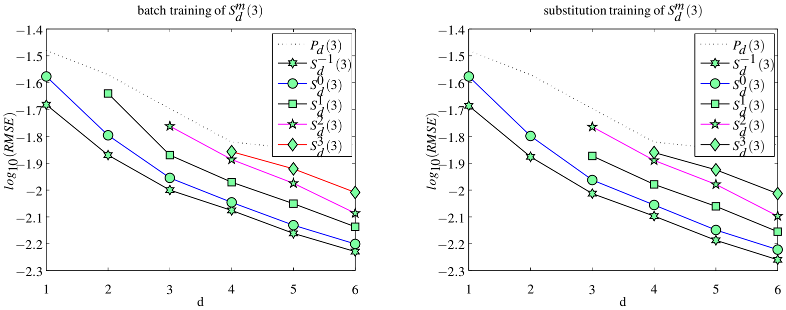


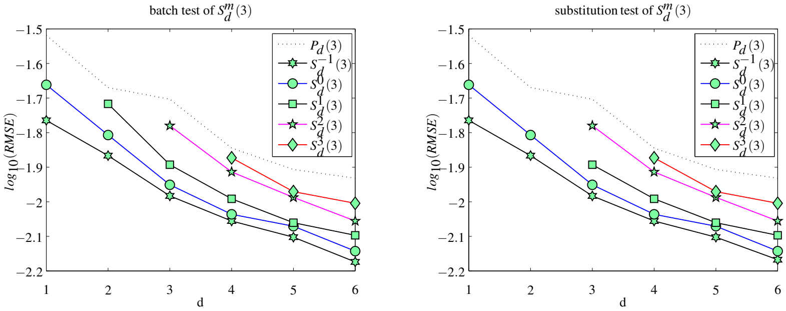
Fig. 1 Different selection of d for $P_d(3)$, C_m



(a) batch MVSBs training.

(b) SB-MVSBs training.

Fig. 2 Different combination of m and d for $S_d^m(3)$, $\mathscr{T}_{12}, C_m$



(a) batch MVSBs test.

(b) SB-MVSBs test.

Fig. 3 Different combination of m and d for $S_d^m(3)$, $\mathscr{T}_{12}, C_m$

Fig. 3(a) and Fig. 3(b) show the RMSE of the testing data set based on the B-coefficients identified using the batch MVSBs method and the SB-MVSBs method respectively. As can be seen from these two figures, the approximation power of the batch MVSBs method and the SB-MVSBs method are very close. Moreover, compared with the results shown in Fig. 1, Fig. 3 indicate that both the batch MVSBs method and the SB-MVSBs method enjoy a higher approximation power than the OPB based recursive identification method.

5 Comparison with the ECRLS-MVSBs and the Batch MVSBs

5.1 Computational Complexity

The computational complexity of the substitution based MVSBs (SB-MVSBs) method is split into two parts. Firstly, according to Eq. (21), the multiplication between the **B** vector and the $\mathbf{U}_2$ matrix needs $m \cdot (m-r)$ with r the rank of the continuity matrix, and m the length of the B-coefficient vector **c**. Similar to the ECRLS method, the computational complexity for the pure regression process using the recursive least squares is $\mathscr{O}\left(3(m-r)^2\right)$. By summing them up, we can get the total computational complexity of the SB-MVSBs method: $C(m,r) = (m-r)\cdot(4m-3r) = 3r^2 - 7mr + 4m^2$. The computational complexity in time of the batch MVSBs method, the ECRLS-MVSBs method and the SB-MVSBs method are tabulated in Table 1.

Given m, function $C(m,r)$ monotonously increases as $r < m$. Therefore the minimum computational complexity of the SB-MVSBs method is $4m^2$ when $r = 0$, while its highest limit is 0. In addition, $C(m,r) = 3n^2$ holds when $r = \frac{(7-\sqrt{37})}{6}m$.

Table 1 Computational Complexity (CC) in time

Methods	batch MVSBs	ECRLS-MVSBs	SB-MVSBs
CC	$\mathscr{O}(m^3)$	$\mathscr{O}(3m^2)$	$(m-r)\cdot(4m-3r)$

5.2 Computational Time Comparison with the ECRLS-MVSBs

In order to reveal the influence of the continuity order m on the computational complexity in time, a numerical experiment is performed with different selection for the continuity order m. In the remainder of this paper, we will always choose the MVSBs function to have only one three dimensional sub-function in all of the

Table 2 Computational time for 20k data of C_m, $\mathscr{T}_{12}$, B-coefficient number 1008, $S_6^m(3)$

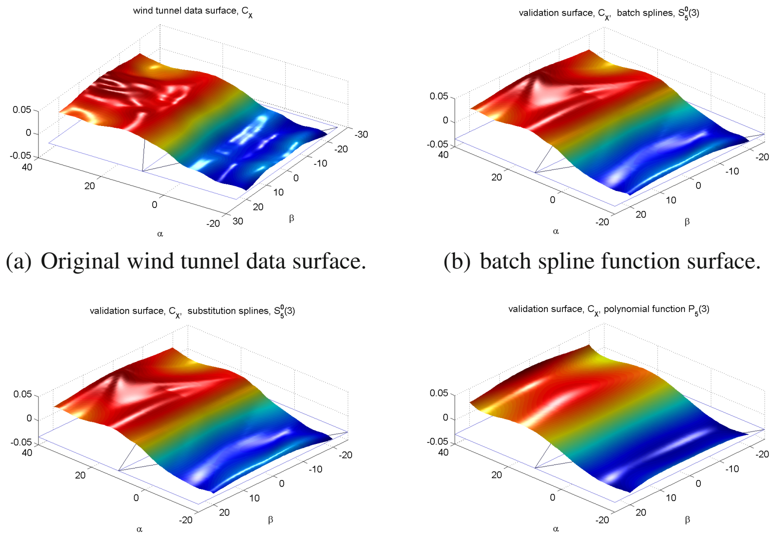
condition	$S_6^{-1}(3)$	$S_6^0(3)$	$S_6^1(3)$	$S_6^2(3)$	$S_6^3(3)$	$S_6^4(3)$
ECRLS	104.5092	105.5291	105.0324	106.2780	106.2854	106.6970
SB-MVSBs(operated)	101.7709	33.2808	13.4270	5.3797	4.2410	3.7263
SB-MVSBs(normal)	139.4835	67.7009	24.0644	7.9068	6.0565	5.6464

numerical experiments. The simulation results are listed in Table 2. In Table 2, '*operated*' means that the $\mathbf{BU}_2$ multiplication shown in Eq. (21) is executed in advance in a batch manner. According to Table 2, the SB-MVSBs method require less computational time than the ECRLS-MVSBs method, and this advantage will become more apparent with the increase of the continuity order m.

5.3 Evaluation Results on the Approximation Power

The OPB based recursive identification method, the batch MVSBs method and the SB-MVSBs recursive identification method are utilized to fit the same training data set of C_x respectively. The models identified using these three different methods respectively are validated using the testing data that are located on the mesh grids. The validation surfaces of C_x are shown in Fig. 4. Apparently, the SB-MVSBs method enjoys an equal fitting accuracy to that of the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.

The OPB based recursive identification method, the batch MVSBs and the SB-MVSBs recursive identification methods are utilized to fit the same training data set of C_m. The models identified using three different methods are validated using the same testing data set as that mentioned previously. The validation surfaces of C_m are plotted in Fig. 5. We can get a similar conclusion as that drawn from last experiment that the SB-MVSBs method has the same fitting power as the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.



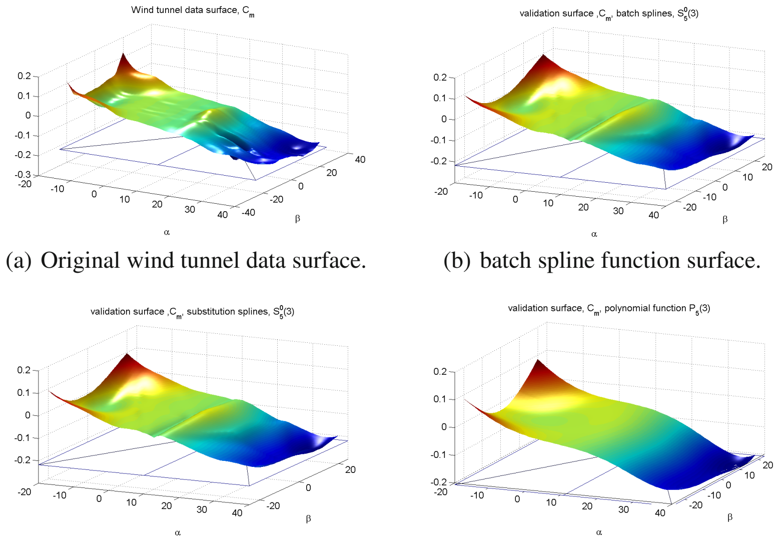
(a) Original wind tunnel data surface.

(b) batch spline function surface.

(c) substitution spline function surface.

(d) Polynomial fitting surface.

Fig. 4 Validation surface of C_x ($\delta_e = 2^o$), $\mathscr{T}_{12}$



(a) Original wind tunnel data surface.

(b) batch spline function surface.

(c) substitution spline function surface.

(d) Polynomial fitting surface.

Fig. 5 Validation surface of C_m ($\delta_e = 2^o$), $\mathscr{T}_{12}$

6 Conclusions

A new substitution based recursive MVSBs method is proposed for the online aerodynamic model identification. In view of the equality constraints contained by the MVSBs, a SVD based transformation is empoyed to convert an originally constrained recursive identification problem into a free-of-constraint identification problem. The proposed recursive model identification method namely SB-MVSBs method was applied to approximate a series of two wind tunnel data sets of F-16 aircraft, and were compared with the batch MVSBs method and the ECRLS-MVSBs method. The numerical simulation results show that the proposed SB-MVSBs method requires less computational time than the batch MVSBs method and the ECRLS-MVSBs method. In addition, the computational time required by the SB-MVSBs decreases with the increase of the continuity order m. The reduction of the computational time is caused by the fact that the kernel space bases based transformation has cut down the scale of the original spline basis based model.

References

1. Awanou, G., Lai, M.J., Wenston, P.: The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differencial Equations. Wavelets and Splines (2005)
2. de Boor, C.: B-form basics. In: Farin, G. (ed.) Geometric Modeling: Algorithms and New Trends. Society for Industrial and Applied Mathematics (1987)
3. Lai, M.J., Schumaker, L.L.: Spline Functions on Triangulations, Cambridge (2007)

4. Lombaerts, T.J., Smaili, M.H., Stroosma, O.: Piloted simulator evaluation results of new fault-tolerant flight control algorithm. Journal of Guidance, Control and Dynamics 32(6), 1747–1765 (2009)
5. Meyer, C.D.: Matrix Analysis and Applied Linear Algebra. SIAM:Society for Industrial and Applied Mathematics (2001)
6. Nguyen, L.T., Ogburn, M.E., Gilbert, W.P.: Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability. Nasa technical paper 1538, Langley Research Center, Hampion, Virginia (1979)
7. van Oort, E.R., Sonneveldt, L., Chu, Q.P., Mulder, J.A.: Full-envelope modular adaptive control of a fighter aircraft using orthogonal least squares. Journal of Guidance, Control and Dynamics 33(5), 1461–1472 (2010)
8. Shewchuk, J.R.: Delauney refinement algorithms for triangular mesh generation. Computational Geometry 22, 21–74 (2001)
9. Stark, J.: Adaptive model selection using orthogonal least squares methods. In: Mathematical, Physical and Engineering Sciences, pp. 21–42. The Royal Society, Britain (1997)
10. de Visser, C.: Global nonlinear model identification with multivariate splines. Ph.D. thesis, Delft University of Technology, The Netherlands (2011)
11. de Visser, C.C., Chu, Q.P., Mulder, J.A.: A new approach to linear regression with multivariate splines. Automatica Journal 45, 2903–2909 (2009)
12. de Visser, C.C., Chu, Q.P., Mulder, J.A.: Differential constraints for bounded recursive identification with multivariate splines. Automatica Journal 47, 2059–2066 (2011)
13. de Visser, C.C., Mulder, J.A., Chu, Q.: Global nonlinear aerodynamic model identification with multivariate splines. In: AIAA Atomospheric Flight Mechanics Conference. AIAA, Chicago (2009)
14. Zhang, X.D.: Matrix Analysis and Applications. Tsinghua University Press (2004)
15. Zhu, Y., Li, X.R.: Recursive least squares with linear constraints. Communications in Information and Systems 7(3), 287–312 (2007)

Spacecraft Fault Detection and Isolation System Design Using Decentralized Analytical Redundancy

Saurabh Indra and Louise Travé-Massuyès

Abstract. Fault detection and isolation (FDI) functionality constitutes a critical element of spacecraft fault protection system capabilities. The FDI schemes currently implemented on board operational spacecraft suffer from a lack of systematic design methods. This leads to issues of decreased robustness. While model based diagnosis techniques can resolve a number of these issues, their operational applicability to spacecraft has been limited, largely due to an unfavourable net value proposition. This paper presents an approach integrating analytical redundancy based diagnosis into a conventional spacecraft FPS architecture. The approach centers on a novel decentralized diagnosis architecture based on analytical redundancy relations. A systematic procedure for designing such decentralized model based diagnosers for spacecraft is discussed, with a focus on the attitude and orbit control system. Analytical redundancy relation based error monitors and activation rules relying on the corresponding fault signatures are derived during the design phase. A comparison with the diagnosis functionality as currently implemented in the Cassini attitude and articulation control system fault protection is presented in terms of the design & development effort. It is demonstrated that the presented diagnoser design approach addresses several issues with the conventional methods, while having reasonable additional costs

1 Introduction

The space missions of the future envisage autonomous spacecraft operation in challenging environments. Robust and capable *fault protection* is an enabling technology for such missions. Fault protection is a mix of hardware and software mechanisms aiming to increase the robustness of space systems. The elements of a fault protection system which detect and (possibly) isolate faults constitute the *diagnoser*.

Saurabh Indra · Louise Travé-Massuyès
CNRS, LAAS, 7. Avenue du colonel Roche, F-31077 Toulouse, France
University of Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077
e-mail: sindra@ieee.org, louise@laas.fr

Traditionally, fault diagnosis onboard spacecraft has relied on rule based techniques. Most of the fault monitors utilized rely on simple mappings from observed symptoms to probable diagnosis, with other techniques being used on a case to case basis. Certain key variables of the system are monitored, and a fault is signalled when the variable is out of the expected nominal range. *Activation rules* respond to subsets of the error monitor outputs and diagnose the cause of anomalous behaviour at the component or functional level. This reliance on *symptoms* instead of an *underlying model of behavior* leads to opacity of structure and behavior. The possibility of different symptoms triggered by the same underlying fault, different priorities among faults, mission modes and other system wide considerations have to be taken into account. Such considerations lead to a *patchwork* of monitors, activation rules and the parameter sets associated with them.

With increasing ambitions for space missions and the associated rise in space system complexity, scaling up such rule based diagnosers is proving difficult. The core issue is the lack of transparency in requirements, design, structure and resulting behavior as discussed by Rasmussen [1].

In contrast to rule based methods, the basic principle of model based diagnosis (MBD) is to use a model of the system with sensed observations during operation to detect and isolate faults. Basing diagnosis decisions on a system model can address some of the crucial scalability and structural transparency issues associated with rule based diagnosers. It would seem then, that utilizing model based techniques could lead to more effective fault protection systems. However the actual use of MBD techniques has been constrained due to the high associated costs and risks relative to the benefits provided.

There are two main streams of MBD, originating from different communities. While the DX or consistency based approach originates from work in the computer science and artificial intelligence areas, the FDI stream is rooted in systems and control theory. The two streams emphasize different diagnosis functionalities. Livingstone and Livingstone 2, flown as experiments onboard the Deep Space 1 and Earth Observor 1 spacecraft are examples of diagnosers based on the DX approach to MBD. However, there has not been significant mission pull for adoption of such consistency based techniques for fault diagnosis onboard operational spacecraft since then. The unfavourable net value proposition for the DX stream of MBD is discussed in Kurien & Moreno [2].

Analytical redundancy based MBD is a technique utilizing the FDI approach. Using observers to model nominal and faulty system dynamics is one way to realize analytical redundancy. An early theoretical survey of these techniques and their utility for aerospace systems can be found in Patton [3] and there are various examples of operational systems [4].

The second route to implementing analytical redundancy is based on *analytical redundancy relations* (ARRs). This technique is based on using sensing and structural redundancies in a system to compile consistency checks known as ARRs *offline*. These ARRs are then evaluated online as residual generators using sensed quantities from the system. We utilize as starting point in our work an ARR based approach to diagnosis based on an algorithm discussed in Krysander et al. [5], and extended in Krysander et al [6].

The underlying concepts, assumptions and approaches of the DX and FDI streams have recently been compared and proved to be equivalent under certain conditions Cordier et al. [7]. However the emphasis in diagnosis functionality and conditions for their optimal usage are different.

One of the most complex and capable FPS operational in space was developed for the Cassini spacecraft and can be considered illustrative of the state of practice of conventional design for interplanetary probes. This FPS is used both to illustrate the challenges involved in FPS design, implementation and operation and to assess the value of applying our decentralized diagnosis architecture. The driving system level FP considerations for the Cassini spacecraft are discussed in Slonsky [8].

Our diagnosis approach is based on ARRs and is therefore relevant for continuous state systems modelled for example with a system of differential-algebric equations or as state space models. The behavior of the attitude and orbit control system is usually modeled in such frameworks. Therefore we concentrate in particular on the subsystem level FP monitoring the attitude and articulation control system of the Cassini as discussed in Brown et al [9].

Instead of utilizing a patchwork of different techniques for the design of fault monitors for different fault types as discussed in Lee [10] and Macala [11], the presented integrated design method utilizes a structural model of the ADCS to derive ARR based fault monitors. The fault signatures associated with these monitors are also derived during the design phase. The approach is based on a novel decentralized ARR based diagnosis architecture. The *hierarchically scalable* nature of the architecture allows systematic design and analysis of fault monitors for different monitoring levels. The architecture thus addresses some of the structural and behavioral transparency issues as discussed in Rasmussen [1] and Slonski [8]. Additionally, the net value proposition of the ARR based diagnosers is demonstrated to be positive compared to the conventional approaches.

The paper is structured as follows. The issues with conventional FPS design are described in section 2, utilizing the fault protection of the attitude and articulation control system of the Cassini as a case study. Section 3 starts with an discussion of the ARR based approach to diagnosis followed by a description of the decentralized diagnosis architecture. A comparison between this architecture and the conventional diagnosis techniques used for the Cassini is then provided in section 4. The paper concludes with a discussion of the contribution and perspective for future work in section 5.

2 Fault Protection Systems

Mechanisms and strategies implemented for increased robustness constitute fault protection. The scope and sophistication of onboard FP functionality is determined by mission specific considerations such as the autonomy level required onboard, communication possibilities with the ground segment etc.

Most spacecraft implement standard FP functions which respond to system level effects. Safe mode responses configure the spacecraft to a power positive, thermally safe state. The safe mode(s) also ensures the availability of a robust link with the ground segment, so that the ground segment has access to housekeeping telemetry. Other examples of standard FP strategies are the command-loss and under-voltage responses. Besides these standard system level responses, subsystem level FP is also implemented depending upon the complexity of the spacecraft and mission requirements. The Cassini FP aims to ensure robustness of the mission to all probable single point failures. We focus on the subsystem level fault protection of the attitude and articulation control system (AACS) in the following discussion.

The conventional *monitor-response architecture* forms the basis of the AACS fault protection system. This structure is illustrated in the figure 1. *Error monitors* and *activation rules* make up the diagnosis elements, while *response scripts* and the *repair manager* implement the reconfiguration functionality. Monitors compare sensed values of quantities to expected values and output a health status. Activation rules use subsets of monitor outputs together with the hardware configuration and activity goals to diagnose the likely fault(s).

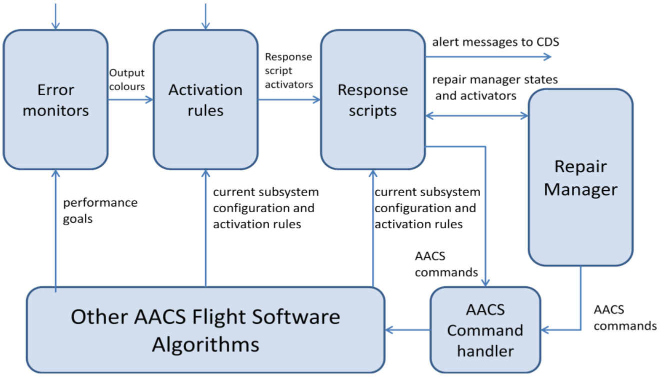

Error monitors
Output colours
Activation rules
Response script activators
Response scripts
alert messages to CDS
repair manager states and activators
Repair Manager
performance goals
current subsystem configuration and activation rules
current subsystem configuration and activation rules
AACS commands
AACS commands
Other AACS Flight Software Algorithms
AACS Command handler

Fig. 1 The structure of the Cassini attitude and articulation control subsystem level fault protection

It is interesting to study the techniques used to implement fault monitors for the different components and control loops of the AACS as illustrated in figure 2. The wide range of underlying diagnosis techniques for these fault monitors can be seen. Component level monitoring is provided by thresholds on individual quantities such as reaction wheel currents. Monitoring at the control loop level is implemented using the control error and its derivative by monitors known as state-space fault monitors. The functioning of the loop is classified as *acceptable* if the control error is below a specified threshold. If the error is reasonably small and decreasing, the functioning is *tolerable*. Large errors which are not decreasing indicate faulty functioning of the control loop. These monitors are a simple form of model based diagnosis as there is a system model encapsulated in the controller trying to minimize the loop error.

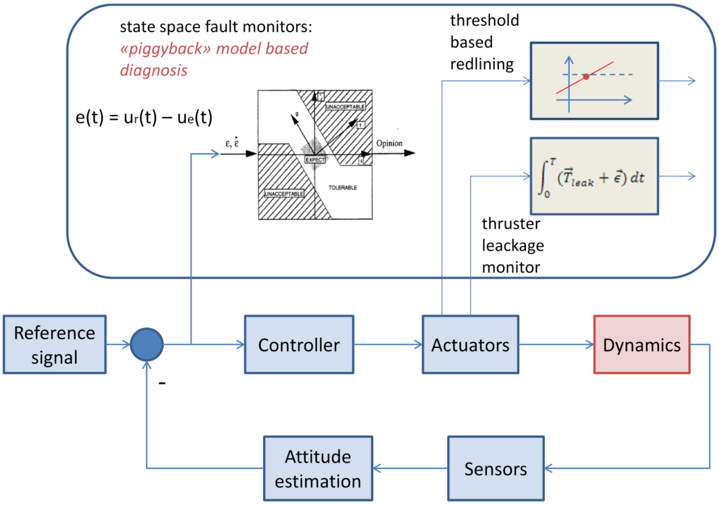


Fig. 2 Different techniques utilized for the Cassini AACS FPS

Such monitors run *piggyback* on the model for control instead of a diagnostic model developed seperately, avoiding additional costs.

The thruster leackage detection monitor is required to detect a leak on one of the eight primary thrusters. Such a leak will cause fuel wastage due to the compensating control which will be triggered. Such leaks need to be detected even while the spacecraft is executing a maneuver. A state space monitor could not be designed as there is no one quantity in the control loop which could signal such a fault during a maneuver. So a model based approach, relying on monitoring deviations from the expected dynamics of the spacecraft was used instead. The resulting thruster leackage monitors are analogous to analytical redundancy relations.

We identify now the issues with the conventional FPS structure and development techniques. The basic problem is the application of various diagnosis techniques and associated analysis methods depending on the detection requirements on a case to case basis. The techniques utilized for the Cassini AACS fault monitors range from rule to model based methods as discussed earlier and seen in figure 2. While the various forms of analysis required for each of the techniques add to the development effort, the resulting structure of the diagnosis elements suffers from a *lack of architectural pattern*. The lack of an integrated architecture complicates the task of setting parameters and working out activation rules. This effort is shifted to a large extent to an ad hoc one based on simulation. With increasing system complexity such an approach does not scale well, leading to opacity of diagnoser structure and behaviour, the possibilty of *emergent behavior*, and consequently lower robustness.

These issues are summarized briefly below:

- Absence of architectural pattern: A bottoms up approach of fitting techniques to requirements & problems on a case to case basis.
- Lack of explicit models explaining what caused a monitor to be triggered.
- Distribution of state and behavioral information among complex parameter sets: Thresholds, persistence counters, disable/enable flags, timers etc.

Many of these issues are connected to the special situation of fault protection 'systems' as compared to functional subsystems like AOCS and power etc. Fault protection functionality for a system is a *set of capabilities* spread over the functional subsystems. However, it is also necessary to view FP capability as constituting a *virtual subsystem* in its own right, because the interactions among capabilities built into seperate subsystems should be worked out as early as possible, and sound systems engineering practices followed during development and testing.

The decentralized architecture developed in this thesis can serve to address a few of these challenges, and this attempt is described in the following sections with the decentralized architecture itself and then with its application to the Cassini FPS.

3 Decentralized Diagnosis with Analytical Redundancy Relations

In order to describe the proposed decentralized diagnosis architecture based on ARRs, we introduce first the basic notions associated with the structural approach to ARRs. Given the emphasis of this paper, and the space constraints, both this introduction and the following description of the decentralized architecture are developed in an *intutive* rather than formal fashion. The reader can look to the references for formal description of the concepts involved.

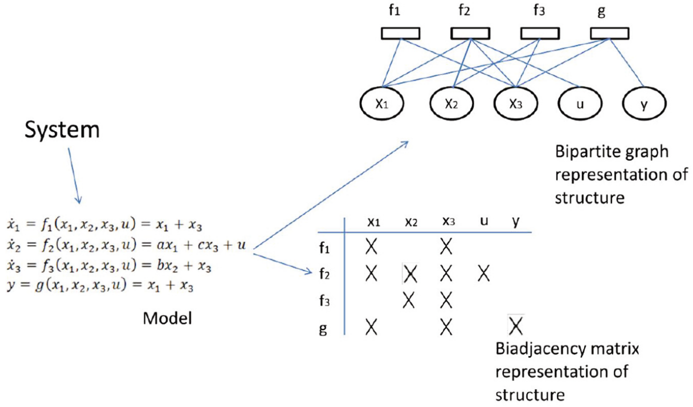
3.1 The Structural Approach to Analytical Redundancy Relations

Analytical redundancy relations rely on using redundancies in the system to compile consistency checks known as residual generators offline. The particular approach to ARR based diagnosis utilized here is based on designing **residual generators** based on **structural** redundancies in the system. These residual generators serve as consistency checks, using sensed quantites from the system to check whether monitored sections of the system are functioning normally. A residual generator takes as input the values of certain observed variables and, in an *ideal case* i.e without unmodeled behavior, noise or disturbances, gives a non-zero output when the behaviour is inconsistent with the model. A detailed description of the structural approach to ARRs can be found in [12].

The process of deriving ARRs begins with a model of the system in the form of a system description as seen in figure 3. The system description consists of a set of equations involving a set of variables. The set of variables is partitioned into a set of *known (or observed)* variables denoted as **Z** and a set of *unknown (or unobserved)* variables denoted as **X**. We refer in the following discussion to the vector of known variables as **z** and the vector of unknown variables as **x**. The system description or model, denoted as $\mathbf{M}(\mathbf{z},\mathbf{x})$ or **M**, is then a set of equations relating the known variables **z** and the unknown variables **x**. The class of models considered here are *differential-algebric equation* systems. Therefore the equations $m_i(z,x) \subseteq M(z,x)$, $i = 1,\ldots,n$, are differential or algebraic equations in **z** and **x**. For the model of figure 3, $\{x_1,x_2,x_3\}$ is the set of unobserved variables, while $\{u,y\}$ is the set of observed variables. Obtaining ARRs for a model involves the elimination of unobserved variables to arrive at a consistency check which can be evaluated based on the sensed quantities.

The *structure* of a system is a representation of which variables are involved in the equations which make up the model of the system. Such a structural abstraction facilitates deriving redundancies while disregarding the actual analytical expressions of the equations making up the system model. Ignoring the analytical expressions enables the consideration of nonlinear systems, and the use of efficient algorithms while deriving *possible* redundancies. However, the results obtained with such a structural representation are best case scenarios. Causality considerations and the algebraic and differential loops in the DAE system ultimately determine which of the theoretically possible structural redundancies can in fact be exploited for the derivation of residual generators. A variable elimination technique and procedure must then be utilized to derive residual generators involving only observed variables.

A *bipartite graph* can be used to represent the structure of the system and deduce possible paths for variable substitution. To define a bipartite graph representation of the structure of a system let us denote the sets of vertices as $\mathscr{C}$ and $\mathscr{V}$, representing the set of constraints and the set of variables respectively. A vertex $c_i \in \mathscr{C}$ is connected by an edge to the vertex $v_j \in \mathscr{V}$ if and only if the constraint c_i involves the


f1
f2
f3
g
X1
X2
X3
u
y
System
Bipartite graph representation of structure
Model
Biadjacency matrix representation of structure

$$\dot{x}_1 = f_1(x_1,x_2,x_3,u) = x_1 + x_3$$
$$\dot{x}_2 = f_2(x_1,x_2,x_3,u) = ax_1 + cx_3 + u$$
$$\dot{x}_3 = f_3(x_1,x_2,x_3,u) = bx_2 + x_3$$
$$y = g(x_1,x_2,x_3,u) = x_1 + x_3$$

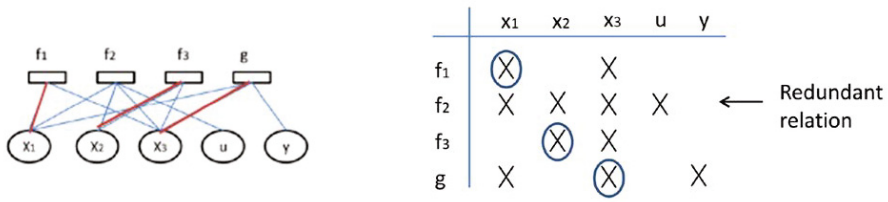
	x1	x2	x3	u	y
f1	X		X		
f2	X	X	X	X	
f3		X	X		
g	X		X		X

Fig. 3 Structural Modeling of a System

variable v_j. Referring to the system model $M(z,x)$ introduced above, the equations $m_i(z,x) \subseteq M(z,x)$, $i = 1,\dots,n$ constitute the set of constraints (C). The set of variables $\mathscr{V}$ is composed of the sensed and unsensed variables $\mathscr{V} = \mathbf{Z} \cup \mathbf{X}$. However for the purpose of finding substitution paths, it is interesting to consider the bipartite graph between the model equations and the unobserved variables - i.e. $\mathscr{V} = \mathbf{X}$.

It can be shown that ARRs correspond structurally to so called *complete matchings* between X and $\mathscr{C}$ on the bipartite graph $G(M \cup X \cup Z, \mathscr{A})$, or equivalently on $G(M \cup X, A)$, where $A \subseteq \mathscr{A}$ and A is a set of arcs such that $a(i,j) \in A$ if and only if variable x_i is involved in relation m_j. A complete matching between X and M, provides a structural path to eliminate the unobserved variables and arrive at a consistency check. A complete matching is denoted as $\mathscr{M}(X,M)$, or simply $\mathscr{M}$ in case there is no ambiguity.

Equivalently, ARRs correspond to **minimal structurally over determined** (MSO) sets, which are sets of equations of the system with one more equation than unknowns Krysander et al [5]. Unobserved variables can be eliminated, and then the redundant equation used to check for consistency as seen in figure 4. While complete matchings on bipartite graphs provide an intutive, graphical view of structural redundancies, the biadjacency matrix and MSO sets approach is used to implement efficient algorithms.



	x1	x2	x3	u	y	
f1	⊗		X			
f2	X	X	X	X		← Redundant relation
f3		⊗	X			
g	X		⊗		X	

Fig. 4 The presence of redundancy in a structural sense: A Minimal Structurally Overdetermined (MSO) Set and a Complete Matching

The number of MSO sets increases exponentially with the degree of structural redundancy present in the system. Rather than deriving all possible MSO sets, the idea of minimal test equation support (MTES), was introduced in Krysander et al. [6] to limit the derived structural redundancies to those responsive to a set of interesting faults. Corresponding to each MTES the corresponding fault sensitivity can also be derived using the algorithm presented.

The (centralized) diagnosis scheme based on analytical redundancy relations can be seen in figure 5. The structural model of the system serves as input to the diagnoser design phase. An MSO or minimal test equation support (MTES) signifies the theoretical presence of a structural redundancy which could be used to develop a consistency check for a part of the system. The corresponding minimal test support (MTS) represents the faults which can be detected with this consistency check. In this way the MTS sets characterize the maximum possible fault isolability. Whether a residual generator can be analytically derived depends upon the causality restrictions on the equations in the set and the presence of algebraic and differential loops. We use in our work the residual generator derivation method proposed in Svard et al [13]. This method relies on deriving a computational sequence to successively solve for the unknown variables involved in an equation set. One redundant equation together with the developed computational sequence constitutes a **sequential residual generator**. After offline design, the diagnoser is implemented as a residual generator bank.

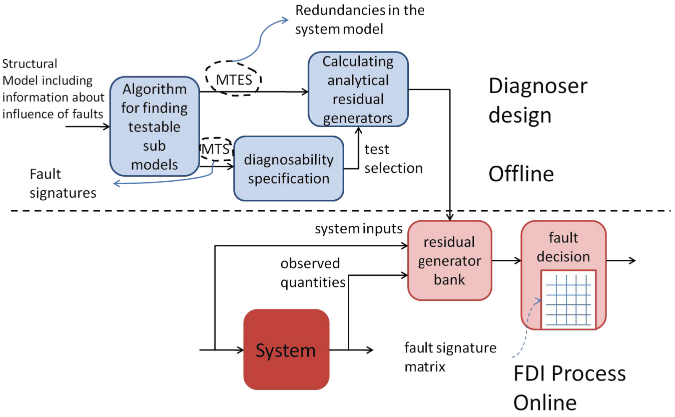

Redundancies in the system model
Structural Model including information about influence of faults
Algorithm for finding testable sub models
MTES
Calculating analytical residual generators
Diagnoser design
MTS
diagnosability specification
test selection
Fault signatures
Offline
system inputs
observed quantities
residual generator bank
fault decision
System
fault signature matrix
FDI Process Online

Fig. 5 Diagnosis with Analytical Redundancy Relations

3.2 The Decentralized Diagnosis Architecture

Having discussed briefly the basic notions of the structural approach to ARR based diagnosis, we present intutively here the decentralized diagnosis architecture. In this architecture, local diagnosers rely on models of their subsystems to arrive at local diagnosis. Ambiguities might arise as faults propogate between subsystems. A supervisor at the higher level serves to resolve ambiguities and provide diagnosis at a higher resolution than that possible with purely local information. The architecture is hierarchically scalable as can be seen in figure 6. This means that the supervisor of one level can function as the local diagnoser for the next higher level.

As discussed earlier, the structural approach to deriving analytical redundancy relations can be viewed as one of finding complete matchings on the bipartite graph representation of the structure.

The following model is used to illustrate the notions. It is composed of six equations r_{1-6} relating the unobserved variables $X = \{x_1, x_2, x_3, x_4, x_5\}$ and the observed variables $Z = \{u, v, w\}$.

$$r_1 : \dot{x}_1 = -x_1^2 + x_3 + u \tag{1}$$

$$r_2 : \dot{x}_2 = x_4^2 \tag{2}$$

$$r_3 : x_1 = 3 \cdot x_2^3 + v \tag{3}$$

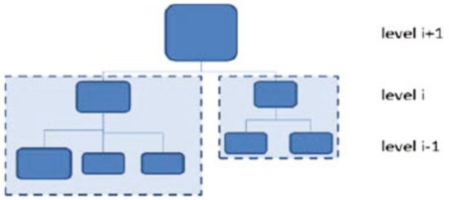
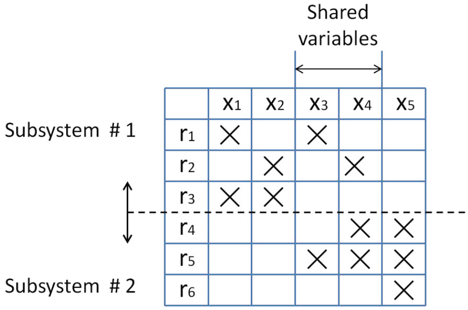


Fig. 6 Illustration of the basic diagnoser structure


Shared variables
Subsystem # 1
Subsystem # 2

	x_1	x_2	x_3	x_4	x_5
r_1	×		×		
r_2		×		×	
r_3	×	×			
r_4				×	×
r_5			×	×	×
r_6					×

Fig. 7 Divison of a system model into subsystems

$$r_4 : y = x_4 + x_5 \tag{4}$$

$$r_5 : \ddot{x}_3 = x_4^2 + x_5 \tag{5}$$

$$r_6 : w = x_5 \tag{6}$$

The biadjacency matrix and bipartite graph representations of the structure of this model can be seen in figure 8.

To introduce the notions behind the decentralized architecture intutively, consider the system is divided into two subsystems as in figure 7. While the variables x_1, x_2 are *local* to subsystem 1 and the variable x_5 local to subsystem 2, the variables x_3, x_4 are *shared* between the two subsystems. The set of variables is therefore divided into local and shared variables.

A complete matching for the global system can be seen in figure 8 both on the bipartite graph and the biadjacency matrix. The use of the complete matching to eliminate all unknown variables is also illustrated with a series of matchings and substitutions. The relation r_6 is used as the redundant relation to serve as the consistency check. Also observe that the sensed variables u, v, y, w are only considered implicitly in the structural representations.

Now consider the situation when we try to use the structural representation of the subsystems as available to the local diagnosers working on the two subsystems as seen in figure 9. The concepts of local complete matchings and shared relations have

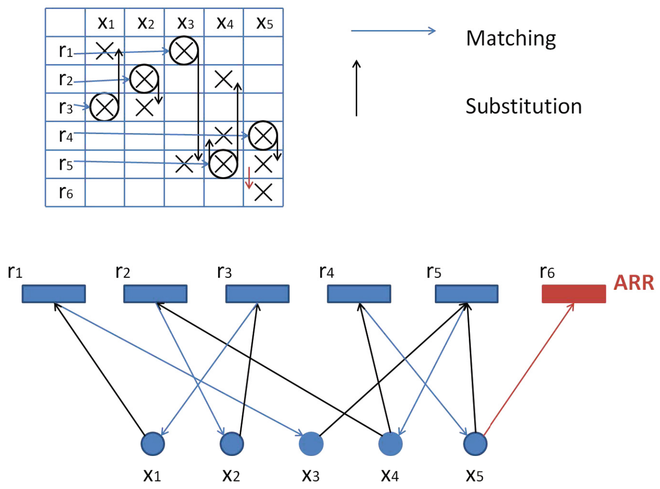


Fig. 8 Structural derivation of a redundancy relation at the global level

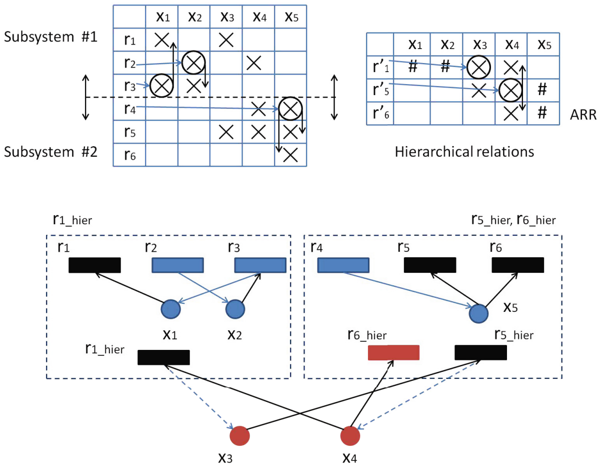


Fig. 9 Structural derivation of decentralized local and hierarchical redundancy relations

been formalized in Indra et al. [14] and Indra et al. [15]. From the perspective of a local diagnoser, while local complete matchings involve only unknown variables local to subsystems and sensed variables, relations involving shared variables can not be evaluated at that level. Such so called *hierarchical* relations are sent to the relevant supervisory level, by all local diagnosers. The supervisory layer attempts to eliminate the unknown variables at its level using these hierarchical relations and arrive at a consistency check if possible. It has been shown in Chanthery et al. [16], that such a decentralized diagnoser is equivalent from the point of view of diagnosability to a centralized diagnoser even if the choice of local matchings differ from the ones used in the global case. This has been demonstrated by showing that the same set of ARRs will be available in both the decentralized and centralized diagnosers from a structural perspective.

4 Application to the Cassini Attitude Control System: A Qualitative Comparison

We present in this section a qualitative comparison between the conventional and ARR based diagnosers in terms of the design and development effort. The application of decentralized ARR based diagnosis to the Cassini AACS is used to facilitate this comparison.

The community developing a class of techniques usually tends to take a relatively narrow view encompassing only the quantitative technical benefits of the methods. However, the decision of whether to use a novel technique for an actual spacecraft and mission is determined by a much broader costs, benefit and risk analysis. It is these *net value* considerations which often serve as bottlenecks in the adoption of new techniques such as MBD.

The challenge of comparing design and development methods in terms of their net value arises from the subjective nature of the considerations involved. However some traction can be obtained by structuring the discussion around the key factors which influence the effort involved in *diagnoser design* and *verification and validation*. Therefore the following discussion will be structured around the following two factors :

- The models used for diagnoser design
- The diagnoser design process

4.1 Models Used for Diagnoser Design

Attempting to unravel the influences and factors involved in the design procedure, we proceed by first discussing the *inputs* to the process - the *models used for diagnoser design*.

A model is typically a set of instructions, equations or constraints which encapsulate knowledge about the expected behavior of a system. Models are abstractions of reality, with a limited range of validity. Expected behavior is always modeled at a certain level of granuality, and in a certain framework.

However in a more general sense, any knowledge about the expected behavior of a system can be considered an implicit *model* of the system. A diagnosis results from reasoning about the expected behavior of a system in the form of a model. However as a model is always an approximate description of the behavior of a system, it has to be made to fit and then validated with real data. The tunable parameters allow the model to be adjusted to fit data from the actual system. A critical distinction therefore needs to be made between the *model structure* and the *model parameters*. To account for the unmodeled dynamics, i.e. behavior not accounted for by the model, *thresholds* are used. This is the case for example with noise and disturbances.

So the distinction between model based diagnosis techniques and conventional rule based methods does not lie in the *presence* of a model, but rather in the utilization of *explicit* models with sophisticated structures. Traditional rule based diagnosis techniques such as thresholding and state-space monitors use very simple model structures, and then rely on model parameters and thresholds to achieve satisfactory response to actual behaviour.

Modeling is always performed for a certain *purpose*, which dictates the aspects which need to be modeled, and also the required granuality. Different models are required for example for simulation and for controller design. Given the considerable effort involved in modeling, keeping modeling costs down is a driving factor when considering the use of new techniques. The conventional error monitor based approach uses the *expected* behavior of the error signal encapsulated in the 'state-space' representation as a simple model as seen in figure 10. The *qualitative* status - expected/unacceptable/tolerable of the control loop is determined based on the behavior of the error signal and its derivative. A fault on any of the components in the control loop can affect the error signal, and consequently the monitors.

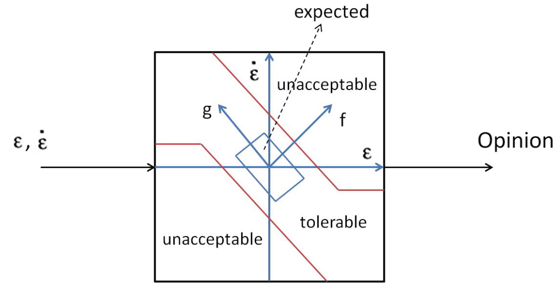


Fig. 10 Regions on the 'State Space' plane model the behaviour of the error signal - in effect modeling effort is the setting of the parameters

What about the models used for diagnoser design with the decentralized ARR based approach presented in this thesis ? The structural model utilized contains information about the constraints and variables involved in the system. An example of such a structural model can be seen in figure 11. While these models are more sophisticated than the simple 'state-space' models, the information encapsulated in them is *conceptually* the same as that contained in control and simulation models of the AOCS as seen in the constraints and variables of table 1. While control and simulation models include the actual analytical expressions of the constraints, structural models represent the same information at a more abstract level. It is possible in principle to extract structural information from the control and simulation models - which are created during the normal engineering process.

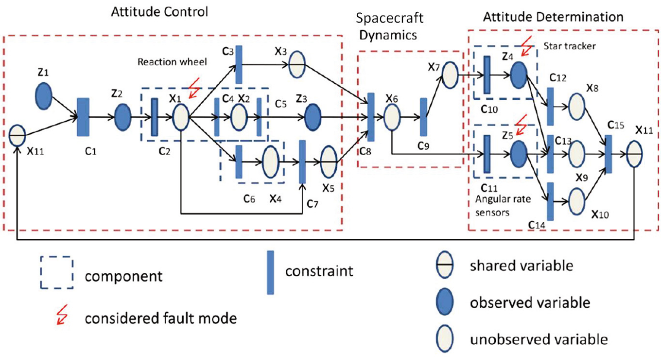


Fig. 11 An example of the structural models used for deriving analytical redundancy relations - the model *structure* plays a much greater role

Table 1 Example of constraints and variables of the structural model

Constraints and Variables	Subsystem	Description
$C_{control}/C_1$	ACS	Control algorithm
C_{RW1}/C_2	ACS	Reaction wheel motor dynamics
C_{RW2}/C_4	ACS	Reaction wheel flywheel dynamics
C_{RW3}/C_3	ACS	Reaction wheel angular momentum integration
C_{dyn}/C_8	DYN (ADS)	Satellite dynamic equations of motion
C_{kin}/C_9	DYN (ADS)	Satellite kinematic equations of motion
C_{RS}/C_{11}	ADS	Rate sensors
C_{VS}/C_{10}	ADS	Vector sensors
C_{est1}/C_{12}	ADS	State estimation with vector sensor alone
$\dot{h}_w/x_1$	ACS	Derivative of flywheel angular momentum
h_w/x_3	ACS	Flywheel angular momentum
ω_w/x_2	ACS	Flywheel angular speed
T_m/x_4	ACS	Magnetic torque
$\mathscr{X}_{ref}/z_1$	ACS	Reference value of state vector
T_c/z_2	ACS	Reaction wheel control torques
$\hat{\omega}_w/z_3$	ACS	Sensed value of reaction wheel flywheel angular speed

In conclusion, how do the two approaches compare in terms of the model used for diagnoser design ? The model *structure* in the case of the ARR based approach is more sophisticated, but contains the same information as control and simulation models. In the case of the conventional error monitor based approach, the model *structure* is very simple, being defined as regions on a plane. Much more of the behavioral information is contained rather in the model parameters - for example the parameters delimiting the regions considered 'normal', 'tolerable' or 'unacceptable'. This is a good example of "behavioral information being spread over parameter sets" as described by Rasmussen [1].

4.2 Diagnoser Design Process

Having discussed the *input* to the diagnoser design process, we consider now the design procedure itself. What constitutes diagnoser design ? We define the design process here as the *derivation of the structure* of the monitors which constitute the diagnoser and then the *setting of the diagnoser parameters* to achieve *optimal* diagnosis. An optimal diagnosis for a given diagnoser would achieve the best possible performance in terms of the considered *quantitative metrics*. Some examples of these quantitative metrics are detection time performance, false alarm rates and missed detection rates.

A simulation of the system, with realistic noise and disturbance models is used to tune the diagnoser, with the injection of realistic faults. The faults to be considered would result from engineering analysis such as FMECA and FTA.

How do the two approaches compare ? We contrast first the derivation of the structure of the diagnoser, and then the setting of the parameters.

In the case of the ARR based diagnoser, the structural model is utilized as input to an algorithm which identifies the monitorable structural redundancies present in the system, with the possiblity of focusing on a set of interesting faults. Then an automatic derivation of the analytical expressions of the residual generators is possible utilizing for example the algorithm proposed in Svard & Nyberg [13].

In contrast, as the structure of the conventional error monitors is the same for the different components in the loop and various faults, the diagnoser design effort for these monitors consists largely not in the derivation of monitor structure but in parameter tuning which is discussed below. The thruster leackage monitors of the Cassini AACS fault protection are conceptually the same as ARRs. But they were used only because conventional error monitors were not able to satisfy requirements and their derivation was not an automated process.

In the diagnoser derivation phase therefore, the possibility of *systematic, integrated* design with the decentralized ARR based method provides a significant improvement over the conventional design approach as utilized for the Cassini AACS FPS which consists of a *patchwork* of techniques.

And how about diagnoser parameter settings ? The setting of diagnoser parameters aims to optimize (and trade off between) FDI performance and robustness for a given diagnoser structure. Thresholds, counters and flags are examples of diagnostic system parameters. The effort involved in tuning the diagnostic system is strongly related to the clarity of the physical relation between the parameters to tune and the underlying properties of the system.

The first difference is in terms of the degree and nature of the role of diagnoser parameters. The extent of the role of diagnoser parameters is *inversly* proportional to the sophistication of the model structure utilized for diagnoser design. Due to the very simple model structure utilized in the conventional design approach, fitting the diagnoser behavior to data from the system relies to a large degree on the model parameters. The use of an explicit and relatively sophisticated model in ARR based approaches implies less reliance on parameters.

The second contrast is caused by the fact that the fault sensitivities of the residual generators in the ARR based diagnoser are structurally decoupled and computed in the design phase. The different techniques utilized in the conventional approach could lead to a fault propogating and triggering monitors at different levels and locations. Studies such as FMECA provide a guide to work out the activation rules in this case, followed by simulation runs with fault injection. Therefore, in the decentralized ARR based approach considerable design effort is shifted from the simulation to the design phase with the activation rules automatically derivable from the fault sensitivities of the ARR based fault monitors. We can conclude that the presented approach leads to diagnosers which are much more *transparent* and therefore easier to tune compared to the conventional methods.

5 Conclusion

The conventional techniques used to design the diagnosis elements of spacecraft fault protection systems suffer from various issues, severely restricting the scalability of such methods as space systems increase in complexity. These issues are illustrated using the example of the fault protection functionality of the Cassini attitude and articulation control subsystem. We then present a decentralized analytical redundancy relation based diagnosis architecture which can address some of them. The application of this architecture is contrasted to the diagnosis elements of the conventional Cassini FPS. The comparison is in terms of qualitative metrics such as diagnoser design effort and system structure. Discussing such qualitative factors is essential as it is ultimately these issues which have restricted the application of model based diagnosis techniques for space systems previously. The benefits of the proposed approach are demonstrated. The decentralized diagnoser enables the deployment of varying levels of diagnosability, which is not possible with a monolithic ARR based diagnoser. In future work we are focusing on possibilities related to optimizing the decentralized diagnoser structure and splitting such decentralized ARR based diagnosers between the space and ground segments.

Acknowledgements. The work discussed in this paper is funded by Thales Alenia Space, Cannes and the Centre National d'Études Spatiales. We wish to thank Xavier Olive of TAS and Raymond Sougmagne of CNES for their advice.

References

1. Rasmussen, R.: GNC fault protection fundamentals. In: Proceeding of the American Astronautical Society 31st Annual AAS Guidance and Control Conference, Breckenridge, Colorado, USA (2008)
2. Kurien, J., R-Moreno, M.D.: Costs and benefits of model-based diagnosis. In: Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, Montana, USA (March 2008)
3. Patton, R.J.: Fault detection and diagnosis in aerospace systems using analytical redundancy. Computing Control Engineering Journal 2(3), 127–136 (1991)

4. Zimpfer, D.J.: Flight Control Health Management. In: System Health Management with Aerospace Applications, pp. 483–495. John Wiley and Sons, United Kingdom (2011)
5. Krysander, M., Åslund, J., Nyberg, M.: An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 38(1) (2008)
6. Krysander, M., Åslund, J., Frisk, E.: A structural algorithm for finding testable sub-models and multiple fault isolability analysis. In: Proceedings of the 21st International Workshop on Principles of Diagnosis, DX-10 (2010)
7. Cordier, M.-O., Dague, P., Levy, F., Montmain, J., Staroswiecki, M., Trave-Massuyes, L.: Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(5), 2163–2177 (2004)
8. Slonski, J.P.: System fault protection design for the Cassini spacecraft, Rapport technique UW-CS-TR-1481, Jet Propulsion Laboratory, California Institute of Technology (October 1995)
9. Brown, G.M., Bernard, D.E., Rasmussen, R.D.: Attitude and articulation control for the Cassini spacecraft: a fault tolerance overview. In: Proceedings of the 14th Digital Avionics Systems Conference (DASC), pp. 184–192 (November 1995)
10. Lee, A.Y.: A model-based thruster leakage monitor for the Cassini Spacecraft. Journal of Guidance, Control and Dynamics (1998)
11. Macala, G.: A state-space fault monitor architecture and its application to the Cassini spacecraft. In: Proceedings of the IEEE American Control Conference, Philadelphia, USA (1998)
12. Blanke, M., Kinnaert, M., Lunze, J.: Diagnosis and Fault-tolerant Control. Springer (2006)
13. Svard, C., Nyberg, M.: Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. Trans. Sys. Man Cyber. Part A 40(6), 1310–1328 (2010)
14. Indra, S., Trave-Massuyes, L., Chanthery, E.: A decentralized FDI scheme for spacecraft: Bridging the gap between model based FDI research & practice. In: Proceedings of the 4th European Conference for Aerospace Sciences (EUCASS), St. Petersburg, Russia (2011)
15. Indra, S., Trave-Massuyes, L., Chanthery, E.: Decentralized diagnosis with isolation on request for spacecraft. In: Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess), Mexico City (2012)
16. Chanthery, E., Indra, S., Trave-Massuyes, L.: The equivalence of global and decentralized ARRs computation. LAAS-CNRS Technical Report No. 11094 (March 2011)

Flight Test Oriented Autopilot Design for Improved Aerodynamic Parameter Identification

Matthias Krings, Karsten Henning, and Frank Thielecke

Abstract. In order to reduce development costs and time, model-based design is widely introduced in the industry leading to a strong need for verified high-fidelity simulation models. An inevitable, but challenging process step to obtain such simulation models for GNC-applications is the aerodynamic parameter identification on the basis of real flight test data. The identification process requires distinct excitation maneuvers in order to constrain the design space to a subset of model parameters reducing the complexity of the identification problem and the correlation within the overall parameter set. Typically, manually flown excitation maneuvers are not exact and fully reproducible concerning the requirements and therefore the amount of rejected data points is significant. In case of remotely piloted aircraft systems, the decoupling of the aircraft and the ground pilot in charge leads to an even less sensitive maneuver control, a further reduced disturbance suppression and even greater difficulties in meeting the initialization requirements. This scenario calls for an automation of aerodynamic parameter identification related flight tests. A practical approach to a flight test oriented autopilot for improved aerodynamic parameter identification is proposed within this paper. The requirements for identification excitation maneuvers and the corresponding design of the autopilot are emphasized and flight test results are presented.

1 Introduction

Increasing automation of aircraft systems introduces a wide variety of complex issues regarding novel system concepts and technologies of prospective aircraft. In order to reduce development costs and time of such technologies, model-based design is widely introduced in the industry leading to a strong need for verified high-fidelity

Matthias Krings · Karsten Henning · Frank Thielecke
Institute of Aircraft Systems Engineering,
Hamburg University of Technology, Nesspriel 5, 21129 Hamburg
e-mail: {matthias.krings,karsten.henning,frank.thielecke}@tuhh.de

simulation models. Especially the development of advanced flight control and envelope protection schemes [1, 2] as well as the development of methods for system diagnosis and monitoring, e.g. loads observer for structural loads analysis and monitoring [3, 4, 5, 6], call for qualified flight mechanical models for evaluation and validation at an early design stage.

Although there is a strong trend towards numerical determination of model parameters, e.g. CAD, CFD, etc., system identification on the basis of real test data is still inevitable, even though it is only for validation of numerical findings. Due to the well known structure of flight mechanical simulation models the identification is often narrowed to the aerodynamic parameters, but can be extended to an identification of mass properties and actuator dynamics. Nevertheless, the effort of identifying the plant properties disproportionally increases with the number of parameters to be considered and with the data quality required. Therefore, distinct excitation maneuvers are required in order to constrain the design space to a subset of parameters reducing the complexity of the identification problem and the correlation within the overall parameter set [7, 8].

In the context of an identification of the aircraft's aerodynamic properties, the definition of these maneuvers shall aim on a separation of longitudinal and lateral motion, on a specific magnitude and timing of the command inputs and on an initialization at a predefined point within the flight envelope. Typically, these manually flown excitation maneuvers are not exact and fully reproducible concerning these requirements and therefore the amount of rejected data points is significant [5, 6]. This problem is further exacerbated by identifying the aerodynamic parameters of a remotely piloted aircraft system. The decoupling of the aircraft and the ground pilot in charge leads to a less sensitive maneuver control, a reduced disturbance suppression and difficulties in meeting the initialization requirements. This scenario calls for an automation of aerodynamic parameter identification related flight tests.

A practical approach to a flight test oriented autopilot for improved aerodynamic parameter identification is therefore suggested within this paper, which is organized as follows. First, a general description of the system identification process and a specification of common maneuvers for identification of aerodynamic parameters are given in Section 2, followed by the flight test oriented autopilot in Section 3. In Section 4 an application example and related flight test results are presented.

2 System Identification and Maneuvers

System identification represents a process of determining a model structure and related model parameter of a dynamic system with known system excitation and response. This general approach is depicted in Fig. 1 and is denoted as *Quad-M* process [7]. Herein, the four elements: maneuver design, measurement accuracy, method and model definition are the key enablers to identification results with high quality and reliability.

Among other a gray box approach is chosen defining a physically motivated model structure of the flight dynamics. The classical representation, which can be

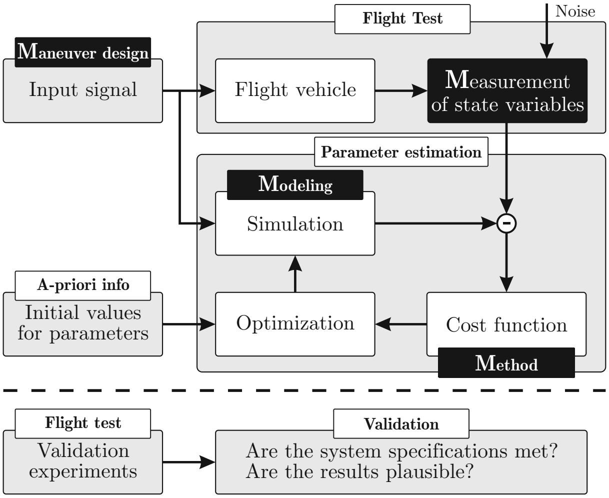


Fig. 1 General process flow (Quad-M process acc. to Ref. [7])

found in Ref. [9], comprises the 6DOF equation of motion and a polynomial representation of the aerodynamic properties. Actuator dynamics might be also taken into account. The unknown model parameters are quantified by comparison of the model and the measured aircraft system response. This procedure, widely known as the output error method, is described in detail in several Ref. [5, 6, 7].

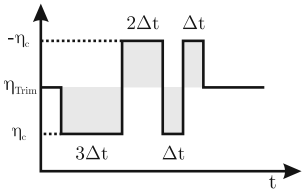
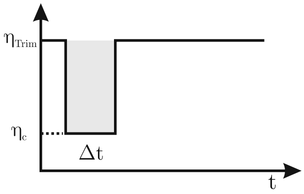
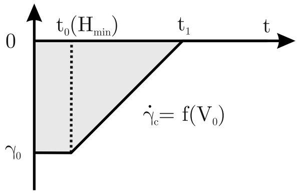
While the mass properties and the actuator dynamics are identified within laboratory test, e.g. weighing, the aerodynamic derivatives are determined on the basis of flight tests. During these flight tests the designated aircraft is excited by well-defined input signal sequences, which form the basis for an efficient, unambiguous solution of the identification problem. Therefore, the input signal sequences are subject to certain conditions, in particular [5, 6]:

- the cause variables of the aerodynamic model shall be excited,
- the excitation shall allow an identification of the parameters without correlation,
- the maneuvers shall be initialized based on a steady straight symmetric flight,
- the data basis shall contain at least one set of measurements for identification as well as one for validation and
- the maneuvers shall be repeated with variable excitation magnitude and initial flight condition in order to capture nonlinearities due to viscosity effects.

One might optimize the input signal sequences on the basis of the estimation error criterion used within the identification process [7, 10, 11]. However, the optimality of these input sequence heavily depends on the fidelity of the model and thus, in an early phase of the identification process, these excitation maneuvers are quite often not suitable [12]. Considering small and/or slow aircraft the circularity problem in defining optimal input sequences might be hard to resolve. Due to the low Reynolds numbers the preliminary numerical findings from classical CFD methods are relatively poor and hence, the initial model might be inappropriate to break the circularity problem. Therefore, the input design technique must be robust to unknown errors in the a priori model. Here, a practical approach is suggested, which comprises the well-known multistep input sequences with a low number of design parameters and a good traceability of aircraft's response.

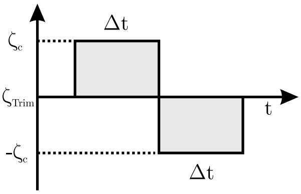
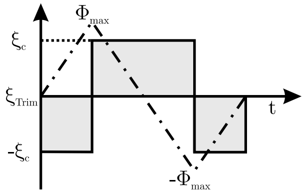
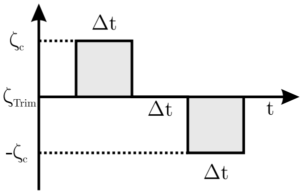
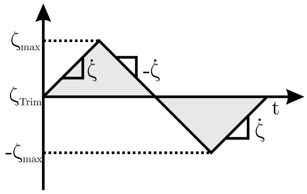
In order to identify the full parameter set of the longitudinal and lateral aerodynamics and the corresponding coupling effects seven maneuvers are recommended in Ref. [5, 6, 7]. The maneuvers and their major properties are listed in Tab. 1 and Tab. 2.

Table 1 Longitudinal aerodynamic parameter identification maneuvers

Maneuver name	Maneuver requirements	Maneuver commands
Short-period mode maneuver (SPM)	no lateral motion, constant thrust lever position	 2Δt Δt -η_c η_{Trim} η_c 3Δt Δt t
Phugoid maneuver (PM)	no lateral motion, constant thrust lever position	
Level deceleration maneuver (LD)	no lateral motion, constant thrust lever position, constant speed break settings	 t₀(H_{min}) t₁ t 0 γ̇_c= f(V₀) γ₀

The input signals as well as the corresponding aircraft response are measured with high-precision. The recorded system response is then compared to the response of the simulation model, based on the same input signal sequence. Due to the initial, insufficient knowledge of the model parameters, the response of the simulation model differs considerably from the response of the real aircraft. A cost function, defined by the principle of maximum likelihood estimation [7], and hence, the output error, are minimized by manipulation of the uncertain model parameters. Detailed information regarding implementation and specific optimization algorithms for system identification problem formulations can be found in Ref. [5] to Ref. [8].

Table 2 Lateral aerodynamic parameter identification maneuvers

Maneuver name	Maneuver requirements	Maneuver commands
Dutch roll maneuver (DR)	remaining motivators in trim position	
Bank-to-bank maneuver (BTB)	remaining motivators in trim position	
Steady heading steady sideslip maneuver (SHSS)	maintaining track by means of aileron deflection, maintaining speed by means of elevator deflection, constant thrust lever position	
Wings level sideslip maneuver (WLS)	maintaining zero bank angle by means of aileron deflection, maintaining constant speed by means of elevator deflection, constant thrust lever position	

3 Flight Test Oriented Autopilot

The flight test oriented autopilot, presented in this paper, is required to provide three different modes of operation:

- improving the flying qualities in manual flight mode,
- guiding the aircraft on the basis of predefined flight path parameters and
- performing identification maneuvers corresponding to Tab. 1 and Tab. 2.

Despite the multiple utilization and the implementation on various aircraft types the autopilot itself is required to have an easy-to-handle design with lean overhead structures. A cascaded control strategy was selected with inner flight control loops improving the flying qualities and the outer flight guidance loops regulating the aircraft rigid body motion on the basis of predefined flight path parameters. With regard to

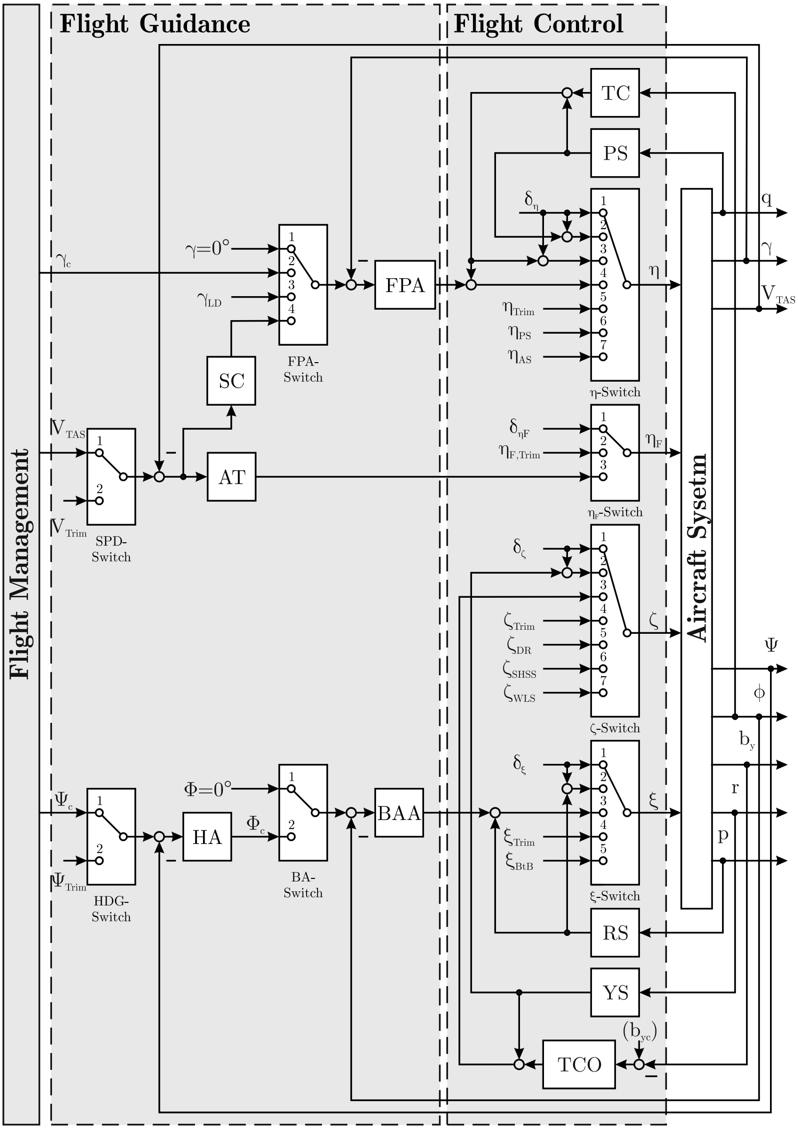


Fig. 2 Autopilot structure

the previously defined identification maneuvers, the global controller structure has a clear dissociation of longitudinal and lateral control tasks, as depicted in Fig. 2.

The inner control loops comprise not only stability augmentation systems (PS, RS, YS) but also a turn coordination (TCO) and a turn compensation (TC). The guidance part of the autopilot comprises on the one hand a flight path angle displacement autopilot (FPA) and an autothrottle function (AT) occupying the longitudinal motion, and on the other hand a bank angle displacement (BAA) and a heading autopilot (HA) occupying the lateral motion of the aircraft. An alternative in regulating the airspeed is inevitable due to the aerodynamic parameter identification requirement of a steady straight descent with constant thrust lever position. Therefore a speed controller (SC) is introduced in order to regulate the airspeed via elevator deflections. The regulators of the inner loops are chosen to be proportional only, whereas the regulators of the outer loops are designed as proportional-integral controller. In conjunction with the chosen maneuver input sequences the overall concept aims on an intuitive design of a minimal set of parameters and hence, leads to reduced complexity and good traceability throughout flight testing. A minor performance compared to more advanced controller structures and maneuver design methods is therefore to be accepted.

A distributed state machine, depicted in Fig. 3, takes on the management of the overall GNC-system not only controlling the mode switching solely but also monitoring the current flight and sensor conditions. In dependence on the available sensor data and command channels, the requested system modes are executed. A transient-free mode switching is ensured by a trimming routine capturing the current flight condition and motivator commands. The general structure of the state machine reflects the structure of the overall GNC framework. Its modular design enables an easy augmentation of the existing system with additional modes and control laws. Alongside other, already existing control laws, e.g. a *Default* mode, which keeps the motivators in neutral position, a *RC Teacher* mode, which is the safety pilot mode,

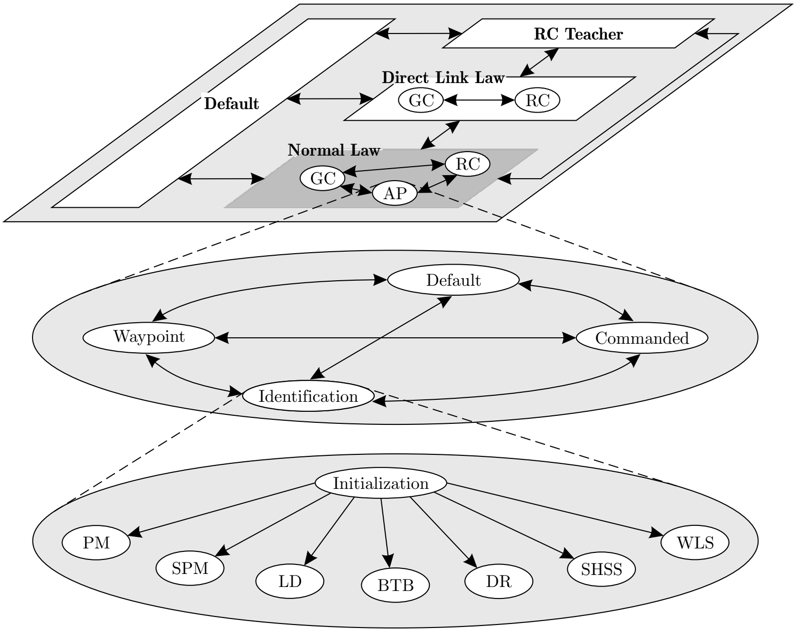


Fig. 3 Autopilot state machine

and a *Direct Link Law* mode, which is a direct feed-through of joystick (GC) and remote control (RC) commands, the *Normal Law* mode accommodates the proposed autopilot scheme. The *Normal Law* mode reflects the two stage approach of the preceding paragraph: a manual flight control part (GC and RC) and an automated flight guidance part (AP). The latter part is divided into a *Default* mode capturing and maintaining the current flight condition (airspeed, flight path angle and heading), a *Commanded* and a *Waypoint* mode providing an interface to a manual and an automated set up of the flight path parameters, and an *Identification* mode comprising the automated identification process (initialization and identification maneuvers).

4 Application Example and Comparative Flight Test Results

A first application case of the flight test oriented autopilot represents the unmanned flight test platform ULTRA-Dimona, which is the most visible aspect of the ULTRA-project founded by the TUHH-Institute of Aircraft Systems Engineering. A simulation model of this aircraft, e.g. required for software-in-the-loop and hardware-in-the-loop simulations, comprises a preliminary set of aerodynamic parameters derived by vortex lattice methods. In order to enhance the fidelity of this simulation model, identification flight test campaigns are required to determine more accurate parameter sets.

This section briefly introduces the ULTRA-project environment and the unmanned flight test platform ULTRA-Dimona. The subsequent presentation of flight test results indicates the reliability and performance of the basic autopilot functions for aerodynamic parameter identification. Due to uncertainties within the aerodynamic modeling, these results should be seen as a preliminary evaluation of the overall autopilot performance.

4.1 Unmanned Low-Cost Testing Research Aircraft at TUHH

Increasing automation of aircraft systems introduces a wide variety of complex issues regarding novel systems concepts and technologies of prospective manned aircraft. Facing these issues, the Institute of Aircraft System Engineering at Hamburg University of Technology founded the project ULTRA [1] (Unmanned Low-cost Testing Research Aircraft). Establishing flight test capabilities, the ULTRA-project conducts a representative framework for research and education adopting industry standard software and hardware components. This framework includes not only the ability of flight testing with the cost-effective, scaled, unmanned motorglider ULTRA-Dimona (see Fig. 4), but also capabilities of a laboratory infrastructure, enabling software-in-the-loop and hardware-in-the-loop simulations.

The ULTRA-Dimona is equipped with a precise navigation platform enabling highly accurate measurements of the rigid-body motion. Air data sensors as well as measurements of control surface positions and motor speed complete the

[1] ULTRA-Project: `www.fst.tu-harburg.de/ultra`

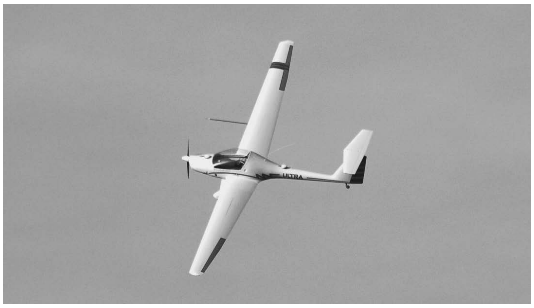


Fig. 4 Unmanned flight test platform ULTRA-Dimona

instrumentation of the unmanned flight test platform providing an ideal basis for aerodynamic parameter identification. An on-board *dSPACE* real-time system provides the capability of easily implementing, tuning and executing flight guidance and control algorithms.

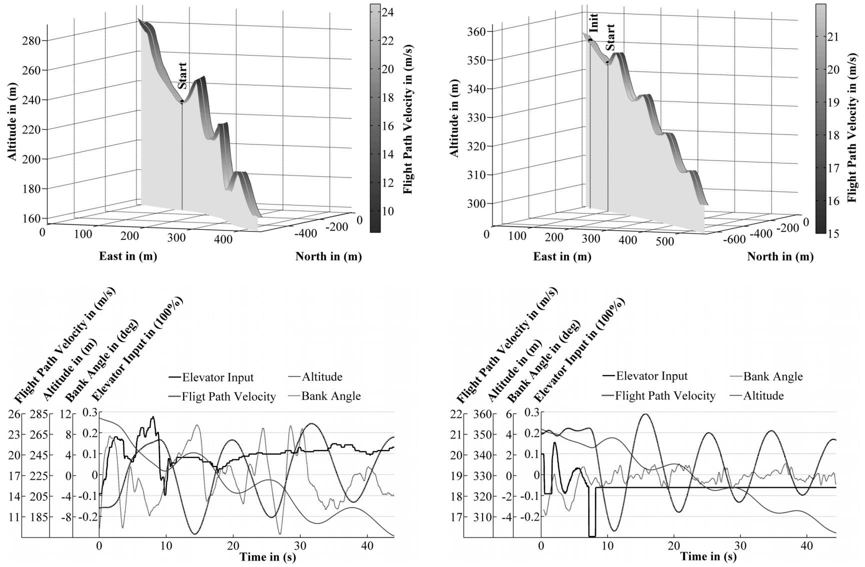
The flight test oriented autopilot approach for improved system identification was implemented on this target system and first tested within a hardware-in-the-loop simulation. The design of the controller parameters was based on a preliminary system model, where the mass properties of the flight test platform as well as the motivator dynamics were already identified throughout laboratory tests and the aerodynamic properties were determined on the basis of vortex lattice methods and slightly adapted in order to match the experience of manually controlling the real aircraft.

4.2 Flight Test Scenario for Aerodynamic Parameter Identification

A typical flight test scenario for identification of the aerodynamic system parameters is divided into manually and automatically flown parts. A safety pilot flies the aircraft to a starting position characterized by a desired airspeed, altitude and heading. Whereas the airspeed and altitude are defined by a specific flight point within the envelope of the aircraft, the heading is typically chosen as the wind direction. At this initialization position the aircraft is handed over to the autopilot. The current airspeed and heading are captured and define the steady straight descent representing the trim condition of the identification maneuver. During this flight phase the thrust lever position is kept constant and the airspeed is controlled by means of elevator deflections. The actual identification maneuver is accomplished out of this trim condition. The point of maneuver completion, a predefined altitude threshold or the range of vision respectively define the point in time, when the aircraft is handed back to the safety pilot. A subsequent repetition of this procedure might be carried out.

4.3 Flight Test Results

The autopilot was tested throughout two flight test campaigns in 2012 according to the previous procedure definition and the identification maneuvers defined in Tab. 1 and Tab. 2. Comparative identification maneuvers were flown manually. Representative for the enhancement of the flight test data quality two different identification maneuvers, the phugoid maneuver (Fig. 5) and the bank-to-bank maneuver (Fig. 6), are discussed hereinafter in detail.



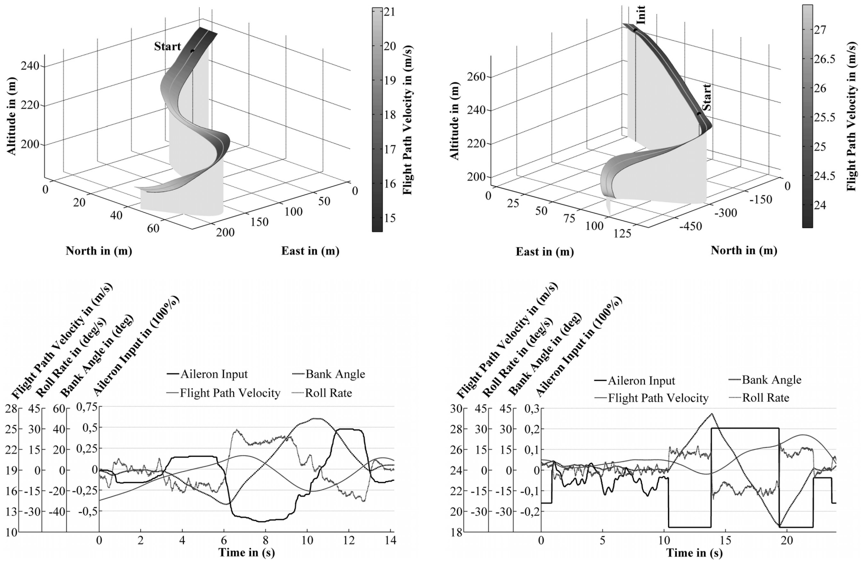
(a) manually flown

(b) automatically flown

Fig. 5 Flight trajectory and selected quantities of phugoid identification maneuver

Generally, the phugoid eigenmotion is easy to stimulate and hence, the phugoid maneuver is one of the less complex maneuvers. Nevertheless, Fig. 5 shows significant differences in manually and automatically flown phugoid maneuvers. One can emphasize two decisive points: 1) the capability of establishing a steady straight descent during the initialization phase and 2) the capability of successfully suppressing any lateral movement of the aircraft during the maneuver phase. Due to the decoupling of the aircraft and the pilot in charge, the indirect and delayed estimation of the velocity as well as the aircraft attitude leads to significant deviations compared to the automatically performed identification maneuver. In the manual case a steady straight descent was not achieved (flight path velocity increase of about $10m/s$), whereas the autopilot is capable to keep the velocity within a $1m/s$ band. Due to the use of the autopilot, the bank angle variation is minimized around factor ten compared to the manually flown maneuver.

The comparison of manually and automatically flown bank-to-bank maneuvers, depicted in Fig. 6, highlights another aspect. There is no clear indication to the pilot, like a mechanical stop at the remote control, whether the input command is kept at a predefined steady amplitude or not. As a result of this lack of information it is hardly possible to keep the roll rate constant during different legs of the maneuver. Furthermore, the probability of over excessive or too cautious maneuver inputs and therefore, the probability of gathering unusable identification data is reasonable. That this is not the case in automatically performing the identification maneuver illustrates the right-hand side of Fig. 6.



(a) manually flown (b) automatically flown

Fig. 6 Flight trajectory and selected quantities of bank-to-bank identification maneuver

Particularly with the lateral maneuvers the autopilot suppresses not only disturbances but also regulates distinct flight conditions, e.g. steady heading or steady bank angle. The performance of the flight test oriented autopilot is therefore inversely proportional to the modeling uncertainties and thus dependent on the accuracy of the controller design. Hence, an iterative process, compromising controller design and aerodynamic parameter identification, has to be carried out in order to achieve an optimal performance level. The evaluation of the automatically flown test campaigns suggests, however, that the automation of identification maneuvers will help to increase the reproducibility, reliability and accuracy of the overall aerodynamic parameter identification process at an early stage.

5 Conclusion

A flight test oriented autopilot design for improved aerodynamic parameter identification has been presented. The distinct requirements on identification maneuvering and resulting design rules have been discussed. Based on this analysis a generic autopilot scheme with easy-to-handle design and lean overhead structure was derived. Comparative flight test results have shown the reliability and functionality of the proposed autopilot scheme suggesting that the automation of identification maneuvers will help to increase the reproducibility and accuracy of the overall aerodynamic parameter identification process.

References

1. Krings, M., Thielecke, F.: An Integrated Approach to Predictive Flight Guidance and Envelope Protection. In: AIAA Guidance, Navigation and Control Conference, AIAA–2012–4712, AIAA (Augest 2012)
2. Krings, M., Thielecke, F.: Ein Integrierter Ansatz für ein Prädiktives Autopiloten- und Flugbereichssicherungssystem. In: Deutscher Luft- und Raumfahrtkongress, DGLR (September 2012)
3. Henrichfreise, H., Bensch, L., et al.: Estimation of Gusts and Structural Loads for Commercial Aircraft. In: International Forum on Aeroelasticity and Structural Dynamics. CEAS/AIAA (June 2009)
4. Haar, J., Montel, M., et al.: Flight Test Aircraft SPRINT for Loads Observer Assesment. In: International Workshop on Aircraft System Technologies. Shaker Verlag (April 2011)
5. Montel, M., Haar, J., Thielecke, F.: Modellierung und Identifikation der Längsbewegung des Flugversuchsträgers UW-9 SPRINT zur Modellbasierten Beobachtung von Strukturlasten. In: Deutscher Luft- und Raumfahrtkongress, DGLR (September 2011)
6. Montel, M., Haar, J., Thielecke, F.: Identifikation der Seitenbewegung und Validierung des Aerodynamischen Gesamtmodells für den Flugversuchsträger UW-9 SPRINT. In: Deutscher Luft- und Raumfahrtkongress, DGLR (September 2012)
7. Jategaonkar, R.V.: Flight Vehicle System Identification: A Time Domain Methodology. AIAA, Reston (2006)
8. Klein, V., Morelli, E.A.: Aircraft System Identification: Theory and Practice. AIAA, Reston (2006)
9. Brockhaus, R., Alles, W., Luckner, R.: Flugregelung. Springer-Verlag GmbH, Heidelberg (2011)
10. Mulder, J.A.: Design and Evaluation of Dynamic Flight Test Manoeuvres. Deflt University of Technology, Delft (1986)
11. Mulder, J.A., Sridhar, J.K., Breeman, J.H.: Identification of Dynamic Systems - Applications to Aircraft. AGARD Flight Test Techniques Series, Part 2, NATO Advisory Group for Aerospace Research and Development (1994)
12. Morelli, E.A.: Flight Test Validation of Optimal Input Design and Comparison to Conventional Inputs. In: AIAA Atmospheric Flight Mechanics Conference. AIAA (1997)

A Flight State Estimator that Combines Stereo-Vision, INS, and Satellite Pseudo-Ranges

Franz Andert, Jörg Dittrich, Simon Batzdorfer, Martin Becker,
Ulf Bestmann, and Peter Hecker

Abstract. This paper presents a flight state estimator which couples stereo vision, inertial (INS), and global navigation satellite system (GNSS) data. The navigation filter comes with different operation modes that allow loosely coupled GNSS/INS positioning and, for difficult conditions, improvements using visual odometry and a tighter coupling with GNSS pseudo-range (PSR) data. While camera systems are typically used as an additional relative movement sensor to enable positioning without GNSS for a certain amount of time, the PSR data filtering allows to use satellite navigation also when less than four satellites are available. This makes the filter even more robust against temporary dropouts of the full GNSS solution. The application is the navigation of unmanned aircraft in disaster scenarios which includes flights close to ground in urban or mountainous areas. The filter performance is evaluated with sensor data from unmanned helicopter flight tests where different conditions of the GNSS signal reception are simulated. It is shown that the use of PSR data improves the positioning significantly compared to the dropout when the signals of less than four satellites are available.

1 Introduction

Positioning and navigation with limited satellite reception is one of the current challenges for unmanned vehicles. Global satellite navigation has its known drawbacks such as a varying accuracy due to the satellite constellation, atmospheric errors, or

Franz Andert · Jörg Dittrich
German Aerospace Center (DLR), Institute of Flight Systems,
38108 Braunschweig, Germany
e-mail: {franz.andert,joerg.dittrich}@dlr.de

Simon Batzdorfer · Martin Becker · Ulf Bestmann · Peter Hecker
Technische Universität Braunschweig,
Institute of Flight Guidance, 38108 Braunschweig, Germany
e-mail: {s.batzdorfer,m.becker,u.bestmann,p.hecker}@tu-bs.de

possible signal interruption and reflection. Unmanned aircraft navigation becomes problematic especially in the proximity of ground objects, for example in flights through urban or natural canyons. Especially such scenarios require abilities to reduce the positioning uncertainty for safe flights without collisions. The combination of satellite navigation (GNSS), such as GPS or the upcoming Galileo system, with inertial systems (INS) is quite common. But the ability to compensate longer satellite signal dropouts depends on the accuracy and drift rates of the INS, and the available technology for small and lightweight unmanned aircraft is presently insufficient [6].

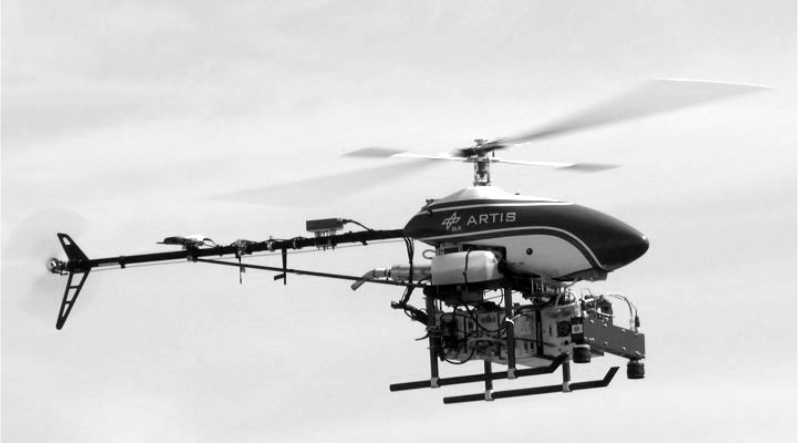


Fig. 1 DLR's 13 kg helicopter with a stereo camera, onboard image processing and GNSS/INS navigation filtering

The application context of this paper is low-altitude outdoor exploration flights in disaster scenarios with the unmanned helicopter shown in fig. 1. Since cameras are often on board these vehicles for various applications and their motion can be obtained from image sequences, it is straightforward to use them for improving the navigation solution here as well. The developed solution should be able to be run under difficult conditions and also in unknown areas, this is why the usage of a-priori knowledge from maps as proposed in [11, 14] is not suitable here. With that, only relative movements are determinable from the camera images so that the presented solution will be influenced by accumulating errors as soon as satellite navigation becomes unavailable. Contrary to many other approaches, this paper does not address scaling issues that come with monocular cameras being solved by integrating inertial measurements into the motion estimator [22, 27] which determines the scale with respect to the observed accelerations, or by using additional sensors like a barometer [1] or distance sensors like laser scanners as proposed by [25]. Instead, this approach uses a calibrated stereo camera to determine relative 3D movements and rotations. Similar approaches are used in [10, 13, 16]. However, the developed filter might be further improved by the mentioned related work so that laser scanners, monocular cameras, barometers, and many other sensors can be combined with the presented filter instead of the stereo camera.

Beside the usage of a camera as an additional sensor for the GNSS/INS navigation filter, the paper addresses the problem of partial GNSS signal dropouts. This concerns signal receptions of three or less satellites which do not give the complete position information by themselves. In these cases, a classical loosely coupled GNSS position support of drifting inertial data would fail. However, the PSR data give some hints about the position (e.g. a line in the case of three satellites) which can be matched with the information from the inertial and vision system [29]. If now the position can be recovered, the proposed navigation filter reduces the chance of positioning dropouts, especially in cluttered outdoor environments where the number of visible satellites may be often low.

2 Related Work

The idea of including visual information into a navigation filter to localize a vehicle in obstacle-prone or indoor environments is very promising. Within the last years, many technical advances have been evolved from the off-the-shelf availability of small and easily manageable aircraft (like quadrotors) and lightweight cameras and computers with sufficient performance. This section gives a brief overview of the different ideas that act as a basis for the principles developed in this paper.

2.1 Image Processing and Visual Odometry

Camera motion estimation is generally based on the motion visible in the image sequences. This requires a scenery with mostly unmoved objects within the camera's field of view and some identifiable patterns to find homologous points in the images representing the same stationary object points. Technically, this refers to identifying a sparse set of homologous points determinable by feature detection and tracking algorithms. In the stereo image case, corresponding points between two image pairs are to be identified with the advantage that the absolute scale of motion is determinable if the objects are within the usable range of stereo-based distance measurements.

One common visual odometry principle breaks the camera positioning down to a camera motion or relative orientation estimation between two images or image pairs. For the stereo case that produces 3D image features, the transformation between the two resulting point clouds can be determined by general registration algorithms [5, 8] or those optimized for stereo vision [17]. The results are camera pose updates that can be integrated into the camera trajectory. The easiest way would now be to incrementally integrate all succeeding images, but due to the large amount of little erroneous steps, the accumulation error will be rather large. Better results are to be expected when keyframes are used. For each update step, this means to look back in the image and feature history and find the oldest point cloud with enough overlap with the newest one so that a registration is still possible.

The other very common principle has its roots in photogrammetric triangulation and resectioning as well as in simultaneous localization and mapping (SLAM), see [25] for a recent overview within the aerial robotics domain. The idea is to project the observed image points of all images into the same 3-D coordinate system, and to get the current camera position by the registration between this map and the current image points. The correspondences between map and image points are usually known since all map points have been derived from the previous images. In addition to this registration, the current image points are fused to the map, which means to add new points and to reduce the statistical errors of the existing ones.

2.2 Vision-Based Navigation Filtering

Based on the different methods to measure the camera's motion, there are a lot of ways how vision data are integrated into INS or GNSS/INS navigation filters. All presented approaches have in common that they are based on a constant and known (but potentially biased) camera alignment on the vehicle. With that, camera measurements can be transformed into the vehicle-relative system, yielding a sensor that serves vehicle motion components.

An early approach for UAVs that is completely integrated into the navigation and control system is presented in [18]. A monocular, downward-looking camera is used, and image feature positions are directly integrated as measurements into an Extended Kalman Filter (EKF) which estimates the flight state. This is simple since no registration or other camera orientation estimation algorithms are applied, but also effective since it is shown that GPS dropouts of more than one minute can be handled including the stable control of an unmanned helicopter. However, the approach only integrates in-plane translational movements from the camera images and assumes a correct measurement of the ground distance.

The more complex image processing procedures such as the mentioned relative motion estimation and SLAM methods are also promising to enhance existing flight estimators. The main difference between both techniques is that relative motion consists of translational and angular movements between two images and acts more as a kind of speed sensor to be integrated into the flight state estimator [27]. Contrary to that, SLAM directly returns the camera pose based on the current and all previous images, and especially the position seems to be integrable into the navigation filter [1]. Both methods have been successfully integrated into aerial systems. However, all of these methods, including the direct use of image feature positions, are all affected by accumulating errors over time. An exception is the case where longer back-dated features are still visible in the current image so that a direct registration is possible. This means for the practical use that accumulating errors can theoretically be eliminated as long as the vehicle is in hovering mode. For example, this is confirmed by [3] where a vision-based and drift-free hover stabilization is presented.

2.3 Positioning with a Low Number of Satellites

Navigation with three satellites would be easy if a precise, i.e. atomic clock is present [23]. Since this is not an option for small unmanned aircraft and due to the cost not even for most other applications, the data from other sensors are to be brought in to get a plausible estimation of the current position. GNSS/INS navigation filters can generally handle dropouts of the GNSS signal by integrating inertial data. For example, it is shown in [28] that the drift can be significantly reduced when GNSS/INS raw data such as pseudo-ranges and phase information are tightly coupled into the state estimator. Nevertheless, this is not free from drift given incomplete satellite reception, which might still be insufficient for UAV navigation where no tactical grade inertial measurement units are used.

3 Image-Based Motion Estimation

This section describes the method to determine relative motion estimates from camera image sequences. As the principles are widely known from the related literature, the basics are only briefly introduced. Focus of this section are supplementary implementations, explained in more detail.

3.1 Determining Optical Movements

Visible disparities of a set of selected image points $\{\mathbf{p}:(x,y)^\top\}$ are the basis for camera movement determination, see fig. 2. Based on an initial point set $\{\mathbf{p}_t^L\}$ for the left camera from corner detection or the previous tracking step, a tracking over time $\{\mathbf{p}_t^L\} \rightarrow \{\mathbf{p}_{t+1}^L\}$ is performed, here with the *Lucas-Kanade* algorithm [19]. Stereo comparison for every time stamp $\{\mathbf{p}_t^L\} \rightarrow \{\mathbf{p}_t^R\}$ is done in an analogous way. Following stereoscopic math (e.g. [20]), this returns a point cloud with 3D features $\{\mathbf{q}:(x,y,z)^\top\}$ in camera coordinates for every image frame time stamp t. For further calculations, these point clouds are transformed to the vehicle coordinate frame by using an initially measured camera alignment.

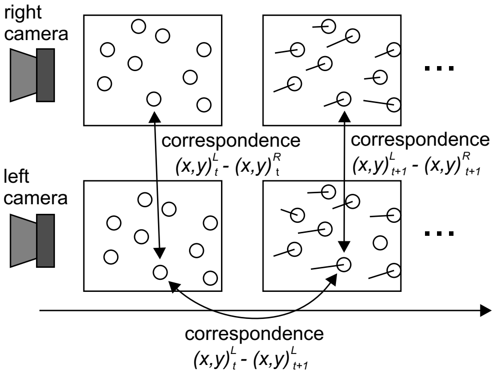
3.2 Ego-motion Estimation

Generally, the relative motion is denoted as a 4×4 transformation matrix $\mathbf{T}$ containing the 3×3 rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ in the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{t}_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix}. \tag{1}$$

Let the matrix $\mathbf{T}_{t_1:t_2}$ define the relative movements between the time stamps t_1 and t_2. From stereo image points, it is determined through the rigid transform of the point clouds $\{\mathbf{q}_{t_1}\}$ and $\{\mathbf{q}_{t_2}\}$. Several algorithms which do or do not require initial correspondences have been tested to do this job, and it turned out that the iterative

Fig. 2 Relationship between the image sequence and the optical movements. The correspondence of points is determined between different images of the left camera (temporal movement) and between the two images from the cameras (spatial disparities). The circles and the lines show the feature points and their movement vector.


right camera
left camera
correspondence
$(x,y)^L_t$ - $(x,y)^R_t$
correspondence
$(x,y)^L_{t+1}$ - $(x,y)^R_{t+1}$
correspondence
$(x,y)^L_t$ - $(x,y)^L_{t+1}$
...
...
t

closest point (ICP) algorithm with nonlinear optimization backend from the *Point Cloud Library* [24] performs best if no other hints like inertial data are given. Since the correspondence of points $[\mathbf{q}_{t_1}, \mathbf{q}_{t_2}]$ is known from feature tracking, the input point clouds are reduced to the corresponding elements that exist in both. This avoids false convergence and improves the estimation results.

The fitness of the transformation is returned by the error covariance matrix $\mathbf{Cov}(\{\mathbf{T}_{t_1:t_2}\mathbf{q}_{t_1}\}, \{\mathbf{q}_{t_2}\})$ based on point distances between the point clouds transformed to the same coordinate system. Here, the sets $\{\mathbf{q}_{t_1}\}$ and $\{\mathbf{q}_{t_2}\}$ only include the points that remain relevant for transformation estimation, i.e. outliers are not included. The matrix elements are e.g.

$$\mathrm{cov}_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_{t_1,i} - x_{t_2,i})(y_{t_1,i} - y_{t_2,i}), \tag{2}$$

the other matrix elements are calculated analogously. Since the centroids of both transformed point clouds are equal after an ICP transform, the mean distance is zero and omitted in the equation.

3.3 Using Key Frames

While classic visual odometry is based on incremental transformations $\mathbf{T}_{t-1:t}$, the usage of key frames means to estimate the transformation $\mathbf{T}_{t-\tau:t}$ between the current frame t and the oldest possible frame $t - \tau$ (i.e. τ frames older) from the image sequence. As already mentioned, this will presumably reduce the accumulating error. Transformations are determinable as long as an overlap exists between an older image and the current one. Practically, this refers to the availability of corresponding homologous points which were already detected in both images. Fortunately, this does not mean that the image sequence has to be saved. It is sufficient to store the

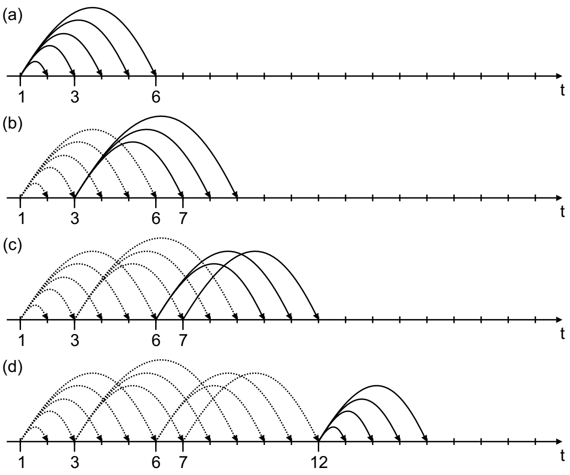


Fig. 3 Using key frames to determine camera movements. Comparing every new frame with the first one (a), jumping to the next key frames (b,c), using the last frame as key if no other key frame is available (d). The curved arrows show which frames are being compared during the sequence.

features $\{\mathbf{q}_t\}$ of the images. Because the probabilities of a successful transformation determination between the current image and two older ones with the same feature set are supposedly similar, it is further sufficient to store the features of those images where new features are added and mark them as a key frame. This results in a reduced set of key frames with only the oldest frame from each sub-sequence where the images have all the same tracked features.

Figure 3 illustrates how the key framing process works. In (a), the ideal case is shown where the current image features are compared to the features from image 1. Every time new features are added to the tracking list, a key frame (i.e. the list of features) is stored. In the shown example, key frames from the time stamps 3 and 6 are saved. This does not mean that they are immediately used for comparison with the next frames. Older key frames will still be used as long as the transformation calculation is successful.

A key frame is not valid anymore and will be deleted if the number of feature points having corresponding ones in the current frame becomes too low, or if the attempt to estimate the relative transformation fails. In these cases, the next oldest possible key frame will be used until a valid transformation returns. An example is shown in (b) where frame 3 is used from now on as the oldest valid key frame. The advantage of storing multiple key frames is shown in (c) where frame 6 is used as the key frame for the time step 10 and 11, and after it becomes invalid, another rather old frame 7 is available at time step 12. Beyond that, subfigure (d) shows the case where no valid key frame is available at time step 13 and the last frame 12 will be used as the new key from now on. Nevertheless, it remains the rather unlikely case where no transformation to older frames is determinable at all. Resulting odometry gaps are handled by the navigation filter.

3.4 Using Feedback of the Predicted Flight State

In the presented setup illustrated in fig. 6, the predicted flight state is coupled back to estimate relative motions. With the states corresponding to the images, the point clouds are transformed to geodetic coordinates, and the ego-motion estimation directly returns the geodetic movement and rotation. In addition to that, the rotation is already obtained from inertial data (i.e. both geodetic point clouds do not have any rotation to each other in the ideal case), and it is sufficient to estimate the translational movement. This can easily be done by calculating the difference of the point cloud centroids. Combined with an outlier filtering that removes corresponding points with large distances remaining after transformation, this returns the camera movement $\mathbf{t}_{t_1:t_2}$. In the results section, it is shown from recorded image data that this performs better than the estimation of all six degrees of freedom as described before. Therefore, only the translational movement is being coupled with the flight state estimation filter.

4 State Estimation

The flight state estimation follows the common principles described in [7, 9] which is a part of the navigation research at the TU Braunschweig. The state $\mathbf{x}$ is denoted as the vector

$$\mathbf{x} = (\mathbf{p}^\top, \mathbf{v}^\top, \mathbf{q}^\top, \mathbf{b}_a^\top, \mathbf{b}_\omega^\top)^\top \tag{3}$$

with WGS84 position vector $\mathbf{p}_{(3\times1)}$ (latitude ϕ, longitude λ, ellipsoidal height h), velocity vector $\mathbf{v}_{(3\times1)}$, attitude quaternion $\mathbf{q}_{(4\times1)}$, and biases of acceleration $\mathbf{b}_{a\,(3\times1)}$ and turn rates $\mathbf{b}_{\omega\,(3\times1)}$.

The Extended Kalman Filter (EKF) loop to estimate $\mathbf{x}$ contains the high-frequency time update, i.e. the prediction step with inertial data

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)\\ \mathbf{P}_k^- &= \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\boldsymbol{\Phi}_{k-1}^\top + \mathbf{Q}\end{aligned} \tag{4}$$

with the predicted state $\hat{\mathbf{x}}_k^-$ and its covariance $\mathbf{P}_k^-$ at step k based on the previous estimation and the input vector from inertial data $\mathbf{u}_k$ (see sec. 4.1). Lower-frequency GNSS and vision data are measurement vectors $\mathbf{z}_k$ (see sec. 4.2 and 4.3). If $\mathbf{z}_k$ is available, the correction step is

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^\top(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\top + \mathbf{R}_k)^{-1}\\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)\right)\\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-\end{aligned} \tag{5}$$

yielding the estimator gain matrix $\mathbf{K}_k$, the corrected estimation $\hat{\mathbf{x}}_k$ and its covariance matrix $\mathbf{P}_k$. The other symbols are: measurement matrix $\mathbf{H}_k$, nonlinear measurement function $h(\hat{\mathbf{x}}_k^-)$, transition matrix $\boldsymbol{\Phi}_k$, process noise covariance matrix $\mathbf{Q}$, and measurement noise covariance matrix $\mathbf{R}_k$. There are two different update steps based on either vision or GNSS, which are run in succession if data from both are available. This means that the vision update step yields a new state prediction, which is updated by GNSS data afterwards. Detailed explanations about the prediction and update steps are given in the next subsections.

4.1 Prediction with Strapdown Calculation

The inertial system measures three-dimensional body-fixed accelerations and turn rates. This defines the vector $\mathbf{u} = (\mathbf{a}, \omega)$. Based on this information, the earth's gravity and known initialization values for position, velocity, and attitude, the so-called strapdown calculation $f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ returns new values for every time stamp. The calculation consists of differential equations implemented in the navigation software. Basically, it integrates the accelerations to velocities and twice to positions and the turn rates to the attitude angles. Further details such as the compensation for the earth's rotation and equations are given in [7].

4.2 Update with Image Data

As already mentioned, it turned out that visual odometry performs best when the estimated flight state is coupled back to the image processing, which will then estimate only position differences between two images. Let the vector $\mathbf{t}_{t_1:t_2}$ (see eq. 1) be the estimated motion, $\mathbf{p}_{t_1}$ "plus" $\mathbf{t}_{t_1:t_2}$ would return the current position estimate. Although the cameras are triggered by the navigation clock based on inertial and GNSS data, the times t_1 and t_2 may differ slightly from the filter update time stamps. Hence, the closest time stamps t_{k_1} and t_{k_2} of filter updates k_1 and k_2 are the basis for the measurement

$$\mathbf{z}_k = \mathrm{p2w}(\mathbf{t}_{t_1:t_2} \cdot \frac{t_{k_2} - t_{k_1}}{t_2 - t_1}, \mathbf{p}_{k_1}) \tag{6}$$

of the image-based position. The function $\mathrm{p2w}(\mathbf{t}, \mathbf{p}_0)$ denotes the function which converts a Cartesian coordinate translation vector $\mathbf{t}$ relative to a fundamental point $\mathbf{p}_0$ to the geodetic system [21]. Since the values of $\mathbf{z}_k$ directly give the position, the corresponding measurement matrix is constant

$$\mathbf{H}_k = \begin{pmatrix} 1\;0\;0\;0\;\ldots\;0 \\ 0\;1\;0\;0\;\ldots\;0 \\ 0\;0\;1\;0\;\ldots\;0 \end{pmatrix} \tag{7}$$

and accordingly, $h(\hat{\mathbf{x}}_k^-)$ returns the first three values of the vector $\hat{\mathbf{x}}_k^-$.

The matrix $\mathbf{R}_k$ is taken from the covariance matrix as described in sec. 3.2. Like the motion, it is "stretched" slightly with the quotient of the different time durations. It is

$$\mathbf{R}_k = \left(\frac{t_{k_2} - t_{k_1}}{t_2 - t_1}\right)^2 \mathbf{Cov}(\{\mathbf{T}_{t_1:t_2}\mathbf{q}_{t_1}\}, \{\mathbf{q}_{t_2}\}). \tag{8}$$

4.3 Update with GNSS Data

The update step uses GPS pseudo-range data and follows the principles presented e.g. in [12, 15]. Details of this method are described in the literature, the basic approach is the following: For the i-th satellite ($i = 1,\dots,n$), the used data include the pseudo-range ρ_i, its standard deviation σ_i as well as the time errors and satellite positions from the ephemeris data. To include this into the state filter, the measurement vector $\mathbf{z}_k$ is built by the measured ranges, it is

$$\mathbf{z}_k = (z_1, z_2, \dots, z_n)_k^\top \tag{9}$$

containing the corrected pseudo-ranges from the n visible satellites. It is

$$z_i = \rho_i - c_0\left(\Delta t - t_{\mathrm{Sat},i} + \Delta t_{\mathrm{tropo},i} + \Delta t_{\mathrm{iono},i}\right) \tag{10}$$

with the measured pseudo-ranges $\rho_1, \rho_2, \dots, \rho_n$ and the time differences from clock, tropospheric, and ionospheric errors multiplied by the speed of light c_0.

The observation matrix $\mathbf{H}$ for the k-th update is the Jacobian matrix

$$\mathbf{H}_k = \begin{pmatrix} \frac{\partial\rho_1}{\partial\phi} & \frac{\partial\rho_1}{\partial\lambda} & \frac{\partial\rho_1}{\partial h} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\rho_n}{\partial\phi} & \frac{\partial\rho_n}{\partial\lambda} & \frac{\partial\rho_n}{\partial h} & 0 & \dots & 0 \end{pmatrix} \tag{11}$$

of pseudo-range derivatives with respect to the geodetic position.

To map $\hat{\mathbf{x}}_k^-$ onto a predicted measurement, the function $h(\hat{\mathbf{x}}_k^-)$ returns a vector $\hat{\mathbf{z}}_k^- = \left(\hat{z}_1^-, \hat{z}_2^-, \dots, \hat{z}_n^-\right)_k^\top$ with the predicted pseudo-ranges. It is the Euclidean distance

$$\hat{z}_i^- = \|\mathbf{p}_{\mathrm{Sat},i} - \mathrm{w2e}(\hat{\mathbf{p}}^-)\|_2 \tag{12}$$

between the cartesian earth-centered and earth-fixed (ECEF) i-th satellite position $\mathbf{p}_{\mathrm{Sat},i}$ and the result of the function $\mathrm{w2e}(\hat{\mathbf{p}}^-)$ which converts the predicted WGS84 position into the ECEF system [21].

The error covariance matrix $\mathbf{R}_k$ is defined by the standard deviations σ_i of the pseudo-ranges, it is

$$\mathbf{R}_k = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)_k. \tag{13}$$

5 Experimental Setup

Goal of this work is the implementation of a vision-aided navigation filter and its evaluation during flight test. This section gives an overview about the hardware, general software architecture, and the reference system for validation.

5.1 Flight Hardware

Testing vehicle (fig. 4) is the 13 kg helicopter ARTIS (Autonomous Rotorcraft Testbed for Intelligent Systems) of the DLR Institute of Flight Systems [2]. The navigation sensors are a ublox-6 GPS as the GNSS receiver and a custom-built IMU with two 2-axis accelerometers (Bosch SMB 225) and and three 1-axis gyros (Bosch SMG 074) including calibration and temperature compensation. Image sensors are two digital global shutter firewire cameras (AVT Marlin F131B, resolution: max. 1280×960 px, framerate: max. 30 Hz, lens/focal length: 1265 px) with a baseline of 30 cm. Synchronous image exposures are triggered with a signal based on the pulse per second output from the GPS receiver. The navigation module (fig. 5) combines GPS, INS, and navigation computer in a single box, separated from the image processing computer.

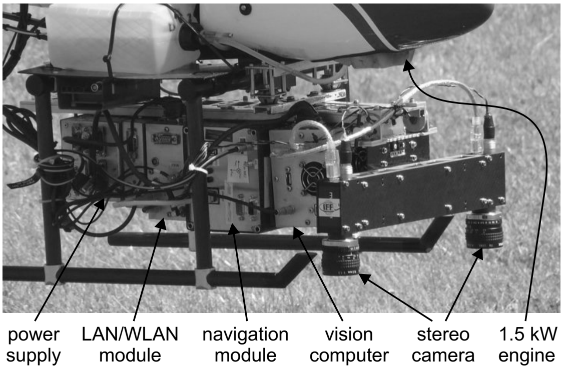


Fig. 4 ARTIS helicopter with navigation and image processing payload

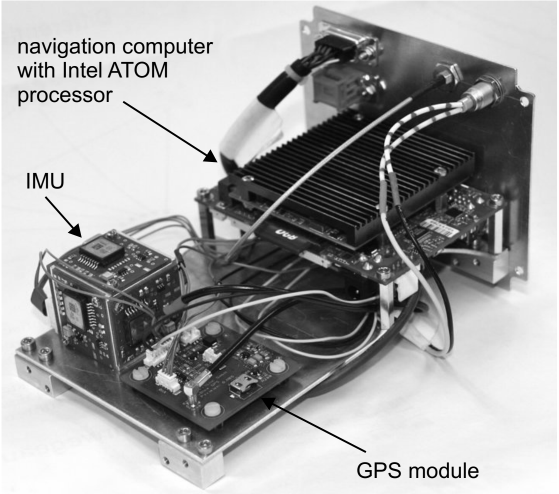


Fig. 5 The navigation module with its internal components

5.2 Processing Software Architecture

As navigation and image processing are separated by the hardware, the tightly coupled filter is based on two software frameworks with bidirectional data exchange. Fig. 6 shows the core components.

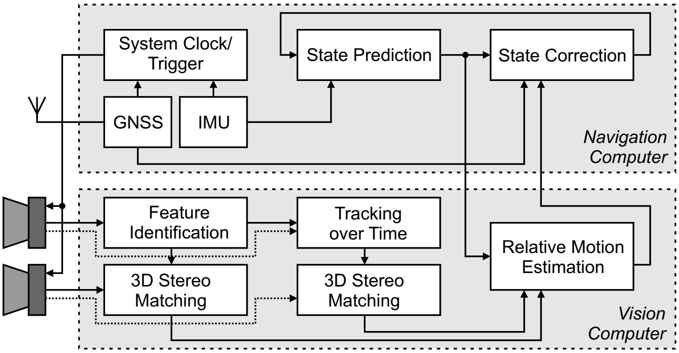


Fig. 6 Navigation and image processing software components.

The flight state estimator follows Extended Kalman Filter (EKF) principles with a state prediction based on high-frequency inertial measurements (here: 100 Hz update) and a state correction with measurements from sensors with lower update rates. These are the GPS pseudo-range (PSR) data and the camera-based relative motion. Relative motions are estimated with six degrees of freedom (DoF) onboard the image processing computer from homologous image points that are identified with a feature tracking algorithm. The relative motion estimation is additionally supported by the predicted flight state.

5.3 Reference Measuring

The computed solution with full and simulated limited GNSS reception is referenced to an augmented high-precision positioning based on the raw satellite data and post-processing corrections from state survey services [26]. The reference position has an accuracy of few centimeters when the availability of satellite data is sufficient as in the flight tests on a model aircraft flight field. As anticipation for future flights in urban environments with real signal dropouts, laser-based tracking and measuring was also established as a reference independently from GNSS signals [4].

6 Flight Testing and Evaluation

The presented methods are tested with data recorded during remotely controlled helicopter flights over a model airfield with a grass runway, some vegetation, and a small house at one side where the ground control station car is parked. The cameras are looking downward, example images are shown in fig. 7. In the following section, the development steps are tested constructively. First, it is shown whether the usage of key frames from sec. 3.3 is a useful procedure within visual odometry. Second, the visual relative movement calculation is improved with the feedback from the flight state prediction from sec. 3.4. And finally, these movements are forwarded to the state estimator from sec. 4.

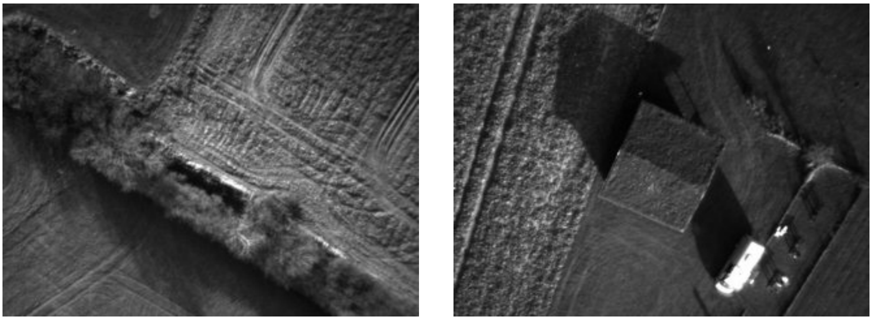


Fig. 7 Examples of the analyzed image sequence. Images of the left camera of the stereo rig at the edge of the airfield (left) and over the ground control station car next to the house (right).

6.1 Visual Odometry without and with Key Frames

In this pre-test, the relative movements $\mathbf{T}_{t_1:t_2}$ are integrated to absolute cartesian vehicle orientations $\mathbf{X}_{t_2}$ (position and rotation). Analogous to eq. 1, $\mathbf{X}$ are 4×4 matrices describing the vehicle position and attitude (direction cosine matrix) at the indexed time. This integration can be done if an initial $\mathbf{X}_0$ is available. The values of $\mathbf{X}_0$ are taken from the GNSS/INS flight state at the beginning of the image sequence. The set $\{\mathbf{X}_t : t = 1, \ldots, t_{\max}\}$ denotes now the path calculated only from accumulating image data.

In the incremental version, $\mathbf{X}_t$ is recursively

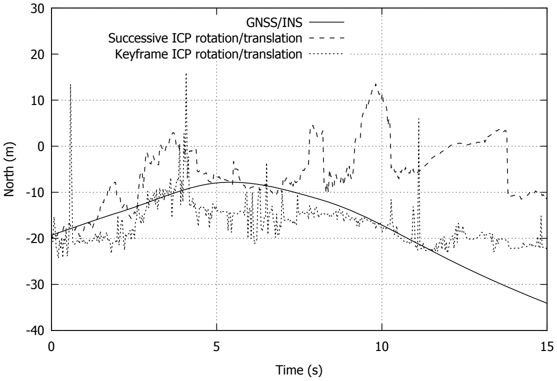
$$\begin{aligned} \mathbf{X}_t &= \mathbf{T}_t \cdot \mathbf{T}_{t-1} \cdot \ldots \cdot \mathbf{T}_1 \cdot \mathbf{X}_0 \\ \mathbf{X}_t &= \mathbf{T}_t \cdot \mathbf{X}_{t-1}, \end{aligned} \tag{14}$$

which means to accumulate a number of t relative movement steps. Contrary to that, the version with key frames calculates

$$\mathbf{X}_t = \mathbf{T}_{t-\tau:t} \cdot \mathbf{X}_{t-\tau}. \tag{15}$$

Depending of the size of $\tau - 1$ skipped frames within every accumulation step, much fewer relative movements have to be accumulated. This means theoretically, that the accumulation error is reduced and theoretically dissolved when the first frame can

Fig. 8 Trajectory $\mathbf{X}_t$ by accumulating relative orientations from the Iterative Closest Point (ICP) algorithm, x-position coordinate. The graph shows a successive accumulation of 150 image relations within 15 seconds. The other graph shows the accumulation with key frames and fewer steps. Reference (solid): GNSS/INS position.



be kept as a key frame. This is the case when the current image is still overlapping with the first one, for example in hovering flight. However, the benefit from key frames should be decreasing when flying faster.

As a result from a recorded image sequence, fig. 8 shows the integrated positions calculated from successive frames and with the use of key frames. Several trials have suggested that the optical flow of 150 to 250 features should be measured by the tracker for suitable results. The vision-based position starts without relative error to the GNSS/INS path and drifts due to translational and rotational errors with every update. The differences between both curves can be interpreted as follows: In the successive accumulation, the relative steps are quite small, and thus the errors (e.g. large jump at 13.5 s) are directly transferred to the next step (dashed curve). Contrary to that, the relative steps are larger when referring to older key frames (short-dashed curve) and since the current step can cause a jump from one key frame to the next, some more fluctuation is transferred to the resulting positions. A positive effect is that erroneous steps are not integrated in every case, which is visible through the (removable) peaks (at 1 s, 4 s, 11 s) in the position coordinate. And the overall accumulation error is as expected lower than with successive relative accumulation. However, both methods accumulate errors so that a camera should not be used as the only sensor for navigation.

6.2 Visual Position Estimation with State Feedback

This next evaluation takes a longer image sequence and couples the flight state back to the visual motion estimation. Here, the GNSS/INS flight state is coupled back so that the resulting absolute position by visual relative measurements is not accumulating errors with regard to the GNSS/INS path. (Of course it is eventually drifting when the state estimator drifts in cases where no satellite signals are available.) This test only uses the key frame version of visual relative measurements. The relative transformation $\mathbf{T}_{t-\tau:t}$ is calculated by the transformed image point clouds into the geodetic system, i.e. the sets $\{\mathbf{q}^g_{t-\tau}\}$ and $\{\mathbf{q}^g_t\}$ with

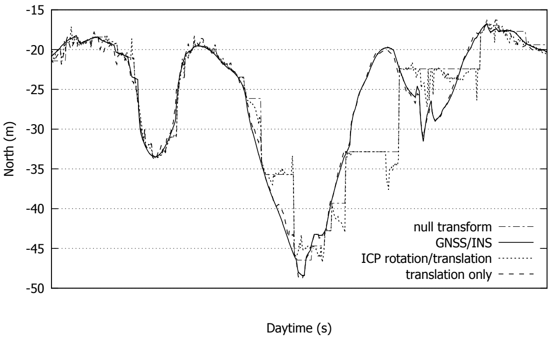


Fig. 9 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, x-coordinate. The plot includes visual 6-DoF estimation (ICP), 3-DoF estimation (translation only), null transform, and GNSS/INS reference.

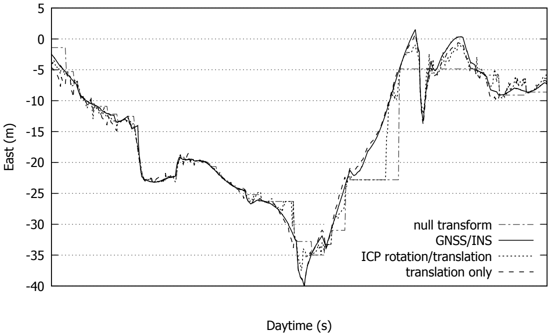


Fig. 10 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, y-coordinate.

$$\begin{aligned} \mathbf{q}^g_{t-\tau} &= \hat{\mathbf{X}}_{t-\tau} \cdot \mathbf{q}_{t-\tau}, \text{and} \\ \mathbf{q}^g_t &= \hat{\mathbf{X}}^-_t \cdot \mathbf{q}_t \end{aligned} \tag{16}$$

by using the transformation matrices containing the position and rotation of the corrected old state $\hat{\mathbf{X}}_{t-\tau}$ or the predicted current state $\hat{\mathbf{X}}^-_t$.

Results from flight tests are shown in the figures 9–11. The plots show excerpts from a 15-minute flight. It is examined whether a visual estimation of $\mathbf{T}_{t-\tau:t}$ with only three translational degrees of freedom (sec. 3.4) with the help of inertial rotations gives additional performance compared to the previous visual estimation of $\mathbf{T}_{t-\tau:t}$ with the full six degrees of freedom by using the ICP algorithm. Contrary to the vision-only method above, only 50 to 100 tracked features are required to get the viable results that are presented here. This decreases the computation time for image processing. For a better visualization where the state is fed back, a curve based on null transforms $\mathbf{T}_{t-\tau:t} = \mathbf{I}_{4\times4}$ is drawn into the plots. This results in horizontal lines with jumps to the GNSS/INS comparison plot every time a new key frame is used.

In fact, the figures do not prove a drift-free state estimation with only the visual odometry, but they indicate its behavior when used in combination with state feedback. The curves can be interpreted as the input of the vision-based positions into the navigation filter. The plots show that the visual 3-DoF estimation performs significantly better than the 6-DoF estimation, especially when the time between two key frames is large such as in the time between 48880 s and 48885 s, or 48885 s and

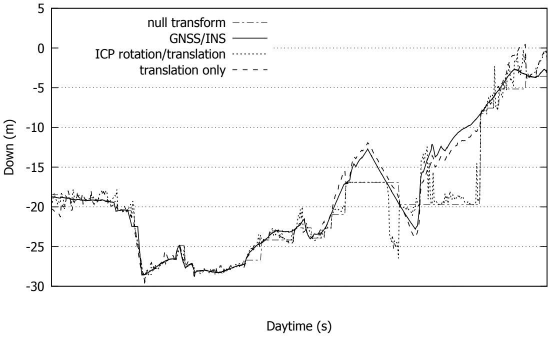


Fig. 11 Visual trajectory $\mathbf{X}_t$ by back-coupling the estimated state, z-coordinate.

48893 s. The higher performance of the 3-DoF estimation (translation only) is visible for all three axes. The most obvious reasons are scene geometry and texturing, and the rather narrow field of view of the cameras: The aircraft flies over a mostly planar scenery while looking downward, and therefore it is hard to distinguish between forward movements and pitch rotations, or between sideward movements and roll rotations from vision only. Here, the INS-based image feature point cloud transformation helps, and the 3-DoF estimation does not have to deal with these ambiguities. Beside that, it was observed that the highest uncertainty and errors are with the visual z-direction. This is mainly caused by the nature of stereo geometry with increasing range errors.

6.3 Integration of Visual Movements into the Flight State Estimator

The forward integration of stereo-based movements into the Kalman filter closes the loop between the navigation and image processing components. The following results are again based on the recorded data presented in the previous section. Here, the navigation EKF directly combines GPS pseudo-ranges, inertial data, and the visual 3-DoF movements that were improved with state predictions. With full satellite data reception, it was observed that the GNSS/INS trajectory is only slightly changed when visual data are included. This is due to the state estimator that weights satellite data with a higher confidence.

To show the filter capabilities in the case of satellite signal dropouts, the path is calculated again from the raw data but with partly and fully disabled satellite data. Results are shown in the figures 12 to 14 where a visibility of less than four satellites is simulated. The evaluation is based on two (seconds 48850 to 48870) or three satellites (seconds 48870 to 48900). A further distinction between the reception of zero and incomplete satellite signals allows to show the effects of the GNSS raw data handling. Thus, the presented results give an idea how the filter handles incomplete satellite constellations.

First of all, the results show that the inertial solution (solid curves) has a high drift rate in all axes, its quadratic behavior is especially visible for the x- and the

Fig. 12 Trajectory $\mathbf{X}_t$ from full flight state estimation, x-coordinate. GNSS signal dropout from 48850 s to 48900 s. Data from full dropout (0 satellites) with GNSS/INS only, full and partial dropout with visual 3-DoF estimation, and complete data availability.

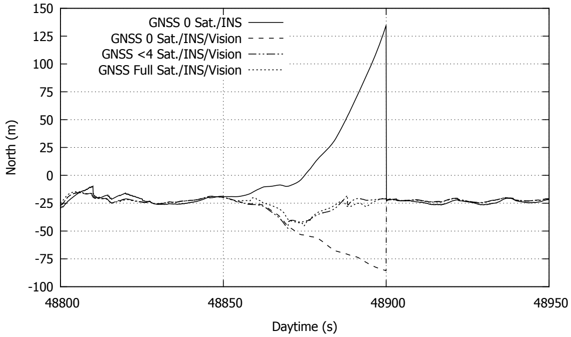


Fig. 13 Trajectory $\mathbf{X}_t$ from full flight state estimation, y-coordinate

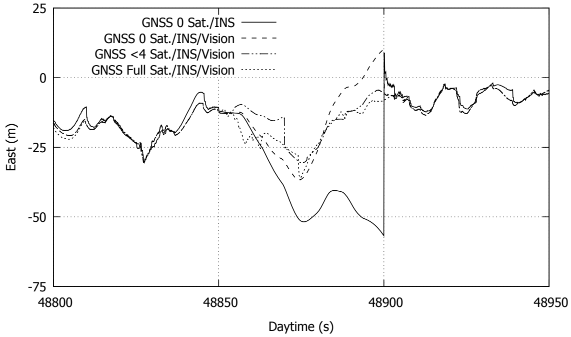
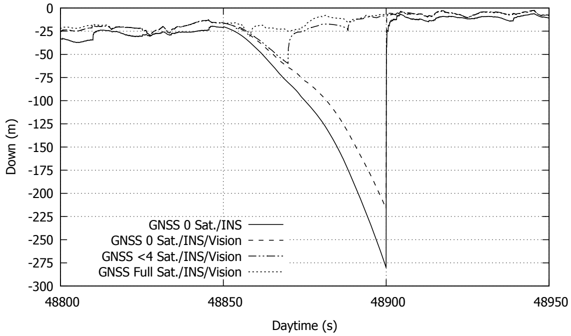


Fig. 14 Trajectory $\mathbf{X}_t$ from full flight state estimation, z-coordinate



z-coordinates. After the 50 second dropout, errors of roughly 150 m (north), 50 m (east), and 250 m (down value) are observed with respect to a full satellite constellation (dotted curve). If now the visual odometry is switched on (dashed curves), the errors can be reduced, but still remain large at 50 m (north), 25 m (east), and 200 m (down value). Especially the down value has the largest error, probably due to the nature of stereo cameras where the highest uncertainty is with the camera z-coordinate along the optical axis.

For partial satellite dropouts, large improvements are produced by the usage of GNSS pseudo-range data. The dash-dot curves show the estimated position in such a case. If at least two satellites are visible (seconds 48850 to 48870), the error is slightly reduced but still too large to navigate correctly. If three satellites are available (seconds 48870 to 48900), the error can be significantly reduced compared to the plot without satellites (dashed curve), and no typical drift or other error accumulations are observed. With that, both visual odometry and satellite pseudo-range data evaluation improve the state estimation under these conditions. In these tests, at least three satellites are required to get a suitable state estimation.

7 Conclusion

Topic of this paper is the improvement of unmanned aircraft state estimation with the help of cameras. Satellite navigation and inertial data fusion is quite common, but comes with a lot of disadvantages like satellite signal errors especially in the proximity of obstacles and the high drift rates of small inertial measurement units which do not allow long integration times. However, this is the basis for the presented developments which use an extended Kalman filter solution for data fusion. The paper analyzes a variety of approaches how to measure the ego-motion with cameras by processing the image sequences. The presented option uses a stereo camera to handle the metric scaling issue that comes with monocular vision and computes the visible 3D movements within a sequence of such image pairs. The ego-motion can now be extracted from the characteristics of such visible movements, but it is often hard to decide whether the movements visible in the images are caused by the camera's rotation or by its displacement. To solve this ambiguity, the presented approach estimates the rotation by the inertial measurements so that the image processing part does only have to estimate the translational movement. It is shown with the tests that this easier estimation of only three degrees of freedom is significantly increasing the overall performance.

Similar to inertial data, visual information will accumulate errors over time because they are relative between two time stamps. Such errors can be reduced when the current image is still overlapping with a rather old one that has been stored as a key frame for movement determination, meaning a theoretical elimination of the accumulation error during hovering mode. In the presented tests, this is drifting also at slower movements, and it has to be tested whether this will drift in hover as well. However, the combination of visual and inertial information is going to reduce the accumulation error when no satellite data is available, being an option for navigation with small unmanned vehicles.

Another aspect affects the absolute positioning with satellite data whose errors and dropout times should be reduced. Since many robotic applications directly use the position or velocity outputs from the receiver, no information is given when less than four satellites are visible. On the other hand, the known coupling methods of satellite pseudo-range with inertial data show that the positioning accuracy can be highly improved in cases where no full satellite solution is possible. Based on that, the presented filter also integrates the range data directly instead of using the pre-computed positions, being able to improve the overall performance in partly occluded areas.

Acknowledgements. The investigations are carried out within the project *Navigation zur Exploration von tieffliegenden UAV in Katastrophenszenarien* (Navigation for exploration with low-flying UAVs in disaster scenarios, NExt UAV). The joint project is funded by the German Federal Ministry of Economics and Technology (BMWi) and administered by the space management of the German Aerospace Center (DLR) Bonn (support codes 50 NA 1002 and 50 NA 1003).

References

1. Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R.: Onboard imu and monocular vision based control for MAVs in unknown in- and outdoor environments. In: IEEE International Conference on Robotics and Automation, pp. 3056–3063 (2012)
2. Adolf, F., Andert, F., Lorenz, S., Goormann, L., Dittrich, J.: An unmanned helicopter for autonomous flights in urban terrain. In: Kröger, T., Wahl, F.M. (eds.) Advances in Robotics Research, pp. 275–285. Springer, Berlin (2009)
3. Ahrens, S., Levine, D., Andrews, G., How, J.P.: Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In: IEEE International Conference on Robotics and Automation, pp. 2643–2648 (2009)
4. Andert, F., Polzin, H., Dittrich, J., Becker, M., Batzdorfer, S., Hecker, P.: Aerial tracking and GNSS-reference localization with a robotic total station. In: AIAA Infotech@Aerospace Conference (2012)
5. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(5), 698–700 (1987)
6. Barbour, N.: Inertial navigation sensors. In: RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology. NATO RTO Lecture Series (2011)
7. Becker, M., Bestmann, U., Sasse, A., Steen, M., Hecker, P.: In flight estimation of gyro and accelerometer scale factors for tactical and MEMS IMUs. In: ION GNSS 20th International Technical Meeting of the Satellite Division, pp. 2056–2065 (2007)
8. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
9. Bestmann, U., Steen, M., Becker, M., Sasse, A., Hecker, P.: Comparison of state and error state INS coupling filter based on real flight test data. In: ION GNSS 20th International Technical Meeting of the Satellite Division, pp. 2611–2618 (2007)
10. Carillo, L., López, A., Lozano, R., Pégard, C.: Combining stereo vision and inertial navigation system for a quad-rotor UAV. Journal of Intelligent and Robotic Systems 65, 373–387 (2012)
11. Conte, G., Doherty, P.: A visual navigation system for UAS based on geo-referenced imagery. In: Conference on Unmanned Aerial Vehicle in Geomatics, UAV-g (2011)
12. Discher, C.: Ein dynamisches fehlermodell für die satellitenortung in einem integrierten INS/GNSS-navigationssystem. Ph.D. thesis, Technische Universität Braunschweig (2003)
13. Eynard, D., Vasseur, P., Demonceaux, C., Frémont, V.: Real time UAV altitude, attitude and motion estimation from hybrid stereovision. Autonomous Robots 33(1-2), 157–172 (2012)
14. Frietsch, N., Meister, O., Schlaile, C., Seibold, J., Trommer, G.: Vision based hovering and landing system for a VTOL-MAV with geolocalization capabilities. In: AIAA Guidance, Navigation and Control Conference (2008)
15. Grewal, M.S., Weill, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons (2001)

16. Grießbach, D., Baumbach, D., Zuev, S.: Vision aided inertial navigation. In: ISPRS EuroCOW Conference (2010)
17. Hirschmüller, H., Innocent, P.R., Garibaldi, J.M.: Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In: International Conference on Control, Automation, Robotics and Vision, pp. 1099–1104 (2002)
18. Koch, A., Wittich, H., Thielecke, F.: A vision-based navigation algorithm for a VTOL UAV. In: AIAA Guidance, Navigation and Control Conference and Exhibit (2006)
19. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
20. Matthies, L., Shafer, S.A.: Error modeling in stereo navigation. IEEE Journal of Robotics and Automation 3(3), 239–248 (1987)
21. NGA: World geodetic system (1984) – its definition and relationships with local geodetic systems. Tech. rep., National Imagery and Mapping Agency (2000)
22. Nützi, G., Weiss, S., Scaramuzza, D., Siegwart, R.: Fusion of IMU and vision for absolute scale estimation in monocular SLAM. Journal of Intelligent and Robotic Systems 61, 287–299 (2011)
23. Ramlall, R., Streter, J., Schnecker, J.F.: Three satellite navigation in an urban canyon using a chip-scale atomic clock. In: International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS), pp. 2937–2945 (2011)
24. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (2011), www.pointclouds.org
25. Sanfourche, M., Delaune, J., Besnerais, G.L., de Plinval, H., Israel, J., Cornic, P., Treil, A., Watanabe, Y., Plyer, A.: Perception for UAV: Vision-based navigation and environment modeling. Aerospace Lab Journal 4 (2012)
26. SAPOS: Satellite positioning service of the german state survey, http://www.sapos.de/
27. Weiss, S., Achtelik, M.W., Lynen, S., Chli, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: IEEE International Conference on Robotics and Automation (2012)
28. Wendel, J., Trommer, G.F.: Tightly coupled gps/ins integration for missile applications. Aerospace Science and Technology 8, 627–634 (2004)
29. Winkler, S., Schulz, H., Buschmann, M., Kordes, T., Vörsmann, P.: Improving low-cost GPS/MEMS-based INS integration for autonomous MAV navigation by visual aiding. In: ION GNSS 17th International Technical Meeting of the Satellite Division, pp. 1069–1075 (2004)

$\mathcal{L}_1$ Adaptive Control for Systems with Matched Stochastic Disturbance

Toufik Souanef, Federico Pinchetti, and Walter Fichter

Abstract. This paper presents a stochastic state feedback $\mathcal{L}_1$ adaptive control for systems with matched disturbances. The proposed approach is characterized through the introduction of a Kalman type fixed gain in the predictor. The main contribution of this work is that closed loop system analysis is demonstrated through a deterministic-like approach that uses the stochastic Laplace transform. The control is designed to accommodate and to be robust to unknown input gain as well as to system uncertainties. Simulation results show good results for the pitch angle control of a small fixed wing UAV.

1 Introduction

$\mathcal{L}_1$ adaptive control was developed for various classes of uncertain systems [1] and has shown good performance with uncertainties in the plant and external deterministic disturbances. However, in many real situations, disturbances and unmodeled dynamics in physical systems are stochastic. Systems with such random dynamics cannot be handled by deterministic analysis and design approaches. Consequently, dedicated tools are required to treat this problem.

In this paper, is considered a $\mathcal{L}_1$ adaptive control method for systems with matched random disturbances, i. e. systems corrupted by stochastic disturbance which acts in the same direction as the control variable. A Kalman type gain is introduced in the predictor architecture and it is shown that the estimation error is exponentially ultimately bounded in the mean square. A deterministic-like approach based on the stochastic Laplace transform [2, 3] is used to analyze the performance bounds of the system.

Toufik Souanef · Federico Pinchetti · Walter Fichter
Institute of Flight Mechanics and Control,
Pfaffenwaldring 7a, 70569 Stuttgart, Germany
e-mail: {toufik.souanef,federico.pinchetti,
fichter}@ifr.uni-stuttgart.de

In order to show the application potential of this approach, simulation results of a pitch rate control of a small fixed wing UAV with large uncertainties in aerodynamic parameters are presented.

2 Problem Formulation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space and given the following system represented as

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + b(\mu u(t) + \theta^\top x(t) + \sigma(t)), \qquad x(0) = x_0 \\ y(t) &= c^\top x(t) \end{aligned} \tag{1}$$

where $x(t)$ is the $\mathbb{R}^n$-valued solution to (1), $y(t)$ is the $\mathbb{R}$-valued observation of the output of the system, $A_m \in \mathbb{R}^{n\times n}$ is a known Hurwitz matrix that defines the desired dynamics of the system, b, $c \in \mathbb{R}^n$ are known constant vectors, $u(t) \in \mathbb{R}$ the control input, $\theta \in \mathbb{R}^n$ is a vector of constant unknown parameters, $\mu \in \mathbb{R}$ is an unknown constant input gain, $\sigma(t)$ is assumed to be a colored noise, i.e. a linear time invariant system driven by a white noise modeled by

$$\begin{aligned} \dot{x}_\sigma(t) &= A_\sigma x_\sigma(t) + b_\sigma w(t), \qquad x_\sigma(0) = 0_{1\times l} \\ \sigma(t) &= c_\sigma^\top x_\sigma(t) \end{aligned} \tag{2}$$

where $w(t) = w(t,\omega) : [t_0, t_f] \times \Omega \to \mathbb{R}^n$ is assumed to be zero mean Gaussian white noise process with zero mean and variance ξ, $x_\sigma \in \mathbb{R}^l$ is the state vector of the disturbance, $A_\sigma \in \mathbb{R}^{l\times l}$ is a known Hurwitz matrix that defines the dynamics of the disturbance, b_σ and $c_\sigma \in \mathbb{R}^l$ are known constant vectors.

Furthermore, the unknown parameter θ is uniformly bounded i.e. $\theta \in \Theta$ where Θ is a known compact convex set, furthermore $L = max_{\theta\in\Theta} \|\theta\|_1$. The unknown input gain μ is partially known, i.e. $\mu \in [\mu_l, \mu_u]$ where $0 < \mu_l < \mu_u$ are given lower and upper bounds of the input gain. The disturbance $\sigma(t)$ is bounded i.e. $|\sigma(t)| < \Delta$ where $\Delta \in \mathbb{R}^+$.

Taking $z(t) = (x(t)\, x_\sigma(t))^\top$ the system (1) can be written in augmented form as

$$\dot{z}(t) = Az(t) + b_u(\mu u(t) + \theta^\top x(t)) + b_w w(t), \qquad z(0) = (x_0\, 0_{1\times l})^\top \tag{3}$$

where $A = \begin{bmatrix} A_m & b\, c_\sigma \\ 0_{d\times n} & A_\sigma \end{bmatrix}$ $b_u = \begin{pmatrix} b \\ 0_{n\times 1} \end{pmatrix}$ and $b_w = \begin{pmatrix} 0_{l\times 1} \\ b_\sigma \end{pmatrix}$

The control objective is to design a state-feedback adaptive controller, such that the system described in (1) follows the desired model given by

$$\begin{aligned} \dot{x}_m(t) &= A_m x_m(t) + bk_g r(t), \qquad x_m(0) = x_0 \\ y_m(t) &= c^\top x_m(t) \end{aligned} \tag{4}$$

where $r(t)$ is is the reference input, $x_m(t)$ is the desired state vector and the static gain k_g is chosen $k_g = -1/(c^\top A_m^{-1} b)$.

3 $\mathcal{L}_1$ Adaptive Controller

Similar to the approach for systems with deterministic uncertainties [1] the proposed approach of $\mathcal{L}_1$ stochastic adaptive control is composed of the state predictor, the adaptation law and the control law Fig. 1.

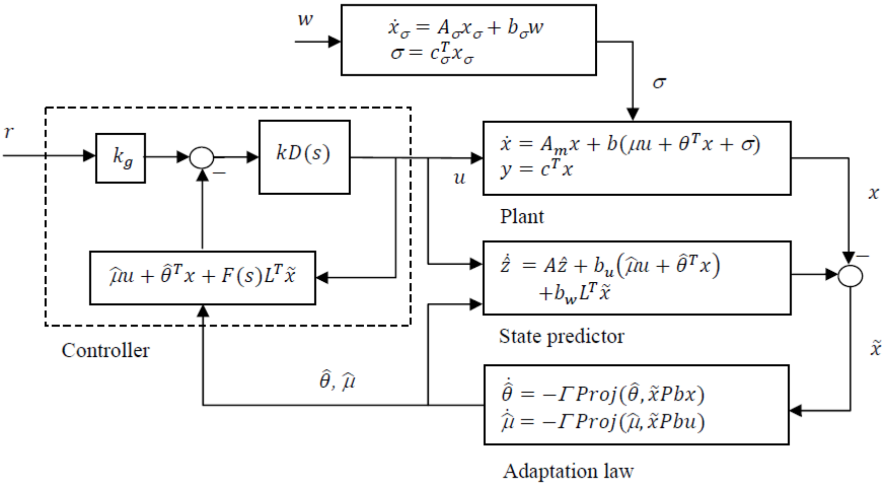

w
$\dot{x}_\sigma = A_\sigma x_\sigma + b_\sigma w$
$\sigma = c_\sigma^T x_\sigma$
σ
r
k_g
$-$
$kD(s)$
u
$\dot{x} = A_m x + b(\mu u + \theta^T x + \sigma)$
$y = c^T x$
Plant
x
$\hat{\mu}u + \hat{\theta}^T x + F(s)L^T \tilde{x}$
$\dot{\hat{z}} = A\hat{z} + b_u(\hat{\mu}u + \hat{\theta}^T x)$
$+ b_w L^T \tilde{x}$
$-$
State predictor
Controller
$\tilde{x}$
$\hat{\theta}, \hat{\mu}$
$\dot{\hat{\theta}} = -\Gamma Proj(\hat{\theta}, \tilde{x}Pbx)$
$\dot{\hat{\mu}} = -\Gamma Proj(\hat{\mu}, \tilde{x}Pbu)$
Adaptation law

Fig. 1 Block diagram of the control architecture

The expression of the state predictor, where a Kalman type gain is introduced, is given by

$$\begin{aligned} \dot{\hat{z}}(t) &= A\hat{z}(t) + b_u\left(\hat{\mu}u(t) + \hat{\theta}^\top(t)x(t)\right) + b_w L^\top \tilde{x}(t), \qquad \hat{z}(0) = z_0 \\ \hat{y}(t) &= (c^\top \, 0_{lx1})\hat{z}(t) \end{aligned} \tag{5}$$

where $\hat{z}(t) = \left(\hat{x}(t) \;\; \hat{x}_\sigma(t)\right)^\top$ is the state vector of the predictor, $\tilde{x}(t) = \hat{x}(t) - x(t)$ is the error prediction of the state vector, $\hat{\theta}$ is the estimate of the unknown parameter θ and $L \in \mathbb{R}^n$ is the Kalman type static gain vector.

The control law is given by

$$u(s) = kD(s)(k_g r(s) - \hat{\eta}(s) - F(s)L^\top \tilde{x}(s)) \tag{6}$$

where $\hat{\eta}(s)$ is the Laplace transformation of the term $\hat{\mu}(t)u(t) + \hat{\theta}^\top(t)\hat{x}(t)$, $F(s) = c_\sigma^\top(s\mathbb{I} - A_\sigma)^{-1}b_\sigma$ is the transfer function of the disturbance model and $D(s)$ is a transfer function that leads to a strictly proper stable filter $C(s) = \mu kD(s)/(1 + \mu kD(s))$ with $C(0) = 1$.

The adaptation law is defined by

$$\begin{aligned} \dot{\hat{\mu}}(t) &= \Gamma Proj\left(\hat{\mu}(t), -\tilde{x}^\top(t)Pbu(t)\right) \\ \dot{\hat{\theta}}(t) &= \Gamma Proj\left(\hat{\theta}(t), -\tilde{x}^\top(t)Pbx(t)\right) \end{aligned} \tag{7}$$

where $\Gamma \in \mathbb{R}$ is the adaptation rate, $P = P^\top > 0$ is the solution of the Lyapunov equation $A_m^\top P + PA_m = -Q$ with $Q = Q^\top > 0$ arbitrary and $Proj(\cdot,\cdot)$ is the projection operator described by the following [8]

Definition 1. Suppose that $f(\theta)$:$R^n \to R$ is a continuously differentiable smooth convex function denoted by

$$f(\theta) = \frac{(\varepsilon+1)\theta^\top\theta - \theta_{max}^2}{\varepsilon\theta_{max}^2}$$

with θ_{max} is the norm bound imposed on the vector θ and $\varepsilon > 0$ is an arbitrary tolerance bound. The gradient vector of f evaluated at θ is noted by $\bigtriangledown f(\theta)$. For a constant $\delta > 0$, consider a convex compact set with a smooth boundary given by

$$\Omega_c = \{\theta \in R^n | f(\theta) < \delta\}$$

the projection operator is defined by

$$Proj(\theta, y) = \begin{cases} y - \frac{\bigtriangledown f(\theta)(\bigtriangledown f(\theta)^\top)}{\|\bigtriangledown f(\theta)\|^2} y f(\theta) & \text{if } f(\theta) > 0 \text{ and } \bigtriangledown f(\theta)^\top y > 0 \\ y & \text{if not} \end{cases}$$

4 Analysis of the Control Architecture

Analysis of the properties of the control architecture involves showing boundedness of the prediction error and demonstrating performance bounds, i. e. the error between reference system and the closed-loop plant adaptive Controller.

4.1 Prediction Error Dynamics

In this section, the bound the estimation error is shown using mean square stochastic stability, being minimum mean square a frequently used criterion in estimation theory. First, the following definitions are recalled from [4].

Definition 2. Consider the continuous-time stochastic process described by the Itô stochastic differential equation, with a global and unique time-continuous solution

$$dx(t) = f(x,t)dt + g(x,t)dW(t) \tag{8}$$

where $W(t)$ is an independent Wiener process, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For any given $V(x) > 0 \in C^2$, associated with the stochastic system (8), the differential operator $\mathcal{L}$ is defined by:

$$\mathcal{L}V(x,t) = \frac{\partial V(x,t)}{\partial x^\top} f(x,t) + \frac{1}{2} Tr \left\{ g^\top(x,t) \frac{\partial^2 V}{\partial x \partial x^\top} g(x,t) \right\} \tag{9}$$

Definition 3. Consider the stochastic differential equation (8). Then $x(t)$ is said to be exponentially ultimately bounded in the mean square if there exist positive constants c_1, c_2 and c_3, such that for all $t \geq 0$ the following expectation is true

$$\mathbb{E}[\|x(t)\|] < c_1 e^{-c_2 t} + c_3 \tag{10}$$

where $\mathbb{E}(\cdot)$ denotes the expected value operator.

Next, it is shown in the following theorem, that the prediction error is bounded in mean square.

Theorem 1. The estimation error of the augmented system (3) with the state predictor (5) and the adaptation law (7) is mean-square exponentially ultimately bounded.

Proof. From (3) and (5) the expression of the dynamics of the prediction error can be written in Itô form as

$$d\tilde{z}(t) = \left(A\tilde{z}(t) + b_u \left(\tilde{\mu} u(t) + \tilde{\theta}^\top(t) x(t) \right) + b_w L^\top \tilde{x}(t) \right) dt - \xi b_w dW(t) \tag{11}$$

where $\tilde{\mu}(t) = \hat{\mu}(t) - \mu$, $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$ and $W(t)$ is an increment of a Wiener process (Brownian motion) with zero-mean Gaussian distribution and variance given by $\mathbb{E}[dW(t) dW^\top(t)] = 1$.

Under the assumption that the unknown parameters μ and θ are constant, using the adaptation law (7) it can be written

$$\begin{aligned} d\tilde{\mu}(t) &= \Gamma Proj\left(\hat{\mu}(t), -\tilde{x}(t) P b u(t)\right) dt \\ d\tilde{\theta}(t) &= \Gamma Proj\left(\hat{\theta}(t), -\tilde{x}(t) P b x(t)\right) dt \end{aligned} \tag{12}$$

Taking $\zeta(t) = \begin{pmatrix} \tilde{x}(t) & \tilde{x}_\sigma(t) & \tilde{\mu}(t) & \tilde{\theta}(t) \end{pmatrix}^\top$ equations (11) and (12) are written

$$d\zeta(t) = f(\zeta, t) dt + g(\zeta, t) dW \tag{13}$$

where $f(\zeta,t) = \begin{pmatrix} A_m \tilde{x}(t) + b\left(\tilde{\mu} u(t) + \tilde{\theta}^\top(t) x(t) + c_\sigma^\top \tilde{x}_\sigma(t)\right) \\ A_\sigma \tilde{x}_\sigma + b_\sigma L^\top \tilde{x}(t) \\ \Gamma Proj\left(\hat{\mu}(t), -\tilde{x}(t) P b u(t)\right) \\ \Gamma Proj\left(\hat{\theta}(t), -\tilde{x}(t) P b x(t)\right) \end{pmatrix}$ and $g(\zeta,t) = \begin{pmatrix} 0_{n\times 1} \\ -\xi b_\sigma \\ 0 \\ 0 \end{pmatrix}$

Considering the Lyapunov function candidate

$$V(\zeta, t) = \tilde{x}^\top(t) P \tilde{x}(t) + \tilde{x}_\sigma^\top(t) P_\sigma \tilde{x}_\sigma(t) + \Gamma^{-1} \left(\tilde{\theta}^\top(t) \tilde{\theta}(t) + \tilde{\mu}^2(t) \right) \tag{14}$$

with $P_\sigma = P_\sigma^\top > 0$ is the solution of the Lyapunov equation $A_\sigma^\top P_\sigma + P_\sigma A_\sigma = -Q_\sigma$ with $Q_\sigma = Q_\sigma^\top > 0$ arbitrary. Thus, the expression of the differential generator of $V(\zeta,t)$ is written

$$\begin{aligned}\mathcal{L}V(\zeta,t) =& \tilde{x}^\top(t)(PA_m + A_m^\top P)\tilde{x}(t) + 2\tilde{x}^\top(t)Pb\tilde{\mu}u(t) + 2\tilde{x}^\top(t)Pb\tilde{\theta}^\top(t)x(t)\\ &+ 2\tilde{x}^\top(t)Pbc_\sigma^\top\tilde{x}_\sigma(t) + \tilde{x}_\sigma^\top(t)(P_\sigma A_\sigma + A_\sigma^\top P_\sigma)\tilde{x}_\sigma(t)\\ &+ 2\tilde{x}_\sigma^\top(t)P_\sigma b_\sigma L^\top\tilde{x}(t) + 2\tilde{\mu}(t)Proj\left(\hat{\mu}(t), -\tilde{x}(t)Pbu(t)\right)\\ &+ 2\tilde{\theta}^\top(t)Proj\left(\hat{\theta}(t), -\tilde{x}(t)Pbx(t)\right)\\ &+ \frac{1}{2}\xi^2 Tr\left\{b_\sigma^\top P_\sigma b_\sigma\right\}\\ =& -\tilde{x}^\top(t)Q\tilde{x}(t) - \tilde{x}_\sigma^\top(t)Q_\sigma\tilde{x}_\sigma + 2\tilde{x}^\top(t)\left(Pbc_\sigma^\top + Lb_\sigma^\top P_\sigma\right)\tilde{x}_\sigma(t)\\ &+ 2\tilde{\mu}(t)\left(\tilde{x}^\top(t)Pbu(t) + Proj(\hat{\mu}(t), -\tilde{x}(t)Pbu(t))\right)\\ &+ 2\tilde{\theta}^\top(t)\left(x(t)\tilde{x}^\top(t)Pb + Proj(\tilde{\theta}(t), -\tilde{x}(t)Pbx(t))\right)\\ &+ \frac{1}{2}\xi^2 Tr\left\{b_\sigma^\top P_\sigma b_\sigma\right\}\end{aligned}\tag{15}$$

Given the adaptation law in (7) one can derive the following bound

$$\begin{aligned}\mathcal{L}V(\zeta,t) \leq& -\lambda_{min}(Q)\tilde{x}^\top(t)\tilde{x}(t) - \lambda_{min}(Q_\sigma)\tilde{x}_\sigma^\top(t)\tilde{x}_\sigma(t)\\ &+ 2\tilde{x}^\top(t)\left(Pbc_\sigma^\top + Lb_\sigma^\top P_\sigma\right)\tilde{x}_\sigma(t) + \frac{1}{2}\xi^2 Tr\left\{b_\sigma^\top P_\sigma b_\sigma\right\}\end{aligned}\tag{16}$$

where $\lambda_{max}(\cdot)$, $\lambda_{min}(\cdot)$ are respectively the maximum/minimum eigenvalue of a matrix.

Choosing $L = -Pbc_\sigma^\top P_\sigma b_\sigma (b_\sigma^\top P_\sigma^2 b_\sigma)^{-1}$ enables the elimination of the third right hand term of inequality (16). Furthermore, for simplicity, the arbitrary matrices Q_σ and Q can be chosen such that $\lambda_{min}(Q_\sigma) = \lambda_{min}(Q)$ and consequently

$$\mathcal{L}V(\zeta,t) \leq -\lambda_{min}(Q)\|\tilde{z}\|^2 + \frac{1}{2}\xi^2 Tr\left\{b_\sigma^\top P_\sigma b_\sigma\right\}\tag{17}$$

Moreover, since the projection-based adaptation law ensures that $\hat{\theta}(t) \in \Theta$ and $\mu_l \leq \hat{\mu}(t) \leq \mu_u$, hence the Lyapounov function $V(\zeta,t)$ in (14) can be bounded as

$$\begin{aligned}V(\zeta,t) \leq& \tilde{x}^\top(t)P\tilde{x}(t) + \tilde{x}_\sigma^\top(t)P_\sigma\tilde{x}_\sigma(t) + \frac{1}{\Gamma}\left(4\theta_{max}^2 + (\mu_u - \mu_l)^2\right)\\ :\leq& \, max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]\|\tilde{z}\|^2 + \frac{1}{\Gamma}\left(4\theta_{max}^2 + (\mu_u - \mu_l)^2\right)\end{aligned}\tag{18}$$

where $\theta_{max} = max_{\theta\in\Theta}\|\theta\|^2$.

Further given

$$\lambda_{min}(Q)\|\tilde{z}\|^2 = \frac{\lambda_{min}(Q)}{max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]} max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]\|\tilde{z}\|^2 \tag{19}$$

it follows that

$$\lambda_{min}(Q)\|\tilde{z}\|^2 \geq \frac{\lambda_{min}(Q)}{max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]}\left(V(\zeta,t) - \frac{1}{\Gamma}\left(4\theta_{max}^2 + (\mu_u - \mu_l)^2\right)\right) \tag{20}$$

and thus the upper bound in (17) can be used to obtain

$$\mathcal{L}V(\zeta,t) \leq -k_1 V(\zeta,t) + k_2 \tag{21}$$

where

$$k_1 = \frac{\lambda_{min}(Q)}{max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]}$$

and

$$k_2 = \frac{1}{\Gamma}\frac{\lambda_{min}(Q)\left(4\theta_{max}^2 + (\mu_u - \mu_l)^2\right)}{max\left[\lambda_{max}(P), \lambda_{max}(P_\sigma)\right]} + \frac{1}{2}\xi^2 Tr\left\{b_\sigma^\top P_\sigma b_\sigma\right\}$$

From [5], it follows that

$$\begin{aligned} \mathbb{E}\left[V(\zeta,t)\right] &\leq v_0\, e^{-k_1 t} + \frac{k_2}{k_1}\left(1 - e^{-k_1 t}\right) \\ :&\leq \left(v_0 - \frac{k_2}{k_1}\right)e^{-k_1 t} + \frac{k_2}{k_1} \end{aligned} \tag{22}$$

where $v_0 = \mathbb{E}\left[V(\zeta,0)\right]$.

Given

$$min\left(\lambda_{min}(P), \lambda_{min}(P_\sigma), \Gamma^{-1}\right)\|\zeta\|^2 \leq V(\zeta,t) \tag{23}$$

it can be written

$$\mathbb{E}\left[\|\zeta\|^2\right] \leq \mathbb{E}\left[V(\zeta,t)\right]/min\left(\lambda_{min}(P), \lambda_{min}(P_\sigma), \Gamma^{-1}\right) \tag{24}$$

and consequently the prediction error $\zeta = [\tilde{x}(t),\, \tilde{x}_\sigma(t),\, \tilde{\mu}(t)$ and $\tilde{\theta}(t)]^\top$ is exponentially ultimately bounded in mean square and the proof is complete □

Lemma 1. The following bound holds, almost surely, for the prediction error of state vector of the plant

$$\|\tilde{x}(t)\| \leq \rho(t),\ \ \forall t \geq 0$$

where

$$\rho(t) = \frac{1}{\sqrt{\lambda_{min}(P)}}\sqrt{\frac{k_2}{k_1}\left(2e^{k_1\delta}-1\right)e^{\varepsilon t}+e^{2k_1\delta}\left(v_0-\frac{k_2}{k_1}\right)e^{-(k_1-2\varepsilon)t}}$$

where $\delta > 0$, and $\varepsilon \in (0, k_1/2)$ are arbitrary constants. Furthermore, we have the Lyapunov exponent

$$\limsup_{t\to\infty}\frac{1}{t}\log\|\tilde{x}(t)\| \leq 0$$

Proof. If equation (21) is verified, thus using [7] theorem 7, it follows that for the arbitrary constants δ, and ε defined above, there exists a random instant t_0 such that for all $t > t_0$, we have

$$\lambda_{min}(P)\|\tilde{x}(t)\|^2 \leq V(\zeta,t) \leq \frac{k_2}{k_1}\left(2e^{k_1\delta}-1\right)e^{\varepsilon t}+e^{2k_1\delta}\left(v_0-\frac{k_2}{k_1}\right)e^{-(k_1-2\varepsilon)t} \quad (25)$$

Furthermore, given $\lambda_{min}(P)\|\tilde{x}(t)\|^2 \leq V(\zeta,t)$, thus for $t > 0$ the following inequality holds

$$\frac{1}{t}\log\left(\lambda_{min}(P)\|\tilde{x}(t)\|^2\right) \leq \frac{1}{t}\log\left(V(\zeta,t)\right) \quad (26)$$

and using here again [7] theorem 7 it follows that

$$\limsup_{t\to\infty}\frac{1}{t}\log\|\tilde{x}(t)\| \leq \limsup_{t\to\infty}\frac{1}{t}\log\left(V(\zeta,t)\right) \leq 0 \quad (27)$$

and the proof is complete □

Remark. Note that choosing high adaptation gain, contribute to the optimization of the bound of the prediction error $\|\tilde{x}(t)\|$. Indeed minimizing the factor k_2/k_1 leads to an optimal bound of the estimation error.

Given

$$\frac{k_2}{k_1} = \frac{1}{\Gamma}\left(4\theta_{max}^2+(\mu_u-\mu_l)^2\right)+\frac{1}{2}\xi^2 Tr\{b^\top Pb\}\frac{max\left[\lambda_{max}(P),\lambda_{max}(P_\sigma)\right]}{\lambda_{min}(Q)}$$

For high adaptation gain, this factor can be approximated to become

$$\frac{k_2}{k_1} \simeq \frac{1}{2}\xi^2 Tr\{b^\top Pb\}\frac{max\left[\lambda_{max}(P),\lambda_{max}(P_\sigma)\right]}{\lambda_{min}(Q)}$$

solving this problem of optimization by LMI methods as in [7] will lead to an optimal almost sure bound of the prediction error.

4.2 Closed Loop Reference System

In this section, the reference system, i.e. the closed loop system with nominal parameters, is introduced and its stability is shown through the use of stochastic Laplace transform [2, 3]. Stochastic Laplace transform is an extension of the theory of Laplace transforms in the context of the Itô-Doob stochastic calculus. This method provides an algebraic approach for finding Itô-Doob type stochastic integrals and solving stochastic linear differential equations of the Itô-Doob type.

Definition 4. Let $g(t,W(t))$ be a real valued function of two variables $(t,W(t))$ defined for all real numbers $t \geq 0$ and $W(t)$ be a Wiener process. The Laplace transform of g in the sense of the Itô-Doob integral or stochastic Laplace transform is denoted by

$$G^W(s) = L^W(g(t,W(t))) = \int_{t=0}^{\infty} e^{-st} g(t,W(t)) dW(t) \tag{28}$$

for all values of s for which this improper integral exists.

Note that the stochastic Laplace transform inherits linearity, derivative, integral and convolution properties of deterministic Laplace transforms [2, 3].

Next, in order to derive the dynamics of the reference system of the plant, the case of known parameters is considered and it is written as

$$\begin{aligned} x_{ref}(t) &= A_m x_{ref} + b(\mu u_{ref}(t) + \theta^\top x_{ref}(t) + \sigma(t)) \quad x_{ref}(0) = x_0 \\ y_{ref}(t) &= c^\top x_{ref} \end{aligned} \tag{29}$$

The control law is given by

$$u_{ref}(s) = \frac{C(s)}{\mu}\left(k_g r(s) - \theta^\top x_{ref}(s) - F(s)K^\top \tilde{x}_{ref}(s)\right) \tag{30}$$

Defining proper BIBO stable transfer functions $H(s) = (s\mathbb{I} - A_m)^{-1} b$ and $G(s) = H(s)(1 - C(s))$, the stability of the closed loop reference system is demonstrated through the following lemma.

Lemma 2. If the filter $C(s)$ is designed such that the $\mathcal{L}_1$ norm condition [1]

$$\|G(s)\|_{\mathcal{L}_1} L < 1$$

is verified, then the closed-loop reference system in (29) and (30) is BIBS stable with respect to the reference input and initial conditions.

Proof. From (2) the expression of the stochastic disturbance is written in Itô form as

$$\begin{aligned} dx_\sigma(t) &= A_\sigma x_\sigma dt + b_\sigma \xi dW(t) \\ \sigma(t) &= c_\sigma x_\sigma(t) \end{aligned} \tag{31}$$

writing (31) as an integral equation leads to

$$x_\sigma(t) = \int_0^t A_\sigma x_\sigma(\tau)d\tau + b_\sigma \xi \int_0^t dW(\tau) \tag{32}$$

Using the symbol L as a Laplace operator and taking the Laplace transformation of (32) it can be written

$$\begin{aligned} X_\sigma(s) =& L\left(\int_0^t A_\sigma x_\sigma(\tau)d\tau + \xi b_\sigma \int_0^t dW(\tau)\right) \\ =& A_\sigma L\left(\int_0^t x_\sigma(\tau)d\tau\right) + b_\sigma \xi L\left(\int_0^t dW(\tau)\right) \end{aligned} \tag{33}$$

Using properties of the stochastic Laplace transform [2, 3] it can be written

$$L(dW(t)) = L^W(1) \tag{34}$$

and consequently

$$X_\sigma(s) = \frac{1}{s} A_\sigma X_\sigma(s) + b_\sigma \xi \frac{L^W(1)}{s} \tag{35}$$

which leads to

$$(s\mathbb{I} - A_\sigma)X_\sigma(s) = b_\sigma \xi L^W(1) \tag{36}$$

and hence, the Laplace transformation of the stochastic disturbance $\sigma(t)$ is written

$$\Sigma(s) = c_\sigma^\top X_\sigma(s) = \xi F(s) L^W(1) \tag{37}$$

Consequently, the closed loop reference system is written

$$\begin{aligned} x_{ref}(s) =& G(s)\theta^\top x_{ref}(s) + C(s)H(s)k_g r(s) \\ & - C(s)H(s)F(s)K^\top \tilde{x}_{ref}(s) + \Sigma(s) + x_{in}(s) \end{aligned} \tag{38}$$

where $x_{in}(s) = (s\mathbb{I} - A_m)^{-1} x_0$. Thus, from [1] lemma A.7.1 it follows that for all $\tau \geq 0$ the following bound holds

$$\begin{aligned} \left\|x_{ref\ \tau}\right\|_{\mathcal{L}_\infty} \leq & \left\|G(s)\theta^\top\right\|_{\mathcal{L}_1} \left\|x_{ref\ \tau}\right\|_{\mathcal{L}_\infty} + \left\|C(s)H(s)k_g\right\|_{\mathcal{L}_1} \left\|r_\tau\right\|_{\mathcal{L}_\infty} \\ & + \left\|C(s)H(s)F(s)K^\top\right\|_{\mathcal{L}_1} \left\|\tilde{x}_{ref\ \tau}\right\|_{\mathcal{L}_\infty} + \left\|\sigma_\tau\right\|_{\mathcal{L}_1} + \left\|x_{in}\right\|_{\mathcal{L}_\infty} \end{aligned} \tag{39}$$

Given the condition of $\mathcal{L}_1$ stability [1] equation (39) is written

$$\begin{aligned} \left\|x_{ref\ \tau}\right\|_{\mathcal{L}_\infty} \leq & \frac{\left\|C(s)H(s)k_g\right\|_{\mathcal{L}_1}\|r_\tau\|_{\mathcal{L}_\infty}}{1-\|G(s)\|_{\mathcal{L}_1}L} \\ & + \frac{\left\|C(s)H(s)F(s)K^\top\right\|_{\mathcal{L}_1}\left\|\tilde{x}_{ref\ \tau}\right\|_{\mathcal{L}_\infty}+\Delta+\|x_{in}\|_{\mathcal{L}_\infty}}{1-\|G(s)\|_{\mathcal{L}_1}L} \end{aligned} \tag{40}$$

Since $r(t), x_{in}(t)$, $\psi(t)$ and $\tilde{x}_{ref}(t)$ are bounded it is straightforward that the reference state $x_{ref}(t)$ in (40) is bounded and the proof is complete. □

4.3 Performance Bounds

The following theorem states on the transient performances of the closed loop system i. e. the tracking errors between the reference system and the plant with $\mathcal{L}_1$ adaptive control, and it is shown that the transient regime is strongly connected to the estimation error.

Theorem 2. Given the closed loop system (1), (6) and the reference system (29) (30), the following bound holds

$$\begin{aligned} \left\|x(t)-x_{ref}(t)\right\|_\infty &\leq \frac{2}{\sqrt{\lambda_{min}(P)}}\left(\varphi_{H_1}(t)*\rho(t)\right) \\ \left\|u(t)-u_{ref}(t)\right\|_\infty &\leq \frac{2}{\sqrt{\lambda_{min}(P)}}\left(\varphi_{H_2}(t)*\rho(t)\right) \end{aligned}$$

where $\varphi_e(t)=\max_{i=1,\ldots,n}\sqrt{\sum_{j=1}^m e_{ij}^2(t)}$, e_{ij} (t) is the ith row, jth column of the impulse response matrix of $E(s)$ [1] and $H_1(s)$, $H_2(s)$ are defined below.

Proof. The control law in (6) can be written as

$$u(s)=\frac{C(s)}{\mu}\left(k_g r(s)-\theta^\top x_{ref}(s)-F(s)K^\top\tilde{x}_{ref}(s)-\tilde{\eta}(s)\right) \tag{41}$$

where $\tilde{\eta}(s)$ is the Laplace transform of the term $\tilde{\mu}(t)u(t)+\tilde{\theta}^\top(t)x(t)$. The closed loop system takes the form

$$x(s)=G(s)\theta^\top x(s)+C(s)H(s)\left(k_g r(s)-F(s)K^\top\tilde{x}(s)-\tilde{\eta}(s)\right)+\Sigma(s)+x_{in}(s) \tag{42}$$

From (38) it follows that

$$\begin{aligned} x_{ref}(s)-x(s)=&G(s)\theta^\top\left(x_{ref}(s)-x(s)\right)\\ &-C(s)H(s)\left(F(s)K^\top\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right)+\tilde{\eta}(s)\right) \end{aligned} \tag{43}$$

Given (11) the Laplace transform of error dynamics of the plant $\tilde{x}(t)$ and the reference $\tilde{x}_{ref}(t)$ is written

$$\begin{aligned} \tilde{x}(s)&=H(s)\tilde{\eta}(s)+H(s)F(s)K^\top\tilde{x}(s)-\Sigma(s)\\ \tilde{x}_{ref}(s)&=H(s)F(s)K^\top\tilde{x}_{ref}(s)-\Sigma(s) \end{aligned} \tag{44}$$

Replacing (44) in (43) leads to

$$x_{ref}(s)-x(s)=H_1(s)\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right)\le 2H_1(s)\,\rho(s) \tag{45}$$

where $H_1(s)=-\left(\mathbb{I}-G(s)\theta^\top\right)^{-1}C(s)$. Using the same approach in [1] lemma 2.2.6, (45) leads to the bound of the state vector.

To show the bound of the control law, from equations (6) and (30) it is written

$$u_{ref}(s)-u(s)=-\frac{C(s)}{\mu}\left(\theta^\top\left(x_{ref}(s)-x(s)\right)+\left(K^\top\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right)-\tilde{\eta}(s)\right)\right) \tag{46}$$

From Lemma A.12.1 in [1] there exists a vector c_0 such that

$$\begin{aligned} u_{ref}(s)-u(s)=&-\frac{C(s)}{\mu}\theta^\top(x_{ref}(s)-x(s))\\ &-\frac{C(s)}{\mu}\frac{1}{c_0^\top H(s)}c_0^\top H(s)\left(F(s)K^\top\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right)-\tilde{\eta}(s)\right) \end{aligned} \tag{47}$$

From (44) and it follows that

$$\begin{aligned} u_{ref}(s)-u(s)=&-\frac{C(s)}{\mu}\theta^\top(x_{ref}(s)-x(s))\\ &-\frac{C(s)}{\mu}\frac{1}{c_0^\top H(s)}c_0^\top\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right) \end{aligned} \tag{48}$$

and there exists also a vector c_1 such that

$$\begin{aligned} u_{ref}(s)-u(s)=&-\frac{C(s)}{\mu}\theta^\top(x_{ref}(s)-x(s))\\ &-\frac{C(s)}{\mu}\frac{1}{c_0^\top H(s)}c_0^\top\frac{1}{c_1^\top H_1(s)}c_1^\top H_1(s)\left(\tilde{x}_{ref}(s)-\tilde{x}(s)\right) \end{aligned} \tag{49}$$

Defining $H_2(s) = -\frac{C(s)}{\mu}\left(\theta^\top + \frac{1}{c_0^\top H(s)} c_0^\top \frac{1}{c_1^\top H_1(s)} c_1^\top\right)$ leads to

$$u_{ref}(s) - u(s) = H_2(s)\left(x_{ref}(s) - x(s)\right) \tag{50}$$

Using the same approach in [1] lemma 2.2.6, equation (50) leads to the expression of the bound of the control law and completes the proof. □

5 Simulation Results

The control law proposed and analyzed in previous sections is now applied to the pitch dynamics of a fixed wing UAV, a model of the Monsun BO 209. Tracking performance under stochastic matched disturbance and model uncertainties is shown through simulations.

