Bogusław Dołęga
Robert Głębocki
Damian Kordos
Marcin Żugaj
Editors

Advances in Aerospace Guidance, Navigation and Control

Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, in April 2017

Springer

Advances in Aerospace Guidance, Navigation and Control

Bogusław Dołęga · Robert Głębocki
Damian Kordos · Marcin Żugaj
Editors

Advances in Aerospace Guidance, Navigation and Control

Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, in April 2017

Springer

Editors

Bogusław Dołęga
Avionics and Control Systems Department
Rzeszów University of Technology
Rzeszów
Poland

Damian Kordos
Avionics and Control Systems Department
Rzeszów University of Technology
Rzeszów
Poland

Robert Głębocki
The Institute of Aeronautics and Applied Mechanics
Warsaw University of Technology
Warsaw
Poland

Marcin Żugaj
The Institute of Aeronautics and Applied Mechanics
Warsaw University of Technology
Warsaw
Poland

ISBN 978-3-319-65282-5 ISBN 978-3-319-65283-2 (eBook)
https://doi.org/10.1007/978-3-319-65283-2

Library of Congress Control Number: 2017949119

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The 2017 CEAS (Council of European Aerospace Societies) Specialist Conference on Guidance, Navigation and Control (CEAS EuroGNC) was organized together by Warsaw University of Technology (WUT), the largest technical university in Poland and the Rzeszów University of Technology University of Technology (RzUT), an important centre in aviation. The EuroGNC Conference was held on 25–27 April 2017. It was chaired by Robert Głębocki, from the Faculty of Power and Aeronautical Engineering (WUT), Faculty which has a long history in aerospace engineering and providing important contribution to the aviation development in Poland. The EuroGNC Conference was co-chaired by the Bogusław Dołęga, On behalf of the RzUT, from faculty of Mechanical Engineering and Aeronautics (RzUT), faculty which achieved the status of an Advanced Technology Centre within the AERONET Aviation Valley. The International Program Committee, that gathered over 40 finest scientists and researchers form all over the world, put an effort and commitment leading the preparation of the Conference to the success.

The CEAS EuroGNC 2017 Conference aims to promote scientific and technical excellence in the fields of Guidance, Navigation and Control (GNC) in aerospace and other fields of technology. The Conference joins together the industry with the academia research. It creates an opportunity for better understanding the incoming challenges in the development of novel GNC methods, applications and technologies. Submitted papers were carefully reviewed, and the best 40 were selected to be published in this monograph. The book covers four main topics: Guidance and Control, Control Theory Application, Navigation, UAV Control and Dynamic. The papers included focus on the most advanced and actual topics in guidance, navigation and control research areas.

In aviation, the mission diversity constantly changes that consequently extorts new problems in terms of control and dynamics. As an alternative to satellites, the High Altitude Long Endurance (HALE) aircraft needs to cope with set of specific problems. A new flight path control laws for a multi-body aircraft was implemented to perform a one-year mission. A new design method and structure for the inner-loop was applied; the H_∞ loop shaping in the frequency domain was successfully used.

Development of optimization and control laws design methods resulted in a new workflow for the clearance of flight control laws with continuous control or disturbance inputs using Optimal Control Theory and Post-optimal Sensitivity Analysis.

The sensors integration and the data fusion is still a challenge in navigation. The optimal performance of the conventional Kalman filters is deteriorated if there is uncertainty in the process and measurement noise covariance. To reduce this effect, a Fuzzy Adaptive Iterated Extended Kalman Filter (FAIEKF) and Fuzzy Adaptive Unscented Kalman Filter (FAUKF) were developed to overcome this issue. The UAV domain is growing extremely fast, and many challenges that already solved in aviation become still a challenge due to the complexity and variety of the platforms. The increasing availability and affordability of the UAVs platforms allowed to perform a new set of various missions that demand new control laws, techniques and have to cope with different navigation problems. One of the emerging challenges is to manage number of UAVs performing common mission in an efficient and safe manner. The modified sequential greedy algorithm was proposed to enhance the efficiency of task allocation for cooperative parcel delivery problem of multiple unmanned aerial vehicles (UAVs). The other significant UAV aspect is safety which has an important role in aviation in general. Generally, UAVs have not reached sufficient level of integrity to be implemented in the civil airspace. The reconfigurable method for UAV platform was developed to obtain performances of a damaged aircraft as close as possible to an undamaged one using advanced algorithms.

The CAES EuroGNC 2017 Conference would not be a success if not a strong support of many people. On behalf of National Organizing Committee, we would like to thank to all contributors to the conference, especially to the Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, the Institute of Aeronautics and Applied Mechanics, International Committee, American Institute of Aeronautics and Astronautics (AIAA), and to the all reviewers of technical papers.

We hope that this book will help the reader to understand the most advanced and emerging challenges in Guidance, Navigation and Control domain.

Warsaw, Poland
April 2017

Robert Głębocki
Bogusław Dołęga

About the Book

For the 4th CEAS Specialist Conference on Guidance, Navigation and Control, the International Technical Committee established a formal review process. Each paper was reviewed in compliance with good journal practices by independent and anonymous reviewers. At the end of the review process, papers were selected for publication in this book.

The members of the International Technical Committee are:

Daniel Alazard	ISAE, France
Mark Balas	University of Wyoming, USA
Samir Bennani	ESA/ESTEC, The Netherlands
Daniel Choukroun	Ben-Gurion University of the Negev, Israel
John Crassidis	University at Buffalo, USA
François Defay	ONERA, France
Jörg Dittrich	DLR, Germany
Bogusław Dołęga	Rzeszów University of Technology, Poland
Chris Edwards	University of Leicester, UK
Alexej Efremov	Moscow Aviation Institute, Russia
Pierre Fabre	Airbus, France
Patrick Fabiani	ONERA, France
Walter Fichter	Universität Stuttgart, Germany
Chris Fielding	BAE Systems, UK
Benoit Frapard	EADS Astrium, France
Robert Głębocki	Warsaw University of Technology, Poland
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
Erik-Jan van Kampen	TU Delft, The Netherlands
Karl-Heinz Kienitz	ITA, Brasil
Youdan Kim	Seoul National University, South Korea
Philipp Kraemer	Eurocopter, Germany

Gertjan Looye	DLR, Germany
Marco Lovera	Politecnico di Milano, Italy
Mark Lowenberg	Bristol University, UK
Robert Luckner	Berlin Technical University, Germany
Felix Mora-Camino	ENAC, France
Bob Mulder	TU Delft, The Netherlands
Janusz Narkiewicz	Warsaw University of Technology, Poland
Guillermo Ortega	ESA, The Netherlands
Henry de Plinval	ONERA, France
Charles Pussot-Vassal	ONERA, France
Arthur Richards	Bristol University, UK
Stephan Theil	DLR, Germany
Frank Thielecke	TU Hamburg, Germany
Andrzej Tomczyk	Rzeszów University of Technology, Poland
C.C.deVisser	TU Delft, The Netherlands
Ali Zolghadri	IMS Bordeaux, France

Contents

Part I Guidance and Control

Part II Estimation and Navigation

Part III Atmospheric Applications

Part IV Space Applications

Part I
Guidance and Control

Reconfiguration Control Method for Faulty Actuator on UAV

Adèle Boche, Henry De Plinval and Jean-Loup Farges

1 Introduction

Unmanned Aerial Vehicles (UAVs) are used in many areas, notably one can mention search and rescue operations or photography. As a result, the number of UAVs has dramatically increased with slower safety improvements, leading to a net growth of the number of incidents. Consequently, the safety of UAVs is more and more important. Studies have identified the hazards that can impact on the flight safety. On the one hand, the faults that can arise in flight have been considered in several works [8], and on the other hand, the impacts of operating risks on UAVs have also been studied [3]. These studies indicate that UAVs resiliency and their capability to be reconfigured in face of hazards shall be developed.

Many control systems are subject to component failure. Thus, several works focus on reconfiguring systems to improve their safety. There are two types of reconfiguration methods, the passive one and the active one. In the case of passive reconfiguration, the system is made robust against faults without an explicit detection and identification of those faults [2]. Active reconfiguration consists in detecting and identifying faults, and then changing consequently the control. For large aircraft, hardware redundancy ensures safety, for instance by tripling key components. The analytical redundancy seems to be more adapted for UAVs because of the cost, price, mass, and consumption of component triplication. Analytical redundancy uses signals generated from a mathematical model. In the last decades, many works have been

A. Boche (✉) · H. De Plinval · J.-L. Farges
ONERA - The French Aerospace Lab, 2 Avenue Edouard Belin, 31000 Toulouse, France
e-mail: Adele.Boche@onera.fr

H. De Plinval
e-mail: Henry.de_Plinval@onera.fr

J.-L. Farges
e-mail: Jean-Loup.Farges@onera.fr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_1

done on this subject [4, 7, 9]. Recently, an active control reconfiguration framework, based on continuous and discrete models of faults, has been proposed [1].

The purpose of the work presented here is to investigate the applicability of this reconfiguration framework to UAVs fault tolerance problems. This investigation begins with the use of the framework for the simplified longitudinal dynamics of Altimum 4 model in a landing scenario. This article is organized as follow. In Sect. 2, the problem is described and an overview of the reconfiguration framework is provided. Then, in Sect. 3, the system is modelled following the continuous and discrete framework, and the controllers used are developed. In Sect. 4, the reconfiguration method is adapted in order to take into account specificities of the problem. Finally, Sect. 5 shows simulation results for actuator faults.

2 Objective

2.1 Problem

In this article, the hazards considered are the faults of an actuator. The locking and the loss of effectiveness case are studied. In order to consider a dramatic case of failure, the scenario is based on UAV autopilot landing procedure.

The UAV used here is an Altimum 4 built by the company “L’Avion Jaune” and adapted by ONERA for research purposes. It is a fixed-wing UAV with back thermal engine. This UAV has the design of a flying wing of 4 m wingspan, as shown in Fig. 1. The pitch and roll of the UAV are controlled using a set of elevons which have the function of either elevators or ailerons.

The Altimum 4 behaviour in flight has been modelled [5]. Based on this model, it is assumed that the longitudinal and the vertical models can be decoupled.

The problem addressed is then to control the longitudinal motion of the UAV to continue to follow the landing glide slope despite the possible fault of the actuator.

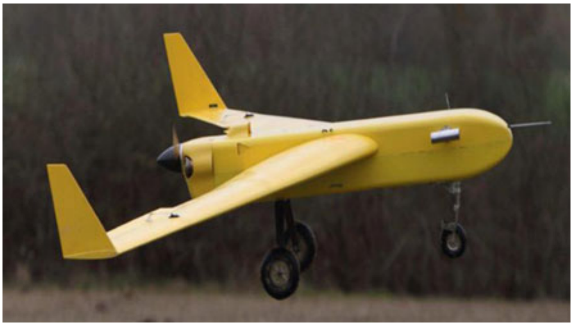


Fig. 1 The Altimum 4

2.2 *Overview of the Framework*

The control reconfiguration framework considers a coupling between a continuous model and a discrete model.

2.2.1 Continuous Model

Several aspects of the system behaviour and its controllers are represented by continuous dynamics. The continuous model includes:

- A continuous state equation which includes the traditional states together with several fault parameters.
- A set of controllers designed for nominal and faulty modes. The framework considers only single faults, a nominal mode and a mode for each fault and a controller designed for each mode.
- An indicator of performance which is calculated from the error between a reference trajectory and the actual one.
- An estimation scheme based on the continuous state equation.

2.2.2 Discrete State Model

Some aspects of the system and controller behaviour are modelled with discrete state dynamics. The discrete state model includes:

- System modes -nominal, locking, loss of effectiveness-.
- Transition probabilities between these modes.
- A decision making framework to chose the best suitable controller for the continuous model.
- A reward associated to each couple -mode, controller-, which is computed from the indicator of performance of the continuous model.
- Constraints on fault parameters for each mode.

This discrete model is formally a specific case of a non stationary Partially Observable Markov Decision Process (POMDP) [1]. It is defined by the 7-tuple $\{E, S, A, T, R, \Omega, O\}$ with:

- $E = \{0, 1, ..., N\}$ is a set of time steps.
- $S = M \times K$ is the set of states, with M the set of modes and K the set of controllers. Note that $|M| = |K|$ because one controller is developed for each mode. A state is defined by the current mode m_i and the current controller k_j; $s_{i,j} = (m_i, k_j)$.
- A is the set of actions. Note that $|A| = |K|$ because an action is the choice of the controller for next time step.
- $T = S \times A \times S \rightarrow [0, 1] = T_M.T_K$ where $T_M : M \times M \rightarrow [0; 1]$ is a set of conditional transition probabilities between modes, and $T_K : A \times K \rightarrow \{0, 1\}$ indicates the deterministic selection of a controller by an action.

- $R : S \times A \times E \rightarrow \mathbb{R}$ is the time dependant cost function,
- $\Omega = \{\omega\}$ is the observation, and
- $O : M \times \Omega \rightarrow [0; 1]$ is a set of conditional observation probabilities.

2.3 Overview of the Reconfiguration Method

The proposed reconfiguration method can be described with its off-line and on-line parts. Off-line, once the modes defined, one computes associated controllers, aimed at maximizing the performance indicator. Then, for each couple -mode, controller-, the expected performance indicator is computed. On-line, the reconfiguration method performs at each sampling period the following steps:

1. Use of the selected controller to control the system,
2. Estimation of state and fault parameters using the continuous state equation,
3. Update of mode probabilities from estimated fault parameters,
4. Decision making step to select the controller minimizing the future expected cost.

3 Modelling of the Problem

3.1 Modelling of the Continuous Behaviour

The UAV longitudinal model linearised for a typical landing, with a given air speed and a given slope can be described as:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A_c\mathbf{x}(t) + B_c\mathbf{e}(t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{x}(t) + \mathbf{v}(t) \end{aligned} \tag{1}$$

where $\mathbf{x} = \begin{bmatrix} u & w & q & x & z & \theta \end{bmatrix}^T$, $\mathbf{e} = \begin{bmatrix} \delta_x & \delta_l & \delta_r \end{bmatrix}$, $\mathbf{w}$ correspond to modelling errors, and $\mathbf{v}$ correspond to the measurement noise. The states are: horizontal velocity u, vertical velocity w, pitch rate q, horizontal position x, vertical position z and pitch angle θ; the control inputs are: thrust command δ_x, left elevator command δ_l and right elevator command δ_r; the outputs correspond to the states.

Physically, asymmetric control of δ_l and δ_r induces a roll motion in the lateral model. It is assumed that this induction is compensated by the action of the lateral controller on left and right ailerons.

The system is linearised, thus the inputs, the states and the outputs are the differences between the actual state and the trim state. Figure 2 depicts the general structure of the system with possible actuator faults in the discrete domain.

In the Fig. 2, k is the time index, $\mathbf{u}_k \in \mathbb{R}^l$ is the desired control input, $\mathbf{e}_k \in \mathbb{R}^l$ is the effective control, $\mathbf{x}_k \in \mathbb{R}^n$ is the system state, $\mathbf{y}_k \in \mathbb{R}^m$ corresponds to the system

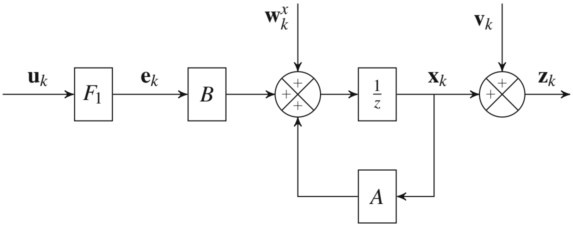


Fig. 2 Discrete time model of the continuous state dynamics with possible actuator faults

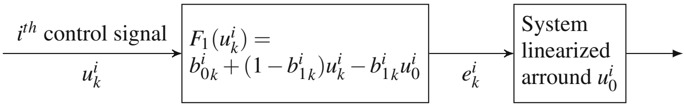


Fig. 3 Modelling of the ith actuator faults for a trimmed system

outputs, $\mathbf{z}_k \in \mathbb{R}^m$ corresponds to the measurement, $\mathbf{w}_k^x$ is a zero-mean white Gaussian noise sequence with covariance Q_k^x representing the modelling errors, $\mathbf{v}_k$ is a zero-mean white Gaussian measurement noise sequence with covariance R_k, A and B are known matrices, and F_1 is a function which models the actuator faults.

The function F_1 is different from the one of the original framework. Indeed, the model of the system used here is linearised. Thus, considering $U^i(t)$ the ith desired input for the non linear model and $E^i(t)$ the ith effective input for the non linear model, the function F_1 of the original framework is applicable to $E^i(t)$ and $U^i(t)$ and $e^i(t)$ and $u^i(t)$ are delta:

$$\begin{aligned}
E^i(t) &= {b_0}^i(t) + (1 - {b_1}^i(t))U^i(t) \\
u^i(t) &= U^i(t) - u_0^i \\
e^i(t) &= E^i(t) - e_0^i
\end{aligned} \tag{2}$$

where u_0^i and e_0^i are respectively the trim desired and effective inputs.

Because the system is linearised for a trim without fault, $e_0^i = u_0^i$. And thus, the function F_1 is as follows:

$$e_k^i = F_1(u_k^i) = b_{0k}^i + (1 - b_{1k}^i)u_k^i - b_{1k}^i u_0^i \tag{3}$$

Because, the inputs are the differences between actual inputs and trim inputs, the function linking the effective control to the desired control presents an additional term, $-b_{1k}^i u_0^i$, with respect to the same function for a genuine linear system. Physically, this additional term corresponds to the fact that the aircraft is trimmed for a non faulty state. The fault model is presented in its context on Fig. 3.

The output $\mathbf{e}_k$ corresponds to the delta physical motion performed by the actuator, that is limited in its amplitude. $\mathbf{e}_k$, and $\mathbf{u}_k$ are bounded in $[-\bar{\mathbf{u}} - \mathbf{u_0}, \bar{\mathbf{u}} - \mathbf{u_0}]$, where $\bar{\mathbf{u}}$ is the maximum value for the absolute value of the actual input.

Starting from Fig. 2 and the modelling of F_1 (3), the discretised equations of the continuous system are:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k + B\mathbf{b_0}_k + D(\mathbf{u}_k)\mathbf{b_1}_k + \mathbf{w}_k^x \\ \mathbf{z}_k &= \mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{4}$$

with $D(\mathbf{u}_k) = B \begin{bmatrix} -u_k^1 - u_0^1 \cdots & 0 \\ \vdots \quad \ddots & \vdots \\ 0 \quad \ldots & -u_k^l - u_0^l \end{bmatrix}$

The values of the parameters $\mathbf{b_0}$ and $\mathbf{b_1}$ change in the case of faults. From the estimation viewpoint, these values and their behaviour are unknown. In the continuous part of the model their behaviours are supposed to be random walks:

$$\begin{aligned} \mathbf{b_0}_{k+1} &= \mathbf{b_0}_k + \mathbf{w}_k^{b_0} \\ \mathbf{b_1}_{k+1} &= \mathbf{b_1}_k + \mathbf{w}_k^{b_1}. \end{aligned} \tag{5}$$

3.2 Modes and Controllers

There is one controller k for each mode m. Each controller sends different control signal, $\mathbf{u}_k$, for the effective control $\mathbf{e}_k$ to be like:

$$\mathbf{e}_k = -K\widehat{\mathbf{x}}_k \tag{6}$$

Because $\mathbf{x}_k$ is not available, the state estimate $\widehat{\mathbf{x}}_k$ is used in the control law.

The controller for faulty mode should be adaptive, based on the estimation of the fault parameters $\widehat{\mathbf{b_0}}_k$ and $\widehat{\mathbf{b_1}}_k$. Without fault, an adaptive controller is not required because $\mathbf{b_0}$ and $\mathbf{b_1}$ are null. Besides, $\widehat{\mathbf{b_0}}$ and $\widehat{\mathbf{b_1}}$ are noisy because they are estimates. So, in the nominal mode, a controller not taking into account the estimated parameters seem to be more suited than an adaptive controller which is sensitive to noise.

3.2.1 Linear Quadratic Regulator for Nominal Mode

The gain K is calculated using the Linear Quadratic Regulator (LQR) method [6]. This method permits to take explicitly the performance objective into account.

For a discrete linear system, with a performance index defined as J_0^∞, with:

$$J_k^{k''} = \sum_{k'=k}^{k'=k''} (\mathbf{x}_{k'}{}^T Q_{LQ} \mathbf{x}_{k'} + \mathbf{u}_{k'}{}^T R_{LQ} \mathbf{u}_{k'}) \tag{7}$$

the optimal control feedback minimizing the performance index is given by:

$$\begin{aligned}\mathbf{u}_k &= -K\mathbf{x}_k \\ K &= (R_{LQ} + B^T PB)^{-1}(B^T PA)\end{aligned} \tag{8}$$

where P is the unique positive definite solution to the discrete time algebraic Riccati equation $P = A^T PA - (A^T PB)(R_{LQ} + B^T PB)^{-1}(B^T PA) + Q_{LQ}$.

In the nominal case, $\mathbf{e}_k = \mathbf{u}_k$ and so the nominal controller is the following:

$$\mathbf{u}_k = -K\widehat{\mathbf{x}}_k. \tag{9}$$

3.2.2 Controllers for Faulty Modes

The definition of the faults to be considered is a part of the modelling. In the original framework, two faults of the jth actuator are considered separately; locking when ${b_0}^j \in [-\overline{u}, \overline{u}]$ and ${b_1}^j = 1$ and loss of effectiveness when ${b_0}^j = 0$ and ${b_1}^j \in]0, 1]$, and two different controllers are developed. However with the fault model for trimmed aircraft of Eq. 3, the loss of effectiveness case have a common feature with the blocking case: a non controlled term to be compensated. For this reason, the framework is adapted in order to consider a single fault corresponding to ${b_0}^j \neq 0$ or ${b_1}^j \neq 0$. The controller for faulty mode is designed as follows.

By definition, in the faulty case, one actuator, e.g. the jth one, has the following form: $\delta + (1 - \gamma)u_k^j - \gamma u_0^j$, while the others are in nominal mode, i.e.:

$$\forall i \neq j \qquad b_{0k}^i = 0 \qquad b_{1k}^i = 0 \tag{10}$$

and,

$$b_{0k}^j = \delta \in [-\overline{u}, \overline{u}] \qquad b_{1k}^j = \gamma \in [0, 1] \qquad b_{0k}^j \neq 0 \vee b_{1k}^j \neq 0 \tag{11}$$

Thus, $\mathbf{e}_k = \left[u_k^1 \cdots u_k^{j-1}\ \delta + (1-\gamma)u_k^j - \gamma u_0^j\ u_k^{j+1} \cdots u_k^l\right]^T$.

To be able to compensate for the faulty actuator, it is assumed that the action of this actuator is a linear combination of the actions of other actuators, i.e. the jth column of B is a linear combination of the other columns of this matrix, i.e. $B_{.j} = \sum_{i \neq j} \lambda_i B_{.i}$, with $B = [B_{.1} \cdots B_{.i} \cdots B_{.l}]$.

On the one hand, having the same feedback than in absence of fault implies:

$$\begin{aligned} B\mathbf{e}_k = -BK\widehat{\mathbf{x}}_k &= -\sum_{i \neq j} B_{.i}K_{i.}\widehat{\mathbf{x}}_k - B_{.j}K_{j.}\widehat{\mathbf{x}}_k && (12) \\ &= -\sum_{i \neq j} B_{.i}(K_{i.} + \lambda_i K_{j.})\widehat{\mathbf{x}}_k && (13)\end{aligned}$$

with $K^T = [K_{1.}^T \cdots K_{i.}^T \cdots K_{l.}^T]$.

On the other hand, the control implies:

$$\begin{aligned}
B\mathbf{e}_k &= \sum_{i\neq j}(B_{.i}u_k^i) + B_{.j}\delta + B_{.j}(1-\gamma)u_k^j - B_{.j}\gamma u_0^j \\
&= \sum_{i\neq j} B_{.i}(u_k^i + \lambda_i\delta - \lambda_i\gamma u_0^j) + B_{.j}(1-\gamma)u_k^j \qquad (14)\\
&= \sum_{i\neq j} B_{.i}(u_k^i + \lambda_i\delta - \lambda_i\gamma u_0^j - \lambda_i\gamma u_k^j) + B_{.j}u_k^j \qquad (15)\\
&= \sum_{i\neq j} B_{.i}(u_k^i + \lambda_i\delta - \lambda_i\gamma u_0^j + \lambda_i(1-\gamma)u_k^j) \qquad (16)
\end{aligned}$$

There are many possibilities for identifying terms related to the feedback with terms related to the control.

Identifying the terms in $B_{.i}$ and in $B_{.j}$ in Eqs. 12 and 15 gives:

$$\mathbf{u}_k = \begin{bmatrix} -(K_{1.} + \lambda_1\gamma K_{j.})\widehat{\mathbf{x}}_k - \lambda_1\delta + \lambda_1\gamma u_0^j \\ \vdots \\ -K_{j.}\widehat{\mathbf{x}}_k \\ \vdots \\ -(K_{l.} + \lambda_l\gamma K_{j.})\widehat{\mathbf{x}}_k - \lambda_l\delta + \lambda_l\gamma u_0^j \end{bmatrix} \qquad (17)$$

With this controller, even if the jth actuator is locked, a signal is still send. Thus, if the actuator recover its performance, the estimator is able to detect it.

In the original framework, two different controllers are defined for locking and loss of effectiveness. For locking, the controller is obtained by assuming $\gamma = 1$ and by identifying the terms in $B_{.i}$ in Eqs. 13 and 16. There is no term in $B_{.j}$ thus u_k^j is arbitrary set to δ. For loss of effectiveness, the controller is obtained by assuming $\delta = 0$ and by identifying the terms in $B_{.i}$ and $B_{.j}$ in Eqs. 12 and 14.

3.3 Discrete Modelling

3.3.1 Time Step

In the original framework, the discrete time model of the continuous state dynamic and the model of the discrete state dynamic have a common time step. The application of the framework to UAVs indicates that the discretisation of Eq. 1 makes sense only considering a sample time that is too small when compared to the time need for performing the computations of step 4 in Sect. 2.3. For this reason, the time step of the steps 3 and 4 is a multiple of the time step of steps 1 and 2. The ratio between sampling times is an integer greater or equal to one, noted μ.

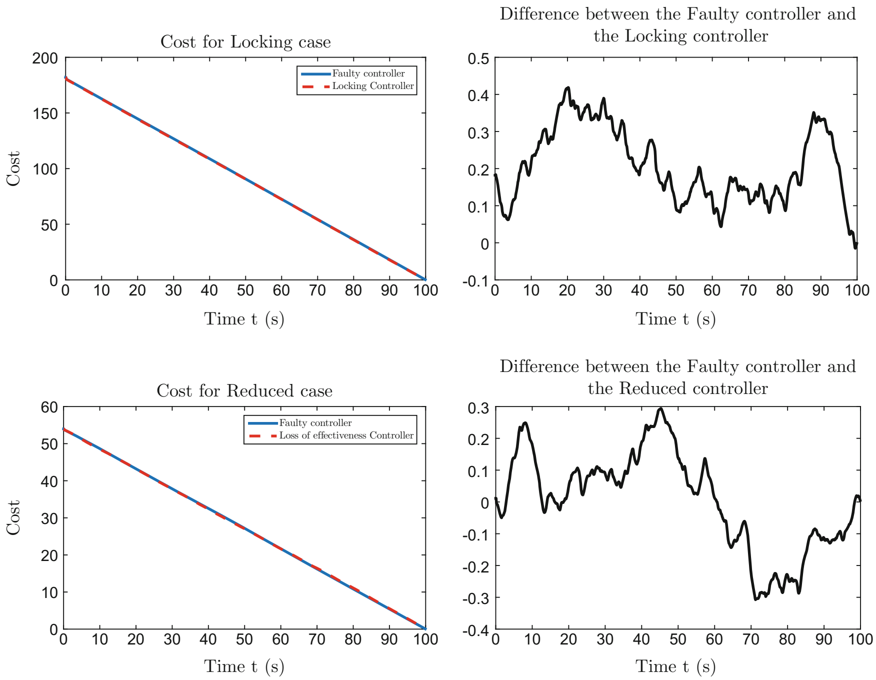


Fig. 4 Performance J_k^{1000} of controllers: in the *upper* part, comparison in the locking case between the controller taking into account both faults and the controller taking into account only locking; in the *lower* part, comparison in loss of effectiveness case between the controller taking into account both faults and the controller taking into account only loss of effectiveness

3.3.2 States, Actions and Transition Probabilities

For the first study, it is assumed that only one actuator can be faulty. The landing trajectory for an airspeed of 20 m/s and a slope of $-3°$, is considered for 100 seconds with a sampling time of 0.1 seconds. The average of the cost J_k^{1000} is computed by performing 100×9 runs respectively for different values for noise sequences and for variation of b_0 or b_1. As shown in Fig. 4, results indicate that a single controller taking into account the values of ${b_0}^j$ and ${b_1}^j$ together presents the same performance than the controller considering only ${b_0}^j$ in case of locking and the same performance that the controller considering only ${b_1}^j$ in case of loss of effectiveness.

Thus, only two modes are considered, the nominal mode m_0 and the faulty mode m_1, and two controllers are developed, k_0 and k_1.

S has four states -two modes and two controllers- and A has two actions a_0, respectively a_1, which select the controller k_0, respectively k_1.

The conditional transition probabilities between modes are stationary and as shown in Fig. 5.

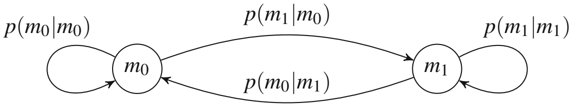


Fig. 5 Directed probabilistic graph; m_0 the nominal mode, m_1 the faulty mode

The corresponding T_M is:

$$T_M = \begin{bmatrix} p(m_0|m_0) & p(m_0|m_1) \\ p(m_1|m_0) & p(m_1|m_1) \end{bmatrix}. \tag{18}$$

3.3.3 Cost Function

In this article, the costs are calculated numerically with the state, i.e. the error between the actual state and the trim. Penalties are added to avoid controller switching without reason and add a conservatism reward. The cost function is as follows:

$$R_n(\{m, k\}, a) = E_{/x_{n.\mu}} I_n^{n''} + P_n(k, a) \tag{19}$$

where

$$\begin{aligned} I_n^{n''} &= J_{n.\mu}^{n''.\mu} \\ n'' &= \min\{n+1, N\} \end{aligned} \tag{20}$$

where $E_{/x_{n.\mu}}$ indicates the expectation with respect to the continuous state.

For the nominal mode, the expectation is computed by performing 100 runs with different values for noise sequences and averaging the cost values. For non nominal mode, the 100 runs are repeated 18 times, varying the fault parameter. The average is then computed considering the 9 × 100 runs for variation of b_0 and 9 × 100 runs for variation of b_1. Figure 6 shows the evolution of the cost $E_{/x_{n.\mu}}(I_n^N)$.

For the nominal case shown in the upper part of Fig. 6; the controllers k_0 and k_1 are very close. Indeed, b_0 and b_1 are null, thus k_1 controls a fault close to zero, which leads to a controller similar to k_0. The difference comes from the noise. Nevertheless the performance of k_0 is slightly better than the performance of k_1. For the faulty case, in the lower part of Fig. 6; the controller k_1 has better performance all along the horizon since the faulty actuator is compensated by the other actuator.

Arbitrary penalties, $P(k_0, a_1) = 1$ and $P(k_1, a_0) = 1$, are added in order to avoid changes of controller at the end of the simulation. Indeed, in Fig. 6, it is shown that costs in nominal mode are similars. For more detailed models those penalties allow to take into account that, for actuators and controllers with internal state, the commutation is not instantaneous and not costless.

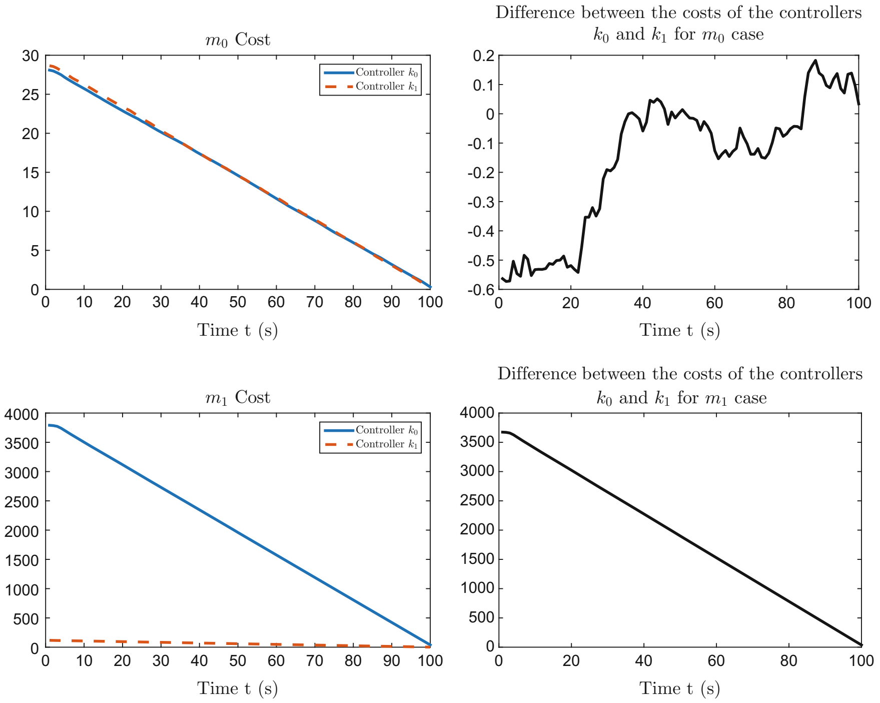


Fig. 6 Cost $E_{/x_{n.\mu}}(I_n^{100})$; *upper* part nominal mode, *lower* part faulty mode

3.3.4 Observations

The classical POMDP framework requires observations taking values in a discrete set; but, here, the observation is based on the information of the continuous processus provided by an estimation scheme: $\omega = \{\widehat{\mathbf{b_0}}_k, \widehat{\mathbf{b_1}}_k\}$.

As explained in [1], one needs to discretise the observation domain to recover a discrete observation for the POMDP. This is done by forcing an *a priori* discretisation on the observation space. Let $\{\Omega^1, ..., \Omega^{|\Omega|}\}$ be a partition of the continuous space in which the ω parameters evolves. To avoid a long computational time, Ω is partitioned in four areas as follows:

- $\widehat{b_0} \in [-0.1, 0.1]$ and $\widehat{b_1} \in [0, 0.1]$
- $\widehat{b_0} \in [-0.1, 0.1]$ and $\widehat{b_1} \in [0.1, 0.9]$
- $\widehat{b_0} \in [-0.1, 0.1]$ and $\widehat{b_1} \in [0.9, 1]$
- $\widehat{b_0} \in [-\overline{u}, -0.1[\cup]0.1, \overline{u}]$ and $\widehat{b_1} \in [0.9, 1]$.

3.3.5 Conditional Observation Probabilities

The probability to have the parameters in a given set Ω^i is given by:

$$p(\Omega^i|m) = \int_{\omega\in\Omega^i} p(\omega|m)\mathrm{d}\omega \tag{21}$$

Note that this is a double integral with $\mathrm{d}\omega$ standing for $\mathrm{d}\widehat{b_0}\mathrm{d}\widehat{b_1}$. Finally, to this $p(\Omega^i|m)$ corresponds a $p(\Omega^i|s)$ because it is assumed that the current controller does not change the estimation of fault parameters.

The conditional observation probabilities are calculated as in [1].

$$p(\omega|m_{nominal}) = \frac{1}{2\pi\sqrt{|P|}}e^{-\frac{1}{2}\widehat{b}^T P^{-1}\widehat{b}} \tag{22}$$

$$P(\omega|m_{locking}) = \frac{1}{4\pi\overline{u}\sqrt{|P|}}\int_{-\overline{u}}^{\overline{u}} e^{-\frac{1}{2}\left(\begin{bmatrix}b_0\\1\end{bmatrix}-\widehat{b}\right)^T P^{-1}\left(\begin{bmatrix}b_0\\1\end{bmatrix}-\widehat{b}\right)}\mathrm{d}b_0 \tag{23}$$

$$P(\omega|m_{loss\,of\,effectiveness}) = \frac{1}{2\pi\sqrt{|P|}}\int_0^1 e^{-\frac{1}{2}\left(\begin{bmatrix}0\\b_1\end{bmatrix}-\widehat{b}\right)^T P^{-1}\left(\begin{bmatrix}0\\b_1\end{bmatrix}-\widehat{b}\right)}\mathrm{d}b_1 \tag{24}$$

where P is the 2×2 matrix at the lower right part of the covariance matrix associated to state and parameter estimation, $P(k+1|k)$.

However, in this article, there is only one faulty mode which gather the locking and the loss of effectiveness case. Assuming the same a priori probability for locking and loss of effectiveness leads to:

$$p(\omega|m_{faulty}) = \frac{1}{2}p(\omega|m_{locking}) + \frac{1}{2}p(\omega|m_{loss\,of\,effectiveness}). \tag{25}$$

4 Reconfiguration Method

4.1 Estimation Method

To estimate the state **x** and the parameters b_0 and b_1, an augmented system is considered, which includes the system natural state -u, v, q, x, z and θ- with Eq. 4, together with the fault parameters. Then a Kalman filter is applied to the resulting augmented system, which may be described by:

$$\begin{aligned} \mathbf{X}_{k+1} &= A^{aug}\mathbf{X}_k + B^{aug}\mathbf{u}_k + \mathbf{W}_k^{aug} \\ \mathbf{y}_k &= C^{aug}\mathbf{X}_k + \mathbf{v}_k \end{aligned} \tag{26}$$

with $A^{aug} = \begin{bmatrix} A & B_{.,2} & D_{.,2}(\mathbf{u}_k) \\ 0_{1,6} & 1 & 0 \\ 0_{1,6} & 0 & 1 \end{bmatrix}$, $B^{aug} = \begin{bmatrix} B \\ 0_{1,3} \\ 0_{1,3} \end{bmatrix}$, $C^{aug} = \begin{bmatrix} I_{6,6} \; 0_{6,1} \; 0_{6,1} \end{bmatrix}$, and $\mathbf{W}_k^{aug} = \begin{bmatrix} \mathbf{w}_k^x \\ \mathbf{w}_k^{b_0} \\ \mathbf{w}_k^{b_1} \end{bmatrix}$

In the estimation algorithm, the covariance $P(k+1|k)$ is reopened when the following test is not passed:

$\forall i \in [1, m]$,

$$(P(k+1|k)_{ii} + R_{ii})^{-1}(y_{k+1}^i - x(k+1|k)_i)^2 \leq (p_\sigma)^2 \tag{27}$$

where p_σ correspond to the required accuracy. Thereafter, $p_\sigma = 5$.

The covariance is reopened as follows:

$$P(k+1|k)_{reset} = \begin{bmatrix} P(k+1|k)_{6,6} & 0_{6,1} & 0_{6,1} \\ 0_{1,6} & \overline{u}/(p_\sigma)^2 & 0 \\ 0_{1,6} & 0 & 1/(p_\sigma)^2 \end{bmatrix} \tag{28}$$

When the test of Eq. 27 is not passed, it suggest a sudden change of the parameters -that is not taken into account in the model-, and thus a switching of mode.

The probabilities of modes are gathered in a vector b. At each step the vector b is predicted:

$$b_{n+1|n} = T_M b_{n|n} \tag{29}$$

Then, the two mode probabilities are updated from the observation ω of the estimated fault parameters defined at Sect. 3.3.4. The Bayes rule is used:

$$b_{n+1|n+1}(m) = b_{n+1|n}(m)\frac{p(\omega|m)}{\sum_{m'} b_{n+1|n}(m')p(\omega|m')}. \tag{30}$$

4.2 Switching Decision Optimization

In the previously defined framework, the goal is to construct a policy $\pi : (E, \Delta) \to A$ (Δ the set of belief state) which minimize a criteria. The objective is to minimize the cost R. The following value function represents the problem to be solved:

$$V_{n+l}(b_{n+l|n}, k) = \min_{a \in A}[r(b_{n+l|n}, k, a) + \gamma \sum_{i=1}^{|\Omega|} p(\Omega^i|b_{n+l|n}) V_{n+l+1}(b_{n+l+1|n}^{\Omega^i}, a)] \tag{31}$$

where $\gamma \in [0, 1[$ is a discount factor, and $\begin{cases} r(b,k,a) = \sum_m b(m) R_n(\{m,k\}, a) \\ p(\Omega^i|b_{n+l|n}) = \sum_{m'} p(\Omega^i|m') b_{n+l+1|n}(m') \\ b_{n+l+1|n}^{\Omega^i}(m) = \frac{p(\Omega^i|m) b_{n+l+1|n}(m)}{p(\Omega^i|b_{n+l|n})} \end{cases}$

The action of the policy is the one ensuring the minimum in Eq. 31.

This problem can be solved with Dynamic Programming over a finite horizon from time $n+l$ to time $n+l+H$. The policy π specifies the action a which maximizes the value function $V_{n+l}(b_{n+l|n}, k)$. To avoid a long computation time, the horizon is $H = 5$.

Equation (31) is solved by:

1. expending from the current belief state the tree of possible b values in function of all possible future actions and discrete observations and,
2. computing backwards the value of V.

In order to compensate for a small value for H, the quantity: $\sum_{n'=n+l+H+1}^{N} R_n(\{m,k\}, a)$ with a constant controller is used for initializing $V_{n+l+H+1}$.

5 Simulations

5.1 Model

For an airspeed of 20 m/s and a slope of −3°, the continuous matrices A_c and B_c are given as follows:

$$A_c = \begin{bmatrix} -0.1817 & 0.6663 & -1.2815 & 0 & 0 & -9.8062 \\ -0.5734 & -5.0995 & 11.2881 & 0 & 0 & -0.2742 \\ 0.4952 & -5.4031 & -5.0905 & 0 & 0 & 0 \\ 0.9996 & 0.0279 & 0 & 0 & 0 & 1.0465 \\ -0.0279 & 0.9996 & 0 & 0 & 0 & -19.9726 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 11.3028 & -0.2944 & -0.2944 \\ 0 & 7.8851 & 7.8851 \\ -4.3890 & 34.2382 & 34.2382 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The system is discretised with a sampling period of 0.1s. The ratio between sampling times $\mu = 10$.

Table 1 Fault scenario

	Time	0 s	10 s	20 s	50 s	70 s
Scenario		Nominal	Locked	Locked	Nominal	Reduced
Fault parameters	b_0	0	−0.2	−0.1	0	0
	b_1	0	1	1	0	0.5

Parameters used in the simulation are as follows. $Q_k^x = diag\{0.01^2, 0.01^2, 0.01^2, 0.01^2, 0.01^2, 0.01^2\}$, $Q_k^b = diag\{0.001^2, 0.001^2\}$, and $R_k = diag\{0.1^2, 0.1^2, (\frac{0.05\pi}{180})^2, 0.1^2, 0.1^2, (\frac{0.05\pi}{180})^2\}$.

Note that the perturbation introduced here does not correspond to a realistic wind model. However, an accurate wind model could be introduced by adding states and considering a Gaussian white noise. The values of R_k are taken following the sensor accuracy.

Only faults of the second actuator are considered:

$$\begin{aligned} e_k^1 &= u_k^1 \\ e_k^2 &= b_{0k} + (1 - b_{1k})u_k^2 - b_{1k}u_0 \\ e_k^3 &= u_k^3 \end{aligned} \tag{32}$$

The mode transition graph is the one of Fig. 5 with:

- $p(m_0|m_0) = e^{-10^{-5}}$
- $p(m_1|m_0) = 1 - p(m_0|m_0)$
- $p(m_0|m_1) = 10^{-4}$
- $p(m_1|m_1) = 1 - p(m_0|m_1)$.

5.2 *Fault Scenario*

To test the design, the scenario include loss of effectiveness case and locking case, as shown in Table 1.

5.3 *Results*

5.3.1 Estimation of State and Fault Parameters

The dashed curves in lower parts of Figs. 7 and 8 presents the state estimation error. These errors are small.

The value of b_0 and b_1 and their estimates are shown in Fig. 9.

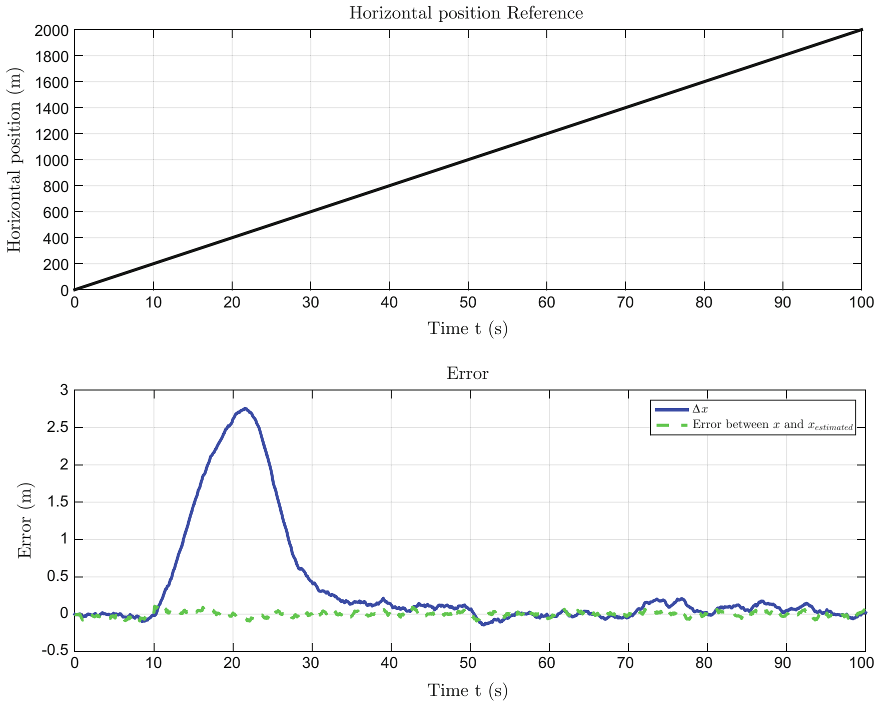


Fig. 7 Horizontal position

The estimator detects the fault in less than 10 s. For each event, the covariance is reopened to detect the fault. For the locking event, it is reopened several times, indicating an adaptation that it is more difficult than for the other events.

5.3.2 Probabilities

The probabilities of the modes are shown in Fig. 10.

The probability of the most likely mode is close to one, except for a short time during the transition. The recovery is more difficult to detect because the covariance is reopened at 52.4 s, indicating that the estimation is not sure.

5.3.3 Action

As shown in Fig. 11, the controller is changed as follow:

- When the locking case is detected, at 11 s, the controller is changed at the next step, at 12 s. Indeed the effectiveness of the nominal controller k_0 in the case of fault is lower than the one of the associated controller k_1.

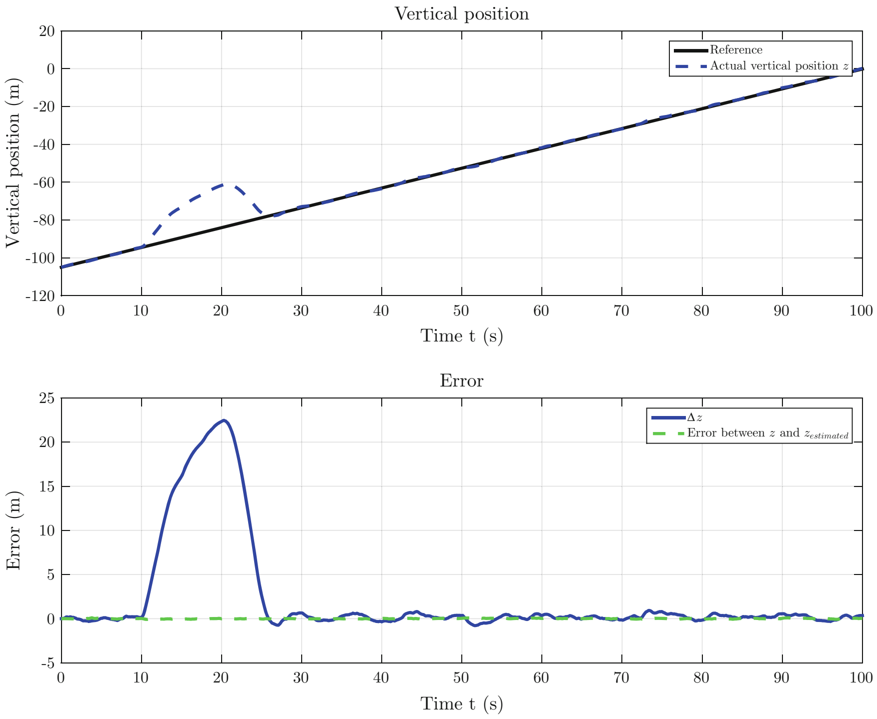


Fig. 8 Vertical position

- When the effectiveness of the actuator is back at 50 s, the controller is not changed from k_1 to k_0 because the expected reward is not large enough compared to the penalty associated to any change of controller. Indeed, as indicated in the upper part of Fig. 6, the difference of performance between k_1 and k_0 is too small.

5.3.4 Control

Figure 12 presents the control for the three actuators. Note that the curves represent the differences between actual inputs and trim inputs.

Between 10 and 50 s, the actuator is locked, first at -0.2 rad and then at -0.1 rad. Between 50 and 70 s, the actuator recovered its performance. After 70 s, the actuator is reduced of 75%.The effective control e^2 is well estimated. When the left elevator is locked at -0.2 rad, the right elevator command is saturated. Thus, the effectiveness of the left elevator can not be entirely compensated and the thrust control is changed.

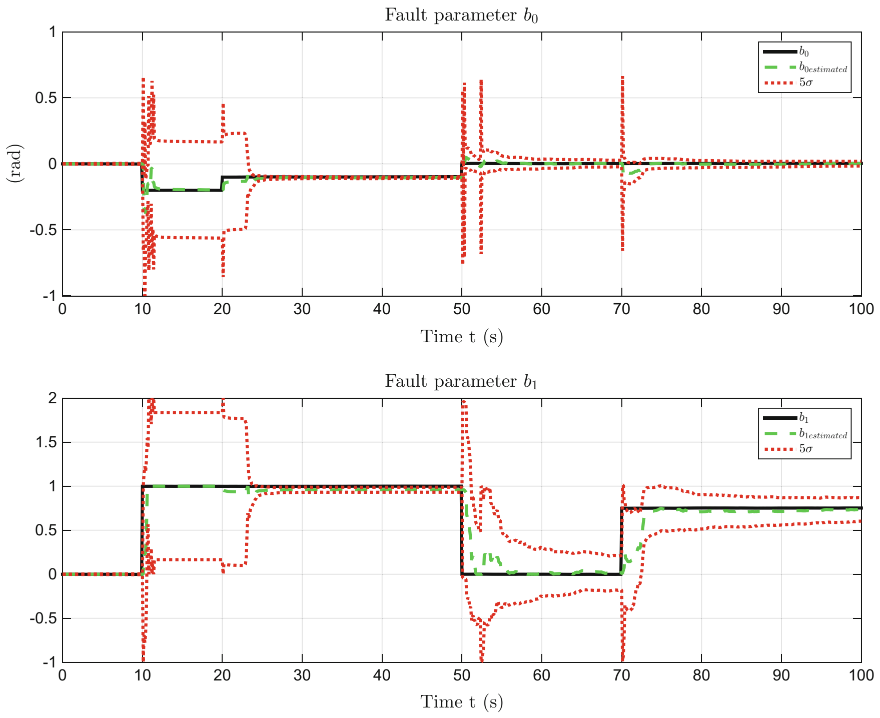


Fig. 9 b_0 and b_1

5.3.5 Tracking the Reference

The plain curves in the lower part of Figs. 7 and 8 present the tracking error for respectively horizontal position and vertical position. The errors are most of the time close to zero, indicating that despite the presence of faults the system state follows the reference leading to an acceptable performance. When the left elevator is locked at -0.2 rad, the tracking error is important. It induces a temporary reference tracking error Δx of about 23 m for altitude and 2.8 m for horizontal position. Indeed, the right elevator command is saturated and then cannot compensate for the faulty left elevator. In this case, despite of a change on thrust control, it is not possible to recover a good tracking.

Finally, with this FTC method, a gain of 25% on the sum of tracking errors is obtained with respect to the use of only the nominal controller. However, for this scenario, the difference between the new FTC method and a FTC based only on the adaptive controller of Eq. 17 is not significant. This is due in part to the fact that for the scenario the noise level on state dynamics and measurement is quite low inducing accurate b_0 and b_1 estimations and in part to the fact that the first fault occurs quite early. However, the mode management of the method allows to take into account

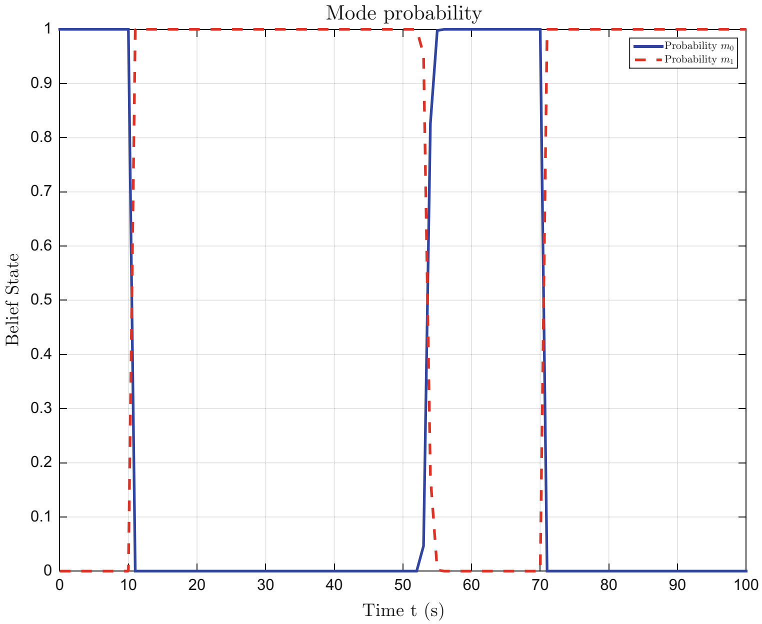


Fig. 10 Probabilities of modes

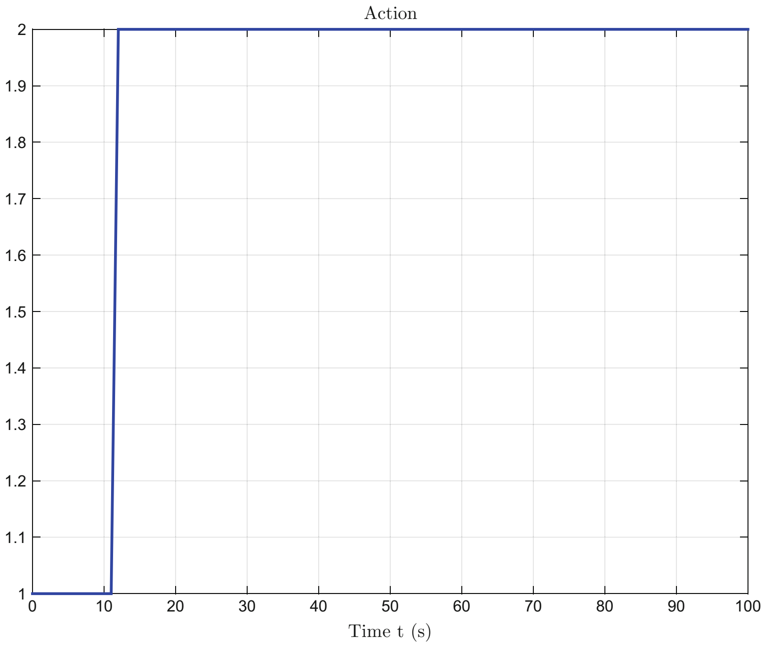


Fig. 11 Action: 1 nominal controller, 2 faulty controller

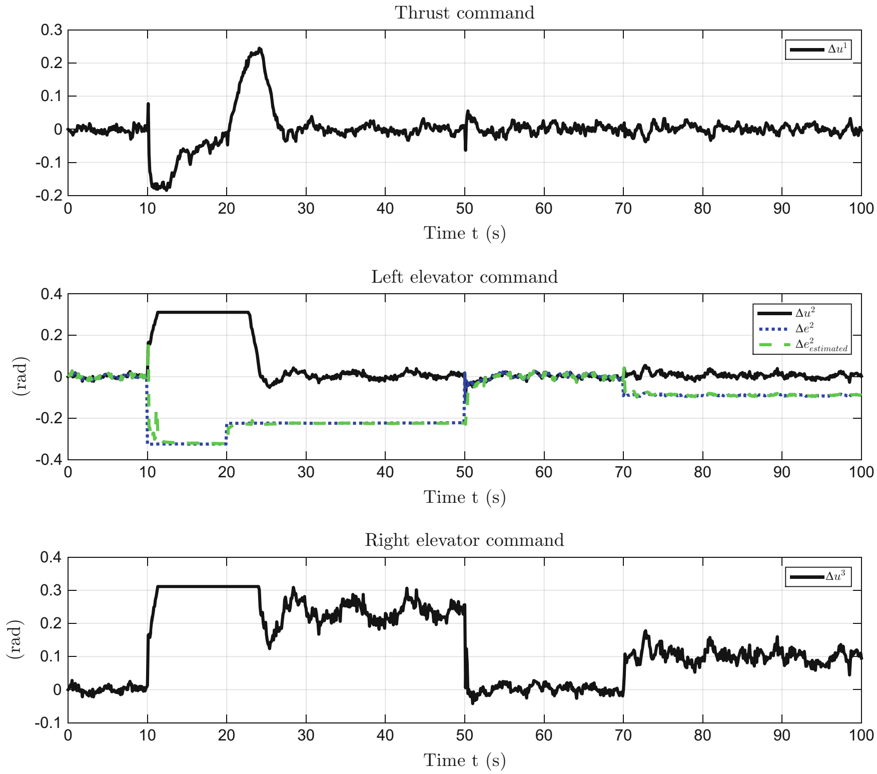


Fig. 12 Command for the three actuators: in the *upper* part, the thrust control input, in the *middle* part, the desired and effective control inputs of the *left* elevator command, in the *lower* part, the control input of the *right* elevator command

possible faults of different devices while it would be difficult to extend the adaptive control in a similar way.

6 Conclusion

In this work, the continuous and discrete framework has been successfully applied to fault tolerance for UAV control. For this, modes are defined and adaptive controllers are developed and qualified in terms of performance for those modes. Simulation of the FTC has been performed with a longitudinal model of Altimum 4 for one fault scenario: a landing approach when an actuator become faulty. It indicates a consistent and effective behaviour and has shown good results; in the event of component faults, the performance is improved. In order to confirm these good results those simplified simulations should be completed by simulations using the non-linear model of the UAV, an accurate wind model and a model of the dynamics of the actuators.

The analysis of the applicability of the continuous and discrete framework for fault tolerant control to UAV induces some improvements to this framework. First, the fault model is extended in order to consider correctly the impact of a fault on a trimmed UAV. Secondly, an analysis is performed in order to define the fault modes and thus, the method for developing adaptive controllers for fault modes is generalized. Finally, different time steps are allowed for the discrete and continuous parts of the fault tolerant control.

The framework used in this paper can be the basis for additional researches. The framework could be applied to both longitudinal and lateral model of UAVs. The addition of multiple actuators faults should be taken into account in future work. Finally, a reconfiguration for sensors faults could be developed based on the framework.

References

1. Boche A, Farges J-L, de Plinval H (2016) A continuous and discrete framework for reconfiguration of control of faulty systems. Accepted at the 2016 IEEE symposium series on computational intelligence (IEEE SSCI 2016)
2. Fang L, Wang JL, Yang G-H (2002). Reliable robust flight tracking control: an LMI approach. IEEE Trans Control Syst Tech 10(1):76–89
3. Hayhurst KJ, Maddalon JM, Miner PS, Szatkowski GN, Ulrey ML, DeWalt MP, Spitzer CR (2007) Preliminary considerations for classifying hazards of unmanned aircraft systems
4. Hwang Inseok, Kim Sungwan, Kim Youdan, Seah Chze Eng (2010) A survey of fault detection, isolation, and reconfiguration methods. IEEE Trans Control Syst Tech 18(3):636–653
5. Lesprier J, Biannic J-M, Roos C (2015) Modeling and robust nonlinear control of a fixed-wing UAV. In 2015 IEEE Conference on Control Applications (CCA), pp 1334–1339
6. Lewis FL, Syrmos VL. Optimal control. Wiley & Sons, Newyork 1995
7. Lunze J, Richter JH (2008). Reconfigurable fault-tolerant control: a tutorial introduction. Eur J Control 14(5):359–386
8. Snooke NA (2015) Automated failure effect analysis for PHM of UAV. In Handbook of unmanned aerial vehicles Springer, Berlin, pp 1027–1051
9. Zhang Youmin, Jiang Jin (2008) Bibliographical review on reconfigurable fault-tolerant control systems. Annu Rev Control 32(2):229–252

Flight Control Law Testing Using Optimal Control and Postoptimal Sensitivity Analysis

Johannes Diepolder, Saurabh Saboo, Venkata Sravan Akkinapalli, Stefan Raab, Jiannan Zhang, Pranav Bhardwaj, Michael Krenmayr, Benedikt Grüter and Florian Holzapfel

1 Introduction

For the clearance of flight control laws, it is required to show that the aircraft cannot be driven to an unsafe states, especially considering inherent modeling uncertainties and disturbances like wind gusts. Current industry practice is to use mostly gridding or monte carlo methods for testing closed loop systems. These methods are straightforward to use, but often computationally expensive, due to the fact that numerous simulations have to be performed to clear the flight control law. Additionally, they do not refine the search space around worst cases, and may miss solutions in between grid points. In essence, these methods evaluate multiple scenarios to show that for all possible combinations of uncertain parameters or maneuvers, safe aircraft operation is ensured. Contrary, optimization based clearance techniques try to find worst cases by means of optimization algorithms. These methods do not investigate the whole solution space for all possible scenarios, but try to find worst cases for which the respective criteria is violated. In the past, considerable effort was made to develop suitable optimization algorithms for this kind of task.

Reference [12] investigates the use of a novel global optimization technique based on dividing rectangles for the nonlinear clearance criteria of a hypersonic re-entry vehicle. The results showed that this approach has potential for significantly enhancing the confidence for the clearance task. In [6], the authors compare probabilistic

J. Diepolder (✉) · S. Saboo · V.S. Akkinapalli · S. Raab · J. Zhang · P. Bhardwaj · M. Krenmayr · B. Grüter · F. Holzapfel
Institute for Flight System Dynamics, TU Munich, Boltzmannstraße 15, 80747 Garching bei München, Germany
e-mail: johannes.diepolder@tum.de

S. Saboo
e-mail: ssaboo@live.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_2

monte carlo methods, and a non-dominated sorting genetic algorithm for multi-criteria optimization for a nonlinear AIRBUS aircraft simulation model.

Reference [8] tests the functionality of a maneuver load limiter using the clonk stability criterion. The authors used a genetic algorithm and adaptive simulated annealing for the worst case parameter search. Reference [13] augments *Differential Evolution* and genetic algorithms by a local optimization method for the clearance task. This results in a hybrid optimization scheme with improved convergence and computational efficiency. The method is illustrated using a six-degree of freedom fighter aircraft model (ADMIRE). In [16] local and global optimization methods are investigated to enhance current industry practice for the clearance task. Especially, the class of evolutionary algorithms like genetic algorithms, evolutionary strategies and *Differential Evolution* showed high potential in finding worst case pilot inputs for criteria such as the maximum angle of attack exceeding (alpha protection) or high velocity/Mach Number (high speed protection). One of the main outcomes from this study is a workflow for the optimization based clearance, which complements existing techniques (Gridding/Monte Carlo) by global and local optimization methods and sensitivity analysis. In this paper, we propose to enhance this workflow by the use of *Optimal Control Theory* and *Postoptimal Sensitivity Analysis*. The paper is organized as follows. First we present the novel methodology and the proposed workflow. Then we give a short introduction to *Differential Evolution*, *Optimal Control Theory* and *Postoptimal Sensitivity Analysis*. To illustrate the methodology, we investigate the height loss for a preliminary control design of a tilt rotor vehicle, from hover-mode to forward flight.

2 Optimal Control Based Clearance of Flight Control Laws

The workflow proposed in [16] for the optimization based clearance of flight control laws consists of four steps. First, a monte carlo analysis is performed to find worst cases for the respective criteria in the flight envelope. These worst cases are then used in a second step to initialize global optimization runs, to explore the search space surrounding those worst cases. The results of global optimization are then refined by local optimization methods, such as the quasi-Newton method in combination with cyclic coordinate descent. The last step consists of performing sensitivity analysis for these refined solutions e.g. by parameter perturbation and re-simulation. Reference [5] motivates the use of optimal control theory for the determination of worst case pilot inputs and wind gusts. The authors investigated an aircraft model, linearized around a horizontal steady state flight condition in the longitudinal and lateral plane. Using a direct collocation scheme implemented in the MATLAB software FALCON.m, the authors were able to obtain meaningful results for different criteria, such as the maximum angle of attack exceeding in the longitudinal plane and the maximum angle of sideslip exceeding in the lateral plane. These results motivate further investigation of using *Optimal Control Theory* for flight control law testing. For the solution of optimal control problems using direct transcription methods (e.g.

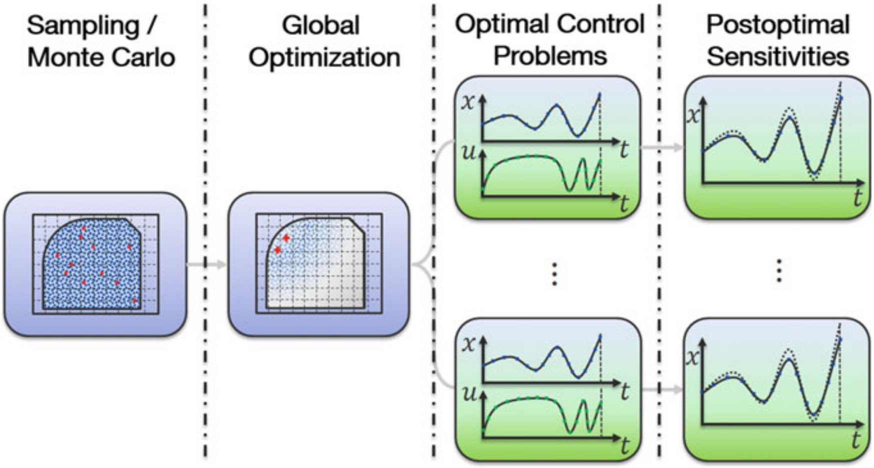


Fig. 1 Novel workflow for optimal control based clearance of flight control laws

shooting or collocation) in combination with gradient based optimization, it is crucial to provide good initial guesses. Therefore, we propose to enhance the workflow from [16] by replacing the last two steps namely local optimization and sensitivity analysis, by *Optimal Control Methods* and *Postoptimal Sensitivity Analysis* (Fig. 1).

3 Theoretical Background

3.1 Global Optimization: Differential Evolution

Differential Evolution (DE) is a global optimization algorithm, which has increasingly gained attention due to its simple structure on one side, and its effectiveness for a wide range of optimization problems on the other [14]. For the current application, we use the classic DE implementation "*DE/rand/1/bin*", for which the base vector is selected *randomly*, *one* weighted difference vector is added to it and the parameter crossover from the mutant vector follows a *binomial* distribution. The basic algorithm is depicted in Algorithm 1. Here, we define the population of generation $g = 1 \ldots g_{max}$ as $\mathbf{P}_z^g$. This population consists of N_p parameter vectors $\mathbf{z}_i^g$ with $i = 0, 1, \ldots, N_p - 1$. Each of the parameter vectors has N_z real valued elements $z_{i,j}^g$ for $j = 0, 1, \ldots, N_z - 1$. Similarly, we define the mutant vector population $\mathbf{P}_m^g$ and the trial vector population $\mathbf{P}_t^g$:

$$\mathbf{P}_z^g = \bigcup_{i \in 0,1,\ldots,N_p-1} \mathbf{z}_i^g, \mathbf{z}_i^g = [z_{i,0}^g, z_{i,1}^g, \ldots, z_{i,N_z-1}^g]^T \tag{1}$$

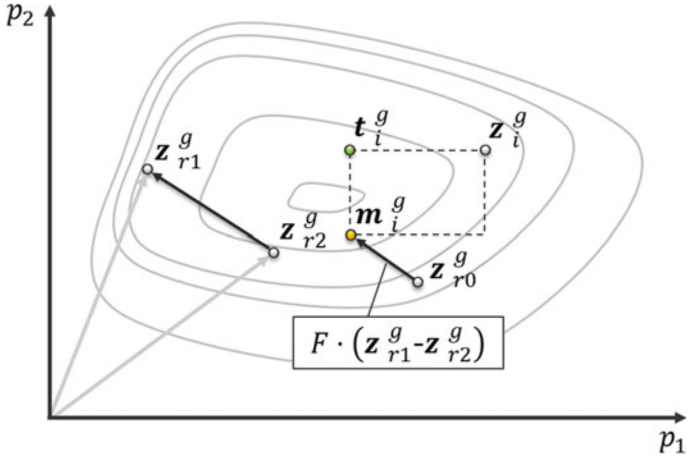


Fig. 2 Illustration of mutation and crossover for differential evolution in a two dimensional parameter space

$$\mathbf{P}_m^g = \bigcup_{i \in 0,1,\ldots,N_p-1} \mathbf{m}_i^g, \mathbf{m}_i^g = [m_{i,0}^g, m_{i,1}^g, \ldots, m_{i,N_z-1}^g]^T \tag{2}$$

$$\mathbf{P}_t^g = \bigcup_{i \in 0,1,\ldots,N_p-1} \mathbf{t}_i^g, \mathbf{t}_i^g = [t_{i,0}^g, t_{i,1}^g, \ldots, t_{i,N_z-1}^g]^T \tag{3}$$

At first, the initial population is generated by randomly assigning values between the upper bounds $\mathbf{z}_{ub}$ and lower bounds $\mathbf{z}_{lb}$:

$$z_{i,j}^0 = rand(0, 1) \cdot (z_{ub,j} - z_{lb,j}) \tag{4}$$

To evolve the existing generation, we first create the mutant population $\mathbf{P}_m^g$. For each mutant vector, we randomly pick three parameter vectors $\mathbf{z}_{r0}^g$, $\mathbf{z}_{r1}^g$ and $\mathbf{z}_{r2}^g$, and add the weighted difference vector of $\mathbf{z}_{r1}^g$ and $\mathbf{z}_{r2}^g$ with weight $F \in [0, 1]$ to the base vector $\mathbf{z}_{r0}^g$:

$$\mathbf{m}_i^g = \mathbf{z}_{r0}^g + F \cdot (\mathbf{z}_{r1}^g - \mathbf{z}_{r2}^g) \tag{5}$$

From the mutant vector $\mathbf{m}_i^g$, the trial vector $\mathbf{t}_i^g$ is constructed via crossover of the elements, with crossover probability $p_c \in [0, 1]$:

$$t_{i,j}^g = \begin{cases} m_{i,j}^g & if(rand(0, 1) \leq p_c) \\ z_{i,j}^g & else \end{cases} \tag{6}$$

Figure 2 illustrates the generation of a trial vector in a two dimensional parameter space. The grey lines in Fig. 2 represent the isolines of the cost function J.

In the selection phase, the trial vectors $\mathbf{t}_i^g$ compete against the target vectors $\mathbf{z}_i^g$ from the current population, by comparing their objective function values $J(\mathbf{t}_i^g)$ and $J(\mathbf{z}_i^g)$. The vectors with the lower cost values win and become part of the next generation:

$$\mathbf{z}_i^{g+1} = \begin{cases} \mathbf{t}_i^g & if\, J(\mathbf{t}_i^g) \leq J(\mathbf{z}_i^g) \\ \mathbf{z}_i^g & else \end{cases} \tag{7}$$

Algorithm 1 Differential Evolution - *DE/rand/1/bin*

```
1: $F \leftarrow [0, 1]$, $p_c \leftarrow [0, 1]$,$g \leftarrow 0$
2: for all $i \in 0, 1, ..., N_p - 1$, $j \in 0, 1, ..., N_z - 1$ do
3:     $z_{i,j}^0 = rand(0, 1) \cdot (z_{ub,j} - z_{lb,j})$
4: end for
5: while $g \leq g_{max}$ do
6:     $g \leftarrow g + 1$
Mutation
7:     for all $i \in 0, 1, ..., N_p - 1$ do
8:         Select $\mathbf{z}_{r0}^g, \mathbf{z}_{r1}^g, \mathbf{z}_{r2}^g$
9:         $\mathbf{m}_i^g = \mathbf{z}_{r0}^g + F \cdot (\mathbf{z}_{r1}^g - \mathbf{z}_{r2}^g)$
10:    end for
Crossover
11:    for all $i \in 0, 1, ..., N_p - 1$, $j \in 0, 1, ..., N_z - 1$ do
12:        $t_{i,j}^g = \begin{cases} m_{i,j}^g & if(rand(0, 1) \leq p_c) \\ z_{i,j}^g & else \end{cases}$
13:    end for
Selection
14:    for all $i \in 0, 1, ..., N_p - 1$ do
15:        $\mathbf{z}_i^{g+1} = \begin{cases} \mathbf{t}_i^g & if\, J(\mathbf{t}_i^g) \leq J(\mathbf{z}_i^g) \\ \mathbf{z}_i^g & else \end{cases}$
16:    end for
17: end while
```

Please note that for the workflow proposed in Sect. 2, we modify the steps 2–4 in Algorithm 1, and use *Latin Hypercube Sampling* (LHS) to initialize the first generation. By doing so, we ensure that the algorithm is “warmstarted” with the worst cases found by LHS, and explore the region around these worst cases using DE.

3.2 Optimal Control Theory

3.2.1 General Problem Statement

Optimal Control Theory [1–3] seeks to solve the following problem: Find the optimal control $\mathbf{u}^*(t)$ which minimizes the Bolza cost function J

$$J = e(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t. \tag{8}$$

subject to the dynamic equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{9}$$

the initial and final boundary conditions at the initial time t_0 and the final time t_f

$$\psi(\mathbf{x}(t_0), \mathbf{x}(t_f)) = \mathbf{0} \tag{10}$$

and the equality and inequality path constraints

$$\mathbf{C}_{P_{eq}}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \tag{11}$$

$$\mathbf{C}_{P_{ineq}}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \tag{12}$$

which are defined on the time interval $t \in [t_0, t_f]$. The general case of the Bolza cost function includes the Mayer cost $e(\mathbf{x}(t_f), t_f)$, which only depends on the state values at the end of the time horizon and the Lagrange cost $L(\mathbf{x}(t), \mathbf{u}(t), t)$, which is integrated from the starting time t_0 to the terminal time t_f. Note, that the above formulation also allows for the introduction of the parameter p by introducing an additional state x_{n+1} with:

$$\dot{x}_{n+1} = 0 \tag{13}$$

and the initial condition

$$x_{n+1}(t_0) = p \tag{14}$$

3.2.2 Solution Strategies

Most of the solution strategies for optimal control problems can be classified into two basic approaches. The first one is named the indirect approach and relies on deriving necessary conditions (*Minimum Principle*) and numerically solving a resulting boundary value problem. To do so, it is useful to introduce the Hamiltonian H by adjoining the right hand side of the dynamic equation (9), to the Lagrange cost function L, using the costates $\boldsymbol{\lambda}(t)$ and the multiplier l_0:

$$H = \boldsymbol{\lambda}(t) \cdot \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + l_0 \cdot L(\mathbf{x}(t), \mathbf{u}(t), t) \tag{15}$$

The *Minimum Principle* states that for the optimal control u^*, the Hamiltonian H has to be stationary:

$$H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}^*(t), t) = \min_{\mathbf{u}} H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}(t), t) \tag{16}$$

and additionally the Euler Lagrange Equation:

$$\dot{\boldsymbol{\lambda}}^T(t) = -\nabla_{\mathbf{x}} H(\mathbf{x}^*(t), \boldsymbol{\lambda(t)}, \mathbf{u}^*(t), t) \tag{17}$$

as well as the transversality conditions

$$\boldsymbol{\lambda}(t_0) = -\nabla_{\mathbf{x}_0}[l_0 \cdot e(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f)) + \boldsymbol{\sigma}^T \cdot \psi(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f))]^T \tag{18}$$

$$\boldsymbol{\lambda}(t_f) = -\nabla_{\mathbf{x}_f}[l_0 \cdot e(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f)) + \boldsymbol{\sigma}^T \cdot \psi(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f))]^T \tag{19}$$

have to be satisfied for $(l_0, \boldsymbol{\lambda}^T, \boldsymbol{\sigma}^T) \neq \mathbf{0}$. The solution of the necessary conditions usually requires the knowledge of the switching structure of the problem, and in most real world applications with complex dynamics or constraints, the solution using indirect methods is very difficult and often only possible for heavily simplified models.

The second approach for solving optimal control problems is called the direct approach and transcribes the infinite dimensional optimal control problem into a finite dimensional parameter optimization problem. There exist various methods for the transcription e.g. single shooting, multiple shooting or collocation. All of these methods discretize the dynamic equation in time, and introduce additional equality constraints, which ensure that the dynamic equation is satisfied. The main difference between these transcription schemes is how the dynamics are discretized on the time grid $\mathbf{t} = [t_0, t_1 ... t_f]$.

In case of single shooting using a Euler discretization scheme

$$\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i) \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i) \tag{20}$$

we only introduce the initial state $\mathbf{x}_0$ and the controls $\mathbf{u}_{0...f}$ additional to the n_p parameters $\mathbf{p}_{1...n_p}$ to obtain the optimization parameter vector $\mathbf{z}_{SS}$:

$$\mathbf{z}_{SS} = [\mathbf{p}_{1...n_p}, \mathbf{x}_0, \mathbf{u}_{0...f}]^T \tag{21}$$

In order to improve numerical stability it is often necessary to introduce n_s additional shooting-nodes $\mathbf{x}_{s_{0...n_s-1}}$ at distinct time points t_i at which the dynamic equality constraints

$$\mathbf{C}_{dyn,j} = \mathbf{x}_{s_j} - \mathbf{x}(t_i) = 0, \, j = 0...n_s - 1 \tag{22}$$

have to be fulfilled. This method is called multiple shooting and yields the following parameter vector:

$$\mathbf{z}_{MS} = [\mathbf{p}_{1...n_p}, \mathbf{x}_{s_{0...n_s-1}}, \mathbf{u}_{0...f}]^T \tag{23}$$

A third option is to use collocation methods e.g. the trapezoidal collocation scheme for which we introduce one state and control per time discretization point. The dynamic constraints $\mathbf{C}_{dyn,i}, i = 0...f - 1$ in this case become:

$$\mathbf{C}_{dyn,i} = \mathbf{x}_{i+1} - \mathbf{x}_i - \frac{1}{2} \cdot (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i) + \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, t_{i+1})) \cdot (t_{i+1} - t_i) \tag{24}$$

and the parameter vector contains the parameters $p_{1...n_p}$ and the fully discretized states and controls $\mathbf{x}_{0...f}, \mathbf{u}_{0...f}$:

$$\mathbf{z}_{Col} = [\mathbf{p}_{1...n_p}, \mathbf{x}_{0...f}, \mathbf{u}_{0...f}]^T \tag{25}$$

In case of collocation methods the resulting optimization problem exhibits a strongly decoupled and hence sparse structure at the cost of a greatly increased number of optimization variables and constraints.

Additional to the dynamic constraints $\mathbf{C}_{dyn}$ the equality and inequality constraints (11) and (12) are discretized on the time grid $\mathbf{t}$:

$$\mathbf{C}_{P_{eq},i}(x_i, u_i) = 0 \tag{26}$$

$$\mathbf{C}_{P_{ineq},i}(x_i, u_i) \leq 0 \tag{27}$$

The residual vector $\mathbf{F}$ holds the cost function (8), the dynamic constraints (22) or (24) and the discretized inequality and equality constraints (26), (27) to be satisfied:

$$\mathbf{F} = [\psi(\mathbf{x}(t_0), \mathbf{x}(t_f)), \mathbf{C}_{P_{eq},0...f}, \mathbf{C}_{P_{ineq},0...f}, \mathbf{C}_{dyn,0...n_s-1/f-1}]^T \tag{28}$$

Both, the parameter vector $\mathbf{z}$ and the residual vector $\mathbf{F}$ are required to satisfy box constraints with the respective upper and lower bounds $\mathbf{z}_{ub}$, $\mathbf{z}_{lb}$,$\mathbf{F}_{ub}$, $\mathbf{F}_{lb}$:

$$\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub} \tag{29}$$

$$\mathbf{F}_{lb} \leq \mathbf{F} \leq \mathbf{F}_{ub} \tag{30}$$

3.3 Post-optimal Sensitivity Analysis

Postoptimal Sensitivity Analysis [4, 7] provides the sensitivities of the optimal solution with respect to parameters $\boldsymbol{\omega}$. The derivation of the local sensitivities is based on the Fiacco equation, which is obtained by applying the implicit function theorem to the Karush–Kuhn–Tucker (KKT) conditions. The resulting sensitivity equation for the parameter vector $\mathbf{z}$, and the Lagrange-multipliers $\boldsymbol{\lambda}$, for the constraints reads:

$$\begin{pmatrix} \frac{d\mathbf{z}}{d\boldsymbol{\omega}} \\ \frac{d\boldsymbol{\lambda}}{d\boldsymbol{\omega}} \end{pmatrix} = \begin{bmatrix} \nabla_{\mathbf{zz}}\mathscr{L} & \nabla_{\mathbf{z}}\mathbf{C}_a{}^T \\ \nabla_{\mathbf{z}}\mathbf{C}_a & \mathbf{0} \end{bmatrix}^{-1} \cdot \begin{pmatrix} \nabla_{\mathbf{z}\boldsymbol{\omega}}\mathscr{L} \\ \nabla_{\boldsymbol{\omega}}\mathbf{C}_a \end{pmatrix} \tag{31}$$

In (31), $\mathbf{C}_a$ are the active equality constraints and $\mathscr{L}$ is the Lagrangian defined as:

$$\mathscr{L}(\mathbf{z}, \boldsymbol{\lambda}; \boldsymbol{\omega}) = J(\mathbf{z}; \boldsymbol{\omega}) + \boldsymbol{\lambda}^T \cdot \mathbf{C}_a(\mathbf{z}; \boldsymbol{\omega}) \tag{32}$$

This equation provides a first order approximation for perturbations $\Delta\boldsymbol{\omega}$, to obtain updates for the parameter vector and the Lagrange multipliers:

$$\hat{\mathbf{z}} = \mathbf{z} + \frac{d\mathbf{z}}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} \tag{33}$$

$$\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda} + \frac{d\boldsymbol{\lambda}}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} \tag{34}$$

Furthermore, the first and second order sensitivities of the objective function can be computed as follows [4]:

$$\frac{dJ}{d\boldsymbol{\omega}} = \nabla_{\mathbf{z}} J \frac{d\mathbf{z}}{d\boldsymbol{\omega}} + \nabla_{\boldsymbol{\omega}} J \tag{35}$$

$$\frac{d^2 J}{d\boldsymbol{\omega}^2} = \left(\frac{d\mathbf{z}}{d\boldsymbol{\omega}}\right)^T \nabla_{\mathbf{zz}}\mathscr{L} \frac{d\mathbf{z}}{d\boldsymbol{\omega}} + 2\left(\nabla_{\boldsymbol{\omega}\mathbf{z}}\mathscr{L} \frac{d\mathbf{z}}{d\boldsymbol{\omega}}\right)^T + \nabla_{\boldsymbol{\omega}\boldsymbol{\omega}}\mathscr{L} \tag{36}$$

These sensitivities are used to study the effect of parameters on the objective function of the optimization problem. In particular, they provide sensitivities of second order for the worst cases, found by solving the optimal control problem for the respective clearance criteria.

4 Case Study: Testing of the Preliminary Design of a Tilt-Rotor

4.1 Plant - Model Equations

For the case study a 2-D model of a vertical take-off and landing vehicle with tilt rotor mechanism is considered. The reduced model has three degrees of freedom where the translational motion is described in x_B and z_B direction and the rotation is considered around the pitch axis y_B. This aerial vehicle possesses seven control effectors: two rotors on the fuselage, two rotors on the wings with tilt mechanisms and symmetrical elevators at the tail. The propulsion, gravitational and aerodynamic forces acting on the airframe are described in the following.

4.1.1 Propulsion Forces and Moment

Thrust and drag forces from the propeller are taken as follows:

$$T = C_T \omega^2 \rho \frac{d^4 \pi^2}{4} \tag{37}$$

$$D = C_H \omega^2 \rho \frac{d^4 \pi^2}{4} V_A^R \tag{38}$$

Here V_A^R is the aerodynamic velocity at the aircraft reference point R, and C_H and C_T are drag and thrust constants of the propeller, ω is the motor rotational speed, d is the diameter of the rotor and ρ is the atmospheric density. The resulting propulsion force $(\mathbf{F}_P^P)_P$ at the propulsion point P in the propulsion frame P reads:

$$(\mathbf{F}_P^P)_P = \begin{bmatrix} -D \\ -T \end{bmatrix}_P \tag{39}$$

Using the tilt rotor angle δ, the transformation $\mathbf{M}_{PB}$ from body frame B to propulsion frame P is defined as follows:

$$\mathbf{M}_{PB} = \begin{bmatrix} cos(\delta) & sin(\delta) \\ -sin(\delta) & cos(\delta) \end{bmatrix} \tag{40}$$

Note, that the tilt angle δ is zero for rotors on fuselage. Using this transformation we obtain the propulsion forces and moments at the reference point R in the body fixed frame B:

$$(\mathbf{F}_P^R)_B = \mathbf{M}_{PB}^T \cdot (\mathbf{F}_P^P)_P \tag{41}$$

$$(\mathbf{M}_P^R)_B = \mathbf{r}^{RP} \times (\mathbf{F}_P^P)_B \tag{42}$$

4.1.2 Gravitational Forces and Moments

The gravitational forces and moments resulting from aircraft mass m and acceleration g is transformed from the North-East-Down frame O to the body fixed frame B using the Euler angle θ around the pitch axis:

$$(\mathbf{F}_G^R)_B = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ m \cdot g \end{bmatrix}_O \tag{43}$$

$$(\mathbf{M}_G^R)_B = \mathbf{r}^{RG} \times (\mathbf{F}_G^R)_B \tag{44}$$

Here $\mathbf{r}^{RG}$ is the distance from the reference point R to the center of gravity G.

4.1.3 Aerodynamic Forces and Moments

For the aerodynamic forces and moments, the wing and tail section are considered separately. The aerodynamic angle of attack for the wing α_w is determined as follows:

$$\alpha_w = \text{atan2}\left(\frac{(w_A^G)_B^E - (\omega_y^{IB})_B \cdot (x^{RW})_B}{(u_A^G)_B^E + (\omega_y^{IB})_B \cdot (z^{RW})_B} \right) + \alpha_0 + i \tag{45}$$

Here $(u_A^G)_B^E$ and $(w_A^G)_B^E$ are aerodynamic velocities at the center of gravity G in x_B and z_B direction of the body frame B. $(\omega_y^{IB})_B$ is the angular velocity around the pitch axis, $(x^{RW})_B$ and $(z^{RW})_B$ are the distances from the wing reference point W to the reference point R, α_0 is the angle of attack at zero velocities and i is the inclination angle. The transformation matrix from body axis to wing axis is determined as:

$$\mathbf{M}_{WB} = \begin{bmatrix} \cos(\alpha_w) & -\sin(\alpha_w) \\ \sin(\alpha_w) & \cos(\alpha_w) \end{bmatrix} \tag{46}$$

The aerodynamic forces for the wing $(\mathbf{F}_W^W)_W$ originating from the lift and drag forces L_W and D_W at the wing reference point W in wing frame W are calculated as follows:

$$L_W = \frac{1}{2} \cdot \rho V_A^{R^2} S_W C_{L\alpha,W} \alpha_w \tag{47}$$

$$D_W = \frac{1}{2} \cdot \rho V_A^{R^2} S_W C_{D\alpha,W} \alpha_w \tag{48}$$

$$(\mathbf{F}_W^W)_W = \begin{bmatrix} -D_W \\ -L_W \end{bmatrix}_W \tag{49}$$

Here $C_{L\alpha,W}$ and $C_{D\alpha,W}$ are the wing aerodynamic coefficients, α_W is the angle of attack of the wing and S_W is the wing reference area. The aerodynamic forces and moments from the wing at the reference point R denoted in the body frame B are determined as:

$$(\mathbf{F}_W^R)_B = \mathbf{M}_{WB}^T \cdot (\mathbf{F}_W^W)_W \tag{50}$$

$$(\mathbf{M}_W^R)_B = \mathbf{r}^{RW} \times (\mathbf{F}_W^W)_B \tag{51}$$

with the distance from the reference point R to the wing reference point W being $\mathbf{r}^{RW}$. The whole tail section of the aircraft is allowed to move, and hence acts as elevators. Similar to (45) the angle of attack of the tail α_t is calculated as follows:

$$\alpha_t = \operatorname{atan2}\left(\frac{(w_A^R)_B^E - (\omega_y^{IB})_B \cdot (x^{Rt})_B}{(u_A^R)_B^E + (\omega_y^{IB})_B \cdot (z^{Rt})_B}\right) \tag{52}$$

and the transformation matrix from body axis B to tail axis t is determined as

$$\mathbf{M}_{tB} = \begin{bmatrix} \cos(\alpha_t) & -\sin(\alpha_t) \\ \sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix} \tag{53}$$

The aerodynamic forces $(\mathbf{F}_t^t)_t$ for the tail at the tail reference point t in tail frame t are computed from the lift and drag forces L_t and D_t using the aerodynamic coefficients $C_{L\alpha,t}$ and $C_{D\alpha,t}$, the reference area S_t, the angle of attack at the tail α_t and the elevator deflection η:

$$L_t = \frac{1}{2} \cdot \rho V_A^{R^2} S C_{L\alpha,t}(\alpha_t + \eta) \tag{54}$$

$$D_t = \frac{1}{2} \cdot \rho V_A^{R^2} S C_{D\alpha,t}(\alpha_t + \eta) \tag{55}$$

$$(\mathbf{F}_t^t)_t = \begin{bmatrix} -D_t \\ -L_t \end{bmatrix}_t \tag{56}$$

The aerodynamic forces and moments from the tail t at reference point R in body frame B can then be determined as:

$$(\mathbf{F}_t^R)_B = \mathbf{M}_{tB}^T \cdot (\mathbf{F}_t^t)_t \tag{57}$$

$$(\mathbf{M}_t^R)_B = \mathbf{r}^{Rt} \times (\mathbf{F}_t^t)_B \tag{58}$$

with the distance from the reference point R to the tail reference point r being $\mathbf{r}^{Rt}$.

4.1.4 Total Forces and Moment

The forces and moments are summed together to obtain the total forces $(\mathbf{F}_T^R)_B$ and the moment $(M_T^R)_B$ in the body frame B and acting on the airframe at reference point R:

$$(\mathbf{F}_T^R)_B = (\mathbf{F}_P^R)_B + (\mathbf{F}_G^R)_B + (\mathbf{F}_W^R)_B + (\mathbf{F}_t^R)_B \tag{59}$$

$$(M_T^R)_B = (M_P^R)_B + (M_G^R)_B + (M_W^R)_B + (M_t^R)_B \tag{60}$$

4.1.5 Equations of Motion

The state derivatives are finally determined from following equations of motions with the aircraft mass m and the moment of inertia around the pitch axis I_{yy}:

- Position:

$$\begin{bmatrix} \dot{x} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ sin(\theta) & -cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} (u_K^R)_B^E \\ (w_K^R)_B^E \end{bmatrix} \tag{61}$$

- Attitude:

$$\dot{\theta} = (\omega_y^{IB})_B^B \tag{62}$$

- Translation:

$$\begin{bmatrix} (\dot{u}_K^R)_B^{EB} \\ (\dot{w}_K^R)_B^{EB} \end{bmatrix} = \begin{bmatrix} -(\omega_y^{IB})_B \cdot (w_K^R)_B^{EB} \\ (\omega_y^{IB})_B \cdot (u_K^R)_B^{EB} \end{bmatrix} + \frac{(\mathbf{F}_T^R)_B}{m} \tag{63}$$

- Rotation:

$$(\dot{\omega}_y^{IB})_B^B = (I_{yy})^{-1} (M_T^R)_B \tag{64}$$

Furthermore we consider PT2 transfer functions between the commanded values δ_c and the effector reactions δ

$$G(s) = \frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{65}$$

This is done for each of the control effectors, namely the elevator η, the tilt angles for the rotors at the wing tips δ_1 and δ_2, the two propellers at the left and right wing tips ω_1 and ω_2 and the propellers at the tail section ω_3 and ω_4. In order to introduce a smooth wind disturbance we additionally introduce a first order filter

$$(\dot{\mathbf{V}}_W^R)_B^{EB} = \frac{(\mathbf{V}_W^R)_{B,cmd}^E - (\mathbf{V}_W^R)_B^E}{\tau_w} \tag{66}$$

for the wind components of $(\mathbf{V}_W^R)_B^E$: the velocities $(u_W^R)_B^E$ and $(w_W^R)_B^E$ relative to the earth E, acting on the reference point R and denoted in the body fixed frame B.

The resulting state space model for the plant thus consists of 22 states comprised of the rigid body equations of motions, the second order actuator transfer functions and the first order filters for the wind disturbance.

4.2 Incremental NDI Control Structure

4.2.1 Governing Equations

The baseline control law implements the incremental nonlinear dynamic inversion technique [15]. Consider an aircraft dynamic model,

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u}) \tag{67}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{68}$$

using Taylor expansion,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{69}$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \left.\frac{\partial [\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u})]}{\partial \mathbf{x}}\right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{70}$$

and differentiating equation (68):

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} \tag{71}$$

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}}_0 + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left.\frac{\partial [\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u})]}{\partial \mathbf{x}}\right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{72}$$

yields:

$$\dot{\mathbf{y}} - \dot{\mathbf{y}}_0 = \delta\dot{\mathbf{y}} = \mathbf{A}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{x} + \mathbf{B}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{u} \tag{73}$$

The control effectors directly controls accelerations, while the states are integrations of accelerations. For high control frequencies, the increments of states are much smaller than the increments of inputs. Therefore, Eq. (73) can be further simplified by eliminating the term $\mathbf{A}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{x}$:

$$\delta\dot{\mathbf{y}} = \mathbf{B}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{u} \tag{74}$$

4.2.2 Controller Structure

The pilot commands velocities in x_O and z_O direction of the NED frame, and the pitch angle θ. A linear reference model is designed to force the controlled variables to follow a reference trajectory. A relative degree 1 reference model is implemented for the velocities, and a relative degree 2 reference model is used for the pitch angle. Furthermore error controllers are implemented to account for model uncertainties. The reference model and error controller states add six additional states to the 22 plant states, which results in a total number of 28 states for the closed loop system. The on-board plant model, which is used to determine the control effectiveness B consists of the complete set of aerodynamic, propulsion and the gravitation model from Sect. 4.1. Current state accelerations are estimated using equations of motion in this on-board plant model and the estimated reaction $\boldsymbol{\nu}_{est}$ and the control effectiveness matrix $\mathbf{B}$ are given as outputs. The desired pseudo-command $\Delta\boldsymbol{\nu}_{des}$ is then calculated as

$$\Delta\boldsymbol{\nu}_{des} = \boldsymbol{\nu}_{ref} - \boldsymbol{\nu}_{est} \tag{75}$$

In this preliminary case study the required control effector deflection $\Delta\mathbf{u}_{cmd}$ is determined using weighted-pseudo inverse with weighting matrix Λ_u:

$$\Delta\mathbf{u}_{cmd} = \Lambda_u(\mathbf{B}\Lambda_u)^{+}\Delta\boldsymbol{\nu}_{des} \tag{76}$$

Finally $\Delta\mathbf{u}_{cmd}$ is added to $\mathbf{u}_{i-1}$ from the previous time step, and subsequently fed to the plant.

4.3 Problem Formulation and Results

4.3.1 General Problem Formulation

The problem we solve using the proposed workflow from Sect. 2, is to test for height loss during the transition phase from hovering mode to forward flight. The cost function J to be minimized is the maximum height loss over the time horizon $t \in [t_0, t_f]$:

$$\min_{\mathbf{z}} J(\mathbf{z}) = \min_{\mathbf{z}} \min_{t\in[t_0,t_f]} h(t) \tag{77}$$

The simulation is performed by Euler's method using a step size of $\Delta t = 0.001$ s with an initial time of $t_f = 0$ s and a final time of $t_f = 10$ s. This time span is sufficient to allow the vehicle to perform the complete transition. Additionally, we set the initial condition for hover flight with zero velocities in 10 m height. Throughout the optimization, the commanded velocity u_{cmd} is $5\frac{\mathrm{m}}{\mathrm{s}}$ and the other control inputs are zero:

$$\begin{pmatrix} u_{cmd} \\ w_{cmd} \\ \theta_{cmd} \end{pmatrix} = \begin{pmatrix} 5\frac{\mathrm{m}}{\mathrm{s}} \\ 0 \\ 0 \end{pmatrix} \tag{78}$$

4.3.2 Step 1: Latin Hypercube Sampling

In the first step, *Latin Hypercube Sampling* is performed with 250 samples using the MATLAB function *lhsdesign()*. For the wind gust parametrization we introduce the timepoints for the wind commands T_u and T_w and the command values for the wind gusts $u_{W,cmd}$ and $w_{W,cmd}$

$$t_0 \leq T_u \leq t_f \tag{79}$$

$$t_0 \leq T_w \leq t_f \tag{80}$$

$$u_{W,cmd,lb} \leq u_{W,cmd} \leq u_{W,cmd,ub} \tag{81}$$

$$w_{W,cmd,lb} \leq w_{W,cmd} \leq w_{W,cmd,ub} \tag{82}$$

with $u_{W,cmd,lb} = w_{W,cmd,lb} = -2\frac{\mathrm{m}}{\mathrm{s}}$ and $u_{W,cmd,ub} = w_{W,cmd,ub} = +2\frac{\mathrm{m}}{\mathrm{s}}$ and the time constant $\tau_W = 3\,\mathrm{s}$ for the filter in (66).

Additionally, we add two parameters for the uncertainties in mass Δm and moment of inertia ΔI_{yy}:

$$\Delta m_{min} \leq \Delta m \leq \Delta m_{max} \tag{83}$$

$$\Delta I_{yy_{min}} \leq \Delta I_{yy} \leq \Delta I_{yy_{max}} \tag{84}$$

These parameters have the bounds $\Delta m_{min} = \Delta I_{yy_{min}} = 90\%$ and $\Delta m_{max} = \Delta I_{yy_{max}} = 110\%$ and are used to pertube the nominal values $I_{yy_{nom}}$ and m_{nom} as follows:

$$m = \Delta m \cdot m_{nom} \tag{85}$$

$$I_{yy} = \Delta I_{yy} \cdot I_{yy_{nom}} \tag{86}$$

Please note, that the bounds of the wind disturbance and parameter uncertainties are chosen very carefully here as the control structure is still under development.

4.3.3 Step 2: Differential Evolution

The following global optimization is initialized by the 25 worst cases found by LHS. In case of *Differential Evolution*, we now use five parameters for each wind gust command $u_{cmd,1,\ldots5}$, $w_{cmd,1,\ldots5}$ on a fixed equidistant grid for the control inputs. To translate the step inputs from LHS with step times T_u and T_w on the DE grid, we take the closest value on the equidistant time grid (see Fig. 3).

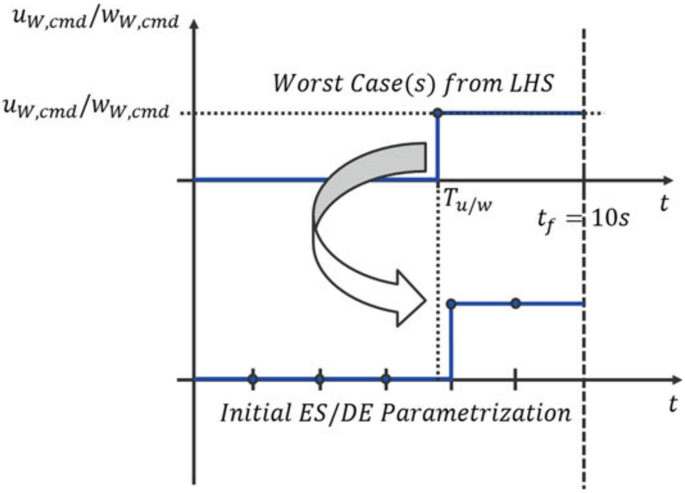


Fig. 3 Translation of the worst case wind command input from LHS to DE

For the classic DE algorithm "*DE/rand/1/bin*" implemented in the solver *DeMat* [14], we set the population size to $N_p = 25$, the weighting factor to $F = 0.8$, the crossover probability to $p_c = 0.7$ and the maximum number of generations to $g_{max} = 50$.

4.3.4 Step 3: Optimal Control

The initial guess for the subsequent *Optimal Control* (OC) problem is the best solution from the DE optimization. In the example presented here the minimum height found from global optimization was at the final time t_f. The cost function is now altered to maximize the height loss at this particular time point.

Now we refine the discretization even further using 40 discretization points for each of the wind gust commands u_{cmd} and w_{cmd}. For this particular problem we found it sufficient to use single shooting due to the short time frame of the optimization. For other problems, especially those who exhibit highly sensitive parameters or long time intervals it may be required to introduce additional shooting nodes to lower the sensitivities and thus enhance convergence.

The solver used here is the SQP-solver SNOPT [9] which requires the gradient of the problem. We compute the gradient by the complex step method as described in [11]. This method exploits the fact that using a complex step $x + ih$ for a function $f(z)$ with sufficiently small step size h we can approximate the derivative of this function up to machine precision by dividing the imaginary part of $f(z + ih)$ and by h. For the scalar case the equation reads:

Table 1 Summary of worst cases

Step	Height loss (m)	Δm (%)	ΔI_{yy} (%)
Step 1: LHS	0.42	106.45	93.47
Step 2: DE	0.46	109.80	96.56
Step 3: OC	0.58	110.00	90.00

$$\frac{\partial f}{\partial z} \approx \frac{\Im[f(z+ih)]}{h} \tag{87}$$

On the one hand, the main drawbacks of this method are the fact that the complex evaluation is computationally more expensive and that some of the functions, such as $<$, $>$, *min()*, *max()* and *abs()* to name a few have to be redefined in order to evaluate the model properly. On the other hand the derivatives are extremely accurate (in the order of the machine precision) which reduces the "numerical noise" introduced from finite difference approximations. Additionally, in MATLAB and SIMULINK most of the built-in functions can be used both for complex and real algebra such that the additional modeling requirements are greatly reduced.

4.3.5 Step 4: Postoptimal Sensitivities

For the *Postoptimal Sensitivities* we need to compute the Hessian of the Lagrangian (see (31)) with respect to the optimal parameter vector $\mathbf{z}_{opt}$. Similarly we again use the complex step method from [10] to compute the second derivatives.

The first and second order sensitivities $\frac{dJ}{d\boldsymbol{\omega}}$ and $\frac{d^2J}{d\boldsymbol{\omega}^2}$ of the cost function (77) w.r.t. $\boldsymbol{\omega} = [\Delta m, \Delta I_{yy}]^T$ are computed using (35) and (36) which allows us to approximate the change in the cost function ΔJ (worst case) when changing the parameters by $\Delta\boldsymbol{\omega}$:

$$\Delta J = \frac{dJ}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} + \Delta\boldsymbol{\omega}^T \cdot \frac{d^2J}{d\boldsymbol{\omega}^2} \cdot \Delta\boldsymbol{\omega} \tag{88}$$

4.3.6 Results

The comparison between the baseline height history without uncertainties and the height trajectories for the different steps (LHS, DE, OC) for the transition phase is depicted in Fig. 4.

The worst case height loss and the corresponding parameter combinations are summarized in Table 1.

The *Postoptimal Sensitivities* for $\boldsymbol{\omega} = [\Delta m, \Delta I_{yy}]^T$ in this example revealed that for the worst case the uncertainty in the mass Δm has significant influence on the optimal solution, but the uncertainty in the moment of inertia ΔI_{yy} is almost neglectable.

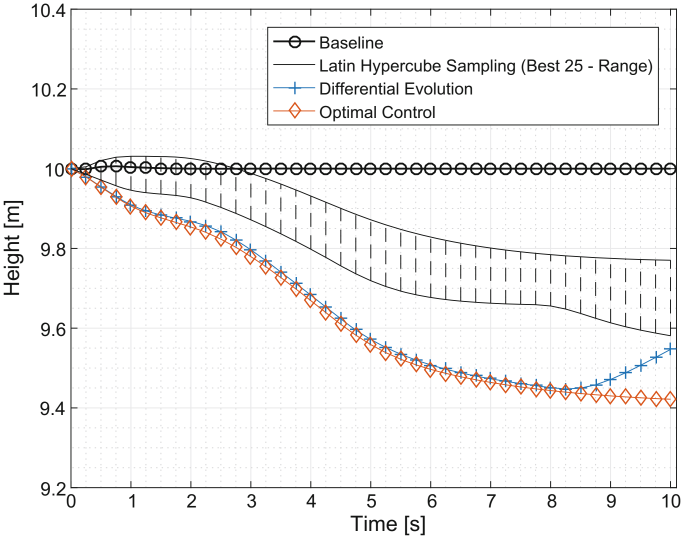


Fig. 4 Comparison of the height loss during transition from hover-mode to forward-flight

$$\frac{dJ}{d\boldsymbol{\omega}} = [-0.98, 6.52 \cdot 10^{-6}]^T \tag{89}$$

$$\frac{d^2J}{d\boldsymbol{\omega}^2} = \begin{bmatrix} -0.34 & -2.43 \cdot 10^{-5} \\ -2.43 \cdot 10^{-5} & -4.76 \cdot 10^{-7} \end{bmatrix} \tag{90}$$

This can be explained by the fact that the influence of I_{yy} is mostly seen in dynamic maneuvers where the attitude of the vehicle changes. In the example presented here, the controller tries to keep the attitude angle at zero as required by $\theta_{cmd} = 0$ and only accelerates by tilting the rotors at the wings forward. This means that for the most part of the transition phase the attitude remains approximately constant zero which results in an acceleration maneuver without attitude change. For this maneuver I_{yy} has almost no influence whereas the uncertainty in mass significantly influences the acceleration and height loss for the worst case found.

Furthermore, as the parameters are subject to optimization we would expect a zero first order sensitivity $\frac{dJ}{d\boldsymbol{\omega}}$ at the optimal point, which is not necessarily the case for the second order sensitivity $\frac{d^2J}{d\boldsymbol{\omega}^2}$. In this specific case study the parameter values converged to the boundary of the admissible parameter set. Therefore, the first order sensitivity $\frac{dJ}{d\boldsymbol{\omega}}$ is non-zero and provides the sensitivity of the cost function value with respect to a change in the boundary value.

5 Conclusions

In the present paper, we investigated a novel workflow for the clearance of flight control laws with continuous control or disturbance inputs using *Optimal Control* Theory and *Postoptimal Sensitivity Analysis*. For this methodology, sampling and a global optimization method are used to obtain good initial guesses for the application of direct *Optimal Control* Methods. Additionally, first and second order approximations for the objective function (clearance criteria) were revised which can be derived from *Postoptimal Sensitivity Analysis*. The workflow is illustrated in a case study for the preliminary design of an INDI controller for a tilt rotor system. In the example presented here the maximum height loss during the transition phase of this tilt-rotor system with respect to uncertainties in mass, moment of inertia and wind gusts was investigated. The results gave meaningful insights concerning the robustness of the controller and allowed to determine the sensitivities of the solution with respect to the uncertainties. Future work will be devoted towards the extension for more realistic models and additional clearance criteria.

Acknowledgements The authors wish to thank Matthias Bittner and Patrick Piprek for the helpful discussions and useful comments. This work was partly supported by the DFG grant HO4190/8-1.

References

1. Ben-Asher JZ (2010) Optimal control theory with aerospace applications. AIAA education series, American Institute of Aeronautics and Astronautics
2. Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming. Advances in design and control. Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 2nd edn. Philadelphia
3. Bryson AE, Ho YC (1975) Applied optimal control: optimization, estimation, and control. Taylor and Francis, New York
4. Büskens C (1998) Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustands-Beschränkungen: Münster (Westfalen), Univ., Diss
5. Diepolder J, Gabrys A, Schatz S, Bittner M, Grüter B, Holzapfel F, Ben-Asher JZ (eds) (2016) Flight control law clearance using worst-case inputs (To be published)
6. Fernandes de Oliveira R, Puyou G (2011) On the use of optimization for flight control laws clearance: a practical approach. 18th IFAC World Congress 44(1):9881–9886
7. Fiacco AV (1983) Introduction to sensitivity and stability analysis in nonlinear programming, vol 165. Mathematics in science and engineering. New York, Academic Press
8. Forssell LS, Åke Hydén (eds) (2003) Flight control system validation using global nonlinear optimisation algorithms
9. Gill PE, Murray W, Saunders MA (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev 47(1):99–131
10. Lai KL (2007) Extensions of the first and second complex-step derivative approximations. J Comput Appl Math
11. Martins JRRA (2003) The complex-step derivative approximation. ACM Trans Math Softw 29(3):245–262
12. Menon PP, Bates DG, Postlethwaite I, Marcos A, Fern V, Ez Bennani S Worst-case analysis of flight control laws for re-entry vehicles

13. Menon PP, Kim J, Bates DG, Postlethwaite I (2006) Clearance of nonlinear flight control laws using hybrid evolutionary optimization. IEEE Trans Evol Comput 10(6):689–699
14. Price KV, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Natural computing series. Springer, Berlin
15. Sieberling S, Chu QP, Mulder JA (2010) Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. J Guid Control Dyn 33(6):1732–1742
16. Varga A, Hansson A, Puyou G (2012) Optimization based clearance of flight control laws: a civil aircraft application. Lecture notes in control and information sciences. Springer, Berlin

An Unusual Structure for a Feedforward Gust Load Alleviation Controller

Nicolas Fezans

1 Introduction and Motivation for Active Load Alleviation with Anticipation Capabilities

Inhomogeneous wind fields such as turbulence and gusts are causing variations of the global and local (considering their distribution over the whole airframe) aerodynamic forces and moments that are applied to the aircraft structure. In addition to causing structural loads that the structure should be designed to support, these additional forces and moments also cause passenger discomfort and anxiety. Active load alleviation of gusts and turbulence is not a new topic: the investigations made on active load control to solve the Lockheed C-5A fatigue issues and leading to the development of the Active Lift Distribution Control System (ALDCS) dates back forty years [9]. Already at that time, the trade-off between structure mass and use of active control technologies was present. Historically, there have been two main drivers for investigations on active load alleviation within the last forty years:

- either a structure design was available but was for some reasons too weak (e.g. due to an increase in payload capacity) and the use of active control solved (or was meant to solve) the problem,
- or designers were interested in saving mass (and thereby increasing airplane efficiency) thanks to active load reduction systems.

Loads are not only generated by gusts and turbulence but can be caused (among others) by maneuvers or during touchdown and ground operations. In order to optimize weight savings various load cases might need to be considered simultaneously. Even if not in the scope of this paper, gust and turbulence load alleviation functions are often combined with a maneuver load alleviation function which shifts the lift

N. Fezans (✉)
DLR, Institute of Flight Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: nicolas.fezans@dlr.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_3

distribution over the wingspan towards the wing root during maneuvers with high load factors. Over the last decades, many load alleviation functions have been successfully implemented on numerous airplanes such as: Lockheed C-5A, Lockheed L-1011-500, Boeing B-1, Northrop Grumman B-2, Airbus A320, Airbus A330/A340, Airbus A380, Boeing 787, Airbus A350. Reference [26] and the references therein give an interesting overview of the applications of active load alleviation.

The numerous successful applications of active control technologies for airplane gust alleviation logically ended up reaching even the maximum technology readiness level (TRL) of 9 for some of these systems. Consequently, the orientation of the research activities of DLR on gust alleviation moved from more classical gust alleviation system design (such as in OLGA [3, 13, 20], or LARS [8, 18, 19]) to the investigation of more advanced solutions for a further improved alleviation performance. The investigations presented hereafter aim at improving the anticipation capability of future feedforward load alleviation functions. Complementary works also made by DLR on improved controller synthesis methods for multi-objective and robust feedback load alleviation (see [14–17, 21, 22]) are not in the scope of this paper.

Previous works focusing on feedforward control and its anticipation of near future loads had been performed over the last decades and had yield in particular to various systems developed during the AWIATOR project [7, 10–12, 25]. These systems also considered the use of LIDAR sensors for feedforward load alleviation purposes. As for these previous investigations, the feedforward load alleviation function is based on the idea that with a better anticipation of the near future loads a higher load alleviation performance can be achieved. Consequently, in all these systems one of the major components is dedicated to the determination of the expected near future loads. This is realized by gathering information on the wind field ahead of the aircraft, which in the concept shown hereafter, is based on a Doppler LIDAR sensor and a rather extensive processing of the measurements. This information is then used to alleviate these future loads in feedforward.

The processing steps allowing to determine the wind field ahead of the aircraft were described in [6] and will not be further described here. The information on the wind field ahead of the aircraft is therefore the starting point of the feedforward controller that is presented hereafter: it is used to alleviate the loads that are expected to be induced by the inhomogeneous part of this wind field.

Anticipating the future loads opens new possibilities in terms of load alleviation, but cannot replace a feedback controller that directly acts on the closed-loop behavior of the structural modes. These two functions (feedforward and feedback) are radically different and complementary. The proposed feedforward load alleviation has been designed as an add-on to a feedback load alleviation controller. As a consequence, the feedforward function mainly aims at providing load alleviation commands that are complementary to what a good feedback load alleviation would provide. With other words, the parts of the future loads that could be alleviated by a feedback load alleviation can but do not necessarily need to be alleviated by the feedforward function. The parts of the future loads that cannot be alleviated by a feedback load alleviation function (regardless of how well tuned it is) are the primary target for the feedforward load alleviation function.

A typical example of loads that cannot be well alleviated by a feedback load alleviation function but can significantly be reduced by a feedforward load alleviation function are the loads resulting from gusts with large amplitude and relatively large scales. The anticipation capability of the feedforward enables counteracting them by pitching the aircraft, which is roughly one order of magnitude more effective for lift increase/reduction than what can be done with the actuators located on the wings (ailerons, spoilers/speedbrakes, flaps).

The load alleviation functions (both in feedback and in feedforward) are in general add-ons to the basic control laws (later referred as "EFCS"), which provide the flight control augmentation function (laws based on n_z / C* or C*U / RCAH etc.). These "basic" laws can be developed very soon during the development and flight test program of the aircraft and are usually the main drivers for the handling qualities of the aircraft. The other functions (such as feedback and feedforward gust and turbulence load alleviation) are in general trying to satisfy other criteria without deteriorating the handling qualities that are provided by the "basic" laws.

From a pure control theory point of view it could be argued that imposing a structure to the controller with separated modules is adding constraints and therefore preventing reaching the "optimal solution." However, this is a rather theoretical debate since the controller design problem considering all control objectives and constraints at once on a high-fidelity flexible aircraft model will not be tractable. Moreover a complete integration of the control functions would also be very challenging in terms of time management and communication due to the fact that the design and the validation of these functions typically involve several disciplines with specialists spread across various departments. In this work, the typical decomposition of the overall control system into the "basic" control laws and load alleviation functions is not discussed but simply considered as a constraint. Only the feedforward load alleviation function and in particular its structure is in the scope of this paper. However, two complementary functions (a maneuver load alleviation function and a gust/turbulence load alleviation function in feedback) were also developed during this work. These three functions are represented in Fig. 1 with the other parts of the complete closed loop system. In this figure, various possible interconnections of the feedforward module with the rest of the system are shown in blue (dashed) and magenta (fine dashed). Not all of the shown interconnections are always required: the need for each one of them depends on the exact behaviors of the various controllers and thereby on the undesired interactions that might have to be prevented. The block-diagram shown in Fig. 1 assumes that the maneuver load alleviation function acts as a pre-filter between the pilot inputs and the "rigid-body controller" (in general part of the basic EFCS). This might however not necessarily be the case: combining maneuver load alleviation and feedback load alleviation also makes sense and would lead to a modified structure.

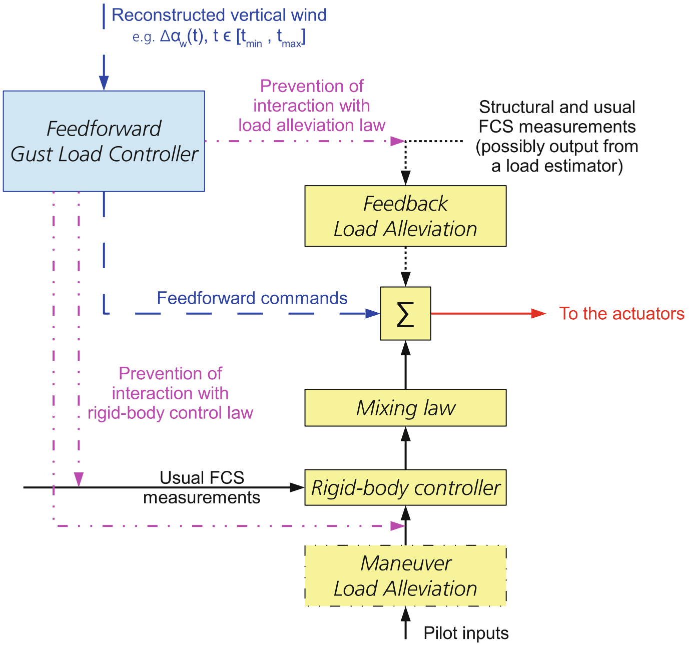

Reconstructed vertical wind
e.g. $\Delta\alpha_w(t)$, $t \in [t_{min}, t_{max}]$
Feedforward Gust Load Controller
Prevention of interaction with load alleviation law
Structural and usual FCS measurements (possibly output from a load estimator)
Feedback Load Alleviation
Feedforward commands
Σ
To the actuators
Prevention of interaction with rigid-body control law
Mixing law
Usual FCS measurements
Rigid-body controller
Maneuver Load Alleviation
Pilot inputs

Fig. 1 Flight control system architecture with active load alleviation functions

2 Motivation for a Novel Feedforward Controller Structure

In the previous section the decomposition of the complete flight controller into various subfunctions was shortly presented and discussed. This section focuses more specifically on the internal structure of the feedforward gust and turbulence load alleviation function.

In the specification of the desired load alleviation behavior made by the loads specialists who were involved in the definition of the problem addressed in the present paper, a typical wording came repeatedly: “small amplitude disturbances should not be alleviated using the spoilers.” While this is definitely reasonable from an airplane performance point of view, this implicitly specifies that a highly nonlinear allocation constraint is desired for the controller. This constraint makes the direct[1] application of some of the most powerful tools and results of the control theory impossible. It also raised the question of defining a simple controller structure, which can easily be tuned and permits to obtain such a highly nonlinear behavior. Pitching the aircraft

[1] Note that indirectly, after having exploited the structure proposed hereafter, the advanced tools provided by the linear control theory could actually be applied to each of the resulting subproblems.

up or down is the most effective way to change the aerodynamic loads. For an effective load alleviation with a restricted bandwidth and based on pitching actions, the pitching commands shall be initiated before encountering the disturbances. After having considered the motivations for such a desired alleviation behavior, the practical aspects regarding structural loads at the horizontal tailplane and the fuselage as well as for passenger comfort, the proposed alleviation concept was finally expressed as follows:

- The low frequencies of the atmospheric disturbances should be alleviated by pitching up or down the airplane. For passenger comfort reasons, it shall be possible to select a different behavior for low amplitude disturbances or even to restrict this behavior to large amplitude disturbances (i.e. relevant for peak loads).
- In the medium-frequency range:
  - The tuning parameters should allow to choose whether disturbances with very small amplitudes in the medium-frequency range are alleviated or not (e.g. to avoid unnecessary actuator cycles, or to reduce power consumption).
  - The disturbances with relatively small amplitudes should be alleviated using only trailing edge deflections or camber variations (so basically with ailerons but eventually also with innovative flaps if available).
  - The larger disturbances should be alleviated by any possible means of control, including spoilers or any other device even if they tend to deteriorate the airplane performance.
- The higher-frequency components of the disturbance will not be alleviated at all with the feedforward function.

The limits between "low," "medium," and "higher" frequencies as well as the thresholds between "very small," "small," and "larger" amplitudes are tuning parameters for the feedforward load alleviation function. Note that the bandwidth of the feedback load alleviation function is usually expected to be higher than the one of the feedforward load alleviation function. There is a priori no reason to impose any relationship between the bandwidths of feedforward and feedback functions, however the limited wind sensor spatial resolution lowers the effective achievable bandwidth of the feedforward load alleviation function.

Designing a controller providing the aforementioned behavior is not trivial. Even if the amplitude decomposition were ignored and thereby only a frequency decomposition were considered, specifying this frequency decomposition without impairing the anticipation of the feedforward controller in a controller design problem would involve relatively complex filters or design constraints. Eventually, the problem will probably be hardly tractable and only researchers in automatic control will be able to manipulate that problem and tune the design parameters. The controller structure that is proposed in the next section addresses this problem by decomposing the problem into more manageable subproblems and by providing tuning parameters that can easily be interpreted physically.

3 Sketch of the Proposed Feedforward Controller Structure

The design of the feedforward gust and turbulence load alleviation strategy that was described in the previous section is clearly not straightforward. It must behave strongly nonlinearly at some frequency ranges and consists of at least two completely separated types of alleviation: one based on local actions on the wing and the other based on pitching up or down the aircraft. At the same time the phase-lag of the alleviation commands should not be too high, otherwise the effectiveness of the load alleviation will vanish. Since a piece of the future wind profile is known in advance, it is actually possible to perform the required filtering operations without adding phase-lag.

The concept proposed here is very unusual but relies on a very simple idea: the filtering and highly nonlinear allocation between the different alleviation substrategies will be enforced by a preprocessing step. Note that even if the developed controller structure has been fully driven by the requirements of the considered application, the resulting concept and structure can certainly be useful for numerous other applications.

The overall architecture resulting from the direct application of the aforementioned idea to the feedforward load alleviation controller can be represented schematically by the block-diagram of Fig. 2. A time-frequency transformation/decomposition of the signal that is known partly in advance (here the wind ahead of the aircraft) is performed first. The exact result of this operation depends on the time-frequency technique used, but the information contained in the original signal is now expressed as “at this point in time, the signal contains these frequencies with these amplitude and phase” whereas it was previously expressed as “at this point in time, the signal had this value.” With most techniques, the information was not deteriorated and could be transformed back with no loss. The capacity of restoring the original signal is indeed very important for the proposed architecture and the inverse transformation will be used later.

The following step is to select the components of the signal that are of interest for the system and to treat them. For the sake of the example, the high-frequency noise on the signal could be removed by erasing the corresponding transformed data (e.g. the amplitude coefficients for all frequencies higher than a given threshold can be set to 0). If the altered transformed representation of the signal would be transformed back to the time domain, it would seem that the original signal was low-pass filtered but with no phase-shift. The nonlinear behavior desired for the pitching actions (i.e. only consider low-frequency high-amplitude components of the signal) can be obtained by selecting only the low frequencies of the signal and among them neglecting the components whose amplitudes are below a certain threshold. By transforming the remaining components back into the time domain, it would appear that the desired characteristics have been separated from the rest of the signal.

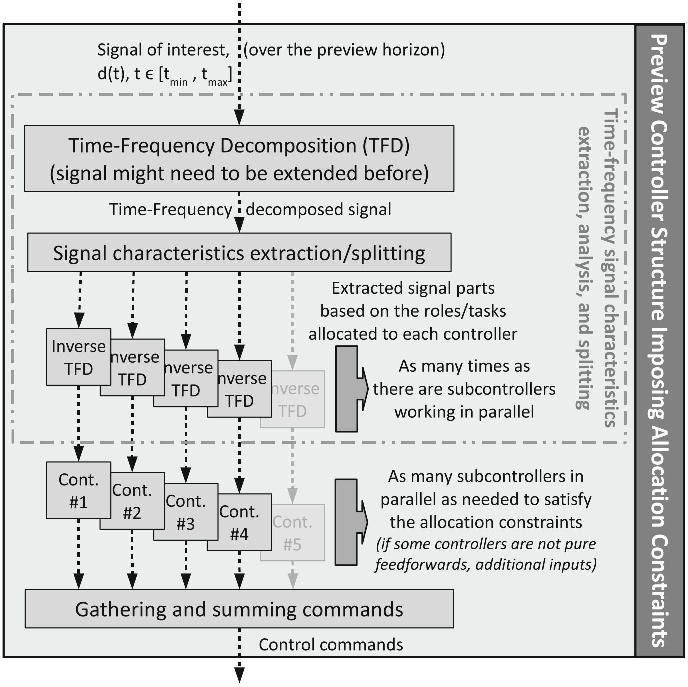

Signal of interest, d(t), t ∈ [t_min , t_max] (over the preview horizon)
Time-Frequency Decomposition (TFD) (signal might need to be extended before)
Time-Frequency decomposed signal
Signal characteristics extraction/splitting
Extracted signal parts based on the roles/tasks allocated to each controller
Inverse TFD
As many times as there are subcontrollers working in parallel
Time-frequency signal characteristics extraction, analysis, and splitting
Cont. #1
Cont. #2
Cont. #3
Cont. #4
Cont. #5
As many subcontrollers in parallel as needed to satisfy the allocation constraints
(if some controllers are not pure feedforwards, additional inputs)
Gathering and summing commands
Control commands
Preview Controller Structure Imposing Allocation Constraints

Fig. 2 Overall structure for the feedforward load alleviation with allocation constraints

Once back in the time domain, the various signals that are obtained by using this preprocessing can then be used by several controllers working in parallel. Each controller has a restricted and well defined role, which eases its design. For instance, the low-frequency high-amplitude part of the signal can be given as input to a simple controller that can only provide an additional elevator command: the tuning of such a function is very simple and the validation of its behavior also. Another controller can take care of some medium-frequency parts of the signal by providing symmetrical aileron commands.

This section gave a first insight on the structure that is proposed and on the way this structure can be useful for realizing the behavior that was mentioned earlier. The concrete implementation that was made uses the so-called Fast Orthogonal Wavelet Transform, which is a time-scale instead of a time-frequency technique but the basic idea remains the same. This implementation is detailed in the next section.

4 Signal Characteristics Extraction and Splitting Using the Fast Orthogonal Wavelet Transform

4.1 *Wavelets and the Fast Orthogonal Wavelet Transform*

The development of wavelet transforms was a major step in local time-frequency decomposition of continuous and discrete signals [24]. The complete theory of the various wavelet transforms and their use for signal processing goes far beyond the scope of this paper. Consequently, only a few remarks required to understand the approach used for the signal characteristics extraction and signal splitting step of the proposed feedforward controller will be provided here. For a deeper insight in wavelet theory and applications the reader is referred to [24]. Note that wavelets are extensively used in image compression algorithms, which also have the advantage of providing examples that are well-suited for illustrating some particular behavior. As a consequence a vast part of the literature on wavelets uses images as examples, but the principle remains the same: the time dimension is replaced by both dimensions of the image and the frequency dimension by the local changes of the image when following a straight line on the image.

Wavelets are sometimes described as “wave-like oscillations with an amplitude that begins and ends at zero.” For all wavelets, the amplitude starts and ends at 0, but can follow pretty much any profile in between. A few mathematical properties of wavelets are important to mention here:

- the average of a wavelet $\psi \in \mathscr{L}^2(\mathbb{R})$ is zero (i.e. $\int_{-\infty}^{+\infty} \psi(t)\, dt = 0$),
- it is normalized ($\|\psi\| = 1$),
- and it is centered in the neighborhood of $t = 0$.

Based on the wavelet basis function ψ, a scale and shifted version of it can be defined as:

$$\forall (u, s) \in \{\mathbb{R}, \mathbb{R}^{+*}\}, \psi_{u,s} = \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) , \tag{1}$$

where $\mathbb{R}^{+*}$ is the set of strictly positive real numbers. Note that the normalization still holds for the scaled and shifted wavelets: $\forall (u, s) \in \{\mathbb{R}, \mathbb{R}^{+*}\}$, $\|\psi_{u,s}\| = 1$. The wavelet transform of a signal $f \in \mathscr{L}^2(\mathbb{R})$ at time u and scale s reads:

$$\forall (u, s) \in \{\mathbb{R}, \mathbb{R}^{+*}\}, Wf(u, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt\ , \tag{2}$$

where $\psi^* \left(\frac{t-u}{s} \right)$ is the complex conjugate of $\psi \left(\frac{t-u}{s} \right)$. The function $\psi_{u,s}$ can here be seen as a (flipped) convolution filter applied to the signal f. The zero-average property of $\psi_{u,s}$ and its limited support (region of $\mathbb{R}$ where it is nonzero) implies that the constant part of the signal and the very low frequencies will not be transmitted through this “filtering operation.” The frequencies contained in the function $\psi_{u,s}$ also limit the frequencies (both minimum and maximum) contained in the “filtered” signal. In other words, the wavelet transform will behave as a band-pass filter that considers only the input signal in the neighborhood of $t = u$. By applying this transform for different values of u and s different frequency-bands and portions of the signal can be considered separately.

The Fast Orthogonal Wavelet Transform (FOWT) is one particular type of wavelet transform. It allows to decompose a discrete time signal in several successive steps. Each step leads to a decomposition of the signal into

- its approximation or general trend (lower-frequency part)
- its details (higher-frequency part).

The obtained approximation is undersampled (factor 2) between each step, so that the next decomposition generates signal characteristics with a lower frequency than the previous step. This general process is schematically represented in Fig. 3. The decomposition preserves time-wise correlation, which means, that information about the signal characteristics at a certain time is still available. Furthermore, each signal can be recomposed with an inverse process using the obtained approximations and details.

These decomposition steps are based on convolutions between the signal (the original signal at the first step or the approximation of the signal at the current level) and a pair of complementary discrete filters. The complementary property leads to

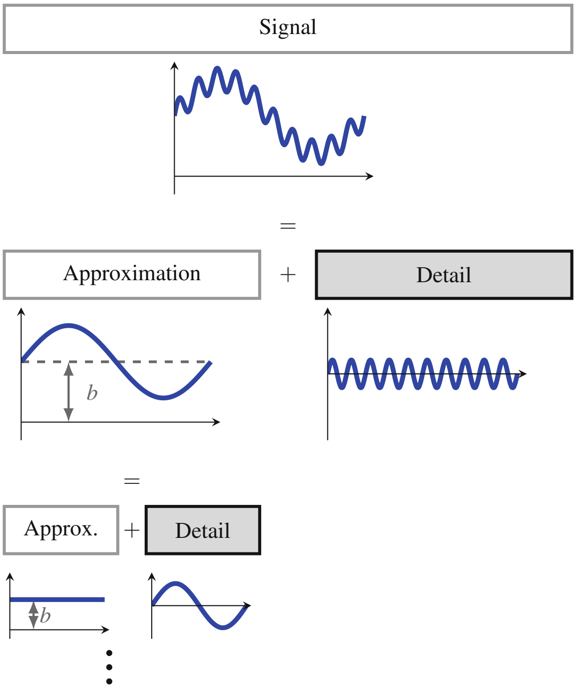


Fig. 3 Schematical representation of the signal decomposition steps based on the FOWT [4]

have (at each level) the sum of the signals corresponding to the "detail" and the "approximation" part equal to the original signal.

For the type of processing that is considered in this paper, it is very important to use a wavelet basis for which perfect reconstruction filters with a finite impulse response exist. This allows to reconstruct easily the original signal from its decompositions (with other words this allows to define the inverse transform). The filters used for reconstructing the signal (i.e. for restoring the signal based on its decomposition as series of coefficients) are different from the filters used for decomposing it. This leads to have four filters, noted D_h, D_l, R_h, and R_l hereafter: D and R respectively stand for decomposition and reconstruction and the indices h and l stand for high-frequency (i.e. detail part) and low-frequency (i.e. approximation part).

The choice of the wavelet base influences the possibility of recovering signal properties from the decompositions. Basically, the wavelet base function is correlated with the signal at different frequencies and magnitudes (similar to a Fourier-Transform). For the work in this paper, the so-called bior3.9 wavelet was chosen. The four corresponding filters are defined by the coefficients provided in Eqs. (3)–(6).

$$D_h = [0_{1\times 8}\ ,\ -0.1767766953\ ,\ 0.5303300859\ ,\ -0.5303300859\ ,\ 0.1767766953\ ,\ 0_{1\times 8}] \tag{3}$$

$$D_l = \begin{array}{l} [\ -0.0006797444\ ,\ 0.0020392331\ ,\ 0.0050603192\ ,\ -0.0206189126\ ,\ \ldots \\ \ \ -0.0141127879\ ,\ 0.0991347825\ ,\ 0.0123001363\ ,\ -0.3201919684\ ,\ \ldots \\ \ \ \ \ 0.0020500227\ ,\ 0.9421257007\ ,\ 0.9421257007\ ,\ \ \ 0.0020500227\ ,\ \ldots \\ \ \ -0.3201919684\ ,\ 0.0123001363\ ,\ 0.0991347825\ ,\ -0.0141127879\ ,\ \ldots \\ \ \ -0.0206189126\ ,\ 0.0050603192\ ,\ 0.0020392331\ ,\ -0.0006797444\ \ ] \end{array} \tag{4}$$

$$R_h = \begin{array}{l} [\ -0.0006797444\ ,\ -0.0020392331\ ,\ 0.0050603192\ ,\ \ \ 0.0206189126\ ,\ \ldots \\ \ \ -0.0141127879\ ,\ -0.0991347825\ ,\ 0.0123001363\ ,\ \ \ 0.3201919684\ ,\ \ldots \\ \ \ \ \ 0.0020500227\ ,\ -0.9421257007\ ,\ 0.9421257007\ ,\ -0.0020500227\ ,\ \ldots \\ \ \ -0.3201919684\ ,\ -0.0123001363\ ,\ 0.0991347825\ ,\ \ \ 0.0141127879\ ,\ \ldots \\ \ \ -0.0206189126\ ,\ -0.0050603192\ ,\ 0.0020392331\ ,\ \ \ 0.0006797444\ \ ] \end{array} \tag{5}$$

$$R_l = [0_{1\times 8}\ ,\ 0.1767766953\ ,\ 0.5303300859\ ,\ 0.5303300859\ ,\ 0.1767766953\ ,\ 0_{1\times 8}] \tag{6}$$

Figure 4 presents a generic block diagram for a signal manipulation in the time-scale domain. The decomposition filters D_h and D_l are applied in combination with undersampling operations to decompose the signal provided in input in several steps. In this figure, the decomposition was made up to the third level. The number of decomposition levels to use should be chosen depending on the needs of each specific application and relates to the lowest frequency band to be decomposed. The fact that the input signal has a limited length limits the number of decomposition levels that can be used (each undersampling operation shortens the signal) and therefore the lowest frequency band that can be considered. In the considered application

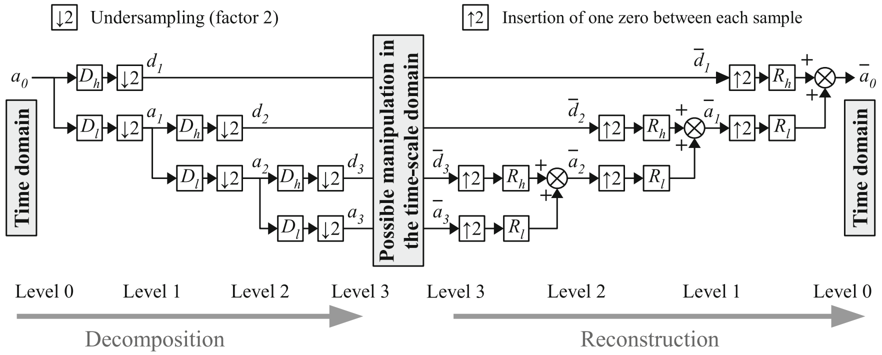


Fig. 4 Representation of the signal decomposition and reconstruction steps based on a filter bank implementation the FOWT. The decomposition is shown up to level 3. The blocks D_h, D_l, R_h, and R_l are convolutions with the corresponding filter kernels (basically a vector when processing 1D signals). In the *middle* of this figure, the location at which a module for analysis and manipulation of the signal in the time-scale domain can be inserted is represented

to feedforward active load alleviation, this property was limiting the anticipation capability. As a consequence the signal that is passed to the FOWT (noted a_0 in Fig. 4) is an artificially extended version of the original signal: see Sect. 4.2 for the details on the signal extension. On the right of the block diagram shown in Fig. 4, the reconstruction operations based on the filters (R_h and R_l) and on the insertion of zeros between each sample (counterpart of the undersampling operations in the decomposition). The type of manipulations that can be made in the time-scale domain (block in the middle) will be described later in Sect. 4.3.

4.2 Artificial Signal Extension

The area ahead of the aircraft for which the wind information can be reconstructed based on the LIDAR measurements is limited in size due to the limited sensor range. Increasing the measurement range (when possible) is expected to cause a significant deterioration of the measurement quality (higher noise levels). Some measurements located behind the aircraft current position are kept to prevent effects related to the border of the estimation domain (see [6] for more information). This part is not directly relevant anymore for the load alleviation functions. As a consequence, considering the foreseeable sensor performances the typical length of the reconstructed wind profile will represent roughly 1 s of flight or 250 m.

This limited duration is an issue for the extraction of the low-frequency content of the wind profile, which is meant to be alleviated with pitching commands. The proposed solution consists in extending the wind field artificially with an exponential decay between the end of the profile and converging to long-term (over 10–20 s) average of the wind. Making this artificial extension converge to the observed average instead of zero is meant to increase the robustness of the solution against slight sensor calibration errors and also against relatively large areas with non-negligible mean vertical wind (e.g. thermal effects, turbulence, lee waves, or downburst). The time-constant τ_{decay} for the exponential decay shall be set to a value in the neighborhood of the inverse of the lowest frequency $f_{GLA,low}$ that is worth to consider for the load alleviation functions (i.e. $\tau_{decay} \in [\frac{0.2}{f_{GLA,low}}, \frac{5}{f_{GLA,low}}]$).

This artificial extension of the field can be seen as some kind of "educated guess" on the general trend of the wind field outside of the domain where it was explicitly estimated. It is clear that this "guess" is likely to be relatively far from the reality at the medium and high frequencies, but it suffices that it is representative enough for the low frequencies. Indeed, the low-frequency pitching command generation is the only reason for needing a longer wind prediction horizon. Note that this guess or approximation does not need to be good in the medium and high frequency ranges.

4.3 Overview of Wavelet Thresholding and of Its Use for Extracting the Interesting Parts of the Signal

The previous sections presented the way the signal can be artificially extended to permit the extraction of its lowest frequencies (Sect. 4.2) and how a signal can be transformed to the time-scale domain and back using a FOWT filter bank (Sect. 4.1). In the middle of Fig. 4 the place at which a manipulation of the signal in the time-scale domain can be performed is indicated. This section presents the way these manipulations were made for the application to active load alleviation.

The technique used to manipulate the coefficients at a given level such that low amplitude coefficients are ignored is commonly known as "wavelet thresholding" or "wavelet shrinkage." The two most common thresholding function functions are shown in Fig. 5.

The soft-thresholding function can be interpreted as a dead-zone: the coefficients smaller (in absolute value) than the chosen threshold are set to zero whereas the threshold magnitude is subtracted to the others. This yields to effectively ignore low-amplitude parts of the signal but also reduces the amplitude of the larger ones and thereby possibly prevents to reach the maximum achievable alleviation performance. The hard-thresholding function is not affected by this problem but the discontinuity

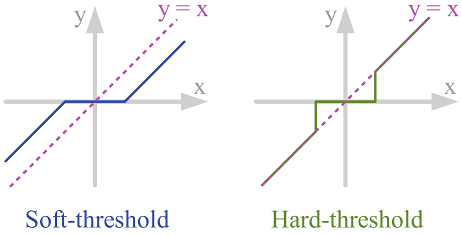


Fig. 5 The two most common thresholding function: soft and hard

at the threshold causes artefacts in the reconstructed signals, at times for which the wavelet coefficients are spread on both sides of the threshold.

The desired shrinkage function shall be zero (or very small) for low input values, converge to the linear function $y = f(x) = x$ for x large, and be continuous and tunable in between (but not necessarily smooth). The "smooth sigmoid-based shrinkage" family of functions (SSBS) [1, 2, 23] satisfies all these requirements and also provides a few additional degrees-of-freedom. It reads:

$$\delta_{t,\tau,\lambda} : \mathbb{R} \to \mathbb{R},\ x \mapsto \frac{sign(x)\ max(0, |x| - t)}{1 + e^{-\tau(|x|-\lambda)}}\quad . \tag{7}$$

The role of the parameters t and λ can easily be understood based on Fig. 6. The parameter τ defines the abruptness of the transition between the point $(0, t)$ and the asymptotes $y = f(x) = x - t$ in $+\infty$ and $y = f(x) = x + t$ in $-\infty$. In the application considered in this paper the additional degree-of-freedom provided by t is not used and the value of t is always set to 0 here. This corresponds to a case where the SSBS function behaves as the soft-threshold for amplitudes far below the threshold λ, behaves as the hard threshold for amplitudes far above the threshold λ, and provides a smooth transition between these two behaviors in the neighborhood of the threshold λ.

The way the wavelet thresholding technique can be used for the considered application to the separation of signal characteristics for use by different load alleviation controllers as proposed in Sect. 3 is explained hereafter. In a first step, the specification of the limit between the low-frequency range (that will be alleviated by means of pitching commands) and the medium-frequency range (alleviated with the actuators on the wing) can easily be made by defining that this limit shall occur between decomposition level (i.e. number of decomposition stages) P and level $P - 1$. The signal to be used for the pitching actions is then based on the selected decomposition

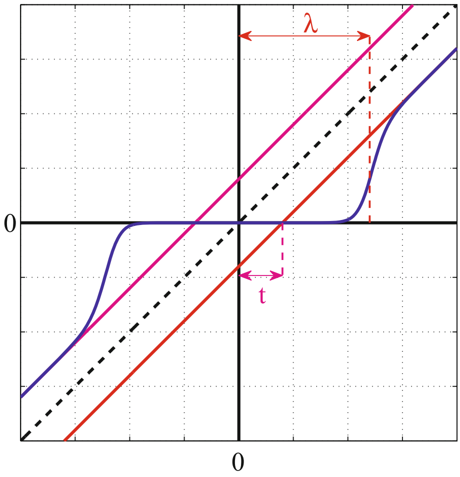


Fig. 6 Representation of the SSBS function (*blue line*). Asymptotes in $+\infty$ and in $-\infty$ are respectively represented in *red* and in *magenta*. The *black dashed line* represents the $y = x$ function

level P, assuming during the reconstruction process that all “detail” coefficients at the lower levels (i.e. $P-1, P-2, \ldots, 0$) were equal to zero. Similarly, the signal containing the remaining frequencies (all but the low frequencies) can be reconstructed by using the coefficients at levels $P-1$ and below and by ignoring the coefficients from level P.

This strategy permits to split several frequency bands but still does not restrict the pitching-based alleviation to large amplitude disturbances. For this, the coefficients at level P are modified using the wavelet thresholding technique based on SSBS functions with $t=0$. Even though that was not considered interesting for this application,[2] it would be possible to consider the part of the coefficients at level P that is not considered for the pitching actions and to provide them to one of the other controllers.

In order to completely filter out the higher frequencies a level R (with $R<P$) can be defined and the “detail” coefficients for the levels between 0 and R can simply be set to zero. The signal extracted by the operations described above is the signal that is later passed to the first or “pitching commands” controller, which can be seen as the block labeled “Cont. #1” in Fig. 2.

The second and third controllers (“Cont. #2” and “Cont. #3”) can act on the wing control surfaces: only on the ailerons in the case of controller 2 and both on ailerons and spoilers in the case of controller 3. Both are working in the same frequency band but controller 2 is the only one active for small disturbances and controller 3 becomes active for larger disturbances. A similar signal extraction via wavelet thresholding with an SSBS function as for controller 1 is first applied to the intermediate levels (levels $P-1$ and below). Then for each of the considered levels the obtained wavelet coefficients are distributed to either controller 2 (the small ones) or controller 3 (the large ones). Indeed, the distribution is performed in a smooth manner using a logistic function $l_{B,M}$ for each of the considered levels.

$$l_{B,M} : \mathbb{R} \to [0,1], x \mapsto \frac{1}{1+e^{-B(|x|-M)}} \tag{8}$$

The function $l_{B,M}$ defines a coefficient between 0 and 1 such that for small x the value is (very close to) 0 and for large x it is (very close to) 1. So for a given wavelet coefficient c_{23} that was selected to be alleviated either by controller 2 or by controller 3, the coefficients c_2 and c_3 that correspond to the respective parts to be alleviated by controller 2 and by controller 3 are defined as follows:

$$c_2 = \big(1 - l_{B,M}(c_{23})\big)\ c_{23} \quad \text{and} \quad c_3 = l_{B,M}(c_{23})\ c_{23} \tag{9}$$

[2] The low amplitudes are removed here because it is expected that they essentially resulted from the sensor noise via the wind reconstruction algorithm: considering them and taking actions based on them is likely to not provide any load improvement. It would even generate load cycles in the structure while flying through calm air.

From the definition of $l_{B,M}$ the role of both parameters can easily be understood:

- M defines the location (here a threshold on the wavelet coefficient amplitude) whereat the transition between controllers 2 and 3 is occurring: when the coefficient is equal to M then 50% of it is taken care of by controller 2 and the other 50% by controller 3.
- B defines the growth rate: a large value for B causes an abrupt transition between 0 and 1.

The desired behavior necessarily corresponds to cases for which $B > 0$ and $M \geq 0$.

4.4 Illustration of the Obtained Signal Decomposition Technique

The filtering process presented in the previous section is applied to the reconstructed and extended wind signal provided by the wind reconstruction process of [6]. The result is shown in Fig. 7, where one of the reconstructed results obtained in [6] is considered as input signal. This reconstructed wind profile is shown in magenta. Its extension (see Sect. 4.2) is shown in black and the extracted three signals (1 per controller) are shown by the dashed-cyan lines. Note that these signals are the portions of the reconstructed vertical wind profile (in m/s) that is provided at each moment in time to the three controllers (pitch, symmetrical deflections of ailerons,

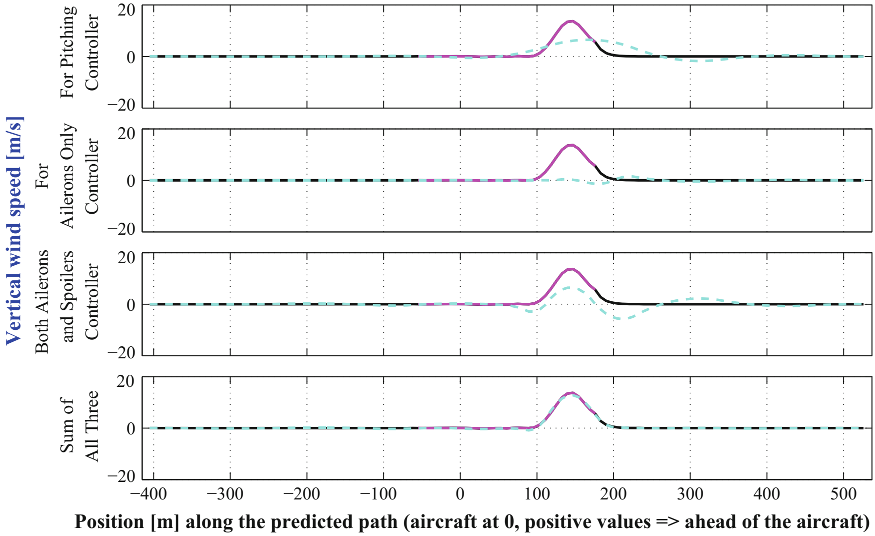


Fig. 7 Illustration of the behavior of the signal extension, filtering, and characteristics extractions on a partly reconstructed 1-cosine gust

and symmetrical deflections of ailerons and spoilers) and not the control commands. These wind profile portions are updated every 0.3 s but the corresponding controllers continuously generate control commands based on the last profile that they received in input (which can therefore be up to 0.3 s old) and the airplane current position. In Fig. 7 (as for the following ones), the aircraft is located at the position 0 m (i.e. the vertical wind profiles shown are drawn along the predicted aircraft path: positive values on the abscissa are ahead of the aircraft). Note that even with no wind at the current aircraft position, the prediction of the coming gust permits to anticipate pitching actions.

The case of Fig. 7 is a particularly favorable example, since the signal extension seem to prolongate almost perfectly the "1−cosine" gust profile. Figure 8 shows the example of the wind reconstruction result which was obtained 0.3 s before: this time the fact that the extension of the signal is just an decreasing exponential curve can clearly be seen. However, roughly the same input is generated for the pitching controller in Figs. 7 and 8, which shall not impair its capacity to anticipate. The pitching controller can begin to act even earlier, even though not even the half of the gust was located in the wind reconstruction mesh. The behavior obtained with lower disturbance amplitude is shown in Fig. 9. In this figure, the same gust than before is considered but at a later point in time: the aircraft moved forward such that the gust is now almost completely behind the wind reconstruction mesh. Still a short peak of about 2 m/s amplitude is contained in the reconstructed vertical wind (plus the imperfections due to the noise on the measurements). Negligible deviations from zero can be seen on the first signal (for pitching) and only minimal amplitudes are extracted for the ailerons-based controller. The comparison with the signals shown

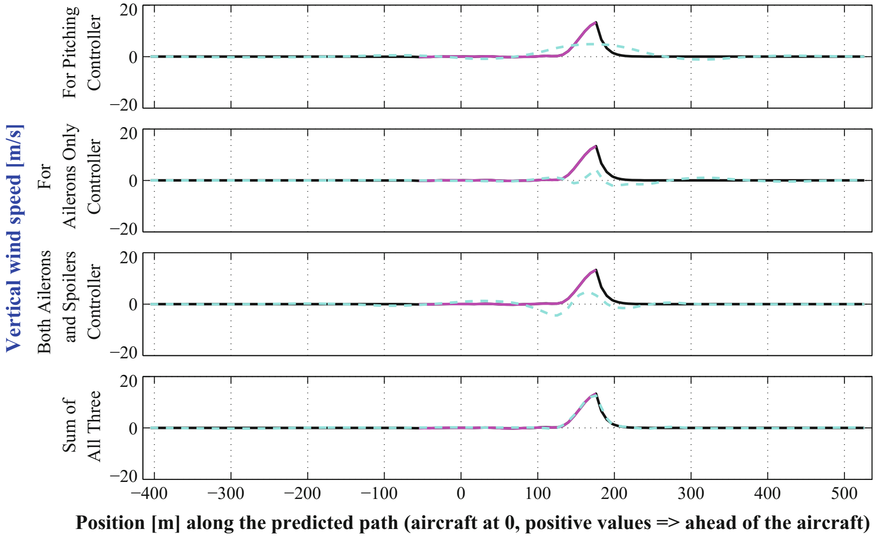


Fig. 8 Same as in Fig. 7, but 0.3 s earlier when less than a half of the gust was reconstructed

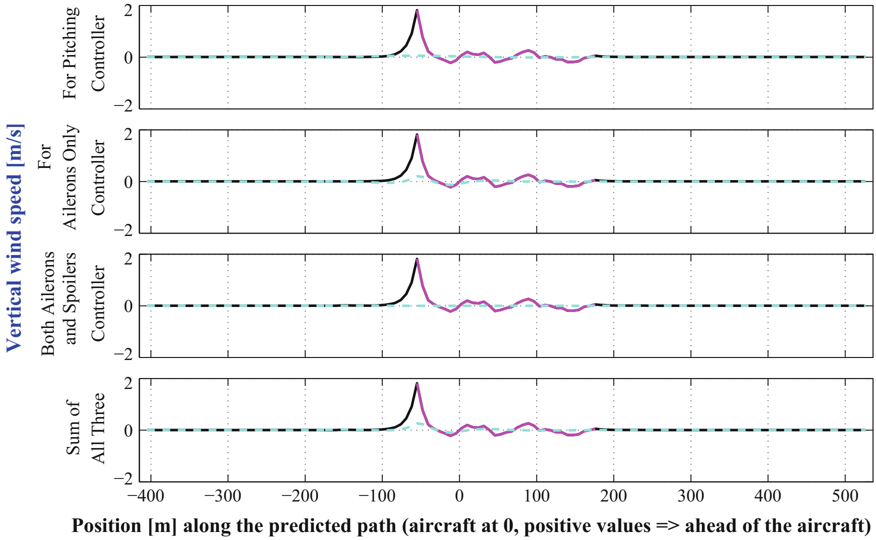


Fig. 9 Same as in Figs. 7 and 8, at the end of the encounter with the gust. Note that the scale for the ordinate significantly differs from the scale in Figs. 7 and 8

in Figs. 7 and 8 illustrates the strongly nonlinear behavior that was set to prevent any propagation of the sensor measurement noise (through an imperfectly reconstructed wind profile) up to the actuator commands. This behavior also permits to avoid affecting the passenger comfort and activating the feedforward load alleviation when no disturbance was detected that is large enough to cause significant loads.

5 Results on the Application to Airplane Feedforward Gust and Turbulence Load Alleviation

The proposed feedforward load alleviation controller structure has been tested in simulation. The considered aircraft model is an integrated flexible simulation model (based on the Airbus XRF1 model) in which the full nonlinear equation of motions are coupled with the aeroelastic model. It comprises also actuator and sensor models. This model was developed by the DLR Institute of System Dynamics and Control, who also designed a basic flight control system (later referred to as "EFCS") as well as an active load alleviation function (later referred to as "FBALC") based on more classical sensors (e.g. accelerometers) distributed on the aircraft structure.

The feedforward load alleviation function that uses the structure proposed here has been called "GLAREWISE+TFAC," which stands for "Gust Load Alleviation using REmote WInd SEnsing and with Time-Frequency-based Allocation Constraints." As it can be observed in the previous sections, this feedforward function does not try to alleviate the disturbances perfectly, but rather focuses on the signal components that are known with a high level of confidence and for which the anticipation capability of the feedforward enables a better load alleviation performance than what can be achieved with a feedback function. The reason for that is that from a system architecture point of view, there are three main possible system designs: EFCS with no load alleviation function, EFCS and feedback load alleviation, and EFCS plus integrated feedback+feedforward load alleviation. A feedforward load alleviation alone would already lead to a significant system complexity, but still not permits to directly act on the modes of the flexible aircraft (e.g. add some damping or increase the robustness of the load alleviation).

The cases considered hereafter are gusts with a one-cosine shape and all having the same amplitude. Three of the typical gust lengths considered for the certification are shown hereafter: 150, 300 and 350 ft. The 300 and 350 ft gusts are shown in both directions (upward and downward), whereas the 150 ft gust is only shown in the upward direction. Figure 10 shows the time responses of the wing root bending moment (left) and of the corresponding vertical accelerations at the middle of the cabin (right). For all five simulations shown, the black line corresponds to the "EFCS" case, the dashed magenta line corresponds to the "EFCS and FBALC" case, and finally the dash-dotted cyan line corresponds to the "EFCS, FBALC, and GLAREWISE+TFAC" case. The anticipation capability of the GLAREWISE+TFAC function can easily be seen by the fact that the dash-dotted cyan line begins to vary before the other two: this variation is mainly due to the pitching command that anticipate that loads in the opposite direction are expected to occur very shortly after. In all these simulations the gusts begin at the time t = 6 s. The first seconds are not shown here and are not relevant for the load analysis: during this time the aircraft flies simply in its trimmed condition. These seconds need however to be simulated in order to bring the internal states of the GLAREWISE+TFAC function to a representative state (number of LIDAR measurements in the buffer, spatial distributions of these measurements, etc.).

The 300 ft case is the most critical in terms of wind bending root moment for the considered airplane. As a consequence, the tuning of the feedforward function (filtering, characteristics extraction, individual controller tuning) was strongly oriented to providing load alleviation for gusts having a length in the neighborhood of 300 ft. The tuning was not oriented toward pure structural loads but also strongly considered the comfort, which can be seen on the vertical accelerations. The load reduction performance achieved in the 150 ft case on the wing bending root moment is not as good (relative improvement of the maximum load during the simulation) than in the 300 ft case. Note however that the absolute values are still lower than the absolute values obtained with both load alleviation functions in the 300 ft case. The vertical accelerations were still improved on the 150 ft case: reduced maximum value thanks to the GLAREWISE+TFAC anticipation capability and lower oscillations thanks to the mode damping provided by the FBALC function. On the 300 ft case, the reductions both in terms of wing bending root as in terms of vertical load factor are very promising. Note that the anticipation of the GLAREWISE+TFAC

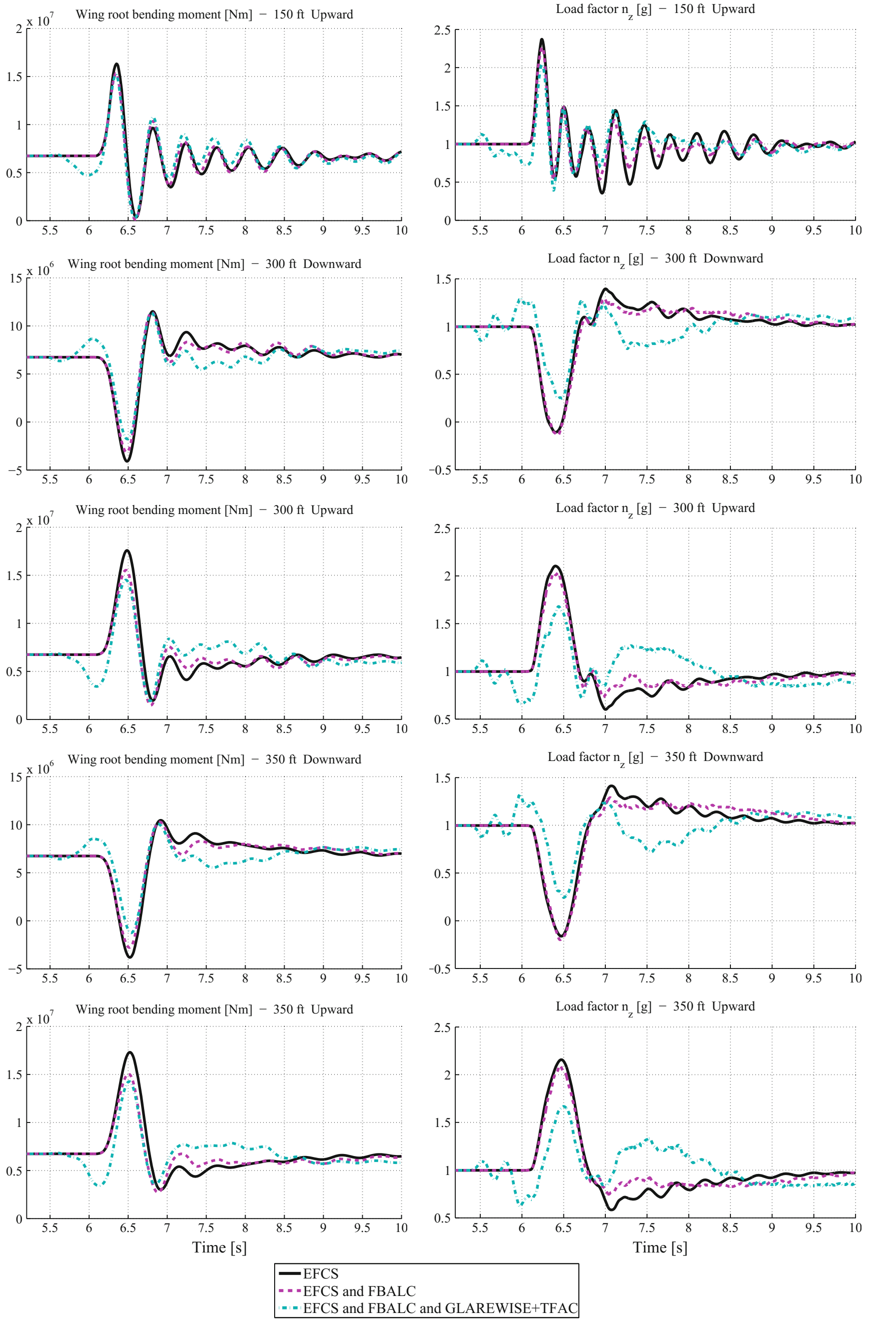


Fig. 10 Comparison of wing root bending moments and vertical acceleration (*middle* of cabin) over time with various 1-cosine gust of length 150, 300, and 350 ft

function permitted to prevent negative load factors in the downward 300 ft and 350 ft cases.

In terms of required computational power the FOWT-based decomposition, the signal characteristics extraction and splitting based on the SSBS and logistic functions, and the inverse FOWT are all very simple operations. The computation resources needed (CPU, memory) are not expected to pose any practical difficulty and are totally deterministic.

6 Conclusion and Outlook

An unusual feedforward controller structure that allows to take allocation constraints that can be expressed in the time-frequency or time-scale domains has been presented. This novel controller structure was motivated by the design of a feedforward controller for airplane gust and turbulence load alleviation based on Doppler LIDAR measurements. As shown in the application that has motivated this development, some strongly nonlinear constraints can be guaranteed by the design of the structure. The developed structure is not restricted to the considered application and could be applied successfully to many other feedforward control problems with some prior knowledge of the future disturbances, references, or parameters. More information on the nonlinear and robust controller design for the feedback load alleviation controller (used in this paper in addition to the feedforward function) will be published in the near future [5]. The control approach presented here focuses in designing a controller structure permitting to decompose the feedforward function into various subfunctions regardless of how these subfunctions are designed (e.g. with H_2, H_∞, or even adaptive control techniques as proposed in [27, 28]).

Acknowledgements The author thanks the European CleanSky initiative for the partial funding of this work (under grant CSJU-GAM-SFWA-2008-01) as well as the project partners involved in the SFWA WP1.2 and especially Airbus, which provided the flexible long range aircraft model data (XRF1 model), and the colleagues Thiemo Kier and Hans-Dieter Joos from DLR Institute of System Dynamics and Control, who prepared the integrated model and provided the corresponding simulation environment and basis controller (both EFCS and FBALC).

References

1. Atto AM, Pastor D, Mercier G (2008) Smooth sigmoid wavelet shrinkage for non-parametric estimation. In: Proceedings of the 2008 IEEE international conference on acoustics, speech, and signal processing (ICASSP)
2. Atto AM, Pastor D, Mercier G (2008) Wavelet shrinkage: unification of basic thresholding functions and thresholds. Signal Image Video Process (SIViP) 5(1):11–28. https://doi.org/10.1007/s11760-009-0139-y
3. Böhret H, Krag B, Skudridakis J (1985) OLGA an open-loop gust alleviation system. In: Proceedings of the AGARD CP 384 meeting, Toronto, Canada

4. Deiler C (2014) Data parser approaches for (online) parameter estimation. CEAS Aeronaut J 5(3):345–357
5. Fezans N, Joos H-D (2017) Combined feedback and lidar-based feedforward active load alleviation. In: 2017 AIAA atmospheric flight mechanics conference (submitted)
6. Fezans N, Schwithal J, Fischenberg D (2015) In-flight remote sensing and characterization of gust, turbulence, and wake vortices. In: Proceedings of the 2015 German aerospace congress (Deutscher Luft- und Raumfahrtkongress), Germany
7. Hahn K-U, Hecker S (2004) Gust load alleviation system. Technical report, DLR, Braunschweig, Germany. AWIATOR, Technical report, DLR-TR-3.1.1-12
8. Hahn K-U, König R (1992) Attas flight test and simulation results of the advanced gust management system LARS. In: Proceddings of the AIAA atmospheric flight mechanics conference, SC, USA
9. Hargrove WJ (1976) The C-5A active lift distribution control system. NASA, Technical report, pp N76–31148
10. Hecker S (2005) Gust load alleviation system using adaptive elements. Technical report, DLR, Oberpfaffenhofen, Germany. AWIATOR, Technical report, DLR-TR-3.1.1-13
11. Hecker S, Hahn K-U (2003) Gust computation system. Technical report, DLR, Oberpfaffenhofen, Germany. AWIATOR, Technical report, DLR-TR-3.1.1-11
12. Hecker S, Hahn K-U (2006) Gust load alleviation system using turbulence sensor and adaptive elements. Technical report, DLR, Oberpfaffenhofen, Germany. AWIATOR, Technical report, DLR-TR-3.1.1-14
13. Hoffmann G (1976) Stabilisierung. Böenkompensation und Schwingungsdämpfung am elastischen beweglichen Flugzeugmodell im Windkanal. Technical report, DFVLR (now part of DLR), Cologne, Germany
14. Joos H-D (1997) Multi-objective parameter synthesis (MOPS). In: Magni JF, Benani S, Terlouw J (eds) Robust flight control: a design challenge. Lecture notes in control and information sciences. Springer, Berlin, pp 199–217
15. Joos H-D (1999) A methodology for multi-objective design assessment and flight control synthesis tuning. Aerosp Sci Technol 3(3):161–176 April
16. Joos H-D (2011) Worst case parameter search based clearance using parallel nonlinear programming methods. In: Varga A, Hansson A, Puyou G (eds) Optimization based clearance of flight control laws. Lecture notes in control and information science. Springer, Berlin
17. Joos H-D, Bals J, Looye G, Schnepper K, Varga A (2002) A multi-objective optimisation based software environment for control systems design. In: Proceedings of the 2002 IEEE international conference on control applications and international symposium on computer aided control systems design (CCA/CACSD), Glasgow, Scotland
18. König R, Hahn K-U (1990) Load alleviation and ride smoothing investigations using attas. In: Proceedings of the 17^{th} Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden
19. König R, Hahn K-U, Winter J (1994) Advanced gust management systems - lessons learned and perspectives. In: Proceedings of the AGARD flight mechanics panel symposium on active control technology: applications and lessons learned, Torino, Italy
20. Krag B (1979) The wind tunnel behaviour of a scaled model with a gust alleviation system in a deterministic gust field. Trans Inst Meas Control 1(3)
21. Looye G, Joos H-D (2001) Design of robust dynamic inversion control laws using multi-objective optimization. In: Proceedings of the AIAA guidance, navigation, and control conference and exhibit, Montreal, Canada AIAA-2001-4285
22. Looye G, Joos H-D, Willemsen D (2001) Application of an optimization-based design process for robust autoland control laws. In: Proceedings of the AIAA guidance, navigation, and control conference and exhibit, Montreal, Canada
23. Lorenz DA (2004) Wavelet shrinkage in signal and image processing. Ph.D. thesis, University of Bremen
24. Mallat S (2009) A wavelet tour of signal processing - the sparse way. Academic Press, Dublin. ISBN: 978-0-12-374370-1

25. Rabadan GJ, Schmitt NP, Pistner T, Rehm W (2010) Airborne lidar for automatic feedforward control of turbulent in-flight phenomena. J Aircr 47(2)
26. Regan CD, Jutte CV (2012) Survey of applications of active control technology for gust alleviation and new challenges for lighter-weight aircraft. Tech report TM-2012-216008, NASA, Dryden Flight Research Center, Edwards, CA, USA
27. Wildscheck A (2008) An adaptive feed-forward controller for active wing bending vibration alleviation on large transport aircraft. Ph.D. thesis, TU Munich
28. Zeng J, Moulin B, de Callafon R, Brenner MJ (2010) Adaptive feedforward control for gust load alleviation. J Guid Control Dyn 33(2)

Aspects of a Consistent Modeling Environment for DO-331 Design Model Development of Flight Control Algorithms

Markus Hochstrasser, Simon P. Schatz, Kajetan Nürnberger, Markus Hornauer, Stephan Myschik and Florian Holzapfel

1 Introduction

Within various projects, the *Institute of Flight System Dynamics* at *TU München* developed a modular flight control software for usage in applications ranging from manned autopilot functions to unmanned operations from ground. The controllers were developed using a model-based approach with *MATLAB*, *Simulink*, and *Stateflow*. To guarantee high-quality models and code, and to pave the way for future certification, a model-based development process based on industry standards proposed for airborne software was developed.

In the chosen development approach according to DO-178C [19] and DO-331 [20], *Simulink* and *Stateflow* models step in the place of "Software Low-Level Requirements" and "Software Architecture". The "Design Models", as they are called by DO-331 in this context, are directly used to generate ANSI C "Source Code". The use of *MATLAB*, *Simulink*, and *Stateflow* in such a context requires a special setup,

M. Hochstrasser (✉) · S.P. Schatz · K. Nürnberger · M. Hornauer · F. Holzapfel
Institute of Flight System Dynamics, Technische Universität München, Munich, Germany
e-mail: markus.hochstrasser@tum.de

S.P. Schatz
e-mail: simon.p.schatz@tum.de

K. Nürnberger
e-mail: kajetan.nuernberger@tum.de

M. Hornauer
e-mail: markus.hornauer@tum.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

S. Myschik
Chair of Flight Mechanics and Flight Control, Universität der Bundeswehr München, Neubiberg, Germany
e-mail: stephan.myschik@unibw.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_4

preparation, and limitation of the tool capabilities. The required documentation and configuration is in general created during the Software Planning Process and in the following denoted as “Modeling Environment”. Estrada [6] emphasizes the importance of an environment, but does not discuss the details of its content.

The Modeling Environment describes a package of settings, libraries, and templates that are made available to developers in order to support them in implementing models and generate code that is safe and compliant to the defined process. The difficulty in setting up such an environment originates from the close connection between Design Model and Source Code and the standards both have to fulfill. Since code generation is normally done by a single function call, the Design Model not only describes the design itself, but also the appearance of the generated code. A well-prepared Modeling Environment can significantly improve the results of static code analysis and standard compliance checking in subsequent steps and reduce the remaining effort as shown in [9].

Another challenge is that model developers usually do not have the experience of a C programmer, but are responsible for the generated safety-critical code and its compliance to standards as well. Thus, the fulfillment is mainly determined by the quality and consistency of the conventions and settings from the provided Modeling Environment.

The Modeling Environment normally depends on multiple factors and may vary from company to company. However, the presented basic set is required in most of the projects. Since the content may be influenced by the chosen development process, the application to be developed and the hardware on which the application shall be executed, Sect. 2 introduces the applied model-based development process and Sect. 3 gives a short overview about the flight control computer hardware and the controller design. Based on this, Sect. 4 contains a detailed discussion of the suggested Modeling Environment.

2 Model-Based Software Development Process

The objective was to develop a controller application in alignment with guidance material for a CS-23 aircraft. In Book 2 of CS-23 [7], the relevant paragraph CS 23.1309 “Equipment, systems and installations” does not provide acceptable means of compliance for digital flight control systems, but EASA Certification Review Items refer to FAA Advisory Circular AC 23.1309-1E [8], which itself refers DO-254 [18] and DO-178B/C [19] as guidance material for software and hardware development.

With DO-178C, RTCA published additional supplements addressing modern techniques for software development like model-based approaches (DO-331 [20]) or formal methods (DO-333 [21]). Especially model-based development became popular in the recent years, since it provides means to implement complex software, simulate the models in an early development stage, and automatically generate code.

Tool vendors for model-based development software try to provide a consistent tool chain, allowing the fulfillment of DO-331 objectives in an effective and mostly automated way. Most popular are the workflows provided by *MathWorks* [32] and *Esterel* [5].

For the given project, the *Institute of Flight System Dynamics* decided to setup a development process based on the *MathWorks DO Qualification Workflow* [32] for applications up to DAL B. The workflow proposes to use *Simulink* and *Stateflow* to implement the Design Model, replacing Software Low-level Requirements as well as Software Architecture from the conventional DO-178C process. The approach coincides with DO-331 MB.1.6.3 Example 1 and is also presented in [4, 17].

C Source Code is directly generated from the Design Model using *Embedded Coder* (EC). Since the code generator is not shipped with a Tool Qualification Kit, *MathWorks* provides *Simulink Code Inspector* to automate code review and verify compliance of the Source Code with the Design Model as well as traceability. Using SLCI restricts features of *Simulink* and *Stateflow* to a robust and safe subset.

The workflow and tool chain at the institute subsequent to the generation of Source Code is described in [10]. The interplay with a system design process for control algorithm development is outlined in [11].

Using a Design Model and directly generating code out of it brings the DO-178C Design Process closer to the Coding Process. The work to be done in the Coding Process reduces to a single function call of the coder. The fulfillment of Source Code objectives from Table MB.A-5 now can only be influenced by the Design Model. The *Simulink* and *Stateflow* models thus have to address rules from a Software Design Standard, a Model Standard but also a Code Standard.

3 Hardware and Software Context

3.1 Flight Control Computer Hardware

The addressed controller runs on a single flight control computer (FCC). This is possible due to the system architecture of the research aircraft and the safety concept of the experimental flight control system [1].

The flight control computer has been developed together with industry partners regarding certification aspects.[1] The Power PC 32-bit architecture has a clock speed of 533 MHz with a double-precision floating point unit and works in Big-Endian mode. It also has a high-speed interface to two Cortex M-I/O-processors, providing various UART, ARINC-825, and ARINC-429 interfaces along with multiple discrete IOs.

[1] http://www.fsd.mw.tum.de/infrastructure/gnc-subsystems/ [Cited on 6 January 2017].

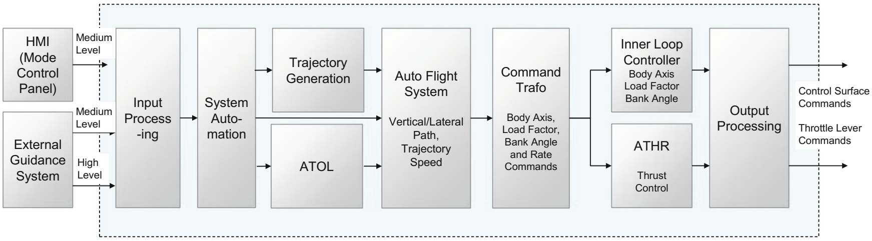


Fig. 1 Structure of the modular integrated flight control system

3.2 *Controller Design Considerations*

3.2.1 Controller Architecture

Figure 1 shows a simplified version of the cascaded controller and the identified software modules. Input and output processing modules contain reorganization of the incoming and outgoing data. Core of the algorithm is an aircraft dependent baseline innerloop controller, whose input commands are the load factors. The auto flight system contains a trajectory/path controller and a controller for well-known autopilot functionalities like attitude, altitude, heading, or speed hold. Further information about the auto flight module and the integrated trajectory/path controller is provided by [14, 15, 22]. For high-level commands, e.g., waypoints, the controllable path is calculated by the trajectory generation module as described in [23, 25, 26]. Additionally, there exists a module for automatic landing and take off (ATOL). All mode switching as well as the startup behavior is implemented in the system automation module as outlined in [16].

3.2.2 Software Modules and Interfaces

The controller is concurrently developed by various developers that are working on and are responsible for clearly defined software modules. A software module consists of a set of Simulink Models and dependent data passing the software lifecycle together. A software module has an own revision index, a defined interface and separately allocated requirements. The schematic draft of the system in Fig. 1 also outlines the most important software modules.

3.2.3 Floating-Point Arithmetic

Since the FCC has a double-precision floating point unit, this arithmetic is preferred. Although real necessity for double-precision is only given when dealing with WGS-84 waypoint positions, it was decided to honor simplicity and perform all calculations with double-precision.

Understanding generated shared utility functions (see Sect. 4.2.4) and treatment of special floating point quantities (NaN, Inf,...) requires a deeper look into floating point arithmetic. The FCC partly supports floating point arithmetic standardized by IEEE-754:2008 [12]. Here, the *binary64* format with base two, an exponent length of ten bytes and a mantissa length of 52 bytes (plus one bit for the sign) is used. This complies with the internal floating point representation and arithmetic of *MATLAB*.[2]

The treatment of special floating point quantities is shortly addressed in the following section. The internal floating point exception handling as specified in IEEE-754 is not considered.

3.2.4 Interface to C Framework

The model-based controller is embedded in a conventionally developed C framework. The top-level model of the controller application, as simplified in Fig. 2, provides one in- and outport for every incoming and every outgoing physical interface of the FCC. The port data types are designed as Simulink Bus objects, which are translated to C structures during code generation. The coder settings therefor are described in Sect. 4.2.1.

The input and output buses have sub-buses for every readable/writable message, which itself contain elements for every message payload parameter. Additionally, every message bus has an update indication, notifying that a new message has been received, or a message as to be sent, respectively.

The execution of the whole software is as follows: Every iteration the surrounding C-Framework decodes incoming messages and copies the received data into the exposed C structures of the generated code. After that, it calls the step function of the application. As soon as the step function has finished, the framework copies the data from the outgoing C structures into the messages to be sent.

It is also visible in Fig. 2 that the incoming data is rearranged into functional structures in system "datain" at first (e.g., into `sensors_in`). This system additionally performs basic input monitoring like saturation to a specific range. From this point on, no special floating point quantity shall be in the software any more and no subsequent software module shall introduce such.

4 Modeling Environment for Safety-Critical Software with Simulink and Stateflow

This chapter summarizes the most important aspects, which a consistent Modeling Environment for *MATLAB Release 2016b* must fulfill to create safety-critical software. For other releases, variations are possible. The Modeling Environment is

[2] http://de.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html [Cited on 09/05/2016].

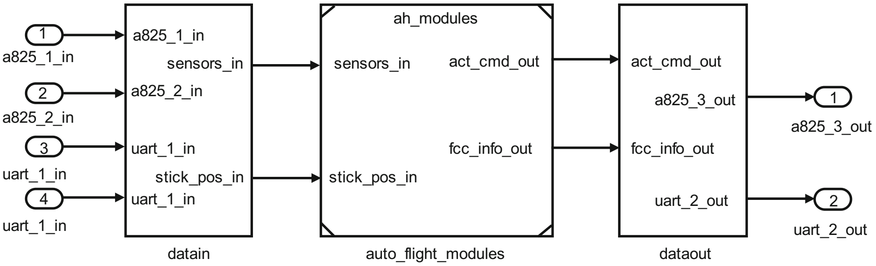


Fig. 2 Schematic draft of the top-level controller model

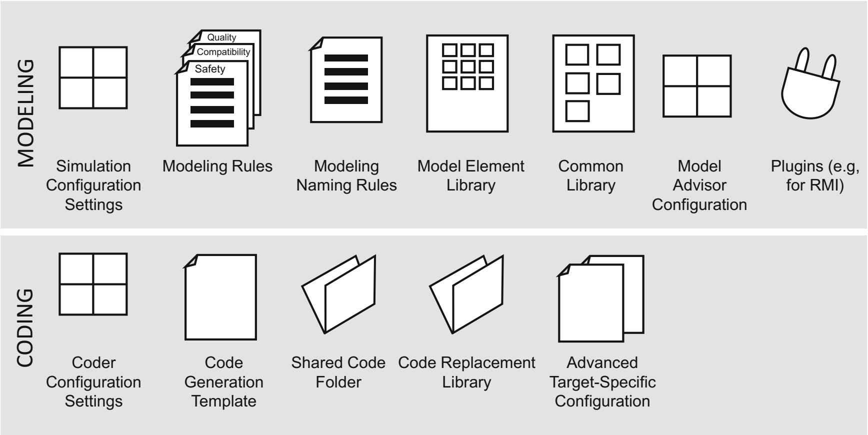


Fig. 3 Modeling Environment provided to developers at the *Institute of Flight System Dynamics*. The whole package is under version control and its content must be read or installed by the developers in Simulink prior to implementation

defined during the software planning process and mainly documented in the Software Model Standard (DO-331 MB.11.23 “Model Standards”) with respect to

- existing modeling rules,
- standards for generated code,
- tools used in the process,
- strategies for integration, and
- planned target hardware.

The Modeling Environment used at the institute is depicted in Fig. 3. For this paper, the Modeling Environment is divided into a modeling and coding part. In reality, the border is indistinct. For example, *Simulink* uses a single configuration file/object, mixing options for simulation and for code generation with EC. This is similar for modeling rules that influence the model, but with respect to the generated code.

Section 4.1 introduces the part of the Modeling Environment that primarily impacts the Design Model in its function as Software Low-Level Requirements and Software Architecture. Section 4.2 focuses on details relevant for code generation.

4.1 Environment for Design and Simulation

4.1.1 Modeling Rules

Basis for the finally applied set of rules in the presented application were the *MAAB Control Algorithm Modeling Guidelines* [27], the *MathWorks High-Integrity System Modeling Guidelines* [30], and the *MathWorks Code Generation Guidelines* [29].

This basic set was reduced by removing duplicates resulting from the independent historical origin of the rules. Additionally, restrictions imposed by subsequent tools, especially SLCI, allowed the exclusion of further guidelines (e.g., since no *Embedded MATLAB Charts* and *Functions* are supported, all rules on *MATLAB* code could be excluded). Instead and supplemental to this basic collection, custom rules have been defined. Table 1 lists a summary of the applicable modeling rules at the institute. In some cases, the custom rules replace or overwrite parts of the *MAAB* and *MathWorks* rules. Then, the respective, underlying rules have been marked as not applicable and were redefined.

Not included in Table 1 are rules concerning compatibility. Compatibility rules e.g., for SLCI are documented in the corresponding tool operational requirements. In the case of SLCI, most of the compatibility rules are also shipped with checks.

During the verification process, the rules may either be reviewed or automatically checked using *Simulink Model Advisor*. The collection of Model Advisor checks is provided to the developer within the Modeling Environment as “Model Advisor Configuration” to allow continuous checking before the subsequent verification process is triggered.

A closer look at the set of applicable rules reveals, that not all of them have the same criticality. Rules addressing model appearance are for example necessary to achieve readable models, but do not impact the generated code. In contrast, an integer overflow behavior setting is of much higher importance. Thus, the rules are categorized into three groups:

Table 1 Composition of modeling rules

Source	# Rules original	# Applicable[a]
MAAB (v. 3.0)	109	54
MW HI (R2016b)	101	72
MW CG (R2016b)	12	5
Custom	51	51

[a] For projects at the *Institute of Flight System Dynamics*

Safety-critical Rules These rules describe model settings, architecture, and patterns that significantly contribute to safe code. If these rules are not respected, it is possible that errors are introduced in the code, which are hard to find in subsequent verification processes. Additionally, rules are contained that play an important role in the process and would require a significant restructuring and redefinition. Related Model Advisor checks of this group require a tool qualification, and a failure or warning is not accepted and not justifiable.

Compatibility Rules These rules ensure compatibility with subsequent tools and the process. Incompatibility introduced by the model developer is reliably detected by the affected qualified tools, thus these rules have no direct relevance for safety. However, when incompatibility is discovered late, this may lead to significant rework. This category of rules requires no tool qualification for the checks, a failure is not accepted, but a warning may be formally justified.

Quality Rules Under this category, all non-critical rules are summarized. They do not have a direct impact on safety, or their impact is analyzed in subsequent steps with qualified tools. The rules address readability, maintainability, appearance and comfort settings. A single violation is not critical, but has impact on the quality of the model. Related Model Advisor checks shall not fail, but a warning may be acceptable without formal justification. Quality rules may be summarized to a quality index. Then, an acceptable quality range may be defined and checked. Quality metrics and their automated evaluation are a rising topic for models [24].

4.1.2 Modeling Naming Rules

Beside modeling rules, modeling naming rules have been established. To avoid overloading of the modeling rules, they were defined in a separate document. The naming rules give syntax and semantics for names of files and folders, workspace objects, as well as blocks and signals belonging to the Design Model. The work required for defining proper naming conventions should not be underestimated due to the direct impact of the model on code and the high number of different files and workspace objects to be covered.

The purpose of naming conventions is to support

Compliance to Modeling Rules As described above, the developed set of modeling rules bases on *MAAB*, *MathWorks High Integrity*, and *MathWorks Code Generation Guidelines*. At least the first two define naming rules, which have to be respected.

Integration of Software Modules Since the whole application is split into different software modules, naming conventions must ensure that integration of models and code is possible without conflicts.

For example, all software modules at the institute currently initialize their objects into the base workspace, since SLCI had an incompatibility with *Data Dictionaries* for a long time. Since the model workspace did not support Simulink.Bus objects or Simulink.Parameter and Simulink.Signal objects with explicitly set storage class, the shared base workspace was the only remaining solution. Naming conventions ensure that the software modules do not conflict and overwrite data when loaded together. This special issue was resolved by introducing a short identifier for every separately developed module prepended to all data initialized in the base workspace (e.g. `af_sensor_data_Bus` with prefix `af` for the autopilot module).

In the Source Code, the uniqueness of global identifiers and file names across software modules is directly influenced by naming conventions of the model (e.g. naming of an exported header file). Naming conflicts lead either to mangling in the code generation process, or errors later on during compilation.

Compliance to Coding Rules Coding rules like MISRA C:2012 [33] include guidance on naming, e.g. enforcing uniqueness of identifiers within a specified range of characters. Due to automatic code generation, the naming of workspace objects is the only way to influence MISRA:2012 compliance of the generated code beside tuning the coder settings.

A typical issue is that the coder adds prefixes to identifiers, which leads to longer identifiers in the generated code and may result in violation of MISRA C:2012 rule 5.1 requiring a distinct external identifier within the first 31 characters for C99 (due to this definition, postfixes are in general no issue for MISRA compliance). A bus `sensor_data_Bus` in a model called `flight_control` may introduce the external identifier `flight_ control_rtZsensor_data_Bus`. In general, EC can automatically limit identifier length. This however reduces readability and has limitations.[3] To avoid a violation, both restrictions on the model and bus name length are required. Helpful for figuring out limitations of the coder is also the *MathWorks Embedded Coder Compliance Report* [31].

In newer releases of EC, the treatment of pre- and postfixes has been significantly improved. In some cases, if the length of the generated identifier exceeds the maximum identifier length specified in the coder settings, pre- and postfixes are removed before the original name is shortened.

Readability of Models Workspace object names should have a clear syntax and already provide some information about their type and physical meaning. Best example is the label of a signal line, which may provide a hint of its boolean data type by the postfix `_flg`. However, more important for a controller application is that it states its physical meaning in a clear way. A major part of the modeling naming rules is thus defining the naming of coordinate systems, angles, transformation matrices (e.g., `M_O_B` for M_{OB} as transformation from body-fixed to the north-east-down system), units (e.g., `mDs2` for m/s^2) or the translation of mathematical characters (e.g., `mu` for μ or `x_ddot` for $\ddot{x}$).

Readability of Code As already mentioned, coding standards introduce naming restrictions to ensure compliance with a majority of compilers. Code generators like EC provide functionality to enforce this compliance. However, most of these automatic procedures reduce readability and their usage should be avoided if possible

[3] Limitations on controlling the identifier format are documented in the *Embedded Coder User's Guide* [28] pp. 36–33 "Identifier Format Control Parameters Limitations".

by designing the model correctly. For example, a concept to enforce uniqueness of identifiers is mangling, meaning that similar identifiers are shortened and an arbitrary but unique sequence of characters is appended. For example, when using *Virtual Subsystems*, two similarly named *Unit Delay* blocks may be placed in the same Source Code scope. The created global C variables of two blocks called `int_unit_delay` with a maximum identifier length of 15 (exemplary) and mangling length of 4 would be `int_unit_delay` and `int_unit__h4pj`, which reduces readability and impedes debugging as well as verification later on.

4.1.3 Traceability Plugin

One solution used at the institute to document System Requirements and Software High-Level Requirements is *Polarion REQUIREMENTS*.[4] The application is web-based and runs on a web server. On model-side, the *MathWorks* toolbox *Simulink Verification and Validation*[5] provides the Requirement Management Interface (RMI) to annotate model elements and establish a link to any requirement management tool that provides a corresponding plugin. The plugin for connecting *Polarion* and the RMI is provided by the *Polarion Connector for Simulink*.[6] The plugin must be registered in *Simulink* and is thus included in the Modeling Environment, too.

4.1.4 Model Element Library

The model element library contains atomic blocks and small subsystems, which do not generate separate functions, but are inlined in the code of the higher model level. These subsystems have to be small enough to be testable in the model and C-function, in which they are embedded. The usable atomic blocks are mainly defined by compatibility with SLCI, which comes along with the `slcilib` library. This basic block set was modified to additionally comply with the applicable modeling rules.

4.1.5 Simulation Configuration Settings

The configuration settings mainly relevant for simulation are solver, optimization and diagnostic settings. Most of them are given by the guidelines from *MathWorks*, *MAAB* or SLCI. Only a few remain eligible.

[4] https://polarion.plm.automation.siemens.com/products/polarion-requirements [Cited on 6 January 2017].

[5] http://de.mathworks.com/products/simverification/ [Cited on 6 January 2017].

[6] Polarion Connector for Simulink, see http://extensions.polarion.com/extensions/173-polarion-connector-for-simulink [Cited on 6 January 2017].

Table 2 Model configuration settings

Name	Description
Top-level Model (Coding)	Configuration for the top-level model generating an interface of type void-void (see also Sect. 4.2.1)
Reusable Model (Coding)	A reusable model reference can be used for multiple instances. However, only signals with custom storage class "Simulink Global" are allowed (see also Sect. 4.2.1)
Singleton Model (Coding)	A model reference that can only be embedded once. This model can include "Exported Global" signals (see also Sect. 4.2.1)
Test Harness (Simulation)	Since the test harness encapsulates the whole model-in-the-loop simulation with e.g., the flight dynamic model or actuator models, its configuration has to differ slightly to be compatible

For the presented controller, the solver settings restrict execution to "single rate" and with a discrete fixed-step solver. For diagnostics, conservative settings are chosen. Optimization settings mainly affect the generated code and are thus discussed in the next section.

Although referred to as a single configuration set in here, there are normally multiple model configurations. Currently, four slightly different model configuration settings are used in the presented Modeling Environment as listed in Table 2.

Model configuration settings are set in models as "Configuration References". This increases maintainability and consistency over all software modules.

4.1.6 Common Library Module

Beside the model element library, a so-called *Common Library* is provided to the developer. This library contains subsystems that are too large to be inlined and are thus reusable model references. These libraries are considered a separate software module, and follow an own software verification process.

Content of the library are subsystems adding robustness to critical function calls, like protected divisions, square roots or integrators protected against windup.

4.2 Environment for Code Generation

The generated Source Code shall comply with ANSI ISO/IEC 9899:1999 [13] and satisfy the rules and directives for autogenerated code (Appendix E) of MISRA C:2012 [33].

As already mentioned, EC is used for code generation. The coder provides hundreds of settings and customization possibilities. The following sections demonstrate the minimum of customization that is required from the viewpoint of a DO-331 software development process. In general, it is advisable to avoid extensive customization, since any deviation from the default configuration may have an impact on tool compatibility and robustness [3].

4.2.1 Coder Configuration Settings

Coder configuration settings are manifold and they are defined as modeling rules, since the border between simulation and coder settings is blurry. Many simulation settings are directly used by the coder (for example the single- or multi-rate or most of the optimization settings).

Code Interface The code interface differs depending on the model. For the top-level model, whose functions are externally called by the surrounding C framework, a simple non-reusable interface of type void-void is generated. This creates structures for input and output, as well as void-void `initialize` and `step` functions. For nested models, either a reusable or a non-reusable interface can be chosen, depending on whether multiple or only a single instance shall be allowed. The code of non-reusable models may be easier to test separately, since signals can be exposed as exported global variables. All functions of nested models pass parameters as individual arguments to avoid the overhead of copying the input variables into a single structure.

Code Packaging Code packaging is set for modularity. However, modular code generation cannot be achieved solely by coder settings. The right usage of built-in custom storage classes is another important point.

Code Style For the code style, a nominal parenthesis level is used as well as nominal casting. Note that “Standard compliant” casting is not yet compatible with SLCI. Comments must be adjusted to provide sufficient traceability information in the code. The identifier control was set to standard values.

Support of Functionality Code is generated for a single rate system. Since a floating point unit is available on the target computer, floating point arithmetic is activated. Further features like complex numbers, infinite numbers or absolute time are disabled, since they are neither required nor would they solely be checkable with SLCI due to the large amount of additional code.

Code Optimization To allow verification with SLCI, advanced code optimizations are disabled by setting `AdvancedOptControl` to `-SLCI`. The optimization group contains the powerful feature “Signal Storage Reuse”. Although SLCI is compatible with all of the nested settings, their usage must be analyzed carefully. “Signal Storage Reuse” on the one hand can drastically improve the performance of the generated code and reduce the memory requirements, but on the other hand it can significantly reduce readability and robustness concerning change. For example, a locally reused variable may be named according to its first occurrence. In other contexts, in which it is reused, this name may be misleading and may not fit to the actual meaning of the stored value. Furthermore, if the first occurrence changes its variable name, changes all over the generated code will appear. The actual fix around the first occurrence is hard to find and verify in the code.

4.2.2 Modeling Rules Concerning Coding

The structure of the code is not only influenced by the configuration settings, but also by block or signal line settings.

Input and Output Data types of Atomic Blocks Atomic blocks often allow an implicit type conversion. Implicit type conversions affect readability and should be avoided. The data type should be preserved. The block "Sum of elements" for example allows setting an accumulator data type as well as an output data type. Including the input data type, a conversion between three data types is necessary if set differently. Although supported by EC, this is a SLCI incompatibility in most cases.

Integer Saturation on Overflow Blocks for integer calculations typically provide the option to generate code that saturates on integer overflow. This produces significant extra code and is not verifiable by SLCI. Integer saturation on overflow should thus be deactivated in the block and - if required - modeled separately. Exception of the rule is the Abs block, where hisl_0001 of the High Integrity Guidelines explicitly advises to set the option (due to the differing positive and negative ranges of signed integers). It was observed, that developers are normally not aware of this exception, since they take preset library blocks. The result was dead code for every Abs block in early verification, since they added their own protections.

Subsystem Settings Simulink knows various kinds of subsystems, basically virtual and non-virtual (atomic) subsystems. Virtual subsystems can be considered as visual help. The coder eliminates them. By setting a subsystem to "atomic", the coder is forced to keep the generated code of the subsystem together. Further options allow to specify, how this is achieved, either by a function (`reusable` or `non-reusable`) or by a packaged block of code (`inlined`). Up to release 2016a, SLCI only supported the `inlined` option. Structuring code by functions, which is necessary for testing, was only possible by model references. With Release 2016b, `non-reusable` functions are verifiable, too.

Simulink Data Objects By explicitly specifying the storage class of Simulink Data Objects[7] (Simulink.Bus, Simulink.Signal and Simulink.Parameter) and, for example, their header or source file name, the behavior of the code generator can be influenced.

[7] http://de.mathworks.com/help/simulink/ug/working-with-data-objects.html [Cited on 6 January 2017].

Simulink.Parameter objects should be used for non-scalar or structured parameters to prevent that the coder generates a common data pool [9]. Simulink.Signals for example may be used to expose signals for global access (for testing) or to define reused signals and thus prevent data copies.

4.2.3 Customization of the Coding Process and Advanced Target-Specific Configuration

The coding process, controlled by Target Language Compiler (TLC) files, has not been modified for the presented controller. Slightly customized is only the Code Generation Template (CGT), which allows formatting the high-level organization of the code. The general arrangement of the code remained, only the comment fields were updated to include further necessary information and exclude the timestamp (to ease comparison of revisions with Diff tools and identify real changes). Additionally, a data alignment specification is registered to include alignment information for the compiler.

4.2.4 Replacement of Shared Utility Functions

When generating C-code from a Simulink model, some atomic blocks expand to complex functions. These functions are placed in a shared location, so they are only generated once for all models. They have canonical function and file names differing for every combination of settings in the block mask.

For these shared C functions, the following aspects have to be considered:

1. In the presented case, the Design Model replaces Software Low-Level and Software Architecture. Therefore, the Design Model must be sufficiently descriptive and granular. If an atomic block produces complex code, this argumentation does not hold any more.
2. Requirements cannot be linked into shared utility C-code, since these functions are regenerated every build.
3. SLCI just verifies the call, but not the correctness of the function itself.
4. Shared utility function complicate configuration management if their generation is not controlled. Although the functions have canonical names, the number of C functions, that can be created, is immense, even with a reduced block set (thousands). Limiting the shared utility functions to a dedicated subset of functions is task of the modeling rules, which have to restrict the allowed atomic block settings.

As a consequence of points 1 to 3, it is desirable to replace the generated shared functions by C functions developed and verified along a conventional DO-178C software development process. One way is using so-called *Code Replacement Libraries*. They allow replacing specified function calls in the generated C-code by custom calls. This avoids the generation of the shared utilities.

Not yet solved with this solution is point 4. If the modeling rules are not restrictive enough, shared code beyond the replaced function may be generated. A repeated review for unsupported, additionally generated functions is necessary after building the code.

With Release 2016b, the new EC setting ExistingSharedCode was introduced, which allows the specification of an already existing shared code directory. EC then scans the specified directory for the functions it requires during code generation and copies the files to the new shared folder. Identified are the files by their canonical filename. Further, setting the option UseOnlyExistingSharedCode allows to automatically abort the coding process, if not all required files are found. Thus, the developer gets direct feedback, whether the model generates unsupported shared functions or not.

For the presented controller, only four auto-generated functions are required. Reason is, that SLCI limits the options e.g., for Lookup Tables and their dimensions. Additionally, the integrated integer overflow protection functionality is disabled for all blocks except the Abs Block (see Sect. 4.2.2).

- rt_roundd: Although the ISO C standard [13] defines the round function that rounds halfway cases away from zero regardless the floating point rounding direction, EC generates rt_roundd, which only bases on ceil and floor functions. Additionally, it prevents rounding if the units in the last place (ULPs) are equal or greater than 1.0. The rt_roundd function has probably been introduced to also ensure calculation equality between simulation and code for non-standard compliant C libraries.
- rt_modd: This is a protected version and safe implementation of the double-precision floating point modulo operation generated by the Mathematical Function block. The function is implemented to preserve simulation and code equality for non-standard compliant C math libraries. If the typically used fmod function shall be called directly, the Rem block setting has to be used. However, even the C standard does not define the behavior for a divisor equal to zero for this function.
- look1_binlca and look2_binlca: Functions originating from Lookup Table blocks with double-precision input, output and accumulator data type, binary search and linear interpolation. Extrapolation method is "cut" and out-of-range protection is not removed.

4.2.5 Replacement of Compiler-Supplied Libraries

Beside the shared functions, the controller code generated with EC addresses additional standard C functions like sin or memcpy. Although these basic functions are normally supplied with the compiler, they have to be verified separately and must satisfy DO-178C objectives [2].

Depending on the used libraries, a *Code Replacement Library* must be registered in Simulink before generating code, to replace standard C function calls (e.g., a sin(...) to cert_sin(...)).

5 Summary

Using *Simulink* and *Stateflow* for implementing a Design Model is not an out-of-the box solution, but requires a planned setup and preparation of the tool. In this paper, aspects of a consistent Modeling Environment were presented in the light of the used hardware, application design considerations, and the chosen model-based development process.

The paper structured the Modeling Environment in two parts, one containing the relevant components concerning the role of the Design Model as Software Design and Software Architecture, and a second part with focus on code generation. For the first part, an overview of the applicable rules was given and model libraries as well as a important configuration settings have been highlighted. The code generation part pointed out the major settings relevant for EC and strategies to deal with code libraries.

The details in this paper should not be seen as the ultimate solution for every company and workflow due to the varying context, but the key aspects can be considered in every safety-critical project.

The close connection between Design Model and Source Code in a code-generation based process emphasizes the importance of presciently planned rules, restrictions, and settings for *MATLAB*, *Simulink*, and *Stateflow*, since any setting in the Design Model may directly impact the Source Code. Making the Modeling Environment consistent may require an iterative process and experience.

The discussion also showed the central role of SLCI. This tool should not only be seen as a sole verification tool since it also plays an important part in defining a safe and robust subset of *MATLAB*, *Simulink*, and *Stateflow* features for safety-critical applications.

The outlined Modeling Environment is a snapshot of the effort taken at the *Institute of System Dynamics*. Further investigation is planned on modular code generation, and optimal settings for improved compatibility with 3rd party tools used along the workflow, like WCET analyzers or special compilers, all coming along with additional restrictions.

References

1. Braun B, Philip S, Peter L, Dambeck J, Holzapfel F (2013) Multi-purpose flying sensor testbed: AIRTEC 2013 aerospace sensors/aerospace testing. Frankfurt a. M., 6 November 2013
2. Certification Authorities Software Team (2004) Position paper CAST-21 - compiler-supplied libraries, January 2004
3. Dillaber E, Kendrick L, Jin W, Reddy V (eds) (2010) Pragmatic strategies for adopting model-based design for embedded applications. SAE Int
4. Erkinnen T, Potter B (2009) Model-based design for DO-178B with qualified tools: AIAA modeling and simulation technologies conference and exhibit. American Institute of Aeronautics and Astronautics Inc, Hyatt Regency McCormick Place, Chicago Illinois

5. Esterel Technologies SA (2015) Efficient development of safe avionics software with DO-178C objectives using SCADE suite: methodology handbook, June 2015
6. Estrada RG, Sasaki G, Dillaber E (2013) Best practices for developing DO-178 compliant software using model-based design. AIAA Infotech@Aerospace. Boston. https://doi.org/10.2514/6.2013-4566
7. European Aviation Safety Agency EASA: certification specifications for normal, utility, aerobatic and commuter category aeroplanes: CS-23 Amendment 3
8. Federal Aviation Administration FAA (2011) System safety analysis and assessment for part 23 airplanes (AC 23.1309-1E) 17 November 2011
9. Hochstrasser M, Hornauer M, Holzapfel F (2016) Formal verification of flight control applications along a model-based development process: a case study. In: DGLR Workshop - Software Safety. München, 05 October 2016. http://www.dglr.de/fileadmin/inhalte/dglr/fb/q3/veranstaltungen/L63_Q34_2016_Software_Safety/2016_DGLR_Workshop_TUM_samoconsult.pdf
10. Hornauer M, Holzapfel F (2011) Model based testing for CS-23 avionic and UAV applications: DGLR workshop 2011. In: DGLR Workshop - Verifikation in der modellbasierten Software-Entwicklung, München
11. Hornauer M, Schuck F, Holzapfel F (2013) Wechselwirkungen zwischen GNC algorithmus und software. In: DGLR Workshop - Durchgängige Entwicklung von GNC Funktionen - vom Algorithmus zur Embedded Software. München
12. IEEE Computer Society (2008) IEEE standard for floating point arithmetic (IEEE 754-2008), August 2008
13. ISO/IEC (1999) Programming languages C - 2nd edn, December 1999
14. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: IEEE control systems society (ed) proceedings of 14th international conference on control, automation, robotics and vision
15. Karlsson E, Schatz SP, Baier T, Dörhöfer C, Gabrys A, Hochstrasser M, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Schneider V, Philip S, Steinert L, Zollitsch AW, Holzapfel F (2016) Automatic flight path control of an experimental DA42 general aviation aircraft. In: IEEE control systems society (ed.) proceedings of 14th international conference on control, automation, robotics and vision
16. Krause C, Holzapfel F (2016) Designing a system automation for a novel UAV demonstrator. In: IEEE control systems society (ed.) proceedings of 14th international conference on control, automation, robotics and vision
17. Potter B (2012) Complying with DO-178C and DO-331 using model-based design
18. RTCA (2000) DO-254 - design assurance guidance for airborne electronic hardware
19. RTCA (2011) DO-178C - software considerations in airborne systems and equipment certification
20. RTCA (2011) DO-331 - model-based development and verification supplement to DO-178C and DO-278A
21. RTCA (2011) DO-333 formal methods supplement to DO-178C and DO-278A
22. Schatz SP, Holzapfel F (2014) Modular trajectory/path following controller using nonlinear error dynamics. In: 2014 IEEE international aerospace electronics and remote sensing technology (ICARES), pp. 157–163. IEEE. https://doi.org/10.1109/ICARES.2014.7024374
23. Schatz SP, Schneider V, Karlsson E, Holzapfel F, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Zollitsch AW (2016) Flightplan flight tests of an experimental DA42 generation aviation aircraft. In: IEEE control systems society (ed.) proceedings of 14th international conference on control, automation, robotics and vision
24. Scheible J (2012) Automatisierte qualitätsbewertung am beispiel von matlab simulink-modellen in der automobil-domäne. Dissertation, Eberhard Karls Universität Tübingen, Tübingen. https://publikationen.uni-tuebingen.de/xmlui/handle/10900/49708
25. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2014 IEEE international aerospace electronics and remote sensing technology (ICARES). IEEE

26. Schneider V, Piprek P, Schatz SP, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Karlsson E, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Holzapfel F (2016) Online trajectory generation using clothoid segments. In: IEEE control systems society (ed) proceedings of 14th international conference on control, automation, robotics and vision
27. The MathWorks automotive advisory board: MathWorks automotive advisory board control algorithm modeling guidelines using MATLAB, Simulink, and Stateflow: R2016b
28. The MathWorks Inc. Embedded coder user's guide: R2016b
29. The MathWorks Inc. Guidelines and factors to consider for code generation: R2016b
30. The MathWorks Inc. Modeling guidelines for high-integrity systems: R2016b
31. The MathWorks Inc. (2014) Embedded coder R2014b - MISRA AC AGC compliance considerations
32. The MathWorks Inc. (2016) DO qualification R2016b: model-based design workflow for DO-178C
33. The Motor Industry Software Reliability Association (2013) MISRA-C:2012 - Guidelines for the use of C language in critical systems, March 2013

Robust Incremental Nonlinear Dynamic Inversion Controller of Hexapod Flight Simulator Motion System

Yingzhi Huang, D.M. Pool, O. Stroosma and Qiping Chu

1 Introduction

The Stewart platforms, also known as hexapod systems, are most commonly used as motion bases for modern flight simulators. In this configuration, a moving upper platform is driven by six linear actuators connected to the stationary base to provide motion in 6 degree-of-freedom(DOF), as shown in Fig. 1. Commonly an Universal-Prismatic-Spherical(UPS) structure is designed in such systems, in which the lower and upper platforms are connected to the legs through passive universal joints and spherical joints and the two parts of each leg are connected by a prismatic joint where electrical or hydraulic actuation force is acted on. The parallel structure provide higher rigidity and accuracy compared with its serial robot counterparts. A typical example of such a system is the SIMONA research simulator (SRS) in Delft University of Technology as shown in Fig. 1 [18].

The main goal of the flight simulator motion systems is to track the reference motion calculated by aircraft aerodynamics model and pilot input as accurate as possible. Different from industrial applications, the pilot-in-the-loop flight simulators require high performance motion systems to enhance the fidelity of simulation flight. Typical PID controller is hardly eligible for the high performance requirement due

Y. Huang (✉) · D.M. Pool · O. Stroosma · Q. Chu
Control and Simulation Section, Faculty of Aerospace Engineering,
Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands
e-mail: Y.Huang-2@tudelft.nl

D.M. Pool
e-mail: D.M.Pool@tudelft.nl

O. Stroosma
e-mail: O.Stroosma@tudelft.nl

Q. Chu
e-mail: Q.P.Chu@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_5

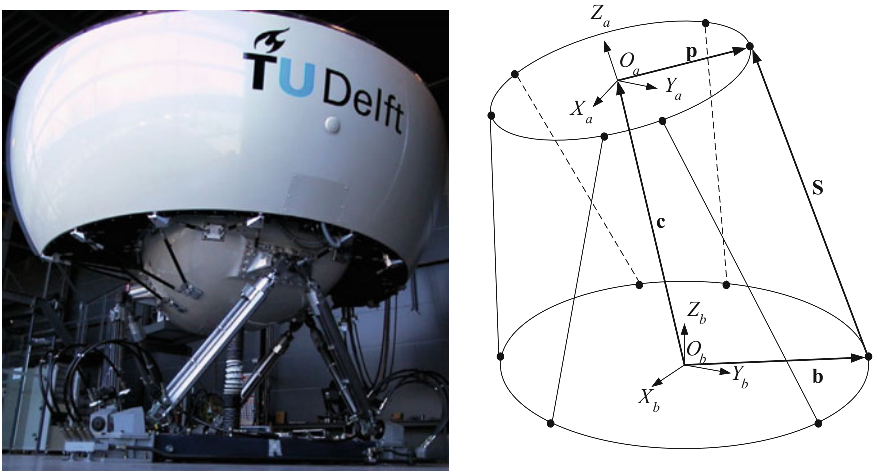


Fig. 1 A Stewart platform based motion system SRS at TU Delft and it's schematic drawing [6]

to the highly nonlinear dynamics of the parallel manipulator. Thus more advanced control techniques have been studied intensively [5].

Nonlinear dynamics inversion [9, 13] (NDI) is capable of exactly linearise the nonlinear dynamics and is also referred to as computed torque control [12] in the field of robot control. However, it's well known that NDI is seriously dependent on the accuracy of the model that for a complex flight simulator system, any model parametric uncertainty, unmodelled dynamics and disturbances such as joint friction will significantly degrade the performance [1]. Thus NDI is rarely used directly in parallel robot motion control. Adaptive control is a possible technique to deal with parametric uncertainties and has been proposed for similar applications [4, 11, 12]. Nevertheless, the computational burden of adaption motion controller based on a well modeled system is significant that real-time application is hard to achieve. Robust control which is also capable of overcoming model uncertainties, is studied by only a few papers addressing the studied control system [1, 7, 8] and a sliding mode methodology is commonly used that may lead to chattering problem.

The Incremental Nonlinear Dynamic Inversion (INDI) is a sensor-based control technique for nonlinear systems which is much more robust than NDI [14, 15]. By computing the incremental input commands using the acceleration measurement instead of calculating the total inputs, the controller need less model information and is not sensitive to model mismatches while proving a high performance. These features are useful for a parallel flight simulator which suffers from model uncertainties or even load-varying mechanics due to change of pilots.

In this paper the application of INDI methodology to a 6-DOF hexapod flight simulator outer-loop motion control is presented, parametric uncertainties and unmodelled dynamics are introduced to model dynamics to simplify the controller design and reduce computational burden. This is the first time that INDI is proved to be suitable for a parallel robot motion control, and the fact that an important assumption, i.e. time scale separation principle, holds for the dynamics of such systems. This work hires a detailed flight simulator model with consideration of actuator inertial and joint friction based on SIMONA Research Simulator in Delft University of Technology. The actuator dynamics are not considered in order to focus on the INDI controller performance on the parallel robotic dynamics. In future work with practical implementation of proposed control, the influences of actuator dynamics will be included and considered.

This paper is organized as follows. In Sect. 2, the kinematics and dynamics of the 6-DOF hexapod motion system are briefly derived. Section 3 presents the motion control system design based on INDI technique in detail and simulation results are given in Sect. 4. The main conclusions are summarized in Sect. 5.

2 System Kinematics and Dynamics

A schematic drawing of a hexapod motion system is shown in Fig. 1. The Newton–Euler approach [2, 3] is used to derive the system dynamic equations in Cartesian space. The platform position and velocity are defined as

$$\begin{aligned} \mathbf{sx} &= \left[\mathbf{c}^T, \boldsymbol{\Phi}^T\right]^T \\ \dot{\mathbf{x}} &= \left[\dot{\mathbf{c}}^T, \boldsymbol{\omega}_p^T\right]^T \end{aligned} \tag{1}$$

where $\mathbf{c}$ denotes the translation vector of the upper platform in inertial frame E_b, $\boldsymbol{\Phi}$ is the Euler angles between the body fixed frameE_a and E_b and $\boldsymbol{\omega}_p$ is the angular velocity of the upper platform.

Different from serial manipulators, the inverse kinematics of parallel robots are very simple, which can be given by calculating the leg vector $\mathbf{S}$ in Fig. 1 as

$$\begin{aligned} \mathbf{S} &= \mathbf{c} + T_{ba}\mathbf{p} - \mathbf{b} \\ &= \mathbf{c} + \mathbf{q}_p - \mathbf{b} \end{aligned} \tag{2}$$

where $\mathbf{p}$ and $\mathbf{q}_p$ are upper universal joint locations in reference frames E_a and E_b respectively, and T_{ba} the rotational matrix between them.

In terms of the platform dynamics, the dynamics of a single leg is derived in literature [3] by combining Euler's equation of the entire leg and Newton's equation of the moving rod as

$$-\mathbf{F}_s = \mathbf{Q}\ddot{\mathbf{c}} - \mathbf{Q}\tilde{\mathbf{q}}_p\dot{\boldsymbol{\omega}}_p + \mathbf{V} - F\mathbf{s} \tag{3}$$

where $\mathbf{F}_s$ is the force vector acting on the upper universal joint, F is the actuation force generated on the prismatic joint, $\mathbf{s}$ is the unit vector to $\mathbf{S}$, and $\mathbf{Q}$ and $\mathbf{V}$ are matrix depending on leg inertial and dynamics properties.

Similarly, the dynamics of the upper moving platform is described by writing the Newton and Euler's equation as

$$\sum_{n=1}^{6} (\mathbf{F}_s)_i + M_p \mathbf{g} = M_p \mathbf{a}_p \tag{4}$$

and

$$\sum_{n=1}^{6} \left(\mathbf{q}_p \times \mathbf{F}_s\right)_i + M_p \mathbf{R} \times \mathbf{g} = M_p \mathbf{R} \times \mathbf{a}_p + \mathbf{I}_p \dot{\boldsymbol{\omega}}_p + \boldsymbol{\omega}_p \times \mathbf{I}_p \boldsymbol{\omega}_p \tag{5}$$

Combining Eqs. 3, 4 and 5, the closed form of Stewart platform dynamic equation is obtained as [3, 6]

$$\mathbf{M}\left(\mathbf{sx}\right)\ddot{\mathbf{x}} + \eta\left(\dot{\mathbf{x}}, \mathbf{sx}\right) = \mathbf{H}\left(\mathbf{sx}\right)\mathbf{F} \tag{6}$$

where

$$\mathbf{M} = \begin{bmatrix} M\mathbf{E}_3 & -M\tilde{\mathbf{R}} \\ M\tilde{\mathbf{R}} & \mathbf{I}_p + M\left(R^2\mathbf{E}_3 - \mathbf{R}\mathbf{R}^T\right) \end{bmatrix} + \sum_{n=1}^{6} \begin{bmatrix} \mathbf{Q}_i & -\mathbf{Q}_i\tilde{\mathbf{q}}_i \\ \tilde{\mathbf{q}}_i\mathbf{Q}_i & -\tilde{\mathbf{q}}_i\mathbf{Q}_i\tilde{\mathbf{q}}_i \end{bmatrix}$$

$$\eta = \begin{bmatrix} M\{\boldsymbol{\omega}_p \times \left(\boldsymbol{\omega}_p \times \mathbf{R}\right) - \mathbf{g}\} \\ \boldsymbol{\omega}_p \times \mathbf{I}_p + M\mathbf{R} \times \{\left(\boldsymbol{\omega}_p \cdot \mathbf{R}\right)\boldsymbol{\omega}_p - \mathbf{g}\} \end{bmatrix} + \sum_{n=1}^{6} \begin{bmatrix} \mathbf{V}_i \\ \mathbf{q}_i \times \mathbf{V}_i \end{bmatrix}$$

F is the actuation forces of all actuators

$$\mathbf{F} = [F_1 \quad F_2 \quad F_3 \quad F_4 \quad F_5 \quad F_6]$$

and **H** is

$$\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 & \mathbf{s}_6 \\ \mathbf{q}_{p1} \times \mathbf{s}_1 & \mathbf{q}_{p2} \times \mathbf{s}_2 & \mathbf{q}_{p3} \times \mathbf{s}_3 & \mathbf{q}_{p4} \times \mathbf{s}_4 & \mathbf{q}_{p5} \times \mathbf{s}_5 & \mathbf{q}_{p6} \times \mathbf{s}_6 \end{bmatrix}$$

It is now clear that matrix **H** is actually the transpose matrix [16] of the Jacobian matrix relating leg length velocity $\dot{\mathbf{L}}$ with platform velocity $\dot{\mathbf{x}}$ that

$$\dot{\mathbf{L}} = \mathbf{J}_{l,x}\dot{\mathbf{x}} = \mathbf{H}^T\dot{\mathbf{x}} \tag{7}$$

Take time derivative of Eq. 7 we have

$$\ddot{\mathbf{L}} = \mathbf{H}^T\ddot{\mathbf{x}} + \dot{\mathbf{H}}^T\dot{\mathbf{x}} = \mathbf{H}^T\ddot{\mathbf{x}} + \mathbf{U}(\mathbf{sx}, \dot{\mathbf{x}}) \tag{8}$$

Thus

$$\ddot{\mathbf{x}} = \mathbf{H}^{-T}[\ddot{\mathbf{L}} - \mathbf{U}(\dot{\mathbf{x}}, \mathbf{sx})] \tag{9}$$

Substituting Eq. 9 into Eq. 6 one gets

$$\ddot{\mathbf{L}} = \mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\mathbf{F} - \mathbf{H}^T\mathbf{M}^{-1}\boldsymbol{\eta}(\dot{\mathbf{x}}, \mathbf{sx}) + \mathbf{U}(\dot{\mathbf{x}}, \mathbf{sx}) \tag{10}$$

At this point we have the dynamic equations of Stewart platform both in operation space as given in Eq. 6 and partly in joint space with Eq. 10. For a motion control system, the actuation forces **F** are the system inputs. The matrix on the right hand side of Eq. 10 still depend on operation space states **sx** and $\dot{\mathbf{x}}$ and the cumbersome forward kinematics of parallel robots are required in a model based feedback control approach like NDI. However, it will be shown in the following chapters that this problem is inherently solved with INDI approach, thus Eq. 10 should be enough for INDI based controller design in joint space.

3 Motion System Controller Design

The application of INDI strategy on a hexapod flight simulator is previously mentioned in [5] by the author. The basic idea is to calculate the required incremental control inputs at the given moment to achieve a linearized relation. To do this, consider system dynamic equation Eq. 10, a first-order Taylor series estimation of the right hand side is given by

$$\begin{aligned}
\ddot{\mathbf{L}} = {} & \left(\mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\right)_{\mathbf{sx}_0}\mathbf{F}_0 - \left(\mathbf{H}^T\mathbf{M}^{-1}\right)_{\mathbf{sx}_0}\boldsymbol{\eta}\left(\dot{\mathbf{x}}_0, \mathbf{sx}_0\right) + \mathbf{U}\left(\dot{\mathbf{x}}_0, \mathbf{sx}_0\right) \\
& + \left(\mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\right)_{\mathbf{sx}_0}\left(\mathbf{F} - \mathbf{F}_0\right) \\
& + \frac{\partial}{\partial\dot{\mathbf{x}}}\left[-\mathbf{H}^T\mathbf{M}^{-1}\boldsymbol{\eta}\left(\dot{\mathbf{x}}, \mathbf{sx}\right) + \mathbf{U}\left(\dot{\mathbf{x}}, \mathbf{sx}\right)\right]_{\dot{\mathbf{x}}_0}\left(\dot{\mathbf{x}} - \dot{\mathbf{x}}_0\right) \\
& + \frac{\partial}{\partial\mathbf{sx}}\left[\mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\mathbf{F} - \mathbf{H}^T\mathbf{M}^{-1}\boldsymbol{\eta}\left(\mathbf{x}, \dot{}\mathbf{sx}\right) + \mathbf{U}\left(\mathbf{x}, \dot{}\mathbf{sx}\right)\right]_{\mathbf{sx}_0}\left(\mathbf{sx} - \mathbf{sx}_0\right)
\end{aligned} \tag{11}$$

where subscribe 0 denotes the time point around which the Taylor series expansion is taken.

For a nonlinear system, the first-order estimation is accurate enough within a very small time increment. The first terms on the right hand side of the equation is actually $\ddot{\mathbf{L}}_0$, i.e.

$$\ddot{\mathbf{L}}_0 = \left(\mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\right)_{\mathbf{sx}_0}\mathbf{F}_0 - \left(\mathbf{H}^T\mathbf{M}^{-1}\right)_{\mathbf{sx}_0}\boldsymbol{\eta}\left(\dot{\mathbf{x}}_0, \mathbf{sx}_0\right) + \mathbf{U}\left(\dot{\mathbf{x}}_0, \mathbf{sx}_0\right) \tag{12}$$

Which is leg length acceleration which can be obtained by sensor measurement. In case of SIMONA flight simulator motion system, leg length are measured by Temposonics sensors, thus the acceleration information should be obtained by numerical differentiation. The INDI based controller is sensitive to sensor measurement, thus is is also considered a sensor-based approach.

Now consider the other terms in Eq. 11. The last two terms are considered much smaller than the second term according to the time scale separation principle [14, 15, 17]. This is explained as the change of actuation forces $\delta\mathbf{F}$ has a direct effect on the system acceleration change, which is represented by the left hand side of Eq. 11. The system velocity only change by integrating the system acceleration and the position change is even integrated by the velocity. Thus when the integration sample time is very small, these two terms are also small while the input change can be significantly large. This indicates that the dynamics of system input is faster than system velocity and position. It is noted that this assumption only holds with very small sample time and very fast actuator dynamics. According to the aforementioned analysis, Eq. 11 is finally simplified as

$$\ddot{\mathbf{L}} = \ddot{\mathbf{L}}_0 + \left(\mathbf{H}^T\mathbf{M}^{-1}\mathbf{H}\right)_{\mathbf{sx}_0}(\mathbf{F} - \mathbf{F}_0) \tag{13}$$

Equation 13 is the incremental form of the system dynamic equation given by Eq. 10, which is actually a linear approximation around system states in zero time for a small time increment. By comparing these two equations we can see that nonlinear terms, $\mathbf{U}$ and $\boldsymbol{\eta}$ which consists of Coriolis/centripetal effects and gravity, disappeared from the original equation. This indicates that the INDI controller designed based on the incremental form system dynamics is not dependent on that part of the model. However, the dynamics of the neglected part of the model is still compensated by INDI since the leg length acceleration measurement already contain those information. Thus it is already clear now that INDI is not sensitive to model and parametric uncertainties in non-control-related parts in Eq. 6, e.g. gravity and Coriolis effect terms.