The short period dynamics can be written in matrix form as in [9]

$$\underbrace{\begin{pmatrix}\dot{\alpha}\\ \dot{q}\end{pmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}\frac{Z_\alpha}{V_a} & 1+\frac{Z_q}{V_a}\\ M_\alpha & M_q\end{bmatrix}}_{A} \underbrace{\begin{pmatrix}\alpha\\ q\end{pmatrix}}_{x} + \underbrace{\begin{pmatrix}\frac{Z_\delta}{V_a}\\ M_\delta\end{pmatrix}}_{b} \underbrace{\delta_e}_{u}$$

$$y = q = \underbrace{(0 \;\; 1)}_{c^\top} x$$

where α is the angle of attack, q is the pitch angular velocity, V_a is the trimmed airspeed, $(Z_\alpha,\, Z_q, Z_\delta)$ and $(M_\alpha, M_q, M_\delta)$ are the partial derivatives of the aerodynamic force Z and the pitching moment M, with respect to α, q, and δ_e, respectively. The control input is the elevator angle δ_e.

Numerical values for the considered UAV trimmed at $V_{a0} = 20\ m/s$, $\alpha_0 = 4°$, $q_0 = 0°/s$, and altitude h$_0$= 50 m are given by

$$A = \begin{bmatrix}-11.0447 & 0.9644\\ -242.4575 & -14.4717\end{bmatrix}, \quad b = \begin{pmatrix}-0.2840\\ -112.4126\end{pmatrix}.$$

Note that actuator dynamics are assumed to be negligible.

The system is affected by an additive matched disturbance $\sigma(t)$ with dynamics $F(s) = 1/s+1$ and input $w(t)$ as a white noise with variance $\xi = 1$.

The desired dynamics matrix A_m for system (1) is chosen such that it meets military specifications for category A, level-1 flight handling qualities requirement and eigenvalues for the short period mode are given from [9] as $\lambda_{1,2} = -5.6 \pm 4.2j$, i. e. a pulsation $\omega_n = 7 rad/s$ and a damping $\varsigma = 0.8$.

$\mathcal{L}_1$ adaptive controller parameters are set $\Gamma = 100000$, $D(s) = 1/s$ and $k = 2000$. Note however that even if this value of k is high it affects only the filter $C(s)$ and it has no effect on stability margins of the controller [10].

The control approach is based on the augmentation of a linear controller by the $\mathcal{L}_1$ adaptive controller, as the common approach in aerospace systems. This allows the

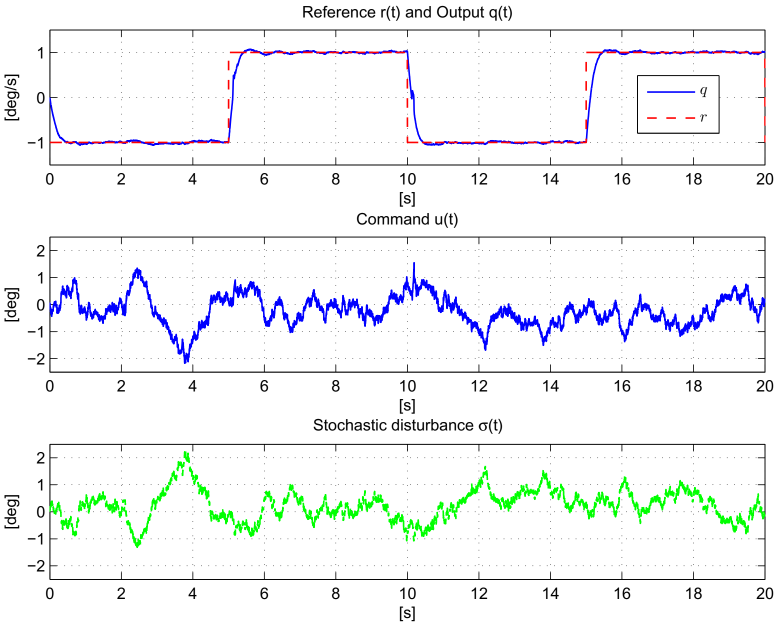


Fig. 2 Output of the closed loop system for system without failures

use of the available knowledge about the system dynamics. The adaptive controller is then added to compensate unknown parameters and / or disturbances effect.

The controller is designed to be robust against uncertainties and the compact sets are set to $\mu = (0.41.4)$, $\Theta = \left\{\vartheta = (\vartheta_1, \vartheta_2) \in \mathbb{R}^2 : \vartheta_i \in (-2, 2), i = 1, 2\right\}$. From the definition of Θ it follows that $L = 2$, which results in $\|G(s)L\|_1 = 0.2277$ which satisfies the $\mathcal{L}_1$ stability condition.

Fig. 2 depicts the response of the system to a square signal reference. It can be seen that the control architecture reduces the effect of noise disturbance.

It is noted in Fig. 3 that the prediction error of the state vector is bounded and negligible, thus it can be concluded that the predictor is stable and it presents good prediction performance.

To show pertinence of the developed controller, failures are introduced to the system as a loss of actuator effectiveness $\mu = 0.5$ and the UAV becomes marginally stable $M_q = 0$, and statically unstable $M_\alpha > 0$. Such drastic and perhaps unrealistic situation is motivated by the intent to demonstrate the effectiveness of the proposed approach. Fig. 4 and Fig. 5 show that even under large uncertainties, the system has good performance and compensates disturbance effect. Moreover, the elevator command is within acceptable limits.

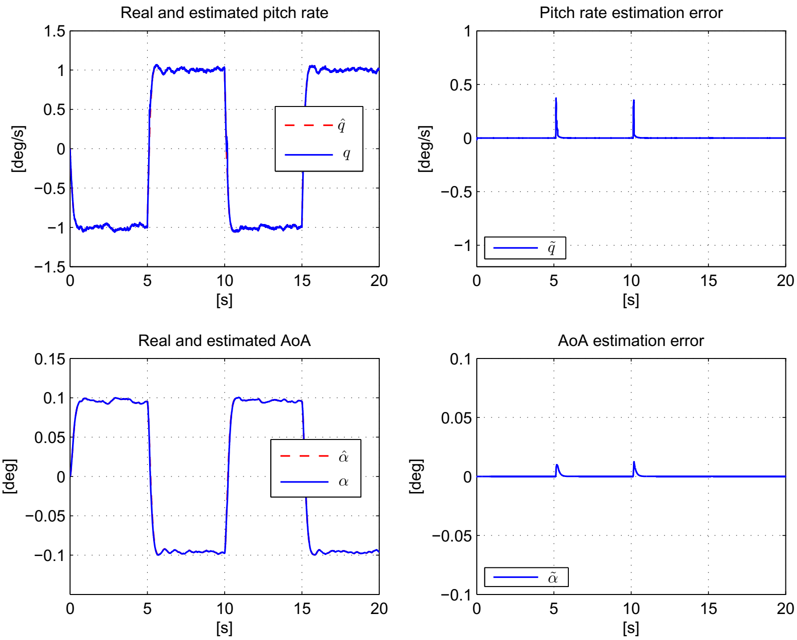


Fig. 3 Estimation performance of the system without failures

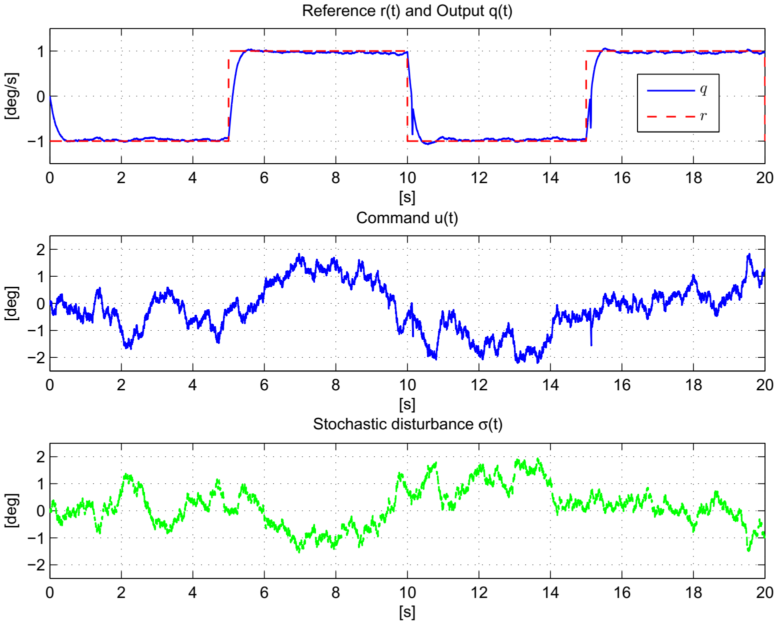


Fig. 4 Output of the closed loop system for system with failures

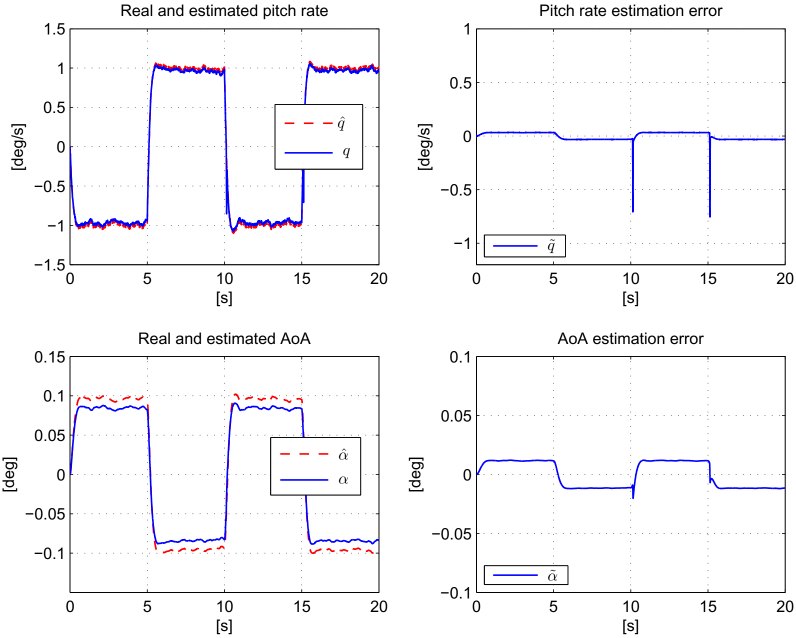


Fig. 5 Estimation performance of the system with failures

6 Conclusion

In this paper, a $\mathcal{L}_1$ adaptive control scheme for systems with matched stochastic uncertainties has been proposed. A Kalman type gain is introduced in the predictor architecture and it is shown that the estimation error is exponentially ultimately bounded in the mean square. Closed loop boundedness and performance are analyzed using the stochastic Laplace transform. This permits the use of an analysis approach similar to systems with deterministic disturbances.

Simulation results showed good performances for pitch angle control of a small fixed wing UAV in the presence of strong disturbances (such as air turbulences).

This work is a starting point on the application of $\mathcal{L}_1$ adaptive approach to the control of stochastic systems and several areas of investigation remain for this proposed control method, including non matched disturbances and output-feedback systems with measurement noise. Furthermore, the use of stochastic Laplace transform opens a lot of perspectives for random systems control and analysis.

References

1. Cao, C., Hovakimyan, N.: L$_1$ Adaptive Control Theory Guaranteed Robustness with Fast Adaptation. SIAM, Philadelphia (2010)
2. Kirby, R.D.: Qualitaitve Behaviour of Dynamical Vector Fields. PhD thesis, University of Texas at Arlington (2007)

3. Kirby, R.D., Ladde, A.G., Ladde, G.S.: Stochastic Laplace Transform with Application. Communications in Applied Analysis 14(4), 373–392 (2010)
4. Kuchner, H.J.: Stochastic Stability and Control. Academic Press (1967)
5. Zakai, M.: On the ultimate boundedness of moments associated with solutions of stochastic differential equations. SIAM Journal on Control 5(4), 588–593 (1967)
6. Tarn, T.J., Rasis, Y.: Observers for nonlinear stochastic systems. IEEE Transactions on Automatic Control 21(4), 441–448 (1976)
7. Xie, L., Khargonekar, P.P.: Lyapunov-based adaptive state estimation for a class of nonlinear stochastic systems. Automatica 48, 1423–1431 (2012)
8. Pomet, J.B., Praly, L.: Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Transactions on Automatic Control 37, 729–740 (1992)
9. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation, 2nd edn. Wiley, New York (2003)
10. Cao, C., Hovakimyan, N.: Stability margins of L1 adaptive control architecture. IEEE Transactions on Automatic Control 55, 480–487 (2010)

Part III
Atmospheric Applications

Decoupling the Eye: A Key toward a Robust Hovering for Sighted Aerial Robots

Augustin Manecy, Raphaël Juston, Nicolas Marchand, and Stéphane Viollet

Abstract. Inspired by natural visual systems where gaze stabilization is at a premium, we simulated an aerial robot with a decoupled eye to achieve more robust hovering above a ground target despite strong lateral and rotational disturbances. In this paper, two different robots are compared for the same disturbances and displacements. The first robot is equipped with a fixed eye featuring a large field-of-view (FOV) and the second robot is endowed with a decoupled eye featuring a small FOV (about ±5°). Even if this mechanical decoupling increases the mechanical complexity of the robot, this study demonstrates that disturbances are rejected faster and the computational complexity is clearly decreased. Thanks to bio-inspired visuo-motor reflexes, the decoupled eye robot is able to hold its gaze locked onto a distant target and to reject strong disturbances by profiting of the small inertia of the decoupled eye.

Part of this paper reprinted from "Bio-Inspired Hovering Control for an Aerial Robot Equipped with a Decoupled Eye and a Rate Gyro" by A.Manecy, S. Viollet and N. Marchand, which appeared in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. 2012 IEEE [15].

Augustin Manecy · Nicolas Marchand
GIPSA-Lab Laboratory,
Control Systems Dept., SySCo Team,
CNRS-Univ. of Grenoble,
ENSE^3 BP 46, 38402,
St Martin d'Hères Cedex, France
e-mail: augustin.manecy@univ-amu.fr,
nicolas.marchand@gipsa-lab.fr

Augustin Manecy · Raphaël Juston · Stéphane Viollet
Aix-Marseille Université,
Biorobotic Dept. of ISM, CNRS,
ISM UMR7287, 13288, Marseille Cedex 09, France
e-mail: {raphael.juston,stephane.viollet}@univ-amu.fr

Acronyms

FOV	Field Of View.
rVOR	Rotational Vestibulo-Ocular Reflex.
tVOR	Translational Vestibulo-Ocular Reflex.
VFR	Visual Fixation Reflex.
ZSL	Zero-Setting System.
VFL	Visual Feedback Loop.
D-EYE	Decoupled eye system.
F-EYE	Fixed eye system.

1 Introduction

Several methods have been developed during the last few years which have enabled UAVs to fly increasingly autonomously (to perform automatic taking off and landing, etc.). Even if it is possible to estimate the attitude of an UAV using only an IMU, this last cannot be used to estimate the position without drifts. As a consequence, IMUs are combined with other sensors to provide both an attitude and pose estimations. Some ways consist in the fusion of the IMU and the GPS to avoid drifts in estimation, as in [29] or [1] and more recently [19]. And most of the strategies used for this purpose were based on a combination of vision sensors and Inertial Measurement Units. For instance, [8] used a trajectometric system of measurement to determine the position and orientation of a quad-rotor. Despite the high frame rate and the good resolution of this system, the robot cannot be said to be fully autonomous because of the off-board data processing system on which it depends. A similar system involving the use of a CCD camera was developed by [13]. Another strategy consisted of using active markers placed in the environment (one under the robot and one in front of it), as described by [18] to assess the robot's attitude and position. Along similar lines, using an embedded camera and a different set of geometrical markers (five), [33] provided a robot with an accurate means of estimating both its position and its orientation. A similar task was also accomplished in a study by [30], using active markers and a simple visual sensor borrowed from a remote Wii control (Wiimote). Other methods based on the use of optical flow have been presented, as in [22], for determining the altitude, position and speed of a robot flying above a road with specific geometric characteristics. In all these approaches, the robot's position was estimated via visual sensors, as in [8] combined with an embedded inertial measurement unit. In the latter case, the gyrometer's bias was classically compensated for by means of accelerometers (see [5]).

To make UAVs more autonomous, several methods relying on SLAM algorithms use monocular camera to navigate in an unknown GPS-denied environment ([28]). Other SLAM methods use laser rangefinder as an active optical sensor, combined with a laser mirror for height estimation as in [2] or [7]. SLAM was also performed by merging both information from a camera and from a laser rangefinder [23].

We focused our work on the advantages provided by a decoupled eye embedded onboard a UAV. In previous robotic studies, a decoupled eye is classically used to track a target and compensate for the UAV displacements around this target. This objective was achieved for example in [17], where a target was successfully tracked (from an autonomous helicopter). The positions and attitude of the helicopter were estimated by means of a data fusion algorithm between a GPS, a magnetometer and an IMU. We demonstrate here that a decoupled eye (with a narrow field of view) could be used not only to track a target, but also to estimate the position and the attitude of a UAV using only the retinal error, the orientation of the eye with respect to the robot and a rate gyro.

In this way, the hovering robot presented here was assumed to have an ”eye” with a controlled rotation relative to its ”body”. In addition, the eye was taken to have a narrow FOV of only a few degrees (a kind of *fovea*). The visual sensor is able to rotate and thus to change the gaze direction (the line of sight). As described in [3], a *fovea* equipped with a gaze control mechanism of this kind is a considerable step forward in the computational modeling of vision, where visual and control systems with many degrees of freedom have to solve difficult problems without gaze control mechanisms. The present additional degree of freedom mimics the characteristic mechanical decoupling between the eye and body of many animals, such as the hoverfly. The present robot's eye can be said to be a sensitive, accurate visual position-sensing device (PSD) [10], which is able to detect the position of an edge (or a bar) within its very small field of view (here, $FOV = \pm 5^\circ$ in comparison with the FOV of more than 50° in the case of the robots used in previous references). This sensor's performances in an edge-detection task were a 40-fold improvement in terms of the resolution, as compared with the previous interphotodiode angular resolution [26]. The visual sensor in question can therefore be said to be endowed with hyperacuity [31]. For further details about the performances (i.e., the accuracy and calibration) of this hyperacute visual PSD, see [27] and [26]. Gaze stabilization is a difficult task because the eye control system must compensate both quickly and accurately for any sudden, untoward disturbances caused by the vagaries of the supporting head or body. This finely adapted mechanism is way beyond what can be achieved in the field of present-day robotics. The only information available onboard the present robot, in line with what occurs in its natural counterparts, is the inertial measurement provided by the biased rate gyro and the eye-in-body orientation provided by a hall effect sensor.

The robot with a decoupled eye is presented in the next section, along with its nonlinear dynamic model. In Section 3, the original nonlinear observer used to estimate the robot's position, attitude (roll axis only) and unbiased rotational speed is presented. The eye control system and the robot's overall control system based on this observer are then described. In section 4, the advantages of implementing a feedforward gaze control process are discussed and detailed comparisons are made between the behavior of the simulated robots with and without a decoupled eye. A comparison between a fixed eye robot endowed with a large FOV and the decoupled eye robot with a narrow field of view is equally done to demonstrate advantages provided by the decoupling.

2 System Overview

In this section, we describe the aerial robot and the computational resources embedded on-board. We describe also the nonlinear model of the robot which was used to simulate the different scenarios.

2.1 The Twin Engine Hovering Robot

In this part, we introduce the robot and its equipment. We explain the different objectives and describe our bio-inspired approach.

2.1.1 The Robot's Hardware

As shown in the CAD in figure 1, the twin-engine aerial robot we have designed will have three degrees of freedom (a rotation around the horizontal axis, θ_r, a right and left translation, X, and an eye rotation θ_{er} with respect to the robot's body). Thanks to a mechanical decoupling between the eye and its mechanical support (the head), the eye can rotate freely in the robot's frame. In addition, the eye's orientation can be finely controlled by means of an extremely compact, fast and accurate servo (MKS). The robot, which will weigh only about 150 grams in all, will be completely autonomous, thanks to its embedded computational resources and the on-board power supply described as depicted in figure 2.

As shown in figure 4, the HyperRob's roll angle θ_r can be controlled by applying a differential rotational speed to the propellers. The robot was assumed in the present simulations to fly at a constant altitude H. This assumption is not restrictive, since it is known that the altitude can be decoupled from the roll and lateral

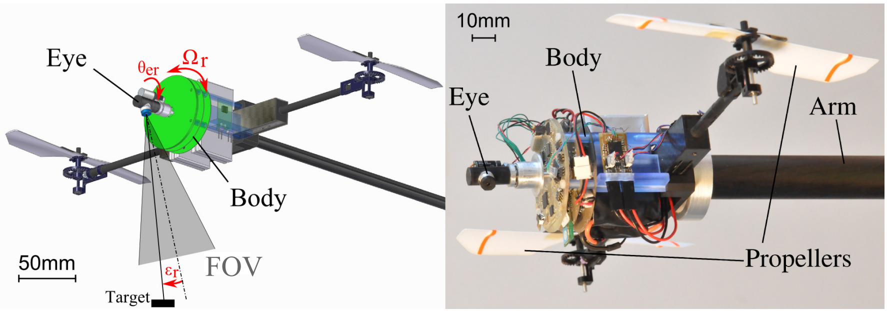


Fig. 1 (a) CAD of the 150-grams micro-air vehicle, called HyperRob, in which a fast micro servo (MKS servo) controls the orientation of the eye (angle θ_{er}) relative to that of the body. The orientation of the robot around the roll axis (angle θ_r) is controlled by applying a differential rotational speed to the propellers. The robot itself was mounted here at the tip of a rotating arm allowing to move freely in the horizontal plane. (b) Picture of the HyperRob robot fixed at the tip of an arm rotating in the azimut plane.

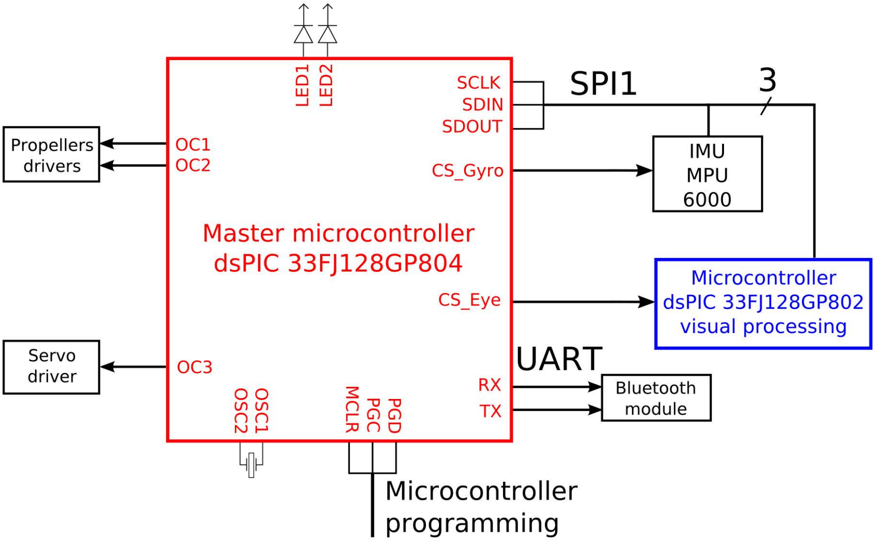


Fig. 2 Simplified scheme of the embedded electronics. The robot is equipped with two Microchip dsPIC microcontrollers (16 bits). The main microcontroller (dsPIC 33FJ128GP804) runs a multirate Simulink-based program, which is in charge of the main control tasks. A secondary microcontrollers (dsPIC 33FJ128GP802) is used to process the visual signals of the eye. The main microcontroller sends the set point specifying both the eye's angular position and the throttle of the two propellers via PWM signals. The autopilot uses solely the rate gyro (roll axis) of an inertial measurement unit (MPU6000) and digital signals from the visual processing unit. A Bluetooth wireless device connected to the UART peripheral can be used by the operator to log data received from the robot and to send the robot data and the start/stop instructions.

movements in systems of this kind, which belong to the same class as aircraft capable of performing planar vertical take- off and landing ([22], [30] or [13]). As the flying robot is under-actuated, its position X on the horizontal axis is controlled by adjusting its attitude around the roll axis. The robot is therefore in a state of equilibrium when $\theta_r = 0°$. In this study, the robot was assumed to be hovering above a target on the ground.

2.1.2 The Bio-inspired Approach

Setting our hovering control strategy in a bio-inspired minimalistic framework meant that the objective was to stabilize the under-actuated hovering robot using only a drifting rate gyro and an eye with a narrow field of view. As can be seen from Figure 3, our robot has many points in common with the fly:

- **A rate gyro:** the fly has two gyroscopic halteres organ measuring its body's angular speed in the three degrees of rotation (pitch, roll and yaw) [9], whereas the robot is equipped with a classical MEMS rate gyro.
- **An optical position sensing device:** the fly's compound eye is able to locate a contrasting target placed in a small frontal part of the visual field [6], [4], while the robot is equipped with an eye endowed with hyperacuity [12], with which it can accurately locate the targets occurring in its small FOV.
- **A neck:** the fly has no less than 23 pairs of muscles with which to control its head's orientation [25]. The robot has a decoupled eye, which is actuated by means of a tiny position servomotor (MKS).
- **A proprioceptive sensor in the neck:** the fly has prosternal organs consisting of a pair of mechanosensitive hair fields located in the neck [20], while the robot is equipped with a contactless magnetoresistive sensor measuring the orientation of the eye relative to the head.
- **A gaze stabilization:** in the freely flying sand wasp, active gaze stabilization mechanisms prevent the incoming visual information from being affected by disturbances, such as large body rotations around the roll axis [32]. The robot therefore uses two bio-inspired oculomotor reflexes to compensate for its own body movements.

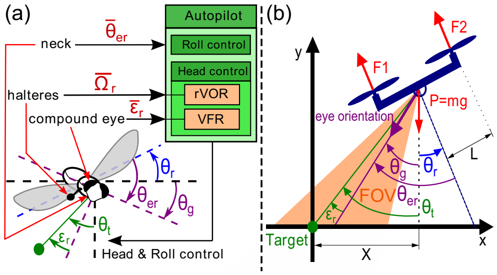


Fig. 3 Similarities between the fly (a) and the hovering robot with a decoupled eye (b). These two dynamic under-actuated systems are able to measure their body's rotational speed Ω_r by means of a rate gyro (in the case of the robot) and halteres (in that of the fly) and to locate a contrasting target θ_t placed in a small part of their FOV. The fly has no less than 23 pairs of muscles in its neck, with which to stabilize its gaze θ_g, whereas the robot controls the angular position of its eye θ_{er} by means of a small servomotor. Here, the fly and the robot are hovering over a ground target.

In what follows, we have used the following notations:

- θ_t: the angular position of the target in the inertial frame.
- θ_r: the robot's roll angle.
- θ_{er}: the angle between the eye and the robot in the robot's frame. This angle is mechanically constrained to a maximum value: $|\theta_{er}| < \theta_{er\_MAX}$.
- θ_g: the angular position of the gaze in the inertial frame $\theta_g = \theta_{er} + \theta_r$.
- ε_r: the retinal error defined by $\varepsilon_r = \theta_g - \theta_t$.

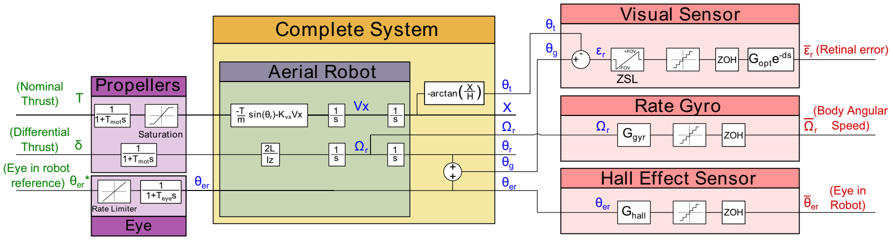


Fig. 4 Block diagram of the complete system. The robot is equipped with a rate gyro, which measures the rotational speed around the roll axis and a decoupled eye locked onto a distant target. Green variables are the control input signals, blue ones are the main physical variables of interest and red ones are the measured values.

- X: the position of the robot along the horizontal axis in the inertial frame.
- V_x: the speed of the robot on the horizontal axis in the inertial frame.
- Y: the position of the robot on the vertical axis in the inertial frame. Y was assumed to be constant ($Y(t) = H \quad \forall t$).
- Ω_r: the robot's rotational roll speed.

Estimated values are denoted by an additional hat (e.g., $\hat{\theta}$), reference values by a star (e.g., $\theta^\star$) and measured values by a bar (e.g., $\bar{\theta}$).

2.2 Non-linear Model of the Hovering Robot

A classical nonlinear dynamic model was adopted for the robot in the inertial frame:

$$\begin{cases} \dot{V}_x = \dfrac{-(F_1+F_2)\sin(\theta_r)}{m} - K_{vx}V_x \\ \dot{X} = V_x \\ \dot{\Omega}_r = \dfrac{L(F_2-F_1)}{I_z} \\ \dot{\theta}_r = \Omega_r \end{cases} \tag{1}$$

where L is the distance between the center of the robot and the propellers, I_z is the inertial momentum around the roll axis, F_1 and F_2 are the thrust generated by propellers 1 and 2, respectively, and K_{vx} is the flapping coefficient, which is assumed to be constant.

An internal speed loop makes the robot's rotational speed (Ω_r) exactly follow the rotational speed set points ($\Omega_r^\star$) dictated by the attitude controller (see 3.4). To control the rotational speed (Ω_r), it was assumed that the propellers are controlled directly via the thrust value. The control input signal to a propeller is therefore composed of a nominal thrust and a differential thrust. The nominal thrust ($T^\star$) counteracts the gravity and the differential thrust ($\delta^\star$) generates the torque responsible for the roll rotation. The propeller's control input signals are defined by $F_1^\star(p) = T^\star(p) - \delta^\star(p)$ and $F_2^\star(p) = T^\star(p) + \delta^\star(p)$. The propellers' dynamics were assumed to correspond to a first order system with a time constant equal to τ_{mot} (see table 2).

Figure 4 shows the complete model of the robot including the propellers, the rate gyro, the visual sensor and the angle sensor (used to measure the eye-in-robot angle θ_{er}). In the dynamic simulations of the robot, which were computed with the Matlab/Simulink environment, the set of parameters presented in table 1 were used.

3 Gaze, Attitude and Position Stabilization

In this section we present the hierarchy and the interconnections between the different controllers. The stabilization problem was divided in three independent stabilization loops. As the robot is under actuated, a first low level controller allows to track the rotational speed reference. This reference is yielded by the high level position and attitude controller which use the roll angle to achieve the wished position. And finally a reflex based controller allows to lock the target in the small FOV of the eye thanks to adjusting the eye-in-robot angle.

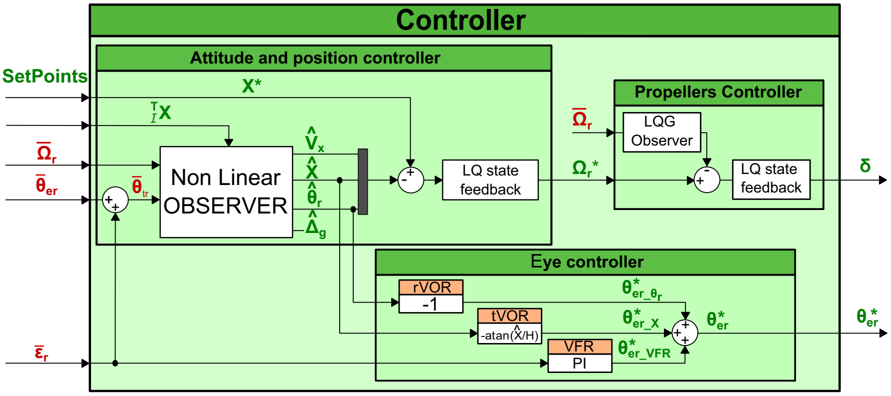


Fig. 5 Block-diagram of the controller. The nonlinear observer determines the robot's attitude on the sole basis of its angular speed and the eye orientation measurements. The eye controller is composed of three oculomotor reflexes (ORs), which are in charge of keeping the eye locked on the target whatever translational or rotational disturbances may occur. Measured (simulated) signals are presented in red and control input signals in green.

3.1 The Nonlinear Observer

In the classical approach, the attitude estimation problem is solved thanks to a nonlinear complementary filter ([14]) using both the accelerometers and the rate gyros. In this case, the rate gyros offer good information for estimating robot attitude during aggressive maneuvers. And the accelerometers are used as an inclinometer relative to the gravity acceleration allowing to compensate rate gyro bias. In this work, the estimation problem is solved without using classical IMU, and the complete attitude, position relative to target and rate gyro's bias are obtained with only two measurements.

The nonlinear observer is the cornerstone of our hovering control strategy, because it delivers the inputs of the position and attitude controller and especially for the VOR (see 3.2). So on the basis of two measurements which are the rotational speed $\bar{\Omega}_r$ and the retinal error $\bar{\varepsilon}_r$, the nonlinear observer yields an estimate for the linear speed V_x, the position X, the roll angle θ_r and the rate gyro's bias Δ_g:

$$\begin{cases} \begin{pmatrix} \dot{\hat{V}}_x \\ \dot{\hat{X}} \\ \dot{\hat{\theta}}_r \\ \dot{\hat{\Delta}}_g \end{pmatrix} = \begin{pmatrix} -\frac{T}{m}\sin(\hat{\theta}_r) - K_{vx}\hat{V}_x + L_1\tilde{\theta}_{tr} \\ \hat{V}_x + L_2\tilde{\theta}_{tr} \\ \bar{\Omega}_r - \hat{\Delta}_g + L_3\tilde{\theta}_{tr} \\ L_4\tilde{\theta}_{tr} \end{pmatrix} \\ \hat{\theta}_{tr} = \underbrace{-\arctan\left(\frac{\hat{X}}{H}\right)}_{\hat{\theta}_g} - \hat{\theta}_r \end{cases} \tag{2}$$

Where $\tilde{\theta}_{tr} = (\underbrace{\bar{\theta}_{er} + \bar{\varepsilon}_r}_{\bar{\theta}_{tr}} - \hat{\theta}_{tr})$

We introduce here the angle θ_{tr} which corresponds to the orientation of the target in the robot's frame. This new angle improves our previous work ([15]), where it was assumed that $\theta_{er} = \theta_{tr}$ and that the observer's innovation term was $\tilde{\theta}_{er} = \bar{\theta}_{er} - \hat{\theta}_{er}$. As a consequence, the retinal error was used to improve the estimation of the angular position of the target. This modification improves the quality of the estimations and leads to a better roll stabilization (see figure 9.b) in comparison to our previous work.

We decided to implement a nonlinear observer because of the strongly nonlinear equations giving the evolution of the linear speed V_x and the robot's position X (see equation (2)). Non-linearities in $\hat{V}_x$ give more accurate estimations during transient responses, whereas non linearities in $\hat{\theta}_{er}$ result in non steady state error in the position estimates. Observer gain $L = \begin{pmatrix} L_1 & L_2 & L_3 & L_4 \end{pmatrix}^T$ was tuned by applying the classical LQG method to the system linearized around the origin.

3.2 The Eye Controller

This controller is essential for the robot, because it makes the robot's gaze locked onto the target on the ground. This gaze control loop has to be really fast and accurate to provide to the eye the capacity to reject disturbances which the robot without decoupled eye could not reject (due to its bigger inertia). This fast controller is achieved by merging three complementary oculomotor reflexes which combine feedforward and feedback control:

- A *rotational* vestibulo ocular reflex, called the rVOR, yields the output signal $\theta^{\star}_{er\_\theta_r}$, which is simply equal to the opposite of the estimated roll angle $\hat{\theta}_r$.
- A *translational* vestibulo ocular reflex, called the tVOR, which depends on the estimated robot's linear position $\hat{X}$, assuming that the robot's altitude is known. This reflex compensates for any translation applied to the robot's body by producing the output signal $\theta^{\star}_{er\_X}$, which contributes to keeping the eye locked onto the target.
- A visual fixation reflex (VFR), where the visual feedback loop cancels the retinal signal error ε_r by controlling the eye's orientation θ_{er} via the control input signal $\theta^{\star}_{er\_VFR}$ (see figure 6).

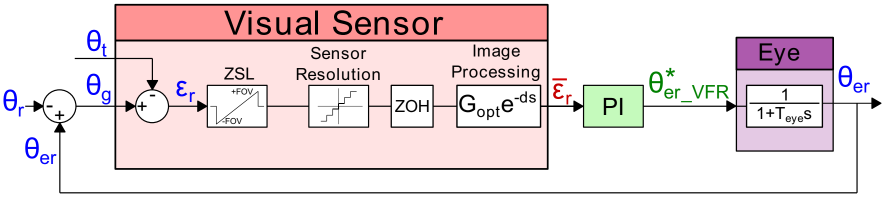


Fig. 6 Block diagram of the visual feedback loop used to implement the visual fixation reflex (VFR). A Proportional-Integral controller cancels the retinal error ε_r. The ZSL "Zero-Setting Limiter" serves here to prevent runaway of the eye if the target is lost. G_{opt} is the static gain of the visual sensor and d is a pure delay introduced by the image processing.

It is worth noting that the ZSL function shown in figure 6 clamps the retinal error back to zero whenever the latter tends to become higher (or lower) than a specified positive (or negative) level. This ZSL, which was used in previous studies ([27], [11]) serves the same purpose as the limiter block used to model the inhibition of the smooth pursuit reflex whenever the position error goes beyond a fixed threshold [21], [24].

3.2.1 Visual Fixation Reflex (VFR)

The VFR depicted in figure 6 plays the most important role because it makes the robot:

- reject any lateral disturbances (gusts of wind),
- correct unmodelled dynamics in the feedforward terms,
- follow a moving target with a high level of accuracy.

The VFR controller is a simple proportional-integral controller, which keeps the retinal error close to 0° by producing a reference angle ($\theta^{\star}_{er\_VFR}$). When designing the PI controller, the pure delay resulting from the computational time and the visual sensor's latency was approximated by a first order Pade approximation.

3.2.2 Vestibulo Ocular Reflexes (VOR)

As in its natural counterpart, the human oculomotor reflexes [16], the VOR based on a feedforward control (see figure 5) causes the robot to compensate for movements of two different kinds:

- **roll:** the rVOR uses the estimated roll angle $\hat{\theta}_r$ to compensate for rotation of the body.
- **lateral translation:** the tVOR based on the estimated robot's linear position $\hat{X}$ minimizes the effects of lateral displacements on the retinal error ε_r.

To summarize, the reference angle $\theta^{\star}_{er}$ (see figure 5) results from the three reflexes (rVOR, tVOR and VFR) as follows:

$$\theta^{\star}_{er} = \underbrace{\theta^{\star}_{er\_\theta_r}}_{rVOR} + \underbrace{\theta^{\star}_{er\_X}}_{tVOR} + \underbrace{\theta^{\star}_{er\_VFR}}_{VFR} \tag{3}$$

Figure 7 shows the contribution of the three reflexes during a 1-m imposed lateral displacement with respect to the target (a voluntary movement achieved by changing the set point $X^{\star}$) when a 20-cm lateral perturbation (an untoward disturbance corresponding to a gust of wind) was applied to the robot at $t = 7s$. It is worth noting that during the imposed translation, the rVOR reacted first (green curve), followed by the VFR (black curve) compensating for the roll variation. In the case of a lateral perturbation without any rotational component, it can be seen, as might be expected, that the VFR reacted faster than the rVOR and tVOR reflexes.

3.3 The Rotational Speed Controller

This rotational speed controller makes the robot's follow any change in the rotation speed set points yielded by the attitude-position controller. The propellers are assumed to be driven directly via the thrust. This assumption is not too restrictive because the thrust could be easily obtained thanks:

$$F_i = c_T \omega_i^2 \tag{4}$$

Where F_i is the thrust of the rotor i and ω_i is the rotor's rotation speed. The constant c_T correspond to the thrust coefficient which could be simply identified using static thrust tests.

This controller consists of a classical controller-observer obtained thanks to the LQ methodology, instead of a PID controller, in order to include the rate gyro's filter dynamics and the motor dynamics. This dynamics are considered to be two first order:

$$\begin{pmatrix} \dot{\Omega}_r \\ \dot{\bar{\Omega}}_r \end{pmatrix} = \begin{pmatrix} \frac{-2L}{I_z \tau_{mot}} & 0 \\ \frac{1}{\tau_{gyr}} & \frac{-1}{\tau_{gyr}} \end{pmatrix} \begin{pmatrix} \Omega_r \\ \bar{\Omega}_r \end{pmatrix} + \begin{pmatrix} \frac{2L}{I_z \tau_{mot}} \\ 0 \end{pmatrix} \delta^{\star} \tag{5}$$

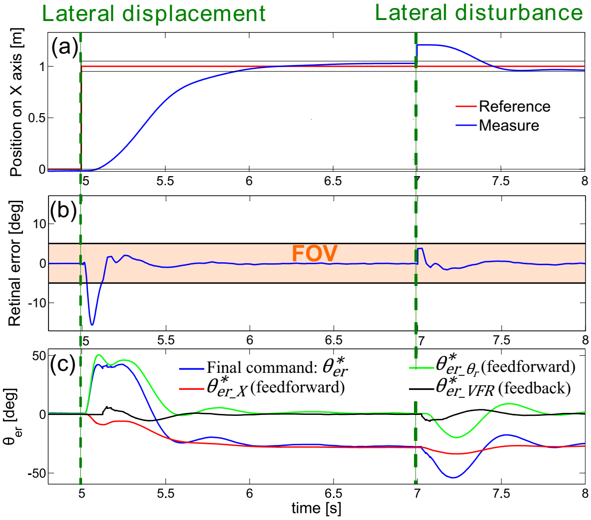


Fig. 7 Response of the robot to a 1-m imposed lateral displacement (a) and a 20-cm lateral disturbance applied at $t = 7s$ (90% of the disturbance is rejected within 0.9s). During the lateral displacement, the VORs react first, and elicit an eye rotation to compensate for the changes in the robot's roll behavior. The VFR then compensates for the observer's model errors, and keeps the target within the FOV. When a lateral perturbation occurs, the VFR responds by generating a rotation to keep the target within the FOV. And only a few milliseconds later, the VORs take over (when the robot begins to move to counteract the displacement due to the disturbance).

where Ω_r is the actual rotational roll speed and $\bar{\Omega}_r$ is the rotational speed measured by the rate gyro.

Any static errors occurring in the Ω_r tracking is rejected by an integral effect. Thanks to this controller, the closed loop response time in is less than $20ms$, and the noise in the differential thrust control is less than 2%.

3.4 Position and Attitude Controller

The position and attitude controller were implemented by means of another LQR controller using the states estimated by the nonlinear observer (V_x, X and θ_r). Since the dynamics of the rotational speed feedback loop, which have been previously described (see 3.3), were assumed to be much faster than the attitude-position controller, they were neglected.
To design the LQR state feedback, the robot's model (1) was linearised around the origin with the equilibrium control input $\Omega_{req} = 0$. To cancel any steady state errors, an integral effect was added to the position X.

Figure 7 shows a 1-meter displacement along the X axis.

4 Advantages of the Decoupled Eye

The aim of this project is to demonstrate that a decoupled ”eye” system presents many benefits contrary to a classical fixed ”eye” even if the field of view is restricted to few degrees.

4.1 Advantage of the VORs

Figure 8 shows the response of the robot’s position to a 1-m reference step input imposed by the setpoint $X^{\star}$ (see figure 5). When the ORs were turned off, only the VFR remained active (the VORs were off). It can be clearly seen from the strong oscillatory response of the robot with no VORs shown in figure 8a and the retinal error shown in figure 8b that the robot with VORs never lost the target (except the initial peak due to the time response of the eye) and showed much smoother and stiffer dynamics than the robot with no VORs. It is worth noting that the robot with no VORs finally succeeded in stabilizing, thanks to the ZSL (see figure 6), which prevented the robot’s attitude from becoming unstable when the target was lost.

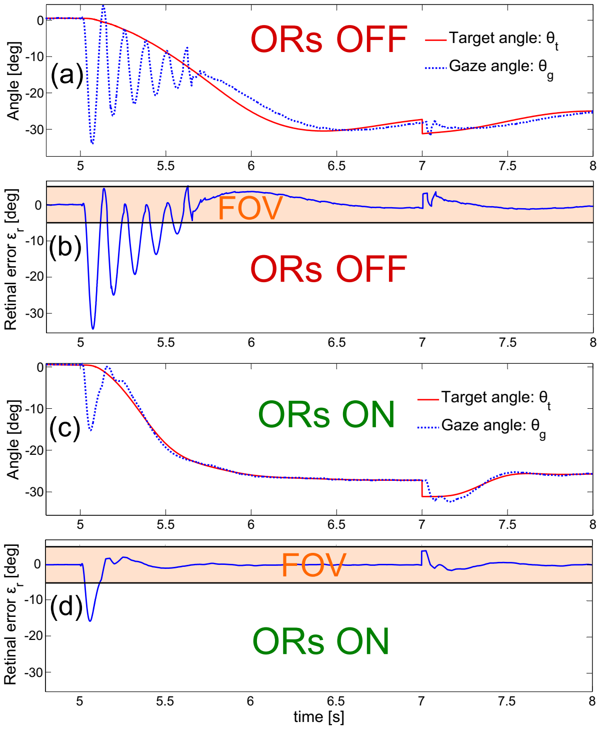


Fig. 8 A 1-m step change in the setpoint $X^{\star}$ led the robot with no ORs (ORs off) to produce a strong oscillatory response in its gaze orientation (a) and thus in the retinal error (b). At $t = 7s$ a 20-cm lateral disturbance was applied. Unlike the robot with the ORs switched off, the robot with the ORs switched on can be seen to have rotated its gaze smoothly in order to shift its linear position by 1m with respect to the target (c and d).

4.2 Better Disturbances Rejection

In this part we present a comparison of the comportment of two identical robots, one with a fixed eye (denoted F-EYE robot) and the other with the decoupled eye (denoted D-EYE robot). In a first time we will compare two robots with the same FOV, and in a second time, we will assume that the F-EYE robot is equipped with a large FOV (±50°).

4.2.1 Comparison with a Fixed Eye Robot Featuring the Same FOV

The two robots have exactly the same FOV and as a consequence of the limited FOV, the F-EYE robot can not generate aggressive maneuvers. So, this implies that the position-controller of the F-EYE robot is less aggressive. Indeed, as it is shown in figure 10a and 10b, when a lateral disturbance appears, it generates a step in the retinal error, allowing to the robot to up to date its position and correct it. But, as the robot is under-actuated, to adjust its position, it is necessary to increase its roll angle which increases again the retinal error. As a consequence, the limit of the FOV is reached for the F-EYE robot, which implies a limitation in roll angle. This limitation implies inevitably a slower rejection of disturbance.

Remark 1. The controller of the D-EYE robot and the F-EYE robot are the same except that the eye controller was removed for the F-EYE robot (as the eye can not rotate). Similarly, the LQ state feedback of the attitude and position controller is quite different. The weighting coefficient of the control input for the F-EYE is 10 times bigger than the one for the D-EYE robot, to avoid going out of FOV. Indeed,

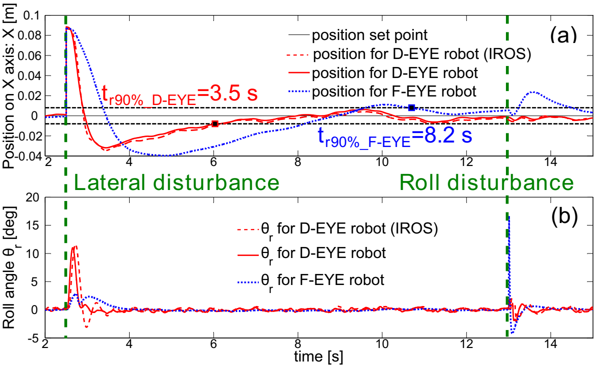


Fig. 9 Responses of the D-EYE and F-EYE hovering robots to a 8-cm lateral disturbance applied at $t = 2.5s$ and a strong 20° rotational disturbance applied at $t = 13s$. The robot with a decoupled eye can be seen to have rejected 90% of the lateral disturbance within 3.5s, whereas the robot with a fixed eye took 8.2s to reject the same disturbance. In b), one can see the improvement with respect to the roll angle provided by adding the retinal error at the observer's input, in comparison to our previous work (IROS).

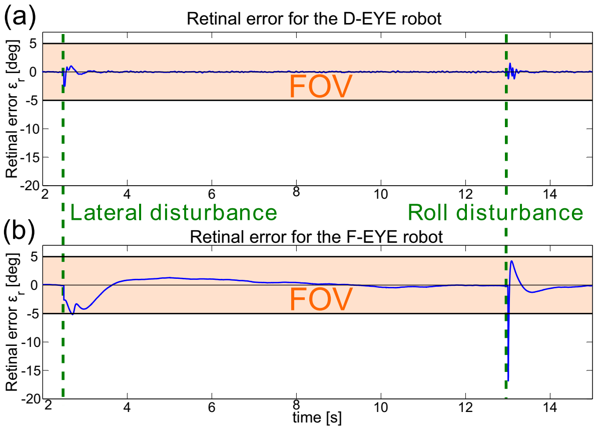


Fig. 10 Retinal error with the two disturbances described in figure 7. The retinal error occurring in the case of the robot with a decoupled eye (a) can be seen to have stayed within the limits of the FOV, but (b) reached the limit of the FOV after the lateral perturbation and left the FOV completely for 30ms in response to the rotational disturbance of 20° .

to correct its position the robot has to generate roll angle, which implies to increase the retinal error for the F-EYE robot. So, to avoid the target to go out of the F-EYE robot's FOV, the position controller is chosen a little less aggressive. This less aggressive controller is achieved by increasing the weighting coefficient of the control input. So we have chosen the lower cost (that is mean the most aggressive comportment), allowing to perform a 1-m displacement without to lose the target for the F-EYE robot.

For the F-EYE and the D-EYE robots, the LQ cost function for the state feedback is respectively:

$$\begin{cases} J_F = \int (S^T Q_F S + \Omega_r^T R_F \Omega_r) dt \\ J_D = \int (S^T Q_D S + \Omega_r^T R_D \Omega_r) dt \end{cases} \tag{6}$$

Where $Q_F = Q_D$, $R_F = 10R_D$ and S the state vector $S = (V_x, (X - X^{\star}), \theta_r, \Delta_g)^T$.

Figure 9 shows the position adopted by the robot and its roll angle in response to a strong lateral 8-cm perturbation (an impulsional perturbation similar to a gust of wind) applied at time $t = 2.5s$ and to a 20° rotational step disturbance around the roll axis applied at $t = 13s$. Despite the large lateral disturbance, it can be seen from figure 9a that the D-EYE robot rejects 90% of the perturbation within 3.5s, whereas the F-EYE robot rejects the same perturbation within a much longer period of 8.2s. The faster dynamics of the D-EYE robot was obtained at the expense of a much more aggressive control of the roll dynamics (see red curve in figure 9b), while keeping the retinal error within the ±5° (limits of the FOV visible on figure 10a). This fast rejection is not achievable with the F-EYE robot due to the limit of FOV which are reached (figure 10b). Similarly, the roll disturbance was completely rejected by the D-EYE robot thanks to the ORs, whereas the retinal error left the F-EYE robot's FOV for 30ms and it took 1s for the error to be completely canceled out.

4.2.2 Comparison with a Fixed Eye Robot Featuring a Large FOV

Visual Computational Resources for the F-EYE Robot

The F-EYE robot is supposed to be equipped with a classical monocular camera with a large FOV of ±50°. Such of camera requires at least 1000 pixels to provide the same resolution as the F-EYE visual sensor (0.1°).

Visual Computational Resources for the D-EYE Robot

The D-EYE robot will be equipped with a custom-made visual sensor endowed with hyperacuity (i.e., the ability to locate a target with a resolution greater than the one composed by the pixel pitch [31]). So, the visual sensor will be composed of few pixels (2 pixels), submitted to an active mechanical vibration in order to obtain a resolution as small as 0.1°for a FOV of only ±5°([10], [12] and [26]). The visual signal processing is described in [10] and will run on a tiny embedded target (dsPIC, see figure 2).

Simulation Results

In the previous part, the controller of the F-EYE robot was less aggressive because of the small FOV and the risk of losing the target for too much roll angle. In this part, we provide to the F-EYE robot a larger FOV of ±50° than the D-EYE robot. And thanks to this larger FOV, it is possible to have exactly the same controller for the two robots, that is mean that in (6) $Q_F = Q_D$ and $R_F = R_D$.

Figures 11 and 12 show the response of the two robots to a strong 15-cm lateral perturbation applied at time $t = 3s$ and to a 35° rotational step disturbance around the roll axis applied at $t = 13s$. These two perturbations are bigger than the perturbations applied previously in 4.2.1). It is worth noting that even if the controllers are exactly the same, the D-EYE robot rejects faster the lateral and the roll disturbances than the F-EYE robot. This phenomenon could be explained by the fact that, unlike the F-EYE robot, the D-EYE robot is able to cancel all rotational movements

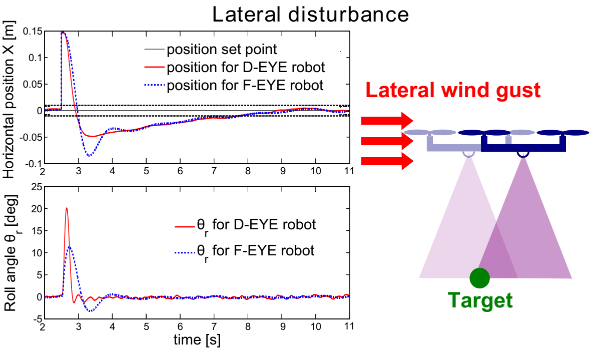
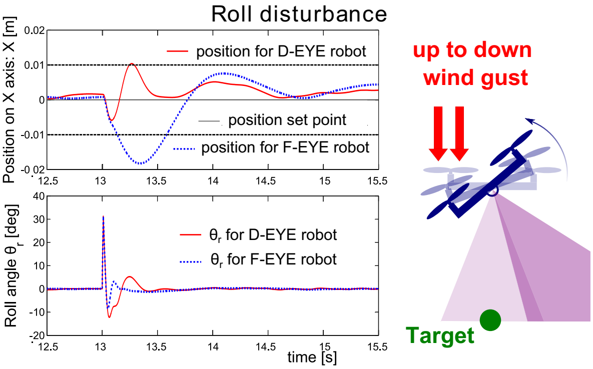


Fig. 11 Response of the D-EYE and F-EYE hovering robots to a 15-cm lateral disturbance applied at $t = 2.5s$. The D-EYE robot rejects quiet better the disturbances with a smaller overshoot even if its FOV is only ±5°, while the FOV of the F-EYE robot is 50°. This faster rejection is due to the faster convergence of the nonlinear observer.

Fig. 12 Response of the D-EYE and F-EYE hovering robots to a strong 35° roll disturbance applied at $t = 13s$. The D-EYE robot rejects faster the lateral displacement resulting from the roll disturbances with a smaller overshoot.



applied on its eye. As a consequence, the D-EYE robot can easily disambiguate a lateral from a rotational disturbance. As a consequence the D-EYE robot provides therefore richer information for the nonlinear observer, which is then able to correct the estimated lateral position with a faster dynamics than the observer of the F-EYE robot. In the figure 11, it can be seen that the D-EYE robot presents a 32-% overshoot whereas the F-EYE robot response has a 57-% overshoot. The same phenomenon appears in figure 12 for the roll disturbance where the D-EYE robot has a much more aggressive response in roll and stays close to zero centimeter at $\pm 1.2cm$ whereas the F-EYE robot is much less accurate with a shift in its lateral position of more than $1.9cm$.

5 Conclusion

In this paper, a new bio-inspired method of stabilizing an under-actuated hovering aerial robot equipped with a decoupled eye is presented. The results of the simulations performed show that the oculomotor reflexes responsible for stabilizing the gaze and controlling the eye's orientation greatly improve the robot's ability to compensate for strong lateral or rotational (up to 20°) disturbances.

Despite the small field of view of the robot's eye, the new degree of freedom introduced by the decoupled eye easily compensates for this handicap. The fast dynamics of the ORs allowed the robot keep the target within its FOV and served to determine the eye-in-robot's angular position, which unbiased the rate gyro. For this purpose, a new approach to "unbias" the rate gyro was developed, in which the visual loop was used to assess the drift.

In our approach, the accelerometers are replaced by the proprioceptive measurement of the gaze orientation (the eye-in-robot) which plays the same role. In this case, the eye behaves as an inclinometer relative to the target direction. We have shown that the eye is an efficient means to estimate the complete attitude without need of accelerometer. The eye can provide also an accurate and unbiased position estimation of the robot's position, assuming the altitude is known.

The method presented here could be extended in future studies to the stabilization of a robot around the pitch axis. Further research will focus on designing a complete oculomotor system with 3 degrees of freedom and means of stabilizing a hovering autonomous quadrotor with 6 degrees of freedom. However, the use of a decoupled eye goes far beyond the stabilization of a hovering robot, and this development opens promising lines of approach for designing new methods of controlling the 3-D position of a robot by anchoring its gaze on specific objects of interest.

Acknowledgements. The authors would like to thank Robert Mahony for fruitful theoretical discussions on the decoupled eye. This work was supported by the CNRS, the University of the Mediterranean and the French National Research Agency (ANR) with the EVA project.

Appendix

Table 1 Simulation parameters

Parameter	Description	Value	Unit
H	Altitude of the hovering robot	2.0	m
L	Half span of the robot	0.15	m
m	Mass of the robot	0.1	kg
I_z	Inertia momentum	2.0×10^{-5}	$kg.m^2$
K_{vx}	Flapping coefficient	0.5	$N.s.m^{-1}$
FOV	Field of View	±5	$^\circ\ (deg)$
d	Visual sensor's latency	10	ms
G_{opt}	Optic sensor's gain	1	-
τ_{hall}	Angle sensor's time constant	1	ms
τ_{gyr}	Rate gyro's time constant	4.3	ms
τ_{mot}	Propeller's time constant	20	ms
τ_{eye}	Eye's time constant	10	ms

Table 2 Sensors and actuators characteristics

Sensors			
	Visual sensor	Angle sensor	Rate gyro
Transfer function	$G_{opt}(s)=G_{opt}e^{-ds}$	$G_{hall}(s)=\frac{1}{1+\tau_{hall}s}$	$G_{gyr}(s)=\frac{1}{1+\tau_{gyr}s}$
Resolution	0.1 [°]	0.1 [°]	1 [°.s^{-1}]
Sample frequency	40 [Hz]	1 [kHz]	1 [kHz]
Noise amplitude	± 0.1 [°]	± 1 [°]	±5[°.s^{-1}]
Actuators			
	Propeller motor	Eye motor	
Transfert function	$G_{mot}(s)=\frac{1}{1+\tau_{mot}s}$	$G_{eye}(s)=\frac{1}{1+\tau_{eye}s}$	
Rate limiter	-	1000 [°.s^{-}1]	

References

1. Abdelkrim, N., Aouf, N., Tsourdos, A., White, B.: Robust nonlinear filtering for ins/gps uav localization. In: 2008 16th Mediterranean Conference on Control and Automation, pp. 695–702 (2008), doi:10.1109/MED.2008.4602149
2. Bachrach, A., He, R., Roy, N.: Autonomous flight in unknown indoor environments. International Journal of Micro Air Vehicles, 1756-8293, 217–228 (2010), doi:10.1260/175682909790291492
3. Ballard, D.H.: Animate vision. Artificial Intelligence 48(1), 57–86 (1991), doi:10.1016/0004-3702(91)90080-4
4. Boeddeker, N., Kern, R., Egelhaaf, M.: Chasing a dummy target: smooth pursuit and velocity control in male blowflies. Proc. R. Soc. Lond. B 270, 393–399 (2003)
5. Castellanos, J., Lesecq, S., Marchand, N., Delamare, J.: A low-cost air data attitude heading reference system for the tourism airplane applications. In: 2005 IEEE Sensors, p. 4. IEEE (2005)
6. Collett, T.S., Land, M.F.: Visual control of flight behaviour in the hoverflysyritta pipiens l. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 99(1), 1–66 (1975)
7. Grzonka, S., Grisetti, G., Burgard, W.: Towards a navigation system for autonomous indoor flying. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 2878–2883 (2009), doi:10.1109/ROBOT.2009.5152446
8. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.M., Hirzinger, G., Rus, D.: Energy-efficient autonomous four-rotor flying robot controlled at 1 khz. In: 2007 IEEE International Conference on Robotics and Automation, pp. 361–366 (2007), doi:10.1109/ROBOT.2007.363813
9. Hengstenberg, R.: Mechanosensorey control of compensatory head roll during flight in the blowfly calliphora erythrocephala meig. Journal of comparative Physiology A 163, 151–165 (1988)
10. Juston, R., Viollet, S.: A miniature bio-inspired position sensing device for the control of micro-aerial robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal (2012) (in press)
11. Kerhuel, L., Viollet, S., Franceschini, N.: Steering by gazing: An efficient biomimetic control strategy for visually guided micro aerial vehicles. IEEE Transactions on Robotics 26(2), 307–319 (2010), doi:10.1109/TRO.2010.2042537
12. Kerhuel, L., Viollet, S., Franceschini, N.: The vodka sensor: A bio-inspired hyperacute optical position sensing device 12(2), 315–324 (2012), doi:10.1109/JSEN.2011.2129505
13. Kim, J., Kang, M.S., Park, S.: Accurate modeling and robust hovering control for a quadrotor vtol aircraft. Journal of Intelligent & Robotic Systems 57, 9–26 (2010), 10.1007/s10846-009-9369-z
14. Mahony, R., Hamel, T., Pflimlin, J.M.: Nonlinear complementary filters on the special orthogonal group. IEEE Transactions on Automatic Control 53(5), 1203–1218 (2008), doi:10.1109/TAC.2008.923738
15. Manecy, A., Viollet, S., Marchand, N.: Bio-Inspired Hovering Control for an Aerial Robot Equipped with a Decoupled Eye and a Rate Gyro. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal (2012) (in press)
16. Miles, F.: The neural processing of 3-d visual information: evidence from eye movements. European Journal of Neuroscience 10(3), 811–822 (1998), doi:10.1046/j.1460-9568.1998.00112.x

17. Mondragón, I., Olivares-Méndez, M., Campoy, P., Martínez, C., Mejias, L.: Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. Autonomous Robots 29, 17–34 (2010), http://dx.doi.org/10.1007/s10514-010-9183-2, doi:10.1007/s10514-010-9183-2
18. Mori, R., Hirata, K., Kinoshita, T.: Vision-based guidance control of a small-scale unmanned helicopter. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, pp. 2648–2653 (2007), doi:10.1109/IROS.2007.4399623
19. Nemra, A., Aouf, N.: Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering. IEEE Sensors Journal 10(4), 789–798 (2010), doi:10.1109/JSEN.2009.2034730
20. Preuss, T., Hengstenberg, R.: Structure and kinematics of the prosternal organs and their influence on head position in the blowfly calliphora erythrocephala meig. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 171(4), 483–493 (1992)
21. Robinson, D.: The mechanics of human smooth pursuit eye movement. The Journal of Physiology 180(3), 569–591 (1965)
22. Rondon, E., Garcia-Carrillo, L.R., Fantoni, I.: Vision-based altitude, position and speed regulation of a quadrotor rotorcraft. In: Conference on Intelligent Robots and Systems (2010)
23. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained mav. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 20–25 (2011), doi:10.1109/ICRA.2011.5980357
24. Stark, L., Young, L.: Defining biological feedback control systems. Ann. N. Y. Acad. Sci. 117, 426–444 (1964)
25. Strausfeld, N., Seyan, H., Milde, J.: The neck motor system of the fly calliphora erythrocephala. 1. muscles and motor neurons. J. Comp. Physiol. A 160, 205–224 (1987)
26. Viollet, S., Franceschini, N.: Super-accurate visual control of an aerial minirobot. In: Autonomous Minirobots for Research and Edutainment, AMIRE (2001)
27. Viollet, S., Franceschini, N.: A high speed gaze control system based on the vestibulo-ocular reflex. Robotics and Autonomous Systems 50, 147–161 (2005)
28. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. Journal of Field Robotics 28(6), 854–874 (2011), http://dx.doi.org/10.1002/rob.20412, doi:10.1002/rob.20412
29. Wendel, J., Meister, O., Schlaile, C., Trommer, G.F.: An integrated gps/mems-imu navigation system for an autonomous helicopter. Aerospace Science and Technology 10(6), 527–533 (2006), http://www.sciencedirect.com/science/article/pii/S1270963806000484, doi:10.1016/j.ast.2006.04.002
30. Wenzel, K.E., Rosset, P., Zell, A.: Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. Selected Papers from the 2nd International Symposium on UAV, pp. 297–311 (2009)
31. Westheimer, G.: Visual hyperacuity. In: Ottoson (ed.) Sensory Physiology, vol. 1. Springer, Berlin (1981)
32. Zeil, J., Boeddeker, N., Hemmi, J.: Vision and the organization of behaviour. Curr. Biol. 18(8), R320–R323 (2008), doi:10.1016/j.cub.2008.02.017
33. Zhang, T., Kang, Y., Achtelik, M., Kuhnlenz, K., Buss, M.: Autonomous hovering of a vision/imu guided quadrotor. In: International Conference on Mechatronics and Automation (2009)

Flight Guidance and Control of a Tethered Glider in an Airborne Wind Energy Application

Sören Sieberling

Abstract. One of the concepts of an upcoming technology known as airborne wind energy is the pumping kite system. The pumping kite system uses a conventional gliders to fly highly dynamic crosswind patterns downwind of a generator to which it is connected by a tether to harvest wind energy. Operating the pumping kite system requires a novel view on conventional flight control. A tether based reference frame is introduced that in effect decouples the longitudinal and lateral motion which can thereby be designed independently and thus allowing the highly dynamic motion of the glider to be controlled through simple control schemes. Furthermore the longitudinal motion is constrained through the tether of which the tangential velocity is controlled by the generator providing an additional control input besides the elevator to control longitudinal motion. Flight tests demonstrate that using the tether based flight control system reasonably simple and commonly used control methods provide satisfactory flight performance.

1 Introduction

Harvesting wind energy is being investigated for several decades with many interesting outcomes as a result, [6]. One of the concepts introduced by [3] in the early eighties, describes a technique using a tethered glider that drives a generator on the ground through tension in the tether. [3] refers to this concept as lift power, also known as pumping-kite power. Today the pumping-kite system represents a subset of airborne wind energy, which has been a growing industry in the past decade with the sole objective of outperform conventional wind turbines in terms of cost per unit of energy.

The operational principle of the pumping kite system is discussed in Sec. 2, which is followed by Sec. 3 providing the physical framework. Sec. 4 presents the flight

Sören Sieberling
Ampyx Power B.V., Lulofsstraat 55 - 13, 2521AL Den Haag, The Netherlands
e-mail: sieberling@ampyxpower.com

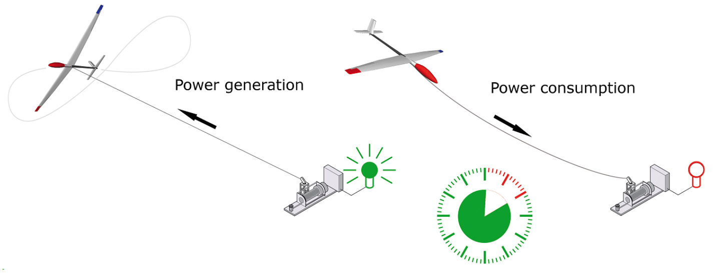


Fig. 1 The pumping kite system during power generation (left) and during reset (right).

control system of the pumping kite system split into longitudinal and lateral control. Sec. 5 demonstrates some results from test flights, followed by the conclusions in Sec. 6.

2 Operational Principle of the Pumping Kite System

To convert wind into electrical energy the pumping kite system consists of an glider, a generator and a tether connecting the two. At the generator side the tether is wound on a drum that connects to the generator (together called winch). Lift is generated by flying crosswind patterns (similar to kites on a beach) downwind of the generator and transfered to the ground through tension in the tether that is experienced as torque by the generator.

The conversion of wind into electrical energy only takes place when the tether is unwound from the drum such that the lift can actually do work. Because of the tether being unwound, the operation of the pumping kite system consists of two phases; power generation and tether retrieval (reset), Fig. 1. When the maximum tether length is reached the glider changes its flight path from crosswind flight to flying straight toward the generator, while the tether is wound again. This tether retrieval maneuver results in low tether tension due to low drag, making the power consumption during tether retrieval significantly less than the power generated during the power generation phase at high tether tension due to high lift. This results in net power at the end of each power cycle.

One advantage of the pumping kite system compared to conventional wind energy becomes apparent when considering the wind power available, which is a function of ρ the air density and V_{w} the wind speed, Eq. 1.

$$P_{\mathrm{w}} = \frac{1}{2}\rho V_{\mathrm{w}}^3 \tag{1}$$

A specific device can generate more power than another by being located in an area with more wind (higher wind power density). The operational altitude of the pumping kite system is between 400 and 500 m. When assuming that the wind profile increases logarithmic with altitude [7, 4], wind power at for example 80 m hub height compared to 400 m doubles (above agricultural land). In terms of wind speeds this translates roughly into 25% wind speed increase.

A second advantage is that wind turbines are not expected to grow in the near future [5]. Consider that power is cubically related to airspeed. With airspeed over a wind turbine blade increasing linearly, it follows that most power in a wind turbine is generated by its blade tips. When assuming that the power generated by a section of the wind turbine blade is only a function of airspeed, hence neglecting the effect of airfoil and chord variation, almost 60% of the power would be generated by the outer 25% of the blades. In effect the pumping kite system could be compared to this outer section of the wind turbine blades. The difference to wind turbines is that it does not require a large tower and can operate with only one 'blade', thereby removing one of big upscaling obstruction of conventional wind turbines.

3 Physical Framework

The operation of the pumping kite system and the possible power output as a function of the wind speed is best described by basic aerodynamic equations and derivatives thereof, [3]. For simplicity the influence of mass on the system is neglected in these derivations. The glider is assumed to have no roll angle compared to the tether. And the wind is assumed to have a constant velocity, (V_{w}), parallel to the ground plain.

The lift generated by the glider is given by Eq. 2.

$$L = \frac{1}{2}\rho V_{\mathrm{TAS}}^2 S C_{\mathrm{L}} \qquad (2)$$

With L the lift, V_{TAS} the true airspeed of the glider, S the wing surface area and C_{L} the lift coefficient. The drag of the system consists of two components, glider drag and tether drag. The drag of the glider is given by Eq. 3.

$$D_{\mathrm{ac}} = \frac{1}{2}\rho V_{\mathrm{TAS}}^2 S C_{\mathrm{D}} \qquad (3)$$

With D_{ac} the glider drag and C_{D} the drag coefficient, approximated by

$$C_{\mathrm{D}} = C_{\mathrm{D}_0} + \frac{C_{\mathrm{L}}^2}{\pi A R e} \qquad (4)$$

Where C_{D_0} represents the zero lift drag coefficient and the remaining term the induced drag with e the span efficiency factor and AR the aspect ratio. The tether drag is approximated by assuming that the tether is straight, [2]. Since the lift of the glider is about one order of magnitude larger than the tether drag, the resulting errors are negligible. The effect of the true wind on the tether drag is also neglected. The drag of an infinitesimal section of the tether ds is given by Eq. 5.

$$\mathrm{d}D = \frac{1}{2}\rho V_{\mathrm{ds}}^2 C_{\mathrm{D_c}} t \,\mathrm{d}s \qquad (5)$$

With $V_{\mathrm{d}s}$ the speed of an infinitesimal section of the tether, t the tether thickness and $C_{\mathrm{D_c}}$ the tether drag coefficient. By assuming a straight line (the speed changes linearly with position on the tether) the sectional speed is approximated by Eq. 6

$$V_{\mathrm{ds}} = \frac{s}{l} V_{\mathrm{TAS}} \tag{6}$$

With l the tether length. Substituting Eq. 6 into Eq. 5 and equating moments around the generator of the tether drag (integrated over the tether length) on one hand and a resulting force on the glider, the resulting tether drag is derived, Eq. 7, [2].

$$D_{\mathrm{c}} l = \int_0^1 s \mathrm{d}D = \int_0^1 s \frac{1}{2} \rho \left(\frac{s V_{\mathrm{TAS}}}{l} \right)^2 C_{\mathrm{D_c}} t \mathrm{d}s = \frac{1}{8} \rho V_{\mathrm{TAS}}^2 t l^2 C_{\mathrm{D_c}} \tag{7}$$

The lift to drag ratio (G) of the pumping kite system is then obtained by dividing Eq. 2 by Eq. 3 and Eq. 7 resulting in Eq. 8.

$$G = \frac{C_{\mathrm{L}}}{C_{\mathrm{D}} + \frac{t l C_{\mathrm{D_c}}}{4S}} \tag{8}$$

Where the right hand component of the denominator will also be referred to as effective tether drag coefficient. In a massless system, the system lift to drag ratio equals the ratio of forward and upward airspeed (Fig. 2), or in other words the lift to drag ratio describes the forward speed of the glider as a function of wind speed, tether angles and tether speed, Eq. 9.

$$V_{\mathrm{p}} = G (V_{\mathrm{w}} \sin \Theta \cos \Phi - V_{\mathrm{t}}) = G V_{\mathrm{eff}} \tag{9}$$

With V_{p} the forward speed, Θ the in the wind plane rotation of the tether (Fig. 3), Φ the out of the wind plane rotation of the tether (Fig. 3) and V_{t} the tether speed. Note that this equation holds only for a massless system, or in other words when the system is in equilibrium.

For rigid wing kites, the difference between V_{p} and V_{TAS} becomes negligible due to the high system lift to drag ratio, Eq. 10.

$$V_{\mathrm{TAS}} \approx V_{\mathrm{p}} \tag{10}$$

Furthermore, when assuming a high lift to drag ratio, the tether tension is approximately equal to the lift. With the tension then given, the resulting power during the power generating phase is computed by Eq. 11.

$$P = \eta_{\mathrm{m}} T V_{\mathrm{t}} \tag{11}$$

With η_m the mechanical efficiency of the motor and other winch components. In the derivation above the tether speed has not been specified. [3] however demonstrates that for maximum power production the tether speed, should be 1/3 of the wind speed component perpendicular to the glider, thus $V_{\mathrm{w}} \sin \Theta \cos \Phi / 3$.

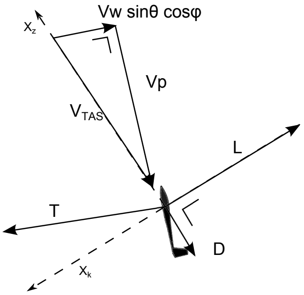


Fig. 2 Force and speed diagram during power production for forces in equilibrium, showing identical ratios between L/D and $V_p/V_w \sin\Theta \cos\Phi$. The dashed lines indicate the kinematic reference frame (X pointing in the direction of the airspeed).

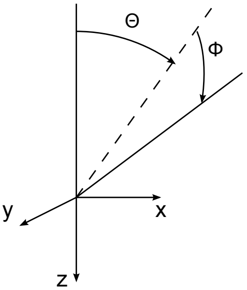


Fig. 3 Definition of tether angles with respect to wind direction blowing along the X-axis. Θ in the wind plane rotation and Φ out of the wind plane rotation.

Note that these derivations are a simple means to analyzing the pumping kite systems characteristics. The biggest assumption is that the system is massless. In reality the glider is constantly maneuvering to stay inside the wind window and thus constantly accelerating hence not being in equilibrium. The true motion is governed by the equations of motion conventional to aircraft, Eqs. 12.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + g\begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{12a}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = J^{-1}\left(\begin{bmatrix} L \\ M \\ N \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left\{ J \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\}\right) \tag{12b}$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{12c}$$

$$\begin{bmatrix} V_\mathrm{N} \\ V_\mathrm{E} \\ V_\mathrm{D} \end{bmatrix} = T_\mathrm{nb}\begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} V_{\mathrm{wind}_\mathrm{N}} \\ V_{\mathrm{wind}_\mathrm{E}} \\ V_{\mathrm{wind}_\mathrm{D}} \end{bmatrix} \tag{12d}$$

With u, v, w velocity along respectively the body x, y, z axis, p, q, r rotational rates along respectively the body x, y, z axis, ϕ, θ, ψ the euler angles, V_N, V_E, V_D velocity respectivily north, east, down, g the gravitational constant, X, Y, Z, aerodynamics and tether forces along respectively x, y, z axis, L, M, N aerodynamic and tether moments around respectively the x, y and z axis and T_nb the transformation matrix

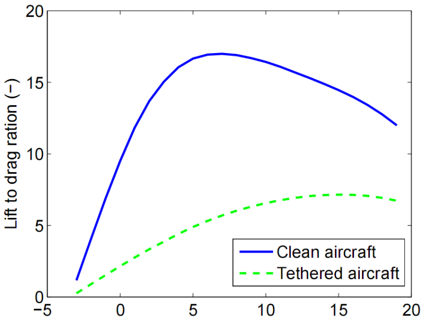


Fig. 4 The lift to drag ratio of a clean free flying glider vs angle of attack compared to the lift to drag ratio of a tethered glider with a fixed tether length of 400m vs. angle of attack.

from the body to the inertial reference frame, which can be found in any common book on aircraft dynamics.

When using the derivation as a qualitative means to describe the pumping kite system scaling, two observations can be made, [8]:

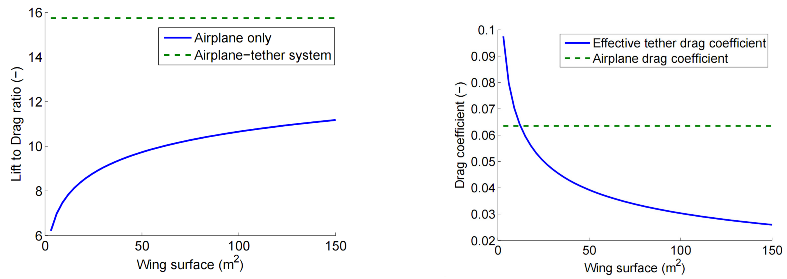
(A) From the discussion above it is concluded that when flying at higher lift to drag ratios the power output grows. Since not only the glider lift to drag ratio is of concern but the lift to drag ratio of the system, higher lift to drag ratio within operational flight envelopes almost always correspond to higher lift and therefor higher angle of attack, Fig. 4.

As is also stated by Eq. 9, higher lift to drag ratio also corresponds to higher true airspeed, which results in the unconventional flight characteristic for a tethered glider that the airspeed is raised by pitching up instead of pitching down.

(B) For free flying glider the lift to drag ratio provides a means to compare glider performance because it is dimensionless. For tethered glider the system lift to drag ratio is however not independent of dimension, meaning that system lift to drag ratios of differently sized systems should not be compared without corrections.

The scaling dependency of the system lift to drag ratio is caused by the tether dimensioning. When expressing the scaling of glider in terms of wing surface, twice as much wing surface results in twice as much lift and drag, hence tension. Twice the amount of tension requires twice the amount of tether, hence the tether cross-sectional surface doubles. Since the system lift to drag ratio does not depend on tether cross section but on tether diameter, the effective tether drag coefficient will grow with the square root of the tether cross section, hence proportional to the square root of the wing surface. Therefore the relative contribution of the tether drag becomes smaller.

Fig. 5(a) illustrates the system lift to drag ratio of differently sized pumping kite systems having identical aerodynamic characteristics in terms of glider lift and coefficients. Furthermore the tether length is identical for different sizes. Sizing of



(a) Lift to drag ratio

(b) Drag coefficient

Fig. 5 System lift to drag ratio and total drag coefficient as a function of glider dimension

the tether thickness is based on the tension at a fixed airspeed and lift. In reality the tether length will grow slightly for increasing systems as will airspeed, which would have a softening effect on the differences in system lift to drag ratio. Fig. 5(b) illustrates the glider and tether drag coefficients, which illustrates that the system lift to drag ratio grows for aerodynamically identical glider as the system is scaled up, thus tether drag is a bigger problem for smaller systems.

4 Guidance and Control

Compared to most conventional glider, the maneuvering of the pumping kite can be perceived as aggressive and nonlinear. In only a few seconds the roll angle changes from plus to minus 60°, while the heading is changing direction 180°and the pitch angle is going up and down from 60 to -20°.

When however changing the control reference to a tether based reference frame, the same maneuvers become rather mild and practically speaking linear in the sense that dynamic coupling can be neglected and longitudinal and lateral control can be separated. The tether based ’Euler’ angles then vary only up to 20°for the roll and pitch angle.

The transformation from inertial reference frame to the tether based reference frame is defined by:

1. Rotation around the earth fixed Z-axis by Ψ, the wind direction, with corresponding direction cosine matrix $T_{\mathrm{z}_{(\psi)}}$
2. Rotation around the Y-axis by $-\Theta$, with corresponding direction cosine matrix $T_{\mathrm{y}_{(-\Theta)}}$
3. Rotation around the X-axis by Φ, with corresponding direction cosine matrix $T_{\mathrm{x}_{(\Phi)}}$

$$C_{\mathrm{ti}} = T_{\mathrm{x}_{(\phi)}} T_{\mathrm{y}_{(-\theta)}} T_{\mathrm{z}_{(\psi)}} \tag{13}$$

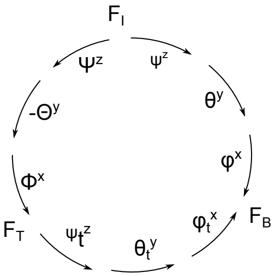


Fig. 6 Reference frame transformation summary. Reference frames are indicated by the capital letter F, with the subscript indicating the specific frame. Superscripts are used to indicate the axis of rotation.

With C_{ti} the direction cosine matrix mapping inertial coordinates into tethered coordinates and the other way around by taking the transpose, $C_{\mathrm{it}} = C_{\mathrm{ti}}^{\mathrm{T}}$. When introducing the conventional axis transformation from the inertial reference frame to the body fixed reference frame by means of the Euler angles (roll (ϕ), pitch (θ) and yaw (ψ)) as C_{bi}, the direction cosine matrix to the body frame from the tethered frame is given by Eq. 14.

$$C_{\mathrm{bt}} = C_{\mathrm{bi}} C_{\mathrm{it}} \tag{14}$$

The tethered Euler angles are then derived consequently as in Eqs. 15. The transformations are summarized in Fig. 6.

$$\phi_{\mathrm{t}} = \tan^{-1}\left(C_{\mathrm{bt}_{(2,3)}} / C_{\mathrm{bt}_{(3,3)}}\right) \tag{15a}$$

$$\theta_{\mathrm{t}} = \sin^{-1}\left(-C_{\mathrm{bt}_{(1,3)}}\right) \tag{15b}$$

$$\psi_{\mathrm{t}} = \tan^{-1}\left(C_{\mathrm{bt}_{(1,2)}} / C_{\mathrm{bt}_{(1,1)}}\right) \tag{15c}$$

4.1 Longitudinal Control

As is concluded in section 3, the crux for high power outputs is flying at high lift coefficients. Since the lift coefficient directly relates to angle of attack, this translates into a longitudinal control objective, being angle of attack tracking, which is performed by the elevator. Angle of attack control alone is however not sufficient to control motion along the tether based Z-axis.

Another important, but unconventional factor to the longitudinal control of the pumping kite system is the constrained imposed by the tether. The motion along the direction of the tether (heave) is defined by the rotational velocity of the winch, which thereby becomes a second longitudinal control input. It is important to view the control on the glider and the control of the winch as one.

Both control inputs have effects on the tether tension. Lowering the angle of attack lowers the lift generation of the wing (hence tension), thereby also lowering

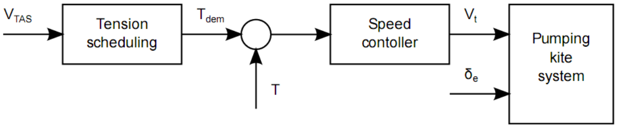


Fig. 7 Block diagram indicating the functioning of the longitudinal control scheme during power generation

the lift to drag ratio of the system (Fig. 4), which lowers the airspeed (which again lowers the tension). Raising the winch speed, lowers the effective wind at the plane, which lowers the airspeed and thereby the tension. It is not hard to understand that both control inputs in effect influence the same parameters, which can easily grow into instabilities. What makes this control problem more complicated is that communication between winch and glider is over radio suffering from transmission delays and that the winch has implementation delays orders of magnitude larger than the glider implementation delays.

4.1.1 Power Generation Phase

Mostly because of the system latencies the longitudinal control is chosen to use the elevator passively (in a fixed position comparable to a flap) and have the winch speed control the tether tension, by varying reel out speed. Simulations indicate that angle of attack control would become possible when latencies are reduced to about 50ms. Note that this implies not flying at the optimal $V_w/3$ as derived by [3]. By fixing the elevator in effect the angle of attack and thus the lift coefficient is set (and thus the lift to drag ratio). The winch is programmed to control the tension in the tether by varying the reel out speed, using a PI controller. The tension demand is scheduled with scheduled tension demand versus airspeed. The tension demand is scheduled such that at low airspeeds the tension demand rises, while at high airspeeds it drops. A rising tension demand will result in slower winch speed and thereby raising the effective wind experienced by the glider and visa versa.

Note that the winch control thereby does depend on the glider measurement by requiring information on the true airspeed. As mentioned before the communication between glider and winch uses radio and contains latencies, which is why the angle of attack is not controlled actively. Compared to angle of attack changes, the airspeed however changes orders of magnitude slower since physically it is a derivative of higher or lower lift to drag ratio in tethered flight. Therefor the system latencies are acceptable in this control architecture.

4.1.2 Reset Phase

As the tether length reaches its maximum the reset phase is triggered and activates a second set of longitudinal controls. In this setup the winch becomes the passive component by simply setting the winch speed to reeling in at maximum speed, thereby making the time for resetting the system as small as possible. The elevator is

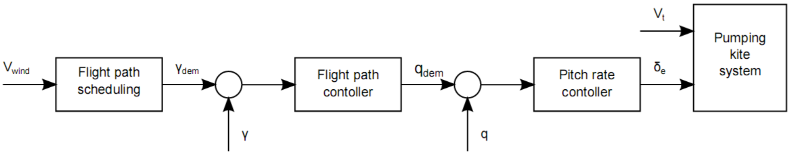


Fig. 8 Block diagram indicating the functioning of the longitudinal control scheme during system reset

controlled to maintain a specific flight path angle that is derived from a sink demand setting, which is scheduled against wind speed.Since the winch speed in this phase is set, so is the ground speed. Which implies that the true airspeed, and thus the drag, rises with increasing wind. The scheduling is therefore chosen to have the reset phase consist of a steeper dive for higher wind speeds such that gravity compensates for the glider drag as much as possible, yet never makes it exceed the tether speed.

4.1.3 Phase Transitions

The change between control strategies is instant, therefore the plane enters the reset phase with a fast pitch down maneuver. The tether tension is thereby suddenly lowered and possibly the tether gets slack for a brief period. Shortly after the winch will however pick up the pace and straighten out the tether.

The transition back into the power generation flight path requires extra attention. The plane is flying into the wind at ground speeds roughly equal to the winch speed, such that the effective wind speed is that of the true wind plus that of the tether. The tether tension is low and approximately a full order of magnitude lower than during power generation. If the winch would respond too slow to the plane flying back into the pattern, considering the equations of Sec. 3, unfeasible airspeeds would arise. On the other hand if the winch responds too fast, the tension will drop completely and the plane would not make the turn back into the pattern at all, or it would build up momentum and at some point instantly tense the tether resulting in high shock loads. In other words the tether may not get slack, but it can also not build up tension, making the transition back into the pattern a delicate maneuver.

This creates hard requirements on the synchronisation, making the radio latencies unacceptable. For ’communication’ in this situation the tether tension is therefor used. Yet switching back to the conventional tension controller is not an option since it is tuned to operate at reel out speeds during power generation and besides it would not be sensitive enough to responds to tension changes in the order of only 100N. Another simple controller is therefore designed, controlling the winch acceleration directly. The tension error that is measured with respect to a set reference tension, in the order of magnitude of the tension during the reset phase, is multiplied by a gain to yield the winch acceleration. This method, in combination with the fixed elevator, has proven to be fast and provides sufficient margin for different wind speeds.

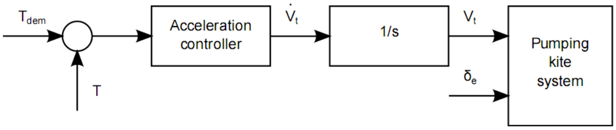


Fig. 9 Block diagram indicating the functioning of the longitudinal control scheme during transition into the power generation phase

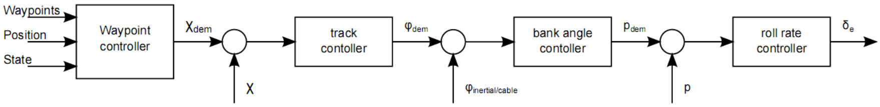


Fig. 10 Block diagram indicating the functioning of the lateral control scheme

4.2 Lateral Control

Essentially the lateral control is not much different as for conventional glider. The main difference compared to free flying glider is that the lateral control is not defined with respect to the inertial reference frame, but with respect to tether frame as defined in the beginning of this section.

The flight path is defined by waypoints in the spherical tether coordinates, Fig. 3, making the waypoints independent of tether length and wind direction. For flight guidance these spherical coordinates are mapped into the tether reference frame, with the origin in the generator, thereby generating waypoints in Cartesian coordinates. The tether coordinates of the flight path thereby do depend on wind direction and tether length (R), Eq. 16.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{i}} = C_{\mathrm{it}} \begin{bmatrix} 0 \\ 0 \\ R \end{bmatrix}_{\mathrm{t}} \tag{16}$$

Different from the longitudinal control, only one controller governs the actual waypoint tracking. The lateral controller derives the closest point on the set flight path from its current position. Taking the closest point as a starting point a variable look ahead distance (scheduled vs. tether length) is travelled over the flight path to determine a so called look ahead point. The direction toward this look ahead point is the track demand. A controller transforms the track error into a roll angle reference and the error thereof is by means of a PD controller transformed into a roll rate, Fig. 10.

The difference between phases lies in the waypoint scheduling governed by a waypoint controller and the roll angle selection (Euler vs tethered Euler). During power generation the gliders flight path is a lying figure of 8, Fig. 11. During the system reset it is a straight line starting at the location where the corresponding phase was activated and ending in the coordinates of pattern reentry.

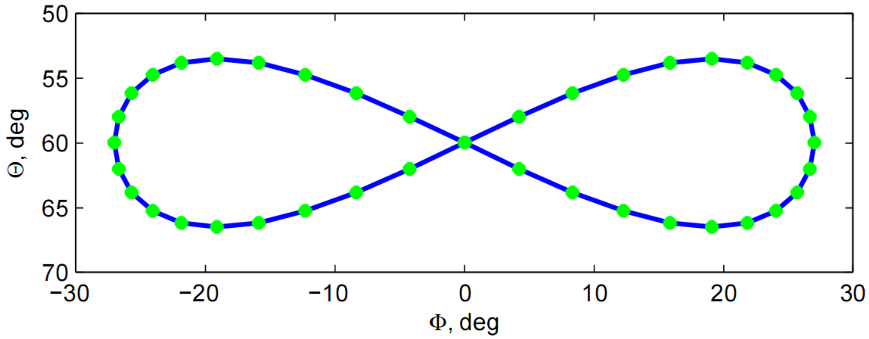


Fig. 11 Scheduled waypoints during the power generation phase in tether angles

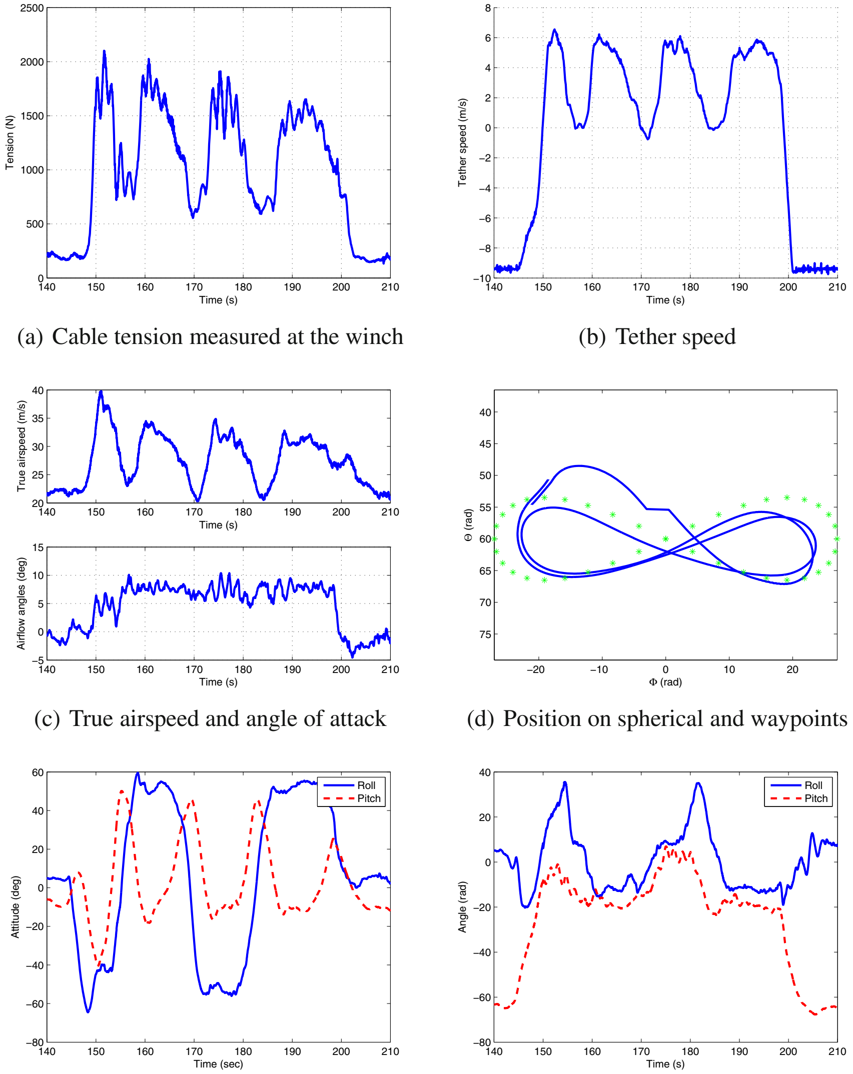
4.3 State Machine

Since the maneuvering of the pumping kite system throughout an operational cycle, is too diverse to be governed by a single controller and or guidance scheme (crosswind pattern flying during power generation vs. straight descending flight during system reset), a higher level supervision is required to switch between controllers. For the pumping kite system this task is fulfilled by a state machine. Depending on this state different controllers are active and others are reset.

Each state has a predefined set of criteria (flags, demands) that must be met to transition into a next state. Depending on the state, one or several transitions are possible. Furthermore in each state an abort can be triggered when exceeding the margins to the flight envelope, which triggers a completely independent control system with its own state machine and consequent states and controllers to take over. Table 1 presents an overview of the relevant states for power generation.

Table 1 Pumping kit states used in power generation and corresponding criteria to complete the task of that state and trigger transition into a next state

State	Condition	Next state
Takeoff	Completed when reaching a set altitude and climb rate	Climb
Climb	Completed when reaching a set altitude	Pattern entry
Pattern entry	Completed when within a set range to the pattern	Power generation
Power generation	Completed when reaching a dynamically set tether length	Pattern exit
Pattern exit	Completed when reaching a set waypoint	Reset
Reset	Completed when reaching a minimum tether length	Pattern re-entry
Pattern re-entry	Completed when within a set range to the pattern	Power generation



(a) Cable tension measured at the winch

(b) Tether speed

(c) True airspeed and angle of attack

(d) Position on spherical and waypoints

(e) glider Euler angles

(f) Tethered Euler angles

Fig. 12 Test results of one power cycle at 7 m/s wind speed, measured at 6.5 m reference height. Starting point of the graphs is the reset phase, followed by power generation and ending in another reset phase.

5 Flight Performance of the Control System in Test Flights

The performance of the pumping kite system is illustrated by Figs. 12. The graphs present one complete power cycle starting in a reset phase, followed by a complete power generation phase and ends in the middle of another reset phase. The tether tension and speed, the true airspeed and angle of attack, Euler angles and tether based position and attitude are presented.

Comparing tethered Euler angles to the conventional one illustrates the linearizing effect mentioned in Sec 4. The tethered pitch angle varies approximately 20°from -20°to 0°and the tethered roll angle varies within -10°to 30°. The large negative pitch angle that is observed is due to the system being reset, where the plane flies toward the winch. With the tether being much longer than the flight altitude, this results in large negative pitch angles.

The angle of attack is tracked satisfactory in the beginning (to within plus or minus 1°). In the middle some oscillations are observed that however dampen out toward the end of the phase. Note that the oscillations in angle of attack have a strong correlation to the tether tension, but much less to the glider airspeed.

The flight path tracking performance is good. What is observable is the plane cutting the turns and staying within the flight path, which is a consequence of the waypoint tracking algorithm fixing the track demand to a location ahead on the set flight path. The strong overshoot is the spherical representation of the reset phase, where the plane flies toward the winch but does not sink as fast as the pattern would, resulting in a high entry into the pattern.

6 Conclusions

Through choosing the tether based reference frame and carefully selecting control variables, simple control schemes are well capable of flying the highly aggressive patterns of the pumping kite system. In lateral direction the glider is controlled by means of a waypoint controller that produces track demands. Tracking errors generate roll angle demands, which again are used to create roll rate demands each through linear controllers. The longitudinal motion is controlled by fixing the elevator and thereby fixing the lift coefficient and system lift to drag ratio in combination with a tension controller determining the tether velocity.

References

1. Diehl, M., Houska, B.: Windenergienutzung mit schnell fliegenden Flugdrachen: eine Herausforderung für die Optimierung und Regelung - Wind Power via Fast Flying Kites: a Challenge for Optimization and Control. at-automatisierungstechnik (2009)
2. Houska, B., Diehl, M.: Optimal Control for Power Generating Kites. In: Proc. 9th European Control Conference (2007)
3. Loyd, M.L.: Crosswind Kite Power. J. of Energy (1980)
4. Manwell, J.F., McGowan, J.G., Rogers, A.L.: Wind Energy Explained Theory, Design and Application. John Wiley and Sons, Ltd. (2002)

5. Thresher, M., Robinson, M., Veers, P.: To Capture the Wind. IEEE Power and Energy Magazine (2007)
6. Williams, P., Lansdorp, B., Ockels, W.: Optimal Cross-Wind Towing and Power Generation with Tethered Kites. J. of Guidance, Control and Dynamics (2008)
7. Wortman, A.J.: Introduction to Wind Turbine Engineering. Butterworth (1982)
8. Sieberling, S., Ruiterkamp, R.: The PowerPlane an Airborne Wind Energy System Conceptual Operations. In: Proc. 11th AIAA ATIO Conference (2011)

Lateral Fly by Wire Control System Dedicated to Future Small Aircraft

Matthias Heller, Thaddäus Baier, and Falko Schuck

Abstract. Compared to common transport aircraft (airliners), it is fact that the General Aviation (GA) sector exhibits a significant higher rate of accidents. Even though the sources are manifold, two main reasons may be identified. First, General Aviation Pilots generally have a relative low training level and small number of flight hours compared to airliner pilots and thus, their flight experience and hazard awareness is consequently limited. The second reason is, that recent transport aircraft feature a significant higher technical standard possessing various beneficial pilot assistant systems supporting the pilot to fly the aircraft safely at the same time reducing pilot's workload extensively. The most vital assistant systems, well-known as Fly-by-Wire Flight Control Systems (FbW FCS), provide directly the appropriate control deflections according to the pilot's commands and (measured) flight condition and thus are capable to assume important safety enhancing tasks. In addition to ensuring excellent and homogenized flying/handling qualities along the whole envelope, they offer functionalities like pilot input monitoring, provision of warnings plus active envelope protection yielding a substantial increase of passenger, crew and aircraft safety towards the key objective "carefree handling". Unfortunately, this valuable safety increase did not find its way into the general aviation sector although it is standard in current transport planes and modern business jets. This is due to the tremendous cost of typical Fly-by-Wire control technology always requiring complex redundancy and reversionary systems to fulfill the strict certification requirements. However, in order to accomplish an

Matthias Heller
Rudolf Diesel Industry Fellow at TUM-IAS
Garching, Germany
email: matthias.heller@tum.de

Thaddäus Baier · Falko Schuck
Institute of Flight System Dynamics (FSD)
and Institute for Advanced Study (IAS),
Technische Universität München (TUM),
D-85748 Garching, Germany
email: {th.baier,schuck}@tum.de

equivalent safety enhancement for GA aircraft and thus to diminish the high accident rates and so to protect human lives, the well-proved beneficial features of active Flight Control Systems have to be made available and affordable for them.

An essential contribution to this subject is the major objective of the ambitious Technology Research Program “Future Small Aircraft (FSA)” of the Austrian aircraft manufacturer Diamond Aircraft Industries in cooperation with the Institute of Flight System Dynamics of the Technische Universität München. Within this joint multinational research program concerning upcoming Future Small Aircraft, (amongst others) the development of an appropriate FbW lateral flight control system is expedited. Although the control law design is primarily aimed for provision of excellent handling qualities and pilot’s assistance, one main focus is also set on the elaboration of special processes, tools and hardware solutions enabling the progression of control algorithms which are perfectly tailored to the specific needs of manufacturers of small and medium-sized planes.

1 Introduction

By comparison of the accident statistics of General Aviation (GA) versus common transport aircraft (airliners) it becomes obvious that the General Aviation sector exhibits a significant higher rate of accidents (Ref. [8]). This fact is not new and even though the sources are manifold, two main reasons may be identified. On the one hand, General Aviation Pilots mostly hold a Private Pilot License (PPL) only and hence, their number of flight hours and thus their experience commonly is considerably limited in contrast to airliner pilots having an Airline Transport Pilot License (ATPL). On the other hand, current (modern) transport aircraft feature a noteworthy higher technical standard providing various beneficial pilot assistant systems in order to support the pilot to fly the aircraft safely and to reduce pilot's workload extensively.

The most important and effective assistant systems, which are well-known as Fly-by-Wire Flight Control Systems (FbW FCS) intervene directly and actively into the aircraft's control and besides improving and homogenizing flying and handling qualities considerably, they offer a wide range of functionalities including pilot input monitoring, provision of warnings plus limitations and advanced protections and hence, increase the passenger, crew and aircraft comfort and safety competently towards the overall ultimate objective "carefree handling".

This valuable increase in safety, which has become standard in current transport planes and modern business jets, unfortunately did not find its way into the general aviation sector due to the tremendous cost of typical Fly-by-Wire control technology always requiring complex redundancy and reversionary systems in order to fulfill the strict certification requirements and specifications. However, in order to achieve a corresponding safety enhancement for GA aircraft and hence to reduce the high accident rates and to protect human lives, the well-proved beneficial functionalities of active Flight Control Systems (active FCS) definitely have to be made available and affordable for them. Particularly, this holds in consideration of the expected significant rise in the number of GA planes and movements (Ref. [10]), which otherwise would come along with a further increase in accidents and victims, which has to be avoided. Consequently, specifically tailored FbW FCS technology suitable and in particular affordable for GA airplanes has to be made available. For instance, recent accomplishments in the area of actuators, sensors and flight control computers (FCC) offer potential to design more cost-effective active assistance systems to be utilized within GA planes in the future.

A considerable contribution to this subject is one major objective of the ambitious Technology Research Program “Future Small Aircraft (FSA)” of the Austrian aircraft manufacturer Diamond Aircraft Industries in cooperation with the Institute of Flight System Dynamics of the Technische Universität München. Within this joint multinational Research program concerning upcoming Future Small Aircraft, first a hybrid control concept for longitudinal dynamics has been proposed, compare Ref. [7], which now is extended by the development of an appropriate lateral flight control system. Although the control law layout is aimed for provision of excellent homogeneous flying/handling qualities and pilots assistance, another main focus is set on the preparation of special processes, tools and hardware requirements/solutions supporting the design of control algorithms which are perfectly adapted to the specific needs of manufacturers of small and medium-sized planes.

Summarized, the development process applied comprises the following main issues:

- Elaboration of an universal controller structure for lateral dynamics which is suitable for all typical airplane configurations (fixed-wing aircraft with empennage) independent whether a full Fly-by-Wire or “Hybrid Flight Control System” will be applied, compare Ref. [7]
- Provision of a layout methodology denoted "Model Reference Direct Eigenstructure Assignment (*MR DEA*)" which is adapted to the controller structure in order to determine the gain sets/tables along the entire envelope
- Development of a tool for stability and robustness assessment incorporating a detailed uncertainty model which is suited for the controller structure proposed in order to facilitate the certification (e.g. μ-Analysis)

Testing of the whole development chain will be accomplished by implementing of the lateral controller developed into an appropriate general aviation aircraft simulator (DA-42 FTD) and finally, by actually flying it on the corresponding in-flight simulator "*Fliegender Erprobungsträger Bayern*", a research aircraft based on a DA-42 M-NG airframe planned and developed at the Institute of Flight System Dynamics of Technische Universität München.

2 Flight Dynamics Modeling

A small aircraft featuring an active Fly-by-Wire Flight Control System (FbW FCS) represents a new class of general aviation airplanes. By utilizing the benefits of such a system, some basic aircraft design objectives (e.g. static stability and damping characteristics) may be shifted to other focuses and thus, the aircraft concept may differ from today's conventional shape. Nevertheless, the novel FbW FCS must be implemented, demonstrated and tested within a suitable flying testbed before designing an entire new aircraft relying on such a novel FbW FCS. For this purpose, the Institute of Flight System Dynamics owns a fully-fledged Flight Control System Development and Integration Environment:

- An integrated tool chain to efficiently support Model Based Development (MBD) of functional algorithms for onboard applications. All tools used are compliant with airworthiness requirements.
- A DA-42 Flight Training Device (D-SIM42 FTD simulator) with extensive capability to simulate malfunctions of multiple aircraft systems for design and validation of control laws and pilot in the loop verification.
- A DA-42 Airframe and Control System Iron Bird for component tests and verification, integration tests for research aircraft and hardware in the loop simulation in connection with the Flight Training Device.
- And in particular a research aircraft Diamond DA-42 MPP NG (Multi Purpose Platform New Generation), particularly dedicated as in-flight simulator with an Experimental Fly-by-Wire (EFbW) control system, see Fig. 1.

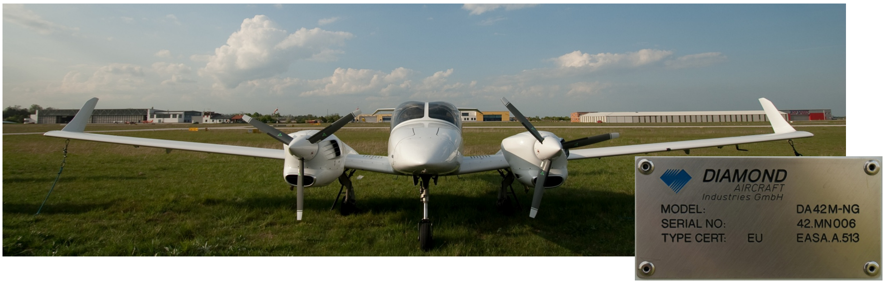


Fig. 1 Research Flying Testbed (DA-42 MNG) featuring EFbW FCS

With this continuous "end-to-end" development and integration chain at hand, it suggests itself to utilize the Diamond DA-42 aircraft as reference configuration. For this type of aircraft the feasibility, advantages and reliability of the active FbW FCS for general aviation aircraft will be analyzed, verified and finally (in-flight) demonstrated.

Regarding the functional layout and development of the control system, an appropriate high-fidelty flight simulator for functional testing is of vital importance. The available D-SIM42 Flight Training Device is equipped with original glass cockpit, *Garmin G1000 Avionics package* with primary flight display (*PFD*) as well as a multi-function display in order to achieve the most realistic cockpit environment. The DA-42 simulator is thus a perfect tool for controller functionality pilot-in-the-loop testing under "real world" conditions. Unfortunately, the flight dynamics model as well as all internal signal processing are completely capsuled "black boxes" and entirely isolated from external inputs. Both is adverse for an application of the simulator as design and implementation platform for the active FCS to be developed.

For this reason, an new nonlinear six 6 dof model called "FSD DA-42/FSA 6 DOF Flight Dynamics & Simulation Model" has been implemented using *MATLAB / Simulink* and its top-level block structure is shown in Fig. 2. Particular emphasis was placed on a high fidelity physical modeling and on a exact reproduction of the dynamics of the D-SIM42 FTD simulator.

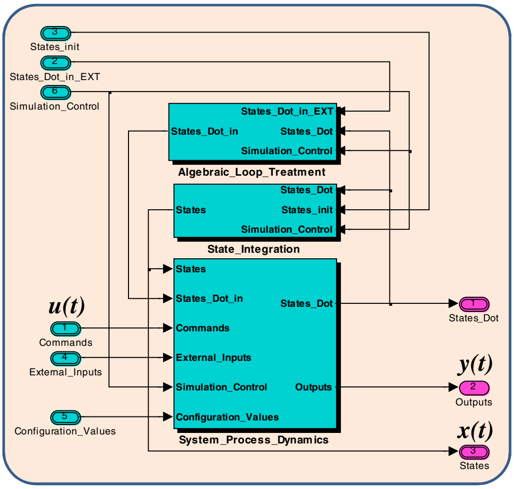


Fig. 2 FSD DA-42/FSA Nonlinear 6 DOF Flight Dynamics & Simulation Model

The validation of the FSD DA-42/FSA 6 DOF flight dynamics model was accomplished by systematic simulator flight tests conducted in the D-SIM42 device as detailed within Ref. [7] assuring an adequate matching versus the D-SIM42 FTD.

3 Lateral Controller Design

3.1 Objectives, Derived Requirements, Deduced "Design Philosophy"

The main objective of the FSA lateral control design is to provide excellent flying/handling qualities along the entire envelope in order to reduce pilot's workload significantly when flying the airplane manually. Related objectives are to increase the safety and comfort of passengers and crew together with the efficiency of the aircraft (fuel consumption).

To achieve these goals, it is required to improve and homogenize the stability and command characteristics to keep them intuitive and predictable over the operational envelope and to accomplish an effective gust load rejection. Additionally the effects of configuration changes shall automatically be compensated and a further reduction of the pilot's workload shall be achieved by partial automation of secondary controls, e.g. spoiler or thrust setting.

Consequently, the following primary design objectives may be deduced and committed in terms of our "design philosophy":

- Modification of the aircraft's stability characteristics:
  The basic stability characteristics and thus the flying qualities, shall be modified by assignment of "optimal" damping, frequency and time constants to the different eigenmodi of the lateral motion.
- Augmentation of the command behavior:
  - Rate Command/Attitude Hold (*RC*/*AH*) versus angle of sideslip (*AOS*) Command/zero lateral load characteristics:

    The provision of velocity vector roll and angle of sideslip command has shown to be very intuitive and predictable for the pilot in combination with attitude hold or zero lateral load factor characteristics respectively when the inceptor is released.

  - Feed Forward Path Augmentation:

    By implementation of a "direct link" feed forward branch, it is possible to improve the aircraft's control sensitivity and to cancel out the integrator poles of the corresponding command transfer function n order to obtain a more "crisp" control behavior
- Decoupling of control inputs (introduction of decoupled "auxiliary control effectors"):

  To manually fly a velocity vector roll, the pilot would have to apply coordinating rudder command and vice versa a pure buildup of sideslip would require adding adequate aileron deflections to the pedal command. This maneuver coordination for the decoupling of experimental roll and yaw axes shall be accomplished automatically by means of auxiliary control inputs. A respective control axes decoupling is not only convenient for the pilot, it also makes the subsequent controller layout much more intuitive and straightforward.
- Counteracting external disturbances (gusts/turbulence) and compensating for changes of the aircraft configuration:

  External disturbances shall be suppressed efficiently as well as changes in the configuration (e.g. by deflecting the wing flaps, by a malfunction of an engine or by extending and retracting the landing gear) shall be compensated as far as possible in order to further reduce the pilot's workload.

3.2 Controller Structure

As described within the previous section, the controller to be developed has to comply with several different objectives. The specific design goals are thereby typically associated with the different branches in the controller structure. Subsequently, the overall controller structure as depicted in Fig. 3 is introduced and the design philosophy is detailed by discussion of every branch.

Generally, the controller consists of the *Command Signal Generation* providing roll rate and angle of sideslip commands, decoupling *Yaw* and *Roll Axis Controller* relying on the calculated commands and sensor feedbacks creating generalized input variables that correspond to a demanded roll and yaw control moment w.r.t. the experimental-axis, respectively as well as the *Control Allocation* converting the generalized input variables to equivalent allocated control surface deflection commands.

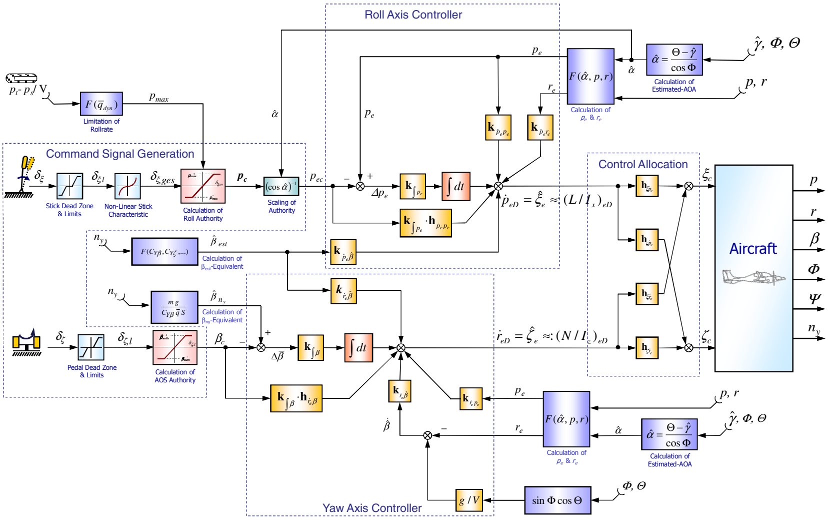

Roll Axis Controller
$p_t - p_s / V$
$F(\bar{q}_{dyn})$
Limitation of Rollrate
p_{max}
$\hat{\alpha}$
p_e
r_e
$F(\hat{\alpha}, p, r)$
Calculation of p_e & r_e
$\hat{\alpha} = \frac{\Theta - \hat{\gamma}}{\cos \Phi}$
Calculation of Estimated-AOA
$\hat{\gamma}, \Phi, \Theta$
p, r
$\mathbf{k}_{\dot{p}_e p_e}$
$\mathbf{k}_{\dot{p}_e r_e}$
Command Signal Generation
δ_ξ
$\delta_{\xi l}$
$\delta_{\xi ges}$
p_c
p_{ec}
Stick Dead Zone & Limits
Non-Linear Stick Characteristic
Calculation of Roll Authority
$(\cos \hat{\alpha})^{-1}$
Scaling of Authority
−
+
Δp_e
$\mathbf{k}_{\int p_e}$
$\int dt$
$\mathbf{k}_{\int p_e} \cdot \mathbf{h}_{\dot{p}_e p_e}$
$\dot{p}_{eD} = \hat{\xi}_e \approx: (L / I_x)_{eD}$
Control Allocation
$\mathbf{h}_{\xi \dot{p}_e}$
$\mathbf{h}_{\zeta \dot{p}_e}$
$\mathbf{h}_{\xi \dot{r}_e}$
$\mathbf{h}_{\zeta \dot{r}_e}$
ξ_c
ζ_c
Aircraft
p
r
β
Φ
Ψ
n_y
n_y
$F(C_{Y\beta}, C_{Y\zeta}, ...)$
$\hat{\beta}_{est}$
Calculation of β_{est}-Equivalent
$\mathbf{k}_{\dot{p}_e \hat{\beta}}$
$\boldsymbol{k}_{\dot{r}_e \hat{\beta}}$
$\frac{m g}{C_{Y\beta} \bar{q} S}$
$\hat{\beta}_{n_y}$
Calculation of β_{ny}-Equivalent
δ_ζ
$\delta_{\zeta l}$
Pedal Dead Zone & Limits
Calculation of AOS Authority
β_c
$\Delta \bar{\beta}$
$\mathbf{k}_{\int \beta}$
$\mathbf{k}_{\int \beta} \cdot \mathbf{h}_{\dot{r}_e \beta}$
$\dot{r}_{eD} = \hat{\zeta}_e \approx: (N / I_z)_{eD}$
$\mathbf{k}_{\dot{r}_e \dot{\beta}}$
$\mathbf{k}_{\dot{r}_e p_e}$
$\dot{\hat{\beta}}$
g / V
$\sin \Phi \cos \Theta$
Φ, Θ
Yaw Axis Controller

Fig. 3 Controller Structure

To ensure proper function of the controller, high and low frequency measurements have to be available appropriately, as listed in the following table (whereas the "hat-variables" represent not directly measured but estimated signals).

Table 1 Measurement Signals

High Frequency Measurements	*Low Frequency Measurements*
$n_y,\ p,\ r$	$\Phi,\ \Theta,\ (\hat{\gamma})$

- ***Control Allocation ("Decoupled virtual controls")***
  As previously stated, the aircraft shall roll around the velocity vector and yaw around the experimental *z*-axis (generating a pure $\dot{\beta}$) for improved command behavior and easier controllability (decoupling of roll and yaw motion) and thus, to increase flight efficiency.

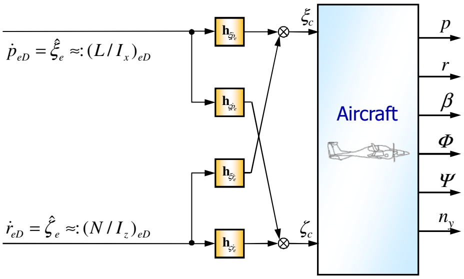

$\dot{p}_{eD} = \hat{\xi}_e \approx: (L/I_x)_{eD}$
$\dot{r}_{eD} = \hat{\zeta}_e \approx: (N/I_z)_{eD}$
$\mathbf{h}_{\xi \dot{p}_e}$
$\mathbf{h}_{\zeta \dot{p}_e}$
$\mathbf{h}_{\xi \dot{r}_e}$
$\mathbf{h}_{\zeta \dot{r}_e}$
ξ_c
ζ_c
Aircraft
p
r
β
Φ
Ψ
n_y

Fig. 4 Control Allocation (Subfigure to Figure 3)

With such an approach, using a ("reversionary") direct link law a lateral stick input of the pilot (*roll rate command*) yields only a roll acceleration ($\dot{p}_e$) around the aerodynamic flow direction without any excitation of the angle of sideslip. Conversely, a pedal input (*angle of sideslip command*) would introduce a pure yaw rate (r_e) without any excursion of the bank angle, i.e. the aircraft does not tilt to the side like a typical unaugmented aircraft.

To achieve such decoupled virtual controls, the lateral control surfaces (aileron and rudder) need to be coordinated in such a manner, that a roll rate command generates a pure moment around the x_e-axis. Accordingly, an angle of sideslip command has to produce a pure moment around the z_e-axis.

Eqn. (1) provides the linear state space model of the lateral motion w.r.t. experimental axes. Via the entries N_ξ and L_ζ in the input matrix $\boldsymbol{B}$, an aileron deflection ξ results in an adverse rotational (yaw) acceleration ($\dot{r}_e$) and a rudder deflection (ζ) results in an undesired rotational (roll) acceleration $\dot{p}_e$, respectively.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} \quad \Leftrightarrow \quad \begin{pmatrix} \dot{p}_e \\ \dot{r}_e \\ \dot{\beta} \\ \dot{\Phi} \end{pmatrix} = \begin{bmatrix} L_p & L_r & L_\beta & 0 \\ N_p & N_r & N_\beta & 0 \\ Y_p & Y_r - 1 & Y_\beta & \dfrac{g_0}{V_0}\cos\Theta_0 \\ \dfrac{\cos\gamma_0}{\cos\Theta_0} & \dfrac{\sin\gamma_0}{\cos\Theta_0} & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} p_e \\ r_e \\ \beta \\ \Phi \end{pmatrix} + \begin{bmatrix} L_\xi & L_\zeta \\ N_\xi & N_\zeta \\ Y_\xi & Y_\zeta \\ 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} \xi \\ \zeta \end{pmatrix} \qquad (1)$$

The necessary feed forward law for *control allocation* in order to decouple the control axes may be derived from the desired change of states due to the control surface deflections:

$$\begin{pmatrix} \dot{p}_{eD} \\ \dot{r}_{eD} \end{pmatrix} = \begin{bmatrix} L_\xi & L_\zeta \\ N_\xi & N_\zeta \end{bmatrix} \cdot \begin{pmatrix} \xi \\ \zeta \end{pmatrix} =: \begin{pmatrix} \hat{\xi}_e \\ \hat{\zeta}_e \end{pmatrix} \tag{2}$$

Eqn. (2) introduces the virtual "auxiliary" control surfaces $\hat{\xi}_e$ and $\hat{\zeta}_e$ which act directly on the x_e respectively z_e-axis. Inversion of Eq. (2) provides a suitable feed forward control law:

$$\begin{pmatrix} \xi \\ \zeta \end{pmatrix} = \begin{bmatrix} h_{\xi\dot{p}_e} & h_{\zeta\dot{p}_e} \\ h_{\xi\dot{r}_e} & h_{\zeta\dot{r}_e} \end{bmatrix} \cdot \begin{pmatrix} \dot{p}_{eD} \\ \dot{r}_{eD} \end{pmatrix} \quad \textit{with} \quad \begin{bmatrix} h_{\xi\dot{p}_e} & h_{\zeta\dot{p}_e} \\ h_{\xi\dot{r}_e} & h_{\zeta\dot{r}_e} \end{bmatrix} = \begin{bmatrix} L_\xi & L_\zeta \\ N_\xi & N_\zeta \end{bmatrix}^{-1} \tag{3}$$

Substitution of (ξ, ζ) in the state space model (1) by the control allocation law Eq. (3) yields the dynamics augmented by decoupled "auxiliary" control inputs as suitable basis for the subsequent controller layout:

$$\begin{pmatrix} \dot{p}_e \\ \dot{r}_e \\ \dot{\beta} \\ \dot{\Phi} \end{pmatrix} = \begin{bmatrix} L_p & L_r & L_\beta & 0 \\ N_p & N_r & N_\beta & 0 \\ Y_p & Y_r - 1 & Y_\beta & \frac{g_0}{V_0}\cos\Theta_0 \\ \frac{\cos\gamma_0}{\cos\Theta_0} & \frac{\sin\gamma_0}{\cos\Theta_0} & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} p_e \\ r_e \\ \beta \\ \Phi \end{pmatrix} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \breve{Y}_\xi & \breve{Y}_\zeta \\ 0 & 0 \end{bmatrix}}_{\breve{B}} \cdot \begin{pmatrix} \hat{\xi}_e \\ \hat{\zeta}_e \end{pmatrix} \tag{4}$$

Nevertheless, as may be seen from Eq. (4) the virtual "auxiliary" controls still exhibit undesired side-effects on the $\dot{\beta}$-equation due to the application of a partial inversion of the input matrix $\boldsymbol{B}$. However, via analytical as well as numerical analyses it could be verified, that the magnitudes of the residual derivatives $\breve{Y}_\xi, \breve{Y}_\zeta$ are negligible compared to the other contributions of the side-force equation, compare Table 2. Hence, for the succeeding controller design they can be set to zero as it was subsequently justified within the "Controller Analysis" without neglected side-force residuals, see Chapter 5.

Table 2 Example Values of $\breve{Y}_\xi$ and $\breve{Y}_\zeta$

	DA 42 (Cruise) [Source: Simulation Model FSD]		Do 328 (Cruise) [Data extracted from Ref. [4]]
	110kts	90kts	
$\breve{Y}_\xi = \frac{Y_\xi N_\zeta - Y_\zeta N_\xi}{L_\xi N_\zeta - L_\zeta N_\xi}$	$-7.49 \cdot 10^{-4}$	-0.00170	$9.205 \cdot 10^{-4}$
$\breve{Y}_\zeta = \frac{-Y_\xi L_\zeta + Y_\zeta L_\xi}{L_\xi N_\zeta - L_\zeta N_\xi}$	-0.0120	-0.0179	-0.0150

Remark: Alternatively, a corresponding feed forward control allocation law may be introduced by means of the *Moore-Penrose Pseudo-Inverse* of the control matrix $\boldsymbol{B}$. However, it could be shown, that the results of both methods are almost identical within the numerical accuracy.

- ***Command Signal Generation***

  The Command Signal Generation path scales and limits the stick and pedal deflections appropriately and provides predictable and intuitive stick/pedal characteristics to the pilot. Subsequently, an authority calculation in terms of the maximum commanded roll rate and angle of sideslip according to the airworthiness limitations is performed depending on the current flight condition.

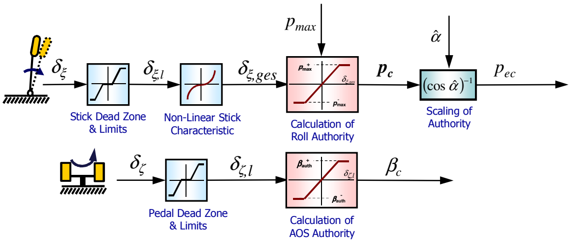
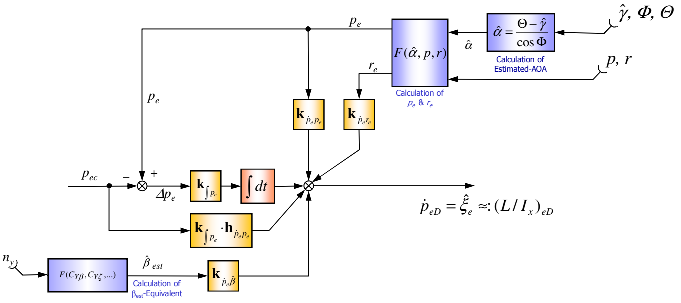


Fig. 5 Command Signal Generation (Subfigure to Figure 3)

- ***Roll Axis Controller***

  After introduction of the decoupled "virtual controls" (see previous section) the roll axis controller basically represents a classical *PI*-scheme augmented by a direct link path in order to ensure the desired RC/AH characteristic, see Fig. 6.

  Here, the integrator path ensures zero steady-state error w.r.t. the commanded roll rate. The direct link path on the other hand is used to improve the roll command behavior (initial reaction) of the aircraft w.r.t. excellent handling qualities by "cancelling" out the integrator pole of the stick-to-roll transfer function. Via the gains $k_{\dot{p}_e r_e}$ and $k_{\dot{p}_e \beta}$, the yaw axis cross coupling on the roll axis will be eliminated. The specific layout of the roll controller gains is detailed within next section "*Model Reference Direct Eigenstructure Assignment*".



p_e
$F(\hat{\alpha}, p, r)$
$\hat{\alpha} = \frac{\Theta - \hat{\gamma}}{\cos\Phi}$
$\hat{\gamma}, \Phi, \Theta$
$\hat{\alpha}$
Calculation of Estimated-AOA
p, r
r_e
Calculation of p_e & r_e
p_e
$\mathbf{k}_{\dot{p}_e p_e}$
$\mathbf{k}_{\dot{p}_e r_e}$
p_{ec}
−
+
Δp_e
$\mathbf{k}_{\int p_e}$
$\int dt$
$\mathbf{k}_{\int p_e} \cdot \mathbf{h}_{\dot{p}_e p_e}$
$\dot{p}_{eD} = \hat{\xi}_e \approx: (L/I_x)_{eD}$
n_y
$F(C_{Y\beta}, C_{Y\zeta}, ...)$
$\hat{\beta}_{est}$
Calculation of β_{est}-Equivalent
$\mathbf{k}_{\dot{p}_e \hat{\beta}}$

Fig. 6 Roll Axis Controller (Subfigure to Figure 3)

- ***Yaw Axis Controller***

  Correspondingly, based on the control decoupling functionality (see previous section), the yaw axis controller consists of an analogous *PI*-structure augmented by a direct link path in order to modify the damping and natural frequency of the dutch roll and to shape adequate command characteristics, see Fig. 7.

  Again, the integrator path ensures steady-state accuracy with regard to the commanded angle of sideslip (AoS) versus the n_y equivalent AoS signal denoted β_{ny} in Figs. 5 and 7. Again, the feed-forward (direct link) branch is introduced to improve the command behavior of the aircraft in terms of excellent handling qualities by "cancelling out" the integrator pole. Via the gain $k_{\dot{r}_e p_e}$ the main cross coupling from the roll to the yaw axis will be compensated.

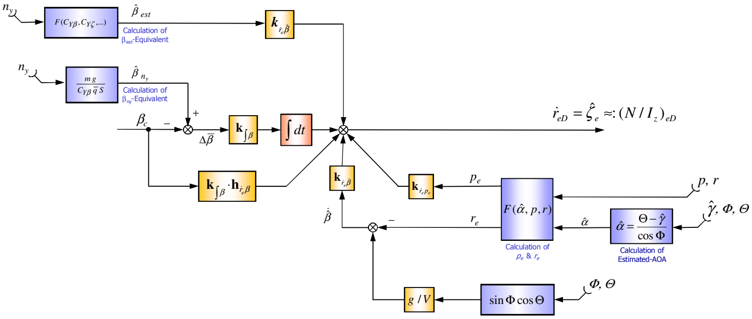


Fig. 7 Yaw Axis Controller (Subfigure to Figure 3)

- ***Estimated and Equivalent Flow Angle Signals***

  Due to the fact, that General Aviation/small aircraft commonly do not feature a sideslip and/or an angle of attack vane/sensor, there is generally no applicable measurement of the angle of sideslip and/or the angle of attack (AoA) available and for flight control purposes they have to be estimated conveniently. An appropriate subsidiary signal for (measured) AoS which can be used for sideslip feedback may be calculated as follows (Ref. [9]):

$$\hat{\beta}_{est} = \left(\frac{m\,g\,n_{ye}}{\bar{q}\,S} - C_{Q\zeta}\,\zeta - C_{Qr}\,r - C_{Qp}\,p \right) \frac{1}{-C_w + C_{Q\beta}} \tag{5}$$

  Currently, the necessary aerodynamic derivatives for the AoS estimation in Eq. (5) are taken from the aerodynamic model within the simulator. However, in future it is intended to validate or (if applicable) substitute them via flight test parameter identification.

  Similarly, the angle of attack signal utilized by the controller is generated by the simple, but well-proven relationship (Ref. [9]):

$$\hat{\alpha} = \frac{\Theta - \hat{\gamma}}{\cos\Phi} \tag{6}$$

In contrast to the roll axis branch, where the controller uses only one feedback signal (roll rate p_e), the yaw axis controller applies two different feedback signals: $\hat{\beta}_{est}$ and $\hat{\beta}_{n_y}$. Here, $\hat{\beta}_{n_y}$ represents a scaled equivalent sideslip signal proportional to the lateral load factor which is defined as:

$$\hat{\beta}_{n_y} = \frac{m\,g\,n_{ye}}{\bar{q}\,S\,C_{Q\beta}} \tag{7}$$

The use of $\hat{\beta}_{n_y}$ for the integral feedback path (in addition to $\hat{\beta}_{est}$ for proportional feedback) is motivated by a desired zero side force flight condition (n_y = 0) in case of zero pedal (i.e. β_c =0). If the aerodynamic AoS or the corresponding estimated $\hat{\beta}_{est}$ (acc. to Eqn. (5)) would be utilized for integral feedback, even in straight and level flight a small residual side force would remain ($n_y \neq 0$) due to unavoidable minor asymmetric effects (e.g. propulsion, lateral c.g. shift, asymmetric aerodynamics) and hence, a sustained yaw/turn rate and consequently drift would occur. On the other hand, applying an n_y-equivalent integral feedback acc. to Eq. (7) signal always ensures zero side force in (zero pedal) steady state condition und thus, a well-coordinated flight without any drift.

3.3 Model Reference Direct Eigenstructure Assignment (MR DEA)

An intuitive and preserving full visibility approach for the layout of the specific controller gains (feedback, feedforward plus cross-feed), which is directly suited to assure excellent flying qualities has been elaborated and is proposed here referred to as “Model Reference Direct Eigenstructure Assignment (*MR DEA*)”.

The basis for the calculation of the controller gains is formed by the state space model of the closed loop system with the control allocation in terms of "decoupled virtual controls" applied as introduced before. Concerning the linear gain layout, the differences between real and estimated AoS, i.e. β, $\hat{\beta}_{est}$ and $\hat{\beta}_{n_y}$ are negligible and hence, are not taken into account and the unified feedback signal is β. However, the validity of this assumption has to be proved within the controller analysis and (nonlinear) assessment, which could have been accomplished successfully.

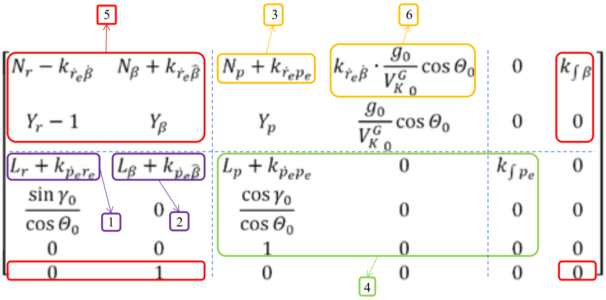
First, the control allocation gains ($h_{\xi\dot{p}_e}$, $h_{\zeta\dot{p}_e}$, $h_{\xi\dot{r}_e}$, $h_{\zeta\dot{r}_e}$) are calculated as given by Eqn. (3). Next, based on the introduction of the "decoupled virtual controls" (see Fig. 3) the closed loop state space model (including the two integrator states $x_{\int\Delta p}$, $x_{\int\Delta\beta}$) with the side force residuals $\breve{Y}_{\xi}$, $\breve{Y}_{\zeta}$ set to zero can be evaluated to:

$$\begin{pmatrix} \dot{r}_e \\ \dot{\beta} \\ \dot{p}_e \\ \dot{\boldsymbol{\Phi}} \\ \dot{x}_{\int \Delta p} \\ \dot{x}_{\int \Delta\beta} \end{pmatrix} = \begin{bmatrix} N_r - k_{\dot{r}_e\dot{\beta}} & N_\beta + k_{\dot{r}_e\hat{\beta}} & N_p + k_{\dot{r}_e\dot{p}_e} & k_{\dot{r}_e\dot{\beta}}\dfrac{g_0}{V_0}\cos\Theta_0 & 0 & k_{\int\beta} \\ Y_r - 1 & Y_\beta & Y_p & \dfrac{g_0}{V_0}\cos\Theta_0 & 0 & 0 \\ L_r + k_{\dot{p}_e\dot{r}_e} & L_\beta + k_{\dot{p}_e\hat{\beta}} & L_p + k_{\dot{p}_e p_e} & 0 & k_{\int p_e} & 0 \\ \dfrac{\sin\gamma_0}{\cos\Theta_0} & 0 & \dfrac{\cos\gamma_0}{\cos\Theta_0} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} r_e \\ \beta \\ p_e \\ \boldsymbol{\Phi} \\ x_{\int \Delta p} \\ x_{\int \Delta\beta} \end{pmatrix}$$

$$+ \begin{bmatrix} 0 & k_{\int\beta}\, h_{\dot{r}_e\beta} \\ 0 & 0 \\ k_{\int p_e}\, h_{\dot{p}_e p_e} & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} p_{ec} \\ \beta_c \end{pmatrix} \tag{8}$$

Fig. 8 depicts the closed-loop system matrix $\boldsymbol{A}_{CL}$ with the decisive subsystems and specific elements highlighted according to their significance w.r.t. the (desired) closed-loop dynamics and the controller gain layout as further explained below.

Initially, in order to provide a proper "best possible" decoupling of the roll and the yaw (control) axis, the dedicated controller cross feed gains of elements (boxes) number 1, 2 and 3 each are adjusted to cancel the corresponding entry out, i.e. to make the coupling element equal to zero. The closed-loop derivative number 6, which represents a special case of controller induced coupling will be discussed later separately.



$$\begin{bmatrix} N_r - k_{\dot{r}_e\dot{\beta}} & N_\beta + k_{\dot{r}_e\hat{\beta}} & N_p + k_{\dot{r}_e p_e} & k_{\dot{r}_e\dot{\beta}} \cdot \dfrac{g_0}{V^G_{K\,0}}\cos\Theta_0 & 0 & k_{\int\beta} \\ Y_r - 1 & Y_\beta & Y_p & \dfrac{g_0}{V^G_{K\,0}}\cos\Theta_0 & 0 & 0 \\ L_r + k_{\dot{p}_e r_e} & L_\beta + k_{\dot{p}_e\hat{\beta}} & L_p + k_{\dot{p}_e p_e} & 0 & k_{\int p_e} & 0 \\ \dfrac{\sin\gamma_0}{\cos\Theta_0} & 0 & \dfrac{\cos\gamma_0}{\cos\Theta_0} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 8 Closed Loop System Matrix

After application of this pre-decoupling and omitting the remaining coupling elements in the closed-loop system matrix $\boldsymbol{A}_{CL}$ (due to the fact that they are small and hence, with minor impact on the dynamics as will be verified in Chapter 4), two almost decoupled subsystems are achieved with corresponding dynamic matrices $\boldsymbol{A_R}$ and $\boldsymbol{A_Y}$ of roll and yaw axis, respectively (green box #4 and red boxes # 5 in Fig. 8):

$$A_R = \begin{bmatrix} L_p + k_{\dot{p}_e p_e} & 0 & k_{\int p_e} \\ \dfrac{\cos\gamma_0}{\cos\Theta_0} & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad A_Y = \begin{bmatrix} N_r - k_{\dot{r}_e \dot{\beta}} & N_\beta + k_{\dot{r}_e \hat{\beta}} & k_{\int \beta} \\ Y_r - 1 & Y_\beta & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{9}$$

The desired closed-loop (reference) dynamics of the roll and yawing motion (including the controller introduced integrator poles) are each represented by their characteristic polynomials. Regarding the roll axis we specify

$$N_{CL,R}(s) = (T_{R,d}\, s + 1)(T_{IR,d} s + 1) s = (T_{R,d}\, T_{IR,d} s^2 + (T_{R,d} + T_{IR,d}) s + 1) s \tag{10}$$

where T_{R,d_D} is the desired roll time constant, $T_{IR,d}$ the desired integrator time constant and the free "s" represents the neutrally stable spiral mode according to the aspired RC/AH characteristics.

Similarly, for the yaw channel the reference characteristic polynomial is:

$$\begin{aligned} N_{CL,Y}(s) &= (s^2 + 2\zeta_{DR,d}\, \omega_{0DR,d}\, s + \omega_{0DR,d}^2)(T_{IY,d} s + 1) \\ &= T_{IY,d}\, s^3 + (2\zeta_{DR,d}\, \omega_{0DR,d}\, T_{IY,d} + 1) s^2 + (2\zeta_{DR,d}\, \omega_{0DR,d} + T_{IY,d} \omega_{0DR,d}^2) s + \omega_{0DR,d}^2 \end{aligned} \tag{11}$$

with the desired dutch roll damping $\zeta_{DR,d}$, frequency $\omega_{0DR,d}$ and the integrator time constant $T_{IY,d}$, respectively.

By calculation of the corresponding characteristic polynomials of the closed-loop subsystem matrices in Eq. (10) and coefficient comparison with Eqs. (11, 12) the basic feedback gains $(k_{\dot{p}_e p_e}, k_{\int p_e}, k_{\dot{r}_e \dot{\beta}}, k_{\dot{r}_e \hat{\beta}}, k_{\int \beta})$ can be determined successively in order to achieve the desired stability characteristics (in terms of closed-loop roll time constant, dutch roll damping and frequency) which are assigned according to the so-called MIL Level 1* specifications (Ref. [1-3]) (i.e. middle of Level 1 boundary).

For a typical open loop dynamics or a pure yaw rate feedback the controller induced cross coupling element (# 6 in Fig. 8) is zero. However, in order to provide a suitable turn coordination and thus, to improve the handling qualities effectively, an elegant measure is to remove the steady turn yaw rate $\sin\Phi \cos\Theta\, g / V$ from the measured (over all) yaw rate feedback signal (see Fig. 3, in fact an equivalent sideslip time-derivative will be fed back here). This yields the artificial cross coupling (#6 in Fig. 8) which supports a well coordinated flight without sideslip excitations or deviations when turning/banking.

Finally, the feed-forward gains of the two direct link paths (Figs. 6 & 7) may assigned straight-forward via the corresponding closed-loop transfer functions for roll rate and sideslip. From Eq. (9) the transfer functions $g_{p_e p_{ec}}(s)$ and $g_{\beta\beta_c}(s)$ may be easily derived (e.g. applying the *Cramer* rule) as :

$$g_{p_e p_{ec}} = \frac{k_{\int p_e}\, h_{\dot{p}_e p_e} (s - 1/h_{\dot{p}_e p_e})}{(s + 1/T_R)(s + 1/T_{IR})} \text{ , where } k_{\int p_e} = -1/(T_R T_{IR}) \tag{12}$$

$$g_{\beta\beta_c} = \frac{k_{\int\beta} h_{\dot{r}_e\beta}\left(Y_r - 1\right)\left(s - 1/h_{\dot{r}_e\beta}\right)}{(s^2 + 2\zeta_{DR}\,\omega_{0DR}\,s + \omega_{0DR}^2)(s + 1/T_{IY})}\text{, where } k_{\int\beta} = \omega_{0DR}^2/(T_{IY}(1 - Y_r)) \qquad (13)$$

Here the design objective is an crisp "typical aircraft-like" initial response (without perceiving the integrator lags) and hence, the feed-forward gains $h_{\dot{p}_e p_{ec}}$ and $h_{\dot{r}_e\beta}$ each are computed to cancel out the corresponding integrator pole in the denominator polynomial, i.e. $h_{\dot{p}_e p_e} = -T_{IR}$, $h_{\dot{r}_e\beta} = -T_{IY}$.

A great advantage of the design methodology presented is, that based on the in-depth system knowledge in terms of the structure of the (closed-loop) system matrices ($\boldsymbol{A}_{CL}$, $\boldsymbol{B}_{CL}$) and the correlation with most favorable flying qualities (acc. to the so-called MIL Level 1* specifications (Ref. [1-3])) the controller gains can be determined straight-forward without any iteration in order to ensure the assigned reference dynamics (compare Ref. [5]). In contrast, classical direct eigenstructure assignment e.g. requires a numerical method to compute the feedback gain matrix $\boldsymbol{K}$ with the quality of solution (achieved eigenstructure) is strongly depending on a careful specification of desired (attainable) eigenvalue/eigenvector sets (nullspace projection) and the assignability of single controller gains to specific flying quality requirements is almost lost (e.g. Ref. [6]).

4 Controller Analysis

In order to verify the controller design philosophy and to demonstrate the capability of the approach proposed an extensive controller assessment has been performed. As a first step, a linear controller and robustness analysis is accomplished by calculation of the closed loop eigenvalues and eigenvectors, linear simulations (e.g. step responses) and classical *SISO* (single input single output) nichols plots (simultaneous gain/phase margins for rudder an aileron cuts).

Fig. 9 depicts the open-loop, the desired and the closed loop-poles (eigenvalues) for a representative flight condition of the DA-42 with V_{TAS} = 139 kts, h =916.3 m, m = 1590 kg and an intermediate c.g. position x_{CG} = -2.3727 m (mean design point).

Generally, the resulting closed-loop poles match the assigned desired locations very well. Merely the closed loop dutch roll oscillation a shows a perceptible offset compared to the specified "ideal" location featuring a relative error in natural frequency ω_{0DR} about 0.5 % and an error in relative damping of 0.025 %. However, these deviations are really small and can be proven to depend on the intentionally neglected coupling derivatives (as Y_p) and side force residuals ($\breve{Y}_\xi, \breve{Y}_\zeta$) within the layout procedure. Moreover, they act in a conservative way (higher damping) and hence, are regarded to be (fully) acceptable.

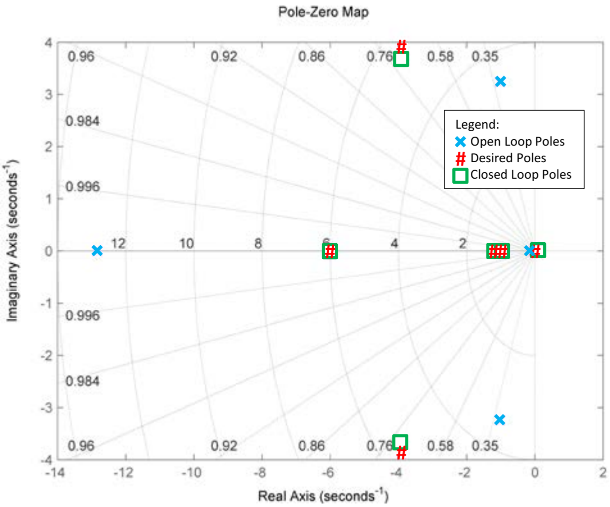


Fig. 9 Open-Loop / Desired / Closed Loop Poles (Eigenvalues)

The best way to illustrate the capability of the *MR DEA* design procedure applied may yield the comparison of the open-loop, the "ideal" reference and the resulting closed-loop dynamics in terms of the corresponding system matrices $\boldsymbol{A}$, see Fig. 8 and Table 3. The light grey shadowed elements represent the controller decoupling effects and the dark grey highlighted entry embodies the artificial cross coupling supporting a well coordinated flight without sideslip excitations ("turn compensation") as discussed above. As may bee seen from Table 3 in comparison with Fig. 8 the decoupling capability of the control scheme presented is very effective and the aspired structure revealing two decoupled subsystems for roll and yaw channel could be achieved quite fairly.

Table 3 Open- and Closed-Loop A-Matrix

States	Open Loop $\boldsymbol{A}$-Matrix				Closed Loop $\boldsymbol{A}$-Matrix					
r_e	-1.8764	10.1548	-0.4523	0.0000	-8.5759	36.0948	0	0.9185	0	30.2855
β	-0.9988	-0.2341	0.0054	0.1371	-0.9337	-0.4908	-0.0004	0.1282	0.0014	-0.2933
p_e	1.0826	-22.9306	-12.8038	-0.0000	0	0	-7.0000	-0.0000	-6.0000	0
Φ	0	-0.0000	1.0001	0.0000	0	-0.0000	1.0001	0.0000	0	0
$x_{\int \Delta p}$	-	-	-	-	0	0	1.0000	0	0	0
$x_{\int \Delta \beta}$	-	-	-	-	0	1.0000	0	0	0	0

The excellent decoupling potential is further highlighted by the computation of the associated open-loop versus the closed-loop eigenvectors, see Tables 4 and 5. The adverse entries concerning the states p_e and Φ within the dutch roll eigenvector vanish and conversely, the undesirable effects of p_e and especially β are cancelled out in the spiral eigenvector.

Table 4 eigenvectors open loop

Open – Loop			
States	Dutch Roll	Roll Mode	Spiral Mode
r_e	0.8359	-0.0379	0.1312
β	0.0630 ± 0.2473i	-0.0034	0.0226
p_e	-0.1656 ± 0.4357i	-0.9963	-0.0294
Φ	-0.1077 ± 0.0851i	0.0776	0.9907

Table 5 eigenvectors closed loop

Closed – Loop					
States	Dutch Roll	Roll Mode	Spiral Mode	Yaw Int. Dyna.	Roll Int. Dyna.
r_e	0.9833	0.0111	-0.1336	-0.5342	-0.0160
β	0.1248 ± 0.1286i	0.0057	-0.0000	-0.6061	-0.1111
p_e	-0.0000 ± 0.0000i	0.9733	0.0000	0.0000	-0.5701
Φ	0.0000 ± 0.0000i	-0.1622	-0.9910	-0.0000	0.5701
$x_{\int \Delta p}$	0.0000 ± 0.0000i	-0.1622	0.0000	-0.0000	0.5701
$x_{\int \Delta \beta}$	-0.0008 ± 0.0327i	-0.0010	-0.0078	0.5892	0.1110

Summarizing, the *MR DEA* approach proposed provides an effective means for excellence in flying quality design in terms of directly assigning a desired reference dynamics to the closed-loop system matrices based on an in-depth knowledge of their shape (structure and settings) in order to meet typical ideal "Level 1" flying quality requirements.

In a second step, the linear design has been verified successfully by repetition of the controller analysis (as presented above) after integration of the flow angle estimations plus necessary filters (i.e. n_y-noise filter, etc.), sensor and actuator models along the entire envelope plus a subsequent *SISO* robustness prove by means of Nichols plots (sufficient phase and gain margins).

5 Nonlinear Simulation Results (Controller Assessment)

With the primary intention to demonstrate (i.e. to fly) the controller and its capabilities in flight on our Research Flying Testbed (DA-42 MNG), the designed lateral control system has been appropriately discretized and implemented within the DA-42 FTD simulator to perform a comprehensive (nonlinear) assessment including simulator flight test to gain pilot evaluation/ratings.

Initially, single axis square inputs of pedal and lateral stick are applied to validate the functionality (especially the command and decoupling behavior) of the nonlinear flight control system implementation. Fig. 10 shows the closed-loop response to a β_C = 5 deg pedal input (duration 8 sec) for the DA-42 representative reference flight condition as specified before (V_{TAS} = 139 kts, h =916.3 m, c.g. position x_{CG} = -2.3727 m, m = 1590 kg). It should be noted (as detailed within Chapter 3) that the integrator feedback signal is a scaled equivalent sideslip $\hat{\beta}_{n_y}$ proportional to the lateral load factor and hence, this signal reveals the typical all-pass behavior, initially to the wrong side due to the rudder side force opposite to the final $C_{Y\beta}\,\beta$. Moreover, the $\hat{\beta}_{n_y}$ -integrator in conjunction with the decoupling design ensures steady-state accuracy without any bank angle and restores zero lateral load factor after pedal release guaranteeing a coordinated flight (n_y=0).

Accordingly, in Fig. 11 the time histories of a p_C = 10 deg/sec square wave input (duration 2 sec, then 4 sec zero input followed by 2 sec opposite square) for the same reference flight condition are depicted. Correspondingly, the p_e-integrator provides zero steady-state error and the *MR DEA* approach in combination with the pre-decoupling "virtual controls" yield well coordinated aileron and rudder deflections assuring a pure velocity vector roll without significant (equivalent) sideslip excitation.

Currently, the controller gains are fixed and it could be shown that with such a set an acceptable behavior in terms of flying/handling qualities along the whole envelope can be achieved. However, in a next step, the control system layout will be enhanced by a straightforward gain-scheduling depending on calibrated airspeed involving a strictly limited number of grid points as preliminary investigations raise expectations that about 3 to 5 interpolation points are quite sufficient.

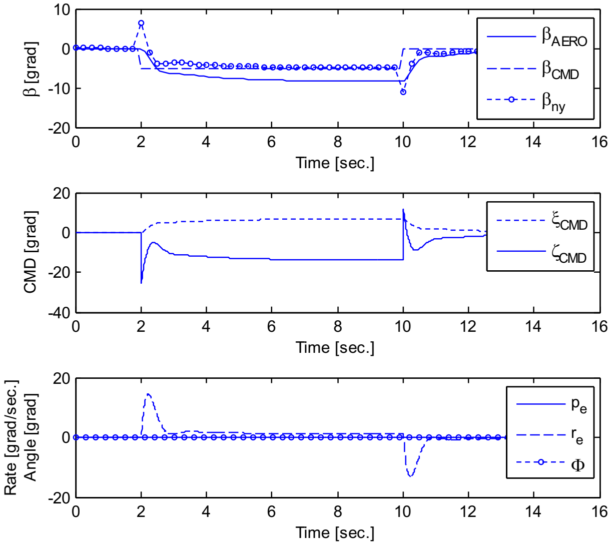


Fig. 10 Nonlinear Simulation of square β_{CMD} single axis input

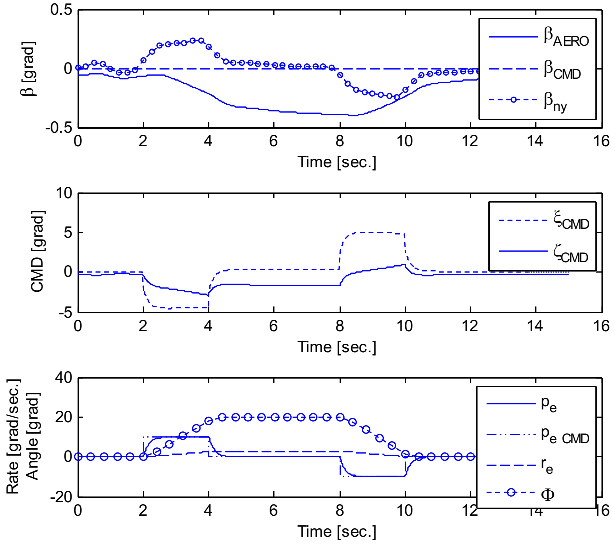


Fig. 11 Nonlinear Simulation of square p_{CMD} single axis Input

6 Conclusions

The development of a novel Fly-by-Wire Control System for the augmentation of the lateral dynamics, specifically tailored to future small general aviation aircraft is presented. This control system provides excellent flying qualities and pilot assistance under special regard of the cost-benefit ratio (i.e. acceptable engineering, implementation and certification efforts).

The unique requirements and constraints posed by the introduction of an advanced active Flight Control System into small aircraft such as *a priori* limited (by rate and position) actuator deflections or the compliance with general aviation typical sensor equipment, are addressed. Besides an appropriate feasible structural layout according to the particular needs of such a low cost/low complexity but high reliability flight control system, the feedforward and feedback gain design w.r.t. relevant flying and handling qualities criteria and its interactions are considered. Additionally, the complete development chain ranging from the nonlinear flight dynamics engineering model via the general aviation aircraft simulator up to the in-flight simulator DA-42 NG available at the Institute of Flight System Dynamics of Technische Universität München is presented.

It can be demonstrated, that current state of the art benefits of modern Fly-by-Wire technology may be provided by the active lateral Flight Control System considered by simultaneously observing the objectives and limitations specified. The control system design is confirmed by recent simulator flight tests which reveal not only a significant flying and handling qualities improvement attaining high pilot's acceptance. Additionally, the safety and assistance functionalities like turn compensation, velocity vector roll and command decoupling as well as bank angle protection excellently fulfill the pilot's demand for effective safety increase and assistance under manual control.

Consequently, the next steps after some modifications and fine tunings due extensive simulator flight tests are represented by systematic pilot testing plus evaluation and subsequently, a comprehensive stability and robustness assessment (linear as well as nonlinear) will be performed. Amongst others, this includes on the one hand a *SISO* robustness analysis based on Nichols charts (single loop cuts) which supports and facilitates the certification process. Additionally, a detailed uncertainty model will be elaborated as fundamental basis for the overall *MIMO* robustness prove (μ-Analysis) ensuring robust stability and performance along the entire envelope with special regard to typical real-word system delays and parametric as well as dynamic uncertainties.

Acknowledgments. This paper was prepared with the support of the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative.

References

1. Department of Defense, MIL-STD-1797, Military Standard, Flying Qualities of Piloted Vehicles, Washington (1995)
2. Department of Defense, MIL-F-8785C, Military Specification, Flying Qualities of Piloted Airplanes, Washington (1980)
3. Department of Defense, MIL-HDBK-1797, Military Standard Handbook, Flying Qualities of Piloted Vehicles – Handbook, Washington (1997)
4. Brockhaus, R.: Flugregelung, 2nd edn. Springer, Heidelberg (1996)
5. Garrard, W.L.: Lateral Directional Aircraft Control Using Eigenstructure Assignment. Journal of Guidance, Control, and Dynamics 21(3), 523–525 (1998)
6. Heller, M.: Untersuchung zur Steuerung und Robusten Regelung der Seitenbewegung von Hyperschall-Flugzeugen. Herbert Utz Verlag, Muenchen (1999)
7. Heller, M., Schuck, F., Peter, L., Holzapfel, F.: Hybrid Control System for a Future Small Aircraft. In: AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, AIAA-2011-6635 (August 2011)
8. Emma, K.: Australia probes high fatal accident rate in GA sector, July 19-25, p. 21. Flight International, Reed Business Information Ltd. (2011)
9. Heller, M., Myschik, S., Holzapfel, F., Sachs, G.: Low-cost Approach based on Navigation Data for Determining Angle of Attack and Sideslip for Small Air-craft. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, AIAA-2003-5777 (August 2003)
10. U.S. Department of Transportation - Federal Aviation Administration - Aviation Policy and Plans, FAA Aerospace Forecast Fiscal Years 2010-2030, pdf Document online https://www.faa.gov/data_research/aviation/aerospace_forecasts/2010-2030/ (January 22, 2013)

Dynamic Trajectory Control of Gliders

Rui Dilão and João Fonseca

Abstract. A new dynamic control algorithm in order to direct the trajectory of a glider to a pre-assigned target point is proposed. The algorithms runs iteratively and the approach to the target point is self-correcting. The algorithm is applicable to any non-powered lift-enabled vehicle (glider) travelling in planetary atmospheres. As a proof of concept, we have applied the new algorithm to the command and control of the trajectory of the Space Shuttle during the Terminal Area Energy Management (TAEM) phase.

1 Introduction

Space vehicles travel at extreme conditions of speed and acceleration that typically do not allow for a "man-in-the loop" approach, forcing, at least partially, automation of the flight controls. Thus, automated guidance and control systems are a critical component for any re-usable space flight vehicle.

For example, the implementation of control mechanisms for atmosphere re-entry and automatic landing systems used in the Space Shutle focused either on pre-programmed manoeuvres following a nominal pre-computed trajectory, or hopping across different nominal trajectories whenever the vehicle deviates from an initially selected trajectory, [11] and [4].

Rui Dilão · João Fonseca
Nonlinear Dynamics Group,
Instituto Superior Técnico,
Av. Rovisco Pais, 1049-001 Lisbon, Portugal
e-mail: rui@sd.ist.utl.pt, jlpfonseca@gmail.com

Rui Dilão
Institut des Hautes Études Scientifiques,
35, route de Chartres, 91440 Bures-sur-Yvette, France

A typical return flight from space has three main phases:

1) Atmospheric re-entry phase: In this initial re-entry phase the transition from spacecraft to aircraft flight mode occurs. The typical altitudes for this phase are in the range 120-40 km.
2) Glide to the landing site phase, usually referred as Terminal Area Energy Management (TAEM), occurring in the altitude range 40-3 km.
3) Final approach and landing phase, occurring in the altitude range 3-0 km.

While in the atmosphere re-entry phase, the biggest priority is to ensure that the structural constraints of the vehicle are not exceeded; during the TAEM phase, the biggest priority is to ensure that the vehicle reaches the Heading Alignment Circle (HAC) where preparation for landing is initiated.

On a typical mission, the TAEM phase begins at the altitude of $25,000-40,000$ m at a speed around $2-6$ M (Mach), and finishes at the HAC at the altitude of $1,500-3,000$ m, with a speed of the order of 0.20 M.

In this paper, we propose a new dynamic control algorithm in order to redirect the trajectory of gliders to a pre-assigned target point. This algorithm runs iteratively enabling a self-correcting approach to the HAC and is applicable to any non-powered lift-enabled vehicle (glider) travelling in planetary atmospheres.

This paper is organised as follows. In section 2, we present the equations of motion of a glider and we discuss the approximations we use to define the controllability conditions. In section 3, we briefly discuss the phenomenology of aircraft gliding motion, instrumental for the design of a dynamic control strategy. In section 4, we derive the dynamic control algorithm, and in section 5 we present realistic simulations for the Space Shuttle TAEM guidance and control. Finally, in section 6 we discuss the main conclusions of the paper.

2 Gliding Motion

We consider that aircraft gliding motion in a planetary atmosphere is well described by a point mass vehicle model under the influence of a gravity field, [3], [5], [9] and [2]. In this case, the equations of motion of a gliding aircraft (no thrust forces) are,

$$\left\{\begin{array}{l} m\dot{V} = -mg(z)\sin\gamma - D(\alpha,Ma) \\ mV\dot{\gamma} = -mg(z)\cos\gamma + L(\alpha,Ma)\cos\mu \\ mV\dot{\chi}\cos\gamma = L\sin\mu \end{array}\right. , \left\{\begin{array}{l} \dot{x} = V\cos\chi\cos\gamma \\ \dot{y} = V\sin\chi\cos\gamma \\ \dot{z} = V\sin\gamma \end{array}\right. \tag{1}$$

where m is the aircraft mass, $V=\sqrt{V_x^2+V_y^2+V_z^2}$ is the aircraft speed, γ is the flight path angle as defined in figures 1 and 2, μ is the bank angle as defined in figure 2c), $D(\alpha,Ma)$ and $L(\alpha,Ma)$ are the drag and lift forces induced by the atmosphere, α is the angle of attack and Ma is the Mach number. In general, the Mach number Ma is a function of V and z. The function $g(z)=g_0(R_E/(R_E+z))^2$ is the gravity acceleration, $g_0=9.80665$ m/s^2 is the Earth standard gravitational acceleration constant and $R_E=6.371\times10^6$ m is the Earth (or planetary) mean radius.

In the local reference frame of the aircraft, figure 1, $V \in (0,\infty)$, $\gamma \in [-\pi/2,\pi/2]$ and $\chi \in [0,2\pi]$. The bank angle μ is defined in the interval $[-\pi/2,\pi/2]$. In this reference frame, positive values of μ correspond to left turns and negative values of μ correspond to right turns. As usual, $(x,y,z) \in \mathbb{R}^3$ and $(\dot{x},\dot{y},\dot{z}) \in \mathbb{R}^3$. In the system of equations (1), α and μ can be seen as input parameters.

To define the local system of coordinates, we have used a flat-Earth approach. As we want to analyse the motion of gliders during the TAEM phase, the height at which the TAEM phase starts is very small when compared to the Earth radius, justifying our analysis. However, this approach can be further refined by using an ellipsoidal coordinate system adequate to Earths shape, such as the WGS-84 coordinate system.

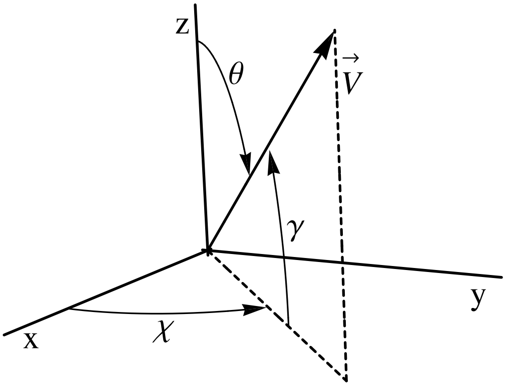


Fig. 1 Local coordinate system for the point mass glider model. The origin of coordinates is located at the centre of mass of the aircraft, and the vector **V** is the velocity vector not necessarily collinear with the aircraft longitudinal axis.

In figure 2a)-b), we show the angle of attack α defined as the angle between the longitudinal reference line of the aircraft and the vector velocity of the aircraft. In airplanes, the angle of attack is always a positive angle. While in most aircrafts attack angles are always smaller than 15^o, the Space Shuttle is capable of attack angles up to 45^o, [8] and [7]. In figure 2c), we show the bank angle, defining the inclination of the aircraft in the plane containing the velocity vector.

The drag and lift forces in the system of equations (1) are given by,

$$\begin{aligned} D(\alpha,Ma) &= \bar{q}SC_D(\alpha,Ma) = \tfrac{1}{2}\rho(z)V^2SC_D(\alpha,Ma) \\ L(\alpha,Ma) &= \bar{q}SC_L(\alpha,Ma) = \tfrac{1}{2}\rho(z)V^2SC_L(\alpha,Ma) \end{aligned} \tag{2}$$

where $\bar{q} = \rho(z)V^2/2$ is the dynamic pressure, S is the wing area of the aircraft, $\rho(z)$ is the atmosphere density as a function of altitude (Appendix) and Ma is the Mach number. For each specific aircraft, the functions $C_D(\alpha,Ma)$ and $C_L(\alpha,Ma)$ are the aerodynamic drag and lift coefficients determined in wind tunnel experiments.

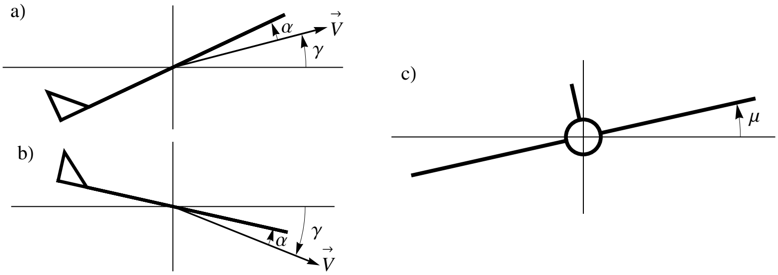


Fig. 2 In a) and b), we show the flight path angle γ and the angle of attack α of an aircraft. In c) we show the bank angle μ, measuring the inclination of the aircraft in the plane containing the velocity vector and the horizontal direction. The flight path angle γ depends on the angle of attack, on the aerodynamic coefficients of the aircraft and on the Mach number. The control of a glider is done by the manipulation of the angles of attack and bank.

Introducing the expressions (2) into equations (1), we obtain the final form for the equations of motion of a glider,

$$\begin{cases} \dot{V} = -g(z)\sin\gamma - \left(\frac{1}{2m}\rho(z)SC_D(\alpha,Ma)\right)V^2 \\ \dot{\gamma} = -\frac{g(z)}{V}\cos\gamma + \left(\frac{1}{2m}\rho(z)SC_L(\alpha,Ma)\right)V\cos\mu \\ \dot{\chi} = \left(\frac{1}{2m}\rho SC_L\right)V\frac{\sin\mu}{\cos\gamma} \end{cases}, \begin{cases} \dot{x} = V\cos\chi\cos\gamma \\ \dot{y} = V\sin\chi\cos\gamma \\ \dot{z} = V\sin\gamma. \end{cases} \tag{3}$$

where $\rho(z)$ is calculated in the Appendix.

The aircraft gliding trajectory is described by the system of equations (3), enabling a simple geometric solution of the gliding aircraft control problem.

When a glider is falling under a gravity field it converges to a steady state motion with a constant velocity and constant flight path angle given by, [1],

$$\begin{aligned} V^* &= \sqrt{\frac{2mg}{\rho S}}\frac{1}{(C_D^2 + C_L^2\cos^2\mu)^{1/4}} \\ \gamma^* &= -\arctan\frac{C_D}{C_L\cos\mu} \end{aligned} . \tag{4}$$

The geometry of the solutions of equation (3) in phase space are analysed in detail in [1].

3 Phenomenology of Space Shuttle Gliding Motion

Using wind tunnel data for the operational range of aircrafts during the TAEM phase, we have done fits for the aerodynamic drag and lift coefficients C_D and C_L of the Space Shuttle and these are well described by the parameterised functions,

$$
\begin{aligned}
C_L(\alpha, Ma) &= (a_1 + a_2\alpha + a_3\alpha^2)K(Ma)^{b_1+\alpha b_2} \\
C_D(\alpha, Ma) &= (0.01 + f_1 Ma^{f_2} + d_3\alpha^2)K(Ma)^{e_1+\alpha e_2}
\end{aligned}
\tag{5}
$$

where,

$$
K(Ma) = \frac{1}{2}\left(1 + \sqrt{\left|1 - \left(\frac{Ma}{M_c}\right)^2\right|}\right) \tag{6}
$$

is a simplification of the Van Karman functions expanded to supersonic regimes, [8]. In table 1, we show, for the Space Shuttle, the parameter estimation of expressions (5) and (6) with wind tunnel data.

Table 1 Parameters of the aerodynamic drag and lift coefficients (5) for the Space Shuttle, estimated from wind tunnel data, [6]. The significance of the fits have been determined with a chi-squared test. The large values of the absolute value of the t-statistics measures the likelihood of the parameters in the fits. The low values of the p-values mean that the fits are highly significant and the probability of finding a value outside the fitted ones are in the range $10^{-8} - 10^{-63}$.

Parameter	Estimated	Standard error	t-statistics	P-value
a_1	-0.053	0.009	-6.15	9.8×10^{-8}
a_2	2.73	0.06	43.0	1.8×10^{-43}
a_3	-1.55	0.09	-18.0	2.0×10^{-24}
b_1	-1.01	0.09	-11.3	7.4×10^{-16}
b_2	1.1	0.1	8.7	7.6×10^{-12}
d_3	1.79	0.02	99.0	1.1×10^{-63}
e_1	-1.4	0.1	-12.6	1.2×10^{-17}
e_2	1.5	0.1	11.3	5.8×10^{-16}
f_1	0.028	0.004	6.46	2.9×10^{-8}
f_2	1.4	0.2	8.57	1.0×10^{-11}
M_c	1.25	0.03	49.6	1.0×10^{-46}

Introducing the expressions of $C_L(\alpha, Ma)$ and $C_D(\alpha, Ma)$ into (4), changing the angle of attack α and the bank angle μ leads to changes in the local steady states of the glider (see (4)), enabling a guided control of the direction of motion and of the glider speed.

To control the aerodynamic behaviour of an aircraft, two main parameters are under the control of the aircraft commands: i) the bank angle μ, and ii) the attack angle α.

The bank angle μ determines the inclination of the aircraft and is used for turn manoeuvres, figure 2c).

The no-lift angle α_{nL}, the max-glide angle α_{maxgl} and stall angle α_{stall} are particular limits of the angle of attack of an aircraft, figure 2a) and 3.

In figure 3, we show, for several values of the Mach number, the behaviour of the ratio L/D, as a function of the angle of attack α for the Space Shuttle. All the curves

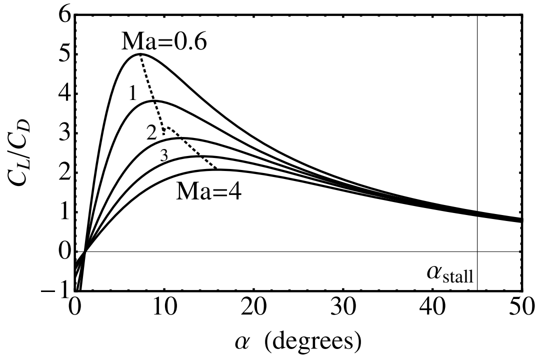


Fig. 3 Ratio L/D, as a function of the angle of attach α, for the Space Shuttle at different Mach numbers, calculated from (5)-(6) and table 1. The no-lift parameter is $\alpha_{nL} = 1.5^o$, the stall angle is $\alpha_{stall} = 45^o$ and α_{maxgl} is given by (7). The Space Shuttle is a glider and thus can only move across its L/D curve. For higher speeds this curve will become increasingly flat and the max-glide angle α_{maxgl} will move further to the right reaching saturation.

intersect at the no-lift angle α_{nL}. The no-lift angle α_{nL} is the angle for which L/D is zero due the absence of the lift force and is independent of the speed. The max-glide angle α_{maxgl} is the angle that maximises the ratio L/D, and is dependent on the Mach number. The stall angle α_{stall} is the angle at which lift dependency looses linearity and lift peaks before beginning to decrease. The stall angle is independent of the Mach number.

With the functions (5)-(6), we have approximated the max-glide angle α_{maxgl} as a function of Mach number. For the case of the Space Shuttle, we have obtained,

$$\alpha_{maxgl} = \begin{cases} 0.0906 + 0.0573Ma + 0.0071Ma^2 & (Ma \leq 1.25) \\ 0.1070 + 0.0577Ma - 0.0037Ma^2 & (1.25 < Ma < 5) \end{cases} \tag{7}$$

determined with a correlation coefficient of $r^2 = 0.999$. The Mach number is defined by $Ma = V/V_{sound}$ where the sound speed is calculated with,

$$V_{sound} = \sqrt{\gamma T(z) R_s} \tag{8}$$

and $T(z)$ is given in table 2 in the Appendix. $\gamma = 1.4$ is the diatomic gas constant and $R_s = 287.04$ J/(kg K).

4 Dynamic Trajectory Control of Gliders

A glider is not always in an equilibrium state but naturally converges to it given enough time. Our algorithm will take advantage of this behaviour by determining the equilibrium conditions needed to reach the target, imposing them on the system and letting it evolve in time.

To define the control problem, we consider the initial condition,

$$(x_0, y_0, z_0, V_0, \gamma_0, \chi_0)$$

defining the initial coordinates of the TAEM phase. Let,

$$(x_f, y_f, z_f)$$

be the space coordinates of the target, which coincide with the central point in the HAC region. We consider that the target point is only defined by the spatial coordinates of the HAC, and the direction of the velocity vector is arbitrary. In fact, this is possible at low altitudes (3 km) because the atmosphere is dense enough to allow the glider to preform turns in short distances and the vehicle is always travelling near the equilibrium speed.

The intermediate coordinates of the glider path are,

$$(x_i, y_i, z_i, V_i, \gamma_i, \chi_i)$$

where $i = 0, 1, \ldots, f$. These intermediate coordinates are evaluated at time intervals T_{con}.

In the configuration space (x, y, z), we define the direction vector from the current position of the glider to the target point as,

$$\mathbf{P}_i = (x_f - x_i, y_f - y_i, z_f - z_i) \, . \tag{9}$$

In order to direct the aircraft to the target, we control the attack and bank angles separately.

In the **attack angle heading control**, we analyse the glider trajectory in the three dimensional ambient space (x, y, z), and we command the glider trajectory path angle by controlling the angle of attack α.

In the **bank angle heading control**, the control procedures will be done in the (x, y) plane by adjusting the bank angle μ.

At the step number i of the dynamic control process, the initial conditions are $(x_i, y_i, z_i, V_i, \gamma_i, \chi_i)$. At this stage, the angle of attack and bank angle are α_i and μ_i. Then, we calculate the new values of the glider control parameters α_{i+1} and μ_{i+1} by the two procedures described below. With these new values for α and μ , the aircraft will follow a new trajectory during the time interval T_{con}, figure 4.

This control process is done sequentially in time, until the glider reaches de HAC region. In practical terms, the control mechanisms stops when the distance from the spacecraft to the centre of the HAC point attains a minimum.

We analyse now in detail the two control and command procedures for α and μ.

Attack angle heading control

The attack angle heading command and control was designed so that the vehicle is always re-orienting vertically to the HAC point through a straight line path.

The tangent of the angle between the $x - y$ projection and the z component of the direction vector P_i to the target point is computed at each iteration, and we

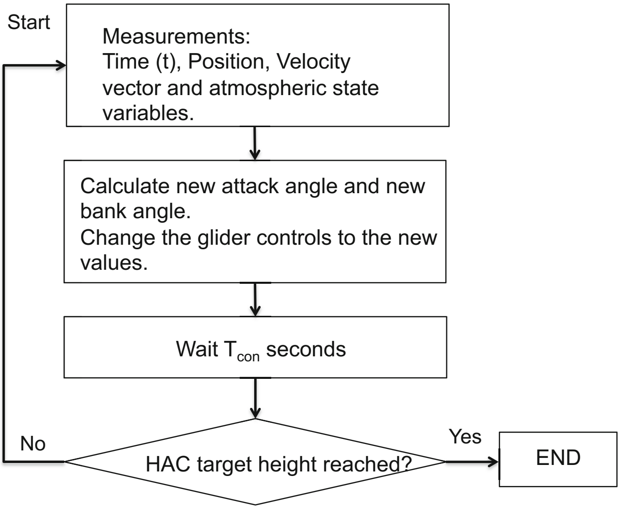

Start
Measurements:
Time (t), Position, Velocity vector and atmospheric state variables.
Calculate new attack angle and new bank angle.
Change the glider controls to the new values.
Wait T_con seconds
HAC target height reached?
No
Yes
END

Fig. 4 Block diagram for the controller algorithm

obtain,

$$G_{i+1} = \frac{(z_f - z_i)}{\sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}}$$

where (x_i, y_i, z_i) is the current position of the glider. At this position, the glider has flight path γ_i. Then, to direct the motion of the glider to the target with a steady flight path, by (4), we must have,

$$G_{i+1} = \tan\gamma = -\frac{1}{C_L/C_D \cos\mu_{i+1}}. \tag{10}$$

Assuming that it is possible to direct the motion to the target using a null bank angle, $\mu_{i+1} = 0$, we solve equation (10) in order to the ratio C_L/C_D, and we obtain the solution c_{i+1}. Then:

a) If c_{i+1} is bigger than $C_L(\alpha_{maxgl}, Ma_i)/C_D(\alpha_{maxgl}, Ma_i)$, the target cannot be reached in a straight-line and the max-glide attack angle will be selected, $\alpha_{i+1} = \alpha_{maxgl}$. The curve of C_L/C_D as a function of α and of the Mach number Ma is given by (5) and (6), and α_{maxgl} is calculated from (7) and (8).
b) If c_{i+1} is smaller than $C_L(\alpha_{stall}, Ma_i)/C_D(\alpha_{stall}, Ma_i)$, the target cannot be reached in a straight-line and the stall angle will be selected, $\alpha_{i+1} = \alpha_{stall}$.
c) Otherwise, the attack angle α_{i+1} is computed by solving the equation $C_L(\alpha, Ma_i)/C_D(\alpha, Ma_i) = c_{i+1}$.

At this stage, we have chosen a new attack angle α_{i+1}. With this new attack angle, we re-orient dynamically and vertically the aircraft trajectory to the target.

Bank angle heading control
The bank angle heading control was constructed in such a way that, in the (x,y) plan, the aircraft is always re-orienting horizontally to the HAC.

The angular misalignment between the direction vector to the target point (9) and the speed in the (x,y) plane is measured using the dot product. The direction is measured by the z component of the exterior product $(\wedge)$ between the direction vector to the target point $\mathbf{P}_i$ and the aircraft speed $\mathbf{V}_i$. With $\mathbf{P}'_i = P_{i_x} e_x + P_{i_y} e_y$ and $\mathbf{V}'_i = V_{i_x} e_x + V_{i_y} e_y$, in order to align the aircraft to the target point in the (x,y) plane, the new bank angle is,

$$\begin{aligned} \mu_{i+1}^{hea} &= -T_{hard} \arccos \frac{\mathbf{P}'_i.\mathbf{V}'_i}{||\mathbf{P}_i|| \times ||\mathbf{V}_i||} \mathrm{Sign}((\mathbf{P}_i \wedge \mathbf{V}_i)_z) \\ &= -T_{hard} \arccos \left(\frac{P_{i_x} V_{i_x} + P_{i_y} V_{i_y}}{\sqrt{(P_{i_x}^2 + P_{i_y}^2)(V_{i_x}^2 + V_{i_y}^2)}} \right) \mathrm{Sign}\left[P_{i_x} V_{i_y} - P_{i_y} V_{i_x} \right] \end{aligned} \tag{11}$$

where, we have introduced a new constant $T_{hard} \in [0,1]$. The higher this constant, the faster the vehicle will turn for the same angular deviation.

We impose now a security threshold in the bank angle, μ_{max}. A typical value for the maximum bank angle is $\mu_{max} = \pm 70^o$. Therefore, the new control bank angle is,

$$\mu_{i+1} = \min\{|\mu_{i+1}^{hea}|, |\mu_{max}|\}.\mathrm{Sign}(\mu_{i+1}^{hea}) . \tag{12}$$

5 Simulations

In the previous section, we have described a control mechanism in order to guide a glider to a target. At each time step, the algorithm determines the shortest path to the target and determines the unique values of the attitude commands of the glider that are compatible with the aerodynamic characteristics of the glider. We now test this algorithm with some numerical simulations.

We have taken the glider initial coordinates $(x_0, y_0, z_0) = (0, 0, 40000)$ m, $V_0 =$ 1000 m/s, $\gamma_0 = 0$, $\chi_0 = 0$, $\mu_0 = 0^o$, $\alpha_0 = 30^o$, $T_{con} = 0.1$ and $T_{hard} = 1.0$, and we calculated the trajectories of the glider by numerically intreating equations (3) with a fourth order Runge-Kutta integration method.

The goal was to reach some target point that we have defined as the centre point of the HAC. We have chosen three different target HAC points with coordinates,

1) $(x_f, y_f, z_f) = (200000, 10000, 3000)$ m (figure 5).
2) $(x_f, y_f, z_f) = (50000, 10000, 3000)$ m (figure 6).
3) $(x_f, y_f, z_f) = (0, 10000, 3000)$ m (figure 7).

and we have calculated the controlled trajectories from the same initial point. The arrival to the HAC point occurs when the distance from the glider to the centre of the HAC point attains a minimum. This distance error will be denoted by e_d. In figures 5, 6 and 7, we show the glider controlled trajectories as function of time and

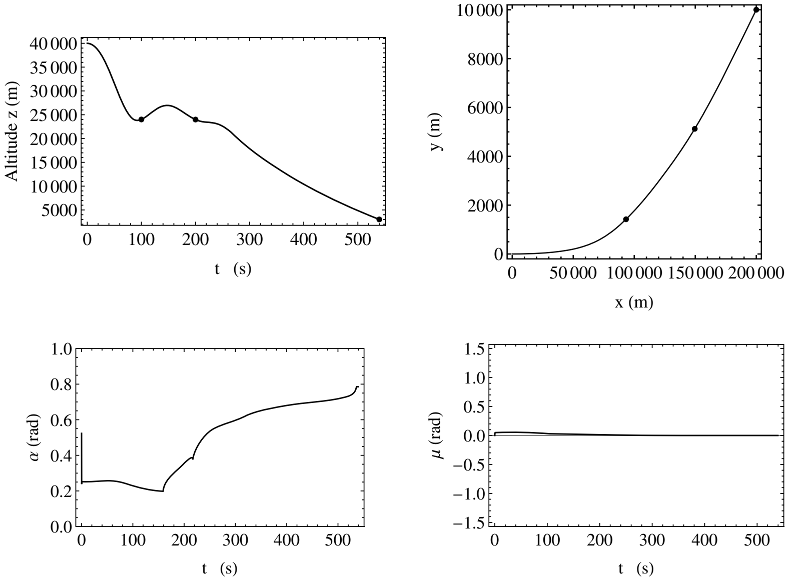


Fig. 5 Trajectory of the glider in the ambient space and control commands as a function of time. The coordinates of the HAC point target are $(x_f, y_f, z_f) = (200000, 10000, 3000)$ m. The time of arrival at the HAC is$t = 539.6$ s, with a distance error $e_d = 14.6$ m and final speed $V_f = 0.203$ M. The dots indicate the position of the glider after 100 s and 200 s of flight and the HAC position. Positive values of μ correspond to left turns and negative values of μ correspond to right turns.

the sequence of the attack and bank angle values as computed by the command and control algorithm. We have computed the time of arrival at the HAC, the final speed at the HAC (V_f) measured in Mach number units, and the distance error e_d.

The basic features of this algorithm is to guide the aircraft to the HAC point with very low distance errors. The choice of the initial conditions has been done insuring that the initial energy of the glider is enough to arrive at the target point. In this study, we have chosen target points within the maximum range calculated numerically by imposing the condition that the flight is always done with zero bank angle and maximum glide angle. In this case, the ratio C_L/C_D is maximal and the drag on the glider is minimal. For the initial conditions chosen and the Space Shuttle parameters, the range is of the order of 286 km.

Dynamic aircraft trajectories computed with the algorithm presented here depend on the control time T_{con}. For the conditions in figure 5, we have evaluated the distance error from the centre of the HAC as a function of T_{con}. For $T_{con} \leq 30$, we have found that,

$$e_d = 13.7 e^{0.049 T_{con}} . \tag{13}$$

In figure 8, we show the dependence of the distance error on the control time T_{con} for the initial and final conditions of the simulation in figure 5.

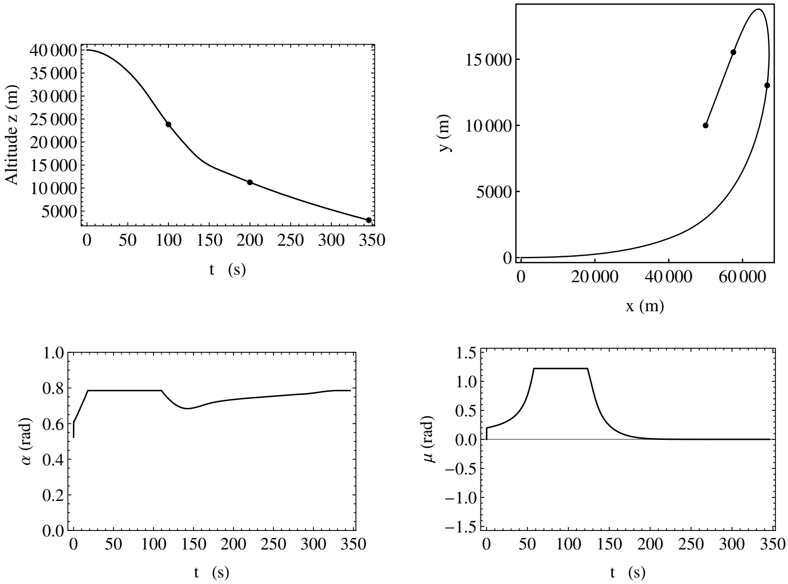


Fig. 6 Trajectory of the glider in the ambient space and control commands as a function of time. The coordinates of the HAC point target are $(x_f, y_f, z_f) = (50000, 10000, 3000)$ m. The time of arrival at the HAC is$t = 345.9$ s, with a distance error $e_d = 23.1$ m and final speed $V_f = 0.205$ M. The dots indicate the position of the glider after 100 s and 200 s of flight and the HAC position.

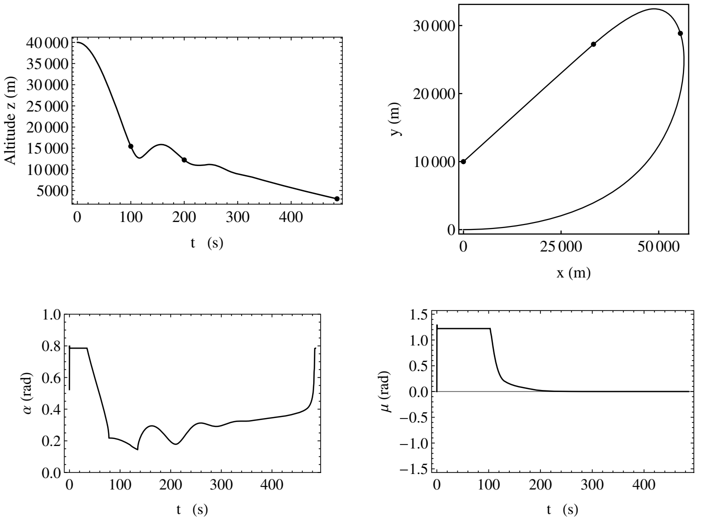


Fig. 7 Trajectory of the glider in the ambient space and control commands as a function of time. The coordinates of the HAC point target are $(x_f, y_f, z_f) = (0, 10000, 3000)$ m. The time of arrival at the HAC $t = 485.9$ s, with a distance error $e_d = 51.3$ m and final speed $V_f = 0.200$ M. The dots indicate the position of the glider after 100 s and 200 s of flight and the HAC position.

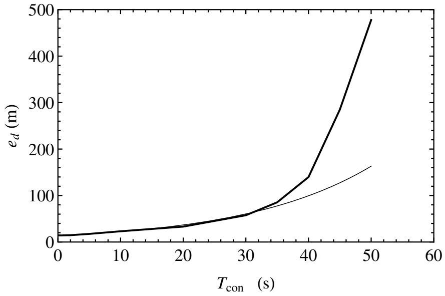


Fig. 8 Dependence of the distance error on the control time T_{con} for the initial and final conditions of the simulation in figure 5. For $T_{con} \leq 30$, the distance arrow follows the approximate exponential law (13), represented by the thin line.

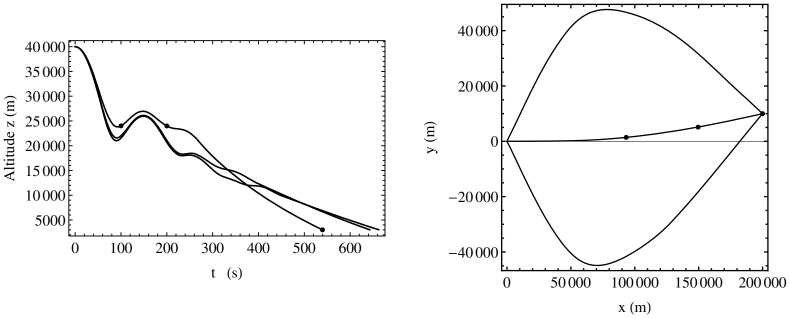


Fig. 9 Different trajectories calculated with the initial parameter $\chi_0 = \pi/4, 0, -\pi/4$. The other parameters are the same as in figure 5. For these trajectories, the distance errors are $e_d = 34.6$ m, $e_d = 14.6$ m and $e_d = 52.2$ m, respectively.

We have also tested the dependence of the controlled trajectories as a function of the entry angle χ_0. In figure 9, we show the trajectories as in figure 5 but with $\chi_0 = \pi/4, 0, -\pi/4$. In this three cases, the distance errors are $e_d = 34.6$ m, $e_d = 14.6$ m and $e_d = 52.2$ m, respectively. For larger values of angles χ_0, the distance error can be as large as 69 km ($\chi_0 = \pi/2$).

6 Conclusions

We have derived a new algorithm for the command and control during the TAEM phase of re-usable space vehicles. The algorithm determines locally the shortest path to the target point, compatible with the aerodynamic characteristics of the aircraft. We have tested the ability of the algorithm to guide the Space Shuttle during the TAEM re-entry orbit, proving the feasibility of the algorithm, even using control times of the order of 30 s. Further refinements of the algorithm are under study [1].

Acknowledgements. This work has been developed in the framework of a cooperation with AEVO GmbH (Munich) and we would like to acknowledge João Graciano for suggestions and critical reading of this paper. RD would like to thank IHÉS, where the final version of the paper has been prepared.

Appendix

The Earth atmosphere parameters are based on the 1976 US Standard Atmosphere Model. For the first seven layers we have used the formulas described in [10]. In table 2, we show the parameterisation of the thermodynamic quantities for the Earth atmosphere.

Table 2 Characteristic parameters for the lower layers of the atmosphere. z_0 is the lower altitude of the layer, $R = 8.31432$ J/(mol kg) and $R_s = 287.04$ J/(kg K) are gas constants, $M_{air} = 0.0289644$ kg/mol, $g(z) = g_0(R_E/(R_E+z))^2$ is the gravity acceleration, $g_0 = 9.80665$ m/s^2 is the standard gravitational acceleration constant and $R_E = 6.371 \times 10^6$ m is the Earth mean radius.

Layer	z_0 (m)	T_0 (K)	λ_0(K/m)	P_0 (Pa)
1	0	288.15	−0.0065	101325.00
2	11019	216.65	–	22632.10
3	20063	216.65	0.0010	5474.89
4	32162	228.65	0.0028	868.02
5	47359	270.65	–	110.91
6	51412	270.65	−0.0028	66.94
7	71802	214.65	−0.0020	3.96

Layer	T (K)	P (Pa)	ρ (kg/m^3)
1	$T_0+\lambda_0(z-z_0)$	$P_0(\frac{T_0}{T})^{g(z)M_{air}/(R\lambda_0)}$	$\frac{P}{TR_s}$
2	T_0	$P_0 e^{-g(z)M_{air}(z-z_0)/(RT)}$	$\frac{P}{TR_s}$
3	$T_0+\lambda_0(z-z_0)$	$P_0(\frac{T_0}{T})^{g(z)M_{air}/(R\lambda_0)}$	$\frac{P}{TR_s}$
4	$T_0+\lambda_0(z-z_0)$	$P_0(\frac{T_0}{T})^{g(z)M_{air}/(R\lambda_0)}$	$\frac{P}{TR_s}$
5	T_0	$P_0 e^{-gM_{air}(z-z_0)/(RT)}$	$\frac{P}{TR_s}$
6	$T_0+\lambda_0(z-z_0)$	$P_0(\frac{T_0}{T})^{g(z)M_{air}/(R\lambda_0)}$	$\frac{P}{TR_s}$
7	$T_0+\lambda_0(z-z_0)$	$P_0(\frac{T_0}{T})^{g(z)M_{air}/(R\lambda_0)}$	$\frac{P}{TR_s}$

References

1. Dilão, R., Fonseca, J.: Trajectory generation and dynamic control of unpowered vehicles during the TAEM phase (preprint, 2013)
2. Gallais, P.: Atmospheric Re-Entry Vehicle Mechanics. Springer
3. Hull, D.G.: Fundamentals of Airplane Flight Mechanics. Springer
4. Jiang, Z., Ordonez, R.: Trajectory Generation on Approach and Landing for RLVs Using Motion Primitives and Neighboring Optimal Control. In: Proceedings of the 2007 American Control Conference (July 2007)

5. Miele, A.: Flight Mechanics. Theory of Flight Paths, vol. I. Addison-Wesley, Reading (1962)
6. Ramsey, P.E.: Space Shuttle Aerodynamic Stability, Control Effectiveness and Drag Characteristics of a Shuttle Orbiter at Mac Numbers from 0.6 to 4.96, 1972 NASA/MSFC (1972)
7. Raymer, D.P.: Aircraft Design: A Conceptual Approach, 4th edn. AIAA Education Series
8. Shevell, R.S.: Fundamentals of Flight, 2nd edn. Prentice Hall
9. Trelat, E.: Optimal Control of a Space Shuttle and Numerical Simulations. In: Proceedings of the Fourth International Conference on Dynamical Systems and Differential Equations, Wilmington NC USA (May 2002)
10. US Standard Atmosphere, NASA-TM-X-74335, 1976 NASA National Aeronautics and Space Administration (1976)
11. Vernis, P., Ferreira, E.: On-Board Trajectory Planner for the TAEM Guidance of a Winged-Body, EADS Space Transportation

Cooperative Autonomous Collision Avoidance System for Unmanned Aerial Vehicle

Yazdi Ibrahim Jenie, Erik-Jan van Kampen, and Bart Remes

Abstract. Autonomous collision avoidance system (ACAS) was defined and investigated in this paper to support UAVs integration to the national airspace system. This includes not only UAVs on-board system, but also the definition of requirements, collision avoidance structure, and the avoidance rules. This paper focuses on the cooperative avoidance, where UAVs (or any aircraft) involved avoid each other using rules previously agreed by involved parties. A novel algorithm of avoidance was developed, named as Selective Velocity Obstacle (SVO) method. Several simulations were conducted and show satisfying result on how well the algorithm work to avoid separation violations. In the end of the paper, using Monte Carlo simulation, violation probabilities were derived for three setups. These simulations shows the performance of the developed algorithm for cooperative ACAS, and suggesting the need to derive a new parameter, i.e., the minimum required turning rate of avoidance.

1 Introduction

Like other technologies which were first started at a military base, UAVs will start affecting civilian live in just a couple years from now. Several industries even has been erected and commercially provides low end UAVs technologies for various non-military purpose, most of them are recreational and remote-controlled toys and fly in a secluded area with minimum impact on the airspace. However, with the fast advancement of technology, Civilian UAVs are not just toys anymore. The variation of mission that a UAV could handle became large, that government department like Police and Fire Brigade began to count the possibility of deploying UAVs more often, in a non-secluded area. DeGarmo and Nelson [4] give several predictions on what will become of UAVs in the future that affects civilians life, each of them will

Yazdi Ibrahim Jenie · Erik-Jan van Kampen · Bart Remes
Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands
e-mail: {Y.YazdiIbrahimJenie,E.vanKampen,B.D.W.Remes}@tudelft.nl

exposed a certain level of danger. The discussion of UAVs (or UAS in wider term) being used by non-military purposes become a topic of integrating UAS into the National Airspace System (NAS).

In order to be used widely in the National Airspace System, Unmanned Aircraft System required to be able to demonstrate an equivalent level of safety. This includes a solid definition of its Collision Avoidance System, which should be applicable not only between UAVs, but also take into account the already-settled manned aircraft traffic.

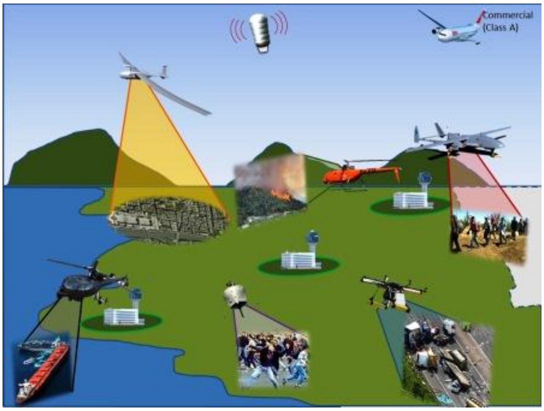


Fig. 1 DeGarmo and Nelson [4] predictions on what will become of UAVs in the future that affects civilians life

Thus, the research presented in this paper aims **to define and investigate the collision avoidance system for UAVs, in context of integrating UAVs into the National Airspace System**. This includes not only the UAVs on-board system, but also the definition of requirements, collision avoidance structure, and the avoidance rules. A mathematical model also being developed to simulate the capability of the defined system, along with several parameter derivations that described the systems level of safety. It will become clear in Sect. 2 that there will be two main part of the collision avoidance structure, the cooperative and non-cooperative avoidance. This paper, however, only focused on global structure and the cooperative part of the system. The other part will be included in the continuation of this research.

This paper presents the research as follows. After this introduction, the second section discuss the derivation of collision avoidance structure designed for UAVs to integrate with the national airspace system. In order to accommodate the cooperative avoidance, Sect. 3 would define the rules of avoidance, based on the right-of-way rules that applied in the manned-flight. The On Board Collision System for UAVs would be proposed in Sect. 4, along with the algorithms that define the avoidance criteria. Then, Sect. 5 presents the simulation on avoidance using the structure, rules and on-board system defined in the previous three sections. A mathematical model was developed for this purpose and explained briefly in Sect. 5 as well. Using Monte Carlo method, safety parameters are investigated in Sect. 6, and then the paper ends with some conclusions and suggestions in Sect. 7.

2 Defining UAVs Collision Avoidance System Structure

Manned flight established its collision avoidance system in several layers of safety. Dalamagkidis, et al., [3] described the six layers of safety that are available in manned civil flight, shown in Fig. 2. The gray area highlights the techniques to ensure separation, rather than to avoid possible collisions.

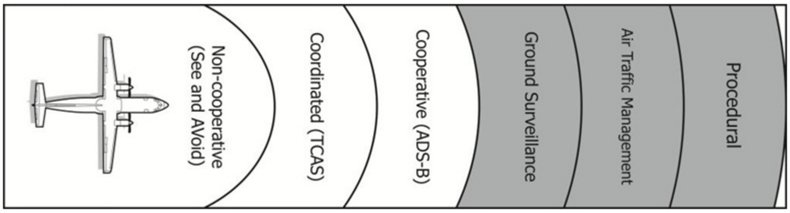


Fig. 2 Collision Avoidance System Structure for manned-flight

On Cooperative and Coordinated layer, the avoidance system designed to handle collision-probable scenario where all aircraft involved follows a same previously agreed rules. On manned flight, the Right-of-Way rules were commonly applied [10]. This rule state that when an aircraft, gets the right-of-way based on its conditions, it have privilege to continue its course, while other that do not, have to conduct necessary avoidance maneuver. Pilots in manned flight were directed by ADSB or TCAS to follow those rules. Since the ADS-B will dominated the navigations in the near future [11], TCAS layer is merged into the Cooperative layer.

For non-cooperative layer, the avoidance system required to handle more complex scenarios. These include static obstacle, aircraft that follows different rules; aircraft that does not follow any rules at all (rogue); and moreover, objects with violent intentions (aim to collide). On manned flight, there is still no specific system to provide avoidance in this layer, except to use their own pilots judgments.

In the context of integrating UAVs flight into the National Airspace system, UAVs required also to avoid collision with the already established manned-flight, besides avoiding collisions between each other. However, due to many different characteristics in UAVs compares to manned flight, several adjustments are required.

Unlike manned aircraft, which have limited manufacturers and operators, UAVs could be produced anywhere from a small scale company, and operated by almost anyone. Handling all those UAVs traffic using area-based ATM system appears to be unpractical. It is more reasonable to focus the design of UAVs collision avoidance system in the last two layers on the safety layer shown in Fig. 2.

Barfield [1] designed a comprehensive structure for as requirements for an autonomous collision avoidance system (ACAS). The structure divided the avoidance into two sphere, named de-confliction and avoidance sphere. In the de-confliction sphere, an aircraft could avoid an obstacle while still maintaining its original path. While in the avoidance sphere, Aircraft should solely escape as fast as possible.

Barfields de-confliction and avoidance sphere could be treated as cooperative and non-cooperative layer of safety, respectively. This will imply the followings:

1. The cooperative avoidance will be conducted inside the de-confliction sphere. The non-cooperative avoidance is conducted inside the avoidance sphere.
2. The cooperative avoidance will incorporate the common data of neighboring vehicle in the area (from broadcaster i.e. ground surveillance or GPS) and apply the Right-of Way rule (also adjusted for UAVs later in Sect 3). The non-cooperative avoidance should also use any on-board sensor available and avoid the non-cooperative vehicle using somewhat more loose rules.
3. The cooperative avoidance is a de-confliction maneuver that still takes into account the original flight path, with the point to start the maneuver could take place anywhere in the de-confliction sphere. The non-cooperative avoidance is an aggressive maneuver aims solely to escape as fast/soon as possible and neglects its original flight path.
4. The cooperative avoidance maneuver should in any case avoid the violation of the avoidance sphere. The non-cooperative avoidance should in any case avoid collision with obstacle. Turn rate requirements for avoidance could be set base on this.

Although it is not explicitly described, Barfield choice of for the radius of the spheres (1.5 second and 25 seconds) might be derived from manned flights TCAS. Therefore, as shown in Fig. 3, the Traffic Warning sphere is introduced to complete the sturcture. This outer sphere spans until 40 second distance. In this final sphere,

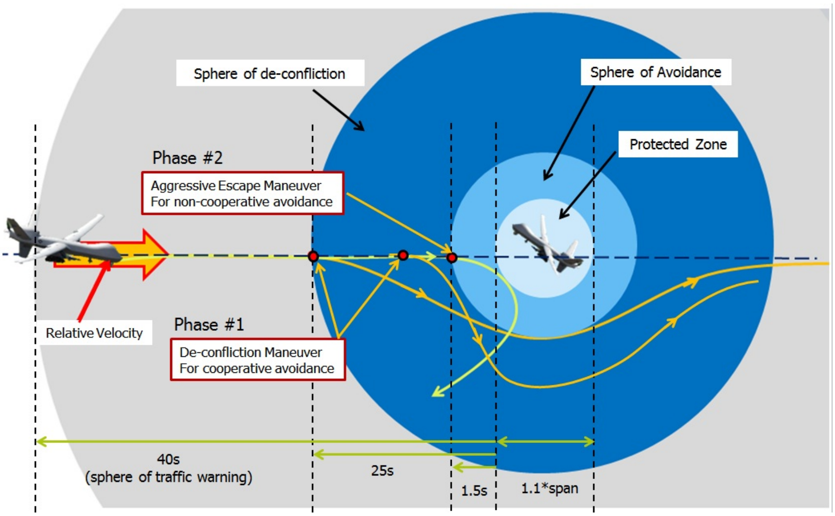


Fig. 3 Concept of Collision Avoidance System Structure on UAVs

the Collision Avoidance system should begin to give warning to operators about the traffic ahead.

These (1) Traffic Warning, (2) De-confliction, and (3) Avoidance -Sphere define a novel structure of Collision Avoidance System for UAVs, in the context of integration to the National Airspace System. This structure should work seamlessly with the manned-flight, since it uses the same parameters they have already established.

3 UAVs Cooperative Avoidance Rules

As explained in the introduction, this paper will present the cooperative part of UAVs avoidance system, where the avoidance maneuvers are based on common avoidance rules. Similar with the avoidance structure described in the previous section, it is best to start defining the rule from the already established rule in the manned flight, stated in [10]. The following sub section will describe the suggested rule of cooperative avoidance in two parts, i.e., category priorities and situational priorities.

3.1 Category Priorities for UAVs

With the large variation of UAVs, it is only logical to set some category priorities for them. Many documents have presented classifications of UAVs, especially based on its dimension (size) or weight, e.g. CAP 722 [9].

Quite different, on the category priorities, manned flight use the performance of aircraft category; aircraft that have slower or lower performance in maneuvering will get the right of way [10]. Based on this, UAVs need to be categorized based on performance. Furthermore, since the Collision Avoidance structure defines in the last section is based on time-described distances, velocity would be a good parameter for the categorization.

Spreading out the CAP 722 classification that based on weights, it appears that UAVs could easily be categorized by its cruise velocity. The new classification that based on velocity is listed in Table 1. The class on the upper row will always have right of way (priority) to the lower rows.

Using the velocities limits, the structure of collision avoidance system could be easily defined for each categories, or for between categories. Derivation of the spheres radius (when each category meets an assumed static object) could be observed also in Table 1.

In scenario when a UAV from one category meets another category UAVs (i.e. A Small-Slow UAVs face a Light UAVs), the spheres radius will change according to the relative velocity limit of both UAVs. Table 2 shows calculation result for the avoidance sphere radius, in case where each category meets one another. Some starred column indicates the unlikely-to-happen scenario due to difference on operation altitudes. On the continuity of this research, analysis will be extensively focused on the Small Slow UAVs, especially to plan the real-world experiments on the avoidance concepts.

Table 1 UAVs classifications, base on Velocity

CAP 722 Classifications	Velocity Classification	Velocity [km/h]	Velocity [m/s]	Sphere Radius [m] (with assumed Static Object) 1.5s[a]	 25s[b]	 40s[c]
Small UAVs	Small Slow UAVs	< 50	< 13.89	20.83	347.22	555.56
	Small Fast UAVs	< 100	< 27.78	41.67	694.44	1111.11
Light UAVs	Light UAVs	< 250	< 69.44	104.17	1736.11	2777.78
Large UAVs	Large Slow UAVs	< 500	< 138.89	208.33	3472.22	5555.56
	Large Fast UAVs	> 500	> 138.89	416.67	6944.44	11111.11

[a] Avoidance Sphere; [b] De-confliction Warning Sphere; [c] Traffic Warning Sphere; (see Sect. 2)

Table 2 Avoidance sphere radius for each categories encounter

in encounter with	Static Object	SS -UAVs	SF -UAVs	L -UAVs	LS -UAVs	LF -UAVs
SS[a] -UAVs	20.83	41.67	62.50	125.0	229.17*	437.5*
SF[b] -UAVs	41.67	62.5	83.33	145.83	250.0*	458.33*
L[c] -UAVs	104.17	125.00	145.83	208.33	312.5*	520.83*
LS[d] -UAVs	208.33*	229.17*	250.00*	312.5*	416.67	625.00
LF[e] -UAVs	416.67*	437.5*	458.33*	520.83*	625.00	833.33

[a] Small Slow; [b] Small Fast; [c] Light; [d] Large Slow; [e] Large Fast
*highly unlikely encounters

Another priority that needs to be defined is the interaction with manned aircraft. Barfield proposed UAVs to follows Asimovs three robotic laws [1]. In short, UAVs should always give the right of way to manned aircraft, regardless their velocity or weight.

3.2 Situational Priorities for UAVs

The situational priorities in UAVs flight could easily be adopted from the manned-flight. This is true especially for the cooperative collision avoidance. The summary of these rule listed as follows:

1. On converging encounter, the one on the right hand have the right of way.
2. On head-on encounter, both aircraft should move to the right side.
3. The one that are about to be taken over have the right of way.
4. Avoidance should not go over or under, or in front of other aircraft that have right of way, except when it is clear.

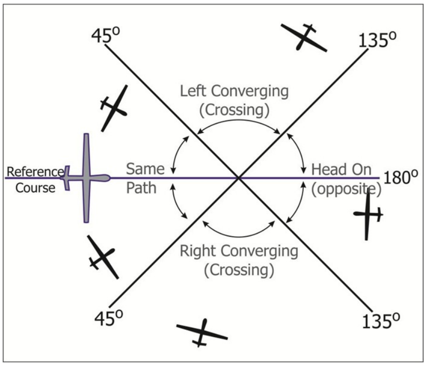


Fig. 4 Flight path definitions for manned air traffic control, adapted from [12]

For UAVs system, the converging, head-on, and taking over encounter need to be defined quantitatively. One way to define those is to use the definition of crossing, opposite and same flight path for manned flight Air Traffic control, stated in [12] which are described in the Fig. 4.

The adaptation of those definitions in the UAVs ACAS is described thoroughly in the next section.

For non-cooperative avoidance, on the other hand, definitions of its situational priorities will not be discussed further in this paper; instead it will be investigate on the continuation of this research.

4 Defining the On-Board Collision Avoidance System for UAVs

Based on the collision avoidance structure and rules, an on-board collision avoidance system functional concept is derived in this section. The design where influence by the twelve requirements set by Barfield [1].

4.1 System Functional Concept

An autonomous system for collision avoidance (ACAS) was highly suggested for UAVs applications, including in [1], since the task of avoidance in UAVs could not be handled only by pilot/operators. This is due the fact that the UAVs operator will only manage the UAVs flight to finish it mission autonomously, and even if there are such ground pilots controlling the UAVs, they do not have the required awareness of the surroundings.

Nuisance free is another requirement that need to be fulfilled by the UAVs ACAS. This means that the ACAS should be separated from the normal control system that is operating the UAVs, and only interferes when its needed. Interrupt and restore criteria should be defined for this purpose. In accordance to this, warning cues to the pilot when the system detects traffics are also required. In Fig. 5, these concept where compactly drawn, with also highlighted the used of ADS-B.

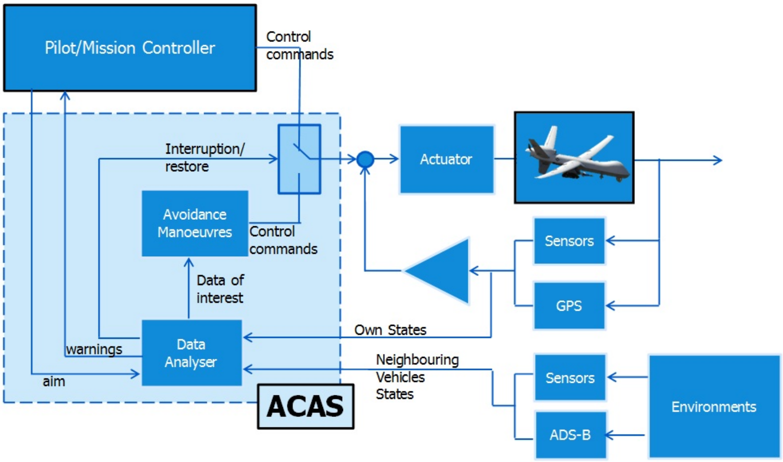


Fig. 5 Cooperative ACAS system concept, integrated with the normal mission controller of a UAVs

4.2 *Avoidance Algorithms: Selective Velocity Obstacle Method*

A method called the Velocity Obstacle (VO) Method [2,6,7,8], or sometime the Forbidden Zone Beam Method [5], is used to define avoidance criteria. The VO-method was chosen due to its simple implementation and geometrically understandable. A complete explanation of the original VO-method could be found in [6]. To be suitable for the implementation in UAVs ACAS, including adopting the rules described in previous section, several modifications were made, producing a new branch of the Velocity Obstacle Method, which from this point, will be referred as the Selective Velocity Obstacle Method (SVO).

4.2.1 Velocity Obstacles (Original) in UAVs Collision Avoidance System

This section presented the explanation of the original VO [6] in context for UAVs ACAS applications, explained in previous sections. Since the focus is to set an algorithm in each UAVs separately, the own UAVs (should-avoid) and obstacle are treat differently. This original VO will be referred as OVO.

First we designated A_o and A_i to symbolized the should-avoid agent and the obstacle agent, respectively. Let S_{a_i} be the avoidance sphere, centered by the A_i position X_i, and moving with constant velocity V_i. Let X_o be the position of A_o, moving with constant velocity V_o. According to OVO method, to decide if these two agents are on a collision course, it is sufficient to consider their current positions together with their relative velocity $V_R = V_o - V_i$. If we elongate the V_r from X_o by a sufficient positive scaling (symbolized as $\lambda_R = \{X_o + \mu V_R | \mu \geq 0\}$), it is clear that the two agent are on a collision course, if and only if λ_R cuts the area S_{a_i} or formally, $S_{a_i} \cap \lambda_R \neq 0$. The set of that cuts is called collision cone CC_{oi}.

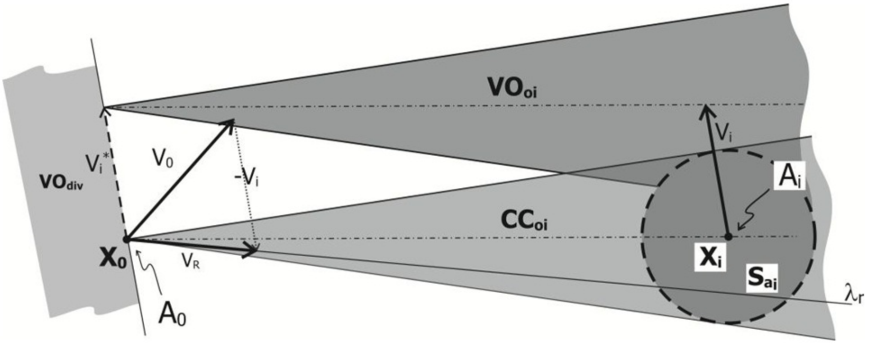


Fig. 6 VO cone definition in the original concept, adapted from [6]

To be able to decide directly whether V_o will collide or not, it was suggested to define the so called velocity obstacle set/cone of S_{A_i} from X_o, as:

$$VO_{oi} = \{V_o|(V_o - V_i) \in CC_{oi}\} \tag{1}$$

Or,

$$VO_{oi} = V_i + CC_{io} \tag{2}$$

Thus, for A_o, any velocity $V_o \in VO_{oi}$ from X_o,will lead to a violation on S_{a_i}, and any velocity $V_o \notin VO_{oi}$ will avoid those violations.

In reality, it might happen that A_o was confronted with more than one obstacle. In this general cases, Let $i = 1,2,3,...,n$, the number of obstacle under consideration. The velocity obstacle that A_o need to define for all the obstacle is simply the union of each velocity obstacle,

$$VO = \cup_i VO_{oi} \tag{3}$$

For any velocity $V_o \notin VO$ from X_o, A_o will not violate any S_{a_i}, where $i = 1,2,3,...,n$.

Figure 6 also shows another area named the VO diverging area, VO_{div}. This area defined as one of two areas separated by the infinite elongation of vector V_i through X_o, that does not contain any set of VO. Fiorinni [6] already define this area as a set of vector that A_o could chose to diverge completely from the obstacle. However, this area has not been employed in any of VO previous research. It will become clear that VO_{div} could set a handy definition on the avoidance maneuver.

On the OVO, a simple navigation scheme based on which velocity could be chosen to ensure no collision is used. The position and velocity of each agent were continuously tracked, and all information was used to update V_o. The velocity is chosen based on the goals of the agents, for example to avoid while still in the same path, or taking its maximum velocity to avoid each other.

4.2.2 Selective Velocity Obstacle (SVO)

SVO was designed to accommodate rules and requirements of the UAVs ACAS system. The idea is to selectively use any VO area developed around the velocity vector V_o, based on the position of each VO position from X_o. Using this, the algorithm will select which VO should be avoided, and which VO could be ignored. These areas, different from VO, relate to the obstacle velocity shadow, V_i^* from X_o, or, the origin of each VO. The additional areas explained here were meant to represent the rules described in Sect. 3, however, could easily be modified for other rule schemes.

First we define two circle centered by X_o,i.e., S_{V_o}, and S_{cat1}, with radius of V_o, and V_{cat1} respectively. V_{cat1} is the velocity limit of a UAV category explained in Subsect. 3.1, which for the Slow-Small UAVs, is 13.89 m/s. Next, using A_o motion axis (or wind-axis) as the frame of reference (where V_o is pointing up), we divide S_{cat1} into four equal set of velocity vector coming from X_o, named S_{r1}, S_{r2}, S_{r3}, and S_{r3}, as shown in Fig. 7. Notice that this represents the flight path definitions explained in Subsect. 3.2.

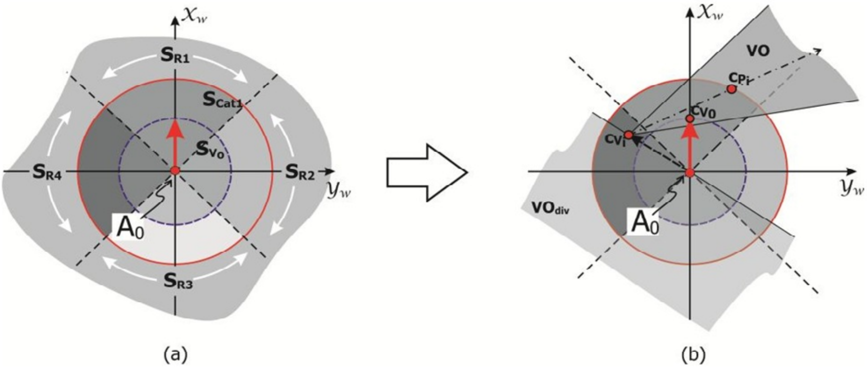


Fig. 7 Selection Circle on SVO (a) Area definition; (b)VO (and VO_{div}) implemented.

Lastly, we define three points that will set the criteria, c_{V_o}, c_{V_i}, and c_{P_i}. c_{V_o} is simply the end of V_o vector from X_o. c_{V_i}, on the other hand, is the end point of the shadow of the obstacle velocity, V_i^* from X_o, or simply, the origin of the Velocity Obstacle VO. C_{P_i} is the intersection point of VO axis with the edge of S_{cat1}. This last point is not really necessary, and could be replace by the real position of the obstacle, X_i. However, it is added for a compact figure and explanations.

Next section will describe how the additional areas were used to selectively treat the Velocity Obstacles.

4.2.3 Algorithm for the Selective Velocity Obstacle

With those setups, we could finally define the algorithm required to accommodate all rules into the UAVs ACAS via this Selective Velocity Obstacle Method.

As mentioned before, the algorithm is designed to still give UAVs freedom to choose their own avoidance maneuver, as long as they follows the rules explained in Chap. 3. Generally, there will be three main maneuver type that UAVs ACAS need to handle, which are (1) Avoid, (2) Maintain, and (3) Restore, denoted as q_1, q_2, and q_3, respectively. Restore here means that the ACAS give back the control to the original controller/pilot so the UAVs could continue its mission. The ACAS itself need only to define what maneuver it should take for the Maintain and Avoid.

Thus, the avoidance rules for the SVO for cooperative avoidance of UAVs, for category one (Slow-Small UAVs) are mathematically modeled as follows:

$$\begin{cases} q_1, \text{ if } & c_{V_o} \in VO_{oi} \wedge c_{V_i} \in S_{cat1} \cap \left(S_{r3} \cup S_{r4} \cup \left(S_{r1} \cap S_{V_0}\right)\right) \\ q_2, \text{ if } & c_{V_o} \notin VO_{oi} \cup VO_{div} \wedge c_{V_i} \in S_{cat1} \\ q_3, & \text{otherwise} \end{cases} \tag{4}$$

Here, the velocity obstacle only need to be avoided when the origin of any VO (c_{V_i}) lies inside S_{r3}, representing head-on encounter, inside S_{r4}, representing right-encounter, or inside $S_{r1} \cap S_{r2}$, which simply represent a take-over maneuver of a slower vehicle in the same path. Notice that these algorithms only activated when c_{V_i} is inside S_{cat1}, interrupting the normal controller. In case of c_{V_o} already escapes VO_{oi} but still not inside VO_{div}, the system will treat it as not safe enough to give back the control to the original controller, and instead, it maintain its course and wait for any event that still could happen, including being back again inside VO_{oi}. Only when c_{V_i} is inside VO_{div}, should the restoration maneuver happen.

As it might have been notice, SVO also discard the set of reachable velocities that originally used in the OVO [6]. The main reason of this is the fact that UAVs commonly use rotation as the control input for maneuvering, instead of arbitrary velocity vectors. Thus, SVO describe a turning rates (ω_{avo}) required for avoidance maneuver, which will depend on velocities, distances and positions. This turning rate will be derived on the continuation of this research.

5 Implementations

Using the defined collision avoidance structure in Sect. 2, the cooperative avoidance rule in Sect. 3, and the on-board ACAS system and algorithms in Sect. 4, several computer simulation were conducted. A MATLAB program was developed and designed to be highly customizable that it could accommodate any initial positions and velocities, avoidance rules and algorithms used, the UAVs involved dynamics, normal control systems, and many more. This MATLAB program is still on-going development and will also be used in the continuation of the research.

5.1 Mathematical Model and Simulation Setup

Since it will be applied in a relatively large area, we could treat vehicles involved as a point mass, eliminating the need to model each aircraft dynamics. The mathematical model of each aircraft motion was linear, discrete and single phased, focusing more on the development of the right algorithm to accommodate avoidance. Position and velocity data of each aircraft were broadcasts between each other in same time step, simulating the use of ADSB that support this cooperative avoidance.

Depends on how many agents involved in a scenario, the MATLAB program first generate them as an object that embedded these linear discrete equation that describe each agent propagations through the simulation.

$$\begin{aligned} &x(k+1) = Ax(k) \\ &x = \begin{bmatrix} x \\ y \\ V_x \\ V_y \end{bmatrix}; A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & -\omega\Delta t \\ 0 & 0 & \omega\Delta t & 1 \end{bmatrix}; \omega = \begin{cases} \omega_{avo}; & \text{if } q_1 \\ 0; & \text{if } q_2 \\ \omega_{Goal}; & \text{if } q_3 \text{ or } q_{init} \end{cases} \end{aligned} \tag{5}$$

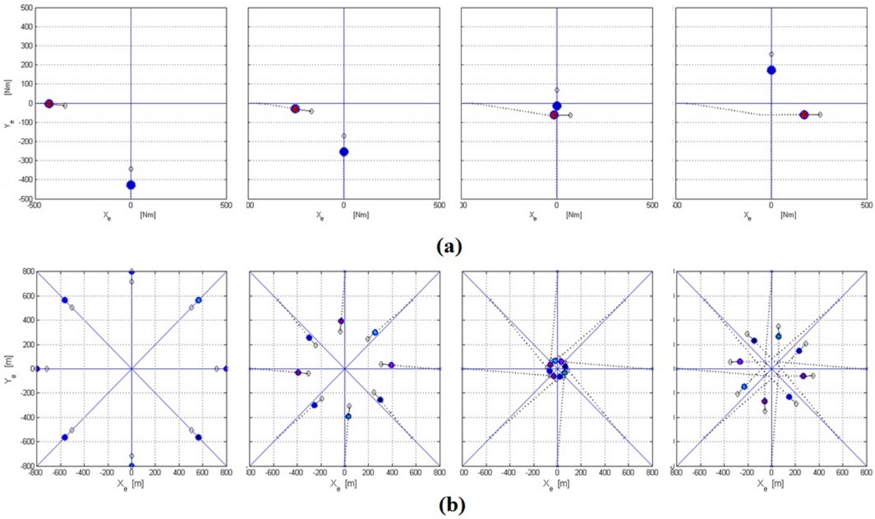
Inputs for Eq. 5 were highly depends on the result from SVO algorithm, explained before (q_1, q_2, or q_3). q_{init} is simply the initial setup before any detection of obstacles. In the conducted simulation, these values are simply the direction to each agent original end point. ω denotes the modes turning rates, where it is ω_{avo}, 0, ω_{Goal} on mode q_1, q_2, and q_3, respectively. ω_{avo} was assumed to be 5 deg/s (0.0873 rad/s) for every agent. ω_{Goal} obtained from any normal controller that is used, that guides the UAV back to its original mission. In this research, ω_{Goal} simply direct each UAV to its original way points.

Unit time step ($\Delta t = 1$ second) was used for every simulation, in assumption it also match the ADSB update rates. For simplification on these preliminary simulations, all avoidance happens on the edge of de-confliction sphere. Lastly, all agent considered is a Category 1 UAVs, the Slow-Small UAVs (see Subsect. 3.1).

5.2 Simulation Results

There are unlimited collision scenarios which could be tested, even though only working on one UAV category. A few important scenarios were presented in this paper, selected according to the converging, head-on and same path areas described before in Sect. 3. The entire results are presented using agent position time-captures from above (top view) on four important positions. The arrow on each agent represents the velocity vector. Notice that the entire rules described in Sect. 3 were fulfilled for each avoidance.

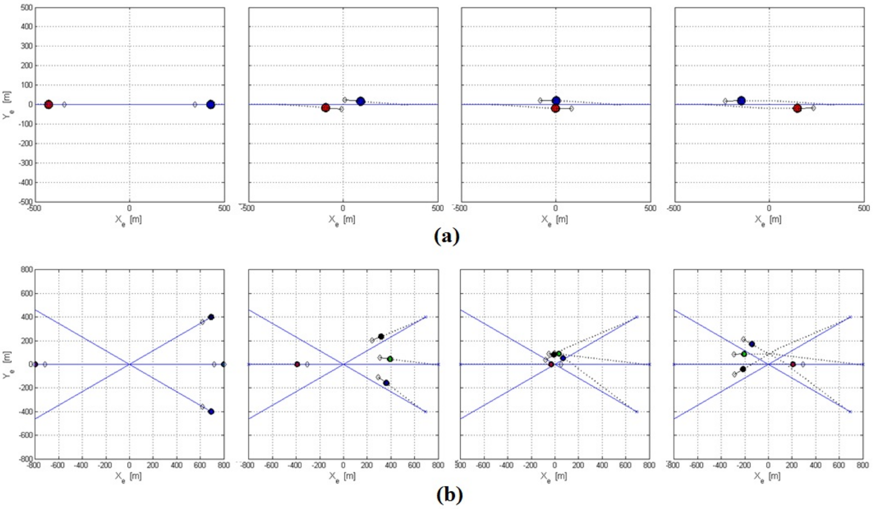
On Head-On encounter (Fig. 9), since both agents are avoiding each other to the right, the course deviation is not as large as the converging case (Fig. 8). Interesting to observe in Fig. 10-b that the agent heading to the right did not conduct any avoidance; instead, it goes straight as its original course. Analysis revealed that this happens because the other three agents on the opposite are closer to each other, and start avoiding each other sooner. Those maneuvers create a situation where the one agent heading to the right will not collide at all, and hence it keeps it original flight path.



(a)

(b)

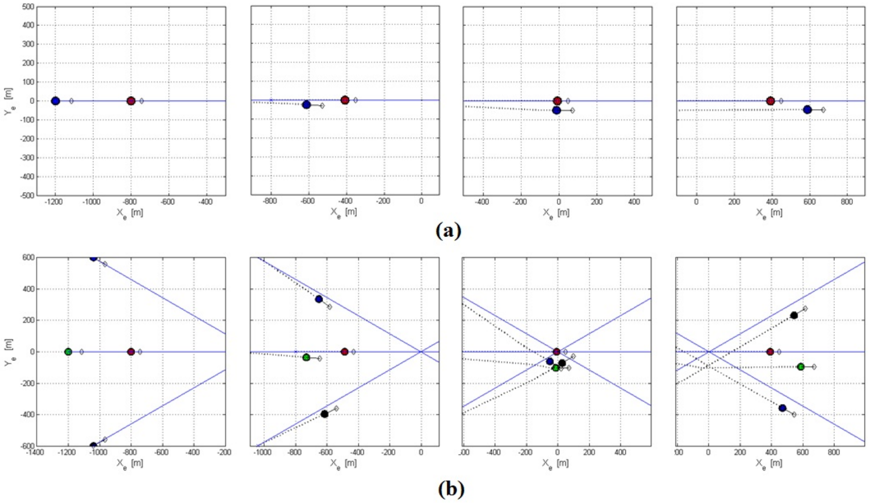
Fig. 8 Simulation on converging encounter scenario; (a) two-agents, 90^0 encounter from the right, (b) eight-agents, symmetrical circle encounters



(a)

(b)

Fig. 9 Simulation on Head-On encounter scenario; (a) two-agents, directly Head-On (b) four-agents, 30^0, 0^0 and -30^0 encounter forward



(a)

(b)

Fig. 10 Simulation Take-Over scenario; (a) two-agents, same path (b) four-agents, 30^0, 0^0 and -30^0 encounter from behind

On Fig. 10, to be able to simulate a taking over encounter, a different velocity is required. Therefore, one agent, which will be taken over, has 8 m/s velocity, as opposed to other agents behind it that use 12 m/s. In the end, all taking-over where successfully conducted, even when there are more than one agent taking over.

6 Violation Probability (Using Monte Carlo Simulations)

The entire simulations in Sect. 5 were conducted smoothly without any avoidance sphere violations. However, these results not necessarily mean the avoidance system and algorithm guaranteed to works for every scenario. Therefore, this section will present Monte Carlo simulations where a large number of random scenarios were tested, in order to find the violation probability of the avoidance. The derivations were conducted for two, three, four and five UAVs (agents). Similar with the simulations in Sect. 5, this violation probability derivation in this paper will only discusses the first category of UAVs.

The derivation of violation probability shows how well the performance of the algorithms developed, and even, act as a tool to find any scenario that make the algorithm fail. In accordance to the Equivalent Level of Safety (ELOS), this violation probability needs to be zero. ELOS are based on the failure of the system due to time. The algorithm itself should be guaranteed to solve any scenario possible.

6.1 Monte Carlo Simulation Setup

To assess the performance of both proposed system structure and the algorithm (and rules), several parameters were introduced. The cooperative ACAS performance was measured using the probability of separation violations, P_{vio}, formulated as:

$$P_{vio} = \frac{N_{vio}}{N_{MC}} \tag{6}$$

Where N_{vio} and N_{MC} denotes number of scenario that collision happen and number of Monte Carlo samples, respectively. The value of P_{vio} will fluctuated with N_{MC}, and as N_{MC} become larger, it should converge to a certain value, which then defined as the final value.

Other parameter to set up the Monte Carlo simulation are the selected area of interest, A_{int}, the area of separations, A_{sep}, and the area density, $\rho_{A_{int}}$, formulated as:

$$\rho_{A_{int}} = \frac{NA_{sep}}{A_{int}} \tag{7}$$

Where N denotes the number of agents involved. Notice that A_{sep} is a circle area with radius of half of the de-confliction sphere, conserving the total de-confliction distance.

The position (x_n, y_n) of each agent is randomized on the X-Y planes, while keeping no violation in the beginning of simulation. The x_n, y_n position is assumed to be spread randomly in a square, instead of a circle area, for simplifications. As can be observed in Table 3, the position range is set according to the number of agents, and the radius of Traffic Sphere used (the 40s sphere, see Sect. 2), which have radius r_{tra}. Consequently, A_{int} and $\rho_{A_{int}}$ also depends on this sphere, where becomes constant for every number of agents involved, set at 0.3. Two other initial parameters were randomized as well, the headings (ψ_i) and the velocity magnitudes (V_i), detailed in Table 3. Using these setup, it is possible to have a scenario where the agents are not bound to violate each other, and thus make it possible also to derived the violation probability where ACAS is not implemented.

All avoidance maneuver used the same turning rate of avoidance (ω_{avo}), i.e., 5 degree/seconds, or 0.0873 radian/seconds. Furthermore, all avoidance starting point take place on the edge of the de-confliction sphere (25 seconds sphere).

Results of this Monte-Carlo simulation (coded MC01) are presented in the next section (Fig. 12 and 13). Those results, however, neglect the freedom that each cooperative agent should have, to choose their own avoidance maneuver. Therefore, another Monte-Carlo simulation was conducted (coded MC02), with one more randomized variable, D_{avo}, which denote the ratio of avoidance starting point with the de-confliction sphere radius. The turning rate of avoidance (ω_{avo}), however, was still assumed to be 5 degree/second.

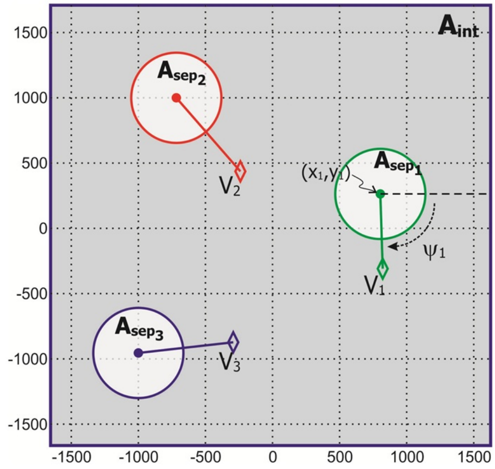


Fig. 11 Random scenario of encounters (e.g. 3 agents involved) for Monte Carlo simulation setup

Table 3 Ranges of randomized parameters for Monte Carlo simulations

Variable	Range / Value	Length
Positions (x_n, y_n)	$\left[-\frac{1}{2}N \times r_{tra}, \frac{1}{2}N \times r_{tra}\right]$	[m]
Velocity Magnitude (V_i)	$[8, 13]$	[m/s]
Heading (psi_i)	$[-\pi, \pi]$	[rad]
Avoidance Point (D_{avo})	$[1, 1]^a; [0, 1]^b$	[-]
Monte-Carlo Samples (N_{MC})	10^6	samples

[a] for MC01; [b] for MC02;

6.2 Results and Analysis

Figure 12 shows the results of P_{vio} versus the number of Monte-Carlo samples. The figures compiles results from both Monte-Carlo simulations (MC01 and MC02), with addition of the calculation result where no ACAS is used (coded MC00). It could be observed that the Monte Carlo simulation produced convergent results even before the number of samples of 10^6. This result holds for each agent number configurations. Figure 13 shows the final violations probability for each number of involved agents, for MC00, MC01 and MC02.

Several analyses were made based on the Monte Carlo simulation. First one is regarding the Area Density parameter $\rho_{A_{int}}$. Evidently, this parameter is less dominant than the number of agents involved; even though the area of interest (A_{int}) enlarged as more agents involved, violation probability (P_{vio}) still become larger. This may be caused by the encounters combinations between agents in the area.

MC01 results was satisfying, resulting zero violations for every number of agents scenario. MC02, however, only shows violations reductions that is still

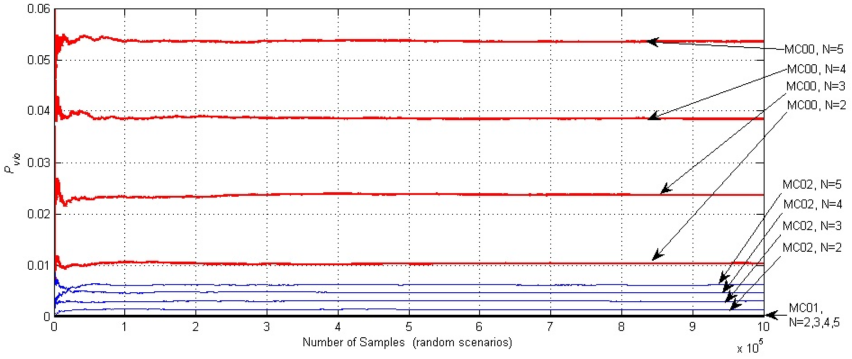
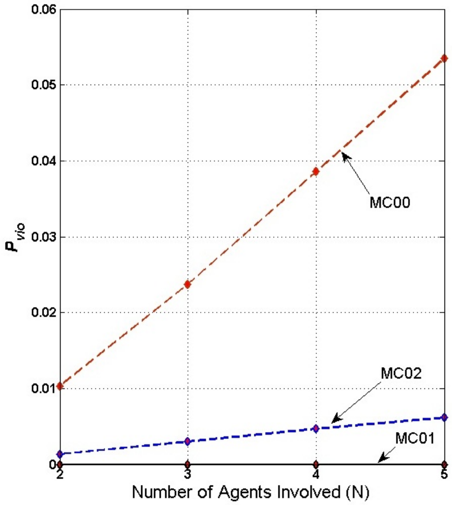


Fig. 12 Monte Carlo simulation convergence results on Probabilities of Separation Violation for two-, three-, four- and five- agents, on MC00, MC01, and MC02

Fig. 13 Collision Probabilities comparison with the use of cooperative ACAS (MC01 and MC02) and without (MC00).



unacceptable. On observations on those failed cases, it was concluded that distance might be the problem, since every failure happen at Avoidance Point (D_{avo}) lower than 0.5. This also explain why MC01 results zero violation; MC01 only use $D_{avo} = 1$. Observation on those failed scenario also reveals that agents are indeed avoiding, however, the distances were too close, and the avoidance is not fast enough. This suggest the need to adjust the avoidance turning rate (ω_{avo}) according to D_{avo}. If the adjustment of ω_{avo} could be derived, it could be set as a requirements for these cooperative avoidance between UAVs, the minimum required turning rate, $\omega_{a.r}$. This derivation, however, will not be discussed in this paper.

7 Conclusions

Several concluding remarks could be summarized form this research, including as follows:

1. In order for UAVs to be integrated to the national airspace system, a complete collision avoidance system was investigated. This include not only the system on-board a UAV, but also the structure and rules that could define a common guideline for UAVs avoidance.
2. The structure of the collision avoidance system for UAVs is then divided into two main parts, the cooperative part, which was in accordance to a de-conflicting maneuver, and the non-cooperative part, which will use an aggressive avoidance maneuver. This paper, however, only continue to focus only on the cooperative collision avoidance system.
3. To support the Cooperative Collision Avoidance, several ground rules were defined based on the rules of the air in manned-flight.
4. Finally, a functional concept for the on-board system was defined, incorporating several requirements. A Novel algorithm for cooperative ACAS for UAVs, named Selective Velocity Obstacle (SVO) method, was introduced.
5. A MATLAB program was created as a tool to simulate various scenario of collision. All simulation of the selected scenarios were conducted smoothly and the use of designed cooperative ACAS evidently could prevent separation violations.
6. To quantitize the probability of violations, and then state the performance of the designed cooperative ACAS, a Monte Carlo simulations were conducted. The results suggest the need to derive a minimum requirements for avoidance turning rate, $\omega_{a.r}$, base on distances of avoidance.

References

1. Barfield, F.: Autonomous collision avoidance. The technical requirements. In: National Aerospace and Electronics Conference (NAECON), vol. 1, pp. 808–813. IEEE, Dayton (2000)
2. van der Berg, J., Lin, M., Manocha, D.: Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. In: IEEE International Conference on Robotics and Automation, Pasadena, CA, USA (2008)
3. Dalamagkidis, K., Valavanis, K.P., Piegl, L.A.: On Integrating Unmanned Aircraft Systems into the National Airspace System. In: International Series on Intelligent Systems, Control, and Automation: Science and Engineering, vol. 36. Springer Science+Business Media, B.V. (2009)
4. DeGarmo, M.T., Nelson, G.M.: Prospective unmanned aerial vehicle operations in the future national airspace system. Collection of Technical Papers - AIAA 4th Aviation Technology, Integration, and Operations Forum, ATIO, vol. 1, pp. 172–179. Scopus, Chicago (2004)
5. Ellerbroek, J., Visser, M., van Dam, S.B.J., Mulder, M., van Paassen, M.M.: Towards an Ecological Design of a Four-Dimensional Separation Assistance Interface. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Chicago, IL FBZ paper (2009)

6. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. International Journal on Robotics Research 17(7), 760–772 (1998)
7. Kluge, B.: Recursive Probabilistic Velocity Obstacles for Reflective Navigation. In: Proc. of 1st Int. Workshop on Advances in Service Robots, Bardolino (2003)
8. Snape, J., van der Berg, J., Guy, S.J., Manocha, D.: The Hybrid Reciprocal Velocity Obstacle. IEEE Transactions on Robotics 27(4), art. no. 5746538, 696–706 (2011)
9. Civil Aviation Authority (n.d), CAP 722 Unmanned Aircraft System Operations in UK Airspace? Guidance, `http://www.caa.co.uk` (retrieved January 10, 2013)
10. Federal Aviation Administration (n.d), Federal Aviation Regulations (FAR) Chapter I, subchapter F Air Traffic and General Operating Rules, 91.113 Right-of-way rules: Except water operations, `http://www.ecfr.gov` (retrieved January 10, 2013)
11. Federal Aviation Administration, FAA Flight Plan? Charting the Path for the Next Generation, Washington, DC (2008-2012)
12. Federal Aviation Administration, (n.d) JO 7110.65U, Air Traffic Control, Chapter 1: General, `http://www.faa.gov` (retrieved January 13, 2013)

Nonlinear Non-cascaded Reference Model Architecture for Flight Control Design

Fubiao Zhang, Florian Holzapfel, and Matthias Heller

Abstract A nonlinear reference model architecture motivated by dynamic inversion based flight control is introduced. As a novel feature, only one integrated reference model is used to provide reference commands, for longitudinal axis: the flight path angle, vertical load factor (or angle of attack), and pitch rate, while admitting flight path rate command as input; for lateral axis, bank angle and roll rate; for directional axis, lateral load factor and yaw rate. Flight dynamics, actuator dynamics with rate and position limits, and envelope protections can also be incorporated in a straight forward way in the reference model. One advantage of this non-cascaded reference model is that at least the attitude of the reference response can be restored and flown at an early stage of the flight control system design cycle. The second feature is that the reference model is parameterized, allowing the opportunity of updating the knowledge of aircraft dynamics (e.g. damaged) and flying qualities design. With these two aspects, the physical consistency in terms of the reference commands among different channels and reference commands reasonable with respect to true aircraft dynamics can be assured. Although designed for General Aviation aircraft, the framework can be generalized for other aircraft considering only rigid body dynamics.

Nomenclature

A	Aerodynamic frame
B	Body-fixed frame
O	NED frame
E	Earth centered earth fixed frame

Fubiao Zhang · Florian Holzapfel · Matthias Heller
Institute of Flight System Dynamics (FSD),
Technische Universität München (TUM),
Boltzmannstraße 15, D-85748 Garching, Germany
e-mail: {fubiao.zhang,florian.holzapfel,matthias.heller}@tum.de

K	Kinematic frame
$\overline{K}$	Rotated kinematic frame
L, M, N	The moments around x, y and z axis of B frame, respectively
p, q, r	Angular rates around x, y and z axis of B frame, respectively
θ, ϕ, φ	Pitch angle, bank angle, azimuth angle, respectively
γ_K, μ_K, χ_K	Flight path angle, flight path bank angle, flight path azimuth angle, respectively
α, β	Angle of attack, sideslip angle, respectively
η, ξ, ζ	Elevator, aileron and rudder deflection, respectively
V_A^G	Aerodynamic speed of the Center of gravity
V_K^G	Kinematic speed of the Center of gravity
$\dot{\rho}_{\overline{K}}$	Modified flight path angle rate, $\dot{\rho}_{\overline{K}} = \dot{\gamma}_K / \cos \mu_K$
$\overline{q}$	Dynamic pressure
c	Command signal
ff	Feed forward signal
r	Reference signal
$extra$	Extra command compared to signal limit
alw	Allowed magnitude of the signal

1 Introduction and Motivation

Fly-by-Wire control systems have been successfully applied in both commercial and military aviation for several decades, despite its high price and complexity, this paper presents part of the efforts to apply this concept in the General Aviation aircraft, in an attempt to make flight safer and more intuitive.

Many general aviation aircrafts are operated by only one pilot, with envelope protection and high automation, fly-by-wire control systems can offer carefree handling, relieving the pilot from tedious stabilizing work. The more pilot is unburdened from a low-level, high frequency routine handling tasks, the more attention he can pay for other higher level tasks, e.g. navigation, communication, making decisions in case of adverse weather condition and so on. It might be safe to say one basic higher level task of manual flight control is to change the flight path, to this point, flight path angle rate command seems reasonable, as it gives pilot direct influence on a state at a higher end of the control loops, while the inner loop (i.e. pitch attitude, pitch rate) control tasks can be automatically executed by the flight control laws, thus the workload could be reduced while equal or better performance could be expected. From a handling qualities point of view, it also makes sense to interpret the pilot's stick (command) input as flight path rate. Because within the pilot's operation bandwidth, it is highly desirable to achieve a

$1/s$ structure from pilot's input to the targeted variable to be controlled, which is generally speaking the flight path.

Pioneer work has been done focusing on flight path augmentation [1-4], encouraging the command system selected in this paper. For instance in Ref [1], a flight path angle and ground track control wheel steering system was proposed to reduce pilot's workload and improve safety during maneuvering. In Ref [2], the author described the 7J7 manual control functions, which was also featured by flight path rate command, and concluded that the functions are favorably received by pilots from major airlines in the simulator tests. As for general aviation aircraft [3], piloted simulation results also showed that direct influence on flight path got positive feedback from the pilots. The authors of Ref [4] compared three different command systems with conventional Fokker 100: a rate command system, a flight path vector command system and a C^* command system, the results are that the fly-by-wire control system itself does not improve performance significantly, however, when equipped with a flight director system, the flight path command system reduced much workload. Recent research conducted in TU Delft investigated the integration of flight path oriented control and perspective flight path displays [5-6].

The paper at hand is limited to the flight path control problems only without dealing with display designs. Here a modified flight path angle rate command $\dot{\rho}_{\overline{K}}$ in the rotated kinematic frame ($\overline{K}$) is proposed, that is for flight in the vertical plane, pilot's stick deflection is seen as flight path angle rate command $\dot{\gamma}_K$ in the kinematic (K) frame, however, when the aircraft banked from the flight path, the stick deflection is interpreted as commanding a change in the flight path rate in the vertical symmetric plane. A comment to be made here is that the actual feedback variable does not have to be the same as command, here γ_K could be used to stabilize the inertial flight path angle.

To implement a $\dot{\rho}_{\overline{K}}$ command system, this paper presents a novel reference model architecture. This architecture was motivated by building only one integrated, physically meaningful, nonlinear reference model to provide reference commands of different channels in the dynamic inversion based flight control. By contrast, many of the popular designs use cascaded reference models. The advantages could be explained later. However, the application of this concept is not restricted to dynamic inversion based control applications, other model following based designs can also utilize its advantages.

2 Functionality Requirements of the Flight Control System

The reference model design is driven by the functionality requirements of the flight control system. Requirements documents were formulated quite detailed before the flight control design started. These functionality requirements shall be met in terms of desired performance or less restrictive adequate performance when the control loops are closed. Since the reference model design itself is forming ideal dynamic behavior of the closed loop system, it is actually open loop specifications, without considering temporarily the influence of the error dynamics introduced by feedback. The error dynamics, on the other hand, would deteriorate the ideal performances when there is disturbance or uncertainty in the plant model. With this in mind, the desired performance must be met by the reference model itself. As far as this paper is concerned, several desired performance requirements are listed.

Table 1 High Level Functionality Requirements

No.	Functionality	Desired Performance
1	Flight path angle rate command	$\delta F_{s,lon} = \dot{\rho}_{\overline{K}}$
2	Flight path angle Limitation	$-25^\circ \leq \gamma_K \leq 25^\circ$
3	Pitch angle limitation	$-50^\circ \leq \theta \leq 50^\circ$
4	Load factor limitation	$-1 \leq (n_z)_B \leq 3.8$
5	AoA protection	$-5^\circ \leq \alpha_A \leq 12^\circ$
6	High(low) speed limitation	$V_s \leq V_A \leq V_{\max}$
7	Turn compensation	$\Delta n_{z,c} = \Delta n_{z,turn\_compensation}$
8	Turn coordination	$\dot{\theta} = 0, \dot{\phi} = 0, (n_y)_B = 0$
9	Velocity vector roll	$r_K / p_K = \tan \alpha_A$
10	Bank angle limitation	$-75^\circ \leq \phi \leq 75^\circ$
11	Spiral stability	Neutral stable: $-48^\circ \leq \phi \leq 48^\circ$ Stable: $48^\circ \prec \lvert\phi\rvert \leq 75^\circ$
12	Trimmed AoA limitation	$-2^\circ \leq \alpha_{trim} \leq 9^\circ$

3 Transition of Functionality Requirements to Design Constrains

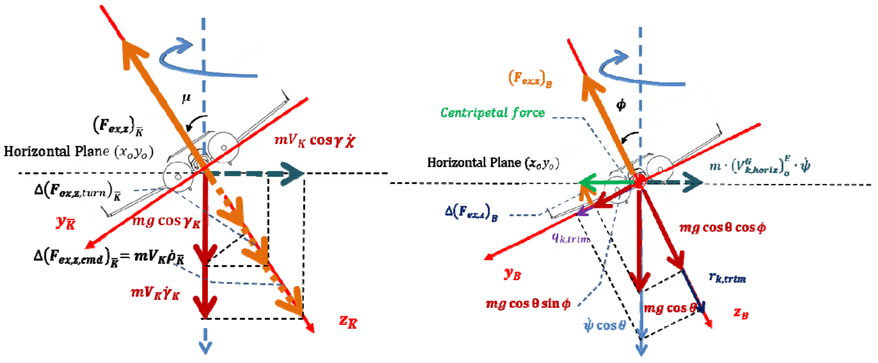
In this section, functionality requirements are interpreted in a lower operational level to guide the flight control design.

3.1 Modified Flight Path Angle Rate Command

For flight path angle rate command realization, here a new variable $\dot{\rho}_{\overline{K}}$, modified flight path rate, is introduced. As mentioned above, $\dot{\rho}_{\overline{K}}$ is equivalent to $\dot{\gamma}_K$ for non-turn maneuver, whereas for turning flight $\dot{\rho}_{\overline{K}}$ corresponds to the rotation rate of velocity vector in the symmetric plane. The force situation is illustrated in Figure 1 (a) in the rotated kinematic frame ($\overline{K}$), when the external force in the z-axis of $\overline{K}$ frame is tilted. Two equations could be concluded to illustrate what $\dot{\rho}_{\overline{K}}$ means:

$$\Delta(n_z)_{\overline{K}} \cdot g = V_K \cdot \dot{\rho}_{\overline{K}} \qquad (1)$$

$$\dot{\rho}_{\overline{K}} = \dot{\gamma}_K / \cos \mu_K \qquad (2)$$


μ
$(F_{ex,z})_{\overline{K}}$
Horizontal Plane $(x_o y_o)$
$mV_K \cos\gamma\,\dot{\chi}$
$\Delta(F_{ex,z,turn})_{\overline{K}}$
$y_{\overline{K}}$
$mg\cos\gamma_K$
$\Delta(F_{ex,z,cmd})_{\overline{K}} = mV_K\dot{\rho}_{\overline{K}}$
$mV_K\dot{\gamma}_K$
$z_{\overline{K}}$
$(F_{ex,z})_B$
ϕ
Centripetal force
Horizontal Plane $(x_o y_o)$
$m \cdot (V^G_{k,horiz})^E_o \cdot \dot{\psi}$
$\Delta(F_{ex,z})_B$
$q_{k,trim}$
$mg\cos\theta\cos\phi$
y_B
$r_{k,trim}$
$mg\cos\theta\sin\phi$
$mg\cos\theta$
$\dot{\psi}\cos\theta$
z_B

(a) (b)

Fig. 1 Turn Analysis in $\overline{K}$ (a) and B (b) Frame

3.2 Turn Compensation

A natural choice for considering turn compensation in the flight path dynamics environment would be in the $\overline{K}$ frame, as shown in figure 1(a). External force (excludes gravity) along the rotated kinematic frame z-axis $(F_{ex,z})_{\overline{K}}$, which mainly comes from lift, contributes in two aspects. First, $(F_{ex,z})_{\overline{K}}$ balances the gravity component on the z-axis, which is $mg\cos\gamma_K\cos\mu_K$. On the other hand, one part of $(F_{ex,z})_{\overline{K}}$ is used for turn compensation, here called $\Delta(F_{ex,z,turn})_{\overline{K}}$. The turn compensation part acts together with the gravity component on $\overline{K}$ frame y-axis, which is $mg\cos\gamma_K\sin\mu_K$, to provide the necessary centripetal force for the turn.

Or put it in another way,

$$\left(F_{ex,z}\right)_{\bar{K}} = \underbrace{mg\cos\gamma_K\cos\mu_K}_{gravity\_balancing} + \underbrace{\Delta\left(F_{ex,z,turn}\right)_{\bar{K}}}_{turn\_compensation} = mg\frac{\cos\gamma_K}{\cos\mu_K} \tag{3}$$

However, μ_K is not directly accessible for measurement and must be calculated from the available sensor signals. In this case, a better analysis might be done in the body fixed frame, as illustrated in figure 1 (b). Then the required external force is

$$\left(F_{ex,z}\right)_B = \underbrace{mg\cos\theta\cos\phi}_{gravity\_balancing} + \underbrace{\Delta\left(F_{ex,z,turn}\right)_B}_{turn\_compensation} = mg\frac{\cos\theta}{\cos\phi} \tag{4}$$

As it is quite desirable that the additionally needed lift could be generated without an intentional command from the pilot, it could be provided automatically by the flight control system, leaving the pilot with less workload. In this paper, such signals are denoted by the subscript 'trim'. Now the necessary load factor for the compensation is given by:

$$\left(n_{z,trim}\right)_B = \frac{\cos\theta}{\cos\phi} \tag{5}$$

3.3 Constraints for Coordinated Turn

Consider first the attitude propagation equation:

$$\dot{\phi} = p_K + q_K \sin\phi\tan\theta + r_K\cos\phi\tan\theta \tag{6}$$

$$\dot{\theta} = q_K\cos\phi - r_K\sin\phi \tag{7}$$

$$\dot{\psi} = q_K\sin\phi/\cos\theta + r_K\cos\phi/\cos\theta \tag{8}$$

The strapdown equation could be obtained by solving for the rotation rates.

$$\begin{bmatrix} \left(p_K\right)_B^{OB} \\ \left(q_K\right)_B^{OB} \\ \left(r_K\right)_B^{OB} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{9}$$

Recall the equation of motion of lateral acceleration in the body fixed frame

$$\left(\dot{v}_K\right)_B^{EB} = -\left(r_K\right)_B^{OB}\cdot\left(u_K\right)_B^E + \left(p_K\right)_B^{OB}\cdot\left(w_K\right)_B^E + g\sin\phi\cos\theta + \left(F_y\right)_B/m \tag{10}$$

In a steady state coordinated turn, the following conditions hold:

$$\left(\dot{v}_K\right)_B^{EB} = 0\,, \left(F_y\right)_B = 0 \tag{11}$$

Together with the constraints

$$\dot{\theta} = 0, \dot{\phi} = 0, \left(n_y\right)_B = 0 \tag{12}$$

The rate change of azimuth angle could be solved:

$$\dot{\psi} = \frac{g \sin\phi \cos\theta}{\cos\phi \cos\theta \cdot \left(u_K\right)_B^E + \sin\theta \cdot \left(w_K\right)_B^E} \tag{13}$$

In the case $\left(u_K\right)_B^E$ and $\left(w_K\right)_B^E$ from navigation system are not available. An approximation has to be used,

$$\dot{\psi} = g \tan\phi / V_A \tag{14}$$

Until now, to execute the coordinate turn, the following body rotation rates must be generated by the flight control system.

$$\left(p_K\right)_{B,trim}^{OB} = -\sin\theta \cdot \tan\phi \cdot g / V_A \tag{15}$$

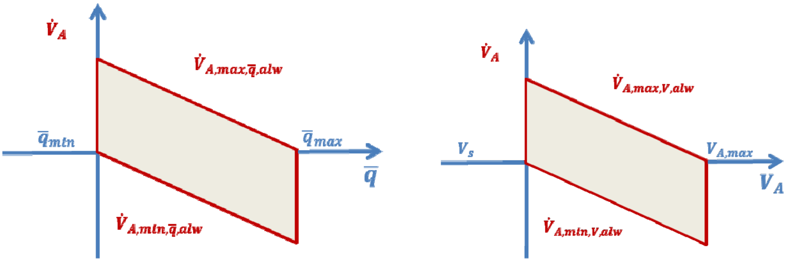
$$\left(q_K\right)_{B,trim}^{OB} = \sin\phi \cos\theta \cdot \tan\phi \cdot g / V_A \tag{16}$$

$$\left(r_K\right)_{B,trim}^{OB} = \sin\phi \cos\theta \cdot g / V_A \tag{17}$$

3.4 Precondition for Envelope Protections and Phase Plane Based Protection Concept

One precondition for envelope protections is the assurance of the controllability over the aircraft. This is especially true when the dynamic pressure is already quite low. Thus the decrease of the airspeed must not be so fast that no time is left before the aircraft can be recovered from a dangerous situation when the aircraft is actually close to the envelope boundary. Similarly, the increase of the airspeed shall also be limited to avoid overstressing the aircraft.

Here only the high speed limitation requirement is explained, but it illustrates the basis idea of phase plane based protection and merging of other sources of limiting factors. Realization of other requirements are done along with the introduction of the reference model



(a) (b)

Fig. 2 Allowable $\dot{V}_A$ in the Grey Area

To prevent the dynamic pressure from going beyond the safe range, the allowed $\dot{V}_A$ is proportional to its distance to the boundary values, as shown in figure 2 (a).

$$\dot{V}_{A,\max,\bar{q},alw} = K_{V_{\max},\bar{q},prot}(\bar{q}_{\max} - \bar{q}) \tag{18}$$

$$\dot{V}_{A,\min,\bar{q},alw} = K_{V_{\min},\bar{q},prot}(\bar{q} - \bar{q}_{\min}) \tag{19}$$

Similarly, $\dot{V}_A$ shall also be limited when the airspeed is close to the boundaries, as shown in figure 2 (b):

$$\dot{V}_{A,\max,V,alw} = K_{V_{\max},V,prot}(V_{\max} - V_A) \tag{20}$$

$$\dot{V}_{A,\min,V,alw} = K_{V_{\min},V,prot}(V_A - V_s) \tag{21}$$

When consider more than one factors, conservative limits should be taken,

$$\dot{V}_{A,\max,alw} = \min\left\{\dot{V}_{A,\max,\bar{q},alw}, \dot{V}_{A,\max,V,alw}\right\} \tag{22}$$

$$\dot{V}_{A,\min,alw} = \max\left\{\dot{V}_{A,\min,\bar{q},alw}, \dot{V}_{A,\min,V,alw}\right\} \tag{23}$$

4 Nonlinear Reference Model Architecture

The reference model is nonlinear in a way that reference commands are formulated by the summation of two parts: the trim term and the maneuver part. The maneuver part is the output of the desired linear dynamics. The trim term remains when pilot's stick centered to realize a certain function like turn compensation. Another three common features could be seen later in the reference model: 1) all the output signals are limited by its own upper and lower bound based on physical knowledge. 2) all the inputs of the integrators are limited for the sake of anti-windup. 3) all the reference signals are forward propagated based on the flight dynamics, whereas the saturation (when a certain signal hits the envelope boundary) and boundary overshoot information are backward propagated to the previous integrators to avoid overloading, which is also based on flight dynamics.

4.1 Longitudinal Reference Model

The overall architecture for longitudinal reference model is shown in figure 3.

With

$K_{s,lon}$ Stick scaling gain

$K_{c,lon}$ Dynamic Gain

$M_{actuator}$ Actuator dynamic model with rate and position limits

$M_{aircraft,sp}$ Desired aircraft short period dynamics

M_{load_factor} Algebraic Model of load factor output

M_{path} Dynamic model to generate flight path reference command

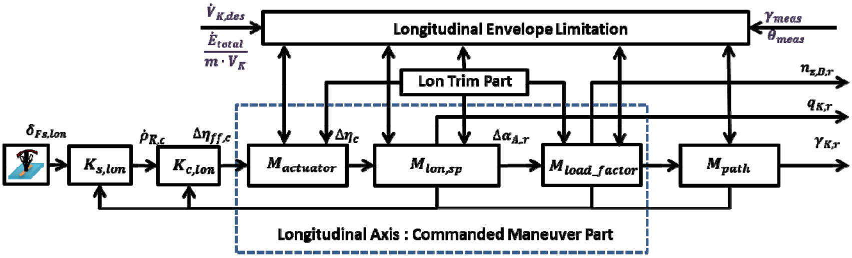


Fig. 3 Architecture of the Longitudinal Reference Model with Limitation

4.1.1 Stick Scaling and Dynamic Gain

Figure 4 illustrates how the signal propagates in the feed forward path. The stick scaling gain makes sure the full deflection always commands the largest possible load factor despite changes of flight condition, which is in low speed mainly limited by the aerodynamic capability and high speed the structural stiffness.

With

$$K_{c,lon} = -\frac{V_K^G}{g} \cdot \frac{g}{V_A^G \cdot Z_\alpha} \cdot \frac{Z_\alpha M_{q,des} - M_{\alpha,des}}{M_\eta} \tag{24}$$

The scaled stick command is converted to the commanded input of the elevator model.

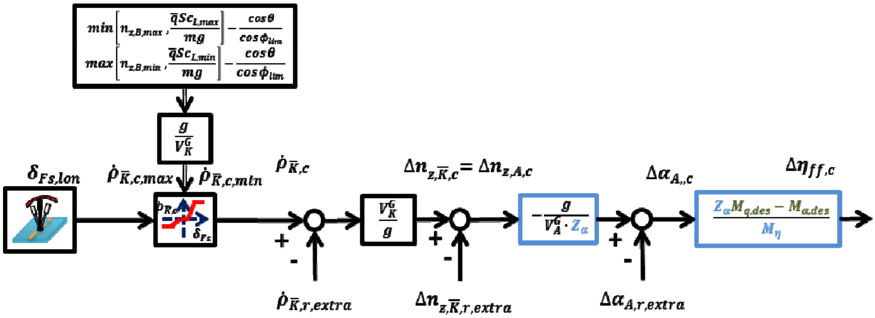

min
max
$\delta_{Fs,lon}$
$\dot{\rho}_{\bar{K},c,max}$
$\dot{\rho}_{\bar{K},c,min}$
$\dot{\rho}_{\bar{K},c}$
$\Delta n_{z,\bar{K},c} = \Delta n_{z,A,c}$
$\Delta\alpha_{A,,c}$
$\Delta\eta_{ff,c}$
$\dot{\rho}_{\bar{K},r,extra}$
$\Delta n_{z,\bar{K},r,extra}$
$\Delta\alpha_{A,r,extra}$
$\frac{g}{V_K^G}$
$\frac{V_K^G}{g}$
$-\frac{g}{V_A^G \cdot Z_\alpha}$
$\frac{Z_\alpha M_{q,des} - M_{\alpha,des}}{M_\eta}$
+
-

Fig. 4 Stick Scaling and Dynamic Gain $K_{c,lon}$

4.1.2 Actuator Model with Limits

In figure 5, a second order actuator model with acceleration limits, rate limits and position limits is shown here. Therefore the output of the reference model would not be too aggressive to be executed. Besides the limits mentioned above, the back propagated saturation and overshoot information is also taken account on the actuator level.

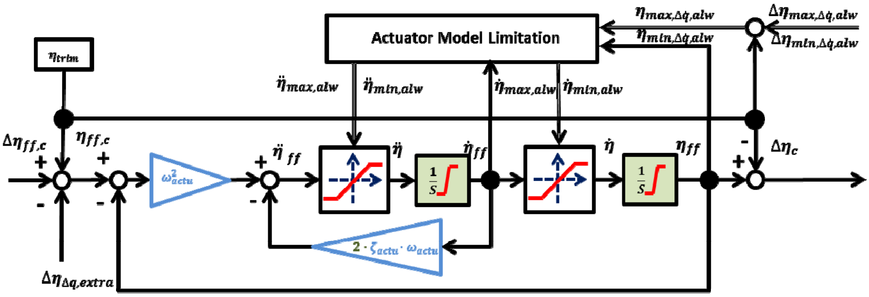


Fig. 5 Actuator Model with Limits

4.1.3 Desired Short Period Dynamics

The linearized short period dynamics is used here, the subscript "des" indicates they could be specified to reach the desired dynamics. $M_{\alpha,des}$ and $M_{q,des}$ can be designed according to flying qualities requirements (mainly from military standards [12-14]).

$$\begin{bmatrix} \Delta\dot{\alpha}_A \\ \Delta\dot{q}_K \end{bmatrix} = \begin{bmatrix} Z_\alpha & 1 \\ M_{\alpha,des} & M_{q,des} \end{bmatrix} \begin{bmatrix} \Delta\alpha_A \\ \Delta q_K \end{bmatrix} + \begin{bmatrix} Z_\eta \\ M_\eta \end{bmatrix} \Delta\eta \tag{25}$$

This linear dynamics seems to make sense because when the pilot commands a maneuver from a certain trimmed flight, the response of the aircraft is quite predictable. In figure 6, the overshoot information, that excessive command of pitch rate $\Delta p_{K,r,extra}$ is propagated back with a gain $M_{\alpha,des}/M_{\eta}$, which would convert $\Delta p_{K,r,extra}$ to $\Delta \eta_{\Delta q,extra}$, and subtracted from the actuator command shown in figure 5. In this way, output signals are not "hard" limited.

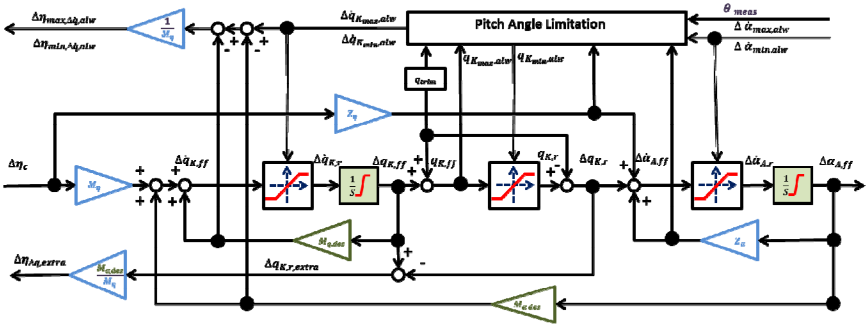


Fig. 6 Desired Short Period Dynamics

4.1.4 Model of Load Factor and Flight Path Angle Reference Command

The maneuver part of the vertical load factor is converted from the AoA with

$$\Delta n_{z,\overline{K},ff} = -\frac{V_K^G \cdot Z_\alpha}{g} \cdot \Delta\alpha_{A,r} \tag{26}$$

Further, after limitation is done, inertial flight path rate could be calculated as:

$$\dot{\gamma}_{K,ff} = -\frac{g}{V_K^G} \cdot \cos\phi_{\lim} \cdot \Delta n_{z,\overline{K},r} \tag{27}$$

To be accurate, flight path bank angle μ_K instead of Euler angle $\phi_{\lim}$ should be used. Unfortunately, this is not practical due to measurement problem. However the consequence of this replacement must be evaluated. Figure 7 also shows the back propagation of overshoot information to the dynamic gain part in figure 4.

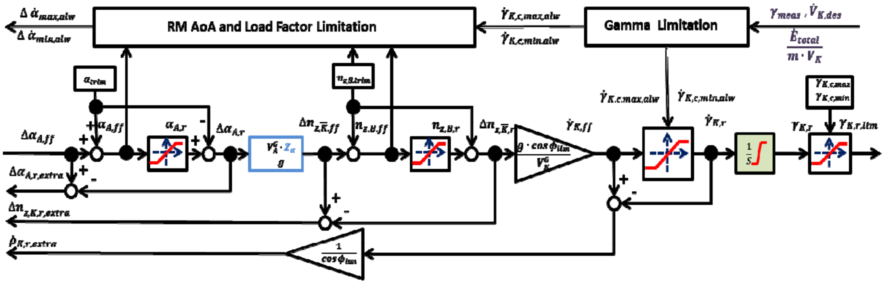

$\Delta\dot{\alpha}_{max,alw}$
$\Delta\dot{\alpha}_{min,alw}$
RM AoA and Load Factor Limitation
$\dot{\gamma}_{K,c,max,alw}$
$\dot{\gamma}_{K,c,min,alw}$
Gamma Limitation
$\gamma_{meas}, \dot{V}_{K,des}$
$\frac{\dot{E}_{total}}{m \cdot V_K}$
α_{trim}
$n_{z,B,trim}$
$\gamma_{K,c,max}$
$\gamma_{K,c,min}$
$\Delta\alpha_{A,ff}$
$\alpha_{A,ff}$
$\alpha_{A,r}$
$\Delta\alpha_{A,r}$
$\frac{V_A^G \cdot Z_\alpha}{g}$
$\Delta n_{z,\overline{K},ff}$
$n_{z,B,ff}$
$n_{z,B,r}$
$\Delta n_{z,\overline{K},r}$
$\frac{g \cdot \cos\phi_{lim}}{V_K^G}$
$\dot{\gamma}_{K,ff}$
$\dot{\gamma}_{K,r}$
$\gamma_{K,r}$
$\gamma_{K,r,lim}$
$\frac{1}{s}$
$\Delta\alpha_{A,r,extra}$
$\Delta n_{z,K,r,extra}$
$\dot{p}_{K,r,extra}$
$\frac{1}{\cos\phi_{lim}}$
+
-

Fig. 7 Model of Load Factor and Flight Path Angle

4.2 Longitudinal Envelope Protections

Here following the speed protection case, the phase plane protection concept is used to form the protection signals as input for the dynamic limitation block. To save space, only the upper bound of a certain protected signal is shown here, as the lower bound is a symmetric mirror of this structure. Most of the protection implementations are well illustrated by the figures nearby.

4.2.1 Flight Path Angle Limitation

Besides the absolute limit of the flight path angle, another limit from energy consideration is also introduced:

As is well known, the total energy of the aircraft could be described as

$$E_{total} = 1/2 \cdot mV_K^2 + mgh \tag{28}$$

The energy change rate is

$$\dot{E}_{total} = mV_K\dot{V}_K + mg\dot{h} = mV_K\dot{V}_K + mgV_K \cdot \sin\gamma_K \tag{29}$$

If the throttle is fixed, indicating no external energy is flowing into the system and the allowed speed change rate $\dot{V}_{K,alw}$ is specified, then the achievable flight path angle limited by this energy condition is:

$$\gamma_K = \arcsin((\dot{E}_{total}/mV_K - \dot{V}_{K,alw})/g) \tag{30}$$

Both the absolute and energy limits are considered as shown in figure 8.

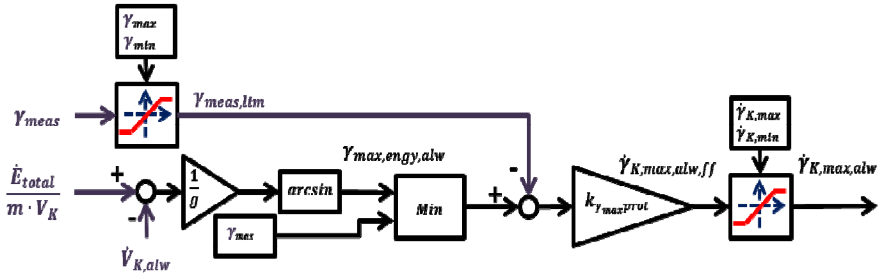


Fig. 8 Flight Path Angle Limitation

4.2.2 Reference Model AoA and Load Factor Limitation

The limitation of the both AoA and load factor can be done at the same time as both of them can be limited by limiting the rate change of AoA, which physically is the source of change in the AoA or load factor.

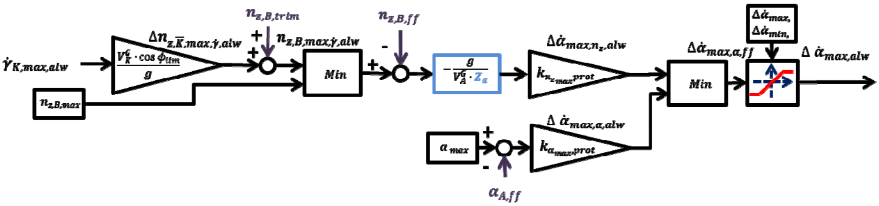

$\dot{\gamma}_{K,max,alw}$
$\frac{V_K^G \cdot \cos\phi_{lim}}{g}$
$n_{z,B,max}$
$\Delta n_{z,\bar{K},max,\dot{\gamma},alw}$
$n_{z,B,trim}$
+
+
$n_{z,B,max,\dot{\gamma},alw}$
Min
$n_{z,B,ff}$
+
-
$-\frac{g}{V_A^G \cdot Z_\alpha}$
$\Delta\dot{\alpha}_{max,n_z,alw}$
$k_{n_{z_{max}}prot}$
α_{max}
+
-
$\alpha_{A,ff}$
$\Delta\dot{\alpha}_{max,\alpha,alw}$
$k_{\alpha_{max}prot}$
$\Delta\dot{\alpha}_{max,\alpha,ff}$
Min
$\Delta\dot{\alpha}_{max},$
$\Delta\dot{\alpha}_{min},$
$\Delta\dot{\alpha}_{max,alw}$

Fig. 9 Reference Model AoA and Load Factor Limitation

The saturation information from the flight path angle model is represented by the maximum allowed flight path angle rate $\dot{\gamma}_{K,\max,alw}$. In contrast to the command loop, in the protection loop, $\dot{\gamma}_{K,\max,alw}$ is converted back to the maximum allowable incremental of load factor $\Delta n_{z,\bar{K},\max,\dot{\gamma},alw}$. In the next step after adding the current trim part of the load factor, the maximum allowed load factor limited by flight path angle $n_{z,B,\max,\dot{\gamma},alw}$ could be obtained. Then the smaller one when compared to the absolute load factor upper bound due to structure stiffness, would be selected as the final upper bound of the allowable load factor.

Note that the limitation is only done within the reference model, that is with this structure, closed loop response of AoA, for instance, would not be expected to be still in the safe range, as the error dynamics induced by the controller would make the transient close loop response be different from the reference command to be followed. Nevertheless, this limitation in the reference model is still necessary, as the tracking error of a successfully designed controller will converge to zero, then the closed loop AoA would follow the reference command in the steady state.

4.2.3 Pitch Angle Limitation

For pitch angle limitation, the pitch angle differential equation (7) is used to bound pitch rate,

$$q_{K_{\max},\theta,alw} = \left(\dot{\theta}_{\max,alw} + r_K \cdot \sin\phi\right) / \cos\phi \tag{31}$$

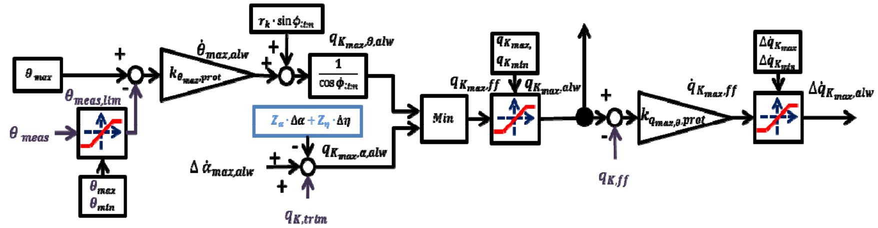


Fig. 10 Pitch Angle Limitation

Another factor from AoA and load factor limitation is considered dynamically by solving for the allowed pitch rate when the allowed rate change of AoA is limited. As shown in figure 10.

$$\Delta q_{K_{\max},\alpha,alw} = \Delta\dot{\alpha}_{\max,alw} - Z_{\alpha}\cdot\Delta\alpha - Z_{\eta}\cdot\Delta\eta \tag{32}$$

4.2.4 Actuator Limitation

There is one point to be mentioned for actuator limitation, the information from other protections are carried by $\Delta\dot{q}_{K_{\max},alw}$ and used here through the equation:

$$\Delta\eta_{\max,\Delta\dot{q},alw} = \left(\Delta\dot{q}_{K_{\max},alw} - M_{\alpha,des}\cdot\Delta\alpha - M_{q,des}\cdot\Delta q\right)/M_{\eta} \tag{33}$$

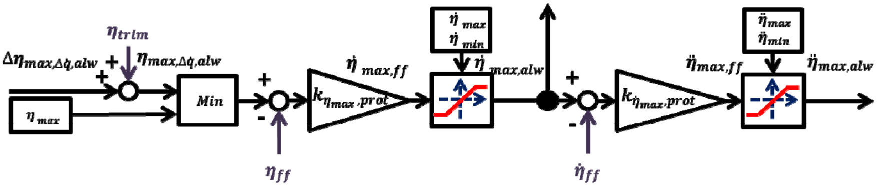


Fig. 11 Actuator Model with Limits

4.3 Lateral and Directional Reference Model Architecture

The general architecture for lateral and directional reference model is shown in figure 12

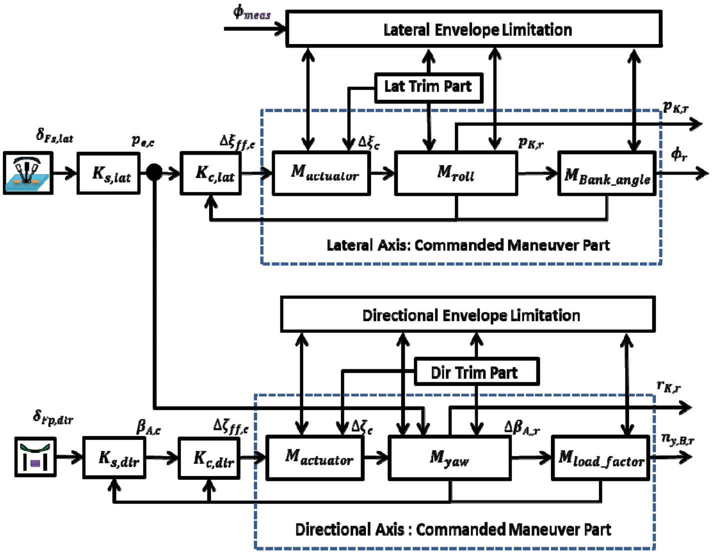


Fig. 12 Architecture of the Lateral and Directional Reference Model

4.3.1 Lateral Reference Model

For lateral reference model, roll channel dynamics featuring a first order behavior is implemented here.

$$\Delta\dot{p}_K = L_{p,des}\cdot\Delta p_K + L_\xi\Delta\xi \tag{34}$$

As lateral stick deflection is commanding velocity vector roll, the body fixed frame roll rate command is simply:

$$\Delta p_{K,c} = p_{e,c}\cdot\cos\alpha_A \tag{35}$$

Spiral stability can be restored for bank angle $48^\circ \prec |\phi| \le 75^\circ$ by actively adding proper roll rate to obtain the roll reference command as shown in figure 13. However, by this way, even with full lateral stick deflection, the maximum achieved bank angle can not reach exactly 75° . Moreover, to prevent stall due to turn compensation in large bank angles, trimmed AoA is also seen as a factor to limit the bank angle, as

$$\alpha_{trim} = C_L^{-1}\cdot\cos\theta\cdot mg/(\cos\phi\cdot\bar{q}\cdot S) \tag{36}$$

The lateral reference model structure is shown in figure 13.

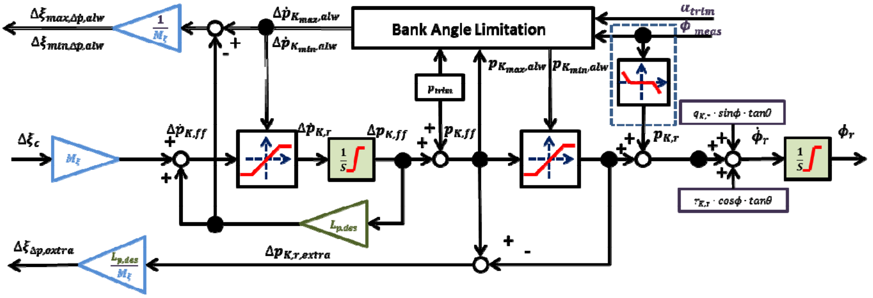


Fig. 13 Lateral Reference Model

4.3.2 Directional Reference Model and Envelope Protection

The directional reference model is specified as a second order system.

$$\begin{bmatrix}\Delta\dot{r}_K\\ \Delta\dot{\beta}_A\end{bmatrix} = \begin{bmatrix}N_{r,des} & N_{\beta,des}\\ -1 & Y_\beta\end{bmatrix}\begin{bmatrix}\Delta r_K\\ \Delta\beta_A\end{bmatrix} + \begin{bmatrix}N_\varsigma\\ Y_\varsigma\end{bmatrix}\Delta\varsigma \tag{37}$$

One special point is the fact that yaw rate also contributes to the evolution of pitch angle, and shall be limited to complete the pitch angle limitation. From the pitch angle differential equation (7)

$$r_{K_{\max},\theta,alw} = \left(q_K \cdot \cos\phi - \dot{\theta}_{\max,alw}\right) / \sin\phi \tag{38}$$

Since this equation would become singular at zero bank angle. Another strategy is used here, that is the yaw rate command $\Delta r_{K,ff}$, shall be limited as a function of current pitch angle.
With

$$\Delta r_{K,ff} = p_{e,c} \cdot \sin\alpha_A + \Delta r_{K,r} \tag{39}$$

The allowable $\Delta r_{K,ff}$ is implemented as shown in figure 14 with the logic in the table 2.

Table 2 Allowable $\Delta r_{K,ff}$ When Pitch Angle Hits the Envelope Boundary

$\Delta r_{K,r}$	$\theta = 50^\circ$	$\theta = -50^\circ$
$\phi \geq 0$	$\left[0^\circ \sim 100^\circ \cdot K_{\theta_{\min},prot}\right]$	$\left[-100^\circ \cdot K_{\theta_{\max},prot} \sim 0^\circ\right]$
$\phi \prec 0$	$\left[-100^\circ \cdot K_{\theta_{\min},prot} \sim 0^\circ\right]$	$\left[0^\circ \sim 100^\circ \cdot K_{\theta_{\max},prot}\right]$

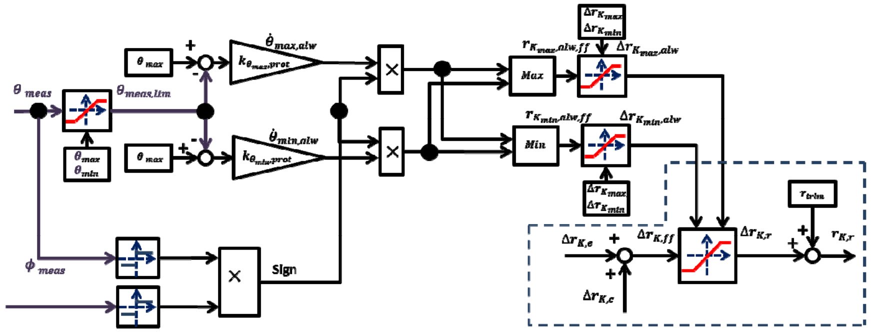


Fig. 14 Logic to Limit Yaw Rate for Pitch Angle Limitation

5 Simulation Results

The feasibility of this reference model is illustrated by trimming and linearizing a general aviation aircraft (Diamond DA42) model in straight and level flight at the speed of 93 knots and height of 1500 m, the attitude response of the reference model is restored using equation (6)-(8), with reference rotation rates.

Simulation results are shown in figure 15, the dashed or dotted lines are limits for the signals. For the stick deflections during the period $t = 2 \sim 4$ seconds, no limits are violated.

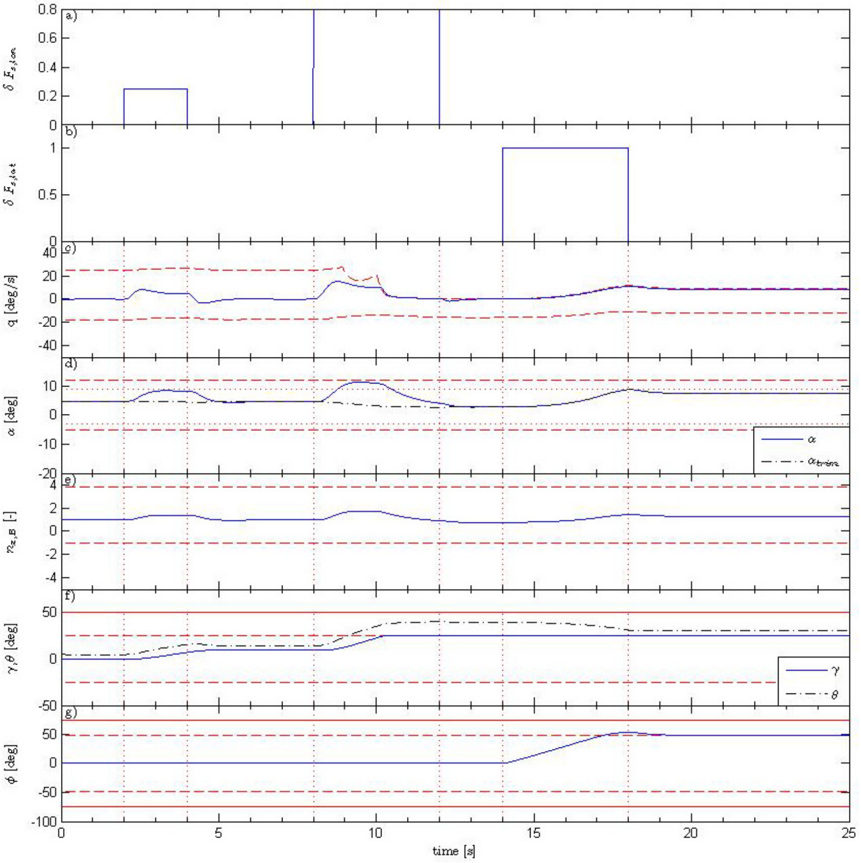


Fig. 15 Response of the Reference Model to the Stick Commands

A flight path rate command/flight path angle hold response could be seen in figure 15 (f), pitch rate (c) and AoA (d) response are typical aircraft response to a step command. Then an even larger pull is given by the stick from $t = 8 \sim 12$ seconds, the flight path angle soon reaches its limit at about $t = 10.2$ seconds, the limitation is evident to be effective as the nose of the reference aircraft starts to pitch down (figure 15 (c)), the reference pitch rate hits its upper bound and remains constrained by its upper bound as long as the flight path angle is at its upper bound, and the commanded AoA is automatically being reduced although the stick still pulls for more flight path angle till $t = 12$, the trimmed AoA (figure 15 (d), dash-dot line) is getting smaller as not so much load factor is needed for a higher pitch angle (recall equation (5)), this is a physically meaningful consequence of the aircraft response to envelope protection.

During the period $t = 14 \sim 18$, and full lateral stick deflection is commanded, as time elapses, the bank angle starts to build, then turn compensation could be seen working as shown in figure 15 (d) and (e), more lift is being generated as bank angle is increasing, to leave 3° space for pilot's active maneuver command, the trimmed AoA is limited to $-2^{\circ} \le \alpha_{trim} \le 9^{\circ}$, further deflection of lateral stick would be neglected if α_{trim} reaches its upper bound. A velocity vector roll could be partly seen as pitch angle started to go down to help for the rolling around velocity vector during this period. The spiral stability is restored as the bank angle returns to 48° degree at the release of the stick. Another functionality, turn coordination could also be seen after $t = 18$, because the reference aircraft dynamics is holding both pitch angle and bank angle unchanged.

During all the simulation time, the AoA, load factor and pitch angle are well within the envelopes defined by the functionality requirements in table 1.

6 Conclusion

The usefulness of the proposed reference model architecture is demonstrated by numerical simulation, as the results shown it can fulfill the functionality requirements. A more intuitive way to evaluate the reference model is to fly manually with joystick and visualize with software like Flight Gear, this part is omitted here.

To conclude, this reference model architecture has two benefits. Firstly, the integrated reference model takes into account flight dynamics, actuator dynamics and limits and envelope protection aspects. The designer can understand immediately the consequence of modifying the reference model, as it seems to be a copy of the aircraft and all the signals have their physical meaning, previous knowledge of flight dynamics applies to the reference model. Secondly, the non-cascaded structure decoupled the specific controller design with the desired reference dynamics, whereas the reference dynamics could be seen as the ideal dynamics formed by requirements. Hence at an early age of flight control system design, the ideal dynamics including envelope protections could already be evaluated by piloted simulation, and thus reduces the iterations in the design cycle.

More aspects concerning this nonlinear reference model could be explored in the future such as design the parameters in a way to provide good flying qualities. Another interesting aspect could be increasing pilot's awareness of protection situation by audio warning or stick force changes.

Acknowledgement. Special thanks must be given to Dipl.-Ing. Falko Shuck for providing the model of the general aviation aircraft. The authors also thank the reviewers for the helping comments.

References

1. Lambregts, A.A., Cannon, D.G.: Development of a Control Wheel Steering Mode and Suitable Displays that Reduce Pilot Workload and Improve Efficiency and Safety of Operation in the Terminal Area and in Windshear. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, pp. 609–620. AIAA, New York (1979)
2. Sankrithi, M.M.K.V., Bryant, W.F.: 7J7 Manual Flight Control Functions, AIAA paper 87-2454 (1987)
3. Stewart, E.C., Ragsdale, W.A., Wunschel, A.J.: An Evaluation of Automatic Control System Concepts for General Aviation Airplanes. In: AIAA Atmospheric Flight Mechanics Conference, United States, Minneapolis, August 15-17, pp. 330–343 (1988)
4. Van der Geest, P., Nieuwpoort, A., Borger, J.: A Simulator Evaluation of Various Manual Control Concepts for Fly-by-Wire Transport Aircraft. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, pp. 181–199. AIAA, Washington, DC (1992)
5. Mulder, M., Veldhuijzen, A.R., van Paassen, M.M., Mulder, J.A.: Integrating Fly-by-Wire Controls with Perspective Flight-Path Displays. Journal of Guidance, Control and Dynamics 28(6) (November-December 2005)
6. Borst, C., Mulder, M., van Paassen, M.M., Mulder, J.A.: Path-Oriented Control/Display Augmentation for Perspective Flight-Path Displays. Journal of Guidance, Control, and Dynamics 29(4) (July-August 2006)
7. Holzapfel, F., Schuck, F., Höcht, L., Sachs, G.: Flight Dynamics Aspects of Path Control. In: AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina, August 20-23 (2007)
8. Holzapfel, F., Sachs, G.: Dynamic Inversion Based Control Concept with Application to an Unmanned Aerial Vehicle. In: AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA-2004-4907 (2004)
9. Johnson, E.: Limited Authority Adaptive Flight Control. PhD thesis, Georgia Institute of Technology (2000)
10. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation. John Wiley and Sons Inc. (2003)
11. Tischler, M.B.: Advances in Aircraft Flight Control. Taylor & Francis (1996)
12. Gibson, J.: The Definition, Understanding and Design of Aircraft Handling Qualities. Delft University Press (1997)
13. Gibson, J.: Development of a design methodology for handling qualities excellence in fly by wire aircraft (1999)
14. Anon. Flying Qualities of Piloted Aircraft MIL-HDBK-1797. US Department of Defense (1997)
15. Flight Control Design-Best Practices. AGARD Report, RTO-TR-029 (December 2000)
16. Wünnenberg, H.: Handling Qualities of Unstable Highly Augmented Aircraft. AGARD Advisory Report. 29 (1991)
17. Diamond Aircraft. Airplane Flight Manual DA-42 NG (2009)

Aircraft Longitudinal Guidance Based on a Spatial Reference

Hakim Bouadi, Daniel Choukroun, and Felix Mora-Camino

Abstract. In this study, instead of using time as the independent variable to describe the guidance dynamics of an aircraft, distance to land, which can be considered today to be available online with acceptable accuracy and availability, is adopted. A new representation of aircraft longitudinal guidance dynamics is developed according to this spatial variable. Then a nonlinear inverse control law based-on this new representation of guidance dynamics is established to make the aircraft follow accurately a vertical profile and a desired airspeed. The desired airspeed can be regulated to make the aircraft overfly different waypoints according to a planned time-table. Simulations results with different wind conditions for a transportation aircraft performing a descent approach for landing under this new guidance scheme are displayed.

1 Introduction

World air transportation traffic has known a sustained increase over the last decades leading to airspace near saturation in large areas of developed and emerging countries. For example, up to 27,000 flights cross European airspace every day while the number of passengers is expected to double by 2020. The available infrastructure of current ATM (Air Traffic Management) will no longer be able to stand this growing demand unless breakthrough improvements are made. In the air traffic management environment defined by SESAR and NextGen projects, two main objectives are targeted, strategic data link services for

Hakim Bouadi · Felix Mora-Camino
MAIAA, Air Transport Dept., ENAC, Toulouse, France, 31400
e-mail: {hakim.bouadi,felix.mora}@enac.fr

Daniel Choukroun
Space Systems Engineering, Faculty of Aerospace Engineering,
Delft University of Technology, Delft, The Netherlands, 2629 HS
e-mail: D.Choukroun@tudelft.nl

sharing of information and negotiation of planning constraints between ATC (Air Traffic Control) and the aircraft in order to ensure planning consistency and the use of the 4D aircraft trajectory information in the flight management system for ATC operations [1], [2] and [3]. This means that in addition to following the trajectory cleared by ATC, aircraft will progress in four dimensions, sharing accurate airborne predictions with the ground systems, and being able to meet time constraints at specific waypoints with high precision when the traffic density requires it [4], [5], [6] and [7]. This will allow better separation and sequencing of traffic flows while green climb/descent trajectories will be feasible in terminal areas.

Current civil aviation guidance systems operate with real time corrective actions to maintain the aircraft trajectory as close as possible to the planned trajectory or to follow timely ATC tactical demands based either on spatial or temporal considerations [8] and [9]. While wind remains one of the main causes of guidance errors [10], [11] and [12], these new solicitations by ATC are attended with relative efficiency by current airborne guidance systems. However, these guidance errors are detected for correction by navigation systems whose accuracy has known large improvements in the last decade with the hybridization of inertial units with satellite information. Nevertheless, until today vertical guidance remains problematic, [13] and [14], and corresponding covariance errors [15] are still large considering the time-based control laws which are applied by flight guidance systems [16].

In this paper, we consider the problem of designing a new longitudinal guidance control laws for an auto guidance system so that more accurate vertical tracking and overfly time are insured. There, instead of using time as the independent variable to describe the guidance dynamics of the aircraft, we adopt distance to land, which can be considered today to be available online with acceptable accuracy and availability. A new representation of aircraft vertical guidance dynamics is developed according to this spatial variable. Then a nonlinear inverse control law based-on this new proposed spatial representation of guidance dynamics is established to make the aircraft follow accurately a vertical profile and a desired airspeed [17] and [18]. The desired airspeed is then regulated to meet two main constraints related to the stall speed and the maximum operating speed and to make the aircraft overfly different waypoints according to a planned time-table.

Simulation experiments with different wind conditions are performed for a transportation aircraft performing a general descent approach for landing. It appears that with this new guidance scheme, vertical 2D+Time guidance can be achieved with accuracy.

2 Horizontal Distance to Land as an Independent Variable

The motion of an approach/descent transportation aircraft along a landing trajectory will be referenced with respect to a RRF (Runway Reference Frame) whose origin is located at the runway entrance as shown in Figure 1.

The vertical plane components of the inertial speed are such as:

$$\dot{x} = -V_{air} \cos \gamma_{air} + w_x \tag{1}$$

$$\dot{z} = V_{air} \sin \gamma_{air} + w_z \tag{2}$$

Then we can write:

$$V_{air} = \sqrt{(\dot{x} - w_x)^2 + (\dot{z} - w_z)^2} \tag{3}$$

$$\gamma_{air} = -arctg\left(\frac{\dot{z} - w_z}{\dot{x} - w_x}\right) \tag{4}$$

where x and z are the vertical plane coordinates of the aircraft centre of gravity in the RRF, V_{air} is the airspeed modulus, γ_{air} is the airspeed path angle, w_x and w_z are the wind components in the RRF.

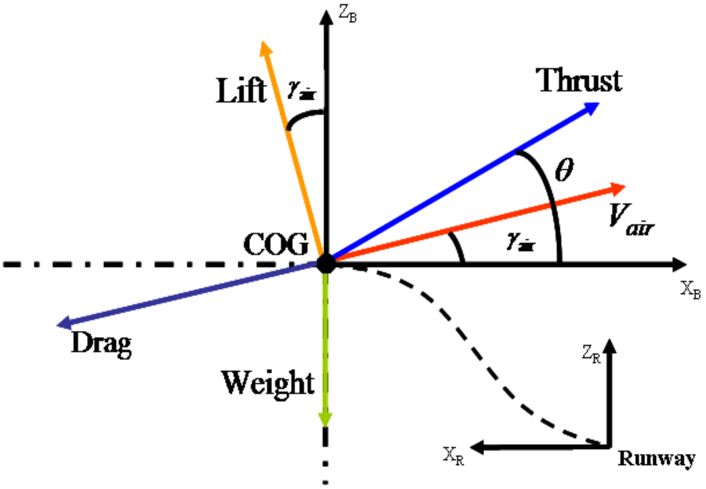


Fig. 1 Runway reference frame with aircraft forces and angles

Supposing that during an approach/descent manoeuvre the distance-to-land time function $x(t)$ is invertible it is possible to express all the flight variables with respect to x and its derivatives instead of using time. Then for any time variable v, we have:

$$dv/dx = (dv/dt)(dt/dx) = \dot{v}/V_G \tag{5}$$

where the ground speed V_G at position x is given here by:

$$V_G = \dot{x} = -V_{air} \cos\gamma_{air} + w_x \tag{6}$$

Then the following notation is adopted for successive derivatives with respect to x:

$$\frac{d^k *}{dx^k} = *^{[k]} \tag{7}$$

3 Space Referenced Longitudinal Flight Dynamics

The aircraft longitudinal guidance dynamics can then be rewritten as:

$$z^{[1]} = \frac{dz}{dx} = \frac{dz}{dt}\frac{dt}{dx} = \frac{V_{air}\sin\gamma_{air} + w_z}{V_G} \tag{8}$$

$$\theta^{[1]} = q/V_G \tag{9}$$

$$V_{air}^{[1]} = \frac{1}{mV_G}\left[T\cos\alpha - D(z, V_{air}, \alpha) - mg\sin\gamma_{air} + m(\dot{w}_x\cos\gamma_{air} - \dot{w}_z\sin\gamma_{air})\right] \tag{10}$$

$$\gamma_{air}^{[1]} = \frac{1}{mV_{air}V_G}\left[T\sin\alpha + L(z, V_{air}, \alpha) - mg\cos\gamma_{air} - m(\dot{w}_x\sin\gamma_{air} + \dot{w}_z\cos\gamma_{air})\right] \tag{11}$$

where T, D and L are respectively the thrust, drag and lift forces. The lift and drag forces are classically given by:

$$L = \frac{1}{2}\rho(z)V_{air}^2 S\, C_L(\alpha) \tag{12}$$

$$D = \frac{1}{2}\rho(z)V_{air}^2 S\, C_D(\alpha) \tag{13}$$

where $\rho(z), S$, C_L and C_D represent the air density with respect to the altitude, the wing surface area, the lift and drag coefficients, respectively and where α denotes the angle of attack with here :

$$\alpha = \theta - \gamma_{air} \tag{14}$$

Assuming first order dynamics with time constant τ for the engines, we get between commanded thrust T_C and effective thrust T the following relation:

$$T^{[1]} = \frac{T_C - T}{\tau V_G} \tag{15}$$

then, with respect to $z^{[2]}$ we get:

$$z^{[2]} = \frac{1}{V_G^2}\left[\left(V_{air}^{[1]} \sin\gamma_{air} + V_{air}\gamma_{air}^{[1]} \cos\gamma_{air} + w_z^{[1]}\right)V_G - \left(V_{air} \sin\gamma_{air} + w_z\right)V_G^{[1]}\right] \tag{16}$$

Here q and T_C can be taken as independent control inputs to the above guidance dynamics while w_x and w_z are perturbation inputs. Equivalent control q is the result of pitch control on a very short time scale performed by the autopilot:

$$\dot{q} = M / I_y \tag{17}$$

where I_y denote the inertia moment and M is the pitch moment which can be approximated by affine expressions such as:

$$M = \frac{1}{2}\rho V_{air}^2 S\bar{c}\left(C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{q\bar{c}}{2V_{air}} + C_{m_{\delta_e}}\delta_e\right) \tag{18}$$

with $\bar{c}$ and δ_e represent the mean chord line and the elevator deflection, respectively.

4 Longitudinal Trajectory Tracking Objectives

Here the considered guidance objectives consist for the aircraft first in following accurately a space-referenced vertical profile $z_d(x)$ defined in accordance with economic and environmental constraints, and second in matching a desired time table $t_d(x)$ during the approach manoeuver according to air traffic control directives. Of course, speed constraints must be satisfied during the manoeuver.

Trying to meet directly the second objective in presence of wind can lead to hazardous situations with respect to airspeed limits. So this objective is expressed through the on-line definition of a desired airspeed to be followed (it is supposed that online estimates of wind speed components are available). From a desired smooth time table $t_d(x)$, we get a desired ground speed $V_{G_d}(x)$:

$$V_{G_d}(x) = 1/(dt_d(x)/dx) \tag{19}$$

then, tacking into account an estimate of the longitudinal component of wind speed, a space-referenced desired airspeed $V_{air_d}(x)$ can be defined for low speeds by introducing a minimum margin with respect to the stall speed at the current desired level:

$$V_{air_d}(x) = \mathrm{Max}\{V_S(z_d(x)) + \Delta V_{\min}, V_{G_d}(x) - \hat{w}_x(x)\} \tag{20}$$

where V_S, $\Delta V_{\min}$ and $\hat{w}_x$ are respectively the stall speed, the minimum margin speed and the estimate of the horizontal wind speed. For high speeds, an airspeed less than the maximum operating speed at the current desired level:

$$V_{air_d}(x) = \mathrm{Min}\{V_{\mathrm{MO}}(z_d(x)), V_{G_d}(x) - \hat{w}_x(x)\} \tag{21}$$

where V_{MO} denotes the maximum operating speed.
In all other cases the desired airspeed is chosen such as:

$$V_{air_d}(x) = V_{G_d}(x) - \hat{w}_x(x) \tag{22}$$

5 Space-Based NLI Tracking Control

In this section the space-based nonlinear inverse control technique introduced in [18] to perform aircraft trajectory tracking is displayed. The relative degrees of output variables V_{air} and z can be determined from the following equations which are affine with respect to q and T_C :

$$V_{air}^{[2]} = \frac{1}{V_G^2}\left[A_V(z,\alpha,V_{air},T,W) + B_{V_q}(z,\alpha,V_{air},T,W)q + B_{V_T}(z,\alpha,V_{air},T,W)T_C\right] \tag{23}$$

$$z^{[3]} = \frac{1}{V_G^2}\left[A_z(z,\alpha,V_{air},T,W) + B_{z_q}(z,\alpha,V_{air},T,W)q + B_{z_T}(z,\alpha,V_{air},T,W)T_C\right] \tag{24}$$

where W represents the wind parameters w_x, w_z, $\dot{w}_x$, $\dot{w}_z$ and $\ddot{w}_x$ and $\ddot{w}_z$. The rather complex expressions of scalars A_V , B_{V_q} , B_{V_T} and A_z , B_{z_q} , B_{z_T} in (17) and (18) are detailed in [17].

The B_i terms are in general different from zero and the spatial relative degree of V_{air} and z are respectively $r_V = 1$ and $r_z = 2$. Then if V_{air} and z are chosen as tracked variables, there will be no internal dynamics to worry about. Now, since in standard flight conditions the control matrix B given by:

$$B = \begin{pmatrix} B_{z_q} & B_{z_T} \\ B_{V_q} & B_{V_T} \end{pmatrix} \tag{25}$$

is invertible, it is possible to perform a direct dynamic inversion to get effective trajectory tracking control laws, [22]. So we get:

$$\begin{pmatrix} q \\ T_C \end{pmatrix} = \begin{pmatrix} B_{z_q} & B_{z_T} \\ B_{V_q} & B_{V_T} \end{pmatrix}^{-1} \times \begin{pmatrix} V_G^2 D_z(x) - A_z \\ V_G^2 D_{V_{air}}(x) - A_V \end{pmatrix} \tag{26}$$

with:

$$D_z(x) = z_d^{[3]}(x) + k_{1z}\xi_z^{[2]}(x) + k_{2z}\xi_z^{[1]}(x) + k_{3z}\xi_z(x) \tag{27}$$

$$D_{V_{air}}(x) = V_{air_d}^{[2]}(x) + k_{1v}\xi_{V_{air}}^{[1]}(x) + k_{2v}\xi_{V_{air}}(x) \tag{28}$$

where with $\xi_z(x)$ and $\xi_{V_{air}}(x)$ are the tracking errors related to the desired altitude $z_d(x)$ and desired airspeed profile $V_{air_d}(x)$:

$$\xi_z(x) = z(x) - z_d(x) \tag{29}$$

$$\xi_{V_{air}}(x) = V_{air}(x) - V_{air_d}(x) \tag{30}$$

Then the tracking error variables follow the linear dynamics:

$$\xi_{V_{air}}^{[2]}(x) + k_{1v}\xi_{V_{air}}^{[1]}(x) + k_{2v}\xi_{V_{air}}(x) = 0 \tag{31}$$

$$\xi_z^{[3]}(x) + k_{1z}\xi_z^{[2]}(x) + k_{2z}\xi_z^{[1]}(x) + k_{3z}\xi_z(x) = 0 \tag{32}$$

where k_{1v}, k_{2v} , k_{1z} , k_{2z} and k_{3z} are real parameters which must be chosen such as the roots of $s^2 + k_{1v}s + k_{2v}$ and $s^3 + k_{1z}s^2 + k_{2z}s + k_{3z}$ produce adequate tracking error dynamics (stability and reduced oscillations). Here s denotes the Laplace variable.

Observe here that while the successive spatial derivatives of desired outputs $z_d(x)$ and $V_{air_d}(x)$ can be directly computed, the computation of the successive spatial derivatives of actual outputs $z(x)$ and $V_{air}(x)$ includes the wind parameters which have been replaced by their estimates.

Then we get a new two level control structure where the first layer corresponds to a fast control loop for the pitch rate (autopilot) and the thrust (autothrottle) on a time scale basis, while the second control layer, operating on a space scale basis, corresponds to a slow control loop of groundspeed and height.

6 Simulation Results

The proposed guidance approach is illustrated using the Research Civil Aircraft Model (RCAM) which has the characteristics of a wide body transportation aircraft, see again [23], with a maximum allowable landing mass of about 125 tons with a nominal landing speed of 68m/s. There, the control signals are submitted to rate limits and saturations as follows:

$$-15\frac{\pi}{180}\,\text{rad/s} \le \dot{\delta}_e \le 15\frac{\pi}{180}\,\text{rad/s} \tag{33}$$

$$-25\frac{\pi}{180}\,\text{rad} \le \delta_e \le 10\frac{\pi}{180}\,\text{rad} \tag{34}$$

$$-1.6\frac{\pi}{180}\,\text{rad/s} \le \dot{T}_C \le 1.6\frac{\pi}{180}\,\text{rad/s} \tag{35}$$

$$0.5\frac{\pi}{180}\,\text{rad} \le T_C \le 10\frac{\pi}{180}\,\text{rad} \tag{36}$$

While the minimum allowable speed is $1.23V_{stall}$ with $V_{stall} = 51.8\text{m/s}$ and the angle of attack is limited to the domain $\lfloor -11.5^{\circ}, 18^{\circ} \rfloor$ where $\alpha_{stall} = 18^{\circ}$.

6.1 Simulation Results in No-Wind Condition

In a no wind condition, Figure 2 displays altitude tracking performances resulting from a space NLI guidance scheme, while Figure 3 provides closer views of altitude and tracking performance during initial transients. Figure 4 displays airspeed tracking performances of a space NLI guidance scheme when the aircraft is initially late according to the planned time table. It appears clearly that the aircraft increases its airspeed to the maximum operating speed during 12000m until it catches up its delay.

Figures 5 and 6 display respectively the evolution of respectively the angle of attack, the flight path angle, the elevator deflection and the throttle setting during the whole manoeuvre. Since the angle of attack remains in a safe domain and the considered longitudinal inputs remain by far unsaturated this demonstrates the feasibility of the manoeuvre.

Figure 7 and Figure 8 show respectively airspeed and time tracking performances in two cases. The first one considers a delay situation for an aircraft according to a reference time table where the aircraft maintains its airspeed at V_{MO} until it compensates the initial delay. In the second situation the aircraft is initially in advance with respect to the planned time table and in this case the speed controller sets its airspeed to the minimum allowable speed until the time tracking error is eliminated.

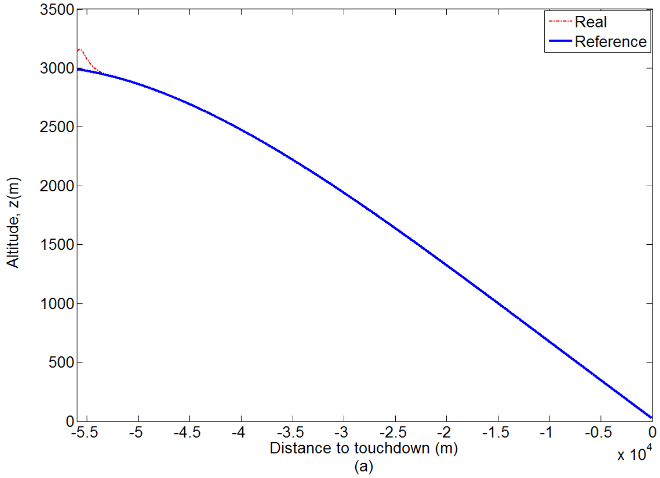


Fig. 2 Desired vertical tracking performance with space NLI (no wind).

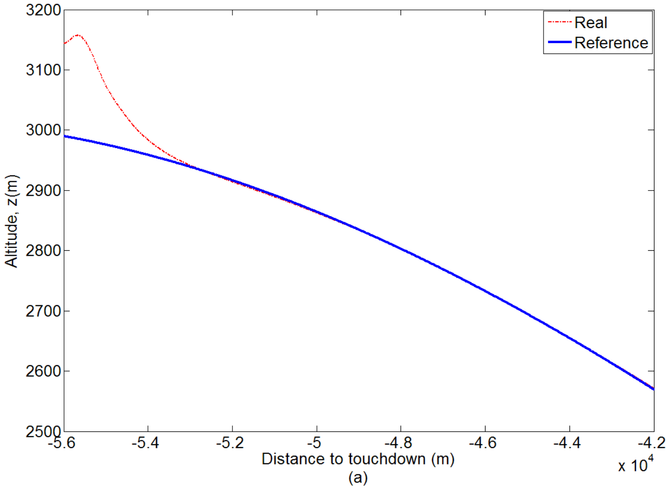


Fig. 3 Initial vertical tracking by space NLI (no wind)

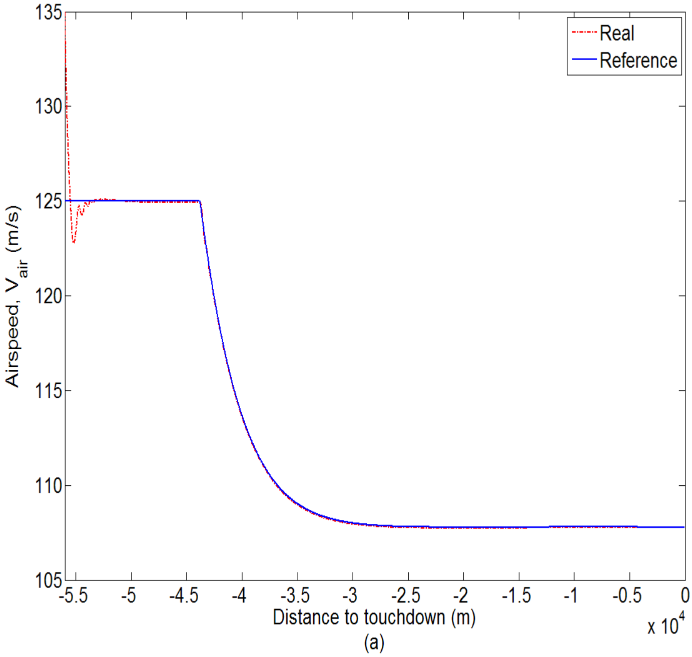


Fig. 4 Desired airspeed tracking performance (no wind)

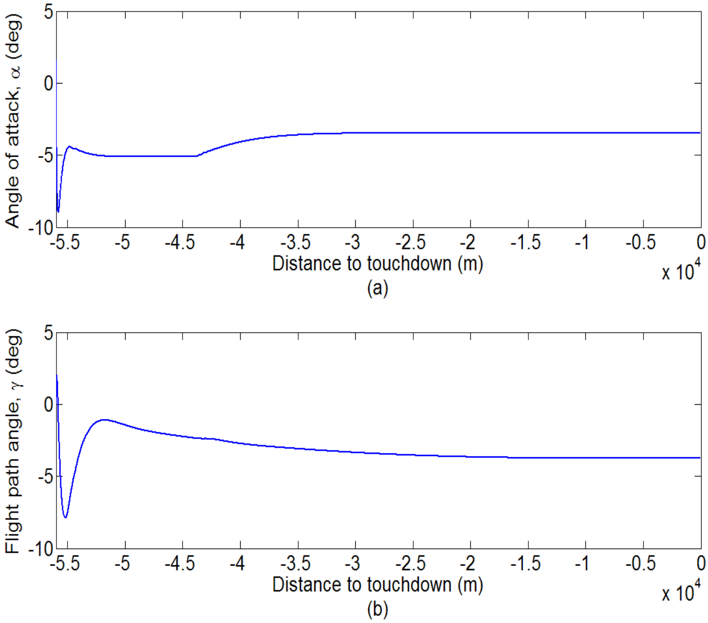


Fig. 5 Angle of attack and flight path angle with space NLI (no wind)

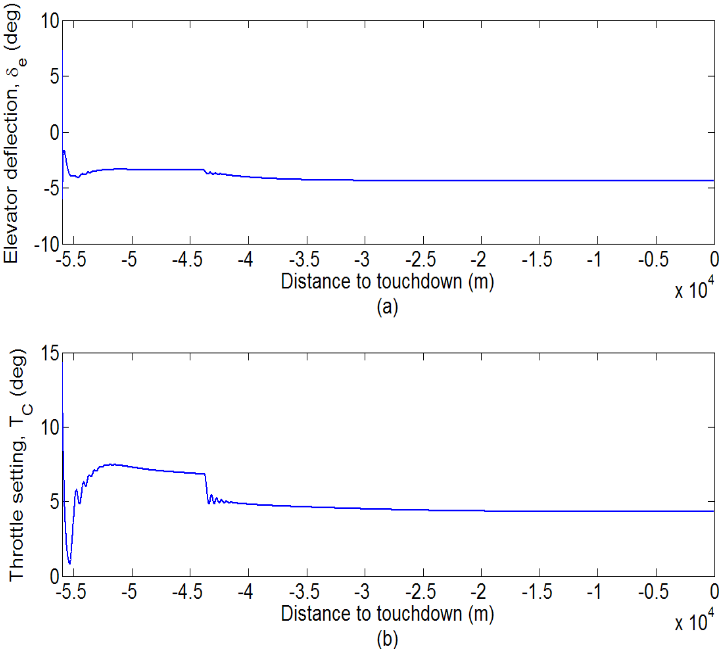


Fig. 6 Control inputs with space NLI (no wind)

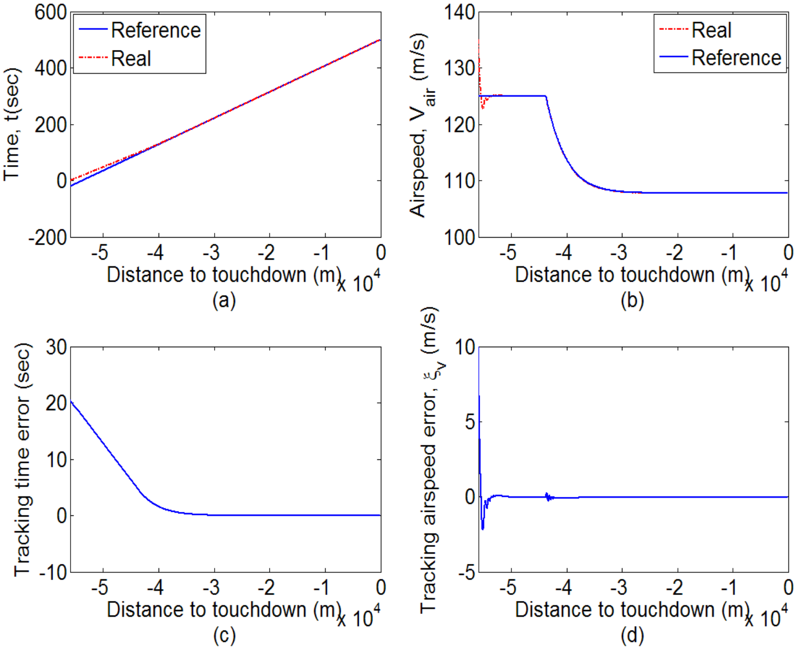


Fig. 7 Delayed initial situation and recover

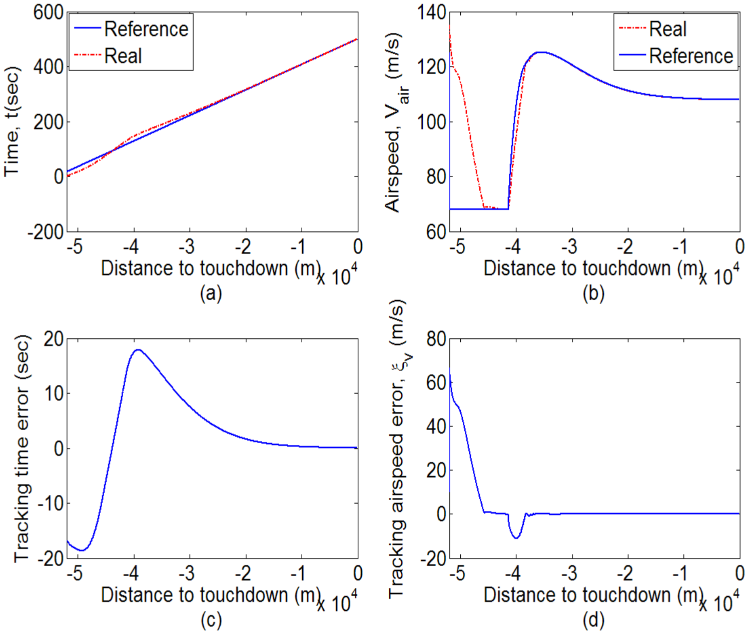


Fig. 8 Advanced initial situation and recover

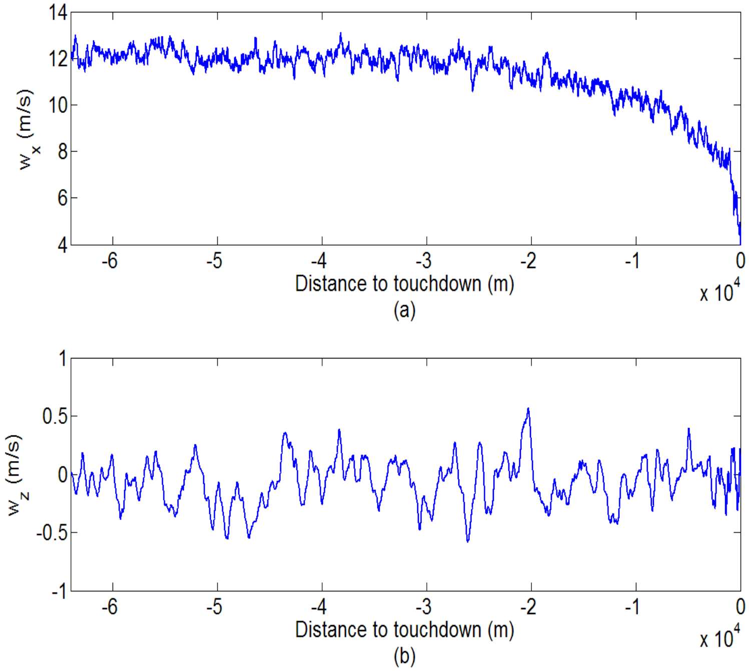


Fig. 9 Example of wind components space history

6.2 Simulation Results in the Presence of Wind

Here a tailwind with a mean value of 12m/s has been considered. Figure 9 provides an example of realization of such wind.

Since in this study the problem of the online estimation of the wind components has not been tackled, it has been supposed merely that the wind estimator will be similar to a first order filter with a space constant equal to 28m in the other case (space NLI guidance). Then the filtered values of these wind components have been fed to the space NLI guidance control law.

Figure 10 and Figure 11 display altitude, airspeed and time tracking performances in the presence of the wind when the actual time table is late and in advance situations according to the reference time table, respectively.

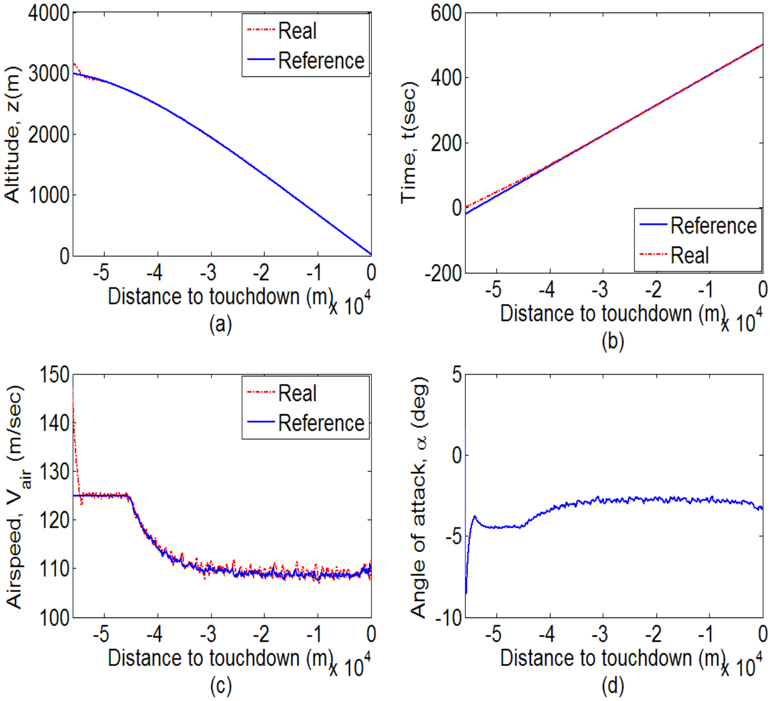


Fig. 10 Delayed initial situation and recover with wind

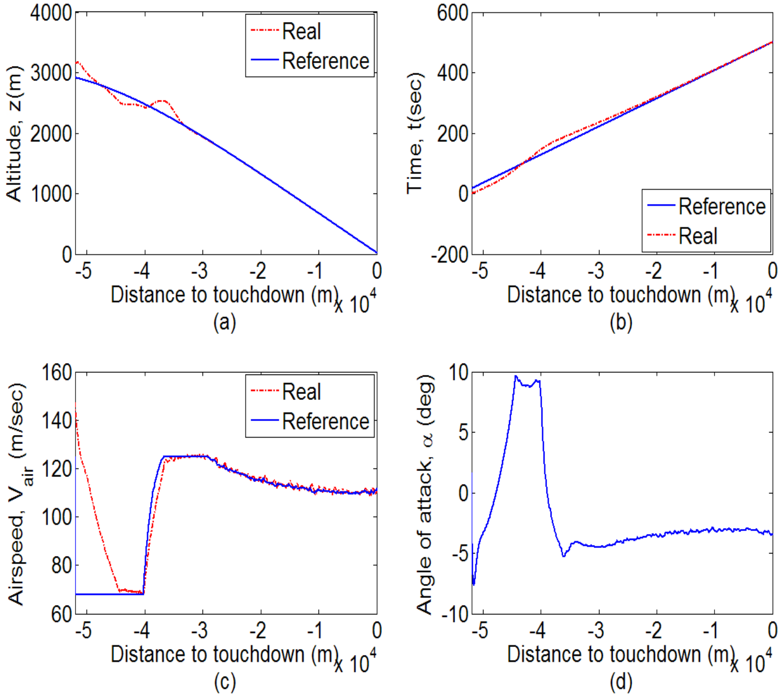


Fig. 11 Advanced initial situation and recover with wind

7 Conclusion

In this communication a new longitudinal guidance scheme for transportation aircraft has been proposed.

The main objective here has been to improve the tracking accuracy performance of the guidance along a desired longitudinal trajectory referenced in a spatial frame. This has led to the development of a new representation of longitudinal flight dynamics where the independent variable is ground distance to a reference point. The nonlinear inverse control technique has been applied in this context so that tracking errors follow independent and asymptotically stable spatial dynamics around the desired trajectories. It has been shown also that a similar tracking objective expressed in the time frame cannot be equivalent when the desired airspeed changes as it is generally the case along climb and approach for landing.

Tracking performances obtained from this spatial NLI guidance scheme have been analyzed through a simulation study considering the descent maneuver of a transportation aircraft in wind and no wind conditions.

To get applicability this new guidance approach still should overcome important challenges related mainly with navigation and online wind estimation performances. Then an improved integration of on board flight path optimization functions including the consideration of neighbouring traffic and the guidance function will become possible.

References

[1] http://www.sesarju.eu/programme/highlights/i-4d-flight

[2] De Smedt, D., Putz, T.: Flight Simulations Using Time Control With Different Levels of Flight Guidance. In: IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, Florida, USA, October 25-29, pp. 2.C.5-1–2.C.5-15 (2009)

[3] Jurgen, T., Offman, E.: Aircraft in the Future ATM System-Exploiting the 4D Aircraft Trajectory. In: IEEE/AIAA 23rd Digital Avionics Systems Conference, USA, pp. 3.B.2-1–3.B.2-21 (2004)

[4] Pappas, G., Tomlin, C., Lygeros, J., Godbole, D., Sastry, S.: A Next Generation Architecture for Air Traffic Management Systems. In: IEEE Proceeding of the 36th Conference on Decision and Control, San Diego, California, USA, pp. 2405–2411 (December 1997)

[5] Korn, B., Kuenz, A.: 4D FMS for Increasing Efficiency of TMA Operations. In: IEEE/AIAA 25th Digital Avionics Systems Conference, USA, pp. 1E4-1–1E4-8 (2006)

[6] Young, S.-Y.: Real-Time 4-D Trajectory Planning for RNP Flight Operations. In: IEEE Integrated Communications, Navigation and Surveillance Conference, pp. 1–9 (2009)

[7] Cano, M., Sanchez-Escalonilla, P., Dorado, M.M.: Complexity Analysis in the Next Generation of Air Traffic Management System. In: IEEE/AIAA 26th Digital Avionics Systems Conference, Dallas, Texas, USA, pp. 3.D.4-1–3.D.4-9 (2007)

[8] Miele, A., et al.: Guidance Strategies for Near-Optimum Takeoff Performance in Wind Shear. Journal of Optimization Theory and Applications 50(1) (1986)

[9] Miele, A., et al.: Optimization and Gamma/Theta Guidance of Flight Trajectories in a Windshear. Presented at the 15th ICAS Congress, London (1986)

[10] Miele, A., et al.: Optimal Trajectories and Guidance Trajectories for Aircraft Flight Through Windshears. In: Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii, pp. 737–746 (1990)

[11] Psiaki, M.L., Stengel, R.F.: Analysis of Aircraft Control Strategies for Microburst Encounter. Journal of Guidance, Control, and Dynamics 8(5), 553–559 (1985)

[12] Psiaki, M.L., Park, K.: Thrust Laws for Microburst Wind Shear Penetration. Journal of Guidance, Control, and Dynamics 15(4) (1992)

[13] Singh, S.N., Rugh, W.J.: Decoupling in a Class of Nonlinear Systems by State Feedback. ASME Journal of Dynamic Systems, Measurement, and Control, Series G 94, 323–329 (1972)

[14] Stengel, R.F.: Toward Intelligent Flight Control. IEEE Trans. on Systems, Man, and Cybernetics 23(6), 1699–1717 (1993)

[15] Mulgund, S.S., Stengel, R.F.: Optimal Nonlinear Estimation for Aircraft Flight Control in Wind Shear. Automatica 32(1) (January 1996)

[16] Psiaki, M.L.: Control of Flight Through Microburst Wind Shear Using Deterministic Trajectory Optimization. Ph.D. Thesis, Department of Mechanical and Aerospace Engineering, Princeton University, Report No. 1787-T (1987)

[17] Bouadi, H., Mora-Camino, F.: Space-Based Nonlinear Dynamic Inversion Control for Aircraft Continuous Descent Approach. In: IEEE Evolving and Adaptive Intelligent Systems Conference, Madrid, Spain, May 17-18, pp. 164–169 (2012)

[18] Bouadi, H., Mora-Camino, F.: Aircraft Trajectory Tracking by Nonlinear Spatial Inversion. Accepted in AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota, USA, August 13-16 (2012)

[19] Etkin, B.: Dynamics of Atmospheric Flight. John Wiley and Sons, Inc., New York (1985)

[20] Isidori, A.: Nonlinear Control Systems. Springer, Berlin

[21] Frost, W., Bowles, R.: Wind Shear Terms in the Equations of Aircraft Motion. Journal of Aircraft 21(11), 866–872 (1984)

[22] Campbell, C.W.: A Spatial Model of Wind Shear and Turbulence for Flight Simulation. Tech. Rep. TP-2313, NASA George C. Marshall Space Flight Centre, Alabama 35812 (May 1984)

[23] Magni, J.-F., et al.: Robust Flight Control, A Design Challenge. Springer, London

Adaptive Trajectory Controller for Generic Fixed-Wing Unmanned Aircraft

Maximilian Mühlegg, Johann C. Dauer, Jörg Dittrich, and Florian Holzapfel

Abstract. This work deals with the construction of a nonlinear adaptive trajectory controller, which is easily applicable to a multitude of fixed wing unmanned aircraft. Given a common signal interface, the adaptive trajectory controller is divided into a generic part, which is common for each vehicle, and into a part, which is unique. The generic part of the control architecture bases on a common inversion model which is used for feedback linearization. However, the dynamics of the aircraft and the inversion model differ, thus introducing model uncertainties to the feedback linearized system. The effect of modeling uncertainties is reduced by the application of a concurrent learning model reference adaptive controller, which uses neural networks in order to approximate the uncertainty. Leveraging instantaneous as well as stored data concurrently for adaptation ensures convergence of the adaptive parameters to a set of optimal weights, which minimize the approximation error. Performance and robustness against certain model uncertainties is shown through numerical simulation for two significantly different unmanned aircraft.

1 Introduction

Unmanned Aircraft (UA) have attracted a high amount of attention in recent years. Research institutes encounter a lot of challenging tasks, while handling an increasing number of aerial vehicles. In order to perform flight tests in different research

Maximilian Mühlegg · Florian Holzapfel
Institute of Flight System Dynamics,
Technische Universität München, Boltzmannstraße 15,
85748 Garching bei München, Germany
e-mail: {maximilian.muehlegg,florian.holzapfel}@tum.de

Johann C. Dauer · Jörg Dittrich
Institute of Flight Systems, German Aerospace Center,
Lilienthalplatz 7, 38108 Braunschweig
e-mail: {johann.dauer,joerg.dittrich}@dlr.de

areas, reliable flight control systems need to be available for each aircraft. In UA projects a considerable amount of time and effort is spent creating reliable control architectures. In an attempt to decrease the effort for the controller design the Institute of Flight Systems at the German Aerospace Center (DLR) analyzed in collaboration with the Institute of Flight System Dynamics (FSD) at the Technische Universität München, to which extent such a control architecture can be generalized. As a basis for this the two significantly different aircraft Prometheus (DLR) and ExtremeStar (FSD) serve as testing platforms. The goal of this paper is to develop a flight control architecture, which can easily be applied to different fixed-wing aircraft at both institutes, thus aiming to decrease the time and effort needed in order to set up a flight control system. The underlying assumption in this connection is that the fixed wing UA share the same dynamical structure.

In classical linear control theory the parameters of the aircraft and controller gains satisfying desired performance and robustness requirements have to be found for a set of trimming points throughout the flight envelope. While this approach has shown to be reliable to control an aircraft, parameter identification and the selection of gains has to be performed for each UA individually ([6, 22]). Furthermore, the received database might become unreliable once the configuration of the aircraft changes, further increasing the amount of time and effort which needs to be put into the design of the flight control system. This disadvantage especially applies to (fixed wing) UA, since they operate in a huge range of altitude and velocity.

Using nonlinear control methods, trimming points and their blending can be avoided. One such technique is feedback linearization, also called dynamic inversion. The idea is to transform the nonlinear system into an equivalent linear form ([11]). For the resulting system linear control methods such as a linear tracking control design can be applied. The benefit compared to the classic linearization is that the transformed system is valid throughout a wider part of the flight envelope and not limited to the close vicinity of selected trimming points. However, feedback linearization requires accurate knowledge of the dynamics and parameters of the nonlinear system. In general, nonlinear systems are of infinite order and only estimates of the real parameters are available. Therefore, feedback linearization can only be performed with respect to a model of the nonlinear dynamical system, inevitably introducing uncertainties to the system.

Model Reference Adaptive Control (MRAC) is concerned with reducing the impact of such modeling uncertainties ([1, 18]). In the framework of a feedback linearized system, the idea is to make the real dynamics behave like the model chosen for the inversion. If the structure of the uncertainty is known, it can be linearly parameterized by a weighted combination of its (known) basis ([1, 18, 26]). If the structure is unknown universal approximators can be employed to reduce the impact of the uncertainty ([4, 11, 13, 14, 19, 23]). Therefore a-priori chosen regressor functions are weighted by a set of adaptive parameters, which are updated based on the instantaneous tracking error. The underlying assumption is that there exists an ideal set of weights which result in the approximation with the smallest error. In classical adaptive control the convergence of the adaptive parameters to their optimal values is only achieved if the regressor vector is persistently excited (PE) ([26]). To ensure

PE on the regressor vector is often operationally undesirable and might not even be possible.

Concurrent learning uses instantaneous information concurrently with specifically stored data for future updates of the parameters ([5, 7]): The idea is, that if data points are stored at a time when the regressor vector was excited, this information can be used for future updates to ensure parameter convergence. For this purpose the stored data only has to meet a requirement of linearly independency on the stored data, the regressor vector is not required to be PE.

In order to construct a trajectory controller based on feedback linearization and concurrent learning MRAC for a multitude of UA, a common signal interface needs to be defined. Especially, the states chosen for the feedback linearization approach need to be available through a suitable set of sensors and sensor fusion algorithms. The adaptive controller is then used to drastically reduce the effects of the uncertainty resulting from a deviation between the plant dynamics and the inversion model. In fact, the concurrent learning adaptive controller ensures that the same inversion model can be used for a multitude of fixed wing UA. In order to show that the control architecture is applicable to a multitude of UA and in order to show robustness against uncertainties, the controller is tested in numerical simulation for two significantly different fixed wing UA, namely Prometheus (DLR) and ExtremeStar (FSD).

In this paper, for a given vector a, $(a)_*$ denotes the frame in which a is notated and a_* denotes the physical type of a. For example $(V_K)_B$ denotes the kinematic velocity notated in the body-fixed frame. Furthermore, (ω^{xy}) describes the rotational rates of the y system relative to the x system. All signals are notated with respect to the center of gravity if not stated otherwise. The outline of this paper is as follows: In Section 2 the control strategy based on feedback linearization and an adaptive element is described. In Section 3 we present the results from numerical simulation in a Software- and Hardware-in-the-Loop framework. The paper is concluded in Section 4.

2 Control Architecture

The concept of the proposed control strategy is presented in Figure 1. The goal of the paper is to define a control architecture which is simply applicable to a variety of fixed wing UA. The underlying assumption is that multiple fixed wing UA share the same dynamic structure. However, it is not yet possible to construct an inversion controller which is simply applicable to every aircraft. Instead, the proposed nonlinear trajectory controller is divided into two parts: A generic inversion controller, including reference models, and the control allocation.

The generic inversion controller is independent of the individual fixed wing UA and is constructed based on a shared inversion model. Still, the parameters in this block, in particular proportional and integral controller gains as well as the parameter of the reference model dynamics, have to be adjusted in order to suit each individual platform. Another part of this block is the concurrent learning adaptive controller, which minimizes the deviation between the plant dynamics and the chosen inversion model.

In order for this block to be valid for several platforms, a common signal interface has to be defined. This interface has to meet two requirements. On the one hand, the model of the plant has to be feedback linearizable with the chosen set of states provided by the interface. On the other hand, the chosen signals have to be measurable or at least computable from measurable data by the application of appropriate sensor fusion algorithms.

Finally, the control allocation maps the desired outputs of the generic inversion controller onto the control devices of the aircraft. Since the number and type of control devices can vary strongly between UA, this part has to be constructed individually for each aircraft.

The outline of this section is as follows: Section 2.1 defines a common signal interface for the proposed generic inversion controller. In section 2.2 dynamic inversion is performed with respect to a selected inversion model. Section 2.3 introduces the adaptive component which reduces the effects of modeling uncertainties. Finally, Section 2.4 briefly discusses the control allocation.

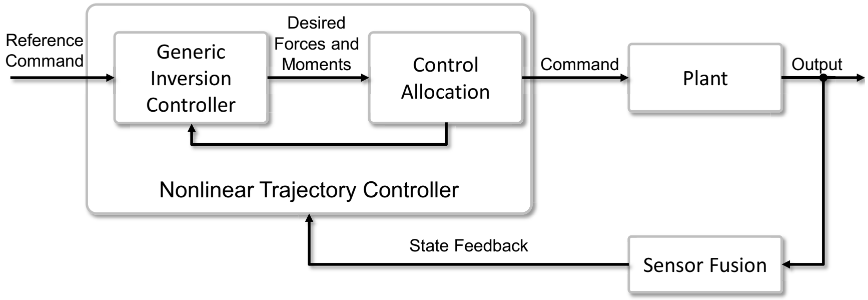

Reference Command
Generic Inversion Controller
Desired Forces and Moments
Control Allocation
Command
Plant
Output
Nonlinear Trajectory Controller
State Feedback
Sensor Fusion

Fig. 1 General control strategy

2.1 Signal Interface Definition

Unmanned aircraft can be equipped with a huge variety of sensors each relaying on different information about the states of an aircraft. Furthermore, it is possible to perform feedback linearization of a dynamic system relying on different sets of states, the attitude dynamics can, for example, be inverted by using either Euler angles or kinematic attitude angles. Table 1 gives a summary of the specific signal interface definitions used in this work. All signals which are required as an input are labeled with ’in’. The commands generated by the inversion controller are labeled with ’out’. A feasible set of sensors which ensures that the required signals in Table 1 can be obtained and the proposed control architecture can be used is given by a pitot tube, an Inertial Measurement Unit, a Magnetometer and a GPS module. Furthermore, the body-fixed moments and the force which acts in the direction of the velocity vector need to be estimated. For this purpose the chosen inversion model can be used together with a model of the actuators ([11]).

Table 1 Interface definitions for the proposed generic inversion controller

Character	Context	Symbol	Unit	Explanation
Reference Command (In)	Translation	$V_{K,CMD}$	$[\frac{m}{s}]$	Kinematic velocity
		$\gamma_{K,CMD}$	$[rad]$	Flight-path azimuth angle
		$\chi_{K,CMD}$	$[rad]$	Flight-path inclination angle
Sensor Fusion (In)	Position	h	$[m]$	Altitude
	Translation	$\begin{pmatrix} u \\ v \\ w \end{pmatrix}_B$	$[\frac{m}{s}]$	Body fixed velocity vector
	Attitude	$\begin{pmatrix} \Phi \\ \Theta \\ \Psi \end{pmatrix}$	$[rad]$	Euler angles
	Rotation	$\begin{pmatrix} p \\ q \\ r \end{pmatrix}_B$	$[\frac{rad}{s}]$	Body-fixed rotational rates
	Air Data	$\bar{q}$	$[\frac{N}{m^2}]$	Dynamic pressure
Estimated quantities (In)	Force	$(F_x)_K$	$[N]$	Estimated force in kinematic x-direction
	Moment	$\begin{pmatrix} L \\ M \\ N \end{pmatrix}_B$	$[Nm]$	Estimated moments
Control variable (Out)	Force	$(F_{x,CMD})_K$	$[N]$	Force in kinematic x-direction
	Moment	$\begin{pmatrix} L_{CMD} \\ M_{CMD} \\ N_{CMD} \end{pmatrix}_B$	$[Nm]$	Commanded moments

2.2 Approximate Model Inversion for Fixed Wing Aircraft

This section discusses the fundamentals of approximate model inversion and how this technique can be used to invert the dynamics of a fixed wing UA. Let $x(t) \in \mathbb{R}^n$ be the known state vector and let $\delta(t) \in \mathbb{R}^m$ denote the control input.