Based on Eq. 13, typical procedure of NDI can be designed by inverting the dynamics if the coupling matrix is invertible, hence the INDI control law is given in incremental form as

$$\delta\mathbf{F} = \left(\mathbf{H}^{-1}\mathbf{M}\mathbf{H}^{-T}\right)_{\mathbf{sx}_0}\left(\boldsymbol{\nu} - \ddot{\mathbf{L}}_0\right) \tag{14}$$

where $\boldsymbol{\nu}$ is the virtual control and the control input as the required actuation force is given by

$$\mathbf{u} = \mathbf{F} = \mathbf{F}_0 + \delta\mathbf{F} \tag{15}$$

Substituting Eq. 14 into Eq. 15, the full linearisation is achieved and the system is turned into a double integrator:

$$\boldsymbol{\nu} = \ddot{\mathbf{L}} \tag{16}$$

Simple linear controller can be designed for the double integrator, for example

$$\boldsymbol{\nu} = \ddot{\mathbf{L}}_d + \mathbf{K}_p \left(\mathbf{L}_d - \mathbf{L}\right) + \mathbf{K}_d \left(\dot{\mathbf{L}}_d - \dot{\mathbf{L}}\right) \tag{17}$$

where subscript d denotes the reference trajectory given by the host to be tracked by the motion system. In case of SIMONA simulator, the reference trajectory is given in operation space, which means inverse kinematics are required for the proposed joint space controller.

The state feedback used in INDI control law Eq. 13 is another problem for parallel robots. It is well known that the forward kinematic problem of 6-DOF does not have an analytic solution. The mass matrix $\mathbf{M}\left(\mathbf{sx}\right)$ and Jacobian matrix $\mathbf{H}\left(\mathbf{sx}\right)$ are given in the operation space while the measurement of system states is performed in joint space. That means numerical iteration are required to calculate system states in Cartesian space as well as the mass matrix and the Jacobian matrix, which will largely add the computational burden. However, as will be discussed as follows, this problem is inherently tackled by the robustness feature of INDI controller.

Assuming ideal sensor measurements, define the control related matrix in Eq. 13 as the control effectiveness matrix as

$$\mathbf{G}\left(\mathbf{sx}\right) = \mathbf{H}^T\left(\mathbf{sx}\right)\mathbf{M}^{-1}\left(\mathbf{sx}\right)\mathbf{H}\left(\mathbf{sx}\right) \tag{18}$$

In existence of model inaccuracies in the effectiveness matrix, for instance mass mismatch due to modification of flight cockpit or different pilot numbers, the system dynamics is then written as

$$\ddot{\mathbf{L}} = \ddot{\mathbf{L}}_0 + \left[\mathbf{G}_n\left(\mathbf{sx}_0\right) + \varDelta\mathbf{G}\left(\mathbf{sx}_0\right)\right]\left(\mathbf{F} - \mathbf{F}_0\right) \tag{19}$$

where the subscript n denotes the nominal condition.

Since only nominal part of the system is known, the following control law is actually applied according to Eq. 14:

$$\delta\mathbf{F} = \mathbf{G}_n^{-1}\left(\mathbf{sx}_0\right)\left(\boldsymbol{\nu} - \ddot{\mathbf{L}}_0\right) \tag{20}$$

Substituting Eq. 20 into Eq. 19, the system under control yields

$$\ddot{\mathbf{L}} = \boldsymbol{\nu} + \varDelta\mathbf{G}\left(\mathbf{sx}_0\right)\mathbf{G}_n^{-1}\left(\mathbf{sx}_0\right)\left(\boldsymbol{\nu} - \ddot{\mathbf{L}}_0\right) \tag{21}$$

Again, using the assumption that very high sampling rate is used in INDI, the change of leg length acceleration during one small sample time is very small that $\ddot{\mathbf{L}} \approx \ddot{\mathbf{L}}_0$ holds and Eq. 21 yields

$$\mathbf{A}\ddot{\mathbf{L}} = \mathbf{A}\boldsymbol{\nu} \tag{22}$$

where

$$\mathbf{A} = \mathbf{I}_{6\times6} + \Delta\mathbf{G}\left(\mathbf{sx}_0\right)\mathbf{G}_n^{-1}\left(\mathbf{sx}_0\right) \tag{23}$$

This result indicates that with relative high sampling rate, the linearized relation still holds with INDI controller in existence of model uncertainties in control effectiveness matrix. This also imply that the INDI based controller is not sensitive to almost all kinds of model inaccuracies in mass matrix, gravity effect terms as well as the Coriolis/centripetal effect terms. Thus the proposed control scheme is robust to model uncertainties of a hexapod parallel robot.

Taking advantage of the robustness of INDI methodology to model inaccuracies, tow important modification can be made to simplify the controller design. First, the complicated forward kinematics of parallel robot can be removed from the control structure. As the controller is robust to control effectiveness matrix inaccuracies, the trajectory set-points can be used to calculate the mass matrix $\mathbf{M}$ and Jacobian matrix $\mathbf{H}^T$ instead of system position feedback, since they are considered to be very close if the motion controller works. Second, simplified mass matrix can be used by neglecting less important parts like the mass of legs, which will largely reduce the computation burden. By doing that, the INDI control law Eq. 14 can be simplified to

$$\delta\mathbf{F} = \left(\mathbf{H}^{-1}\mathbf{M}_s\mathbf{H}^{-T}\right)_{\mathbf{sx}_d}\left(\boldsymbol{\nu} - \ddot{\mathbf{L}}_0\right) \tag{24}$$

where $\mathbf{M}_s$ is the simplified mass matrix and $\mathbf{sx}_d$ is the system position trajectory set-points.

The proposed motion control system is presented in Fig. 2. The input of the INDI controller is the virtual control $\boldsymbol{\nu}$ which is subtracted by actuator acceleration measurement and the output is the calculated actuation force required to track the motion. The virtual control is calculated by a linear controller which hire state feedback and inverse kinematics of Stewart platform. The reference trajectory set-points are used to calculate the control effectiveness matrix. Using the aforementioned two simplifications, the controller share the advantage of a feed forward control scheme, i.e. the avoidance of forward kinematics and availability of simplified model, while still offer a full system linearisation just as a typical NDI did with precise model information.

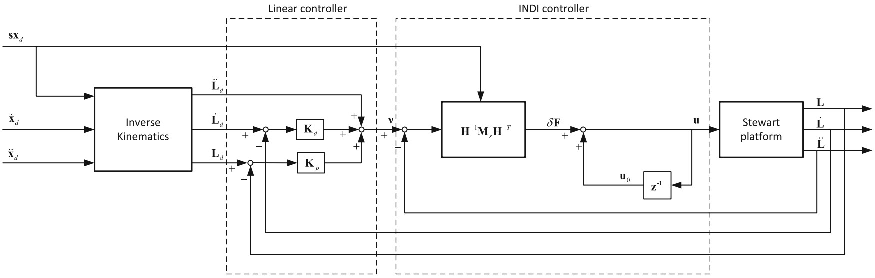


Fig. 2 INDI based motion controller architecture

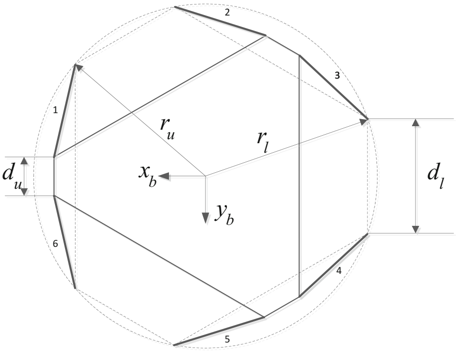


Fig. 3 SRS top view [6]

The proposed INDI based motion controller is robust to model uncertainties at a cost of sensitivity to sensor measurement. It is note that the INDI technique is based on three main assumptions: high sampling rate, fast actuator dynamics and ideal sensor measurement. The performance of proposed control system will be tested with simulation in the following chapter.

4 Simulation Results

The performance of the proposed controller is verified by numerical simulations, the leg length tracking performance is presented since the controller is designed in the joint space. A detailed model of flight simulator motion system based on SIMONA Research Simulator(SRS) in Delft University of Technology is previously introduced in literature [6] and is used for simulations in this work, only the actuators are set to be ideal force generators. The simulations are performed in MATLAB/Simulink environment and the fourth-order Runge–Kutta integration is used to solve the

Table 1 Geometric and inertial parameters

Parameters	Value
Upper/lower gimbal radius, r_a r_b	1.6, 1.65 m
Upper/lower radius spacing, d_u d_l	0.2, 0.6 m
Piston/cylinder masses, m_2 m_1	0.12, 0.15 tons
Piston/cylinder inertia wrt cog, i_2 i_1	20, 36 kg m^2
Piston/cylinder cog wrt to gimbal, r_2 r_1	0.7, 0.5 m
Platform mass M_p	3.2 tons
Platform nonzero inertial I_{xx} I_{yy} I_{zz} I_{xz}	7, 7, 8, 0.5 tons m^2

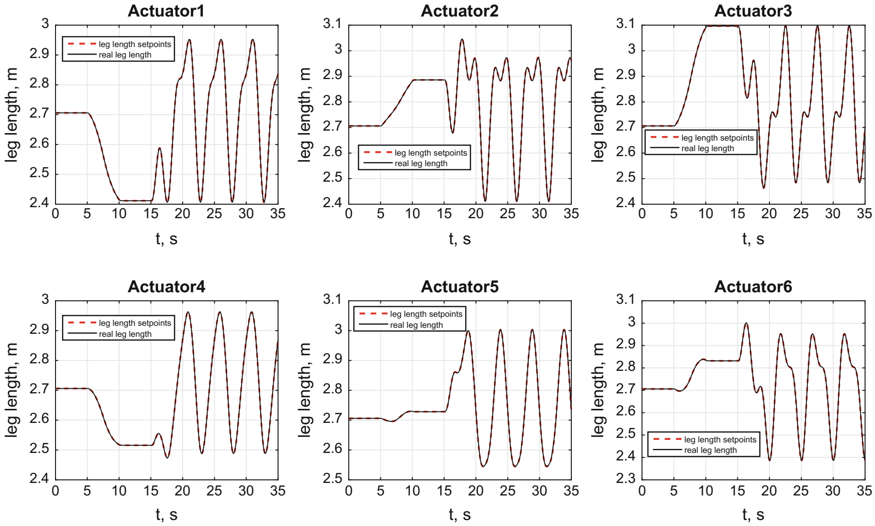


Fig. 4 Position tracking performance with nominal INDI controller

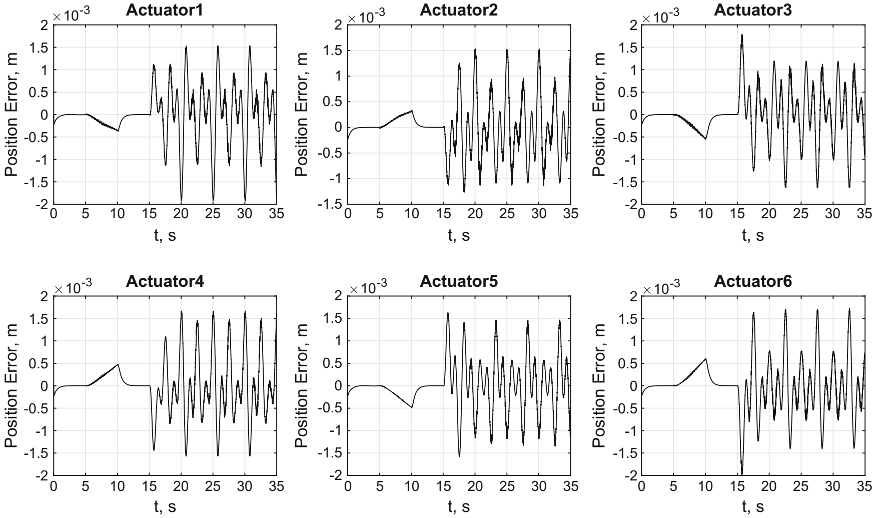


Fig. 5 Position tracking error with nominal INDI controller

system dynamic equation. The geometry of SRS simulator motion system is shown in Fig. 3 while the geometric and inertial parameters are given in Table 1. For all the simulations, the sampling rate is set to 1000 Hz, which is the same with the current controller implemented on SRS simulator.

Motion tracking tasks are performed in all the simulations. The motion profile of a state reconstruction experiment [6, 10] performed on the SRS is used to give

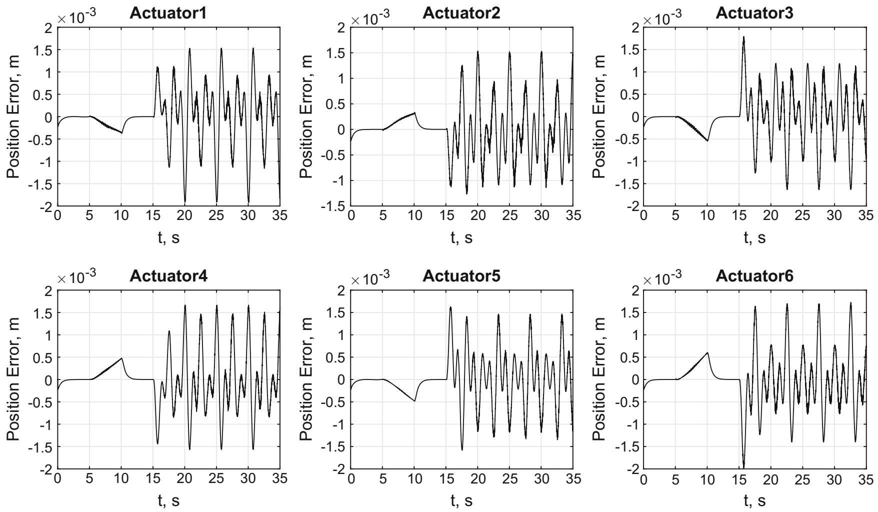


Fig. 6 Position tracking error with simplified INDI controller

reference trajectory in operation space. The origin of the upper platform moves in a horizontal circular path with a radius of 0.5 m with a period of 5 s.

Figure 4 presents the tracking performance of each leg under nominal condition while real state feedback is used to calculate matrix **M** and **H**. The leg length tracking errors are shown in Fig. 5. It is shown that the controller gives stable and good tracking performance, with a maximum tracking error limited within 2×10^{-3}m. This result is already a verification of robustness of INDI controller to model inaccuracies in gravity and Coriolis related terms, since that part of the model is not used in the controller.

Figure 6 gives tracking errors when the system position set-point, instead of the real state feedback, are used to calculate the control effectiveness matrix. In this simulation the modified control law Eq. 24 is used. It is shown that the performance almost remain intact. This result verified that the INDI controller is not sensitive to model mismatch in the control effectiveness matrix, and the proposed simplified control law will provide similar performance as a complete state feedback approach will do.

In order to further test the robustness of the proposed controller, a significant model mismatch is set for the system. In Fig. 7 the position tracking error of each leg when there is 50% mismatch in the upper platform mass is presented, which is generally beyond what we expect in reality. It is indicated that the performance is almost intact again. This result verifies that INDI is very robust to parameter mismatch in the control effectiveness matrix, hence the validation of using a simplified mass matrix $\mathbf{M}_s$ is also enhanced.

The previous simulation results verify that at least at an practical sampling rate, the time scale separation principle assumption holds for a hexapod flight simulator

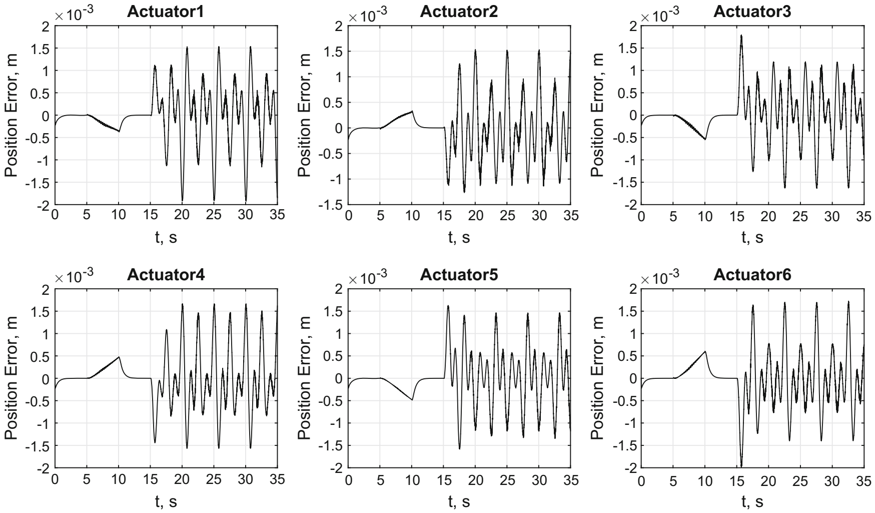


Fig. 7 Position tracking error with simplified INDI controller with 50% mass mismatch

motion system. The INDI technique is a promising approach for the studied motion system and a wider range of parallel or serial robots to overcome model uncertainties with simple design procedure and small computational burden.

5 Conclusion

Incremental nonlinear dynamic inversion is a promising approach for parallel robot motion control. The time scale separation principle, which is the base of INDI technique, holds for the studied system at a practical sampling rate. The INDI based controller is not sensitive to almost all kinds of model and parametric inaccuracies, which allows for an accurate system linearization without precise model information. Furthermore, the robustness of INDI to model uncertainties allows for trajectory set-points to be used with simplified mass matrix, which will largely reduce the computational burden. Since INDI technique depends on accurate sensor measurement and fast actuator dynamics, the influence of measurement noise and bias, as well as actuator dynamics should be investigated as future work before the proposed control scheme is implemented to a promising real world application.

References

1. Becerra-Vargas M, Eduardo MBelo (2011) Robust control of flight simulator motion base. J Guid Control Dyn 34(5):1519–1528
2. Dasgupta B, Mruthyunjaya TS (1998) A Newton-Euler formulation for the inverse dynamics of the stewart platform manipulator. Mech Mach Theory 33(8):1135–1152
3. Dasgupta B, Mruthyunjaya TS (1998) Closed-form dynamic equations of the ceneral stewart platform through the Newton-Euler approach. Mech Mach Theory 33(7):993–1012
4. Honegger M, Codourey A, Burdet E (1997) Adaptive Control of the Hexaglide, a 6 dof Parallel Manipulator. In Proceedings of the 1997 IEEE international conference on robotics and automation, 543–548, 1997
5. Huang Y, Pool DM, Stroosma O, Chu QP, Mulder M (2016) A review of control schemes for hydraulic stewart platform flight simulator motion systems
6. Huang Y, Pool DM, Stroosma O, Chu QP, Mulder M (2016) Modeling and simulation of hydraulic hexapod flight simulator motion systems. In AIAA modeling and simulation technologies conference, 2016
7. Kim HS, Cho YM, Lee KI (2005) Robust nonlinear task space control for 6 DOF parallel manipulator. Automatica, 41:1591–1600
8. Kim DH, Kang JY, Lee KL (2000) Robust tracking control design for a 6 DOF parallel manipulator. J Robot Syst 17(10):527–547
9. Lee Se-Han, Song Jae-Bok, Choi Woo-Chun, Hong Daehie (2003) Position control of a Stewart platform using inverse dynamics control with approximate dynamics. Mechatronics 13:605–619 Jul
10. Miletović I (2014) Optimal reconstruction of flight simulator motion cues. Unpublished Msc. thesis, Delft University of Technology, 2014
11. Nguyen CC, Antrazi SS, Zhou ZL, Campbell CE Jr (1992) Adaptive control of a stewart platform-based manipulator. J Robot Syst 10(5):657–687
12. Shang Wei-Wei, Cong Shuang, Ge Yuan (2012) Adaptive computed torque control for a parallel manipulator with redundant actuation. Robotica 30:457–466
13. Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics Modelling Planning and Control. Springer, Berlin
14. Sieberling S, Chu QP, Mulder JA (2010) Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. J Guid Control Dyn 33(6):1732–1742 Nov
15. Simplício P, Pavel MD, van Kampen E, Chu QP (2013) An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion. Control Eng Pr 21:1065–1077
16. Sjirk Koekebakker (2001) Model based control of a flight simulator motion system. PhD thesis, Delft University of Technology, 2001
17. Smeur EJ, Chu QP, de Croon G (2016) Adaptive incremental nonlinear dynamic inversion for attitude control of micro aerial vehicles. J Guid Control Dyn 39(3):450–461,
18. Stroosma O, van Paassen MM, Mulder M (2003) Using the SIMONA research simulator for human-machine interaction research. In: AIAA modeling and simulation technologies conference and exhibit, 2003

Limit Cycle Oscillation Amplitude Tailorng Based on Describing Functions and μ Analysis

Andrea Iannelli, Andrés Marcos and Mark Lowenberg

1 Introduction

One of the major challenges faced by aeronautical industry nowadays is to achieve lightweight aircraft configurations which enable to reduce fuel consumptions and operating costs ensuring at the same time a feasible design in terms of safety constraints. Among the most dangerous instability phenomena exacerbated by wing flexibility there is flutter.

Flutter is a self-excited instability in which aerodynamic forces acting on a flexible body couple with its natural structural vibration modes producing oscillatory motion. The level of vibration may result in sufficiently large amplitudes to provoke failure and often this requirement dictates the design of the structure. The study of this problem hence entails a multi-disciplinary approach, involving structural dynamics and aerodynamics, commonly tackled by means of aeroelasticity. Flutter analysis has been widely investigated and several well-established techniques are available, representing the state-of-practice (see [11] for a thorough review).

Despite the large amount of effort spent in understanding flutter, it is acknowledged that predictions based only on computational analyses are not totally reliable. Currently this is compensated by the addition of conservative safety margins to the analysis results as well as expensive flutter test campaigns. One of the main criticalities arises from the sensitivity of aeroelastic instability to small variations in parameter and modeling assumptions. In addressing this issue, in the last two decades

A. Iannelli (✉) · A. Marcos · M. Lowenberg
Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, UK
e-mail: andrea.iannelli@bristol.ac.uk

A. Marcos
e-mail: andres.marcos@bristol.ac.uk

M. Lowenberg
e-mail: m.lowenberg@bristol.ac.uk

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_6

researchers looked at robust modeling and analysis techniques from the robust control community, specifically linear fractional transformation (LFT) models and μ (structured singular value) analysis [19, 25]. The so-called flutter robust analysis [5, 17] aims to quantify the gap between the prediction of the nominal stability analysis (model without uncertainties) and the worst-case scenario when the whole set of uncertainties is contemplated. In recent studies [12, 13, 16] other issues of this consolidated framework have been considered as: better understanding of the modelling options and their effects on the predictions, application of μ as flutter sensitivity tool and reconciliation of the robust analyses results with the physical understanding of the systems.

The framework of μ technique is straightforwardly applicable to the study of robust stability (RS) and performance (RP) of linear plants. However the increase in flexibility on one side and a more realistic description of the system on the other compel to consider cases where the linear hypothesis no longer holds. Aeronautical industry, for example, has recently shown interest in evaluating the effect of the uncertainties on instabilities prompted by the control surface freeplay [20].

A way to introduce nonlinearities in the frequency domain framework, borrowed from the control community, is represented by the Describing Function (DF) method [9]. This has been employed for the prediction of onset, frequency and amplitude of Limit Cycle Oscillations (LCOs) in aeroservoelastic applications [7, 10].

The present study applies the framework μ-DF to a well-known benchmark [22] in order to assess the nonlinear flutter behavior of a typical section configuration with control surface freeplay. Within the showcased framework, μ analysis can be applied in order to investigate the robustness of the LCO in the face of uncertainties [3]. Here a novel application of these tools to the study of nonlinear systems is proposed, which enables to exploit the worst-case capability of μ in order to tailor the LCO properties of the system with the smallest effort in terms of change of nominal design (*optimal passive* design remedies).

The layout of the article is as follows. Section 2 presents the linear system considered in the analyses. An overview for each one of the techniques employed in the work (DF, LFT and μ) is given in Sect. 3. In Sect. 4 the application of the built framework to the chosen test bed is shown. The rationale behind the proposed tailoring of the nonlinear behavior of the system and the corresponding results are provided in Sect. 5. Finally, Sect. 6 gathers the main conclusions of the work.

2 Typical Section System

2.1 Model

The *typical section* model was introduced in the early stages of aeroelasticity to investigate dynamic phenomena such as flutter [4]. Despite its simplicity, it captures essential effects in a simple model representation, see Fig. 1.

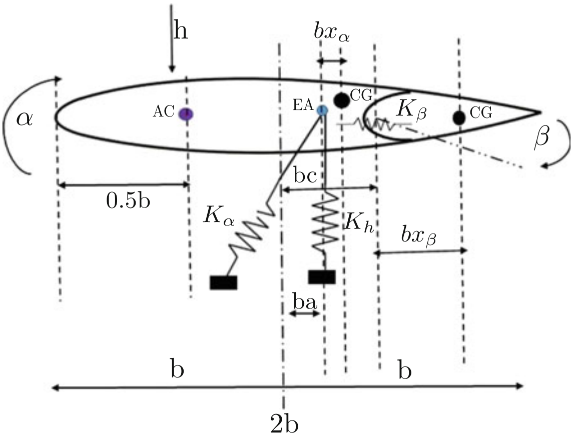


Fig. 1 Typical section sketch

From the structural side, it basically consists of a rigid airfoil with lumped springs simulating the 3 degrees of freedom of the section: plunge h, pitch α and trailing edge flap β. The positions of the elastic axis (EA), center of gravity (CG) and the aerodynamic center (AC) are also marked. The main parameters in the model, see Fig. 1, are: K_h, K_α and K_β –respectively the bending, torsional and control surface stiffness; half chord distance b; dimensionless distances a, c from the mid-chord to respectively the flexural axis and the hinge location, and x_α and x_β, which are dimensionless distances from flexural axis to airfoil center of gravity and from hinge location to control surface center of gravity. S_α and S_β, not explicitly reported in the picture and employed later on in the analyses, are the equivalent of the latter with dimension, i.e. $S_\alpha = bx_\alpha$ and $S_\beta = bx_\beta$.

In addition to the above parameters, the inertial characteristics of the system are given by: the wing mass per unit span m_s, the moment of inertia of the section about the elastic axis I_α, the moment of inertia of the control surface about the hinge I_β. If structural damping is considered, this can be expressed specifying the damping ratios for each DOFs and then applied through modal damping approach [6].

The Theodorsen unsteady formulation proposed in [23] is employed as aerodynamic model. This approach is based on the assumption of a thin airfoil moving with small harmonic oscillations in a potential and incompressible flow. Despite its simplicity, such idealization is pertinent to flutter analysis since this implies a condition of neutral stability of the system. The same hypotheses underline most of more improved aerodynamic approaches (e.g. Doublet Lattice Method).

In order to present the basic model development approach, $\mathbf{X}$ and $\mathbf{L_a}$ are defined as the vectors of the degrees of freedom and of the aerodynamic loads respectively:

$$\mathbf{X}(t) = \begin{bmatrix} \frac{h(t)}{b} \\ \alpha(t) \\ \beta(t) \end{bmatrix}; \quad \mathbf{L_a}(t) = \begin{bmatrix} -L(t) \\ M_\alpha(t) \\ M_\beta(t) \end{bmatrix} \tag{1}$$

The set of differential equations describing the dynamic equilibrium [11] can then be recast in matrix form using Lagrange's equations:

$$\big[\mathbf{M_s}\big]\ddot{\mathbf{X}} + \big[\mathbf{C_s}\big]\dot{\mathbf{X}} + \big[\mathbf{K_s}\big]\mathbf{X} = \mathbf{L_a} \tag{2}$$

where $\big[\mathbf{M_s}\big]$, $\big[\mathbf{C_s}\big]$ and $\big[\mathbf{K_s}\big]$ are respectively the structural mass, damping and stiffness matrices. The expression of $\mathbf{L_a}$, provided in the Laplace s domain, is:

$$\begin{aligned} &\mathbf{L_a}(s) = q\big[\mathbf{A}(\bar{s})\big]\mathbf{X}(s) \\ \text{with}\quad &\mathbf{L_a}(s) = \mathscr{L}\big[\mathbf{L_a}(t)\big]; \quad \mathbf{X}(s) = \mathscr{L}\big[\mathbf{X}(t)\big] \end{aligned} \tag{3}$$

where the dynamic pressure q and the dimensionless Laplace variable $\bar{s}$ (=$s\frac{b}{V}$ with V the airspeed) are introduced. $\mathbf{A}(\bar{s})$ is called the generalized Aerodynamic Influence Coefficient (AIC) matrix, and is composed of generic terms $\mathbf{A}(\bar{s})_{ij}$ representing the transfer function from each degree of freedom j in $\mathbf{X}(s)$ to each aerodynamic load component i in $\mathbf{L_a}(s)$.

Due to the expression of the AIC matrix which does not have a rational dependence on the Laplace variable s, the final aeroelastic equilibrium is inherently expressed in frequency-domain and is given by:

$$\Big[\big[\mathbf{M_s}\big]s^2 + \big[\mathbf{C_s}\big]s + \big[\mathbf{K_s}\big]\Big]\mathbf{X} = q\big[\mathbf{A}(\bar{s})\big]\mathbf{X} \tag{4}$$

A rational approximation of $\big[\mathbf{A}(\bar{s})\big]$ has to be performed in order to express the equilibrium in state space, which is essential to deal with aeroservoelastic problems and control based techniques. In this work Minimum State (MS) [14] method is employed, which propose the following expression for the aerodynamic operator:

$$\big[\mathbf{A}\big] \approx \big[\mathbf{A}_{MS}\big] = \big[\mathbf{A_2}\big]\bar{s}^2 + \big[\mathbf{A_1}\big]\bar{s} + \big[\mathbf{A_0}\big] + \big[\mathbf{D}'\big]\Big(\bar{s}\big[\mathbf{I}\big] - \big[\boldsymbol{\Gamma}\big]\Big)^{-1}\big[\mathbf{E}'\big]\bar{s} \tag{5}$$

where $\big[\mathbf{A_i}\big]$ are real coefficient matrices obtained imposing a matching of the operators ($\mathbf{A}$ and $\mathbf{A}_{MS}$) at determined frequencies of oscillations. The last block consists of the so-called *lag terms* which basically represent high-pass filters of the form $\frac{\bar{s}}{\bar{s}+\gamma_i}$ with the aerodynamic roots γ_i as cross-over frequencies (defining the diagonal matrix $\boldsymbol{\Gamma}$) and the gains provided by the real matrices $\big[\mathbf{D}'\big]$ and $\big[\mathbf{E}'\big]$. They are iteratively determined through a nonlinear least square since Eq. (5) is bilinear in these two unknowns. The impact on robust flutter analysis of the different expressions for the AIC matrix (original frequency-domain operator or state-space approximations) when aerodynamic uncertainties are considered has been investigated in [12].

The resulting state-space equation, which includes N augmented aerodynamic states equal in number to the lag roots γ_i, is here reported in short-hand notation:

$$\begin{bmatrix}\dot{\hat{\mathbf{X}}}_{\mathbf{s}}\\ \dot{\hat{\mathbf{X}}}_{\mathbf{a}}\end{bmatrix} = \begin{bmatrix}\chi_{ss} & \chi_{sa}\\ \chi_{as} & \chi_{aa}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{X}}_{\mathbf{s}}\\ \hat{\mathbf{X}}_{\mathbf{a}}\end{bmatrix} \tag{6}$$

where $\hat{\mathbf{X}}_{\mathbf{s}}$ and $\hat{\mathbf{X}}_{\mathbf{a}}$ are respectively the vector of structural and aerodynamic states.

2.2 Linear Nominal Flutter Analysis

Nominal flutter analysis studies the conditions at which the dynamic aeroelastic system loses its stability. As the airspeed V varies the system's behavior in terms of response and stability changes. The result is the prediction of the so-called flutter speed V_f, below which the system is guaranteed to be stable.

The stability of the system studied here is related to the spectrum of the state-matrix defined in Eq. (6). Six aerodynamic roots γ_i equally spaced between 0.1 and 0.7 are chosen to approximate the Theodorsen operator. This is done in order to span the range of reduced frequencies where the flutter frequency is expected to be.

The nominal parameters defining the typical section studied in this work are taken from [22]. This test bed focuses on a three degrees of freedom typical section configuration with control surface freeplay. This nonlinearity, which will be examined later in Sect. 3.1.2, affects the diagonal term of the matrix $\mathbf{K_s}$ corresponding to the control surface rotation (i.e. K_β). If this term is assumed as K_β^L (i.e. the control surface stiffness without freeplay) and all the other parameters hold their nominal values, a linear analysis of the system can be performed.

In Fig. 2 the eigenvalues corresponding to the structural modes of the system are depicted as the airspeed increases from 1 $\frac{\text{m}}{\text{s}}$ (square marker) to 30 $\frac{\text{m}}{\text{s}}$. The system exhibits a plunge-pitch flutter, highlighted with the circle marker. This phenomenon is commonly called *binary flutter* since it is mainly featured by the interaction of two modes.

In Table 1 a comparison for the linear flutter speed V_L with the ones reported in other works studying this problem is reported.

Fig. 2 Nominal linear system: pole locations as a function of airspeed

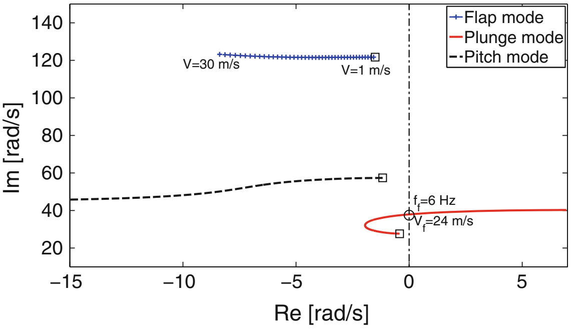


Table 1 Comparison of results in terms of flutter speed and frequency for the linear case

	Flutter velocity $\left[\frac{\text{m}}{\text{s}}\right]$	Flutter frequency [Hz]
Present work	24	6
[18]	23.9	6.1
[6]	23.9	6
[3]	23.2	6

3 Quasi-linear Robust Framework

3.1 *Describing Function Method*

The analysis and design of linear control plants pivots on a complex-valued function, the frequency response. However, this function cannot be defined for nonlinear systems, hence frequency domain techniques cannot be directly applied. The Describing Function (DF) method [9, 15] aims to overcome this obstacle by providing in these cases an alternative definition of frequency response. The basic concepts of this technique are here presented, with particular emphasis on this application.

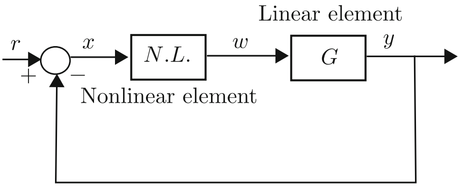
3.1.1 Main Concept

The concept of *quasi-linearization* is the basis of the application of DF. This expedient originates by the goal to retain the advantages of a linear approximation without the constraint of requiring small departures of the variables from the nominal operating values. This can be achieved if the approximation of the nonlinear operator depends on some properties of the input. Dependence of performance on signal amplitude, the basic characteristic of nonlinear behavior, is thus retained.

The DF method is mostly applied to systems which can be recast in the framework of Fig. 3. This class of system is characterised by having separable linear and nonlinear parts connected in a single loop configuration, in the field of aeroservoelasticity [8] (e.g. saturation, freeplay, hysteresis).

The quasi-linearization process requires that the input signal *form* is specified. In analogy to what is done in frequency domain analysis, it is widely employed the concept of sinusoidal-input describing function (SIDF), in the following simply

Fig. 3 Feedback representation of a nonlinear system



abbreviated as DF. The interest in periodic signals is mainly dictated by the aim to detect and analyse steady oscillations in nonlinear systems, also known as Limit Cycle Oscillations (LCOs).

The *key* hypothesis of the DF method is that only the fundamental component has to be retained from the generical periodic output w(t). This is an approximation because the output of a nonlinear element corresponding to a sinusoidal input usually contains higher harmonics. This is motivated by the assumption that the higher harmonics in the output are filtered out, i.e. the linear element satisfies the low-pass filter condition (or *filter hypothesis*):

$$\|G(j\omega)\| \gg \|G(jn\omega)\| \quad \text{for} \quad n = 2, 3, \dots \tag{7}$$

The DF of a nonlinear element is thus the complex fundamental-harmonic gain of a nonlinearity in the presence of a driving sinusoid of amplitude A:

$$\begin{aligned} & N(A,\omega) = \frac{Me^{j(\omega t+\phi)}}{Ae^{j(\omega t)}} = \frac{M}{A}e^{j\phi} = \frac{b_1 + ja_1}{A} \\ \text{with} \quad & x = A\sin(\omega t); \quad w(t) \approx a_1(A,\omega)\cos(\omega t) + b_1(A,\omega)\sin(\omega t) \\ & M(A,\omega) = \sqrt{a_1^2 + b_1^2}; \quad \phi(A,\omega) = \arctan\left(\frac{a_1}{b_1}\right) \end{aligned} \tag{8}$$

This method hence consists in treating the nonlinear element of Fig. 3 in the presence of sinusoidal input as if it were a linear element with a frequency response $N(A,\omega)$.

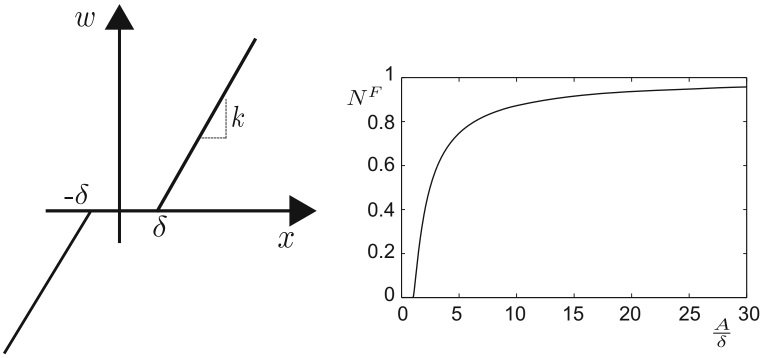
3.1.2 DF for the Freeplay Nonlinearity

Due to the well established usage of DF method in control community, the expression of $N(A,\omega)$ can be found for the majority of nonlinearities commonly encountered in applications [9]. Freeplay, also called dead-zone or threshold, often arises in mechanical and electrical systems where the first part of the input is needed to overcome an initial opposition at the output, as schematically depicted in Fig. 4a. Its explicit mathematical definition (refer to Fig. 4a for the symbology) is given by:

$$w = \begin{cases} k(x-\delta); & |x| > \delta \\ 0; & |x| < \delta \end{cases} \tag{9}$$

Having defined the (positive) freeplay size δ, the describing function N^F, obtained analytically, is then:

$$N^F(A) = \begin{cases} 0; & A < \delta \\ \frac{k}{\pi}\left[\pi - 2\sin^{-1}\left(\frac{\delta}{A}\right) - 2\left(\frac{\delta}{A}\right)\sqrt{1-\left(\frac{\delta}{A}\right)^2}\right]; & A > \delta \end{cases} \tag{10}$$

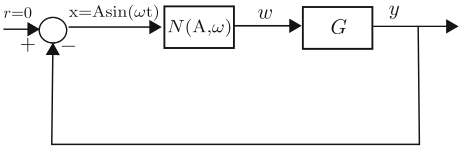


(a) Freeplay nonlinearity

(b) Associated Describing Function

Fig. 4 Freeplay nonlinearity: graphical representation and associated Describing Function

Fig. 5 Quasi-linearized system specialised for LCOs study through DF



Due to the properties (static and memoryless) hold by this nonlinearity, its describing function is a pure real gain not depending on frequency, but only on the amplitude A of the signal, in particular on its ratio with δ. In Fig. 4b N^F is depicted for the case $k = 1$.

3.1.3 Detection of LCOs

One of the main applications of DF method is the study of the existence of steady oscillations in a nonlinear system. In particular Limit Cycle Oscillations (LCOs) [24] are of engineering interest, which are defined as initial condition-independent isolated periodic orbits which occurs in unforced dissipative systems. LCOs are usually avoided in mechanical systems since they are likely to reduce fatigue's life and provoke critical damages. The aeronautical industry has sensibly strengthen his focus on this phenomena in the last decade [8] and thus increasing effort has been devoted on this topic by the research community. It is well ascertained that the knowledge of the existence of a limit cycle, as well as its approximate amplitude and frequency, is a prerequisite for a good system design [20].

The methodology employed to study LCOs through DF takes the clue from the feedback representation of a nonlinear system in Fig. 3, specialised in Fig. 5 by specifying the signal circulating through the system as a sinusoid, thus replacing the nonlinear element by its describing function N, and considering null input (since focus is on stability properties of the system).

Linear theory is then applied to the quasi-linearized system, searching for points of neutral stability which are interpreted as LCOs in the original (nonlinear) system. The well-known feedback relations involve the frequency response of the signals (in capital letters) and the transfer functions of the operators, and the resulting necessary condition for oscillations are:

$$\begin{aligned} X(j\omega) &= -G(j\omega)W(j\omega) \\ W(j\omega) &= N(j\omega)X(j\omega) \\ \big[1 + G(j\omega)N(A,\omega)\big]X(j\omega) &= 0 \end{aligned} \tag{11}$$

Solution of the characteristic equation gives the conditions in terms of A and ω such that the system exhibits self-sustained oscillations.

The characteristic equation in Eq. (11) gives only a necessary condition for the occurrence of periodic oscillations in the nonlinear system. Indeed stability of the oscillation has still to be verified. The question of LCO stability is generally posed in terms of the states trajectory behavior following amplitude or frequency perturbations. If the LCO returns to its original equilibrium state it is defined *stable*, otherwise it is *unstable*. To tackle the stability problem, the methodology proposed in [2] is here adopted. The idea is to study the variation of the real part σ of the eigenvalue associated to the LCO solution following a perturbation in A. A stable limit cycle requires $\frac{\partial\sigma}{\partial A} < 0$, since a positive perturbation in amplitude moves the trajectory outside the limit cycle and requires a decay in the amplitude (negative real part) to move the trajectory back to the limit cycle, and conversely an unstable limit cycle will have $\frac{\partial\sigma}{\partial A} > 0$.

3.2 Linear Fractional Transformation and μ Analysis

Robust flutter analysis [17] deals with flutter instability predictions when the aeroelastic model is subject to uncertainties. Examples of the latter are low confidence in the values of parameters and coefficients of the matrices, or neglected dynamics in the nominal model. Once the problem is described within the LFT framework, μ analysis enables to predict for a given airspeed if the set of uncertainties is capable to lead to instability. A very brief description of these tools is here provided (see [19, 25] and references herein).

If the *coefficient matrix* $\mathbf{M}$ is defined as a proper transfer matrix, $\mathscr{F}_u$, namely the upper LFT, is the closed-loop transfer matrix from input $\mathbf{u}$ to output $\mathbf{y}$ when the nominal plant $\mathbf{M_{22}}$ is subject to a perturbation matrix $\boldsymbol{\Delta_u}$ (Fig. 6).

$\mathbf{M_{11}}$,$\mathbf{M_{12}}$ and $\mathbf{M_{21}}$ reflect a priori knowledge of how the perturbation affects the nominal map. Once all varying or uncertain parameters are *pulled out* of the nominal plant, the problem appears as a nominal system subject to an artificial feedback. The algebraic expression for $\mathscr{F}_u$ is given by:

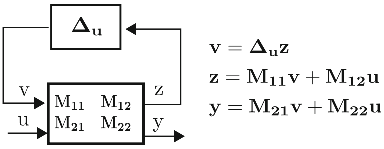


Fig. 6 Upper LFT

$$\mathscr{F}_u(\mathbf{M}, \boldsymbol{\Delta}_u) = \mathbf{M}_{22} + \mathbf{M}_{21}\boldsymbol{\Delta}_u(\mathbf{I} - \mathbf{M}_{11}\boldsymbol{\Delta}_u)^{-1}\mathbf{M}_{12} \tag{12}$$

This LFT is well posed if and only if the inverse of $(\mathbf{I} - \mathbf{M}_{11}\boldsymbol{\Delta}_u)$ exists.

The structured singular value $\mu_{\boldsymbol{\Delta}}(\mathbf{M})$ of a matrix $\mathbf{M} \in C^{n\times n}$ with respect to the uncertain matrix $\boldsymbol{\Delta}$ is defined below:

$$\mu_{\boldsymbol{\Delta}}(\mathbf{M}) = \frac{1}{\min_{\boldsymbol{\Delta}}(\bar{\sigma}(\boldsymbol{\Delta}) : \det(\mathbf{I} - \mathbf{M}\boldsymbol{\Delta}) = 0)} \tag{13}$$

where $\bar{\sigma}(\boldsymbol{\Delta})$ is the maximum singular value of $\boldsymbol{\Delta}$ and $\mu_{\boldsymbol{\Delta}}(\mathbf{M}) = 0$ if there is no $\boldsymbol{\Delta}$ satisfying the determinant condition. Note that this definition can be specialized to determine whether the LFT $\mathscr{F}_u(\mathbf{M}, \boldsymbol{\Delta}_u)$ is well posed once the generic matrix $\mathbf{M}$ in the above definition is replaced by $\mathbf{M}_{11}$ and $\boldsymbol{\Delta}$ belongs to the corresponding uncertainty set $\boldsymbol{\Delta}_u$. For ease of calculation and interpretation, this set is typically norm-bounded $\|\boldsymbol{\Delta}\|_\infty < 1$ (without loss of generality by scaling of $\mathbf{M}_{11}$). In this manner, if $\mu_{\boldsymbol{\Delta}}(\mathbf{M}_{11}) \leq 1$ then the result guarantees that the analyzed system is robustly stable (RS) to the considered uncertainty.

4 Analysis of the Typical Section LCO Due to Freeplay

Considerable work has been done in the past two decades in investigating the effects of structural nonlinearities on aeroelastic phenomena. Examples of this effort are the experimental [6] and analytical studies [22] conducted to precisely assess the nonlinear flutter behavior of a three-degrees-of-freedom typical section configuration with control surface freeplay. The analytical studies employed both the direct time integration and the DF method with the aim to determine LCO amplitudes and frequencies and offer a comparison of the predicted responses with experimental data. These benchmark results have driven further studies widening the investigations of this testcase: harmonic-balance [18], continuation methods [10] and μ analysis [3].

4.1 Detection of Freeplay-Induced LCOs

The investigations builds on the linear model described in Sect. 2.1 and the DF method to deal with the freeplay nonlinearity. The application of the latter enables to give an expression for the elastic moment M_β^E acted by the control surface:

$$\begin{aligned} M_\beta^E &= K_\beta^{QL}\beta \\ K_\beta^{QL} &= N^F(\beta)K_\beta^L \end{aligned} \tag{14}$$

where K_β^L is the flap stiffness in the linear case (δ=0), K_β^{QL} is the quasi-linear flap stiffness and N^F is the DF provided in Eq. (10) and depicted in Fig. 4b, specialised to this case taking k=1 and A=β. In other words, the linear stiffness K_β^L is replaced by an equivalent stiffness K_β^{QL} in the diagonal term of $\mathbf{K_s}$ corresponding to the control surface, which is a function of the flap rotation β.

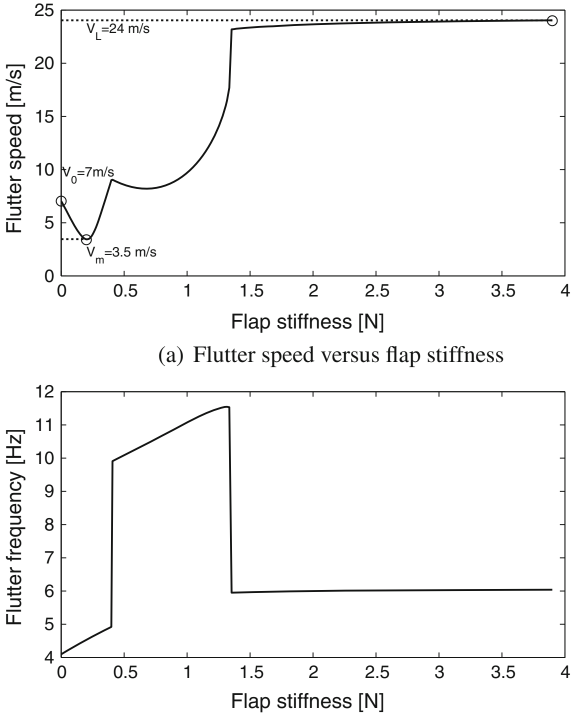
The procedure to detect necessary conditions for LCO in the plant can thus be initialized. A fundamental harmonic solution for the flap rotation β=$\beta_s \sin(\omega t)$ is assumed and from Eq. (10) the corresponding value of N^F is obtained (the freeplay size δ is fixed). This enables to calculate the quasi-linear stiffness K_β^{QL} from Eq. (14) and thus evaluate through eigenvalue analysis the corresponding flutter speed V_f, defined as the lowest airspeed which drives the system unstable, and the corresponding flutter frequency ω_f, as outlined in Sect. 2.2 for a value of the stiffness corresponding to K_β^L. Due to the existing relation between K_β^{QL} and β_s, the results can then be shown using graphs K_β^{QL}-V_f and K_β^{QL}-ω_f if the focus is on linear flutter features, or analogously using V_f-β_s and V_f-ω_f if the LCO phenomenon is emphasized. As stressed later on, these two representations are directly related since they both originate from the procedure just described. The DF method is instrumental in guaranteeing the connection and enabling to transfer the information coming from multiple linear flutter analyses to LCO characterization.

Figure 7a shows the values of flutter speed corresponding to a variation of the flap stiffness between 0 and the linear value K_β^L (that is, as the associated describing function N^F varies from 0 and 1). Important airspeeds are highlighted: V_0 is the flutter speed for zero stiffness of the control surface; V_m is the minimum flutter speed, attained for $K_\beta^{QL} \approx 0.2$ N; V_L is the linear flutter speed (determined in Fig. 2). Figure 7b shows the values of the flutter frequencies.

It is worth noticing, recalling Fig. 7b, that two distinctive flutter frequencies are detected, as the value of the flap stiffness is varied. This aspect is important for what will be considered in Sect. 5, thus it is here further investigated.

In Fig. 8 it is shown a plot similar to the one reported in Fig. 7a, but in this case considering *all* the airspeeds which correspond to a crossing of the imaginary axis, and not only the lowest (i.e. V_f). It is apparent from the graph that for a large range of flap stiffness (0 N< K_β^{QL} <1.3 N) both pitch and plunge modes experience instability. In particular, three regions can be clearly detected: in the first one, characterized by $K_\beta^{QL} \in [0, 0.4]$ N, the plunge mode is the first to go unstable (flutter frequency ≈ 4–5 Hz); in the second region, $K_\beta^{QL} \in [0.4, 1.3]$ N, the pitch mode is responsible for the lowest unstable speed (flutter frequency ≈ 10–11 Hz); finally, for $K_\beta^{QL} \in [1.3\text{ N}, K_\beta^L]$, only the plunge mode goes unstable (flutter frequency = 6 Hz). The unstable eigenvalue is associated to its elastic mode through the application of the Modal Assurance Criteria [1]. The abrupt jumps in Fig. 7b

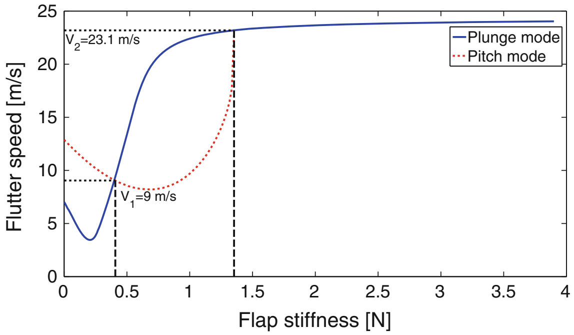
Fig. 7 Flutter speed and frequency versus flap stiffness as $K_{\beta}^{QL} \in [0, K_{\beta}^{L}]$



(a) Flutter speed versus flap stiffness

(b) Flutter frequency versus flap stiffness

Fig. 8 Flutter speed versus flap stiffness associated at each elastic mode by means of MAC



has thus to be ascribed to the change in the elastic mode reaching earlier the flutter condition, which depends on the value of flap stiffness.

Based on the previous analyses, the DF method enables to infer conclusions about the nonlinear response of the system due to the freeplay, i.e. amplitude, frequency and stability of LCOs. In particular, the conditions of neutral stability for the quasi-linearized system (reported in Fig. 7a) are associated to LCOs of the nonlinear system, with amplitude obtained from Eq. (14) and frequency provided by Fig. 7b. In Fig. 9

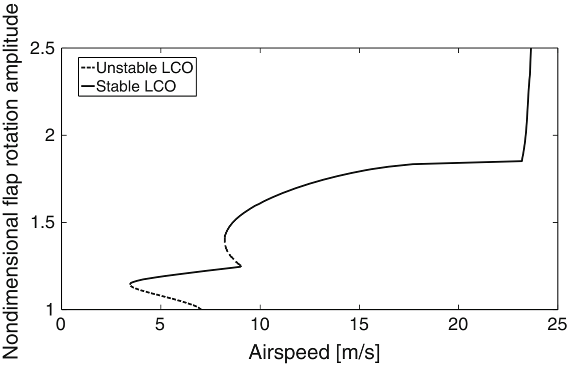


Fig. 9 Flap rotation LCO amplitude $\frac{\beta_s}{\delta}$ against airspeed. Stable and unstable oscillations highlighted

the nondimensional flap rotation amplitude $\frac{\beta_s}{\delta}$ is plotted against the airspeed. Stable and unstable oscillations are depicted respectively with a continuous and dashed lines.

The shown results are in good agreement with what reported in the aforementioned references applying DF method to the same test case. The applicability of the filter hypothesis (Eq. 7) is thoroughly addressed in [18]. In this work the Harmonic Balance, a refinement of DF method where also higher harmonics than the fundamental one can be retained, is employed. Two cases are studied, with respectively 1 harmonic, leading to equivalent results to the DF method, and 3 harmonics considered. In the latter case, the predictions almost perfectly match the results obtained through nonlinear time-marching, which is taken as the reference result. When only the first harmonic is considered, some discrepancy can be detected in the LCO branch of Fig. 9 corresponding to the low frequency instability, with a higher amplitude predicted. It is also observed that test data [6], especially in terms of oscillation amplitude, are for some small airspeed ranges only approximately captured. This is ascribed in [10] to more complex limit cycles that exhibit several types of nonlinear behavior including quasiperiodicity and thus can not be correctly modelled in this framework. It is however concluded by many [3, 6, 10, 22] that DF represents a valid tool of analysis for this problem.

5 LCO Graph Tailoring

In [3] it is tackled the task to provide robust stability assessment of this benchmark problem by means of μ analysis. The chief goal is to quantitatively evaluate the degradation of the curve in Fig. 9 in terms of amplitude increase and change in critical airspeeds. Each point of the plot corresponds to a quasi-linearization of the system, and thus can be recast in the LFT framework (Eq. 12) and its stability analysed by means of μ. In this section a different perspective on the LCO study adopting robust techniques is proposed.

As an introduction, it is recalled that flutter is generally associated to catastrophic phenomena featured by oscillations with increasing amplitudes. The presence of nonlinearities tend to bound the amplitude of vibration, leading to limited amplitude flutter, generally named LCOs. These periodic oscillations, albeit less critical than a diverging response, can result in structural fatigue issues which might lead to failure, and in general undesired behavior of the airframe. There are no precise guidelines defining values of acceptable LCOs. While the overall goal is to limit the LCO amplitudes below the amplitudes caused due to turbulence during flight (and this poses a first requirement), one of the accepted industrial practices is to limit the resulting accelerations in prescribed points of the airframe [20]. These values can be related as well to the LCO amplitude. It hence emerges on the one side, the interest in quantitatively evaluating this phenomenon (e.g. as done in Sect. 4 by means of DF approach) and on the other hand to limit it. This can be achieved either by changing the design (*passive* methods) or directly during flight by feedback control (*active* methods). The chief goal of this section is to demonstrate how μ analysis can be applied to tackle the task of achieving a reduction in LCO amplitude. The classic LFT-μ framework outlined in Sect. 3.2 and Eqs. (12)–(13) is applied. The idea is to consider the quantities held in $\boldsymbol{\Delta}$ as *tailoring variables*, that is $\boldsymbol{\Delta}$ is the variable space made of parameters which can be exploited in order to achieve a better nonlinear behavior, here defined as an LCO curve featuring smaller amplitudes.

5.1 Object Definition Through DF Framework

Recalled here the connection between the two plots described in Figs. 7a and 9. Specifically, the top plot Fig. 7a represents a linear flutter analysis applied to the system as the flap stiffness is varied. This curve is used as a basis to build the graph in Fig. 9 depicting the LCO amplitude as a function of the airspeed by making use of the relation given by the DF method between the flap amplitude and the quasi-linear stiffness (Eqs. (10)–(14)). It is thus straightforward that decreasing the LCO amplitude is equivalent to moving towards the left the curve in Fig. 7a. Note also that the points lying on this curve are all featured by a condition of neutral stability (they all have one eigenvalue with zero real part).

The structured singular value is usually applied to predict a measure of the distance of a stable plant from the neutral stability condition. In this particular case, it is employed to obtain the same measure for an *unstable* plant, attributing to it the meaning of distance to regain neutral stability. This usage of μ has to be considered exceptional and, when employed, care should be put in interpreting the results. In this application, the outcome provided by μ is not used to infer conclusions on the robustness of the plant, but, as better explained later, to understand how and which parameters should be modified to achieve the specified goal. In Fig. 10 it is depicted a pictorial explanation of this concept.

The nominal plant (having one eigenvalue with positive real part) is identified by the square marker and is specified by the nominal parameters of the system, the

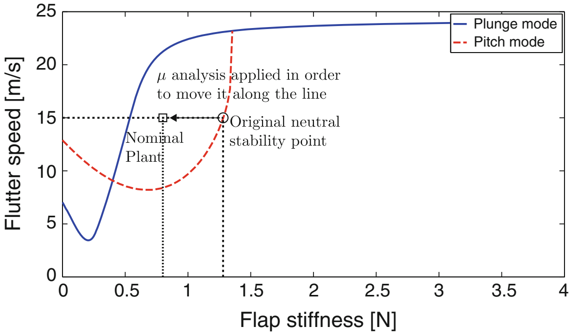


Fig. 10 Application of μ analysis to modify the LCO plot: definition of the nominal plant and depiction of the sought change in position of the neutral stability point

definition of an airspeed V and a value for the quasi-linear flap stiffness K_β^{QL}. The peak value found with μ analysis gives therefore the lowest size of perturbation matrix which leads to a neutral stability condition. After the perturbation pointed out by μ is applied to the system's parameters, the plant marked with the circle loses the condition of neutral stability, which is *allocated* to the one with the square. This graphically also means that the dashed curve, as desired, has moved towards the left, and hence the associated LCO exhibits smaller amplitude.

An important feature of this strategy is that it allows to specify the desired reduction in LCO amplitude at a certain airspeed V by selecting the value of K_β^{QL} at which the neutral stability condition is allocated. Moreover, the worst-case scenario nature of μ analysis turns to have here a very practical consequence: the critical perturbation matrix $\boldsymbol{\Delta}^{cr}$ (associated to the lower bound analysis) represents the *smallest* effort in terms of change of nominal design, because it shows the combination of parameters leading to the required goal with the overall minimum perturbation size. For this reason it is here referred to as an *optimal passive* design remedy.

Figure 10 recalls that there might be two different modes going unstable depending on the values of the flap stiffness. The selected plant for the analysis showed in the plot, for example, is *in between* the two instabilities. As a result, the combination of design parameters allowing a reduction in amplitude of the LCO associated to the pitch mode (dashed curve shifted left) could lead to an increase of the one associated to the plunge (continuous curve shifted right), with eventually only a partial or negligible objective achievement. Once defined the variable space, a μ-based sensitivity analysis as the one proposed in [13] is performed in order to detect which parameters affect more the pitch instability and less the plunge one and thus make effective and efficient design changes.

Finally, as a constraint on this tailoring, it is prescribed that the proposed improvement should not worsen the overall stability of the system, which can be characterized by means of the values of V_m and V_L (defined in Fig. 7a).

5.2 Application of μ for the LCO Tailoring

This procedure starts with the definition of the nominal plant and the block $\boldsymbol{\Delta}$. The last task consists in the definition of the variable space for the tailoring. In the showcased example it includes terms of structural mass $\mathbf{M_s}$ and stiffness $\mathbf{K_s}$ matrices, but in principle no limitations exist for this choice. The following (each one with a range of variation of 20% from the nominal value) are selected: S_α, I_α, I_β, K_h and K_α. The LFT describing the problem is defined by the block:

$$\boldsymbol{\Delta}^{7,R} = \mathrm{diag}(\delta_{S_\alpha}\mathbf{I}_2, \delta_{I_\alpha}, \delta_{I_\beta}\mathbf{I}_2, \delta_{K_h}, \delta_{K_\alpha}) \tag{15}$$

where the size of the uncertainties (total $\boldsymbol{\Delta}$ dimension) and their nature (R for real) is recalled in the superscripts. $\mathbf{I}_n$ indicates the identity matrix with size n (for repeated uncertainties). For what concerns the definition of the nominal plant, an airspeed is selected such that it lies in a region characterised by high LCO amplitude, for example $V = 15\frac{\mathrm{m}}{\mathrm{s}}$. Then a value of flap stiffness is chosen, which unequivocally defines the sought LCO amplitude for that operating point. A value of 0.85 N is taken, and hence the nominal plant corresponds to the one depicted with the square marker in Fig. 10.

The following step consists in performing the μ-based sensitivity analysis in order to understand which parameters among those in Eq. (15) affect the pitch instability responsible for the high amplitude branch, and how to modify those in order to achieve the goal. It is worth noticing that what is performed here consists basically in estimating the LCO amplitude sensitivity with respect to the parameters hold in $\boldsymbol{\Delta}$. In an active control perspective, this study could drive the design of the feedback control law to actively reduce the LCO amplitudes and possibly eliminate LCOs. The importance of this step is acknowledged in literature. In [21] for example this is tackled differentiating the equations based on the HB method and obtaining analytical sensitivities in the framework of the typical section. The method here adopted (not fully shown here due to space constraints) is believed to be less subject to the complexity of the plant analysed since it does not rely on analytical formulations, but it can be straightforwardly applied once the problem is recast in this framework. Based on the information provided by the sensitivity analysis, the following conclusions are inferred:

- The parameters that affect the pitch instability, but that do not affect the plunge one, can be advantageously taken as suggested by the worst-case analysis so as to achieve the objective, e.g. $\delta_{I_\alpha} = 1$ and $\delta_{I_\beta} = -1$
- The parameters that do not affect the pitch instability, but do affect the plunge one, can be taken with opposite sign to counteract potential undesired effect on it, e.g. $\delta_{K_h} = -1$
- The parameters that affect in the same way the two instabilities (e.g. K_α) have to be carefully considered because of the multiple effects they may have.

With respect to the last point, it is worth remembering that this *optimization* process is subject to the constraint of preserving similar values for V_m and V_L (which are

Table 2 Change in the tailoring variables to achieve the LCO tailoring

	Design 1 (%)	Design 2 (%)	Design 3 (%)
K_α	0	−20	+20
K_h	−20	−20	−20
I_α	+20	+20	+20
I_β	−20	−20	−20
S_α	0	0	0

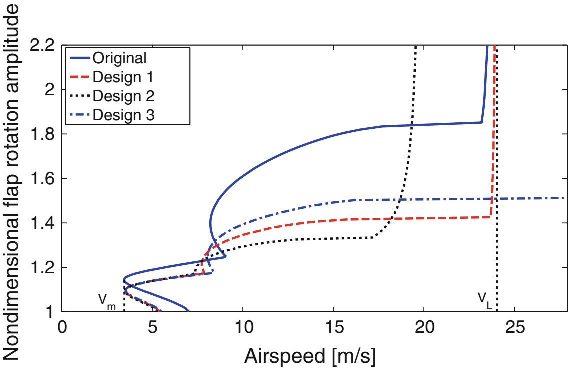


Fig. 11 Flap rotation LCO amplitudes $\frac{\beta_s}{\delta}$ after the values of the design variables are modified according to Table 2 and comparison with the original graph

assumed as indicators of the overall stability of the system). From this perspective, it should be recalled from Fig. 8 that both speeds are associated to the plunge instability. It is thus predictable that an adverse choice of parameters for the latter (as it is $K_\alpha = -1$ from the worst-case analysis) is likely to worsen this scenario. The variable K_α represents hence a trade-off between two different objectives: on the one side further decreasing the LCO amplitude, and on the other preserving the airspeed range where LCO occurs. Built on these reflections, three different designs are considered, see Table 2. In the table are expressed the suggested change of the variables as percentage from the nominal value. S_α is unaltered since it barely affects any of the instabilities and thus would represent an ineffective design change. K_h, I_α and I_β are modified according to the previous comments. The three designs differ only on the pitch stiffness K_α, in order to observe its global effect. The corresponding LCO amplitude curves are plotted in Fig. 11 against the original case.

All the proposed designs achieve the main goal of reducing the LCO amplitude as well as fulfilling the requirement of maintaining the same lowest LCO occurrence speed V_m, as shown by a comparison to the original curve. Design 2 reaches the minimum LCO amplitude among the curves but presents, as side effect, a drastic reduction in the value of V_L. This is in line with the previous analyses which suggested a reduction in K_α as beneficial to lower the LCO amplitude. They also pointed out that the same change would affect detrimentally the plunge stability, responsible for the linear flutter speed. Design 3 goes in the opposite direction for both objectives since it increases the value of this stiffness parameter. Finally Design 1, which does not alter the value of K_α, seems the best compromise since on one side it maintains the same linear flutter speed, and on the other it achieves a noticeable amplitude reduction. Of course other designs could be proposed which contemplate *absolute* variations of K_α smaller than 20% and thus span solutions between Design 1 and Design 3.

6 Conclusions

This work investigates a methodology to combine linear modeling of aeroelastic systems with Describing Function to take into account control surface freeplay and μ to perform quasi-linear analysis with uncertainties in the system. The contribution of the work is twofold.

The outcome of nominal analyses is presented, showing a good correlation with other benchmark results. The implications of using DF method as tool for nonlinear assessments are discussed. Moreover, it is stressed how DF is instrumental in guaranteeing the connection between the conditions of neutral stability evaluated in the quasi-linearized system and the conclusions inferred about the nonlinear response of the system in terms of onset, amplitude, frequency and stability of LCO.

The last section of the paper shows a possible application of μ analysis to actively or passively (depending on how the suggested methodology is implemented) modify the properties of the nonlinear response. In particular, μ is employed to understand which changes in the parameters values are more crucial to achieve a decrease of the LCO amplitude with the constraint to not affect the critical speeds of the system. The outlined procedure features a dual usage of this technique. It is first exploited its worst-case capability to identify the combination of parameters leading to the neutral stability condition (this, as explained, is equivalent to moving the LCO curve) with the smallest departure from the nominal values (optimal design changes). Secondly, a μ-based sensitivity is applied to establish which are, inside the variable space considered, the most effective and efficient parameters for the prescribed goal. The results in terms of proposed designs showcase the potential of the methodology.

Acknowledgements This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636307, project FLEXOP.

References

1. Allemang RJ, Brown DL (1982) A correlation coefficient for modal vector analysis. In: 1st International modal analysis conference
2. Anderson MR (1998) Pilot-induced oscillations involving multiple nonlinearities. J Guid Control Dyn 21(5):786–791

3. Baldelli DH, Lind RC, Brenner M (2005) Robust aeroservoelastic match-point solutions using describing function method. J Aircr 42(6):1597–1605
4. Bisplinghoff RL, Ashley H (1962) Principles of aeroelasticity. Wiley, New York
5. Borglund D (2004) The μ-k method for robust flutter solutions. J Aircr 41(5):1209–1216
6. Conner MD, Tang DM, Dowell EH, Virgin LN (1997) Nonlinear behavior of a typical airfoil section with control surface freeplay: a numerical and experimental study. J Fluids Struct 11(1):89–109
7. Danowsky B, Thompson PM, Kukreja S (2013) Nonlinear analysis of aeroservoelastic models with free play using describing functions. J Aircr 50(2):329–336
8. Dowell E, Edwards J, Strganac T (2003) Nonlinear aeroelasticity. J Aircr 40(5)
9. Gelb A, Vander Velde WE (1968) Multiple-input describing functions and nonlinear system design. McGraw-Hill, New York
10. Gordon JT, Meyer EE, Minogue RL (2008) Nonlinear stability analysis of control surface flutter with freeplay effects. J Aircr 45(6):1904–1916
11. Hodges DH, Pierce GA (2011) Introduction to structural dynamics and aeroelasticity, 2nd edn. Cambridge aerospace series. Cambridge University Press, New York
12. Iannelli A, Marcos A, Lowenberg M (2016) comparison of aeroelastic modeling and robust flutter analysis of a typical section. In: IFAC ACA, August 2016
13. Iannelli A, Marcos A, Lowenberg M (2016) Modeling and robust body freedom flutter analysis of flexible aircraft configurations. In: IEEE MSC, September 2016
14. Karpel M (1981) Design for active and passive flutter suppression and gust alleviation. Technical report 3482, NASA
15. Khalil HK (1996) Nonlinear systems. Prentice Hall, Upper Saddle River
16. Kotikalpudi A, Pfifer H, Seiler P (2016) Sensitivity of robust flutter boundary to model uncertainties in aeroservoelastic systems. In: AIAA Scitech, January 2016
17. Lind R, Brenner M (2012) Robust aeroservoelastic stability analysis. Advances in industrial control. Springer, London
18. Liu L, Dowell EH (2005) Harmonic balance approach for an airfoil with a freeplay control surface. AIAA J 43(4):802–815
19. Packard A, Doyle J (1993) The complex structured singular value. Automatica 29(1)
20. Pitt DM, Bansal P (2014) Uncertainties in control surface free-play and structural properties and their effect on flutter and LCO. In: AIAA Scitech conference, January 2014
21. Shukla H, Patil M (2016) Control of limit cycle oscillation amplitudes in nonlinear aerelastic systems using nonlinear normal modes. In: AIAA Scitech conference, January 2016
22. Tang D, Dowell EH, Virgin LN (1998) Limit cycle behavior of an airfoil with a control surface. J Fluids Struct 12(7):839–858
23. Theodorsen T (1935) General theory of aerodynamic instability and the mechanism of flutter. Technical report 496, Naca
24. Wiggins P (2003) Introduction to applied nonlinear dynamical systems and chaos. Springer, New York
25. Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall Inc., Upper Saddle River

Development of an Automatic Flight Path Controller for a DA42 General Aviation Aircraft

Erik Karlsson, Simon P. Schatz, Thaddäus Baier, Christoph Dörhöfer, Agnes Gabrys, Markus Hochstrasser, Christoph Krause, Patrick J. Lauffs, Nils C. Mumm, Kajetan Nürnberger, Lars Peter, Volker Schneider, Philip Spiegel, Lukas Steinert, Alexander W. Zollitsch and Florian Holzapfel

1 Introduction

The control of the aircraft flight path, i.e. the kinematic course and climb angles, as well as the aerodynamic speed along the flight path, is an essential part of all modern flight control systems for both manned and unmanned aircraft, either for the execution of direct medium-level commands from the pilot or ground operator, or, as in traditional, cascaded control structures, as basis for higher-level trajectory following and navigation control modes. Especially for unmanned applications, robust, high-authority, and high-bandwidth flight path control is of interest, in order to allow more aggressive maneuvering, thus going beyond traditional low-bandwidth autopilot operation and waypoint based trajectory following.

The Institute of Flight System Dynamics (FSD) at the Technische Universität München (TUM) has developed a highly modular flight guidance and control system for manned and unmanned aircraft, that provides the full range of typical and beyond state-of-the-art automated flight control functionalities in order to meet most of the real-world operational requirements for manned and unmanned aircraft systems: automatic takeoff and landing, voice commanded flight management and trajectory generation systems [11, 12], automatic trajectory [9, 10] and flight path controllers [3], as well as inner loop baseline and automatic thrust controllers. The modularity and configurability allow easy adjustment and adaptation to new aircraft configurations.

The paper at hand focuses on the development of the full envelope, full authority, full bandwidth automatic flight path controller, initially presented in [3]. The contributions of this paper include: extended derivation of the flight path control laws;

E. Karlsson (✉) · S.P. Schatz · T. Baier · C. Dörhöfer · A. Gabrys · M. Hochstrasser ·
C. Krause · P.J. Lauffs · N.C. Mumm · K. Nürnberger · L. Peter · V. Schneider ·
P. Spiegel · L. Steinert · A.W. Zollitsch · F. Holzapfel
Institute of Flight System Dynamics, TU München,
Boltzmannstr. 15, 85748 Garching, Germany
e-mail: erik.karlsson@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_7

overview of requirements, design objectives and verification activities; extended flight test results.

As the command variables of the controller are the aircraft flight path, the control bandwidth is determined by the achievable path curvature, i.e. incremental specific forces in the kinematic frame. Thus, for optimal tracking performance, the flight path controller utilizes a generic interface to aircraft-specific inner loop controllers for the control of the normal and lateral, i.e. path transverse, and linear, i.e. path tangential, specific forces. Further, such specific force based inner loop controllers provide good disturbance rejection already at the acceleration level, before disturbances manifest themselves into attitude, velocity and position errors, enhancing path control performance.

The spectrum of current and planned target aircraft configurations for the modular flight control system in general, and the automatic flight path controller in particular, requires a high degree of parameter design and performance assessment automation, for easy adaptation to new configurations. A parametrized configuration independent implementation, as well as generic, configuration independent simulation, design and assessment routines and environments, parametrized with configuration specific requirements and design data, ensure the reusability and rapid adaptation to new aircraft configurations.

2 State-of-the-Art

Traditional Single-Input/Single-Output (SISO) autopilot control structures, with independent control of each axis, neglect the coupling between the path variables - an acceptable approach for low bandwidth systems with high settling times, and low actuation authority, but inherently prone to dangerous control objective conflicts, as arbitrary flight path and speed targets cannot be maintained with saturated energy rate control. More modern integrated control approaches, such as Total Energy Control System (TECS), see e.g. [4, 5, 7] use coupled energy flow rate and distribution control to decouple flight path and speed tracking, and to integrate airspeed-based envelope protections [6] in order to prevent loss of control. For successful decoupling, the pitch control bandwidth has to match that of the typically much slower thrust dynamics, thus unnecessary reducing flight path control bandwidth. Such approaches may be sufficient for low-bandwidth autopilot applications, e.g. for larger, manned aircraft. For more agile aircraft, high-bandwidth flight path maneuvering with temporary sacrifice of speed tracking may be preferred over always coordinated flight path and speed tracking.

The flight path controller presented in this paper uses dynamic coupling of the control planes to provide deterministic control objective conflict resolution, protection of the energy integrity of the aircraft, and robustness against non-achievable flight path commands, by means of active energy rate and force prioritization between the vertical, lateral and energy control. The paper focuses on the nominal path control functions; the energy rate and force prioritizations active at the edges of the envelope are outside the scope of this paper.

3 Controller Environment and Flight Test Platform

3.1 Modular Application Architecture

Figure 1 shows a principal overview of the structure of the modular flight guidance and control system application. Depending on the operational requirements for a given target platform, and the therefore necessary functionalities, the corresponding functional modules can be integrated to a configuration-specific control application. The modularity and configurability aim at keeping as much as possible identical for different platforms in order to avoid significant rework. The functional modules include a system automation, interfacing with external command sources and handling the overall operating modes, flight plan trajectory generation and automatic takeoff and landing modules generating commands for the outer loops of the auto-flight system, i.e. the trajectory and flight path control modules. The commands from the outer loop are evaluated depending on the active operating modes and the respective commands are transformed and forwarded to the inner loop controller/auto-thrust, which compute the final commands to the Actuator Control Electronics (ACE).

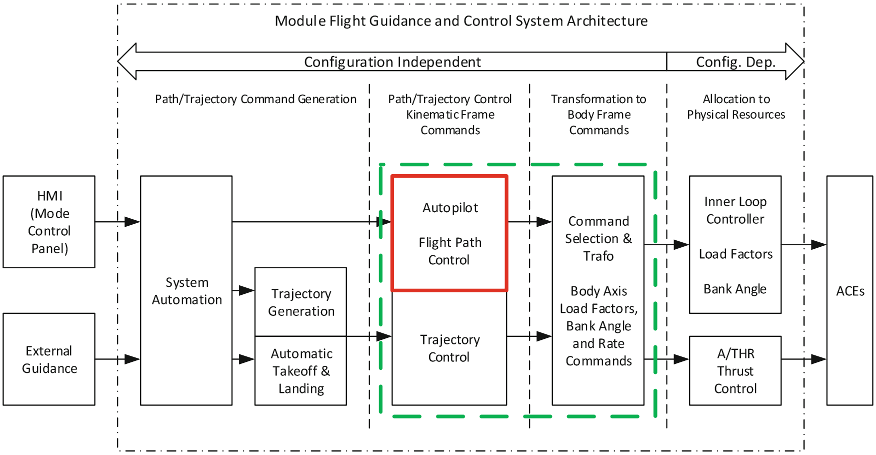

Module Flight Guidance and Control System Architecture
Configuration Independent
Config. Dep.
Path/Trajectory Command Generation
Path/Trajectory Control Kinematic Frame Commands
Transformation to Body Frame Commands
Allocation to Physical Resources
HMI (Mode Control Panel)
External Guidance
System Automation
Trajectory Generation
Automatic Takeoff & Landing
Autopilot
Flight Path Control
Trajectory Control
Command Selection & Trafo
Body Axis Load Factors, Bank Angle and Rate Commands
Inner Loop Controller
Load Factors
Bank Angle
A/THR Thrust Control
ACEs

Fig. 1 Modular flight guidance and control system architecture

Fig. 2 The DA42 M-NG of the Institute of Flight System Dynamics

3.2 The DA42 OE-FSD Demonstrator Platform

For validation and verification of experimental control algorithms, Fly-By-Wire (FBW) technologies and equipment for manned and unmanned applications, the Institute of Flight System Dynamics has, together with the manufacturer, extensively modified a Diamond Aircraft Industries DA42 M-NG (Multi-Purpose Platform, Next Generation) aircraft, to serve as flying testbed, see Fig. 2. A digital, experimental flight control system is installed parallel to the normal mechanical controls, interfacing with the mechanical controls via both electrical and overload clutches. A safety system ensures the safe engagement and disengagement of the experimental flight control system.

4 Controller Requirements and Design Objectives

4.1 Functional and Operational Requirements

The functional and operational requirements define the flight control functions, which beyond the direct control of the flight path includes additional functions such as speed by pitch/flight level change, altitude capture and hold, pitch hold and radio navigation modes, all outside the scope of this paper, the mode transition logics, activation and deactivation logics, limitations and interfaces, as well as human-machine-interface, i.e. mode control panel, and operational concept. The functional and operation requirements are the basis for the controller design, and for the implementation of the design model.

4.2 Performance Requirements

For each flight control function, desired and adequate performance are defined. The desired performance corresponds to a design goal during the gain and parameter design, possibly not achievable for all aircraft configurations and load conditions over the envelope. The adequate performance defines the minimum acceptable performance to be met for all configurations, load conditions and points in the operational envelope. Due to the variety of possible application platforms, a set of all-encompassing desired performance criteria cannot be stated, but must rather be defined depending on the target aircraft configuration and its operational aspects; for a manned aircraft, smooth transitions to new flight path commands may be preferred before tracking bandwidth; for an unmanned aircraft, high path control bandwidth utilizing the full capabilities of the aircraft dynamics may be the primary objective. For certain control variables, e.g. speed by thrust control, the tracking and disturbance performance is usually a trade-off with actuation activity. Applicable aerospace standards specify acceptable levels of adequate system performance for certain functions, e.g. [8], generally adhered to for the design of automatic flight control systems for manned and unmanned aircraft.

4.2.1 Tracking and Disturbance Requirements

Tracking requirements include specifications of maximum command response overshoot, settling times and accuracy in smooth and turbulent air. Table 1 lists some exemplary desired and adequate performance requirements for the flight path control functions. The accuracy requirements for random turbulent air are e.g. given as root-mean-square deviations, at specified root-mean-square gust intensities as function of the altitude and a cumulative exceedance probability corresponding to light turbulence.

4.2.2 Robustness and Sensitivity Requirements

As with the adequate performance for the flight control functions, requirements regarding control loop stability margins and sensitivity against uncertainties in critical

Table 1 Exemplary desired and adequate path tracking accuracy requirements

Function	Capture Overshoot		Accuracy (Smooth Air)		Accuracy (Turb. Air)	
	Desired	Adequate	Desired	Adequate	Desired	Adequate
Flight path angle	2%	10%	0.1°	0.5° *	1° RMS	5° RMS *
Track angle	0.5°	1.5° *	0.1°	0.5° *	1° RMS	5° RMS *
Speed by thrust	2 kts	4 kts	2 kts	4 kts	4 kts RMS	10 kts RMS

*Adequate performance derived from AS94900 [8] aerodynamic parameters can be found in e.g. [8]. In the frequency range of the rigid body dynamics, i.e. up to the first aeroelastic mode, a gain margin of at least 6 dB and phase margin of at least 45° are required for all aerodynamically closed loops, measured at the actuator inputs. Assessment of robustness against uncertainties in the aerodynamic data set shall be performed using variations of critical stability derivatives, e.g. $C_{m,0}$, $C_{m\alpha}$ and $C_{n\beta}$, of up to 20 percent, thereby maintaining at least half of the stability margins for the nominal system.

5 Controller Design

This section describes the flight dynamics preliminaries and the structure of the flight path controller. The controller structure is based on an inversion of the translational equations of motions, i.e. the aircraft path dynamics, with reference models for the desired command tracking dynamics, error controllers for disturbance rejection, and feedback hedging signals to account for inner loop and actuation dynamics and saturations.

5.1 Translational Equations of Motion

The differential equations describing the motion of the aircraft may be derived from the second law of Newton, according to which the rate of change of the linear momentum $\mathbf{p} \in \mathbb{R}^3$ of an object in an inertial frame is equal to the net force $\sum \mathbf{F} \in \mathbb{R}^3$ acting upon it,

$$\sum \mathbf{F} = \left(\frac{d}{dt}\right)^I (\mathbf{p}(t))^I = \left(\frac{d}{dt}\right)^I \int_m \left(\mathbf{V}^P(t)\right)^I \cdot dm, \tag{1}$$

where $(\mathbf{V}^P)^I(t) \in \mathbb{R}^3$ is the velocity of a point P with mass dm, relative to the Earth-Centered Inertial (ECI) frame I. It is assumed that the aircraft is a rigid body, i.e. the relative position between points on the aircraft does not change over time, $(\dot{\mathbf{V}}^{RP})^B = \mathbf{0}$, and that the influence of the mass flow onto the impulse of the aircraft can be neglected, $\dot{m}(\mathbf{V}^P)^E \approx 0$. For a given point P of the rigid body aircraft, its position vector $\mathbf{r}^P \in \mathbb{R}^3$, is given by

$$\left(\mathbf{r}^P\right) = \left(\mathbf{r}^R\right) + \left(\mathbf{r}^{RP}\right), \tag{2}$$

where $\mathbf{r}^R \in \mathbb{R}^3$ is the position of the aircraft reference point R. The corresponding velocity relative to the Earth-Centered Inertial (ECI) frame is given by

$$\left(\mathbf{V}^P\right)^I = \left(\mathbf{V}^R\right)^E + \left(\boldsymbol{\omega}^{IE}\right) \times \left(\mathbf{r}^R\right) + \left(\mathbf{V}^{RP}\right)^B + \left(\boldsymbol{\omega}^{IB}\right) \times \left(\mathbf{r}^{RP}\right) \tag{3}$$

where $(\mathbf{V}^R)^E$ is the velocity relative to the center of the Earth, $(\boldsymbol{\omega}^{IE})$ the rotation of the Earth, $(\mathbf{V}^{RP})^B$ is the relative velocity of the point P, and $(\boldsymbol{\omega}^{IB})$ the rotation of the aircraft relative to the inertial frame. Differentiating the velocity vector with respect to the North-East-Down (NED) frame O, Newton's second law can be written as

$$\sum \left(\mathbf{F}^R\right) = m \left(\dot{\mathbf{V}}_K^R\right)^{EO} + m\mathbf{a}_{add}, \tag{4}$$

where $\mathbf{a}_{add}$ is a term containing all additional accelerations such as the Coriolis term, the acceleration due to the transport rate, and the acceleration due to difference of reference point R and center of gravity G. For the applications of the flight path controller, assuming a flat, non-rotating earth is sufficient, and if further choosing the center of gravity as reference point, $\mathbf{a}_{add} = \mathbf{0}$, and thus subsequently neglected.

For the analysis of the flight path dynamics, a formulation of the translational equations of motion considering the kinematic frame K is desired. Hence, the acceleration relative to the North-East-Down (NED) frame O is split up to include the kinematic acceleration due to flight path dynamics,

$$\left(\dot{\mathbf{V}}_K^G\right)^{EO} = \left(\dot{\mathbf{V}}_K^G\right)^{EK} + \left(\boldsymbol{\omega}^{OK}\right) \times \left(\mathbf{V}_K^G\right)^E, \tag{5}$$

where $(\dot{\mathbf{V}}_K^G)^{EK}$ is the kinematic acceleration relative to the kinematic frame, and $(\boldsymbol{\omega}^{OK}) \times (\mathbf{V}_K^G)^E$ is the acceleration due to a change in flight path, i.e. a rotation of the kinematic frame K with respect to the NED frame O. Using polar coordinates in the kinematic frame, i.e. the kinematic velocity magnitude $V_K^G \in \mathbb{R}$, the course angle $\chi_K^G \in \mathbb{R}$ and the flight path angle $\gamma_K^G \in \mathbb{R}$, the kinematic equations of motion can be given as

$$\dot{V}_K^G = \Delta f_{x,K}, \qquad \Delta f_{x,K} \triangleq \frac{(X_A^G + X_P^G)_K}{m} - g \sin \gamma_K, \tag{6}$$

$$\dot{\chi}_K = \frac{f_{y,K}}{V_K^G \cos\gamma_{\mathrm{K}}}, \qquad f_{y,K} \triangleq \frac{(Y_A^G + Y_P^G)_K}{m}, \tag{7}$$

$$\dot{\gamma}_K = \frac{-\Delta f_{z,K}}{V_K^G}, \qquad \Delta f_{z,K} \triangleq \frac{(Z_A^G + Z_P^G)_K}{m} + g \cos \gamma_K, \tag{8}$$

with $m \in \mathbb{R}$ denoting the mass of the aircraft, $g \in \mathbb{R}$ the acceleration due to gravity, and $(X_A^G)_K$, $(Y_A^G)_K$, $(Z_A^G)_K \in \mathbb{R}$ as well as $(X_P^G)_K$, $(Y_P^G)_K$, $(Z_P^G)_K \in \mathbb{R}$ are the components of the aerodynamic and propulsion forces at the center of gravity, respectively, noted in the kinematic frame.

However, the task of the flight path controller is to control the aerodynamic velocity $V_A^G \in \mathbb{R}$ along the flight path, and not the kinematic. The kinematic velocity is the vector sum of the aerodynamic velocity vector and the wind velocity vector,

$$\left(\mathbf{V}_A^G\right)^E = \left(\mathbf{V}_K^G\right)^E - \left(\mathbf{V}_W^G\right)^E, \tag{9}$$

thus, for a constant wind field,

$$\left(\dot{\mathbf{V}}_A^G\right)^{EO} = \left(\dot{\mathbf{V}}_K^G\right)^{EO} - \left(\dot{\mathbf{V}}_W^G\right)^{EO} = \left(\dot{\mathbf{V}}_K^G\right)^{EO}, \tag{10}$$

and

$$\left(\dot{\mathbf{V}}_A^G\right)^{EA} + \left(\boldsymbol{\omega}^{OA}\right) \times \left(\mathbf{V}_A^G\right)^{E} = \left(\dot{\mathbf{V}}_K^G\right)^{EK} + \left(\boldsymbol{\omega}^{OK}\right) \times \left(\mathbf{V}_K^G\right)^{E}, \tag{11}$$

which gives for the aerodynamic velocity magnitude,

$$\dot{V}_A^G = \Delta f_{x,A}, \qquad \Delta f_{x,A} \triangleq \frac{(X_A^G + X_P^G)_A}{m} - g \sin \gamma_A^G, \tag{12}$$

where $\gamma_A^G \in \mathbb{R}$ is the aerodynamic flight path angle, and $(X_A^G)_A$ and $(X_P^G)_A$ are the aerodynamic and propulsion forces in the aerodynamic frame A.

5.2 Inversion of the Path Dynamics

The objective of the nonlinear dynamic inversion, also known as input/output linearization or feedback linearization [13], is to find a non-linear state transformation of a non-linear plant so that the transformed system has a linear input-output dynamics. The order of the feedback linearized system is referred to as the relative degree, and determines up to which derivative the dynamics of the system is directly controllable. The application to the path dynamics is straight forward. The path dynamics can be considered to be of relative degree one, i.e. it takes one differentiation of the path dynamics variables in order for the virtual plant inputs (i.e. the specific forces) to appear in the equations of motion, see Eqs. (6)–(8). Thus, an inversion architecture corresponding to the relative degree of the path dynamics is desired. With the pseudo control vector $\boldsymbol{\nu} = [\nu_{\dot{V}_A}, \nu_{\dot{\chi}_K}, \nu_{\dot{\gamma}_K}]^T$, the control law for the path loops becomes

$$\mathbf{u_C} = \begin{bmatrix} \Delta f_{A,K,C} \\ f_{y,K,C} \\ \Delta f_{z,K,C} \end{bmatrix} = \begin{bmatrix} \nu_{\dot{V}_A} \\ V_K \cos \gamma_K \nu_{\dot{\chi}_K} \\ -V_K \nu_{\dot{\gamma}_K} \end{bmatrix}. \tag{13}$$

5.3 Reference Models for the Path Dynamics

The path controller includes linear second order reference models for the airspeed, course/heading angle and flight path angle/vertical speed, see Fig. 3. As stated in the previous section, a first order reference model would be sufficient to produce the required reference state and derivatives for the path variables. However, in order

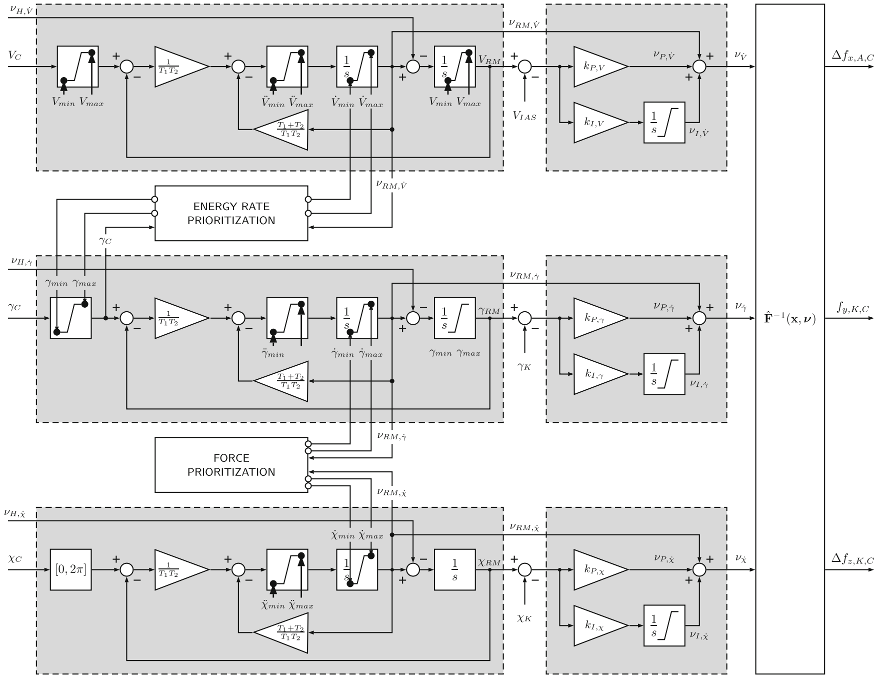


Fig. 3 Reference model based path control structure with energy rate and force prioritization cross-feeds

to also smoothen the reference state derivatives, second order reference models are applied according to

$$\dot{\nu}_{RM} = \frac{1}{T_1T_2}\left(y_C - y_{RM}\right) - \frac{T_1+T_2}{T_1T_2}\nu_{RM}, \tag{14}$$

$$\dot{y}_{RM} = \nu_{RM} - \nu_H. \tag{15}$$

where T_1 and T_2 are the reference model time constants, y_C is the flight path command, y_{RM} and $\dot{y}_{RM}$ the reference model states, ν_{RM} the pseudo control to the inversion, and

$$\nu_H = \nu - \hat{\nu}, \tag{16}$$

a hedging signal added to adjust the reference model dynamic according to the plant response $\hat{\nu}$, see Sect. 5.4. The reference models include multiple limiters; limitation of the input command value, limitation of the reference state first derivative, i.e. linear acceleration, flight path angle rate (limit imposed by permissible specific normal force) and turn rate, limitation of the reference state second derivative, e.g. kinematic bank angle rate, and integrator output limitation, to account for the hedging signal which is added after the rate limiter. The time constants of each reference model are scheduled over static and dynamic pressure to account for available inner loop dynamics. The design of the time constants is based on linear models of the closed inner loop dynamics with step response overshoot as primary design criterion and closed loop stability margins as monitor criteria.

5.4 Plant Response Estimation and Pseudo-control Hedging

The consideration of the path dynamics as a system of relative degree one is justified by the fact that the path dynamics and dynamics of subsequent inner loops, i.e. the moment and actuator dynamics, are sufficiently time-scale separated. However, due to the dynamics of the inner loops, the plant will not perfectly follow the desired reference dynamics given by the reference models. This is encountered by employing PCH [1], feeding back hedging signals $\nu_H = \nu - \hat{\nu}$, i.e. the difference between command and estimated plant response, to slow down the reference model output in order to account for the reaction dynamics of the plant. From the measured specific forces in the body axis frame, the specific force vector in the kinematic frame is calculated according to

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_K = \mathbf{M}_{KB} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_B = \mathbf{M}_{KO}\mathbf{M}_{OB} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_B , \tag{17}$$

with $\mathbf{M}_{KO}\ (\gamma_K, \chi_K)$ and $\mathbf{M}_{OB}\ (\Psi, \Theta, \Phi)$ being the transformation matrices from the O into the K frame, and from the B into the O frame, respectively. The kinematic acceleration and path curvatures are then estimated using the Eqs. (6–8),

$$\hat{\nu}_{\dot{V}_K} = f_{x,K} - g \sin \gamma_K , \tag{18}$$

$$\hat{\nu}_{\dot{\chi}_K} = \frac{1}{V_K \cos \gamma_K} f_{y,K} , \tag{19}$$

$$\hat{\nu}_{\dot{\gamma}_K} = \frac{1}{V_K} \left(f_{z,K} - g \cos \gamma_K \right) . \tag{20}$$

For the control of the airspeed, the aerodynamic acceleration is to be estimated, not the kinematic; however, the aerodynamic flight path and course angles, γ_A and χ_A, are unknown without wind information. Assuming a neglectable sideslip, i.e. $\chi_A \approx \Psi$, and a horizontal, sufficiently stationary wind field, i.e. $\gamma_A \approx \arcsin(\dot{h}/V_{TAS})$, the aerodynamic acceleration can be estimated analog to Eqs. (17)–(18), with $\mathbf{M}_{AB}(\gamma_A, \chi_A = \Psi, \Theta, \Phi)$ according to

$$f_{x,A} = \left[\cos(\Theta - \gamma_A)\ \sin\Phi \sin(\Theta - \gamma_A)\ \cos\Phi \sin(\Theta - \gamma_A) \right] \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_B , \tag{21}$$

and

$$\hat{\nu}_{\dot{V}_A} = f_{x,A} - g \sin \gamma_A. \tag{22}$$

The hedging signals are feedback signals; hence, the reference models cannot be considered as open-loop, feedforward elements when considering the stability of the system. Furthermore, they have to be considered as feedback signals when assessing the system robustness and stability, see Sect. 6.

5.5 Path Loop Error Dynamics and Stabilizing Controllers

The pseudo-control hedging ensures that the reference model state is adjusted to the estimated response of the plant. However, due to a number of different reasons, e.g. neglected dynamics, sensor errors and delays, and external disturbances, the estimated response of the plant and the reference model state do not perfectly match the response of the true plant. To stabilize the error dynamics, PI feedback controllers are included, and their output added to the pseudo controls entering the inversion,

$$\nu_{EC} = k_P(y_{RM} - y) + k_I \int (y_{RM} - y)dt, \tag{23}$$

$$\nu = \nu_{RM} + \nu_{EC}. \tag{24}$$

The design of the error controller gains is based on linear models of the closed inner loop dynamics with disturbance settling times and closed loop stability margins as design criteria.

5.6 Energy Rate and Force Distribution Prioritization

The reference models are cross-coupled in order to provide dynamic control objective conflict resolution, by limiting acceleration and flight path commands in case of saturated energy flow, i.e. thrust control, see [2], and desired vertical and lateral path curvature in case of saturated transverse force control, i.e. normal specific force. A detailed description of the energy rate and force distribution prioritization is outside the scope of this paper.

5.7 Command Selection and Transformation

The flight path controller outputs path curvature commands to the inner loop. Consequently, just the lateral and vertical curvature commands, $f_{y,K}$ and $\Delta f_{z,K}$, respectively, are forwarded to the inner loop. For the inner loop control, these commands are transformed to a bank angle command in the lateral plane and a normal body load factor command in the longitudinal plane. First the incremental specific force commands are transformed into the O frame according to

$$\frac{1}{g}\begin{pmatrix}\Delta f_{x,C}\\ f_{y,C}\\ \Delta f_{z,C}\end{pmatrix}_O = \frac{1}{g}\mathbf{M}_{OK}\begin{pmatrix}\Delta f_{x,C}\\ f_{y,C}\\ \Delta f_{z,C}\end{pmatrix}_K \tag{25}$$

The command specified here contains only the specific forces required for curvature. Hence, the steady state specific force divided by gravity is added,

$$\frac{1}{g}\begin{pmatrix}f_{x,C}\\ f_{y,C}\\ f_{z,C}\end{pmatrix}_O = \frac{1}{g}\begin{pmatrix}\Delta f_{x,C}\\ f_{y,C}\\ \Delta f_{z,C}\end{pmatrix}_O - \begin{pmatrix}0\\ 0\\ 1\end{pmatrix}_O \tag{26}$$

The full specific force command vector is then further transformed into the intermediate body-fixed system Z,

$$\frac{1}{g}\begin{pmatrix}f_{x,C}\\ f_{y,C}\\ f_{z,C}\end{pmatrix}_Z = \frac{1}{g}\mathbf{M}_{ZO}\begin{pmatrix}f_{x,C}\\ f_{y,C}\\ f_{z,C}\end{pmatrix}_O \tag{27}$$

In order to obtain the load factor and bank angle command to the inner loop, consider the following transformation between the intermediate body-fixed (Z) and the body-fixed (B) system,

$$\frac{1}{g}\begin{pmatrix}f_{x,C}\\ f_{y,C}\\ f_{z,C}\end{pmatrix}_B = \begin{bmatrix}1 & 0 & \\ 0 & \cos\Phi & \sin\Phi\\ 0 & -\sin\Phi & \cos\Phi\end{bmatrix}\frac{1}{g}\begin{pmatrix}f_{x,C}\\ f_{y,C}\\ f_{z,C}\end{pmatrix}_Z \tag{28}$$

The x-axis is not controlled by the inner loop, since aerodynamic speed control is used. Therefore, the x-component is not considered, which results in a small error in the z-component. From the second and third rows, the following equations for the normal and lateral load factor commands can be obtained,

$$\frac{f_{y,B,C}}{g} = \frac{f_{y,Z,C}}{g}\cos\Phi + \frac{f_{z,Z,C}}{g}\sin\Phi, \tag{29}$$

$$\frac{f_{z,B,C}}{g} = -\frac{f_{y,Z,C}}{g}\sin\Phi + \frac{f_{z,Z,C}}{g}\cos\Phi. \tag{30}$$

From the requirement to fly a coordinated curve, i.e. $f_{y,B} = 0$, the required bank angle for achieving the lateral force command can be calculated from Eq. (29) by setting $f_{y,B,C} = 0$ and solving for Φ,

$$\Phi_C = \arctan\left(-\frac{f_{y,Z,C}}{f_{z,Z,C}}\right). \tag{31}$$

In order to obtain the body-fixed load factor command $f_{z,B,C}/g$, Eq. (31) rewritten as $f_{y,Z,C} = -f_{z,Z,C} \tan \Phi_C$ can be inserted into Eq. (30), which gives

$$\frac{f_{z,B,C}}{g} = \frac{f_{z,Z,C}}{g} \sin \Phi_C \tan \Phi_C + \frac{f_{z,Z,C}}{g} \cos \Phi_C = \frac{f_{z,Z,C}}{g} \frac{1}{\cos \Phi_C}. \tag{32}$$

Note that a load factor command calculated according to Eq. (32), based on a bank angle command according to Eq. (31), would lead to a velodrome-like curve, since the dynamics of the load factor is normally much faster than the bank angle dynamics, i.e. the aircraft would reach the required load factor increment for curve compensation (calculated based on the command bank angle) before the corresponding bank angle has been achieved, leading to a "pull-up then bank" behavior. This can be mitigated by calculating the load factor command based on the actual bank angle instead of the commanded, i.e.

$$\frac{f_{z,B,C}}{g} = \frac{f_{z,Z,C}}{g} \frac{1}{\cos \Phi}. \tag{33}$$

This would on the other hand result in a small delay in the load factor response. It would be possible to use a modified bank angle command $\tilde{\Phi}_C$ for the calculation of the load factor command, e.g. according to $\tilde{\Phi}_C = \Phi + \Delta T \cdot p$, where T is some time constant and p the current roll rate, or according to some function of the difference between the bank angle command according to Eq. (31) and the measured bank angle, $\tilde{\Phi}_C = f(\Phi_C - \Phi)$.

5.8 Normal and Lateral Force Control - Inner Loop

The longitudinal inner loop controller employs the normal body load factor and the pitch rate as feedback for a PI controller. The lateral inner loop controller is a MIMO control structure designed to track the bank angle command Φ_{cmd} and reduce the body lateral acceleration $f_{y,B}$ to zero. For this purpose, the roll and pitch rates are fed back along with the lateral acceleration and the bank angle command.

The current gains are chosen in such a way that the inherent dynamics of the DA42 are not changed significantly. In the lateral plane, the roll dynamics and the natural frequency of the Dutch roll are maintained while the yaw damping is increased. Furthermore, the spiral pole is set to -1 for the whole envelope. In the longitudinal plane, the pitch damping is increased and the nominal load factor command system is implemented. The tracking performance is achieved by the feedforward part while the feedback is designed primarily for disturbance rejection.

5.9 Linear Force Control - Thrust Loop

The thrust loop controls the linear specific force $\Delta f_{x,A}$ by adjusting the throttle position δ_T. The required thrust change is given by

$$\begin{aligned} \Delta T_C &= \frac{\Delta \dot{E}_{tot,C}}{V_A} = m\left(\Delta f_{x,A,C} - \Delta f_{x,A}\right) \\ &= m \cdot \left(\nu_{\dot{V}_A} - \hat{\nu}_{\dot{V}}\right), \end{aligned} \tag{34}$$

where $\nu_{\dot{V}_A}$ is the pseudo control acceleration command from the speed loop, and $\hat{\nu}_{\dot{V}}$ is the estimated kinematic acceleration (according to Eq. (18)). The required change in throttle position $\Delta\delta_{T,C}$ for a given thrust change is calculated by first taking the inverse of the thrust as function of speed to determine the maximum thrust available for the current speed, and set the throttle according to the ratio of the required and available maximum thrust. An integrator part ensures static accuracy for the thrust control in the case of engine model errors or uncertainties,

$$\Delta\delta_{T,C} = \frac{1}{\frac{\Delta T}{\Delta\delta_T}} \cdot \left(k_{P,\delta_T}\Delta T + k_{I,\delta_T}\int \Delta T dt\right). \tag{35}$$

The integrator further allows a direct feedforward throttle rate command, $\dot{\delta}_{T,C}$, e.g. from the vertical flight path loop, in order to compensate the additional required thrust during vertical path maneuvering. For initial flight testing, due to uncertain propulsion dynamics, the bandwidth of the thrust control loop was chosen quite low, in order to provide smooth throttle commands in the presence of turbulence and sufficient stability margins and robustness.

6 Controller Verification

6.1 Linear and Non-linear Control Law Assessment

The linear assessment is based on the linear models of the closed inner loop and plant dynamics, i.e. models with the flight path controller and autothrust outputs $f_{z,B,C}/g$ and $\delta_{T,C}$ as inputs, and the aircraft states as outputs. The linear flight path control structure, i.e. the reference models and error controllers, together with linearized models of the plant response estimation, inversion and command transformation for different flight conditions, are connected to full linear versions of the closed loop longitudinal and lateral control systems, respectively. The linear systems are implemented as generalized state-space models in MATLAB, with parametrized gains and switches, in order to configure the models for different control modes and simplify analysis of loop stability margins. For each loop, step responses and settling times are analyzed with respect to the requirements, as well as stability margins using Bode and Nichols charts.

The nonlinear assessment includes verification of step responses and accuracy in smooth and turbulent air. A dedicated test harness including the flight path controller, the inner loop controller, flight dynamics model, and sensor and actuation models including transfer delays, is used for automated assessment over the envelope, see Fig. 4.

6.2 Model-in-the-Loop Verification

Objective of the Model-in-the-Loop (MIL) simulation is to verify correct functionality of the controller implementation, i.e. the design model from which the application code is generated. The Model-in-the-Loop (MIL) verification uses a similar test setup as the nonlinear performance assessment, but with extended input mapping to be able to trigger every model input signal, in order to reach all states of the model. The Model-in-the-Loop (MIL) uses the MATLAB Unit Test Framework to automatize the testing, with customized functions to initialize and run the simulation model and evaluate the outputs. Test vectors are given as MATLAB time series objects, and a testing grid is defined over which the tests are to be performed. Model coverage is measured incrementally over a complete test run.

6.3 Hardware- and Aircraft-in-the-Loop Simulation

Objective of the Hardware-in-the-Loop (HIL) and Aircraft-in-the-Loop (AIL) simulation is to verify the correct functionality of the system, i.e. the controller running on its target hardware with a real-time environment simulation, of all digital, analog and discrete interfaces, and all interfacing subsystems such as actuators, data concentrator units and human-machine interfaces, in a laboratory setup, and installed in the aircraft, respectively. The flight control sensors, i.e. navigation and air data systems are simulated, but feeding the controller via real interfaces.

7 Flight Test Results

Initial flight testing was performed in early 2016, with incremental testing of first inner loop controller, then flight path controller [3], and finally waypoint based trajectory navigation [10]. Figure 5 shows results from flight testing performed in August 2016, in light turbulent conditions, from a segment of the flight test where the flight path controller was active in all three axes. Figure 5 shows good tracking performance of both the vertical and lateral flight path loops, as well as inner loop load

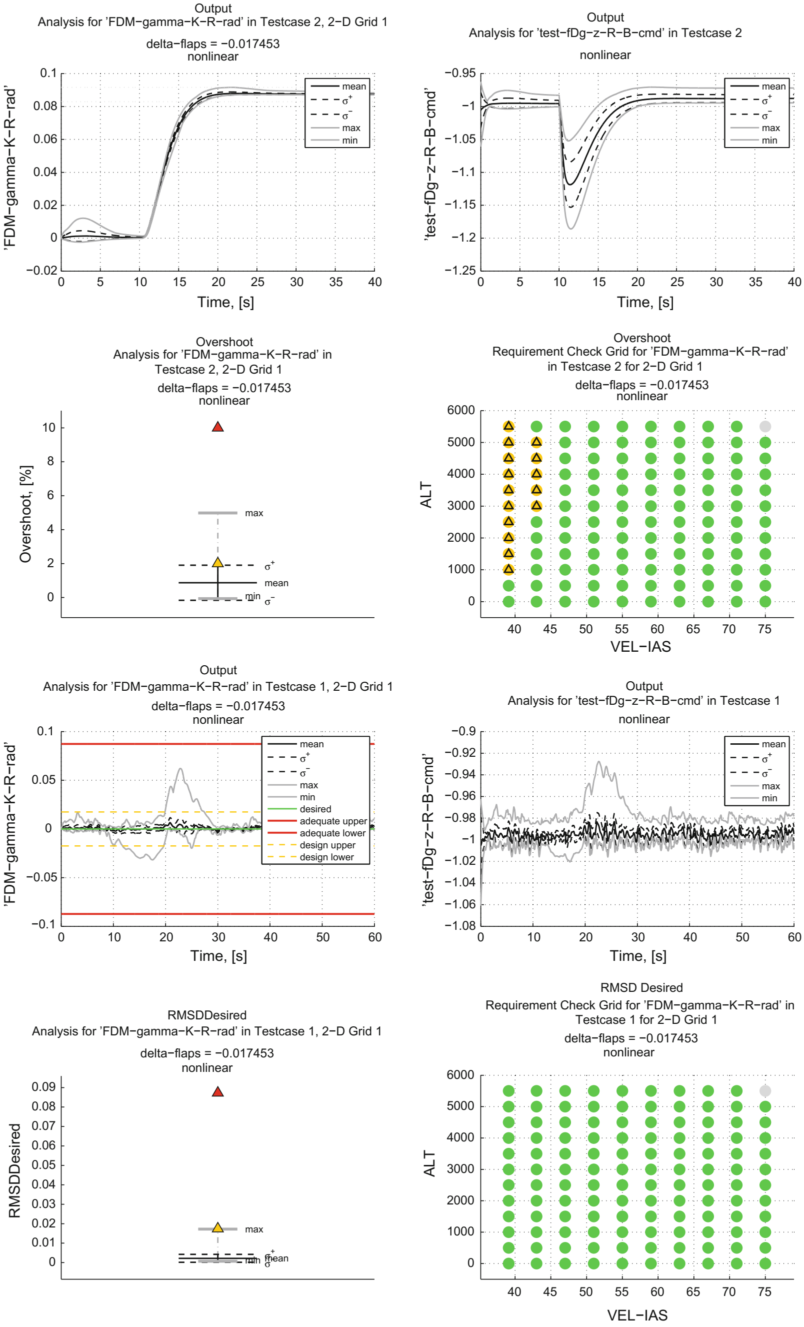


Fig. 4 Automated nonlinear assessment over the envelope with statistics and requirements checking

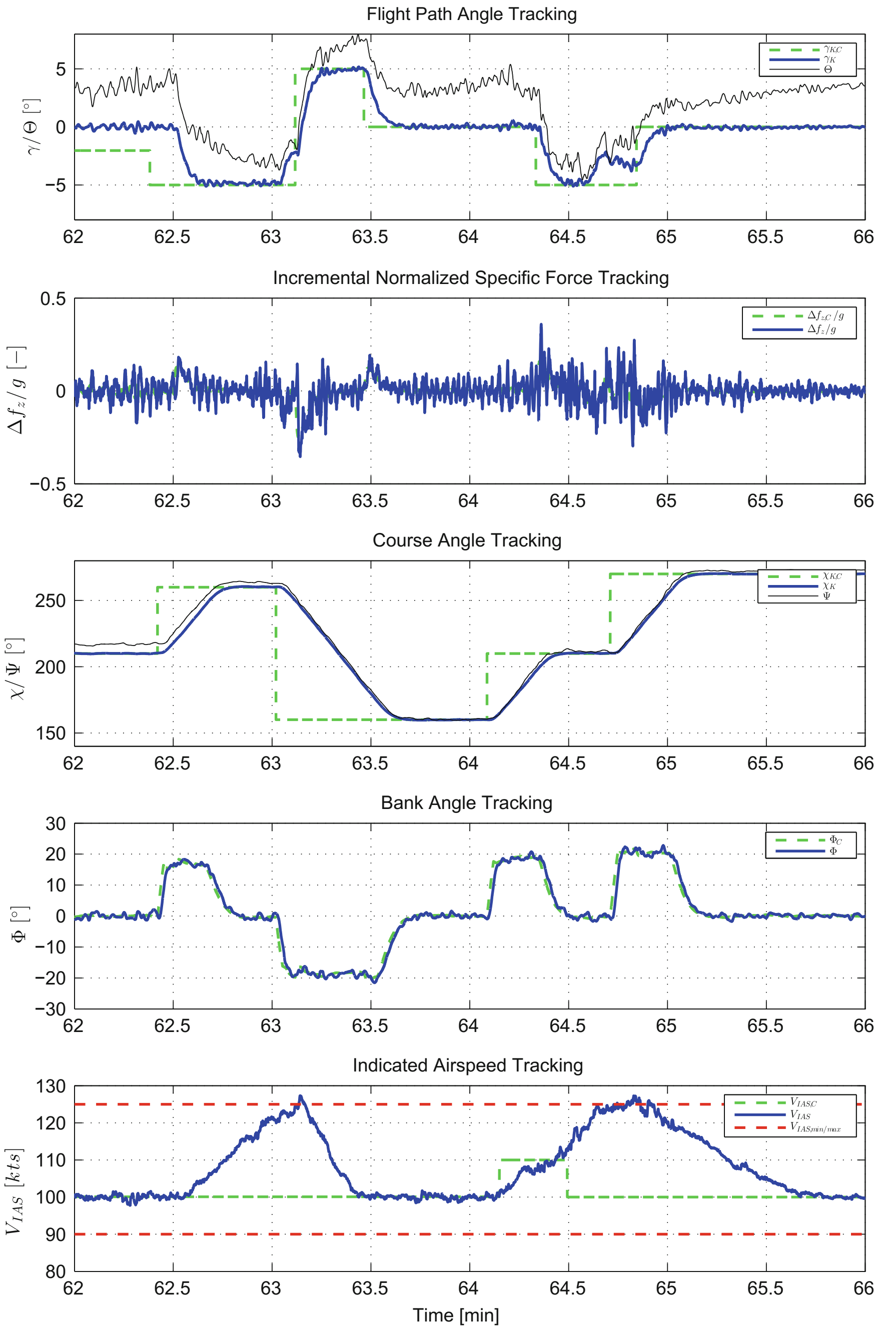


Fig. 5 Vertical and lateral flight path command tracking

factor and bank angle commands, during separate and concurrent maneuvers in the vertical and lateral plane, respectively. Flight path angle commands were smoothly acquired and maintained largely within 0.2°. Course angle commands were likewise smoothly acquired without overshoots and maintained within 0.5°. The indicated airspeed increases during descending maneuvers, as the thrust is saturated at its minimum, however returns to the commanded 100 kts when the aircraft levels off.

8 Conclusions and Outlook

The development of an automatic flight path controller, as part of a modular automatic flight guidance and control system, has been presented along with flight test results using a DA42 flying testbed. Verification activities included linear and nonlinear controller assessment, model-level simulations, hardware- and aircraft-in-the-loop testing and finally flight testing. The flight test results has demonstrated the feasibility of the control approach, showing good vertical and lateral path tracking and disturbance performance also in turbulent atmospheric conditions. Further flight testing will focus on extending the flight envelope and allow increased control bandwidth, in order to allow more aggressive path and trajectory control.

Acknowledgements Part of this research was supported by German Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

References

1. Johnson EN, Calise AJ (2003) Limited authority adaptive flight control for reusable launch vehicles. J Guid Control Dyn 26(6):906–913
2. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: The 14th international conference on control, automation, robotics, and vision
3. Karlsson E, Schatz SP, Baier T, Dörhöfer C, Gabrys A, Hochstrasser M, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Schneider V, Spiegel P, Steinert L, Zollitsch AW, Holzapfel F (2016) Automatic flight path control of an experimental DA42 general aviation aircraft. In: The 14th international conference on control, automation, robotics, and vision
4. Lambregts AA (1983) Integrated system design for flight and propulsion control using total energy principles. In: AIAA aircraft design, systems and technology meeting
5. Lambregts AA (1983) Vertical flight path and speed control autopilot design using total energy principles. In: AIAA guidance and control conference
6. Lambregts AA (2013) Flight envelope protection for automatic and augmented manual control. In: EuroGNC
7. Lambregts AA (2013) TECS generalized airplane control system design - an update. In: EuroGNC
8. SAE International (2007) AS94900: General specification for aerospace - flight control systems - design, installation and test of piloted military aircraft

9. Schatz SP, Holzapfel F (2014) Modular trajectory/path following controller using nonlinear error dynamics. In: 2014 IEEE international conference on aerospace electronics and remote sensing technology (ICARES). IEEE, pp 157–163
10. Schatz SP, Schneider V, Karlsson E, Holzapfel F, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Zollitsch AW (2016) Flightplan flight tests of an experimental DA42 general aviation aircraft. In: The 14th international conference on control, automation, robotics, and vision
11. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2015 IEEE international conference on aerospace electronics and remote sensing technology (ICARES). IEEE
12. Schneider V, Piprek P, Schatz SP, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Karlsson E, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Zollitsch AW, Holzapfel F (2016) Online trajectory generation using clothoid segments. In: The 14th international conference on control, automation, robotics, and vision
13. Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs

Active Control Objective Prioritization for High-Bandwidth Automatic Flight Path Control

Erik Karlsson, Thaddäus Baier, Christoph Dörhöfer, Agnes Gabrys, Markus Hochstrasser, Christoph Krause, Patrick J. Lauffs, Nils C. Mumm, Kajetan Nürnberger, Lars Peter, Simon P. Schatz, Volker Schneider, Philip Spiegel, Lukas Steinert, Alexander W. Zollitsch and Florian Holzapfel

1 Introduction

Classical automatic flight path control structures, with independent Single-Input, Single-Output (SISO) control of each flight path variable (i.e. airspeed, climb angle, track angle), suffer from inherent proneness to dangerous control objective conflicts, and energy mismanagement, as arbitrary combinations of speed and flight path are unreachable with saturated energy rate (i.e. thrust) control, and arbitrary curvatures of vertical and lateral flight path are precluded with saturated path perpendicular force control. The higher the bandwidth of the flight path control, the more dangerous the decoupled control, as the aircraft energy and its distribution quickly degrades. Especially for unmanned aircraft applications, robust, high-authority, and high-bandwidth flight path control is desired, in order to allow more aggressive maneuvering, going beyond traditional low-bandwidth autopilot operation and waypoint based trajectory following. Rudimentary flight envelope protections are sometimes incorporated to protect airspeed and prevent loss of control; however, usually as "last line of defense" and not for smooth and deterministic control objective prioritization during normal maneuvering. Integrated path and speed control approaches, such as Total Energy Control System (TECS), see e.g. [8, 9, 11] uses coupled energy flow rate and distribution control to decouple flight path and speed tracking, and to integrate airspeed-based envelope protections [10], but are less suitable for high-bandwidth maneuvering, due to the sacrifice of pitch control bandwidth in order to achieve the desired path/speed control decoupling.

E. Karlsson (✉) · T. Baier · C. Dörhöfer · A. Gabrys · M. Hochstrasser ·
C. Krause · P.J. Lauffs · N.C. Mumm · K. Nürnberger · L. Peter · S.P. Schatz ·
V. Schneider · P. Spiegel · L. Steinert · A.W. Zollitsch · F. Holzapfel
Institute of Flight System Dynamics, TU München, Boltzmannstr. 15,
85748 Garching, Germany
e-mail: erik.karlsson@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_8

In this paper, an active energy rate and force distribution prioritization, as integrated part of the flight path controller of a modular flight guidance and control system, is presented. The approach is based on, and an extension of previous suggestions made in [2–4], and initial simulation results of the energy rate prioritization presented in [6]. This contributions of this paper include: the integration of the binary energy rate and force prioritizations to allow mixed priorities; full integration of the functionalities into the flight path controller and validation using simulation of the full closed loop system; initial flight test results of the energy integrity protection function.

The integrated prioritizations allow speed or flight path angle maneuvering to be prioritized in case of saturated energy control, with automatic speed priority at the edges of the envelope in order to ensure the airspeed integrity of the aircraft, and vertical or lateral flight path curvature to be prioritized in case of saturated path transverse force control. The approach is validated using high-fidelity simulations of the full closed loop system, and partly validated in flight tests, pending full flight testing.

2 Controller Environment

2.1 Modular Application Architecture

Figure 1 shows a principal overview of the structure of the modular flight guidance and control system application, of which the flight path controller and its con-

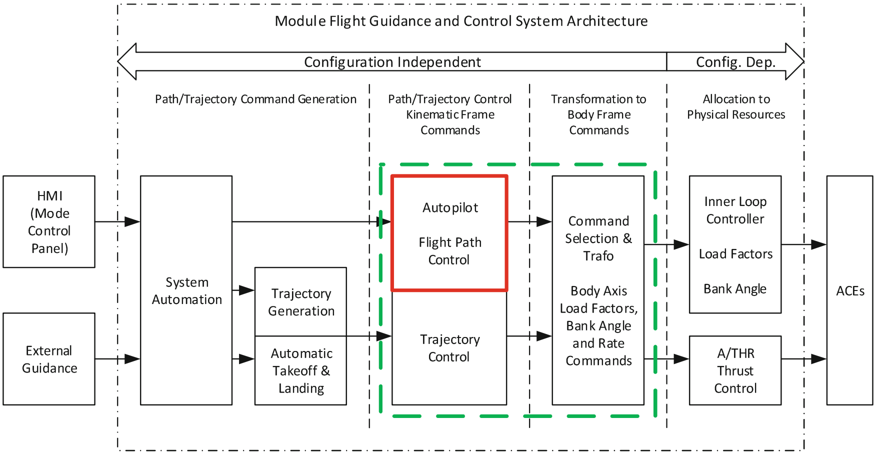

Module Flight Guidance and Control System Architecture
Configuration Independent
Config. Dep.
Path/Trajectory Command Generation
Path/Trajectory Control Kinematic Frame Commands
Transformation to Body Frame Commands
Allocation to Physical Resources
HMI (Mode Control Panel)
External Guidance
System Automation
Trajectory Generation
Automatic Takeoff & Landing
Autopilot
Flight Path Control
Trajectory Control
Command Selection & Trafo
Body Axis Load Factors, Bank Angle and Rate Commands
Inner Loop Controller
Load Factors
Bank Angle
A/THR Thrust Control
ACEs

Fig. 1 Modular flight guidance and control system architecture, with flight path controller highlighted

trol objective prioritization functionalities are integrated. The modular architecture allows for partial customization and the inclusion/exclusion of functional modules, depending on the operational concept for a given target platform and required level of automation. The functional modules include: a system automation module, interfacing with external command sources, and responsible the overall flight phase operating modes; flight plan trajectory generation [14, 15] and automatic takeoff and landing modules, calculating and generating the commands for the outer loops of the autoflight system, i.e. the trajectory [12, 13] and flight path control modules [7]; an outer loop command switching and transformation module, calculating the commands for the inner loop and autothrust controllers, computing the final commands to the Actuator Control Electronics (ACE).

2.2 The DA42 OE-FSD Demonstrator Platform

For validation and verification of experimental control algorithms, Fly-By-Wire (FBW) technologies and equipment for manned and unmanned applications, especially the aforementioned modular flight guidance and control system, the Institute of Flight System Dynamics has, in cooperation with the manufacturer, extensively modified a Diamond Aircraft Industries DA42 M-NG (Multi-Purpose Platform, Next Generation) aircraft, to serve as flying testbed. An experimental Fly-By-Wire (FBW) control system, with electro-mechanical actuators, is installed parallel to the mechanical controls. The experimental system actuates the control surfaces via both electrical and overload clutches, and a safety monitoring system ensures the safe and controlled engagement and disengagement of the experimental flight control system.

3 Flight Path Controller

This section briefly describes the flight path controller structure, into which the prioritization functionalities are integrated. For a more detailed description, see [7]. The command variables of the controller are the aircraft flight path states, hence, the control bandwidth is determined by the achievable path curvature, i.e. specific forces in the kinematic frame. The basic principle for the flight path control is a reference model based dynamic inversion [16] of the translational equations of motions (for a detailed derivation, see e.g. [1, 17]), given by

$$\mathbf{u}_C = \begin{bmatrix} \Delta f_{x,A,C} \\ f_{y,K,C} \\ \Delta f_{z,K,C} \end{bmatrix} = \begin{bmatrix} \nu_{\dot{V}_A} \\ V_K \cos\gamma_K \nu_{\dot{\chi}_K} \\ -V_K \nu_{\dot{\gamma}_K} \end{bmatrix}, \tag{1}$$

with $\nu_{\dot{V}_A}$, $\nu_{\dot{\chi}_K}$, and $\nu_{\dot{\gamma}_K}$ being the speed, vertical and lateral flight path loop pseudo controls, $\Delta f_{x,A,C}$ the commanded linear specific force in the aerodynamic frame, and $f_{y,K,C}$, and $\Delta f_{z,K,C}$ the commanded specific forces in the kinematic frame. Cross-coupled reference models for each path variable produce the desired tracking dynamics, see Sect. 3.1, PI error controllers ensure tracking of the reference states, and pseudo-control hedging [5] accounts for inner loop dynamics and plant response deficits. The controller structure is aircraft independent with a generic interface to aircraft-specific inner loop and thrust controllers for the control of the normal, lateral and linear specific forces, see Sect. 3.2. The advantage of the specific force based inner loops is good disturbance rejection, before disturbances manifest themselves into flight path, velocity or position errors.

3.1 Cross-Fed Reference Models for the Path Dynamics

The path controller includes linear second order aperiodic reference models for the airspeed, course/heading angle and flight path angle/vertical speed (Fig. 2) according to

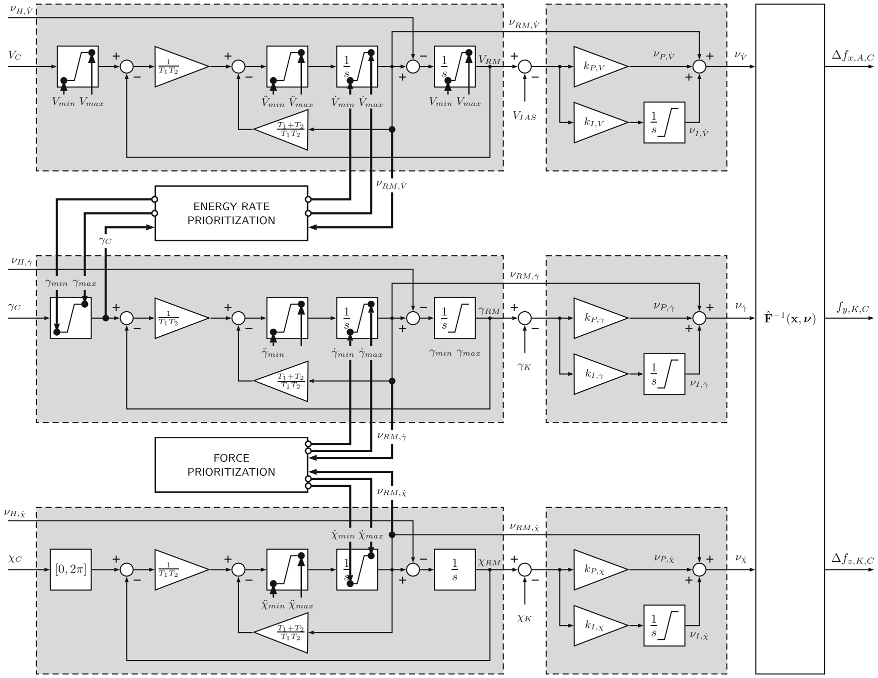


Fig. 2 Reference model based path control structure with energy rate and force prioritization cross-feeds

$$\dot{\nu}_{RM} = \frac{1}{T_1 T_2}\left(y_C - y_{RM}\right) - \frac{T_1 + T_2}{T_1 T_2}\nu_{RM}, \tag{2}$$

$$\dot{y}_{RM} = \nu_{RM} - \nu_H, \tag{3}$$

where T_1 and T_2 are the reference model time constants, y_C is the flight path command, y_{RM} and $\dot{y}_{RM}$ the reference model states, ν_{RM} the pseudo control to the inversion, and

$$\nu_H = \nu - \hat{\nu}, \tag{4}$$

a hedging signal added to adjust the reference model dynamic according to the plant response $\hat{\nu}$. The reference models are coupled to provide the desired energy rate and force prioritizations between the vertical, lateral and energy control planes at the edges of the aircraft speed and maneuver envelopes, see Sects. 3.3 and 3.4, to provide robustness against non-achievable flight path commands, and ensure smooth and achievable commands to the inner loops. The reference models include multiple limiters: limitation of the input command value; limitation of the reference state first derivative, i.e. linear acceleration, flight path angle rate (limit imposed by permissible specific normal force) and turn rate; limitation of the reference state second derivative, e.g. kinematic bank angle rate, and integrator output limitation, to account for the hedging signal, which is added after the rate limiter. These limiters are utilized by the prioritization functions to actively adjust reference states and their derivatives to fulfill the energy rate and force constraints.

3.2 Command Selection and Transformation

The flight path controller outputs specific force commands to the inner loop and thrust control loop, equivalent to the desired path curvatures and path acceleration. Detailed descriptions of these loops are beyond the scope of this paper.

Before entering the inner loop controller, the kinematic frame specific force commands are transformed to a bank angle command in the lateral plane and a normal body axis specific force command in the longitudinal plane. The longitudinal inner loop controller employs the normal body axis specific force and the pitch rate as feedback for a PI controller. The lateral inner loop controller is a MIMO control structure designed to track the bank angle command Φ_{cmd} and reduce the body lateral acceleration $f_{y,B}$ to zero. The thrust loop controls the linear specific force $\Delta f_{x,A}$ by adjusting the throttle position δ_T.

3.3 Energy Rate Distribution Prioritization

The weight specific total energy of an object (neglecting any rotational energy), i.e. its energy height h_E, is determined by its height and speed relative to some arbitrary reference (for an aircraft, e.g. the Earth's surface), i.e. its kinetic and potential energies,

$$h_E = \frac{E_{tot}}{mg} = \frac{E_{pot}}{mg} + \frac{E_{kin}}{mg} = h + \frac{V_K^2}{2g}. \tag{5}$$

A rate of change of the energy height, i.e. an energy flow rate or *specific excess power*

$$\dot{h}_E = \frac{V_K \dot{V}_K}{g} + \dot{h} = V_K \left(\frac{\dot{V}_K}{g} + \sin \gamma_K \right), \tag{6}$$

gives that an increased flow of energy to the aircraft may be used to increase the altitude or the speed, i.e. converted to a climb angle γ_K or acceleration $\dot{V}_K$. The achievable combination of acceleration and flight path angle is determined by the achievable specific excess power (energy flow rate) from the aircraft propulsion system (or active drag control). Analog to the energy height, which represents the achievable height for a given total energy, its rate of change can be expressed in terms of the achievable acceleration or flight path angle, if all specific excess power is converted accordingly,

$$\dot{h}_E = V_K \frac{\dot{V}_E}{g} = V_K \sin \gamma_E, \tag{7}$$

where $\dot{V}_E$ is the energy rate equivalent acceleration,

$$\dot{V}_E = \dot{V}_K + g \sin \gamma, \tag{8}$$

i.e. the achievable horizontal acceleration at zero flight path angle, and γ_E is the energy rate equivalent climb angle (or simply the energy angle),

$$\gamma_E = \arcsin \left(\sin \gamma + \frac{\dot{V}_K}{g} \right), \tag{9}$$

i.e. the achievable flight path angle at zero horizontal acceleration. The energy angle is often displayed on primary flight displays. From Eqs. (8) and (9), the potential flight path angle $\gamma_{\dot{V}_C}$ for a given desired acceleration $\dot{V}_C$ is given by

$$\gamma_{\dot{V}_C} = \arcsin \left(\frac{\dot{V}_E - \dot{V}_C}{g} \right) = \arcsin \left(\frac{\dot{V}_K - \dot{V}_C}{g} + \sin \gamma \right), \tag{10}$$

and the potential acceleration $\dot{V}_{\gamma_C}$ for a desired flight path angle γ_C by

$$\dot{V}_{\gamma_C} = g \cdot (\sin\gamma_E - \sin\gamma_C) = \dot{V}_K + g\,(\sin\gamma - \sin\gamma_C)\,. \tag{11}$$

In case of active energy flow rate control, i.e. engaged autothrust, the potential energy rate equivalent flight path angle or acceleration can be calculated including the estimated available maximum or minimum energy flow rate $\Delta\dot{E}_{max/min} = mV_K\Delta f_{x,max/min}$ automatically provided by the autothrust, with

$$\Delta f_{x,max/min} = \frac{\Delta f_x}{\Delta\delta_T}\left(\delta_{T,max/min} - \delta_{T,C}\right), \tag{12}$$

according to

$$\gamma_{\dot{V}_C,pot} = \arcsin\left(\frac{\dot{V}_K - \dot{V}_C}{g} + \sin\gamma + \frac{\Delta f_{x,max/min}}{g}\right), \tag{13}$$

and

$$\dot{V}_{\gamma_C,pot} = \dot{V}_K + g\left(\sin\gamma - \sin\gamma_C + +\Delta f_{x,max/min}\right). \tag{14}$$

3.3.1 Speed Verses Vertical Path Prioritization

The Eqs. (13) and (14) may now be used to set the limiters in the corresponding reference model, in order to prioritize either the commanded flight path angle (by limiting the pseudo control acceleration in the speed reference model) or the commanded speed (by limiting the reference flight path angle). For speed priority, the constraints on flight path angle are given by

$$\gamma_{max,vprio} = \arcsin\left(\sin\gamma_K - \frac{\nu_{RM,\dot{V}} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,max}}{g}\right), \tag{15}$$

$$\gamma_{min,vprio} = \arcsin\left(\sin\gamma_K - \frac{\nu_{RM,\dot{V}} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,min}}{g}\right), \tag{16}$$

and for path priority, the constraints on the acceleration are given by

$$\nu_{RM,\dot{V},max,pprio} = \hat{\nu}_{\dot{V}} - g\,(\sin\gamma_C - \sin\gamma_K) + \Delta f_{x,max}, \tag{17}$$

$$\nu_{RM,\dot{V},min,pprio} = \hat{\nu}_{\dot{V}} - g\,(\sin\gamma_C - \sin\gamma_K) + \Delta f_{x,min}. \tag{18}$$

3.3.2 Mixed Energy Rate Authority Prioritization

The previous section presented a binary speed versus flight path angle prioritization, where the energy rate distribution always is performed in favor of one of the two variables. In the following, an integration of the two modes of prioritization is presented, where an available energy rate distribution authority, based on the permissible maximum and minimum acceleration limits, is budgeted between acceleration and flight path angle. The specific speed-normalized energy rate distribution authority, a measure of how fast energy may be traded between acceleration and flight path, is given by

$$\frac{\dot{E}_{dist,max}}{m \cdot V_K} = \left(\dot{V}_{max,nom} - \dot{V}_{min,nom}\right), \tag{19}$$

i.e. the energy rate distribution between flight path and acceleration is limited so that the permissible nominal acceleration limits are not violated. A energy rate distribution factor $\kappa_E \in [0, 1]$ is introduced, defining the amount of energy rate distribution authority allocated to flight path and acceleration control, where $\kappa_E = 0$ indicates full flight path priority, and $\kappa_E = 1$ full speed priority, and a value in between a mixed priority. The flight path angle and acceleration limits are then a function of κ_E, according to

$$\gamma_{max,prio} = \arcsin\left(\sin\gamma_K - \frac{\left((1-\kappa_E)\,\dot{V}_{min,nom} + \kappa_E \nu_{RM,\dot{V}}\right) - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,max}}{g}\right), \tag{20}$$

$$\gamma_{min,prio} = \arcsin\left(\sin\gamma_K - \frac{\left((1-\kappa_E)\,\dot{V}_{max,nom} + \kappa_E \nu_{RM,\dot{V}}\right) - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,min}}{g}\right), \tag{21}$$

and

$$\nu_{RM,\dot{V},max,prio} = \hat{\nu}_{\dot{V}} + g\left(\sin\gamma_K - \left((1-\kappa_E)\sin\gamma_C - \kappa_E \frac{\dot{V}_{max,nom}}{g}\right)\right) + \Delta f_{x,max}, \tag{22}$$

$$\nu_{RM,\dot{V},min,prio} = \hat{\nu}_{\dot{V}} + g\left(\sin\gamma_K - \left((1-\kappa_E)\sin\gamma_C - \kappa_E \frac{\dot{V}_{min,nom}}{g}\right)\right) + \Delta f_{x,min}. \tag{23}$$

If the energy rate distribution factor κ_E equals 0, i.e. full path priority, Eqs. (20) and (21) reduce to

$$\gamma_{max,prio} = \arcsin\left(\sin\gamma_K - \frac{\dot{V}_{min,nom} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,max}}{g}\right), \tag{24}$$

$$\gamma_{min,prio} = \arcsin\left(\sin\gamma_K - \frac{\dot{V}_{max,nom} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,min}}{g}\right), \tag{25}$$

i.e. the flight path angles are limited to ensure an energy redistribution within the maximum and minimum nominal acceleration limits. Equations (22) and (23) are reduced to Eqs. (17) and (18), respectively, ensuring that the desired flight path angle may be tracked. If the energy rate distribution factor κ_E equals 1, i.e. full speed priority, Eqs. (20) and (21) reduce to Eqs. (15) and (16), respectively, ensuring that the desired pseudo-control acceleration may be tracked.

3.3.3 Energy Integrity Protection

In order to ensure the energy integrity of the aircraft, the energy distribution must be automatically prioritized in favor of the airspeed at the edges of its envelope. Therefore, the acceleration limits $\nu_{RM,\dot{V},max/min}$ for the speed reference model are determined as a function of the distance to the airspeed limit $V_{IAS,max/min}$ and the current estimated acceleration $\hat{\nu}_{\dot{V}}$, and the flight path angle limits $\gamma_{max,min}$ adjusted so that the energy distribution ensures the limit acceleration is maintained, thereby preventing an over- or undershoot of the limit airspeed.

Two phase plane regions are defined; one defined by the distance ΔV_{trans} to the limit speed, where the energy distribution is actively controlled in favor of airspeed (transition region), and one, defined by the distance ΔV_{prot} beyond the speed limit (protection region) where the throttle limits are additionally adjusted beyond nominal limits to prevent a low energy state (from e.g. Maximum Continuous to Take-Off/Go-Around Thrust) according to

$$\dot{V}_{max} = k_{V,max}\left(V_{IAS,max} - V_{IAS}\right) \tag{26}$$

$$\begin{aligned}\dot{V}_{trans,max} &= k_{V,max}\left(V_{IAS,trans,max} - V_{IAS}\right)\\ &= k_{V,max}\left(V_{IAS,max} - \Delta V_{trans} - V_{IAS}\right)\end{aligned} \tag{27}$$

$$\begin{aligned}\dot{V}_{prot,max} &= k_{V,max}\left(V_{IAS,prot,max} - V_{IAS}\right)\\ &= k_{V,max}\left(V_{IAS,max} + \Delta V_{prot} - V_{IAS}\right)\end{aligned} \tag{28}$$

$$\dot{V}_{min} = k_{V,min}\left(V_{IAS,min} - V_{IAS}\right) \tag{29}$$

$$\begin{aligned}\dot{V}_{trans,min} &= k_{V,min}\left(V_{IAS,trans,min} - V_{IAS}\right)\\ &= k_{V,min}\left(V_{IAS,min} + \Delta V_{trans} - V_{IAS}\right)\end{aligned} \tag{30}$$

$$\begin{aligned}\dot{V}_{prot,min} &= k_{V,min}\left(V_{IAS,prot,min} - V_{IAS}\right)\\ &= k_{V,min}\left(V_{IAS,min} - \Delta V_{prot} - V_{IAS}\right)\end{aligned} \tag{31}$$

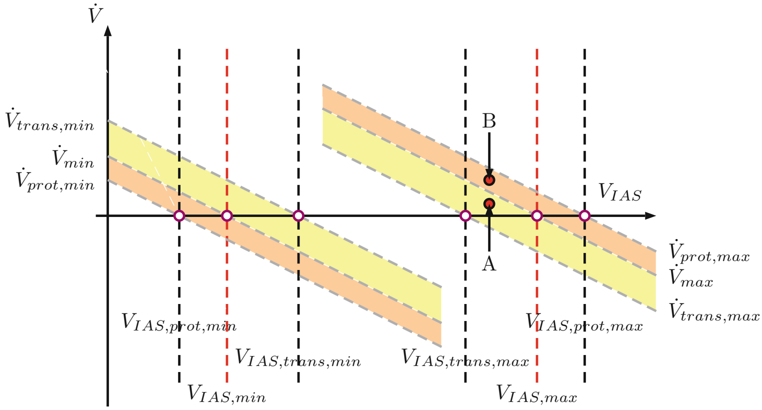


Fig. 3 Speed-acceleration phase plane regions for energy integrity protection

The phase plane regions with their corresponding limit accelerations are visualized in Fig. 3, where the point A represents a situation where the aircraft approaches its upper speed limit (positive acceleration), and has entered the transition region where the acceleration and flight path limits are adjusted to prevent an overshoot. In case the acceleration is even larger, point B, the throttle limits are further adjusted (in this case towards their absolute minimum) in order to further utilize the possible energy flow rate.

Within the transition region, the acceleration limits are linearly reduced from their nominal limits to the protection limits according to

$$\nu_{RM,\dot{V},max} = \dot{V}_{max,nom}\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{max}}{\dot{V}_{trans,max} - \dot{V}_{max}}\right) \tag{32}$$

$$\nu_{RM,\dot{V},min} = \dot{V}_{min,nom}\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{min}}{\dot{V}_{trans,min} - \dot{V}_{min}}\right) \tag{33}$$

with $\nu_{RM,\dot{V},max/min}$ being the acceleration limit set in the speed reference model, $\dot{V}_{max/min,nom}$ the nominal limits, $\hat{\nu}_{\dot{V}}$ the estimated linear acceleration, and $\dot{V}_{trans,min}$ and $\dot{V}_{min}$ the transition region limit accelerations according to Eqs. (26)–(31). The acceleration limits are visualized in Fig. 4. The protection flight path angles equivalent to the minimum/maximum acceleration in the low/high speed transition regions can be derived from Eq. (10), again with the available thrust increment considered in case of autothrust active,

$$\gamma_{max,prot} = \arcsin\left(\sin\gamma_K - \frac{\nu_{RM,\dot{V},min} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,max}}{g}\right), \tag{34}$$

$$\gamma_{min,prot} = \arcsin\left(\sin\gamma_K - \frac{\nu_{RM,\dot{V},max} - \hat{\nu}_{\dot{V}}}{g} + \frac{\Delta f_{x,min}}{g}\right). \tag{35}$$

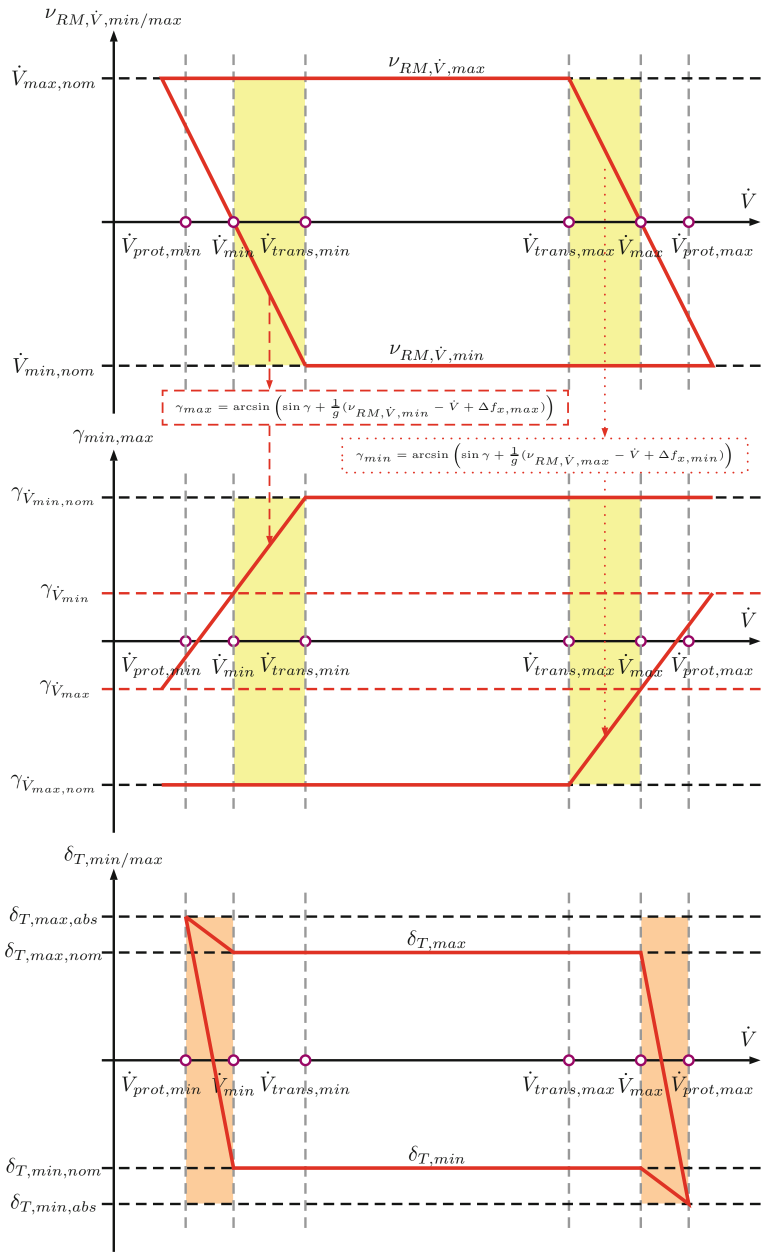


Fig. 4 Acceleration, flight path and throttle limits for energy integrity protection

The flight path angle limits and their relation to the acceleration limits are visualized in Fig. 4. Analogously, the throttle limits $\delta_{T,min/max}$ are adjusted within the high/low speed protection region, from the nominal limits $\delta_{T,nom,min/max}$ to their absolute limits $\delta_{T,abs,min/max}$ in order to provide maximum or minimum available thrust, see Fig. 4. In the low speed protection region, the maximum and minimum throttle limits are given by

$$\delta_{T,max,prot,LS} = \delta_{T,max,abs} + \left(\delta_{T,max,nom} - \delta_{T,max,abs}\right)\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{prot,min}}{\dot{V}_{min} - \dot{V}_{prot,min}}\right), \tag{36}$$

$$\delta_{T,max,prot,LS} = \delta_{T,max,abs} + \left(\delta_{T,min,nom} - \delta_{T,max,abs}\right)\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{prot,min}}{\dot{V}_{min} - \dot{V}_{prot,min}}\right), \tag{37}$$

and in the high speed protection region analogously by

$$\delta_{T,max,prot,HS} = \delta_{T,min,abs} + \left(\delta_{T,max,nom} - \delta_{T,min,abs}\right)\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{prot,max}}{\dot{V}_{max} - \dot{V}_{prot,max}}\right), \tag{38}$$

$$\delta_{T,max,prot,HS} = \delta_{T,min,abs} + \left(\delta_{T,min,nom} - \delta_{T,min,abs}\right)\left(\frac{\hat{\nu}_{\dot{V}} - \dot{V}_{prot,max}}{\dot{V}_{max} - \dot{V}_{prot,max}}\right). \tag{39}$$

It is possible to extend the throttle limit adjustment within the protection region to include means of drag control, e.g. speed brakes, to further control the energy flow rate in the case of high speed protection.

3.4 Path Curvature Force Distribution Prioritization

A curvature of the vertical or lateral flight path is achieved by changing the magnitude of the total force perpendicular to the flight path in the vertical and lateral plane, respectively. For most conventional aircraft configurations the primary force acting perpendicular to the flight path is the lift force. (Other means include thrust vectoring and side force control.) For high-bandwidth concurrent maneuvering in the vertical and lateral plane, the permissible transverse force has to be taken into account and, in case of saturation, a prioritization or trade-off between desired curvature in the vertical and lateral plane activated. The force distribution is realized by cross-coupling the flight path angle and track angle reference models according to Fig. 2, and setting the flight path angle rate and track angle rate pseudo-control command limiters, respectively.

The flight path transverse specific forces in the kinematic frame are given by

$$f_{y,K} = V_K \cos\gamma_K \dot{\chi}, \tag{40}$$

$$f_{z,K} = -V_K \dot{\gamma} - g\cos\gamma_K, \tag{41}$$

i.e. the magnitude of the commanded transverse forces, $f_{z,\overline{K},C}$ for achieving the desired vertical and lateral path curvatures, respectively, must be equal to or less than the maximum permissible magnitude of the transverse specific force $|f_{z,\overline{K}}|_{max}$,

$$|f_{z,\overline{K},C}| = \sqrt{\left(f_{z,K,C}\right)^2 + \left(f_{y,K,C}\right)^2} \leq |f_{z,\overline{K}}|_{max} \tag{42}$$

with

$$f_{y,K,C} = V_K \cos\gamma_K \cdot \nu_{\dot{\chi}}, \tag{43}$$
$$f_{z,K,C} = -V_K \cdot \nu_{\dot{\gamma}} - g\cos\gamma_K. \tag{44}$$

The maximum magnitude of the transverse specific force is determined by either the maximum achievable lift coefficient, or the maximum permissible structural load, or some desired maneuver bandwidth criteria.

3.4.1 Vertical Verses Lateral Plane Maneuvering Prioritization

From Eqs. (42)–(44), constraints on the desired flight path curvatures $\nu_{RM,\dot{\gamma}}$ and $\nu_{RM,\dot{\chi}}$ may be derived, in order to achieve the desired curvature in the prioritized plane. For prioritized maneuvering in the vertical plane, the constraints on the lateral path curvature commands are given by

$$\begin{aligned} \nu_{RM,\dot{\chi},max,vert} &= \frac{1}{V_K\cos\gamma_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(f_{z,K,C}\right)^2}, && (45)\\ &= \frac{1}{V_K\cos\gamma_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(-V_K\cdot\nu_{RM,\dot{\gamma}} - g\cos\gamma_K\right)^2}, \\ \nu_{RM,\dot{\chi},min,vert} &= -\frac{1}{V_K\cos\gamma_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(f_{z,K,C}\right)^2}, && (46)\\ &= -\frac{1}{V_K\cos\gamma_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(-V_K\cdot\nu_{RM,\dot{\gamma}} - g\cos\gamma_K\right)^2}. \end{aligned}$$

In order for the expression under the square root to be positive, the following constraints apply to the commanded vertical path curvature,

$$\nu_{RM,\dot{\gamma},max,vert} = \frac{1}{V_K}\left(|f_{z,\overline{K}}|_{max} - g\cos\gamma_K\right), \tag{47}$$
$$\nu_{RM,\dot{\gamma},min,vert} = \frac{1}{V_K}\left(-|f_{z,\overline{K}}|_{max} - g\cos\gamma_K\right). \tag{48}$$

Analogously, maneuvering in the lateral plane may be prioritized, either by allowing the full transverse force to produce a lateral path curvature (i.e. with bank angle equal to 90 degrees), or by allowing the residual force when maintaining the current flight path angle for maneuvering in the lateral plane, thereby inhibiting further curvature of the vertical flight path. The constraints on the vertical path curvature are given by

$$
\begin{aligned}
\nu_{RM,\dot{\gamma},max,lat} &= \frac{1}{V_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(f_{y,K,C}\right)^2} - \frac{g}{V_K}\cos\gamma_K && (49)\\
&= \frac{1}{V_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(V_K\cos\gamma_K \cdot \nu_{RM,\dot{\chi}}\right)^2} - \frac{g}{V_K}\cos\gamma_K \\
\nu_{RM,\dot{\gamma},min,lat} &= -\frac{1}{V_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(f_{y,K,C}\right)^2} - \frac{g}{V_K}\cos\gamma_K && (50)\\
&= -\frac{1}{V_K}\sqrt{|f_{z,\overline{K}}|^2_{max} - \left(V_K\cos\gamma_K \cdot \nu_{RM,\dot{\chi}}\right)^2} - \frac{g}{V_K}\cos\gamma_K
\end{aligned}
$$

and the constraints on the lateral curvature, when utilizing full transverse force,

$$
\begin{aligned}
\nu_{RM,\dot{\chi},max,lat,full} &= \frac{1}{V_K\cos\gamma_K}|f_{z,\overline{K}}|_{max}, && (51)\\
\nu_{RM,\dot{\chi},min,lat,full} &= -\frac{1}{V_K\cos\gamma_K}|f_{z,\overline{K}}|_{max}, && (52)
\end{aligned}
$$

and when utilizing transverse force limited for lateral maneuvering, leaving a residual transverse force for maintaining current climb angle,

$$
\begin{aligned}
\nu_{RM,\dot{\chi},max,lat,man} &= \frac{g}{V_K}\sqrt{\left(\frac{|f_{z,\overline{K}}|_{max}}{g\cos\gamma_K}\right)^2 - 1}, && (53)\\
\nu_{RM,\dot{\chi},min,lat,man} &= -\frac{g}{V_K}\sqrt{\left(\frac{|f_{z,\overline{K}}|_{max}}{g\cos\gamma_K}\right)^2 - 1}. && (54)
\end{aligned}
$$

3.4.2 Mixed Force Authority Prioritization

The previous section presented a binary vertical versus lateral flight path curvature prioritization, where the transverse specific force distribution always is performed in favor of one of the two planes. In the following, an integration of the two path curvature modes of prioritization is presented, where the available transverse specific force, is budgeted between vertical and lateral flight path curvature. Analog to the energy-based prioritizations, a force distribution factor $\kappa_F \in [0, 1]$ is introduced, defining the amount of transverse force authority is allocated to vertical and lateral flight path curvature control, where $\kappa_F = 0$ indicates full vertical flight path curvature priority, and $\kappa_F = 1$ full lateral path curvature maneuver priority, and a value in between a mixed priority. The permissible flight path curvature limits are then a function of κ_F, according to

$$\nu_{RM,\dot{\gamma},max,prio} = \frac{1}{V_K}\sqrt{f^2_{z,\overline{K},min} - \left(\kappa_F V_K \cos\gamma_K \cdot \nu_{RM,\dot{\chi}}\right)^2} - \frac{g}{V_K}\cos\gamma_K, \tag{55}$$

$$\nu_{RM,\dot{\gamma},min,prio} = \frac{1}{V_K}\left(-f_{z,\overline{K},max} - g\cos\gamma_K\right), \tag{56}$$

and

$$\nu_{RM,\dot{\chi},max,prio} = \frac{1}{V_K\cos\gamma_K}\sqrt{f^2_{z,\overline{K},min} - \left((1-\kappa_F)V_K \cdot \nu_{RM,\dot{\chi}} + g\cos\gamma_K\right)^2}, \tag{57}$$

$$\nu_{RM,\dot{\chi},min,prio} = -\frac{1}{V_K\cos\gamma_K}\sqrt{f^2_{z,\overline{K},min} - \left((1-\kappa_F)V_K \cdot \nu_{RM,\dot{\chi}} + g\cos\gamma_K\right)^2}. \tag{58}$$

4 Simulation Results of Energy Rate and Force Prioritizations

Closed loop simulations of the flight path control system (including inner loop and autothrust controllers) where performed using a high-fidelity 6 degree-of-freedom simulation model of the DA42 M-NG, including sensor and actuator models. Figure 5, left column, shows the airspeed and flight path command tracking in full path priority mode, with autothrust engaged. Here, the acceleration limits are continuously adjusted in order to distribute the energy flow rate in favor of the desired flight path angle, and allow for the flight path angle command to be tracked (ensuring vertical flight path maneuvering integrity). Figure 5, right column, shows the airspeed and flight path command tracking in full speed priority mode, with autothrust engaged. The flight path angle limits are continuously adjusted in order to distribute the energy flow rate in favor of the desired acceleration, and allow for the speed command to be tracked (ensuring speed maneuvering integrity). Figure 6, left column, show concurrent maneuvering in the vertical and lateral planes with vertical plane curvature priority. The lateral curvature, and thus the bank angle, is limited in order for the vertical flight path commands to be tracked. Figure 6, right column, shows the same maneuvers with full lateral plane curvature priority. Here, the vertical path curvature is not achieved until lateral plane maneuvering is finished and thus desaturating the normal specific force control, limited at $-1.5\,\mathrm{g}$ (z_B-axis pointing downwards).

Figure 7 shows the energy integrity protection and automatic speed prioritization at the edges of the airspeed envelope, where the path priority is overridden and the flight path angle limits are adjusted based on the minimum/maximum acceleration limits to ensure energy integrity, and no airspeed over- or undershoot.

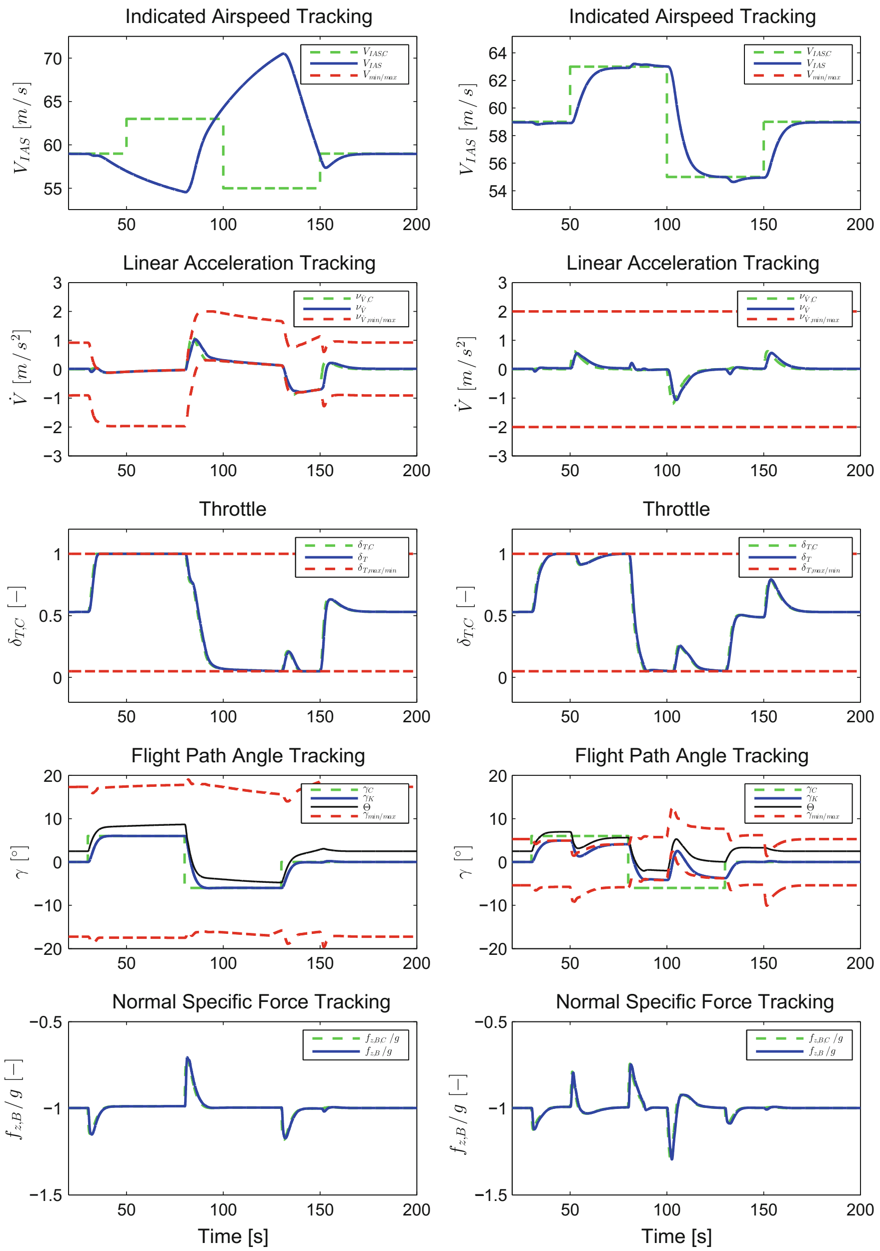


Fig. 5 Energy rate prioritization, path priority (*left*) and speed priority (*right*)

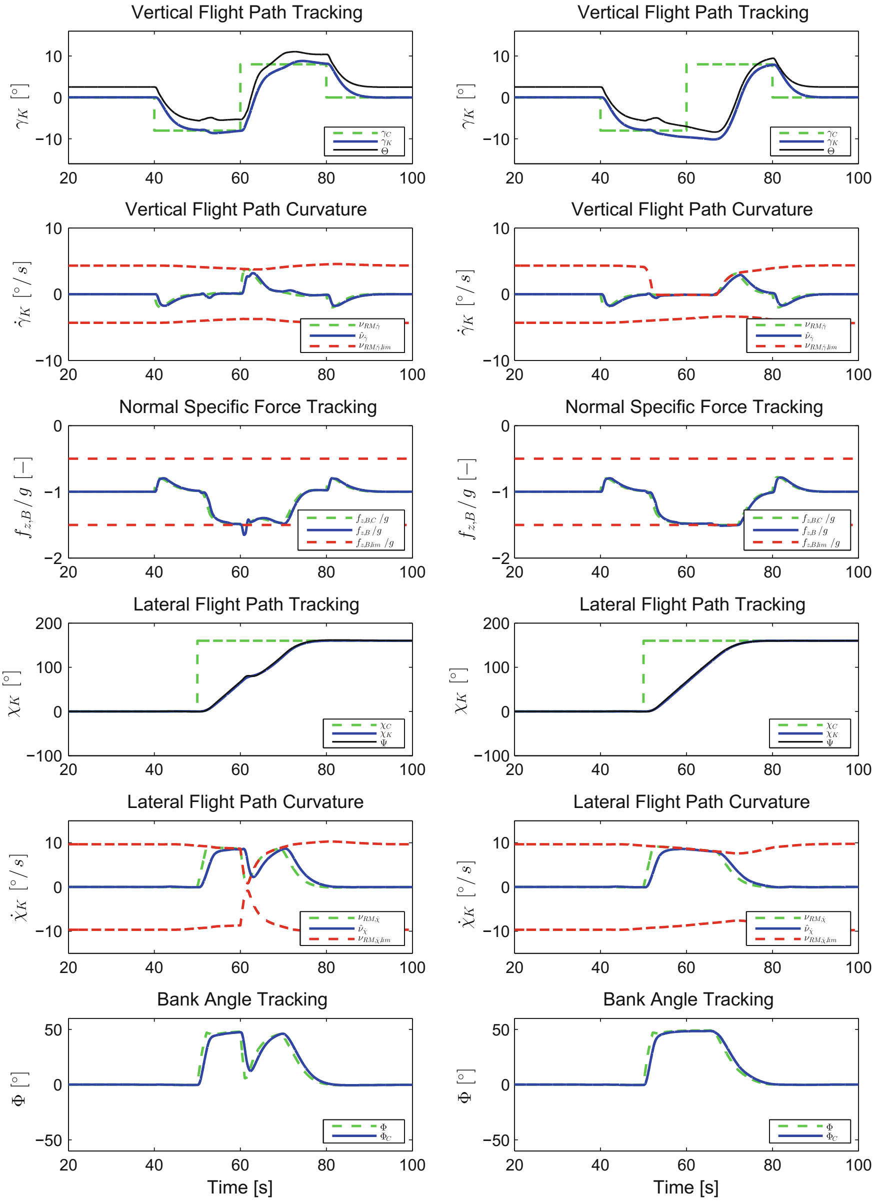


Fig. 6 Force prioritization, vertical path curvature priority (*left*) and lateral path curvature priority (*right*)

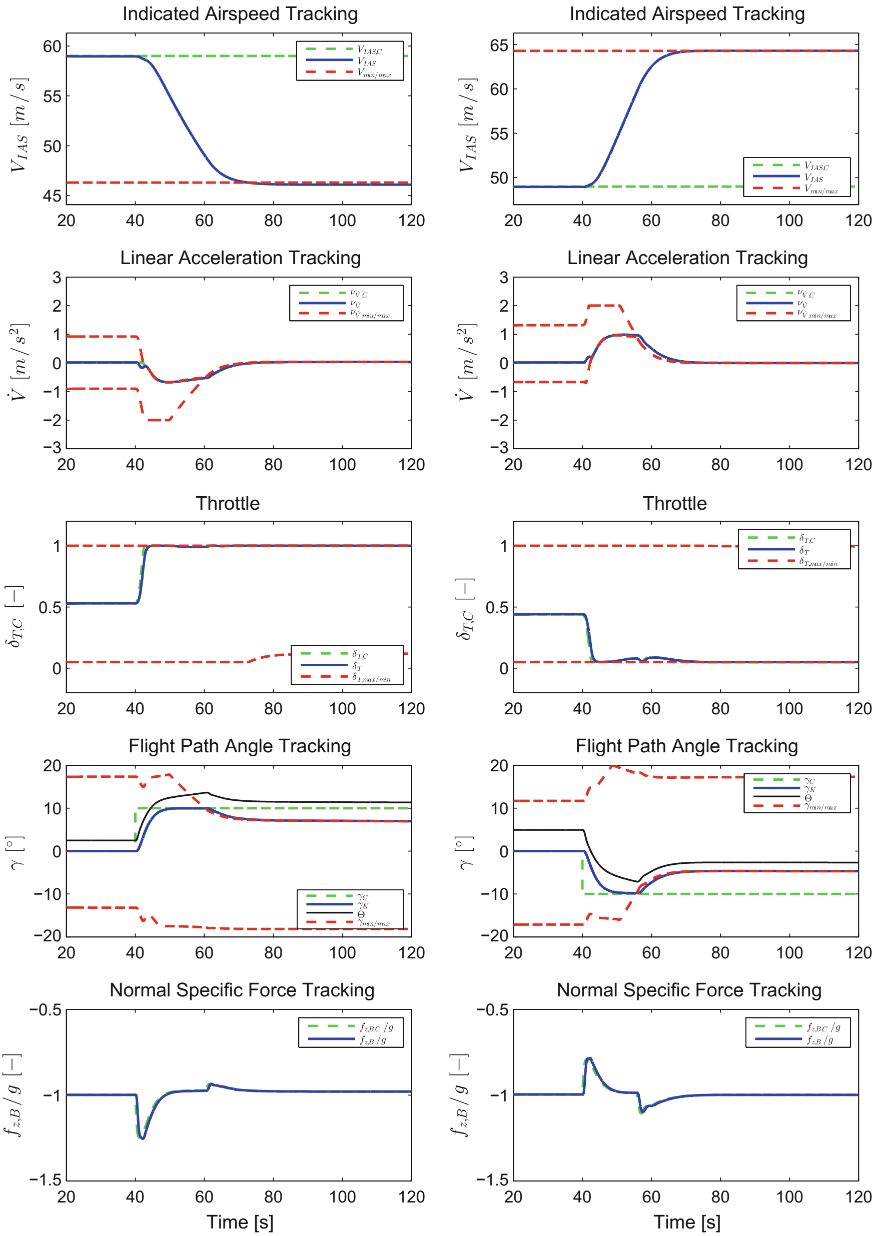


Fig. 7 Energy integrity protection, low speed protection (*left*) and high speed protection (*right*)

5 Flight Test Results of Energy Integrity Protection

The nominal function of the flight path controller has been successfully validated in flight test during 2016, for a limited flight envelope, see [7]. Figure 8 shows initial results from flight testing of the energy integrity functionality performed in August 2016, in light turbulent conditions. For initial evaluation, the airspeed envelope was reduced to 90–125 kts, at safe distance from the aircraft envelope. As seen in Fig. 8, the flight path angle is adjusted in order for the aircraft to stay in the permissible airspeed envelope. Due to restricted flight testing time available, and suboptimal weather conditions, the level of correspondence with simulation results was hard to

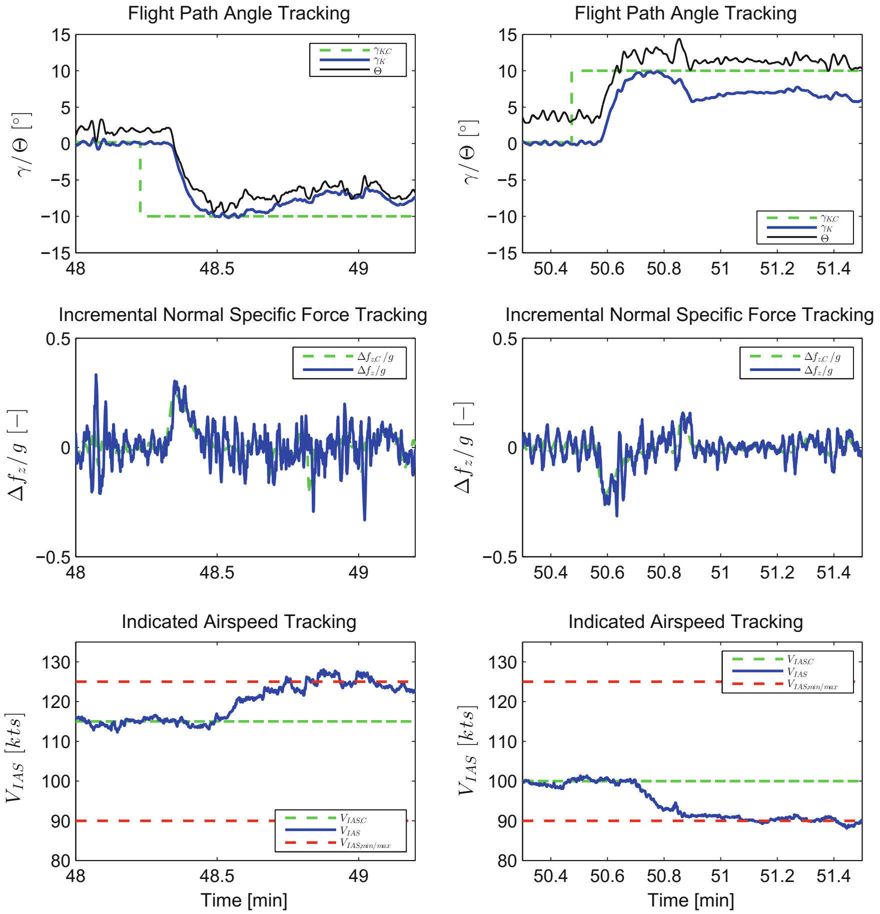


Fig. 8 Energy integrity protection, flight test results, low speed protection (*left*) and high speed protection (*right*)

establish, although correct functionality verified. Earlier flight testing of the nominal flight path controller functions has shown good correspondence with simulation results.

6 Conclusions and Outlook

An approach for active energy rate and force distribution prioritization, for deterministic maneuvering integrity and energy protection, has been integrated into the flight path controller of a modular flight guidance and control system, and has been validated using a high-fidelity simulation of the closed loop system for various conditions and for the current permissible bandwidth of the system, as well as partially validated in flight tests. The results show that the implemented approach functions as intended, for the current permissible bandwidth. Next steps will comprise further validation of the approach in simulation and in flight tests, with increasing of control bandwidth. The nominal flight path controller has previously been verified in flight tests, with good correspondence to simulation results.

Acknowledgements Part of this research was supported by German Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

References

1. Etkin B, Reid LD (1996) Dynamics of flight: stability and control, 3rd edn. Wiley, New York
2. Holzapfel F, Höcht L, Schuck F, Myschik S, Sachs G (2008) Nonlinear flight-path control - a flight dynamics perspective. Jarhrbuch 2007 der deutschen gesellschaft für luft- und raumfahrt 6:3627–3640
3. Holzapfel F, Schuck F, Höcht L, Kurth F, Sachs G (2007) Non-linear high bandwidth control of UAVs for autonomous mission capability. In: 45th AIAA aerospace sciences meeting and exhibition
4. Holzapfel F, Schuck F, Höcht L, Sachs G (2007) Flight dynamics aspects of path control. In: AIAA guidance, navigation and control conference and exhibition
5. Johnson EN, Calise AJ (2003) Limited authority adaptive flight control for reusable launch vehicles. J Guid Control Dyn 26(6):906–913
6. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: The 14th international conference on control, automation, robotics, and vision
7. Karlsson E, Schatz SP, Baier T, Dörhöfer C, Gabrys A, Hochstrasser M, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Schneider V, Spiegel P, Steinert L, Zollitsch AW, Holzapfel F (2016) Automatic flight path control of an experimental DA42 general aviation aircraft. In: The 14th international conference on control, automation, robotics, and vision
8. Lambregts AA (1983) Integrated system design for flight and propulsion control using total energy principles. In: AIAA aircraft design, systems and technology meeting
9. Lambregts AA (1983) Vertical flight path and speed control autopilot design using total energy principles. In: AIAA guidance and control conference
10. Lambregts AA (2013) Flight envelope protection for automatic and augmented manual control. In: EuroGNC

11. Lambregts AA (2013) TECS generalized airplane control system design - an update. In: EuroGNC
12. Schatz SP, Holzapfel F (2014) Modular trajectory/path following controller using nonlinear error dynamics. In: 2014 IEEE international aerospace electronics and remote sensing technology (ICARES). IEEE, pp. 157–163
13. Schatz SP, Schneider V, Karlsson E, Holzapfel F, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Zollitsch AW (2016) Flightplan flight tests of an experimental DA42 general aviation aircraft. In: The 14th international conference on control, automation, robotics, and vision
14. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2015 IEEE international aerospace electronics and remote sensing technology (ICARES). IEEE
15. Schneider V, Piprek P, Schatz SP, Baier T, Dörhöfer C, Hochstrasser M, Gabrys A, Karlsson E, Krause C, Lauffs PJ, Mumm NC, Nürnberger K, Peter L, Spiegel P, Steinert L, Zollitsch AW, Holzapfel F (2016) Online trajectory generation using clothoid segments. In: The 14th international conference on control, automation, robotics, and vision
16. Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs
17. Stevens BL, Lewis FL (2003) Aircraft control and simulation, 2nd edn. Wiley, Hoboken

Nonlinear Modular 3D Trajectory Control of a General Aviation Aircraft

Simon P. Schatz and Florian Holzapfel

1 Introduction

In the past decades, the use of fly-by-wire (FBW) became popular as a concept for flight controls. Nowadays, the FBW concept is becoming standard for large transportation aircraft, but research projects of Part-23 general aviation aircraft are also directed towards FBW concepts. In some cases, full FBW is of interest, in some cases the mechanical flight control system is altered by interconnected actuators of (sometimes) lower authority. Especially the emergence of unmanned aerial vehicles (UAV) and electrical flying further led to multiple applications of FBW systems, which incorporate multicopters of small MTOW as well as platforms of several tons.

Both manned and unmanned sensing platforms require waypoint based navigation to complete their missions and as a consequence, trajectory generation and control plays a significant role for these systems. In this paper, the trajectory controller of a fully integrated auto-flight control system is considered. At the institute, a modular control structure was developed in the context of adopting various components of the guidance and control structure to different platforms without the need of redesigning the controllers for the specific applications. Consequently, a range of modules were designed, which range from classical autopilot concepts to trajectory generation and automatic landing as well as to automatic aerodynamic trimming. All of these components may vary for different applications such as UAVs with unstable aircraft dynamics or classical autopilots for manned flight, but the rework can be decreased significantly due to the modular structure.

S.P. Schatz (✉) · F. Holzapfel
Institute of Flight System Dynamics, Technische Universität München,
Boltzmannstr. 15, 85748 Garching Bei München, Germany
e-mail: simon.p.schatz@tum.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_9

The trajectory control module is designed as a controller to drive any deviations in the lateral and vertical plane to zero and hence, is a candidate to be utilized by multiple higher-level functions such as waypoint navigation, automatic takeoff and landing, or altitude hold modes. From the controller's point of view, the deviations and their derivatives are used as control errors whereas other information about the trajectory such as trajectory angles and their rates are further utilized to improve the controller's performance, especially in cases of dynamical trajectories.

Previous work conducted in the field of trajectory control usually rely on multiple loops to address the problem of trajectory tracking. Commonly, there are two outer loops, which first compute a track / flight path angle command due to deviations and then compute inner loop commands to control the angular error. In this field, work has been conducted ranging from linear controllers to nonlinear methods such as backstepping or nonlinear dynamic inversion. Examples of such methods can be found in [1–10]. In this paper, a second-order approach for the outer loop is presented, which only consists of one control loop using both the position deviations in the lateral and vertical plane as well as their derivatives as control errors. This reduces the required time scale separation as there are just two second order loops used in the modular structure: trajectory controller and inner loop controller. The trajectory controller's design is based on the method of nonlinear dynamic inversion (NDI, see e.g. [11, 12]), which inherently leads to utilizing the feedforward terms contained in the dynamical trajectory, along with a PD controller.

In this paper, previous work [13, 14] is extended, specifically the second order error dynamics of the lateral and vertical position deviations, which are the basis for the derivation of the controller, are addressed in detail and the dynamic inversion applied to these error dynamics is derived for the 3D controller. Additionally, the command transformation to the inner loop controller is derived and further information is given regarding the trajectory control and the environment. Ultimately, specific simulation results are presented to show the controller's lateral tracking performance in comparison to a linear controller and in presence of significant wind as additional test data to the first principal flight test demonstration published in Ref. [14].

The remainder of this paper is arranged as follows: Section 1.1 states the nomenclature used in this paper. The environment of the trajectory controller is presented in Sect. 2, where the trajectory generation (Sect. 2.1), the command transformation (Sect. 2.2), and the inner loop controller (Sect. 2.3) are discussed in more detail. Section 3 deals with the trajectory control design, specifically the discussion of the kinematic equations of motion, the derivation of the nonlinear error dynamics, its dynamic inversion, and the control design. In Sect. 4 simulation results are given to show the controller's performance in presence of wind and also by comparison to a linear PD controller without feedforward terms. Finally, concluding remarks are given in Sect. 5.

1.1 Nomenclature

In this paper, mainly vectors, which are noted bold, and scalars, noted in italic letters, are used. Furthermore, $\overrightarrow{(\cdot)}$ indicates that a vector is an element of the Euclidean space $\mathbb{R}^3$. In addition, Euclidean vectors considered in this paper are the following:

- $\left(\vec{\mathbf{r}}^{\,P}\right)_B$ denotes a position vector at the point P given in the body frame.
- $\left(\vec{\mathbf{V}}_K^{\,P}\right)^E$ denotes a kinematic velocity vector at the point P differentiated with respect to the E frame.
- $\left(\vec{\boldsymbol{\omega}}^{\,OK}\right)$ denotes an angular rate of the K frame with respect to the O frame.

All vector variables used in this paper can be compounded based on the concept introduced by these three examples.

2 Control Environment

The Institute of Flight System Dynamics owns a Diamond Aircraft Industries DA42 NG aircraft, which is adapted to serve as flying testbed. For this purpose, the aircraft has been modified to be controlled using a fly-by-wire system, which gears into the existing mechanical flight control system utilizing both an electrical and an overload clutch. The testbed is equipped with a safety system, which enables the safety pilot to either open the clutches by pushing a disengage button or by application of a sufficiently large stick force.

Figure 1 (taken from [13, 14] for clarity) shows the overall modular structure of the integrated autoflight system. The modules of the integrated auto-flight system range from the input monitoring and system automation part [15], where the incoming signals of sensors and ground station are evaluated and the overall operating modes are computed based on the result, to the actuator control electronics (ACEs) and the engine control unit (ECU).

The modular concept was chosen to keep certain parts of identical structure throughout different platforms. For this, certain interfaces are maintained throughout various projects. As an example, the control structure of the autopilot persists over several platforms and just requires adaptation of certain parameters such as gains and limits to the dynamics of the aircraft. The inner loop module, however, may differ significantly over different applications as it depends fully on aircraft dynamics and control allocation.

The trajectory controller is a part of the auto-flight-system, which in this context denotes the autopilot functionalities such as heading hold or flight level change mode and the speed controller / auto-thrust. The auto-flight-system also incorporates energy protection and prioritization, limiting the speed and the flight path angle such that no adverse conditions occur. In context of vertical trajectory control or upon application of the altitude hold mode, along with deviations also the permissible flight path angles are forwarded to the trajectory controller, which are then used to additionally

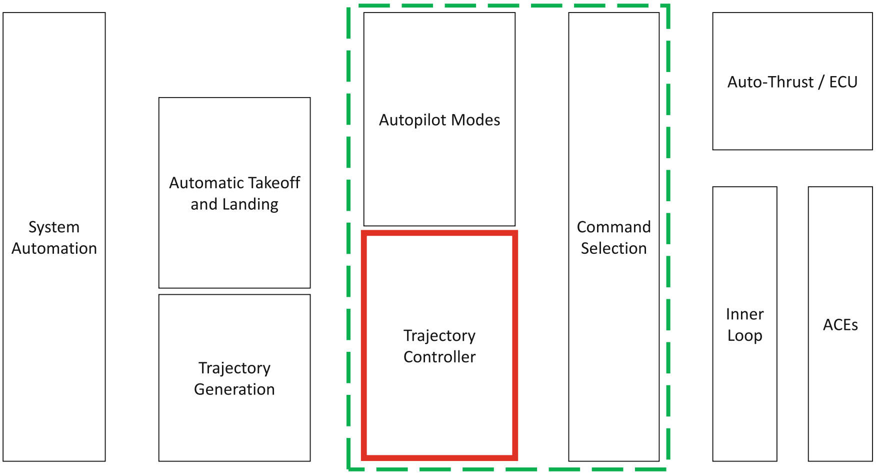


Fig. 1 Integrated autoflight system

limit the trajectory controller's specific force commands. For more details on the auto-flight-system refer to [16, 17].

The commands of the auto-flight-system and the trajectory controller, which are further dedicated as outer loop, are selected and forwarded to the inner loop controller. While the outer loops provide normalized specific force commands in the kinematic frame with respect to unaccelerated flight, these commands are transformed by a configuration-specific module, which is further depicted in Sect. 2.2, to the inputs required by the corresponding inner loop, which then computes the final commands to the ACEs and ECUs. The inner loop of the DA42 is briefly discussed in Sect. 2.3.

The trajectory controller deals with any deviations as long as they comply with the interface described in Sect. 2.1. In the current implementation for the DA42, trajectories and the corresponding deviations may either be provided by the automatic takeoff and landing module or a trajectory generation module [18, 19], which is based on waypoints. The latter is briefly discussed in Sect. 2.1. Furthermore, the trajectory controller is used to drive the altitude deviation to zero when the altitude hold mode is applied.

2.1 Trajectory Generation

The paper deals with a modular trajectory controller, which is designed to drive the deviations in a trajectory frame to zero. From the controller's point of view, any source of deviations and additional trajectory information is permitted. In this paper, fly-by trajectories generated by the trajectory generation module are considered. Here, the interface to this module is briefly introduced.

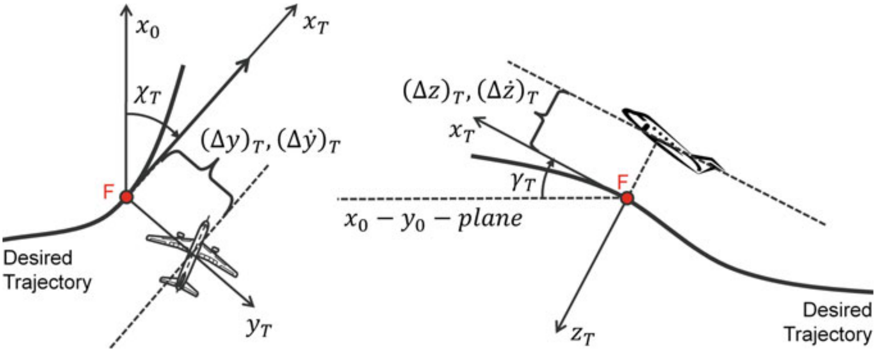


Fig. 2 Trajectory frame with deviations and their corresponding time derivatives

For this purpose, the trajectory frame T, which is related to the kinematic frame K as commonly used in aerospace applications, is introduced as follows:

- The x-axis is aligned with and in the direction of the desired velocity at the footpoint on the trajectory.
- The z-axis points downwards parallel to the projection of the local normal to the WGS84 ellipsoid into a plane perpendicular to the x-axis.
- The y-axis points parallel to the earth's surface and yields a right hand system with the previous two axes.

This frame is further illustrated in Fig. 2 (taken from [13, 14]), where the deviations $(\Delta y)_T$, $(\Delta z)_T \in \mathbb{R}$ and its time derivatives $(\Delta \dot{y})_T$, $(\Delta \dot{z})_T \in \mathbb{R}$ are also displayed.

The trajectory controller is driven by the following inputs:

- The deviations between aircraft and desired trajectory

$$\left(\vec{\mathbf{r}}^{FR}\right)_T = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_T \triangleq \begin{pmatrix} x^R - x^F \\ y^R - y^F \\ z^R - z^F \end{pmatrix}_T , \tag{1}$$

- the corresponding time derivatives of these deviations

$$\left(\vec{\mathbf{V}}_K^{FR}\right)_T^T = \begin{pmatrix} \Delta \dot{x} \\ \Delta \dot{y} \\ \Delta \dot{z} \end{pmatrix}_T^T \triangleq \begin{pmatrix} \dot{x}^R - \dot{x}^F \\ \dot{y}^R - \dot{y}^F \\ \dot{z}^R - \dot{z}^F \end{pmatrix}_T^T , \tag{2}$$

- the desired trajectory angles χ_T, γ_T,
- the desired trajectory angular rates $\dot{\chi}_T$, $\dot{\gamma}_T$,
- the desired trajectory angular accelerations $\ddot{\chi}_T$, $\ddot{\gamma}_T$,
- the desired kinematic acceleration at the trajectory footpoint $\dot{V}_K^F$.

Note that $\vec{\mathbf{r}}^{FR}$, $\vec{\mathbf{V}}_K^{FR} \in \mathbb{R}^3$ and all the other components are real valued.

The current state of the trajectory generation module, previously presented in [18, 19], which generates a trajectory online based on 3+1 given waypoints, employs waypoints of the type fly-by, fly-over, and loiter, where just fly-by trajectories and a loiter are considered in this paper. Note that the trajectory is only generated for three-dimensional trajectories not employing the x-position on the trajectory, i.e. the time component, as the airspeed is controlled independently by the auto-flight system to maintain the aerodynamic integrity of the system.

A fly-by trajectory is generated using the current "from" waypoint, the fly-by waypoint, and the next waypoint after the fly-by to generate a circle passing the fly-by waypoint. Note that the radii of the circle of the fly-by maneuvers and loiters are computed based on the current kinematic velocity, a wind estimation as well as the current and commanded airspeeds. The maximum expected kinematic velocity throughout the turn is used to calculate the radius by aiming for a turn rate of $5°/s$. In order to smoothly start the curve, the trajectory generation is currently based on clothoids [19–22] for fly-by maneuvers, which provide a linear devolution of the trajectory angular rate $\dot{\chi}_T$. As a consequence, however, the angular acceleration $\ddot{\chi}_T$ employs non-continuous characteristics and hence, the angular accelerations remain unused throughout this paper.

2.2 Outer - Inner Loop Command Interface

As mentioned earlier, the outer loops provide normalized specific force commands in the kinematic frame with respect to unaccelerated flight, i.e. curvature commands given by

$$\vec{\mathbf{u}}_{ol,nrm,K} = \begin{bmatrix} \frac{\Delta f_{x,cmd,K}}{g} & \frac{f_{y,cmd,K}}{g} & \frac{\Delta f_{z,cmd,K}}{g} \end{bmatrix}^{\mathrm{T}}, \tag{3}$$

where g represents the acceleration due to gravity. In the following, the curvature commands are first transformed to the north-east-down (NED, O) frame, then converted to absolute normalized specific forces including gravity, and ultimately rotated about the heading Ψ as well as the pitch angle Θ, yielding

$$\vec{\mathbf{u}}_{ol,nrm,B'} = \mathbf{T}_{B'O}\left(\Psi, \Theta\right) \left(\mathbf{T}_{OK}\left(-\gamma_K, -\chi_K\right) \vec{\mathbf{u}}_{ol,nrm,K} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}_O \right), \tag{4}$$

where $\mathbf{T} \in \mathbb{R}_i^{3\times 3}$ depict transformation matrices, γ_K is the kinematic flight path angle, and χ_K denotes the kinematic track / course angle. Finally, the command transformation to the body frame can be obtained by

$$\vec{\mathbf{u}}_{ol,nrm,B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix} \vec{\mathbf{u}}_{ol,nrm,B'}, \tag{5}$$

where Φ depicts the roll angle of the aircraft. Now, as no lateral acceleration in the body frame is desired for a coordinated turn, the second row of (5) is utilized to compute

$$0 = \frac{f_{y,cmd,B'}}{g}\cos(\Phi_{cmd}) + \frac{f_{z,,B'}}{g}\sin(\Phi_{cmd}), \tag{6}$$

ultimately yielding the roll angle command

$$\Phi_{cmd} = \operatorname{atan}\left(-\frac{f_{y,cmd,B'}}{f_{z,cmd,B'}}\right). \tag{7}$$

Taking into account the second and third row of (5) and utilizing Φ_{curr}, we have

$$f_{y,cmd,B'} = -f_{z,cmd,B'}\tan(\Phi_{curr}) \tag{8}$$

$$\frac{f_{z,cmd,B}}{g} = -\frac{f_{y,cmd,B'}}{g}\sin(\Phi_{curr}) + \frac{f_{z,cmd,B'}}{g}\cos(\Phi_{curr}), \tag{9}$$

and the normal body load factor command in the longitudinal plane as commanded to the inner loop can be computed as

$$\frac{f_{z,cmd,B}}{g} = \frac{f_{z,cmd,B'}}{g\cos(\Phi_{curr})}, \tag{10}$$

where Φ_{curr} is currently chosen as the measurement of the roll angle, but can be altered by a small feedforward term based on the roll rate and the computational rate. The command transformation as given by Eqs. (8) and (10) is designed in this form to cope with the different time scales regarding longitudinal and lateral motion. Therefore, the given longitudinal command is computed using the current measurement of the roll angle as it takes more time to built up the angle rather than the longitudinal acceleration to maintain the appropriate curvature in both axes.

2.3 Inner Loop

As mentioned in Sect. 2.2, the inner loop comprises a normal body load factor command system in the longitudinal plane and a roll angle command system in the lateral plane. In case of the longitudinal plane, the normalized normal acceleration in the body frame and the pitch rate q are used for a PI controller. The lateral inner loop comprises feedback of roll and yaw rates, p and r, respectively, along with the normalized body lateral acceleration and the roll angle for a MIMO control architecture.

Since the control design of the inner loop is based on linear approaches such as LQR [23] for the longitudinal plane and eigenstructure assignment [24] for the lateral plane, gain scheduling is required to compose the controller over the whole envelope. In order to increase the validity of the linear controller and incorporate turn compensation, the feedback signals are treated to achieve smaller errors as inputs to the inner loop controller. As a consequence, the feedback signals as seen from the inner loop are given by

$$\frac{\delta f_{z,B}}{g} = \frac{f_{z,B}}{g} + \frac{\cos(\Theta)}{\cos(\Phi)}, \tag{11}$$

$$\delta p = p - \frac{g}{V_K^R}\tan(\Phi), \tag{12}$$

$$\delta q = q - \frac{g}{V_K^R}\tan(\Phi)\cos(\Theta)\sin(\Phi), \tag{13}$$

$$\delta r = r - \frac{g}{V_K^R}\sin(\Phi)\cos(\Theta), \tag{14}$$

where V_K^R denotes the absolute kinematic velocity of the reference point and $\tan(\mu_K)$ is approximated by $\tan(\Phi)$ for turn compensation as the kinematic bank angle μ_K is not available for measurement. In order to remain in the same magnitude, also the normal body load factor command is treated by means of

$$\frac{\delta f_{z,cmd,B}}{g} = \frac{f_{z,cmd,B}}{g} + \frac{\cos(\Theta)}{\cos(\Phi)}. \tag{15}$$

The current gains are chosen in such a way that the inherent dynamics of the DA42 are not changed significantly. In the lateral plane, the roll dynamics and the natural frequency of the dutch roll are maintained while the yaw damping is increased. Furthermore, the spiral pole is set to -1 for the whole envelope. In the longitudinal plane, the pitch damping is increased and the normal body load factor command system is implemented. The tracking performance is achieved by the feedforward part, while the feedback is designed primarily for disturbance rejection.

3 Trajectory Controller

In this section, the proposed trajectory controller is presented. Section 3.1 gives a short derivation of the kinematic equations of motion for the reference point of a rigid body aircraft. The error dynamics used for the trajectory controller are derived in Sect. 3.2. The nonlinear dynamic inversion controller, which is applied based on these error dynamics, is presented and analyzed in Sect. 3.3.

3.1 Kinematic Equations of Motion

This section deals with the kinematic equations of motion used for the path dynamics of an aircraft. For this purpose, the results known from e.g. [25, 26] are computed based on Newton's Second Axiom. For the usage addressed in this paper, the reference point R of a rigid body aircraft is considered.

Assumption 1 The aircraft is assumed to be a rigid body. As a consequence, the relative position between points on the aircraft does not change over time, resulting in

$$\left(\vec{\mathbf{V}}_K^{RP}\right)^B = 0. \tag{16}$$

For computing the equations of motion, consider the position vector $\vec{\mathbf{r}}^{\,P} \in \mathbb{R}^3$

$$\left(\vec{\mathbf{r}}^{\,P}\right) = \left(\vec{\mathbf{r}}^{\,R}\right) + \left(\vec{\mathbf{r}}^{\,RP}\right), \tag{17}$$

and the corresponding velocities in the inertial frame given by

$$\left(\vec{\mathbf{V}}_K^{R}\right)^I = \left(\vec{\mathbf{V}}_K^{R}\right)^E + \left(\vec{\boldsymbol{\omega}}^{IE}\right) \times \left(\vec{\mathbf{r}}^{\,R}\right), \tag{18}$$

$$\left(\vec{\mathbf{V}}_K^{RP}\right)^I = \left(\vec{\boldsymbol{\omega}}^{IB}\right) \times \left(\vec{\mathbf{r}}^{\,RP}\right), \tag{19}$$

which are valid according to Assumption 1. In addition, the following assumption is made, which is fairly standard [25]:

Assumption 2 The influence of the mass flow onto the impulse of the aircraft can be neglected:

$$\dot{m}\left(\vec{\mathbf{V}}_K^{P}\right)^E \approx 0. \tag{20}$$

Differentiating the position vector (17) twice and incorporating the velocities (18), (19), and the Assumptions 1 and 2, Newton's Second Axiom can be given as

$$\sum\left(\vec{\mathbf{F}}^{R}\right) = m\left(\dot{\vec{\mathbf{V}}}_K^{R}\right)^{EO} + m\,\vec{\mathbf{a}}_{add}. \tag{21}$$

Remark 1 Note that $\vec{\mathbf{a}}_{add}$ is a term containing all additional accelerations such as the Coriolis term, the acceleration due to the transport rate, and the acceleration due to difference of reference point R and center of gravity G. When considering a flat, non-rotating earth and choosing the center of gravity as reference point, $\vec{\mathbf{a}}_{add} = \vec{\mathbf{0}}$. For reasons of brevity, $\vec{\mathbf{a}}_{add}$ is neglected in the subsequent sections.

Making use of the transformation into the kinematic frame

$$\left(\dot{\vec{\mathbf{V}}}_{K}^{R}\right)^{EO} = \left(\dot{\vec{\mathbf{V}}}_{K}^{R}\right)^{EK} + \left(\vec{\boldsymbol{\omega}}^{OK}\right) \times \left(\vec{\mathbf{V}}_{K}^{R}\right)^{E}, \tag{22}$$

the kinematic equations of motion can be computed as

$$\dot{V}_K^R = \Delta f_{x,K}, \tag{23}$$

$$\dot{\chi}_K = \frac{f_{y,K}}{V_K^R \cos\gamma_{\mathrm{K}}}, \tag{24}$$

$$\dot{\gamma}_K = \frac{-\Delta f_{z,K}}{V_K^R}, \tag{25}$$

where the specific forces are defined as

$$\Delta f_{x,K} \triangleq \frac{(X_A^R + X_P^R)_K}{m} - g\sin\gamma_{\mathrm{K}}, \tag{26}$$

$$f_{y,K} \triangleq \frac{(Y_A^R + Y_P^R)_K}{m}, \tag{27}$$

$$\Delta f_{z,K} \triangleq \frac{(Z_A^R + Z_P^R)_K}{m} + g\cos\gamma_{\mathrm{K}}, \tag{28}$$

where $m \in \mathbb{R}$ denotes the mass of the aircraft and $(X_A^R)_K$, $(Y_A^R)_K$, $(Z_A^R)_K \in \mathbb{R}$ as well as $(X_P^R)_K$, $(Y_P^R)_K$, $(Z_P^R)_K \in \mathbb{R}$ are the components of the aerodynamic and propulsion forces at the reference point, respectively, noted in the kinematic frame.

For the further derivation of the trajectory controller, the aircraft dynamics related to the trajectory are given by the kinematic Eqs. (23)–(25). Note that the forces affecting these dynamics require an inner loop controller supplying the respective actuator commands for these loads. In order to achieve this, the commands in the kinematic frame are transformed into roll angle and normal body load factor commands as discussed in Sect. 2.2, which are then utilized by the inner loop controller briefly introduced in Sect. 2.3, whereas the airspeed is controlled by a speed controller as introduced in [16, 17].

3.2 Nonlinear Error Dynamics

In this section, the nonlinear error dynamics between the reference point of the aircraft and a reference trajectory are computed. The relative deviation is defined in (1) and its time derivative is given by Eq. (2). This relative velocity can be further computed as

$$\left(\vec{\mathbf{V}}_K^{FR}\right)_T^T = \left(\vec{\mathbf{V}}_K^{FR}\right)_T^E + \left(\vec{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\vec{\mathbf{r}}^{FR}\right)_T , \tag{29}$$

where the angular rate is specified as

$$\left(\vec{\boldsymbol{\omega}}^{TO}\right)_T = \begin{pmatrix} \dot{\chi}_T \sin\gamma_T \\ -\dot{\gamma}_T \\ -\dot{\chi}_T \cos\gamma_T \end{pmatrix} . \tag{30}$$

As mentioned earlier, the controller is designed to use the specific forces as inputs to the inner loop. For this purpose, the error dynamics are differentiated once again with respect to time, yielding

$$\left(\dot{\vec{\mathbf{V}}}_K^{FR}\right)_T^{TT} = \left(\dot{\vec{\mathbf{V}}}_K^{FR}\right)_T^{ET} + \left(\dot{\vec{\boldsymbol{\omega}}}^{TO}\right)_T^T \times \left(\vec{\mathbf{r}}^{FR}\right)_T + \left(\vec{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\vec{\mathbf{V}}_K^{FR}\right)_T^T , \tag{31}$$

which can be further derived using

$$\left(\dot{\vec{\mathbf{V}}}_K^{FR}\right)_T^{ET} = \left(\dot{\vec{\mathbf{V}}}_K^{R}\right)_T^{EK} + \left(\vec{\boldsymbol{\omega}}^{TK}\right)_T \times \left(\vec{\mathbf{V}}_K^{R}\right)_T^E - \left(\dot{\vec{\mathbf{V}}}_{K,des}^{F}\right)_T^{ET} . \tag{32}$$

Employing

$$\left(\vec{\boldsymbol{\omega}}^{TK}\right)_T = \left(\vec{\boldsymbol{\omega}}^{TO}\right)_T + \left(\vec{\boldsymbol{\omega}}^{OK}\right)_T , \tag{33}$$

we arrive at

$$\begin{aligned} \left(\dot{\vec{\mathbf{V}}}_K^{FR}\right)_T^{TT} = & \left(\dot{\vec{\mathbf{V}}}_K^{R}\right)_T^{EK} + \left(\vec{\boldsymbol{\omega}}^{OK}\right)_T \times \left(\vec{\mathbf{V}}_K^{R}\right)_T^E - \left(\dot{\vec{\mathbf{V}}}_{K,des}^{F}\right)_T^{ET} \\ & + \left(\dot{\vec{\boldsymbol{\omega}}}^{TO}\right)_T^T \times \left(\vec{\mathbf{r}}^{FR}\right)_T + \left(\vec{\boldsymbol{\omega}}^{TO}\right)_T \times \left(\left(\vec{\mathbf{V}}_K^{FR}\right)_T^T + \left(\vec{\mathbf{V}}_K^{R}\right)_T^E\right) , \end{aligned} \tag{34}$$

where the terms on the top row of (34) contain the kinematic aircraft dynamics (22) with the control inputs $\Delta f_{x,K}$, $f_{y,K}$, $\Delta f_{z,K}$, and the other terms are related to the inputs to the trajectory controller specified in Sect. 2.1.

3.3 Nonlinear Dynamic Inversion Controller

The error dynamics (34) are now transformed using nonlinear dynamic inversion [11, 12]. For this purpose, (34) is written as

$$\ddot{\vec{\mathbf{y}}}_{full} = \mathbf{G}_{full}\left(\gamma_K, \gamma_T, \Delta\chi\right)\vec{\mathbf{u}}_{full} + \vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right), \tag{35}$$

where $\Delta\chi \triangleq \chi_K - \chi_T$ and

$$\kappa \triangleq \left(\left(\vec{\mathbf{r}}^{FR}\right)_T, \left(\vec{\mathbf{V}}_K^{FR}\right)_T^T\right) \tag{36}$$

for brevity. The input vector $\vec{\mathbf{u}}_{full} \in \mathbb{R}^3$ is given by

$$\vec{\mathbf{u}}_{full} = \left[\Delta f_{x,K} \quad f_{y,K} \quad \Delta f_{z,K}\right]^{\mathrm{T}}. \tag{37}$$

Additionally, the matrix $\mathbf{G}_{full}\left(\gamma_K, \gamma_T, \Delta\chi\right) \in \mathbb{R}^{3\times3}$ essentially consists of the transformation from the kinematic frame into the trajectory frame

$$\mathbf{T}_{TK} = \mathbf{T}_{TK}\left(-\gamma_K, -\Delta\chi, \gamma_T\right) \in \mathbb{R}^{3\times3}, \tag{38}$$

and the terms of (34) not containing the aircraft dynamics are incorporated in the nonlinearity vector $\vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right) \in \mathbb{R}^3$. Furthermore, $\ddot{\vec{\mathbf{y}}}_{full} \in \mathbb{R}^3$ represents the second time derivative of the output according to the methodology of nonlinear dynamic inversion, where the output is chosen

$$\vec{\mathbf{y}}_{full} = \left(\vec{\mathbf{r}}^{FR}\right). \tag{39}$$

Now, by choosing the control law

$$\vec{\mathbf{u}}_{full} = \mathbf{G}_{full}^{-1}\left(\gamma_K, \gamma_T, \Delta\chi\right)\left(\vec{\boldsymbol{\nu}}_{full} - \vec{\mathbf{f}}_{full}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa\right)\right), \tag{40}$$

the plant (35) would be transformed into

$$\ddot{\vec{\mathbf{y}}}_{full} = \vec{\boldsymbol{\nu}}_{full}, \tag{41}$$

where $\vec{\boldsymbol{\nu}}_{full} \in \mathbb{R}^3$ denotes the pseudo control, which is chosen by the designer. Note that (41) represents an integrator chain, which is stable for an adequate choice of this pseudo control.

3.3.1 Reduced Dynamic Inversion

In an aircraft application, the focus lies on aerodynamic integrity of the aircraft. Hence, for application to general aviation aircraft with lower never exceedance speeds, the ground speed is not the control variable. Therefore, the control law (40) is not applied in this paper since it is designed for geometric deviations incorporating the kinematic x-axis.

In case of the DA42, the calibrated airspeed is used as a control variable employing an auto-thrust to compute thrust lever commands to the ECU. In case of waypoint navigation and altitude hold modes, the speed is controlled by thrust whereas speed by pitch is active for flight level change modes. Note that the trajectory controller always implies speed by thrust as it is used to drive geometrical deviations in both y- and z axes to zero. A planned climb from one waypoint to another is consequently performed using speed by thrust. In case of unachievable vertical changes, certain protections occur, which are addressed in Ref. [17].

As the x-axis is surrendered in the DA42 set-up, the specific force in direction of the kinematic x-axis, $\Delta f_{x,K}$, is not used as a control variable anymore. Consequently, the control system is reduced to second order and $\Delta f_{x,K}$ is taken as a measurement. In order to compute the pseudo control $\nu_{full,x}$, the corresponding row of (40) is employed using $\Delta x = 0$, $\Delta \dot{x} = 0$ and by insertion of $\nu_{full,x}$, we arrive at the reduced system given by

$$\mathbf{u} = \mathbf{G}^{-1}\left(\gamma_K, \gamma_T, \Delta\chi\right)\left(\boldsymbol{\nu} - \mathbf{f}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa, \Delta f_{x,K}\right)\right), \tag{42}$$

where $\mathbf{u} = \begin{bmatrix} f_{y,K} & \Delta f_{z,K} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ is the control input, $\mathbf{f}\left(\gamma_K, \gamma_T, \Delta\chi, \kappa, \Delta f_{x,K}\right) = \begin{bmatrix} f_y & f_z \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ is the nonlinearity vector with the components

$$\begin{aligned} f_y = {} & \Delta f_{x,K}\sin(\Delta\chi)\cos(\gamma_T) - V_K^R\dot{\chi}_T\cos(\gamma_K)\cos(\Delta\chi) \\ & - \left(\dot{\chi}_T\dot{\gamma}_T\cos(\gamma_T) + \ddot{\chi}_T\sin(\gamma_T)\right)\Delta z - \dot{\chi}\sin(\gamma_T)\Delta\dot{z}, \end{aligned} \tag{43}$$

$$\begin{aligned} f_z = {} & \Delta f_{x,K}\left(\cos(\gamma_K)\sin(\gamma_T)\left(\cos(\Delta\chi) - 1\right) - \sin(\gamma_K - \gamma_T)\right) \\ & + V_K^R\dot{\gamma}_T\left(\cos(\gamma_K)\cos(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T)\right) \\ & + V_K^R\dot{\chi}_T\cos(\gamma_K)\sin(\gamma_T)\sin(\Delta\chi) + \dot{\chi}\sin(\gamma_T)\Delta\dot{y} \\ & + \left(\dot{\chi}_T\dot{\gamma}_T\cos(\gamma_T) + \ddot{\chi}_T\sin(\gamma_T)\right)\Delta y, \end{aligned} \tag{44}$$

and the input matrix $\mathbf{G} \in \mathbb{R}^{2\times 2}$ is given by

$$\mathbf{G} \triangleq \begin{bmatrix} \cos(\Delta\chi) & \sin(\Delta\chi)\sin(\gamma_K) \\ -\sin(\Delta\chi)\sin(\gamma_T) & \sin(\gamma_K)\sin(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T) \end{bmatrix}. \tag{45}$$

Note that $\mathbf{G}$ is not invertible if the determinant given by

$$\det\left(\mathbf{G}\right) = \cos(\gamma_K)\cos(\gamma_T)\left(\cos(\Delta\chi) - 1\right) + \cos(\gamma_K - \gamma_T) \tag{46}$$

is zero. This is the case for $|\Delta\chi| \approx 90°$ as the remaining term $\sin(\gamma_K)\sin(\gamma_T)$ is zero for horizontal trajectories and / or horizontal flight. Since fly-by trajectories are planned smoothly, no large deviations of the course angles occur and hence, the inverse can be computed in these cases. Actually, the difference in course angles $\Delta\chi$ stays close to zero even in the presence of wind as the trajectory controller is designed to follow the geometric path rather than using a cascaded structure with course angle commands as commonly used in the literature [1–10]. However, since the determinant takes effect as an additional gain, the difference in course angles is limited to $|\Delta\chi| = 60°$, which corresponds to a gain of 2 if $\gamma_T = 0$ and $\gamma_K = 0$.

For the ideal case of $\Delta\chi = 0$, the inverse of the input matrix becomes diagonal

$$\mathbf{G}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\cos(\gamma_K - \gamma_T)}, \end{bmatrix} \tag{47}$$

which facilitates the possibility to evaluate the effects onto different axes, while the small terms due to the difference in course angles $\Delta\chi$ also disappear from the nonlinearity terms (43), (44), yielding

$$f_y = -V_K^R \dot{\chi}_T \cos(\gamma_K) - (\dot{\chi}_T \dot{\gamma}_T \cos(\gamma_T) + \ddot{\chi}_T \sin(\gamma_T))\, \Delta z - \dot{\chi} \sin(\gamma_T) \Delta\dot{z}, \tag{48}$$

$$f_z = V_K^R \dot{\gamma}_T \cos(\gamma_K - \gamma_T) + (\dot{\chi}_T \dot{\gamma}_T \cos(\gamma_T) + \ddot{\chi}_T \sin(\gamma_T))\, \Delta y + \sin(\gamma_T) \Delta\dot{y} \\ - \Delta f_{x,K} \sin(\gamma_K - \gamma_T), \tag{49}$$

where the first terms correspond to feedforward elements due to the curvature $\dot{\chi}_T$ and $\dot{\gamma}_T$, respectively, the specific force along the kinematic x-axis $\Delta f_{x,K}$ is incorporated by means of another feedforward element to deal with accelerations, and the remaining terms serve as elements to rotate the deviations into the changing trajectory frame. While the ideal case $\Delta\chi = 0$ renders Eqs. (47)–(49), the small terms for $\Delta\chi \neq 0$ correlate to additional coupling terms.

3.3.2 Pseudo Control Law

The pseudo controls are chosen as PD controller as per

$$\boldsymbol{\nu} \triangleq \begin{pmatrix} \nu_y \\ \nu_z \end{pmatrix} = \begin{pmatrix} k_{p,y} \Delta y + k_{d,y} \Delta\dot{y} \\ k_{p,z} \Delta z + k_{d,z} \Delta\dot{z} \end{pmatrix}, \tag{50}$$

where the gains $k_i \in \mathbb{R}^-$ can be computed according to

$$\begin{aligned} k_{p,y} &= -\omega_y^2, \\ k_{d,y} &= -2\omega_y, \\ k_{p,z} &= -\omega_z^2, \\ k_{d,z} &= -2\omega_z, \end{aligned} \tag{51}$$

where ω_y and ω_y are the desired natural frequencies for an aperiodic second order system since no overshoots are desired for trajectory tracking.

4 Illustrative Examples

In this section, the functionality and the performance is further analyzed using illustrative examples. For this purpose, two simulation studies are presented. In the first simulation example, the nonlinear dynamic inversion controller is compared to a linear controller to evaluate the performance with respect to classical linear approaches. Then, the trajectory controller's performance when subject to wind is discussed using another example. Both tests are conducted with the same NDI controller as reference at a commanded airspeed of 90kts.

For the comparison with the linear controller, all input and output treatments applied to the controller remain as designed for the NDI approach. That is, the implicit gain given by the inverse of the input matrix, $\mathbf{G}^{-1}$, the command limits, and the command transformation to the inner loop are retained. Furthermore, the exact same PD control law (51) is used, where $\omega_y = 0.23\frac{rad}{s}$ and $\omega_z = 0.45\frac{rad}{s}$ are utilized in both cases. Consequently, the nonlinearity term given in Eq. (43) is the only remaining difference between the control laws.

For the comparison, a flightplan in triangular form, which consists of various fly-by maneuvers and a loiter at the end, is simulated. The flightplan and the actual flight paths are displayed in Fig. 3. Note that the planned trajectory is identical as the kinematic velocity used for radius computation does not vary throughout the simulation as no wind is applied to address the conceptual differences.

It is evident that the linear controller cannot follow the trajectory in turns and overshoots significantly both when entering and leaving the turn. This fact is further supported by Fig. 4, where large deviations are obvious. In comparison, the NDI controller as illustrated in Fig. 5 remains within deviations of 10 m when building up the roll angle required for turning into or leaving the clothoid, respectively. This deviation can be further reduced by a faster inner loop controller, which, however, is not in the focus right now as conservatism is of higher interest for the current phase of flight testing.

Furthermore, the linear controller exhibits significant steady-state deviations in turns. Although also the NDI controller cannot achieve steady-state accuracy during turns due to the lack of an integrator (in particular during the loiter at the end of the flightplan in Fig. 5), the nonlinearity term approximately succeeds in driving the deviation from the trajectory to zero by means of the feedforward term. Just using a linear error controller as made use of in various applications inherently results in large deviations with respect to curved trajectories as an error needs to be built up for the controller to react. The specific forces commanded by the trajectory controller are in the same magnitude for both controllers, where the NDI controller achieves the slightly faster reaction and the basic turn specific force command by the feedforward term while the linear controller generates its command amplitude by the control error.

For the second example, a spiral pattern is considered, which also consists of various fly-by maneuvers and ends with a loiter. This pattern is chosen to solely contain turns with changing kinematic velocities as the trajectory generation module employs a combination of kinematic velocity, wind estimate, airspeed command, and

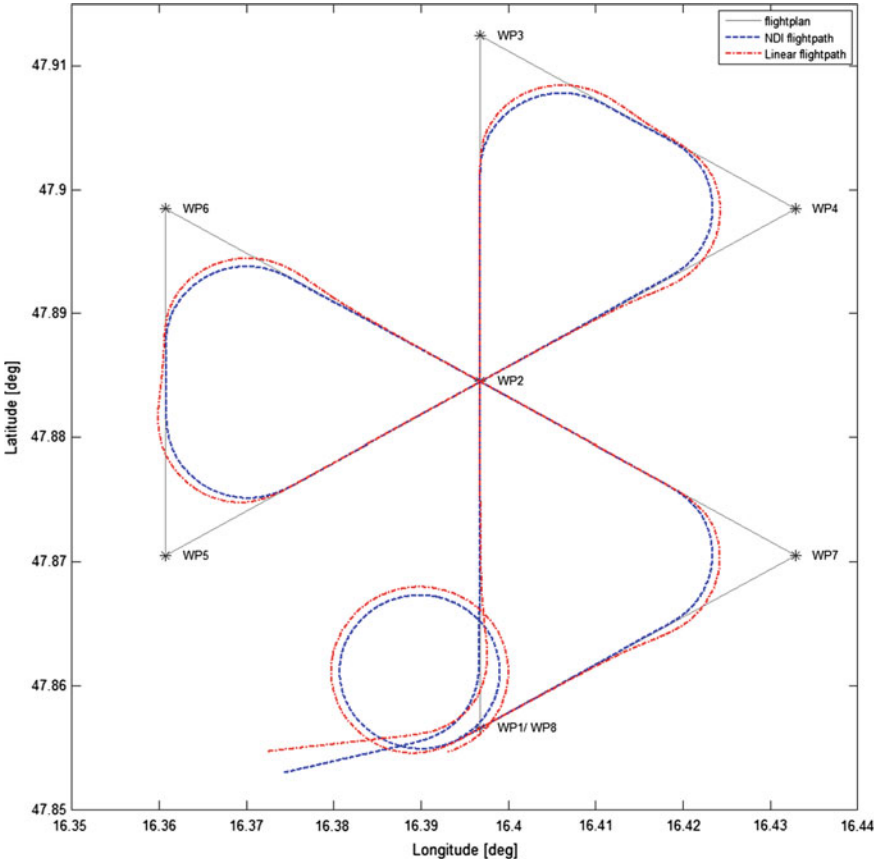


Fig. 3 Flight plan: triangular pattern

the actual airspeed to compute the trajectory's curve as noted in Sect. 2.1. Hence, the trajectory controller can be analyzed with respect to changing velocities, especially with respect to the planned curve. The flightplan displayed in Fig. 6 exhibits the different curves at south wind of 0, 20 and 40kts. As the wind streams towards north, a larger kinematic velocity occurs flying "up" whereas flying south employs the minimum kinematic velocity. As the trajectory generation module utilizes the maximum expected velocity throughout a turn, the planned radii become larger with increasing wind. In particular, this is evident for the loiter maneuver at the end of the flightplan.

Figures 7 and 8 show that there are almost no differences in deviations and course angles between no wind and 20kts of wind. The commands, however, change depending on the kinematic velocity, i.e. the commands required to follow the curve vary significantly to achieve the same objectives. Whereas the turn commands in Fig. 7 retain a certain amplitude, the command's magnitudes change depending on the

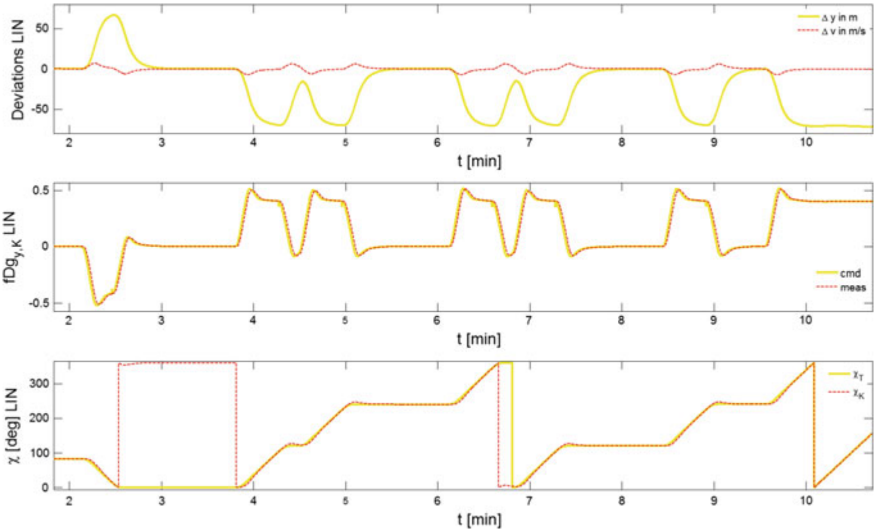


Fig. 4 Linear controller

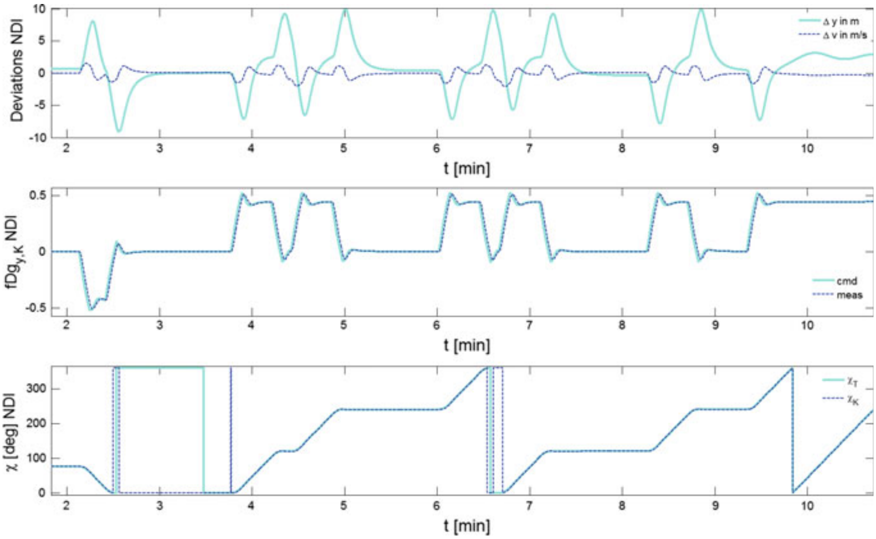


Fig. 5 NDI controller

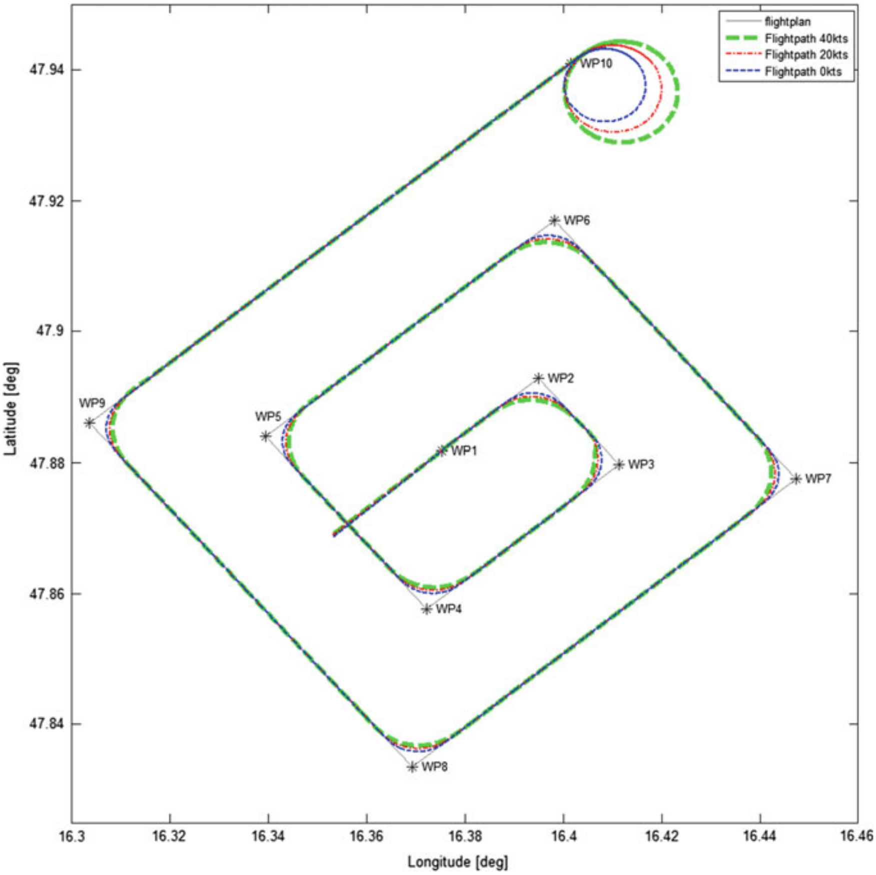


Fig. 6 Flight plan: spiral pattern, wind comparison

direction in the case of 20kts wind as displayed in Fig. 8. When performing the loiter at the end of the flightplan, the change is most evident.

As a conclusion, the trajectory controller achieves superior performance with respect of a similar linear PD controller and is robust against the influences of wind.

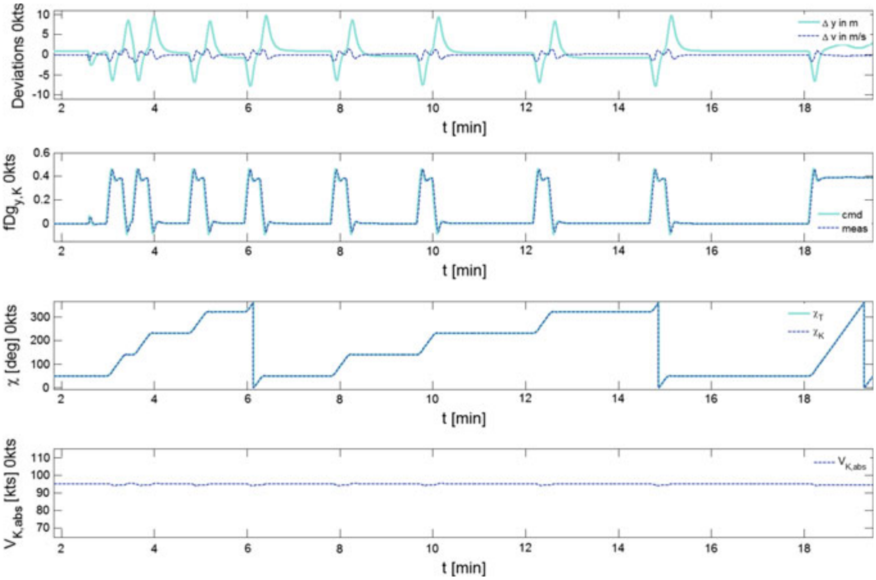


Fig. 7 NDI controller, no wind

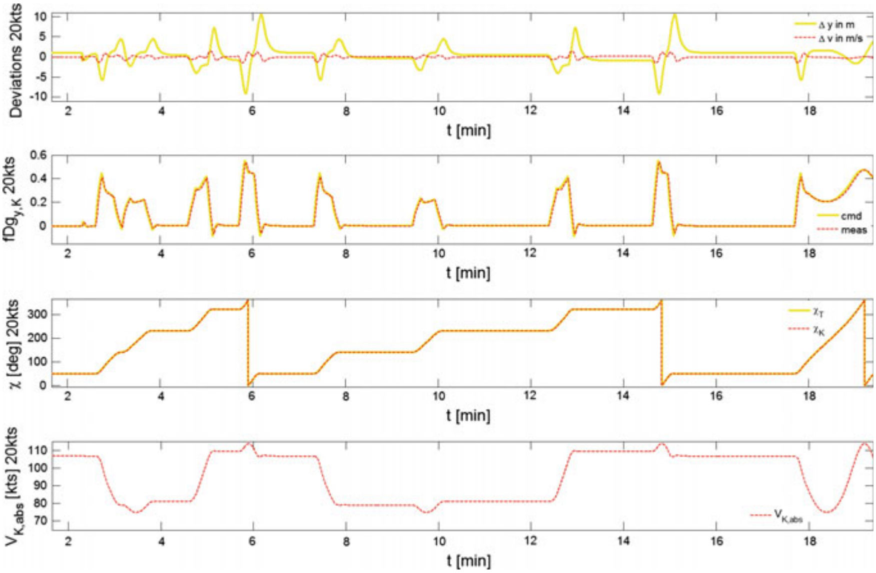


Fig. 8 NDI controller, 20kts N-wind

5 Conclusions

In this paper, a modular trajectory controller was proposed and the concept for fly-by trajectories was described in detail. Specifically, the environment including the interfaces was presented and a brief introduction to the inner loop was given. In simulations, the performance of the nonlinear controller was compared to a linear controller of the same order and the influence of significant wind was analyzed.

In addition to previously-published principal flight tests, further flight tests with the DA42 experimental aircraft of the institute are being conducted focusing on automatic landing by generation of smooth landing trajectories. The modular design of the overall control architecture enables fast applications to other aircraft. Specifically, flight tests with a motor glider and a fixed-wing aircraft of 6 tons are about to be conducted while a small UAV of 150 kg is being tested in hardware-in-the-loop environments.

Acknowledgements I would like to offer my special thanks to my colleague Volker Schneider, who is responsible for the trajectory generation module, which was also used to generate the figures of this paper. Furthermore, I would also like to extend my thanks to Agnes Gabrys, Erik Karlsson, and Alexander Zollitsch, who provided the inner loop, the autopilot module including energy protections, and the high-fidelity simulation model, respectively.

References

1. Beard RW, Kingston D, Quigley M, Snyder D, Christiansen R, Johnson W, McLain T, Goodrich M (2005) Autonomous vehicle technologies for small fixed-wing uavs. J Aerosp Comput Inf Commun 2(1):92–108
2. Low CB (2010) A trajectory tracking control design for fixed-wing unmanned aerial vehicles. In: IEEE multi-conference on systems and control, pp 2118–2123
3. Ren W, Beard RW (2004) Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. IEEE Trans Control Syst Technol 12(5):706–716
4. Subbarao K, Ahmed M (2014) Nonlinear guidance and control laws for three-dimensional target tracking applied to unmanned aerial vehicles. J Aerosp Eng 27(3):604–610
5. Nelson DR, Barber DB, McLain TW, Beard RW (2007) Vector field path following for miniature air vehicles. IEEE Trans Robot 23(3):519–529
6. Zhu S, Wang D, Low CB (2013) Ground target tracking using uav with input constraints. J Intell Robot Syst 69(1–4):417–429
7. Beard RW, Ferrin J, Humpherys J (2014) Fixed wing uav path following in wind with input constraints. IEEE Trans Control Syst Technol :1
8. Zhufeng Xie, Yuanqing Xia, Mengyin Fu (2011) Robust trajectory-tracking method for uav using nonlinear dynamic inversion. In: IEEE 5th international conference on cybernetics and intelligent systems (CIS) pp 93–98
9. Pashilkar AA, Ismail S, Ayyagari R, Sundararajan N (2013) Design of a nonlinear dynamic inversion controller for trajectory following and maneuvering for fixed wing aircraft. In: IEEE symposium on computational intelligence for security and defense applications (CISDA) pp 64–71
10. Oliveira T, Encarnação P (2013) Ground target tracking control system for unmanned aerial vehicles. J Intell Robot Syst 69(1–4):373–387
11. Slotine J-JE, Li W (1991) Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, NJ

12. Khalil HK (2002) Nonlinear Systems, 3rd edn. Prentice Hall, Englewood Cliffs, NJ
13. Schatz SP, Holzapfel F (2014) Modular trajectory, path following controller using nonlinear error dynamics. In: IEEE international aerospace electronics and remote sensing technology (ICARES) pp 157–163
14. Schatz SP, Schneider V, Karlsson E, Holzapfel F et al (2016) Flightplan flight tests of an experimental da42 general aviation aircraft. In: The 14th International Conference on Control, Automation, Robotics, and Vision
15. Krause C, Holzapfel F (2016) Designing a system automation for a novel uav demonstrator. In: The 14th International Conference on Control, Automation, Robotics, and Vision
16. Karlsson E, Schatz SP et al (2016) Automatic flight path control of an experimental da42 general aviation aircraft. In: The 14th International Conference on Control, Automation, Robotics, and Vision
17. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: The 14th International Conference on Control, Automation, Robotics, and Vision
18. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2014 IEEE International Aerospace Electronics and Remote Sensing Technology (ICARES). IEEE
19. Schneider V, Piprek P, Schatz SP et al (2016) Online trajectory generation using clothoid segments. In: The 14th international conference on control, automation, robotics, and vision
20. Meek DS, Walton DJ (2004) A note on finding clothoids. J Comput Appl Math 170(2):433–453
21. Henrie J, Wilde D (2007) Planning continuous curvature paths using constructive polylines. J Aerosp Comput Inf Commun 4(12):1143–1157
22. Wilde DK (2009) Computing clothoid segments for trajectory generation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS) pp 2440–2445
23. Lavretsky E, Wise KA (2013) Robust and Adaptive Control: With Aerospace Applications, Ser. Advanced textbooks in control and signal processing. Springer, New York
24. Andry AN, Shapiro EY, Chung J (1983) Eigenstructure assignment for linear systems. IEEE Trans Aerosp Electron Syst AES–19(5):711–729
25. Etkin B, Reid LD (1996) Dynamics of Flight: Stability and Control, 3rd edn. Wiley, New York
26. Stengel RF (2004) Flight Dynamics. Princeton University Press, Princeton and NJ

Modular Trajectory Generation Test Platform for Real Flight Systems

Volker Schneider and Florian Holzapfel

1 Introduction

Over the last decades air traffic has been steadily increasing. Furthermore, the planned integration of Unmanned Aerial Systems (UAS) and Optionally Piloted Vehicles (OPV) in the non-segregated airspace confronts the field of trajectory generation with tasks to combine a high level of safety with broad flexibility. It requires the construction of complex flight paths, which are needed for further airspace- and route-planning. One example is the development of new operational concepts including a 4-dimensional trajectory for the civil airspace as part of the programmes SESAR or NextGen as shown in [1].

Various approaches exist to meet the upcoming challenges, which can be clustered according to different criteria.

One famous approach is the use of optimal control for smooth path planning [2, 3]. These approaches use iterative algorithms, which lead to special safety considerations. Since the scope of the presented test platform lies on algorithms without iteration parts, they were not considered in the requirement specification of the current trajectory generation system.

The path parametrization is another important section of research. Here, for example, Bézier curves [4], clothoids for transition between arc and straight line segments [5, 6], or logistic curves as turn approaches [7] are discussed. Both two-dimensional [8] as well as the coupling of all axes to a three-dimensional algorithm [9, 10] are being used.

V. Schneider (✉) · F. Holzapfel
Institute of Flight System Dynamics, Technische Universität München,
Boltzmannstr. 15, 85748 Garching bei München, Germany
e-mail: volker.schneider@tum.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_10

Due to the wide range of possible research fields, the Institute of Flight System Dynamics has developed a test platform, which allows the evaluation of new algorithms in real flight. A short description of the system can be found in Chap. 6. The flight control system software consists of a stabilizing inner loop system, an autopilot system [11, 12] and trajectory flying.

The trajectory flying part is separated into the trajectory generation, which is presented in this paper and the trajectory control. Trajectory control uses a second order error dynamics controller [13, 14].

The present paper focusses on the functional design of the trajectory generation system. The objective of the system is to combine two main aspects. On the one hand, a robust interface to the adjacent systems with standardized specifications intends to ensure the usage of the system together with external devices, e.g. from other project partners. On the other hand, the structure is meant to provide the ability to easily extend the functionalities or to replace sub modules, which is necessary to use it as a test platform for different research approaches.

In Chap. 2 the requirements for the trajectory generation testing platform are stated. Furthermore, the interfaces to the adjacent modules are specified. Chapter 3 presents the functional design, including the sub functions of the trajectory generation, while Chap. 4 especially considers non-nominal flight cases and presents strategies to deal with them from trajectory generation view. In Chap. 5, system aspects of the trajectory generation are discussed, including the allocation of the functions to different systems due to their criticality. Chapter 6 presents two project examples, where the trajectory generation system is used in different platforms.

1.1 Nomenclature

In the present paper the following waypoint names are used:

FROM – The waypoint, the aircraft comes from.
TO – The current active waypoint, the aircraft proceeds to.
NEXT – The waypoint, the aircraft will proceed to after reaching the TO.
MOD – A waypoint, which is modifiable within the current waypoint buffer.

The following coordinate frames are mentioned:

Trajectory frame: The trajectory frame is defined with its origin being located in the trajectory reference point. The x-axis points along the trajectory tangent, the y-axis points orthogonal to the x-axis to the right side of the trajectory.

North East Down (NED) frame: The origin of the NED frame is located in the aircraft reference point, the x-axis points towards north, and the y-axis points to the east, the z-axis points downwards, orthogonal to the earth surface.

2 Requirements and Interfaces

Since the trajectory generation platform is developed to cover a broad range of projects and scopes, the functional requirements focus on a generic design. They also distinguish between a first prototype demonstration and the future vision of the functional extent, especially facing interface considerations. The top level functional requirements are as follows:

1. The system shall be able to handle up to 8 flight plans, where each flight plan can be activated during flight.
2. Each flight plan shall handle up to 99 waypoints.
3. The operator shall be able to start a flight plan at any desired flight plan waypoint.
4. The operator shall be able to edit at least one flight plan during flight via list update.
5. The system shall be able to handle ARINC 424 "Navigation System Database" [17] leg types, oriented on the subset of RTCA-DO 236C "Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation" [18]. (Track to Fix, Radius to Fix, Course to Fix, Holding to altitude, Holding to manual termination, Holding to Fix, Direct to Fix). The first prototype shall provide "Track to Fix" leg type.
6. For a Track to Fix leg, the system shall be able to perform a Flyby or Flyover maneuver or to perform a Loiter maneuver until exit command by the operator. The maneuvers shall be performed as illustrated in Fig. 1.
7. The operator shall be able to command an "immediate Loiter" as a waiting maneuver within each phase of trajectory flying.

Furthermore requirements regarding the module and interface design are stated:

1. The system shall be developed as a research platform, which is clustered in replaceable sub modules.

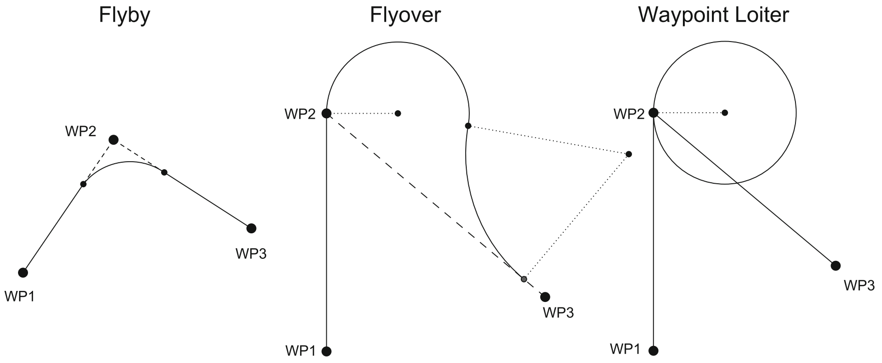


Fig. 1 Required transitions between two "Track to Fix" legs

2. The sub modules shall provide a function separation, regarding safety aspects. Low and high critical functions shall run independently, so that lower critical functions can easily be substituted by new developments without affecting the rest.
3. The system shall be able to combine vertical autopilot modes and autothrust modes with horizontal waypoint flying.

Regarding the requirements, the system interface is specified in a way that allows a wide variation of different trajectory generation algorithms without changing the interface to the adjacent systems.

The input interface contains a commanded flight plan number, a commanded waypoint number and furthermore at least one flight plan list to edit online. Each flight plan consists of 99 waypoints, where Table 1 shows the waypoint attributes used.

Each waypoint is uniquely addressable by a waypoint ID. Besides position, altitude and desired speed, each waypoint contains an attribute "ARINC 424 leg type", which is responsible for choosing up to 3 generic attributes "Property 1–3". For the prototype development, the leg type "Track to Fix" needs an additional attribute marking the desired transition type. In this case, "Property 2" and "Property 3" are being left meaningless.

The output interface was designed to support a second order error dynamics controller as published in [13, 14]. Hence, additional values, compared to a conventional system, need to be provided. They are illustrated in Fig. 2.

The output interface of the trajectory generation system comprises the error dynamics up to the first order, namely the error between aircraft and trajectory ref-

Table 1 Waypoint properties

Attribute	Description	Unit
Waypoint ID	A unique waypoint ID to explicitly address the waypoint	–
Longitude	WGS84 longitude position	Rad
Latitude	WGS84 latitude position	Rad
Altitude	Waypoint altitude (WGS84 or barometric, dependent on sensor source selection in the trajectory parameter list)	Meter
IAS speed	IAS speed on the leg before the waypoint	Meter per second
ARINC 424 leg type	Leg type according to specification "ARINC 424 – Navigation System Database"	–
Property 1	Free waypoint properties, which are dependent on the ARINC 424 leg type, e.g. transition type for "Track to Fix"	Dependent
Property 2		Dependent
Property 3		Dependent

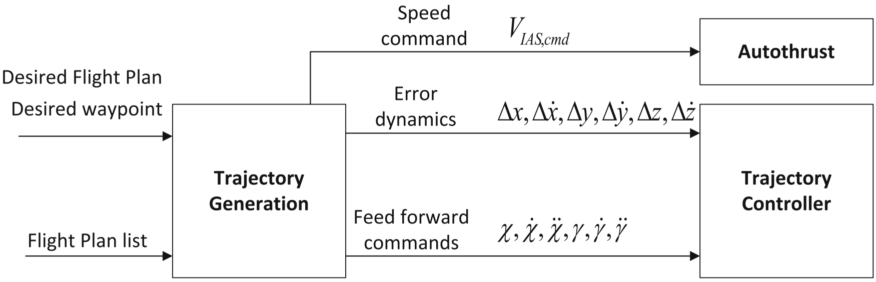


Fig. 2 Description of the system output interface

erence point plus the first derivative. All deviations and derivatives are given in the trajectory frame. Furthermore the trajectory angles and the corresponding derivatives are provided up to the second order.

Note that, in general, the trajectory controller is able to control the aircraft in all three axes, where the Δx command corresponds to a desired kinematic speed. However, the use of kinematic speed as control variable could lead to unsafe flight states depending on the actual wind condition. Hence a direct airspeed control is preferred, where speed protections can be realized much easier. As mentioned in the introduction, the trajectory generation system is part of an integrated Flight Guidance and Control System containing an independent autothrust with indicated airspeed (IAS) control. Thus, the trajectory generation also provides an IAS command output, which is directly used as control variable for the autothrust system during waypoint flying.

With the use of this approach the question arises on how to determine the trajectory reference point. Letting the reference run point through the trajectory with a constant kinematic speed is invalid, since the aircraft does not use kinematic speed as control variable. Hence, the current system determines the trajectory reference point always as the point on the trajectory with the shortest distance to the aircraft. This is the point on the trajectory where the vector between aircraft and reference point is orthogonal to the actual trajectory tangent. Since the deviations are given in the trajectory frame, this approach always results in a deviation $\Delta x = 0$ and in the corresponding derivative $\Delta\dot{x} = 0$. The interface is nevertheless installed, since special projects could also use approaches with a kinematic speed command.

3 Functional Design

The objective of the given system was to provide a basis for a wide range of projects with different scopes containing replaceable sub functions. Therefore, the first step was to classify the trajectory generation function into atomic units, which form the main functional parts of the system. The classification yields the following 6 main functional groups:

1. Flight plan handling
2. Geometry calculations
3. Function moding
4. Trajectory reference point
5. Error dynamics
6. Derivatives

In the following subsections, each of those sub modules will be explained in detail.

3.1 Flight Plan Handling

This module comprises all functions, which are related to the administration of the available flight plans. This includes flight plan edition, such as adding or removing waypoints, the determination of the current active waypoints and the moding, which is responsible for flight plan activation or switching the waypoint when reaching the current TO.

3.2 Geometry Calculations

The geometry calculations module contains all calculations which are needed for the correct flight path generation. They can be determined using the current active waypoints and the aircraft position. In general, the geometry calculations can be separated into different sub parts. The position calculations include the transformation of the aircraft position and the waypoint positions into all necessary frames. All waypoints and positions are initially given in WGS84, which requires a spherical or even ellipsoid geometry calculation. For the cross track error dynamics in horizontal behavior however, a Euclidian approach can be applied as long as the great circle trajectory completely lies in a plane, orthogonal to the xy-plane of the used frame. Figure 3 illustrates this aspect using the North-East-Down (NED) frame.

The NED frame, which has its origin at the aircraft reference point, fulfills the upper requirement as long as the aircraft proceeds exactly on the commanded great circle. If there exist deviations between commanded trajectory and aircraft flight path, the resulting error can be neglected, assuming that for waypoint flying, the aircraft is always located near the current trajectory. Since the NED frame moves with the aircraft, the effect of a changing course angle on a great circle flight is automatically covered. Hence, the NED frame is used as the basis for trajectory calculation regarding the cross track error and its deviations.

The vertical error Δz cannot be determined using Euclidian geometry. Assuming a flight with constant altitude on a great circle as illustrated in Fig. 3, the projection

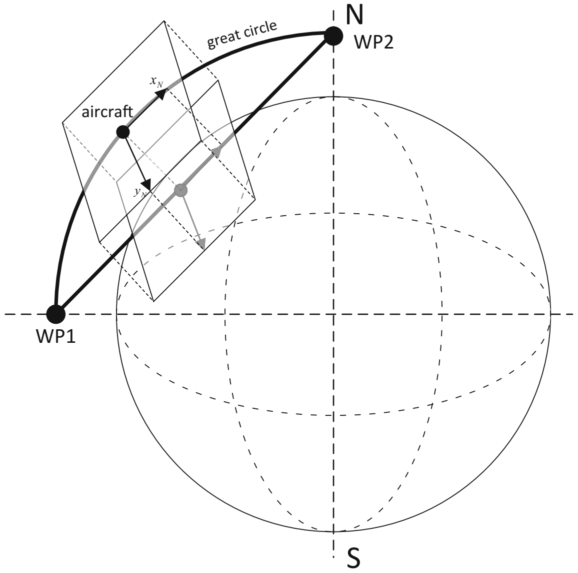


Fig. 3 Use of the North-East-Down frame during great circle flight for cross track error determination

on the corresponding linear connection would result in a change in altitude command during flight. Hence, the vertical geometry always needs to be calculated using the measured barometric or WGS84 altitude. An implementation example can be found within Sect. 3.4.

3.3 Function Moding

The function moding is responsible for the correct activation and deactivation of all necessary sub modes, such as straight line flight, leg transition (Flyby or Flyover) and Loiter maneuver. The key aspect for the moding design was the ability to easily extent the modes according to additional functionalities in the future. Hence, a two level design was implemented. The top level contains modes, which are not necessarily connected to an active flight plan. An example here is the required function to activate an operator Loiter within each situation. This also comprises its use, when the autopilot is active without waypoint flying. The second level contains sub modes for each top level part. The most obvious are those of the trajgen mode, where the sub structure separates into Straight Line, Flyby, Loiter at waypoint or a Flyover command.

Figure 4 illustrates the moding design containing the currently implemented modes.

Note, that Fig. 4 shows a vertical part, which contains the “Standard Mode” and the “Separated Mode”. Since horizontal and vertical axes are calculated separately, the moding needs to face this aspect, too. In most cases, the vertical modes are

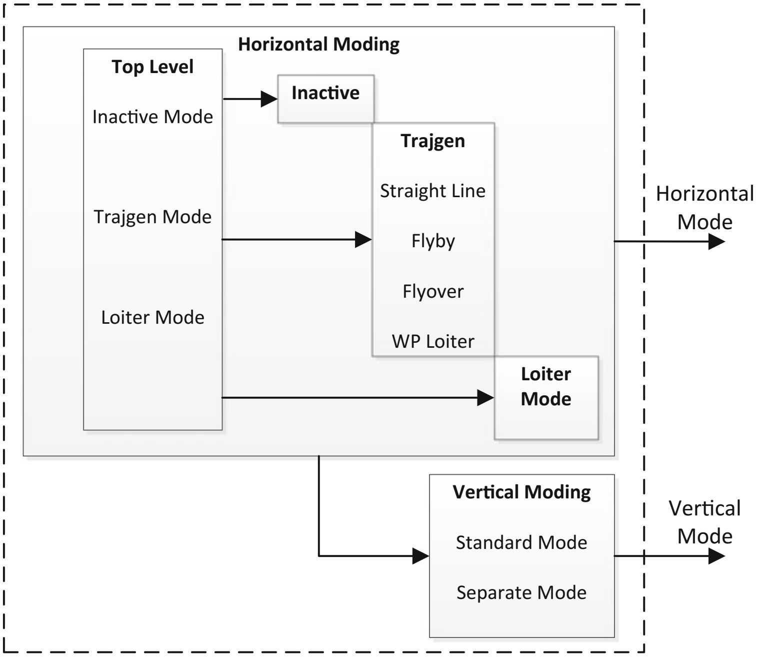


Fig. 4 Moding design including the separation of horizontal and vertical behavior

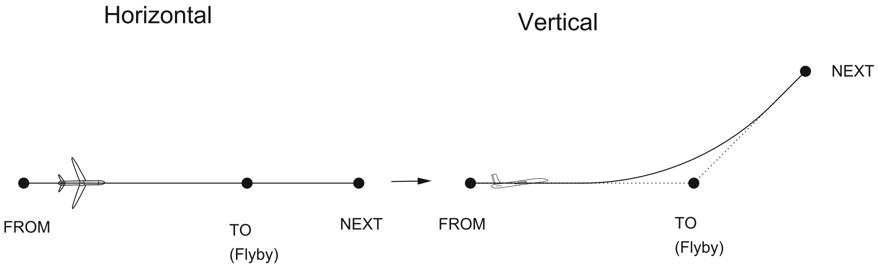


Fig. 5 Vertical Flyby maneuver

identical to the corresponding horizontal behavior. These cases are represented by the “Standard Mode”. However, in special flight situations the vertical moding needs to act independently. Assuming a vertical Flyby, where the current and the following leg are in line, the Vertical Flyby has to be activated during horizontal straight line flight. Figure 5 illustrates such a case.

Here, the “Separate Mode” is activated at the moment when the Vertical Flyby maneuver needs to start. Note, that any interruption of the current separate maneuver, such as a new flight plan command or an operator Loiter command, resets the vertical moding back to “Standard Mode”.

3.4 Trajectory Reference Point Calculation

This module is responsible for all calculations regarding the flight path. As mentioned in Sect. 2, the reference point is defined as the point on the trajectory with the shortest distance to the aircraft. The current implementation comprises straight line calculations, arc segments, as shown in [15] and clothoid maneuvers for the transition between straight line and arc, as published in [5]. The module comprises the reference point and the trajectory course angle calculation for the horizontal behavior ($x_{T,F}$, $y_{T,F}$, $\chi_{T,F}$) and the trajectory altitude and flight path angle ($h_{T,F}$, $\gamma_{T,F}$) calculation for the vertical behavior. As also explained in Sect. 3.2, the values for vertical movement need to be determined using the barometric or WGS84 altitude instead of the linear geometry. For the current projects, the implementation of altitude and flight path angle distinguishes between straight line flight and turn flight. For straight line flight, a linear altitude change between starting point and ending point is assumed.

$$h_{line} = \left(h_{Wp2} - h_{Wp1}\right) \cdot \frac{d_{Wp1ref}}{d_{Wp1Wp2}} + h_{Wp1} \tag{1}$$

$$\gamma_{line} = atan\left(\frac{h_{Wp2} - h_{Wp1}}{d_{Wp1Wp2,hor}}\right) \tag{2}$$

For a turn flight, the altitude change is realized assuming a 5th order polynom, where the coefficients are determined with the following constraints.

$$\begin{gathered} 0 \le x \le d \\ h\,(0) = h_{start}, \qquad h\,(d) = h_{end} \\ \gamma\,(0) = \gamma_{start}, \qquad \gamma\,(d) = \gamma_{end} \\ \dot{\gamma}\,(0) = 0, \qquad \dot{\gamma}\,(d) = 0 \end{gathered}$$

$$h_{turn}\,(x) = a \cdot x^5 + b \cdot x^4 + c \cdot x^3 + d \cdot x^2 + e \cdot x + f \tag{3}$$

$$\gamma_{turn}\,(x) = atan(5a \cdot x^4 + 4b \cdot x^3 + 3c \cdot x^2 + 2d \cdot x + e) \tag{4}$$

3.5 Error Dynamics and Derivatives

As mentioned above, the integrated Flight Guidance and Control System uses a second order error dynamics controller. Hence, the necessary output for the trajectory generation system is not only the reference point on the trajectory but the error dynamics and its derivatives. The deviation itself can simply be determined by transforming the trajectory reference point as well as the aircraft position into a trajectory frame. This was also shown in [15].

The derivative determination is treated as an own sub system since the trajectory generation system shall be developed as a generic system, which can be used for different controller applications. Hence, the level of derivative calculation can differ. In [15, 16], two ways for the determination of the derivatives were presented. The current implementation uses the second approach, where the current aircraft position is predicted using the equations of motion. The predicted aircraft position is then used to determine a predicted trajectory reference point. The difference between the current reference point and predicted reference point is then the basis for the desired derivative value.

4 Non-nominal Behavior

For a safe and deterministic aircraft behavior in all situations, the trajectory generation algorithm has to consider waypoint combinations, which yield non-nominal behavior.

In general, a non-nominal case is a combination of waypoints, which forms a non-achievable trajectory due to limited aircraft performance. From the trajectory generation view it is important to focus on a deterministic behavior of the system. An automatic flight plan replanning would result in a route, which is not necessarily desired. Only the operator himself is authorized to redesign given waypoints before they get in command. On the other hand, a non-reachable waypoint, must not result in undesired aircraft maneuvers, which try to reach the waypoint. Hence, the following remarks are implemented for the trajectory generation system:

1. The switch conditions for the aircraft are not based on the aircraft position, but on the calculated trajectory reference point. This guarantees, that the flight plan is switched to the next waypoint, even if the aircraft has not reached the waypoint.
2. A safe aircraft flight state is guaranteed by controller limitations and not by the trajectory generation. Hence, non-reachable waypoints are commanded to the controller the same way as nominal waypoints. This approach guarantees a maximum controller performance usage to get as close to the desired point as possible.

There are few exceptions, where the trajectory generation can directly influence the commanded path, since the nominal flight path would result in uncommon aircraft maneuvers.

1. A Flyby Maneuver for a “Track to Fix” leg is only allowed for course changes which are smaller than 125°, since maneuver for bigger course changes would result in turns, which are very far away from the TO. For bigger course changes, an automatic Flyover is performed at the TO.
2. A Flyby maneuver for a “Track to Fix” is only allowed as long as the target leg is long enough to finalize the Flyby. Otherwise, the Flyby maneuver is replaced

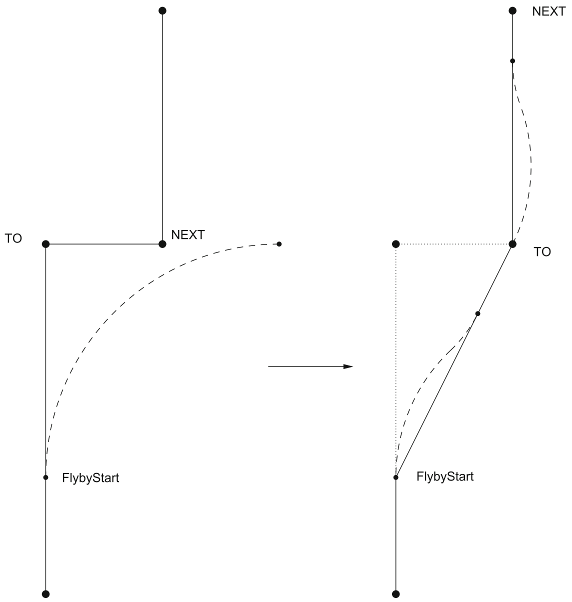


Fig. 6 Flyby abort caused by too short target leg

by a Flyover at the Flyby starting point and a direct connection between starting point and the NEXT, as illustrated in Fig. 6.

3. A Flyby maneuver is aborted and replaced by a Flyover maneuver, if the distance between aircraft and the TO waypoint is insufficient to perform the Flyby maneuver at the moment of activation. The maneuver is aborted, since otherwise it would result in an initial distance between aircraft and trajectory, yielding a non-desired step in the controller commands. Figure 7 illustrates this case.

Note that the list above is based on the current implementation status. A function extension can result in further situations, where a direct influence of the trajectory generation system on the flight path is necessary.

5 System Structure and Internal Interface Considerations

The functional breakdown, described in Sect. 3, clusters the trajectory generation into atomic sub parts with various tasks. The requirements, expressed in Sect. 2, also describe the need to distinguish those functions from a safety point of view, that is to cluster them according to their criticality. Analysing the extracted atomic sub functions, there are two safety related categories:

Flight Planning: Those functions normally have a lower criticality, since they do not directly command changes of the flight path or even direct flight control commands.

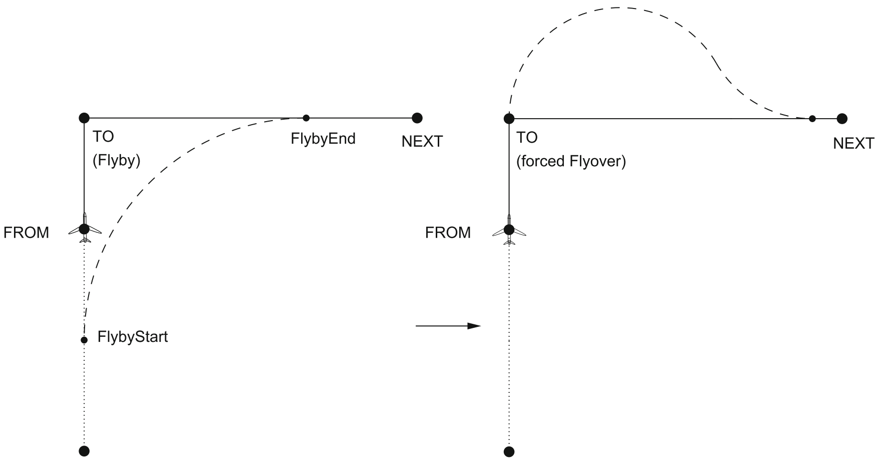


Fig. 7 Flyby abort caused by non-sufficient distance to TO waypoint

They are therefore not able to bring the aircraft into dangerous flight conditions. Furthermore, those functions are not sensitive to a periodical update, and therefore not time critical.

Flight Path Generation: This functional group contains all tasks which are necessary to generate the controller commands. Since those commands need to be updated at each time step and they directly affect the aircraft behavior, these functions have a higher criticality. The "Flight Path Generation" comprises the geometry calculation, the function moding, trajectory reference point calculations and the error dynamics including the derivative determination.

Figure 8 illustrates the function separation according to safety aspects into two independent models.

A separation of functions due to their criticality is only valid as long as there exists an interface between them, which is able to cover the separation aspects. The interface design needs to consider that the core functionalities have all information needed, even in case of failure. Hence, at least a sub set of flight planning data must remain available for the Flight Path Generation, containing those waypoints, that are directly used for flight path construction, namely FROM, TO and NEXT. Even the use of such a buffer leads to defective situations at the moment of a waypoint switch in case of an interface error, since the former FROM has to be updated externally while being in command. To avoid such a situation, a fourth waypoint MOD is added to the waypoint buffer, which nominally contains the AFTERNEXT. During waypoint flying, a waypoint switch is then realized as illustrated in Fig. 9.

As soon as the current TO is reached, the waypoint buffer is updated without external data. The update of the TO, which is recognized via the ID change, triggers the Flight Planning to refresh the MOD with a new waypoint from the flight plan.

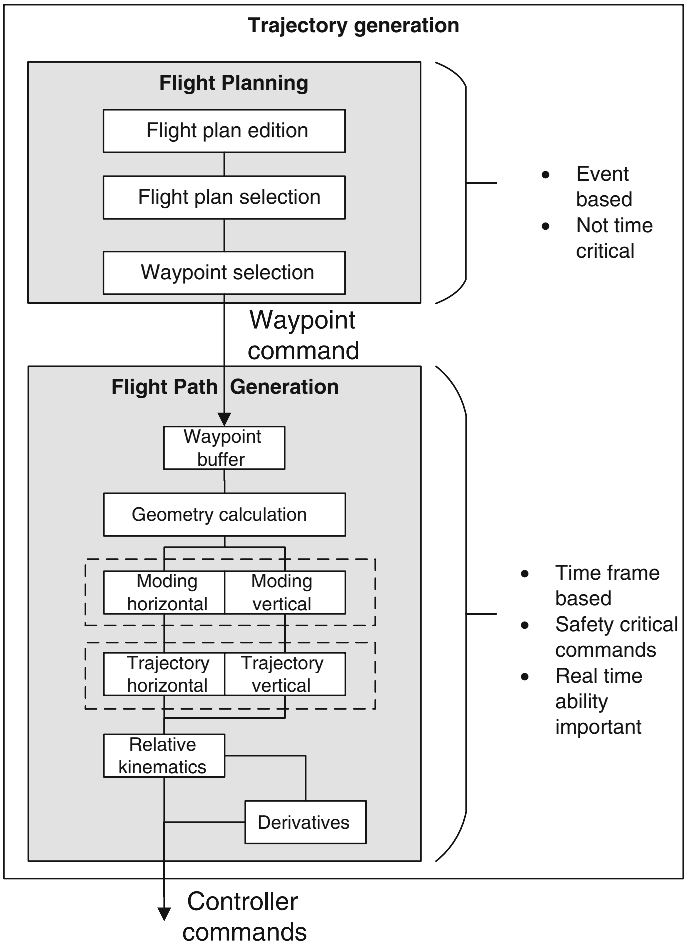
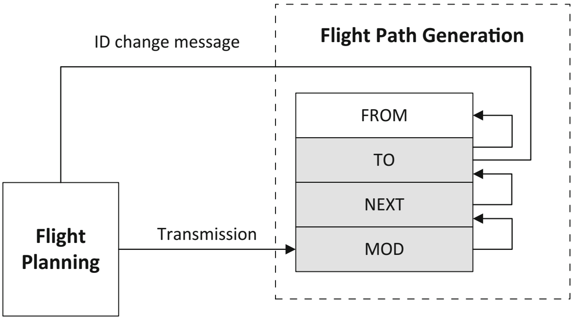


Fig. 8 System structure of the trajectory generation test platform

Fig. 9 Waypoint switch during waypoint flying



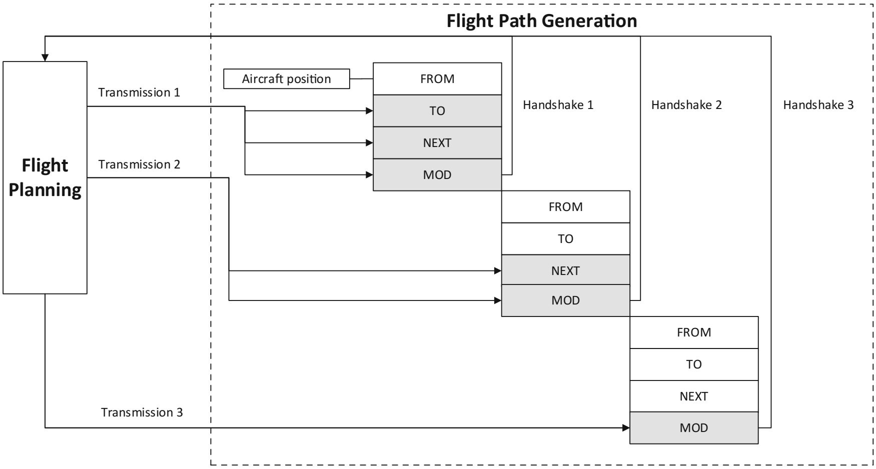


Fig. 10 Serial transmission process for a new flight plan command

Since the MOD is then not in command, the update is not time critical and an external update is valid.

A special case for waypoint update is the command of a new flight plan. In this case, the whole waypoint buffer has to be updated and there is no alternative to an access to the waypoints in command. To ensure a correct interpretation of the new buffer content, the safest way is to send all waypoints at once. However, the design for two independent systems directly yields the problem of a limited interface bandwidth. Hence a serial transmission approach was developed as shown in Fig. 10.

Each waypoint transmission contains an attribute, if the actual data is a TO, a NEXT or a MOD. A “TO” transmission results in an update of the waypoint buffer, setting the FROM to the actual aircraft position and all other buffer entries to the current TO. In case that this is the only message, the flight path generation algorithm interprets the same waypoint ID for TO and NEXT as the flight plan end, proceeds directly to this waypoint and performs a Loiter maneuver until a new command is given.

The attribute “NEXT” for the transmission replaces NEXT and MOD in the buffer. In case that this was the last message, the algorithm proceeds to TO, then to NEXT and recognizes the end of the flight plan there.

Finally, a “MOD” transmission only replaces the MOD and is identical to the waypoint switch transmission, presented above.

A complete update of the waypoint buffer is hence proceeded with three transmissions, where the actual MOD ID is used for handshake back to the flight plan algorithm.

6 Integration Examples

The following subsection presents two project examples, where the described modular architecture was integrated and used on different implementation stages and various flight systems.

6.1 Diamond DA42 Experimental Aircraft

The most important demonstration example is given by the experimental aircraft, operated by the Institute of Flight System Dynamics, TU Munich. It is a modified Diamond DA 42 M-NG, a CS-23 aircraft with two piston engines and a maximum take-off weight of about 2000 kg. The institute's aircraft is equipped with an experimental flight control system, which has direct access to the control surfaces via additionally installed actuation. An experimental Flight Control Computer (FCC) allows the operation of self-implemented algorithms ranging from pilot augmentation up to autopilot or waypoint flying modes to operate the aircraft highly automated. For all flights, a safety pilot is on board, who has access to the conventional flight controls. A special safety system, which was explicitly developed for this aircraft, guarantees that the safety pilot can always override the controls of the experimental system. The standard crew of the aircraft comprises the safety pilot, an experimental pilot, who can control the aircraft by an active stick or a mode control panel, which are connected to the experimental system and a flight test engineer who always supervises the behavior of the experimental system.

The flight control software for the system contains an inner loop, which generates the actuator commands for surfaces and engines. The autopilot system is a load factor based module. It contains a module for basic autopilot operations for all 3 axes and a trajectory controller, which is responsible for horizontal and vertical trajectory control. This controller is the basis for the presented trajectory generation system as well as for the auto landing module.

All modules are supervised by a system automation, which is responsible for the correct activation and deactivation of all functions dependent on the operator commands and on the results of the input monitoring.

The flight control software structure is illustrated in Fig. 11.

The trajectory generation system was developed and demonstrated with this aircraft in two steps.

The first version was implemented using a combination of straight lines and arc segments for the trajectory reference point calculation. This version was mainly used for simulation and Hardware-in-the-loop tests. Simulation results were presented in [15]. This approach was also used to demonstrate the principal functionality of the system during the first real flights with the DA42 experimental aircraft. The functions of the system comprised the separation into Flight Planning and Flight Path Generation, the flight plan- and waypoint handling as well as the maneuvers of Flyby, Flyover, Loiter at waypoint and Loiter by operator.

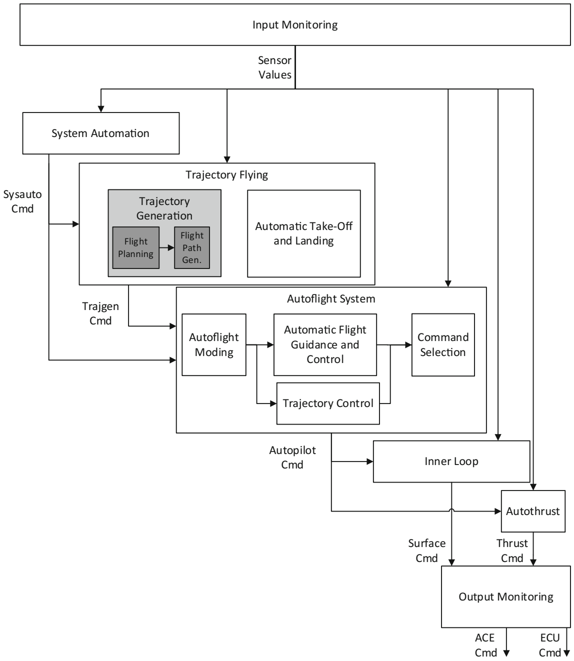


Fig. 11 Breakdown of the Flight Guidance and Control module for the DA42 experimental aircraft

Since a straight line and arc concept leads to the problem of curvature steps between straight line and arc, the trajectory reference point sub module was replaced by a more accurate approach, which uses clothoid curves for the transition between straight lines and arc segments. The extension was demonstrated during a real flight experiment [5]. For the extended implementation, the general system architecture of the trajectory generation system has not been changed. Only the clothoid specific block has been adapted, so that the whole system could be upgraded quite easily.

6.2 *Flight Path Generation for Electric UAV Project*

For a project concerning the development of a flight control system for an electric UAV, the Institute of Flight System Dynamics also provided an autopilot system including waypoint flying. The Flight Planning part however was not task of the institute but realized by another project partner. Hence, the Flight Path Generation is used as a standalone module within the trajectory generation system in this project. Since the Flight Path Generation has a fix input interface, which is able to run without time critical restrictions, the external project partner was completely free in implementing the Flight Planning. Besides this aspect and the non-existent autolanding module, the software structure is more or less comparable to the structure described in Sect. 6.1. Figure 12 illustrates the functional architecture of the whole project system.

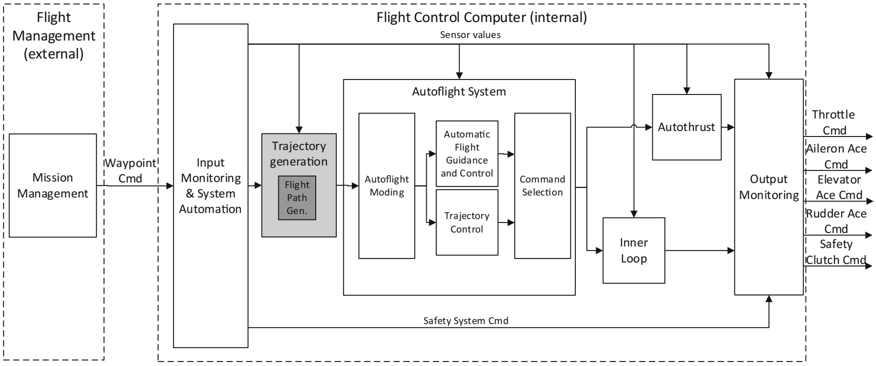


Fig. 12 Breakdown of the Flight Guidance and Control System for the standalone Flight Path Generation operation within the trajectory generation system

Note, that due to a limited FCC-performance, only a functional subset of the Flight Path Generation was realized. It merely calculates the course angle between the aircraft and the TO at each time step and using this value as control variable. The general separation of the functional sub modules was kept too, since this allows a later upgrade to the more complex clothoid approach.

7 Conclusion

The present paper provides a modular platform for trajectory generation applications. The software was designed as part of an integrated Flight Guidance and Control System, where waypoint flying is possible in combination with standard autopilot commands. The focus of the development was the modular structure, which allows its usage within a broad range of research projects.

Future work is seen in extending the given functionality, as the current implementation provides only a subset of waypoint flying opportunities. An example is the implementation of Holding patterns or additional ARINC 424 leg types.

Acknowledgements I would like to offer my special thanks to my colleague Simon Schatz, who is responsible for the trajectory control module, which is the basis for the trajectory generation system. Furthermore, special thanks to Christoph Krause, who supported the Hardware-In-The-Loop-Simulation of the electric UAV project, and to Daniel Gierszewski, who supported the creation of the presented paper.

References

1. Gardi A, de Ridder K, Sabatini R, Ramasamy S (2013) 4-Dimensional trajectory negotiation and validation system for the next generation air traffic management. In: AIAA guidance, navigation, and control conference
2. Diaz-Mercado Y, Lee SG, Egerstedt M, Young S-Y (2013) Optimal trajectory generation for next generation flight management systems. 2013 IEEE/AIAA 32nd digital avionics systems conference (DASC), pp 3C5-1–3C5-10
3. Zhao Y, Tsiotras P (2013) Time-optimal path following for fixed-wing aircraft. J Guid Control Dyn 36(1):83–95
4. Choi J-W, Curry R, Elkaim G (2008) Path planning based on Bézier curve for autonomous ground vehicles. In: Advances in electrical and electronics engineering - IAENG special edition of the world congress on engineering and computer science (WCECS), San Francisco, California, USA, 2008, pp 158–166
5. Schneider V et al (2016) Online trajectory generation using clothoid segments. In: IEEE interntational conference on control, automation, robotics and vision (ICARCV)
6. Brandse J, Mulder M, Van Paassen M (2004) Advanced trajectory design for the tunnel-in-the-sky display: the use of clothoids. In: AIAA guidance, navigation, and control conference and exhibit
7. Upadhyay S, Ratnoo A (2016) Smooth trajectory planning for MAVs with airspace restrictions. In: AIAA guidance, navigation, and control conference
8. Bauer P, Dorobantu A (2013) Optimal waypoint guidance, trajectory design and tracking. In: 2013 American control conference (ACC), pp 812–817
9. Babaei AR, Mortazavi M (2010) Three-dimensional curvature-constrained trajectory planning based on in-flight waypoints. J Aircr 47(4):1391–1398
10. Ambrosino G et al (2006) Algorithms for 3D UAV path generation and tracking. In: IEEE conference on decision and control
11. Karlsson E et al (2016) Automatic flight path control of an experimental DA42 general aviation aircraft. In: IEEE interntational conference on control, automation, robotics and vision (ICARCV)
12. Karlsson E, Gabrys A, Schatz SP, Holzapfel F (2016) Dynamic flight path control coupling for energy and maneuvering integrity. In: IEEE interntational conference on control, automation, robotics and vision (ICARCV)
13. Schatz S, Holzapfel F (2014) Modular trajectory/path following controller using nonlinear error dynamics. In: IEEE international conference on aerospace electronics and remote sensing technology (ICARES)
14. Schatz SP et al (2016) Flightplan flight tests for experimental DA42 general aviation aircraft. In: IEEE interntational conference on control, automation, robotics and vision (ICARCV)
15. Schneider V, Mumm N, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2015 IEEE international conference on aerospace electronics and remote sensing technology (ICARES)
16. Mumm, N, Schneider V, Holzapfel F (2015) Nonlinear continous and differentiable 3D trajectory command generation. In: 2015 IEEE international conference on aerospace electronics and remote sensing technology (ICARES)
17. ARINC 424-20: Navigation System Database (2011)
18. RTCA DO 236C: Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation (2013)

nxControl: Ground Mode for Manual Flight Control Laws with Longitudinal Load Factor Command

K. Schreiter, S. Müller, R. Luckner and D. Manzey

1 Introduction

Future air traffic is expected to grow and demanding requirements will intensify the needs for more complex flight trajectories (e.g., Flightpath 2050 [4]). Especially, precision requirements for flight path and speed enable a better utilisation of limited airspace. The precision requirements hold for ground operation as runway capacity is a bottleneck as well. Today, fully automated flight control systems fulfil these requirements. However, pilots should be able to take over manual control at any time. Furthermore, the Federal Aviation Administration (FAA) emphasised the loss of manual flying skills due to excessive use of full automation [5] that could be prevented by more manual flight in daily operations. As the expected requirements on future flight precision will raise pilot workload at manual flight, improved handling characteristics can counteract this trend. Modern cockpits are equipped with fly-by-wire technology and flight control computers. The underlying flight control laws improve handling qualities and stability of manual flight. Demand control laws for sidestick/yoke and pedal inputs are used to achieve precise manual flight at lower workload. But, there is a lack of control laws for manual thrust and spoiler control.

K. Schreiter (✉) · R. Luckner
Department of Flight Mechanics, Flight Control and Aeroelasticity,
Technische Universität Berlin, Berlin, Germany
e-mail: karolin.schreiter@tu-berlin.de

R. Luckner
e-mail: robert.luckner@tu-berlin.de

S. Müller · D. Manzey
Department of Work, Engineering and Organizational Psychology,
Technische Universität Berlin, Berlin, Germany
e-mail: simon.mueller@tu-berlin.de

D. Manzey
e-mail: dietrich.manzey@tu-berlin.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_11

The nxControl system introduced by [11, 14] is designed as supplement for the existing manual control laws. It replaces conventional command of fan rotation speed or spoiler deflection and controls thrust and spoilers according to pilot's command of total longitudinal load factor in flight path direction $n_{xk,tot}$. Various designations are used for the total longitudinal load factor: specific excess thrust, potential flight path angle, acceleration potential, change rate of specific energy level, and total energy angle, see [2, 3, 6, 7]. To simplify the representation, $n_{xk,tot}$ will be abbreviated longitudinal load factor n_x. As derived by the longitudinal force equation of aircraft motion, $n_{xk,tot}$ is directly influenced by thrust force F and drag force D related to aircraft weight W and wind incidence angle α_W:

$$n_{xk,tot} = \frac{F - D}{W} + \sin \alpha_W \quad = \frac{\dot{V}_K}{g} + \sin \gamma \quad = \sin \gamma_E \ . \tag{1}$$

These external forces influence speed and altitude changes ($\dot{V}_K$ and γ) and therefore the change of kinetic and potential energy. The load factor n_x therefore equals the specific change of total energy represented by the total energy angle γ_E. nxControl uses these relations to control thrust via engines and drag via spoilers allocated with a hierarchical logic, as described in Sect. 3. It enables pilots to directly control the total energy state change of the aircraft instead of using the variables pitch and fan rotation speed as a proxy. This assures more precise manual flight with lower workload despite expected higher future requirements.

The impact of nxControl was tested in two evaluation studies described in [11, 13, 14] that covered three different flight scenario types: Single standard flight state changes, a standard approach with instrumented landing system (ILS), and a demanding steep approach with required navigation performance (RNP). After a short training phase with nxControl, the pilots were able to use the new system to achieve the same flight precision in standard scenarios as with their well-trained conventional manual controls but with a lower lever activity and therefore lower workload. Moreover, the results of the demanding manual RNP approaches showed significantly better precision with respect to speed control and a significant reduction of subjective physical pilot workload. As flight path precision is already supported by normal load factor control (n_z-control with sidestick), nxControl did not affect the accuracy of altitude and flight path angle.

The next step was to supplement the nxControl control laws with a ground mode to support manual control in following phases: landing (including touch-and-go), taxi, and take-off (including rejected take-off). There are existing systems that support pilots during the time critical phases of take-off and landing (e.g. autobrake). However, using the conventional control on ground in combination with nxControl in flight would require a mode change at the transitions of flight and ground motion. Such mode changes from augmented to direct controls, and therefore changes in aircraft behaviour, could disrupt the normal procedure of the pilots and may lead to a lack of situation awareness. The ground mode of nxControl therefore enables continuity in augmented manual control of thrust and spoilers. Additionally, it supports the control of wheel brakes and thrust reverser that mainly affect the longitudinal load factor at landing. With the ability to precisely control the deceleration/acceleration $\dot{V}$, it is possible to precisely influence the used runway length s. With the kinematic equations

$$s = \frac{\dot{V}}{2} t^2 + V_0\, t + s_0 \ , \ \ V = \dot{V}\, t + V_0 \ , \tag{2}$$

the used runway length while landing can be calculated with initial values for runway distance s_0 and speed V_0 at touch-down, taxi speed V_T and deceleration $\dot{V}$:

$$s = \frac{V_T^2 - V_0^2}{2\,\dot{V}} + s_0 \ \ . \tag{3}$$

Note that this equation is simplified with the assumption of constant mean deceleration.

The existing autobrake systems support the landing phase by controlling wheel brakes. The systems provide various fixed deceleration levels that the pilots can select, e.g., three levels at Airbus A320 aircraft [16]. This corresponds to three different landing distances. However, these landing distances do not necessarily relate to the distance between touchdown point and runway exits. Additionally, the touch down point varies for every landing. To adjust the effects of the autobrake system and to reach the appropriate runway exit, the pilots need to take over manual braking at a certain point. However, pilots neither have sufficient nor effective information about the impact of manual braking and thrust reverser. Airbus has implemented an autobrake mode, called “Brake to Vacate” into Airbus A380. This system allows for choosing a specific runway exit and the system automatically controls deceleration with wheel brakes. If the pilots use thrust reverser, the autobrake system adjusts brake force to maintain the given deceleration. However, the impact of thrust reverser is not transparent for the pilots.

The command control system for longitudinal load factor nxControl allows both precise setting and flexible adjustment of deceleration rate while getting a feedback on its impact. This enables pilots to reach predefined targets while it permits corrections if unexpected situations appear. In this way, nxControl allows a strategic use of runway length and occupation time with the pilot in-the-loop.

The following sections give an overview of the status quo on ground operations, examine preceding considerations of pilots to nxControl ground mode, and describe the controller design as well as the human-machine interface (HMI). Additionally, a feasibility study with airline pilots was conducted to test the control law functions in different situations of landing, taxi and take-off. The results show pilots’ control strategy with nxControl as well as their assessment of the nxControl system.

2 Fundamentals on Ground Operations

The nxControl design process is pilot-centred, that means the technical and procedural aspects as well as requirements of the future users are simultaneously considered. The aim of this approach is to address all relevant issues of the given human-machine-system. While designing the nxControl ground mode, ground operations as well as transitions between flight and ground had to be taken into account. Therefore, recent procedures for take-off, landing, and taxi were analysed and enriched by preceding considerations of pilots about nxControl on ground.

2.1 Typical Procedures on Ground and Transition Phases

An overview of the flight procedures is provided as collected from discussions with pilots participating in previous studies, flight crew operation manuals (FCOM), e.g., [1], as well as state-of-the-art literature, e.g., [10].

Final Approach At least 1 000 feet above ground, a stabilised flight state is required to prepare a safe landing. The aircraft must be in a steady decent on glide path (trimmed and required thrust set) with wings level and landing configuration (flaps full and gear extended). The pilots have to prepare landing by arming the spoilers for lift dump and setting the autobrake system mode. The choice of brake mode depends on runway and weather conditions and can be set to off (manual brake only), or levels between low and maximum. However, the maximum level is not recommended for landing. No autobrake may be used if the runway is long and dry as well as if the runway is very short (full manual brake required). For short and contaminated runways or at poor visual conditions, medium autobrake levels are recommended. Autobrake low is used for good weather and runway conditions.

At decision height, the pilots decide whether the landing is continued or a go-around procedure is initiated. When landing is continued, the flare phase starts shortly above ground (20–30 feet) and the pilots set the engines to idle.

Landing After touch-down the spoilers are automatically fully extended as lift dump if they were armed before. The brakes are controlled depending on the preset mode. If the autobrake system is armed, constant default deceleration rates are adjusted by the system corresponding to the level. Autobrake is disengaged if the pilots take control by manual braking. Furthermore, autobrake is turned off automatically at taxi speed. If manual braking was chosen, the pilots change incidence angle of the pedals. In addition to wheel brakes, pilots can use thrust reverser that is most effective at higher speeds. As idle forward thrust impede deceleration, idle thrust reverse is recommended but it is not mandatory. Thrust reverser are recommended to use until a minimum speed of 70 knots to prevent the suction of exhaust gas by the engines but are not limited to this speed. The pilots decelerate the aircraft until it reaches taxi speed. Spoilers are retracted at this speed.

Go-Around If pilots decide to abort landing, the go-around procedure is initiated. In first place, engines must be set to maximum (TO/GA) and the pitch angle must be increased gradually to approx. 15°. To reduce drag, flaps are retracted one step. After stabilising, a second approach is initiated at the same or an alternative airport.

Taxi After landing or before take-off, the aircraft rolls along taxi ways. The maximum taxi speed is 20–25 knots. To start taxiing, pilots set thrust slightly above idle and release wheel brakes. When taxi speed is reached, thrust is set to idle and adjusted afterwards if necessary. If the idle thrust is greater than rolling friction, pilots have to maintain speed by sequenced or continuous braking. Deceleration is initiated by setting thrust to idle and pushing the pedal brakes. Before turns, pilots set thrust to idle and reduce speed with brakes. Pilots keep brakes released and thrust in idle at the turn as long as speed is not varying to much. Otherwise, corrections of thrust and brakes are necessary.

Take-Off Pilots configure the aircraft before take-off (pitch trim, flaps). Spoilers and autobrake mode maximum are armed to be prepared for take-off rejection. Then, thrust levers are set to maximum or flexible take-off position (TO/GA or FLX) and pedal brakes are released for acceleration. Until the aircraft reaches decision speed V_1, the pilots can decide whether take-off is continued or rejected. Above V_1, the take-off must be continued. At rotation speed V_R the aircraft is ready to rotate and lifts off with a climb rate set by commanding a recommended pitch angle. After take-off, the climb phase follows, where thrust, gear, and flaps setting is adjusted.

Rejected Take-Off When a pilot decides to abort a take-off, the aircraft must be decelerated as fast as possible. The engines are set to idle and thrust reverse is activated. The autobrake system provides maximum braking force. If the braking force of the autobrake system is not sufficient, pilots shall brake manually. Therefore, deceleration during a rejected take-off procedure is similar to landing on a short runway.

2.2 Pilot Considerations Regarding nxControl Ground Mode

The eleven airline pilots, who participated in the first flight simulator study for testing the concept and the preliminary nxControl design in flight mode [11–13], were interviewed about advantages and disadvantages of extending nxControl with a ground mode. The new concept was explained for a landing procedure as example. The answers are summarised below and categorised into three topics: Overall concept, controller, and human-machine interface aspects.

The participants mentioned that the ground mode concept is a consistent extension of the flight mode. The control of the aircraft's reaction and the command of a precise value for total acceleration would be more useful than a direct control of the individual aircraft components. The control of deceleration could enable the selection of a requested runway exit by means of an optimized use of brake and thrust force.

However, the pilots were sceptical about the interpretability of the energy angle in reference to the braking distance and recommended a predictor for a selected runway exit or the end of runway. Some participants were concerned about the complexity of the system that could reduce the understanding and direct influence of the pilots. Also, one participant mentioned that existing automatic systems supporting landing procedure might be redundant with the new system.

As advantage of the demand controller, the pilots mentioned the potential of optimal allocation of the control elements, e.g., the relation of wheel brakes and thrust reverser. Additionally, the wheel brakes could be used more balanced than in the conventional manual way that could prevent brake wear. Compared to the existing autobrake system, the demand controller could also offer a more variable control of the ground velocity after touchdown without deactivation of the automation. Though, the participants showed concerns about the situation awareness, e.g., in case of failures. The detection of subsystem failures could be delayed. Also, the priority of the controlled elements should be clearly defined and authority to activate the thrust reverser should rest with the pilots.

The concept of replacing several manual control inputs by one control lever was perceived ambivalently. On the one hand, controlling just one input device with one command value could ease operations and therefore lower workload. On the other hand, the participants were worried about their situation awareness regarding which actuation element is used. Additionally, the visual head-down feedback at the displays might be difficult to use in the dynamic situation of landing and therefore interfere with a precise command.

3 Functions and Architecture of the nxControl Ground Mode

The controller architecture and the benefits of nxControl flight mode were described in [14]. Recapped, the control and command variable is the longitudinal load factor defined in Eq. 1. Instead of flight path speed (inertial speed) the calibrated airspeed V_{CAS} is primarily used by the control laws, as pilots control energy state by V_{CAS}. However, to damp turbulent atmospheric influences, inertial speed and airspeed are complimentary filtered. A hierarchical allocation (daisy chain) of control laws for engines and spoilers (with additional activation button) shall yield best pilot awareness of the controller behaviour. Feedback of current flightpath angle, energy angle and power limits are given on primary and secondary displays to the pilot. Additionally, the inceptor nxLever provides haptic feedback for its neutral position (zero command). The new ground mode controller is harmonised with the flight mode, to prevent transients during mode switching. So, the ground mode is fully integrated into the entire controller structure. The ground mode adopts and extends the flight mode requirements with the considerations described in Sect. 2.2 in a human-centred design. The nxControl law comprises the control allocation logic, the single controllers for thrust, spoilers, wheel brakes, and reverser commands as well as the feedback of load factor n_x. The pilot is the higher-level controller that gives inputs and gets feedback of this control loop via the adapted human-machine interface.

3.1 Control Allocation Logic

Figure 1 shows the control law architecture of the nxController. The control allocation logic distributes the control error signal to the specific control laws. The hierarchical structure of the flight mode is extended with logics that activate and deactivate the

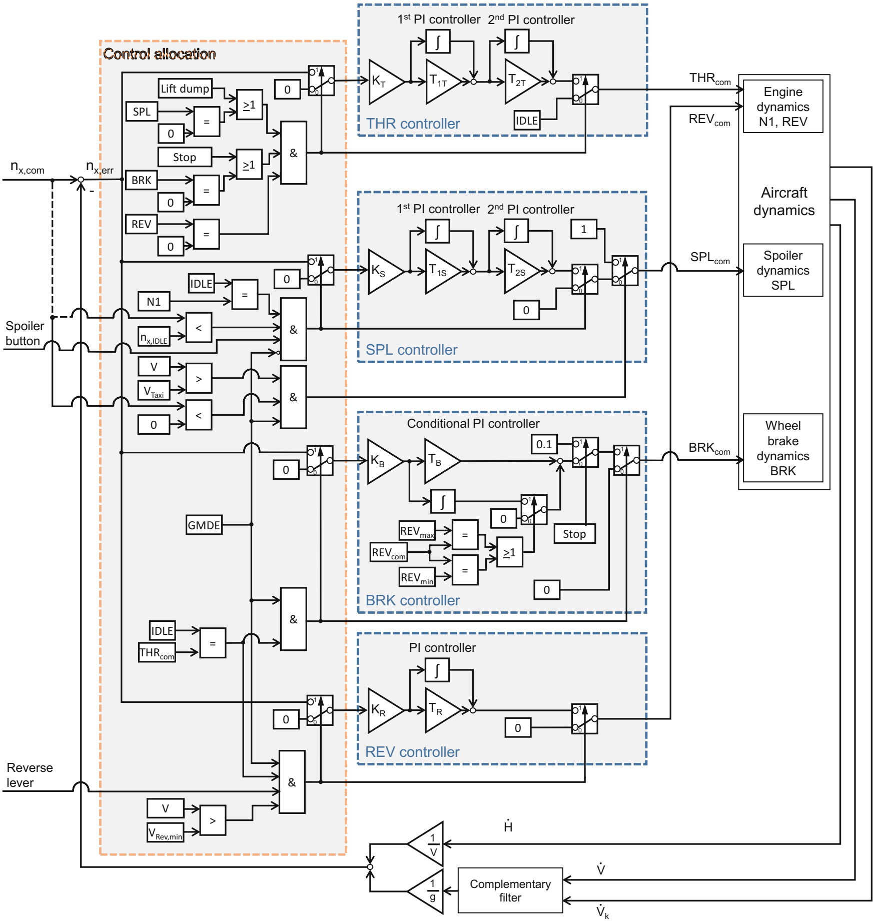


Fig. 1 Control law architecture with control allocation logic and the individual control laws for forward thrust, spoilers, wheel brakes, and thrust reverser

control elements wheel brakes and thrust reverser on ground. As in flight, the thrust controller is active as long as the spoilers (SPL) are fully retracted (except for lift dump), wheel brakes (BRK) are released (except at full stop) and thrust reverser (REV) is deactivated. Spoiler controller is active if fan rotation speed (N1) is commanded to idle and pilots allowed spoiler use by moving the spoiler switch at the nxLever. When the aircraft is on ground (flag ground mode GMDE) and faster than taxi speed, the spoilers are not controlled but fully extended for lift dump if the pilots command deceleration. Otherwise, a positive command represents cancellation of deceleration phase and therefore ground spoilers are retracted. Wheel brakes are controlled if the aircraft is on ground (GMDE) and N1 is commanded to idle. If the aircraft is slower than 0.5 knots and the command value is below zero (stop flag), wheel brakes are activated with a constant deflection to prevent unintended rolling. The use of thrust reverser is only allowed on ground and above the minimum thrust reverser speed. If engines are in idle and the pilots engage the reverser lever at the inceptor, the thrust reverser controller is active.

This allocation logic supports all phases of a turn-around and also all phase transitions, e.g., from flight to landing or take-off to rejected take-off. In addition, the alignment to the given flight procedures allows for mental awareness of the pilots. The separation of the actuation elements also allows for separate controller designs.

3.2 Controller Design

The design of the thrust (THR) and spoiler (SPL) controller is described in [14]. The controllers for wheel brakes and thrust reverser were designed with the same requirements and the same process. The structures of the control laws and the preliminary controller gains were developed with linear models and given requirements (see below). Subsequently, the controller gains were optimised using a nonlinear simulation model and additional requirements for time response of load factor n_x.

The linear models for wheel brake and thrust reverse behaviour were derived from time signals of a highly sophisticated nonlinear simulation. The step responses of the load factor induced by reverser or wheel brake inputs were analysed and modelled by linear transfer functions. A simple but sufficient approximation of the response behaviour is given by first order lag elements for both control elements:

$$F_{n_x,BRK} = \frac{k_{BRK}}{(\tau_{BRK}s + 1)} \,, \tag{4}$$

$$F_{n_x,REV} = \frac{k_{REV}}{(\tau_{REV}s + 1)} \,. \tag{5}$$

The gains k and time lags τ depend on aircraft weight and configuration.

As for the flight mode, the main requirement for the controller design is steady state accuracy. For the systems given by (4) and (5), a PI controller with the controller transfer function F_C

$$F_C = K_C \, \frac{T_C s + 1}{s} \tag{6}$$

is sufficient and leads to the closed loop transfer function F_{CL}

$$F_{CL} = \frac{kK_C}{\tau} \, \frac{T_C s + 1}{s^2 + \underbrace{\frac{1}{\tau}(kK_C T_C + 1)}_{2\zeta\omega_n} s + \underbrace{\frac{kK_C}{\tau}}_{\omega_n^2}} \; . \tag{7}$$

Equation (7) shows that the controller influences both zeroes and poles of the closed control loop. The controller gains can be calculated by setting damping ratio ζ and natural frequency ω_n and therefore the poles. To avoid all-pass behaviour, the controller gain T_C must be positive. Additionally, overshooting actuation signals after step inputs should be avoided. Therefore, the product $|K_C T_C|$ needs to be lower than the reciprocal system gain $1/|k|$. These two demands limit the pole placements, e.g., by minimum and maximum natural frequency depending on damping ratio:

$$\omega_{n,min} \;=\; \frac{1}{2\zeta T_C} < \omega_n < \frac{1}{\zeta T_C} \;=\; \omega_{n,max} \tag{8}$$

These limitations in the frequency domain were the basis for the time response design of the closed loop system. This was performed with the software tool MOPS (Multi-Objective Parameter Synthesis) of the German Aerospace Centre [9]. The described linear model as well as the sophisticated nonlinear model were implemented to generate time responses to predefined step inputs. Requirements for damping ratio, rise time, overshoot, time delay margin, actuation signal damping, and steady state error were defined as so called bad/good criteria. Depending on aircraft configuration and weight, sets of controller gains for wheel brakes and thrust reverser were calculated.

The described controller design regards the control element dynamics separately. As wheel brakes and thrust reverser act in the same direction, a distribution function for the actuation signals is necessary. The distribution function can be freely selected respecting given limitations, e.g., using less reverser to reduce noise emissions. The designed controller architecture shown in Fig. 1 respects two cases, braking on the runway with or without thrust reverser. If the pilots do not activate thrust reverser at the inceptor device, the main actuation elements are the wheel brakes with the given PI controller. If the pilots activate thrust reverser, wheel brakes become supportive elements and the thrust reverser becomes the main control element. Therefore, the integrative part of the wheel brake controller is cut off as the thrust reverse controller takes over the integrative behaviour. If the thrust reverser reaches maximum deflection, the integrator of the wheel brake controller is switched on again to achieve the commanded value by the pilot. Possible noise restrictions can be respected by lower maximum deflections for thrust reverse controller (not depicted). For emergency brake, e.g., at rejected take-off, the brake and reverser controllers are bypassed with full deflection commands (not depicted).

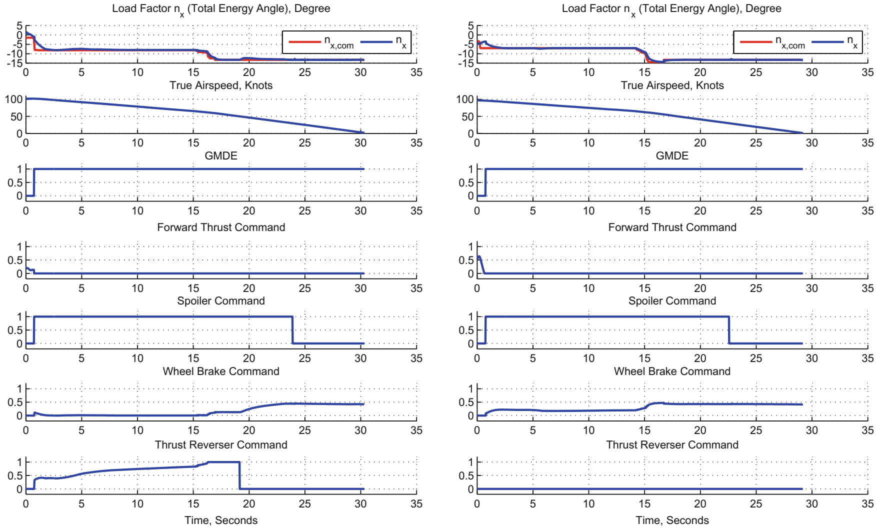


Fig. 2 Longitudinal load factor, speed, and actuation signals by nxController during a simulated manual landing with (*left*) and without (*right*) thrust reverser activation

Figure 2 shows the result of this controller design for two manually flown landing procedures, one with thrust reverser allowed (left) and one without thrust reverser (right). In both cases the initial command value for longitudinal load factor (in the form of total energy angle) of approx. $-7.5°$ after touch down with landing speed of 108 knots was reduced after around 15 s to approx. $-13°$. Independent of wheel brake and thrust reverser, forward thrust command is set to its minimum. Additionally, the spoiler command is set to its maximum as the spoilers are used for lift dump until the aircraft reaches taxi speed of 20 knots. In the left figure, it can be seen that the reverser command gradually rises with time until it reaches the maximum deflection. At this point the wheel brake controller takes over the integrative behaviour and rises the wheel brake command. After around 19 s the thrust reverser is switched off because the aircraft reached 50 knots minimum thrust reverser speed. This disturbance is balanced by the wheel brakes. The right figure shows that the wheel brakes are the only control element and the command for thrust reverser is zero. Both cases show an accurate longitudinal load factor response and therefore a similar speed reduction and duration of the procedure.

3.3 Human-Machine Interface

The human-machine interface includes visual outputs and a manual inceptor. The HMI for the flight mode and its functionality are described in [11, 14]. The visual feedback is given by additional symbols for flight path and total energy angle at

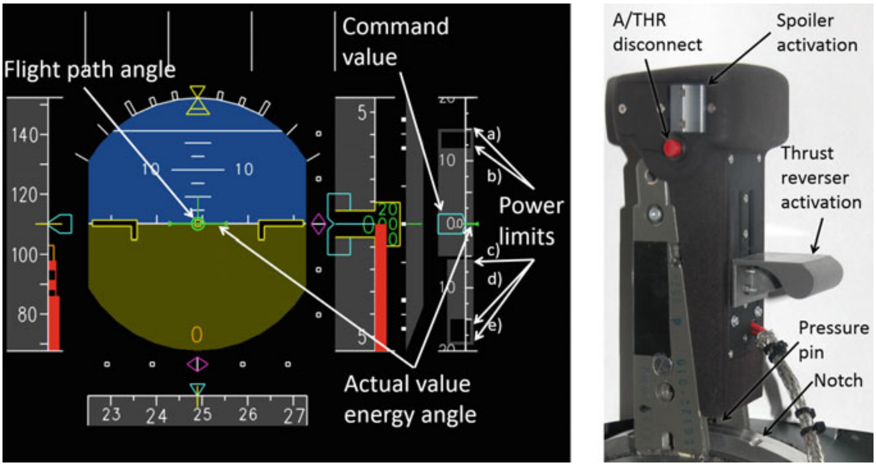


Fig. 3 HMI of the nxControl system; nxPFD: Total energy angle and flight path angle at artificial horizon, vertical degree scale for energy angle, command value and power limits next to vertical speed scale; nxLever: handle with spoiler and thrust reverser switch

the artificial horizon of the primary flight display (nxPFD) as well as an additional vertical degree scale for energy angle, command value and power limits of engines and spoilers at the engine display (nxStatus). This concept is also used for the ground mode with some improvements. Figure 3 shows the new version of the nxPFD at a situation after touch down and with a commanded energy angle of zero degree. The nxStatus scale was moved to the nxPFD at the right side of the vertical speed tape to improve scanning pattern. Also, the colours and symbols were adjusted. A functional change for the ground mode was necessary for the indication of power limitations. As the actuation elements differ on ground compared to flight, new symbols had to be introduced. On ground, power limits for forward thrust, wheel brakes and the combination of wheel brakes and thrust reverser are shown on the nxStatus scale. Both upper limits represent the maximum energy angle with (a) maximum thrust (TO/GA) and (b) flexible thrust (FLX). The first lower limit (c) indicates the energy angle with idle thrust. Commands below this value are achieved with wheel brakes limited by the lower limit of the filled tape (d). If the pilot activates thrust reverser the hollow lower limit (e) can be reached.

The concept for the inceptor of the nxControl flight mode, called nxLever (see Fig. 3), was adopted for the ground mode. One handle is used to command the target value for load factor in form of the energy angle although multiple engines are controlled. By pulling the reverser lever up, the use of reverse thrust by the controller is activated. The pressure pin on the bottom together with a grooved rail provide the mechanism for haptic feedback. The pin snaps in on predefined positions. The notch at the middle position, representing a zero command, is also used in flight mode. Two more notches are provided for typical deceleration rates on ground. One notch lies at the command value of −10°, as it is equal to the autobrake level *low*.

The second notch represents the command value −15° that equals the autobrake level *medium* on some Airbus aircraft. These positions were freely chosen and should be adapted for particular aircraft. Commands below the last notch represent emergency brake mode. In that case, all control elements are fully deflected to minimize energy angle (thrust reverser only after activation).

4 Study with Pilots

An evaluation study was conducted with seven airline pilots in the fixed-base research flight simulator SEPHIR (Simulator for Educational Projects and Highly Innovative Research) at the Chair of Flight Mechanics, Flight Control and Aeroelasticity of Technische Universität Berlin [15]. The simulator is based on a VFW614 ATD and the cockpit is equipped with displays and sidestick (including control laws) similar to an Airbus aircraft.

All pilots were male and in possession of Airbus type ratings (A320: 6, A380: 1). The pilots' age ranged between 25 and 64 years with a mean of 39 years (15 years standard deviation). Their experience varied widely from 330 to 25 000 total flight hours with a mean of 8 440 h (9 146 h standard deviation). Most of the pilots were ranked as First Officer, only one participating pilot was Captain.

The pilots were trained with the nxControl system, which took 1.5 h. Afterwards, the pilots had to fulfil different tasks in different scenarios with the nxControl system. The test scenarios represent the flight phases on ground: landing, taxi and take-off. All tasks were described by target values, but the procedure to fulfil the tasks lay in the hands of the pilots. Thus, the pilot strategy with the new system could be examined. There was no statistical comparison to the conventional control strategy. As part of the pilot-centred design, the study aimed at examining the feasibility of the concept, necessary changes to procedures, and exposing any issues that might have been disregarded in advance. The focus lay on the questions whether the pilots could succeed fulfilling their task, whether the pilots were aware of the system functionality, or whether *show stopper* for the ground mode concept have to be expected and how the pilots asses the system.

4.1 Test Scenarios and Interview Questionnaires

The tests were conducted at Frankfurt/Main airport former runway 25R (now 25C) shown in Fig. 4. The first scenario was a final approach with touch down and deceleration to runway exit *Ato* (*Landing Procedure Long*). In this scenario the use of the thrust reverser was not allowed. The second scenario was another final approach with touchdown and deceleration to the near runway exit *Gto* that required the use of thrust reverser. Before these two scenarios, the pilots were requested to calculate the mean deceleration rate, transformed into a total energy angle, that was necessary for

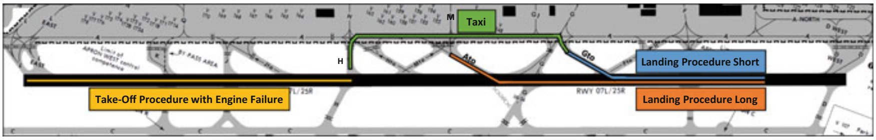


Fig. 4 Section of Frankfurt/Main (EDDF) airport chart [8] with runway 25R

the different runway lengths. The reversed Eq. (3) was provided. The objective was to build up an awareness about load factor values at landing. Both scenarios started on glide slope fully configured and with landing speed of 108 knots calibrated airspeed. After touch down, the pilots had to decelerate to 20 knots taxi speed which led to mean energy angles of $-7°$ in scenario one and $-13°$ in scenario two. The mean values were introduced as reference values but the pilots were free to perform their preferred braking strategies. After deceleration, the pilots had to leave the runway and to stop at the holding point of the runway exits.

The third scenario was taxi on ground from runway exit *Gto* to the new take-off position near runway entrance *H*. The pilots were requested to accelerate to 20 knots taxi speed, stop at the intersection *M*, and continue taxi to the holding point of runway entrance *H*.

Scenario four and five were take-off procedures beginning near runway entrance *H* with engine failures. In scenario four, the pilots had to accelerate with TO/GA thrust setting. The engine failure appeared shortly before 100 knots decision speed V_1 that induced take-off rejection. In scenario five, thrust setting FLX was requested to accelerate and the engine failure appeared after V_1. Therefore, the pilots had to continue take-off procedure with reduced climb performance at 120 knots speed. All scenarios were performed twice and only the second trial was evaluated.

After each scenario, the pilots were asked about their opinion on the nxControl system. The interview covered the topics procedure, steering strategy, HMI, and controller/aircraft behaviour. The topic procedure included questions about potential conflicts with existing procedures as well as safety aspects. Also, the pilots were asked how they used the system and what kind of steering strategy they had in mind. As the HMI changes the standard cockpit layout, the questions addressed the scanning and input behaviour as well. Especially, the use of the added information on the primary flight display as well as the haptic feedback of the nxLever were investigated. The last topic addressed controller logic and aircraft response. The pilots were asked if the aircraft behaviour was as expected, if they could anticipate the actuation element behaviour, and if they could successfully fulfil their task.

4.2 Results and Discussion

The simulation data of commanded total energy angle, resulting calibrated airspeed, state of ground mode (GMDE), activation of thrust reverser by the pilots, and the

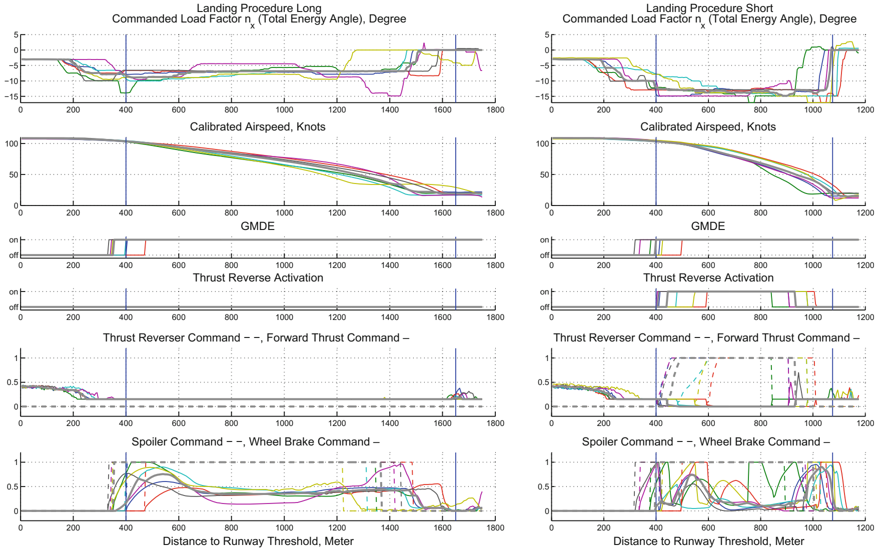


Fig. 5 Longitudinal load factor command, speed, and actuation signals by nxController of all participants and median values (*bold grey line*) against distance from runway threshold at manual landing procedures without (*left*) and with (*right*) thrust reverser activation by the pilot

commands by the nxController to forward and reverse thrust, spoiler, and wheel brakes is presented in the Figs. 5, 6 and 7. The results of all participants are shown in the background and were averaged with the median (bold grey line) that will be the basis of result description.

4.2.1 Landing Procedures

Figure 5 shows the results of both landing scenarios. The ordinates represent the distance from runway threshold. At 400 m distance, the touch down area is located. The runway exits are located at 1 650 and 1 075 m respectively. In both scenarios, the progress can be divided in the phases flare, first deceleration after touch down, main deceleration, capture taxi speed, and turn to runway exit.

In scenario one, flare phase was entered by reducing thrust setting. Therefore, the pilots changed energy angle command from −3° to averaged −7°, which corresponded to the calculated reference value. This caused forward thrust to decrease to idle (represented by the value 0.15). After touch down, the command value was further lowered for the first speed reduction. Consequently, the controller increased the wheel brake command and set the spoilers into lift dump position. At approx. 600 m distance, pilots' command was returned to nearly −7°, which lowered wheel brake command to the appropriate value. This resulted in constant speed reduction until taxi speed. Then, the pilots changed their command value gradually to zero

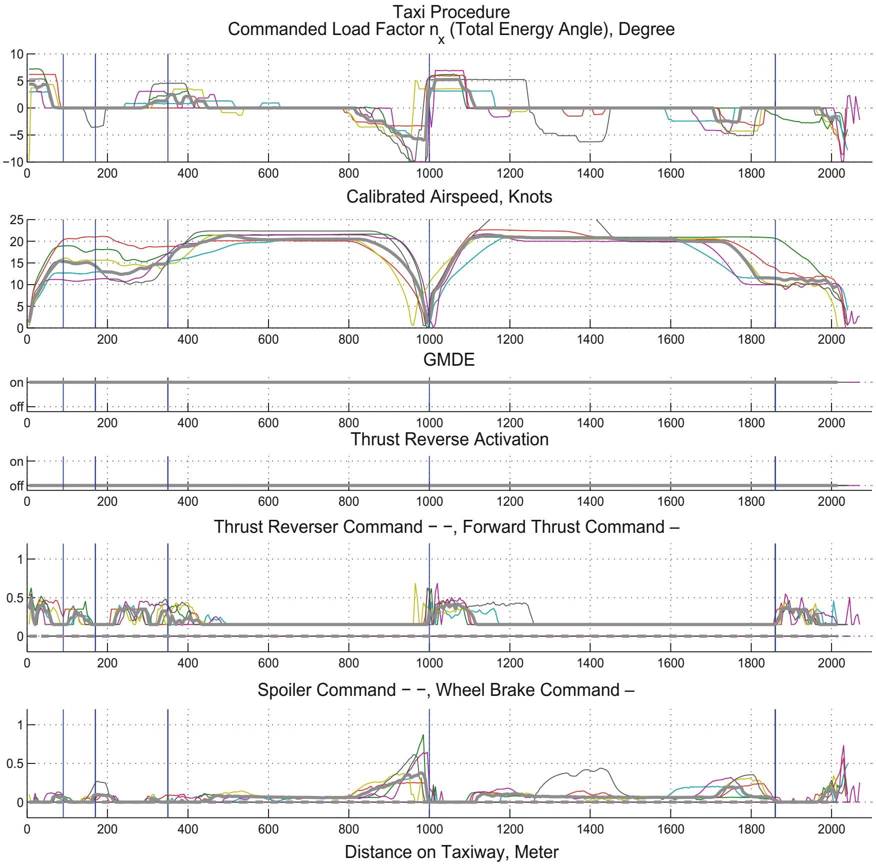


Fig. 6 Longitudinal load factor command, speed, and actuation signals by nxController of all participants and median values (*bold grey line*) against distance on taxiway at taxi procedure

(middle position of the nxLever). This released wheel brakes. As the idle thrust force of the VFW614 is higher than the rolling friction, braking is necessary to maintain speed. At turns, friction is rising and therefore energy angle would decrease. But, the nxController compensated this by rising the thrust setting.

In the second scenario, the flare phase was entered in a similar way, but with a lower command value for the total energy angle. The pilots changed their command from −3° to averaged −10°. This value corresponded to a notch of the nxLever (haptic feedback). After touch down, the pilots averagely commanded the reference value of −13° for first deceleration and allowed thrust reverser by activation on the nxLever. Therefore, the wheel brake command increased until thrust reverse became active. Afterwards, the brake command decreased until thrust reverse reached its maximum and additional amount of braking became necessary. Around 700 m after threshold, the pilots reduced energy angle command under the reference value for a

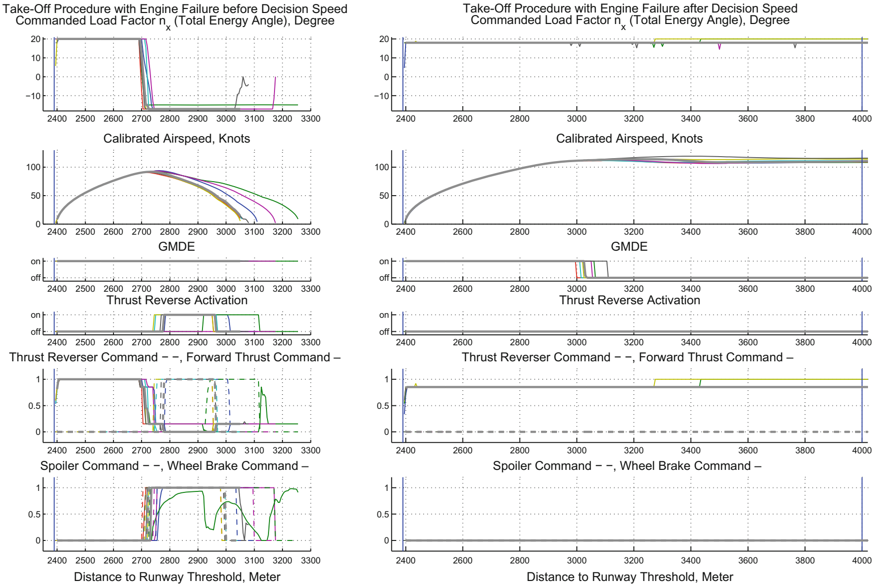


Fig. 7 Longitudinal load factor command, speed, and actuation signals by nxController of all participants and median values (*bold grey line*) against distance from runway threshold at manual take-off procedures with engine failure before (*left*) and after (*right*) decision speed

stronger speed decrease. Note that commands below −15° correspond to emergency brake mode. As speed reached 50 knots, thrust reverser was automatically deactivated and therefore wheel brake command increased again. With the beginning turn to the runway exit, the pilots commanded 0° energy angle to hold taxi speed. Therefore, brakes were released and thrust partially increased to compensate additional friction.

In the interviews, the pilots mostly recalled the steering strategies represented by the median behaviours. Four of the seven pilots stated that they used the calculated reference energy angle in favour of constant deceleration. Due to safety considerations, two pilots chose a higher deceleration rate at first and then reduced the deceleration to the reference value. One pilot's strategy was to keep runway occupation time short with a low deceleration after touch-down and increased it closer to the runway exit. Some pilots mentioned that they used emergency brake position by mistake as they tended to use the lever's end stop position while decelerating. To avoid accidentally selecting emergency brake, a harder notch prior to emergency brake or secondary inputs commanding emergency brake should be considered.

Concerning the handling of nxControl, a few pilots mentioned the extended head-down time to select a specific energy angle with nxStatus scale. In their opinion, this was very unusual and the priority in this phase should mostly remain on the outside view. However, one pilot supposed that the lengthened head-down time could be caused by missing deceleration clues in the fixed-base simulator.

Despite criticism that braking with hands is unusual, there were also comments on its advantage: The pedals would have one function and the pilots could concentrate on lateral control.

The pilots were also asked, if nxControl made these tasks more difficult or easier. For scenario one, six pilots answered with "slightly easier" or "easier", one pilot opted for "unchanged". For scenario two, five pilots stated "slightly easier" and two pilots stated "slighly more difficult".

4.2.2 Taxi on Ground

Figure 6 shows the results of taxi scenario. The ordinate represents the distance covered on taxiway. Between 90 and 170 m and between 170 and 350 m, a left turn followed by a right turn led to the main taxiway. At 1 000 m distance, intersection *M* was located. The turn to runway entrance *H* began at 1 850 m.

For acceleration at the beginning of the scenario and after the full stop at 1 000 m, the pilots commanded a positive total energy angle (averaged 4–5°, maximum 8°). Consequently, engine command for forward thrust rose above idle level and brakes were released. To overcome static friction, the initial thrust command was higher than necessary for taxiing. Before the first two turns, the pilots accelerated to approx. 13 knots and selected the zero command afterwards to maintain speed while changing the direction. As already seen at the turns after landing (Sect. 4.2.1), the additional friction in a turn is compensated by additional thrust. The following acceleration to taxi speed was again achieved by positive energy angle command – but with a lower value (averaged approx. 1.5°, maximum 3°) – which induced lower brake commands and higher thrust commands. At the straight and steady segments after acceleration, the pilots commanded 0° energy angle to maintain speed. Therefore, brake commands increased to compensate idle thrust force. The full stop at 1 000 m was initiated with a negative energy angle command. In average, the pilots chose $-5°$ and in maximum $-10°$ leading to an incrementally rising brake command until full stop was reached. The same sequence appeared at deceleration before the turn to runway entrance *H*, but with a lower absolute command value (averaged approx. $-3°$, minimum $-5°$). The pilots reduced taxi speed to approx. 10 knots for the following turn.

The interviews revealed two main strategies while taxiing. One group determined and used reasonable reference values to accelerate and decelerate, e.g., ±5°. The other group did not utilize specific energy angle values. They used engine noise and lever position as orientation. Most pilots approved and made use of the middle notch (0° command) due to tactile feedback to comfortably hold taxi speed. However, some noted that it was unusual at first to taxi with lever not in idle position.

A few pilots disliked the high fan rotation speeds when starting to roll. Moreover, almost every pilot was initially surprised that thrust was increased in a turn to compensate for additional friction. This controller behaviour was rated differently by the participating pilots. Some of them thought that this functionality is unnecessary or disturbing. Especially the A380 pilot stated that this could be a safety issue when ground vehicles were in close proximity. Other pilots appreciated this behaviour in respect to holding taxi speed in turns and regarding the consistency of nxControl's overall concept. To resolve this issue, the commanded thrust could be restricted to moderate limits at low speeds used while taxiing. The disunity among pilot's feedback was mirrored when asked if nxControl made this taxi task easier or more difficult. Three pilots chose "slightly more difficult" or "more difficult", one pilot chose "unchanged", and three pilots chose "slightly easier".

As additional comment, it was noted that in some cases of taxi differential thrust or brake is used, e.g., in turns on slippery runways. This could be addressed by an additional function of lateral control laws that would balance thrust and brakes.

4.2.3 Take-Off Procedure

Figure 7 shows the results of both take-off scenarios. The ordinate represents the distance from runway threshold. Both scenarios started at 2 400 m. Engine failures appeared in both scenarios at 90 knots (around 2 700 m) and 105 knots (around 2 800 m) respectively. At 4 000 m distance, the end of runway is located.

The first take-off scenario was initiated by commanding the total energy angle to TO/GA setting (depicted as 20°). This caused full-thrust command for acceleration. After engine failure before V_1, the take-off was rejected by retarding the energy angle command to the emergency brake position represented by values below $-15°$ (only one participant did not use the emergency mode). Most participants (except one) then added thrust reverser to safely decelerate and to compensate the yawing moment of the failed engine. Consequently, the command for forward thrust immediately decreased to idle followed by increasing commands for thrust reverse, brakes, and ground spoilers to their full position. By reaching 50 knots speed, thrust reverse was automatically deactivated.

The beginning of the second take-off scenario was similar to the first. The commanded energy angle was at FLX position depicted by the value 18°. Therefore, the command for engines increased to 85%. With the engine failure, the acceleration rate decreased. As the failure appeared at rotation speed, this emerged to a decreased climb rate compared to the nominal case (not shown in the figure). After lift-off, the most pilots chose to maintain the FLX setting and stabilised climb with steady speed. Some participants added more thrust by setting TO/GA.

The majority of pilots stated at the interview that the take-off procedure using nxControl corresponded closely to conventional thrust control. Furthermore the deceleration phase while rejected take-off was very similar to conventional procedure. The pilots steering strategy was to push the lever in TO/GA or FLX position and pull it all the way back and enable thrust reverser to decelerate. In the case of engine failure after V_1, three pilots mentioned that they consciously used flight path angle and/or energy angle to stabilise their flight and air speed. The pilots, which did not command emergency brake or thrust reverser, stated that it happened accidentally. This may be caused by their lack of practice since some company policies only allow captains to taxi and perform take-off decisions.

Two minor conflicts with current procedures were reported. First, regarding the rejected take-off procedure, thrust reverse should be used until complete stop. This can be resolved by an adjusted controller logic. Second, it is not possible to increase fan rotation speed with brakes applied to check for synchronisation of all engines before take-off. It could be complied with this procedure by using parking brake.

Concerning the usability, two pilots thought that they could perform the first take-off scenario “easier” or “slightly easier” with nxControl, while the other five pilots answered “unchanged”. For the second take-off scenario, four pilots stated “slightly easier” and three pilots chose “unchanged”.

5 Conclusion

To maintain manual flight in future air traffic, supporting systems that allow commanding demand values instead of control surface deflections will become necessary. Today’s cockpits provide such functions spawned by fly-by-wire technology. The nxControl system supplements these functions with a demand control law and display elements for the longitudinal load factor represented by the total energy angle. In flight, the system controls thrust and spoiler deflection according to pilot’s commands and aims at supporting the manual energy management. It relieves the pilot of controlling the parameters fan rotation speed or lever deflection as proxy for energy change by means of direct control of the relevant physical parameters.

After validation of the nxControl concept during flight in a previous study, the concept was extended for operation on ground to prevent mode changes at the time critical phases of take-off and landing as well as to enable a precise but still flexible control of take-off and landing distances. Therefore, the control logic of the demand controller was supplemented by the actuation elements wheel brakes and thrust reverser. The control philosophy and the control law design process was consistently applied to the new ground mode. In addition, the HMI was improved considering the results of the previous flight mode study as well as the concept extensions for ground mode.

In a study with seven airline pilots in a research flight simulator, the presented ground concept was tested at landing, taxi, and take-off. The results showed the feasibility of the new system. The pilots used the system as it was intended in all given scenarios after a limited training phase. At two landing scenarios, the pilots reached the near and far runway exits at taxi speed with goal oriented inputs. The use of the command value as well as the thrust reverser activation was not rated as a problem. Taxi on ground was also feasible with the system despite the use of inceptor and actuation elements slightly differ from the conventional case, e.g., rising thrust at turns with zero command, taxi with lever in middle position instead of the rear position, and braking with the lever instead of the pedals. The behaviour of the participants at a rejected or continued take-off after engine failure was similar to the standard procedure. Therefore, nxControl did not affect the conventional behaviour of the pilots in this very fast take-off situation and would not lead to loss of basic skills. Moreover, some participants mentioned the good stabilisation opportunity after take-off with engine failure. As disadvantage, the pilots mentioned a longer head-down time to set the appropriate command that may have been reinforced by the missing physical loads of the fixed-base simulator. Also, some procedural details ought to be adjusted when using nxControl, e.g., using parking brake until initial engine synchronisation at take-off or introducing energy angle values as reference in briefing material.

Despite some improvement opportunities, the study showed that the extended nxControl system can be used on ground and during transition phases. Therefore, the system can provide support of manual control in all phases of an aircraft turn-around. Mode changes at the time critical transitions could therefore be avoided. In addition, the direct manual control of the aircraft reaction is possible and could provide more precise control with lower workload at future demanding trajectories.

Acknowledgements This work is funded by the DFG (German Research Foundation) under contract LU 1397/3-2, MA 3759/3-2. The authors thank the pilots who participated in the described studies as well as I. Karakaya, S. Lademann, M. Schaumburg, and S. Sellmann for their support.

References

1. Airbus Industries (2006) A330 flight crew operating manual – 1 system description (Rev. 20), 2 flight preparation (Rev. 19), 3 flight operations (Rev. 19). Toulouse
2. Brockhaus R, Alles W, Luckner R (2011) Flugregelung, 3rd edn. Springer, Berlin
3. Deutsches Institut für Normen (1990) DIN 9300 Teil 1: Begriffe, Größen und Formelzeichen der Flugmechanik
4. EC (European Commission) (2011) Flightpath 2050, Europe's vision for aviation. Report of the High Level Group on Aviation Research
5. FAA (Federal Aviation Administration) (2013) Safety alert for operators, manual flight operations, SAFO 13002. http://www.faa.gov/other_visit/aviation_industry/airline_operators/airline_safety/safo/all_safos/media/2013/SAFO13002.pdf
6. Filippone A (2012) Advanced aircraft flight performance. Cambridge University Press
7. International Standard Organisation (ISO) (1988) Flight dynamics - concepts, quantities and symbols - Part 1: Aircraft motion relative to the air, ISO1151-1
8. Jeppesen Sanderson, JeppView 3.5.2.0 (2006) Airport chart eddf frankfurt/main, germany. http://eaip.austrocontrol.at/lo/150430/Charts/LOWS/LO_AD_2_LOWS_24-6-6_en.pdf. 27 October 2006
9. Joos H, Bals J, Looye G, Schnepper K, Varga A (2002) A multi-objective optimisation-based software environment for control systems design. In: IEEE international symposium on computer aided control system design, pp 7–14 (2002)
10. Mensen H (2013) Handbuch der Luftfahrt, 2nd edn. Springer, Berlin. https://doi.org/10.1007/978-3-642-34402-2
11. Müller S, Schreiter K, Manzey D, Luckner R (2016) nxControl instead of pitch-and-power: a concept for enhanced manual flight control. CEAS Aeronaut J 107–119 (2016). https://doi.org/10.1007/s13272-015-0169-9
12. Schreiter K, Müller S, Luckner R, Manzey D (2014) nxControl instead of pitch-and-power: concept and first results of a control system for manual flight. Deutscher Luft- und Raumfahrtkongress, Augsburg 2014. Urn:nbn:de:101:1-201507202447
13. Schreiter K, Müller S, Luckner R, Manzey D (2016) Verbesserung von Flugpräzision und Arbeitsbeanspruchung bei manuellen RNP-Anflügen durch Vorgaberegler und Anzeigen für den Energiewinkel (nxControl). Deutscher Luft- und Raumfahrtkongress 2016, Braunschweig

14. Schreiter K, Müller S, Luckner R, Manzey D (2017) Enhancing manual flight precision and reducing pilot workload by using a new manual control augmentation system for energy angle. In: AIAA guidance, navigation, and control conference, AIAA science and technology forum and exposition 2017, Grapevine, Texas
15. TU Berlin, Department of Flight Mechanics, Control and Aeroelasticity: Sephir - simulator for educational projects and highly innovative research. http://www.fmra.tu-berlin.de/menue/forschung/ausstattung/sephir/
16. Villaumé F (2009) Brake-to-Vacate system. FAST 44 – flight, airworthiness, support, technology 44:17–25. ISSN 1293-5476

Helicopter Pilot Model for Pitch Attitude Tracking Task

Milan Vrdoljak, Franz Viertler, Manfred Hajek and Matthias Heller

1 Introduction

Aircraft and rotorcraft present examples of complex man-machine-interface what has been recognized from the development of the flight control system and since then a numerous descriptions of this interface were dedicated to the human part, a mathematical description of the pilot. With rising demand on improvement of flight safety and accuracy of manual pilot tasks for modern highly augmented aircraft or rotorcraft, development of flight control system requires consideration of the whole pilot-vehicle system (PVS) including a reliable description of the human part. Pilot can be seen as a feedback loop closure element of the PVS, strongly influencing the dynamic properties of the whole PVS.

A great number of human sensing mechanisms available for the detection of the aircraft states and their perturbations, with psychological elements such as stress, motivation or fear, and along with a human ability to quickly adapt and learn in

M. Vrdoljak (✉)
Faculty of Mechanical Engineering and Naval Architecture, Department of Aeronautical Engineering, University of Zagreb, I. Lucica 5, 10002 Zagreb, Croatia
e-mail: milan.vrdoljak@fsb.hr

F. Viertler · M. Hajek
Department of Mechanical Engineering, Institute of Helicopter Technology (HT), Technical University of Munich (TUM), Boltzmannstr. 15, 85748 Garching, Germany
e-mail: viertler@tum.de

M. Hajek
e-mail: hajek@tum.de

M. Heller
Department of Mechanical Engineering, Institute of Flight System Dynamics (FSD), Technical University of Munich (TUM), Boltzmannstr. 15, 85748 Garching, Germany
e-mail: matthias.heller@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_12

variable conditions, makes the task of pilot behavior modeling a difficult task. Furthermore, an individual pilot characteristics and specificity of the required flight task on given flight dynamics of particular aircraft/rotorcraft puts additional demand on pilot modeling techniques. Due to these reasons definition of a general model of the pilot is not available, yet they are usually restricted to a specific task applicable only for a particular aircraft/rotorcraft.

Applications of the pilot model in general would be for the task of design of optimal PVS capable of achieving a required level of flying qualities (FQ) and handling qualities (HQ). Specifically it could be applied for pilot's display design, flight control system design, improvement of agreements between simulator-based and flight-based estimation of HQ, PVS HQ assessment, analysis of aircraft's pilot coupling phenomena (PIO) and for the development of criteria for FQ and HQ.

In this paper emphasis is on the application of classical control theory for development of helicopter pilot model.

1.1 Subject Overview

In classical control theory approach a pilot model is described with transfer functions as a dynamic control element, a more or less complex block in the PVS. The most widely used pilot model is the McRuer's crossover model which described PVS as an integrator around the crossover frequency [24, 25]. The idea is that the pilot will adapt to the aircraft dynamics and the PVS dynamics will remain unchanged, at least in the crossover frequency range. Values of the model parameters depend on the aircraft dynamics, but also on the task. Examples of helicopter application of crossover pilot model with time delay for hover task are given for multi-loop case in [4, 26] and for single-loop case in [34].

A second group of pilot models, proposed by Hess [10, 11], can be described as a structural model with a proprioceptive feedback. This sensory model depends on the adaptability of the pilot to the altering flight dynamics around the crossover frequency. Experimental data are required for the adaptation of this model. Limitations of Hess' structural model [5–7] are: resonant peek of modeled closed loop PVS is considerably lower in high frequency range; assumption of constant crossover frequency for different dynamic configurations; visual model described only with gain coefficient. Structural model has been applied for the description of helicopter pilot model by Hess for cross-coupling [20] and near-Earth flight [17] based on the pilot-in-the-loop simulations and flight tests.

There are different rotorcraft applications of structural model: for assessment of handling qualities [12, 15]; for the design and evaluation of cockpit display for hovering task [18]; for the analysis of simulator fidelity assessment [19, 35]; for flight task of lateral reposition of the helicopter with a slung load [8, 21] with correlated multi-loop pilot model (inner loop with roll angle feedback and outer loop with lateral position feedback). Additional analysis of multi-loop of helicopter in hover task is presented in [13–15]. Multi-axis pilot model for pitch and roll is used in analysis of the ability of the refined adaptive pilot model to accommodate significant and sudden variations in the elements of the flight control system [16].

Another application of structural model for helicopter in hover based on linearized helicopter flight dynamics with stabilization feedback is presented in [29]. Pilot model was implemented for four control channels (longitudinal and lateral cyclic, collective and collective for the tail rotor), although they have been defined as single axis models, one independent from another. Flight task given in [29] is preservation of rotorcraft's constant attitude with zero vertical velocity in hover in single axis turbulent atmosphere.

Vorst [32] also used structured helicopter pilot model for description of ADS–33 maneuvers [2]. A separate pilot model was made for each control channel, without evidence of their interaction. A time delay block from the structural pilot model used in this study was removed, due to the stability problems.

A comprehensive review of pilot modeling can be found in [23, 33].

1.2 Efremov's Modification of the Structural Model

A modification of Hess' structural model (Fig. 1) was proposed by Efremov [5, 7] introducing more complex visual system model, a nonlinear pilot's remnant, modifications of inner loop and adjustment rules. This quasi-linear modified structural model provides better agreement with experimental results in comparison to original structural model, especially at higher frequencies for aircraft configurations.

Focus of this research will be on the application of the Efremov's pilot model, that has been proven for aircraft configuration [5, 7, 30], for the helicopter pilot modeling. This pilot model has a defined and easy to interpret structure with emphasis on display system while the vestibular system less important. A flight simulator with high fidelity visual system and a fixed base presents a good set-up for the experiments with emphasis on visual model.

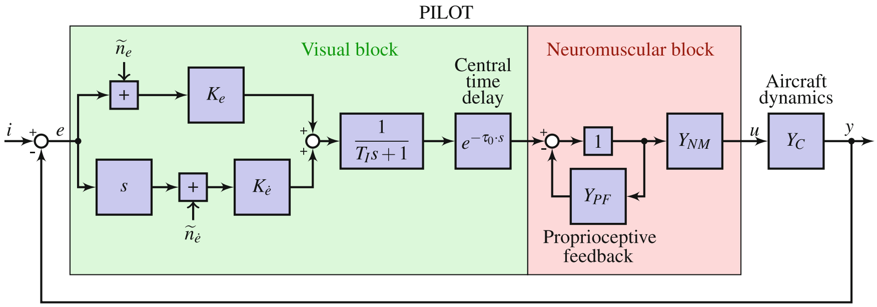


Fig. 1 PVS model structure of the Efremov's modification of the structural model

2 Analytical Analysis

In the analytical analysis of the pilot model a linearized helicopter flight dynamics model of the helicopter was applied in the off-line PVS simulations.

2.1 Helicopter Model and Flight Task

The rotorcraft type chosen for the analysis is a BO-105 helicopter, a twin-turbine light weight utility helicopter. A flight task chosen for this analysis is a pitch tracking in forward flight. The helicopter model has a high level of coupling between the longitudinal and lateral motion what complicates the pilot model. A simple tracking task for pitch attitude is chosen for the case of forward speed where the coupling is of less significance. Analysis is limited to longitudinal motion only and longitudinal command pilot model presents a single channel and single loop problem.

The flight model for BO-105 is available in literature [9, 27, 28] and for this work a reference is a linearized model of the 8th order with coupled longitudinal and lateral dynamics from [28]. In this analysis a manipulator system was not considered, and since the helicopter BO-105 doesn't have a stability augmentation system, the vehicle system is presented only with flight dynamics model of interest.

Flight task of interest is a pitch tracking at forward flight with airspeed of 80 kt, specifically it concentrates on a relation of the pitch attitude of the rotorcraft in reference to the longitudinal cyclic control for the given pitch command. For this flight task a reduced model from [28] with decoupled dynamics is applied describing just longitudinal rotorcraft dynamics at forward speed of 80 kt

$$Y_{\theta_{1s}}^{\Theta}(s) = \frac{46.82s^2 + 39.33s + 1.034}{s^4 + 4.854s^3 + 2.609s^2 + 0.3468s + 0.2426}\,, \tag{1}$$

what presents a transfer function from longitudinal cyclic θ_{1s} to the pitch angle Θ.

2.2 Pilot Model Structure

For the structure of the pilot model an Efremov's pilot-vehicle-system model [5, 7] is used (Fig. 1). Elements of the given structure have defined transfer functions as following.

Visual system has transfer function

$$Y_{vis}(s) = K_p \cdot \frac{T_L s + 1}{T_I s + 1}\,, \tag{2}$$

with lead time constant $T_L = \frac{K_{\dot{e}}}{K_e}$ and visual block gain $K_P = K_e$. Neuromuscular block is described as

$$Y_{NM}(s) = \frac{\omega_{NM}^2}{s^2 + 2\zeta_{NM}\omega_{NM}s + \omega_{NM}^2}\,, \tag{3}$$

with given values of $\zeta_{NM} = 0.1$ and $\omega_{NM} = 12$ rad/s according to [5]. Proprioceptive feedback is

$$Y_{PF}(s) = K_{PF}\frac{s^2}{T_{PF}^2 s^2 + 2\zeta_{PF}T_{PF}s + 1}\,. \tag{4}$$

Central time delay, with $\tau_0 = 0.2$ s, is described with Padé approximation, 4th order [30]

$$Y_{t_0}(s) = e^{-\tau_0 \cdot s} = \frac{s^4 - 100s^3 + 4500s^2 - 1.05 \cdot 10^5 s + 1.05 \cdot 10^6}{s^4 + 100s^3 + 4500s^2 + 1.05 \cdot 10^5 s + 1.05 \cdot 10^6}\,. \tag{5}$$

Following the given transfer functions ((2)–(5)) a pilot model would be

$$Y_P(s) = Y_{vis}(s) \cdot Y_{t_0}(s) \cdot \frac{Y_{NM}(s)}{1 + Y_{PF}(s)}\,. \tag{6}$$

2.3 Pilot-Vehicle System

With given transfer functions for the pilot (6) and for the vehicle dynamics (1) an open loop transfer function for PVS would be:

$$Y_{OL}(s) = Y_{vis}(s) \cdot Y_{t_0}(s) \cdot \frac{Y_{NM}(s)}{1 + Y_{PF}(s)} \cdot Y_{\theta_{1s}}^{\Theta}(s)\,. \tag{7}$$

Closed loop for the PVS system is

$$Y_{CL}(s) = \frac{Y_{OL}(s)}{1 + Y_{OL}(s)}\,. \tag{8}$$

Unknown parameters in the given structure can be described as a parameter set:

$$\begin{bmatrix} K_P & T_L & T_I & K_{PF} & \zeta_{PF} & T_{PF} \end{bmatrix}. \tag{9}$$

A fixed values are chosen for parameters of damping from proprioceptive feedback $\zeta_{PF} = 1$ and lag time constant from visual block $T_I = 0.01$s according to [30], so parameters of the pilot model left to be determined are: K_P, T_L, K_{PF} and T_{PF}.

For the helicopter model (1), with unstable phugoid mode, a determination of the parameter set (9) was conducted following the optimization algorithm [30] based on minimum variance of the error

$$\sigma_e^2 = \sigma_{e,i}^2 \cdot \left[1 + \frac{1 + T_L^2 \frac{\sigma_{\dot{e},i}^2}{\sigma_{e,i}^2} + \frac{\sigma_{e0}^2}{\sigma_{e,i}^2}}{\frac{1}{K_{ni}^*} - \int_0^\infty |Y_{CL}|^2 d\omega} \cdot \int_0^\infty \frac{|Y_C L|^2}{1 + \omega^2 T_L^2} d\omega \right] . \tag{10}$$

Variances correlated to the input signal, of error $\sigma_{e,i}^2$ and error rate $\sigma_{\dot{e},i}^2$ are defined as

$$\sigma_{e,i}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| \frac{1}{1 + Y_{OL}} \right|^2 S_{ii} d\omega ,$$
$$\sigma_{\dot{e},i}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega^2 \left| \frac{1}{1 + Y_{OL}} \right|^2 S_{ii} d\omega ,$$

while residual remnant is neglected, $\sigma_{e0}^2 = 0$. Factor of the level of remnant spectral density K_{ni}^* is defined with the fraction f_i of the pilot's attention for the task

$$K_{ni}^* = \frac{0.01}{f_i} .$$

Standardized form of the power spectral density of the input forcing function is chosen as

$$S_{ii} = \frac{K_i^2}{\left(\omega^2 + \omega_i^2\right)^2} .$$

For the given flight task following parameter values were chosen: $f_i = \frac{2}{3}$, $K_i = \sqrt{2}$ and $\omega_i = 0.5$ rad/s.

Optimization algorithm includes additional constrains, regarding the open loop M gain and ϕ phase margin, applied in order to ensure stable result for the PVS model:

$$M > 1 \quad \text{and} \quad \phi > 10^\circ . \tag{11}$$

For the flight task of forward pitch tracking task at 80 kt, with helicopter dynamics (1) with open loop transfer function (7) for PVS, a parameter set resulting from the optimization algorithm with minimal variance of the error is:

$$\left[\begin{array}{cccccc} K_P & T_L & T_I & K_{PF} & \zeta_{PF} & T_{PF} \end{array} \right] = \left[\begin{array}{cccccc} 1.04 & 0.268 & 0.01 & 40.0 & 1.0 & 3.59 \end{array} \right] . \tag{12}$$

Results for the PVS with parameter set (12) are presented on Figs. 2, 3, 4 and in Table 1.

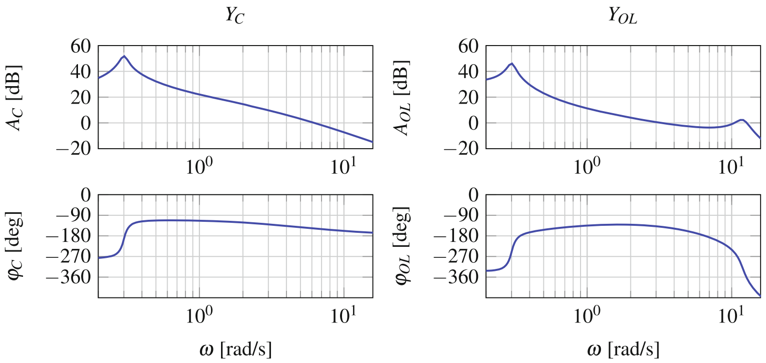


Fig. 2 Bode plots for the PVS elements: aircraft dynamics Y_C and open loop Y_{OL}

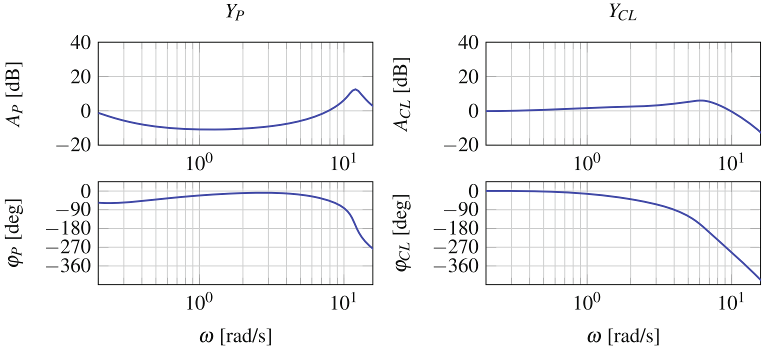


Fig. 3 Bode plots for the PVS elements: pilot model Y_P and closed loop Y_{CL}

Fig. 4 Nichols plot for the open loop of the PVS, Y_{OL}

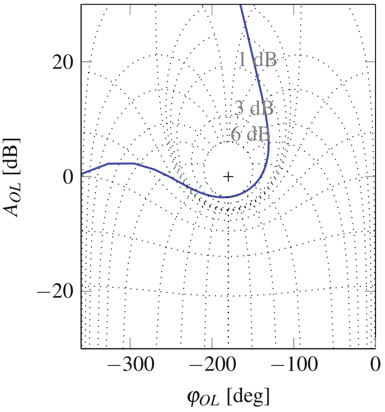


Table 1 Characteristics of the open loop PVS, Y_{OL}

Gain margin: $M = 3.57\,\text{dB}$	Gain margin frequency: $\omega_M = 6.54\,\text{rad/s}$
Phase margin: $\phi = 39.6^\circ$	Crossover frequency: $\omega_c = 3.26\,\text{rad/s}$
Variance $\sigma_e^2 = 0.155$	

3 Experimental Analysis

In addition to the analytical pilot model obtained from the optimization algorithm and used in off-line simulations, an objective of the research was to conduct a nonlinear on-line simulations, an experiment at flight simulator with pilot-in-the-loop.

3.1 Experiment Setup

Experiment was conducted on a research helicopter flight simulator The Rotorcraft Simulation Environment – ROSIE [31] at the Institute of Helicopter Technology at Technical University of Munich.

ROSIE is a pilot-in-the-loop flight simulator with fixed base and high fidelity visual system. It utilizes GENSIM, a nonlinear flight model with the blade aerodynamic model based on blade element/momentum theory with simple analytical downwash models and rigid blades [3, 22]. The research environment enables modifications in the simulation and also access to the output data. More detailed description of ROSIE can be found in [31].

A flight task for the pilot-in-the-loop on-line simulations is task of pitch tracking in forward flight at 80 kt. Experiment is designed to record the pitch attitude of the rotorcraft as a response to the pilot's control following (tracking) the commanded pitch. Commanded pitch is given as a time variant signal at the pilot's primary flight display in the form of a magenta bar (Fig. 5), to be followed by the rotorcraft's pitch. Three different signals for the excitation function were prepared, according to [1]: two variants of the series of step/ramp inputs (with different amplitudes; signals 1 and 2) and a sum of sinusoids (with different frequencies and amplitudes; signal 3).

Experiment on the ROSIE (Fig. 6) was performed in two sessions and thirteen simulation flights were recorded in total. Three different simulation flights recordings for each signal of pitch command were used in this analysis. These nine simulation flights were flown by two pilots, experienced simulator pilots. They were well prepared for the experiment, with the detailed introduction to the flight task, each performing a trial run. Flight model applied for the experiment was coupled in lateral and longitudinal motion what can be noticed by the small value of roll angle at Fig. 5. Because of this, pilots were instructed to neglect the lateral motion (pedal and lateral stick inputs), after they have achieved trimmed horizontal flight at 80 kt, and to focus only on the commanded pitch given at the primary flight display.

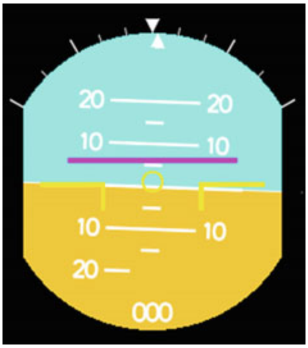


Fig. 5 Pitch command (*magenta bar*) added in the primary flight display

Fig. 6 Experiment flight in ROSIE

3.2 *Experiment Results*

The recordings of the rotorcraft's pitch attitude following a pilot response, as well as the pitch command signals 1, 2 and 3 are presented on Figs. 7, 8 and 9, respectively.

It is of interest to compare responses of the analytical pilot presented in previous section, with results from on-line simulations with pilot-in-the-loop. It should be noted that the PVS with analytical pilot model is based on the simplified helicopter model which doesn't include the definition of the pilot controls (pilot manipulator). Furthermore, simulations of linear analytical pilot model where conducted in time domain with remnant assumed to be zero. Comparisons of the time domain recordings from pilot-in-the-loop simulations, presented as a zone that covers all flight

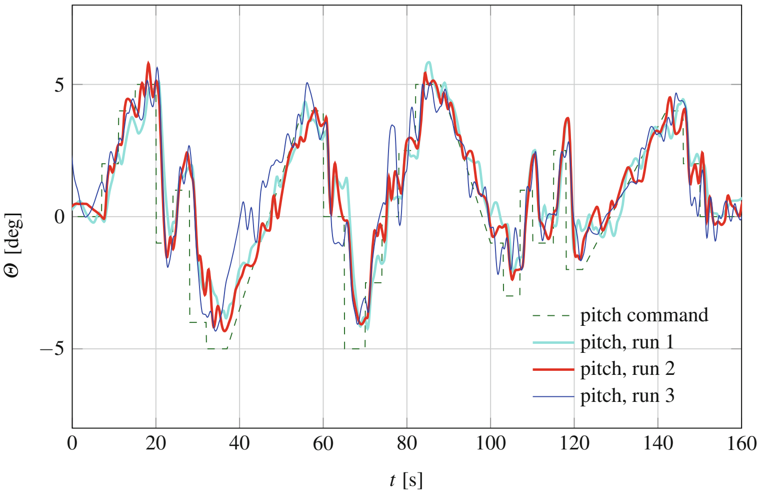


Fig. 7 Measurements of the helicopter's pitch attitude from simulation flights with pitch command signal 1

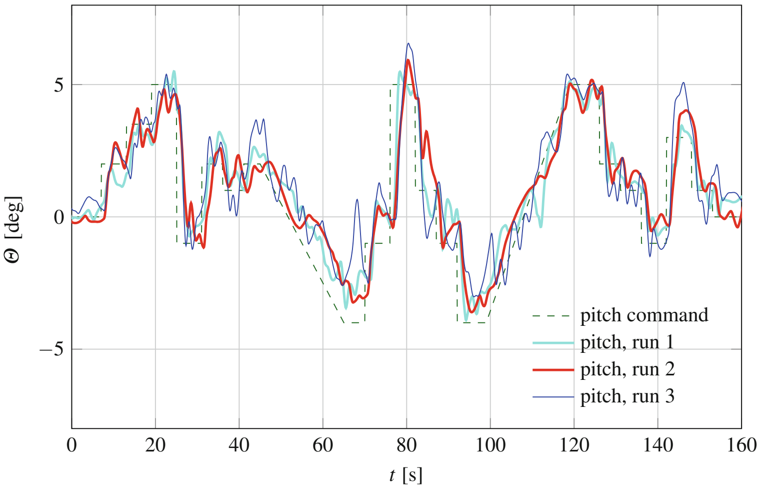


Fig. 8 Measurements of the helicopter's pitch attitude from simulation flights with pitch command signal 2

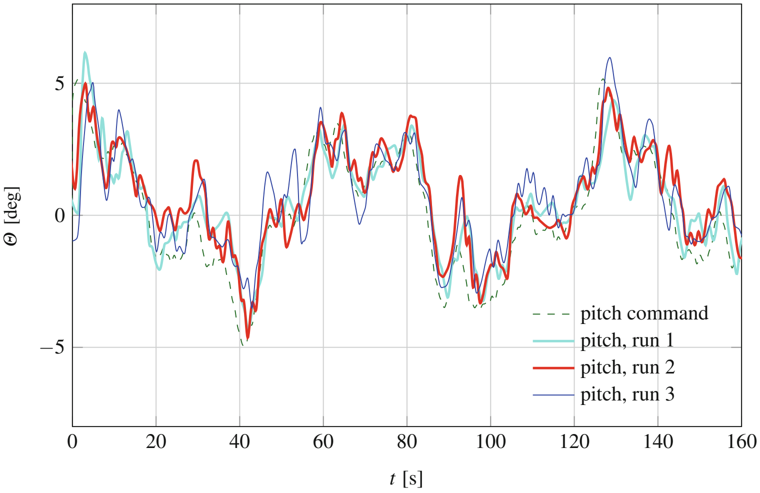


Fig. 9 Measurements of the helicopter's pitch attitude from simulation flights with pitch command signal 3

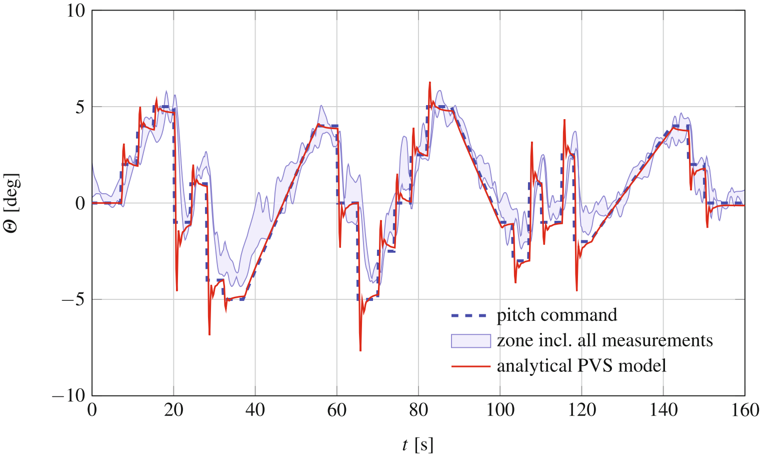


Fig. 10 Time domain comparison between all measured helicopter's pitch attitudes and response of the PVS with analytical pilot model for pitch command signal 1

measurements for each signal, with rotorcraft's pitch response of analytical pilot model (6) with optimal parameter set (12) in PVS model (8) are given on Figs. 10, 11 and 12. For a better illustration of this results comparison for signal 3 is given on Fig. 13 for the selected time range.

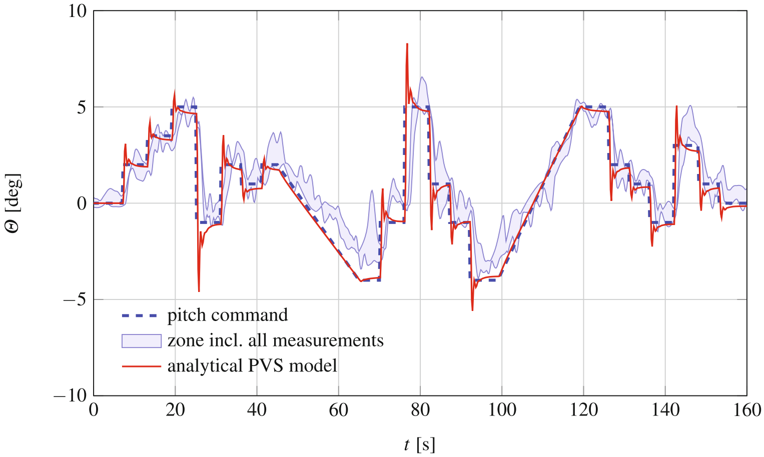


Fig. 11 Time domain comparison between all measured helicopter's pitch attitudes and response of the PVS with analytical pilot model for pitch command signal 2

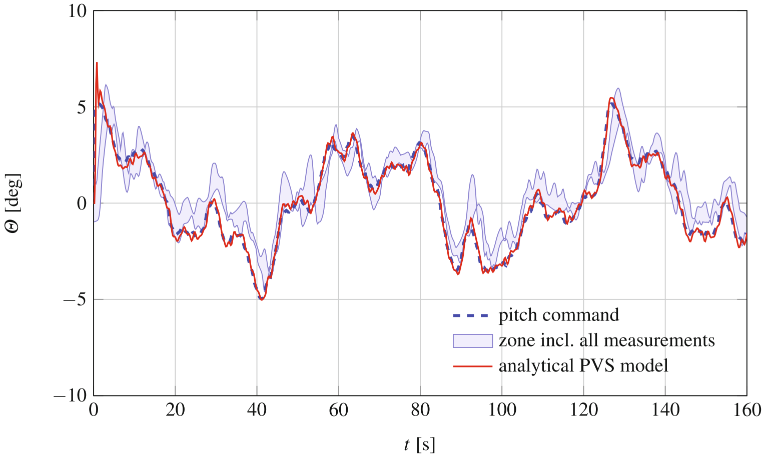
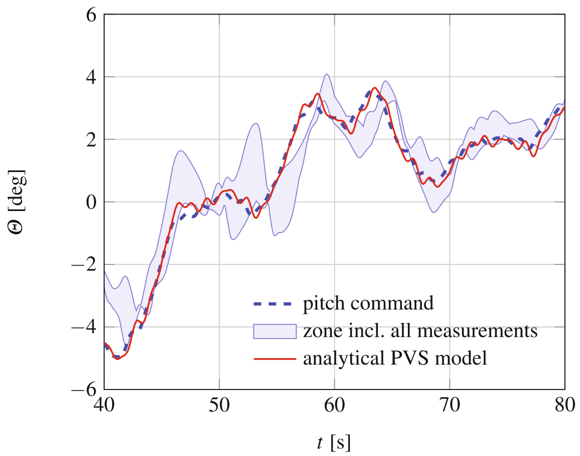


Fig. 12 Time domain comparison between all measured helicopter's pitch attitudes and response of the PVS with analytical pilot model for pitch command signal 3

4 Conclusion

Efremov's modification of structural model was applied for the helicopter pilot model for the pitch attitude tracking task. In the pilot-vehicle system approach a reduced model of BO-105 helicopter with decoupled dynamics was analyzed, with analytical approach for the estimation of the pilot model applicability. The resulting analytical pilot-vehicle system with minimal variance of the error, including the given constrain,

Fig. 13 Time domain comparison between all measured helicopter's pitch attitudes and response of the PVS with analytical pilot model for pitch command signal 3: a detailed view from Fig. 12 for the selected time range



presents a stable PVS with higher value of the crossover frequency. Its stability gain and phase margins could be improved but this is still left to be investigated, possibly with some additional adaptations in the pilot model structure.

Experiments were conducted on the Rotorcraft Simulation Environment (ROSIE) at the Institute for Helicopter Technology, TUM, a pilot-in-the-loop flight simulator with fixed base and high fidelity visual system. Helicopter's pitch responses from the pilot's longitudinal cyclic control are recorded as an output to the input of pitch command displayed at the primary flight display. Three different input signals of pitch command were investigated with two pilots.

The PVS with analytical pilot model, not including model of the pilot manipulator and remnant, was analyzed with the same input pitch command signals used in experiments. From the time domain comparison of the experiment results with the results of the PVS with the analytical pilot model (especially for pitch command signal 3, Figs. 12 and 13) it can be concluded that the described pilot model with minimal error variance presents an acceptable model for the description of the pilot-vehicle system of the helicopter for the pitch tracking task.

Acknowledgements This research was supported by Deutscher Akademischer Austauschdienst – DAAD.

References

1. Anon (1997) Flying qualities of piloted aircraft. Technical report MIL-HDBK-1797B, U.S. Department of Defense
2. Anon (2000) Aeronautical design standard performance specification, handling qualities requirements for military rotorcraft. Technical report ADS-33E-PRF, United States Army Aviation and Missile Command Engineering Directorate, Redstone Arsenal, AL, USA

3. Dietz M, Maucher C, Schimke D (2010) Addressing today's aeromechanic questions by industrial answers. In: American helicopter society specialists' conference on aeromechanics. San Francisco
4. Dillow JD (1971) The 'paper-pilot'-a digital computer program to predict pilot rating for the hover task. Technical report, DTIC Document
5. Efremov A, Tjaglik M (2011) The development of perspective displays for highly precise tracking tasks. Advances in aerospace guidance, navigation and control. Springer, Berlin, pp 163–174
6. Efremov AV, Ogloblin AV (2006) Progress in pilot-in-the loop investigations for flying qualities prediction and evaluation. In: 25th international congress of the aeronautical sciences - ICAS2006
7. Efremov AV, Alexandrov VV, Koshelenko AV, Tjaglik MS, Tzyan TV (2010) Development of pilot modeling and its application to manual control tasks. In: 27th international congress of the aeronautical sciences - ICAS2010
8. Heffley R (2001) Application of classical-control techniques for computer modeling and simulation of helicopter pilotage tasks. In: 57th annual forum of American helicopter society
9. Heffley RK, Jewell WF, Lehman JM, Van Winkle RA (1979) A compilation and analysis of helicopter handling qualities data. NASA Contract Rep 3144:375
10. Hess RA (1980) Structural model of the adaptive human pilot. J Guid Control Dyn 3(5):416–423
11. Hess RA (1982) Prediction of aircraft handling qualities using analytical models of the human pilot. Technical report TR-84233, NASA
12. Hess RA (1989) Theory for aircraft handling qualities based upon a structural pilot model. J Guid Control Dyn 12(6):792–797
13. Hess RA (2006) Simplified approach for modelling pilot pursuit control behaviour in multi-loop flight control tasks. Proc Inst Mech Eng Part G: J Aerosp Eng 220(2):85–102
14. Hess RA (2008) Obtaining multi-loop pursuit-control pilot models from computer simulation. Proc Inst Mech Eng Part G: J Aerosp Eng 222(G2):189–199
15. Hess RA (2009) Analytical assessment of performance, handling qualities, and added dynamics in rotorcraft flight control. IEEE Trans Syst Man Cybern Part A: Syst Hum 39(1):262–271
16. Hess RA (2016) Modeling human pilot adaptation to flight control anomalies and changing task demands. J Guid Control Dyn 39(3):655–666
17. Hess RA, Chan K (1988) Preview control pilot model for near-earth maneuvering helicopter flight. J Guid Control Dyn 11(2):146–152
18. Hess RA, Gorder PJ (1990) Design and evaluation of a cockpit display for hovering flight. J Guid Control Dyn 13(3):450–457
19. Hess RA, Siwakosit W (2001) Assessment of flight simulator fidelity in multiaxis tasks including visual cue quality. J Aircr 38(4):607–614
20. Hess RA, Watson DC (1986) Cross coupling in pilot-vehicle systems. J Guid Control Dyn 9(6):614–620
21. Hess RA, Zeyada Y, Heffley RK (2002) Modeling and simulation for helicopter task analysis. J Am Helicopter Soc 47(4):243–252
22. Johnson W (2012) A history of rotorcraft comprehensive analyses. Technical report TP-2012-216012, NASA
23. Lone M, Cooke A (2014) Review of pilot models used in aircraft flight dynamics. Aerosp Sci Technol 34:55–74
24. McRuer D, Jex H (1967) A review of quasi-linear pilot models. IEEE Trans Hum Factors Electron HFE-8(3):231–249
25. McRuer D, Graham D, Krendel E (1965) Human pilot dynamics in compensatory systems: theory, models and experiments with controlled element and forcing function variations. Technical report AFFDL-TR-65-15, Air Force Flight Dynamics Laboratory
26. Miller DP, Vinje EW (1968) Fixed-base flight simulator studies of VTOL aircraft handling qualities in hovering and low-speed flight. No. 67-152 in AFFDL-TR. Defense Technical Information Center

27. Ockier CJ (1998) Evaluation of the ADS-33D handling qualities criteria using the Bo 105 helicopter. DLR, Dt. Zentrum für Luft-und Raumfahrt e. V., Abt. Unternehmensorganisation und-information
28. Padfield GD (2007) Helicopter flight dynamics. Blackwell Publishing, Oxford
29. Perhinschi MG (1998) A study of helicopter handling qualities in turbulence using a human pilot structural model. In: AIAA-4148, atmospheric flight mechanics conference, Boston
30. Schuck F (2014) Ein integriertes auslegungskonzept zur sicherstellung exzellenter handling qualities für kleinflugzeuge. Ph.D. thesis, München, Technische Universität München, Dissertation
31. Viertler F, Hajek M (2015) Requirements and design challenges in rotorcraft flight simulations for research applications. In: Proceedings of AIAA SciTech-modeling and simulation technologies conference
32. Vorst J (2001) A pilot model for helicopter manoeuvres. Technical report NLR-TP-98448, National Aerospace Laboratory NLR
33. Yilmaz D, Jump M, Linghai L, Jones M (2011) State-of-the-art pilot model for RPC prediction report. Technical report ACPO-GA-2010-266073 D2.3, TU Delft
34. Yilmaz D, Pavel M, Jones M, Jump M, Lu L (2012) Identification of pilot control behavior during possible rotorcraft pilot coupling events. In: Proceedings of 38th European rotorcraft forum. Amsterdam
35. Zeyada Y, Hess RA (2000) Modeling human pilot cue utilization with applications to simulator fidelity assessment. J Aircr 37(4):588–597

UAV Control System Reconfiguration Under Physical Constrains

Marcin Żugaj

1 Introduction

The number of Unmanned Aerial Systems (UAS) operations increase, but safety of such systems still appears not to be sufficient to integrate with the civil airspace. Various institutions around the world are now focusing on the safety aspects of the UAS usage [7–9, 12]. The researches have shown that safety should be taken into account from the beginning of the development process and that it does not have to significantly increase the cost of the system. It is well known that the safety of a complex system depends on the safety of its elements. However, there are ways to assure that the failure of a single element or subsystem will not lead to an accident.

The UAV aircraft safety depends on several unpredictable factors, such as an aircraft system failure or hostile actions both inside and outside the aircraft. When failures occur, the most important action is to maintain the aircraft controllability. That is why the flight control system reliability has significant impact on the aircraft safety. A typical control system of fixed wing aircraft uses several control surfaces to control the aircraft, and some of them work in pairs and are located on opposite sides of the longitudinal plane of symmetry of the aircraft (Fig. 1). This configuration is easy to handle but its redundancy is highly limited. The redundancy can be increased by splitting all paired control surfaces and controlling them individually. This feature of the control system is referred to as "structural redundancy" [2, 18] and results in increasing the flexibility of aircraft handling in the event of fault. The general approach to design a fault tolerant flight control system is to use the hardware redundancy (multiplication of control system equipment, e.g. actuators, power sources, wires, etc.). Unfortunately, redundant hardware increases the control system complexity, its weight, volume and cost [14]. That is why reconfiguration of the

M. Żugaj (✉)
The Institute of Aeronautics and Applied Mechanics, Warsaw University
of Technology, 00-665 Warszawa ul. Nowowiejska 24, Warszawa, Poland
e-mail: zugaj@meil.pw.edu.pl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_13

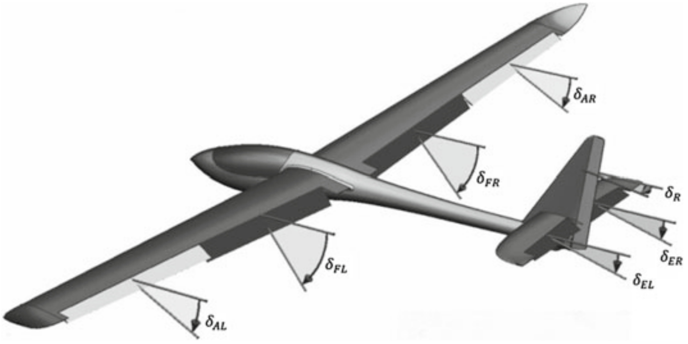


Fig. 1 A UAV airframe configuration

flight control system seems to be a better solution for improving a small fixed wing UAV reliability.

The Reconfiguration of the flight control system [6, 16] makes the control system fault tolerant and ensure the aircraft controllability in the event of fault. The process of flight control system reconfiguration is aimed to take advantage of the working part of the control system in the case of partial system failure. The reconfiguration is usually performed by software algorithms designed to use the remaining control surfaces to compensate the failure effects and modify the strategy of control surfaces handling [2]. The reconfiguration may be performed on three levels. Level 1 performs two actions: sets control surfaces to compensate the failure effects and modifies the strategy of control surfaces handling [2]. In level 2, an attempt is made to rearrange the autopilot control laws to adopt to different aircraft performances [1, 4]. In level 3, on the basis of a prediction about the future situation, refinement of the flight trajectory is performed [10, 15].

The efficiency of the control system reconfiguration depends on the control system structural redundancy (number and positions of the control surfaces), the occurred failure and the performed manoeuvre. Despite the fact that the decoupled control system is more reliable for the reconfiguration process and can be supported by a broad scope of reconfiguration schemes (e.g. Model Reference Adaptive Control [17], Control Allocation [11, 13], Multiple Model Control [3]), the reconfiguration efficiency would be limited by the control surfaces deflection constrains. Moreover, it could be reduced by the cross-coupling effects which result from individual deflection of decoupled control surfaces.

The aim of the present work is to analyse the efficiency of the UAV control system reconfiguration. A six-degree-of-freedom nonlinear model with decoupled control surfaces is employed to determine an optimal control allocation within the bounds of the control surfaces deflection in case of typical control system failures, such as control surface jam [5, 13]. Various algorithms for optimal control allocation are investigated to assess the influence of control system redundancy and constrains on reconfiguration efficiency. The analysis of susceptibility of the control system to the reconfiguration and the simulation tests of control system reconfiguration are performed for various single and multiple control surfaces failures for a broad scope of input signals.

2 Fixed-Wing UAV Dynamic Model

The UAV airplane of 2.6 m wingspan, 1.25 m length, 3 kg max. mass, and 10 N max. thrust is modelled in this study, Fig. 1. The control forces and moments are produced by two ailerons and two flaps, placed on the wings trailing edges, two elevators placed on the horizontal stabilizer trailing edge, and a rudder located on the trailing edge of the fin. The aircraft is propelled by one electric motor with a constant pitch tracking propeller placed in front of the fuselage.

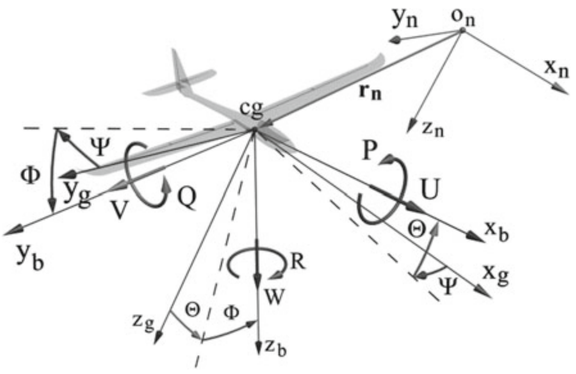
The elevators, ailerons and flaps normally work in pairs and operation of each pair is controlled by a single control input, but each control surface is treated as an individual control input in reconfigurable configuration. The presented UAV dynamic model contains the models of control loads (forces and moments generated by control surfaces) of each individual control surface, and can be widely used for analysis and validation of control methods, including control system reconfiguration.

The aircraft equations of motion are derived in the body coordinate system $O_b x_b y_b z_b$ (Fig. 2) fixed to the airplane fuselage. The centre of the system is placed at the UAV centre of gravity. The $O_b x_b$ axis lies in the plane of aircraft symmetry and is directed forward. The $O_b y_b$ axis is perpendicular to the aircraft plane of symmetry and points right, the $O_b z_b$ axis points "down".

The aircraft translations and attitude angles are calculated in the inertial coordinate system $O_n x_n y_n z_n$; the centre of this system is placed at an arbitrary point on the Earth surface. The $O_n z_n$ axis is along the vector of gravity acceleration, and points down. The $O_n x_n z_n$ plane is horizontal, tangent to the Earth surface, the $O_n x_n$ axis points to the North and the $O_n y_n$ axis to the East.

The vector $\mathbf{y} = [x_n \ y_n \ z_n \ \Phi \ \Theta \ \Psi \ ]^T$ defines the position and attitude of the aircraft (Fig. 2). It is composed of the vector of the aircraft position $\mathbf{r_n} = [x_n \ y_n \ z_n]^T$ in the

Fig. 2 The coordinate systems overview



ground system of coordinates $O_n x_n y_n z_n$ and the roll Φ, pitch Θ and yaw Ψ angles describing the aircraft attitude. The airplane state vector $\mathbf{x} = [\mathbf{v}\ \boldsymbol{\omega}]^T$ is composed of linear velocity components $\mathbf{v} = [U\ \ V\ \ W]^T$ and the angular rate $\boldsymbol{\omega} = [P\ \ Q\ \ R]^T$ in the body system of coordinates $O_b x_b y_b z_b$.

The vectors of the aircraft state, position and attitude are related by a kinematic equation:

$$\dot{y} = \mathbf{T}\mathbf{x} \tag{1}$$

The transformation matrix $\mathbf{T}$ is composed of two matrices: $\mathbf{T}_V$ relating to velocities and $\mathbf{T}_\Omega$ – to rates:

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_V & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\Omega \end{bmatrix} \tag{2}$$

where:

$$\mathbf{T}_V = \begin{bmatrix} \cos\Theta \cdot \cos\Psi & \sin\Theta \cdot \sin\Phi \cdot \cos\Psi - \cos\Phi \cdot \sin\Psi & \cos\Phi \cdot \sin\Theta \cdot \cos\Psi + \sin\Phi \cdot \mathit{s}\mathrm{in}\Psi \\ \cos\Theta \cdot \sin\Psi & \sin\Theta \cdot \sin\Phi \cdot \sin\Psi + \cos\Phi \cdot \cos\Psi & \cos\Phi \cdot \sin\Theta \cdot \sin\Psi - \sin\Phi \cdot \cos\Psi \\ -\sin\Theta & \sin\Phi \cdot \cos\Theta & \cos\Phi \cdot \cos\Theta \end{bmatrix} \tag{3}$$

$$\mathbf{T}_\Omega = \begin{bmatrix} 1 & \sin\Phi \cdot \tan\Theta & \cos\Phi \cdot \tan\Theta \\ 0 & \cos\Phi & \text{-}\sin\Phi \\ 0 & \sin\Phi \cdot \sec\Theta & \cos\Phi \cdot \sec\Theta \end{bmatrix} \tag{4}$$

The airplane equations of motion have been obtained by summing up forces and moments from inertia, gravity $\mathbf{f}_G$, aerodynamic $\mathbf{f}_A$, and propulsion $\mathbf{f}_T$ loads [19]:

$$\mathbf{A}\dot{\boldsymbol{x}} + \mathbf{B}(\mathbf{x})\,\mathbf{x} = \mathbf{f}_A(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta}) + \mathbf{f}_G(\mathbf{y}) + \mathbf{f}_T(\mathbf{x}, \mathbf{y}, \delta_T) \tag{5}$$

The matrix $\mathbf{A}$ describes the inertia properties of the aircraft, and the matrix $\mathbf{B}(\mathbf{x}) = \boldsymbol{\Omega}(\mathbf{x})\,\mathbf{A}$ [19]:

$$\mathbf{A} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & -I_{xz} \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & -I_{xz} & 0 & I_z \end{bmatrix}, \quad \boldsymbol{\Omega}(\mathbf{x}) = \begin{bmatrix} 0 & -R & Q & 0 & 0 & 0 \\ R & 0 & -P & 0 & 0 & 0 \\ -Q & P & 0 & 0 & 0 & 0 \\ 0 & -W & V & 0 & -R & Q \\ W & 0 & -U & R & 0 & -P \\ -V & U & 0 & -Q & P & 0 \end{bmatrix} \tag{6}$$

where $m$is the aircraft mass, I_x, I_y, I_z are moments of inertia, and I_{xz} is the product of inertia.

The aerodynamic loads vector has the form:

$$\mathbf{f_A}(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta}) = \left[\mathbf{f_a}(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta})\ \mathbf{m_a}(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta})\right]^T, \tag{7}$$

where $\boldsymbol{\delta}$ is the control vector, consisting of control surfaces deflections:

$$\boldsymbol{\delta} = \left[\delta_{AR}\ \delta_{AL}\ \delta_{ER}\ \delta_{EL}\ \delta_{R}\ \delta_{FR}\ \delta_{FL}\right]^{T}, \tag{8}$$

where: δ_{AR}—right aileron deflection angle, δ_{AL}—left aileron deflection angle, δ_{ER}—right elevator deflection angle, δ_{EL}—left elevator deflection angle, δ_{FR}—right flap deflection angle, δ_{FL}—left flap deflection angle, and δ_{R}—rudder deflection angle.

The aerodynamic force and moment vectors can be expressed as:

$$\mathbf{f_a} = \mathbf{f}_{\mathbf{a}s} + \mathbf{F}_{\mathbf{A}\boldsymbol{\delta}} = \mathbf{f}_{\mathbf{a}s} + \left[X_{A\delta}\ Y_{A\delta}\ Z_{A\delta}\right], \tag{9}$$

and

$$\mathbf{m_a} = \mathbf{m}_{\mathbf{a}s} + \mathbf{M}_{\mathbf{A}\boldsymbol{\delta}} = \mathbf{m}_{\mathbf{a}s} + \left[L_{A\delta}\ M_{A\delta}\ N_{A\delta}\right], \tag{10}$$

where: $\mathbf{f_{as}}$ and $\mathbf{m_{as}}$ are the vectors of the polar component and stability derivatives, and $\mathbf{F}_{\mathbf{A}\boldsymbol{\delta}}$ and $\mathbf{M}_{\mathbf{A}\boldsymbol{\delta}}$ are the vectors of control derivatives.

3 Reconfiguration of the Flight Control System

3.1 Reconfiguration Algorithms Definitions

A typical configuration of the control system of a fixed-wing aircraft consists of primary control surfaces, which are used to control the aircraft attitude, and secondary control surfaces, used to modify the aircraft flight performances during take-off, landing and transition flight conditions. One of the typical control system failures is the control surface jam, and the reconfigurable flight control system should be able to ensure the controllability of the aircraft in the event of such a failure. The controllability of the aircraft with failed control system depends on the type of the failure and the flight conditions. Generally, it can be defined as an ability of the aircraft to perform manoeuvres which assure as safe a flight as possible. The reconfigurable flight control system should provide the ability to return the aircraft to the airport, land on off-airport location or event perform controlled crash out of the urban area, depending on the magnitude of the failure. For the purposes of a substantial number of control surfaces there is not only one configuration of control surfaces deflection, which enables to control the aircraft after failure. The success of the reconfiguration process depends on the number of operating control surfaces and their deflection limits (bounds). The reconfigured control system provides good controllability performances of the aircraft if the two following conditions are fulfilled: the reconfigured control system produces the same control force and moment vectors as the fault-free system, and the smallest deflections of the working control surfaces are achieved.

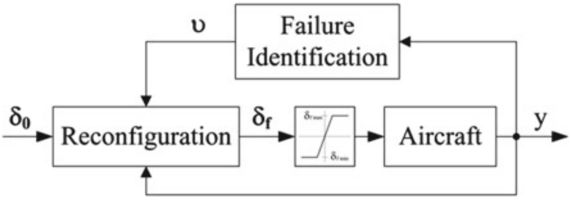


Fig. 3 Structure of the control system reconfiguration method

The first condition guarantees the same modules, directions and senses of the control load vectors, and the second condition allows to avoid the control system bounds problem. In fact, the reconfiguration is an optimal control allocation issue where the appropriate control loads are produced within the system bounds.

There is a broad scope of methods of reconfiguration of the flight control system but their efficiencies depend on the structural redundancy and the bounds of the control system. The control system susceptibility to the reconfiguration should be analysed at first, before the appropriate reconfiguration scheme is applied and the nonlinear aircraft dynamic model with decoupled control surfaces is a good tool to perform such an analysis.

The Fig. 3 presents a typical structure of the reconfiguration of the flight control system via control allocation scheme. The input signals $\boldsymbol{\delta_0}$vector comes from pilot or autopilot, and contains demanded signals for the ailerons δ_{0A}, the elevators δ_{0E} and the rudder δ_{0R} in fault-free conditions. The reconfiguration algorithm distributes these signals to all working primary and secondary control surfaces (denoted as vector $\boldsymbol{\delta_f}$) in the case of a control system failure. The reconfiguration method analysed in this study uses the optimization algorithm to calculate the working control surfaces deflections to obtain, in an ideal case, the same control loads as in an undamaged system with respect to the control system bounds. The reconfiguration algorithm equation has the form:

$$\min_{\boldsymbol{\delta_f}} f_{\delta_f} \quad for \quad \mathbf{f_{Acon}} = 0, \tag{11}$$

where f_{δ_f} is the objective function, $\mathbf{f_{Acon}}$ is the vector of constrains function, and the control system bounds are expressed by:

$$\boldsymbol{\delta_f}_{\min} < \boldsymbol{\delta_f} < \boldsymbol{\delta_f}_{\max}, \tag{12}$$

where $\boldsymbol{\delta_f}_{\min}$ and $\boldsymbol{\delta_f}_{\max}$ are the vectors of maximum and minimum range of working control surfaces deflection (bounds).

The four algorithms have been investigated during this study to analyse the influence of the control system redundancy configuration and the constrain functions on the reconfiguration efficiency. The first one (denoted as *case1*) employs primary and secondary control surfaces without decoupling them. The reconfigured control signal is distributed to rudder δ_{fR}, and ailerons δ_{fA}, elevators δ_{fE} and flaps δ_{fF} which still work in pairs:

$$\boldsymbol{\delta_f} = \begin{bmatrix} \delta_{fA} & \delta_{fE} & \delta_{fR} & \delta_{fF} \end{bmatrix}, \tag{13}$$

the objective function minimizes the sum of squares of these control surfaces deflection:

$$f_{\delta_f} = \sqrt{\sum_i \delta_{fi}^2}, \tag{14}$$

and the constrain function compares the components of control moment vector, and y and z components of control force vector of referenced control system with the failed one:

$$\mathbf{f_{Acon}} = \begin{cases} \mathbf{F_{A\delta\_ref}} - \mathbf{F_{A\delta\_fail}} \\ \mathbf{M_{A\delta\_ref}} - \mathbf{M_{A\delta\_fail}} \end{cases} = \begin{cases} \mathbf{F_{A\delta}^{y,z}}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta_0}\right) - \mathbf{F_{A\delta}^{y,z}}\left(\mathbf{x}, \mathbf{y}, \{\boldsymbol{\delta_f}, \boldsymbol{\upsilon}\}\right) \\ \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta_0}\right) - \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \{\boldsymbol{\delta_f}, \boldsymbol{\upsilon}\}\right) \end{cases}, \tag{15}$$

where $\boldsymbol{\upsilon}$ is the vector of jammed control surfaces deflection and:

$$\boldsymbol{\delta_0} = \left[\ \delta_{0A}\ \delta_{0E}\ \delta_{0R}\ \right]. \tag{16}$$

This form of the optimization procedure allows to obtain the same control force and moment vectors as for undamaged aircraft with the lowest deflections of the working control surfaces. The objective function minimizes the control surfaces deflection and the constrain function provides equality of control force and moment vectors components, which guarantees the same modules, directions and senses of the control loads vectors of the reference (fault-free) system and the failed system after reconfiguration. The x component of the force vector is omitted because it is assumed that this component can be compensated by the thrust force.

The second algorithm (*case2*) employs the decoupled primary and secondary control surfaces and the working control surfaces vector $\boldsymbol{\delta_f}$ consisting of $7 - m$ elements, where m is the number of failed control surfaces:

$$\boldsymbol{\delta_f} = \left[\ \delta_{f1}\ \delta_{f2} \cdots \delta_{f7-m}\ \right]. \tag{17}$$

These two algorithms are used to analyse the influence of the control system redundancy on reconfiguration efficiency. The next two algorithms are formulated to assess the form of constrains function. These algorithms are based on the second one, and the only constrain function is modified. It is assumed that the variation in z force can be compensated by the change of the angle of attack. That is why equality of z component of the control force is removed from the constrain function in the *case3*:

$$\mathbf{f_{Acon}} = \begin{cases} \mathbf{F_{A\delta\_ref}} - \mathbf{F_{A\delta\_fail}} \\ \mathbf{M_{A\delta\_ref}} - \mathbf{M_{A\delta\_fail}} \end{cases} = \begin{cases} Y_{A\delta}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta_0}\right) - Y_{A\delta}\left(\mathbf{x}, \mathbf{y}, \{\boldsymbol{\delta_f}, \upsilon\}\right) \\ \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta_0}\right) - \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \{\boldsymbol{\delta_f}, \boldsymbol{\upsilon}\}\right) \end{cases}. \tag{18}$$

The algorithm *case4* is used to assess the reconfiguration efficiency without the control force equality, and the only control moment is considered in the constrain function:

$$\mathbf{f_{Acon}} = \mathbf{M_{A\delta\_ref}} - \mathbf{M_{A\delta\_fail}} = \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\delta_0}\right) - \mathbf{M_{A\delta}}\left(\mathbf{x}, \mathbf{y}, \{\boldsymbol{\delta_f}, \boldsymbol{\upsilon}\}\right). \tag{19}$$

3.2 Reconfiguration Algorithm Performances Analysis

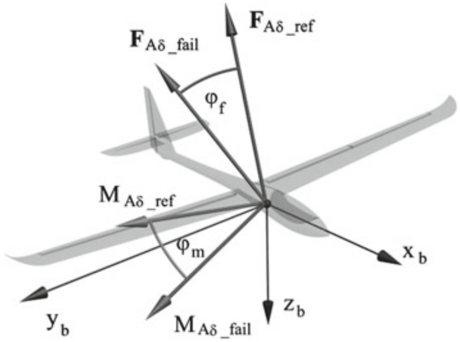
The analysis of the susceptibility of the control system to the reconfiguration and the simulation tests of control system reconfiguration using proposed algorithms have been performed using the nonlinear simulation model of fixed-wing UAV. The tests of control system reconfigurability for single failures of right aileron, left elevator and the rudder, and multiple failures of all possible combinations of these three control surfaces include the reconfigured control system response to the sequence of combinations of input signals $\boldsymbol{\delta_0}$ within a range same as in a normal flight, i.e. ±5° for roll δ_{0A} and pitch δ_{0E} controls, and ±3° for yaw δ_{0R} control with an interval of 1°. The control surface jam at deflection in a range of ±5° for ailerons and elevators, and ±3° for rudder have been chosen as the control system failures. The failed surfaces deflection has remained constant during each particular test. All tests have been performed for cruise flight conditions (velocity of 16 m/s and altitude 30 m), and constant mass properties (mass of 2.22 kg, CG of 32% of MAC).

The reconfiguration efficiency has been assessed by analysis of the working control surfaces deflection and comparison of the control force and moment vectors produced by the failed control system with those produced by the fault-free one. The differences between the fault-free and failed control loads vectors have been expressed by the force and moment vectors deviation angles φ_f and φ_m (Fig. 4), and module differences defined as:

$$\Delta F_{A\delta} = 1 - \left|F_{\mathrm{A\delta\_fail}}\right| / \left|F_{\mathrm{A\delta\_ref}}\right|,\ \Delta M_{A\delta} = 1 - \left|\mathrm{M}_{\mathrm{A\delta\_fail}}\right| / \left|\mathrm{M}_{\mathrm{A\delta\_ref}}\right|, \qquad (20)$$

where *fail* and *ref* sub-indices relate to the control load vectors of the reconfigured and faultless control systems respectively. The working control surfaces deflection, the deviation angles, and the module differences have been subjected to statistical analysis. Furthermore, the raw data of φ_f and φ_m angles are classified in three classes: Low - the value does not exceed 30°; Medium - the value is between 30 and 90°; High - the value is between 90 and 180°. Similarly, the values of control surfaces deflection are classified in four classes: Low - the value does not exceed 1/3 of the maximum

Fig. 4 The control loads deviation angles definition


$\mathbf{F}_{A\delta\_ref}$
$\mathbf{F}_{A\delta\_fail}$
φ_f
$\mathrm{M}_{A\delta\_ref}$
φ_m
x_b
y_b
$\mathrm{M}_{A\delta\_fail}$
z_b

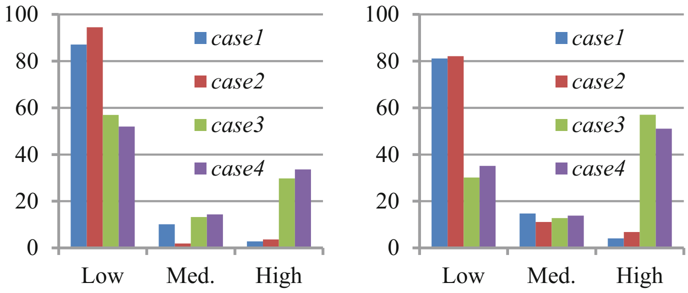


Fig. 5 Proportion of control force vector deviation angle in each class for single (*left*) and double (*right*) failure

Table 1 The statistics of control force vector deviation angle (in degrees) for single and double failure

	Single failure				Double failure			
Statistic	*Case1*	*Case2*	*Case3*	*Case4*	*Case1*	*Case2*	*Case3*	*Case4*
Mean	11.5	6.7	56.8	63.5	17.9	18.6	99.6	89.6
Std.	25.9	24.0	65.0	66.3	29.5	36.3	68.9	68.6
Max	180	180	180	180	180	180	180	180
Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

deflection; Medium - the value is between 1/3 and 2/3 of the maximum deflection; High - the value is between 2/3 of the maximum deflection and full deflection; Max - the control surface reaches its full deflection.

The algorithms of *case1* and *case2* provide lower control force deviation than in the cases 3 and 4 (Fig. 5), which is obvious, since the equality of force vector components has been removed from the constrains equations of Algorithms 3 and 4. Most samples of φ_f values for cases 1 and 2 fall in the *Low* region. The samples for cases 3 and 4 fall mainly in the regions *Low* and *High*, and the samples distribution depends on the failure type: there are more samples in the *Low* region for single failures, and in *High* for multiple failures. The mean values of deviation angle are not higher than 12° and the standard deviations do not exceed 26° in *case1* and *case2*, while the means and standard deviations for case3 and 4 are much higher (Table 1). The lowest values of moment deviation φ_m are achieved by the algorithms of *case3* and *case4*, since almost 100% of samples fall in the class *Low* (Fig. 6). The φ_m values of cases 1 and 2 are distributed in all classes, and the number of the samples in the *Low* class does not exceed 75%, regardless of the failure type. The deviation angles means and standard deviations are close to zero for cases 3 and 4, and are significantly higher for cases 1 and 2 (Table 2). The maximum value of deviation angle reaches 180° (reverse action) in some cases. It means that the reconfiguration algorithm can lead to the loss of control for some combinations of failure and input signals. The most control surfaces deflections are located in the *Low* region for both single and

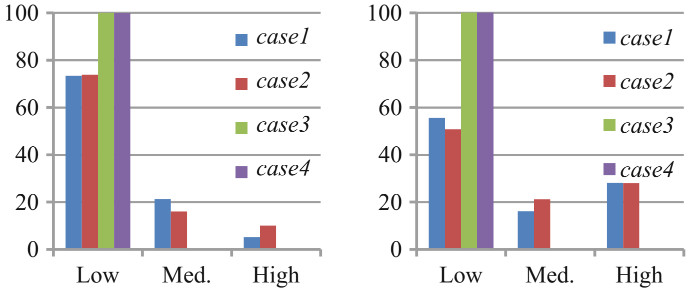


Fig. 6 Proportion of control vector moment deviation angle in each class for single (*left*) and double (*right*) failure

Table 2 The statistics of control moment vector deviation angle (in degrees) for single and double failures

	Single failure				Double failure			
Statistic	*case1*	*case2*	*case3*	*case4*	*case1*	*case2*	*case3*	*case4*
Mean	24.0	23.4	0.03	0.04	52.7	1.9	0.6	0.3
Std.	34.0	39.3	0.6	0.9	56.2	52.5	2.0	1.1
Max	180	180	43.0	43.0	179.7	179.7	20.0	21.5
Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

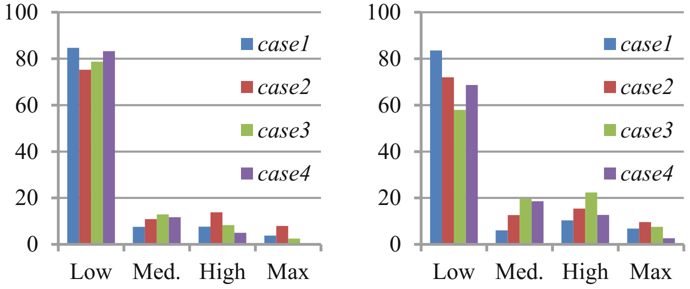


Fig. 7 Proportion of control surfaces deflections in each class for single (*left*) and double (*right*) failures

multiple failures, although the samples distribution changes according to the failure type (Fig. 7). The proportion of surfaces which reach the full deflection is not higher than 10%. The analysis of module differences $\Delta F_{A\delta}$ and $\Delta M_{A\delta}$ prove the above conclusions. The algorithms *case3* and *case4* provide lower module differences in control moment vector and higher in control force vector than cases 1 and 3. This analysis is not presented in the paper due to its complexity, especially for reference vectors modules close to zero.

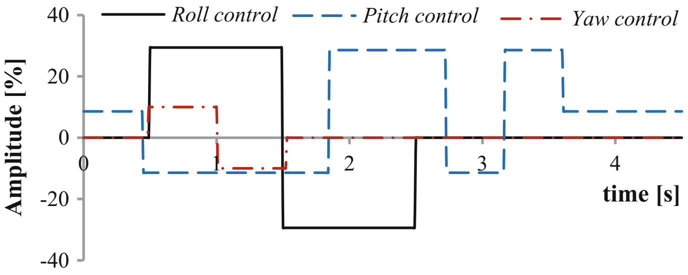


Fig. 8 The simulated input signals as a percentage of their maximum values

The simulation tests of the algorithms have been performed for simple aircraft manoeuvres. The simulated dataset is a collection of input signals $\boldsymbol{\delta_0}$ and aircraft state variables scattered by 7 failure configuration (the same as above) and 7 configuration of simulated input signals, i.e. multi-impulse signal 3211 for pitch δ_{0E}, double impulses for roll δ_{0A} and yaw δ_{0R}, and all combinations of these three signals (Fig. 8). The results of the analysis show that the algorithm *case2* provides good performances for single failure with both single and multiple input signals excitation. The damaged aircraft response is similar to or the same as the response of undamaged aircraft in most test cases (Fig. 9). However, the algorithm causes an unstable behaviour of the aircraft in some case, especially for multiple failure (Fig. 10). The algorithms *case3* and *case4* provide moderate reconfiguration performances for almost all failures and input signals configurations (Fig. 9). What is significant, these algorithms work properly in the case of multiple failures (Fig. 10). The algorithm *case1* provides pure reconfiguration performances, and the aircraft response is often similar to the response of damaged aircraft without reconfiguration (Fig. 9), which proves that decoupled control system guarantees better reconfiguration performances.

The detailed analysis indicates that the aileron jam is the most problematic failure for reconfiguration in the case of all algorithms. It results from its position and size. The aileron produces high aerodynamic force and moment increment, which are difficult to compensate by the rest of the control surfaces. This problem could be partially avoided by redesigning the flaps driving system. The negative flap deflection would help to compensate the lift force increment caused by non symmetric aileron deflection. The redesign of the fin could also improve the efficiency of the reconfiguration. The tests show that the weak efficiency of the reconfigured control system results from the fairly high roll moment produced by the rudder, which compounds the aileron jam reconfiguration.

The algorithm of *case3* and *case4* provide only moderate reconfiguration performances for single failure and fairly good (better than in *case1* and *case2*) for multiple failure, which results from the reduction of constrains equations. The tests show that both force vector and moment vector constrains can only provide the optimal solution, but are too stiff for multiple failures reconfiguration. Therefore modification of the constrains can enable the reconfiguration of the flight control system in presence of

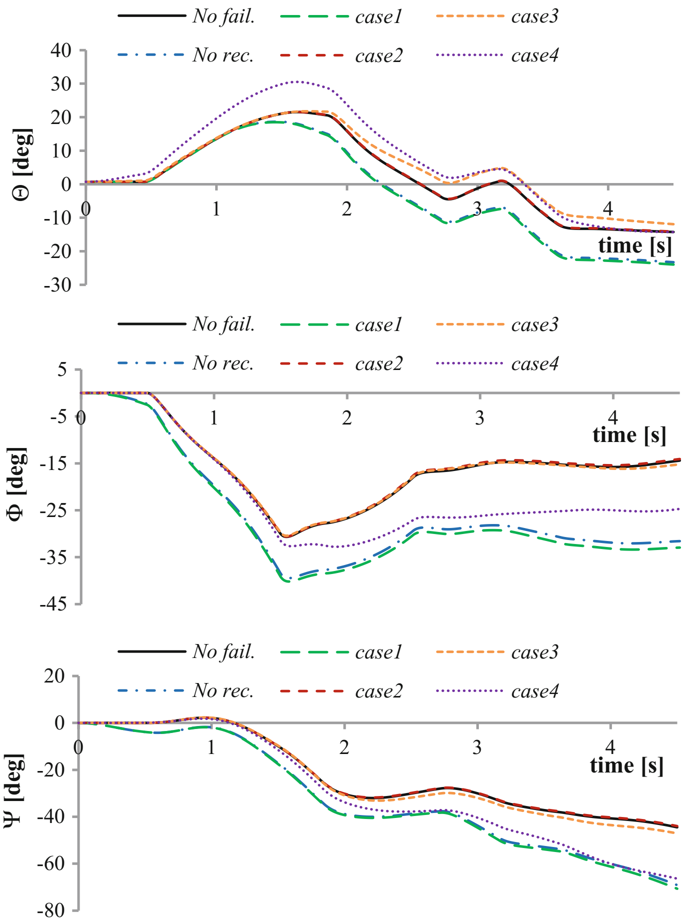


Fig. 9 An example of aircraft response for rudder jam at 3°, and pitch and roll control signals

more serious (e.g. multi-surface) failures with lower controllability performances. The tests analysis also shows that the constrains stiffness problem can be avoided by violating the optimisation bounds. The reconfiguration algorithm finds the optimal solutions within the bounds (Eq. 12), but if the solution which satisfy the constrains cannot be found it can give unexpected results. The reconfiguration algorithm which allows to violate the bounds finds the optimal solution regardless of the bounds if it cannot be found in the bounds at first. It means that it may provide the control signals $\boldsymbol{\delta_f}$, the values of which may exceed the physical bounds, but the control surfaces cannot exceed them. The tests indicate that the reconfiguration performances are

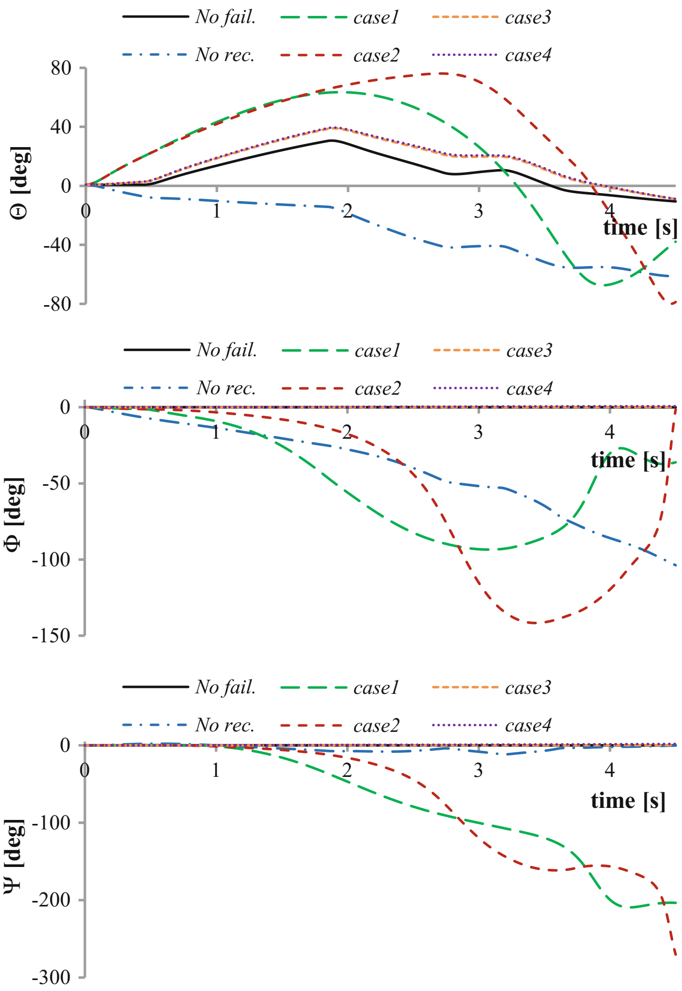


Fig. 10 An example of aircraft response for left aileron and right elevator jam at 5° both, and pitch control signal

better for algorithms where bounds violation applies despite the fact that the physical bounds of control surfaces deflection still exist. It results from the control signals distribution around the bounds, which is reflected by a more feasible constrains solution. If the signals exceed the bounds but, by a small amount, the algorithm finds the constrained minimum, the control force and moment vectors deviations caused by physical bounds are not as significant as in the case of an inviolate algorithm. The Fig. 11 presents an example of comparison of the algorithm of *case2* with and without the bounds violation. The algorithm without bounds violation cannot find a feasible solution within the bounds, which leads to unexpected results and unstable

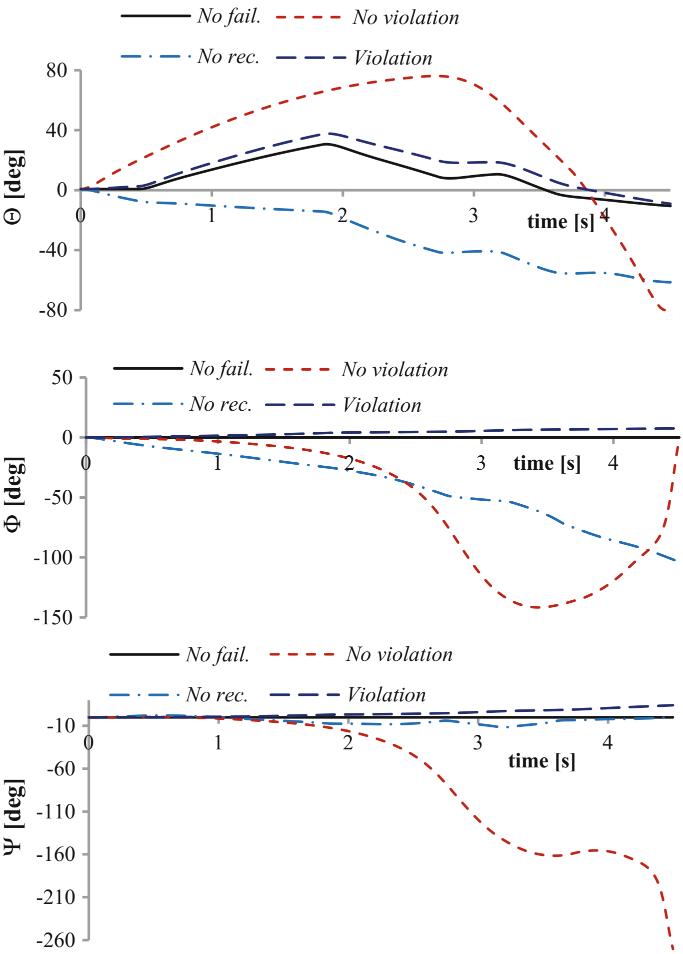


Fig. 11 An example of comparison of algorithm *case2* with and without bounds violation, aircraft response for left aileron and right elevator jam at 5° both, and pitch control signal

behaviour of the aircraft. The algorithm with bounds violation finds the constrained minimum out of the bounds. It provides the flaps deflections out of the limits, but the flap driving system is not able to reach these positions. As can be seen in Fig. 11, the response of the aircraft is fairly good despite the fact that the flaps do not reach the demanded position. It results from a better estimation of other working control surfaces deflection. The test results analysis indicates that the negative flaps deflection is the most common bounds violation, which confirms that the redesign of the flap driving system will improve the reconfiguration efficiency.

4 Conclusions

The influence of UAV control system redundancy and constrains on control system reconfiguration performances are investigated and the nonlinear model-based control allocation method for control system reconfiguration in the event of control surface jam is presented in the paper. The six degrees of freedom UAV aircraft model is used to assess and demonstrate the proposed control system reconfiguration algorithms performances.

The simulation tests results proved the influence of control system redundancy and constrains on reconfiguration performances. The analysis of control system susceptibility to reconfiguration also shows the reconfiguration limitations due to airframe configuration.

References

1. Bodson M (2003) Reconfigurable nonlinear autopilot. J Guid Control Dyn 26(5):719–727
2. Burcham B (1997) Landing safely when flight controls fail. Aerosp Am 35(10):20–23
3. Fekri S, Gu D, Postlethwaite I (2009) Lateral imbalance detection on a UAV based on multiple models. In: Joint 48th IEEE conference on decision and control and 28th Chinese control conference, pp 8488–8493. Shanghai, 16–18 Dec 2009
4. Hass RA, Wells, SR (2003) Sliding mode control applied to reconfigurable flight control design. J Guid Control Dyn 26(3):452–461
5. Kozak V, Shevchuk D, Kravchuk N, Vasilyev M (2013) Intelligent active fault tolerant control system and flight safety. In: Th 7-th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and application, pp 912–916. Berlin, 12–14 Sept 2013
6. Kozak V, Shevchuk D, Vovk V, Levchenko M (2014) Automation of aircraft control Rreconfiguration in flight special situations. In: Proceedings of IEEE 3rd international conference on methods and systems of navigation and motion control, pp 14–17. Kyiv (2014)
7. Lin X, Fulton NL, Horn MET (2014) Quantification of high level safety criteria for civil unmanned aircraft systems. In: Proceedings of aerospace conference, big sky, pp 1–13
8. Loh R, Bian Y, Roe T (2009) UAVs in civil airspace: safety requirements. IEEE Aerosp Electron Syst Maga 24(January):5–17
9. Lyu Y, Pan Q, Zhao C, Zhang Y, Hu J (2016) Vision-based UAV collision avoidance with 2D dynamic safety envelope. IEEE Aerosp Electron Syst Mag 31(7):16–26
10. Masui K, Tomita H, Komatsu Y (2004) Flight experiment on flight path optimization algorithm for aircraft in trouble. In: Proceedings of 24th international conference of the aeronautical science
11. Naskar A, Patra S, Sen S (2015) Reconfigurable direct allocation for multiple actuator failures. IEEE Trans Control Syst Technol 23(1):397–405
12. Nussberger A, Grabner H, Van Gool L (2016) Robust aerial object tracking from an airborne platform. IEEE Aerosp Electron Syst Mag 31(7):38–46
13. Peni T, Vanek B, Szabo Z, Bokor J (2014) Supervisory fault tolerant control of the NASA AirStar aircraft. Am Control Conf Portland 4–6(June):666–671
14. Steinberg M (2005) A historical overview of research in reconfigurable flight control. Proc Inst Mech Eng Part G: J Aerosp Eng 219:263–275
15. Suzuki S, Kawamura F, Masui K (2004) Autonomous flight control and guidance system of accident aircraft. In: Proceedings of 24th international conference of the aeronautical science

16. Yang Z, Hua S, Hongzhuan Q, Chengrui L (2012) Control reconfigurability of nonlinear system based on control redundancy. In: 10th IEEE international conference on industrial informatics (INDIN), pp 815–820. Beijing
17. Zhang Y, Jiang J (2003) Bibliographical review on reconfigurable fault tolerant control systems. In: 5th IFAC symposium on fault detection, supervision and safety for technical processes. Washington, D.C
18. Żugaj M, Bibik P, Jacewicz M (2016) UAV aircraft model for control system failures analysis. J Theor Appl Mech 54(4): 1405–14015
19. Żugaj M, Narkiewicz J (2009) Autopilot for reconfigurable flight control system. ASCE J Aerosp Eng 22(1): 78–84

Part II
Estimation and Navigation

Identification of a Cessna Citation II Model Based on Flight Test Data

M.A. van den Hoek, C.C. de Visser and D.M. Pool

1 Introduction

As a result of new aviation legislation, from 2019 on all air-carrier pilots are obliged to go through flight simulator-based stall recovery training [1]. This implies that all aircraft dynamics models driving flight simulators must be updated to include accurate pre-stall, stall, and post-stall dynamics. For this reason, the Control and Simulation (C&S) division at Delft University of Technology has set up a task force to develop a new methodology for high-fidelity aircraft stall behavior modeling and simulation. This research effort is twofold. First, the current simulation framework is to be updated together with the implementation of a newly developed aerodynamic model identified from flight test data obtained from TU Delft's Cessna Citation II laboratory aircraft. As second part of this research effort, an aerodynamic stall model for the Citation II based on flight test data will be developed and integrated into the upgraded simulation framework.

At this moment, the C&S division uses a simulation model of the Cessna Citation I, known as the Delft University Aircraft Simulation Model and Analysis Tool (DASMAT) [2] as its baseline model. This simulation model was designed as standard Flight CAD package for control and design purposes within the C&S division of the Faculty of Aerospace Engineering, Delft University of Technology. DASMAT is known for a number of deficiencies; most significant is its unsatisfactory match with the current laboratory aircraft's flight dynamics. The Citation I model is the result of a flight test program executed for the development of mathematical models

M.A. van den Hoek (✉) · C.C. de Visser · D.M. Pool
Delft University of Technology, Delft, The Netherlands
e-mail: m.vandenhoek@onderzoeksraad.nl

C.C. de Visser
e-mail: c.c.devisser@tudelft.nl

D.M. Pool
e-mail: d.m.pool@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_14

describing the aerodynamic forces and moments, engine performance characteristics, flight control systems and landing gear [3]. Earlier attempts at modeling the longitudinal forces and the pitching moment were made by Oliveira et al. [4]. However, parameter estimates were only obtained for a limited range of flight conditions with a very limited set of measurements. In addition, in the same paper the authors state that dependency of the aerodynamic model from higher order terms, such as α^2 and terms relating to the time rate of change of the aerodynamic angles, such as $\dot{\alpha}$, are yet to be investigated [4].

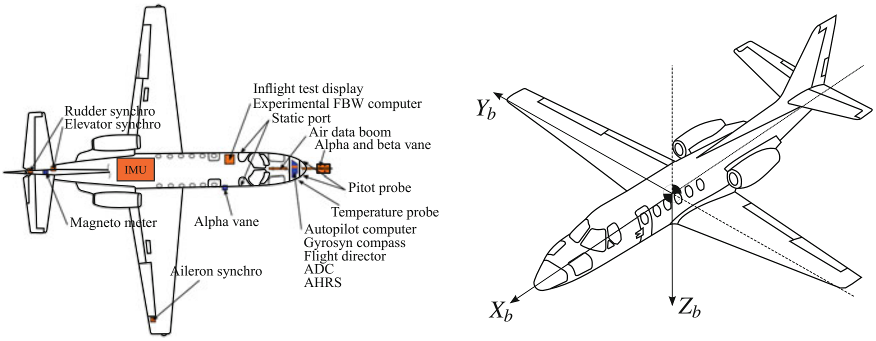
The estimation of stability and control derivatives from flight test data can be formulated in the framework of maximum likelihood estimation [5]. In the context of this paper, aerodynamic model identification will be done by employing the Two-Step Method (TSM) [6, 7]. This method effectively decomposes the non-linear model identification problem into a non-linear flight path reconstruction problem and linear parameter estimation problem, allowing the use of linear parameter estimation techniques for a significant simplification of the latter procedure. This decomposition can be made under certain conditions concerning accuracy and type of the in-flight measurements [7]. New to the TSM approach is the use of the Unscented Kalman Filter [8] (UKF) for an improved accuracy and robustness of the state estimates in the first step.

2 Research Vehicle and Flight Data

In this paper, aerodynamic model identification was applied to the Cessna Citation II laboratory aircraft, model 550, which is co-owned by Delft University of Technology (DUT) and the Netherlands Aerospace Center (NLR). The Citation II is a twin-jet business aircraft, with two Pratt & Whitney JT15D-4 turbofan engines. Both engines deliver a maximum thrust of 11.1 kN each. The maximum operating speed is limited at 198.6 m/s, with a maximum operating altitude of approximately 13 km [9].

2.1 Instrumentation

The Flight Test Instrumentation System (FTIS) of the Cessna Citation II laboratory aircraft combines the sensor measurements from a variety of instrumentation systems. An overview of the instrumentation systems is highlighted in Fig. 1a and summarized in Table 1.



(a) Overview of the aircraft instrumentation systems

(b) Definition of the aircraft body-fixed reference frame or coordinate system $\mathbb{F}_b$

Fig. 1 Aircraft instrumentation systems and reference frame

Table 1 Flight Test Instrumentation System sensor variables used in flight path reconstruction with their associated 1σ standard deviation and sampling rate F_s

Parameter	Unit	1σ std	F_s [Hz]	*Source
Altitude	m	3.00×10^{-1}	16.67	Static probe
X_b-axis rotation	rad	8.70×10^{-3}	50	Sperry vertical gyro
Y_b-axis rotation	rad	8.70×10^{-3}	50	Sperry vertical gyro
Z_b-axis rotation	rad	1.73×10^{-2}	10	Gyrosyn compass
True airspeed	m/s	1.00×10^{-1}	16.67	Pitot-static probe
Angle of attack	rad	3.50×10^{-3}	1000	Alpha vane
Angle of sideslip	rad	3.50×10^{-3}	1000	Beta vane
X_b-axis linear acceleration	m/s^2	2.00×10^{-2}	100	Q-Flex 3100 accelerometer
Y_b-axis linear acceleration	m/s^2	2.00×10^{-2}	100	Q-Flex 3100 accelerometer
Z_b-axis linear acceleration	m/s^2	3.00×10^{-2}	100	Q-Flex 3100 accelerometer
X_b-axis rotational rate	rad/s	2.00×10^{-3}	100	LITEF μFORS rate gyro
Y_b-axis rotational rate	rad/s	2.00×10^{-3}	100	LITEF μFORS rate gyro
Z_b-axis rotational rate	rad/s	5.00×10^{-3}	100	LITEF μFORS rate gyro

*Sampling rate values correspond to the new FTIS. Data obtained from the old FTIS have different sampling rates

3 Flight Path Reconstruction

In this section, the methodology for the flight path reconstruction procedure is presented.

3.1 Kalman Filtering Preliminaries

The set of stochastic differential equations, in the context of aircraft dynamics, can in general be described by:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \mathbf{G}(\mathbf{x}(t), t)\mathbf{w}(t)\\ \mathbf{z}_n(t) &= \mathbf{h}\left[\mathbf{x}(t), \mathbf{u}(t), t\right]\\ \mathbf{z}(t) &= \mathbf{z}_n(t) + \mathbf{v}(t)\end{aligned} \tag{1}$$

where $\mathbf{f}[\cdot]$ is the non-linear state transition function and $\mathbf{h}[\cdot]$ the non-linear measurement function. The process noise and (output) measurement noise are assumed to be zero-mean, white and uncorrelated and can be parametrized by:

$$\mathbb{E}\{\mathbf{v}\mathbf{v}^{\mathsf{T}}\} = \mathbf{Q} \qquad \mathbb{E}\{\mathbf{w}\mathbf{w}^{\mathsf{T}}\} = \mathbf{R} \qquad \mathbb{E}\{\mathbf{w}\mathbf{v}^{\mathsf{T}}\} = 0 \tag{2}$$

where the diagonal elements of the process and measurement noise covariance matrices are composed of the squared standard deviation as given in Table 1. The full kinematic model is given by combining the differential equations for the flat earth position, body velocity components and the equations of rotational motion. The whole set of differential equations is then given by:

$$\begin{aligned}\dot{z}_E &= -u\sin\theta + (v\sin\phi + w\cos\phi)\cos\theta\\ \dot{u} &= a_x - g\sin\theta - qw + rv\\ \dot{v} &= a_y + g\cos\theta\sin\phi - ru + pw\\ \dot{w} &= a_z + g\cos\theta\cos\phi - pv + qu\end{aligned} \qquad \begin{aligned}\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta\\ \dot{\theta} &= q\cos\phi - r\sin\phi\\ \dot{\psi} &= \frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta}\end{aligned} \tag{3}$$

In this set of kinematic equations, the IMU measurements are used as system input. In order to model the noise characteristics and bias of the IMU signals, these were modeled as:

$$\begin{aligned}a_{x_m} &= a_x + \lambda_{a_x} + w_x\\ a_{y_m} &= a_y + \lambda_{a_y} + w_y\\ a_{z_m} &= a_z + \lambda_{a_z} + w_z\end{aligned} \qquad \begin{aligned}p_m &= p + \lambda_p + w_p\\ q_m &= q + \lambda_q + w_q\\ r_m &= r + \lambda_r + w_r\end{aligned} \tag{4}$$

where λ indicates the bias of the associated signal and w indicates the process noise of the subscripted variable.

In the context of this paper, angle of attack and angle of sideslip measurements were primarily obtained through the use of an intrusive nose boom (see Fig. 1a). To this end, the set of observation equations was extended by including the equation for the angle of attack and angle of sideslip as measured by the boom [10] including the sensor biases [11]. This model contains an unknown fuselage-upwash coefficient $C_{\alpha_{\text{up}}}$ together with a kinematically induced angle of attack and angle of sideslip, under the assumption of a zero vertical wind component and alignment of the boom with the X_b-axis. The complete set of observation equations, or the navigation model, is given by:

$$\begin{aligned} & & V_{\text{TAS}_m} &= \sqrt{u^2 + v^2 + w^2} + v_{V_{\text{TAS}}} \\ h_m &= h + v_h & \alpha_v &= (1 + C_{\alpha_{\text{up}}}) \tan^{-1} \frac{w}{u} + \frac{(q - \lambda_q) x_{v_\alpha}}{\sqrt{u^2 + v^2 + w^2}} + v_\alpha \\ \phi_m &= \phi + v_\phi & & \\ \theta_m &= \theta + v_\theta & & \\ \psi_m &= \psi + v_\psi & \beta &= \tan^{-1} \frac{v}{\sqrt{u^2 + w^2}} - \frac{(r - \lambda_r) x_{v_\beta}}{\sqrt{u^2 + v^2 + w^2}} + v_\beta \end{aligned} \tag{5}$$

where v is the standard notation for the measurement noise of the subscripted variable and x_v denotes the location of the boom along the X_b-axis for the alpha and beta vane.

For use in flight path reconstruction with a Kalman filter, the set of equations in Eq. (3) was extended with the time derivatives of additional states that require reconstruction, i.e. sensor biases. Commonly, the state transition function is simply assumed to be zero since the bias is constant in reality. For increased excitation of the sensor bias state, the state transition function for the linear accelerations and fuselage-upwash coefficient was modeled as zero-mean unit-variance random walk scaled by a factor k, as earlier applied in the work of Mulder et al. [12]:

$$\dot{\lambda} \sim k \cdot \mathcal{N}(0, 1) \tag{6}$$

The bias state transition function for the rotational rates was assumed to be zero for its usually very small bias. On balance, the state vector together with the augmented bias terms is given by:

$$\mathbf{x} = [h \quad u \quad v \quad w \quad \phi \quad \theta \quad \psi \quad \lambda_{a_x} \quad \lambda_{a_y} \quad \lambda_{a_z} \quad \lambda_p \quad \lambda_q \quad \lambda_r \quad C_{\alpha_{\text{up}}}]^\mathsf{T} \tag{7}$$

3.2 *Kalman Filtering Procedure*

To begin with the formulation of the augmented UKF [8, 13–15], the augmented state vector and covariance matrix are defined as:

$$\hat{\mathbf{x}}^a(k) = \left[\hat{\mathbf{x}}(k|k)^\mathsf{T} \; \mathbf{v}(k)^\mathsf{T} \; \mathbf{w}(k)^\mathsf{T}\right]^\mathsf{T} \tag{8}$$

$$\mathbf{P}^a(k) = \begin{bmatrix} \mathbf{P}(k) & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{bmatrix} \tag{9}$$

where **v** and **w** in the augmented state vector represent the means of the process and measurement noise; these can therefore be assumed to have zero mean, hence their values will be zero. The augmented state vector and covariance matrix can then easily be transformed to the unscented domain by:

$$\begin{aligned} \mathscr{X}_i^a(k) &= \left[\hat{\mathbf{x}}^a(k) + \sqrt{(L+\lambda)\mathbf{P}^a(k)}\right] \quad i = 1, 2, \ldots, L \\ \mathscr{X}_i^a(k) &= \left[\hat{\mathbf{x}}^a(k) - \sqrt{(L+\lambda)\mathbf{P}^a(k)}\right] \quad i = L+1, L+2, \ldots, 2L \end{aligned} \tag{10}$$

This set of transformed points, indicated by $\mathscr{X}^a$, is referred to as the set of sigma points. Parameters L and λ are, respectively, the dimensionality of the state vector and a scaling factor defined as $\lambda = \alpha^2(L+\kappa) - L$. α is a parameter to reflect the spread of the sigma points around its mean, state vector $\hat{\mathbf{x}}$, and β is a factor to account for any prior knowledge. The latter is set to a value of 2 for Gaussian distributions. κ is an extra scaling factor which is usually set to zero. Subsequently, the weights for the set of transformed means and covariances are defined as follows:

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\ W_0^{(c)} &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\ W_i^{(m)} = W_i^{(c)} &= \frac{1}{2(L+\lambda)} \qquad i = 1, 2, \ldots, 2L \end{aligned} \tag{11}$$

From this point, the equations of the UKF become more trivial. Analogously to the EKF, the state vector which is now expressed as sigma points are propagated through the system's dynamics:

$$\mathscr{X}^a(k+1|k) = \mathscr{X}^a\,(k|k) + \int_{t_k}^{t_{k+1}} \mathbf{f}\left[\mathscr{X}^{a,x}(k|k), \mathbf{u}(k), \mathscr{X}^{a,v}(k|k), \tau\right] d\tau \tag{12}$$

where $\mathscr{X}^{a,x}$ refers to the columns of the sigma points matrix related to the state and superscript v refers to the sigma points related to the process noise. The one step ahead state estimation can be calculated by:

$$\hat{\mathbf{x}}\,(k+1|k) = \sum_{i=0}^{2L} W_i^{(m)} \mathscr{X}^a\,(k+1|k) \tag{13}$$

and the one step ahead covariance matrix by:

$$\mathbf{P}\,(k+1|k) = \sum_{i=0}^{2L} W_i^{(c)} \left(\mathscr{X}_i^{a,x} - \hat{\mathbf{x}}(k|k)\right)\left(\mathscr{X}_i^{a,x} - \hat{\mathbf{x}}(k|k)\right)^{\mathsf{T}} \tag{14}$$

Again, similarly to the EKF, the sigma points representing the state vector and measurement noise are propagated through the measurement equations and subsequently the transformed means for the measurements are calculated:

$$\mathscr{Y}(k+1|k) = \mathbf{h}\left[\mathscr{X}^{a,x}(k+1|k),\, \mathscr{X}^{a,w}(k+1|k)\right] \tag{15}$$

with the transformed measurements given by taking the mean of the transformed sigma points:

$$\hat{\mathbf{y}} = \sum_{i=0}^{2L} W_i^{(m)} \mathscr{Y}_i(k+1|k) \tag{16}$$

The measurement covariance and measurement-state cross-covariance can be calculated by:

$$\mathbf{P_{yy}} = \sum_{i=0}^{2L} W_i^{(c)} \left(\mathscr{Y}_i(k+1|k) - \hat{\mathbf{y}}(k|k)\right)\left(\mathscr{Y}_i(k+1|k) - \hat{\mathbf{y}}(k|k)\right)^{\mathsf{T}} \tag{17}$$

$$\mathbf{P_{xy}} = \sum_{i=0}^{2L} W_i^{(c)} \left(\mathscr{X}_i^{a,x} - \hat{\mathbf{x}}(k|k)\right)\left(\mathscr{Y}_i - \hat{\mathbf{y}}(k|k)\right)^{\mathsf{T}} \tag{18}$$

Finally, to complete the definition of the augmented UKF, gain matrix $\mathscr{K}$, corrected state estimation $\hat{\mathbf{x}}(k+1|k+1)$ and corrected covariance matrix estimation $\mathbf{P}(k+1|k+1)$ are expressed as:

$$\mathscr{K}(k+1) = \mathbf{P_{xy}}\mathbf{P_{yy}}^{-1} \tag{19}$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathscr{K}\left\{\mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1|k)\right\} \tag{20}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathscr{K}(k+1)\mathbf{P_{yy}}\mathscr{K}^{\mathsf{T}}(k+1) \tag{21}$$

For additional numerical stability and guaranteed semi-definite state covariance matrix, the square-root implementation of the UKF can be used [16]. This type uses the Cholesky decomposition to address certain numerical advantages in the calculation of the transformed statistical properties. Further extensions to the UKF, e.g. the Sigma-Point Kalman Filter [17] and its iterative counterpart [18] were introduced later. However, these filters populate the whole state-space with sigma points instead of only a selected optimal range. Therefore, the computational burden of such filters do not outweigh the advantages and their application is restricted [19].

4 Aerodynamic Model Identification

4.1 Preliminaries

The six non-dimensional forces and moments can be calculated by:

$$C_X = \frac{m\left(a_x - \lambda_{a_x}\right) - T_x}{\overline{q}S} \tag{22}$$

$$C_Y = \frac{m\left(a_y - \lambda_{a_y}\right)}{\overline{q}S} \tag{23}$$

$$C_Z = \frac{m\left(a_x - \lambda_{a_z}\right)}{\overline{q}S} \tag{24}$$

$$C_l = \frac{I_{xx}}{\overline{q}Sb}\left(\dot{p} - \frac{I_{xz}}{I_{xx}}\left(\left(p-\lambda_p\right)\left(q-\lambda_q\right)+\dot{r}\right) + \frac{I_{zz}-I_{yy}}{I_{xx}}\left(q-\lambda_q\right)\left(r-\lambda_r\right)\right) \tag{25}$$

$$C_m = \frac{I_{yy}}{\overline{q}S\overline{c}}\left(\dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}}\left(p-\lambda_p\right)\left(r-\lambda_r\right) + \frac{I_{xz}}{I_{yy}}\left(\left(p-\lambda_p\right)^2-\left(r-\lambda_r\right)^2\right) - M_T\right) \tag{26}$$

$$C_n = \frac{I_{zz}}{\overline{q}Sb}\left(\dot{r} - \frac{I_{xz}}{I_{zz}}\left(\dot{p}-\left(q-\lambda_q\right)\left(r-\lambda_r\right)\right) + \frac{I_{yy}-I_{xx}}{I_{zz}}\left(p-\lambda_p\right)\left(q-\lambda_q\right)\right) \tag{27}$$

where λ denotes the bias obtained from the flight path reconstruction procedure for each of the accelerations and rotational rates. Since the derivatives of the rotational rates are not measured directly, these can be obtained by numerical differentiation. Corrections to the non-dimensional force in X_b and the non-dimensional pitch rate were made by making use of an engine model. The engine-produced thrust in Z_b was neglected and assumed to be approximately zero.

4.2 Parameter Estimation

The principles of regression analysis are well known and previously applied in many different researches in the framework of aerodynamic system identification [20–22]. The ordinary least squares (OLS) estimator, defined as the minimum residual

$$\boldsymbol{\Theta}_{\text{OLS}} = \min_{\Theta\in\mathbb{R}} \|\mathbf{X}\cdot\boldsymbol{\Theta} - \mathbf{y}\| \tag{28}$$

where $\|\cdot\|$ denotes the L^2 norm in Euclidean space $\mathbb{R}^n$. The well-known solution in terms of linear operations is given by:

$$\hat{\boldsymbol{\varTheta}}_{\text{OLS}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathsf{T}}\mathbf{y} \tag{29}$$

According to the Gauss-Markov theorem, the OLS estimator is the best linear unbiased estimator under the assumption that the variance of the residuals should be homoscedastic and the correlation terms should vanish [23]. In addition, under the assumption of a normally distributed residuals vector the OLS estimator is identical to the maximum likelihood estimator, effectively attaining the Cramér-Rao lower bounds (CRLB) [24]. The standard bounds of the parameter estimates are given by the diagonal elements of the variance-covariance matrix:

$$\text{Cov}\left\{\boldsymbol{\varTheta}\right\} = \mathbb{E}\left\{\left(\hat{\boldsymbol{\varTheta}} - \boldsymbol{\varTheta}\right)^{\mathsf{T}}\left(\hat{\boldsymbol{\varTheta}} - \boldsymbol{\varTheta}\right)\right\} = \sigma^2\left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \tag{30}$$

where σ^2 can be approximated by the mean squared error of the residuals. Using the estimated covariance, pair-wise correlation of the estimated parameters can be assessed by:

$$\text{Corr}\left\{\hat{\boldsymbol{\varTheta}}\right\} = \begin{pmatrix} \frac{1}{\sigma(\hat{\varTheta}_1)} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma(\hat{\varTheta}_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sigma(\hat{\varTheta}_p)} \end{pmatrix} \text{Cov}\left\{\hat{\boldsymbol{\varTheta}}\right\} \begin{pmatrix} \frac{1}{\sigma(\hat{\varTheta}_1)} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma(\hat{\varTheta}_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sigma(\hat{\varTheta}_p)} \end{pmatrix} \tag{31}$$

Because aircraft parameter estimation is often associated with data collinearity [25], a biased parameter estimation technique known as Principal Components Regression (PRC) was used. PCR is able to increase the accuracy of the parameter estimates in case of multi-collinearity among the predictor variables [20].

4.3 Model Structure Selection

Stepwise regression [26] is a method specifically aimed at data-driven selection of an appropriate model structure from a set of candidate regressors. Later modifications to this approach restricted the selection of candidate regressors to higher order terms, starting at a fixed linear model structure [27]. The pool of candidate regressors is to be formed by single terms, cross-interactions and higher order terms corresponding to the independent variables in the model. The downside of the stepwise regression method is that it includes addition and elimination criteria [28]. In addition, regressors cannot be evaluated independently because of their interaction with other regressors in the selected model structure.

Table 2 Estimated parameters mean variance, minimum variance and maximum variance for the C_X model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{X_0}	−0.051	−0.594	0.019	1.553×10^{-5}	4.134×10^{-8}	4.710×10^{-4}
C_{X_α}	0.862	−0.213	12.733	1.115×10^{-3}	2.059×10^{-5}	5.349×10^{-2}
C_{X_q}	−4.465	−100.213	17.117	8.591×10^{-1}	1.296×10^{-2}	8.320
$C_{X_{\delta_e}}$	−0.172	−3.602	0.842	2.572×10^{-3}	3.688×10^{-5}	2.736×10^{-2}

Table 3 Estimated parameters mean variance, minimum variance and maximum variance for the C_Y model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{Y_0}	0.004	−0.056	0.059	8.638×10^{-8}	3.190×10^{-10}	8.079×10^{-7}
C_{Y_β}	−0.794	−2.258	−0.169	4.389×10^{-4}	1.362×10^{-6}	4.080×10^{-3}
C_{Y_p}	−0.159	−4.163	2.583	1.403×10^{-2}	3.772×10^{-5}	1.152×10^{-1}
C_{Y_r}	1.958	−1.813	13.569	2.199×10^{-2}	3.163×10^{-5}	1.496×10^{-1}
$C_{Y_{\delta_a}}$	−0.180	−4.305	1.397	2.083×10^{-3}	1.548×10^{-6}	2.282×10^{-2}
$C_{Y_{\delta_r}}$	0.839	−1.988	26.784	4.846×10^{-2}	1.152×10^{-6}	1.427
$C_{Y_{\beta 2}}$	2.754	−14.888	48.476	1.028	2.795×10^{-5}	9.398

Table 4 Estimated parameters mean variance, minimum variance and maximum variance for the C_Z model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{Z_0}	−0.213	−0.941	0.025	1.575×10^{-4}	1.075×10^{-6}	5.183×10^{-3}
C_{Z_α}	−4.037	−8.231	2.868	8.074×10^{-3}	2.369×10^{-4}	4.290×10^{-1}
C_{Z_q}	−57.766	−267.955	189.902	$1.320\times10^{+01}$	2.363×10^{-1}	$1.979\times10^{+02}$
$C_{Z_{\delta_e}}$	−0.836	−6.355	25.163	4.456×10^{-2}	7.952×10^{-4}	6.847×10^{-1}

More recently, Morelli [21, 29] and Grauer [30] applied a multi-variate polynomial model obtained from an orthogonal model structure selection to various aircraft. The latter model structure selection technique transforms the full set of candidate regressors to the orthogonal domain in order to test the significance of each parameter. By defining a predicted square error (PSE) [30], selection of the orthogonal basis functions can be done by minimization of the latter metric. Terms contributing less than a certain threshold value can also be removed from the model structure.

The process of orthogonal basis functions model structure selection begins with the orthogonalization process of the set of candidate regressors:

$$\mathbf{p}_0 = 1, \quad \mathbf{p}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \gamma_{kj}\mathbf{p}_k \quad \text{for} \quad j = 1, 2, \ldots, n \tag{32}$$

where $\mathbf{x}_j$ is the j^{th} vector of independent variables and coefficient γ_{kj} is defined as:

$$\gamma_{kj} = \frac{\mathbf{p}_k^{\mathsf{T}} \mathbf{x}_j}{\mathbf{p}_k^{\mathsf{T}} \mathbf{p}_k} \quad \text{for} \quad k = 0, 1, \ldots, j-1 \tag{33}$$

Orthogonal vectors $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n$ now form the columns of orthogonal regression matrix $\mathbf{P}$. The parameter estimate can now be obtained by the least squares estimator in Eq. (29). This can be done by subsequently calculating the contribution to the total least-squares cost independently for each candidate regressor with:

$$J(\hat{\mathbf{a}}_j) = \frac{\left(\mathbf{p}_j^{\mathsf{T}} \mathbf{y}\right)^2}{\mathbf{p}_j^{\mathsf{T}} \mathbf{p}_j} \tag{34}$$

a selection can be made based on the PSE, which is defined as:

$$\text{PSE} = \frac{1}{N} (\mathbf{y} - \hat{\mathbf{y}})^{\mathsf{T}} \left(\mathbf{y} - \hat{\mathbf{y}}\right) + \sigma_{\max}^2 \frac{n}{N} \tag{35}$$

The maximum model fit error variance can be obtained from:

$$\sigma_{\max}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{\mathbf{y}})^2 \tag{36}$$

5 Results

In this section the results of the flight path reconstruction, model structure selection and parameter estimation procedure are presented.

Table 5 Estimated parameters mean variance, minimum variance and maximum variance for the C_l model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{l_0}	−0.002	−0.020	0.010	1.826×10^{-8}	1.182e−10	3.285×10^{-7}
C_{l_β}	−0.073	−0.143	−0.006	1.407×10^{-6}	9.575×10^{-8}	1.490×10^{-5}
C_{l_p}	−0.494	−0.710	0.056	2.656×10^{-5}	1.727×10^{-6}	1.508×10^{-4}
C_{l_r}	0.376	0.024	0.785	6.498×10^{-5}	4.639×10^{-7}	4.298×10^{-4}
$C_{l_{\delta_a}}$	−0.178	−0.276	0.121	6.081×10^{-6}	1.585×10^{-7}	9.996×10^{-5}
$C_{l_{\delta_r}}$	0.102	−1.309	2.314	6.865×10^{-4}	2.784×10^{-8}	1.619×10^{-2}

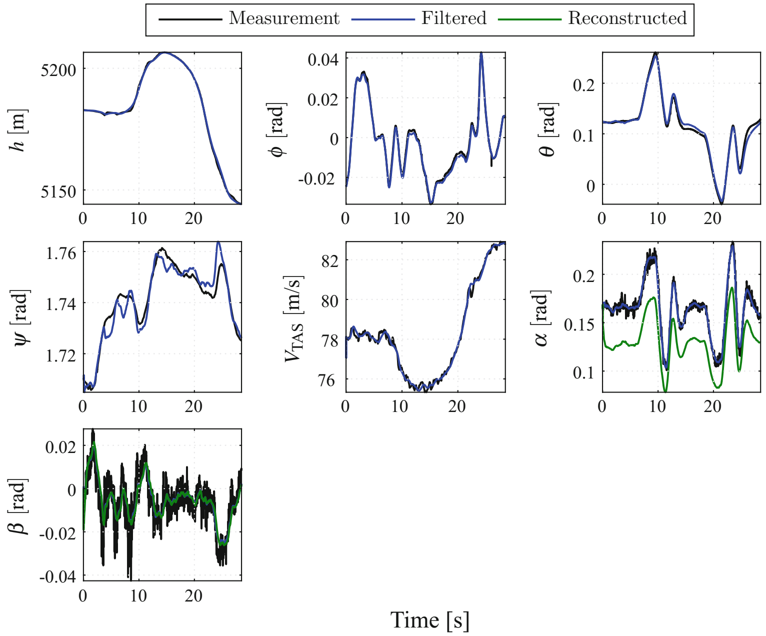


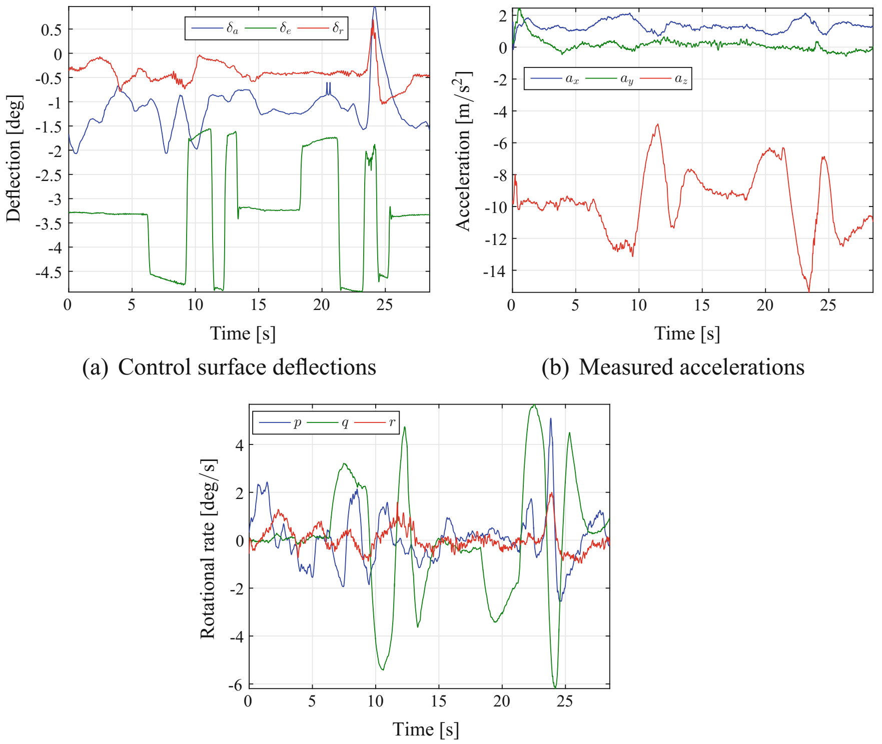
Fig. 2 Measurement values together with filtered and reconstructed measurement sequences

5.1 Flight Path Reconstruction

The results for the flight path reconstruction procedure comprises a total of more than 200 individually reconstructed dynamic maneuvers, both longitudinally and laterally induced. For this reason, only a selection of results is shown in this paper. For a typical 3-2-1-1 dynamic maneuver in elevator, the results are depicted in Figs. 2 and 3. In these figures, the filtered and reconstructed measurements and control surface deflections during the maneuver are shown.

5.2 Aerodynamic Model Identification

The results from the model structure selection procedure and parameter estimation are presented in this section together with a model validation by applying the identified least squares model to flight derived non-dimensional forces and moments together with a comparison versus the currently implemented aerodynamic model in the DASMAT simulation framework (Fig. 4).



(a) Control surface deflections

(b) Measured accelerations

(c) Measured rotational rates

Fig. 3 Control surface deflections for a typical δ_e induced longitudinal 3-2-1-1 dynamic maneuver

Table 6 Estimated parameters mean variance, minimum variance and maximum variance for the C_m model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{m_0}	0.021	-0.022	0.089	4.918×10^{-7}	1.252×10^{-8}	5.698×10^{-6}
C_{m_α}	-0.488	-0.855	-0.253	2.509×10^{-5}	2.856×10^{-6}	1.904×10^{-4}
C_{m_q}	-11.935	-22.066	-1.489	3.466×10^{-2}	2.968×10^{-3}	2.920×10^{-1}
$C_{m_{\delta_e}}$	-1.250	-1.508	-0.351	1.204×10^{-4}	9.907×10^{-6}	1.097×10^{-3}

The final model structure of the non-dimensional forces and moments in X_b, obtained from an orthogonal least squares model selection scheme, consisted of a total of 5 terms, i.e. C_{X_0}, C_{X_α}, C_{X_q}, $C_{X_{\delta_e}}$, $C_{X_{\alpha 2}}$. However, the term related to the squared angle of attack was removed from the model for its high pairwise correlation with the angle of attack term. Identified values for the parameters as tabulated in Table 2. Tabulated values represent the parameters in the total number of locally identified models. The minimum, maximum and mean values for the estimated para-

Table 7 Estimated parameters mean variance, minimum variance and maximum variance for the C_n model, obtained from an orthogonal least squares model structure selection approach

	$\overline{\theta}$	θ_{min}	θ_{max}	$\overline{\sigma}(\theta)$	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$
C_{n0}	0.000	−0.002	0.002	1.158×10^{-8}	2.084e−10	1.326×10^{-7}
C_{n_β}	0.079	−0.056	0.145	3.689×10^{-6}	1.548×10^{-7}	5.965×10^{-5}
C_{n_p}	−0.142	−0.677	0.284	1.307×10^{-4}	5.361×10^{-6}	3.267×10^{-3}
C_{n_r}	−0.295	−0.474	0.374	1.005×10^{-4}	3.055×10^{-6}	5.440×10^{-4}
$C_{n_{\delta_a}}$	−0.025	−0.155	0.073	4.720×10^{-5}	5.616×10^{-7}	1.049×10^{-3}
$C_{n_{\delta_r}}$	−0.065	−0.611	0.578	7.338×10^{-4}	1.783×10^{-7}	1.770×10^{-2}

meters and corresponding variance were included as performance measure to indicate consistence of the estimates.

The models for the 6 dimensionless forces and moments resulting from the model structure selection procedure and parameter estimation (see Tables 2, 3, 4, 5, 6 and 7 for parameter estimates) were parametrized as follows:

$$C_X = C_{X_0} + C_{X_\alpha}\alpha + \cancel{C_{X_{\alpha^2}}\alpha^2} + C_{X_q}\hat{q} + C_{X_{\delta_e}}\delta_e \tag{37}$$

$$C_Y = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\hat{p} + C_{Y_r}\hat{r} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r + C_{Y_{\beta^2}}\beta^2 \tag{38}$$

$$C_Z = C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_q}\hat{q} + C_{Z_{\delta_e}}\delta_e \tag{39}$$

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\hat{p} + C_{l_r}\hat{r} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \tag{40}$$

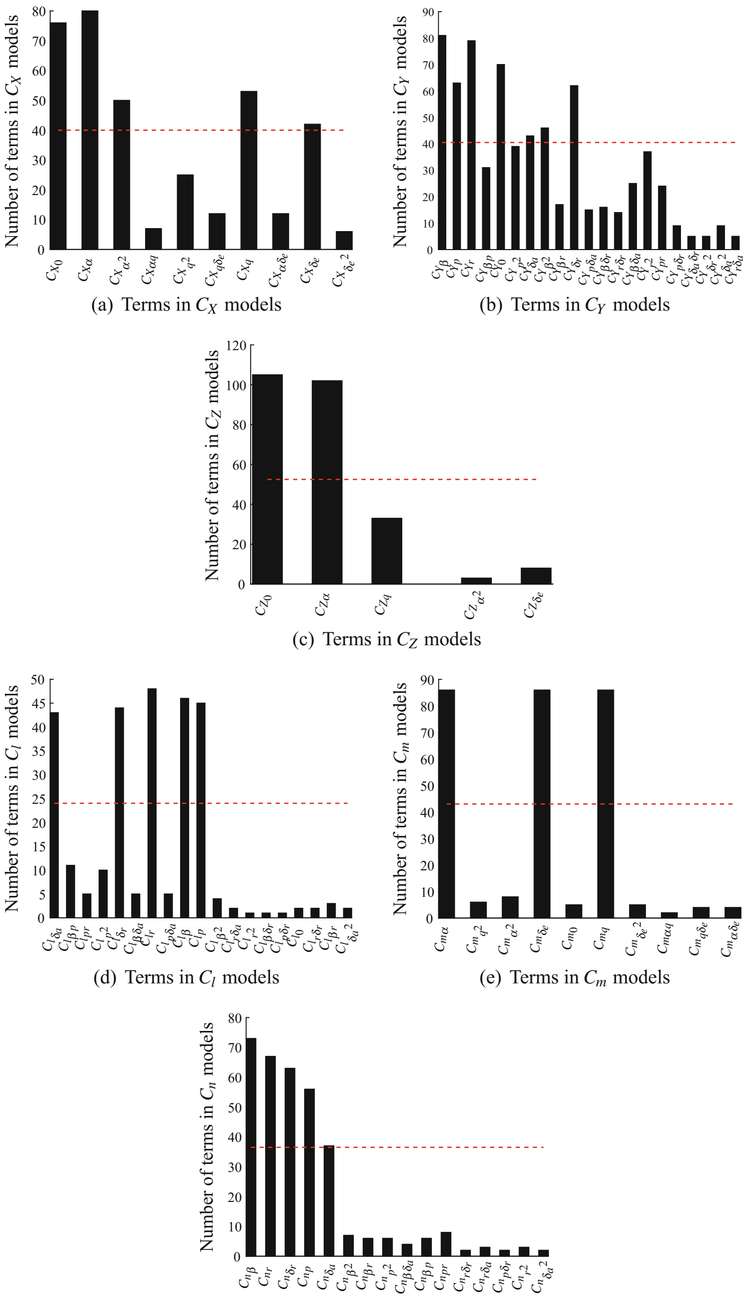
$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\hat{q} + C_{m_{\delta_e}}\delta_e \tag{41}$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\hat{p} + C_{n_r}\hat{r} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \tag{42}$$

5.3 Model Validation

The identified models for all six non-dimensional forces and moments were applied to an independent validation data set consisting of 20% of the total data set. A comparison between the longitudinal forces and moments obtained from flight data, predicted by the old DASMAT model, and the new model are shown in Fig. 5. The results for the lateral forces and moments are qualitatively similar. In addition, fit statistics in terms of the coefficient of determination and the relative root mean square error (RRMSE) are tabulated in Table 8.