The general nonlinear dynamics of the aircraft can be written as

$$\dot{x}(t) = f(x(t), \delta(t)), \tag{1}$$

where the function f is assumed to be unknown yet sufficiently smooth. That is, its partial derivatives up to the required order are defined and continuous. Additionally, the control input $\delta(t)$ is assumed to be bounded and piecewise continuous. Since the exact model in (1) is usually neither available nor invertible, we introduce an approximate inversion model $\hat{f}(x(t), \delta(t))$. The inversion model $\hat{f}$ needs to be continuous and invertible with respect to $\delta(t)$. Given a pseudo-control input $\nu(t) = \hat{f}(x(t), \delta(t))$ these requirements need to be fulfilled in order to be able to find a control command $\delta(t)$ by dynamic inversion such that:

$$\delta(t) = \hat{f}^{-1}(x(t), \nu(t)) \tag{2}$$

This approximation results in a model error $\Delta \in \mathbb{R}^n$, which can be formulated in additive form:

$$\dot{x}(t) = v(x(t), \delta(t)) + \Delta(x(t), \delta(t)) \tag{3}$$

The feedback linearization approach used in this work is based on the dynamics of a fixed wing aircraft, which are not derived in detail here but can be found among others in [3]. The validity of the differential equations is connected to a set of assumptions. The earth is assumed to be flat and non-rotating. Hence, the transport rate and the angular velocity of the earth are neglected. These assumptions are valid, since the presented UA only fly short distances and operate for a limited time only. The aircraft is seen to be a rigid body, the relative motion of aircraft mass elements is thus considered to be zero. Furthermore, the mass as well as mass distribution are considered to be quasi stationary. For the inversion model the atmosphere is assumed to be static, that is, there is no wind. As a result the intermediate kinematic frame as well as the kinematic attitude angles equal the aerodynamic frame and the aerodynamic attitude angles. Furthermore, the kinematic attitude angles can be used in order to invert the attitude dynamics. However, in reality the atmosphere is not static. The forces and moments resulting from wind are therefore considered as disturbances.

Figure 2 shows the general concept for the feedback linearization approach. The system is cascaded along the dynamical chain of the aircraft, each loop having a relative degree of one. For each loop a separate reference model and tracking controller is constructed. The advantage of such a cascaded system is that the resulting analytical terms are easy to handle. A drawback of the cascaded approach is that the bandwidth of the overall system is reduced compared to an inversion with relative degree three. However, it is argued in [11] that in the second case signals can be required, which are heavily corrupted by measurement noise, thus favoring a cascaded approach. In the following the feedback linearization approach based on [11] for a fixed wing UA is depicted.

The actuator dynamics are neglected in the proposed inversion model. Since the input exhibits its own dynamics and limitations, an additional uncertainty is added to the system. Especially if the inverted system is combined with an adaptive element, actuator saturation can cause instability of the closed loop system through unbounded parameter growth during saturation. Pseudo Control Hedging (PCH) is a method which slows down the reference model dynamics by a measure of the expected reaction deficit of the plant, thus hiding actuator dynamics from the error dynamics and allowing adaptation even in the presence of saturation ([13]). Note, that from any loop the next inner loop can be viewed as a kind of actuator dynamics. Hence, PCH can be added separately to each loop and allows the simultaneous application of an adaptive element and artificial saturations in inner loops.

2.2.1 Inversion of the Path Dynamics

The outer loop is concerned with finding the proper kinematic attitude angles and thrust of the aircraft in order to follow a desired flight path. A nonzero angle of sideslip hereby leads to increased drag and an increase in the required thrust, reducing

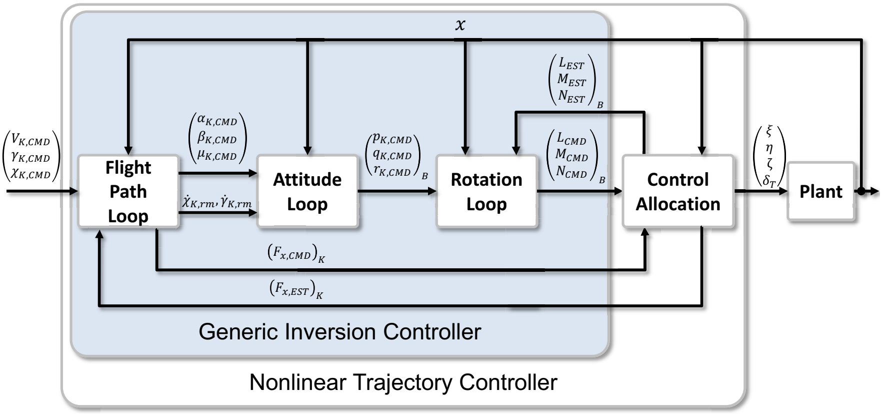

x
$\begin{pmatrix} V_{K,CMD} \\ \gamma_{K,CMD} \\ \chi_{K,CMD} \end{pmatrix}$
Flight Path Loop
$\begin{pmatrix} \alpha_{K,CMD} \\ \beta_{K,CMD} \\ \mu_{K,CMD} \end{pmatrix}$
$\dot{\chi}_{K,rm}, \dot{\gamma}_{K,rm}$
Attitude Loop
$\begin{pmatrix} p_{K,CMD} \\ q_{K,CMD} \\ r_{K,CMD} \end{pmatrix}_B$
Rotation Loop
$\begin{pmatrix} L_{EST} \\ M_{EST} \\ N_{EST} \end{pmatrix}_B$
$\begin{pmatrix} L_{CMD} \\ M_{CMD} \\ N_{CMD} \end{pmatrix}_B$
Control Allocation
$\begin{pmatrix} \xi \\ \eta \\ \zeta \\ \delta_T \end{pmatrix}$
Plant
$(F_{x,CMD})_K$
$(F_{x,EST})_K$
Generic Inversion Controller
Nonlinear Trajectory Controller

Fig. 2 Cascaded structure of the feedback linearization approach for the generic inversion controller

the efficiency of the flight. In order to preserve an aerodynamically efficient flight, the kinematic angle of sideslip β_K is commanded to be zero. Basis for the inversion of the path dynamics are the following equations of motion derived with Newtons Second Law:

$$\begin{aligned} \dot{V}_K &= \tfrac{T-D}{m} - gsin(\gamma_K) \\ \dot{\chi}_K &= \tfrac{Lsin\mu_K}{mV_K cos(\gamma_K)} \\ \dot{\gamma}_K &= \tfrac{Lcos\mu_K}{mV_K} - \tfrac{g}{V_K} cos(\gamma_K) \end{aligned} \tag{4}$$

Here T, D and L denote the thrust, the drag and the lift respectively. In the equations above it is assumed that the thrust only acts in the direction of the velocity vector. The path dynamics can be inverted by applying the dynamic inversion concept to equation (4). However, Holzapfel argues in [11] that especially if the input of a dynamical system is described by an irrational function, a pure mathematical inversion can result in a multitude of solutions, which are often physically meaningless. Alternatively, a physically and geometrically motivated approach is used. For feedback linearization the forces in a plane perpendicular to the current velocity vector are considered. For the inversion model, assume that the lift coefficient C_L is only dependent on the zero-lift derivative $C_{L,0}$ and the derivative resulting from a change in the angle of attack $C_{L,\alpha}$. With $C_L = C_{L,0} + C_{L,\alpha}\alpha_K$ and $L = \frac{1}{2}\rho V_K^2 S C_L$ the linearizing feedback for the kinematic attitude angles results in

$$\begin{aligned} \alpha_{K,CMD} &= \tfrac{2L_{CMD}}{\rho V_K^2 S C_{L,\alpha}} - \tfrac{C_{L,0}}{C_{L,\alpha}} \\ \beta_{K,CMD} &= 0 \\ \mu_{K,CMD} &= arctan\left(\tfrac{\nu_\chi}{\nu_\gamma + \frac{g}{V_K} cos\gamma_K}\right), \end{aligned} \tag{5}$$

where $\nu_{OL} = [\nu_V, \nu_\gamma, \nu_\chi]^T$ denotes the pseudo control variables in the outer loop and $L_{CMD} = mV_K \sqrt{(\nu_\gamma + \frac{g}{V_K} cos(\gamma_K))^2 + \nu_\chi^2}$. The thrust equation is inverted by

$$T_{CMD} = F_{x,CMD} = m(\nu_V + gsin(\gamma_K)) + D. \tag{6}$$

For each state of the path dynamics a separate first order reference model can be constructed. A linear feedback with proportional and integral components is used in order to construct the pseudo control variable ν_{OL}. Finally, PCH is used in order to slow down the reference model dynamics of the path states by a measure of the reaction deficit of the plant.

2.2.2 Inversion of the Attitude Dynamics

The middle loop is concerned with finding the angular rates in order to realize the desired kinematic attitude angles α_K and μ_K as well as keeping the angle of sideslip β_K equal to zero in order to preserve an aerodynamically efficient flight. The basis for the inversion of the attitude dynamics is a purely kinematic relation between the angular rates and the kinematic attitude angles. The equations of motion can be derived from the strapdown equation ([25]):

$$\begin{pmatrix} \dot{\alpha}_K \\ \dot{\beta}_K \\ \dot{\mu}_K \end{pmatrix} = M\left[M_{KB}(\omega_K^{OB})_B - (\omega_K^{OK})_K\right], \tag{7}$$

where $M_{KB} = M_{BK}^T$ denotes the transformation matrix between the body-fixed and the kinematic frame and

$$M = \begin{bmatrix} 0 & \frac{cos(\mu_K)}{cos(\beta_K)} & \frac{sin(\mu_K)}{cos(\beta_K)} \\ 0 & sin(\mu_K) & -cos(\beta_K) \\ 1 & -tan(\beta_K)cos(\mu_K) & tan(\beta_K)sin(\mu_K) \end{bmatrix}. \tag{8}$$

The linearizing feedback for the attitude dynamics is given by

$$(\omega_{K,CMD}^{OK})_B = M_{BK}[(\omega_K^{OK})_K + M^{-1}\nu_{ML}], \tag{9}$$

where ν_{ML} denotes the pseudo control variables for the middle loop and

$$(\omega_K^{OK})_K = \begin{pmatrix} -\dot{\chi}_K sin(\gamma_K) \\ \dot{\gamma}_K \\ \dot{\chi}_K cos(\gamma_K) \end{pmatrix}. \tag{10}$$

The time derivatives $\dot{\gamma}_K$ and $\dot{\chi}_K$ are not measurable. Instead they can be computed from the load factors, which in turn are dependent on the accelerations and are therefore highly susceptible to measurement noise. As an alternative the states of the reference model from the outer loop $\dot{\gamma}_{K,rm}$ and $\dot{\chi}_{K,rm}$ can be used to construct the linearizing feedback. The pseudo-control input ν_{ML} is constructed using a proportional feedback controller. If the attitude dynamics are to be analyzed separately, an integrator can be added to the linear error controller in order to ensure steady state accuracy.

2.2.3 Inversion of the Rotational Dynamics

The basis for the inversion of the rotational dynamics are the equations of angular momentum. They are derived applying the law of conservation of angular momentum with respect to the center of gravity:

$$(\dot{\omega}_K^{OB})_B^B = I_{BB}^{-1}\{(M)_B - (\omega_K^{OB})_B \times \left[I_{BB}(\omega_K^{OB})_B\right]\} \tag{11}$$

Here $(M)_B$ denotes the moments about the aircraft and I_{BB} denotes the mass moment of inertia. The linearizing feedback for the inner loop is given with

$$M_{CMD})_B = (\omega_K^{OB})_B \times \left[I_{BB}(\omega_K^{OB}))_B\right] + I_{BB}\nu_{IL}, \tag{12}$$

where ν_{IL} denotes the pseudo control variable for the inner loop. For each angular rate a separate reference model is constructed. For the pseudo-control ν_{IL} a proportional feedback is used. Finally, PCH is applied to the inner loop, thus allowing the addition of an adaptive element in the presence of actuator dynamics.

2.3 Adaptive Controller

Feedback linearization is concerned with transforming a nonlinear system such that it exhibits linear input-output behavior. A major drawback of this approach is that this method is highly susceptible to parameter errors and unmodeled dynamics. In order to decrease the impact of model uncertainties an adaptive element is added to the feedback linearized system. Model Reference Adaptive Control for feedback linearized systems aims to make the uncertain system behave like the underlying inversion model. MRAC achieves this goal by utilizing a model of the uncertainty. If the uncertainty can be parameterized linearly a weighted combination of the basis of the uncertainty can be employed as an adaptive element ([2, 15, 17]). The assumption on structural knowledge of the uncertainty can be relaxed by requiring it to be at least continuous and defined over a compact domain. In this case neural networks have been repeatedly used as adaptive elements ([4, 11, 13, 14, 19, 23]). Instead of the basis of the uncertainty, neuro-adaptive control utilizes a set of chosen basis functions. In particular, a Gaussian Network with radial basis functions ([21]) is employed in this work. However, also other activation functions such as Sigmoids or B-splines are imaginable ([11, 24]). The output of the adaptive element $\nu_{ad} \in \mathbb{R}^m$ is given by

$$\nu_{ad} = W^T(t)\sigma(x(t)). \tag{13}$$

Here $W(t) \in \mathbb{R}^{(n_2+1)\times m}$ denotes the adaptive weights and $\sigma(x(t)) \in \mathbb{R}^{(n_2+1)}$ denotes the regressor vector containing n_2 radial basis functions and a constant bias. According to the universal approximation property of Radial Basis Function Neural Networks ([20]) we have, that given a fixed number of radial basis functions n_2, there exist ideal weights $W^* \in \mathbb{R}^{(n_2+1)\times m}$ and a vector $\varepsilon \in \mathbb{R}^m$ such that given a compact domain $D \subset \mathbb{R}^n$ the following approximation holds for all $x \in D$:

$$\Delta(x,\delta) = W^{*^T}\sigma(x) + \tilde{\varepsilon}(x) \tag{14}$$

The functional approximation error $\bar{\varepsilon} = sup_{x\in D}\|\tilde{\varepsilon}(x)\|$ can be made arbitrarily small by increasing the number of radial basis functions ([20]). Define the tracking error as $e(t) = x(t) - x_{rm}(t)$, where $x_{rm}(t) \in \mathbb{R}^n$ represents the states of a reference model. For a positive definite matrix $Q \in \mathbb{R}^{n\times n}$ there exists a positive definite matrix $P \in \mathbb{R}^{n\times n}$ which satisfies the Lyapunov equation $A_e^T P + PA_e + Q = 0$, where $A_e \in \mathbb{R}^{n\times n}$ is Hurwitz and denotes the state of the error dynamics formed by the respective pseudo control variables. A common update law ([1, 11, 18]) for the adaptive weights is then given by

$$\dot{W}(t) = -\Gamma\sigma(x)e^T PB - m(x,W). \tag{15}$$

Here $m(x,W) \in \mathbb{R}^{m\times(n_2+1)}$ denotes a modification term, which is required in order to guarantee boundedness of the adaptive weights and therefor stability of the closed loop system. These include among others σ - Modification ([12]), e - Modification ([17]) or Q-Modification ([28]).

The linear control design based on the chosen inversion model is only valid if the effect of the uncertainty on the feedback linearized system is minimized. This in turn requires the convergence of the adaptive parameters to a set of weights which approximates the uncertainty best and therefore minimizes the functional approximation error. In classical adaptive control parameter convergence is subject to a condition of PE on the regressor vector. To ensure PE regressor vectors in neuro-adaptive control is in most cases neither possible nor operationally desirable.

In this work, instead of one of the previously mentioned modification terms, concurrent learning adaptive control ([5, 7]) is used. Concurrent learning uses online recorded information concurrently with current data in order to update the adaptive parameters. The key idea is storing information at a time when the regressor vector was exciting, and using this data for future updates. This allows the parameters to converge to a set of weights which minimize the functional approximation error.

Concurrent learning achieves parameter convergence by comparing the current estimation of the uncertainty ν_{ad} with the stored one and updating the adaptive parameters based on this deviation. Therefore, the regressor vector $\sigma_j(x)$ at certain time instants t_j is stored in a matrix $\sigma_H = [\sigma_1, \sigma_2, ..., \sigma_p]$, where $\sigma_H \in \mathbb{R}^{(n_2+1)\times p}$ is called history stack in the following and $p \geq (n_2+1)$. Only regressor vectors are stored which are linearly independent to the already stored data points. Once the history stack is full, methods such as the minimum singular value maximization approach ([8]) exist to include additional points by exchanging them with older data. Apart from the regressor vectors, also the model uncertainty Δ_j has to be determined. With regard to equation (3) the model uncertainty Δ_j at a time instant t_j is calculated by

$$\Delta_j = \dot{x}_j - \nu_j. \tag{16}$$

In order to solve equation (16), knowledge about the first state derivative $\dot{x}_j$ is required. In most cases, state derivatives cannot be measured directly and have to be estimated. For concurrent learning, the estimates of $\dot{x}_j$ do not have to be available instantaneously. Rather estimation methods can be applied which require an amount of time in order to arrive at a good estimate. One such technique is optimal fixed point smoothing ([10]). Optimal fixed point smoothing arrives at a state estimate at time t by using all available data in a time frame $0 \leq t \leq T$. In particular, the smoother combines a Forward Kalman Filter up to time t with a backwards iterated Kalman Filter which uses all data from t up to T (see e.g. [16] for smoother equations). After estimation is finished, the model uncertainty Δ_j is stored along with the respective regressor vector σ_j. For each point the training signal based on stored data can be calculated as follows:

$$\varepsilon_j = W^T(t)\sigma_j - \Delta_j \tag{17}$$

The modified update law of equation (15) becomes

$$\dot{W}(t) = -\Gamma\sigma(x)e^T PB - \Gamma\sum_{j=1}^{p}\sigma_j\varepsilon_j^T. \tag{18}$$

If the history stack contains at least $n_2 + 1$ linearly independent data points, the update law in (18) ensures robustness of the closed loop system and convergence of the adaptive parameters to a set of weights, which minimize the functional approximation error. For reasons of brevity a proof is omitted here. For further information refer to [5] and [7].

2.4 Control Allocation

The following section gives an overview of the component of the controller which remains vehicle specific, the control allocation (CA). Every vehicle type is uniquely designed in respect to position, size and number of the control surfaces. The calculation of the actual surface deflections based on the forces and moment requirements is thus treated vehicle dependent. Control allocation has been extensively studied in literature with respect to nominal flight behavior and as active methods to increase fault tolerance ([27]).

In general CA is treated differently depending on the number of actuators in relation to degrees of freedom. For over- and under actuated systems, CA becomes an optimization problem, while for an equal number of independent actuators to degrees of freedom CA falls back to an algebraic relation. In this work an incremental CA was chosen, which is often also referred to as direct CA. Incremental refers to the fact that a global relation between moment or forces to deflection is not available. Rather, a change in moment implies a change of the deflection using linear relation based on the effectiveness matrix B_{eff} ([9]). In this work, force to thrust calculation and moment to deflections of control surfaces is done separately. A changes of the Moments ΔM is hence given by

$$\Delta M = B_{eff}\Delta\delta_{rot}, \tag{19}$$

where $\Delta\delta_{rot}$ is the change in rotational control surfaces. The control input $\delta_r ot$ is calculated using an estimate of the current control input $\delta_{est,rot}$,

$$\delta_{rot} = \delta_{est,rot} + B_{eff}^{-1}\Delta M. \tag{20}$$

An equivalent, scalar relation yields the thrust. For the two fixed-wing UA of this paper (see below for further information) two different control allocation approaches have to be considered.

Prometheus is equipped with pairs of elevators, aileron and rudder surfaces. Both elevators and rudder are deflected symmetrically while ailerons are deflected anti-symmetrically thus resulting in a set of three independent control variables. The resulting effectiveness matrix is quadratic in nature. In contrast, Extreme Star offers a total of 16 control devices which can be actuated independently. However, for the purpose of this paper only eleven control inputs are considered. These include both canards, both ailerons, both flaps, both elevators, the rudder and the throttle for each main wing motor. As a result the control effectiveness matrix is non-quadratic. Optimization methods such as constraint minimization can optionally be applied to find an adequate solution.

3 Numerical Simulation

This section outlines numerical simulations in order to assess the performance of the presented control architecture for different UA. Furthermore, the robustness of the nonlinear controller against disturbances is evaluated.

3.1 Test-Beds and Simulation Environment

The fixed-wing unmanned aircraft *Prometheus* is a development by the German Aerospace Center, Braunschweig, Germany, see Figure 3. The high-wing in pusher configuration has a MTOW of $25\,kg$ and is equipped with a PC104 based avionics system containing flight control computer, vision computer and the common set of sensors required for automated flight. Flight control and guidance algorithm use a

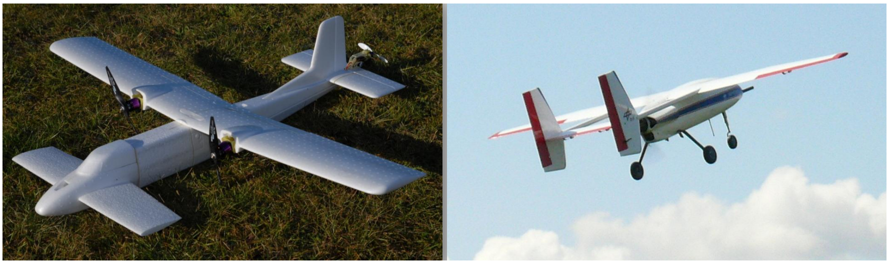


Fig. 3 Extreme Star (left) and Prometheus (right)

QNX operating system while vision application run on a Linux based system. Aims of the project are the evaluation of methods and algorithms of unmanned aircraft. These include path-planning, mission management, flight control, sensor fusion and environmental awareness. An additional goal is the operation of multiple UA of different type within a unique software framework. Especially for the Prometheus project, also fixed-wing specific research from operation perspective, such as aerial refueling, are of interest.

The *FSD ExtremeStar* bases on the off-the-shelf polystyrene model airplane Multiplex TwinStar II and was modified by the AkaModell Munich on behalf of the Institute of Flight System Dynamics of the Technische Universität München. In particular these modifications include the addition of canards with variable incidence, the extension of the fuselage, the conversion of the trailing edge of the inner wing part into flaps and the replacement of the existing motors with higher performance motors with pitch axis thrust vectoring. Furthermore, an additional tiltable third propeller was attached to the tail of the aircraft. An independent control of left and right side control devices is possible, therefore offering a total of 16 actuators. The main purpose of the airplane is to analyze new and existing control methods for an UA with a large number of control devices. For the purpose of this work only eleven control inputs are considered.

Algorithms normally run through two specific simulation stages before going into flight test as depicted in Figure 4. First stage is the software-in-the-loop (SIL) simulation, which is used for development of the algorithms themselves. They are tested in a single computer setup, simulating all aspects of the unmanned aircraft system (UAS). The SIL typically consists of the mission manager, which handles the mission components and provides an interface to the ground control station. The flight controller receives commands of the mission management and the flight dynamics are used to simulate the UA flight mechanics. Optionally, a realistic sensor fusion can be used, which contains sensor emulation as well as an algorithm like a Kalman filter for state estimation ($\hat{x}$). The advantages of the SIL simulation are a fast development cycle, reduced recourse requirements and modular level of abstraction and thus simulation complexity.

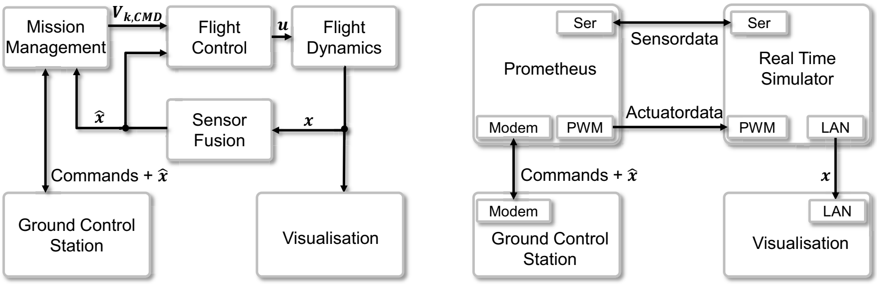


Fig. 4 Software-in-the-loop (left) and hardware-in-the-loop (right) simulations

After integration of the algorithm into the flight software, it is tested within the hardware-in-the-loop (HIL) simulation. For this work a HIL framework was only available for Prometheus. For the simulation setup the complete vehicle is integrated into the simulation including the processing units of the avionics. A real time simulator based on dSpace simulates the flight dynamics as well as sensor emulation. The emulated sensor data is supplied via serial (Ser) connection to the avionics. The UA is controlled via the GCS in the same manner as it is in flight test. In both simulation cases, SIL and HIL, the same visualization is used to give a user feedback of the real state of the UA.

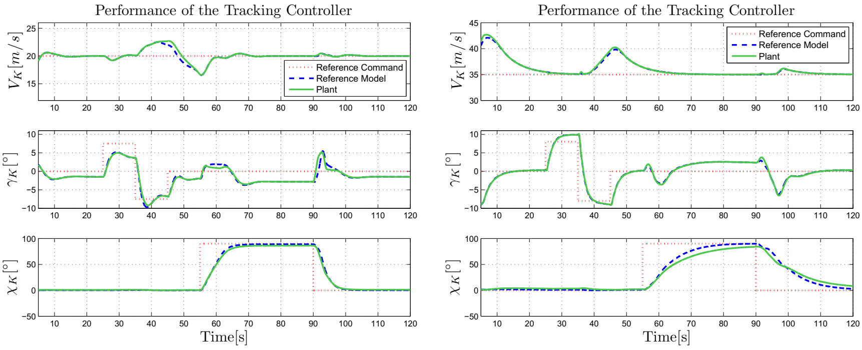
3.2 Test Scenario and Results from Numerical Simulation

At first, the control architecture is tested in the SIL framework. For robustness considerations parameter uncertainties are artificially introduced to the system. In particular, the mass moment of inertia I_{BB} is increased by 15%, the aerodynamic derivative of the lift due to an angle of attack $C_{L,\alpha}$ is decreased by 10% and the mass of the aircraft m is increased by 10%. Additionally, a constant wind from northeast disturbs the aircraft with $V_W = [-5, -2.5, 0]^T \frac{m}{s}$.

During the numerical simulation, the aircraft shall track a series of path commands represented by step inputs. The simulation runs a total of $120s$ with a time step of $0.01s$. The initial conditions for the path states are given with $V_{K,0} = 40\,\frac{m}{s}$, $\gamma_{K,0} = -10^\circ$ and $\chi_{K,0} = 0^\circ$ for FSD ExtremeStar and $V_{K,0} = 40\,\frac{m}{s}$, $\gamma_{K,0} = -10^\circ$ and $\chi_{K,0} = 0^\circ$ for Prometheus. In order to compare the results each aircraft receives the same commands. The reference signals are comprised of several step inputs. After $25\,s$ the climb angle is commanded to be $\gamma_K = 7.5^\circ$ for $10\,s$ and $\gamma_K = -7.5^\circ$ for $10\,s$ after that. After $55\,s$ the aircraft are commanded to perform a 90° right turn followed by a 90° left turn after $90\,s$. During the simulation the velocity is held constant at $V_K = 20\,\frac{m}{s}$ for FSD ExtremeStar and $V_K = 35\,\frac{m}{s}$ for Prometheus, respectively.

Figures 5(a) and 5(b) show the tracking performance of Prometheus and ExtremeStar respectively if no adaptive controller is used in any loop. For Prometheus in Figure 5(a) it can be seen that the plant follows the reference trajectory without major deviations even in the presence of parameter errors and the external disturbance. However, Pseudo Control Hedging in the respective loops alters the reference trajectory of the longitudinal axis in the outer loop such that it deviates from the commanded signal significantly. This is due to the presence of modeling uncertainties in the inner loop and outer loop. In order to achieve steady state accuracy in the longitudinal axis, the effect of modeling uncertainties needs to be reduced, which can be achieved by applying adaptive elements to the system.

Similar to Prometheus, ExtremeStar in Figure 5(b) is able to track the reference model accurately, but PCH prevents the reference model from tracking the commanded signal. In particular, while the performance of the lateral motion increases compared to Prometheus, the longitudinal performance worsens, thus prohibiting a successful operation of the aircraft.



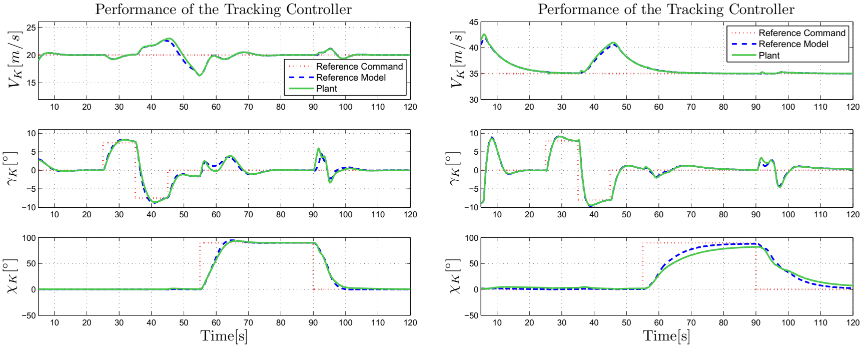
(a) (b)

Fig. 5 Simulation of ExtremeStar (5(a)) and Prometheus (5(b)) in the SIL framework with wind and uncertainties but without adaptive controller

It can be observed that the two aircraft exhibit different performance properties. This deviation is attributed to the fact that both aircraft significantly differ in architecture and parameters, in particular size, mass and number of control surfaces.

In the following the same simulations are performed while adaptive elements are added to the system. Since the relations in the attitude loop are purely kinematic, no model uncertainty are expected here. Hence, adaptive elements are only added to the inner and outer loop. In particular a radial basis function neural network with concurrent learning update laws augments the inner loop, consisting of 125 neurons, which are evenly distributed in the state space. The RBF receives only the rotational rates as inputs. The learning rate is set to $\Gamma = 1.5$. For concurrent learning a total of 130 points are stored in a static history stack. Additionally for testing purposes a neural network with sigmoid activation functions is added to the outer loop, consisting of a total of nine neurons. Note, that in this case also the input weights are updated, thus the neurons do not have to be distributed in the state space a priori. Hence, the number of neurons needed for adaptation is significantly lower than in the case of radial basis functions neural network. The input vector to the outer loop network consists of the path, attitude and rotational states. The learning rates are set to $\Gamma_V = 2$ for the input weights and $\Gamma_W = 0.1$ for the output weights. The latter are chosen to be small in order to prevent the propagation of errors resulting from fast adaptation to the middle loop. For the concurrent learning update law a total of 30 points are stored.

Figures 6(a) and 6(b) show the tracking performance of Prometheus and Extreme Star with adaptive elements. It can be seen that especially the performance in the longitudinal motion drastically increases in both cases. This improvement leads to the conclusion that the neural networks approximate the uncertainty such that its effects on the plant dynamics are significantly reduced. Furthermore, the control



(a) (b)

Fig. 6 Simulation of ExtremeStar (6(a)) and Prometheus (6(b)) in the SIL framework with wind and uncertainties; inner loop augmented by radial basis function neural network, outer loop augmented by single hidden layer neural network with sigmoid activation functions.

architecture is seen to be robust not only against the parameter uncertainties but also against the disturbance by wind.

In addition to the SIL simulation, a HIL framework based on dSpace is available for Prometheus. The simulation runs for a total of $175\,s$ with a time step of $0.01\,s$. The initial values for the desired path are given with $V_{K,0} = 30\,\frac{m}{s}$, $\gamma_{K,0} = 0°$ and $\chi_{K,0} = 0°$. The reference signals are comprised of several step inputs. Prometheus shall slow to $V_K = 20\,\frac{m}{s}$ after $80\,s$ and accelerate to $V_K = 40\,\frac{m}{s}$ after $138\,s$. During the different velocity phases climb anlge and course angle are varied. After $70\,s$ the climb angle is commanded to be $\gamma_K = 7.5°$ for $5\,s$ and $\gamma_K = -7.5°$ for $5\,s$ after that. The same maneuver is initiated after $125\,s$ and $165\,s$. In the second maneuver, the duration of the steps is extended to $8\,s$. Finally, after $41\,s$, $95\,s$ and $150\,s$ the UA is commanded to perform three consecutive 90° right turns.

Figure 7 shows the tracking performance of the nonlinear adaptive path controller for the HIL simulation. It can be seen that the controller is able to follow the desired path if the real Flight Control System Hardware is used. The deviation between the reference model output and the measured states is minor. The jump in the reference model of the course angle after $130\,s$ is attributed to the fact that χ_K is limited to $(-\pi ... \pi]$. However, it can also be seen that for higher velocities the trajectory generated by the reference model deviates from the commanded signal. This is attributed to the fact that the uncertainties in the inner loop and outer loop, especially the generated lift and the gradients in the control allocation, are dependent on the velocity. Concurrent learning is formulated for constant optimal weights. Hence, the change in velocity and consequently a change of the optimal weights lead to a drop in performance. However, the control architecture is still seen to be robust.

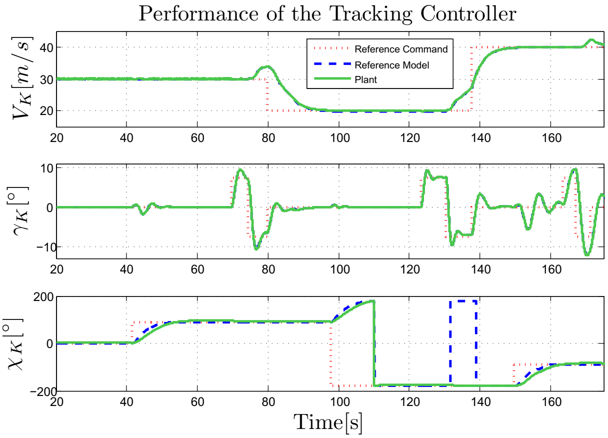


Fig. 7 Hardware-in-the-Loop simulation of Prometheus

4 Conclusion

In this paper we showed how an adaptive trajectory control architecture can be set up in order to be simply applicable to multiple fixed-wing UA. Key part of the controller is a generic inversion part, which feedback linearizes a chosen inversion model. Adaptive elements, in the form of concurrent learning neural networks, significantly decrease the effect of model uncertainties between the plant dynamics and the inversion model, therefore increasing the applicability of the control architecture to a broad spectrum of fixed-wing aircraft. While the former can be simply applied to any fixed-wing UA which shares the same signal interface, the gains of the linear controller have to be selected and a control allocation has to be constructed for each aircraft individually. Still, by generalizing the feedback linearization, the amount of time required to set up a reliable flight control system decreases significantly. Results from numerical simulation showed that the exclusive use of feedback linearization doesn't result in acceptable performance. Hence, adaptive elements have to be used in order to decrease the effects of the uncertainty. By using concurrent learning adaptive control the adaptive parameters are driven to a set of weights which minimize the approximation error, thus increasing the performance significantly compared to instantaneous learning laws. Further improvement of the proposed control architecture can be achieved by e.g. finding methods to generalize the construction of the control allocation. Furthermore, the performance and limitations of the proposed architecture need to be evaluated in flight tests.

Acknowledgements. The authors gratefully acknowledge the support of both, the Unmanned Aircraft Department of DLR and the Institute of Flight System Dynamics at the Technische Universität München. For the valuable discussions we especially thank Fabian Klüssendorf, Leonhard Höcht and Thomas Bierling.

References

1. Åström, K.J., Wittenmark, B.: Adaptive control, 2nd edn. Dover Publications, Mineola (2008)
2. Bierling, T., Höcht, L., Holzapfel, F., Maier, R., Wildschek, A.: Comparative analysis of mrac architectures in a unified framework. In: AIAA Guidance, Navigation, and Control Conference, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics (2010)
3. Brockhaus, R., Alles, W., Luckner, R.: Flugregelung, 3rd edn. Springer, Berlin (2010)
4. Calise, A.J., Rysdyk, R.T.: Nonlinear adaptive flight control using neural networks. IEEE Control Systems Magazine 18(6), 14–25 (1998)
5. Chowdhary, G.: Concurrent Learning for convergence in Adaptive Control without Persistency of Excitation. PhD thesis, Georgia Insitute of Technology, Atlanta and GA (2010)
6. Chowdhary, G., Jategaonkar, R.: Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. Aerospace Science and Technology 14(2), 106–117 (2010)
7. Chowdhary, G., Johnson, E.: Flight test validation of a neural network based long term learning adaptive flight controller. In: AIAA Guidance, Navigation, and Control Conference. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics (2009)
8. Chowdhary, G., Johnson, E.N.: A singular value maximizing data recording algorithm for concurrent learning. In: American Control Conference, San Francisco, CA (June 2011)
9. Edwards, C., Lombaerts, T., Smaili, H.: Fault tolerant flight control: A benchmark challenge. Springer, Berlin (2010)
10. Gelb, A.: Applied optimal estimation. M.I.T. Press, Cambridge (1974)
11. Holzapfel, F.: Nichtlineare adaptive Regelung eines unbemannten Fluggerätes. PhD thesis, Technische Universität München, Munich (2004)
12. Ioannou, P.A., Kokotovic, P.V.: Instability analysis and improvement of robustness of adaptive control. Automatica 20(5), 583–594 (1984)
13. Johnson, E.N.: Limited Authority Adaptive Flight Control. PhD thesis, Georgia Insitute of Technology, Atlanta and GA (2000)
14. Lewis, F.L.: Nonlinear network structures for feedback control. Asian Journal of Control 1(4), 205–228 (1999)
15. Monahemi, M.M., Krstic, M.: Control of wing rock motion using adaptive feedback linearization. Journal of Guidance, Control, and Dynamics 19(4), 905–912 (1996)
16. Mühlegg, M., Johnson, E., Chowdhary, G.: Concurrent learning adaptive control of linear systems with noisy measurements. In: AIAA Guidance, Navigation, and Control Conference, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics (2012)
17. Narendra, K., Annaswamy, A.: A new adaptive law for robust adaptation without persistent excitation. IEEE Transactions on Automatic Control 32(2), 134–145 (1987)
18. Narendra, K.S., Annaswamy, A.M.: Stable adaptive systems. Dover Publications, Mineola (2005)
19. Nguyen, N.T., Jacklin, S.A.: Neural net adaptive flight control stability. In: Verification and Validation Challenges, and Future Research, IJCNN Conference (2007)
20. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation 3(2), 246–257 (1991)

21. Sanner, R.M., Slotine, J.-J.E.: Gaussian networks for direct adaptive control. IEEE Transactions on Neural Networks 3(6), 837–863 (1992)
22. Seanor, B.A.: Flight Testing of a Remotely Piloted Vehicle for Aircraft Parameter Estimation Purposes. PhD thesis, West Virginia University, Morgantown and West Virginia (2002)
23. Shankar, P., Yedavalli, R.K., Burken, J.J.: Self-organizing radial basis function networks for adaptive flight control. Journal of Guidance, Control, and Dynamics 34(3), 783–794 (2011)
24. Cong, S., Song, R.: An improved b-spline fuzzy-neural network controller, pp. 1713–1717
25. Stevens, B.L., Lewis, F.L.: Aircraft control and simulation, 2nd edn. J. Wiley, Hoboken (2003)
26. Tao, G.: Adaptive control design and analysis. Wiley-Interscience, Hoboken (2003)
27. Verhaegen, M., Kanev, S., Hallouzi, R., Jones, C., Maciejowski, J., Smail, H.: Fault tolerant flight control - a survey. In: Morari, M., Thoma, M., Edwards, C., Lombaerts, T., Smaili, H. (eds.). LNCIS, pp. 47–89. Springer, Heidelberg (2010)
28. Volyanskyy, K.Y., Haddad, W.M., Calise, A.J.: A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond sigma - and e-modifications. IEEE Transactions on Neural Networks 20(11), 1707–1723 (2009)

Stereo Vision Based Obstacle Avoidance on Flapping Wing MAVs

Sjoerd Tijmons, Guido de Croon, Bart Remes, Christophe De Wagter,
Rick Ruijsink, Erik-Jan van Kampen, and Qiping Chu

Abstract. One of the major challenges in robotics is to develop a fly-like robot that can autonomously fly around in unknown environments. State-of-the-art research on autonomous flight of light-weight flapping wing MAVs uses information such as optic flow and appearance variation extracted from a single camera, and has met with limited success. This paper presents the first study of stereo vision for onboard obstacle detection. Stereo vision provides instantaneous distance estimates making the method less dependent than single camera methods on the camera motions resulting from the flapping. After hardware modifications specifically tuned to use on a flapping wing MAV, the computationally efficient Semi-Global Matching (SGM) algorithm in combination with off-board processing allows for accurate real-time distance estimation. Closed-loop indoor experiments with the flapping wing MAV DelFly II demonstrate the advantage of this technique over the use of optic flow measurements.

1 Introduction

Autonomous flight of flapping wing MAVs (FWMAVs) is a considerable challenge. The main reason for this is that their light weight prevents the use of heavy and energy-consuming laser scanners that are successful on heavier MAVs such as quad rotors [1] [2]. Still, there have been several attempts at achieving autonomous flight with FWMAVs. Hines at al. [3] describes an FWMAV design that is currently not able to fly on its own, but experiments show it is able to control its pitch and roll angle by using actuators that change the wing shape. Lin et al. [4] shows the altitude

Sjoerd Tijmons · Guido de Croon · Bart Remes · Christophe De Wagter ·
Rick Ruijsink · Erik-Jan van Kampen · Qiping Chu
Control and Operations Division, Faculty of Aerospace Engineering,
Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
e-mail: `s.tijmons@tudelft.nl`,
`{guido.de.croon,microuav}@gmail.com`

control of the 10 gram FWMAV called *Golden Snitch*. No onboard processing or sensing is used for this task. Using an external stereo camera the position of the vehicle is determined, and further control is performed by a ground station. Duhamel et al. [5] presents an experiment with a 101 milligram flapping wing microrobot called *RoboBee*. Using an onboard optic flow sensor and a well textured screen, the altitude is successfully controlled offboard in a closed-loop experiment, with only small oscillations and a slight drift. Baek et al. [6] performs closed-loop altitude control on a 12 gram ornithopter by using an external camera. In a follow up on this research [7], a 13 gram ornithopter is presented that is able to fly autonomously to a target, using an onboard infrared sensor for target tracking and 3-axis gyroscopes for attitude estimation. During 20 trials a success rate of 85% is reached. Garcia Bermudez et al. [8] performes optic flow measurements on a 7 gram ornithopter. Heavily down-sampled onboard camera images are stored onboard during flight, and uploaded to a computer afterwards to compute optic flow. The main finding is a strong coupling between body motion and the sensed optic flow. Tedrake et al.[9] shows autonomous flight of an ornithopter with a 2-meter wingspan. Only pitch control has been tested successfully using an IMU.

With DelFly II several autonomy experiments have been performed dealing with various control tasks [10]. These tests range from height control with an external camera to height control and path following with an onboard camera and offboard processing. Also a novel appearance cue for obstacle avoidance is introduced [11] [12]. It is based on the principle that when an object is approached, its colors and detailed texture become more and more visible, while other objects move out of sight. It is shown that this cue is a useful complement to optic flow for detecting obstacles with the DelFly.

This experiment showed that optic flow is still not sufficient to perform obstacle avoidance on FWMAVs. To perform good optic flow measurements the camera images should be noiseless and rotation rates should be known, requiring three gyroscopes that can measure the rotational speeds of the vehicle. Measurements should be performed onboard, but the amount of onboard processing power is currently too limited. Therefore the video signal is sent to a ground station, which implies a low frame rate. The frame rate of 30 FPS and line-by-line recording of the camera result in large image distortions that affect the optic flow quality.

In this paper the use of stereo vision is proposed to circumvent these problems. For optic flow image sequences are used, while stereo vision uses images taken at the same time. Vehicle motion has therefore a smaller influence on the quality of the measurements and the video frame rate is of no importance on the quality of individual measurements. Furthermore, it gives an instantaneous overview of obstacles in sight of the camera.

In Section 2 a description is given of the DelFly system including stereo cameras and ground station. Section 3 discusses stereo vision and the algorithm used in this study. The performance of the stereo vision system is presented in Section 4. Closed-loop autonomy experiments and their results are discussed in Section 5. Finally a summary of the conclusions is given in Section 6.

2 System Design

Since the research in this study focused on FWMAVs, tests were performed with the DelFly II. Its design is shown in Figure 1. The most defining feature of the DelFly is that there is always a camera and transmitter onboard (in this study two cameras). The current version of DelFly II is also equipped with gyrometers, a pressure meter, and onboard processing for these high-frequency measurements. Additional defining features are its biplane wing model and its tail. For more details, the interested reader is referred to [12]. Figure 2 shows an overview of all system components and their interactions.

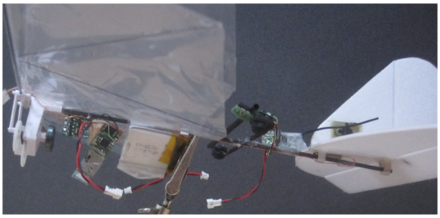


Fig. 1 Side-view of DelFy II including stereo vision cameras

For communication with the ground station a Bluetooth transceiver is used. This system operates at the same frequency as the NTSC transmitter of the stereo system: 2.4GHz. Wi-Fi networks normally operate around this frequency as well. As a result the images received on the ground can become noisy, as is illustrated in figure 3. The ground station is a 2.30GHz dual-core system running on Windows 7. The system is prone to several types of delay. It takes around 60ms to receive the stereo images on the ground. Processing is then performed in real-time (40 ms) and control signals are then send via Bluetooth. This is the slowest step, which at least takes around 60ms. However, because of interference from the other systems operating around the same frequency, this delay varies over time and can become more than 200ms in some cases.

The main feature of interest is the stereo vision camera, which will be discussed in more detail. Due to the stereo camera system the weight of the DelFly in this

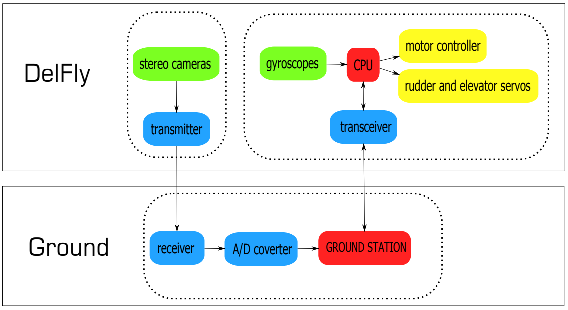


Fig. 2 Diagram of the interaction between all system components

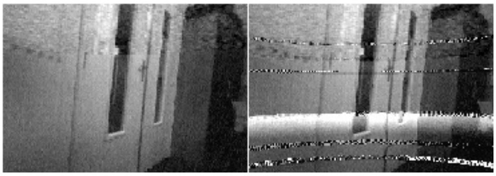


Fig. 3 Example of noise due to an interfering source. Left is noise free, right contains severe noise

current configuration is more than usual. Normally the total weight including sensors and batteries is under 17 gram. However, the stereo vision system, including a separate battery, accounts for 5.2 gram. The total weight of the DelFly in this configuration is 21.1 gram.

The selected configuration of the DelFly for this study is for slow-forward flight because of the purpose of indoor obstacle avoidance. In this configuration the speed can still be increased to several meters per second, but it can also fly stable with only 0.6m/s. Hovering is not possible due to the heavy weight of the configuration. The speed is controlled by the tail elevator. The rudder can be used to make turns. The turn speed can be controlled accurately with a servo. However, there is variation in the response of the DelFly to a rudder input. The turns are therefore not strictly circular. Furthermore, giving too much rudder input will result in a fast spiral motion. Still, with sufficient rudder input the turn diameter is less than 1m.

2.1 Stereo Vision Camera

The stereo camera system is the main sensor on the DelFly. Its components can be seen in more in detail in figure 4. The setup consists of two synchronized CMOS 720x240 cameras (with an offset of 7.6 cm) running at 25 Hz and a 2.4 GHz NTSC transmitter. The cameras have a field of view of ±60 degrees horizontally. Because there is only one transmitter, the video streams from both cameras have to be combined as one. In the initial setup this was done as follows: an NTSC frame consist of an even field and an odd field. To combine two synchronized NTSC cameras, the even lines of the first camera are scanned first, and then the camera source is switched and the uneven lines of the second camera are scanned. This image-based scheme results in frames which consist of image lines from the left and right camera alternately. The resulting frame size is still the same (720x480) but the resolution for each camera has now been reduced to 720x240 pixels.

During early tests with the camera system a shortcoming of this setup was noticed. The result from the stereo matching process was strongly affected by the motion of the camera. During static tests the results were promising and proved to be reliable, but during motion the results would become distorted. Since all even lines are scanned before the uneven lines in this image-based scheme, there is a time difference between the scan lines from the left and the right image. The first line of the 'transmitted' image comes from the right camera, the second line comes from the left camera. When the camera is at rest, it can be roughly assumed that these two lines are observing the same features. When the camera is in motion, this

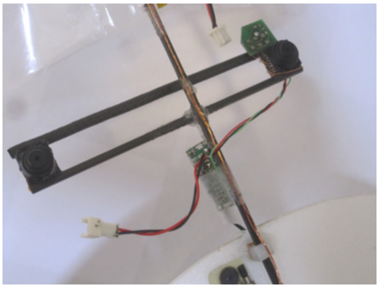


Fig. 4 Stereo camera system. The base line of the cameras is 7.6cm.

assumption does not hold anymore because of the time difference of approximately 20ms (half the time between two frames) between the lines. During this time the cameras might have changed orientation and the left and right image lines cover different view directions. As a result, the output from the stereo matching process becomes distorted.

The hardware of the camera system was changed such that each time after a scan line has been scanned, the system switches to the other camera. As a result of this line-based scheme the frames sent by the transmitter now consist of two sets of two images that have been taken at different times. This is illustrated in figure 5. Two images (one from the left camera and one from the right camera) are captured on the even lines first (light colors), and after that another set of stereo images is captured on the uneven lines (dark colors). The images on the uneven lines are always the most recent stereo images, and these are used for stereo processing. Each individual image now has a resolution of 720x120 pixels. The benefit of this approach is that the time difference between the stereo images has been reduced significantly. Instead of switching between cameras after 240 lines have been scanned, switching is now done after each single scanline.

So by changing the hardware synchronization from an image-based scheme to a line-based scheme, the time difference has been reduced with a factor 1/240 to roughly 83μs. For the purpose of stereo matching it is assumed that two consecutive uneven image lines (which always contain image lines from both cameras) cover the same image areas.

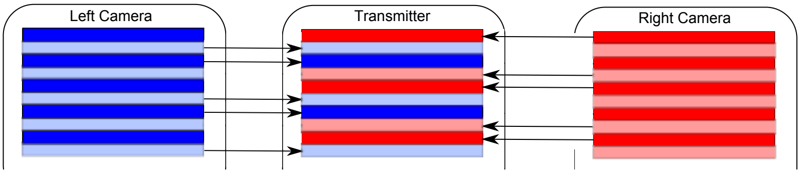


Fig. 5 Line-based synchronization scheme designed for FWMAV stereo vision. The 'transmitted' image consists of image lines from the left and right camera's. The even lines (light) are scanned first and consist of image lines from the left (blue) and right (red) camera alternately. It takes about 83μs to scan one image line. After all even lines have been scanned, the uneven lines (dark) are scanned from the left and right camera alternately.

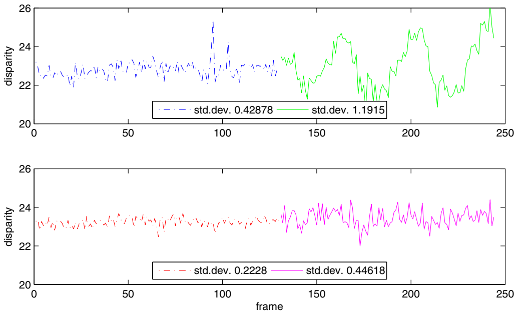


Fig. 6 Comparison between the camera reading methods. **Top** initial method **Bottom** implemented method. During the first 135 frames there is no motion (dash-dotted lines), further on there is a relative motion between the camera and the chessboard (solid lines).

The impact of this modification is shown in figure 6. A small test was performed where the stereo camera setup was positioned at a fixed height above a large chessboard (to assure texture). A record was made of the camera stream while during the first few seconds the scene was static. After a few seconds, the chessboard was slid back and forth (left-right in the camera view) to introduce motion. The disparity was then computed for both types of camera implementations to see the effect of motion on the output. From the figure it is clear that the 'initial' system (top plot) performs significantly worse as soon as the scene starts to move. From the data one can see the left-right motion of the chessboard. When the chessboard slides to the left, the images appear to move towards each other. Hence a smaller disparity is measured. When sliding in the other direction, larger disparities are measured. From the top plot it can be seen that this motion is not visible from the measurements. But it should be noted that the measurements show smaller deviations during the first seconds of the experiment when there was no motion.

In this setup the effective resolution is reduced to a quarter of the original resolution. However, this is not an issue since the images are sub-sampled to a resolution of 160x108 to perform stereo processing at 25Hz. As noted before the camera images can be subject to noise. Furthermore, in the current setup both cameras make use of the same intensity calibration parameters, which only apply to one of them. As a result there is a major difference in sensitivity to bright image features. The cameras are also very sensitive to direct and reflected sunlight. This can blind the cameras. Also high frequency light sources can have a disturbing effect.

3 Stereo Vision Methods

Computer stereo vision is the extraction of 3D information from digital images. In general this implies that images from two or more cameras are evaluated by an algorithm that tries to compute which pixels correspond to the same physical object. When this matching is done, it is known for each pixel how large it is shifted in other images. By knowing the characteristics of the cameras, these shifts (denoted as 'disparities'), can be converted to real xyz-coordinates. By using all image pixels together a 3D reconstruction of the scene can be obtained.

A considerable amount of research has been done for decades on the problem of computational stereo vision. This research is still ongoing with focuses on quality and computational efficiency. These are conflicting aspects. A concise overview of computer stereo vision methods that have been developed over the years is beyond the scope of this paper. Interested readers are referred to the Middlebury taxonomy of Scharstein and Szeliski [13] and the evaluation of Tombari et al. [14] for overview articles. For stereo vision on a flapping wing MAV, the main requirement is implementability in real time systems. Real-time performance can be obtained in two ways: by using efficient algorithms or by using special hardware implementations. In this study the focus lies on efficient algorithms. Using for example a Graphical Processing Unit(GPU), Field Programmable Gate Array (FPGA), Application-Specific Integrated Circuit (ASIC) or Digital Signal Processor (DSP) allows the use of optimized computation strategies that are very specific and have a limited applicability. Since the aim of this study is to converge to full autonomy, onboard processing is also a topic of interest. It is believed that if algorithms cannot be implemented on a CPU in real-time, they will also be no candidate for on board processing in future systems. The focus in this study is therefore further limited to methods that enable real-time performance on CPUs.

Comparison

Stereo vision algorithms can be divided in four groups depending on the optimization strategy they are based on: Winner-Takes-All, One-Dimensional Optimization, Multi-Dimensional Optimization and Global Optimization. Figure 7 shows a comparison between the these types of optimization. Global Optimization is left out of this comparison because of its computational complexity. From each of the other three types an example from the OpenCV library was taken to demonstrate the most important differences. The figure shows the result of each type of optimization method for the same image. The stereo images were sub-sampled such that each method had real-time performance. The parameters were tuned to obtain the best result.

The Block Matching method shows a relatively sparse result. Dominant features, such as vertical lines, are matched quite well, but in between these features a lot

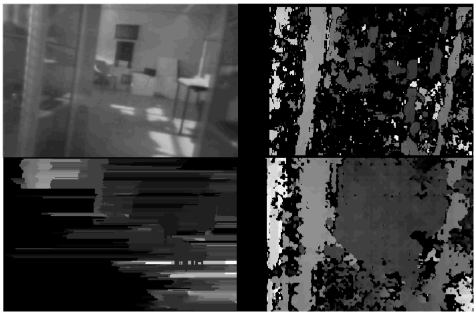


Fig. 7 Comparison of three different types of stereo vision methods. **Top-Left** test image **Top-Right** Block Matching (Winner-Takes-All) **Bottom-Left** Dynamic Programming (One-Dimensional Optimization) **Bottom-Right** Semi-Global Block Matching (Multi-Dimensional Optimization)

of unknown regions are left empty (black pixels). Even the shadows on the ground apparently do not provide enough texture for good matching. The information from this method is partly useful, in that it provides information on obstacles close by. But this information would be much more useful if the method would be able to indicate that the center zone of the image contains only obstacles far away. Note that the center zone even contains blobs of white pixels that indicate non-existing close objects.

The Dynamic Programming method performs even worse. The main structures in the image can not even be distinguished. This result might not be fully representative for dynamic programming algorithms since these perform better than winner-takes-all methods in general. In the top-left corner of the image the streaking effect is visible: the image lines appear as if they are a little bit randomly shifted horizontally (typical effect of Dynamic Programming). The bottom-left part of the image is almost empty (no reliable matches) and the right part of the image does not show clear objects. This illustrates the short-coming of Dynamic Programming: matching errors influence the results for the remainder of the image lines. The bad matching results in the left part of the image spoil the results in the right part of the image. The fact that this implementation uses pixel-to-pixel matching costs might have a negative influence of the final result.

Compared to the other two methods, Semi-Global Block Matching gives significantly better results. The main structure of the scene is clearly visible in the disparity map: two cabinets close by on both sides and in between there is space with obstacles much further away. Also here some regions are left empty but the amount of known disparities is substantially larger. False matches are also visible but their number is also small. This method gives the most useful information, and is potentially useful enough for obstacle detection. The result is also notable because the method relies on simple pixel-to-pixel matching costs.

According to literature Semi-Global Matching represents a good trade-off between computational efficiency and performance [15] [16] [17].

Based on the findings from literature and the above results that support these findings it was decided to use the Semi-Global Matching [18] method for implementation in the obstacle avoidance strategies that were developed and tested in this study.

4 System Performance

The performance characteristics in terms of distance measurements accuracy are discussed in this section. These are based on static and flight tests.

4.1 Static Accuracy Measurements

An important performance measure for the stereo vision system is its accuracy of measuring distances to objects. To measure its actual performance without the influence of platform vibrations, a static test was done. For this test the camera was fixed at several distances (100,150,200,250,300,400,500 and 600cm) from a screen. The screen was a chess mat that was hanging vertically in the field of view of the camera. The stereo vision system was used in the same way as it is during flight. Disparity maps were computed from 1100 frames per measurement point. From each disparity map a small patch of 10x10 pixels was taken from the center of the map to compute the mean disparity. This disparity was used for calculating the distance from the camera to the screen. The results are shown in figure 8. From the results it can be observed that, at least for the static case, the stereo camera system is capable of measuring the distance to obstacles up to 500cm with a mean error of less than 50cm. For the task of obstacle avoidance this can be regarded as an acceptable performance. Obstacles that are even farther away will be detected with a lower distance accuracy. The mean error is larger than 140cm in these cases.

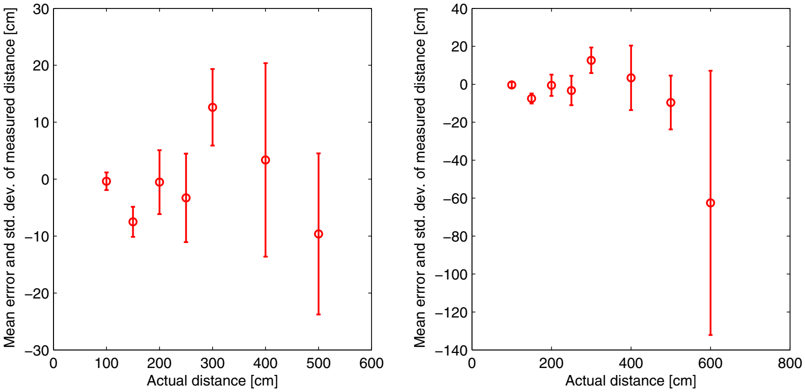


Fig. 8 Distance measurement accuracy for the static case. The left plot is a detailed version of the right plot.

4.2 Accuracy Measurements during Flight

The accuracy of the stereo vision system has also been measured in flight. The experiment was performed using a free-flying DelFly at a speed of approximately 60cm/s. The DelFly was flying in the direction of the chess mat. Two external cameras were used to track the position of the DelFly. Tracking was performed as follows: two video cameras were positioned such that the chess mat would be in their field of view and also the area in front of the chess mat (around 5m). The cameras were positioned on both sides of the flight path of the DelFly. By using a powerful background subtraction routine [19] and blob tracking, a special small marker positioned under the DelFly could be tracked. By using triangulation routines from OpenCV, the three-dimensional flight path (w.r.t the chess mat) of the DelFly was determined. The measurements from the onboard camera and the external cameras were synchronized by looking for specific features in the recorded videos.

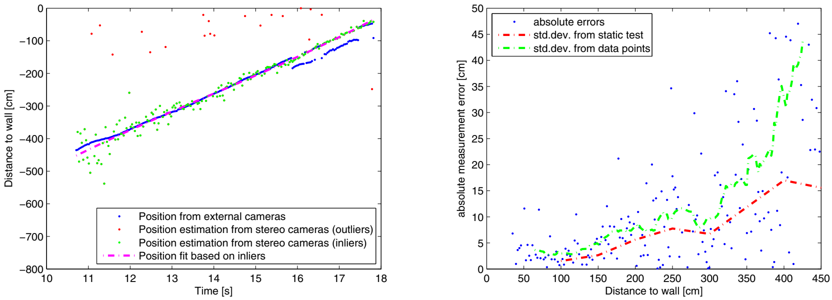


Fig. 9 Distance measurement accuracy for the flight test. The left plot shows the actual distance and estimated distance over time. The right plot shows the estimation error with reference to the actual distance.

Figure 9 shows the result from the first flight test. The blue points in the left plot indicate the distance between the DelFly and the mat, based on measurements from the external cameras. At small distances the blue points show some discontinuities. This is a result from the background subtraction. At small distances the DelFly flies between the cameras and the mat. The white marker on the DelFly will at some points not be noticed when it is in front of a white chess board field. The tracking routine will then find another point on the DelFly, leading to triangulation errors. These measurement errors should therefore be ignored.

The red and green dots are onboard distance measurements. As can be seen from the plot, most of these points are concentrated around the blue points. However, some very clear outliers (red dots) are visible. These measurements result from a hardware problem. In some cases the video frames received by the ground station are mixed-up. The order of the scan-lines is then different from the normal case and the left-right images going to the stereo processing routine contain wrong combinations: two images from the same camera, or swapped left-right images. This results in corrupt disparity maps. Another problem is a typical haze effect which results in images that are a mixture of two images.

These bad results (red dots) were left out by detecting and omitting corrupt frames. The curve fit is based on the good measurements (green dots). The right plot in the figure shows the deviation of the measurement points based on the curve fit. A running average (green dashed line) was computed based on the average error with a windows size of 21. Also the standard deviation for the static case is shown in the figure for comparison.

From the left plot it can be observed that the tracked distances and the measured distances show a very good correspondence. The main observation from the right plot is that the onboard measurements have a larger standard deviation than those obtained during the static test. For distances larger than 350 cm the error seems to grow rapidly, but this at a moment that the DelFly is still turning towards the mat.

5 Obstacle Avoidance

This section discusses the results from tests with two different obstacle avoidance strategies.

5.0.1 Direct Yaw Control

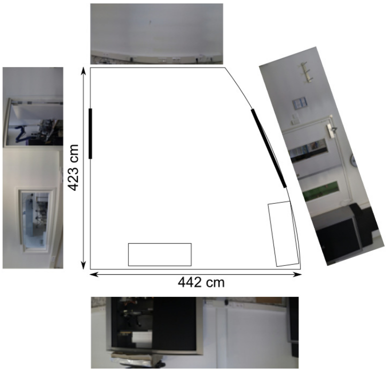
The turn logic for this strategy is straightforward. From the disparity map obtained by the stereo vision algorithm it is computed how many pixels belong to obstacles that are on short range (less than 1.1m). These pixels are summed separately for the left and right halves of the image, forming so-called 'obstacle-signals'. If the left obstacle-signal reaches a threshold a turn to the right is initiated, and vice versa. If both obstacle-signals reach the threshold at the same time, a right turn is initiated. The threshold value has been chosen such that image noise and computational errors do not induce unnecessary turns. The turn is initiated by giving a predefined step input to the rudder. This rudder input is a fixed value that can be set separately for left and right turns. Its value was chosen such that the turns are steady and symmetrical (around 40 cm radius) and the turn speed is not too fast to avoid spiral motions. The turn will end only as soon as both obstacle-signals become lower than another (and smaller) threshold. As soon as the lower threshold has been reached, the rudder will go back to its trim position. However, if one of the obstacle-signals reaches the higher threshold again within a predefined safety-time, the DelFly will continue its previous turn, regardless of which of the two 'obstacle'-signals reached the threshold. This will prevent the DelFly to turn back into the direction it just turned away from, since this is most likely not a safe maneuver.

The experiment was conducted in a room of ∼4.23x4.42m. Figure 10 shows a floor plan of the room. The images on the sides give a good impression of its appearance. Except for the walls the main obstacles are two black cabinets. The door on the left was closed during the experiments, and part of the window on the left was covered to prevent window collisions. It should be noted that the images in the figure only show a part of the scene (mainly the top part) while the onboard cameras of the DelFly could see more of the lower parts of the room. The lights were most of the time switched off during the experiments since they resulted in a flickering effect in the stereo cameras. During the experiments the 'obstacle'-signals were logged, as well as turn events. Furthermore, an onboard image was captured at the moment a turn event (left/right turn or end of turn) occurred. The elevator was given a constant input such that the speed would be around 0.6m/s during the test.

This experiment was repeated several times and resulted in various observations. As a general result it can be stated that the obstacle detection performed well. The obstacle avoidance strategy showed some expected flaws. This will be illustrated by data recorded during one of the flights.

Figure 11 shows a situation during the first seconds of one of the test flights. The sketch on the right indicates the position of the DelFly at the start of the flight and during the first turn. During the first seconds, it flies close to the wall. But, as can be seen in the left onboard image, the wall on the right is outside the field of view.

Fig. 10 Floor plan of the test room. The images around show the walls, doors and cabinets in the room.



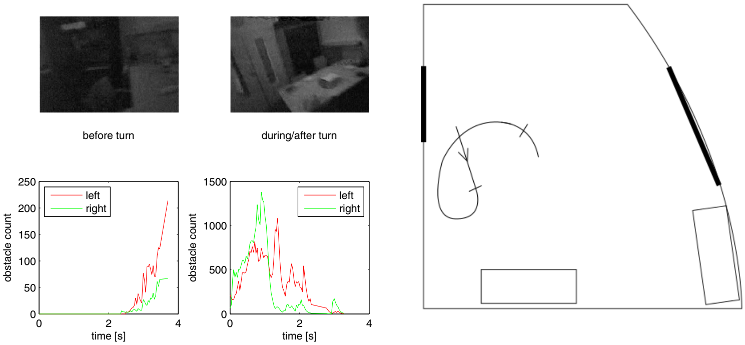


Fig. 11 Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn. For dimensions, see figure 11.

The left bottom plot shows the ’obstacle’-signals during the last seconds before the turn. In this case these values are initially zero because the obstacle detection was not activated yet. From the plot it can be seen that the cabinet, (mainly) on the left side in the image, lets the left obstacle signal increase faster than the right signal, as expected. When the left signal exceeds the threshold (in this experiment set at 200), a turn to the right is initiated successfully. As a result, the DelFly turned into the direction of the wall. It was prevented from colliding manually.

The middle bottom plot shows the amount of obstacles detected during the turn. It can be seen that during the first half second of the turn, the amount of right ’obstacles’ increases first. This is because the wall now enters the field of view. After one second the DelFly has turned around and the right obstacle signal decreases. Since the wall is now in the left side of the view, the left obstacle signal is now very high. While turning away from the wall, the left obstacle signal decreases. It can be observed that it takes fairly long before a safe flight direction was found during the turn. First it takes two seconds before the left obstacle signal decreases below the threshold (set at 50). If this threshold would have been the same (also 200) the turn would have been ended approximately one second earlier. However, earlier experiments showed that for a threshold value of 200, turns would very frequently be ended too early (and then continued immediately, but with some delay). Also note that at the end of the turn, the left obstacle signal decreases below 50, but at the same time the right signal increases again. From the right onboard image in figure 11 it can be observed that the DelFly rolls while making a turn. The table in the image appears to be shifted up in the right side of the image. Apparently it is then detected as an obstacle.

In the sketch on the right it is indicated at which points the turn was initiated and ended. The end point corresponds to the location where the right onboard image (see figure) was taken. In the sketch it is indicated that after this moment the DelFly continued its turn a bit longer. This is a result from the delay between the ground station and the DelFly.

Figure 12 shows how the DelFly continued after the first turn. It is flying into the direction of the same cabinet as before, but now it is in the right side of the camera field of view (top left image). The left bottom plot indicates that indeed an obstacle is detected on the right side. During the first 0.2s after turn initiation (middle bottom plot), the 'obstacle'-signals increase quickly since the DelFly approaches the cabinet. Then, after some delay, the turn command is received onboard and the DelFly starts to turn to the left. Note that around one second later, both signals drop quickly. However, it takes another second before the signals drop below the threshold value. Apparently this is caused by the other cabinet in the corner. It should be noted that during this turn significant noise occurred. As a result, no obstacle detection was performed between 1.23s and 1.66s after turn initiation. This is also the case in the left bottom plot. In that case there are no measurements between 0.77s and 1.0s after the previous turn.

These examples show that the DelFly successfully detects obstacles in its field of view at sufficient range to perform obstacle avoidance. Also during the turns the obstacle detection provides reliable information which makes it possible to decide at which point the turn can be ended safely.

Situations as described in the first example can occur because of the direct nature of the turn strategy in combination with the limited field-of-view of the stereo cameras. During some of the experiments these situations occurred rarely and the DelFly could fly autonomously for longer than 1 minute.

An important observation during the tests is the endurance of the DelFly in its current configuration. As discussed earlier, almost full throttle needs to be applied right from the start of the flight. Within one minute, full throttle is required. Within 2-3 minutes the batteries cannot deliver sufficient power to keep the DelFly at a constant height anymore.

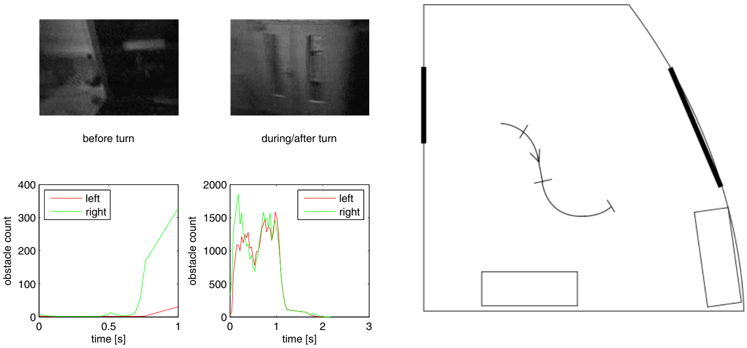


Fig. 12 Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn. For dimensions, see figure 11.

5.0.2 Direct Yaw and Pitch Control

The second experiment is a follow-up of the first one, and it was done the same way. The only difference is the addition of a simple pitch control rule. During unobstructed flight, the elevator is in its fixed position such that the DelFly will fly at a speed of around 0.6m/s. As soon as an obstacle needs to be avoided, a turn is initiated the same way as in the first experiment. At the same time the elevator input is changed such that the DelFly will loose its speed and start to hover. As a result the DelFly will change its heading (by yawing) while it keeps its position. Obstacles can be avoided without the risk of making a turn and colliding with another object out of the camera field of view.

Before this test was conducted, it was already known that the DelFly in its current configuration is too heavy for hovering. It will definitely loose height at the turning points. However, the experiment can be useful in demonstrating that this simple avoidance strategy is suitable for an (FW)MAV as long as it is able to hover. Future designs of the DelFly might be able to hover more efficiently and could use this strategy for maneuvering in small spaces.

As explained this second avoidance strategy is meant to demonstrate the benefit of making turns without forward speed. An example situation is shown in figure 6. The DelFly approaches the cabinet and at some point a turn is initiated. From the bottom middle plot it can be seen that initially the amount of left detected obstacles increases because of control delay and initial forward speed. The DelFly turns to the right and the obstacle-signals decrease. From the right onboard image (top middle image) it was observed that the DelFly has lost some height and is now flying slightly above table height. As discussed earlier, this is an expected (but unwanted) result due to the bad hover performance of this specific configuration of the DelFly.

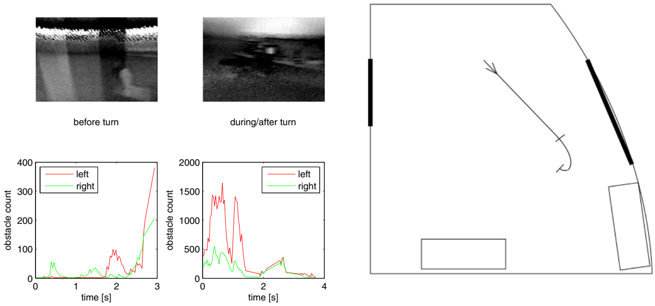


Fig. 13 Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn. For dimensions, see figure 11.

Furthermore this results in the additional problem that there are now other obstacles detected, such as those that are on the tables. Also partly because of noise (and bad obstacle detection measurements for that reason) the turn is ended at a point much further than one would expect.

It was observed that with this strategy obstacle detection and avoidance could be performed successfully without the problem of colliding with out-of-sight obstacles. This demonstrates the advantage of stereo vision over optic flow measurements.

5.0.3 Look-Ahead Yaw Control

The previous experiments demonstrated that the DelFly is not able to hover. Furthermore it was demonstrated that responding directly to obstacles in the field-of-view will result in collisions with obstacles that are outside the field-of-view. Therefore a third turn strategy is discussed here.

In this new strategy the DelFly continuously flies with a constant speed (fixed elevator setting). A turn is initiated when too many obstacles (pixels with a large disparity value) are detected in the safety region. The safety region is defined such that it covers an area large enough for the DelFly to turn around 360 degrees. Because of the limited field-of-view of the camera, this turn area will lie ahead of the current position of the DelFly. Figure 14 shows this safety region. The region is defined in the camera reference frame, with the x-direction positive to the right, the y-direction positive up and the z-direction positive in the direction of flight. Starting at the origin (position of the camera), two oblique lines define the camera-field of view. The dashed line is the trajectory the DelFly will follow as soon as too many obstacles are detected. After 225cm a right turn will be initiated. During the turn the same safety region is used to detect a new safe flight direction. As soon as it is found, the turn will be terminated. Because a turn might be terminated by mistake

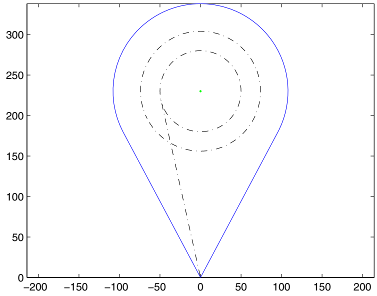


Fig. 14 Turn strategy using continuous turns. The dashed line is the DelFly flight trajectory. The area between the blue line is the obstacle-free region.

or an overshoot can occur due to delays, the turn will be continued immediately if the new direction of flight is not regarded safe anymore. This is only possible within a fraction of a second after turn termination. This type of turn recovery has been taken into account in the safety region definition. This is why the outer circle has been drawn. Around this outer circle extra safety margin has been included to accommodate for the width of the DelFly and inaccuracies in range estimations. Note that the turn area has been centered in the image in order to minimize the size of the safety region. As a result, the flight trajectory towards the turn area is drawn as a slanted line. For this reason the stereo vision cameras were mounted on the DelFly with an offset angle to align the drawn flight trajectory with the flight direction of the DelFly. In other words, the cameras are pointed a little bit to the right.

In this strategy only rudder commands are used. Because the obstacle measurements are sensitive to noise, filtering is required to increase robustness. For this reason a logical diagram was developed as shown in figure 15 (left) which decides upon rudder inputs. In this logic each turn is divided in phases. During Phase 1 the DelFly flies straight with 1m/s (faster than during earlier experiments to increase flight endurance). A threshold is used for the turn decision based on the amount of pixels that exceed the disparity constraint defined by the safety region. To suppress noisy measurements, filtering is applied as follows. Each time the threshold is exceeded (250 pixels), the current time is stored. It can then be checked if the threshold is exceeded ten times within one second. If this is the case, it is concluded that there is an obstacle. The earliest detection time of these ten detections is then used as reference time for the second phase. In Phase 2, the DelFly still flies straight and waits until it has reached the point in figure 14 where the turn needs to be started. This time to turn is just over 2 seconds. However, because the turn response of the DelFly to rudder inputs is sluggish initially, and because of communication delays, the time to turn was tuned experimentally and set a a value of 1500ms. After this time has elapsed the turn is initiated in Phase 3. During the turn it is checked if the current direction of flight is obstacle free. As soon as a lower threshold of 200 pixels is reached, Phase 4 starts. No filtering is used here because this will result in unwanted delay. The turn speed of the DelFly is around 1 rad/s and small delays result in large flight direction differences. To compensate for the quick decision making and turn overshoots, it is checked in Phase 4 if the new flight direction is indeed a safe

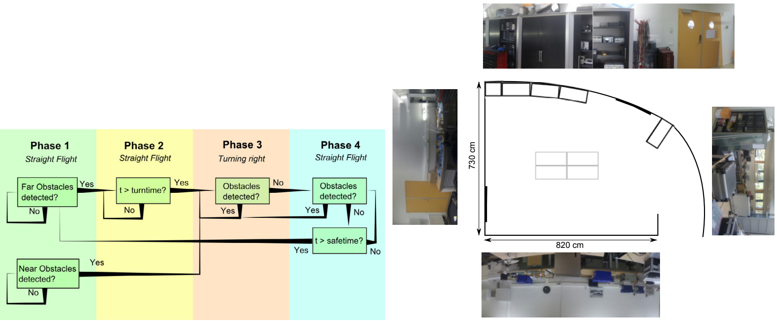


Fig. 15 Left: Flight phase diagram for rudder control. Right: Floor plan of the test room

direction to fly. If within one second after turn termination the obstacle threshold of 250 pixels is exceeded again, the turn is resumed in Phase 3. Otherwise the new flight direction is regarded as safe and Phase 1 starts again. In Phase 1 also another check is performed to detect obstacles at short range. The avoidance region defined as in the first experiment is used here. A threshold of 500 pixels is used. If it is exceeded three times in row, Phase 3 is activated to start turning immediately. The main reason for including this rule is the sensitivity of the DelFly to wind disturbances which changes the flight trajectory to such an extent that obstacles initially out of the field-of-view will result in collisions. This rule is used to prevent unexpected collisions but does not guarantee that the DelFly will be able to continue safely. Tests with this turn strategy were performed in a larger test room because of the size of the safety region. In the test room from the first experiment the DelFly would keep turning continuously. The test room is visualized in figure 15 (right).

Figure 16 shows the result from the test with the best result that has been obtained. During this test the DelFly flew around for 72.6s without hitting any object. The experimenter did not have to pull the DelFly up to keep it at a constant height. It should be noted that the experiment was ended without reason. The DelFly was still performing autonomous flight and the total successful test time could have been longer than the reported length.

The experiment starts at a point where the DelFly is coming out of a turn that was performed early after start up. This is point t=0 in figure 16. The track colors indicate the flight phases the controller is in according to figure 15 (left). During start of the experiment Phase 4 is active where the DelFly is ending its turn. During the next ten seconds Phase 1 is active and the DelFly should perform straight flight. From the flight track it can be observed that the flight is far from straight. Due to non-zero wind speeds in the test room, caused by ventilation and air conditioning systems, the DelFly swerves significantly. At this time this does not result in avoidance problems. When the DelFly approaches the upper wall after 10 seconds, Phase 2 and 3 are activated. These phases are combined in the figure. The flight track during the subsequent Phase 4 goes right past the cabinet. However, at this point the DelFly was flying above cabinet height and for that reason it was not considered to be an

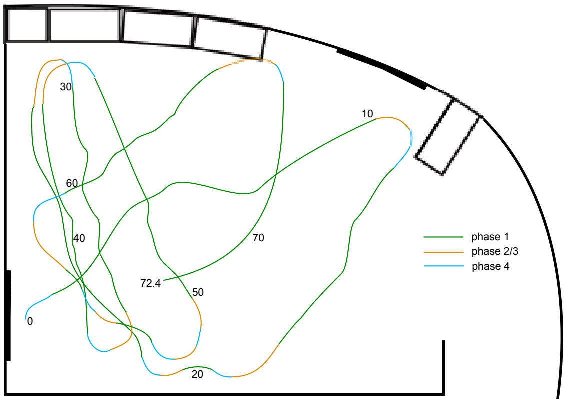


Fig. 16 Flight track of the DelFly during the experiment. The numbers indicate the flight time, the colors represent the flight phases.

obstacle. Only the wall needed to be avoided at this point. The flight is then continued in the direction of the lower wall. Note that in all cases where Phase 4 is active, the turn continues. In many cases the new flight direction is around 90 degrees further to the right. This is a result from all system delays, including video reception delay, processing time and radio control delay. After 20 seconds in the experiment Phase 1 is active again. Note that the flight track unexpectedly deflects to the left. As a result a new turn is triggered which directs the DelFly back to the upper wall. This sudden left turn can be explained by yaw/roll instability and is unpredictable. The cabinets in the top then force the DelFly to go back (at around 30 seconds). Note that back at the bottom wall the DelFly preserves a larger distance to the wall compared to other turns. Again the DelFly goes to the top cabinets and back. Just after 50s the lower wall is approached again. In this case an early turn is initiated which is ended too early. As a result Phase 4 is activated while the DelFly still continuous in the direction of the wall. Because the wall is detected again Phase 3 is active again after 684ms. The turn is then continued till the flight direction is now in the direction of the cabinets again. Again the DelFly unexpectedly turns quickly to the left and flies in the direction of the doors on the left. These are detected early and a slight turn follows immediately. Another turn is then initiated 1517ms later to avoid the left wall. At t=60 the DelFly is performing a straight flight in the direction of the top wall at a height above the cabinets. The wall behind the cabinets is then detected and avoided successfully. The flight ends after 72.6s without colliding with any obstacle. Note that during the last part of the flight the DelFly gradually but severely makes a turn to the right. Near obstacles (see bottom of Phase 1 in figure 15 (left) were never detected. This means that the DelFly never tried to avoid obstacles that were detected late.

6 Conclusions

From the results presented in this paper it can be concluded that stereo vision can be applied successfully for obstacle detection and avoidance on FWMAVs. It was shown that real-time stereo vision can provide accurate and sufficient obstacle information. By making use of suitable camera hardware the flapping motion of FW-MAVs has a minor influence on the stereo vision algorithm. In this respect this method outperforms optic flow techniques.

The small camera system is capable of giving distance estimates with a standard deviation of 20cm up to 5m. Even for texture-poor areas the accuracy is still adequate. The weight of the camera system and extra required battery leads to a reduced flight endurance and a reduced flight envelope i.e. hovering is not possible.

Closed-loop experiments showed that stereo vision can provide robust and reliable obstacle information that allows the DelFly to perform successful obstacle avoidance. An autonomous flight time of 72.6 seconds has been obtained as the best result.

One of the focuses of future research will be on the camera design. Lighter cameras with a wider field of view should result in better performance. Another important focus will be on onboard image processing. This will eliminate communication delays and the need for a ground station within communication range.

References

1. Griffiths, S., Saunders, J., et al.: Obstacle and Terrain Avoidance for Miniature Aerial Vehicles. In: Advances in Unmanned Aerial Vehicles (2007), doi:10.1007/978-1-4020-6114-1_7
2. Kownacki, C.: Guidance and obstacle avoidance of MAV in uncertain urban environment (2009), `http://www.emav2009.org/EMAV_final_papers/EMAV2009_papers.html` (cited October 1, 2012)
3. Hines, L., Arabagi, V., Sitti, M.: Free flight simulations and pitch and roll control experiments of a sub-gram flapping-flight micro aerial vehicle. In: IEEE ICRA, pp. 1–7 (2011)
4. Lin, S., Hsiao, F., et al.: Altitude control of flapping-wing mav using vision-based navigation. In: IEEE ICRA, pp. 3644–3650 (2009)
5. Duhamel, P.E., Perez-Arancibia, N., et al.: Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor. In: IEEE ICRA, pp. 4228–4235 (2012)
6. Beak, S., Fearing, R.: Flight forces and altitude regulation of 12 gram i-bird. In: IEEE RAS and EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob), pp. 454–460 (2010)
7. Beak, S., Garcia Bermudez, F., Fearing, R.: Flight control for target seeking by 13 gram ornithopter. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2011)
8. Garcia Bermudez, F., Fearing, R.: Optical flow on a flapping wing robot (Tech. Rep.) (2009), `http://www.eecs.berkeley.edu/~fgb/pubs.shtml` (cited October 1, 2012)

9. Tedrake, R., Jackowski, Z., et al.: Learning to fly like a bird (Tech. Rep.) (2006), `http://groups.csail.mit.edu/robotics-center/public_papers/Tedrake09.pdf` (cited October 1, 2012)
10. de Croon, G., de Clerq, K., et al.: Design, aerodynamics, and vision-based control of the Delfly. The Int. J. on MAVs 1(2), 71–97 (2009)
11. de Croon, G., de Weerdt, E., et al.: The appearance variation cue for obstacle avoidance. IEEE Transactions on Robotics 28(2), 529–534 (2012)
12. de Croon, G., Groen, M., et al.: Design, aerodynamics, and autonomy of the Delfly. Bioinspiration and Biomimetics 7(2) (2012)
13. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. International Journal for Computer Vision 47(1-3), 7–42 (2002)
14. Tombari, F., Mattoccia, S., Di Stefano, L.: Classification and evaluation of cost aggregation methods for stereo correspondence. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
15. Christian, B., Hesselbarth, S., Flatt, H., Blume, H., Pirsch, P.: Real-Time Stereo Vision System using Semi-Global Matching Disparity Estimation: Architecture and FPGA-Implementation. In: International Conference on Embedded Computer Systems, pp. 93–101 (2010)
16. Gehrig, S.K., Everli, F., Meyer, T.: Real-Time Semi-Global Matching on the CPU. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 85–92 (2010)
17. Chen, B., Chen, H.: A realization of semi-global matching stereo algorithm on GPU for real-time application. In: MIPPR 2011: Pattern Recognition and Computer Vision (2011)
18. Hirschmüller, H.: Stereo processing by Semi-Global matching and mutual information. IEEE Trans. Pattern Anal. Machine Intell. 30(2), 328–341 (2008)
19. Barnich, O., van Droogenbroeck, M.: Vibe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing 20(6), 1709–1724 (2011)

The Total Energy Control Concept for a Motor Glider

Maxim Lamp and Robert Luckner

Abstract. In this article the Total Energy Control System (TECS) that was introduced by Lambregts to control the vertical flight path and the velocity of an aircraft by using the total energy and the energy distribution between the flight path and the acceleration, will be taken up, modified, extended and tested on a motor glider. The TECS concept has been extended by using the airbrakes as additional control elements to manipulate the total energy. For motor gliders and utility aircraft with a high glide ratio this increases the sink performance and the range of possible missions, like steep approaches. Further modifications are done to improve the height accuracy during normal operation and during flare manoeuvre and to improve the control response reaching its saturations. A height protection is introduced to make a safe flight near to the ground possible. The usage and generation of required sensor signals from existing sensor data is introduced. Examples of flight test results are given.

Nomenclature and Abbreviations

C_D	$[-]$	drag coefficient
C_L	$[-]$	lift coefficient
C_m	$[-]$	moment coefficient
E_{dist}	$[Nm]$	energy distribution
E_{kin}	$[Nm]$	kinetic energy
E_{pot}	$[Nm]$	potential energy
E_{spec}	$[Nm]$	specific energy
E_{tot}	$[Nm]$	total energy
F_D	$[N]$	drag force
F_G	$[N]$	weight force
F_T	$[N]$	thrust force
H	$[m]$	height

<u>Indizes</u>	
0	intial value
A	aerodynamic
K	flight path
W	Wind
cas	calibrated airspeed
cmd	command
f	body fixed coordinate frame
max	maximum
min	minimum
res	resulting
tas	true airspeed

Maxim Lamp · Robert Luckner
Berlin Institute of Technology,
Department of Flight Mechanics, Flight Control and Aeroelasticity,
Berlin, Germany
e-mail: {Maxim.Lamp,Robert.Luckner}@ilr.tu-berlin.de

K	$[...]$	gain
P	$[Nm/s]$	power
V	$[m/s]$	velocity
g	$[m/s^2]$	constant of gravitation
m	$[kg]$	mass
n	$[g]$	acceleration
Θ	$[rad]$	pitch angle
Φ	$[rad]$	bank angle
γ	$[rad]$	flight path angle

Abbreviations	
AD	Air Data System
AFCP	Automatic Flight Control Panel
AFCS	Automatic Flight Control System
AW	Anti Wind-Up
FCL	Flight Control Laws
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MSU	Magnet Sensor Unit
TECS	Total Energy Control System

1 Introduction

The aim of the research project LAPAZ[1] is to develop and to demonstrate an Automatic Flight Control System (AFCS) [5]. The AFCS is developed for a single engine, long-endurance motor glider, the STEMME S15 of the German aircraft manufacturer STEMME. The Flight Control Laws (FCL) of the AFCS provide a high-precision flight path control, automatic navigation control along 3D trajectories and automatic landing as well as automatic takeoff functions [11]. It can be used in piloted and unmanned operations (optionally piloted).

Since the aircraft is a motor glider with a high glide ratio, it can be used for long endurance missions at high altitudes. To use those attributes, it is of advantage that the FCL consider the total energy of the flying system as Lambregts has shown, when he introduced the Total Energy Control System (TECS). TECS uses thrust and elevator to control the total energy and the energy distribution [6] – [10]. This concepts is characterized by a simple structure that relates to laws of physics and can be easily adopted to different aircraft types.

In [6] the thrust, receptively the engine is used as the only control element to modify the total energy. If the engine is in idle position, a further reduction of the total energy is not possible. That means the largest achievable sink rate in stationary flight is a function of the geometric configuration and aerodynamic characteristic of the aircraft. Especially during descents with simultaneous deceleration, e.g. landing approaches, the sink performance can limit the mission scope. Using a glider or a motor glider with high glide ratio, the usage of airbrakes is an adequate option to control the total energy by varying the drag force.

In the FCL of the LAPAZ project, the existing TECS concept has been taken up, modified and extended. Figure 1 depicts the TECS core system from [6]. It can be divided into a total energy and an energy distribution control path. Section 2 discusses the total energy and thrust branch, Sect. 4 the energy distribution and elevator branch.

[1] ”Luftarbeitsplattform für die Allgemeine Zivilluftfahrt”, aerial work platform for general aviation.

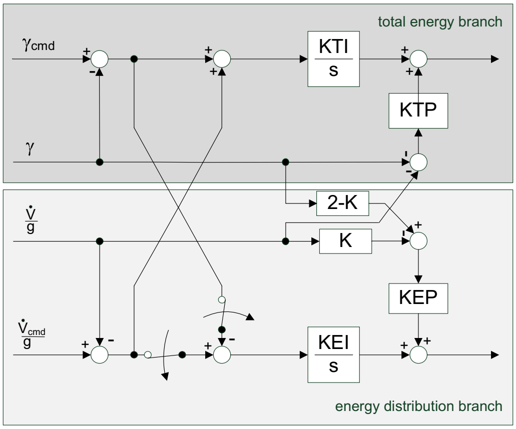


Fig. 1 Basic Total Energy Control System from [6]

One significant change is the usage of the airbrakes as an additional control element. Other changes address the improvement of the height accuracy, the possibility to use TECS during flare manoeuvres and of the behaviour of the control elements reaching their saturation.

2 Using the Total Energy Control Concept with Airbrakes

This section describes the control concept that Lambregts introduced in [6] – [10]. After a brief repetition of the theoretical aspects, the control concept will be expanded.

If the rotational energy is neglected, the total energy E_{tot} is the sum of kinetic energy E_{kin} and potential energy E_{pot} and can be expressed as

$$E_{tot} = E_{pot} + E_{kin} = \frac{m}{2} \cdot V_K^2 + m \cdot g \cdot H \tag{1}$$

with aircraft mass m, height H, flight path velocity V_K and gravitational constant g. Under the assumption of constant mass, the excess power is given by

$$\frac{\dot{E}_{tot}}{F_G} = \frac{V_K \cdot \dot{V}_K}{g} + \dot{H} \quad . \tag{2}$$

A division by the flight path velocity yields to the specific energy rate $\dot{E}_{spec}$, where F_G is the weight force and the dot indicates the time derivative leads to

$$\dot{E}_{spec} = \frac{\dot{E}_{tot}}{F_G \cdot V_K} = \frac{\dot{V}_K}{g} + \frac{\dot{H}}{V_K} \tag{3}$$

$$\dot{E}_{spec} = \frac{\dot{V}_K}{g} + sin(\gamma) \quad . \tag{4}$$

The specific energy rate represents a measure of the system's energy gain or loss. Simplifying the aircraft as a point mass in an inertial system without wind, the second axiom of motion along the flight path leads to

$$m \cdot \dot{V}_K = F_T - F_D - sin(\gamma) \cdot F_G \quad , \tag{5}$$

where F_T is the thrust force, F_D is the drag force and F_G is the weight.

Assuming small flight path angles $(sin(\gamma) \approx \gamma)$ and by comparison of eq. (4) and eq. (5) the specific energy rate can be expressed by

$$\dot{E}_{spec} = \frac{\dot{V}}{g} + \gamma = \frac{F_T - F_D}{F_G} \quad . \tag{6}$$

This obvious shows, that the thrust force and the drag force can be used to influence $\dot{E}_{spec}$ and thus the total energy of the system. In stationary flight the total energy is balanced, thrust force and drag force must be equal $F_{D_0} = F_{T_0}$. In this case the specific energy rate is zero. A change in total energy can be achieved by changing the thrust and drag forces individually or simultaneously

$$F_G \cdot \Delta\dot{E}_{spec} = F_G \cdot \left(\frac{\Delta\dot{V}_K}{g} + \Delta\gamma \right) \quad . = \Delta F_T - \Delta F_D \tag{7}$$

At this stage the assumption of no wind $V_K = V_A$ was made. That may imply, that TECS uses only kinetic and potential energy in a pure physical sense. However, flight control has to fulfil multiple requirements, depending on flight phase and mission objectives. In many flight phases the objectives regard the airspeed, close to the ground, flight path velocity becomes important. The velocity equation $V_K = V_A + V_W$, that relates flight path velocity V_K, airspeed V_A and wind velocity V_W, indicates that it is not possible to control V_K and V_A at the same time, when the wind velocity changes. Additionally, it is not simple to measure flight path velocity and flight path angle with high bandwidth and accuracy over the full flight envelope. Depending on the available sensors, compromises have to be found. Instead of using V_K, $\dot{V}_K$ and γ the measured and estimated signals V, $\hat{\dot{V}}$ and $\hat{\gamma}$ that are explained in Sect. 5 will be used in the FCL.

In the concept of the Total Energy Control System [6]–[10], the only control element to modify the total energy is the thrust force, respectively the engine. To generate a thrust command, an integral controller (I-controller) with a proportional feedback is used. Here, the total energy control concept will be expanded by using the airbrakes as an additional control element. There are two possibilities to integrate the airbrakes as a control element into the existing controller concept.

The first possibility is to use the existing linear controller to compute an incremental force command ΔF, which can be divided into incremental thrust force and drag force commands, see Fig. 2. Those commands can then be converted into appropriate engine or airbrake commands. In the simplest case, thrust and drag forces can be commanded by using a function that computes airbrake commands if the engine is at idle. The advantage of this concept is the simple structure of the

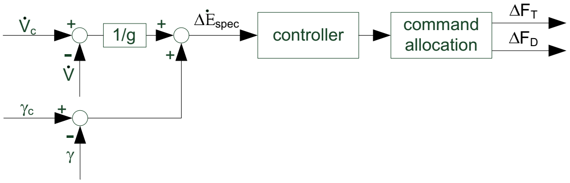


Fig. 2 Control of specific energy rate by one controller and subsequent separation into incremental thrust and drag force commands

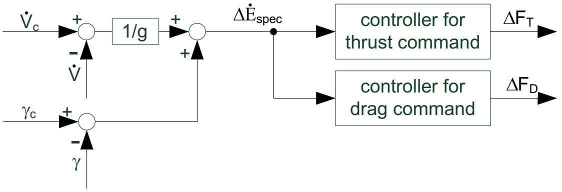


Fig. 3 Control of specific energy rate by using individual controllers for incremental thrust and drag force commands

control law, because only one controller has to be designed. However, the dynamic behaviour and the effectiveness of the engine and the airbrakes are different, which will be shown in Sect. 3. With the design of only one linear controller these differences cannot be directly considered, which is a significant disadvantage.

The second possibility is to use two separate linear controllers, one to compute the required incremental thrust force ΔF_T and one to compute the required incremental drag force ΔF_D, as can be seen in Fig. 3. These two controllers can be designed independently and can have different structures, which allows taking the differences in dynamic behaviour and effectiveness of the two control elements into account. Simultaneous use of both control elements is also possible. This can be an advantage for example during landing approaches in turbulent atmospheric conditions and if large engine activity has to be avoided. During the approach the engine can run approximately constant, while the airbrakes can be used to correct gusts or wind shear disturbances. The concept also has a beneficial effect in case of a missed approach (Go Around) manoeuvre. By retracting the airbrakes in this case, the drag is reduced rapidly while the total lift coefficient increases.

As with the first concept, there are also disadvantages. Within the control ranges of the control elements there are unlimited possible combinations of thrust and drag commands, including undesirable combinations. For example, cruise flight with partially extended airbrakes and simultaneous increased thrust setting could be conceivable, which will result in excessive fuel consumptions. This problem can be solved either by using different dynamics for the respective controllers or by using logical switching.

For the FCL of the LAPAZ project, a control structure based on the second concept is used. The usage of the airbrakes is optional and can be switched on and off. Figure 4 shows the structure of the total energy controller. The incremental thrust command ΔF_T results from the deviation of the specific energy rate and is controlled by using an I-controller, combined with a proportional feedback of the potential flight path angle $(\hat{\dot{V}}/g + \hat{\gamma})$, similar to [6]. A difference to the scheme in

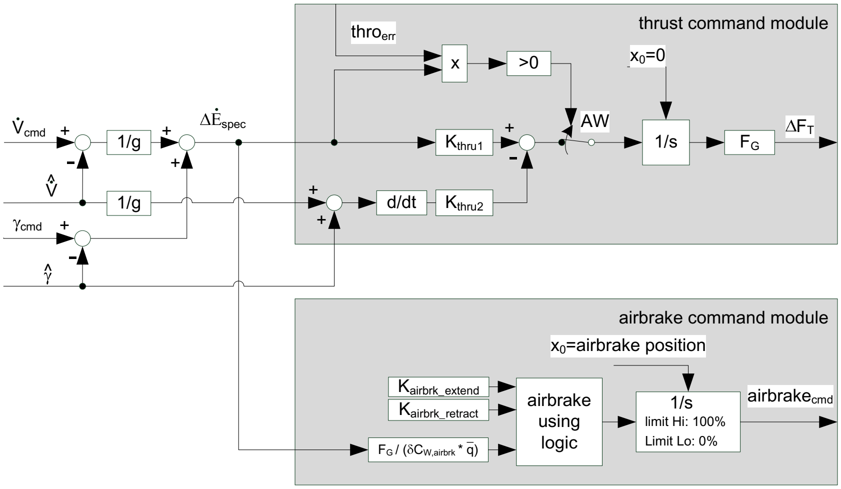


Fig. 4 Control scheme for total energy control with thrust and airbrakes commands

[6] is the feedback of the signal $(\hat{\dot{V}}/g+\hat{\gamma})$, which is subtracted behind the integrator in the original control scheme. In the control concept here, it is subtracted before the integrator. In order to do so, it must be differentiated first. The final outcome of this signal by differentiation and subsequent integration remains the same: $\frac{\hat{V}}{g}+\hat{\gamma}=\int\frac{\hat{\dot{V}}/g+\hat{\dot{\gamma}}}{dt}dt$. Because the feedback signal is subtracted before the integrator it is cancelled when the anti wind-up control is active. As a result the feedback will not lead to unwanted thrust commands when thrust is saturated, especially in turbulent atmospheric conditions where disturbances in $\hat{\gamma}$ and $\hat{\dot{V}}$ will directly influence the thrust command.

Figure 5 shows a simulation study to demonstrate the reaction of the throttle by adding the feedback of the potential flight path angle before and behind the integrator. The simulation starts at a descent with a vertical speed of approximately $\dot{H}=-1.8\,m/s$ and an airspeed of $V_{cas}=130\,km/h$. The throttle is in idle (8%). At time t=96 s, turbulence is activated with a standard deviation of about $0.5\,m/s$. Until time t=120 s the damping feedback was added before the integrator, afterwards it was added behind the integrator. As can be seen, the throttle command is much smoother when the feedback is added in front of the integrator. The bottom plot of Fig. 5 depicts the throttle command $thro_{cmd}$ and the true throttle position at engine. It shall be noted, that the simulation model includes friction, hysteresis and elasticity of the throttle actuator and the control linkage between the actuator and the throttle at the engine.

To compute the required incremental drag force and subsequent airbrakes command, an I-controller is used. Its output can be regarded as an airbrakes trim. The dynamics of the airbrakes command controller are different for the case of airbrakes extension and retraction. The gain for retraction is higher than the gain for extension. In case of retraction this prevents from using the airbrakes and the throttle together at the same time. Beside of this, the safety increases in case of retraction. The behaviour in case of extension can be adjusted to fit with the throttle behaviour, so that, after reaching idle, a commanded total energy reduction can command the airbrakes in a way that the behaviour is the same as with reduction of throttle. A logic prohibits that the airbrakes are further extended during turn.

The motor glider STEMME S15 is equipped with a single piston engine with an adjustable propeller. The FCL generate throttle and propeller revolution commands to drive the engine at its best performance. Figure 6 depicts schematically the structure of the engine command module. The incremental thrust command ΔF_T together with the airspeed and the propeller efficiency are converted to an incremental power command ΔP. This is send to an integrator with a very low time constant, which

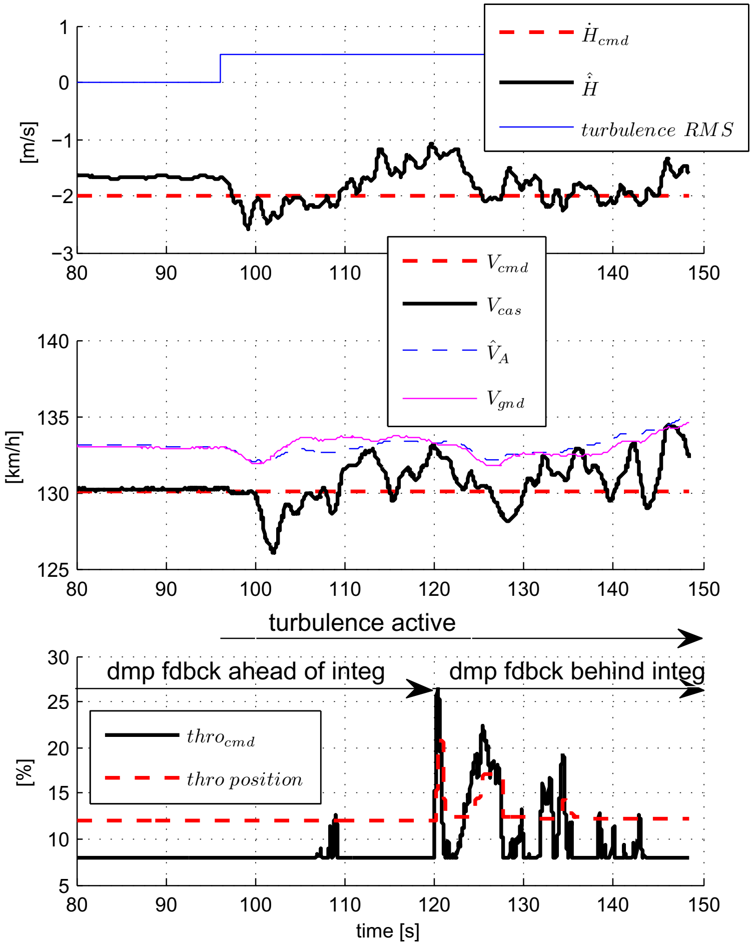


Fig. 5 Simulation results, proportional feedback before and behind the integrator

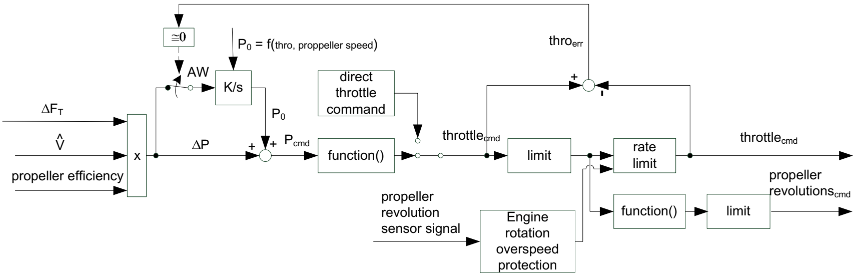


Fig. 6 Scheme of the engine command module

output can be regarded as a trim of power. The throttle command is computed from the sum of the trimmed power and the incremental power command. The absolute value is limited. An engine protection function prevents the engine to exceed the maximum allowable rotation speed by limiting the throttle command ratio. Unlike in [6] there is no knowledge about the current net thrust nor the engine pressure ratio (EPR) and no performance prediction model of the engine is used to determine the maximum and minimum power. So these informations cannot be used to prevent a wind-up of the integrator in the total energy controller. The anti wind-up is achieved here by using the difference between the throttle command and the throttle command limit to feed an anti wind-up switching logic (AW) in the engine command module and in the thrust command module, see Fig. (4) and Fig. (6).

3 Analysis of Throttle and Airbrakes Responses

A simulation analysis, using the non-linear aircraft model of the motor glider STEMME S15 and the AFCS model was conducted to analyse the differences in reaction of total energy due to step changes in throttle setting and airbrake deflection. The simulation includes the dynamic behaviour of the engine and the airbrakes. The initial trimmed condition was a steady descent with a vertical speed of $\dot{H} = -1.5\,m/s$ and an airspeed of $V = 130\,km/h$. The throttle and the airbrakes control loops were opened and throttle and airbrakes were directly stimulated in the aircraft simulation model. The dynamic behaviour of the engine and airbrakes are included. While one control element was stimulated at a time, the respective other control element was held at its position. The elevator command loop stayed closed. Thus the FCL commanded the elevator to compensate the flight path deviation (*flight path priority*, as described in Sect. 4).

Figure 7 shows the responses of the specific total energy rate $\dot{E}_{spec}$ (Fig 7a and 7e), of the deviation of specific total energy rate $\Delta\dot{E}_{spec}$ (Fig 7b and 7f), of the flight path deviation $\Delta\gamma$ (Fig 7c and 7g) and of the deviation of acceleration along to flight path $\Delta\dot{V}/g$ (Fig 7d and 7h) due to increase of throttle command $thro_{cmd}$ and decrease of the airbrake command $airbrk_{cmd}$ and vice versa. During the simulation the total energy was reduced by decreasing the throttle command from the 50% to

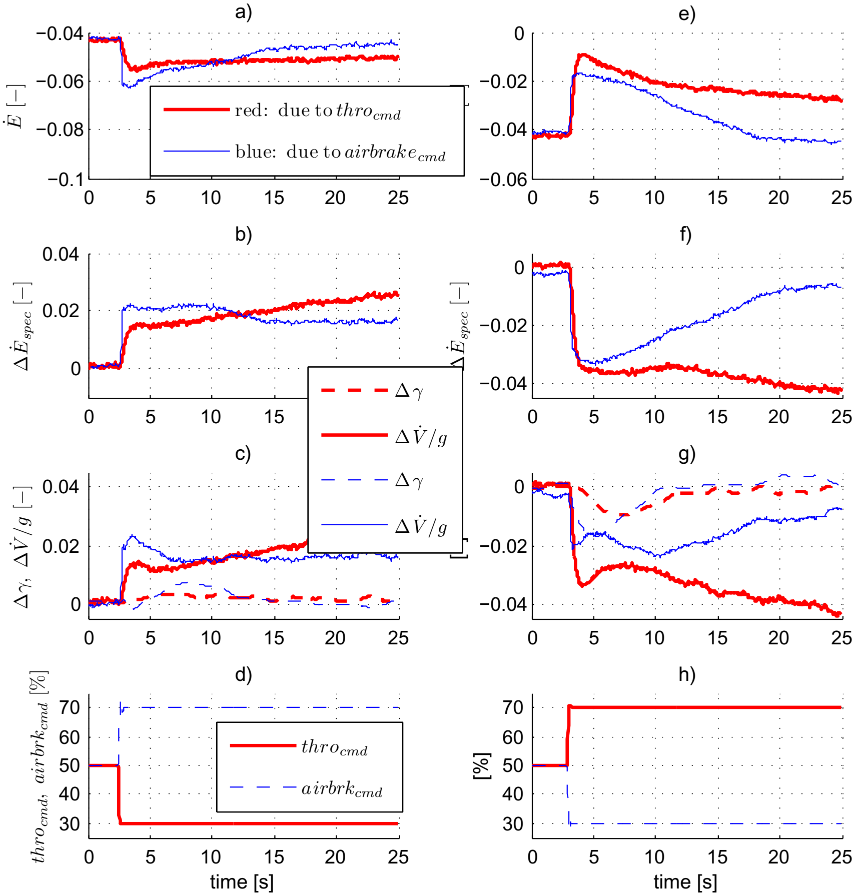


Fig. 7 Step responses of throttle and airbrake commands, regarding to specific energy rate

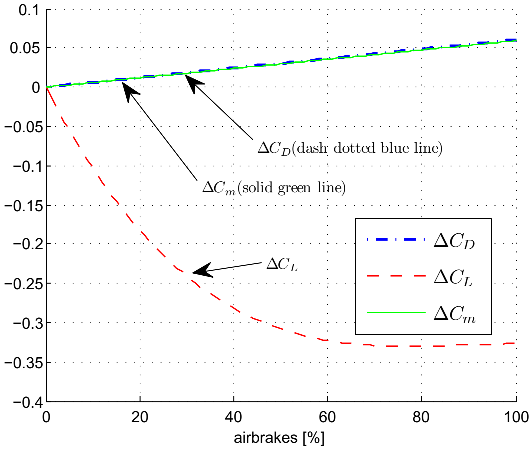
30% or by extending the airbrakes from 50% to 70%, see Fig. 7 a-d. Comparing the deviations of the total energy rates, it can be seen that the airbrakes are somewhat more effective in the initial reaction than the throttle. Both control elements mainly produce changes in the flight path acceleration, see Fig. 7 c. The deviations in the flight path angle are small, as it is controlled by the elevator.

Simulations with appropriate control deflections to increase total energy were also performed, see Fig. 7 e-h. With respect to total energy, both control elements have approximately the same effect. While a step change in throttle mainly leads to a deviation in the flight path acceleration, a step in airbrake deflection leads to deviations in the flight path acceleration and the flight path angle. The deviation in the flight path angle is later reduced by elevator deflections commanded by the FCL. The error in the flight path acceleration remains.

In addition to the change in drag, the airbrakes also generate a change in the total lift coefficient of the aircraft and in the pitching moment, see Fig. 8. So it is possible to manipulate not only the drag, which means the path acceleration, but also the lift, which means the flight path. Thus the airbrakes are well suited for simultaneous velocity and flight path changes, e.g. for simultaneous deceleration and descent during initiation of landing approach.

Figure 8 shows that the change of the lift coefficient due to airbrake extension is non-linear. The gradient goes to zero beyond the 50% position. That is why the reaction in the deviation of the flight path is so small during the experiments where the airbrakes were extended from 50% to 70%. It must be taken into account, that the airflow around the wing induced by the propeller is not modelled in the simulation model. This will also impact the total lift of the aircraft.

Fig. 8 Changes in lift, drag and pitching moment coefficients due to airbrake deflection



4 Modification of the Energy Distribution Concept

The redistribution of the total energy between the kinetic and potential energy can be achieved by using the elevator, as described in [6] or by pitch attitude as in [12]. This latter method is adopted here. If there is no change in total energy, that means $\Delta \dot{E}_{spec} = \frac{\Delta \dot{V}}{g} + \Delta\gamma = 0$, an increase in potential energy results in a reduction of kinetic energy and vice versa

$$-\frac{\Delta \dot{V}}{g} = \Delta\gamma \quad . \tag{8}$$

According to [6] the energy distribution error $\Delta \dot{E}_{dist}$ is defined by

$$\Delta \dot{E}_{dist} = -\frac{\Delta \dot{V}}{g} + \Delta\gamma \quad . \tag{9}$$

Reducing this energy distribution error to zero means that the total energy rate is divided equally in potential and kinetic energy. This concept of equal distribution into potential and kinetic energy is used in [6] and other studies, if the engine is not saturated. If the engine is saturated, concepts are proposed which prioritize the energy distribution in favour of the potential or kinetic energy. This is called *flight path priority* and *speed priority*. In those cases the elevator controls the velocity respectively the flight path angle. By using speed priority, a speed protection can be achieved [6]. The commanded must not be lower than the allowed minimum speed or greater than the maximum speed.

By designing the total energy controller and the energy distribution controller with the same dynamic behaviour, a decoupling of the longitudinal and the vertical control axes of the aircraft can be achieved [6]. The slow dynamic behaviour of the engine limits the bandwidth for achieving pure flight path and speed control decoupling.

Since the change of the total energy is a function of the dynamic behaviour and the effectiveness of the engine or the airbrakes, it cannot be assured that the deviation in total energy rate, for example due to a change of the velocity command or due to a wind disturbance, can be reduced rapidly. In that case the concept of equal distribution of total energy into potential and kinetic energy may lead to a deviation in both, in flight path angle and in velocity.

In the LAPAZ project, high-precision flight path following is required. To minimize the path deviation, deviations in airspeed are allowed, as long as they are not violating the speed envelope. The flight path priority is used as the standard mode to distribute the energy. In the LAPAZ demonstrator the pilot has the possibility to limit the vertical speed manually and by this to control the direction of the energy distribution. This can be done via an Automatic Flight Control Panel (AFCP), designed especially for the LAPAZ demonstrator.

To detect a violation of the permitted speed envelope while using flight path priority, a resulting acceleration command $\dot{V}_{cmd,res}$ is computed based on the deviation in flight path angle $\Delta\gamma$ and the current path acceleration $\hat{\dot{V}}$

$$-\frac{\Delta\dot{V}}{g} = \Delta\gamma \qquad \text{with} \quad \Delta\dot{V} = \dot{V}_{cmd,res} - \dot{V} \tag{10}$$

$$\dot{V}_{cmd,res} = -\Delta\gamma\cdot g + \dot{V} \quad . \tag{11}$$

A comparison of the resulting acceleration command with the acceleration commands corresponding to the minimum and the maximum speeds ($\dot{V}_{min}$ and $\dot{V}_{max}$) conduct the priority switching logic to switch to speed priority. By using the minimum and maximum acceleration commands, which are calculated by current mass, altitude and flap setting, a speed priority is realized.

Lambregts introduces in [6] a similar value $\dot{V}_{max} = \dot{V} + \gamma \cdot g$ which in case of engine saturation limits the acceleration, so that a maximum level ($\gamma = 0$) acceleration/deceleration can be obtained. Unlike here the value $\dot{V}_{cmd,res}$ is not calculated by using γ but by using the flight path deviation $\Delta\gamma$. The resulting acceleration command is not used for limiting the acceleration, instead it is used to trigger the priority switch, as described before.

In an earlier version of the FCL, the only possibility to regulate the energy distribution during descent while the flight path priority was active, was to command the vertical speed. Flight tests in the course of the LAPAZ project have shown, that while using flight path priority during descent, this can confuse the pilot. For example, the pilot tried to reduce the increasing velocity during descent with high sink rate and throttle at idle by reducing the commanded velocity instead by reducing the sink rate, not regarding, that the velocity error is not controlled during flight path priority. This finding resulted in the requirement to use speed priority during descent if the control elements to control the total energy are saturated.

In [6] a linear I-controller with a proportional feedback was introduced to compensate the energy distribution error. This controller concept is used in a modified form here. Figure 9 shows the scheme of the energy distribution controller. The module *energy distribution control priority logic* contains the priority switching logic. Compared with the structure in [6] the proportional feedback does not contain the path acceleration $\dot{V}$. This design choice reduces unwanted pitch commands in turbulent atmospheric conditions.that would be induced by $\dot{V}$ feedback.

Furthermore, after differentiating, the feedback signal $\hat{\gamma}$ is subtracted before the integrator instead of behind, as in the *total energy control module*, see Fig. 4.

The integrator is initialized with the current pitch attitude angle. Together with the commanded incremental changes the integrator computes an absolute pitch attitude command for the inner loop pitch controller. Adding an attitude trim variable Θ_0, as in [12] is not necessary, since the integrator output is the total pitch attitude command and not an incremental attitude command.

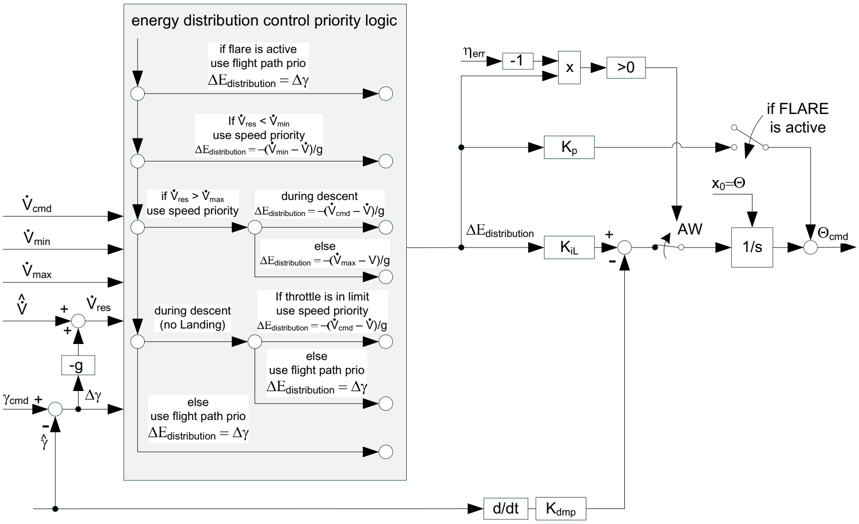

energy distribution control priority logic
if flare is active
use flight path prio
$\Delta E_{distribution} = \Delta\gamma$
If $\dot{V}_{res} < \dot{V}_{min}$
use speed priority
$\Delta E_{distribution} = -(\dot{V}_{min} - \dot{V})/g$
if $\dot{V}_{res} > \dot{V}_{max}$
use speed priority
during descent
$\Delta E_{distribution} = -(\dot{V}_{cmd} - \dot{V})/g$
else
$\Delta E_{distribution} = -(\dot{V}_{max} - V)/g$
during descent
(no Landing)
If throttle is in limit
use speed priority
$\Delta E_{distribution} = -(\dot{V}_{cmd} - \dot{V})/g$
else
use flight path prio
$\Delta E_{distribution} = \Delta\gamma$
else
use flight path prio
$\Delta E_{distribution} = \Delta\gamma$
$\dot{V}_{cmd}$
$\dot{V}_{min}$
$\dot{V}_{max}$
$\hat{\dot{V}}$
+
+
$\dot{V}_{res}$
-g
γ_{cmd}
+
-
$\hat{\gamma}$
$\Delta\gamma$
η_{err}
-1
x
>0
if FLARE
is active
K_p
$x_0 = \Theta$
$\Delta E_{distribution}$
K_{iL}
+
-
AW
1/s
Θ_{cmd}
d/dt
K_{dmp}

Fig. 9 Scheme of energy distribution control

A major disadvantage of the used I-controller with additional proportional feedback is the low bandwidth. Therefore, this controller structure cannot be used for flight modes or missions that require a large bandwidth. The low bandwidth is a limit for automatic flare manoeuvres, especially in wind shear, gusty wind conditions or after a steep approach. Lambregts adds in [9] the rate of flight path command $\dot{\gamma}_{cmd}$ as a feed forward signal to the input of the thrust command integrator and to the integral and proportional term of the elevator path. This provides a faster flight path angle response and is used for manual control in *Control Wheel Steering* mode.

In the LAPAZ project, a proportional controller was added to the existing controller structure. It can be activated if necessary. The usage of this proportional component adds a zero to the transfer function and allows to shape and enlarge the bandwidth. So the previous controller structure can be used with this little modification. Niedermeier uses in [13] a similar modification. His energy distribution controller consists of a classical proportional-integral controller (PI-controller). The proportional controller part is always active.

The architecture of the mode logic of the vertical modes is shown in Fig. 10. The structure, introduced in [6] is expanded by the vertical speed limitation and a height protection. By using the vertical speed limitation, the pilot is able to control the direction of energy distribution, as described above. The vertical speed limitation only becomes active if necessary.

The height protection limits flight path commands in order to avoid flights below a minimum height above ground $H_{gnd,min}$. Using the height protection enables safe flight at low heights above ground, which expands the range of possible missions of the utility aircraft. By limiting the commanded flight path angle, instead of a switching, this protection becomes active without transient.

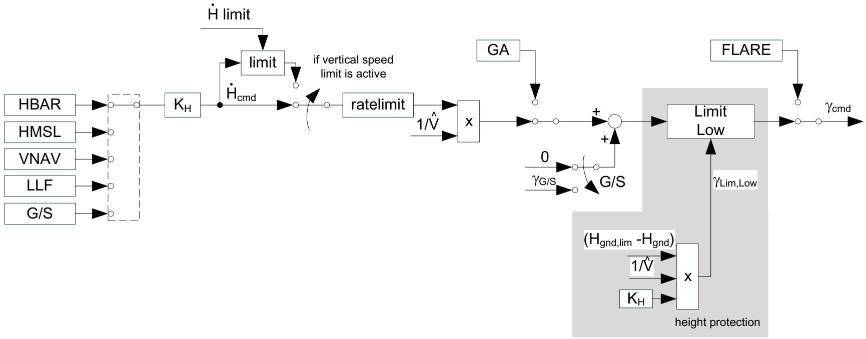


Fig. 10 Mode switching architecture of height and flight path control

5 Sensor Signals and Filters

The AFCS of the LAPAZ demonstrator has a duo-duplex architecture with redundant sensors, see Fig. 11. The Attitude and Heading Reference Systems (AHRS) are triplex redundant. The Air Data Systems (AD), the Magnetic Sensor Units (MSU)

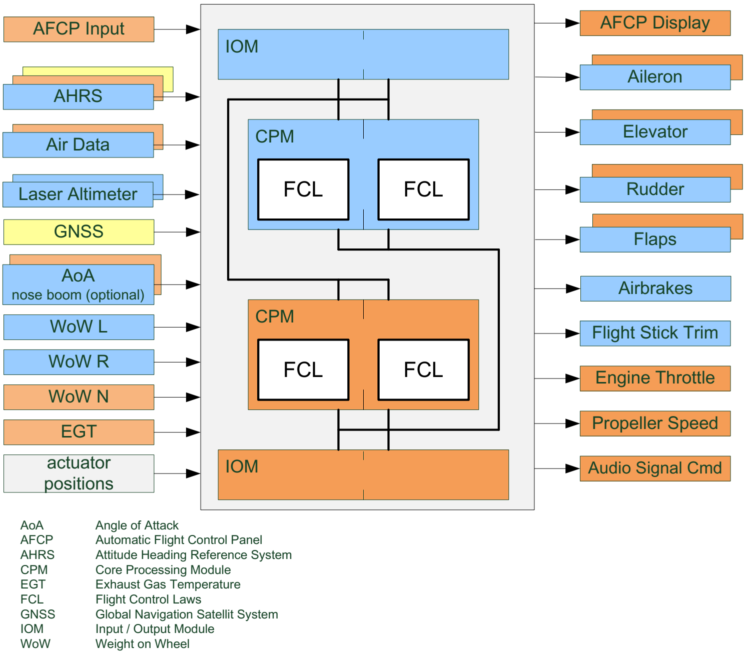


Fig. 11 Architecture of the Automatic Flight Control System

and the height above ground signals from the Laser Altimeters (LA) are duplex redundant. The Global Navigation Satellite System (GNSS) is simplex (no redundancy).

Since simplex sensor signals can only be range checked and cannot be monitored, the integrity of those signals cannot be assured. For safety critical flight control functions and modes, such as flight path control, consolidated sensor signals of high integrity are required. In the S15 demonstrator signals of the flight path angle γ and the ground speed V_{gnd} are only simplex. In an operational system, the redundancy will be duplex or triplex. As the test pilot has to monitor the system behaviour closely to assure flight safety, then simplex sensor signals can be used to demonstrate the Automatic Take-Off and Automatic Landing functions.

The flight path acceleration $\dot{V}_K$ is computed by the body accelerations in longitudinal, lateral and normal direction $n_{x,f}$, $n_{y,f}$ and $\Delta n_{z,f,up}$ and the attitude angles Φ and Θ

$$\dot{V}_K = g \cdot \begin{bmatrix} cos\Theta & sin\Theta sin\Phi & sin\Theta cos\Phi \end{bmatrix} \cdot \begin{bmatrix} n_{x,f} \\ n_{y,f} \\ n_{z,f} \end{bmatrix} \quad . \tag{12}$$

It must be noted, that the body normal acceleration $\Delta n_{z,f,up}$ from the AHRS is zero for stationary horizontal flight. The positive sense is up. It has to be transformed by $n_{z,f} = -(\Delta n_{z,f,up} + 1)$.

The velocity $\hat{V}_A$ is computed, using a complementary filter, see Fig. 12. The true airspeed V_{tas} is passed through a low pass filter with a high time constant ($\tau_1 = 10\,s$) to restrain high-frequency atmospheric turbulence. The resulting lag is compensated by adding the flight path acceleration $\dot{V}_K$. The computed velocity $\hat{V}_A$ is used to compute the flight path angle $\hat{\gamma}$ and flight path angle commands from the vertical speed commands.

To take into account wind changes of low bandwidth, e.q. wind shear, low path filtered flight path acceleration $\dot{V}_K$ and airspeed acceleration are added, see Fig. 12. After passing a low pass filter, the complementary filtered acceleration $\hat{\dot{V}}$ is used for total energy control.

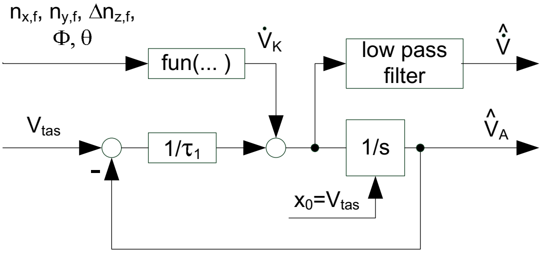


Fig. 12 Block diagram of the velocity and acceleration signal filter

An observer is used to compute the inertial vertical speed $\hat{\dot{H}}$ from the body normal acceleration $\Delta n_{z,f,up}$ and the barometrical altitude H_{baro}, see Fig. 13. The inertial vertical speed is required to compute the flight path angle.

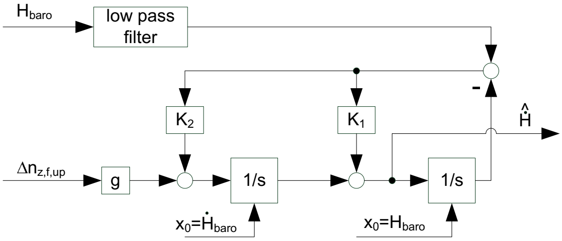


Fig. 13 Block diagram of the vertical speed signal filter

Beside the flight path angle γ, which can be calculated by the inertial vertical speed and the flight path speed V_K or the ground speed V_{gnd}, an approximated flight path angle $\hat{\gamma}$ is computed by the inertial vertical speed $\dot{H}$ and the complementary filtered velocity $\hat{V}_A$

$$sin(\hat{\gamma}) = \frac{\hat{\dot{H}}}{\hat{V}_A} \quad . \tag{13}$$

The flight path angle γ is used for earth referenced missions, like glide slope tracking. The estimated flight path angle $\hat{\gamma}$ is used for airmass referenced missions, e.g.

altitude changes with given vertical speeds. By using this signal, vertical speed following in wind conditions can be achieved, since the energy, induced by wind to the system, can be considered.

If there is no wind, flight path angle and estimated flight path angle are the same ($\gamma = \hat{\gamma}$). In Fig. 14, the geometric interrelation of the wind, the flight path angle and the computed flight path angle is shown in the longitudinal vertical plane. With increasing wind, the difference between both angles increases.

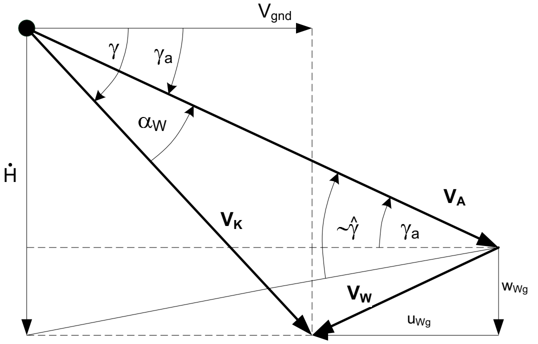


Fig. 14 Geometric interrelation of the wind, the flight path angle and the computed flight path angle

6 Flight Tests Results

The first automatic flight took place in August 2010. Since then, more than 80 hours of successful flight tests were done with the AFCS. In these test flights the TECS function was validated. The first successful automatic landing took place on March 22nd, 2012.

The performance of TECS with and without airbrakes was investigated in flight. Figure 15 shows a time plot of two descents, commanded by steps in height of $\Delta H_{cmd} = -300\,m$. The vertical speed limit was set to $\dot{H}_{Lim} = -10\,m/s$. The target is to reach $\dot{H}_{Lim}$ during the descent. The commanded airspeed of $V_{cas} = 150\,km/h$ should be kept constant. The use of the airbrakes for total energy control was optionally activatable.

Between t=0 s and 150 s the airbrakes were not used. After t=150 s the airbrakes were used. In the first descent without airbrakes, the throttle was commanded to idle position. The maximum speed protection was detected and the speed priority was activated. The elevator controls the commanded velocity, see the *energy distribution control priority logic module* in Fig. 9. The small airspeed deviation at the beginning of the descent, that was induced by the fast reduction of the thrust, was quickly reduced. The sink rate approached a constant value of $\dot{H} \approx -3\,m/s$. The vertical speed command of $\dot{H}_{Lim} = -10\,m/s$ was not reached. Shortly before reaching the commanded height, the FCL switched from speed priority to flight path priority. This caused direct reaction of the airspeed which increases by 14 km/h and the vertical speed which increases by 2.5 m/s. This behaviour is not satisfactory.

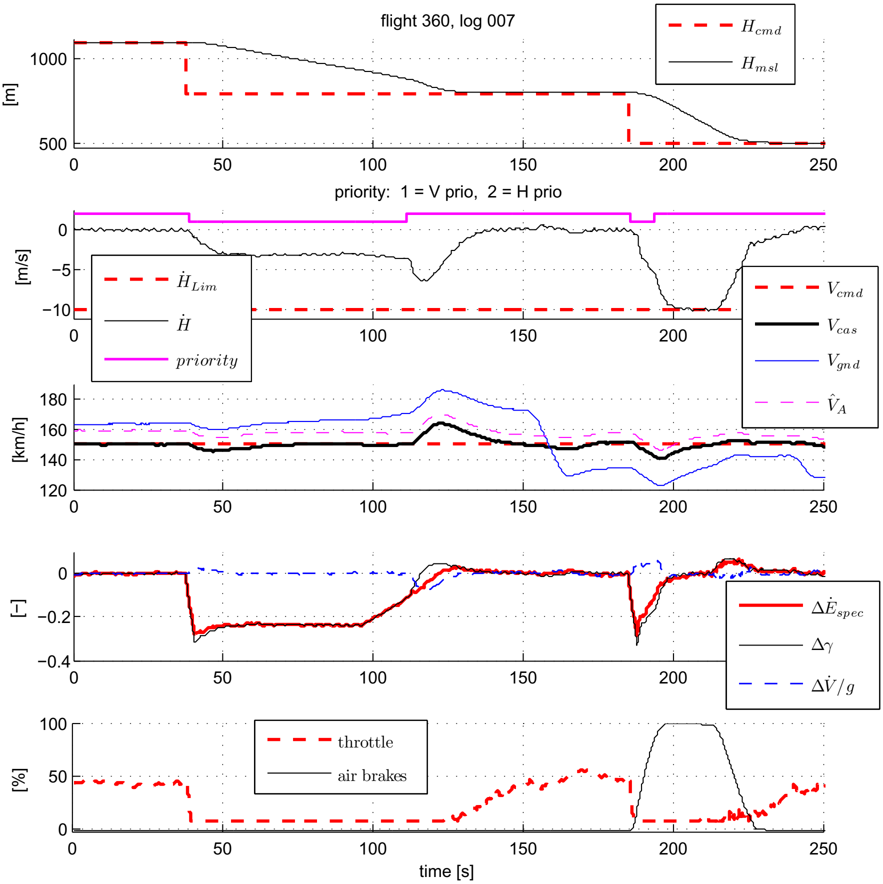


Fig. 15 Flight test results, comparison of sink performance with and without use of airbrakes

Here some work has to be done. The throttle was commanded in time to maintain the airspeed and the height. At t=150 s the aircraft makes a 90° left turn into the wind, which can be seen at the ground speed, which reduces from approximately 170 km/h to 135 km/h. During the turn, the height and the calibrated airspeed were maintained very well.

In the second descent, starting at t≈180 s, the use of the airbrakes was allowed. After the new height command, the FCL commanded the throttle to idle and extended the airbrakes to improve the sink rate. After switching to speed priority due to a detection of maximum speed protection, the flight continues with flight path priority, since the calculated value $\dot{V}_{cmd,res} = -\Delta\gamma \cdot g + \dot{V}$ falls below $\dot{V}_{max}$ due to reduction of $\Delta\gamma$. The vertical speed command of $\dot{H}_{Lim} = -10\,m/s$ was reached and well maintained without overshoot, the deviation between the commanded flight path and the flight path $\Delta\gamma$ goes to zero. The initial velocity error of 9 km/h is caused by the rapid reduction of the thrust and by the subsequent airbrake extension. After

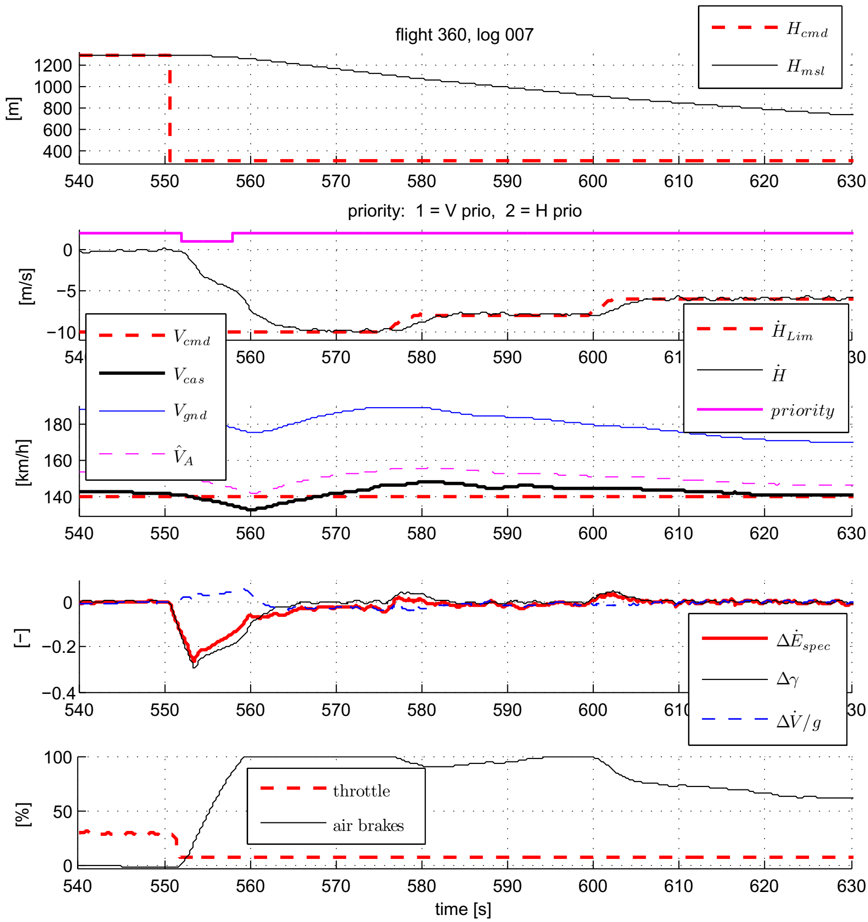


Fig. 16 Flight test results, variation of presettings for vertical speed

that the deviation in the velocity is slowly reduced. Shortly before reaching the commanded height, the airbrakes are retracted and the throttle is commanded in time. Since the sink rate was larger in the second descent than in the first, the descent time was shorter. As the descent took place with maximum airbrake deflection and with the throttle at its minimum limit, it demonstrates the maximum sink performance of the airplane for this configuration and this airspeed.

In another flight test, a descent was commanded ($\Delta H_{cmd} = -1000\,m$) with varying commands of vertical speed, see Fig. 16. The first vertical speed command was $\dot{H}_{Lim} = -10\,m/s$. This was reduced to $\dot{H}_{Lim} = -8\,m/s$ and after that to $\dot{H}_{Lim} = -6\,m/s$. The function which uses the airbrakes as control element was active. Directly after the descent command, the throttle was commanded to idle.

The descent begins with speed priority and continues with path priority, like in the flight test described before. The controlled aircraft follows the given vertical speed commands well. The airbrakes were commanded as needed. The error in the total energy rate was reduced rapidly. The error in velocity reduces slowly, while the tailwind reduces, which shows a comparison of the ground speed V_{gnd} and the calibrated airpseed V_{cas}.

7 Conclusion

The TECS concept, introduced by Lambregts, was taken up, modified and extended and tested in flight. The additional use of airbrakes proved to be an adequate possibility to manipulate the total energy of a aircraft and to extend the sink performance. This allows flying steeper vertical flight path missions. The function was validated in flight with satisfactory results. Improvements of speed and flight path accuracy during transient behaviour, i.e. transition from descent to horizontal flight without using airbrakes are under investigation.

References

1. Bruce, K.R.: NASA B737 Flight Test Results of the Total Energy Control System. In: AIAA Guidance, Navigation and Control Conference, AIAA 86-2143 CP (August 1986)
2. Bruce, K.R.: NASA B737 Flight Test Results of the Total Energy Control System, NASA Contractor Report 178285 (January 1987)
3. Bruce, K.R.: Integrated Autopilot/Autothrottle Based on a Total Energy Control Concept: Design and Evaluation of Additional Autopilot Modes, NASA Contractor Report 4131, (April 1988)
4. Bruce, K.R.: Integrated Autopilot/Autothrottle for the NASA TSRV B-737 Aircraft: Design and Verification by Nonlinear Simulation, NASA Contractor Report 4217 (1989)
5. Dalldorff, L., Luckner, R., Reichel, R.: A Full-Authority Automatic Flight Control System for the Civil Airborne Utility Platform S15 - LAPAZ. In: Euro GNC 2013 (2013)
6. Lambregts, A.A.: Vertical Flight Path and Speed Control Autopilot Using Using Total Energy Principles, AIAA Paper 83-2239 (August 1983)
7. Lambregts, A.A.: Integrated System Design for Flight and Propulsion Control Using Total Energy Principles, AIAA Paper 83-2561 (October 1983)
8. Lambregts, A.A.: Functional Integration of Vertical Flight Path and Speed Control using Energy Principles. In: 1st Annual NASA Aircraft Controls Workshop, October 25-27. NASA LRC (1983)
9. Lambregts, A.A.: Total Energy Based Flight Control System, World Intellectual Property Organization, International Publication No. WO 84/01345 (April 12, 1984)
10. Lambregts, A.A.: Total Energy Based Flight Control System, United States Patent No. 6,062,513 (May 16, 2000)
11. Lamp, M.: Flight Control Law Development for the Automatic Flight Control System LAPAZ. In: Euro GNC 2011 (April 2011)

12. Looye, G.H.N.: An Integrated Approach to Aircraft Modelling and Flight Contorl Law Design. PhD Thesis (2007) ISBN/EAN: 978-90-5335-148-2
13. Niedermeier, D., Lambregts, A.A.: Design of an intuitive Flight Control System. In: CAES 2009 (2009)
14. Ganguli, S., Balas, G.: A TECS Alternative using Robust Multivariable Control, AIAA 2001-4022
15. U.S. Department of Transportation, Federal Aviation Administration, Air Traffic Organization Operations Planning, Office of Aviation Research and Development, Washington, DC 20591. : Advanced Guidance and Control - Operational and Safety Benefits, Report No. DOT/FAA/AR-08/27 (June 2008)

TECS Generalized Airplane Control System Design – An Update

Antonius A. Lambregts

Abstract. The Total Energy Control System (TECS) was developed in the early eighties to overcome well known safety/design deficiencies of traditional Single Input/Single output (SISO) based Flight Guidance and Control (FG&C) systems. TECS uses generalized Multi Input/Multi Output (MIMO) based airplane control strategies to functionally integrate all desired automatic and augmented manual control modes and achieve consistently high performance for airplane maneuvering in the vertical plane. This paper documents further insights gained over the past years on TECS design details for achieving precision control decoupling, integration of augmented manual control modes, embedded envelope protection functions and innerloop design using airplane dynamic model inversion. Additionally, non-linear design aspects are discussed, including thrust limiting, energy management, maneuver rate limiting and mode logic.

1 Introduction

Automatic Flight Guidance and Control (FG&C) systems have evolved into highly capable systems. These systems have contributed immensely to the improvement of aviation safety. Unfortunately, these systems still use traditional SISO control strategies that do not provide full 6 degrees of freedom airplane control. Therefore, airplanes equipped with these systems are still vulnerable to Loss of Control (LOC). Furthermore, these systems have become exceedingly complex, due to an excessive number of modes, mode overlap and mode idiosyncrasies, making it a challenge for the flight crew to avoid mistakes using these systems that can jeopardize operational safety. Most of the FG&C system modes are considered "non-flight critical". This means that the flight crew is assumed to recognize and safely manage any failure of function of such modes. However, too often this assumption

Antonius A. Lambregts
Chief Scientific and Technical Adviser for Advanced Control,
Renton, WA, USA
e-mail: Tony.Lambregts@hotmail.com

has proven to be unwarranted. As a result there have been too many automation related incidents and accidents, due to stall, roll divergence after an engine failure, icing etc. The current generation of FG&C systems do not take full advantage of modern MIMO control and functional integration strategies provide simpler, more efficient and less costly designs.

1.1 TECS and THCS Development

FG&C system design and safety deficiencies were well recognized as long ago as the late seventies. In the early eighties NASA initiated research to address these deficiencies. This work resulted in the Total Energy Control System (TECS), which uses a generalized MIMO-based energy control strategy to functionally integrate all vertical flight path and speed control modes. This approach provides inherent envelope protection and avoids open ended SISO mode operations, thereby largely eliminating LOC safety risks. System complexities are reduced sharply by eliminating mode overlap, simplifying mode processing and providing more intuitive Man Machine Interfaces (MMI). Design generalization makes the system directly reusable, thereby reducing development costs for new applications. The system was successfully implemented and flight tested on the NASA B737 in 1985. The counterpart to TECS is the Total Heading Control System (THCS) which integrates all lateral directional control modes. Its design objectives and strategies are analogous to TECS. It was developed in the late eighties on the Condor High Altitude Long Endurance autonomous UAV program. TECS and THCS were successfully applied on the Condor and flight tested to demonstrate autonomous control capability under all operational and variety of failure conditions.

The basic TECS concepts are described in [1, 2]. This paper describes TECS design updates since the early nineties. A companion paper [3] describes design updates to the THCS design. It also describes a simplified TECS/THCS-based Mode Control Panel concept and a Primary Flight Displays concept that incorporates the TECS/THCS control and guidance strategies. Another companion paper [4] provides more details on Flight Envelope protection strategies.

1.2 Design Objectives

The TECS/THCS design objectives include:

- use of ***one*** pilot-like MIMO-based control strategy for all automatic and manual control modes,
- full envelope protection to prevent LOC
- generalized functionally integrated design, consistency of operation between modes
- energy-efficient vertical flight path/speed control (elimination of stand-alone Autothrottle)

- decoupled Mode Command responses, reduced controller activity
- reduced design complexity by elimination of function overlap and using modular design
- simpler, more intuitive Mode Control Panel (MCP), clearer Flight Mode Annunciation (FMA)
- large cost reductions by generalized/reusable design, minimal application-specific development, reduction in laboratory and flight testing and shorter application development cycle.

2 TECS Architecture and Conceptual Design

2.1 TECS Design

TECS uses a generalized MIMO based energy control strategy to provide all vertical flight path and airspeed control mode functions. Thrust is used to control the airplane's Total Energy requirement, the elevator is used to distribute the Total Energy between Potential and Kinetic energy.

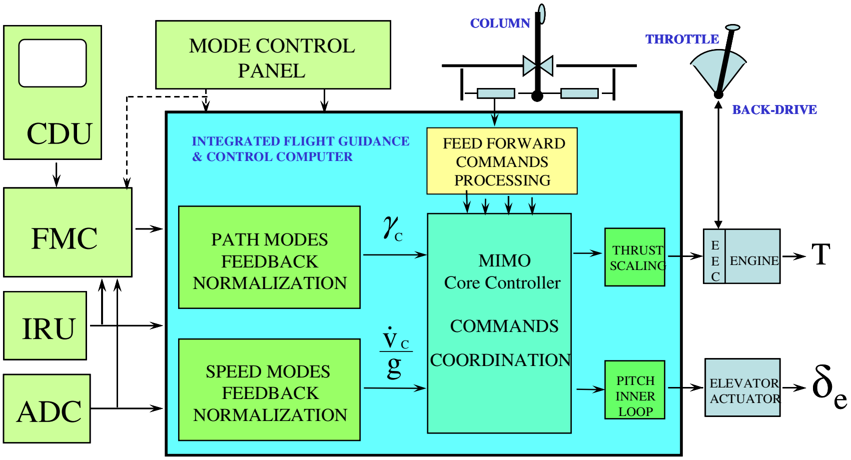


Fig. 1 TECS Architecture

A detailed discussion about the energy based control and design generalization strategy can be found in [1]. The general architecture is shown in Fig. 1.

Outerloop Modes Signal Processing

The outerloop speed mode error is first converted to a true airspeed error and this error is multiplied by a factor K_V / g to produce the non-dimensional longitudinal acceleration signal ($\dot{V}_c / g$). The outerloop path mode error (path deviation) is multiplied by a factor $K_h / \hat{V}_{true}$ to produce the non-dimensional flight path angle signal (γ_c). Speed and vertical path energy errors need to be weighted equally. Thus the gains K_V and K_h should have the same numerical value. The $\dot{V}_c / g$ and γ_c signals are used as the standard inputs to the TECS Core Controller, see Fig. 1. The $\hat{V}_{true}$ signal is a filtered true airspeed signal. This simple outerloop mode signal processing does not include integral control signal paths to assure transient free mode switching and avoid the need for mode integrator initialization logic.

Core Controller

In the Core Controller the associated error signals $(\dot{V}_\varepsilon / g)$ and (γ_ε) are formed. The sum of these error signals $(\gamma_\varepsilon + \dot{V}_\varepsilon / g)$ represents the airplane's specific Total Energy Rate error signal. This signal is used in an integral control signal path, together with $\hat{\dot{V}} / g$ and $\hat{\gamma}$ feedbacks used in proportional signal paths, to develop a normalized thrust command. Likewise, the difference between these error signals $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ represents the airplane's Differential Energy Rate error signal. In the original concept this signal was used in an integral control signal path, together with proportional $\hat{\dot{V}} / g$ and $\hat{\gamma}$ feedback signal paths, to develop the elevator command during operations with the thrust command between Tmin and Tmax. In the current design, explained below, γ_ε is used instead of $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ during operations with the thrust command between Tmin and Tmax. Because the error signals are used only in integral control signal paths, the control effectors respond to a step command from any of the outerloop modes with a control effector rate, resulting in smooth airplane dynamics.

Avoiding Outerloop Mode Tracking Errors

Since the integrators reside in the core controller, the feedback signals ($\hat{\gamma}$ and $\hat{\dot{V}} / g$), used in the TECS Core Controller, must be re-referenced in the low frequency range to the outerloop mode true airspeed and vertical path feedback signals respectively, to avoid possible outerloop mode command tracking offset due to bias errors in these feedback signals. This is done by using free running complementary filters, designed to take into account turbulence and windshear effects on system performance.

Command Response Decoupling

In order to achieve decoupled outerloop command responses, ***the Core Controller must be designed so that in response to a $\dot{V}_c / g$ or a γ_c command, the***

$(\gamma_\varepsilon + \dot{V}_\varepsilon / g)$ and $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ quantities go to zero simultaneously with identical dynamics.

Speed Envelope Protection – Automatic Modes

Generally, for most automatic mode operations, whenever the thrust command is at the upper or lower limit, a ***Speed-on-Elevator Control Priority*** (**SoECP**) is used to maintain the commanded airspeed. For those cases there is no need for separate speed envelope protection functions. However, for the Glide Slope mode a ***Path-on Elevator Control Priority*** (**PoECP**) strategy is used to handle conditions with thrust command at the upper or lower limit. This strategy assures the Glide Slope will be captured when the airplane is at the correct position to do so, often while the thrust is at idle. Momentary open loop speed responses are protected by the Vmin and Vmax control. Vmin/Vmax and Normal Load factor protection is also provided as an integrated part of the augmented manual control modes and are used to protect automatic mode operations for such rare events.

Normal Load Factor Protection – Automatic Modes

In the original TECS design, normal acceleration limiting was achieved by placing a rate limiting function on the γ_c and $\dot{V}_c / g$ signals. These rate limiting circuits introduce new system states which must be initialized at mode engagement and at any time the input to the rate limiter reverses direction, in order to avoid a response delays. To avoid this extra complexity an alternate method for normal load factor limiting, using amplitude limiting on (γ_ε) and $(\dot{V}_\varepsilon / g)$ was developed. This method is described below.

Energy Management during Execution of Simultaneous Flight Path and Airspeed Commands

The rate limiters on the γ_c and $\dot{V}_c / g$ signals (or the amplitude limiters on (γ_ε) and $(\dot{V}_\varepsilon / g)$ signals in the updated design) also provide effective rate limiting on the thrust and elevator commands. In addition, these functions provide efficient airplane energy management during execution of large simultaneous vertical flight path and airspeed commands. If the commands have opposing energy demands, the (γ_ε) and $(\dot{V}_\varepsilon / g)$ error signal inputs to the TECS Core thrust control channel will initially cancel, so thrust will stay constant until the elevator control has exchanged kinetic energy and potential energy to the extent possible. This depends on the relative amplitude of the speed and flight path mode commands. After the energy exchange is completed a change in thrust will be commanded to reduce any remaining airplane total energy error to zero. Thus, thrust is always used efficiently.

If the commands require a substantial energy change in the same direction, the thrust command will quickly go the upper or lower limit with double the rate limit of a single command. After the thrust command reaches the limit, a **SoECP** will be used and a Control Authority Allocation (CAA) amplitude limit is applied to the $(\dot{V}_c / g)$ signal. This CAA-limit, defined as $(\dot{V}_c / g)_{\text{limit}} = K_{em}(\gamma + \hat{\dot{V}} / g)$, effectively limits energy rate used to execute the speed command. (The quantity $(\gamma + \hat{\dot{V}} / g)$ represents the airplane's total energy rate.) Therefore the remaining part of the available energy rate will by default be used to satisfy the flight path command. For example, if during a climb at maximum thrust and therefore $(\gamma + \hat{\dot{V}} / g) > 0$ a value of $K_{em} = .5$ is selected, an accelerate command will be executed with half of the available energy rate. The other half is then allocated to continue the climb at ~half the initial rate. After the speed command is captured the climb rate will return to its original value, or if the altitude command is captured first, the acceleration will increase to capture the commanded speed, see the simulation results in Fig. 8. Similarly, if during idle descent when $(\gamma + \hat{\dot{V}} / g) < 0$ a value of $K_{em} = 1$ is selected, a deceleration command temporarily reduce the climb rate to ~zero to capture the commanded speed and then the idle descent rate will be reestablished. This strategy facilitates the operational requirement for reducing the airplane speed to 250 knots or less, before descending below 10,000ft. In the original design, the required logic for this Energy Management function was complex and not without flaws. These flaws and the fixes developed are discussed below.

2.2 TECS Performance in Turbulence and Windshear

Balancing the control command tracking performance and control effector activity for operation in turbulence and windshear conditions is a difficult problem for any flight control design. Reducing control effector activity inevitably results in deterioration of the command tracking in windshear. The performance objective used here for command tracking in a 1 knot/sec windshear is a peak IAS-error < 5 knots and a vertical path deviation $\Delta h < 20$ ft. The $\hat{V}$, $\hat{h}$, $\hat{\alpha}$, $\hat{\beta}$ filter gains are determined to achieve the preferred compromise between control effector activity in turbulence and reducing induced vertical and side acceleration and path deviation in windshear. It is beyond the scope of this paper to describe details related to choices of controller architecture, feedback signal processing and gains.

3 TECS Design Updates

3.1 Flight Path and Speed Control Decoupling – Revisited

From the start of the TECS and THCS development it has been a design objective to avoid ad hoc design and experimental tuning and instead use design solutions based on first principles of physics, whenever possible. Since then a better theoretical insight has been gained into achieving improved decoupling of the flight path response from the execution of a speed command. In the original TECS concept the differential energy rate error $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ was used as the input to the elevator control channel during operation with the thrust command between the lower and the upper limit. This conceptual architecture did not achieve the desired decoupling of the vertical path response during execution of an airspeed command. Therefore in the early TECS design and ad hoc method was used to improve path decoupling during speed command execution, with limited success. Later it was realized that an acceleration at constant flight path angle does not require an immediate change in pitch attitude, as is the case when changing flight path angle while maintaining airspeed. In fact, a permanent change in speed at 1 g flight requires a change in Angle of Attack and therefore an equal change in pitch attitude, which must be developed at the output of the integrator of the elevator control channel. Also, the change in angle of attack requires a change in the elevator trim, but this re-trim is accomplished in the updated design as part of the Short Period Model inversion, discussed below. Using this insight, the control strategy was changed from using $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ to using $\dot{V}_\varepsilon / g$ as the input to the elevator control channel, during operation with the thrust command between Tmin to Tmax. Also, a new $\hat{\dot{V}} / g$ signal with the gain K_{trim} is added to the input of the elevator control channel integrator, as shown in the revised Core Controller architecture of Fig. 2, to retrim the pitch attitude command during speed changes.

Since the integral of $(\hat{\dot{V}} / g).K_{trim}.KEI$ must equal the required change in pitch attitude, the value of K_{trim} is calculated using the 1 g relationship between Angle of Attack and true airspeed. Thus, it is found that $K_{trim} = (1/KEI).\{W/(\overline{q}.S)\}/(C_{L_\alpha}.\hat{V}_{true})$, where W = airplane weight, $\overline{q}$ = dynamic pressure, S is the airplane wing reference area and C_{L_α} = lift coefficient change per unit angle of attack change. In addition, the proportional signal path through the gain KEP has been revised to now use γ only, at all times by selecting K=2, see next section below. These changes do not alter the energy redistribution nature of the elevator control, but do tend to favor suppression of path control tracking errors over speed control tracking errors in turbulence and windshear.

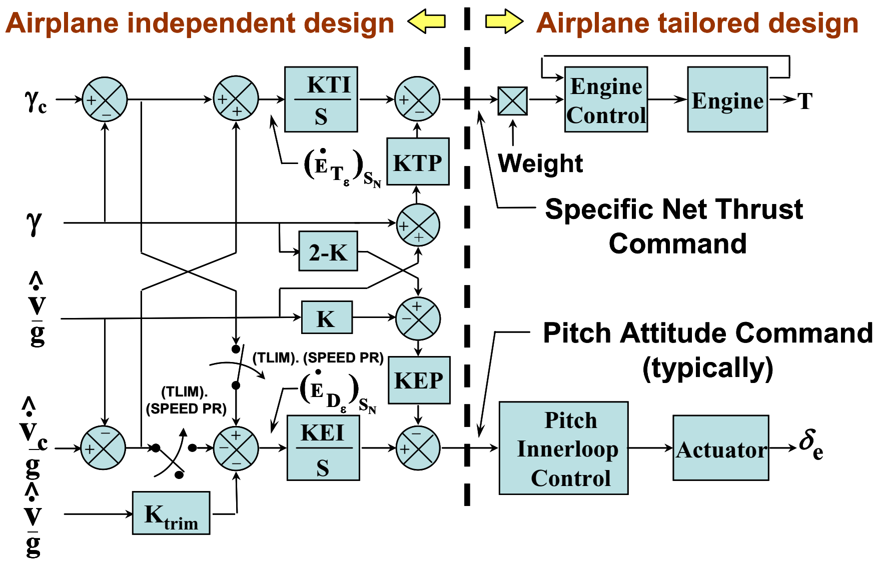


Fig. 2 TECS Core Controller Architecture

3.2 TECS Core – Elevator Control Channel Design

Classical Approach

The intent of the Energy Control strategy is to develop a priori coordinated thrust and elevator control commands, in order to decouple the outerloop flight path and speed command responses. If this is realized, the elevator and thrust control channels can be treated as independent decoupled SISO components of the system. So, the elevator control channel can be designed using the Short Period airplane pitch dynamics Model Inversion, together with the simple first order representation of flight path angle response to pitch attitude. The thrust control channel design can also be approached as a simple "SISO" Energy Control problem.

During the early TECS development root locus analyses was used to define the Core Controller gains and gain schedules to support flight path and speed control modes at all flight conditions. This approach provides little or no physical insight into the reason why and how the gains need to change. A better approach to gain insight is to use first principles of physics, including Model Inversion, feedback concatenation/normalization, Control Bandwidth Separation or Pole Placement. These concepts are discussed below.

Feedback Normalization/Concatenation, Pole Placement

Consider Fig. 3, representing a plant model of 3 chained integrators, with concatenated feedback loops/gains closed around each plant state.

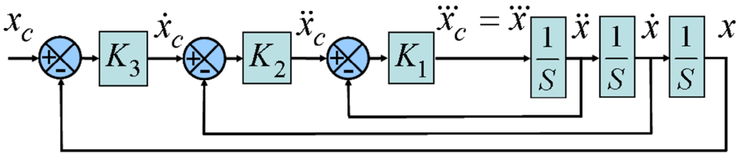


Fig. 3 Control Loop Concatenation

The Transfer Function (TF) for this system is:

$$\frac{x}{x_c} = \frac{K_1 K_2 K_3}{S^3 + K_1 S^2 + K_1 K_2 S + K_1 K_2 K_3} \tag{1}$$

Note that the TF for a similar system with any number of concatenated system states can be written directly. The normalized loop gains relate directly to the physical properties of the controlled system, such as natural frequency and damping. Then, if the gain associated with each feedback loop, starting from the innermost loop, is dropped by a factor 4 or greater, the poles of the x / x_c transfer function will all fall on negative side of the real axis in a root locus plot. Alternatively, considering (1), it is easy to use Pole Placement to achieve the desired dynamics and find the required gains.

Elevator Control Using Short Period Model Inversion

In our approach only the airplane rotational degrees of freedom are inverted. The Short Period Model Inversion and rebuilding of the airplane pitch dynamics is shown in Fig. 4. Note that pitch attitude (θ) can be used instead of Angle of Attack (α) to form the new desired pitch dynamics, as long as the frequency of the new augmented Short Period is selected to be in the frequency range where $\theta \approx \alpha$. Fig. 4 does not show the pitching moment due to thrust, but this effect is included in the full design and analyses and simulations. The Model Inversion approach used here is not more risky that a classical design approach, since the same conservative gain and phase margins will be maintained to provide robustness against airplane model errors and unmodeled dynamics. ***If the flight test results do not match the simulation results from the generalized control system design, there are only two possible causes: errors in the design implementation or, insufficient fidelity of the airplane and sensor models.*** In that case it is more productive to correct possible design and implementation errors and, if necessary, develop higher fidelity airplane models, rather than revert to an ad hoc “trial and error” approach. The resulting new Short Period dynamics are represented by the TF:

$$\frac{\theta}{\theta_c} = \frac{K_q K_\theta}{S^2 + K_q S + K_q K_\theta} \tag{2}$$

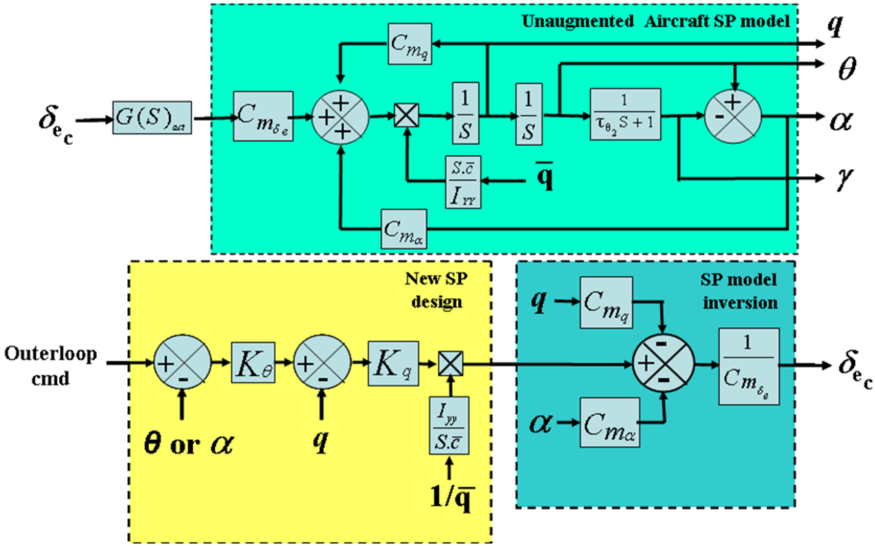


Fig. 4 Short Period Model Inversion and build up of new Short Period Dynamics

To provide the proper flight path angle control dynamics to support closing of the outerloop mode feedback, proportional γ feedback and integral control signal path of γ_ε are closed around the γ / θ_c transfer function, as shown in Fig. 5.

The γ_c coming from the automatic, or the augmented manual mode which develops γ_c by integration of the vertical control inceptor signal (δ_{vci}). [An alternate architecture using (γ_ε) in the proportional signal path has also been developed [6], but this architecture has disadvantages for the automatic control modes.] The controller structure of Fig. 5 was specifically developed to support the manual mode bandwidth requirement, by using pole zero cancellation to create effectively a lower order $[\gamma / \delta_{vci}]$ TF, as explained in section 4, below.

To make this possible the [γ / θ] heave dynamic have been made to appear as an explicit part of the $[\gamma / \gamma_c]_{auto}$ TF. By selecting $K_{EP} = \tau_{\theta_2}$ the TF for the automatic modes becomes (K_{FFI} and K_{FFP} both set to zero):

$$\left[\frac{\gamma}{\gamma_c}\right]_{auto} = \frac{\theta}{\gamma_c}.\frac{\gamma}{\theta} = \frac{1}{[\{1/(K_q.K_\theta.K_{EI})\}S^3 + \{1/(K_\theta.K_{EI})\}S^2 + (1/K_{EI})S + 1](\tau_{\theta_2}S + 1)} \tag{3}$$

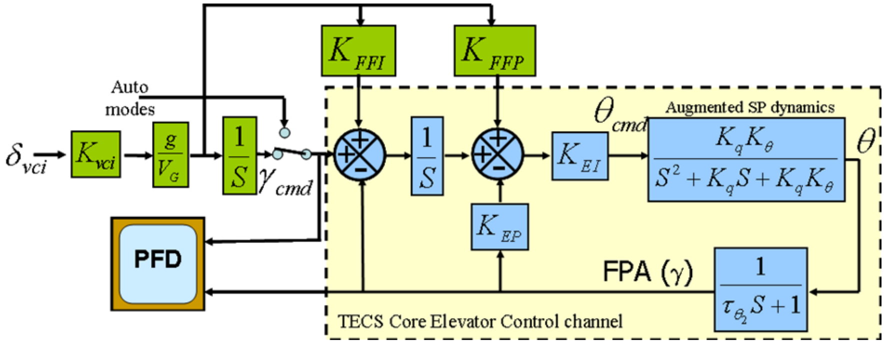


Fig. 5 TECS Core elevator control channel

Note that the variation of τ_{θ_2} no longer affects the system stability. The heave time constant is defined as:

$$\tau_{\theta_2} = \frac{C_L . V_{true}}{g . C_{L_\alpha}} = \frac{W . V_{true}}{g . C_{L_\alpha} . \bar{q} . S} \tag{4}$$

This τ_{θ_2} can be readily computed/updated by the onboard flight control computer.

3.3 *Normal Load Factor Limiting – Revisited*

Using γ_ε instead of the $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ as the input to the elevator control channel during operations with the thrust between the lower and upper limit also solved a Normal Load Factor (NLF) control issue. In the earlier design, during execution of simultaneous $\dot{V}_c / g$ and γ_c with opposite signs, the effective Normal Load Factor limit was twice the intended value, because **both** $\dot{V}_c / g$ and γ_c contributed to the effective NLF-limit. The new control strategy, using **either** γ_ε **or** $\dot{V}_\varepsilon / g$, eliminates this problem. As mentioned above, in the current design NLF control for the automatic modes is achieved by placing an amplitude limit on the γ_ε and $\dot{V}_\varepsilon / g$ signals. This amplitude limit is calculated as follows. Given a $\mathrm{NLF_{limit}}$, the normal acceleration limit is $g.\mathrm{NLF_{limit}}$. So the desired flight path angle rate limit becomes $\dot{\gamma} = \mathrm{NLF_{limit}}.g / V_G$. According to (3) for a ramping γ_c the flight path angle response will lag the command by an amount:

$$\tau_\gamma = 1 / KEI + \tau_{\theta_2} \tag{5}$$

Then, $\gamma_{\varepsilon} = \tau_{\gamma}.\dot{\gamma}$. Therefore the amplitude limit on γ_{ε} should be:

$$\gamma_{\varepsilon_{\text{limit}}} = (\text{NLF}_{\text{limit}}.g / \text{V}_{\text{G}})(1/KEI + \tau_{\theta_2}) \tag{6}$$

Also, the same limit needs to be applied to the $\dot{V}_{\varepsilon} / g$ signal, to achieve the same NLF control during operations with **SoECP** when the thrust command is at the upper or lower limit.

When the γ_{ε} or the $\dot{V}_{\varepsilon} / g$ signal at its limit, the feedback path to the integrator is effectively broken, so it must be shown that the remaining elevator/thrust control configuration maintains satisfactory γ and ΔT response dynamics.

3.4 TECS Core – Thrust Control Channel Design

The basic control decoupling requirement is met when the responses of $(\gamma_{\varepsilon} - \dot{V}_{\varepsilon} / g)$ and $(\gamma_{\varepsilon} + \dot{V}_{\varepsilon} / g)$ due to a γ_c or a $\dot{V}_c / g$ command are identical. The equation for $\Delta T_{required}$ is:

$$\Delta T_{required} = W.(\Delta\gamma + \dot{V} / g) + \Delta Drag \tag{7}$$

It can be shown that during automatic mode operations the drag change due to incremental NLF (which is limited to .1 g), can be neglected. From this equation it follows that ideally, in order to maintain $\dot{V}_{\varepsilon} / g = 0$ during the execution of a γ_c maneuver, the following TF identity must hold:

$$[(\Delta T / W) / \gamma_c]_{thrust} = [\gamma / \gamma_c]_{elevator} \tag{8}$$

Then by inference, in order to maintain $\gamma_{\varepsilon} = 0$ during execution of an acceleration command, the following TF identity will hold too:

$$[(\Delta T / W) / (\dot{V}_c / g)] = [(\Delta T / W) / \gamma_c] \tag{9}$$

The above decoupling requirements can be achieved by matching thrust control channel dynamics to mimic the elevator control channel dynamics. One approach is to select $K_{TI} = K_{EI}$ and $K_{TP} = K_{EP}$, and insert a matching filter before the final Thrust Command that represents the augmented pitch attitude and heave dynamics, as well as the inverted engine dynamics $[1/(T/T_C)]$. This approach and also another more traditional approach, matching the TF frequency/amplitude responses, were evaluated. Both approaches produced very good control decoupling results. However, the more traditional design resulted in a simpler design and achieved slightly better decoupling of the flight path and airspeed command responses. It also resulted in much lower gains K_{TP} and K_{TI} and therefore lower control activity in turbulence. Simplicity has its advantages.

Provisions for Thrust and Elevator Command Saturation

The classical way to limit the final thrust command is to continually calculate and apply the integrator limit by subtracting the contribution of the proportional $(\gamma + \hat{\dot{V}} / g)$ signal path from the externally provided Net Thrust Limit (NTL). Alternatively, the integrator may be moved to end of the net thrust command signal processing path, where its output can be simply limited to the engine's net thrust command limit, provided by the FADEC. In that case, a differentiator function must be placed in the proportional $(\gamma + \hat{\dot{V}} / g)$ signal path. In the later TECS designs the latter approach is used, because it is simpler.

Likewise, similar provisions must be implemented to prevent windup of the integrator in the elevator command processing signal path.