A time-domain comparison between the new least-squares model and DASMAT for a longitudinally induced 3-2-1-1 maneuver is presented in Fig. 6. This figure



(a) Terms in C_X models

(b) Terms in C_Y models

(c) Terms in C_Z models

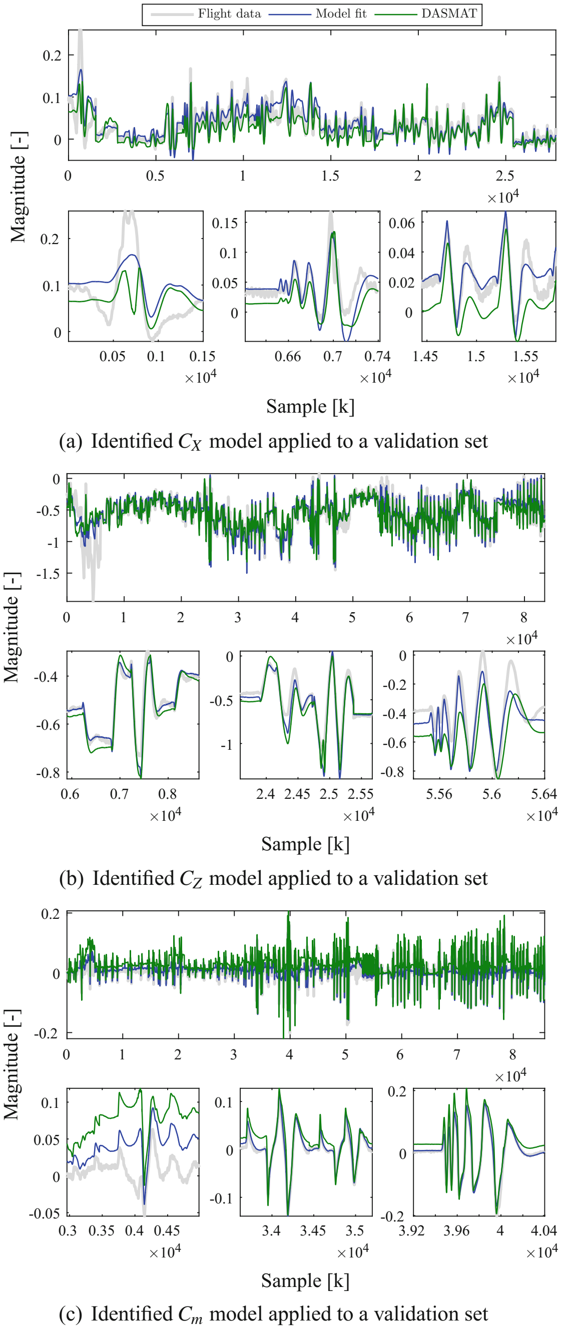
(d) Terms in C_l models

(e) Terms in C_m models

(f) Terms in C_n models

Fig. 4 Absolute number of model terms selected in the longitudinal and lateral models obtained from an orthogonal least squares model structure selection procedure

Fig. 5 The identified models for the longitudinal non-dimensional forces and moments (*blue*) applied to validation data, consisting of 20% of the total data collection, in comparison with the currently implemented aerodynamic model in the DASMAT simulation framework (*green*) and the flight derived non-dimensional forces and moments (*grey*). For every model, three close-ups of interesting portions of the complete validation set are presented below each subplot



(a) Identified C_X model applied to a validation set

(b) Identified C_Z model applied to a validation set

(c) Identified C_m model applied to a validation set

Table 8 Fit statistics for the least squares model and the existing DASMAT (D) model averaged over all validation sets

	C_X	C_Y	C_Z	C_l	C_m	C_n
R^2	0.76	0.77	0.77	0.75	0.76	0.85
R_D^2	0.60	0.55	0.64	0.25	0.00	0.50
$RRMSE$ (%)	6.76	5.32	6.38	4.96	5.8	4.72
$RRMSE_D$ (%)	8.79	7.34	7.97	8.65	12.65	8.50

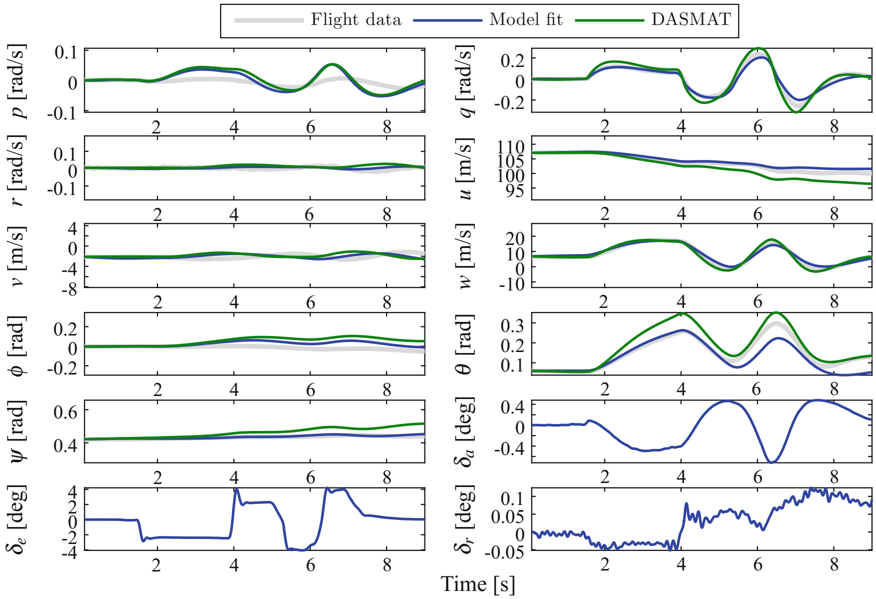


Fig. 6 Time domain response of the newly implemented aerodynamic model together with the currently implemented aerodynamic model in the DASMAT simulation framework and the flight derived aircraft states and control surface deflections for a longitudinally induced δ_e 3-2-1-1 maneuver

indicates an increased fidelity of the predicted aircraft states by the new least-squares model in comparison to the DASMAT model. Most significant is the better fit of the new model for the velocity in the direction of the X_b axis and the Euler angles.

6 Conclusion

In this paper, the methodology regarding the identification of an aerodynamic model for flight simulation training from flight test data was developed for the normal post-stall flight envelope. By employing the Two-Step Method (TSM), the Unscented

Kalman Filter (UKF) was used in cooperation with linear parameter estimation techniques. Results indicate that the state estimates and measurement reconstructions by the UKF are in good agreement with the presented data.

This research effort results in a simple and parsimonious set of aerodynamic models describing the 6 non-dimensional forces and moments. The model presented in this paper outperforms the current aerodynamic model implemented in the DASMAT framework in terms of goodness of fit, in all 6 degrees of freedom, when compared to the recorded forces and moments of the Cessna Citation II laboratory aircraft. The explained variance of the non-dimensional forces was increased with at least 13%. More significant improvements were made to the non-dimensional moments; an increase of the explained variance of at least 35% was achieved.

The work presented in this paper will serve as a basis for the integration of a stall and post-stall model, resulting from a parallel research effort. Together, these models will be used in future research into, e.g., the behavior of pilots during aerodynamic stall and the development of new control algorithms.

References

1. Federal Aviation Administration, Qualification, Service, and Use of Crewmembers and Aircraft Dispatchers, Technical Report, Department of Transportation, US, 2013
2. van der Linden CAAM (1998) DASMAT - Delft University Aircraft Simulation Model and Analysis Tool. Delft University Press, Delft
3. Mulder JA, Baarspul M, Breeman JH, Nieuwpoort AMH, Verbaak JPF, Steeman PSJM (1987) Determination of the Mathematical Model for the New Dutch Government Civil Aviation Flying School Flight Simulator, Society of Flight Test Engineers, 18th Annual Symposium. Delft University of Technology, Amsterdam
4. Oliveira J, Chu QP, Mulder JA, Balini HMNK, Vos WGM (2005) Output error method and two step method for aerodynamic model identification. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, pp 1–9, August, 2005
5. Nahi NE (1969) Estimation Theory and Applications. John Wiley and Sons, New York
6. Mulder JA, Chu QP, Sridhar JK, Breeman JH, Laban M (1999) Non-linear aircraft flight path reconstruction review and new advances. Prog Aerosp Sci 35(7):673–726
7. Mulder JA, Sridhar JK, Breeman JH (1994) Identification of Dynamic Systems- Applications to Aircraft Part 2: Nonlinear Analysis and Manoeuvre Design, vol 3. North Atlantic Treaty Organisation, Neuilly Sur Seine
8. Julier SJ, Uhlmann JK (1997) A New Extension of the Kalman Filter to Nonlinear Systems. International Symposium for Aerospace Defense Sensing Simulutation and Controls 3(2):26
9. Cessna Aircraft Company (1990) Operating Manual Model 550 Citation II, Unit -0627 And On. Technical Report, Wichita, Kansas, USA
10. Laban M (1994) On-Line Aircraft Aerodynamic Model Identification, Ph.d. thesis, Delft University of Technology, Delft
11. de Visser CC (2011) Global Nonlinear Model Identification with Multivariate Splines, Ph.D. thesis, Delft University of Technology, Delft
12. Mulder M, Lubbers B, Zaal PMT, van Paassen MM, Mulder JA (2009) Aerodynamic hinge moment coefficient estimation using automatic fly-by-wire control inputs. In: Proceedings of the AIAA modeling and simulation technologies conference and exhibit, chicago (IL), No 2, pp 2009–5692

13. Chowdhary G, Jategaonkar R (2010) Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter. Aerosp Sci Technol 14(2):106–117
14. Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. Proc IEEE 92(3):401–422
15. Wan EA, Van Der Merwe R (2000) The unscented kalman filter for nonlinear estimation, adaptive systems for signal processing, communications, and control symposium 2000. AS-SPCC IEEE 2002:153–158
16. Van Der Merwe R, Wan EA (2001) The square-root unscented Kalman filter for state and parameter-estimation. In: Proceedings. (ICASSP '01). 2001 IEEE international conference on acoustics, speech, and signal processing, vol 6, pp 3461–3464
17. van der Merwe R, Wan EA (2004) Sigma-Point Kalman Filters for integrated navigation. In: Proceedings of the 60th annual meeting of the institute of navigation (ION), pp 641–654
18. Sibley G, Sukhatme GS, Matthies LH (2006) The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo, *Rss*
19. Armesto L, Tornero J, Vincze M (2008) On multi-rate fusion for non-linear sampled-data systems: Application to a 6D tracking system. Robot Auton Syst 56(8):706–715
20. Klein V (1989) Estimation of aircraft aerodynamic parameters from flight data. Prog Aerosp Sci 26(1):1–77
21. Morelli EA (1998) Global nonlinear parametric modelling with application to F-16 aerodynamics. In: Proceedings of the american control conference, vol 2, pp 997–1001
22. Morelli E, Derry SD (2005) Aerodynamic parameter estimation for the X-43A (Hyper-X) from flight data. In: AIAA atmosperic flightmechanics conference and exhibit, August 2005
23. Goldberg MA, Cho HA (2003) Introduction to Regression Analysis. WIT Press, Southampton, UK
24. Watson PK, Teelucksingh SS (2002) A Practical Introduction to Econometric Methods: Classical and Modern. University of the West Indies Press
25. Morelli EA (2006) Practical aspects of the equation-error method for aircraft parameter estimation. AIAA Atmos Flight Mech Conf 6114:1–18
26. Batterson JG, Klein V (1989) Partitioning of flight data for aerodynamic modeling of aircraft at high angles of attack. J Aircr 26(4):334–339
27. Klein V, Batterson JG, Murphy PC (1981) Determination of Airplane Model Structure from Fflight Data by Using Modified Stepwise Regression. Technical Report, NASA Langley Research Center, Hampton
28. Lombaerts T, Oort EV, Chu QP, Mulder JA, Joosten D (2010) Online aerodynamic model structure selection and parameter estimation for fault tolerant control. J Guid Control Dyn 33(3):707–723
29. Morelli E (2012) Efficient global aerodynamic modeling from flight data. In: 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. aerospace sciences meetings, american institute of aeronautics and astronautics Jan 2012
30. Grauer JA, Morelli EA (2014) Generic global aerodynamic model for aircraft. J Aircr 52(1):13–20

Characterising Angular Accelerometer Calibration Setup Disturbance Using Box–Jenkins Method

D. Jatiningrum, C.C. de Visser, M.M. van Paassen, E. van Kampen and M. Mulder

1 Introduction

Angular acceleration feedback is a powerful concept when designing a model-free, state-of-the-art Fault-Tolerant Flight Control (FTFC) system. Such feedback is currently being obtained by differentiating the rate gyro signals [1], which amplifies the noise component and introduces delay [2]. Considering feedback is a valuable tool to adjust the performance and to ultimately meet the desired result, the feedback signal accuracy will determine the system's successful control actions.

To incorporate the angular acceleration feedback in the FTFC system design, analysis and simulation, a proper angular accelerometer model is of substantial concern. Creating a controller based on an incorrect angular accelerometer, for example, could result in the control not meeting the desired control variables. Therefore, establishing a proper angular accelerometer system identification method is crucial to achieving the correct model.

The frequency response data generates the current angular accelerometer model in a transfer function form [3]. Frequency domain data provides the required accuracy to calculate the Frequency Response Function (FRF) in this approach. Specifying the

D. Jatiningrum (✉) · C.C. de Visser · M.M. van Paassen · E. van Kampen · M. Mulder
Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1,
2629HS Delft, The Netherlands
e-mail: D.Jatiningrum@tudelft.nl

C.C. de Visser
e-mail: C.C.deVisser@tudelft.nl

M.M. van Paassen
e-mail: M.M.vanPaassen@tudelft.nl

E. van Kampen
e-mail: E.vanKampen@tudelft.nl

M. Mulder
e-mail: M.Mulder@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_15

sensor bandwidth as well as recognising the system's phase are the additional advantages of this method. The Auto-Regressive with exogenous terms (ARX) model is employed as the first stage estimation due to its efficiency in estimating the polynomial because it is the result of solving linear regression equations in analytic form. Moreover, the result is distinctive; it always fulfils the global minimum of the loss function. The ARX model, therefore, is favoured, especially when the model order is high.

However, the coupling between the deterministic and stochastic dynamics can distort the estimation of the ARX model when the disturbance of the system is not white noise. The model order is set higher to minimise the equation error, especially for a low signal-to-noise ratio. Nevertheless, since the ARX model defines both the system dynamics and noise properties using the similar set of poles, the resulting model order may be needlessly high, which can alter some dynamic aspects of the model such as the stability.

This study establishes an angular accelerometer model that considers the calibration measurement setup noise in a separate term. Errors may exist in the calibration table (CT) regarding installation error, wobble error and angular error [4], as well as sensor quantisation error [5], which influence the generated model noise and decrease accuracy. The search for a suitable model to better describes the relationship employs the Box–Jenkins (BJ) structure [6], which provides a complete model with disturbance properties independently parametrised from system dynamics as rational functions [7]. Box–Jenkins analysis refers to a systematic method of identifying, fitting, checking, and using integrated autoregressive, moving average (ARIMA) time series models,[1] with numerous application in the different fields [8–10]. The Box–Jenkins model is useful when one has disturbances that enter later in the method, such as, measurement noise on the output. Measurements performed throughout the frequency range using sinusoidal excitation at each frequency point, resulting in some input-output data sets. The process then carries out a multi-channels modelling approach to obtain the concatenated model.

The paper is divided into five sections, opening with an introduction to the topic. Section 2 contains background knowledge regarding the measurement setup and the BJ model properties. Section 3 describes the experiment plan including the system identification approach. Section 4 discusses the transfer function model development as well as its performance criteria evaluation. Finally, Sect. 5 gives final remarks on the angular accelerometer BJ model for sensor-based FTFC system application.

2 Modelling the Disturbance

A 2-axis, high precision position CT generates the excitation input for the angular accelerometer. The CT only measures the angular displacement, from where the angular rate and angular acceleration of the motion are derived. The derivation means

[1] http://www.ncss.com/software/ncss/time-series-and-forecasting-in-ncss/#ARIMA.

that there is a delay present not only in the CT estimated angular acceleration but also between the input motion and the angular accelerometer measurement data. Such delay is expected to have an impact on the identified model.

2.1 Angular Accelerometer and Calibration Table Measurement Set-Up

Figure 1 illustrates the angular accelerometer and CT measurement setup in a block diagram that is formulated the previous work [11]. Five subsystems construct the framework: (I) the control centre, (II) the turntable, (III) the sensor, and (IV) the sensor data acquisition system. Table 1 lists the blocks, variables, and disturbances involved.

In modelling the sensor using this setup, several types of disturbances are present. Subsystem II includes motor related noise m from the turntable axis' motor. Load disturbance d from the payload package, external or environmental disruption t, and sensor noise s, affect the sensor output in subsystem III. Lastly, measurement noise n is a part of recording in subsystem IV.

Holding that the CT is position-based, the angular accelerometer calibration would be inclined to model the sensor by referring to position feedback f or position estimate p_1. These references ensure precise and accurate standard. Nevertheless, they contain all the mentioned disturbances in the preceding paragraph. A more perceptive angle is to designate the CT angular acceleration a_1 or p_2 instead for an equivalent input-output relation. Even so, this system will still include d, t, s, and n in the system dynamics but rule out m, at the very least.

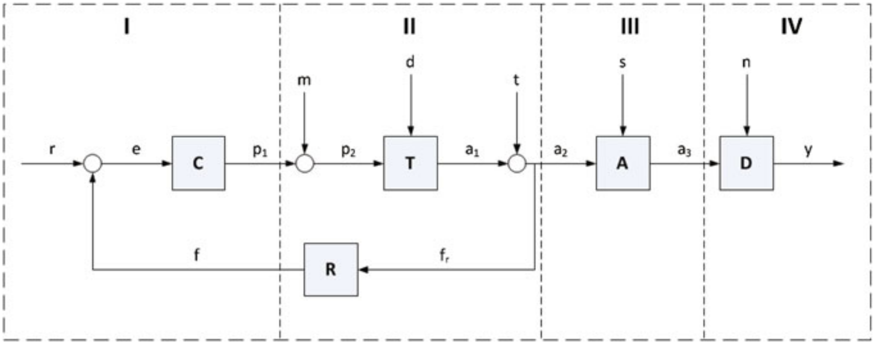


Fig. 1 Angular accelerometer calibration measurement setup framework [11]

Table 1 List of blocks, variables and disturbances of the system block diagram

Blocks		Variables		Disturbances	
C	Control center	r	User input or position demand	m	Motor related noise
T	Turntable	e	Error	d	Load disturbance
R	Position transducer	p_1	Position estimate	s	Sensor noise
A	Angular accelerometer	p_2	Position command	t	External/environmental disturbance
D	Sensor data acquisition system	a_1	Generated angular acceleration	n	Measurement noise
		a_2	Angular acceleration motion		
		a_3	Angular accelerometer raw data		
		y	Sampled angular accelerometer data		
		f_r	Raw position feedback		
		f	Position feedback		

2.2 Approach to Model the Disturbance

Consider that the angular accelerometer is an unknown system; various parametric black-box model structures apply to represent the sensor. In this regard, rational transfer functions are primarily the immediate means of parametrising the deterministic (G) and stochastic (H) part of the system. Among the models, the equation error or ARX is the first choice in many applications due to its valuable property where the predictors distinguish a linear regression.

The general family of discrete model structure is [12]:

$$\mathscr{A}(q)y(k) = \frac{\mathscr{B}(q)}{\mathscr{F}(q)}u(k) + \frac{\mathscr{C}(q)}{\mathscr{D}(q)}e(k) \tag{1}$$

where $u(k)$ is the input channel, $y(k)$ is the output channel, and $e(k)$ is the disturbance channel. $\mathscr{A}$, $\mathscr{B}$, $\mathscr{F}$, $\mathscr{C}$, and $\mathscr{D}$ are polynomials in the shift operator q, defined as follows:

$$\begin{aligned}
\mathscr{A}(q) &= 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_{n_a}q^{-n_a}\\
\mathscr{B}(q) &= \quad\; b_1q^{-1} + b_2q^{-2} + \cdots + b_{n_b}q^{-n_b}\\
\mathscr{C}(q) &= 1 + c_1q^{-1} + c_2q^{-2} + \cdots + c_{n_c}q^{-n_c}\\
\mathscr{D}(q) &= 1 + d_1q^{-1} + d_2q^{-2} + \cdots + d_{n_d}q^{-n_d}\\
\mathscr{F}(q) &= 1 + f_1q^{-1} + f_2q^{-2} + \cdots + f_{n_f}q^{-n_f}
\end{aligned}$$

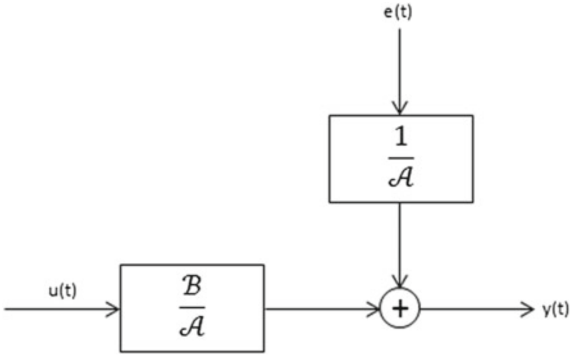


Fig. 2 The ARX model structure

In the above equations, a, b, c, d, and f are the polynomial's coefficients, whereas n_a, n_b, n_c, n_d, and n_f are the polynomials order.

The ARX model represented in Fig. 2 can be obtained from the general model by choosing $\mathscr{C}(q) = \mathscr{D}(q) = \mathscr{F}(q) = 1$, and $\mathscr{A}(q)$ and $\mathscr{B}(q)$ arbitrary polynomials

$$y(k) = \frac{\mathscr{B}(q)}{\mathscr{A}(q)}u(k) + \frac{1}{\mathscr{A}(q)}e(k) \tag{2}$$

The drawback of the ARX model is that disturbances are part of the system dynamics. The transfer function of the deterministic and stochastic part of the structure has the similar pairs of poles. This coupling can be nonsensical since they do not necessarily share the same array of poles. Although, a good signal-to-noise ratio can reduce this disadvantage.

The Box–Jenkins methodology[2] is a five-step method for identifying, selecting, and assessing conditional mean models that valid for discrete, univariate time series data. The steps start with establishing the stationarity of the time series by a successive difference to achieve stationarity. The example autocorrelation function (ACF) and partial autocorrelation function (PACF) of a stationary series decline exponentially or cut off entirely after a few lag. Subsequently, some formulations transform the series by subtracting the mean of the set from each data point. The subtraction yields a conditional mean model for the data or a series with a mean of zero. Specifying the model and estimating the model parameters are the third step. The models can be extended to include the desired degree of seasonal autoregressive and seasonal moving average terms. Next, conduct goodness-of-fit checks to ensure the model describes the data adequately. This limit means residuals should be uncorrelated, homoscedastic, and normally distributed with constant mean and variance. Afterwards, use the model to forecast or generate simulations over an expected time extent.

[2] http://nl.mathworks.com/help/econ/box-jenkins-methodology.html, accessed on April 2016.

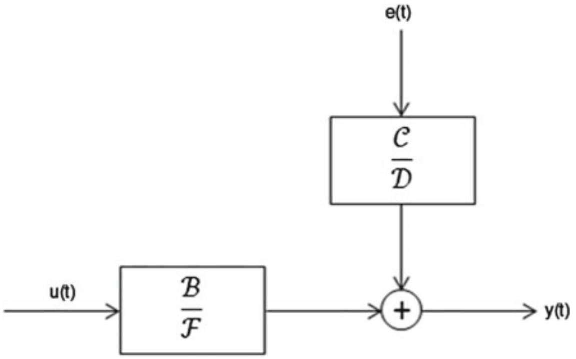

e(t)
$\frac{\mathcal{C}}{\mathcal{D}}$
u(t)
$\frac{\mathcal{B}}{\mathcal{F}}$
+
y(t)

Fig. 3 The BJ model structure

The Box–Jenkins model structure is illustrated in Fig. 3; it can be derived from Eq. 1 by choosing $\mathscr{A}(q) = 1$, and the other polynomials arbitrarily

$$y(k) = \frac{\mathscr{B}(q)}{\mathscr{F}(q)}u(k) + \frac{\mathscr{C}(q)}{\mathscr{D}(q)}e(k) \tag{3}$$

3 Experiment and Model Identification Method

The experiment is conducted using the frequency response test method for the angular accelerometer, which covers its dynamic range. CT is capable of introducing the relevant frequency content since it typically includes a programmable servo motor and an optical encoder that supports programmed revolution on the motor shaft. Nonetheless, mechanical limitations and resolution concerns encourage the use of a detailed motion profile to excite the angular accelerometer.

The followings are the method of this study in developing the Box–Jenkins model for the angular accelerometer.

1. Frequency response measurement.
   The motion profile in the plan is a single frequency sinusoid, executed in different frequency points over the system's dynamic range from 0.4 to 10 Hz. Frequency points are built in logarithmic spacing, with the increment of 20% from the previous incidence. Additionally, the measurements include round frequency values from 1 to 10 Hz. The excitation produces a constant angular acceleration, with 400 deg/s^2 or 70% of the maximum sensor range chosen as the maximum amplitude of the profile.
   As required for estimating Box–Jenkins model, the measured data set consist of input-output pairs in the time domain for each test frequency points. The method is suitable for time series of medium to long length, which is categorised as at

least 50 observations. In this case, the recorded pairs are resolved well above the minimum, with 2^{17} data points.

2. The selection of identification and validation data.
   The data sets are divided into two groups based on the test frequency sets, one for identification and the other one for validation. The identification process uses fewer data than the validation. This arrangement yields 9 data sets of round frequencies for identification and 19 data sets in the increment of 20% ranged from 0.4 to 10 Hz for validation.
3. Box–Jenkins model iteration.
   In this paper, two inputs, namely angular acceleration command and angular acceleration estimate, are compared. Their description in the calibration measurement framework leads to different delays between the reference and sensor data. The estimated delay between input and output signal are first calculated accordingly. Estimate of the delay via cross-correlation returns $nk_c = 30$ samples (angular acceleration command as the input), and $nk_e = 9$ samples (angular acceleration estimate as the input).
   The model's estimation is then treated as a multi-experiments problem. This treatment means that the estimation procedure is applied to multiple experiments data at once, to obtain the preferred model(s). The initial condition option was set to 'auto', the `bj` estimation then determine that it is best to use 'backcast', where the initial conditions were estimated using the best least squares fit.
   The first iteration was performed with the following options:

   - The orders n_b, n_f, n_c, and n_d are varied from 1 to 5.
   - Additional estimations are performed for $nk_c = 29$ samples, $nk_c = 31$ samples, $nk_e = 8$ samples, and $nk_e = 10$ samples.

4. Selection of candidates from initial models.
   The iteration delivers 625 early models, from which five candidate models are preferred for each nk in the two inputs. This selection is based on the Akaike Information Criterion (AIC) of each model as follows [13]:

$$\mathrm{AIC} = 2k - 2\log L \tag{4}$$

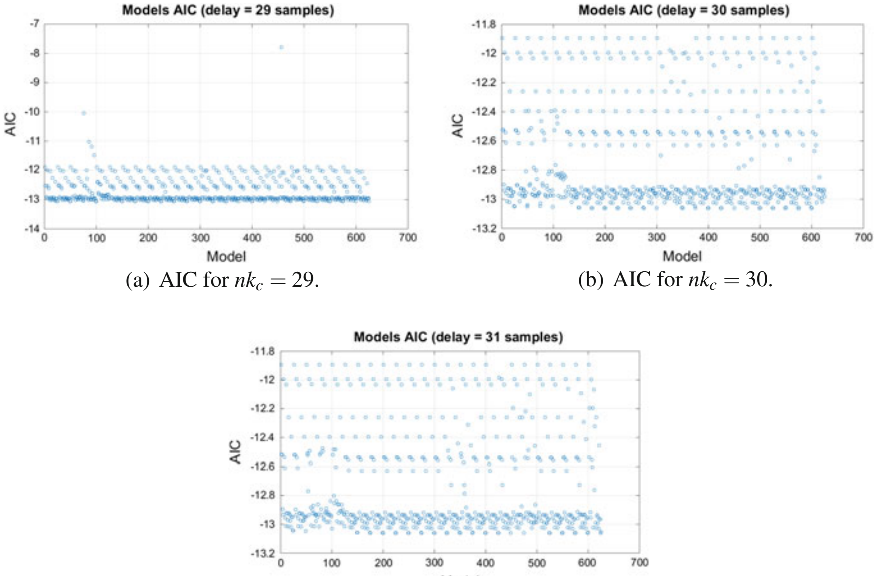
   where k is the model's number of parameters, and L is the maximised value of the likelihood function for the model. Akaike's theory confers that the most accurate model has the smallest AIC [7].
5. Comparing candidate models performance.
   Selecting the 'optimal' order from the candidate models is one of the fundamental and most subjective tasks in the Box–Jenkins analysis. In addition to the AIC, evaluation criteria are extended to Mean Squared Error (MSE), Final Prediction Error (FPE), and ultimately the percentage (%) of fit between the models and the reference data.

4 Box–Jenkins Model for Angular Accelerometer

This section analyses the iteration result in step 3 of Sect. 3, based on the measurement data sets and a group from step #1 and #2. Subsequently, step #4 and #5 will be discussed.

4.1 *Select Candidates from Initial Models Using the Akaike Information Criterion*

The first examination is to evaluate the Akaike Information Criterion (AIC) for all initial 625 models for each nk_c and nk_e as shown in Figs. 4 and 5. There are ample models with low AIC according to the figures. Nevertheless, in this study, five models with the lowest AIC for every nk or the total of fifteen models for each input are chosen as the candidate models.

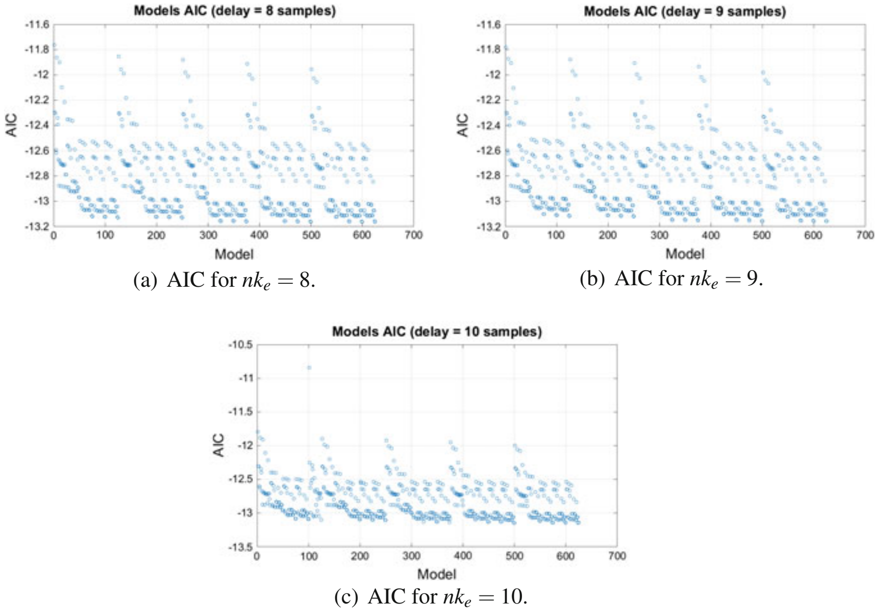


(a) AIC for $nk_c = 29$.

(b) AIC for $nk_c = 30$.

(c) AIC for $nk_c = 31$.

Fig. 4 Akaike information criterion (AIC) evaluation for all 625 initial models for each nk_c



(a) AIC for $nk_e = 8$.

(b) AIC for $nk_e = 9$.

(c) AIC for $nk_e = 10$.

Fig. 5 Akaike information criterion (AIC) evaluation for all 625 initial models for each nk_e

4.2 Evaluation Based on the Model Performance Criteria

After the candidate models were preferred, the data analysis should involve the decent tradeoff between underfitting and overfitting. Once again, AIC will play a role in ranking the models but three additional criteria: MSE, FPE and %Fit, are also included. The candidate models performances are presented in Tables 2 and 3.

The information in the table is ranked based on the lowest AIC, displaying the five best models for each nk_c. In general, most of them have an extremely close AIC and FPE values. As can be seen from the table, however, a model with lowest AIC is in fact not necessarily the model with the best fit. The MSEs, on the other hand, have slight variations as it depends on the model fit and should be considered as a related entity. Therefore, models fit are employed to select the models which gives 325 ($n_b = 3, n_c = 5, n_d = 5, n_f = 3, nk_c = 29$), 425 ($n_b = 4, n_c = 5, n_d = 5,$ $n_f = 2$, $nk_c = 30$), and 425 ($n_b = 4$, $n_c = 5$, $n_d = 5$, $n_f = 2$, $nk_c = 31$) as the best models in the group.

Similarly, models in Table 3 hold significantly close performances as well. Evaluating the options, the selected models for the angular acceleration estimate input are 600 ($n_b = 5$, $n_c = 5$, $n_d = 5$, $n_f = 4$, $nk_e = 8$), 600 ($n_b = 5$, $n_c = 5$, $n_d = 5$, $n_f = 4$, $nk_e = 9$), and 500 ($n_b = 4$, $n_c = 5$, $n_d = 5$, $n_f = 5$, $nk_e = 10$).

Table 2 Models with lowest AIC for angular acceleration command as the input

Model	Order					Fit [%]		MSE		AIC	FPE
	nb	nc	nd	nf	nk_c	Min	Max	Min	Max		
575	5	5	5	3	29	−391.9585	51.675393	0.2986674	29.252729	−13.05992	2.13E-06
550	5	5	5	2		91.12284	99.125697	9.78E-05	0.0095248	−13.05988	2.13E-06
225	2	5	5	4		98.405812	99.829739	3.71E-06	0.0003072	−13.05987	2.13E-06
325	3	5	5	3		99.546733	99.91966	8.26E-07	2.46E-05	−13.05987	2.13E-06
525	5	5	5	1		73.313107	97.377891	0.0008793	0.0860806	−13.05986	2.13E-06
499	4	5	4	5	30	86.706018	98.692406	0.0002187	0.0213609	−13.06036	2.13E-06
575	5	5	5	3		−14.21239	88.781457	0.0160962	1.5766521	−13.06012	2.13E-06
500	4	5	5	5		67.337535	96.790815	0.0013172	0.1289458	−13.06012	2.13E-06
425	4	5	5	2		97.391019	99.73503	8.98E-06	0.0008227	−13.05988	2.13E-06
525	5	5	5	1		74.017156	97.446941	0.0008336	0.0815986	−13.05987	2.13E-06
575	5	5	5	3	31	93.283216	99.336853	5.62E-05	0.005453	−13.05989	2.13E-06
550	5	5	5	2		74.808787	97.524675	0.0007836	0.0767022	−13.05988	2.13E-06
425	4	5	5	2		96.855372	99.683916	1.28E-05	0.0011952	−13.05988	2.13E-06
525	5	5	5	1		84.963562	98.521667	0.0002795	0.0273275	−13.05988	2.13E-06
400	4	5	5	1		65.455704	96.606236	0.001473	0.1442322	−13.05987	2.13E-06

Table 3 Models with lowest AIC for angular acceleration estimate as the input

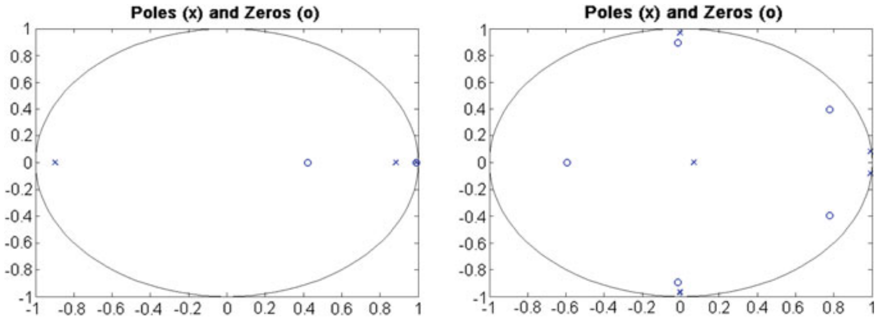
Model	Order					Fit [%]		MSE		AIC	FPE
	nb	nc	nd	nf	nk_e	Min	Max	Min	Max		
625	5	5	5	5	8	99.813035	99.936639	5.13E-07	4.19E-06	−13.16405	1.92E-06
500	4	5	5	5		99.813001	99.936552	5.15E-07	4.19E-06	−13.16401	1.92E-06
600	5	5	5	4		99.813042	99.936665	5.13E-07	4.19E-06	−13.16391	1.92E-06
499	4	5	4	5		99.812388	99.93683	5.10E-07	4.22E-06	−13.16202	1.92E-06
599	5	5	4	4		99.812427	99.936947	5.08E-07	4.22E-06	−13.1619	1.92E-06
500	4	5	5	5	9	99.812994	99.933947	5.58E-07	4.19E-06	−13.15895	1.93E-06
625	5	5	5	5		99.812996	99.934054	5.56E-07	4.19E-06	−13.15883	1.93E-06
600	5	5	5	4		99.813013	99.933948	5.58E-07	4.19E-06	−13.1582	1.93E-06
499	4	5	4	5		99.812392	99.934304	5.52E-07	4.22E-06	−13.15715	1.93E-06
624	5	5	4	5		99.812392	99.934332	5.52E-07	4.22E-06	−13.1571	1.93E-06
500	4	5	5	5	10	99.812751	99.930749	6.13E-07	4.20E-06	−13.15027	1.94E-06
499	4	5	4	5		99.812186	99.930968	6.09E-07	4.23E-06	−13.14881	1.95E-06
600	5	5	5	4		99.812705	99.93052	6.17E-07	4.20E-06	−13.14843	1.95E-06
624	5	5	4	5		99.812176	99.931629	5.98E-07	4.23E-06	−13.14806	1.95E-06
575	5	5	5	3		99.812583	99.931277	6.04E-07	4.21E-06	−13.14447	1.96E-06

4.3 Poles and Zeros Plot of the Selected Models

Figures 6, 7 and 8 present the poles and zeros plot of selected models for each nk_c, and Figs. 9, 10 and 11 for each nk_e. Figures on the left-hand side are for the input-output, whereas images on the right-hand side are the noise output. The pole-zero plot for a discrete-time model represents a stable system if the poles are inside the unit circle, whereas the zeros can be inside or outside of the unit circle [14].

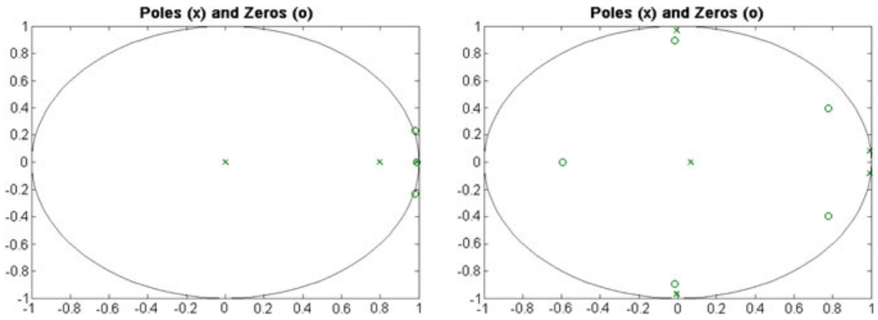
4.4 Bode Plot of the Selected Models

The following Fig. 12 shows the Bode plot, where the three selected models of angular acceleration command are displayed. The Bode plot of angular acceleration estimate models is presented in Fig. 13. It is observed that nk_e models roughly have a higher



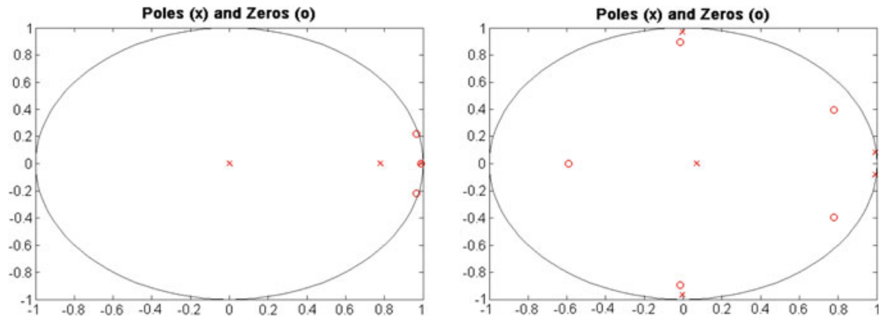
(a) $u \rightarrow y$ (b) $e@y \rightarrow y$

Fig. 6 Poles and zeros of model 325 (3553-29)



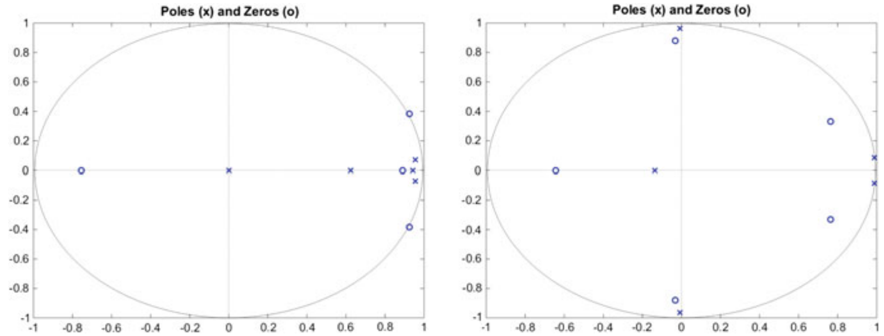
(a) $u \rightarrow y$ (b) $e@y \rightarrow y$

Fig. 7 Poles and zeros of model 425 (4552-30)



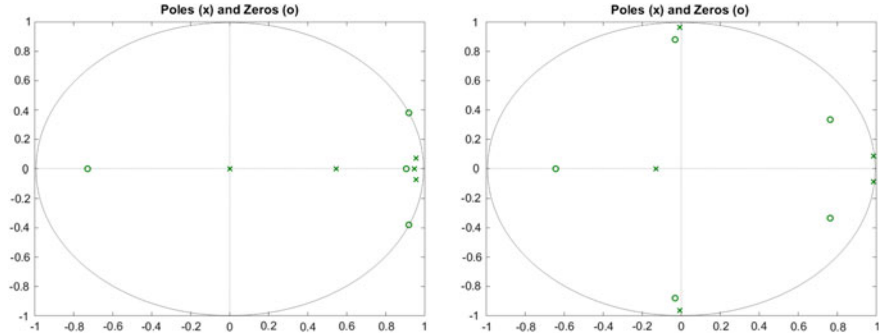
(a) $u \to y$ (b) $e@y \to y$

Fig. 8 Poles and zeros of model 425 (4552-31)



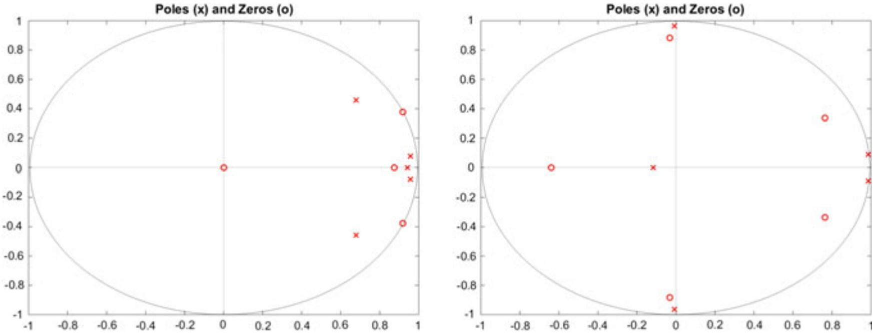
(a) $u \to y$ (b) $e@y \to y$

Fig. 9 Poles and zeros of model 600 (5554-8)



(a) $u \to y$ (b) $e@y \to y$

Fig. 10 Poles and zeros of model 600 (5554-9)



(a) $u \rightarrow y$ (b) $e@y \rightarrow y$

Fig. 11 Poles and zeros of model 500 (4555-10)

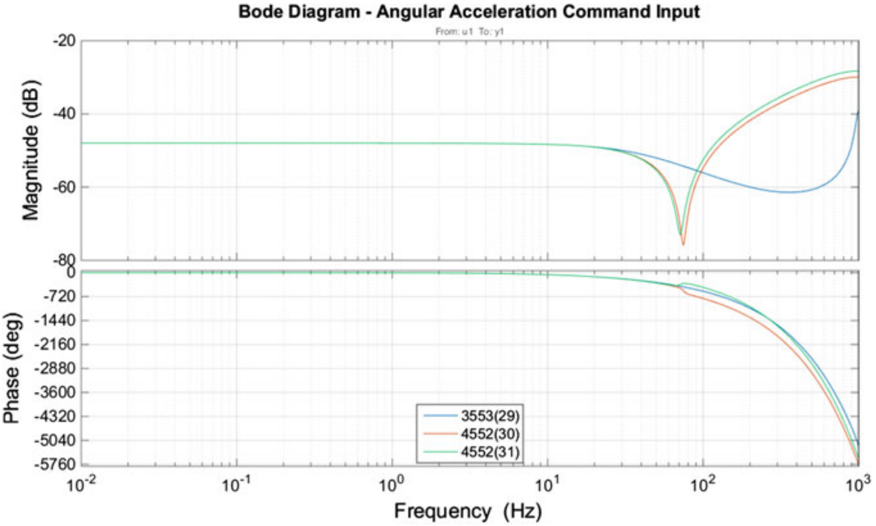


Fig. 12 Bode plot of the selected models with acceleration command as the input

order than nk_c's models. The higher order is due to the shape of the reference signal, where the angular acceleration estimate is noisier than the angular acceleration command. These two sets of identification were performed with different dynamics, based on the position of their respective input signal in the framework. Consequently, their frequency response characteristics will differ as well.

Equivalent selection could also be made for the lower order candidate models with approximately close AIC value to the preferred forms in this study. As observed in Figs. 4 and 5, many structures in this category exist at the smallest AIC value spectrum. The choice would be beneficial, for example, in the implementation of the sensor model in the control systems. While higher order model can accurately

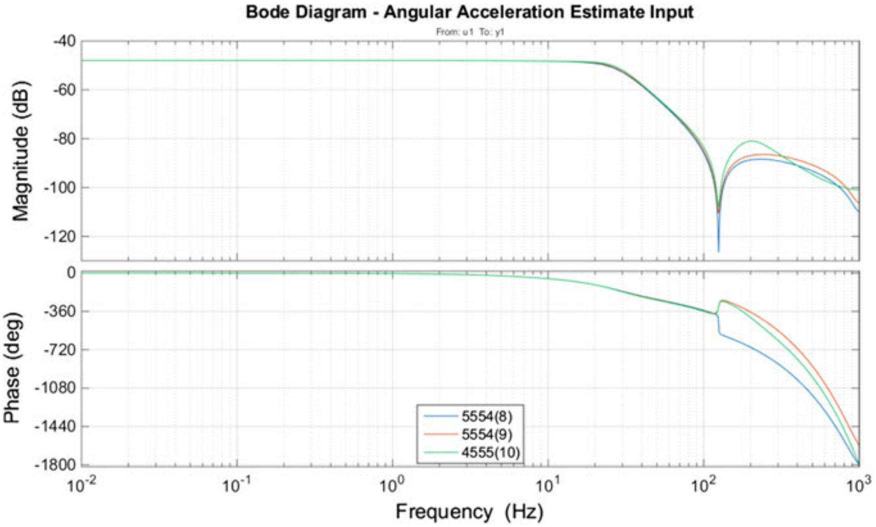


Fig. 13 Bode plot of the selected models with acceleration estimate as the input

disclose the sensor dynamics, the reduction keeps the order of the controller down without compromising the robustness of the closed loop system [15].

5 Conclusion

The Box–Jenkins method was used to estimate models for the angular accelerometer from frequency response measurement data. The model was independently parametrised to express the disturbance in the measurement system apart from the system's dynamics. Before model estimation, the angular accelerometer data were established in the stationarity, and zero mean time-series. The design iteration was performed up to the 5th order to produce a relatively straightforward model structure. Using the AIC, 15 candidate models were selected for every nk in each input types. MSE and FPE were employed as the additional evaluation criterion. Nevertheless, the $\%Fit$ concludes the selected models.

Results show that model with the lowest AIC value not necessarily the model with the best fit. With AIC values are extremely close between candidate models, the best fit was chosen to resolve the model selection since it reflects the minimum difference with the reference data. The best fit data for CT angular acceleration command as the input was found in the $n_b = 2$, $n_c = 5$, $n_d = 5$, and $n_f = 4$ order, whereas for CT angular acceleration command as the input, the best fit is in the order of $n_b = 5$, $n_c = 5$, $n_d = 5$, and $n_f = 4$.

References

1. Bacon BJ, Ostroff AJ (2000) Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation. In: Proceedings of the AIAA guidance, navigation and control conference and exhibit. American Institute of Aeronautics and Astronautics, Reston
2. Sieberling S, Chu QP, Mulder JA (2010) Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. J Guid Control Dyn 33(6):1732–1742
3. Jatiningrum D, de Visser CC, van Paassen MM, Mulder M (2016) Modelling an angular accelerometer using frequency response measurement. In: Proceedings of the AIAA guidance, navigation and control conference. American Institute of Aeronautics and Astronautics, Reston
4. Lv P, Liu J, Lai J, Zhang L (2014) Decrease in accuracy of a rotational SINS caused by its rotary table's errors. Int J Adv Robot Syst 11(74). https://doi.org/10.5772/58579
5. Savage PG (2002) Analytical modeling of sensor quantization in strapdown inertial navigation error equations. J Guid Control Dyn 25(5):833–842
6. Box GEP, Jenkins GM, Reinsel GC (1994) Time series analysis: forecasting and control, 5th edn. John Wiley & Sons, Inc., Hoboken, New Jersey
7. Ljung L (1999) System identification: theory for the user. Prentice Hall, Upper Saddle River
8. Pintelon R, Schoukens J (2006) Box-Jenkins identification revisited–part I: theory. Automatica 42:63–75
9. Pintelon R, Schoukens J (2006) Box-Jenkins identification revisited–II: applications. Automatica 42:77–84
10. Pintelon R, Schoukens J (2007) Box-Jenkins identification revisited–part III: multivariable systems. Automatica 43:868–875
11. Jatiningrum D, de Visser CC, van Paassen MM, Mulder M (2015) A framework for calibrating angular accelerometers using a motion simulator. In: Proceedings of the AIAA modeling and simulation technologies conference. American Institute of Aeronautics and Astronautics, Reston
12. van den Bosch PPJ, van der Klauw AC (1994) Modeling, identification and simulation of dynamical systems. CRC Press Inc, Boca Raton
13. Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19:716–723
14. Nise NS (2011) Control systems engineering. Wiley, Boca Raton
15. Pratt RW (2000) Flight control systems - practical issues in design and implementation. Institution of Engineering and Technology, Stevenage

Improved Hybrid Navigation for Space Transportation

Guilherme F. Trigo and Stephan Theil

1 Introduction

Navigation on-board most launchers and sounding rockets is purely inertial: measurements from strapdown accelerometers and gyroscopes are integrated from an initial condition yielding a high-rate navigation solution. While self-contained and robust, inertial navigation suffers from a fundamental drawback: drift. If uncorrected, the inertial propagation errors will boundlessly grow over time. To compensate for this, inertial sensors are made increasingly accurate, raising cost, and often also volume and mass. Still, inertial drift is unavoidable, posing important performance and operational constraints on both launchers and sounding rockets. Trajectory maintenance and injection greatly depend on navigation accuracy. Not rarely, delivered spacecraft need to perform costly orbital manoeuvres to correct for deficient injection, potentially reducing operation life-time, causing loss of commercial/scientific profit. With drift, a strong limitation is also set on launch mission duration and profile. For instance, return phases in missions of reusable vehicles simply cannot cope with the error inertial propagation yields, requiring other measurement sources [17]. In sounding rockets, with lower budget and operational requirements, when used, inertial sensors are generally of (comparatively) lower grade. Considerable uncertainties in on-board tracking information are thus typical. Moreover, the available propagated inertial attitude is often of poor quality, potentially limiting scientific payloads or forcing these to feature their own solutions.

GNSS receivers have long been used to complement inertial systems. In fact, the two sensors gracefully ease each others' flaws: the GNSS receiver bounds inertial drift, while the high-rate inertial measurements bridge the low-rate GNSS outputs [7]. This combination can be achieved in three main ways: *loosely* coupled uses

G.F. Trigo (✉) · S. Theil
GNC Systems Department, German Aerospace Center (DLR), Institute of Space Systems,
Bremen, Germany
e-mail: guilherme.trigo@dlr.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance,
Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_16

receiver navigation solution to correct the inertial solution; *tightly* coupled avoids filter cascading by using GNSS raw measurements; while *ultra-tight* (*deep* for some authors) goes even further by driving the receiver tracking loops with the inertially aided correlators outputs [7]. In this contribution we study the application of a tightly coupled GPS/INS (Inertial Navigation System) hybrid system to rocket navigation.

In a launch environment GNSS technology has a set of important vulnerabilities. As non-self-contained, it can suffer external signal disturbance and disruption. Jamming and spoofing, for instance, pose real threat to signal quality and veracity, while ionospheric scintillations can cause temporary signal loss. Additionally, signal tracking is not immune to the high-dynamics, vibration and shocks experienced in-flight. Although complete mitigation of all these risks is not possible, improved receiver design/tuning and integrity monitoring can lessen their effects. Moreover, combination with inertial sensors improves availability and robustness.

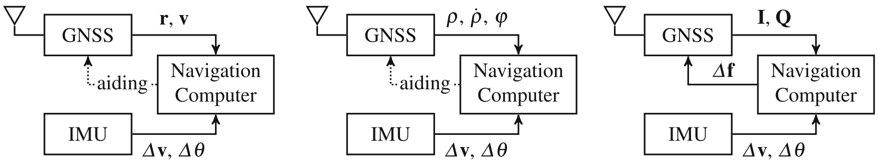
Despite the fundamental drawbacks described, which raised considerable opposition to the use GNSS technology in rocket applications, several examples of GNSS-based systems for space transportation have been conceived and tested. In the U.S., the Space Integrated GPS/INS, known as SIGI, a primary navigation system designed by Honeywell under a NASA contract, has been extensively tested including on-board of the Space Shuttle [29]. Also in the U.S., GPS metric tracking systems (GPS-MT), a class of on-board GPS-based tracking and telemetry systems, have been targeted as key technology after decomissioning of C-band radar facilities. Under this initiative, United Launch Aliance has developed a system using COTS components [2], which is routinely flown on Atlas V and Delta IV launchers as main tracking means. Other GPS-MT systems for small launchers and sounding rockets are described by Slivinsky et al. [18] and williams et al. [28]. In Russia, a hybrid INS/GLONASS/GPS primary navigation system has been tested on-board of the Fregat upper stage. After numerous flights, a 100-fold improvement was attained in orbital injection accuracy in GTO/GEO missions with respect to purely inertial performance [3]. In Europe, DLR has developed and flight-proved the Orion GPS receiver for rocket applications [13], and more recently the Phoenix-HD receiver [11]. Due to the lack of redundancy in the tracking of the European small launcher VEGA, a COTS GPS receiver is currently flown as part of the ALTS (Autonomous Localization and Telemetry Subsystem). This is planned to be coupled with a COTS inertial sensor in coming flights [5]. In 2014, as part of the OCAM-G experiment, a cooperation between ESA and European national space agencies and industry partners, a set of three GNSS receivers (including the Phoenix) flew on-board of the Ariane-V launcher [9]. Results of this experiment have been used by Braun et al. [1] to study an inertially-aided GNSS tracking system. In terms of primary navigation, an extensive study by Airbus DS with ESA funding, the HiNAV, developed a prototype GNSS/INS system for European launch and re-entry vehicles, revealing promising results [16]. Finally, DLR has conceived and successfully flown a tightly coupled GPS/INS system, the Hybrid Navigation System (HNS), on the SHEFEX2 experimental flight. Envisaged as primary navigation, the HNS uses a COTS IMU, a Phoenix receiver and a DLR-developed star sensor [20, 23].

Following the HNS development and SHEFEX2 flight, the Pseudorange-only filter updating was found to yield slow velocity estimation during engine burns [21]. As accuracy is critical during propelled flight phases, both for optimal vehicle steering and for safety monitoring, a GNSS velocity-based measurement should be included. Pseudorange-rate is a common choice, however, Time-Differenced Carrier Phase (TDCP), which is a measure of delta-range, has shown superior performance in highly dynamic applications such as missile [26] and agile UAV [30]. The time-differencing cancels out the phase ambiguity terms, avoiding the need for ambiguity fixing by differential operation or ambiguity resolution [25]. The application of TDCP observables to the rocket tightly coupled navigation problem is here investigated. Receiver clock error affecting GNSS measurements can be eliminated by channel differencing as done by Steffes [19], however, this reduces measurement availability (as it requires at least two satellites). A reduced order receiver clock model is here described, which takes into account the correlation between clock error in Pseudorange (PR) and TDCP measurements. Furthermore, Tropospheric delay-rate is shown to be a major error source in a rocket ascent scenario. A way of robustly correcting for this effect within the navigation filter is proposed.

We begin with a discussion on GNSS/INS architectures in the light of rocket navigation. An explanation on the GPS TDCP observable and the models used for receiver clock error and tropospheric delay precedes a detailed description of the hybrid navigation algorithm. The proposed system performance is then evaluated using real GNSS data.

2 GNSS/INS Hybrid Navigation Architectures

There are several ways to combine (or couple) GNSS measurements with inertial ones, from an IMU or INS. Figure 1 shows the most common coupling architectures. Excluding the *uncoupled* configuration, where INS and GNSS each independently provide a navigation solution, the simplest set-up is the *loosely* coupled (Fig. 1a). This uses the GNSS navigation solution to correct the inertial propagation through a fusion algorithm (e.g. a Kalman filter) [7]. The tightly coupled architecture (Fig. 1b), instead, uses raw GNSS measurements (e.g. Pseudorange, Pseurange-rate, Carrier Phase), allowing the navigation filter to still draw information from sets of fewer than



(a) Loose **(b)** Tight **(c)** Ultra-tight

Fig. 1 GNSS/IMU hybridization architectures

four tracked satellites. This is crucial given the tracking vulnerabilities of GNSS under rocket dynamics. A less obvious advantage of the tightly coupled scheme is the avoidance of a cascade of two filters: the GNSS receiver fix is obtained through filtering, making it heavily time-correlated with a time-varying statistical profile, which may easily lead to severe mismodelling, degrading robustness and performance [7]. In these two configurations the fused estimates can be provided as aiding to the GNSS receiver to expedite SV (re)acquisition. Finally, in an ultra-tight coupling (Fig. 1c) the receiver accumulated correlator outputs (**I** and **Q**) are fused with the inertial measurements, and fed back to drive the receiver tracking loops. This scheme achieves the best performance, however, it requires extensive access to the internal functioning of the receiver or parallel development of both receiver and hybrid system. The design complexity of such a system is thus far greater than that of the less coupled options. The tightly coupled set-up is chosen for this study for allowing the use of COTS components while still yielding strong levels of performance and robustness.

A further design option has to do with the open-/closed-loop nature of the estimated navigation corrections [7]. In the open-loop configuration the fusion algorithm estimates corrections to be applied to the inertial propagator output (position, velocity and attitude). These corrections may grow indefinitely as the inertial solution drifts. Instead, the inertial propagation may be regularly reset using the fused estimates in a closed-loop set-up. This makes sure that the filter remains close to the origin, reducing linearization errors and numerical issues due to unbounded state growth. Moreover, it allows for the inertial sensor online calibration, offering higher levels of performance and robustness. This option was adopted in this study.

Within the closed-loop set up, the level of modularity between fusion filter and inertial propagation is another design degree of freedom. If the two are independently defined, having the filter estimating error quantities, the set-up is known as *indirect* (or error-state) filtering. The *direct* (or total-state) filtering is achieved by merging the inertial algorithm and the filter propagation step. The resulting filter estimates total kinematics quantities, instead of errors or corrections. Although architecturally distinct the performances of these two set-ups can be made virtually equivalent [7]. Differences in behaviour may still arise from the way the corrections are performed, as observed by Wendel et al. [27]. The *direct* filtering scheme is, in general, more computationally intensive [27] and offers less design flexibility. Furthermore, as Steffes [22] shows, the *indirect* filtering architecture provides a simple yet powerful way to deal with measurement latency in the real-time implementation.

3 Time-Differenced Integrated Carrier Phases

The integrated carrier phase quantity is obtained within the GPS receiver through the accumulation of the phase increments from the carrier tracking loop. It is directly proportional to the range between receiver and emitting satellite, being by far the GPS raw measurement with the lowest noise level [8]. However, the initial phase value is unknown, preventing direct usage of this observable. Although this ambiguity term can be estimated using, for instance, ambiguity resolution/fixing methods in stand-alone or differential operation, allowing the integrated carrier measurement to be used as a range observable, this is not always possible or convenient.

A way to use these measurements without explicitly computing the ambiguities, is by forming Time-Differences of the integrated Carrier Phase (TDCP) measurements. Being a constant term, it is effectively cancelled out. The result is a measure of delta-range, i.e. a displacement or an average velocity. The dual-epoch nature of the delta-range measurements makes their use in Kalman filtering more complicated than the usual single-epoch measurement. Nevertheless, several methods have been successfully used in [26, 30]. In Sect. 4 one of these methods will be adapted to the navigation filter at hand.

For channel i at epoch k the TDCP is formed as

$$\begin{aligned}\lambda\Delta\tilde{\varphi}_{i,k} &= \lambda(\tilde{\varphi}_{i,k} - \tilde{\varphi}_{i,k-1}) = \left(\rho_{i,k} + \lambda \mathrm{N}_i + c\tau_{i,k}\right) - \left(\rho_{i,k-1} + \lambda \mathrm{N}_i + c\tau_{i,k-1}\right) \\ &= \Delta\rho_{i,k} + c\Delta\tau_{i,k} \end{aligned} \tag{1}$$

where λ is the carrier wave-length and c the speed of light in vacuum. Note that the phase ambiguity N_i vanishes. $\Delta\tau$ gathers the differences of the remaining (timing) errors as

$$\tau_{i,k} = \tau_{\mathsf{r},k} + \tau_{\mathsf{s},i,k} + \tau_{\mathrm{T},k,i} - \tau_{\mathrm{I},k,i} + \tau_{\mathrm{M},k,i} + \nu_{k,i} \tag{2}$$

where τ_{r} and $\tau_{\mathsf{s},i}$ are, respectively, the receiver and satellite clock offsets; $\tau_{\mathrm{T},i}$ and $\tau_{\mathrm{I},i}$ are tropospheric and ionospheric delays; $\tau_{\mathrm{M},i}$ is the delay caused by signal reflection and multi-path; and $\nu_{\rho,i}$ is measurement noise.

Satellite clock error is (mostly) removed with information from the navigation message, while multipath delta-error is generally of impulse nature, occurring in reflective conditions (e.g. on the launch pad). The profile and effect of the receiver clock error and tropospheric delay are discussed in the following points.

Receiver Clock and Common-Mode Error

The common mode error is one that affects all channels equally. This is generally attributed to the receiver clock which is often modelled as a second order system [4] as

$$\begin{bmatrix}\dot{\tau}_{\mathsf{r}} \\ \ddot{\tau}_{\mathsf{r}}\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\tau_{\mathsf{r}} \\ \dot{\tau}_{\mathsf{r}}\end{bmatrix} + \mathbf{w}_{\tau_{\mathsf{r}}} \ , \tag{3}$$

where $\mathbf{w}_{\tau_{\mathsf{r}}} \sim N\left(\mathbf{0}, \mathbf{Q}_{\tau_{\mathsf{r}}}\right)$. This system is then discretised, with noise model obtained, for instance, through an Allan variance fit as described in [24].

The receiver used in the present study, a DLR Phoenix-HD unit, uses the same internal clock model to correct both pseudorange and integrated carrier phase measurements. The TDCP common-mode error can be seen as an average clock frequency error over the differencing interval Δt (here 1 s), i.e.

$$\Delta\tau_{\mathsf{r},k} = \tau_{\mathsf{r},k} - \tau_{\mathsf{r},k-1} \approx \Delta t\,\dot{\tau}_{\mathsf{r},k} \ . \tag{4}$$

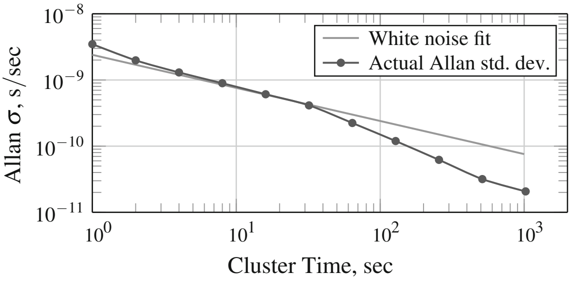


Fig. 2 Clock frequency error Allan std. dev. for a DLR Phoenix-HD receiver

Figure 2 shows the Allan standard deviation of the clock frequency error of a DLR Phoenix-HD receiver during a static test. This was obtained from pseudoranges preprocessed to extract the remaining errors. It is clear that the frequency error is dominated by white noise, leading the clock phase (or bias) to approximately behave as random walk. The receiver clock model used in the navigation filter can then be

$$\text{Clock-bias state process:} \qquad \tau_{\mathrm{r},k+1} = \tau_{\mathrm{r},k} + \mathrm{w}_{\tau_{\mathrm{r}},k} \ , \tag{5}$$

$$\text{TDCP measurement:} \qquad \mathbf{y}_k = \boldsymbol{h}_k(\mathbf{x}_k) + \mathbf{1}c\Delta\tau_{\mathrm{r},k} + \boldsymbol{\nu}_k \ , \tag{6}$$

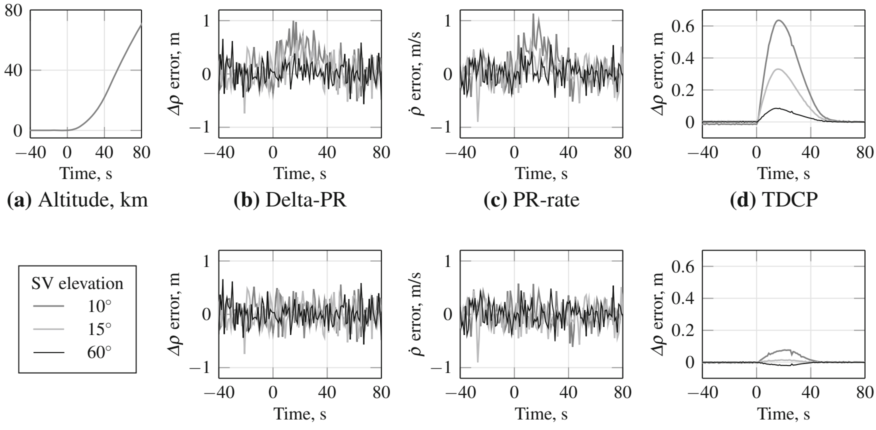
$$\text{with} \quad \mathrm{w}_{\tau_{\mathrm{r}},k} \sim N\left(0, \sigma^2_{\tau_{\mathrm{r}},k}\right) \ , \quad \Delta\tau_{\mathrm{r},k} \sim N\left(0, \sigma^2_{\Delta\tau_{\mathrm{r}},k}\right) \ , \quad \boldsymbol{\nu}_k \sim N\left(\mathbf{0}, \mathbf{R}_{\nu,k}\right) \ , \tag{7}$$

$$\text{and} \qquad E\left\{\mathrm{w}_{\tau_{\mathrm{r}},k}\, \Delta\tau_{\mathrm{r},k+1}\right\} = \sigma_{\tau_{\mathrm{r}},k}\, \sigma_{\Delta\tau_{\mathrm{r}},k+1} = \sigma^2_{\tau_{\mathrm{r}},k} = \sigma^2_{\Delta\tau_{\mathrm{r}},k} \ . \tag{8}$$

In Eq. (6), **1** is a column vector of *ones*. The correlation (8) between clock process noise and TDCP common-mode error will be taken into account upon derivation of the navigation filter. Note that the fact that the clock drift is modelled as white noise avoids the need for one extra filter state, easing computational requirements.

Tropospheric Delay Effect

Refraction of GNSS signals in the Troposphere introduces a delay that depends on signal travel path and atmospheric conditions. For a land-based receiver this delay is fairly constant, affecting mostly position estimation. In a rocket application, the most important effect is not in position (though it is still affected) but in velocity. In fact, the ascending rocket motion through the atmosphere makes the tropospheric delay on each channel change quickly, yielding a hump-like error in range-rate (or delta-range), i.e. in velocity information. While in pseudorange (PR) and pseudorange-rate (PR-rate) this effect is mostly buried in measurement noise, in TDCP the considerably lower (single-channel) noise renders this effect flagrant. Figure 3b–d show this effect on such observables (from SVs at different elevations) measured with a DLR Phoenix-HD GPS receiver stimulated by a GSS7700 SPIRENT GNSS emulator, which runs a NATO STANAG troposphere model [14]. The altitude profile followed is shown in Fig. 3a.



(a) Altitude, km **(b)** Delta-PR **(c)** PR-rate **(d)** TDCP

(e) Corrected Delta-PR **(f)** Corrected PR-rate **(g)** Corrected TDCP

Fig. 3 Error in delta-PR, PR-rate and TDCP (at 1 Hz) during rocket ascent, with and without tropospheric correction (see model in Appendix). Lift-off at $t = 0\,\mathrm{s}$

The tropospheric delay-rate should be corrected or accounted for in filter tuning, or the error in velocity estimation may surpass covariance bounds during this phase. This can worsen strapdown divergence in case of signal outage, which is rather likely during this flight leg given the violent dynamic events that may then occur. Many algorithms exists for tropospheric delay compensation. Most make use of atmospheric data either measured on-site or looked-up from tables of average regional/global values [10, 15]. However, even with such correction the residual error in TDCP may still be considerably higher than the measurement noise level (Fig. 3e–g). Therefore, for filter robustness, the measurement error covariance should account for the presence of such tropospheric residual.

Assume that the true tropospheric delay and delta-delay in channel i can be given in terms of the modelled quantities as[1]

$$\tau_{\mathrm{T},i,k} = (1 + s_{\mathrm{T}})\; h_{\tau_{\mathrm{T}},i,k}(\mathbf{x}_k) \tag{9}$$

$$\Delta\tau_{\mathrm{T},i,k} = (1 + s_{\mathrm{T}})\Delta h_{\tau_{\mathrm{T}},i,k}(\mathbf{x}_k) = (1 + s_{\mathrm{T}})\left(h_{\tau_{\mathrm{T}},i,k}(\mathbf{x}_k) - h_{\tau_{\mathrm{T}},i,k-1}(\mathbf{x}_k)\right)\;, \tag{10}$$

where $s_{\mathrm{T}} \sim N(0, \sigma_{s_{\mathrm{T}}}^2)$ is a constant unknown scale factor. $h_{\tau_{\mathrm{T}},i,k}(\mathbf{x}_k)$ is the tropospheric delay correction model function.

The covariance of the post-correction residual vectors, $\boldsymbol{\delta\tau}_{\mathrm{T},k}$ and $\boldsymbol{\delta\Delta\tau}_{\mathrm{T},k}$, is

$$\begin{aligned} E\left\{\boldsymbol{\delta\tau}_{\mathrm{T},k}\boldsymbol{\delta\tau}_{\mathrm{T},k}^{\mathsf{T}}\right\} &= E\left\{\left(\boldsymbol{\tau}_{\mathrm{T},k} - \hat{\boldsymbol{\tau}}_{\mathrm{T},k}\right)\left(\boldsymbol{\tau}_{\mathrm{T},k} - \hat{\boldsymbol{\tau}}_{\mathrm{T},k}\right)^{\mathsf{T}}\right\} \\ &= \sigma_{s_{\mathrm{T}}}^2 \boldsymbol{h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)\boldsymbol{h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)^{\mathsf{T}} + \mathbf{H}_{\tau_{\mathrm{T}},k}\mathbf{P}_k\mathbf{H}_{\tau_{\mathrm{T}},k}{}^{\mathsf{T}}\;, \end{aligned} \tag{11}$$

[1] As it shall be seen, the state vector of the navigation filter described in Sect. 4 allows $\Delta\tau_{\mathrm{T},k}$ to be given only in terms of $\mathbf{x}_k$, needing not $\mathbf{x}_{k-1}$.

and

$$E\left\{\delta\boldsymbol{\Delta\tau}_{\mathrm{T},k}\delta\boldsymbol{\Delta\tau}_{\mathrm{T},k}^{\mathsf{T}}\right\} = E\left\{\left(\boldsymbol{\Delta\tau}_{\mathrm{T},k} - \boldsymbol{\Delta\hat{\tau}}_{\mathrm{T},k}\right)\left(\boldsymbol{\Delta\tau}_{\mathrm{T},k} - \boldsymbol{\Delta\hat{\tau}}_{\mathrm{T},k}\right)^{\mathsf{T}}\right\}$$
$$= \sigma_{s_\mathrm{T}}^2 \boldsymbol{\Delta h}_{\tau_\mathrm{T},k}(\hat{\mathbf{x}}_k)\boldsymbol{\Delta h}_{\tau_\mathrm{T},k}(\hat{\mathbf{x}}_k)^{\mathsf{T}} + \mathbf{H}_{\Delta\tau_\mathrm{T},k}\mathbf{P}_k\mathbf{H}_{\Delta\tau_\mathrm{T},k}^{\mathsf{T}} \;, \qquad (12)$$

with $\boldsymbol{\Delta h}_{\tau_\mathrm{T},k}(\hat{\mathbf{x}}_k) = \boldsymbol{h}_{\tau_\mathrm{T},k}(\hat{\mathbf{x}}_k) - \boldsymbol{h}_{\tau_\mathrm{T},k-1}(\hat{\mathbf{x}}_k)\;,\quad \mathbf{H}_{\Delta\tau_\mathrm{T},k} = \mathbf{H}_{\tau_\mathrm{T},k} - \mathbf{H}_{\tau_\mathrm{T},k-1}\;,$ (13)

and $\mathbf{H}_{\tau_\mathrm{T},k} = \left.\dfrac{\partial \boldsymbol{h}_{\tau_\mathrm{T},k}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\hat{\mathbf{x}}_k}\;,\quad \mathbf{H}_{\tau_\mathrm{T},k-1} = \left.\dfrac{\partial \boldsymbol{h}_{\tau_\mathrm{T},k-1}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\hat{\mathbf{x}}_k}\;.$ (14)

$\mathbf{P}_k$ is the filter state estimate covariance matrix. The value of σ_{s_T} is selected through simulation or flight data analysis, and according to the performance of the correction algorithm in use, to render the filter robust against the delay residuals. The tropospheric delay correction model used in this study can be found in Appendix.

4 Hybrid Navigation Algorithm Design

A diagram of the hybrid navigation system here developed is shown in Fig. 4. The system is tightly coupled with closed-loop indirect filtering (see Sect. 2). The IMU measurements are propagated at high-rate (400 Hz) by the strapdown routine while the raw GPS outputs (pseudoranges and integrated carrier phases), received at much lower rate (1 Hz), are used to update an error-state Kalman filter (running at 1 Hz). The estimated state corrections, including IMU error terms, are fed back to the strapdown routine. As explained in the preceding section, a tropospheric delay model corrects the GPS measurements and is used for the measurement update noise model, as means of rendering the filter robust against correction residuals.

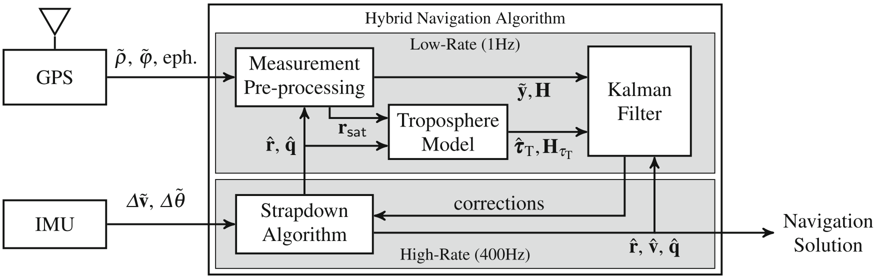

Hybrid Navigation Algorithm
Low-Rate (1Hz)
GPS
$\tilde{\rho}$, $\tilde{\varphi}$, eph.
Measurement Pre-processing
$\tilde{\mathbf{y}}, \mathbf{H}$
Kalman Filter
$\mathbf{r}_{\mathrm{sat}}$
Troposphere Model
$\hat{\mathbf{r}}$, $\hat{\mathbf{q}}$
$\hat{\boldsymbol{\tau}}_\mathrm{T}, \mathbf{H}_{\tau_\mathrm{T}}$
IMU
$\Delta\tilde{\mathbf{v}}$, $\Delta\tilde{\theta}$
Strapdown Algorithm
corrections
High-Rate (400Hz)
$\hat{\mathbf{r}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{q}}$
Navigation Solution

Fig. 4 Navigation algorithm block diagram

4.1 Reference Frames and Time Indexing

The WGS84 (World Geodetic System 1984) Earth-Centered Earth-Fixed (ECEF) is here denoted E; an Inertial reference frame, used to support the definition of absolute physical quantities, is denoted I; and a Body-fixed frame, with origin in the center of the IMU, is written B.

The time index of the high-rate tasks is denoted j, whereas that of the low-rate ones is k. The high-rate index is reset at each low-rate step, $t_k = t_{j=0}$, being $t_{k+1} = t_{j=N}$, with $N = 400$.

4.2 Strapdown Propagation

The vehicle body attitude with respect to ECEF frame is represented using the quaternion $\mathbf{q}_B^E$. This quantity is propagated according to

$$\mathbf{q}_{B_{j+1}}^{E_{j+1}} = \begin{bmatrix} \left(-\frac{1}{2} + \frac{1}{48}\|\boldsymbol{\Delta\theta}_j^B\|^2 + \frac{1}{24}\left[\boldsymbol{\Delta\theta}_{j-1}^B \times\right]\right) \boldsymbol{\Delta\theta}_j^B \\ 1 - \frac{1}{8}\|\boldsymbol{\Delta\theta}_j^B\|^2 \end{bmatrix} \mathbf{q}_{B_j}^{E_j} \mathbf{q}_{E_j}^{E_{j+1}} \ , \tag{15}$$

where the transition quaternion from B_j to B_{j+1} follows the third order algorithm by McKern [12]. $\left[\boldsymbol{\Delta\theta}_{j-1}^B \times\right]$ is the skew-symmetric matrix of $\boldsymbol{\Delta\theta}_{j-1}^B$.

The rigid-body translational kinematics can be given in the inertial frame as

$$\mathbf{v}_{j+1}^I = \mathbf{v}_j^I + \int_{t_j}^{t_{j+1}} \left(\mathbf{C}_B^I \mathbf{a}_{\mathsf{sf}}^B + \mathbf{g}^I\right) \mathsf{d}t = \mathbf{v}_j^I + \boldsymbol{\Delta}\mathbf{v}_{\mathsf{sf},j}^I + \boldsymbol{\Delta}\mathbf{v}_{\mathsf{g},j}^I \tag{16}$$

$$\mathbf{r}_{j+1}^I = \mathbf{r}_j^I + \int_{t_j}^{t_{j+1}} \mathbf{v}^I \mathsf{d}t = \mathbf{r}_j^I + \frac{\Delta t_j}{2}\left(\mathbf{v}_j^I + \mathbf{v}_{j+1}^I\right) \ , \tag{17}$$

where $\mathbf{g}^I$ and $\mathbf{a}_{\mathsf{sf}}^B$ are gravity and specific-force accelerations, respectively. The integral in (17) was solved using a trapezoidal method, with $\Delta t_j = t_{j+1} - t_j$. System (16) and (17) is easily transformed to ECEF yielding the propagation laws for $\mathbf{v}_j^E$ and $\mathbf{r}_j^E$.

The gravity induced Delta-V, $\boldsymbol{\Delta}\mathbf{v}_{\mathsf{g},j}^I$, can be approximated evaluating the gravity field model at the midpoint of the integration period as

$$\boldsymbol{\Delta}\mathbf{v}_{\mathsf{g},j}^I \ \approx \ \Delta t_j \, \mathbf{C}_{E_{j+\frac{1}{2}}}^I \mathbf{g}_{j+\frac{1}{2}}^E = \Delta t_j \, \mathbf{C}_{E_{j+\frac{1}{2}}}^I \boldsymbol{g}^E\left(\mathbf{r}_{j+\frac{1}{2}}^E\right) \ , \tag{18}$$

where $\boldsymbol{g}^E(\mathbf{r}^E)$ is the ECEF gravity model function. The specific force Delta-V, $\boldsymbol{\Delta}\mathbf{v}_{\mathsf{sf},j}^I$, is also assumed to be taken at the midpoint of the integration period to account for the rotation of Body frame during this interval. It is given by

$$\boldsymbol{\Delta}\mathbf{v}^{I}_{\mathsf{sf},j} \approx \mathbf{C}^{I}_{E_j}\mathbf{C}^{E_j}_{B_j}\mathbf{C}^{B_j}_{B_{j+\frac{1}{2}}}\boldsymbol{\Delta}\mathbf{v}^{B}_{j} , \tag{19}$$

where $\boldsymbol{\Delta}\mathbf{v}^{B}_{j}$ is the accelerometer measurement from t_j to t_{j+1}.

4.3 Kalman Filter Equations

The error-state filter equations from [20] are here modified to accommodate the correlation between measurement and process noise. Recall from Sect. 3 that, in the present system, the clock error in the TDCP is perfectly correlated with the driving noise of the clock-bias state.

The whole-state system and measurement models are written as

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_{k+1|k}(\mathbf{x}_k) + \mathbf{w}_k \tag{20}$$

$$\mathbf{y}_k = \boldsymbol{h}_k(\mathbf{x}_k) + \boldsymbol{\varepsilon}_k + \boldsymbol{\nu}_k , \tag{21}$$

where $\mathbf{w}_k \sim N\left(\mathbf{0},\ \mathbf{Q}_{k+1|k}\right)$ is the process noise, and $\boldsymbol{\varepsilon}_k \sim N\left(\mathbf{0},\ \mathbf{R}_{\varepsilon,k}\right)$ and $\boldsymbol{\nu}_k \sim N\left(\mathbf{0},\ \mathbf{R}_{\nu,k}\right)$ are measurement noises. $\boldsymbol{\varepsilon}$ is correlated to $\mathbf{w}$ according to

$$E\left\{\boldsymbol{\varepsilon}_k \mathbf{w}^{\mathsf{T}}_{k-1}\right\} = \mathbf{B}^{\mathsf{T}}_k . \tag{22}$$

The error-state system and measurement models are given, respectively, by

$$\boldsymbol{\delta}\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1} = \boldsymbol{\phi}_{k+1|k}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - \boldsymbol{\phi}_{k+1|k}(\hat{\mathbf{x}}_k) + \mathbf{w}_k \tag{23}$$

$$\boldsymbol{\delta}\mathbf{y}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k = \boldsymbol{h}_k(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - \boldsymbol{h}_k(\hat{\mathbf{x}}_k) + \boldsymbol{\varepsilon}_k + \boldsymbol{\nu}_k . \tag{24}$$

The covariance of the a priori error-state follows[2]

$$\mathbf{P}_{(k+1)^-} \approx \Phi_{k+1|k}\mathbf{P}_{k^+}\Phi^{\mathsf{T}}_{k+1|k} + \mathbf{Q}^{\mathsf{T}}_{k+1|k} , \tag{25}$$

where $\Phi_{k+1|k}$ is the Jacobian matrix of the vector function $\boldsymbol{\phi}_{k+1|k}(\mathbf{x}_k)$ evaluated at the a posteriori whole-state estimate at time step k, $\hat{\mathbf{x}}_{k^+}$.

The filter correction step is given by

$$\boldsymbol{\delta}\hat{\mathbf{x}}_{k^+} = \boldsymbol{\delta}\hat{\mathbf{x}}_{k^-} + \mathbf{K}_k\left(\boldsymbol{\delta}\mathbf{y} - \boldsymbol{h}_k\left(\hat{\mathbf{x}}_{k^-} + \boldsymbol{\delta}\hat{\mathbf{x}}_{k^-}\right) + \boldsymbol{h}_k\left(\hat{\mathbf{x}}_{k^-}\right)\right) . \tag{26}$$

with the Kalman gain $\mathbf{K}_k$ given as in [6] as

$$\mathbf{K}_k = \left(\mathbf{P}_{k^-}\mathbf{H}^{\mathsf{T}}_k + \mathbf{B}_k\right)\left(\mathbf{H}_k\mathbf{P}_{k^-}\mathbf{H}^{\mathsf{T}}_k + \mathbf{H}_k\mathbf{B}_k + \mathbf{B}^{\mathsf{T}}_k\mathbf{H}^{\mathsf{T}}_k + \mathbf{R}_{\varepsilon,k} + \mathbf{R}_{\nu,k}\right)^{-1} \tag{27}$$

[2] A priori and a posteriori estimates are denoted, respectively, by the index superscripts $-$ and $+$.

yielding

$$\mathbf{P}_{k^+} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k^-} - \mathbf{K}_k\mathbf{B}_k^{\mathsf{T}}\ , \tag{28}$$

where $\mathbf{H}_k$ is the Jacobian matrix of the measurement vector function $\boldsymbol{h}_k(\mathbf{x}_k)$ evaluated at the a priori whole-state estimate, $\hat{\mathbf{x}}_{k^-}$.

After a measurement update cycle the whole state is updated with the error-state vector, being the latter reset,

$$\hat{\mathbf{x}}_{k^+} = \hat{\mathbf{x}}_{k^-} + \boldsymbol{\delta}\hat{\mathbf{x}}_{k^+}\ , \qquad \boldsymbol{\delta}\hat{\mathbf{x}}_{k^+} \leftarrow \mathbf{0}\ . \tag{29}$$

4.4 State Vector

As stated in the previous point the navigation system at hand uses two state vectors: the whole state and error state vectors. For each estimated whole state there is a corresponding error state. States can also be split into: strapdown (kinematics) states, IMU states (bias and scale-factor states) and measurement model states, as

$$\mathbf{x}_k = \left[\mathbf{x}_{\mathsf{kin},k}^{\mathsf{T}}\ \mathbf{x}_{\mathsf{imu},k}^{\mathsf{T}}\ \mathbf{x}_{\mathsf{meas},k}^{\mathsf{T}}\right]^{\mathsf{T}}\ . \tag{30}$$

In general the error states are simple additive disturbances of the whole-states as in (23). The attitude is an exception. The error-state is instead a 3-component angle as

$$\begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}_k^B \\ 1 \end{bmatrix} \approx \mathbf{q}_{B_k}^{E_k}\left(\hat{\mathbf{q}}_{B_k}^{E_k}\right)^{-1}\ . \tag{31}$$

The set of kinematics whole-states $\mathbf{x}_{\mathsf{kin},k}$ includes $\mathbf{v}_k^E$, $\mathbf{r}_k^E$ and $\mathbf{q}_{B_k}^{E_k}$, being the corresponding error-states $\boldsymbol{\delta}\mathbf{v}_k^E$, $\boldsymbol{\delta}\mathbf{r}_k^E$ and $\boldsymbol{\delta\theta}_k^B$. The remaining state sets, $\mathbf{x}_{\mathsf{imu}}$ and $\mathbf{x}_{\mathsf{meas}}$, are introduced in the following sections.

4.5 IMU Error Model

The accelerometer and gyroscope measurements were modelled including the disturbing effects of scale factor and bias errors, as

$$\boldsymbol{\Delta}\tilde{\mathbf{v}}_{\mathsf{sf},j}^B = \left(\mathbf{I} + \mathrm{diag}\left(\mathbf{s}_{\mathsf{a},j}^B\right)\right)\boldsymbol{\Delta}\mathbf{v}_{\mathsf{sf},j}^B + \Delta t_j\mathbf{b}_{\mathsf{a},j}^B + \boldsymbol{\nu}_{\mathsf{a},j} \tag{32}$$

$$\boldsymbol{\Delta}\tilde{\boldsymbol{\theta}}_j^B = \left(\mathbf{I} + \mathrm{diag}\left(\mathbf{s}_{\mathsf{g},j}^B\right)\right)\boldsymbol{\Delta\theta}_j^B + \Delta t_j\ \mathbf{b}_{\mathsf{g},j}^B + \boldsymbol{\nu}_{\mathsf{g},j}\ . \tag{33}$$

where $\boldsymbol{\Delta}\tilde{\mathbf{v}}_{\mathsf{sf},j}^B$ and $\boldsymbol{\Delta}\tilde{\boldsymbol{\theta}}_j^B$ are the measured increments, while $\boldsymbol{\Delta}\mathbf{v}_{\mathsf{sf},j}^B$ and $\boldsymbol{\Delta\theta}_j^B$ are the actual quantities. $\boldsymbol{\nu}_{\mathsf{a},j}$ and $\boldsymbol{\nu}_{\mathsf{g},j}$ are measurement noise terms. The bias $\mathbf{b}_{\mathsf{a},j}^B$ and

$\mathbf{b}^B_{\mathsf{g},j}$, and scale-factors $\mathbf{b}^B_{\mathsf{a},j}$ and $\mathbf{b}^B_{\mathsf{g},j}$ are the filter states in $\mathbf{x}_{\mathsf{imu},k}$. To each of these quantities corresponds an error-state: $\boldsymbol{\delta}\mathbf{b}^B_{\mathsf{a},j}$, $\boldsymbol{\delta}\mathbf{b}^B_{\mathsf{g},j}$, $\boldsymbol{\delta}\mathbf{s}^B_{\mathsf{a},j}$, and $\boldsymbol{\delta}\mathbf{s}^B_{\mathsf{g},j}$. These IMU states are modelled as random walk processes with random initial condition. The values used correspond to the fibre-optic gyros and servo accelerometers of the tactical grade IMU of the HNS version flown on-board of SHEFEX2 [20].

4.6 Filter Propagation

The strapdown algorithm presented earlier propagates the kinematics whole-states at high-rate (time-index j). Linearizing the error-state system

$$\boldsymbol{\delta}\mathbf{x}_{j+1} = \boldsymbol{\phi}_{j+1|j}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_j) - \boldsymbol{\phi}_{j+1|j}(\hat{\mathbf{x}}_j) + \mathbf{w}_j \approx \Phi_{j+1|j}\boldsymbol{\delta}\mathbf{x}_j + \mathbf{w}_j \ , \tag{34}$$

which yields

$$\Phi_{j+1|j} = \begin{bmatrix} \Phi^{\mathsf{kin}}_{\mathsf{kin}} & \Phi^{\mathsf{imu}}_{\mathsf{kin}} & \mathbf{0} \\ \mathbf{0} & \Phi^{\mathsf{imu}}_{\mathsf{imu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi^{\mathsf{meas}}_{\mathsf{meas}} \end{bmatrix}_{j+1|j} , \tag{35}$$

and assuming that the accumulated system noise over one low-rate step (index k) is lower than the total state uncertainty, the low-rate transition and process noise covariance matrices can be approximated as

$$\Phi_{k+1|k} \approx \prod_{j=0}^{N-1} \Phi_{j+1|j} \ , \qquad \mathbf{Q}_{k+1|k} \approx \sum_{j=0}^{N-1} \mathbf{Q}_{j+1|j} \ , \tag{36}$$

where $t_k = t_{j=0}$ and $t_{k+1} = t_{j=N}$.

The sub-matrices $\Phi^{\mathsf{kin}}_{\mathsf{kin}}$ and $\Phi^{\mathsf{imu}}_{\mathsf{kin}}$ can be easily derived from the linearization of the translational kinematics (16) and (17), with (18) and (19), after transforming all to the ECEF frame, and the error-angle kinematics given in [20] by

$$\boldsymbol{\delta\theta}^B_{j+1} = \hat{\mathbf{C}}^{B_{j+1}}_{B_j}\boldsymbol{\delta\theta}^B_j - \boldsymbol{\delta\Delta\theta}^B_j + noise \ . \tag{37}$$

The mapping from IMU error-states $\boldsymbol{\delta}\mathbf{x}_{\mathsf{imu}}$ to the error in the increments $\boldsymbol{\delta\Delta}\mathbf{v}^B$ and $\boldsymbol{\delta\Delta\theta}^B$ is obtained through the (approximate) inverse of the IMU error model (32) and (33)

$$\boldsymbol{\delta\Delta}\mathbf{v}^B_j = -\operatorname{diag}\big(\boldsymbol{\Delta}\tilde{\mathbf{v}}^B_j\big)\boldsymbol{\delta}\mathbf{s}^B_{\mathsf{a},j} - \Delta t_j\boldsymbol{\delta}\mathbf{b}^B_{\mathsf{a},j} - \boldsymbol{\nu}_{\mathsf{a},j} \tag{38}$$

$$\boldsymbol{\delta\Delta\theta}^B_j = -\operatorname{diag}\big(\boldsymbol{\Delta}\tilde{\boldsymbol{\theta}}^B_j\big)\boldsymbol{\delta}\mathbf{s}^B_{\mathsf{g},j} - \Delta t_j\boldsymbol{\delta}\mathbf{b}^B_{\mathsf{g},j} - \boldsymbol{\nu}_{\mathsf{g},j} \ . \tag{39}$$

The IMU error random walk states mean that $\Phi^{\mathsf{imu}}_{\mathsf{imu}} = \mathbf{I}$. As for $\Phi^{\mathsf{meas}}_{\mathsf{meas}}$, it shall be easily formed upon definition, in the next points, of the filter update models.

4.7 Filter Updates

The filter update models used for Pseudorange and Time-differenced Integrated Carrier Phase measurements are as follows.

Pseudorange Update

The model for a set of pseudorange measurements can be written in the form

$$\mathbf{y}_{\rho,k} = \boldsymbol{h}_{\rho,k}(\mathbf{x}_k) + \boldsymbol{\nu}_{\rho,k} \ , \tag{40}$$

where the measurement (vector) function $\boldsymbol{h}_{\rho}(\mathbf{x}_k)$ is for channel i given by

$$h_{\rho,i,k}(\mathbf{x}_k) = \rho_{i,k}\left(\mathbf{x}_k\right) + c\left(\tau_{\mathsf{r},k} + \tau_{\mathrm{T},i,k}\left(\mathbf{x}_k\right)\right) + \mathrm{b}_{\rho,i,k} \ , \tag{41}$$

with range

$$\rho_{i,k}\left(\mathbf{x}_k\right) = \left\| \mathbf{C}^{E_k}_{E(t_{\mathsf{s},i,k})}\mathbf{r}^E_i\left(t_{\mathsf{s},i,k}\right) - \mathbf{r}^E_k - \mathbf{C}^{E_k}_{B_k}\mathbf{l}^B_{\mathsf{a}} \right\| \ . \tag{42}$$

The satellite position $\mathbf{r}^E_i$ is evaluated at the emitting instant $t_{\mathsf{s},i,k}$ and translated to the ECEF coordinates at time of reception. $\mathbf{l}_{\mathsf{a}}$ is the lever-arm from IMU to GPS antenna. The pseudorange bias state $\mathrm{b}_{\rho,i}$ is modelled as a random constant plus random walk process (i.e. $\Phi^{\mathrm{b}_{\rho,i}}_{\mathrm{b}_{\rho,i}} = 1$). The receiver clock-bias state τ_{r} has the dynamics given by (6). The tropospheric delay $\tau_{\mathrm{T},i,k}(\mathbf{x}_k)$ is given by (9).

The error-measurement model is similar to (24) as

$$\boldsymbol{\delta}\mathbf{y}_{\rho,k} = \boldsymbol{h}_{\rho,k}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - \boldsymbol{h}_{\rho,k}(\hat{\mathbf{x}}_k) + \boldsymbol{\nu}_{\rho,k} \ . \tag{43}$$

where for the ith channel

$$h_{\rho,i,k}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - h_{\rho,i,k}(\hat{\mathbf{x}}_k) = \delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k) + c\left(\delta\tau_{\mathsf{r},k} + \delta\tau_{\mathrm{T},i,k}\left(\boldsymbol{\delta}\mathbf{x}_k\right)\right) + \delta\mathrm{b}_{\rho,i,k} \ , \tag{44}$$

with

$$\delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k) \approx -\hat{\mathbf{e}}^{E}_{\rho,i,k}{}^{\mathsf{T}}\left(\boldsymbol{\delta}\mathbf{r}^E_k + \hat{\mathbf{C}}^{E_k}_{B_k}\left[\mathbf{l}^B_{\mathsf{a}}\times\right]\boldsymbol{\delta\theta}^B_k\right) \tag{45}$$

$$\delta\tau_{\mathrm{T},i,k}\left(\boldsymbol{\delta}\mathbf{x}_k\right) \approx \mathrm{s}_{\mathrm{T},i}h_{\tau_{\mathrm{T}},i,k}\left(\hat{\mathbf{x}}_k\right) + \mathbf{h}_{\tau_{\mathrm{T}},i,k}\boldsymbol{\delta}\mathbf{x}_k \ . \tag{46}$$

$\hat{\mathbf{e}}^{E}_{\rho,i,k}$ is the estimated unit range vector from receiver to satellite i, and $\mathbf{h}_{\tau_{\mathrm{T}},i,k}$, as derived in Appendix, is the Jacobian row vector of the tropospheric delay model function $h_{\tau_{\mathrm{T}},i,k}(\mathbf{x}_k)$ with respect to the error-state vector.

The covariance of the error-measurement (43) is

$$E\left\{\boldsymbol{\delta}\mathbf{y}_{\rho,k}\boldsymbol{\delta}\mathbf{y}_{\rho,k}^{\mathsf{T}}\right\} \approx \mathbf{H}_{\rho,k}\mathbf{P}_k\mathbf{H}_{\rho,k}^{\mathsf{T}} + c^2\,\sigma_{s_{\mathrm{T}}}^2\,\boldsymbol{h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)\boldsymbol{h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)^{\mathsf{T}} + \mathbf{R}_{\rho,k}\ , \tag{47}$$

where

$$\mathbf{H}_{\rho,k} = \left[\mathbf{h}_{\rho,1,k}^{\mathsf{T}} \cdots \mathbf{h}_{\rho,i,k}^{\mathsf{T}} \cdots \mathbf{h}_{\rho,n,k}^{\mathsf{T}}\right]^{\mathsf{T}}\ , \tag{48}$$

with non-null partial derivatives

$$\begin{aligned} \mathbf{h}_{\rho,i,k}^{\mathrm{r}} &= -\hat{\mathbf{e}}_{\rho,i,k}^{E}{}^{\mathsf{T}} + c\,\mathbf{h}_{\tau_{\mathrm{T}},i}^{\mathrm{r}}\ , \quad & \mathbf{h}_{\rho,i,k}^{\theta} &= -\hat{\mathbf{e}}_{\rho,i,k}^{E}{}^{\mathsf{T}}\hat{\mathbf{C}}_{B_k}^{E_k}\left[\mathbf{l}_{\mathsf{a}}^{B}\times\right] + c\,\mathbf{h}_{\tau_{\mathrm{T}},i}^{\theta}\ , \\ \mathrm{h}_{\rho,i,k}^{\tau_{\mathrm{r}}} &= c\ , & \mathrm{h}_{\rho,i,k}^{\mathrm{b}_{\rho,i}} &= 1\ . \end{aligned} \tag{49}$$

$\mathbf{h}_{\tau_{\mathrm{T}},i,k}^{\mathrm{r}}$ and $\mathbf{h}_{\tau_{\mathrm{T}},i,k}^{\theta}$ are the components of $\mathbf{h}_{\tau_{\mathrm{T}},i,k}$ corresponding to $\boldsymbol{\delta}\mathbf{r}_k^E$ and $\boldsymbol{\delta\theta}_k^B$.

Time-Differenced Integrated Carrier Phase Update

The TDCP measurement set can be modelled as

$$\mathbf{y}_{\Delta\rho,k} = \boldsymbol{h}_{\Delta\rho,k}(\mathbf{x}_k) + \mathbf{1}c\Delta\tau_{\mathrm{r},k} + \boldsymbol{\nu}_{\Delta\rho,k}\ , \tag{50}$$

where $\boldsymbol{\nu}_{\Delta\rho,k} \sim N\left(\mathbf{0}, \mathbf{R}_{\Delta\rho,k}\right)$ holds the single-channel noises and $\Delta\tau_{\mathrm{r},k} \sim N\left(0, \sigma_{\Delta\tau_{\mathrm{r}},k}^2\right)$, the TDCP clock error in (6), is perfectly correlated to the process noise of τ_{r}, as (8).

The state-to-measurement mapping function is for channel i

$$h_{\Delta\rho,i,k}(\mathbf{x}_k) = \rho_{i,k}\left(\mathbf{x}_k\right) - \rho_{i,k-1}\left(\mathbf{x}_k\right) + c\Delta\tau_{\mathrm{T},i,k}\left(\mathbf{x}_k\right)\ , \tag{51}$$

where $\rho_{i,k}\left(\mathbf{x}_k\right)$ is as (42), $\Delta\tau_{\mathrm{T},i,k}\left(\mathbf{x}_k\right)$ follows (10), and

$$\rho_{i,k-1}\left(\mathbf{x}_k\right) = \left\|\mathbf{C}_{E(t_{\mathsf{s},i,k-1})}^{E_{k-1}}\mathbf{r}_i^E\left(t_{\mathsf{s},i,k-1}\right) - \mathbf{r}_{\mathsf{a},k-1,k}^{E_{k-1}}\right\|\ . \tag{52}$$

$\mathbf{r}_{\mathsf{a},k-1,k}^{E_{k-1}}$ is a delayed state of GPS antenna position. This is a constant state, with null process noise, that is reset after each measurement update cycle as

$$\left\{\hat{\mathbf{r}}_{\mathsf{a},k-1}^{E_{k-1}}\right\}_{k^+} \leftarrow \left\{\hat{\mathbf{r}}^E + \hat{\mathbf{C}}_B^E\mathbf{l}_{\mathsf{a}}^B\right\}_{k^+}\ . \tag{53}$$

Note that $\mathbf{r}_{\mathsf{a},k-1}^{E_{k-1}}$ is treated as a *considered* state, i.e. it is not updated ($\mathbf{K}_k^{\mathrm{r}_{\mathsf{a}}} = \mathbf{0}$). This renders the computational effort required by the inclusion of this state relatively low. Upon reset its covariance and cross-covariance (to the remaining states) are set to

$$\left\{\mathbf{P}_{\mathrm{xr}_{\mathsf{a}}}\right\}_{k^+} \leftarrow \left\{\mathbf{P}_{\mathrm{xx}}\Phi_{\mathrm{r}_{\mathsf{a}},0}^{\mathsf{x}}{}^{\mathsf{T}}\right\}_{k^+}\ , \qquad\qquad \left\{\mathbf{P}_{\mathrm{r}_{\mathsf{a}}\mathrm{r}_{\mathsf{a}}}\right\}_{k^+} \leftarrow \mathbf{P}_{\mathrm{r}_{\mathsf{a}},0}\ , \tag{54}$$

where $\mathbf{P}_{\mathrm{xx}}$ is the covariance matrix of the remaining states, $\Phi_{\mathrm{r_a x},0}$ is sparse with only

$$\Phi^{\mathrm{r}}_{\mathrm{r_a},0} = \mathbf{I} \;, \qquad \Phi^{\theta}_{\mathrm{r_a},0} = \hat{\mathbf{C}}^{E_k}_{B_k}\left[\mathbf{l}^{B}_{\mathsf{a}} \times\right] \;, \tag{55}$$

and $\mathbf{P}_{\mathrm{r_a},0}$ is a small initial covariance used to avoid numerical problems.

The error-measurement $\boldsymbol{\delta}\mathbf{y}_{\Delta\rho,k}$ follows a similar logic to (43), as

$$\boldsymbol{\delta}\mathbf{y}_{\Delta\rho,k} = \boldsymbol{h}_{\Delta\rho,k}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - \boldsymbol{h}_{\Delta\rho,k}(\hat{\mathbf{x}}_k) + \mathbf{1}c\Delta\tau_{\mathrm{r},k} + \boldsymbol{\nu}_{\Delta\rho,k} \;. \tag{56}$$

being for channel i

$$h_{\Delta\rho,i,k}(\hat{\mathbf{x}}_k + \boldsymbol{\delta}\mathbf{x}_k) - h_{\Delta\rho,i,k}(\hat{\mathbf{x}}_k) = \delta\Delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k) + c\delta\Delta\tau_{\mathrm{T},i,k}\,(\boldsymbol{\delta}\mathbf{x}_k) \tag{57}$$

and

$$\delta\Delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k) \approx \delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k) + {\hat{\mathbf{e}}^{E}_{\rho,i,k-1}}^{\mathsf{T}}\boldsymbol{\delta}\mathbf{r}^{E_{k-1}}_{\mathsf{a},k-1} \;, \tag{58}$$

$$\delta\Delta\tau_{\mathrm{T},i,k}\,(\boldsymbol{\delta}\mathbf{x}_k) \approx \mathrm{s}_{\mathrm{T},i}\,\Delta h_{\tau_{\mathrm{T}},i,k}\left(\hat{\mathbf{x}}_k\right) + \mathbf{h}_{\tau_{\mathrm{T}},i,k}\boldsymbol{\delta}\mathbf{x}_k - \mathbf{h}^{\mathrm{r}}_{\tau_{\mathrm{T}},i,k-1}\boldsymbol{\delta}\mathbf{r}^{E_{k-1}}_{\mathsf{a},k-1} \;. \tag{59}$$

where $\delta\rho_{i,k}(\boldsymbol{\delta}\mathbf{x}_k)$ is as (45).

The covariance of $\boldsymbol{\delta}\mathbf{y}_{\Delta\rho,k}$ is formed accounting for the correlation (8) as

$$\begin{aligned} E\left\{\boldsymbol{\delta}\mathbf{y}_{\Delta\rho,k}\boldsymbol{\delta}\mathbf{y}^{\mathsf{T}}_{\Delta\rho,k}\right\} \approx{}& \mathbf{H}_{\Delta\rho,k}\mathbf{P}_k\mathbf{H}^{\mathsf{T}}_{\Delta\rho,k} + c^2\sigma^2_{\mathrm{s_T}}\boldsymbol{\Delta h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)\boldsymbol{\Delta h}_{\tau_{\mathrm{T}},k}(\hat{\mathbf{x}}_k)^{\mathsf{T}} \\ &+3c^2\sigma^2_{\Delta\tau_{\mathrm{r}},k}\mathbf{1}\mathbf{1}^{\mathsf{T}} + \mathbf{R}_{\Delta\rho,k} \;, \end{aligned} \tag{60}$$

where

$$\mathbf{H}_{\Delta\rho,k} = \left[\mathbf{h}^{\mathsf{T}}_{\Delta\rho,1,k}\cdots\mathbf{h}^{\mathsf{T}}_{\Delta\rho,i,k}\cdots\mathbf{h}^{\mathsf{T}}_{\Delta\rho,n,k}\right]^{\mathsf{T}} \;, \tag{61}$$

with non-null components

$$\mathbf{h}^{\mathrm{r}}_{\Delta\rho,i,k} = \mathbf{h}^{\mathrm{r}}_{\rho,i,k} \;, \qquad \mathbf{h}^{\theta}_{\Delta\rho,i,k} = \mathbf{h}^{\theta}_{\rho,i,k} \;, \qquad \mathbf{h}^{\mathrm{r_a}}_{\Delta\rho,i,k} = {\hat{\mathbf{e}}^{E}_{\rho,i,k-1}}^{\mathsf{T}} - c\,\mathbf{h}^{\mathrm{r}}_{\tau_{\mathrm{T}},i,k-1} \;. \tag{62}$$

5 Testing Results

The navigation algorithm here presented was *Hardware-in-the-Loop* (HIL) tested using real GPS data. A SPIRENT GNSS emulator was used to stimulate a Phoenix-HD receiver, having its output been filtered offline. Atmospheric disturbances such as Ionospheric and Tropospheric delays (STANAG tropospheric model [14]) were included in the emulation. The trajectory used follows a similar profile to that of the SHEFEX2 mission [21]. Figure 5 shows the altitude time history of the two-burn sounding rocket flight. The IMU measurements were simulated using a tactical-grade unit model with specifications as shown in Table 1.

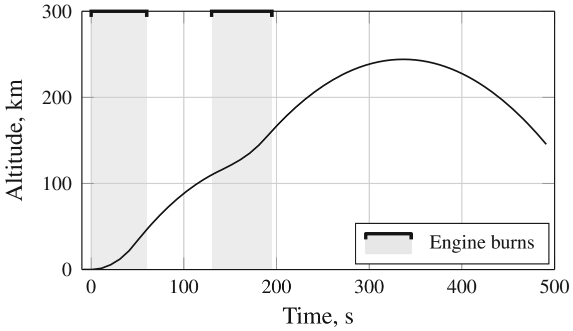


Fig. 5 Altitude profile of SHEFEX2 mission

Table 1 IMU (1σ) specifications [20]

	Gyroscope	Accelerometer
Sensor range	±1000 deg/s	±5 g
Bandwidth	500 Hz	200 Hz
Axis misalignment	0.5 mrad	0.5 mrad
Angle/vel. random walk	$0.03\,\text{deg}/\sqrt{\text{h}}$	$50\,\mu\text{g}/\sqrt{\text{Hz}}$
Bias repeatability	1 deg/h	2 mg
Bias instability	0.03 deg/h	$50\,\mu$g
Scale-factor repeatability	300 ppm	1500 ppm

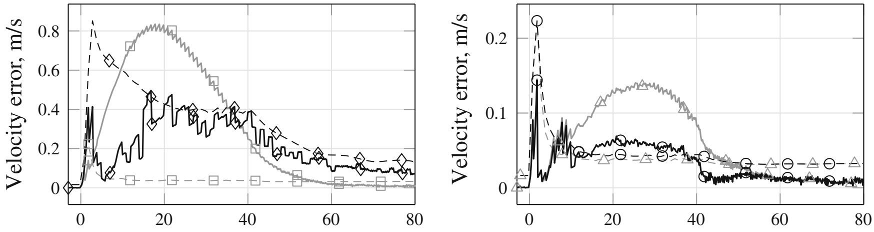
Tropospheric Delay Correction

Velocity estimation performance during the atmospheric ascent is compared in Fig. 6 for four different set-ups using both TDCP and PR updates:

(1) uncorrected measurements and update models neglecting Tropospheric delay;
(2) uncorrected measurements but inflated measurement model noise values;
(3) corrected measurements neglecting correction residuals; and
(4) corrected measurements considering correction residuals (as in Sect. 3).

To compensate for the uncorrected atmospheric errors in the GPS signal, Steffes [19] used 1σ covariance of the PR measurements scheduled in altitude as: 50 m for altitudes below 10 km (also accounting for multipath on the pad), 10 m for altitudes within 10–200 km, and 2 m for altitudes above 200 km. Configuration (2) employs these same values and a similar conservative schedule for the TDCP measurements.

The advantage of the configurations using corrected measurements is clear upon comparison of Fig. 6a, b. Note, however, that set-ups without covariance compensation, either against the entire tropospheric delay or the leftover correction residual, show signs of severe mismodelling, behaving over-optimistically. The proposed compensation method presented in Sect. 3 successfully maintains the filter coherence, further reducing the impact of the residual delay on the estimation.



(a) Uncorrected TDCP + PR

(b) Corrected TDCP + PR

	Line Style[a]	Tropo Corrected	Covariance Compensation
1)	—□—	no	none
2)	—◇—	no	**R** scheduled *vs* altitude as in [**19**]
3)	—△—	yes	none
4)	—⊖—	yes	as described in Section 3 (Time - Differenced Integrated Carrier Phases)

[a] Dashed lines show 1σ covariance bound.

Fig. 6 Velocity estimation error with and without tropospheric delay correction and covariance compensation

Nominal Performance

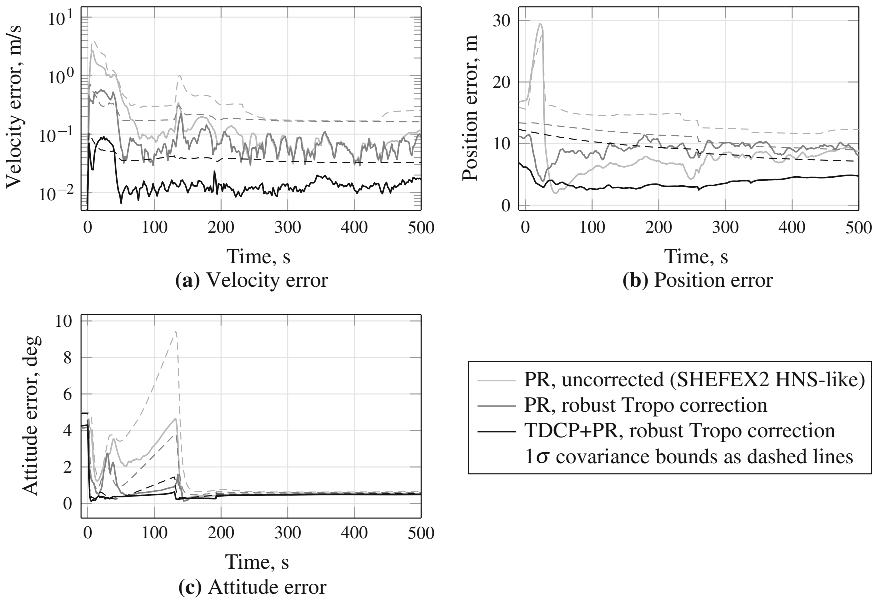
In Fig. 7 the estimation error in terms of the kinematic states of the algorithm developed herein (filter 4 in the previous analysis) is compared to:

- an uncorrected PR-only SHEFEX2 HNS-like configuration [19, 20]; and
- a PR-only set-up with tropospheric corrections as described in Sect. 3.

Once again, using tropospheric delay correction, even if only employing PR updates, considerably improves estimation during the atmospheric flight leg, especially in terms of velocity and attitude. However, it is the addition of TDCP measurements that yields the greatest improvement. As expected, given the delta-range nature of the TDCP observables and its low noise, velocity error sees the most significant reduction (about 10-fold throughout). Attitude estimation also benefits from the more accurate velocity information, especially during the more dynamic flight phases.

GNSS Outage

Vehicle dynamics can cause loss of lock in the GNSS tracking loops. Events such as separation, ignition, peak dynamic pressure, sonic transition are especially critical [9]. Figure 8 compares the developed filter to the configurations described in the previous point under a 30 s outage starting at t =20 s. While all filters visibly drift, the proposed configuration clearly sees its velocity, position and attitude estimation diverge the least during the outage. The lower estimation error, granted especially by the use of TDCP updates, results in a better initial state for the pure inertial propagation during drop-out, thus yielding superior outage performance.



(a) Velocity error

(b) Position error

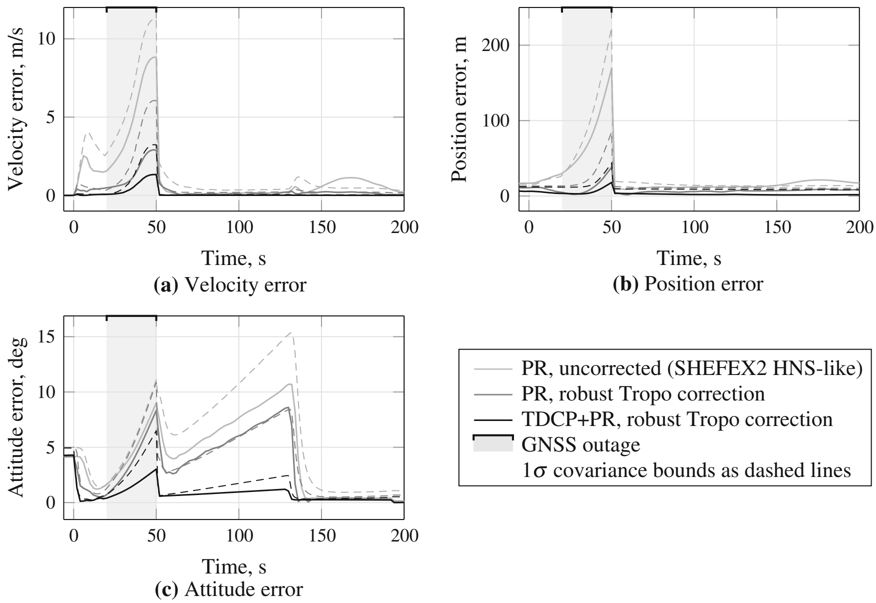
(c) Attitude error

Fig. 7 Nominal estimation performance of the proposed system compared to two PR-only filters

6 Conclusion

A tightly coupled, Time-differenced Carrier Phase and Pseudorange updated, hybrid navigation system has been presented. In the light of rocket launch conditions, the chosen architecture yields added robustness with respect to a loose coupling, while being realizable with COTS components, contrarily to the ultra-tight set-up. The receiver clock error dynamics affecting both TDCP and PR observables has been analysed for the receiver at hand and approximated by a single-state model, avoiding the wasteful cross-channel differences while carrying little extra computational effort. Tropospheric delta-delay has been found to be a major disturbance of the TDCP measurements. Although conservative measurement model tuning at lower atmosphere could render the filter coherent under this effect, estimation performance is poor when compared to a feed-forward corrected set-up. The residuals of this correction, however, still caused severe mismodelling. The proposed correction scheme with covariance compensation has shown better robustness against such residuals, maintaining filter coherence. The performance of the overall system has been compared to a SHEFEX2 HNS-like baseline configuration through HIL simulation, displaying significant improvements in nominal and outage conditions.

Acknowledgements The authors would like to thank ESA, Guidance, Navigation & Control Section at the European Space Research and Technology Centre for granting the funding, through the Network/Partnering Initiative contract 4000111837/14/NL/MH, which enabled this research.



(**a**) Velocity error

(**b**) Position error

(**c**) Attitude error

Fig. 8 Estimation performance comparison in case of GNSS outage

In addition, the authors would like to thank Oliver Montenbruck, Markus Markgraf and Benjamin Braun from the German Space Operations Center (DLR) for the support to this work.

Appendix

Parkinson et al. [15] suggests the following coarse tropospheric delay model, which does not require in situ atmospheric measurements nor the use of look-up tables:

$$h_{\tau_\mathrm{T},i,k}\left(\mathbf{x}_k\right) = \tfrac{1}{c}\, m(\mathrm{E}_{i,k})\mathit{\Delta}(\mathrm{h}_{\mathsf{a},k}) \ , \tag{63}$$

$$\text{with} \quad m(\mathrm{E}_{i,k}) = \frac{1.0121}{\sin \mathrm{E}_{i,k} + 0.0121} \ , \quad \mathit{\Delta}(\mathrm{h}_{\mathsf{a},k}) = 2.4405\, e^{-0.133\times 10^{-3}\ \mathrm{h}_{\mathsf{a},k}} \ . \tag{64}$$

The *Zenit* delay $\mathit{\Delta}$ is a function of the receiver antenna altitude $\mathrm{h}_{\mathsf{a},k}$, and the mapping function m depends on the satellite apparent elevation $\mathrm{E}_{i,k}$. c is the speed of light.

The sensitivity vector of this model to the error-state vector can be given as

$$\mathbf{h}_{\tau_\mathrm{T},i,k} = \frac{\partial h_{\tau_\mathrm{T},i,k}\left(\mathbf{x}_k\right)}{\partial \boldsymbol{\delta}\mathbf{x}_k} = -\tfrac{1}{c}\, m(\mathrm{E}_{i,k})\mathit{\Delta}(\mathrm{h}_{\mathsf{a},k}) \begin{bmatrix} 0.133 \times 10^{-3} \\ m(\mathrm{E}_{i,k}) \end{bmatrix}^\mathsf{T} \begin{bmatrix} \frac{\partial \mathrm{h}_{\mathsf{a},k}}{\partial \boldsymbol{\delta}\mathbf{x}_k} \\ \frac{\partial \sin \mathrm{E}_{i,k}}{\partial \boldsymbol{\delta}\mathbf{x}_k} \end{bmatrix} \ , \tag{65}$$

$$\text{with} \qquad \mathrm{h}_{\mathsf{a},k} = \left\| \mathbf{r}^{E}_{\mathsf{a},k} \right\| - \mathrm{R}_{\oplus} \ , \qquad \sin \mathrm{E}_{i,k} = \left(\mathbf{e}^{E}_{\mathsf{a},k} \right)^{\mathsf{T}} \mathbf{e}^{E}_{\rho,i,k} \ , \tag{66}$$

where $\mathrm{R}_{\oplus}$ is the Earth radius, $\mathbf{r}^{E}_{\mathsf{a},k}$ is the receiver antenna position, $\mathbf{e}^{E}_{\mathsf{a},k}$ is the unit direction from the ECEF origin to the receiver antenna, and $\mathbf{e}^{E}_{\rho,i,k}$, as in (45), is the unit range vector. The non-null partial derivatives in (65) are then

$$\frac{\partial \sin \mathrm{E}_{i,k}}{\partial \boldsymbol{\delta} \mathbf{r}^{E}_{k}} = \frac{\hat{\mathbf{e}}^{E}_{\mathsf{a},k}}{\|\hat{\mathbf{r}}^{E}_{\mathsf{a},k}\|}^{\mathsf{T}} \left[\hat{\mathbf{e}}^{E}_{\rho,i,k} \times \right]^{2} - \frac{\hat{\mathbf{e}}^{E}_{\rho,i,k}}{\hat{\rho}_{i,k}}^{\mathsf{T}} \left[\hat{\mathbf{e}}^{E}_{\mathsf{a},k} \times \right]^{2} , \qquad \frac{\partial \mathrm{h}_{\mathsf{a},k}}{\partial \boldsymbol{\delta} \mathbf{r}^{E}_{k}} = \left(\hat{\mathbf{e}}^{E}_{\mathsf{a},k} \right)^{\mathsf{T}} \ , \tag{67}$$

$$\frac{\partial \sin \mathrm{E}_{i,k}}{\partial \boldsymbol{\delta\theta}^{E}_{k}} = \frac{\partial \sin \mathrm{E}_{i,k}}{\partial \boldsymbol{\delta} \mathbf{r}^{E}_{k}} \hat{\mathbf{C}}^{E_k}_{B_k} \left[\mathbf{l}^{B}_{\mathsf{a}} \times \right] , \qquad \frac{\partial \mathrm{h}_{\mathsf{a},k}}{\partial \boldsymbol{\delta\theta}^{E}_{k}} = \frac{\partial \mathrm{h}_{\mathsf{a},k}}{\partial \boldsymbol{\delta} \mathbf{r}^{E}_{k}} \hat{\mathbf{C}}^{E_k}_{B_k} \left[\mathbf{l}^{B}_{\mathsf{a}} \times \right] . \tag{68}$$

References

1. Braun B, Markgraf M, Montenbruck O (2016) Performance analysis of IMU-augmented GNSS tracking systems for space launch vehicles. CEAS Space J 8(2):117–133
2. Burke E, Rutkowski E, Rutkowski E (2008) Vehicle based independent tracking system (VBITS): a small, modular, avionics suite for responsive launch vehicle and satellite applications. In: 6th Responsive Space Conference
3. Dishel V, Mezhiritskiy E (2011) Principals of integrated Ins/GLONASS+GPS GNC systems for space launchers. Results Of Realized Missions And Future Prospects. In: 8th International ESA Conference on Guidance, Navigation & Control Systems
4. Farrell JA (2008) Aided Navigation - GPS with High Rate Sensors. McGraw-Hill, New York
5. Giannini M, Melara M, Roux C (2014) Alts localization system of vega launcher: Vv02 post-flight analysis & Gps issues for future hybrid navigation. In: 9th International ESA Conference on Guidance, Navigation & Control Systems
6. Gibbs B (2011) Advanced Kalman Filtering, Least-Squares and Modeling. Wiley, Hoboken
7. Groves PD (2008) Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. Artech House, Boston
8. Guochang X (2007) GPS Theory, Algorithms and Applications, 2nd edn. Springer, Berlin
9. Hauschild A, Markgraf M et al (2015) Results of the GNSS receiver experiment OCAM-G on Ariane-5 flight VA 219. In: 6th European Conference for Aeronautics and Space Sciences
10. Hofmann-Wellenhof B, Lichtenegger H, Wasle E (2008) GNSS - Global Navigation Satellite Systems. Springer, Wien
11. Markgraf M, Montenbruck O (2005) Phoenix-HD - a miniature GPS tracking system for scientific and commercial rocket launches. In: 6th International Symposium on Launcher Technologies
12. McKern RA (1968) A Study of Transformation Algorithms for Use in a Digital Computer. MA thesis. University of Illinois
13. Montenbruck O, Markgraf M (2004) Global positioning system sensor with instantaneous-impact-point prediction for sounding rockets. J Spacecr Rocket 41:644–650
14. NATO Standardization Agreement (STANAG) (1993) Doc. 4294, Edition 1. NATO
15. Parkinson BW, Spilker JJ Jr (1996) Global Positioning System: Theory and Applications, Vol. I. AIAA, Washington
16. Polle B, Frapard B et al (2008) Robust INS/GPS hybrid navigator demonstrator design for launch, re-entry and orbital vehicles. In: 7th International ESA Conference on Guidance, Navigation and Control Systems
17. Schlotterer M (2008) Navigation system for reusable launch vehicle. In: 31st Annual AAS Guidance and Control Conference

18. Slivinsky S, Nesbit C et al (2002) Development and demonstration of a ballistic missile range safety technology system. J Inst Navig 49:91–102
19. Steffes SR (2012) Real-time navigation algorithm for the SHEFEX2 hybrid navigation system experiment. In: AIAA Guidance, Navigation, and Control Conference
20. Steffes SR (2013) Development and Analysis of SHEFEX-2 Hybrid Navigation System Experiment. Ph.D. thesis. University of Bremen
21. Steffes SR (2014) Computationally distributed real-time dual rate Kalman filter. J Guid Control Dyn 37:1064–1068
22. Steffes SR, Theil S, Samaan MA (2014) Post-mission analysis of flight results from the SHEFEX2 hybrid navigation system. In: 9th International ESA Conference on Guidance, Navigation & Control Systems
23. Theil S, Steffes SR et al (2009) Hybrid navigation system for spaceplanes, launch and re- entry vehicles. In: 16th AIAA/DLR/DGLR International Space Planes and Hypersonic Systems and Technologies Conference
24. van Dierendonck AJ, McGraw JB, Brown RG (1984) Relationship between Allan variances and Kalman filter parameters. In: 16th Annual PTTI Meeting
25. van Graas F, Soloviev A (2004) Precise velocity estimation using a stand-alone GPS receiver. J Inst Navig 51:283–292
26. Wendel J, Trommer GF (2004) Tightly coupled GPS/INS integration for missile applications. Aerosp Sci Technol 8:627–634
27. Wendel J, Schaile C, Trommer GF (2001) Direct Kalman filtering of GPS/INS for aerospace applications. In: International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation
28. Williams A, Villa M, Puig-Suari J (2014) Platform independent launch vehicle avionics. In: 28th Annual AIAA/USU Conference on Small Satellites
29. Willms B (1999) Space integrated GPS/INS (SIGI) navigation system for space shuttle. In: 18th Digital Avionics Systems Conference
30. Zhou J, Knedlik S, Loffeld O (2012) INS/GPS for high-dynamic UAV-based applications. Int J Navig Obs 2012:1–11

Multi Sensor Fusion Based on Adaptive Kalman Filtering

Setareh Yazdkhasti and Jurek Z. Sasiadek

1 Introduction

Data fusion techniques have been widely employed in multisensory environments to fuse data from multiple sensors in order to achieve lower detection errors and higher reliability. One of the most popular approaches for navigation sensor fusion is implementing the conventional Kalman filters [1–3]. The significant difficulties in designing the conventional Kalman filters (i.e. EKF, UKF, and IEKF) for sensor fusion is incomplete prior information on covariance matrices [4]. Poor knowledge of the models may seriously reduce the Kalman filters' performance, and make the filters unstable. Numerous model proposals have been improved to address this problem over the last twenty years, including the adaptive Kalman filtering approach, which has proven to be an effective strategy for managing the limitations associated with the conventional Kalman filters. An adaptive Extended Kalman Filter for the localization of mobile robots were developed by Jetto and Longhi [5]. In their work, the data provided by sonar and odometrical sensors were fused through an adaptive EKF to provide online estimates of a robot's position. An adaptive two stage EKF for estimating unknown fault bias in an INS/GPS loosely coupled system, was proposed by Kim and Lee [6]. An adaptive EKF using artificial neural networks was developed by Stubberu et al. [7], who designed a neuro-observer that can learn system uncertainties and improve the overall performance of an uncertain control system in the state-estimator model. An adaptive UKF algorithm for target tracking with unknown process noise statistics was introduced by Shi et al. [8]. In this algorithm, a modified Sage-Husa noise statistics estimator was introduced to assess the system process noise variance adaptively. An adaptive fading UKF with Q-Adaptation for attitude estimation was introduced by Soken et al. [9].

The adaptive tuning of the Kalman filter with the fuzzy logic has been very popular for managing systems with dynamical uncertainties, particularly in the field of adap-

S. Yazdkhasti (✉) · J.Z. Sasiadek
Carleton University, 1125 Colonel By Dr, Ottawa, ON K1S 5B6, Canada
e-mail: setareh.yazdkhasti@carleton.ca

J.Z. Sasiadek
e-mail: jurek.sasiadek@carleton.ca

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_17

tive control. A fuzzy adaptive strong tracking Kalman filter for ultra-tight GPS/INS integration was introduced by D. Jwo et al. [10]. In this work, the performance of the filter improved by adaptively modifying the sub-optimal fading factor via fuzzy logic. J.Z. Sasiadek et al. [11–14], proposed an adaptive filter known as the Fuzzy Adaptive Extended Kalman Filter (FAEKF), to adapt the process and measurement noise covariance matrices. The method was based on exponential data weighting, to protect the EKF from divergence. In this work, the EKF has been modified using the fuzzy logic.

This paper focuses on the development of new integration algorithms based on the combination of the Fuzzy Logic Controllers (FLCs) and conventional Kalman filters such as the Iterated Extended Kalman Filter (IEKF) and Unscented Kalman Filter (UKF) to provide reliable and accurate navigation solutions. The proposed algorithms are based on the correction of both the process noise covariance matrix **Q**, and the measurement error covariance **R**. The FLCs are implemented to adjust the exponential weighting of weighted UKF and IEKF and protect the filters from divergence. The rest of the paper is organized as follows. Section 2 describes two proposed adaptive Kalman filtering approaches. Section 3 introduces the Fuzzy Logic Controllers. In order to validate the effectiveness of the proposed adaptive algorithms, simulation results are discussed in Sect. 4 and finally, in Sect. 5, the conclusions of this work are given.

2 The Adaptive Estimation Algorithms

2.1 System Description

The non-linear dynamic and measurement models are given by:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{1}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where the vectors $\mathbf{x}_k \in \Re^8$, represents the state of the system at the time point k. The process noise is given by $\mathbf{w}_k$, and $\mathbf{z}_k \in \Re^4$, corresponds to the observed measurement signal, driven by Gaussian white noise $\mathbf{v}_k$.

In order to protect the conventional Kalman filters from divergence when there are uncertainties in the system noise covariances, exponential data weighting is applied. Two new adaptive formulations, weighted Iterated Extended Kalman Filter and weighted Unscented Kalman Filter can be described as follow.

2.2 Weighted Iterated Extended Kalman Filter

The model and implementation equations for the weighted Iterated Extended Kalman Filter are defined as the following recursive equations.

2.2.1 Initialization

$$\hat{\mathbf{x}}_0 = E(\mathbf{x}_0), \qquad \mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \tag{3}$$

2.2.2 Prediction

The predicted state can be defined as:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \tag{4}$$

The covariance matrices of the adaptive IEKF defined as:

$$\mathbf{R}_k = \alpha^{-2(k+1)}\mathbf{R} \tag{5}$$

$$\mathbf{Q}_k = \alpha^{-2(k+1)}\mathbf{Q} \tag{6}$$

where, $\alpha \geq 1$. For $\alpha > 1$, as the time increases the covariance matrices decrease which means the recent data has more credibility, due to the exponential decreased noise covariance with time. When $\alpha = 1$, the filter is acting like a regular IEKF. It should be noted that α is the output of the fuzzy controllers.

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^-\mathbf{F}_{k-1}^T + \alpha^{-2(k+1)}\mathbf{Q} \tag{7}$$

where, $\mathbf{F}_{k-1}$, the linear approximation equation can be present in form of:

$$\mathbf{F}_{k-1} \approx \frac{\partial \mathbf{f}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_k} \tag{8}$$

By defining the weighted covariance as

$$\mathbf{P}_k^- = \alpha^{-2k}\mathbf{P}_{k|k-1} \tag{9}$$

By calculating $\hat{\mathbf{x}}_k$, $\mathbf{K}_k$, $\mathbf{P}_k$ at each iteration about the most recent estimate. The Kalman gain can be computed as:

$$\mathbf{K}_{k,i} = \mathbf{P}_k^-\mathbf{H}_{k,i}^T(\mathbf{H}_{k,i}\mathbf{P}_k^-\mathbf{H}_{k,i}^T + \alpha^{-2(k+1)}\mathbf{R})^{-1} \tag{10}$$

The superscript i, $(i = 0, 2, \ldots, \tau)$, is the number of iteration steps. By using (9), the Kalman gain can be rewritten as:

$$\mathbf{K}_{k,i} = \mathbf{P}_{k|k-1}\mathbf{H}_{k,i}^{T}(\mathbf{H}_{k,i}\mathbf{P}_{k|k-1}\mathbf{H}_{k,i}^{T} + \mathbf{R}/\alpha^{2})^{-1} \tag{11}$$

where

$$\mathbf{H}_{k,i} \approx \frac{\partial \mathbf{h}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k,i}} \tag{12}$$

The predicted measurement estimation can be rewritten as:

$$\hat{\mathbf{x}}_{k,i+1} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{k,i}(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,i}) - \mathbf{H}_{k,i}(\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k,i})) \tag{13}$$

where, $\hat{\mathbf{x}}_{k,i}$ presents the estimate at time point k and i_{th} iteration. The iteration process is initialized with $\hat{\mathbf{x}}_{k,0} = \hat{\mathbf{x}}_{k|k-1}$.

And the posterior covariance matrix defined as:

$$\mathbf{P}_{k,i} = (\mathbf{I} - \mathbf{K}_{k,i}\mathbf{H}_{k,i})\mathbf{P}_{k|k-1} \tag{14}$$

The iterative process will not be stopped until a certain termination condition is met.

2.3 Weighted Unscented Kalman Filter

2.3.1 Initialization

$$\hat{\mathbf{x}}_0 = E(\mathbf{x}_0), \qquad \mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \tag{15}$$

2.3.2 Prediction

For the L elements state vector, a set of $(2L + 1)$ sigma-points are created according to the following:

$$\boldsymbol{\chi}_{k-1} = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \sqrt{(L+\lambda)}\sqrt{\mathbf{P}_{k-1}} \quad \hat{\mathbf{x}}_{k-1} - \sqrt{(L+\lambda)}\sqrt{\mathbf{P}_{k-1}}] \tag{16}$$

where each column of $\boldsymbol{\chi}_{k-1}$, represents a sigma-point. $\sqrt{\mathbf{P}_{k-1}} = chol(\mathbf{P}_{k-1})$ is the square root of the state error covariance [15], and the scaling parameter λ defined as:

$$\lambda = \sigma^2(L + \kappa) - L \tag{17}$$

where, σ, $1e^{-4} \leq \sigma \leq 1$, determines the size of the sigma-points distribution and κ influences the accuracy of the approximation [16]. Once the sigma-points have been generated, each point is propagated through out the non-linear state equation as:

$$\boldsymbol{\chi}_{k|k-1}^{(i)} = \mathbf{f}(\boldsymbol{\chi}_{k-1}, \mathbf{u}_k), \qquad i = 0, \ldots, 2L \tag{18}$$

The mean and covariance are approximated using a weighted mean and covariance of the transformed points as:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} \boldsymbol{\eta}_i^{(m)} \boldsymbol{\chi}_{k|k-1}^{(i)} \tag{19}$$

$$\mathbf{P}_{k|k-1} =$$

$$\sum_{i=0}^{2L} \boldsymbol{\eta}_i^{(c)} (\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})^T + \alpha^{-2(k+1)}\mathbf{Q} \tag{20}$$

where the mean weight vector $\boldsymbol{\eta}_i^{(m)}$, and the covariance weight vector $\boldsymbol{\eta}_i^{(c)}$ associated with the i_{th} point are defined as:

$$\boldsymbol{\eta}_i^{(c)} = \boldsymbol{\eta}_i^{(m)} = 1/(2(L + \lambda)), \quad i = 1, \ldots, 2L \tag{21}$$

$$\boldsymbol{\gamma}_{k|k-1}^{(i)} = \mathbf{h}(\boldsymbol{\chi}_{k|k-1}^{(i)}) \tag{22}$$

Then the mean of the measurement vector is calculated as:

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2L} \boldsymbol{\eta}_i^{(m)} \boldsymbol{\gamma}_{k|k-1}^{(i)} \tag{23}$$

Covariance and cross-covariance matrices of the proposed adaptive UKF are defined as:

$$\mathbf{P}_k^{yy} = \sum_{i=0}^{2L} \eta_i^{(c)} (\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1})(\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1})^T + \alpha^{-2}\mathbf{R} \tag{24}$$

$$\mathbf{P}_k^{xy} = \alpha^{-2(k-1)} \sum_{i=0}^{2L} \eta_i^{(c)} (\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1})^T \tag{25}$$

Similar to the weighted IEKF, When $\alpha = 1$, the filter is acting like a regular UKF. For $\alpha > 1$, as the time increases the covariance matrices decrease.

2.3.3 Update

$$\mathbf{K}_k = \mathbf{P}_k^{xy}(\mathbf{P}_k^{yy})^{-1} \tag{26}$$

The Kalman gain $\mathbf{K}_k$, is then used to update the state and covariance estimates as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \tag{27}$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_k^{yy}\mathbf{K}_k^T \tag{28}$$

3 Fuzzy Logic Controllers

The conventional Kalman filters provide an effective means of estimating the state of a system from noisy measurements when the covariances of the system are known, and the system is well defined. However, in some cases, there are uncertainties in the system noise covariances, which can cause filters to become unstable. Adaptive tuning of the conventional Kalman filters via fuzzy logic has been one of the promising strategies to protect the filters from divergence when dealing with parameter uncertainty and non-white process noise [13]. In order to adjust the filters, the FLCs continuously monitors and tunes the noise level in the filters, internal model. The residuals of Kalman filter should be zero mean white noise process; if not, divergence will happen. Hence, in this paper, the covariance and mean values of the residuals are used as inputs to the FLCs to decide the degrees of divergence.

The block diagram for the GPS/INS navigation sensor fusion using the FAUKF and FAIEKF is shown in Fig. 1. The proposed fuzzy adaptive Kalman filters have been validated for two different cases.

3.1 Fuzzy Logic Adaptive System for Parameter Uncertainty

The uncertain or time varying parameters in **Q** and **R** matrices make the conventional Kalman filters diverge or coverage to a large bound. When the filter does not work well, the FLCs would apply a suitable weighting factor to Enhance the accuracy of the filter. Two groups of fuzzy controllers have been defined for parameter uncertainty.

3.1.1 First Fuzzy Controller

In this controller, the mean values and covariance of the residuals are inputs to the FLC to determine the degree of divergence. The exponential weighting α is the first

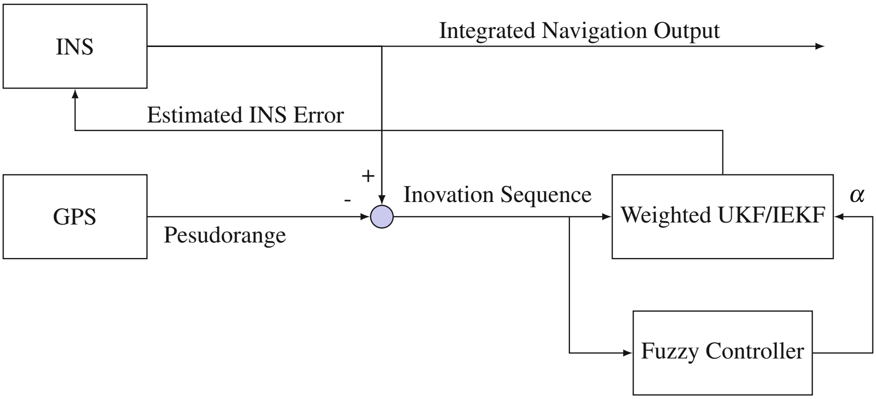


Fig. 1 Block diagram representation of fuzzy adaptive Kalman filter for GPS/INS integration

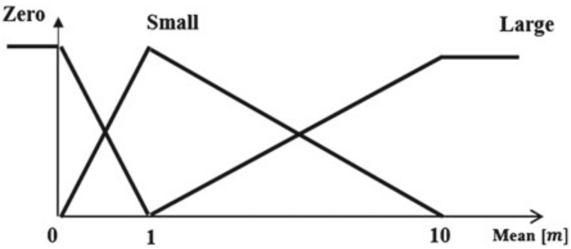


Fig. 2 Mean value membership functions for parameter uncertainty [J.Z. Sasiadek et al. [13]]

fuzzy controller output. The FLC will select the suitable α to optimally adapt the Kalman filter. The membership function of FLC inputs (i.e. the mean and covariance of residuals) and the output α are illustrated in Figs. 2, 3 and 4, respectively. The characteristics of a fuzzy system are highly dependent on the relevant rules. The proposed fuzzy logic controller used 9 rules as shown in Table 1.

3.1.2 Second Fuzzy Controller

The second controller was designed to detect the changes in the **R** matrix, and adapt the filter accordingly. The measurement covariance matrix, **R**, is related to the residual's covariance; thus, any changes in the measurement covariance matrix can alter the covariance of the residuals. With this controller, the UKF and IEKF is adapted by selecting the appropriate scale. For example, the FLC applies a large scale to adjust the α if it determines that the covariance of the residual is greater than expected.

$$\alpha_{k+1} = (\alpha_k - 1) * scale + 1 \tag{29}$$

The 9 rules for this fuzzy controller are shown in Table 2.

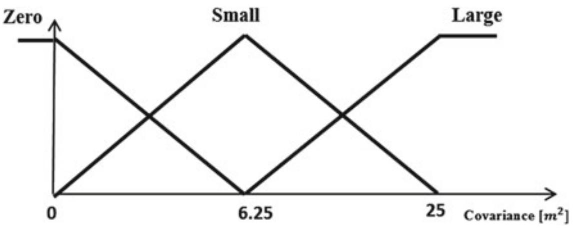


Fig. 3 Covariance membership functions for parameter uncertainty [J.Z. Sasiadek et al. [13]]

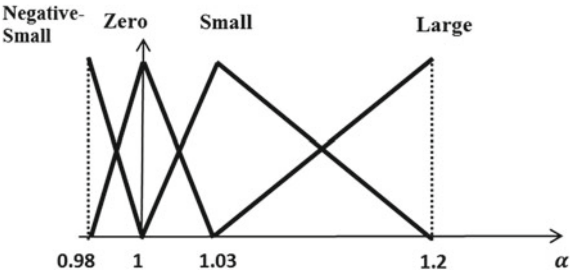


Fig. 4 α membership functions for parameter uncertainty [J.Z. Sasiadek et al. [13]]

Table 1 Rule table of α for parameter uncertainty

α	Mean value		
	Z	S	L
Z	Z	S	S
S	S	L	S
L	Z	NS	NS

Z – Zero; S – Small; NS– Negative Small; L– Large

Table 2 Rule table of *scale* for parameter uncertainty

scale	Mean value		
	Z	S	L
Z	Z	Z	Z
S	S	S	S
L	L	S	Z

Z – Zero; S – Small; L– Large

3.2 Fuzzy Logic Adaptive System for Non-white Process Noise

The conventional Kalman filters require that both, the process noise $\mathbf{w}_k$, and the measurement noise $\mathbf{v}_k$ are zero-mean white noise with known covariance $\mathbf{Q}$ and $\mathbf{R}$. In practice, sometimes the process noise could be correlated with itself. Implementing the FLC is one alternative to this problem

The membership functions for this fuzzy control are shown in Figs. 5, 6 and 7 respectively.

Table 3 Rule table of α for non-white noise

α	Mean value		
	Z	S	L
Z	S	Z	Z
S	Z	L	M
L	L	M	Z

Z – Zero; S – Small; M– Medium; L– Large

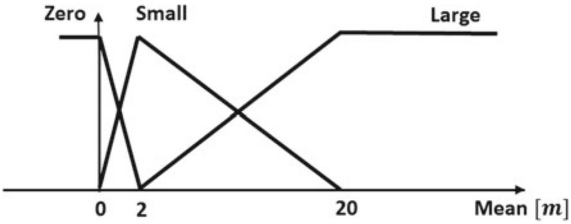


Fig. 5 Mean value membership functions for non-with process noise [J.Z. Sasiadek et al. [13]]

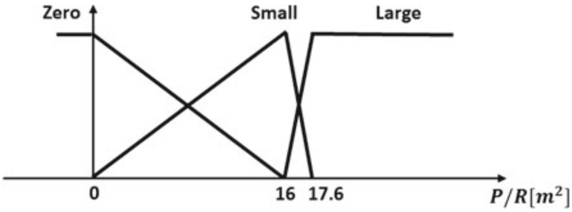


Fig. 6 Covariance membership functions for non-with process noise [J.Z. Sasiadek et al. [13]]

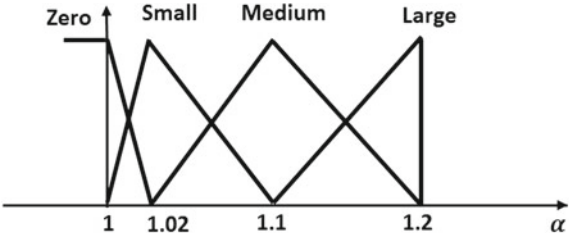


Fig. 7 α membership functions for non-with process noise [J.Z. Sasiadek et al. [13]]

There are 9 rules for the FLC in the present of non-white noise, Table 3, hence, little computational time is required.

4 Simulation Results

Numerical simulation has been done to validate the performance of the proposed FAUKF and FAIEKF in comparison with FAEKF and the conventional EKF, UKF and IEKF approaches for INS/GPS integration when dealing with non-white noise and uncertain or time varying parameters are considered to exist in the process and

Table 4 The position errors with the non-white noise inputs

Mean of INS [m]	Position RMS Error			Position RMS Error		
	$x = 2$	$y = 0.5$	$z = 1$	$x = 1$	$y = 1$	$z = 1$
	X	Y	Z	X	Y	Z
FAUKF	3.20	1.28	2.53	1.5	1.7	2.4
FAEKF	7.23	3.34	5.83	3.19	3.53	4.14
FAIEKF	7.03	3.14	5.03	2.9	3.13	3.94
EKF	24.91	8.58	8.36	14.82	10.67	6.7
IEKF	24.51	7.51	7.78	14.6	10.50	5.6
UKF	10.23	14.71	14.09	5.1	3.6	6.32

measurement noise matrices. More detailed descriptions of weighted EKF and standard Kalman filters approaches for GPS/INS integration are given in J.Z. Sasiadek et al. [13] and S. Yazdkhasti et al. [17, 18], respectively (Fig. 5).

The state vectors used in the simulation included three states for INS position errors, three for INS velocity errors and two states for GPS range drift and range bias as:

$$\mathbf{x}_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, C\Delta t, C\dot{\Delta} t] \tag{30}$$

where, x points east, y points north and z is the attitude, $C\Delta t$, $C\dot{\Delta} t$ represent GPS range and drift states (Fig. 6).

For the first part of the simulation, the process noise $\mathbf{w}_k$ assumed as a non-zero mean process noise. The white noise with a standard deviation of 4 m was applied to the GPS measurements. The simulation was done for different INS mean values ($\mathbf{w}_k$), for the East (x), North (y), and Altitude (z) respectively (Fig. 7).

The summary of root mean square (RMS) errors of the six GPS/INS configurations provided in Table 4. As it is indicated, the fuzzy adaptive filters clearly improved the performance of the conventional Kalman filters. The position errors of FAEKF, FAUKF and FAIEKF are much smaller than that of EKF, UKF, and IEKF. Among the six approaches, the FAUKF demonstrates superior navigation accuracy performance in comparison with other filters.

The residuals of the six filters are shown in Fig. 8. The residual is the difference between the best measurement prediction based on the filter's internal mode and the actual measurement; hence, it can be used to evaluate filter's performance. From Fig. 8 it could be noticed that the residual of the EKF, UKF, and IEKF have a large drift, while the residual mean value of FAEKF, FAUKF, and FAIEKF are smaller than that of EKF, UKF, IEKF.

For the second part of the simulation, the developed algorithms have been tested and evaluated for various parameter uncertainties. The white noise with a standard deviation of 5 m was applied to the GPS measurements. The INS standard deviations are 0.006, 0.006, and 0.0006 m for the East (x), North (y), and Altitude (z),

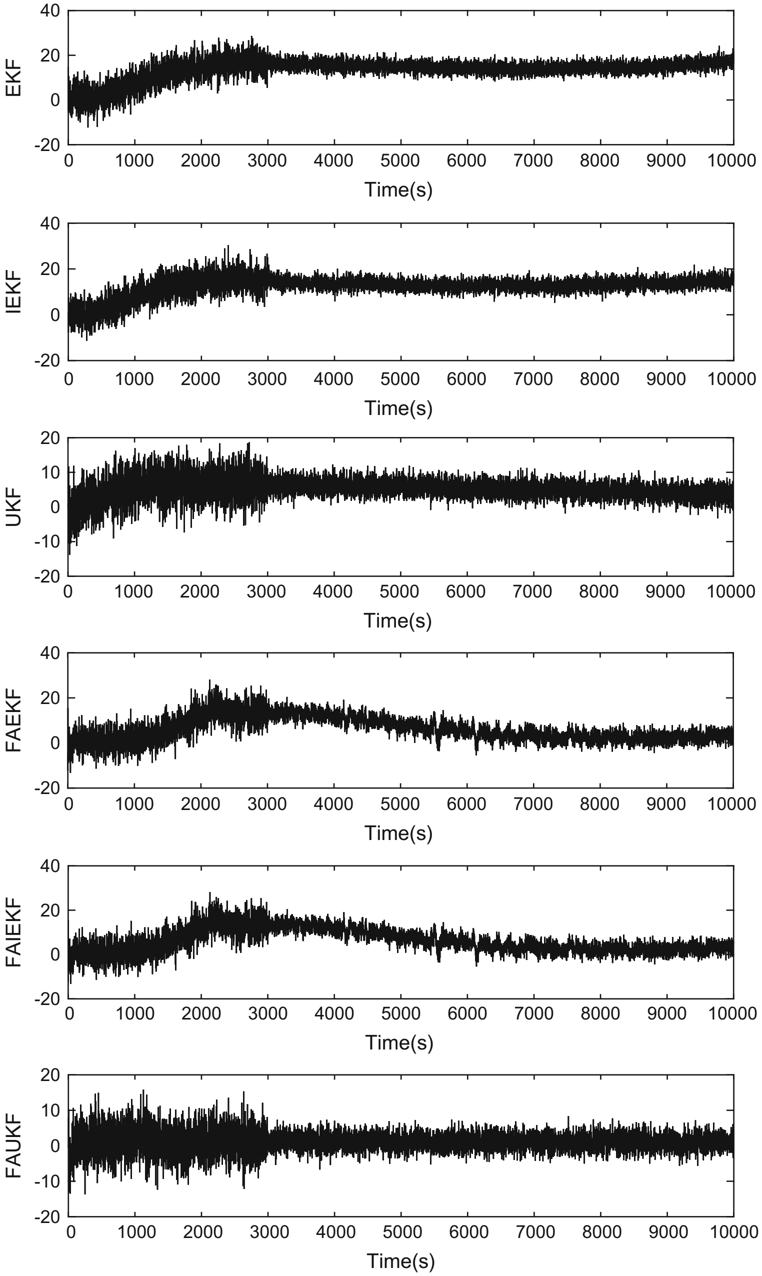


Fig. 8 Inovation sequence (residual)

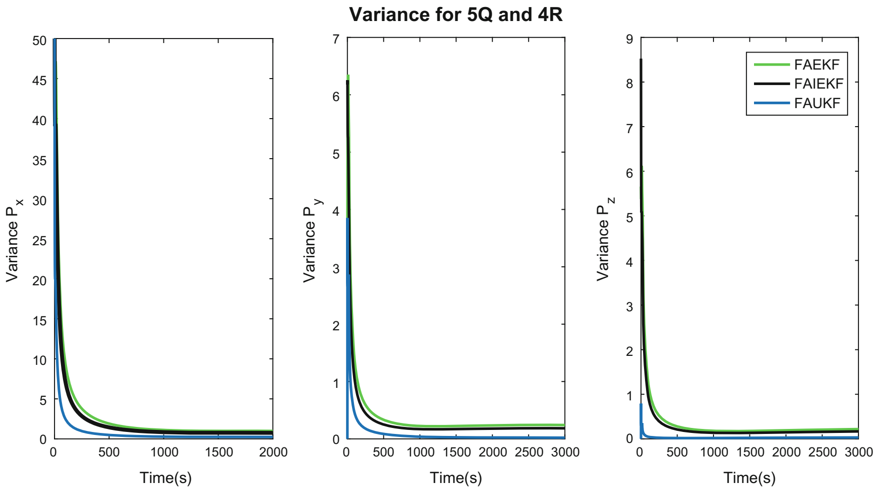


Fig. 9 Variance of the Adaptive Filters for 5Q and 4R

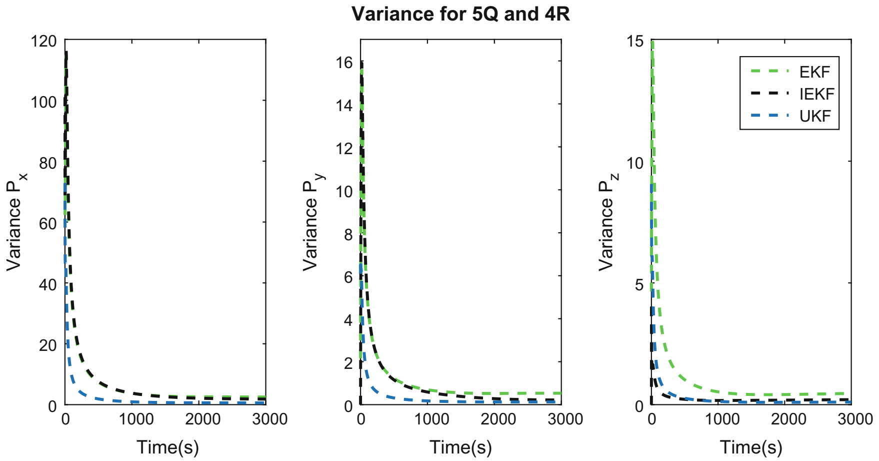


Fig. 10 Variance of the convential filters for 5Q and 4R

respectively. Figures 9 and 10 illustrates the covariance of Fuzzy adaptive filters and standard Kalman filters when high uncertainty, 5**Q** and 4**R**, exists. 5**Q** and 4**R** mean the real time parameters are 5 and 4 times as large as the designed **Q** and **R**. The simulation was repeated for different covariance values (5**Q**, **R**), Figs. 11 and 12 and (**Q**, 4**R**), Figs. 13 and 14.

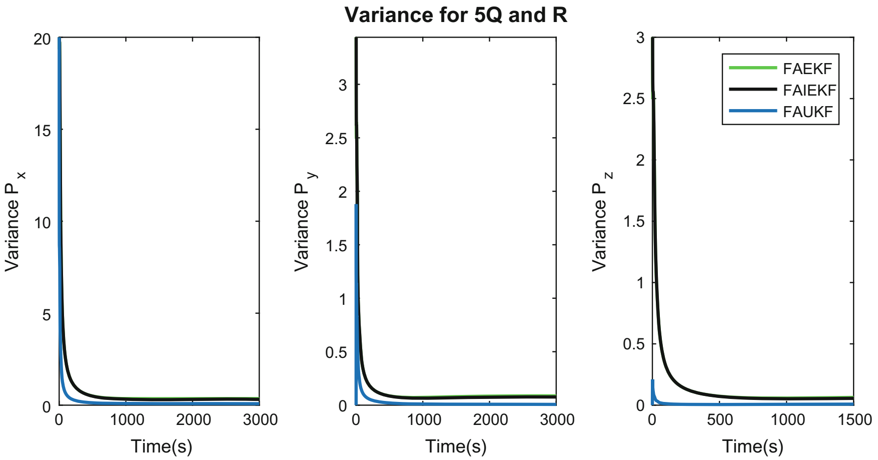


Fig. 11 Variance of the adaptive filters for 5Q and R

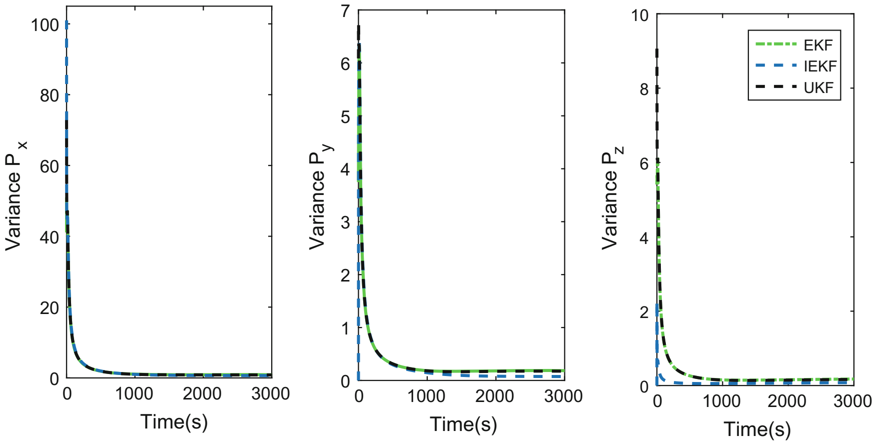


Fig. 12 Variance of the convential filters for 5Q and R

Due to better treatment of parameter uncertainties, the FAUKF and FAIEKF has shown performance improvement when compared to the conventional UKF and IEKF, respectively.

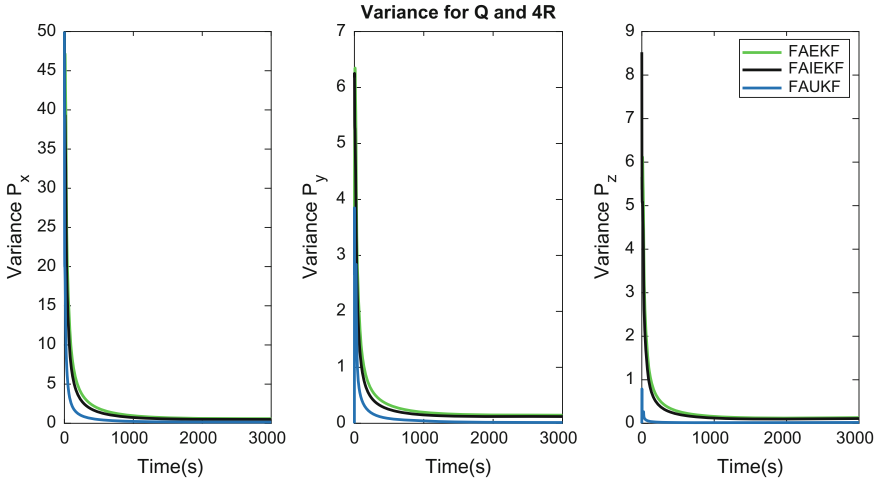


Fig. 13 Variance of the adaptive filters for Q and 4R

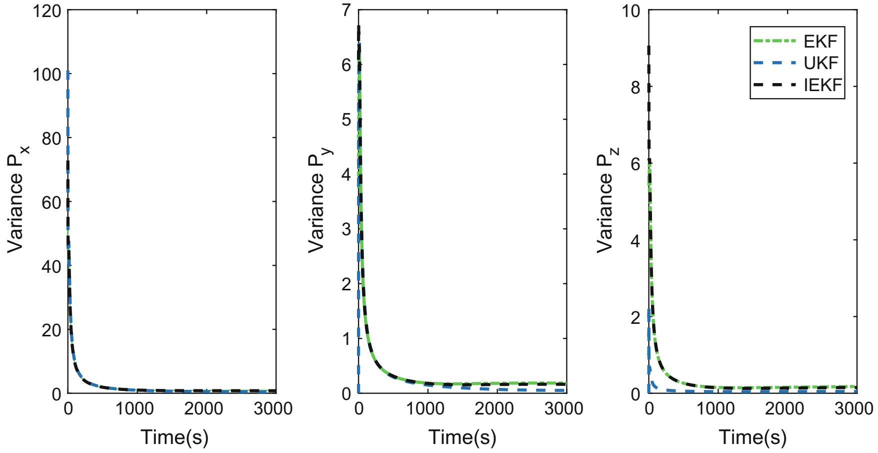


Fig. 14 Variance of convential filters for Q and 4R

5 Conclusion

This paper develops new data fusion techniques for GPS/INS integration by incorporating Fuzzy Logic controller (FLC) and the conventional Kalman filters. The Fuzzy Adaptive Unscented Kalman Filter (FAUKF) and a Fuzzy Adaptive Iterated Extended Kalman Filter (FAIEKF) have been developed to improve the Unscented Kalman Filter and Iterated Extended Kalman filter Performance, respectively. By monitoring the residual, the FLC can detect uncertainty or time varying Parameters in both, the process, and the measurement noise covariance matrices. And then adapt the Kalman gain in real time to improve the IEKF and UKF performance and avoid the filters from divergence. Performance comparisons on FAUKF, FAIEKF, FAEKF, UKF, EKF, and IEKF have been conducted. The simulation results show the proposed FAUKF leads to very accurate results.

Appendix

GPS Satellite Geometry

Four pseudo range measurements are used as a measurement model of the Kalman filter.

$$h_1 = \sqrt{(X_1 - x)^2 + (Y_1 - y)^2 + (Z_1 - z)^2} + C\Delta t_1$$

$$h_2 = \sqrt{(X_2 - x)^2 + (Y_2 - y)^2 + (Z_2 - z)^2} + C\Delta t_2$$

$$h_3 = \sqrt{(X_3 - x)^2 + (Y_3 - y)^2 + (Z_3 - z)^2} + C\Delta t_3$$

$$h_4 = \sqrt{(X_4 - x)^2 + (Y_4 - y)^2 + (Z_4 - z)^2} + C\Delta t_4 \tag{31}$$

where, (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), (X_3, Y_3, Z_3), (X_4, Y_4, Z_4) are the positions of the four GPS satellites respectively, and (x, y, z) are the position of the vehicle. The GPS satellite assumed to be in circular orbits.

$$X_j = R[cos\theta_j cos\Omega_i - sin\theta_j sin\Omega_j cos55°] \tag{32}$$

$$Y_j = R[cos\theta_j sin\Omega_j + sin\theta_j sin\Omega_j cos55°] \tag{33}$$

$$Z_j = R[sin\theta_j sin55°] \tag{34}$$

where

$$\theta_j = \theta_0 + T\frac{360}{43082}\,\text{deg} \quad j = 1, \ldots, 4$$

$$\Omega_j = \Omega_0 - T\frac{360}{86164}\,\text{deg}$$

$$R = 26560000\,\text{m} \tag{35}$$

Table 5 Satellite parameters

GPS satellite	Ω_0°	θ_0°
Satellite 1	326	68
Satellite 2	26	340
Satellite 3	146	198
Satellite 4	86	271

$$\mathbf{P}_0 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \tag{36}$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 t^3/3 & \sigma_x^2 t^2/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_x^2 t^2/2 & \sigma_x^2 t^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_y^2 t^3/3 & \sigma_y^2 t^2/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_y^2 t^2/2 & \sigma_y^2 t^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_z^2 t^3/3 & \sigma_z^2 t^2/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_z^2 t^2/2 & \sigma_z^2 t^2 & 0 & 2/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & (S_a t + \frac{S_b t^3}{3}) c^2 & \frac{S_b t^2 c^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{S_b t^3 c^2}{2} & S_b \end{bmatrix} \tag{37}$$

where, σ_x, σ_y, and σ_z represent standard deviations associated with x, y, and z, respectively. t is the sample time, c is the speed of light $S_a = 0.4(10)^{-18}$, standard deviation of clock offset, and $S_b = 1.58(10)^{-18}$, is the standard derivation associated with velocity (Table 5).

References

1. Zhao H, Wang Z (2012) Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. IEEE Sens J 12(5):943–953
2. Rigatos G, Tzafestas S (2007) Extended Kalman filtering for fuzzy modelling and multi-sensor fusion. Math Comput Model Dyn Syst 13:251–266
3. Carlson NA (1994) Federated Kalman filter simulation results. J Inst Navig 41:297–321
4. Escamilla-Ambrosio PJ, Mort N (2002) Multi-sensor data fusion architecture based on adaptive Kalman filters and fuzzy logic performance assessment. Presented in the fifth international conference on information fusion 2:1542–1549

5. Jetto L, Longhi S, Venturini G (1999) Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. IEEE Trans Robot Autom 15(2):219–229
6. Kim KH, Lee JG, Park CG (2009) Adaptive two-stage extended Kalman filter for a fault-tolerant INS-GPS loosely coupled system. IEEE Trans Aerosp Electron Syst 45(1):125–37
7. Stubberud SC, Lobbia RN, Owen M (1995) An adaptive extended Kalman filter using artificial neural networks. In: Presented in the 34th IEEE conference on decision and control, vol 2, pp 1852–1856
8. Shi Y, Han C, Liang Y (2009) Adaptive UKF for target tracking with unknown process noise statistics. In: Presented in 12th International conference on information fusion, FUSION'09, pp 1815–1820
9. Soken HE, Hajiyev C (2011) Adaptive fading UKF with Q-adaptation: application to picosatellite attitude estimation. J Aerosp Eng 26(3):628–36
10. Jwo DJ, Yang CF, Chuang CH, Lee TY (2013) Performance enhancement for ultra-tight GPS/INS integration using a fuzzy adaptive strong tracking unscented Kalman filter. Nonlinear Dyn 73(1–2):377–95
11. Sasiadek JZ, Wang Q, Zeremba MB (2000) Fuzzy adaptive Kalman filtering for INS/GPS data fusion. In: Proceedings of the IEEE international symposium intelligent control, pp 181–186
12. Sasiadek JZ (2002) Sensor fusion. Annu Rev. Control 26(2):203–28
13. Sasiadek JZ, Wang Q (2003) Low cost automation using INS/GPS data fusion for accurate positioning. Robotica 21(03):255–60
14. Sasiadek JZ, Hartana P (2004) Sensor fusion for navigation of an autonomous unmanned aerial vehicle. Presented in IEEE international conference on robotics and automation ICRA 4:4029–4034
15. Wan EA, Van Der Merwe R (2000) The unscented Kalman filter for nonlinear estimation. In: Presented in systems for signal processing, communications, and control symposium. AS-SPCC, pp 153-158
16. Wan EA, Van Der Merwe R (2000) The unscented Kalman filter for nonlinear estimation. In: Presented in adaptive systems for signal processing, communications, and control symposium. AS-SPCC, pp 153–158
17. Yazdkhasti S, Sasiadek JZ, Ulrich S (2016) Performance enhancement for GPS/INS fusion by using a fuzzy adaptive unscented Kalman filter. In: Presented in 21st international conference methods and models in automation and robotics (MMAR), pp 1194–1199
18. Yazdkhasti S, Sasiadek JZ (2017) Sensor fusion using fuzzy adaptivr uncented Kalman filter for uncertain systems. In: Accepted in 11th annual IEEE international system conference. Montreal, Canada

Aircraft Damage Pattern Recognition Using Aerodynamic Coefficients and Fuzzy Logic

Y. Zhang, C.C. de Visser, Q.P. Chu and E.J. van Kampen

1 Introduction

The maintenance of safe and reliable performance of flight vehicles is always highly demanded, for the flight transportation system are an indispensable support for the modern society and world economy. However, in spite of high-level redundancy systems, aircraft still has the risk of faults and failures in three major parts, which are generally referred to as sensors, actuators and airframe. Specifically, fault detection and isolation (FDI) methods on sensors and actuators have been intensively used and well developed [1]. The integration of these technologies generate multiple choices and attractive issues in preventing accidents due to degrading mechanical properties like fatigues, cracks and debonding, which are generally categorized as structural health monitoring (SHM). A multitude of various SHM methods as well as nice reviews can be found in literature [2, 3].

Generally the approaches taken in SHM are classified into physics-based type and data-based type [3]. The conventional physics-based methods rely on identification techniques such as the estimation of structure stiffness and model parameters, which are considered deterministic without uncertainties. The problem with this method is that it's difficult to evaluate the reliability of the estimated damage case. On the other hand, the data-driven approach takes more considerations into uncertainties, which relies on system measurements for training and assessment. Nevertheless, the data-based method sometimes may lack insights into the nature of the damage, which makes it less 'physical', or straightforward. Various methods have been developed

Y. Zhang (✉) · C.C. de Visser · Q.P. Chu · E.J. van Kampen
Control and Simulation Section, Faculty of Aerospace Engineering,
Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
e-mail: y.zhang-9@tudelft.nl

C.C. de Visser
e-mail: C.C.deVisser@tudelft.nl

Q.P. Chu
e-mail: Q.P.Chu@tudelft.nl

E.J. van Kampen
e-mail: E.vanKampen@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_18

to find a balance between the two approaches and enhance the confidence in damage diagnosis in the existence of uncertainties. However, successful applications of these methods in flight vehicles face a lot of technical challenges. One difficulty is the acquisition of extensive training data from repetitive experiments and onboard sensors. Also, the knowledge of damage location and damage type are not always well known a prior, and the operational and environmental conditions sometimes vary significantly due to diverse uncertainties [3]. So the main focus of this paper is on dealing with aircraft structural damage under uncertainties.

One of the approaches dealing with uncertainties as well as the lack of deterministic knowledge is based on fuzzy logic, which provides a potential way of sufficient training and accurate pattern recognition based on imperfect data. The damage type can be determined by fuzzy set method through the assignment of membership functions to each input. Various literature [2–10] can be found using fuzzy logic to detect and assess structural damage of composites in various applications of industries. Most of the methods use non-destructive vibration-based data like natural frequencies, material properties like Young's moduli, mode shape curvatures and stiffness based on finite element models. For aircraft, however, there are no such onboard sensors to provide real-time data for monitoring and diagnosing. In order to determine which part of the aircraft (e.g. fuselage, wings, tails) is under crisis, new damage indicators need to be defined using information that already existed onboard. Therefore, in this paper a different way of detecting and estimating aircraft structural damage is proposed by using aerodynamic coefficients of moments and forces as featured inputs to the fuzzy logic system.

2 Modeling of Aircraft Structural Damage

One major bottleneck for damage assessment of aircraft is that repeated destructive experiments are hardly feasible, thus there's little real flight data for analysis. However, with the increasing concern for the safety of civil airliners, a number of experiments on subscale models in windtunnel and Computational Fluid Dynamics (CFD) software have been conducted to satisfy the expanding need for multiple sources of data. Earlier researches done by NASA Langley Research Center are a series of wind tunnel tests on a swept-wing airplane model to determine the effects of simulated wing damage on the aerodynamic characteristics of the model at Mach numbers larger than one. Wing damage was simulated by removal of either a leading-edge or a trailing-edge portion or an entire wing panel [11]. In recent years, the Generic Transport Model (GTM), a 5.5% scale model of a commercial airplane has been the subject of a series of extensive wind tunnel tests and CFD experiments [12] undertaken by NASA for the exploration of loss-of-control events involving various contributions such as stall/ high-angle-of-attack, airframe damage and icing [13].

With the aid of these data and reports, the aerodynamic characteristics of the partially damaged aircraft can be studied, which provides the theoretic foundation of our paper. The effect of structural damage is mainly a combination of aerodynamic change, mass property shift and control degradation. The control effectiveness can be evaluated in a relatively independent way through the isolation of actuator faults by advanced actuators fault detection systems. Mass properties, seem not have a substantial effect on flight characteristics compared with the impact of aerodynamic forces and moments, according to an experiment conducted by Shah [14], in which a large, asymmetric mass change (physical separation of an engine) was modeled. In this paper we mainly focus on the aerodynamic effects of structural damage based on the wind tunnel tests conducted by Shah et al. [14, 15], in which the damage in the form of partial or total tip loss is applied to three major parts of the aircraft that provides aerodynamic forces and moments: the horizontal tails, the vertical tail, and the wings [16]. In order to correctly identify the damage type, we need to model the damage in a simple but general way that can reflect the trend of experimental damage data.

2.1 Horizontal Stabilizer Damage

The horizontal stabilizers provide the only stabilizing contribution to pitching moment. So the damage to horizontal stabilizers often causes significant changes in both static and dynamic longitudinal stability. When damage occurs, a steady decrease in pitching moment is most likely to happen. A trend from negative to positive (unstable) in C_{m_α} with increasing area of damage is observed in the wind tunnel data. As an important indication of dynamic stability, the pitching damping C_{m_q} over the angle of attack range is reduced with the scale roughly proportional to the ratio of tip loss.

2.2 Vertical Tail Loss

The effect of vertical tail damage on directional characteristics is similar to that of horizontal tail damage on pitch axis, and it rarely induces incremental rolling moments since the damage is basically symmetric. The change in lateral static and dynamic stability is seen as a steady decrease in yawing moment at the same angle of sideslip with increasing size of tip loss. As to static stability, the value of C_{n_β} experiences a progressive decrease proportional to the scale of damage at the same angle of attack, but the change with respect to α remains almost the same. Yaw damping steadily decreases and approaches to zero (unstable limit) with the increase of the damage area.

2.3 Wing Damage

The wing damage is different from tail damage in the progressive reduction of lift curve slope versus angle of attack. More importantly, wing tip loss also results in incremental rolling moment due to unequal normal force contributions from the left wing to the right wing, which are roughly proportional to damaged scale at given angle of attack.

2.4 Aerodynamic Coefficients as the Damage Indicator

As is explained before, the conventional damage indicators for structural monitoring and diagnosing are not available due to limited onboard sensors. Instead, the measured translational accelerates $A_{(x,y,z)_m}$ and the rotational rates p_m, q_m, r_m from inertial navigation system (INS) are used to calculate the dimensionless forces and moments of the aircraft [17]:

$$C_X = \frac{mA_x}{1/2\rho V^2 S} \quad C_Y = \frac{mA_y}{1/2\rho V^2 S} \quad C_Z = \frac{mA_z}{1/2\rho V^2 S} \quad C_L = C_X \sin\alpha - C_Z \cos\alpha \tag{1}$$

$$\begin{aligned} C_l &= \frac{\dot{p} I_{xx} + qr\left(I_{zz} - I_{yy}\right) - (pq + \dot{r}) I_{xz}}{\frac{1}{2}\rho V^2 S b} \\ C_m &= \frac{\dot{q} I_{yy} + rp\,(I_{xx} - I_{zz}) + \left(p^2 - r^2\right) I_{xz}}{\frac{1}{2}\rho V^2 S \bar{c}} \\ C_n &= \frac{\dot{r} I_{zz} + pq\left(I_{yy} - I_{xx}\right) + (qr - \dot{p}) I_{xz}}{\frac{1}{2}\rho V^2 S b} \end{aligned} \tag{2}$$

where C_L denotes the dimensionless aerodynamic lift force.

These calculated forces and moments are closely related to the integrity of aircraft components. Moreover, based on the discussion on wind tunnel results, the structural damage directly influences the value of aerodynamic coefficients, which means that we can use estimated aerodynamic coefficients as the damage indicators through least squares based on the following model structure:

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{q\bar{c}}{V} + C_{L_{\delta_e}}\delta_e \\ C_Y &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V} + C_{Y_r}\frac{rb}{2V} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\ C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r + C_{l_\alpha}\alpha \\ C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{q\bar{c}}{V} + C_{m_{\delta_e}}\delta_e \\ C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \end{aligned} \tag{3}$$

The damage indicators are listed in Table 1. It is noticed that different aerodynamic coefficients are picked on which different damage cases have the most dominant influence. That is how we distinguish between two different damage locations based on the value of their unique damage indicators.

3 Formulation of Fuzzy Logic System

Unlike what we listed in Table 1, the actual damage cases are never deterministic. Even if we have all the time and condition to collect data from a given airplane, it would be impossible to obtain all the information to predict all the possible phenomena. Under the situation of uncertainties, what we can do is to make generalizations and predictions of the current and future system behaviors based on the past experience and data [3]. Specifically, we need to use inductive reasoning and inference to find a balance between the vague, uncertain world and its precise, deterministic estimation. By using linguistic variables, fuzzy logic system is a marvelous tool that provides a possible way of dealing with vagueness and imprecision of information from limited experiments. In this paper, a fuzzy logic system is designed and integrated in the problem of damage assessment of the aircraft using aerodynamic coefficients as damage indicators.

As is illustrated in Fig. 1, a fuzzy Inference system (FIS) is basically composed of five functional blocks, in which the rule base contains a number of fuzzy if-then rules that need to be generated or trained offline and the membership functions of the fuzzy sets used in the fuzzy rules are stored in the database for retrieval. The

Table 1 Damage indicators for different damage locations

Indicator	Horizontal tail (HT)	Vertical tail (VT)	Wing
Aerodynamic coefficients	C_{m_α}, C_{m_q}	C_{n_β}, C_{n_r}, C_{Y_β}	C_{L_α}, C_{l_α}

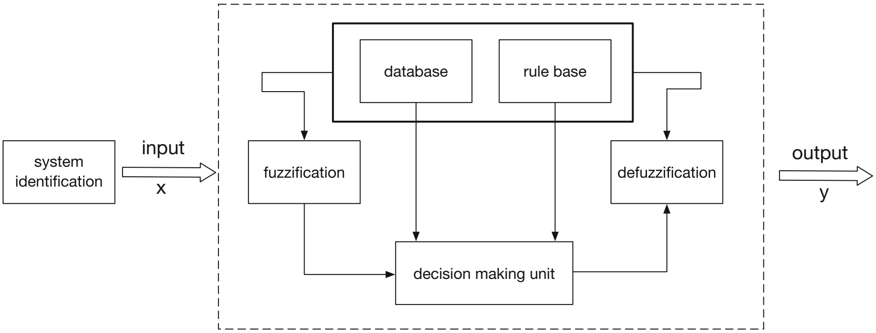


Fig. 1 Schematic representation of a fuzzy inference system

fuzzification process transforms the crisp inputs into degrees of match with linguistic variables using membership functions. A decision-making unit will then be used to performs the inference operations on the rules, i.e. to map fuzzy sets to fuzzy sets and determines the way they are combined. Finally, the defuzzifier again transforms the fuzzy values into crisp results [4, 7].

3.1 Input and Output

The inputs to the fuzzy logic system are the aerodynamic coefficients listed in Table 1, which are identified from measured forces and moments according to Eq. 3. The outputs are the location and severity of structural damage expressed in linguistic variables. Mathematically, FLS performs a mapping from crisp inputs x to outputs y [4, 7]:

$$y = F(x) \tag{4}$$

The linguistic variables describing damage parts include four aerodynamic surfaces of the aircraft, which are horizontal tail (HT), vertical tail (VT), left wing (LW), right wing (RW). Each variable is split into a set of terms covering its universe of damage severity like:

$$T(\text{left wing}) = \{\text{undamaged, slight damage, moderate damage, severe damage}\} \tag{5}$$

In this paper, we only consider single damage for simplicity, and each damage case is simulated in the well-developed DASMAT citation model in a Maltab/Simulink environment with measurement noises, based on which the aerodynamic coefficients are identified via recursive least squares. The damaged aircraft model is modeled as a linear change to the original model of total forces and moments from aerodynamic look-up table:

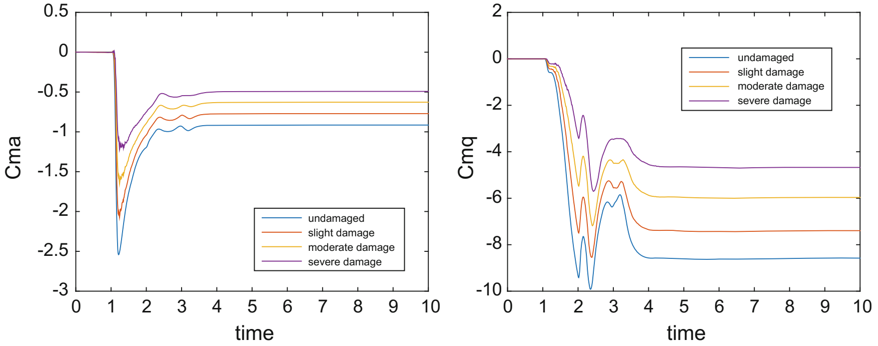
$$\begin{aligned} C_{X_{dmg}} &= (1 + \Delta C_X)C_{X_0} & C_{Y_{dmg}} &= (1 + \Delta C_Y)C_{Y_0} & C_{Z_{dmg}} &= (1 + \Delta C_Z)C_{Z_0} \\ C_{l_{dmg}} &= (1 + \Delta C_l)C_{l_0} + \Delta C_{l_\alpha}\alpha & C_{m_{dmg}} &= (1 + \Delta C_m)C_{m_0} & C_{n_{dmg}} &= (1 + \Delta C_n)C_{n_0} \end{aligned} \tag{6}$$

For each damage location and severity level, the changed values of aerodynamic forces and moments are list in Table 2. Due to limited sources of data, the figures in the table are assigned based on the information like changing magnitude and gradient abstracted from the experimental results done by others. So the table does not represent the aerodynamic changes of damage in a numerically precise way, but only for the proof of concept and the feasibility of algorithms adopted in this paper. Further improvements might be obtained with more experimental data and thorough analysis.

The identified aerodynamic coefficients under three damage cases are displayed in Figs. 2, 3 and 4. For wing damage only the results of left wing are shown because

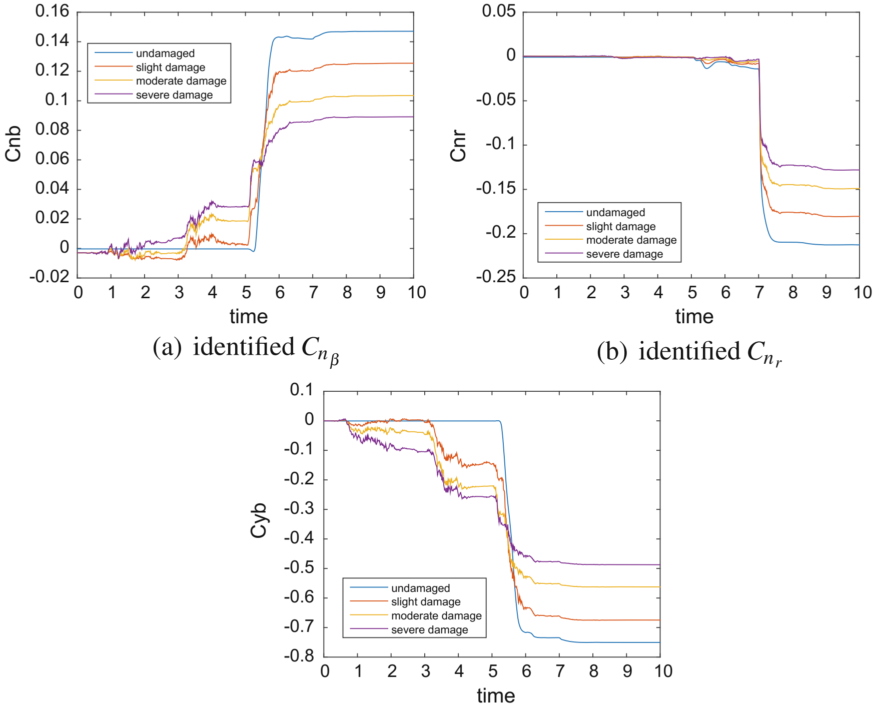
Table 2 Damage indicators for different damage locations

	HT		VT			
	ΔC_m		ΔC_n	ΔC_Y		
Undamaged	[−0.15, −0.1]		[−0.1, −0.05]	[−0.15, −0.1]		
Slight	[−0.4, −0.2]		[−0.35, −0.15]	[−0.35, −0.2]		
Moderate	[−0.55, −0.3]		[−0.5, −0.2]	[−0.6, −0.3]		
Severe	[−0.7, −0.5]		[−0.8, −0.55]	[−0.85, −0.6]		
	LW			RW		
	ΔC_L	ΔC_l	ΔC_{l_α}	ΔC_L	ΔC_l	ΔC_{l_α}
Undamaged	[−0.08, −0.02]	[−0.08, −0.02]	[−0.003, −0.001]	[−0.08, −0.02]	[0.02,0.08]	[0.001,0.003]
Slight	[−0.2, −0.1]	[−0.15, −0.08]	[−0.004, −0.002]	[−0.2, −0.1]	[0.08,0.15]	[0.002,0.004]
Moderate	[−0.35, −0.2]	[−0.3, −0.1]	[−0.005, −0.003]	[−0.35, −0.2]	[0.1,0.3]	[0.003,0.005]
Severe	[−0.6, −0.3]	[−0.5 − 0.4]	[−0.006, −0.004]	[−0.6, −0.3]	[0.4,0.5]	[0.004,0.006]



(a) identified C_{m_α} (b) identified C_{m_q}

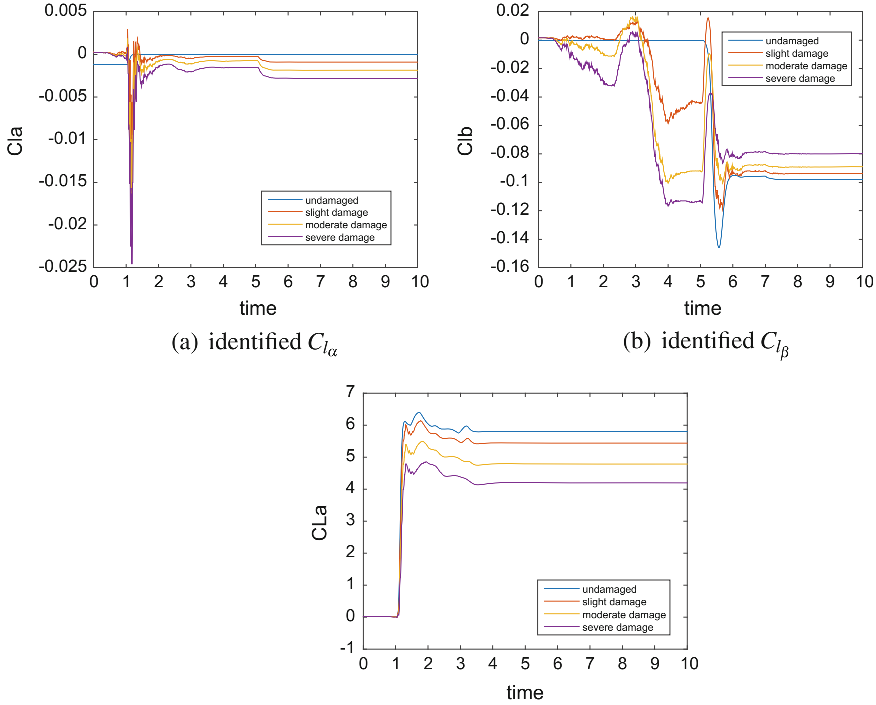
Fig. 2 The identified damage indicators for horizontal stabilizer tip loss



(a) identified C_{n_β} (b) identified C_{n_r}

(c) identified C_{Y_β}

Fig. 3 The identified damage indicators for vertical tail tip loss



(a) identified C_{l_α}

(b) identified C_{l_β}

(c) identified C_{L_α}

Fig. 4 The identified damage indicators for left wing tip loss

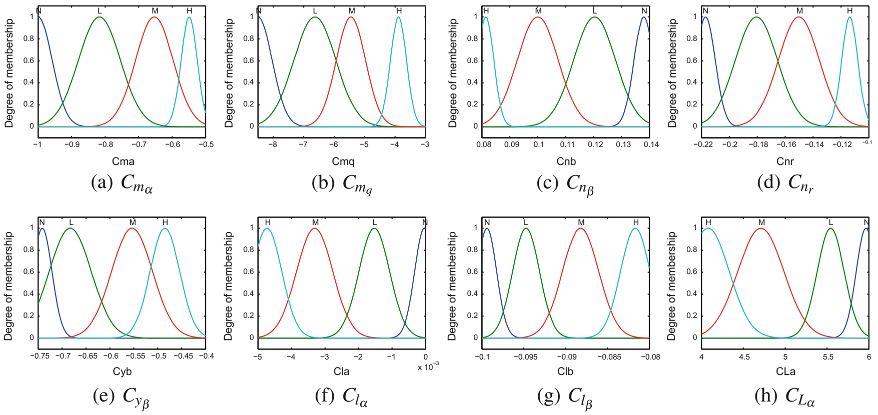
the figures of right wing has little difference with the left wing only with some signs reversed. The simulation time is 10 s, and all the coefficients have converged to stable values, which shows clearly the aerodynamic effects of different scales of structural damage.

3.2 Fuzzification

After obtaining the input data, the next step in FIS is fuzzification, where the inputs are transformed into degrees of membership of the linguistic variables calculated from the membership functions $\mu(x)$, which takes on the values between 0 and 1 [4, 7]. Fuzzy sets with Gaussian membership functions are used for the input variables, which are defined as:

$$\mu(x) = e^{-0.5((x-m)/\sigma)^2} \tag{7}$$

where m is the midpoint of the fuzzy set and σ is the standard deviation associated with each linguistic variable. In this application, the linguistic variables for numerical



(a) C_{m_α} (b) C_{m_q} (c) C_{n_β} (d) C_{n_r}

(e) C_{y_β} (f) C_{l_α} (g) C_{l_β} (h) $C_{L\alpha}$

Fig. 5 The input membership function of each damage indicator

inputs are defined below:

$$T(x) = \{\text{negligible (N), low (L), medium (M), high (H)}\} \tag{8}$$

The membership function of each damage indicator, i.e. aerodynamic coefficients are displayed in Fig. 5.

3.3 Rule Generation and Inference Engine

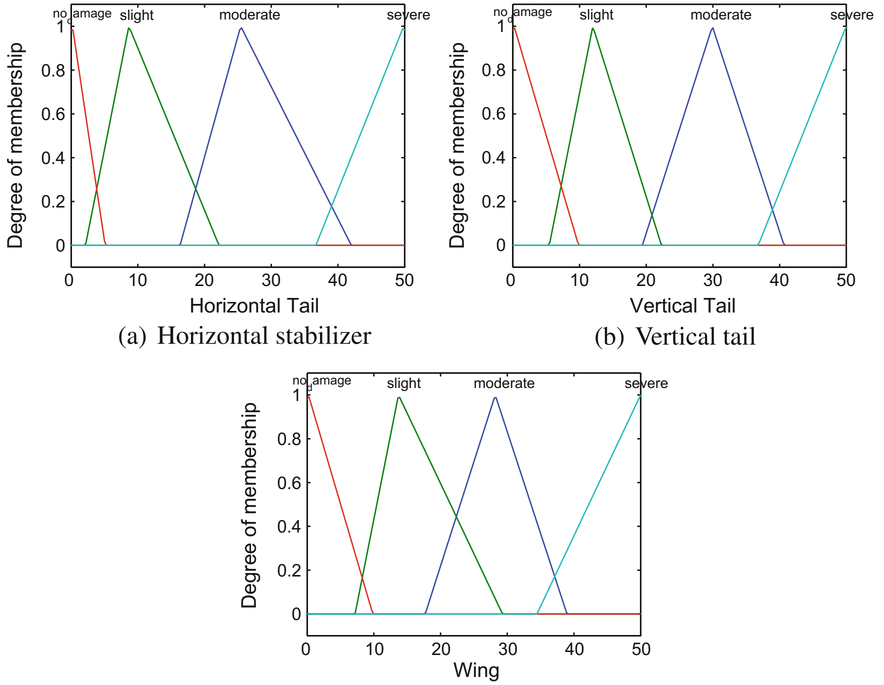
Rules for the fuzzy system can be expressed as [4, 7]:

$$\begin{aligned} R_i : \textbf{IF} \quad & x_1\ is\ F_1 \ \text{ AND } \ x_2\ is\ F_2 \ \text{ AND} \ldots x_m\ is\ F_m \\ \textbf{THEN}\ & y = K_i \end{aligned} \tag{9}$$

where $F_i \in V_i$ and $K_i \in W$ are fuzzy sets characterized by membership functions μ_{F_i} and μ_{K_i}. Each rule corresponds to one class, i.e., damage pattern in our application.

The rules can be either defined or trained through a supervised learning. Each rule can be evaluated by applying fuzzy operator and implication method. For the **IF** part, which is also called the antecedent, the fuzzy operator is applied to obtain one number by combining the input membership degree of each part of the antecedent. Various types of fuzzy operators could be taken. In this paper we pick the **AND** operation which multiplies all the membership degrees of the inputs for each rule:

$$\mu_{R_i}(x) = \mu_{F_1}(x_1) \cdot \mu_{F_2}(x_2) \cdot \cdots \cdot \mu_{F_m}(x_m) \tag{10}$$



(a) Horizontal stabilizer

(b) Vertical tail

(c) Wing

Fig. 6 The output membership function of each damage part of the aircraft

The resulting number is then applied to the output membership function $\mu_{K_i}(y)$, as is illustrated in Fig. 6 and implication process is implemented for each rule as:

$$\mu_{R_i}(x, y) = \mu_{R_i}(x) \cdot \mu_{K_i}(y) \tag{11}$$

Given the input and output membership function, the training and generation of fuzzy rules can be followed by the procedure explained below [4, 7]:

1. Each given damage case is simulated in the aircraft model according to Table 2 and the corresponding damage indicators are identified. The membership degrees of each damage indicator are calculated through the input membership functions
2. For each damage indicator having four degrees of membership based on the linguistic measures in Eq. 8, select the membership with the maximum degree.
3. One rule is obtained for each damage case by combining all the damage indicators with linguistic measures having maximum membership degrees.

Based on the above procedure, the trained rules from simulation data are displayed in Table 3.

Table 3 Rules of fuzzy inference system

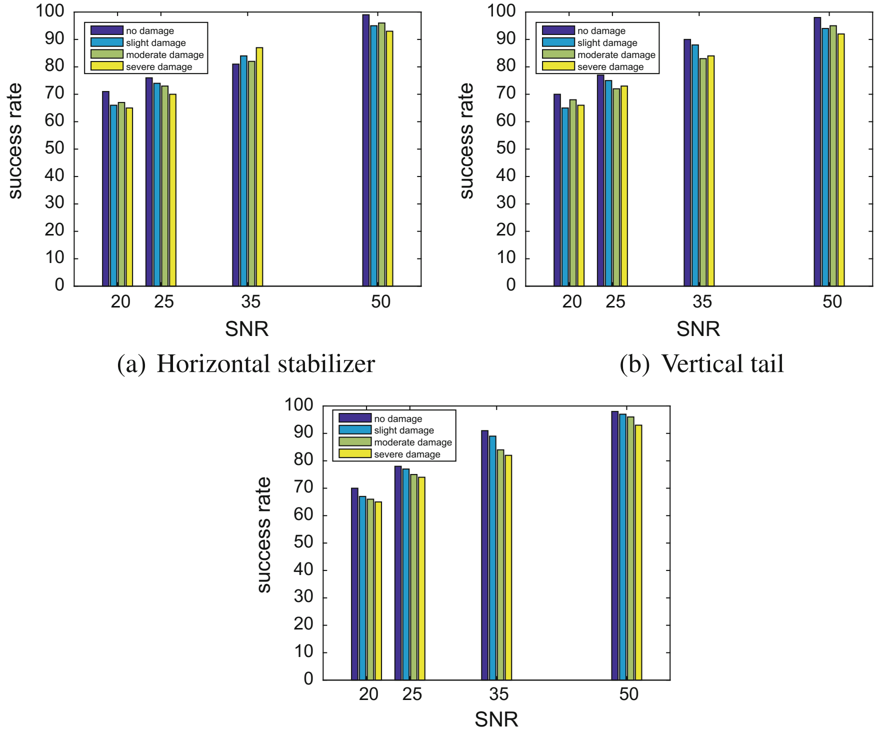
IF							THEN
C_{m_α}	C_{m_q}	C_{n_β}	C_{n_r}	C_{Y_β}	C_{L_α}	C_{l_α}	Damage case
N	N	N	N	N	N	N	Undamaged
L	N						Slight damage at HT
M	L						Moderate damage at HT
H	H						Severe damage at HT
		L	N	L			Slight damage at VT
		M	M	M			Moderate damage at VT
		H	M	H			Severe damage at VT
					L	L	Slight damage at wing
					M	M	Moderate damage at wing
					H	H	Severe damage at wing

3.4 Defuzzification

Once the fuzzy rule are applied to a given input, we have a set of degree of memberships for each damage pattern. Widely used defuzzification methods include maximum matching and centroid defuzzification. The latter one is often used when a crisp output value is required, while in pattern recognition we only need to know the class index [4, 7]. So in this paper, maximum matching is used to find the class which has the maximum matching degree of the rules. Finally the damage pattern with the highest degree of membership is selected containing the information of damage location as well as damage severity.

3.5 Damage Pattern Recognition Using Fuzzy Logic

To validate the methods proposed in this paper, validation data are generated from the DASMAT simulation model on which four levels of damage severity for different parts of the aircraft are applied. In order to test how well the proposed FIS deal with uncertainties, noises with four different signal-to-noise ratios (SNR) are added in to simulate the uncertainty present in experimental measurements and modeling process. For each case, one thousand noisy data points are generated. The performances is evaluated by success rate, which is the ratio of the damage cases that are correctly recognized by the FIS. Figure 7 shows the results of different noise levels for damage in left horizontal stabilizer, vertical tail and left wing respectively. It can be observed that the success rate increases when larger SNR, i.e., less noises are applied, and the success rates of severe damage are slightly lower then that of less severe damage.



(a) Horizontal stabilizer

(b) Vertical tail

(c) Wing

Fig. 7 The success rate of damage pattern recognition at different locations and noise levels

4 Conclusion

A new method using fuzzy logic for damage pattern recognition is proposed and implemented in this paper to compensate for the limited experimental condition and data of damaged aircraft. The aerodynamic coefficients sensitive to the integrity of the aerodynamic surfaces are picked as damage indicators. The selected coefficients are estimated, fuzzified and mapped to a damage case via a set of fuzzy rules. Simulation results show that by using FIS, the damage pattern recognition can deliver vague but useful information on the health condition of the aircraft. Future work can be done to enhance the accuracy of the method by using neural networks to train the membership functions and fuzzy rules so that the parameters of FIS can always be updated with new incoming measurement information.

References

1. Hwang I, Kim S, Kim Y, Seah CE (2010) A survey of fault detection, isolation, and reconfiguration methods. IEEE Trans Control Syst Technol 18(3):636–653
2. Bakhary N, Hao H, Deeks AJ (2007) Damage detection using artificial neural network with consideration of uncertainties. Eng Struct 29(11):2806–2815
3. Lopez I, Sarigul-Klijn N (2010) A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: challenges and opportunities. Prog Aerosp Sci 46(7):247–273
4. Chandrashekhar M, Ganguli R (2009) Damage assessment of structures with uncertainty by using mode-shape curvatures and fuzzy logic. J Sound Vib 326(3–5):939–957
5. Chandrashekhar M, Ganguli R (2009) Uncertainty handling in structural damage detection using fuzzy logic and probabilistic simulation. Mech Syst Signal Process 23(2):384–404
6. Fang X, Luo H, Tang J (2005) Structural damage detection using neural network with learning rate improvement. Comput Struct 83(25–26):2150–2161
7. Ganguli R (2001) A fuzzy logic system for ground based structural health monitoring of a helicopter rotor using modal data. J Intell Mater Syst Struct 12(6):397–407
8. Jang JSR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685
9. Taha MMR, Lucero J (2005) Damage identification for structural health monitoring using fuzzy pattern recognition. Eng Struct 27(12):1774–1783
10. Takagi H, Hayashi I (1991) NN-driven fuzzy reasoning. Int J Approx Reason 5(3):191–212
11. Hayes C (1968) Effects of simulated wing damage on the aerodynamic characteristics of a swept-wing airplane model
12. Frink N, Pirzadeh S, Atkins H (2010) CFD assessment of aerodynamic degradation of a subsonic transport due to airframe damage. In: 48th AIAA aerospace sciences meeting, Orlando
13. Shah G, Foster J, Cunningham K (2010) Simulation modeling for off-norminal conditions-where are we now? In: AIAA modeling and simulation technologies conference, Toronto
14. Shah G, Hill M (2012) Flight dynamics modeling and simulation of a damaged transport aircraft. In: AIAA modeling and simulation technologies
15. Shah G (2008) Aerodynamic effects and modeling of damage to transport aircraft. In: AIAA guidance, navigation and control conference and exhibit
16. Zhang, Y., de Visser, CC., Chu, QP (2016) Online physical model identification for database-driven safe fight envelope prediction of damaged aircraft. In: AIAA atmospheric flight mechanics conference, pp 1–11, January 2016
17. Tol HJ, Visser CCD, Sun LG, Kampen EV, Chu QP (2016) Multivariate spline-based adaptive control of high-performance aircraft with aerodynamic uncertainties. J Guid Control Dyn 39(4):781–800

Part III
Atmospheric Applications

Ground Effect Analysis for a Quadrotor Platform

Davide Del Cont Bernard, Mattia Giurato, Fabio Riccardi and Marco Lovera

1 Introduction

In the rotorcraft literature, ground effect is defined as the increase in thrust (at constant power) of a rotor operating near the ground. While ground effect has been studied extensively for conventional helicopters (see, e.g., [1–3]), the phenomenon has received limited attention as far as small-scale multirotor helicopters are concerned, even though there is significant evidence that the performance of quadrotors, both in terms of stability [4] and attitude/altitude control, degrades when operating close to the ground surface. In the literature on multirotors, this degradation is generically attributed to ground effect, even though this interaction is then frequently neglected at the modelling stage. However during takeoff and landing and, possibly, in near-ground operation, ground effect is not avoidable and must be handled if one is to achieve satisfactory flight performance in the case of autonomous operation.

In this paper the results obtained in an experimental campaign aimed at characterising ground effect for a quadrotor platform will be reported. In particular, the main concern is to assess the mean value of thrust as a function of altitude from ground, i.e., to carry out a classical characterisation of ground effect in the sense of the early literature from the '30s and '40s. More precisely, a number of results have been obtained, in different conditions: first with the platform constrained both in position and attitude, then with one attitude degree of freedom released and finally in free

D. Del Cont Bernard · M. Giurato · F. Riccardi · M. Lovera (✉)
Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Milan, Italy
e-mail: marco.lovera@polimi.it

D. Del Cont Bernard
e-mail: davide.delcont@mail.polimi.it

M. Giurato
e-mail: mattia.giurato@polimi.it

F. Riccardi
e-mail: fabio.riccardi@polimi.it

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_19

flight. In this paper, for space limitations, emphasis will be placed on the case of the constrained platform.

The paper is organised as follows: in Sect. 2 an overview of the state of the art on the problem of multirotor ground effect is provided, while Sect. 3 is devoted to the description of the adopted quadrotor platform and Sect. 4 to the description of the experimental setup. Subsequently, the results on the characterisation of ground effect for a single rotor are provided in Sect. 5, while the case of the complete platform is discussed in Sect. 6.

2 State of the Art

2.1 Classical Modelling for Ground Effect

As was mentioned in the Introduction, ground effect has been studied both numerically and experimentally for almost a century, see [1] for an overview or [2, 3]. In particular, experiments usually focus on the following conditions:

- Variation of power required P at constant thrust with respect to the Out-of-Ground-Effect (OGE) value P_∞: $\left[\frac{P}{P_\infty}\right]_T = k_g$
- Variation of thrust T at constant power with respect to the OGE value T_∞: $\left[\frac{T}{T_\infty}\right]_P = f_g$,

where coefficients k_g and f_g are related by

$$k_g = \frac{P}{P_\infty} = \frac{v_i}{v_{i\infty}} = \left(\frac{T}{T_\infty}\right)^{-3/2} = -f_g^{-3/2}, \tag{1}$$

where, in turn, v_i and $v_{i\infty}$ denote, respectively, the In-Ground-Effect (IGE) and OGE induced velocity. Furthermore, experimentally a dependence on:

- Distance from ground, normalised to rotor radius h/R
- Disk loading C_T/σ
- Blade geometry (e.g., taper and twist)

is observed. Some classical expressions for the above mentioned coefficients are provided hereafter:

$$f_g = \left[1 - \frac{1}{(4h/R)^2}\right]^{-1} \qquad \text{Cheeseman and Bennet} \tag{2}$$

$$f_g = \left[0.9926 + \frac{0.03794}{(h/2R)^2}\right]^{2/3} \qquad \text{Hayden} \tag{3}$$

2.2 Ground Effect in Quadrotors

While there is increasing attention to ground effect in quadrotors, only a small number of references is available at the moment. In [5] it is suggested to use the radius of a rotor equivalent to the sum of the areas of the individual rotor as a geometric parameter to apply classical results for single rotor helicopters. In [6] a model for ground effect based on the expression $\frac{T_{input}}{T_{output}} = 1 - \rho \left(\frac{R}{4h}\right)^2$ is used, where ρ is a parameter estimated from experimental data. For the identification of ρ, hovering tests at different altitudes have been carried out and it was noted that the effect was measurable for distances up to $h/R = 4$. In [7] it is claimed that ground effect is measurable up to almost $h/R = 6$. It is also suggested to use the classical formulas for conventional helicopters by replacing the main rotor radius with the distance from the outer tip of one rotor to the vehicle's center. In [8] a commercial Draganflyer X8 multirotor, having 4 pairs of coaxial, counter-rotating rotors, is considered. Unlike other studies, the requirement of carrying out the experiments at constant power so as to obtain results comparable with the Cheeseman formula, has been enforced. However, a characterisation in terms of the rotor rpm is used. It is concluded that ground effect can be seen for distances up to $h/R = 6$. Reference [4] reports frequency-domain analyses of roll dynamics as a function of altitude and a slow-down of roll dynamics as the distance from ground decreases is reported.

Finally, a number of references dealing with the problem of the experimental characterisation of propellers for small-scale multirotors is available, see for example [9–11].

3 Considered Quadrotor Platform

The quadrotor helicopter considered in the context of these experiments is shown in Fig. 1. The quadrotor is a custom model with a distance of 550 mm between opposite rotor axes and an overall take-off weight of about 1.5 kg. The relevant parameters are reported[1] in Tables 1 and 2.

The Flight Control Unit (FCU) uses as electronic boards the R2P (Rapid Robot Prototyping) modules (see [12]). R2P is an open source HW/SW framework providing components for the rapid development of robotic applications. The R2P framework use and integration is obtained through a modular hardware and an embedded real-time publish/subscribe middleware which allows distributed control loops to be set up in a flexible way. The R2P IMU-Module has a built-in 3-axis gyroscope, 3-axis accelerometer and a 3-axis magnetometer the outputs of which have been logged with a frequency of 100 Hz. The on-board attitude control system has been designed using the approach based on structured H_∞ synthesis described in [13], starting from grey-box identified models (see [14]).

[1] During the experiments on the testbed the power was supplied by a 12 V laboratory power supply.

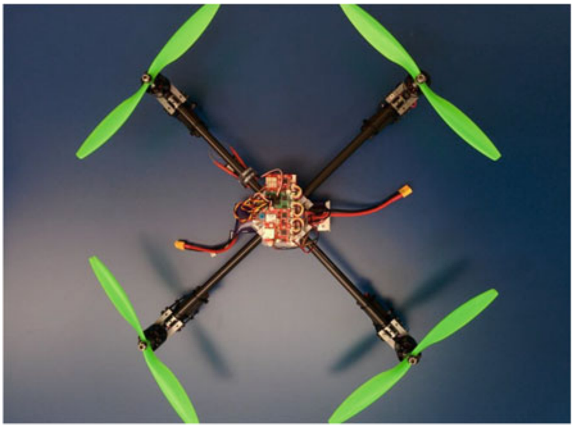


Fig. 1 Quadrotor platform used in this study

Table 1 Main parameters of the quadrotor helicopter

Variables		Value
Frame configuration		X
Frame model		HobbyKing talon V2.0
Arm length	b	0.275 m
Take-off weight	m	1.51 kg
Roll inertia	I_{xx}	0.035 kg × m^2
Pitch inertia	I_{yy}	0.035 kg × m^2
Motors		HP2814 - RCTimer
KV		710 rpm/V
ESC		RCTimer NFS 30 A
Battery		LiPo Turnigy nano-tech 4000 mAh 3S

Table 2 Main parameters of the propellers

Variables		Value
Radius	R	0.1524 m
Average chord length	$\bar{c}$	0.02 m
Solidity	σ	0.083
Profile		Unknown
Flapping frequency	f_{1f}	9 Hz

4 Experimental Setup

The quadrotor platform has been rigidly attached to the test bed (see Fig. 2) in order to keep a constant distance from the ground, and adjust it during the different experiments.

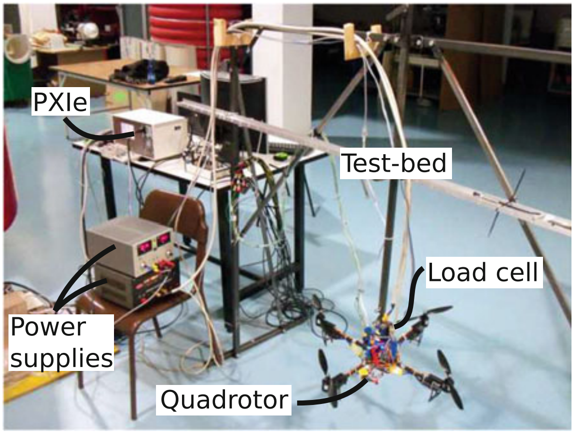


Fig. 2 Complete view of the experimental setup

The sensors used during these experiments were:

- Load cell: a six degrees of freedom load cell has been used in order to measure the vertical force and the longitudinal torques.
- Tachometer: a custom tachometer has been used to measure the rotational speed of each motor/propeller.
- Current sensor: in order to measure the power consumption, each motor has been equipped with a current sensor.

All the data were logged by means of a LabVIEW visual interface and a PXIe-1078 system from National Instruments using two different data acquisition boards: a PXI-6284 for the load cell and current sensors and a PXI-6123 for the tachometer. Each motor was controlled directly by the quadrotor FCU.

Finally, note that during the tests practical issues with the available setup (i.e., limitations in the current supply and saturation of the load cell) prevented from reaching 100% throttle, so that results for throttle up to 80% for the isolated rotor and up to 70% for the complete quadrotor are presented.

5 Results: Isolated Rotor

Before getting to the quadrotor case with four interacting rotors and wakes, the behaviour of one single isolated rotor is studied. This analysis also makes possible a comparison with the above-mentioned classical results from the rotorcraft literature.

For the purpose of this analysis, a rotor was removed from the quadrotor and mounted alone directly on a load cell and the test stand (see Fig. 3).The measurement setup is the one described in the previous section concerning the entire quadrotor case. For each altitude taken into account, sequences of throttle step commands were

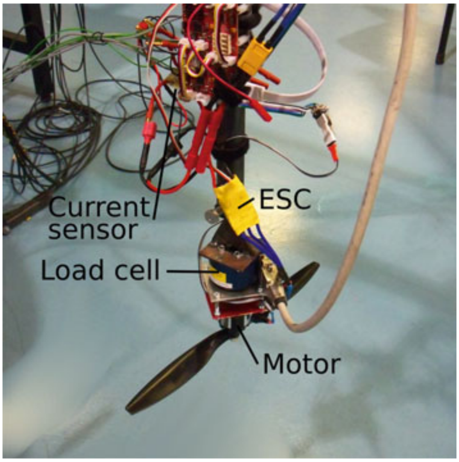


Fig. 3 Experimental setup for single-rotor tests

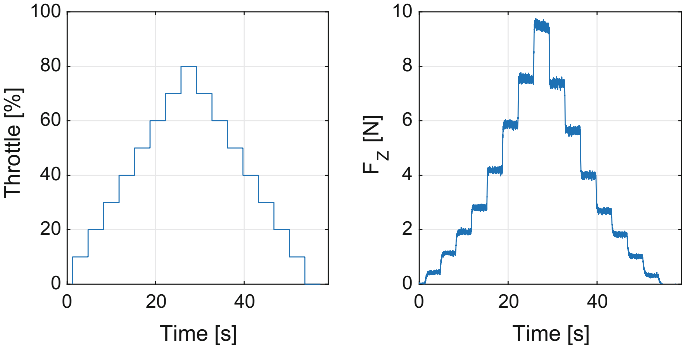


Fig. 4 Imposed throttle sequence (on the *left*) and thrust response (on the *right*)

sent to the Electronic Speed Controller (ESC), and the response in terms of angular velocity, loads, current and voltage on the ESC were recorded, stored, and later processed. From each of the throttle steps (see Fig. 4), averages were taken of all interesting variables, in order to obtain a complete and systematic characterization of the behaviour at a given altitude (Fig. 5).

Some results concerning the operation of the rotor outside the range of altitude where the ground effect can be observed, i.e., at $h/R \simeq 4.7$, are first presented.

It is interesting to examine the trend of the thrust coefficient C_T, plotted in Fig. 6. C_T is routinely approximated as a constant, but in this case a significant variability and hysteresis are apparent. These facts seem to be related to the peculiar working

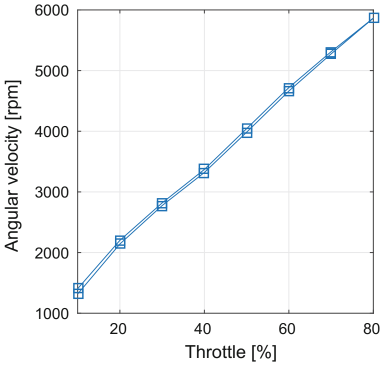


Fig. 5 Throttle versus rpm (out of ground effect)

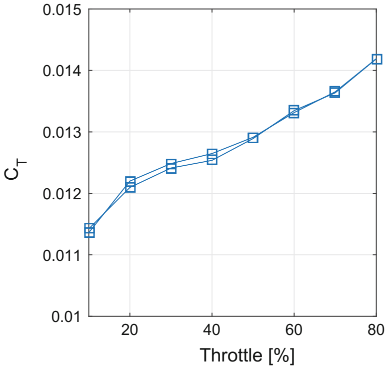


Fig. 6 Thrust coefficient C_T versus throttle

condition (low Reynolds number and significant blade flexibility), resulting in a change of geometry according to the loading conditions.[2]

From the analyses on a single rotor, the energy efficiency of the motor+ESC assembly can be easily evaluated. In fact, the mechanical power can be calculated from the readings of the load cell and the tachometer, whereas the electrical power can be evaluated from the current through the ESC and supply voltage measurements. One can see from Fig. 7 that the relationship between mechanical and electrical power is nearly linear. Furthermore, a global actuator efficiency can be defined as the ratio between the output mechanical power and the given electrical power: $E = P_{mec}/P_{ele}$ (see Fig. 8). This efficiency weigths both the motor efficiency and the ESC efficiency.

The resulting data, obtained applying the previously described workflow, can be plotted in graphs where any of measures between powers, current, voltage, forces and moments can be expressed depending on altitude and throttle. For example, in Fig. 9, one can see how the thrust (one of the most important parameters) changes with altitude and throttle.

[2] Some tests have been conducted on a much stiffer carbon fiber blade and C_T seemed to be definitely more constant.

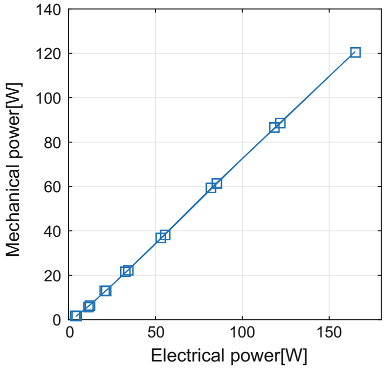


Fig. 7 Actuators: generated mechanical power versus electrical power input

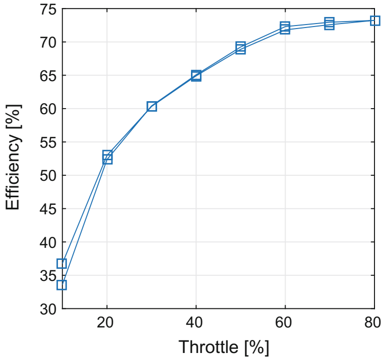


Fig. 8 Actuators: global efficiency

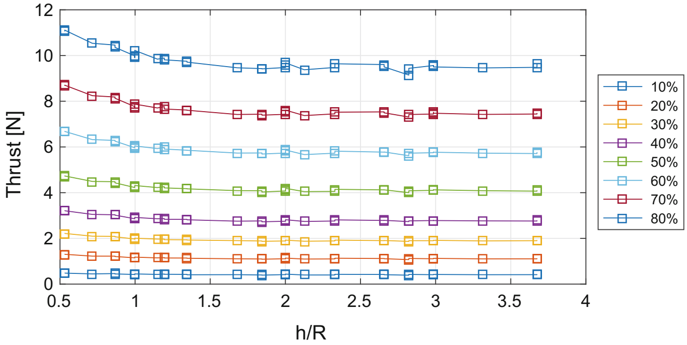


Fig. 9 Single rotor: thrust versus height from ground for different throttle levels

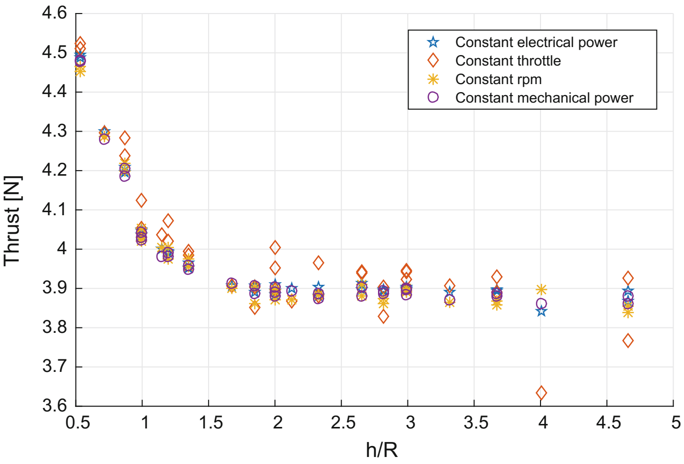


Fig. 10 Comparison between different ways of presenting the results (isolated rotor)

Table 3 Variables corresponding to a hovering condition

Variable	Value
Thrust (OGE)	3.9 N
Electrical Power	51 W
Mechanical Power	35 W
Throttle	48 %
Angular velocity	3890 rpm

In order to compare the performance of the system at different altitudes, it is useful to keep constant some of the parameters. Traditionally, ground effect has been studied at constant thrust or constant power [1, 3]. In this case, along with power and thrust, other variables can be held constant, for example throttle or the input electrical power. In Fig. 10, one can compare these different ways of seeing the thrust change with altitude, with different variables being fixed.

Figure 10 was obtained linearly interpolating a large amount of data. For each altitude, power and thrust or thrust and throttle or throttle and rpm data were interpolated; the query points for the interpolation were chosen to be same required for hover, and can be seen in Table 3.

In order to draw the plots of thrust versus altitude, like the one at constant mechanical power of Fig. 12, for each altitude, the curves of mechanical power and thrust are available and the thrust corresponding to a given mechanical power (for example 35 W) can be found by interpolating the data (see Fig. 11 for a graphical interpretation). Repeating the procedure for every considered altitude allows to construct all the curves in Figs. 10 and 12.

In the following, for the sake of simplicity all the results are presented at constant electrical power, since the mechanical power cannot be evaluated in the quadrotor configuration due to moments cancellation of opposing motors. From Fig. 10 it is

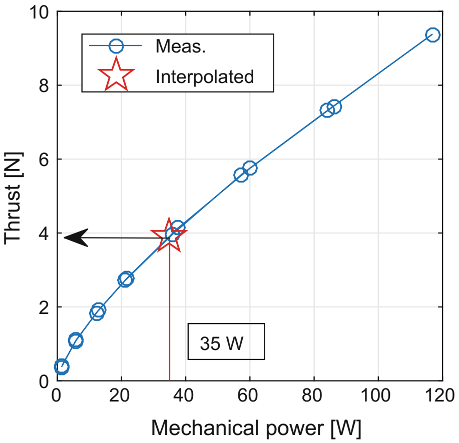


Fig. 11 Example of data interpolation in order to trace the curves at constant power (altitude: $h/R = 2.13$)

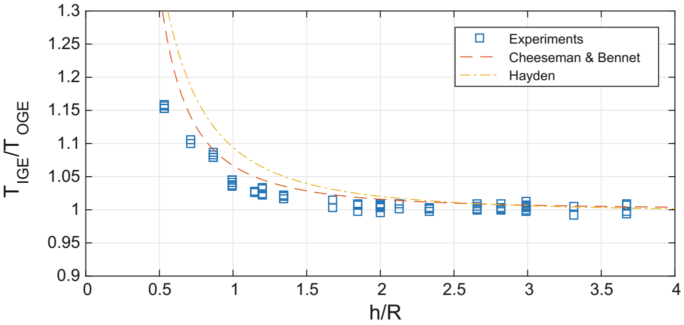


Fig. 12 Single rotor: comparison between experimental data and classical curves, at hovering power

obvious that keeping constant the mechanical power is nearly equivalent to keeping constant the electrical power or the throttle. All throttle-related results however look more noisy than the power related ones.

Now with all these data available, a comparison with the classical models from the rotorcraft literature becomes possible. More precisely, in Fig. 12 one can see a plot of the thrust at constant power, compared with the estimates of Cheeseman and Bennet and Hayden models (see Sect. 2 for details). It should not come as a surprise that there is not a perfect agreement between models and experimental data; in fact the models are based on hypothesis of inviscid potential flow whereas the experimental data are concerning a rotor working at very low Reynolds numbers. The main conclusion one can draw is that for altitudes greater than $h/R < 2.5$, ground effect is nearly negligible (less than 1%).

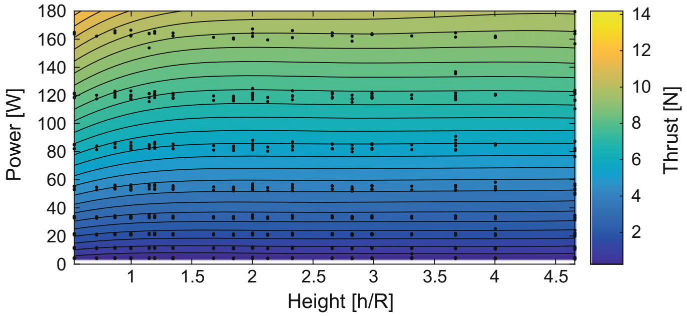


Fig. 13 Single rotor: polynomial best fit of all experimental results

Finally, in Fig. 13 all the experimental results are reported in a comprehensive contour plot showing the thrust as a function of height and electrical power. It has been obtained through a polynomial best fit over all the gathered data (whose position is shown with black dots). It is apparent that, at a given power, there is a significant increase in thrust when a rotor operates in ground proximity.