Priority Use of Elevator When the Thrust Command Is at a Limit

The change to using γ_ε instead of $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ as the input to the TECS Core elevator control channel during MIMO control, allowed the elevator control priority logic for conditions with thrust at the upper or lower limit to be simplified considerably. It also allows smooth, transient-free execution of simultaneous airspeed and flight path commands for all possible combinations of amplitude and timing to be achieved. The updated Elevator Control Priority works as follows. When the thrust command is within the linear control range between Tmin and Tmax, Path-on-Elevator Control Priority (**PoECP**) is used. **PoECP** also remains in effect after the thrust command reaches Tmax or Tmin when $\gamma_c \leq .5(\hat{\gamma} + \hat{\dot{V}} / g)$ and one of the following modes is engaged:

- the FPA mode , or
- the Altitude Acquisition/Hold mode, or
- the Glide Slope mode, or
- the Augmented Manual control mode, and the control inceptor is at neutral
- the Augmented Manual control mode, and the control inceptor is deflected and the Vmin/Vmax envelope protection control priority is not in effect

Using **PoECP** during operation in the Altitude Acquisition/Hold mode or the Glide Slope control mode when the thrust command is at the upper or lower limit is self evident. In that case the airplane will stay on the commanded flight path and accelerate/decelerate according to the available "excess energy rate". This strategy assures that the Glide Slope will be captured, when the airplane is at the right position for capture, either from below or from above the glide slope. When the thrust command reaches Tmin or Tmax, Speed-on-Elevator Control Priority (**SoECP**) is invoked in the following situations:

- Altitude Acquisition or FPA mode engaged and $T_c = T_{\max}$ and $\gamma_c > K_{ECP}.(\hat{\gamma} + \hat{\dot{V}} / g)$, or
- Altitude Acquisition or FPA mode engaged and $T_c = T_{\min}$ and $\gamma_c < K_{ECP}.(\hat{\gamma} + \hat{\dot{V}} / g)$, or
- Augmented Manual Control mode engaged and the control inceptor is at neutral, or
- Augmented Manual Control mode engaged and the control inceptor is not at neutral and either the Vmin control develops a more nose down command than the Manual FPA Control mode, or the Vmax control develops a more nose up command than the Manual FPA Control mode.

Currently, $K_{ECP} = .5$ has been selected. When $\gamma_c > K_{ECP}.(\hat{\gamma} + \hat{\dot{V}} / g)$ is not true (for small values of γ_c) speed commands will be executed using **PoECP**, e.g. when the Altitude Hold mode is engaged, or during a shallow climb in the FPA mode. When $\gamma_c > K_{ECP}.(\hat{\gamma} + \hat{\dot{V}} / g)$ is true, then there would be little control authority to accelerate using the thrust only, so in that case the **SoECP** is invoked and part of the energy rate used for climbing is transferred to accelerate the airplane and capture the commanded speed quicker. Examples of Energy Management cases are shown in Fig. 8. When **SoECP** is invoked the γ_ε signal input to the Core Controller is replaced with the $\dot{V}_\varepsilon / g$ signal, see Fig. 2. In that case, to allow for execution of a simultaneous flight path command, Control Authority Allocation (CAA) is applied to the longitudinal acceleration command $(\dot{V}_c / g)$. The CCA function is explained in more detail in the next section. Then, when the thrust command computation computes a thrust rate command that drives the thrust command out of the limit, the control priority reverts back to flight path control priority. The thrust coming off its limit always coincides with the start of the final flight path or speed command capture phase.

3.5 Energy Management during Execution of Simultaneous Flight Path and Airspeed Commands – Revisited

In the earlier TECS design the Energy Management function and associated logic was rather complex and not without flaws. One of the reasons was that the differential energy rate error $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ was used as the input to the elevator control channel during operations with the thrust command between Tmin and Tmax. When during the capture of a flight path command the thrust came off its limit, the Core elevator control channel would revert to using $(\gamma_\varepsilon - \dot{V}_\varepsilon / g)$ as its input signal. This made it difficult to smoothly capture smooth vertical path, if at that time the $\dot{V}_\varepsilon / g$ signal was not close to zero. The revised control priority logic discussed above, using **PoECP** during operations with the thrust command between Tmin and Tmax has made it possible to greatly simplify the Control Authority Allocation (CAA) associated signal processing. In the current design, the Core elevator control channel only uses the $\dot{V}_\varepsilon / g$ signal as its input when the SoECP is invoked. Therefore, in the current design, the $\dot{V}_\varepsilon / g$ signal that is routed the Core elevator control channel has the CCA amplitude limit [$K_{em}(\gamma + \hat{\dot{V}} / g)$] applied to it full time. No CCA amplitude limit is applied the $\dot{V}_\varepsilon / g$ signal that is routed the thrust control channel, so the thrust control channel always uses the basic $(\gamma_\varepsilon + \dot{V}_\varepsilon / g)$ signal. The CCA associated logic used in the earlier design has been eliminated.

3.6 TECS Automatic Modes Simulation Results

Simulation

A complete TECS/THCS system simulation capability was developed in MATLAB-Simulink. The simulation includes all TECS and THCS modes and design features discussed above in this paper, as well as a full flight regime six degrees of freedom nonlinear airplane simulation. Realistic 2nd order actuator models including rate and position limits were included, along with a rate limited 2nd order engine model. The system time responses below were generated using this simulation. No “design tuning” is used for any of the maneuvers shown below. The airplane model represents a generic 100-125 passenger twin turbofan engine transport airplane at 120,000 lbs.

Results

Fig. 6a and 6b show the airplane response to a 25 knots step command (upper plot) and a 100 knots step command (lower plot) in the IAS and Altitude Hold modes.

The responses are very smooth and the flight path coupling error due to the execution of the speed command is very small: the temporary altitude deviation was a little over 1 ft for the 25 knots speed increase (within the “linear” thrust operating range) and ~2 ft altitude deviation for the 100 knots speed change, which involved a 20 second period with the thrust command at the upper (Tmax) limit.

In Fig. 7 the system responses are shown for the IAS and Altitude Acquisition modes for step altitude commands of 500 ft (upper plot), and 5000 ft (lower plot). For both cases the responses are smooth and without an overshoot of the command. For the 500ft step command case the thrust does not reach the limit (Tmax).

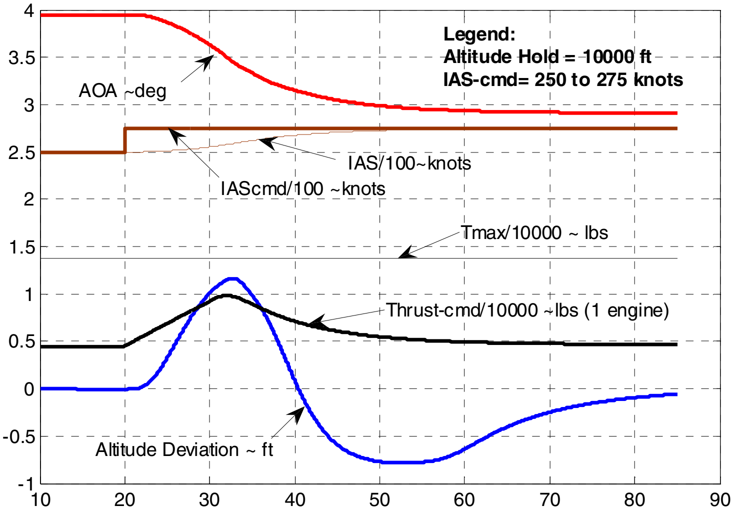


Fig. 6a TECS Responses to step IAS-cmd = 25 knots

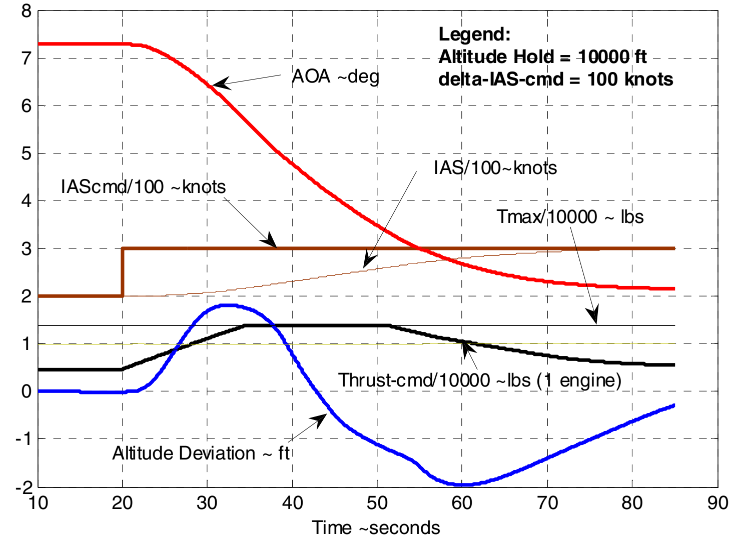


Fig. 6b TECS Responses to step IAS-cmd = 100 knots

For the 5000 ft step command case the thrust reaches Tmax and stays there for ~60 seconds before the thrust is reduced smoothly during the final linear exponential capture. There are no perceptible control transients resulting from the reversion from **PoECP** to **SoECP** and vice versa. In both cases the maximum speed deviation is limited to ~.2 knots. Notice also that the NLF is limited to .1 and that Tmax slowly decreases during the climb, due to the air density effect on the engine thrust.

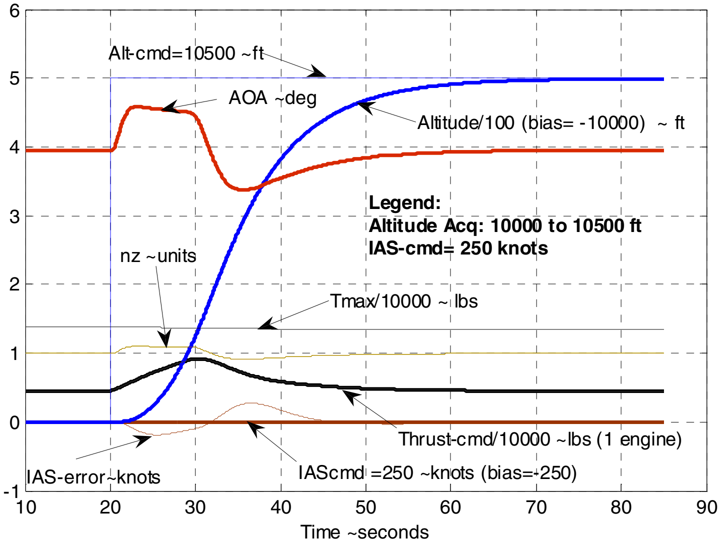


Fig. 7a Response to step Alt Acq-cmd: 10000 to 10500 ft

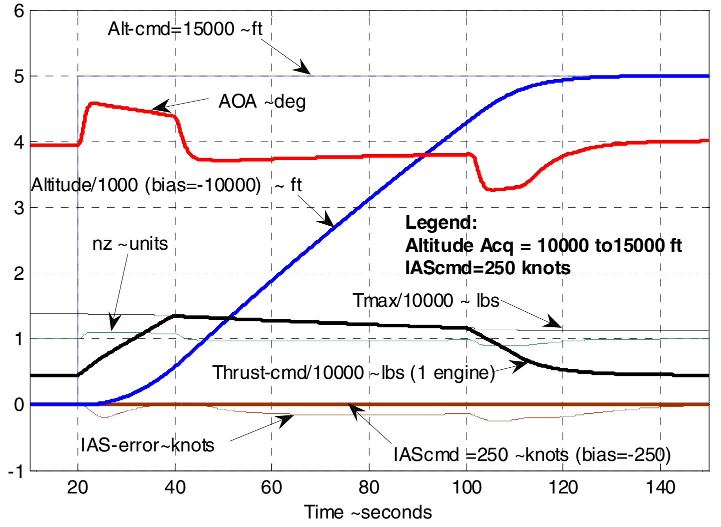


Fig. 7b Response to step Alt Acq-cmd: 10000 to 15000 ft

In Fig. 8 upper plot, the system responses are shown for the IAS and Altitude Acquisition modes for a step Alt-cmd = 3000 ft at t= 20 seconds and a step IAS-cmd = 100 knots at t=60 seconds.

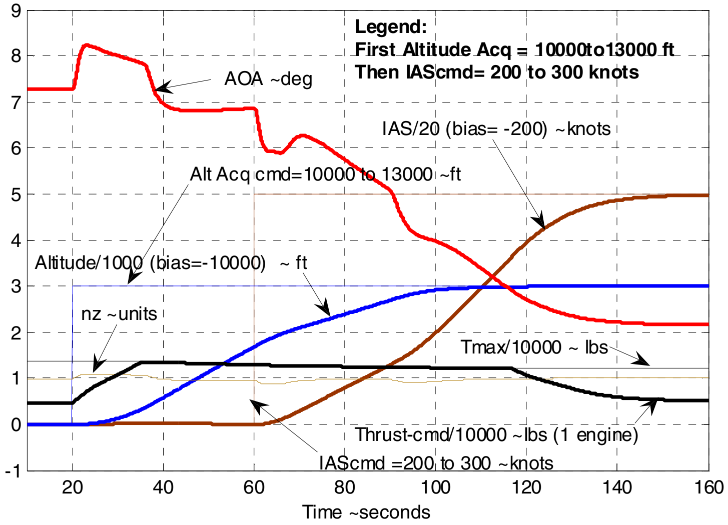


Fig. 8a Energy Management for Alt Acq-cmd = 10000 to 13000 ft, followed by IAS-cmd= 200 to 300 knots at t= 60 sec

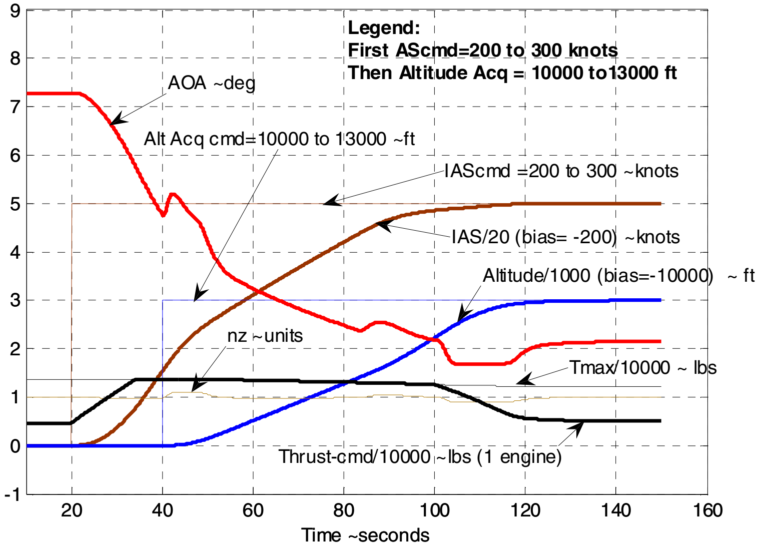


Fig. 8b Energy Management for IAS-cmd = 200 to 300 knots, followed by Alt Acq-cmd = 10000 to 13000 ft at t= 60 sec

The timing of the commands are chosen to demonstrate the “pilot like” Energy Management feature built into the system. The altitude command drives the thrust command to Tmax, establishing **SoECP**. Then when the IAS-cmd follows, the Control Authority Allocation (CAA) limit, $K_{em}(\hat{\gamma}+\hat{\dot{V}}/g)$ with $K_{em}=.5$, is placed on the $\dot{V}_c/g$ signal, causing the airplane to reduce the climb rate by ~50 % to execute the IAS-cmd. During the execution of the IAS-cmd the γ_c drops below $.5(\hat{\gamma}+\hat{\dot{V}}/g)$, causing the elevator control priority to revert from **SoECP** to **PoECP**, allowing the ALT-cmd to be captured. At this point the Total Energy demand is not yet satisfied, so the thrust stays at Tmax and the excess energy rate is now causing the acceleration to increase. Then, when the airspeed comes within the capture range, the thrust command drops below Tmax, **PoECP** is reestablished and the commanded airspeed is captured exponentially.

In Fig. 8 lower plot, the system responses are shown for the same step Alt-cmd =3000 ft and

IAS-cmd=100 knots, but order is reversed. Now the IAS-cmd causes the thrust to ramp up to Tmax, to accelerate the airplane while maintaining **PoECP**, until at t=40 seconds the step Alt-cmd results in $\gamma_c > .5(\hat{\gamma}+\hat{\dot{V}}/g)$, causing a reversal of the elevator control priority to **SoECP** with the CCA acceleration limit $(\dot{V}_c/g)_{\text{limit}} = .5(\hat{\gamma}+\hat{\dot{V}}/g)$ applied to the $\dot{V}_c/g$ signal that is routed to the elevator control channel. This causes the acceleration to drop to $\dot{V}_c/g = .5(\hat{\gamma}+\hat{\dot{V}}/g)$, thereby transferring ~50 % of the energy rate to the execution of the climb command, while maintaining Tmax. Next, the IAS-cmd is captured first, but since at this point still $\gamma_c > .5(\hat{\gamma}+\hat{\dot{V}}/g)$ the **SoECP** is maintained. During the IAS-cmd capture, the excess energy is transferred to increase the climb rate, while maintaining T= Tmax . When the thrust command drops below Tmax , the “linear” **PoECP** capture of the Alt-cmd begins.

4 Flight Path Angle Based Augmented Manual Control

4.1 Specific Design Objectives

The FBW Augmented Manual Control mode provides pilot maneuvering capability in the vertical plane by using the vertical control inceptors. A earlier standalone Flight Path Angle based augmented manual control mode was developed and flight demonstrated/evaluated under the NASA TCV Program 1976-79 [3]. Key design objectives include:

- good handling qualities at all flight conditions, with precision maneuvering capability
- reduced pilot workload using "Direct FPA Control" strategy , eliminating need for continuous compensatory pitch attitude control.
- carefree maneuvering capability to the limits of the safe airplane performance, using envelope protection to reduce the risk of Loss of Control (LOC)
- operational consistency with automatic modes, one pilot "mental model" for all operations
- clean and simple integration of the manual and automatic modes, shared function elements

In order to achieve the last three objectives the Augmented Manual control mode is designed as a simple augmentation to the automatic FPA control mode. Thus, the Core elevator control channel of Fig. 5 provides the basic airplane control when the pilot is not using the control inceptor. The airplane then maintains the last pilot-established earth referenced γ, regardless of changes in airspeed, airplane configuration changes or disturbances due to turbulence and windshear. As a result this FPA Rate Command Hold strategy largely eliminates the need for the pilot to use a ***Continuous Compensatory Control Strategy***. Instead the control tracking performance tends to improve when the pilot adopts an ***Intermittent Maneuver Control Strategy.*** So, the main reasons for going to a "Direct FPA Control" strategy are to reduce the tedious kind of workload controlling flight path perturbations, and to simplify interception and tracking a vertical path in space. This operation can be further enhanced by proper flight displays, e.g. a HUD or Synthetic Vision background display of the airport and runway. This makes it easy to capture a desired Glide Path and from there on, the FPA-based Augmented manual control algorithm will track the pilot established flight path with little or no need for pilot corrections. This capability, then called "Velocity Vector Control", was first developed and demonstrated by NASA under the TCV program in the late 1980-ties [5].

4.2 Design Implementation

For the design shown in Fig. 5 the control inceptor command signal is processed in three very basic feed forward command paths. The first signal path integrates the inceptor command to establish the reference γ_c . The second and third signal paths shape the control responses of the airplane to achieve the exact response dynamics prescribed by a specified ideal (classical) handling qualities model. Briefly, the TECS Core elevator control channel $[\gamma / \gamma_c]_{auto}$ TF, equation (3), has a unity numerator and a fourth order denominator. The feedback and feed forward gains of this Core Controller can be selected such that the resulting augmented manual $[\gamma / \delta_{vci}]$ TF results in a $[\theta / \delta_{vci}]$ TF that represents a specified ideal handling qualities model, for example:

$$\frac{\theta}{\delta_{vci}} = \frac{K_{vci}}{S}\frac{g}{V_G}\frac{(\tau_{\theta_2}S+1)\omega_{SP}}{S^2+2\zeta_{SP}\omega_{SP}S+\omega_{SP}^2} \tag{10}$$

Here δ_{vci} is the vertical control inceptor deflection, K_{vci} is the vertical control inceptor gain. Since

$$\frac{\gamma}{\delta_{vci}} = \frac{\theta}{\delta_{vci}}\frac{\gamma}{\theta} = \frac{\theta}{\delta_{vci}}\frac{1}{(\tau_{\theta_2}S+1)} \tag{11}$$

it follows in order to achieve (10) , the final $[\gamma / \delta_{vci}]$ TF must be

$$\frac{\gamma}{\delta_{vci}} = \frac{K_{vci}}{S}\frac{g}{V_G}\frac{\omega^2}{S^2+2\zeta\omega S+\omega^2} \tag{12}$$

Here V_G is the groundspeed. This $[\gamma / \delta_{vci}]$ TF can be realized by using feed the forward gains K_{FFP} and K_{FFI} to create two zeros designed to cancel two poles of the $[\gamma / \gamma_c]_{auto}$ TF, equation (3). One of these numerator zeros is used to cancel the τ_{θ_2} associated pole and the second zero is used to cancel the first order pole that is part of the third order part of the denominator of the $[\gamma / \gamma_c]_{auto}$ TF. Thus the "ideal" SP frequency and damping coefficient in (10) can be specified. For example: selecting $\omega = \omega_{SP} = 2$ rad/sec and $\zeta = \zeta_{SP} = 1$ results in:

$$\frac{\gamma}{\delta_{vci}} = \frac{K_{vci}}{S}\frac{g}{V_G}\frac{(K_{FFP}.S^2+K_{FFI}.S+1)}{(.25S^2+1S+1)(\tau_D S+1)(\tau_{\theta_2}S+1)} \tag{13}$$

The second order numerator of (13) must cancel the two first order poles. Therefore:

$$(K_{FFP}.S^2+K_{FFI}.S+1) = (\tau_D S+1)(\tau_{\theta_2}.S+1) \tag{14}$$

However, to determine K_{FFP} and K_{FFI}, τ_D must be known, or one of the feedback gains must be known. The simplest way is to select τ_D. Then from (14) it follows that $K_{FFP} = \tau_D.\tau_{\theta_2}$ and $K_{FFI} = \tau_D + \tau_{\theta_2}$. For example for $\tau_D = 1$, it follows that $K_{FFP} = \tau_{\theta_2}$ and $K_{FFI} = 1 + \tau_{\theta_2}$. Also the following identity must hold:

$$(.25S^2+1S+1)(\tau_D S+1) = [\{1/(K_q K_\theta K_{EI})\}S^3 + \{1/(K_\theta K_{EI})\}S^2 + \{1/(K_{EI})\}S+1] \tag{15}$$

The right hand part of equation (15) is the third order part of the original $[\gamma / \gamma_c]_{auto}$ TF, Equation (3). For $\tau_D = 1$ the gains become $K_q = 5$ (rad/sec^2)/(rad/sec), $K_\theta = 1.6$ (rad/sec)/rad and $K_{EI} = .5$ rad/rad. The gain $K_{EI} = .5$ adequately supports the outerloop altitude and airspeed modes bandwidth of .1 rad/sec. The γ response lag relative to γ_c becomes $\tau_\gamma = 1$ second.

The above sketched approach for designing the $[\gamma / \gamma_c]_{auto}$ and $[\gamma / \delta_{vci}]$ TF allows a quick evaluation of the change in the $[\gamma / \gamma_c]_{auto}$ dynamics and the gains, in particular K_q and K_{EI}, for other choices of τ_D. Here it was assumed that the linear elevator control actuator transfer function will have its lowest first order pole located at -20 rad/sec or higher. This allows for a gain K_q up to 5(rad/sec^2)/(rad/sec) or somewhat higher, while still assuring that the lowest frequency pole of the actuator dynamics will not couple with the first order pole associated with the K_q control loop, to form a lowly damped oscillatory mode. The selected gains also allows for the addition of structural mode filters, if needed. The robustness margins can be increased further by increasing τ_D which lowers K_q, but reduces K_{EI}. If the "ideal response model" is different than the one defined by equation (10), or if it needs to change for different flight conditions, it is a simple matter to recalculate the corresponding gains. More details on this FPA based Augmented Manual control mode design can be found in [3,4].

4.3 Augmented Manual Mode – Thrust Control

No changes to the basic thrust control channel are required for the Augmented manual mode, except the feed forward gains KT_{FFP} and KT_{FFI} (implemented analogous to K_{FFP} and K_{FFI}) can be used to minimize speed deviations due to vertical maneuvering. However, $K_{FFP} = 0$ was found to relax throttle response during vertical stick inputs, albeit at the expense of incurring a slightly larger speed error.

4.4 Pilot Display Requirements for Manual γ-Control Loop Closure

The FPA-based augmented manual control mode was designed to meet the classical handling qualities requirements intended for pitch attitude control, so a standard Primary Flight Display can be used for closing the pilot control loop using pitch attitude. However, to realize "direct FPA Control" requires the addition of the FPA information to the PFD. In the earlier NASA "Velocity Vector Control" development program [5] it was found that $\tau_\gamma = 1$ second, although less than for the unaugmented airplane, it is still too large for the pilot to be able to close the loop on γ directly. Therefore a quickened γ display signal must be used. The obvious candidate signal is γ_c. Pilot control loop closure around γ_c, instead of around the actual airplane dynamics, was found to work very well, since only 90 degrees lag is incurred in this loop, discounting the pilot's lag. So then the pilot can use very high gain without PIO risk. To avoid displaying both γ and γ_c, a blended γ_{quick} signal can be used that responds like γ_c during maneuvering and reverts to γ when the pilot is out of the control loop. It is based on equation (12):

$$\gamma_{quick1} = \gamma + \frac{(1/\omega^2)S^2 + (2\zeta/\omega)S}{(1/\omega^2)S^2 + (2\zeta/\omega)S + 1}\gamma_c \tag{16}$$

or, by defining $\tau_q = (2\zeta/\omega)$ a first order approximation of (20) becomes:

$$\gamma_{quick2} = \gamma + \frac{\tau_q S}{\tau_q S + 1}\gamma_c \tag{17}$$

Still another approach to "on demand" γ quickening was proposed in [6]. It adds a pitch rate signal to γ.

4.5 Augmented Manual Mode – Envelope Protection

Speed Envelope Protection. The FPA-based augmented manual control mode should normally be operated with the autothrust engaged because of the lack of speed stability at constant throttle setting. However the airspeed should be allowed to drift after the thrust reaches the upper or lower limit and the pilot commands a γ in excess of the airplane's steady state performance capability, or during maneuvering with the autothrust disengaged, as long as Vmin and Vmax protection is provided when the airplane's excess kinetic energy runs out. Therefore simple independent Vmin, and Vmax control functions have been developed that work as follows. **When the autothrust is engaged**, the Vmin control function is armed to allow engagement using **SoECP** after the thrust-command reaches Tmax and after the Vmin control develops a pitch command that is more nose down than the pitch command developed by the manual FPA control. Likewise, the Vmax control function is armed to allow engagement using **SoECP** after the thrust-command reaches Tmin and after the Vmax control develops a pitch command that is more nose up than the manual FPA control. Also, the Vmin target is lowered in proportion to the nose up δ_{vci} deflection, from the command speed at zero δ_{vci} deflection to $1.05 V_{stall_\varphi}$ for full nose up deflection. Likewise, the Vmax target is increased in proportion to the nose down δ_{vci} deflection, from the command speed at zero δ_{vci} deflection to Vmo/Mmo + XX knots for full nose down deflection. **When the autothrust is disengaged**, the Vmin/Vmax envelope protection function is always armed to engage. In this case the Vmin target is 1.2 V_{stall_φ} at zero δ_{vci} deflection. The Vmin target is lowered in proportion to the nose up δ_{vci} deflection to $1.05 V_{stall_\varphi}$ for full nose up deflection. Here $V_{stall_\varphi} = V_{stall_{1g}} \sqrt{1/\cos\varphi}$ is the stall speed for the airplane in a level coordinated banked turn. Likewise, the Vmax target is increased in proportion to the nose down δ_{vci} deflection, from Vmo/Mmo at zero δ_{vci} deflection to Vmo/Mmo + XX knots for a full nose down deflection.

Normal Load Factor Control. To prevent excessive positive n_z that can result in stall or undesirable negative n_z, both a bank angle command limit and a full δ_{vci} deflection n_z - command limit are imposed. The vertical maneuver authority, $(n_{z_v})_{\max}$, is calculated according to equation (18) below, and the NLF-command is scheduled so that a full vertical control inceptor deflection always commands the maximum safe NLF. However, it was found that simple command limiting cannot prevent exceeding the NLF limits for extreme stop to stop vertical inceptor deflections at high speed (e.g. due to PIO). Therefore a simple innerloop feedback NLF-limit control function was also implemented. A more detailed discussion on Envelope Protection requirements and design for automatic and augmented manual control mode can be found in the companion paper [4].

4.6 Scheduling of the Command Gain, K_{vci}

To minimize the possibility of overstressing or stalling the airplane, the vertical control inceptor command gain K_{vci} needs to be scheduled as a function of stick deflection and airspeed, such that full vertical stick deflection commands the maximum safe Normal Load Factor (n_z), at any speed. At speeds greater than the Maneuver Speed $n_{z_{authority}} = n_{z_{structLim}}$. Generally at design weight, $n_{z_{structLim}} = 2.5$. Below the Maneuver Speed $n_{z_{authority}} = n_{z_{aeroLim}} = V^2 / V^2_{stall1g}$.

The vertical control inceptor deflection (δ_{vci}) is normalized to +1 for full nose up deflection and -1 for full nose down deflection. For $\delta_{vci} = 1$ The maximum available normal load factor for vertical maneuvering, $(n_{z_v})_{\max}$, is:

$$(n_{z_v})_{\max} = n_{z_{authority}} - \Delta n_{z_\varphi} - \Delta n_{z_{stall\text{Margin}}} \qquad (18)$$

In equation (19) $n_{z_{authority}}$ is the lower of $n_{z_{structLim}}$ or the $n_{z_{aeroLim}}$; $\Delta n_{z_\varphi} = (1/\cos\varphi) - 1$ is the incremental load factor due to roll angle, assuming a coordinated turn; $\Delta n_{z_{stall\text{Margin}}}$ is a selected safety margin, typically equal to .1. For this study it was decided that the negative n_z control authority should be limited to $(n_{z_v})_{\min} = 0$, rather than $(n_{z_v})_{\min} = -1$, because a capability to command $(n_{z_v})_{\min} = 0$ in a vertical maneuver gives plenty maneuver authority for a transport airplane and protects the passengers and the airplane against possible injuries and damage. (An arrangement should be provided to change this limit to $(n_{z_v})_{\min} = -1$ in case the airplane becomes inverted, since the airplane must remain controllable at any attitude.) Thus, with the n_{z_c} defined at three points for δ_{vci} =1, 0, -1, the above requirements can be met by defining the incremental normal load factor $(\Delta n_{z_{cv}})$, commanded as a function of $(n_{z_v})_{\max}$ and δ_{vci}, using the following parabolic schedule:

$$\Delta n_{z_{cv}} = [\{.5(n_{z_v})_{\max} - 1\}.\delta_{vci} + .5(n_{z_c})_{\max}].\delta_{vci} = K_{vci}.\delta_{vci} \qquad (19)$$

Therefore:

$$\dot{\gamma}_c = \Delta n_{z_{cv}}.g / V_G. = K_{vci}.\delta_{vci}.g / V_G \qquad (20)$$

Here $\Delta n_{z_{cv}}$ is the incremental load factor commanded by the vertical control inceptor deflection. It should be noted that for this schedule the command gradient, $(\Delta n_{z_{cv}} / \delta_{vci})_{\delta_{vci}=0} = .5(n_{z_v})_{\max}$, is a function of airspeed and the inceptor deflection. When combined with a passive inceptor that has a fixed force gradient, it produces a "stick force per g" $(F_{vci} / \Delta n_{z_{cv}})$ that at high speeds decreases with increasing deflection and this is generally regarded as unacceptable for handling qualities. Another stick command schedule that allows for a selectable command gradient, $\Delta n_{z_{cv}} / \delta_{vci})_{\delta_{vci}=0} = grad$, can be defined using a (1-cos) function and the requirements: for $\delta_{vci} = 1$ $\Delta n_{z_{cv}} = (n_{z_v})_{\max} - 1$; for $\delta_{vci} = 0$

$\Delta n_{z_{cv}} = 0$; and for $\delta_{vci} = -1$ $\Delta n_{z_{cv}} = -1$. Then for $\delta_{vci} \geq 0$ the relationship becomes:

$$\Delta n_{z_{cv}} = \delta_{vci}.grad + \{(n_{z_v})_{\max} - 1 - grad\}[1 - \cos\{\delta_{vci}.(\pi/2)\}] \tag{21}$$

and for $\delta_{vci} < 0$ the relationship becomes:

$$\Delta n_{z_{cv}} = \delta_{vci}.grad + (-1 + grad).[1 - \cos\{\delta_{vci}.(\pi/2)\}] \tag{22}$$

Although for this command the gradient at zero control effector deflection can be selected, at high speed the $F_{vci} / \Delta n_{z_{cv}}$ gradient still decreases with increasing deflection. So this schedule may also be unsuitable for use with a passive control inceptor that has a fixed force/deflection gradient. Still another alternative is to use a constant command gradient, $(\Delta n_{z_{cv}} / \delta_{vci})$, at any defection and airspeed. This approach would produce a constant $F_{vci} / \Delta n_{z_{cv}}$, when using a passive constant force gradient control inceptor, but it also has a number of design and handling qualities issues. These include: matching full deflection command with maximum maneuver authority; unequal maximum positive and negative deflection; command gradient discontinuity around zero inceptor deflection and possibly for large nose up deflection; and possible need for flat zones where $\Delta n_{z_{cv}} / \delta_{vci} = 0$, used to prevent exceeding $(n_{z_v})_{\max}$. For these reasons, the FAA is currently sponsoring research to define design guidelines and certification requirements for passive and active control inceptor command gain and feel force gradient.

4.7 Control Anticipation Parameter (CAP) Requirement

The CAP is defined as the ratio $\ddot{\theta}_{t=0} / \Delta n_{z_{t=\infty}}$. From the $[\gamma / \delta_{vci}]$ TF defined by (14), it follows that:

$$\ddot{\theta}_{c_{t=0}} = \delta_{vci}.K_{vci}.(g/V_G).K_{FFP}.K_{EI}.K_{\theta}.K_q \tag{23}$$

For a constant stick input the final incremental load factor is:

$$(\Delta n_{z_c})_{t=\infty} = (V_G / g)\dot{\gamma}_c = (V_G / g).\delta_{vci}.K_{vci}.(g/V_G) \tag{24}$$

Therefore

$$CAP = (g/V_G).K_{FFP}.K_{EI}.K_{\theta}.K_q \tag{25}$$

Thus, for an approach condition with $V = 215$ ft/sec and $\tau_{\theta_2} = 2$ and a selected $\tau_\gamma = 1$, it follows that the required value $K_{FFP} = \tau_{\theta_2} = 2$, resulting in CAP = 1.2. This is well within the .28 to 3.6 CAP range allowed in MIL-STD-1757A for level 1 handling qualities. With the feedback gains K_{EI}, K_θ and K_q pre-determined, the only parameter that can change CAP is K_{FFP}. However, K_{FFP} is a critical design parameter that cannot be varied much from the calculated value in the above analysis and still achieve acceptable responses. Furthermore, to achieve harmony between the initial pitch acceleration and the final n_z response K_{FFI} (here controlling the steady state n_z lag relative to $\delta_{vci}.K_{vci} / S$ must be selected within a narrow range, to achieve the desired value of τ_γ. Another way to analyze CAP is to look the variation of CAP as a function of τ_D. Changing τ_D has no effect on the final $[\gamma / \delta_{vci}]$ TF , nor does it change the product $K_{EI}.K_\theta.K_q$, but K_{FFP} varies in proportion to τ_D, so it is possible to change CAP value without changing the $[\gamma / \delta_{vci}]$ response! One more observation: for higher values of τ_D the basic $[\gamma / \gamma_c]_{auto}$ TF incurs more lag, because it reduces K_{EI} (see equation (5), so in order to still achieve the same final γ / δ_{vci} response, the feed forward gains and K_{FFI} increase to compensate for the increased lag. So then the control augmentation relies more on the K_{FFP} direct feed through signal path to the elevator and less on the feedback control signal paths. Conclusion: CAP is a dubious Handling Quality criterion that may need further updates or replacement.

4.8 FPA Based Augmented Manual Control Simulation Results

The same simulation as used above for the automatic modes was used to generate the time responses below for the FPA-based Augmented Manual control mode as defined above. Fig. 9a shows the responses to $\delta_{\text{stick}_{pitch}}$ =.1, starting at t=20 second for a duration of 5 seconds, resulting in a change of flight path angle of ~2 degrees. The responses are very smooth, without an overshoot of the γ_c, or oscillations. Note the responses in Angle of Attack (AOA) and NLF (n_z) are also controlled very smoothly. The flight path angle response lag is as designed: $\tau_\gamma = 1$ second. The pitch attitude shown is biased by the amount of the trim pitch attitude before the maneuver starts, to show its lead relative to lead γ_c .

The pitch attitude leads the γ-response by τ_{θ_2} and causes the attitude to drop back when the control inceptor is released, if $\tau_{\theta_2} > 1$. The IAS-error remains less than 1knot. The thrust command does not reach Tmax.

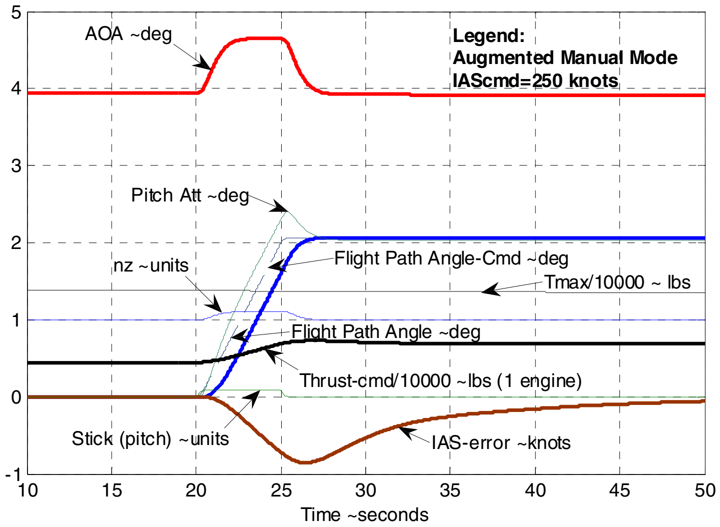


Fig. 9a FPA-based Augmented Manual Mode: responses to δ_{vci} =.1 from t= 20 to 25 sec

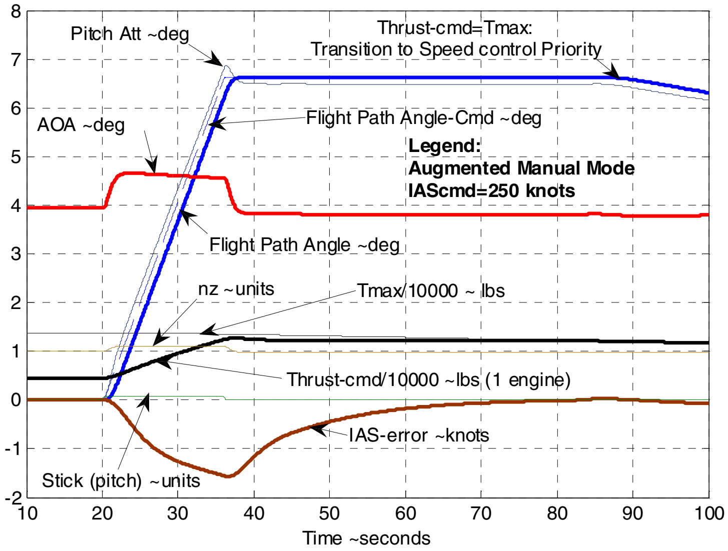


Fig. 9b FPA-based Augmented Manual Mode: responses to δ_{vci} =.1 from t = 20 to 36 sec, Transition to SoECP

Fig. 9b shows the responses to δ_{vci}=.1, starting at t=20 second for a duration of 16 seconds, resulting in a change of flight path angle of ~6.6 degrees. During this maneuver the thrust command increase to within a small margin of Tmax. The IAS-error reaches a maximum of ~1.5 knot and then gradually reduces to ~zero. Then, as Tmax decreases with increasing altitude, the thrust command reaches Tmax at t=87 seconds and this causes a reversion from **PoECP** to **SoECP**, in order to maintain the airspeed. At that point the flight path angle will start to fall off in proportion to the thrust fall off.

Fig. 10 shows the responses to δ_{vci}=.1, starting at t=20 second held indefinitely. This causes the flight path angle to rise until the Vmin control engages (using **SoECP)**, a short time after the thrust command reaches Tmax. At that point the flight path angle and reaches ~ 9 degrees, then starts to decline as a result of the Vmin control priority. As discussed above, the Vmin control has been designed to mimic speed stability, allowing a final speed deviation in proportion to the δ_{vci} deflection. So in the upper plot, the speed bleeds of 10 % of the speed margin to 1.05Vstall.

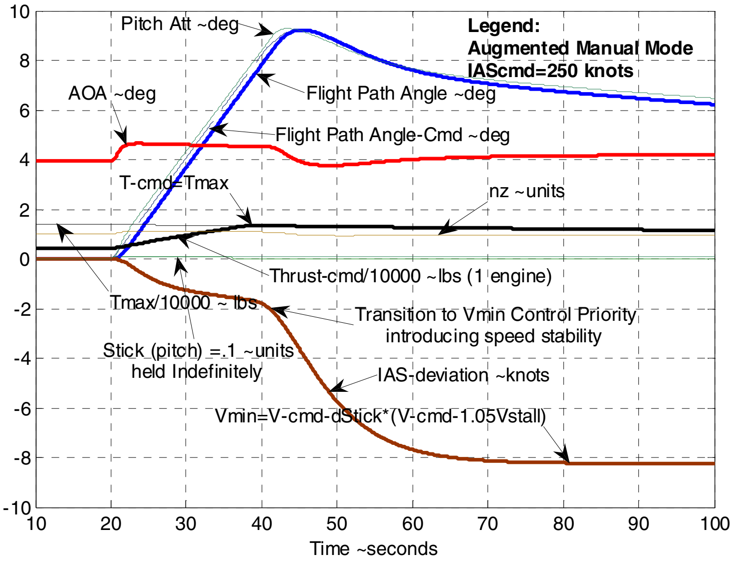


Fig. 10 FPA-based Augmented Manual Mode, responses to δ_{vci}=.1 held indefinitely

In the Fig. 11 a full nose up deflection is applied (δ_{vci}=1). This results in a very aggressive maneuver using all available NLF authority. For this case, the thrust command very quickly reaches Tmax and Vmin control priority is established very shortly before the NLF and the AOA reach their peak value, at ~2.2 and 13 degrees respectively. The stall AOA is 15 degrees. The flight path angle reaches a peak value of ~22 degree. The real peak pitch attitude reached is ~38 degrees (34 degrees as shown + 4 degrees trim value before the start of the maneuver). The final speed settles at 1.05Vstall. The control responses and reversion to Vmin control are very smooth and without transients.

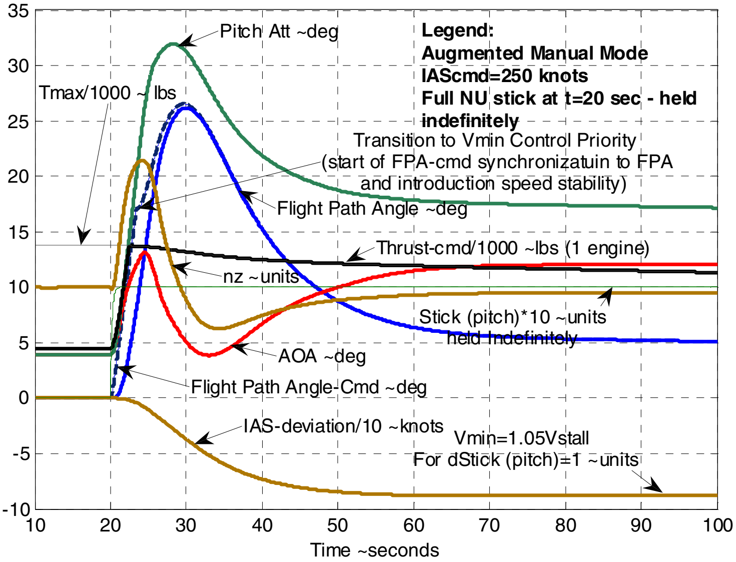


Fig. 11 FPA-based Augmented Manual Mode, responses to δ_{vci} =1 held indefinitely

5 Additional TECS and THCS Related Developments

5.1 Ecological PFD

The TECS energy based control strategy and the THCS Heading control strategy can be embedded into the Primary flight Display (PFD), to bring out control guidance cues for using manual Thrust, Pitch and Roll control to efficiently and simultaneously capture and track airspeed, Altitude and heading targets in an energy efficient exponential and overshoot-free manner. This enhanced Ecological PFD concept is summarized in the companion paper [3] and described in more detail in [8].

5.2 TECS/THCS Mode Control Panel

In order for the pilot to be able to use one mental model for all FG&C operations and to minimize effort needed for reuse, the TECS & THCS designs use one generalized guidance and control strategy for all automatic and augmented manual control modes. For the same reason this strategy has also been applied to the design of the FG&C Mode Control Panel (MCP) and Flight Mode Annunciation (FMA) function on the FFD. These developments are described in the companion paper [3].

5.3 Interactive Real-Time TECS/THCS Demonstration System

An interactive Real-Time TECS/THCS Demonstration System was develop, using The Simulink Real Time Workshop program, including the twin engine transport airplane simulation, an interactive TECS/THCS Mode Control Panel with integrated Controller Pilot Data link Communication functions, a joystick manual control capability and several versions of Primary Flight Displays including a Flight Mode Annunciation Panel.

6 Conclusion

This paper describes recent design enhancements of the Total Energy Control System (TECS). TECS uses a “pilot like” MIMO energy-based guidance and control strategy to generalize and functionally integrate all automatic and augmented manual control modes for airplane control in the vertical plane. This design strategy enables the pilot to use one mental model for all FG&C operations. It also minimizes the effort needed for design application on various airplane programs. To limit the scope of this paper, additional related subject matter is covered in two companion papers [3] and [4]. Companion paper [3] covers design updates to the Total Heading Control System (THCS), the enhanced ecological PFD [also called Energy Management PFD (EMPFD)], and the design of the Mode Control Panel and the Flight Mode Annunciation Panel. THCS uses strategies analogous to TECS, to generalize and functionally integrate all automatic and augmented manual lateral-direction control modes. TECS and THCS provide full 6 degrees of freedom airplane control capability to the limits of the safe flight envelope, without allowing LOC by departure outside the safe flight envelope. It eliminates stand alone SISO based Autothrottle, Yaw Damper/ Turn Coordination and Trust Asymmetry Compensation systems. The EMPFD incorporates the TECS and THC guidance strategies to enhance pilot awareness of airplane maneuver capabilities and provide guidance cues to use the controls in an effective and energy efficient manner. A second companion paper [4] discusses various options for designing flight envelope protection functions.

References

1. Lambregts, A.A.: Vertical Flight Path and speed Control Autopilot design Using Total Energy principles, AIAA 83-2239CP
2. Lambregts, A.A.: Automatic Flight Controls Concepts and methods. Koninklijke Nederlandse Vereniging voor Luchtvaart, Jaarverslag (1996)

3. Lambregts, A.A.: THCS Generalized Airplane Control System Design. In: 2013 CEAS Conference on Guidance, Navigation and Control, Delft, The Netherlands (2013)
4. Lambregts, A.A.: Flight Envelope Protection Strategies for Automatic and Augmented Manual Control. In: 2013 CEAS Conference on Guidance, Navigation and Control (2013)
5. Lambregts, A.A., Cannon, D.G.: Development of a Control Wheel Steering Mode and Suitable Displays that Reduce Pilot Workload and Improve Efficiency and Safety of Operation in the Terminal Area and in Windshear, AIAA G&C paper 79-1887
6. Lambregts, A.A.: Fundamentals of Fly-By-Wire Augmented Manual Control, SAE 05WAC-62
7. Niedermeier, D., Lambregts, A.A.: Design of an Intuitive Flight Control System. In: 2009 CEAS Conference (2009)
8. Bray, R.: A Head-Up Display Format for Transport Aircraft Approach and Landing, NASA TM-81199; NASA HUD Report 11; N80-29295 (July 1980)
9. Lambregts, A.A., Rademaker, R., Theunissen, E.: A New Ecological Primary Flight Display Concept. In: DASC 2008 (2008)

Integrated Modelling of an Unmanned High-Altitude Solar-Powered Aircraft for Control Law Design Analysis

Andreas Klöckner, Martin Leitner, Daniel Schlabe, and Gertjan Looye

Abstract. Solar-powered high-altitude unmanned platforms are highly optimized and integrated aircraft. In order to account for the complex, multi-physical interactions between their systems, we propose using integrated simulation models throughout the aircraft's life cycle. Especially small teams with limited ressources should benefit from this approach. In this paper, we describe our approach to an integrated model of the Electric High-Altitude Solar-Powered Aircraft ELHASPA. It includes aspects of the environment, flight mechanics, energy system, and aeroelasticity. Model variants can be derived easily. The relevant parts of the model are described and the model's application is demonstrated.

1 Introduction

Solar-powered aircraft have received increasing interest by the scientific and industrial community in the past fifty years. The subject of early solar flight is summarized in depth by Boucher[3]. Voit-Nitschmann[21] and Noth[14] extend the overview and the latter also includes a list of flown solar aircraft with their basic characteristics. Recent developments of the Solar Impulse[20] and the QinetiQ Zephyr[15] prove that the technology is ready to allow for continuous solar-powered flight.

However, solar-powered and high-altitude aircraft rely on highly optimized aircraft design. This results in high lift configurations, fragile light-weight construction and restrictive power reserves. These aircraft are thus constantly operated at their physical limits. All aspects of the systems work in a highly integrated manner and influence one another in a way, which is difficult to predict. Examples of the interactions can be found between flight dynamics and the energy system or between the

Andreas Klöckner · Martin Leitner · Daniel Schlabe · Gertjan Looye
German Aerospace Center (DLR), Robotics and Mechatronics Center,
Institute of System Dynamics and Control, Oberpfaffenhofen, D-82234 Weßling, Germany
e-mail: andreas.kloeckner@dlr.de

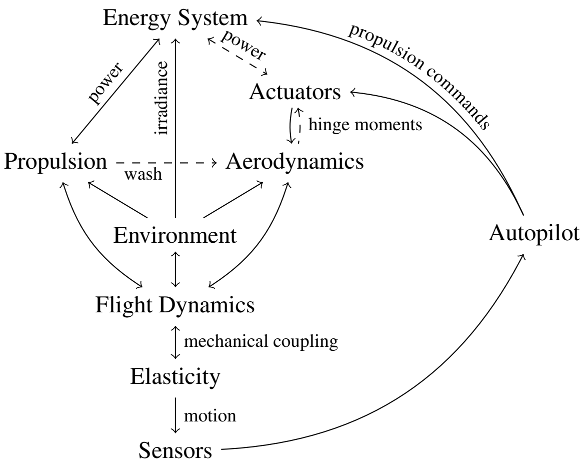


Fig. 1 A multitude of interactions must be taken into account, when developing or operating high-altitude solar-powered aircraft. The modules of the simulation are shown together with the mutual influences. Arrows indicate causality. Examplary couplings are labelled with the type of influence. Dashed arrows indicate interactions, which are not covered by the present model.

flight dynamics and the elastic modes of the aircraft. When an autopilot feedback loop is introduced, the system becomes even more complex. Fig. 1 summarizes the interactions to be taken into account, when developing or operating high-altitude solar-powered aircraft.

The high level of integration of all these modules renders the development, simulation and operation of high-altitude solar-powered aircraft a difficult task. Thus, we propose to use integrated models during the whole life cycle of solar aircraft. They provide faster and more accurate feedback to the design team of the aircraft than separate models, which are traditionally used. Refined versions of the models can later be used for more realistic simulation, training, control design and hardware-in-the-loop tests. An integrated model structure finally also provides consistent modelling of all parts of the system and for all design and operational analyses to be carried out. These features are especially valuable for small teams with limited ressources.

In order for this approach to work, the modelling technique must be suitable for a variety of physical fields. The resulting models must be fast enough for closed-loop simulation. Also, model versions of differing level of detail and reduced degrees of freedom must be derived with little effort. The approach of the present study makes use of Modelica [6] and the Flight Dynamics Library [12] to solve these difficulties.

Modelica is an open-source modelling standard, explicitly designed for multi-disciplinary and multi-physics modelling and simulation. Its equation-based approach also allows for automatically generating model derivations such as linear state-space or non-linear dynamic inverse models. The model equations can be exported to C-compiled code, allowing for fast simulation and integration in Simulink S-functions or embedded systems using the Functional Mockup Interface [2].

The DLR Flight Dynamics Library provides a great variety of basic aircraft models as well as environmental models necessary for six degrees of freedom (6DOF) and aeroelastic simulation of conventional aircraft. Positions can be expressed in the

World Geodetic System (WGS84) and the rotation of the earth is taken into account. Detailed models of the magnetic and gravitational fields of the earth are included in the library as well as standard atmosphere data and wind simulation.

A major advantage of the Flight Dynamics Library is that it extensively uses standard Modelica mechanical connectors, such as MultiBody frames. This allows to connect any additional model with force or position interfaces to the airframe by a standardized interface without changing the model equations. Interactions other than mechanical connections are typically realized with Modelica's "outer" models providing global functions for the environmental properties. Buses are used for internal control signals. The present model makes use of these techniques and keeps up the interfacing philosophy.

Previous work on this subject has been published in [11]. In this paper, we enhance the model with elastic modes and an autopilot to gain a more holistic view of the system. Thus, all interactions as shown in Fig. 1 are taken into account, except for the dashed lines.

In the following sections, we first give an introduction to the solar aircraft used in our study. We then introduce the overall model structure in section 3. Section 4 summarizes the main components of the integrated model. In section 5, we briefly present selected appplications for the model and section 6 provides conclusions and an outline of future work.

2 The ELHASPA Aircraft

In this paper, the Electric High-Altitude Solar-Powered Aircraft (ELHASPA) serves as an example for the developed modelling approach. It is developed and built by a consortium including the German Aerospace Center's (DLR) Robotics and Mechatronics Center (RMC) and several industrial partners. The unmanned aircraft is intended to advance technologies and test applications for continuous high-altitude solar-powered flight. A design model of the aircraft is shown in Fig. 2.

ELHASPA's light-weight carbon fiber construction has a span width of 23 m, a wing area of 25 m^2 and a total mass of 100 kg. It is designed to fly at speeds of 6 m/s to 15 m/s and in stratospheric altitudes of more than 15 km. It has two separate

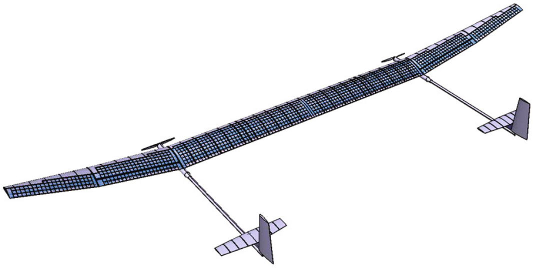


Fig. 2 The ELHASPA aircraft has a wing span of 23 m and a mass of 100 kg. It has two separate avionics systems in its two fuselages and is intended to advance technologies and test applications for continuous high altitude solar flight.

fuselages with identical energy and avionics systems. The two ailerons and the all-movable elevators and rudders are independently steerable.

3 Integrated Model Structure

While Modelica was chosen as the basic modelling environment, simulation and autopilot development is performed in Simulink. All model components and their interactions illustrated in Fig. 1 are therefore Modelica-based except for the autopilot. The connections are mostly realized by the use of Modelica's outer models and

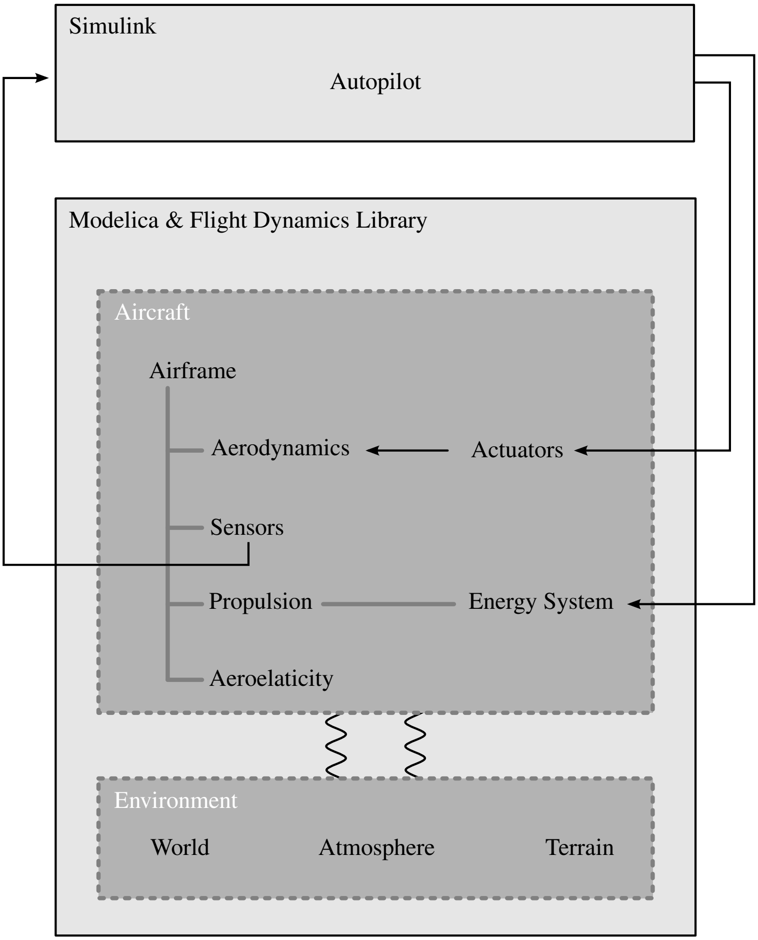


Fig. 3 The overall model is comprised of a main and a controller module, realized in Modelica and Simulink respectively. Interactions between aircraft and environment are managed by ”outer” models in Modelica (ʌ/ʌ), whereas internal aircraft components are mostly linked by physical connectors (▬). The autopilot is connected to the aircraft model using regular signal exchange (⟶).

Modelica's standard mechanical connectors. The model is then exported to Simulink and connected to the autopilot. Sensor and control data are passed as regular signals. The overall model structure is shown in Fig. 3.

The model consists of three main parts. The aircraft model is separated from the environment model, such that environmental information is globally available to all submodules and possibly multiple aircraft. Data is exchanged between the aircraft and the environment modules by Modelica's ”outer” models, providing global functions and variables. The autopilot model is again separated from the plant model and can be implemented in Simulink, Modelica or any suitable modelling language. It is connected to the plant by mere signal flows, making possible this multi-tool approach.

The aircraft model is driven by a central airframe module, which implements the equations of motion. All modules requiring or generating position, attitude, force or moment information are connected to the airframe by physical ”MultiBody” connectors. This includes aerodynamics, sensors, propulsion and aeroelasticity modules. The propeller of the propulsion system is driven by a rotational flange connector, which is connected to the motor flanges of the energy system. Actuator models shape the command signals for the aerodynamic's control surface deflections.

A variety of base models is already available in the Flight Dynamics Library. The modularity of the library is maintained by defining additional modules and modular upgrades to existing modules. For example, the atmospheric effects on radiation are contained in a model called ”IrradianceUpgrade”, which is then overloaded on the existing standard atmosphere model. All modules can be seamlessly interchanged with the modules already provided by the Flight Dynamics Library.

The following section summarizes the modules particular to the ELHASPA case.

4 Components of the ELHASPA Model

The main parts of the ELHASPA model are described briefly in this section. This includes the radiation calculation within the environment model, the energy system, the propulsion and the aerodynamics modules of the aircraft. Special attention is paid to the unsteady aerodynamics and aeroelasticity. The autopilot is described in a separate paper [9].

4.1 Environment

The environment model provides reference systems and gravity models for the world and atmospherical quantities relevant to flight mechanics such as the wind, static pressure and air density. For solar aircraft, it is extended with additional components to simulate the position of the sun and the resulting solar irradiance as well as atmospheric refraction (ray deviation) and extinction (ray attenuation) of the solar radiation. This allows to simulate day/night cycles including a fine resolution at the horizon and attenuation of the solar radiation for altitudes close to the earth's surface.

The calculation of the sun's position is derived from Modelica's Satellite Dynamics library and is build on a simplified analytical sun model by Schlyter [19]. This model implements the sun as a regular earth satellite on an elliptical orbit with changing orbital elements. The mean solar irradiance is given by $I_0 = 1366\,\mathrm{W/m^2}$ and varies slightly with the distance between the sun and the earth.

The atmospheric refraction R is modelled to deviate rays of sunlight from the direct path according to Saemundsson [18]. The atmospheric extinction is modelled using the relative airmass AM according to Young [23] and Honsberg and Bowden [8]. These models are valid for operation in relatively low altitudes. For actual stratospheric flight simulation, additional effects of the solar spectrum and modified extinction calculations have to be included in the models.

The modelled irradiation characteristics are depicted in Fig. 4. A more detailed description of the modifications to the standard Flight Dynamics Library models can be found in [11].

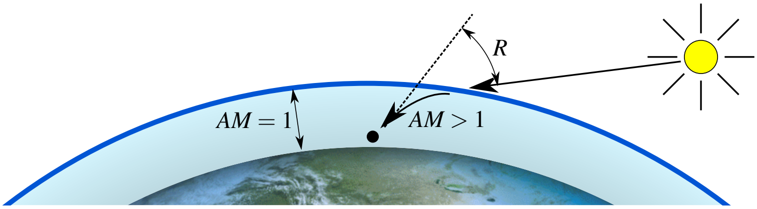


Fig. 4 The additional environmental model includes simulation of the sun location, the atmospheric refraction R and the atmospheric extinction as a function of the relative airmass AM.

4.2 Energy System

The energy system consists of independent electrical systems for each fuselage of the aircraft. Each side contains a solar energy generation block, a battery pack, an electric motor, and further technical loads. The batteries and the solar generation block are directly connected to the high voltage power bus. The motor is driven by a motor controller. The low voltage technical loads are powered by a voltage converter. A battery manager provides feedback on the current state of charge to the solar energy generators. Fig. 5 depicts one side's electrical system.

Each solar generation block contains several solar panels. Each solar panel has a dedicated position on the aircraft structure and thus is exposed to an individual irradiance. The entire panel is modelled by means of one diode connected in parallel to a current source. The source generates a current proportional to the solar irradiance. An additional maximum power point tracker sets the operating voltage of the panels such that the maximum power is retrieved. All outputs of the maximum power point trackers are then connected in parallel to the main power bus.

The battery pack is modelled by a voltage source and an inner resistance, which are both stored in look up tables as functions of the state of charge and current

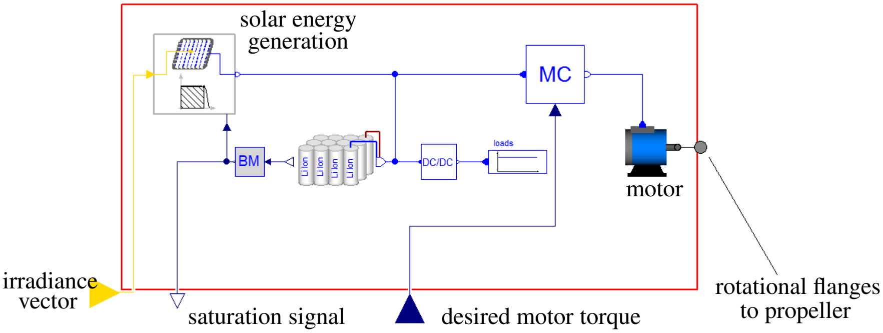


Fig. 5 One half of the energy system model including solar cells driven by the incoming irradiance, batteries, a motor controller (MC) driving a motor and technical loads. The battery manager (BM) provides feedback on the batteries' state of charge.

flow. The battery's state of charge is obtained by integrating the electric current. An additional battery manager is needed in order to prevent the battery from overload. To this end, the battery manager further reduces the voltage of the solar panels in such a way, that no surplus power is generated. The internal saturation signal is provided for use in the control loops. It indicates the amount of currently unused solar power.

The electrical motor is the major consumer of electrical power and drives the aircraft propeller through a Modelica rotational flange. The model consists of an electromagnetic force, which ideally transforms electrical energy into mechanical energy, an electrical resistor to consider copper losses, and a variable friction at the mechanical side to incorporate iron losses. The motor torques are commanded through a control signal.

Technical loads like computers, servos, and communication devices are modelled by a fixed electrical resistance. These loads are powered by a voltage converter considering constant-efficiency as well as fixed losses of the converter.

4.3 Propulsion

The aircraft is equipped with two fixed-pitch propellers. Their characteristics have been estimated with the help of a Blade Element Method provided by the JavaProp software [7]. The resulting thrust force and the torque on the propeller shaft have been calculated for a grid of about 16 000 points at different propeller turn rates, aircraft airspeeds and altitudes. The propeller is driven through a Modelica standard rotational connector, which allows to connect arbitrary motor models to the propeller model, such as direct speed inputs or detailed motor models. The data is interpolated at runtime to yield the current propeller thrust and shaft torque. The resulting forces and moments are applied through a MultiBody connector to the airframe.

4.4 Steady Aerodynamics

The baseline steady aerodynamics of the aircraft are modelled using the Athena Vortex Lattice (AVL) software [4], a potential flow simulation for thin airfoils. The estimation is augmented with airfoil data obtained from Xfoil [5]. These routines provide fast aerodynamic estimation and they do not require extensive modelling. Thus, the aerodynamic methods described here are capable of easily following design changes and providing feedback to the design process in turn. Qualitative assessment based on the pilot's experience from training and the first test flight indicates a good match between simulation and real flight.

The aerodynamic model is composed of the main wing, two horizontal, and two vertical stabilizers. It is depicted in Fig. 6 along with the AVL coordinate system. The model contains 1680 panels and six independently actuated control surfaces. The differing axes definition from standard notation is handled by an extension to the Flight Dynamics Library's base models.

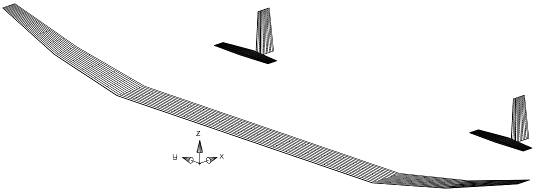


Fig. 6 ELHASPA's vortex lattice mesh contains 1680 panels for the main wing, horizontal and vertical stabilizers and six independently actuated control surfaces

As the vortex lattice method is a detailled but mostly linear modelling tool, all surfaces are augmented with Xfoil polars. The polars provide nonlinear drag effects, which lead to a stop of the simulation in case of stall.

Since the computation of the flow field within AVL cannot be carried out in real time, the aerodynamic model is reduced to a polynomial approximation of the AVL model. To this end, a number of 10 000 points are calculated in the relevant flight envelope. The resulting aerodynamic coefficients are fitted in a least squares sense to multi-dimensional polynomials $\mathrm{p}(x_1, x_2, ...)$ in the aerodynamic angles (α and β), the rotational rates (p, q, r) and the control surfaces $(\delta_a, \delta_e, \delta_r)$. These polynomials also cover coupling effects of the form $x_1^m \cdot x_2^n$. The output equations from the above procedure have the following form:

$$
\begin{aligned}
C_L &= \mathrm{p}\,(\alpha, q, \delta_a, \delta_e) && \text{(Lift coefficient)}\\
C_D &= \mathrm{p}\,(\alpha, q, \delta_a, \delta_e) && \text{(Drag coefficient)}\\
C_Y &= \mathrm{p}\,(\beta, p, r, \delta_a, \delta_r) && \text{(Side-force coefficient)}\\
C_l &= \mathrm{p}\,(\beta, p, r, \delta_a, \delta_e, \delta_r) && \text{(Rolling moment coefficient)}\\
C_m &= \mathrm{p}\,(\alpha, q, \delta_a, \delta_e) && \text{(Pitch moment coefficient)}\\
C_n &= \mathrm{p}\,(\beta, p, r, \delta_a, \delta_e, \delta_r) && \text{(Yaw moment coefficient)}
\end{aligned}
$$

4.5 Unsteady Aeroelasticity

By combining a traditional nonlinear rigid body flight dynamics model with linear aeroelastic dynamics, the equations of motion of the flexible aircraft can be derived. These equations combine the rigid body motion of the aircraft and the elastic deflections with respect to a body fixed system. A set of simplifying assumptions and constraints as outlined for example by Waszak and Buttrill [22] and Reschke [16] then lead to traditional nonlinear rigid body and flexible equations in decoupled form.

The flexible part of the integrated model considers the representation of the aircraft by a chosen number of linear elastic modes obtained through modal analysis of the underlying condensed structural model of the aircraft. Rigid motion is defined by six rigid body modes corresponding to the six degrees of freedom of a body reference frame fixed to the center of gravity in three-dimensional space. The classic linear elastic equations of motion are then generalized by the elastic modes and driven by the corresponding generalized coordinates.

In the Modelica Flight Dynamics Library arbitrary structural forces can easily be incorporated by flexible connectors. Structural grid indices are used to specify the location of the modal displacements within the modal basis to obtain the corresponding generalized flexible forces. Updating the resulting forces and moments about the rigid motion reference frame then simply becomes a matter of incorporating the momentary structural grid displacements and rotations.

Rigid body modes in the body fixed reference frame are added to the modal basis so that the nonlinear rigid body and elastic equations of motion are coupled after generalizing a set of distributed aerodynamic forces. In the presented case unsteady aerodynamic forces acting on the structural frame have been calculated using the so called doublet lattice method (DLM) (see e.g. Albano and Rodden [1]). DLM aerodynamic forces are derived from an aerodynamic panel model of the lifting surfaces. The result are constant complex valued matrices of aerodynamic influence coefficients (AICs) relating the harmonically varying downwash on each of those panels to the pressure difference across each of the other panels. Obtaining the aerodynamic forces thus becomes a matter of determining the downwash due to rigid body motion and elastic deflections and multiplying the resulting panel pressures with the respective area of the panel [10].

To be compatible to the generalized equations of motion, three subsequent post-processing steps are required. First the DLM coefficients are available as complex valued matrices in frequency domain only and have to be transformed to the time domain. Next the aerodynamic grid is splined to the structural grid by a conventional beam spline as for example implemented in the commercial aeroelastic solvers of Msc's NASTRAN [13], so that rigid and flexible motion can be related to the applied downwash as well as integrated aerodynamic pressures to structural forces. Roger's rational function approximation (see [17]) was therefore applied which allows conversion to time domain by a simple inverse Laplace transformation, but also leads to additional differential equations to capture the phase shift towards the corresponding quasi-steady forces. In a second step, the generalization of the so obtained real

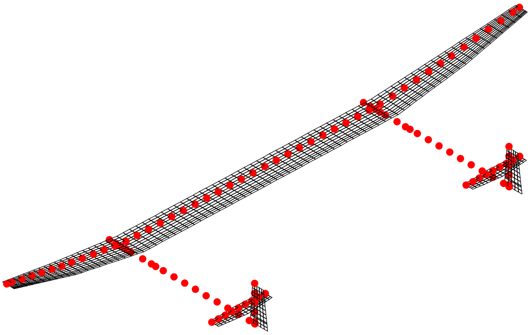


Fig. 7 DLM panel mesh of the ELHASPA aircraft. The red dots represent the grid points of the condensed structural model.

valued unsteady aerodynamic coefficients using the modal basis of rigid and flexible modes leads to rigid and elastic forces as well as coupling forces between the former two. The coupling forces provide the influence from rigid body motion onto the flexible structure and vice versa such that the aircraft's elastic structural behaviour is fully integrated in the simulation model. Additional modes were added to the modal basis to account for excitation by control surface deflections and turbulence.

5 Selected Applications

The integrated model is used throughout the development and commissioning of the ELHASPA aircraft. Valuable feedback to the design team include aerodynamic parameter studies and flutter estimations. In this section, two selected applications are introduced. The preliminary analysis of a control law is illustrated by comparing controlled flight of the full model to a reduced mission simulation. The second example illustrates the influence of wing flexibility on the harvested solar energy.

The reduced mission simulation has been previously introduced in [11]. In a first step, the ELHASPA model is reduced to a point mass model neglecting elasticity and the energy system. For this reduced model, a non-linear dynamic inverse model is automatically generated using Modelica. The inverse model directly receives the trajectory as inputs and computes all relevant aircraft states from it. The reduced mission simulation is approximately 50 000 times faster than realtime. The results are compared to a second simulation. It includes the fully flexible ELHASPA model and the autopilot being developed for the aircraft (see [9] for details). The results are compared for a turn of 180° taking approximately 100 seconds.

Figure 8 shows the track angle χ and the roll angle ϕ for the described maneuver. The track angle develops a similar constant turn rate in both cases. However, the transient behavior is more detailed in the full model case. The autopilot reacts quicker to the command and uses an acquire-and-hold procedure to capture the new track angle command.

The simulated roll angle is different for both cases: The inverse model illustrates the ideal kinematic response of ELHASPA to the track angle change. The controlled

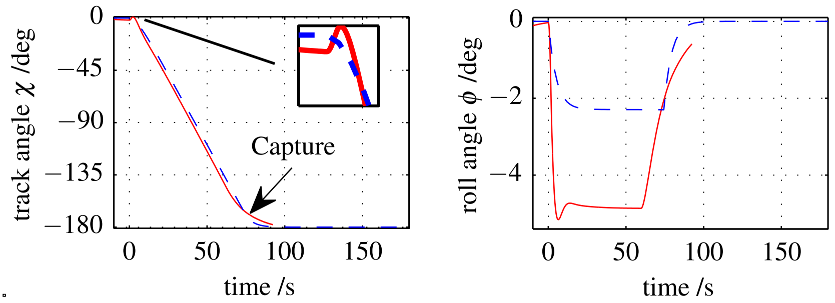


Fig. 8 The track and roll angle plots for the point mass inverse model (– –) and the controlled flexible model (—)

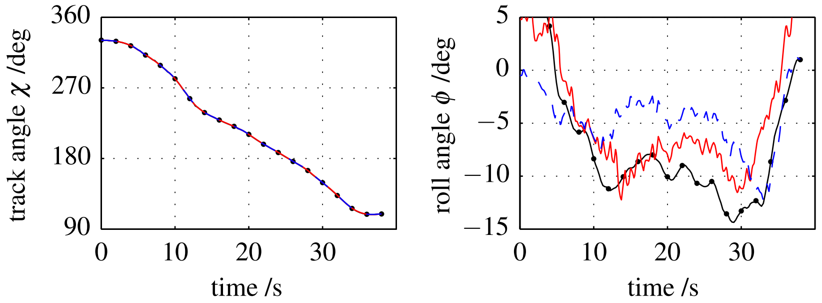


Fig. 9 A trajectory from flight experiments (-•-) is re-simulated using an inverse point mass model with the reconstructed (—) and with zero (– –) sidelip angle

model shows actual dynamic behavior as expected from the controlled aircraft. During the turn, the controlled model builds up an adverse sideslip angle. This must be compensated for by a higher roll angle.

A similar effect can be observed in data from the first manual test flights. Note, that they have not been optimized for model validation. Figure 9 shows a turn of approximately 220° taking about 40 seconds. The trajectory data is used to steer an inverse point mass model taking into account steady aeroelastics. First, the reconstructed sideslip angle is used. The model follows the given trajectory exactly and the good agreement in terms of roll angle are an indicator of a good lateral simulation model. If the same trajectory is simulated maintaining zero sideslip angle, the necessary roll angle is visibly decreased. This is according to expectations and confirms the effect found with the controlled simulation.

To show the response of the model to unsteady wind profiles, a '1-cosine' gust profile with a short gust reference length is applied to the fully flexible model. The gust gradually applies downwash to the wing while travelling downstream. The profile and the rigid body part of the model response are shown in Fig. 10. The aircraft response is corrected by the trim angle of attack of approximately −7°, which is due to the wing's high angle of incidence.

The initial response to the gust is a mere change in sensed angle of attack α. This change is then also converted to a kinematic pitch attitude. The dynamics of the aircraft are strongly damped and return slowly to their trim state. Minor oscillations can still be observed, right after the gust has passed.

Figure 11 illustrates the flexible response of ELHASPA's lifting surfaces to the '1-cosine' gust. It also illustrates the flexibility influence on a solar panel attached to the right wing tip. Each solar panel is fixed relative to a single structural grid. By using flexible instead of rigid connector blocks, the position and orientation of the grid points and hence the wing's deformation is taken into account when calculating the irradiance.

ELHASPA's elastic response is dominated by the first symmetric bending mode, which bends the central section between the engines upwards and rotates the outer wing tips downwards. In trimmed flight, this is indicated by the negative x-axis rotation of the outward structural grid on the right wing tip as shown in the graph on the left hand side. The gust then hits the lifting surface leading to an inital upward

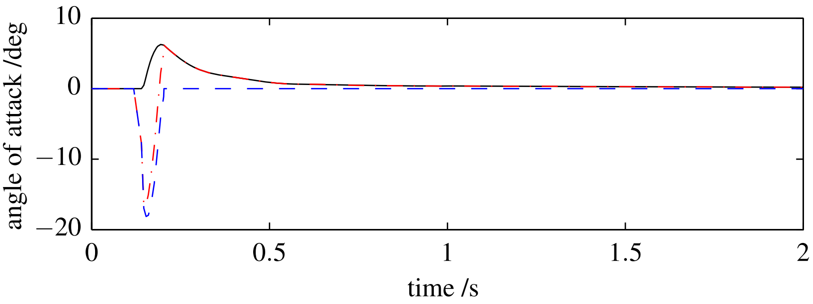


Fig. 10 The applied '1-cosine' gust profile (– –) is depicted in terms of angle of attack. The rigid body part of the response is shown by the kinematic and aerodynamic angles of attack α_F (—) and α (– - –). The aircraft angles of attack are corrected by the trim state.

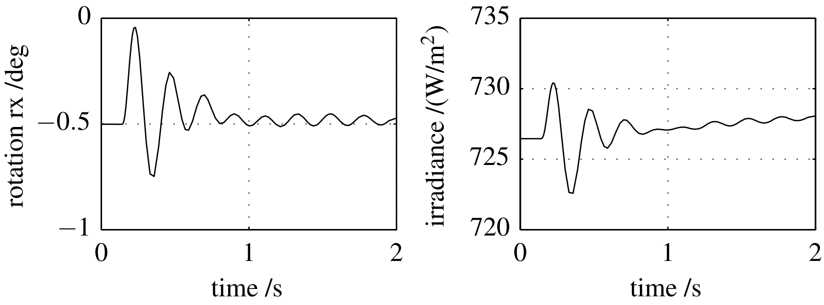


Fig. 11 The flexible response to the '1-cosine' gust is illustrated by the behavior of a structural grid point and the attached solar panel. The pair is located at the outward right wing side. The rotation rx about the x-axis is measured in loads axes of the structural grid with positive x is downstream. The resulting variation in irradiance can be seen in the right graph.

deflection followed by damped oscillations. As can be seen in the graph on the right side, these high frequency oscillations in the grid orientation manifest themselves in a corresponding change of irradiance on the solar panel. The low frequency rigid body motion initiated by the coupling with the elastic model can also be observed.

6 Conclusions

We have presented an integrated modelling approach for the Electric High-Altitude Solar-Powered Aircraft ELHASPA. It meets all design requirements: The relevant multi-physics interactions between flight dynamics, energy system, aeroelastics and the environment are covered. The system is modular and can be extended easily to cover remaining aspects. The approach provides a consistent modelling scheme of all aspects of the aircraft. The simulation is sufficiently fast for closed-loop simulations. Model variants for most imaginable applications can be derived easily, yielding also extremely fast reduced variants.

The model is used in the overall design, flight test and operation process. This includes application on different operating systems such as realtime embedded systems and desktop computers. The use of a single integrated model proves efficient for a small project team.

Future work will concentrate on validating and extending the model. All modules will be validated and updated with flight and ground tests. Results of recently conducted load tests will be used to validate the elastic model. Additional modules will be included to allow for high-fidelity simulation of longer flights in high altitudes, such as temperature effects and servo energy consumption. The model will also be exploited for the continuing development and operation, especially of the autopilot and control laws as well as mission planning and management.

References

[1] Albano, E., Rodden, W.: A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flow. AIAA Journal 7, 279–285 (1969)

[2] Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Clauß, C., Elmqvist, H., Junghanns, A., Mauss, J., Monteiro, M., Neidhold, T., et al.: The functional mockup interface for tool independent exchange of simulation models. In: Modelica 2011 Conference, pp. 20–22 (March 2011)

[3] Boucher, R.: History of solar flight. AIAA paper 84–1429 (1984)

[4] Drela, M., Youngren, H.: Athena vortex lattice (2004), `http://raphael.mit.edu/avl`

[5] Drela, M., Youngren, H.: Xfoil – subsonic airfoil development system (2008), `http://raphael.mit.edu/xfoil`

[6] Fritzson, P.: Principles of object-oriented modeling and simulation with Modelica 2.1. Wiley-IEEE Press (2004)

[7] Hepperle, M.: JavaProp Users Guide (2010)

[8] Honsberg, C., Bowden, S.: PVCDROM (2010), `http://www.pveducation.org/pvcdrom/properties-of-sunlight/air-mass`

[9] Kastner, N., Looye, G.: Generic TECS based autopilot for an electric high altitude solar powered aircraft. In: CEAS EuroGNC. CEAS, Delft (submitted for publication, 2013)

[10] Kier, T.M., Looye, G.H.N.: Unifying manoeuvre and gust loads analysis models. IFASD paper (2009)

[11] Klöckner, A., Schlabe, D., Looye, G.: Integrated simulation models for high-altitude solar-powered aircraft. In: AIAA Modeling and Simulation Technologies Conference. AIAA, Minneapolis, AIAA 2012-4717 (2012)

[12] Looye, G.: The new DLR flight dynamics library. In: Proceedings of the 6th International Modelica Conference, vol. 1, pp. 193–202 (2008)

[13] MSC: Nastran Version 68 Aeroelastic Analysis User's Guide (2004)

[14] Noth, A.: History of solar flight. Aircraft & spacecraft systems design lecture notes, Autonomous Systems Lab, ETH Zürich, Switzerland (2008)

[15] QinetiQ Group, PLC, Zephyr UAV (2009), `http://www.qinetiq.com/what/products/Documents/Zephyr-UAV.pdf`

[16] Reschke, C.: Integrated flight loads modelling and analysis for flexible transport aircraft. PhD thesis, Institut für Flugmechanik und Flugregelung der Universität Stuttgart (2006)

[17] Roger, K.: Airplane math modeling methods for active control design. AGARD (1977)

[18] Saemundsson, T.: Atmospheric Refraction. Sky and Telescope 72, 70 (1986)

[19] Schlyter, P.: How to compute planetary positions (2011), `http://stjarnhimlen.se/comp/ppcomp.html`

[20] Solar Impulse S. Solar impulse – around the world in a solar airplane (2011), `http://solarimpulse.com/`

[21] Voit-Nitschmann, R.: Solar- und Elektroflugzeuge – Geschichte und Zukunft. In: Jahrbuch aus Lehre und Forschung der Universität Stuttgart, Universität Stuttgart, pp. 88–99 (2001)

[22] Waszak, M.R., Buttrill, C.S.: Modeling and model simplification of aeroelastic vehicles: An overview. Tech. rep., NASA (1992)

[23] Young, A.T.: Air mass and refraction. Applied Optics 33(6), 1108–1110 (1994)

Design and Flight Testing of Nonlinear Autoflight Control Laws Incorporating Direct Lift Control

Thomas Lombaerts and Gertjan Looye

Abstract. This publication describes how direct lift control has been incorporated in a nonlinear autoflight control algorithm. Direct lift control demonstrated its use in earlier aircraft designs. In two recent internal DLR projects, accurate flight path tracking in atmospheric disturbances was an important research goal, where direct lift control could potentially provide an important contribution. In these projects, use has been made of nonlinear control techniques. Objective of this research publication is to incorporate direct lift control in these nonlinear control laws. Simulations as well as flight tests have shown that more accurate flight path changes are achieved by this addition. Direct lift control can be particularly useful for path tracking as well as in turbulent air, since it facilitates lift changes without pitching. The non-minimum phase behaviour of the aircraft caused by the elevators is eliminated.

1 Introduction

Besides the regular control surfaces which can be found on a conventional aircraft wing, such as (high and low speed) ailerons, flaps, slats and speedbrakes or spoilers, some specific aircraft types have an additional control capability at the trailing edge of the wing to influence lift directly, this is called direct lift control (DLC). For most of the latter aircraft, DLC is provided by means of the spoilers in the lower deflection ranges, such as is the case for the Lockheed L-1011 Tristar civil airliner [14], the Boeing C-17 Globemaster III military airlifter, and the aircraft-carrier based jets the Grumman F-14 Tomcat and the Lockheed S-3 Viking. Their use is to manage sink rate finely so that a more accurate and smoother glideslope can be flown. This is done through the lift force directly instead of indirectly through the angle of attack

Thomas Lombaerts · Gertjan Looye
German Aerospace Center (DLR), Robotics and Mechatronics Center,
Institute of System Dynamics and Control,
Oberpfaffenhofen, D-82234 Weßling, Germany
e-mail: {thomas.lombaerts,gertjan.looye}@dlr.de

α and subsequently the pitch rate q, which takes more time. As a result, direct lift control facilitates lift changes without pitching. More precisely, the non-minimum phase behaviour of the aircraft caused by the elevators is eliminated in this way. This is especially useful during the approach. In the literature, many possible purposes have been mentioned for direct lift control, namely flight path tracking[6], station keeping for in flight refueling[15] and increasing passenger comfort during turbulence[5]. Ref. [7] makes a comparative analysis of different control strategies for the use of continuously variable trailing edge control surfaces during landing. This history confirms the efficiency of direct lift control for flight guidance.

Two recent internal DLR projects focus on flight guidance, namely "Wetter-und-Fliegen" and "TOPGAL". "Wetter-und-Fliegen" (completed in 2011) aimed at augmenting safety and efficiency of air transport. This high-level goal has been achieved by building systems for automatic control to increase the performance of the aircraft when confronted with wind gusts, wake vortices and thunderstorms. TOPGAL (Total Performance System for GBAS (Ground Based Augmentation System) based Automatic Landings, completed in 2012) investigates GBAS based navigation in conjunction with flight management and autopilot systems. A so-called Ground Based Augmentation System enables aircraft to use satellite navigation for guidance on a precision approach. In both projects, accurate path tracking and atmospheric disturbance rejection are important research aspects. It is especially in this context that direct lift control can provide an important contribution, as the aforementioned history has shown.

In the VFW-614 based ATTAS laboratory aircraft (Advanced Technologies Testing Aircraft System) of DLR, a direct lift control capability has been included primarily for the purpose of in-flight simulation, however they can also be used for autoflight guidance. ATTAS has DLC flaps as part of the landing flaps, which makes them only usable at flap settings 1, 5 and 14. ATTAS's direct lift control capability provides an opportunity for the flight guidance research in the internal DLR projects "Wetter-und-fliegen" and "TOPGAL". In this context, a set of nonlinear autoflight control laws incorporating direct lift control has been designed and flight tested on ATTAS. This is an extension of a previously developed set of nonlinear flight control laws, which has been presented in Ref. [11]. This latter autoflight control algorithm is based on three consecutive dynamic inversion loops, which have been separated by means of the principle of time scale separation. This control setup can be augmented for flight path control by making use of the direct lift control flaps. This augmentation serves a double purpose. First, aircraft responses to flight path angle commands are made faster and more accurate. Secondly, turbulence induced disturbances are rejected to improve passenger comfort, as will be shown in the results. The purpose of the paper is to show the integration of the concept of direct lift control in nonlinear autoflight control laws.

Section 2 provides a description of the simulation model of the ATTAS laboratory aircraft. A summary of the benchmark nonlinear autoflight controller structure is given in section 3, while the improvement of direct lift control is elaborated in section 4. Simulation results and flight test results are given in sections 5 and 6. Finally, section 7 provides some conclusions and recommendations.

2 Description ATTAS Model

This section describes the high fidelity aircraft simulation model which has been constructed using the Modelica Flight Dynamics Library [13]. The aircraft is the flying test-bed ATTAS (Advanced Technologies Testing Aircraft System) [3]), which is operated by the German Aerospace Center (DLR). ATTAS is based on a VFW 614, a small 44-passenger civil transport aircraft with two Rolls-Royce turbofan engines mounted on top of the wings, a wing span of 21.5 m and a length of 25 m, as shown in Fig. 1. This simulation model was originally developed in the frame of the EU project REAL [18] for automatic landing control laws design and is the basis for the flight control law design described in this paper.

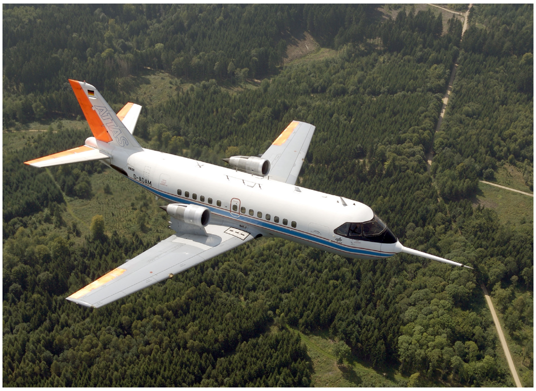


Fig. 1 DLR VFW-614 ATTAS in flight; source: DLR

The ATTAS model has been implemented using the object-oriented modeling language Modelica and the Flight Dynamics Library, developed at the department of Robotics and Mechatronics. An object diagram showing the most important components is depicted in Fig. 2. From this model, simulation and design analysis models are automatically generated, as well as the inner core of the Nonlinear Dynamic Inversion-based flight control laws [13].

The ATTAS simulation model includes flight dynamics, systems and environment models. The most important components are briefly described below. These descriptions focus on aspects relevant for the purpose of control design.

Aerodynamics

Aerodynamic model equations have been defined for typical aerodynamic variables like angle of attack α, angle of side slip β, true airspeed V_{TAS}, Mach number M, etc. For this, ATTAS-specific aerodynamic coefficient data have been implemented[1, 4]. In addition, multiplicative uncertainty has been added to the aerodynamic model coefficients in order to cover potential differences with the actual aerodynamics.

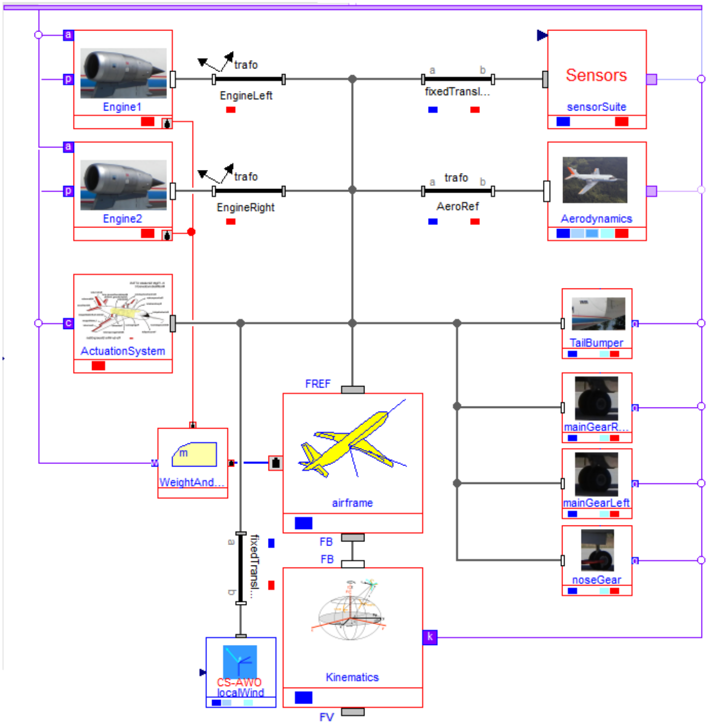


Fig. 2 Modelica object diagram with the most important ATTAS components

Propulsion

The ATTAS is equipped with Rolls-Royce turbofan engines. In this component, the thrust is computed and equations of motion for engine shaft dynamics are taken into account, including effects such as hysteresis in the fuel control unit.

Actuator dynamics

The ATTAS is amongst others equipped with ailerons, rudder, elevators and a trimmable horizontal stabiliser. These surfaces are driven by hydraulic actuators, which have been modelled as first order transfer functions with rate and position limits. Speed dependent deflection command limiting has been incorporated as well. Moreover, a flight control system signal transport delay of 150ms (including multiplicative uncertainty) has been added in the aileron, elevator and rudder channels.

Sensor dynamics

The ATTAS is equipped with a range of sensors for air data, inertial measurements (accelerometers, etc.), guidance (ILS, Radio altitude), etc. These sensors are combined into a sensor system model. As far as the air data sensors are concerned, there are two alternatives to measure the angle of attack α. This will be elaborated further. The sensor dynamics are modelled by first order transfer functions representing the sensor lag. When the conventional vanes are used to measure the angle of attack α, an additional transport delay of $300ms$ has to be incorporated as well. For some measurement data, quantizer effects or noise is added. For example, when the angle of attack sensor of ATTAS is on a long nose boom, its flexibility adds a 6 Hz disturbance to the signal. For this reason, the signal has to be combined with inertial measurements using complementary filtering techniques as explained by Looye[13]. The relevance of these dynamics and disturbances for specific measurements is shown in table 1.

Table 1 Instrumentation error information for measuring equipment

sensed quantity	variables	sensor	delay	quantizer	noise type
		lag		effect	
speed (various)	V	✓		✓	
body angular rates	p, q, r	✓			
load factors	n_x, n_y, n_z	✓			
Euler angles and course	ϕ, θ, ψ, χ	✓		✓	
aerodynamic angle (nose boom)	α	✓			sine form
aerodynamic angle (vanes)	α	✓	✓		
flight path angle	γ	✓		✓	
fan speed	$N1$				white noise

Mind that raw measurement data contain much more disturbance effects, e.g. raw inertial measurements from gyros and accelerometers are always perturbed by a bias and noise. However, the research presented in this paper considers a loosely coupled flight instrumentation system, using processed observations as opposed to the tightly coupled counterpart with raw measurement data.

Wind and turbulence

Wind as well as turbulence have been incorporated in the simulation model. Wind is treated here as a stationary phenomenon. Turbulence is a stochastic process that can partially be defined by power spectra of the velocity components. Turbulence is modeled by the commonly used Dryden spectra[16].

3 Autoflight Controller Structure

The autoflight controller consists of three consecutive dynamic inversion loops, based on the concept of time scale separation, as illustrated in Fig. 3. The inner dynamic inversion loop is in common with the manual control law. The middle and outer control loops are autoflight specific. The inner loop is based on the aircraft dynamics, and is therefore aircraft specific. The second loop is based on the kinematics and thus independent of aircraft type. Finally, the third loop consists of a kinematic (from navigation quantities towards bank angle and symmetric aerodynamic forces) as well as a dynamic part (from symmetric aerodynamic forces towards aerodynamic angles). The commanded sideslip angle in the second loop is typically zero, and does not appear therefore in the outer loop[1]. The commanded quantities in the outer loop: course angle χ, flight path angle γ and speed ΔV are typical navigation quantities for waypoint guidance. Mind that the dynamic part of the control setup complies with the principle of time scale separation, since the aerodynamic moments have typically a higher bandwidth than the aerodynamic forces.

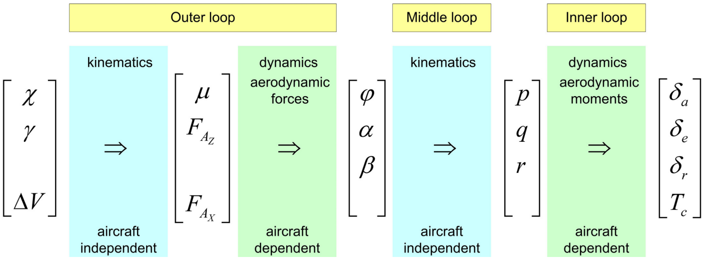

Outer loop
Middle loop
Inner loop
$\begin{bmatrix}\chi\\ \gamma\\ \Delta V\end{bmatrix}$
kinematics
$\Rightarrow$
aircraft independent
$\begin{bmatrix}\mu\\ F_{A_Z}\\ F_{A_X}\end{bmatrix}$
dynamics aerodynamic forces
$\Rightarrow$
aircraft dependent
$\begin{bmatrix}\varphi\\ \alpha\\ \beta\end{bmatrix}$
kinematics
$\Rightarrow$
aircraft independent
$\begin{bmatrix}p\\ q\\ r\end{bmatrix}$
dynamics aerodynamic moments
$\Rightarrow$
aircraft dependent
$\begin{bmatrix}\delta_a\\ \delta_e\\ \delta_r\\ T_c\end{bmatrix}$

Fig. 3 Global overview of the time scale separation based control setup

In the next sections, the control laws in the inner, middle and outer control loop are given. Each loop consists of a dynamic inversion control, a first or second order linear controller and a first order reference model. More detailed explanations about the concept of nonlinear dynamic inversion and the control loops as they have been implemented here, and how they were derived, can be found in Ref. [8, 11].

Inner control loop

The inner control loop for angular rate control is defined as follows [9]:

$$\begin{bmatrix}\delta_a\\ \delta_e\\ \delta_r\end{bmatrix} = \begin{bmatrix} b\tilde{C}_{l_{\delta_a}} & 0 & b\tilde{C}_{l_{\delta_r}}\\ 0 & \bar{c}\tilde{C}_{m_{\delta_e}} & 0\\ b\tilde{C}_{n_{\delta_a}} & 0 & b\tilde{C}_{n_{\delta_r}} \end{bmatrix}^{-1} \cdot$$
$$\left\{ \frac{\mathbf{I}}{\frac{1}{2}\rho V^2 S}\left(\begin{bmatrix}\nu_p\\ \nu_q\\ \nu_r\end{bmatrix} + \mathbf{I}^{-1}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \times \left(\mathbf{I}\begin{bmatrix}p\\ q\\ r\end{bmatrix}\right)\right) - \begin{bmatrix} bC_{l_{\text{states}}}\\ \bar{c}C_{m_{\text{states}}}\\ bC_{n_{\text{states}}}\end{bmatrix} \right\} \qquad (1)$$

where the virtual inputs $\begin{bmatrix}\nu_p & \nu_q & \nu_r\end{bmatrix}^T$ are the time derivatives of the rotational rates of the aircraft, which are selected to be the control variables in order to obtain rate

[1] Note that, due to the setup of the control laws and the nature of dynamic inversion, this can still lead to non-zero steady state values of β, as long as they are in balance with steady state roll angle ϕ and lateral specific force A_y, such that the first order time derivative of the sideslip angle $\dot{\beta} = 0$, see also Ref. [8].

control. They are provided by the first order linear controllers as shown in eq. (2) – (4). The first part of eq. (1) performs the control inversion, while the second part contains the state inversion.

It should be noted that this dynamic inversion is not perfect due to the presence of the multiplicative uncertainties in the aerodynamic model as already explained in section 2. However, the linear controller has shown to be capable to deal with these modeling errors, as can be seen in sections 5 and 6.

The first order linear controllers have the following control laws:

$$\nu_p = K_p\left(p_{\text{ref}} - p_m\right) + \dot{p}_{\text{ref}} \tag{2}$$

$$\nu_q = K_q\left(q_{\text{ref}} - q_m\right) + \dot{q}_{\text{ref}} \tag{3}$$

$$\nu_r = K_r\left(r_{\text{ref}} - r_m\right) + \dot{r}_{\text{ref}} \tag{4}$$

The first order reference models are defined as follows:

$$p_{\text{ref}} = \frac{1}{T_p s + 1} p_{\text{comm}}, \quad q_{\text{ref}} = \frac{1}{T_q s + 1} q_{\text{comm}}, \quad r_{\text{ref}} = \frac{1}{T_r s + 1} r_{\text{comm}} \tag{5}$$

The interconnection of aircraft model, dynamic inversion control laws, first order linear controllers and first order reference models is shown in Fig. 4. The feedforward channel of the first order time derivative from reference model to the linear controller is visible here and eliminates a time delay which would otherwise be present in the tracking of the reference signal.

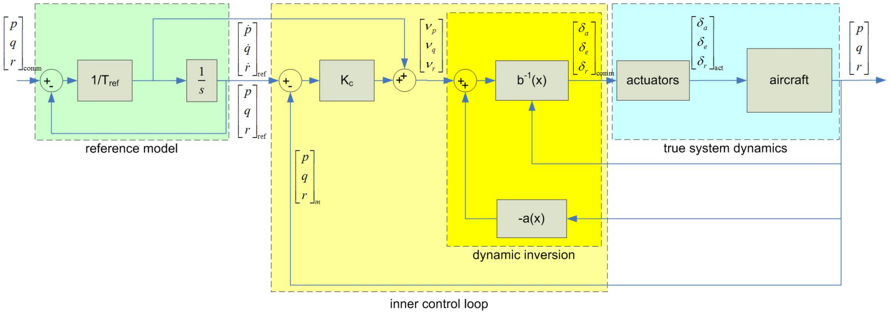

$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_{\text{comm}}$
+
-
1/T_ref
$\frac{1}{s}$
reference model
$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{\text{ref}}$
$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_{\text{ref}}$
+
-
K_c
+
+
$\begin{bmatrix} \nu_p \\ \nu_q \\ \nu_r \end{bmatrix}$
+
+
b^-1(x)
$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}_{\text{comm}}$
actuators
$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}_{\text{act}}$
aircraft
$\begin{bmatrix} p \\ q \\ r \end{bmatrix}$
true system dynamics
$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_m$
-a(x)
dynamic inversion
inner control loop

Fig. 4 Interconnection of aircraft model, dynamic inversion control laws, controllers and reference models

The parameter values, for time constants as well as control gains, are enumerated in Tab. 2. In the next sections, the control laws in the middle and outer control loop will be given.

Middle control loop

From the theory of flight dynamics, see ref. [17], the overall second level NDI control law is:

$$\begin{bmatrix} p_{\text{comm}} \\ q_{\text{comm}} \\ r_{\text{comm}} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ -\frac{v_b}{\sqrt{V^2-w_b^2}} & \frac{u_b}{\sqrt{V^2-w_b^2}} & 0 \\ \frac{w_b}{\sqrt{V^2-v_b^2}} & 0 & \frac{-u_b}{\sqrt{V^2-v_b^2}} \end{bmatrix}^{-1} \cdot \left\{ \begin{bmatrix} \nu_{\dot\phi} \\ \nu_{\dot\alpha} \\ \nu_{\dot\beta} \end{bmatrix} - \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{V^2-w_b^2}}\left(A_z+g\cos\theta\cos\phi\right) \\ \frac{1}{\sqrt{V^2-v_b^2}}\left(A_y+g\cos\theta\sin\phi\right) \end{bmatrix} \right\} \tag{6}$$

where the virtual inputs $\begin{bmatrix} \nu_{\dot\phi} & \nu_{\dot\alpha} & \nu_{\dot\beta} \end{bmatrix}^T$ are the time derivatives of the roll angle and aerodynamic angles of the aircraft, which are selected to be the control variables in order to obtain angle control. They are provided by the first order linear controllers as shown in eq. (7) – (9).

The linear controllers have the following control laws:

$$\nu_{\dot\phi} = K_\phi\left(\phi_{\text{ref}} - \phi_m\right) + \dot\phi_{\text{ref}} \tag{7}$$

$$\nu_{\dot\alpha} = K_\alpha\left(\alpha_{\text{ref}} - \alpha_m\right) + \dot\alpha_{\text{ref}} \tag{8}$$

$$\nu_{\dot\beta} = \left(K_\beta + \frac{K_{\beta_I}}{s}\right)\left(\beta_{\text{ref}} - \beta_m\right) \tag{9}$$

The first order reference models are defined as follows:

$$\phi_{\text{ref}} = \frac{1}{T_\phi s+1}\phi_{\text{comm}}, \quad \alpha_{\text{ref}} = \frac{1}{T_\alpha s+1}\alpha_{\text{comm}}, \quad \beta_{\text{ref}} = 0 \tag{10}$$

The parameter values, for time constants as well as control gains, are enumerated in Tab. 2.

Outer control loop

The command laws in the outer loop for the symmetric forces in the aerodynamic frame of reference $F_{A_{X_{\text{comm}}}}, F_{A_{Z_{\text{comm}}}}$ and the roll angle μ are [8, 11]:

$$\mu_{\text{comm}} = \arctan\left(\frac{\dot\chi_{\text{comm}}\cos\gamma}{\dot\gamma_{\text{comm}} + g\frac{\cos\gamma}{V}}\right) \tag{11}$$

$$F_{A_{X_{\text{comm}}}} = m\left(\dot V_{\text{comm}} + g\sin\gamma\right) \tag{12}$$

$$\begin{aligned} F_{A_{Z_{\text{comm}}}} &= \sqrt{F_{\text{required}}^2 - F_{A_Y}^2} \\ &= -\cos\gamma\sqrt{m^2\left[\left(g+\frac{V\dot\gamma_{\text{comm}}}{\cos\gamma}\right)^2 + \left(V\dot\chi_{\text{comm}}\right)^2\right] - \left(\frac{F_{A_Y}}{\cos\gamma}\right)^2} \end{aligned} \tag{13}$$

where $\mu_{\text{comm}} \approx \phi_{\text{comm}}$ and $F_{A_{X_{\text{comm}}}}, F_{A_{Y_{\text{comm}}}}, F_{A_{Z_{\text{comm}}}}$ are the combined aerodynamic and thrust force components in the aerodynamic frame of reference. The quantities $\left[\dot{V}_{\text{comm}}\ \dot{\gamma}_{\text{comm}}\ \dot{\chi}_{\text{comm}}\right]^T$ are the time derivatives of the navigation quantities of the aircraft, which are selected to be the control variables in order to obtain navigation control. They are provided by the second order linear controllers as shown in eq. (17) – (19).

The command laws for angle of attack and thrust coefficient are as follows:

$$\alpha_{\text{comm}} = \frac{-\frac{F_{A_{Z_{\text{comm}}}}}{0.5\rho V^2 S} - C_{L_0} - C_{L_{LG}} LG}{\left(C_{L_{\alpha,WF}} + C_{L_{\alpha,M,WF}} M\right) Y + \frac{S_h}{S_w}\left(C_{L_{\alpha_h}} + C_{L_{\alpha_{Mh}}}\left(M - 0.3\right)\right)\left(1 - \frac{\partial \varepsilon_h}{\partial \alpha}\right)} \tag{14}$$

where: $$Y = \left(\frac{1}{4}\left(1 + \sqrt{X}\right)^2\right), \quad X = \frac{1}{2}\left(1 - \tanh\left(C_1\left(\alpha_m - \tau_2 \dot{\alpha}_m - \alpha^*\right)\right)\right) \tag{15}$$

$$T_{C_{\text{comm}}} = \frac{F_{A_{X_{\text{comm}}}} + C_D 0.5\rho V^2 S}{\cos\left(\alpha - 3^\circ\right)} \tag{16}$$

with $C_{L_\bullet}$ the aerodynamic derivatives for the lift force, 'LG' stands for 'landing gear', 'WF' depicts 'wing-fuselage combination', S_h and S_w are the surface areas of horizontal stabilizer and wing respectively. ε_h is the downwash at the horizontal stabilizer. α_m is the measured angle of attack and α^* is the angle of attack at $X = 0.5$. X is the location on the wing where the airflow separates. The first and second order linear controllers have the following control laws:

$$\dot{V}_{\text{comm}} = \left(K_V + \frac{K_{V_I}}{s}\right)\left(V_{\text{ref}} - V_m\right) \tag{17}$$

$$\dot{\gamma}_{\text{comm}} = \left(K_\gamma + \frac{K_{\gamma_I}}{s}\right)\left(\gamma_{\text{ref}} - \gamma_m\right) + \dot{\gamma}_{\text{ref}} \tag{18}$$

$$\dot{\chi}_{\text{comm}} = K_\chi\left(\chi_{\text{ref}} - \chi_m\right) + \dot{\chi}_{\text{ref}} \tag{19}$$

where a limited integrator has been used in the speed controller in order to prevent windup effects.

The first order reference models are defined as follows:

$$V_{\text{ref}} = \frac{1}{T_V s + 1} V_{\text{comm}}, \quad \gamma_{\text{ref}} = \frac{1}{T_\gamma s + 1} \gamma_{\text{comm}}, \quad \chi_{\text{ref}} = \frac{1}{T_\chi s + 1} \chi_{\text{comm}} \tag{20}$$

The commanded values for speed, flight path angle and course are provided by the pilot through the mode control panel. The parameter values, for time constants as well as control gains, are enumerated in Tab. 2.

Control gains

The parameter values for the three control loops are given in Tab. 2. The principle of time scale separation can be observed in this table.

Table 2 Parameter values for time constants and control gains

Navigation loop		Aerodynamic loop		Rate loop	
Time	Control	Time	Control	Time	Control
constants	gains	constants	gains	constants	gains
$T_\chi = 0.2$	$K_\chi = 0.4$	$T_\phi = 1.8$	$K_\phi = 0.4653$	$T_p = 2.8$	$K_p = 3.4$
$T_\gamma = 0.4$	$K_{\gamma_P} = 0.9$ $K_{\gamma_I} = 0.1$	$T_\alpha = 1.8$	$K_\alpha = 2.3025$	$T_q = 2.8$	$K_q = 1.1735$
			$K_{\beta_P} = 0.9163$ $K_{\beta_I} = 0.8167$	$T_r = 2.8$	$K_r = 2.3$
$T_V = 0,1$	$K_{V_P} = 0.35$ $K_{V_I} = 0.05$				

4 Direct Lift Control Flaps

The autoflight controller as discussed in section 3 can be extended by means of the direct lift control flaps. Since they have a direct influence on the lift force, without non-mimum phase transient as is the case with the elevator, they can improve the agility of the aircraft. As a consequence, the slightly modified global overview is illustrated in Fig. 5.

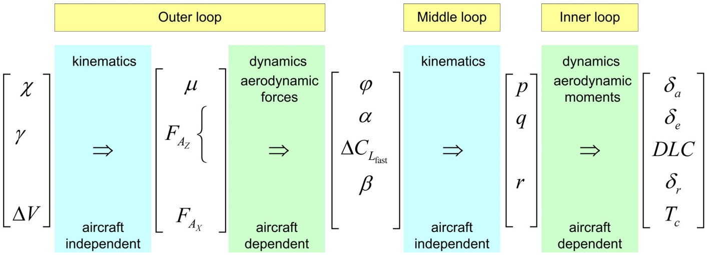


Fig. 5 Global overview of the control setup including direct lift control

Influence of DLC on lift force

The direct lift control flaps (DLC) have a third order influence on the lift force C_L. From ref. [4], this influence can be represented by the following relationship:

$$\Delta C_{L_{DLC}} = 2\left(L_1 \delta_{DLC} + L_2 \delta_{DLC}^2 + L_3 \delta_{DLC}^3\right)$$

The DLC's are operational in flight conditions with flaps 1, 5, and 14. Their nonlinear influence on the lift is illustrated in Fig. 6. This has been implemented in a look up table for DLC deflection ranges between −30° and +20°. Inverse table lookup is allowed to extrapolate. The computed control deflections are then allowed up to the aforementioned saturation limits.

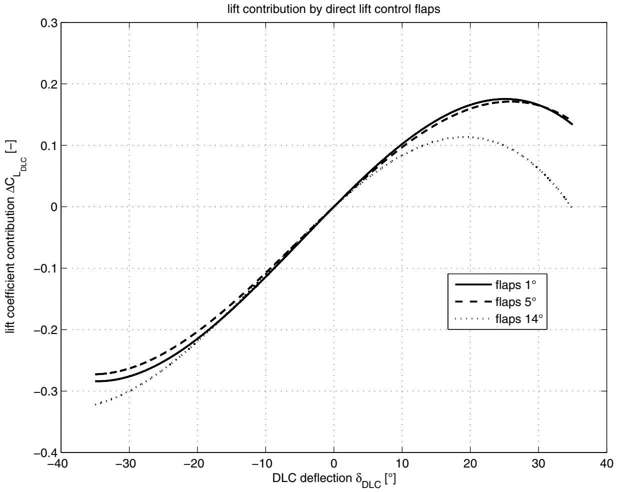


Fig. 6 Lift contribution by direct lift control flaps

DLC control law

The third order relationship above can be used as a nonlinear mapping from DLC flap deflection towards lift force contribution. Inverse implementation in the flight control law can be achieved by a lookup table. Since the reference model for the angle of attack α is in fact a first order filter, the autoflight commanded angle of attack is actually low-pass filtered while processed through the α reference model. Simultaneously, the remaining lift force component to be provided by the DLC's is defined as:

$$\Delta C_{L_{DLC_{\text{comm}}}} = C_{L\alpha}(\alpha_{\text{comm}} - \alpha_m) \tag{21}$$

where α_{comm} is unfiltered. Based upon this lift component contribution, the required DLC deflection is calculated by means of the earlier mentioned lookup table. An overview of this direct lift control setup is given in Fig. 7. Since direct lift control acts complementary to the slower angle of attack α (because of the low pass filtering in the reference model), its behaviour has in fact washout characteristics. Adding DLC increases significantly the γ channel bandwidth. Experiment results shown in sections 5 and 6 will illustrate the beneficial influence of these DLC flaps.

5 Simulation Results

Simulations have been included to demonstrate the beneficial influence of direct lift control on tracking performance as well as atmospheric disturbance rejection.

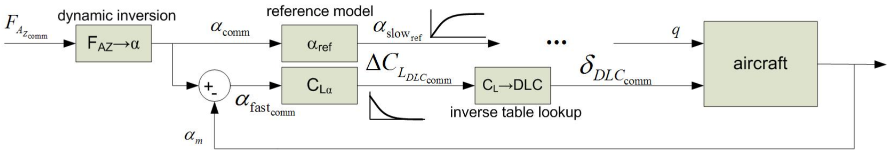


Fig. 7 Frequency based distribution of commanded vertical aerodynamic force $F_{A_{Z_{comm}}}$ over slower reference angle of attack $\alpha_{slow_{ref}}$ and faster direct lift control flap deflection δ_{DLC}

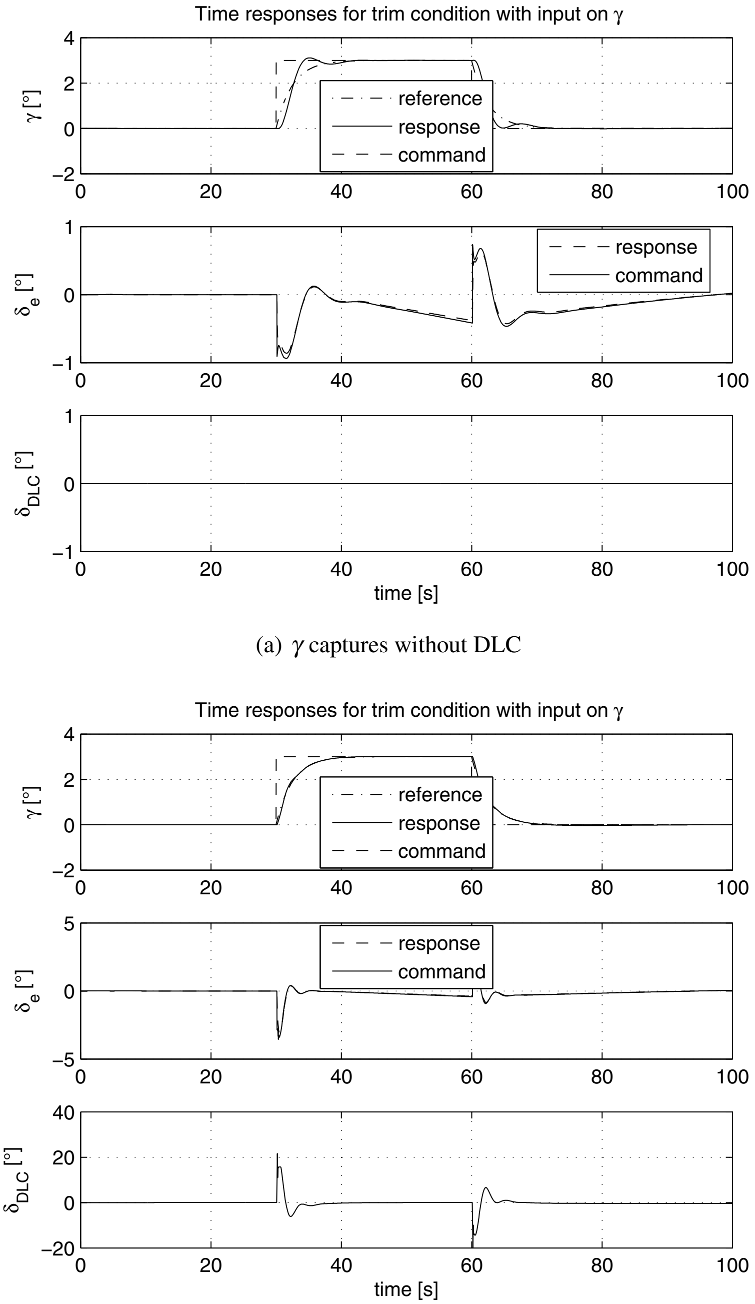
5.1 Tracking Performance of Direct Lift Control

A change in flight path angle $\Delta\gamma = 3°$ has been commanded in the simulations, with and without direct lift control. The simulations can be found in Fig. 8. Figure 8(a) shows the situation without direct lift control contribution. The flight path angle change is achieved by the elevator only. A time delay in and minimal initial opposite response (so-called non minimum phase behaviour) of the flight path angle γ can be seen. The reason for this is explained in Fig. 9(a). The elevator deflects upward ($\delta_e < 0$ in Fig. 8(a)), which causes an instant and brief decrease in total lift on the aircraft ($\Delta L_{tail} < 0$ in Fig. 9(a)). Simultaneously, a pitch up moment $M > 0$ is generated, which sets off a sequence of positive pitch rate $q > 0$, positive change in angle of attack ($\Delta\alpha > 0$), positive lift change $\Delta L > 0$ and thus a climbing flight path angle $\gamma > 0$. Due to the principle of time scale separation, as illustrated in Fig. 3, this sequence causes a time delay.

Figure 8(b) illustrates the aircraft response with direct lift control. The flight path angle change is achieved by cooperating elevator and DLC flaps. The response tracks the reference signal γ_{ref} very accurately. Complementary to the elevator action, which takes some time to build up a change in flight path angle, the DLC flaps act as a washout filter. As illustrated in Fig. 9(b), the downward deflection of the DLC flaps ($\delta_{DLC} > 0$ in Fig. 8(a)) has a direct influence on the total lift of the aircraft $\Delta L > 0$ and thus generates a climbing flight path angle $\gamma > 0$ without time delay and without pitching. Fig. 5 illustrates this as well in the perspective of time scale separation. As can be seen in Fig. 8(b), the DLC flaps have only an intermediate short term "washout" influence to eliminate the time delay and non-minimum phase effect of the elevators. The elevator action is still dominant in the longer term. This is also illustrated in Fig. 10.

5.2 Turbulence Disturbence Rejection of Direct Lift Control

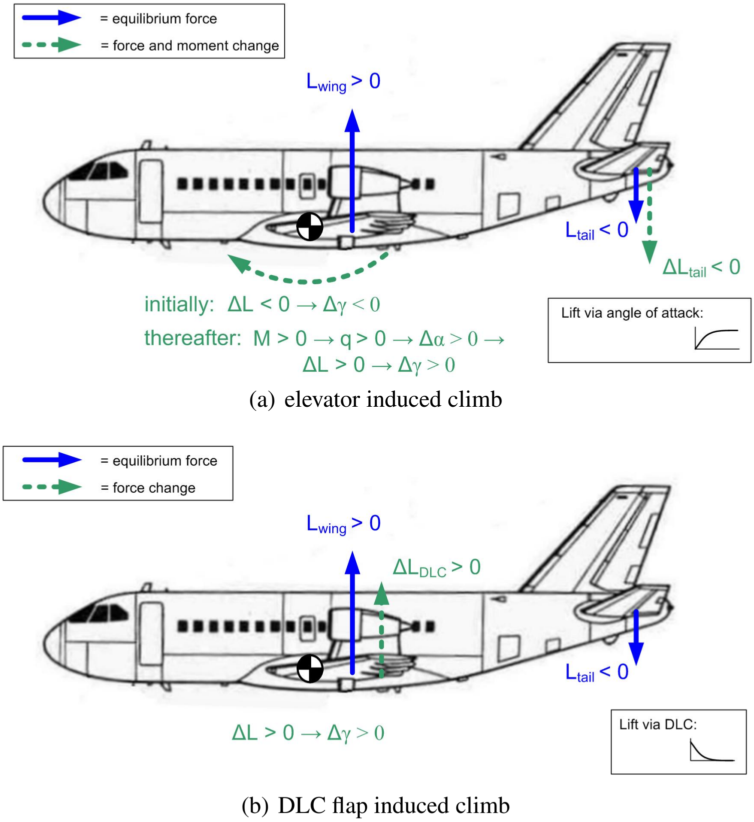
Besides improving the tracking performance, direct lift control can contribute in atmospheric disturbance rejection, since it reacts faster on tracking errors. However, this could not be verified during the test flight, since no turbulence has been encountered. Therefore, turbulence has been activated in the computer simulations. Two simulation runs have been made through the same turbulence profile, with and without direct lift control activated. During these level flight simulation runs, zero flight



(a) γ captures without DLC

(b) γ captures with DLC

Fig. 8 Comparison of time responses of flight path angle captures with and without direct lift control


= equilibrium force
= force and moment change
$L_{wing} > 0$
$L_{tail} < 0$
$\Delta L_{tail} < 0$
initially: $\Delta L < 0 \rightarrow \Delta\gamma < 0$
thereafter: $M > 0 \rightarrow q > 0 \rightarrow \Delta\alpha > 0 \rightarrow \Delta L > 0 \rightarrow \Delta\gamma > 0$
Lift via angle of attack:
= equilibrium force
= force change
$L_{wing} > 0$
$\Delta L_{DLC} > 0$
$L_{tail} < 0$
$\Delta L > 0 \rightarrow \Delta\gamma > 0$
Lift via DLC:

(a) elevator induced climb

(b) DLC flap induced climb

Fig. 9 Comparison of climb dynamics induced by elevator or DLC flaps

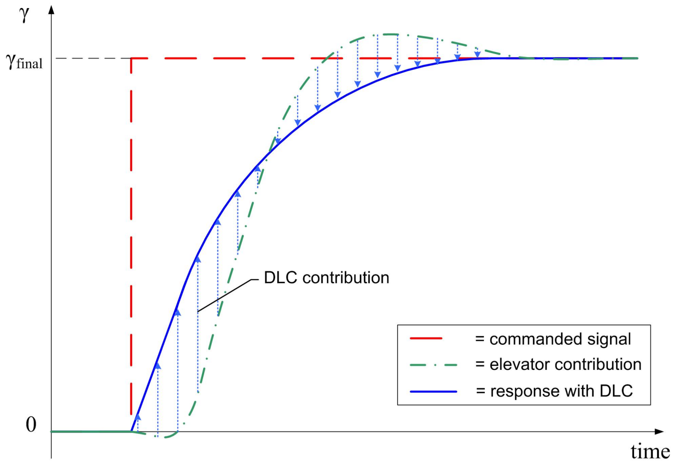


Fig. 10 Elevator and DLC contributions to flight path angle change

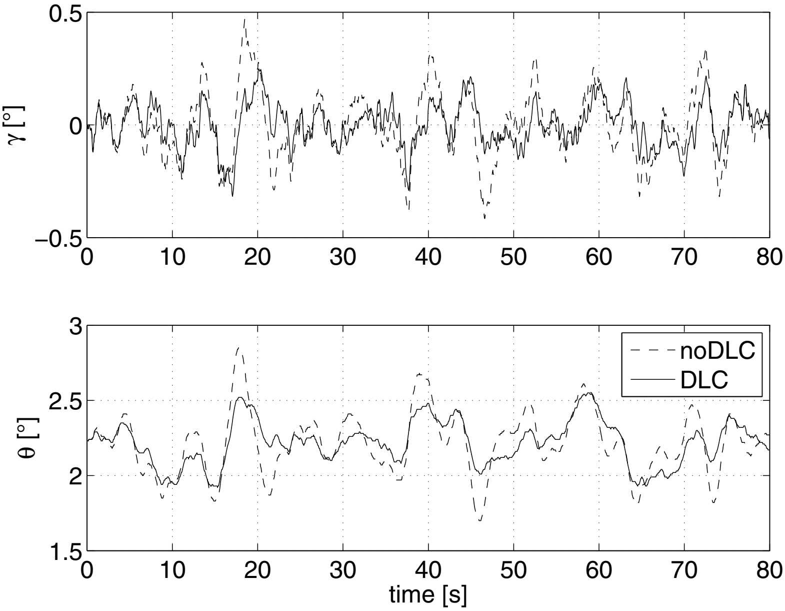


Fig. 11 Turbulence handling with and without DLC

path angle was being tracked. Fig. 11 compares the time histories of flight path angle γ and pitch attitude angle θ during this 80s simulation runs. From the figure, it can be seen that the time responses have in general smaller turbulence induced peaks, which is caused by DLC. This confirms that direct lift control effectively reduces turbulence induced disturbances and as such increases passenger comfort.

6 Flight Test Results

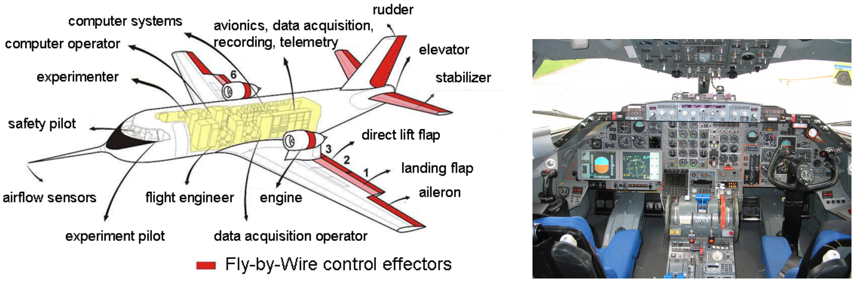
On November 3 2011, a flight test has been performed to evaluate the performance of the nonlinear autoflight control system in a relevant environment. First the ATTAS laboratory aircraft will be introduced. Thereafter the experiment procedure is explained. Finally, the flight test results are shown.

6.1 ATTAS Flying Laboratory

The aforementioned flying test-bed ATTAS (Advanced Technologies Testing Aircraft System) of DLR has been used for this test flight. ATTAS is based on a VFW 614, a 44-passenger civil transport aircraft. The original conventional mechanical control system of the basic aircraft was supplemented by the German aerospace industry (MBB) and the DLR with an electrical flight control system (Fly-By-Wire). Figure 12(a) shows an overview of the aircraft and which control effectors can be steered through the fly-by-wire system. If the ATTAS electronic flight control system is used, the safety pilot can go back to the mechanical control system at any time by pressing the safety switch. In this perspective, the safety pilot is the central instance of the ATTAS safety concept. As such, the mechanical control system of the basic aircraft is the backup system of the electric flight control system.

The left hand side cockpit controls (evaluation pilot's seat) are disconnected from the right hand side mechanical basic aircraft controls of the safety pilot. The evaluation pilot has a two-axes sidestick, shown in fig. 12(b), FBW-thrust levers, a landing flap lever, and programmable electronic primary and navigational displays available. The fly-by-wire systems architecture comprises full dual redundant control systems with four computers in each of both lanes.

The fully fueled aircraft can load 3.5 tons of test equipment. Additional to the standard control surfaces ATTAS has six direct lift control (DLC) flaps at the trailing edge of the landing flaps. These flaps permit a very fast influence on the lift. ATTAS has a maximum cruising altitude which amounts to 25000 feet (7620 m) and its maximum cruising speed is 288 knots (148.2 m/s, CAS). Furthermore, ATTAS has a very low landing speed of about 100 knots (51.4 m/s). Due to its size, its relatively spacious cabin, its additional load capacity and its flight characteristics the VFW 614 is an ideal flying test-bed for various applications. Active flight control is one of the most sophisticated fields of research of ATTAS, research results have been presented in ref. [2, 12]. Recently, flight test results of nonlinear manual and autoflight control in ATTAS have been presented in [10, 11].



(a) An overview of the technical modifications incorporated in ATTAS. Besides the conventional elevator, rudder, and aileron control surfaces, the engines, six direct lift control flaps, landing flaps, and stabilizer are under fly-by-wire (FBW) control. Source: DLR

(b) ATTAS cockpit view, with experiment pilot seat on the left and safety pilot seat on the right; Source: DLR

Fig. 12 DLR VFW-614 ATTAS

6.2 Experiment Procedure

The DLC related experiments have been performed in one flight condition. Altitude was around FL150, speed regime was 180 KIAS with flaps 1°, gear was up and spoilers in. Autotrim and autothrottle were both active during the tests. After taking off at Braunschweig Airport, ATTAS has flown north to the coastline of the Baltic Sea, and then south back to the airport, as illustrated in fig. 13. The experiments have been performed throughout the whole flight.

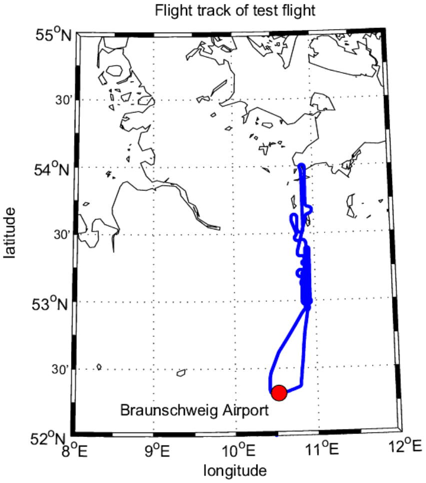
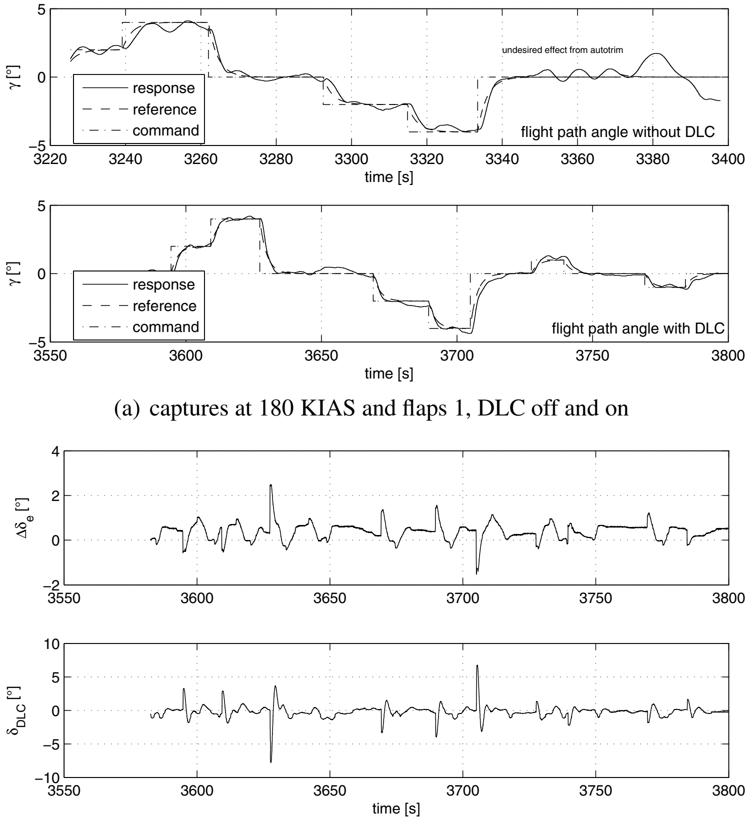


Fig. 13 Flight track of test flight, date: 3/11/2011

During the experiments, the evaluation pilot executed various flight path angle captures in order to verify the performance of the tracking mode with and without DLC. These captures are initiated by the experiment pilot through the mode control panel. In order to obtain pure step commands, an activation switch has been incorporated as well, which needs to be pressed after the requested value for the angle change has been dialed. More information about the other captures and experiments during this flight can be found in Ref. [11].

6.3 Results

The flight path angle captures are shown in Fig. 14. The oscillating behaviour of the flight path angle after angle capture is caused by the perturbing autotrim actions. The influence of the direct lift control (DLC) flaps is illustrated in Fig. 14(a). The corresponding elevator and direct lift control surface actions can be seen in Fig. 14(b). By comparing the graphs in Fig. 14, it can be seen that the DLC's assist in tracking the reference flight path angle γ_{ref} more accurately. Comparing the direct lift control surfaces with the elevator behaviour in Fig. 14(b) reveals opposite deflections. This is because the DLC's have a direct influence on the lift force, and thus on the flight path angle. This is more efficient than the elevator, which influences the lift force primarily through the angle of attack, which causes a time delay in the responses. Moreover, non minimum phase behaviour is present between elevator and lift force, which is not the case for the DLC's.



(a) captures at 180 KIAS and flaps 1, DLC off and on

(b) corresponding elevator and DLC deflections

Fig. 14 Flight path angle captures with and without direct lift control

7 Conclusions and Recommendations

An improvement to previously developed nonlinear autoflight control laws has been designed, based on direct lift control. The previously developed nonlinear dynamic inversion based control laws are built up in a modular way consisting of three layers, according to the principle of time scale separation. As such, adaptation of the controller to different aircraft types does not require major conversion work of the controller. The extension with direct lift control adds an additional control channel in this setup and preserves the modular structure. These control laws have been flight tested on the DLR ATTAS fly-by-wire testbed. Flight tests have shown that direct lift control assists in tracking the flight path angle γ more accurately, since time delays and non-minimum phase behaviour are effectively eliminated. More precisely, direct lift control assists in establishing flight path angle changes without time delay and without pitching. Moreover, the system increases flight path accuracy as well as passenger comfort during turbulence, by reducing turbulence induced perturbations of the flight path angle γ as well as of the pitch attitude angle θ.

This direct lift control strategy has been designed for and flight tested in the ATTAS fly-by-wire testbed, since this aircraft has dedicated direct lift control flaps, as explained in the introduction. Currently planned new aircraft designs have no dedicated direct lift control flaps. However, the results which have been obtained in this research warrant further investigations if the same performance could be achieved by steering the spoilers or similar appropriate control surfaces on the aircraft wing in a symmetrical manner.

Acknowledgements. The authors would like to acknowledge the following colleagues at DLR Braunschweig, for their assistance in preparing and test flying the nonlinear autoflight controller: controller software implementer and ground simulator operator Dirk Leißling, safety pilot Hans-Jürgen Berns, experiment pilot Stefan Seydel, flight test engineer Michael Preß and flight engineer Regina Gebhard. Last but not least, we would like to acknowledge the "Programmdirektion Luftfahrt" of the German Aerospace Center DLR.

References

1. Realattas benchmark definition. Technical Report BRPR-CT98-0627/TP-04v1, REAL Consortium (2000)
2. Bauschat, J.-M., Monnich, D., Willemsen, W., Looye, G.: Flight testing robust autoland control laws. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2001-4208 (2001)
3. Buchholz, J.J., Bauschat, J.-M., Hahn, K.-U., Pausder, H.J.: ATTAS & ATTHeS in-flight simulators. In: AGARD CP -577: Flight Simulation - Where are the Challenges? Braunschweig (1996)
4. Fischenberg, D.: Attas in-air aerodynamikmodell. Institutsbericht IB 111-97/31, German Aerospace Center DLR, Institut fuer Flugmechanik, Braunschweig (September 1997)
5. Gerrits, M.: Direct lift control for the cessna citation ii. Master's thesis, Eindhoven University of Technology (1994)
6. Kelley, W.W.: Simulator evaluation of a flight-path-angle control system for a transport airplane with direct lift control. Technical report NASA-TP-1116, NASA Langley Research Center (1978)
7. Lauterbach, M., Luckner, R.: Vergleich verschiedener regelstrategien für die stufenlose verstellung der hinterkantenklappen im landeanflug. In: Deutscher Luft- und Raumfahrtkongress, pp. 307–318 (2010) (in German)
8. Lombaerts, T.J.J.: Fault Tolerant Flight Control. A Physical Model Approach. PhD thesis, Delft University of Technology (2010)
9. Lombaerts, T.J.J., Huisman, H.O., Chu, Q.P., Mulder, J.A., Joosten, D.A.: Nonlinear reconfiguring flight control based on on-line physical model identification. AIAA Journal of Guidance, Control and Dynamics 32(3), 727–748 (2009)
10. Lombaerts, T.J.J., Looye, G.H.N.: Design and flight testing of manual nonlinear flight control laws. In: AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA-2011-6469. American Insitutue of Aeronautics and Astronautics (2011)
11. Lombaerts, T.J.J., Looye, G.H.N.: Design and flight testing of nonlinear autoflight control laws. In: AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA-2012-4982. American Insitutue of Aeronautics and Astronautics (2012)

12. Looye, G., Joos, H.-D.: Design of autoland controller functions with multiobjective optimization. AIAA Journal of Guidance, Control and Dynamics 29(2), 475–484 (2006)
13. Looye, G.H.N.: An Integrated Approach to Aircraft Modelling and Flight Control Law Design. PhD thesis, Delft University of Technology (2007)
14. Lykken, L.O., Shah, N.M.: Direct lift control for improved automatic landing and performance of transport aircraft. Journal of Aircraft 9(5), 325–332 (1972)
15. McNeill, W.E., Gerdes, R.M., Innis, R.C., Ratcliff, J.D.: A flight study of the use of direct-lift-control flaps to improve station keeping during in-flight refueling. Technical Memorandum TM X-2936, NASA Ames Research Center, Moffett Field, CA94035 (October 1973)
16. Mulder, J.A., van der Vaart, J.C.: Aircraft responses to atmospheric turbulence. Delft University of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands (May 1998)
17. Mulder, J.A., van Staveren, W.H.J.J., van der Vaart, J.C., de Weerdt, E.: AE3-302 Flight Dynamics, Lecture Notes. Delft University of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands (January 25, 2006)
18. Rouwhorst, W.F.J.A.: Robust and effcient autopilots control laws design, demonstrating the use of modern robust control design methodologies in the autoland system design process - the real project. In: Proceedings of the Aeronautics Days 2001, Hamburg, Germany (January 2001)

A Frequency-Limited $\mathscr{H}_2$ Model Approximation Method with Application to a Medium-Scale Flexible Aircraft

Pierre Vuillemin, Charles Poussot-Vassal, and Daniel Alazard

Abstract. In this paper, the problem of approximating a medium-scale MIMO LTI dynamical system over a bounded frequency range is addressed. A new method grounded on the SVD-Tangential model order reduction framework is proposed. Based on the frequency-limited gramians defined in [5], the contribution of this paper is to propose a *frequency-limited iterative SVD-Tangential interpolation algorithm* (FL-ISTIA) to achieve frequency-limited model approximation without involving weighting filters. The efficiency of the approach is addressed both on standard benchmark and on an industrial flexible aircraft model.

1 Introduction

1.1 Motivation

Computer-based modeling software are often used in order to accurately capture the mathematical model of physical systems or phenomena. They enable to handle complex systems with an enhanced accuracy. These models allow time and cost saving in the development process, but they often involve a large number of variables and thus require a lot of resources when analysed or simulated. Some modern *analysis* or *synthesis* tools may thereby become inefficient for such high dimensional models. A relevant approach to solve this issue is to *approximate the model* with a smaller one.

Pierre Vuillemin · Daniel Alazard
Université de Toulouse and Onera - The French Aerospace Lab,
F-31055 Toulouse, France
e-mail: pierre.vuillemin@onera.fr, daniel.alazard@isae.fr

Charles Poussot-Vassal
Onera - The French Aerospace Lab, F-31055 Toulouse, France
e-mail: charles.poussot-vassal@onera.fr

The reduction process can be subject to several constraints which depend on the purpose of the model. A commonly used constraint is the closeness between the reduced-order model input/output behaviour and the full-order one over all frequencies. Though it is a very interesting problem (see [9, 15, 17]), forcing models to be close over all frequencies may be too binding. Indeed (i) some frequencies are physically meaningless and can be viewed as uncertainties, (ii) in practice, actuators and sensors bandwidth are limited which make some frequencies irrelevant for control purpose and (iii) when vibration control has to be performed, some frequencies are more specifically of interest. Therefore considering the problem of reducing the full-order model such that a good approximation is found *over a bounded frequency range* can be more appropriate and appealing for engineers. This is the problem treated in this paper.

1.2 Projection-Based Problem Statement

The reduction problem which consists in approximating a large-scale model by projection is recalled in Problem 1.

Problem 1. Given a continuous, stable and strictly proper MIMO LTI dynamical model Σ defined as

$$\Sigma := \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_u}$ and $C \in \mathbb{R}^{n_y\times n}$. The projection-based model order reduction problem consists in finding $V, W \in \mathbb{R}^{n\times r}$ with $W^T V = I_r$ such that the reduced-order model $\hat{\Sigma}$ of order $r \ll n$ defined as

$$\hat{\Sigma} := \begin{cases} \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t) \\ \hat{y}(t) = \hat{C}\hat{x}(t) \end{cases} \tag{2}$$

where $\hat{A} = W^T AV$, $\hat{B} = W^T B$ and $\hat{C} = CV$, accurately reproduces the behaviour of the full-order system Σ over the whole frequency domain.

The accuracy can be evaluated through the $\mathscr{H}_2$-norm of the error system $\Sigma - \hat{\Sigma}$. This measure, called mismatch error, is a good indicator of the global error between the models and is commonly used in many research papers [9, 15, 17].

1.3 Frequency-Limited Model Approximation Problem

In this paper, a similar formulation is addressed for the frequency-limited case (see Problem 2).

Problem 2. Given a continuous, stable and strictly proper MIMO LTI dynamical model Σ as in (1), the projection-based frequency limited approximation problem consists in finding projectors $V, W \in \mathbb{R}^{n\times r}$ in order to construct the reduced-order model $\hat{\Sigma}$ as in (2) such that $\hat{\Sigma}$ well approximates Σ over a given bounded frequency range.

In this paper, this problem will be tackled for the frequency range $[0,\omega]$ because low frequencies are particularly of interest for the intended applications. The accuracy of the approximation over $[0,\omega]$ will be evaluated through the frequency-limited $\mathscr{H}_2$-norm proposed in [11] and recalled later in Definition 2.

1.4 Paper Structure

The paper is divided as follow. In Section 2 some preliminary results on the standard $\mathscr{H}_2$ model approximation are recalled. Then in Section 3, specific tools for frequency-limited model approximation are presented. In Section 4, the proposed frequency-limited approximation method is introduced. It is applied and compared on standard benchmark models and on an industrial flexible aircraft model in Section 5. Section 6 concludes this article.

2 Preliminary Results on $\mathscr{H}_2$ Model Approximation

2.1 $\mathscr{H}_2$-Optimal Model Approximation

The model approximation problem formulated previously in Problem 1 can be viewed as the minimization of the following entity

$$\mathscr{J}_{\mathscr{H}_2}\left(\hat{A},\hat{B},\hat{C}\right) = \|\Sigma - \hat{\Sigma}\|^2_{\mathscr{H}_2} \tag{3}$$

which represents the mismatch error between the full-order and the reduced-order models in terms of $\mathscr{H}_2$-norm, i.e. over the whole frequency range (see Definition 1).

Definition 1. *The $\mathscr{H}_2$-norm of a stable and strictly proper system Σ whose transfer function is $H(s) = C(sI_n - A)^{-1}B$, is given by*

$$\begin{aligned}\|H\|^2_{\mathscr{H}_2} &= \frac{1}{2\pi j}\int_{-\infty}^{\infty} \mathbf{trace}\left(H(j\omega)H(-j\omega)^T\right) ds \\ &= \mathbf{trace}\left(B^T \mathscr{Q} B\right) \\ &= \mathbf{trace}\left(C \mathscr{P} C^T\right)\end{aligned} \tag{4}$$

where $\mathscr{P}$ and $\mathscr{Q}$ are the controllability and the observability gramians given in the frequency domain by the following integrals:

$$\mathscr{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty} T(\nu)BB^T T^*(\nu)d\nu \tag{5a}$$

$$\mathscr{Q} = \frac{1}{2\pi}\int_{-\infty}^{\infty} T^*(\nu)C^T C T(\nu)d\nu \tag{5b}$$

with $T(\nu) = (j\nu I - A)^{-1}$.

Minimizing $\mathscr{J}_{\mathscr{H}_2}$ is a non-convex problem, thus finding a global minimizer is a complex task. Finding a local minimizer is a more tractable problem. The most commonly used approach consists in derivating the first-order necessary optimality conditions which have been first addressed by Wilson [17]. Based on the interpolatory framework of Grimme [6], the *Iterative Rational Krylov Algorithm* (IRKA) proposed in [9] enables to fill these conditions and leads to a local minimizer. However due to numerical issues, it is rather dedicated to SISO systems. For MIMO systems, the tangential interpolatory framework [4] seems to be more appropriate. Equivalent first-order optimality conditions have also been derived for this case [15] and are recalled in Theorem 1.

Theorem 1. *If $\nabla_{\hat{A}}\mathscr{J}_{\mathscr{H}_2} = 0, \nabla_{\hat{B}}\mathscr{J}_{\mathscr{H}_2} = 0$ and $\nabla_{\hat{C}}\mathscr{J}_{\mathscr{H}_2} = 0$, which are the gradients of $\mathscr{J}_{\mathscr{H}_2}$ with respect to $\hat{A}$, $\hat{B}$ and $\hat{C}$ respectively, then the following tangential interpolation conditions are satisfied for $i = 1,\dots,r$:*

$$\begin{aligned} [H(-\hat{\lambda}_i) - \hat{H}(-\hat{\lambda}_i)]\hat{b}_i &= 0 \\ \hat{c}_i^*[H(-\hat{\lambda}_i) - \hat{H}(-\hat{\lambda}_i)] &= 0 \\ \hat{c}_i^* \tfrac{d}{ds}[H(s) - \hat{H}(s)]|_{s=-\hat{\lambda}_i}\hat{b}_i &= 0 \end{aligned} \tag{6}$$

where the $\hat{\lambda}_i$ are the eigenvalues of $\hat{A}$, $\{\hat{b}_1,\dots,\hat{b}_r\} = \hat{B}^T R$ and $\{\hat{c}_1,\dots,\hat{c}_r\} = \hat{C}L$ (where L and R are the left and right eigenvectors associated to $\hat{\lambda}_i$).

Theorem 1 expresses the necessary conditions to find a local minimum of $\mathscr{J}_{\mathscr{H}_2}$. Hence the optimal model approximation problem consists in finding $\left\{\hat{\lambda}_i, \hat{c}_i, \hat{b}_i\right\}$ such that (6) is satisfied. Theorem 2 then makes the link with Problem 1 and shows how the projectors V and W are constructed to fulfil these conditions.

Theorem 2. *Let $V \in \mathbb{C}^{n\times r}$ and $W \in \mathbb{C}^{n\times r}$ be full rank matrices such that $W^T V = I_r$. Let $\sigma_i \in \mathbb{C}^r$, $\hat{b}_i \in \mathbb{C}^{n_u}$ and $\hat{c}_i \in \mathbb{C}^{n_y}$ (for $i = 1,\dots,r$) be given sets of interpolation points and left and right tangential directions, respectively. Assume that points σ_i are selected such that $\sigma_i I_n - A$ are invertible. If, for $i = 1,\dots,r$,*

$$\begin{aligned} (\sigma_i I_n - A)^{-1} B\hat{b}_i \in \boldsymbol{span}(V) \\ and\ (\sigma_i I_n - A^T)^{-1} C^T \hat{c}_i^* \in \boldsymbol{span}(W), \end{aligned} \tag{7}$$

then, the reduced-order system $\hat{H}(s)$ satisfies the tangential interpolation conditions given in Theorem 1.

The *Iterative Tangential Interpolation Algorithm* (ITIA) suggested in [15] is a very efficient way to achieve Theorem 2. The IRKA and ITIA are numerically efficient and lead to local minimizers of $\mathscr{J}_{\mathscr{H}_2}$. Nevertheless they theoretically do not preserve stability of the full-order model[1] and can lead to poor approximant when

[1] Yet in practice it is often the case. Moreover algorithmic procedures such as restarting can be used to avoid instability.

applied to ill-conditioned models. Moreover for approximating medium-scale dynamical systems, numerical efficiency is less crucial than it can be in (very)large-scale cases. That is why it may be more adequate to use a method which is heavier than IRKA or ITIA from a computational point of view but which offers more guarantees and more robustness to parameters selection. Such a method has first been proposed by Gugercin in [7] and is called *Iterative SVD-Rational Krylov Algorithm* (ISRKA). It requires to compute only one gramian and it is directly applicable to SISO, MISO and SIMO systems. A similar algorithm for MIMO systems, called *Iterative SVD Tangential Interpolation Algorithm* (ISTIA) has been proposed in [13]. It is the basis of this work and for sake of completeness, its main properties are recalled there after.

2.2 ISTIA

This algorithm consists in using one single gramian to construct one of the two projectors involved in the approximation by projection. Indeed one projector is designed by solving one single Lyapunov equation while the second one is iteratively constructed to achieve one sided tangential interpolation and thus fulfil a subset of the optimality conditions presented in Theorem 1. For instance, V and W can be constructed such that

$$\mathbf{span}\,(V) = \left[(\sigma_1 I_n - A)^{-1}B\hat{b}_1, \dots, (\sigma_r I_n - A)^{-1}B\hat{b}_r\right] \tag{8}$$

where σ_i are the shift points and $\hat{b}_i$ corresponding right tangential directions, and

$$W = \mathscr{Q}V(V^T\mathscr{Q}V)^{-1} \tag{9}$$

where $\mathscr{Q}$ is the observability gramian. See [7] and [13] for more details on the selection of interpolation points and for the complete version of the algorithm.

This method is numerically more expensive than the IRKA but it offers also more guarantees. Indeed, if the full-order model is stable, then the reduced-order one will be stable as well. The proof can be found in [7]. It consists in considering that $\mathscr{Q} = I_n$. Hence, $W = V$ and $V^TV = I_r$. The Lyapunov equation becomes,

$$A^T + A + C^TC = 0 \tag{10}$$

By left and right multiplying (10) by V^T and V, it comes

$$\hat{A}^T + \hat{A} + \hat{C}^T\hat{C} = 0 \tag{11}$$

which indicates, by inertia results [12], that $\hat{A}$ is stable. For the same reasons as in [7], $\hat{A}$ is even asymptotically stable.

3 Preliminary Results on Frequency-Limited Model Approximation

So far, only the $\mathcal{H}_2$ optimal model approximation has been considered but a lot of studies concern the model approximation over a bounded frequency range. Useful tools related to this issue are presented in this section.

3.1 The Frequency-Weighted Approach

The most common approach to tackle the issue of reducing a model over a bounded frequency interval consists in considering input and/or output filters $W_i(s)$ and $W_o(s)$ so that the reduction is achieved on the filtered full-order system $\tilde{H}(s)$ given by

$$\tilde{H}(s) = W_o(s)H(s)W_i(s) \tag{12}$$

where $H(s) = C(sI_n - A)^{-1}B$. The weighted model reduction problem has often been tackled by weighted versions of the balanced truncation, see for instance [8] and references therein for an overview of these methods. More recently, this problem has been tackled from an interpolatory point of view, see [3] and [2].

Despite interesting results, the use of weights is very limiting since their choose is a time consuming and challenging task for engineers. For instance to achieve frequency-weighted balanced truncation, weights have to be stable and minimum phase filters. To alleviate this practical difficulty, a weight-free approach is preferred in this paper.

3.2 Frequency-Limited Gramians and Balanced Truncation

In [5], the authors proposed to narrow the frequency range of the integrals in (5a) and (5b) in order to get gramians in frequency interval $[0, \omega]$:

$$\mathscr{P}_\omega = \frac{1}{2\pi}\int_{-\omega}^{\omega} T(\nu)BB^T T^*(\nu)d\nu \tag{13a}$$

$$\mathscr{Q}_\omega = \frac{1}{2\pi}\int_{-\omega}^{\omega} T^*(\nu)C^T CT(\nu)d\nu \tag{13b}$$

with $T(\nu) = (j\nu I_n - A)^{-1}$.

These gramians are solutions of the two following Lyapunov equations:

$$A\mathscr{P}_\omega + \mathscr{P}_\omega A^T + W_c(\omega) = 0 \tag{14a}$$

$$A^T\mathscr{Q}_\omega + \mathscr{Q}_\omega A + W_o(\omega) = 0 \tag{14b}$$

where

$$W_c(\omega) = S(\omega)BB^T + BB^T S^*(\omega) \tag{15a}$$

$$W_o(\omega) = S^*(\omega)C^T C + C^T C S(\omega) \tag{15b}$$

and

$$\begin{aligned} S(\omega) &= \frac{1}{2\pi}\int_{-\omega}^{\omega} T(\nu)d\nu \\ &= \frac{j}{2\pi}\mathbf{log}\left((A + j\omega I_n)(A - j\omega I_n)^{-1}\right) \end{aligned} \tag{16}$$

A direct application of the frequency-limited gramians in model order reduction is the frequency-limited balanced truncation presented in [5]. It consists firstly in balancing $\mathscr{P}_\omega$ and $\mathscr{Q}_\omega$, that is to say to find a basis so that both gramians are equals and diagonals:

$$\mathscr{P}_\omega = \mathscr{Q}_\omega = \mathbf{diag}\left(\sigma_{n_1} I_{n_1}, \ldots, \sigma_{n_q} I_{n_q}\right) \tag{17}$$

where σ_i is a singular value with multiplicity n_i. Then the model is classically truncated to obtain the reduced-order model.

Since $W_c(\omega)$ and $W_o(\omega)$ are not positive semi-definite, the frequency-limited gramians $\mathscr{P}_\omega$ and $\mathscr{Q}_\omega$ are not guaranteed to be positive semi-definite (see [12]). Hence the reduced-order model obtained this way might be unstable. To preserve stability, [8] has proposed a modification to this method but it drastically impacts the quality of the reduced-order model.

As it is mentioned in [8], using frequency-limited gramians for balanced truncation can be viewed as a frequency-weighted balanced truncation method with *perfect filters*.

3.3 $\mathscr{H}_{2,\omega}$-Norm: Frequency-Limited $\mathscr{H}_2$-norm

The $\mathscr{H}_2$-norm is a convenient metric for measuring the quality of an approximant *over the whole frequency range*, however it is less relevant if the approximant has to be good only over a finite frequency interval $[0, \omega]$. In this case, another metric has to be considered.

A frequency-bounded $\mathscr{H}_2$-norm has been addressed in [1] and recalled more recently in [11] as a restriction of the $\mathscr{H}_2$-norm over the frequency range $[0, \omega]$. Its state-space representation directly comes from the definition of the frequency-limited gramians given in Definition 2.

Definition 2. *Given a stable and strictly proper MIMO linear dynamical system Σ with $H(s) = C(sI_n - A)^{-1}B$, the $\mathscr{H}_{2,\omega}$-norm is defined as follow*

$$\begin{aligned} \|H\|^2_{\mathscr{H}_{2,\omega}} &= \frac{1}{2\pi}\int_{-\omega}^{\omega} \mathbf{trace}\left(H(j\nu)H(-j\nu)^T\right)d\nu \\ &= \mathbf{trace}\left(C\mathscr{P}_\omega C^T\right) \\ &= \mathbf{trace}\left(B^T \mathscr{Q}_\omega B\right) \end{aligned} \tag{18}$$

where $\mathscr{P}_\omega$ and $\mathscr{Q}_\omega$ are the frequency-limited gramians defined by (13a) *and* (13b).

Remark 1. *Frequency-limited gramians can also be expressed over the frequency interval $\Omega = [\omega_1, \omega_2]$. Indeed, $\mathscr{P}_\Omega = \mathscr{P}_{\omega_2} - \mathscr{P}_{\omega_1}$ and $\mathscr{Q}_\Omega = \mathscr{Q}_{\omega_2} - \mathscr{Q}_{\omega_1}$. Hence a restriction of the $\mathscr{H}_2$-norm over the interval Ω can easily be expressed in a similar way.*

Property 1. *If $H(s)$ is a stable and strictly proper linear dynamical system, then its frequency-bounded $\mathscr{H}_2$-norm tends towards its $\mathscr{H}_2$-norm when the frequency bound tends towards infinity,*

$$\|H\|_{\mathscr{H}_{2,\omega}} \underset{\omega\to\infty}{\longrightarrow} \|H\|_{\mathscr{H}_2} \tag{19}$$

Proof. Applying the residue theorem in (18) leads to the result. Another proof can be found in [16]. □

To illustrate how the $\mathscr{H}_{2,\omega}$-norm behaves as a function of ω, the Los-Angeles hospital model is used (see [10]). It has 48 states, 1 input and 1 output. Its frequency response and its $\mathscr{H}_{2,\omega}$-norm are computed for several values of ω on Figure 1. It enables to illustrate Property 1 and the fact that the $\mathscr{H}_{2,\omega}$-norm evolves by steps. When ω crosses the abscissa of a peak of the frequency response, the $\mathscr{H}_{2,\omega}$-norm of $H(s)$ crosses a step. This can be viewed as the contribution of the gain of the peak in the global input/output energy represented by the $\mathscr{H}_2$-norm.

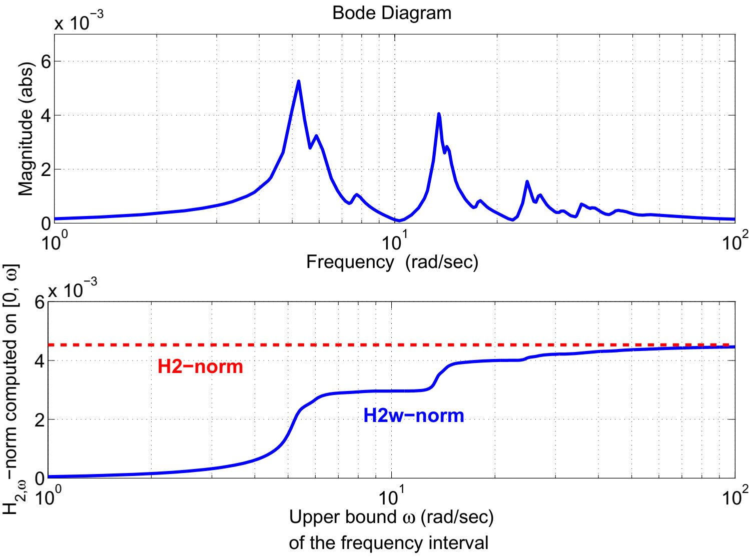


Fig. 1 Evolution of the $\mathscr{H}_{2,\omega}$-norm against the frequency (Los Angeles hospital model)

4 Frequency-Limited Iterative SVD-Tangential Interpolation Algorithm

The proposed algorithm, namely the *Frequency-Limited Iterative SVD-Tangential Interpolation Algorithm*, or FL-ISTIA (see Algorithm 1) is very similar to the ISTIA. Indeed, one projector is built through tangential interpolation (step 1 and 9) whereas the other is obtained through the computation of a gramian. The main difference lies in the fact that the gramian used to build the second projector is a frequency-limited gramian.

For numerical purpose, the right projector W is obtained by enforcing *orthogonality*, as in step 3. Then, from step 4 to 11, the construction of projectors is repeated by using as new interpolation points the *mirror images* of the eigenvalues of the reduced-order model, and, as new interpolation directions, the *right eigenvectors* associated with these eigenvalues (steps 6-8). The process is repeated until the interpolation points variation is smaller than a user defined tolerance ε.

Algorithm 1. Frequency-Limited Iterative SVD-Tangential Interpolation Algorithm (FL-ISTIA)

```
Require: $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times n_u}, C \in \mathbb{R}^{n_y\times n}, R \in \mathbb{R}^{+*}$, $\{\sigma_1^{(0)},\dots,\sigma_r^{(0)}\} \in \mathbb{C}^{n\times r}$ with $|\sigma_i^{(0)}| \leq R, i=1,\dots,r$, $\{\hat{b}_1,\dots,\hat{b}_r\} \in \mathbb{C}^{n\times r}$, $\varepsilon > 0$
 1: Construct,
        $\mathbf{span}(V) = \left[(\sigma_1^{(0)} I_n - A)^{-1} B\hat{b}_1, \dots, (\sigma_r^{(0)} I_n - A)^{-1} B\hat{b}_r\right]$
 2: Solve $\mathscr{Q}_\omega A + A^T \mathscr{Q}_\omega + W_o(\omega) = 0$ in $\mathscr{Q}_\omega$
 3: Compute $W = \mathscr{Q}_\omega V (V^T \mathscr{Q}_\omega V)^{-1}$
 4: while $|\sigma^{(i)} - \sigma^{(i-1)}| > \varepsilon$ do
 5:     $i \leftarrow i+1, \hat{A} = W^T AV, \hat{B} = W^T B$
 6:     Compute $\hat{A}X = \mathbf{diag}(\lambda(\hat{A}))X$
 7:     Compute $\left[\hat{b}_1,\dots,\hat{b}_r\right] = \hat{B}^T X^{-T}$
 8:     Set $\sigma^{(i)} = -\lambda(\hat{A})$
 9:     Construct,
            $\mathbf{span}(V) = \left[(\sigma_1^{(i)} I_n - A)^{-1} B\hat{b}_1, \dots, (\sigma_r^{(i)} I_n - A)^{-1} B\hat{b}_r\right]$
10:     Compute $W = \mathscr{Q}_\omega V (V^T \mathscr{Q}_\omega V)^{-1}$
11: end while
12: Construct $\hat{\Sigma} : (W^T AV, W^T B, CV)$
Ensure: $V, W \in \mathbb{R}^{n\times r}, W^T V = I_r$ and $\mathbf{Re}\Big(\lambda(\hat{A})\Big) < 0$
```

Some remarks about this algorithm can be addressed :

- Unlike the ISTIA, the stability of the reduced-order model cannot be guaranteed. Indeed, since $\mathscr{Q}_\omega$ (step 2) is not guaranteed to be positive semi-definite, the reasoning done previously with the ISTIA in Section 2.2 is no longer valid. Yet in practice, instability has not been observed and numerical procedures such as *restarting* can be used to alleviate this drawback.
- The initial shift points are selected so that their modulus is less than the frequency bound ω, i.e. $|\sigma_i^{(0)}| \leq \omega$, $i = 1, \ldots, r$. It is done to favour the interpolation of the full-order model under this bound. A similar constraint could be imposed on the following interpolation points (step 8) but the selection of tangential directions would then become an issue.
- The frequency-limited controllability gramian $\mathscr{P}_\omega$ can identically be used instead of the observability one, $\mathscr{Q}_\omega$. In this case, the tangential subspace to be constructed is $\mathbf{span}(W) = \left[(\sigma_1^{(i)} I_n - A^T)^{-1} C^T \hat{c}_1, \ldots, (\sigma_r^{(i)} I_n - A^T)^{-1} C^T \hat{c}_r\right]$ (where $\{\hat{c}_1, \ldots, \hat{c}_r\} = \hat{C}X$).
- As in all Krylov-like procedures, to obtain real valued projection V and W matrices and increase computation speed, the starting shift grid should be either real or complex conjugate. Indeed, one can use the fact that, if, $v_2 = v_1^*$, then $\mathbf{span}[v_1, v_2] = \mathbf{span}[\mathbf{Re}(v_1), \mathbf{Im}(v_1)]$.
- Since this procedure requires to solve a n-th order Lyapunov equation, it is limited to medium-scale dynamical systems. It could be extended to larger systems with the use of low rank approximations of the gramian.
- The FL-ISTIA is equivalent to the ISTIA as ω increases. Indeed, as ω increases, the realisation given by the FL-ISTIA tends (element-wise) towards the one given by the ISTIA. This comes from the fact that frequency-limited gramians tends (element-wise) towards infinite gramians as ω tends towards infinity.

5 Applications

In this section, the Iterative SVD-Tangential Interpolation Algorithm (ISTIA), the frequency-limited balanced truncation (FL-BT) and the Frequency-Limited Iterative SVD-Tangential Interpolation Algorithm (FL-ISTIA) are compared through two standard benchmarks and one industrial flexible aircraft model.

The quality of the approximation over $[0, \omega]$ is evaluated through the $\mathscr{H}_{2,\omega}$-norm (see Definition 2) of the relative error ε_ω, i.e.:

$$\varepsilon_\omega = \frac{\|\Sigma - \hat{\Sigma}\|_{\mathscr{H}_{2,\omega}}}{\|\Sigma\|_{\mathscr{H}_{2,\omega}}} \tag{20}$$

5.1 *Standard Benchmark Models Case*

As a first application of the Iterative Frequency-Limited SVD Tangential Interpolation Algorithm (FL-ISTIA), the clamped beam model is used. It is a standard benchmark model [10] with 348 states, 1 input and 1 output.

The clamped beam model is reduced to order $r = 12$ using the three reduction methods. The upper bound ω of the frequency interval of reduction $[0, \omega]$ is gradually increased from 2rad/s to 20rad/s. Results are represented in Figure 2.

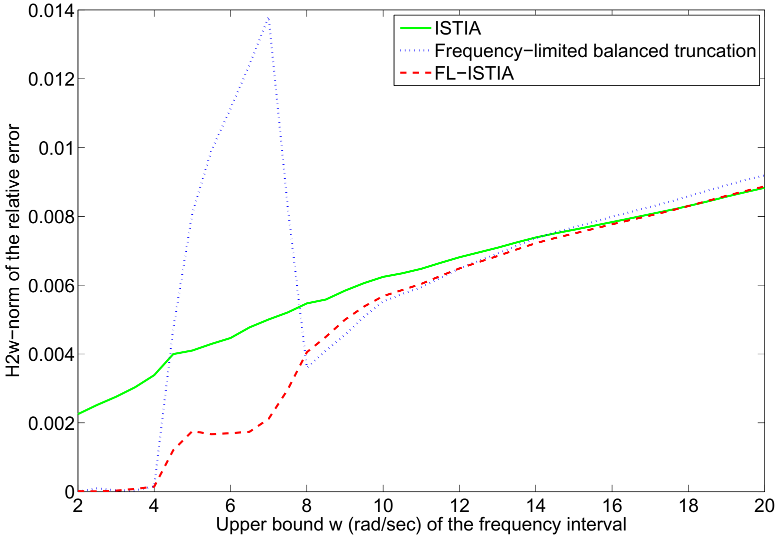


Fig. 2 $\mathscr{H}_{2,\omega}$-norm of the relative error against the upper frequency bound ω (clamped beam model, $r = 12$)

On this example, the FL-BT and the FL-ISTIA are quite similar excepted from 4rad/s to 8rad/s where the first method fails to correctly reduce the model. This may come from numerical issues related to the computation of frequency-limited gramians or the balancing of the system. This suggests that the FL-ISTIA is numerically more reliable.

Figure 2 also clearly illustrates the fact that the FL-ISTIA and the ISTIA become equivalent as ω increases since they lead to the same reduced-order model. The frequency-limited aspect of the approximation methods considered here is well illustrated by Figure 3 where the frequency responses of the error systems are plotted. The upper bound of the frequency interval used in this case was $\omega = 2,5$rad/s. It appears that the error is very low from 0 to $\omega = 2,5$rad/s and it rises after this bound.

Similar results can be observed when the procedure is applied to the Los Angeles hostpital model (see Figure 4).

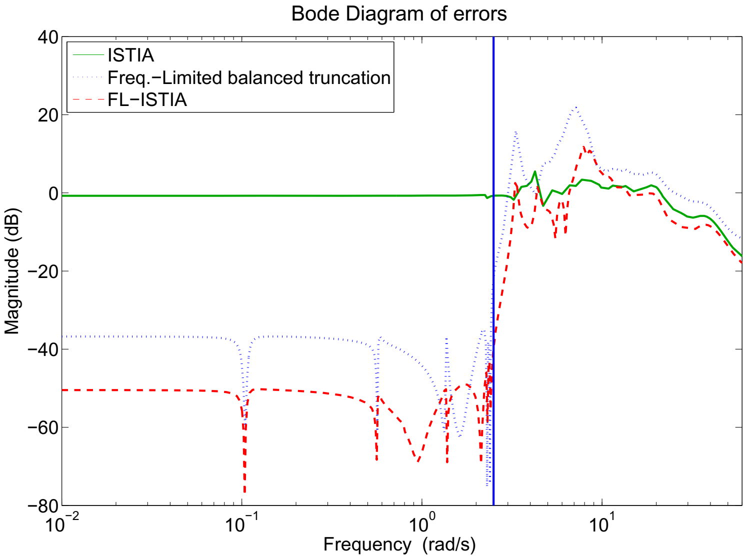


Fig. 3 Frequency responses of the error system for $\omega = 2,5$rad/s (clamped beam model, $r = 12$)

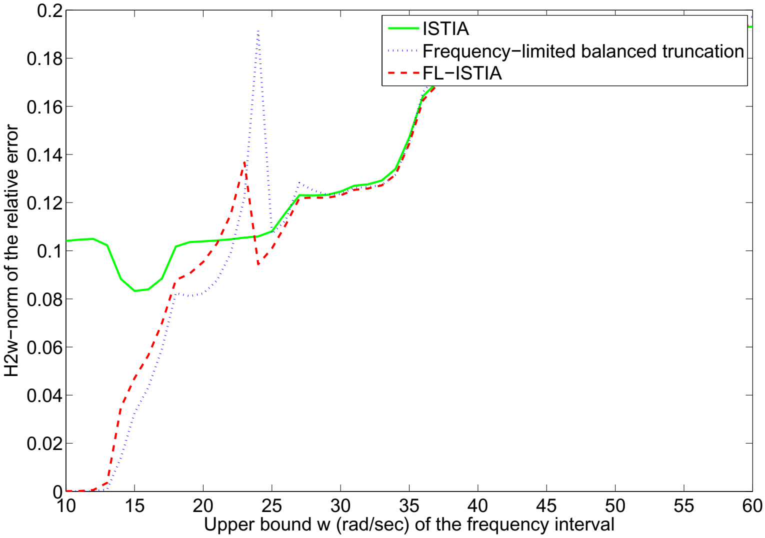


Fig. 4 $\mathscr{H}_{2,\omega}$-norm of the relative error against the upper frequency bound ω (Los Angeles hospital model, $r = 12$)

5.2 Flexible Aircraft Model Case

The second application is done in a similar way on a flexible aircraft model which comes from the industry. It has 289 states, 4 outputs and 3 inputs.

The flexible behaviour of an aircraft leads to a model with poorly damped modes, i.e. eigenvalues close to the imaginary axis and its rigid behaviour leads to eigenvalues which are real or almost real and very close to 0. All this make the model very ill-conditioned and thus hard to reduce with classical approaches [13].