6 Results: Complete Quadrotor

In this section the experimental results obtained considering the complete quadrotor will be presented and discussed. The aim is to characterize the ground effect on global rotor performance parameters and put in evidence the differences respect to the previous analysis conducted on the isolated rotor.

As described in Sect. 3 the vehicle is assembled using COTS low-cost components. As a consequence it is not guaranteed that the four propulsion systems composed by ESC, brushless motor and rotor are close to each other in terms of performance; in fact the noticed discrepancies are not negligible. In particular the motors appear to be the components with the wider variability.[3] In order to evaluate these differences, the same experiments carried out on the isolated rotor were repeated on the quadrotor but activating only one rotor at a time (on a limited range of throttle, in order to not overload the load cell). The results for the four rotors ($R1 \rightarrow R4$) are collected in Fig. 14.

Taking into account the above issue and the necessity to define some global parameters able to characterise the quadrotor working state for different altitudes from ground, the following assumptions will be adopted hereafter:

[3] For example, motor number 3, also tested with different ESC and rotor, shows at the same working conditions an angular velocity about 20% larger with respect to the other units, as if the actual KV parameter were different from the value declared by the manufacturer. This issue is visible in Fig. 14.

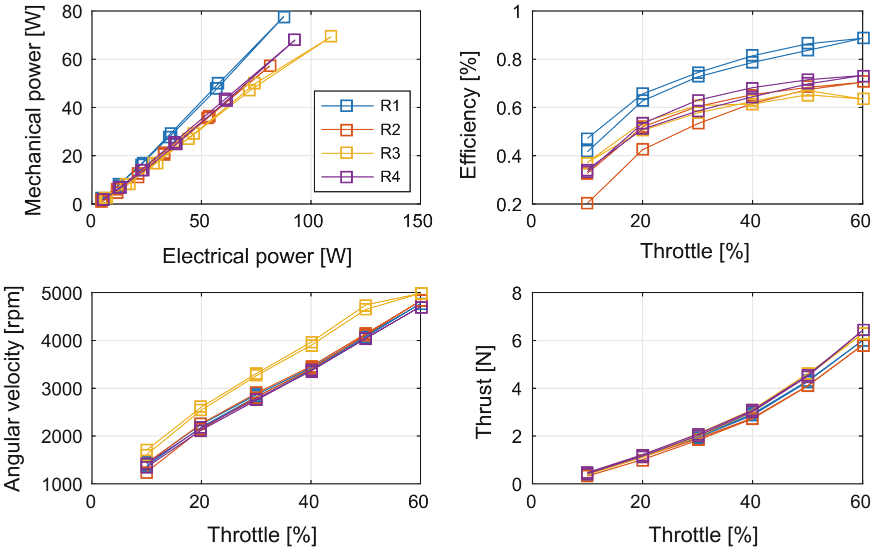


Fig. 14 Complete quadrotor: comparison between each ESC and motor+rotor system performance

- *Throttle*: the input command to the ESC is identical for each propulsion unit;
- *Power*: it is considered as the sum of the power required by each rotor, $P = \sum_{i=1}^{4} P_i$;
- *Thrust*: it is considered as the sum of each rotor thrust, $T_Z = \sum_{i=1}^{4} T_{Zi}$;
- *Angular velocity*: it is considered as the average of the angular velocity of each rotor, $\Omega = \frac{1}{4}\sum_{i=1}^{4} \Omega_i$.

Using the above global quantities instead of the single ones, the obtained results are clearly repeatable as shown in Fig. 15, where the results of more than 100 experiments at different altitudes are reported.

The same test procedure previously described for the isolated rotor characterization was adopted for the complete quadrotor case: at each altitude from ground different values of throttle (from 10 to 70%, step 10%) were tested and all the available measures were logged. For example, the rotor thrust as a function of altitude and throttle is shown in Fig. 16.

As reported in Fig. 17, the complete quadrotor case is quite distinct from the isolated rotor one. The effect of the ground proximity on the thrust is extended up to almost 4 rotor radii of height, doubling the limit of about $h/R = 2$ found for the isolated rotor test.

It can be observed that the experimental data qualitatively follows the trend of the classical formulations in the isolated rotor case while for the complete quadrotor this is no longer true, hence it seems that the classical formulas valid for full-scale helicopters are not able to model correctly the ground effect for small multi-rotor vehicles.

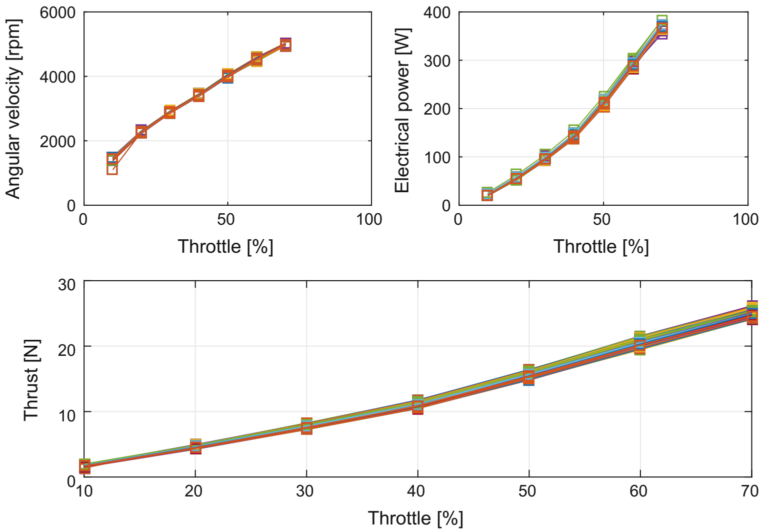


Fig. 15 Complete quadrotor: superimposed data of more than 100 experiments at different heights

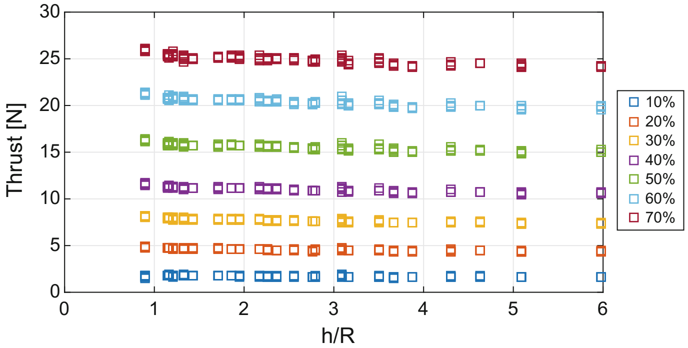


Fig. 16 Complete quadrotor: total rotor thrust as a function of non-dimensional height from ground and throttle

An effect that can potentially explain the different behavior in the complete quadrotor case respect to the isolated rotor one is the download due to the rotors inflow that impinges on the airframe elements. An interesting and exhaustive work about the topic, but considering a quad tilt-rotor (QTR) architecture was presented in [15] concerning an experimental campaign and in [16] about a CFD analysis: the mutual interactions between rotors-fuselage/wing-ground were investigated, characterizing the transition from the penalizing download affecting the rotorcraft OGE (14% of total thrust) to the beneficial upload (5% of total thrust) entering in ground effect.

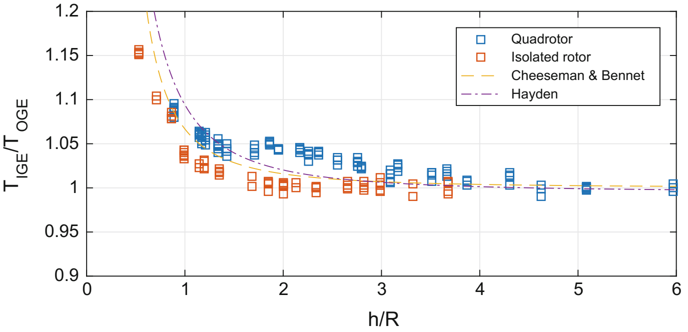


Fig. 17 Non-dimensional rotor thrust at constant power as a function of non-dimensional height from ground: comparison between isolated rotor, complete quadrotor experimental data and literature classical formulations

The presence of the quadrotor airframe surely influences the obtained results and the amount may be proportional to the ratio of the impinged surface under the rotor and the rotor area. For the QTR this ratio is $A_{airframe}/A_{rotors} \simeq 0.37$ (a large value due to the wings), while for the considered quadrotor it results $A_{airframe}/A_{rotors} \simeq 0.13$, more similar to the value for a conventional helicopter $A_{airframe}/A_{rotors} \simeq 0.08$ (see [17]).

The differences between single rotor and complete quadrotor curves in Fig. 17 become evident at about $h/R \leq 3$, the height at which the download should start reducing (see [18, Fig. 4.1]) and the upload effect due to ground starts rising. The discrepancy between the two curves reaches about $5\% T_{OGE}$, that is a value comparable with the download variation in the range $1 \leq h/R \leq 3$ for a conventional helicopter (see [17, Fig. 20]).

Another effect that could be responsible of these discrepancy is the aerodynamic interaction among the rotors, that could be present even in spite of the expected strong wake contraction characterizing very small and hence low Reynolds number rotors such as the considered ones, up to 4 : 1 wake area contraction with respect to rotor area, instead of classical 2 : 1 value for full-scale helicopters, see [3]. To the best of the Authors' knowledge a characterization of the aerodynamics interaction between rotors of small multi-rotor aerial vehicles is not presently available in the literature.

As previously done for the isolated rotor case in Fig. 18 a contour plot showing the thrust behaviour as a function of altitude, throttle and power is reported. The plot has been obtained through a polynomial interpolation over all the gathered experimental data.

In conclusion an interesting validation of the obtained results on the complete quadrotor can be performed referring to one of the few works available about quadrotors in ground effect, [19]: the vehicle was smaller with respect to the considered one (rotor radius equal to 4 cm) and the data was gathered directly in flight, simply measuring the rotors angular velocity reduction to maintain the hover when approaching the ground (the results taken from [19] are reported in Fig. 19). The same plot was

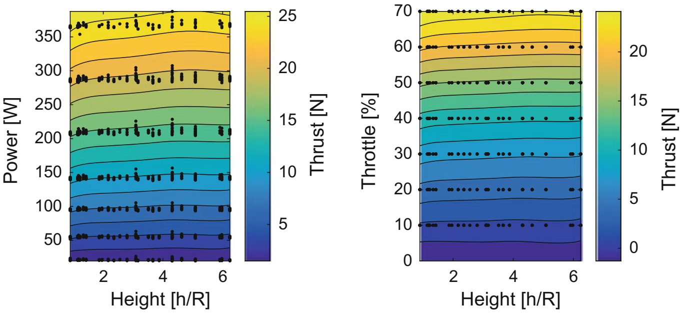


Fig. 18 Complete quadrotor: thrust as a function of height, throttle and power, obtained through polynomial interpolation over all the gathered experimental data

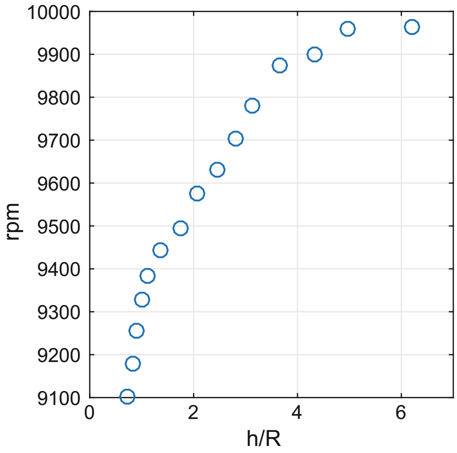


Fig. 19 Average rotors angular velocity as a function of non-dimensional height at constant thrust, data from [19]

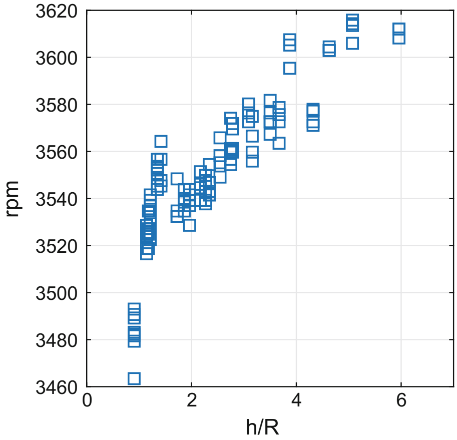
obtained from the conducted experiments data in Fig. 20, representing the average rotors angular velocity as a function of height at constant thrust.

Comparing Fig. 19 with Fig. 20 it is evident that the trend is qualitatively the same, in particular:

- for $h/R \geq 6$ the angular velocity is almost constant varying the altitude from ground;
- for $4 \leq h/R \leq 5$ a slope change in the dependence of angular velocity on distance from ground is visible;
- for $1 \leq h/R \leq 4$ an almost linear variation of the angular velocity with the height from ground is evident;
- for $h/R \leq 1$ a steep angular velocity reduction approaching the ground occurs.

An analogous behaviour characterized by distinct segments can be recognized also in Figs. 17 and 18.

Fig. 20 Average rotors angular velocity as a function of non-dimensional height at constant thrust, conducted test data



7 Concluding Remarks

In this paper the results of an extensive experimental campaign aimed at characterising ground effect for a small-scale quadrotor platform have been presented and discussed. In particular, both an isolated rotor and the complete platform have been studied. Experimental data in the isolated rotor case qualitatively follow the trend of the classical formulations of ground effect from the rotorcraft literature. For the complete quadrotor, on the other hand, this is no longer true: hence it seems that the classical formulas valid for full-scale helicopters are not able to model correctly the ground effect for small multi-rotor vehicles. In particular the effect of the ground on the thrust is extended up to almost 4 rotor radii of height for the quadrotor case, doubling the limit of about $h/R = 2$ found for the isolated rotor tests. Two possible phenomena that should explain the observed discrepancy between complete quadrotor and isolated rotor results have been discussed: the airframe download strong variation in proximity of the ground and the interferences between the four rotor wakes. Further experimental campaigns should be carried out extending the present work in order to verify and characterize both the hypothesized effects.

References

1. Johnson W (2013) Rotorcraft aeromechanics. Cambridge University Press, Cambridge
2. Padfield GD (2008) Helicopter flight dynamics. John Wiley & Sons, Newyork
3. Leishman GJ (2006) Principles of helicopter aerodynamics. Cambridge University Press, Cambridge
4. Aich S, Ahuja C, Gupta T, Arulmozhivarman P (2014) Analysis of ground effect on multirotors. In: 2014 International conference on electronics, communication and computational engineering (ICECCE), IEEE, New York pp 236–241
5. Bangura M, Mahony R (2012) Nonlinear dynamic modeling for high performance control of a quadrotor. In: Australasian conference on robotics and automation, pp 1–10,

6. Danjun L, Yan Z, Zongying S, Geng L (2015) Autonomous landing of quadrotor based on ground effect modelling. In 2015 34th Chinese control conference (CCC), IEEE, Newyork, pp 5647–5652
7. Davis E, Spollard J, Pounds P (2015) Passive height stability and trajectory repeatability of a quadrotor maneuvering in ground effect with regulated voltage bus. In: Australasian conference on robotics and automation (ACRA 2015)
8. Sharf I, Nahon M, Harmat A, Khan W, Michini M, Speal N, Trentini M, Tsadok T, Wang T (2014) Ground effect experiments and model validation with Draganflyer X8 rotorcraft. In: 2014 International conference on unmanned aircraft systems (ICUAS), IEEE, Newyork, pp 1158–1166
9. Saedan M, Puangmali P (2015) Characterization of motor and propeller sets for a small radio controlled aircraft. In: 2015 10th Asian Control Conference (ASCC), IEEE, Newyork pp 1–6
10. Yoon M (2015) On driving signal of electronic speed controller for small multi-rotor helicopter. Int J Eng Res Tech 4(11):456–459
11. Yoon M (2015) Experimental identification of thrust dynamics for a multirotor helicopter. Int J Eng Res Tech 4(11):206–209
12. Bonarini A, Matteucci M, Migliavacca M, Rizzi D (2014) R2P: An open source hardware and software modular approach to robot prototyping. Robot Auton Syst 62(7):1073–1084
13. Riccardi F, Lovera M (2014) Robust attitude control for a variable-pitch quadrotor. In IEEE multi-conference on systems and control. IEEE, Antibes, France
14. Panizza P, Riccardi F, Lovera M (2014) Identification of the attitude dynamics for a variable-pitch quadrotor. In 40th European rotorcraft forum. Southampton, UK
15. Radhakrishnan A, Schmitz FH (2003) An experimental investigation of a quad tilt rotor in ground effect. In: AIAA paper, vol. 3517
16. Gupta V, Baeder JD (2002) Quad tiltrotor aerodynamics in ground effect. In: Proceedings of the 58th annual Forum of the American Helicopter Society, pp. 48–61
17. Trept T (1984) A 0.15-scale study of configuration effects on the aerodynamic interaction between main rotor and fuselage. NASA-CR-166577
18. Radhakrishnan AM (2006) An experimental investigation of ground effect on a quad tilt rotor in hover and low speed forward flight. PhD thesis, University of Maryland, MD
19. Powers C, Mellinger D, Kushleyev A, Kothmann B, Kumar V (2013) Influence of aerodynamics and proximity effects in quadrotor flight. In: Experimental robotics, Springer, Berlin, pp 289–302

Nonlinear Model Predictive Flight Path Control for an Unmanned Powered Paraglider

Fabian Binz, Philipp Hartmann and Dieter Moormann

1 Introduction

Parafoil and payload systems offer high payload capacities, robustness and ease of deployment. Thus, they are used in both military and civil applications like remote sensing, aerial survey or airdrop supply systems. Besides their versatility, powered paragliders (PPGs) are interesting objects of investigation because of their unique flight dynamics. The center of mass being suspended below the parafoil leads to a high roll and pitch stability. At the same time, the high wind-to-airspeed ratio makes the aircraft susceptible to wind and gusts.

A large body of research on parafoil and payload systems focuses on modeling their complex, nonlinear dynamics. References [2, 4, 13] provide multiple detailed, nonlinear system descriptions with 8 or 9 degrees of freedom (DOF). These models are derived on the basis of a two-body system description with a joint representing the flexible connection between the payload and the parafoil. Reference [13] then continues to derive simpler models with only 6 DOF, which are more suitable for controller design. Assuming that the lateral and longitudinal motion are decoupled, model complexity can be further reduced to three DOF [5]. The parameters used in these models are either known or obtained using system identification techniques [8].

Concerning the flight guidance of parafoil and payload systems, several approaches can be found in the literature. At the highest level they share a common structure, namely, a separation between heading control and flight path guidance. The heading control is achieved using (nonlinear) model predictive control (MPC) [10, 11] or classic PID control [7].

To track the reference flight path, several control schemes have been proposed. Most commonly, the distance to a point on the reference flight path is mapped to a commanded heading, see [11]. More recently, the increased computational power of modern avionics allowed the use of MPC directly for flight guidance. In this regard,

F. Binz (✉) · P. Hartmann · D. Moormann
Institute of Flight System Dynamics, RWTH Aachen University,
Aachen, Germany
e-mail: binz@fsd.rwth-aachen.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_20

Ref. [12] presents experiment results of a parallel implementation of an MPC on a graphics processor. Here, the parallel processing capabilities are used to find a flight path which is robust against uncertainties introduced by wind gusts. Similarly, [6] presents a parallel implementation of nonlinear model predictive control (NMPC) on a FPGA-based flight computer enabling sample rates of 10 Hz.

Against this background, the control scheme for the horizontal flight path guidance presented here focuses on simplicity and ease of implementation. Especially in outer flight guidance control loops the inherent properties of PPGs can easily lead to large control deviations. One reason for this is the fact that classic feedback controllers typically only *react* to changes of the reference input. The controller proposed here is a predictive controller and thus can *act* according to predicted, future changes of the reference input. It can be argued that this property makes predictive controllers especially suited for flight guidance problems, since the reference flight path is typically known ahead of time. While the results gathered in this paper were obtained with a PPG, they are applicable to other parafoil and payload systems.

The paper is structured as follows: First, Sect. 2 gives an overview of the characteristics of PPGs along with a brief introduction into NMPC. Based on this, Sect. 3 presents the conceptual approach of the control scheme, then Sect. 4 discusses several aspects of its realization. The results of evaluating the controller in simulations and flight tests are summarized in Sect. 5.

2 Fundamentals

This section introduces basics concerning the flight dynamics of PPGs and will highlight some challanges these characteristics pose for accurate flight path guidance. Since the control scheme used here uses a black-box model of the plant, no analytical description of the flight dynamics of a PPG will be given. For a discussion of different modelling techniques the reader may refer to [8, 13, 14].

The discussion of model predictive control will concentrate on its conceptual ideas and keep formulas to a minimum. Comprehensive material on (nonlinear) model predictive control may be found in [3, 9].

2.1 Powered Paragliders

Powered paragliders belong to the class of parafoil and payload systems, which are characterized by the payload being suspended with lines below the parafoil, see Fig. 1. The only means of steering is the deflection of the trailing edge of the parafoil. The deflection is accomplished using two lines, each connected to one side of the trailing edge, often called brakes. Pulling one line alters the parafoils shape on the correspondig side. This asymmetric brake deflection creates a yawing moment, which in turn leads to a yawing rate. Pulling both brakes symmetrically primarily leads to an

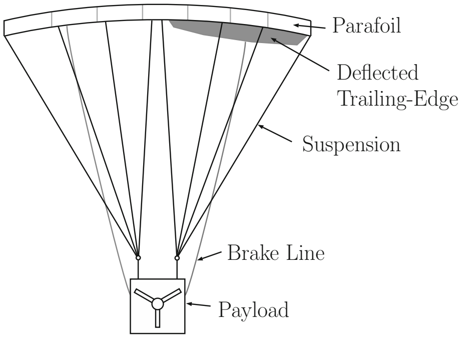


Fig. 1 Conceptual overview of a powered paraglider

increase in drag and can be used to control the airspeed and thus the glide path of the parafoil. Additionally, *powered* paragliders are equipped with a propulsion system, which is located in the same frame as the actual payload. Due to the suspension below the parafoil, an increase in thrust primarily increases the incidence angle of the parafoil and thus the vertical velocity. In comparison, the effect on the horizontal airspeed is small.

Due to their construction, PPGs typically exhibit high stability in both yaw and pitch. The long suspension lines allow relative motion between the parafoil and the payload. Because in the PPG considered here, all sensors are located in the payload, this relative motion has to be taken into account during controller design. With regard to heading control the relative motion in form of a self-induced yawing oscillation has to be considered.

Since PPGs are often used in harsh environments robust sensing equipment is beneficial. To that end, the control scheme presented here only requires heading information of the payload, aswell as ground speed and position. The heading can be obtained using off-the-shelf devices, while the ground speed and position are obtained by a GPS device. Notably, no airspeed information is required. This small number of required sensors is sufficient for our control scheme because of two assumptions which are made concerning the flight mechanics of PPGs: a constant airspeed and a negligible stationary side-slip.

2.2 Model Predictive Control

Model predictive control (MPC) is an umbrella term for a family of control algorithms which offer a conceptually simple solution to a wide range of control problems. While the core concept is simple, the implementation of an MPC often requires substantial processing power. This hindered the application of MPC in fast control loops, as those typically found in aircraft, in the past.

All MPC algorithms share the idea, that control action is based on predictions of the state trajectory of the plant. For different future control input sequences, different trajectories of the plant are predicted. These predicted trajectories are rated using a cost function. Finding the optimal control input sequence with regard to the cost function constitutes solving an optimization problem. In a closed-loop setting this optimization is carried out in every timestep and only the first element of the optimal input sequence is used as an actual input to the plant.

The complexity of solving the optimization problem is highly dependent on the model used. For linear models, analytical solutions can readily be derived. Nonlinear models on the other hand require nonlinear optimization algorithms, which typically require much processing power and a complex implementation.

One can identify four separate components which will be present in any MPC: the model, the observer, the cost function and the optimizer. A general discrete-time model is given by

$$x(k+1) = f\big(x(k), u(k)\big) \tag{1}$$

Here, the next state $x(k+1)$ is a (nonlinear) function f of the previous state $x(k)$ and the previous input $u(k)$. A sequence $x(\cdot)$ is called the (state) trajectory of the system. In general, the state x cannot be measured directly, but has to be calculated based on the measurable outputs and inputs $y(k) = g(x(k), u(k))$. Thus, most model predictive control schemes will include an observer.

The predicted trajectories are rated using a cost function of the form

$$J_H = \sum_{k=0}^{H-1} \ell(x_u(k), u(k)) \tag{2}$$

The index H denotes the prediction horizon, x_u is the state trajectory obtained using the input sequence $u(\cdot)$ and ℓ are the *pointwise* costs. Often, additional terms are included to improve or proof stability, but for this work the simple formulation in (2) is sufficient.

One of the advantages of MPC compared to other control schemes is its ability to incorporate constraints. These can be introduced at several points, either as hard constraints or as soft constraints. Hard constraints, as those often found in inputs, can be easily included as part of the model or the optimizer. Soft constraints (e.g. a flight state, which should be avoided) can be included by assigning a large cost to the corresponding state or input.

Given the cost function, the model and the currently observed state, the optimizer has to find the input sequence $u(\cdot)$ which minimizes the cost function J_H over the prediction horizon H and at the same time satisfies the given constraints. The optimal control problem (OCP) thus is:

$$
\begin{aligned}
&\text{Find } u^*(\cdot) = \underset{u(\cdot)}{\operatorname{argmin}}\ J_H = \underset{u(\cdot)}{\operatorname{argmin}} \sum_{k=0}^{H-1} \ell(x_u(k), u(k)) \\
&\text{where } x_u(0) = x_0 \qquad \text{(OCP)} \\
&\text{and } x_u(k+1) = f(x_u(k), u(k))
\end{aligned}
$$

while satisfying the given constraints.

Solving (OCP) yields an optimal control sequence $u^*(\cdot)$, where the first element provides the input to the plant during the next timestep.

Prediction Horizon The time covered by the prediction horizon H is $\Delta t = H \cdot T$. Since T effectively introduces a dead time into the control loop it is desirable to keep T small. However, for a fixed Δt the consequences of a smaller timestep T are twofold: First, we increase the number of timesteps we need to predict. Secondly, the computation time per timestep increases, because (OCP) needs to be solved for more variables. Due to this using an MPC quickly becomes infeasible for small timesteps T.

3 Conceptual Approach

A simplified block diagram of the horizontal flight path guidance is shown in Fig. 2. There are two control loops: the outer loop controls the horizontal flight path using a model predictive controller and generates heading commands for the inner loop, the heading controller. Dividing the controller into two loops, the "fast" heading controller loop and the "slow" flight guidance loop, has several advantages over a solution based on model predictive control only. The most apparent advantage is that it allows the reuse of an existing heading controller. At the same time, both controllers can be designed separately. From the viewpoint of the MPC, the heading controller is now part of the plant. One can achieve a mostly linear control response from commanded heading to actual heading by designing the heading controller correspondingly. Thus a linear model can be used to represent the heading-controlled

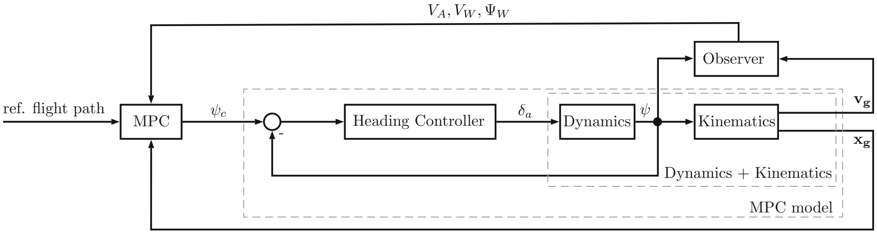


Fig. 2 MPC control loop

PPG. Another, more subtle effect is that, due to this division it is natural to let the "fast" heading controller run at a higher frequency than the "slow" flight path controller.

In addition to the linear model of the heading-controlled PPG, the underlying model of the MPC also includes the flight path kinematics, which are inherently nonlinear. Due to this, the optimization problem (OCP) also becomes nonlinear. To predict the future flight path, the current wind speed has to be estimated. This is done using a non-standard observer, which is only based on heading information and ground velocity measurements.

The reference flight path is constructed using straight-line segments. Turns are constructed in compliance with the maximum achievable turn rate and then approximated using multiple straight-line segments.

In the following sections, each component briefly presented above will be discussed in more detail.

3.1 Model of Heading-Controlled PPG

The model consists of two parts: the dynamics and the kinematics. The dynamics model the behaviour of the heading-controlled plant, thus the input is the commanded heading Ψ_c and the output is the actual heading Ψ. As mentioned before, the suspension of the payload leads to heading oscillations. While high-DOF models, which model the relative motion of the canopy and the payload, exist, considering these effects is not useful for the purposes of flight path guidance. This is due to the fact that the actuators are too slow to counteract the oscillations, thus they ideally should not react to them at all. Additionally, considering the flight path, heading oscillations about a setpoint will in effect still lead to an approximately straight-line path.

The model of the heading-controlled PPG is a transfer function obtained using black-box system identification, see Sect. 4.2. It is of the form

$$G_{\Psi\Psi_c} = \frac{1 + n_1 s + n_2 s^2 \dots}{1 + d_1 s + d_2 s^2 + \dots} \tag{3}$$

Note that, while most of the coefficients of the numerator and denominator polynomials need to be estimated, we fix $n_0 = 1$. In other words, the static-gain of $G_{\Psi\Psi_c}$ is 1, because we assume that the heading controller will achieve zero stationary control deviation. Additionally, $G_{\Psi\Psi_c}$ shall be strictly proper, since there should be no direct feed-through. For the actual discrete-time implementation, the dynamics (3) need to be discretized.

The kinematic model is based on the assumptions that airspeed, wind speed and wind direction only vary slowly and that the side-slip of the PPG is negligible. Thus the kinematics are given by

$$\begin{aligned} \mathbf{V_g} &= \mathbf{V_A} + \mathbf{V_W} \\ \mathbf{x_g} &= \int \mathbf{V_g}\, dt \end{aligned} \tag{4}$$

Here, $\mathbf{V_A}$ and $\mathbf{V_W}$ denote the aerodynamic and the wind speed respectively. As before, (4) needs to be discretzied for the actual implementation.

The complete model consists of the dynamics of the heading controller (3) and the kinematics (4).

To predict a trajectory of a system the initial states need to be known. The states of the kinematic model are comprised of the position $\mathbf{x_g}$, the airspeed V_A, the wind speed V_W and the wind direction Ψ_W. The states of the dynamic model are dependent on the actual dynamic model chosen in the system identification process. In the case of a discrete-time transfer function model without direct feed-through the output is a function of the previous outputs and the previous inputs:

$$\Psi(k) = -a_1\Psi(k-1) - a_2\Psi(k-2) - \dots + b_1\Psi_c(k-1) + b_2\Psi_c(k-2) + \dots$$

Thus the previous inputs and outputs can be considered states of the model. Akin to internal model control (IMC), the previous outputs $\Psi(k-1), \Psi(k-2), \dots$ are obtained by updating an internal model at each timestep k with the previously chosen input $\Psi_c(k-1)$. This has the advantage, that the heading oscillations of the PPG do not affect the MPC model. However, this approach only works, if the plant and the model do not diverge over time. We assume that this is given, since the dynamics model has a static gain of 1 and the heading controller is designed accordingly. Using this technique effectively decouples the flight guidance loop from the dynamics of the PPG.

3.2 Observer

As mentioned in Sect. 3.1 the model state is given by

$$\mathbf{x} = \left[\mathbf{x_g}\ V_A\ V_W\ \Psi_W\ \Psi(k-1)\ \Psi(k-2)\ \dots\ \Psi_c(k-1)\ \Psi_c(k-2)\dots\right]^T \tag{5}$$

All state variables, but the airspeed V_A, wind speed V_W and wind direction Ψ_W, are either directly measurable or known. Based on the assumption of negligible side-slip, slowly varying airspeed, wind speed and wind direction (c.f. Sect. 3.1) the unknown state variables V_A, V_W and Ψ_W can be observed from available measurements.

We first introduce the ground speed in direction of the current heading $V_{g,\Psi}$, which can be calculated using the following formula:

$$V_{g,\Psi} = \left|\begin{pmatrix} V_{g,x} \\ V_{g,y} \end{pmatrix}\right| \cos(\chi - \Psi) = V_g \cos(\chi - \Psi) \quad \text{where} \quad \chi = \operatorname{atan2}(V_{g,y}, V_{g,x}) \tag{6}$$

Fig. 3 Velocity vectors

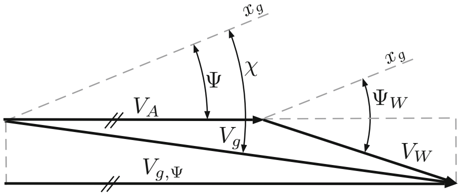


Figure 3 shows the relationship between $\mathbf{V_A}$, $\mathbf{V_W}$ and $\mathbf{V_g}$ and gives the following equation for $V_{g,\Psi}$:

$$V_{g,\Psi} = V_A + V_W \cos(\Psi - \Psi_W) \tag{7}$$

In principle, given a set of observations $V_{g,\Psi}$ and Ψ the unknown parameters V_A, V_W and Ψ_W can be determined using nonlinear regression. A more efficient approach is to introduce a nonlinear transformation such that the unknown parameters can be determined using linear regression:

$$\begin{aligned} V_{g,\Psi} &= V_A + V_W \cos(\Psi - \Psi_W) \\ &= V_W \cos(\Psi_W) \cos(\Psi) + V_W \sin(\Psi_W) \sin(\Psi) + V_A \\ &= V_{Wx} \cdot \cos(\Psi) + V_{Wy} \cdot \sin(\Psi) + V_A \end{aligned} \tag{8}$$

Equation (8) is linear in the parameters V_{Wx}, V_{Wy} and V_A. The transformation back to the original parameters is given by

$$V_W = \sqrt{V_{Wx}^2 + V_{Wy}^2} \tag{9}$$

$$\Psi_W = \operatorname{atan2}(V_{Wy}, V_{Wx}) \tag{10}$$

For N observations the following system of equations can be formulated:

$$\begin{pmatrix} \cos(\Psi_1) & \sin(\Psi_1) & 1 \\ \cos(\Psi_2) & \sin(\Psi_2) & 1 \\ \vdots & \vdots & \vdots \\ \cos(\Psi_N) & \sin(\Psi_N) & 1 \end{pmatrix} \begin{pmatrix} V_{Wx} \\ V_{Wy} \\ V_A \end{pmatrix} = \begin{pmatrix} V_{g,\Psi,1} \\ V_{g,\Psi,2} \\ \vdots \\ V_{g,\Psi,N} \end{pmatrix} \Leftrightarrow \mathbf{Ax} = \mathbf{b} \tag{11}$$

The least-squares problem corresponding to Eq. (11) can be solved using standard algorithms. However, the choice of observations to include in the regression is not as obvious. Always choosing the last N observations can quickly lead to a case where $\mathbf{A}$ becomes singular, for example when the PPG is flying with a constant heading. In this case, all rows of $\mathbf{A}$ will be nearly equal and the least-squares problem becomes badly conditioned. Instead, we divide the heading range from -180 to 180° into N equal slices. For each slice, P observations (Ψ, $V_{g,\Psi}$) are recorded and thus a mean and a variance can be calculated. A large variance in a slice will likely be due to changes in the wind vector, thus for the regression this slice should have a smaller influence on the estimate than a section with low variance. By introducing the diagonal matrix $\mathbf{W} = \mathrm{diag}({}^{1}/_{\sigma_i})$, where σ_i is the standard deviation in the i-th slice, a new regression problem can be formulated:

$$\hat{\mathbf{b}} = \hat{\mathbf{A}}\mathbf{x} \quad \text{where} \quad \hat{\mathbf{b}} = \mathbf{W}\overline{\mathbf{b}} \quad \text{and} \quad \hat{\mathbf{A}} = \mathbf{W}\overline{\mathbf{A}} \tag{12}$$

$\overline{\mathbf{b}}$ and $\overline{\mathbf{A}}$ are defined using mean values in each slice as described above. The observer can be classified as an online least-squares estimator with directional forgetting [1].

3.3 Cost Function

The cost function penalizes deviations of a predicted flight path from the reference flight path. As stated in Eq. (2) the total cost is the sum of pointwise costs ℓ. The main goal is to minimize the distance to the reference flight path. We use the standard practice to minimize the squared distance, since it enables a more efficient implementation. In addition to penalizing the distance we also penalize deviations from the reference flight path azimuth. This prevents the optimizer from finding solutions which are close to the reference flight path, but opposite to the direction associated with the current line segment. To allow approaching the reference flight path in a 90° angle, this term is weighted with the inverse of the distance to the reference flight path. The pointwise cost function is thus

$$\ell(x, u) = \mu_1 d^2(x, x_{ref}(\cdot)) + \mu_2 \frac{\Delta\chi^2}{d(x, x_{ref}(\cdot))} \tag{13}$$

where the current fligh path azimuth is given by $\chi_i = \mathrm{atan2}(y_{g,i+1} - y_{g,i}, x_{g,i+1} - g_{g,i})$, $\mathbf{x_i} = (x_{g,i}, y_{g,i})$ is the i-th element of the predicted flight path and $d(x, x_{ref}(\cdot))$ denotes the distance from the reference flight path. Equation (13) introduces the additional weighting factors μ_1 and μ_2, which make the corresponding terms dimensionless. For the optimization, the relative weighting of both terms is essential, so μ_1 can be fixed to $1/\mathrm{m}^2$. μ_2 should be large enough to prevent the optimizer from choosing the wrong directions. Suitable values can be found in simulation studies. The cost function (13) only penalizes states, not inputs, since the hard input constraints are handled during optimization.

3.4 Optimizer

The optimizer's task is to minimize the cost function J_H (2) with regard to the input sequence $u(\cdot)$. Given the dynamics of the heading-controlled PPG we know that a maximum absolute commanded turn rate $|\dot{\Psi}_{c,\max}|$ can be achieved. This creates

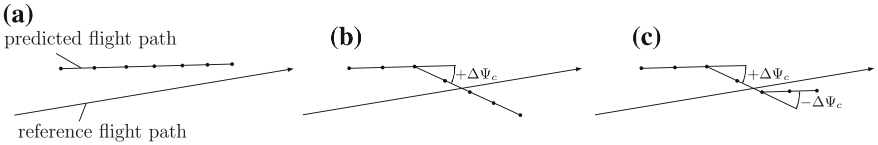


Fig. 4 Optimizer working principle

a constraint, which has to be satisfied during optimization. Since the optimization problem (OCP) is formulated in a discrete-time setting, the maximum absolute commanded turn rate $\dot{\Psi}_{c,\max}$ is replaced by a maximum change in the commanded heading $|\Delta\Psi_{c,\max}|$.

A common problem in solving (OCP) is that changes in the inputs have non-local effects, i.e. a change in the commanded heading affects the entire predicted flight path in the following time steps. A consequence of this property is that minimizing J_H by optimizing the input sequence at one point can easily lead to the situation depicted in Fig. 4b. Initially the pointwise cost ℓ decreases, but increases again after the predicted trajectory crosses the reference flight path. In total, the costs J_H might not be lower for the predicted flight path in Fig. 4b than for the initial path in Fig. 4a. Thus the predicted flight path in Fig. 4b will not be considered further during gradient-based optimization. An effective way to circumvent this problem is to introduce changes to the commanded heading not only at one, but at two points in the predicted flight path. This leads to the predicted flight path shown in Fig. 4c.

The optimization algorithm works by successively introducing opposing changes at two prediction points, denoted by indices i and j. The index i runs from 1 to $H - 1$, while the index j runs from $i + 1$ to H. For each pair (i, j) a small change $\Delta\Psi_c$ is introduced to the commanded heading (added at i, substracted at j and vice versa). This process continues as long as the total cost decreases, otherwise the algorithm advances to the next pair (i, j). The number of iterations per pair (i, j) is bounded, because the introduced change is limited to $\pm\Delta\Psi_{c,\max}$. Additionally, the total number of iterations is bounded, to ensure that the algorithm produces a result in the required time T. The number of index pairs (i, j) is in the order of $\frac{1}{2}H^2$. At each pair, the model has to be executed H times to give the predicted flight path. Thus the total complexity of the algorithm is of order H^3.

If it takes T_{once} seconds to step the model once and calculate the pointwise cost ℓ, predicting a complete flight path takes $H \cdot T_{\text{once}}$ seconds. In the worst case, the optimizer needs to evaluate $\frac{\Delta\Psi_{c,\max}}{\Delta\Psi} \cdot H^2$ complete trajectories. For a fixed prediction time Δt and an MPC timestep of T, the number of prediction steps H is $\frac{\Delta t}{T}$. The smallest MPC timestep is thus found to be

$$T_{\min} = \sqrt[4]{\frac{\Delta\Psi_{c,\max}}{\Delta\Psi_c}\Delta t^3 T_{\text{once}}} \tag{14}$$

The values of Δt and $\Delta\Psi_{c,\max}$ depend on the plant dynamics, while T_{once} and $\Delta\Psi_c$ are design variables.

4 Implementation

The following section discusses the implementation of the previously presented control scheme.

Technical Data The actual PPG used was a SUSI62 provided by Dr. H.-P. Thamm Geo-Technic (Fig. 5). Its technical data are summarized in the Table 1.

The flight control hardware was developed at the Institute of Flight System Dynamics of the RWTH Aachen University. Besides an integrated IMU and a GPS module it has two dedicated microcontrollers for running control algorithms. Both microcontrollers are clocked at 164 MHz and contain a floating point unit (FPU). Due to this architecture, the previously discussed division between heading control and flight path control could be easily transferred to the actual implementation. While the heading controller runs at a frequency of 200 Hz, the flight path controller runs only at 1 Hz, see Sect. 4.4.

Fig. 5 The powered paraglider SUSI62

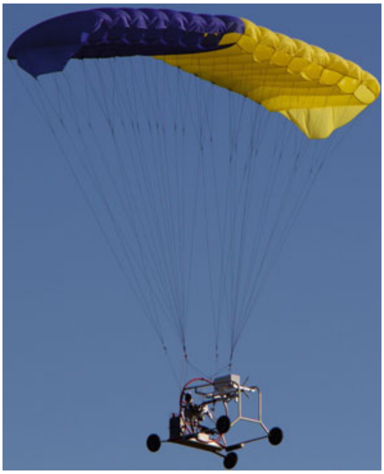


Table 1 Technical data

Wing area	6 m^2
Empty weight	$\approx$ 10 kg
Payload capacity	up to 5 kg
Airspeed	$\approx$ 8 m/s
Motor	62 cm^3, 2-stroke, gasoline
Flight time	up to 1 h

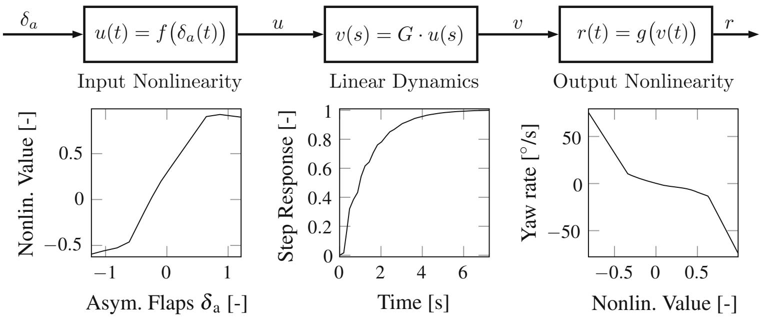


Fig. 6 Nonlinear Hammerstein–Wiener model

4.1 Heading Controller

Heading control is achieved by a PID-based controller. In contrast to other works [7] the process model is a black-box model obtained using system identification. The input to the system is the asymmetric brake deflection $\delta_a = \delta_{left} - \delta_{right}$. The output is the yawing rate r, which approximates the heading rate of change $\dot{\Psi}$ for small bank angles. To capture the nonlinear properties of both the actuators and the flight dynamics of the PPG we use a nonlinear Hammerstein–Wiener model. Hammerstein–Wiener models combine static input and output nonlinearities with a linear system in between.

Figure 6 shows the characteristics of the individual components of the identified model. Several conclusions can be drawn from these figures: The input nonlinearity shows a non-zero output in case of a zero input. Additionally the flap's effect does not change beyond approximately −0.2 in the negative or 0.3 in the positive direction. The output nonlinearity shows a plateau from −0.4 to 0.5 units, which has the effect of a dead-zone as well as of a dead-time. The step response of the linear block shows a rise time of about 2 s.

On the basis of these nonlinear dynamics we tuned a PID controller which achieves zero stationary control deviation and a linear response for reasonably small input changes. Figure 7 shows data from flight tests, where the commanded heading was controlled manually. Even though the heading controlled plant exhibits large overshoots, this behaviour is acceptable as long as a suitable model can be found.

4.2 Kinematic and Dynamic Model

Dynamics As implied by the considerable overshoots, a second-order model is needed to describe the plant behaviour to a sufficient accuracy. Based on flight-test data the following continuous-time transfer function model was identified:

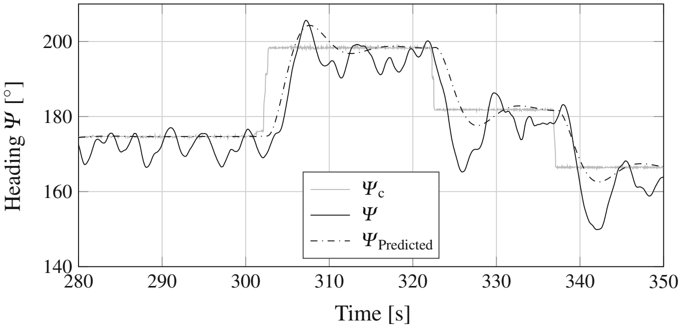


Fig. 7 Response of heading controller

$$
\begin{aligned}
G(s) &= \frac{K\cdot(1+T_z s)}{1+2\zeta T_w s+(T_w s)^2} \\
\text{where}\quad K = 1;\ T_z &= -0.29\text{ sec};\ T_w = 1.47\text{ sec};\ \zeta = 0.40
\end{aligned} \tag{15}
$$

In addition to the plant behaviour, Fig. 7 shows the model response Ψ_{Model}. The model shows a similar behaviour with regard to the step response aswell as the overshoots to some degree. The large oscillations caused by the suspension cannot be represented since these are not caused by a measurable input to the plant. Since the MPC is formulated in a discrete-time setting, Eq. (15) was discretzied using a zero-order hold method and the MPC sample time T.

Kinematics The observer discussed in Sect. 3.2 estimates the airspeed, wind speed and wind direction. Due to this, the velocities in (4) are expressed in polar coordinates. Discretizing (4) with a sample time of T yields

$$
\begin{aligned}
\mathbf{v_g}(k) &= V_A(k)\begin{bmatrix}\cos\Psi(k)\\ \sin\Psi(k)\end{bmatrix} + V_W(k)\begin{bmatrix}\cos\Psi_W(k)\\ \sin\Psi_W(k)\end{bmatrix} \\
\mathbf{x_g}(k+1) &= \mathbf{x_g}(k) + \mathbf{v_g}(k)\cdot T
\end{aligned} \tag{16}
$$

We use euler-integration because simulation results indicated no improvement when using more elaborate integration schemes.

4.3 Wind Estimation

Section 3.2 introduced the observer that is used to estimate the aerodynamic speed and the wind velocity. The two essential parameters of this observer are the number of sections N and the number of observations per section P. As a first approximation, N influences the convergence time and P the noise of the estimate. In simulation studies $N = 12$ and $P = 10$ were found to result in a good tradeoff between convergence speed and noise level.

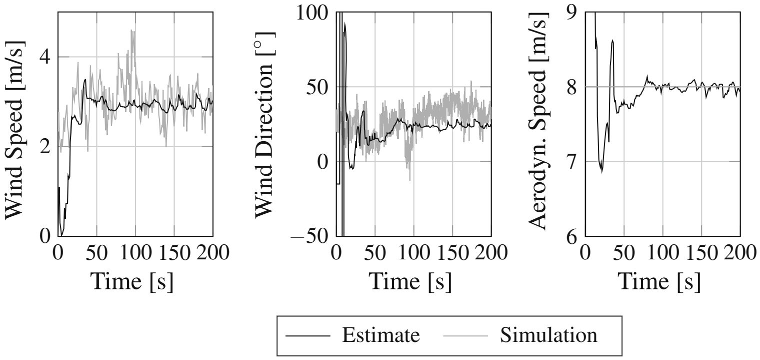


Fig. 8 Observer simulation results

Figure 8 shows one such simulation study. The simulated wind is based on measured wind data. During the first 50 s the heading slices are filled with new observations, leading to large changes in the estimate. The length of this initial phase depends on the actual flight path. After this initial phase, the solution stabilizes, effectively smoothing the simulated velocities.

4.4 Optimizer

The main advantage of the optimization algorithm presented in Sect. 3.4 is its conceptual simplicity. While this simplicity also translates to its implementation, the runtime of the MPC is still the major limiting factor for an application to faster control loops.

MPC timestep Equation (14) allows a calculation of the minimum MPC timestep T_{min}. It takes about $T_{\text{once}} = 19\mu\text{s}$ to step the model once and calculate the pointwise cost ℓ. Given the rise time of about 2 s of the plant (see Sect. 4.1), the prediction horizon was $\Delta t = 10$ s. The maximum heading change in one timestep was set to $\Delta\Psi_{c,\text{max}} = 0.3\text{rad}$ ($\approx 17°$), while the incremental change was set to $\Delta\Psi_c = 0.05$ rad ($\approx 3°$). This results in a minimum MPC timestep of $T_{\text{min}} = 0.58$ s (i.e. a maximum control frequency of 1.72 Hz). Since T_{min} only constitutes the MPC computation time, we set the actual MPC timestep to 1 s to allow the microcontroller to execute other tasks besides computing the heading command.

Initial solution Starting optimization from a good initial solution considerably reduces the required iterations to find the optimal solution. If the model is sufficiently accurate we assume that the predicted flight path will be close to the actual future flight path. The next optimal solution will then be close to the current optimal solution, minus the first element. Using the last optimal solution to initialize the current solution reduces the required number of iterations considerably. Once a solution is found, it takes only about 50 iterations to find the optimal solution in the subsequent timesteps. This results in a computation time of about 5 ms, which is well below the theoretical worst-case of about 500 ms.

5 Evaluation

The following section presents an evaluation of the previously described control scheme. The individual components of the MPC will be discussed separately where applicable. First, results from a simulation study are discussed and then compared to results from flight tests.

Reference Flight Path All results are obtained using the same reference flight path. The flight tests were conducted at a small airfield with a limited area of about 250 m ×500 m. Thus the primary requirement for the reference flight path was to stay within these allowed limits. We chose a flight path in shape of the number 8, featuring both left and right turns. The minimum turn radius is 70 m. With an airspeed of 8 m/s and a maximum expected wind speed of 5 m/s this corresponds to a maximum turn rate of 0.16 rad/s (≈9°/s). This is well within the maximum turn rate of $\dot{\Psi}_{c,\max} = 0.3$ rad/s (≈17°/s) imposed by the heading controller.

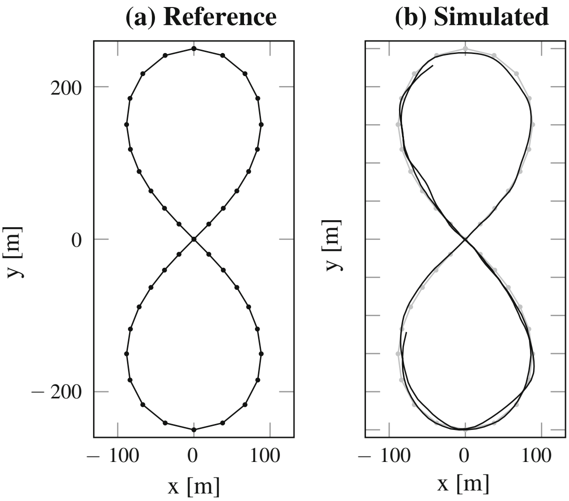
5.1 Simulation Study

The following simulation results were obtained with a mean wind speed of $V_W = 3$ m/s in direction of the positive y-axis ($\Psi_W = 90°$). Wind gusts up to 1 m/s were added to the mean wind velocity. The estimated velocities behave similiarly to the estimates in Fig. 8 and thus are not discussed further. For the simulation, we used the nonlinear model described in Sect. 4.2.

Figure 9 shows the reference flight path and the simulated flight path. The simulated flight path stays within approximately ±10 m of the reference flight path. As is to be expected, the largest deviations occur when the velocity estimation is inaccurate. This inaccuracy is however part of the overall design, since the actuators of the PPG are too slow to counteract wind gusts. Additionally, the deviation from the reference flight path rises during fast turn maneuvers, where the PPG achieves comparatively large turn rates. This suggests an insufficient model accuracy for this flight state.

Effects of inaccurate velocity estimates The observer is based on the assumption that the observed variables are only varying slowly. Due to its working principle, changes in the wind direction or speed can only be detected reliably if enough new observations are gathered in new slices. Especially during long straight-line flight segments this condition will not be satisfied. To quantify the effect of erroneous estimtates of wind velocity and airspeed, we carried out simulation studies. In these

Fig. 9 Simulated flight along reference flight path



studies, the velocity estimates were set to a fixed value while an additional crosswind was introduced in the simulation. We found that the deviation from the reference path rises linearly with the error in wind speed estimation. An erroneous wind speed estimate of 5 m/s will lead to an error of about 30 m. The relevance of this result depends on the properties of the reference flight path and the required accuracy.

It should be noted that even in the case of an unaltered estimate, a mean deviation of about 2 m is present. This seems to result from the optimizers discretized search space. The optimizer is not able to find a commanded heading that leads to zero mean deviation.

5.2 Flight Test

A flight test was conducted to validate the results obtained in the simulation study described previously. Due to a biased heading measurement, the flight path guidance results are distorted and not included here. The effects of this bias are the same as those of inaccurate velocity estimates. Thus, resulting from an incorrect wind correction angle the actual flight path was offset from the reference flight path by about 20 m. Nevertheless, the resulting flight path was similar to the simulation results with respect to its variation and reproducability.

Only those components of the MPC not affected by the biased heading measurement will be discussed below.

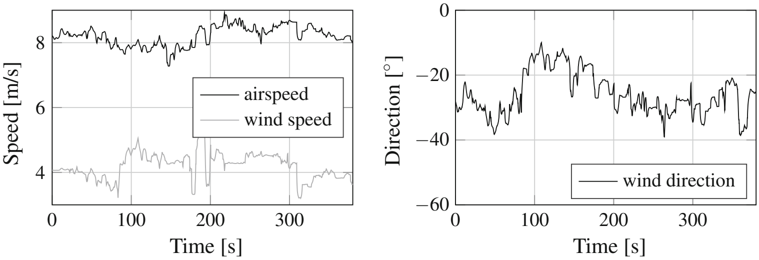


Fig. 10 Velocity estimation in flight test

5.2.1 Observer

Figure 10 shows the estimated wind speed and airspeed during one test flight. After take-off, the estimated airspeed converges to approximately 8 m/s, which is in the expected range. Due to the lack of airspeed sensors the estimated velocities could only be checked for plausibility and couldn't be validated further.

5.2.2 Model

Figure 11 compares the measured heading during a flight with the predicted heading. The left plot shows the heading commanded by the model predictive flight path controller. The heading controller has a considerable lag of approximately 2 s. The right plot shows the heading predicted by the internal dynamics model. Of course, not all dynamics are described by the simple linear model. A typical example of this can be seen around 610 s. Here, shortly after the turn starts, the payload swings back again before following the motion of the parafoil. Besides such nonlinear behaviour, the lag and the rise time of the plant are represented well by the linear model.

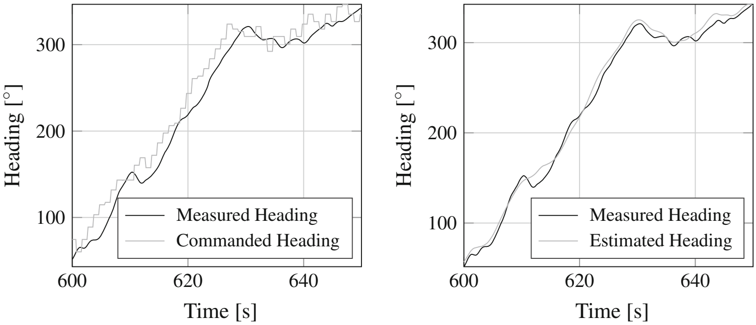


Fig. 11 Comparison between measured and predicted heading

It should be noted, that the model was identified during a flight where the heading was commanded manually. In contrast, the data underlying Fig. 11 was produced during an MPC-guided flight. Performing another system identification procedure on these flight data results in a model, very similar to the original model. Thus, despite the difference in excitation, the previously identified model could be used for the MPC without change.

6 Conclusion

We presented a nonlinear model predictive control scheme for the horizontal flight path guidance of a PPG. Model predictive controllers are inherently more complex than many classic control solutions. Because of this, the control scheme presented here was designed with a focus on simplicity, both conceptual and regarding its implementation. Using a heuristic optimization algorithm we were able to implement the control scheme on an embedded flight control platform. The control scheme adopts the common division of flight path control into heading control and flight path guidanace. While the heading controller was designed as a classic feedback controller, the flight path guidance was realized using an MPC. To predict the future flight path of the underlying model, an estimation of the mean wind velocity and airspeed was implemented. The estimation was transformed to a linear least-squares problem, yielding a robust estimation of the observed variables at a reduced computational cost.

The control scheme was developed and tested in a simulation environment. Here, a maximum deviation from the reference fligth path of 10 m was achieved. Flight tests were then conducted to validate this simulation result. While the actual flight path guidance could not be validated due to a biased heading measurement, the individual components of the MPC could be validated successfully. It was shown that the linear model of the heading dynamics describes the behaviour of the heading-controlled PPG well. Also, the estimation of the wind speed and airspeed yielded plausible results.

References

1. Astrom KJ, Wittenmark B (2008) Adaptive control, 2 edn. Dover Publications, New York
2. Cumer C, Toussaint C, Le Moing T, Poquillon E, Coquet Y (2012) Simulation of generic dynamics flight equations of a parafoil/payload system. In: 2012 20th mediterranean conference on control & automation (MED)
3. Grüne L, Pannek J (2011) Nonlinear model predictive control. Springer, London
4. Hur G, Valasek J (2003) System identification of powered parafoil-vehicle from flight test data. In: AIAA atmospheric fligth mechanics conference and exhibit
5. Jann T (2001) Aerodynamic model identification and GNC design for the parafoil-load system ALEX. In: 16th AIAA aerodynamic decelerator systems technology conference and seminar

6. Joos A, Fichter W (2011) Parallel implementation of constrained nonlinear model predictive controller for an FPGA-Based onboard flight computer. Springer, Berlin
7. Ochi Y, Kondo H, Watanabe M (2009) Linear dynamics and PID flight control of a powered paraglider. In: AIAA guidance, navigation and control conference
8. Rogers RM, et al. (2002) Aerodynamic parameter estimation for controlled parachutes. In: AIAA atmospheric flight mechanics conference and exhibit
9. Rossiter JA (2003) Model-based predictive control. Taylor & Francis Inc., London
10. Slegers N, Costello M (2005) Model predictive control of a parafoil and payload system. J Guid Control Dyn 28(4)
11. Slegers N, Kyle J, Costello M (2006) Nonlinear model predictive control technique for unmanned air vehicles. J Guid Control Dyn, vol 29
12. Slegers N, Brown A, Rogers J (2015) Experimental investigation of stochastic parafoil guidance using a graphics processing unit. Control Eng Pract, vol: 36
13. Toglia C, Vendittelli M (2010) Modeling and motion analysis of autonomous paragliders. Department of computer and system sciences Antonio Ruberti technical reports, 2(5)
14. Watanabe M, Ochi Y (2007) Modeling and motion analysis for a powered paraglider (ppg). In: SICE, 2007 annual conference, IEEE, pp 3007–3012

Multiple-Phase Trajectory Optimization for Formation Flight in Civil Aviation

Sander Hartjes, Marco E.G. van Hellenberg Hubar and Hendrikus G. Visser

1 Introduction

The potential to significantly reduce aircraft induced drag in extended formation flight has been clearly demonstrated in a range of numerical and experimental studies [1–4]. In contrast to close formation flight, in an extended formation aircraft are longitudinally separated by 5–40 wingspans. The studies presented in [1–4] all agree that extended formation flight has the potential to significantly reduce fuel and operating costs of long-haul airline operations.

One of the key areas on which past research on (extended) formation flight has focused is the planning and organization of flight formations on a network-wide scale [3–8]. Most studies addressing the problems of global routing and assignment typically rely on a bi-level (or two-stage) approach. At the first level/stage, the routing/mission design problem is considered. In the routing/mission design problem, a candidate set of two or three long-haul origin/destination flights is considered that might join in, respectively, a two or three aircraft formation. The routing/mission problem deals with locating the rendezvous and splitting points for the flights involved and with scheduling the associated altitude/speed profiles such that the overall mission (fuel) cost is minimized. Of course, an optimized flight formation option is only accepted as a mission possibility if it saves (fuel) cost relative to flying the aircraft solo from origin to destination. The second level/stage then seeks to optimize the network by selecting the best subset of formation and solo missions given the complete set of all possible combinations of individually optimized formation and solo missions. It is noted that the second level/stage is highly combinatorial in nature, as the number of options to join flights in formations grows dramatically with the number of flights in the network. Given the fact that for a network of realistic size, a huge number of formation options needs to be evaluated,

S. Hartjes · M.E.G. van Hellenberg Hubar · H.G. Visser (✉)
Faculty of Aerospace Engineering, Delft University of Technology,
P.O. Box 5058, 2600 GB Delft, The Netherlands
e-mail: h.g.visser@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_21

it is readily clear that for the first stage mission design problem a computationally efficient optimization approach is warranted. For this reason, simplified, low fidelity aircraft models have typically been adopted to rapidly assess the (fuel) cost and trip times for the formations missions. Aircraft performance models employed for this purpose range from the Breguet-range equation at constant altitude evaluated along a geometrically constructed route [6], to a parameterized aircraft performance model in a three-dimensional space, featuring a limited number of design parameters [7]. Although computationally efficient, these low fidelity models do not provide very accurate results and are not quite capable of dealing with real-world complexities such as wind/weather influences and schedule delays.

In this study, a higher-fidelity mission design optimization model is developed that is capable of dealing with these real-world complexities at a reasonable computational cost. More specifically, the study presented in this paper deals with the development of a multi-phase optimal control framework that supports the synthesis of two-aircraft long-haul formation missions, using a point-mass dynamic model formulation. For each of the two flights considered, the origin/destination city pair is given, along with the scheduled departure time. The primary reason for resorting to a multi-phase trajectory optimization formulation for the mission design problem is that it allows the concurrent optimization of the single two-aircraft formation flight leg, and the four solo flights legs connecting, respectively, the two origin airports to the rendezvous point and the two destination airports to the splitting point of the formation. It is noted that in the adopted multi-phase set-up both aircraft are forced to join in formation, even when flying the city-pair trajectories in solo flight might be more favorable. For this reason, the optimization framework also assesses the optimal performance of the two city-pair (great-circle) trajectories flown solo, to enable a comparison. The potential (fuel) cost improvements realized by formation flight are then assessed against the cost of the combined solo flights.

In a case study, involving the Transatlantic crossing of two Boeing B747-400 aircraft, the developed multiple-phase optimization tool is deployed to optimize the trajectories of the aircraft that join in formation and experiments are conducted to investigate what the general characteristics and the potential benefits of formation flight are. In particular, it is explored whether fuel savings can still be obtained if no increase in trip time is permitted relative to flying solo. Also, it is assessed to what extent (fuel) cost benefits can still be achieved when one of the participating aircraft suffers a departure delay.

2 Trajectory Optimization Formulation

2.1 Flight Path Modelling

In this study we seek to optimize a two-aircraft flight formation for minimum cost or fuel burn. A basic scenario is considered in which two independent long-haul

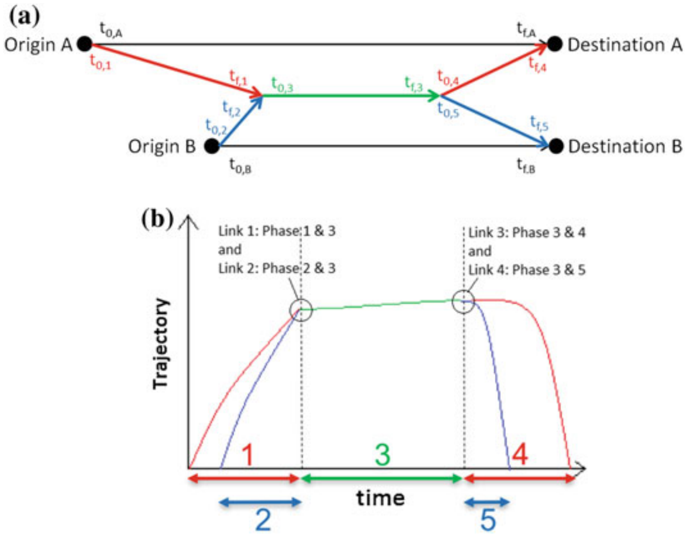


Fig. 1 A schematic representation of the five-phase assembly of a two-aircraft formation

flights are assumed to depart from closely located origin airports towards two closely located destination airports, flying minimum (fuel) cost great circle trajectories. It is assumed in this scenario that the considered flights can join in formation with modest deviations from their original solo paths. Also, a no-wind condition is assumed in the baseline scenario. Another major assumption relates to scheduled departure times of the two flights considered. Essentially, it is assumed in the basis scenario that the departure times of the two flights are perfectly synchronised to achieve minimum (fuel) cost. In a next stage, the influence of departure times that are not perfectly synchronised is explored.

In Fig. 1a, the basic routing problem in the baseline scenario is sketched, assuming two flights, labelled A and B, respectively. Both the solo and formation flight missions are indicated in Fig. 1a. The two independent solo flights depart at times $t_{0,A}$ and $t_{0,B}$, respectively, arriving at their destinations at $t_{f,A}$ and $t_{f,B}$, respectively. The formation flight is modelled as a multi-phase problem, comprising 5 stages. The first two stages connect the departing flights to the rendezvous point, where the formation stage is started (stage 3). Stages 4 and 5 connect flights A and B to their respective destination airports, after the formation has split up. The initial times for stages 1 and 2 are, respectively, $t_{0,1} = t_{0,A}$and $t_{0,2} = t_{0,B}$. For rendezvous it is required that at the end of stages 1 and 2, $t_{f,1} = t_{f,2} = t_{rendezvous} = t_{0,3}$. It is noted that in the baseline scenario pertaining to the two-aircraft formation only the time instance $t_{rendezvous}$ is specified a priori. The ideal departure times $t_{0,1}$ and $t_{0,2}$ for flights A and B joining in formation are then determined in the optimization process. The imposed departure times $t_{0,A}$ and $t_{0,B}$ for the single flights are set equal to the values $t_{0,1}$ and $t_{0,2}$ a posteriori, to allow a reference solution based on solo flights.

Figure 1b shows the five-phase solution schematically in the vertical plane, highlighting the required phase linking conditions. As the aircraft will not join in formation during the take-off and approach phases of flight, these segments will remain the same for aircraft that will or will not join in formation For this reason, the take-off and landing phases are not considered in this study and, the initial and final points of the mission are, respectively, the entry and exit points of the Terminal Movement Area (TMA), located at an altitude of 10,000 ft AGL.

2.2 Equations of Motion

The system of differential equations of motion employed in this study essentially describes the movement of two variable-weight point masses in a three-dimensional space. The employed system model is somewhat simplified in the sense that equilibrium of forces normal to the flight path is assumed (i.e., a so-called intermediate point-mass model is employed [9]) for both aircraft, flying either solo or in formation. The resulting equations of motion for the solo flight legs (flights A and B), as well as for the lead aircraft in the formation flight leg can be written as:

$$\dot{\phi} = \frac{V\,cos\gamma\,sin\chi \;+\; V_{W_N}}{R_E + z} \tag{1}$$

$$\dot{\lambda} = \frac{V\,cos\gamma\,cos\chi \;+\; V_{W_E}}{(R_E + z)\,cos\phi} \tag{2}$$

$$\dot{z} = V\,sin\gamma \tag{3}$$

$$\dot{V} = \frac{g_0(T - D)}{W} - g_0\,sin\gamma \tag{4}$$

$$\dot{\chi} = \frac{g_0}{V cos\gamma}\frac{L\,sin\mu}{W} \tag{5}$$

$$\dot{W} = -F_c \quad , \tag{6}$$

where ϕ is the latitude, λ is the longitude, z is the altitude, R_E is the Earth radius, V is the airspeed, γ is the flight path angle, χ is the heading angle, W is the aircraft gross weight, g_0 is the gravitational acceleration, T is the thrust, D is the drag, μ is the bank angle and F_c is the fuel flow. The fuel flowF_c is a function of altitude, velocity and thrust, i.e., $F_c = F_c$ (z, V, T). Furthermore, V_{W_N} and V_{W_E} are components of the wind velocity in North-South direction, and East-West direction, respectively. However, in the baseline scenario the wind components are ignored. The intermediate point-mass model comprises six state variables (the left-hand side in the equations above) and three control variables: the engine control setting η, the aerodynamic roll angle μ and the flight path angle γ. The engine control setting variable η, which is subject to the constraint $0 \leq \eta \leq 1$, has the following effect on the thrust of the aircraft:

$$T = (T_{max} - T_{min})\eta + T_{min} \quad , \tag{7}$$

where $T_{max}(h, V)$ and $T_{min}(h, V)$ are the maximum and idle thrust of the engine, respectively. The aerodynamic forces are the usual. For lift:

$$L = C_L q S = C_L \frac{1}{2} \rho V^2 S \quad , \tag{8}$$

where C_L is the lift coefficient, q is the dynamic pressure, ρ is the air density and $S$is the wing surface area. And for drag:

$$D = C_D(C_L) q S \quad \Rightarrow \quad D = D(z, V, C_L) \quad , \tag{9}$$

where C_D is the drag coefficient for which a parabolic drag polar will be assumed:

$$C_D = C_{D_0}(M) + K(M) C_L^2 \quad , \tag{10}$$

which consists of two parts: the zero-lift drag component C_{D_0} and the induced drag component $K(M)C_L^2$. Both drag coefficient components are dependent on the Mach number M.

For the trailing aircraft in the formation flight leg, a highly simplified system model is used. Essentially, the speed and altitude dynamics that govern the lead aircraft are also used to govern the trailing aircraft. As a result, the two flights A and B essentially operate as one entity in the formation leg. However, as the weight and the drag characteristics of the trailing aircraft in the formation are typically not the same as for the lead aircraft, the thrust level required for station keeping is different from the lead aircraft as well. The thrust level required for the trailing aircraft can be readily assessed from the specific energy rate balance:

$$\dot{E}_{lead} = \dot{E}_{trail} \quad \Rightarrow \quad \left[\frac{V(T-D)}{W}\right]_{lead} = \left[\frac{V(T-D)}{W}\right]_{trail} \tag{11}$$

And thus:

$$\begin{aligned} T_{trail} &= \left[\tfrac{(T-D)}{W}\right]_{lead} \cdot W_{lead} + D_{trail} \ , \\ subject\ to: &\quad 0 \leq \eta_{trail} = \tfrac{(T_{trail} - T_{min})}{(T_{\max} - T_{min})} \leq 1 \end{aligned} \tag{12}$$

It is noted that specific energy E is defined as:

$$E = z + \frac{V^2}{2g_0} \tag{13}$$

The reduction in fuel consumption due to formation flight is a result of a reduction in induced drag for the trailing aircraft. To evoke this induced drag reduction, a formation flight induced drag reduction factor ε is introduced in the parabolic drag polar of the trailing aircraft

$$C_D = C_{D_0} + \ (1 - \varepsilon) K C_L^2 \quad , \tag{14}$$

A fairly wide range of (measured or calculated) reduction factors can be found in the literature. In this case study a fairly conservative value in induced drag reduction of 25% ($\varepsilon = 0.25$) is assumed for the trailing aircraft of a two-aircraft formation. The adopted value is based on the results reported in [8], for a weight ratio 1 between lead and trailing aircraft.

The aircraft performance model that has been considered in this study pertains to a Boeing B747-400 wide-body aircraft [10]. The same performance model is used for both aircraft in the formation (or in solo flight), excepting the induced drag reduction factor. The take-off weight considered for the two aircraft in a (solo or formation) mission is typically different, and largely depends on the specified O/D stage length.

2.3 Optimization Criteria

The primary goal of flight formation is to achieve an overall reduction in fuel burn. The performance index related to the combined fuel consumption of flights A and B is defined separately for the single flights and the two-aircraft formation flight missions. For the solo mission, the fuel burn criterion is defined as:

$$J_{fuel,single} \ = \ [W_A(t_{0,A}) \ - \ W_A(t_{f,A})] \ + \ [W_B(t_{0,B}) \ - \ W_B(t_{f,B})] \tag{15}$$

For the formation flight mission, the fuel burn criterion aggregates the fuel consumed in the five flight phases:

$$J_{fuel,formation} \ = \ \sum_{i=1}^{5} [W_i(t_{0,i}) \ - \ W_i(t_{f,i})] \quad , \tag{16}$$

where W_i is the weight of the aircraft A in flight phase $i = 1, 4$, of aircraft B in flight phase $i = 2, 5$, and of aircraft A + B in flight phase i = 3.

Since the assembly of a flight formation generally requires the aircraft to make a detour, the trip times for the aircraft in a formation is typically larger than for the solo flights. This implies that, while flying formation might reduce the total fuel cost, it does not necessarily result in a lower Direct Operating Cost (DOC), which is a combination of both time and fuel cost. To be able to allow a tradeoff between time and fuel cost in formation flight, a second objective function is defined, which is essentially a weighted composite of mission time and fuel criteria:

$$\begin{aligned} J_{DOC,formation} &= \alpha \sum_{i=1}^{n_a} [t_{f,i} - t_{0,i}] \\ &+ (1-\alpha) \cdot \sum_{i=1}^{n_a} [W_i(t_{0,i}) - W_i(t_{f,i})] \quad , \end{aligned} \tag{17}$$

where α is an interpolation parameter $0 \leq \alpha \leq 1$, that can be selected to enable a shift in emphasis in the optimization process between mission time and mission fuel. In the case study conducted herein, the parameter α is systematically varied to allow the construction of a Pareto-front.

2.4 Constraints, Staging and Boundary Conditions

The multi-phase formulation for the formation flight mission is subject to a range of boundary conditions and path and control constraints. At the initial times for both flights A and B, the position coordinates (latitude, longitude), altitude, speed and heading are specified. The same state variables are also fixed at the final times. It is recalled that in the baseline scenario, neither the initial times nor the final times are specified. Only the time at which the rendezvous takes place is fixed upfront. In the subsequent delay scenario, the initial times established in the baseline scenario are then fixed as boundary conditions, whilst removing the boundary condition related to the fixed rendezvous time. In this scenario, departure delay for one of the flights can be readily introduced by merely shifting the fixed initial time of the flight concerned.

The most complex set of constraints/boundary conditions relates to the initial and final weights of flights A and B. To minimize fuel consumption, a flight should land with a minimum amount of fuel remaining in the tanks. In normal circumstances, aircraft take enough fuel on board for the planned trip, plus a reserve. This reserve usually consists of contingency fuel and a final reserve fuel. The final reserve fuel is the minimum fuel required to fly for 30–45 min at 1500 ft at holding speed above the destination or alternate aerodrome. Contingency fuel is carried to account for additional fuel consumption during the scheduled flight caused by, e.g. wind or ATM restrictions. In general this is around 5% of the trip fuel. Aircraft that join in formation will have to deviate from their optimal solo routes, resulting in an increase of the distance covered. In addition, the aircraft must also take into account that the other aircraft with which it will join, might not show-up (due to technical problems, delay, etc.). This implies that an aircraft should carry enough fuel to fly the detour distance solo. In order to enforce this, a constraint is introduced in the formulation that fixes the final weight of the aircraft. For aircraft flying a solo mission, the final weight of the aircraft is the sum of the Operational Empty Weight (OEW), the payload ($W_{payload}$) and a fuel reserve of 5% of the total fuel capacity (W_{FC}):

$$W_i(t_{f,i}) = OEW + \ W_{payload} + \ 0.05W_{FC} \quad , \quad i = A, B \tag{18}$$

With the final weight of the solo flight fixed, the unspecified initial weight of the aircraft will be established in the trajectory optimization process. Evidently, the maximum take-off weight and fuel tank capacity constraints are taken into consideration. The initial/final weights of the lead aircraft in a formation are established in a fashion similar to that for the solo flights. It is noted that the take-off weight of the lead aircraft in a formation is usually somewhat larger than that of its corresponding solo flight, due the fact that the lead aircraft needs to cover a larger distance to join the formation, but does not enjoy an induced drag reduction.

For the trailing aircraft in a formation, the assessment is more complicated. Indeed, if the trajectory of the trailing aircraft would be optimized for the final weight given by Eq. (18), the initial weight that would result from the multi-phase trajectory optimization is too low. The reason for this is that in the multiphase trajectory optimization the trailing aircraft enjoys an induced drag reduction and associated fuel burn savings, but it will not benefit from this drag reduction when it is forced to fly the same trajectory solo due to contingency circumstances. To allow for this, an additional state variable is added to the dynamic model of the trailing aircraft which represents the “pseudo” weight of the trailing aircraft. The final value of the pseudo weight of the trailing aircraft is fixed following Eq. (18). The pseudo weight of the trailing aircraft evolves over time in the same way as the actual aircraft weight, except that in the fuel flow calculations the influence of drag reduction due to formation flight is ignored (evolution as in a solo flight). An additional constraint is then introduced in the multiphase formulation that equates the “actual” initial weight of the trailing aircraft with the “pseudo” initial weight of the trailing aircraft. This constraint ensures that the final weight of the trailing aircraft is optimized for a fixed initial weight that would allow the aircraft to fly the (longer) formation route solo in case the other aircraft does not show up.

In a multi-phase formulation so-called staging conditions need to be included. Staging conditions are constraints that specify how the state at the end of a particular phase corresponds to the initial state in a subsequent phase. In the present formulation, the staging conditions are quite simple in the sense that the initial state of a particular phase is directly and fully connected to the terminal state of the preceding phase. In Fig. 1b a schematic representation is shown of the vertical flight paths covering the five phases. These five phases are connected by four staging conditions (links):

- Link 1: Connects phase 1 with phase 3
- Link 2: Connects phase 2 with phase 3
- Link 3: Connects phase 3 with phase 4
- Link 4: Connects phase 3 with phase 5

In the trajectory optimization formulation it is important to decide upfront which of the two aircraft is designated as the lead aircraft. In the present set-up, where two aircraft of the same type are considered, it is readily clear what the best choice is: the least heavy aircraft of the two is designated as the lead aircraft, as the heavy aircraft can benefit relative more from an induced drag reduction.

3 Trajectory Optimization Framework

The numerical trajectory optimization approach implemented in the optimization framework is based on a pseudo-spectral multiple-phase optimal control method. More specifically, a trajectory optimization tool named "GPOPS" has been implemented. GPOPS stands for "General Pseudospectral OPtimal Control Software" and is a MatLab based general purpose software for solving multiple-phase optimal control problems using pseudo-spectral methods [11]; it does so using the Radau pseudo-spectral method [12]. It uses the technique of collocation at Legendre-Gauss points to transcribe the trajectory optimization problem to a non-linear programming problem (NLP). GPOPS is used in combination with "IntLab" (Interval Laboratory) which is a third party automatic differentiator, and with the non-linear programming solver "SNOPT" (Sparse Nonlinear OPTimizer).

4 Case Study

In a case study several numerical experiments have been performed to investigate the benefits of formation flight and the effect of schedule delays on the formation. Unless stated otherwise, in the experiments the trajectories are optimized for minimum fuel and aircraft A is the trailing aircraft. The aircraft are assumed to operate in a standard atmosphere in a no-wind condition.

4.1 Baseline Scenario

In the baseline scenario the following two flights join in formation:

1. Aircraft A: A B744 from London to Atlanta
2. Aircraft B: A B744 from Madrid to New York City.

The resulting horizontal flight paths are presented in Fig. 2 and the vertical flight paths, velocity profiles and specific energy profiles are presented in Fig. 3a–c, respectively.

When looking at the results, one of the first things that could be noticed are the rapid transitions in the altitude and the velocity variables near the joining point. Indeed, in a formation leg the two aircraft fly at a slightly higher altitude and a slightly

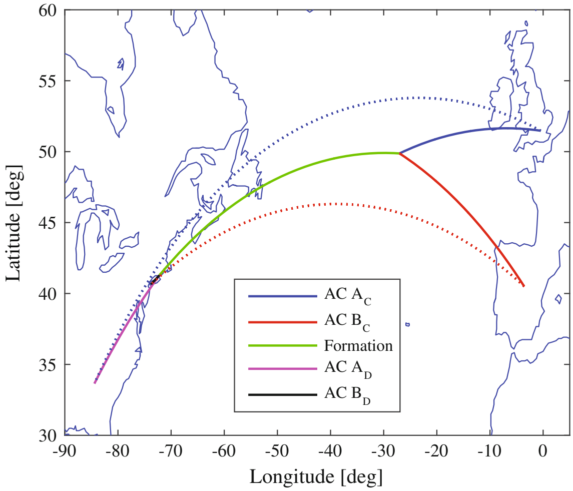


Fig. 2 Formation ground tracks for the baseline scenario

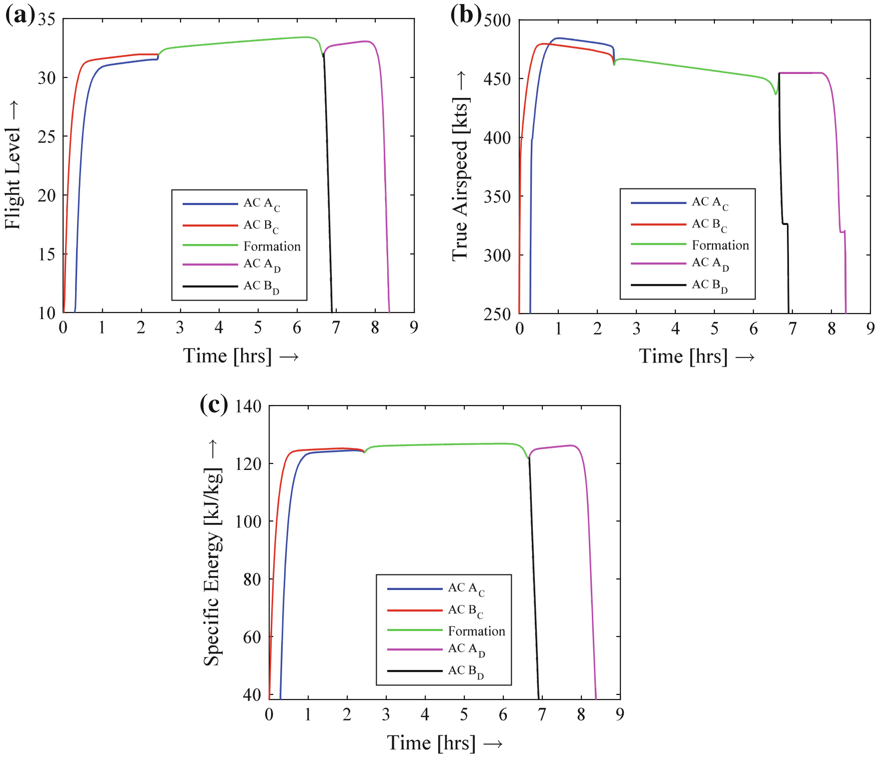


Fig. 3 Formation trajectory results for the baseline scenario (vertical plane)

lower speed than in single flight. This can be readily explained by the fact that the cruise condition for best specific range is at a somewhat lower speed relative to flying solo due to the lower induced drag of the trailing aircraft in formation. The reduction in induced drag leads to a lower total drag and a shift in the dynamic pressure at which the minimum total drag is obtained. At a given altitude, the minimum total drag - and even more relevant from a specific range perspective, the drag to speed ratio - is now achieved at a lower speed and therefore, the aircraft in formation slow down.

The trajectory behaviour at the splitting point of the formation is perhaps even more surprising. In particular, it can be observed that the formation actually descents (both in terms of altitude and specific energy) before it splits-up. It is noted that the leading aircraft reduces its engine control setting η in this early descent to about 70%, but the lead aircraft requires a somewhat lower thrust setting as it is subject to a relatively lower drag. After splitting, aircraft A (the previously trailing aircraft) actually starts to climb again to the optimal cruise conditions for solo flight (and thus to a higher cruise speed than if formation flight). Evidently this (energy) climb requires additional thrust, and thus a higher fuel flow. Apparently, the additional fuel burn resulting from this late climb is offset by the fuel reduction in the early descent and flying in formation for a longer period of time. These peculiar “dents” in the energy histories occurred in all conducted numerical experiments to some extent, including those featuring different aircraft types. Numerical experiments in which the occurrence of these dents is precluded bear out that the mission fuel is indeed (modestly) increased in those cases. However, from an operational perspective, these constrained trajectories might actually still be preferred.

The main results for the baseline case are summarized in Table 1. Although the fuel consumption of the lead aircraft (B) increases compared to the solo flight, the overall fuel consumption in the formation mission decreases with approximately 1.8%.

4.2 Departure Delay Scenario

In the baseline scenario the lowest total fuel consumption is obtained when aircraft A departs 1,008 s later than aircraft B. So, for this set of flights the optimal initial

Table 1 Results for the baseline scenario

Results	Solo flight			Formation flight		
	Aircraft A	Aircraft B	Total	Aircraft A	Aircraft B	Total
Fuel (kg)	79,093	66,239	145,332	75,530	67,238	142,768
Time (h)	7.92	6.80	14.71	8.09	6.90	14.99
Distance (km)	6,760	5,760	12,521	6,825	5,835	12,660

Table 2 Results for the delay scenario

Results	Solo flight			Formation flight		
	Aircraft A	Aircraft B	Total	Aircraft A	Aircraft B	Total
Fuel (kg)	79,093	66,239	145,332	76,696	66,867	143,563
Time (h)	7.92	6.80	14.71	8.37	6.67	15.06
Distance (km)	6,760	5,760	12,521	6,850	5,790	12,640

times are $t_{0,A} = 1,008\,\text{s}$ (0.28 h) and $t_{0,B} = 0$. In the departure delay scenario, the departure times of both aircraft are fixed (whilst removing the fixed rendezvous time constraint), but now a departure delay of 30 min is assigned to aircraft B, relative to the baseline scenario. Since in this delay scenario, aircraft A is the first one to depart, its departure time $t_{0,A}$ is set to zero. Hence in this scenario we consider:

1. Aircraft A: A B744 from London to Atlanta with initial time $t_{0,A} = 0\,\text{s}$
2. Aircraft B: A B744 from Madrid to New York with initial time $t_{0,B} = 792\,\text{s}$.

The formation mission results for the departure delay scenario are summarized in Table 2. In Table 2 the results for the corresponding solo flights are presented as well to allow a comparison.

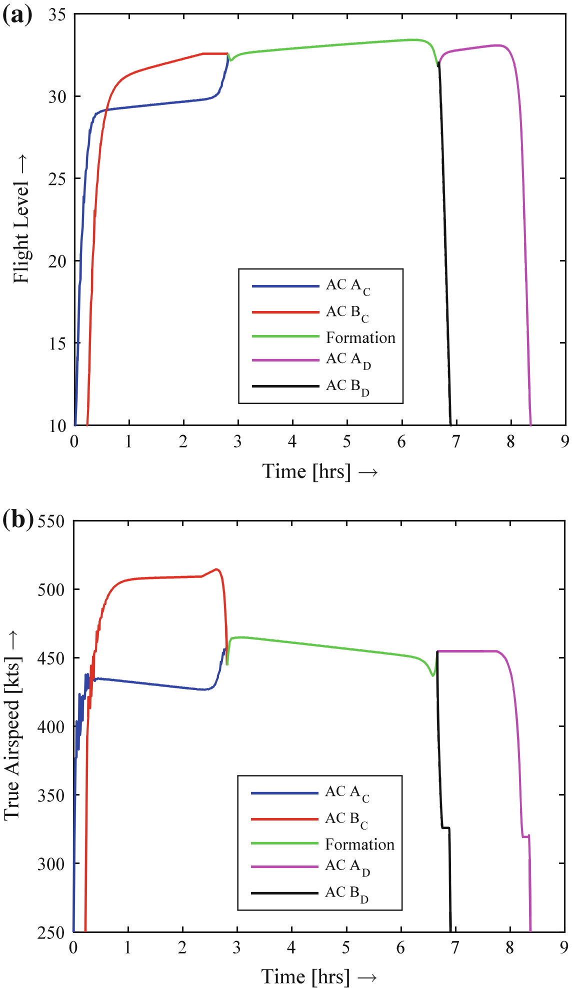
The results of the delay scenario reveal that the fuel burn for the formation mission with one delayed aircraft is still (about 1.2%) lower than that for the corresponding solo flights, while the total flight time increased only marginally compared to the no-delay (baseline) case. However, while aircraft B suffered the delay, the consequences of the delay recovery are particularly felt by aircraft A, with its flight time increasing appreciably compared to the flight time of its corresponding solo flight. The shorter flight time of aircraft B is partly due to the increase in velocity (see Fig. 4) and partly due to the route change (Fig. 5). Aircraft A has to absorb a time delay and it does so by flying at the maximum endurance speed during a large part of the climb and by stretching its flight path in such a way that the flight path of its formation partner (aircraft B) is reduced.

Figure 6 presents the resulting rendezvous points of several formations featuring one delayed aircraft. More specifically, Fig. 6 shows the displacement of the rendezvous point location when aircraft A or B are delayed by, respectively, 10, 15 or 30 min. It is readily clear that when one of the aircraft is delayed, the rendezvous point is displaced in both longitudinal and lateral direction.

4.3 Direct Operating Cost Scenario

This scenario has been conceived to demonstrate what the effect on the performance of the aircraft in the formation is when in the trajectory optimization process the emphasis is gradually shifted from optimizing for fuel consumption to optimizing

Fig. 4 Formation trajectory results for the delay scenario (vertical plane)



for flight time. A parametric study has been conducted in which the interpolation parameter α in the performance index (17) is systematically varied between zero and one. Since $\alpha = 1$ corresponds to a cost function that purely considers flight time, it is left out of consideration. In this scenario, the same set-up is employed as in the baseline case, with the exception of the different performance index. In Fig. 7 the results of this parametric investigation are summarized.

The blue dots shown in Fig. 7 represent the results for the formation mission, where the dot in the upper left corner represents $\alpha = 0$ (optimized for minimum total fuel consumption). When the values of α is increased (in steps of 0.1), Fig. 7 shows that the total flight time decreases while the total fuel burn increases. The combined results for the baseline (minimum fuel burn) solo flights are shown in Fig. 7 as well

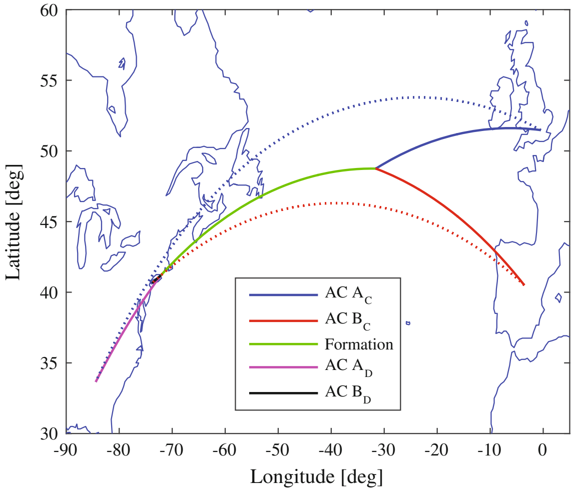


Fig. 5 Formation ground tracks for the delay scenario

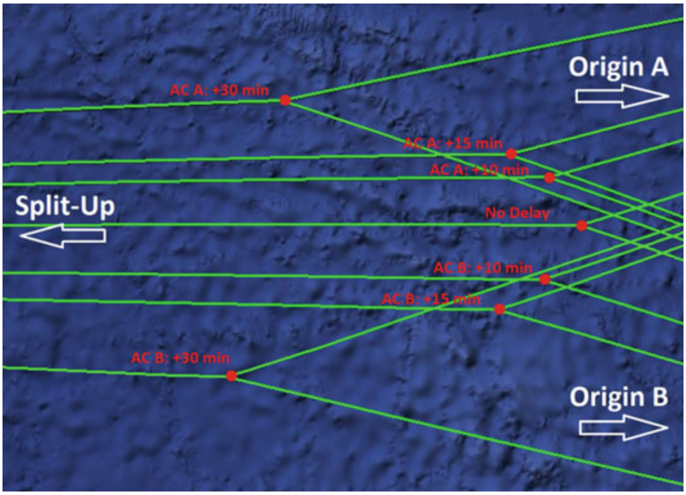


Fig. 6 Shift of rendezvous point when aircraft A or B is delayed relative to baseline scenario

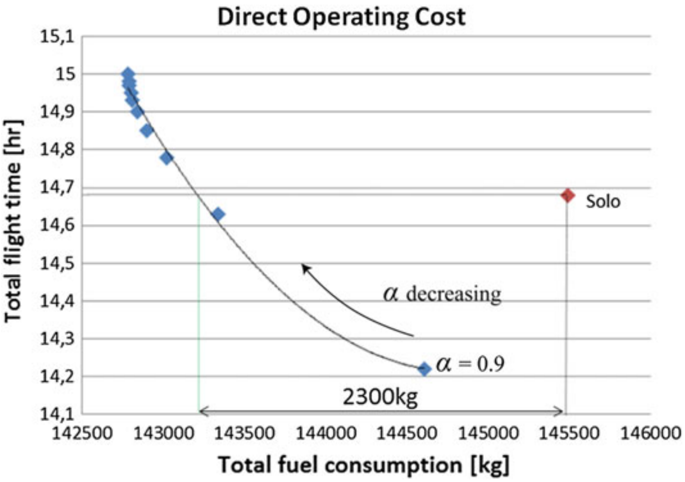


Fig. 7 Flight formation optimal fuel-time performances for various values of the weighting parameter α

(represented by the red dot). It can be readily observed that the corresponding solo flights consume significantly more fuel compared to flying in formation, for any value of the parameter α in the considered range. When the value of the parameter α is set to $\approx$0.78 in Eq. (17), the total flight time of the formation equals the total flight time of the corresponding solo flight, while the total fuel consumption is 2,300 kg (1.58%) lower. This shows that, compared to flying solo, formation flight can lead to significant reduction in fuel burn, even when trip time is preserved. It should be kept in mind though, that this analysis is for this particular set of flights and aircraft type combination. For other origin and destination pairs or aircraft type combinations, the results might deviate from the results obtained in this scenario.

5 Conclusions

This paper has shown the possibility of optimizing the trajectories of aircraft that join in formation. The developed multiple-phase optimization tool is able to optimize trajectories of two or more aircraft for minimum total fuel consumption or Direct Operating Cost. Several numerical experiments have been performed in a case study to investigate the benefits of formation flight and the behaviour of the aircraft when joining in formation.

The results from the experiments show that formation flight can lead to significant reductions in fuel consumption, while not increasing the trip time. The results show that when aircraft fly in formation, the optimal airspeed decreases and the optimal altitude increases compared to the corresponding optimal solo flights.

Furthermore, the effects of departure delay on formations have been examined. This analysis revealed that in the cases that a departure delay was assigned to one of the participating formation partners, the location of the rendezvous point was shifted. In contrast to what has been often assumed in the literature, the rendezvous point location will not merely shift along the original (non-delayed) optimal route, but it will also deviate laterally (in the direction of the solo route of the delayed aircraft) such as to shorten the route of the delayed aircraft that seeks to join the formation.

Although in this paper only the optimization of two-aircraft formations was addressed, the developed optimization framework is capable of optimizing the trajectories of three-aircraft formations as well. For future research, it might be of interest to investigate the benefits of larger formations. Furthermore, the optimization in this research mainly focused on minimizing the total fuel burn or DOC, assuming a standard atmosphere and a no-wind condition. In future research, the benefits and characteristics of formation flight in real-world (weather/wind) conditions need to be explored. Finally, as optimizing the trajectories of more than two aircraft with the developed optimization framework proved to be very time-consuming, future research needs to be aimed at speeding up the computational process to render the optimization approach tractable in the context of bi-level planning of flight formations on a network scale.

References

1. Pahle J, Berger D, Venti M, Duggan C, Faber J, Cardinal K (2012) An initial flight investigation of formation flight for drag reduction on the C-17 aircraft. In: AIAA atmospheric flight mechanics conference, 13–16 August 2012, Minneapolis, U.S.A., p 13
2. Ning SA, Flanzer TC, Kroo IM (2011) Aerodynamic performance of extended formation flight. J Aircr 48(3):855–865
3. Veldhuis L, Voskuijl M, Fransen B (2013) Formation flight - fine-tuning of theoretical performance prediction. In: AIAA aerospace sciences meeting, 7–10 January 2013, Grapevine, Texas, U.S.A., p 17
4. Ning SA, Kroo I, Aftosmis MJ, Nemec M, Kless JE (2014) Extended formation flight at transonic speeds. J Aircr 51(5):1501–1510
5. Bower GC, Flanzer TC, Kroo IM (2009) Formation geometries and route optimization for commercial formation flight. In: AIAA applied aerodynamics conference, 22–25 June 2009, San Antonio, Texas. U.S.A., p 18
6. Kent T, Richards A (2015) Analytic approach to optimal routing for commercial formation flight. J Guid Control Dyn 38(10):1872–1884
7. Xu J, Ning SA, Bower G, Kroo I (2014) Aircraft route optimization for formation flight. J Aircr 51(2):490–501
8. Xue M, Hornby GS (2012) An analysis of the potential savings from using formation flight in the NAS. In: AIAA guidance, navigation, and control conference, 13–16 August 2012, Minneapolis, U.S.A, p 12
9. Visser HG (2010) Airplane performance optimisation. In: Encyclopaedia of aerospace engineering. Wiley
10. Visser HG (2008) Environmentally optimized resolutions of in-trail separation conflicts for arrival flights. J Aircr 45(4):1198–1205

11. Rao AV, Benson DA, Darby CL, Patterson M, Francolin C, Sanders I, Huntington GT (2010) Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method. ACM Trans Math Softw 37(2), Article 22., 22.1–22.39
12. Francolin C, Rao AV (2012) Direct trajectory optimization and costate estimation of state inequality path-constrained optimal control problems using a radau pseudospectral method. In: AIAA guidance, navigation, and control conference, 13–16 August 2012, Minneapolis, U.S.A, p 11

Optimal Scheduling Algorithm for Air Traffic Point Merge System Using MILP

Youkyung Hong, Somang Lee, Keumjin Lee and Youdan Kim

1 Introduction

Terminal Maneuvering Area (TMA), the area in the vicinity of an airport, is a highly congested airspace where incoming arrivals and outgoing departures are concentrated. To improve the efficiency and safety of flight operations in TMA, Standard Instrument Departure (SID) and Standard Terminal Arrival Route (STAR) are defined. Without the interventions by air traffic controllers, aircraft should follow those predefined routes with aid of the flight management system. However, during peak hours, air traffic controllers often provide manual instructions of heading angle and/or speed changes to maintain proper separations between aircraft arriving into the same runway.

Recently, Point Merge System (PMS), which is a new STAR design for merging inbound traffics, has been proposed by Eurocontrol [1]. The PMS system allows air traffic controllers a more systematic way to sequence and schedule arrival aircraft by providing the predefined procedure, i.e., sequencing leg, to absorb necessary delays. As shown in Fig. 1, in PMS, incoming flights can fly with economical speed along the sequencing leg until the appropriate spacing to a preceding aircraft at the merge point is guaranteed. When a direct-to instruction is given, the flights can make a continuous

Y. Hong (✉) · S. Lee · Y. Kim
Mechanical and Aerospace Engineering, Seoul National University,
Seoul, Republic of Korea
e-mail: youkyh1@snu.ac.kr

S. Lee
e-mail: cjsomang@snu.ac.kr

Y. Kim
e-mail: ydkim@snu.ac.kr

K. Lee
College of Aviation and Management, Korea Aerospace University,
Goyang, Republic of Korea
e-mail: keumjin.lee@kau.ac.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_22

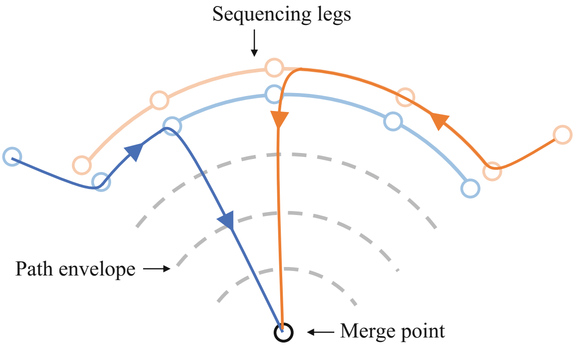


Fig. 1 Typical structure of point merge system for the arrival traffic flows in the opposite direction

descent to the merge point, and therefore minimize the environmental impacts and fuel consumption. Furthermore, the safety level in TMA can be improved because the tactical vectoring is replaced with procedural operations, and the workload of human air traffic controllers can be also reduced.

Several studies have been performed for PMS. First, by using data collection experiments, the performance of PMS was compared with the current working method of managing arrival traffic such as heading instructions [2]. In a follow-up study, the performance of PMS was assessed in more complex environment, and the application of continuous descent from further upstream airspace was investigated [3]. A method for designing fast-time models for performance comparison between PMS and conventional vectoring was provided [4]. PMS was applied to Istanbul international airport, and then the pre-implementation studies through real time simulation was described [5]. However, most previous studies on PMS have only focused on performance validation. Much less attention has been given to optimize actual flight operations conducted within PMS.

In this study, optimal scheduling for PMS is investigated using a Mixed Integer Linear Programming (MILP). A number of different MILP formulations have been proposed to solve the optimal planning and scheduling problems for the aircraft movements on the airport surface [6–8]. In our MILP formulation, decision variables and constraint equations are derived by combining the discrete routing and continuous time scheduling problems, which is similar to those in the previous study for metroplex traffic [9]. However, by utilizing the characteristics of PMS, the details of the constraint equations become more relevant, and therefore the number of variables can be significantly reduced. In real operations, the proposed optimization algorithm may suggest the direct-to instruction at the optimum time, whereas it is determined intuitively by controller in the current air traffic control environment. In addition, the arrival time of flights can be easily estimated when advisories are made based on the proposed optimal scheduling algorithm. Note that the arrival time estimation is one of the challenging issues in air traffic management.

The remainder of this study is organized as follows. In Sect. 2, the concept of PMS is briefly described. Section 3 shows the proposed optimal scheduling algorithm for

PMS using MILP. In Sect. 4, numerical simulations are performed to demonstrate the performance of the proposed method. Section 5 states the conclusions of this study and provides some directions for future research.

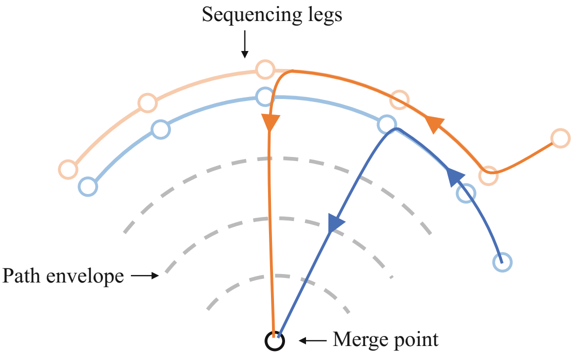
2 Point Merge System

This section briefly introduces a PMS. PMS is a new technique for integrating arrival air traffic flows into airport developed by Eurocontrol in 2006. Following the first PMS implementation at Oslo in 2011, PMS is now operational in several international airports around the world such as Seoul (2012), Paris (2013), Hannover (2014), and so on [1].

Figure 1 shows a typical structure of PMS which is characterized by the two features: the merge point and sequencing legs. As shown in Fig. 1, the merge point is used as the final point for arrival traffic integration. On the other hand, the sequencing legs which are equidistant from the merge point are used to delay the inbound traffics to build a schedule with an adequate separations. Note that different sequencing legs are vertically separated by 1,000 ft and therefore aircraft from one direction can be free from the conflict with aircraft from the other direction. Incoming flights fly along the sequencing legs at a constant speed, and then they start continuous descent approach towards the merge point when a direct-to instruction is given by air traffic controllers. Note that the direct-to instruction can be issued when the appropriate spacing is achieved with the preceding flight in the sequence.

Depending on the airspace and runway conditions of airport, various configurations of PMS can be applied. Figure 2 shows the type of PMS which is used in Jeju international airport in the Republic of Korea. As shown in Figs. 1 and 2, unlike the typical PMS, arrival flows in the PMS of Jeju international airport are in the same direction. In this study, our attention is restricted to the PMS type used in Jeju international airport. The ultimate objective of this study is to develop a decision support

Fig. 2 Point merge system for the arrival traffic flows in the same direction



tool for air traffic controllers and help them to assess the impacts of their decisions on managing arrivals in the PMS of Jeju international airport. Note that the proposed approach could be extended to the different types of PMS.

3 Problem Formulation

3.1 Performance Index

The performance index to design the optimal scheduling algorithm for PMS can be defined as follows:

$$J = \sum_{f=1}^{F} \sum_{r=1}^{R} A_{f,r} \ \left(T_{f,r,p_I} + T_{f,r,p_F}\right) \tag{1}$$

where $A_{f,r}$ is a binary variable that takes value 1 when flight f is assigned to the route r and zero otherwise, and $T_{f,r,p}$ is a continuous variable representing the time that flight f reaches the point p on route r. Also, p_I and p_F are the first and final merge points in PMS, respectively, and F and R are the number of flights and routes, respectively. In (1), the second term is included to minimize the total time required for all flights to reach the final merge point. However, if the performance index is defined only using the second term, then the entering time of each flight into PMS may be delayed. In other words, a flight waits outside PMS until a suitable spacing and sequencing to a preceding flight is obtained, and then approaches the final merge point right after passing the first point of PMS. To encourage flight to stay in the sequencing leg, the first term in (1) is introduced.

Note that (1) is a nonlinear performance index where two variables are multiplied. To be transformed into a linear form, following auxiliary variable $\delta^T_{f,r,p} = A_{f,r} T_{f,r,p}$ is used [10]:

$$J = \sum_{f=1}^{F} \sum_{r=1}^{R} \left(\delta^T_{f,r,p_I} + \delta^T_{f,r,p_F}\right) \tag{2}$$

subject to

$$\begin{aligned}
\delta^T_{f,r,p_I} &\geq T_{f,r,p_I} - M\left(1 - A_{f,r}\right) \\
\delta^T_{f,r,p_I} &\leq T_{f,r,p_I} + M\left(1 - A_{f,r}\right) \\
\delta^T_{f,r,p_F} &\geq T_{f,r,p_F} - M\left(1 - A_{f,r}\right) \\
\delta^T_{f,r,p_F} &\leq T_{f,r,p_F} + M\left(1 - A_{f,r}\right)
\end{aligned}$$

where M is an arbitrarily large number. Note that (1) is equivalent to (2).

3.2 Constraints

Six constraint equations are required for the optimal scheduling in PMS based on MILP. The first constraint is that each flight is assigned to only one route, which can be obtained as follows.

$$\sum_{r=1}^{R} A_{f,r} = 1 \qquad \forall f \tag{3}$$

The second and third constraints are to make the time range of each flight at the first and final merge points in PMS, respectively. The second constraint can be formulated as follows:

$$A_{f,r}\left(T_{f,r,p_I} - T^E_{f,r,p_I}\right) \geq 0 \qquad \forall f, r \tag{4}$$

$$A_{f,r}\left(T_{f,r,p_I} - T^L_{f,r,p_I}\right) \leq 0 \qquad \forall f, r \tag{5}$$

where T^E_{f,r,p_I} and T^L_{f,r,p_I} are the earliest and last time at the first point. Similarly, the third constraint can be obtained as follows:

$$A_{f,r}\left(T_{f,r,p_F} - T^E_{f,r,p_F}\right) \geq 0 \qquad \forall f, r \tag{6}$$

$$A_{f,r}\left(T_{f,r,p_F} - T^L_{f,r,p_F}\right) \leq 0 \qquad \forall f, r \tag{7}$$

where T^E_{f,r,p_F} and T^L_{f,r,p_F} are the earliest and last time at the final merge point. Note that the second and third constraints should be transformed into linear inequations using the auxiliary variable $\delta^T_{f,r,p}$. For example, the second constraints (4) and (5) can be transformed into following equations.

$$\delta^T_{f,r,p_I} - A_{f,r}T^E_{f,r,p_I} \geq 0 \qquad \forall f, r \tag{8}$$

$$\delta^T_{f,r,p_I} - A_{f,r}T^L_{f,r,p_I} \leq 0 \qquad \forall f, r \tag{9}$$

The fourth constraint is to determine the sequence of flight at the first and final merge points. Contrary to the discrete routing and continuous time scheduling problem for metroplex traffic [9], the scheduling and sequencing points are only considered as the first and final merge points in this study. Because the schedule and sequence of flight are not determined for all points in PMS, the number of variables related to the computational efforts are not increased much. The fourth constraint can be formulated as follows:

$$S_{f,f',r,r',p_I} + S_{f',f,r',r,p_I} = A_{f,r}A_{f',r'} \qquad \forall f \neq f', r = r' \tag{10}$$

$$S_{f,f',r,r',p_F} + S_{f',f,r',r,p_F} = A_{f,r}A_{f',r'} \qquad \forall f \neq f', r, r' \tag{11}$$

where $S_{f,f',r,r',p}$ is a binary variable that takes value 1 when flight f on route r is prior to flight f' on route r' at shared point p. In PMS, flights on different routes enter the first point at different altitudes. In addition, because different routes (i.e., different sequencing legs) are vertically separated, the sequence at the first point is only determined among flights on same route. By considering these characteristics of PMS, r' is equal to r in (10). However, at the final merge point, the sequence of flight should be determined whatever their assigned routes are. Note that (10) and (11) should be transformed into a linear form by using an auxiliary variable $\delta^A_{f,f',r,r'} = A_{f,r}A_{f',r'}$. For example, (11) is equivalent to the following equation.

$$S_{f,f',r,r',p_F} + S_{f',f,r',r,p_F} = \delta^A_{f,f',r,r'} \quad \forall f \neq f', r, r' \tag{12}$$

$$-A_{f,r} + \delta^A_{f,f'r,r'} \leq 0 \quad \forall f \neq f', r, r' \tag{13}$$

$$-A_{f',r'} + \delta^A_{f,f'r,r'} \leq 0 \quad \forall f \neq f', r, r' \tag{14}$$

$$A_{f,r} + A_{f',r'} - \delta^A_{f,f'r,r'} \leq 1 \quad \forall f \neq f', r, r' \tag{15}$$

The fifth constraint is to make a time separation for successive flight at the first and final merge points. It can be formulated as follows:

$$SEP_{f,f',p_I} \leq A_{f',r'}T_{f',r',p_I} - A_{f,r}T_{f,r,p_I} + M(1 - S_{f,f',r,r',p_I}) \quad \forall f \neq f', r, r' \tag{16}$$

$$SEP_{f,f',p_F} \leq A_{f',r'}T_{f',r',p_F} - A_{f,r}T_{f,r,p_F} + M(1 - S_{f,f',r,r',p_F}) \quad \forall f \neq f', r, r' \tag{17}$$

where SEP_{f,f',p_I} and SEP_{f,f',p_F} are the minimum separation time required at the first and final merge points. Again, (16) and (17) should be transformed into a linear form using the auxiliary variable $\delta^T_{f,r,p}$. The last constraint is to calculate the transit time of each flight in PMS. It can be represented as follows:

$$T_{f,r,p_F} = T_{f,r,p_I} + \left(P_{f,r} - 2\right) \Delta T_L + \Delta T_R \qquad \forall f, r \tag{18}$$

where $P_{f,r}$ is the number of points that flight f on route r passes through in PMS. Because $P_{f,r}$ includes the first and final merge points, $P_{f,r}$ is greater than or equal to two. Also, ΔT_L and ΔT_R are the transit time between successive points in the sequencing legs and the flight time for final descent to the merge point from the legs, respectively.

3.3 Model Summary

The performance index and constraint equations to design the optimal scheduling for PMS can be summarised as follows.

$$\min \ J = \sum_{f=1}^{F} \sum_{r=1}^{R} \left(\delta^T_{f,r,p_I} + \delta^T_{f,r,p_F} \right) \tag{19}$$

subject to

$$\begin{aligned}
\delta^T_{f,r,p_I} &\geq T_{f,r,p_I} - M\left(1 - A_{f,r}\right) && \forall f, r \\
\delta^T_{f,r,p_I} &\leq T_{f,r,p_I} + M\left(1 - A_{f,r}\right) && \forall f, r \\
\delta^T_{f,r,p_F} &\geq T_{f,r,p_F} - M\left(1 - A_{f,r}\right) && \forall f, r \\
\delta^T_{f,r,p_F} &\leq T_{f,r,p_F} + M\left(1 - A_{f,r}\right) && \forall f, r
\end{aligned}$$

$$\sum_{r=1}^{R} A_{f,r} = 1 \qquad \forall f$$

$$\begin{aligned}
\delta^T_{f,r,p_I} - A_{f,r} T^E_{f,r,p_I} &\geq 0 && \forall f, r \\
\delta^T_{f,r,p_I} - A_{f,r} T^L_{f,r,p_I} &\leq 0 && \forall f, r \\
\delta^T_{f,r,p_F} - A_{f,r} T^E_{f,r,p_F} &\geq 0 && \forall f, r \\
\delta^T_{f,r,p_F} - A_{f,r} T^L_{f,r,p_F} &\leq 0 && \forall f, r
\end{aligned}$$

$$\begin{aligned}
S_{f,f',r,r',p_I} + S_{f',f,r',r,p_I} &= \delta^A_{f,f',r,r'} && \forall f \neq f', r = r' \\
S_{f,f',r,r',p_F} + S_{f',f,r',r,p_F} &= \delta^A_{f,f',r,r'} && \forall f \neq f', r, r' \\
-A_{f,r} + \delta^A_{f,f'r,r'} &\leq 0 && \forall f \neq f', r, r' \\
-A_{f',r'} + \delta^A_{f,f'r,r'} &\leq 0 && \forall f \neq f', r, r' \\
A_{f,r} + A_{f',r'} - \delta^A_{f,f'r,r'} &\leq 1 && \forall f \neq f', r, r'
\end{aligned}$$

$$\begin{aligned}
SEP_{f,f',p_I} &\leq A_{f',r'} T_{f',r',p_I} - A_{f,r} T_{f,r,p_I} + M(1 - S_{f,f',r,r',p_I}) && \forall f \neq f', r, r' \\
SEP_{f,f',p_F} &\leq A_{f',r'} T_{f',r',p_F} - A_{f,r} T_{f,r,p_F} + M(1 - S_{f,f',r,r',p_F}) && \forall f \neq f', r, r'
\end{aligned}$$

$$T_{f,r,p_F} = T_{f,r,p_I} + \left(P_{f,r} - 2\right) \Delta T_L + \Delta T_R \qquad \forall f, r$$

4 Numerical Simulation

Numerical simulations are performed to demonstrate the performance of the proposed optimal scheduling algorithm for PMS using MILP. In the simulations, to mimic the real PMS, all flights are assumed to fly at a constant speed of 210 knots. By considering the structure of PMS in Jeju international airport in the Republic of Korea, the length of arc between successive points in the sequencing legs and the radius of arc are set to 5 and 17 nm, respectively. Thus, the transit time between successive points $\varDelta T_L$ and $\varDelta T_R$ in (18) are determined as 1.43 and 4.86 min, respectively. In addition, the minimum separation time requirements SEP_{f,f',p_I} and SEP_{f,f',p_F} in (16) and (17) are set to 5 and 1 min, respectively. All numerical computations are performed using a desktop PC with an Intel Core (3.60 GHz) processor. The simulation results are obtained using CPLEX, a standard optimization software package for MILP [11].

In the first simulation, PMS composed of six points is considered; one point is the final merge point and the other points are in the sequencing legs. Also, it is considered that four arrival flights exist on two different routes; route 1 is assigned for one flight, and route 2 is assigned for the other flights. Figure 3 shows the trajectory of each flight, where each triangle denotes the position and orientation of the flight every

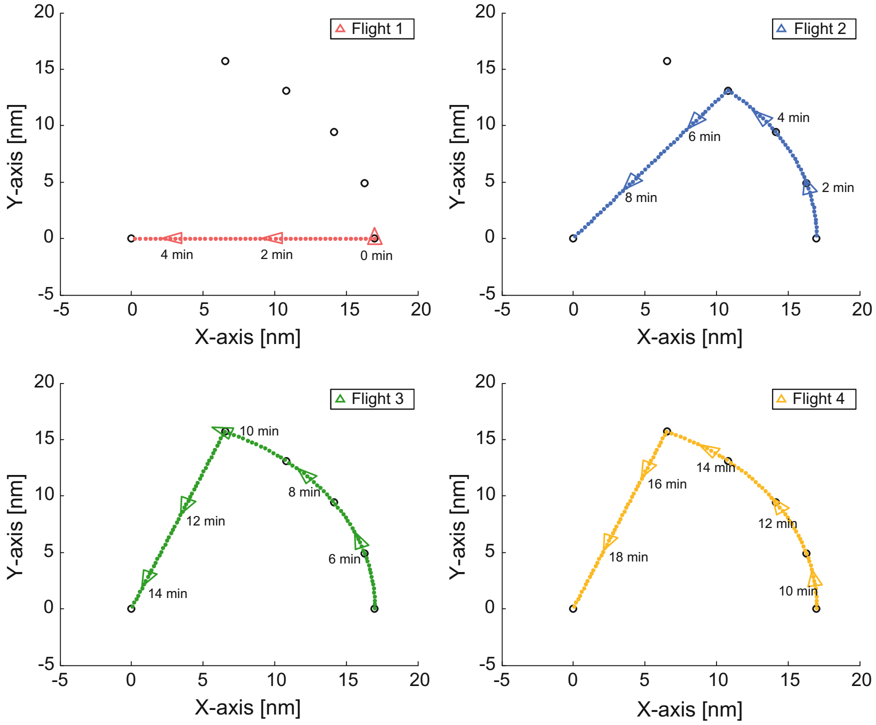


Fig. 3 Trajectory of each flight when the point merge system is composed of 6 points

Table 1 Results when the point merge system is composed of 6 points

Route	Flight	$P_{f,r}$	Point	$T_{f,r,p}$
1	1	2	p_I	0.00
			p_F	4.86
2	2	5	p_I	0.71
			p_F	9.86
	3	6	p_I	4.28
			p_F	14.86
	4	6	p_I	9.28
			p_F	19.86

2 min until each flight reaches the final merge point of PMS. The simulation results are summarized in Table 1. As shown in Fig. 3 and Table 1, flights 3 and 4 are held before entering PMS. The reason is that the necessary delays for the flights 3 and 4 cannot be achieved, although they pass through all six points in PMS. In addition, the performance index is 63.71 min, which is the summation of the total time at the first point(=14.27 min) and the final merge point(=49.44 min). Even though the separation constraints at the final merge point can be satisfied by using the proposed algorithm, it seems that PMS points do not have enough capacity to provide efficient flight for four aircraft simultaneously.

The number of points constituting PMS is increased from six to twelve in the second simulation. The trajectory of each flight is shown in Fig. 4, and the simulation results are summarized in Table 2. As shown in Fig. 4 and Table 2, the entering time of each flight is hastened; therefore, the performance index is reduced to 52.89 min which is the summation of the total time at the first point(=3.14 min) and the final merge point(=49.75 min). A comparison of the two results reveals that, depending on the traffic congestion, the number of points in PMS needs to be adjusted to reduce the total transit time required for all flights.

In the third and forth simulations, it is additionally verified that the entering or final merging time of a particular flight can be predetermined by using the proposed algorithm. Figure 5 and Table 3 show the third simulation results where flight 1 is managed to enter PMS at 3.00 min. The flight 1 passes the first point at 3.00 min instead of 0.00 min determined in the previous simulations, and then it is managed to stay in the sequencing legs by passing through six points. As the sequencing and scheduling of the flight 1 are changed, those of the other flights should be rearranged. Therefore, the performance index is slightly increased to 56.03 min, which is the summation of the total time at the first point(=6.14 min) and the final merge point(=49.89 min). Figure 6 and Table 4 show the fourth simulation results where flight 1 is managed to pass the final merge point in PMS at 10.00 min. As shown in Fig. 6 and Table 4, the flight 1 enters PMS at 0.85 s and then passes through four points in the sequencing legs in order to pass the final merge point at 10.00 min. These experiments reveal that the entering or final merging time can be managed at the request of human controller. However, because additional constraints are included

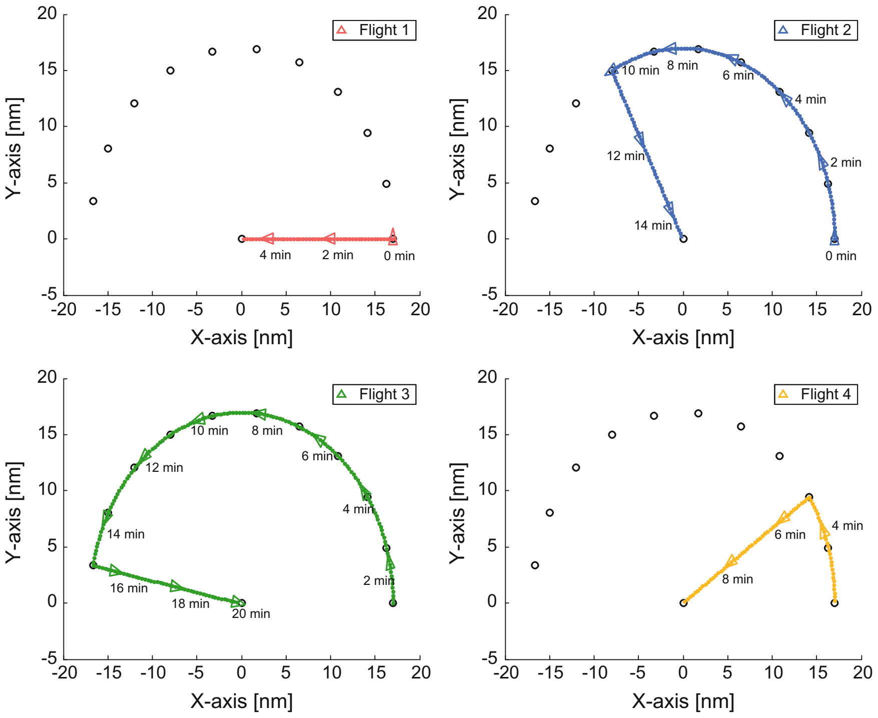


Fig. 4 Trajectory of each flight when the point merge system is composed of 12 points

Table 2 Results when the point merge system is composed of 12 points

Route	Flight	$P_{f,r}$	Point	$T_{f,r,p}$
1	1	2	p_I	0.00
			p_F	4.86
2	2	9	p_I	0.00
			p_F	14.87
	3	12	p_I	1.00
			p_F	20.16
	4	4	p_I	2.14
			p_F	9.86

to achieve the special requirement, the total elapsed time in PMS of all flights is increased.

Note that the number of variables required to solve the simulations is 192, where the number of flight and routes are set to 4 and 2, respectively. In the problem formulation, the schedule and sequence of flight are determined at the first and final points in PMS, not all points. Because the number of point constituting PMS is not related to computational effort, various configurations of PMS can be considered

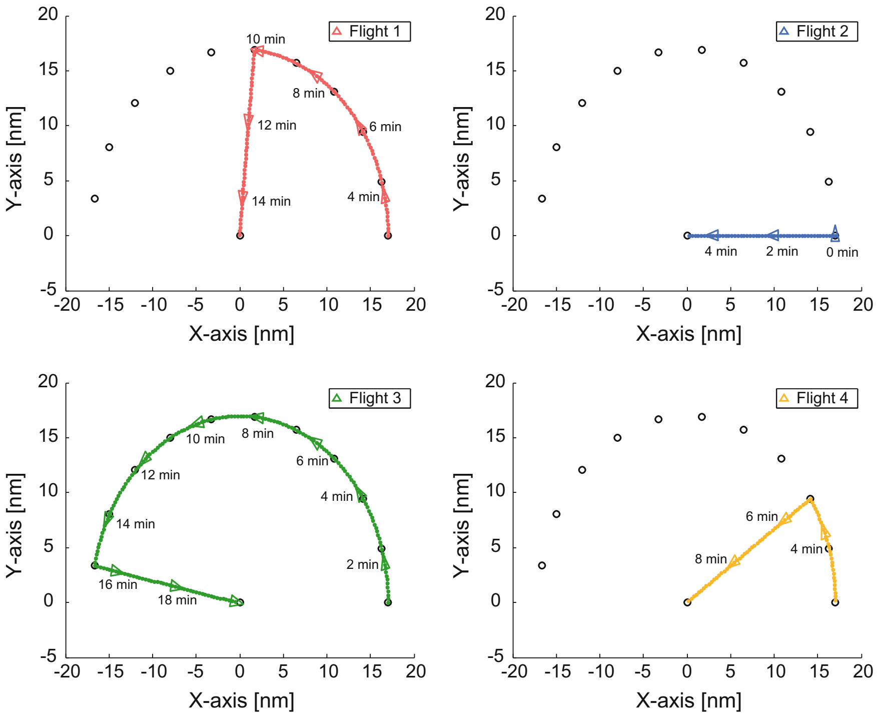


Fig. 5 Trajectory of each flight when flight 1 is managed to enter the point merge system at 3.00 min

Table 3 Results when flight 1 is managed to enter the point merge system at 3.00 min

Route	Flight	$P_{f,r}$	Point	$T_{f,r,p}$
1	1	7	p_I	3.00
			p_F	15.01
2	2	2	p_I	0.00
			p_F	4.86
	3	12	p_I	1.00
			p_F	20.16
	4	4	p_I	2.14
			p_F	9.86

to integrate arrival traffic flows effectively. Furthermore, the CPU time is less than 0.35 s in all numerical computations.

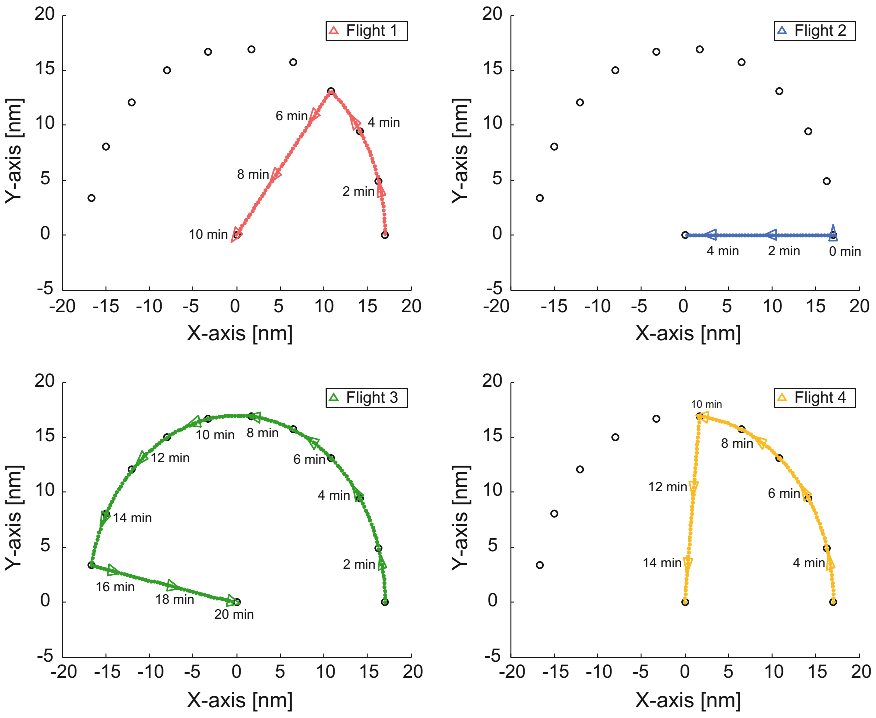


Fig. 6 Trajectory of each flight when flight 1 is managed to pass the final merge point at 10.00 min

Table 4 Results when flight 1 is managed to pass the final merge point at 10.00 min

Route	Flight	$P_{f,r}$	Point	$T_{f,r,p}$
1	1	5	p_I	0.85
			p_F	10.00
2	2	2	p_I	0.00
			p_F	4.86
	3	12	p_I	1.00
			p_F	20.16
	4	7	p_I	2.99
			p_F	15.00

5 Conclusion

In this study, the optimal scheduling algorithm for PMS (Point Merge System) was proposed based on MILP (Mixed Integer Linear Programming). In the proposed algorithm, the performance index was designed to minimize the total transit time of all flights in PMS, and six constraint equations were derived by considering the characteristics of PMS. During this process, nonlinear equations were transformed into linear equations using the auxiliary variables to solve the optimal problem using MILP. In addition, to reduce the number of variables, the first and final points in PMS were only considered as the scheduling points. Therefore, the computational efforts are not significant even if a lot of points are included in PMS. Various numerical simulations have been performed to demonstrate the performance of the proposed method. As a result, it can be concluded that the proposed algorithm can be implemented in real time. Furthermore, the optimal solution can be adjusted reflecting controller's intention.

For future works, the discrete PMS considered in this study can be extended to the continuous PMS. In real operation, a flight can get the direct-to instruction to the final merge point even though it is in the middle of successive points in the sequencing legs. It seems that more practical solutions can be obtained by using the continuous PMS. Also, the optimization problem to determine the schedule and sequence of flight in PMS might require a nonlinear optimization technique such as population-based genetic or particle swarm optimization algorithm to deal with nonlinearity. In the current MILP formulation, it is considered that each aircraft is flying at the same speed. However, speed variation may exist in real operation. To consider a variety of speed classes, additional sequencing legs with scheduling points spaced differently might be included. However, if the number of flight, route and point are increased, the number of auxiliary variables to handle the nonlinearity increases. Therefore, more tractable optimization schemes are required to solve such a complex problem in real time.

Acknowledgements This work was supported by the Development of Integrated Departure/Arrival Management Technologies Project funded by the Ministry of Land, Infrastructure and Transport.

References

1. Eurocontrol (2006) Point merge: Improving and harmonising arrival operations with existing technology. https://www.eurocontrol.int/services/point-merge-concept
2. Boursier L, Favennec B, Hoffman E, Trzmiel A, Vergne F, Zeghal K (2007) Merging arrival flows without heading instructions. In: 7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain
3. Favnnec B, Hoffman E, Trzmiel A, Vergne F, Zeghal K (2009) The point merge arrival flow integration technique: towards more complex environments and advanced continuous descent. In: AIAA Aviation Technology, Integration, and Operations Conference, Hilton Head, SC
4. Ivanescu D, Shaw C, Tamvaclis CL (2009) Models of air traffic merging techniques: evaluating performance of point merge. In: 9TH AIAA Aviation Technology, Integration, and Operations Conference, Hilton Head, SC
5. Meric OS, Usanmaz O (2013) A new standard instrument arrival: the point merge system. Aircr Eng Aerosp Technol 85(2):136–143
6. Beasley JE, Krishnamoorthy M, Sharaiha YM, Abramson D (2000) Scheduling aircraft landings - the static case. Transp Sci 34(2):180–197
7. Beasley JE, Krishnamoorthy M, Sharaiha YM, Abramson D (2004) Displacement problem and dynamically scheduling aircraft landings. J Oper Res Soc 55(1):54–64

8. Roling PC, Visser HG (2008) Optimal airport surface traffic planning using mixed-integer linear programming. Int J Aerosp Eng 2008(1):1–11
9. Capozzi B, Atkins S, Choi S (2009) Towards optimal routing and scheduling of metroplex operations. In: AIAA Aviation Technology, Integration, and Operations Conference, Hilton Head, SC
10. Bemporad A, Morari M (1999) Control of systems integrating logic, dynamics, and constraints. Automatica 35(3):407–427
11. Ibm ILOGCPLEX, V12.1 (2009) User's manual for CPLEX. International Business Machines Corporation, Armonk, NY

Flight Path Control for a Multi-body HALE Aircraft

Alexander Köthe and Robert Luckner

1 Introduction

High Altitude Long Endurance (HALE) aircraft have become an interesting alternative for satellites, e.g. for communication and surveillance tasks, since they offer more flexibility in operation. Due to their lightweight construction, their high-aspect-ratio and the use of solar panels, they can be operated in the stratosphere for up to 336 h (cf. QinetiQ Zephyr [9]). In contrast to satellites, HALE aircraft are not bound to a specific trajectory. They can be operated in a specific region for a certain amount of time. When a mission is completed, they can be recovered, relocated and used for another mission. Current HALE aircraft have large wing spans that can lead to large deformation with structural geometrical nonlinearities. On the flight to the mission altitude, HALE aircraft have to pass the troposphere, in which most of the weather events (gusts, turbulences) occur. Gusts may cause high aerodynamic loads leading to high wing bending moments that influence the fatigue strength of the aircraft. Additionally, non-uniform gust excitations of the highly flexible vehicle may result in larger deformations than caused by uniform ones [17]. A new alternative to one-wing HALE aircraft are Multi-Body Aircraft (MBA) that are investigated at TU Berlin's department of Flight Mechanics, Flight Control and Aeroelasticity. This concept is based on the linkage of several individual rigid-body aircraft to a single HALE aircraft in the mission altitude. A high-aspect-ratio can be accomplished without consideration of structural geometrical nonlinearities. Moreover, each single aircraft can be brought to the stratosphere with a helium balloon or fly on their own up to the mission altitude ensuring low aerodynamic loads and hence bending moments due to atmospheric disturbance in the troposphere. Lastly, the modularity

A. Köthe (✉) · R. Luckner
Technische Universität Berlin, Marchstraße 12, 10587 Berlin, Germany
e-mail: alexander.koethe@campus.tu-berlin.de

R. Luckner
e-mail: robert.luckner@tu-berlin.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_23

of a MBA permits exchange and return to ground of a single individual aircraft for repairing purposes without significantly affecting the flight mission.

A flight mechanic design for such a MBA HALE based on the DARPA Vulture program specification [11] was carried out. The objective of the Vulture program is the development of a heavier-than-air aircraft that can be operated with a payload of 1,000 lb (450 kg) over a period of five years. Besides those flight performance specifications, every aircraft has to be designed with a nearly rigid wing that leads to a heavier wing structure with negligible structural modes. For the coupling in the mission altitude each aircraft has to be able to fly individually. Therefore, the aircraft requires an adequate wing to produce sufficient lift, a horizontal and vertical stabilizer for stability, control surfaces (ailerons, elevator and flaps) for lateral and vertical controllability and a flight control system. The result of the design process was a formation with ten coupled aircraft that possesses a total wing span of 215 m with an aspect ratio of 66.8 and a total mass of 4090 kg. Additionally the outer aircraft of the MBA have a dihedral angle of 2.5°. This 80% cranked semi-span dihedral configuration reduces the span width, since the Oswald parameter is positive affected. The MBA HALE is able to fly 365 days in the mission altitude of 20 Km with a payload of 450 kg and a maximum (north or south) latitude of 40°. The detailed properties are given in Table 1. Figure 1 presents a illustration of the designed MBA HALE. As controls, the left and right flaps along the complete half span, an elevator, a rudder and one engine of each aircraft are used. In total the MBA has 50 control inputs.

A previous general concept investigation showed that an optimal linkage of two aircraft can be achieved with a joint that allows a pitch and roll motion between two aircraft and restrict a yaw motion [8]. As a consequence the degrees of freedom are much higher in comparison to a conventional fixed wing aircraft. This has an impact on the flight control system. This paper describes the development of a flight path control law for altitude and heading control. The flight control law is structured into the inner-loops that ensure damping as well as attitude and airspeed control and the outer-loop for altitude and heading control. The flight dynamics of the formation are discussed and the requirements that are derived from the aircraft design mission and the flight mechanical analysis are defined. The control law structure for the

Table 1 Properties of the designed Multi-body High Altitude Long Endurance Aircraft

Coupled aircraft	$i = 10$	Mission altitude	$H = 20$ km
Mission latitude	$Lat \leq 40°$	Complete span width	$b = 215.27$ m
Aspect ratio	$\Lambda = 66.76$	Total mass	$m = 4089.9$ kg
Design speed	$V = 33.15\ \frac{\mathrm{m}}{\mathrm{s}}$	Required energy	$E_{\mathrm{req}} = 3.2212$ GJ
Design lift coefficient	$C_L = 1.242$	Zero drag coefficient	$C_{D,0} = 0.0061$
Total payload	$m_{\mathrm{Pay}} = 450$ kg	Required payload power	$P_{\mathrm{req,pay}} = 5$ kW

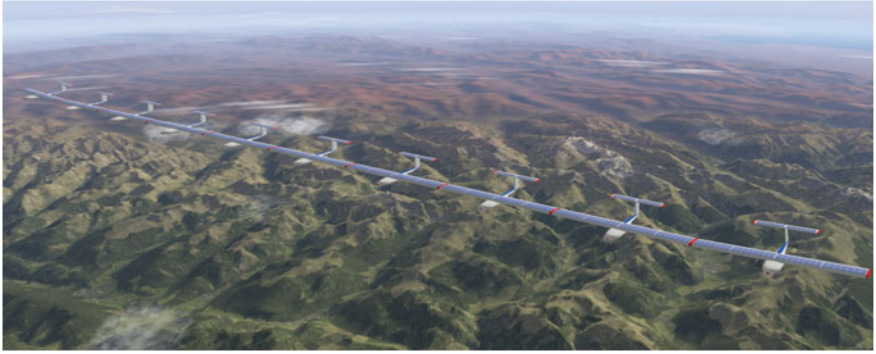


Fig. 1 Illustration of the designed Multi-body High Altitude Long Endurance Aircraft

inner-loop and outer-loop is introduced and the controller gains are determined. The control laws are tested in linear and nonlinear simulations for command following and disturbance rejection.

2 Flight Dynamics of Multi-body Aircraft

The flight dynamic model of the MBA for the flight control design is basesd on the equations of motion and models of external forces and moments (aerodynamics, thrust and weight). The equations of motion for the Multi-Body Aircraft are assembled following Kane's formalism [7] by using the software tool Autolev / Motion Genesis [10]. Aerodynamic forces and moments are modeled by a Vortex-Lattice method. Based on those methods and the design parameter, a nonlinear flight dynamic model is build up in Simulink. This model is linearized and analyzed by numerical perturbation.

2.1 Equations of Motion

The origin of the equations of motion is Kane's dynamical equation,

$$\tilde{\mathbf{F}}_r + \tilde{\mathbf{F}}_r^{\star} = 0 \quad (r = 1, \ldots, p)\,, \tag{1}$$

where $\tilde{\mathbf{F}}_r$ are the generalized active forces, $\tilde{\mathbf{F}}_r^{\star}$ are the generalized inertial forces and p is the number of degrees of freedom of the system in the reference frame. In Eq. 1, the denotation "generalized force" includes inertial and active forces as well as inertial and active moments (translation and rotation) [7]. The generalized inertial force is determined by

$$\tilde{\mathbf{F}}_r^\star = -\sum_{j=1}^{l} {}^N\mathbf{F}_k^{CG,j} \frac{\partial^N \mathbf{v}^{CG,j}}{\partial u_r} - \sum_{j=1}^{l} {}^N\mathbf{M}_k^{CG,j} \frac{\partial^N \omega^{B,j}}{\partial u_r}, \tag{2}$$

where ${}^N\mathbf{v}^{CG,j}$ is the velocity of the centre of gravity of the jth body in the Newtonian frame, ${}^N\omega^{B,j}$ the angular velocity of the body frame against the Newtonian frame of the jth body, u_r the generalized speeds and $\mathbf{F}_k$ and $\mathbf{M}_k$ are the mass force and mass torque of the jth body decomposed as

$${}^N\mathbf{F}_k^{CG} = m\left(\frac{d\ {}^B\mathbf{v}^{CG}}{dt} +^N \omega^B \times^B \mathbf{v}^{CG}\right) \text{ and } {}^N\mathbf{M}_k^{CG} = \mathbf{I}^N\dot{\omega}^B +^N \omega^B \times \left(\mathbf{I}\ {}^N\omega^B\right), \tag{3}$$

and l the number of rigid bodies in the system. The generalized active force is given by

$$\tilde{\mathbf{F}}_r = \sum_{j=1}^{l} {}^N\mathbf{F}_A^{CG,j} \frac{\partial^N \mathbf{v}^{CG,j}}{\partial u_r} + \sum_{j=1}^{l} {}^N\mathbf{M}_A^{CG,j} \frac{\partial^N \omega^{B,j}}{\partial u_r}, \tag{4}$$

where $\mathbf{F}_A$ and $\mathbf{M}_A$ are the active forces and moments acting at or around the center of gravity. Both, velocities and angular velocities in Eqs. 2 and 4 are based on generalized speeds u_r $(r = 1, \ldots, p)$ that are derived from the generalized coordinates q_s $(r = 1, \ldots, n)$, where n is the number of generalized coordinates and p the number of generalized speeds, with

$$u_r = \sum_{s=1}^{n} Y_{rs}\ \dot{q}_s + Z_r \quad (r = 1, \ldots, p) \tag{5}$$

where Y_{rs} and Z_r are functions of the generalized coordinates and time. Equation 5 is called kinematical differential equation [7].

Generalized coordinates and speeds are restricted by constraint equations. Considering a formation of i aircraft, the number of generalized coordinates is determined by

$$n = 6 \times i - M, \tag{6}$$

where M is the number of holonomic constraint equations and every aircraft possesses the position (three Cartesian coordinates) and the orientation against the Newtonian frame (three Euler angles[1]) as generalized coordinates. Holonomic constraints are expressed by the generalized coordinates, while N nonholonomic constraint equations can be expressed as m relationships

[1] Euler angles is the common denotation in flight mechanics. In multi-body dynamics the orientation angles are called Cardan angles.

$$u_r = \sum_{s=1}^{p} A_{rs}\, u_s + B_r \quad \text{with} \quad p = n - m \quad (r = p + 1, \ldots, n) \tag{7}$$

where A_{rs} and B_r are functions of the generalized coordinates and time [7]. In case of using Eq. 7, holonomic constraint equations can be formed into nonholonomic constraint equations, but not necessarily vice versa. Hence the number of generalized speeds p is given by

$$p = 6 \times i - M - N. \tag{8}$$

The selected joint of the MBA allows a pitch and roll motion between two neighboring aircraft A and B. The other generalized speeds are restricted with four non-holonomic constraints

$$\begin{matrix} \left({}^N\mathbf{v}^{CAB} -^N \mathbf{v}^{CBA}\right)\mathbf{e}_{xg} = 0 & \left({}^N\mathbf{v}^{CAB} -^N \mathbf{v}^{CBA}\right)\mathbf{e}_{yg} = 0 \\ \left({}^N\mathbf{v}^{CBA} -^N \mathbf{v}^{CAB}\right)\mathbf{e}_{zg} = 0 & \left({}^N\omega^{A} -^N \omega^{B}\right)\mathbf{e}_{zg} = 0 \end{matrix}, \tag{9}$$

where *CAB* is the connection point between both aircraft. The $N = 3$ nonholonomic motion constraints for the body fixed velocities can be further expressed as $M = 3$ holonomic constraint equations. This is no longer applicable for the motion constraints of the rotation. Using the roll rate u_1, pitch rate u_2 and yaw rate u_3 as generalized speeds and the roll angle q_1, pitch angle q_2 and yaw angle q_3 as generalized coordinates, the kinematic differential equations are derived in [14] as

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 1 & \sin(q_1)\tan(q_2) & \cos(q_1)\tan(q_2) \\ 0 & \cos(q_1) & -\sin(q_2) \\ 0 & \frac{\sin(q_1)}{\cos(q_2)} & \frac{\cos(q_1)}{\cos(q_2)} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{10}$$

Even if u_3 is equal to zero, pitch and roll rate influence all other generalized speeds due to the nonholonomic motion constraints. Thus, $N = 1$ nonholonomic constraints exist, but no holonomic constraints. As a consequence, every coupled aircraft results in $N = 4$ nonholonomic constraints and $M = 3$ holonomic constraints. Hence a formation with ten coupled aircraft and motion constraints of Eq. 9 has 33 generalized coordinates and 24 generalized speeds. By neglecting the navigation states (Ψ and three Cartesian coordinates), four additional generalized coordinates are removed. The arising equations of motions are highly nonlinear due to the rotational degrees of freedom, but the existing nonlinearities can be described mathematically exact. This is an advantage in comparison to structural nonlinearities that can only be approximated by mathematical models. Linearizing those nonlinear equations leads to a first order differential equation system with 53 states.

2.2 External Active Forces and Moments

The active forces and moments that have to be considered for the equations of motion are: aerodynamic forces (in the aerodynamic reference frame) $\mathbf{R}_A$, thrust (in the body fixed reference frame) $\mathbf{T}$ and weight (in the Newtonian frame) $\mathbf{W}$ of each aircraft as well as aerodynamic moments (in the aerodynamic reference frame) $\mathbf{M}_A$ and thrust moments (in the body fixed reference frame) $\mathbf{M}_T$. For each single aircraft, it is assumed that the thrust acts at the center of gravity. This results in a zero thrust moment. Regarding to Eq. 4, the active force at the jth aircraft in the body fixed reference frame is determined as

$$^{b}\mathbf{F}_{A}^{CG,j} = \mathbf{T}_{b,a,j}\ \mathbf{R}_{A,a,j} + \mathbf{T}_{b,n,j}\ \mathbf{W}_{n,j} + \mathbf{T}_{b,j}, \tag{11}$$

where $\mathbf{T}_{b,a}$ is the transformation matrix from the aerodynamic reference frame (index a) to the body fixed reference frame (index b) and $\mathbf{T}_{b,n}$ is the transformation matrix from the Newtonian reference frame[2] (index n) to the body fixed reference frame. The active moment of the jth aircraft in the body fixed reference frame for Eq. 4 is computed with

$$^{b}\mathbf{M}_{A}^{CG,j} = \mathbf{T}_{b,a,j}\ \mathbf{M}_{A,a,j} + \underbrace{\mathbf{M}_{b,F}}_{=0}. \tag{12}$$

The gravity of the jth aircraft is defined in the Newtonian reference frame as

$$\mathbf{W}_{n,j} = \begin{bmatrix} 0, & 0, & m_j\, g \end{bmatrix}^T \tag{13}$$

where m_j is the mass of the jth aircraft and g is the gravitational acceleration. The thrust of the jth aircraft is calculated with

$$\mathbf{T}_{b,j} = \begin{bmatrix} F_j, & 0, & 0 \end{bmatrix}^T \tag{14}$$

where F_j is the thrust of the jth engine.

Aerodynamic forces and moments are generated by the wing, horizontal stabilizer and vertical stabilizer. The overall wing is built up by the wings of the single aircraft. As a consequence, a wing with a high aspect ratio is formed. The aerodynamic forces and moments are calculated in MATLAB with the vortex lattice method described in [6]. For this purpose, the wing is divided into a finite number of surfaces. The flow conditions on every surface depend on both the air density and aerodynamic parameter (angle of attack, sideslip angle and airspeed) that are also a function of the yaw rate of the whole formation and the pitch and roll rate of every aircraft. The software tool Autolev / Motion Genesis computes those kinematic equations after the definition of motion constraints. Based on the aerodynamic conditions, the vortex

[2] In flight mechanics often called geodetic reference frame that is considered as an inertial system.

lattice method determines lift, yaw (considering also the zero drag coefficient) and induced drag of every surface. By rotation of the normal vectors of the surfaces, flap deflections are considered. For the MBA every single aircraft is equipped with flaps along the complete left $\eta_{K,left}$ and right wing $\eta_{K,right}$ with 25% flap chord. Based on the aerodynamic forces at every surface and the corresponding collocation point, lift, induced drag and yaw can be assembled to each single aircraft. If the surface collocation point of the complete wing belongs to the ith aircraft, the forces are assigned to the ith aircraft. The corresponding aerodynamic moments are calculated by the lever arm of the surface collocation point (that belongs to the ith aircraft) to the center of gravity of the ith aircraft. To consider the downwash, horizontal and vertical stabilizer are included to the vortex lattice computation. The forces and moments can be directly assigned to the corresponding aircraft. The influence of rudder ζ and elevator η with 25% flap chord of vertical and horizontal stabilizer is computed by rotation of the normal vector of the corresponding surfaces. Finally, the part of the zero drag is added to each aircraft.

2.3 Linear Analysis of the Multi-body Aircraft

Based on the equation of motion, a Simulink Model is generated and linearized to analyze the flight dynamics of the formation. Additionally to the previous introduced flight dynamic model, the difference of the yaw to the yaw angle of the main aircraft

$$\Delta\Psi_i = \Psi_5 - \Psi_i \quad \text{with} \quad i \in \{1, \dots, 10\} \setminus \{5\} \tag{15}$$

is considered as state instead of the absolute yaw angle of every aircraft. The state vector

$$\mathbf{x} = \begin{matrix} \big[q_{\mathrm{AC5}}, \alpha_{\mathrm{AC5}}, V_{\mathrm{AC5}}, \gamma_{\mathrm{AC5}}, r_{\mathrm{AC5}}, \beta_{\mathrm{AC5}}, \dots \\ p_{\mathrm{AC5}}, \Phi_{\mathrm{AC5}}, \mathbf{x}_{\mathrm{AC1}}, \mathbf{x}_{\mathrm{AC2}}, \mathbf{x}_{\mathrm{AC3}}, \mathbf{x}_{\mathrm{AC4}}, \dots \\ \mathbf{x}_{\mathrm{AC6}}, \mathbf{x}_{\mathrm{AC7}}, \mathbf{x}_{\mathrm{AC8}}, \mathbf{x}_{\mathrm{AC9}}, \mathbf{x}_{\mathrm{AC10}}\big]^T \end{matrix} \quad \text{with} \quad \mathbf{x}_{\mathrm{AC}i} = \begin{bmatrix} q_{\mathrm{AC}i} \\ \Theta_{\mathrm{AC}i} \\ p_{\mathrm{AC}i} \\ \Phi_{\mathrm{AC}i} \\ \Delta\Psi_{\mathrm{AC}i} \end{bmatrix}^T \tag{16}$$

has 53 elements (resulting of generalized speeds and coordinates). The fifth aircraft is selected as the main aircraft. A first analysis showed that the eigenvalues related to the yaw motion of every aircraft are close to the origin of the complex plane. Hence those nine states are neglected and the state vector is reduced to 44 elements. The eigenvalues of the system are plotted in Fig. 2. Twelve out of these eigenvalues are unstable. In total, the system has eight real eigenvalues (two are unstable) and 18 complex conjugate eigenvalue pairs (five are unstable). Based on the eigenvectors, a clear separation between lateral, longitudinal and linkage eigenvalues is not possible. The eigenvectors shows a strong coupling between all states in all modes.

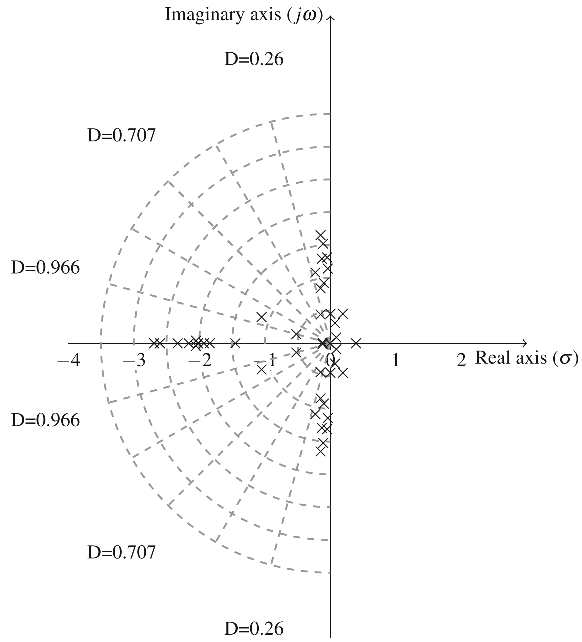


Fig. 2 Illustration of the eigenvalues of the linearized Multi-body HALE Aircraft in the complex plane

Nevertheless, a short period mode, a phygoid, a Dutch roll, an unstable spiral as well as roll and pitch linkage modes are recognizable. In [8] the eigenvectors for a formation of three coupled aircraft are investigated in detail.

3 Control Law Design

Although the aircraft dynamics strongly deviate from classical aircraft, the cascade control principle for conventional aircraft is applied to the Multi-Body aircraft. Hence, the flight path control laws contain three loops: The basic flight control law shall ensure damping of rates, airspeed, and wing shape holding control, while the outer-loop is used for attitude control (flight control) and flight path control (flight guidance). Only cruise flight is considered. The HALE aircraft shall operate in the mission altitude without significant altitude change, but with heading changes to ensure a cruise flight above a certain area. Therefore basic control laws are required to control pitch as well as bank angle of the aircraft. Attitude control is achieved with a special concept. The left middle aircraft is used as reference for the pitch angle of all left aircraft and the right middle aircraft for the other ones. In a left side turn, the left aircraft reduce their pitch angle, while the other aircraft hold or increase the pitch. With this, an imbalance in the lift distribution occurs and the aircraft banks to the left side. This is nothing else than the classical aileron concept. If the pitch

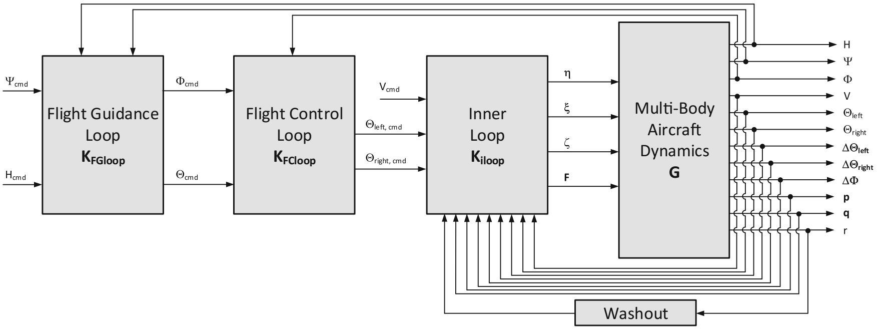


Fig. 3 Block diagram for the cascade flight path control law structure of the MBA

angle of the formation is changed, the demand values of left and right pitch angle have to be identical. Left and right pitch angle are controlled within the inner-loop. Changing of the demand values for left and right pitch angle for attitude control is carried out in the first outer-loop. Based on the attitude control, the second outer-loop for azimuth and altitude control is designed. Figure 3 illustrates the corresponding control law structure. The outputs $\Delta\Theta$ and $\Delta\Phi$ represent the wing shape differences and will be introduced in Sect. 3.2. In addition the inner-loop and two outer-loops, a washout filter is included in the flight control system. Because of the yaw damping, the inner-loop tries to damp also steady yaw rates that occur in turned flight. This is not desirable. With the help of a washout only frequencies relating to the Dutch roll are passed to the inner-loop. Steady yaw rates are not affected.

3.1 Requirements

Requirements for the flight control laws are derived from the analysis of the nominal plant and the aircraft design. The nominal plant has ten unstable eigenvalues. The stabilization of the plant is the main requirement of the flight control law. After the Helios mishap, the NASA in its function as an investigation organization recommend the use of nonlinear methods in the aircraft modeling and the flight control law design [12]. Multi-Body aircraft's equations of motions are nonlinear due to the trigonometric functions caused by joint modeling and kinematic differential equations of the individual aircraft. The flight control law design process shall consider those nonlinear effects. Due to gusts and turbulence, disturbances act on the aircraft that have to be rejected by damping of the rates [2] and by an airspeed controller. Flying over a one-year period with solar energy requires a low energy consumption that is ensured by flying with minimum power consumption, which, in turn, requires flying at a high lift coefficient. The aircraft operates close to the stall speed. To avoid stall, the flight control law has to achieve speed holding with high precision. In a basic investigation it turned out that the planar wing is optimal for low induced drag.

This leads to the requirement that the wing shape has to hold straight. To save energy, an extensive control surface activity has to be avoided. The long wing carries the danger of stall in turned flight. The outer wing has a high airspeed while the inner wing airspeed is very low.

Regarding airspeed and bank angle control, quantitative requirements can be defined. The aircraft flies with 90% of the maximum lift coefficient. To avoid stall, a maximum drop in the airspeed of $1.198\ \frac{\mathrm{m}}{\mathrm{s}}$ is allowed. This maximum airspeed reduction has to be considered also in turn flight. While the aircraft yaws, the airspeed is increased on the outside of the curve and decrease on the inside of the curve. Considering the half span of the aircraft and the maximum permissible airspeed drop, the maximum yaw rate is $0.64°\ \mathrm{s}^{-1}$. An alternative approach would include the direct control of the local angles of attack instead of controling the speed of the (central) reference aircraft of the flight formation. Because such an approach would require more sensors and thus render the Flight Control Law more complex, it is not yet taken into account. Note, however, that measuring the local angles of attack is actually conceivable for the real application case of a HALE aircraft, eventually allowing to achieve a flight envelope protection.

Quantitative requirements for the flight path flight control law, like described in SAE AS94900 for piloted aircraft, do not exist for unmanned aircraft (especially for HALE). Those requirements depend on the actual mission. Currently, the MBA HALE shall be used for surveillance task and broadcast of television or internet. The altitude of the aircraft shall be maintained and heading changes shall be carried out in an adequate time and with an adequate radius above the area of interest. In this basic investigation only the functionality of a flight path with an MBA is shown.

3.2 Inner-Loop Control Law

The flight dynamics of the MBA are described as a nonlinear multiple input, multiple output (MIMO) differential equation system with significant coupling between the state variables. In general, there exist a variety of control design methods for nonlinear plants. In this paper the $\mathcal{H}_\infty$ robust control design method is used to describe the nonlinear dynamics as uncertainties. This method allows a definition of the requirements in the frequency domain and can deal directly with an uncertainty description during the design process [15]. A disadvantage of the method is the high order of the control laws. Nevertheless the computation effort can be distributed to all ten flight control computer.

The inner-loop shall achieve stability, damp pitch (every aircraft), roll (every aircraft) and yaw (formation) rate and controls the left and right pitch angle as well as the airspeed. The differences in the pitch angles are expressed by

$$\begin{aligned} \Delta\Theta_i &= \Theta_{\mathrm{AC}5} - \Theta_{\mathrm{AC}i} \quad \text{with} \quad i \in \{1, \ldots, 4\}\,, \\ \Delta\Theta_i &= \Theta_{\mathrm{AC}6} - \Theta_{\mathrm{AC}i} \quad \text{with} \quad i \in \{7, \ldots, 10\}\,, \end{aligned} \tag{17}$$

with the fifth aircraft (Θ_{AC5}) and sixth aircraft (Θ_{AC6}) as reference for left and right side pitch angle. The bank angle differences are formulated by deviation of the separate aircraft to the fifth aircraft (as reference aircraft) with

$$\Delta\Phi_i = \Phi_{\text{AC5}} - \Phi_{\text{AC}i} \quad \text{with} \quad i \in \{1, \ldots, 10\} \setminus \{5\}. \tag{18}$$

Those 17 differences lead to 41 total output values (21 rates, 17 wing shape differences, left and right pitch angle, and the airspeed) that are controlled within the inner-loop. As input variables all elevator deflections are used. The left and right wing flap are summarized into a common aileron

$$\xi_{\text{AC}i} = \eta_{\text{K, right, AC}i} - \eta_{\text{K, left, AC}i} \quad \text{with} \quad i \in [1, \ldots, 10] \tag{19}$$

of every aircraft. The thrust and the rudder are set to identical values for every aircraft. This leads to 21 input variables. Hence, the linearized state-space model used for the inner-loop design is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{Ax}(t) + \mathbf{Bu}(t) \ , \\ \mathbf{y}(t) &= \mathbf{Cx} + \mathbf{Du}(t) \end{aligned} \tag{20}$$

with $\mathbf{y}$ as control variables, $\mathbf{u}$ as control inputs and $\mathbf{x}$ as states. By applying the Laplace transformation (assuming that the feedforward matrix $\mathbf{D}$ is equal to zero), this state-space models yields the system

$$\begin{aligned} s\,\hat{\mathbf{x}}(s) &= \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s) \ , \\ \hat{\mathbf{y}}(s) &= \mathbf{C}\hat{\mathbf{x}}(s) \end{aligned} \tag{21}$$

that describes the plant in the frequency domain. Hence, the transfer function of the plant for inner-loop design is determined by

$$\mathbf{G}(s) = \hat{\mathbf{y}}(s)\ \hat{\mathbf{u}}^{-1}(s) = \mathbf{C}\left[s\,\mathbf{I} - \mathbf{A}\right]^{-1}\mathbf{B}. \tag{22}$$

3.2.1 $\mathcal{H}_\infty$ loop shaping

Figure 4 illustrates the augmented plant that is used for the $\mathcal{H}_\infty$ synthesis. The subject of the applied loop shaping is to minimize γ of the norm

$$\|\mathbf{F}_l(\mathbf{P}, \mathbf{K})\|_\infty < \gamma \tag{23}$$

under the constraint that $\mathbf{K}$ stabilizes the plant. The lower linear fractional transformation $\mathbf{F}_l$ represents the closed loop transfer function from the exogenous inputs to the exogenous outputs. The exogenous inputs are the demand values $\mathbf{r}$ for output values $\mathbf{y}$ that are controlled within the inner and disturbances values $\mathbf{d}$ that represent wind velocities acting on every aircraft. As discussed later, the output of the uncer-

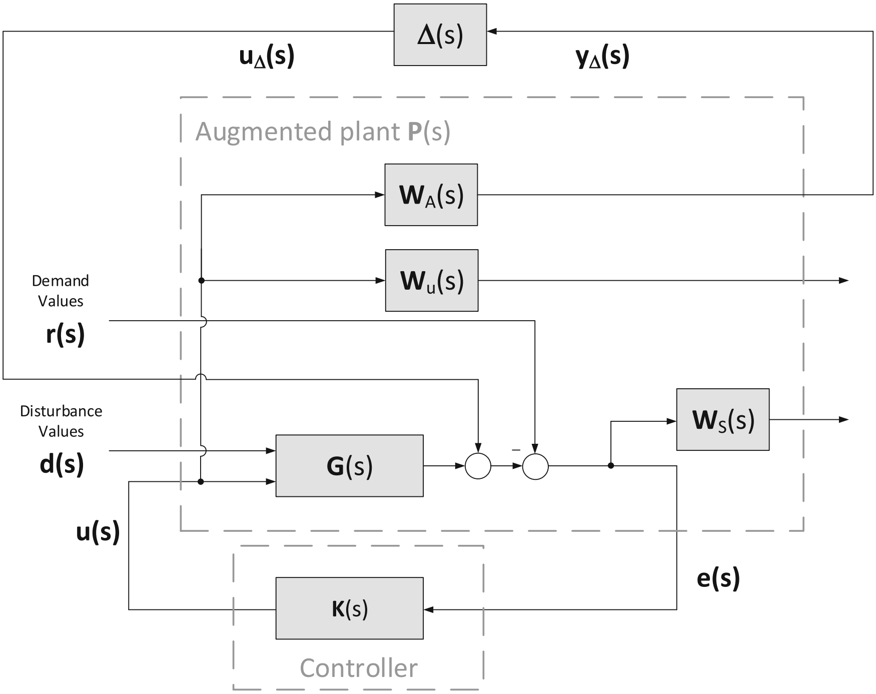


Fig. 4 Block Diagram of the $\mathcal{H}_\infty$ inner-loop synthesis with design weights $\mathbf{W}_S$, $\mathbf{W}_U$, deviation matrix $\mathbf{W}_A$ for an additive perturbation configuration and uncertainty matrix Δ

tainty matrix $\mathbf{u}_\Delta$ are also used as inputs in the control law design. The exogenous outputs are the weighted control inputs $\mathbf{W}_u\mathbf{u}$ to restrict controls, and the weighted error of the control values $\mathbf{W}_e\mathbf{e}$ to shape the sensitivity functions. Using an additive input perturbation configuration to describe the uncertainties, the deviation matrix $\mathbf{W}_A\mathbf{u}$ is used as additional exogenous output. For the use of normalized weights, the plant has to be normalized to the maximum input signal $u_{\max,i}$, the maximum disturbance values $d_{\max,j}$ and the maximum errors $e_{\max,k}$ [15]. The general input vector

$$\tilde{\mathbf{u}} = [\mathbf{u} \quad \mathbf{d}]^T \tag{24}$$

collects the control inputs and disturbance values. The maximum values for every general input $l \in \{1, \ldots, i + j\}$ and every output k are collected in two diagonal matrices with

$$\mathbf{D}_{\tilde{u}} = \text{diag}\left(\tilde{u}_{\max,l}\right) \quad \text{and} \quad \mathbf{D}_e = \text{diag}\left(y_{\max,j}\right). \tag{25}$$

In the case of the flight control law for the MBA, the maximum input signal for the elevator, ailerons and rudder is set to 10° and for the thrust to 50 N. According to

Table 2 Parameter of the sensitivity weight for the different control variables

	Number	Bandwidth	Target error	Target overshoot
		$\omega_{\mathrm{BW}}\ \left[\mathrm{rad\ s^{-1}}\right]$	A	M
Pitch rate	10	10	100	1.15
Roll rate	10	10	100	1.15
Yaw rate	1	10	100	1.15
Bank angle differences	9	10	100	1.15
Left pitch angle differences	4	10	100	1.15
Right angle differences	4	10	100	1.15
Left pitch angle	1	5	100	1.1
Right pitch angle	1	5	100	1.1
Airspeed	1	5	100	1.1

the certification specification for large aircraft CS-25 the maximum gust velocity is 6.36 $\frac{\mathrm{m}}{\mathrm{s}}$.[3] This value is rounded to 7 $\frac{\mathrm{m}}{\mathrm{s}}$ and used as maximum disturbance value. The error for the pitch and roll rates is set to 5 $\frac{1^{\circ}}{\mathrm{s}}$. The maximum error for the yaw rate follows from the requirements with 0.64 $\frac{1^{\circ}}{\mathrm{s}}$. The error of wing shape differences is set to 5° and for left and right pitch angle to 5°. Based on the predefined requirements, a maximum airspeed error of 1.198 $\frac{\mathrm{m}}{\mathrm{s}}$ is considered. The normalized plant $\mathbf{G}_n$ is computed with

$$\mathbf{G}_n = \mathbf{D}_e^{-1}\,\mathbf{G}\,\mathbf{D}_u\,. \tag{26}$$

The performance weights for shaping the sensitivity function are defined as block-diagonal matrix with the transfer function

$$w_{\mathrm{S}} = \frac{\frac{s}{M} + \omega_{\mathrm{BW}}}{s + \omega_{\mathrm{BW}}\ A}, \tag{27}$$

on the main diagonal. The parameter for the different control variables are listed in Table 2. The weight for the control surface deflections is selected with the transfer function

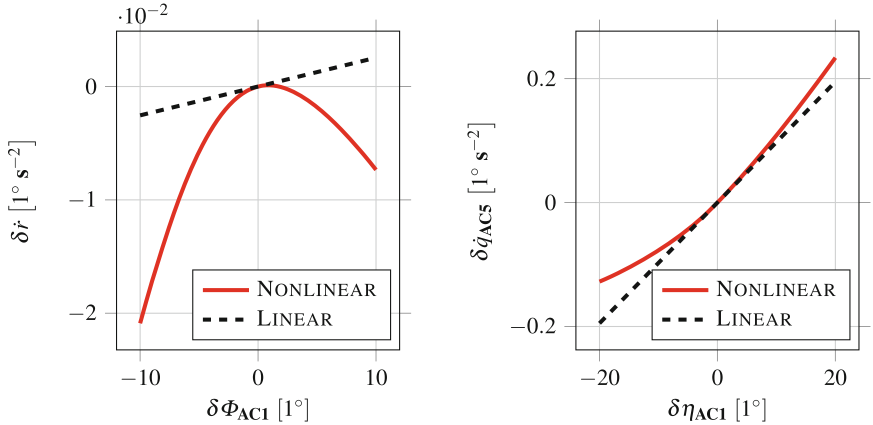
$$W_{\mathrm{U}} = \mathrm{diag}\left(\frac{\left(\frac{1}{\omega_{\mathrm{BW,\ actuator}}}s + 1\right)^2}{10\left(\frac{1}{10\omega_{\mathrm{BW,\ actuator}}}s + 1\right)^2}\right) \tag{28}$$

[3] The MBA belongs to CS-23 aircraft, but the flight envelope in terms of gust ends at 15,240 m (CS 23.333, [3]). Since the CS-25 considers altitudes up to 18,288 m (CS 25.341, [4]), this specification is applied.

that shall ensure that the controller only commands inputs that are within the bandwidth of the actuators $\omega_{\text{BW, actuator}} = 18.95\ \frac{\text{rad}}{\text{s}}$. The uncertainty is described as unstructured uncertainty in the additive perturbation configuration with

$$\mathbf{W}_{\text{A}}(s)\,\Delta(s) = \mathbf{G}_{\text{P},i} - \mathbf{G}_0,, \tag{29}$$

with $\mathbf{G}_0$ as nominal system (based on the linearization) and $\mathbf{G}_{\text{P},i}$ as family of perturbed system dynamic that cover the nonlinearities. The determination of the perturbed systems is carried out as follow: The considered linear plant arises out of the linearization of the nonlinear equation of motion. Linearization accuracy drops off for highly nonlinear systems since the applied Taylor series truncates quadratic and higher order terms of the derivation variables [1]. Hence, the linear approximation of the fully nonlinear trajectory is only valid in a small interval around $\delta x = 0$. To increase the range of valid solutions to an interval of interest $[\delta x_1, \delta x_2]$, the gradient of the fully nonlinear trajectory of every δx within the range of interest is determined. Based on the maximum $\left(\frac{\partial f}{\partial x}\right)_{\max}$ and minimum gradient $\left(\frac{\partial f}{\partial x}\right)_{\min}$, the maximum negative and positive variation to the nominal parameter $\left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_0\\u=u_0}}$ is determined. Those variations represent the possible range of the corresponding entry in the system matrix $\mathbf{A}$ and, hence, the parameter can be assumed as uncertain. It has to be noted that the parameter could be varied only negatively or positively to the nominal parameter. The same procedure can be carried out for the input values. The differences between fully nonlinear trajectory and linear approximation for one entry of the system matrix and input matrix are exemplary illustrated in Fig. 5. The consideration of all parameter



(a) Effect of changes in the bank angle of the first aircraft on the yaw rate derivative of the formation

(b) Effect of changes in the elevator of the first aircraft on the pitch rate derivative of the fifth aircraft

Fig. 5 Differences between the fully nonlinear trajectory and the linear approximation

would lead to a very large computation effort. In order to avoid this, conditions are introduced to decide if a parameter is assumed as uncertain or not. If the difference between calculated maximum or minimum slope in the range of interest and the slope in system or input matrix is smaller than 0.1, the parameter is not assumed to be uncertain. After applying this criteria, the system matrix contains 245 uncertain parameter that correspond to 12% of all entries. In the input matrix, 476 uncertain parameter remain, corresponding to 22% of all derivatives of the input matrix. In sum, 721 uncertain parameter are identified for the whole system description. Since the plant is not squared (different numbers of inputs and outputs), the unstructured uncertainties have to describe by a reduced number of inputs and outputs. Most uncertainties in the system matrix are related to the rates and the uncertain parameter in the input matrix are exclusively connected to the aerodynamic surface deflections. Hence, only the 21 rates (pitch and roll rate of every aircraft and the yaw rate of the formation) and the 21 aerodynamic control surface inputs (aileron and rudder of every aircraft as well as the rudder that is equal for all aircraft). The plant is now squared. With the help of MATLAB the uncertain plant is determined and a sample of 7210 random uncertain models is generated. Different unstructured uncertainty descriptions are applied to the uncertain models and the plant, since the results differs from method to method [15]. The criteria of low entries in the deviation matrix is fulfilled by using an additive perturbation configuration. The maximum singular values for the differences of all 7210 uncertain models to the nominal plant together with an upper bound are determined with the transfer function

$$w_{\mathrm{a}} = \frac{\frac{1}{0.05}s}{\left(\frac{1}{0.05}s+1\right)\left(\frac{1}{1}s+1\right)}. \tag{30}$$

The weight and 50 randomly selected maximum singular values of the differences between uncertain models and the nominal plant are illustrated in Fig. 6. The results of the random sampling are representative for the complete uncertain model family.

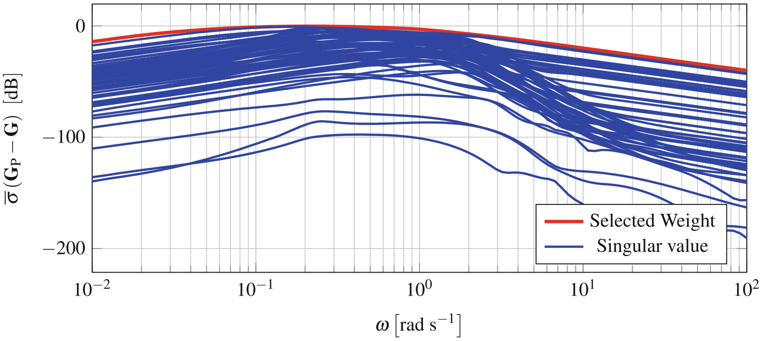


Fig. 6 Maximum singular value of the differences between 50 randomly selected uncertain models and the nominal plant and the corresponding weight for the deviation matrix

The deviation matrix $\mathbf{W}_\mathrm{A}$ is a block diagonal matrix with the transfer function of Eq. 30 on the main diagonal.

With the introduced weights, the control law process is carried out and a control law with the order of 191 is calculated. The controller itself is stable and the maximum frequency is 7.2 Hz. Evaluation of robust stability is carried out by using the $\mathbf{M}\Delta$ structure. In this control loop description, the controller and augmented plant are connected with each other. This leads to the system $\mathbf{M}$. As exogenous inputs and outputs for this system only the outputs of the uncertainty matrix Δ and outputs of the deviation weight are used. Based on the small gain theorem, the condition

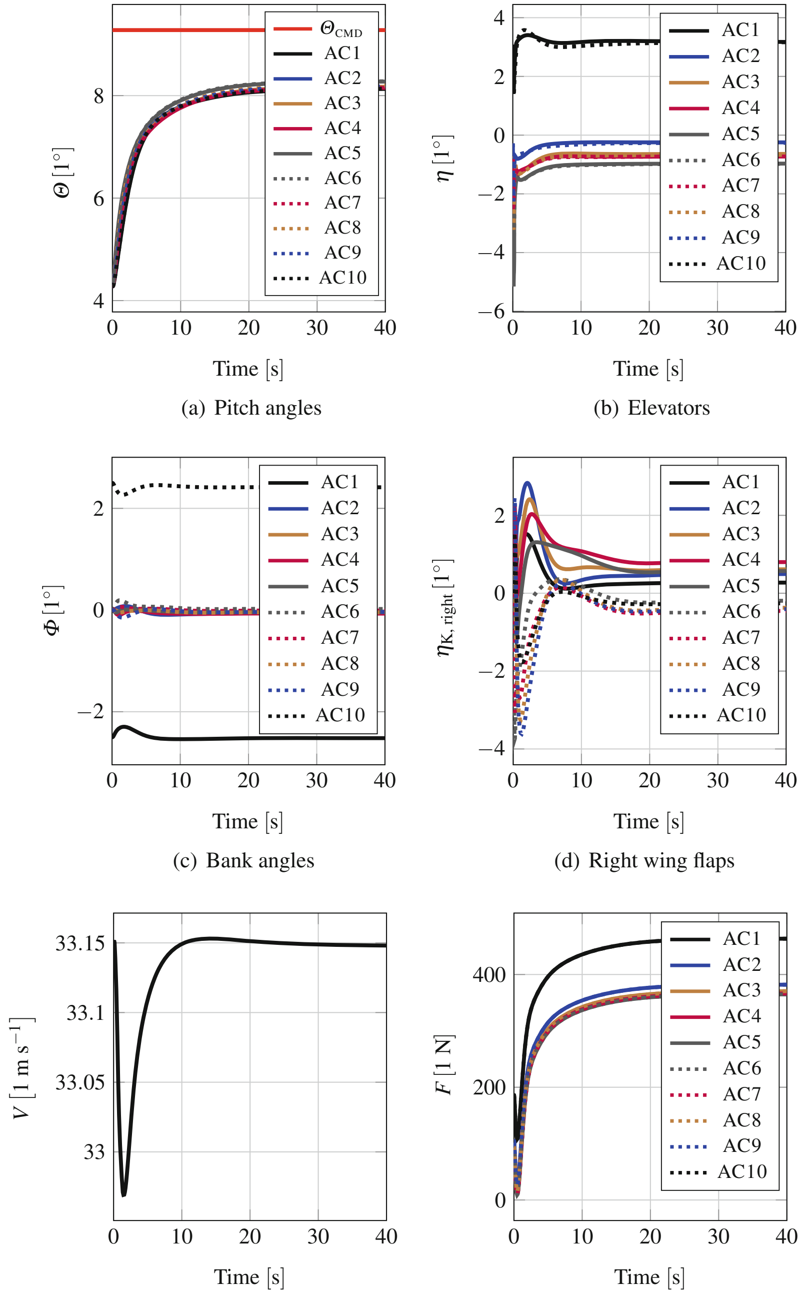
$$\overline{\sigma}\left(\mathbf{M}\right) < 1 \tag{31}$$

has to be fulfilled for robust stability with $\overline{\sigma}$ as maximum singular value [15]. The developed control law fulfills the condition of Eq. 31 is robustly stable against the modeled uncertainties. Since the order is too high, model reduction with the balanced truncation method is applied [15]. To define the minimal achievable control law order, the order of the control law was reduced from full (order 191) to one by one order steps until the stop criteria (that robust stability is still achieved) is violated. In this context, the control law was successfully reduced to the order of 61.

3.2.2 Results

The designed inner-loop controller with an order of 61 is now evaluated with respect to the time dimension. Therefore, the control law is implemented in the non-linear simulation environment of the MBA. In a nonlinear simulation study, the left and right pitch angle command is increased to 5°. The results are illustrated in Fig. 7. All pitch angles respond simultaneously. This indicates that the wing shape controller relating to the pitch angle works well. A steady error occurs, which is expected by considering the sensitivity function. The steady value of the pitch angle is reached within 20 s. The corresponding elevator deflections are low. No high frequency deflections are identified. Deviations in the bank angle are very low and, hence, the requirements of the wing shape holding relating to the bank angle is completely fulfilled. Also, the amounts and frequencies of wing flap deflections are low and within an acceptable limit. The drop in the airspeed is very low. That was one requirement to avoid stall conditions. Nevertheless, the thrust is very high and exceeds the limits of the engine that works in steady flight condition with half power. Thus, for the outer-loops it has to be considered to limit the maximum pitch angle to 2°. In summary, the inner-loop fulfills all requirements completely.

The aircraft response to a non-uniform distributed vertical gust is investigated. Figure 8 shows the nonlinear results. The gust acts at the second aircraft and is modeled as an $1 - \cos$ gust with



(a) Pitch angles

(b) Elevators

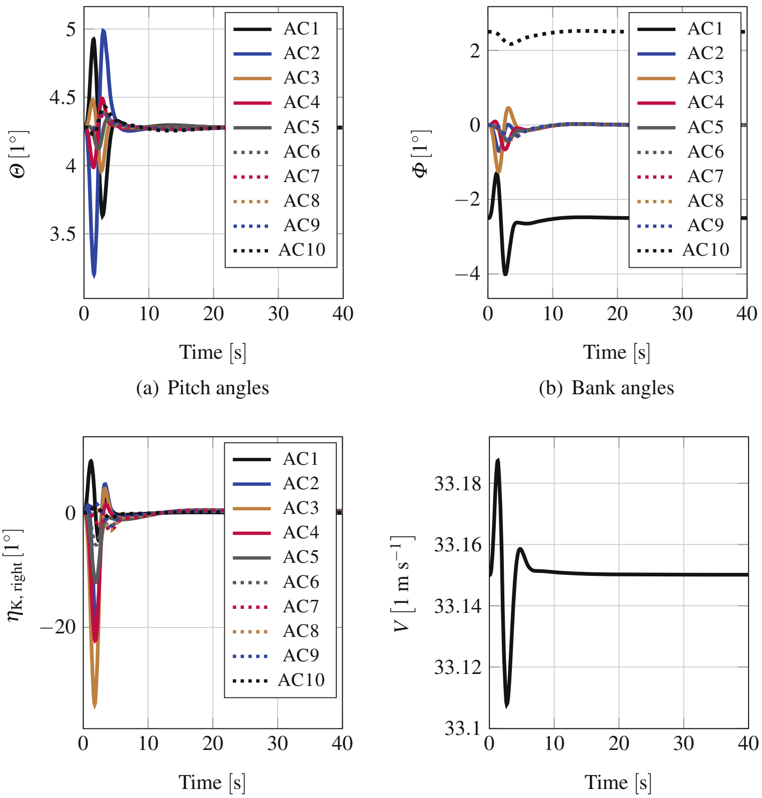
(c) Bank angles

(d) Right wing flaps

(e) Airspeed

(f) Engine thrust

Fig. 7 Nonlinear simulation results of inner-loop for a step input to the *left* and *right side* pitch angle command of 5°



(a) Pitch angles

(b) Bank angles

(c) Right wing flaps

(d) Airspeed

Fig. 8 Nonlinear simulation results of inner-loop for a vertical gust acting on the second aircraft

$$w_w\,(t) = \frac{U_{\mathrm{gust}}}{2}\left(1 - \cos\left(2\,\pi\,\frac{V_{\mathrm{ref}}}{2}\,t\right)\right), \tag{32}$$

with $U_{\mathrm{gust}} = 6.36\ \frac{\mathrm{m}}{\mathrm{s}}$ as gust amplitude (c.f. CS-25 for an altitude of 18,288 m [4], L as gust length that is equal to 25 times the chord length and $V_{\mathrm{ref}} = 33.15\ \frac{\mathrm{m}}{\mathrm{s}}$ as reference airspeed [18]. In the aircraft response, it becomes clear that the gust acts at the second aircraft, because the bank of the first and third aircraft are contrarily affected. The values of the wing shape differences are small and acceptable. Also, the airspeed changes are within satisfactory limits, since the airspeed drop is very low. Although a non-uniformly distributed gust acts on the formation, the bank angle changes are very small. The linkages, as shown by the bank angle differences, do not transfer bending moments and hence the Multi-Body HALE aircraft is most suitable to reject non-uniformly distributed gust. The thrust changes are very low, but the wing flap deflection of the third aircraft is very high. Because of the low speed operation of the aircraft such a deflection should be acceptable.

In the nonlinear simulation, the results with respect to the shaped sensitivity function are confirmed. All controller work as required and fulfill the requirements. Wing shape holding is achieved in gust and maneuvers. The airspeed does not drop below the minimum value. Non-uniform distributed gusts are rejected by the inner-loop, which is a big advantage in comparison to state of the art HALE aircraft [17].

3.3 Outer-Loop Control Law

The outer-loop design is based on the single input, single output description between the input variable and the control variable. Since pitch angle control is already established by the inner-loop, the outer-loop relating to flight control has only to cover the functionality of bank angle control. For turned flight, at first a washout filter is integrated in the inner-loop. Therefore, a transfer function with

$$G_{\mathrm{Washout}} = \frac{T_{\mathrm{D}} s}{T_{\mathrm{D}} s + 1} \tag{33}$$

is selected with $T_{\mathrm{D}} = 0.2$ s. This value is based on the Dutch roll frequency with inner-loop. The integration of this filter destabilizes. This change is expected since the spiral mode has to be unstable for turned flight. The left $\Theta_{\mathrm{left,\,cmd}}$ and right $\Theta_{\mathrm{right,\,cmd}}$ pitch angle command to the inner-loops represent an overlay of the pitch angle command Θ_{cmd} and the signal caused by the bank angle controller ξ_{bank} with

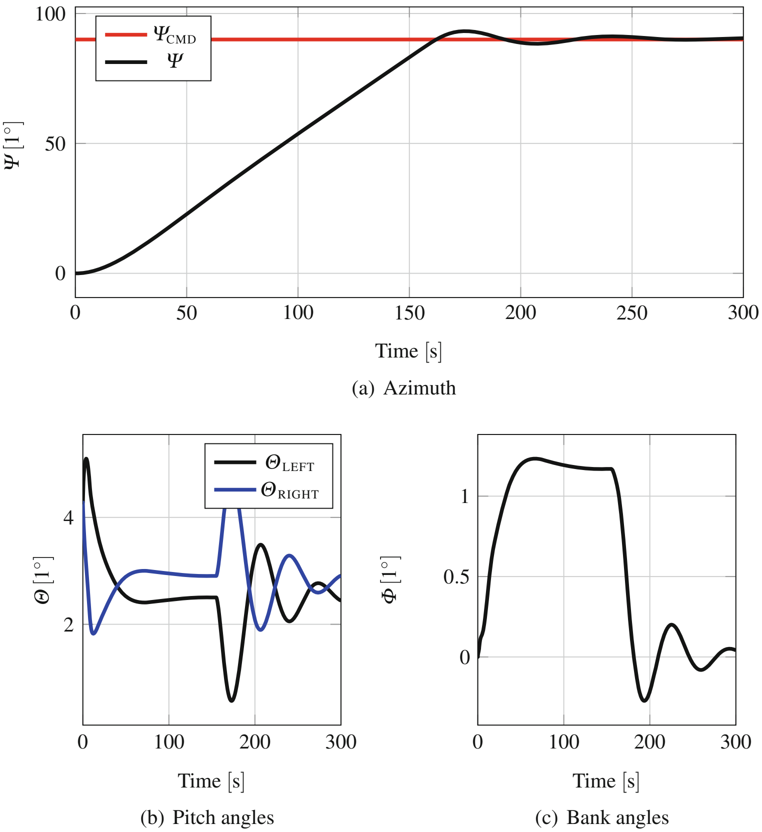
$$\begin{aligned} \Theta_{\mathrm{left,\,cmd}} &= \Theta_{\mathrm{cmd}} + \xi_{\mathrm{bank}} \\ \Theta_{\mathrm{right,\,cmd}} &= \Theta_{\mathrm{cmd}} - \xi_{\mathrm{bank}} \end{aligned} . \tag{34}$$

The bank angle controller is designed with the root locus method [2]. The transfer function from ξ_{bank} to the bank angle is used. Therefore, the system is reduced by the balanced truncation to a second order system. The corresponding control law is

$$\xi_{\mathrm{bank}} = k_{\xi\Phi} \left(\Phi_{\mathrm{cmd}} - \Phi_{\mathrm{AC5}} \right) \tag{35}$$

and the proportional gain is determined with $k_{\xi\Phi} = 2$. With this approach, pitch as well as bank angle are controlled for the MBA and the outer-loop relating to flight control is designed.

Regarding the flight control outer-loop, the pitch angle Θ_{cmd} and the bank angle Φ_{cmd} are available as input values and the altitude H and the azimuth Ψ have to be controlled by the flight guidance outer-loop. This classical flight control principle is also applied to conventional aircraft [2]. For both values, a proportional control law with



(a) Azimuth

(b) Pitch angles

(c) Bank angles

Fig. 9 Nonlinear simulation results of flight guidance outer-loop with a commanded azimuth of 90°

$$\begin{aligned} \Theta_{\mathrm{cmd}} &= k_{\Theta H} \underbrace{(H_{\mathrm{cmd}} - H)}_{e_H} \quad \text{and} \\ \Phi_{\mathrm{cmd}} &= k_{\Phi\Psi} \underbrace{(\Psi_{\mathrm{cmd}} - \Psi)}_{e_\Psi} \end{aligned} \tag{36}$$

is used. The maximum commanded pitch angle is limited to 2.0° for a lower required thrust. Having an altitude error of 5 m, the gain $k_{\Theta H}$ commands 2.0° to the flight control inner-loop. The bank angle command is bounded to 1° and $k_{\Phi\Psi} = 0.25$ is used for the azimuth control loop. The nonlinear simulation results for a step command in the azimuth angle of 95° are illustrated in Fig. 9. The MBA needs approximately 160 s until the demand value is reached. This corresponds to a yaw rate of 0.56° s^{-1}. A small overshoot occurs. The left and right side pitch angle react adversely to each other. In the intercept/capture phase of the desired azimuth, some low frequent and well damped oscillations occur. In summary, the proportional gain controller is able to achieve fast tracking of an azimuth.

4 Conclusion and Outlook

In this paper, a flight path flight control law for a Multi-Body HALE aircraft was presented. This aircraft is able to operate continuously one year in the mission altitude. Kane's formalism was used to assemble the equations of motion for the Multi-Body aircraft. The uncontrolled aircraft shows a nominal unstable behavior with 10 right hand side poles.

The developed flight control law bases on the classical cascade flight control principle. Since the flight dynamics of the MBA strongly differ to classical aircraft, a new design method and structure for the inner-loop had to be applied. Therefore, the H_∞ loop shaping in the frequency domain was successfully used. Besides a clear and unique definition of the requirements, the method allows the use of uncertainties to take non-modeled flight dynamics into account. With the design method, a robust inner-loop was designed that successfully stabilizes the plant and achieves wing shape holding, airspeed, left and right side pitch control as well as gust rejection. Especially the results of non-uniform distributed gust load rejection underline the benefit of a Multi-Body Hale aircraft in comparison to a one wing HALE aircraft. Based on the inner-loop flight control law, classical outer-loop flight control laws for azimuth angle and altitude control were applied. The formation follows the azimuth angle very fast. That allows surveillance and broadcast missions over a certain area.

Alongside the benefits of the design method for the inner-loop, the order of the designed inner-loop is very high (61). In the case of a classical aircraft, this order would be too high, but the Multi-Body aircraft concept offers new possibilities to handle the computational effort. Calculation routines can be distributed among many computers and this allows a fast computation even for high order control laws. A suitable concept has to be discussed in further investigations. The functionality of the control laws (especially for wing shape holding) shall be proven in a flight test campaign with a small Multi-Body aircraft demonstrator that is already constructed and possesses the same degrees of freedom like the HALE Multi-Body aircraft.

References

1. Armellin R et al (2010) Nonlinear mapping of uncertainties: a differential algebraic approach. In: 4th international conference on astrodynamics tools and techniques, ESAC, Spain
2. Brockhaus R, Alles W, Luckner R (2013) Flugregelung, vol 3. Springer, Berlin
3. European aviation safety agency, certification specifications and acceptable means of compliance for normal, utility, aerobatic, and commuter category aeroplanes (CS-23), amendment 4, 2015
4. European aviation safety agency, certification specifications and acceptable means of compliance for large aeroplanes (CS-25), Amendment 18, 2016
5. Gu DW, Petkov PH, Konstantinov MM (2005) Robust control design with MATLAB®. Springer Science & Business Media, Berlin
6. Katz J, Plotkin A (2001) Low-speed aerodynamics. Cambridge University Press, Cambridge
7. Kane, Thomas R, Levinson, David A, (1985) Dynamics, theory and applications. McGraw Hill, New York

8. Köthe A, Luckner R (2015) Flight mechanical modeling and analysis of Multi-Body aircraft. In: International forum on aeroelasticity and structural dynamics (IFASD). St. Petersburg, Russia
9. Lee B, Poomin P, Chuntaek K (2014) Power managements of a hybrid electric propulsion system powered by solar cells, fuel cells, and batteries for UAVs. In: Handbook of unmanned aerial vehicles. Springer, Berlin
10. Levinson, DA, Kane, TR (1990) AUTOLEV - a new approach to multibody dynamics. In: Multibody systems handbook. Springer, Berlin
11. Marsh G (2010) Best endurance under the sun. In: Renewable energy focus 11.5, Oxford
12. Noll TE et al (2004) Investigation of the Helios prototype aircraft mishap volume I mishap report. National aeronautics and space administration
13. Niewoehner RJ, Isaac IK (1994) Design of an autoland controller for a carrier-based F-14 aircraft using H_∞ output-feedback synthesis. In: American control conference, vol 3. IEEE, New York
14. Phillips WF (2004) Mechanics of flight. Wiley, New York
15. Skogestad S, Postlethwaite I (2007) Multivariable feedback control: analysis and design, vol 2. Wiley, New York
16. Tucker MR, Walker DJ (1997) H_∞ mixed sensitivity. In: Robust flight control. Springer, Berlin
17. Su W, Cesnik CE (2011) Dynamic response of highly flexible flying wings. AIAA J 49(2)
18. Wright JR, Cooper JE (2008) Introduction to aircraft aeroelasticity and loads. Wiley, United kingdom

Task Allocation of Multiple UAVs for Cooperative Parcel Delivery

Gyeongtaek Oh, Youdan Kim, Jaemyung Ahn and Han-Lim Choi

1 Introduction

As the ability of unmanned aerial vehicles (UAVs) has been improved, the demand of the UAV mission is also increased. Especially, the aerial transport capability of UAVs has become the primary interest in delivery business. For instance, Amazon shared video clips describing the future delivery system with its prototype UAV, called Prime Air. DHL tested its tilt-wing UAV, called Parcelcopter 3.0, during three months in the alpine condition. As shown in the aggressive developments by major companies, it is expected that UAVs will take important roles in the future delivery system.

In general, the maximum payload of a UAV is relatively smaller than that of the ground transport such as a truck or train which can carry a number of parcels having different payloads. If a UAV is permitted to carry only one parcel at a time, then a variety of UAVs should be developed to satisfy the requirements on the delivery of various payload, which will need a massive cost for development and maintenance. From this point of view, a UAV that can carry a heavy item with its neighboring UAVs can reduce the cost.

G. Oh · Y. Kim (✉)
Department of Mechanical and Aerospace Engineering,
Seoul National University, Seoul 08826, South Korea
e-mail: ydkim@snu.ac.kr

G. Oh
e-mail: laimo@snu.ac.kr

J. Ahn · H.-L. Choi
Department of Aerospace Engineering, Korea Advanced Institute of Science
and Technology, Daejeon 34141, South Korea
e-mail: jaemyung.ahn@kaist.ac.kr

H.-L. Choi
e-mail: hanlimc@kaist.ac.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_24

Several research on cooperative transport [1, 13, 16–18] have focused on the motion control of multiple robots, where a single product is assigned to the robots. The aforementioned research, however, did not address about a scheduling issue in real-world implementation, where multiple parcels should be delivered to different drop-off points. In [4, 7], the scheduling problem was considered for a truck-UAV cooperation, however, a parcel that should be carried by multiple UAVs was not considered.

For multiple parcels case, which cannot be carried by a single UAV, the efficiency of the mission depends on a decision of team members and delivery orders. According to the well-known taxonomy for task allocation (TA) [5], this type of problem is called the single-task robots (ST)–multi-robot tasks (MR) problem, where ST means that each robot is capable of executing at most one task at a time and MR indicates that each task requires multiple robots for its completion. The ST-MR problem is often referred to as a *coalition formation* problem, where a coalition is a group of agents that conduct a common task [10].

Sandholm and Lesser [11] stated that the vehicle routing problem, combined with the coalition formation problem, is too hard to solve optimally. Shehory and Kraus [12] adopted a greedy policy, which precedes the higher valued coalition's task. Oh et al. [9] proposed the meta-heuristic solver of the ST-MR problem using particle swarm optimization. Distributed coalition formation algorithms for a multi-robot system have also been proposed [8, 14, 15]. However, these studies concentrated on the type of task that is completed when the fleet of UAVs visits the task location, whereas different pick-up point and drop-off point are designated to each task in the cooperative parcel delivery problem.

In this study, a cooperative parcel delivery problem using multiple UAVs is defined as an integer programming problem. The visiting orders of UAVs are constrained by a graph theoretic notion, which excludes infeasible solutions. To obtain the suboptimal solution of the problem with less computation time, the improved sequential greedy algorithm (SGA) is proposed. Numerical simulation demonstrates that the performance of the proposed algorithm is better than that of the traditional SGA.

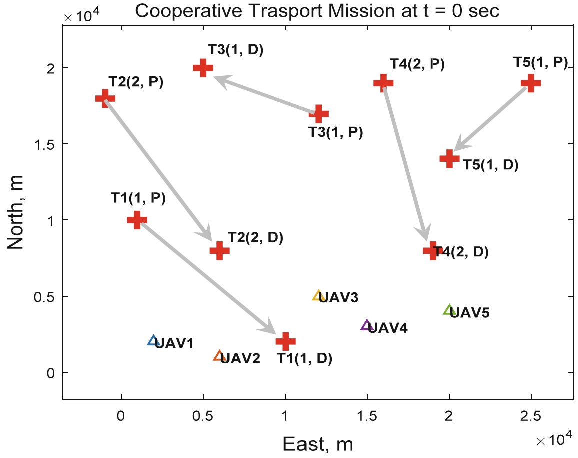
This paper is organized as follows. Section 2 defines the cooperative parcel delivery problem. The traditional SGA and the proposed SGA are provided in Sect. 3. Numerical result and analysis are presented in Sect. 4. Finally, Sect. 5 discusses the conclusion.

2 Problem Statement

Let us consider N agents and M delivery tasks as shown in Fig. 1. Agent $i \in \{1, 2, \ldots, N\}$ is defined as its initial position $\mathbf{u}_i$, average speed v_i, payload c_i, time for parcel pick-up t_i^p, and time for parcel drop-off t_i^d.

In this study, homogeneous agents are considered and it is assumed that $v_i = V$, $t_i^p = t_p$ and $t_i^d = t_d$. Task $k \in \{1, 2, \ldots, M\}$ is defined as its pick-up location $\mathbf{y}_k^p$, drop-off location $\mathbf{y}_k^d$, weight of the parcel w_k, and the time window for delivery

Fig. 1 Mission environment



$[t_k^m, t_k^M]$. In Fig. 1, P and D denote the pick-up and drop-off points, respectively, and the number inside the parenthesis means that the required number of agents, which is determined by the payload of agents and the weight of the parcel. In this setting, the cooperative parcel delivery problem can be formulated as an integer programming problem as follows.

$$\underset{\mathbf{P}}{\text{Maximize}} \;\; J = \sum_{k=1}^{M} s_k(t_k(\mathbf{P}), t_k^m, t_k^M) \tag{1}$$

$$\text{subject to} \quad \sum_{i \in \mathbf{a}_k(\mathbf{P})} c_i \geq w_k, \quad \forall k \in \mathbb{K} \tag{2}$$

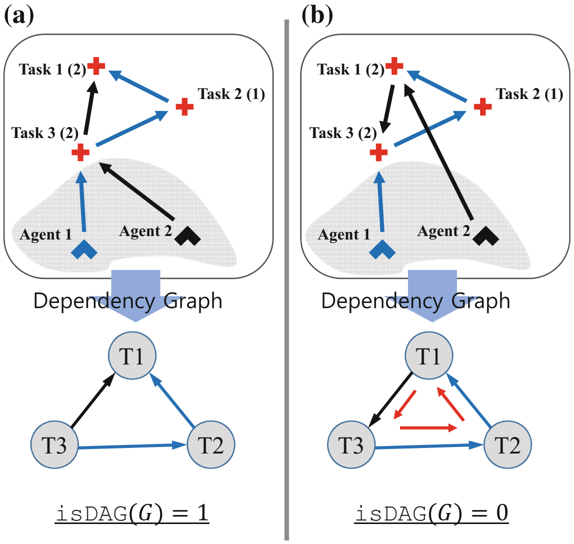
$$\mathtt{isDAG}(G) = 1, \quad G = (\mathbb{K}, \mathscr{E}(\mathbf{P})) \tag{3}$$

where $\mathbb{K} \triangleq \{1, 2, \ldots, M\}$ is the index set of the tasks, and the optimization variable $\mathbf{P} = (p_{i,m})$ denotes the visiting orders of all agents. For instance, $p_{i,m} = k$ means that task k is the mth task of the agent i; $p_{i,m} = 0$ denotes that no task is allocated to the agent i as the mth order. The objective function is set as the summation of all scores, where score s_k for the task k is the function of the termination time t_k of the task k and its time window. Equation (2) constrains that the total payloads of a coalition should be greater than the weight of the target task, where $\mathbf{a}_k$ is a coalition set containing the indices of agents assigned to the task k.

Note that (3) is introduced to allow simultaneous arrival of agents for carrying a heavy item together. The dependency graph G describes the resultant precedence between the tasks [2]. The directed edge set $\mathscr{E}(\mathbf{P})$ is defined as follows.

$$\mathscr{E}(\mathbf{P}) = \{(p_{i,j}, p_{i,j+1}) | i \in \mathbb{I}, \; j \in \{1, ..., \mathtt{card}(\mathbf{p}_i) - 1\}\} \tag{4}$$

Fig. 2 Illustrative examples of the DAG constraint



where $\texttt{card}(\cdot)$ denotes the number of nonzero elements in a set (or a vector), and $\mathbf{p}_i$ is the ith row of $\mathbf{P}$. The simultaneous arrival fails when the dependency graph contains a directed cycle. Therefore, designating the type of the dependency graph as a directed acyclic graph (DAG), which is a directed graph with no directed cycles [6], plays a role of the filtering. The function $\texttt{isDAG}(G)$ is one if the graph G is DAG and zero otherwise. Two illustrative examples are shown in Fig. 2, where simultaneous arrivals are possible in (a) whereas it is impossible in (b).

On the other hand, the objective function should be defined such that agents can finish many tasks within time windows. Thus, in this study, the score s_k is defined as follows.

$$s_k = \begin{cases} s_0 + \dfrac{t_k^M - t_k}{t_k^M - t_k^m}, & \text{if } t_k^m \le t_k \le t_k^M \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where s_0 is the default reward of success, and t_k can be expressed as follows.

$$t_k\left(\mathbf{P}\right) = \left(\max_{i \in \mathbf{a}_k} t_{ETA}\left(i, k\right)\right) + t_p + d_k/V + t_d \tag{6}$$

where $d_k (= ||\mathbf{y}_k^p - \mathbf{y}_k^d||)$ is the distance between pick-up point and drop-off point of the task k, and $t_{ETA}(i, k)$ is the estimated time of arrival (ETA) of the agent i for pick-up point of the task k. Then, the first term of the right hand side is the meeting time of the coalition at the pick-up point of the task k, and the other terms are the required time for delivery.

TA problem defined in (1)–(3) is a overlapping coalition formation problem combined with a vehicle routing problem. A number of possible coalitions and visiting orders make this integer programming problem be intractable, thus considerable computation time is required to obtain an exact optimal solution. In this study, heuristic approaches are considered to obtain suboptimal solution with less computation time.

3 Task Allocation Algorithm

In this section, modified SGA is proposed after brief introduction of traditional SGA.

3.1 Agent-Based Sequential Greedy Algorithm

In this section, a traditional SGA [9] is briefly summarized. The SGA matches a capable coalition with its corresponding task repeatedly in a short-term perspective. There exist many possible coalitions, and therefore the SGA selects a coalition leader by calculating the ETA between all agents and tasks. After the coalition leader and the corresponding task are determined, coalition members having less ETA for the target task are selected. This procedure is repeated until the whole tasks are allocated. To distinguish between the traditional SGA and the proposed SGA, the former is called agent-based SGA (ASGA) in this study.

The merit of the ASGA is that the DAG constraint is automatically satisfied, because newly allocated task is augmented at the end of the current sequence of tasks, thereby a directed cycle cannot appear. Also, computational burden can be significantly reduced compared to the exact algorithm. Therefore, the ASGA may be a possible choice for practical application of the cooperative timing mission. The detailed procedure of the ASGA is summarized in Algorithm 1, where $\mathbf{p}^{(i)} \oplus_{end} \{k\}$ denotes that the task k is augmented at the end of the agent i's path vector $\mathbf{p}^{(i)}$.

3.2 Task-Based Sequential Greedy Algorithm

In ASGA, two greedy decisions are made for each matching between a task and its coalition: (i) a target task and its corresponding coalition leader are determined concurrently by greedy sense, and then (ii) several agents are selected as the coalition members by greedy sense. In the cooperative parcel delivery problem, however, the first decision is much more important than the second one because task execution order is fixed by the first one. In addition, t_k of the problem depends on the latest member, which results in the small influence of the second decision. Based on this consideration, the task-based sequential greedy algorithm (TSGA) is proposed.

Algorithm 1 Sequential Greedy Algorithm

```
procedure SGA($\mathbb{I}$, $\mathbb{K}$)
    $\mathbb{K}_0 = \mathbb{K}$
    for iter=1:$n(\mathbb{K})$ do
        $(i^*, k^*) = \mathrm{argmin}_{(i,k)\in\mathbb{I}\times\mathbb{K}_0}\ t_{ETA}(i, k)$
        $\mathbf{p}^{(i^*)} = \mathbf{p}^{(i^*)} \oplus_{end} \{k^*\}$
        $\mathbb{I}_0 = \mathbb{I} \setminus \{i^*\}$
        for z=1:$(Z_{k^*} - 1)$ do
            $j^* = \mathrm{argmin}_{j\in\mathbb{I}_0}\ t_{ETA}(j, k^*)$
            $\mathbb{I}_0 = \mathbb{I}_0 \setminus \{j^*\}$
            $\mathbf{p}^{(j^*)} = \mathbf{p}^{(j^*)} \oplus_{end} \{k^*\}$
        end for
        $\mathbb{K}_0 = \mathbb{K}_0 \setminus \{k^*\}$
    end for
end procedure
```

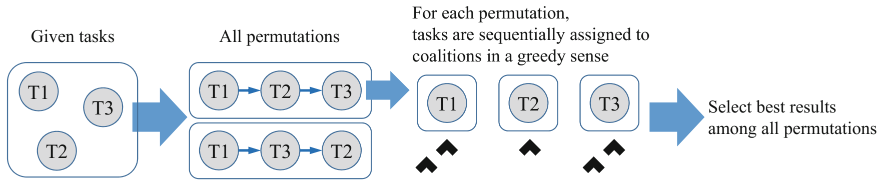


Fig. 3 Overall procedure of TSGA

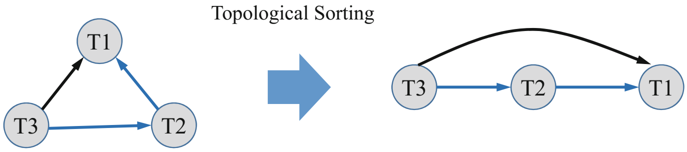


Fig. 4 Example of topological sorting

The fundamental idea of the TSGA is to replace the important decision of the AGSA with the process of exact algorithm, which means that all possible task execution orders are investigated as a candidate in the TSGA. For instance, $M!$ task execution orders are considered when M tasks are given. Coalition members including a leader for each task are determined by a greedy algorithm. Among the resultant objective functions for each task execution order, the maximum-valued case is selected as the solution of the TSGA. Overall procedure of the TSGA is summarized in Fig. 3.

The solution of the TSGA naturally satisfies (3). It is known that a graph can be topologically sorted if and only if the graph is DAG [3]. The topological sorting of a directed graph G is a linear ordering of all the vertices; if G contains an edge (u, v), then u appears before v in the order. The illustrative example of topological sorting of dependency graph G is shown in Fig. 4. Because the TSGA considers all permutations of task execution order, all feasible domain of (3) can be investigated by the TSGA.

Note that the TSGA is not a polynomial algorithm, it may require much computation time for big problems with large N and M. For problems with moderate size of N and M, however, the computation time will not be an issue because the computational effort for t_{ETA} is light. In addition, the solution of the TSGA is always better than or at least same as that of the ASGA because candidates of the task execution order of the TSGA includes the task execution order determined by the ASGA.

4 Numerical Simulation

Numerical simulation is conducted to demonstrate the performance of the proposed task allocation algorithm for cooperative parcel delivery. Simulation is performed using a desktop computer with an Intel Core i5-4670 @ 3.40 GHz with 16 GB RAM and MATLAB on a Windows 7 operating system. Five UAVs and five tasks in 2-dimensional simulation environment are considered as shown in Fig. 1. It is assumed that average speed of UAV is 20 m/s, pick-up and drop-off time $t_p = t_d = 10$ s, payload $c_i = 1$ kg, default score for delivery s_0 =1, and weights of parcels are randomly selected. Also, collisions between UAVs are assumed to be autonomously avoided by vertical separation or formation flight. For the first scenario, the time windows of tasks are set as $[t_k^m, t_k^M]$ = [0, 1 h]. The TA results of ASGA and TSGA are shown in Fig. 5 and summarized in Table 1.

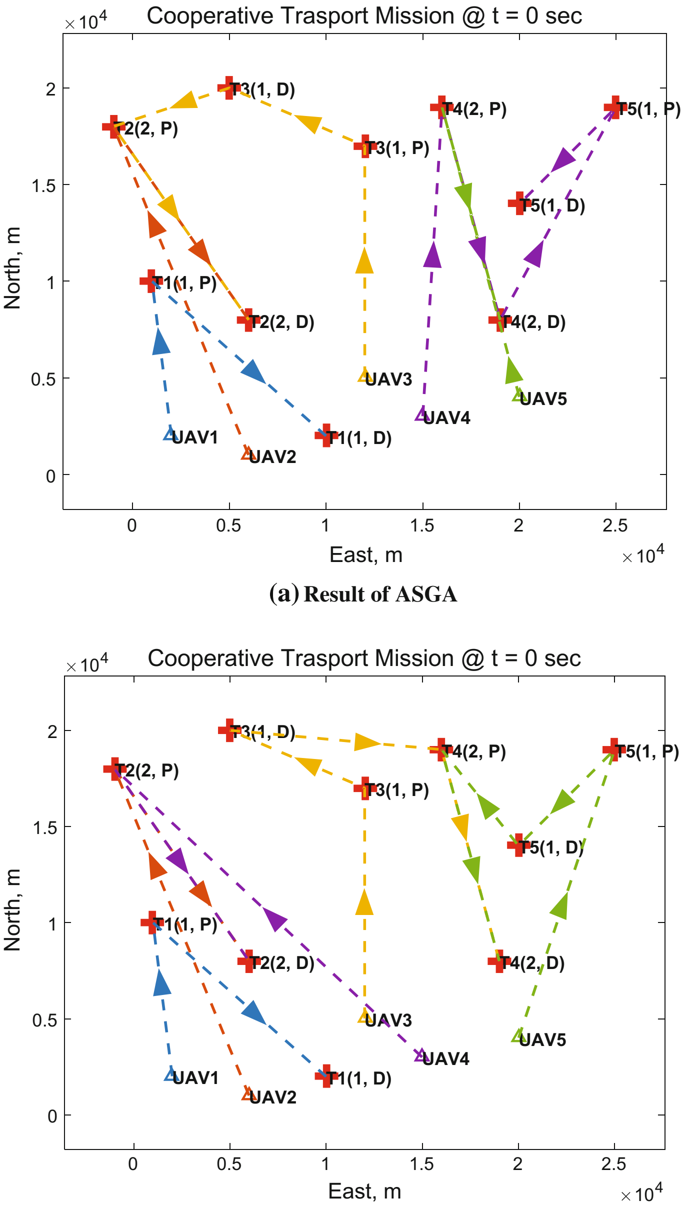
In Fig. 5a, UAV2 waits for UAV3 at the pick-up point of T2, which is not efficient. On the other hand, as shown in Fig. 5b, waiting time in T2 is reduced because UAV4 is assigned to T2 instead of UAV3. Also, UAV3 and UAV5 meet at T4 after finishing their tasks T3 and T5, which can be carried by a single UAV. As a result, the objective function and mission completion time of the TSGA are better than those of the ASGA. The runtime of the TSGA is greater than the ASGA, but the gap is less than 0.1 s.

Let us consider another scenario, where time windows of tasks are set as $[t_k^m, t_k^M]$ = [0, 30 min]. This scenario may be helpful to pizza company who has multiple stores and delivery orders from different locations. The TA results of ASGA and TSGA are shown in Fig. 6 and summarized in Table 2.

In this scenario, the resultant path of the ASGA is same as that of previous scenario, which reveals the limitation of the ASGA. Because task execution order is determined in a sequential manner without considering an objective function, the resultant TA

Table 1 TA results when $[t_k^m, t_k^M]$ = [0, 1 h]

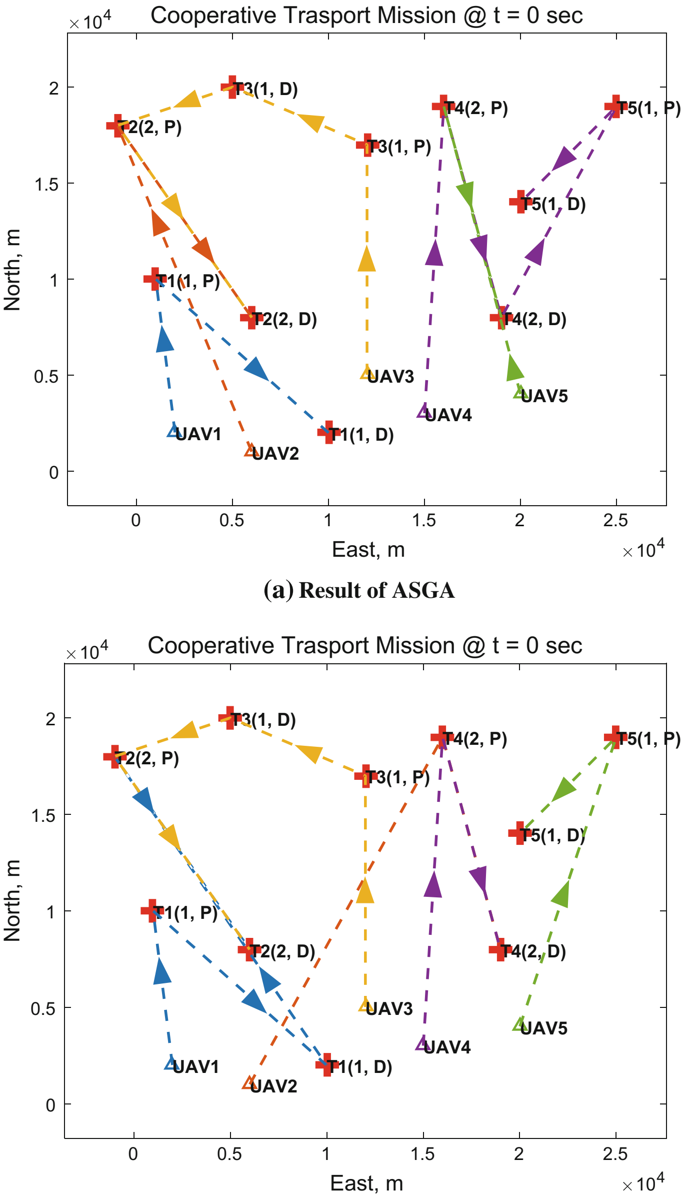
	ASGA	TSGA
Objective function	7.8454	8.0388
Mission completion time (s)	2391.7	2143.1
Runtime (s)	0.027	0.095



(a) **Result of ASGA**

(b) **Result of TSGA**

Fig. 5 TA results with time window $[t_k^m, t_k^M] = [0, 1\,\mathrm{h}]$



(a) Result of ASGA

(b) Result of TSGA

Fig. 6 TA results with time window $[t_k^m, t_k^M] = [0, 30\,\text{min}]$

Table 2 TA results when $[t_k^m, t_k^M] = [0, 30\,\text{min}]$

	ASGA	TSGA
Objective function	4.1013	5.3279
Mission completion time (s)	2391.7	2626.3
Runtime (s)	0.031	0.096

result cannot handle the penalty from the violation of time windows. In TSGA, on the other hand, all possible task execution orders are investigated and the best scored one is selected as a solution, where the violation effect of time windows is considered during calculating score (5). Therefore, the solution of the TSGA in Fig. 6b is different from that of Fig. 5b. Considering that the mission completion time in Tables 1 and 2 is greater than 30 min, it may be thought that the given time window is a harsh constraint. Indeed, the ASGA failed to deliver the parcels of T2 and T5 within 30 min whereas the TSGA allows T5 (forgives T2) to maximize the objective function by completing the other four tasks. As a result, the objective function of the TSGA is better than that of the ASGA while the mission completion time is slower than the ASGA. Especially, the degradation of objective function due to shortened time windows in the ASGA (7.8 → 4.1) is more severe than that in the TSGA (8.0 → 5.3).

Because the objective function is defined as the summation of score functions, not the mission completion time, the TSGA prefers the solution having high objective function than short mission completion time. When the time windows of all tasks are loose, the objective function is expected to be aligned with the mission completion time due to the low possibility of time windows violation. However, when time windows of all tasks are tight, the TSGA may select the longer route to complete more tasks within time windows.

To see the effect of calculation time and objective function for more agents, additional experiments are conducted with $3 \le N \le 8$ and $M = 3$. The runtime and total score for increasing N does not change eventually as shown in Fig. 7. As a result, it can be stated that the number of agents N is a minor factor for runtime and objective function with constant M. Likewise, to see the effect of calculation time and objective function for more tasks, further experiments are performed with $3 \le M \le 8$ and $N = 3$. As shown in Fig. 8, the gap of the objective function between ASGA and TSGA grows for more tasks, which comes from that TSGA makes better decision than ASGA when the given time is insufficient. The runtime of TSGA, however, shows exponential growth according to M.

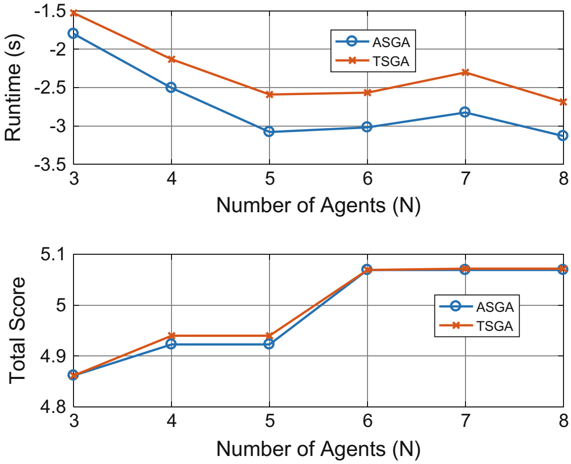


Fig. 7 Effect of N when $M = 3$ and time window $[t_k^m, t_k^M]$ = [0, 1 h]

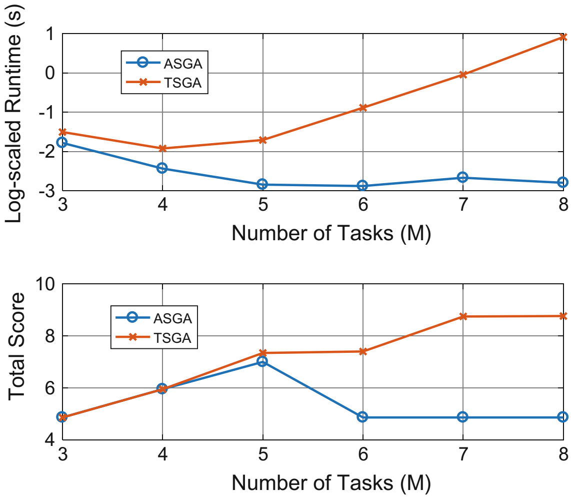


Fig. 8 Effect of M when $N = 3$ and time window $[t_k^m, t_k^M]$ = [0, 1 h]

5 Conclusions

In this study, the cooperative parcel delivery problem using multiple UAVs was formulated in the form of an integer programming problem. Based on the analysis of problem characteristics, the modified sequential greedy algorithm was proposed. Through the numerical simulation, the proposed scheme showed better performance than the traditional method by consuming more computation time. In real-world application, the adequate solver should be determined by considering the problem size and the expected time to solve.

References

1. Aiyama Y, Hara M, Yabuki T, Ota J, Arai T (1999) Cooperative transportation by two four-legged robots with implicit communication. Robot Auton Syst 29(1):13–19
2. Balmas F (2004) Displaying dependence graphs: a hierarchical approach. J Softw Maint Evol Res Pract 16(3):151–185
3. Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms. MIT University Press, Cambridge
4. Ferrandez SM, Harbison T, Weber T, Sturges R, Rich R (2016) Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J Ind Eng Manag 9(2):374–388
5. Gerkey BP, Matarić MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. Int J Robot Res 23(9):939–954
6. Gross JL, Yellen J (2003) Handbook of graph theory. CRC Press, Boca Raton
7. Murray CC, Chu AG (2015) The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp Res Part C: Emerg Technol 54:86–109
8. Oh G, Kim Y, Ahn J, Choi HL (2015) Market-based task assignment for cooperative timing missions over networks with limited connectivity. In: AIAA Guidance, navigation, and control conference, Kissimmee, FL
9. Oh G, Kim Y, Ahn J, Choi HL (2016) PSO-based optimal task allocation for cooperative timing missions. In: 20th IFAC symposium on automatic control in aerospace, Sherbrooke, Canada
10. Sandholm TW, Lesser VR (1995) Coalition formation among bounded rational agents. In: International joint conference on artificial intelligence, Quebec, Canada
11. Sandhlom TW, Lesser VR (1997) Coalitions among computationally bounded agents. Artif Intell 94(1):99–137
12. Shehory O, Kraus S (1998) Methods for task allocation via agent coalition formation. Artif Intell 101(1):165–200
13. Stouten B, de Graaf AJ (2004) Cooperative transportation of a large object-development of an industrial application. In: IEEE international conference on robotics and automation, New Orleans, LA
14. Weerdt Md, Zhang Y, Klos T (2012) Multiagent task allocation in social networks. Autonom Ag Multi-Ag Syst 25(1):46–86
15. Whitten AK, Choi HL, Johnson LB, How JP (2011) Decentralized task allocation with coupled constraints in complex missions. In: IEEE American control conference, San Francisco, CA
16. Yamashita A, Fukuchi M, Ota J, Arai T, Asama H (2000) Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3D environment. In: IEEE international conference on robotics and automation, San Francisco, CA
17. Yang X, Watanabe K, Izumi K, Kiguchi K (2004) A decentralized control system for cooperative transportation by multiple non-holonomic mobile robots. Int J Control 77(10):949–963
18. Yufka A, Parlaktuna O, Ozkan M (2010) Formation-based cooperative transportation by a group of non-holonomic mobile robots. In: IEEE international conference on systems man and cybernetics, Istanbul, Turkey

Circumnavigation with Side-Bearing Angle

Sanghyuk Park

1 Introduction

Today, autonomous unmanned aerial vehicles (UAVs) are widely used. Typical applications include aerial observation, aerial surveillance, and aerial search, where the UAV is often required to orbit around a stationary target at a given distance [1, 2]. This type of motion is often called circumnavigation [3].