The full-order model is reduced to order $r = 12$ by the three reduction methods. The upper bound R of the frequency interval goes from 1rad/s to 40rad/s and the $\mathscr{H}_{2,\omega}$-norm of the relative error is plotted with respect to ω on Figure 5.

Here the FL-BT leads to poor reduced-order models. The fact that the model is ill-conditioned increases the numerical issues arising in the computation of the frequency-limited gramians and in the balancing of the system. One example of those numerical issues is illustrated by the 2-norm of the Lyapunov equation

$$r_\omega = \|A^T \mathscr{Q}_\omega + \mathscr{Q}_\omega A + W_o(\omega)\|_2$$

which should be almost equal to zero. Yet, for $\omega = 14$rad/s, $r_\omega > 10^3$.

This error on the frequency-limited gramians directly impacts the FL-BT whereas it has little consequences on the FL-ISTIA. Indeed, until 18rad/s, FL-ISTIA leads to a better reduced-order model than the ISTIA and for larger ω they become

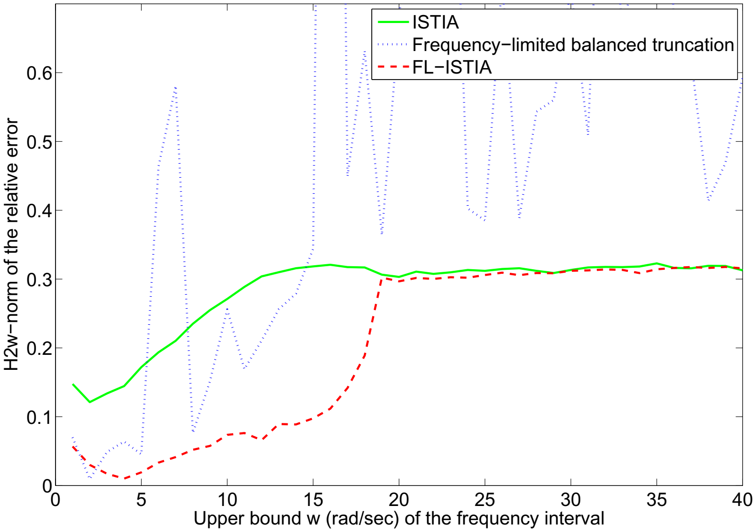


Fig. 5 $\mathscr{H}_{2,\omega}$-norm of the relative error with respect to the upper frequency bound ω (aircraft model, $r = 12$)

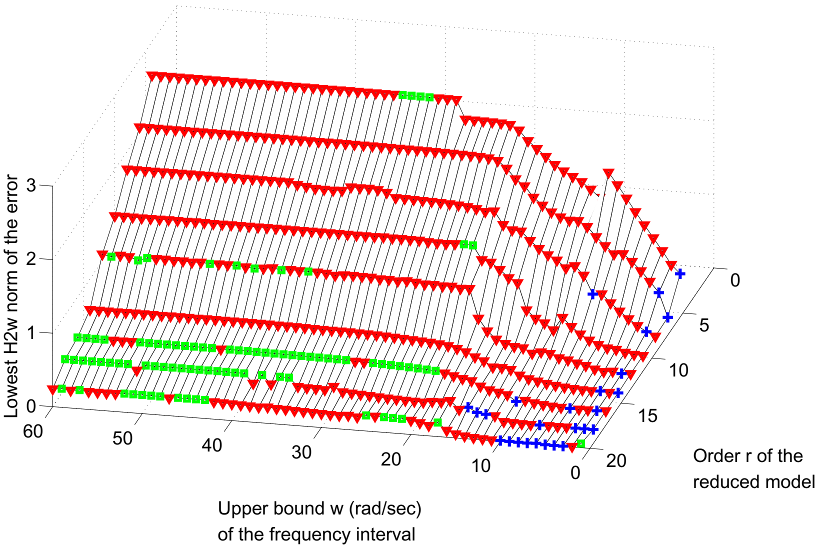


Fig. 6 $\mathscr{H}_{2,\omega}$-norm of the error with respect to the upper frequency bound ω and the order r of the reduced model (aircraft model)

equivalent. This equivalence comes from the fact that most of the spectral information is gathered in $0-20$rad/s.

When using the FL-BT and the FL-ISTIA, two parameters can be adjusted for the approximation: the upper bound ω of the frequency interval and the order r of the reduced model. Figure 6 represents the best approximation in terms of $\mathscr{H}_{2,\omega}$-norm among those provided by the ISTIA (green squares), the FL-BT (blue crosses) and the FL-ISTIA (red triangles) for several frequencies going from 1rad/s to 60rad/s and approximation orders going from $r=4$ to $r=20$. It plots the lowest $\mathscr{H}_{2,\omega}$ error, that is to say that when a method is better than the other, then its $\mathscr{H}_{2,\omega}$ error is plotted.

For this model it is clear that the FL-ISTIA mostly leads to a better approximation than the FL-BT independently of the frequency and order. Indeed, red triangles are predominant excepted for small frequencies ω and large order r where the FL-BT is the best method. This can be explained by the fact that a large number of interpolations cannot be achieved over a tight frequency range if there are not enough different behaviours to catch.

6 Conclusion

In this paper, a new application of the frequency-limited gramians proposed in [5] has been presented. Indeed they have been used in the *Iterative SVD-Tangential Interpolation Algorithm* (ISTIA, [13]) instead of infinite gramians which leads to an extended frequency-limited version of this algorithm called FL-ISTIA.

The relevance of the FL-ISTIA has been illustrated through two standards benchmark models and one flexible aircraft model. These tests have revealed that the method is as efficient as the frequency-limited balanced truncation but also more robust to numerical issues which makes it more tractable for ill-conditioned models.

Besides, the proposed algorithm will be soonly made available in the *MORE Toolbox* [14].

References

1. Anderson, M.R., Emami-Naeni, A., Vincent, J.H.: Measures of merit for multivariable flight control. Technical report, Systems Control Technology Inc., Palo Alto, California, USA (1991)
2. Anic, B., Beattie, C.A., Gugercin, S., Antoulas, A.C.: Interpolatory Weighted-$\mathscr{H}_2$ Model Reduction. Submitted to Automatica (2012)
3. Beattie, C., Gugercin, S.: Weighted model reduction via interpolation, Milano, Italy, pp. 12757–12760 (August 2011)
4. Gallivan, K.A., Vandendorpe, A., Van Dooren, P.: Model reduction of MIMO systems via tangential interpolation. SIAM J. Matrix Anal. Appl. 26(2), 328–349 (2005)
5. Gawronski, W., Juang, J.: Model reduction in limited time and frequency intervals. International Journal of Systems Science 21(2), 349–376 (1990)
6. Grimme, E.J.: Krylov Projection Methods For Model Reduction. PhD thesis, University of Illinois at Ubarna-Champaign (1997)
7. Gugercin, S.: An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. Linear Algebra and its Applications 428(8-9), 1964–1986 (2008)
8. Gugercin, S., Antoulas, A.C.: A survey of model reduction by balanced truncation and some new results. International Journal of Control 77(8), 748–766 (2004)
9. Gugercin, S., Antoulas, A.C., Beattie, C.: $\mathscr{H}_2$ model reduction for Large-Scale linear dynamical systems. SIAM Journal on Matrix Analysis and Applications 30(2), 609 (2008)
10. Leibfritz, F., Lipinski, W.: Description of the benchmark examples in *COMPl$_e$ib* 1.0. Technical report, University of Trier (2003)
11. Masi, A., Wallin, R., Garulli, A., Hansson, A.: Robust finite-frequency $\mathscr{H}_2$ analysis, pp. 6876–6881. IEEE (December 2010)
12. Ostrowski, A., Schneider, H.: Some theorems on the inertia of general matrices. Journal of Mathematical Analysis and Applications, 72–84 (1960)
13. Poussot-Vassal, C.: An Iterative SVD-Tangential Interpolation Method for Medium-Scale MIMO Systems Approximation with Application on Flexible Aircraft. In: The Proceedings 50th IEEE Conference on Decision and Control - European Control Conference (CDC-ECC 2011), Florida, pp. 7117–7122 (December 2011)
14. Poussot-Vassal, C., Vuillemin, P.: MOdel REduction Toobox. In: MSC (2012), `http://www.onera.fr/staff-en/charles-poussot-vassal/`
15. Van Dooren, P., Gallivan, K.A., Absil, P.A.: $\mathscr{H}_2$-optimal model reduction of MIMO systems. Applied Mathematics Letters 21(12), 1267–1273 (2008)
16. Vuillemin, P., Poussot-Vassal, C., Alazard, D.: A new expression for the frequency-limited $\mathscr{H}_2$-norm. Submitted to Automatica (2012)
17. Wilson, D.A.: Model reduction for multivariable systems. International Journal of Control 20(1), 57–64 (1974)

Hardware-in-the-Loop Flight Simulator – An Essential Part in the Development Process for the Automatic Flight Control System of a Utility Aircraft

André Kaden, Bernd Boche, and Robert Luckner

Abstract. An automatic flight control system (AFCS) is designed for the utility aircraft STEMME S15, a high-performance motor glider. The AFCS shall automatically control the aircraft with high precision during surveillance, reconnaissance and measurement flights. To test the AFCS a ground test facility in form of a hardware-in-the-loop (HIL) simulator was built. The correct integration of the AFCS and its flight control functions into the aircraft are verified by this test system. HIL simulation is part of a cost-effective development process for safety-critical systems that will be established as part of this project. This paper gives an overview of the development process and describes the concept, the functional principle and the design of the HIL simulator. A comparison of flight test and simulation results of the first automatic landing of the S15 is shown, as example for the HIL simulator validation.

1 Introduction

The automatic flight control system (AFCS) is an integral part of the utility aircraft STEMME S15. The system is complex, highly-integrated and it has safety critical functions. It consists of software and hardware. The development process of such systems is subject to strict rules that are described in SAE ARP 4754 [1]. The potential of making mistakes during the development of complex systems is high. In order to assure the correctness of such a highly-integrated system, a structured system development process has to be established. That includes rigorous, systematic, and repeatable testing. The objective of *Aerospace Systems Engineering* is to develop and optimize the appropriate design methods in order to

André Kaden · Bernd Boche · Robert Luckner
Berlin Institute of Technology, 10587 Berlin, Germany
e-mail: {andre.kaden,bernd.boche}@ilr.tu-berlin.de,
robert.luckner@tu-berlin.de

realize an airworthy product that meets all requirements and can be certified by the authorities. In order to facilitate the approval process, policies and standards for system development in aviation exist. Guidelines for the development of software are described in RTCA DO-178B [2].

An essential element of the development process is the functional test. It is carried out on various test rigs on component, system and aircraft level. It is important to use efficient development methods to achieve the goal of the overall system validation and certification that is affordable. This is crucial especially for small and midsize enterprises. One possibility is the choice of appropriate test benches. It is important to ensure that tests can be performed cost and time efficient without reducing their quality. This report describes a ground test facility, called hardware-in-the-loop (HIL) flight simulator that replaces an *Iron Bird* by using the prototype aircraft in conjunction with a flight simulator.

2 Research Project LAPAZ

The acronym LAPAZ stands for Air Utility Platform for General Civil Aviation, *Luft-Arbeits-Plattform für die Allgemeine Zivilluftfahrt* in German. The LAPAZ project is funded by the Federal Ministry of Economics and Technology (BMWi) in the National Aerospace Research Program LUFO IV. Project partners are the STEMME AG as the coordinator, University of Stuttgart's Institute of Aircraft Systems (ILS), and TU Berlin's Department of Flight Mechanics, Flight Control and Aeroelasticity (FMRA). The objective of the research project is to develop and demonstrate a reliable and fault-tolerant automatic flight control system for a utility aircraft. ILS is responsible for the development of the fault-tolerant platform for the flight control system, including all redundancy mechanisms. FMRA develops the flight control laws (FCL), the flight mechanical simulation model, the human-machine interface as well as the FCL development process. For the planned EASA CS23 certification as utility aircraft, a specially designed development process for complex, safety-critical systems will be established. The software development is based on it. The following Section gives an overview of the development process with focus on the part that is related to develop the flight control functions of the AFCS.

2.1 Development Process

The overall LAPAZ development process follows the V-Model that is shown in Fig. 1. Initially all functional and non-functional requirements for the automatic flight control system are defined in the top-level specification. They are gradually refined top-down from aircraft level via system and assembly level to the hardware and software requirements on component level. After encoding and production, the gradual integration and verification (bottom-up) follows. Each process step ends with verification tests. The final step includes validation tests. The ultimate validation is the acceptance by the market and the customer.

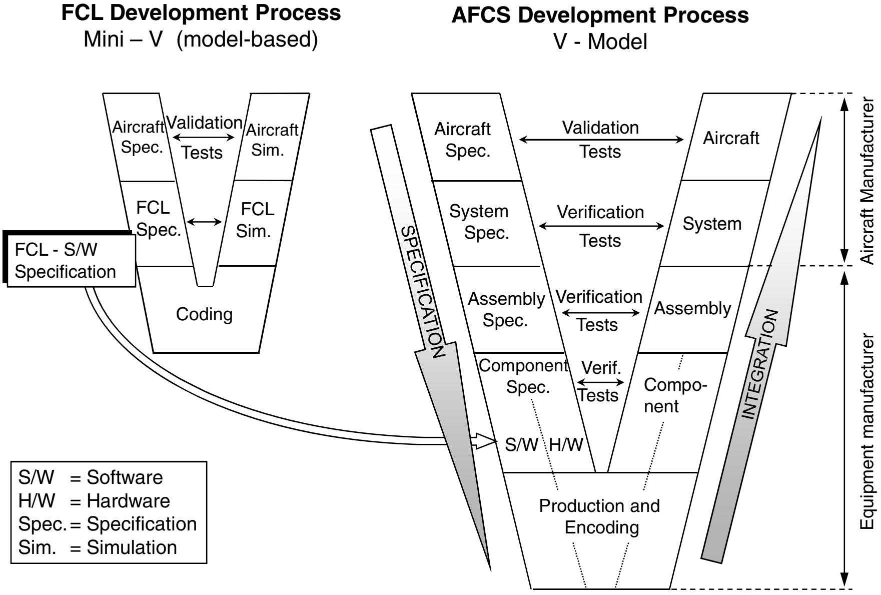


Fig. 1 System development process according to the V-model [3]

It should be noted that the V-model is an idealization. In reality, several iterations occur that are not shown here. For example, if an error in the system specification is detected during validation tests on aircraft level, a new iteration of the V-process will be initialized starting with updating the system specification.

As LAPAZ is a research project, the AFCS development proceeds incrementally. That means not all functions are created simultaneously. Instead, starting from a basic functionality, new features are added gradually. Each of them is developed in accordance to the V-model including the validation by flight tests. At the beginning, the sequence, in which the partial functions shall be incrementally developed, has to be defined. This is achieved by continuously considering the interdependencies between the functions, the implementation risk, and the importance for the final product. The advantage of this procedure in comparison to the development of the entire functionality in one shot is the early attainment of feedback from flight tests to the development process. Thereby gradual improvements are possible, so that the existing functionalities are subject to a maturation process. The incremental approach leads to continuously growing software architecture. In order to avoid sub-optimal structures, the basic system architecture has to be fully defined in the very beginning. Any restructuring would be extremely costly and has to be avoided.

The development of flight control laws (FCL) is based on a flight mechanical simulation model of the aircraft. The software is specified by means of Simulink® and Stateflow® that are toolboxes of the software package MATLAB®. Source code of the FCL is generated through the use of the Real-Time Workshop®

Embedded Coder™ (RTW-EC). After compilation, an executable application is created that can be integrated on the target system. The integration work is done gradually from subsystem to the overall system.

The entire integration process is accompanied by verification and validation testing to demonstrate the functionality according to the specifications (verification), as well as the correctness of the requirements (validation). First model-based tests are carried out during FCL design using linear and non linear flight simulations, see Fig. 2.

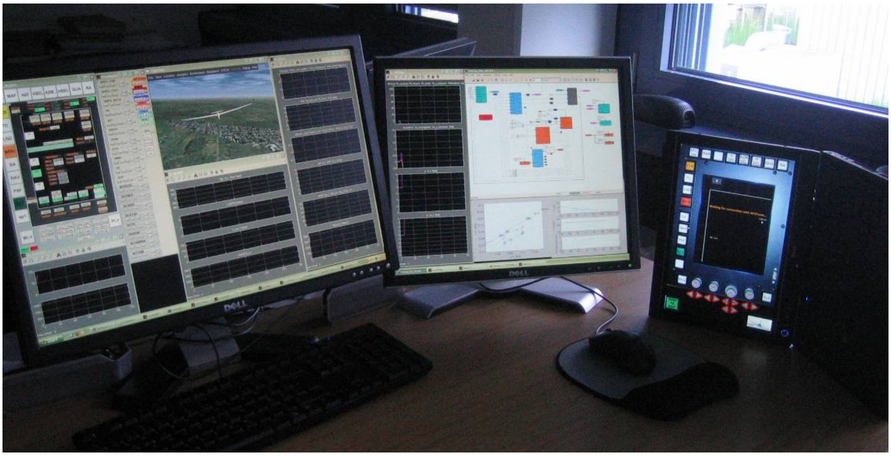


Fig. 2 Work station and Automatic Flight Control Panel (AFCP) for flight control law development

When the design is frozen, extensive automatic offline tests are performed to check all functional requirements (software-in-the-loop (SIL) simulations), followed by real-time simulator tests with pilot and SIL. The real-time tests are performed on TU Berlin's research simulator SEPHIR (*Simulator for Educational Projects and Highly Innovative Research*). After C code generation using RTW-EC and compilation, it is verified that the code will correctly run on the target system. This is achieved by using the so-called host simulator that is an emulation of the target system. The host simulator has been provided by the project partner ILS who implements the FCL code on the target system. Test vectors are specified and used to compare the correct implementation of the software. They comprise the input vector that is generated and used as input for the FCLs, and the corresponding output vector that is computed by the FCL Simulink code in the development environment. If the output test vector can be identically reproduced on the host simulator, the FCL software is delivered to ILS for implementation on the target system. ILS proves the correct implementation of the FCL software on the target system, by verifying that the embedded FCL software behaves as in the development environment. The testing of the software on the target system is a fundamental requirement of the Directive RTCA DO-178B [2] that requires a validated and representative test environment. In Section 6.4.1 *Test Environment*, this is described as follows: *„An excellent test environment includes the software loaded into the target computer and tested in a high fidelity simulation of the target computer environment."* [2]

The HIL simulation is the next important step in the development process. It is performed at STEMME and is the first step to verify the correct integration of the AFCS into the aircraft. This verification step is particularly important because not all aircraft requirements can be tested model-based and on system level. With the HIL simulator, the following aspects can be tested:

1. *Hardware/Software Integration Testing* to ensure that the application software (FCL) in the target computer will satisfy the high-level requirements (Subsection 6.4.3 a [2]) in the fully integrated environment,
2. *Software Integration Testing*, to ensure that the application software (FCL) interacts correctly with the AFCS middleware and satisfies the software requirements and software architecture (Subsection 6.4.3 b, [2]) in the fully integrated environment,
3. Interaction of the pilot with the AFCS via the Automatic Flight Control Panel AFCP (human-machine-interface).

2.2 STEMME S15 Prototype

The utility aircraft S15 is a variant of the motor glider S6. It is designed for commercial applications and shall be certified according to EASA CS-23 [4]. In the LAPAZ project an AFCS was integrated into the S15 prototype [11]. The AFCS signal flow is schematically shown in Fig. 3. A detailed description can be found in [5], [6], [7], and [11]. Details for controller development are given in [8].

3 Hardware-in-the-Loop Simulator

This Section describes the concept, the design and the functionality of the HIL simulator. According to the definition of the LAPAZ development process, the HIL simulator shall be used in different applications. For those applications, requirements have been formulated, that are the basis for the design concept. One of the most important requirements is that the HIL simulator shall replace the *Iron Bird* for ground tests on aircraft level. This approach meets demands for time and cost efficiency, it also minimizes possible sources of error as it uses original hardware instead of simulations. The concept is to replace the *Iron Bird* by the prototype of the STEMME S15 as part of the HIL simulator.

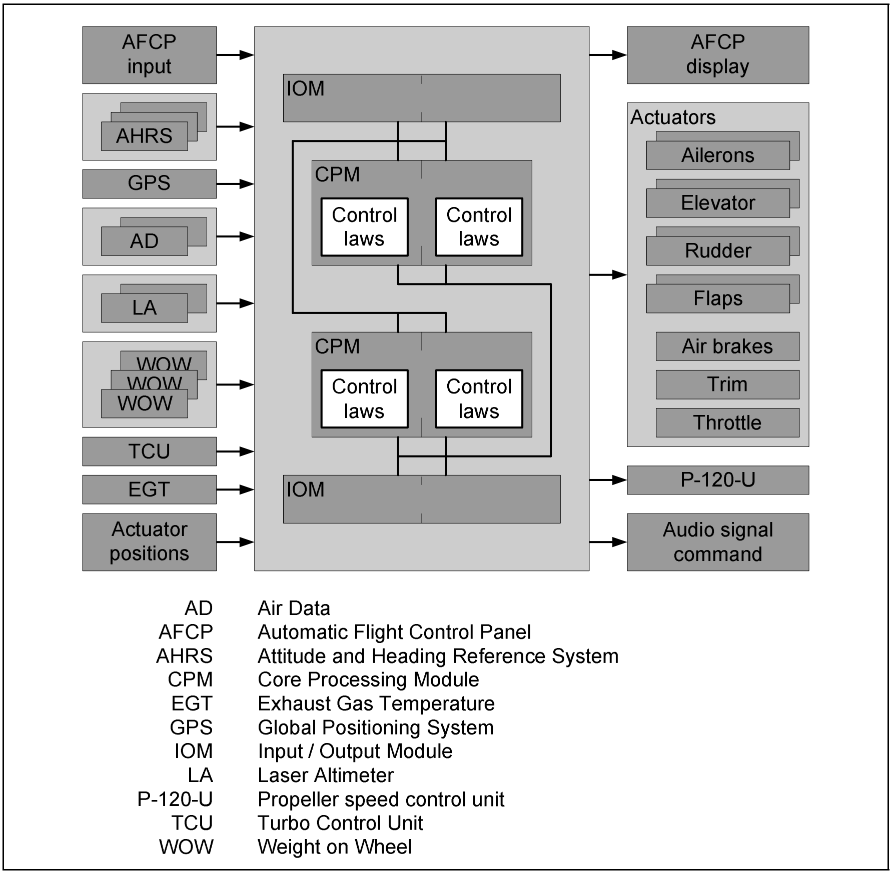


Fig. 3 Overview of automatic flight control system AFCS

3.1 Design Concept

The main task of the HIL simulator is the functional ground test of the AFCS hardware and software, in order to verify the correct function of the flight control laws in the integrated state. Additionally, there are further applications. The flight test program can be safely examined before flight. Its feasibility can be assessed by the test pilot. Abnormal situations can be simulated to test and practice procedures where the test pilot has to take over manual control and to prove that they are safe. The replay of recorded flight test data allows post-flight analyses. The reproducibility of experiments allows the comparison of different controller versions. Furthermore, the HIL simulator serves as a demonstrator for the automatic flight control system at air shows or in front of the customers. The following requirement list is derived from these applications:

- inclusion of as many as possible original aircraft and AFCS components,
- generation of the required sensor and engine data on ground by real-time simulation,
- measuring the control surface deflections to drive the S15 flight simulation,
- processing and transmitting simulated sensor data to the AFCS,
- mobility of the test system for transport,
- robustness of the test facility for use on rugged test sites,
- presentation of the visual scenery on a projection system and the cockpit instruments on LCD screens.

This requirement list is the basis for the design concept of the ground test system. Figure 4 schematically shows the HIL simulator. It consists of the aircraft (STEMME S15 prototype) and a simulation unit.

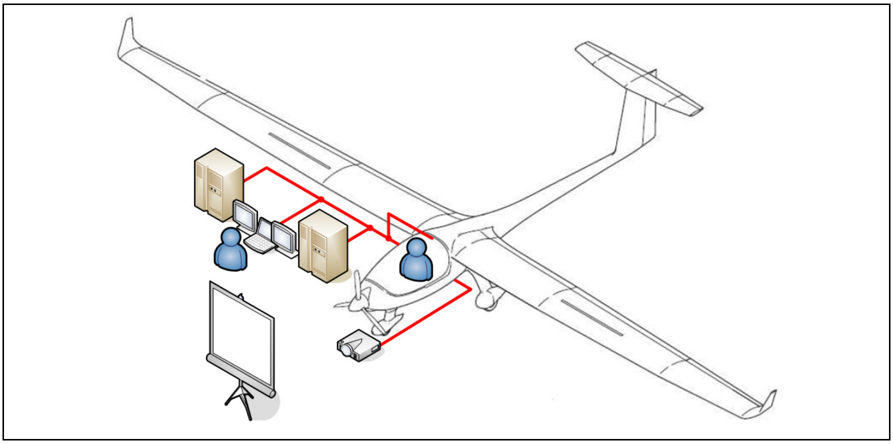


Fig. 4 Concept for establishing the HIL simulator

The simulation unit is mobile and ruggedized. All hardware is integrated into two wheeled transport cases that are connected to the S15 for testing. The workplace of the test pilot is in the cockpit. From there, he controls the autopilot with the Automatic Flight Control Panel (AFCP) that is installed in the front panel. He monitors the aircraft motion on simulated instruments. Even manual flight control by using the control elements of the S15 is possible. A visual system that is installed in front of the aircraft, projects the outside view on a screen.

3.2 Functional Principle

The HIL simulator simulates the STEMME S15 flight dynamics throughout the entire flight envelope, i.e. in air and ground operations and allows flying in manual mode and with the AFCS. In the following the functional principle of the simulator is described, see. Fig. 5.

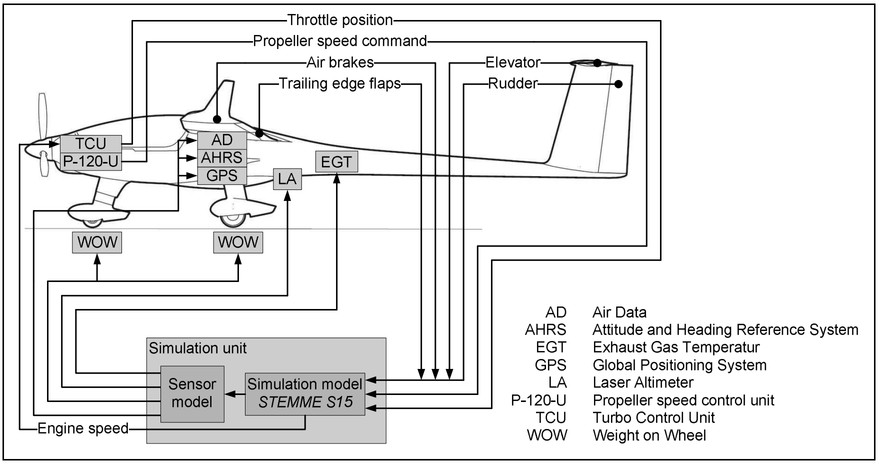


Fig. 5 Functional principle of the HIL simulator

Due to missing aircraft motion all AFCS sensors are inoperative. This applies to the air data systems (AD), the inertial platforms (AHRS), the satellite navigation system (GPS), the laser altimeter (LA), and the ground contact sensors (WOW). As the aerodynamic loads on the linkage and the actuators are missing, the deformation of the mechanical linkage between actuator and control surface is not exactly the same as in flight. For safety, environmental and economic reasons, the engine is not running during HIL simulator tests. Therefore sensor values for the propeller speed and exhaust gas temperature (EGT) are missing as well.

A high-fidelity flight simulation generates the missing sensor signals that are required as input variables for a correct operation of the AFCS. The simulation runs on the simulation unit, see Fig. 5. It includes a non-linear flight mechanical model of the aircraft STEMME S15. The model consists of a six degree of freedom simulation of the rigid aircraft and the first sixteen structural modes. The modelling of structural vibrations is important for testing of highly dynamic AFCS functions such as gust load alleviation. In addition, it includes sensor models, that consist of a dynamic model (dead time, etc.) and an error model (noise, bias, drift, etc.). The simulated sensor and engine data are sent to the flight control computers as input signals.

Input signals for the flight simulation are the deflections of the control elements. They are measured by potentiometers. The use of the deflections instead of the AFCS actuator command eliminates the necessity to model and simulate the overall transfer function from actuator to control surface (linkage kinematics, elasticity, friction, etc.). Currently, the effect of aerodynamic loads on the linkage and actuators is not simulated, as it would require additional hardware to apply forces and moments to the control surfaces. The effect of aerodynamic loads on the control linkage can be analytically considered in the simulation model. The engine model uses the throttle position (from the Turbo Control Unit, TCU) and the propeller speed (from the propeller speed control unit, P-120-U) as input command.

3.3 Configuration

The HIL simulator has a similar architecture as FMRA's research flight simulator SEPHIR. It uses the same hardware components that are used in other simulators of the department and that have been proven in practice. Thus, existing software modules can be adapted to the HIL simulator by minimal programming effort. The architecture of the simulator is modular and the system is expandable. Mostly standard components are used. Procurement and maintenance costs are low. Figure 6 shows the main hardware components of the simulation unit and their interaction.

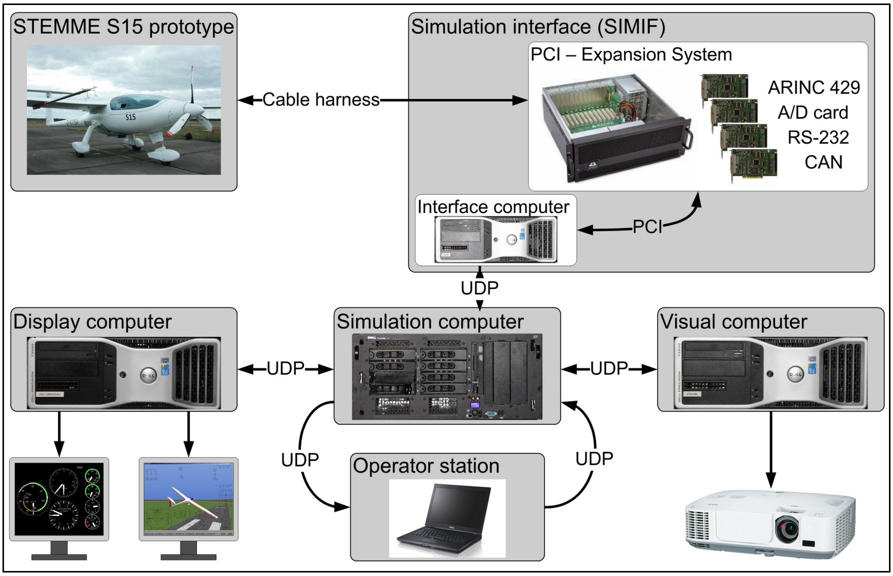


Fig. 6 Overview of the hardware components

The actual simulation process runs on a high performance real-time computer with a UNIX operating system that is optimized for time-critical applications. The simulation runs at 125 Hz.

The simulation interface (SIMIF) enables data transfer between the S15 and the simulation computer. It consists of a conventional personal computer (PC) and a PCI[1] expansion system for increasing the number of PCI cards. The interface cards include the transmission standards ARINC 429, RS-232 and CAN and allow analog/digital (A/D) conversion. The outputs of the cards are connected via a cable harness to the input/output modules of the computers of the AFCS. The process sequence of the data transfer is performed in the following order:

[1] PCI (Peripheral Component Interconnect) is a computer bus for integrating hardware devices into a computer.

- receive data from the S15 prototype (e.g. control surface deflections),
- send data via User Datagram Protocol (UDP) to the simulation,
- receive data via UDP from the simulation, and
- send data to the S15 prototype.

These four steps form a cycle process with a frequency of 125 Hz. Latency resulting from this simulation process is discussed in Section 4.1. All communication between the client computers and the simulation unit is realized via UDP. The simulation is controlled by a notebook. Various simulation parameters can be modified in real time, for example atmospheric conditions or sensor offsets. Simulated aircraft position and attitude are processed by a visual system to present the outside view to the pilot. The image generator software PHILOSIM is a product of the company Philotech. Aircraft instruments and a view of an outside observer on the aircraft are generated by an additional PC and are indicated on LCD monitors. Figure 7 shows the HIL simulator connected to the STEMME S15 during a functional test of the AFCS.

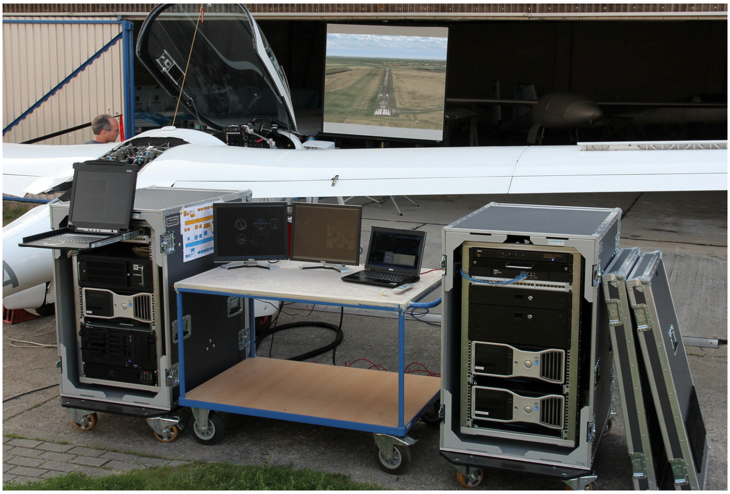


Fig. 7 STEMME S15 connected to the HIL simulator

3.4 Simulation Environment

To develop the FCL and to perform the subsequent tests (PC, SIL and HIL simulation) a highly accurate flight mechanical simulation model of the S15 has been developed. It is implemented in MATLAB and Simulink, see [9]. Using the model in the HIL simulator, the sub-models for actuators, linkages and AFCS mechanisms (consolidation, degradation) are not required as they are replaced by original hardware. The following effects are simulated:

- six degrees of freedom rigid body motion coupled with aeroelastic degrees of freedom,
- two-point aerodynamic (wing and tailplane) with stall and ground effect, that has been identified and modelled using flight test data, see [9],
- propulsion system consisting of engine and propeller,
- wing-mounted elastic boom for the angle of attack vane,
- landing gear,
- terrain,
- position in WGS84 coordinates,
- standard atmosphere and atmospheric turbulence,
- sensor dynamics.

Fig. 8 shows the integration of the S15 flight simulation model into the HIL simulator. It is embedded into a Simulink framework model and a real-time simulation environment that is generated with RTW-EC. The Simulink framework model includes components for data transfer between the simulation and the client computers of the simulation unit. Corresponding UDP ports are responsible for the transmission of data packages. For this reason, additional Simulink S-function blocks are created in the C programming language. The received data is processed and assigned to the inputs of the simulation model. Necessary initial values of the simulation are read from an input text file. Also, the binary model output data are converted for each sensor according to the respective communication protocol (for example ARINC 429) before sending. A block for data recording to an external file is also part of the framework model. The user can start and stop the data recording as required. The data are stored in binary format.

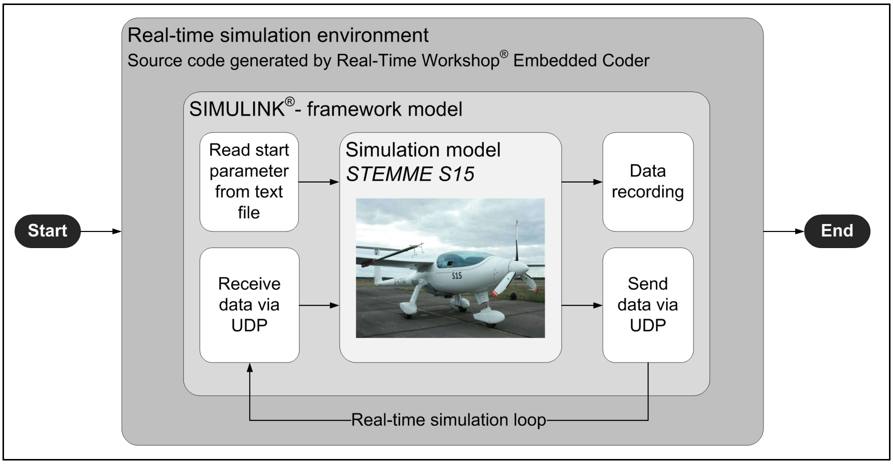

Real-time simulation environment
Source code generated by Real-Time Workshop® Embedded Coder
SIMULINK®- framework model
Read start parameter from text file
Simulation model
STEMME S15
Data recording
Start
End
Receive data via UDP
Send data via UDP
Real-time simulation loop

Fig. 8 Architecture of the simulation environment

Both models (framework and S15 flight simulation) are programmed in form of Simulink block diagrams, which has to be converted into C source code by means of RTW-EC in order to accelerate computation. The generated source code has to be supplemented with real-time functionalities that attach the entire simulation process to the real-time clock of the simulation computer using the POSIX application interface [10].

4 Validation

The validation of the HIL simulator has to prove that the simulator correctly represents the behaviour of the S15 prototype throughout the whole flight envelope. Several analyses were done for validation:

1. The flight simulation model was validated during the identification of the parameters for the aerodynamic and the thrust model, see [9]. The flight test data were partly used for model identification and partly for verification of the flight mechanical behaviour.
2. During the integration phase of the HIL simulator the correct implementation of the transmission protocols for the simulated sensor signals was verified.
3. A plausibility check of the simulated sensor values was done using the recorded data.

The next two Subsections deal with the analysis of the HIL simulation latency and the comparison of flight tests and simulation results.

4.1 Latency of the Simulation Unit

The HIL simulator consists of two main components, the S15 aircraft and the simulation unit. The HIL simulation consists of three principal asynchronous processes (P1, P2, and P3):

P1: automatic flight control system and S15 components,
P2: data transfer and
P3: flight simulation.

Process P1 represents the AFCS and all S15 components that belong to it. It comprises sensor signal acquisition, computation of redundancy management and flight control functions, actuator control loops as well as mechanical linkage to control surfaces and measurement of their positions. Process P1 is carried out on the S15 prototype. Process 2 and 3 run on the simulation unit. Part of processes P2 and P3 are the following data handling tasks:

- data handling by interface cards (e.g. A/D conversion),
- data transfer between PCI expansion system and SIMIF computer via PCI,
- data transfer between SIMIF and the simulation computer.

These tasks are handled within microseconds and their latency can be neglected here. Figure 9 shows the time differences due to sensor data generation[2]. The diagram shows all three main processes, without the clock rate of process 1. The time between reception of an input signal and the generation of corresponding output signal is defined as latency. Process P2 runs on the simulation interface computer (SIMIF) and P3 runs on the simulation computer. Both use the same clock rate. As the processes run asynchronously, latencies can vary. For the favourable case a) the latency is at least 8 ms. The cycle shift in the unfavourable case b) leads to a latency of at most 17 ms. As the maximum sensor update rate is 50 Hz (20 ms), the latencies are tolerable and do not significantly affect the behaviour of the controlled system.

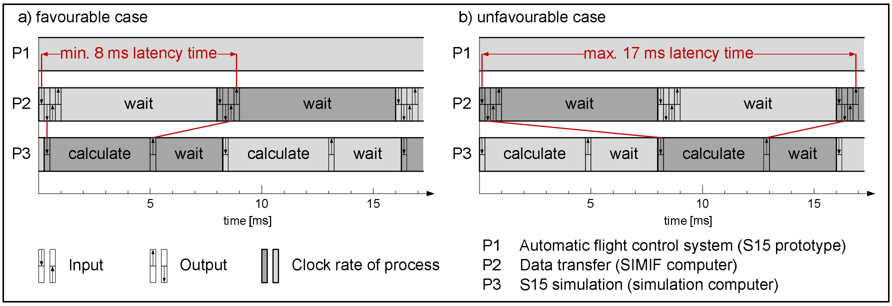


Fig. 9 Latencies of simulation a) favourable case; b) unfavourable case

4.2 Validation of HIL Simulation with Flight Test

In this Section the results of the HIL simulation are compared to recorded data from flight tests. The main focus lies on the flight dynamical behaviour of the aircraft that is controlled by the automatic flight control system. For the validation, a manoeuvre has to be defined; the flight conditions (initial state, atmosphere, mass, centre of gravity, etc.) have to be identical (or at least similar). The results are stored with the same data recording system in flight and in HIL tests.

The S15 simulation model has been validated with flight test data from specific identification flight. As a validation example for the closed loop system, the first automatic landing of the S15 prototype on March 22nd 2012 in Neuhardenberg is compared to results from the HIL simulation. During the landing process the aircraft follows a defined three dimensional trajectory with a predefined speed. The results are stored in both cases within the AFCS by the so-called SPY functionality. Amongst others, all sensor values are recorded there. The requirement for identical flight conditions is only partly met for the regarded case. During flight test, the turbulence level was low and the wind velocity was small; in the HIL simulation both were zero.

[2] The simulated time lags as well as the sensor update rates are neglected.

The observations are restricted to the symmetrical plane. The crosswind component plays a minor role in longitudinal motion. Mass and centre of gravity are similar for simulation and flight test.

Important parameters of the aircraft longitudinal motion over the distance to the runway threshold[3] are shown in Fig. 10 and Fig. 11. In Fig. 10, sensor values for

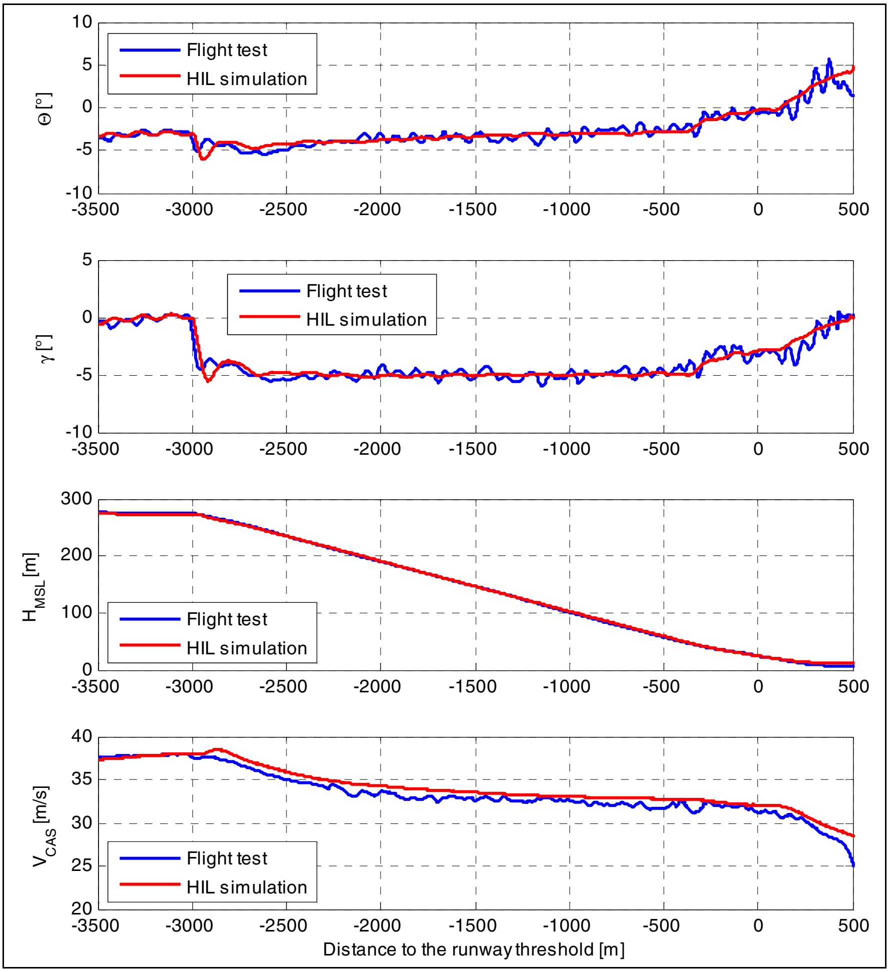


Fig. 10 Pitch angle Θ, flight path angle γ, height H_{MSL} und air speed V_{CAS} over the distance to the runway threshold

[3] The runway threshold is located in the origin of the abscissa.

pitch angle Θ, flight path angle γ, height above MSL (*Mean Sea Level*) H_{MSL} and calibrated airspeed (CAS*)* V_{CAS} are displayed. Figure 11 compares the height above MSL with the height above ground (GND - *Ground*) H_{GND}. Not surprisingly, the sensor values of the HIL simulation are relatively smooth, whereas the flight test data are noisy. This is attributed to atmospheric disturbances. During flight test and simulation, the gust load alleviation function was not active.

During the final approach, the graphs of simulation and flight test match well for all depicted sensor values. Particularly this applies to the vertical flight profile shown in the third diagram of Fig. 10. Comparing flight test and HIL simulation it can be stated that in both cases the system shows basically the same control behaviour.

The third graph of Fig. 10 shows that after touchdown of the S15 on the runway (about 400 m after the threshold) the simulated height above MSL differs from the actually measured height. Figure 11 illustrates this in detail. The first diagram shows that the deviation is about 3 m. The height above MSL is measured by GPS. According to the AIP (*Aeronautical Information Publication*), the published landing elevation of the runway 08 in Neuhardenberg is 10 m above MSL. Correspondingly, the runway for the simulation is placed at this height. In reality, the GPS sensor of the AFCS measures a runway height of about 6-7 m MSL. This difference is attributed to the measurement accuracy. The height deviation is within the specified accuracy of the GPS that is augmented by EGNOS (*European Geostationary Navigation Overlay Service*).

It is noteworthy that the first automatic landing was accomplished successfully. This is due to the robust design of the controller. Until the S15 reaches the runway threshold, the altitude above MSL is used as control variable. The vertical flight profile specifies that the aircraft crosses the threshold at 15 m above ground. The controller uses the runway elevation that is entered by the pilot (for HIL simulation and flight test the runway elevation was set to 10 m MSL). Due to the error

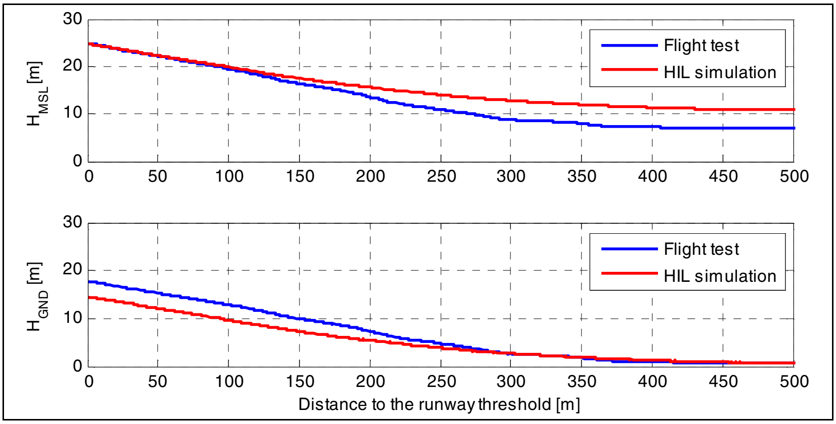


Fig. 11 Height above MSL and GND over the distance to the runway threshold

in the measured MSL during the flight test, the S15 crosses the threshold 3 m too high, about 18 m above ground (second diagram Fig. 11). This value is within the specified tolerance, so that the landing could be continued. Having passed the threshold the aircraft is over the runway and the height above ground measured by the laser altimeters is used for control. This assures the adaptation of the flare to the actual runway. From 300 m after the threshold to touch down the trajectory relative to ground from simulation and flight test match again.

5 Conclusion and Outlook

As part of the research project LAPAZ, a hardware-in-the-loop (HIL) simulator was build. It has been validated and it is successfully used for functional tests of the STEMME S15 automatic flight control system. The HIL simulator is an integral part of the development process of this safety-critical system and contributes significantly to the success of the research project. Previous to the integration tests on the HIL simulator, the computer platform, its redundancy management and the correct integration of the FCL software are rigorously tested on a test system that is specifically designed for this purpose by the project partner ILS. During the HIL simulator tests the test pilot can sit in the S15 cockpit in his familiar working environment. The tests that are performed before the actual flight tests improve the efficiency of the flight tests and reduce the risk for test pilot and aircraft. Standard and abnormal AFCS behaviour are realistically represented and the corresponding reaction of the pilot can be analysed or trained. So, this concept has proven to be highly valuable. In comparison with an *Iron Bird* both time and money can be saved, as well as floor space for storage. The HIL simulator can be linked to the aircraft in less than 15 minutes. The modular structure of the simulator facilitates the extension of the system if needed. The comparison of simulation and flight test shows that the HIL simulator realistically reproduces the behaviour of the STEMME S15. Therefore, findings from the HIL simulations can be used in the development of the AFCS.

References

[1] SAE: Certification Considerations for Highly-Integrated or Complex Aircraft Systems, ARP 4754, Systems Integration Requirements Task Group, AS-1C, ASD, SAE, Warrendale (1996)

[2] RTCA: Software Considerations in Airborne Systems and Equipment Certification, DO-178B, RTCA, Inc., Washington, D.C. (1992)

[3] Brockhaus, R., Alles, W., Luckner, R.: Flugregelung, 3. Auflage. Springer, Heidelberg (2011) ISBN 978-3-642-01442-0

[4] EASA: Certification Specification for Normal, Utility, Aerobatic and Commuter Category Aeroplanes, CS-23, EASA (2003)

[5] Hesse, S., Reichel, R., Görke, S., Dalldorff, L.: Eine skalierbare Plattform für sicherheitskritische, automatische Flugsteuerungssysteme der allgemeinen Zivilluftfahrt. In: Deutscher Luft- und Raumfahrt Kongress (2009)

[6] Hesse, S., Görke, S.: An Affordable, Fault-Tolerant Automatic Flight Control System for the Utility Aircraft Stemme S15. In: CEAS EuroGNC 2011, Conference in Guidance, Navigation and Control in Aerospace, München (2011)

[7] Polenz, S., Cake, F., Görke, S., Reichel, R.: SAFAR eine Fly-by-Wire Steuerung für ein Flugzeug der General Aviation (DIAMOND DA42). In: Deutscher Luft- und Raumfahrt Kongress (2011)

[8] Lamp, M., Luckner, R.: Flight Control Law Development for the Automatic Flight Control System LAPAZ. In: CEAS EuroGNC 2011, Conference in Guidance, Navigation and Control in Aerospace, München (2011)

[9] Meyer-Brügel, W., Luckner, R.: Flight Mechanical Simulation Models for Design and Test of Automatic Flight Control Functions. In: CEAS EuroGNC 2011, Conference in Guidance, Navigation and Control in Aerospace, München (2011)

[10] Burns, A., Wellings, A.: Real-Time Systems and Programming Languages: Ada 95, Real-Time Java and Real-Time C / Posix, 3rd edn. Addison Wesley Langmain (March 2001) ISBN: 0-201-72988-1

[11] Dalldorff, L., Luckner, R., Reichel, R.: A Full-Authority Automatic Flight Control System for the civil airborne utility platform S15-LAPAZ. In: CEAS EuroGNC 2013, Conference in Guidance, Navigation and Control in Aerospace, Delft (2013)

Autonomous Wind Tunnel Free-Flight of a Flapping Wing MAV

Christophe De Wagter, Andries Koopmans, Guido de Croon,
Bart Remes, and Rick Ruijsink

Abstract. A low-cost high performance control system is developed to enable autonomous untethered flight inside a wind tunnel. Such autonomous flight is desirable for aerodynamic experiments on flapping wing MAVs, since fixing the fuselage has been shown to significantly alter wing deformations, air flow and performance on vehicles with a periodically moving fuselage. To obtain autonomous untethered flight, 3D position information is obtained from off-board WiiMote infrared tracking sensors with a total system accuracy of 0.8mm and an update rate of 80Hz in a quarter cubical meter control box. This information is sent to a 1.5 gram onboard autopilot containing communication, inertial measurements as well as onboard infrared tracking of an in-tunnel LED to achieve the high performance control needed to position itself precisely in the wind tunnel flow. Flight tests were performed with the 16 gram flapping wing MAV DelFly II. The achieved control performance is shown to be sufficient for many new research purposes, like researching the influence of a fixed fuselage in flapping wing aerodynamic measurements and obtaining more precise performance characteristics.

1 Introduction

The aerodynamics of flapping wings is a broadly studied subject. Besides theoretical, numerical, and animal studies [19, 2, 13, 20, 5, 3, 11, 21], there is an increasing number of studies on artificial wings of flapping wing Micro Air Vehicles (MAVs)

Christophe De Wagter · Guido de Croon · Bart Remes · Rick Ruijsink
Micro Aerial Vehicle Lab, Faculty of Aerospace Engineering,
Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands
e-mail: {c.dewagter,g.c.h.e.decroon,
B.D.W.Remes,r.ruijsink}@tudelft.nl

Andries Koopmans
Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands
e-mail: a.koopmans@student.tudelft.nl

(e.g., [4, 7, 12, 17]). Fluid-structure interaction makes the study of flapping wing aerodynamics computationally very complex, but also makes it hard to obtain measurements without any external influences.

Until now, studies on flapping wing MAVs are always performed with a fixed fuselage [6, 8, 9, 7, 12, 10, 17]. For example, research on the *DelFly II* flapping wing MAV (Fig. 1) has always been studied with a fixed fuselage be it for computational fluid dynamics, force measurements[6], or Particle Image Velocimetry (PIV)[9, 10]. While DelFly II was designed to minimize the fuselage rocking motion, some inevitable motion remains. This means that a fixed support changes the flight conditions of the otherwise periodically moving fuselage.

This paper proposes a low cost high performance system that enables the DelFly II to fly untethered at a fixed location in a constant wind flow. This allows for new highly needed insight [15, 18] to be obtained about the aerodynamics, free flight deformations, flight characteristics, performance aspects and transient behavior of flapping wing MAVs. It is a first step toward free-flight PIV measurements in a wind tunnel[16]. While this has been achieved with living creatures [5, 3, 11], it has not been done before with a flapping wing MAV.

In the remainder of the paper, first the system setup is explained (Section 2). Subsequently, in Section 3, the sensor fusion necessary to estimate the DelFly's state is discussed. The manner in which the state estimate is used for controlling the DelFly during free flight is explained in Section 4. Then, results are discussed of stationary (Section 5) and transient (Section 6) tests in the wind tunnel. Finally, conclusions are drawn in Section 7.

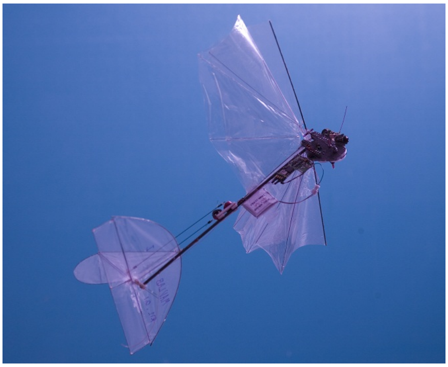


Fig. 1 DelFly II has a double pair of flapping wings, driven by a 2 Watt sensorless brushless motor that drives gears with pushrods to the wings. The horizontal stabilizer has an elevator and the vertical stabilizer a rudder. Both are driven by servo's. The flapping frequency is controlled by the power setting of the main motor controller. Onboard electronics comprise of a radio or modem for communication, an autopilot processor with sensors and a special made brushless motor controller.

2 System Overview

Precise high performance position control is an important requirement to allow close up measurements, while the characteristics of the vehicle should not be altered by adding too much weight. Therefore a combined onboard plus off-board solution is used.

A high update rate tracking system measures the position of the DelFly externally. Simultaneously, an onboard microcontroller performs inertial measurements. It also reads analysis results of the onboard camera looking at an infrared LED placed in the tunnel in front of the DelFly and which acts as a heading reference [1]. External position measurements are sent to the DelFly to allow full onboard fast control loops to follow the desired setpoints as illustrated in Fig. 2).

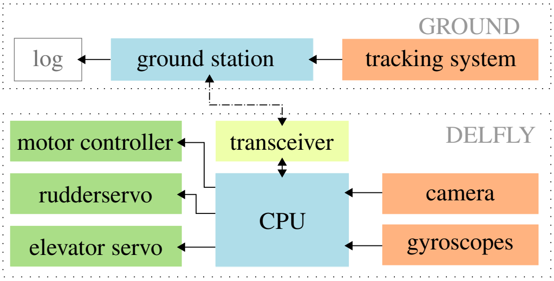


Fig. 2 Schematic overview of all the system components. It consists of a ground section with the heavier sensors and a section onboard the DelFly with the high rate low latency sensors to allow light high performance control.

Position data is gathered and processed on the ground and packed with commands. These are then sent to the DelFly autopilot over the wireless Bluetooth link. Final data merging and control is done onboard while telemetry is being returned to the ground for logging purposes (Fig. 3).

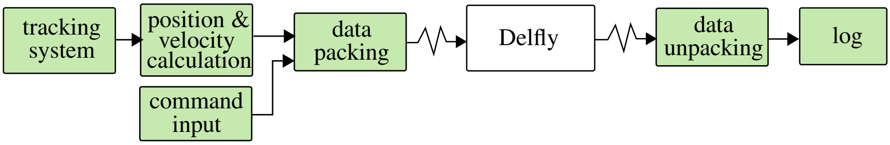


Fig. 3 Flow diagram of ground station functionality

2.1 Onboard: Autopilot System

A 20 Mega-Instructions Per Second (MIPS) capable 4 by 4 millimeter microcontroller[1] is heart of the autopilot system. A Bluetooth serial modem[2] is used as bidirectional digital communication link at 38400bps. A Wii-Mote camera sensor is

[1] AVR ATmega88PA.

[2] Panasonic PAN1321.

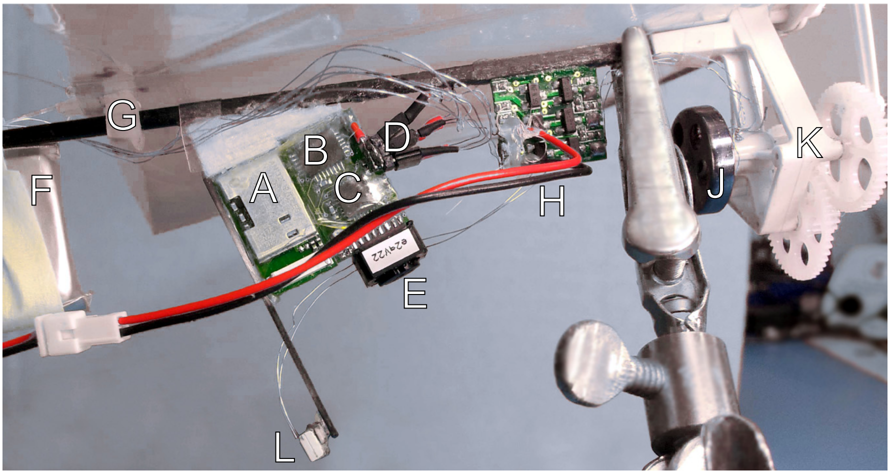


Fig. 4 Side view of the DelFly. **(A)** Bluetooth module, **(B)** 3-axis gyroscope, **(C)** CPU, **(D)** Servo and motor connectors (from top to bottom: motor controller, elevator servo, rudder servo), **(E)** WiiMote Camera, **(F)** 180mAh LiPo battery, **(G)** Trailing edge tensioner, **(H)** Motor controller, **(J)** Brushless motor, **(K)** Gear housing, **(L)** Tracking LED.

stripped down to 0.33 gram at 8 by 8 by 4 millimeter and placed onboard looking forward. It provides 200Hz pixel positions of up to 4 LED. Finally the Invensense ITG3200 MEMS 3-axis gyro provides up to 1kHz of inertial measurements. An overview picture is found in Fig. 4.

The WiiMote camera is chosen because it is cheap, available, fast, small, and all the preprocessing is already done by the build-in integrated circuit (IC) in the camera. Calibration tests using stepper-motor tables with a moving LED showed that no camera distortion worth un-distorting was present. Furthermore it interprets the image, finds the four brightest Infrared (IR) point, and calculates the respective x and y coordinates, which takes a significant workload away from the autopilot microcontroller. The WiiMote camera has a pixel resolution of 128 by 96 of sufficient quality to allow interpolation up to 1024 by 768 subpixels. This corresponds to 0.03° with the 44 by 33 degree field of view.

2.2 *Off Board Tracking System*

The tracking system consists of two non-modified WiiMote controllers mounted on a rigid support. Using multiple view camera geometry the observed LED position is reconstructed in 3D.

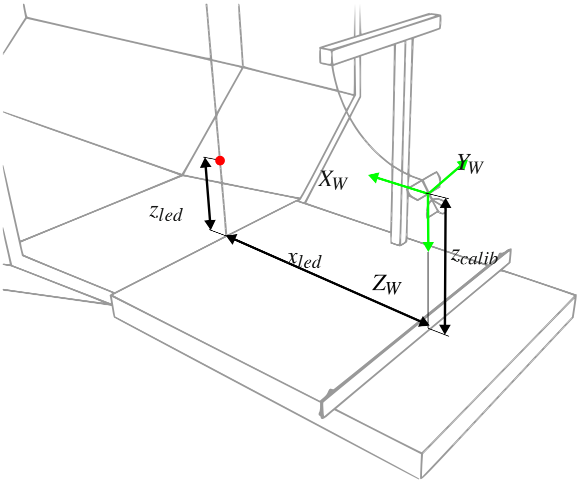


Fig. 5 Schematic view of the OJF (Open Jet Facility) wind tunnel. The octagonal wind tunnel nozzle is on the left, with a LED (red) attached to a fish wire stretched over the opening. The DelFly II is attached to the T-shaped beam by means of a thin wire. The tracking cameras are mounted on the horizontal beam across the platform, and look towards the DelFly at a 45° angle.

3 Sensor Fusion

3.1 Linear LS Triangulation

After un-distortion of lens effects, a camera can be well represented by the pinhole model [14] which maps point **x** in homogeneous world coordinates into its projection **u**.

$$s\mathbf{u} = N[R|t]\mathbf{x}$$

The camera observation **u** of an observed LED at position **x** is defined by $\mathbf{u} = P\mathbf{x}$, with $\mathbf{x} = (x,y,z,w)$ the position vector and P is the camera matrix. Vector **u** is in homogeneous coordinates, i.e. $\mathbf{u} = s(u,v,1)^\top$, where u and v are the observed point coordinates, and s is an unknown scale factor. Denoting $p_i^\top$ as the ith row of matrix P, we can rewrite $\mathbf{u} = P\mathbf{x}$ as

$$su = \mathbf{p}_1^\top \mathbf{x}, \quad sv = \mathbf{p}_2^\top \mathbf{x}, \quad s = \mathbf{p}_3^\top \mathbf{x} \tag{1}$$

Eliminating s using the last equation, we have

$$u\mathbf{p}_3^\top \mathbf{x} = \mathbf{p}_1^\top \mathbf{x} \tag{2}$$

$$v\mathbf{p}_3^\top \mathbf{x} = \mathbf{p}_2^\top \mathbf{x} \tag{3}$$

From the two views we have 4 linear equations in the coordinates of $\mathbf{x}$, written in the form $A\mathbf{x} = \mathbf{0}$ with A a 4×4 matrix. For non-infinity target $\mathbf{x} = (x, y, z, 1)^\top$ is set to reduce homogeneous equations to a set of four non-homogeneous equations with three unknowns. To combine the measurements, the observed point $\mathbf{x}$ is also expressed in the reference frame of the second camera. This is done introducing rotation matrix R and translation vector t which describe the known relative position and orientation of the two cameras. We also rewrite R and t in homogeneous coordinate format. The camera matrixes for cameras 1 and 2 are

$$P_1 = N_1[\mathbf{I}_3 \ \mathbf{0}] \tag{4}$$

$$P_2 = N_2[R \ T] \tag{5}$$

Now using Eq. 3 and considering both measurements with the camera matrices P_1 and P_2, we can construct matrix A as

$$A = \begin{bmatrix} u_1\mathbf{p_1}^\top(3) - \mathbf{p}_1^\top(1) \\ v_1\mathbf{p_1}^\top(3) - \mathbf{p}_1^\top(2) \\ u_2\mathbf{p_2}^\top(3) - \mathbf{p}_2^\top(1) \\ v_2\mathbf{p_2}^\top(3) - \mathbf{p}_2^\top(2) \end{bmatrix} \mathbf{x}$$

where the number in brackets indicates the row vector taken from the respective matrix. The fourth element of $\mathbf{x}$ is 1 and brought to the right because of the assumption of the homogeneous coordinate being finite. Writing A as $[\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3 \ \mathbf{a}_4]$ with each vector $\mathbf{a}_n$ representing a column, we get

$$\left[\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3\right] (x, y, z)^\top = -\mathbf{a}_4$$

This non-homogeneous equation can be solved by using the left pseudo inverse

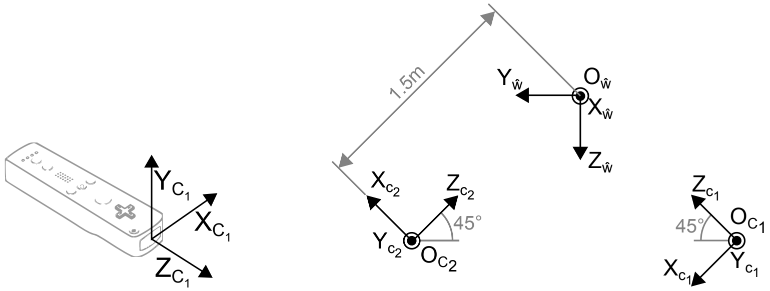
$$\mathbf{x} = -(A^\top A)^{-1} A^\top \mathbf{a}_4$$

which gives $\mathbf{x}_{C_1}$ relative to camera 1, as depicted by the subscript C_1. To express $\mathbf{u}$ in the windtunnel coordinates $\mathscr{F}_{\hat{w}}$ (see Fig. 6) an extra transformation is applied

$$\mathbf{x}_{\hat{w}} = R_{C_1}^{\hat{w}} (\mathbf{x}_{C_1} + \mathbf{T}_{C_1}^{\hat{w}}) \tag{6}$$

where $\mathbf{x}_{\hat{w}}$ is the coordinate in $\mathscr{F}_{\hat{w}}$, $R_{C_1}^{\hat{w}}$ the rotation matrix describing the rotation from $\mathscr{F}_{C_1}$ to $\mathscr{F}_{\hat{w}}$, and where $\mathbf{T}_{C_1}^{\hat{w}}$ describes the position of the origin $\mathscr{O}_{\hat{w}}$ relative to the origin of $\mathscr{O}_{C_1}$, expressed in $\mathscr{F}_{C_1}$. Finally the rotation and translation of the dual camera setup (Fig. 6) are

$$\mathbf{T}_{C_1}^{\hat{w}} = \begin{bmatrix} 0 \\ 0 \\ 1.5 \end{bmatrix} \qquad R_{C_1}^{\hat{w}} = \begin{bmatrix} 0 & 1 & 0 \\ \cos 45^\circ & 0 & \cos 45^\circ \\ \cos 45^\circ & 0 & -\cos 45^\circ \end{bmatrix} \tag{7}$$



(a) Single WiiMote

(b) Tracking System Setup

Fig. 6 Axis definition of a Wiimote and Tracking System with two WiiMotes. The realive positioning is shown in subfigure (b).

3.2 Velocity

The velocity is obtained by taking the discrete derivative directly from the position measurements.

$$v_k = \frac{x_k - x_{k-1}}{\Delta_T}$$

where v_k is the velocity at time step k, x_k the position at time k and Δ_T the time step size. This is done by a discrete Kalman filter to smoothen the results and fill the gaps when the LED was not detected during a frame.

3.3 Heading

To significantly improve the observability of the DelFly attitude, during slow hovering flight of DelFly with high pitch angle a heading reference is computed from an on-board camera combined with an IR Light Emitting Diode (LED) placed in the middle of the wind tunnel. The horizontal pixel coordinate **u** as seen in DelFly camera frame $\mathscr{F}_C$ (Fig. 7) of the reference LED is transformed to the heading χ in degrees by

$$\chi = s_u(u - c_u) \tag{8}$$

where s_u is the angle one horizontal pixel represents and c_u the center pixel that defines the mid point of the field of view in pixels.

At faster forward flight the DelFly flies at much smaller pitch angles and the on-board camera is no longer looking forward and is therefore unable to see the reference LED inside the wind tunnel. The heading χ is therefore calculated from

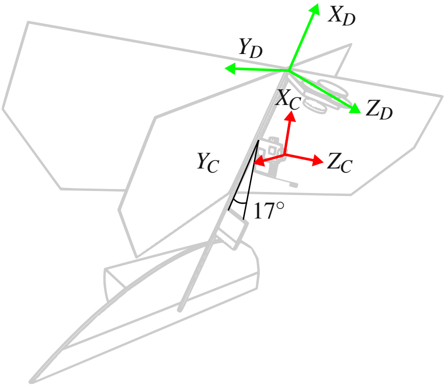


Fig. 7 Axis definition of the Delfly body reference frame $\mathscr{F}_D$ and the on-board camera reference frame $\mathscr{F}_C$. The autopilot PCB is rotated 17 degrees around the Y_D axis. The on-board camera reference frame is defined with the X-axis pointing upward, the Y-axis pointing to the negative Y_D axis, and Z-axis pointing in the flight direction and indicated with a C subscript.

difference in horizontal positions of 2 LED on the ground about a meter in front of the DelFly

$$\chi = \tan^{-1}\left(\frac{u_1 - u_2}{v_1 - v_2}\right) \tag{9}$$

where u is the horizontal pixel coordinate and v the vertical pixel coordinate of their respective pixel. Small-angle approximation is applied to save on calculation time on the on-board processor.

3.4 Calibration and Accuracy

At first the *extrinsic* parameters needed for the Linear-LS triangulation method of the stereoscopic camera pair is determined using OpenCV's *stereoCalibration2* routine. Then the *inertial* calibration, which maps the observations from the stereo camera frame $\mathscr{F}_w$ to the actual wind tunnel flight frame $\mathscr{F}_{\hat{w}}$. A rectangular LED board was placed with one corner at the point that should become the origin of the new coordinate system and its two side nicely aligned with the tunnel. A reading with the tracking system was taken and the TRIAD algorithm was used to acquire the correction rotation matrix and translation vector.

The accuracy of the hardware and algorithm was checked by moving the LED in a predetermined pattern by means of a micrometer precise stepper motor table. The setup was tested with a dog-leg pattern. The LED was moved 300 mm from the left to the right, and then 200 mm backwards (positive z direction). (Fig. 8)

The standard deviation of the error is 0.3734 mm in the x-direction and 0.6378 mm in the z-direction. It can be seen that the error shows a saw-tooth behavior probably caused by the pixel rounding of the cameras. Nevertheless the average deviation is sub millimeter and maximum error never more than 2 millimeters.

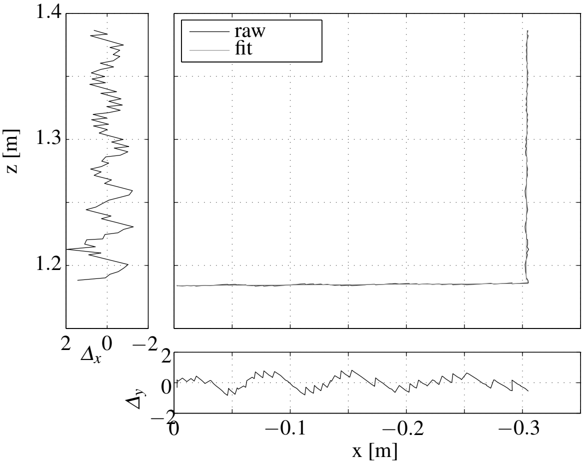


Fig. 8 Calibration track with corresponding errors. Coordinates are expressed in the camera reference frame $\mathscr{F}_{C_1}$.

4 Controller Architecture

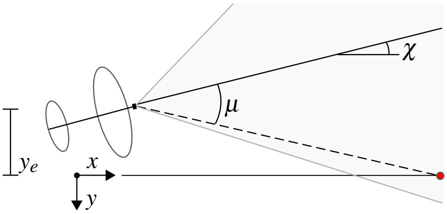
For the controller design, an important factor is the limited processing power available. Therefore the design goals focus on getting the required performance with sufficiently low processing power.

4.1 Lateral Control of Slow Forward Flight

As the distance of the DelFly to its heading LED is constant, the angle μ at which DelFly sees the target is homogenous. A heading controller is set to minimize μ. This constant steering towards the LED $K_\mu \mu$ creates a P-controller minimizing the lateral offset (Fig. 9). A damping term $K_p p_C$ is added based on low pass filtered low latency gyroscope measurements p_C, which measures rotations around the onboard camera X-axis. K_p is the so-called roll gyro gain.

During slow hovering flight the DelFly II flies almost nose up at very high pitch angles, while the rudder generated moments around the Z_D. To dampen Z_D

Fig. 9 Schematic top-view of an lateral error. The red dot indicates the location of the infra red reference LED. μ is the angle towards the LED while χ is the heading of DelFly II.



oscillations a yaw damper is added constructed from the rates r_C and p_C aligned with the Z_C and X_C axes respectively (Fig. 7)

$$r_D = r_C \cos(17^\circ) + p_C \sin(17^\circ) \tag{10}$$

The yaw damper is inserted as $K_r r_D$ where K_r is the yaw damper gain. To reduce the steady state error an integrator term $K_y^I \int(y - y_{ref})$ is added, where K_y^I is the integrator gain. The total controller for slow forward flight becomes after dropping out y_{ref} which is zero for the middle of the wind tunnel

$$\delta_r = K_\mu \mu + K_p p_C + K_r r_D + K_y^I \int y + K_k \tag{11}$$

where K_k is a value trim to be set manually when necessary.

4.2 Lateral Control of Forward Flight

When the DelFly is flying at higher velocities, the reference LED right in front of the DelFly is out of view, because the camera is looking almost downwards. The first part of the total controller shown in Eq. 11 is replaced by

$$\delta_r = K_\chi \left(y_e K_y - \chi\right) \tag{12}$$

including the lateral off-set $y_e = (y_{ref} - y)$. The total three loop control scheme is shown in Fig. 10.

Including an integrator term, setting the desired lateral offset y_{ref} and some rewriting the total controller becomes

$$\delta_r = K_y y + K_\chi \chi + K_r r + K_y^I \int y + K_k \tag{13}$$

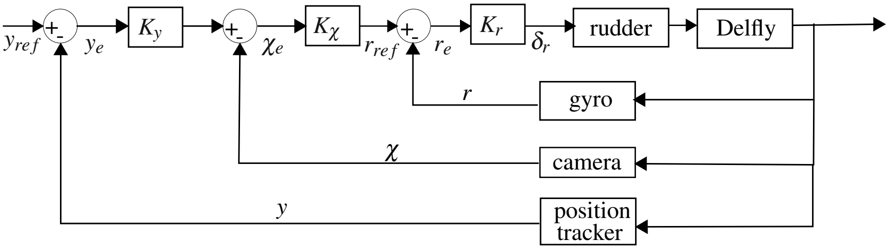


Fig. 10 DelFly II lateral position controller

4.3 Longitudinal Control of Slow Forward Flight

When flying at low speeds, the thrust vector of the flapping wing DelFly II is pointing almost straight up. It mostly controls the climb rate while the elevator influences the forward position. During forward flight however throttle most influences the forward speed and elevator the altitude. So at slow hovering flight for the vertical position z we have

$$\delta_{th} = f_{PID}(z - z_{ref}) \tag{14}$$

and the horizontal position x

$$\delta_e = f_{PID}(x - x_{ref}) \tag{15}$$

where δ_{th} and δ_e are resp. the thrust and elevator input, z and x the current horizontal and vertical position and z_{ref}, x_{ref} the reference position we want to achieve. During flight the battery will drain and the voltage significantly drops, as shown in Fig. 11. This is well handled by the integral term in the controller removing the need for feed forward compensations.

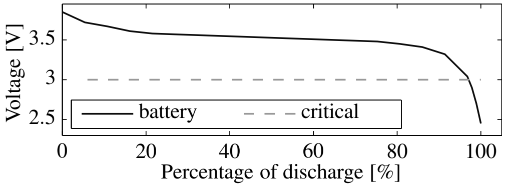


Fig. 11 Typical Lithium-Polymer discharge curve of a single cell. DelFly II uses a single 180mAh cell as power source. As the voltage drops, the current is increased to keep the same power level.

For zero references the total control law for the throttle setting becomes

$$\delta_{th} = K_z z + K_z^I \int z + K_n \tag{16}$$

with K_z the proportional gain, K^I the integral gain and K_n the trim setting. Due to the very low wing loading high damping of DelFly II, the d-gain can be left out. For the elevator input

$$\delta_e = K_x x + K_x^I \int x + K_q q + K_m \tag{17}$$

is used where a pitch damper $K_q q$ is added and K_m is the trim setting.

4.4 Longitudinal Control of Forward flight

Unlike the slow hovering flight with rotorcraft-like characteristics, when the DelFly flies at higher velocities, the pitch angle decreases to about 30° for velocities around 2 m/s and upwards and the DelFly starts to behaves like a conventional fixed wing aircraft. The controls are configured to pitch from altitude error and throttle from speed error

$$\delta_e = f_{PID}(z - z_{ref}) \tag{18}$$

$$\delta_{th} = f_{PID}(x - x_{ref}) \tag{19}$$

The total control law for the throttle setting becomes then

$$\delta_{th} = K_x\, x + K_{v_x} v_x + K_x^I \int (x) + K_n \tag{20}$$

and for the elevator input

$$\delta_e = K_z\, z + K_{v_z} v_z + K_z^I \int (z) + K_q q + K_m \tag{21}$$

where $K_q\ q$ is the pitch damper, K_n and K_m the manual trim settings and the zero reference values x_{ref} and z_{ref} have been left out. The control laws very similar to the horizontal flight laws in Eq. 16 and Eq. 17, except that the axes on which controls surface act on are exchanged and position derivative terms are needed.

5 Station Keeping Experiments

A series of test and tuning flights were performed in the Open Jet Facility windtunnel of the Delft University of Technology. All gains were manually tuned in the various flight regimes. One must keep in mind that the hand built DelFly II weighs only 16 grams[8]. Imperfections, wear over time, undesired vibrations of the half a gram servos, resolution of all actuators and motor control, residual vortices in the wind tunnel as well as the lack of a full dynamics model for the development of the controller play important roles in the achievable performance.

A first series of tests tried to keep DelFly II as close as possible to the center of the tunnel. Results presented here concern slow hovering flight at $0.8 m/s$. Fig. 12 show a test flight with relatively low value for the integrator gains. It shows how an untrimmed DelFly II rejects steady state errors.

A longer 14 minute flight in Fig. 13 shows the integrator nicely compensates for the dropping battery voltage (Fig. 11) by increasing the throttle over time.

Looking closer at the position signal, very low frequency oscillations are seen in the y and z direction, illustrated in Fig. 14. The resolution of throttle control of DelFly II is only 170 discrete steps due to brushless motor controller constraints. This motor controller was specially developed to be able to face the highly nonconstant load of the pushrods to the wings, flapping at about 13Hz. During a single

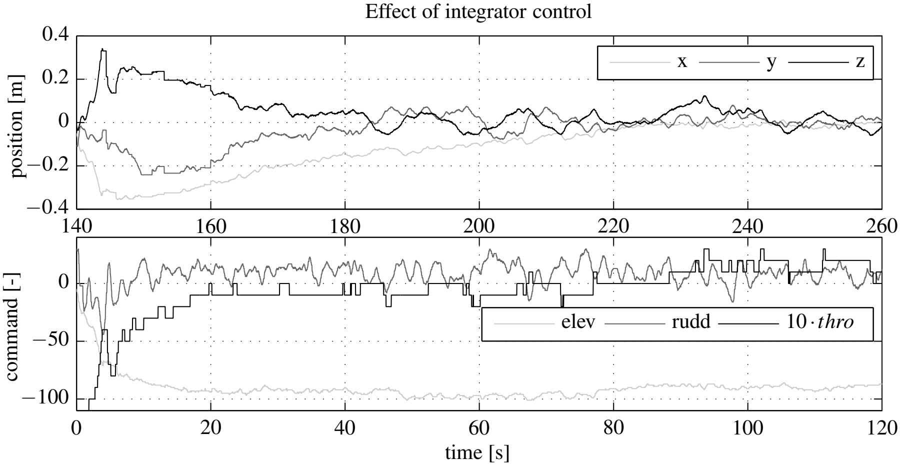


Fig. 12 Initial test run with low integral gains while tuning the gains of the DelFly II without a full dynamic model

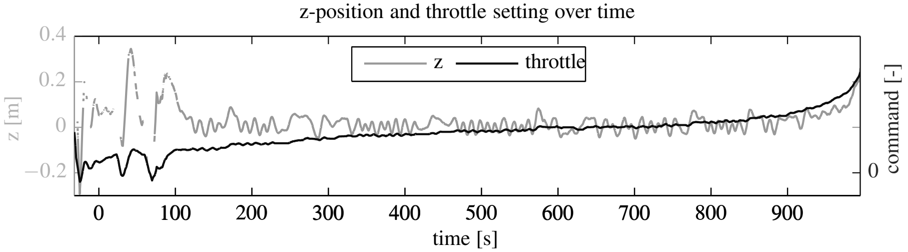


Fig. 13 Required Power as function of Battery Voltage

flap cycle the motor must accelerate and decelerate with a 1 to 2 ratio. Additional trade-offs in the motor controller concerning efficiency make it hard to resolve this power resolution limitation.

Upon inspection it can be seen that the oscillations are driven by the throttle alternating between two values. The DelFly descents every time the throttle is reduced one unit, and ascends when the throttle is increased. When the battery voltage decreases over time as seen in Fig. 11, eventually the required power matches a settable power level and oscillations disappear. Table 1 shows the performance in this case and stands for the performance that could be obtained of finer throttle control were possible.

The manual tuning with the unknown dynamics of DelFly was done by performing many runs with different gains and searching for better RMS performance. Fig. 15 illustrates the effect of several yaw damping gains and Fig. 16 of several pitch damping gains on the station keeping performance of the DelFly II.

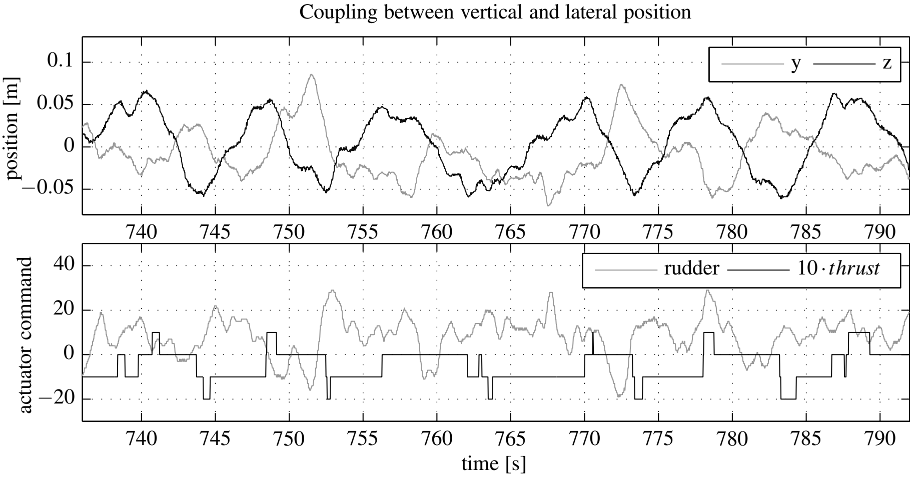


Fig. 14 Close-up of a minute of autonomous flight. The throttle has been scaled 10× for clarity. It is alternating between one or two values. The y- and z-position are coupled: the DelFly moves up and right, or down and left, with a period of about 8 seconds.

Table 1 Maximum deviations from the windtunnel center and the RMS error values with correct power setting.

direction	maximum deviation [cm]	RMS [cm]
x	1.6	0.82
y	4.3	1.8
z	1.5	0.95

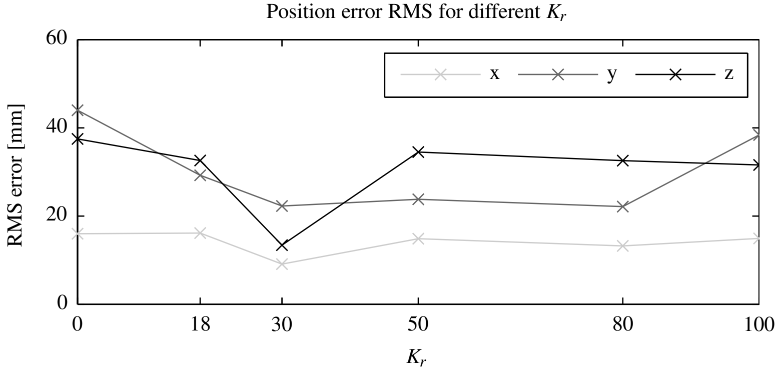


Fig. 15 RMS error of the x,y and z position for different gain settings of the yaw damper. The sample duration was about 40 seconds for each setting.

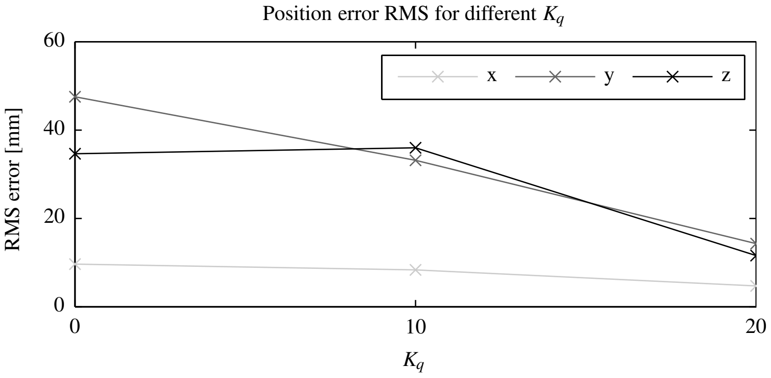


Fig. 16 RMS error of the x,y and z position for different gain settings of the pitch damper

6 Transient Behavior Experiments

In order to further analyze the stability and disturbance rejection of the system, and simultaneously obtain new information about the transient behavior of DelFly II, a series of step response tests were performed. These are depicted in Figures 17 to 20.

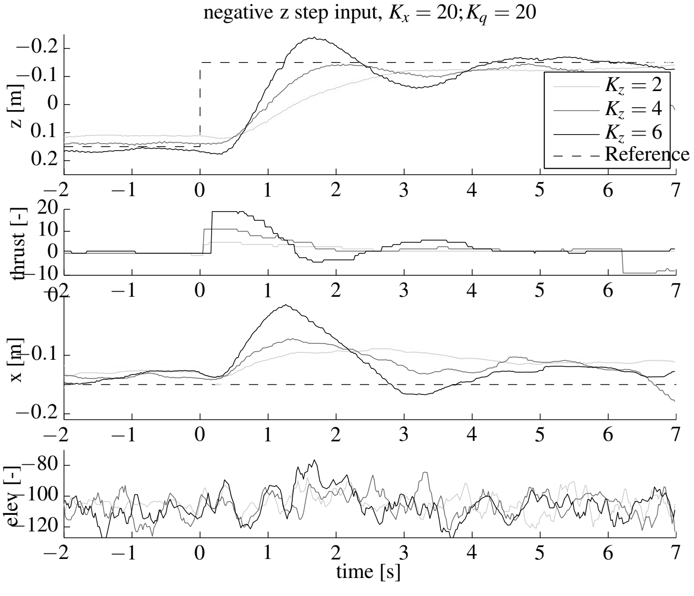


Fig. 17 Step 30cm up. The throttle increase clearly results in a forward motion increase. For a gain of $K_z = 6$ the system shows under damped behavior, at K_2 over damped behavior.

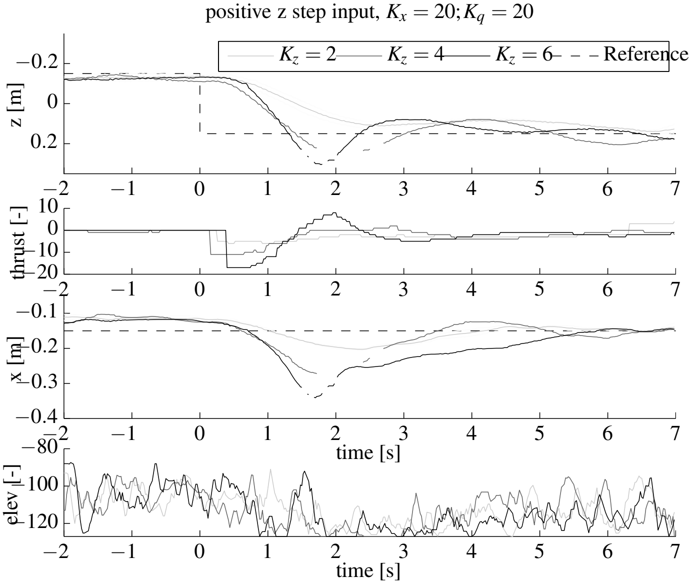


Fig. 18 Step 30cm down. DelFly is drifted backwards due to the lower thrust setting which perturbs the horizontal equilibrium. Comparing with the step up, the same gain $K_z = 4$ shows overshoot instead of critical damping behavior, showing the high non-linearity in the behavior of DelFly II.

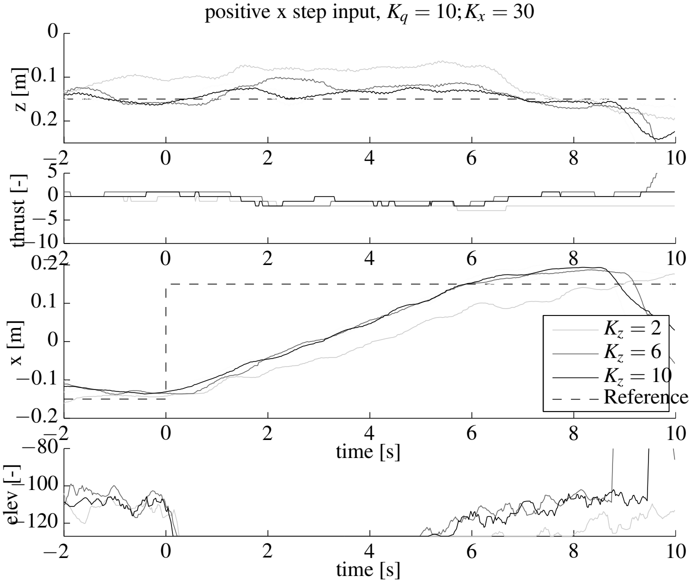


Fig. 19 Horizontal 30cm step forward. The DelFly reacts very slowly due to saturation of the elevator. As slow hovering flight is on the negative side of the power curve, at higher forward speed less thrust is needed, which causes an upward perturbation.

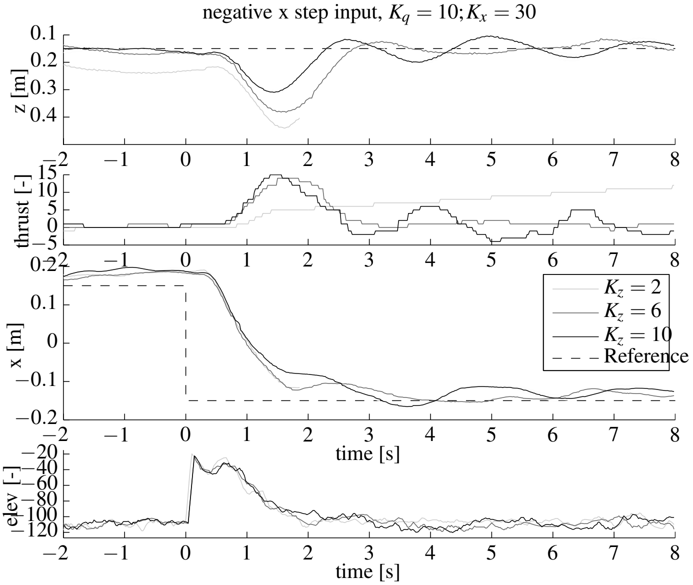


Fig. 20 Horizontal 30cm back. DelFly reacts very quickly, drifting backward with the wind-tunnel flow. When flying at lower forward speed, extra thrust is needed which is visible as a descending perturbation for the altitude controller.

7 Conclusions and Recommendations

A low-cost high performance control system is developed. The weight of the on-board part is less than two grams, while the sensors consists of only three WiiMote sensors, a few LED and a gyroscope chip. The total system station keeping performance was shown to be in the order of centimeters. The most important factors that limit the performance are the resolution of the power setting. Then the residual vorticity in the tunnel and unsteady periodic aerodynamics together with the 16 grams very low wing loading vehicle impose considerable perturbations to the control system, while manufacturing imperfections, wear over time and undesired vibrations of the miniature actuators also impact the performance. Nevertheless the achieved performance is still largely sufficient for many new research purposes, amongst which the analysis of the influence of fixing the fuselage in flapping wing research and more precise performance characteristics of DelFly II.

Aspects that need additional attention in further work are a lower latency data link to reduce delays and make them more constant than the Bluetooth modems. But most of all a increased throttle resolution, for instance using modulation techniques.

References

1. Baek, S.S.: Autonomous Ornithopter Flight with Sensor-Based Behavior. Ph.D. thesis, University of California, Berkely (2011)
2. Berg, C.V., Ellington, C.: The three-dimensional leading-edge vortex of a 'hovering' model hawkmoth. Phil. Trans. R. Soc. Lond. B, 329–340 (1997)
3. Biesel, W., Nachtigall, W.: Pigeon flight in a wind tunnel *. Journal of Comparative Physiology B: Biochemical, Systemic, and Environmental Physiology, 99–109 (1987)
4. Birch, J., Dickson, W., Dickinson, M.: Force production and flow structure of the leading edge vortex on flapping wings at high and low reynolds numbers. Journal of Experimental Biology 207(7), 1063–1072 (2004)
5. Bomphrey, R.J., Lawson, N.J., Taylor, G.K., Thomas, A.L.R.: Application of digital particle image velocimetry to insect aerodynamics: measurement of the leading-edge vortex and near wake of a Hawkmoth. Experiments in Fluids 40(4), 546–554 (2006), http://www.springerlink.com/index/10.1007/s00348-005-0094-5, doi:10.1007/s00348-005-0094-5
6. Bradshaw, N.L., Lentink, D.: Aerodynamic and Structural Dynamic Identification of a flapping wing Micro Air Vehicle. Aerospace Engineering, 6248–6248 (August 2008)
7. Clercq, K.M.E.D., Kat, R.D., Remes, B., Van, B.W., Bijl, H.: on DelFly II: Unsteady Lift Enhancement. Measurement 1(4) (2009)
8. Croon, G.C.H.E.D., Clercq, K.M.E.D., Ruijsink, R., Remes, B.: Design, aerodynamics, and vision-based control of the DelFly 1(2), 71–98 (2009)
9. De Clercq, K.M.E., Kat, R.D., Remes, B., Van Oudheusden, B.W., Bijl, H.: Flow visualization and force measurements on a hovering flapping wing MAV Delfly II. International Journal of Micro Air Vehicles (2009)
10. Groen, M.: PIV and force measurements on the flapping-wing MAV DelFly II (December 2010), http://www.delfly.nl/MSc-Groen.pdf
11. Hedenström, A., Muijres, F.T., Busse, R., Johansson, L.C., Winter, Y., Spedding, G.R.: High-speed stereo DPIV measurement of wakes of two bat species flying freely in a wind tunnel. Experiments in Fluids 46(5), 923–932 (2009), http://www.springerlink.com/index/10.1007/s00348-009-0634-5, doi:10.1007/s00348-009-0634-5
12. Kim, D.K., Han, J.H., Kwon, K.J.: Wind tunnel tests for a flapping wing model with a changeable camber using macro-fiber composite actuators. Smart Materials and Structures 18(2) (2009)
13. Liu, H., Kawachi, K.: A numerical study of insect flight. Journal of Computational Physics 146, 124–156 (1998)
14. Page, G.F.: Multiple view geometry in computer vision, by richard hartley and andrew zisserman, vi+560 p. CUP, Cambridge (2003) ISBN 0-521-54051-8; Robotica 23(2), 271–271 (2005), http://dx.doi.org/10.1017/S0263574705211621, doi:10.1017/S0263574705211621
15. Pines, D.J., Bohorquez, F.: Challenges facing future micro-air-vehicle development. Journal of Aircraft 43(2), 290–305 (2006), http://cat.inist.fr/?aModele=afficheN\&cpsidt=17667289
16. Raffel, M., Willert, C., Wereley, S.T., Kompenhans, J.: Particle Image Velocimetry, 2nd edn. Springer (2007)
17. Thielicke, W., Kesel, A., Stamhuis, E.: Reliable force predictions for a flapping-wing micro air vehicle: A "vortex-lift" approach. International Journal of Micro Air Vehicles 3(4), 201–216 (2011)

18. Ulrich, E.R., Pines, D.J., Humbert, J.S.: From falling to flying: the path to powered flight of a robotic samara nano air vehicle. Bioinspiration & Biomimetics 5(4), 045009 (2010), http://www.ncbi.nlm.nih.gov/pubmed/21098960, doi:10.1088/1748-3182/5/4/045009
19. Willmot, P., Ellington, C., Thomas, A.: Flow visualization and unsteady aerodynamics in the flight of the hawkmoth, manduca sexta. Phil. Trans. (1997)
20. Windte, J., Pfingsten, K.C.: On 2d motion parameters for flapping wing propulsion. In: European Micro Air Vehicle Conference and Competitions 2004, pp. 1–10 (2004)
21. Yang, T., Wei, M., Zhao, H.: Numerical study of flexible flapping wing propulsion. AIAA Journal 48(12), 2909–2915 (2010)

Application of Frequency-Limited Adaptive Quadrocopter Control

Kirk Y.W. Scheper, Daniel Magree, Tansel Yucelen,
Gerardo De La Torre, and Eric N. Johnson

Abstract. Adaptive control systems have long been used to effectively control dynamical systems without excessive reliance on system models. This is due mainly to the fact that adaptive control guarantees stability, the same however, cannot be said for performance; adaptive control systems may exhibit poor tracking during transient (learning) time. This paper discusses the experimental implementation of a new architecture to model reference adaptive control, specifically, the reference system is augmented with a novel mismatch term representing the high-frequency content of the system tracking error. This mismatch term is an effective tool to remove the high frequency content of the error signal used in the adaptive element update law. The augmented architecture therefore allows high-gain adaptation without the usual side-effect of high-frequency oscillations. The proposed control architecture is validated using the Georgia Tech unmanned aerial vehicle simulation tool (GUST) and also implemented on the Georgia Tech Quadrocpoter (GTQ). It is shown that the new framework allows the system to quickly suppress the effect of uncertainty without the usual side effects of high gain adaptation such as high-frequency oscillations.

1 Introduction

Unmanned Aerial Vehicles (UAV), like many other forms of robots, are often used to perform tasks too dangerous or repetitive for humans. Examples in industry show UAVs being used for surveillance and mapping of areas after natural disasters or in enemy territory in the case of military application. UAVs should operate with little user influence in hostile changing environments. That is, UAVs should have the ability to quickly adapt to their surroundings and act appropriately to changing

Kirk Y.W. Scheper · Daniel Magree · Tansel Yucelen · Gerardo De La Torre · Eric N. Johnson
Georgia Institute of Technology, Atlanta, GA 30332, USA
e-mail: k.y.w.scheper@student.tudelft.nl, dmagree@gatech.edu,
{tansel.yucelen,glt3,eric.johnson}@ae.gatech.edu

conditions. This paper aims to extend this research by providing a solution to high-gain adaptive control without high-frequency oscillations, which ultimately allows for fast adaptation of advanced control systems in UAVs.

It is well known that standard model reference adaptive control (MRAC) approaches employ high-gain learning rates to achieve fast adaptation in order to rapidly reduce system tracking errors in the face of system uncertainties. High-gain learning rates, however, lead to increased controller efforts and can cause high-frequency oscillations, which may violate actuator rate saturation constraints [4] and/or excite unmodeled system dynamics [10, 11] resulting in system instability.

Motivated from this standpoint, this paper resorts to a recently developed frequency-limited adaptive control architecture [12]. The contribution of this framework comes from suppressing undesired high frequency system oscillations using a new reference system architecture. Specifically, the proposed reference system captures a desired closed-loop dynamical system behavior modified by a novel mismatch term representing the high-frequency content between the uncertain dynamical system and this reference system, i.e., the system error. In particular, this mismatch term allows the limiting of the frequency content of the system error dynamics both in transient and steady state, which is used to drive the adaptive controller. The purpose of this methodology is to prevent the update law from attempting to learn the high-frequency content of the system error. This key feature of the framework yields fast adaptation without incurring high-frequency oscillations in the transient performance.

The proposed augmentation to the reference system is implemented on a quadrocopter model simulated in the Georgia Institute of Technology unmanned aerial vehicle simulation tool (GUST) developed by the Unmanned Aerial Vehicle Research Facility (UAVRF). The GUST software package combines a high-fidelity vehicle and environment model, onboard flight control software, and ground station software. GUST may be operated in hardware in the loop (HITL) mode or software in the loop (SITL) mode. In HITL mode, the flight control software and ground station interface with physical sensors, actuators, and communication links. In SITL mode, the flight control software and ground station interface with the vehicle model and simulated communication links. This design ensures that the same flight control software is used in simulation and in flight reducing software development time.

The vehicle model used in the SITL mode is a six rigid body degree of freedom model complete with additional engine and rotor dynamics. The vehicle model simulates sensor noise, delay, location, orientation, actuator dynamics and saturation. For this paper, the SITL mode is used to obtain the simulation results and flight tests were conducted using the HITL mode onboard the flight platform.

The flight platform used in this paper is a quadrocopter UAV based on the commonly used Pelican airframe [2] from Ascending Technologies (AscTec). The UAVRF uses this platform to perform research on indoor flight navigation and guidance algorithms and is referred to as the Georgia Tech Quadrocopter (GTQ). The GTQ is a fully autonomous vehicle which uses a laser SLAM based navigation system to traverse cluttered indoor environments. A detailed description of the platform

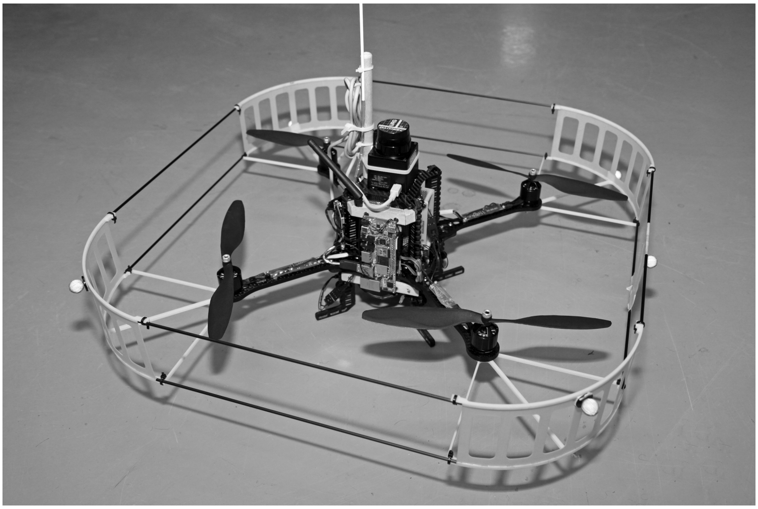


Fig. 1 Picture of the AscTec Pelican indoor flight platform known as GTQ. This is the flight platform used to evaluate the performance of the frequency-limited adaptive control architecture in this paper.

and navigation system can be found in [3]. A picture of the GTQ can be seen below in Figure (1).

In the following sections, we first outline the adaptive update law used and give a description of the proposed augmented reference system architecture. Then the control system implemented is described in detail. High fidelity simulation and flight test results are then discussed. The performance of the frequency-limited architecture is then compared to a that of standard MRAC and the closed loop reference model framework as described in [9, 5].

The notation used in this paper is fairly standard. Specifically, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+$ (resp., $\overline{\mathbb{R}}_+$) denotes the set of positive (resp., non-negative-definite) real numbers, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., non-negative-definite) real matrices, $\mathbb{S}^{n\times n}$ denotes the set of $n \times n$ symmetric real matrices, $\mathbb{D}^{n\times n}$ denotes $n \times n$ real matrices with diagonal scalar entities, $(\cdot)^T$ denotes transpose, $(\cdot)^{-1}$denotes inverse, and $\triangleq$ denotes equality by definition.

2 Preliminaries

We will begin by considering the nonlinear dynamical system given by

$$\dot{x}_p = A_p x_p(t) + B_p \Lambda u(t) + B_p \delta_p(x_p(t)), \;\; x_p(0) = x_{p_0}, \;\; t \in \overline{\mathbb{R}}_+, \tag{1}$$

where $x_p(t) \in \mathbb{R}^{n_p}$ is the state vector available for feedback, $u(t) \in \mathbb{R}^m$ is the control input, $\delta_p : \mathbb{R}^{n_p} \to \mathbb{R}^m$ is an *uncertainty*, $A_p \in \mathbb{R}^{n_p \times n_p}$ is a known system matrix, $B_p \in \mathbb{R}^{n \times m}$ is a known control input matrix, and Λ is the unknown control effectiveness matrix. Furthermore, we assume the pair (A_p, B_p) is controllable and the uncertainty can be parametrized as

$$\delta_p(x) = W_p^T \sigma_p(x_p), \tag{2}$$

where $W_p \in \mathbb{R}^{s \times m}$ is an *unknown* weight matrix and $\sigma_p : \mathbb{R}_p^n \to \mathbb{R}^s$ is a known basis function of the form $\sigma_p(x_p) = [\sigma_{p_1}(x_p), \sigma_{p_2}(x_p), \ldots, \sigma_{p_s}(x_p)]^T$.

Remark 1. For the case where the basis function $\sigma_p(x_p)$ is unknown, parametrization in (2) can be relaxed, for example, by considering $\delta_p(x_p) = W_p^T \sigma_p^{nn}(V_p^T x_p) + \varepsilon_p^{nn}(x_p), x_p \in \mathfrak{D}_{x_p}$, where $W_p \in \mathbb{R}^{s \times m}$ and $V_p \in \mathbb{R}^{n_p \times s}$ are unknown weight matrices, $\sigma_p^{nn} : \mathfrak{D}_{x_p} \to \mathbb{R}^s$ is a known basis composed of neural network functional approximators, $\varepsilon_p^{nn} : \mathfrak{D}_{x_p} \to \mathbb{R}^m$ is an unknown residual error, and $\mathfrak{D}_{x_p}$ is a compact subset of $\mathbb{R}^{n_p}$ [6].

Remark 2. If the uncertainty is time varying the adaptive element update law can be extended using, for example, Theorem 7.1 from [12].

To address command following, let $c(t) \in R^{n_c}$ be a given bounded piecewise continuous command for tracking (or $c(t) = 0$ for stabilization) and $x_c(t) \in R^{n_c}$ be the integrator state satisfying

$$\dot{x}_c = E_p x_p(t) - c(t), \;\; x_c(0) = x_{c_0}, \tag{3}$$

where $E_p \in \mathbb{R}^{n_c \times n_p}$ allows to choose a subset of $x_p(t)$ to be followed by $c(t)$. Now (1) can be augmented with (2) as

$$\dot{x} = Ax(t) + B\Lambda u(t) + BW_p^T \sigma_p(x_p(t)) + B_r c(t), \;\; x(0) = x_0, \tag{4}$$

where $x(t) \triangleq \left[x_p^T(t), x_c^T(t)\right]^T \in \mathbb{R}^n, n = n_p + n_c$ is the augmented state vector, $x_0 \triangleq \left[x_{p_0}^T, x_{c_0}^T\right]^T \in \mathbb{R}^n$,

$$A \triangleq \begin{bmatrix} A_p & 0_{n_p \times n_c} \\ E_p & 0_{n_c \times n_c} \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{5}$$

$$B \triangleq \begin{bmatrix} B_p^T & 0_{n_c \times m} \end{bmatrix}^T \in \mathbb{R}^{n \times m} \tag{6}$$

$$B_r \triangleq \begin{bmatrix} 0_{n_p \times n_c} & -I_{n_c \times n_c} \end{bmatrix}^T \in \mathbb{R}^{n \times n_c} \tag{7}$$

Next, let us consider the reference system capturing the desired closed-loop dynamical system performance given by

$$\dot{x}_r = A_r x_r(t) + B_r c(t), \;\; x_r(0) = x_{r0}, \tag{8}$$

where $x_r(t) \in \mathbb{R}^n$ is the reference state vector, $A_r \in \mathbb{R}^{n \times n}$ is the Hurwitz reference system matrix, and $B_r \in \mathbb{R}^{n \times m}$ is the command input matrix. Also, their exist a matrix $K \in \mathbb{R}^{m \times n}$ such that $A_r = A - BK$ holds.

Let us now consider the feedback control law given by

$$u(t) = u_n(t) + u_a(t), \tag{9}$$

where $u_n(t) \in \mathbb{R}^m$ and $u_a(t) \in \mathbb{R}^m$ are the nominal and adaptive control laws, respectively. Let u_n be given by

$$u_n(t) = -Kx(t), \tag{10}$$

Combining (9), (10) into (4) yields

$$\dot{x} = A_r x(t) + B_r c(t) + B\Lambda \left[u_a + W^T(t)\sigma(x(t))\right], \tag{11}$$

where $W \triangleq \left[\Lambda^{-1} W_p^T, (\Lambda^{-1} - I_{m\times m})K\right]^T \in \mathbb{R}^{(s+n)\times m}$ is an unknown (aggregated) weight matrix and $\sigma(x(t)) \triangleq \left[\sigma_p^T(x(t)), x^T(t)\right]^T \in \mathbb{R}^{s+n}$ is a known (aggregated) basis function. Now, considering (11), let the adaptive control law be defined as

$$u_a(t) = -\hat{W}(t)\sigma(x(t)), \tag{12}$$

where $\hat{W}(t) \in \mathbb{R}^{s\times m}$ is the estimate of W satisfying the weight update law given by

$$\dot{\hat{W}}(t) = \Gamma\sigma(x(t))e^T(t)PB, \;\; \hat{W}(0) = \hat{W}_0, \tag{13}$$

where $\Gamma \in \mathbb{R}_+^{s\times s} \cap \mathbb{S}^{s\times s}$ is the learning rate matrix, $e(t) \triangleq x(t) - x_r(t)$ is the system error state vector, and $P \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$ is a solution of the Lyapunov equation

$$0 = A_r^T P + PA_r + R, \tag{14}$$

where $R \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$ can be viewed as an additional learning rate. Note that since A_r is Hurwitz, it follows from converse Lyapunov theory [6] that there exists a unique P satisfying (14) for a given R.

The system error dynamics is given from (8), (11) and (12).

$$\dot{e}(t) = A_r e(t) - B\Lambda\tilde{W}^T(t)\sigma(x(t)), \;\; e(0) = e_0, \tag{15}$$

where $\tilde{W}(t) \triangleq \hat{W}(t) - W \in \mathbb{R}^{(s+n)\times m}$ is the weight error and $e_0 \triangleq x_0 - x_{r0}$. Proofs of Lyapunov stability of the weight matrix $W(t)$ and error vector $e(t)$ and the convergence of $e(t) \to 0$ as $t \to \infty$ can be found in reference [1].

3 Frequency-Limited Adaptive Control Framework

The augmented MRAC system architecture, as defined in [12], is achieved by introducing a mismatch term to the reference system dynamics. The mismatch term captures the high-frequency content between the uncertain dynamical system and the reference system. This augmenting of the reference system term allows the designer to limit the frequency content of the system error dynamics which is used to drive the adaptive element. The proposed architecture is visualized in the diagram shown below in Figure (2).

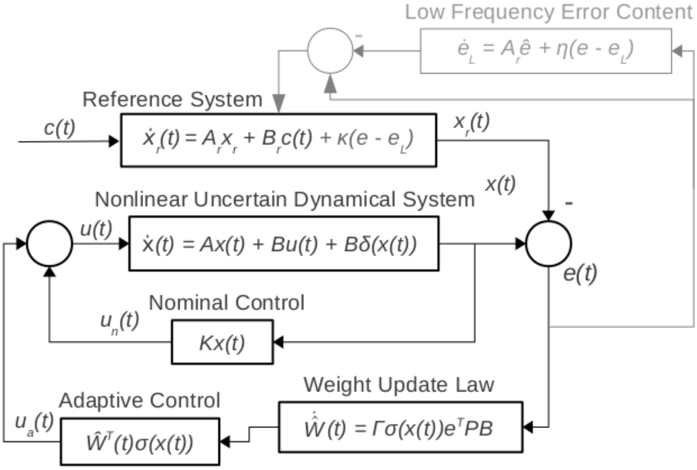

Low Frequency Error Content
$\dot{e}_L = A_r\hat{e} + \eta(e - e_L)$
-
Reference System
c(t)
$\dot{x}_r(t) = A_r x_r + B_r c(t) + \kappa(e - e_L)$
$x_r(t)$
Nonlinear Uncertain Dynamical System
x(t)
-
u(t)
$\dot{x}(t) = Ax(t) + Bu(t) + B\delta(x(t))$
e(t)
Nominal Control
$u_n(t)$
Kx(t)
Weight Update Law
$\dot{\hat{W}}(t) = \Gamma\sigma(x(t))e^T PB$
Adaptive Control
$u_a(t)$
$\hat{W}^T(t)\sigma(x(t))$

Fig. 2 Visualization of the proposed frequency-limited adaptive control architecture. Notice not only is the reference system driven by the external command but also by the mismatch term containing the high frequency error content. The augmented system is shown in grey.

To define our augmented system let us first consider $e_L \in \mathbb{R}^n$, the low pass filtered error of $e(t)$ given by

$$\dot{e}_L = A_r e_L + \eta(e - e_L), \;\; e_L(0) = 0, \tag{16}$$

where $\eta \in \mathbb{R}_+$ is a filter gain. From the low pass error update law it can be seen that the filter gain η is used to limit the bandwidth of this signal. This term must be large enough to capture the system dynamics to be input to the adaptive element but small enough to reject the high frequency content of this signal. Considering this in the classical loop shaping context, the user can now allow faster adaptation of the system to uncertainty by amplifying the adaptive learning rate while suppressing unwanted high frequency adaptation and subsequent high frequency oscillation.

Remark 3. As $\eta \to \infty$ the frequency-limited architecture approaches standard MRAC. As $\eta \to 0$ the frequency-limited architecture approaches the closed loop reference model framework described in [9, 5].

Now, as the purpose of the frequency limitation is to force our error to approach the behavior of the low pass filtered error, we can introduce a mismatch term which is a minimization of the relative distance of the trajectories of the two signals $e(t)$ and $e_L(t)$. This can be captured in the cost function given by $\mathfrak{J}(e, e_L) = \frac{1}{2}||e - e_L||_2^2$, and note that the negative gradient of this cost function with respect to $e(t)$ is given by

$$\frac{\delta[-\mathfrak{J}(e(t) - e_L(t))]}{\delta e(t)} = -(e(t) - e_L(t)) \tag{17}$$

Incorporating the mismatch term into the reference system in (8) results in the following augmented reference system

$$\dot{x}_r = A_r x_r + B_r c(t) + \kappa(e - e_L), \;\; x_r(0) = x_{r_0}, \tag{18}$$

where $\kappa \in \mathbb{R}_+$. Through singular perturbation analysis of the augmented reference system it can be shown that the high frequency content of the error signal is globally exponentially stable and vanishes in a fast manner for a sufficiently high κ[1]. Theorem 5.1 in [12] also discusses that even though the proposed architecture is predicated on a modified reference system given by (18) the augmented system actually converges to the ideal reference system given in (8).

Remark 4. It should be noted that increasing κ with a constant learning rate Γ causes the distance between the augmented system and the ideal reference system to increase so Γ and κ should be increased together for the best overall system performance.

Updating (15) with (18) results in a the system error dynamics now given by

$$\dot{e}(t) = A_r e(t) - B\Lambda \tilde{W}^T(t)\sigma(x(t)) - \kappa(e(t) - e_L(t)), \;\; e(0) = e_0, \tag{19}$$

This framework can now be extended to the flight platform as is discussed in the next section.

4 Application to a High-Fidelity Autonomous Quadrocopter Model

This section describes the implementation of the frequency limitation architecture in the Georgia Tech unmanned aerial vehicle simulation tool (GUST). Additional information of the control architecture can be found in [8].

Let us first consider the nonlinear system which represents the dynamics a free flying body in the following form

$$\dot{p} = v, \tag{20}$$
$$\dot{v} = a(p, v, q, u_f, u_m), \tag{21}$$
$$\dot{q} = \dot{q}(q, \omega), \tag{22}$$
$$\dot{\omega} = \alpha(p, v, q, \omega, u_f, u_m), \tag{23}$$

where $p \in \mathbb{R}^3$ is vehicle position, $v \in \mathbb{R}^3$ is velocity, $q \in \mathbb{R}^4$ is the attitude quaternion and $\omega \in \mathbb{R}^3$ is angular velocity. $u_f \in \mathbb{R}^1$ and $u_m \in \mathbb{R}^3$ are primary force and moment actuators respectively. We may now define the state vector and control vector

$$x \triangleq [p^T \;\; v^T \;\; q^T \;\; \omega^T]^T, \tag{24}$$
$$u \triangleq [u_f^T \;\; u_m^T]^T, \tag{25}$$

[1] For a detailed description of the singular perturbation analysis, see Section VI in [12].

It is assumed that psuedocontrol hedging as described in [7] copes with limitations to the control inputs such as actuator amplitude saturation, actuator rate saturation, and actuator dynamics so we will not include detailed actuator dynamics in this formulation. The approximated translational and rotational dynamics can be formulated from (20)-(23) as the following

$$a_{des} = \hat{a}(x, q_{des}, u_{f_{des}}) \tag{26}$$
$$\alpha_{des} = \hat{\alpha}(x, u_{m_{des}}) \tag{27}$$

where, a_{des} and α_{des} are commonly referred to as the pseudocontrol. $\hat{a}$ and $\hat{\alpha}$ represent the available approximation of the real rates obtained from the navigation system sensors. In addition, $u_{f_{des}}$, $u_{m_{des}}$ and q_{des} are the control inputs and attitude expected to achieve the desired pseudocontrol. When a_{des} and α_{des} are chosen such that they are invertible, the desired control and attitude can be written as

$$\begin{bmatrix} q_{des} \\ u_{f_{des}} \end{bmatrix} = \hat{a}^{-1}(x, a_{des}), \tag{28}$$
$$u_{m_{des}} = \hat{\alpha}^{-1}(x, \alpha_{des}), \tag{29}$$

Combining the inverse law in (21) with (28) and (23) with (29), the following closed loop translational and rotational dynamics result

$$\dot{v} = a_{des} + \bar{\Delta}_a(x,u) - a_h, \tag{30}$$
$$\dot{\omega} = \alpha_{des} + \bar{\Delta}_\alpha(x,u) - \alpha_h, \tag{31}$$

where,

$$\bar{\Delta} = \begin{bmatrix} \bar{\Delta}_a \\ \bar{\Delta}_\alpha \end{bmatrix} \triangleq \begin{bmatrix} a(x,u) - \hat{a}(x, q_{des}, u_{f_{des}}) \\ \alpha(x,u) - \hat{\alpha}(x, u_{m_{des}}) \end{bmatrix} \tag{32}$$

are the static model error due to imperfect model inversion and the pseudocontrol hedging signals a_h and α_h are given by

$$a_h = \hat{a}(x, q_{des}, u_{f_{des}}) - \hat{a}(x, \hat{u}_f) \tag{33}$$
$$\alpha = \hat{\alpha}(x, u_{m_{des}}) - \hat{\alpha}(x, \hat{u}_m) \tag{34}$$

where $\hat{u}_f$ and $\hat{u}_m$ either estimations or measurements of the real control outputs. The pseudocontrols are given by

$$a_{des} = a_c + a_{pd} - \bar{a}_{ad} \tag{35}$$
$$\alpha_{des} = \alpha_c + \alpha_{pd} - \bar{\alpha}_{ad} \tag{36}$$

where a_c and α_c are outputs of reference systems for the translational and attitude dynamics, respectively; a_{pd} and α_{pd} are outputs of proportional-derivative (PD) compensators; and finally, a_{ad} and α_{ad} are the outputs of an adaptive element designed to cancel model error $\bar{\Delta}$. This formulation uses a single hidden layer neural network to update the adaptive law [8].

The full reference system dynamics are given by

$$\dot{p}_r = v_r, \tag{37}$$

$$\dot{v}_r = a_c(p_r, v_r, p_c, v_c), \tag{38}$$

$$\dot{q}_r = \dot{q}_r(q_r, \omega_r), \tag{39}$$

$$\dot{\omega} = \alpha_c(q_r, v_r, q_c, q_c \oplus q_{des}, \omega_c), \tag{40}$$

where p_r and v_r are the outer-loop reference system states and q_r and ω_r are the inner-loop reference system states. The external command signal is $x_c = [p_c^T\, v_c^T\, q_c^T\, \omega_c^T]^T$. Note that the attitude desired by the outer loop is now added to the commands for the inner-loop controller. Here, $q_c \oplus q_{des}$ denotes quaternion addition.

Now, the error dynamics can be stated as

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} p - p_r \\ v - v_r \\ Q(q, q_r) \\ \omega - \omega_r \end{bmatrix} \tag{41}$$

where $Q(\cdot,\cdot) : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^3$ gives the three component error vector between quaternions. The output of the PD compensators can be written as

$$\begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = - \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e \tag{42}$$

where $R_p, R_d \in \mathbb{R}^{3\times 3}$ and $K_p, K_d \in \mathbb{R}^{3\times 3}$ are the linear gain, positive definite matrices determined from [8] as

$$R_p = \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2}, \quad R_d = \frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \tag{43}$$

$$K_p = \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2, \quad K_d = 2\zeta_i\omega_i + 2\zeta_o\omega_o \tag{44}$$

where the index i and o represent inner and outer loop respectively and ω and ζ are the frequency and damping ratio of the second order reference system dynamics.

The tracking error dynamics can be found directly by differentiating the error as

$$\dot{e} = \begin{bmatrix} v - v_r \\ \dot{v} - \dot{v}_r \\ \omega - \omega_r \\ \dot{\omega} - \dot{\omega}_r \end{bmatrix} \tag{45}$$

The overall tracking error dynamics can be expressed as

$$\dot{e} = A_r e + B_r[\nu_{ad} - \bar{\Delta}(x, u)] \tag{46}$$

where $\bar{\Delta}$ is given by (32) and the reference system is described by

$$A_r = \begin{bmatrix} 0 & I_3 & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I_3 \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, B_r = \begin{bmatrix} 0 & 0 \\ I_3 & 0 \\ 0 & 0 \\ 0 & I_3 \end{bmatrix}, v_{ad} = \begin{bmatrix} a_{ad} \\ \alpha_{ad} \end{bmatrix} \tag{47}$$

An approximate model for the attitude dynamics of the helicopter was generated by linearizing the nonlinear model around hover and neglecting coupling between the attitude dynamics

$$\alpha_{des} = \hat{\alpha} = \hat{A}_1 \omega_b + \hat{A}_2 v_b + B u_{m_{des}} \tag{48}$$

where $\hat{A}_1$ and $\hat{A}_2$ are linearized dynamics, and v_b and ω_b are body frame velocity and angular velocity. Choosing the control matrix B such that it is invertible, inverting (48), we are left with the following

$$u_{m_{des}} = B^{-1}(\alpha_{des} - \hat{A}_1 \omega_b - \hat{A}_2 v_b) \tag{49}$$

The translational dynamics are modeled as a point mass with a thrust vector that is a function of the plant orientation. A simple model can be used here as any unmodeled dynamics is compensated in the adaptive element. The simple relationships between thrust, attitude, and accelerations can be described as

$$a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{u_{coll}} \end{bmatrix} (u_{coll_{des}} - u_{coll_{trim}}) + L_{bv} g \tag{50}$$

where the subscript *coll* represents collective control input and $Z_{u_{coll}}$ is the control derivative for acceleration in the vertical axis. L_{bv} is the direction cosine matrix that transforms a vector from the vehicle (or local) frame to the body frame and g is an assumed gravity vector. The desired specific force along the body z axis may be evaluated as

$$f_{sf} = a_{des} - L_{bv} g \tag{51}$$

The required collective input can be evaluated as

$$u_{coll_{des}} = f_{sf} / Z_{U_{coll}} + u_{coll_{trim}} \tag{52}$$

The attitude augmentation required to orient the thrust vector to attain the desired translational accelerations are given by the following small-angle corrections from the current reference body attitude and attitude command:

$$\Delta\Phi_1 = a_{des_2} / f_{sf}, \quad , \Delta\Phi_2 = a_{des_1} / f_{sf}, \quad \Delta\Phi_1 = 0, \tag{53}$$

For this simplified model, heading change has no effect on accelerations in the x,y plane, and hence, $\Phi_3 = 0$. These three correction angles may now be used to generate the attitude quaternion correction desired by the outer loop, thus,

$$q_{des} = q(\Delta \Phi_1, \Delta \Phi_2, \Delta \Phi_3) \tag{54}$$

where $q(\cdot)$ is a function that expresses an Euler-angle-based rotation as a quaternion.

Now, with the framework for the implementation of the control scheme and the adaptive element described let us describe the formulation of the frequency limitation within that scheme. As the inner and outer loops are fully decoupled, applying the frequency-limited framework to the outerloop results in the following.

$$\dot{e}_L = \begin{bmatrix} 0 & I_3 \\ -R_p & -R_d \end{bmatrix} e_L + \eta \left(\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} - e_L \right) \tag{55}$$

This can be expanded to the following

$$\dot{e}_{L_1} = e_{L_2} + \eta(e_1 - e_{L_1}) \tag{56}$$

$$\dot{e}_{L_2} = -R_p e_{L_1} - R_d e_{L_2} + \eta(e_2 - e_{L_2}) \tag{57}$$

Inserting this result into (46) results in the following updated error dynamics

$$\dot{e} = A_r e + B_r[\nu_{ad} - \bar{\Delta}(x,u)] - \kappa \left(\begin{bmatrix} e_1 \\ e_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} e_{L_1} \\ e_{L_2} \\ 0 \\ 0 \end{bmatrix} \right) \tag{58}$$

5 Simulation and Flight Test Results

The following section shows the results of both Simulation and flight tests for translational position step commands to the GTQ quadrocopter using the GUST high fidelity flight simulator. For the tests with the frequency limited MRAC active, η was chosen as $5rad/s$ so as to pass the frequency range of the expected system dynamics of UAV while rejecting the high frequency content of the control signal. The κ term was determined by simple trail and observation.

First, let us consider the simulation results starting with Figure (3) which shows the results of standard MRAC with high gain adaptation, the outerloop learning rate was set as $\Gamma_{ol} = 20$. The figure shows oscillation in the position tracking, large amplitude oscillations in the adaptive control signal and large outputs of the actuators, with the actuators often hitting their magnitude limits. This flying performance would not be suitable for a real flight.

Figure (4) shows the effect of the frequency-limited architecture. The test was completed with the same adaptive gain as above but with the additional settings of $\kappa = 20, \eta = 5$. It can be seen that the large oscillation in the position tracking is reduced, the tracking error itself is reduced and the actuator output is now within an acceptable magnitude and frequency range.

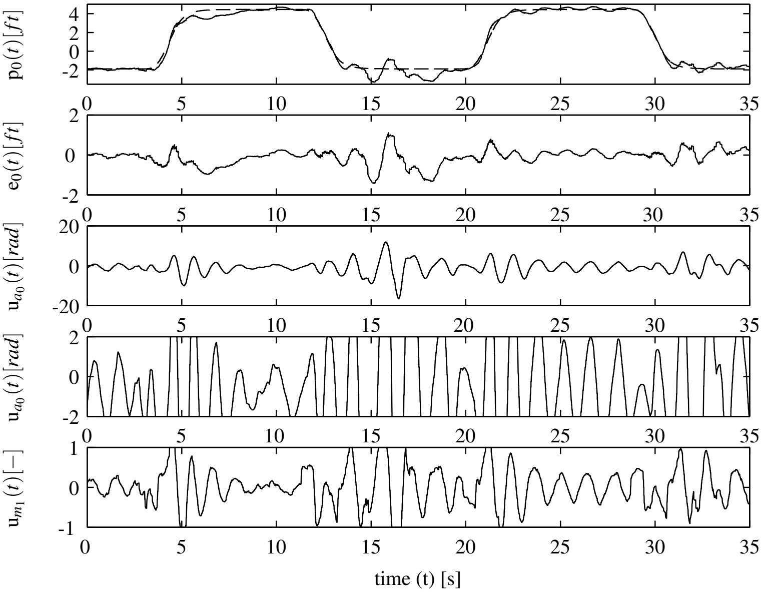


Fig. 3 Simulation test results for translational commands using standard MRAC with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 0, \eta = 0$). (Dashed line is commanded position). Plots 3 and 4 are identical but are plotted with differing scales.

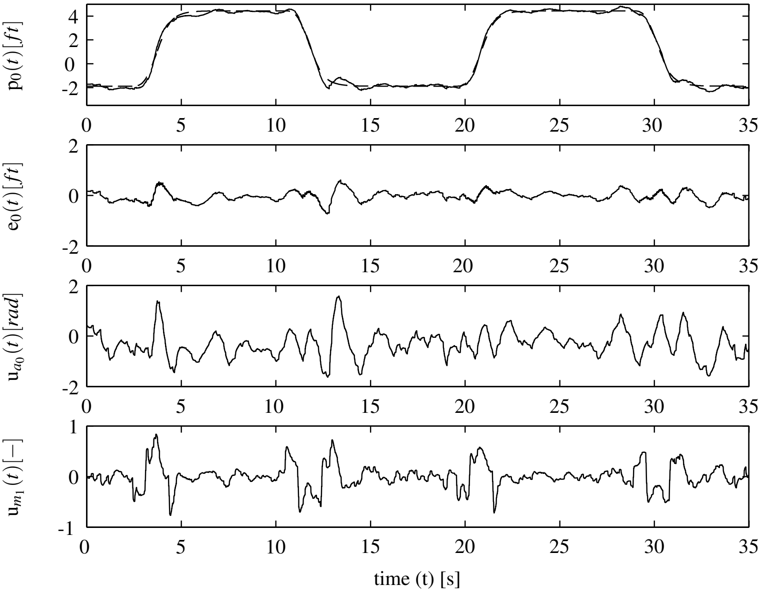


Fig. 4 Simulation test results for translational commands using the frequency-limited adaptive control architecture with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 20, \eta = 5$). (Dashed line is commanded position).

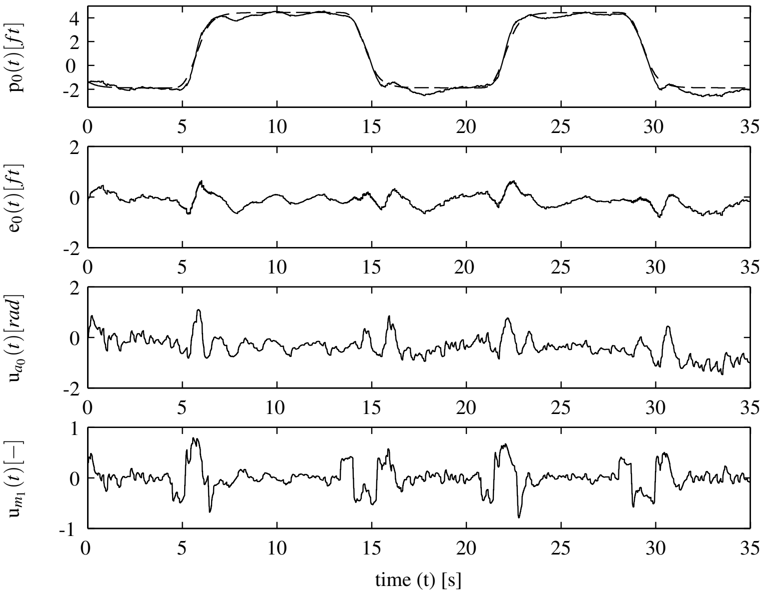


Fig. 5 Simulation test results for translational commands using approach from [9] with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 20, \eta = 0$). (Dashed line is commanded position).

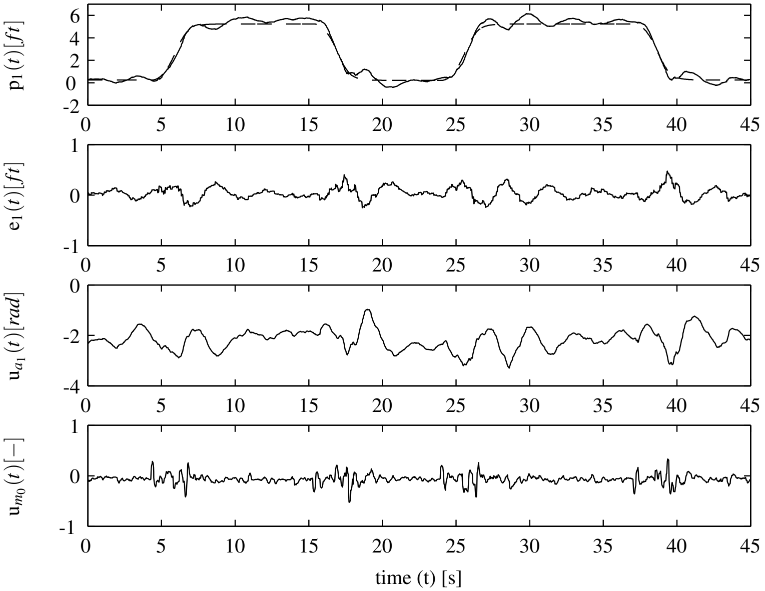


Fig. 6 Flight test results for translational commands using the frequency-limited adaptive control architecture with the following settings ($\Gamma_{ol} = 3 \cdot I_3, \kappa = 6, \eta = 5$). (Dashed line is commanded position).

Now, compare the performance of the frequency-limited architecture to the closed loop reference architecture suggested in [9, 5] as shown in Figure (5). Notice that the adaptive signal appears to have more high frequency content, this results in a higher frequency content in the actuator output. This observation can be explained as possible noise amplification in the adaptive signal due to the raw error feedback in the reference system.

These simulations show that the frequency-limited architecture improves the performance of high adaptive gain maneuverer by reducing the high frequency oscillations while ensuring the actuator output signal is within practicable limits.

The flight results are shown in Figures (6) and (7). Figure (6) shows the performance of the frequency-limited architecture with high gain learning rate of $\Gamma_{ol} = 3$. At this learning rate, without the augmented architecture activated, the platform would become unstable due to the influence of computational delays and noisy sensor readings. The performance of the frequency-limited architecture can be compared to the nominal case of standard MRAC with a learning rate of $\Gamma_{ol} = 1.5$ in Figure (7). Notice that the tracking error observed with the nominal case are approximately a factor of two higher than that of the frequency-limited adaptive control, this highlights the improvement in tracking performance using the new architecture.

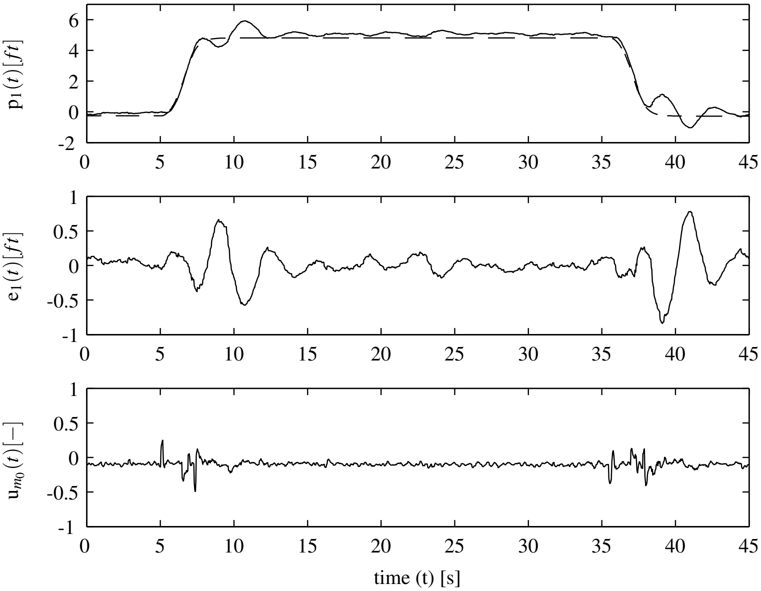


Fig. 7 Flight test results for translational commands using standard MRAC with the following settings ($\Gamma_{ol} = 1.5 \cdot I_3, \kappa = 0, \eta = 0$). (Dashed line is commanded position).

6 Conclusion

In this paper, we presented an application of a frequency-limited adaptive control framework to a high-fidelity autonomous quadrocopter model. We first presented the framework of frequency-limited adaptive control. Next we presented the implementation of the controller on the outerloop of an existing model inversion adaptive controller for an autonomous quadrocopter. Finally we compared standard and frequency-limited controller simulation results during translation maneuver in simulations and show the performance of the controller in real world flight tests. All results showed that the frequency-limited controller improved tracking as compared to the standard model reference adaptive control and kept control outputs within reasonable limits.

References

1. Annaswamy, A.M., Narendra, K.S.: Stable adaptive systems (1989)
2. Ascending Technologies: Astec Pelican
3. Chowdhary, G., Sobers Jr., D.M., Pravitra, C., Christmann, C., Wu, A., Hashimoto, H., Ong, C., Kalghatgi, R., Johnson, E.N.: Self-contained ranging sensor aided autonomous guidance, navigation, and control for indoor flight. AIAA Journal of Guidance Control and Dynamics (2011)
4. Dydek, Z.T., Annaswamy, A.M., Lavretsky, E.: Adaptive control and the NASA X-15-3 flight revisited. IEEE Control Systems Magazine 30, 32–48 (2010)
5. Gibson, T.E., Annaswamy, A.M., Lavretsky, E.: Adaptive systems with closed-loop reference models: Stability, robustness, and transient performance. IEEE Transactions on Automatic Control (2012) (submitted)
6. Haddad, W.M., Chellaboina, V.: Nonlinear dynamical systems and control. a lyapunov-based approach (2008)
7. Johnson, E.N., Calise, A.J.: Limited authority adaptive flight control for reusable launch vehicles. AIAA Journal of Guidance, Control, and Dynamics 26, 906–913 (2003)
8. Johnson, E.N., Kannan, S.K.: Adaptive trajectory control for autonomous helicopters. AIAA Journal of Guidance, Control, and Dynamics 28, 524–538 (2005)
9. Lavretsky, E.: Reference dynamics modification in adaptive controllers for improved transient performance. In: AIAA Guidance, Navigation and Control Conference (2011)
10. Rohrs, C.E., Valavani, L.S., Athans, M., Stein, G.: Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. IEEE Transactions on Automatic Control 30, 881–889 (1985)
11. Yucelen, T., Haddad, W.M.: A robust adaptive control architecture for disturbance rejection and uncertainty suppression with l_∞ transient and steady-state performance guarantees. International Journal of Adaptive Control and Signal Processing (2012), doi:10.1002/acs.2281
12. Yucelen, T., De La Torre, G., Johnson, E.N.: Frequency-limited adaptive control architechture for transient response improvement. In: American Control Conference (2012) (to appear)

Aeroservoelastic Investigations of a High-Aspect-Ratio Motor Glider

Flávio J. Silvestre

Abstract. This paper presents aeroservoelastic investigations for the STEMME S15 prototype using a methodology of modelling the flexible aircraft dynamics in the time domain. The effects of the flexibility in the closed-loop stability according to the sensor positioning are discussed, for a pitch and a yaw damper. The modelling of the flexible dynamics is based on the mean axes approximation, without considering the inertial coupling between the rigid-body and the elastic degrees of freedom. The structural dynamics is linearly represented in modal coordinates. To determine the incremental aerodynamics due to elastic deformations, an unsteady strip theory formulation in the time domain is used, considering the exponential representation of the Wagner function and the resulting stripwise aerodynamic lag states. Spanwise correction to account for three-dimensional effects at the wing tip based on the quasi-steady circulation distribution was applied. The validation of the open-loop flexible aircraft simulations with flight test results are also presented.

1 Introduction

The use of aircraft control associated with aeroelasticity, or the inclusion of aeroelastic effects on the aircraft control design, gave rise to a new terminology, frequently used in the last years in aeronautical engineering: Aero**servo**elasticity. In the editorial of a special issue of the AIAA Journal of Aircraft about the Active Flexible Wing Program [5], Noll and Eastep [4] outlined the increasing importance of aeroelasticity and aeroservoelasticity in the vehicle conceptual and preliminary design process. Nowadays aeroservoelastic stability analyses are required for clearance of flight control laws either if the control laws are designed for influencing aeroelastic

Flávio J. Silvestre
Instituto Tecnológico de Aeronáutica, Divisão de Engenharia Aeronáutica,
Praça Marechal Eduardo Gomes, 50 - Vila das Acácias CEP: 12.228-900,
São José dos Campos, Brazil
e-mail: flaviojs@ita.br

behaviour, such as flutter suppression functions, or if they are designed for manual or automatic flight control of the aircraft. Complex models of the flexible aircraft dynamics that integrate CFD (computational fluid dynamics) and FEM (finite element method) codes have been developed for open-loop and closed-loop simulation and analysis, but they require very much computing time. Moreover, the complexity of CFD/FEM models still hampers their usage in flight control system design [13].

Even in major aircraft industries, integrated aeroelastic models are still only applied for flight control validation, and not in the flight control system development. A schematic flow chart representing the industrial flight control law design process is shown in ref. [10], based on Fielding and Luckner [7] considerations, indicating that aeroservoelastic analyses are carried out only after the off-line design. The costs associated with these methods are also very high, inhibiting their application by small aircraft industries. According to Krüger [9], it is (also) necessary to offer faster and cheaper processes for the stability analysis of small, sport class of aircraft.

In this scenario, simpler but still reliable models for aeroservoelastic applications are desired. To fulfil these demands, a novel methodology for modelling the dynamics of slightly flexible, high-aspect-ratio aircraft in the time domain has been developed by Silvestre [16]. The objective of this paper is to demonstrate how the effects of the flexibility can be taken into account in flight control law design using this methodology. For this, the examples of a yaw damper and a pitch damper for the STEMME S15 are considered, and the effects of the flexibility according to the angular rates sensor position are demonstrated. In section 2 the methodology is presented; section 3 details the S15; subsequently the validation of the methodology is commented on in section 4; the aeroservoelastic investigations are discussed in section 5, and finally the most important points are summarised in section 6 completing the scope of this paper.

2 The Flexible Dynamics Model

Within the project LAPAZ a non-linear high-fidelity flight simulation model for the rigid aircraft, the motor glider STEMME S15, was built up in MATLAB® Simulink environment at TUB, as described by Meyer-Brügel in [15]. This complete and complex model involves the six-degrees-of-freedom equations of motion with quaternion based attitude calculation, considering the earth's rotation and defining the earth as an ellipsoid. Not only the body flight dynamics was modelled, but also the dynamics of various systems composing the aircraft, the power plant dynamics, the dynamics of all actuators for all control surfaces and for the engine, the landing gear and the nose wheel actuation, and the dynamics of all sensors installed on the aircraft that are used for generating inputs to the flight control system. The model is also equipped with a terrain model, necessary to validate the flight control laws at low level flight. The aircraft is considered as a rigid body.

The calculation of “rigid-body”[1] aerodynamic forces and moments is based on parameter identification, assuming the aerodynamic force and moment coefficients to be represented by parametric polynomial equations. The equations have a maximal polynomial rank of three and contain cross coupling terms up to second order. The parameters have been estimated from flight test results utilizing parameter identification methods. For the concerned aircraft class an extensive flight test campaign was conducted, involving 16 different manoeuvre types which have been realised in 16 different reference flight conditions. Overall, approximately 40 GB of data was generated, analysed and evaluated by Meyer-Brügel [15]. The most important variables of the identified non-linear aerodynamic model are the angle of attack, sideslip angle, angular rates, aerodynamic control surfaces deflections and motor rotation. The model takes also into account ground effect and stall.

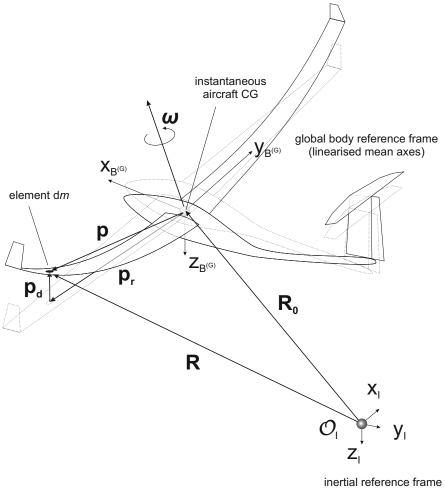


Fig. 1 Flying flexible aircraft, and the inertial and mean axes reference frames

To account for the aeroelasticity, a methodology was developed by Silvestre [16] to extend the rigid-body simulation model. This methodology is based on the equations of motion for slightly flexible aircraft, derived at the reference system of the so called mean axes, as explained in ref. [3]. The mean axes are defined as a floating reference frame where the linear and angular momenta caused by elastic deformations vanish [2]. For slightly flexible aircraft, small structural displacements can be assumed, and linearised mean axes constraints can be applied instead. Using the modal approach to model the structural dynamics, the linearised mean axes are found to lay at the centre of gravity (CG) of the undeformed structure, as in Fig. 1.

Applying Lagrangian mechanics, the use of the modal approach and the practical mean axes constraints lead to the final equations of motion of the flexible aircraft.

[1] “Rigid-body” aerodynamics refers to the aerodynamic forces and moments resulting from global change of the aircraft state, regardless the changes caused by the flexible displacements.

These equations are formally the rigid-body equations of motion, supplemented by the differential equations of the structural dynamics in modal coordinates (the modal amplitudes). The hypothesis of small elastic displacements results in no inertial coupling among the flight dynamics and the structural dynamics. The only coupling is of an aerodynamic nature.

The unsteady incremental aerodynamic forces and moments are modelled using the incompressible potential theory. Jones' exponential approximation of the Wagner function [1] is used to model the circulatory contribution in the time domain. The strip theory is applied, and the quasi-steady circulation distribution is used to account for three-dimensional effects. Working with n_e elastic modes in the model, which may be determined in a ground vibration test or calculated using a finite-element model of the aircraft structure, each elastic mode is linearised at the elastic axis (EA) of the lifting surface, as shown in Fig. 2.

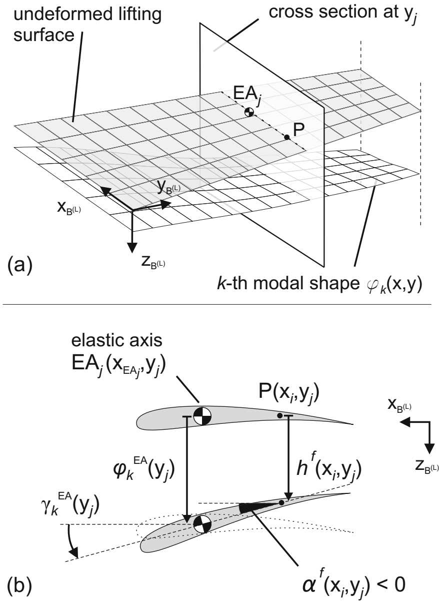


Fig. 2 Approximation of the elastic modal shapes at the elastic axis by equivalent displacement and torsion: (a) representation of the k-th modal shape on the LBRF; (b) cross section y_j

Considering both the non-circulatory (NC) and the circulatory (C) contributions, the incremental aerodynamic forces $\mathbf{F}_{\mathrm{A}}|_{\mathrm{B}^{(\mathrm{G})}}$ and moments $\mathbf{M}_{\mathrm{A}}|_{\mathrm{B}^{(\mathrm{G})}}$ can be written as:

$$\begin{aligned}\mathbf{F}_{\mathrm{A}}|_{\mathrm{B}^{(\mathrm{G})}}(t) = {} & \rho(t)\mathbf{F}_{\mathrm{A}\ddot{\boldsymbol{\eta}}}(t)\ddot{\boldsymbol{\eta}}(t) \\ & +\rho(t)V(t)\mathbf{F}_{\mathrm{A}\dot{\boldsymbol{\eta}}}(t)\dot{\boldsymbol{\eta}}(t) \\ & +\rho(t)V^2(t)\mathbf{F}_{\mathrm{A}\boldsymbol{\eta}}(t)\boldsymbol{\eta}(t) \\ & +\rho(t)V(t)\mathbf{F}_{\mathrm{A}\boldsymbol{\lambda}}(t)\left(\boldsymbol{\lambda}_1(t)+\boldsymbol{\lambda}_2(t)\right), \qquad (1)\end{aligned}$$

$$\begin{aligned}\mathbf{M}_{\mathrm{A}}\big|_{\mathrm{B(G)}}(t) = {} & \rho(t)\mathbf{M}_{\mathrm{A}\ddot{\boldsymbol{\eta}}}(t)\ddot{\boldsymbol{\eta}}(t)\\ & +\rho(t)V(t)\mathbf{M}_{\mathrm{A}\dot{\boldsymbol{\eta}}}(t)\dot{\boldsymbol{\eta}}(t)\\ & +\rho(t)V^2(t)\mathbf{M}_{\mathrm{A}\boldsymbol{\eta}}(t)\boldsymbol{\eta}(t)\\ & +\rho(t)V(t)\mathbf{M}_{\mathrm{A}\boldsymbol{\lambda}}(t)\left(\boldsymbol{\lambda}_1(t)+\boldsymbol{\lambda}_2(t)\right).\end{aligned} \tag{2}$$

The generalised force at the *k*-th elastic mode can be similarly put into the form:

$$\begin{aligned}Q_{\eta_k}(t) = {} & \rho(t)\mathbf{Q}_{\eta_{k\ddot{\boldsymbol{\eta}}}}(t)\ddot{\boldsymbol{\eta}}(t)\\ & +\rho(t)V(t)\mathbf{Q}_{\eta_{k\dot{\boldsymbol{\eta}}}}(t)\dot{\boldsymbol{\eta}}(t)\\ & +\rho(t)V^2(t)\mathbf{Q}_{\eta_{k\boldsymbol{\eta}}}(t)\boldsymbol{\eta}(t)\\ & +\rho(t)V(t)\mathbf{Q}_{\eta_{k\boldsymbol{\lambda}}}(t)\left(\boldsymbol{\lambda}_{\mathbf{1}}(t)+\boldsymbol{\lambda}_{\mathbf{2}}(t)\right).\end{aligned} \tag{3}$$

wherein ρ is the air density, V is the true airspeed, $\boldsymbol{\eta}$ is the vector of the modal amplitudes, $\boldsymbol{\lambda}_{\mathbf{1}}$ and $\boldsymbol{\lambda}_{\mathbf{2}}$ are the vectors of the stripwise first and second lag states, according to the exponential approximation of the Wagner function. The twelve matrix coefficients in Equations (1), (2) and (3) are determined in ref. [16]

The equations of motion of the flexible aircraft are summarised in the following. The matrices used in Equations from (5) to (7) are defined according to Tab. 1.

- Equations of the flight dynamics:

$$\begin{aligned}\dot{\mathbf{V}}\big|_{\mathrm{B(G)}}(t) &= -\,\boldsymbol{\omega}\big|_{\mathrm{B(G)}}(t)\times\mathbf{V}\big|_{\mathrm{B(G)}}(t)+\mathbf{T}_{\mathrm{B(G)I}}(t)\,\mathbf{G}\big|_{\mathrm{I}}+\tfrac{1}{m}\,\mathbf{F}^{\mathrm{ext}}\big|_{\mathrm{B(G)}}(t)\\ \dot{\boldsymbol{\omega}}\big|_{\mathrm{B(G)}}(t) &= -\mathbf{J}^{-1}\left(\boldsymbol{\omega}\big|_{\mathrm{B(G)}}(t)\times\left(\mathbf{J}\,\boldsymbol{\omega}\big|_{\mathrm{B(G)}}(t)\right)\right)+\mathbf{J}^{-1}\,\mathbf{M}^{\mathrm{ext}}\big|_{\mathrm{B(G)}}(t),\end{aligned} \tag{4}$$

wherein $\mathbf{F}^{\mathrm{ext}}\big|_{\mathrm{B(G)}}$ and $\mathbf{M}^{\mathrm{ext}}\big|_{\mathrm{B(G)}}$ are the external (propulsion and aerodynamic) forces and moments acting over the aircraft, including the forces and moments caused by the aircraft elasticity; $\mathbf{G}\big|_{\mathrm{I}}$ is the gravity acceleration vector($\left[0\ 0\ g\right]^T$), m is the aircraft mass and $\mathbf{J}$ is the aircraft inertia matrix; $\mathbf{T}_{\mathrm{B(G)I}}$ is the transformation matrix from the inertial to the global body reference frame; $\mathbf{V}\big|_{\mathrm{B(G)}}$ and $\boldsymbol{\omega}\big|_{\mathrm{B(G)}}$ are respectively the aircraft linear and angular velocities relative to the inertial reference system but written on the global body reference frame.

- Equations of the structural dynamics:

$$\begin{aligned}\begin{bmatrix}\dot{\boldsymbol{\eta}}(t)\\ \ddot{\boldsymbol{\eta}}(t)\end{bmatrix} &= \begin{bmatrix}\mathbf{0}_{n_e\times n_e} & \mathbf{I}_{n_e\times n_e}\\ \boldsymbol{\Pi}_1(t) & \boldsymbol{\Pi}_2(t)\end{bmatrix}\begin{bmatrix}\boldsymbol{\eta}(t)\\ \dot{\boldsymbol{\eta}}(t)\end{bmatrix}+\begin{bmatrix}\mathbf{0}_{n_e\times n_e}\\ \boldsymbol{\Pi}_3(t)\end{bmatrix}\left(\boldsymbol{\lambda}_1(t)+\boldsymbol{\lambda}_2(t)\right)\\ &\quad+\begin{bmatrix}\mathbf{0}_{n_e\times n_e}\\ \boldsymbol{\Pi}_4(t)\end{bmatrix}\mathbf{X}_{\mathrm{state}}(t)+\begin{bmatrix}\mathbf{0}_{n_e\times n_e}\\ \boldsymbol{\Pi}_5(t)\end{bmatrix}\mathbf{u}_{\mathrm{control}}(t).\end{aligned} \tag{5}$$

The matrices $\boldsymbol{\Pi}_1$, $\boldsymbol{\Pi}_2$, $\boldsymbol{\Pi}_3$, $\boldsymbol{\Pi}_4$ and $\boldsymbol{\Pi}_5$ are defined as:

$$\begin{aligned}
\boldsymbol{\Pi}_1(t) &= \left[\boldsymbol{\mu} - \rho(t)\mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t)\right]^{-1}\left[-\boldsymbol{\mu}\boldsymbol{\omega}_n^2 + \rho(t)V^2(t)\left(\mathbf{Q}_{\boldsymbol{\eta}\boldsymbol{\eta}}^{(\mathrm{NC})}(t) + \mathbf{Q}_{\boldsymbol{\eta}\boldsymbol{\eta}}^{(\mathrm{C})}(t)\right)\right]\\
\boldsymbol{\Pi}_2(t) &= \left[\boldsymbol{\mu} - \rho(t)\mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t)\right]^{-1}\left[-2\boldsymbol{\mu}\boldsymbol{\xi}\boldsymbol{\omega}_n + \rho(t)V(t)\left(\mathbf{Q}_{\boldsymbol{\eta}\dot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t) + \mathbf{Q}_{\boldsymbol{\eta}\dot{\boldsymbol{\eta}}}^{(\mathrm{C})}(t)\right)\right]\\
\boldsymbol{\Pi}_3(t) &= \left[\boldsymbol{\mu} - \rho(t)\mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t)\right]^{-1}\rho(t)V(t)\mathbf{Q}_{\boldsymbol{\eta}\boldsymbol{\lambda}}^{(\mathrm{NC})}(t)\\
\boldsymbol{\Pi}_4(t) &= \left[\boldsymbol{\mu} - \rho(t)\mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t)\right]^{-1}\frac{1}{2}\rho(t)V^2(t)S\bar{c}\mathbf{Q}_{\boldsymbol{\eta}\mathbf{X}_{\mathrm{state}}}^{(\mathrm{QS})}(t)\\
\boldsymbol{\Pi}_5(t) &= \left[\boldsymbol{\mu} - \rho(t)\mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(\mathrm{NC})}(t)\right]^{-1}\frac{1}{2}\rho(t)V^2(t)S\bar{c}\mathbf{Q}_{\boldsymbol{\eta}\mathbf{u}_{\mathrm{control}}}^{(\mathrm{QS})}(t).
\end{aligned}\tag{6}$$

In the latter Equations (5) and (6), $\mathbf{I}_{n_e\times n_e}$ and $\mathbf{0}_{n_e\times n_e}$ are respectively the identity and the null matrices of order n_e; $\mathbf{Q}_{\boldsymbol{\eta}\mathbf{X}_{\mathrm{state}}}^{(\mathrm{QS})}$ and $\mathbf{Q}_{\boldsymbol{\eta}\mathbf{u}_{\mathrm{control}}}^{(\mathrm{QS})}$ are respectively the contribution of the aircraft state and the deflection of the control surfaces to the elastic generalised loads, which were assumed to be quasi-stationary in this work. They are calculated with quasi-steady aerodynamics and the strip theory, as in ref. [11]. The elements of the state vector $\mathbf{X}_{\mathrm{state}}$ are the aircraft angle of attack α, sideslip angle β, and the non-dimensional (divided by the factor $V/\left(\frac{\bar{c}}{2}\right)$) angular rates $\bar{p}$, $\bar{q}$, and $\bar{r}$; the elements of the control vector $\mathbf{u}_{\mathrm{control}}$ are the deflection of the control surfaces: aileron, elevator, flaps, and rudder. Refer to ref. [11] for the expressions of the structural stability derivatives in $\mathbf{Q}_{\boldsymbol{\eta}\mathbf{X}_{\mathrm{state}}}^{(\mathrm{QS})}$ and $\mathbf{Q}_{\boldsymbol{\eta}\mathbf{u}_{\mathrm{control}}}^{(\mathrm{QS})}$.

- Equations of the aerodynamic lag states:

$$\begin{aligned}
\dot{\boldsymbol{\lambda}}_1(t) &= 2V(t)a_1\boldsymbol{c}^{-1}\boldsymbol{\lambda}_1(t) + A_1\dot{\mathbf{w}}_{3/4}^f(t)\\
\dot{\boldsymbol{\lambda}}_2(t) &= 2V(t)a_2\boldsymbol{c}^{-1}\boldsymbol{\lambda}_2(t) + A_2\dot{\mathbf{w}}_{3/4}^f(t),
\end{aligned}\tag{7}$$

wherein $\dot{\mathbf{w}}_{3/4}^f(t)$ is the vector collecting the stripwise accelerations at the three-quarter-chord, which can be determined with the time derivative of the spanwise downwash at the three-quarter-chord, calculated in Eq. (8). It is a function of the modal amplitudes, velocities and accelerations, as well as the aircraft velocity:

$$\begin{aligned}
w_{3/4}^f(y_j,t) &= \boldsymbol{\varphi}^{\mathrm{EA}}(y_j)\dot{\boldsymbol{\eta}}(t) - V(t)\boldsymbol{\gamma}^{\mathrm{EA}}(y_j)\boldsymbol{\eta}(t) + \left(x_{3/4_j} - x_{\mathrm{EA}_j}\right)\boldsymbol{\gamma}^{\mathrm{EA}}(y_j)\dot{\boldsymbol{\eta}}(t)\\
&= \left[\boldsymbol{\varphi}^{\mathrm{EA}}(y_j) + \left(x_{3/4_j} - x_{\mathrm{EA}_j}\right)\boldsymbol{\gamma}^{\mathrm{EA}}(y_j)\right]\dot{\boldsymbol{\eta}}(t) - V(t)\boldsymbol{\gamma}^{\mathrm{EA}}(y_j)\boldsymbol{\eta}(t),
\end{aligned}\tag{8}$$

wherein $\boldsymbol{\varphi}^{\mathrm{EA}}$ and $\boldsymbol{\gamma}^{\mathrm{EA}}$ are respectively the vectors equivalent displacement and torsion at the lifting surface's elastic axis of all considered elastic modes.

Together with the equations of the kinematics of flight, Equations (4), (5) and (7) are already in the form to be programmed for flight simulations of the flexible aircraft. Next section shows the integration of the aeroelastic model to the rigid-body flight simulation model of S15 prototype.

Table 1 Matrix definitions in the aeroelastic model

Matrix	Definition	Dimension
$\boldsymbol{\mu}$	diagonal matrix, containing the modal mass μ_k in the main diagonal (modal mass matrix)	$n_e \times n_e$
$\boldsymbol{\xi}$	diagonal matrix, containing the structural damping factors ξ_k in the main diagonal	$n_e \times n_e$
$\boldsymbol{\omega}_n$	diagonal matrix, containing the natural frequencies ω_{nk} in the main diagonal	$n_e \times n_e$
$\mathbf{0}_{n_e \times n_e}$	null matrix	$n_e \times n_e$
$\mathbf{I}_{n_e \times n_e}$	identity matrix	$n_e \times n_e$
$\mathbf{Q}_{\boldsymbol{\eta}\boldsymbol{\eta}}^{(NC)}, \mathbf{Q}_{\boldsymbol{\eta}\dot{\boldsymbol{\eta}}}^{(NC)}, \mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(NC)}$	diagonal matrices, containing respectively the line vectors $\mathbf{Q}_{\eta_k\boldsymbol{\eta}}^{(NC)}, \mathbf{Q}_{\eta_k\dot{\boldsymbol{\eta}}}^{(NC)}, \mathbf{Q}_{\eta_k\ddot{\boldsymbol{\eta}}}^{(NC)}$	$n_e \times n_e$
$\mathbf{Q}_{\boldsymbol{\eta}\dot{\boldsymbol{\eta}}}^{(C)}, \mathbf{Q}_{\boldsymbol{\eta}\ddot{\boldsymbol{\eta}}}^{(C)}$	diagonal matrices, containing respectively the line vectors $\mathbf{Q}_{\eta_k\dot{\boldsymbol{\eta}}}^{(C)}, \mathbf{Q}_{\eta_k\ddot{\boldsymbol{\eta}}}^{(C)}$	$n_e \times n_e$
$\mathbf{Q}_{\boldsymbol{\eta}\boldsymbol{\lambda}}^{(C)}$	diagonal matrix, containing the line vectors $\mathbf{Q}_{\eta_k\boldsymbol{\lambda}}^{(C)}$	$n_e \times n_s$
$\mathbf{Q}_{\boldsymbol{\eta}\mathbf{X}_{\text{state}}}^{(QS)}$	quasi-stationary contribution of aircraft states to the elastic generalised loads, considering $\mathbf{X}_{\text{state}} = \left[1\ \alpha\ \beta\ \bar{p}\ \bar{q}\ \bar{r}\right]^T$ (see ref. [11])	$n_e \times 6$
$\mathbf{Q}_{\boldsymbol{\eta}\mathbf{u}_{\text{control}}}^{(QS)}$	quasi-stationary contribution of the control surfaces to the elastic generalised loads, considering aileron, flaps, elevator and rudder (see ref. [11])	$n_e \times 4$
$\boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2, \boldsymbol{\Pi}_3, \boldsymbol{\Pi}_4, \boldsymbol{\Pi}_5$	see Eq.(6)	$n_e \times n_e$
$\mathbf{c}$	diagonal matrix, containing the chords of all strips, c_j, in the main diagonal	$n_s \times n_s$
$\dot{\mathbf{w}}_{3/4}^{f}$	column vector which collects the downwash accelerations at the three-quarter-chord position of all strips	$n_s \times 1$

3 Aircraft Description

The modelling methodology of ref. [16], summarised in the last section, is applied to the STEMME S15 "Demonstrator", a prototype of the commercial motor gliders STEMME S6 and S15[2] from the German aircraft manufacturer STEMME AG[3]. Having a higher wing stiffness than the production aircraft, the S15 prototype is an experimental aircraft platform that is used in the project LAPAZ (abbreviation for ***L**uft-**A**rbeits-**P**lattform für die **A**llgemeine **Z**ivilluftfahrt* in German, translated as aerial work platform for the general civil aviation), a project financially

[2] S6 is the CS-22 certified sport version of the aircraft; S15 is the CS-23 certified utility aircraft with increased MTOW and hard points under the wings that allow mounting pods.

[3] Further information about the aircraft manufacturer at:
`http://www.stemme.de`

supported by the German National Aeronautics Research Programme (LuFo IV, abbreviation for ***Lu**ftfahrt**fo**rschungsprogramm* IV). In its first phase, the project intended to develop and demonstrate a reliable, high-precision automatic flight control system for the S15 to support measurements in flight for geo-exploration and surveillance tasks. The partners are the German aircraft manufacturer STEMME, the Universität Stuttgart and the Technische Universität Berlin (TUB), TUB being responsible for the flight simulation models as well as the flight control law design for the automatic flight control system. The first project phase, LAPAZ I[4], ended in October 2010 with flight tests, which proved that the flight control system had been successfully implemented. In a second phase, LAPAZ II[5], inaugurated in October 2010, automatic take-off and landing as well as gust load alleviation functions will be demonstrated. The landing function having already been successfully demonstrated by the end of March 2012, see ref. [17].

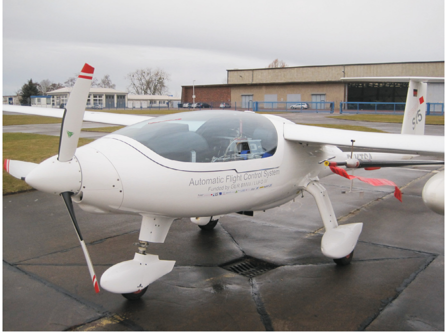


Fig. 3 The STEMME S15 prototype used in the LAPAZ project at Strausberg Airport.

The S15 is a utility aircraft, for which an optionally piloted version is being developed. It is equipped with a Bombardier-Rotax motor 914S, tricycle landing gear, and has two seats. The wing span is 18m and the horizontal empennage is in T-tail configuration. Figure 3 shows the aircraft at the aerodrome of Strausberg, Germany. The main properties are listed in ref. [17], and a summary is presented in Tab. 2 for the aircraft empty configuration. The inertia moments were determined experimentally using a pendulum and techniques of system identification by Rudenko [14]. The inertia products I_{xy}, I_{xz} and I_{yz} were not estimated and thus considered null in a first approximation. The position of the centre of gravity in Tab. 2 was estimated as well in ref. [14], with the exception of the lateral position, approximated by zero assuming symmetry relative to the xz-plane.

[4] Further information at TU Berlin's website of the project (February 2nd, 2012): `http://www.fmra.tu-berlin.de/menue/forschung/projekte/flugregelung/lapaz_i/`

[5] Further information at TU Berlin's website of the project (March 8th, 2012): `http://www.fmra.tu-berlin.de/menue/forschung/projekte/flugregelung/lapaz_ii/`

Table 2 General properties of S15 "Demonstrator" in empty configuration (without pilot, co-pilot, and fuel)

Property	Symbol	Value Unit
reference area	S	17.40 m^2
wing's mean aerodynamic chord	$\bar{c}$	1.0064 m
wing's span	$\bar{b}$	18.0 m
mass (empty)	m	750 kg
inertia moment, x	I_{xx}	5262 $kg.m^2$
inertia moment, y	I_{yy}	1913 $kg.m^2$
inertia moment, z	I_{zz}	7015 $kg.m^2$
position of the CG, x	x_{CG}	-2.8070 m
position of the CG, y	x_{CG}	0 m
position of the CG, z	x_{CG}	-0.2640 m

The S15 demonstrator has a stall speed at MTOW of approximately 90.0km/h (25.0m/s), and a maximum speed in level flight with maximum continuous power of approximately 280.0km/h (77.8m/s).

As discussed in section 2, the structural dynamics in the aeroelastic model are represented using the modal approach. The Institute of Reliability (*Institut für Zuverlässigkeitstechnik*) of the Technical University Hamburg-Harburg (TUHH) carried out a GVT (Ground Vibration Test) campaign with the S15 in 2008 for flutter calculations [12]. This work uses the available results. For the aeroelastic model, the modal eigenvectors up to 30Hz were considered, which are given in Tab. 3. This is assumed to be sufficient for the intended investigations that address aeroservoelastic effects of the controlled aircraft. Table 3 refers to the fixed-control-surface GVT results[6].

Figure 4 shows the linearised modal shape of the first symmetric wing torsion together with the points obtained at the GVT, where the accelerations were measured. Observe that just the modal shape's component perpendicular to the lifting surface, thus the only component considered in the incremental aerodynamic model, is displayed. The linearised modal shape fits very well the points determined in the GVT. Corresponding equivalent elastic displacements and torsions can be seen spanwise in Fig. 5.

With the spanwise aerodynamic properties, the strip geometry, and spanwise equivalent flexible displacements and torsions about the elastic axis, an aeroelastic database can be created. The aeroelastic database enables the determination of the matrices of unsteady coefficients, given in section 2.

[6] The characterisation of the elastic modes in symmetric and anti-symmetric modes is merely an approximation.

Table 3 First 16 elastic modes of S15 (in order of modal frequency) determined at the GVT test campaign at the TUHH [12], in empty configuration and with fixed control surfaces, which are considered in the aeroelastic model

#	Definition	Frequency ω_n (Hz)	Structural modal damping ξ (%)	Symmetry
1	1^{st} bending, wings	3.29	0.55	symmetric
2	1^{st} swing, wings and body	4.24	1.24	anti-symmetric
3	1^{st} bending, wings	7.34	1.00	anti-symmetric
4	1^{st} swing, wings	7.39	1.02	symmetric
5	1^{st} torsion, body	8.07	1.75	anti-symmetric
6	1^{st} bending, body	9.02	0.84	symmetric
7	2^{nd} bending, wings	11.55	0.72	symmetric
8	1^{st} swing, horizontal fin	11.72	1.34	anti-symmetric
9	1^{st} bending, horizontal fin	13.55	0.86	anti-symmetric
10	2^{nd} bending, wings	15.37	1.09	anti-symmetric
11	2^{nd} bending, body	19.74	1.74	symmetric
12	2^{nd} swing, wings and body	20.13	0.86	anti-symmetric
13	3^{rd} bending, wings	21.02	2.06	symmetric
14	2^{nd} swing, wings	25.32	1.98	symmetric
15	1^{st} torsion, wings	27.87	1.26	symmetric
16	1^{st} torsion, wings	28.99	3.01	anti-symmetric

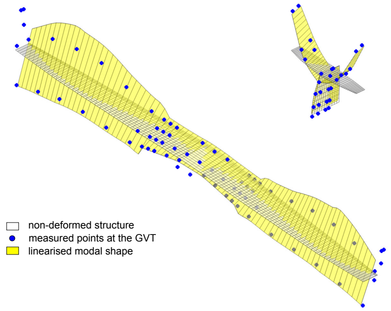


Fig. 4 Linearised modal shape of the first symmetric wing torsion and the eigenvector points from the GVT (only the modal shape's components perpendicular to the corresponding lifting surfaces are shown)

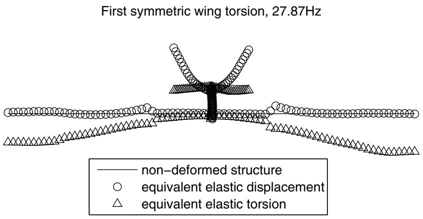

First symmetric wing torsion, 27.87Hz
non–deformed structure
○ equivalent elastic displacement
△ equivalent elastic torsion

Fig. 5 Equivalent spanwise elastic displacement and torsion of the first symmetric wing torsion, used in the simulation model

4 Model Validation

To validate the aeroelastic simulation program in the time domain, manoeuvres were specified to a limited portion of the aircraft flight envelope. Basically, manoeuvres adding sufficient energy into the system were chosen, while keeping the excitation amplitudes at a low level for safety reasons. Accelerometers distributed over the aircraft structure measured the structural behaviour. The results were compared to time simulation for the same (also measured) inputs and flight condition as similar as possible.

For the validation of the aeroelastic simulation model, an independent aeroelastic measurement system was integrated. The equipment consisted basically of 7 ICP (integrated circuit piezoelectric) one-dimensional accelerometers of the type 352C65 from the sensor manufacturer PCB[7], low noise cables and an acquisition system, a small portable SCADAS Recorder device from LMS[8]. The 352C65 sensor type is a miniature accelerometer with 11.2mm height and less than 10.0mm diameter, weighing only 2 grams. It measures accelerations up to ±50g peak in a frequency range from 0.5 up to 10,000Hz and operates at temperatures from -54°C up to 93°C. It is ideal for applications where it is desired to add as little as possible extra weight to the test structure. The LMS SCADAS recorder is an autonomous acquisition system built on the SCADAS mobile technology [8]. The SCM05 version has up to 40 channels and a GPS antenna, and the acquired data are recorded on a compact flash card. The data can be processed later using the device in a PC front end configuration. The unity has the plant size of a laptop (345-92-300mm), weighs 6.2kg and its internal battery has 1h operation autonomy. A photo of the SCADAS Recorder is shown in Fig. 6, mounted over the co-pilot seat in S15, in a front-end configuration to program the compact flash card for the automatic recording function.

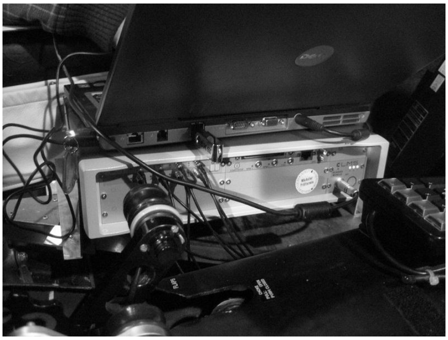


Fig. 6 SCADAS Recorder placed over the S15 co-pilot seat (here in a front-end configuration operating with a laptop to configure the compact flash card before flight)

[7] http://www.pcb.com/

[8] http://www.lmsintl.com/

The accelerometers were distributed in two measurement sets, namely:

- measurement set 1, specially for the behaviour of the wing - five sensors were placed on the left wing, organised at three different stations, including the wing tip; one sensor was mounted on the right wing tip to differentiate symmetric from anti-symmetric elastic modes; the remaining sensor, common to both measurement sets, was placed on the horizontal empennage as a link to both measurements;
- measurement set 2, specially for the behaviour of the tail - three sensors were placed on the vertical empennage, while the remaining four sensors were distributed on the horizontal empennage.

Figure 7 shows the assembly of the sensors S1 and S2 of measurement set 1, evidencing the placement details and its exact location, fixation and cable disposition, as well as the solutions to pass them through into the fuselage to the SCADAS recorder.

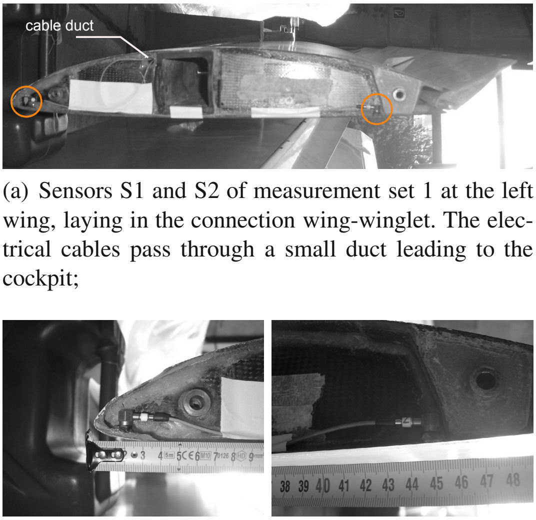


Fig. 7 Assembly of sensors S1 and S2 in measurement set 1 at the left wing

(a) Sensors S1 and S2 of measurement set 1 at the left wing, laying in the connection wing-winglet. The electrical cables pass through a small duct leading to the cockpit;

(b) Sensor S1 close to the wing leading edge in detail;

(c) Sensor S2 close to the wing trailing edge in detail.

For the excitations, the control surfaces were used. Advantage was taken of the S15's full-authority flight control system (AFCS). The desired excitation signals were programmed and their index and parameters could be selected using the reprogrammed auto-pilot control panel (AFCP), developed by TUB. As a whole, the flight test campaign used the following excitation signals:

- sinusoidal inputs of constant frequency;
- sinusoidal inputs of increasing and decreasing frequencies (sine-sweep);
- 3-2-1-1 step input.

The response at sensor S1 positioned at the left wing tip as well as the deflection at the left wing's internal flap (IL) can be seen in Fig. 8 for the sinusoidal input using the flaps, at a constant frequency of 3.29Hz and 120km/h IAS. A very good coherence between measured and simulated acceleration at S1 in the permanent response and an almost unnoticeable phase difference can be observed.

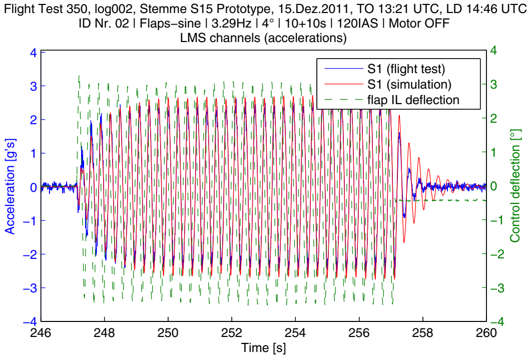


Fig. 8 S1 accelerations and flap deflection in a 3.29Hz flap sine manoeuvre at 120km/h IAS

Overall good results were reported from the comparison of the simulated results and the flight test data. The damping factors predicted by the simulation model were lower than the damping factors determined from flight test data. Therefore it can be concluded that the model is conservative, i.e. the model is on the safe side. In general, and also considering the difficulties in calculating the damping ratios from flight test data due to signal degradation (noise for example), the values predicted by the flight simulation model can be considered satisfactory - an average of up to 80% of the measured value for modes 2 (first anti-symmetric swing, wings and fuselage) and 3 (first anti-symmetric wing bending). The amplitudes of the accelerations in the permanent excitation, regarding the flaps and ailerons, were predicted also satisfactorily, with averages under 20% error. For the rudder, important differences in amplitude appear, which die out with the flight speed, pointing out the importance of the rudder effectiveness, not considered in the model. Phase differences become important for the excitation of modes 2 and 3. The analyses of the sweep manoeuvres in the frequency domain indicated a small difference in the resonance frequencies between model and flight measurement. Small differences in frequency have a great impact on the phase.

5 Aeroservoelastic Stability Analysis

In this section, the aeroelastic model will be used to predict the influence of aircraft control on the stability of the elastic modes. Two simple examples are analysed: a yaw damper, with feedback from the yawing rate to the rudder; and a pitch damper, with feedback from the pitching rate to the elevator.

The integrated dynamic model of the flexible aircraft can be used to predict the contribution of the flexibility to the angular rates based on the position of the respective sensor. Normally the sensors that measure the angular rates are located close to the aircraft CG and at the aircraft symmetry plane, where the flexibility effects are minimal. To demonstrate the effects of the flexibility, two different hypothetical sensor locations are chosen, which are shown in Fig.9. Point P1 lies 0.67m from the aircraft symmetry plane at the right wing, and 0,60m aft from the wing leading edge. Point P2 lies at the VTP, 0.64m from the VTP leading edge, at a distance 0.16m from the fuselage.

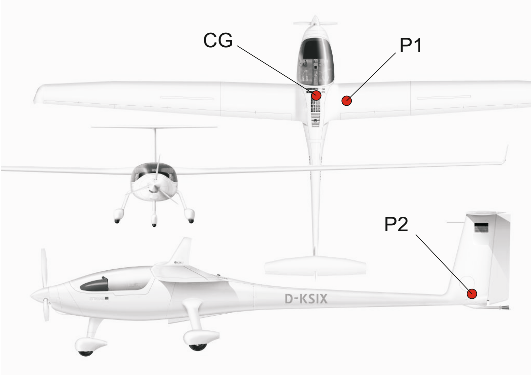


Fig. 9 Hypothetical sensor positioning to measure the aircraft angular rates

Being $\boldsymbol{\Phi}(\mathbf{P})$ the vectorial function of the elastic displacements at a generic point $\mathbf{P}$ of the aircraft structure, the elastic contributions to the roll, pitch and yaw rates measured at this point are, respectively:

$$\begin{aligned}
\Delta p^{\text{flex}} &= \left[\frac{\partial \boldsymbol{\Phi}_z(\mathbf{P})}{\partial y} - \frac{\partial \boldsymbol{\Phi}_y(\mathbf{P})}{\partial z}\right]\dot{\boldsymbol{\eta}} \\
\Delta q^{\text{flex}} &= \left[-\frac{\partial \boldsymbol{\Phi}_z(\mathbf{P})}{\partial x} + \frac{\partial \boldsymbol{\Phi}_x(\mathbf{P})}{\partial z}\right]\dot{\boldsymbol{\eta}} \\
\Delta r^{\text{flex}} &= \left[\frac{\partial \boldsymbol{\Phi}_y(\mathbf{P})}{\partial x} - \frac{\partial \boldsymbol{\Phi}_x(\mathbf{P})}{\partial y}\right]\dot{\boldsymbol{\eta}}.
\end{aligned} \tag{9}$$

The elastic modes contribute to the angular rates at their eigen frequency. If the angular rates are fed back to the control system, there is a demand for the control system to act at the same frequencies. The result is a coupling of the control with the elastic mode, which can increase or decrease the modal damping.

For the following aeroservoelastic investigations, the aircraft was linearised at the flight condition of Tab. 4. The yaw damper is simply a feedback from the yaw rate to the rudder command, so that:

Table 4 Reference condition for aeroservoelastic investigations

Parameter	Value	Unit
mass	1000	kg
CG position (from nose)	2.64	m
velocity (TAS)	50	m/s
altitude	1200	m
throttle	98	%
RPM	1953.5	RPM
flaps	0	°
sideslip angle	0	°
bank angle	0	°

$$\begin{aligned}\delta_{\text{rudder}} &= K_r r^{\text{sensor}} \\ &= K_r \left(r^{\text{RB}} + \Delta r^{\text{flex}}\right).\end{aligned} \tag{10}$$

Here, r^{RB} denotes the "rigid body" yaw rate, i.e. the yaw rate due to the angular motion of the aircraft without flexibility effects, while r^{sensor} is the measured yaw rate at the respective sensor, thus accounting for the flexibility effects.

Similarly, the pitch damper consists in a feedback of the pitch rate to the elevator command, so that:

$$\begin{aligned}\delta_{\text{elevator}} &= K_q q^{\text{sensor}} \\ &= K_q \left(q^{\text{RB}} + \Delta q^{\text{flex}}\right).\end{aligned} \tag{11}$$

Reaction time and filters were ignored in this demonstration. Figures 10 and 11 show the root-loci for the gain K_r varying from 0 to 1°/(°/s), for the yaw rate measurement at point P1 and P2 respectively. In the first case, the root-loci of the elastic modes remain almost unchanged as K_r increases. Point P1 lies on the wing, thus an incremental elastic yaw rate would arise from a wing swing mode. Staying close to the fuselage, rotations due to the swing mode are locally negligible. On the other side, for measurements of yaw rate at point P2, Fig. 11 shows that elastic modes 2, 8, 12 and 16 get unstable. The first anti-symmetric wing swing (mode 2) gets unstable for $K_r = 0.3°/(°/\text{s})$. This elastic mode, with a high contribution from HTP and VTP, has a great influence over Δr^{flex}. The same can also be said from modes 5 (first fuselage torsion), 8 (first swing, horizontal fin), 12 (second swing, wings and body) and 16 (first anti-symmetric wing torsion) - refer to Tab. 3 to the association of elastic mode number and mode description. In Fig. 12 the root-loci of the dutch roll mode are shown in detail, comparing both measurement points P1 and P2. No important differences are observed - for a slightly flexible aircraft, the elastic contribution to the angular rate is small compared to the body angular rate.

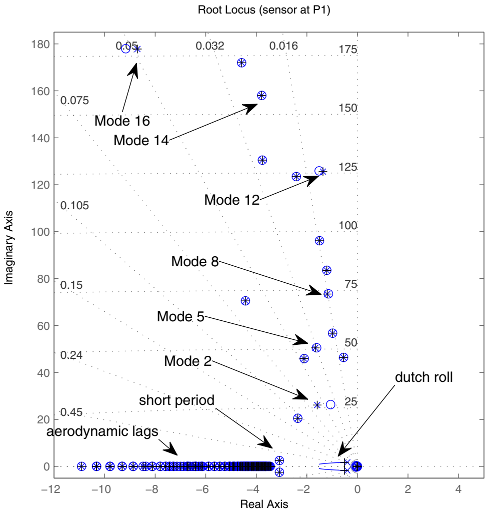


Fig. 10 Root-loci of the flexible aircraft dynamics at 50m/s with yaw damper, the gain K_r varying from 0 to 1s; the yaw rate measured from point P1. × poles; **o** zeros; + $K_r = 0.1$s.

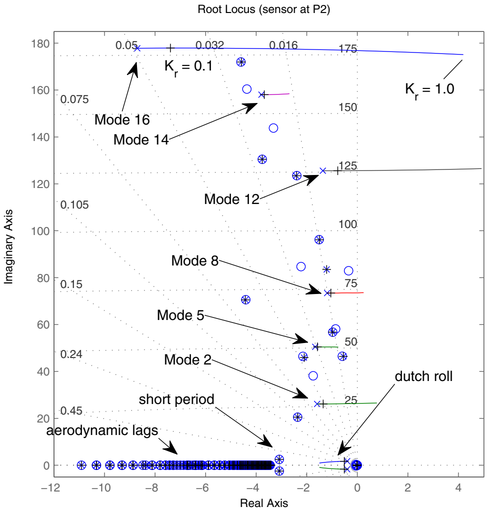


Fig. 11 Root-loci of the flexible aircraft dynamics at 50m/s with yaw damper, the gain K_r varying from 0 to 1s; the yaw rate measured from point P2. × poles; **o** zeros; + $K_r = 0.1$s.

The root-loci of th e aircraft dynamics with the pitch damper are displayed in Figures 13 and 14 respectively for measurements of pitch rate at points P1 and P2. The flexibility can induce a pitch rate at point P1 by means of wing torsion or fuselage bending. At point P2, the most important effect is the fuselage bending. In Fig. 13 it can be observed that mode 6 (first symmetric fuselage bending) gets unstable (for K_q approximately 0.4°/(°/s)). Mode 14 (second symmetric wing swing) moves in the direction of the unstable region, and mode 15 (first symmetric torsion of the

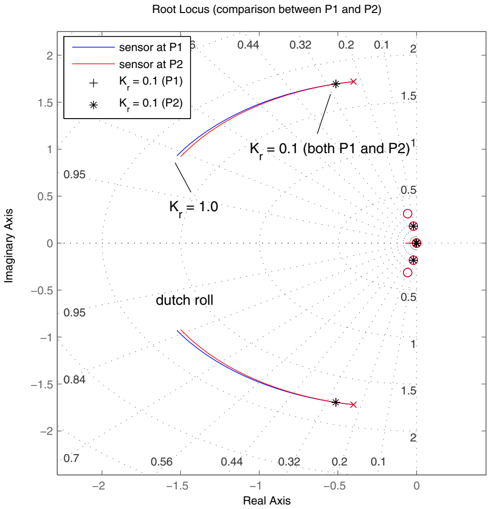


Fig. 12 Root-loci of the flexible aircraft dynamics at 50m/s with yaw damper, the gain K_r varying from 0 to 1s; comparison of low frequency root-loci for sensor positioning at P1 and P2. × poles; **o** zeros; + $K_r = 0.1$s (P1); $*$ $K_r = 0.1$s (P2).

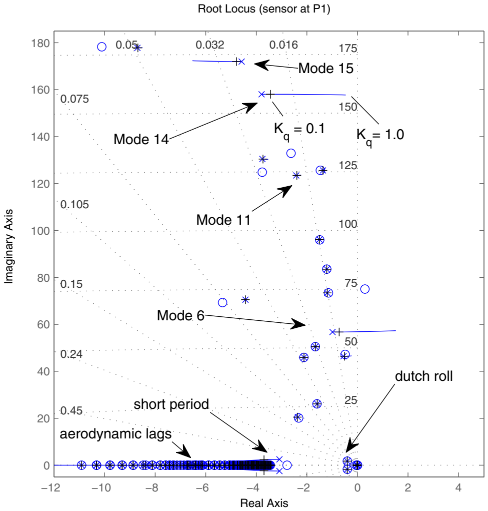


Fig. 13 Root-loci of the flexible aircraft dynamics at 50m/s with pitch damper, the gain K_q varying from 0 to 1; the pitch rate measured from point P1. × poles; **o** zeros; + $K_q = 0.1$s.

wings) gets more stable. Turning to Fig. 14 for point P2, mode 6 gets more stable, while mode 11 (second fuselage bending) gets unstable for K_q approximately 0.75 °/(°/s). The different root-loci of the elastic mode 6 are shown in Fig. 15 comparing the sensor position at P1 and P2 - observe the importance of the sensor placement. In the same figure, the root-loci of the short-period are shown, and no important differences can be observed regarding the sensor position, for the same reason discussed above for the yaw damper.

Fig. 14 Root-loci of the flexible aircraft dynamics at 50m/s with pitch damper, the gain K_q varying from 0 to 1; the pitch rate measured from point P2. × poles; **o** zeros; $+ K_q = 0.1$s.

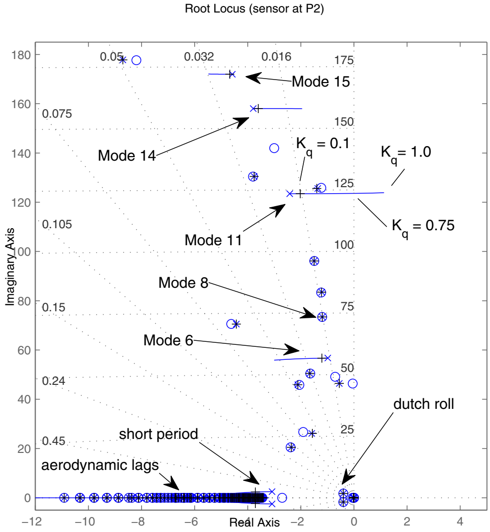
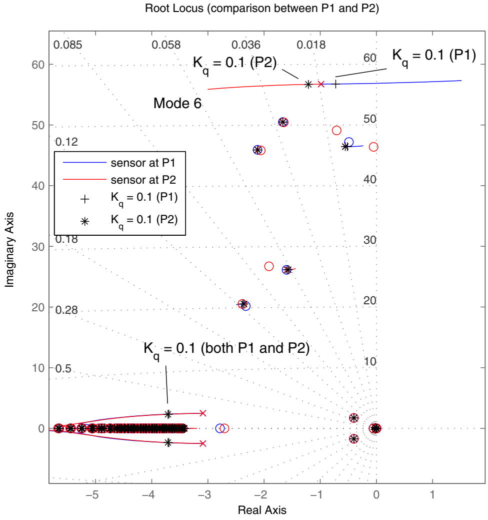


Fig. 15 Root-loci of the flexible aircraft dynamics at 50m/s with pitch damper, the gain K_q varying from 0 to 1; comparison of low frequency root-loci for sensor positioning at P1 and P2. × poles; **o** zeros; $+ K_q = 0.1$s (P1); $* K_q = 0.1$s (P2).



6 Conclusions

This paper demonstrated the application of a methodology of modelling the dynamics of slightly flexible, high-aspect-ratio aircraft in the time domain for aeroservoelastic investigations. In the examples given in this paper, the effects of the flexibility measured by angular rate sensors were illustrated concerning the stability of the closed-loop system for an yaw and a pitch damper. It is clearly seen that feeding back a signal containing flexible contributions (at different frequencies) can make the overall system unstable. Similarly, control could be applied to increase the damping of the elastic modes. Selecting two different sensor positions, the influence of different modes in the closed-loop system were outlined. These simple examples evidence the relevance of this methodology. Validation of this methodology with flight test results attested the fidelity of the model and therefore substantiates its applicability to this kind of problem. It is expected that this methodology will be used to assist the control law design for the class of aircraft discussed here.

Acknowledgements. This work was developed at the Technical University of Berlin (TUB) with assistance of STEMME AG and the Technical University of Hamburg-Harburg (TUHH), with a scholarship from the German Academic Exchange Service (DAAD). The author is very thankful to these organisations for the support, as well as to Prof. Dr.-Ing. Robert Luckner (TUB) for his technical supervision.

References

1. Wagner, H.: Über die Entstehung des dynamischen Auftriebes von Tragflügeln. Zeitschrift für angewandte Mathematik und Mechanik (1925), doi:10.1002/zamm.19250050103
2. Milne, R.D.: Dynamics of the Deformable Aeroplane. Her Majesty's Stationary Office (1964), http://aerade.cranfield.ac.uk/ara/arc/rm/3345.pdf
3. Waszak, M.R., Schmidt, D.K.: Flight Dynamics of Aeroelastic Vehicles. Journal of Aircraft (1988), doi:10.2514/3.45623
4. Noll, T.E., Eastep, F.E.: Active Flexible Wing Program (Editorial). Journal of Aircraft (1995), doi:10.2514/3.56918
5. Pery III, B., Cole, S.R., Miller, G.D.: Summary of an Active Flexible Wing Program. Journal of Aircraft (1995), doi:10.2514/3.46677
6. Bisplinghoff, R.L., Ashley, H., Halfman, R.L.: Aeroelasticity. Dover Publ., Mineola (1996) ISBN: 0-486-69189-6
7. Fielding, C., Luckner, R.: Industrial Considerations for Flight Control (2000), doi:10.2514/5.9781600866555.0001.0055
8. LMS. LMS SCADAS Mobile: User and Installation Manual. LMS International (2007), http://www.lmsintl.com/
9. Krüger, W.: Neue Bauweisen, neue Konfigurationen, neue Fragen: Herausforderungen für die Aeroelastische Simulation. Kolloquium zum Abschied von Prof. Hönlinger (2008), http://elib.dlr.de/56117/
10. Looye, G.H.N.: An Integrated Approach to Aircraft Modelling and Flight Control Law Design. PhD. Dissertation, Technische Universiteit Delft (2008) ISBN: 978-90-5335-148-2
11. Silvestre, F.J.: Dynamics and Control of a Flexible Aircraft. In: Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, Hawaii, USA (2008)
12. Weltin, U.: Flattersicherheit S6T. Internal Report (not published), Technische Universität Hamburg-Harburg (2008) (reserved)
13. Danowsky, B.P., Thompson, P.M., Farhat, C., Lieu, T., Harris, C., Lechniak, J.: Incorporation of Feedback Control into a High-Fidelity Aeroservoelastic Fighter Aircraft Model (2010), doi:10.2514/1.47119

14. Rudenko, A.: Experimentelle Bestimmung der Massenträgheitsmomente eines Kleinflugzeuges mit Hilfe eines Pendelversuches unter Anwendung numerischer Methoden der Systemidentifikation. Abschlussarbeit an der Technischen Universität Berlin (2010)
15. Meyer-Brügel, W., Luckner, R.: Flight Mechanical Simulation Models for Design and Test of Automatic Flight Control Functions. In: Proceedings of the CEAS EURO GNC 2011 Conference, München, Germany (2011)
16. Silvestre, F.J.: Methodology for Modelling the Dynamics of Slightly Flexible, High-Aspect-Ratio Aircraft in the Time Domain for Aeroservoelastic Investigations. Doctoral Thesis at TU Berlin, Berlin, Germany (2012)
17. Dalldorff, L., Luckner, R., Reickel, R.: A Full-Authority Automatic Flight Control System for the Civil Airborne Utility Aircraft S15 - LAPAZ. In: 2nd Euro GNC 2013, Delft, The Netherlands (2013)

Part IV
Space Applications

Relative Optical Navigation for a Lunar Lander Mission

Mark J. Verweld

Abstract. This work explores the problem of providing relative velocity navigation for an autonomous precision landing approach on the moon without the use of telemetry or known points of support. An error-state Unscented Kalman Filter for the fusion of inertial and optical imaging sensors is presented. These sensors include a star tracker, a monocular surface camera and a laser altimeter. The filter estimates position, velocity and attitude, which, together with an initial position based on crater matching, allows for trajectory following to the surface. A main difficulty is the scale ambiguity in optical flow. The laser altimeter has been included to resolve this ambiguity and allow for velocity and altitude estimation. The scenario of a lunar landing from parking orbit was chosen to test and verify the developed navigation method in simulation using a high resolution surface model of the moon.

1 Introduction

The goals of future space exploration missions include investigation of local surface phenomena on moons, planets and asteroids as well as the building and support of outposts. Autonomous, precise and safe landings near hazardous terrain are key requirements for such missions[1]. These requirements call for a complex navigation system capable of providing accurate state estimation independent of supporting telemetry. Optical imaging sensors may form an important component in such a system as they allow low-latency measurements to be taken independently from Earth, enabling autonomy. Integrating optical measurements into the

Mark J. Verweld
DLR, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: Mark.Verveld@dlr.de

[1] 2006 Solar System Exploration Roadmap for NASA's Science Mission Directorate, http://solarsystem.nasa.gov/multimedia/downloads/SSE_RoadMap_2006_Report_FC-A_opt.pdf

navigation system provides position and attitude determination with respect to the target body allowing for a precise autonomous landing.

Several past and present projects have investigated the problem of autonomous planetary precision landing. The ESA commissioned Navigation for Planetary Approach & Landing (NPAL) [4] project focused on guidance and navigation algorithms based on the tracking of unknown landmarks as well as navigation camera hardware design and the PANGU planetary surface generator. An Extended Kalman Filter (EKF) was used for state estimation. The following VisNAV [3] project extended NPAL developed vision based navigation schemes for use in a broader scope of planetary navigation. The NASA commissioned Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) [1,19] project develops precision navigation and hazard detection and avoidance for planetary landing. The goals include landing ability without surface illumination, thus requiring active sensors. Development is directed at a flash LIDAR for this purpose. Map matching is also mentioned. Johnson et al. [5,15] match descent images to a map in which features have a known 3D position using SIFT keypoints. They combine this with persistent and image-to-image feature tracking and fuse the optical data with inertial measurements in an EKF. S. Li et al. [12] combine feature tracking with a LIDAR and apply it to navigation during the landing on asteroids. They match features found in the image of a camera with distances found using the LIDAR to enable 6 degree-of-freedom relative position estimation. In a second paper [13] by the same authors a navigation scheme for planetary landing is discussed on the basis of an EKF which fuses an IMU and the optical flow from a single camera without known support points. It shows a significant reduction in the position error growth rate compared to inertial navigation alone.

The Autonomous Terrain based Optical Navigation for landers (ATON) project, part of which is the work presented in this paper, develops the optical navigation and obstacle avoidance technology to satisfy planetary precision landing requirements. The scenario studied in the project is that of a lunar landing from a 100 km circular parking orbit. The sensors available for this task are a six degree of freedom inertial measurement unit (IMU), a star tracker for inertial attitude determination, a surface camera and a laser altimeter. There are three modes of navigation employed, which may be combined depending upon the available visual input.

Initially, navigation uses the surface camera together with a database of known surface features (craters in the case of the moon), the star tracker and IMU. This allows for direct position and attitude determination in a moon fixed frame of reference. As the lander gets closer to the surface fewer features in the camera field of view may be matched to the database. Therefore a relative velocity navigation mode is included which uses the optical flow (OF) from unknown surface features to estimate the vehicle velocity with respect to the lunar surface and integrate this to yield the position. Finally, as the lander gets the intended landing area in sight, it will navigate relative to a hazard free landing spot. This work presents the relative velocity navigation mode and assumes navigation based solely on this mode.

The navigation filter uses the Unscented Kalman Filter (UKF) algorithm at it's core. We have chosen the UKF for its reported [6,7,11,20] accuracy benefits over the

EKF in problems with strong nonlinearities in their observation models. The delays associated with especially image processing required for both the star tracker and surface camera have to be accounted for in the navigation filter. Using an indirect or error state approach simplifies this process [16, 18]. We have combined both concepts into the error-state UKF (eUKF).

The paper is composed as follows. Section 2 explains the sensor concept and develops the equations modeling sensor behavior. Section 3 describes the navigation filter and its eUKF equations. A significant part of the ATON project has been the development of a comprehensive lunar landing simulation environment including accurate sensor emulation and detailed lunar surface rendering using ray tracing and specular lighting techniques. The surface digital elevation map (DEM) for the visual simulation uses data from the Kaguya mission [9]. This simulation environment used for testing and validation of the presented method will be further explained in section 4. The results obtained in the simulation will be presented and discussed in section 5. Finally, section 6 draws overall conclusions about the presented navigation method and gives recommendations for further work.

2 Modeling

This section provides an overview of the spacecraft equations of motion and the observation models used for the sensor fusion. The navigation filter as presented in section 3 uses error states in its prediction step so the equations developed in this section will then be cast into their error state formulation.

2.1 Kinematics

The vehicle navigation state $\mathbf{x}$ is expressed in Cartesian moon-fixed, moon-centered coordinates denoted by subscript m. It consists of the velocity $\mathbf{v}_m$, the position $\mathbf{r}_m$, the lander attitude quaternion q_m, the bias of the accelerometers $\mathbf{b}_a$ and the bias of the gyroscopes $\mathbf{b}_g$. The IMU, delivering a specific force vector $\mathbf{a}$ and a rotational rate vector ω, is used in the filter's prediction step to propagate this state vector. The subscript *imu* denotes the measured values, whereas the subscript b denotes the true values in the body-fixed frame of reference. We employ a stochastic model due to [2] using Gaussian white noise processes designated by the $\mathbf{n}$ terms in $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{a}_{imu} \\ \omega_{imu} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_b + \mathbf{b}_a + \mathbf{n}_{a_1} \\ \omega_b + \mathbf{b}_g + \mathbf{n}_{g_1} \end{bmatrix}, \qquad \begin{bmatrix} \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{a_2} \\ \mathbf{n}_{g_2} \end{bmatrix}. \tag{1}$$

A kinematic system of continuous time differential equations describes the state derivative w.r.t. time as a function of the state itself and the IMU outputs, where $w(t)$ collects all additive Gaussian white noise terms:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{v}}_m \\ \dot{\mathbf{r}}_m \\ \dot{q}_m \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \end{bmatrix} = f(\mathbf{x}(t), \mathbf{y}(t)) + w(t) = \begin{bmatrix} R_{b\rightarrow m}\mathbf{a}_b + \mathbf{g}_m \\ \mathbf{v}_m \\ \frac{1}{2}B(q_m)\,\omega_b \\ 0 \\ 0 \end{bmatrix} + w(t)\ . \tag{2}$$

The vector $\mathbf{g}_m$ is the gravitational acceleration at position $\mathbf{r}_m$ above the lunar surface. This may be expressed using Newton's law of gravitation as

$$\mathbf{g}_m = -\mathbf{r}_m \frac{GM}{\|\mathbf{r}_m\|_2^3}\ ,$$

where GM is the gravitational parameter of the moon in this case. The 4×3 matrix $B(q_m)$ is the quaternion derivative matrix [17] and $R_{b\rightarrow m}$ is the rotation matrix [17] from body-fixed to moon-fixed frame.

Since the Kalman filter will be computing in discrete time, a state transition function

$$\mathbf{x}(k+1) = \phi(\mathbf{x}(k), \mathbf{y}(k)) + w(k)\ , \quad w(k) \sim \mathcal{N}(\mathbf{0}, Q_k) \tag{3}$$

may be derived from Eq. (2) by applying Euler discretization. The estimated state is computed using the same ϕ:

$$\hat{\mathbf{x}}(k+1) = \phi(\mathbf{x}(k), \mathbf{y}(k)) \tag{4}$$

2.2 Observation Models

The optical sensors used for navigation include a surface camera with 40° field of view and 1024 × 1024 pixel resolution, an optical star tracker with an accuracy in the order of 10 arcsec and optionally a laser altimeter with 5 m standard deviation. The configuration is shown in Fig. 1. In order to fuse these optical sensors with the IMU, they have to be modeled by equations describing their outputs as a function of time, the state vector **x** and the IMU outputs **y**.

The star tracker delivers the attitude quaternion of the spacecraft in the International Celestial Reference Frame (ICRF). This may be computed using time, the lunar ephemeris and **x**.

Images from the surface camera are processed using a feature tracker based on the Lucas-Kanade [14] algorithm. It yields a set of feature positions, represented by unit length direction vectors ξ_c in the camera frame (as denoted by the subscript c), plus their projected displacement rates χ_c, also referred to as optical flow. The directions ξ_c themselves do not contain information on the camera motion and they are taken as a given. The displacement rates however may be expressed as a function of **x**, **y** and ξ_c assuming all motion is due to the spacecraft. The starting point is to express the motion of these features as an angular rate Ω_c. This angular rate consists of a component due to translational motion Ω_{c_t} and a component due to rotational motion Ω_{c_r}. The translational velocity vector of the spacecraft in the camera frame $\mathbf{v}_c$ must be considered with respect to the direction ξ_c and the distance D to each

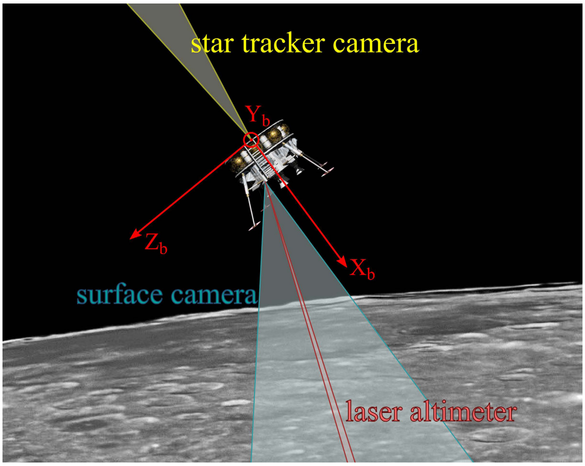


Fig. 1 Schematic optical sensor configuration

feature. Only the component of $\mathbf{v}_c$ which is perpendicular to ξ_c contributes to Ω_{c_t}. This is expressed by the crossproduct. After dividing by D one arrives at the angular rate (in rad/s) due to translation:

$$\Omega_{c_t} = \frac{\mathbf{v}_c \times \xi_c}{D} \tag{5}$$

The rotational component of Ω_c is equal to the component of the rotational rate of the spacecraft which is again perpendicular to ξ_c. Applying the crossproduct again achieves this although the resulting direction is 90° false. By applying the crossproduct with the unit vector ξ_c a final time we arrive at the desired angular rate due to rotation:

$$\Omega_{c_r} = (\omega_c \times \xi_c) \times \xi_c \tag{6}$$

The total angular rate of the feature's motion in the camera frame of reference is the sum of the translational and rotational components:

$$\Omega_c = \Omega_{c_t} + \Omega_{c_r} = \frac{\mathbf{v}_c \times \xi_c}{D} + (\omega_c \times \xi_c) \times \xi_c \,, \tag{7}$$

The camera frame is defined such that the image is projected parallel to the X-Y plane, with the X-axis pointing up and the Y-axis to the right when looking through the lens. The Z-axis coincides with the optical axis to form a right-handed coordinate frame.

The two dimensional optical flow vector χ_c is the projected feature displacement rate as seen in the camera frame. We use rectified images such that a pinhole

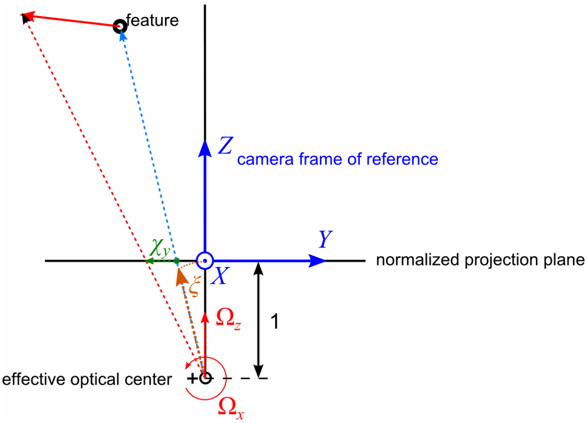


Fig. 2 Diagram showing the y-component of χ

projection model may be applied. The geometry is drawn in Fig. 2, from which the following expression may be found:

$$\chi_c = \begin{bmatrix} -\Omega_{c,z}\frac{\xi_{c,y}}{\xi_{c,z}} + \frac{1+\xi_{c,x}^2}{\xi_{c,z}^2}\Omega_{c,y} \\ \Omega_{c,z}\frac{\xi_{c,x}}{\xi_{c,z}} - \frac{1+\xi_{c,y}^2}{\xi_{c,z}^2}\Omega_{c,x} \end{bmatrix} . \quad (8)$$

The subscripts x, y and z denote the component along that axis in the camera frame of reference.

In order to use Eqs. 7 and 8 in the navigation filter, the scale of the optical flow geometry in Figure 3 must be resolved. This scale appears in Eq. (7) as the distance D to a feature on the lunar surface. D must be described as a function of $\mathbf{x}$ and $\mathbf{y}$.

One approach, used in the NPAL project [4], relies on the rate of change in the translational optical flow to determine distances to tracked features. From Eq. (7), only optical flow due to translation scales with distance. By comparing the acceleration of features due to translational motion of the camera to the specific force as measured by the accelerometers, one can observe the distances to these features. However, the distance estimates require sufficiently large accelerations in relation to the distances for this method to work. It may be expected to work in the final approach phase where the vertical descent generates a diverging optical flow field. The current application requires the relative velocity navigation earlier in the approach where conditions are unfavorable. We therefore developed another approach.

This approach to derive a relation between the navigation state $\mathbf{x}$ and the distance to a tracked feature on the lunar surface involves using the estimated vehicle position $\mathbf{r}_m$, its attitude q_m and information about the shape of the moon. This method has the advantage that its accuracy does not depend on the details of the vehicle motion. The current implementation is based on an ellipsoid model, although one could in principle substitute a detailed DEM of the surface. The method is well suited to

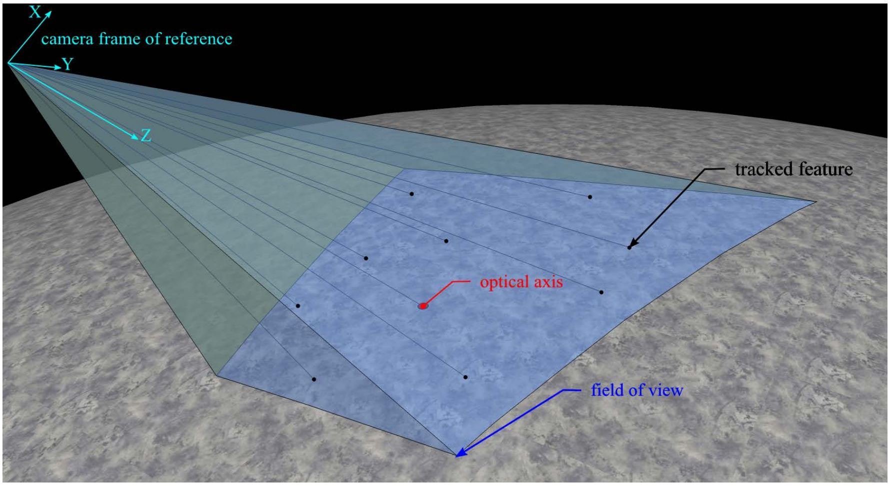


Fig. 3 Schematic geometry for the surface camera

incorporate corrections from the laser altimeter. This presents another advantage over the derivative based method.

The distances D belonging to the feature directions ξ_c are estimated by calculating the intersection with the lunar ellipsoid Λ which, in a moon-centered, Cartesian reference frame, has the following form

$$\Lambda : \mathbf{r}_{le}^T \Theta \mathbf{r}_{le} = 1 \,, \tag{9}$$

with $\mathbf{r}_{le}$ being the set of position vectors on the lunar ellipsoid and Θ being a 3×3 matrix defining the size and shape of the ellipsoid. To calculate D belonging to the intersection of a feature in direction ξ_m with Λ from viewing position $\mathbf{r}_m$, we define a line of sight $\mathbf{s}$ parametrized in terms of D:

$$\mathbf{s}(D) = D\xi_m + \mathbf{r}_m \,. \tag{10}$$

The problem may now be formulated as follows:

$$\mathbf{s} = \mathbf{r}_{le} \,. \tag{11}$$

Substituting Eq. (11) into Eq. (9) yields

$$\mathbf{s}^T \Theta \mathbf{s} = 1 \,. \tag{12}$$

Further, substituting Eq. (10) makes the distance appear:

$$(D\xi_m + \mathbf{r}_m)^T \Theta (D\xi_m + \mathbf{r}_m) = 1 \,. \tag{13}$$

which is a quadratic equation having either 0, 1 or 2 solutions for D. The distance to the surface is the smallest positive solution. If no positive solution is found, the distance may be considered infinite to represent the fact that the camera is looking at the sky. In this case Eq. (7) only depends on the rotational rate ω, which still provides information to correct the gyroscope bias $\mathbf{b}_g$.

In order to correct for the local surface height w.r.t. Λ, a common scaling factor c is applied to Θ yielding a local fit for the ellipsoid:

$$\Theta_{local} = \frac{1}{c^2}\Theta \ . \tag{14}$$

Substituting Eq. (14) into Eq. (13) yields

$$c^2 = \left(D_{laser}\xi_m + \mathbf{r}_m\right)^T \Theta \left(D_{laser}\xi_m + \mathbf{r}_m\right) \ . \tag{15}$$

When a laser altimeter measurement arrives, Eq. (15) may be solved for c by selecting ξ_m equal to the laser altimeter direction and substituting the laser distance measurement value D_{laser}.

As such, the ellipsoid has been fit to the point on the lunar surface measured by the laser altimeter. The direction of the laser altimeter is chosen to coincide with the optical axis of the surface camera, see Figure 3. This implicitly assumes a smooth ellipsoidal surface and any height deviation, e.g. due to mountain ridges or craters, must be treated as a measurement error. This has been incorporated in the observation model by making the measurement covariance for each tracked feature a function of the square of its Euclidean distance d to the intersection of the optical axis with the lunar surface.

$$R_\chi\left(d\right) = R_{\chi,0} + \delta R_\chi d^2 \tag{16}$$

3 Filter Algorithm

We have chosen to use the Unscented Kalman Filter [7] (UKF) as mentioned in the introduction as it is equivalent in accuracy to a second order filter without the need for deriving the Jacobian and Hessian matrices of the process and observation models.

In the ATON lunar lander, measurements from the IMU are available at a high sample rate and without significant delay while measurements from the optical sensors have a varying delay of up to 0.5 s and will be available at lower sample rates. This asynchronous, delayed nature of correction updates must be taken into account in the navigation filter. An indirect, or error-state formulation allows the full state to be propagated quickly based on inertial data, while the error state is updated using the other sensors. Feedback after such an update corrects the full state. The IMU is able to follow the high frequency motion of the spacecraft very accurately, while the inertial error propagation equations are low frequency [16]. The processing of optical sensor data may therefore be done in a separate thread as data become available. Logged recording time stamps allow the filter to account for delays by taking the error state and covariance at the corresponding filter timestep and applying the resulting innovation to the current filter state.

Let the error state be defined as $\delta\mathbf{x} =: \hat{\mathbf{x}} - \mathbf{x}$. By combining this definition with Eq. (3) and Eq. (4), the error state transition function becomes

$$\begin{aligned}\delta\mathbf{x}(k+1\,|\,k) &= \hat{\mathbf{x}}(k+1\,|\,k) - \mathbf{x}(k+1)\\ &= \phi\left(\hat{\mathbf{x}}(k|k),\mathbf{y}(k)\right) - \phi\left(\mathbf{x}(k),\mathbf{y}(k)\right) - w(k)\\ &= \phi\left(\hat{\mathbf{x}}(k|k),\mathbf{y}(k)\right) - \phi\left(\hat{\mathbf{x}}(k|k) - \delta\mathbf{x}(k|k),\mathbf{y}(k)\right) - w(k)\end{aligned} \tag{17}$$

where the notation $(k+1\,|\,k)$ means for timestep $k+1$ with information from timestep k. The full state propagation as presented in section 2 uses a quaternion attitude description for computational efficiency and stability. The error of the quaternion is not additive, but rather has the form:

$$\delta q = q \otimes \hat{q}^{-1}\;.$$

However, the error state definition as well as the Unscented transform used in the UKF assume additive error behavior. Therefore, in the error state filter the attitude is described using the axis-angle vector θ_m:

$$\theta_m =: \mathbf{e}\cdot\Phi, \qquad q_m = \begin{bmatrix}\mathbf{e}\sin\left(\frac{\Phi}{2}\right)\\ \cos\left(\frac{\Phi}{2}\right)\end{bmatrix}, \qquad \mathbf{e}^T\mathbf{e} = 1\;, \tag{18}$$

where $\mathbf{e}$ is the principle axis and Φ the angle of rotation about that axis. θ_m has the required additive error behavior. In the case of the error state, we assume incremental attitude changes. The error quaternion δq therefore corresponds very closely to a small rotation, so the fourth component will be close to unity and the attitude information is contained in the vector component $\delta\mathbf{q}$: $\delta q \simeq \left[\delta\mathbf{q}^T\;\,1\right]^T$. Further, if the rotation $\delta\Phi$ is sufficiently small, $\delta\mathbf{q}$ may be approximated as $\delta\mathbf{q} \simeq \frac{1}{2}\delta\theta$. Using this assumption and the error state definition, the continuous time differential equation for $\delta\theta_m$ becomes [16]:

$$\frac{d}{dt}\delta\theta_m = [[\omega_{imu}]]\,\delta\theta_m - \delta\mathbf{b}_g - \mathbf{n}_{g_1}\;, \tag{19}$$

where $[[\omega_{imu}]]$ is a 3×3 skew symmetric matrix. Equation (19) may be discretized to yield the state transition function for $\delta\theta_m$.

The error state $\delta\mathbf{x}$ shows the same stochastic behavior as the full state $\mathbf{x}$ since the estimated state $\hat{\mathbf{x}}$ is not a stochastic variable. Therefore, the error covariance matrix P and the process noise covariance matrix for the error state are identical to those of the full state.

The delayed feedback error state UKF equations may be summarized as follows.

1. The error state estimate and covariance are augmented with the mean and covariance of the process noise $w(k)$:

$$\delta\mathbf{x}^a(k|k) = \left[\,\delta\mathbf{x}^T(k|k)\; E\left\{w^T(k+1)\right\}\,\right]^T ,$$

$$P^a(k|k) = \begin{bmatrix} P(k|k) & 0 \\ 0 & Q_k \end{bmatrix} .$$

2. A set of $2L+1$ prediction sigma points is derived from the augmented state and covariance where L is the dimension of the augmented error state:

$$\varsigma_{p,0}(k|k) = \delta\mathbf{x}^a(k|k) ,$$

$$\varsigma_{p,i}(k|k) = \delta\mathbf{x}^a(k|k) + \left(\sqrt{(L+\lambda)P^a(k|k)}\right)_i , \text{ for } i = 1\ldots L ,$$

$$\varsigma_{p,i}(k|k) = \delta\mathbf{x}^a(k|k) - \left(\sqrt{(L+\lambda)P^a(k|k)}\right)_{i-L} , \text{ for } i = (L+1)\ldots 2L ,$$

where

$$\left(\sqrt{(L+\lambda)P^a(k|k)}\right)_i$$

is the ith column of the matrix square root of

$$(L+\lambda)P^a(k|k)$$

using the definition: The matrix square root A of B satisfies $B =: AA^T$.

3. The prediction sigma points are propagated through the state transition function for $\delta\mathbf{x}$:

$$\varsigma_{p,i}(k+1|k) = \phi\left(\hat{\mathbf{x}}(k|k), \mathbf{y}(k)\right) - \phi\left(\hat{\mathbf{x}}(k|k) - \varsigma_{p,i}(k|k), \mathbf{y}(k)\right) , \text{ for } i = 0\ldots 2L .$$

4. The propagated sigma points are recombined to produce the predicted state and covariance:

$$\delta\mathbf{x}(k+1|k) = \sum_{i=0}^{2L} W_s(i)\,\varsigma_{p,i}(k+1|k) ,$$

$$P(k+1|k) = \sum_{i=0}^{2L} W_c(i)\left[\varsigma_{p,i}(k+1|k) - \delta\mathbf{x}(k+1|k)\right]\left[\varsigma_{p,i}(k+1|k) - \delta\mathbf{x}(k+1|k)\right]^T ,$$

where the weights for the state and covariance are given by:

$$W_s(0) = \frac{\lambda}{L+\lambda} ,$$

$$W_c(0) = \frac{\lambda}{L+\lambda} + \left(1 - \alpha^2 + \beta\right) ,$$

$$W_s(i) = W_c(i) = \frac{1}{2(L+\lambda)} ,$$

$$\lambda = \alpha^2(L+\kappa) - L .$$

Values for α, β and κ have to be chosen to tune the prediction step. Some guidelines to choose these constants for a particular problem are given in Ref. 8.

5. For the update step the availability of new data from the surface camera and/or star tracker is checked. If these are not available, the current prediction is used for the next timestep: So $\delta\mathbf{x}(k+1\,|\,k+1) = \delta\mathbf{x}(k+1\,|\,k)$ and $P(k+1\,|\,k+1) = P(k+1\,|\,k)$ and no feedback to the full state takes place. Otherwise, the predicted state and covariance corresponding to timestep $l \leq k$ of the incoming data are augmented with the mean and covariance of the measurement noise of the available sensors R^*:

$$\delta\mathbf{x}_u^a(l+1\,|\,l) = \left[\,\delta\mathbf{x}^T(l+1\,|\,l)\; E\left\{v^T(l)\right\}\,\right]^T ,$$

$$P_u^a(l+1\,|\,l) = \begin{bmatrix} P(l+1\,|\,l) & 0 \\ 0 & R^* \end{bmatrix} .$$

6. A set of $2L+1$ update sigma points is derived from $\delta\mathbf{x}_u^a(l+1\,|\,l)$ and $P_u^a(l+1\,|\,l)$ where L is the dimension of the augmented state:

$$\varsigma_{u,0}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) ,$$

$$\varsigma_{u,i}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) + \left(\sqrt{(L+\lambda)\,P_u^a(l+1\,|\,l)}\right)_i , \text{ for } i = 1\ldots L ,$$

$$\varsigma_{u,i}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) - \left(\sqrt{(L+\lambda)\,P_u^a(l+1\,|\,l)}\right)_{i-L} , \text{ for } i = (L+1)\ldots 2L .$$

7. An appropriate observation function $h^*(\mathbf{x})$ is composed from the complete observation function by selecting the sensors having new data since the last IMU timestep.

8. The update sigma points are fed to h^*, using $\mathbf{x} = \hat{\mathbf{x}} - \delta\mathbf{x}$ where the sigma points $\varsigma_{u,i}$ are substituted for $\delta\mathbf{x}$:

$$\gamma_i(l+1\,|\,l) = h^*\left(\hat{\mathbf{x}}(l+1\,|\,l) - \varsigma_{u,i}(l+1\,|\,l)\right) , \text{ for } i = 0\ldots 2L .$$

9. The result is recombined to yield the predicted measurement and predicted measurement covariance:

$$\mathbf{z}(l+1\,|\,l) = \sum_{i=0}^{2L} W_s(i)\,\gamma_i(l+1\,|\,l) ,$$

$$P_{zz} = \sum_{i=0}^{2L} W_c(i)\left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right]\left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right]^T .$$

10. The UKF Kalman gain is computed as:

$$K_{l+1} = P_{xz}P_{zz}^{-1} ,$$

where the state-measurement cross-covariance matrix is expressed as:

$$P_{xz} = \sum_{i=0}^{2L} W_c(i)\left[\varsigma_{u,i}(l+1\,|\,l) - \delta\mathbf{x}(l+1\,|\,l)\right]\left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right]^T .$$

11. The error state update equation using $\mathbf{z}(l+1)$ is:

$$\delta\mathbf{x}(k+1\,|\,k+1) = \delta\mathbf{x}(k+1\,|\,k) + K_{l+1}\left(\mathbf{z}(l+1) - \mathbf{z}(l+1\,|\,l)\right) .$$

12. The updated covariance is:

$$P(k+1\,|\,k+1) = P(k+1\,|\,k) - K_{l+1}P_{zz}K_{l+1}^T .$$

13. At this point the updated error state $\delta\mathbf{x}(k+1\,|\,k+1)$ is fed back to the estimated full state $\hat{\mathbf{x}}$ which has been propagated using IMU data to timestep $k+1$ since the last update at timestep $k-n$:

$$\hat{\mathbf{x}}(k+1\,|\,k+1) = \hat{\mathbf{x}}(k+1\,|\,k-n) - \delta\mathbf{x}(k+1\,|\,k+1) .$$

Subsequently, the error state must be reset to zero: $\delta\mathbf{x}(k+1\,|\,k+1) \Rightarrow 0$. The covariance $P(k+1\,|\,k+1)$ retains its value however, since it represents the uncertainty in $\delta\mathbf{x}$ which is the same as the uncertainty in $\hat{\mathbf{x}}$.

4 Simulation Environment

The simulation environment has been designed in a modular fashion to allow different configurations to work with common parts. For testing during development there is a Software-in-the-Loop (SiL) model which uses logged state data and prerendered images from a selection of scenarios to generate further sensor outputs including realistic signal delays. Figure 4 shows an image from the lunar surface rendering. The surface model is based on data from the Kaguya [9] mission.

The results in this work are generated using the SiL configuration with the following characteristics:

- the IMU sensor model has a bias drift of 1°/hr, 300 ppm scale factor stability and $0.03^\circ/\sqrt{\text{hr}}$ white noise running at 100 Hz update rate,
- the star tracker model has 5 arcsec angular accuracy, 5 Hz update rate and variable delay (jitter) of up to 0.2 s,
- the surface camera has a resolution of 1024×1024 pixels, $40^\circ \times 40^\circ$ field of view and produces 30 frames per second. Including the Lucas-Kanade tracker, the delay for optical flow may be up to 0.2 s depending on the number of features to be tracked which was capped at 40,
- the laser altimeter is accurate to 5 meter with 0.5 meter bias and an update rate of 0.5 Hz.

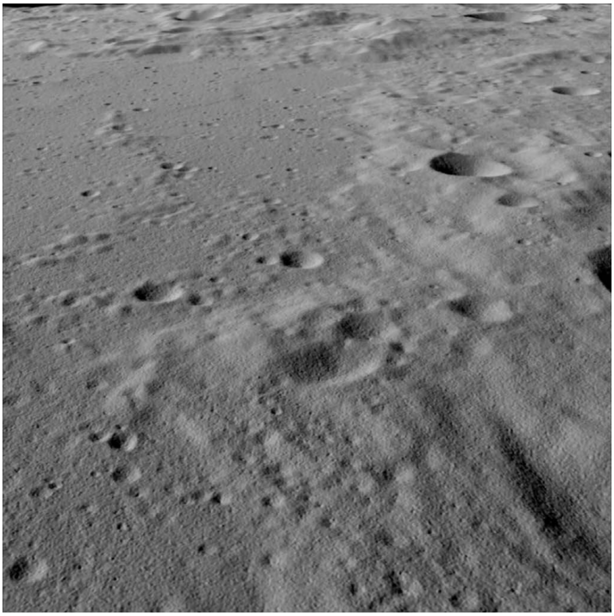


Fig. 4 An example image of the surface camera simulation

5 Results

The scenario presented here starts at 700 km downrange from the landing zone at an altitude of about 10 km. The spacecraft travels at 1700 m/s at that point. The trajectory is shown in Fig. 5. It is assumed that at this point the absolute crater navigation does not find enough matching craters in its database anymore and hands over its estimated state to the relative navigation. To assess the expected error accumulation due to the relative velocity navigation mode, the true state has been taken as initial state. Due to the fact that the sensor signals don't give direct positioning information, an initial position error would not contribute to the error growth during this phase of the landing.

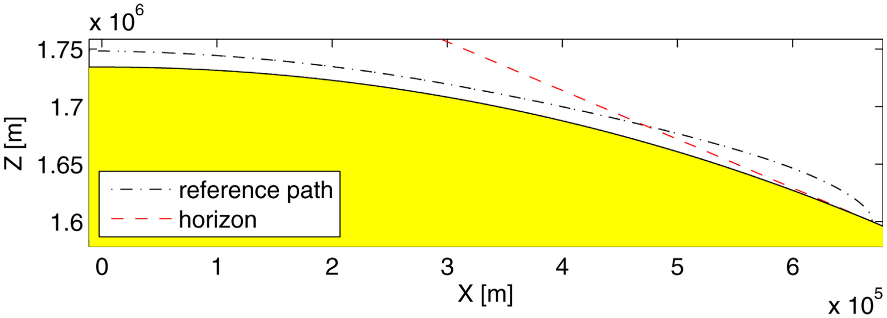
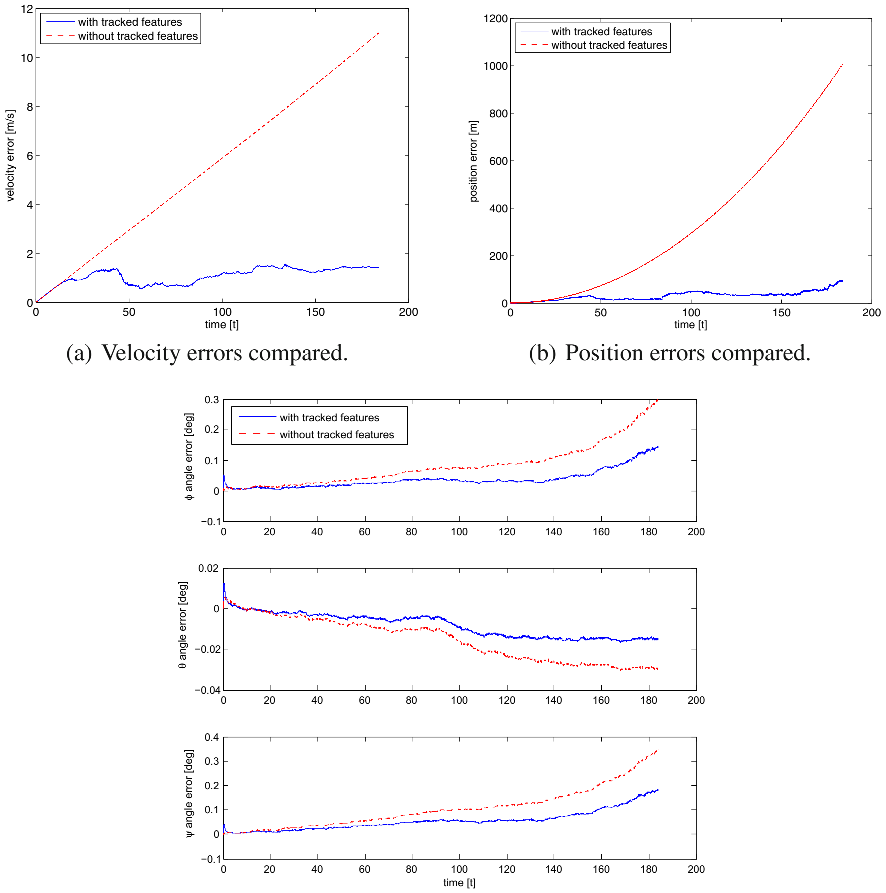


Fig. 5 Spacecraft trajectory from scenario start to landing



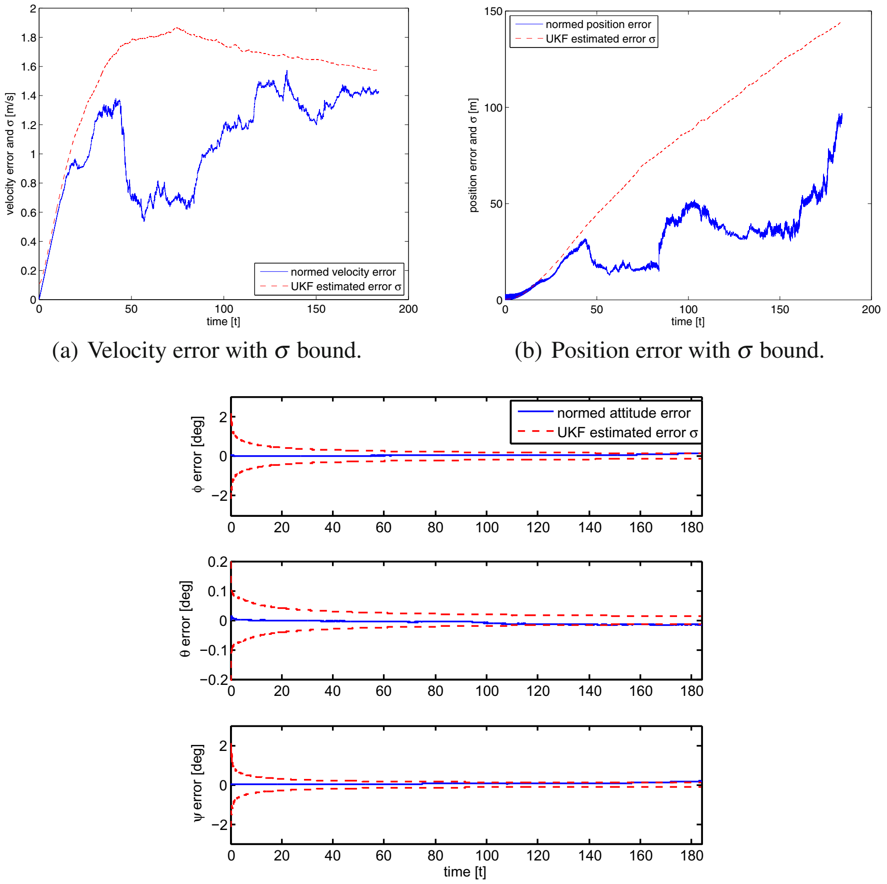
(a) Velocity errors compared.

(b) Position errors compared.

(c) Attitude errors compared.

Fig. 6 Resulting errors for a powered descent trajectory with and without surface camera

The same scenario has been flown with and without the surface camera to show the precision gained from using optical flow. With the surface camera and laser altimeter turned off, the remaining sensors are the IMU and star tracker. The results in Fig. 6 show the position, velocity and attitude errors comparing the two cases. The results without the surface camera show a typical error growth expected for an inertial navigation system. The velocity error increases linearly in time and the position quadratically. This can be attributed to the accelerometer biases which cannot be estimated in this case. Attitude is mainly driven by the star tracker, but benefits from optical flow as well. (Note that Euler angles have been used here merely for interpretation purposes: the filter works with quaternions and the axis-angle vector internally as described in subsection 2.1 and section 3.) The relative velocity optical navigation limits the velocity error to about 1.4 m/s. Although the position error is



(a) Velocity error with σ bound.

(b) Position error with σ bound.

(c) Attitude error with σ bound.

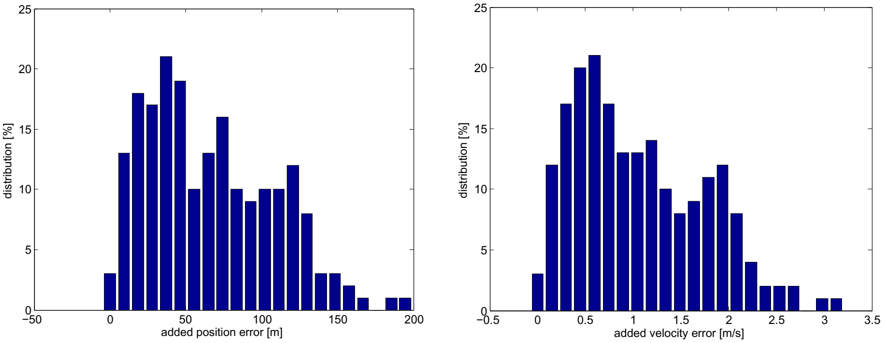
Fig. 7 Standard deviation error bounds estimated by the UKF for the case with surface camera

still unbounded, its growth is also significantly reduced. Figure 7 shows the case with surface camera including the 1σ-bounds derived from the auto covariances in P estimated by the UKF. The errors remain below this bound showing that the filter produces a consistent estimate. From the development of σ in time one can further conclude that the velocity and attitude are observable as σ converges. The position is not observable as may be expected when only surface features with unknown position are available.

To investigate the effects of initial position and velocity errors a Monte Carlo simulation with 100 runs was performed. The initial errors were normally distributed. Together with the resulting added errors their statistical properties are listed in Table 1.

Table 1 Monte Carlo Analysis: statistical properties of the errors

	initial position error [m]	initial velocity error [m/s]	added position error [m]	added velocity error [m/s]
mean	95	0.4	65	1.07
standard deviation	32	0.5	42	0.68



(a) Position error distribution.

(b) Velocity error distribution.

Fig. 8 Monte Carlo Analysis: resulting error distributions

The added errors are the normed additional position and velocity errors at the end of the relative velocity navigation phase. Their distributions are shown in Fig. 8.

6 Conclusion

The results so far show a significant position error growth reduction when optical flow and star tracker sensors are used to update the IMU state predictions. An important element of the navigation filter is the way it takes the processing delay for the optical data into account. Although currently the filter implementation is single threaded, the algorithm lends itself to a dual threaded structure, allowing fast processing of IMU data and enough processing time for the optical updates including delay compensation.

It should be noted that the crater navigation accuracy will degrade gradually and relative navigation will work in parallel for part of the descent. This combined navigation mode is planned as part of the ATON project, but the current paper concentrates on the evaluation of relative velocity navigation only. Also, the actual landing site will be autonomously selected based on LIDAR and camera image processing. This process starts as soon as the landing site crosses the horizon. From then on, navigation will continue relative to the selected landing site. This marks the endpoint for the relative velocity navigation mode. The main goal is to provide slow

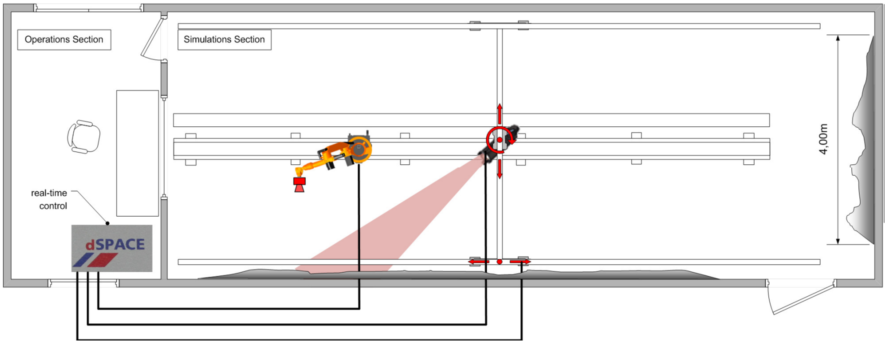


Fig. 9 The TRON facility can generate planetary approach images using an industrial robotic arm, scaled relief models and special lighting

Fig. 10 DLR's pilotless helicopter superARTIS shown at the ILA 2012 in Berlin

position error growth in the gap between crater navigation and landing site relative navigation. The results in Table 1 show that the position error can be expected to be 65 m with a standard deviation of 42 m under the chosen sensor characteristics and navigation errors at the start of the relative phase.

Future work will include demonstration and validation of Technology Readiness Level 4 (TRL4) using a Hardware-in-the-Loop (HiL) configuration which will use the TRON Facility [10], see Fig. 9, as well as flight tests using superARTIS, see Fig. 10.

Acknowledgements. The author would like to thank his colleagues in the ATON project team for their work and fruitful cooperation in building the simulation environment and especially Franz Andert for his help building the feature tracker. Furthermore, he is indebted to the referees for their careful reviews and helpful suggestions.

References

1. Brady, T., Bailey, E., Crain, T., Paschall, S.: ALHAT System Validation. In: 8th International ESA Conference on Guidance, Navigation & Control Systems, Karlovy Vary, Czech Republic. ESA (June 2011)
2. Farrenkopf, R.L.: Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators. Journal of Guidance and Control 1(4), 282–284 (1978)
3. Flandin, G., Polle, B., Frapard, B., Vidal, P., Philippe, C., Voirin, T.: Vision based navigation for planetary exploration. In: Advances in the Astronautical Sciences, vol. 133, pp. 277–296 (February 2009)
4. Frapard, B., Mancuso, S.: Vision Navigation for European Landers and the NPAL Project. In: 6th International ESA Conference on Guidance, Navigation and Control Systems, Loutraki, Greece. ESA (October 2005)
5. Johnson, A.E., Ansar, A., Matthies, L.H., Trawny, N., Mourikis, A.I., Roumeliotis, S.I.: A General Approach to Terrain Relative Navigation for Planetary Landing. In: Infotech, Rohnert Park, California, United States. AIAA (May 2007)
6. Julier, S., Uhlmann, J.K.: Handbook of Multisensor Data Fusion. In: Data Fusion in Nonlinear Systems, ch. 13. CRC Press LLC (2001)
7. Julier, S.J., Uhlmann, J.K.: A New Extension of the Kalman Filter to Nonlinear Systems. In: Proceedings of SPIE, vol. 3068, pp. 182–193 (1997)
8. Julier, S.J., Uhlmann, J.K.: Unscented Filtering and Nonlinear Estimation. Proceedings of the IEEE 92(3), 401–422 (2004)
9. Kato, M., Sasaki, S., Takizawa, Y.: The kaguya mission overview. Space Science Reviews 154, 3–19 (2010), 10.1007/s11214-010-9678-3
10. Krüger, H., Theil, S.: TRON - Hardware-in-the-loop test facility for lunar descent and landing optical navigation. In: 18th IFAC Symposium on Automatic Control in Aerospace. IFAC (2010)
11. Lee, D., Pernicka, H.: Vision-Based Relative State Estimation Using the Unscented Kalman Filter. International Journal of Aeronautical and Space Sciences 12(1), 24–36 (2011)
12. Li, S., Cui, P., Cui, H.: Autonomous Navigation and Guidance for Landing on Asteroids. Aerospace Science and Technology 10(3), 239–247 (2006)
13. Li, S., Cui, P., Cui, H.: Vision-aided Inertial Navigation for Pinpoint Planetary Landing. Aerospace Science and Technology 11(6), 499–506 (2007)
14. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: International Joint Conference on Artificial Intelligence, vol. 3, pp. 674–679 (August 1981)
15. Mourikis, A.I., Trawny, N., Roumeliotis, S.I., Johnson, A.E., Ansar, A., Matthies, L.: Vision-aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. IEEE Transactions on Robotics 25(2), 264–280 (2009)
16. Roumeliotis, S.I., Sukhatme, G.S., Bekey, G.A.: Smoother based 3D Attitude Estimation for Mobile Robot Localization. In: IEEE International Conference on Robotics and Automation, Detroit, MI, USA, pp. 1979–1986 (May 1999)
17. Schaub, H., Junkins, J.L.: Analytical Mechanics of Space Systems. AIAA Education Series (2003) ISBN 1-56347-563-4

18. Steffes, S.: Extended Kalman Filter for HNS Navigator. Technical report, DLR (May 2011)
19. Striepe, S.A., Epp, C.D., Robertson, E.A.: Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) Project Status as of May 2010. In: International Planetary Probe Workshop 2010, Barcelona, Spain (June 2010)
20. Verveld, M.J., Chu, Q.P., de Wagter, C., Mulder, J.A.: Optic Flow Based State Estimation for an Indoor Micro Air Vehicle. In: Guidance, Navigation, and Control Conference and Exhibit, Toronto, Ontario Canada. AIAA (August 2010)

Toward an Autonomous Lunar Landing Based on Low-Speed Optic Flow Sensors

Guillaume Sabiron, Paul Chavent, Laurent Burlion, Erwan Kervendal, Eric Bornschlegl, Patrick Fabiani, Thibaut Raharijaona, and Franck Ruffier

Abstract. For the last few decades, growing interest has returned to the quite challenging task of the autonomous lunar landing. Soft landing of payloads on the lunar surface requires the development of new means of ensuring safe descent with strong final conditions and aerospace-related constraints in terms of mass, cost and computational resources. In this paper, a two-phase approach is presented: first a biomimetic method inspired from the neuronal and sensory system of flying insects is presented as a solution to perform safe lunar landing. In order to design an autopilot relying only on optic flow (OF) and inertial measurements, an estimation method based on a two-sensor setup is introduced: these sensors allow us to accurately estimate the orientation of the velocity vector which is mandatory to control the lander's pitch in a quasi-optimal way with respect to the fuel consumption. Secondly a new low-speed Visual Motion Sensor (VMS) inspired by insects' visual systems performing local angular 1-D speed measurements ranging from $1.5^\circ/\mathrm{s}$ to $25^\circ/\mathrm{s}$ and weighing only 2.8 g is presented. It was tested under free-flying outdoor

Guillaume Sabiron · Thibaut Raharijaona · Franck Ruffier
Aix-Marseille Université,
Biorobotic Dept. of ISM, CNRS, ISM UMR 7287,
13288, Marseille cedex 09, France
e-mail: {Guillaume.Sabiron,Thibaut.Raharijaona,
Franck.Ruffier}@univ-amu.fr

Guillaume Sabiron · Paul Chavent · Laurent Burlion · Patrick Fabiani
ONERA - The French Aerospace Lab, F-31055 Toulouse, France
e-mail: {Guillaume.Sabiron,Paul.Chavent,
Laurent.Burlion,Patrick.Fabiani}@onera.fr

Erwan Kervendal
Astrium Satellites, Toulouse, France
e-mail: Erwan.Kervendal@astrium.eads.net

Eric Bornschlegl
European Space Agency ESTEC, 2200 AG Noordwijk, The Netherlands
e-mail: Eric.Bornschlegl@esa.int

conditions over various fields onboard an 80 kg unmanned helicopter. These preliminary results show that the optic flow measured despite the complex disturbances encountered closely matched the ground-truth optic flow.

1 Introduction

Landing on extraterrestrial bodies is still a critical stage in any exploration missions. A safe and soft landing is therefore required even though the task is way harder than on the Earth. On the Moon, the lack of atmosphere (acting as a natural brake or allowing the use of a parachute as on Mars [6]) and also the lack of usual sensing systems such as Global Positioning System (GPS) increase the complexity. On top of that, the time delay introduced by Earth to Moon communications forces the landing to be autonomous and based on robust and reliable sensors. Furthermore, sharp constraints in terms of embedded mass led us to design a very lightweight biologically inspired sensor that measures the visual angular velocity (in °/s) of the images sweeping backward across the visual field which is known as the optic flow. The visual cues seem to be a promising way to achieve autonomous lunar landing. Recently, several studies have shown various visual techniques such as LIDAR (LIght Detection And Ranging) techniques [36, 37] or other vision based navigation systems to estimate position and velocity parameters [7, 11, 16, 25, 34, 41, 47, 50], to perform hazard avoidance [48] or to control unmanned spacecraft [23, 24, 51]. In [51], the optic flow regulation principle [45] was applied to autonomous lunar landing problems using a feedback loop and tested by performing simulations on PANGU software (Planet and Asteroid Natural scene Generation Utility) developed for ESA by the University of Dundee which is a tool used to simulate visual environment of planetary surfaces(see [9, 38] for more information). In [24], based on numerical simulations, optimal trajectories were calculated in terms of the duration of the landing phase or the fuel consumption while keeping the OF constant. In [32], a fully optic flow-based visual servo control system was developed, in which a large visual field was combined with a centroid in order to estimate the direction of the speed vector in the case of small aerial robotic vehicles. In the extended Kalman filter (EKF) method described in [34], both of the above visual approaches were combined with an Inertial Measurement Unit (IMU), and accurate estimates of the lander's terrain-relative position, attitude, and velocity were obtained. In the preliminary strategy presented here, we focus on means to estimate from optic flow signals useful information such as the orientation of the velocity vector. Finding means of sensing the optic flow onboard unmanned aerial and terrestrial vehicles has been a key research topic during the last few decades. Several flight control systems based on optic flow cues have been constructed so far for performing hazardous tasks such as hovering and landing on a moving platform [21], avoiding obstacles [1, 3, 20], following terrain [18] and tracking a moving target [28]. Insects are capable of impressive flying behavior thanks to the built-in abilities they have developed and improved over several hundred millions of years, despite their small size and hence limited neural resources. Based on the findings obtained at our Laboratory on the fly's visual sensory system [12], several versions of the 2-pixel Local Motion Sensor (LMS) [10, 13, 14, 44, 46] were developed, using an algorithm introduced by [5, 39], which was later called the ”time of travel scheme” (see [2, 33]). Several vision-based systems have been previously designed to measure the optic flow onboard UAVs (Unmanned Aerial Vehicles) [8, 19, 22] and in particular in the range experienced during lunar landing [20, 28, 52]. Most of these visual systems were quite demanding in terms of their computational requirements and/or their weight or were not very well characterized, except for the optical mouse sensors [4], with which a standard error of approximately $\pm 5^\circ/s$ around $25^\circ/s$ was obtained in the case of an optical mouse sensor measuring motion in a $\pm 280^\circ/s$ overall range. However, to our knowledge, very few studies have been published so far in which optic flow systems have been implemented and tested outdoors onboard an unmanned aircraft subject to vibrations, where the illuminance cannot be easily controlled (see [1] in the case of linear 1-D motion sensors and see [18, 20, 27, 49] in that of 2-D optic flow sensors). A particular effort has been made in this study to cope the sensor's measurement range $[1.5^\circ/s; 25^\circ/s]$ with the one experienced during a lunar landing approach phase approximately of $[2^\circ/s; 6^\circ/s]$. It therefore seemed to be worth testing the reliability of the present 1-D optic flow-based visual sensor on a platform featuring the most similar conditions as on a spacecraft during lunar landing in terms of vibration dynamics and optic flow measurement range. Finally the sensor has been embedded onboard a free-flying helicopter called ReSSAC (ReSSAC stands in French for *Recherche et Sauvetage par Système Autonome Coopérant)* and tested in terms of its resolution, accuracy, sensitivity over a series of trees. This paper is organized as follows. The intended control strategy, the reference descent trajectory and the basic equations of the lunar lander dynamics allowing the estimation of the orientation of the velocity vector are described in Sect. 2. Section 3 gives a brief description of the new 1-D visual motion device, outlines the processing algorithm and the optical/electrical assembly, and finally presents the results of the outdoor experiments.

2 Lunar Landing Using Bio-inspired Measurements

Here, the principle of the control problem studied in this work is introduced. The landing scenario is presented along with initial and final states constraints. Then, the future approach of the biologically inspired feedback control loops is discussed.

2.1 Autonomous Lunar Landing Strategy

Lunar landing trajectory has been divided into four different phases in [15] (see Fig. 1):

1. De-orbit Phase,
2. Approach Phase,
3. Final Descent,
4. Free Fall.

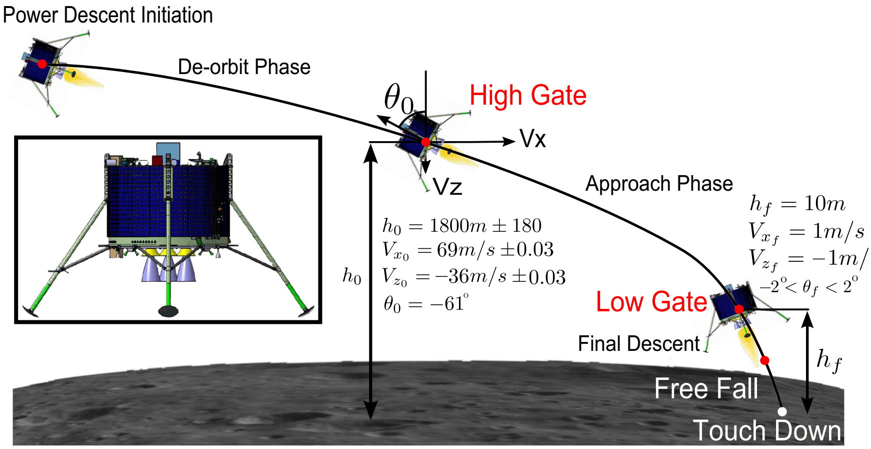


Fig. 1 Reference trajectory for lunar landing and 3D representation of the lunar module (courtesy: Astrium). The landing part addressed in this work is the approach phase defined between the high gate (1800 m ± 10%AGL) and the low gate (10 m AGL). The objectives of the lander is to reach the low gate (10 m high) with both vertical and horizontal velocities lower than 1 m/s and a pitch angle in the range ±2°. Modified from [26].

In this research work, a solution of the autonomy problem of the approach phase defined from the high gate (1800 m $\pm 10\%$ Above Ground Level -AGL-) to the low gate (10 m AGL) is studied. High gate corresponds to the height from which the landing site becomes visible from the spacecraft vision system. Low gate corresponds to the height from which the visual contact with the landing site is no longer available due to the dust raised by the thrusters. Initial parameters are a horizontal speed $V_{x_0} = 69 \pm .03$ m/s, a vertical speed $V_{z_0} = -36 \pm .03$ m/s, a pitch angle $\theta_0 = -61^{\circ}$, a ground height $h_0 = 1800 \pm 180$ m and a mass $m_{ldr_0} = 762 \pm 11$ kg (see Fig. 1). This reference trajectory is thus very similar to the Apollo test case scenario used in [23, 24, 51]. The considered solution features demanding terminal constraints at the low gate ($h_f = 10$ m) which are the following:

- $V_{x_f} \leq 1$ m/s,
- $V_{z_f} \geq -1$ m/s,
- $|\theta_f| < 2^{\circ}$.

We do not introduce position accuracy since the intended landing strategy is not aiming at a pinpoint or even at a precision landing but only at a soft landing. The objectives are thus listed in terms of velocity and attitude. The position on the x-axis is left free and concerning the altitude, the low gate will eventually be reached since the vertical velocity stays negative. In the current approach, the propellant mass consumption should be decrease as much as possible by the autonomous lunar landing strategy. The main challenge is that the entire state vector is not available from the measurements. For instance, velocities and position are neither measured nor estimated, only accelerations, angular rates, attitude, mass and optic flow are measured and thus available to feed the controllers. To land softly on the moon the autopilot should be able to reduce the velocity vector magnitude and to control the velocity vector orientation. This could be achieved by acting jointly on the lander's pitch and the lander's main thrust, the two available control signals. In [24, 51], authors have shown the substance of the pitch control law in the design to achieve optimal performances since the system is underactuated. In the solution currently under investigation, the main idea is to design an autopilot that keeps the main thrust antiparallel to the velocity vector orientation, in order to minimize fuel consumption. This principle states the pitch angle of reference θ_{ref} to be fed into the controller:

$$\theta_{ref} = -\gamma - \frac{\pi}{2} \tag{1}$$

Where γ denotes the flight path angle (angle between the orientation of the speed vector and the local horizontal) as described in Fig. 2.

2.2 Lander's Dynamic Modeling and Optic Flow Equations

The autopilot under consideration consists mainly of an optic flow-based control system operating in the vertical plane (e_x,e_z), which control the spacecraft's main thruster force and pitch angle. To stabilize the lander, it is necessary to cope with non-linearities and the inherent instability. Since there is no atmosphere on the

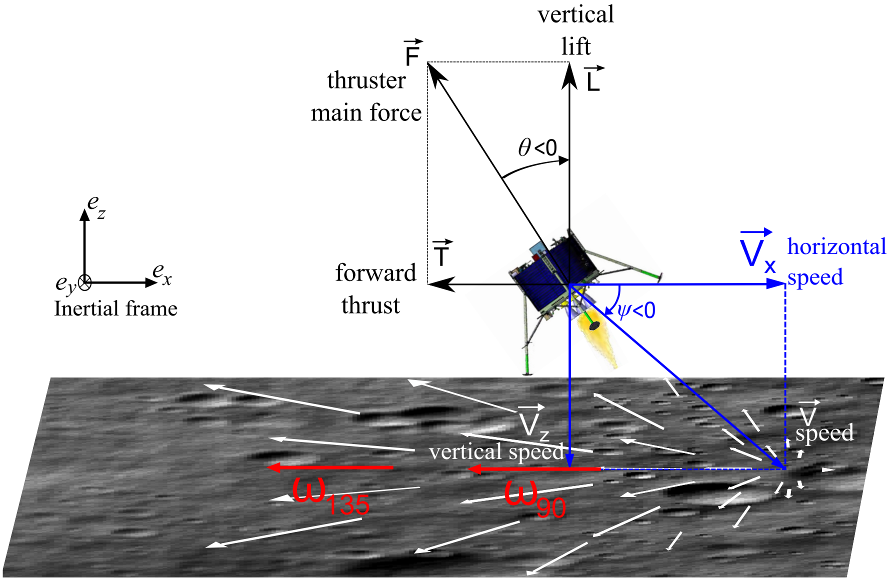


Fig. 2 Diagram of the lander, showing its speed vector V and the mean thruster force F

Moon: no friction, wind or drag forces are applied to the lander. In the present model, the heave and surge dynamics are coupled via the lander's pitch (see Fig. 2). It is worth noting that it is not suitable using the scenario described in Fig. 1 to measure ω_{45} (where the subscript denotes the elevation angle: the angle between the gaze direction and the local horizontal) to determine the direction of the velocity vector, since it is near the focus of expansion where the motion is always null (see Fig. 2). To include physical constraints into the model, the following assumption is adopted:

$$(\mathbf{H1}) \begin{cases} \text{The thrusters can produce only positive forces and the maximum} \\ \text{thrust is limited to 3820 N, which means } 0 \leq u_{th} \leq 3820 \text{ N.} \end{cases}$$

The dynamic motion of the lander can be described in the time domain by the following dynamic system in the inertial frame (e_x, e_y, e_z):

$$\begin{cases} a_{ldr_z}(t) = \dfrac{\cos(\theta(t))}{m_{ldr}(t)} u_{th}(t) - g_{Moon} \\ a_{ldr_x}(t) = \dfrac{\sin(\theta(t))}{m_{ldr}(t)} u_{th}(t) \end{cases} \tag{2}$$

Where $u_{th} = \|F\|$ corresponds to the control force applied to the lander, $a_{ldr_{x,z}}$ are the lander's accelerations in the lunar inertial reference frame, m_{ldr} stands for the lander's mass, θ is the pitch angle, t denotes the time and g_{Moon} the lunar gravity constant ($g_{Moon} = 1.63$ m/s^2). The lander's mass depends directly on the fuel consumption, as given by the following relation:

$$\dot{m}_{ldr} = \frac{-1}{I_{sp} \cdot g_{Earth}} u_{th}(t), \tag{3}$$

where $I_{sp} = 311$ s corresponds to the specific impulse and $g_{Earth} = 9.81$ m/s^2 to the Earth's gravity. This means that:

$$m_{ldr}(t) = m_{ldr}(t_0) - \frac{1}{I_{sp} \cdot g_{Earth}} \int_{t_0}^{t} u_{th}(\varepsilon) \mathrm{d}\varepsilon \tag{4}$$

where $m_{ldr}(t_0) = 762$ kg is the lander's mass at high gate level. Since the initial mass is known and the lander's mass depends linearly on the integral of the lander's thruster control signal, the mass can be computed and assessed at any time during the simulated descent. The inner pitch control system is modeled as follows:

$$\frac{I}{R} \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = u_{pitch}(t) \tag{5}$$

u_{pitch} is the control input signal acting on the spacecraft's pitch, θ is measured via an IMU, I the moment of inertia of the lander and R the eccentricity of the thrusters from the center of mass. In this study, the well-known problem of error growth in IMU is not taken into account, the attitude measurement are thus considered accurate all along the descent. Once the dynamic model of the spacecraft is defined, one need to state the optic flow equations to find what information can be deduced from this visual cue. The ground-truth optic flow $\omega_{grd-trh}$ can be described as the sum of the two distinct components defined by [29], i.e. the translational and rotational optic flow:

$$\omega_{grd-trh} = \omega_T + \omega_R \tag{6}$$

The translational optic flow ω_T depends on the linear velocity V expressed in the inertial frame associated with the vector basis (e_x,e_y,e_z), the distance from the ground D in the gaze direction and Φ, the angle between the gaze direction and the heading direction.

$$\omega_T = \frac{V}{D}\sin(\Phi) \tag{7}$$

The rotational optic flow ω_R depends only on the angular speed Ω_j expressed in the body fixed frame, where j denotes the axis of rotation, and on the elevation angle λ between the gaze direction and the axis of rotation which is always $\frac{\pi}{2}$ in the 2D case (see [43] for a graphical illustration).

$$\omega_R = \Omega_j \sin(\lambda) \tag{8}$$

Finally the general equation of the optic flow is as follows:

$$\omega_{grd-trh} = \frac{V}{D}\sin(\Phi) + \Omega_j \sin(\lambda) \tag{9}$$

From the previous equation and regarding hazardous terrain avoidance, one can see that as soon as an obstacle appears in the line of sight of the sensor, the distance D will be reduced/increased whether it is a boulder or a crater. This will cause the autopilot to react by decelerating or by accelerating accordingly ensuring the safety of the lander no matter the topography of the terrain.

2.3 Optic Flow Measurements ω_{90} and ω_{135} for Velocity Vector Orientation Estimation and Control

In order to be able to strongly reduce the lander's speed during the approach phase, one needs to act both on the pitch angle and on the magnitude of the thrust as explained in Sect. 2.1. Since γ is required for the pitch angle reference signal θ_{ref} generation and is not measured, one needs to find a way to estimate it by the means of the available sensors such as optic flow sensors and IMU. The main question is how to fuse different visual angular speed measurements to obtain useful information about unavailable measurements of the state vector. It is straightforward to note that the optic flow cue is related to the orientation of the velocity vector. Therefore, under the assumption that the sensors are embedded on a gimbal system one can derive the orientation angle γ from optic flow sensors positioned in different directions. From (9), under the assumption of a practically flat ground (i.e. $D = h/\cos(\frac{\pi}{2} - \Phi + \gamma)$, $-\Phi + \gamma$ denotes the angle between the gaze direction and the local horizontal), and gimbaled mounted sensors (i.e. the camera rotation with respect to an inertial reference frame is kept to zero thanks to the attitude measurement provided by the IMU):

$$\omega_{90} = \frac{V_x}{h} \tag{10}$$

with $V = V_x/\cos(\gamma)$ and h the ground height,

$$\omega_{135} = \frac{V}{h/\cos(\pi/4)}\frac{\sqrt{2}}{2}\left(\cos(\gamma) - \sin(\gamma)\right) = \frac{\omega_{90}}{2}\left(1 - \tan(\gamma)\right) \tag{11}$$

Finally one can obtain:

$$\tan(\gamma) = 1 - 2\frac{\omega_{135}}{\omega_{90}} \tag{12}$$

It is worth noting that both the horizontal and vertical dynamics are expressed in $\tan(\gamma)$ with $\tan(\gamma) = \frac{V_z}{V_x}$. Thanks to (12) featuring only visual information, a pitch controller based on optic flow and pitch measurement θ provided by the IMU could be designed through (5) in order to ensure the collinearity between the lander's main thruster force and its velocity vector orientation. The low speed visual motion sensors are thus the cornerstones of this autonomous lunar landing strategy. Since the optic flow controller and the pitch controller are based on the output signals of the optic flow sensors, it seemed to be worth testing the reliability of this sensor in real-life conditions. This is the purpose of the following section.

3 VMS-Based Optic Flow Measurements Obtained Onboard ReSSAC

As presented in the previous section, the control signals (pitch and thrust) which depend mainly on the optic flow measurement are strongly linked to the precision, robustness, sensitivity of the low speed visual motion sensors. That is why we had to develop and test a new VMS dedicated to angular speed measurements that cope with the range experienced during lunar landing, which is a range of low optic flow. Conveniently, the optic flow range experienced during lunar landing corresponds roughly to the range experienced with a high scale helicopter UAV which has been used to test the sensor on Earth.

3.1 Bio-inspired Optic Flow Processing

A mandatory step in the maturation of a technology is to design and embed the previously simulated device on a real-life complex system. In order to validate the feasibility of the work performed in simulation using low speed VMS in Sect. 2, an experimental approach is presented. A low resolution visual motion sensor based on six pixels array and dedicated to low speeds has been developed to demonstrate the feasibility of measuring the 1-D local angular speed on a lunar landing like scenario on Earth. We tested the sensor onboard the ReSSAC unmanned helicopter to validate the bio-inspired algorithm on such low speeds at high heights and with strong disturbances (i.e. helicopter vibrations, uncontrolled illuminance, rough terrain). This sensor is an updated version of the 2-pixels Local Motion Sensors based on neuromorphic findings [12] designed in our laboratory.

3.2 Presentation of the Low-Speed Visual Motion Sensor

The new low-speed visual motion sensor consists mainly of a low-cost plastic lens placed in front of an off-the-shelf photosensor array. The photosensor used in this study, which is called the LSC, was purchased from iC-Haus: it features six photodiodes, each having a large sensitive area of 300 × 1600 μm and an integrated preamplifier. The LSC conveys the visual signals received to a hybrid analog/digital processing algorithm, where the optic flow value ω_{meas} is computed. The cheap, lightweight lens used here was a CAX183 from Thorlabs (focal length 18.33 mm, f-number 4.07). A custom-made protective case was added in order to protect the low-weight sensor and the optical assembly from unfavorable weather conditions (see Fig. 3.a for pictures and Fig. 3.b for an exploded view). The new visual motion sensor and its custom-made protective case weighed 29.4 g. Many of the parameters of the original visual motion detecting scheme presented in [5, 39] have been updated, especially in terms of the optical angles and the cut-off frequency of the temporal filters. The six optical axes formed by the photodiodes are separated by an interreceptor angle $\Delta\varphi$ see Fig. 4. By defocusing the lens (i.e., by adjusting the distance between the lens and the photosensors), we obtained Gaussian angular sensitivity functions for each photoreceptor with a correlation coefficient greater than 99% ($R^2_{LSC} > 0.990$), in line with what occurs in the common fly's eye [17]. These features were assessed by slowly rotating the lens in front of a point light source placed at a distance of 85 cm. The local 1-D angular speed ω_{meas} measured by the sensor was defined as the ratio between the interreceptor angle $\Delta\varphi$ and the time elapsing Δt between the moments when two adjacent photodiode signals reach the threshold (i.e., the time of travel of a contrast from the optical axis of one photodiode to the optical axis of the following one).

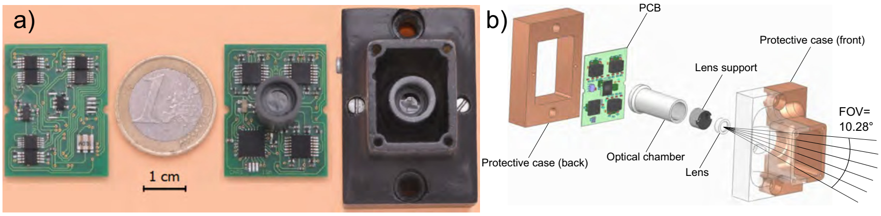


Fig. 3 (a) Top and bottom view of the electronic board (size: 33 × 40 mm) of a low-speed visual motion sensor with its lens mounted on the LSC photosensor array. The custom-made protective case is presented on the right. (b) Exploded view of the complete assembly, including the custom-made protective case (front and back), the electronic board, and the optical assembly (lens, lens support, optical chamber).

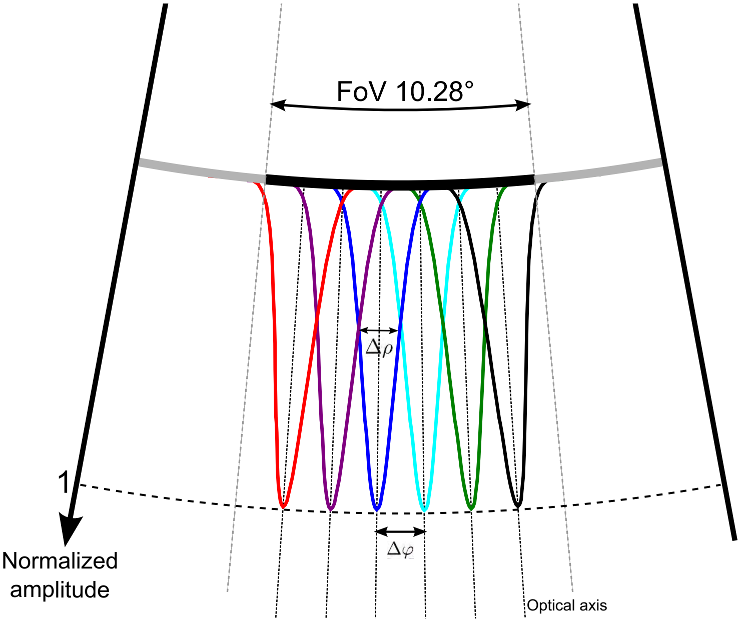


Fig. 4 Approximate Gaussian angular sensitivity functions of the LSC photosensor array with a CAX183 plastic aspherical lens. By adjusting the distance between the lens and the LSC photosensor, we obtained a correlation coefficient almost equal to 1 ($R^2_{LSC} > 0.990$), and a $\Delta\varphi$ value approximately equal to $\Delta\rho$.

$$\omega_{meas} = \frac{\Delta\varphi}{\Delta t} \tag{13}$$

In [10], the measurement range of the sensor covered a large range of high speeds from $50^\circ/\mathrm{s}$ to $300^\circ/\mathrm{s}$, whereas the present study focused on low velocities giving a range of $1.5^\circ/\mathrm{s}$ to $25^\circ/\mathrm{s}$, which is more than tenfold slower. In order to stay in the same range of Δt, whose accuracy of measurement depends on the microcontroller's sampling frequency, we therefore had to narrow $\Delta\varphi$. $\Delta\varphi$ corresponds to the angle separating two adjacent photodiodes optical axis: it depends on the focal lens, on the pitch (distance between the center of two adjacent photodiodes) and also on the distance from the photodiode plane to the focal point which is the easiest setting to adjust. The large 18.33 mm focal length increases the defocalizing effects of the lens, giving a suitably small mean interreceptor angle of $\overline{\Delta\varphi} = 1.488^\circ$. The second advantage of the defocusing process is that it adds a blurring effect giving Gaussian-shaped angular sensitivity functions. As found to occur in some diurnal insects [30],

$$\Delta\varphi = \Delta\rho \tag{14}$$

Achieving a tight $\Delta\rho$ made it possible for the sensor to respond to higher spatial frequency contrasts. The acceptance angle, defined by $\Delta\rho$, acts like an

Table 1 Characteristics of the new low-speed visual motion sensor

Parameter	Value
Focal length of the lens CAX183 (mm)	18.33
f_{number} of the lens (#)	4.07
Angular velocity range (°/s)	[1.5; 25]
Field of view of a single photodiode (°)	2.90×15.44
Sensitivity (°/s/LSB)	4.58 e-5
Mean interreceptor angle $\overline{\Delta\varphi}$ (°)	1.488
Mean acceptance angle $\overline{\Delta\rho}$ (°)	1.448
Photodiode size (µm)	$300 \times 1,600$
Pixel pitch (µm)	420
Resolution (°/s) [min; max]	[0.01; 0.21]
Mass of the visual motion sensor in a stand-alone version (g)	2,8

optical low pass spatial filter. We eventually reached $1.4^\circ \geq \Delta\varphi \approx \Delta\rho < 1.5^\circ$, corresponding to a field of view in the direction of the visual motion sensor of 10.28° $\left(\sum_{i=1}^{5} \Delta\varphi_i + \frac{\Delta\varphi_1}{2} + \frac{\Delta\varphi_5}{2}\right)$. Table 1 gives the optical characteristics of the sensor. The general processing algorithm consists of two parts: an analog processing part converts the six visual signals into electrical signals with a high signal to noise ratio, and the digital processing part then simultaneously computes five optic flow values plus the median value (see Fig. 5). The analog processing begins with a programmable gain connected to the microcontroller via a SPI communication bus [46]. A passband filter then differentiates the visual signal and acts as an anti-aliasing filter. The digital processing algorithm starts with a second order fixed-point notch filter centered on the ReSSAC's main rotor frequency. The center frequency of the filter is $f_0 = 13.8Hz$ with a Q-factor $Q = 6.9$ at a sampling frequency $f_s = 500\ Hz$. Its transfer function, which has been defined in [35], is as follows:

$$H_{notch}(z) = b\frac{1 - 2\cos(\omega_0)\,z^{-1} + z^{-2}}{1 - 2b\cos(\omega_0)\,z^{-1} + (2b-1)z^{-2}} \tag{15}$$

with

$$b = \frac{1}{1 + \frac{\sqrt{1-G_B^2}}{G_B}\tan\left(\frac{\Delta\omega}{2}\right)}$$

where $\Delta\omega$ is the full width at a level G_B^2 and ω_0 is the center frequency. We chose $\omega_0 = 2\pi\frac{f_s}{f_0}$, $\Delta\omega = 2\pi\frac{\Delta f}{f_s}$ with $\Delta f = 2\ Hz$ and $G_B^2 = -3\ dB$. As the visual angular speed ω_{meas} is quite low, the temporal frequency f_t of the visual signal (which consists of contrasts) is also quite low, as expressed by the following equation [31]:

$$f_t = \omega_{meas} \times f_{spatial} \tag{16}$$

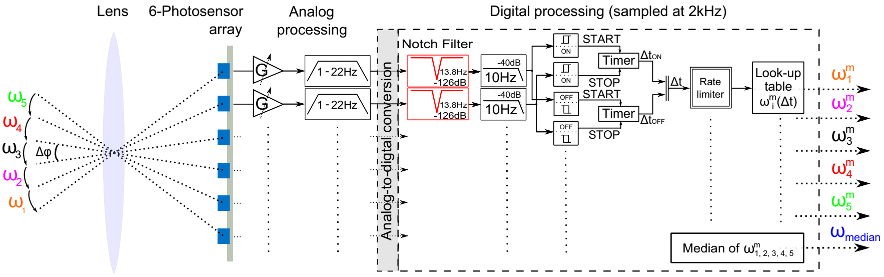


Fig. 5 General processing architecture of the low-speed visual motion sensor. First of all, the spatial sampling and low-pass filtering steps are carried out by the defocused lens. The six photodiode signals are amplified by a programmable gain in order to increase the signal to noise ratio, before being filtered by an analog bandpass filter ($1-22\ Hz$). The digital stage begins with a second order fixed-point notch filter centered on the main rotor frequency of ReSSAC, $13.8\ Hz$. It is followed by a second order fixed-point low pass filter with a cut-off frequency set at $10\ Hz$. A hysteresis thresholding process is associated with the computation of the time Δt elapsing between two adjacent signals (with either ON or OFF contrasts). Lastly, after an outlier filtering step, the output signal of the 1-D visual motion sensor is obtained from a precomputed look-up table and the median value is calculated.

where $f_{spatial}$ is the spatial frequency (in $cycles/^{\circ}$) associated with the contrasting pattern.

Therefore, a second order fixed-point low pass filter was used to enhance the signal to noise ratio by removing the noise remaining at frequencies of more than $10\ Hz$. The algorithm called the "Time of travel scheme" implemented here consists mainly of a hysteresis thresholding process with separate ON and OFF pathways [5, 39, 40, 42, 45] followed by the Δt computation, the result of which is fed into a corresponding table. Lastly, the five simultaneously computed optic flows ω_i^m are combined by the median operator in order to increase the robustness and the refresh rate of the output [40]. The microcontroller used for this purpose is a dsPIC33FJ128GP802 working at a sampling frequency of $2\ kHz$, except for the digital filters, which are sampled at a rate of $500\ Hz$. Special efforts were made to optimize the algorithm, and a computational load of only 17% was eventually obtained.

3.3 Free-Flying Results with ReSSAC

The characteristics of the present visual motion sensor (VMS) were assessed by performing optic flow measurements under controlled motion conditions (orientation and velocity) outdoors. Pure rotational motion was applied to the sensor with angular speed variations ranging from $1^{\circ}/s$ to $20^{\circ}/s$ using a previously described outdoor set-up [10]. The triangular response pattern obtained corresponds closely to the reference angular speed (see Fig. 6). It can therefore be said that this new tiny sensor is able to accurately compute the 1-D visual angular speed during a rotational

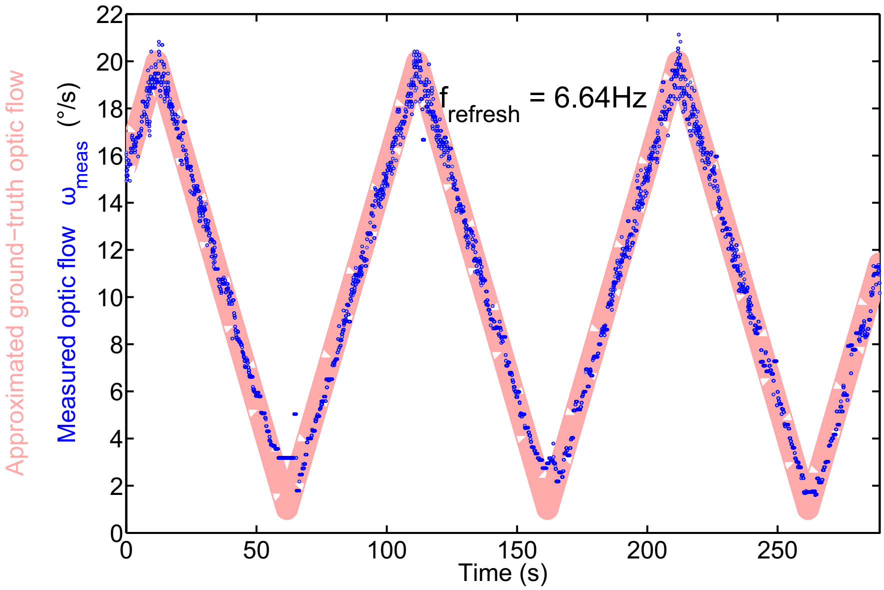


Fig. 6 Dynamic outdoor response of the low-speed VMS (blue), as compared with the ground-truth optic flow (red). The visual motion sensor was rotated by means of a conveyor belt driven by a stepping motor (103H5208-0440 from Sanyo-Denki) [10]. Rotations from $1°/\mathrm{s}$ to $20°/\mathrm{s}$ were applied to the sensor, which is designed to operate in the $1.5°/\mathrm{s}$ to $25°/\mathrm{s}$ range. The optic flow measured closely matched the reference signal, with a refresh rate of 6.64 Hz. Since no synchronization signal was available, the ground-truth optic flow has been roughly synchronized here.

motion within its operating range. The refresh rate is defined as the ratio between the total number of new measurements of each ω_i occurring within the acceptable range $[1.5°/\mathrm{s} - 25°/\mathrm{s}]$ and the time elapsing. The median value is delivered at 2 kHz (output data rate) even if the measure is not refreshed, that is why the refresh rate metric is needed to evaluate the performance of the sensor. The mean refresh rate achieved during the dynamic performances evaluation was $f_{refresh} = 6.64$ Hz: this value depends on the richness of the visual environment, as well as on the actual angular speed. The low-speed VMS performances were then studied on a six degrees of freedom UAV during free flight over fields. The ONERA's ReSSAC unmanned helicopter was used to test the sensor's dynamic responses. The characteristics of ReSSAC (Yamaha RmaX) in terms of the mass balance have been described in [53]. Its mass, its flight envelope and the vibration dynamics due to the main rotor's rotational speed presented us with quite a challenging ground-truth optic flow profile. The flight was performed in South-western France in mid-July around 5pm on a bright sunny day: the mean illuminance was approximately 10000 lx. Another tests should be performed on special setup at ESA or Astrium facilities to be sure that the sensor would respond similarly in a moon-like environment in terms of robustness

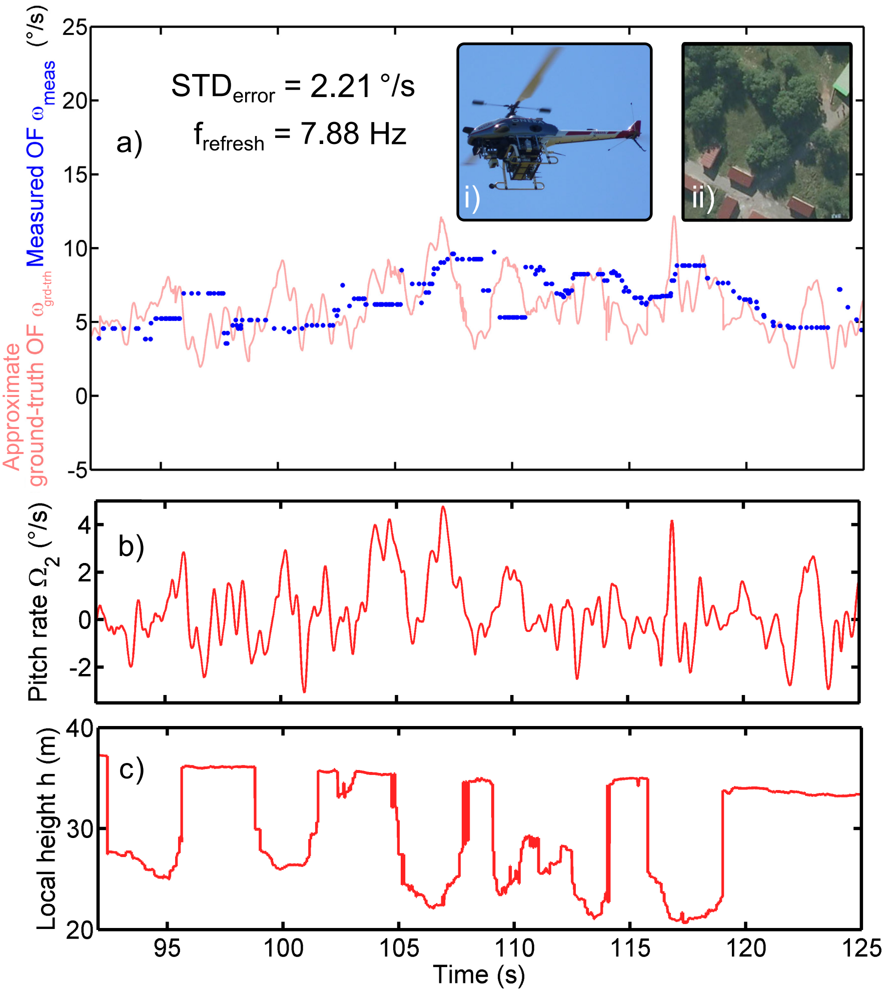


Fig. 7 Low-speed visual motion sensor and flight data sensed on-board the ReSSAC UAV. (a) Approximate ground-truth optic flow (see (17)) (red) and measured optic flow $\omega_{meas} = \Delta\varphi/\Delta t$ (blue dots). Despite the strong variations mainly due to vibrations, the low-speed visual motion sensor's output closely matched the approximate ground-truth optic flow, giving a standard deviation of 2.21°/s and a refresh rate of 7.88 Hz. The effects of strong variations in the local height due to the successive trees and houses are directly reflected in the low-speed VMS measurement signal. (i) ReSSAC unmanned helicopter in-flight (ii) Aerial view of the flight environment obtained on `http://geoportail.fr`. (b) Pitch angular rate of ReSSAC as measured by the IMU. (c) Local ground height measured by combining GPS data and previously mapped LIDAR data. The nominal height was around 40 m. But due to the variable relief, the local height often changed suddenly by 15 meters.

to moon-like contrasts and illuminance. Onboard ReSSAC, the 1-D local optic flow measured is subject to several variations as follows. Since the roll and pitch angles are small during the whole flight, the distance to the ground in the gaze direction D can be approximated as $D \approx h/\left(\cos(\varphi)\cos(\theta)\right)$, where φ denotes the roll angle, θ denotes the pitch angle and h denotes the local ground height. In our case, $\Phi = -\theta + \gamma + \frac{\pi}{2}$ (with the sensor oriented downward, $\gamma < 0$, $\theta < 0$), $\lambda = \frac{\pi}{2}$ and $\Omega_j = \Omega_2$, where Ω_2 is the pitch angular velocity defined in the body fixed reference frame, the ground-truth optic flow (see (9)) is therefore computed as described in (17).

$$\omega_{grd-trh} = \left(\frac{V}{h}\cos(\theta)\cos(\varphi)\sin\left(\theta + \gamma + \frac{\pi}{2}\right)\right) + \Omega_2 \tag{17}$$

During the experiment described below, the approximate ground-truth optic flow $\omega_{grd-trh}$ was computed using data from the IMU, the GPS (OEM4 G2 from NovAtel) and the data grabbed by a LIDAR (Sick LDMRS 400001) during previous flights over the same fields. We computed the ground-truth optic flow as precisely as possible but since the data are coming from cartographic data previously recorded and from different sensors with different accuracy as well as different noise sources: $\omega_{grd-trh}$ is the approximate ground-truth optic flow. The low speed visual motion sensor was embedded at the front end of ReSSAC pointing directly downward with a clear field of view. Fig. 7 shows the nice response of the low-speed visual motion sensor mounted onboard the unmanned ReSSAC helicopter. Despite the complex ground-truth optic flow, the visual motion sensor responded appropriately to the visual stimuli. The standard deviation of the error between the ground-truth optic flow $\omega_{grd-trh}$ and the measured optic flow ω_{meas} was less than $2.25°/\mathrm{s}$, which is quite low. The refresh rate $f_{refresh}$ was greater than 7.8 Hz, which is even slightly higher than in the dynamic measurements performed during a rotating motion on ground. Fig. 7.b, giving the pitch angular rate of ReSSAC and Fig. 7.c the local ground height shows how well the sensor responded to its visual environment and how complex is the visual motion combining at the same time erratic relief and pitch variations. Once again, the low-speed VMS accurately sensed these height variations and yielded similar values to the ground-truth value. The robust and accurate performances observed during this experiment show that the low-speed visual motion sensor is highly suitable for use in many high-scaled robotic applications.

4 Conclusion

In this paper, we gathered several emerging technologies to achieve an autonomous lunar landing based on low-speed optic flow sensors. The mathematical expression describing the dynamic model of the spacecraft as well as the optic flow fusion allowing some useful states estimation have been presented along with the intended control strategy based on two parallel control loops acting on the lander's pitch angle and main thruster. The main idea of this promising biologically inspired approach is to avoid the direct measurement of the height or velocity. We intend to design the thrust control law using optic flow measurements while keeping the main thruster force collinear to the velocity vector in a concern of mass consumption optimization. We showed that with a sensor set-up based on two optic flow sensors pointing downward in different directions (90° and 135° from the local horizontal), we obtained a direct measurement of γ the orientation of the velocity vector allowing us to generate a reference signal for the pitch angle controller. The next step will be to develop the full simulations featuring two control loops to achieve at low gate the final conditions stated in the reference scenario presented in this paper. The loop controlling the main thruster will be designed using optic flow cue. Secondly the gimbal setup used in simulations needs to be addressed. Since the main benefits of this minimalistic optic flow technology is the weight efficiency and simplicity, a gimbal system is then not suited for this purpose. Increasing the number of VMS and thus enlarging the sensory field of view is a potential way to achieve such challenge. The extension of the solution to a real-life, 3D setup, is an interesting objective to be addressed. Furthermore, during the flight on ReSSAC, even if the yaw and roll angle are considered small, they exist during the whole flight without deteriorating the measurements performances. Once the strategy of the lunar landing was introduced, we presented a new lightweight visual motion sensor able to compute accurately the optic flow in the range experienced during a quasi-optimal approach phase of a lunar landing. This new VMS has been developed, and then tested both on the ground and in flight onboard a 80 kg unmanned helicopter called ReSSAC over an unknown complex outdoor environment and under real-life dynamic and vibratory conditions. Encouraging results of this experiment showed that this sensor is perfectly suited for aeronautics or aerospace applications since it sensed accurately the local 1-D angular speed ranging from $1.5^\circ/\mathrm{s}$ to $25^\circ/\mathrm{s}$ with a quite frequently refreshed measurement. In future work emphasis will be placed on the lunar landing simulations using Matlab/Simulink© and the PANGU software using the $\tan(\gamma)$ estimation method and the control scheme presented here.

Acknowledgements. We are most grateful to N. Franceschini, S. Viollet, G. Graton, F. Roubieu and F. Expert for their fruitful suggestions and comments during this study. We thank P. Mouyon, H. de Plinval, A. Amiez and V. Fuertes for the successful flight of ReSSAC, T. Rakotomamonjy and A. Piquereau for their expertise in helicopter dynamics, M. Boyron for his assistance with the electrical design and J. Diperi for his involvement in the mechanical design. We would like to thank the two anonymous referees for providing us with constructive comments and suggestions. This research work is co-funded by European Space Agency and Astrium Satellites under ESA's Networking/Partnering Initiative program (NPI) for advanced technologies for space.

References

1. Barrows, G., Neely, C.: Mixed-mode VLSI optic flow sensors for in-flight control of a Micro Air Vehicle. In: SPIE: Critical Technologies for the Future of Computing, pp. 52–63 (2000)
2. Benson, R., Delbrück, T., Touretzky, D.S.: Direction selective silicon retina that uses null inhibition. In: Adv. Neural Inf. Process. Syst., vol. 4, pp. 756–763. Morgan Kaufman, San Mateo (1992)

3. Beyeler, A., Zufferey, J., Floreano, D.: OptiPilot: control of take-off and landing using optic flow. In: European Micro Aerial Vehicle Conference (EMAV) (2009)
4. Beyeler, A., Zufferey, J.C., Floreano, D.: Vision-based control of near-obstacle flight. Auton. Robot. 27, 201–219 (2009)
5. Blanes, C.: Appareil Visuel élémentaire pour la navigation à vue d'un robot mobile autonome, Master's thesis (Advisor: N. Franceschini), Neurosciences, Univ. Aix-Marseille II (1986)
6. Braun, R., Manning, R.: Mars exploration entry, descent and landing challenges. In: IEEE Aerosp. Conf., Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, Big Sky, Montana (2006)
7. Cheng, Y., Ansar, A.: Landmark Based Position Estimation for Pinpoint Landing on Mars. In: IEEE Int. Conf. Robot. Autom. (ICRA), pp. 1573–1578 (2005)
8. Conroy, J., Gremillion, G., Ranganathan, B., Humbert, J.: Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. Auton. Robot. 27, 189–198 (2009)
9. Dubois-Matra, O., Parkes, S., Dunstam, M.: Testing and Validation of Planetary Vision-based navigation systems with PANGU. In: 21st International Symposium on Space Flight Dynamics, ISSFD (2009)
10. Expert, F., Viollet, S., Ruffier, F.: Outdoor Field Performances of Insect-Based Visual Motion Sensors. J. Field Robot. 28, 529–541 (2011)
11. Flandin, G., Polle, B., Frapard, B., Vidal, P., Philippe, C., Voirin, T.: Vision Based Navigation for Planetary Exploration. In: 32nd Annual AAS Rocky Mountain Guidance and Control Conference (2009)
12. Franceschini, N., Riehle, A., Nestour, A., Stavenga, D., Hardie, R.: Directionally selective motion detection by insect neurons. In: Facets of Vision, pp. 360–390. Springer (1989)
13. Franceschini, N., Pichon, J., Blanes, C.: From insect vision to robot vision. Philos. T. R. Soc. Lond. 337, 283–294 (1992)
14. Franceschini, N., Ruffier, F., Serres, J.: A Bio-Inspired Flying Robot Sheds Light on Insect Piloting Abilities. Curr. Biol. 17, 329–335 (2007)
15. Frapard, B., Champetier, C., Kemble, S., Parkinson, B., Strandmoe, S., Lang, M.: Vision-Based GNC Design for the LEDA Mission. In: ESA 3rd International Conference on Spacecraft GNC (1996)
16. Frapard, B., Polle, B., Flandin, G., Bernard, P., Vétel, C., Sembely, X., Mancuso, S.: Navigation for Planetary Approach and Landing. In: 5th International ESA Conference on GNC, p. 159 (2002)
17. Götz, K.: Optomotorische untersuchung des visuellen systems einiger Augenmutanten der fruchtfliege Drosophila. Biol. Cybern. 2, 77–92 (1964)
18. Garratt, M., Chahl, J.: Vision-Based Terrain Following for an Unmanned Rotorcraft. J. Field Robot. 25, 284–301 (2008)
19. Green, W., Oh, P., Barrows, G.: Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. In: IEEE Int. Conf. Robot. Autom. (ICRA), vol. 3, pp. 2347–2352 (2004)
20. Griffiths, S., Saunders, J., Curtis, A., Barber, B., McLain, T., Beard, R.: Maximizing miniature aerial vehicles. IEEE Robot. Autom. Mag. 13, 34–43 (2006)
21. Hérissé, B., Hamel, T., Mahony, R., Russotto, F.-X.: Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow. IEEE T. Robot. 28(1), 77–89 (2012)
22. Hrabar, S., Sukhatme, G., Corke, P., Usher, K., Roberts, J.: Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In: IEEE Int. Conf. Intell. Rob. Syst. (IROS), pp. 3309–3316 (2005)

23. Izzo, D., de Croon, G.: Landing with time-to-contact and ventral optic flow estimates. J. Guid. Control Dynam. 35(4), 1362–1367 (2011)
24. Izzo, D., Weiss, N., Seidl, T.: Constant-Optic-Flow Lunar Landing: Optimality and Guidance. J. Guid. Control Dynam. 34, 1383–1395 (2011)
25. Janschek, K., Tchernykh, V., Beck, M.: Performance Analysis for Visual Planetary Landing Navigation Using Optical Flow and DEM matching. In: AIAA Guidance, Navigation and Control Conference and Exhibit (2006)
26. Jean-Marius, T., Strandmoe, S.E.: Integrated vision and navigation for a planetary lander, ESA, Estec. Technical report, AEROSPATIAL, Espace et Défense, Les Mureaux, France (1998)
27. Kendoul, F., Fantoni, I., Nonamib, K.: Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. Robot. Auton. Syst. 57, 591–602 (2009)
28. Kendoul, F., Nonami, K., Fantoni, I., Lozano, R.: An adaptive vision-based autopilot for mini flying machines guidance, navigation and control. Auton. Robot. 27, 165–188 (2009)
29. Koenderink, J., Doorn, A.: Facts on optic flow. Biol. Cybern. 56, 247–254 (1987)
30. Land, M.: Visual Acuity In Insects. Annu. Rev. Entomol. 42, 147–177 (1997)
31. Landolt, A., Mitros, A.: Visual sensor with resolution enhancement by mechanical vibrations. Auton. Robot. 11(3), 233–239 (2001)
32. Mahony, R., Corke, P., Hamel, T.: A Dynamic Image-Based Visual Servo Control Using Centroid and Optic Flow Features. J. Dyn. Sys., Meas., Control 130(1), 1–12 (2008)
33. Moeckel, R., Liu, S.-C.: Motion Detection Circuits for a Time-To-Travel Algorithm. In: IEEE Int. Symp. Circ. S., pp. 3079–3082 (2007)
34. Mourikis, A.I., Trawny, N., Roumeliotis, S.I., Johnson, A.E., Ansar, A., Matthies, L.: Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. IEEE Trans. Robot. 25(2), 264–280 (2009)
35. Orfanidis, S.J.: Introduction to signal processing. Prentice-Hall Inc., Upper Saddle River (1995)
36. Parkes, S., Silva, V.: GNC sensors for planetary landers: a review. In: Data Systems in Aerospace (DASIA), pp. 1–9 (2002)
37. Parkes, S., Dunstan, M., Matthews, D., Martin, I., Silva, V.: LIDAR-based GNC for Planetary Landing: Simulation with PANGU. In: Harris, R.A. (ed.) Data Systems in Aerospace (DASIA), Prague, Czech Republic, p. 18.1 (2003)
38. Parkes, S., Martin, I., Dunstan, M.: Planet Surface Simulation with PANGU. In: 8th International Conference on Space Operations, pp. 1–10 (2004)
39. Pichon, J.-M., Blanes, C., Franceschini, N.: Visual guidance of a mobile robot equipped with a network of self-motion sensors. In: Wolfe, W.J., Chun, W.H. (eds.) SPIE Conf. on Mobile Robots IV, pp. 44–53 (1989)
40. Roubieu, F., Expert, F., Boyron, M., Fuschlock, B., Viollet, S., Ruffier, F.: A novel 1-gram insect based device measuring visual motion along 5 optical directions. In: IEEE Sens. Conf., pp. 687–690 (2011)
41. Roumeliotis, S., Johnson, A., Montgomery, J.: Augmenting inertial navigation with image-based motion estimation. In: IEEE Int. Conf. Robot. Autom. (ICRA), pp. 4326–4333 (2002)
42. Ruffier, F., Viollet, S., Amic, S., Franceschini, N.: Bio-inspired optical flow circuits for the visual guidance of micro air vehicles. In: IEEE Int. Symp. Circ. S. (ISCAS), pp. 846–849 (2003)

43. Ruffier, F.: Pilote automatique biomimétique. Système générique inspiré du contrôle visuomoteur des insectes pour: le suivi de terrain, la réaction au vent et l'atterrissage automatiques d'un micro-aéronef. PhD thesis, Institut National Polytechnique de Grenoble (2004) (in French)
44. Ruffier, F., Franceschini, N.: Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In: IEEE Int. Conf. Robot. Autom. (ICRA), pp. 2339–2346 (2004)
45. Ruffier, F., Franceschini, N.: Optic flow regulation: the key to aircraft automatic guidance. Robot. Auton. Syst. 50, 177–194 (2005)
46. Ruffier, F., Expert, F.: Visual motion sensing onboard a 50-g helicopter flying freely under complex VICON-lighting conditions. In: International Conference on Complex Medical Engineering, pp. 634–639 (2012)
47. Shang, Y., Palmer, P.: The dynamic motion estimation of a lunar lander. In: 21st International Symposium on Space Flight Dynamics (ISSFD), pp. 1–14 (2009)
48. Strandmoe, S., Jean-Marius, T., Trinh, S.: Toward a vision based autonomous planetary lander. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, AIAA-99-4154 (1999)
49. Tchernykh, V., Beck, M., Janschek, K.: An Embedded Optical Flow Processor for Visual Navigation using Optical Correlator Technology. In: IEEE Int. Conf. Intell. Rob. Syst. (IEEE/RSJ), Beijing, pp. 67–72 (2006)
50. Trawny, N., Mourikis, A.I., Roumeliotis, S.I., Johnson, A.E., Montgomery, J.: Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. J. Field Robot. 24, 357–378 (2007)
51. Valette, F., Ruffier, F., Viollet, S., Seidl, T.: Biomimetic optic flow sensing applied to a lunar landing scenario. In: IEEE Int. Conf. Robot. Autom. (ICRA), pp. 2253–2260 (2010)
52. Watanabe, Y., Fabiani, P., Le Besnerais, G.: Simultaneous visual target tracking and navigation in a GPS-denied environment. In: Int. Conf. Adv. Robot. (ICAR), pp. 1–6 (2009)
53. Watanabe, Y., Lesire, C., Piquereau, A., Fabiani, P., Sanfourche, M., Le Besnerais, G.: The ONERA ReSSAC Unmanned Autonomous Helicopter: Visual Air-to-Ground Target Tracking in an Urban Environment. In: American Helicopter Society 66th Annual Forum (2010)

Online Estimation of Mean Orbital Elements with Control Inputs

Weichao Zhong and Pini Gurfil

Abstract. Estimating the mean orbital elements is essential for satellite orbit determination as well as guidance and autonomous orbital transfer. Whereas offline estimation of mean elements can be performed using batch processing and analytical satellite theories, online estimation requires recursive filtering. This paper proposes a unique formulation for mean orbital elements estimation, wherein the semianalytical theory is used for generating both the process and measurement equations, but the mean elements estimation is performed using an Unscented Kalman Filter. A comprehensive performance evaluation for both controlled and uncontrolled orbits shows the potential applicability of the method and its advantages compared to Brouwer-based approaches.

1 Introduction

In many cases of practical interest, satellite guidance and orbit control laws utilize mean orbital elements, rather than osculating elements, as inputs. This pertains both for open-loop and closed-loop guidance and control [1,2]. The use of mean elements – which are usually defined as a single-period-averaged osculating elements – has clear benefits, including reduced sensitivity to high-frequency content and short-periodic oscillations, which implies much lower fuel consumption. It makes more sense to have the satellite track an orbit defined by mean elements than have it respond to the fast variations of the osculating elements. This observation holds both for single- and multiple-satellite missions [3].

Weichao Zhong
Harbin Institute of Technology, Harbin 150001, China
e-mail: zhongweichaohit@gmail.com

Pini Gurfil
Technion – Israel Institute of Technology, Haifa 32000, Israel
e-mail: pgurfil@technion.ac.il

With the onset of autonomous satellite missions, capable of performing closed-loop autonomous orbital transfer [4], the importance of efficient on-board mean elements estimation becomes evident. Whereas in traditional satellite missions the mean elements can be estimated on the ground and uploaded to the satellites within the visibility slots, autonomous missions must be capable of producing real-time mean element estimates based on measurements obtained from the on-board sensors – typically GPS receivers.

In general, there are three main alternatives for estimating the mean orbital elements. The first is to use an analytical theory, such as the Brouwer artificial satellite theory [5], the Kozai theory [6], the drag-extended Brouwer theory [7], or newer theories, including those discussed in Refs. [8–15] to name only a few. The second alternative is to use batch processing to convert from osculating to mean elements using least-square-type approaches. Both of these alternatives could be problematic: Brouwer-type theories are sensitive to noise and modeling errors and cannot easily accommodate thrust; and batch processing is not adequate for real-time on-board implementation as it requires data accumulation for at least a complete orbital period (in addition, the actual orbital period changes, which may introduce additional errors).

The third alternative, which is the focus of the current work, is to use recursive filtering to estimate the mean elements. This approach has not received much attention in the literature, and there are only a handful of works that consider on-board mean elements estimation from osculating elements measurements [4]. This problem is not trivial, since it is difficult to write a good dynamical model for mean element propagation, formulate a mapping from mean elements back to osculating elements to get a measurement-type equation, and in addition find a filter that would be capable of properly processing the highly-nonlinear state and measurement equations.

In this work, a new approach for on-board estimation of the mean elements from osculating element measurements is proposed. First, a semianalytical dynamical model that includes zonal/tesseral/sectorial harmonics and drag is formulated to capture the daily, long-periodic, and secular evolution of the mean orbital elements. Because there is a clear tradeoff between precision and complexity, the semianalytical model is truncated to include the control inputs, the long-periodic and secular terms up to $J_4/C_{33}/S_{33}$ and exponential drag. The mapping from mean to osculating elements, which is used as a measurement equation, is obtained by adding the short-periodic effects of drag and zonal/tesseral/sectorial harmonics to the mean elements. This unique formulation is then matched up with a square-root unscented Kalman filter (SR-UKF) [16] that serves as the mean elements estimator. The sigma point selection is performed based on the spherical simplex (SS) algorithm [17], which significantly reduces the number of propagated sigma points, thus allowing for improved computational efficiency and possible real-time implementation. A comprehensive performance evaluation for both controlled and uncontrolled orbits shows the potential advantages of the proposed filtering approach vis-à-vis Brouwer-type methods.

2 Preliminaries

In an inertial reference frame, the equations of motion for the perturbed Keplerian two-body problem are written as

$$\ddot{\mathrm{r}}+\frac{\mu}{r^3}\mathrm{r}=\mathrm{F} \tag{1}$$

where $\mathrm{r}\in\mathbb{R}^3\backslash\{0\}$ is the position vector, $r=\|\mathrm{r}\|$ and F is the perturbation. The solution for the respective position and velocity vectors can be written as [18]

$$\mathrm{r}=\mathrm{f}_r\left[a(t),e(t),i(t),\Omega(t),\omega(t),M_0(t),t\right],\ \mathrm{v}=\mathrm{f}_v\left[a(t),e(t),i(t),\Omega(t),\omega(t),M_0(t),t\right] \tag{2}$$

where $\{a(t),e(t),i(t),\Omega(t),\omega(t),M_0(t)\}$ are the classical osculating orbital elements – semimajor axis, eccentricity, inclination, right ascension of the ascending node (RAAN), argument of perigee and mean anomaly at epoch, respectively. The variational equations for the classical elements are usually written in two forms. If $\mathrm{F}=\partial R/\partial \mathrm{r}$, for some perturbing potential R, then one obtains the Lagrange planetary equations (LPE) [18]

$$\frac{\mathrm{d}a}{\mathrm{d}t}=\frac{2}{\tilde{n}a}\frac{\partial R}{\partial M} \tag{3a}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t}=\frac{1-e^2}{\tilde{n}a^2e}\frac{\partial R}{\partial M}-\frac{\sqrt{1-e^2}}{\tilde{n}a^2e}\frac{\partial R}{\partial \omega} \tag{3b}$$

$$\frac{\mathrm{d}i}{\mathrm{d}t}=\frac{\cot i}{\tilde{n}a^2\sqrt{1-e^2}}\frac{\partial R}{\partial \omega}-\frac{1}{\tilde{n}a^2\sqrt{1-e^2}\sin i}\frac{\partial R}{\partial \Omega} \tag{3c}$$

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t}=\frac{1}{\tilde{n}a^2\sqrt{1-e^2}\sin i}\frac{\partial R}{\partial i} \tag{3d}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t}=\frac{\sqrt{1-e^2}}{\tilde{n}a^2e}\frac{\partial R}{\partial e}-\frac{\cot i}{\tilde{n}a^2\sqrt{1-e^2}}\frac{\partial R}{\partial i} \tag{3e}$$

$$\frac{\mathrm{d}M_0}{\mathrm{d}t}=-\frac{1-e^2}{\tilde{n}a^2e}\frac{\partial R}{\partial e}-\frac{2}{\tilde{n}a}\frac{\partial R}{\partial a} \tag{3f}$$

where $\tilde{n}=\sqrt{\mu/a^3}$ is the mean motion.

A more general formulation of the variational equations, which allows for non-conservative specific forces, yields the Gauss variational equations (GVE). In the GVE, the perturbation is represented in **RSW**, i.e., $\Delta\mathrm{F}=F_R\hat{\mathrm{R}}+F_S\hat{\mathrm{S}}+F_W\hat{\mathrm{W}}$. The resulting equations are [18]:

$$\frac{\mathrm{d}a}{\mathrm{d}t} = \frac{2}{\tilde{n}\sqrt{1-e^2}}\left(e\sin f\,F_R + \frac{\tilde{p}}{r}F_S\right) \tag{4a}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \frac{\sqrt{1-e^2}}{\tilde{n}a}\left(\sin f\,F_R + \left(\cos f + \frac{e+\cos f}{1+e\cos f}\right)F_S\right) \tag{4b}$$

$$\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{r\cos(\omega+f)}{\tilde{n}a^2\sqrt{1-e^2}}F_W \tag{4c}$$

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = \frac{r\sin(\omega+f)}{\tilde{n}a^2\sqrt{1-e^2}\sin i}F_W \tag{4d}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{\sqrt{1-e^2}}{\tilde{n}ae}\left(-\cos f\,F_R + \sin f\left(1+\frac{r}{\tilde{p}}\right)F_S\right) - \frac{r\cot i\sin(\omega+f)}{\sqrt{\mu\tilde{p}}}F_W \tag{4e}$$

$$\frac{\mathrm{d}M_0}{\mathrm{d}t} = \frac{1}{\tilde{n}a^2e}\left((\tilde{p}\cos f - 2er)\,F_R - (\tilde{p}+r)\sin f\,F_S\right) \tag{4f}$$

where $\tilde{p} = a\left(1-e^2\right)$ is the semilatus rectum and f is the true anomaly.

With $\alpha \triangleq [a(t), e(t), i(t), \Omega(t), \omega(t), M_0(t)]$ as a state vector, and a mapping of the form $t \mapsto f(t,\alpha)$, the LPE (3) and GVE (4) can be written in a unified form as

$$\dot{\alpha} = \mathrm{g}(\alpha,t) \tag{5}$$

In many cases of interest, such as satellite orbit control, guidance and orbit propagation, the osculating elements are replaced by mean elements. A common definition of the mean elements relies on the averaging operator, which, for some given vector-valued function $\mathrm{s}(t)$, is defined by

$$\bar{\mathrm{s}} = \langle \mathrm{s}(t)\rangle \triangleq \frac{1}{T}\int_0^T \mathrm{s}\,\mathrm{d}t = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{s}\frac{1}{\sqrt{1-e^2}}\left(\frac{r}{a}\right)^2\mathrm{d}f = \frac{1}{2\pi}\int_0^{2\pi}\mathrm{s}\,\mathrm{d}M \tag{6}$$

where M is the mean anomaly. Thus, the mean elements are obtained by

$$\bar{\alpha} = \langle \alpha(t)\rangle \triangleq \frac{1}{T}\int_0^T \alpha(t)\,\mathrm{d}t \tag{7}$$

where T denotes the orbital period.

An analytical mean-elements calculation scheme was proposed by Kozai [6] and Brouwer [5], who used the averaging technique to isolate the short-periodic terms, α_{short}, so that

$$\bar{\alpha}(t) = \alpha(t) - \alpha_{short}(t) \tag{8}$$

Substituting Eq. (8) into Eq. (5) yields

$$\dot{\bar{\alpha}}(t) + \dot{\alpha}_{short}(t) = \mathbf{g} = \bar{\mathrm{g}} + \mathrm{g}_{short} \tag{9}$$

In this formulation, $\dot{\bar{\alpha}}(t)$ consists of secular and long-periodic terms, namely

$$\bar{\mathrm{g}} = \mathrm{g}_{sec} + \mathrm{g}_{long} \tag{10}$$

where

$$g_{sec} = g_{sec}\left(\bar{a}, \bar{e}, \bar{i}\right), \, g_{long} = g_{long}\left(\bar{a}, \bar{e}, \bar{i}, \bar{\Omega}, \bar{\omega}\right), \, g_{short} = g_{short}\left(\bar{a}, \bar{e}, \bar{i}, \bar{\Omega}, \bar{\omega}, \bar{M}\right) = g - \bar{g} \tag{11}$$

In the following discussion, this mean element calculation scheme will be coupled with a recursive filtering algorithm to yield accurate and computationally-tractable estimation of the mean elements.

3 Modeling the Orbital Dynamics

In this section, the semianalytical astrodynamical models used for the propagation of the mean elements will be discussed, including zonal, tesseral and sectorial harmonics, as well as drag.

3.1 The Zonal Part of the Geopotential

The perturbing gravitational potential including zonal harmonics only is given by [19]:

$$R_{zonal} = -\frac{\mu}{r}\sum_{n=2}^{\infty} J_n \left(\frac{r_e}{r}\right)^n P_n\left(\sin\varphi\right) \tag{12}$$

where $J_n, n = 2, 3, \ldots$ are the zonal gravitational coefficients, φ is the latitude, $\sin\varphi = \sin i \sin u$, $u = \omega + f$ is the argument of latitude, r_e is Earth's mean equatorial radius and $P_n(x)$ denotes a Legendre polynomial of the first kind of order n, which is expressed as [20]

$$P_n(x) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^2 - 1\right)^n \tag{13}$$

It is customary to average R_{zonal} prior to the substitution into the LPE (3). Applying the averaging operator (6) gives:

$$\bar{R}_{zonal} = \frac{1}{2\pi}\int_0^{2\pi} R_{zonal}\frac{1}{\sqrt{1-e^2}}\left(\frac{r}{a}\right)^2 \mathrm{d}f = R_{sec} + R_{long} \tag{14}$$

To evaluate the integral in Eq. (14), the zonal potential should be written as a function of the classical orbital elements. This procedure can be found in [19, 21].

3.2 Second-Order Effects

As pointed out by Brouwer [5], due to the nonlinear nature of the variational equations, the short-periodic terms contribute both secular and long-periodic inputs of $\mathcal{O}(J_2^2)$.

The resulting short-periodic averaged potential, after transforming into the classical orbital elements, is given by

$$\bar{R}_{short} = \frac{3J_2^2 r_e^4 \mu}{128\eta^7 a^5} \left[5\eta^2 - 4\eta - 5 - \cos^2 i \left(18\eta^2 - 24\eta - 10\right) + \right. \\ \left. \cos^4 i \left(5\eta^2 + 36\eta + 35\right) + e^2 \left(30\cos^4 i - 32\cos^2 i + 2\right)\cos 2\omega\right] \tag{15}$$

where $\eta = \sqrt{1-e^2}$.

3.3 The Tesseral-Sectorial Part of the Geopotential

The dominant tesseral and sectorial harmonics (degree 2 and order 2) have a period of approximately half day, which is much longer than the orbital period [19, 22]. Hence, tesseral and sectorial harmonics may have a significant effect on the long-periodic dynamics. It is thus imperative to include the tesseral and sectorial harmonics in the mean elements dynamical model. The perturbing gravitational potential of Tesseral and sectorial harmonics is given by [19]:

$$R_{Tesseral} = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=1}^{n} \left(\frac{r_e}{r}\right)^n P_{nm}(\sin\varphi)\left[C_{nm}\cos m\lambda + S_{nm}\sin m\lambda\right] \tag{16}$$

where λ is the geographic longitude of the satellite measured eastward from the Greenwich meridian, C_{nm}, S_{nm} are harmonic coefficients and $P_{nm}(x)$ are the associated Legendre polynomials of degree n and order m, which are expressed as

$$P_{nm}(x) = \frac{1}{2^n n!}\left(1-x^2\right)^{\frac{m}{2}} \frac{\mathrm{d}^{n+m}}{\mathrm{d}x^{n+m}}\left(x^2-1\right)^n = \left(1-x^2\right)^{\frac{m}{2}} \frac{\mathrm{d}^m}{\mathrm{d}x^m} P_n(x) \tag{17}$$

The geopotential up to degree 3 and order 3 can be written down based on Eq. (16):

$$\begin{aligned} R_{22} &= 3\cos^2\varphi \frac{\mu}{r}\left(\frac{r_e}{r}\right)^2 \left(C_{22}\cos 2\lambda + S_{22}\sin 2\lambda\right) \\ R_{31} &= \frac{3}{2}\left(5\sin^2\varphi - 1\right)\frac{\mu}{r}\left(\frac{r_e}{r}\right)^3 \left(C_{31}\cos\lambda + S_{31}\sin\lambda\right) \\ R_{32} &= 15\sin\varphi\cos^2\varphi \frac{\mu}{r}\left(\frac{r_e}{r}\right)^3 \left(C_{32}\cos 2\lambda + S_{32}\sin 2\lambda\right) \\ R_{33} &= 15\cos^3\varphi \frac{\mu}{r}\left(\frac{r_e}{r}\right)^3 \left(C_{33}\cos 3\lambda + S_{33}\sin 3\lambda\right) \end{aligned} \tag{18}$$

Using the averaging method presented in Section 2, the long-periodic perturbing potentials of the tesseral and sectorial harmonics up to degree 3 and order 3 can be derived:

$$\begin{aligned}
R_{long,22} &= \frac{3}{2}\sin^2 i\left(1-e^2\right)^{-\frac{3}{2}}\frac{\mu}{a}\left(\frac{r_e}{a}\right)^2\left(C_{22}\cos 2\Omega_s + S_{22}\sin 2\Omega_s\right)\\
R_{long,31} &= \frac{3}{2}e\left(1-e^2\right)^{-\frac{5}{2}}\frac{\mu}{a}\left(\frac{r_e}{a}\right)^3\begin{pmatrix} C_{31}\left(\cos\Omega_s\cos\omega\left(\frac{5}{4}\sin^2 i-1\right)+\sin\Omega_s\sin\omega\cos i\left(1-\frac{15}{4}\sin^2 i\right)\right)\\ +S_{31}\left(\cos\Omega_s\sin\omega\cos i\left(\frac{15}{4}\sin^2 i-1\right)+\sin\Omega_s\cos\omega\left(\frac{5}{4}\sin^2 i-1\right)\right)\end{pmatrix}\\
R_{long,32} &= \frac{15}{4}e\left(1-e^2\right)^{-\frac{5}{2}}\sin^2 i\frac{\mu}{a}\left(\frac{r_e}{a}\right)^3\begin{pmatrix} C_{32}\left(\cos 2\Omega_s\sin\omega\left(3\sin^2 i-2\right)-2\sin 2\Omega_s\cos\omega\cos i\right)\\ +S_{32}\left(2\cos 2\Omega_s\cos\omega\cos i+\sin\Omega_s\sin\omega\left(3\sin^2 i-2\right)\right)\end{pmatrix}\\
R_{long,33} &= \frac{45}{4}e\left(1-e^2\right)^{-\frac{5}{2}}\frac{\mu}{a}\left(\frac{r_e}{a}\right)^3\begin{pmatrix} C_{33}\left(2\cos 3\Omega_s\cos\omega\cos i+\sin 3\Omega_s\sin\omega\left(3\sin^2 i-2\right)\right)\\ +S_{33}\left(2\sin 3\Omega_s\cos\omega\cos i+\cos 3\Omega_s\sin\omega\left(2-3\sin^2 i\right)\right)\end{pmatrix}
\end{aligned} \tag{19}$$

where $\Omega_s = \Omega - \theta$, and θ is the Greenwich sidereal angle.

3.4 Atmospheric Drag

The specific force due to atmospheric drag can be modeled as

$$\mathrm{F}_{drag} = -\frac{1}{2}\frac{SC_D}{m}\rho\left(\mathbf{v}-\mathbf{v}_{atm}\right)\left\|\mathbf{v}-\mathbf{v}_{atm}\right\| \tag{20}$$

where m is the satellite mass, S is the cross-sectional reference area, C_D is the drag coefficient defined with respect to the cross-sectional area, and ρ is the atmospheric density. The vector v_{atm} is the atmospheric velocity. If the atmosphere is assumed to be spherical and co-rotating with the Earth, then $\mathbf{v}_{atm} = \left[0\ 0\ \omega_e\right]^{\mathrm{T}} \times \mathrm{r}$.

The model used herein approximates the atmospheric density as [18]

$$\rho = \rho_0 \exp\left(\frac{r_{p_0} - r}{H}\right) \tag{21}$$

where ρ_0 is the atmospheric density at the initial perigee radius, r_{p_0}, and H is the density scale height of the atmosphere. The drag specific force vector $\mathbf{F}_{drag}$, written in terms of the osculating elements, can be expressed in the **NTW** frame as follows [23]:

$$\begin{aligned}
F_T &= -\frac{1}{2}K_1\tilde{n}^2a^2\rho\frac{1+2e\cos f+e^2}{(1-e^2)}\\
F_N &= 0\\
F_W &= -\frac{1}{2}K_2\tilde{n}a\rho r\cos(\omega+f)\sin i\left(\frac{1+2e\cos f+e^2}{(1-e^2)}\right)^{1/2}
\end{aligned} \tag{22}$$

where $K_1 = (C_D S_1/m\,)\,Q$, $K_2 = (C_D S_2/m\,)\,\omega_e\sqrt{Q}$, and $Q = \left(1 - r_{p_0}\omega_e\cos i/v_{p_0}\right)^2$. The variables S_1 and S_2 are the respective cross-sectional areas perpendicular to the tangential and subnormal directions, and v_{p_0} is the velocity at the initial perigee. To use the GVE (4), the drag components are first transformed from **NTW** into **RSW** using the transformation

$$
\begin{aligned}
F_R &= \frac{e\sin f}{\sqrt{1+2e\cos f+e^2}}F_T + \frac{1+e\cos f}{\sqrt{1+2e\cos f+e^2}}F_N \\
F_S &= \frac{1+e\cos f}{\sqrt{1+2e\cos f+e^2}}F_T - \frac{e\sin f}{\sqrt{1+2e\cos f+e^2}}F_N
\end{aligned} \tag{23}
$$

Substituting Eqs. (22) and (23) into Eq. (4) yields:

$$
\begin{aligned}
\dot{a}_{drag} &= -\frac{K_1\tilde{n}a^2}{\left(1-e^2\right)^{3/2}}\rho\left(1+2e\cos f+e^2\right)^{3/2} \\
\dot{e}_{drag} &= -\frac{K_1\tilde{n}a}{\left(1-e^2\right)^{1/2}}\rho\left(\cos f+e\right)\left(1+2e\cos f+e^2\right)^{1/2} \\
\mathrm{d}i/\mathrm{d}t_{\,drag} &= -\frac{K_2 a}{4(1+e\cos f)^2}\rho\sin i\left(1+\cos\left(2(\omega+f)\right)\right)\left(1-e^2\right)\left(1+2e\cos f+e^2\right)^{1/2} \\
\dot{\Omega}_{drag} &= -\frac{K_2 a}{4(1+e\cos f)^2}\rho\sin\left(2(\omega+f)\right)\left(1-e^2\right)\left(1+2e\cos f+e^2\right)^{1/2} \\
\dot{\omega}_{drag} &= -\frac{K_1\tilde{n}a}{e\left(1-e^2\right)^{1/2}}\rho\sin f\left(1+2e\cos f+e^2\right)^{1/2} - \cos i\cdot\dot{\Omega}_{drag} \\
\dot{M}_{drag} &= -\frac{K_1\tilde{n}a}{e\left(1+e\cos f\right)}\rho\sin f\left(1+e\cos f+e^2\right)\left(1+2e\cos f+e^2\right)^{1/2}
\end{aligned} \tag{24}
$$

The atmospheric density can be expanded into a series dependant upon modified Bessel functions [24, 25], a procedure that results in the following variational equations for the secular terms,

$$
\begin{aligned}
\dot{a}_{drag,sec} &= -K_1\rho_0\tilde{n}a^2\left(1+e^2\left(\frac{3}{4}+\frac{a}{H}+\frac{a^2}{4H^2}\right)+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right) \\
\dot{e}_{drag,sec} &= -K_1\rho_0\tilde{n}a\left(\frac{e}{2}+\frac{ae}{2H}+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right) \\
\mathrm{d}i/\mathrm{d}t_{\,drag,sec} &= -\frac{1}{4}K_2\rho_0 a\sin i\left(1+e^2\left(\frac{3}{4}-\frac{a}{H}+\frac{a^2}{4H^2}\right)+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right) \\
\dot{\Omega}_{drag,sec} &= 0 \\
\dot{\omega}_{drag,sec} &= 0 \\
\dot{M}_{drag,sec} &= \frac{3}{4}K_1\rho_0\tilde{n}^2 a\left(1+e^2\left(\frac{3}{4}+\frac{a}{H}+\frac{a^2}{4H^2}\right)+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right)(t-t_0)
\end{aligned} \tag{25}
$$

while the long-periodic terms are given by:

$$
\begin{aligned}
\dot{a}_{drag,long} &= 0\\
\dot{e}_{drag,long} &= 0\\
\mathrm{d}i/\mathrm{d}t_{\;drag,long} &= -\frac{1}{4}K_2\rho_0 a\sin i\left(\left(\frac{11}{8}-\frac{a}{H}+\frac{a^2}{8H^2}\right)e^2\cos 2\omega+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right)\\
\dot{\Omega}_{drag,long} &= -\frac{1}{4}K_2\rho_0 a\left(\left(\frac{11}{8}-\frac{a}{H}+\frac{a^2}{8H}\right)e^2\sin 2\omega+\mathscr{O}\left(e^3\right)\right)\exp\left(\frac{r_{p0}-a}{H}\right)\\
\dot{\omega}_{drag,long} &= -\cos i\cdot\dot{\Omega}_{drag,long}\\
\dot{M}_{drag,long} &= 0
\end{aligned}
\tag{26}
$$

Finally, the effect of the short-periodic terms can be obtained following the method of Kozai [6].

4 Mean Elements Estimator

The fully-assembled semianalytical model described in Section 3 may be written in the form

$$
\begin{aligned}
\dot{\alpha} = {} & \dot{\alpha}_{sec}\left(\bar{\alpha}\right)+\dot{\alpha}_{long}\left(\bar{\alpha}\right)+\dot{\alpha}_{sec,2}\left(\bar{\alpha}\right)+\dot{\alpha}_{long,2}\left(\bar{\alpha}\right)\\
& +\dot{\alpha}_{long,nm}\left(\bar{\alpha}\right)+\dot{\alpha}_{drag,sec}\left(\bar{\alpha}\right)+\dot{\alpha}_{drag,long}\left(\bar{\alpha}\right)+\mathrm{u}\left(\bar{\alpha}\right)+\mathrm{w}
\end{aligned}
\tag{27}
$$

where u denotes control inputs (either impulsive or continuous), whose effect is modeled using the GVE (4). The incorporation of the control forces using the GVE (4) involves an inherent approximation, since the GVE (4) are written in osculating elements. However, as previous studies have indicated, this approximation is adequate [3]. The vector w appearing in Eq. (27) is an additive process noise, which reflects modeling uncertainties. It is assumed that w is a white noise with power spectral density Q.

The satellite on-board sensors – e.g., a GPS receiver – provide measurements of the inertial position and velocity or unfiltered outputs for the osculating orbital elements by using the inverse of relations (2). Thus, in light of the discussion in Section 3, it is possible to write

$$
\alpha_{osc} = \bar{\alpha}+\alpha_{short}\left(\bar{\alpha}\right)+\alpha_{drag,short}\left(\bar{\alpha}\right)+\alpha_{short,nm}\left(\bar{\alpha}\right)+\upsilon \tag{28}
$$

where υ is the measurement noise, which is a discrete white noise process with the covariance matrix R.

It is readily seen that Eqs. (27) and (28) constitute nonlinear process and measurement equations, respectively, which adhere to the form

$$\dot{\mathrm{x}}\left(t_k\right)=\mathbf{f}\left(\mathrm{x}\left(t_k\right),t_k\right)+\mathbf{w}\left(t_k\right) \tag{29}$$

$$\mathbf{y}\left(t_k\right)=\mathbf{h}\left(\mathrm{x}\left(t_k\right)\right)+\upsilon\left(t_k\right) \tag{30}$$

where $\mathrm{x}\equiv\bar{\alpha}\in\mathbb{R}^n$ is the state vector, $\mathrm{y}\equiv\alpha_{osc}\in\mathbb{R}^n$ is the measurement vector, $\mathbf{f}:\mathbb{R}^n\times\mathbb{R}^+\rightarrow\mathbb{R}^n$, and $\mathbf{h}:\mathbb{R}^n\rightarrow\mathbb{R}^n$. To obtain the estimated mean elements $\hat{\bar{\alpha}}$ from the osculating elements measurements, a nonlinear estimation algorithm should be used. In this work, the square-root form of the spherical simplex unscented Kalman filter (SR-UKF) is utilized.

As mentioned in the Introduction, the UKF is a sigma-point Kalman filter [26–28]. The square-root form of the UKF [16] provides improved numerical stability and keeps the covariance positive semidefinite. The spherical simplex sigma point selection method reduces the number of sigma points and decreases the computational time [17], and is hence more suitable for on-board real-time implementation. For completeness, the main steps necessary for the implementation of mean elements estimation based on the SR-UKF are elaborated herein.

4.1 Spherical Simplex Sigma-Point Selection

For an n-dimensional space, this sigma-point selection strategy provides $n+2$ sigma points, instead of the traditional $2n+1$ points. The point selection algorithm of the spherical simplex unscented transform for an n-dimensional system can be found in [17].

4.2 SR-UKF Algorithm

After the sigma points have been calculated, the SR-UKF can be applied as follows [16]:

(a) Calculate sigma points:

$$\chi_{k-1|i}=\hat{\mathrm{x}}_{k-1}+S_{\mathbf{x}_{k-1}}\mathrm{Z}_i, \qquad i=0,\ldots,n+1 \tag{31}$$

where S_{x} denotes a Cholesky factor, initialized by taking $S_0=\mathrm{chol}\left(E\left[\mathbf{x}_0-\hat{\mathbf{x}}_0\right]\left[\mathbf{x}_0-\hat{\mathbf{x}}_0\right]^{\mathrm{T}}\right)$, $\mathrm{chol}\left(\cdot\right)$ denotes the Cholesky factorization, and $\hat{\mathbf{x}}_0=E\left[\mathbf{x}_0\right]$.

(b) Time update equations:

$$\dot{\chi}_{k-1|i}=\mathbf{f}\left(\mathbf{x}_{k-1},t_{k-1}\right), \qquad i=0,\ldots,n+1 \tag{32}$$

Discretization of the state equations will generally degrade the filter performance; Runge-Kutta integration is used instead for propagating the continuous nonlinear state equations, avoiding discretization. Here $\chi_{k|i}$ is obtained. Now,

$$\hat{\mathrm{x}}_k^- = \sum_{i=0}^{n+1} w_i^m \chi_{k|i} \tag{33a}$$

$$S_{\mathbf{x}_k}^- = \mathrm{qr}\left(\left[\sqrt{w_1^c}\left(\chi_{k|1:n+1} - \hat{\mathbf{x}}_k^-\right), \sqrt{Q}\right]\right) \tag{33b}$$

$$S_{\mathbf{x}_k}^- = \mathrm{cholupdate}\left(S_{\mathbf{x}_k}^-, \chi_{k|0} - \hat{\mathbf{x}}_k^-, w_0^c\right) \tag{33c}$$

$$\chi_{k|i}^* = \hat{\mathbf{x}}_k^- + S_{\mathbf{x}_k}^- Z_i, \qquad i = 0, \ldots, n+1 \tag{33d}$$

$$\gamma_{k|i} = \mathbf{h}\left(\chi_{k|i}^*\right), \qquad i = 0, \ldots, n+1 \tag{33e}$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{n+1} w_i^m \gamma_{k|i} \tag{33f}$$

where $\mathrm{qr}(\cdot)$ is a function that performs the QR decomposition, and $\mathrm{cholupdate}(\cdot)$ is a function that carries out the rank 1 Cholesky factor updating.

(c) Generate the Kalman gain and perform the measurement update:

$$S_{\mathbf{y}_k} = \mathrm{qr}\left(\left[\sqrt{w_1^c}\left(\gamma_{k|1:n+1} - \hat{\mathbf{y}}_k^-\right), \sqrt{R}\right]\right) \tag{34a}$$

$$S_{\mathbf{y}_k} = \mathrm{cholupdate}\left(S_{\mathbf{y}_k}, \gamma_{k|0} - \hat{\mathbf{y}}_k^-, w_0^c\right) \tag{34b}$$

$$P_{\mathbf{x}\mathbf{y}_k} = \sum_{i=0}^{n+1} w_i^c \left[\chi_{k|i} - \hat{\mathbf{x}}_k^-\right]\left[\gamma_{k|i} - \hat{\mathbf{y}}_k^-\right]^{\mathrm{T}} \tag{34c}$$

$$K_k = \left(P_{\mathbf{x}\mathbf{y}_k} / S_{\mathbf{y}_k}^{\mathrm{T}}\right) / S_{\mathbf{y}_k} \tag{34d}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k^-\right) \tag{34e}$$

$$U = K_k S_{\mathbf{y}_k} \tag{34f}$$

$$S_{\mathbf{x}_k} = \mathrm{cholupdate}\left(S_{\mathbf{x}_k}^-, U, -1\right) \tag{34g}$$

where the operator A/B denotes the matrix division of B into A, i.e. a solution for x to the equation $AA^T\mathrm{x} = A^T B$ (see [29] for additional implementation details).

5 Results

The objective of this section is to evaluate the performance of the SS SR-UKF as a mean elements estimator subject to the semianalytical dynamical model. The mean elements estimation performance is evaluated using three test cases: Uncontrolled orbits (including a Monte-Carlo simulation), impulsive orbital corrections and continuous low-thrust orbital transfer. The "true" orbits are generated based on the High Precision Orbit Propagator in STK® , including a 21 × 21 gravity model (EGM96), drag according to the ISA-1976 model [30] and other perturbations, including solar radiation pressure, lunisolar third-body gravitational attraction and tides. For the first example, a lifetime of one year is chosen so that the long-periodic terms, whose period is around five months, can manifest themselves in the orbital dynamics.

5.1 Initial Conditions and Parameter Values

The mission epoch is 1 Mar 2012 10:00:00.000 UTC, and it lasts until 28 Feb 2013 10:00:00.000 UTC. Table 1 gives the initial conditions in terms of osculating orbital elements.

The observation data are generated by contaminating the STK® -generated position and velocity vectors with GPS measurement errors. The one-sigma three-dimensional position and velocity errors are 5 m and 2 cm/s, respectively. The projection of the position and velocity measurement noise covariance onto osculating orbital elements space can be obtained by means of a Monte-Carlo simulation, yielding the results presented in Table 2, where the measurement noise covariance matrix R is given by

$$R = \text{diag}\left[\text{cov}_a, \text{cov}_e, \text{cov}_i, \text{cov}_\Omega, \text{cov}_\omega, \text{cov}_M\right] \tag{35}$$

The time update was performed based on model (27), which included zonal harmonics up to J_4 and tesseral/sectorial terms up to C_{33}/S_{33}. Table 3 lists the numerical values of the geopotential coefficients and other parameters related to the astrodynamical models. The SS SR-UKF parameters are listed in Table 4.

To obtain faster convergence, a judicious initialization of the estimated mean elements is required. To that end, the following initialization is used:

$$\hat{\bar{\alpha}}(t_0) = \alpha_{osc}(t_0) - \alpha_{short}(\alpha_{osc}(t_0)) - \alpha_{drag,short}(\alpha_{osc}(t_0)) \tag{36}$$

Equation (36) generates an approximation of the initial estimated values by replacing the mean elements with the (measured) osculating elements in the expressions for the short-periodic variations.

Table 1 Initial osculating orbital elements values

Parameter	Numerical value
a	7000 km
e	0.01
i	55 deg
Ω	10 deg
ω	10 deg
M	10 deg

Table 2 Measurement noise covariance of the osculating orbital elements

Parameter	Numerical value
cov_a	500.4 m^2
cov_e	8.966×10^{-12}
cov_i	1.886×10^{-12} rad^2
cov_Ω	9.020×10^{-13} rad^2
cov_ω	3.179×10^{-8} rad^2
cov_M	3.086×10^{-8} rad^2

Table 3 Geopotential and drag model coefficients

Parameter	Numerical value
J_2	$1082.62668355 \times 10^{-6}$
J_3	$-2.53265648533 \times 10^{-6}$
J_4	$-1.61962159137 \times 10^{-6}$
C_{22}	$1.57446037456 \times 10^{-6}$
S_{22}	$-9.03803806639 \times 10^{-7}$
C_{31}	$2.19263852917 \times 10^{-6}$
S_{31}	$2.68424890297 \times 10^{-7}$
C_{32}	$3.08989206881 \times 10^{-7}$
S_{32}	$-2.11437612437 \times 10^{-7}$
C_{33}	$1.00548778064 \times 10^{-7}$
S_{33}	$1.97222559006 \times 10^{-7}$
ω_e	$7.2921158553 \times 10^{-5}$rad/s
r_e	6378.137 km
μ	$3.98600436 \times 10^{14}$m^3/s^2
H	68.7 km
ρ_0	2.34×10^{-13} kg/m^3

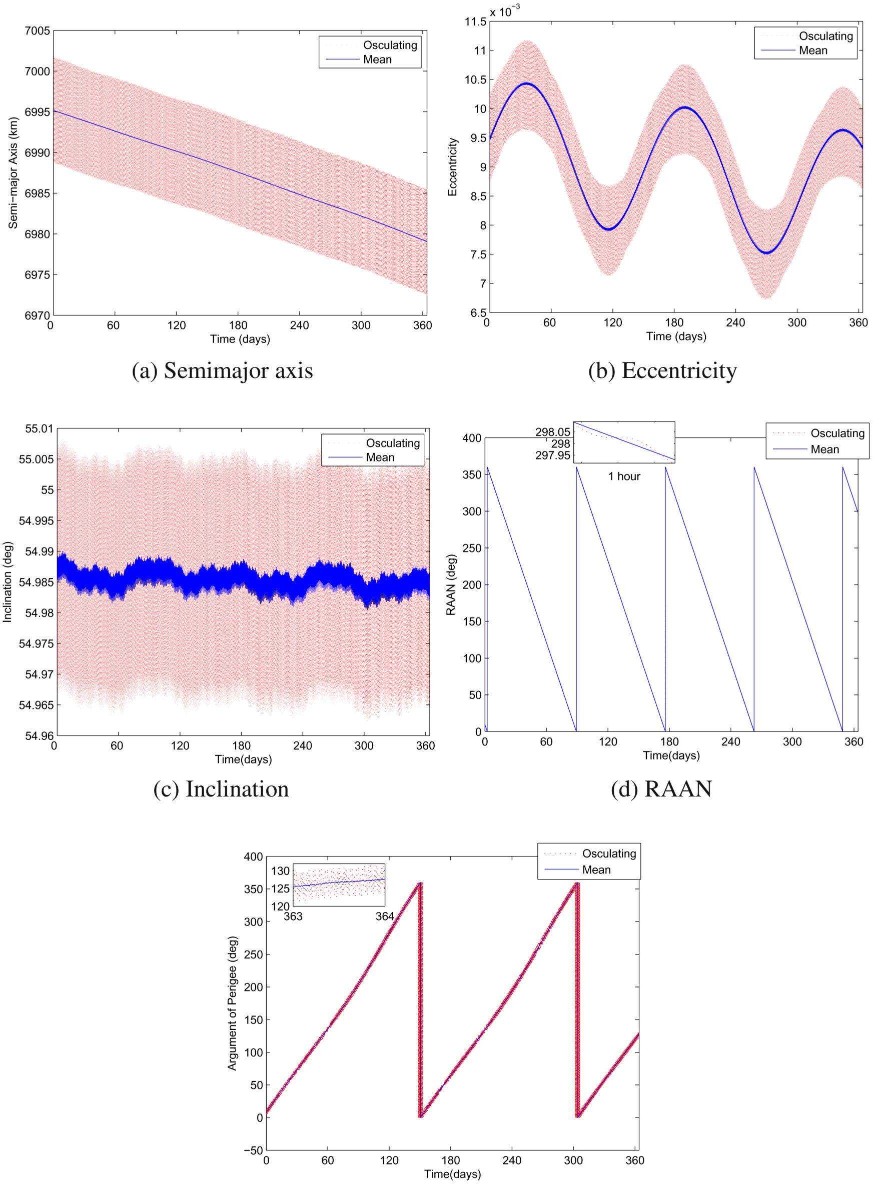
Table 4 Filter parameters

Parameter	Numerical value
W_0	0.25
σ	1
β	0

5.2 Uncontrolled Orbits, Single Run

The simulation study starts by evaluating the ability of the filter to capture the long-periodic and secular mean element evolution for a one-year mission. The results are presented in Fig. 1, which compares the osculating and mean values of the semimajor axis (Fig. 1a), eccentricity (Fig. 1b), inclination (Fig. 1c), RAAN (Fig. 1d), and argument of perigee (Fig. 1e). It is evident that the filter captures the long-periodic evolution of the eccentricity and inclination and the slow secular evolution of the semimajor axis (due to atmospheric drag).

The next step is to evaluate the mean elements estimation errors. However, as opposed to standard filtering problems, in which the states of the process model are used as reference for comparing the estimated states, in the case at hand the semi-analytical model of the mean elements, given by Eq. (27), is truncated on purpose, and does not include effects such as lunisolar attraction, gravitational perturbations beyond order 4 and solar radiation pressure, to allow for computational efficiency. Thus, it makes little sense to use it as a reference for evaluating the estimation errors. Instead, a batch numerical averaging procedure is carried out to evaluate the



(a) Semimajor axis

(b) Eccentricity

(c) Inclination

(d) RAAN

(e) Argument of perigee

Fig. 1 Sample single 1-year simulation run comparing the osculating and estimated mean orbital elements

"true" mean elements per definition (7). The batch averaging relies on the extended Simpson quadrature rule, given by [31]

$$\begin{aligned}\bar{\alpha} &= \frac{1}{T}\int_0^T \alpha(t)\,\mathrm{d}t = \frac{1}{N}\int_{t_0}^{t_{N-1}} \alpha(t)\,\mathrm{d}t \\ &= \frac{1}{N}\left(\frac{1}{3}\alpha(t_0) + \frac{4}{3}\alpha(t_1) + \frac{2}{3}\alpha(t_2) + \frac{4}{3}\alpha(t_3) + \right. \\ &\left. \cdots + \frac{2}{3}\alpha(t_{N-3}) + \frac{4}{3}\alpha(t_{N-2}) + \frac{1}{3}\alpha(t_{N-1})\right) + \mathscr{O}\left(\frac{1}{N^4}\right)\end{aligned} \tag{37}$$

where the alternation of 2/3 and 4/3 continues throughout the interior of Eq. (37). The vector of estimation errors is then defined as

$$\mathrm{e}_\alpha \triangleq \bar{\alpha} - \hat{\bar{\alpha}} \tag{38}$$

The estimation errors (38) are compared to a direct application of the Brouwer artificial satellite theory as formulated in Ref. [32]. The purpose of the comparison between the Brouwer and SS SR-UKF-based estimation of the mean elements is to inquire whether a direct application of the Brouwer theory may be used as a substitute for recursive filtering; as will be seen shortly, the answer to this question is strictly negative.