Several methods can be adopted for circumnavigation. Most existing works are based on a two-step approach: the first step is path planning [4–7], and the second step is path following [8–20]. In this approach, when the vehicle is initially far from the target point, a proper reference path should be first constructed so that the commanded path ultimately leads to the final circular path around the target. Then, the UAV is controlled to follow the commanded path at the guidance/control level.

There are other methods specialized for circumnavigation that are not based on the two-step approaches. A method using the vehicle position and the bearing angle to the target has been proposed in [2]. Another method using the range and range rate to the target has been proposed in [3, 21, 22]. The strength of this method is that it can be used in a situation where GPS is unavailable.

This work presents a guidance law that enables circumnavigation using the side-bearing angle. The side-bearing angle, in this work, is the deviation angle from the normal to the velocity direction. The analysis and application of the proposed guidance method demonstrate a number of benefits over other existing approaches. First, the proposed guidance law is expressed by a simple formula. Second, the guidance law is globally asymptotically stable. The simple guidance law maneuvers a vehicle to circle over a target with any initial position and heading configuration. In other

S. Park (✉)
Aerospace and Mechanical Engineering, Korea Aerospace University,
76 Hanggongdaehang-ro, Deogyang-gu, Goyang-si, Gyeonggi-do 412-791, Korea
e-mail: park@kau.ac.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_25

words, the conventional path planning in the two-step approach can be skipped. Third, the new method has only one control parameter. To facilitate the proper selection of the parameter, a linear analysis that is provided in this work can be used. Finally, the guidance law is simple and straightforward to apply in actual flight applications. In this work, two types of flight experiments are presented. The first experiment used a conventional GPS-based navigation system. The side-bearing angle was computed with the GPS position and velocity for a given target position. The flight test successfully demonstrated that the aircraft could circumnavigate a series of reference target points using the proposed method. The second flight experiment used a vision-based control system. This type of application can be used in a situation where GPS is not available. In particular, in this experiment, a fixed camera (as opposed to a gimbal camera) was attached to the test-bed UAV looking out at the side. Then, the pixel position of a target was used to compute the side-bearing angle. It was also successfully demonstrated that the aircraft can circumnavigate a series of target points using the vision-based system.

This paper is organized as follows. Section 2 describes the guidance law and provides a proof of the global asymptotic stability. Section 3 discusses the linear analysis, which can be used when choosing the control parameter. Section 4 provides the flight test result from using a GPS-based control system. Section 5 presents the flight test results from using a vision-based control system. Section 6 concludes this paper.

2 Guidance Law and Global Asymptotic Stability

In this section, the guidance law is developed using a Dubins vehicle model [4], which gives a good approximation for fixed-wing aircraft trajectories [23]. The kinematic model is given by

$$\dot{N} = V \cos \psi \tag{1}$$

$$\dot{E} = V \sin \psi \tag{2}$$

$$\dot{\psi} = \frac{a}{V} \tag{3}$$

where $N$and E are the coordinates of the vehicle's position; $V$is the velocity, which is assumed positive; $\psi$is the course angle of the velocity; and a is the normal acceleration, which is the control variable. The variables in models (1)–(3) are depicted in Fig. 1.

The target is stationary, and it is assumed that the side-bearing angle η in Fig. 2 is available. The objective of the guidance law is to guide the vehicle to orbit around the target with a given radius R_{ref} so that the side-bearing angle η converges to zero. The lateral acceleration command of the proposed guidance law in this work is

Fig. 1 Diagram for the guidance law

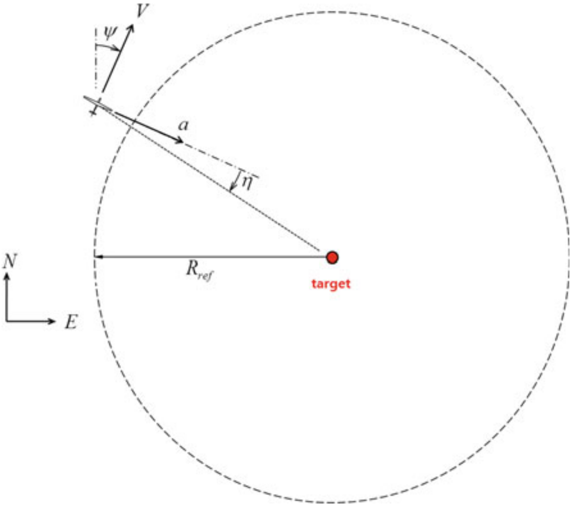
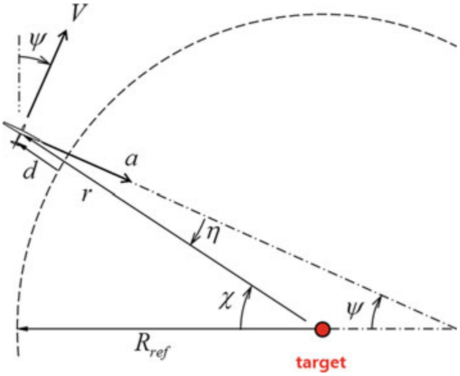


Fig. 2 System geometry for stability analysis



$$a = \frac{V^2}{R_{ref}} + K \sin \eta \tag{4}$$

where K is a design parameter.

Next, Lyapunov stability is presented and used to show the global asymptotic stability under the guidance law in Eq. 4. Figure 2 illustrates the system geometry with various other necessary variables to prove the stability. In the following development, all angles and angular rates are defined to be positive in the clockwise direction, as in Fig. 2, and it is assumed that the inner-loop control system exactly produces the commanded acceleration.

As shown in the figure, the three angles are related by $\eta + \psi = \chi$. Taking the derivatives, we have

$$\dot{\eta} = \dot{\chi} - \dot{\psi} \tag{5}$$

The velocity vector has a transverse component $r\dot{\chi} = V \cos \eta$, which can be rearranged as

$$\dot{\chi} = \frac{V \cos \eta}{r} \tag{6}$$

When the guidance law in Eq. 4 is applied to Eq. 3, we have

$$\dot{\psi} = \frac{V}{R_{ref}} + \frac{K}{V} \sin \eta \tag{7}$$

Next, considering the radial velocity component and substituting Eqs. 6 and 7 into Eq. 5 yield

$$\dot{r} = V \sin \eta \tag{8}$$

$$\dot{\eta} = \frac{V \cos \eta}{r} - \frac{V}{R_{ref}} - \frac{K}{V} \sin \eta \tag{9}$$

Equations 8 and 9 serve as the system equations with the two new state variables r and η. The stationary point of the system is found by setting $\dot{r} = 0$ and $\dot{\eta} = 0$; specifically,

$$r_0 = R_{ref} \text{ and } \eta_0 = 2\pi n (n = 0, \pm 1, \pm 2, \ldots) \tag{10}$$

Next, the global asymptotic stability is shown. Let a Lyapunov function candidate be

$$L = \frac{1}{2}\left(\frac{r}{R_{ref}} - \cos \eta\right)^2 + \frac{1}{2}\sin^2 \eta \tag{11}$$

which is positive definite. Taking the time derivative yields

$$\dot{L} = \left(\frac{r}{R_{ref}} - \cos \eta\right)\left(\frac{\dot{r}}{R_{ref}} + \dot{\eta} \sin \eta\right) + \dot{\eta} \sin \eta \cos \eta$$

Substituting $\dot{r}$ and $\dot{\eta}$ with the system dynamics in Eqs. 8 and 9 leads to

$$\dot{L} = \left(\frac{r}{R_{ref}} - \cos \eta\right)\left(\frac{V \sin \eta}{R_{ref}} + \frac{V \cos \eta \sin \eta}{r} - \frac{V \sin \eta}{R_{ref}} - \frac{K}{V} \sin \eta \sin \eta\right)$$
$$+ \left(\frac{V \cos \eta}{r} - \frac{V}{R_{ref}} - \frac{K}{V} \sin \eta\right) \sin \eta \cos \eta$$

$$
\begin{aligned}
&= \frac{V}{R_{ref}} \cos\eta \sin\eta - \frac{rK}{R_{ref}V} \sin^2\eta + \frac{K}{V} \sin^2\eta \cos\eta - \frac{V\cos^2\eta \sin\eta}{r} \\
&+ \frac{V\cos^2\eta \sin\eta}{r} - \frac{V}{R_{ref}} \sin\eta \cos\eta - \frac{K}{V} \sin^2\eta \cos\eta \\
&= -\frac{rK}{R_{ref}V} \sin^2\eta \le 0
\end{aligned} \tag{12}
$$

In other words, the time derivative of the Lyapunov function is negative semi-definite. According to LaSalle's theorem [24], as $t \to \infty$, the system approaches the largest invariant set inside a set of points defined by $\dot{L} = 0$, which is, in this case, the set $S = \{(r, \eta) | \eta = 2\pi n\}$, where $n = 0, \pm 1, \pm 2, \ldots$. In this set, we have $\dot{r} = 0$. Additionally, $\dot{\eta} \neq 0$ for all points except the points $(R_{ref}, 2\pi n)$. Thus, the largest invariant set inside the set S is the points $(R_{ref}, 2\pi n)$. Therefore, the global asymptotic stability is proved.

Simulation examples are presented to illustrate the stability. The vehicle is initially at the origin heading north. The target is placed at ($N = 600$ m, $E = 0$ m). Additionally, $V = 10$ m/s and $R_{ref} = 100$ m are used in the simulation. Figures 3, 4 and 5 show the simulation results for three different control gains: $K = 0.2$, $K = 1.0$, and $K = 2.0$. Each figure shows the vehicle trajectory and the side-bearing angle. The angle is given in the range between ±180°. When the gain is relatively small

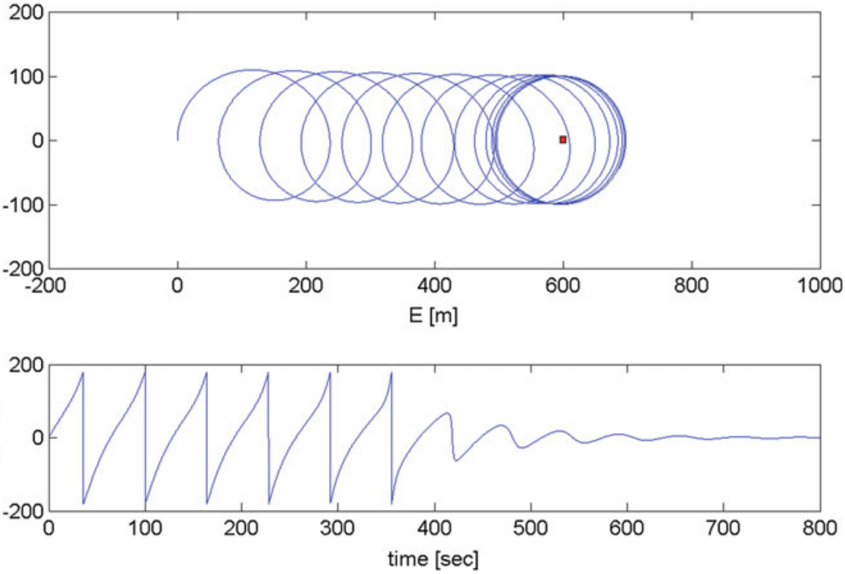


Fig. 3 Simulation with $K = 0.2$

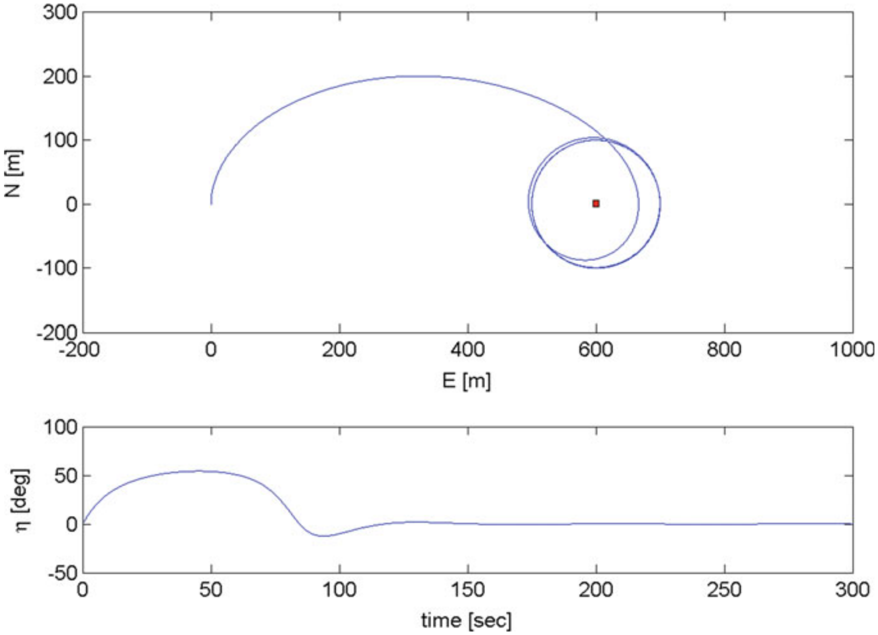


Fig. 4 Simulation with $K = 1.0$

($K = 0.2$), the trajectory makes several rounds of circular motion that drifts toward the target. It ultimately converges to the desired circle, as shown in Fig. 3. When the gain is selected at a moderate level ($K = 1.0$), as in Fig. 4, the trajectory converges to the desired circle rather directly without performing circular motion before reaching the target. When the gain is selected as the large value ($K = 2.0$), the vehicle approaches the desired circle following a spiral-like path from the outside by making a long detour, as indicated in Fig. 5. In all cases, however, the trajectory converges to the required circle around the target point. Additionally, the side-bearing angle converges to zero.

To further illustrate the global asymptotic stability, more simulations for various initial heading conditions are conducted. The results are summarized in Fig. 6. The vehicle starts at (0, 0) with initial heading conditions of 0, 20, 40, …, up to 340°. The control gain K is set to 1.0 in the simulations. In all cases with different initial headings, the trajectories converge to the desired circle.

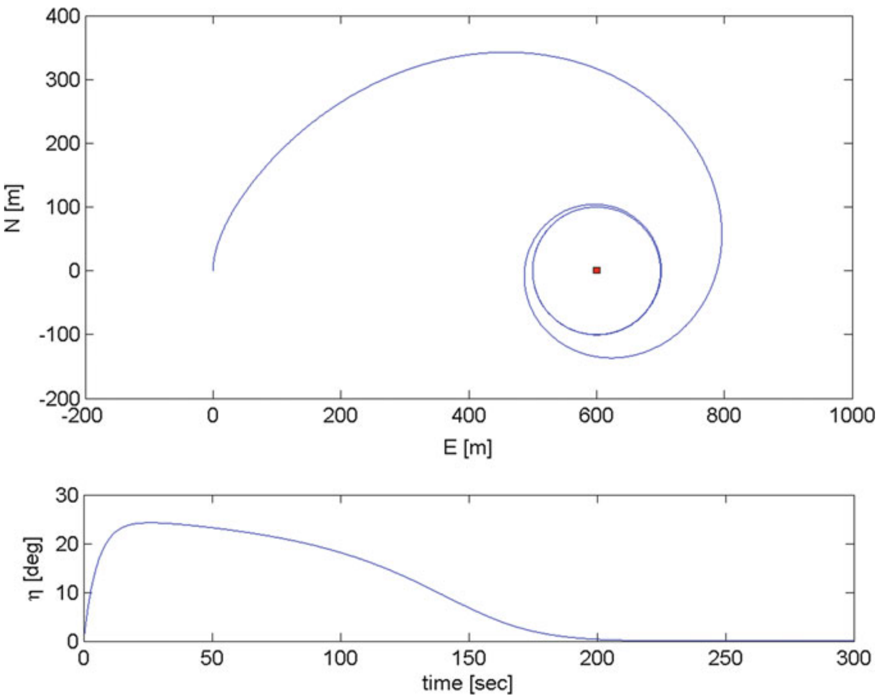


Fig. 5 Simulation with $K = 2.0$

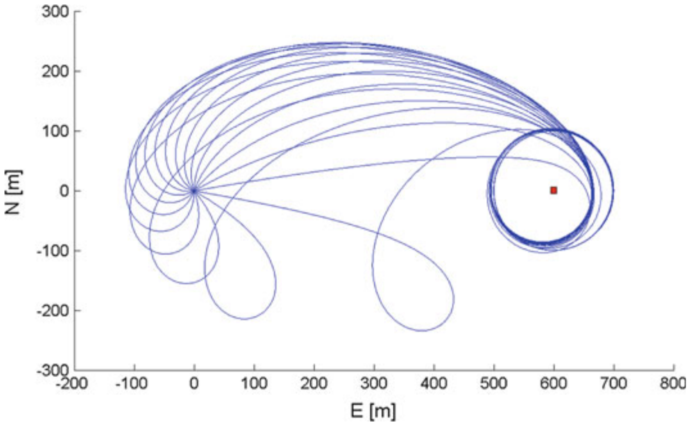


Fig. 6 Simulation with various initial headings ($\boldsymbol{K} = 1.0$)

3 Linear Analysis

This section presents a linear analysis of the system under the proposed guidance law. Through this analysis, a proper value of the control parameter may be determined. In the linear analysis, the perturbed dynamics around the equilibrium point are considered. Thus, it is assumed that η is a small angle. The cross-track error is also assumed to be small, i.e., $d << R_{ref}$. With these assumptions and recalling $r = R_{ref} + d$ in Fig. 2, the system equations are approximated as follows. First, Eq. 8 can be approximated as

$$\dot{d} = V\eta \tag{13}$$

Next, Eq. 9 can be approximated as

$$\begin{aligned}\dot{\eta} &= \tfrac{V\cos\eta}{r} - \tfrac{V}{R_{ref}} - \tfrac{K}{V}\sin\eta \approx \tfrac{V}{R_{ref}+d} - \tfrac{V}{R_{ref}} - \tfrac{K}{V}\eta \\ &= \tfrac{V}{R_{ref}\left(1+d/R_{ref}\right)} - \tfrac{V}{R_{ref}} - \tfrac{K}{V}\eta \approx \tfrac{V}{R_{ref}}\left(1-\tfrac{d}{R_{ref}}\right) - \tfrac{V}{R_{ref}} - \tfrac{K}{V}\eta \\ &= -\tfrac{V}{R_{ref}^2}d - \tfrac{K}{V}\eta\end{aligned} \tag{14}$$

Taking the derivative of Eq. 14 and applying Eq. 13 yield

$$\ddot{\eta} + \frac{K}{V}\dot{\eta} + \frac{V^2}{R_{ref}^2}\eta = 0 \tag{15}$$

Alternatively, taking the derivative of Eq. 13 and applying Eq. 14 lead to

$$\ddot{d} + \frac{K}{V}\dot{d} + \frac{V^2}{R_{ref}^2}d = 0 \tag{16}$$

Notice that Eqs. 15 and 16 are of the same form and are a well-known second-order linear system. The undamped natural frequency and the damping ratio are determined as follows:

$$\omega_n = \frac{V}{R_{ref}} \text{ and } \zeta = \frac{KR_{ref}}{2V^2} \tag{17}$$

This result provides a guideline for selecting the design parameters. First, the ratio of the vehicle speed and the radius of the desired circle determines the system frequency. Thus, for a given vehicle speed, the radius of the circle can be selected by considering any sensor delays and the inner-loop bandwidth in acceleration tracking. Once the vehicle speed and the radius of the circle are determined, the control gain K should be selected to provide a reasonable damping level.

In the previous simulation examples in Sect. 2, the case with $K = 0.2$ corresponds to a damping ratio of 0.1, the case with $K = 1.0$ corresponds to a damping ratio of

0.5, and the case with $K = 2.0$ corresponds to a damping ratio of 1.0. This explains the behavior patterns of the simulation results. When the gain is relatively small, the system exhibits oscillations, with many pre-circling turns before converging to a stationary point. On the other hand, when the gain is set relatively large, the system exhibits a well-damped behavior.

4 Flight Test with a GPS-Based Control System

The proposed guidance law was first applied to a GPS-based flight control system. The side-bearing angle is computed in this case using the GPS position and velocity along with the target position as follows.

In Fig. 7, $\vec{r}_A$ and $\vec{r}_T$ are the aircraft position vector and the target position vector, respectively. The relative position vector of the target with respect to the aircraft is $\vec{r}_{T/A} = \vec{r}_T - \vec{r}_A$. Let the coordinate of the relative position vector be

$$\left\{\vec{r}_{T/A}\right\} = \left\{\begin{matrix} r_{T/A_N} \\ r_{T/A_E} \end{matrix}\right\} \tag{18}$$

Next, let v_N and v_E be the velocity components in the north and east directions, respectively, and let V be the vehicle speed. Then, a unit vector normal to the velocity direction is obtained as

$$\{\hat{e}_n\} = \frac{1}{V}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\left\{\begin{matrix} v_N \\ v_E \end{matrix}\right\} \tag{19}$$

In this case, a clockwise direction is assumed as the orbiting direction. Next, applying the cross-product of the above two vectors yields

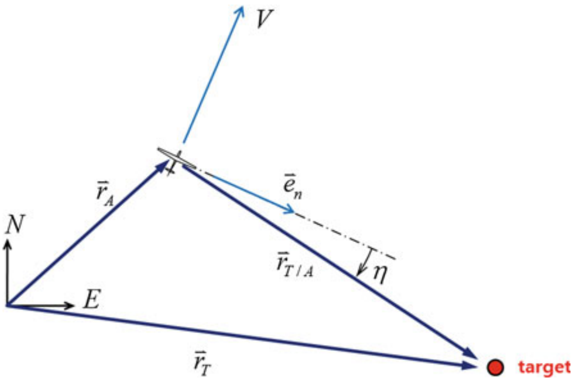


Fig. 7 Geometry used to compute the side-bearing angle with the GPS-based flight control system

Fig. 8 Test-bed UAV

$$\vec{e}_n \times \vec{r}_{T/A} = \left|\vec{e}_n\right| \left|\vec{r}_{T/A}\right| (\sin\eta)\,\hat{e}_D = \left|\vec{r}_{T/A}\right| (\sin\eta)\,\hat{e}_D \tag{20}$$

From this, we have

$$\sin\eta = \frac{e_{n_N} r_{T/A_E} - e_{n_E} r_{T/A_N}}{\sqrt{r^2_{T/A_N} + r^2_{T/A_E}}} \tag{21}$$

The above algorithm for the sin η computation and the proposed guidance law were implemented in flight tests with the test-bed aircraft shown in Fig. 8. The aircraft has a total mass of 2.0 kg and a wing span of 2.1 m. A typical flight speed is approximately 12 m/s. The aircraft is equipped with a flight control system. The onboard flight sensors include a GPS, an airspeed sensor, and an inertial sensor package with attitude, angular rate, and accelerometer outputs. There are several closed loops implemented in this aircraft: an elevator control loop is used to track altitude commands; a thrust control loop is implemented to track airspeed commands; and a rudder control loop is designed to improve the Dutch roll mode and to track roll commands.

When following an acceleration command using the guidance law, the acceleration command is converted to a roll attitude command using the well-known coordinated-turn aircraft kinematics, where the roll and sideways acceleration are related by $\phi = \tan^{-1}(a/g)$.

A series of flight tests was then performed. A set of data from one of the experiments is shown in Figs. 9 and 10. The recorded vehicle trajectory is shown in Fig. 9. The vehicle started at the initial position (N = −138 m, E = 135 m). The control gain K was set using an equivalent damping ratio of 0.6 according to Eq. 17, and the radius of the circle was set as 60 m. Then, the aircraft was commanded to circumnavigate a series of three target points defined in Table 1. They are marked with small circles in the plot. As indicated in this plot, the aircraft successfully orbited around the series

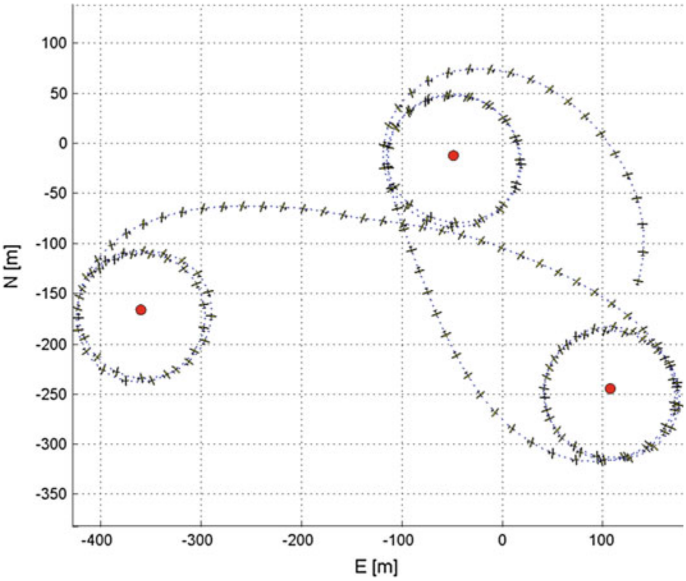


Fig. 9 GPS-based flight test result: Trajectory

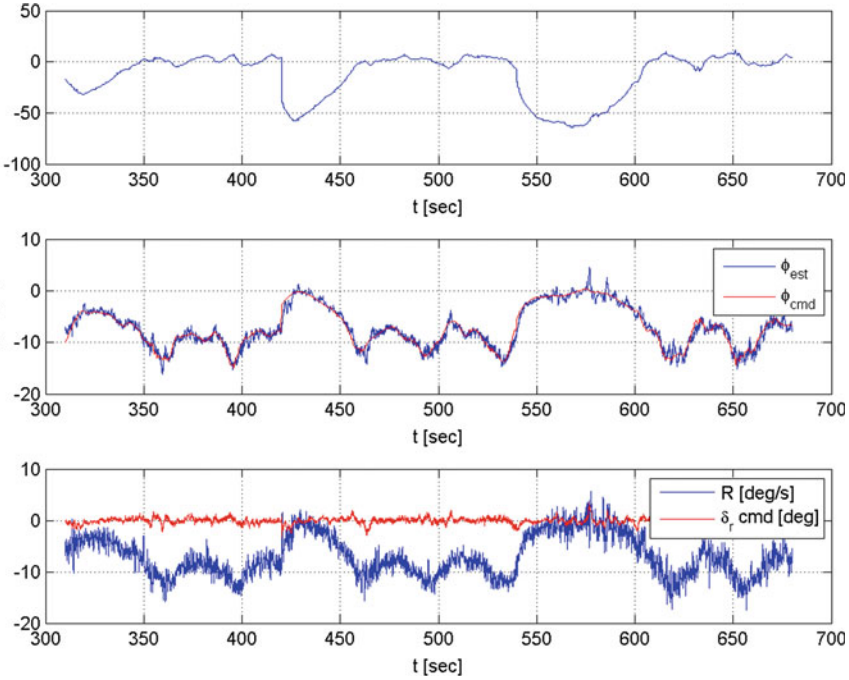


Fig. 10 GPS-based flight test result: side-bearing angle, roll angle

Table 1 Target locations in GPS-based flight experiment

Target	North [m]	East [m]	Remark
#1	−13.0	49.0	Runway entrance
#2	−244.9	107.1	Basketball court
#3	−166.3	−360.2	Small bridge

of three target points. The turn direction (right or left) in this experiment was chosen at the target switching moment based on the direction of the vehicle's velocity with respect to the new target point.

Other relevant flight variables are shown in Fig. 10. The first plot shows the side-bearing angle. When the vehicle was closely orbiting the target point, this angle was maintained within approximately ±5°. The sign of this angle in the flight software was defined such that it is negative when the target point is ahead of the vehicle. Therefore, the angle abruptly changes to large negative values when the target point transition occurs. The second and third plots in Fig. 10 show the roll angle, the roll angle command, the yaw rate, and the rudder commanded recorded during the flight test.

5 Flight Test with a Vision-Based Control System

The proposed guidance law can also be applied to vision-based flight control. The vision-based control can be very useful in a situation whereby GPS is not available. This section presents the relevant flight test results. The GPS in this experiment was used only for recording purposes.

In the experiment, the commonly used gimbal camera was not installed due to complexity and weight penalty concerns. Instead, a fixed camera was installed in the test-bed UAV. Considering the flight characteristics of a fixed-wing aircraft, the camera was attached looking out to the side on the right, as depicted in Fig. 11. It was also tilted downward by approximately 30° to better show the ground below during flight.

In the vision-based flight test, the side-bearing angle is computed using the vision information as follows. First, consider the aircraft diagram in Fig. 12, where the angles ψ_c and ξ_c define the orientation of the camera installation. In the experiment, $\psi_c = 90°$ (for looking out on the right side) and $\xi_c = 30°$.

The rotational transformation matrix from the *NED* local inertial frame to the camera frame ($x_c y_c z_c$ in Fig. 13) is

$$[C]_{NED\rightarrow cam} = [C]_{\xi_c}\,[C]_{\psi_c}\,[C]_{\phi}\,[C]_{\theta}\,[C]_{\psi} \tag{22}$$

where $[C]_{\phi}\,[C]_{\theta}\,[C]_{\psi}$ is the conventional rotational transformation matrices from the *NED* frame to the aircraft body fixed frame, and the additional rotation matrices $[C]_{\xi_c}$, $[C]_{\psi_c}$ are for the rotation from the aircraft frame to the camera frame:

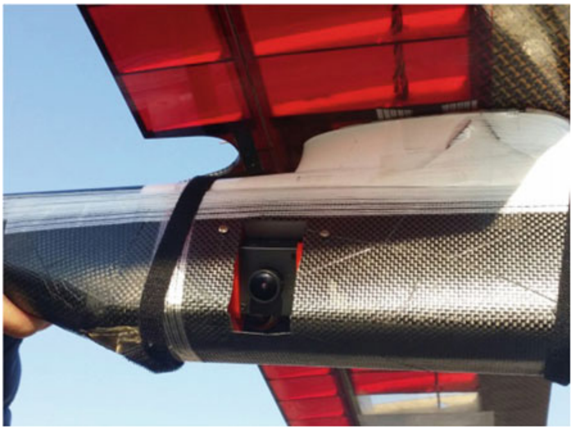


Fig. 11 Fixed-camera on the test bed aircraft looking out to the side

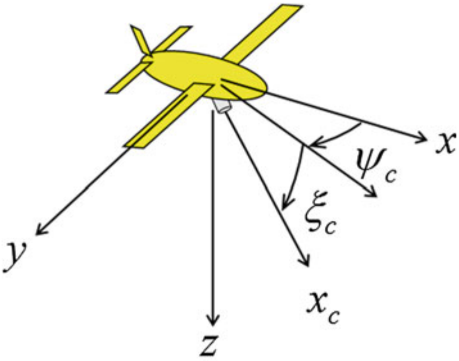


Fig. 12 Definition of camera attachment angles

$$[C]_{\psi_c} = \begin{bmatrix} \cos\psi_c & \sin\psi_c & 0 \\ -\sin\psi_c & \cos\psi_c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad [C]_{\xi_c} = \begin{bmatrix} \cos\xi_c & 0 & \sin\xi_c \\ 0 & 1 & 0 \\ -\sin\xi_c & 0 & \cos\xi_c \end{bmatrix} \tag{23}$$

Additionally, in Fig. 13, the imaginary plane for the camera image is displayed between the camera and the target so that the target image is placed on the straight line that joins the camera and the target.

Let α be the field-of-view angle of the camera, and let l_y be half of the plane's horizontal length. The length l_y may also be considered as half of the number of pixels on the horizontal axis. The distance from the camera to the imaginary screen is $l_x = l_y/\tan(\alpha/2)$. Additionally, let y_c and z_c be the horizontal and vertical positions of the target in the camera image, respectively. The unit vector corresponding to the camera's center x_c direction, when coordinatized into the *NED* frame, is obtained as

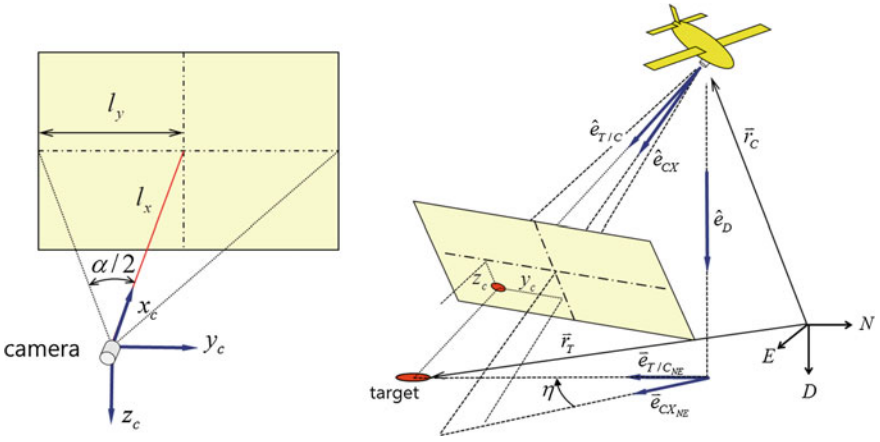


Fig. 13 Geometry used to compute the side-bearing angle with the vision-based flight control system

$$\left\{\hat{e}_{CX}\right\}_{NED} \equiv \begin{Bmatrix} e_{C_N} \\ e_{C_E} \\ e_{C_D} \end{Bmatrix} = [C]_{\psi}^{-1}\,[C]_{\theta}^{-1}\,[C]_{\phi}^{-1}\,[C]_{\psi_c}^{-1}\,[C]_{\xi_c}^{-1} \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \tag{24}$$

The unit vector corresponding to the camera's target-pointing direction, when coordinatized into the *NED* frame, is computed as

$$\left\{\hat{e}_{T/C}\right\}_{NED} \equiv \begin{Bmatrix} e_{T/C_N} \\ e_{T/C_E} \\ e_{T/C_D} \end{Bmatrix} = \frac{1}{\sqrt{l_x^2 + y_c^2 + z_c^2}}\,[C]_{\psi}^{-1}\,[C]_{\theta}^{-1}\,[C]_{\phi}^{-1}\,[C]_{\psi_c}^{-1}\,[C]_{\xi_c}^{-1} \begin{Bmatrix} l_x \\ y_c \\ z_c \end{Bmatrix} \tag{25}$$

The projections of the above two unit vectors on the ground (*NE* plane) are

$$\left\{\vec{e}_{CX_{NE}}\right\}_{NED} = \begin{Bmatrix} e_{CX_N} \\ e_{CX_E} \\ 0 \end{Bmatrix}, \left\{\vec{e}_{T/C_{NE}}\right\}_{NED} = \begin{Bmatrix} e_{T/C_N} \\ e_{T/C_E} \\ 0 \end{Bmatrix} \tag{26}$$

The cross-product of the above two vectors is associated with the angle η by

$$\vec{e}_{CX_{NE}} \times \vec{e}_{T/C_{NE}} = \left|\vec{e}_{CX_{NE}}\right| \left|\vec{e}_{T/C_{NE}}\right| (\sin\eta)\,\hat{e}_D \tag{27}$$

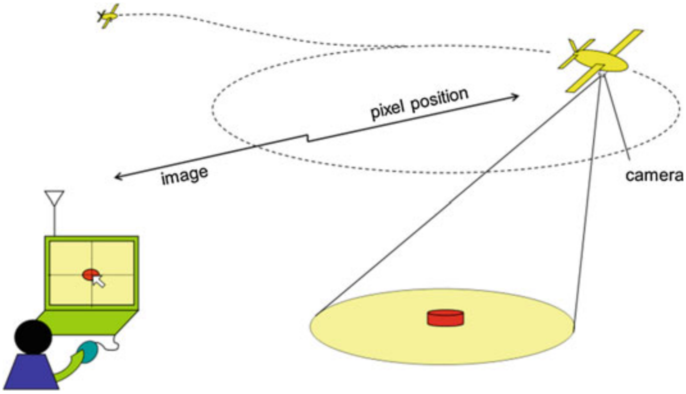


Fig. 14 Experimental setup for vision-based flight control system

Finally, from Eq. 28, sin η can be obtained as

$$\sin\eta = \frac{e_{CX_N}e_{T/C_E} - e_{CX_E}e_{T/C_N}}{\sqrt{e^2_{CX_N} + e^2_{CX_E}}\sqrt{e^2_{T/C_N} + e^2_{T/C_E}}} \tag{28}$$

The above algorithm for the sin η computation and the proposed guidance law was implemented in the flight test. The test-bed UAV has a total mass of 3.0 kg and a wing span of 3.2 m. A typical flight speed is approximately 10 m/s. The aircraft is equipped with a flight control system similar to the one in Sect. 4. It also contains the vision equipment. The camera's field-of-view is 90°, and the transmitted image size is 640 by 480 pixels.

The experimental setup for target recognition and the overall operation concept is depicted in Fig. 14. An interesting feature in this setup is that the identification of the target was performed by a human operator. The transferred camera image was displayed on the ground monitoring computer. The operator chose a target and performed target tracking on the screen using a computer mouse. Then, the target pixel position on the screen was transmitted back to the aircraft for the computation of the side-bearing angle. With this setup, the ground operator was able to manipulate the aircraft very intuitively from a first-person point of view while circumnavigating a series of ground targets of his choice.

A series of flight tests was performed. The control gain K was set for an equivalent damping ratio of 0.5 according to Eq. 18, and the radius of the reference circle was set as 100 m. The altitude was held at approximately 70 m above the ground by the pitch control loop. The ground operator performed the aforementioned mouse tracking for the series of targets defined in Table 2. After performing one round of circling on each target, the operator switched to the next target via mouse-tracking manipulation.

Table 2 Target locations in vision-based flight experiment

Target	North [m]	East [m]	Remark
#1	–366.9	105.0	Playground
#2	–670.5	129.3	Dormitory building
#3	–664.0	–78.5	Library building
#4	–571.3	–98.4	Headquarter building
#5	–341.3	–20.4	Square

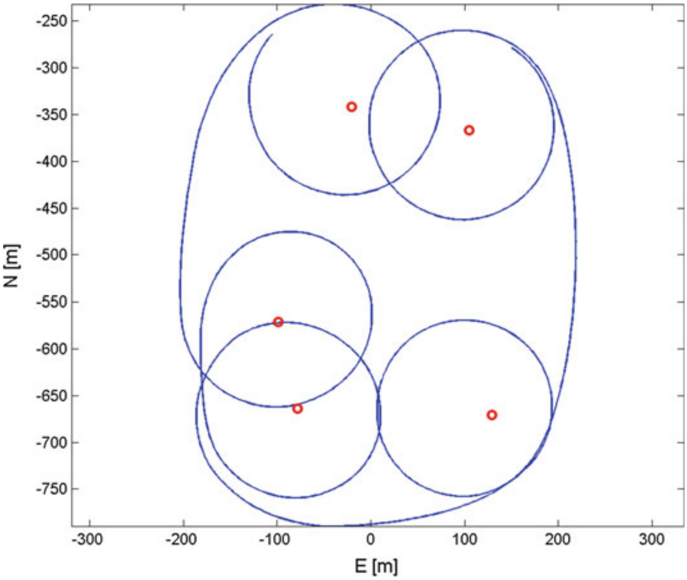


Fig. 15 Vision-based flight test result: Trajectory

A typical flight test's results are shown in Figs. 15, 16 and 17. The flight trajectory and the five target points in Table 2 are shown in Fig. 15. The figure indicates that the aircraft performed a series of clockwise circumnavigation maneuvers over the target points. Other relevant recorded variables are shown in Fig. 16. The first plot shows the target pixel position, which was recorded using the human operator's mouse tracking on the screen. The plot indicates that the pixel position repeatedly converges to the center of the screen after the target transitions. The second plot shows the side-bearing angle. The plot also repeatedly converges to approximately zero as the aircraft circumnavigates a series of target points. The roll angle and its command are also displayed in the third plot. A snapshot of the ground station computer during the flight experiment is shown in Fig. 17. The left side of the screen shows the onboard camera image transmitted from the aircraft. The cross mark (+) over a building on the screen is the target pixel position, which is designated by the

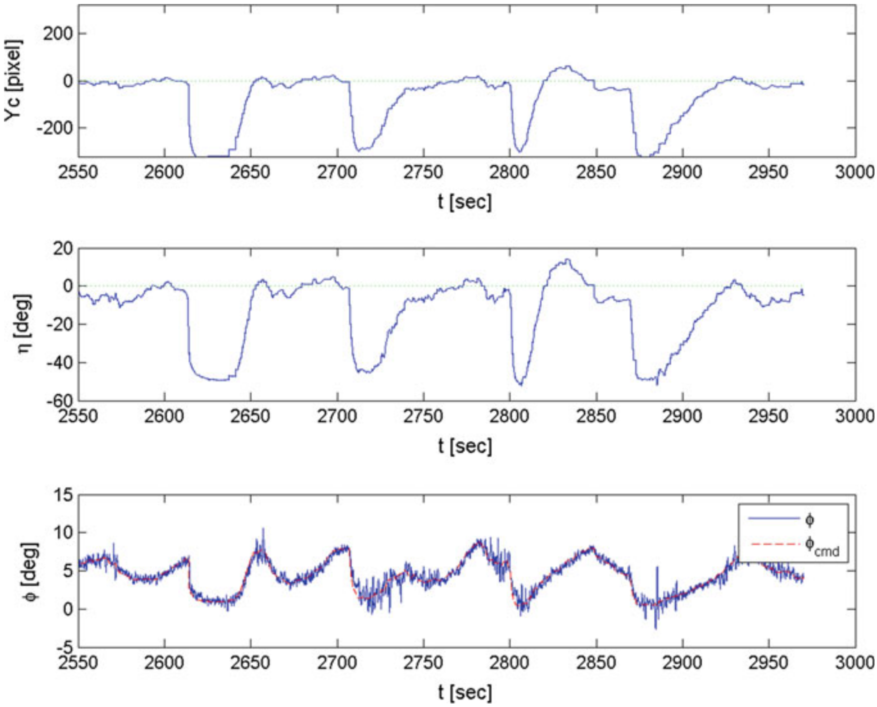
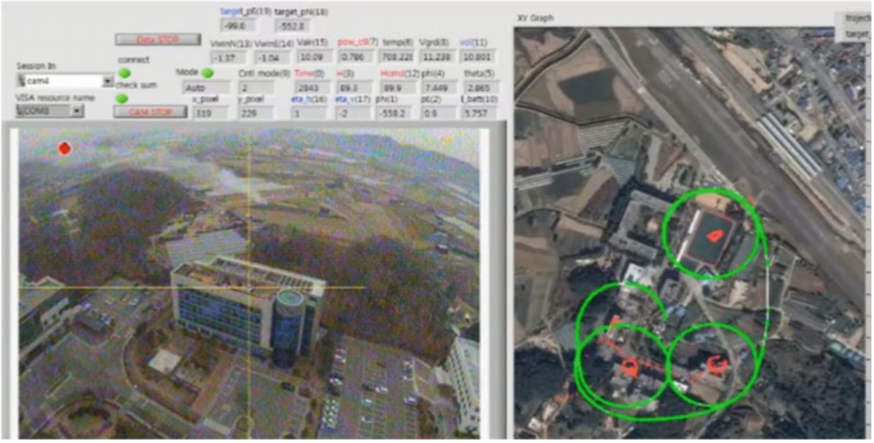


Fig. 16 Vision-based flight test results: pixel position, side-bearing angle, and roll angle



target_pE(19) -99.6
target_pN(18) -552.8

VwinN(13) -1.37
VwinE(14) -1.04
Vair(15) 10.09
pow_ctl(7) 0.786
temp(6) 708.221
Vgrd(8) 11.238
vol(11) 10.801

Session In cam4
VISA resource name COM3

Mode Auto
Cntl mode(9) 2
Time(0) 2843
H(3) 89.3
Hcmd(12) 89.9
phi(4) 7.449
theta(5) 2.865

x_pixel 319
y_pixel 229
eta_h(16) 1
eta_v(17) -2
pN(1) -558.2
pE(2) 0.9
t_batt(10) 5.757

Fig. 17 Ground station snapshot during vision-based flight test

human operator's mouse tracking at that time. The right side of the screen shows the flight trajectory overlapped on the map of the experimental site. This indicates that the aircraft was circling over target #4 at that moment.

6 Conclusion

This work presented a guidance law for circumnavigation using side-bearing angle. The proposed guidance method exhibits a global asymptotic stability. This allows one to skip the path-planning task, and the guidance law can be started for any initial configuration of the vehicle relative to the target position. A linear analysis was performed to help choose a proper control parameter. Two types of flight experiments were performed to demonstrate the performance of the proposed guidance law. In the GPS-based flight test, the side-bearing angle was computed using the GPS position and velocity. In the vision-based flight experiment, a fixed camera was installed on one side of the aircraft, and the side-bearing angle was obtained using the pixel position of the target image. In both flight experiments, it was successfully demonstrated that the proposed guidance law could guide the vehicle to circumnavigate a series of target points.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation ofKorea (KRF) funded by the Ministry of Education (2015R1D1A1A 01060574). The author thanks Yong-Rae Kim for his contributions to the flight tests.

References

1. Tang Z, Ozguner U (2005) Motion planning for multitarget surveillance with mobile sensor agents. IEEE Trans Robot 21(5):898–908
2. Deghat M, Shames I., Anderson BDO, Yu C (2014) Localization and circumnavigation of a slowly moving target using bearing measurements. IEEE Trans Autom Control 59:8 (2014)
3. Shames I, Dasgupta S, Fidan B, Anderson BDO (2012) Circumnavigation using distance measurements under slow drift. IEEE Trans Autom Control 57(4):889–903
4. Dubins L (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Am. J. Math 79:3
5. McGee T, Hedrick J (2007) Optimal path planning with a kinematic airplane model. J Guid Control Dyn 30(2):629–633
6. Anderson R, Bakolas E, Milutinovic D, Tsiotras P (2013) Optimal feedback guidance of a small aerial vehicle in a stochastic wind. J Guidance Control Dyn 36:4, 975–985
7. Anderson R, Milutinovic D (2014) On the construction of minimum-time tours for a dubins vehicle in the presence of uncertainties. J Dyn Syst Meas Control 137(3):031007 (8 pages)
8. Sujit PB, Saripalli S, Sousa JB (2014) Unmanned aerial vehicle path following: a survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. IEEE Control Syst 34(1):42–59
9. Yoshitani N (2010) Flight trajectory control based on required acceleration for fixed-wing aircraft. In: 27th international congress of the aeronautical sciences. Nice, France
10. Rhee I, Park S, Ryoo C (2010) A tight path following algorithm of an UAS based on PID control. In: SICE annual conference 2010. Taipei, Taiwan

11. Cabecinhas D, Silvestre C, Rosa P, Cunha R (2007) Path-following control for coordinated turn aircraft maneuvers. In: AIAA guidance, navigation, and control conference, Hilton Head, SC, AIAA Paper 2007–6656
12. Yamasaki T, Balakrishnan SN, Takano H (2013) Separate-channel integrated guidance and autopilot for automatic path-following. J Guidance Control Dyn 36(1):25–34
13. Jung D, Tsiotras P (2008) Bank-to-Turn Control for a Small UAV using Backstepping and Parameter Adaptation, 17th IFAC World Congress. Seoul, Korea
14. Aguiar AP, Kaminer I, Ghabcheloo R, Pascoal A, Xargay E, Hovakimyan N, Cao C, Dobrokhodov V (2008) Time-coordinated path following of multiple UAVs over time-varying networks using L1 adaptation. AIAA guidance, navigation, and control conference, Honolulu, HI, AIAA Paper 2008–7131
15. Yang K, Sukkarieh S, Kang Y (2009) Adaptive nonlinear model predictive path tracking control for a fixed-wing unmanned aerial vehicle. AIAA guidance, navigation, and control conference, Chicago, IL, AIAA Paper 2009–5622
16. Gates DJ (2010) Nonlinear path following method. J Guidance Control Dyn 33(2):321–332
17. Morales J, Martinez JL, Martinez MA, Mandow A (2009) Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner. EURASIP J Adv Signal Process 2009(3):1–10
18. Park S, Deyst J, How JP (2007) Performance and lyapunov stability of a nonlinear path-following guidance method. J Guidance Control Dyn 30(6):1718–1728
19. Park S (2012) Autonomous aerobatics on commanded path. Aerosp Sci Technol 22(1):64–74
20. Cho N, Kim Y, Park S (2014) Three-dimensional nonlinear path-following guidance law based on differential geometry. In: Proceedings of the 19th IFAC world congress (2014)
21. Cao Y, Muse J, Casbeer D, Kingston D (2013) Circumnavigation of an unknown target using UAVs with range and range rate measurements. In: Proceedings of the 52nd IEEE conference on decision and control (CDC)
22. Hashemi AR, Cao Y, Yin G, Casbeer D (2014) UAV circumnavigation using noisy range-based measurements without GPS information. ASME J Dyn Syst Meas Control
23. Ren W, Beard R, Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. IEEE Trans Control Syst Technol 12(5):706–716
24. La Salle JP (1976) Stability of nonautonomous systems. Nonlinear Anal. Theory Methods Appl 1(1):83–91

Mission Control Concept for Parcel Delivery Operations Based on a Tiltwing Aircraft System

M. Schütt, P. Hartmann, J. Holsten and D. Moormann

1 Introduction

The tiltwing aircraft configuration is a combination of two different concepts of aircraft that allows stationary flight in a wide range of velocities from hover to fast forward flight. Within a tiltwing configuration the advantages of rotary-wing and fixed-wing configurations are combined to perform energy efficient high speed flight and vertical take-off and landing (VTOL). For a research project, an aircraft in tiltwing configuration (displayed in Fig. 1) was used within a parcel delivery application [6]. During operation, a remote location in a mountainous area had to be supplied with time-sensitive goods in a frequent and fully automated way. This included all different control levels during flight beyond the line of sight and interaction of the aircraft and ground facilities.

In literature some solutions for different aspects of tiltwing control are available, two approaches can be found in [9, 11]. Both show control concepts that subdivide the flight envelope in discrete points and present manual flight only. Indoor transition methods for a quad rotor-biplane without considering disturbances are described in [7]. Piloted approach maneuvers requiring a runway were analyzed in [2]. In [10] the L1 adaptive control architecture for the manually piloted NASA tiltwing GL-10 is presented. In previous articles the authors have developed a velocity controller [3, 12] to mask the complex flight mechanics and variations of the tiltwing configuration. Higher level control to perform a transition from hover to cruise flight in form of a straight lined maneuver is presented in [4]. For fully automated operations, e.g. in severe weather conditions, several additional issues have to be solved. A higher level mission control system is necessary to allocate tasks to different sub-controllers during all flight phases.

M. Schütt (✉) · P. Hartmann · J. Holsten · D. Moormann
Institute of Flight System Dynamics, RWTH Aachen University, Aachen, Germany
e-mail: schuett@fsd.rwth-aachen.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_26

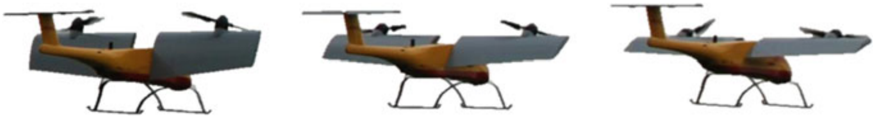


Fig. 1 Tiltwing aircraft during transition from Hover to Cruise flight

Within this paper a control concept for the entire mission operation, based on an existing maneuver control concept [3, 4], is proposed. The control system for flight guidance includes precision take-off and landing, hover as well as cruise flight. The conversion between hover and cruise requires special attention to allow stationary flight and avoid discontinuities. The design of the waypoint pattern and the integration of different landing spots is discussed with regard to regulations and environmental constraints. The fully automated flight is observed and commanded by an operator in charge of safety procedures. Assessment of the entire design and the tiltwing's characteristics was performed by simulation and flight tests. The performance of the mission control system has been proven during a 3-month trial operation in civil airspace.

This paper is structured as follows: Sect. 2 presents the specific features of the mission scenario and the aircraft system, including ground facilities. In Sect. 4 the control system associated with the mission controller is described. The mission control system itself is presented in Sect. 4.3, before presenting the operator interaction to control safe operation. In Sect. 5 the assessment and operation results are discussed. At last, Sect. 6 gives concluding remarks.

2 Mission Scenario: Parcel Delivery Operation

The tiltwing aircraft is used within a parcel delivery scenario, where a remote location in the Bavarian Alps had to be supplied with parcels on a regular basis. The trial operation was conducted by 'RWTH Aachen University' and 'DHL Parcel'. Time-sensitive goods had to be transported from 'Reit in Winkl' to the 'Winklmoosalm'. Two parcel handling stations, called skyports, were used as landing spots. In the following the characteristics of the mission and specific challenges in terms of regulations are presented.

2.1 Mission Overview

The route of the mission, connecting 'Reit in Winkl' and 'Winklmoosalm', is eight kilometers long and has an altitude difference of 460 m. As the scenario is located in a mountainous area (compare Fig. 2), the optimal route in terms of energy consumption

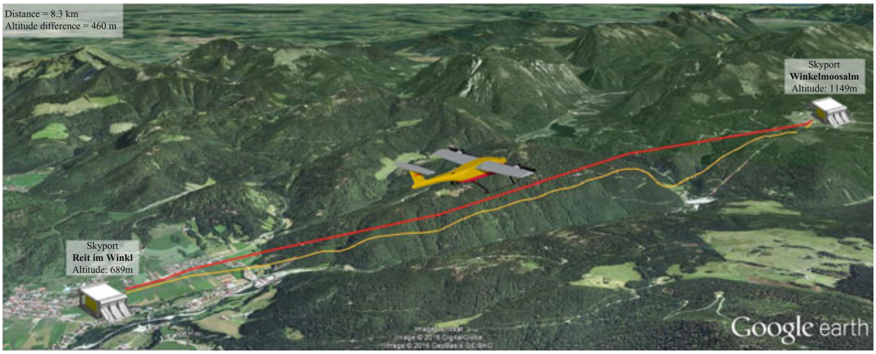


Fig. 2 Overview of the Route from Reit im Winkl to Winklmoosalm

is generated as a combination of shortest distance and lowest altitude difference. The route follows a mountainside and a highway, requiring special attention during design of cruise flight control in terms of track error tolerance.

To accomplish the mission, the aircraft had to fly beyond line of sight of the operator, which leads to special restrictions, as described in the following section. Flight beyond line of sight also requires high level of automation, to ensure safe operation. This includes flight control but also the entire turnaround process, which is handled completely automated by a manipulator inside the skyport. Most of the procedures are predefined and do not need confirmation from the operator. Essential and safety-relevant procedures, like time of take-off, are commanded manually supported by the system. The interactions of operator and automated safety procedures are explained in detail in Sect. 4.3.3.

The weather conditions in a mountainous area cause unknown wind conditions at altitude even if the wind is known at ground level, requiring robust flight control. Additionally there is a risk of icing, because of temperatures below zero degrees at altitude. For good performance at start, precision landing and approach maneuvers, the wind at the two skyports is measured in speed and direction.

As one of the skyports is located in an urban environment, buildings and streets are in close distance. Take-off and landing are performed on a small platform mounted on the skyport. For safety reasons, a certain area around the skyport was fenced off. The ground track of the aircraft has to be controlled precisely during maneuver, therefore track errors and overshoot had to be minimized.

2.2 Regulations

The design and testing of the controller had to consider the legal situation in Germany. Unmanned flight beyond line of sight is generally not allowed. For this research project an permission was granted, based on an airspace restriction. Therefore the project had to undergo a safety review in cooperation with authorities and partners involved. Different safety and environmental protection requirements were considered for designing the route.

The route was designed to keep the road intersections at a minimum and cross roads perpendicularly. The flight level between departure and approach was set to a minimum of 80 m above ground level for environmental protection reasons, the maximum altitude above ground was reached during a valley transit at 420 m. The flight path had to be predictable during the entire route, which influenced the design of the waypoint pattern, presented in Sect. 4.3.1. The limits for maximum accepted track deviation were set to 10 m. Thanks to the airspace restriction mainly interactions with emergency service helicopters had to be dealt with, as these were expected to cross the airspace frequently in the skiing-region. Therefore an ADS-B transponder broadcasted the UAVs position during all flights. A detailed analysis of the permission process and different regulations can be found in [6].

2.3 Skyport

The skyport acts as the landing spot and completes the turnaround process of the aircraft, as shown in Fig. 3. On top of the skyport a platform of 3 × 3 m is installed for take-off and landing of the tiltwing aircraft. While on the platform, the aircraft can be protected from snow, ice and vandalism under a dome.

Each skyport is equipped with an manipulator to interact with the aircraft. The manipulator is able to fasten the aircraft after touchdown and center it on top of the skyport for unloading. The container, including the battery and payload, are dismounted by the manipulator and stored inside the skyport. Inside the skyport

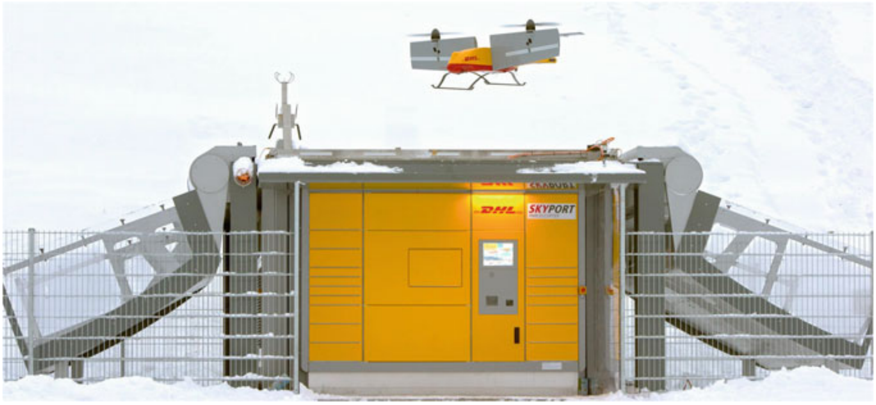


Fig. 3 Skyport and Tiltwing Aircraft during operation [8]

multiple slots store, charge and heat the batteries of different containers. Before take-off the manipulator turns the aircraft to head into the wind for take-off.

The transition maneuver from cruise to hover and precision landing on the platform require detailed information on the wind condition at the landing spot, as stated in Sect. 4. Therefore a anemometer is installed on each skyport. Additionally one of the skyports is equipped with a real-time-kinematic reference station, broadcasting satellite navigation correction data to the aircraft.

3 Parcelcopter: Tiltwing Aircraft System

In the following the flight mechanics and systems of the tiltwing aircraft configuration are presented. The aircraft system, consisting of the demonstrator aircraft, datalink and ground control station, is designed to allow for long range flight beyond the line of sight.

3.1 Flight Mechanics

The tiltwing configuration combines advantages of rotary-wing and fixed-wing configurations. During cruise the wing is rotated to a fixed wing configuration producing sufficient lift for level flight. In this configuration the thrust of the main propulsion system is used to overcome the drag of the aircraft. The tiltwing configuration includes all control surfaces typical to an airplane. For hovering and VTOL the tiltwing aircraft is able to rotate its wing around the lateral axis. As the main propulsion system is installed on the wing, the thrust is used to compensate the weight of the aircraft. The ailerons are located in the slipstream of the propellers to gain yaw control in hover. The elevator loses its effect without incident flow, which is why an auxiliary device (e.g. a ducted fan) to control pitching moment for hover is needed.

During the transition from hover to cruise flight and vice versa a distinct assignment of control surface effects and aircraft's rotational axis is not possible. Because of the changing aerodynamics at different flight states the actuators' effectiveness varies in direction and magnitude. Nevertheless the tiltwing aircraft can perform stationary flight in the entire flight envelope from hover to cruise. By adjusting the tilt angle and thrust to the current wind speed, precise vertical starting and landing is possible even at high wind speeds with low pitch angles. Avoiding lateral wind components, which would cause constant roll angles, a tiltwing heads into the wind while hovering and during VTOL. A tiltwing aircraft features efficient and fast cruise flight capabilities in contrast to multirotor systems. Therefore large cruise ranges are achievable also at headwind conditions.

3.2 Aircraft

The DHL-Parcelcopter is a tiltwing configuration with a wingspan of 2 m. The electric main propulsion system, including two counter rotating propellers with a diameter of 0.7 m, is installed in the wing. The aircraft can be seen hovering above the skyport in Fig. 3 and during flight in Fig. 1. The entire aircraft structure is of composite, leading to a MTOM of 14 kg, including a payload of 2 kg. The powertrain is designed to be weatherproof and with a focus on low noise emissions, as well as efficiency. The conversion by tilting the wing is performed by a spindle drive with limited actuator dynamic. The elevator is located in a T-tail configuration to reduce wing downwash influences. The ducted fan is included into the fuselage to reduce the drag during cruise flight.

To interface with the skyport's landing platform the aircraft is equipped with a landing skid. Furthermore the lower mid part of the of the fuselage is detachable and contains the batteries and the payload compartment. This fuselage part is called container and is exchanged for every flight. It can be automatically detached and recharged by the manipulator inside the skyport. The batteries have been optimized for this mission in terms of range and maximum payload. In total the hover time is kept as short as possible, due to an increase of power consumption in hover of approximately two-thirds compared to cruise flight.

The aircraft's flight control hardware is based on *ARM Cortex-M4* microcontrollers. The control algorithm is designed in *MATLAB/Simulink* and automatically translated to C-code. The navigation system consists of a MEMS-based IMU, a heated pitot-static-system, a barometer, a magnetometer and precision real-time-kinematic satellite navigation receiver. The aircraft's heading can be estimated precisely thanks to the dual-antenna operation of a *Novatel OEM-617D-GNNS-receiver*. The navigation solution's precision (relative to the reference station) is up to 1 cm in the horizontal plane and 2 cm in the vertical axis.

3.3 Ground Control Station and Datalink

The datalink to the ground control station is established via both skyports and has a redundant design. Based on two different data modems the system is able to ensure the required high range in the mountainous area. The data modems are *IMST LoRa* using 868 MHz and a *Sierra Wireless* mobile communication transceiver.

All telemetry and command data are processed by the Ground Control Station (GCS). One of the skyports is directly connected to the GCS. Both skyports are connected to each other and act as datalink repeater. Additionally a datalink repeater is located half way between both skyports. This maintains a steady datalink to the aircraft during the entire flight. Detailed explanation and analysis of the communication structure is given in [13].

The wind data and navigation correction data are transferred to the aircraft via datalink. The operator, located in the GCS, receives all flight data to observe the flight. The GCS needs a constant connection to the aircraft, so the operator is always able to intervene. During flight the operator could send different commands triggering safety procedures or change the route of the aircraft. The operator is also able to command the manipulators of the skyports after touchdown.

4 Mission Control System

The mission control system is designed to control the tiltwing during the entire mission. Figure 4 illustrates the flight mission elements in a conceptual way.

The typical flight starts with a vertical take-off from a skyport. After take-off the position controller holds the aircraft at the starting waypoint until the departure maneuver is initiated. The waypoint controller guides the aircraft during cruise flight until the approach maneuver is started. After reaching the target in hover state, the vertical descend and landing procedure is performed. Different alternative landing spots (ALS) are located along the route. Next to each skyport an ALS can be reached in hover flight.

An overview of the control system is shown in Fig. 5, illustrating the cascaded controller design approach: The innermost loop is the velocity controller, including attitude control. The flight path control loop is designed for different flight states: For take-off, landing and hover a position controller is controlling the aircraft's flight path. During cruise flight the flight path of the aircraft is controlled by a waypoint controller. To accomplish fully automated flight, the controllers have to interact and control authority has to be allocated by a mission control system, described in Sect. 4.3.

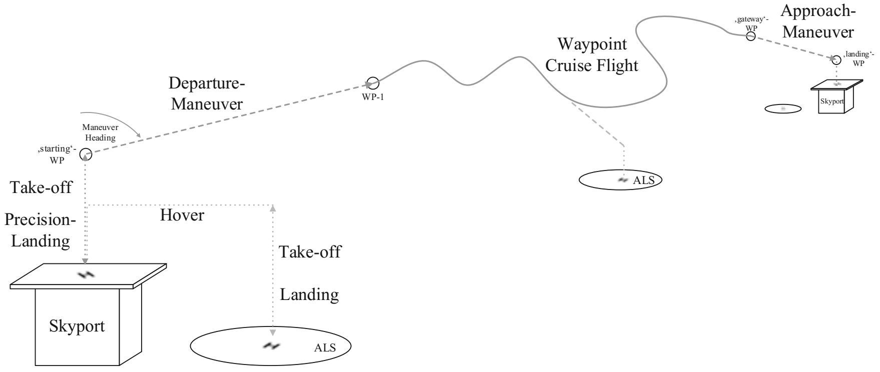


Fig. 4 Overview different flight mission elements

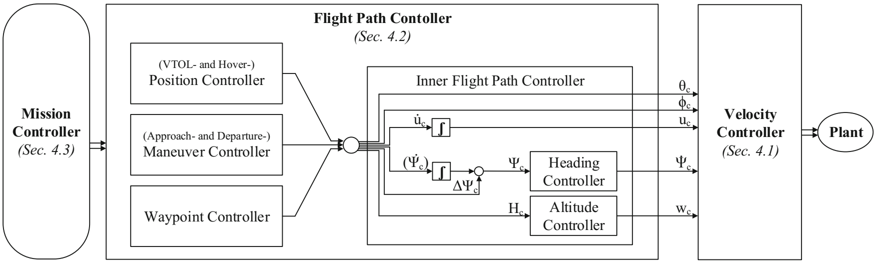


Fig. 5 Overview of the control system (adapted from [4])

4.1 Velocity Controller

A convertible aircraft has to perform flight state transitions to make use of its entire flight envelope regarding forward speed. The complex flight mechanics and variations in the different flight states pose a challenging problem in terms of analysis and control. The performance of the velocity control concept, presented in [3], was demonstrated, using different tiltwing aircraft and performing trimmed flight at any airspeed within the flight envelope. The velocity controller allows for flight over the whole flight envelope and unifies the significant variations of flight mechanical characteristics. Thanks to this, the velocity controller is a well chosen inner control loop with a homogeneous set of command variables in all flight states.

The velocity controller uses a horizontal, body-fixed coordinate system, controlling airspeeds in longitudinal u_c and vertical direction w_c, with the longitudinal motion mostly controlled by tilt angle variations. The aircraft's heading rate $\dot{\Psi}_c$ is controlled, because the absolute heading is irrelevant for the velocity controller, and a turn coordinator controls a corresponding bank angle depending on the airspeed. The attitude angles θ_c and ϕ_c complete the set of command variables.

4.2 Flight Path Controller

The flight path controller consists of two different parts: Sub-controllers, that are activated by a mission controller, and a inner flight path controller, that is permanently active. In the following the inner flight path controller is described first, including the generation of the command variables heading and airspeed. Subsequently the three sub-controllers are presented.

The heading controller computes the velocity controller's command variable heading rate $\dot{\Psi}_c$. The controller employs a PD feedback. The proportional and derivative gains are gain-scheduled from hover to cruise, depending on airspeed. The heading's rate of change $(\dot{\Psi}_c)$, commanded by the three sub-controllers, is integrated, to prevent discontinuities.

The altitude controller generates the command variable vertical speed w_c for the velocity controller. The altitude controller feedbacks the altitude deviation proportionally and feedforwards the slope of the flightpath. Due to the aircraft's flight envelope, the vertical speed is limited depending on the forward airspeed. Most of the missions' altitude difference is climbed during waypoint flight, as vertical ascend and descend is kept short for energy consumption reasons. The altitude controller can be switched off for constant vertical speeds w_c during start and landing.

The commanded airspeed u_c is the most significant value for the velocity controller, as it has major influence on flight mechanical variation. To initialize the different sub-controller without discontinuities, an integrator is in charge to generate a continues airspeed command value u_c.

Position Controller The position controller is designed for hover of the tiltwing aircraft. Following a classical approach, the controller is divided in a longitudinal and a lateral part. Therefore the error in position and velocity between the aircraft and the designated position is transformed from the earth fixed to the body fixed coordinate system, see Fig. 6.

Both controller parts share the same structure and consist of a proportional/ derivative PD-part and an integral I-part, the control law is displayed in Fig. 6. Each PD-part feedbacks the position and velocity error to the corresponding attitude angle, showing fast response characteristics. While the PD-part features fast response for small position deviations or wind gusts, the I-part is used to compensate the effect of constant wind. As a consequence the aircraft ends up in a position presented in gray, heading into the wind at an adapted airspeed. The I-parts are used to change the inputs of the previously explained integrators for airspeed and heading.

Maneuver Controller The maneuver controller has to guide the aircraft's transition from hover to cruise flight and vice versa. Generally tiltwing aircrafts try to avoid sideslip angles especially during transition for a symmetric incident flow at this phase of significantly varying aerodynamics and flight mechanics. A classical approach

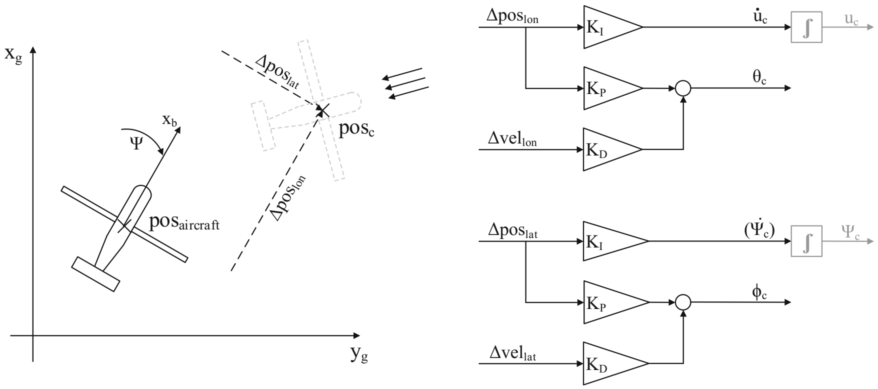


Fig. 6 The coordinate system and control law of the position controller (adapted from [4])

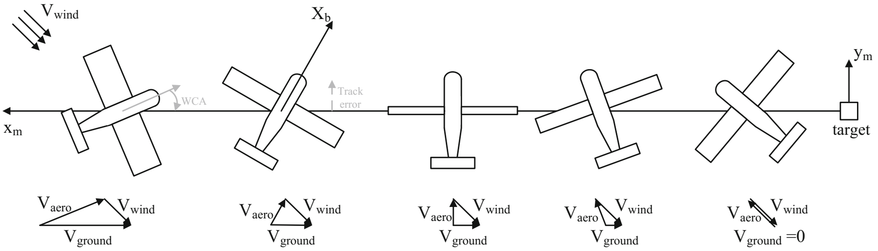


Fig. 7 The coordinate system and progress of the maneuver (adapted from [4])

including a turn flight to start decelerating while heading into the wind would end in a hover state without any sideslip angles. The covered ground track due to the turn is changing, depending on different wind conditions. The total area overflown for turn flight approach is not suitable for the given scenario in an urban environment. Instead, a straight lined maneuver was developed, which is described briefly in the following. For more details of the design of the maneuver controller see [4].

This control approach is taking advantage of the tiltwing to adapt the wing angle and works identically for all different wind conditions. The maneuver's principle of function is almost identical for departure and approach. Hence, only the approach is described here. The controller uses a coordinate system that has its origin located at the target and the x-axis pointing in direction of the departure track (compare Fig. 7). Due to the limited fenced off area all deviations from track and overshoot at the landing position should be kept at a minimum.

The process is displayed in Fig. 7 for a tail wind situation. The aircraft approaches from left to right, flying on a linear track. It decelerates and turns the heading into the wind in a coordinated way. When decelerating, the aircraft has to adapt the wind correction angle to compensate the lateral wind component. As soon as the wind correction angle reaches 90°, the aircraft has to start accelerating until reaching a hover state with airspeed opposing the wind speed. The maneuver works in a similar way for any other wind condition, which mainly determines the final hover state, while the process of turning and accelerating stays the same. The maneuver controller's task can be divided into three parts. The first task is to decelerate the aircraft as fast as possible, as Hover should be limited due to energy consumption reasons. Therefore the heading's rate of change ($\dot{\Psi}_c$) and airspeeds rate of change $\dot{u}_c$ should be as large as possible, but always coordinated to stay on track. The second part is to keep the aircraft on the straight lined track and to minimize the track error. The control law is working in the same way as the position controller. The third part is to reach hover at the designated position. Therefore, the current wind condition has to be known and the length of the maneuver at any given wind condition has to be pre-calculated. This allows starting the maneuver at just the right distance before the target to reach hover at target.

The departure maneuver doesn't need feedback, it is open loop and finished when reaching the cruise airspeed.

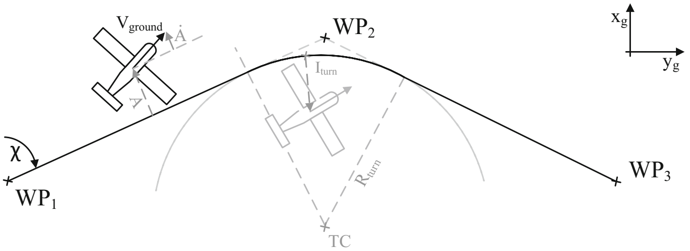


Fig. 8 Coordinate system of the paypoint controller and turn geometry

Waypoint Controller The waypoint controller uses a classic position line approach to follow a designated path. A preceding filter calculates the course χ, current lateral track error A and its derivative $\dot{A}$. A turn geometry is calculated, dependent on maximum roll angle and cruise airspeed, see Fig. 8. The combination of three waypoints (last, current and upcoming) defines the turn center TC and the starting and end points of the turn.

During straight flight and turn flight the three corresponding values (χ, A and $\dot{A}$) are transferred to the controller. The controller manipulates the heading command based on a PI-algorithm combined with a damping part. The control gains are higher than usual for cruise flight. The controller causes strong responses on track errors, which is driven by the terrain and surrounding infrastructure of the mission. The track follows a mountainside next to a highway, to which the aircraft has to stay in a sufficient distance. Therefore track errors and overshoot during turns are highly undesired.

Each waypoint has several attributes and most of them are employed by the mission controller, described in Sect. 4.3.1. Two of the attributes 'altitude' and 'airspeed' of each waypoint are interpreted by the waypoint controller. The altitude is interpolated linearly between last and upcoming waypoint by the waypoint controller to generate a current altitude command. The airspeed attribute of the upcoming waypoint defines the commanded airspeed by manipulating the integrator input of the inner flight path controller. The waypoint controller is only allowed to change the airspeed in a small range of cruise speeds.

4.3 *Mission Controller*

The mission controller is designed as a state machine, as shown in Fig. 9. The mission controller is directing the entire flight and allocating the control authority.

The definition of the route in terms of waypoints and nodes is depicted first. An overview on the standard procedures of an operational flight is presented and additional safety features are described subsequently.

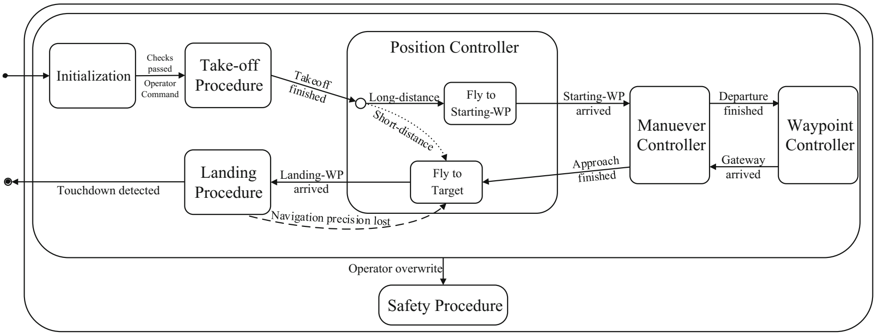


Fig. 9 State diagram of the mission controller

4.3.1 Mission Definition Data

Different information for waypoints, legs and departure or target nodes are stored in attributes. These attributes are interpreted by the mission controller, generating the route.

Waypoint Pattern Each waypoint is defined by an individual tag and position coordinates, displayed in Table 1. Additional attributes are ‘airspeed’ and ‘altitude’ (above mean sea level). Every node represents a possible take-off and landing spot. The current route is thereby defined by selecting a departure-node and target-node. A set of possible follow up waypoints are assigned to every waypoint. Depending on the selected target-node, the waypoints are pointing to a follower. The target-node can be changed during flight, causing a change of the following order of waypoints.

In Fig. 10 a one-way route is illustrated. Node A is selected for departure and Node B as target, creating a route as each waypoint is pointing to another in ascending order from WP 0 to WP 6. A target change to Node C affects WP 3, pointing at WP 7 instead of WP 4. This makes WP 3 a ‘decision’-waypoint, changing its following WP as a result of target change. The waypoint controller calculates a turn starting ahead of the waypoint, dependent on the maximum bank angle as shown in Fig. 8. A change of target right before reaching WP 3 would cause large track deviations, as the turn is initiated too late. Therefore every decision-waypoint is protected from target change by another waypoint located in a short distance ahead. This waypoint includes the attribute target ‘locked’, causing a change of target being not accepted by the mission controller on the following leg. The distance of the this leg between ‘protection’- and ‘decision’-waypoint is dependent on the maximum bank angle, ensuring a turn flight to any follower of the ‘decision’-waypoint. While flying the leg from WP 2 to WP 3 a change of target from B to C doesn’t affect the route, as the target is ‘locked’. After passing WP 3 all following waypoints keep the order independent of the target, for both targets the follower of WP 4 is WP 5 for example. A waypoint pointing at a landing waypoint is tagged as a ‘gateway’. Only on the leg from gateway- to landing-WP (from WP 5 to WP 6), the maneuver can be started

Table 1 Attributes of waypoints

ID	pos	'Airspeed'	'Altitude'	'Gateway'	'Locked'	'Default'	Target 'B'	Target 'C'
WP 1	x,y	16	115	no	no	WP 2	WP 2	WP 2
WP 2	x,y	19	250	no	yes	WP 3	WP 3	WP 3
WP 3	x,y	19	250	no	no	WP 4	WP 4	WP 7
...								
WP 5	x,y	16	180	yes	yes	WP 6	WP 6	WP 6

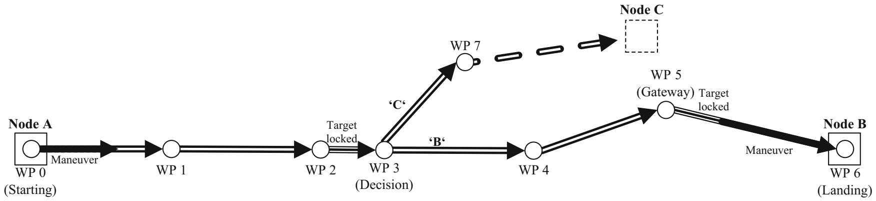


Fig. 10 Schematic overview of the waypoint pattern and the attributes of a waypoint

by the mission controller. After passing the gateway a change of target is no longer possible.

For safety procedures (described in Sect. 4.3.3) each waypoint has additional attributes like 'default' target. The properties of the waypoint pattern make every possible route predefined and predictable. Even a change of target, turn around or activation of safety procedure ends in a well-defined route. This way it is possible to simulate all flight paths in advance. For regulatory reasons and due to close distance to the mountainside this is a helpful feature of the waypoint pattern.

Node Definition Each node is working as a take-off and landing spot, therefore its position coordinates and altitude are attributes. Skyports as well as ALS are defined as nodes, the attributes can be found in Table 2. The approach and departure from and to a node are performed via maneuvers. The attribute 'maneuver heading' of each node is pointing in the direction of the following leg (see Fig. 4). Each node has an associated 'starting'-waypoint in a certain altitude above ground. After vertical take-off the departure maneuver is initiated from this 'starting'-waypoint. The departure maneuver is finished when reaching cruise flight on the first leg in direction of the 'maneuver heading'. At the target node, the deceleration maneuver is initiated while flying onto the gateway leg. Dependent on the current wind condition the mission controller activates the maneuver controller, to reach hover state at the corresponding 'starting'-waypoint above the target-node. The minimum gateway leg length is defined by the maximum maneuver length.

Landing onto a skyport requires a high precision navigation solution, due to the small landing platform. As the ALS are larger, no high-precision navigation solution is necessary to land on it. Besides the maneuver definitions, a node is also character-

Table 2 Attributes of nodes

ID	Pos	'Altitude'	'Maneuver heading'	'Starting'-WP
Node A	x,y	100	90	WP 1
Node B	x,y	150	−75	WP 6
Node C	x,y	...		

ized by the need for high precision navigation solution. If necessary, the node points to one of the ALS in close distance, this safety procedure is described in Sect. 4.3.3.

Mission Types Next to each skyport an ALS is located on the ground. A maneuver is related to both nodes pointing in the direction of the mission's route. During daily operations the demonstrator aircraft had to be maintained and checked on a regular basis. Therefore, next to the standard waypoint flight another mission type was introduced for short distance flight. If short distance flight is selected, the mission controller will guide the aircraft to target, conducting hover flight only.

4.3.2 Regular Modes of Operation

The state machine of the mission controller can be summarized as shown in Fig. 9. For accomplishing the mission, the controller works in the following way:

Initialization A operational flight starts with a set of initialization processes before take-off. The operator selects the current skyport as the starting node and the desired as target node, this defines the upcoming route. Additionally the mission type is selected to be 'long distance' or 'short distance' flight. Current wind data of the anemometer of the starting skyport are broadcasted to the GCS. The operator commands the manipulator to turn the aircraft to head into the wind and adjusts the initial aerodynamic speed to compensate the headwind at the moment of take-off.

Take-off After this initiation process, the take-off procedure is started by entering the next state of the mission controller. During the engine ramp up, the mission controller imposes a certain downward-force on the velocity controller to prevent the aircraft from lifting off. Just after the engines reach their final value this force is reset to zero and a constant rate of climb is commanded to ensure a defined liftoff.

Position Control As long as the aircraft is close to the platform, high attitude angles can be avoided by only damping the horizontal speeds and inhibiting the position feedback. Additionally the commanded vertical speed is transferred directly to the velocity controller; the altitude controller is not active in this phase. At a certain altitude, the position controller is activated to ensure the aircraft reaches its end of vertical climb directly above the platform. Short period disturbances like gusts are compensated by roll and pitch angle, while constant wind change cause change in heading or forward airspeed via the integrators.

Departure Maneuver After reaching the starting waypoint, the mission controller activates the departure maneuver controller. The direction of the maneuver is taken from the node attribute ‘maneuver heading’. As the maneuver works identically for any given wind condition, the aircraft simply starts its procedure of accelerating and turning. As soon as the airspeed reaches cruise speed, the maneuver control loop resembles the control law of the waypoint controller for straight flight. That way the pitch command is faded out, as there is no need for longitudinal position adjustments at higher airspeeds. Additionally the maneuver controller alters the lateral command variable from roll angle ϕ_c to heading's rate of change ($\dot{\Psi}_c$). Thanks to this gain scheduling no discontinuities occur during switching.

Waypoint Flight After reaching cruise speed, the mission controller activates the waypoint controller. During cruise flight a combination of waypoints define the route. The mission controller identifies the upcoming waypoint and changes the waypoints during the turn at the correct time. As the gateway waypoint already includes a reduced ‘airspeed’ attribute, on the leg to the gateway waypoint the airspeed is reduced and the target is locked. The maneuver controller has to schedule the approach maneuver on the gateway leg, therefore the current wind condition at target has to be updated.

Approach Maneuver After reaching the gateway waypoint, the waypoint controller remains active. The maneuver controller is activated by the mission controller at a certain distance to target, located on the gateway leg. The maneuver controller initializes the start of deceleration adjusted to the maneuver length for the current wind condition. The aircraft starts decelerating and heading into the wind until the groundspeed along the track to target is zero. At this moment the mission controller allocates control authority to the position controller to compensate residual displacements or overshoots. The position controller guides the aircraft to the landing waypoint.

Landing When reaching the landing waypoint at certain tolerance, the vertical landing procedure is started. Thanks to the capability to adjust the wing angle and heading to the current wind condition, the tiltwing is able to approach the platform even at high wind speeds without need for high attitude angles. During vertical landing the mission controller commands a constant descend rate directly to the velocity controller. After touchdown detection, the engines are shut down and the aircraft is prepared for the turnaround process. Therefore the wing is tilted in a pre-defined position and the aircraft is moved by the manipulator to the center of the platform for unloading.

4.3.3 Operational Procedures

Next to scheduling the operational procedure, another task of the mission controller is to allow for operator and safety pilot overwrites. Additionally, the mission controller performs automated checks and activates different safety procedures, if necessary.

Operator The operator is responsible and in charge of operation during the whole flight. He uses the GCS to observe the flight and has the permanent authority to interrupt. During default procedure the operator has to confirm the start procedure before the engines are run up. During the run up of the engines the start can be aborted. After take-off the operator has different possibilities to activate or confirm safety procedures. An operator overwrite causes the mission controller to change the state, according Fig. 9. During operational flights the operator never had to overwrite the default procedure.

Automated Checks Before take-off a number of initialization processes and automated checks are performed to ensure safe flight. The navigation solution has to provide a minimum precision. The position of the aircraft has to match the selected starting node. The aircraft's heading and measured indicated airspeed have to be adjusted to the wind condition measured by the anemometer.

Before descending onto the skyport, the precision of the navigation solution is checked. If the accuracy is not sufficient, the target is changed to the corresponding ALS. The position controller stays active and guides the aircraft to the landing waypoint.

During descend, the mission controller is able to estimate the wind speed. While heading into the wind, the aircraft's airspeed compensates the wind speed to obtain zero groundspeed. The mission controller uses the commanded airspeed to estimates the current wind speed by employing a model of empirical boundary layer data. The estimated wind speed is used to adjust the commanded touchdown point, which increased the precision of the landing. After touchdown detection the engines are shut down automatically.

Safety Procedures The safety procedures include several operations, which need confirmation of the operator. The aircraft is always able to return to the starting point by target change of the operator. Additionally the operator can guide the aircraft to a number of predefined ALS along the route. These ALS also constitute nodes, including maneuver heading. This way a target change to an ALS also simply changes the order of waypoints, leading to the new target.

5 Assessment

The mission control system was designed and tested in a simulation environment. During flight tests the simulation was validated and controller performance could be analyzed. The trial operation proved the suitability for practical use of the entire system.

5.1 Simulation

The simulation consists of a closed loop model of the velocity controller and the aircraft, which is the controlled system. In combination with this model, the different

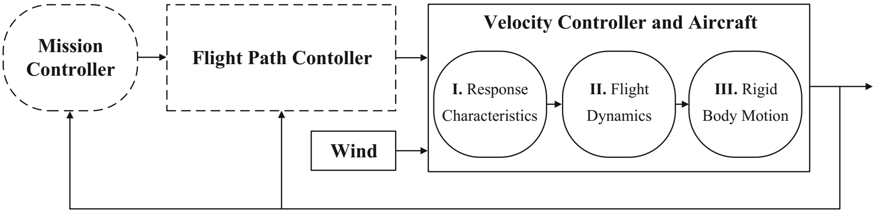


Fig. 11 Overview of the simulation environment

high level controllers could be designed and tested. There was no demand for modeling the flight mechanical variations as they are mainly compensated by the velocity controller. Especially the complex allocation of the control surfaces and forces and moments in the different flight states did not have to be considered.

Simulation Structure Like the controller, also the simulation is implemented in *MATLAB/Simulink*. The model of the controlled system can be divided into three parts, displayed in Fig. 11. The first part describes the different response characteristics of the commanded variables. These characteristics were identified from closed loop flight data as first-order low pass filters with time delays.

The second part addresses the flight dynamics of the closed loop system. The coefficients and derivatives were again identified from previous flight data. The identification was performed at two different airspeeds, 4 and 16 m/s, covering hover and cruise flight characteristics. Thanks to the velocity controller, changes concerning response characteristics between these flight states were remarkable small. In the third part the equations of motion are calculated by a simple kinematic model. Additionally wind is added to the airspeeds to gain velocities in the inertial coordinate system. Therefore high-resolution wind including gustiness was implemented. Wind speed and gustiness can be scaled and rotated as needed. Additionally a boundary layer model was implemented, reducing the wind speed depending on the aircraft's altitude and the roughness of the surface.

Results The velocity controller is able to mask the complexity of the tiltwing aircraft. Because of this feature, the system consisting of aircraft and velocity controller is well accessible to extensive but reasonable simplification. The aircraft's complex flight mechanics could be described in a rather simple simulation. All mission controller parts could be implemented to the simulation. This helped for comprehensive test of the mission control system at different wind conditions. Especially all safety procedures, which may be unsafe to test in flight, could be examined in detail.

5.2 Flight Tests

To validate the simulation environment and test selected aspects of the control system, flight tests were performed near Aachen. For different wind conditions the effect on the aircraft was compared to the simulation results. Also the effect on the maneuver performance could be observed in the simulation very similar to flight tests. In total the closed loop system characteristics are covered by the simulation in all different flight states, which made the simulation an important tool for the design of the controller.

As the higher modes of operation of the mission controller could be tested and validated in the simulation environment, the focus during flight tests was on the analysis of specific controller parts. In particular the precision landing was tested extensively in real flight. Influences of ground effect and boundary layer effects, had significant impact on the performance.

5.3 Parcel Delivery Operation

During the entire operation in the Bavarian Alps all flights were successfully performed in full automation. No operator intervention was necessary and no safety procedure was triggered. The overall unmanned aircraft system worked as expected even under rough weather conditions. Wind speed during take-off and landing at the ground level was 7 m/s at the maximum. Wind speed at flight level was only estimated from measured groundspeed and wind correction angle. The maximum wind speed observed during flights was 13 m/s at a wind correction angle of 31° and a groundspeed of 25.4 m/s.

The flight time of the operational flights differ in more than 140 s (355 up to 500 s), which indicates the influence of the wind on the flight duration. As the aircraft is controlled to fly at a constant airspeed, the groundspeed during cruise varies between 14 and 29 m/s. The vertical and horizontal track-error during cruise flight stay within the required limits of 10 m. The precision landing mechanism was validated by more than 130 landings. In [1] these precision landings were analyzed in reference to the wind condition. An evaluation of the maneuver controller and its performance can be found in [4]. In general all flights well matched previous simulations, as no unexpected effects could be observed.

6 Conclusion

In this paper a mission control system for fully automated tiltwing operation was presented. The DHL-Parcelcopter delivery operation included vertical take-off and landing and a straight lined maneuver for conversion from cruise flight and vice versa. The overall mission controller design was driven by constraints, such as precision landing on a small platform and flight beyond line of sight.

A design approach, including test flight in line of sight, was combined with a dedicated simulation environment. Controllers for different flight phases and their interaction were presented. The higher level controllers are based on a velocity controller, utilizing the complex flight mechanics of the tiltwing aircraft. The flight path controller consists of three sub-controllers, each operating during a different flight phase. A special focus was set on the mission controller, directing the different controllers during the flight phases. The attributes and procedures of the mission controller were presented in combination with external facilities to ensure safe operations. The mission definition data for standard operations and safety procedures were discussed.

The designed controller allowed automated flight beyond line of sight of the tiltwing aircraft during a 3-month trial operation in the Bavarian Alps. The functionality of the aircraft system was evaluated in a number of different test flights. The robustness and reliability of the mission control system was proven during the flight campaign. All deviations and overshoots stayed within the required limits. The different challenges due to high level of automation and flight in civil airspace were well managed by the control system.

References

1. Dobrev Y, Hartmann P, Schütt M, Moormann D (2016) Entwurf und Validierung eines Präzisionslandesystems für unbemannte Tiltwing-Fluggeräte. Deutscher Luft- und Raumfahrtkongress, Braunschweig, Germany
2. Frost CP, Franklin JA, Hardy GH (2002) Evaluation of flying qualities and guidance displays for an advanced Tilt-Wing STOL transport aircraft in final approach and landing. In: Biennial international powered lift conference, Williamsburg, US-VA. https://doi.org/10.2514/6.2002-6016
3. Hartmann P, Meyer C, Moormann D (2016) Unified approach for velocity control and flight state transition of unmanned Tiltwing aircraft. In: AIAA guidance, navigation, and control conference, San Diego, US-CA. https://doi.org/10.2514/6.2016-2101
4. Hartmann P, Schütt M, Moormann D (2017) Control of departure and approach maneuvers of Tiltwing VTOL aircraft. In: AIAA guidance, navigation, and control conference. US-TX (accepted for publication), Gaylord. https://doi.org/10.2514/6.2017-1914
5. Holsten J, Ostermann T, Moormann D (2011) Design and wind tunnel tests of a tiltwing UAV. CEAS Aeronaut J 2:69–79. https://doi.org/10.1007/s13272-011-0026-4
6. Holsten J, Hartmann P, Schütt M, Moormann D (2016) DHL Paketkopter 3.0 Automatische unbemannte Flüge ausserhalb der Sichtweite zwischen zwei Packstationen mit einem Tiltwing-Fluggerät, Deutscher Luft- und Raumfahrtkongress, Braunschweig, Germany
7. Hrishikeshavan V, Bawek D, Rand O, Chopra I (2013) Development of transition control methodology for a quad rotor-biplane micro air vehicle from hover to forward flight. In: AHS international specialists' meeting on unmanned rotorcraft, Scottsdale, US-AZ
8. DHL-Parcelcopter. http://www.dpdhl.com/en/mediarelations/specials/parcelcopter.html. 20 Oct 2016
9. Ostermann T, Holsten J, Dobrev Y, Moormann D (2012) Control concept of a Tiltwing UAV during low speed manoeuvring. In: Proceeding of the 28th international congress of the aeronautical sciences. Brisbane, Australia
10. Rothhaar PM et al (2014) NASA langley distributed propulsion VTOL Tilt-Wing aircraft testing, modeling, simulation, control, and flight test development. In: 14th AIAA aviation technology. integration, and operations conference, Atlanta, US-GA. https://doi.org/10.2514/6.2014-2999

11. Sato M, Muraoka K (2014) Flight controller design and demonstration of quad-Tilt-Wing unmanned aerial vehicle. J Guidance Control Dyn 38(6):1071–1082. https://doi.org/10.2514/1.G000263
12. Schütt M, Hartmann P, Moormann D (2014) Fullscale Windtunnel investigation of actuator effectiveness during stationary flight within the entire flight envelope of a Tiltwing MAV. In: Proceedings of international micro air vehicle conference. Toulouse, France, pp 77–83
13. Voget N, Binz F, Dobrev Y, Moormann D (2016) DHL Paketkopter 3.0 Auslegung und Implementierung eines Zuverlässigen Kommunikationsnetzes für den Sicheren Betrieb Unbemannter Fluggeräte ausserhalb der Sichtweite, Deutscher Luft- und Raumfahrtkongress, Braunschweig, Germany

Part IV
Space Applications

Attainable Landing Area Computation of a Lunar Lander with Uncertainty by Reachability Analysis

Yunus Emre Arslantas and Stephan Theil

1 Introduction

In recent years, advances of the proven technological heritage and new sophisticated tools led to more challenging space missions. Along with the government initiatives, private companies also increase their efforts for space exploration and exploitation. As a result of these contributions, concepts like reusable launch vehicles, asteroid mining and establishing bases on extraterrestrial space bodies are getting closer to reality. State of the art guidance and control algorithms are required for the success of these missions. In particular, the determination of hazards, the implementation of landing maneuvers and maximizing the likelihood of a safe landing is of paramount importance.

From the safety point of view it is important to ensure that the target landing area is suitable for landing due to surface properties and that is also within the reach of the lander. At this point two sets of information are important:

- **Hazard Maps**: obtained by optical sensors and contain information about the craters, the roughness of the landing surface and the slope of the terrain.
- **Attainable Map**: related to the physical limits of the lander like available on-board propellant, state constraints (final attitude and velocity for safe landing) and control constraints (avoiding saturation of controllers).

This paper proposes an algorithm to obtain the attainable map of a lunar lander for a generic lunar mission scenario using reachability analysis. Reachability deals

Y.E. Arslantas (✉)
Department of Mathematics and Computer Science, University of Bremen,
28359 Bremen, Germany
e-mail: yunusemr@uni-bremen.de

S. Theil
DLR Institute of Space Systems, Robert-Hooke Str. 7, 28359 Bremen, Germany
e-mail: Stephan.Theil@dlr.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_27

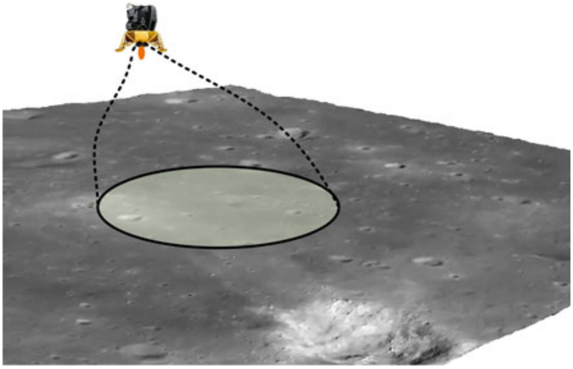


Fig. 1 Attainable landing area for a lunar lander

with the problem of whether from a given point, a system will eventually be able to reach another given point as illustrated in Fig. 1. Reachability analysis could either be used for verification or determination of the safety of a system (like computation of the safe landing envelope of an aircraft [1], the landing corridor of a reusable launch vehicle or reentry problems [2–4]. The reachability problem is handled by set-valued methods [5], level-set methods [6] or OCP based methods [7].

In order to obtain the reachable set (RS) of the lunar landing problem, the candidate landing area is characterized by discrete equidistant landing points. For each point a distance function, which acts as the objective function of an associated optimal control problem (OCP), is defined. These infinite dimensional OCPs are transcribed into a Nonlinear Programming (NLP) problem using Pseudo-Spectral methods (PSM).

PSM uses a linear combination of Legendre polynomials to obtain discretization points or nodes. This methodology could be classified into different groups according to the used discretization points, namely Legendre-Gauss, Legendre-Gauss-Radau or Legendre-Gauss-Lobatto [8]. In these approaches, the continuous functions are collocated using the corresponding nodes, leading to a sparse structure of the Jacobian. Finally the NLP is solved with commercial off-the-shelf solvers to obtain the state information at the landing points. The union of the points approximate the reachable landing area with an error, due to the discretization of the domain.

In order to verify the effect of uncertainties, which could be encountered during the mission phase, Monte Carlo simulations are performed. The main source of uncertainties are the initial state or system parameters. Later on, the probabilistic attainable maps are obtained by counting the number of successful landings on the candidate points. The rest of the paper is organized as follows:

Section 2 defines basic terms associated with reachability analysis. Section 3 describes the dynamical model of the lunar lander including the mission scenario and the associated constraints. Section 4 presents an optimization based methodology for the computation of reachable sets, the discretization of the infinite dimensional OCP and the design of the controller. Monte Carlo simulations with uncertainties and probabilistic attainable maps are discussed in Sect. 5. Finally Sect. 6 concludes the paper with a short summary.

2 Reachability

Consider a dynamical system with:

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t), t\right), \quad t \in \mathbb{I} = [t_0, t_f], \\ x(t_0) &= x_0 \in X, \end{aligned} \tag{1}$$

where $X \subset \mathbb{R}^n$ denotes the state space and $U \subset \mathbb{R}^m$ the input space. $x_0 = x(t_0)$ is the state at initial time t_0 while $x_f = x(t_f)$ denotes the final state. The control input is contained in the space of admissible control signals, that is $u(t) \in \mathscr{U} = \mathrm{L}^\infty([t_0, t_f], U)$. For each point in time, the solution or state trajectory $x(\cdot) : \mathbb{R}_+ \rightarrow X$ of (1) is an element of the state space $x(\cdot) \in X$.

Definition 1 Given x_0 and the set of all admissible $\mathscr{U}$, the reachable set of (1) at time t_f is defined as

$$\mathscr{R}_{tf}(x_0) = \{x_f \in X | \exists u \in \mathscr{U} \wedge \exists x(\cdot) \in X \text{ s.t. (1) holds } \wedge x_f = x(t_f)\}. \tag{2}$$

In other words, a reachable set includes the set of add final states of a given dynamical system for a final time and given constraints.

In order to determine an approximated RS, the two dimensional subset of the state space G_s is discretized by equidistant grid points g_s as illustrated in Fig. 2.

The OCP defined by (3) is solved to find control inputs that steer the system from the initial state to a sufficiently small neighborhood of a corresponding grid point g_s. If an optimal solution is found, the point is said to be reachable. Otherwise, if the solution cannot be found, the candidate point is excluded from the set. Instead, final states of the associated problem are included in the set. In other words, the solution which minimizes the distance between the candidate landing point on crossrange-

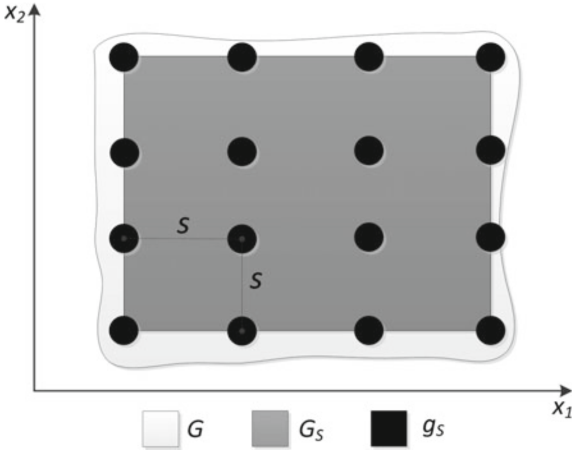


Fig. 2 Discretization of the landing area with uniform grid points

downrange plane and final states obtained from the solution of NLP are taken into account. The set which encloses all these final points constitutes the approximated RS. The OCP is formulated as

$$\begin{aligned} \min_{u \in \mathscr{U}} \quad & \frac{1}{2}\|x(t_f) - g_s\|_2^2 \\ \dot{x} &= f(x(t), u(t)), \text{ a.e. in } [t_0, t_f] \\ x_0 &\in X_0 \subseteq \mathbb{R}^{\mathrm{n}} \end{aligned} \tag{3}$$

with inequality constraints

$$\begin{aligned} x_L &\leq x(t) \leq x_U \\ u_L &\leq u(t) \leq u_U \end{aligned} \tag{4}$$

where $\dot{x} = f(x(t), u(t))$ is the equality constraint to be satisfied. For a given x_0, let $u^\star(\cdot; g_s)$ be the solution of (3), and $x^\star(\cdot; g_s)$ is the corresponding solution of the underlying system dynamics. Hence the approximation of the reachable set is defined as

$$\mathscr{R}_{t_{f_s}}(x_0) = \bigcup_{g_s \in G_s} \{x^\star(t_f; g_s)\}. \tag{5}$$

where $t_f \in \mathbb{I}$. State information at final time t_f are used to obtain an approximation of the RS.

3 Dynamical Model and Mission Scenario

The equations of motion of the lunar lander are taken from [9]. However, instead of a non-throttable main engine, the thrust level is modulated. Moreover, orthogonal thrusters are discarded in the model. Only one main engine provides thrust during the final phase of the landing, leading to following equations of motion

$$\begin{pmatrix} \ddot{d} \\ \ddot{h} \\ \ddot{c} \end{pmatrix} = \begin{pmatrix} \frac{r}{m\tilde{h}\mathrm{c}(\tilde{c})}(-T_1\mathrm{c}(\tilde{d}) + T_2\mathrm{s}(\tilde{d})) + 2\dot{d}(\frac{\dot{c}}{r}\mathrm{t}(\tilde{c}) - \frac{\dot{h}}{\tilde{h}}) \\ \frac{1}{m}[(-T_1\mathrm{s}(\tilde{d}) - T_2\mathrm{c}(\tilde{d}))\mathrm{c}(\tilde{c}) - T_3\mathrm{s}(\tilde{c})] + [(\dot{d}\mathrm{c}(\tilde{c}))^2 + \dot{c}^2]\frac{(\tilde{h})}{r^2} - \frac{MG}{(\tilde{h})^2} \\ \frac{r}{m(\tilde{h})}[(T_1\mathrm{s}(\tilde{d}) + T_2\mathrm{c}(\tilde{d}))\mathrm{s}(\tilde{c}) - T_3\mathrm{c}(\tilde{c})] - \frac{\dot{d}^2}{r}\mathrm{s}(\tilde{c})\mathrm{c}(\tilde{c}) - \frac{2\dot{c}\dot{h}}{\tilde{h}} \end{pmatrix} \tag{6}$$

where d is the downrange, c is the crossrange, h is the altitude and m is the mass of the lander. In addition the following abbreviations are used in (6)

$$\tilde{h} = h + r, \qquad \mathrm{s}(\tilde{d}) = \sin\frac{d}{r}, \qquad \mathrm{c}(\tilde{d}) = \cos\frac{d}{r}$$
$$\mathrm{s}(\tilde{c}) = \sin\frac{c}{r}, \qquad \mathrm{c}(\tilde{c}) = \cos\frac{c}{r}, \qquad \mathrm{t}(\tilde{c}) = \tan\frac{c}{r}.$$

The transformed thrust vector is given by

$$\begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} \cos(\beta - \frac{d}{r})T cos_\chi \\ \sin(\beta - \frac{d}{r})T cos_\chi \\ -T sin_\chi \end{pmatrix} \tag{7}$$

where T is the modulated thrust value. The thrust vector is defined in non-dimensional form. The first derivatives of the pitch angle β and yaw angle χ are also added to the system of differential equations, where ω_β and ω_χ are the commanded angular rate

$$\dot{\beta} = \omega_\beta, \quad \dot{\chi} = \omega_\chi. \tag{8}$$

The thrust dependent fuel consumption, depending on σ_{SFC}, is defined as

$$\dot{m} = -|T| \cdot \sigma_{SFC} \tag{9}$$

All states of the dynamical system defined by (6), (8), (9) are represented in the following state vector

$$x = (\dot{d}, \dot{h}, \dot{c}, d, h, c, \beta, \chi, m)^T \tag{10}$$

while the control vector is represented by

$$u = (T, \omega_\beta, \omega_\chi)^T. \tag{11}$$

3.1 Mission Scenario

The reachable sets for the lunar lander are obtained for the terminal landing phase of the mission. The spacecraft starts a landing maneuver from an altitude of 96 km and moves on an elliptical descent orbit with an altitude of 15 km. Powered descent starts at this altitude. The main thruster reduces the velocity of the spacecraft until an altitude of 2 km is achieved. At the approach phase, an attitude adjustment maneuver is performed between 2 km and 300 m. The final phase of the landing maneuver is supposed to start at an altitude of 300 m as shown in Fig. 3.

The initial conditions are defined by the state of the dynamical system at an altitude of $h_0 = 300$ m using the optimal trajectory in [9]. The mission requires the following initial and terminal conditions

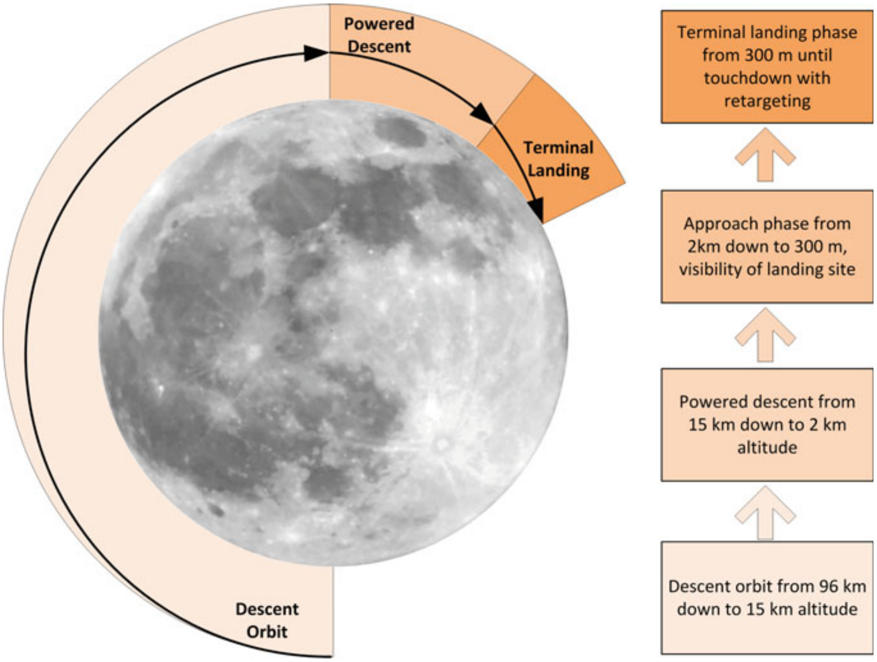

Powered Descent
Terminal Landing
Descent Orbit
Terminal landing phase from 300 m until touchdown with retargeting
Approach phase from 2km down to 300 m, visibility of landing site
Powered descent from 15 km down to 2 km altitude
Descent orbit from 96 km down to 15 km altitude

Fig. 3 Mission scenario for the powered descent and landing

$$x_0 = (\dot{d}_0, \dot{h}_0, \dot{c}_0, \text{free}, h_0, \text{free}, \ \beta_0, \ \chi_0, \ m_0)^{\mathrm{T}} \tag{12}$$

$$x_f = (0\ , 0\ , 0\ , \text{free}, 0\ , \text{free}, -\frac{\pi}{2}, \ \text{free}, m_f)^{\mathrm{T}}. \tag{13}$$

Additional constraints results from final landing condition. A landing is considered successful if the condition $|\Delta x(t_f)| \leq \Delta x_{max}$ holds with

$$\Delta x_{max} = (1{}^{\mathrm{m}}\!/_{\mathrm{s}}, 1{}^{\mathrm{m}}\!/_{\mathrm{s}}, 1\,{}^{\mathrm{m}}\!/_{\mathrm{s}}, d_f, 1\mathrm{m}, c_f, 10^\circ, 180^\circ, free). \tag{14}$$

The initial conditions, final conditions and constraints are summarized in Table 1, where x_u and x_l denote upper and lower bounds on states respectively for the whole time interval. Similarly u_0 and u_f represents initial and final control inputs, u_u and u_l are associated upper and lower bounds on the control inputs which are summarized in the Table 2.

In this setup, the mass of the lander is the sum of the propellant mass and the mass of the structure. The allowed minimum mass of the vehicle is defined as hard constraint during trajectory optimization.

Table 1 State constraints used for the solution of OCP in (3)

	x_0	x_f	x_l	x_u
$\dot{d}$(m/s)	5	0	Free	Free
$\dot{h}$(m/s)	–19	0	Free	Free
$\dot{c}$(m/s)	0	0	Free	Free
d(m)	0	Free	Free	Free
h(m)	300	0	0	Free
c(m)	0	Free	Free	Free
β(°)	–86	–90	–90	90
χ(°)	0	Free	0	180
m	0.5397	≤ 0.5	0.5	0.5397

Table 2 Constraints on the control inputs for the solution of OCP in (3)

	u_0	u_f	u_l	u_u
T	0	Free	0	1.222
ω_β(°/s)	0	Free	–2	2
ω_χ(°/s)	0	Free	–2	2

4 Optimal Control Problem

In this study the infinite dimensional OCP in (3) is converted into a corresponding finite dimensional NLP which is solved via a direct method. The basic principle is to collocate the differential equations, the cost functional and related constraints at a finite number of collocation points. After the continuous OCP is transcribed into a finite dimensional NLP, and the resulting set of nonlinear algebraic constraints are solved numerically.

The collocation points are obtained using linear combination of the roots of Legendre Polynomials or their derivatives. The Gauss Pseudospectral Method (GPM), Lobatto Pseudospectral Method (LPM) and Radau Pseudospectral Method (RPM) with its flipped version (f-RPM) are the well known Pseudospectral methods for discretization of continuous problem with respect to the time domain [10–12]. Pseudospectral methods are advantageous due to:

- their straightforward implementation leading to a sparse NLP structure after discretization of the OCP;
- the avoidance of the Runge phenomenon [13];
- exponential convergence behavior for smooth problems [8];

In this study, a flipped variation (f-RPM) of the RPM is used to transcribe the OCP in (3). The roots of the linear combination of Legendre polynomials are obtained by RPM and flipped in the domain [–1, 1]. SPARTAN (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory Analysis) is used for the discretization of the problem [14, 15].

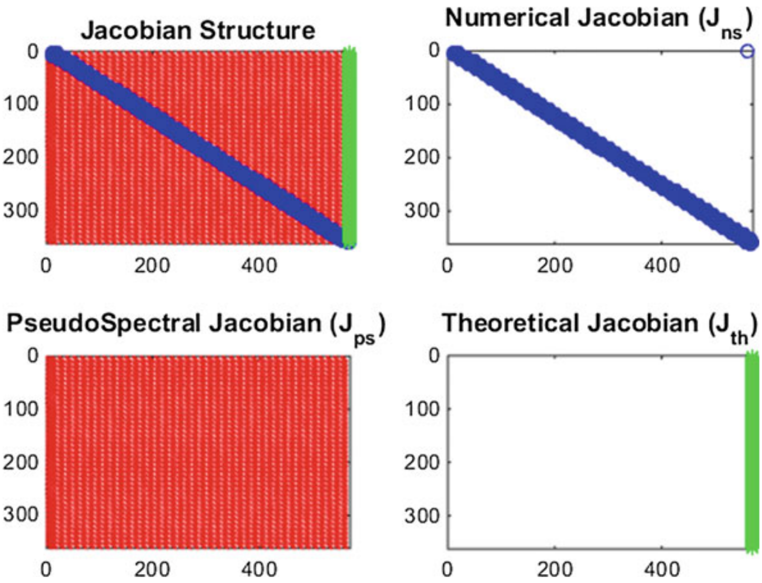


Fig. 4 Hybridization and representation of the different contributors of the jacobian $\mathscr{J}$ for the NLP

SPARTAN obtains the NLP by hybridization of the jacobian $\mathscr{J}$ by evaluating the different contributors as illustrated in Fig. 4. The methodology is described in [16], and tested for the computation of the reachable sets in [17, 18].

After solving the NLP of the associated OCP in (3), state information and control inputs for each grid point $g_s \in G_s$ are obtained. In addition to the exit flag of the solver, the result is also validated by propagating the control inputs by Runge-Kutta integration schemes and comparing it with the state information obtained within the same solution of the OCP. After the computations, the RS of the lunar lander is obtained including trajectories and control inputs for each of the related landing points with a final time $t_f = 40$ s.

4.1 Feedback Controller

The trajectory control of the lunar lander consist of a combination of feedforward and feedback controls as illustrated in Fig. 5.

For the purpose of the design of the feedback controller the dynamics of the lunar lander are represented by the nonlinear state space model as shown in (1). Assuming that the aforementioned state space representation is analytic and that a real landing trajectory is close to the optimal trajectory $\mathbf{x}(t) = \hat{\mathbf{x}}(t) + \delta\mathbf{x}(t)$, the linearization of (1) along $\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{u}}(t)$ yields

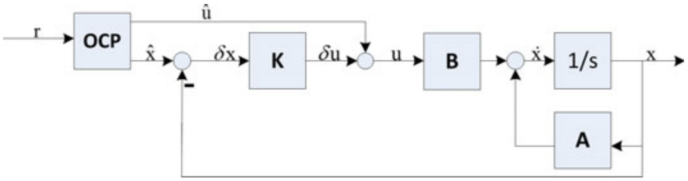


Fig. 5 Block diagram of the closed loop system

$$\dot{\delta\mathbf{x}}(t) = \mathbf{A}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) \tag{15}$$

with $\mathbf{A}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}},\hat{\mathbf{u}}} \in \mathbb{R}^{n\times n}$ and $\mathbf{B}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\hat{\mathbf{x}},\hat{\mathbf{u}}} \in \mathbb{R}^{n\times m}$, $t \in [t_0, t_f]$ and corresponding initial conditions. Considering a time variant control law for the linear time variant dynamical system leads to

$$\begin{aligned} \delta\mathbf{u}(t) &= -\mathbf{K}(t)\delta\mathbf{x}(t) \\ \mathbf{u}(t) &= \hat{\mathbf{u}}(t) - \mathbf{K}(t)\delta\mathbf{x}(t), \quad t \in [t_0, t_f] \end{aligned} \tag{16}$$

The changes in the dynamics of the lunar lander during a landing maneuver along a reference trajectory are supposed to be slow compared to the sampling rate of the discretized version of the OCP in (3). For this reason the gain matrix of the feedback controller $\mathbf{K}(t)$ is calculated only at each point t_i, $i \in \mathrm{I}$ with $\mathrm{I} = 1, ..., l$ of the control discretization by minimization of the cost function

$$\mathscr{J}(\delta\mathbf{x}, \delta\mathbf{u}) = \int_0^\infty \delta\mathbf{x}(\tau)^{\mathbf{T}}\mathbf{Q}(t_i)\delta\mathbf{x}(\tau) + \delta\mathbf{u}(\tau)^{\mathbf{T}}\mathbf{R}(t_i)\delta\mathbf{u}(\tau)\mathrm{d}\tau \tag{17}$$

for all $i \in \mathrm{I}$. Assuming the stabilizability of the underlying system, and positive semi-definite $\mathbf{Q}(t_i)$ and $\mathbf{R}(t_i)$ for all $i \in \mathrm{I}$, the feedback matrix that solves the OCP is given as a function

$$\mathbf{K}(t_i) = \mathbf{R}(t_i)^{-1}\mathbf{B}(t_i)^{\mathrm{T}}\mathbf{S}(t_i), \quad i \in \mathrm{I} \tag{18}$$

of the unique stabilizing solution $\mathbf{S}(t_i)$ of the algebraic Riccati equation

$$\mathbf{A}(t_i)^{\mathrm{T}}\mathbf{S}(t_i) + \mathbf{S}(t_i)\mathbf{A}(t_i) - \mathbf{S}(t_i)\mathbf{B}(t_i)\mathbf{R}(t_i)^{-1}\mathbf{B}(t_i)^{\mathrm{T}}\mathbf{S}(t_i) + \mathbf{Q}(t_i) = 0, \quad i \in \mathrm{I}, \tag{19}$$

such that the eigenvalues of $\mathbf{A}(t_i)\mathbf{B}(t_i)\mathbf{K}(t_i)$, $i \in \mathrm{I}$ are negative. A linear interpolation has been applied to the $\mathbf{K}(t_i)$, $i \in \mathrm{I}$ to obtain a time-continuous control law.

The main goal for the feedback control is to achieve the final landing point by following the reference trajectory while attenuating external disturbance or uncer-

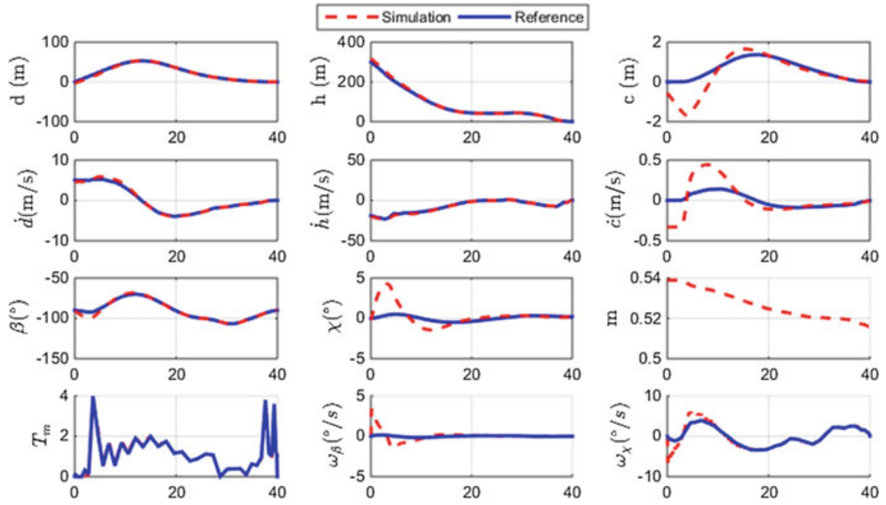


Fig. 6 Simulation of the closed loop system for the tracking reference trajectories

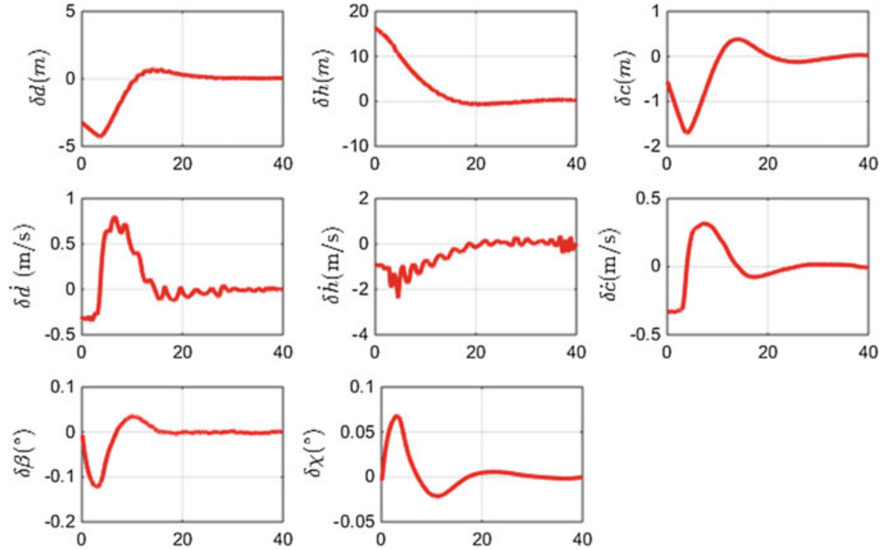


Fig. 7 State error between the reference trajectory and the simulated results

tainties for the initial state. The weighting matrices **Q** and **R** are kept constant during the simulations for each reference trajectory.

Figure 6 shows the simulation of the closed loop system with initial disturbance. Similarly, Fig. 7 shows the discrepancy between the reference trajectory and the simulated states of the closed loop system. It can be seen from the plots that the

Table 3 Uncertainties, results for the Monte Carlo simulations and properties of the maximum area ellipses inside the 100% successful landing region

	Range	Area ratio	Ellipse ratio	Center (m)	Orientation (°)	Semi-major Semi-minor Axis (m)
$\Delta \dot{d}$ (m/s)	[–2 2]	52.75	43.65	[100 0]	10	[1250 1100]
$\Delta \dot{h}$ (m/s)	[–2 2]	50.95	41.59	[50 0]	0	[1200 1100]
$\Delta \dot{c}$ (m/s)	[–2 2]	53.22	43.86	[50 0]	0	[1200 1150]
Δd (m)	[–30 30]	52.02	43.89	[100 0]	10	[1250 1100]
Δh (m)	[–30 30]	55.49	45.56	[100 0]	0	[1250 1150]
Δc (m)	[–30 30]	53.09	43.76	[50 0]	0	[1200 1150]
$\Delta \beta$ (°)	[–2 2]	90.96	64.29	[–100 0]	20	[1500 1350]
$\Delta \chi$ (°)	[–2 2]	89.30	61.34	[0 100]	40	[1600 1200]
$\boldsymbol{\Delta m}$	**5%**	**46.09**	**41.92**	**[100 0]**	**0**	**[1050 1050]**
$\Delta \sigma_{\mathrm{SFC}}$	3%	46.92	41.93	[100 0]	0	[1200 1100]
Total (#1)		29.18	34.81	[0 0]	0	[1000 950]
Total (#2)		**26.25**	**31.66**	**[100 –50]**	**0**	**[950 900]**

controller is able to track the given reference trajectory by attenuating the initial state error.

Monte Carlo simulations are carried out to obtain the successful attainable landing area. The uncertainties for the initial state and the specific fuel consumption are included as error sources during the simulations. Table 3 shows the range of the error for each state and specific fuel consumption. The uncertainties for the initial state are obtained from [19] by imposing the values with the same proportion. The uncertainty for the specific fuel consumption of the main engine σ_{SFC} is around 2% [20], but in this paper taken as 3% with a conservative assumption. The cost of attenuating the disturbance and uncertainty is the amount of extra propellant consumed by the main engine. After the Monte Carlo simulations, the results that satisfy the safe landing conditions are filtered leading to nonconvex successful landing regions. The success rate for each corresponding point inside the nonconvex set is evaluated by counting the number of simulations that achieves a successful landing with respect to total number of simulations. The final result is called the successful attainable landing area (SALA) of the lunar lander.

The borderlines that define the nonconvex hull of the successful landing area are obtained using alpha shapes. Finally, an ellipse with the maximum area is fitted inside the nonconvex set to evaluate the ratios of the reachable area, the successful landing area and the area of the ellipse. The parameters of the ellipse are discretized, and the maximum area is computed by a brute force method. Figure 8 shows the comparison of the reachable landing area without the controller, with the controller and ellipse that fits inside the nonconvex successful landing area.

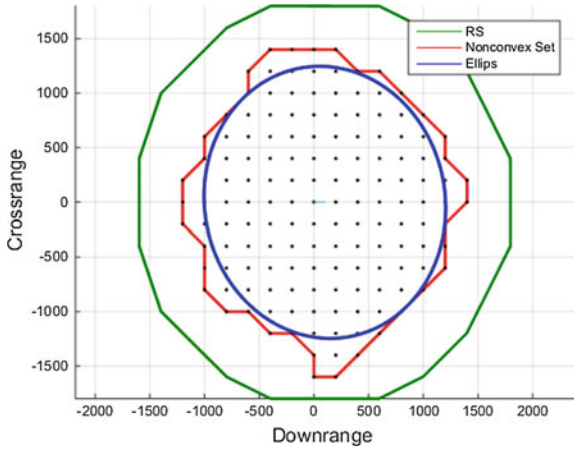


Fig. 8 Representation of the reachable landing area, nonconvex 100% successful landing region and maximum area ellipse

Table 3 summarizes the range of uncertainties and the results obtained from the Monte Carlo simulations. Two scenarios which include respectively 3 and 5% uncertainty in the specific fuel consumption together with the initial state errors are denoted as Total (#1) and Total (#2). Area ratio denotes the ratio of the area for the reachable set and the nonconvex region. Similarly, the ellipse ratio shows the proportion of the areas for the reachable set and the ellipse.

Table 3 also shows the center, semi-major and semi-minor axis as well as the orientation of the ellipse that fits inside the nonconvex attainable area.

The change of the successful landing area is most sensitive to the mass of the spacecraft or the amount of propellant considering the area ratios. As the uncertainty increases for the initial mass, the points along the boundary of the reachable set is not attainable by the lander during Monte Carlo simulations. Although the tracking controller tracks the reference trajectory, the successful landing condition is not satisfied in some cases. Specific fuel consumption is yet another important parameter affecting the area of the success region. For the cases, where the main engine consumes more propellant per unit thrust, the available propellant is not enough to steer the vehicle to the desired landing point.

On the other hand, the ellipse is not orientated w.r.t. the coordinate axis and the center of the ellipse lies close to the origin in downrange and crossrange coordinates, where the landing maneuver starts.

For the cases Total (#1) and Total (#2), all uncertainties are included in the simulations. For each single landing point inside the reachable set, 500 Monte Carlo simulations are performed. The results show that the successful landing area is 26.25% of the reachable landing area, which is the maximum landing region that can be achieved by the lander. The extra propellant consumed for correcting maneuvers result in a substantial decrease of the successful landing area.

5 Probabilistic Attainable Maps

Probabilistic attainable areas are obtained for each uncertainty source as illustrated with Figs. 9a and 13b. The successful landing area for all uncertainties together are represented with Fig. 14a, b. For each specific case, 500 Monte Carlo simulations are performed using a fixed structure of the feedback controller with the same gains.

As the initial error for the velocity along downrange is introduced, the successful landing region becomes smaller along crossrange, leading to a slender successful landing region as illustrated in Fig. 9a. Similarly, the initial error for the velocity along crossrange results in a smaller successful landing area, with unattainable regions close to the maximum and minimum crossrange of the reachable set in Fig. 10a. The initial error for the rate of altitude has a symmetric behavior in crossrange and downrange coordinates as shown in Fig. 9b. Although initial position error decreases the success rate of the landing, similar attainable landing maps are obtained in Figs. 10b and 11b. The results for the position error is also verified from the properties of the ellipses inside the nonconvex 100% successful landing region in Table 3.

On the other hand, the initial error for the attitude of the spacecraft in β and χ is not very critical in Fig. 12a, b. The success rate of landing decreases slightly for some points on the boundary of the reachable set. However, similar to the discussion for the properties of the ellipses, the mass and the specific fuel consumption of the vehicle is decreasing the success rate of landing considerably in Fig. 13a, b. Finally, for the cases in Total (#1) and Total (#2), the success rate of the landing decreases substantially with a symmetrical behaviour as illustrated in Fig. 14a, b.

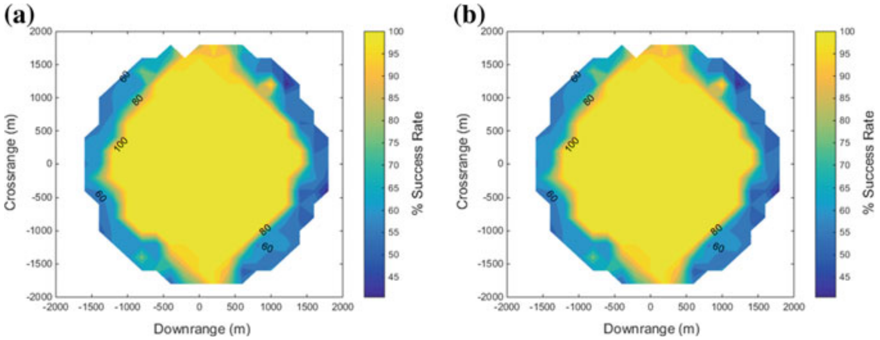


Fig. 9 SALA with initial error in downrange velocity $\dot{d}$ and altitude rate $\dot{h}$

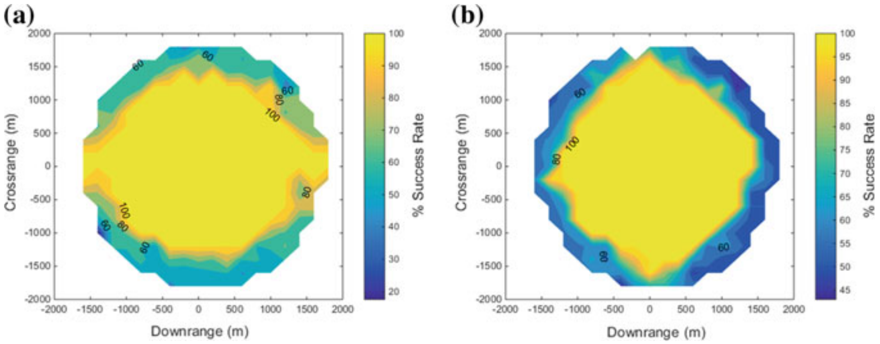


Fig. 10 SALA with initial error in crossrange velocity $\dot{c}$ and downrange d

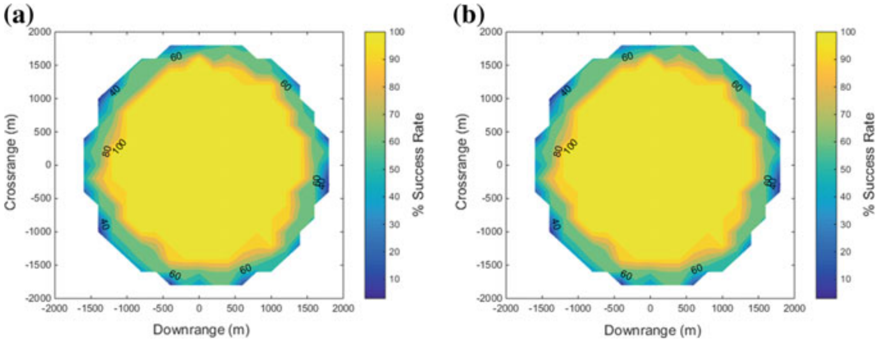


Fig. 11 SALA with initial error in altitude h and crossrange c

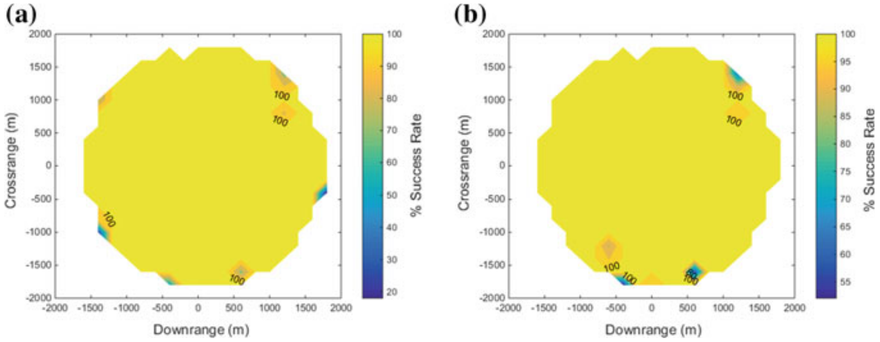


Fig. 12 SALA with initial error in pitch β and yaw χ

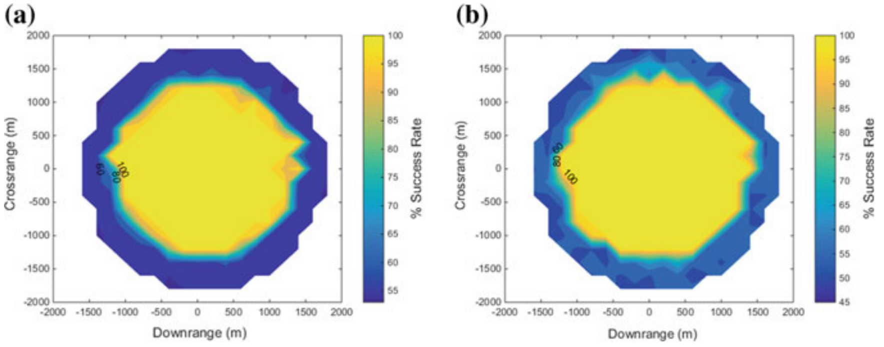


Fig. 13 SALA with initial error in mass m and specific fuel consumption σ_{SFC}

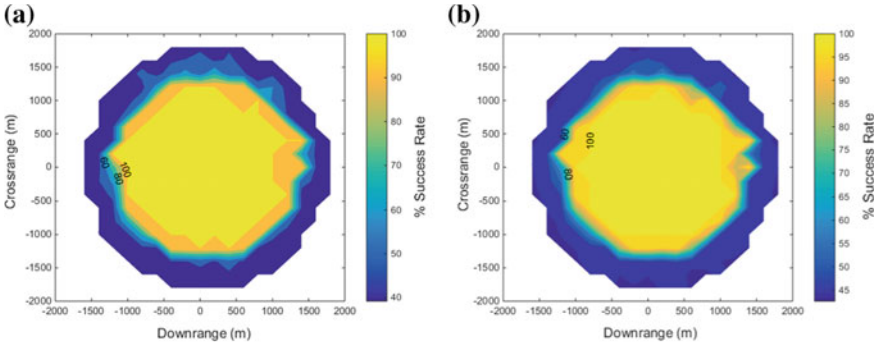


Fig. 14 SALA for scenario total (#1) and total (#2)

6 Conclusions

The attainable landing area is obtained with reachability analysis, leading to the attainable landing area of the lunar lander for the given mission scenario. A two degree of freedom controller is designed to stabilize the system and attenuate disturbances or initial uncertainties in the states. Monte Carlo simulations are performed to obtain the successful attainable landing area of the lander with probabilistic maps. In order to assess the results of the Monte Carlo simulations, a maximum area ellipse is fitted inside the nonconvex attainable landing area.

The probabilistic maps show the behaviour of the change of success rate with respect to the uncertainties. It is seen that, when uncertainties are introduced, the area of the attainable landing region decreases due to the amount of extra propellant consumed to track the reference trajectories in all scenarios. The error for the initial velocity in crossrange and downrange decreases the success rate of the landing starting from extreme points in the downrange and crossrange coordinates respectively. Initial position errors in three orthogonal coordinates have similar probabilistic maps, with similar characteristics of the maximum area ellipse. On the other hand, the initial attitude error has no significant effect on the probabilistic maps.

The lander is most sensitive to changes of the initial mass for a successful landing. The ratio of the attainable area with respect to the successfully attainable area is 46.09% for the uncertainty in the initial mass. The scenario in Total (#2) has the smallest attainable landing area, as expected, in which only 26.25% of the initial reference map is attainable when all uncertainties are introduced in the MC simulations.

The guidance and control algorithms also have an impact on the computed RS for the given dynamical system and uncertainties. The developed method provides a tool to evaluate the performance of the G & C algorithms and optimize the parameters to obtain the maximum landing area w.r.t the reference RS. Future work includes the assessment of different G & C algorithms and the optimization of the parameters for increasing the likelihood of a successful landing.

Acknowledgements This research is supported and funded by DLR (German Aerospace Center), DAAD (German Academic Exchange Service) Research Fellowship Programme and the Excellence Initiative of the German Research Foundation (DFG).

References

1. Baier R, Büskens C, Chahma IA, Gerdts M (2007) Aproximation of reachable sets by direct solution methods of optimal control problems. Optim Methods Softw 22(3):433–452
2. Kitsios I, Lygeros J (2005) Aerodynamic envelope computation for safe landing of the hl-20 personnel launch vehicle using hybrid control. In: Intelligent control, 2005. Proceedings of the 2005 IEEE international symposium on, mediterrean conference on control and automation (2005)
3. Benito J, Mease KD (2008) Characterizing the controllable entry states and reachable sites for planetary landing. In: 6th international planetary probe workshop. Atlanta, GA
4. Benito J, Mease KD (2010) Reachable and controllable sets for planetary entry and landing. J Guidance Control Dyn 33:2010
5. Aubin JP, Frankowska H (1990) Set valued analysis. Birkhauser (1990)
6. Mitchell IM (2002) Application of level set methods to control and reachability problems in continuous and hybrid problems. PhD thesis, Standford University (2002)
7. Baier R, Gerdts M (2009) A computational method for non-convex reachable sets using optimal control. In: Proceedings of the European control conference (ECC) 2009, August 23–26, Budapest, Hungary (2009)
8. Garg D (2011) Advances in global pseudospectral methods for optimal control. PhD thesis, University of Florida (2011)
9. Oehlschlägel T, Theil S, Krüger H, Knauer M, Tietjen J, Büskens C (2011) Optimal guidance and control of lunar landers with non-throttable main engine. In: Advances in aerospace guidance. Navigation and control. Springer, Berlin
10. Garg D, Patterson MA, Francolin C, Darby CL, Huntington G, Hager W, Rao AV (2011) Direct trajectory optimization and costate estimation offinite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method. Comput Optim Appl 49:335–358
11. Fahroo M, Ross F (2011) On discrete-time optimal conditions for pseudospectral methods. In: AIAA/AAS astrodynamics specialist conference, (Keystone, Colorado), August 2006

12. Huntington GD (2007) Advancement and analysis of a gauss pseudospectral transcription for optimal control problems. PhD thesis, Massachusetts Institute of Technology
13. Runge C (1901) Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. Zeitschrift für Mathematik und Physik 46:224–243
14. Sagliano M, Samaan M, Theil S, Mooij E (2014) Shefex-3 optimal feedback entry guidance. In: AIAA space 2014 conference and exposition
15. Sagliano M (2014) Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems. Oper Res Lett 42:213–216
16. Sagliano M, Theil S (2013) Hybrid jacobian computation for fast optimal trajectories generation. In: AIAA guidance, navigation and control (GNC) conference
17. Arslantas YE, Oehlschlägel T, Sagliano M, Theil S, Braxmaier C (2014) Approximation of attainable landing area of a moon lander by reachability analysis. In: 17th international conference on hybrid systems: computation and control
18. Arslantas YE, Oehlschlägel T, Sagliano M, Theil S, Braxmaier C (2014) Safe landing area determination for a moon lander by reachability analysis. In: International astronautical congress, Toronto
19. Kerr ML, Hagenfeldt M, Ospina JA, Ramn JM, Penin LF (2013) Esa lunar lander: approach phase concept and g and c performance. In: AIAA guidance, navigation, and control (GNC) conference, (Boston, MA), August 19–22 2013
20. Davidian KJ, Dieck RH, Chuang I (1987) A detailed description of the uncertainty analysis for high area ratio rocket nozzle tests at the NASA lewis research center, technical report, NASA

Analysis of Optimization Strategies for Solving Space Manoeuvre Vehicle Trajectory Optimization Problem

Runqi Chai, Al Savvaris and Antonios Tsourdos

1 Introduction

Trajectory optimization problems in terms of space vehicles [2, 4, 6, 9] have attracted significant attentions. One of the current objectives is the development of Space Manoeuvre Vehicles (SMV) for a dynamic mission profile. The Mach number and the flight altitude of the space vehicle vary largely during the whole flight phase, the aerodynamic feature of the vehicle has large uncertainties and nonlinearities. Due to these reasons, it is difficult to calculate analytical solutions of this type of problems. Therefore, numerical methods are commonly used to approximate the optimal solution. Numerical methods for solving optimal control problems are divided into two major classes: indirect methods and direct methods [1, 7, 10]. However, it is difficult to solve the trajectory design problem using indirect methods based on maximum principle. Hence, direct optimization method has been widely used for trajectory optimization.

All the direct methods aim to transcribe the continuous-time optimal control problems to a Nonlinear Programming problem (NLP). The resulting NLP can be solved numerically by well-developed algorithms such as gradient-based methods or derivative free algorithms. Sequential Quadratic Programming (SQP) and Interior point (IP) methods are used successfully for the solution of large scale nonlinear programming problems. The search direction in the SQP method is determined by solving the Quadratic Program (QP) problem whereas IP transcribes the inequality constraints to equality constraints by introducing slack variables. In recent years, derivative free

R. Chai (✉) · Al. Savvaris · A. Tsourdos
Cranfield University, Bedfordshire MK43 0AL, United Kingdom
e-mail: r.chai@cranfield.ac.uk

Al. Savvaris
e-mail: a.savvaris@cranfield.ac.uk

A. Tsourdos
e-mail: a.tsourdos@cranfield.ac.uk

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_28

methods has become more popular in the application of optimal control problems. However, the actual advantage of using a global method such as Genetic Algorithm (GA) and Differential Evolution (DE) is difficult to appreciate, in particular when stochastic procedures are applied. In this paper, a number of global search methods are tested for solving the SMV trajectory optimization problem.

Hereafter, the paper is organised as follows. Section 2 introduces the equations of motion, constraints of the SMV and the method used to discretize the optimal control problem. In Sect. 3, the general procedures of typical gradient-based methods and derivative free methods are detailed. Following that, Sect. 4 presents comparative results between the solution calculated using different strategies.

2 Problem Formulation

The mission scenario investigated in this paper focuses on the atmospheric skip entry, targeting the entry into the atmosphere down to a predetermined position and the required controls involved in returning back to low earth orbit. The overall mission can be found in Fig. 1. General skip reentry can be divided into five phases: initial roll, down control, up control, Kepler and final entry. Considering the mission of the SMV is to overfly the ground target with specific altitude, the most challenging phase 2 and 3 will be considered in this paper.

2.1 Equations of Motion

To formulate a Space Manoeuvre Vehicle's skip entry flight, the equations of three-dimensional motion for a point mass about a static Earth are integrated. The equations of motion can be summarised as:

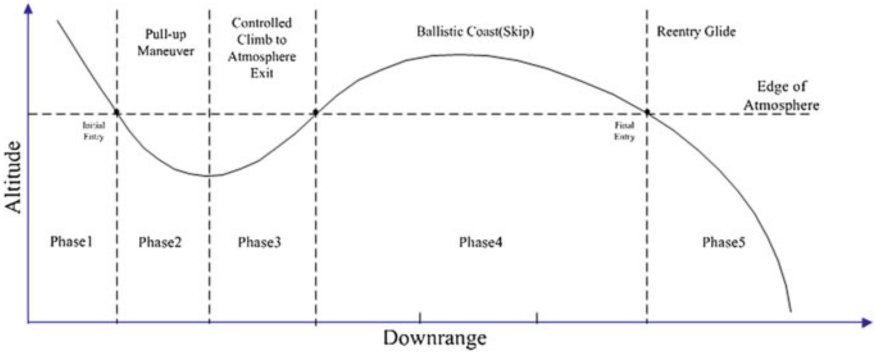


Fig. 1 Mission profile

$$
\begin{aligned}
\dot{r} &= V \sin\gamma \\
\dot{\theta} &= \tfrac{V\cos\gamma\sin\psi}{r\cos\phi} \\
\dot{\phi} &= \tfrac{V\cos\gamma\cos\psi}{r} \\
\dot{V} &= \tfrac{T\cos\alpha - D}{m} - g\sin\gamma \\
\dot{\gamma} &= \tfrac{L\cos\sigma + T\sin\alpha}{mV} + (\tfrac{V^2 - gr}{rV})\cos\gamma \\
\dot{\psi} &= \tfrac{L\sin\sigma}{mV\cos\gamma} + \tfrac{V}{r}\cos\gamma\sin\psi\tan\phi \\
\dot{m} &= -\tfrac{T}{I_{sp}g}
\end{aligned}
\tag{1}
$$

where r is the radial distance from the Earth center to the vehicle, θ and ϕ are the longitude and latitude, respectively. V is the Earth-relative velocity. The relative flight-path angle can be denoted as γ. ψ is the relative velocity heading angle measured clockwise from the north. m is the mass of vehicle and t is time. Angle of attack and bank angle are α and σ, separately. The thrust is defined as T. The states and controls are described as $X = [r, \theta, \phi, V, \gamma, \psi, m]^T$ and $U = [\alpha, \sigma, T]^T$. In the model given by Eq. (1), three autopilot equations are introduced using the technique of first order lag to describe the rate constraint of the controls.

$$
\begin{aligned}
\dot{\alpha} &= K_\alpha(\alpha_c - \alpha) \\
\dot{\sigma} &= K_\sigma(\sigma_c - \sigma) \\
\dot{T} &= K_T(T_c - T)
\end{aligned}
\tag{2}
$$

in which α_c, σ_c and T_c are the demand angle of attack, bank angle and thrust, respectively. The atmosphere model, lift L and drag D can be defined as:

$$
\begin{array}{ll}
g = \frac{\mu}{r^2} & \rho = \rho_0 \exp\frac{r - r_0}{h_s} \\
L = \frac{1}{2}\rho V^2 C_L S & D = \frac{1}{2}\rho V^2 C_D S \\
C_D = C_{D0} + C_{D1}\alpha + C_{D2}\alpha^2 & C_L = C_{L0} + C_{L1}\alpha
\end{array}
\tag{3}
$$

where $S = 2690\,\text{ft}^2$ is the reference area, ρ is the density of the atmosphere and $\rho_0 = 0.002378\,\text{slug/ft}^3$ is the density of the atmosphere at sea-level. $\mu = 1.4076539 \times 10^{16}\text{ft}^3/\text{s}^2$ is gravitational parameter of the earth. $r_0 = 20902900\,\text{ft}$ is earth radius, L and D are the lift and drag whereas C_L and C_D are lift and drag coefficient determined by angle of attack α and Ma, respectively.

2.2 Boundary and Path Constraints

In this paper, two types of constraint are considered in the skip entry process. To complete the mission, the boundary conditions for the states and controls are:

$$
\begin{aligned}
&[r, \phi, \theta, V, \gamma, \psi, m, \alpha, \sigma, T] \\
&= [r_0, \phi_0, \theta_0, V_0, \gamma_0, \psi_0, m_0, \alpha_0, \sigma_0, T_0] \\
&.[r(t_b), r(t_f)] = [r_b, r_f]
\end{aligned}
\tag{4}
$$

where t_b, t_f are time points for the SMV reaching the bottom point and going back to the final boundary conditions. More precisely, as shown in Fig. 1, t_b and t_f are the time points of the end of phase 2 and phase 3, respectively. Correspondingly, r_b and r_f are the altitude values at t_b and t_f, respectively. Based on the boundary conditions illustrated by Eq. (4), the whole process can be divided into two phases, the descent phase and exit phase.

To protect the structure integrity for the SMV, three path constraints including heating rate, dynamic pressure and load factor must be satisfied during the entire flight phase:

$$\begin{gathered}
\dot{Q}_d = K_Q \rho^{0.5} V^{3.07}(c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3) < \dot{Q_{dmax}} \\
P_d = \frac{1}{2}\rho V^2 < P_{dmax} \\
n_L = \frac{\sqrt{L^2+D^2}}{mg} < n_{Lmax}
\end{gathered} \tag{5}$$

where $c_0 = 1.067, c_1 = -1.101, c_2 = 0.6988, c_3 = -0.1903$ and $K_Q = 9.289 \times 10^{-9} BTU \cdot s^{2.07}/ft^{3.57}/slug^{0.5}$, respectively. Mission-dependent parameters are $Q_{dmax} = 200 BTU/ft^2/s$, $P_{dmax} = 13406.4583 Pa$ and $n_{Lmax} = 2.5$ representing the allowable maximum heating rate, dynamic pressure and acceleration, respectively.

2.3 Cost Function

According to the mission requirement given by the industrial sponsor of this project, to complete the mission in the shortest time, minimizing the mission duration (e.g. t_f) is chosen as the objective function. Let J represent objective function:

$$J = \min t_f \tag{6}$$

By setting the cost function given by Eq. (6), the SMV trajectory problem can be considered as an optimal control problem which has minimum cost function value and satisfies the initial and terminal variables constraints, three path constraints and dynamic equations.

2.4 Discrete Method

The SMV optimal control problem is discretized using a direct multiple shooting method. The basic idea of the direct multiple shooting method is to transform the original optimal control problem into NLP by parameterizing only the control variables. The controls can be approximated by interpolation at the discretized time nodes $[\tau_1, \tau_2, \ldots, \tau_N]$. Then the equation of motion are integrated with a fourth order Runge–Kutta method. The approximation of controls are:

$$u(\tau) \approx U(\tau) = \sum_{i=0}^{N} U_i L_i(\tau) \tag{7}$$

where $L_i(\tau)$ is a basis of Lagrange polynomials. After using direct multiple shooting method, the resulting NLP problem is solved by applying different optimization strategies illustrated in Sect. 3.

3 Optimization Strategies

A key ingredient to solve optimal control problems is the ability of solving NLP problems. Numerical methods for solving NLP fall into categories: gradient-based methods and derivative free algorithms.

3.1 Gradient Based Methods

3.1.1 Sequential Quadratic Programming

The most commonly used gradient-based methods are SQP and IP or barrier methods. The aim for SQP algorithm is to transform the original problem to a series of QP subproblems by approximating the augmented Lagrangian quadratically and linearizing the constraints using Taylor expansion. The resulting augmented Lagrangian is:

$$L(x, \lambda, u) = f(x) + \lambda^T h(x) + u^T g(x) \tag{8}$$

Using quadratic model to approximate Eq. (8), the QP subproblem is:

$$\begin{gathered} \min \tfrac{1}{2} d^T H(x_k, \lambda_k, u_k) d + \nabla f(x_k)^T d \\ h(x_k) + \nabla h(x_k) d = 0 \\ g(x_k) + \nabla g(x_k) d \leq 0 \\ d \in \Re^n \end{gathered} \tag{9}$$

where a (x_k, λ_k, u_k) presents the current iterate point whereas $H(x_k, \lambda_k, u_k)$ is the Hessian matrix. Commonly, the Hessian is calculated using $H(x_k, \lambda_k, u_k) = \nabla_{xx} L(x_k, \lambda_k, u_k)$ or a suitable approximation defined by the user. $\nabla h(x_k)$ and $\nabla g(x_k)$ are the Jacobian matrix of the vector of equality constraints and inequality constraints, respectively. The index k stands for the number of iteration for the optimization algorithm and $k = 0, 1, 2, \ldots$.

If the active set is defined as Λ, a sequence of linear equations are constructed as Karush-Kuhm-Tucker (KKT) system. Then by using Newton method, the KKT condition of Eq. (9) can then be calculated.

3.1.2 Interior Point Method

Another well-known and effective algorithm is the IP method. Numerous updates and modifications have been done on this approach during the last several decades. The ability for IP converging to a stationary point can be guaranteed theoretically. Before applying the IP to the general form of problems, Eq. (8) should be transformed by introducing $s \in \Re^m$, $\tau > 0$ and:

$$f_\tau(x) = f(x) + \tau \Sigma_{i=1}^{l} h_i(x) \tag{10}$$

The IP strategy consists of reducing the inequality constraints in Eq. (8) using slack variables $s = (s_1, s_2, \ldots, s_m)$, where all the elements in the vector should be positive. Then, the modified problem can be summarised as:

$$\begin{gathered} \min f_\tau(x) - \mu \Sigma_{j=1}^{m} log(s_j) \\ g(x) + s = 0 \\ x \in \Re^n, s \in \Re^m \end{gathered} \tag{11}$$

And the augmented Lagrangian for Eq. (11) is:

$$L(x, s, \lambda, u) = f_\lambda(x) - \mu \Sigma_{j=1}^{m} log(s_j) + u^T(g(x) + s) \tag{12}$$

In Eq. (11), the term μ stands for a barrier variable and the smaller it is, the closer are the solutions. Both SQP and IP are using Newton iteration to get KKT system and the converge solution.

3.2 *Derivative Free Algorithms*

In this paper, four derivative free global search algorithms are investigated: Genetic Algorithm (GA) and Differential Evolution(DE) that belong to the generic class of Evolutionary Algorithms (EA), Particle Swarm Optimization (PSO) that belongs to the class of agent-based algorithms, and Artificial Bee Colony (ABC) that is classified to the colony-based algorithms.

3.2.1 Genetic Algorithm

GA [3] is one kind of evolution algorithms, which generates solutions to optimization problems taking inspiration from the natural selection and survival of the fittest in the biological world. Each iteration of a GA involves a competitive selection that eliminates poor solutions. It is regarded as one of the most robust and reliable optimization algorithms which has no requirement for gradient information and initial guess. Recombination and mutation are applied to generate new solutions so that the population can have more diversity. As for the control parameters of GA, the population size is given as 500 individuals. Single values are used for the crossover and mutation probability, $CR = 0.7$ and $p_i = 0.5$, respectively.

3.2.2 Differential Evolution

The main attempt of DE [9] is to calculate the variation vector $v_{i,G+1}$ of a solution vector $x_{i,G+1}$ by taking the weighted difference between two additional solutions. This can be described as:

$$v_{i,G+1} = e[(x_{i3,G} - x_{i,G}) + F(x_{i2,G} - x_{i1,G})] \tag{13}$$

where $F = 0.7$, i is integer number randomly chosen within the interval $[1, NP]$. The equation of e can be written as:

$$e(x) = \begin{cases} 1, \; rand \leq CR; \\ 0. \; rand > CR. \end{cases} \tag{14}$$

where *rand* is a random number within [0, 1]. The selection process is largely depended on the fitness function defined by the user. If the new candidate can have a better fitness value then it can be selected to the next generation.

3.2.3 Particle Swarm Optimization

PSO [8] is a population-based derivative free optimization algorithm developed in 1995. PSO was inspired by the social behaviour of bird flocking or fish schooling. The general concept of PSO consists of changing the velocity of every candidate at each iteration. The new individual in the next generation can be calculated by:

$$v_{i,G+1} = \omega v_{i,G} + u_{i,G} \tag{15}$$

where ω is a weighting function which is proportional to the number of iterations G. The process has two stochastic components given by the two random number r_1 and r_2. The corresponding terms are elastic component controlled by $c_1 = 2$ and convergence term controlled by $c_2 = 2$. The first term tends to recall the individual back to the old position whereas the second term drives the entire population toward convergence. The search is applied until a stopping condition is satisfied.

3.2.4 Artificial Bee Colony

The artificial bee colony (ABC) algorithm was originally presented by Dervis Karaboga in 2007 [5] and was inspired by the collective behavior of honey bees.

The ABC algorithm has been tested to exhibit a good performance in the function optimization problems. For each iteration of the ABC algorithm, both global search and local search are conducted so that the probability of achieving the optimum is increased dramatically. In each iteration of the algorithm, the searching principle is defined as:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{16}$$

where v denotes the new position. Such a searching strategy adaptively reduces the searching step when the candidate approaches the optimal solution. The control parameters for ABC algorithm are $NP = 200$, $Iter = 100$ and $Limit = 10$, respectively.

4 Simulation Results

4.1 Parameters Setting

The initial, terminal boundary conditions, box constraints and aerodynamic coefficients of the skip process can be found in [2]. Comparative simulations using classical gradient optimization techniques and derivative free optimization methods are presented. It should be noticed that only the skip entry phase shown in Fig. 1 is taken into account in the paper. The initial altitude is around 80 km, where is the assumed edge of the atmosphere.

4.2 Time History of the State and Control for Different Methods

The results of optimal trajectories are shown in Figs. 2, 3, 4, 5, 6, 7, 8 and 9.

4.3 Analysis of the Solutions

From Figs. 2, 3, 4, 5, 6, 7, 8 and 9, the general trend of trajectories is split into two phases: descending and climbing.

(1) Descending phase: In order to achieve the target position (around 164000 ft altitude) and minimize the mission time (since the overall objective is to minimize the time duration, it is equivalent to minimize the time duration for each phase), Fig. 2 shows that the SMV goes down directly at the beginning of the mission. To avoid path constraints becoming active, angle of attack should increase to slow down the vehicle so that the heating and dynamic pressure do not increase significantly.

Fig. 2 Altitude generated by using gradient methods and derivative free methods

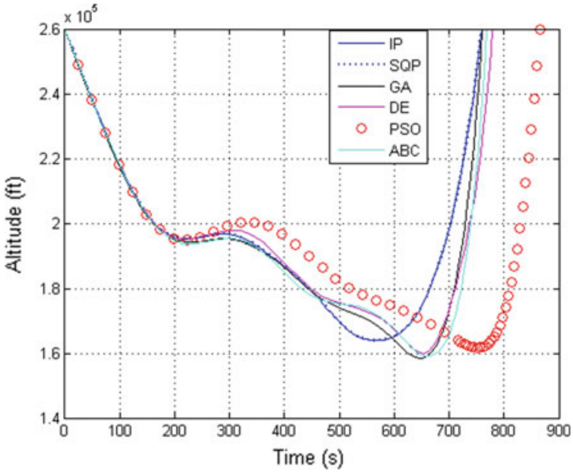
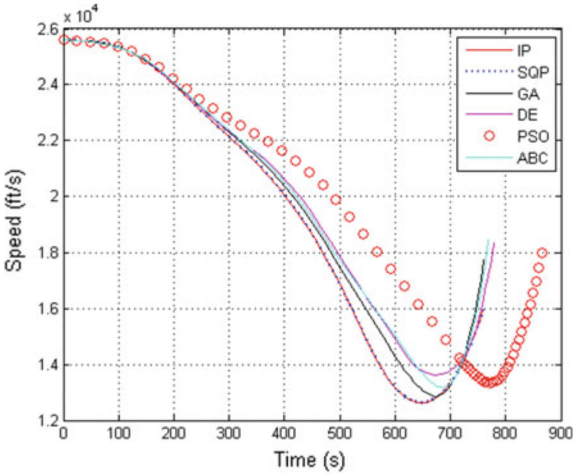


Fig. 3 Speed generated by using gradient methods and derivative free methods



(2) Climbing phase: After reaching the target point, the vehicle fires its engine so that the SMV can have enough kinetic energy to go back to the final point. With the decreasing of air density and mass, the aerodynamic heating, dynamic pressure and load factor will decrease during the climbing phase. The trend of angle of attack can be found in Fig. 4 where the angle of attack is increased during the whole climbing phase. This is because in the climbing phase, without violating path constraints, it can have positive influences in terms of acceleration.

With regard to the performance of different methods, all the global approaches manage to generate skip entry trajectories between the predetermined initial position and terminal position without violating the path constraints. This can be seen from Figs. 6, 7 and 8. When the nonlinearity of the cost functions or path constraints become higher, which means it is difficult to calculate the gradient information

Fig. 4 Angle of attack generated by using gradient methods and derivative free methods

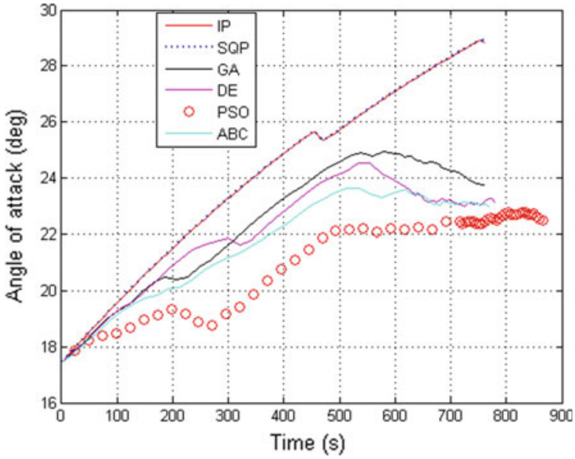
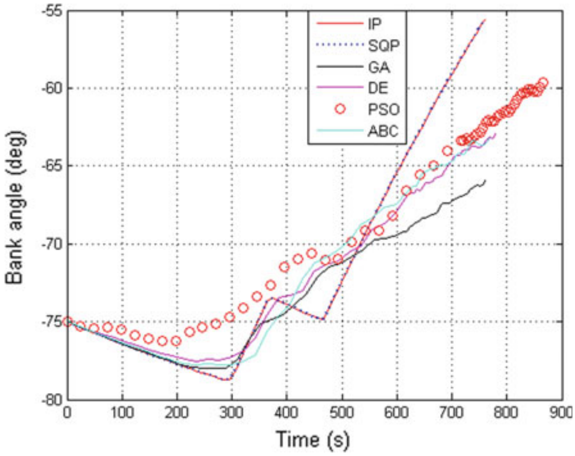


Fig. 5 Bank angle generated by using gradient methods and derivative free methods



for gradient techniques, the global methods become the only way to solve the SMV trajectory optimization problem. However, there are some limits for global strategies. As can be seen from Figs. 2, 3, 4, 5, 6, 7, 8 and 9, the results are significantly different between PSO-based method and others. This can be explained that in this case, the initial guess generated by the PSO-based method is not close enough to the optimal solution. Moreover, to combine the optimization processes with discrete methods, global techniques cannot be as flexible as gradient methods. Also, it is hard to verify the optimality for the solutions from global techniques whereas the SQP and other gradient methods have KKT conditions.

Consequently, all the figures provided above confirm the feasibility of the gradient and derivative free algorithms. By using different optimization strategies, the SMV can reach the target position without violating three path constraints and boundary conditions.

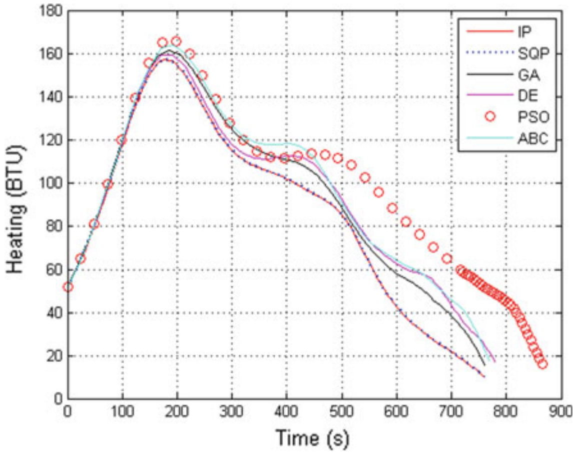


Fig. 6 Heating generated by using gradient methods and derivative free methods

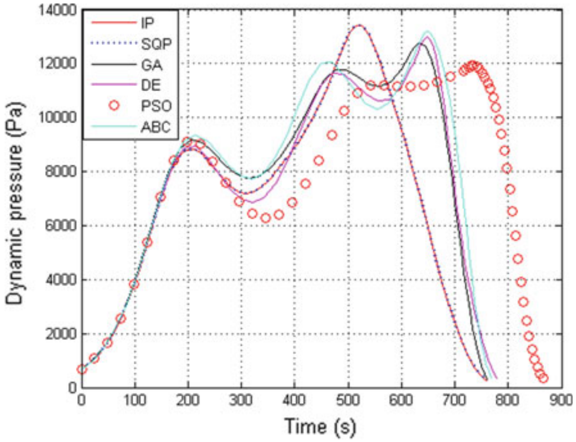


Fig. 7 Dynamic pressure generated by using gradient methods and derivative free methods

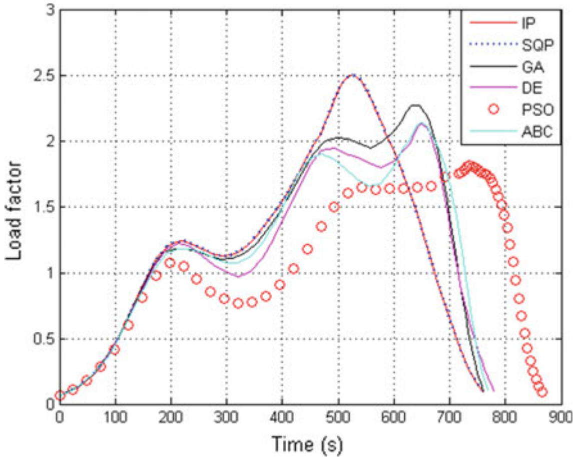


Fig. 8 Load factor generated by using gradient methods and derivative free methods

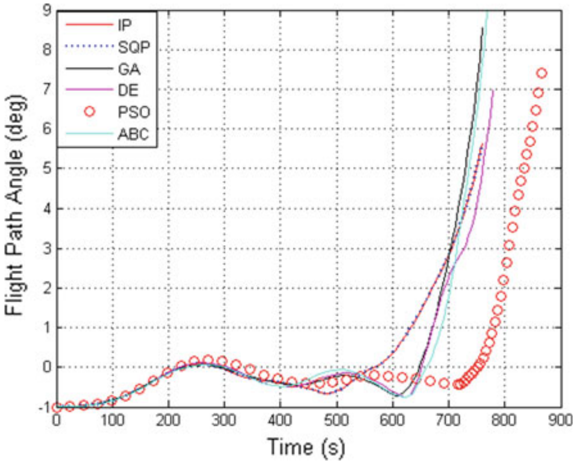


Fig. 9 Flight path angle generated by using gradient methods and derivative free methods

5 Conclusions

In this paper, the gradient-based and derivative free algorithms are applied to solve SMV trajectory design problem. In order to transform the continuous optimal control problem to static NLP problem, direct multiple shooting method is implemented to discrete the equations of motion and path constraints. Simulation results indicated that the proposed two kind of strategies can generate feasible solution for the trajectory design problem. By applying the gradient-based method, the number of iterations, function evaluations, and computational time can be decreased compared with derivative free methods. Therefore, although the solutions generated from derivative free methods can be accepted, there are still a lot of room for improvement in terms of using these techniques in trajectory optimization problems.

References

1. Betts JT (1998) Survey of numerical methods for trajectory optimization. J Guidance Control Dyn 21(2):193–207. https://dx.doi.org/10.2514/2.4231
2. Chai R, Savvaris A, Tsourdos A (2016) Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method. Acta Astronautica 123:62–70. https://dx.doi.org/10.1016/j.actaastro.2016.02.020
3. Dingni, Z., Yi, L.: RLV reentry trajectory optimization through hybridization of an improved GA and a SQP algorithm. In: Guidance navigation, and control and co-located conferences. American Institute of Aeronautics and Astronautics (2011). https://dx.doi.org/10.2514/6.2011-6658
4. Gan, C., Zi-ming, W., Min, X., Si-lu, C.: Genetic algorithm optimization of RLV reentry trajectory. In: International space planes and hypersonic systems and technologies conferences. American Institute of Aeronautics and Astronautics (2005). https://dx.doi.org/10.2514/6.2005-3269

5. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J Glob Optim 39(3):459–471. https://dx.doi.org/10.1007/s10898-007-9149-x
6. Kenan, Z., Wanchun, C.: Reentry vehicle constrained trajectory optimization. In: International space planes and hypersonic systems and technologies conferences. American Institute of Aeronautics and Astronautics (2011). https://dx.doi.org/10.2514/6.2011-2231
7. Peter, G., Klaus, W.: Trajectory optimization using a combination of direct multiple shooting and collocation. In: Guidance, navigation, and control and co-located conferences. American Institute of Aeronautics and Astronautics (2001). https://dx.doi.org/10.2514/6.2001-404710.2514/6.2001-4047
8. Rahimi A, Dev Kumar K, Alighanbari H (2012) Particle swarm optimization applied to spacecraft reentry trajectory. J Guidance Control Dyn 36(1):307–310. https://dx.doi.org/10.2514/1.56387
9. Rajesh, A.: Reentry trajectory optimization: evolutionary approach. In: Multidisciplinary analysis optimization conferences. American Institute of Aeronautics and Astronautics (2002). https://dx.doi.org/10.2514/6.2002-5466
10. Reddien GW (1979) Collocation at gauss points as a discretization in optimal control. SIAM J Control Optim 17(2):298–306. https://dx.doi.org/10.1137/0317023

A Space-Borne GNSS Receiver for Evaluation of the LEO Navigation Based on Real-Time Platform

Hung-Yuan Chang, Wen-Lung Chiang and Kuo-Liang Wu

A space-borne GNSS receiver is indispensable for comprehensive validation of performance of the proposed navigation function on a test bench mainly comprising an Attitude Control and Dynamics/Environments Software Simulator (hence ACDESS) running under a RTOS. In this paper, an ACDESS platform built on the PXI system manufactured by National Instruments is proposed. The platform consisting of necessary interfaces for communication between two PXI sub-platforms works with a GNSS RF signal generator which feeds the motion data to RF signal controller in real-time. The proposed NSPO ESGNSSR is aiming for closing the control loop in this test bench. The end-to-end tests of NSPO ESGNSSR in LEO satellite scenarios achieving a positioning accuracy better than 8 m (1σ, 3D) suggests its promising perspectives for NSPO's future space missions. All responses in the loop meet the mission requirements as well.

1 Introduction

Today, the global positioning system (hence GPS) receiver plays a more significant role in the autonomous competence enhancement of navigation and the facilitation of operation for a variety of spacecrafts and rockets [1, 2]. Thus, many key players in the global space arena pour numerous resources into the research and development of

H.-Y. Chang (✉) · W.-L. Chiang · K.-L. Wu
Flight Control Division, NSPO, HsinChu, Taiwan
e-mail: yuan@nspo.narl.org.tw

W.-L. Chiang
e-mail: peterchiang@nspo.narl.org.tw

K.-L. Wu
e-mail: alexwu@nspo.narl.org.tw

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_29

Table 1 Comparision of ESGNSSR and other space-borne GNSS receivers

Model	SEGNSSR	Topstar 3000	SGR-20	Mosaic
Manufacturer	NSPO	Thales Alenia	SURRY	EADS-Astrium
Position accuracy	8 m, 3D, 95%	10 m, 3D, 1s*	10 m, 3D, 95%	10 m, 3D**
Velocity accuracy	5 cm/s, 3D, 95%	1 cm/s, 3D, 1s*	15 cm/s, 3D, 95%	1 cm/s, 3D**
Time transfer with OCXO	80 ns, 95%	200 ns, 3s*	500 ns, 95%	100 ns**
Dimension	150*138*38.5 mm^3	276*117*170 mm^3	160*160*50 mm^3	242*262*25 mm^3
Power consumption	3.5 W	16 W	5.5 W	10 W
Weight (w/o ant.)	0.6 kg	3 kg	0.95 kg	0.7 kg
Operation temperature	–25 ~ 60 °C	–15 ~ 50 °C (operating) –25 ~ 60 °C (start-up)	–20 ~ 50 °C	–25 ~ 60 °C
Vibration	15 g	13 g	15 g	–
Mission involved		FORMOSAT-2, KOMSAT-2, AMS02, PROBA-2...	UoSAT1 UoSAT12, PROBA-1, BILSAT-1, 14 units flown (2009)	TerraSAR-X, SARLupe, Aeolus, TanDEM-X...

the autonomous space-borne GNSS receivers, including NSPO of Taiwan [3]. A brief comparison of several current space-borne GNSS receives including the ESGNSSR is summarized in Table 1 [4–6]. The ESGNSSR may be competitive among them. The flight model of this receiver manufactured in the early 2015 passed a series of comprehensive functional tests and environmental acceptance tests, etc. which were completed by the end of 2015. One mission considered using ESGNSSR during its development phase is the FORMOSAT-7/COSMIC-2 [7]. A real-time platform equipped with ESGNSSR presented in this paper is used to verify the navigation requirements of this LEO satellite. Moreover, the environmental stress screening process of ESGNSSR will be concisely introduced in this paper as well.

In response to the booming global navigation satellite systems, NSPO is gradually enhancing the capability of L1-GPS receiver and turning it to be a high-precise navigation unit, a multi-mode and multi-band receiver, and even a science payload, the reflectometry receiver of a global navigation satellite system. The fundamental purpose of this extension study is to port some software algorithms, such as signal acquisition and correlation, routinely repetitive computing, and time-consuming functions to the FPGA whose CPU, a digital signal processor (DSP), is solely dedicated to parts of the tracking algorithm, whole navigation algorithm, total orbit propagation algorithm, and so on for the sake of reducing the CPU workload. Due to the fast-paced development and evolution of the FPGA nowadays, the new system architecture upgradable via an FPGA should be able to achieve the goal of being a high-precision navigation receiver, and even a scientific receiver. Last but not least, the test results show that the new system architecture not only retains the original overall performance, but also sets aside more resources available for possibility of future extension.

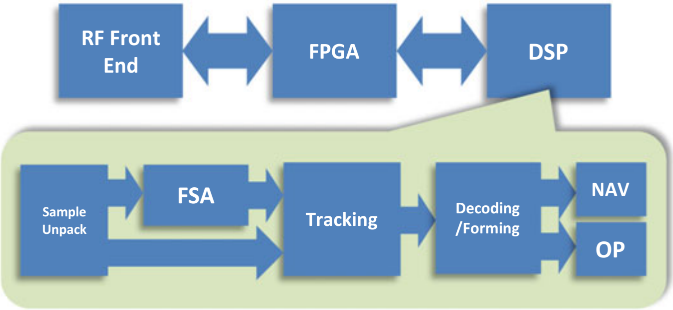


Fig. 1 The key parts of the entire SGR and the calculation process of software algorithm

1.1 Software-Defined GPS Receiver

The predecessor of this ESGNSSR is a software-defined GPS receiver (SGR) in which the L1-GPS baseband signal processing is performed in DSP resulting in 50% of its throughput consumed [8]. The key parts of the entire SGR, the relationships among them, and the calculation process of software algorithm in DSP are shown in Fig. 1.

The Sample Unpack module is mainly for trimming RF raw data and uses the fixed-point DSP instruction set to further achieve the goal of data arrangement. The FSA (Fast Search Algorithm) module is mainly responsible for the signal Fourier transformation, and screening out preliminary satellite signals. The Tracking module takes care of the signal correlation and signal phase-locked loop. Decoding/Forming module is used to decode the used satellite data. Both NAV (Navigation) and OP (Orbit Propagator) modules are in charge of calculating decoded data for navigation.

The entire software architecture is designed to schedule various on-demands computing for real-time tasks according to the weight of each function. As shown in Fig. 2, an external interrupt source is required for driving EDMA to move the RF data from FPGA to DSP. The real-time task in entire software is triggered by EDMA. The execution of the Tracking task in this software system must be completed within 1 ms; otherwise the post-processing results such as NAV and OP will lose reliability. The calculations of NAV and OP are allocated to use the remaining time after the Tracking task is complete in each millisecond. The lowest-weight function, FSA, is only responsible for searching new visible satellites and generating extra information in data pool waiting to be used when system is in idle state

In this SGR, the FPGA in the whole system only plays one role in the data communication and coordination between the DSP and other peripherals. There is not any logic implementation of the software algorithm relevant to the L1-GPS receiver implemented in FPGA. All the logic functions implemented in the FPGA are shown in Fig. 3. The SPI (Serial Programing Interface) module acts as a user interface for setting the operating parameters of RFFE. The CDET (Command Detector) module provides a user interface to execute special instructions. The Wait CE (Chip Enable) module can be used to adjust bus timing.

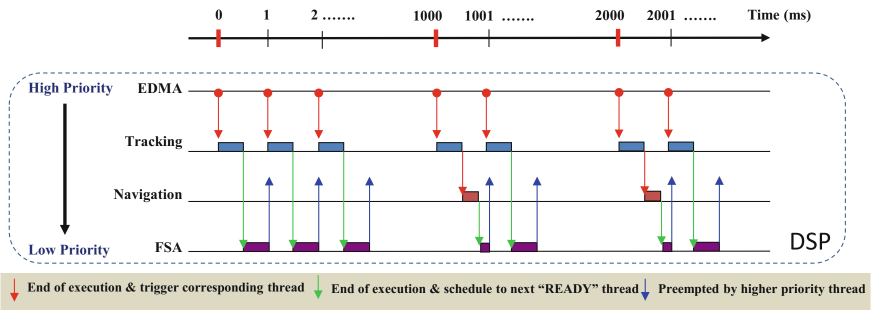

0
1
2 ....
1000
1001 ....
2000
2001 ....
Time (ms)
High Priority
Low Priority
EDMA
Tracking
Navigation
FSA
DSP
End of execution & trigger corresponding thread
End of execution & schedule to next "READY" thread
Preempted by higher priority thread

Fig. 2 The real-time scheduling of tasks

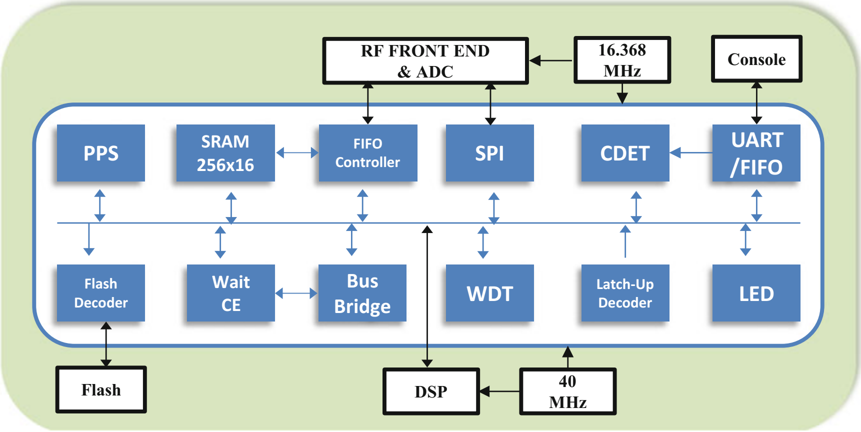


Fig. 3 The block diagram of original logical function

It may be a good entry point for understanding the purpose of this paper by considering the current usage of system resources. The actual utilization of DSP bus reached 46.9%, and the real-time tasks also occupied up to 20% of CPU utilization as shown in Fig. 4.

By comparison with the FPGA, the usage of logic gates is currently about 3.78% while memory usage is about 2.68%, as listed in Table 2. This table is derived from the report of system resource usage after the logic circuit has been compiled. Therefore, the idle resources in FPGA have to be used properly in order to reduce the usage of DSP and achieve the goal of being a multi-mode, multi-band, high-precision navigation receiver or even other scientific objectives.

In response to the flourishing global navigation satellite systems, NSPO has planned to gradually extend this SGR to become a multi-mode, multi-band, and high-precision navigation receiver and even be suitable for scientific applications. Starting from 2013, NSPO initiated the preliminary study of extension feasibility in order to achieve this goal. The main purpose is to port the highly repetitive and time-consuming functions from DSP to FPGA in the early research phase of this

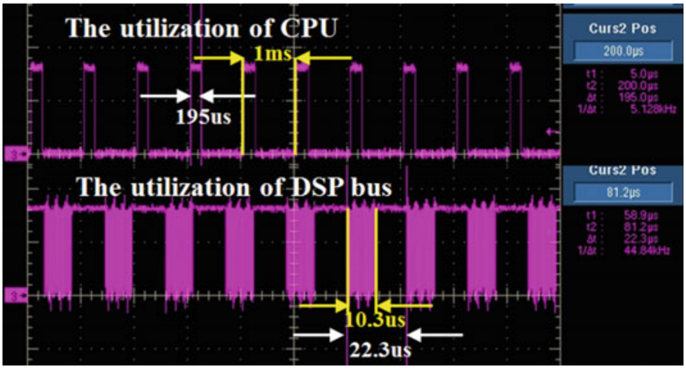


Fig. 4 The system resource usage of SGR

Table 2 FPGA compile report of SGR

CORE	Used:	2844	Total:	75264	3.78%
IO_(W/_clocks)	Used	78	Total:	147	53.06%
Differential_IO	Used:	0	Total:	65	0.00%
GLOBAL_(Chip+Quadrant)	Used:	5	Total:	18	27.78%
PLL	Used:	0	Total:	2	00.00%
RAM/FIFO	Used:	3	Total:	112	2.68%
Low_Static_ICC	Used:	0	Total:	1	0.00%
FlashROM	Used:	0	Total:	1	0.00%
User_JTAG	Used:	0	Total:	1	0.00%

extension work, such as the functions of signal acquisition and signal correlation. The processor is primarily responsible for navigation solution, orbit propagation and other tasks with non-real-time requirements. This paper will present the work and achievements of all the pre-extension research in detail after introducing the mission application and environmental certification of this SGR in Sect. 1.2.

1.2 Space Mission and Certification Test

FORMOSAT-7/COSMIC-2 Program is a major collaboration space program between Taiwan and the U.S. In this program, the designated representative of Taiwan is NSPO and its U.S. counterpart is NOAA (National Oceanic and Atmospheric Administration). This program includes 12 radio occultation (RO) science mission satellites plus one augmented NSPO-Built satellite. The 12 satellites are planned to be launched and deployed in two separate clusters of 6 satellites into the designated low and high inclination angle orbits in 2017 and 2019, respectively. The basic information about FORMOSAT-7/COSMIC-2 program is summarized in Table 3.

Table 3 The specification of FORMOSAT-7/COSMIC-2

Specifications	Magnitude
Number of satellite:	13 satellites as a goal
Mass:	<300 kg /Satellite (wet)
Mission orbit:	1st set: low-inclination angle (24 deg), 520 ~550 km circular orbit
	2nd set: high-inclination angle (72 deg), 720 ~750 km circular orbit
Constellation:	One satellite per orbital plane, 6 orbital planes per cluster of satellites
Mission life:	5 years

The NSPO-Built satellite scheduled to be launched in the 2nd cluster will further enhance the radio occultation (RO) data sampling density of the COSMIC-2 constellation and also serve as a qualification platform for NSPO's indigenous key components development. For the NSPO-Built satellite, four in-house components were considered including Fiber Optical Gyro, GPS Receiver, On-Board Computer, and Power Control and Distribution Unit.

The harsh environment, such as total ionizing dosage, single event effect, extreme temperature and launch vibration always cast certain threaten to GPS receivers intended for space applications. Usually the thermal cycling, thermal vacuum, vibration, EMI/EMC, and ionizing radiation are all required testing items for GPSR's environmental stress screening process to meet corresponding mission requirements. Around Nov. 2012, the engineering model of SGR passes part of the entire qualification test campaign, such as the thermal cycle test, the vibration test, and the radiation test. NSPO also completed all EQM qualification tests according to the preset schedule in 2015. Figure 5 shows the on-site snapshots of all environmental

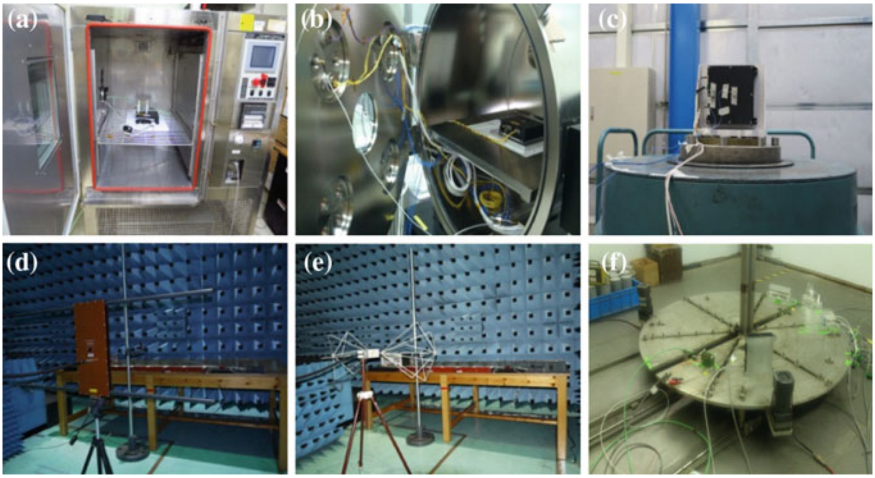


Fig. 5 On-site photos of all environmental tests

tests, including thermal cycling (a), thermal vacuum (b), vibration (c), EMI (d), EMC (e), and radiation (f) tests.

The temperature trending data shown in Fig. 6 were obtained from the thermal cycling test completed in April 2015. The SGR functioned normally while the operation temperature cycled between −40 and +80 °C in one-week testing period.

The temperature trending data shown in Fig. 7 were obtained from the thermal vacuum cycling test completed in May 2015. The SGR also functioned normally while the operation temperature cycled between −40 and +80 °C in one-week testing period.

The vibration test, the EMI/EMC test, and the radiation test, were all completed one by one before the end of 2015. This SGR passed the vibration test in which a

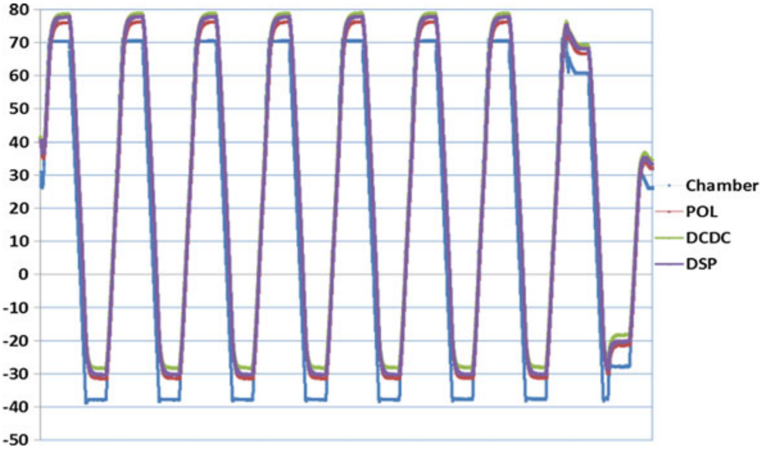


Fig. 6 Thermal cycling test of SGR

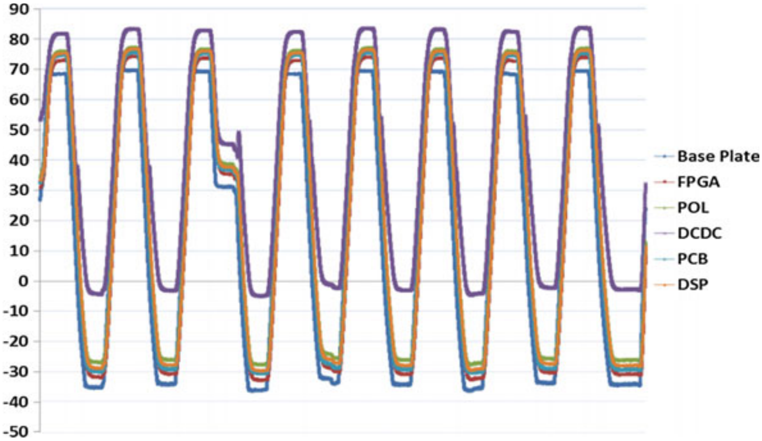


Fig. 7 Thermal vacuum cycling test of SGR

testing value of 20 g applied and remained healthy throughout the test. It then passed the EMI/EMC test when a frequency interference between 1G and 2G Hz applied during the test, and never exceeded the threshold set by the MIL-STD-461F RE102 while the testing frequency ranged from 1 G to 18G Hz. It also had normal power consumption before having an exposure dose of 34 Krads.

2 Extendable Space-Borne GNSS Receiver

In order to serve as a multi-mode, multi-band, and high-precision navigation receiver, the prototype of purposed ESGNSSR has been completely designed in 2015. The primary difference between current and previous functions of FPGA is that some extra logics implementing L1-GPS receiver algorithms are added. The new logic block diagram of FPGA is shown in Fig. 8.

The block diagram of extended logic circuits is shown in Fig. 9 in which the following three logics, Sample Unpack module, FSA module, and Correlation module, were ported from DSP to FPGA.

The DMA/Data Switch and Router module is mainly responsible for transferring the RF raw data to the Sample Unpack module and storing the data generated by the FSA and Correlation modules in FIFO buffer for later use. The System Control module is responsible for performing the judgment and management of all tasks in extended logic and it also needs to handle data communication and handshaking between the FSA and the Correlation modules in order to ensure data concurrency.

Trying to utilize remaining resources available in FPGA to extend new functions in DSP for the SGR is our goal. The preliminary study of functional extension feasibility mainly focused on using the FPGA to implement the highly repetitive and time-

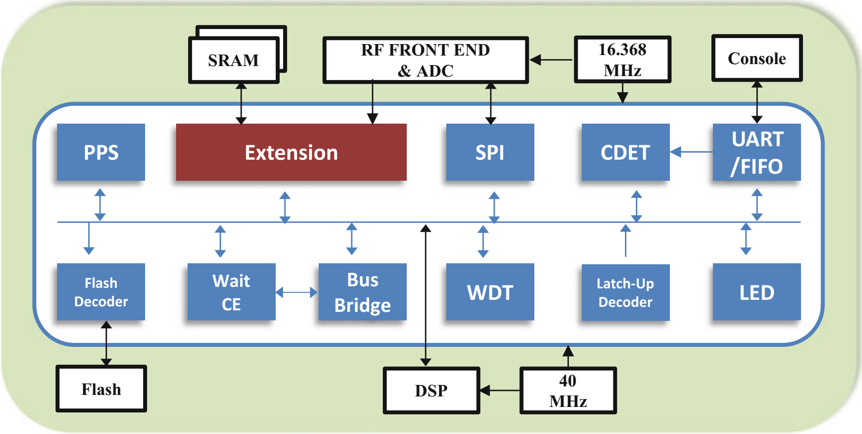


Fig. 8 New logic block diagram of FPGA

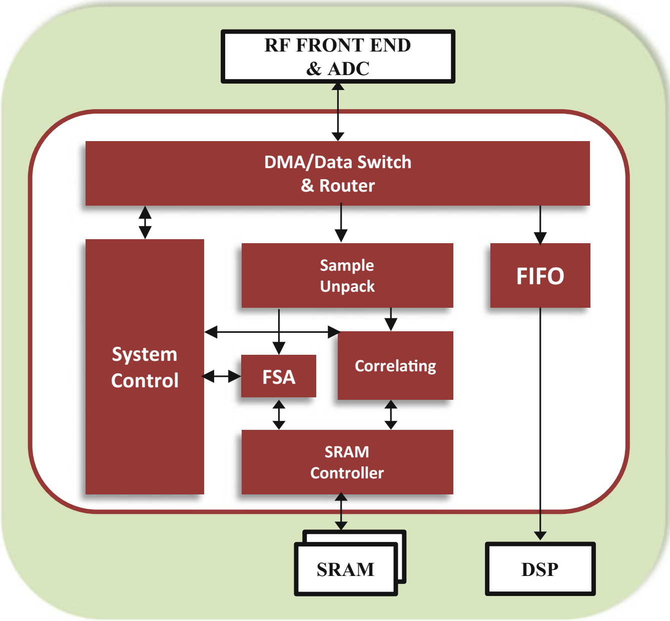


Fig. 9 The block diagram of extended logic

consuming functions, such as Sample Unpack, FSA and Correlation in order to reduce the workload of the DSP. However, the signal phase-locked loop, Decoding/Forming, and navigation solution, etc., are still handled by the DSP. Among all GPS algorithms, these three functions occupy most memory of this SGR. Note that FPGA has only a small memory size of 512K Bits compared to 1M Bytes memory built in the DSP. The limitation of available memory could be the bottleneck if these three algorithms are all implemented in the FPGA. Therefore, two pieces of external SRAM's (1M Words each) were added to the FPGA in order to overcome this problem. One chunk of SRAM is dedicated to storing the signal searching results from the FSA, and the other is set to store results of signal correlation from the Correlator.

2.1 Logic Architecture

The main purpose of the Sample Unpack module is to rearrange the signal data generated from the RF front-end, including data decimation and moving average filtering. In Eq. (1), the incoming signal models after data sampling are concisely described.

$$\begin{aligned} y\left(t_n\right) &= \left[R_e\left(t_n\right) + jI_m\left(t_n\right)\right] \cdot e^{j2\pi f_s t_n} \\ &= I_y\left(t_n\right) + Q_y\left(t_n\right) \end{aligned} \tag{1}$$

where f_s represents the sampling frequency. The logic processing steps of Sample Unpack module are introduced as follows. The real and imaginary parts of the raw signals are separated first. Next, four adjacent data points are selected for arithmetical computation. After data combination, the raw data collected at a sampling rate of 16.368 MHz and two bits in each I and Q parts will form a new data structure which has a size of four bits and is sampled at a rate of 4.092 MHz. The architecture of Sample Unpack shown in Fig. 10 in which the plus or minus sign can be appropriately known from the expansion result of Eq. (1).

The FSA module primarily handles signal search in ESGNSSR. Its acquisition algorithm is described as follows. The first step is to perform FFT with incoming signals and can be expressed in Eq. (2).

$$Y_i\left(k\right) = \sum_{n=0}^{N-1} \left\{I_y\left(t_n\right) + Q_y\left(t_n\right)\right\} \cdot e^{-j2\pi kn/N} \tag{2}$$

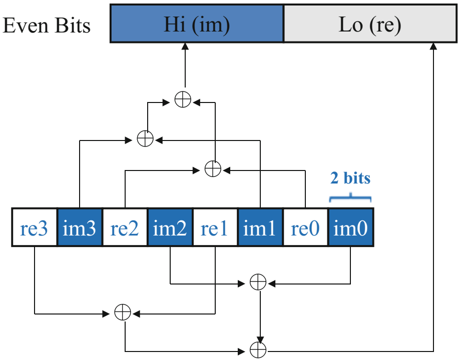
where Y represents incoming signals, k is in range of 0–$N-1$, $N$is number of samples per millisecond, and i is in range of 0 to 7. Next, an FFT is performed with its replica as described in Eq. (3).

$$L\left(k\right) = \sum_{n=0}^{N-1} C_{PRN}\left(t_n\right) \cdot e^{-j2\pi kn/N} \tag{3}$$

where L is the FFT result of replica. Third, an inverse FFT is performed with the multiplication result of $Y_i\left(k\right)$ and $L\left(k\right)$, and it can be formulated as in Eq. (4).

$$M_i\left(k\right) = \sum_{n=0}^{N-1} \left\{Y_i\left(k\right) \cdot L\left(k\right)\right\} \cdot e^{j2\pi kn/N} \tag{4}$$

Fig. 10 The architecture of sample unpack



where M is the result of inverse FFT. Then it sequentially performs 8-point FFT by using the results from inverse FFT, and output the desired information, such as Doppler shift and code shift. The Eq. (5) summarizes this process.

$$E\ (k) = \sum_{n=0}^{N'-1} M_i\ (k) \cdot e^{-j2\pi kn/N'} \tag{5}$$

where $N^{'}$ is 8 and k is in range of 0 to $N' - 1$. The FSA is the most complex RTL module in this study. The implemented architecture of the FSA module is illustrated in Fig. 11 where the 4096-point FFT and the 4096-point IFFT share the same FFT IP, and a time-division multiplexing mechanism is used. Beware that the replicated L1-GPS PRN code is stored in the 1023-PRN-code ROM.

At first, the consecutive data in first eight milliseconds are stored in the external SRAM. The FSA module will let the replica be the first to enter the FFT IP and store the results generated from FFT IP in PRN code FFT buffer. Then the module will retrieve one millisecond data from the external SRAM for the FFT IP to do the FFT operation and provide the results generated from the FFT IP for multiplier to perform the multiplication with the data stored in PRN code FFT buffer. Moreover, the data in PRN code FFT buffer used for multiplication should be applied a frequency bin shift between +130 to −130 in advance. A complex multiplier is then used to perform data multiplication. After multiplication operation finished, the result will enter the FFT IP to perform the IFFT. The final results generated from the IFFT will constitute a correlation buffer and be stored within the external SRAM. Such a process needs to be executed eight times iteratively.

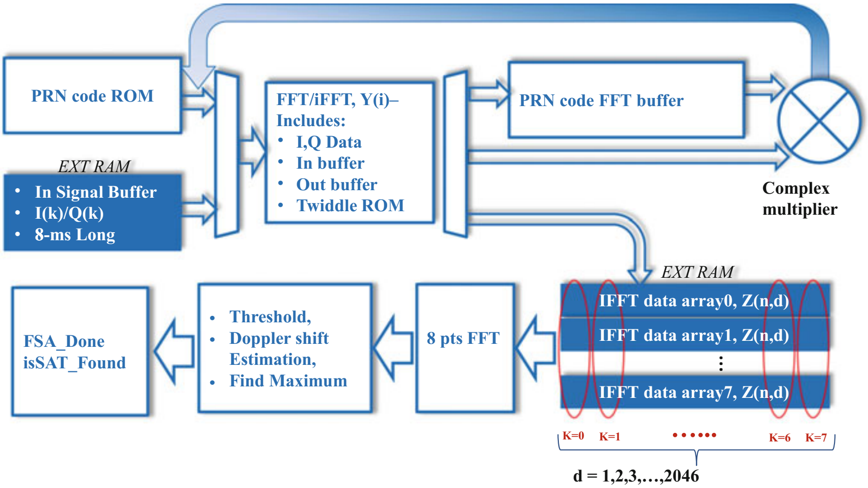


Fig. 11 The architecture of FSA

After completion of IFFT computation by taking 8 ms consecutive data as input, the results are stored temporarily in the correlation buffer. Next, the FFT8, an 8-point FFT IP, will longitudinally retrieve data from this correlation buffer one by one to perform the FFT8. After that, a logic comparator will work on previous results to retain the largest parameters such as sample position, frequency bin, and relevant time stamp in millisecond for later use. Simultaneously, in light of the analysis result, we will select an appropriate threshold which is used to determine whether a L1-GPS signal is found based on the comparison between the largest parameter and this threshold.

The Correlation module is mainly responsible for correlating incoming signals with replica according to parts of the tracking results generated by DSP. Here we can describe the correlation equations in Eqs. (6)–(8).

$$G_P = \sum_{k=0}^{N-1} y\left(t_k\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{6}$$

$$G_E = \sum_{k=0}^{N-1} y\left(t_k + \Delta t\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{7}$$

$$G_L = \sum_{k=0}^{N-1} y\left(t_k - \Delta t\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{8}$$

where f_D is Doppler frequency; Δt is samples, G_P, G_E and G_L are the correlation results in prompt, early, and late phase respectively.

The implemented architecture of the Correlation module is shown in Fig. 12. The Correlation module continually receives data from the Sample Unpack module at a data rate of 8184 samples per mini-second. Moreover, the data are composed of even and odd parts, and each part has 4096 samples. It always keeps 3 ms consecutive data, 3 ms_Buffer, in the external SRAM for the data correlation operation. In every millisecond, the DSP will provide pre-calculated parameters from 12 channels to the Correlation module in advance, including Doppler, code phase, sample position, satellite PRN number, initial phase and phase increment. The Correlation module will finish all calculations in Early, Prompt, and Late phase of 12 channels within one millisecond in order to allow the DSP to be able to initialize the signal phase-locked loop before start of the next one millisecond.

The detailed operations of 3 ms_Buffer in the Correlation module can be categorized into three conditions and are illustrated in Fig. 13. Taking the first condition for example, the input data fill the blue block of 4092 bytes at a rate of 4.092 MHz. The related logic circuits simultaneously retrieve data from the white region of 8184 bytes based on the sample position computed by the FSA while the DSP actuates the Correlation module per millisecond. In this way, the 3 ms_Buffer operating in this way can ensure that the data are always continuous and not overlapped.

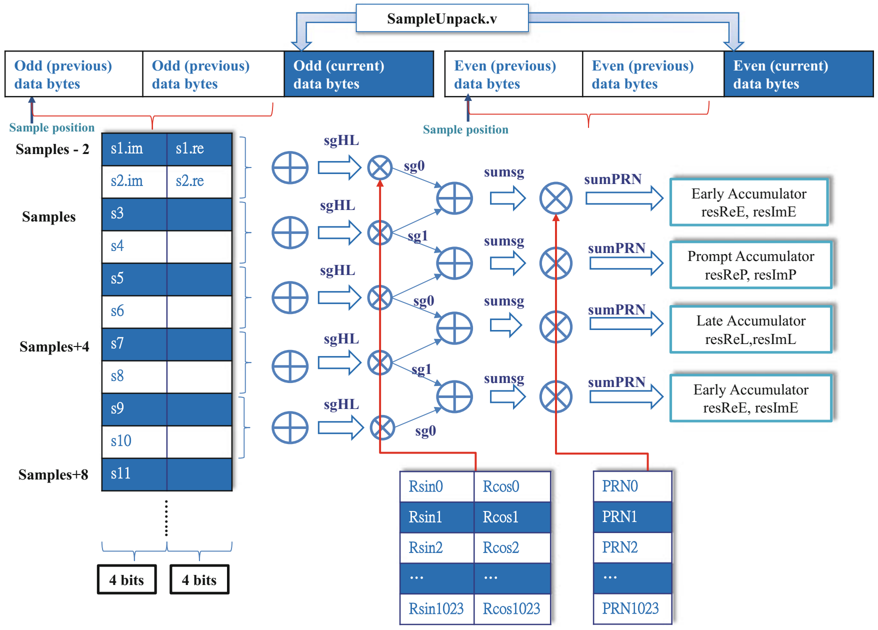

SampleUnpack.v
Odd (previous) data bytes
Odd (previous) data bytes
Odd (current) data bytes
Even (previous) data bytes
Even (previous) data bytes
Even (current) data bytes
Sample position
Sample position
Samples - 2
Samples
Samples+4
Samples+8
s1.im
s1.re
s2.im
s2.re
s3
s4
s5
s6
s7
s8
s9
s10
s11
4 bits
4 bits
sgHL
sg0
sg1
sumsg
sumPRN
Early Accumulator resReE, resImE
Prompt Accumulator resReP, resImP
Late Accumulator resReL,resImL
Early Accumulator resReE, resImE
Rsin0
Rcos0
Rsin1
Rcos1
Rsin2
Rcos2
…
Rsin1023
Rcos1023
PRN0
PRN1
PRN2
…
PRN1023

Fig. 12 The architecture of correlation

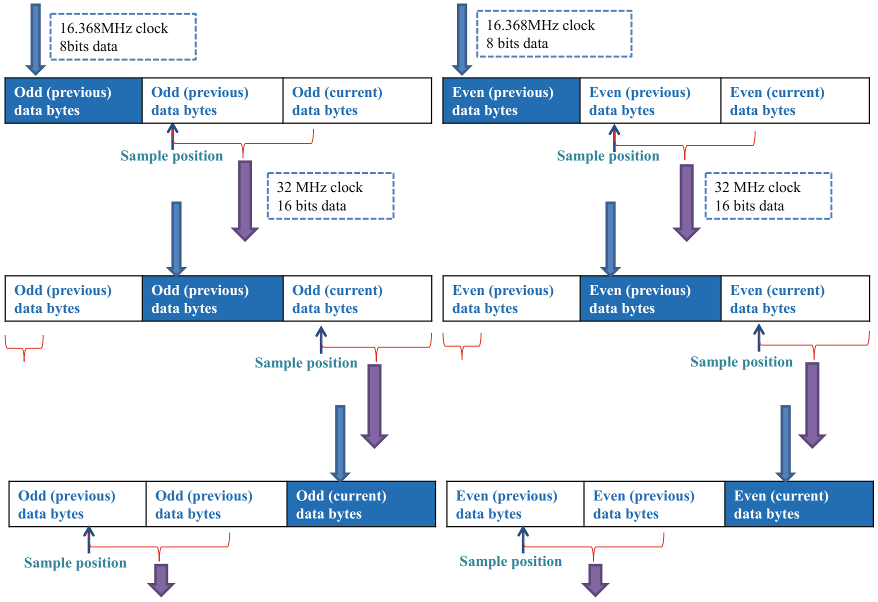

16.368MHz clock 8bits data
16.368MHz clock 8 bits data
Odd (previous) data bytes
Odd (previous) data bytes
Odd (current) data bytes
Even (previous) data bytes
Even (previous) data bytes
Even (current) data bytes
Sample position
32 MHz clock 16 bits data
Sample position
32 MHz clock 16 bits data

Fig. 13 The architecture of 3 ms_Buffer

It is noteworthy that improving utilization of the DSP bus lays the groundwork for hardware interface extension, and allows simultaneous processing of more different types of GPS data in the multi-mode and multi-band GNSS receiver. Besides, more effective utilization of CPU resources not only reduces calculation time of the Navigation function in DSP, but also reserves more available computing resources for future application. The feasibility of functional extension of NSPO's space qualified L1-GPS receiver is verified through a close loop test presented in this paper.

By using more powerful FPGA's, the speed of GPS signal acquisition and the channel number of GPS signal correlation can be enhanced and increased greatly in future development. Therefore, the new enhanced configuration can be compatible with varieties of existing GPS satellites, such as Russian's GLONASS, Europe's GALILEO and China's Compass (Beidou). The goal to develop a multi-mode and multi-band GNSS receiver should be achieved in the near future eventually.

2.2 *Logic Analysis*

All the logic functions presented in this paper use the MODELSIM tool for simulation and verification. The architecture of test bench is illustrated in Fig. 14. First, the recorded GPS signals will be stored in the ROM TABLE, such as GPS_DATA_ROM.v, and then these signals will be imported to the FPGA modules, including the SampleUnpack.v, the FSA.v and the Correlator.v, for test. Meanwhile, both the external RAM simulation model which is RAM_MODEL.v and the Ti-DSP EMIF simulation model which is BUS_MODEL.v are also created for the

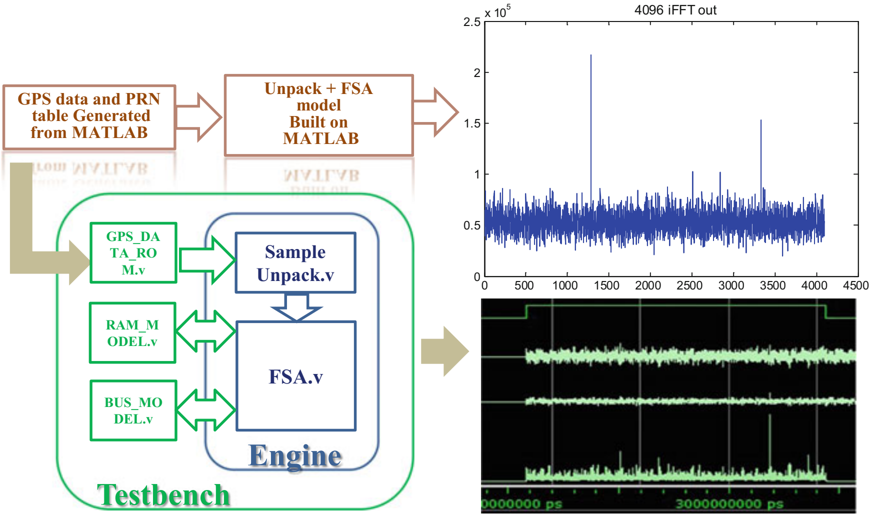


Fig. 14 RTL design simulation/verification flow

test bench in order to simulate all the logic functions as close as to the real condition as possible. The information, such as those GPS signals identified by the FSA module or the computation results of the Correlation module, can be obtained from post-simulation data analysis. Further double check with the algorithms allows us to know if the simulation results are correct as well.

3 The Remote Configuration of a Real-Time Platform

The approach of evaluating the LEO navigation is to build a test bench where resultant motion data due to environment and corresponding control activities can be fed into the RF generator in real time. A simulation scenario of FORMOSAT-7/COSMIC-2 is used to propagate the designated RF signals via a RF signal generator, the Spirent GSS7700, in this study. The proposed test bench consists of two key elements, ACDESS and the GSS7700, controlled by the ACDESS. In the following sections, the in-house ACDESS based on the National Instruments (NI) PXI products will be thoroughly introduced including its hardware interface with the GSS7700.

The original test bench is composed of a remote PC running Windows operation system and a hardware interface capable of maintaining clock synchronization between itself and the GSS7700. Note that, in order to integrate various hardware but still achieve high reliability, the original remote PC of GSS7700 is replaced by NI-PXI products, including the development software CVI, the controller card PXI-8106, the LabView-RT operation system, an M-series interface card PXI-6251, a serial port interface card PXI-8433/4 and so on. The PXI remote platform can provide remote motion data of simulation scenario to the SimGEN PC through TCP/IP network protocol at a rate of 64 Hz. Figure 15 shows the updated configuration of the proposed test bench, which is part of the ACDESS. In order to synchronize devices, the GSS7700 is commanded to generate pulse per second (PPS) signals which are then sent to the PXI platform for clock synchronization. A PXI-6521 in PXI remote platform is used to measure the synchronous pulse signals from the GSS7700 but the PXI does not send any timing compensation signals back to the GSS7700.

As shown in Fig. 15, the PXI products are separated into two dedicated standalone PXI subsets. One serves as the flight software controller and takes care of mode transition, guidance, navigation and attitude control, etc. The other plays the role of hardware modeling and provides information such as flight dynamics, space environment, and ephemeris, etc. The architecture of PXI subsets is illustrated in Fig. 16. Meanwhile, some of the acronyms in Fig. 16 are summarized in Table 4. All of the test results, the performance of ESGNSSR and verification performed on NI-PXI products, are going to be presented in experimental results.

The ESGNSSR is integrated into the ACDESS platform proposed in this paper to form a closed loop test system. The communication process of this system is roughly like this. First, the Dynamics / Environments Software Simulator running on a PXIe-8133 sends the satellite information generated from the numerical models to the GSS7700's controller, SimGEN, through an Ethernet cable. The GSS7700

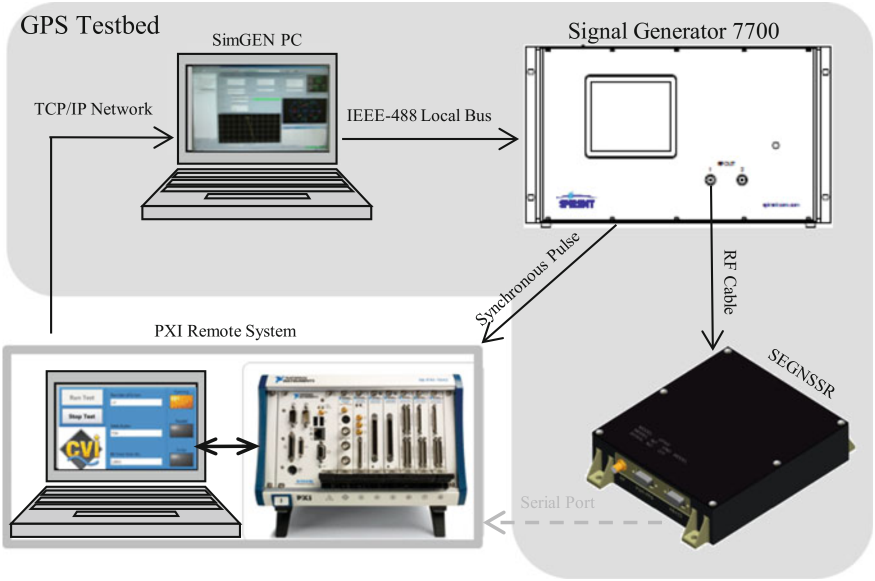


Fig. 15 Configuration of test bench

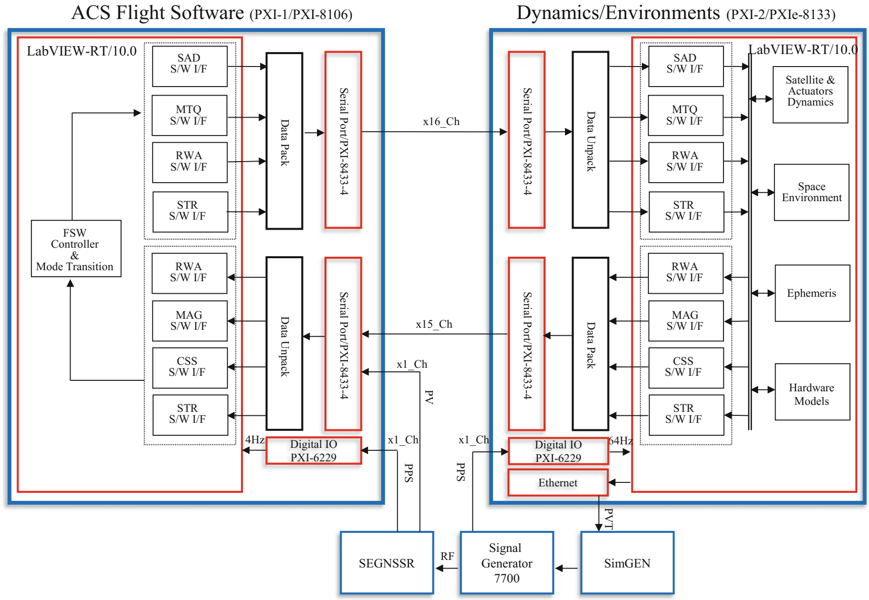


Fig. 16 The Architecture of ACDESS

Table 4 The Glossary of the acronyms used in Fig. 16

FSW	Flight software
SAD	Solar array driver
MTQ	Magnetic torquer
RWA	Reaction wheel assembly
MAG	Magnetometer
CSS	Coarse sun sensor
STR	Star tracker

broadcasts compatible RF signals based on the information received by SimGEN. The GSS7700 is then connected to the ESGNSSR via an RF cable of approximately 1 m long. The ESGNSSR will instantly resolve the PVT information according to the RF signal received at that time and then send the data to the PXI-8106, which is responsible for executing the Attitude Control Software Simulator via a serial port interface. Up to this moment, partial evaluation work of hardware-in-the-loop (HIL) has completed on such a satellite performance verification platform. But it is still necessary to prepare complete information of the remaining components, as listed in Table 4, to allow the whole system operating properly. It is not the purpose of this paper to integrate all components into this closed loop system. Therefore, the rest components will be substituted by corresponding numerical models, and complete their communication tasks through the serial port interface.

4 Experimental Results

4.1 New System Resource

After its function extended, the FPGA regularly provides the calculation results from FSA and Correlation to DSP per millisecond and the Tracking module in DSP will subsequently continue to finish all remaining algorithm steps. Next, the Navigation module is executed after the Tracking module and allowed to use only the remaining time in each millisecond. The real-time scheduling of tasks is depicted in Fig. 17.

Based on the usage of system resources, one can investigate the system performance after the functional extension work. The DSP bus occupies only 0.9% and the real-time tasks utilize less than 6% of the CPU time as shown in Fig. 18.

Significant reduction of the bus work-load can be attributed to no more large amount of RF raw data processed by DSP directly. Only some critical information instead of raw data is processed by the DSP. Similarly, the huge amount of calculation in the Correlator is also passed on to the FPGA. The number of CPU requests from real-time tasks finally decrease significantly in the DSP.

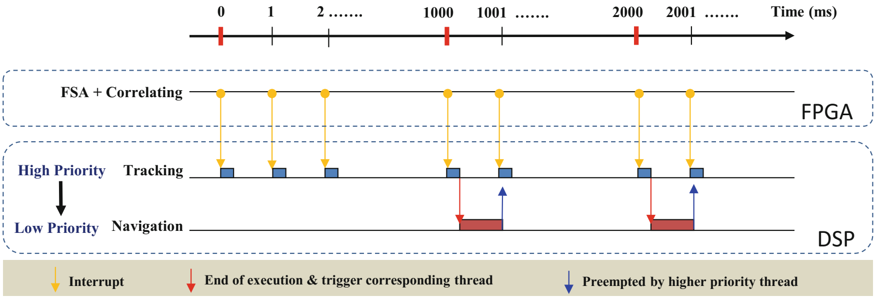


Fig. 17 The real-time scheduling of new tasks

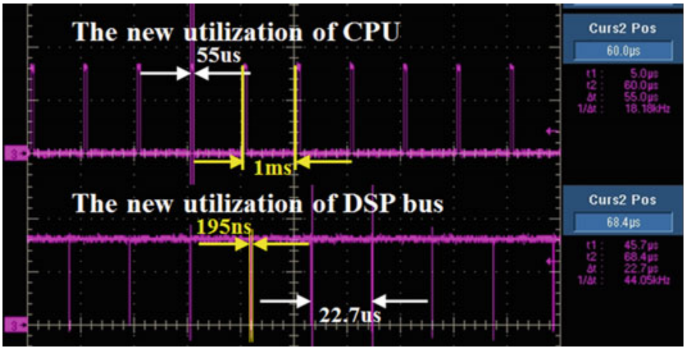


Fig. 18 The system resource usage of ESGNSSR

Table 5 FPGA compile report of ESGNSSR

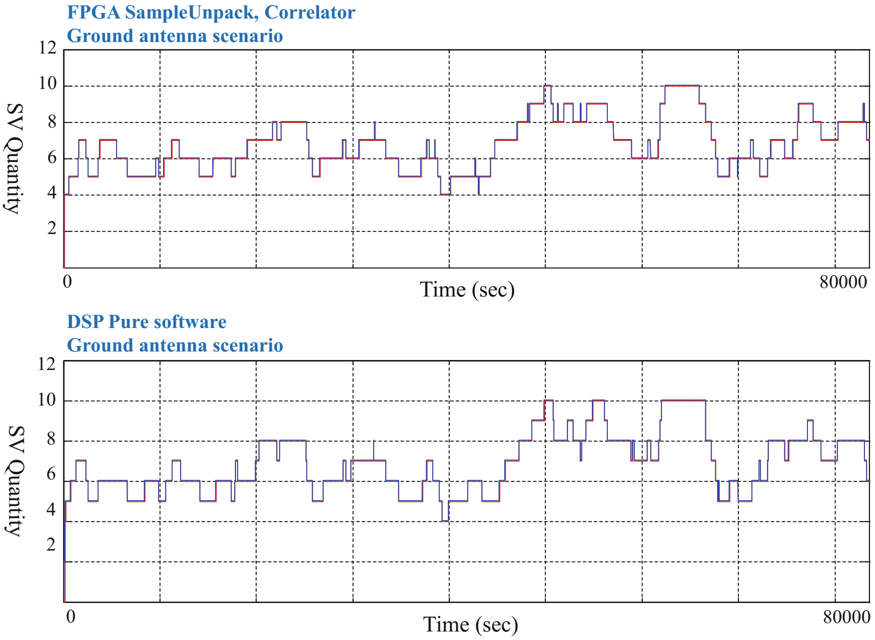
CORE	Used:	70191	Total:	75264	93.26%
IO_(W/_clocks)	Used	146	Total:	147	99.32%
Differential_IO	Used:	0	Total:	65	0.00%
GLOBAL_(Chip+Quadrant)	Used:	6	Total:	18	33.33%
PLL	Used:	1	Total:	2	50.00%
RAM/FIFO	Used:	109	Total:	112	97.32%
Low_Static_ICC	Used:	0	Total:	1	0.00%
FlashROM	Used:	0	Total:	1	0.00%
User_JTAG	Used:	0	Total:	1	0.00%

In the new FPGA logic design, the usage of logic gates and memory currently reaches about 93.26 and 97.32% separately. The resources of the FPGA are almost used up as listed in Table 5 which is derived from the report of system resource usage after the logic circuit compiled.

4.2 Performance Comparison with SGR

By using a GPS-L1 roof antenna mounted on the NSPO I and T facility with a repeater and a divider, a performance test was conducted to compare the GPS signal tracking capability of both versions of receivers, the original GPS software receiver and the one with extended functions. The test result is shown in Fig. 19 in which the blue line represents the number of satellites tracked and red line means the number of satellites used for navigation solution. Both receivers make no big difference in performance when they track GPS signals on the ground.

Moreover, a RF signal generator (GSS7700) manufactured by SPIRENT and a divider were used in another test in which the space scenario was set to be in Low Earth Orbit (LEO) of 600 km altitude for comparing GPS signal tracking capability of both versions of receivers. The test result is shown in Fig. 20 in which the blue line represents the number of tracked GPS satellites and the red line represents the number of GPS satellites used for navigation solution. Both receivers have similar performance in space scenario test case as well. Many other performance verification tests can be found in published papers and there is no need to elaborate any further here.


FPGA SampleUnpack, Correlator
Ground antenna scenario

DSP Pure software
Ground antenna scenario

Fig. 19 Tracking performance in ground scenario

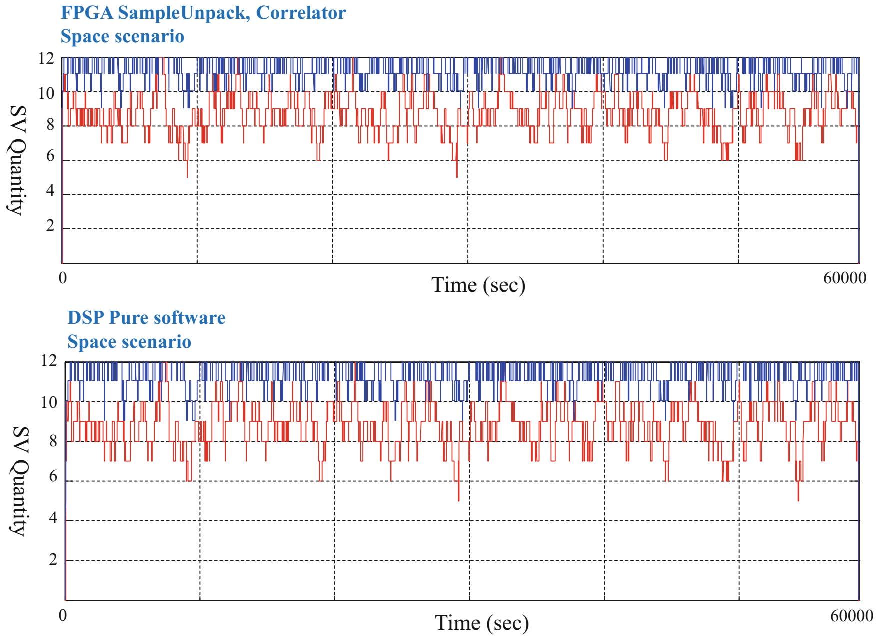


Fig. 20 Tracking performance in space scenario

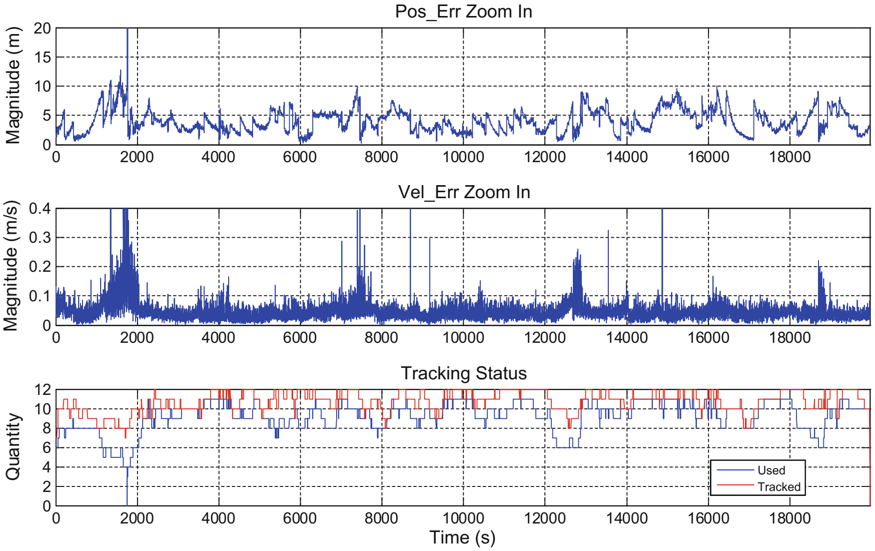


Fig. 21 The performance validation of ESGNSSR

4.3 Performance Validation in ACDESS

As shown in Fig. 21, this navigation error is calculated by the ESGNSSR when the satellite is operating in the Normal Mode scenario. Note that a significant navigation error occurred between 1800 and 1900s due to a large-angle maneuvering of the satellite. However, there are not significant navigation errors bursting during the remaining satellite maneuvers.

5 Conclusions

In this study, the Decimation-In-Time (DIT) Radix-2 is used for FFT. However, it is impossible to perform continuous computation (streaming mode) because its input and output have only one 128Kbits buffer individually. This constraint makes every new FFT process must wait until current FFT process is complete. Hence, the FFT process has a major impact on the real-time requirement in whole system. In addition to implementing new functions for logic circuits, much time and effort were spent on solving time-mismatch issues occurred during system integration involving the FPGA and the DSP. It is worth reminding that appropriate allocation of CPU resources is also an important determining factor in ensuring all real-time tasks completed as scheduled.

Moreover, improving utilization of the DSP bus has paved the way for successful interface extension allowed to accommodate more different types of GPS data in the multi-mode and multi-band GPS receiver. Saving more available CPU resources for future use could be deemed as a remarkable extra achievement of this study. The feasibility of functional extension of NSPO's space qualified SGR has been successfully verified through all tests presented in this paper.

The real-time ACDESS platform proposed in this paper is a functional evaluation system which completely emulates the engineering models of the FORMOSAT-7 NSPO-Built satellite. It includes the complete satellite attitude control algorithms, flight software operation time slots, the integration of hardware interface, and the implementation of dynamics/environment models. To sum up, the successful verification of functionality and performance of this receiver on the designated platform makes NSPO more confident in planning more upcoming satellite navigation missions.

References

1. Kuehl CTF et al (2008) GNSS goes LEON – a LEON-2 based GNSS receiver for space applications. In: 7th internationalesa conference on guidance, navigation and control system
2. Markgraf M et al (2002) A flexible GPS tracking system for sub-orbital and space vehicles. In: 9th Saint Petersburg international conference on integrated navigation systems

3. NSPO, Taiwan (2016) Self-reliant key component of FORMOSAT-7. http://www.nspo.org.tw/2008e/projects/project7/components.html
4. SURREY, UK (2016). SGR-20 space GPS receiver. https://www.sstl.co.uk/Products/Subsystems/Navigation/SGR-20-Space-GPS-Receiver
5. Airbus Defence and Space, France (2014) MosaicGNSS receiver. www.space-airbusds.com/media/document/pdh_8_gps-2014_bd.pdf
6. Gerner JL et al (2000) TOPSTAR 3000 – an enhanced GPS receiver for space applications. In: Final presentation of topstar 3000 and the experimential GPS attitude receiver
7. NSPO, Taiwan (2016) Program description of FORMOSAT-7. http://www.nspo.org.tw/2008e/projects/project7/program-description.html
8. Chang, H-Y et al. (2013) Performance demonstration of NSPO space-borne GPS receiver. In: Proceedings of ION

Trajectory Shaping Guidance Law Based on Downrange-to-Go Polynomial

Namhoon Cho, Youdan Kim, Hyo-Sang Shin and Antonios Tsourdos

1 Introduction

Ground or naval targets are usually equipped with the defensive measures and anti-missile systems such as armoured shield for self-protection, electronic jammer, close-in weapons system, etc. However, those measures cannot provide isotropic and uniform defensive power over all directions due to limited coverage. In other words, a certain direction around the target is more vulnerable to incoming attack than others. From the view of offensive missile, it will be advantageous to have a capability of hitting the target from a desired impact direction.

On the other hand, it is desirable in most cases to regulate the lateral acceleration to zero as the missile approaches to the target. This is to maximize the hit probability and the destructive power of warhead, to reduce the angle-of-attack at the moment of impact, and to allow small correction near the end of engagement. Most importantly, zero terminal acceleration constraint is necessary to avoid command saturation at the end of homing phase, because the control authority and the margin for manoeuvre in response to external disturbances can be maintained by ensuring this constraint.

To meet the above requirements, several guidance laws have been developed to cope with terminal impact angle constraint while reducing terminal manoeuvre

N. Cho · Y. Kim (✉)
Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea
e-mail: ydkim@snu.ac.kr

N. Cho
e-mail: nhcho91@snu.ac.kr

H.-S. Shin · A. Tsourdos
School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, Bedfordshire, United Kingdom
e-mail: h.shin@cranfield.ac.uk

A. Tsourdos
e-mail: a.tsourdos@cranfield.ac.uk

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_30

demand at the same time [1–10]. Time-to-go Polynomial Guidance law (TPG) presented in [2, 4] is of concern in this study. TPG is a class of guidance laws with terminal impact angle and acceleration constraints, which is a general framework of trajectory shaping guidance law design. To design the TPG, the form of acceleration command is given by a polynomial of time-to-go, and the coefficients of the polynomial are determined by terminal boundary conditions. TPG is known as a guidance scheme that provides several promising results in theoretical analysis, flexibility of tuning, and good performance. Moreover, TPG is a general form of trajectory shaping guidance laws for terminal impact angle control, and weighted linear quadratic optimal guidance laws with terminal impact angle constraint can be regarded as a specific type of TPG.

In this study, a modified approach is proposed to improve several shortcomings of the TPG. In [2, 4], TPG was designed by designating acceleration command as a polynomial of *time-to-go*. Note that TPG has been developed on the basis of linearized engagement kinematics with constant speed assumption. Also, the guidance command is given by an explicit function of time-to-go for which an approximate estimate is only available rather than the exact one. Unlike the TPG, in this study, the desired crossrange trajectory is constructed using a polynomial of *downrange-to-go*, which is the first step of a new trajectory shaping guidance law design. By doing this, the proposed guidance law does not depend on (1) linearization of engagement kinematics, (2) assumption of constant speed missile, and (3) inaccurate time-to-go estimate. In summary, the proposed approach overcomes the shortcomings of TPG while sharing similarities in its design philosophy.

The key idea of the proposed approach is to assign a desired *crossrange* pattern as a function of *downrange-to-go*, instead of other options for the independent variable such as time-to-go, range-to-go, or path-length-to-go. If time-to-go is chosen as the independent variable for describing the desired trajectory, then it is hard to deal with time-varying speed case and the final command will depend on inaccurate time-to-go estimate. If path-length-to-go is chosen, then the constant speed assumption can be relaxed. However, the design in this case is identical to the previous one using the time-to-go except the change-of-variable in engagement kinematics, and it is difficult to represent the path-length-to-go as an exact form. Or, if range-to-go is used to describe the desired trajectory, then it will be difficult to consider the terminal impact angle constraint. To treat the shortcomings of the previous methods, downrange-to-go is used to describe the desired trajectory in this study. Also, the proposed approach does not require linearization of engagement kinematics. Note that this approach differs from performing feedback linearization of engagement kinematics first and then designing the virtual control as a polynomial of downrange-to-go.

The rest of this paper is organized as follows: The problem of stationary target interception with terminal constraints is formulated in Sect. 2. The trajectory shaping guidance law is proposed and its properties are discussed in Sect. 3. Numerical simulation is performed to demonstrate the effectiveness of the proposed guidance law and the results are shown in Sect. 4. Concluding remarks are summarized in Sect. 5.

2 Problem Formulation

In this section, assumptions to design a guidance law, equations of planar engagement kinematics, and problem statement are described.

2.1 Assumptions

The following assumptions are considered to design and analyse the guidance law proposed in this study.

Assumption 1 The target is stationary, and the velocity and acceleration vectors of the missile lies on a plane for all time, i.e., the engagement is two-dimensional.

Assumption 2 The missile is a lag-free point-mass such that the actual lateral acceleration equals to the commanded lateral acceleration without time-delay and distortion.

Assumption 3 The information on the position and velocity of the missile and the position of the target is available from sensors without time-delay and noise.

Assumption 4 The (initial) velocity of the missile is within ±90 deg from the desired terminal impact direction.

Note that Assumption 2 does not influence on the tangential acceleration which is related to the change of speed. The lateral (normal) acceleration is only related to the change of flying direction and therefore to the geometric shape of the curve flown by missile.

2.2 Planar Engagement Kinematics

Figure 1 shows the planar homing engagement geometry considered in this study. In Fig. 1, (X_I, Y_I) and $\left(X_f, Y_f\right)$ denote the inertial coordinate system and the impact coordinated system, respectively. The impact coordinate system has its origin on the stationary target T, X-axis is aligned to the desired terminal impact direction, and γ_{f_d} is the flight path angle for the desired terminal impact direction. For the missile M, (x, y) is the position with respect to the impact coordinate system, V_M is the speed, γ_M is the flight path angle, and a_M^n is the lateral acceleration. The downrange-to-go denoted by x_{go} is the remaining distance to the target along X_f-axis, and $x_{go} = -x$ by construction. The flight path angle error is defined as follows

$$\bar{\gamma}_M \triangleq \gamma_M - \gamma_{f_d} \tag{1}$$

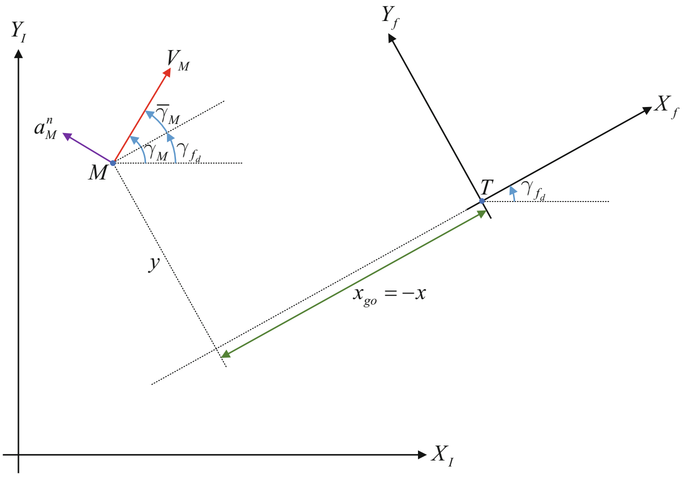


Fig. 1 Planar engagement geometry

The motion of the missile can be represented in the impact coordinate system as

$$\begin{aligned} \dot{x} &= V_M \cos \bar{\gamma}_M, & x(t_0) &= x_0 < 0 \\ \dot{y} &= V_M \sin \bar{\gamma}_M, & y(t_0) &= y_0 \\ \dot{\bar{\gamma}}_M &= \frac{a_M^n}{V_M}, & |\bar{\gamma}_M(t_0)| &< \frac{\pi}{2} \end{aligned} \tag{2}$$

where $\dot{(\,)} = \frac{d}{dt}(\,)$, and $\bar{\gamma}_M(t_0) = \bar{\gamma}_{M_0} = \gamma_{M_0} - \gamma_{f_d}$. Due to Assumption 4 which implies that $\dot{x}(t) \geq 0$ for $\forall t \in [t_0, t_f]$, the downrange x can be used as an independent variable instead of the time t, then equation of motion can be rewritten as

$$y' = \tan \bar{\gamma}_M \tag{3}$$

$$\bar{\gamma}_M' = \frac{a_M^n}{{V_M}^2 \cos \bar{\gamma}_M} = \frac{a_M^n}{{V_M}^2}\sqrt{1 + \tan^2 \bar{\gamma}_M} = \frac{a_M^n}{{V_M}^2}\sqrt{1 + y'^2} \tag{4}$$

where $(\,)' = \frac{d}{dx}(\,)$. The second derivative of y with respect to x can be derived as

$$y'' = \frac{d}{dx}\tan \bar{\gamma}_M = \bar{\gamma}_M' \frac{d}{d\bar{\gamma}_M} \tan \bar{\gamma}_M = \bar{\gamma}_M' \left(1 + \tan^2 \bar{\gamma}_M\right) = \bar{\gamma}_M' \left(1 + y'^2\right) \tag{5}$$

Equation (4) can be rewritten using Eq. (5) as follows.

$$a_M^n = \frac{y''}{\left(1 + y'^2\right)^{\frac{3}{2}}} {V_M}^2 = \kappa {V_M}^2 \tag{6}$$

Note that κ in Eq. (6) is the curvature of the curve flown by missile. Additionally, the derivative of the lateral acceleration with respect to x can be derived as follows.

$$\begin{aligned}\left(a_M^n\right)' &= \frac{\dot{a}_M^n}{V_M \cos\bar{\gamma}_M} = \frac{\dot{a}_M^n}{V_M}\sqrt{1+y'^2} \\ &= \frac{y'''\left(1+y'^2\right) - 3y'y''^2}{\left(1+y'^2\right)^{\frac{5}{2}}} {V_M}^2 + \frac{2y''}{1+y'^2}\dot{V}_M\end{aligned} \tag{7}$$

2.3 *Problem Statement*

The constraints on the miss distance, impact angle, and acceleration at the terminal time can be represented as follows.

$$y\left(t_f\right) = 0 \tag{8}$$

$$\bar{\gamma}_M\left(t_f\right) = 0 \tag{9}$$

$$a_M^n\left(t_f\right) = 0 \tag{10}$$

Let t_f be the time instance when x becomes zero, i.e., $x\left(t_f\right) = 0$. Considering Eqs. (3) and (6), the terminal constraints of Eqs. (8)–(10) are equivalent to the following conditions using the changed independent variable x.

$$y\left(0\right) = 0 \tag{11}$$

$$y'\left(0\right) = 0 \tag{12}$$

$$y''\left(0\right) = 0 \tag{13}$$

Note that, if $\dot{a}_M^n\left(t_f\right) = 0$ is considered as an additional constraint, then, in view of Eq. (7), $y'''\left(0\right) = 0$ is required in addition to Eq. (13).

The problem to be solved in this study is to design a guidance law $a_{M_{\mathrm{cmd}}}^n$ with which all of the terminal constraints given by Eqs. (8)–(10) (or equivalently, Eqs. (11)–(13)) can be achieved.

3 Guidance Law Based on Downrange-to-Go Polynomial

This section is devoted to the development of a new trajectory shaping guidance law based on desired crossrange pattern given by a polynomial of downrange-to-go.

3.1 Design of Guidance Law

3.1.1 Desired Crossrange Pattern

Let the desired crossrange pattern be the polynomial of downrange-to-go which can be written as

$$y\left(x\right) = c_m {x_{go}}^m + c_n {x_{go}}^n = \left[{x_{go}}^m \; {x_{go}}^n\right] \begin{bmatrix} c_m \\ c_n \end{bmatrix} \tag{14}$$

where $x_{go} = x\left(t_f\right) - x = -x$, m and n are the design parameters satisfying $m > n \geq 2$, and c_m, c_n are the constant coefficients. The derivatives of Eq. (14) can be written as follows.

$$y'\left(x\right) = -\left[m{x_{go}}^{m-1} \; n{x_{go}}^{n-1}\right] \begin{bmatrix} c_m \\ c_n \end{bmatrix} \tag{15}$$

$$y''\left(x\right) = \left[m\left(m-1\right){x_{go}}^{m-2} \; n\left(n-1\right){x_{go}}^{n-2}\right] \begin{bmatrix} c_m \\ c_n \end{bmatrix} \tag{16}$$

It can be concluded from Eqs. (14)–(16) that the constraints of Eqs. (11)–(13), which are equivalent to the constraints of Eqs. (8)–(10), will be satisfied if m and n are chosen to be $m > n > 2$. If $m > n > 3$, then the additional constraint on the terminal jerk, namely $\dot{a}_M^n\left(t_f\right) = 0$, can be satisfied at the same time.

3.1.2 Determination of Coefficients

Equations (14) and (15) can be augmented into matrix form as follows.

$$\begin{bmatrix} y\left(x\right) \\ y'\left(x\right) \end{bmatrix} = \begin{bmatrix} {x_{go}}^m & {x_{go}}^n \\ -m{x_{go}}^{m-1} & -n{x_{go}}^{n-1} \end{bmatrix} \begin{bmatrix} c_m \\ c_n \end{bmatrix} \tag{17}$$

The coefficients c_m and c_n can be determined by the initial conditions. Since downrange is used as a new independent variable in this study, the initial conditions given in Eq. (2) can be rewritten as follows.

$$\begin{aligned} y\left(x_0\right) &= y_0 \\ \bar{\gamma}_M\left(x_0\right) &= \bar{\gamma}_{M_0} \end{aligned} \tag{18}$$

Using Eqs. (3) and (18), the initial slope of the crossrange pattern can be calculated as

$$y'\left(x_0\right) = \tan \bar{\gamma}_{M_0} \tag{19}$$

Considering Eqs. (18) and (19) in Eq. (17), the coefficients can be obtained as follows

$$\begin{aligned}\begin{bmatrix} c_m \\ c_n \end{bmatrix} &= \begin{bmatrix} {x_{go_0}}^m & {x_{go_0}}^n \\ -m{x_{go_0}}^{m-1} & -n{x_{go_0}}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_0 \\ \tan\bar{\gamma}_{M_0} \end{bmatrix} \\ &= \frac{1}{m-n} \begin{bmatrix} -n{x_{go_0}}^{-m} & -{x_{go_0}}^{-m+1} \\ m{x_{go_0}}^{-n} & {x_{go_0}}^{-n+1} \end{bmatrix} \begin{bmatrix} y_0 \\ \tan\bar{\gamma}_{M_0} \end{bmatrix} \\ &= \frac{1}{m-n} \begin{bmatrix} -\left(ny_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right){x_{go_0}}^{-m} \\ \left(my_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right){x_{go_0}}^{-n} \end{bmatrix}\end{aligned} \tag{20}$$

where $x_{go_0} = -x_0 > 0$.

3.1.3 Closed-Form Solution

Substituting Eq. (20) into Eq. (17) and considering Eq. (3) yields the closed-form solution for the desired crossrange trajectory in terms of downrange-to-go.

$$\begin{bmatrix} y(x) \\ \tan\bar{\gamma}_M(x) \end{bmatrix} = \begin{bmatrix} {x_{go}}^m & {x_{go}}^n \\ -m{x_{go}}^{m-1} & -n{x_{go}}^{n-1} \end{bmatrix} \begin{bmatrix} {x_{go_0}}^m & {x_{go_0}}^n \\ -m{x_{go_0}}^{m-1} & -n{x_{go_0}}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_0 \\ \tan\bar{\gamma}_{M_0} \end{bmatrix} \tag{21}$$

Because the coefficients obtained in Eq. (20) are constants, Eq. (21) can be reinterpreted as the existence of an invariant quantity **c** throughout the engagement.

$$\begin{aligned}\begin{bmatrix} {x_{go}}^m & {x_{go}}^n \\ -m{x_{go}}^{m-1} & -n{x_{go}}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y \\ \tan\bar{\gamma}_M \end{bmatrix} &= \begin{bmatrix} {x_{go_0}}^m & {x_{go_0}}^n \\ -m{x_{go_0}}^{m-1} & -n{x_{go_0}}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_0 \\ \tan\bar{\gamma}_{M_0} \end{bmatrix} \\ &= \mathbf{c}\end{aligned} \tag{22}$$

3.1.4 Guidance Command

Equation (6) can be rewritten using Eqs. (3) and (16) as

$$\begin{aligned} a_M^n(x) &= y''(x)\cos^3\bar{\gamma}_M(x)\,{V_M}^2 \\ &= \left[m(m-1){x_{go}}^{m-2} \;\; n(n-1){x_{go}}^{n-2}\right] \begin{bmatrix} c_m \\ c_n \end{bmatrix} \cos^3\bar{\gamma}_M(x)\,{V_M}^2 \end{aligned} \tag{23}$$

The open-loop form guidance command can be obtained by substituting Eq. (20) into Eq. (23) as follows.

$$
\begin{aligned}
a^n_{M_{\text{cmd}}} &= \left[m\left(m-1\right){x_{go}}^{m-2}\;\, n\left(n-1\right){x_{go}}^{n-2}\right]\begin{bmatrix} c_m \\ c_n \end{bmatrix}\cos^3\bar{\gamma}_M {V_M}^2 \\
&= \left[m\left(m-1\right){x_{go}}^{m-2}\;\, n\left(n-1\right){x_{go}}^{n-2}\right]\begin{bmatrix} -n{x_{go_0}}^{-m} & -{x_{go_0}}^{-m+1} \\ m{x_{go_0}}^{-n} & {x_{go_0}}^{-n+1} \end{bmatrix}\begin{bmatrix} y_0 \\ \tan\bar{\gamma}_{M_0} \end{bmatrix} \\
&\quad\cdot \frac{1}{m-n}\cos^3\bar{\gamma}_M {V_M}^2 \\
&= -\left[\frac{mn}{x_{go}^2}\left\{\left(m-1\right)\left(\frac{x_{go}}{x_{go_0}}\right)^m - \left(n-1\right)\left(\frac{x_{go}}{x_{go_0}}\right)^n\right\} y_0\right. \\
&\qquad \left. + \frac{1}{x_{go}}\left\{m\left(m-1\right)\left(\frac{x_{go}}{x_{go_0}}\right)^{m-1} - n\left(n-1\right)\left(\frac{x_{go}}{x_{go_0}}\right)^{n-1}\right\}\tan\bar{\gamma}_{M_0}\right] \\
&\cdot \frac{1}{m-n}\cos^3\bar{\gamma}_M {V_M}^2
\end{aligned}
\tag{24}
$$

Because of the relation shown in Eq. (22), finally, the guidance command can be rewritten in closed-loop feedback form as follows.

$$
a^n_{M_{\text{cmd}}} = -\left(\frac{mn}{{x_{go}}^2}y + \frac{m+n-1}{x_{go}}\tan\bar{\gamma}_M\right)\cos^3\bar{\gamma}_M {V_M}^2 \tag{25}
$$

3.2 Properties of Guidance Law

Using the guidance law of Eq. (25), the missile will follow the desired trajectory given by Eq. (14). The critical points and the inflection points of $y\left(x\right)$, at which y' and y'' equals to zero, respectively, occurs at

$$
\begin{aligned}
y' = 0: &\quad x_{go} = 0, \\
&\quad x_{go_{crt}} = \left[-\frac{c_n n}{c_m m}\right]^{\frac{1}{m-n}} = \left[\frac{n\left(my_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right)}{m\left(ny_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right)}\right]^{\frac{1}{m-n}} x_{go_0} \\
y'' = 0: &\quad x_{go} = 0, \\
&\quad x_{go_{inf}} = \left[-\frac{c_n n\left(n-1\right)}{c_m m\left(m-1\right)}\right]^{\frac{1}{m-n}} = \left[\frac{n\left(n-1\right)\left(my_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right)}{m\left(m-1\right)\left(ny_0 + x_{go_0}\tan\bar{\gamma}_{M_0}\right)}\right]^{\frac{1}{m-n}} x_{go_0}
\end{aligned}
\tag{26}
$$

Note that the missile with trajectory $y\left(x\right)$ will change its turning direction at the critical points, and the lateral acceleration will be zero at the inflection points. It can be observed in Eq. (26) that the critical and inflection points of the desired trajectory occurs at the points with certain ratios to the initial downrange-to-go.

According to the extreme value theorem, the maximum $\left|y''\right|$ may occur at $x_{go} = x_{go_0}$, $x_{go} = 0$, or $x_{go} = x_{go_{inf}}$. If $m > n \geq 2$, it is trivial that $y''\left(0\right) = 0$ by the proposed design, and therefore,

$$\max_{x} \left|y''(x)\right| = \max\left(\left|y''(x_0)\right|, \left|y''\left(-x_{go_{inf}}\right)\right|\right) \tag{27}$$

For the curvature κ given in Eq. (6), the following is always true.

$$|\kappa(x)| = \frac{\left|y''(x)\right|}{\left(1 + y'(x)^2\right)^{\frac{3}{2}}} \leq \left|y''(x)\right| \tag{28}$$

Therefore, from Eqs. (27) and (28), the boundedness of curvature can be guaranteed.

$$|\kappa(x)| \leq \max\left(\left|y''(x_0)\right|, \left|y''\left(-x_{go_{inf}}\right)\right|\right) \tag{29}$$

Note from Eq. (6) that the upper bound on curvature implies that of lateral acceleration, and consequently, Eq. (29) can be used to adjust the desired trajectory considering the manoeuvrability limit of missile.

The desired terminal constraints can be achieved with the proposed trajectory shaping guidance law based on the desired crossrange pattern given by a polynomial of downrange-to-go. The characteristics of engagement can be controlled with the choice of design parameters m and n. As mentioned in introduction, the proposed guidance law of Eq. (25) does not depend on (1) linearized engagement kinematics, (2) constant speed assumption, and (3) inaccurate time-to-go estimate. Exact nonlinear engagement kinematics equations are considered in the design process of the proposed guidance law without constant speed assumption. Also, the downrange-to-go entering into the guidance command can be obtained without any approximation.

4 Numerical Simulation

Numerical simulation is performed to demonstrate the effectiveness of the proposed guidance law. The simulation results are described in this section.

4.1 Simulation Environment

The closed-loop form guidance command is utilized to demonstrate the proposed method. The horizontal plane is assumed to be the engagement plane for the simulation. To include the effect of speed change in the simulation, the following simple model is utilized

$$\dot{V}_M = -\frac{C_{D_0}\rho S}{2\bar{m}} {V_M}^2 \tag{30}$$

Table 1 Simulation parameters for the missile and the target

Parameter	Value	Unit
$\left(X_{M_I}(t_0), Y_{M_I}(t_0)\right)$	(0, 1000)	m
$\left(X_{T_I}, Y_{T_I}\right)$	(4000, 0)	m
$V_M(t_0)$	200	m/s
C_{D_0}	0.02	–
$\bar{m}$	90.035	kg
ρ	1.2041	kg/m^3
S	0.2	m^2

Table 2 Guidance parameters, initial flight path angle, and terminal impact angle

	(m, n)	γ_{M_0} [deg]	γ_{f_d} [deg]
Case 1	(5, 4)	0	$-75:15:0$
Case 2	(5, 4)	$-90:15:30$	-45
Case 3	(6, 5), (5, 5), (5, 4), (5, 3), (4, 3), (3, 2)	0	-45

where C_{D_0} is the zero-lift drag coefficient, ρ is the atmospheric density, $\bar{m}$ and S are the mass and reference area of the missile, respectively.

Three simulation cases are considered in this study. In Case 1, simulation is performed using the proposed guidance law with $(m, n) = (5, 4)$ for various terminal impact angles with fixed initial flight path angle, and vice versa in Case 2. In Case 3, simulation is performed for various design parameters with fixed terminal impact angle and initial flight path angle. Initial position of the missile and the target for Cases 1–3, and the physical parameters of the missile are summarized in Table 1. Guidance parameters (m, n), initial flight path angle, and the terminal impact angle for Cases 1–3 are summarized in Table 2.

4.2 Simulation Results

4.2.1 Case 1: Various γ_{f_d}, Fixed γ_{M_0} and (m, n)

Figures 2, 3, 4, 5 and 6 show the trajectories in the inertial coordinate system, the lateral acceleration commands, the speed histories, the crossranges with respect to the impact angle coordinate system, and the flight path angle errors, respectively.

Simulation results of Case 1 shows that various desired terminal impact directions can be handled with the proposed guidance law.

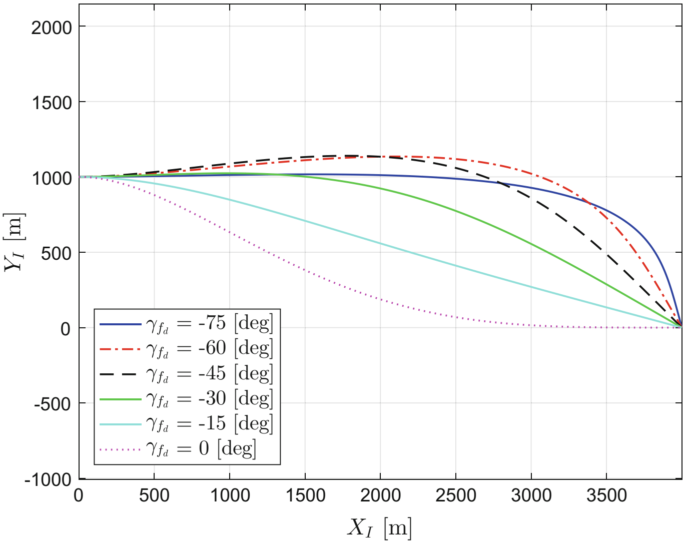


Fig. 2 Case 1: X_I-Y_I trajectory

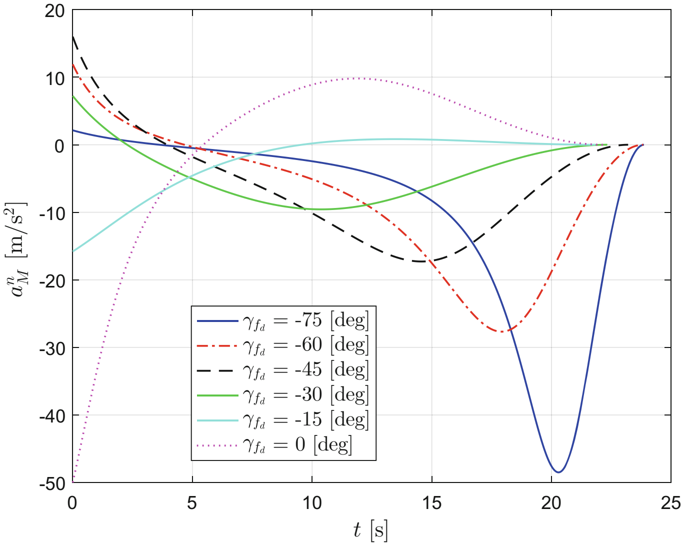


Fig. 3 Case 1: lateral acceleration a_M^n

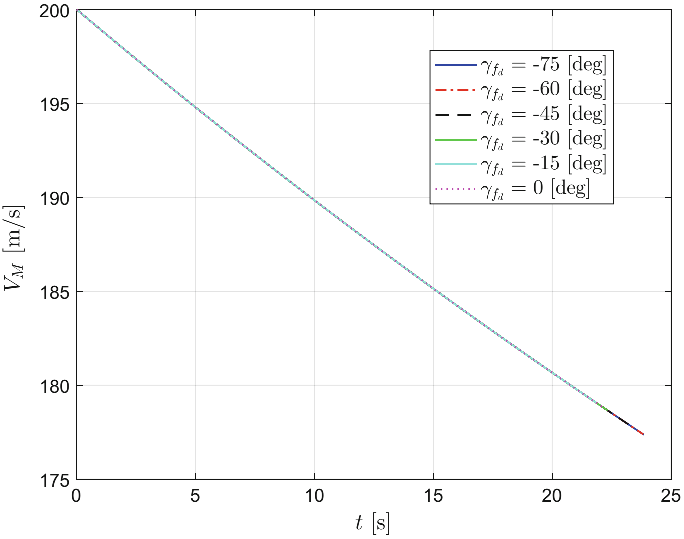


Fig. 4 Case 1: speed V_M

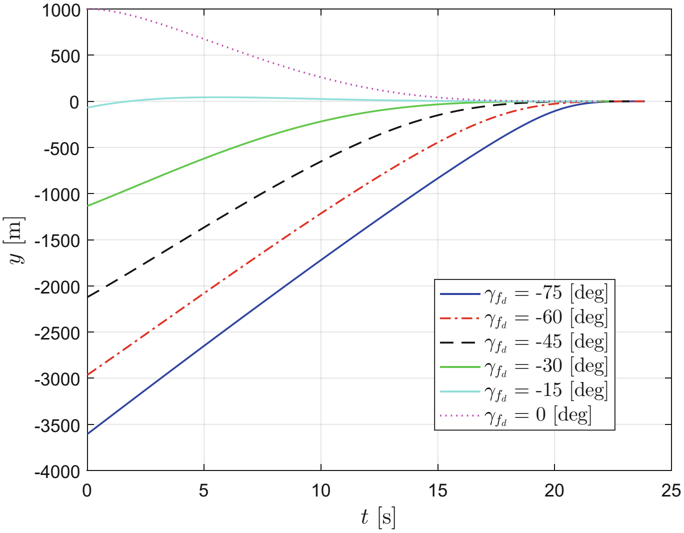


Fig. 5 Case 1: crossrange y

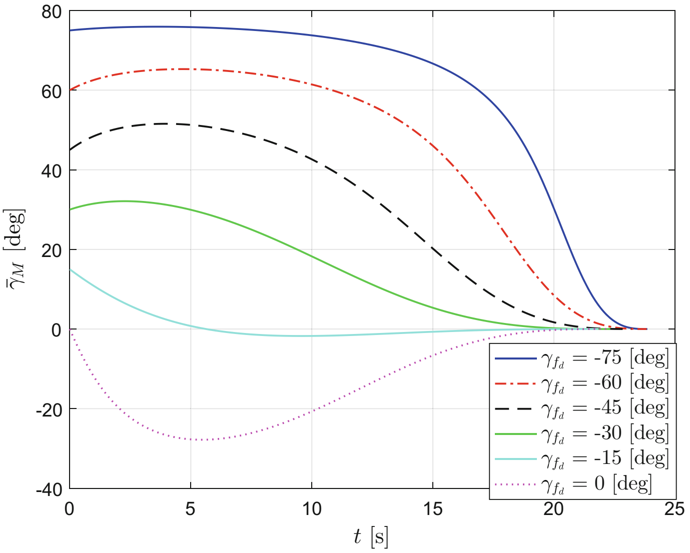


Fig. 6 Case 1: flight path angle error $\bar{\gamma}_M$

4.2.2 Case 2: Various γ_{M_0}, Fixed γ_{f_d} and (m, n)

Figures 7, 8, 9, 10 and 11 show the trajectories in the inertial coordinate system, the lateral acceleration commands, the speed histories, the crossranges with respect to the impact angle coordinate system, and the flight path angle errors, respectively.

Simulation results of Case 2 shows that the proposed guidance law can cope with various initial flight path angles, as long as $\left|\bar{\gamma}_{M_0}\right| < \frac{\pi}{2}$. A limitation of the proposed guidance law is that the missile flies in a pattern that x_{go} is always decreasing with respect to time, but this is not a severe restriction.

4.2.3 Case 3: Various (m, n), Fixed γ_{M_0} and γ_{f_d}

Figures 12, 13, 14, 15 and 16 show the trajectories in the inertial coordinate system, the lateral acceleration commands, the speed histories, the crossranges with respect to the impact angle coordinate system, and the flight path angle errors, respectively.

For a given initial flight path angle and a terminal impact angle, Case 3 shows that the engagement trajectory can be adjusted by the choice of design parameters m and n. The case of $(m, n) = (3, 2)$ is included in Case 3 to show that the terminal acceleration constraint can be met only if $m > n > 2$. Also, the cases of $(m, n) = (5, 3)$, $(4, 3)$ are included in Case 3 to show that the terminal jerk constraint can be met only if $m > n > 3$. In addition, Fig. 13 shows that the magnitude of the initial lateral acceleration increases with greater $m + n$. Furthermore, interception is achieved while the missile is decelerating in all cases. Therefore, it can be concluded

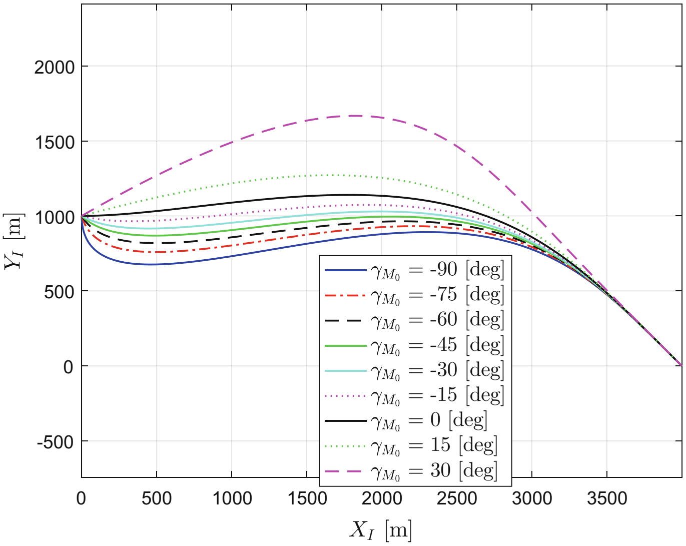


Fig. 7 Case 2: X_I-Y_I trajectory

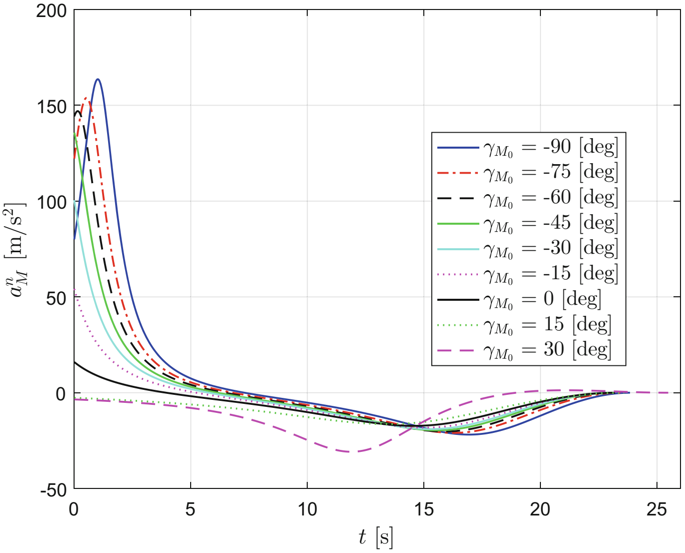


Fig. 8 Case 2: lateral acceleration a_M^n

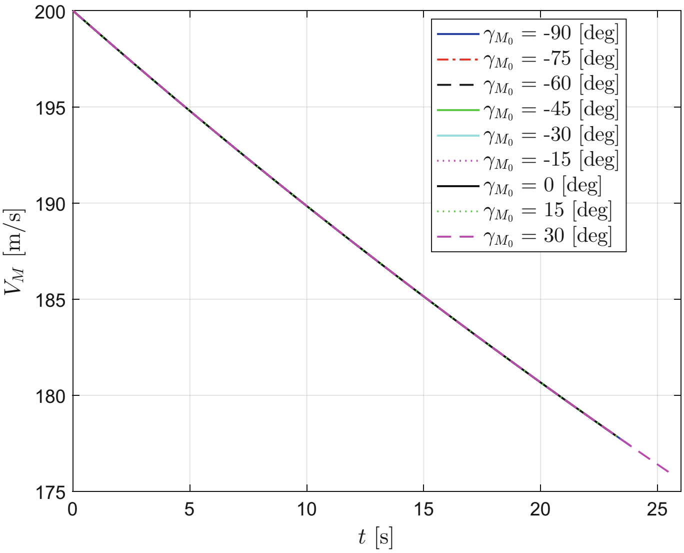


Fig. 9 Case 2: speed V_M

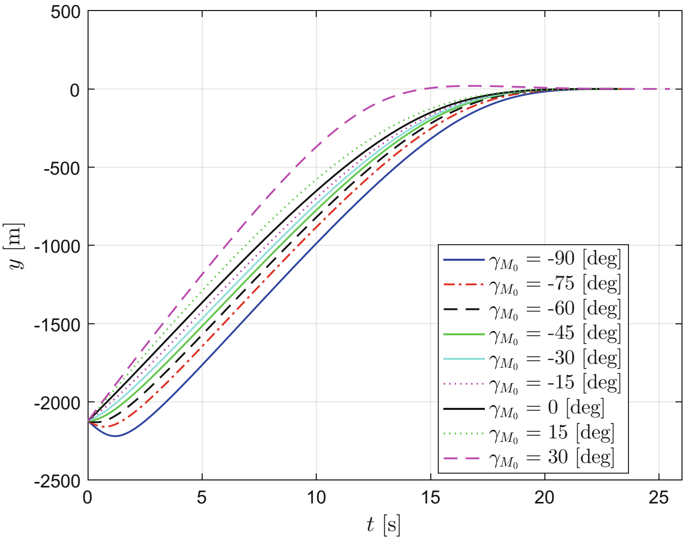


Fig. 10 Case 2: crossrange y

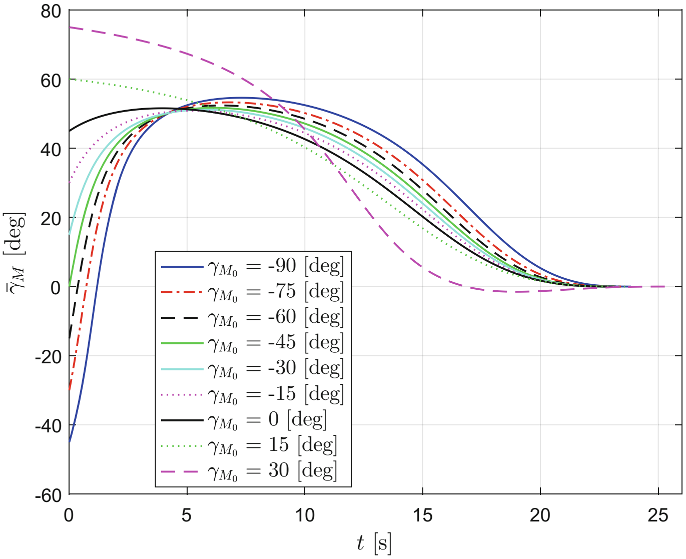


Fig. 11 Case 2: flight path angle error $\bar{\gamma}_M$

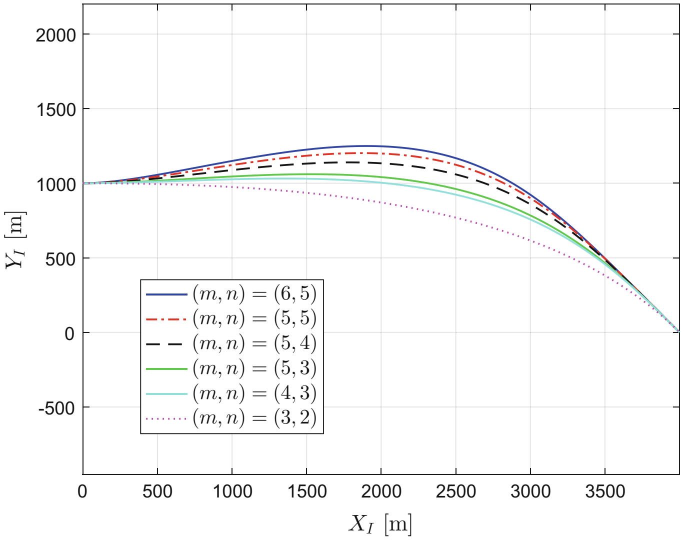


Fig. 12 Case 3: X_I-Y_I trajectory

from the results of Cases 1–3 that the proposed guidance law can achieve stationary target interception with given terminal impact angle and acceleration constraints, together with the flexibility of shaping.

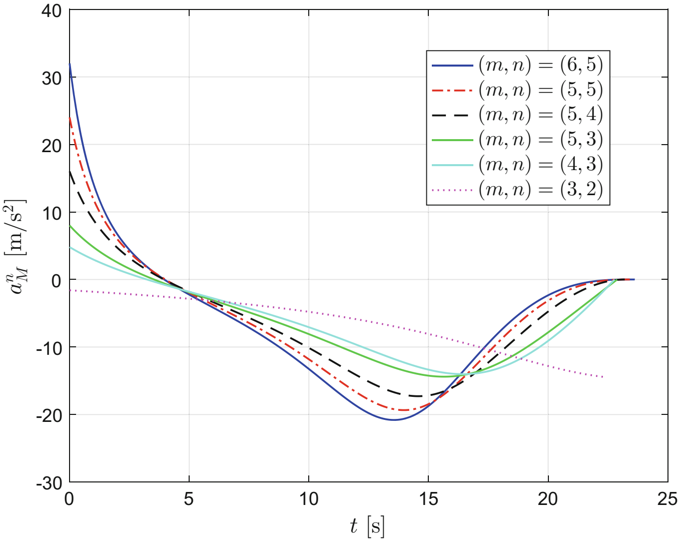


Fig. 13 Case 3: lateral acceleration a_M^n

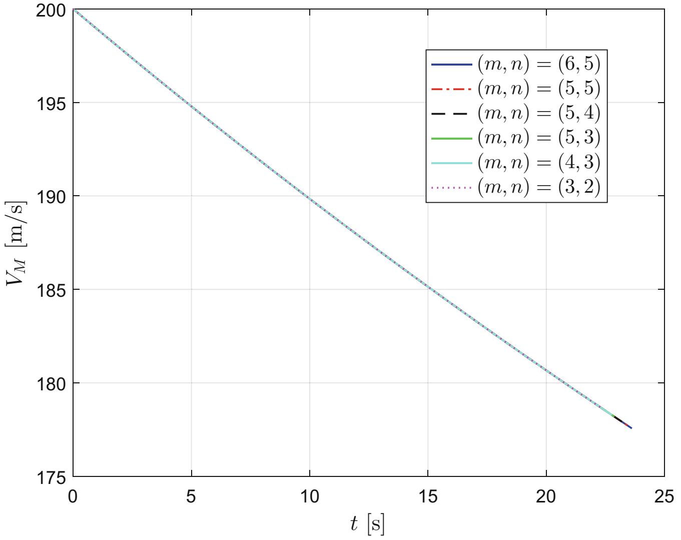


Fig. 14 Case 3: speed V_M

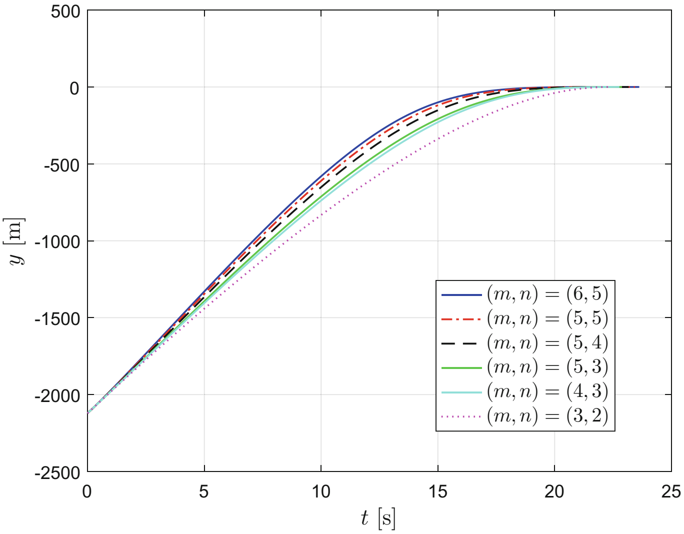


Fig. 15 Case 3: crossrange y

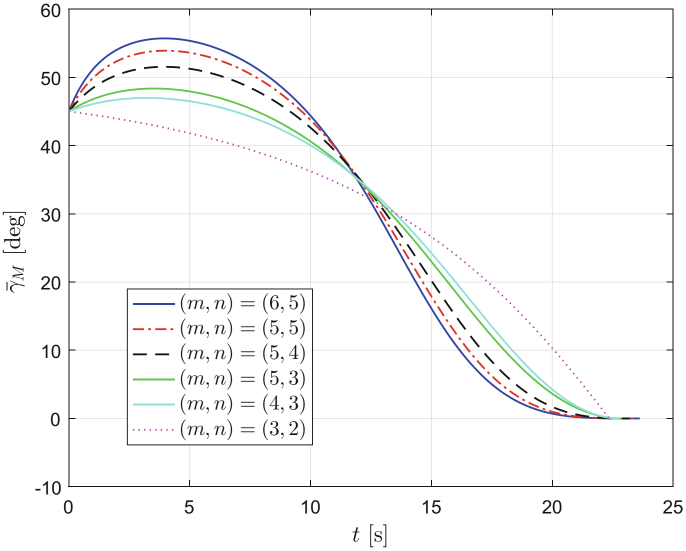


Fig. 16 Case 3: flight path angle error $\bar{\gamma}_M$

5 Conclusion

A new trajectory shaping guidance law was proposed to achieve interception of a stationary target with terminal flight path angle and acceleration constraints. The desired crossrange pattern was designed as a polynomial of downrange-to-go, which might supplement the existing trajectory shaping guidance law. The proposed guidance law was derived without linearization of engagement kinematics, and constant speed assumption. Also, the time-to-go estimate is not required to implement the proposed guidance law.

Acknowledgements This work was conducted at High-Speed Vehicle Research Center of KAIST with the support of Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

References

1. Kim H, Kim HJ (2015) Missile guidance law considering constraints on impact angle and terminal angle of attack. In: AIAA guidance, navigation, and control conference, Kissimmee, FL. https://doi.org/10.2514/6.2015-0861
2. Kim TH, Lee CH, Tahk MJ (2011) Time-to-go polynomial guidance laws with terminal impact angle/acceleration constraints. In: 18th IFAC world congress, Milano, Italy. https://doi.org/10.3182/20110828-6-IT-1002.02304
3. Lee CH (2013) Optimal guidance laws using generalized weighting functions. PhD thesis, Korea Advanced Institute of Science and Technology
4. Lee CH, Kim TH, Tahk MJ, Whang IH (2013) Polynomial guidance laws considering terminal impact angle and acceleration constraints. IEEE Trans Aerosp Electron Syst 49(1):74–92. https://doi.org/10.1109/TAES.2013.6404092
5. Lee YI, Ryoo CK, Kim E (2003) Optimal guidance with constraints on impact angle and terminal acceleration. In: AIAA guidance, navigation, and control conference, Austin, TX. https://doi.org/10.2514/6.2003-5795
6. Moon HB, Jung YK, Ra WS (2015) Terminal acceleration stabilizing guidance law for impact angle constrained interception of a non-maneuvering target. Int J Control Autom Syst 13(6):1410–1422. https://doi.org/10.1007/s12555-014-0196-z
7. Ohlmeyer EJ (2003) Control of terminal engagement geometry using generalized vector explicit guidance. In: American control conference, Denver, CO. https://doi.org/10.1109/ACC.2003.1238981
8. Ohlmeyer EJ, Phillips CA (2006) Generalized vector explicit guidance. J Guidance Control Dyn 29(2):261–268. https://doi.org/10.2514/1.14956
9. Ryoo CK, Cho H, Tahk MJ (2006) Time-to-go weighted optimal guidance with impact angle constraints. IEEE Trans Control Syst Technol 14(3):483–492. https://doi.org/10.1109/TCST.2006.872525
10. Ryoo CK, Kim HJ, Tahk MJ, Lee JI (2009) Optimal guidance law: impact angle & terminal lateral acceleration control. In: 14th IFAC workshop on control applications of optimization, Jyväskylä, Finland. https://doi.org/10.3182/20090506-3-SF-4003.00059

MIMO Attitude Control for a Spinning Rocket

W.C. Leite Filho, J. Guimaraes and L. Galembeck

1 Introduction

When dealing with rockets that present no direct roll control, it is possible that disturbances along the system may introduce some roll rate [10]. The highly coupled nature of the rocket's equations of motion make the study of its dynamic a very complex subject [7, 9]. It is necessary to understand how well a given control system is able to deal with such perturbations.

In this paper, we are interested in studying the performance of a multiple-input-multiple-output (MIMO) controller with roll input and how it compares to a single-input-single-output (SISO) PID controller when used on a rocket under the influence of an uncontrolled spin rate of parabolic profile. The controllers are compared by analyzing both step response and the ability to send the system to a pre-determined trajectory, with and without the influence of wind perturbations.

2 Mathematical Model

Given the presence of uncontrolled spin, it is necessary to consider the complete equations of motion for a sounding rocket with thrust vectoring. The equations are initially derived for the body reference frame, and later related to the inertial reference frame through Euler's differential equations.

The complete motion of the vehicle, as described in the body reference, is given by the set of Eq. (1).

W.C. Leite Filho (✉) · J. Guimaraes · L. Galembeck
Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos, Brazil
e-mail: waldclf@gmail.com

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_31

$$\begin{cases} \dot{u} = -\left(\frac{C_d P_d S}{m}\right) + \frac{T}{m} - g\cos\theta\cos\psi - qw + rv \\ \dot{v} = -Y_\beta \beta + Y_{\beta_y}\beta_y - g(\sin\theta\sin\phi - \cos\theta\sin\psi\cos\phi) - ru + pw \\ \dot{w} = -Z_\alpha \alpha + Z_{\beta_z}\beta_z - g(\sin\theta\cos\phi - \cos\theta\sin\psi\sin\phi) - pv + qu \\ \dot{p} = -L_p p + \frac{I_{yy}-I_{zz}}{I_{xx}} qr \\ \dot{q} = -M_q q + \frac{I_{zz}-I_{xx}}{I_{yy}} pr + M_\alpha \alpha - M_{\beta_z}\beta_z \\ \dot{r} = -N_r r + \frac{I_{xx}-I_{yy}}{I_{zz}} pq - N_\beta \beta + N_{\beta_y}\beta_y \end{cases} \tag{1}$$

The relationship between the angular velocities in the body reference frame and the Euler angles (for a rotation sequence of $\theta - \psi - \phi$) is given by the set of Eq. (2).

$$\begin{cases} \dot{\phi} = p - q\tan\psi\cos\phi + r\tan\psi\sin\phi \\ \dot{\theta} = q\frac{\cos\phi}{\cos\psi} - r\frac{\sin\phi}{\cos\psi} \\ \dot{\psi} = q\sin\phi + r\cos\phi \end{cases} \tag{2}$$

If the rocket has no spin rate (p = 0), the equations can be decoupled and the system becomes completely independent in the θ and ψ directions.

2.1 Resolver

When it comes to real implementation on a spinning rocket, one must consider how the sensors measurements for a given direction correspond to the real Euler angles that describe the vehicle's attitude. In order to relate the measured errors to the modeled angles, a resolver must be used. Its implementation is illustrated by Fig. 1.

For small values of ψ, its behavior can be represented by the set of Eq. (3).

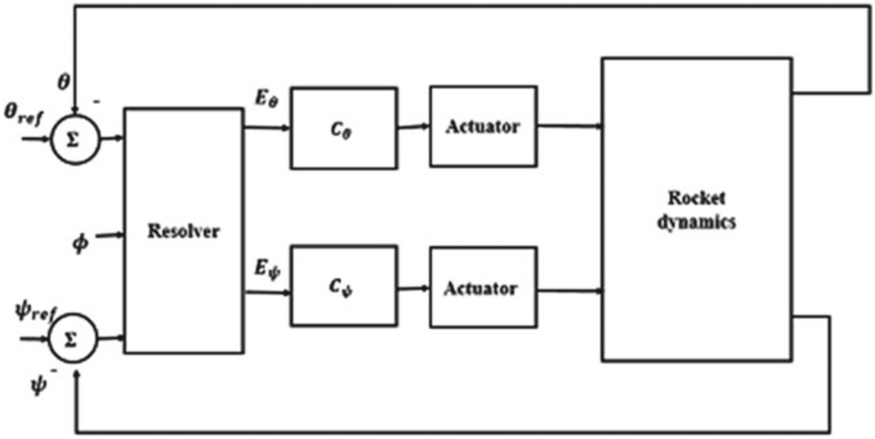


Fig. 1 Resolver implementation

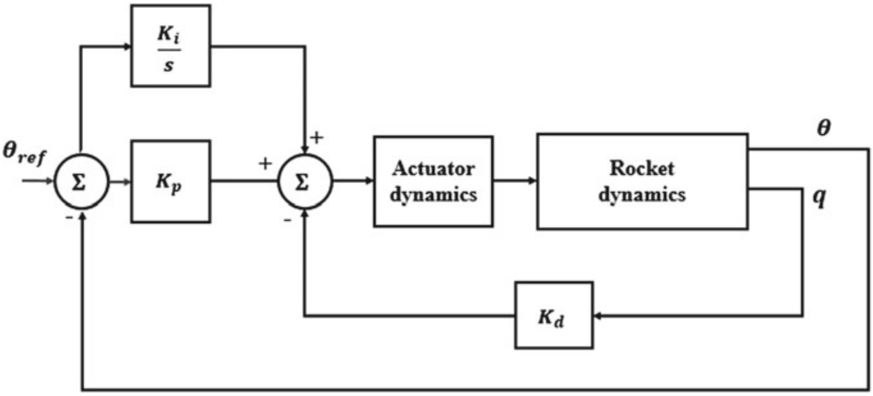


Fig. 2 SISO classic PID control scheme

$$\begin{cases} E_\theta = (\theta_{ref} - \theta)\cos\phi + (\psi_{ref} - \psi)\sin\phi \\ E_\psi = (\theta - \theta_{ref})\sin\phi + (\psi_{ref} - \psi)\cos\phi \end{cases} \tag{3}$$

3 Control Strategies

This paper considers two different control strategies to deal with the residual spin: the first is a decoupled system for pitch and yaw, with two independent single-input-single-output (SISO) controllers and the second, a coupled multiple-input-multiple-output (MIMO) controller.

3.1 Single-Input-Single-Output (SISO)

The initial control strategy consists of two independent PID controllers, designed separately and without taking the spin and other possible interactions into account. Given the symmetry of the vehicle, the same gains are used for pitch and yaw.

The main advantage in the use of SISO controllers rests on its simplicity and well known techniques of gain design [6]. However, it is possible this system is unable to control the real spinning rocket, depending on the rate of rotation.

Figure 2 illustrates a PID SISO controller applied in conjunction with an actuator.

3.2 Multiple-Input-Multiple-Output (MIMO)

Multiple-Input-Multiple-Output controllers allow for better control over interdependent variables [8] and are useful in problems where coupling is relevant.

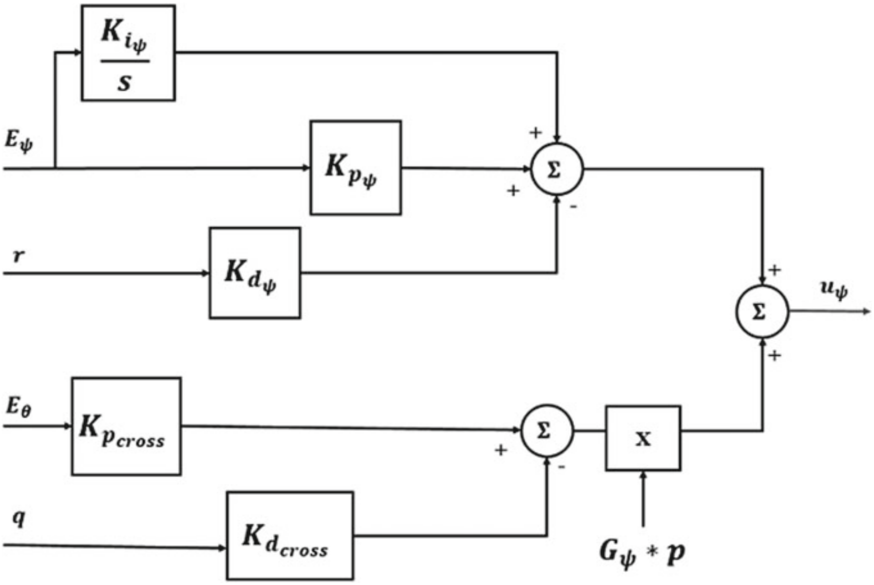


Fig. 3 MIMO structure for yaw control

In its most general formulation when applied to a rocket, a MIMO controller would consider the reference signals to both pitch and yaw simultaneously and generate a control law that attempts to negate the cross-terms, as seen in [10]. However, this approach does not consider the roll rate directly.

The control system proposed here considers a nonlinear MIMO controller that uses not only pitch and yaw measurements but also the roll measurement. It consists of a PID structure for the direct branch and a PD structure for the cross branch. The cross portion of the control signal is then multiplied by the spin rate measured, as well as a fixed gain G. Figure 3 illustrates the structure used for the yaw control signal.

4 Case Study

The vehicle studied presents a design residual spin rate that follows a parabolic profile given by Eq. (4).

$$p(t) = \frac{t^2}{30}[\deg/\mathrm{s}] \tag{4}$$

As part of the design requirements, the step response for the controller should present a settling time of less than four seconds, an overshoot of less than 40% and the smallest ramp error possible. Given structural and physical limitations, it is also

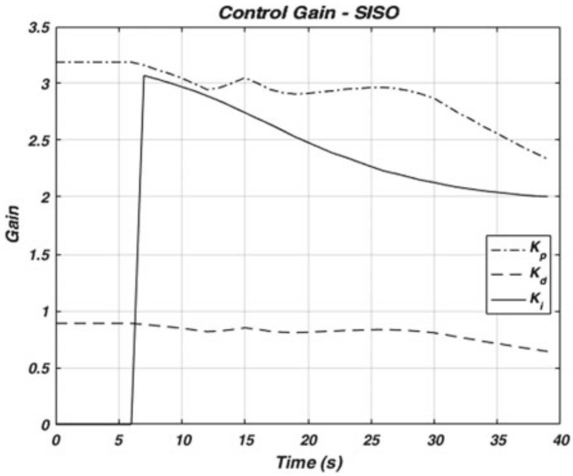


Fig. 4 Control gains – SISO controller

required that the control command never surpasses 4 degrees and that the rising time is larger than 0.5 s.

For simulation purposes, the actuator used will be modeled by a first order system with $\omega = 4.5\,\text{Hz}$.

4.1 SISO Controller Gains

Since the vehicle's parameters change constantly throughout the flight, there must be an appropriate gain scheduling. The PID gains for the SISO approach were tuned in one second intervals using the step and ramp requirements. It is important to understand that those requirements are often contradictory, so eventual trade-offs were made.

The tuning strategy used starts with the critical flight condition, that is, the instant where M_α is maximum, and the gains are calculated so that setting time and minimum ramp error are prioritized. Then, the gains are scheduled using the ratio of thrust coefficients with respect to the critical time, as well as performance metrics.

The gains calculated for the SISO controller are shown in Fig. 4. The integral part of the controller is not used for the first five seconds, in order to avoid eventual liftoff errors.

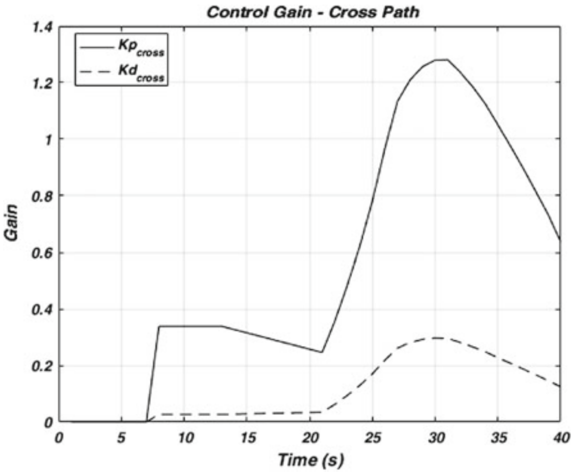


Fig. 5 Control gains – MIMO – cross path

4.2 *MIMO Controller Gains*

For the purpose of comparison, the direct gains for the MIMO controller were considered the same as the SISO case, shown in Fig. 4. The cross gains for the MIMO controllers were calculated using the nonconvex nonsmooth optimization algorithms described in [2–5] through the MATLAB function *systune*.

The gains were tuned so that the system showed the desired step response. The tuning requirements used that resulted in the best results were gain margin of 5dB, phase margin of 30°, a 20% maximum overshoot and a natural frequency close to the SISO step response. During the tuning for each time snapshot, the roll gain was considered unitary (i.e. $G_\theta = -1$ and $G_\psi = +1$). Figure 5 shows the cross gains calculated through this method.

5 Performance Analysis

For comparing the two control systems, their performance was first analyzed with respect to their separate pitch and yaw step response for the critical time, both in ideal conditions and under wind perturbation. Then, the ability to follow the design trajectory with respect to its Euler angles was studied under both conditions. In both cases, the wind perturbation was modelled by a simplified model as a half-sine of 10 m/s of amplitude and a 2 s half-period.

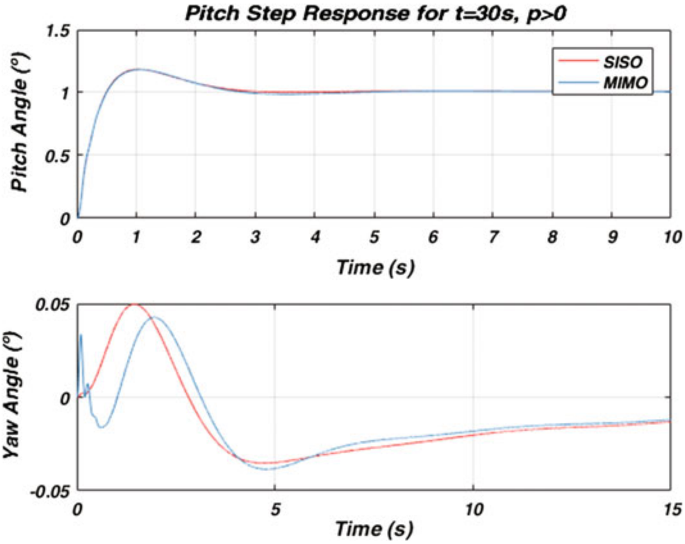


Fig. 6 Pitch step response for critical time (t = 30 s) – p > 0

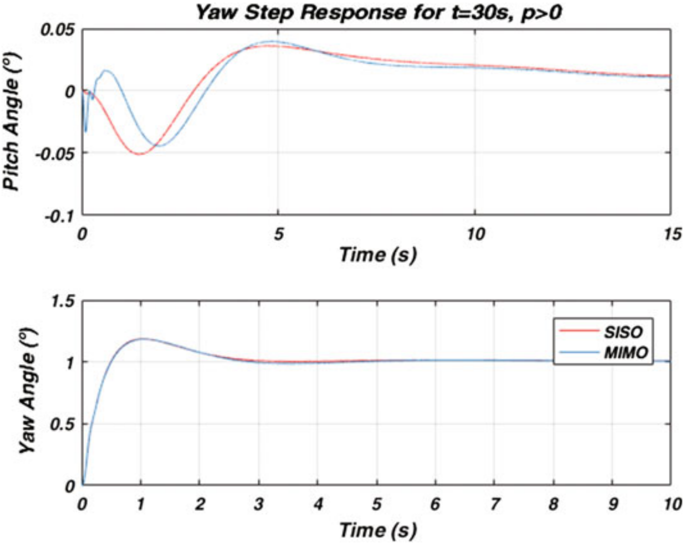


Fig. 7 Yaw step response for critical time (t = 30 s) – p > 0

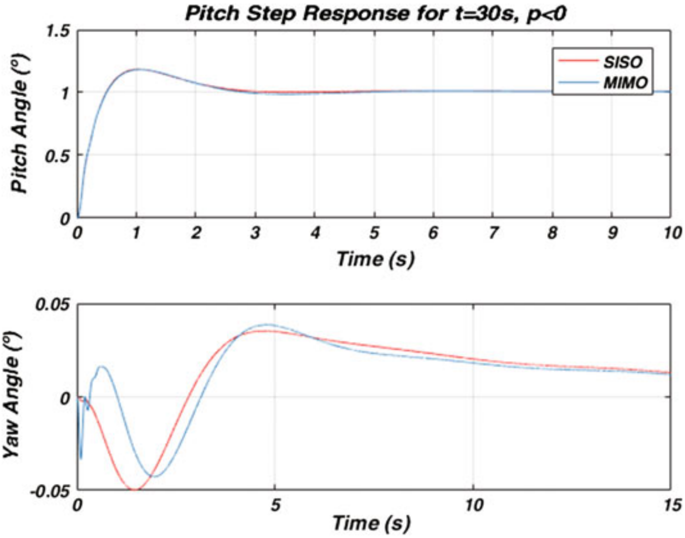


Fig. 8 Pitch step response for critical time (t = 30 s) – p < 0

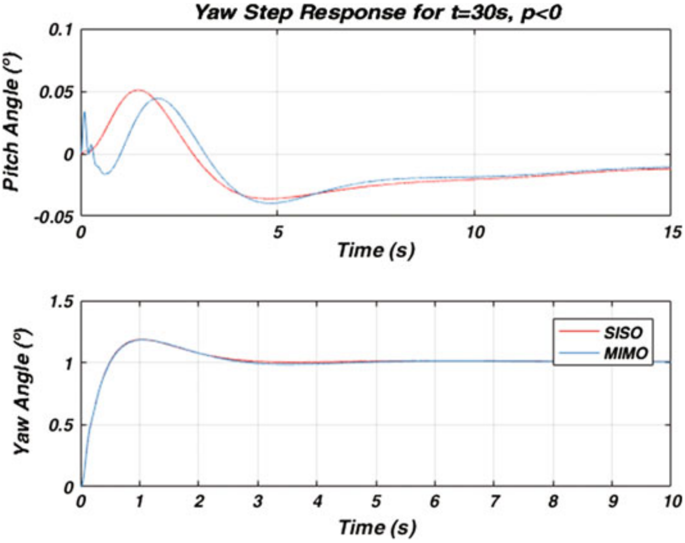


Fig. 9 Yaw step response for critical time (t = 30 s) – p < 0

5.1 Step Response

The step response for the critical time when M_α is maximum ($t = 30\,s$) can be seen on Figs. 6 and 7. Figures 8 and 9 show the corresponding step response for a negative roll rate that follows the same parabolic profile. For the purpose of the step response analysis, the roll gain G was considered unitary and steps were applied separately to each angle reference.

The step response for the MIMO controller, in part due to the tuning process used, is quite similar to the SISO controller. The angle not receiving the step reference shows small oscillations under the MIMO controller, due to the initial large error of the step angle stimulating the cross path.

The step requirements presented on Sect. 4 are respected for each time snapshot even with the MIMO controller, with raising time larger than 0.5 s, overshoot smaller than 40% and settling time smaller than 4 s for each variable under the step reference. Finally, the step response of the stimulated angle showed no undershoot for any time considered, which is desired for this kind of thrust vectoring control (TVC).

5.2 Step Response Under Wind Perturbation

The wind perturbation analysis to the step response considers a gust wind applied at the start of the simulation.

The influence of the wind on the pitch step response can be seen on Figs. 10 and 11.

The effect on the yaw step response is similar. Both images show how the rocket would respond to such perturbations if there were no roll, in which case the two controllers would be equivalent.

It is possible to see that the MIMO controller responds well to the wind perturbations applied. Once again, small oscillations can be seen at the start, but after that the MIMO step response on the opposite angle seems to be either equal or slightly better than the SISO one. The SISO controller creates a higher overshoot on the opposite angle during the transient response for most cases.

It is important to note that in both cases the magnitude of the perturbation on the opposite channel is relatively small.

5.3 Design Trajectory

Initially, the MIMO roll gains were considered, once again, unitary. Figure 12 shows how well the controllers lead the vehicle to a design trajectory for a positive spin rate, while Fig. 13 refers to a negative spin rate.

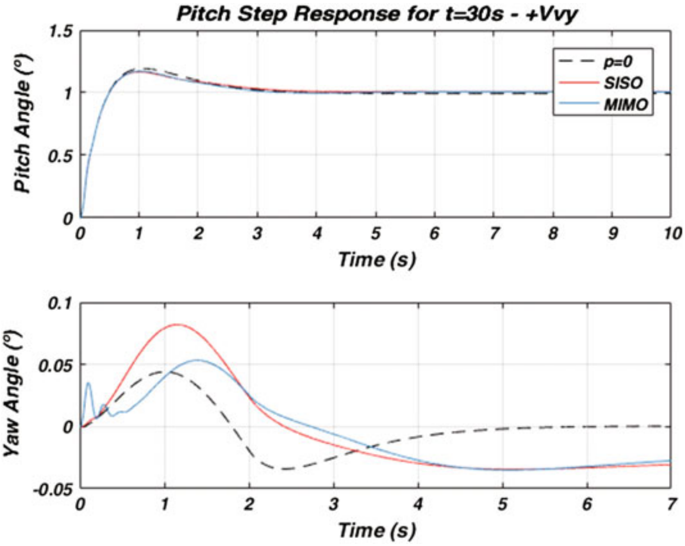


Fig. 10 Pitch step response for critical time (t = 30 s) under wind perturbation - Vvy

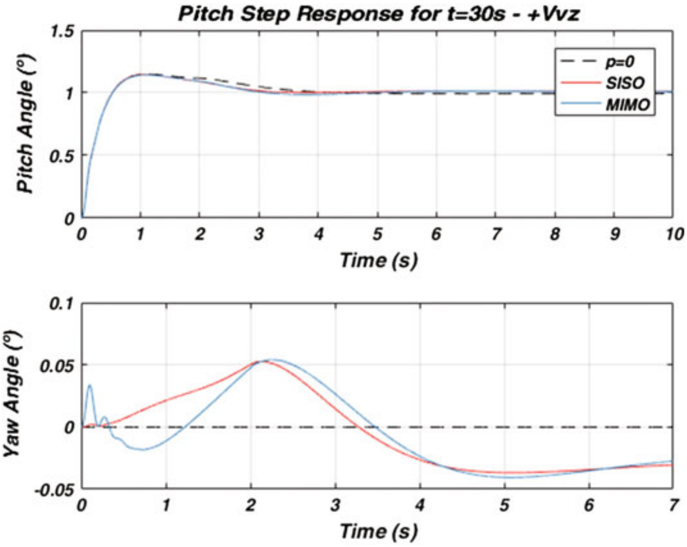


Fig. 11 Pitch step response for critical time (t = 30 s) under wind perturbation - Vvz

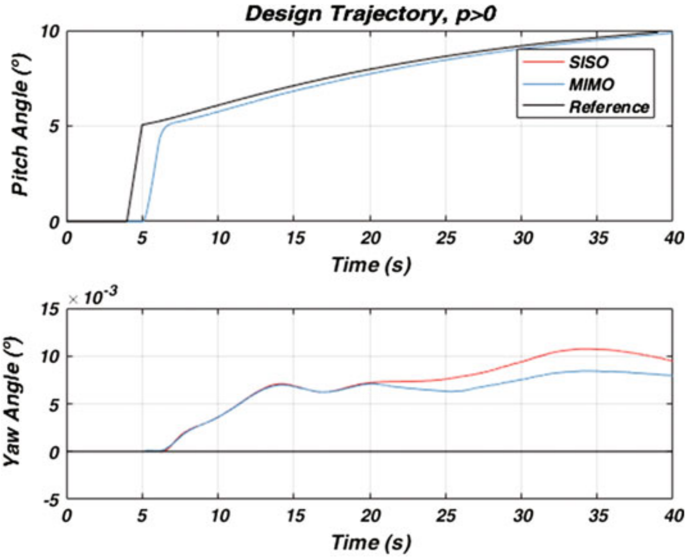


Fig. 12 Design trajectory - $G_\theta = -1$ and $G_\psi = +1 - p > 0$

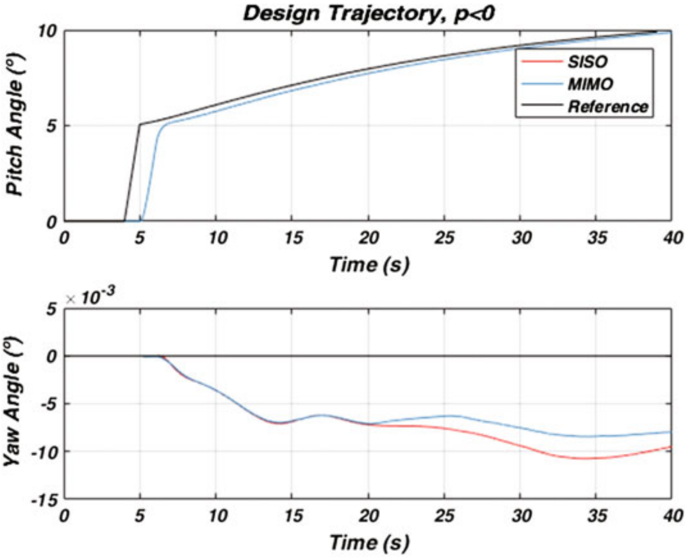


Fig. 13 Design trajectory - $G_\theta = -1$ and $G_\psi = +1 - p < 0$

Both controllers have a very similar response to the pitch reference signal, being almost interchangeable. The main difference appears on the yaw response.

While the reference signal for the yaw angle is zero, the SISO controller shows a drift under residual spin. This drift is mainly due to the integrator term, but that cannot be removed without affecting the step requirements unfavorably.

The MIMO controller is able to improve on this factor. In fact, for unitary roll gains, as a measure of the error on the yaw channel it is possible to see a reduction of 12.77% in the area under the yaw curve for the MIMO controller.

5.3.1 Influence of Roll Gain

An increase in the roll gains actually reduces the area under the curve. For instance, with $G_\theta = -2.5$ and $G_\psi = +2.5$, the reduction in area becomes 27.66% when compared to the SISO case.

Further analysis shows that, for these direct and cross path gains, the maximum value of roll gains for which the system is stable is 3.

This actually means the controller behaves well for this parabolic spin rate profile when the product between p and G is smaller than this value. Smaller values of G provide bigger leeway if there is uncertainty about the maximum roll, but the drift on the yaw angle will be larger.

Figure 14 shows the design trajectory for $G_\theta = -3.2$ and $G_\psi = +3.2$, when the system response is clearly unstable.

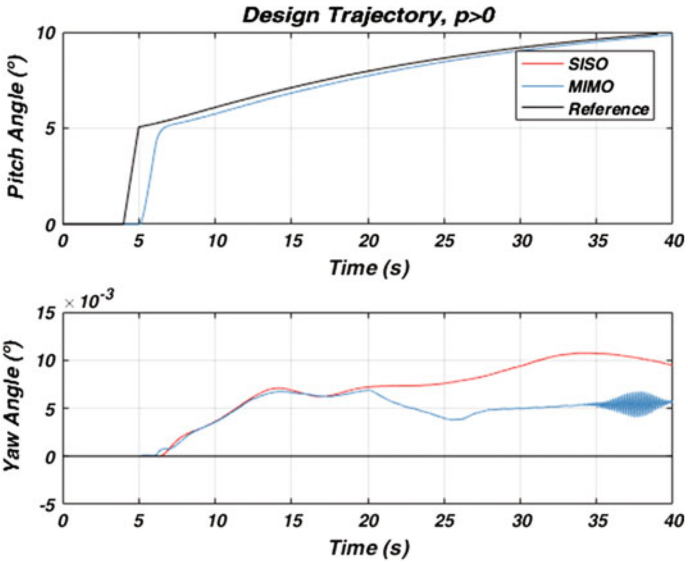


Fig. 14 Design trajectory - $G_\theta = -3.2$ and $G_\psi = +3.2$

This indicates that, while the SISO controller has worse performance as far as the yaw drift, it is stable for larger values of spin. If there is relative certainty about the maximum residual roll encountered and its profile, the MIMO strategy provides better results. However, if it is possible for the vehicle to encounter a very different residual roll when compared to the design roll profile, the SISO controller is more robust and, therefore, preferable.

Finally, even when stable, the MIMO system shows a more prominent oscillatory behavior than the one encountered with the SISO strategy. It is possible, though, that this occurs in consequence of the method used for gain calculation.

5.4 Design Trajectory Under Wind Perturbation

The influence of wind perturbation on the design trajectory was also analyzed. In this case, the wind profile used was based on [1] and varies with the vehicle's altitude.

Figure 15 shows the simulated behavior for a controller with unitary roll gain in the MIMO case, as compared to the previous analyzed SISO controller. Both controllers are stable, but the MIMO strategy presents a smaller drift on the yaw channel when compared to the SISO controller, in the same manner as the results of Sect. 5.3.

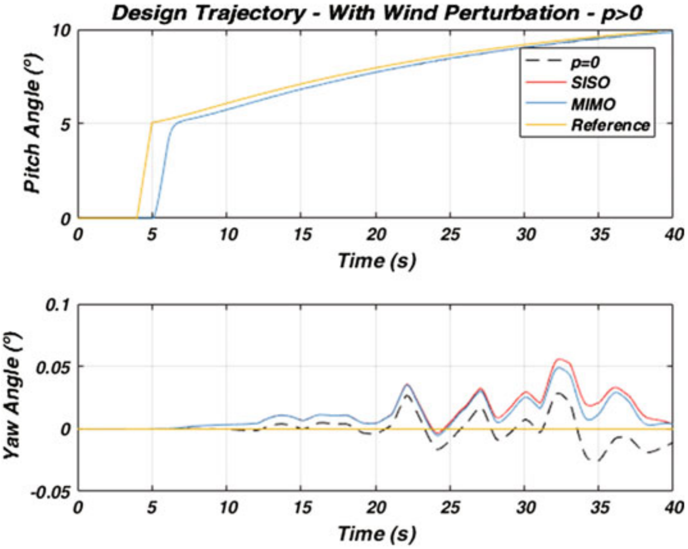


Fig. 15 Design trajectory under wind perturbation - $G_\theta = -1$ and $G_\psi = +1$

6 Conclusion

This paper studies how well a nonlinear MIMO controller with roll input is able to improve the performance of a linear SISO controller when applied to a spinning rocket with no roll control.

The performance metrics used indicate the MIMO strategy shows improvement with respect to the ability to follow the design trajectory while presenting similar step response. The controller's behavior was also favorable when under perturbation of a gust wind.

The MIMO controller, however, is more susceptible to eventual variations in the predicted residual spin, with its apparent robustness being related to the roll gain used on the cross path.

Finally, in this paper, the gains calculated for the MIMO controller were not designed taking robust control theory into account and, therefore, one can only guarantee its behavior with respect to the tuned flight conditions, and not its overall robustness.

Further studies should consider how a variable roll gain changes the MIMO controller performance, and different methods of gain calculation should be tested in an attempt to improve the step response in such a way that robustness is also ensured.

References

1. Adelfang, S. I. Simulation of wind-profile perturbations for launch-vehicle design. J Spacecr Rockets. https://doi.org/10.2514/1.3271
2. Apkarian P, Noll D (2006) Nonsmooth H-infinity Synthesis. IEEE Trans Autom Control 51:71–86
3. Apkarian P, Noll D (2007) Nonsmooth optimization for multiband frequency-domain control design. Automatica 43:724–731
4. Apkarian P, Gahinet P, Buhr C (2014) Multi-model, multi-objective tuning of fixed-structure controllers. Proc Eur Control Conf. https://doi.org/10.1109/ECC.2014.6862200
5. Apkarian P, Dao MN, Noll D (2015) Parametric robust structured control design. IEEE Trans Autom Control. https://doi.org/10.1109/TAC.2015.2396644
6. Garner D (1964) Control theory handbook. NASA Marshall Space Flight Center, NASA TM X-53036
7. Mohammadloo S, Alizadeh MH, Jafari M (2014) Multivariable autopilot design for sounding rockets using intelligent eigenstructure assignment technique. Proc Int J Control Autom Syst. https://doi.org/10.1007/s12555-012-0229-4
8. Pothukuchi RP et al (2016) Using multiple input, multiple output formal control to maximize resource efficiency in architectures. Proc Int Symp Comp Architect. https://doi.org/10.1109/ISCA.2016.63
9. Sun BC, Ryu JH, Tahk MJ (1996) Robust control design for spinning rocket. Proc Soc Instrum Control Eng Annu Conf. https://doi.org/10.1109/SICE.1996.865460
10. White J (1991) Attitude control of a spinning rocket via thrust vectoring. Proc Navig Control Conf. https://doi.org/10.2514/6.1991-2617

Comparison of Multiple Spacecraft Configuration Designs for Coordinated Flight Missions

Federico Fumenti and Stephan Theil

1 Introduction

In recent years the interest in spacecraft coordinated flight increased more and more due to the numerous potential advantages associated with the replacement of a single monolithic object with several smaller ones. A group of small spacecraft working together could enhance scientific observations, augment flexibility and redundancy, reduce costs and risks and overcome physical limitations. At the same time however, new challenges are introduced concerning for example the sharing of data, the communication and the relative motion among the objects. Focusing on this last aspect, it is trivial that when multiple objects are considered, careful attention must be paid to the way they move w.r.t. each other. Constraints may be applied to ensure a safe relative motion and according to their strictness the two branches of formation flight and cluster flight can be distinguished. Due to many technical limitations the spread of the two architectures did not evolve in the same way, up to the point that formations are being largely studied and successfully implemented, while clusters are nowadays still under investigation.

In a formation of satellites the relative configuration is fixed and control actions are required to maintain it. In a typical mission scenario several sensors and instruments demand for tight relative positions to cooperate, but cannot be allocated on the same spacecraft because the required relative distances are too wide and the resulting spacecraft exceeds the launch vehicle capabilities. To overcome this issue the devices can be distributed on different spacecraft which fly together while guaranteeing the satisfaction of the relative distances requirements through the use of control actions. Examples of missions implementing the formation flight concept can be found in TanDEM-X [20], PRISMA [21] and GRACE [26].

F. Fumenti (✉) · S. Theil
DLR, Institute of Space Systems, Robert Hooke Str. 7, 28359 Bremen, Germany
e-mail: federico.fumenti@dlr.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_32

In the case of a cluster there is no need for precise geometry, because the successful outcome of the mission does not strictly depend on specific relative configuration as in the formation flight case. As long as the distances among the spacecraft are held within a maximum and a minimum value to ensure inter-modules communication and avoid collisions, respectively, no control action is required. It all results in less strict relative motion requirements, with a consequent relaxed intervention from the control system. In a typical scenario each member of the cluster allocates a different functionality, like communications, data storage, power generation, etc. and all the functionalities are shared through wireless connections. Examples for potential application of the cluster flight can be identified in the missions PLEIADES [15] and SAMSON [13].

In the design of a coordinated spacecraft based-mission an extremely important task is the definition of the initial configuration, since the application of proper constraints on the relative states will induce a particular desired behaviour in the evolution of the relative motion. Over time corrective maneuvers are required to counteract the changes in the initial relative geometry deriving from the differential perturbations. Therefore, to limit the required fuel and the missions costs, it is highly desirable to have orbits that naturally satisfy the relative motion constraints. Over the years many authors have worked on the development and improvement of mathematical models to grasp the evolution of the relative motion and simplify the application of the required constraints (see e.g., [5, 6, 14, 16, 17, 25, 27, 29]).

Once the initial configuration is defined and the spacecraft are deployed, a station keeping approach could be used to cancel the drifts induced by the differential perturbations. Indeed the initial states ensure satisfaction of the distance constraints and can be seen as reference states to be tracked. This approach is certainly meaningful in a formation, where the relative geometry constraints limit considerably the tolerable differential drifts. Less clear is instead the benefit of cluster keeping through station keeping. In this case the loose constraints involve that the distance boundaries are infrequently violated and when this happens the drift from the reference state could be so large that the recomputation of a new reference state could become more meaningful than the station keeping of the old one. The goal of this work is to evaluate if and if so, by how much the station keeping logic could be beneficial for coordinated flight missions characterized by a different number of spacecraft and different distance boundaries. To study and implement the station keeping logic, it can be advantageous to express the relative motion through relative orbital elements, since the orbital elements have a slow variation over time and corrections of specific elements with theoretically no effect on the others can be obtained using impulsive control (see e.g., [3, 12, 18, 22–25]).

According to the type of constraints initially imposed, various initial configurations can be found, differing from each other by the number of deployable spacecraft, their relative geometry and the effort, in terms of ΔV, they require for station keeping. A survey and comparison of such initial configurations is the topic of this work, which is organized as follows. Section 2 introduces the problem and the techniques used to define the initial states of a group of spacecraft. Section 3 describes how the comparison has been set and the key parameters used, and finally presents the results of the study. Section 4 reports the final conclusions.

2 Problem Statement

The successful outcome of a coordinated flight-based mission strictly depends on the relative motion among the involved spacecraft and to ensure satisfaction of relative motion constraints, much attention must be put on the control actions and on the counteraction of the differential perturbations. To reduce the fuel expenditure specific constraints can be imposed in the definition of the initial conditions, according to the desired behavior, which is dictated by the mission goals.

From the literature research it emerged that several techniques used to define the relative initial conditions of a group of satellites are available, hence it has been decided to examine and compare them to determine if and how the computed initial conditions could be used for the deployment of a cluster of objects.

The techniques for the initial conditions (TIC) that have been studied are introduced in the upcoming sections but at first, in order to make their comparison meaningful, a common test setup is identified: it is supposed that the cluster is centered on a virtual point (VP) and it is assumed that minimum distance constraints (MinDC) as well as maximum distance constraints (MaxDC) are in place. In particular, to prevent collisions a minimum distance D_{min} must be guaranteed between any pair of spacecraft, while escaping drifts are avoided through upper bounding the distance of a spacecraft from the VP by a maximum value D_{max}. Denoting with d_s and d_r the generic distances between any two spacecraft of the cluster and between a spacecraft and the VP, respectively, the MinDC and the MaxDC are given by:

$$d_s > D_{min} \tag{1a}$$

$$d_r < D_{max}. \tag{1b}$$

The VP is supposed to move on a low Earth orbit (LEO) and its initial state is defined in Table 1 in terms of osculating keplerian elements $\mathbf{œ}^{K}_{VP}(t_0) = (a\ e\ i\ \omega\ \Omega\ M)^T$.

The description of each technique is structured in two parts. At first the basic logic is introduced while using the simple chief-deputy framework. The chief is the VP and the state of the deputy is defined such that their relative distance is bounded by D_{min} and D_{max}. Afterwards, it is shown how the same logic can be adapted to configure a cluster of n_m modules while also maximizing n_m. As a matter of fact for the case of a cluster the simple chief-deputy approach does not fit very well as it is, and for several reasons. In the first place, from the stated assumptions it is clear that there is no need to lower bound the distances d_r and to upper bound the distances d_s. Secondly, when several modules are placed into the cluster and the chief-deputy technique is applied to every module, information about the motion of the spacecraft w.r.t. the VP is available, but nothing can be said about a deputy-deputy type of

Table 1 Initial state of the virtual point

Element	Value	Units
Semi-major axis - a	7000.92	km
Eccentricity - e	0.01	
Inclination - i	50.99	deg
Right ascension of the ascending node - $\varOmega$	11.48	deg
Argument of perigee - ω	19.12	deg
Mean anomaly - M	21.00	deg

motion. Plus, the idea of studying the motion of each agent with respect to all the others is inconceivable, since as n_m grows the problem quickly becomes extremely complex and unmanageable.

In order to clarify the descriptions of the investigated techniques in the upcoming sections, it wants to be revised here the distinction between keplerian and non-singular orbital elements, since both of them will be used. The keplerian set of elements $\boldsymbol{œ}^K$ has been actually already introduced through Table 1, while the non-singular set is given by $\boldsymbol{œ}^N = (a\ u\ e_x\ e_y\ i\ \varOmega)^T$, where $e_x = e\cos\omega$ and $e_y = e\sin\omega$ are the components of the eccentricity vector $\mathbf{E} = (e_x\ e_y)^T$ and $u = \omega + M$ is the mean argument of latitude. In addition, it is also worth recalling that when two spacecraft are considered and their states are expressed in terms of orbital elements, the relative motion of the deputy D w.r.t. the chief C can be expressed in terms of relative orbital elements. In the keplerian case, given the vectors $\boldsymbol{œ}^K_C$ and $\boldsymbol{œ}^K_D$, the relative elements are simply computed as the difference between the elements of the two objects:

$$\Delta\boldsymbol{œ}^K = \boldsymbol{œ}^K_D - \boldsymbol{œ}^K_C = (\Delta a\ \Delta e\ \Delta i\ \Delta\varOmega\ \Delta\omega\ \Delta M)^T. \tag{2}$$

In the non-singular case the vectors $\boldsymbol{œ}^N_C$ and $\boldsymbol{œ}^N_D$ lead to the relative elements through a nonlinear combination:

$$\begin{aligned}\Delta\boldsymbol{œ}^N &= \begin{pmatrix}\Delta\widetilde{a} & \Delta\lambda & \Delta e_x & \Delta e_y & \Delta i_x & \Delta i_y\end{pmatrix}^T \\ &= \begin{pmatrix}\Delta a/a & \Delta u + \Delta\varOmega\cos i & \Delta e_x & \Delta e_y & \Delta i & \Delta\varOmega\sin i\end{pmatrix}^T\end{aligned} \tag{3}$$

where $\Delta\widetilde{a}$ is an additional measure of the differential semi-major axis, $\Delta\lambda$ is the differential mean longitude, Δe_x and Δe_y are the components of the relative eccentricity vector $\Delta\mathbf{E} = (\Delta e_x\ \Delta e_y)^T$, and finally Δi_x and Δi_y are the components of the relative inclination vector $\Delta\mathbf{I} = (\Delta i_x\ \Delta i_y)^T$.

Eccentricity/Inclination Vector Separation Technique

The Eccentricity/Inclination (E/I) vector separation technique (EIVS) is particularly attractive since it can naturally enforce collision avoidance. It has been originally proposed to face the problem of satellites colocation in geostationary slots [11] and in the last years widely investigated and successfully applied also for formations of satellites in LEO [7, 8, 19].

Through the use of a rotating reference frame $C_{r\theta h}$ which has its origin coincident with the chief spacecraft, the $r\theta$-plane lying on its orbital plane and the r-axis parallel to its position vector (positive outwards), the relative motion can be expressed in non-dimensional form through:

$$\begin{aligned} \delta x(u) &\approx \varDelta\widetilde{a} && - \varDelta E\cos(u-\vartheta) && \text{(4a)} \\ \delta y(u) &\approx -\frac{3}{2}\varDelta\widetilde{a}u + \varDelta\lambda && + 2\varDelta E\sin(u-\vartheta) && \text{(4b)} \\ \delta z(u) &\approx && + \varDelta I\sin(u-\varphi) && \text{(4c)} \end{aligned}$$

where $\varDelta\mathbf{E}$ and $\varDelta\mathbf{I}$ are expressed in polar notation, with $\varDelta E = \|\varDelta\mathbf{E}\|$ and $\varDelta I = \|\varDelta\mathbf{I}\|$, and with ϑ and φ being the relative perigee and the relative ascending node [7].

Assuming $\varDelta\lambda = 0$ and $\varDelta a = 0$ to cancel the offsets and to prevent the drift in the along track direction, the collision risks can be reduced by setting:

$$\begin{aligned} \vartheta &= \varphi + k\pi && \text{(5a)} \\ D_{min} &\leq a\min\{\varDelta E, \varDelta I\} && \text{(5b)} \end{aligned}$$

with integer k, while the constraint

$$D_{max} \geq a\sqrt{4\varDelta E^2 + \varDelta I^2} \qquad (6)$$

ensures satisfaction of the MaxDC. If the J_2 perturbation is included in the model Eq. (4), $\varDelta a = 0$ is not valid anymore and Eqs. (5) and (6) need to be adapted [7].

To better relate the E/I relative vectors with the distances, it could be useful to consider their dimensional version obtained by multiplying them with the semi-major axis of the VP. The new dimensional parameters can be distinguished from the original ones by the presence of a small hat ($\widehat{\ }$), so that to $\varDelta E$ corresponds a $\varDelta\widehat{E} = a\varDelta E$, to $\varDelta I$ corresponds a $\varDelta\widehat{I} = a\varDelta I$, and so on. Then, in the design phase, it can be helpful to define the elements of the spacecraft in the planes $\varDelta\widehat{e}_x\varDelta\widehat{e}_y$ and $\varDelta\widehat{i}_x\varDelta\widehat{i}_y$.

Let us now see how the EIVS approach can be used to configure a cluster with multiple objects. The problem can be geometrically approached in two steps:

1. In each of the two planes $\varDelta\widehat{e}_x\varDelta\widehat{e}_y$ and $\varDelta\widehat{i}_x\varDelta\widehat{i}_y$ the D_{max} is used to identify a region around the origin, which includes points satisfying the MaxDC;
2. In each region points are chosen with a mutual distance at least equal to D_{min}.

An example of how the described geometric logic can be applied is given in Fig. 1, where each point represents the relative eccentricity (left plot) and the relative inclination (right plot) vector of a spacecraft w.r.t. the VP, which is highlighted as a small circle at the origin of the planes.

Provided that in each plane each point is sufficiently spaced from all the others and that corresponding points from the two planes satisfy the (anti-)parallelism condition (Eq. (5a)), the MinDC can be ensured. In the given example the MinDC are based on a value $D_{min} = 5$ km and are represented by the small circles, inasmuch as they picture the forbidden regions around the spacecraft and, as it can be seen, are not overlapping. The fulfillment of the MaxDC can instead be studied by taking into consideration the distance of the points from the origin of the planes. These constraints are based on a value $D_{max} = 30$ km and are represented by the gray regions, which can be evaluated through Eq. (4). It can be noted that in Fig. 1 corresponding points from the two planes, not only share the same phase of the relative vectors, but have also the same magnitude. This is the reason why in the right plot of Fig. 1 the points are concentrated in the center leaving the outer part of the gray region empty. The condition of equal magnitudes is not required by the EIVS technique, but it has been introduced to limit the differential perturbations experienced by the spacecraft.

The given example shows how the relative E/I vectors can be selected while using simple 2-D geometry. Six points have been chosen from the gray regions but this number can easily grow if a smarter selection is performed. The maximization of n_m therefore becomes a packing problem, since it turns into the research of the maximum number of points that can be placed into a given 2-D region. In this study the regular grid shown in Fig. 2 has been used, where each point is at the center of a regular hexagon and is surrounded by six points located at the vertices of the hexagon.

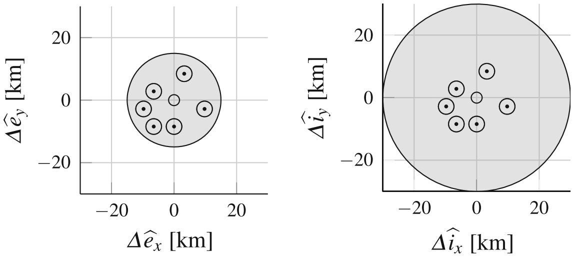


Fig. 1 Example of relative eccentricity (*left*) and inclination (*right*) vectors for a cluster of 6 spacecraft

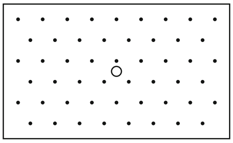


Fig. 2 Regular grid for the packing problem of the EIVS technique

J_2 Invariance Technique

The technique of the J_2 invariance (J2In) consists in placing the spacecraft in orbits for which the relative drift is minimized. When the J_2 perturbation is included in the motion of a spacecraft, its osculating elements experience oscillations and a secular growth. The oscillations are usually considered harmless, while the dangerous effect is the one associated with the secular growth, since it can produce a drift of the orbits. For this reason it can be useful focusing on mean elements[1] $\overline{\boldsymbol{\oe}}^K$, which are free from the oscillations and are only subject to the secular growth.

When the mean elements are used, it must be noted that the J_2 term only alters the elements $[\varOmega \; \omega \; M]$, which experience time drifts modeled as [2]:

$$\frac{d\varOmega}{dt} = -\frac{3}{2} J_2 n \left(\frac{R_E}{p}\right)^2 \cos i \tag{7a}$$

$$\frac{d\omega}{dt} = -\frac{3}{4} J_2 n \left(\frac{R_E}{p}\right)^2 \left(5\cos^2 i - 1\right) \tag{7b}$$

$$\frac{dM}{dt} = n + \frac{3}{4} J_2 n \left(\frac{R_E}{p}\right)^2 \sqrt{1-e^2}\left(3\cos^2 - 1\right). \tag{7c}$$

where $n = \sqrt{\mu/a^3}$ and $p = a\sqrt{1-e^2}$ are the mean motion and the semilatus rectum of the VP, while μ and R_E are the gravitational parameter and the radius of the Earth.

The concept of the J_2 invariance is based on the fact that if the first three elements $[a \; e \; i]$ of the different spacecraft are properly selected, it is possible to make them having the same rates of variation for the last three elements $[\varOmega \; \omega \; M]$. Indeed this approach aims at canceling the relative rates, so that the secular drift can be prevented. Nevertheless, it is very uncommon to match all the three rates, because in this way the constraints would restrict considerably the possible relative orbits. It is instead usually preferred to combine the rate of ω and M so that only the rates of $\dot{\varOmega}$ and of $\dot{\omega} + \dot{M}$ need to be matched.

To perform the matching process, the constraints are expressed in terms of the deputy relative elements, hence the following expressions can be retrieved [24]:

$$f_a = \eta \varDelta a + 2 D a e \varDelta e = 0 \tag{8a}$$

$$f_i = \eta^2 \tan i \, \varDelta i - 4 e \varDelta e = 0 \tag{8b}$$

where the absolute and differential elements used are the ones of the chief and of the deputy, respectively. In addition $\eta = \sqrt{1-e^2}$, $D = \frac{J_2}{4L^4\eta^5}(4+3\eta)(1+5\cos^2 i)$ and $L = \sqrt{a/R_e}$ hold. Note that Eq. (8) allows for computation of $\varDelta a$ and $\varDelta i$ once that $\varDelta e$ is fixed, but the cases where $\varDelta a$ or $\varDelta i$ are given also remain valid. Finally,

[1] Mean elements are usually identified by the presence of an overbar ($^-$), but since the J2In technique does not use osculating elements, to relieve the notation the overbar in this section is dropped.

in concerns to the remaining differential elements $[\Delta\Omega\ \Delta\omega\ \Delta M]$, they can be freely chosen as long as they satisfy the MinDC and the MaxDC conditions.

Let us now approach the problem for the cluster case. If multiple objects are involved in the process, the J_2 invariance should be ensured for all of them, meaning that all the relative orbits should be J_2 invariant w.r.t. each other. In this perspective Eq. (8) should be applied for each pair of elements (sat/VP and sat/sat), thus transforming them into a set of $2n_p$ conditions, with $n_p = 0.5n_m(n_m - 1)$ denoting the number of the pairs. As soon as $n_m > 2$ the system becomes over-determined and needs then to be solved numerically, and so it becomes useful recasting Eq. (8) into the form

$$f_a < \varepsilon \tag{9a}$$

$$f_i < \varepsilon \tag{9b}$$

where ε is a user-defined threshold within which the solution must satisfy the constraints. According to the approximations used in the derivation of Eq. (8) (only terms of $\mathscr{O}(J_2)$ have been retained) [24], in the performed study is has been assumed $\varepsilon = 10^{-3}$ in order to have the same order of magnitude of the J_2 coefficient.

Another step towards the definition of a cluster configuration consists in identifying the range of values from which the parameters Δa, Δe and Δi can be chosen. Given two objects A and B and assuming that their relative orbits are J_2 invariant, it should be implied that the relative orbit of A is invariant w.r.t. B and viceversa. It has been seen instead, that if the differential elements $[\Delta a\ \Delta e\ \Delta i]$ are too large, the above-mentioned *double invariance* can be missing. There are cases indeed where the elements of B are selected to ensure that its relative orbit w.r.t A is J_2 invariant, but then inverting the roles and treating B as chief and A as deputy, the invariance constraints are violated. To avoid this situation, the ranges of the differential elements able to ensure the desired *double invariance* have been researched. According to the eccentricity of the VP, the attention has been posed on an interval $-0.009 < \Delta e < 0.039$, which has then been divided into a regular grid. Next, for all the grid points the double invariance has been pairwise checked and the results are depicted in Fig. 3. The gray area emphasizes the satisfaction of the double invariance, which for example is fulfilled by two objects having orbital elements defined with differential eccentricities equal to 2.7×10^{-2} and -0.3×10^{-2}, but not for those two with orbital elements obtained from the values 3.9×10^{-2} and 1.5×10^{-2}. Please note that the plot is symmetric w.r.t. the plane bisector and therefore for both the given examples, the results do not depend on which of the two values is assigned to which of the two objects. Finally the black-sided square highlights the wanted range for the differential eccentricity, so that if all the orbits of the cluster satisfy

$$\begin{array}{rcl} -0.023\,\text{km} & < \Delta a < & 0.048\,\text{km} \\ -0.007 & < \Delta e < & 0.033 \\ -0.00025\,\text{rad} & < \Delta i < & 0.00120\,\text{rad} \end{array} \tag{10}$$

all the relative orbits are J_2 invariant w.r.t. each other.

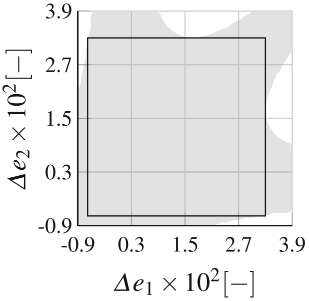


Fig. 3 Map of the J_2 double invariance shown in terms of relative eccentricities

To investigate the cluster case, the definition of the initial configuration can be expressed as the determination of those vectors $\Delta\boldsymbol{\text{œ}}_i^K$ (with $i = 1, 2, \ldots, n_m$), whose elements are bounded by the conditions given in Eq. (10) and also fulfill a system of nonlinear conditions given by the $2n_p$ equations obtainable from Eq. (8) and by additional equations deriving from the application of Eq. (1). To research the solution the solvers FMINCON from MATLAB and SNOPT have been used.

In the way the problem has been implemented, it is trivial that n_m is not a variable, but a parameter provided by the user. Therefore, to maximize the number of spacecraft that the cluster can allocate, it has been decided to proceed by solving the problem with different increasing values of n_m. When a solver cannot find a solution anymore, the cluster is saturated and the maximum value for n_m has been found.

Distance Bounded Natural Orbits Technique

The technique of the distance-bounded natural orbits (DBNO), as suggested by the name itself, aims at finding orbits that satisfy naturally the MinDC and the MaxDC. The technique is presented in Mazal and Gurfil [17] and relies on a constraint which is proven to ensure bounded relative distances when a time invariance assumption for the environmental perturbations is made. This idea is similar to what is done with the J2In approach, but this time the invariance can take all perturbations deriving from the gravitational potential into account.

Given two objects A and B with equal ballistic coefficients, if it results

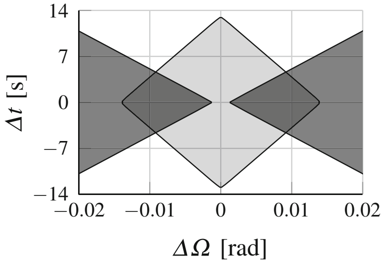
$$\text{œ}_B(t_0) = \text{œ}_A(t_0) + \int_{t_0}^{t_0+\Delta t} \dot{\text{œ}}_A dt + [0\;0\;0\;\Delta\Omega\;0\;0]^T \tag{11}$$

then it also holds

$$2\eta_{min} \sin\left(\frac{|\Delta\Omega|}{2}\right) - V_{max}|\Delta t| \leq d_s(t) \leq 2\eta_{max} \sin\left(\frac{|\Delta\Omega|}{2}\right) + V_{max}|\Delta t| \tag{12}$$

where V_{max} denotes the maximum speed of A, while η_{max} and η_{min} denote its maximum and minimum equatorial projections, given by $\eta_A(t) = \sqrt{x_A^2(t) + y_A^2(t)}$.

Fig. 4 Example of Δt and $\Delta\Omega$ selection while satisfying the MinDC (▆) and the MaxDC (□)



For a given set of constraints on the minimum and maximum distances D_{min} and D_{max}, the values Δt and $\Delta\Omega$ can be chosen while satisfying

$$2\eta_{min}\sin\left(\frac{|\Delta\Omega|}{2}\right) - V_{max}|\Delta t| \geq D_{min} \tag{13a}$$

$$2\eta_{max}\sin\left(\frac{|\Delta\Omega|}{2}\right) + V_{max}|\Delta t| \leq D_{max} \tag{13b}$$

Equation (13) can be observed graphically in Fig. 4, showing for which values of $\Delta\Omega$ and Δt the MinDC and the MaxDC are satisfied, when $D_{min} = 5$ km and $D_{max} = 100$ km are considered. The satisfaction of each constraint is highlighted with a different color, therefore for the mission design a pair $[\Delta\Omega\ \ \Delta t]$ must be selected from the overlapping area.

In the case of n_m spacecraft, the DBNO technique retains validity as long as the constraints expressed through Eq. (13) are applied to each pair. The differences in time and RAAN of the i-th spacecraft w.r.t. the VP can be denoted as $\Delta\Omega_{0i}$ and Δt_{0i}, while the differences between any pair of spacecraft (i, j) can be denoted as

$$\begin{cases} \Delta\Omega_{ij} = \Delta\Omega_{0j} - \Delta\Omega_{0i} \\ \Delta t_{ij} = \Delta t_{0j} - \Delta t_{0i} \end{cases} \tag{14}$$

with $j = 1, 2, \ldots, n_m$ and $j \neq i$.

In addition, the small angles approximation can be introduced to replace the sine functions with their arguments. Indeed in this work several D_{max} up to few hundreds of km will be considered, leading to a maximum differential RAAN of few degrees totally compatible with this approximation. Equation (13) then turns into:

$$\eta_{min}|\Delta\Omega_{ij}| - V_{max}|\Delta t_{ij}| \geq D_{min} \tag{15a}$$

$$\eta_{max}|\Delta\Omega_{0i}| + V_{max}|\Delta t_{0i}| \leq D_{max}. \tag{15b}$$

Concerning Eq. (15b) and the MaxDC, it should be noted that the hypothesis of equal ballistic coefficients is missing in this work, since the VP is not a real spacecraft and cannot be affected by the atmospheric drag. This means that a slow drift of the spacecraft w.r.t. the virtual point should be expected.

When approaching the problem while trying to maximize the number of modules n_m, Eq. (15b) is immediately used to define the domain of the differential time and RAAN from which the sets $[\Delta\Omega_{0i} \;\; \Delta t_{0i}]$ should be selected. On the other hand Eq. (15a) is exploited for the actual selection of the sets $[\Delta\Omega_{0i} \;\; \Delta t_{0i}]$. From a first glimpse at Eq. (15a) and with the help of Fig. 4, it can be seen that it is never possible to satisfy the MinDC with a pure time shift and that a minimum differential RAAN $|\Delta\Omega|_{min}$ is always required. Indeed exploiting Eq. (15a) and the fact that $|\Delta t_{ij}| > 0$, the minimum value $|\Delta\Omega|_{min}$ can be retrieved

$$|\Delta\Omega_{ij}| > \frac{D_{min}}{\eta_{min}} \equiv |\Delta\Omega|_{min} \tag{16}$$

and the differential RAAN can be chosen according to $|\Delta\Omega_{ij}| = (1 + k_\Omega)|\Delta\Omega|_{min}$ with a small $k_\Omega > 0$. In turn an upper bound for the time shift can be defined as

$$|\Delta t_{ij}| < \frac{k_\Omega D_{min}}{V_{max}} \equiv |\Delta t|_{max} \tag{17}$$

and similarly to what has been set for the differential RAAN, the differential times can be chosen according to $|\Delta t_{ij}| = (1 - k_t)|\Delta t|_{max}$ with a small $k_t > 0$. It is worth noticing that the determination of the two boundaries $|\Delta\Omega|_{min}$ and $|\Delta t|_{max}$ and the selection of the two coefficients k_Ω and k_t do not depend on the specific pair (i, j), therefore it is possible to define the values of the four parameters just once and use them to find the initial condition of all the spacecraft.

At this point, it is clear that dividing the $\Delta\Omega$ domain into a grid of points equally spaced by $(1 + k_\Omega)|\Delta\Omega|_{min}$ allows finding the differential RAAN $\Delta\Omega_{0i}$ and maximizing the number of spacecraft n_m at the same time. Concerning the time shifts, a trivial solution can be obtained assuming $\Delta t_{0i} = 0 \;\forall i$. Otherwise if time shifts different from zero are desired they can be selected through the knowledge of the $|\Delta t|_{max}$.

Delayed Elements Technique

The technique of the delayed elements (DeEl) is very similar to the DBNO technique, in as much as it aims at finding relative orbits which are invariant w.r.t. the perturbations deriving from the full gravitational potential. The main difference consists in the fact that the elements of the spacecraft in this case are obtained only through the use of the time difference Δt, which means that all the members of the cluster pass through the same positions of the virtual point, but they do it with a certain time difference Δt [9]. In this way all the spacecraft experience the same perturbations with minimal variations from the gravitational field with the consequence that maneuvers to counteract differential perturbations are greatly reduced.

The cluster obtainable with this technique can be imagined as a train of spacecraft separated in the along track direction, just like pearls on a string. Reference to separation distance d or time difference Δt is equivalent, since these two quantities can be easily related by exploiting the knowledge of the mean motion. Indeed given a spacecraft with mean motion n moving for a time $\varDelta t$, the distance d between the initial and the final positions can be approximated with the traveled arc of trajectory $\overset{\frown}{d}$, which exploiting Kepler's second law can be expressed as

$$\overset{\frown}{d} = \frac{abn\varDelta t}{r} \tag{18}$$

where b is the semi-minor axis of the orbit and r is the position vector magnitude. It is worth noticing that for an elliptical orbit r changes with time and according to the location of the spacecraft along the orbit a different $\overset{\frown}{d}$ can correspond to the same $\varDelta t$. For this reason once that the maximum and the minimum values of r are computed, the minimum and the maximum distances corresponding to the given $\varDelta t$ are also known. From a different point of view, this means also that when the D_{min} and D_{max} values are given, the evaluation of the required time difference $\varDelta t$ is quite straightforward. And this is exactly how the initial conditions of two spacecraft can be defined, since in this case the D_{min} and D_{max} are assumed to be known and can be used to retrieve a range of values from which the $\varDelta t$ should be picked to satisfy the distance constraints. Once that the $\varDelta t$ is chosen, the initial state of a spacecraft B can be computed from that of a spacecraft A through Eq. (11), assuming $\varDelta\varOmega = 0$.

In a similar way, the configuration of an entire cluster can be approached, with the core of the process consisting in the identification of the time shifts associated with the different objects of the cluster. From this perspective, following the example of the DBNO technique, it can be useful to distinguish between the time shifts of the spacecraft w.r.t. the VP $\varDelta t_{0i}$ and the time shifts among the spacecraft

$$\varDelta t_{ij} = \varDelta t_{0j} - \varDelta t_{0i} \tag{19}$$

with $i, j = 1, 2, \ldots, n_m$ and $i \neq j$. The constraints on the distances only appear indirectly, in as much as they are used to define an upper and a lower bound for the values of the time shifts $\varDelta t_{0i}$, which need to be researched while satisfying Eq. (19). In particular, the MaxDC defines the maximum allowed time shift $|\varDelta t|_{max}$ of a spacecraft w.r.t. the VP, meaning that it should result in $\frac{\varDelta t_{0i}}{|\varDelta t|_{max}} \in [-1 \;\; 1]$. On the other hand the MinDC is taken into account computing the minimum time shift $|\varDelta t|_{min}$ corresponding to the minimum distance D_{min} and ensuring that $|\varDelta t_{ij}| > |\varDelta t|_{min} \; \forall (i, j)$.

Finally, when the interval $[-|\varDelta t|_{max} \;\; |\varDelta t|_{max}]$ is divided into sub-intervals of length $|\varDelta t|_{min}$, the problem of maximizing the number of spacecraft n_m is also addressed, with the nodes of the grid denoting the time shifts $\varDelta t_{0i}$.

3 Comparison of the Techniques

To analyze the behaviour of the different techniques and perform their comparison, different points of view have been considered focusing on the number of spacecraft populating the cluster and on the ΔV they require for the corrective maneuvers.

The maximum number of points n_m that can be packed into a given volume of space around the VP clearly depends on the constraints MinDC and MaxDC, since for a given value of D_{min} an increase of D_{max} involves a larger volume around the VP with the opportunity to allocate more points in it, while for a given value of D_{max} an increase of D_{min} involves a larger safety distance between any pair of points with a consequent reduction of their total number for that same volume. This makes then n_m dependent on three different aspects of the problem: the TIC, the D_{min} and the D_{max}. To observe the effect of the distances, the constraints given in Eq. (1a) and (1b) have been applied with several values of D_{min} and D_{max}. For the sake of brevity, in the following a specific set of values $[D_{min}\ D_{max}]$ might also be denoted simply as $D_R = D_{min}/D_{max}$ so that, for example, the notation $D_R = 1/10$ indicates that the cluster needs to satisfy a $D_{min} = 1$ km and a $D_{max} = 10$ km.

Concerning the ΔV, it is clear that due to the differential perturbations experienced, the spacecraft will naturally drift over time and even if the cluster is initialized in such a way to satisfy the distance constraints, sooner or later the relative configuration might become unsafe. The safety of the cluster can be ensured through proper corrective maneuvers, hence the required ΔV can be used as a second key parameter.

3.1 Slots Versus Spacecraft

This section it intended to shortly explain that a distinction between the slots occupied by the spacecraft and the spacecraft themselves has been made.

The slots are those reference locations in which the spacecraft are deployed and that should be tracked by them. They are the entities directly involved in the search process and are treated as mass-less points. The ultimate goal is to find locations that satisfy the distance constraints indefinitely, so that when the spacecraft are deployed in them and track them through station keeping maneuvers, one can be sure that the distance constraints will not be violated. Due to this distinction, in the early phase of the study, for each set [TIC D_{min} D_{max}], the TIC is used to identify slots that ensure satisfaction of the MinDC and the MaxDC. Then, in a second moment the slots are filled with spacecraft and their motion is studied.

3.2 Number of Slots

The identification of the slots consists in defining the initial reference vectors while taking also their natural evolution into account. A set of states satisfying the constraints at the initial time t_0 and violating them at a future time $t > t_0$ would be as useless as one violating them at t_0. An ideal set never violates the constraints, but this is not a realistic case because of the differential perturbations, which bring the vectors to slowly drift towards an unsafe configuration. The drifts grow indeed over time involving an increase in the number of violations. What can be done is to treat the configuration returned by the TIC as a candidate solution, propagate it for a certain timeframe and discard the state vectors violating the distance constraints.

For each set [TIC D_{min} D_{max}] a certain number of candidate slots $n_{s,c}$ is obtained, with the initial state of each slot defined as:

$$\boldsymbol{\oe}_s(t_0) = f(\boldsymbol{\oe}_{VP}(t_0), \Delta_s) \tag{20}$$

where $s = 1, 2, \ldots, n_{s,c}$ and the function f specifies that the initial state of the s_{th} slot depends on the initial state of the virtual point and on some *differences* Δ_s applied to it. In Eq. (20) a generic function f has been used to take into account that those *differences* change from one technique to another. They can be the differential time Δt of the DeEl technique, or the differential mean elements of the J2In technique, and so on.

Once that the states of the candidate slots are available, they are converted in Cartesian coordinates and propagated for five orbits including the J_2 effect. The choice for such a short time frame is due to the fact that some techniques (EIVS, J2In) identify the initial configuration by using simplified models of the relative motion, which become less and less reliable over time. The decision to include the J_2 perturbation is instead motivated considering that its effect is the largest one in a LEO. A spacecraft would also be significantly perturbed by its interaction with the atmosphere, but in this first phase the main characters of the study are the slots, which can be treated as mass-less points free from the effect of the drag.

Once that the Cartesian propagation is completed the relative distances are recovered and checked. At first the check of the maximum distance from the virtual point is considered and the slots violating it are directly discarded from the solution. Afterwards the attention shifts towards the MinDC, whose check takes advantage of the graph theory ([1, 10]). It is trivial to state that a slot violating the MaxDC cannot be part of the solution, but if the slots (x, y) violate the minimum distance there is no need to remove them both, rather is it enough to discard only one of them and two solutions may be used, i.e. one including x without y and another including y without x. However, as the number of MinDC violations increases, it becomes more and more difficult to identify the slots to remove; hence the graph theory might come in handy. In particular, a graph can be constructed from all the pairs of slots satisfying the minimum distance constraints, so that the identification of the most populated cluster can be set as a maximum clique problem MCP ([4, 28]).

For the sake of clarity let us consider a practical example with five candidate slots all satisfying the MaxDC. Suppose instead that a violation of the minimum distance is recorded for the pairs of slots (1, 2), (1, 5) and (2, 3). Solving the MCP for the graph associated with the problem, three maximum cliques can be found:

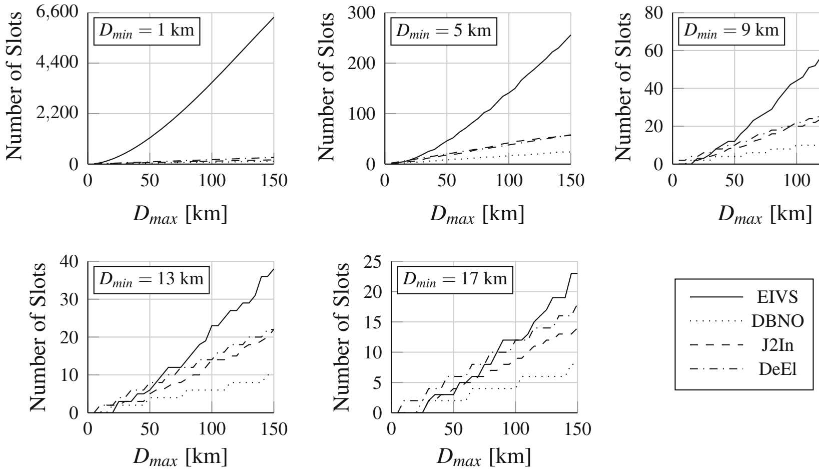


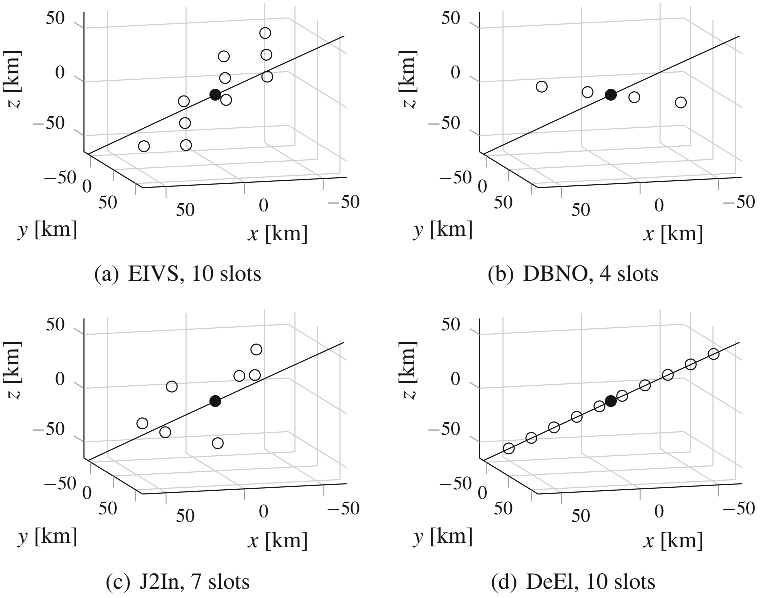
Fig. 5 Number of slots as a function of D_{max} for the different techniques

$$C_1 = \{1\ 3\ 4\} \qquad C_2 = \{2\ 4\ 5\} \qquad C_3 = \{3\ 4\ 5\}$$

This means that given the five candidate slots, the largest safe cluster cannot include more than three of them, and moreover not every three-element combination of the five candidates is acceptable, rather only the combinations C_1, C_2 and C_3.

The combinations of slots obtained from the MCP resolution represent then solutions of the initial configuration problem and once that they are available the search can be considered concluded. Figure 5 shows how the number of slots varies with the D_{max} for all the techniques examined and for D_{min} fixed at 1, 5, 9, 13 and 17 km. As expected, for every TIC, the number of returned slots increases when increasing D_{max} once the D_{min} is given and decreases when increasing D_{min} once the D_{max} is given. The less populated clusters are the ones obtained from the DBNO, whose trend is always the lowest. A slight increase in the slots number is achieved with the J2In and DeEl, which behave similarly, with trends very close to each other. Finally the EIVS is the technique returning the largest number of slots in most of the cases, with the separation between its and the rest of the trends getting larger when reducing the minimum distance boundary. This behaviour depends on the fact that the constraints required by the EIVS are less restrictive than those imposed by the other techniques, with the consequence that a larger number of vectors satisfying them can be found.

Graphically Fig. 6 can also help, giving an overview of the arrangement associated with the different techniques. Each subfigure is associated with a TIC and shows the initial configuration of slots around the VP when the distance boundaries are given by $D_R = 17/85$. The representation is given in an inertial frame translated to shift its origin in the virtual point, which is then located at the center of the subfigures and colored in black, just like the solid lines representing its orbit. As it can be seen, the slots from EIVS are densely arranged on a plane, while the slots from the DBNO and DeEl are arranged on a line and the slots from the J2In are sparsely distributed in a 3-D region.



(a) EIVS, 10 slots

(b) DBNO, 4 slots

(c) J2In, 7 slots

(d) DeEl, 10 slots

Fig. 6 Slots arrangements returned by all techniques for $D_{min} = 17\,\text{km}$ and $D_{max} = 85\,\text{km}$

3.3 ΔV Budget

Once that the study of the slots had been completed, the attention has been shifted towards the behaviour of the spacecraft, with the final goal of evaluating the ΔV budget required to maintain the distance constraints satisfied and the cluster safe. As a matter of fact, the maximum number of spacecraft that can be packed into a cluster is for sure an important parameter to be considered for a cluster mission, but at the same time one cannot forget the key role played by the ΔV budget.

To perform this type of analysis, several sample slots configurations have been selected and for each configuration simulations have been run assigning a spacecraft to each of its slots. In particular, a direct correspondence has been implemented so that each spacecraft is assigned to a single slot and tries to track it using station keeping maneuvers. Each simulation consisted in a propagation of the cluster initial configuration in an inertial frame under the effect of the gravitational potential (terms up to J_{20} have been included) and of the drag. The different TIC examined try to counteract the differential perturbations in a different way, hence it is reasonable to expect a different evolution of the slots over time and a different amount of ΔV required for the station keeping.

The station keeping can be thought of as composed by three phases. At first the spacecraft computes the offset, in terms of differential orbital elements, of its current state from its reference state. Then the maneuver to cancel the offset is computed as a sequence of impulsive control actions aimed at correcting the orbital

Table 2 Number of spacecraft obtainable for five sample cluster configurations

D_R	EIVS	DBNO	J2In	DeEl
1/5	14	4	4	10
5/25	12	4	6	10
9/45	12	4	8	10
13/65	12	4	7	10
17/85	10	4	7	10

elements. Finally the maneuver is executed. Please note that in the second phase each spacecraft computes the maneuver taking only its own elements into account. The relative motion w.r.t. the other spacecraft is entirely neglected, hence during the execution of the control actions small violations of the distance boundaries might be recorded.

For this study, five sample configurations have been observed, one for each value of the D_{min} parameter. The maximum distance constraints are instead based on a value $D_{max} = 5D_{min}$. Table 2 summarizes the five cases showing the number of spacecraft n_m for the four techniques.

For each examined configuration several tests have been performed, since it has been decided to implement a different logic to trigger the station keeping maneuvers:

PM a fixed periodic maneuvering cycle is implemented, i.e. at regular time intervals all the spacecraft measure their offsets, compute the corrective maneuver to cancel them and perform it;

CV each spacecraft computes and performs the required maneuvers only when the violation of one of the constraints MinDC or MaxDC is detected.

The PM logic has the advantage that it allows to know a priori the times at which the computation of the maneuver is performed, but as a drawback the constraints on the distances do not play any role and if a violation occurs nothing is done to fix it. Conversely, in the CV logic the distance constraints are actively used and only the involved spacecraft actually perform a maneuver, but as it will be seen exploring the results, it can happen that some spacecraft require very sparse but large corrections, that can produce severe violations of the constraints.

One parameter for the evaluation and comparison of the results can be identified in the average ΔV required by the cluster, i.e. the total ΔV required by all the spacecraft averaged w.r.t. n_m. This choice is suggested considering the fact that the number of spacecraft changes for each of the four techniques, hence the comparison of the total ΔV would make no sense.

Figures 7 and 8 compare the results for the initial configuration obtained with the technique EIVS for $D_R = 1/5$. The considered time horizon is 10 days. In Fig. 7 the ΔV budget is shown while Fig. 8 gives an overview of the number of required control actions. In both plots a single circle refers to a single spacecraft of the cluster. The spreading of the circles for a given maneuver logic depends on the fact that according to the distance between a spacecraft and the VP, the differential

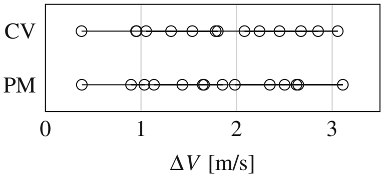


Fig. 7 $\varDelta V$ budget required by an initial configuration from the technique EIVS for two maneuver logics

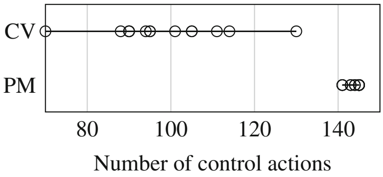


Fig. 8 Number of control actions required by an initial configuration from the technique EIVS for two maneuver logics

Table 3 Distance violations detected by an initial configuration from the technique EIVS for two maneuver logics

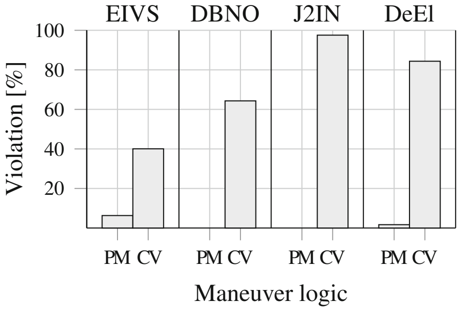
Logic	$D_{min}(\%)$	$D_{max}(\%)$
PM	6.73	1.22
CV	21.18	2.08

perturbations experienced vary and then a different amount of $\varDelta V$ is required. With particular reference to Fig. 8, when the CV maneuver logic is implemented the different differential perturbations cause the spacecraft to violate the distance constraints at different times, that is why the circles are so spread. On the other hand, when the PM logic is used, all the spacecraft perform the corrective maneuvers periodically, hence they all require approximately the same number of control actions. It can then be summarized that for the shown example the maneuver logic does not have a big impact on the average $\varDelta V$, rather it affects the number of control actions used and the way in which the propulsion system is stressed. As a matter of fact, an increase in the time interval between two maneuvers implies larger sparse maneuvers instead of shorter frequent ones.

Table 3 gives instead a measure of the constraints violations recorded in the considered example showing the minimum and the maximum distances reached.

The use of percentage values stems from the comparison of the maximum violations with the foreseen boundaries set at $D_{min} = 1$ km and $D_{max} = 5$ km. The MaxDC violations are on the same order of magnitude for both the maneuver schemes PM and CV, meaning that in both cases the maximum distance reached is up to approximately 2% larger than the 5 Km boundary value. A significant difference can be seen instead on the MinDC side, since in the PM case the minimum distance reached stays above 93% of the 1 Km boundary value, while in the CV case it can decrease until

Fig. 9 Violations overview of the MinDC



78% of it. It must be pointed out that such constraints violations occur sporadically and always within the first one or two orbits after the execution of the first control action, and as such this does not mean that the station keeping strategy does not work. This behaviour was anticipated and can be attributed to the particular way the trajectory corrections have been implemented. As a matter of fact the station keeping is not instantaneous, but is composed of several separate impulsive control actions, taking place at different locations along the orbit. Between the first and the last control action the spacecraft travel on transfer orbit arcs and it is in this short timeframe that the violations are recorded. Indeed, as it can be recalled from the description of the three phases of the station keeping, for each spacecraft the required maneuvers are computed to ensure the reference state is tracked but without taking into consideration the relative motion w.r.t. the other spacecraft during the execution of the maneuver itself.

Despite the fact that the CV control logic actually requires a violation of the distance boundaries to trigger the computation of a correcting maneuver, in general a violation is clearly undesirable. Nevertheless in this part of the study the intention was not really to prevent violations as a whole, rather to see if a station keeping approach could be sufficient to perform cluster keeping and to obtain a raw evaluation of the ΔV budget required by different configuration designs.

In Figs. 9 and 10 an overview of the violations is depicted for the MinDC and the MaxDC, respectively. Each figure is divided into four parts in order to compare the different techniques. Each part illustrates the behaviour of a TIC through two bars, one for each of the two tested maneuver logics. Recalling that for a single TIC and a single maneuver logic five sample configurations have been tested, each bar renders the worst out of the five cases, i.e. the largest percentage violation w.r.t. the reference distance boundary. For example if one considers the EIVS technique and a station keeping strategy applied regularly every five orbits, it can be seen that the maximum violations of the MinDC and the MaxDC are below 10% and 5%, respectively.

Observing the different parts of the figures individually, it can be noticed that the CV bar is always higher than the PM bar. This trend depends on the fact that, as already mentioned, when the timeframe between two consecutive maneuvers increases, the differential drift of the spacecraft from the path of their reference initial states increases as well, so that larger correcting maneuvers are required and

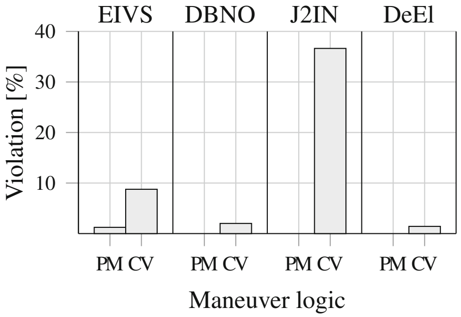


Fig. 10 Violations overview of the MaxDC

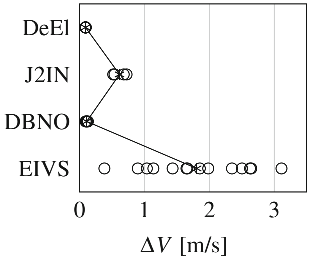


Fig. 11 ΔV budget for clusters obtained with different TIC (maneuver logic PM and $D_R = 1/5$)

larger violations can be experienced. In addition, in the CV cases the bars can become much higher than the PM cases, due to the fact that a maneuver already starts with a violation, which then is accentuated by the application of the corrections.

As a final outcome, it is clear that the station keeping framework is not entirely violations-free and that the relative motion during the transfer paths should also be taken into consideration. If this aspect of the problem was implemented, the control strategy would have been different, consisting of another sequence of control actions but without a substantial change in the overall estimated ΔV budget.

To conclude this part of the study and the comparison of the different techniques, let us now focus on the results of the station keeping investigation in terms of the ΔV required by the spacecraft. So far it has been seen that the different TIC provide a different number of slots and react differently to the distance constraints, but a comparison of the ΔV budget is still missing.

To develop this analysis, two points of view can be considered. Let us focus on a single D_R. On one side one can fix the maneuver logic and pay the attention to entire configurations. On the other side it could be interesting to fix a maximum ΔV budget and see, for each TIC, how many spacecraft comply with this limitation. For a better understanding, Figs. 11 and 12 come in handy, respectively.

Figure 11 focuses on the case $D_R = 1/5$ with station keeping performed every five orbits. Each circle denotes the ΔV required by a spacecraft during the entire time frame of 10 days, while the stars and the solid line connecting them highlight the average ΔV required by the entire cluster. As expected, the less differential perturbations are included in the definition of the initial configuration, the larger the

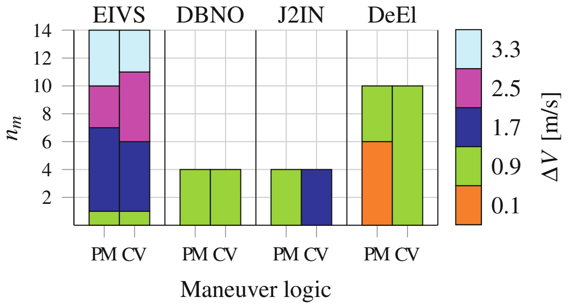


Fig. 12 Number of deployable spacecraft according to different ΔV thresholds for different TIC and different maneuver logics, in the case $D_R = 1/5$

drift among the spacecraft is and the less homogeneous the fuel consumption is. That is why for the DeEl and DBNO the ΔV of the single spacecraft are very close to the average value, while the spreading increases for the EIVS and the J2In.

The second type of research led instead to Fig. 12, which refers again to the case $D_R = 1/5$, but this time all the TIC and the maneuver logics are included. In this analysis it has been identified how many spacecraft could be deployed from the different investigated clusters, when a constraint on the maximum available ΔV is set. For the case given in Fig. 12 five threshold values are considered. It can be observed for example that if the smallest threshold is selected, only spacecraft consuming less than 0.1 m/s can be used, which means that only one configuration is available, i.e. a six-objects cluster from the DeEl technique performing station keeping every five orbits. If the threshold is raised to 0.9 m/s, some solutions from the DBNO and J2In become also available, while to use a cluster from the EIVS a threshold of 1.7 m/s is required. Similar results are obtained from the other D_R, with the DeEl always demonstrating a good compromise between high number of spacecraft and low ΔV budget, followed by the DBNO, the J2In and finally the EIVS. In comparison, the EIVS can always provide the most populated clusters, but the price for this asset is paid in terms of high ΔV requirements.

4 Conclusions

Exploiting the concept of fractionation it is possible to split and replace a single monolithic spacecraft with multiple smaller ones flying in proximity. In such a framework a safe relative motion must be guaranteed, bounding the minimum and the maximum relative distances to prevent, respectively, collisions and escaping drifts. With a proper orbit design, the spacecraft might initially satisfy the relative motion constraints, but due to the differential perturbations acting on them, over time their configuration slowly becomes unsafe and corrective maneuvers must be performed. In this work several designs have been surveyed, each providing a cluster initial configuration while counteracting the differential perturbations with a different approach. The identified test setup assumes that: (1) the cluster is centered on a virtual point;

(2) a minimum distance must be ensured between any pair of spacecraft to prevent collisions; (3) a maximum distance must be ensured between any spacecraft and the virtual point to prevent escaping drifts; (4) each spacecraft tracks a reference state through station keeping maneuvers.

In the first part of the study, it has been investigated how the number of deployable spacecraft changes according to the design and to the minimum and maximum distance constraints. Later on, the obtained initial configurations have been propagated in time and station keeping maneuvers have been implemented to evaluate the ΔV budget required by the single spacecraft and the entire clusters.

It emerged that in the design phase several aspects of the problem need to be taken into account, such as the number of deployable spacecraft, the need to introduce countermeasures against violations of distance boundaries, the differential consumption of fuel, etc. All these aspects are clearly interconnected with each other and it is not possible to identify a configurations which has the best performances from all points of view. According to the specific requirements of the mission the best fitting configuration can be different, but the conducted study provides an indication and an analysis method.

Acknowledgements This research has been funded by the German Israeli Foundation Grant No. 1181-220.10.

References

1. Balakrishnan V (1997) Schaum's Outline of Theory and Problems of Graph Theory. Schaum's Series. McGraw-Hill Education, New York
2. Battin R (1999) In: Przemieniecki JS (ed) An Introduction to the Mathematics and Methods of Astrodynamics, Revised edn. American Institute of Aeronautics & Astronautics, Reston
3. Beigelman I et al (2008) Optimal fuel-balanced impulsive formationkeeping for perturbed spacecraft orbits. J Guid Control Dyn 31(5):1266–1283
4. Carraghan R et al (1990) An exact algorithm for the maximum clique problem. Oper Res Lett 9(6):375–382
5. Carter T et al (1987) Fuel-optimal rendezvous near a point in general Keplerian orbit. J Guid Control Dyn 10(6):567–573
6. Clohessy W et al (1960) Terminal guidance system for satellite rendezvous. J Aerosp Sci 27(9):653–658
7. D'Amico S (2010) Autonomous Formation Flying in Low Earth Orbit. Ph.D. thesis. Delft, Netherlands: Technical University of Delft
8. D'Amico S et al (2006) Proximity operations of formation-flying spacecraft using an eccentricity/ inclination vector separation. J Guid Control Dyn 29(3):554–563
9. de Bruijn FJ et al (2015) Delayed target tracking for along-track formations. J Guid Control Dyn 38(7):1318–1323
10. Diestel R (2000) Electronic library of mathematics. Graph Theory. Springer, New York
11. Eckstein M et al (1989) Colocation strategy and collision avoidance for the geostationary satellites at 19 degrees west. In: International Symposium on Space Dynamics. Toulouse, France
12. Fumenti F et al (2015) Quasi-impulsive maneuvers to correct mean orbital elements in LEO. In: Proceedings of the 3rd CEAS EuroGNC. Toulouse, France

13. Gurfil P et al (2012) The SAMSON project - cluster flight and geolocation with three autonomous nano-satellites. In: 26th AIAA/USU Conference on Small Satellites. SSC12-VII-2. Salt Lake City, UT, USA
14. Hill G (1878) Researches in the lunar theory. Am J Math 1(1):5–26
15. LoBosco D et al (2008) The pleiades fractionated space system architecture and the future of national security space. In: AIAA Space Conference and Exposition. AIAA 2008-7687. San Diego, California: American Institute of Aeronautics and Astronautics
16. Mazal L et al (2014) Closed-loop distance-keeping for long-term satellite cluster flight. Acta Astronaut. 94(1):73–82
17. Mazal L et al (2013) Cluster flight algorithms for disaggregated satellites. J Guid Control Dyn 36(1):124–135
18. Mishne D (2004) Formation control of satellites subject to drag variations and J2 perturbations. J Guid Control Dyn 27(4):685–692
19. Montenbruck O et al (2006) E/I-vector separation for safe switching of the GRACE formation. Aerosp Sci Technol 10(7):628–635
20. Moreira A et al (2004) TanDEM-X: a TerraSAR-X add-on satellite for single-pass SAR interferometry. In: International Geoscience and Remote Sensing Symposium, vol 2
21. Persson S et al (2005) PRISMA - Demonstration mission for advanced rendezvous and formation flying technologies and sensors. In: 56th International Astronautical Congress. IAC-05-B5.6.B.07. Fukuoka, Japan
22. Roscoe CWT et al (2015) Formation establishment and reconfiguration using differential elements in J2-perturbed orbits. J Guid Control Dyn 38(9):1725–1740
23. Schaub H et al (2002) Hybrid cartesian and orbit element feedback law for formation flying spacecraft. J Guid Control Dyn 25(2):387–393
24. Schaub H et al (2001) J2 invariant relative orbits for spacecraft formations. Celest Mech Dyn Astron 79(2):77–95
25. Schaub H et al (2000) Spacecraft formation flying control using mean orbit elements. J Astronaut Sci 48(1):69–87
26. Tapley BD (2008) Gravity model determination from the GRACE mission. J Astronaut Sci 56(3):273–285
27. Tschauner J et al (1965) Rendezvous zu Einem in Elliptischer Bahn um Laufenden Ziel. Acta Astronaut 11(5)
28. Wood DR (1997) An algorithm for finding a maximum clique in a graph. Oper Res Lett 21(5):211–217
29. Yamanaka K et al (2002) New state transition matrix for relative motion on an arbitrary elliptical orbit. J Guid Control Dyn 25(1):60–66

Generalized Image Navigation and Registration Method Based on Kalman Filter

Ahmed A. Kamel, Handol Kim, Dochul Yang, Chulmin Park and Jin Woo

Nomenclature

R_{so}:	ideal geosynchronous radius = 42164000 m
λ_{so}:	ideal satellite longitude
ω_e:	sidereal earth rotation rate = 7.2921159E-05 rad/s
R, L,λ:	radius, geocentric latitude, longitude
R_{eo}:	equatorial radius = 6378136.6 m
f:	earth flattening = 1/298.25642
h:	landmark altitude
ϕ, θ, ψ:	roll, pitch, yaw
ECLF:	earth centered local frame
GEOS:	fixed grid frame
LOS:	line of sight
LRF:	LOS reference frame
IIRF:	imager internal reference frame
ACF:	attitude control frame
ORF:	orbit reference frame

A.A. Kamel (✉)
Kamel Engineering Services, Los Angeles, CA, USA
e-mail: ahmed@kamelengineering.com

H. Kim · D. Yang
Korea Aerospace Research Institute, Daejeon, Republic of Korea
e-mail: hkim@kari.re.kr

D. Yang
e-mail: dcyang@kari.re.kr

C. Park
Korea Aerospace Industries, Sacheon, Republic of Korea
e-mail: cmpark@koreaaero.com

J. Woo
Korea Meteorological Administration, Seoul, Republic of Korea
e-mail: superjwoo@korea.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_33

I_{ixi}, 0_{ixj}:	ixi identity matrix, ixj null matrix
S, C:	Sin, Cos
EW, NS:	East-West, North-South

Subscripts

0, s, e:	initial, satellite, earth
T:	landmark point on earth
ma:	misalignment
m:	number of imager internal misalignment
att:	spacecraft attitude from telemetry
corr:	thermoelastic/attitude correction angles

1 Introduction

The term image navigation and registration and the INR acronym were coined by Kamel [1] and patented in US Patents # 4,688,091, 4,688,092, and 4,746,976 to represent a system that determines image pixel location and register it to fixed grid frame (called FGF in GOES and GEOS in COMS and in this paper). This INR invention became the foundation for subsequent GOES and similar systems worldwide [2–5]. The INR system requirements tightened as spacecraft and ground hardware improved [6–9].

The image navigation part of INR relates to LOS absolute pointing. Section 2 defines the INR and KF state vectors needed for this process. Section 3 describes new INR method (patent application being filed in ROK) based on landmark measurements to determine orbit, attitude correction, and imager misalignments with maneuvers delta V provided by FDS. Also, orbit refinement can be made if FDS provides orbit with coarse accuracy instead of delta V. Section 4 shows the simulation results of this basic system. Section 5 shows how the new method can be adapted to be used for other INR systems implemented nowadays.

The image registration part of INR relates to LOS stability. The objective of image registration is to provide the users with images with pixels that have the same fixed earth location regardless of time. Section 6 provides an algorithm for transferring pixels from LOS frame to GEOS frame needed for pixel data resampling in GEOS frame.

2 INR and KF SV Definitions

The INR and KF SV definitions and the associated time series are given in the next three subsections.

2.1 INR SV Definition

The INR SV is required for transformation from LRF to GEOS for Sect. 3.1.2. This is given by:

$$\mathrm{SV_{INR}} = [\mathrm{SV_{ma}^T}\ \mathrm{SV_{corr}^T}\ \mathrm{SV_{att}^T}\ \mathrm{SV_{orb}^T}]^\mathrm{T} \tag{1}$$

$\mathrm{SV_{ma}}$ is based on IIRF misalignment relative to LRF.
$\mathrm{SV_{corr}}$, $\mathrm{SV_{att}}$, and $\mathrm{SV_{orb}}$ are based on:
$(\phi_{\mathrm{corr}}\ \theta_{\mathrm{corr}}\ \psi_{\mathrm{corr}})$ = ACF attitude relative to IIRF.
$(\phi_{\mathrm{att}}, \theta_{\mathrm{att}}, \psi_{\mathrm{att}})$ = ORF attitude relative to ACF.
$(\phi_{\mathrm{orb}}, \theta_{\mathrm{orb}}, \psi_{\mathrm{orb}})$ = GEOS attitude relative to ORF.

For single mirror imagers, such as GOES I-P, COMS, MTSAT2, $\mathrm{SV_{ma}}$ is given by:

$$\mathrm{SV_{ma}} = [\phi_{\mathrm{ma}}\ \theta_{\mathrm{ma}}]^\mathrm{T} \tag{2.1}$$

$$= \mathrm{SV_{ma,model}} + \mathrm{x_{ma}} \tag{2.2}$$

$$\mathrm{SV_{ma,model}} = [\phi_{\mathrm{ma,model}}\ \theta_{\mathrm{ma,model}}]^\mathrm{T}$$

$$\mathrm{SV_{corr}} = [\phi_{\mathrm{corr}}\ \theta_{\mathrm{corr}}\ \psi_{\mathrm{corr}}]^\mathrm{T} \tag{3.1}$$

$$= \mathrm{SV_{corr,model}} + \mathrm{x_{corr}} \tag{3.2}$$

$$\mathrm{SV_{corr,model}} = [\phi_{\mathrm{corr,model}}\ \theta_{\mathrm{corr,model}}\ \psi_{\mathrm{corr,model}}]^\mathrm{T}$$

The thermoelastic misalignment and correction models are computed in Sect. 3.4 and $(\mathrm{x_{ma}}, \mathrm{x_{corr}})$ are defined in Sect. 2.2 and determined by KF.

$$\mathrm{SV_{att}} = [\phi_{\mathrm{att}}\ \theta_{\mathrm{att}}\ \psi_{\mathrm{att}}]^\mathrm{T}\ \text{from telemetry} \tag{4}$$

$$\mathrm{SV_{orb}} = [\mathrm{R_s}\quad \Delta\lambda_s\quad \mathrm{L}_s\ ]^\mathrm{T} \tag{5.1}$$

$$\mathrm{R_s} = \mathrm{R_{so}}\left(1 + \frac{\Delta \mathrm{R_s}}{\mathrm{R_{so}}}\right),\ \Delta\lambda_\mathrm{s} = \lambda_\mathrm{s} - \lambda_{\mathrm{so}} \tag{5.2}$$

For 3,1,2 type rotation, $\mathrm{SV_{ORF}}$ is given by:

$$\mathrm{SV_{ORF}} = [\phi_{\mathrm{orb}}\ \theta_{\mathrm{orb}}\ \psi_{\mathrm{orb}}]^\mathrm{T} \tag{6.1}$$

For Spacecraft x axis parallel to earth equator (e.g., COMS):

$$\mathrm{SV_{ORF}} = [\mathrm{L_s}\ \Delta\lambda_\mathrm{s}\ 0]^\mathrm{T} \tag{6.2}$$

For Spacecraft x axis parallel to orbit plane (e.g., GOES I-M):

$$\mathrm{SV_{ORF}} = [\mathrm{L_s}\ \Delta\lambda_\mathrm{s}\ \dot{\mathrm{L}}_\mathrm{s}/\omega_\mathrm{e}]^\mathrm{T} \tag{6.3}$$

$\left(\frac{\Delta \mathrm{R_s}}{\mathrm{R_{so}}}, \Delta\lambda_\mathrm{s}, \mathrm{L_s}\ \dot{\mathrm{L}}_\mathrm{s}/\omega_\mathrm{e}\right)$ are Kamel parameters [10, 11] originally used for GOES I-M.

If FDS provides maneuver delta V:

$$\frac{\Delta R_s}{R_{so}} = \frac{\delta\,\mathrm{R_s}}{\mathrm{R_{so}}}, \Delta\lambda_s = \delta\,\lambda_s, \mathrm{L}_s = \delta\,\mathrm{L}_s, \mathrm{L}_s = \delta\,\mathrm{L}_s \tag{7.1}$$

$$\left(\frac{\delta\,\mathrm{R_s}}{\mathrm{Rso}}, \delta\,\lambda_s, \delta\,\mathrm{L}_s, \delta\,\dot{\mathrm{L}}_s\right) = \text{ideal ordeal refinement by KF.}$$

If FDS provides orbit instead of maneuver delta V:

$$\mathrm{R_s} = \mathrm{Rso}\left[\frac{\mathrm{R_{FDS}}}{\mathrm{Rso}} + \frac{\delta\,\mathrm{R_s}}{\mathrm{Rso}}\right] \Delta\lambda_\mathrm{s} = \Delta\lambda_\mathrm{FDS} + \delta\,\lambda_\mathrm{s},$$

$$\mathrm{L}_s = \mathrm{L_{FDS}} + \delta\,\mathrm{L}_s, \dot{\mathrm{L}}_s = \dot{\mathrm{L}}_\mathrm{FDS} + \delta\,\dot{\mathrm{L}}_s \tag{7.2}$$

$$\left(\frac{\delta\,\mathrm{R_s}}{\mathrm{Rso}}, \delta\,\lambda_s, \delta\,\mathrm{L}_s, \delta\,\dot{\mathrm{L}}_s\right) = \text{FDS ordeal refinement by KF.}$$

2.2 *KF SV Definition*

$\mathrm{SV_{KF}} = \mathrm{x}$ is needed to determine $\mathrm{SV_{INR}}$of Sect. 2.1. This is defined as follows:

$$\mathrm{X} = [\mathrm{X}_\mathrm{corr}^\mathrm{T}\quad \dot{\mathrm{X}}_\mathrm{corr}^\mathrm{T}\quad \mathrm{X}_\mathrm{orb}^\mathrm{T}\quad \dot{\mathrm{X}}_\mathrm{orb}^\mathrm{T}\quad \mathrm{X}_\mathrm{ma}^\mathrm{T}\quad \dot{\mathrm{X}}_\mathrm{ma}^\mathrm{T}]^\mathrm{T} \tag{8.1}$$

$$\mathrm{X_{corr}} = [\delta\,\phi_\mathrm{corr}\quad \delta\,\theta\mathrm{corr}\ \delta\,\psi_\mathrm{corr}]^\mathrm{T} \tag{8.2}$$

$$\dot{\mathrm{X}}_\mathrm{corr} = \left[\mathrm{b}_{\phi_\mathrm{corr}}\quad \mathrm{b}_{\theta_\mathrm{corr}}\quad \mathrm{b}_{\psi_\mathrm{corr}}\right]^\mathrm{T} = \text{constant} \tag{8.3}$$

$$\mathrm{X_{orb}} = \left[\frac{\delta\,\mathrm{R_s}}{\mathrm{R_{so}}}\quad \delta\,\lambda_\mathrm{s}\quad \delta\,\mathrm{L_s}\right]^\mathrm{T} \tag{8.4}$$

$$\dot{\mathrm{X}}_\mathrm{orb} = \left[\frac{\delta\,\dot{\mathrm{R}}_\mathrm{s}}{\mathrm{R_{so}}}\quad \delta\,\dot{\lambda}_\mathrm{s}\quad \lambda\dot{\mathrm{L}}_\mathrm{s}\right]^\mathrm{T} \tag{8.5}$$

$$\mathrm{X_{ma}} = [\delta\,\phi_\mathrm{ma}\quad \delta\,\theta\mathrm{ma}]^\mathrm{T} \tag{8.6}$$

$$\dot{\mathrm{X}}_\mathrm{ma} = [\mathrm{b}_{\phi_\mathrm{ma}}\quad \mathrm{b}_{\theta_\mathrm{ma}}]^\mathrm{T} = \text{constant} \tag{8.7}$$

At KF start, $\mathrm{x} = 0_{12+2\mathrm{m}}$.

2.3 SV Time Series

SV_{INR} time series are generated at points spaced by Δt_i for image registration of Sect. 6. This requires interpolation between SV_{KF} time series points determined by landmarks (or star measurements) time series points based on attitude telemetry (e.g., at one second interval) and FDS orbit, $SV_{ma,model}$, and $SV_{corr,model}$ time series (e.g., at one minute interval). The SV_{KF} time series between measurements can be obtained as follows:

$$X(t_i) = A(\Delta t_i)X(t_0), \quad \Delta t_i = t_i - t_i - t_0, t_0 \le t_i \le t_1 \tag{9.1}$$

$$A(\Delta t_i) = \begin{bmatrix} A_{corr}(\Delta t_i) & 0_{6\times 6} & 0_{6\times 2m} \\ 0_{6\times 6} & A_{orb}(\Delta t_i) & 0_{6\times 2m} \\ 0_{2m\times 6} & 0_{2m\times 6} & A_{ma}(\Delta t_i) \end{bmatrix} \tag{9.2}$$

$$A_{corr}(\Delta t_i) = \begin{bmatrix} I_{3\times 3} & I_{3\times 3}\Delta t_i \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} \tag{9.3}$$

$A_{orb}(\Delta t_i)$ obtained from the well-known Euler-Hill equations [12]

$$A_{orb}(\Delta t_i) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{9.4}$$

$$A_{11} = \begin{bmatrix} (4 - 3\,C) & 0 & 0 \\ 6(S - \gamma) & 1 & 0 \\ 0 & 0 & C \end{bmatrix} \tag{9.5}$$

$$A_{12} = \begin{bmatrix} \omega_e^{-1}S & 2\omega_e^{-1}(1 - C) & 0 \\ -2\omega_e^{-1}(1 - C) & \omega_e^{-1}(4S - 3\,\gamma) & 0 \\ 0 & 0 & \omega_e^{-1}S \end{bmatrix} \tag{9.6}$$

$$A_{21} = \begin{bmatrix} 3\omega_e S & 0 & 0 \\ 6\omega_e^{-1}(C - 1) & 0 & 0 \\ 0 & 0 & -\omega_e S \end{bmatrix} \tag{9.7}$$

$$A_{22} = \begin{bmatrix} C & 2\,S & 0 \\ -2\,S & (4C - 3) & 0 \\ 0 & 0 & C \end{bmatrix} \tag{9.8}$$

$$\mathrm{C} = \mathrm{Cos}\ \gamma,\quad \mathrm{S} = \mathrm{Sin}\ \gamma,\quad \gamma = \omega_{\mathrm{e}}\Delta \mathrm{t}_i,\quad \omega_{\mathrm{e}}^{-1} = \frac{1}{\omega_e}.$$

For small Δt_i, $C = 1$ and $S = \gamma = \omega_e \Delta t_i$,

$$\mathrm{A}_{\mathrm{orb}}(\Delta \mathrm{t}_i) = \begin{bmatrix} \mathrm{I}_{3\times 3} & \mathrm{I}_{3\times 3}\Delta \mathrm{t}_i \\ 0_{3\times 3} & \mathrm{I}_{3\times 3} \end{bmatrix} \tag{9.9}$$

Note that Euler-Hill equations used to model orbit and Sect. 3.4 used to model thermoelastic angles leads to significant reduction of the number of landmarks processed by KF compared to using simple linear models that are only valid for short time.

$$\mathrm{A}_{\mathrm{ma}}(\Delta \mathrm{t}_i) = \begin{bmatrix} \mathrm{I}_{\mathrm{m}\times \mathrm{m}} & \mathrm{I}_{\mathrm{m}\times \mathrm{m}}\Delta \mathrm{t}_i \\ 0_{\mathrm{m}\times \mathrm{m}} & \mathrm{I}_{\mathrm{m}\times \mathrm{m}} \end{bmatrix} \tag{9.10}$$

m = number of imager internal misalignments.

For single mirror imager used for GOES I-P, COMS, MTSAT2 and in this paper, m = 2. For two mirror imagers, the number of misalignments depend on the thermoelastic effect on pointing. The leading term was called Orthogonality (O_{ma}) by Kamel because it represents deviation of the scanning axes from being perpendicular. Note that if only O_{ma} has significant effect on pointing [11], the number of misalignments m = 1.

3 Image Navigation Using KF

Figure 1 shows KF flow for the basic INR method. KF uses one landmark at a time to determine best (a-posteriori) state vector and covariance matrix estimate (x_1^+, P_1^+). KF is then re-initialized to make propagation always between t_0 and t_1 and estimation at t_1.

The 3-step process is as follows:

1. a-priori state vector and covariance matrix (x_1^-, P_1^-) obtained from (x_0^+, P_0^+) using the transition matrix A(Δt), $\Delta t = t_1 - t_0$ and error matrix Q(Δt) obtained from system model. This first step is called SV and covariance matrix P propagation

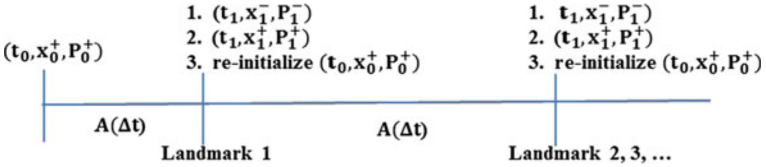


Fig. 1 Kalman filter for the basic INR method

between two successive landmarks.

$$x_1^- = A(\Delta t)x_0^+ \tag{10.1}$$

$$P_1^- = A(\Delta t)P_0^+A(\Delta t)^T + Q(\Delta t) \tag{10.2}$$

2. (x_1^+, P_1^+) obtained from (x_1^-, P_1^-) and measurement model (Z, H,R). This second step is called SV and covariance matrix P estimation at t_1. Kalman assumed the relation- ship between $x_1^+$and x_1^- is given by a form like least squares and determined associated Kalman gain matrix K and covariance matrix P:

$$x_1^+ = x_1^- - K\Delta Z,\ \ \Delta Z = Z - \bar{Z} \tag{11.1}$$

$$K = P_1^- H^T(HP_1^- H^T + R)^{-1} \tag{11.2}$$

$$P_1^+ = (I - KH)P_1^-(I - KH)^T + KRK^T \tag{11.3}$$

The residual ΔZ is computed as follows:

- Compute SV_{INR} from x_1^- using Sects. 2.1 and 2.2.
- Compute landmark residuals using Sect. 3.1.
- If landmark is rejected because residual is outside predetermined limit:
  - Re-initialize KF: $(t_0, x_0^+, P_0^+) = (t_1, x_1^+, P_1^+) = (t_1, x_1^-, P_1^-)$.
  - Skip estimation and go to next landmark.

If landmark is accepted, compute x_1^+ using Eq. (11.1).
Note that $(\Delta x_{corr}^+, \Delta x_{orb}^+, \Delta x_{ma}^+) = (x_{corr}^+, x_{orb}^+, x_{ma}^+) - (x_{corr}^-, x_{orb}^-, x_{ma}^-)$ obtained from Eq. (11.1) can cause jumps in level 1B images at t_1. This can be avoided by replacing $(\dot{x}_{corr}^+, \dot{x}_{orb}^+, \dot{x}_{ma}^+)$ [also obtained from Eq. (11.1) and given by Eqs. (8.3), (8.5) and (8.7)] with $(\dot{x}_{corr}^+, \dot{x}_{orb}^+, \dot{x}_{ma}^+) + (\Delta x_{corr}^+, \Delta x_{orb}^+, \Delta x_{ma}^+)/\,\delta t$, where, $\delta t =$ delta time to next landmark or next KF point. After this slope adjustment, set $(x_{corr}^+, x_{orb}^+, x_{ma}^+) = (x_{corr}^-, x_{ma}^-, x_{ma}^-)$ at t_1.

3. The third step is to re-initialize KF by setting $(t_0, x_0^+, P_0^+) = (t_1, x_1^+, P_1^+)$ to start the next cycle from t_0 to t_1 and compute SV_{INR} from x_1^+ using Sect. 2. This is needed for Sect. 6.

3.1 KF Landmark Residual Computation

The landmark residuals $\Delta Z = Z - \bar{Z}$ are computed from the next two subsections.

3.1.1 Actual Landmark Measurement $\bar{Z}$

In view of Fig. 2, we get:

$$\vec{R}_{To} = \vec{T} - \vec{R}_{So} \tag{12.1}$$

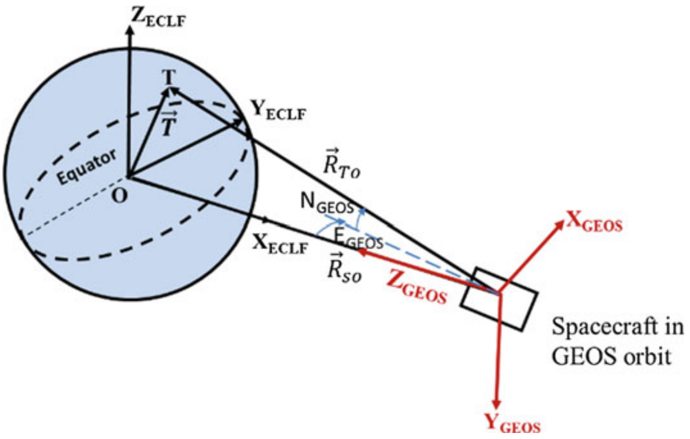


Fig. 2 ECLF to GEOS geometry

Using vector components in GEOS coordinates, we get:

$$\begin{aligned}\vec{\mathrm{R}}_{\mathrm{To}} &= (\mathrm{R_e}+\mathrm{h})\begin{bmatrix}\mathrm{C_{L_T}S_{\Delta\lambda_T}}\\ -\mathrm{S_{L_T}}\\ -\mathrm{C_{L_T}C_{\Delta\lambda_T}}\end{bmatrix}-\mathrm{R_{so}}\begin{bmatrix}0\\0\\-1\end{bmatrix}\\ &= \mathrm{R_{To}}\begin{bmatrix}\mathrm{C_{\bar{N}_{GEOS}}S_{\bar{E}_{GEOS}}}\\ -\mathrm{S_{\bar{N}_{GEOS}}}\\ \mathrm{C_{\bar{N}_{GEOS}}C_{\bar{E}_{GEOS}}}\end{bmatrix}\end{aligned} \tag{12.2}$$

$$\mathrm{R_e} = \mathrm{R_{eo}}(1+\mathrm{aS^2_{L_T}})^{-\frac{1}{2}} \cong \mathrm{R_{eo}}(1-\mathrm{f\,S^2_{L_T}}) \tag{12.3}$$

$$\Delta\lambda_{\mathrm{T}} = \lambda_{\mathrm{T}} - \lambda_{\mathrm{so}},\ \mathrm{a} = (1-\mathrm{f})^{-2}-1 \cong 2\mathrm{f} \tag{12.4}$$

This leads to:

$$\mathrm{R_{To}} = \sqrt{\mathrm{R^2_{so}} + (\mathrm{R_e}+\mathrm{h})^2 - 2\mathrm{R_{so}}(\mathrm{R_e}+\mathrm{h})\mathrm{C_{L_T}C_{\Delta\lambda_T}}} \tag{13.1}$$

$$\bar{\mathrm{E}}_{\mathrm{GEOS}} = \mathrm{Arc\ tan}\left[\frac{(\mathrm{R_e}+\mathrm{h})\mathrm{C_{L_T}S_{\Delta\lambda_T}}}{\mathrm{R_{so}} - (\mathrm{R_e}+\mathrm{h})\mathrm{C_{L_T}C_{\Delta\lambda_T}}}\right] \tag{13.2}$$

$$\bar{\mathrm{N}}_{\mathrm{GEOS}} = \mathrm{Arc\ sin}\left[\frac{(\mathrm{R_e}+\mathrm{h})\mathrm{S_{L_T}}}{\mathrm{R_{To}}}\right] \tag{13.3}$$

$$\bar{\mathrm{Z}} = \begin{bmatrix}\bar{\mathrm{E}}_{\mathrm{GEOS}}\\ \bar{\mathrm{N}}_{\mathrm{GEOS}}\end{bmatrix} \tag{13.4}$$

3.1.2 Estimated Landmark Measurement Z

Transformation of landmark (E_{LRF}, N_{LRF}) coordinates to (E_{GEOS}, N_{GEOS}) coordinates is obtained in the next 4 subsections.

3.1.2.1 $\widehat{U}_{IIRF}$ and $\widehat{R}_{IIRF}$ Computation

For single mirror instruments:

$$E_{IIRF} = E_{LRF} - (\phi_{ma} S_{N_{LRF}} + \theta_{ma} C_{N_{LRF}}) \tag{14.1}$$

$$N_{IIRF} = N_{LRF} - (\phi_{ma} C_{N_{LRF}} - \theta_{ma} S_{N_{LRF}})/C_{E_{LRF}} \tag{14.2}$$

(E_{LRF}, N_{LRF}) = determined landmark (EW, NS) angles.

To get (E_{LRF}, N_{LRF}) from (E_{IIRF}, N_{IIRF}) for inverse transformation, two iterations of Eqs. (14.1) and (14.2) may be needed.

The unit vector $\widehat{U}_{IIRF}$ components in IIRF coordinates is obtained by a rotation N_{IIRF} about X-axis followed by a rotation E_{IIRF} about new Y-axis. This leads to:

$$\widehat{U}_{IIRF} = \begin{bmatrix} S_{E_{IIRF}} \\ -C_{E_{IIRF}}\ S_{N_{IIRF}} \\ C_{E_{IIRF}}\ C_{N_{IIRF}} \end{bmatrix} \tag{14.3}$$

The unit vector $\widehat{R}_{IIRF}$ components in GEOS is given by:

$$\widehat{R}_{IIRF} = C_{IIRF}^{GEOS} \widehat{U}_{IIRF} = [\widehat{R}_{GEOS,X} \quad \widehat{R}_{GEOS,Y} \quad \widehat{R}_{GEOS,Z}]^T \tag{14.4}$$

Note that for inverse transformation, use:

$$\widehat{U}_{IIRF} = C_{GEOS}^{IIRF} \widehat{R}_{IIRF}, C_{GEOS}^{IIRF} = [C_{IIRF}^{GEOS}]^T \tag{14.5}$$

3.1.2.2 IIRF to GEOS Transformation Matrix Computation

Transformation from IIRF to GEOS is 3,1,2, type rotation and can be obtained from Appendix E, Table E-1, Ref. [13] by replacing (ϕ, θ, ψ) with (ψ_C, ϕ_C, θ_C):

$$\begin{aligned} C_{IIRF}^{GEOS} &= \begin{bmatrix} C_\theta C_\psi - S_\theta S_\phi S_\psi & C_\theta C_\psi + S_\theta S_\phi S_\psi & -S_\theta C_\phi \\ -S_\psi C_\phi & C_\psi C_\phi & S_\phi \\ S_\theta C_\psi + C_\theta S_\phi S_\psi & S_\theta S_\psi - C_\theta S_\phi C_\psi & C_\phi C_\theta \end{bmatrix}_C \\ &\cong \begin{bmatrix} 1 & \psi_C & -\theta_C \\ -\psi_C & 1 & \phi_C \\ \theta_C & -\phi_C & 1 \end{bmatrix} \end{aligned} \tag{15.1}$$

In view of Eqs. (3.1), (4), and (6.1) to (6.3) we get:

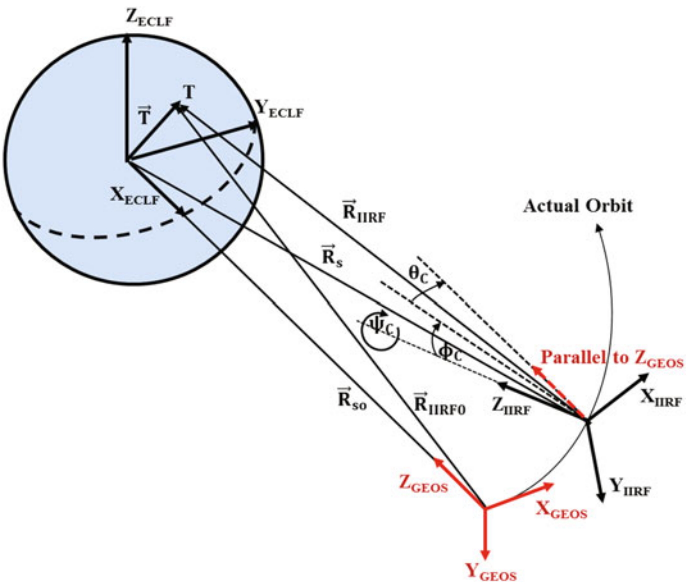


Fig. 3 Image navigation geometry

$$SV_C = \begin{bmatrix} \phi_c \\ \theta_C \\ \psi_C \end{bmatrix} = SV_{ACF} + SV_{corr} \tag{15.2}$$

$$SV_{ACF} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = SV_{ORF} + SV_{att} \tag{15.3}$$

$$SV_{corr} = \begin{bmatrix} \phi_{corr} \\ \theta_{corr} \\ \psi_{corr} \end{bmatrix}, SV_{ORF} = \begin{bmatrix} \phi_{orb} \\ \theta_{orb} \\ \psi_{orb} \end{bmatrix}, SV_{att} = \begin{bmatrix} \phi_{att} \\ \theta_{att} \\ \psi_{att} \end{bmatrix} \tag{15.4}$$

3.1.2.3 R_{IIRF} computation

In view of Fig. 3 and Eq. (14.4), we get:

$$\vec{T} = \vec{R}_S + \vec{R}_{IIRF} \tag{16.1}$$

$$(R_e + h)\begin{bmatrix} C_{L_T} S_{\Delta\lambda_T} \\ -S_{L_T} \\ C_{L_T} C_{\Delta\lambda_T} \end{bmatrix} = R_S \begin{bmatrix} C_{L_S} S_{\Delta\lambda_S} \\ -S_{L_S} \\ -C_{L_S} C_{\Delta\lambda_S} \end{bmatrix} + R_{IIRF} \begin{bmatrix} \widehat{R}_{GEOS,x} \\ \widehat{R}_{GEOS,y} \\ \widehat{R}_{GEOS,z} \end{bmatrix} \tag{16.2}$$

R_{IIRF} can be obtained from Eq. (16.1) as follows:

$$|\vec{T}| = |\vec{R}_S + \vec{R}_{IIRF}| \tag{17.1}$$

$$(\mathrm{R_e+h})^2=\mathrm{R_{IIRF}^2+R_s^2+2R_{IIRF}R_sC_{\alpha_s}} \tag{17.2}$$

$\mathrm{C_{\alpha\,s}} = -$ dot product of unit vectors $\widehat{\mathrm{R}}_\mathrm{s}$ and $\widehat{\mathrm{R}}_\mathrm{IIRF}$
$= -\widehat{\mathrm{R}}_\mathrm{GEOS,x}\mathrm{C_{Ls}S_{\Delta\lambda s}} + \widehat{\mathrm{R}}_\mathrm{GEOS,y}\mathrm{S_{Ls}} + \widehat{\mathrm{R}}_\mathrm{GEOS,z}\mathrm{C_{Ls}C_{\Delta\lambda s}}$

Solution of the quadratic Eq. (17.2) leads to:

$$\mathrm{R_{IIRF}=R_S/r} \tag{17.3}$$

$$\begin{aligned}\mathrm{r} &= \left\{\mathrm{C}_{\alpha_s}-\sqrt{\mathrm{c}_{\alpha_s}^2-\mathrm{c}_{\alpha_{so}}^2}\right\}^{-1}\\ \mathrm{C}_{\alpha_{so}}^2 &= 1-[(\mathrm{R_e+h})/\mathrm{R_s}]^2\end{aligned} \tag{17.4}$$

Note that the parameter r is the same as A (α) in [1] and r in [11] and was called earth curvature parameter by Kamel because its value is dependent on Earth curvature. $\mathrm{R_e}$ is obtained from Eq. (12.3) with $\mathrm{S_{LT}}$ from the middle row of Eq. (16.2):

$$\mathrm{s_{L_T}}=\mathrm{R_s}\left(\mathrm{S_{L_S}}-\frac{\widehat{\mathrm{R}}_\mathrm{GEOS,y}}{\mathrm{r}}\right)/(\mathrm{R_e+h}) \tag{17.5}$$

Note that because $\mathrm{S_{LT}^2}$ is multiplied by small number in Eq. (12.3), one or two iterations using Eqs. (12.3), (17.4) and (17.5), starting with $\mathrm{R_e = R_{eo}}$ in Eqs. (17.4) and (17.5), should be sufficient to get accurate values for $\mathrm{R_e}$ and r.

Note also that if $\mathrm{C}_{\alpha_s}^2 < \mathrm{C}_{\alpha_{so}}^2$, $\sqrt{\mathrm{c}_{\alpha_s}^2-\mathrm{c}_{\alpha_{so}}^2}$ in Eq. (17.4) is imaginary indicating that the image pixel ($\mathrm{E_{LRF}, N_{LRF}}$) corresponds to a point outside earth and ($\mathrm{E_{GEOS}, N_{GEOS}}$) transition from earth to space will be undefined. This can be avoided if a fictitious earth with $\mathrm{C}_{\alpha_{so}} = \mathrm{C}_{\alpha_s}$ is used in Eq. (17.4) for the space portion of the earth images. In this case, Eqs. (17.3) and (17.4) lead to:

$$\mathrm{r}=\frac{1}{\mathrm{c}_{\alpha_s}},\ \mathrm{R_{IIRF}=R_SC_{\alpha_s}} \tag{17.6}$$

3.1.2.4 GEOS Coordinate Computation

In view of Fig. 3, we get:

$$\vec{\mathrm{R}}_\mathrm{IIRF0}=\vec{\mathrm{R}}_\mathrm{IIRF}+\Delta\vec{\mathrm{R}}_\mathrm{s} \tag{18.1}$$

$\vec{\mathrm{R}}_\mathrm{IIRF0}$ from Eqs. (14.4) and (17.3) or (17.6),

$$\Delta\vec{\mathrm{R}}_\mathrm{s}=\vec{\mathrm{R}}_\mathrm{s}-\vec{\mathrm{R}}_\mathrm{s0}=\begin{bmatrix}\mathrm{R_sC_{L_s}S_{\Delta\lambda_s}}\\ -\mathrm{R_sS_{L_s}}\\ \mathrm{R_{so}-R_sC_{Ls}C_{\Delta\lambda_s}}\end{bmatrix} \tag{18.2}$$

(E_{GEOS}, N_{GEOS}) are obtained from Figs. 2 and 3 and Eqs. (18.1) and (18.2):

$$\begin{bmatrix} C_{N_{GEOS}} S_{E_{GEOS}} \\ -S_{N_{GEOS}} \\ C_{N_{GEOS}} C_{E_{GEOS}} \end{bmatrix} = \begin{bmatrix} \widehat{R}_{GEOS0,x} \\ \widehat{R}_{GEOS0,y} \\ \widehat{R}_{GEOS0,z} \end{bmatrix} = \frac{\vec{R}_{IIRF} + \Delta\widehat{R}_s}{|\vec{R}_{IIRF} + \Delta\vec{R}_s|} \tag{18.3}$$

$$E_{GEOS} = \text{Arc tan}\left[\frac{\hat{R}_{GEOS0,x}}{\hat{R}_{GEOS0,z}}\right] \tag{18.4}$$

$$N_{GOES} = -\text{Arc sin}\,\hat{R}_{GEOS0,y} \tag{18.5}$$

$$Z = \left[\frac{E_{GEOS}}{N_{GOES}}\right] \tag{18.6}$$

Note that for star measurements, Z is obtained directly from Eq. (14.4) and Sect. 3.1.2.3 skipped because $\Delta\vec{R}_S$ is insignificant compared to $\vec{R}_{IIRF}$ in Eq. (18.3).

3.2 KF Initial Conditions

KF initial conditions are given by:
t_0 = epoch time = UTC_0 at KF start.
$x_0^+ = SV_{KF}$ at epoch = 0_{12+2m}

$$P_0^+ = \text{error covariance matrix at epoch} = \begin{bmatrix} P_{corr,0} & 0_{6\times6} & 0_{6\times2m} \\ 0_{6\times6} & P_{orb,0} & 0_{6\times2m} \\ 0_{2m\times6} & 0_{2m\times6} & P_{ma,0} \end{bmatrix} \tag{19.1}$$

$$\begin{aligned} P_{corr,0} &= \sigma^2_{corr,0}\begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, P_{orb,0} = \sigma^2_{orb,0}\begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \\ P_{ma,0} &= \sigma^2_{ma,0}\begin{bmatrix} I_{m\times m} & 0_{m\times m} \\ 0_{m\times m} & 0_{m\times m} \end{bmatrix} \end{aligned} \tag{19.2}$$

$\sigma_{corr,0} \cong \sigma_{orb,0} \cong \sigma_{ma,0} \cong 5.0E-05$ for simulation.

3.3 KF Detailed Computation

After level 1 A data block searched for landmarks and determined landmarks are time tagged, KF propagates (t_0, x_0^+, P_0^+) from last event prior to this data block and re-initialized after the(t_1, x_1^+, P_1^+) estimation as shown in Fig. 1. If no landmarks found within the data block, go to end of block of Eq. (21). Otherwise, let $\mathrm{LM_T}$ = total number of determined landmarks within the data block and do the following:

For k = 1 to $\mathrm{LM_T}$ do to **ENDFOR**

$$\Delta t = t_1 - t_0, t_1 = \mathrm{UTC_k} \text{ time at landmark number k.} \tag{20}$$

Propagation: From step number 1 of Sect. 3.
Estimation: From step number 2 of Sect. 3.
Re-initialize KF: From step number 3 of Sect. 3.
ENDFOR

At end of data block, do the following:

$$\Delta t = t_1 - t_0, t_1 = \mathrm{UTC_{end}} = \text{time at end of data block} \tag{21}$$

Propagation: From step number 1 of Sect. 3.
Re-initialize KF: $(t_0, x_0^+, P_0^+) = (t_1, x_1^+, P_1^+) = (t_1, x_1^-, P_1^-)$
Compute $\mathrm{SV_{INR}}$ from x_1^+ using Sect. 2. This is needed for Sect. 6.

If maneuver delta V provided by FDS

At maneuver, do the following:

$$\Delta t = t_1 - t_0, t_1 = \mathrm{UTC_{maneuver}} = \text{ maneuver time} \tag{22.1}$$

Propagation: From step number 1 of Sect. 3.
Re-initialize KF:

$$x_1^+ = x_1^- + \Delta x \tag{22.2}$$

$$P_1^+ = P_1^- + \Delta P \tag{22.3}$$

$$\Delta x = \left[0_{1\times 9} \frac{\Delta v_{FDS,r}}{R_{so}} \frac{\Delta v_{FDS,\lambda}}{R_{so}} \frac{\Delta v_{FDS,L}}{R_{so}} 0_{1\times 2m} \right]^T \tag{22.4}$$

$$\Delta P = \text{ diagonal terms 10 to 12 from delta v error analysis.}$$

$$(t_0, x_0^+, P_0^+) = (t_1, x_1^+, P_1^+) \tag{22.5}$$

Compute $\mathrm{SV_{INR}}$ from x_1^+ using Sect. 2. This is needed for Sect. 6.

If orbit is delta V provided by FDS instead of delta V

At maneuver, do the following:

$$\Delta t = t_1 - t_0, t_1 = \mathrm{UTC_{OD}} = \mathrm{orbitdeterminationtime} \tag{23.1}$$

Propagation: From step number 1 of Sect. 3.
Re-initialize KF:

$$x_1^+ = x_1^- + \delta\,x \tag{23.2}$$

$$P_1^+ = P_1^- + \delta\,P \tag{23.3}$$

$$\delta\,x = \begin{bmatrix} 0_{1\times 6} & \delta\left(\dfrac{\Delta R_s}{R_{so}}\right) & \delta\,\Delta\lambda_s & \delta\,L_s & 0_{1\times(3+2m)} \end{bmatrix}^T \tag{23.4}$$

$$\begin{aligned}(\delta\,\Delta R_s, \delta\,\Delta\lambda_s) = {} & (R_{FDS}, \Delta\lambda_{FDS}, L_{FDS})^- \text{ before OD} \\ & - (R_{FDS}, \Delta\lambda_{FDS}, L_{FDS})^+ \text{ after OD}\end{aligned} \tag{23.5}$$

$$\delta\,P = \text{diagonal terms 7 to 9 from OD error analysis.}$$

$$(t_0, x_0^+, P_)^+ = (t_1, x_1^+, P_1^+) \tag{23.6}$$

Compute SV_{INR} from x_1^+ using Sect. 2. This is needed for Sect. 6.

Transition matrix A: From Sect. 2.3.

Process noise covariance matrix Q: From Ref. [13], Eqs. (13)–(83) and (13)–(89), we get:

$$Q(\Delta t) = V_0 + V\Delta t + \frac{1}{2}[F_x V + V F_x^T]\Delta t^2 + \frac{1}{3} F_x V F_x^T \Delta t^3 \tag{24.1}$$

$$V_0 = \begin{bmatrix} V_{corr,0} & 0_{6\times 6} & 0_{6\times 2m} \\ 0_{6\times 6} & V_{orb,0} & 0_{6\times 2m} \\ 0_{2m\times 6} & 0_{2m\times 6} & V_{ma,0} \end{bmatrix},\ V_{y,0} = \begin{bmatrix} \sigma_{e,y}^2 I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} \tag{24.2}$$

$$V = \begin{bmatrix} V_{corr} & 0_{6\times 6} & 0_{6\times 2m} \\ 0_{6\times 6} & V_{orb} & 0_{6\times 2m} \\ 0_{2m\times 6} & 0_{2m\times 6} & V_{ma} \end{bmatrix},\ V_y = \begin{bmatrix} \sigma_{\nu,y}^2\, I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \sigma_{u,y}^2\, I_{3\times 3} \end{bmatrix} \tag{24.3}$$

where,

y = corr, orb, or ma. For ma, 3 replaced by m.
σ_e = measurement white noise standard deviation, rad.
σ_ν = random walk standard deviation, rad/sec$^{1/2}$.
σ_u = rate random walk standard deviation, rad/sec$^{3/2}$.

$$F_X = \begin{bmatrix} F_{corr} & 0_{6\times 6} & 0_{6\times 2m} \\ 0_{6\times 6} & F_{orb} & 0_{6\times 2m} \\ 0_{2m\times 6} & 0_{2m\times 6} & F_{ma} \end{bmatrix},\ F_{corr} = \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} \tag{25.1}$$

F_{orb} from Euler-Hill equations:

$$F_{orb} = \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ \omega_e^2\, F_{21} & 2\,\omega_e\, F_{22} \end{bmatrix} \cong \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} \tag{25.2}$$

$$\mathrm{F}_{21}=\begin{bmatrix}3&0&0\\0&0&0\\0&0&-1\end{bmatrix},\mathrm{F}_{22}=\begin{bmatrix}0&1&0\\-1&0&0\\0&0&0\end{bmatrix},\mathrm{F}_{\mathrm{ma}}=\begin{bmatrix}0_{\mathrm{mxm}}&\mathrm{I}_{\mathrm{mxm}}\\0_{\mathrm{mxm}}&0_{\mathrm{mxm}}\end{bmatrix}\tag{25.3}$$

This leads to:

$$\mathrm{Q}(\Delta \mathrm{t})=\begin{bmatrix}\mathrm{Q}_{\mathrm{corr}}&0_{6\times 6}&0_{6\times 2\mathrm{m}}\\0_{6\times 6}&\mathrm{Q}_{\mathrm{orb}}&0_{6\times 2\mathrm{m}}\\0_{2\mathrm{m}\times 6}&0_{2\mathrm{m}\times 6}&\mathrm{Q}_{\mathrm{ma}}\end{bmatrix}\tag{25.4}$$

$$\mathrm{Q}_{\mathrm{y}}=\begin{bmatrix}\left(\sigma_{\mathrm{e,y}}^{2}+\sigma_{\mathrm{v,y}}^{2}\,\Delta \mathrm{t}+\frac{1}{3}\,\sigma_{\mathrm{u,y}}^{2}\,\Delta \mathrm{t}^{3}\right)\mathrm{I}_{3\times 3}&\frac{1}{2}\,\sigma_{\mathrm{u,y}}^{2}\,\Delta \mathrm{t}^{2}\mathrm{I}_{3\times 3}\\\frac{1}{2}\,\sigma_{\mathrm{u,y}}^{2}\,\Delta \mathrm{t}^{2}\mathrm{I}_{3\times 3}&\sigma_{\mathrm{u,y}}^{2}\,\Delta \mathrm{t}\mathrm{I}_{3\times 3}\end{bmatrix}\tag{25.5}$$

where,

y = corr, orb, or ma. For ma, $\mathrm{I}_{3\times 3}$ is replaced by $\mathrm{I}_{\mathrm{mxm}}$.

Note that the first element of the above matrix is the same as in [13], Eq. (7)–(143).

The sigma values can be computed using Eq. (25.5), $\mathrm{SV}_{\mathrm{INR}}$ error analysis and estimate of time between measurements. For simulation, this leads to:

$(\sigma_{\mathrm{e,corr,}}\ \sigma_{\mathrm{e,orb,}}\ \sigma_{\mathrm{e,ma}}) = (1.942\mathrm{E}-07, 0, 0)$ rad.

$(\sigma_{\nu,\mathrm{corr}}, \sigma_{\nu,\mathrm{orb}}, \sigma_{\nu,\mathrm{ma}}) = (4.8\mathrm{E}-07, 0, 1.269\mathrm{E}-09)\ \mathrm{rad/s}^{1/2}$.

$(\sigma_{\mathrm{u,corr}}, \sigma_{\mathrm{u,orb}}, \sigma_{\mathrm{u,ma}}) = (4.774\mathrm{E}-10, 9.32\mathrm{E}-13, 2.318\mathrm{E}-11)\ \mathrm{rad/s}^{3/2}$.

Landmark measurement noise covariance matrix R:

$\mathrm{R} = \sigma_{\mathrm{M}}^{2}\,\mathrm{I}_{2\times 2}$

σ_{M} = sigma measurement noise calculated from landmark determination error analysis (=0.1 pixel for simulation).

Landmark location sensitivity matrix H:

H is determined from $(\frac{\partial Z}{\partial x})_{x=0}$ where Z is the estimated landmark measurement from Sect. 3.1.2 using the linear representation of $\mathrm{C}_{\mathrm{IIRF}}^{\mathrm{GEOS}}$ of Eq. (15.1). After some laborious algebraic manipulation, we get:

H = 2 × (12 + 2m) matrix given by:

$$\mathrm{H}=\left(\frac{\partial Z}{\partial x}\right)_{x=0}=[\mathrm{H}_{\mathrm{corr}}\quad \mathrm{H}_{\mathrm{orb}}\quad \mathrm{H}_{\mathrm{ma}}]\tag{26.1}$$

where,

$$\mathrm{H}_{\mathrm{corr}}=-\begin{bmatrix}\mathrm{T}_{\bar{\mathrm{N}}}\mathrm{S}_{\bar{\mathrm{E}}}&1&\mathrm{T}_{\bar{\mathrm{N}}}\mathrm{C}_{\bar{\mathrm{E}}}&0_{1\times 3}\\\mathrm{C}_{\bar{\mathrm{E}}}&0&-\mathrm{S}_{\bar{\mathrm{E}}}&0_{1\times 3}\end{bmatrix}\tag{26.2}$$

For Spacecraft x axis parallel to earth equator (e.g., COMS):

$$\mathrm{H}_{\mathrm{orb}}=-\begin{bmatrix}0&1&\mathrm{T}_{\bar{\mathrm{N}}}\mathrm{S}_{\bar{\mathrm{E}}}&0_{1\times 3}\\0&0&\mathrm{C}_{\bar{\mathrm{E}}}&0_{1\times 3}\end{bmatrix}+\bar{\mathrm{r}}\begin{bmatrix}\frac{\mathrm{S}_{\bar{\mathrm{E}}}}{\mathrm{C}_{\bar{\mathrm{N}}}}&\frac{\mathrm{C}_{\bar{\mathrm{E}}}}{\mathrm{C}_{\bar{\mathrm{N}}}}&0&0_{1\times 3}\\\mathrm{C}_{\bar{\mathrm{E}}}\mathrm{S}_{\bar{\mathrm{E}}}&-\mathrm{S}_{\bar{\mathrm{E}}}\mathrm{S}_{\bar{\mathrm{N}}}&\mathrm{C}_{\bar{\mathrm{N}}}&0_{1\times 3}\end{bmatrix}\tag{26.3}$$

For Spacecraft x axis parallel to orbit plane (e.g., GOES I-M):

Replace $\begin{bmatrix} 0 & 1 & T_{\bar N}S_{\bar E} & 0_{1\times 3} \\ 0 & 0 & C_{\bar E} & 0_{1\times 3} \end{bmatrix}$ by $\begin{bmatrix} 0 & 1 & T_{\bar N}S_{\bar E} & 0_{1\times 2} & T_{\bar N}C_{\bar E}/\,\omega_e \\ 0 & 0 & C_{\bar E} & 0_{1\times 2} & -S_{\bar E}/\,\omega_e \end{bmatrix}$

$$\bar r = \left(C_{\bar\alpha_s} - \sqrt{C^2_{\bar\alpha_s} - 1 + ((R_e + h)/R_{so})^2}\right)^{-1} \tag{26.4}$$

$$C_{\bar\alpha_s} = \mathrm{Cos}\bar\alpha_s = C_{\overline N} = C_{\overline E} \tag{26.5}$$

$$T_{\overline N} = \mathrm{Tan}\overline N_{GEOS},\ S_{\overline N} = \mathrm{Sin}\overline N_{GEOS},\ C_{\overline N} = \mathrm{Cos}\overline N_{GEOS}$$

$$S_{\overline E} = \mathrm{Sin}\overline E_{GEOS}\ ,\ C_{\overline E} = \mathrm{Cos}\overline E_{GEOS}$$

$(\overline E_{GEOS}\ ,\ \overline N_{GEOS})$ = landmark location from Eq. (13.4).

R_e = earth radius at landmark location from Eq. (12.3).

h = landmark altitude.

For star measurements, r = 0 and H_{orb} becomes insensitive to orbit translational part ($\delta R/R_{so}$, $\delta\lambda$, δL). Therefore, stars cannot be used to refine orbit and, therefore, orbit refinement must be deleted from KF as described in Sect. 5.1.

$$H_{ma} = \begin{bmatrix} C_{11} & C_{12} & 0 & 0 \\ C_{21} & C_{22} & 0 & 0 \end{bmatrix} \tag{26.6}$$

$$C_{11} = -\frac{C_{22}}{C_{\overline N}}\ ,\ C_{12} = -\frac{C_{21}}{C_{\overline N}}\ ,\ C_{21} = -\frac{S_{\overline E} - C_{\overline N}}{1 - C_{\overline N}S_{\overline E}}\ ,\ C_{22} = -\frac{S_{\overline N}C_{\overline N}}{1 - C_{\overline N}S_{\overline E}} \tag{26.7}$$

H_{ma} for two mirror imaging systems to be investigated in the future.

3.4 Thermo-Elastic Model Time Series

The thermo-elastic $SV_{ma,model}$ and $SV_{corr,model}$ time series can be obtained form Eqs. (2.1)–(3.2) as follows:

1. Create daily time series at, e.g., one-minute interval for, e.g., seven days using interpolation of SV_{ma} and SV_{corr} data at time $t_{i,n}$, $i = 1, 2, \ldots, 1440$ and $n = 1, 2 \ldots, 7$.
2. The $SV_{ma,model}(t_{i,n})$ and $SV_{corr,model}(t_{i,n})$ for the next day (n = 8) are obtained by averaging the last seven days of $SV_{ma}(t_{i,n})$ and $SV_{corr}(t_{i,n})$ data:

$$SV_{ma,model}(t_{i,8}) = \frac{1}{7}\sum\nolimits_{n=1}^{n=7} SV_{ma}(t_{i,n}) \tag{27.1}$$

$$SV_{corr,model}(t_{i,8}) = \frac{1}{7}\sum\nolimits_{n=1}^{n=7} SV_{corr}(t_{i,n}) \tag{27.2}$$

Note that $\mathrm{SV}_{\mathrm{ma,model}}$ and $\mathrm{SV}_{\mathrm{corr,model}}$ are initially determined by analysis or set to zero.

3. Repeat above process once a day using one-day sliding window.

4 Simulation Results

A hundred landmarks distributed over earth and COMS imaging schedule were used in the simulation [14, 15]. The true $\overline{\mathrm{SV}}_{\mathrm{INR}}$ is calculated using eccentricity =0.0001, inclination = 0.05°, $\mathrm{SV}_{\mathrm{ma,model}}$ and $\mathrm{SV}_{\mathrm{corr,model}}$ amplitudes = 100 μrad, with 24-h period, and attitude amplitude = 300 μrad with 2.4-h period. The maneuver delta V times are obtained from [16], Fig. 8, and magnitudes from [17], Tables 2.

The estimated $\mathrm{SV}_{\mathrm{INR}}$ are shown in Fig. 4 for seven days and is computed using Sect. 3.3 based on $\mathrm{SV}_{\mathrm{ma,model}}$ and $\mathrm{SV}_{\mathrm{corr,model}}$ errors =10 μrad, FDS maneuver delta V errors from [17], Table 3 (or FDS orbit determination error from Table 7). Figure 5 shows $\mathrm{SV}_{\mathrm{errors}}$, $\delta\,\mathrm{SV} = \overline{\mathrm{SV}}_{\mathrm{INR}} - \mathrm{SV}_{\mathrm{INR}}$. Figure 6 shows residual errors computed using Sect. 3.1. The simulated landmarks are obtained using the true $\overline{\mathrm{SV}}_{\mathrm{INR}}$ to transfer $(\bar{\mathrm{E}}_{\mathrm{GEOS}}, \overline{\mathrm{N}}_{\mathrm{GEOS}})$ to $(\bar{\mathrm{E}}_{\mathrm{LRF}}, \overline{\mathrm{N}}_{\mathrm{LRF}})$ based on Sect. 3.1.2 inverse transformation. The estimated $(\mathrm{E}_{\mathrm{LRF}}, \mathrm{N}_{\mathrm{LRF}})$ are then obtained from actual $(\bar{\mathrm{E}}_{\mathrm{LRF}}, \bar{\mathrm{N}}_{\mathrm{LRF}})$ by adding a random normal distribution land- mark determination error with $\sigma_{\mathrm{M}} = 2.8E\text{-}06\,\mathrm{rad}$ for visible landmarks and $\sigma_{\mathrm{M}} = 11.2E\text{-}06\,\mathrm{rad}$ for IR landmarks. Simulation was also successfully used to stress test the basic method for cases using eccentricity = 0.001, inclination = 0.5°, $\mathrm{SV}_{\mathrm{ma,model}}$ and $\mathrm{SV}_{\mathrm{corr,model}}$ amplitudes = 1000 μrad with errors =100 μrad and only IR landmarks.

5 Adaptation to Other Systems

The following 3 subsections show how the basic method described in Fig. 1 can be adapted to be used for systems based on star and landmark measurements, star only measurements with orbit from FDS or GPS, and systems with attitude rate telemetry inserted in the image wideband data.

5.1 Systems Based on Star and Landmark Measurements

For stars:

- KF refinements $(\delta\,\mathrm{R_s}/\mathrm{R_{so}},\ \delta\,\lambda_{\mathrm{s}},\ \delta\,\mathrm{L}_s)$ = (0,0,0) and $(\mathrm{x_{orb}}, \dot{\mathrm{x}}_{\mathrm{orb}})$ deleted from KF state vector.
- Rows and columns associated with $\mathrm{P_{orb,0}}$, $\mathrm{A_{orb}}$, $\mathrm{P_{orb}}$ and $\mathrm{F_{orb}}$ deleted from P_0^+, A, V, $\mathrm{V_0}$, and $\mathrm{F_x}$.
- $\mathrm{H_{orb}}$ deleted from H.

  In this case:

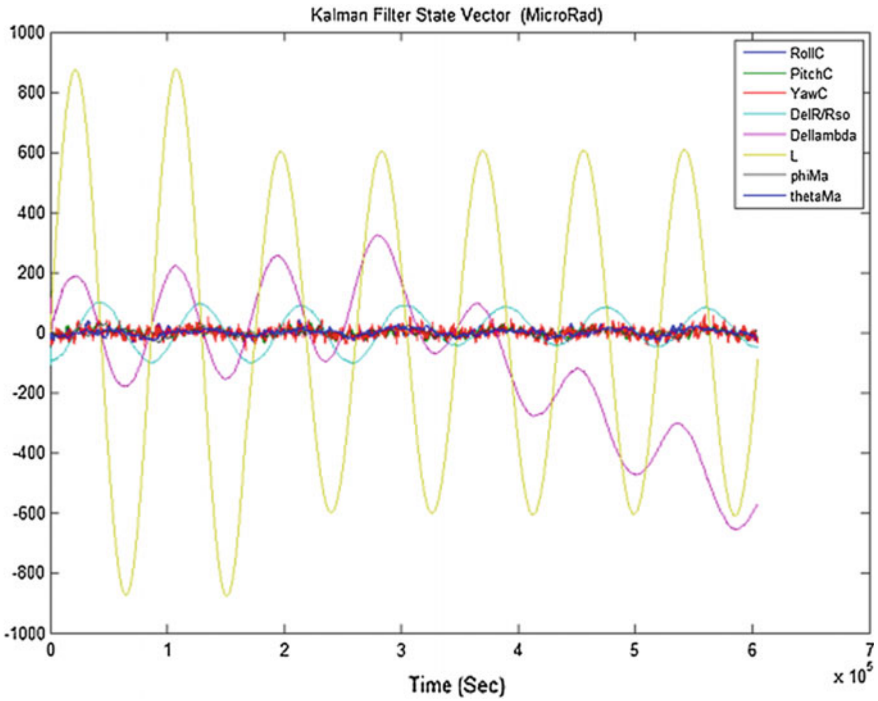


Fig. 4 KF state vector

- one star per minute or 3 stars per 5 min are sufficient to determine KF x_{corr} and x_{ma}.
- SV_{KF} dimension $= 6 + 2m$ instead of $12 + 2m$.
- KF detailed computation is like basic method with landmarks replaced by stars. For landmarks:
- Use KF for orbit refinements ($\delta R_s/R_{so}$, $\delta \lambda_s$, δL_s) using the above deleted items. In this case:
- Few landmarks (e.g.,10 well distributed landmarks over earth) are sufficient to determine (x_{orb}, $\dot{x}_{orb}$).
- SVKF dimension $= 6$ instead of $12 + 2m$.

5.2 Systems Based on Star only Measurements

- KF is same as in Sect. 5.1 for stars. In this case, the orbit must be provided by FDS or GPS.

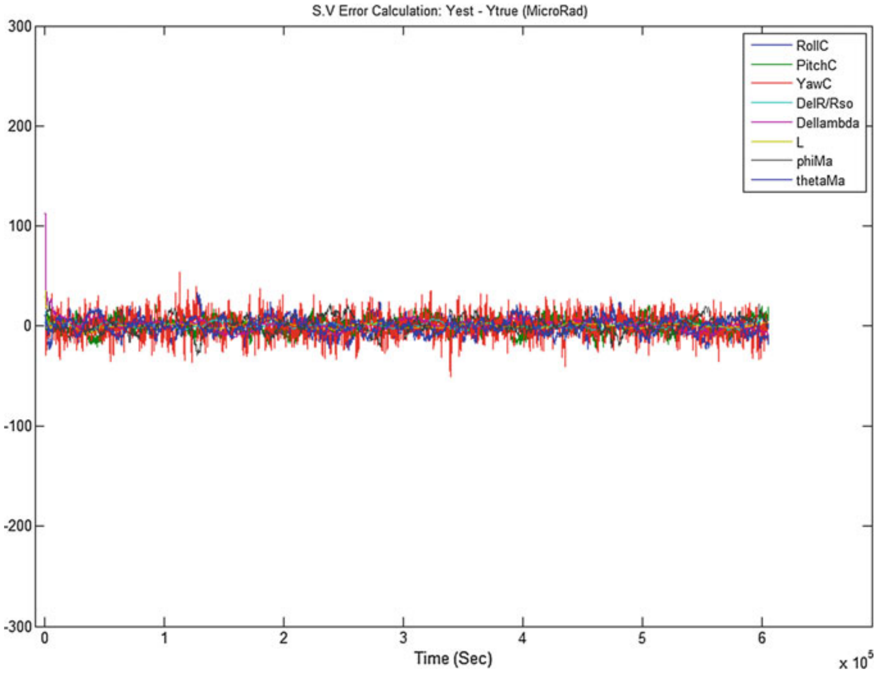


Fig. 5 SV errors $\delta\,\mathrm{SV} = \mathrm{SV}_{\mathrm{INR}} - \mathrm{SV}_{\mathrm{INR}}$

5.3 Systems Based on Inertial Angular Rate Telemetry

The (ω_{sx}, ω_{sy}, ω_{sz}) telemetry represent inertial angular rate along ACF axes in the form of time series spaced at Δt_{att} (e.g., 0.01 s) inserted in the imager wideband data. The rates $\mathrm{SV}_{\mathrm{ACF}}$ of Eq. (15.3) can be obtained from (ω_{sx}, ω_{sy}, ω_{sz}) using Fig. 3 with IIRF replaced by ACF. Starting with $\dot{\theta} + \omega_e$ about $-\mathrm{Y}_{\mathrm{GEOS}}$ axis followed by $\dot{\phi}$ about the new −X axis followed by $\dot{\psi}$ about $-\mathrm{Z}_{\mathrm{ACF}}$ axis and using Eqs. (14.5) and (15.1), we get:

$$\begin{bmatrix}\omega_{sx}\\ \omega_{sy}\\ \omega_{sz}\end{bmatrix} = -\dot{\psi}\begin{bmatrix}0\\ 0\\ 1\end{bmatrix} - \dot{\phi}\begin{bmatrix}C_\psi\\ S_\psi\\ 0\end{bmatrix} - (\dot{\theta} + \omega_e)\begin{bmatrix}S_\psi C_\phi\\ C_\psi C_\phi\\ S_\phi\end{bmatrix} \tag{28.1}$$

This leads to:

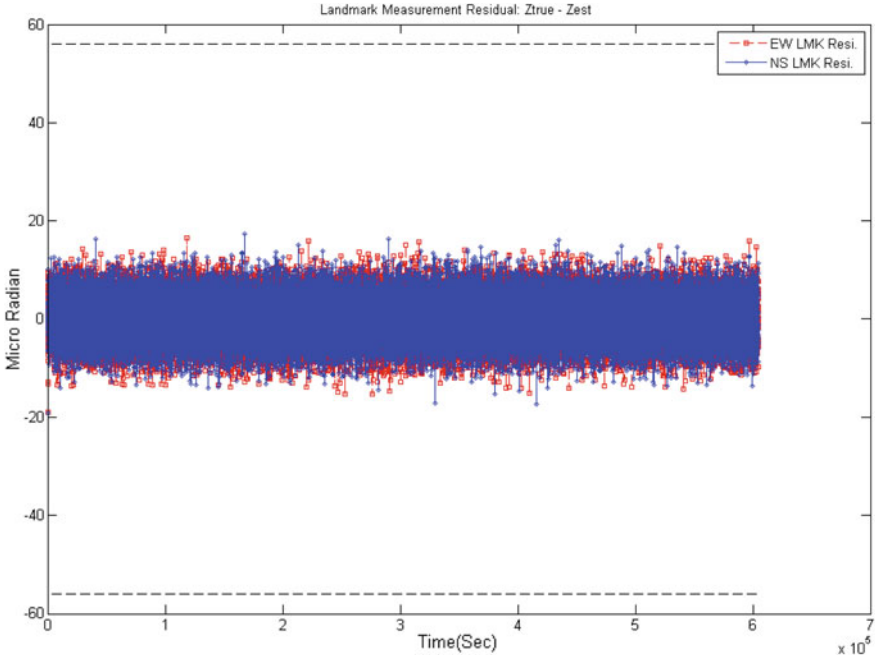


Fig. 6 Landmarks measurement residuals

$$
\mathrm{S\dot{V}_{ACF}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = -\begin{bmatrix} \omega_{sy}\mathrm{S}\,\psi + \omega_{sx}\mathrm{C}\,\psi \\ \omega_{e} + (\omega_{sy}\mathrm{C}\,\psi - \omega_{sx}\mathrm{S}\,\psi)/\mathrm{C}\,\phi \\ \omega_{sx} - (\omega_{sy}\mathrm{C}\,\psi - \omega_{sx}\mathrm{S}\,\psi)\mathrm{S}\,\phi\,/\mathrm{C}\,\phi \end{bmatrix}
$$

$$
\cong -\begin{bmatrix} \omega_{sx} + \omega_{sy}\,\psi \\ \omega_{sy} + \omega_{e} - \omega_{sx}\,\psi \\ \omega_{sz} - \omega_{sy}\,\phi \end{bmatrix} \tag{28.2}
$$

Note that Eq. (28.2) can also be obtained from [13], Appendix E Table E-2 for 2, 1, 3 type rotation by replacing (ϕ, ψ) with $-(\phi, \psi)$ and $(\omega_I, \omega_J, \omega_K)$ with $(\omega_{sy}, \omega_{sx}, \omega_{sz})$ on the right side of the equation and replacing $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ with $-(\dot{\theta} + \omega_e, \dot{\phi}, \dot{\psi})$ on the left side of the equation.

Now, the $\mathrm{SV_C}$ time series spaced by Δt_{att} over $\Delta t = t_1 - t_0$ is computed as follows: Let $j = \mathrm{Integer}(\Delta t/\Delta tatt)$

For $i = 1, \ldots, j$ plus final step from τ_j to t_1

$$
\mathrm{SV_{ACF}}(\tau_i) = \mathrm{SV_{ACF}}(\tau_{i-1}) + \mathrm{S}\dot{V}_{\mathrm{ACF}}(\tau_{i-1})\Delta t_{att} \tag{28.3}
$$

where

$$
\mathrm{S\dot{V}_C}(\tau_{i-1}) = \mathrm{S\dot{V}_{ACF}}(\tau_{i-1}) + \mathrm{S\dot{V}_{corr,model}}(\tau_{i-1}) + \dot{x}\mathrm{corr}(\tau_{i-1}) \tag{28.4}
$$

With

$$\mathrm{SV_C(\tau_i) = SV_{ACF}(\tau_i) + SV_{Corr,model}(\tau_i) + x_{corr}(\tau_i)} \tag{28.5}$$

$\mathrm{SV_{corr,model}}$ obtained from Sect. 3.4 and $\mathrm{x_{corr}}$ determined by KF and defined in Eqs. (8.2) and (8.3).

At KF start (see Sects. 3.2 and 3.4):

$$\mathrm{SV_{KF} = 0_{12+2m} \text{ and } SV_{corr,model} = 0_3} \tag{29.1}$$

In view of Eqs. (15.2), (15.3), and (6.1) to (6.3)

$$\mathrm{SV_C(0) = SV_{ACF}(0) = SV_{ORF}(0) + SV_{att}(0)} \tag{29.2}$$

$$\mathrm{SV_{att}(0) \text{ from telemetry or } = 0_3.} \tag{29.3}$$

At KF re-initialization (see Fig. 1):

$$\mathrm{SV_C(0) = SV_C(t_1) \text{ from Eq. } (28.4)} \tag{29.4}$$

$$\mathrm{SV_{ACF}(0) = SV_{ACF}(t_1) \text{ from Eq. } (28.3)} \tag{29.5}$$

$$\mathrm{S\dot{V}_{ACF}(0) = S\dot{V}_{ACF}(t_1) \text{ from Eq. } (28.2)} \tag{29.6}$$

Note that $\mathrm{SV_{ACF}}$ of Eq. (28.4) is corrected by $\mathrm{x_{Corr}}$ determined by KF to compensate for gyro drift and the first part of H_{orb} in Eq. (26.3) must be deleted.

6 Image Registration Using Resampling

Image registration requires two steps. The first step is to transfer level 1A (column, line) pixel indices (c, ℓ) to (c′, ℓ′) = (c, ℓ) + (Δc, $\Delta\ell$)$_{,\ell}$ in the GEOS fixed frame. The second step is to resample (c′, ℓ′) pixels to generate level 1B data block (see, e.g., [18]). The first step can be performed using Sect. 3.1.2 algorithm to determine ($\Delta\ell$, $\Delta\ell$)$_{c,\ell}$ from $(\Delta\mathrm{E}, \Delta\mathrm{N})_{c,\ell} = (\mathrm{E_{GEOS}, N_{GEOS}})_{c,\ell} - (\mathrm{E_{LRF}, N_{LRF}})_{c,\ell}$ divided by pixel size. Index c and N_{LRF} are fixed over pixel line and index c and $\mathrm{E_{LRF}}$ are fixed over pixel column. The $\mathrm{SV_{INR}}$ time series needed for (c, ℓ) to (c′, ℓ′) transformation is obtained from Sect. 2.3. The processing time of this transformation is significantly reduced by computing $(\Delta C, \Delta L)_C, L$ for a subset of pixels (C, L) uniformly distributed over the level 1A (c, ℓ) array. The $(\Delta C, \Delta L)_C, L$ for the remaining pixels are then computed by EW and NS linear interpolation between the $(\Delta C, \Delta L)_{C,L}$ subset. Note that the number of (C, L) pixel subset is obtained by analysis of $\mathrm{SV_{INR}}$ and attitude jitter effects on registration error. Note also that level 1 A block should be slightly larger than level 1B block to account for the shift caused by orbit, attitude, and misalignment variation over time.

7 Conclusion

INR and KF state vectors suitable for the new INR method were defined. The basic method is based on landmark measurements to determine orbit, attitude correction angles, and imager misalignments with maneuvers delta V (or orbit with coarse accuracy) provided by FDS. The method was proven by simulation. Adaptation of this method to other INR systems and an algorithm for transferring pixels from LOS to the fixed grid GEOS frame needed for pixel data resampling are presented.

Acknowledgements The authors appreciate the support of Eun-joo Kwon and J.B. Park during the simulation effort.

References

1. Kamel AA (1996) GOES image navigation and registration system. SPIE 2812:766–776
2. Lu F et al (2008) Image Navigation for the FY2 geosynchronous meteorological satellite. J Atmos Ocean Tech 25:1149–1165. Accessed 1 July 2008
3. Woo J et al (2011) Analysis of COMS MI image navigation and registration performance, ISRS. Accessed 2–4 Nov 2011
4. Chambon T et al (2013) On-ground evaluation of MTG image navigation and registration (INR) performances. In: Proceedings of SPIE 8889, sensors, systems, and next- generation satellites, vol XVII, p 88891J. Accessed 16 Oct 2013
5. Okuyama A et al (2015) Preliminary validation of Himawari-8/AHI navigation and calibration. Accessed 9 Nov 2015
6. Kamel AA (1999) Precise spacecraft camera image navigation and registration, US patent #5,963,166, 5 Oct 1999
7. Carr JL (2009) Twenty-five years of INR. J Astronaut Sci 57:505–515
8. Harris J et al (2009) Image navigation and registration improvements using GPS. In: Geoscience and remote sensing symposium, 2009 IEEE international. IGARSS, vol 3, pp 247–250
9. De Luccia FJ et al (2016) Image navigation and registration performance assessment tool set for the GOES-R advanced baseline imager and geostationary lightning mapper. Accessed 4 April 2016
10. GOES Earth Location User's Guide (ELUG), DRL 504-11 Revision 1, Appendix C, NOAA, March 1998
11. Virgilio VN (2015) Geolocation of remotely sensed pixels by introspective landmarking, US 8942421, 27 Jan 2015
12. Sidi MJ (1997) Spacecraft dynamics and control. Cambridge University Press, Cambridge
13. Wertz JR (1978) Spacecraft attitude determination and control. D. Reidel Publishing, Hingham (1978) (reprinted in 2000 by Kluwer Academic Publisher. Mass, Boston)
14. Final technical report: as part of the deliverables of the contract for technical COMS technical monitoring and services, Contract No. KARI-13-0007. Accessed 18 April 2014
15. Final technical report: as part of the deliverables of the contract for technical COMS technical monitoring and services, Contract No. KARI-15-0012. Accessed 24 April 2015
16. Lee BS et al (2010) East-west station-keeping maneuver strategy for COMS satellite using iterative process. Advances in space research, the official journal of the committee on space research (COSPAR)
17. Hwang Y et al (2008) Orbit determination accuracy improvement for geostationary single station antenna tracking. ETRI J 30(6):774–782
18. Ormiston JP et al (2008) GOES advanced baseline imager – ground processing development system, 5th GOES users' conference. Accessed 23 Jan 2008

Exoatmospheric DACS Type Missile Controller Based on Sliding Mode Control Considering the Seeker's Field-of-View Limit

Jaeho Lee and Youdan Kim

1 Introduction

The antiballistic missile system aims to defend against a ballistic missile by intercepting the target at a high altitude. Especially, in exoatmospheric environment at which the altitude is greater than 100 km, the aerodynamic control using the missile's fin is not applicable, and therefore thrust based control devices such as divert and attitude control system (DACS) should be used. DACS, which is a thruster system mainly used in a spacecraft system, is composed of a divert control system (DCS) using thrusters located at the missile's center of mass and an attitude control system (ACS) using thrusters located at the missile's tail [12]. Because the guidance and control methods of the missile based on the aerodynamics cannot be applied in exoatmospheric environment, guidance and control approaches [1, 4, 8, 10] and analysis [3] for the exoatmospheric missile have been studied. Especially, the exoatmospheric controller design using the output feedback for the DACS type missile was introduced [1], and the control algorithm and thruster management function taking into account the characteristics of the solid propulsion DACS were proposed [4].

During the terminal phase of the missile, target information is obtained by the seeker for the homing guidance. If the missile has the strapdown seeker, which is mounted on the missile's body, it is crucial to maintain the lock-on condition until intercepting the target. Therefore, there have been lots of research on the guidance of the missile considering the seeker's field-of-view limit to maintain the lock-on condition. Switching approaches, which combine the pure proportional navigation guidance (PPNG) law and schemes holding [9] or decreasing [6] the look angle, were proposed. Impact angle [2, 5] and time [13] control guidance laws considering the seeker's field-of-view limit were also presented. However, these studies were developed using a point mass model of the missile in the 2-dimensional pursuit situation, and consequently the look angle is defined as an angle between the velocity

J. Lee (✉) · Y. Kim
Seoul National University, Seoul 08826, Republic of Korea
e-mail: akfksem@snu.ac.kr

Y. Kim
e-mail: ydkim@snu.ac.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_34

and line-of-sight (LOS) vectors. Considering that the look angle is actually presented as the angle between the x-axes of the missile's body and LOS coordinate systems in the 3-dimensional pursuit situation, the previous studies may not properly deal with the seeker's narrow field-of-view.

In this study, an exoatmospheric DACS type missile controller is designed based on the sliding mode control scheme considering the strapdown seeker's narrow field-of-view in the 3-dimensional pursuit situation. The look angle of the seeker is defined as the angle between the x-axes of the missile's body and LOS coordinate system. It is assumed that the DACS's configuration consists of four thrusters of DCS and six thrusters of ACS, and the missile's 6-degrees-of-freedom (6-DOF) equations of motions are derived. Based on the property of the exoatmospheric missile that the translational motion is decoupled from the rotational motion, the design objectives of the controller are to maintain the lock-on condition until intercepting the target and to generate the accelerations corresponding to the guidance command. To achieve the two objectives, the controller is designed as the two-loop structure; the outer-loop controller generates the pitch and yaw angular rate commands to decrease the look angle using the information of the seeker. In the inner-loop controller, the ACS thrust inputs corresponding to the angular rate commands and stabilizing the roll motion are generated, and the DCS thrust inputs corresponding to the acceleration commands sent by the guidance loop are generated. The controllers of each loop are designed based on the first-order sliding mode control scheme. To demonstrate the performance of the proposed controller, numerical simulation is carried out for the DACS type missile to intercept the ballistic target in the exoatmospheric environment.

The main features of this paper are as follows. First, the look angle of the seeker is defined in the 3-dimensional pursuit situation, which is more practical than the other previous studies. Second, using the characteristics of the exoatmospheric environment, the DACS type controller maintaining the lock-on condition and generating the accelerations corresponding to the guidance command is designed based on the sliding mode control scheme. Finally, guidance and control integrated simulation using the PPNG law and designed controller is performed for the DACS type missile to intercept the ballistic target in the 3-dimensional exoatmospheric pursuit situation.

The paper is organized as follows. In Sect. 2, preliminaries including the problem statement and the 6-DOF equations of motions for the DACS type missile are presented. The controller design is described in Sect. 3, and numerical simulation results are provided in Sect. 4. Finally, conclusions are given in Sect. 5.