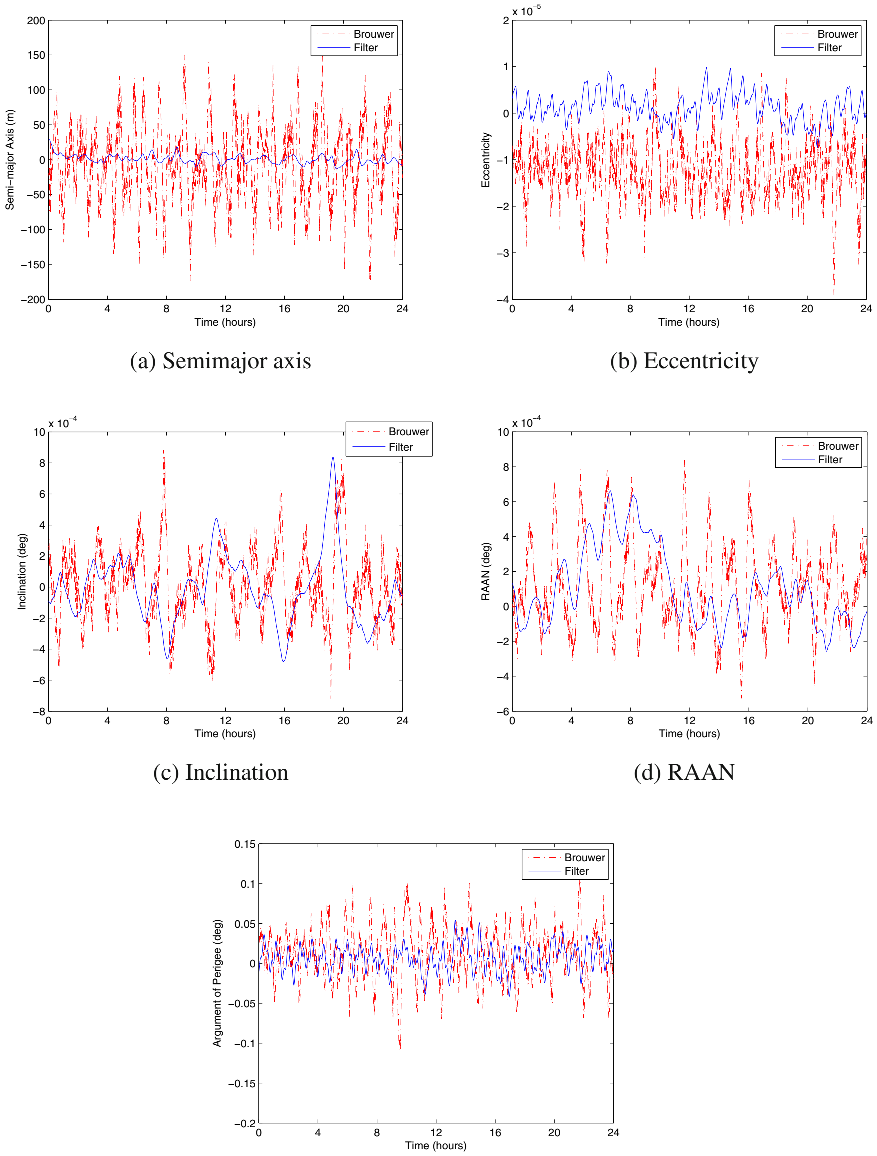
A comparison of estimation errors between the SS SR-UKF and the Brouwer theory is depicted in Fig. 2. In this figure, the estimation errors of the semimajor axis (Fig. 2a), eccentricity (Fig. 2b), inclination (Fig. 2c), RAAN (Fig. 2d) and argument of perigee (Fig. 2e) are shown for a 24-hour period to better illustrate the quantitative aspects of the differences between the Brouwer theory and the recursive filter. It is evident that the Brouwer-based errors are noisier than the filter-based estimation errors. In addition, the eccentricity estimation through Brouwer's theory is slightly biased, whereas in the filter-based estimation it is unbiased.

5.3 Uncontrolled Orbits, Monte-Carlo Runs

The observations in Section 5-5.2 can be substantiated using a long-term Monte-Carlo simulation. The numerical values of the standard deviations are summarized in Table 5. It is evident that the filter yields at least an order of magnitude better

Table 5 Standard deviation values of the mean elements estimation errors for 100 Monte-Carlo Runs

σ	Filter	Brouwer
e_a	0.6118 m	22.42 m
e_e	3.972×10^{-7}	2.994×10^{-6}
e_i	9.380×10^{-6} deg	7.877×10^{-5} deg
e_Ω	6.665×10^{-6} deg	5.486×10^{-5} deg
e_ω	0.0013 deg	0.0103 deg



(a) Semimajor axis

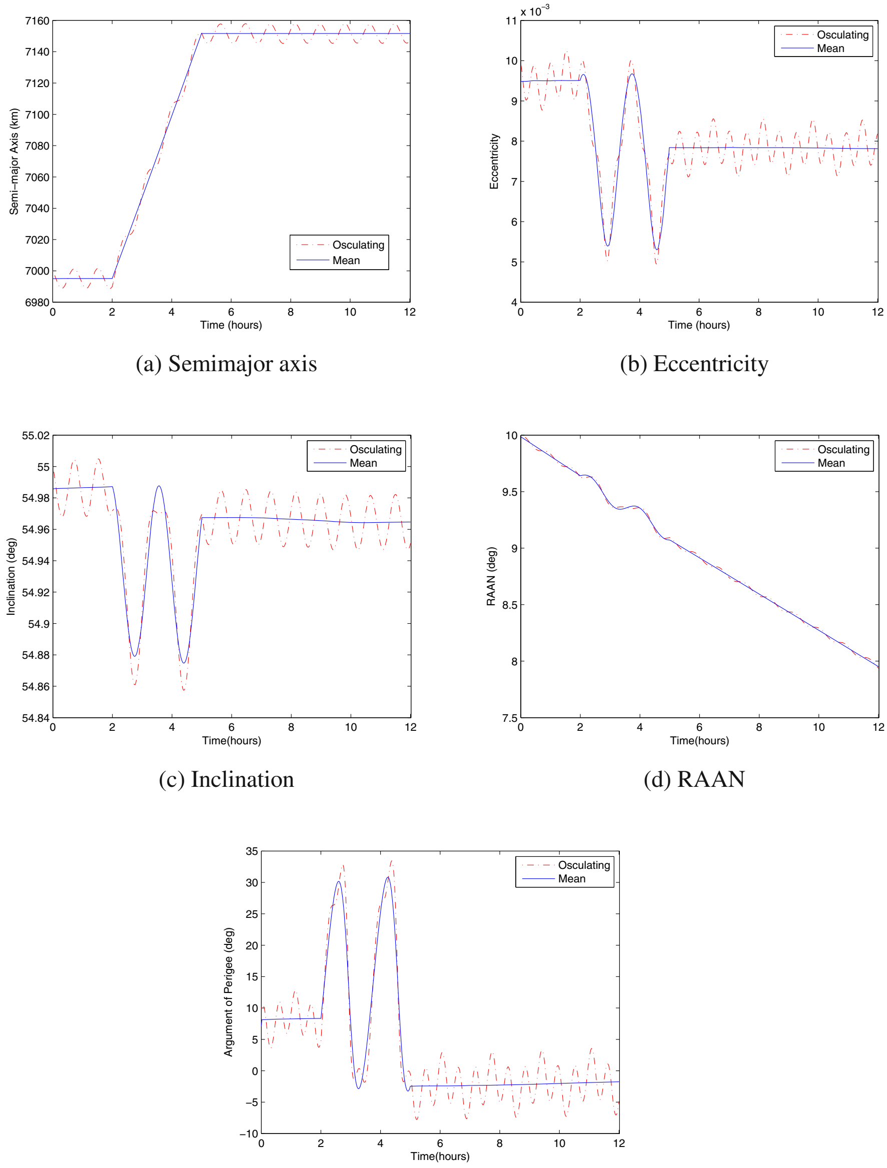
(b) Eccentricity

(c) Inclination

(d) RAAN

(e) Argument of perigee

Fig. 2 Sample simulation run comparing the mean element estimation errors as obtained from the Brouwer theory and the SS SR-UKF filter



(a) Semimajor axis

(b) Eccentricity

(c) Inclination

(d) RAAN

(e) Argument of perigee

Fig. 3 Estimated mean elements under continuous constant-magnitude low thrust. The SS SR-UKF provides unbiased estimation of the mean elements for a 3-hour thrusting arc.

accuracy than the Brouwer theory. The difference is particularly dramatic in the estimation of the semimajor axis, wherein the filter provides a 1-σ accuracy of about 60 cm, compared to a Brouwer 1-σ value of about 22 m.

5.4 Continuous Thrust

The final test case involves continuous thrust. The magnitude of the thrust is $0.08\,N$, as used in the SAMSON mission [33] and the direction is determined by the unit vector $\left[\sqrt{3}/3\,,\sqrt{3}/3\,,\sqrt{3}/3\,\right]$ in the **NTW** frame, indicating that all three maneuver channels are active. A thrusting arc of 3 hours is used. The results for a 12-hour integration are shown in Fig. 3. Although the acceleration caused by the continuous thrusting induces both secular and periodic variations in the mean elements, the filter is capable of providing unbiased estimation of the mean elements, including the semimajor axis (Fig. 3a), eccentricity (Fig. 3b), inclination (Fig. 3c), RAAN (Fig. 3d) and argument of perigee (Fig. 3e).

6 Conclusions

Using the spherical simplex square-root unscented Kalman filter for estimating the mean orbital elements has clear advantages over the Brouwer artificial satellite theory: It can adequately respond to thrust, and it is much less sensitive to measurement noise. In the examined scenarios, the filter provided a sub-meter 1-σ estimation accuracy of the mean semimajor axis, which is almost two orders of magnitude better than the Brouwer-based estimation. The filter was designed from an astrodynamical standpoint – the equations were formulated in a True of Date reference frame, thus taking into account the precession and nutation of the Earth; tesseral and sectorial harmonics were not neglected, thus yielding improved estimation of the inclination and eccentricity variations; and the short-periodic effects were used to transform from mean to osculating elements.

Acknowledgements. This work was supported by the European Research Council Starting Independent Researcher Grant # 278231: Flight Algorithms for Disaggregated Space Architectures (FADER).

References

1. Chobotov, V.: Orbital Mechanics, vol. 1, ch. 14, pp. 335–409. AIAA (2002)
2. Gao, Y.: Low-thrust Nonlinear Guidance by Tracking Mean Orbital Elements. Journal of Guidance, Control, and Dynamics 31(4), 1103–1110 (2008)
3. Schaub, H., Alfriend, K.T.: Impulse Feedback Control to Establish Specific Mean Orbit Elements of Spacecraft Formations. Journal of Guidance, Control and Dynamics 24(4), 739–745 (2001)

4. Salama, O.: Autonomous Orbit Maintenance Law for LEO Sun Synchronous, Earth Repeating Satellites with Electric Propulsion System. In: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, USA, AIAA Paper 2008-7197 (August 2008)
5. Brouwer, D.: Solution of the Problem of Artificial Satellite Theory without Drag. The Astronomical Journal 64, 378–396 (1959)
6. Kozai, Y.: The Motion of a Close Earth Satellite. The Astronomical Journal 64, 367–377 (1959)
7. Brouwer, D., Hori, G.I.: Theoretical Evaluation of Atmospheric Drag Effects in the Motion of an Artificial Satellite. The Astronomical Journal 66, 193–225 (1961)
8. Liu, J.J.F., Alford, R.L.: Semianalytic Theory for a Close-Earth Artificial Satellite. Journal of Guidance, Control, and Dynamics 3(4), 304–311 (1980)
9. Liu, J.J.F.: Advances in Orbit Theory for an Artificial Satellite with Drag. Journal of the Astronautical Sciences 31(2), 165–188 (1983)
10. Liu, J.J.F., Alford, R.L.: A Semi-analytic Theory for the Motion of a Close-earth Artificial Satellite with Drag. In: AIAA, Aerospace Sciences Meeting, New Orleans, LA, USA, AIAA Paper 79-0123 (January 1979)
11. Lane, M.H.: The Development of an Artificial Satellite Theory Using a Power-law Atmospheric Density Representation. In: AIAA Second Aerospace Sciences Meeting, New York, NY, USA, AIAA Paper 65-35 (January 1965)
12. Lane, M.H., Fitzpatrick, P.M., Murphy, J.J.: On the Representation of Air Density in Satellite Deceleration Equations by Power Functions with Integral Exponents. Tech. rep., DTIC Document, Report No. APGC-TDR-62-15 (1962)
13. Hoots, F.R.: Theory of the Motion of an Artificial Earth Satellite. Celestial Mechanics and Dynamical Astronomy 23(4), 307–363 (1981)
14. Lin, L., De-zi, Z.: Combined Perturbation on Near-earth Satellite Orbits. Chinese Astronomy and Astrophysics 5(4), 422–433 (1981)
15. Bezdek, A., Vokrouhlickı, D.: Semianalytic Theory of Motion for Close-earth Spherical Satellites Including Drag and Gravitational Perturbations. Planetary and Space Science 52(14), 1233–1249 (2004)
16. Van Der Merwe, R., Wan, E.: The Square-root Unscented Kalman Filter for State and Parameter-estimation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3461–3464. IEEE (2001)
17. Julier, S.: The Spherical Simplex Unscented Transformation. In: Proceedings of the American Control Conference, vol. 3, pp. 2430–2434. IEEE (2003)
18. Battin, R.: An Introduction to the Mathematics and Methods of Astrodynamics, ch. 10, pp. 471–514. AIAA (1999)
19. Vallado, D., McClain, W.: Fundamentals of Astrodynamics and Applications, vol. 21, ch. 9, pp. 605–693. Springer (2007)
20. Lambeck, K.: Geophysical Geodesy, ch. 5, pp. 171–215. Clarendon Press, Oxford (1988)
21. Liu, L., Hu, S.J., Wang, X.: An Introduction to Astrodynamics, ch. 4, pp. 90–114. Nanjing University Press (2006)
22. Liu, L.: Orbit Theory of Spacecraft, ch. 4, pp. 114–148. National Defence Industry Press (2000)
23. Liu, L., Zhao, D.: Combined Perturbation on Near-earth Satellite Orbits. Chinese Astronomy and Astrophysics 5, 422–433 (1981)
24. Brouwer, D., Clemence, G.M.: Methods of Celestial Mechanics, vol. 1, pp. 45–122. Academic Press, NY (1961)
25. Breiter, S., Metris, G.: Keplerian Expansions in Terms of Henrard's Practical Variables. Celestial Mechanics and Dynamical Astronomy 58(3), 237–244 (1994)

26. Julier, S., Uhlmann, J., Durrant-Whyte, H.: A New Approach for Filtering Nonlinear Systems. In: Proceedings of the American Control Conference, vol. 3, pp. 1628–1632. IEEE (1995)
27. Julier, S., Uhlmann, J., Durrant-Whyte, H.F.: A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. IEEE Transactions on Automatic Control 45(3), 477–482 (2000)
28. Haykin, S., et al.: Kalman Filtering and Neural Networks, ch. 7, pp. 221–269. Wiley Online Library (2001)
29. Alfriend, K.T., Gurfil, P., Vadali, S.R., How, J.P., Breger, L.: Spacecraft Formation Flying: Dynamics, Control, and Navigation, ch. 3, pp. 39–52. Butterworth-Heinemann (2010)
30. Noaa, N., Usaf, U.: Standard Atmosphere. US Government Printing Office, Washington, DC (1976)
31. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes: The Art of Scientific Computing, 3rd edn., ch. 4, p. 160. Cambridge University Press (2007)
32. Schaub, H., Junkins, J.L.: Analytical Mechanics of Space Systems, vol. 1, pp. 573–576. AIAA (2003)
33. Gurfil, P., Herscovitz, J., Pariente, M.: The SAMSON Project - Cluster Flight and Geolocation with Three Autonomous Nano-satellites. In: 26th AIAA/USU Conference on Small Satellites, Salt Lake City, UT, USA, Paper SSC12-VII-2 (August 2012)

Flight Nutation Validation of the COS-B and EQUATOR-S Spacecraft

Hans Kuiper and Edward Bongers

Abstract. The validation of spacecraft flight nutation damping performance can only be obtained when flight data become available. Dedicated space nutation tests, e.g. in a decommissioning phase, are required to enable a systematic evaluation of model, ground test and space performance results. Space nutation flight data, however, are sparsely available. This article deals with the verification and validation of the COS-B and EQUATOR-S nutation flight data on basis of their ground test data and three model types. It will be shown that the Navier-Stokes model solution in the co-rotating body reference frame, as used in the development of the Ulysses, FY-2 and Cluster nutation dampers, is the backbone of liquid damper design of the type "tube-with-endpots".

1 Introduction

The flight nutation performance data of a series of spinning spacecraft (S/C) given in Table 1 was studied. The S/C were selected for reasons of availability of their Nutation Damper (ND) design heritage, flight and ground test as well as qualification data. The given S/C refer to different mission classes: Meteorological in GEO orbit (3, 4) and purely scientific (1, 2, 5, 6). From the latter class #2, 5 and 6 are in highly elliptical Earth orbits whilst #1 is in a solar elliptical orbit. The Attitude and Orbit Control System (AOCS) of S/C #5 and #6 incorporates a pair of meridian NDs, shown in Figs-1, 2 and 3, which are mounted at radius R_m from the spin-axis perpendicular to the equatorial Centre-Of-Mass (COM) plane at $Z = 0$.

Hans Kuiper
Chair Space Systems Engineering, Faculty of Aerospace Engineering
Delft University of Technology, Kluyverweg 1 – 2629HS Delft, The Netherlands
e-mail: J.M.Kuiper@tudelft.nl

Edward Bongers
Dutch Space B.V., Mendelweg 30 – 2333DB Leiden, The Netherlands
e-mail: E.Bongers@dutchspace.nl

All other S/C are equipped with a pair of equatorial NDs mounted in the equatorial plane at height $Z = Z_0$ above the COM plane and $R = R_e$. Fig 2 shows the typical liquid damper design of the type “tube-with-endpots” to be considered in this article.

The only traceable and useful space test validation data are from the COS-B S/C [5] and the EQUATOR-S S/C [12] mission. The applied NDs are both of the meridian type with comparable design driving requirements as becomes clear from Table 2 in section 4. In the following sections a cross verification of different models is made using experimental results on ground as well as space flight data.

Table 1 Selection of nutation damping V&V missions

V&V #	S/C	Prime contractor Principal investigator	ND[1]	Spin-rate [rpm]	Lifetime
1	Ulysses	ESA/NASA	E	5	1990-2008
2	Cluster	ESA/NASA	E	5	2001-
3	FY-2	Chinese GWIC	E	100	1994-
4	MSG	ESA/Eumetsat	E	100	2002-
5	EQUATOR-S	Max Planck	M	48	1997
6	COS-B	ESA	M	10	1975-1982

[1] E=equatorial, M= Meridian

In the exploration of the applicable nutation damping theory different models were distinguished and compared. This article is dedicated to the verification and validation (V&V) of these models given by [2, 3, 4, 11, 12, 14, 15, 17, 18, 23].

The first step is the verification of the model predictions by scaled ground tests using a Performance Test Model (PTM). This typically happens during the phase B of a Nutation Damper (ND) project and is essential to convince the customer (prime contractor) of the health of the design. After the Preliminary Design Review (PDR) milestone is successful the go ahead is given for the extended test and production phase C/D of the qualification (QM) and the flight models (FM).

The ultimate ground test system, to verify and validate flight ND damping performance, is either a horizontal pendulum arrangement like the air-bearing test setup at Dutch Space [5] or a vertical pendulum [11].

In the following section, first the applicability of the selected missions #3 up to 6 will be briefly dealt with whilst #1 and 2 are skipped due to the limited amount of useful data. The ground test equipment as primary instrument for verification of the ND model predictions will be discussed in section 3.

2 Verification and Validation Data

2.1 MSG and FY-2

The meteorological European Meteosat Second Generation (MSG) and Chinese FY-2 S/C in Table 1 are both 100 rpm spin-stabilized geostationary S/C.

The nutation of the sophisticated MSG S/C is dealt with briefly despite the limited amount of traceable V&V flight data. The MSG NDs are of the equatorial tube-with-endpots type and were made by URENCO in the Netherlands. The author was involved independently in the pre-phase A study questions from Marconi Space Systems as well as those from subcontractor Aerospatiale Cannes [9] to supply a basic ND design and Rough Order of Magnitude (ROM) cost estimate. The MSG flight validation data are given by [10, 16, 21]. In these sources the following types of MSG nutation damping are considered:

- **Nutation damping and tank sloshing -** The nutational damping may reveal sharp resonance effects at particular propellant tank fillings. This was studied for MSG by [10]. The authors state that liquid inertial modes and S/C nutation are strongly resonance related and may explain the strong variations in in-orbit tests on INTELSAT-V and in the airbearing tests [1].
- **Nutation due to scan mirror movements -** The MSG S/C incorporates the Spinning Enhanced Visible and Infra-Red Imager (SEVIRI) which is capable to provide measurements of stars, Earth horizon and landmarks, apart from meteorological images.

The MSG nutation due to the SEVIRI scan mirror movements is dealt with by [16]. The author was involved as a system engineer in the design team of the SEVIRI passive radiant cooler [6] as well. The SEVIRI instrument contains a 20 kg mirror that rotates around an axis parallel to the S/C X-axis with the Z-axis being the spin-axis. The equatorial NDs are mounted on the Y-axis. The SEVIRI mirror rotation can be described as an initial stage in which the mirror is repositioned (retraced) in the starting point to obtain the meteorologocal data, a second Black-Body calibration stage at fixed position and a final scanning stage to acquire the image. The mirror movement induces a variation in the S/C inertia matrix and as a consequence modified rotations thus S/C nutation during the scanning and retracing phases.

A S/C attitude dynamics model including the NDs is given by [16] and incorporates viscous damping. The worst case performance yields a time constant $\tau = 2$ min which is compliant with the requirement $\tau < 4$ min [21].

The maximum impact in between two nominal scan cycles equals $\Delta\omega_y = 0.5$ [mrad/s] angular velocity about the principal inertia Y-axis. The results comply with the ND performance design data and comply very well with results derived from telemetry data. This is, beside a confirmation of the correctness of the applied models, a proof of the predicted nominal ND performance.

The strongest ND performance model proofs, however, will be extracted from the EQUATOR-S and especially COS-B flight data in the following sections. First these missions are briefly regarded, secondly the terrestrial test equipment for the scaled ND performance experiments is described and finally the validation by flight data will be considered.

2.2 *EQUATOR-S*

The EQUATOR-S S/C mission was designed and developed at the Max-Planck Institute in Garching with some supplementary items from agencies and industry. The S/C was launched the 2nd of December 1997 into an elliptical Earth orbit with perigee 479 km, apogee 67,275 km, inclination 3.9^{0} and orbit period T = 22.3 hr. The science mission included magneto-spherical plasma and field measurements in the frame of the International Solar Terrestrial Physics Program.

The meridian EQUATOR-S NDs are shown in the right picture of Fig 1. Its build standard is derived from the AMPTE-IRM S/C meridian ND [19] and shown in Fig 2. The endpots are adhesive fastened whilst the similar Fokker type tube-with-endpots NDs are electron beam welded. Space validation data for the EQUATOR-S operational case are given by [11,12].

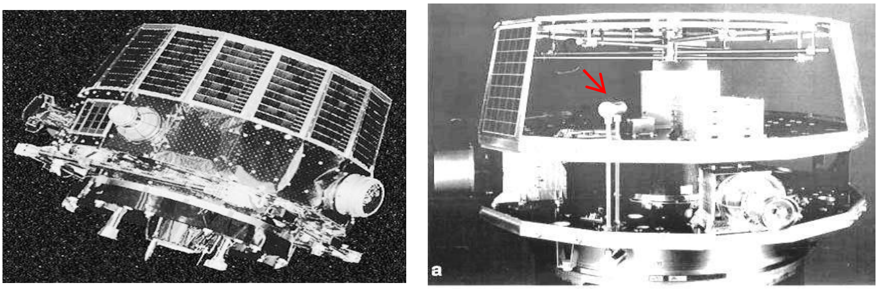


Fig. 1 EQUATOR-S S/C (left picture) and its interior (right picture) showing one of the two meridian NDs vertically mounted [12]. One endpot is red indicated.

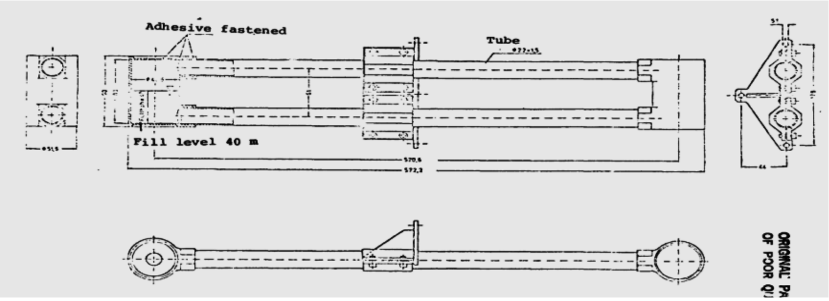


Fig. 2 AMPTE ND build standard [19]. This damper design is adhesive fastened at the endpots whilst the COS-B damper and its successors were electron-beam welded. The drawing shows the liquid and vapor tube with the same radius **a** [mm] whilst the two cylindrical endpots have the radius **b** [mm]. The equilibrium liquid level in the endpots h = H/2 with H the endpot height which is indicated red. Together with the physical tube length L this yields the geometry design set <a, b, L, H>.

2.3 COS-B

The COS-B scientific mission [22] was to study in detail the sources of extra-terrestrial gamma radiation at energies above about 30 MeV. The successful mission had no optical telescopes or complicated scientific instrumentation. The only function was to point in the direction of a star or other object and measuring its gamma-ray emissions. COS-B was an ESA mission, built and equipped by European scientists and launched by NASA. It was ESA's first satellite dedicated to a single experiment enabling an extensive survey of the galaxy in the energy range 50 MeV to 5 GeV. The spacecraft overall mass equals 278 kg and the attitude measurement accuracy 1^0. Its cylindrical sizes are a height 1.2 m and a diameter 1.4 m. The orbit with period T = 37 hour was eccentric (launch 1975-08-09, perigee 350 km, apogee 100,000 km, inclination 90^0) ensuring that the satellite was outside the Earth's radiation belts for most of the time [7]. The end of the mission coincided with the end of its propellant supply, which had been conserved by careful choice of maneuvers. The originally foreseen duration of the mission was two years, but COS-B was finally switched off on 25th April 1982, having functioned successfully for more than 6.5 years. In spring 1982 a dedicated series of space nutation experiments were executed at different frequencies and initial nutation angles induced by thruster firing. The position of the NDs in the spacecraft is shown in Fig 3. The rich amount of COS-B nutation modeling, ground verification and space operational validation data [5] was partly re-analyzed.

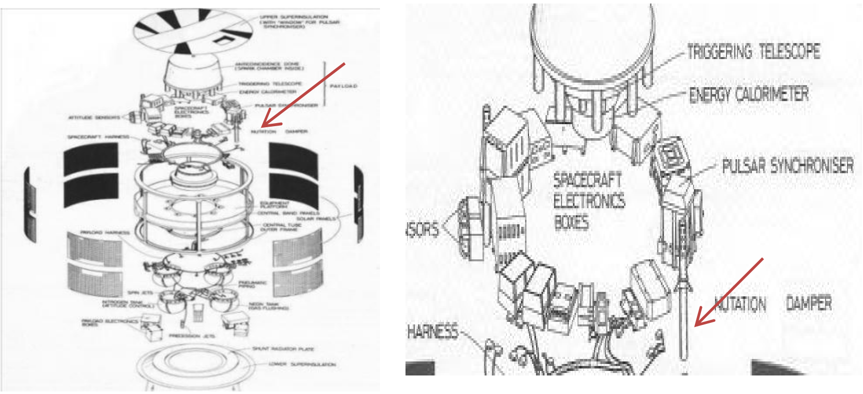


Fig. 3 COS-B S/C exploded view (left) and enlarged upper stage with a meridian ND [7]

3 Airbearing Pendulum Tests

3.1 Vertical Pendulum

The vertical plane pendulum used by [12] to verify the EQUATOR-S meridian ND performance is shown in Fig 4. In the left picture the two black contra weights are visible along the vertical rod whilst the (faintly visible) plexi-glass scale model is at the lower side. This scale model is enlarged in the right picture. In this pendulum configuration with horizontal axis, springs are not required. The angular dependent gravitational acceleration is taken into account in the Euler-Lagrange equations of the damper liquid motion.

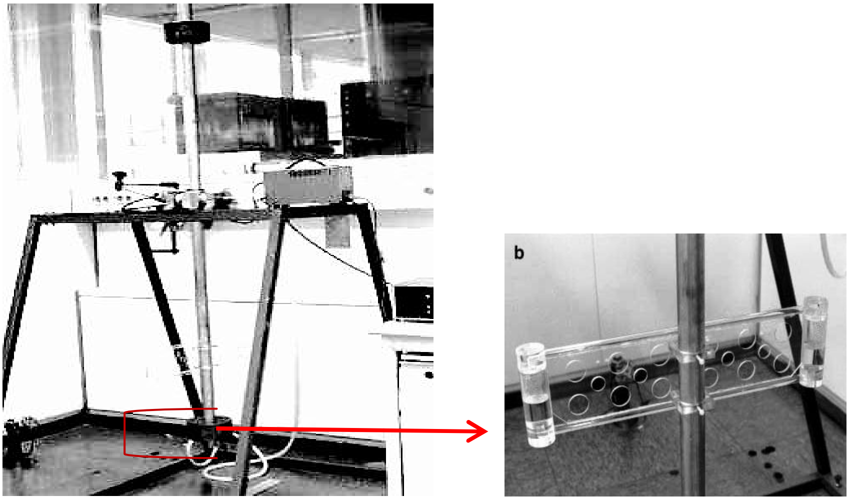


Fig. 4 The vertical pendulum for ND performance verification used by [12]. The meridian plexi-glass Performance Test Model (PTM) is enlarged in the right picture.

3.2 Horizontal Pendulum

The setup of the Fokker ND performance measurement setup with tests oscillations in the horizontal iso-gravity plane is shown in Fig 5. The zoomed-in picture at the right side shows the central air-bearing hub, the oscillation springs and the ND attachment rail at the end of the mounting arm. One of the two wire-suspended arms is recognized as well as the two oscillation springs which provide the harmonic driving torque. For stiffness and stability reasons a steal test-arm with square intersection and length $R_a = 2$ m was chosen. This limits torsion and linear arm oscillations down to < 1% disturbing accelerations of the ND liquid damping performance to be tested. The digital angle read-out - an optical encoder - and computer system are integrated to monitor the oscillation angle with 2.75” resolution.

The spacecraft nutation is simulated by the oscillatory movement about the vertical rotation axis. The desired pendulum time is provided by two adjustable springs and/or additional masses. The horizontal pendulum with vertical axis uses the gravitational 1-g force to scale ("replace") the flight centrifugal acceleration. Visible in Fig 5 are the suspension wires of the ND arms and the oscillation springs. The NDs themselves are not attached.

The damping performance at very small angles is analyzed after automatic recording and processing of the normalized damping versus the angular amplitude. The latter is given by $R_a \cdot \Phi(t)$ over the specified amplitude range with R_a the mounting arm length and $\Phi(t)$ the momentary air-bearing oscillation angle.

The damping performance verification is initially obtained by Performance Test Model (PTM) scale tests. In a later project phase this is done in addition by the Qualification Model (QM) and Flight Model (FM) air-bearing performance tests yielding additional damping performance verification.

The air-bearing arm length R_a and its Mass-Of-Inertia (MOI) I_0 are independent of scaling rules. On basis of measurement accuracy trade-offs a compliant arm length is chosen. The radius for the FY-2 ND was e.g. taken R_a= 2000 ± 5 mm with a total MOI of 199.1 kg·m^2 using two dummy models (DMs) without damping liquid or 190–200 kg·m^2 with two PTMs. The difference is attributed to the 'free' sloshing motion of the damping liquid in the endpots of the ND. For other ND projects specific choices and refined trade-offs are made on the arm lengths and the required stability as imposed by the ND test plan.

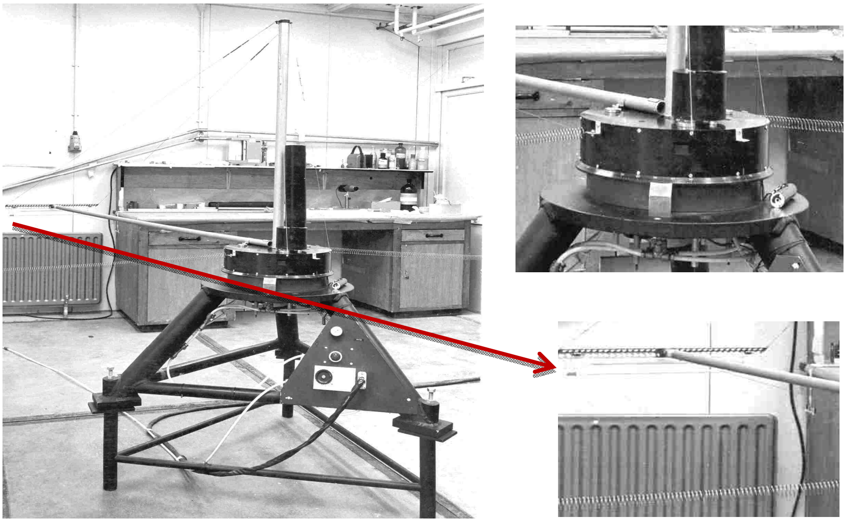


Fig. 5 Air-bearing setup for the ND damping performance tests at Dutch Space in 1990. The red arrow indicates the support rail for the ND which is not attached.

The exponential damping time constant τ_{test} is found using single logarithmic plots of the amplitude $R_a\Phi(t)$ recorded over a time range $\Delta t = t_e - t_0$ using 4 to 5 logarithmic units, i.e., ln $[\Phi(t_0)/\Phi(t_e)] = 4$ to 5 with

$$R_a\Phi(t) = R_a\Phi_0 e^{-t/\tau_{test}}. \tag{1}$$

The time test range is related to the required nutation angle range prescribed by the specific ND project requirements. The ranges are scaled according space-to-ground fluid mechanics scaling rules.

By shifting the extra weights on the test arms the oscillation frequency can be varied. The net ND damping time constant τ_{ND} is finally obtained by

$$\frac{1}{\tau_{ND}} = \frac{1}{\tau_1} - \frac{1}{\tau_2} \tag{2}$$

with τ_1 the test damping time constant using a PTM, QM or FM and τ_2 the test damping time constant using a dummy model (DM). The time constant τ_2 refers to the same test but with the PTM, QM and FMs replaced by dummy models. These DMs have the same mass and aerodynamic geometry but do not contain damping liquid. This enables the separation of the liquid damping performance from the friction of the air-bearing and air resistance. The reduced time constant τ_{ND} approximates the time constant of the nutation liquid damping only. The influence of the test room air-environmental damping and air-bearing friction are therefore practically eliminated by applying Eqn. (2).

The reverse exponential damping time constant (the damping coefficient) is related to the normalized average dissipation per (nutation or test) cycle $\bar{P}$ and the squared test acceleration a_o^2 in the liquid tube by

$$\frac{1}{\tau_{test}} = \frac{R_a^2\Omega^2}{I_{tot}}\left(\frac{\bar{P}}{a_0^2}\right)_{test} \tag{3}$$

with $I_{tot} = I_0 + 2I_{ND}$ the total inertia about the vertical air-bearing axis, I_0 the test arm MOI including resonance tuning weights and I_{ND} the additional inertia due to the test ND thus the PTM, DM, QM or FM.

The damping coefficient given by Eqn. (3) becomes constant only in the laminar oscillation range at very small angles. The constant τ_{test} depends on the air-bearing radius, its MOI, the airbearing angular test frequency and the damping rate of the test model (PTM, QM, FM and DM). Finally by scaling the 1-g damping rate towards its space value, the performance of the FM as function of the S/C inertia ratio λ is found. As basic rule five λ values are taken as minimum number to define a proper performance check of the FM design. A number of precautions are taken to obtain optimal test conditions and accurate results:

- The total MOI of the air bearing including the NDs (or DMs) is measured
- The oscillation arms generate airflows. This effect is reduced by doing the same measurements with DMs and applying Eqn. (2). Additional disturbance effects, possibly caused by people walking around or accidental ventilation

flows, are avoided. It was proposed by the author to do tests with aerodynamic (ellipsoidal encapsulation) profiles attached to the end of the arms around the attached NDs to limit and control the additional airflows even further

- NDs are leveled within arcsecs using a theodolite
- The air-bearing feet are adjusted up to the level where the end of the arms are within arcsec variation over the full angular range at the operational air-bearing operational pressure
- The moving damper liquid exerts a torsional moment on the mounting arms. To limit vibrations the torsional material stiffness *G* [Nm/rad] of the test arms is chosen to limit torsional accelerations down to 1% of the maximum tangential value. The *G* value is verified by applying a zero and a well-defined torsional moment on one end of the mounting arm whilst theodolite measurements are made with a mirror at the end of the arm
- The same kind of engineering arguments hold for the trade-off on the effective test arm length which is driven by a < 1% relative magnitude of the centrifugal to nutation acceleration
- The centrifugal flight acceleration field is curved whilst the terrestrial 1-g field is straight. This implies a curved FM and a straight PTM. In general, however, the FM NDs are chosen straight as well as the PTMs. The effect on the damping performance is negligible. To prove this, consider a straight ND in a curved acceleration gravity field. It is clear that the acceleration decrease in the left part of the liquid tube is symmetrically compensated by an equal increase in the right part. The same argument holds for a curved ND in a straight 1-g terrestrial gravity field.

4 EQUATOR-S and COS-B Data Analysis

4.1 Input Parameters

The two S/C and their ND design data are given in Table 2. In the last rows the relative ND liquid mass and the factor k_0 is given. The factor k_0 accounts in the refined Hagen-Poiseuille (RHP) model [12] for the effective relative amount of damping liquid mass.

The amount of COS-B liquid mass *m* compared to the S/C mass *M* - *with requirement* $m/M \ll 1$ *as prerequisite of a quasi-rigid S/C configuration* - is about one third of the EQUATOR-S value whilst the k_0 factor is comparable. The required 5” extreme nutation angle for COS-B and the less than 0.5” per orbit drift requirements follow from the X-ray science experiment requirements [22].

4.2 Applicable Models

Four applicable ND models are dealt with in the given sources in the introduction. In the frame of the nutation Verification and Validation (V&V) analysis they are briefly summarized here:

1. The Hagen-Poiseuille (HP) model considers a fully developed single directional flow which is in general not valid due to the harmonic nutation sweep-up forces in the liquid tube. Only at low S/C spin-rates with consequently low nutation frequencies the model can be used. The HP model is a special class of the generic Navier-Stokes solution. The model proved to be well applicable to describe and quantify the natural resonance behavior at sweep-up moments of nutation in performance verification tests.
2. The Refined HP (RHP) model [12] is an adapted version of the HP model based on the theory derived by [18]. An additional factor k_0 accounts for the effective relative amount of damping liquid mass in a system with static and dynamic (i.e. liquid) inertia contributors.
3. The Navier Stokes (NS) model derived by solving the Navier Stokes Equation (NSE) in the co-rotating S/C body reference frame with the fluid velocity relative to the liquid tube.
4. NS model [8] applied at ESTEC/ESA based on the solution of the NSE but on basis of the absolute fluid velocity. It was used beside the previous NS model in the Ulysses nutation anomaly study. The power dissipation and resulting damping time constants agree seamless though different inertial systems are used.

Table 2 COS-B [5] and EQUATOR-S [12] meridian ND design data

Spacecraft [source]	**COS-B** [5, 7]		**EQUATOR-S** [11, 12]	
Total mass M [kg]	278		216.9 (dry mass)	
Spin axis inertia I_{ZZ} [kg.m^2]	54.53		45.35^a to 54.67^b	
Spinrate ω_Z (nom) [rpm]	10 ± 0.1		48 to 50	
BOL inertia ratio λ_0 [-]	1.232		1.33^a to 1.406^b	
Nutation amplitude [degree]				
• Initial	3.8		3.3	
• Final	< 5”		< 1.0^c	
Temperature [deg C]	-20 to +30		+12.7 (nomimal)	
Nutation Damper				
Hardware model	**FM**	**PTM**	**FM**	**PTM**^{d}
Materials liquid / housing	PP1/ Al	PP1/ Al	PP3/Al	H_2O/plexi-glass
ND time constant [s]				
• Theory (NS, RHP)	see multiple cases in		38.7 (RHP)	72.3 (RHP)
• FM space/PTM	text		38 ± 3(RHP)	72 ± 6
ground test	668		565	
S/C mounting radius R_m [mm]				
Design FM dimensions	4.0, 20.96, 400, 40		5.0, 23.04, 500, 80	
<a, b, L, H> [mm]	PP1/PP1		PP3/H2O	
FM/PTM damping liquid	9.9	18-21	12.7	18-21
ND liquid temperature [^{0}C]	0.136		0.291	
ND liquid mass m [kg]	0.049		0.134	
ND liquid mass ratio m/M [%]				
RHP k_0 correction factor [-]	0.473		0.374	
see Eqn. (5)				

[a]Booms folded [b]Booms deployed [c]Range from Fig 6 [d]Not flight representative

The HP, its adapted EQUATOR-S version RHP and the NS model were all used to analyze the differences in flight and ground test results. The resonance tuning parameter is the nominal inertia ratio $\lambda_0 = I_Z/I_x$ with I_Z the inertia about the spin-axis and $I_x \cong I_y$ the lateral inertias about the perpendicular X and Y axis. The S/C nutation frequency $\Omega_0 = (\lambda_0 - 1)\omega_z$ with ω_z the S/C spin-rate. This frequency has to be tuned with the ND resonance frequency ω_0 which is is given by

$$\frac{\omega_0^2}{\omega_z^2} = \alpha \frac{R_0}{L_{eff}} \frac{a^2}{b^2} \tag{4}$$

with R_0 the ND mounting radius from to the spin-axis, $L_{eff} \approx L + 2a$ the effective damping length, L the liquid tube length, a the liquid tube radius, b the endpot radius whilst $\alpha = 1.5$ for the HP model and $\alpha = 2$ for the RHP and NS model.

The RHP model is interesting since the amount of EQUATOR-S damping liquid 14% is relatively high making it a case of provoking the issue of S/C quasi-rigidness which is assumed in the regular ND designs. To account for this a static (s) and dynamic (d) part of the inertia tensor is applied so $\boldsymbol{I} = \boldsymbol{I_s} + \boldsymbol{I_d}$. The model is based on the fluid mechanics theory from [18]. This author treats the stationary case of communicating vessels starting with the Bernoulli energy conservation along a streamline applying an incompressible liquid without damping. The extended case includes damping and oscillatory flows with the following restrictions

- Negligible dissipation losses in the endpots and transition parts of the liquid tubes
- Turbulence magnitude at the entrance and exit of the endpots is the same
- Instantaneous damp pressure equilibrium above the endpots.

A liquid inertia correction factor k_0 is implemented additionally which quantifies the fraction of effective viscous damping mass based on the HP velocity profile. It becomes significant at small a/b and ω_Z since

$$k_0 = \frac{4\nu L(\lambda_0 - 1)b^2}{R_0\omega_Z\, a^4} \tag{5}$$

with ν [m^2/s]the kinematic viscosity of the damping liquid. In case k_0 =100% the RHP model equals the HP model with a parabolic velocity profile except for the damper resonance frequency as worked out at Eqn. 4.

In the following sections the different models and their verification will be further explored and explained.

4.3 EQUATOR-S S/C Nutation

The EQUATOR-S ND design was driven by the RHP model [11] based on [18]. It enabled a compliance in the modeling results, air-bearing test results and flight nutation data of the 48 rpm spin-stabilized EQUATOR-S S/C. The RHP model derivation starts from the generic case of communicating vessels with, in addition, an inertia correction factor to account for the moving liquid mass in the dampers.

The determination of the flight operational damping time constant is shown in Fig -6 with the detected nutation angle versus the elapsed flight time. The RHP time constant model results are in good agreement with EQUATOR-S ground test and in orbit nutation data [12]. The application of the RHP model however, revealed substantial differences compared with the NS model results. These will be further analyzed to study the validation range of the diverse models.

First the EQUATOR-S results from [11, 12] were verified (model calibration) by embedding their RHP model in MathCad® code beside the NS as well. Taking their nominal inertia λ_o = 1.33 maximum damping design case with endpot diameter $2b$ = 46.08 mm and liquid tube diameter $2a$ = 10 mm as calibration case at the nutation frequency Ω_o = 1.661 rad/s, yields seamless model agreement. The two MathCad® models yield τ = 50.330 s (NS) and 50.332 s (RHP) versus the RHP value 50.33 s given by [12].

Flight Validation

In Fig 7 the flight results of RHP, NS and HP modeling are shown. The flight ND model is given by $2b$ = 45 mm and $2a$ = 9 mm. The predicted time constants differ slightly since τ = 69.3 s (RHP MathCad) versus 72.3 s [12] from the RHP model. The flight validation value is derived from the exponential decay in Fig 6 and yields τ_{flight} = 72 ± 6 s. The nominal NS model value τ = 61.3 s is read from Fig 7 at the nominal nutation frequency Ω_o = 1.661 rad/s and nominal inertia λ_o = 1.33 (red arrows). The time constant differences between the NS, HP and RHP model predictions at λ_o = 1.33 are within 15% whilst the flight value is within the range of the NS and HP model.

In the horizontal direction the optimal inertia ratios, i.e. with minimum damping time constant, all show significant off-sets compared to the desired nominal inertia λ_o = 1.33 which means a mistuned ND resonance condition. The RHP model shows a -3% off-set whilst the NS and HP model show a -5% offset. It is remarkable that the EQUATOR-S ND design data [11, 12] do not reveal an essential picture like Fig 7 with the RHP model damping time constant as a function of either the nutation frequency Ω or the spacecraft inertia ratio λ. The right picture shows the results for an increase of the effective length by $2a$ (one tube diameter) which is a good estimate for the additional damping from the inlet and outlet flows at the endpots. The differences are negligible due to the low value of $2{\cdot}a\,/\,L$ = 1.8%.

The results from Figs 7 and 8 as well as Eqn. (4) show that the ND mass, length and damping time constant could have been minimized since the liquid tube length can be reduced to -25% (RHP) and even -40% (HP and NS) to get a well-tuned condition at Begin-Of-Life (BOL) inertia conditions.

Ultimate Nutation Angle

The EQUATOR-S flight data show a minimum nutation angle < 0.1^0. No additional requirements, test data or test philosophy is given to investigate the ultimate damping angle as was done for the COS-B S/C to follow.

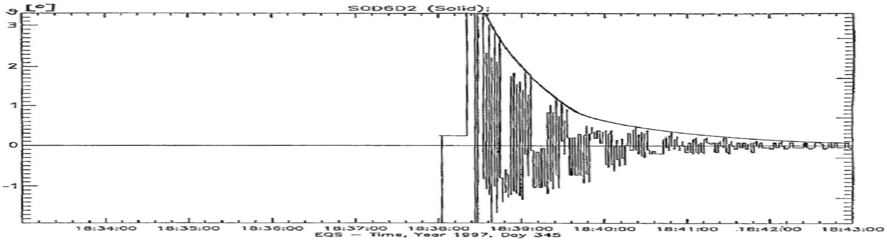
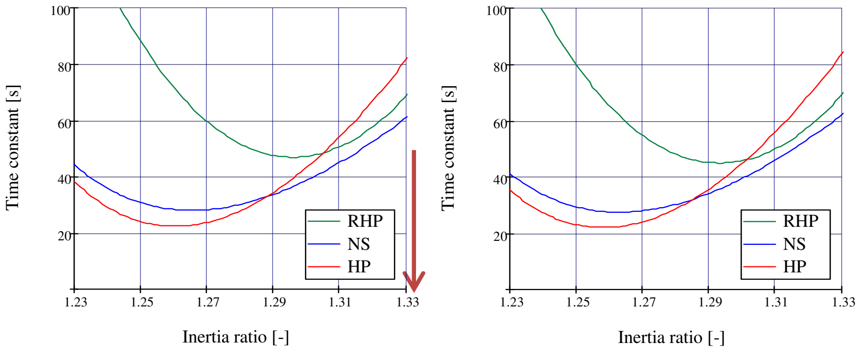


Fig. 6 The nutation angle attenuation of the EQUATOR-S S/C measured in flight yields the flight validation value $\tau_{\text{exp}} = 72 \pm 6$ s from its exponential decay [12]



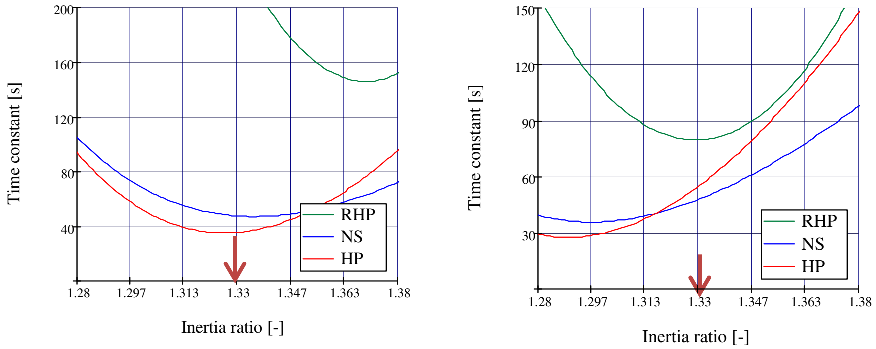
$L = L_0 = 500$ mm

$L = L_0 + 2\text{a} = 509$ mm

Fig. 7 EQUATOR-S S/C flight nutation time constant prediction at T = 12.7 ^{0}C by three ND models. The red arrow indicates the desired tuning condition. The right picture includes the in- and outlet flows yielding an increased effective length by 2a.

4.4 COS-B S/C Nutation

The EOL COS-B S/C flight nutation data were analyzed by [5]. The COS-B AOCS was equipped with two meridian NDs, shown in Fig 9. In the upper picture the liquid and vapor tube, as well as the endpots are partly disguised by its structural housing. The lower picture shows the A-A cross-section with the liquid tube (a = 4 mm) at the bottom side.



$L = L_0 + 2a = 395 + 9 = 404$ mm

$L = L_0 + 2a = 310 + 9 = 319$ mm

Fig. 8 EQUATOR-S S/C BOL **flight** nutation time constant prediction at T = 12.7 ^{0}C by three ND models. The red arrow indicates the desired tuning condition. The left picture shows the -25% reduced optimized length using the RHP model. The right picture shows the -40% reduced optimized length on basis of the HP and NS model.

Nutation experiments were executed during spring 1982 at the S/C End-Of-Life (EOL) cycle. The nutation was induced by thruster firings up to angles of 3.8 degree. The nutation behavior was extracted from the sun sensor telemetry data and analyzed by Fast Fourier Transform (FFT) analysis from the S/C oscillatory behavior, a method described by [20]. The ND flight performance complies well with the NS model predictions. The basic uncertainty in the modeling results is caused by inaccuracies in the Moments-Of-Inertia (MOI) values with less than 2% absolute error, temperature and position angle of the NDs. Taking these into account the re-analyzed results from the NS model [5] are:

- The nutation decays exponentially in all cases.
- No correlation exists between the initial nutation angle and the time constant.
- The in-orbit time constants, resulting from the invoked flight test nutation spin-rates 9.45 and 17.63 rpm, comply well with the results obtained by NS modeling and air-bearing tests.
- Air-bearing PTM results comply well at lower spin rates (8 rpm)
- At the high spin-rates 36.6 and 113.4 rpm the air-bearing tests predict less damping and consequently higher time constants.

The Fokker Space models as well as their test setup procedures were improved since the development of the COS-B ND according to [5] up to the standard used in the design and qualification of the Ulysses, FY-2 and Cluster nutation dampers.

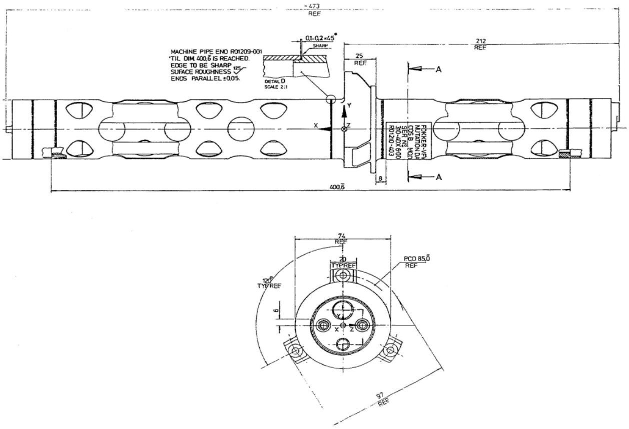
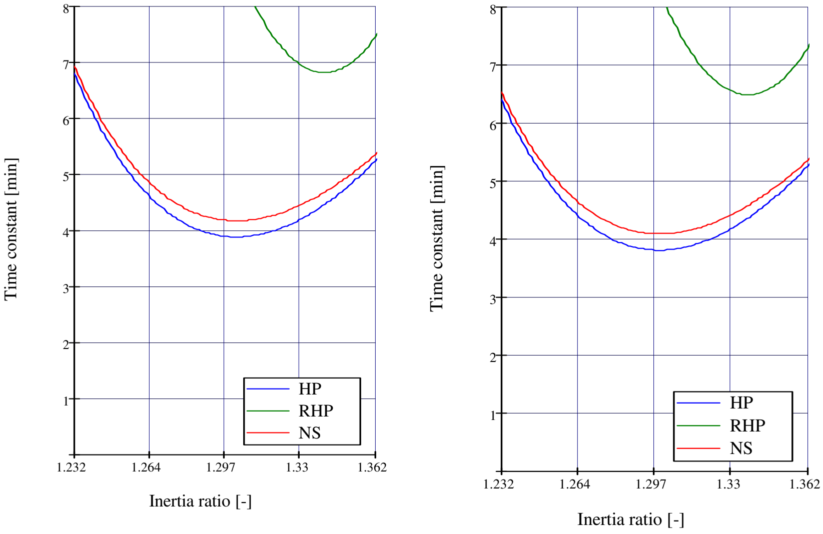


Fig. 9 Meridian COSB nutation damper built by (Fokker) Dutch Space [5]

Flight Validation

The end-of-life (EOL) space nutation experiments [5] were re-analyzed at the low spin-rate range $\omega_z = 9.45 \pm 0.06$ rpm and at the higher range $\omega_z = 17.63 \pm 0.02$ rpm.

In Fig 10 the HP, RHP and NS model results are shown at 9.45 rpm spin-rate with all other parameters nominal as defined by Table 2 and a ND position angle $\alpha = 40^0$ in the equatorial plane with respect to the X-axis. The NS and HP model

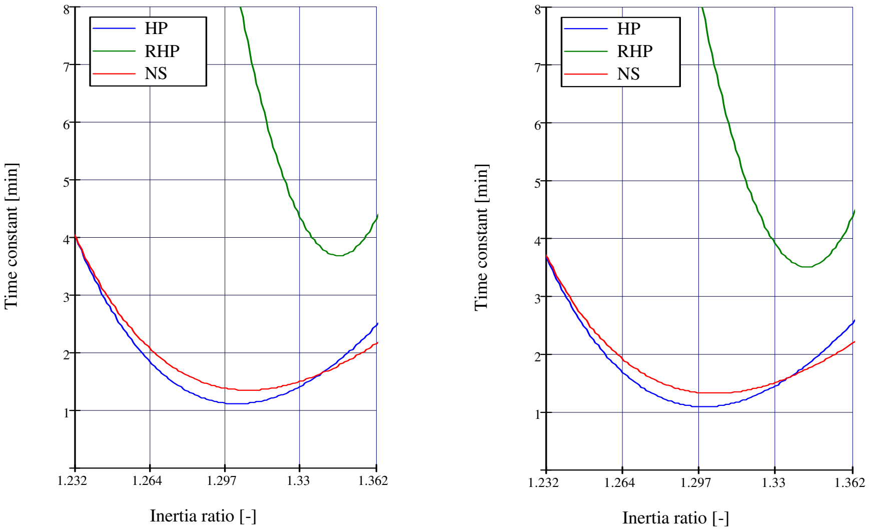


$L = L_0 = 400$ mm $\qquad$ $L = L_0 + 2a = 408$ mm

Fig. 10 COS-B FM nutation damping prediction by the HP, RHP and NS models at the S/C spin-rate 9.45 rpm. The right picture includes the in- and outlet flows accounting for an increased effective length by 2a.

results show a resonance shift of +5% beyond the specified EOL $\lambda_o = 1.232$ value whilst the RHP model shows a shift of +9%.The RHP curve differs significantly with less bandwidth. The NS result at the EOL value $\lambda_o = 1.232$ agrees seamless with the space experiment results [5], i.e. the flight damping time constant $\tau = 7.0$ min with standard deviation $\sigma = 0.5$. Although the NS and HP curve almost coincide as expected at low spin-rates they are not tuned to the End-Of-Life (EOL) design.

The same shifts hold for the model predictions of the nutation performance at 17.63 rpm spin-rate given in Fig 11. The result at $\lambda_o = 1.232$ matches again seamless with the results given by [5] since the extracted flight damping time constant $\tau = 4.2$ min with standard deviation $\sigma = 0.3$. In both cases the RHP model and its time constant prediction is far off and does not apply to the COS-B design and its flight performance validation.



$L = L_0 = 400$ mm

$L = L_0 + 2a = 408$ mm

Fig. 11 COS-B FM damping prediction by the HP, RHP and NS models at 17.63 rpm spin-rate. The right picture includes the in- and outlet flows accounting for an increased effective length by 2a.

The design and qualification procedures have been significantly improved since the development of the COS-B ND up to the standards of the Cluster ND. The design can be typically tuned within 0.5% using these standards. The Dutch ND design heritage is nowadays concentrated at Aeronamic B.V. in Almelo, The Netherlands.

Ultimate Nutation Angle

The COS-B ND contact angle hysteresis below the required 5” flight nutation angle is not experimentally detectable. Therefore a scaled experiment was executed with a glass-water PTM model. The experiments were executed down to air-bearing angular motion levels representative for flight nutation angles less than 0.2”. Because of the low accuracy, caused by air-bearing turbulence flow phenomena in this range, a large number of repetitive experiments were necessary. The experimental values, given in Fig 12, are valid in the vicinity of the flight spin-rate $\omega_z = 0.661$ rad/s. The results were re-analyzed with conclusions:

- The measured (marked) normalized damping values $\bar{P}/a_0^2$ are determined at 1” and 0.2” equivalent flight nutation angles. The 1”and 0.2”series do not show a significant difference (magnitude) in damping performance. At these airbearing amplitudes the typical damping uncertainty is about 30%.
- A resonance shift is not recognized going from the 1” to the 0.2” data series. This shows that a second time constant from contact angle hysteresis is absent for the glass model.
- The ultimate proof was delivered by refined tests using PTM models with the same endpot materials and surface treatment as the FM, e.g. during the FY-2 and Cluster S/C ND qualification tests. The ultimate linear damping limit proved to be beyond the values predicted by the theory from [13].

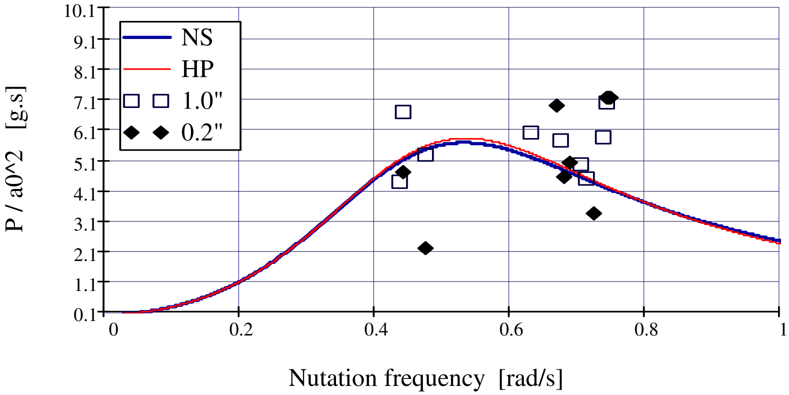


Fig. 12 Re-analyzed NS model results and PTM experiments of the COS-B ND behavior at the extreme small flight nutation angle 1.0” and 0.2” in the vicinity of 0.661 rad/s with nominal flight liquid temperature T = 7.5 ^{0}C. The vertical scale shows the normalized (by the squared nutation acceleration component) average damping dissipation rate.

5 Conclusions

The ND verification and validation data of the following three classes of S/C were investigated:

- The 5 rpm Ulysses and 15 rpm Cluster S/C flight data mention nominal ND performance but do not comprise explicit data or in-orbit invoked nutation experiments.
- The 100 rpm MSG and FY-2 S/C data yield in-orbit AOCS compliance with the ND designs.
- The EQUATOR-S at 48-50 rpm and COS-B S/C at 9.45-17.63 rpm records yield very interesting data.

A systematic model comparison for the latter two was conducted yielding the following results:

- The relative amount of COS-B liquid mass, with requirement $m/M << 1$ as prerequisite of a quasi-rigid S/C configuration, is about one third of the EQUATOR-S value. The "free liquid" k_0 factor accounting for the applicability of the RHP model is comparable.
- The EQUATOR-S RHP (Refined Hagen-Poiseuille Model) flight damping time constant prediction is within 15% compared to the NS and HP model. Peculiar is the fact that the RHP model which drove the ND design, shows a -3% resonance offset from the design value $\lambda_o = 1.33$.
- The EQUATOR-S ND design data [11, 12] do not reveal an essential picture with the modeled time constant as a function of the S/C inertia ratio λ. The frequency tuning of the ND is not mentioned but only the agreement in damping time constant between the RHP model results, air-bearing verification data and nominal flight results. The considerable -3% offset from the ND resonance frequency (with consequently optimum damping) indicates that the mass of the EQUATOR-S ND could have been reduced.
- The plexi-glass EQUATOR-S PTM model has totally different end pot materials, surface treatment (if any) and RMS roughness compared to the Aluminum FM. Despite these differences the PTM tests show a good RHP based agreement with the FM flight and ground test damping time constant results.
- The Dutch Space COS-B flight nutation data have been embedded in the HP, RHP and NS model predictions. At the spin-rates 9.45 and 17.63 rpm excellent damping performance agreement was found with the developed HP and NS MathCad® coded models. The RHP model shows an additional resonance off-set of 4% compared to the NS and HP modeling. In both cases the RHP model and its time constant prediction is far off and does not comply with the COS-B design and its flight performance validation.

- The COS-B ultimate damping angle tests with the PTM glass model did not reveal a second time constant due to end pot hysteresis phenomena. This is concluded after studying and re-analyzing the results of a high number of repetitive tests at 1.0 and 0.2” ultimate angles in air-bearing tests. An ultimate nutation angle analysis for the EQUATOR-S ND design is not given.
- It can be concluded that the NS model derived in the co-rotating S/C body reference frame is the backbone of ND design on basis of its superior V&V status.

References

1. Agrawal, B.: Dynamic Characteristics of Liquid Motion in Partially Filled Tanks of a Spinning Spacecraft. AIAA Journal of Guidance, Control, and Dynamics 16(4), 636–640 (1993)
2. Ancher, L.J., van den Brink, H., Pouw, A.: Asymmetric oscillation of a passive nutation damper. In: Proc. of the CNES-ESA Conf. on Attitude Control of Space Vehicles – Technological and Dynamical Problems Associated with the Presence of Liquids, ESA SP-129, pp. 179–183. Fokker-VF, The Netherlands (1977)
3. van Bakel, C.: On the behaviour of the tube-with-endpots nutation damper – UCN & Eindhoven University of Technology, WFE document 93.105 (August 1993)
4. Batchelor, G.K.: An Introduction to Fluid Dynamics, ch. 4. Cambridge University Press (1980)
5. Bongers, E.: In-orbit Experiments on COS-B nutation damper performance, ESA report TR-R-84-037 (1984)
6. Boyd, D., Kuiper, H.: The Compound Parabolic/elliptic Lightshades: Near-Optimal Shading for Cold Radiators, SAE Technical Paper no. 2000-01-2278 (2000)
7. COS-B S/C mission description. COS-B S/C (2), ESA bulletin no. 2, special (August 1975),
   http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=31447,
   http://www.esa.int/esaSC/120375_index_0_m.html
8. Crellin, E.B.: A model of the performance of a fluid-in-tube nutation damper, Estec Working Paper 1315 (March 1982)
9. Crowden J.B.: Fax No-92927190 from Marconi Space Systems - Aerospatiale Cannes, Nutation Dampers for MSG (1990)
10. Ebert, K., Reger, G.: Sloshing Analysis for Meteosat Second Generation. In: Proc. 2nd European Spacecraft Propulsion Conference, ESA SP-398 (August 1997)
11. Häusler, B., Eidel, W.: Ein Flüssigkeitsnutationdämpfer für den Satellit EQUATOR-S, Forschungsbericht: LRT-WE-9-FB-1, Institut für Luft und Raumfahrttechnik, Universität der Bundeswehr München, Neubiberg, Germany (2001)
12. Häusler, B., Eidel, W., Stöcker, J.: Ein Flüssigkeitsnutationdämpfer für den Satellit EQUATOR-S. Forschung im Ingenieurwesen 67, 27–39 (2002)
13. Hong, J.Z.: Residual Nutation Angle of Satellites with Viscous Nutation Dampers. Acta Astr. 15(1), 1–7 (1987)
14. Hughes Peter, C.: Spacecraft Attitude Dynamics. In: Chapter 4 - Attitude Dynamics of a Rigid Body, Chapter 5 - Effect of Internal Energy Dissipation on the Directional Stability of Spinning Bodies. John Wiley & Sons (1986)

15. Loitsyanskii, L.G.: Mechanics of Liquids and Gases, pp. 496–509. Pergamon Press (1966)
16. Luengo, O.: Dynamic Model of the MSG Attitude, EUMETSAT (GMV consultant). In: ESA Proceedings, ESASP.548-313L (2004)
17. Papanastasiou, T.C., Georgiou, G.C., Alexandrou, A.N.: Viscous Fluid Flow, pp. 243–253. CRS Press (1999)
18. Truckenbrodt, E.: Strömungsmechanik. Springer, Heidelberg (1996)
19. Truckenbrodt, A., Schultysik, B., Mehltretter, J.P.: Nutationsdämpferfür den AMPTE-IRM Satelliten: Abschlu*β*bericht, Technology Consulting Institute for Applied Research GMBH (West Germany), TC-no. 017-Tru-SCh0/sa (December 9, 1982)
20. Wertz, J.R.: Spacecraft Attitude Determination and Control, 73rd edn. Springer (2003)
21. Weymiens, B., Oremus, R., Battrick, B., et al.: Meteosat Second Generation. ESA Publications, ESA BR-153 (1999)
22. Willmore, A.P.: A Proposal for an X-Ray Experiment for Cos-B. In: Non-Solar X- and Gamma-Ray Astronomy, pp. 50–53. IAU (1970)
23. Womersley, J.R.: Method for the calculation of velocity, flow and viscous drag in arteries when the pressure gradient is known. Journal of Physiology 127, 533–563 (1955)

Fault-Tolerant Spacecraft Magnetic Attitude Control

Aviran Sadon and Daniel Choukroun

Abstract. This work is concerned with the development of a suboptimal control algorithm for Markovian jump-linear systems, and its application to fault-tolerant spacecraft magnetic attitude control. For completeness, the jump-linear quadratic optimal controller with full state and mode information is presented. Relaxing the assumption of perfect mode information, a similar optimal control problem is formulated where the mode is observed via discrete measurements. The elements of the measurement matrix, i.e. the probabilities for correct and wrong mode observations are assumed known. The optimal controller is developed, which requires an exponentially growing computational burden, and a suboptimal controller is proposed that only requires knowledge of the current mode measurement. This controller is finite memory and possess some of the classical linear quadratic regulator features such as the linear state feedback structure and a state quadratic optimal cost-to-go. The performances of the suggested algorithm are illustrated through extensive Monte-Carlo simulations on a simple numerical example. A realistic fault-tolerant spacecraft magnetic attitude controller is developed based on the proposed approach. The attitude controller succeeds in mitigating the destabilizing effect of corrupted mode observations while being computationally efficient.

Aviran Sadon
Mechanical Engineering Department,
Faculty of Engineering Sciences,
Ben-Gurion University
e-mail: sadonav@post.bgu.ac.il

Daniel Choukroun
Space Systems Engineering Chair,
Faculty of Aerospace Engineering,
Delft University of Technology
e-mail: d.choukroun@tudelft.nl

1 Introduction

Jumping parameters processes, also called step processes, are commonly used in order to model abrupt parametric variations in the models of dynamical systems due for instance to sensors or actuators failures. Such processes also allow to incorporate state discontinuities in the differential equations that govern the system dynamics and they are useful when approximating non-linear systems by a set of linearized models [1]. Research focused on the development of optimal control algorithms when both the state of the system and the value of the step process ("the mode") are instantaneously available. Stochastic minimum principles were given for non-linear [2] and linear [3] continuous-time systems where the jumping process is modeled as a Markov step processes. Further works on quadratic regulation of Jump Linear systems investigated the influence of additive diffusive process noise on the optimal control [4], the case where the mode transition probability matrix is a stochastic variable [5], and where the state discontinuities have random amplitudes [6]. Following the stochastic Dynamic Programming approach, quadratic regulators of Jump Linear systems were developed in the continuous-time [7] and discrete-time [8] settings. The working assumptions in Ref. 7 were that full state and mode information was available and that a Markov feedback control law was sought. The jumping parameter was modeled as a finite-state continuous-time homogenous Markov step process with ν possible values and a known differential transition matrix. The highlights of that work are that the control is linear in the dynamic state, with gain parameters that are determined by the simultaneous backward propagation of a set of ν coupled Riccati differential equations, where the coupling captures the effect of the stochastic mode-switching on the state dynamics. These gains, however, depend on the current value of the mode. In applications, the mode variations would represent parametric variations in linear dynamics, and such variations are usually not directly accessible to physical measurements. For this reason, the model described is of limited practical interest. On the other hand, when the assumption of full information is relaxed, optimal control and estimation solutions are known to yield algorithms with exponentially growing memory [9]. This motivated the development of suboptimal but practical algorithms that can operate on systems with process and measurement noises and where the state only is observed. To achieve practicality such algorithms rely on approximations in the hypothesis pruning or in the expression for the optimal return function [10]. Such a sub-optimal controller is suggested in Ref [11] under the assumptions of known continuous state while the mode is detected after some random delay, which probabilistic model is given.

This work is concerned with the development of an approximate jump-linear quadratic controller with partial mode information and its application to spacecraft attitude control. This work briefly reviews the general jump-linear quadratic controller under full state and mode information. Relaxing the assumption of perfect mode information, the mode is observed via discrete-valued measurements with known conditional probability distribution. The optimal solution is presented, the conflict between optimality and practicality is noted, and a computationally efficient suboptimal algorithm is suggested. The proposed algorithm has got some of the classical Linear Quadratic regulator features: recursion, linear state feedback, state quadratic optimal cost-to-go, which are direct consequences of the nested property of the information patterns and of the full continuous-state information. The gain computation, however, depends on the specific information structure. The proposed suboptimal controller is a finite memory controller and a look-up table for the gains can be computed off-line. Comparative results of an extensive Monte-Carlo simulation for a simple system illustrate the efficiency of the proposed algorithm that mitigates the destabilizing effect of corrupted mode observations. The detailed developments of the optimal and the proposed suboptimal jump-linear quadratic controllers are presented. The novel approximate controller is compared to other hybrid systems controller. The modeling of failures in the context of spacecraft attitude dynamics are described and their embedding in the framework of jump-linear system is addressed. Focus is made on failures in magnetic sensing and actuators, which are common on-board small satellites. Simulations for an Earth orbiting three-axis magnetically stabilized small spacecraft is performed under realistic modeling assumptions and results from extensive Monte-Carlo simulations are presented.

2 Jump-Linear Quadratic Control Problem

Consider the following discrete time jump-linear dynamical system

$$\mathbf{x}_{k+1} = A(y_k)\mathbf{x}_k + B(y_k)\mathbf{u}_k + \mathbf{w}_k \qquad k = 0, 1, \ldots, N-1 \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{w}_k \in \mathbb{R}^n$, $A(y_k) \in \mathbb{R}^{n\times n}$, $B(y_k) \in \mathbb{R}^{n\times m}$, $\{\mathbf{w}_k :$ k=0,1,...,N-1$\}$ is a zero mean white sequence, with known covariance matrices, W_k, and $\{y_k : k = 0, 1, \ldots, N-1\}$ is a scalar autonomous Markov chain with finite state space $S_y = \{1, 2, \ldots, \nu\}$ and transition matrix P; that is,

$$P[i, j] = p_{ij} = Pr(y_{k+1} = j \mid y_k = i) \tag{2}$$

The sequences $\{y_k\}$, and $\{\mathbf{w}_k\}$ are assumed to be independent one from the other and from the initial conditions $\mathbf{x}_0$. The system represented by the joint state $\{\mathbf{x}_k, y_k\}$, which dynamics is governed by Eq. (1) and by the above assumption on $\{y_k\}$, belongs to the class of Markovian jump-linear systems [7]. Typically, the continuous states, $\mathbf{x}_k$, represent physical quantities while the discrete state, y_k, represents a logical state that describes a “mode” of operation of the system. In the context of fault modeling, occurrences of failures are modeled by the switch of the Markov chain state, y_k, from one value to another, according to a priori transition probabilities. Let J denote the following quadratic cost function

$$J = E\left\{\sum_{k=0}^{N-1} \|\mathbf{x}_{k+1}\|^2_{Q(y_k)} + \|\mathbf{u}_k\|^2_{R(y_k)}\right\} \tag{3}$$

where $Q(y_k) \in \mathbb{R}^{n\times n}$, $Q(y_k) \geq 0$, $R(y_k) \in \mathbb{R}^{m\times m}$, $R(y_k) > 0$, for $k = 0, 1, \ldots, N-1$. The jump-linear quadratic control problem (JLQ) [13, Ch. 2] consists in finding the sequence $\{\mathbf{u}_k\}$ that minimizes the cost J in Eq. (3) subject to Eq. (1) and under the assumptions of state and mode full information. The set of admissible controls is the set of functions of the available information. This type of controller is called "fault-tolerant" because its design simultaneously accounts for failure-driven dynamics of the physical states. The random variables $A(y_k)$, $B(y_k)$, $Q(y_k)$, and $R(y_k)$, will be denoted by A_k, B_k, Q_k, and R_k, respectively. The solution to the full information JLQ problem [2, 8] yields a linear controller where the gains are computed as follows:

Initialize the algorithm with

$$K\{y_N, N\} = 0 \qquad y_N = 1, 2, \ldots, \nu \tag{4}$$

For $k = N-1, N-2, ..., 0$, and $y_k = 1, 2, \ldots, \nu$, compute,

$$\tilde{K}\{y_k, k+1\} = \sum_{j=1}^{\nu} p_{y_k,j}\, K\{j, k+1\} + Q_k \tag{5}$$

$$M\{y_k, k\} = \left(R_k + {B_k}^T \tilde{K}\{y_k, k+1\} B_k\right)^{-1} B_k^T \tilde{K}\{y_k, k+1\} A_k \tag{6}$$

$$K\{y_k, k\} = {A_k}^T \tilde{K}\{y_k, k+1\} (A_k - B_k M\{y_k, k\}) \tag{7}$$

Notice that Eqs. (5)- (7) are functions of the mode y_k and consist, thus, of a set of ν coupled backward Riccati-like equations. These computations yield ν sequences of gain matrices $M\{y_k, k\}$, $k = 0, 1, \ldots, N-1$, which are mode dependent and can be computed off-line. The control algorithm is then applied as follows: at each time step k, y_k and $\mathbf{x}_k$ are known and the adequate gain matrix $M\{y_k, k\}$ is selected. The control, $\mathbf{u}_k^*$, is

$$\mathbf{u}_k^* = -M_k\, \mathbf{x}_k \tag{8}$$

where M_k denotes M$\{y_k, k\}$, for simplicity. Notice that the linear structure of the control law Eq. (8) is an outcome of the optimal control derivation and not an assumption. As a by product, the optimal cost-to-go is computed as follows:

$$J_k^* = \|\mathbf{x}_k\|_{K_k}^2 + \sum_{i=k}^{N-1} tr\left(W_i \tilde{K}_{i+1}\right) \tag{9}$$

where $K_k, \tilde{K}_{i+1}$ denote, respectively, the matrices $K\{y_k, k\}$ and $\tilde{K}\{y_{k}, _{k+1}\}$, from Eqs. (4), (7). The optimal cost-to-go J_k^* is computed forward along with the optimal control sequence.

3 JLQ Control with Discrete Mode Observations

3.1 Problem Formulation

Consider the same discrete time jump-linear dynamical system as given in Eq. (1) with the same assumptions on the dynamics of the mode, on the noises, and on the state-$\mathbf{x}_k$ information. It is also assumed that the mode y_k is observed and that the mode measurements are discrete-valued random variables. Let $\{z_k : k = 0, 1, \ldots, N-1\}$ denote the sequence of observations of the mode y_k and let $S_z = \{1, 2, \ldots, \nu\}$ denote the finite state space of each observation z_k. The mode measurement model is characterized by a priori probabilities of correct and missed detections as follows

$$\Pi_k[i,j] \overset{\triangle}{=} \Pr\left(z_k = i \,| y_k = j\right) \tag{10}$$

Considering the following quadratic cost function,

$$J = E\left\{\sum_{k=0}^{N-1} \|\mathbf{x}_{k+1}\|_{Q(y_k)}^2 + \|\mathbf{u}_k\|_{R(y_k)}^2\right\} \tag{11}$$

where $Q(y_k) \in \mathbb{R}^{n\times n}$, $Q(y_k) \geq 0$ and $R(y_k) \in \mathbb{R}^{m\times m}$, $R(y_k) > 0$, $k = 0, 1, \ldots, N-1$, a sequence of control vectors $\{\mathbf{u}_k\}$ is sought that minimizes J subject to Eq. (1), under the assumption of full information on $\mathbf{x}_k$. The set of admissible controls is the set of functions of the present and past history of the state $\mathbf{x}_k$ and of the observations z_k.

3.2 Optimal and Suboptimal Solution

The optimal solution to this problem was developed via Dynamic Programming (Appendix A) and is summarized next. The detailed developments will be provided in the final manuscript.

Initialize the algorithm with

$$\tilde{S}_N = 0 \tag{12}$$

For $k = N-1, N-2, ..., 0$, compute

$$S_k\left(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}, y_k\right) = \tilde{S}_{k+1} + Q_k \tag{13}$$

$$\Gamma_k\left(\mathcal{X}^k, \mathcal{Z}^k\right) = \left(\overline{B_k^T S_k B_k + R_k}\right)^{-1} \overline{B_k^T S_k A_k} \tag{14}$$

$$\tilde{S}_k\left(\mathcal{X}^k, \mathcal{Z}^k\right) = \overline{A_k^T S_k A_k} - \overline{A_k^T S_k B_k} \Gamma_k \tag{15}$$

where $\mathcal{X}^k \triangleq \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and $\mathcal{Z}^k \triangleq \{z_0, z_1, \ldots, z_k\}$ denote the histories of $\{\mathbf{x}_k\}$ and $\{\mathbf{z}_k\}$, respectively. The variables $\overline{F}$ in Eqs. (14) and (15) are defined as follows:

$$\overline{F}(\mathcal{X}^k, \mathcal{Z}^k) \triangleq E\{F \mid \mathcal{X}^k, \mathcal{Z}^k\} \tag{16}$$

Considering the conditioning sequence in Eq. (16) the computation of these conditional expectations would require a growing memory size. Two approximate solutions are examined, first only the mode observation histories $\mathcal{Z}^k$ are retained, i.e., by computing the following conditional expectation

$$\tilde{\overline{F}}(\mathcal{Z}^k) = E\{F \mid \mathcal{Z}^k\}$$

The computations of $\tilde{\overline{F}}(\mathcal{Z}^k)$ are performed as follows (Proof in Appendix B):

$$\tilde{\overline{F}}(\mathcal{Z}^k) = \sum_{z_{k+1} \in S_z} \sum_{y_k \in S_y} F\left(y_k, \mathcal{Z}^k, z_{k+1}\right) \\ \times \frac{\sum_{y_{k+1}}^{\nu} \left\{\Pr\left(z_{k+1}|y_{k+1}\right) \Pr\left(y_{k+1}|y_k\right)\right\} \sum_{y_{k-1}}^{\nu} \left\{\Pr\left(y_k|y_{k-1}\right) \Pr\left(y_{k-1}, z_k|\mathcal{Z}^{k-1}\right) \Pr\left(y_{k-1}\right)\right\}}{\sum_{y_{k-1}}^{\nu} \left\{\Pr\left(y_{k-1}, z_k|\mathcal{Z}^{k-1}\right) \Pr\left(y_{k-1}\right)\right\}} \tag{17}$$

which yields the following expressions as a function of the model parameters:

$$\tilde{\overline{F}}(\mathcal{Z}^{k-1}, z_k = \zeta) = \sum_{i=1}^{\nu} \sum_{j=1}^{\nu} F(j, i, \mathcal{Z}^k) \tag{18}$$

$$\times \frac{\sum_{r=1}^{\nu} \left\{\pi_{\text{ir}}^{(k+1)} p_{\text{jr}}\right\} \sum_{l=1}^{\nu} \left\{p_{\text{lj}} \Lambda_{\zeta\text{l}}^{(k-1)}(\mathcal{Z}^{k-1}) \Pr\left(y_{k-1} = l\right)\right\}}{\sum_{m=1}^{\nu} \left\{\Lambda_{\zeta\text{m}}^{(k-1)}(\mathcal{Z}^{k-1}) \Pr\left(y_{k-1} = m\right)\right\}} \tag{19}$$

where p_{ij} denotes the element at location i, j in the transition matrix P, $\pi_{\text{ij}}^{(k)}$ denotes the element at location i, j in the matrix Π_k and $\Lambda_{\zeta\text{l}}^{(n)}(\mathcal{Z}^n)$ can be computed as as follows:
Initialize the computation with

$$\Lambda_{\text{ij}}^{(0)}(\zeta) = \frac{\sum_{r=1}^{\nu} \left\{\pi_{\text{ir}}^{(1)} p_{\text{jr}}\right\} \pi_{\zeta\text{j}}^{(0)} \Pr\left(y_0 = j\right)}{\sum_{\tau=1}^{\nu} \left\{\pi_{\zeta\tau}^{(0)} \Pr\left(y_0 = \tau\right)\right\}} \tag{20}$$

For $k = 1, 2, ..., n$, compute

$$\Lambda_{\mathrm{ij}}^{(k)}(\mathcal{Z}^{k-1}, z_k = \zeta) = \frac{\sum_{r=1}^{\nu}\left\{\pi_{\mathrm{ir}}^{(k+1)} p_{\mathrm{jr}}\right\} \sum_{l=1}^{\nu}\left\{p_{\mathrm{lj}} \Lambda_{\zeta \mathrm{l}}^{(k-1)}(\mathcal{Z}^{k-1}) \Pr\left(y_{k-1} = l\right)\right\}}{\sum_{m=1}^{\nu}\left\{\Lambda_{\zeta \mathrm{m}}^{(k-1)}(\mathcal{Z}^{k-1}) \Pr\left(y_{k-1} = m\right)\right\}} \tag{21}$$

The resulting algorithm for that approximation also has substantial computational burden of about ν^k, which leads to the second suggested approximate solution by retaining only the current mode observation z_k, i.e., by computing the following conditional expectation

$$\overline{\overline{F}}(z_k) = E\{F \mid z_k\}$$

The computations of $\overline{\overline{F}}(z_k)$ are performed as follows (Proof in Appendix B):

$$\begin{aligned} \overline{\overline{F}}\left(z_k\right) = & \sum_{z_{k+1} \in S_z} \sum_{y_k \in S_y} F\left(y_k, z_{k+1}\right) \\ & \times \frac{\Pr\left(z_k | y_k\right) \Pr\left(y_k\right) \sum_{y_{k+1} \in S_y}\left\{\Pr\left(z_{k+1} | y_{k+1}\right) \Pr\left(y_{k+1} | y_k\right)\right\}}{\sum_{y_k \in S_y}\left\{\Pr\left(z_k | y_k\right) \Pr\left(y_k\right)\right\}} \end{aligned} \tag{22}$$

which yields the following expressions as a function of the model parameters:

$$\overline{\overline{F}}\left(\zeta\right) = \sum_{i=1}^{\nu} \sum_{j=1}^{\nu} F(j, i) \frac{\pi_{\zeta \mathrm{j}}^{(k)} \sum_{r=1}^{\nu}\left\{p_{\mathrm{rj}}^{(k)} \Pr\left(y_0 = r\right)\right\} \sum_{l=1}^{\nu}\left\{\pi_{\mathrm{il}}^{(k+1)} p_{\mathrm{jl}}\right\}}{\sum_{m=1}^{\nu}\left\{\pi_{\zeta \mathrm{m}}^{(k)} \sum_{\mu=1}^{\nu}\left\{p_{\mu \mathrm{m}}^{(k)} \Pr\left(y_0 = \mu\right)\right\}\right\}} \tag{23}$$

for $z_k = \zeta = 1, 2, \ldots, \nu$, where $p_{\mathrm{ij}}^{(k)}$ denotes the element at location i, j in the power matrix P^k and $\pi_{\mathrm{ij}}^{(k)}$ denotes the element at location i, j in the matrix Π_k. The rationale for that approximation is that the resulting algorithm has got the same computational burden as the full information JLQ controller while, partly, accounting for the imperfect information. The gain computations consist of Eqs. (12) - (15) where $\overline{F}$ is replaced by $\overline{\overline{F}}$ from Eq. (23). Notice that the expressions for $\overline{\overline{F}}$ are functions of z_k, where $z_k = 1, 2, \ldots, \nu$. These computations thus involve a set of ν coupled backward matrix equations. These computation are performed off-line and the gains are stored in a look-up table with ν possible values at each step. The resulting control is linear:

$$\mathbf{u}_k = -\Gamma_k \ \mathbf{x}_k \tag{24}$$

As a by-product, the algorithm produces the cost-to-go:

$$J_k^* = \|\mathbf{x}_k\|^2_{\widetilde{S}_k} + \sum_{i=k}^{N-1} tr(W_i S_i) \tag{25}$$

since Q_k and R_k are design parameters, they can be chosen via trial and error such as to achieve desired performances. Throughout the paper this algorithm will be addressed as the proposed suboptimal JLQ controller. For the special case where there are no errors in the mode observations, it is straightforward to show that the proposed algorithm boils down to the standard JLQ controller.

3.3 Numerical Simulation

The performance of the proposed JLQ control algorithm are illustrated over a simple system via extensive Monte-Carlo simulations. A scalar jump-linear system with three possible values of the mode is considered. The dynamics equation parameters and the cost function weights are given Table 1.

Table 1 System Parameters and Cost Function Weights

y_k	A_k	B_k	Q_k	R_k
1	0.1	1	5	1
2	0.9	2	4	2
3	4	4	2	0.5

The transition probability matrix of the Markov chain y_k is given as follows:

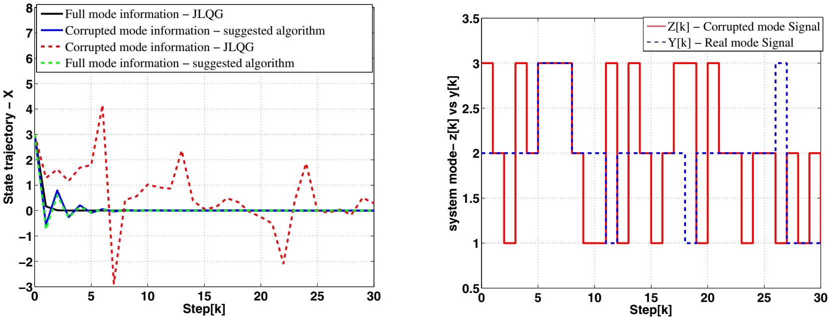
$$P = \begin{pmatrix} 0.6 & 0.1 & 0.3 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix} \tag{26}$$

The a priori probabilities for correct and for wrong detections are given as follows:

$$\Pi = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{pmatrix} \tag{27}$$

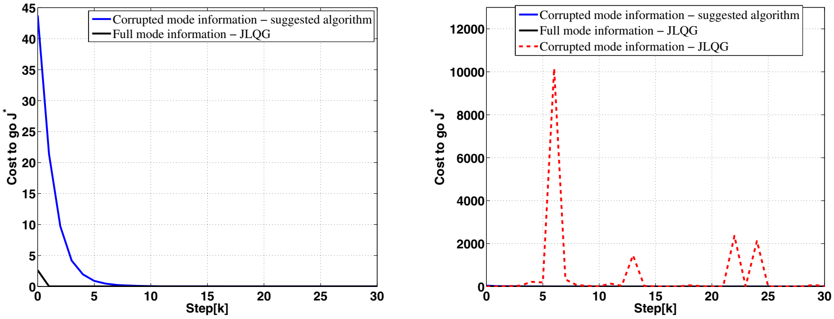
The initial state was chosen to be $x_0 = 3$, the process noise was a zero-mean Gaussian white sequence with a standard deviation of 0.05 and the initial mode distribution was $Pr(y_0) = [1/3, 1/3, 1/3]$. Four simulation cases were examined. In Case 1, there are perfect state and mode measurements and the standard JLQ algorithm is applied. In Case 2, the mode measurement sequence is corrupted, according to the mixing matrix Π in Eq. (27), and the proposed suboptimal JLQ controller is implemented. In Case 3, the novel algorithm is applied in presence of perfect state and mode information. The latter checks the conservativeness of the proposed algorithm. In Case 4, the standard JLQ algorithm is applied in presence of corrupted mode measurements. This clearly illustrates the motivation for the proposed work. For each case 1000 Monte-Carlo simulation runs were performed along a time space of 30 steps and their averages were computed for each step of the state cost-to-go histories. The results are summarized in Figs. 1 and 2.

Another simulation Case was examined, Case 5, in this case the mode measurement sequence is corrupted and the suboptimal JLQ algorithm based on the increasing mode measurement history is applied (Eq. (19)). Comparing that case to Case 2 show that the novel algorithm achieve fairly close results with much less computational burden. For each case, 1 2 and 5, 10000 Monte-Carlo simulation runs were performed along a short time space of 9 steps, due



(a) State Trajectories (b) Mode Sequences

Fig. 1 Monte-Carlo means of the state trajectories and an example for the mode sequences



(a) Cost-to-Go J^* for case 1 and case 2 (b) Cost-to-Go J^* for cases 1,2 and 4

Fig. 2 Monte-Carlo means of the Cost-to-Go J^*

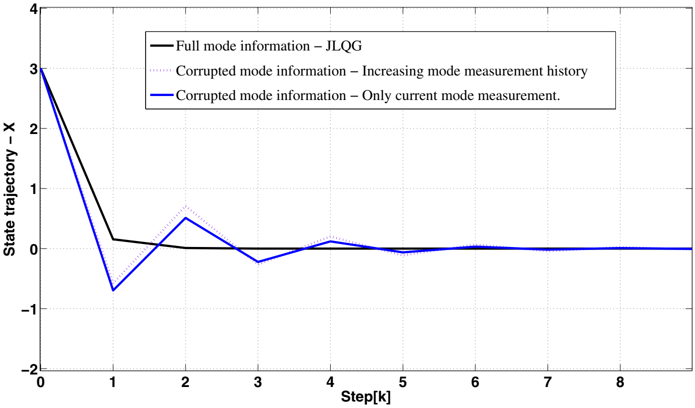


Fig. 3 Monte-Carlo means of the state trajectories

to the substantial computational burden of Case 5, and their averages were computed for each step of the state. The results are summarized in Fig. 3.

4 Fault-Tolerant Spacecraft Magnetic Attitude Control

In this section, the efficiency of the proposed fault-tolerant control approach is illustrated on a spacecraft attitude control problem. The case considered is of a very small satellite flying around the earth on a Low-Earth-Orbit, with a mass of a few kilograms. Three Cartesian coordinate reference frames are considered. The first is the inertial ($\mathcal{I}$) frame, its origin is at Erth's center, the z axis coincide with Earth's rotational axis, the x axis point towards a fixed celestial point and y axis is completes a right hand orthogonal frame. The second reference frame is the orbit ($\mathcal{O}$) frame, its origin is at the spacecraft center of mass, the z axis is directed towards nadir, x axis is along the spacecraft orbital velocity vector and y axis also completes a right-handed frame. The third coordinate frame is the body ($\mathcal{B}$) frame, it is also centered at the spacecraft center of mass and its axes coincide with the spacecraft principal axes. The satellite is assumed to be equipped with magnetic actuators only in order to regulate its nadir-pointing attitude. Magnetic actuators, a.k.a magnetorquers, are typical to small spacecraft close enough to the earth such that the magnetic field intensity allows efficient torques. The dynamical model of the satellite is as follows:

$$J\dot{\boldsymbol{\omega}}^{\mathcal{B}\mathcal{I}} + \boldsymbol{\omega}^{\mathcal{B}\mathcal{I}} \times J\boldsymbol{\omega}^{\mathcal{B}\mathcal{I}} = \boldsymbol{m} \times \boldsymbol{b}_{\mathcal{B}} + \boldsymbol{\tau}_g + \boldsymbol{n}_d \tag{28}$$

where $\boldsymbol{\omega}^{\mathcal{B}I}$ denotes the angular velocity of the satellite $\mathcal{B}$ frame with respect to the inertial I frame along the $\mathcal{B}$ frame. $\boldsymbol{\tau}_g$ is the gravity gradient torque expressed in the $\mathcal{B}$ frame, $\boldsymbol{b}_{\mathcal{B}}$ is the body-referenced Earth magnetic field and

$\boldsymbol{m}$ is the body-referenced magnetic control dipole generated by the magnetorquer. $\boldsymbol{n}_d$ is the perturbation torque, it is modeled as a Gaussian zero-mean noise process, it reflects the uncertainty in the satellite model, which is foremost due to residual magnetic torques and atmospheric drag. The quaternion kinematics model is considered as follows,

$$\dot{\boldsymbol{q}} = \frac{1}{2}\Omega\boldsymbol{q} \tag{29}$$

$$\Omega = \begin{pmatrix} 0 & \omega_z^{\mathcal{BO}} & -\omega_y^{\mathcal{BO}} & \omega_x^{\mathcal{BO}} \\ -\omega_z^{\mathcal{BO}} & 0 & \omega_x^{\mathcal{BO}} & \omega_y^{\mathcal{BO}} \\ \omega_y^{\mathcal{BO}} & -\omega_x^{\mathcal{BO}} & 0 & \omega_z^{\mathcal{BO}} \\ -\omega_x^{\mathcal{BO}} & -\omega_y^{\mathcal{BO}} & -\omega_z^{\mathcal{BO}} & 0 \end{pmatrix} \tag{30}$$

where $\boldsymbol{\omega}^{\mathcal{BO}}$ denotes the body-referenced angular velocity of $\mathcal{B}$ with respect to $\mathcal{O}$. Using rigid body assumptions and small angle approximations, the dynamics and the kinematics modeling equations lend themselves to a set of linear differential equations [12],

$$\dot{\mathbf{x}} = A\mathbf{x} + B(t)\,\mathbf{m} + G\,\mathbf{n}_d \tag{31}$$

And it is expressed as follows:,

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\omega}_x^{\mathcal{BO}} \\ \dot{\omega}_y^{\mathcal{BO}} \\ \dot{\omega}_z^{\mathcal{BO}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4\omega_{or}^2\sigma_1 & 0 & 0 & 0 & 0 & \omega_{or}\,(1-\sigma_1) \\ 0 & 3\omega_{or}^2\sigma_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{or}^2\sigma_3 & -\omega_{or}\,(1+\sigma_3) & 0 & 0 \end{pmatrix} \begin{pmatrix} \phi \\ \theta \\ \psi \\ \omega_x^{\mathcal{BO}} \\ \omega_y^{\mathcal{BO}} \\ \omega_z^{\mathcal{BO}} \end{pmatrix} +$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{b_z}{J_{11}} & -\frac{b_y}{J_{11}} \\ -\frac{b_z}{J_{22}} & 0 & \frac{b_x}{J_{22}} \\ \frac{b_y}{J_{33}} & -\frac{b_x}{J_{33}} & 0 \end{pmatrix} \begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{11}} & 0 & 0 \\ 0 & \frac{1}{J_{22}} & 0 \\ 0 & 0 & \frac{1}{J_{33}} \end{pmatrix} \boldsymbol{n}_d \tag{32}$$

$$\sigma_1 = \frac{J_{22} - J_{33}}{J_{11}} \tag{33}$$

$$\sigma_2 = \frac{J_{33} - J_{11}}{J_{22}} \tag{34}$$

$$\sigma_3 = \frac{J_{11} - J_{22}}{J_{33}} \tag{35}$$

The angles ϕ, θ and ψ are the roll, pitch, and yaw Euler angles, respectively. The magnetic control torques are proportional to the magnetic field components and to the on-board magnetic dipole components (with bounds of $1\,Am^2$). The magnetic field along the trajectory, the attitude, and the attitude rates are assumed known. Three types of modes are considered. The first mode is the "healthy" mode where no failures of the magnetometers nor disturbances in the magnetorquers occur. In the second mode, the magnetic rods are assumed to loose efficiency due to a loss of voltage. The logic of this mode consists in maintaining an attitude stabilization capability to some extent, while saving power. The third mode happens when the magnetometers fail to sense the magnetic field by loosing 90% of their output signal strength. These three modes, i.e., the healthy and the two failed modes, are assumed to occur at random according to the next transition probabilities,

$$P = \begin{pmatrix} 0.985 & 0.01 & 0.005 \\ 0.02 & 0.975 & 0.005 \\ 0.02 & 0.02 & 0.96 \end{pmatrix} \tag{36}$$

and their occurring is detected with same correct and wrong a priori probabilities as in Eq. (27). The weighing matrices, $Q(y)$ and $R(y)$, were tuned such as to fit with the model assumptions and to yield best possible performances. An example for associated parameters for the modes are displayed in Table 2. The dynamical state was initialized with, $\{20[deg]; -15[deg]; 15[deg]; 1\times 10^{-3}[\frac{rad}{sec}]; 5\times 10^{-4}[\frac{rad}{sec}]; -1.2\times 10^{-3}[\frac{rad}{sec}]\}$, and it was assumed that the initial mode was chosen at random in each run using the steady-state distribution $\{0.571,\ 0.317,\ 0.111\}$. Extensive Monte-Carlo simulations were run in order to illustrate the proposed regulator performances. Four cases were tested. Case 1 consists in applying the JLQ regulator of Section 1 when the mode is perfectly known. Case 2 consists in applying the proposed fault-tolerant JLQ regulator when the mode is measured with errors. In Case 3, the fault-tolerant JLQ algorithm is applied while the mode is perfectly measured, and in Case 4 the original JLQ is applied while the mode is imperfectly measured. The results are summarized in Figs. 4 and 5. The plots depict the time histories of the pointing errors for all four cases. Figure 4 shows that the applied JLQ magnetic attitude controllers succeed in achieving a nadir-pointing accuracy of the order of 1 degree in steady state for Cases 1, 2, and 3, and that cases 2 and 3 show relatively close performance. Figure 5 illustrates the fact that the original JLQ attitude controller can not handle imperfect mode measurements.

Table 2 Modes and Parameters

y_k	$B_k(y_k)$	$Q_k(y_k)$	$R_k(y_k)$
Ideal	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{b_z}{J_{11}} & -\frac{b_y}{J_{11}} \\ -\frac{b_z}{J_{22}} & 0 & \frac{b_x}{J_{22}} \\ \frac{b_y}{J_{33}} & -\frac{b_x}{J_{33}} & 0 \end{pmatrix}$	$\begin{pmatrix} 38 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 2.5 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 10 \end{pmatrix}$
Power saving	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{b_z}{J_{11}} & -\frac{b_y}{J_{11}} \\ -\frac{b_z}{J_{22}} & 0 & \frac{b_x}{J_{22}} \\ \frac{b_y}{J_{33}} & -\frac{b_x}{J_{33}} & 0 \end{pmatrix}$	$\begin{pmatrix} 38 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 300 & 0 & 0 \\ 0 & 800 & 0 \\ 0 & 0 & 400 \end{pmatrix}$
90% *decrease*	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{0.1b_z}{J_{11}} & -\frac{0.1b_y}{J_{11}} \\ -\frac{0.1b_z}{J_{22}} & 0 & \frac{0.1b_x}{J_{22}} \\ \frac{0.1b_y}{J_{33}} & -\frac{0.1b_x}{J_{33}} & 0 \end{pmatrix}$	$\begin{pmatrix} 38 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 4 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 10 \end{pmatrix}$

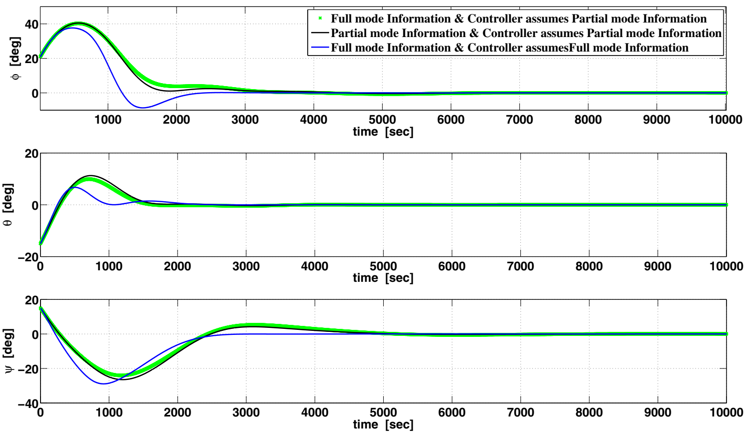


Fig. 4 Pointing errors for Cases 1 to 3. Monte-Carlo averages (100)

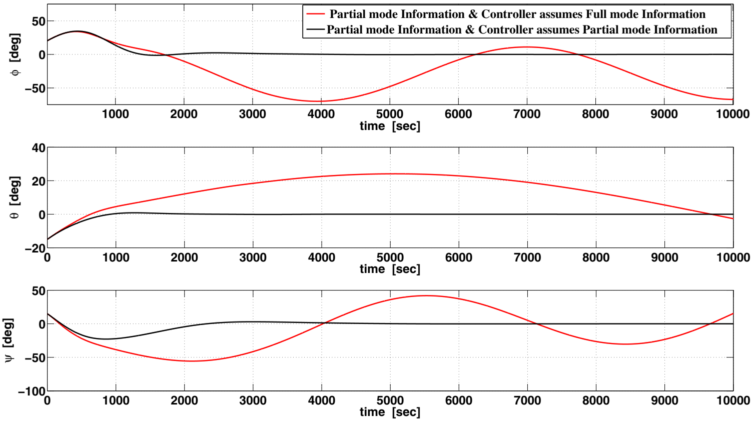


Fig. 5 Pointing errors in Cases 2 and 4. Monte-Carlo averages (100)

5 Conclusion

In this work, a novel suboptimal JLQ controller is suggested for discrete-time dynamical systems under the assumptions of full state information and a priori probabilities for correct and for wrong mode detections. Comparative results of an extensive Monte-Carlo simulation for a simple system illustrates the efficiency and conservativeness of the proposed algorithm that mitigates the destabilizing effect of corrupted mode observations while having the same computational burden as the full information JLQ controller. The fault-tolerant spacecraft attitude control results illustrate the validity of the approach, showing a steady state pointing accuracy of a few degrees, which is typical for magnetic based small satellites control performances. This general approach shows flexibility from the modeling standpoint and proves to be promising for the development of successful fault-tolerant attitude controllers.

Acknowledgements. This research was supported by The Israel Science Foundation (grant No. 1546/08).

Appendix

A Recursive Formulation of the Control Problem via Dynamic Programming:

Given the cost function,

$$J = E\left\{\sum_{k=0}^{N-1} \|x_{k+1}\|^2_{Q_k} + \|u_k\|^2_{R_k}\right\} \tag{37}$$

Consider the problem of minimizing J with respect to $\{\mathbf{u}_k\}$,

$$J_k^*\left(\mathcal{X}^k, \mathcal{Z}^k\right) \overset{\triangle}{=} \min_{U_k^{N-1}} E\left\{\sum_{i=k}^{N-1} \|x_{i+1}\|_{Q_i}^2 + \|u_i\|_{R_i}^2 \,|\mathcal{X}^k, \mathcal{Z}^k\right\} \tag{38}$$

where $\mathcal{X}^k \overset{\triangle}{=} \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and $\mathcal{Z}^k \overset{\triangle}{=} \{z_0, z_1, \ldots, z_k\}$ denote the histories of $\{\mathbf{x}_k\}$ and $\{\mathbf{z}_k\}$, respectively.

Recursive formulation of the control problem:

Start with step $N-1$:

$$\min_{u_{N-1}} J_{N-1}^* = \min_{u_{N-1}} E\left\{\|x_N\|_{Q_{N-1}}^2 + \|u_{N-1}\|_{R_{N-1}}^{\,2} \,|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\}$$

$$x_N = A_{N-1}x_{N-1} + B_{N-1}u_{N-1} + w_{N-1}$$

$$\min_{u_{N-1}} J_{N-1}^* = \min_{u_{N-1}} E\left\{\|A_{N-1}x_{N-1} + B_{N-1}u_{N-1} + w_{N-1}\|_{Q_{N-1}}^2\right.$$

$$\left. + \|u_{N-1}\|_{R_{N-1}}^{\,2} \,|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\}$$

$$= \min_{u_{N-1}} \{\|u_{N-1}\|_{E\left\{B_{N-1}^TQ_{N-1}B_{N-1}+R_{N-1}|\mathcal{X}^{N-1},\mathcal{Z}^{N-1}\right\}}^{\,2}$$

$$+ \|x_{N-1}\|_{E\left\{A_{N-1}^TQ_{N-1}A_{N-1}|\mathcal{X}^{N-1},\mathcal{Z}^{N-1}\right\}}^2$$

$$+ 2x_{N-1}^T E\left\{A_{N-1}^TQ_{N-1}B_{N-1}|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\} u_{N-1}$$

$$+ 2x_{N-1}^T \underbrace{E\left\{A_{N-1}^TQ_{N-1}w_{N-1}|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\}}_{=0}$$

$$+ 2u_{N-1}^T \underbrace{E\left\{B_{N-1}^TQ_{N-1}w_{N-1}|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\}}_{=0}$$

$$+ \underbrace{E\left\{w_{N-1}^TQ_{N-1}w_{N-1}|\mathcal{X}^{N-1}, \mathcal{Z}^{N-1}\right\}}_{\mathrm{Trace}\left\{W_{N-1}Q_{N-1}\right\}}\}$$

$$= \min_{u_{N-1}} \{ \ \left\|u_{N-1} + \left[\overline{B_{N-1}^TQ_{N-1}B_{N-1} + R_{N-1}}\right]^{-1}\right.$$

$$\left.\overline{B_{N-1}^TQ_{N-1}A_{N-1}}x_{N-1}\right\|_{\overline{B_{N-1}^TQ_{N-1}B_{N-1}+R_{N-1}}}^{2}\}$$

$$+ \|x_{N-1}\|_{\overline{A_{N-1}^TQ_{N-1}A_{N-1}} - \overline{A_{N-1}^TQ_{N-1}B_{N-1}}\left[\overline{B_{N-1}^TQ_{N-1}B_{N-1}+R_{N-1}}\right]^{-1}\overline{B_{N-1}^TQ_{N-1}A_{N-1}}}^2$$

$$+ \operatorname{tr}\{W_{N-1}Q_{N-1}\}$$

And for summary the solution for step $N-1$ is,

$$u_{N-1}^* = -\left[\overline{B_{N-1}^TQ_{N-1}B_{N-1} + R_{N-1}}\right]^{-1}\overline{B_{N-1}^TQ_{N-1}A_{N-1}}x_{N-1} = -\mathit{\Gamma}_{N-1}x_{N-1} \tag{39}$$

$$J_{N-1}^* = \|x_{N-1}\|_{\overline{A_{N-1}^TQ_{N-1}A_{N-1}} - \overline{A_{N-1}^TQ_{N-1}B_{N-1}}\mathit{\Gamma}_{N-1}}^2 + \operatorname{tr}\{W_{N-1}Q_{N-1}\} \tag{40}$$

$$J_{N-1}^* = \|x_{N-1}\|_{\tilde{S}_{N-1}}^2 + \operatorname{tr}\{W_{N-1}Q_{N-1}\} \tag{41}$$

Step $N-2$:

$$
\begin{aligned}
&J^*_{N-2}\left(\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right) \equiv \min_{U^{N-1}_{N-2}} E\left\{\sum_{i=N-2}^{N-1} \|x_{i+1}\|^2_{Q_i} + \|u_i\|^2_{R_i}\,|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right\}\\
&= \min_{U^{N-1}_{N-2}} E\left\{\|x_N\|^2_{Q_{N-1}} + \|u_{N-1}\|^2_{R_{N-1}} + \|x_{N-1}\|^2_{Q_{N-2}} + \|u_{N-2}\|^2_{R_{N-2}}\,|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right\}\\
&= \min_{u_{N-2}} E\{\|x_{N-1}\|^2_{Q_{N-2}} + \|u_{N-2}\|^2_{R_{N-2}}\\
&\qquad\qquad + \min_{u_{N-1}} E\{\|x_N\|^2_{Q_{N-1}} + \|u_{N-1}\|^2_{R_{N-1}}\,|\mathcal{X}^{N-1},\mathcal{Z}^{N-1}\}|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\}\\
&= \min_{u_{N-2}} E\left\{\|x_{N-1}\|^2_{Q_{N-2}} + \|u_{N-2}\|^2_{R_{N-2}} + J^*_{N-1}|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right\}\\
&= \min_{u_{N-2}} E\left\{\|x_{N-1}\|^2_{Q_{N-2}} + \|u_{N-2}\|^2_{R_{N-2}} + \|x_{N-1}\|^2_{\tilde{S}_{N-1}} + \operatorname{tr}\{W_{N-1}Q_{N-1}\}\,|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right\}\\
&= \min_{u_{N-2}} E\left\{\|x_{N-1}\|^2_{Q_{N-2}+\tilde{S}_{N-1}} + \|u_{N-2}\|^2_{R_{N-2}}\,|\mathcal{X}^{N-2},\mathcal{Z}^{N-2}\right\} + \operatorname{tr}\{W_{N-1}Q_{N-1}\}
\end{aligned}
$$

The solution for step $N-2$ is,

$$
\begin{aligned}
&u^*_{N-2} = -\left[\overline{B^T_{N-2}\left(Q_{N-2}+\tilde{S}_{N-1}\right)B_{N-2}+R_{N-2}}\right]^{-1}\\
&\overline{B^T_{N-2}\left(Q_{N-2}+\tilde{S}_{N-1}\right)}A_{N-2}x_{N-2}\\
&u^*_{N-2} = -\Gamma_{N-2}x_{N-2}\\
&J^*_{N-2} = \|x_{N-2}\|^2_{\overline{A^T_{N-2}\left(Q_{N-2}+\tilde{S}_{N-1}\right)A_{N-2}}-\overline{A^T_{N-2}\left(Q_{N-2}+\tilde{S}_{N-1}\right)B_{N-2}}\Gamma_{N-2}}\\
&\ + \operatorname{tr}\{W_{N-1}Q_{N-1}\} + \operatorname{tr}\left\{W_{N-2}\left(Q_{N-2}+\tilde{S}_{N-1}\right)\right\}\\
&\ = \|x_{N-1}\|^2_{\overline{A^T_{N-2}S_{N-2}A_{N-2}}-\overline{A^T_{N-2}S_{N-2}B_{N-2}}\Gamma_{N-2}} + \sum_{i=N-2}^{N-1}\operatorname{tr}\{W_iS_i\}
\end{aligned}
$$

It is straightforward to show that the solution for step k is,

$$
u^*_k = -\left[\overline{B^T_k\left(Q_k+\tilde{S}_{k+1}\right)B_k+R_k}\right]^{-1}\overline{B^T_k\left(Q_k+\tilde{S}_{k+1}\right)}A_kx_k = -\Gamma_kx_k \tag{42}
$$

$$
J^*_k = \|x_k\|^2_{\overline{A^T_kS_kA_k}-\overline{A^T_kS_kB_k}\Gamma_k} + \sum_{i=k}^{N-1}\operatorname{tr}\{W_iS_i\} = \|x_k\|^2_{\tilde{S}_k} + \sum_{i=k}^{N-1}\operatorname{tr}\{W_iS_i\} \tag{43}
$$

Where, for $k=N$ initialize the computation with

$$
\tilde{S}_N = 0 \tag{44}
$$

For $k = N-1, N-2, ..., 0$, compute

$$S_k\left(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}, y_k\right) = \tilde{S}_{k+1} + Q_k \tag{45}$$

$$\Gamma_k\left(\mathcal{X}^{k}, \mathcal{Z}^{k}\right) = \left(\overline{B_k^T S_k B_k + R_k}\right)^{-1} \overline{B_k^T S_k A_k} \tag{46}$$

$$\tilde{S}_k\left(\mathcal{X}^{k}, \mathcal{Z}^{k}\right) = \overline{A_k^T S_k A_k} - \overline{A_k^T S_k B_k} \Gamma_k \tag{47}$$

The variables $\overline{F}$ in Eqs. (46) and (47) are defined as follows:

$$\overline{F}(\mathcal{X}^k, \mathcal{Z}^k) \overset{\triangle}{=} E\{F \mid \mathcal{X}^k, \mathcal{Z}^k\} \tag{48}$$

B Conditional Expectation Approximations

In the case examined in this work there is a need to calculate a conditional expectation of the next form,

$$\overline{F}(\mathcal{X}^k, \mathcal{Z}^k) \overset{\triangle}{=} E\{F \mid \mathcal{X}^k, \mathcal{Z}^k\} \tag{49}$$

Considering the conditioning sequence in Eq. (49) the computation of these conditional expectations would require a growing memory size. Two approximate solutions are examined, the first approximation is the case where only the mode observation histories $\mathcal{Z}^k$ are retained, i.e., by computing the following conditional expectation

$$E\left\{F|\mathcal{X}^k, \mathcal{Z}^k\right\} \approx \overline{\overline{F}}(\mathcal{Z}^k) = E\{F \mid \mathcal{Z}^k\} = E\left\{F\left(y_k, \mathcal{Z}^k, z_{k+1}\right)|\mathcal{Z}^k\right\} \tag{50}$$

$$\overline{\overline{F}}(\mathcal{Z}^k) = \sum_{z_{k+1}\in S_z} \sum_{y_k \in S_y} F\left(y_k, \mathcal{Z}^k, z_{k+1}\right) \Pr\left(y_k, z_{k+1}|\mathcal{Z}^k\right) \tag{51}$$

In order to find the expression for the conditional probability $\Pr\left(y_k, z_{k+1}|\mathcal{Z}^k\right)$ we will first examine the analytic solution from step k=0,

$$\Pr\left(y_0, z_1|\mathcal{Z}^0\right) = \frac{\Pr\left(y_0, z_1, \mathcal{Z}^0\right)}{\Pr\left(\mathcal{Z}^0\right)} = \frac{\Pr\left(y_0, z_1, z_0\right)}{\Pr\left(z_0\right)} \tag{52}$$

$$\Pr\left(y_0, z_1|\mathcal{Z}^0\right) = \frac{\Pr\left(z_1|y_0, z_0\right)\Pr\left(z_0|y_0\right)\Pr\left(y_0\right)}{\Pr\left(z_0\right)} \tag{53}$$

$$\Pr\left(y_0, z_1|\mathcal{Z}^0\right) = \frac{\sum_{y_1}^{\nu}\left\{\Pr\left(z_1|y_1\right)\Pr\left(y_1|y_0\right)\right\}\Pr\left(z_0|y_0\right)\Pr\left(y_0\right)}{\Pr\left(z_0\right)} \tag{54}$$

For step $k = 1$,

$$Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\Pr\left(y_1, z_2, \mathcal{Z}^1\right)}{\Pr\left(\mathcal{Z}^1\right)} = \frac{\Pr\left(y_1, z_0, z_1, z_2\right)}{\Pr\left(z_0, z_1\right)} \tag{55}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\Pr\left(z_2|y_1, z_0, z_1\right) \sum_{y_0}^{\nu} \Pr\left(y_0, y_1, z_0, z_1\right) \Pr\left(y_0\right)}{\Pr\left(z_0, z_1\right)} \tag{56}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\Pr\left(z_2|y_1, z_0, z_1\right) \sum_{y_0}^{\nu} \left\{\Pr\left(y_1|y_0, z_0, z_1\right) \Pr\left(y_0\right) \Pr\left(y_0, z_1, z_0\right)\right\}}{\Pr\left(z_0, z_1\right)} \tag{57}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\Pr\left(z_2|y_1, z_0, z_1\right) \sum_{y_0}^{\nu} \left\{\Pr\left(y_1|y_0\right) \Pr\left(y_0\right) \Pr\left(y_0, z_1|Z^0\right) \Pr\left(z_0\right)\right\}}{\sum_{y_0}^{\nu} \Pr\left(y_0, z_0, z_1\right) \Pr\left(y_0\right)} \tag{58}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\Pr\left(z_2|y_1, z_0, z_1\right) \sum_{y_0}^{\nu} \left\{\Pr\left(y_1|y_0\right) \Pr\left(y_0\right) \Pr\left(y_0, z_1|z_0\right) \Pr\left(z_0\right)\right\}}{\sum_{y_0}^{\nu} \left\{\Pr\left(y_0, z_1|\mathcal{Z}^0\right) \Pr\left(z_0\right) \Pr\left(y_0\right)\right\}} \tag{59}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\sum_{y_2}^{\nu} \left\{\Pr\left(z_2|y_2\right) \Pr\left(y_2|y_1\right)\right\} \sum_{y_0}^{\nu} \left\{\Pr\left(y_1|y_0\right) \Pr\left(y_0\right) \Pr\left(y_0, z_1|z_0\right) \Pr\left(z_0\right)\right\}}{\sum_{y_0}^{\nu} \left\{\Pr\left(y_0, z_1|\mathcal{Z}^0\right) \Pr\left(z_0\right) \Pr\left(y_0\right)\right\}} \tag{60}$$

$$\Pr\left(y_1, z_2|\mathcal{Z}^1\right) = \frac{\sum_{y_2}^{\nu} \left\{\Pr\left(z_2|y_2\right) \Pr\left(y_2|y_1\right)\right\} \sum_{y_0}^{\nu} \left\{\Pr\left(y_1|y_0\right) \Pr\left(y_0\right) \Pr\left(y_0, z_1|z_0\right)\right\}}{\sum_{y_0}^{\nu} \left\{\Pr\left(y_0, z_1|z_0\right) \Pr\left(y_0\right)\right\}} \tag{61}$$

And for step $k = 2$,

$$\Pr\left(y_2, z_3|\mathcal{Z}^2\right) = \frac{\Pr\left(y_2, z_3, \mathcal{Z}^2\right)}{\Pr\left(\mathcal{Z}^2\right)} = \frac{\Pr\left(y_2, z_3, z_2, z_1, z_0\right)}{\Pr\left(z_2, \mathcal{Z}^1\right)} \tag{62}$$

$$\Pr\left(y_2, z_3|\mathcal{Z}^2\right) = \frac{\Pr\left(z_3|y_2, \mathcal{Z}^2\right) \sum_{y_1}^{\nu} \left\{\Pr\left(y_2, y_1, z_2, \mathcal{Z}^1\right) \Pr\left(y_1\right)\right\}}{\Pr\left(z_2, Z^1\right)} \tag{63}$$

$$\Pr\left(y_2, z_3|\mathcal{Z}^2\right) = \frac{\sum_{y_3}^{\nu} \left\{\Pr\left(z_3|y_3\right) \Pr\left(y_3|y_2\right)\right\} \sum_{y_1}^{\nu} \left\{\Pr\left(y_2|y_1\right) \Pr\left(y_1\right) \Pr\left(y_1, z_2|\mathcal{Z}^1\right) \Pr\left(\mathcal{Z}^1\right)\right\}}{\sum_{y_1}^{\nu} \left\{\Pr\left(y_1, z_2|\mathcal{Z}^1\right) \Pr\left(\mathcal{Z}^1\right) \Pr\left(y_1\right)\right\}} \tag{64}$$

$$\Pr\left(y_2, z_3|\mathcal{Z}^2\right) = \frac{\sum_{y_3}^{\nu} \left\{\Pr\left(z_3|y_3\right) \Pr\left(y_3|y_2\right)\right\} \sum_{y_1}^{\nu} \left\{\Pr\left(y_2|y_1\right) \Pr\left(y_1\right) \Pr\left(y_1, z_2|\mathcal{Z}^1\right)\right\}}{\sum_{y_1}^{\nu} \left\{\Pr\left(y_1, z_2|\mathcal{Z}^1\right) \Pr\left(y_1\right)\right\}} \tag{65}$$

It is straightforward to show that the solution for step $k > 0$ is,

$$\Pr\left(y_k, z_{k+1}|\mathcal{Z}^k\right) = \\ = \frac{\sum_{y_{k+1}}^{\nu} \left\{\Pr\left(z_{k+1}|y_{k+1}\right) \Pr\left(y_{k+1}|y_k\right)\right\} \sum_{y_{k-1}}^{\nu} \left\{\Pr\left(y_k|y_{k-1}\right) \Pr\left(y_{k-1}, z_k|\mathcal{Z}^{k-1}\right) \Pr\left(y_{k-1}\right)\right\}}{\sum_{y_{k-1}}^{\nu} \left\{\Pr\left(y_{k-1}, z_k|\mathcal{Z}^{k-1}\right) \Pr\left(y_{k-1}\right)\right\}} \tag{66}$$

The computations of $\widetilde{\overline{F}}(\mathcal{Z}^k)$ are performed as follows:

$$\widetilde{\overline{F}}(\mathcal{Z}^k) = \sum_{z_{k+1} \in S_z} \sum_{y_k \in S_y} F\left(y_k, \mathcal{Z}^k, z_{k+1}\right) \times \frac{\sum_{y_{k+1}}^{\nu} \{\Pr(z_{k+1}|y_{k+1}) \Pr(y_{k+1}|y_k)\} \sum_{y_{k-1}}^{\nu} \left\{\Pr(y_k|y_{k-1}) \Pr\left(y_{k-1}, z_k|\mathcal{Z}^{k-1}\right) \Pr(y_{k-1})\right\}}{\sum_{y_{k-1}}^{\nu} \{\Pr(y_{k-1}, z_k|\mathcal{Z}^{k-1}) \Pr(y_{k-1})\}} \tag{67}$$

The second suggested approximate solution is achieved by retaining only the current mode observation z_k, i.e., by computing the following conditional expectation:

$$E\left\{F|\mathcal{X}^k, \mathcal{Z}^k\right\} \approx \overline{\overline{F}}(z_k) = E\{F \mid z_k\} = E\left\{F(y_k, z_{k+1})|z_k\right\} \tag{68}$$

$$\overline{\overline{F}}(z_k) = \sum_{z_{k+1} \in S_z} \sum_{y_k \in S_y} F(y_k, z_{k+1}) \Pr(y_k, z_{k+1}|z_k) \tag{69}$$

The computations of $\Pr(y_k, z_{k+1}|z_k)$ are performed as follows:

$$\Pr(y_k, z_{k+1}|z_k) = \frac{\Pr(y_k, z_{k+1}, z_k)}{\Pr(z_k)} = \frac{\Pr(z_{k+1}|y_k, z_k) \Pr(z_k|y_k) \Pr(y_k)}{\Pr(z_k)} \tag{70}$$

$$\Pr(y_k, z_{k+1}|z_k) = \frac{\sum_{y_{k+1}}^{\nu} \{\Pr(z_{k+1}|y_{k+1}) \Pr(y_{k+1}|y_k)\} \Pr(z_k|y_k) \Pr(y_k)}{\Pr(z_k)} \tag{71}$$

The computations of $\overline{\overline{F}}(z_k)$ are performed as follows:

$$\overline{\overline{F}}(z_k) = \sum_{z_{k+1} \in S_z} \sum_{y_k \in S_y} F(y_k, z_{k+1}) \times \frac{\Pr(z_k|y_k) \Pr(y_k) \sum_{y_{k+1} \in S_y} \{\Pr(z_{k+1}|y_{k+1}) \Pr(y_{k+1}|y_k)\}}{\sum_{y_k \in S_y} \{\Pr(z_k|y_k) \Pr(y_k)\}} \tag{72}$$

References

1. Athans, M., Castanon, D., Dunn, K., Greene, C.S., Lee, W.H., Sandell Jr., N.L., Willsky, A.S.: The Stochastic Control of F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method, Part I: Equilibrium Flight. IEEE Trans. Automat. Contr. AC 22(5) (October 1977)
2. Rishel, R.W.: Dynamic Programming and Minimum Principles for Systems with Jump Markov Disturbances. SIAM J. on Control 9, 338–371 (1975)
3. Sworder, D.D.: Feedback Control of a Class of Linear Systems with Jump Parameters. IEEE Trans. Automat. Contr. AC 14(1), 9–14 (1969)

4. Mariton, M.: On the Influence of Noise on Jump Linear Systems. IEEE Trans. Automat. Contr. AC 32(12), 1094–1097 (1987)
5. Pierce, B.D., Sworder, D.D.: Bayes and Minimax Controllers for a Linear System with Stochastic Jump Parameters. IEEE Trans. Automat. Contr. AC 16, 300–307 (1971)
6. Mariton, M.: Jump Linear Quadratic Control with Random State Discontinuities. Automatica 23(2), 237–240 (1987)
7. Wonham, W.M.: Random Differential Equations in Control Theory. In: Bharucha-Reid, A. (ed.) Probabilistic Methods in Applied Mathematics, vol. 2, pp. 192–194. Academic Press (1970)
8. Chizeck, H.J., Willsky, A.S., Castanon, D.: Discrete-time Markovian-jump Linear Quadratic Optimal Control. Int. J. Contr. 43(1), 213–231 (1986)
9. Ackerson, G.A., Fu, K.S.: On State Estimation in Switching Environments. IEEE Trans. Automat. Contr. AC 15(1), 10–17 (1970)
10. Campo, L.J., Bar-Shalom, Y.: Control of Discrete-Time Hybrid Stochastic Systems. IEEE Trans. Automat. Contr. AC 37(10), 1522–1527 (1992)
11. Choukroun, D., Speyer, J.L.: Mode-Estimator-Free Quadratic Control of Jump-Linear Systems with Mode-Detection Random Delay. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference 2005, San Francisco, CA, pp. 1–20 (August 2005)
12. Psiaki, M.L.: Magnetic Torquer Attitude Control via Asymptotic Periodic Linear Quadratic Regulation. Journal of Guidance, Control and Dynamics 24(2), 386–394 (2001)
13. Costa, O.L.V., Fragoso, M.D., Marques, R.P.: Discrete-Time Markov Jump Linear Systems. Springer (2005)

GNC Challenges and Navigation Solutions for Active Debris Removal Mission

Erwan Kervendal, Thomas Chabot, and Keyvan Kanani

Abstract. Active removal of large space debris has been identified as a key mission to limit growth of debris jeopardizing missions of active satellites. In particular, orbits of economic and strategic importance, Low Earth Orbits, are pervaded with objects such as upper stages of launchers or defunct satellites: collision between large debris has become a likely event in the next five years according to simulations done in Space Agencies. Willing to anticipate such event and limit collision risk, Agencies and industrials investigate feasibility of Active Debris Removal (ADR) mission. Many critical points have yet to be solved, such as legal aspects, cost, debris to be removed and technological challenges to successfully complete the mission. This paper will first initiate a discussion around challenges that has to face the Guidance, Navigation and Control (GNC) sub-system during the ADR mission. Then, two navigation solutions that meet most of navigation challenges for ADR mission will be introduced in this paper. The first solution relies on an active, 3D camera, fused with IMU data in a navigation filter. The second solution relies on a passive, 2D camera and a state-of-the-art Image Processing that provides pseudo-measurements, also fused with IMU data in the navigation filter.

1 Active Debris Removal: Main Challenges

For the past forty years, space debris have been identified as a growing risk for present and future space missions, especially for Low Earth Orbits (LEO). As suggested by Kessler [1], there would even be a critical number of debris for which risk of cascading effect due to collisions in LEO would be inevitable. In 2009, two artificial satellites – Iridium 33 and defunct Kosmos-2251- actually

Erwan Kervendal · Thomas Chabot · Keyvan Kanani
Astrium Satellites, Toulouse, France
e-mail: {erwan.kervendal,thomas.chabot,
keyvan.kanani}@astrium.eads.net

collided at 789 kilometres above Siberia [2] and therefore created clouds of debris on LEO, illustrating not only the risk generated by defunct satellites on orbits but also the detrimental, cascading effect of debris clouds. Recent studies of Space Agencies such as European Space Agency (ESA) [3], Centre national d'Etudes Spatiales (CNES for French Space Agency) [4], Deutsches Zentrum für Luft und Raumfahrt (DLR for German Space Agency) [5], National Aeronautics and Space Administration (NASA) [6] or Japan Aerospace Exploration Agency (JAXA) [7] identified removal of large debris as one of the solutions to limit growing numbers of hazardous objects in LEO, orbits of economic and strategic importance. Interest of Space agencies and consequently space industries has been significantly growing in Europe for the last years, and several feasibility studies of Active Debris Removal (ARD) mission have been investigated.

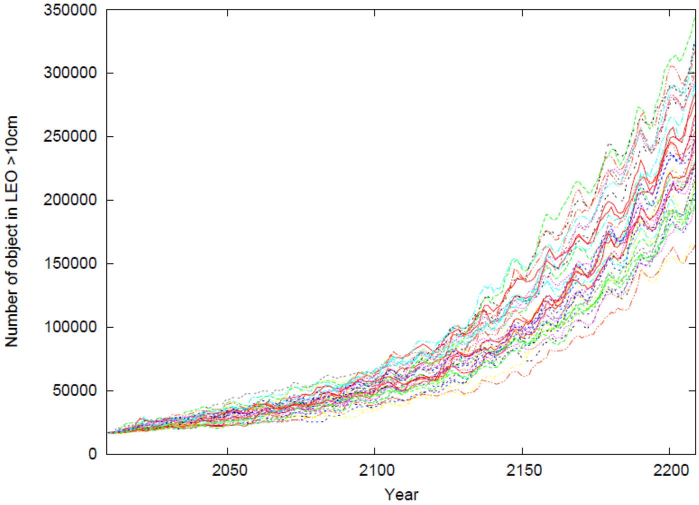


Fig. 1 Projections of debris environment in Low Earth Orbits, considering objects larger than 10 cm, from 2009 to 2209, [3]

In light of numerous papers throughout the world, there are yet many challenges to be overcome from a technological perspective (e.g. ways to approach debris and capture them, mission design to capture one or several targets, selection of targets), from a legal perspective (e.g. property transfer of debris owner to debris remover, insurance, risk transfer), and financial perspective (e.g. business case, cost of mission). For the sake of completeness, a short list of technological issues to be solved is populated below:

- Mission design: numbers of debris, populations of debris to be removed, initial orbit of chaser/launcher, duration of mission, support from ground, level of autonomy, strategy for relative navigation and approach, target identification, capture and de-orbit phases

- Propulsion: required ΔV and thrust level as function of mission design, compatibility with available launchers and targeted set of orbits, available propulsion (electrical vs. chemical)
- Navigation sensors: sensors to be used for relative navigation w.r.t. target during different phases of the mission, able to provide measurements for target identification, estimation of rotating rates and estimation of relative position and velocity, within required accuracy, prior to capture.
- Fault Detection, Isolation and Recovery strategies and safe mode: The global FDIR of an ADR mission with one or several un-cooperative targets has never been investigated yet while close navigation, capture or docking inherently imply risk of collision with or without failure of onboard sensors or actuators. De-orbiting strategies consisting in bringing the debris in low altitude or graveyard orbit also bring up issue on FDIR. Safe mode has to be deeply investigated, in particular during terminal rendezvous (just prior to capture) or de-orbiting phase. Indeed, a typical safe mode relies on Sun –pointing strategy which could be not compatible with capture requirements or de-orbiting guidance.
- Capture devices: harpoon, net, claws, arm have been discussed in many papers and demonstrated on ground or in space. However, no firm baseline has yet been selected and demonstrated on actual debris removal mission.
- De-orbiting strategies and devices: drag sail, de-orbit sail, propulsive packs and active de-orbitation by the chaser are considered.

Many technological issues are indeed related to the Guidance, Navigation and Control (GNC) sub-system as it lies in the heart of the critical phases of the ADR mission. This paper will henceforth focus on the GNC challenges to be faced during the different phases.

2 Challenges of GNC for ADR mission

The design of the GNC sub-system for an ADR mission is a complicated and challenging task as the GNC system shall be adaptable to many environmental conditions (Sun elevation, eclipse, Earth, Moon in background), many targets, and a large span of relative distances between chaser and target while being reliable, autonomous to some extent and CPU efficient. The following discussion will be articulated around the different phases of the ADR mission, presented in the following paragraphs.

2.1 Typical Phases of an ADR Mission

By analogy with rendezvous mission, ADR mission can be divided into five distinct phases for each target: phasing, approach, fly around/inspection, capture and de-orbitation phase.

Phasing

The phasing phase typically consists of estimating orbital parameters of the target's orbit and then "aligning" plane of chaser orbit w.r.t. that of target. Inclination, Right Ascension of Ascending Node (RAAN) and argument of periapsis are main parameters to be corrected in this sequence. Unlike typical rendezvous in LEO, phasing and subsequent phases can be rather challenging as target's orbit can have significant eccentricity.

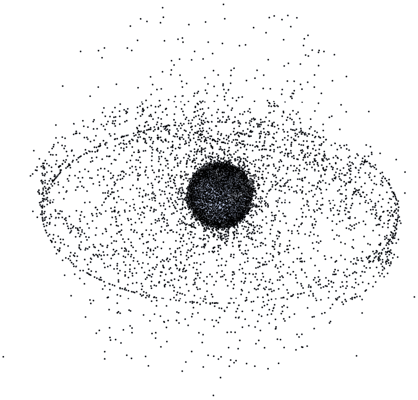


Fig. 2 Approximately 19,000 manmade objects are larger than 10 centimetres in Earth orbit as of July 2009, most orbit close to the Earth, credit NASA Earth Observatory

As far as removal of debris on Earth orbits is concerned, there are no specific needs in autonomy from the chaser during the phasing sequence. Navigation can easily be done with ground in the loop. For instance, the chaser orbit can be estimated through GPS data while the debris orbit can be estimated from ground observations. Several networks are capable of detecting debris as small as 10 cm on LEO [8] implying radar or optical observations. Several studies and projects for improving performances of radar system dedicated to debris removal missions are being conducted [9], [10]. In such configuration, a conservative figure for the accuracy of estimation of relative position between the chaser and the debris is around 400 m (3σ), based on the 300 m (3σ) accuracy of US Air Force published performances [11] and a typical 10 m (3σ) accuracy for chaser position estimation with GPS on LEO, along with conservative margin. The phasing phase, with ground in the loop, should end at about 1 to 2 km from target.

Approach

The approach sequence is initiated at about 2 km from target. Relative navigation takes over ground-based navigation as the chaser needs to get closer and closer to the target (up to 10 meters) through dedicated manoeuvres. Owing to smaller and smaller distance between the chaser and the target, collision risks are growing and

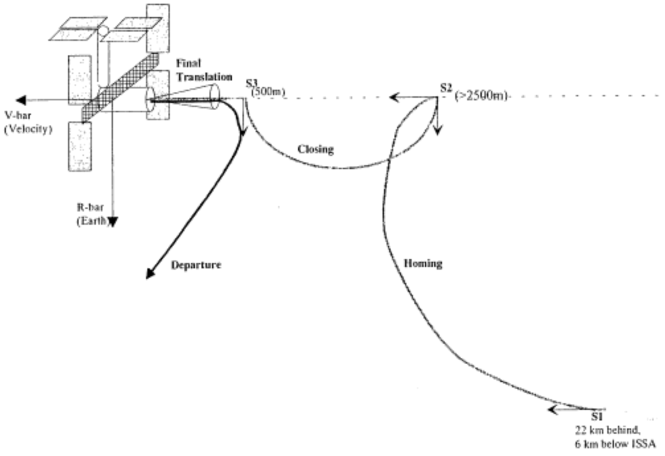


Fig. 3 Example of Rendezvous manoeuvres with objects in LEO: ATV trajectory from phasing to docking with the International Space Station

some autonomy is required, in particular to trigger anti-collision manoeuvres if needed. By analogy with typical rendezvous manoeuvres, V-bar and R-bar "hops" can be performed and perturbations due to atmospheric drag (for LEO), gravity gradient, magnetic torque or solar pressure (for higher orbit) acting on both chaser and target are corrected along manoeuvres.

Fly around/Inspection

Unlike typical rendezvous on LEO, missions for debris removal inherently imply un-cooperative targets. Besides, typical debris on LEO would be flying for several years; they should hence be poorly known (uncertainties on dimensions, mass, inertia…) because of aging effect, collisions with smaller debris, etc. A better "understanding" of the debris is needed. Depending on capture devices, rotation rates, rotation axes, possibly mass and inertia should be estimated. A spot to "grasp" the debris might be designated which inherently implies a mapping of the debris as well. To that end, inspection sequence would consist in stable orbit around the target (e.g. football orbit) or a station keeping at safe distance from target if the target is tumbling enough to allow for complete observations. On the inspection orbit, onboard sensors of chaser can take as many 2D or 3D pictures as needed to estimate necessary parameters. Attitude of the chaser is controlled for optimal inspection operations. Acquired data can either be processed onboard the spacecraft for debris estimation or sent to ground, provided the relative orbit is stable enough during the inspection sequence.

After completion of debris estimation, the capture can be triggered.

Capture

The capture is triggered as soon as necessary data have been estimated by the chaser. Operations and manoeuvres to be performed during the capture highly depend on the selected capture devices. Many capture technologies are being investigated, some have been demonstrated on ground mostly.

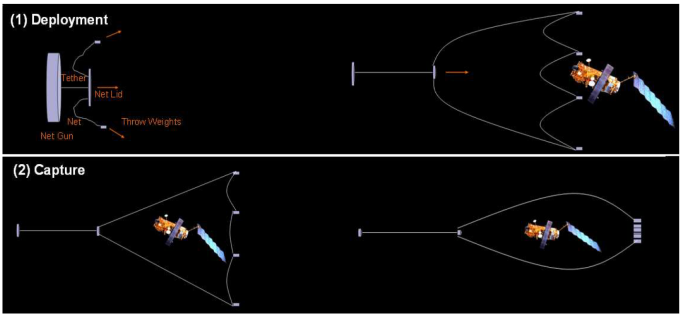


Fig. 4 Deployment and capture sequence using a net

Net has for instance been developed and demonstrated in 0-g flight [12] and can be launched from about 5 meters from target. Provided net diameter is much larger than debris, such a solution should be impacted by performances of attitude control.

Harpoon was also demonstrated in laboratory [13]. It needs to be fired on a specific location of the debris, several meters away from debris.

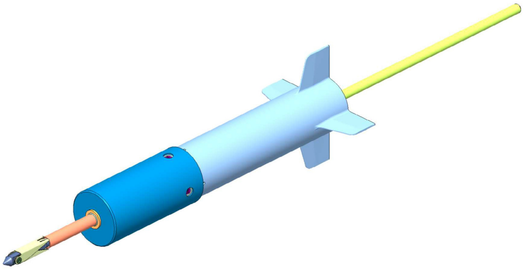


Fig. 5 sketch of harpoon for terrestrial demonstration, made up of conical tip to avoid debris generation and crushable cartridge to limit depth of penetration within debris.

Many other devices such as articulated arm [14] need the chaser to get close to the target (a couple of meters) and grasp the debris on a specific location. If the debris rotation rate is too high (above 1 deg/s), the chaser might need to align with the target's main rotation axis and eventually spin with the same rotation rate. Such strategy would eventually allow to lower the braking torque in the arm at capture. The concept of arm as capture device was demonstrated on Orbital

Express. Claws can also be used as capturing device. Other solutions such as Ion Beam [15] [16] or electrostatic tractor force [17] are approaches that do not need to grasp the debris, only forcing the debris at lower altitude for de-orbitation.

Once the debris is captured, the de-orbit operations can be initiated.

De-orbitation

Once the captured is confirmed, the very first step is to stabilize the captured debris, through control of the chaser-debris system by the chaser's Attitude and Orbit Control System (AOCS). Then, the debris is towed by the chaser either on graveyard orbit or to a lower orbit for atmospheric re-entry and destruction. This can be also done via dedicated "de-orbiting" packs that are separated from the chaser once the debris is captured [18]. The de-orbiting pack, linked to the debris and able to thrust, will then tow the debris while the chaser will continue its journey toward another target. Other device such as electrostatic tether [19] or drag augmenting device [20] can be fixed to the debris and will lower the debris orbit till re-entry.

2.2 GNC Challenges and Possible Solutions

As depicted above, an ADR mission is divided into several phases. From a GNC perspective, there are stringent requirements to be met in order to allow for successful completion of mission as detailed above. Since a definitive concept is not known yet, there are no specific figures for requirements. Several GNC technologies still need to be developed, demonstrated and validated for ADR mission in order to face the following challenges:

- Robust guidance during approach and then de-orbiting phase, with un-cooperative target
- Robust control during approach, capture (stabilization of composite chaser-debris) and de-orbiting
- Identification of critical parameters of the un-cooperative and poorly known target
- Online estimation of relative position, velocity and attitude prior to capture

Robust and Autonomous Guidance

As previously stated, phasing sequence should mainly rely on ground for the navigation and trajectory computation: distance to target and duration of operations are large enough w.r.t. relative dynamics of chaser and debris. Then, as the chaser is getting very close to the target, some autonomy could be needed. Relative distance and implied dynamics might be such that time needed to get a command forth and back from ground would be too large, the chaser should hence be able to autonomously elaborate withdrawal strategies if needed. Careful design of capture sequence and trajectory along with dedicated monitoring system (based on GEO satellite relay for instance) may alleviate need in autonomy. In any case, guidance and control should be able to meet safety requirements which, to some extent, call for robustness to environment uncertainties, robustness to system and sub-system uncertainties and most critically, robustness to debris uncertainty.

During the approach and inspection phases, ground should monitor operations, relative trajectory and possibly correct manoeuvres if needed. Guidance of the chaser should compute in line the best trajectory to reach close vicinity of debris under minimal ΔV budget (especially if several debris removals are considered) and with maximal safety. Typical V-bar and R-bar hops, along with Station Keeping points - allowing ground to monitor and correct operations -, should be the main features of approach's trajectory. V-bar manoeuvres are known to be more ΔV efficient but can lead to collision with debris if there is some thrusters' failure for instance as the manoeuvres consists in a drift in the direction of V-bar axis. R-bar manoeuvres are more ΔV demanding but less safety critical as the motion along V-bar is bounded. In particular, if no ΔV correction is done after a R-bar hop, the chaser will stay on a football orbit (neglecting perturbations), at a relatively stable distance from the debris which is not only safe but also favourable to debris observation if needed. As a consequence, typical manoeuvres plan of such mission should result from a combination of both manoeuvres. Recent advances in domain of guidance for space rendezvous should also benefit to ADR mission for optimal manoeuvre plan computation and correction under uncertainties. For instance, guidance algorithms relying on direct, indirect or analytical methods have shown high performances for typical rendezvous and formation-flying missions (prisma, Simbol-X, ATV) [21]. In particular, such methods can allow to optimize manoeuvres as function of uncertainties on relative navigation, thrust realizations and environment perturbations. They should also compute optimal trajectories to reach target on an eccentric orbit. The next step should therefore to validate CPU efficiency of such algorithms.

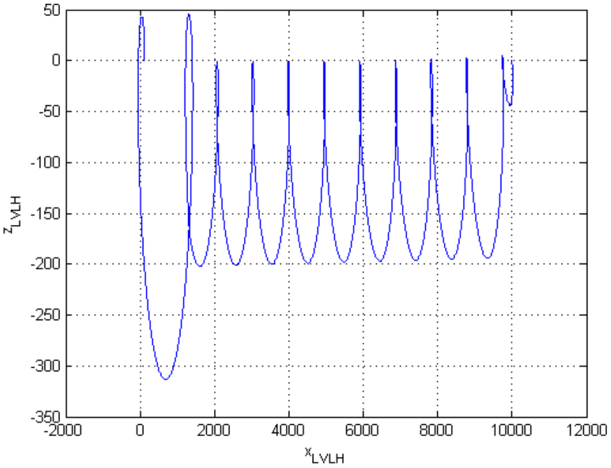


Fig. 6 . Rendezvous performed using analytical guidance algorithm on quasi circular orbit (Prisma). The rendezvous duration is 64620 s (10 orbits) The error on the final state for this rendezvous is less than 1 cm in position, and around 1 mm/s in velocity.

After successful capture and tranquilization of the new composite (discussed in paragraph dedicated to control), a major challenge for the autonomous guidance is to compute manoeuvre plan in order to "tow" the debris from initial orbit to de-orbitation state. This task is all the more challenging since debris could be initially poorly known. As a consequence, significant uncertainty on debris mass, inertia, drag coefficient or impact on thrust realization, is to be considered in autonomously computed trajectory while ensuring no collision. Composite behaviour is currently the most critical tasks as no mathematical model is widely accepted by space industries, nor representative demonstration was yet flown. In particular, the composite behaviour with non rigid link (a cable is used to track the debris if capture by a net, a harpoon or claws) is of most importance. Guidance should therefore take into account perturbations due to towed debris and avoid collision. Sensitivity of guidance algorithms to debris uncertainty and guidance strategies taking into account composite behaviour should be investigated in order to ensure efficient and safe de-orbitation.

Should the de-orbit strategy rely on ion beam or electrostatic tractor force, optimal conditions to "push" or "tract" the debris should also be considered in guidance. Behaviour of such composite (under electrostatic force for instance) should therefore be widely investigated and understood to compute efficient, robust and autonomous guidance.

Robust and Adaptive Control

Alike the guidance, ADR mission will significantly challenge design of control function. Although there is large, world-wide skills to design efficient control loop with high performances for satellites in LEO or GEO, control for tranquilization of debris at capture and efficient towing (depending on capture solution) is still to be investigated.

During phasing and approach, AOCS should be very similar to that of a typical, large satellite. Such designs have been well validated for many years. Attitude control may be wheel-based or thruster-based. Control of relative position and velocity could be done with main engines oriented or not by attitude control. However, capture and towing sequences bring up unusual issues, depending on selected capture devices and de-orbiting strategies.

During the capture, the capture device could be ejected from chaser. Depending on the location of the capture device, it will inevitably create a disturbance torque to be damped by control function as soon as possible in order to resume operations. Right after ejection, the chaser control might have to "tranquilize" the new composite (debris linked to the chaser).

If the link is rigid (e.g. arm) part of the structure might be designed to damp part of the disturbance torque. The remaining of parasite torque should then be controlled and damped by the chaser.

Provided the link is indeed not rigid (a cable for instance); specific strategies to tranquilize the debris have to be investigated and control is to be designed accordingly. Indeed, de-orbit ΔV can be transferred to motion of the debris if not "tranquilized". It could yield to a change in ΔV direction of the whole composite because of debris "free" motion. To tranquilize the composite, the cable should first not be wrapped around the debris as it rotates: a constant force should be applied on the cable. De-spinning force could then be transferred from chaser to debris until tranquilisation is reached, i.e. when parasite torques and forces are within requirements.

However, the tranquilisation sequence might imply the control to be robust enough to large change in mass and inertia between chaser and composite (before and after the capture) as well as flexibility. As a matter of fact, the control function is initially designed to cope up with requirements of phasing, approach, inspection phases and stabilization for capture. Similarly to typical controller tuning for satellites, tuning of control is partly function of accuracy requirements and expected margin for a given set of mass and inertia. With the coupling to the debris, change in mass and inertia is probably much larger than considered uncertainty and the controller might have very degraded performances or even be unstable. A scenario could be considered to face such a challenge.

First, the initial controller is switched to another controller right after the capture. It could yield to a quite unstable situation due to controller transient created by the switch in controller, worsen by torque inherited from debris rotation rates. Then, the new controller – designed to be much more robust to mass and inertia uncertainty - could eventually tranquilize the composite. Nonetheless, due to the large margins to be considered, expected performances should be degraded. As a consequence, right after the tranquilisation is completed, another switch to higher performance controller should be done as de-orbit manoeuvres might need fine pointing accuracy. However, the third controller should consider mass and inertia of the composite in order to meet mission requirements. By definition, the debris is unknown and there are therefore two possibilities to compute the control of such new composite. The first possibility is to consider that the debris is roughly known and that mass and inertia can be estimated a priori. Uncertainty on mass and inertia to be considered for control design should however be large enough to be robust to actual mass and inertia of debris. In all likelihood, performances would not be optimal. The second possibility is to estimate in line mass and inertia of composite through dedicated manoeuvres. Such operations should be complex as it should involve dedicated manoeuvres and ground operations.

To conclude, there are possibly several approaches to meet ADR mission requirements and it seems the control should be as robust as possible to prompt change in mass and inertia, capable of damping sporadic high torques due to capture firing and tranquilisation of debris and capable of fine pointing performances. A possible solution should be to rely on different set of controllers with hard switch. Recently, recent breakthroughs on Linear Parameter Varying design have increased possibility to used adaptive and modern control within space industry [22]. Such solutions would allow to design an unique controller, capable to adapt to different AOCS mode and inherent requirements. Such solutions would be worth investigating further in the frame of the control design of ADR mission.

Identification of Critical Parameters of Debris

By definition, debris are poorly known. Their mass, inertia, dimensions, center of gravity, rotations axes and rotation rates are needed for guidance, control and proper capture as discussed previously.

Some debris such as defunct satellites should be roughly known by the owner and to some extent, a fair initial guess should be available to the chaser. Some other debris such as collided satellite for instance might be rather un-known by the chaser. In both cases however, an in-line identification of parameters should be needed.

Identification of shape, rotation axes and rotation rates are the first needed parameters as they are critical for relative navigation. This estimation should be run during identification phase, a few meters away from the target after approach is completed. As discussed in previous chapter, two different strategies could be considered: either the chaser stays on a station Keeping point while the debris is tumbling – which should provide enough information for complete shape and rotation estimation - , or the chaser is set on a football orbit (radial ΔV) to "orbit" around the debris and observe it for reconstruction. From an identification perspective, the two different strategies should be equivalent.

Reconstruction is a very challenging step that could be critical depending on captures devices (e.g. arm because it might be needed to align the chaser with main rotation axis, or harpoon because it might be needed to precisely hit the debris). However, although there are several papers describing different algorithms or strategies, the reconstruction chain, which includes sensor and image processing algorithms, has to be consolidated. Effort on design, validation and verification is yet to be provided.

Based on existing technologies throughout Europe, there are two main families of sensors: active and passive sensors. Passive sensors such as visible camera like NPAL-based solution (monocular, passive camera) have already been investigated in frame of rendezvous phase [23]. Infrared sensors are also sensors of high interest; it has been off line demonstrated in the frame of Orbital Express rendezvous for instance [24]. Such sensors are known to be power efficient, light weighted, but sensitive to environmental conditions or reflectance of debris. Active sensors have been more recently considered as space rendezvous sensor thanks to recent improvements in active technology. Scanning lidar has been considered by several space agencies [25] [26]. More recently, flash lidar also called 3D camera [27], have been investigated and demonstrated as they do not feature mechanical devices (as opposed to scanning lidar) and can provide an instantaneous 3D picture of the target. A flash lidar, the STORRM mission, was demonstrated by Ball Aerospace on STS-134 in May 2011, in rendezvous and docking with ISS [28]. Nonetheless, such flashing sensors require higher power as the laser energy is to be spread over the Field Of View (as opposed to scanning lidar that focuses laser energy on a single spot, mechanically spread over the FOV).

For the past decade, image processing domain has made significant improvements for reconstruction, based on 2D or 3D data. Several techniques are currently being investigated to reconstruct an unknown target. A recent Innovative Triangle Initiative with ESA, Astrium and INRIA has demonstrated the capability of 3D reconstruction from 2D pictures, based on Structure From Motion (SFM) and Shape From Shading (SFS) methods.

However, techniques based on SFS might suffer from an apriori knowledge of materials reflectance which might not be compatible with MLI of defunct satellites for instance. Illumination conditions are also to be considered. Other technique such as Shape From Silhouette (SFSi) [29] could be a valid algorithm for reconstruction of unknown objects as it relies on building a 3D model from 2D silhouette.

Reconstruction performances of such algorithm have still to be demonstrated on typical debris. Besides, another major step to be overcome is the computer efficiency of such algorithms on space processors.

After reconstruction of the 3D model of the target, the identification phase is completed. Ground can decide the best way to capture the debris.

Online Estimation of Relative State

As discussed previously, the relative navigation should start during approach phase, i.e. at about 2 km from target and should provide necessary outputs for optimal manoeuvres toward target and anti-collision avoidance. The main challenge of the relative navigation function is to work under many environmental conditions, (Sun elevation, eclipse, Earth, Moon in background), for many different targets, and within a large span of relative distances between chaser and target while being reliable, autonomous and CPU efficient. A few solutions, coupling sensors and innovative algorithms, could meet such stringent requirements. For instance, simple and CPU efficient solution can be considered as long as relative position and velocity are only needed (typically during approach phase). Then, more complex algorithms should be considered for estimation of relative attitude once the target has been reconstructed.

Relative Position and Velocity

Relative position and velocity can be easily computed with simple image processing algorithms, relying on passive or active sensors, as discussed for identification phase. It could be computed from 3D data or 2D data.

Regarding 3D data, several solutions can be considered. For instance, relative position and velocity can be estimated from a mean estimation of 3D points, provided by active sensor (flash lidar, scanning lidar). Fused with data from Inertial Measurement Unit (IMU) within navigation filter, this would easily provide mean distance and velocity to target. Indeed, measurements of rotation rate (provided by gyroscope) are needed by the filter to tell rotation from translation of debris in the sensor FOV. Typical accuracy of active sensors is about a few centimetres at beginning of identification phase. The mean distance should therefore be accurate enough for safe operations, at low CPU cost. Another approach could also consist in considering the closest point of the cloud of 3D points provided by the sensor.

The distance would then be used as measurement of distance and be provided to the navigation filter along with IMU data. This solution however will not be robust to outlier. However, as direct 3D measurements are provided by active sensor, such solution should be more power demanding than passive sensor-based solution.

Regarding relative navigation based on 2D data, infrared or visible images could be used and should be able to provide enough information to compute relative distance, provided state of the art image processing. One solution is discussed further in coming paragraph, for which an a-priori model (a priori 3D model or reconstructed model) of the target should be needed to compute relative position and velocity. As 2D data can be provided by passive sensors, such solution should be less power demanding and however highly CPU demanding.

Relative Attitude

Prior to capture, i.e. right after identification phase, the relative navigation should also provide estimation of relative attitude. This estimation can be done as soon as a target's model is available. Onboard image processing could hence rely on 3D model of target and match measurements (2D or 3D) with model to estimate the relative attitude, along with position and velocity. Once the relative attitude is estimated, capture operations can be planned. Solution for comprehensive relative navigation is discussed in the following paragraph as well.

Alike image processing for identification, image processing for relative navigation (in particular for estimation of relative attitude) have yet to be investigated further.

3 *Solutions for* Relative Navigation *with* Debris

As discussed previously, relative navigation with debris is a critical issue for ADR mission. Two solutions for relative navigation have been investigated by Astrium in collaboration with INRIA, with sensors investigated under ESA or European Commission (FP7) studies. The first solution relies on an active, 3D camera - currently developed by CSEM in frame of FP7 Fosternav study - fused with IMU data in a navigation filter. The second solution relies on a passive, 2D camera and a state-of-the-art Image Processing that provides pseudo-measurements, also fused with IMU data in the navigation filter.

3.1 3D/3D Matching

The first relative navigation relies on 3D pictures, provided by active sensor, and 3D model of the debris, either a priori known or estimated during identification phase. 3D pictures are provided by state of the art 3D flash imaging lidar, currently developed in the frame of the FP7 project “Fosternav”.

The key components of flash imaging lidar are the laser illuminating the target and the receiver detector array. These two elements are operated in full synchronisation to generate three dimensional images of the target. The device determines the time-of-flight (TOF) of photons by measuring the phase difference between the modulated illuminating laser beam and the incoming back-reflected light per pixel. One of the main challenges is the design of the laser head that should cover the range chaser-debris over which the relative navigation should be performed, under varying environmental conditions with, possibly, Earth as background and within power capabilities of spacecraft. The current solution considered in the frame of FP7 project should have a range measurement precision of 2-3 cm and a power consumption of 3W (laser illumination not included, no duty-cycle).

Direct 3D measurements should therefore be provided to the image processing algorithms that would filter measurements.

Regarding image processing, many publications on 3D cloud matching are available [31] [32]. Basically, it would be model-based, as it would aim to match the 3D model of the debris with the 3D point cloud provided by the camera. Many algorithms have been developed in the computer vision domain to solve this problem, but for different applications such as search in 3D database or object recognition.

The relative navigation solution therefore consists in a 3D camera that provides 3D pictures to image processing for matching with a known model. As there is always an ambiguity between pure rotation and translation of the target within FOV, fusion of IMU data within navigation filter is needed. As a result, reliable estimation of relative position, velocity and attitude are provided by the proposed navigation solution.

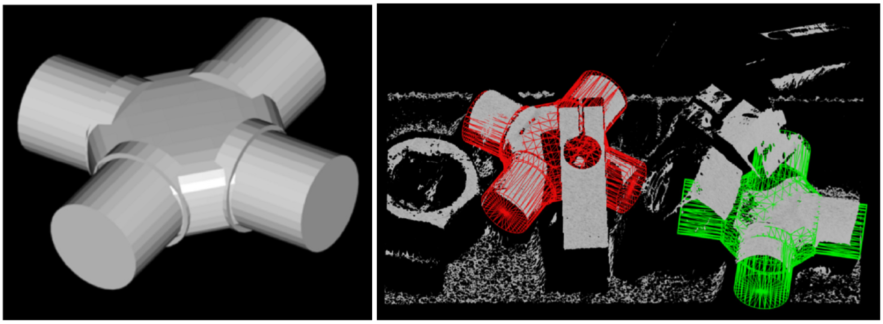


Fig. 7 Example of 3D/3D matching algorithm. (left) a priori known 3D model of object. (right) matching of partial 3D point cloud (depth map) with 3D model [33].

3.2 2D/3D Matching

One of the investigated solutions by Astrium in collaboration with INRIA relies on 2D pictures, post processed to match a 3D model of the target. It therefore can only be used when the 3D model of the debris is a priori known or estimated during identification. Alike the 3D/3D matching, it can provide estimation of relative position, velocity and attitude, necessary conditions for successful capture. Such approach is divided into two steps: initialization and tracking.

Initialization

Initialization aims at detecting the target in an image sequence and at providing the tracking with an initial guess of the target pose, without any prior information on the pose. It consists in matching (detection/matching stage) the image contours with a database of views built during the identification phase. This initialisation is done stepwise.

Identification Learning. A hierarchical model view graph leading to prototype views of the model is built. Each node of the view graph contains an image projection of the target contours at a particular point of view. The points of view are sampled on a sphere. The sampling of the views is optimized to limit the memory size of the database and to insure the whole coverage of the space of possible views.

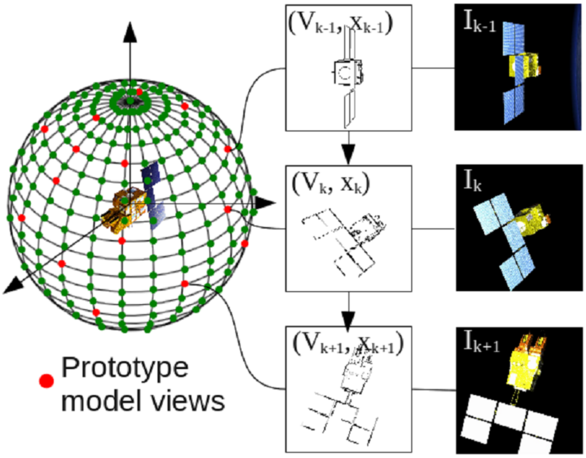


Fig. 8 Principle of initialization. Several views on a sphere are selected to produce prototype views stored in a hierarchical model view graph. Target is then extracted by segmentation and matched over successive frames with closest prototype.

Online Target Detection. Silhouette of the target is extracted in the image using bilayer segmentation techniques. This method consists in minimizing an energy function combining motion and color, along with temporal and spatial priors. It allows distinguishing the foreground shape from the background and has the advantage to be real-time.

Online Matching and Pose Initialization. The view graph is then explored to find the prototype view whose contours correspond the most to the extracted silhouette. The used similarity metric derives from [30]. It considers both the distance and the orientation of edges to match: Once the closest prototype view is found, its associated pose is considered as initialization of the target pose.

The matching stage can be rather time consuming. To cope with real time, a Bayesian framework is set to spread the initialization over several images (temporal initialization). It enables to provide an up to date pose initialization to the GNC system.

Tracking

Once the target has been detected in image, and its pose has been initialized, a frame to frame edge tracking is performed.

Like initialization, tracking is then 3D model based. It aims at finding the target pose which makes best match the projection of the 3D model with the image edges. Tracking and pose estimation are thus simultaneous. Unlike initialization, the edge matching is local. As a consequence, tracking runs in real time but is less robust to high differences between edges, meaning that predicted target pose shall be close enough (tens of pixels) to real one.

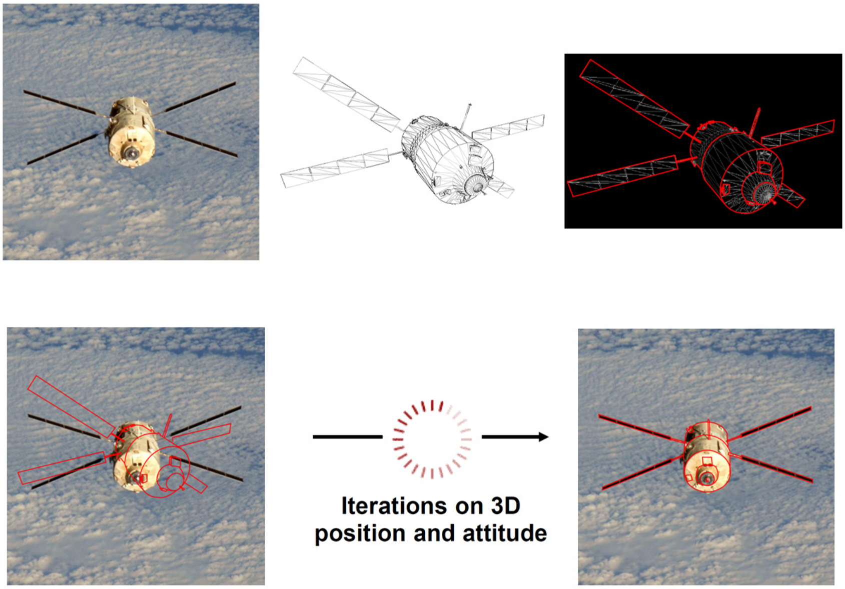


Fig. 9 Principle of tracking. Tracking is performed using a frame (1) and the 3D model of the target (2). The salient edges of the target are extracted (3) and projected into image (4), given an initial pose. Pose is iteratively refined to make projection edges match with image edges (5).

4 Conclusion

Mission dedicated to Active Debris Removal will have many challenges to face. Aside financial and legal issues, several technological solutions have still to be designed, consolidated and validated. From a GNC perspective, there are still many issues to be solved. First, a comprehensive model of debris – chaser behaviour, when linked through a non rigid or rigid link, is to be derived and validated. Then, robust guidance to environmental uncertainties, navigation dispersion, realization errors and most importantly to uncertainties on debris should be consolidated and validated. Adaptive control to prompt change in mass and inertia or sporadic high torques due to capture is also to be investigated. G&C solution for the de-orbiting phase, with towed debris in particular is to be considered further. Lastly, solution of relative navigation capable of reconstructed any kind of debris under changing environmental conditions have to be consolidated and validated as well. The selected navigation solution shall also be capable of real time estimation of relative position, velocity and attitude. These blocks all together should finally be designed to ensure anti-collision, efficient FDIR and possibly safe mode during the very critical phases of approach, identification, capture and de-orbiting.

Many building blocks are already available for these different functions and current industrial studies or academic work in modern control, robust guidance, state of the art image processing or active sensors provide consolidated designs to build upon. In particular, two navigation solutions are being thoroughly investigated by Astrium, CSEM and INRIA. These solutions, either based on a 3D flash sensor or a 2D passive camera coupled with image processing, are capable of providing relative position, velocity and attitude of the chaser w.r.t. unknown debris.

References

1. Kessler, D.J.: Collisional cascading: the limits of population growth in Low Earth Orbit. Advances in Space Research 11(12), 63–65 (1991)
2. Orbital Debris Quarterly News. NASA 13(2) (April 2009)
3. Krag, H., Virgili, B.B.: Removal Target Selection and its environmental effet. In: Cleanspace Workshop on Active Space Debris Removal, ESOC facility, Darmstadt, Germany, September 17-18 (2012)
4. Bonnal, C., Alby, F.: Introduction to 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
5. Metz, M.: DLR Perspective on Sustainable Use of Space. In: Cleanspace Workshop on Active Space Debris Removal, ESOC facility, Darmstadt, Germany, September 17-18 (2012)
6. Liou, J.C.: Challenges and Opportunities for Orbital Debris Environment Remediation. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
7. Kawamoto, S., et al.: Current Status of studies on Active Debris Removal at JAXA. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)

8. Technical Report on Space Debris, United nations Publication, Sales On. E.99.I.17, ISBN 92-1-100813-1
9. Weigel, M., Patyuchenko, A.: Orbit Determination error analysis for a future space debris tracking radar. In: European Space Surveillance Conference, INTA headquarters, Madrid, Spain, June 7-9 (2011)
10. Lockheed Martin Space Fence Radar Prototype Tracking Orbiting Objects, `http://www.lockheedmartin.com`
11. Report on Space surveillance, Asteroids and Comets, and Space Debris, SAB-TR-96-04, US Air Force Scientific advisory Board, vol. 1 (June 1997)
12. Retat, I., Bischof, B., et al.: Net Capture System : a potential orbital Space Debris Removal System. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
13. Reed, J.: Development of a Grappling System for Capturing Heavy Space Debris. In: 63rd International Astronautical Congress, Naples, Italy (October 2012)
14. Rembala, R., Allen, A., Teti, F.: Robotic Capture of Large Orbital Debris. In: Cleanspace Workshop on Active Space Debris Removal, ESOC facility, Darmstadt, Germany, September 17-18 (2012)
15. Kitamura, S., et al.: A reorbiter for large GEO debris Objects using Ion Beam Irradiation. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
16. Bombardelli, C.: A plan to Deorbit Envisat. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
17. Schaub, H., Moorer, D.F.: Touchless reorbiting of large geosynchronous debris. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
18. Gerber, B., Cougnet, C., Alary, D., Utzmann, J., Wagner, A.: The debritor: an "off the shelf" based multimission vehicle for large space debris removal. In: 63rd International Astronautical Congress, Naples, Italy (October 2012)
19. Janhunen, P., Kvell, U., Seppänen, H.: Electrostatic plasma brake tether for deorbiting small satellites. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
20. Lappas, V.: Deorbiting an active debris removal using Gossamer systems and small satellite technology. In: 2nd European Workshop on Active Debris Removal, CNES Headquarters, Paris, France, June 18-19 (2012)
21. Blanc-paques, P., Gogibus, E., Louembet, C., Kara-Zaitri, M.: State of the art guidance techniques for rendezvous and withdrawal strategy. In: 4th International Conference on Astrodynamics Tools and Techniques, Madrid, Spain (2010)
22. Flandin, G., Dinh, M., Scorletti, G., Fromion, V., Beugnon, C., Lemaire, J., Ganet, M., Bérard-Chiappa, C., Biannic, J.-M.: LPV techniques applied to industrial space applications. In: 7th ESA GNC Conference, Tralee, Irlande, Juin 1-5 (2008)
23. Despré, N., Kerambrun, S., et al.: HARVD, an autonomous visioni-based system for rendzvous and docking. In: 4th International Conference on Astrodynamics Tools and Techniques, Madrid, Spain (2010)
24. Weissmuller, T., Leinz, M.: GNC technology demonstrated by the orbital express autonomous rendezvous and capture sensor system. In: 29th Annual AAS Guidance and Control Conference, Breckenridge, Colorado (2006)

25. Moebius, B., Pfennigbauer, M., Pereira do Carmo, J.: Imaging lidar technology - development of a 3D lidar elegant breadboard for rendezvous and cosking, test results, and prospect to future sensor applilcation. In: International Conference on Space Optics, Rhodes, Greece (October 2010)
26. Allen, A.C.M., Langley, C., Mukherji, R., Taylor, A.B., Umasuthan, M., et al.: Rendezvous lidar sensor system for terminal rendezvous, capture and berthing to the international space station. In: Proc. SPIE 6958, Sensors and Systems for Space Applications II, 69580S
27. Pollini, A.: Flash optical sensors for Guidance, navigation and control systems. In: 35th Annual AAS Control and Guidance Conference, Breckenridge, Colorado (February 2012)
28. Gravseth, I.J., Rohrschneider, R., Masciarelli, J.: Vision Navigation Sensor (VNS) results from the STORRM mission. In: 35th Annual AAS Control and Guidance Conference, Breckenridge, Colorado (February 2012)
29. Cheung, K., Baker, S., Kanade, T.: Shape from silhouette across time part I: theory and algorithms. International Journal of Computer Vision 62(3), 221–247 (2005)
30. Reinbacher, C., Ruether, M., Bischof, H.: Pose estimation of known objects by efficient silhouette matching. In: 20th International Conference on Pattern Recognition, ICPR (2010)
31. Huang, J.: Point cloud matching based on 3D self-similarity. In: Computer Vision and Pattern Recognition Workshop, IEEE Computer Society Conference (2012)
32. Suzuki, T., Kitamura, M., Amano, Y., Hashizume, T.: 6_DOF localization for a mobile robot using outdoor 3D voxel maps. In: IEEE/RSJ International Conference on Intelligent Robot and Systems, Taipei, Taiwan (October 2010)
33. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2010)

Author Index