2 Preliminaries

In this study, five types of coordinate systems are used for the 3-dimensional pursuit situation as shown in Fig. 1, where the superscripts $[I]$, $[L]$, $[V_M]$, $[V_T]$, and $[B]$ mean the reference, LOS, missile's velocity, target's velocity, and missile's body coordinate systems, respectively, and $\psi_{A,B}$ and $\theta_{A,B}$ are the Euler angles from A coordinate system to B coordinate system represented by the sequence of z and

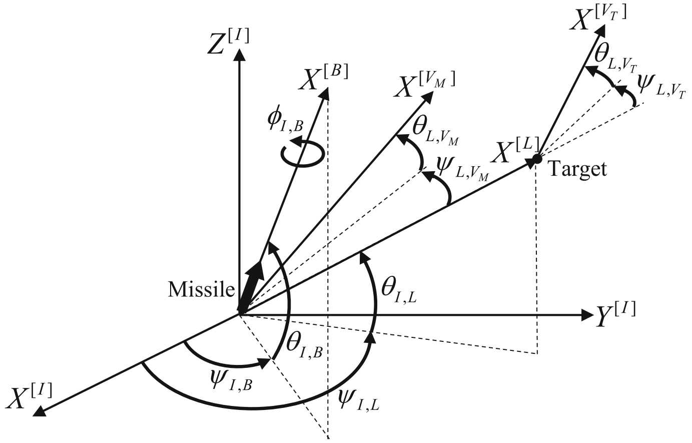
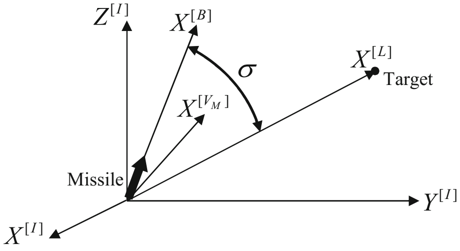


Fig. 1 Pursuit geometry in 3-dimensional space

Fig. 2 Definition of seeker's look angle

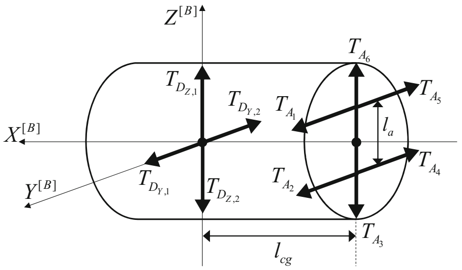


negative y axes. Specifically, $\psi_{I,B}$, $\theta_{I,B}$, and $\phi_{I,B}$ are the Euler angles from the reference coordinate system to the missile's body coordinate system represented by the sequence of z, negative y, and x axes. The target is considered as the point mass.

2.1 Problem Statement

The main objective of this study is to design a controller for the exoatmospheric DACS type missile equipped with the strapdown seeker of the narrow field-of-view. Considering that the translational motion of the missile can be separated from the rotational motion in the exoatmospheric environment, the controller should be designed; to maintain the lock-on condition of the seeker and to generate the missile accelerations corresponding to the guidance command. The look angle of the seeker σ is defined as the magnitude of the angle between the x-axes of the missile's body and LOS coordinate systems as shown in Fig. 2.

Fig. 3 Thruster configuration of DACS



2.2 Equations of Motions

A configuration of the DACS considered in this study is shown in Fig. 3, where $T_{D_{Y,1}}$, $T_{D_{Y,2}}$, $T_{D_{Z,1}}$, and $T_{D_{Z,2}}$ are the four thrusters of the DCS, T_{A_1}, T_{A_2}, T_{A_3}, T_{A_4}, T_{A_5}, and T_{A_6} are the six thrusters of the ACS, l_{cg} is the length from the ACS to the missile's center of mass, and l_a is the length between the side thrusters of the ACS. The pitch and yaw accelerations of the missile's body can be expressed as follows

$$a_z = \frac{T_{D_Z} - T_{A_3} + T_{A_6}}{m} \tag{1}$$

$$a_y = \frac{T_{D_Y} + T_{A_1} + T_{A_2} - T_{A_4} - T_{A_5}}{m} \tag{2}$$

where a_z and a_y are the pitch and yaw accelerations of the missile's body, respectively, m is the mass of the missile, and T_{D_Z} and T_{D_Y} are defined as

$$T_{D_Z} = T_{D_{Z,1}} - T_{D_{Z,2}} \tag{3}$$

$$T_{D_Y} = T_{D_{Y,1}} - T_{D_{Y,2}} \tag{4}$$

The rotational motion of the missile can be represented as follows

$$\dot{\omega}^{[B]} = \mathbf{I}^{-1}(\mathbf{M}^{[B]} - \omega^{[B]} \times \mathbf{I}\omega^{[B]}) \tag{5}$$

where $\omega^{[B]} = [p\ q\ r]^T$ is the angular rate vector with respect to the missile's body coordinate system, $\mathbf{I}$ is the moment of inertia matrix about the center of mass, and $\mathbf{M}^{[B]} = [L\ M\ N]^T$ is the external moment vector acting on the center of mass with respect to the missile's body coordinate system. The external moments are derived as

$$L = \frac{1}{2} T_{A_L} l_a \tag{6}$$

$$M = T_{A_M} l_{cg} \tag{7}$$

$$N = T_{A_N} l_{cg} \tag{8}$$

where T_{A_L}, T_{A_M}, and T_{A_N} are defined as

$$T_{A_L} = -T_{A_1} + T_{A_2} - T_{A_4} + T_{A_5} \tag{9}$$

$$T_{A_M} = -T_{A_3} + T_{A_6} \tag{10}$$

$$T_{A_N} = -T_{A_1} - T_{A_2} + T_{A_4} + T_{A_5} \tag{11}$$

Assuming that the missile has the axially symmetric shape, the moment of inertia matrix can be expressed as

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{12}$$

where I_x, I_y, and I_z are principal moments of inertia about the center of mass. Substituting Eqs. (6)–(12) into Eq. (5), the rotational dynamics of the missile can be obtained as follows

$$\dot{p} = \frac{T_{A_L} l_a}{2I_x} \tag{13}$$

$$\dot{q} = \frac{(I_z - I_x)pr + T_{A_M} l_{cg}}{I_y} \tag{14}$$

$$\dot{r} = \frac{(I_x - I_y)pq + T_{A_N} l_{cg}}{I_z} \tag{15}$$

The relation between the Euler angles from the reference coordinate system to the missile's body coordinate system with the sequence of z, $-y$, and x axes and the body angular rates can be formulated as follows

$$\dot{\phi}_{I,B} = p - q \sin\phi_{I,B} \tan\theta_{I,B} - r \cos\phi_{I,B} \tan\theta_{I,B} \tag{16}$$

$$\dot{\theta}_{I,B} = q \cos\phi_{I,B} - r \sin\phi_{I,B} \tag{17}$$

$$\dot{\psi}_{I,B} = q \sin\phi_{I,B} \sec\theta_{I,B} + r \cos\phi_{I,B} \sec\theta_{I,B} \tag{18}$$

3 Controller Design

The translational motion of the missile can be separated from the rotational motion, and therefore, in this study, the controller is designed to achieve the following two objectives; (i) to maintain the lock-on condition, and (ii) to generate the accelerations corresponding to the guidance command. For this, the proposed controller has two-loop structure as shown in Fig. 4, where the outer-loop controller computes the pitch and yaw angular rate commands decreasing the look angle of the seeker to maintain the lock-on condition by using the seeker's information. The inner-loop computes

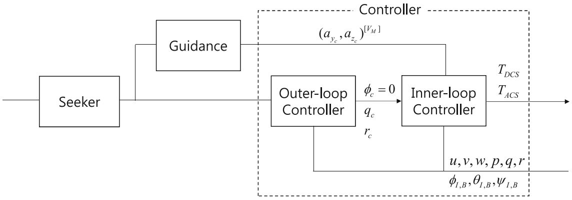


Fig. 4 Two-loop structure of the controller

the ACS inputs to track the angular rate commands of the outer-loop and to make the roll angle zero for the stabilization of the roll motion. The inner-loop also computes the DCS inputs to generate the missile's body accelerations corresponding to the guidance command. The controllers of each loop are designed based on the first-order sliding mode control scheme, and the guidance command of the pitch and yaw accelerations with respect to the missile's velocity coordinate system is generated by using the PPNG law.

3.1 Inner-Loop Controller

3.1.1 ACS Control Input

The design objective of the inner-loop controller is to track the commands of the pitch and yaw angular rates and roll angle. The commands of the pitch and yaw angular rates are generated by the outer-loop controller, and the roll angle command is set to zero for the stabilization of the roll motion. The error vector of the inner-loop is defined as

$$e_I = \begin{bmatrix} e_\phi \\ e_q \\ e_r \end{bmatrix} = \begin{bmatrix} \phi_{I,B} \\ q \\ r \end{bmatrix} - \begin{bmatrix} \phi_c \\ q_c \\ r_c \end{bmatrix} \tag{19}$$

where ϕ_c, q_c, and r_c are the commands of the roll angle and pitch and yaw angular rates, respectively. Let us consider the sliding surfaces of the inner-loop including the integral terms for regulating the error vector as

$$s_\phi = \left(\frac{d}{dt} + \lambda_\phi\right)^{r_\phi} \int_0^t e_\phi d\tau \tag{20}$$

$$s_q = \left(\frac{d}{dt} + \lambda_q\right)^{r_q} \int_0^t e_q d\tau \tag{21}$$

$$s_r = \left(\frac{d}{dt} + \lambda_r\right)^{r_r} \int_0^t e_r d\tau \tag{22}$$

where λ_ϕ, λ_q, and λ_r are the design parameters of the positive real values, and r_ϕ, r_q, and r_r are the relative degrees for the sliding mode control scheme. Setting $r_\phi = 2$ and $r_q = r_r = 1$, Eqs. (20)–(22) can be rewritten as follows

$$s_\phi = \left(\frac{d^2}{dt^2} + 2\lambda_\phi \frac{d}{dt} + \lambda_\phi^2\right)\int_0^t e_\phi d\tau = \dot{e}_\phi + 2\lambda_\phi e_\phi + \lambda_\phi^2 \int_0^t e_\phi d\tau \tag{23}$$

$$s_q = \left(\frac{d}{dt} + \lambda_q\right)\int_0^t e_q d\tau = e_q + \lambda_q \int_0^t e_q d\tau \tag{24}$$

$$s_r = \left(\frac{d}{dt} + \lambda_r\right)\int_0^t e_r d\tau = e_r + \lambda_r \int_0^t e_r d\tau \tag{25}$$

Differentiating Eq. (23) with respect to time and substituting Eq. (16) and its time derivative into the resulting equation yields

$$\begin{aligned}\dot{s}_\phi &= \dot{p} - \dot{q}\sin\phi_{I,B}\tan\theta_{I,B} - \dot{\phi}_{I,B}(q\cos\phi_{I,B}\tan\theta_{I,B} - r\sin\phi_{I,B}\tan\theta_{I,B})\\ &- \dot{r}\cos\theta_{I,B}\tan\theta_{I,B} - \dot{\theta}_{I,B}(q\sin\phi_{I,B}\sec^2\theta_{I,B} + r\cos\phi_{I,B}\sec^2\theta_{I,B}) + \lambda_\phi^2\phi_{I,B}\\ &+ 2\lambda_\phi(p - q\sin\phi_{I,B}\tan\theta_{I,B} - r\cos\phi_{I,B}\tan\theta_{I,B}) - (\ddot{\phi}_c + 2\lambda_\phi\dot{\phi}_c + \lambda_\phi^2\phi_c)\end{aligned} \tag{26}$$

Differentiating Eqs. (24) and (25) with respect to time yields

$$\dot{s}_q = \dot{q} + \lambda_q q - (\dot{q}_c + \lambda_q q_c) \tag{27}$$

$$\dot{s}_r = \dot{r} + \lambda_r r - (\dot{r}_c + \lambda_r r_c) \tag{28}$$

Substituting Eqs. (13)–(17) into Eqs. (26)–(28), the time derivatives of the sliding surfaces can be obtained as follows

$$\begin{aligned}\dot{s}_\phi &= (p - q\sin\phi_{I,B}\tan\theta_{I,B} - r\cos\phi_{I,B}\tan\theta_{I,B})(r\sin\phi_{I,B}\tan\theta_{I,B} - q\cos\phi_{I,B}\tan\theta_{I,B})\\ &- (q\cos\phi_{I,B} - r\sin\phi_{I,B})(q\sin\phi_{I,B}\sec^2\theta_{I,B} + r\cos\phi_{I,B}\sec^2\theta_{I,B})\\ &+ 2\lambda_\phi(p - q\sin\phi_{I,B}\tan\theta_{I,B} - r\cos\phi_{I,B}\tan\theta_{I,B}) + \lambda_\phi^2\phi_{I,B}\\ &- \frac{(I_z - I_x pr)}{I_y}\sin\phi_{I,B}\tan\theta_{I,B} - \frac{(I_x - I_y pq)}{I_z}\cos\phi_{I,B}\tan\theta_{I,B} - (\ddot{\phi}_c + 2\lambda_\phi\dot{\phi}_c + \lambda_\phi^2\phi_c)\\ &+ \frac{l_a}{2I_x}T_{A_L} - \frac{l_{cg}}{I_y}\sin\phi_{I,B}\tan\theta_{I,B}T_{A_M} - \frac{l_{cg}}{I_z}\cos\phi_{I,B}\tan\theta_{I,B}T_{A_N}\\ &= F_\phi - X_{c,\phi} + \frac{l_a}{2I_x}T_{A_L} - \frac{l_{cg}}{I_y}\sin\phi_{I,B}\tan\theta_{I,B}T_{A_M} - \frac{l_{cg}}{I_z}\cos\phi_{I,B}\tan\theta_{I,B}T_{A_N}\end{aligned} \tag{29}$$

$$\dot{s}_q = \frac{(I_z - I_x)pr}{I_y} + \lambda_q q - (\dot{q}_c + \lambda_q q_c) + \frac{l_{cg}}{I_y}T_{A_M} = F_q - X_{c,q} + \frac{l_{cg}}{I_y}T_{A_M} \tag{30}$$

$$\dot{s}_r = \frac{(I_x - I_y)pq}{I_z} + \lambda_r r - (\dot{r}_c + \lambda_r r_c) + \frac{l_{cg}}{I_z}T_{A_N} = F_r - X_{c,r} + \frac{l_{cg}}{I_z}T_{A_N} \tag{31}$$

where F_ϕ, F_q, and F_r are terms relevant to the states, and $X_{c,\phi}$, $X_{c,q}$, and $X_{c,r}$ are terms relevant to the commands with respect to ϕ, q, and r, respectively. Finally, the time derivative of the sliding surface vector of the inner-loop can be expressed as

$$\dot{s}_I = F_I - X_{c,I} + G_I u_I \tag{32}$$

where

$$s_I = \begin{bmatrix} s_\phi \\ s_q \\ s_r \end{bmatrix}, \quad u_I = \begin{bmatrix} T_{A_L} \\ T_{A_M} \\ T_{A_N} \end{bmatrix}, \quad F_I = \begin{bmatrix} F_\phi \\ F_q \\ F_r \end{bmatrix}, \quad X_{c,I} = \begin{bmatrix} X_{c,\phi} \\ X_{c,q} \\ X_{c,r} \end{bmatrix},$$

$$G_I = \begin{bmatrix} \frac{l_a}{2I_x} & -\frac{l_{cg}}{I_y} \sin\phi_{I,B} \tan\theta_{I,B} & -\frac{l_{cg}}{I_z} \cos\phi_{I,B} \tan\theta_{I,B} \\ 0 & \frac{l_{cg}}{I_y} & 0 \\ 0 & 0 & \frac{l_{cg}}{I_z} \end{bmatrix}$$

Let us propose a control input of the ACS including the proportional and sign terms with respect to the sliding surfaces as follows

$$u_I = -G_I^{-1}(F_I - X_{c,I} + K_{p,I} s_I + K_{s,I} \mathrm{sgn}(s_I)) \tag{33}$$

where

$$K_{p,I} = \begin{bmatrix} k_{1,p,I} & 0 & 0 \\ 0 & k_{2,p,I} & 0 \\ 0 & 0 & k_{3,p,I} \end{bmatrix}, \quad K_{s,I} = \begin{bmatrix} k_{1,s,I} & 0 & 0 \\ 0 & k_{2,s,I} & 0 \\ 0 & 0 & k_{3,s,I} \end{bmatrix}$$

with $k_{1,p,I}$, $k_{2,p,I}$, $k_{3,p,I}$, $k_{1,s,I}$, $k_{2,s,I}$, and $k_{3,s,I}$ are the positive real values. Applying the control input (33) to Eq. (32), the time derivative of the sliding surface vector of the inner-loop can be expressed as

$$\dot{s}_I = -K_{p,I} s_I - K_{s,I} \mathrm{sgn}(s_I) \tag{34}$$

Let us consider a following Lyapunov candidate function.

$$V_I = \frac{1}{2} s_I^T s_I \tag{35}$$

Differentiating Eq. (35) with respect to time and substituting Eq. (34) into the resulting equation yields

$$\dot{V}_I = s_I^T \dot{s}_I = -s_I^T K_{p,I} s_I - s_I^T K_{s,I} \mathrm{sgn}(s_I) \tag{36}$$

Note that $\dot{V}_I$ is negative definite, which implies that s_I converges to zero in finite time and is maintained thereafter. If $s_I = 0$ is maintained, the following equalities can be derived from Eqs. (23)–(25).

$$\dot{e}_\phi = -2\lambda_\phi e_\phi - \lambda_\phi^2 \int_0^t e_\phi d\tau \tag{37}$$

$$\dot{e}_q = -\lambda_q e_q \tag{38}$$

$$\dot{e}_r = -\lambda_r e_r \tag{39}$$

Equations (37)–(39) show that the errors of the roll angle and the pitch and yaw angular rates converge to zero.

3.1.2 DCS Control Input

In this study, the PPNG law is used to generate a guidance command with respect to the missile's velocity coordinate system. Because DCS can generate pitch and yaw accelerations with respect to the missile's body coordinate system, the relation between the missile's body and velocity coordinate systems should be used to generate the accelerations corresponding to the guidance command. The missile's body accelerations can be expressed in the missile's velocity coordinate system as follows

$$\begin{aligned} a_B^{[V_M]} &= \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 0 \\ a_y \\ a_z \end{bmatrix}^{[B]} \\ &= \begin{bmatrix} a_y \sin\beta + a_z \sin\alpha \cos\beta \\ a_y \cos\beta - a_z \sin\alpha \sin\beta \\ a_z \cos\alpha \end{bmatrix} \end{aligned} \tag{40}$$

where $a_B^{[V_M]}$ is the missile's body acceleration vector with respect to the missile's velocity coordinate system, α is the angle of attack, and β is the sideslip angle. To generate the accelerations corresponding to the guidance commands of the pitch and yaw accelerations with respect to the missile's velocity coordinate system, the following equalities should be satisfied.

$$a_{z,c}^{[V_M]} = a_z \cos\alpha \tag{41}$$

$$a_{y,c}^{[V_M]} = a_y \cos\beta - a_z \sin\alpha \sin\beta \tag{42}$$

where $a_{z,c}^{[V_M]}$ and $a_{y,c}^{[V_M]}$ are the guidance commands of the pitch and yaw accelerations with respect to the missile's velocity coordinate system. Using Eqs. (41) and (42), the pitch and yaw accelerations of the missile's body with respect to the missile's body coordinate system, which should be achieved, can be obtained as

$$a_{z,c}^{[B]} = \frac{a_{z,c}^{[V_M]}}{\cos\alpha} \tag{43}$$

$$a_{y,c}^{[B]} = \frac{a_{z,c}^{[V_M]}\tan\alpha\sin\beta + a_{y,c}^{[V_M]}}{\cos\beta} \tag{44}$$

Substituting Eqs. (9)–(11) into Eqs. (1) and (2), the pitch and yaw accelerations of the missile's body with respect to the missile's body coordinate system can be obtained as

$$a_z = \frac{T_{D_Z} + T_{A_M}}{m} \tag{45}$$

$$a_y = \frac{T_{D_Y} - T_{A_N}}{m} \tag{46}$$

To match the pitch and yaw accelerations of the missile's body to Eqs. (43) and (44), the DCS inputs can be derived based on Eqs. (45) and (46) as follows

$$T_{D_Z} = \frac{m a_{z,c}^{[V_M]}}{\cos\alpha} - T_{A_M} \tag{47}$$

$$T_{D_Y} = \frac{m}{\cos\beta}(a_{z,c}^{[V_M]}\tan\alpha\sin\beta + a_{y,c}^{[V_M]}) + T_{A_N}. \tag{48}$$

3.2 Outer-Loop Controller

The design objective of the outer-loop controller is to decrease the look angle of the seeker for the lock-on condition using the information of the seeker. The errors of the pitch and yaw angles from the reference coordinate system to the missile's body and LOS coordinate systems are defined as follows

$$e_O = \begin{bmatrix} e_\theta \\ e_\psi \end{bmatrix} = \begin{bmatrix} \theta_{I,B} \\ \psi_{I,B} \end{bmatrix} - \begin{bmatrix} \theta_{I,L} \\ \psi_{I,L} \end{bmatrix} \tag{49}$$

Geometrically, it can be stated that the look angle of the seeker σ converges to zero as e_θ and e_ψ converge to zero. Let us consider the sliding surfaces of the outer-loop including the integral terms as

$$s_\theta = e_\theta + \lambda_\theta \int_0^t e_\theta d\tau \tag{50}$$

$$s_\psi = e_\psi + \lambda_\psi \int_0^t e_\psi d\tau \tag{51}$$

where λ_θ and λ_ψ are the design parameters which are positive real values. Differentiating Eqs. (50) and (51) with respect to time and substituting Eqs. (17) and (18) into the resulting equations yields

$$\begin{aligned}\dot{s}_\theta &= \dot{\theta}_{I,B} - \dot{\theta}_{I,L} + \lambda_\theta(\theta_{I,B} - \theta_{I,L})\\ &= \lambda_\theta\theta_{I,B} - (\dot{\theta}_{I,L} + \lambda_\theta\theta_{I,L}) + q\cos\phi_{I,B} - r\sin\phi_{I,B} \qquad (52)\\ \dot{s}_\psi &= \dot{\psi}_{I,B} - \dot{\psi}_{I,L} + \lambda_\psi(\psi_{I,B} - \psi_{I,L})\\ &= \lambda_\psi\psi_{I,B} - (\dot{\psi}_{I,L} + \lambda_\psi\psi_{I,L}) + q\sin\phi_{I,B}\sec\theta_{I,B} + r\cos\phi_{I,B}\sec\theta_{I,B} \qquad (53)\end{aligned}$$

The time derivative of the sliding surface vector of the outer-loop can be described as

$$\dot{s}_O = F_O - L_O + G_O u_O \tag{54}$$

where

$$s_O = \begin{bmatrix} s_\theta \\ s_\psi \end{bmatrix}, \quad u_O = \begin{bmatrix} q \\ r \end{bmatrix}, \quad F_O = \begin{bmatrix} \lambda_\theta\theta_{I,B} \\ \lambda_\psi\psi_{I,B} \end{bmatrix}, \quad L_O = \begin{bmatrix} \dot{\theta}_{I,L} + \lambda_\theta\theta_{I,L} \\ \dot{\psi}_{I,L} + \lambda_\psi\psi_{I,L} \end{bmatrix},$$

$$G_O = \begin{bmatrix} \cos\phi_{I,B} & -\sin\phi_{I,B} \\ \sin\phi_{I,B}\sec\theta_{I,B} & \cos\phi_{I,B}\sec\theta_{I,B} \end{bmatrix}$$

Note that L_O in Eq. (54) can be obtained using the information from the seeker.

Let us propose a control input of the outer-loop including the proportional and sign terms with respect to the sliding surfaces as follows

$$u_O = -G_O^{-1}(F_O - L_O + K_{p,O}s_O + K_{s,O}\text{sgn}(s_O)) \tag{55}$$

where

$$K_{p,O} = \begin{bmatrix} k_{1,p,O} & 0 \\ 0 & k_{2,p,O} \end{bmatrix}, \quad K_{s,O} = \begin{bmatrix} k_{1,s,O} & 0 \\ 0 & k_{2,s,O} \end{bmatrix}$$

with $k_{1,p,O}$, $k_{2,p,O}$, $k_{1,s,O}$, and $k_{2,s,O}$ are the positive real values. Applying the control input (55) to Eq. (54), the time derivative of the sliding surface vector of the outer-loop can be obtained as

$$\dot{s}_O = -K_{p,O}s_O - K_{s,O}\text{sgn}(s_O) \tag{56}$$

Let us consider the following Lyapunov candidate function.

$$V_O = \frac{1}{2}s_O^T s_O \tag{57}$$

Differentiating Eq. (57) with respect to time and substituting Eq. (56) into the resulting equation, the time derivative of the Lyapunov candidate function can be obtained as

$$\dot{V}_O = s_O^T\dot{s}_O = -s_O^T K_{p,O}s_O - s_O^T K_{s,O}\text{sgn}(s_O) \tag{58}$$

Note that $\dot{V}_O$ is negative definite, which implies that s_O converges to zero in finite time and is maintained thereafter. If $s_O = 0$ is maintained, then the following equalities can be obtained from Eqs. (50) and (51).

$$\dot{e}_\theta = -\lambda_\theta e_\theta \tag{59}$$

$$\dot{e}_\psi = -\lambda_\psi e_\psi \tag{60}$$

Equations (59) and (60) imply that the errors of the pitch and yaw angles from the reference coordinate system to the missile's body and LOS coordinate systems converge to zero. Therefore, the look angle of the seeker σ converges to zero.

4 Numerical Simulation

To demonstrate the performance of the proposed controller, numerical simulation for the DACS type missile to intercept the ballistic target in the exoatmospheric environment is performed during the terminal phase. The missile's data used in the simulation is summarized in Table 1. The scenario of the simulation is a head-on engagement of the high speed missile and ballistic target with a little heading angle error at the exoatmospheric altitude as shown in Fig. 5. In Fig. 5, subscript 0 denotes the initial value, and the missile's body coordinate system coincides with the missile's velocity coordinate system in the initial time to set the angle of attack and sideslip angle to zero. The time history of the seeker's look angle is compared to a magnitude of an angle between the x-axes of the missile's velocity and line-of sight coordinate systems, which is regarded as the seeker's look angle in the previous studies, for the same scenario. In the simulation, the gravity effect is ignored, the arctan function is used to reduce chattering phenomena instead of the sign function [7], and the 3-dimensional PPNG law is used to generate the guidance commands of the pitch and yaw accelerations with respect to the missile's velocity coordinate system as [11]

$$a_{z,c}^{[V_M]} = -N_g \bar{V}_M \dot{\lambda}_y \cos \psi_{L,V_M} \tag{61}$$

$$a_{y,c}^{[V_M]} = -N_g \bar{V}_M \dot{\lambda}_y \sin \theta_{L,V_M} \sin \psi_{L,V_M} + N_g \bar{V}_M \dot{\lambda}_z \cos \theta_{L,V_M} \tag{62}$$

where N_g is the navigation constant, $\bar{V}_M$ is the total velocity of the missile, and λ_y and λ_z are the y and z components of the LOS vector with respect to the reference

Table 1 Data of the missile

Parameter	m [kg]	l_a [m]	l_{cg} [m]	I_x [kg · m^2]	I_y [kg · m^2]	I_z [kg · m^2]
Value	64	0.2	0.6	2.88	11.89	11.89

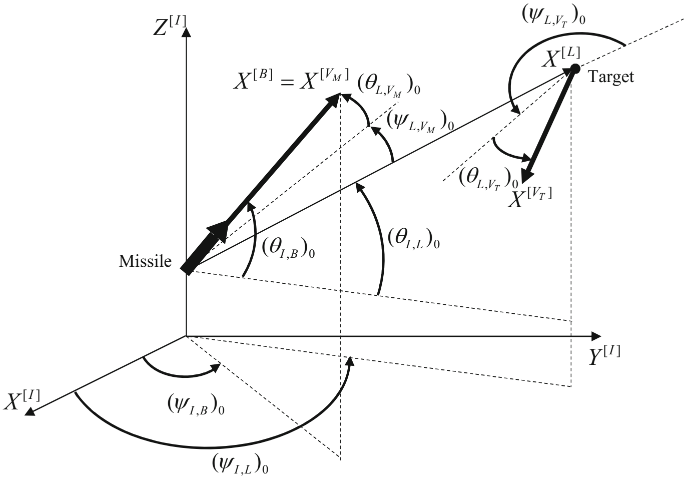


Fig. 5 Initial pursuit geometry of the head-on engagement

coordinate system. Considering the thruster's response time, the thrusters of the ACS and DCS are modeled as the following first-order system.

$$\frac{T}{T_c} = \frac{1}{\tau s + 1} \tag{63}$$

where T and T_c denote the actual and command thrust input, respectively, and τ is the time constant. In this study, τ is chosen as 0.01. The initial values of the simulation are summarized in Table 2, where $[X_M \; Y_M \; Z_M]_0^T$ and $[X_T \; Y_T \; Z_T]_0^T$ are the position vectors of the missile and target with respect to the reference coordinate system, respectively, $[V_{M_x} \; V_{M_y} \; V_{M_z}]^T$ and $[u \; v \; w]^T$ are the missile's velocity vectors with respect to the reference and missile's body coordinate systems, respectively, and $\bar{V}_T$ is the total velocity of the target. Note that the lock-on condition is initially satisfied, from the seeker's field-of-view limit and σ_0 in Table 2. The design parameters are selected as follows

$$\lambda_\phi = 2, \; \lambda_q = 1, \; \lambda_r = 1, \; \lambda_\theta = 1, \; \lambda_\psi = 1, \; N_g = 3$$

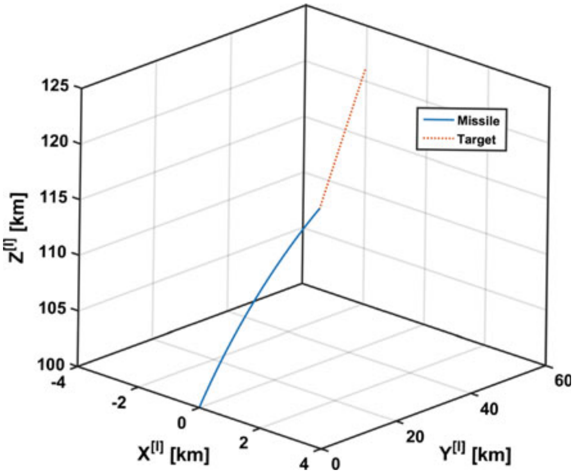
$$K_{p,I} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \; K_{p,O} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \; K_{p,O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; K_{s,O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Figures 6, 7, 8, 9, 10, 11 and 12 show the simulation results. Figure 6 presents the trajectories of the missile and target; the solid line represents the trajectory of the missile and the dashed line represents the trajectory of the target. It can be stated that the missile successfully intercepted the ballistic target in the head-on engagement

Table 2 Initial condition of the simulation

$[X_M\ Y_M\ Z_M]_0^T$ [km]	$[0\ 0\ 100]^T$
$[X_T\ Y_T\ Z_T]_0^T$ [km]	$[0\ 25\sqrt{3}\ 125]^T$
$[\lambda_x\ \lambda_y\ \lambda_z]_0^T$ [km]	$[X_T - X_M\ Y_T - Y_M\ Z_T - Z_M]_0^T = [0\ 25\sqrt{3}\ 25]^T$
$(\psi_{I,L})_0$, $(\theta_{I,L})_0$[deg]	$\left(\arctan\left(\frac{\lambda_y}{\lambda_x}\right)\right)_0 = 90$, $\left(\arccos\left(\sqrt{\frac{\lambda_x^2+\lambda_y^2}{\lambda_x^2+\lambda_y^2+\lambda_z^2}}\right)\right)_0 = 30$
$(\psi_{L,V_M})_0$, $(\theta_{L,V_M})_0$ [deg]	0.5, 0.5
$(\psi_{L,V_T})_0$, $(\theta_{L,V_T})_0$ [deg]	−178.5, −2.5
$(\psi_{I,B})_0$, $(\theta_{I,B})_0$, $(\phi_{I,B})_0$[deg]	$\left(\arctan\left(\frac{V_{M_y}}{V_{M_x}}\right)\right)_0 = 90.6$, $\left(\arccos\left(\sqrt{\frac{V_{M_x}^2+V_{M_y}^2}{V_{M_x}^2+V_{M_y}^2+V_{M_z}^2}}\right)\right)_0 = 30.5, 2$
$[u\ v\ w]_0^T$ [km/s]	$[5\ 0\ 0]^T$
$(\bar{V}_T)_0$ [km/s]	3
Seeker's field-of-view limit [deg]	±1
σ_0 [deg]	$\arccos\left(\frac{V_{M_x}\lambda_x+V_{M_y}\lambda_y+V_{M_z}\lambda_z}{\sqrt{(V_{M_x}^2+V_{M_y}^2+V_{M_z}^2)(\lambda_x^2+\lambda_y^2+\lambda_z^2)}}\right)_0 = 0.71$

Fig. 6 Trajectories of the missile and target



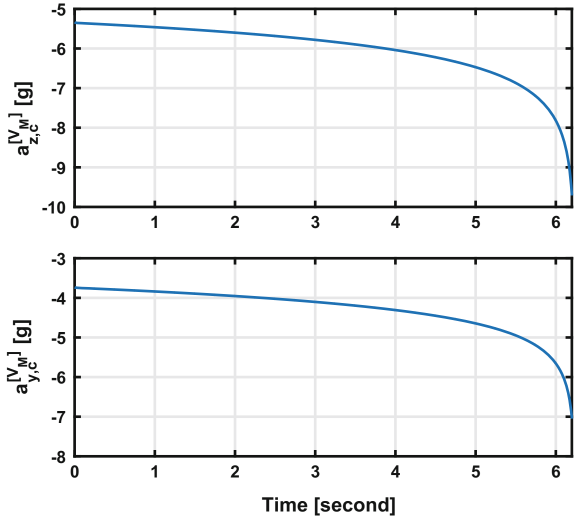
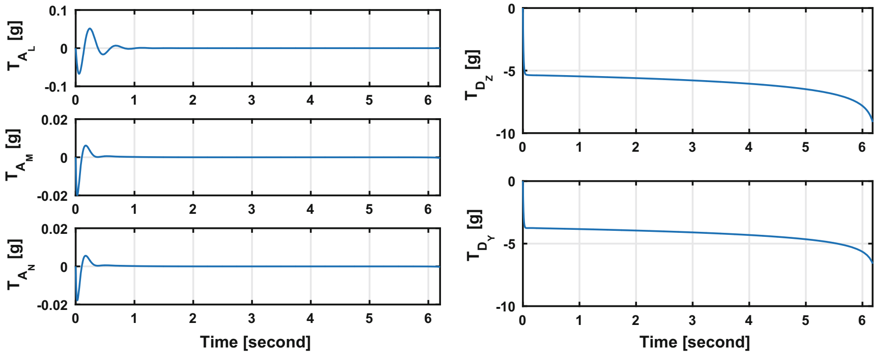


Fig. 7 Time histories of the guidance commands



(a) **ACS inputs**

(b) **DCS inputs**

Fig. 8 Time histories of the DACS inputs

at the exoatmospheric altitude. Figure 7 shows the time histories of the guidance commands of the pitch and yaw accelerations with respect to the missile's velocity coordinate system generated by the PPNG law; the pitch and yaw acceleration commands are within $\pm 10g$ until intercepting the target. Figure 8 presents the time histories of the DACS inputs as the specific forces. All ACS inputs are within $\pm 0.1g$, and especially T_{A_M} and T_{A_N} are within $\pm 0.02g$; the DCS inputs are within $\pm 10g$. Figure 9 shows the time histories of the look angles, where the solid line represents the time history of the seeker's look angle defined in this study and the dashed line represents the magnitude of the angle between the x-axes of the missile's velocity and LOS coordinate systems expressed as σ_V in the figure, and dotted line denotes

Fig. 9 Time histories of the look angles

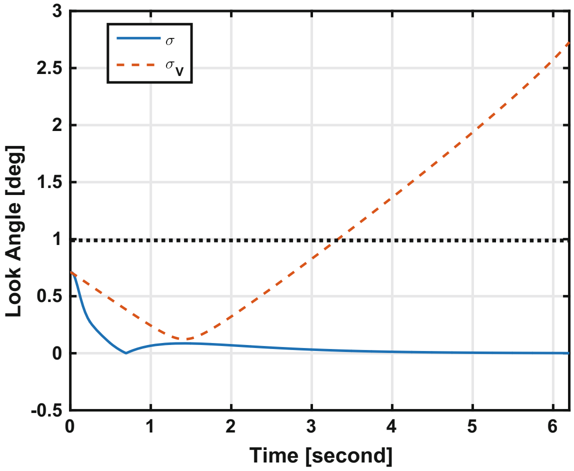
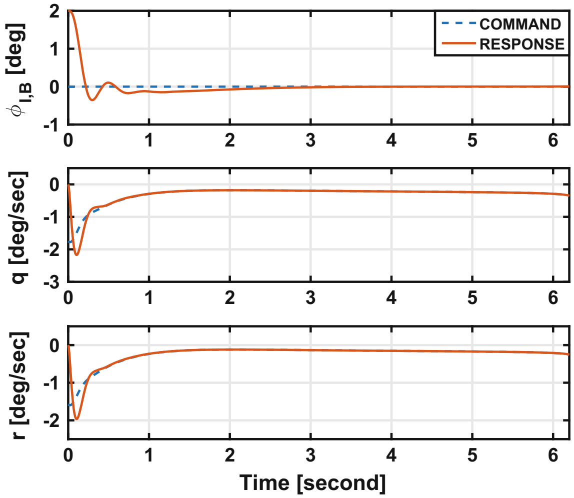


Fig. 10 Time histories of the $\phi_{I,B}$, q, and r



the seeker's field-of-view limit. From this result, it is found that the proposed controller can maintain the lock-on condition to decrease the look angle of the seeker to zero, even though σ_V exceeds the seeker's field-of-view limit after 3.4 seconds. Figure 10 shows the time histories of the $\phi_{I,B}$, q, and r; the solid line represents the responses of the $\phi_{I,B}$, q, and r, and the dashed line denotes the command signals. This result shows that the inner-loop controller generating the ACS inputs provides proper control performance corresponding to the commands of the outer-loop and zero roll angle. Figure 11 shows the time histories of the α and β; the magnitudes of the α and β grow up to 6° and 1.5°, respectively. Not regulating the α and β to zero, the lock-on condition can be maintained by decreasing the look angle while the σ_V is beyond the seeker's field-of-view limit. Figure 12 shows the time history of the

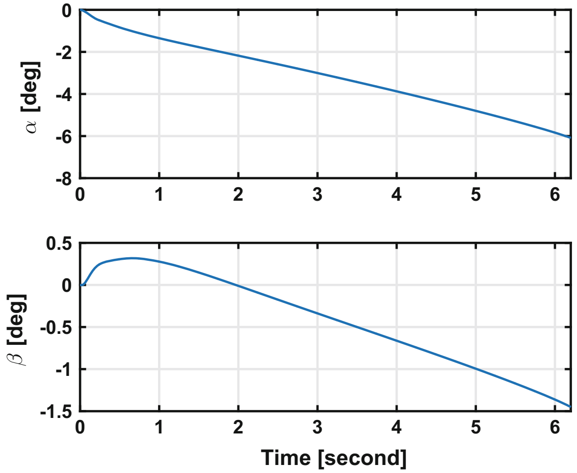


Fig. 11 Time histories of the α and β

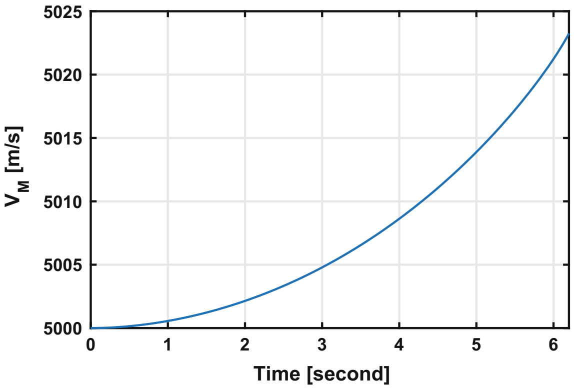


Fig. 12 Time history of the total velocity

missile's total velocity. If the DCS inputs generate the accelerations coinciding with the guidance commands of the pitch and yaw accelerations with respect to the missile's velocity coordinate system, then the total velocity will be maintained the initial value. However, because the DCS inputs generate the pitch and yaw accelerations with respect to the missile's body coordinate system and there exists a gap between the missile's body and velocity coordinate systems, the unnecessary term accelerating or decelerating the missile may be generated, i.e., x component of Eq. (40). Therefore, in this case, the total velocity of the missile increases about 25 m/s until intercepting the target. In summary, it can be stated that the proposed controller shows a satisfactory performance for the exoatmospheric missile with the seeker's narrow field-of-view.

5 Conclusion

A sliding mode based controller for the exoatmospheric divert and attitude control system (DACS) type missile with strapdown seeker was proposed considering the narrow field-of-view. Concerning that the missile's body coordinate system is not always aligned with the missile's velocity coordinate system, the look angle of the seeker was defined as the magnitude of the angle between the x-axes of the missile's body and line-of-sight (LOS) coordinate systems; previous studies have roughly considered the look angle of the seeker to a gap between the missile's velocity and LOS coordinate systems. The six-degrees-of-freedom equations of motions for the exoatmospheric DACS type missile were derived, and two design objectives of generating the accelerations corresponding to the guidance command while maintaining the lock-on condition were considered to design the controller with two-loop structure. The outer-loop controller generates the pitch and yaw rate commands, which decrease the look angle, and sends then to the inner-loop. The inner-loop controller generates the DACS inputs for tracking the commands of the guidance law, angular rate and stabilization of the roll motion. Numerical simulation was performed for the head-on engagement in the exoatmospheric environment using the 3-dimensional pure proportional navigation guidance law. Numerical simulation results demonstrate that the proposed controller has a proper performance in terms of maintaining the lock-on condition and tracking the guidance command.

Acknowledgements This work was conducted at High-Speed Vehicle Research Center of KAIST with the support of Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

References

1. Andrieu V, Lahanier HP (2006) Exoatmospheric interception problem solved using output feedback law. Syst Control Lett 55(8):633–639
2. Erer KS, Tekin R, Ozgoren MK (2015) Look angle constrained impact angle control based on proprotional navigation. In: AIAA guidance, navigation, and control conference, Kissimmee, FL
3. Hablani HB, Pearson DW (2004) Miss distance error analysis of exoatmospheric interceptors. J Guidance Control Dyn 27(2):283–289
4. Joner S, Quinquis I (2006) Control of an exoatmospheric kill vehicle with a solid propulsion attitude control system. In: AIAA guidance, navigation, and control conference, Keystone, CO
5. Kim TH, Park BG, Tahk MJ (2013) Bias-shaping method for biased proportional navigation with terminal-angle constraint. J Guidance Control Dyn 36(6):1810–1815
6. Lee CH, Hyun C, Lee JG, Choi JY, Sung SK (2013) A hybrid guidance law for a strapdown seeker to maintain lock-on conditions against high speed targets. J Electr Eng Technol 8(1):190–196
7. Lee Y, Kim Y, Moon G, Jun BE (2016) Sliding-mode-based missile-integrated attitude control scheme considering velocity change. J Guidance Control Dyn 39(3):423–436
8. Reisner D, Shima T (2013) Optimal guidance-to-collision law for an accelerating exoatmospheric interceptor missile. J Guidance Control Dyn 36(6):1695–1708

9. Sang DK, Tahk MJ, Guidance law switching logic considering the seeker's field-of-view limits. Proc Inst Mech Eng Part G J Aerosp Eng 223(8):1049–1058
10. Shima T, Golan O (2012) Exo-atmospheric guidance of an accelerating interceptor missile. J Franklin Inst 349(2):622–637
11. Song SH, Ha IJ (1994) A Lyapunov-like approach to performance analysis of 3-dimensional pure PNG laws. IEEE Trans Aerosp Electron Syst 30(1):238–248
12. Yingbo H, Yong Q (2003) THAAD-Like high altitude theater missile defense: strategic defense capability and certain countermeasures analysis. Sci Glob Secur 11(2–3):151–202
13. Zhang Y, Wang X, Wu H (2014) Impact time control guidance law with field of view constraint. Aerosp Sci Technol 39(1):361–369

Mechanical/Control Integrated Design of a Flexible Planar Rotatory Spacecraft

J.A. Perez, D. Alazard, T. Loquen and C. Pittet

1 Introduction

Currently Large Space Structures (LSS) are a challenging problem in control system design because they involve large complex kinematic chains composed of rigid and flexible bodies, mostly large-sized, maximally lightened, low-damped and with closed-spaced low natural frequencies. In this case structural modes interfere the controlled bandwidth, provoking a critical Control-Structure Interaction (CSI). Therefore, LSS design is increasingly becoming subject to a close coordination among the different spacecraft sub-systems, demanding methods which tie together spacecraft structural dynamics, control laws and propulsion design. These methods are often called as *Integrated Control/Structure Design* (ICSD), *Plant-Controller Optimization* (PCO) or simply *co-design* (CD).

ICSD methods began being studied in the 80 s as an opposite technique to the current method of separated iterative sequences within the structural and control disciplines. The first integrated design methodologies were those in [1–3]. These methods were based on iterative methodologies with optimization algorithms. Lately, other methods have been proposed such as those solved by LMI (Linear Matrix Inequality) algorithms or with LQG (Linear Quadratic Gaussian) methods like in [4, 5] respectively. However, these approaches give conservative results and their applicability is restricted by problem dimension. Recently, a counterpart technique currently under development in ONERA Toulouse Research Center and ISAE-SUPAERO allows a

J.A. Perez (✉) · T. Loquen
ONERA, 2 Avenue Edouard Belin, 31055 Toulouse, France
e-mail: Jose-Alvaro.perez_gonzalez@onera.fr

T. Loquen
e-mail: thomas.loquen@onera.fr

D. Alazard
ISAE, 10 Avenue Edouard Belin, 31055 Toulouse, France
e-mail: daniel.alazard@isae.fr

C. Pittet
CNES, 18 Avenue Edouard Belin, 31400 Toulouse, France
e-mail: christelle.pittet@cnes.fr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_35

more general approach [6]. Actually, this method is based on structured $\mathscr{H}_\infty$ synthesis algorithms developed in [7] or [8], granting structured controllers and tunable parameters optimization. This synthesis, merged with a correct plant modeling, can reveal important applications of integrated design methodologies.

To be compliant with such an ICSD approach, the plant modeling of LSS must provide a model where the structural or mechanical sizing parameters can be isolated under the general Linear Fractional Representation (LFR). That can be performed using the Two-Input Two-Output Port (TITOP) approach to build the linear dynamical model of Flexible Multi-body System (FMS) from the model of each body. The TITOP model approach was first proposed in [9], then generalized to take into account varying sizing parameters in [10, 11]. For uniform beam, an analytic model was developed in [12] where a first ICSD was presented. Augmented TITOP models for flexible body actuated with piezoelectric strip [13] involve an additional channel between the input voltage and the output piezoelectric charge, allowing any active substructure to be embedded in the whole structure. Some applications of ICSD using such a modeling technique are presented in [14, 15].

This work aims at showing the way to apply these modeling and design tools to perform the ICSD of a flexible planar rotatory spacecraft. Such a system was presented in [16] to illustrate interactions between control and structure in space engineering. The objective is to maximize the length and the tip mass of one appendage, in order to increase the spacecraft payload capacity, while minimizing the total mass of the spacecraft and meeting attitude pointing requirement in spite of external disturbances. This paper is organized as follows. First, the parametric model of the spacecraft is developed. Second the integrated control design is presented. Finally, results and discussion on advantages of using this methodology rather than single control optimization alone are detailed.

2 Flexible Rotatory Spacecraft Modeling

2.1 Spacecraft System Description

The system, considered as an academic example, is composed of a rigid main hub with four identical cantilevered flexible appendages and tip masses as shown in Fig. 1. The configuration parameters are provided in Table 1. Under normal operation, the spacecraft undergoes planar rotational maneuvers around the inertial fixed axis **z**. The spacecraft body frame is attached to the center of mass of the rigid hub, and it is denoted by a right-handed triad **x**, **y** and **z**. The rotation about the axis **z** is denoted by the angle θ and the translational deformation of each tip by w^i_{tip}, with superscript i denoting the beam number.

The system is actuated by three different torques. The main torque, t_{hub} is provided by the main hub about the axis **z**. Two additional input torques, $t_{tip,1}$ and $t_{tip,2}$, are applied at the tip masses 1–3 and 2–4 respectively. These torques can be applied

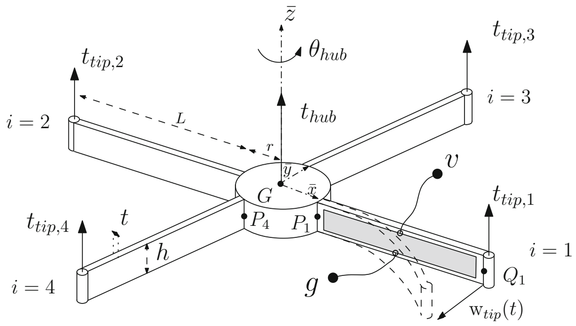


Fig. 1 Rotatory flexible spacecraft

Table 1 Spacecraft configuration parameters

Parameters	Symbol	Value
Hub radius	r	0.31 m
Hub mass	m_h	233.50 kg
Hub inertia	J_h	10.85 kg/m^2
Beam mass density	ρ	1.30 kg/m
Beam elastic modulus	E	75.84 GPa
Beam length	L	1.22 m
Beam thickness	t	3.18 mm
Beam height	h	0.15 m
Tip mass	m_t	2.29 kg
Tip mass inertia	J_t	2.44 g/m^2
Nodes beam FEM	nod	11
Number of AM	asm	13
Piezo parameters	Symbol	Value
Thickness	t_p	2 mm
Width	w_p	30 mm
Volumetric density	ρ_p	7600 kg/m^3
Elastic modulus	E_p	50 GPa
Piezoelectric constant	d_{31}	-150×10^{-12} m/V
Dielectric constant	ε_{33}^T	1.59×10^{-12} F/m

purposely for control reasons or can represent environment disturbances (as it is the case in this paper). In addition, the appendage # 1 is actuated using a piezoelectric strip for active damping of vibrations.

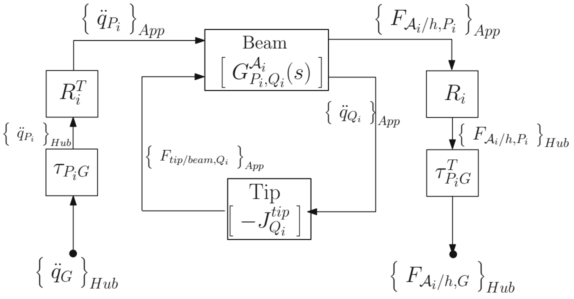

$\{\ \ddot{q}_{P_i}\ \}_{App}$
Beam
$\left[\ G^{\mathcal{A}_i}_{P_i,Q_i}(s)\ \right]$
$\{\ F_{\mathcal{A}_i/h,P_i}\ \}_{App}$
R_i^T
R_i
$\{\ \ddot{q}_{Q_i}\ \}_{App}$
$\{\ \ddot{q}_{P_i}\ \}_{Hub}$
$\{\ F_{\mathcal{A}_i/h,P_i}\ \}_{Hub}$
$\{\ F_{tip/beam,Q_i}\ \}_{App}$
τ_{P_iG}
$\tau^T_{P_iG}$
Tip
$\left[\ -J^{tip}_{Q_i}\ \right]$
$\{\ \ddot{q}_G\ \}_{Hub}$
$\{\ F_{\mathcal{A}_i/h,G}\ \}_{Hub}$

Fig. 2 Model $A_i(s)$ of a single appendage ($i = 2, 3, 4$)

2.2 Spacecraft System Modeling

The whole spacecraft is decomposed in various sub-structures: the hub h and the 4 appendages $\mathscr{A}_i$, $i = 1, \ldots, 4$. Each appendage is itself decomposed into a uniform beam which flexible dynamics is represented by a super-element [11] and a local mass/inertia at the tip of the beam. The TITOP (Two-Input Two-Output) model approach is used to described each sub-structure. The block diagram representation of a single appendage model is depicted in Fig. 2 where:

- $G^{\mathscr{A}_i}_{P_i,Q_i}(s)$ is the planar TITOP model of the flexible uniform beam # i (see [11] for more details): the inputs are the time-derivative of the planar twist $\ddot{q}_{P_i}$ (3 components: 2 linear accelerations and 1 angular acceleration) at point P_i of the beam and the external planar wrench $F_{tip/beam,Q_i}$ (3 components: 2 forces and 1 torque) applied to the beam at point Q_i. The output are the wrench $F_{\mathscr{A}_i/h,P_i}$ applied by the appendage to the hub at point P_i and the time-derivative of the twist $\ddot{q}_{Q_i}$ at point Q_i,
- $J^{tip}_{Q_i}(s)$ is the planar dynamical model of the tip mass and inertia:

$$J^{tip}_{Q_i} = \begin{bmatrix} m_t & 0 & 0 \\ 0 & m_t & 0 \\ 0 & 0 & J_t \end{bmatrix} \tag{1}$$

- τ_{P_iG} is the kinematic models between points G and P_i. In the planar case:

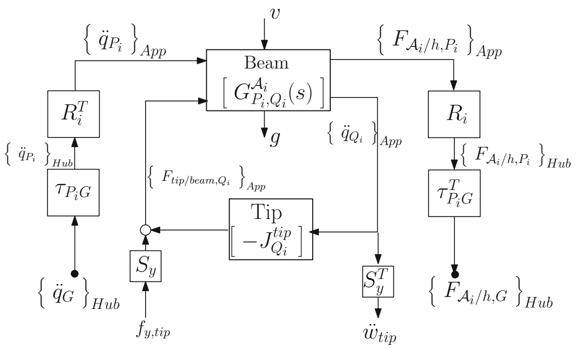
$$\tau_{P_1G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & r \\ 0 & 0 & 1 \end{bmatrix} \quad \tau_{P_2G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -r \\ 0 & 0 & 1 \end{bmatrix} \quad \tau_{P_3G} = \begin{bmatrix} 1 & 0 & -r \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \tau_{P_4G} = \begin{bmatrix} 1 & 0 & r \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

- R_i is the rotation matrix written as follows:

$$R_i = \begin{bmatrix} \cos\beta_i & -\sin\beta_i & 0 \\ \sin\beta_i & \cos\beta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}_{App \to Hub} \tag{3}$$

where β_i is the angle of the ith appendage i with $\mathbf{x}$.

Fig. 3 Model $A_1(s)$ of appendage # 1 $i = 1$)


v
$\{\ \ddot{q}_{P_i}\ \}_{App}$
Beam
$\left[\ G^{\mathcal{A}_i}_{P_i,Q_i}(s)\ \right]$
$\{\ F_{\mathcal{A}_i/h,P_i}\ \}_{App}$
R_i^T
R_i
g
$\{\ \ddot{q}_{Q_i}\ \}_{App}$
$\{\ \ddot{q}_{P_i}\ \}_{Hub}$
τ_{P_iG}
$\{\ F_{tip/beam,Q_i}\ \}_{App}$
Tip
$\left[\ -J^{tip}_{Q_i}\ \right]$
$\{\ F_{\mathcal{A}_i/h,P_i}\ \}_{Hub}$
$\tau^T_{P_iG}$
S_y
S_y^T
$\{\ \ddot{q}_G\ \}_{Hub}$
$f_{y,tip}$
$\ddot{w}_{tip}$
$\{\ F_{\mathcal{A}_i/h,G}\ \}_{Hub}$

For the appendage # 1, actuated with a piezoelectric strip, the extension of TITOP model described in [13] is used. The augmented model $A_1(s)$ involves an additional channel between v (input voltage) and g (output piezoelectric charge). Furthermore a radial disturbance force $f_{y,tip}$ on input and the radial acceleration $\ddot{w}_{tip}$ on output are included to shape the control problem (see Sect. 3.2). For appendage $\mathscr{A}_1$, the model is then described in Fig. 3 where $S_y = [0\quad 1\quad 0]^T$.

The assembly of the various sub-models is described by the block diagram scheme presented in Fig. 4, where:

- $J_G^{Hub} = \begin{bmatrix} m_h & 0 & 0 \\ 0 & m_h & 0 \\ 0 & 0 & J_h \end{bmatrix}$
- $S_z = [0\quad 0\quad 1]^T$

The inputs and the outputs of the whole system mechanical model $(J_G^{sat})^{-1}(s)$ are:

- a vector w of two exogenous inputs: a disturbance $w_{\ddot{\theta}_{out}}$ on the hub angular acceleration and $f_{y,tip}$,
- a vector u of two control signals: the torque on the hub t_{hub} and the piezo input voltage v,
- a vector z of two controlled outputs: the hub angular rate $\ddot{\theta}_{hub}$ and the lateral tip acceleration $\ddot{w}_{tip}$ of appendage 1,
- a vector y of 3 measurements: the hub angular rate $\dot{\theta}_{hub}$ and attitude θ_{hub} and the lateral tip acceleration $\ddot{w}_{tip}$ of appendage 1.

2.3 Spacecraft Model with Varying Parameters

The structured model $(J_G^{sat})^{-1}(s)$ of the spacecraft is quite convenient to take into account some sizing mechanical parameters or design parameters. In this application,

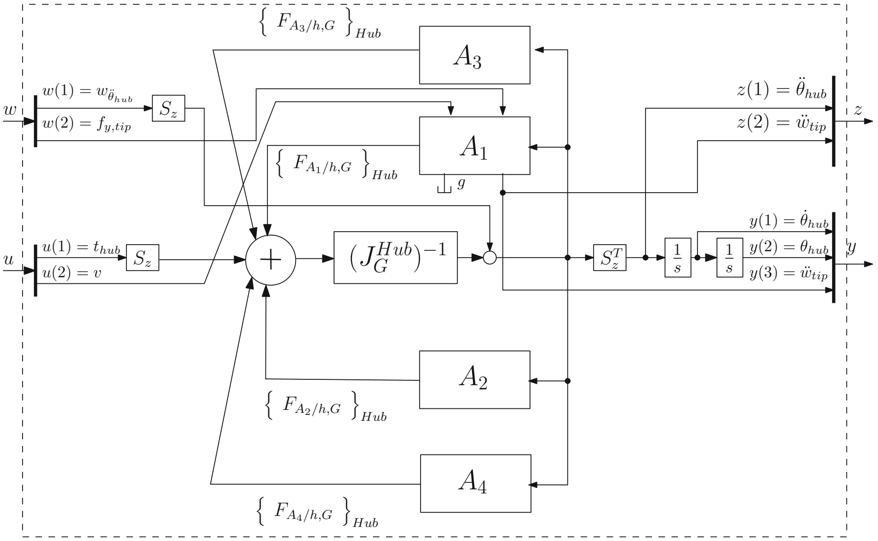


Fig. 4 Rotatory flexible spacecraft $(J_G^{sat})^{-1}(s)$

the design parameters are the length L_i and the tip mass M_i of each appendage. More precisely:

$$L_i = L(1 + 0.3\delta_{L_i}), \quad M_i = m_t(1 + 0.3\delta_{M_i}), \quad i = 1, 2, 3, 4.$$

That is to say: 30% of variations around the nominal value are allowed for each design parameters L_i and M_i in such a way that δ_{L_i} and δ_{M_i} are normalized to vary between -1 and 1. These design can be very easily isolated in the various submodels (L_i in $G_{P_i,Q_i}^{\mathscr{A}_i}(s)$ and M_i in $J_{Q_i}^{tip}$). Thus, using LFR for each parameters and basic LFR operations, one can easily derived the LFR of the whole spacecraft depicted in Fig. 5, also written:

$$(J_G^{sat})^{-1}(s, \delta_{L_1}, \delta_{L_2}, \delta_{L_3}, \delta_{L_4}, \delta_{M_1}, \delta_{M_2}, \delta_{M_3}, \delta_{M_4}) = F_u((M_G^{sat})^{-1}(s), \varDelta) \ .$$

After assembly, the system has a block $\varDelta$ of 125 × 125 size and reads:

$$[\varDelta] = \begin{bmatrix} [\delta_{L_1}]_{30\times 30} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & [\delta_{M_1}]_{2\times 2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & [\delta_{L_2}]_{29\times 29} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & [\delta_{M_2}]_{2\times 2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [\delta_{L_3}]_{29\times 29} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & [\delta_{M_3}]_{2\times 2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & [\delta_{L_4}]_{29\times 29} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & [\delta_{M_4}]_{2\times 2} \end{bmatrix} \tag{4}$$

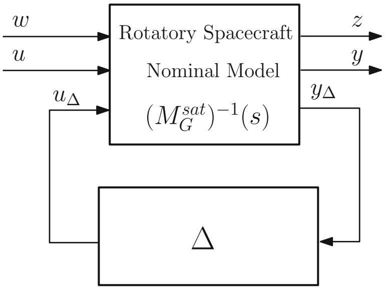


Fig. 5 Rotatory flexible spacecraft model with varying design parameters

That is: 2 parameter occurrences for each tip mass, 29 parameter occurrences for the non-actuated appendage lengths and 30 parameter occurrences for the actuated appendage length $i = 1$. The system results in a state-space system with a Δ block of size 125×125, 11 inputs, 11 outputs and 50 states.

It should be highlighted that the ICSD of the rotatory spacecraft strongly demands a modeling technique such as the TITOP modeling technique, since the boundary conditions of the flexible beams are to be changed when varying the mass located at their tips. Thus, by using the TITOP modeling technique, the impact of mass variation in the whole system will be taken into account when ICSD is performed with structured $\mathscr{H}_\infty$ synthesis. Indeed, in the symmetric nominal parametric configuration ($\Delta = 0$), some flexible modes are uncontrollable from u. Then, the damping of these flexible modes, when excited by a disturbance (on $f_{y,tip}$, $t_{tip,1}$ or $t_{tip,3}$), is not possible. The asymmetry introduced by the ICSD will restore the controllability and will allow active damping of all the modes, as it will be seen in Sect. 4.

3 Integrated Control/Structure Design (ICSD)

3.1 General Approach

Figure 6 shows the standard multi-channel $\mathscr{H}_\infty$ synthesis problem for ICSD of a generic system $G(s)$. The synthesis scheme is build by establishing the feedback of the *augmented* controller $K(s) = \text{diag}(C(s), \Delta)$ with the corresponding inputs/outputs of the nominal system $G(s)$. Additional channels are added in order to weight the controller's frequency response and mechanical performance index. The synthesis scheme has three different channels:

- One multidimensional channel which connects the perturbations of the system, w, to the performance outputs, z, through the weighting W_z.
- One multidimensional channel which connects the inputs of the control system, y_c, to the outputs, u_c, to shape the frequency-domain response of the controller $C(s)$ (roff-off behavior for instance) through the weighting W_C.

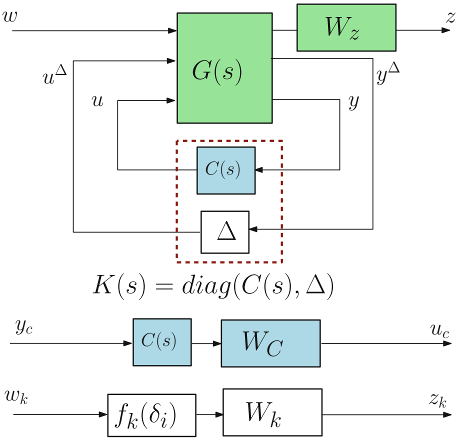


Fig. 6 Block diagram of integrated design optimization

- One multidimensional channel which connects the inputs w_k of the mechanical performance function $f_k(\delta_i)$ block to its outputs z_k, through a weighting W_k. This function $f_k(\delta_i)$ depends directly on the design parameters variations δ_i (i.e. the various independent components of the block Δ).

Structured $\mathscr{H}_\infty$ synthesis computes the sub-optimal tuning of the free parameters of $C(s)$ and Δ embedded in $K(s)$ to enforce closed-loop internal stability, $\mathscr{F}_l(G(s), K(s))$, such that:

$$\min_{K(s)}\{\max\{\|W_k(s)f_k(\delta_i)\|_\infty, \|W_C(s)C(s)\|_\infty\}\}$$

such that

$$\|W_z(s)T_{w\rightarrow z}(s)\|_\infty < \gamma_{perf}$$

i.e., it minimizes the $\mathscr{H}_\infty$ norm between the transfer of the perturbation input w and the performance output z, $T_{w\rightarrow z}(s)$, such that it is constrained to be below $\gamma_{perf} > 0$ to meet the required performances. The problem is in the form of multi-channel $\mathscr{H}_\infty$ synthesis, and it allows the set of desired properties to the augmented controller such as its internal stability [6], frequency template [17] or maximum gain values, while minimizing a mechanical performance index. In substance, the structured $\mathscr{H}_\infty$ integrated design synthesis tunes the free parameters contained in the augmented controller $K(s) = \text{diag}(C(s), \Delta_i)$ to ensure closed loop internal stability and meet normalized $\mathscr{H}_\infty$ requirements through W_z, W_C and W_k.

3.2 Application to the Rotatory Spacecraft

The augmented controller is formed by concatenating the block of tunable parameters Δ with the real controller of the system, $C(s)$. Since the Δ block has been defined in Sect. 2.3, the structure of $C(s)$ is addressed in this section.

The system needs to reject low frequency disturbances in the rigid body DOF, the system's rotation around the hub, and can be helped by inducing active damping through the piezoelectric laminate installed on appendage 1. Thus, the control system will consist of two decentralized loops: one for the rigid body rotation of the hub, θ_{hub}, and the other to damp the first appendage's tip vibrations, $\ddot{w}_{tip_1}$.

The control of the rigid body motion is achieved with a PD (Proportional-Derivative) controller to compute the control torque provided by the reaction wheel located at the hub. The active damping controller (Active: AF) will be a simple rate feedback, integrating the first appendage's vertical acceleration (acceleration feedback strategy). Therefore the control law is structured as follows:

$$\{u\} = [C(s)]\{y\} = \begin{Bmatrix} t_{hub} \\ v \end{Bmatrix} = \begin{bmatrix} k_v & k_p & 0 \\ 0 & 0 & k_a/s \end{bmatrix} \begin{Bmatrix} \dot{\theta}_{hub} \\ \theta_{hub} \\ \ddot{y}_{tip_1} \end{Bmatrix} \tag{5}$$

The proportional control gain k_p, the derivative control gain k_v and the damping gain k_a, together with the tunable parameters of the Δ block, are to be optimized with structured $\mathscr{H}_\infty$ synthesis. The values of the PD controller gains are initialized using the standard tuning assuming the spacecraft is rigid: $k_p = J_t\omega^2$ and $k_v = 1.4J_t\omega$ where J_t is the total inertia and ω the wanted attitude servo-loop bandwidth ($\omega = 1$ rad/s leads to: $K_p = 633.33$ Nm and $k_v = 231.21$ Nms).

As previously specified, the control of the hub rotation must be able to reject low frequency disturbance torque. The desired closed-loop dynamics for perturbation rejection are $\omega = 1$ rad/s and $\xi = 0.7$, which leads to the following weighting filter on the Acceleration Sensitivity Function ASF [6] (i.e. transfer from $w(1) = w_{\ddot{\theta}_{hub}}$ to output $z(1) = \ddot{\theta}_{hub}$ in Fig. 4):

$$W_{\ddot{\theta}_{hub}}(s) = \frac{s^2 + 1.4\omega s + \omega^2}{s^2} = \frac{s^2 + 1.4s + 1}{s^2} \tag{6}$$

The damping of the flexible modes is imposed by a template on the transfer between the hub's acceleration disturbance $w(1) = w_{\ddot{\theta}_{hub}}$ and the performance output $z(2) = \ddot{w}_{tip}$. This template reads: $W_{\ddot{q}P}(s) = L/W_{\ddot{\theta}_{hub}}$ (L is the nominal length of the beam). A static filter $W_d = 1/M_{tot}$ (with M_{tot} the nominal total mass) is added in the transfer $f_{y,tip} \rightarrow \ddot{w}_{tip}$ to add additional damping. In Figs. 7 and 8 frequency-domain responses of the closed-loop transfers $T_{w(1)\rightarrow z(1)}$ (called hub's dynamics) and $T_{w(1)\rightarrow z(2)}$ (called tip's dynamics) when only the 2 gains k_p and k_v of the PD controller are optimized. These responses are compared with the desired frequency response imposed through the templates and $W^{-1}_{\ddot{\theta}_{hub}}(s)$ and $W_{\ddot{q}P}(s)$. It is seen that the

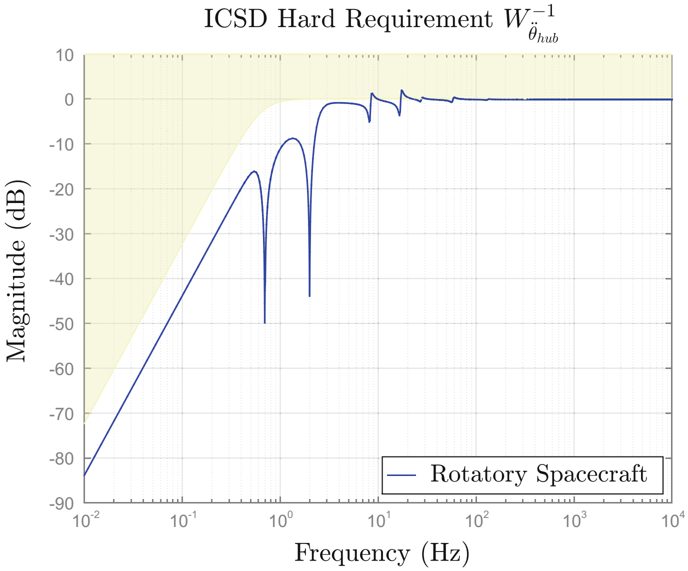


Fig. 7 Constraint (*yellow*) on the hub's dynamics (*blue*)

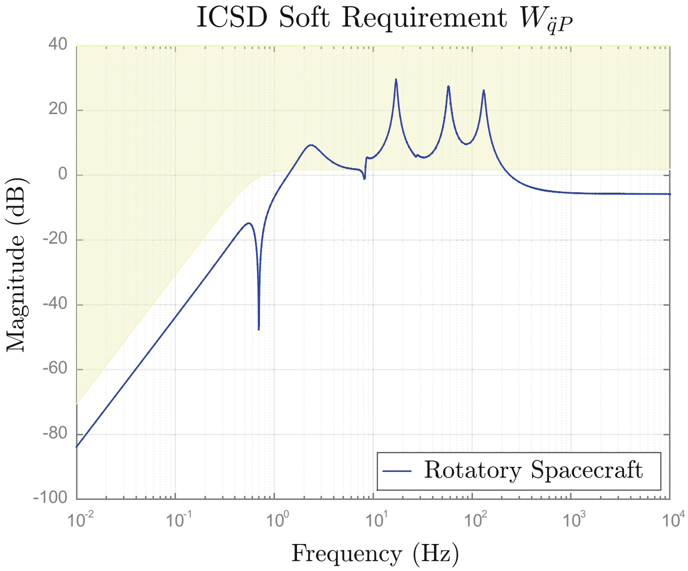


Fig. 8 Constraint (*yellow*) on the tip's dynamics (*blue*)

flexible modes are badly damped and the PD controller gains have a larger value than the needed to respect the template on the ASF of the hub position θ_{hub}.

Once the constraints for rigid and flexible motion have been defined, additional channels can be added to constraint the variation of structural parameters. The constraints for the maximization of the length and tip mass of appendage 1, L_1 and M_1 respectively, are implemented as follows:

$$f_{M_1}(\delta_{M_1}) = \frac{1}{1 + 0.3\delta_{M_1}}; \quad W_{M_1} = 0.75 \tag{7}$$

$$f_{L_1}(\delta_{L_1}) = \frac{1}{1 + 0.3\delta_{L_1}}; \quad W_{L_1} = 0.75 \tag{8}$$

where the W_k values have been fixed to 0.75, a value which can never be reached with the allowed maximal variation, in order to encourage the maximum possible value for δ_{L_1} and δ_{M_1}. The constraint for minimum total mass is:

$$f_{M_{total}}(\delta_{L_i}, \delta_{M_i}) = L\,\rho\,t\,h\,(4 + 0.3\sum_{i=1}^{4}\delta_{L_i}) + m_t(4 + 0.3\sum_{i=1}^{4}\delta_{M_i}); \quad W_{M_{total}} = M_0 \tag{9}$$

where $M_0 = 15.51\,\text{kg}$ is the nominal total mass of all the appendages. Equation (9) is a combination of all the structural parameters that can be varied, weighted by their impact on the system total mass (total beams mass for the lengths, total tip mass for the tip masses).

Finally, the ICSD problem can be summarized in the following way ($T_{i \to o}$ denotes the closed-loop transfer from input i to output o defined in the model of Fig. 4):

Find a stabilizing set of parameters $\Theta = \{k_v, k_p, k_a, \delta_{M_i}, \delta_{L_i}\}$ such that:

$$\begin{aligned}\min_{\Theta}\{\max\{\; & \|W_{M_1} f_{M_1}(\delta_{M_1})\|_\infty, \|W_{L_1} f_{L_1}(\delta_{L_1})\|_\infty, \\ & \|W_{M_{total}} f_{M_{total}}(\delta_{L_i}, \delta_{M_i})\|_\infty, \|W_{\ddot{q}P}^{-1}(s) T_{w(1)\to z(2)}(s)\|_\infty, \\ & \|W_d T_{w(2)\to z(2)}(s)\|_\infty\}\}\end{aligned}$$

under the constraint:

$$\|W_{\ddot{\theta}_{hub}}(s) T_{w(1)\to z(1)}(s)\|_\infty < 1.5.$$

4 Results

The optimization of the controller and the structural parameters is performed using the structured $\mathscr{H}_\infty$ synthesis tool *systune*. The results of the ICSD solution are compared with those of a solution with control optimization alone (COA). Table 2 shows the optimized and nominal structural parameters. It can be seen that the length and tip mass of appendage 1 have been increased the maximum allowed, 30%. For appendages 2, 3 and 4 the tip masses have been minimized while the lengths have been increased almost to the maximum, with the exception of appendage 2. Appendage 3 and its opposite appendage 4 are no longer symmetric since they lengths are slightly different and the tip masses difference is around 0.15 kg. The structural optimization meets the specifications: maximization of mass and length of appendage 1 while minimizing the impact on the total system's mass.

Figures 9 and 10 show the resulting frequency response for $T_{w(1)\to z(1)}$ and $T_{w(1)\to z(2)}$ compared with the desired templates $W^{-1}_{\ddot{\theta}_{hub}}$ and $W_{\ddot{q}P}$ after optimization. The gains of the PD controller have been adjusted to fit the frequency template and flexible modes are shifted and more damped when comparing with the response given in Figs. 7 and 8. The shift of flexible modes is a consequence of the structural optimization, since tip masses and lengths have been modified. The damping is provided by the active damping provided by the piezoelectric material controlled with an acceleration feedback control law.

The ICSD solution enjoys the same robust performance as the COA solution for the hub position control. The NICHOLS diagram of the open loop $t_{hub} \to t_{com}$ response in Fig. 11 shows that the ICSD solution has satisfactory phase and gain margins (GM = 20.5 dB, PM = 69.1°) which are as good as the COA solution (GM = 13.5 dB, PM = 69.1°). However, this is achieved by the ICSD solution with longer appendages which are not symmetric. Indeed, the first flexible mode, located at $\omega =$ 4.4 rad/s appears to be uncontrollable in nominal configuration, corresponding to the symmetric bending of the system's appendages. However, the ICSD solution has tuned the system to be completely asymmetric so that the first flexible mode can be governed with the hub torque as well. This is confirmed in Fig. 12, where the BODE diagram of the open loop response shows that the first flexible mode appears as a

Table 2 Structural data before optimization (COA) and after performing ICSD to the rotatory flexible spacecraft

Parameters	COA	ICSD
Controller	PD + AF	PD + AF
Lengths L_1 & L_2	1.22 m	$L_1 = 1.59$ m & $L_2 = 1.36$ m
Masses M_1 & M_2	2.29 kg	$M_1 = 2.98$ kg & $M_2 = 1.60$ kg
Lengths L_3 & L_4	1.22 m	$L_3 = 1.59$ m & $L_4 = 1.54$ m
Masses M_3 & M_4	2.29 kg	$M_3 = 1.75$ kg & $M_4 = 1.60$ kg
Total Appendages Mass	15.51 kg	15.83 kg

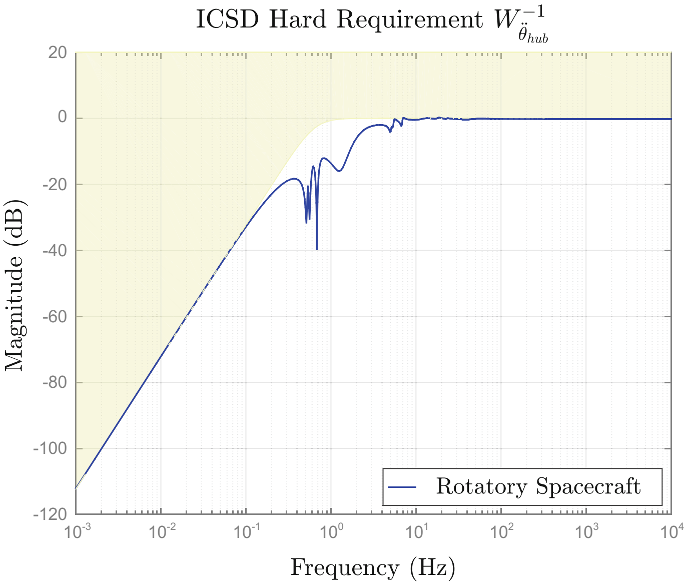


Fig. 9 Constraint (*yellow*) on the hub's dynamics (*blue*)

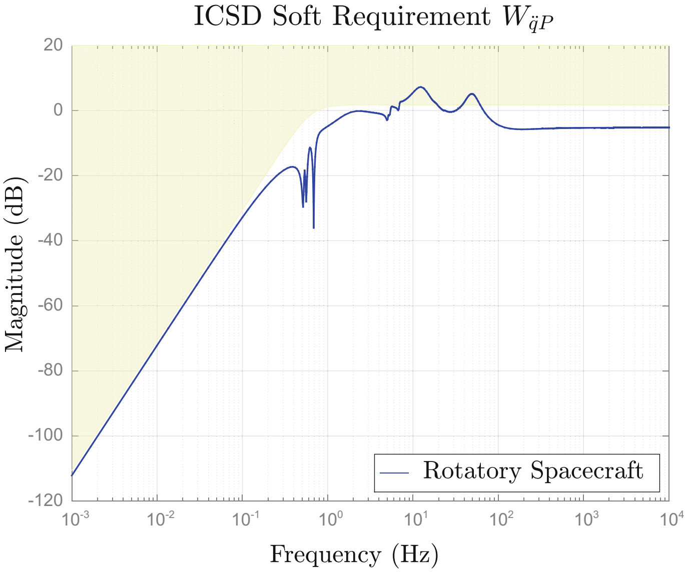


Fig. 10 Constraint (*yellow*) on the tip's dynamics (*blue*)

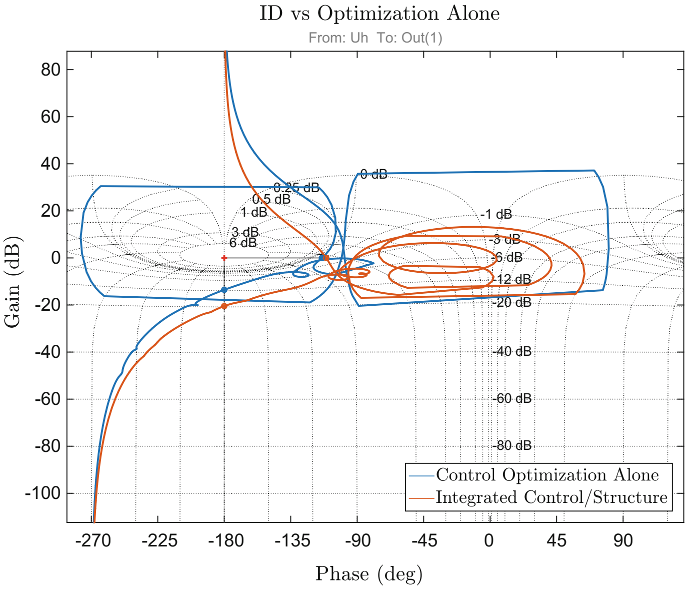


Fig. 11 Nichols plot

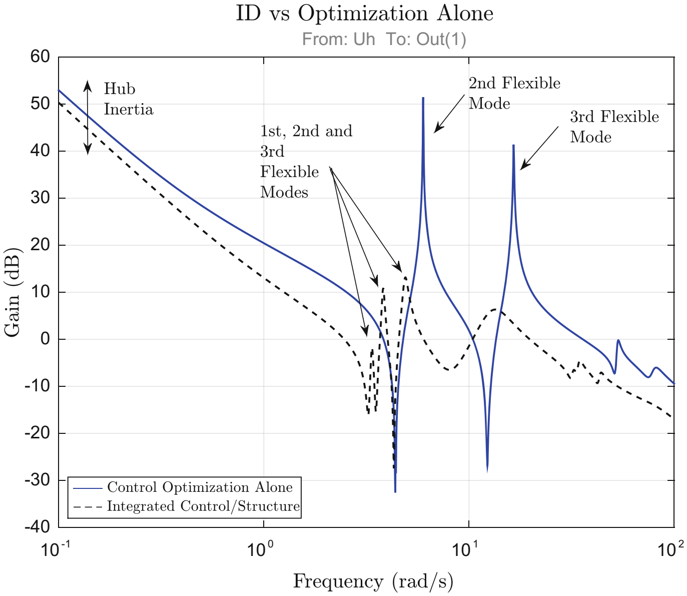


Fig. 12 Bode plot

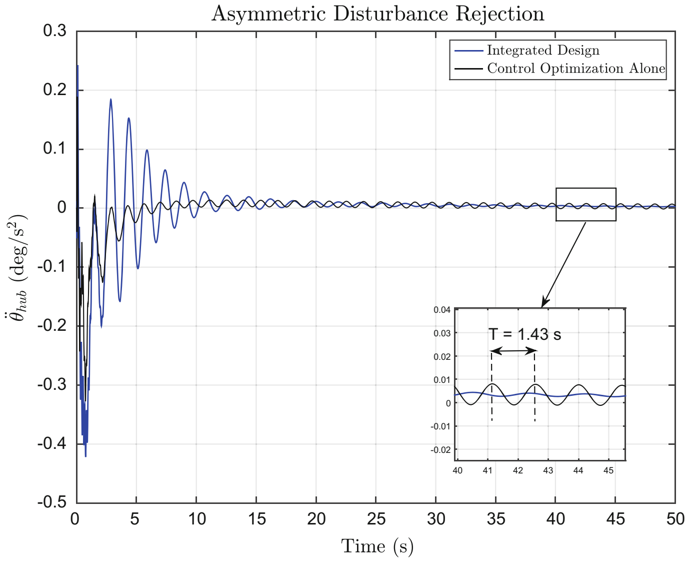


Fig. 13 Hub acceleration $\ddot{\theta}_{hub}$

resonance frequency in the ICSD solution and as an anti-resonance frequency for the COA solution.

The uncontrollability of the first symmetric bending mode at $\omega = 4.4\,\text{rad/s}$ is also verified in the time domain response. Figures 13 and 14 show the closed-loop response of the hub's acceleration $\ddot{\theta}_{hub}$ and the first appendage tip's acceleration $\ddot{y}_{tip_1}$, respectively, to an asymmetric torque disturbance at tips 1 and 3 and for both ICSD and COA solutions. This input excites the symmetric bending mode of appendages 3 and 4, which cannot be damped by the COA solution. It can be seen as well that the acceleration of the COA solution has a higher overshoot than the ICSD solution, even if the ICSD solution has higher tip mass and length. On the other hand, the ICSD solution needs two times more time for damping the tip vibrations and hub's position oscillations.

The results show that ICSD using structured $\mathscr{H}_\infty$ synthesis can be achieved by implementing the desired specifications in $\mathscr{H}_\infty$ form. The structured $\mathscr{H}_\infty$ synthesis optimizes the controller and the structural parameters to better fit the dynamic specifications, while respecting the structural constraints imposed to the varying parameters. The same level of performance can be achieved with a modified structure, discovering new configurations which can improve control performance. The optimization process has provided an intuitive fact: the maximization of the mass of one appendage will decrease the total mass of the others. The optimization has provided a counter-intuitive fact as well: system's asymmetry can help to increase controllability of system's modes and to improve system's performance. Therefore,

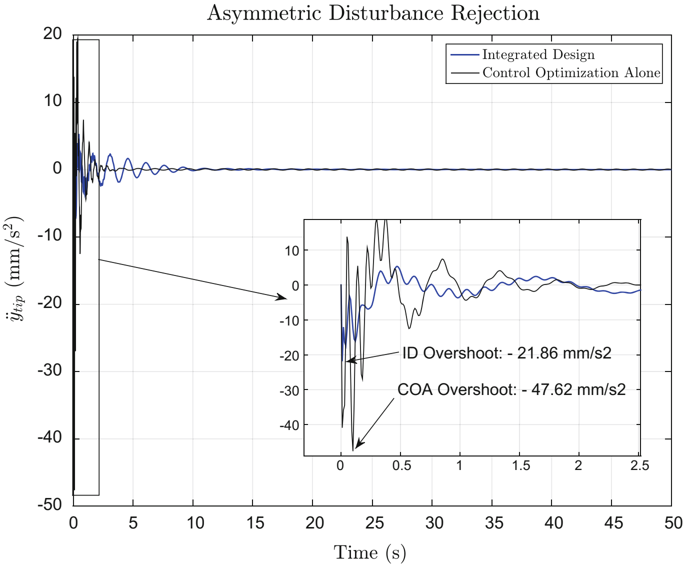


Fig. 14 Tip acceleration $\ddot{w}_{tip}$

integrated design is possible and it takes into account the issues and trade-offs of the physical system.

5 Conclusions and Perspectives

The implementation of dynamic, structural and controller specifications for the integrated control/structure design (ICSD) has been explained in this paper. The specifications for the rigid body motion are implemented weighting the Acceleration Sensitivity Function (ASF), while the flexible motion damping is achieved by projecting the rigid body motion template at different points of the Flexible Multi-body System (FMS). The structural constraints for structured optimization are implemented with additional channels which included the cost functions and weighting filters applied to the varying parameters involved in the constraints. Controller frequency shaping can be stated through a roll-off filter. The application of ICSD on a flexible planar rotatory spacecraft provided quite promising results. Further works will use the same approach to perform the ICSD of a more complex system, a flexible satellite, in the 3D case. The TITOP approach, used to obtain linear parameter varying models of flexible multi-body systems required for such a ICSD, will be extended to flexible multi-body systems with closed-loop kinematic mechanisms.

References

1. Onoda J, Haftka R (1987) An approach to structure/control simultaneous optimization for large flexible spacecraft. AIAA 25:1133–1138
2. Gilbert MG (1988) "Results of an Integrated Structure/Control Law design sensitivity analysis," Technical Report NASA TM-101517, NASA, NASA Langley Research Center, Hampton, VA 23665–5225, Dec 1988
3. Messac A, Malek K (1992) Control structure integrated design. AIAA J 30:2124–2131
4. Hiramoto K, Mohammadpour J, Grigoriadis K (2009) Integrated design of system parameters, control and sensor actuator placement for symmetric mechanical systems. In 48th IEEE conference on decision and control, Shanghai, China, Dec 2009
5. Cimellaro G, Soong T, Reinhorn A (2008) Optimal integrated design of controlled structures. In: 14th world conference on earthquake engineering, Beijing, China, Oct 2008
6. Alazard D, Loquen T, de Plinval H, Cumer C (2013) Avionics/Control co-design for large flexible space structures. In: AIAA guidance, navigation, and control (GNC) conference. Massachusetts, USA, Boston, Aug 2013
7. Gahinet P, Apkarian P (August 2011) Structured H-infinity synthesis using MATLAB. In: 18th IFAC world congress. Milano, Italy
8. Burke J, Henrion D, Lewis A, Overton M (2006) HIFOO a Matlab package for fixed-order controller design and $\mathscr{H}_\infty$ optimization. In: 5th IFAC symposium on robust control design, 2006
9. Alazard D, Perez JA, Loquen T, Cumer C (2015) Two-input two-output port model for mechanical systems. In: AIAA science and technology forum and exposition, Kissimmee, Florida, Jan 2015
10. Perez JA, Alazard D, Loquen T, Cumer C, Pittet C (2015) Linear dynamic modeling of spacecraft with open-chain assembly of flexible bodies for ACS/structure co-design. In: Advances in aerospace guidance, navigation and control, pp 659–678, Springer
11. Perez JA, Alazard D, Loquen T, Pittet C, Cumer C (2016) Flexible multibody system linear modeling for control using component modes synthesis and double-port approach. ASME J Dyn Syst Meas Control, 138
12. Murali H, Alazard D, Massotti L, Ankersen F, Toglia C (2015) Mechanical-attitude controller co-design of large flexible space structures. Advances in aerospace guidance, navigation and control. Springer, Berlin, pp 639–658
13. Perez JA, Alazard D, Loquen T, Pittet C (2016) Linear modeling of a flexible substructure actuated through piezoelectric components for use in integrated control/structure design. In: 20th IFAC symposium on automatic control in aerospace, 2016
14. Perez JA, Pittet C, Alazard D, Loquen T Integrated control/ structure design of a large space structure using structured $\mathscr{H}_\infty$ control. In: 20th IFAC symposium on automatic control in aerospace
15. Perez J, Pittet C, Alazard D, Loquen T, Cumer C (2015) A flexible appendage model for use in integrated control/structure spacecraft design. In: IFAC workshop on advanced control and navigation for autonomous aerospace vehicles. Seville, Spain
16. Junkins JL, Kim Y (1993) Introduction to dynamics and control of flexible structures. AIAA
17. Loquen T, de Plinval H, Cumer C, Alazard D (2012) Attitude control of satellite with flexible appendages: structured H-infinity approach. In: AIAA guidance, navigation, and control (GNC) conference, Mineapolis (Minesota), Aug 2012

SPARTAN: A Novel Pseudospectral Algorithm for Entry, Descent, and Landing Analysis

Marco Sagliano, Stephan Theil, Vincenzo D'Onofrio and Michiel Bergsma

1 Introduction

In the last decades the theoretical development of more refined direct methods, together with a new generation of CPUs, led to a significant improvement of numerical approaches for solving optimal-control problems. One of the most promising class of methods is based on Pseudospectral (PS) Optimal Control, originally formulated by Ross et al. [13]. These methods transform the original infinite-dimensional problem, that is, the continuous Bolza problem, into a finite-dimensional, discrete Nonlinear Programming (NLP) problem, which can be solved with one of the well-known off-the-shelf solvers, like Snopt [7] and Ipopt [17]. The discrete solution can be later converted into a continuous form by using Lagrange polynomials. Several tools implementing PS methods, have been developed, among others DIDO [4]. However, even if performing excellent, this tool requires ad-hoc manual scaling for the problems analyzed, which turns into a large time-consuming process when done by hand, and to the risk of numerical issues if ignored. Alternatively, Rao et al. [12] proposed a self-scaling method, based on the so-called Jacobian Rows Normalization (JRN) scheme. However, one can see that this scaling method, although properly working, is not optimal, as requires multiple computations of the Jacobian matrix associated with the problem, and at the same time does not offer the largest reduction of the condition number, here taken as measure of the numerical conditioning of the problem to be solved. In SPARTAN (Simple Pseudospectral Algorithm for Rapid Trajectory ANalysis) a second self-scaling method, based on the so-called Projected-Jacobian Rows Normalization (PJRN) [14] is implemented.

Another important aspect, related with the solution of the finite NLP problem, is the computation of the Jacobian. In fact, one can see that the exploitation of the Jacobian structure leads to a sum of three contributions, which can be computed

M. Sagliano (✉) · S. Theil · V. D'Onofrio · M. Bergsma
German Aerospace Center, Robert Hooke Str. 7, Bremen, Germany
e-mail: marco.sagliano@dlr.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_36

exactly, and provides computational advantages. The overall result is a state-of-the-art pseudospectral method, which is a valid choice for performing preliminary analyses of entry, descent, and landing scenarios, and that can be easily used to rapidly prototype a solution for complex, nonlinear problems, as it will be shown in this paper. This work is structured as follows: in Sect. 2 the general optimal-control problem we deal with is briefly introduced, while pseudspectral methods, and specifically the flipped-radau PS method, are described in Sect. 3. The proposed improvements on PS methods are fully described in Sects. 4 and 5. Specifically, in Sect. 4 the projected jacobian rows normalization (PJRN) is introduced, while the systematic hybrid Jacobian computation, together with the dual number theory, is explained in Sect. 5. Numerical results obtained for the Space Shuttle entry guidance problem, and the JAXA's Trojan mission-based asteroid descent and landing are shown in Sect. 6. Finally, in Sect. 7 some conclusions are drawn.

2 Optimal Control Problem

There are several approaches for the generation of reference trajectories. Some methods exploit the structure of the specific problems we deal with. Often, they require simplifications to make the problem mathematically tractable, and therefore generate solutions valid under given hypotheses. A different approach, which is gaining popularity, and is helped by the development of the computational capabilities of modern CPUs is the representation of the trajectory generation problem as an optimal-control problem. This means we are looking for solutions minimizing (or maximizing) a given criterion, and satisfying at the same time several constraints, which can be differential (e.g., the equations of motion of a spacecraft) and/or algebraic (e.g., the maximum heat-flux that a vehicle can tolerate). The standard form for representing optimal-control problems is the so-called Bolza problem. Given a state vector $\mathbf{x}(t) \in \mathbb{R}^{n_s}$, a control vector $\mathbf{u}(t) \in \mathbb{R}^{n_c}$, the scalar functions $\Phi(t, \mathbf{x}, \mathbf{u})$ and $\Psi(\mathbf{x}, \mathbf{u})$, and the vector $\mathbf{g}(t, \mathbf{x}, \mathbf{u}) \in \mathbb{R}^{n_p}$ we can formulate the problem as follows.

Minimize (maximize) the cost function J

$$J = \Phi\left[t_f, \mathbf{x}\left(t_f\right), \mathbf{u}\left(t_f\right)\right] + \int_{t_0}^{t_f} \Psi\left[\mathbf{x}(t), \mathbf{u}(t)\right] dt \tag{1}$$

subject to the differential equations

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}, \mathbf{u}\right) \tag{2}$$

and to the path constraints

$$\mathbf{g}_L \leq \mathbf{g}\left(\mathbf{x}, \mathbf{u}\right) \leq \mathbf{g}_U \tag{3}$$

The first term in the cost function (1) takes the name of *Mayer* term, and represents punctual constraints (e.g., the minimization of a distance according to a given metric), while the argument of the integral is called the *Lagrange* term and is used to maximize or minimize variables over the entire mission (e.g., the heat load obtained by integrating the heat-flux over time). Moreover, since we deal with physical systems, the problem has usually bounded states and controls, that is, $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are compact in $\mathbb{R}^{n_s}$ and $\mathbb{R}^{n_c}$, respectively.

$$\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U \tag{4}$$

$$\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \tag{5}$$

Equations (1)–(5) represent a generic continuous optimal control problem. In the next section we will see how this type of OCP can be transcribed by using Pseudospectral methods.

3 Pseudospectral Methods

Numerical methods for solving OCPs are divided in two major classes, namely, indirect methods and direct methods. Indirect methods are based on the Pontryagin Maximum Principle, which leads to a multiple-point boundary-value problem. Direct methods, instead, consist in the proper discretization of the OCP, (or *transcription*), having as a result a finite-dimensional, nonlinear programming (NLP) problem. PS Methods represent a particular area of interest in the frame of the wider class of direct methods. In detail, SPARTAN, an optimal-control package developed by the German Aerospace Center [3, 9, 14, 16] uses the global flipped Radau Pseudospectral Method (fRPM), based on the flipped Legendre-Radau polynomials [6, 15]. This choice allows for a straightforward definition of the initial conditions of the problem. Moreover, the following properties are valid:

- "Spectral" (i.e., quasi-exponential) convergence in the case of a smooth problem
- Runge phenomenon is avoided
- Straightforward implementation
- Sparse structure of the associated NLP problem
- Mapping between the discrete costates of the associated NLP and the continuous costates of the Optimal Control Problem (except for LPM) in virtue of the Pseudospectral Covector Mapping Theorem [8].

In addition, the fRPM shows a smooth convergence of the costates. This is not always the case when other PS methods are employed [6]. Therefore, it is useful to have a look at the fRPM and how it can be conveniently employed to solve OCPs, focusing on the transcription process which defines the corresponding NLP. This does not only involve the choice of the discrete nodes, but also determines the discrete differential and integral operators needed to solve the differential and integral parts of the associated OCP. Therefore, the *transcription* is a more general process than the *discretization*. The minimum fundamental steps of a *transcription* are the following:

- domain discretization
- discrete to continuous conversion of states and/or controls
- characterization of differential and integral operators

The first step is the domain discretization. In the frame of the fRPM, the time domain discretization in n nodes uses the roots of the flipped Legendre-Radau polynomial, defined as the combination of the Legendre polynomial of order n and $n-1$ with coefficient equal to 1 and -1 respectively.

$$R_n(\tau) = L_n(\tau) - L_{n-1}(\tau),\ \tau \in [-1, 1] \tag{6}$$

An example of roots associated with the Legendre-Radau polynomial of order 10 is depicted in Fig. 1a, together with the corresponding polynomial.[1] This discrete representation of the domain is useful to reconstruct continuous representations of the functions $x(t)$ as:

$$x(t) \cong \sum_{i=0}^{n} X_i P_i(t), \quad P_i(t) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{t-t_k}{t_i-t_k} \tag{7}$$

An example of this approximation is depicted in Fig. 1b, where the function $1/(1 + 25\tau^2)$ is reconstructed by using 25 fRPM nodes.[2]

Once that the domain has been discretized, and the discrete-to-continuous conversion of states has been defined, the corresponding differential operator needs to be derived. This is required for the proper representation of the left-hand side of Eq. (2). The differential operator will be in the form

$$\dot{\mathbf{X}}_i \cong \mathbf{D} \cdot \mathbf{X}_i, \ , i = 1, \ldots, n \tag{8}$$

and the dynamics defined in Eq. (2) will be replaced by

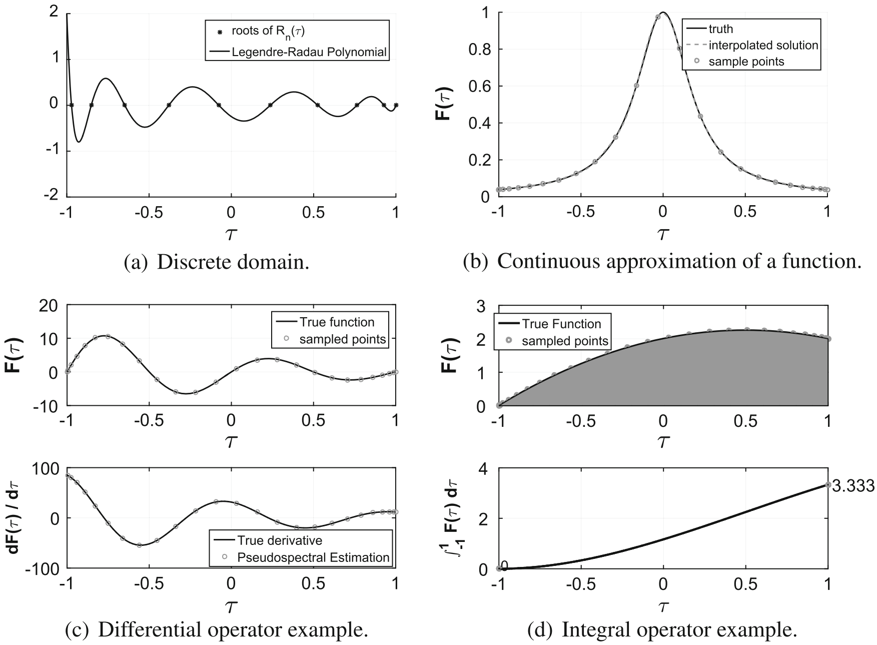
$$\mathbf{D} \cdot \mathbf{X} = \frac{t_f - t_0}{2} \mathbf{f}(t, \mathbf{X}, \mathbf{U}) \tag{9}$$

where t_0 and t_f are the initial and final time, and the term $\frac{t_f - t_0}{2}$ is a scale factor related to the transformation between the physical time domain t, and the pseudospectral time domain $\tau \in (-1, 1]$, given by the following affine transformations.

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}, \quad \tau = \frac{2}{t_f - t_0} t - \frac{t_f + t_0}{t_f - t_0} \tag{10}$$

[1] Note that the $R_n(-1)$ is not a root of the underlying polynomial, therefore it is not a collocation point, although it is required for the evaluation of the polynomial. This choice is motivated by the fact that over the left-open, right-closed interval $(-1, +1]$ these polynomials are orthogonal.

[2] Note that the approximation becomes more accurate when the number of nodes is increased. This is the opposite behavior observed when uniform distributions of nodes, which suffer from the aforementioned *Runge Phenomenon*, are employed.



(a) Discrete domain.

(b) Continuous approximation of a function.

(c) Differential operator example.

(d) Integral operator example.

Fig. 1 Transcription steps: domain discretization (**a**), continuous reconstruction of functions (**b**), definition of differential (**c**) and integral (**d**) operators

In the case of the fRPM the matrix $\mathbf{D}$ has dimensions $[n \times (n+1)]$. Again, this is due to the fact that the states are defined for $n+1$ discrete points, while the controls $\mathbf{U}$ and the derivatives of the states $\mathbf{f}(t, \mathbf{X}, \mathbf{U})$ are defined in the n collocation points. This means that the initial state $\mathbf{X}_0$ is an input and not an output of the optimization, and it is thus assumed to be known. If we look at Eq. (7), and taking the derivative w.r.t. time, we get

$$\dot{\mathbf{x}}(t) \cong \frac{d}{dt} \sum_{i=0}^{n} \mathbf{X_i} P_i(t) = \sum_{i=0}^{n} \mathbf{X_i} \frac{d}{dt} P_i(t) \tag{11}$$

as the nodal points are time-independent. The derivatives in Eq. (11) can be efficiently computed with the Barycentric Lagrange Interpolation [10]. An example of the differential operator is depicted in Fig. 1c, where $\mathbf{D}$ is used to approximate the derivative of a continuous function $F(\tau) = Ae^{-\tau} \sin(\omega\tau)$ sampled in 25 collocation nodes. It can be seen that the polynomial approximation fits the analytical derivative very well. In addition to the differential operator, we need an integral operator. This operator is required as the cost function in Eq. (1) may contain the Lagrange term, which needs a proper discretization. In that case the Gauss quadrature formula is used [1]. This approach consists of replacing the continuous integral with the discrete sum given by:

$$\int_{t_0}^{t_f} \Psi\left[t, \mathbf{x}(t), \mathbf{u}(t)\right] dt = \frac{t_f - t_0}{2} \sum_{i=1}^{n} w_i \Psi\left[\mathbf{X}_i, \mathbf{U}_i\right] \tag{12}$$

It can be shown that Eq. (12) yields exact results for polynomials of order at most equal to $2n - 2$. Once again, the presence of the term $\frac{t_f - t_0}{2}$ is a consequence of the mapping between pseudospectral and physical time domains described in the relationships (10). For the fRPM the weights w_j can be computed as

$$w = \mathit{flip}(\tilde{w}) \tag{13}$$

$$\tilde{w}_j = \begin{cases} \dfrac{2}{n^2}, & j = 1 \\ \dfrac{(1 - \tau_j)}{n^2 L_n(\tau_j)^2}, & j = [2, \ldots, n] \end{cases} \tag{14}$$

where the operator *flip* simply multiplies the input by a factor equal to -1, and sorts the results in increasing order. To give a practical example the integral of the test function $F(\tau) = 2\tau + 2 - \tau^2$ has been computed. Results are then compared with the analytical integral, and with the trapezoidal rule (Fig. 1d). Numerically, we get exactly the analytical result, that is 3.3333, while the application of the trapezoidal rule by using the same number of nodes gives 3.3298. Once that a good approximation of the differential and integral operators have been described, we are ready to formulate the NLP problem which approximates the original OCP as follows:

Minimize (maximize) the cost function J, for n nodes, $i = 1, \ldots, n$:

$$J = \Phi\left[t_f, \mathbf{X}_f, \mathbf{U}_f\right] + \frac{t_f - t_0}{2} \sum_{i=1}^{n} w_i \Psi\left[\mathbf{X}_i, \mathbf{U}_i\right] \tag{15}$$

subject to the nonlinear algebraic constraints

$$\mathbf{F} = \mathbf{D} \cdot \mathbf{X} - \frac{t_f - t_0}{2} \mathbf{f}(t, \mathbf{X}, \mathbf{U}) = \mathbf{0} \tag{16}$$

and to the path constraints

$$\mathbf{g}_L \le \mathbf{G}\left(\mathbf{X}_i, \mathbf{U}_i\right) \le \mathbf{g}_U \tag{17}$$

The discrete states and the controls are bounded, as in the continuous formulation.

$$\mathbf{x}_L \le \mathbf{X}_i \le \mathbf{x}_U \tag{18}$$

$$\mathbf{u}_L \le \mathbf{U}_i \le \mathbf{u}_U \tag{19}$$

This is the formal definition of the Nonlinear Programming Problem to solve. However, even if theoretically the problem could be solved, in practice further issues arise. In particular, the numerical conditioning of the problem, the exploitation of the Jacobian Matrix, and its computation play a major role in the quality of the results and the speed of the computation.

4 Hybridization of Jacobian Matrix

Let us now consider the general structure of the Jacobian matrix associated with the NLP problem deriving from the application of fRPM, defined as follows.

$$\mathbf{Jac} = \left\{ \begin{array}{c} \nabla J \\ \nabla \mathbf{F} \\ \nabla \mathbf{G} \end{array} \right\} \tag{20}$$

The operator ∇ represents the vector of derivative w.r.t. the discrete state vector $\mathbf{X}_{NLP}$, that needs to be defined. An inspection of the NLP problem represented by Eqs. (15)–(19) suggests that this Jacobian matrix has a structure that can be exploited by looking at its parts. This is the subject of this section. In the most general case, considering n_s states, n_c controls, n_g constraints, n collocation points and unknown final time t_f, the Jacobian **Jac** associated with the transcription of an autonomous system of equations will be expressed as a matrix having the following dimensions

$$\dim(\mathbf{Jac}) = \left[n \cdot (n_s + n_g) + 1\right] \times \left[(n+1) \cdot n_s + n \cdot n_c + 1\right]. \tag{21}$$

To maintain a consistency between the states and the controls associated with each node, the following discrete state vector $\mathbf{X}_{NLP}$ is proposed.[3]

$$\mathbf{X}_{NLP} = \left\{ \mathbf{X}_0 \middle| \mathbf{X}_1 \ \mathbf{U}_1 \middle| \mathbf{X}_2 \ \mathbf{U}_2 \middle| .. \ .. \middle| \mathbf{X}_n \ \mathbf{U}_n \middle| t_f \middle| \right\}^T \tag{22}$$

We can observe how the initial control $\mathbf{U_0}$ does not appear in Eq. (22). This is due to the choice of the fRPM as transcription method instead of the traditional RPM. The initial control indeed can be extrapolated once the NLP is solved. The Jacobian is by definition the matrix representing the partial derivatives of a given set of functions $\mathbf{C}(\mathbf{X}_{NLP})$ (i.e., our NLP constraints), which include the cost function J, the dynamics $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$, and, when defined, the constraints $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n\}$, so we can write

$$\mathbf{C}(\mathbf{X}_{NLP}) = \left\{ J \middle| \mathbf{f}_1 \ \mathbf{f}_2 \ldots \mathbf{f}_n \middle| \mathbf{g}_1 \ \mathbf{g}_2 \ldots \mathbf{g}_n \middle| \right\}^T \tag{23}$$

[3] Remark 3: Note that the final element is represented by t_f, in case the problem has open final time. If not this variable is removed from the vector $\mathbf{X}_{NLP}$.

and the corresponding Jacobian matrix is

$$\mathbf{J} = \left[\frac{\partial \mathbf{C}}{\partial \mathbf{X}_{NLP}}\right] = \left[\begin{array}{c|cc|cc|c|cc|c}
\frac{\partial J}{\partial \mathbf{X}_0} & \frac{\partial J}{\partial \mathbf{X}_1} & \frac{\partial J}{\partial \mathbf{U}_1} & \frac{\partial J}{\partial \mathbf{X}_2} & \frac{\partial J}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial J}{\partial \mathbf{X}_n} & \frac{\partial J}{\partial \mathbf{U}_n} & \frac{\partial J}{\partial t_f} \\
\frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_n} & \frac{\partial \mathbf{f}_1}{\partial t_f} \\
\frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_n} & \frac{\partial \mathbf{f}_2}{\partial t_f} \\
\cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\
\frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_n} & \frac{\partial \mathbf{f}_n}{\partial t_f} \\
\frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_n} & \frac{\partial \mathbf{g}_1}{\partial t_f} \\
\cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\
\frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_n} & \frac{\partial \mathbf{g}_n}{\partial t_f}
\end{array}\right] \tag{24}$$

This matrix can be computed numerically in different ways (e.g., analytically or with the classical finite-differences schemes). However, these are not the best approaches since they do not consider the theoretical knowledge contained in the definition of the discrete operator **D**, nor do they take full advantage of the intrinsic sparsity associated with the use of PSMs. Instead, we propose to express the Jacobian matrix as sum of three different contributions.

$$\mathbf{Jac} = \mathbf{Jac}_{Ps} + \mathbf{Jac}_{Du} + \mathbf{Jac}_{Th} \tag{25}$$

We can now analyze each of these terms and describe how to compute them.

4.1 Pseudospectral Jacobian

This part of the Jacobian matrix is intrinsically related to the use of the fRPM. More specifically, it can be seen as the contribution to the Jacobian and to the constraints represented in Eq. (24) given by the use of the discrete differential matrix **D**. In the frame of the discretization of the dynamics, it represents the term

$$\mathbf{D} \cdot \mathbf{X} \tag{26}$$

From a pure algebraic point of view, the differential operator can be seen as a set of linear combinations of the nodal values of each of the states. The Pseudospectral Jacobian is entirely defined once the matrix **D** is computed and expanded. More explicitly, it can be defined as follows

$$\mathbf{Jac}_{Ps} = \left[\begin{array}{ccccc} & & \mathbf{O}_{1\times[(n+1)\cdot n_s+n\cdot n_c+1]} & & \\ \mathbf{D}_{1,0} & .. & .. & \mathbf{D}_{1,n} & \\ .. & .. & .. & .. & \mathbf{O}_{[n\cdot(n_s+n_g)+1\times 1]} \\ \mathbf{D}_{n,0} & .. & .. & \mathbf{D}_{n,n} & \\ & & \mathbf{O}_{n\cdot n_g\times[(n+1)\cdot n_s+n\cdot n_c+1]} & & \end{array}\right] \tag{27}$$

where

$$\mathbf{D}_{i,j} = D_{i,j} \cdot \mathbf{I}_{n_s}, \ \ i \in [1, n], \ \ j \in [0, n] \tag{28}$$

and $\mathbf{I}_{n_s}$ is the identity matrix of dimension n_s. The elements $D_{i,j}$ are the time derivative of the polynomials defined in Eq. (7), evaluated in the collocation nodes. The Pseudospectral Jacobian can then be entirely computed just once, before the beginning of the real optimization process. Moreover, the accuracy of its computation is a consequence of how good the estimate of the roots of the Legendre-Radau Polynomials is, and not of the errors given by the approximation due to the use of numerical differentiation techniques.

4.2 Dual Jacobian

The Dual Jacobian refers to the cost function of Eq. (15), the right-hand side of the differential equations of Eq. (16), and the path constraints of Eq. (17). This contribution is computed by using the dual number theory, which will be briefly described in the next section.

4.2.1 Dual Numbers

In linear algebra, the dual numbers extend the real numbers by adjoining one new element ε with the property $\varepsilon^2 = 0$ (ε is nilpotent). The collection of dual numbers forms a particular two-dimensional commutative associative algebra over the real numbers [5]. Every dual number has the form

$$z = a + b\varepsilon \tag{29}$$

with a and b uniquely determined real numbers and, in particular,

$$\begin{array}{ll} a = real(z), & \text{Real Part} \\ b = dual(z), & \text{Dual Part} \end{array}$$

Dual numbers extend the real numbers in a similar way to the complex numbers. Indeed, as the dual numbers, the complex numbers adjoin a new element i, for which $i^2 = -1$, and every complex number has the form $z = a + bi$ where a and b are real numbers. The definition given in Eq. (29) relies on the idea that $\varepsilon^2 = 0$ with $\varepsilon \neq 0$.

To implement the dual numbers, algebraic operations on these numbers should be properly defined. It is important to underline that the dual number algebra is a non-division algebra; given two dual numbers, division is possible only if the real part of the divisor is different from zero. The dual numbers have been implemented in MATLAB as a new class of numbers [3], using operator overloading. The class includes definitions for standard algebraic operations, logical comparison operations, and other more general functions such as the exponential or the trigonometric functions. This class definition file allows a real-valued analysis code to be easily converted to operate on dual numbers by just changing the variable type declarations, while the structure of the code remains unchanged. The use of the dual numbers allows us to compute exact first derivatives, as it will be explained in the next section.

4.2.2 Dual-Step Differentiation Method

The dual-step differentiation method uses the dual numbers to provide exact first order derivatives. Consider the Taylor series of a function $f(x)$ for $x \in \mathbb{R}$ for a given perturbation value a.

$$f(x+a) = f(x) + af'(x) + \frac{1}{2!}a^2 f''(x) + \frac{a^3 f'''(x)}{3!} + \dots. \tag{30}$$

If we assume that the perturbation a is the dual number

$$a = a_1\varepsilon \qquad \text{with} \qquad \varepsilon^2 = 0 \qquad \text{and} \qquad \varepsilon \neq 0 \tag{31}$$

we can expand in Taylor series around the center x the function $f(x)$ by using a dual step, so that $a^2 = 0$, $a^3 = 0, \dots$, and the Taylor series in Eq. (30) truncates exactly at the first-derivative term, yielding the properties of the approximation that we are seeking:

$$f(x+a) = f(x) + a_1 f'(x)\varepsilon. \tag{32}$$

So, to get $f'(x)$ it is necessary to simply read off the ε component and divide by a_1, yielding the dual-step first derivative formula.[4]

$$f'(x) = \frac{Dual[f(x+a)]}{a_1}. \tag{33}$$

[4] From the inspection of Eq. (32) it is possible to observe that each function extended in the dual plane *hides* its derivative in its dual part. Indeed, the dual-number algebra is such that, when operations are carried out on the real part of the number, derivative information for those operations is formed and stored in the non-real part of the number. The disadvantage is a larger computational cost and, in addition, the need of working with analytical functions.

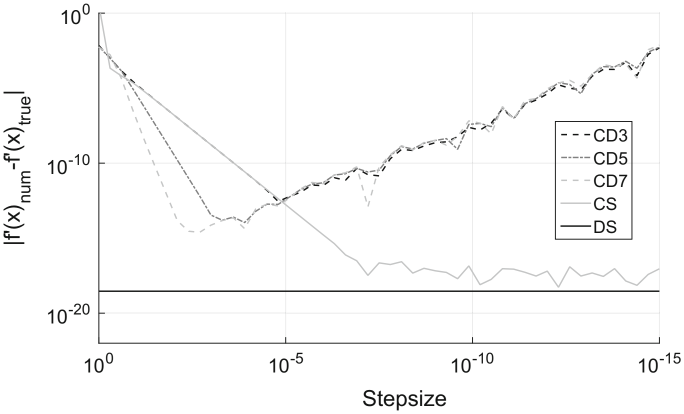


Fig. 2 Comparison of numerical methods for first-derivative computation of the function $2e^{-t^4}\sin(t)$

This formula clearly shows the advantages of the use of the dual-step differentiation method over the central difference and the complex-step approximations. Indeed, since the dual-step derivative approximation does not involve a difference operation and no terms of the Taylor series are ignored, this formula is subject neither to truncation error, nor to round-off error. There is no need to make the step size small and the simplest choice is $a_1 = 1$, which eliminates the need to divide by the step size. Therefore, using the dual-step method, the error between numerical and analytical derivative ($\eta = |f' - f'_{ref}|/|f'_{ref}|$) is machine zero regardless of the selected step size, as illustrated in Fig. 2, where the derivative of the function $2e^{-t^4}\sin(t)$ is computed by using central difference schemes with 3, 5 and 7 points (CD3, CD5, CD7), complex-step (CS), and dual-step (DS). It is clear that the dual-step approach provides exact results, even in presence of highly nonlinear functions. Indeed, considering the central difference (CD) and the complex-step approximations, instead, Fig. 2 shows that, as the stepsize decreases, the error decreases according to the order of the truncation error of the method. However, after a certain value the error for the central difference approximations tends to grow, while the error for the complex-step approximation continuously decreases. This shows the effect of the round-off error, which affects the central differences but not the first derivative complex-step approximation. While central differences and complex-step provide approximated derivatives for these terms, the use of Dual Number Theory permits the computation of zero-epsilon derivatives. The only limit for the use of this technique is the same associated with the use of the complex-step, that is, the need to have analytical functions, i.e., no look-up tables are allowed.

4.2.3 Dual-Number Based Jacobian

In case we are dealing with analytical functions, it is possible to compute their contribution to the Jacobian matrix (i.e., considering the matrix $\mathbf{D}$ equal to $\mathbf{0}$), excluding the last column,

$$\mathbf{Jac}_{Du} = \left[\frac{\partial \mathbf{C}}{\partial \mathbf{X}_{NLP}}\right]_{\mathbf{D}=\mathbf{0}} = -k_t \left[\begin{array}{ccccccccc} & \frac{\partial J}{\partial \mathbf{X}_1} & \frac{\partial J}{\partial \mathbf{U}_1} & \frac{\partial J}{\partial \mathbf{X}_2} & \frac{\partial J}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial J}{\partial \mathbf{X}_n} & \frac{\partial J}{\partial \mathbf{U}_n} & \\ & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_n} & \\ & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_n} & \\ \mathbf{O}_{[n\cdot(n_s+n_g)+1\times n_s]} & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \mathbf{O}_{[n\cdot(n_s+n_g)+1\times 1]} \\ & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{f}_n}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{f}_n}{\partial \mathbf{U}_n} & \\ & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_n} & \\ & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_2} & \cdot\cdot\ \cdot\cdot & \frac{\partial \mathbf{g}_n}{\partial \mathbf{X}_n} & \frac{\partial \mathbf{g}_n}{\partial \mathbf{U}_n} & \end{array}\right] \tag{34}$$

where k_t is equal to the time scale factor defined as $\frac{t_f - t_0}{2}$ for the elements related to the functions $\mathbf{f}$, and equal to 1 for all the other terms of $\mathbf{Jac}_{Du}$. Each of the elements of $\mathbf{Jac}_{Du}$ can be rewritten in dual form. We can therefore write

$$\mathbf{Jac}_{Du} = -k_t Dual\left[\mathbf{C}(\mathbf{X}_{NLP} + \varepsilon)\right]_{\mathbf{D}=\mathbf{0}} \tag{35}$$

The differentiation operation becomes then an evaluation of the single elements of $\mathbf{C}(\mathbf{X}_{NLP})$ in dual sense, and the extraction of the dual part.

4.3 Theoretical Jacobian

Finally, a third contribution, the Theoretical Jacobian, arises in case we deal with problems having an open final time. In this case the NLP state vector $\mathbf{X}_{NLP}$ will have a further variable, that is t_f. The Jacobian associated with this term is simply proportional to the output of the continuous functions by a factor $\frac{t_f - t_0}{2}$ introduced in virtue of the mapping between physical and pseudospectral time of Eq. (10).

$$\mathbf{Jac}_{Th} = -\frac{1}{2}\left[\begin{array}{cc} & 0 \\ & \mathbf{f}_1 \\ & \mathbf{f}_2 \\ \mathbf{O}_{[n\cdot(n_s+n_g)+1]\times[(n+1)\cdot n_s+n\cdot n_c]} & \cdot\cdot \\ & \mathbf{f}_n \\ & \mathbf{O}_{n\cdot n_g\times 1} \end{array}\right] \tag{36}$$

The hybridization of the Jacobian matrix makes the computation of the NLP solution more accurate, as no approximations are taken, except those associated with the transcription process. Hence, significant CPU time is saved when solving the NLP problem, as we will see.

5 Automatic Scaling: Projected Jacobian Rows Normalization

Let us reformulate the NLP of Eqs. (15)–(19). If we group the differential constraints $\mathbf{f}_i$, and the algebraic constraints $\mathbf{g}_i$, $i = 1, \ldots, n$ as

$$\mathbf{F} = \left\{ \mathbf{f}_1 \ \mathbf{f}_2 \ \ldots \ \mathbf{f}_n \right\}^T, \ \mathbf{G} = \left\{ \mathbf{g}_1 \ \mathbf{g}_2 \ \ldots \ \mathbf{g}_n \right\}^T \tag{37}$$

the core of the NLP can be rewritten in the following compact form as function of the vector $\mathbf{X}$.[5]

$$\begin{gathered} \min \ J(\mathbf{X}), \\ \text{s.t. } \mathbf{F}(\mathbf{X}) = 0 \\ \mathbf{g}_L \leq \mathbf{G}(\mathbf{X}) \leq \mathbf{g}_U \\ \mathbf{X}_L \leq \mathbf{X} \leq \mathbf{X}_U \end{gathered} \tag{38}$$

A measure of the quality of a scaling method is the condition number (C.N.) of the Jacobian of the NLP (38), which in the general case is a rectangular matrix given by Eq. (20). Since the Jacobian matrix is involved in the KKT conditions required to solve the NLP, a well-conditioned Jacobian is essential for solving the problem defined in Eq. (38) without excessive rounding errors. This implies that the scaling is not a secondary aspect in the transcription of the optimal control problems. Note that the effective scaling involves two steps: the scaling of the states $\mathbf{X}$, which will be transformed into scaled states $\tilde{\mathbf{X}}$, and the scaling of the constraints $\mathbf{F}$, transformed into the corresponding $\tilde{\mathbf{F}}$. Their combination will result in the scaling of the Jacobian matrix.

5.1 Scaling of NLP States

The states $\mathbf{X}$ of the NLP problem are scaled using the standard linear transformation given in [2], regardless of the NLP scaling method that we use. Specifically, the scaled state $\tilde{\mathbf{X}}$ is given by

$$\tilde{\mathbf{X}} = \mathbf{K}_\mathbf{x} \cdot \mathbf{X} + \mathbf{b}_\mathbf{x} \tag{39}$$

[5] Here $\mathbf{X}$ is meant to be the one defined in Eq. (22), with the subscript dropped to avoid heavy notation.

where $\mathbf{K_x}$ is a diagonal matrix, and $\mathbf{b_x}$ is a vector having the same dimensions as $\mathbf{X}$. Since we always deal with bounded states and control, the diagonal elements of the matrices $\mathbf{K_x}$ and $\mathbf{b_x}$ are defined as:

$$\mathbf{K_x}_{i,i} = \frac{1}{\mathbf{X_U}_i - \mathbf{X_L}_i}, \ \mathbf{b_x}_i = -\frac{\mathbf{X_L}_i}{\mathbf{X_U}_i - \mathbf{X_L}_i} \tag{40}$$

Note that the transformation (40) yields scaled states $\tilde{\mathbf{X}}$ which always lie in the interval [0, 1]. In case of unbounded states, artificial upper and lower boundaries are usually introduced [2].

5.2 Constraints Scaling - State of the Art

Linear scaling techniques use a scaling of the form (41).

$$\tilde{\mathbf{F}} = \mathbf{K_f} \cdot \mathbf{F}, \ \tilde{\mathbf{G}} = \mathbf{K_g} \cdot \mathbf{G} \tag{41}$$

$\mathbf{K_f}$ and $\mathbf{K_g}$ are diagonal matrices. The isoscaling (IS) method is one such technique whereby the constraints $\mathbf{F}$ are scaled exactly like the states, that is,

$$\mathbf{K_f} = \mathbf{K_x},$$

where $\mathbf{K_x}$ is given by Eq. (40), see [2–11]. Note that isoscaling does not help in scaling the constraints $\mathbf{G}$. A possible refinement of this approach has been suggested by Rao [12], who uses randomly sampled points around the vector $\mathbf{X}$, and computes the mean of the norms of the Jacobian rows of $\mathbf{F}$ and $\mathbf{G}$ instead of the norm of the Jacobian rows. Unfortunately, this technique increases the CPU time needed to compute the scaling coefficients, since the Jacobian matrix must be evaluated many more times. Next, we introduce a simple linear scaling technique which does not require additional Jacobian matrix evaluations, and hence is less computationally expensive.

5.3 Projected Jacobian Rows Normalization

Isoscaling bases the scaling of the constraints solely on the scaling of the states. In other words, it does not take into account the relationship between the states and the constraints, which is represented in linearized form by the Jacobian matrix. Conversely, Jacobian rows normalization (JRN) only considers this relationship, without involving the states' normalization in the process. Specifically, in the JRN technique, the diagonal elements of $\mathbf{K_f}$ and $\mathbf{K_g}$ are given by

$$\mathbf{K_f}_{i,i} = \underset{k}{\text{mean}} \frac{1}{|\nabla \mathbf{F}|_i}, \ \mathbf{K_g}_{i,i} = \underset{k}{\text{mean}} \frac{1}{|\nabla \mathbf{G}|_i} \tag{42}$$

where k represents the number of random samples generated to compute the scaling factors. The projected Jacobian rows normalization (PJRN) technique which we propose considers both the states and the constraints' magnitude. Specifically, in the PJRN, the diagonal elements of $\mathbf{K_f}$ and $\mathbf{K_g}$ are given by

$$\mathbf{K_f}_{i,i} = \frac{1}{\left|\nabla \mathbf{F} \cdot \mathbf{K_x^{-1}}\right|_i}, \ \mathbf{K_g}_{i,i} = \frac{1}{\left|\nabla \mathbf{G} \cdot \mathbf{K_x^{-1}}\right|_i} \tag{43}$$

and this scaling generally leads to a better-conditioned Jacobian matrix, and to more uniformly distributed singular values. The Jacobian of the PJRN-scaled NLP can be therefore computed as

$$\tilde{\mathbf{Jac}} = \begin{Bmatrix} \tilde{\nabla} \tilde{J} \\ \tilde{\nabla} \tilde{\mathbf{F}} \\ \tilde{\nabla} \tilde{\mathbf{G}} \end{Bmatrix} = \begin{Bmatrix} K_J \cdot \nabla J \cdot \mathbf{K_x^{-1}} \\ \mathbf{K_f} \cdot \nabla \mathbf{F} \cdot \mathbf{K_x^{-1}} \\ \mathbf{K_g} \cdot \nabla \mathbf{G} \cdot \mathbf{K_x^{-1}} \end{Bmatrix} \tag{44}$$

where K_J is a parameter which normalizes the cost function J. $\mathbf{K_x}$ is given by using Eq. (40), while $\mathbf{K_f}$ and $\mathbf{K_g}$ are computed by using Eq. (42). Note that K_J can be either manually selected, or automatically computed by means of the PJRN, which is the choice adopted in this work. This completes the self-scaling procedure. We can observe the effects of the hybridization of the Jacobian matrix, and the self-scaling procedure in two significant examples, illustrated in the next section.

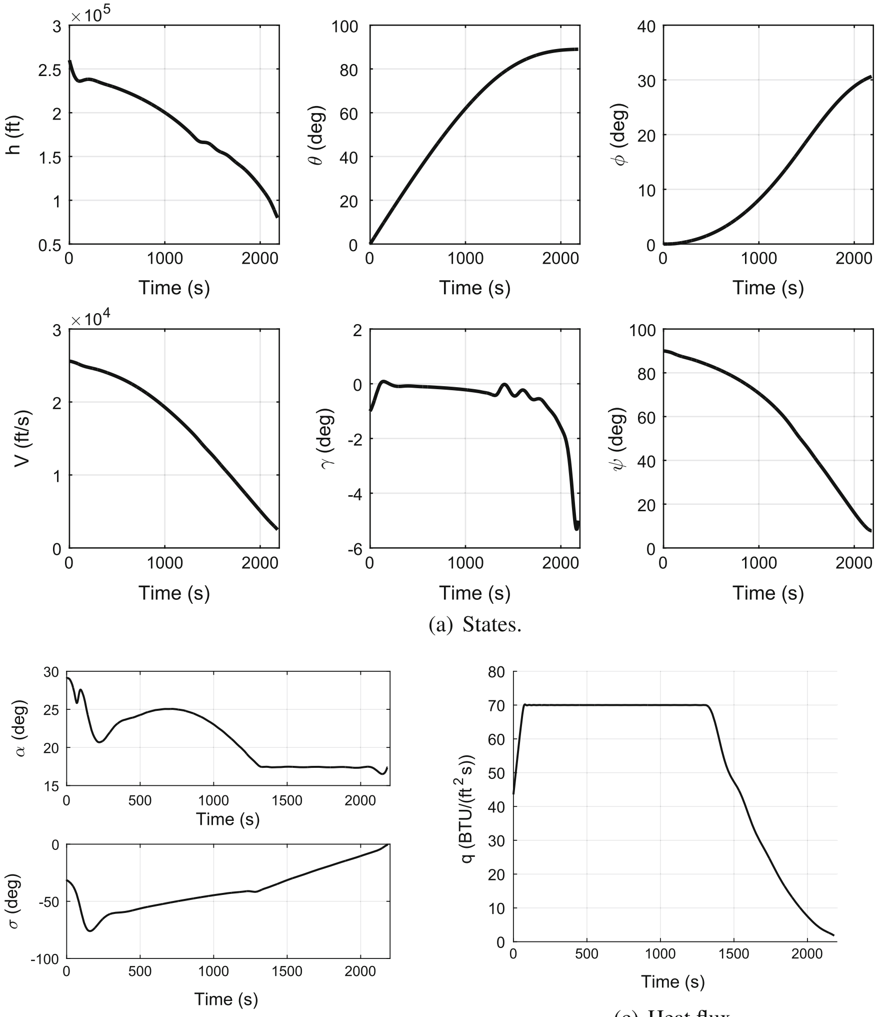
6 Numerical Examples

Two examples are proposed to show the application of the proposed improved pseudospectral method for EDL applications, specifically. In the first example the optimal Space Shuttle entry guidance problem is solved. In the second example the asteroid descent and landing problem for a soft touchdown on an asteroid is shown.[6]

6.1 Entry: Space Shuttle Guidance

This problem deals with the maximization of the crossrange (corresponding in this case to the latitude) of the Space Shuttle during the atmospheric entry phase, while satisfying the maximum heat-rate limit, and final conditions on altitude, velocity and

[6] The tests are repeated three times for a better characterization of the CPU times.



(a) States.

(b) Controls.

(c) Heat flux.

Fig. 3 Space Shuttle entry guidance example

flight-path angle. A full description of the problem can be found in [2]. Results are depicted in Fig. 3a–c.

It can be observed that the results are fully consistent with the ones generated by Betts. The combined use of the hybrid jacobian and scaling techniques leads to accurate and faster results with respect to the standard methods. Indeed, the solution is generated in 8.81 and 9.72 s when JRN and PJRN techniques are used in comparison to 51 and 54 s respectively, when the non-hybrid Jacobian is employed.

The differences becomes larger when IPOPT is used (73 and 155 s when the JRN and PJRN with the hybrid jacobian are used, respectively), versus 817 and 570 s when no knowledge of the Jacobian matrix is exploited. The use of scaling techniques improves the initial conditioning of the problem by several orders of magnitude (the C.N. goes from $7.468 \cdot 10^9$ to 418 for the PJRN and 24813 when the JRN is adopted).

6.2 Descent and Landing: JAXA-DLR Trojan Mission

This problem deals with the descent and landing of a small lander, in the frame of a JAXA-DLR joint-study for the design of a mission targeting Jupiter's Trojan asteroids. The objective is the maximization of the final mass, while having a synchronized, soft touchdown of the lander on the asteroid surface's nadir point at the beginning of the descent phase. To guarantee the synchronization, together with radial position r, radial and tangential components of velocity V_r and V_t, the motion is described by using the relative angle θ_{ast} w.r.t. the landing point, fixed on the surface of the asteroid. The in-plane dynamics is therefore described as

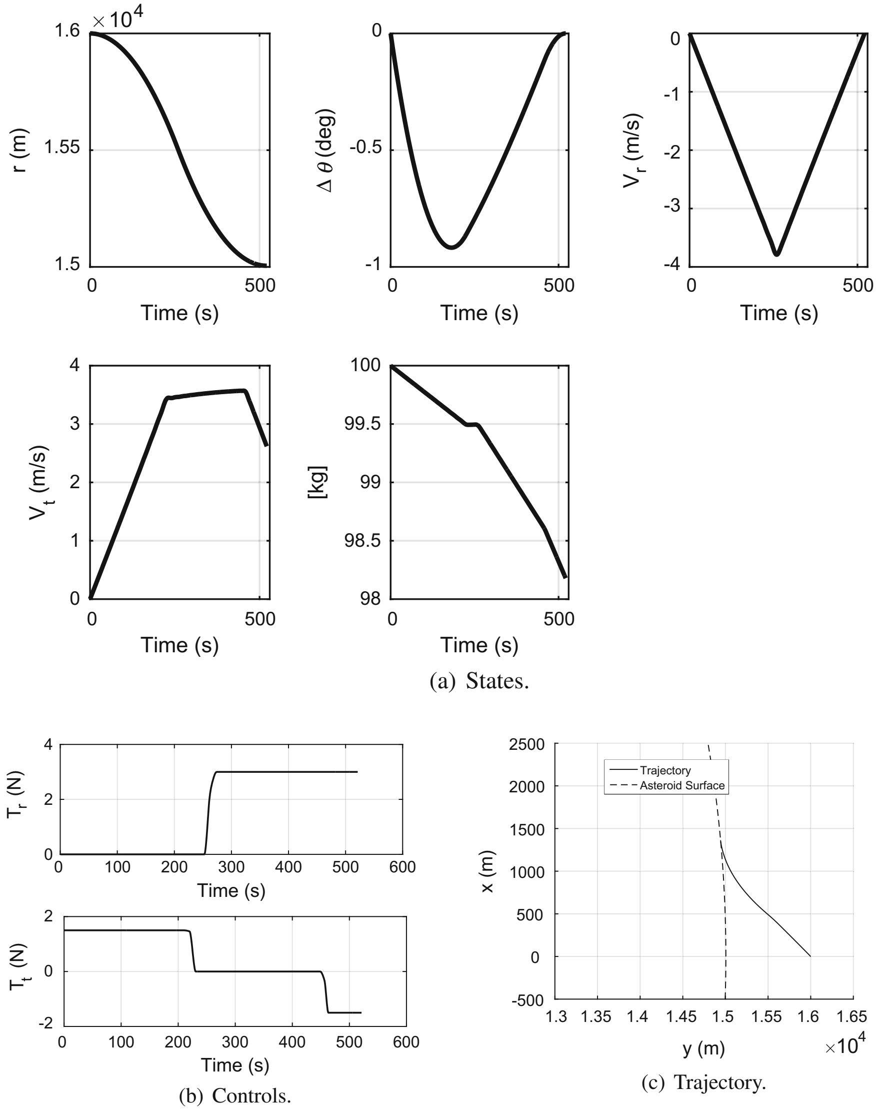
$$\begin{aligned}
\dot{r} &= V_r \\
\dot{\theta}_{ast} &= \frac{V_t}{r} - \omega_{ast} \\
\dot{V}_r &= \frac{V_t^2}{r} - \frac{\mu_{ast}}{r^2} + \frac{T_r}{m} \\
\dot{V}_t &= -\frac{V_r V_t}{r} + \frac{T_t}{m} \\
\dot{m} &= \frac{\|T_r\| + \|T_t\|}{I_{sp} g_0}
\end{aligned} \tag{45}$$

with the asteroid having a gravitational parameter μ_{ast} equal to 3.774 10^6 m^3/s^2, and a radius of 15 km. The initial altitude w.r.t. the surface is 1 km. The final desired altitude is 5 m, without side-velocity w.r.t. the landing point. The lander initial mass is 100 kg with a specific impulse of 68 s. Limits equal to 3 N for the radial component of the thrust and 1.5 for the tangential one are also taken into account.

$$\begin{aligned}
|T_r| &\leq 3 \quad \text{N} \\
|T_t| &\leq 1.5 \ \text{N}
\end{aligned} \tag{46}$$

Results are depicted in Fig. 4a–c, where states, controls, and the trajectory are shown.

One can see that the solution follows a bang-bang structure, as expected. At the end of the mission the relative angle θ_{ast} is equal to 0, and the tangential velocity V_t becomes exactly equal to the one of the landing point on the asteroid surface. Radially, the gravity is initially exploited to accelerate the lander towards the asteroid. The



(a) States.

(b) Controls.

(c) Trajectory.

Fig. 4 Trojan asteroid descent and landing example

thrusters are only used to decelerate the lander during the second-half of the descent, to guarantee a soft touchdown. This solution maximizes the remaining propellant at the cost of relatively large time, equal to 521 s. The trajectory confirms that the lander smoothly reaches the prescribed final point. In terms of numerical performance, if we look at the solution having 50 nodes the use of the PJRN permits to reduce the condition number to 170.5 (while the unscaled condition number is equal to

19400.1 and the JRN generates a condition number equal to 2564.1. Therefore, the condition number is improved by more than one order of magnitude with respect to standard literature methods. The CPU time required to compute a valid solution by using the hybrid Jacobian structure is equal to 43.3 s (PJRN) and 34.7 s (JRN). If we do not exploit the Jacobian structure the required time to compute an optimal solution is equal to 331.4 s (PJRN) and 179.1 s (JRN), respectively. In this case the PJRN does not make so much difference with respect to the JRN, while the hybrid Jacobian computation is highly effective in reducing the required CPU time. Note that when the Jacobian structure is exploited the time is dramatically reduced even if the dual-number class, involving more operations, is invoked.

7 Conclusions

In this paper we have given an overview on pseudospectral methods, together with some key-improvements with respect to the standard transcription described in literature, which include an exploitation of the Jacobian matrix, exactly computed by using the dual-number theory, and a self-scaling approach, which ensures a better numerical conditioning of the problem we want to solve. The proposed techniques have been implemented in SPARTAN, the first European tool implementing the flipped-Radau pseudospectral method. The method, and the tool can be used for preliminary analysis of complex, nonlinear problems involving entry, descent, and landing applications. As examples an entry mission, based on the Space Shuttle entry guidance, and a descent and landing mission for a soft-touchdown on a Trojan asteroid, have been implemented.

Results show the validity of SPARTAN as state-of-the-art tool for entry, descent, and landing guidance analysis, leading to reduced CPU time with respect to standard methods, and a significant improvement of the numerical conditioning of the problems, here measured by the condition number. Its generic structure encourages the use for further scenarios and problems involving complex dynamics and multiple constraints, where no analytical solutions are available, e.g., lunar landing or mars descent missions. Future work will include the extension of the method to deal with uncertainties to compute stochastic optimal trajectories, and the implementation of new cases (e.g., low-thrust interplanetary maneuvers) to explore the use of the tool in different scenarios.

References

1. Abramovitz M, Stegun IA (1695) Handbook of mathematical functions. Dover Publications, New York
2. Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia

3. D'Onofrio V (2015) Implementation of advanced differentiation methods for optimal trajectory computation. Master's thesis, University of Naples Federico II, Naples
4. Elissar (2015) Description of dido optimal control software. http://www.elissarglobal.com/
5. Fike J, Alonso J (2011) The development of hyper-dual numbers for exact second-derivative calculations. In: 49th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, Orlando, USA, number AIAA, p 886
6. Garg D (2011) Advances in global pseudospectral methods for optimal control. PhD thesis, University of Florida, Gainesville
7. Gill PE, Murray W, Saunders MA (2008) User's guide for SNOPT version 7: software for large-scale nonlinear programming. University of California, San Diego
8. Gong Q, Ross IM, Kang W, Fahroo F (2008) Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control. Comput Optim Appl. https://doi.org/10.1007/s10589-007-9102-4
9. Huneker L, Sagliano M, Arslantas YE (2015) Spartan: an improved global pseudospectral algorithm for high-fidelity entry-descent-landing guidance analysis. In: 30th international symposium on space technology and science, Japan, Kobe, 2015
10. Martins JRR, Sturdza P, Alonso JJ (2003) The complex-step derivative approximation. ACM Trans Math Softw 29(3):245–262. https://doi.org/10.1145/838250.838251
11. Nocedal J, Wright SJ (1999) Numerical optimization. Springer, New York
12. Rao AV (2009) A survey of numerical methods for optimal control. In: AAS/AIAA astrodynamics specialist conference, aaS paper 09–334, Pittsburgh, PA, August 10–13, 2009
13. Ross IM, Sekhavat P, Fleming A, Gong Q (2006) Pseudospectral feedback control: foundations, examples and experimental results. In: AIAA guidance, navigation, and control conference, Keystone, USA. https://doi.org/10.2514/6.2006-6354
14. Sagliano M (2014) Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems. Oper Res Lett 42(3):213–216. https://doi.org/10.1016/j.orl.2014.03.003
15. Sagliano, M (2016) Development of a novel algorithm for high performance reentry guidance. PhD thesis. http://elib.suub.uni-bremen.de/edocs/00105082-1.pdf
16. Sagliano M, Theil S (2013) Hybrid jacobian computation for fast optimal trajectories generation. In: AIAA guidance, navigation, and control conference, Boston, USA. https://doi.org/10.2514/6.2013-4554
17. Wächter A, Biegler LT (2006) On the implementation of an interior-point filter linesearch algorithm for large-scale nonlinear programming. Math Program 106(1):25–57; Springer, New York

Maximum Null Motion Algorithm for Single Gimbal Control Moment Gyroscopes

S.A.V. Schallig, Q.P. Chu, S.W. Rhee and E. van Kampen

1 Introduction

Future space platforms, such as next-generation Earth imaging and missile-tracking satellites, require rapid rotational maneuverability [16, 19]. Currently, most European satellites use Reaction Wheels (RWs) as momentum exchange device for their Attitude Control Systems (ACS) [12]. However, RWs do not produce enough torque to meet the high slew rate requirement of agile satellites. An alternative is to use Control Moment Gyroscopes (CMGs), because of their torque amplifying properties [16].

One type of CMG is the Single Gimbal Control Moment Gyroscope (SGCMG). SGCMGs are momentum exchange devices which contain a spinning rotor with an angular momentum. The spinning rotor is mounted on a gimbal, which allows for rotation of the angular momentum vector of the rotor with respect to the spacecraft. Gimbaling of the spinning rotor produces a precessional gyroscopic reaction torque orthogonal to both the rotor spin and gimbal axis. A small gimbal torque input produces a large gyroscopic torque output on the spacecraft, which is why CMGs are torque amplification devices [16]. Usually a cluster of multiple CMGs is used to obtain the desired torque, where the total torque vector is the summation of the individual CMG outputs torque vectors [6].

S.A.V. Schallig (✉) · Q.P. Chu · E. van Kampen
Delft University of Technology, Delft, The Netherlands
e-mail: S.A.V.Schallig@student.tudelft.nl

Q.P. Chu
e-mail: Q.P.Chu@TUDelft.nl

E. van Kampen
e-mail: E.vanKampen@TUDelft.nl

S.W. Rhee
Korea Aerospace Research Institute, Yuseong, Korea
e-mail: srhee@kari.re.kr

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_37

Historically, CMGs were only used for space stations and military satellites. Space stations have a high inertia which requires high torques from the ACS. This high torque requirement together with a moving crew inside the space station, make CMGs ideal for the attitude and stability control. CMGs have been used successfully in Skylab, MIR and the International Space Station [2, 12, 13, 17]. Military satellites often require rapid multitarget acquisition, pointing, and tracking capabilities in order to maximize the number of pictures/day and to be able to quickly target areas of interest or following moving objects on the ground [12, 19].

Although there are several advantages for using CMGs, there have not been many implementations of CMG systems due to the complexity of the mechanical- and control system. Recent advances in digital design tools and micro electronics make SGCMGs a feasible alternative to RWs even for small satellites [3]. However, the inherent singularity problem of SGCMGs is still a major issue.

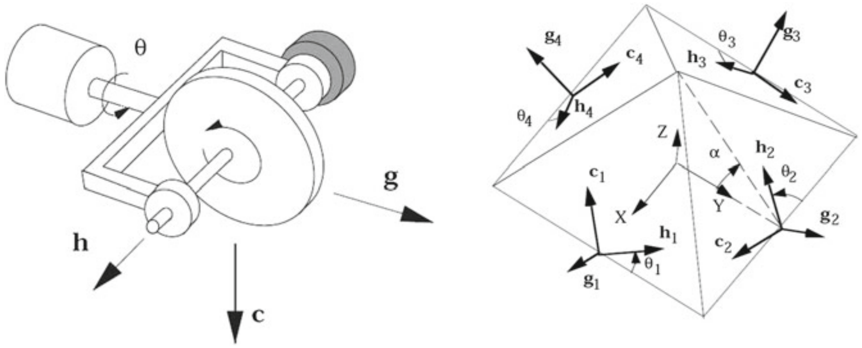
The structure of this paper is as follows. In Sect. 2, the basics of CMGs will be introduced together with the four CMG pyramid configuration. This is followed by the spacecraft model used during simulation in Sect. 3. The model consists of the kinematic and dynamics equations of motion together with the state estimation and attitude controller used. Section 4 will give two steering laws which generate torque error to pass singularities as well as two null motion algorithms. The metrics used to evaluate the performance of the steering laws and null motion algorithms are given in Sect. 5. Finally, numerical simulations of the steering laws and null motion algorithms are given in Sect. 6.

2 Single Gimbal Control Moment Gyroscopes

A schematic overview of a SGCMG is shown in Fig. 1a. This figure defines the fixed unit vector along gimbal axis g, angular momentum vector h, torque vector c and gimbal angle θ. The relation between these quantities is given by Eq. 1.

$$c_i \equiv \frac{dh_i}{dt} = \frac{dh_i}{d\theta_i} = g_i \times h_i \tag{1}$$

It is assumed that each CMG has the same moment of inertia and angular rate, hence the magnitude of the angular momentum is the same for each CMG. Usually a system of several CMGs is used for the attitude control. A common configuration is the four CMG pyramid configuration shown in Fig. 1b. This configuration will also be used throughout this paper because it is a minimum redundant system with full three dimensional control and has a compact size which makes it ideal for satellite applications. When the gimbal accelerations are neglected, the total angular momentum $\boldsymbol{h} \in R^{3\times1}$ of this configuration in de body reference frame is given by



(a) Schematic Overview[6]

(b) Four CMG Pyramid Configuration[6]

Fig. 1 Single gimbal control moment gyroscope

$$\boldsymbol{h}(\boldsymbol{\theta}) = \sum_{i=1}^{4} C_i h_i(\theta_i) = \begin{bmatrix} -c\alpha \sin\theta_1 \\ \cos\theta_1 \\ s\alpha \sin\theta_1 \end{bmatrix} + \begin{bmatrix} -\cos\theta_2 \\ -c\alpha \sin\theta_2 \\ s\alpha \sin\theta_2 \end{bmatrix} + \begin{bmatrix} c\alpha \sin\theta_3 \\ -\cos\theta_3 \\ s\alpha \sin\theta_3 \end{bmatrix} + \begin{bmatrix} \cos\theta_4 \\ c\alpha \sin\theta_4 \\ s\alpha \sin\theta_4 \end{bmatrix} \tag{2}$$

The output torque of the CMGs equals to the change in angular momentum

$$\dot{\boldsymbol{h}} = \frac{d\boldsymbol{h}}{dt} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \cdot \frac{d\boldsymbol{\theta}}{dt} = C\dot{\boldsymbol{\theta}} \tag{3}$$

where C is the Jacobian matrix of Eq. 2. From Eqs. 1 and 2 it follows that the columns of C equal to the torque vectors of the CMGs

$$C \equiv \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} = [c_1\ c_2\ c_3\ c_4] = \begin{bmatrix} -c\alpha\cos\theta_1 & \sin\theta_2 & c\alpha\cos\theta_3 & -\sin\theta_4 \\ -\sin\theta_1 & -c\alpha\cos\theta_2 & \sin\theta_3 & c\alpha\cos\theta_4 \\ s\alpha\cos\theta_1 & s\alpha\cos\theta_2 & s\alpha\cos\theta_3 & s\alpha\cos\theta_4 \end{bmatrix} \tag{4}$$

The CMG model includes gimbal dynamics and is simulated as a second order system

$$\dot{\boldsymbol{\theta}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\dot{\boldsymbol{\theta}}_c \tag{5}$$

where $\dot{\boldsymbol{\theta}}$ is the gimbal rate and $\dot{\boldsymbol{\theta}}_c$ is the commanded gimbal rate. A CMG steering law can be constructed where an attitude controller is used to calculate the required torque $\boldsymbol{h}_{req}$. By using the pseudo-inverse of Eq. 3, the desired gimbal rates can be calculated using

$$\dot{\boldsymbol{\theta}}_c = C^T \left(CC^T\right)^{-1} \dot{\boldsymbol{h}}_{req} \tag{6}$$

This method is called the Pseudo Inverse (PI) steering law and fails when the inverse of CC^T does not exist, which happens when det(CC^T) = 0. This is the case if rank(CC^T) < 3, or equivalently, if rank(C) < 3. In this situation the system of CMGs is in a singular state which occurs when the individual CMG momentum vectors h_i are aligned to the commanded torque direction i.e., the individual CMG momentum vectors h_i have extremal projections onto to the commanded torque direction [16].

3 Spacecraft Model

This chapter will give an overview of the spacecraft model used. First, the rotational kinematics and dynamic equations of motion will be given in Sects. 3.1 and 3.2, respectively. The state estimation used for spacecraft attitude estimation is given in Sect. 3.3. Finally, the attitude controller used is discussed in Sect. 3.4.

3.1 Rotational Kinematics

The attitude of the spacecraft is represented using a quaternion which is defined as [4]

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \tag{7}$$

where $\mathbf{q} = [q_1\ q_2\ q_3]^T$ and

$$\mathbf{q} = \hat{\boldsymbol{e}} \sin(\phi/2) \qquad q_4 = \cos(\phi/2) \tag{8}$$

Eulers eigenaxis and its angle of rotation are denoted by $\hat{\boldsymbol{e}}$ and ϕ, respectively. The quaternion kinematic differential equation is given by Eq. 9.

$$\begin{aligned}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \\ &= \frac{1}{2} \boldsymbol{\Omega}(\omega)\, \bar{\mathbf{q}} \end{aligned} \tag{9}$$

3.2 Rigid-Body Dynamics

In Chap. 2, the gimbal accelerations were neglected in the calculation of the CMG angular momentum and torque. For the full equations of motion these terms will be included. The spacecraft itself is assumed to be rigid body with only CMGs as attitude control actuators. The total angular momentum of the spacecraft including CMGs $\boldsymbol{H} \in R^{3\times 1}$, about the center of mass expressed in the body-fixed reference frame is given by [11]

$$\begin{aligned}\boldsymbol{H} &= \boldsymbol{I}\boldsymbol{\omega} + \sum_{i=1}^{4} C_i^T \bar{h}_i \\ &= \boldsymbol{I}\boldsymbol{\omega} + \sum_{i=1}^{4} \left(C_i^T \boldsymbol{J} C_i \boldsymbol{\omega} + C_i^T \boldsymbol{J} \dot{\bar{\theta}}_i + C_i^T h_i \right)\end{aligned} \tag{10}$$

The angular momentum of each CMG is caused by the rotation of the spacecraft, the gimbaling of the CMG and the spinning of the rotor. The time derivative of Eq. 10 w.r.t. the inertial reference frame N equals the external torque $\boldsymbol{T}_{ext}$

$$\begin{aligned}\boldsymbol{T}_{ext} &= \frac{d}{dt}\left(\boldsymbol{H}\right)_N \\ &= \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \sum_{i=1}^{4} \frac{d}{dt}\left(C_i^T \bar{h}_i\right)_N\end{aligned} \tag{11}$$

where $\frac{d}{dt}\left(C_i^T \bar{h}_i\right)_N$ is the torque generated by the ith CMG. This torque can be written in the body reference frame as

$$\begin{aligned}\frac{d}{dt}\left(C_i^T \bar{h}_i\right)_N = {} & C_i^T \boldsymbol{J} C_i \dot{\boldsymbol{\omega}} + C_i^T \dot{\bar{\theta}}_i \times \boldsymbol{J} C_i \boldsymbol{\omega} \\ & + \omega \times C_i \boldsymbol{J} C_i \boldsymbol{\omega} + C_i^T \boldsymbol{J} \ddot{\bar{\theta}}_i + \boldsymbol{\omega} \times C_i^T \boldsymbol{J} \dot{\bar{\theta}}_i \\ & + C_i^T \dot{\bar{\theta}}_i \times h_i + \boldsymbol{\omega} \times C_i^T h_i\end{aligned} \tag{12}$$

Substituting Eq. 12 in Eq. 11 gives

$$\begin{aligned}\dot{\boldsymbol{\omega}} = - \boldsymbol{I}^{*-1} \Bigg[& \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{T}_{ext} + B\ddot{\boldsymbol{\theta}} \\ & + \left(D_1 + D_2 + D_3\right)\dot{\boldsymbol{\theta}} + \sum_{i=1}^{4} \boldsymbol{\omega} \times C_i^T \left(\boldsymbol{J} C_i \boldsymbol{\omega} + h_i\right) \Bigg]\end{aligned} \tag{13}$$

where

$$
\begin{aligned}
\boldsymbol{I^*} &\equiv \boldsymbol{I} + \sum_{i=1}^{4} C_i^T \boldsymbol{J} C_i \\
D_3\dot{\boldsymbol{\theta}} &\equiv \sum_{i=1}^{4} C_i^T \bar{\dot{\theta}}_i \times \boldsymbol{J} C_i \boldsymbol{\omega} \\
D_2\dot{\boldsymbol{\theta}} &\equiv \sum_{i=1}^{4} \boldsymbol{\omega} \times C_i^T \boldsymbol{J} \bar{\dot{\theta}}_i \\
D_1\dot{\boldsymbol{\theta}} &\equiv \sum_{i=1}^{4} C_i^T \bar{\dot{\theta}}_i \times h_i \\
B\ddot{\boldsymbol{\theta}} &\equiv \sum_{i=1}^{4} C_i^T \boldsymbol{J} \bar{\ddot{\theta}}_i
\end{aligned}
\tag{14}
$$

When the inertia of the CMGs are neglected and by defining the internal control torque $\boldsymbol{\tau} \in R^{3\times1}$, Eq. 13 reduces to

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{\tau} + \boldsymbol{T}_{ext} \tag{15a}$$

$$\dot{\boldsymbol{h}} + \boldsymbol{\omega} \times \boldsymbol{h} = -\boldsymbol{\tau} \tag{15b}$$

Most researches neglect the CMG inertia, because the acceleration torque of the CMG is usually much smaller than the precessional gyroscopic torque of the CMG. Furthermore, it has little influence on the singularity avoidance problem [7].

3.3 State Estimation

It is assumed that the spacecraft has three types of sensors: gyros, line-of-sight attitude sensor and gimbal angle measurement sensors. The gimbal angles are measured by a rotary sensor. Equation 5 is a linear equation so a linear Kalman filter is used to filter the noisy measurements.

The remainder of the section describes the Kalman filtering algorithm used for spacecraft attitude estimation, which consists of five parts [4, 8, 9, 14]. First, the gyro noise model will be given in Sect. 3.3.1 and the attitude sensor will be described in Sect. 3.3.2. The state equation will be given in Sect. 3.3.3. Finally, the prediction and filtering will be described in Sects. 3.3.4 and 3.3.5, respectively.

3.3.1 Gyro Noise Model

The angular rate of the spacecraft is measured using a rate-integrating gyro for which the following continuous time model is used

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{\beta} + \eta_v \tag{16a}$$

$$\frac{d}{dt}\boldsymbol{\beta} = \eta_u \tag{16b}$$

with noise statistics

$$E\left\{\eta_v(t)\,\eta_v^T(\tau)\right\} = \sigma_v^2\delta(t-\tau)\,I \tag{17a}$$

$$E\left\{\eta_u(t)\,\eta_u^T(\tau)\right\} = \sigma_u^2\delta(t-\tau)\,I \tag{17b}$$

It is assumed that η_v and η_u are independent zero mean Gaussian white noise processes. The continuous time noise covariance matrix is given by

$$\boldsymbol{Q}_c = \begin{bmatrix} \sigma_v^2 \cdot I & 0 \\ 0 & \sigma_u^2 \cdot I \end{bmatrix} = \begin{bmatrix} N_v & 0 \\ 0 & N_u \end{bmatrix} \tag{18}$$

During simulation, the following discrete time gyro model is used to generate the measurement data [4, 9]

$$\begin{aligned}\boldsymbol{\omega}_{m,k+1} = \boldsymbol{\omega}_{k+1} &+ \frac{1}{2}\left[\boldsymbol{\beta}_{k+1} + \boldsymbol{\beta}_k\right] \\ &+ \left[\frac{\sigma_v^2}{\Delta t} + \frac{1}{12}\sigma_u^2\Delta t\right]^{1/2} N_v\end{aligned} \tag{19a}$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \sigma_u\Delta t^{1/2}N_u \tag{19b}$$

3.3.2 Attitude Sensor

It is assumed that a line-of-sight attitude sensor is available which directly provides quaternion out measurements. These measurements are corrupted by white noise with statistical information known [9]. The measured quaternion $\bar{\boldsymbol{q}}_m$ is simulated using

$$\bar{\boldsymbol{q}}_m = \bar{\boldsymbol{q}}_n \otimes \bar{\boldsymbol{q}} \tag{20}$$

where $\bar{\boldsymbol{q}}_n$ is a noisy quaternion with

$$\begin{aligned} &E\left\{\boldsymbol{q}_n(t)\right\} = 0 \\ &E\left\{\boldsymbol{q}_n(t)\,\boldsymbol{q}_n^T\left(t'\right)\right\} = \boldsymbol{R}\end{aligned} \tag{21}$$

and $q_{n,4}$ such that the quaternion norm is satisfied.

3.3.3 State Equation

A seven element state vector is defined as

$$\boldsymbol{x} = \begin{bmatrix} \bar{\boldsymbol{q}} \\ \boldsymbol{\beta} \end{bmatrix} \tag{22}$$

The derivatives of the states are computed using

$$\dot{\bar{\boldsymbol{q}}}\left(t\right) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_m - \boldsymbol{\beta} - \eta_1\right)\bar{\boldsymbol{q}}\left(t\right) \tag{23a}$$

$$\dot{\boldsymbol{\beta}} = \eta_2 \tag{23b}$$

Taking the expectation gives

$$\dot{\hat{\bar{\boldsymbol{q}}}}\ \left(t\right) = \frac{1}{2}\boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}\right)\bar{\boldsymbol{q}}\left(t\right) \tag{24a}$$

$$\dot{\hat{\boldsymbol{\beta}}} = 0 \tag{24b}$$

with

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\boldsymbol{\beta}} \tag{25}$$

Due to the unity constraint on the quaternion, it is not possible to express the error quaternion as the arithmetic difference between the true and estimated quaternion. Therefore, a multiplicative extended Kalman filter is used where the error quaternion is defined as

$$\delta\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}} \otimes \hat{\bar{\boldsymbol{q}}}^{-1} \tag{26}$$

Since the error quaternion corresponds to a small rotation, the small angle approximation can be used

$$\delta\bar{\boldsymbol{q}} = \begin{bmatrix} \delta\boldsymbol{q} \\ \delta q_4 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{e}}\sin\left(\delta\phi/2\right) \\ \cos\left(\delta\phi/2\right) \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\phi} \\ 1 \end{bmatrix} \tag{27}$$

The bias error is defined as

$$\varDelta\beta = \beta - \hat{\beta} \tag{28}$$

and the six dimensional error vector is now given by

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \delta\boldsymbol{\phi} \\ \varDelta\beta \end{bmatrix} \tag{29}$$

3.3.4 Prediction

The derivative of error vector is calculated using

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{F}_c \cdot \tilde{\boldsymbol{x}} + \boldsymbol{G}_c \cdot \boldsymbol{\eta} \tag{30}$$

where

$$\boldsymbol{F}_c = \begin{bmatrix} -\left[\hat{\boldsymbol{\omega}}\times\right] & -I \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{G}_c = \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \tag{31}$$

and $\boldsymbol{\eta} = [\eta_1 \; \eta_2]^T$. By assuming that the continuous time system matrix $\boldsymbol{F}_c$ is constant during the integration interval, the state transition matrix Φ can be calculated using

$$\begin{aligned} \boldsymbol{\Phi}\left(t + \Delta t, t\right) &= \exp\left(\boldsymbol{F}_c \Delta t\right) \\ &= I + \boldsymbol{F}_c \Delta t + \frac{1}{2!}\boldsymbol{F}_c^2 \Delta t^2 + \ldots \end{aligned} \tag{32}$$

and the discrete time noise covariance matrix Q_d can be calculated using

$$\boldsymbol{Q}_d = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}\left(t_{k+1}, \tau\right) \boldsymbol{G}_c\left(\tau\right) \boldsymbol{Q}_c \boldsymbol{G}_c^T\left(\tau\right) \boldsymbol{\Phi}^T\left(t_{k+1}, \tau\right) \mathrm{d}\tau \tag{33}$$

The state covariance matrix is now calculated using

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{\Phi} \boldsymbol{P}_{k|k} \boldsymbol{\Phi}^T + \boldsymbol{Q}_d \tag{34}$$

3.3.5 Filtering

The Kalman gain can be calculated using

$$\boldsymbol{K} = \boldsymbol{P}\mathbf{H}^T\left[\mathbf{H}\boldsymbol{P}\mathbf{H}^T + \boldsymbol{R}\right]^{-1} \tag{35}$$

The error state vector correction is calculated using

$$\Delta\hat{\boldsymbol{x}}\left(+\right) = \begin{bmatrix} \delta\hat{\boldsymbol{\phi}}\left(+\right) \\ \Delta\hat{\beta}\left(+\right) \end{bmatrix} = \begin{bmatrix} 2 \cdot \delta\hat{\boldsymbol{q}}\left(+\right) \\ \Delta\hat{\beta}\left(+\right) \end{bmatrix} = \boldsymbol{K}\left(\boldsymbol{z} \otimes \hat{\boldsymbol{q}}^{-1}\right) \tag{36}$$

Using this correction, the states can be updated

$$\hat{\bar{\boldsymbol{q}}}\left(+\right) = \delta\hat{\bar{\boldsymbol{q}}}\left(+\right) \otimes \hat{\bar{\boldsymbol{q}}}\left(-\right) \tag{37a}$$

$$\hat{\beta}\left(+\right) = \hat{\beta}\left(-\right) + \Delta\hat{\beta}\left(+\right) \tag{37b}$$

Finally, the covariance matrix can be updated

$$\boldsymbol{P}_{k+1|k+1} = (I - \boldsymbol{K}\mathbf{H})\,\boldsymbol{P}_{k+1|k}\,(I - \boldsymbol{K}\mathbf{H})^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T \tag{38}$$

3.4 Attitude Control

The attitude error quaternions can be calculated using Eq. 39. This equation relates the current attitude quaternions to the demanded attitude quaternions. The quaternion error vector $\boldsymbol{e} \in R^{3\times 1}$ contains the first three attitude error quaternions, $\boldsymbol{e} = (e_1, e_2, e_3)$, and is the attitude error from target to the current attitude. This attitude error quaternion is used by the attitude controller and is calculated using

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix} \tag{39}$$

where q_{ic} is the commanded quaternion and $\hat{q}_i$ is the estimated attitude quaternion. As an attitude controller a PID-controller with slew rate limit is chosen. The control torque input $\boldsymbol{\tau}$ is calculated using Eq. 40 [19].

$$\boldsymbol{\tau} = -\boldsymbol{I}\left\{2k \underset{L_i}{\text{sat}}\left(\boldsymbol{e} + \frac{1}{T}\int \boldsymbol{e}\right) + c\boldsymbol{\omega}\right\} \tag{40}$$

where k is the proportional controller gain. The integral controller gain is given by k/T, where T is the time constant of the integral controller and c is the derivative controller gain. The controller gains can be set using

$$k = \omega_n^2 + 2\zeta\omega_n/T \tag{41a}$$

$$c = 2\zeta\omega_n + 1/T \tag{41b}$$

where ω_n, ζ and T have to be properly selected. In order to achieve the slew rate limit, a saturation limit is calculated using Eq. 42.

$$L_i = (c/2k)\min\left\{\sqrt{4a_i\,|e_i|}, |\omega_i|_{\max}\right\} \tag{42}$$

where $|\omega_i|_{\max}$ is the maximum angular rate about each axis and a_i is the maximum control acceleration about the ith axis.

4 Steering Law

The steering law is responsible for the control allocation of the CMGs. It uses the commanded torque calculated by the attitude controller to determine the commanded gimbal angles or rates of each CMG.

This chapter will first give the Inverse controller used in the simulations in Sect. 4.1. This is followed by an old and an improved Null Motion algorithm in Sect. 4.2.

4.1 Steering Law Allowing Torque Error

The steering law is derived by solving the following minimization problem [1]

$$\min_{\dot{\boldsymbol{\theta}}} P = \frac{1}{2}\boldsymbol{E}^T\boldsymbol{K}\boldsymbol{E} \tag{43}$$

where

$$\boldsymbol{E} = \begin{bmatrix} \dot{\boldsymbol{h}} - C\dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \tag{44}$$

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_1 & 0 \\ 0 & \boldsymbol{K}_2 \end{bmatrix} \tag{45}$$

and $\boldsymbol{K}_1 > 0$, $\boldsymbol{K}_2 > 0$ are weighting matrices. Substituting Eqs. 44 and 45 in Eq. 43 taking the partial derivative and equating it to zero gives

$$\frac{\partial P}{\partial \dot{\boldsymbol{\theta}}^T} = \boldsymbol{K}_2\dot{\boldsymbol{\theta}} - C^T\boldsymbol{K}_1\dot{\boldsymbol{h}} + C^T\boldsymbol{K}_1 C\dot{\boldsymbol{\theta}} = 0 \tag{46}$$

Solving this equation for $\dot{\boldsymbol{\theta}}$ and setting $\dot{\boldsymbol{h}}$ as the required torque gives

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= \left[C^T\boldsymbol{K}_1 C + \boldsymbol{K}_2\right]^{-1} C^T\boldsymbol{K}_1\dot{\boldsymbol{h}}_{req} \\ &= \boldsymbol{K}_2^{-1}C^T\left[C\boldsymbol{K}_2^{-1}C^T + \boldsymbol{K}_1^{-1}\right]^{-1}\dot{\boldsymbol{h}}_{req} \end{aligned} \tag{47}$$

When the weighting matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are chosen as the identity matrix the Pseudo Inverse steering law given by Eq. 6 is obtained. As discussed in Chap. 2, this method fails if rank$(C) < 3$ and the CMGs are in a singular state. By choosing $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ such that the inverse of $\left[C^T\boldsymbol{K}_1 C + \boldsymbol{K}_2\right]$ exists, it is sometimes possible to escape the singularity. Several methods have been suggested in literature [1, 5, 15, 18], of which two will be given below.

4.1.1 Singularity Robust

The Singularity Robust (SR) steering law is given by

$$\dot{\boldsymbol{\theta}} = C^T \left[CC^T + \lambda I \right]^{-1} \dot{\boldsymbol{h}}_{req} \tag{48}$$

where λ is a scalar which has to be properly selected [1]. This method adds a torque error based on the value of λ. This λ is a trade-off between the exact solution given by Eq. 47 and the bound on the gimbal rates. However, this method fails when the system is singular and a torque is requested along the singular direction. A different steering law which overcomes this problem is given in the next section.

4.1.2 Generalized Singularity Robust

The Generalized SR Inverse steering law given by

$$\dot{\boldsymbol{\theta}} = C^T \left[CC^T + \lambda \boldsymbol{E} \right]^{-1} \dot{\boldsymbol{h}}_{req} \tag{49}$$

where

$$\boldsymbol{E} = \begin{bmatrix} 1 & \varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & 1 & \varepsilon_1 \\ \varepsilon_2 & \varepsilon_1 & 1 \end{bmatrix} > 0 \tag{50}$$

and $\varepsilon_i = 0.01 \sin(0.5\pi t + \phi_i)$ with $\phi_1 = 0$, $\phi_2 = \pi/2$ and $\phi_3 = \pi$ [18]. This steering law always gives a non-zero gimbal rate command for a non-zero $\dot{\boldsymbol{h}}_{req}$. Therefore, any internal singularity can be escaped using the Generalized SR Inverse steering law.

Both methods use the scalar λ to add the torque error. There are several methods which can be used to select this value [1, 10, 18], this paper uses Algorithm 1

Algorithm 1 Calculation of scalar λ

```
if $m < m_{cr}$ then
    $\lambda = 0$
else
    if $\frac{\lambda_0}{m} < \lambda_{max}$ then
        $\lambda = \frac{\lambda_0}{m}$
    else
        $\lambda = \lambda_{max}$
    end if
end if
```

where m_{cr} is the critical value of m, λ_0 is a constant and λ_{max} is the maximum value of the weighting factor [1]. The method does not include any torque error when the system is away from a singularity and the maximum value is limited by λ_{max}. This means the SR inverse steering law does not avoid singularities but steers away from it when it approaches one.

4.2 Null Motion

Null motion is a motion of the CMGs which produces no net output torque on the spacecraft. This can be used to, in some cases, avoid/escape internal singularities. Internal singularities can be categorized in two groups depending if they can be escaped through null motion. Hyperbolic singularities are escapable by means of null motion, whereas elliptic singularities are not.

The null vector $\boldsymbol{n}$ lies in the null space of the Jacobian matrix C. It can be obtained using different methods. This report will use the nondirectional null motion algorithm given in Sect. 4.2.1. A new maximum null motion algorithm will be introduced in Sect. 4.2.2.

4.2.1 Nondirectional Null Motion

In order to avoid local optimum solutions of gradient based Null motion algorithms, a nondirectional Null Motion algorithm [1] was introduced as

$$\dot{\boldsymbol{\theta}} = C^{+}\dot{\boldsymbol{h}}_{req} + \rho\left(\boldsymbol{\theta}\right)\boldsymbol{n} \tag{51}$$

where C^{+} is the steering law and

$$\rho\left(\boldsymbol{\theta}\right) = \begin{cases} m^{6} & \text{if} \quad m \geq 1 \\ m^{-6} & \text{if} \quad m < 1 \end{cases} \tag{52}$$

and the singularity measure m is given by

$$m = \sqrt{det\left(CC^{T}\right)} \tag{53}$$

The null vector of the Jacobian matrix C is calculated using $\boldsymbol{n} = [C_1, C_2, C_3, C_4]$, where $C_i = (-1)^{i+1}\, M_i$ is the order 3 Jacobian cofactor. $M_i = det\left(J_i\right)$ is the order 3 Jacobian minor and $J_i = J$ with the ith column removed.

This nondirectional Null Motion algorithm always adds substantial null motion with the goal to avoid locally optimal configurations. However, this algorithm does not take into account the gimbal rate limits of the CMGs. When a CMG reaches its gimbal rate limit, a torque error is obtained because the achieved gimbal rates are no longer a null vector of the Jacobian matrix C.

4.2.2 Maximum Null Motion

To prevent Null motion from violating the gimbal rate limit, a new algorithm was developed which limits the amount of Null motion. This Maximum Null motion algorithm is shown in Algorithm 2.

Algorithm 2 Maximum Null Motion

```
$\dot{\boldsymbol{\theta}}_{nn} = C^{+}\dot{\boldsymbol{h}}_{req}$
$\boldsymbol{n} = [C_1, C_2, C_3, C_4]$
if $m < 1$ then
    $\rho = m^{-6}$
else
    $\rho = m^{6}$
end if
for i=1,4 do
    $\rho_{lim} = \left(\dot{\theta}_{max} - \dot{\boldsymbol{\theta}}_{nn}(i)\right) / \boldsymbol{n}(i)$
    if $\rho > \rho_{lim}$ then
        $\rho = \rho_{lim}$
    end if
end for
```

where $\dot{\boldsymbol{\theta}}_{nn}$ is the commanded gimbal rate with no Null Motion caused by the Inverse Steering law. The null vector is calculated the same way as with the nondirectional Null Motion and the commanded gimbal angles are calculated using Eq. 51. The innovation of this new method is that the amount of Null Motion is limited by the available gimbal rate.

The amount of Null Motion is based on the singularity measure given by Eq. 53. This value is always positive, which means that Null Motion is always applied towards one direction. Switching this direction, and using e.g. the direction which can achieve the most amount of Null Motion, can cause high frequency change in gimbal rates around the equilibrium point. Therefore, switching the direction of Null Motion is only applied when $\dot{\boldsymbol{\theta}}_c > \dot{\theta}_{max}$ and only when the negative direction produces a smaller gimbal rate command than the positive direction. This can only happen when the gimbal rate command by the steering law without Null Motion produces a higher gimbal rate command than the gimbal rate limit. An overview of this method is shown in Algorithm 3.

5 Performance Metrics

In order to evaluate the different steering laws and null motions algorithms, several performance metrics will be used. The first criterion which will be evaluated is whether the singularity is avoided or not. For this, the singularity measure given by Eq. 53 will be used. This value will become zero when the system is singular.

Algorithm 3 Null Motion Direction

```
$\dot{\boldsymbol{\theta}} = C^{+}\dot{\boldsymbol{h}}_{req} + \rho(\boldsymbol{\theta})\,\boldsymbol{n}$
$\dot{\theta}_{c,max} = max\left(abs\left(\dot{\boldsymbol{\theta}}\right)\right)$
$\rho_{new} = -\rho$
for j=1,4 do
    if abs$\left(\dot{\boldsymbol{\theta}}(j) > \dot{\theta}_{max}\right)$ then
        for i=1,4 do
            $\rho_{lim} = \left(-\dot{\theta}_{max} - \boldsymbol{n}(i)\right)/\boldsymbol{n}(i)$
            if $\rho_{new} < \rho_{lim}$ then
                $\rho_{new} = -\rho_{lim}$
            end if
        end for
        break
    end if
end for
if $max\left(abs\left(C^{+}\dot{\boldsymbol{h}}_{req} + \rho_{new}\boldsymbol{n}\right)\right) < \dot{\theta}_{c,max}$ then
    $\rho = \rho_{new}$
end if
```

Secondly, the pointing error will be used as a performance metric. This value is calculated as the cumulative absolute error. For a constant torque command, this will be the sum of the absolute error between generated and commanded torque in the three principle directions. For the pointing task, this will be the sum of the first three elements of the absolute difference between the true and commanded quaternion.

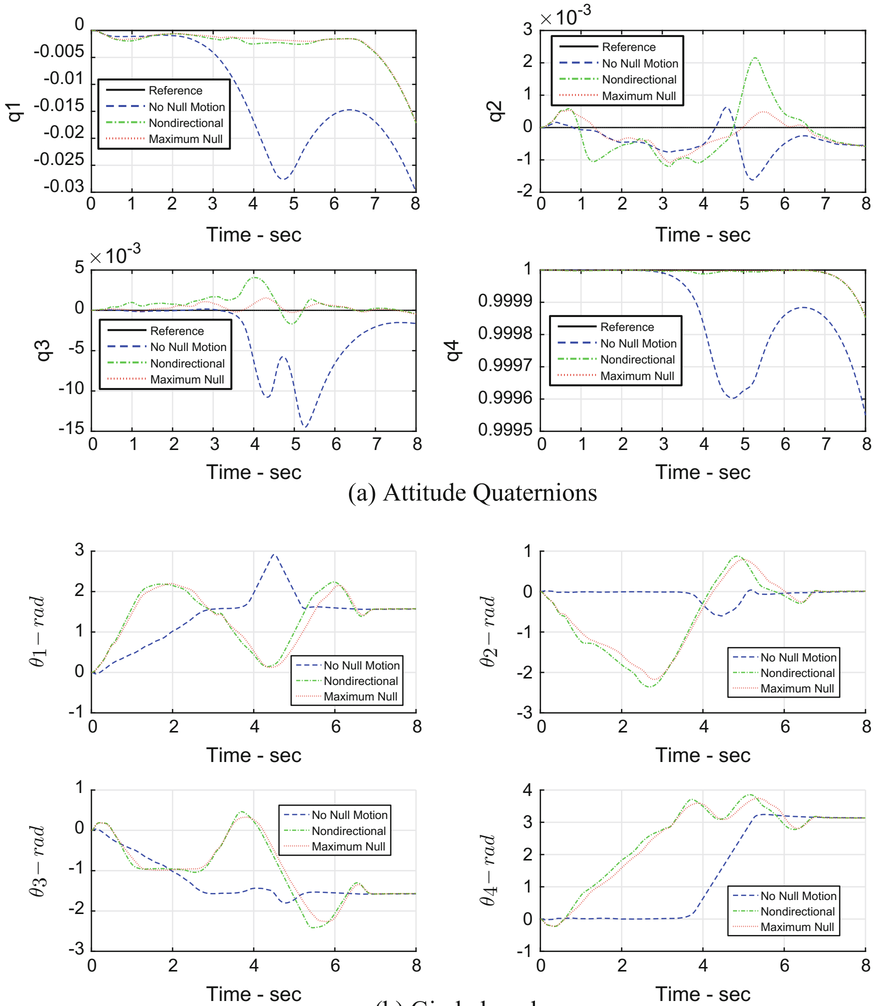
Finally, the cumulative absolute gimbal- angles and rates are added for all gimbals, respectively. This gives a measure of how much the gimbals are moved and how fast they are accelerated. Since a major problem with CMGs is their reliability, it is assumed that large CMG movements and accelerations are bad and that the steering algorithms which produce low scores on these metrics are preferred.

6 Simulation and Discussion

The different steering laws discussed in Sect. 4 are evaluated using the performance metrics discussed in Sect. 5. The simulations uses the four CMG pyramid configuration with $\alpha = 53.13$ deg. The gimbal dynamics are included with $\omega_n = 50$ rad/s and $\zeta = 0.7$ and a gimbal rate limit of $\dot{\theta}_{max} = 2$ rad/s is applied.

6.1 Station Keeping

The majority of time a spacecraft will be in station keeping mode during which a given attitude has to be maintained. Figures 2 and 3 show a station keeping task during



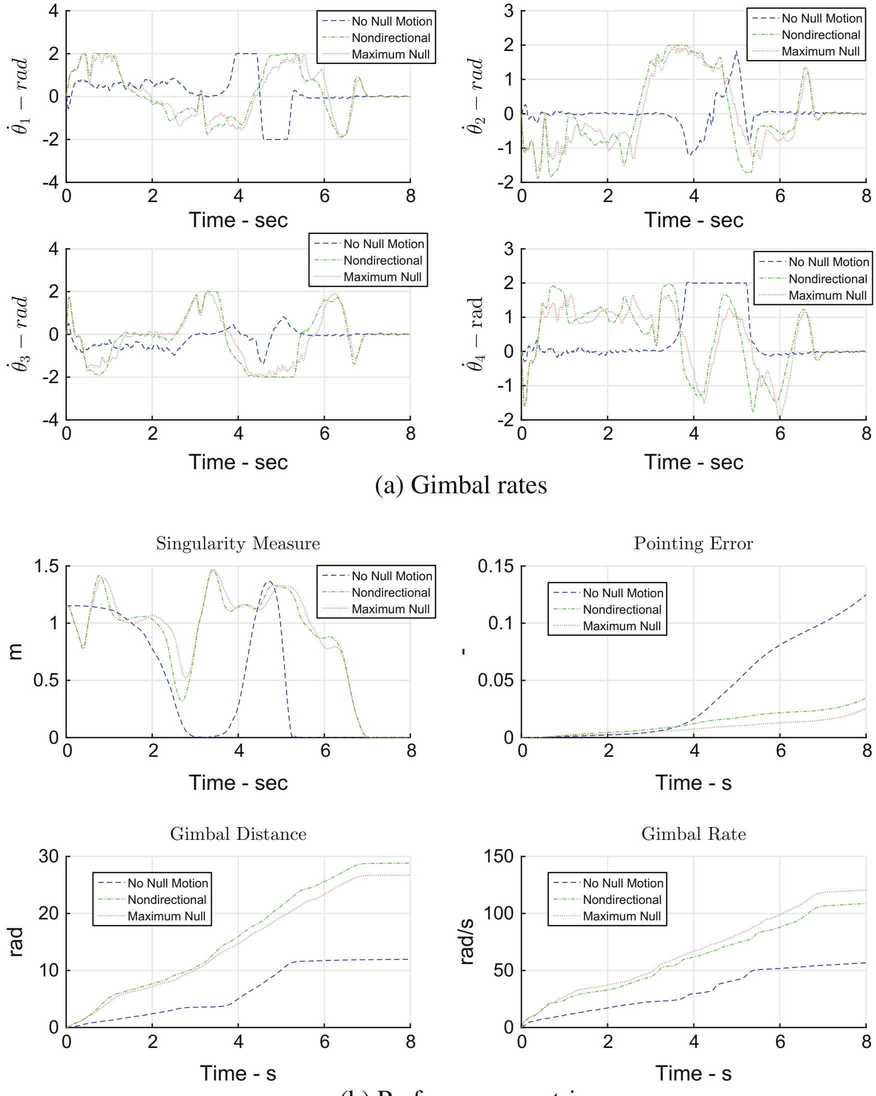
(a) Attitude Quaternions

(b) Gimbal angles

Fig. 2 Quaternions and gimbal angles of the Generalized SR steering law with different null motion algorithms for a station keeping task of $\bar{\boldsymbol{q}} = [0\ 0\ 0\ 1]$ under a constant disturbance torque

which the spacecraft has to maintain an attitude of $\bar{\boldsymbol{q}} = [0\ 0\ 0\ 1]$ under a constant disturbance torque of $\boldsymbol{T}_{ext} = [-0.5\ 0\ 0]^T$. The torque command is generated using the attitude controller given in Sect. 3.3.4.

Figure 2a shows the attitude quaternions during the station keeping task. This figure shows that when no null motion is included the Generalized SR steering law introduces a torque error to pass the internal singularity at three seconds. The two null motion algorithms pass this singularity with less torque error. This results in a lower pointing error as shown in Fig. 3b. The maximum null motion algorithm performs



(a) Gimbal rates

(b) Performance metrics

Fig. 3 Gimbal rates and performance metrics of the Generalized SR steering law with different null motion algorithms for a station keeping task of $\bar{\boldsymbol{q}} = [0\ 0\ 0\ 1]$ under a constant disturbance torque

better than the nondirectional one. After approximately 7.5 s the pointing error will keep increasing with a constant rate, because the system of CMGs is saturated and cannot generate torque to counter the disturbance.

The reduced pointing error of the maximum null motion algorithm does come at the cost of an higher change in gimbal rates, as shown in Fig. 3. There are higher fluctuations in the gimbal rates for the maximum null motion algorithm which results in a higher gimbal rate performance metric.

7 Conclusions and Recommendations

A spacecraft containing SGCMGs was simulated using the complete equations of motion, gimbal dynamics, gimbal rate limit and including system and sensor noise. A new maximum null motion algorithm was developed which performs better than the nondirectional algorithm for a constant torque request using both the SR and Generalized SR Inverse steering law. This is especially relevant for station keeping tasks of the spacecraft under the influence of a constant disturbance torque.

The maximum null algorithm has better performance than the traditional method, but this comes at the cost of higher changes in gimbal angles and rates. Therefore, in practical applications a trade-off is required between the pointing error and the work done by the CMGs. This paper showed that, even including system and sensor noise, the inherent singularity problem of single gimbal control moment gyroscopes remains a serious issue.

This paper described how CMGs can be used for a station keeping task. However, some spacecraft also require precision tracking capabilities. In some cases the required trajectory is known in advanced, so nonlinear optimization algorithms can be used to determine the optimal path in advance. However, in other cases the required trajectory might not be known for which new methods should be developed.

References

1. Bedrossian NS, Paradiso J, Stark C, Rowell D (1990) Steering law design for redundant single-gimbal control moment gyroscopes. J Guid Control Dyn 13(6):1083–1089
2. Chubb W, Seltzer S (1971) Skylab attitude and pointing control system. NASA TN D-6068 (February). http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19710007043.pdf
3. Clark C, Worrall K, Yavuzoglu E (2010) A control moment gyro for dynamic attitude control of small satellites conference on small satellites. In: AIAA/USU conference on small satellites. http://digitalcommons.usu.edu/smallsat/2010/all2010/68/
4. Crassidis JL, Markley FL, Cheng Y (2007) Survey of nonlinear attitude estimation methods. J Guid Control Dyn 30(1):12–28. https://doi.org/10.2514/1.22452
5. Ford KA, Hall CD (2000) Singular direction avoidance steering for control-moment gyros. J Guid Control Dyn 23(4):648–656. https://doi.org/10.2514/2.4610
6. Kurokawa H (1998) A geometric study of single gimbal control moment gyros. Technical report 175, Report of Mechanical Engineering
7. Kurokawa H (2007) Survey of theory and steering laws of single-gimbal control moment gyros. J Guid Control Dyn 30(5):1331–1340. https://doi.org/10.2514/1.27316
8. Lefferts EJ, Markley EL, Shuster MD (1982) A I AA-8 2-0070 Estimation. In: AIAA 20th aerospace sciences meeting

9. Markley FL, Crassidis JL (2014) Fundamentals of spacecraft attitude determination and control. Springer, Berlin
10. Nakamura Y, Hanafusa H (1986) Inverse kinematic solutions with singularity robustness for robot manipulator control. J Dyn Syst Meas Control 108(3):163–171. http://dynamicsystems.asmedigitalcollection.asme.org/article.aspx?articleid=1403812
11. Oh H, Vadali S (1989) Feedback control and steering laws for spacecraft using single gimbal control moment gyros
12. Roser X, Sghedoni M (1997) Control Moment Gyroscopes (CMG'S) and their application in future scientific missions. In: Proceedings third international conference on spacecraft guidance, navigation and control systems, pp 523–528
13. Ross CH, Worley E (1971) Optimized Momentum and Attitude Control System (MACS) for skylab. In: AIAA guidance, control and flight mechanics conference, vol 71-938
14. Trawny N, Roumeliotis SI (2005) Indirect Kalman filter for 3D attitude estimation, Department of Computer Science & Engineering (2005-002), pp 1–25. https://doi.org/10.2514/6.2005-6052. http://www-users.cs.umn.edu/~trawny/Publications/Quaternions_3D.pdf
15. Wie B (2005) Singularity escape/avoidance steering logic for control moment gyro systems. J Guid Control Dyn 28(5): 948–956. https://doi.org/10.2514/1.10136
16. Wie B (2008) Space vehicle dynamics and control. In: AIAA educational series
17. Wie B, Byun KW, Warren VW (1988) A new momentum management controller for the space station. Technical report. https://doi.org/10.2514/6.1988-4132
18. Wie B, Bailey D, Heiberg C (2001) Singularity robust steering logic for redundant single-gimbal control moment gyros. J Guid Control Dyn 24(5):865–872. https://doi.org/10.2514/2.4799
19. Wie B, Bailey D, Heiberg C (2002) Rapid multitarget acquisition and pointing control of agile spacecraft. J Guid Control Dyn 25(1):96–104. https://doi.org/10.2514/2.4854

Immersion and Invariance Adaptive Backstepping Spacecraft Attitude Control with Modified Rodrigues Parameters

Guilherme F. Trigo and Qi-Ping Chu

1 Introduction

The ever-wider span of satellite applications brought a generalized demand for smaller and cheaper spacecraft. Smaller scale, however, carries higher vulnerability to disturbances and lower power capacity, reinforcing the need for increasingly robust and efficient Attitude Control Systems (ACSs). In 2007, Delft University of Technology and Beijing's Tsinghua University joined efforts and started the Formation for Atmospheric Science and Technology demonstration (FAST) [5, 6, 17]. The mission consists of the formation flying of two micro-satellites that will collect high atmospheric data. Due to its academic nature, and adding to the limitations and vulnerabilities of small size/weight, the ACS of FAST-D, the Dutch element of the swarm, faces several challenges such as high pointing accuracy and stability requirements, limited precision of moment of inertia information, and possible moving parts among carried instruments.

The most used solution for such attitude control problem involves the application of linear robust laws in a scheduled manner, resulting in a limited operation envelope in which the system has conservative and uneven performance [13]. Alternatively, nonlinear control laws provide (in most cases) full-envelope control with consistent performance. However, for relying deeply on the correct modelling of the system, the most used nonlinear laws are particularly vulnerable to disturbances or unmodelled dynamics. Adaptive schemes are a common remedy in this case. Due to their strong stability basis, Lyapunov-based control laws (e.g. Backstepping [14]) lend themselves particularly well to adaptive feedback feature.

G.F. Trigo (✉)
GNC Systems Department, German Aerospace Center (DLR),
Institute of Space Systems, Bremen, Germany
e-mail: guilherme.trigo@dlr.de

Q.-P. Chu
Faculty of Aerospace Engineering, Control and Simulation Department,
Delft University of Technology, Delft, The Netherlands

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_38

Adaptive Backstepping by Tuning Functions is a well-known technique in which control and adaptation are derived together following Lyapunov stability conditions [9]. Despite their stability properties, these laws are hard to tune as faster adaptation does not necessarily yield better transient performance. Li et al. [16] applied this method to control the attitude of a spacecraft with uncertain inertia.

A different approach combines, in a modular adaptive scheme, the Backstepping control law with the estimates given by a non-Lyapunov identifier such as the recursive least-squares identifier [25] or the high-gain observer [2]. Since the certainty equivalence principle does not usually hold for nonlinear systems, this configuration requires the addition of nonlinear damping terms for robustness against estimation error growth [15]. This emend, however, can yield undesirably high gain control.

A more recently developed Lyapunov-based adaptive control method employs the Immersion and Invariance (I&I) theory [1, 10–12]. In this framework the estimation error can be assigned a stable dynamics, making it easier to tune than the Tuning Function in the traditional adaptive Backstepping. The estimation law obtained with this method can be combined with a Backstepping controller as done by Sonneveldt et al. [22] to adaptively control a fighter aircraft model.

In this contribution we show that the I&I adaptive Backstepping approach is *input-to-state* stable for a class of strict-feedback systems with time-varying parametric uncertainties. This new stability result allows for a rather general description of the spacecraft uncertainties in the design, as a sum of potentially time-variant disturbance torques. The proposed I&I adaptive attitude controller for this model is derived using command filters which not only simplifies the concept but also provides means of limiting the control signals [3, 28]. Use of a three-element parametrization as the Modified Rodrigues Parameters further simplifies the design in comparison to a quaternion-based one.

We start by introducing the spacecraft model used for the control design. Then, the concept of Immersion and Invariance and its application to adaptive Backstepping are briefly explained. The stability proof of such a control law for a class of systems with varying uncertainties is then shown. Coming to the attitude control design, first a baseline Standard Backstepping attitude controller is constructed, which then is used to draw the I&I adaptive Backstepping attitude control law. A Tuning Functions adaptive Backstepping law is also derived. Finally, comparative testing of the three controllers is carried out for nominal and disturbed scenarios.

2 Spacecraft Model

2.1 Modified Rodrigues Parameters

The Modified Rodrigues Parameters (MRP) are a three-element attitude representation obtained by stereographical projection of the sphere of quaternion parameters [20, 21]. In its positive form, with $(0,\ 0,\ 0,\ -1)$ as point of infinity, the MRP vector $\boldsymbol{\sigma}$ is given by

$$\boldsymbol{\sigma} = \frac{\mathbf{q}_{1:3}}{1 + \mathrm{q}_4} \ , \tag{1}$$

where $\mathbf{q}_{1:3}$ is the quaternion vector part and q_4 is the scalar part. This transformation makes the MRP representation the most near-linearly behaving three-component attitude parametrization [8]. The Direction Cosine Matrix (DCM) corresponding to the rotation represented by $\boldsymbol{\sigma}$ is given by

$$\mathbf{C}(\boldsymbol{\sigma}) = \mathbf{I} - 4 \, \frac{1 - |\boldsymbol{\sigma}|^2}{\left(1 + |\boldsymbol{\sigma}|^2\right)^2} \, [\boldsymbol{\sigma}\times] + \frac{8}{\left(1 + |\boldsymbol{\sigma}|^2\right)^2} \, [\boldsymbol{\sigma}\times]^2 \ , \tag{2}$$

where $[\boldsymbol{\sigma}\times]$ is a skew-symmetric matrix of $\boldsymbol{\sigma}$ and $\mathbf{I}$ is a 3×3 identity matrix. The MRP kinematics is written

$$\dot{\boldsymbol{\sigma}} = \mathbf{B}(\boldsymbol{\sigma})\boldsymbol{\omega} \ , \tag{3}$$

with

$$\mathbf{B}(\boldsymbol{\sigma}) = \frac{1}{2} \, [\boldsymbol{\sigma}\times] + \frac{1}{4} \left(1 - |\boldsymbol{\sigma}|^2\right) \mathbf{I} + \frac{1}{2} \boldsymbol{\sigma}\boldsymbol{\sigma}^{\mathsf{T}} \ , \tag{4}$$

where $\boldsymbol{\omega}$ is the angular velocity between the reference frames selected for the definition of the attitude.

The following identities will be useful when defining the control laws later on

$$\mathbf{B}(\boldsymbol{\sigma})\mathbf{C}(\boldsymbol{\sigma}) = \mathbf{B}(-\boldsymbol{\sigma}) \tag{5}$$

$$\mathbf{C}(\boldsymbol{\sigma})\boldsymbol{\sigma} = \boldsymbol{\sigma} \ . \tag{6}$$

Local Vertical Frame and Attitude Kinematics

Given the Earth observation mission of the spacecraft under study, it is convenient to consider its attitude with respect to an orbital nadir-pointing frame. The Local-Vertical Local-Horizontal (LVLH) or Orbital frame, denoted O, is centered on the spacecraft CoM, has the z-axis pointing to the center of the Earth, the y-axis pointing in the negative direction of the orbital angular momentum, and the x-axis completing the right-handed frame.

The attitude of the spacecraft Body-fixed reference frame B with respect to this Orbital frame, given by $\boldsymbol{\sigma}_O^B$, follows a kinematics law of the form of (3), i.e.

$$\dot{\boldsymbol{\sigma}}_O^B = \mathbf{B}(\boldsymbol{\sigma}_O^B)\boldsymbol{\omega}_{BO}^B \ , \tag{7}$$

where the body angular velocity with respect to the Orbital frame can be split as

$$\boldsymbol{\omega}_{BO}^B = \boldsymbol{\omega}_{BI}^B - \mathbf{C}_O^B \mathbf{n}_{OI}^O \ . \tag{8}$$

$\boldsymbol{\omega}_{BI}^B$ is the absolute angular velocity of the spacecraft written in Body coordinates and $\mathbf{n}_{OI}^O$ the angular velocity vector of the Orbital frame. The latter is given in terms of the orbital rate value, n_{o}, as $\mathbf{n}_{OI}^{O} = \left[0\ -n_{\mathrm{o}}\ 0\right]^{\mathsf{T}}$. The direction cosine matrix $\mathbf{C}_{O}^{B}$ from Orbital to Body frame is computed from $\boldsymbol{\sigma}_{O}^{B}$ using Expression (2) as

$$\mathbf{C}_{O}^{B} \triangleq \mathbf{C}(\boldsymbol{\sigma}_{O}^{B}) . \tag{9}$$

Attitude Error Kinematics

The attitude error parameter $\boldsymbol{\delta\sigma}$ is here defined as representing the rotation from the desired attitude $\boldsymbol{\sigma}_{O,\mathrm{r}}^{B}$ to the actual one $\boldsymbol{\sigma}_{O}^{B}$, i.e.

$$\boldsymbol{\sigma}_{O}^{B} = \boldsymbol{\delta\sigma} \otimes \boldsymbol{\sigma}_{O,\mathrm{r}}^{B} , \tag{10}$$

where $\otimes$ denotes MRP composition.[1]

Inverting (10) yields

$$\boldsymbol{\delta\sigma} = \boldsymbol{\sigma}_{O}^{B} \otimes \left(-\boldsymbol{\sigma}_{O,\mathrm{r}}^{B}\right) = \frac{(1-|\boldsymbol{\sigma}_{\mathrm{r}}|^2)\boldsymbol{\sigma} - \left(1-|\boldsymbol{\sigma}|^2\right)\boldsymbol{\sigma}_{\mathrm{r}} + 2\boldsymbol{\sigma}\times\boldsymbol{\sigma}_{\mathrm{r}}}{1+|\boldsymbol{\sigma}|^2|\boldsymbol{\sigma}_{\mathrm{r}}|^2 + 2\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{r}}} , \tag{11}$$

where the reference frame super/subscripts were omitted for simplicity.

The MRP error kinematics follows

$$\boldsymbol{\delta\dot{\sigma}} = \mathbf{B}\left(\boldsymbol{\delta\sigma}\right)\boldsymbol{\delta\omega} , \tag{12}$$

with angular velocity error given as

$$\boldsymbol{\delta\omega} = \boldsymbol{\omega}_{BI}^{B} - \mathbf{C}\left(\boldsymbol{\sigma}_{O}^{B}\right)\mathbf{n}_{OI}^{O} - \mathbf{C}(\boldsymbol{\delta\sigma})\boldsymbol{\omega}_{BO,\mathrm{r}}^{B} , \tag{13}$$

where $\boldsymbol{\omega}_{BO,\mathrm{r}}^{B}$ is the reference angular velocity.

As it will become evident later on, the error kinematics given in the standard kinematics form by (12) will ease the controller derivation when compared to a partial derivative differential equation form.

2.2 Attitude Dynamics

Considering the spacecraft a rigid-body, the body attitude dynamics with respect to an inertial frame I is given by [26]

$$\mathbf{J}^{B}\dot{\boldsymbol{\omega}}_{BI}^{B} = -\boldsymbol{\Omega}_{BI}^{B}\mathbf{h}_{BI}^{B} + \mathbf{m}^{B} , \tag{14}$$

[1] The rotation represented by $\boldsymbol{\sigma}_{\mathrm{a}} \otimes \boldsymbol{\sigma}_{\mathrm{b}}$ is equivalent to that given by left DCM multiplication $\mathbf{C}\left(\boldsymbol{\sigma}_{\mathrm{a}}\right)\mathbf{C}\left(\boldsymbol{\sigma}_{\mathrm{b}}\right)$.

where $\boldsymbol{\Omega}_{BI}^{B}$ is the skew symmetric matrix of $\boldsymbol{\omega}_{BI}^{B}$, $\mathbf{h}_{BI}^{B} = \mathbf{J}^{B}\boldsymbol{\omega}_{BI}^{B}$ is the total angular momentum, $\mathbf{J}^{B}$ is the inertia tensor and $\mathbf{m}^{B}$ is the sum of all external torques applied to the spacecraft. This latter term is considered to include a gravity gradient induced torque $\mathbf{m}_{\mathrm{g}}^{B}$, a control input torque $\mathbf{m}_{\mathrm{c}}^{B}$ and an unknown (potentially time-variant) disturbance torque $\mathbf{m}_{\mathrm{d}}^{B}$,

$$\mathbf{m}^{B} = \mathbf{m}_{\mathrm{g}}^{B} + \mathbf{m}_{\mathrm{c}}^{B} + \mathbf{m}_{\mathrm{d}}^{B}\,. \tag{15}$$

The gravity gradient induced torque is modelled as [27]

$$\mathbf{m}_{\mathrm{g}}^{B} = 3n_{\mathrm{o}}^{2}\left[\mathbf{c}_3(\boldsymbol{\sigma}_{O}^{B})\times\right]\mathbf{J}^{B}\,\mathbf{c}_3(\boldsymbol{\sigma}_{O}^{B})\,, \tag{16}$$

where $\mathbf{c}_3(\boldsymbol{\sigma}_{O}^{B})$ is the third column of the direction cosine matrix (9).

3 Immersion and Invariance Adaptive Control

This section presents the Immersion and Invariance (I&I) framework, first proposed by Astolfi et al. [1], and its combination with Backstepping control methodology. After a brief introduction to the concept of I&I, the general design logic of an I&I-based adaptive Backstepping controller is explained. For a more detailed version of the text presented in this section please refer to [23].

The name of the I&I framework comes from the fact that the design method uses an *immersion* of the system into the desired dynamics together with the definition of an *invariant* manifold. It can be used to adaptively stabilize systems with uncertain parameters. To illustrate how, consider the scalar system

$$\dot{x} = f(x) + g(x)u\,, \tag{17}$$

where $x \in \mathbb{R}$, $u \in \mathbb{R}$ and the function $f(x)$ depends linearly on the *unknown* parameter $\theta \in \mathbb{R}$ as

$$f(x) = f_0(x) + f_\theta(x)\theta\,, \tag{18}$$

for some *known* functions $f_0(x) : \mathbb{R} \to \mathbb{R}$ and $f_\theta(x) : \mathbb{R} \to \mathbb{R}$. Assume, in a *certainty equivalence* manner, that there exists a *full-information* control law $u = v(x, \theta)$ such that the closed-loop system

$$\dot{x} = f_*(x) = f(x) + g(x)v(x, \theta) \tag{19}$$

has a globally asymptotically stable equilibrium at $x = x_*$.

The problem is solved by finding $\beta(x)$ and $w(x, \hat{\theta})$ such that all trajectories of the extended system

$$\begin{cases} \dot{x} = f(x) + g(x)v(x, \hat{\theta} + \beta(x)) \\ \dot{\hat{\theta}} = w(x, \hat{\theta}) \end{cases} \tag{20}$$

are bounded and satisfy

$$\lim_{t \to \infty} \Big[g(x)v(x, \hat{\theta} + \beta(x)) - g(x)v(x, \theta) \Big] = 0\,. \tag{21}$$

If such task is possible, system (17) is said to be *Adaptively* I&I *Stabilizable*.

Condition (21) is satisfied for all trajectories converging to the manifold

$$\mathcal{M} = \Big\{ (x, \hat{\theta}) \in \mathbb{R}^2 \mid \hat{\theta} - \theta + \beta(x) = 0 \Big\}\ , \tag{22}$$

i.e. if $\mathcal{M}$ is attractive.

Note that, despite departing from a *certainty equivalence* philosophy, the controller developed by I&I method does not follow the *certainty equivalence* principle, in the sense that the parameter estimate is not applied directly to the *full-information* feedback law. Instead, the true parameter θ is replaced by $\hat{\theta} + \beta(x)$, where the function $\beta(x)$ adds a "proportional" action to the "integral" action of the estimate update law. For so, this kind of adaptive scheme is also known as *nonlinear PI adaptation*.

3.1 I&I-Based Adaptive Backstepping Control

The Immersion and Invariance control design principles introduced can be synthesized for the adaptive controller design for nth order parametric strict-feedback systems. Consider a system of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 + \varphi_1(x_1)\theta_1 \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2)x_3 + \varphi_2(x_2)\theta_2 \\ &\ \vdots \\ \dot{x}_n &= f_n(x_1, x_2, \ldots, x_n) + g_n(x_1, x_2, \ldots, x_n)u + \varphi_n(x_n)\theta_n \end{aligned} \tag{23}$$

where $u \in \mathbb{R}$, the functions $g_i(x_1, \ldots, x_i) \neq 0\ \forall\ (x_1, \ldots, x_n) \in \mathbb{R}^n$, $\varphi_i(x_i) : \mathbb{R} \to \mathbb{R}$ are known smooth nonlinear functions and $\theta_i \in \mathbb{R}$ are constant unknown parameters. The goal is to make x_1 asymptotically track a sufficiently smooth reference y_r.

Estimator Design

The first step is the design of an observer for the constant uncertainties θ_i. To this end, define the *off-the-manifold* (error) coordinates

$$\eta_i = \hat{\theta}_i - \theta_i + \beta_i(x_i)\ , \quad i \in \{1, 2, \ldots, n\} \tag{24}$$

where the estimates $\hat{\theta}_i$ are the observer states and $\beta_i(x_i) : \mathbb{R} \to \mathbb{R}$ are smooth functions that will be defined later on. The error dynamics is given by

$$\begin{aligned}\dot{\eta}_i &= \dot{\hat{\theta}}_i + \frac{\partial \beta_i}{\partial x_i}\,(f_i + g_i x_{i+1} + \varphi_i \theta_i) \\ &= \dot{\hat{\theta}}_i + \frac{\partial \beta_i}{\partial x_i}\left(f_i + g_i x_{i+1} + \varphi_i\left(\hat{\theta}_i + \beta_i - \eta_i\right)\right) \end{aligned} \tag{25}$$

where $x_{n+1} \triangleq u$. The update law $\dot{\hat{\theta}}_i$ can be define in such a way that it cancels the known parts of (25). This can be done by writing

$$\dot{\hat{\theta}}_i = -\frac{\partial \beta_i}{\partial x_i}\left(f_i + g_i x_{i+1} + \varphi_i\left(\hat{\theta}_i + \beta_i\right)\right) \tag{26}$$

which yields

$$\dot{\eta}_i = -\frac{\partial \beta_i}{\partial x_i}\varphi_i \eta_i \tag{27}$$

The error system (27) for $i = 1, 2, \ldots, n$ can be seen as linear time-varying, requiring for stability that the diagonal blocks in the dynamic matrix are rendered negative semi-definite. A choice of β_i that results in such is

$$\beta_i = \gamma_i \int_0^{x_i} \varphi_i(\chi) d\chi\,, \quad \gamma_i > 0 \tag{28}$$

Lemma 1 *The error system* (27)*, with functions β_i given as in* (28) *has a uniformly globally stable equilibrium at the origin. Furthermore, $\eta_i \in \mathcal{L}_\infty$ and $\varphi_i \eta_i \in \mathcal{L}_2$, $\forall i \in \{1, 2, \ldots, n\}$ and $\forall x_i(t)$. Additionally, if the regressor functions φ_i and their time-derivatives are bounded, then $\varphi_i \eta_i \to 0$.*

Proof Consider the Lyapunov function $W = \sum_{i=1}^{n} \eta_i^2$. Its time-derivative along (27) satisfies

$$\begin{aligned}\dot{W} &= -2 \sum_{i=1}^{n} \frac{\partial}{\partial x_i}\left(\gamma_i \int_0^{x_i} \varphi_i(\chi) d\chi\right)\varphi_i \eta_i^2 \\ &= -2 \sum_{i=1}^{n} \gamma_i (\varphi_i \eta_i)^2 < 0\,, \qquad \forall\, \varphi_i \eta_i \neq 0\end{aligned}$$

□

Noted that, by Lemma 1 and definition of (24), an estimate is obtained for each $\varphi_i \theta_i$ of the system (23) instead of θ_i. These estimates are given by $\varphi_i(\hat{\theta}_i + \beta_i)$.

Control Law Design

Done the general design of the estimator the backstepping control law for I&I design can now be defined. The design here presented includes command filters. This feature, not only reduces the derivation burden by avoiding the analytic computation of the time-derivatives of the virtual input signals, but also allows constraining of the magnitude and rate of the filtered signals [4].

The controller design starts by defining the tracking error coordinates

$$z_i = x_i - x_{i,\mathrm{c}} \,, \qquad i \in \{1, 2, \ldots, n\} \,, \tag{29}$$

with $x_{i,\mathrm{c}} \triangleq y_\mathrm{r}$. The signals $x_{i,\mathrm{c}}$ are commanded states that are still to be defined. The inclusion of command filters requires the use of compensated error coordinates

$$\bar{z}_i = z_i - \chi_i \,, \tag{30}$$

where χ_i accounts for the effect of the command filters and will be defined further in the design. The dynamics of the compensated errors is given by

$$\dot{\bar{z}}_i = \dot{z}_i - \dot{\chi}_i = f_i + g_i x_{i+1} + \varphi_i \theta_i - \dot{x}_{i,\mathrm{c}} - \dot{\chi}_i \,, \tag{31}$$

where $x_{n+1} \triangleq u$. Let the raw control signals be defined as

$$\begin{aligned} x^0_{i+1,\mathrm{c}} &= g_i^{-1} \left(-f_i - k_i z_i - g_{i-1} \bar{z}_{i-1} - \varphi_i (\hat{\theta}_i + \beta_i) + \dot{x}_{i,\mathrm{c}} \right) - \chi_{i+1} \,, \quad i \in \{1, 2, \ldots, n-1\} \\ u^0 &= g_n^{-1} \left(-f_n - k_n z_n - g_{n-1} \bar{z}_{n-1} - \varphi_n (\hat{\theta}_n + \beta_n) + \dot{x}_{n,\mathrm{c}} \right) \,, \end{aligned} \tag{32}$$

with $k_i > 0, k_n > 0, \dot{x}_{1,\mathrm{c}} \triangleq \dot{y}_\mathrm{r}, \bar{z}_0 = 0$ and $g_0 = 0$. These signals are low-pass filtered to produce $x_{i+1,\mathrm{c}}$ and u and their derivatives $\dot{x}_{i+1,\mathrm{c}}$ and $\dot{u}$ [3]. The effect of the filter on the error coordinates is estimated by

$$\begin{aligned} \dot{\chi}_i &= -k_i \chi_i + g_i (x_{i+1,\mathrm{c}} - x^0_{i+1,\mathrm{c}}) \,, \quad i \in \{1, 2, \ldots, n-1\} \\ \dot{\chi}_n &= -k_n \chi_n + g_n (u - u^0) \,. \end{aligned} \tag{33}$$

Applying (32) and (33) to (31) yields

$$\begin{aligned} \dot{\bar{z}}_i &= f_i + g_i z_{i+1} + g_i \left(x_{i+1,\mathrm{c}} - x^0_{i+1,\mathrm{c}} \right) + g_i x^0_{i+1,\mathrm{c}} + \varphi_i \theta_i - \dot{x}_{i,\mathrm{c}} - \dot{\chi}_i \\ &= -k_i \bar{z}_i - \varphi_i \eta_i + g_i \bar{z}_{i+1} - g_{i-1} \bar{z}_{i-1} \,, \quad i \in \{1, 2, \ldots, n-1\} \end{aligned} \tag{34}$$

$$\begin{aligned} \dot{\bar{z}}_n &= f_n + g_n \left(u - u^0 \right) + g_n u^0 + \varphi_n \theta_n - \dot{x}_{n,\mathrm{c}} - \dot{\chi}_n \\ &= -k_n \bar{z}_n - \varphi_n \eta_n - g_{n-1} \bar{z}_{n-1} \,. \end{aligned} \tag{35}$$

The stability of the closed-loop system is evaluated defining the Lyapunov function $V = W + \sum_{i=1}^n \bar{z}_i^2$, whose time-derivative along the trajectories of $\dot{\bar{z}}_i$ is

$$
\begin{aligned}
\dot{V} &= \dot{W} + 2\sum_{i=1}^{n-1} \bar{z}_i\left(-k_i\bar{z}_i - \varphi_i\eta_i + g_i\bar{z}_{i+1} - g_{i-1}\bar{z}_{i-1}\right) + 2\bar{z}_n\left(-k_n\bar{z}_n - \varphi_n\eta_n - g_{n-1}\bar{z}_{n-1}\right) \\
&= \dot{W} - 2\sum_{i=1}^{n} k_i\bar{z}_i^2 - 2\sum_{i=1}^{n}\bar{z}_i\varphi_i\eta_i + 2\sum_{i=1}^{n-1}\bar{z}_i g_i\bar{z}_{i+1} - 2\sum_{i=1}^{n-1}\bar{z}_i g_{i-1}\bar{z}_{i-1} - 2\bar{z}_n g_{n-1}\bar{z}_{n-1} \\
&= \dot{W} - 2\sum_{i=1}^{n} k_i\bar{z}_i^2 - 2\sum_{i=1}^{n}\bar{z}_i\varphi_i\eta_i \,, \qquad (36)
\end{aligned}
$$

to which the result of Lemma 1 can be applied yielding

$$
\begin{aligned}
\dot{V} &= -2\sum_{i=1}^{n} k_i\bar{z}_i^2 - 2\sum_{i=1}^{n}\bar{z}_i\varphi_i\eta_i - 2\sum_{i=1}^{n}\gamma_i(\varphi_i\eta_i)^2 \\
&= -2\sum_{i=1}^{n} k_i\bar{z}_i^2 - 2\sum_{i=1}^{n}\frac{1}{\gamma_i}\left(\frac{\bar{z}_i}{2} + \gamma_i\varphi_i\eta_i\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\frac{\bar{z}_i^2}{\gamma_i} \\
&\leq -2\sum_{i=1}^{n}\left(k_i - \frac{1}{4\gamma_i}\right)\bar{z}_i^2 \,. \qquad (37)
\end{aligned}
$$

For

$$
\bar{k}_i = k_i - \frac{1}{4\gamma_i} > 0 \qquad (38)
$$

(37) becomes

$$
\dot{V} \leq -2\sum_{i=1}^{n}\bar{k}_i\bar{z}_i^2 < 0 \,, \qquad \forall \bar{z}_i \neq 0 \,, \qquad (39)
$$

proving the global stability of the origin of the compensated tracking errors of system (23) adaptively I&I controlled by (32) together with the estimator designed in the previous section. By Barbalat's Lemma it is possible to further prove $\lim_{t\to\infty}\bar{z}_i = 0$ and $\lim_{t\to\infty}\varphi_i\eta_i = 0$. If the command filters are designed with a high enough bandwidth the compensation signals χ_i will be small, having a negligible effect on the closed-loop system. Hence, the stability properties of $\bar{z}_i$ can, in that case, be extended to the real tracking errors z_i, rendering global tracking of the reference signal. The formal proof of this statement is achieved using Singular Perturbation Theory by Farrell et al. [3].

3.2 Stability-Proof for Time-Varying Uncertainties

The algorithm described thus far has unique stability properties in the case of constant uncertain parameters. In this section, the stability properties of this law will be shown for time-varying uncertainties in systems of the form of (23). The result of this novel proof greatly widens the application span of I&I-based adaptive control, by extending its properties to systems with time-varying uncertainties.

Consider that, in system (23), $\theta_i = \theta_i(t), i \in \{1, 2, \dots, n\}$, are now scalar smooth functions. The dynamics of the *off-the-manifold* coordinate is then given by

$$\dot{\eta}_i = \dot{\hat{\theta}}_i - \dot{\theta}_i + \frac{\partial \beta_i}{\partial x_i} \dot{x}_i \,, \tag{40}$$

which, maintaining the same update law (26), yields

$$\dot{\eta}_i = -\dot{\theta}_i - \frac{\partial \beta_i}{\partial x_i} \varphi_i \eta_i \,. \tag{41}$$

If β_i is chosen as according to (28), by the result of Lemma 1, the time-derivative of the Lyapunov function $W = \sum_{i=1}^{n} \eta_i^2$ along the trajectories of (41) is

$$\dot{W} = -2 \sum_{i=1}^{n} \gamma_i \left(\varphi_i \eta_i\right)^2 - 2 \sum_{i=1}^{n} \eta_i \dot{\theta}_i \,. \tag{42}$$

Assume the following condition holds

$$|\varphi_i(x_i)|^{-1} < \rho_i \,, \quad \forall x_i \in \mathbb{R} \,, \quad \forall i \in \{1, \dots, n\} \,, \tag{43}$$

where ρ_i is a finite positive constant. Maintaining the control law designed in the previous section, the time-derivative of a control Lyapunov function defined as $V = W + \sum_{i=1}^{n} \bar{z}_i^2$, recalling (36), is now given by

$$\begin{aligned} \dot{V} &= -2 \sum_{i=1}^{n} k_i \bar{z}_i^2 - 2 \sum_{i=1}^{n} \bar{z}_i \varphi_i \eta_i - 2 \sum_{i=1}^{n} \gamma_i \left(\varphi_i \eta_i\right)^2 - 2 \sum_{i=1}^{n} \eta_i \dot{\theta}_i \\ &= -2 \sum_{i=1}^{n} \left(k_i - \frac{1}{2\gamma_i}\right) \bar{z}_i^2 - \sum_{i=1}^{n} \frac{1}{\gamma_i} \left(\bar{z}_i + \gamma_i \varphi_i \eta_i\right)^2 - \sum_{i=1}^{n} \frac{1}{\gamma_i} \left(\frac{\dot{\theta}_i}{\varphi_i} + \gamma_i \varphi_i \eta_i\right)^2 + \sum_{i=1}^{n} \frac{\dot{\theta}_i^2}{\gamma_i \varphi_i^2} \\ &\leq -2 \sum_{i=1}^{n} \bar{k}_i \bar{z}_i^2 + \sum_{i=1}^{n} \frac{\rho_i^2}{\gamma_i} \|\dot{\theta}_i\|_\infty^2 \,, \end{aligned} \tag{44}$$

where

$$\bar{k}_i = k_i - \frac{1}{2\gamma_i} > 0 \,. \tag{45}$$

Inequality (44) proves the *input-to-state* stability of the compensated error coordinates, $\bar{z}_i$, with respect to the time-derivative of the uncertainty, $\dot{\theta}_i$. Hence, the trajectories of the closed-loop system converge to a compact set around the origin of the error coordinates that depends on ρ_i, γ_i and $\|\dot{\theta}_i\|_\infty$. Assuming (43) holds, and knowing that all θ_i have bounded first time-derivative, then such set can be made arbitrarily small by increasing the estimation gain γ_i.

4 Attitude Controller Design

In this section the control theory described in the preceding one is applied to the spacecraft model earlier derived. Such model can be represented simply as

$$\begin{aligned}\dot{\boldsymbol{\sigma}}_O^B &= \boldsymbol{f}_\sigma(\boldsymbol{\sigma}_O^B) + \mathbf{B}(\boldsymbol{\sigma}_O^B)\boldsymbol{\omega}_{BI}^B \\ \mathbf{J}^B\dot{\boldsymbol{\omega}}_{BI}^B &= \boldsymbol{f}_\omega(\boldsymbol{\sigma}_O^B, \boldsymbol{\omega}_{BI}^B) + \mathbf{m}_\mathrm{c} + \mathbf{m}_\mathrm{d}\end{aligned} \tag{46}$$

where$\boldsymbol{f}_\sigma(\boldsymbol{\sigma}_O^B)$ holds the orbital angular velocity contribution to the MRP kinematics (7), and$\boldsymbol{f}_\omega(\boldsymbol{\sigma}_O^B, \boldsymbol{\omega}_{BI}^B)$ contains the Coriolis and gravity gradient induced torques in (14). $\mathbf{m}_\mathrm{c}$ is the control torque and $\mathbf{m}_\mathrm{d}$ is a time-varying unknown disturbance torque.

Starting by neglecting the action of disturbances, a baseline Backstepping controller using command filters is designed. Then, such controller is used as the full-information law in the Immersion and Invariance adaptive design. Finally, to serve as performance simulation comparison, an integrated adaptive Backstepping controller with tuning functions estimation is designed based on the same baseline law. For more details on the command filters used by these laws please refer to [23].

4.1 Non-adaptive Backstepping Controller

Assuming an undisturbed model ($\mathbf{m}_\mathrm{d} = \mathbf{0}$), a static Backstepping controller using command filters can be designed defining the error coordinates

$$\mathbf{z}_1 = \delta\boldsymbol{\sigma} \quad \text{and} \quad \mathbf{z}_2 = \boldsymbol{\omega}_{BI}^B - \mathbf{x}_{2,\mathrm{c}}\,, \tag{47}$$

which, for the use of such command filters, should be corrected as

$$\bar{\mathbf{z}}_1 = \mathbf{z}_1 \otimes (-\chi_1) \quad \text{and} \quad \bar{\mathbf{z}}_2 = \mathbf{z}_2 - \chi_2\,. \tag{48}$$

where χ_1 and χ_2 will be defined later on.

The control law

$$\mathbf{x}_{2,\mathrm{c}}^0 = -\mathbf{K}_1\left(\bar{\mathbf{z}}_1 + \chi_1\right) - \mathbf{B}^{-1}(\boldsymbol{\sigma}_O^B)\boldsymbol{f}_\sigma(\boldsymbol{\sigma}_O^B) + \mathbf{C}(\chi_1)\boldsymbol{\omega}_{BO,\mathrm{r}}^B - \chi_2\,, \quad \mathbf{K}_1 > 0 \tag{49}$$

$$\mathbf{m}_\mathrm{c}^0 = -\boldsymbol{f}_\omega(\boldsymbol{\sigma}_O^B, \boldsymbol{\omega}_{BI}^B) - \mathbf{K}_2\mathbf{z}_2 + \mathbf{J}^B\dot{\mathbf{x}}_{2,\mathrm{c}} - \bar{\mathbf{z}}_1\,, \qquad \mathbf{K}_2 > 0 \tag{50}$$

globally asymptotically stabilizes the origin of $(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2)$ of the undisturbed system (46). To prove this start by considering a command filter effect on the error coordinates given by the stable systems

$$\begin{aligned} \dot{\boldsymbol{\chi}}_1 &= \mathbf{B}(\boldsymbol{\chi}_1)\left(-\mathbf{K}_1\boldsymbol{\chi}_1 + \mathbf{x}_{2,\mathrm{c}} - \mathbf{x}^0_{2,\mathrm{c}}\right)\\ \mathbf{J}^B\dot{\boldsymbol{\chi}}_2 &= -\mathbf{K}_2\boldsymbol{\chi}_2 + \mathbf{m}_\mathrm{c} - \mathbf{m}^0_\mathrm{c}\,, \end{aligned} \tag{51}$$

where the raw control signals $\mathbf{x}^0_{2,\mathrm{c}}$ and $\mathbf{m}^0_\mathrm{c}$ are filtered to obtain $\mathbf{x}_{2,\mathrm{c}}$ and $\mathbf{m}_\mathrm{c}$ and their time-derivatives. As done in [23], such command filtering also imposes limits on value and rate of these signals.

Based on the result described by Tsiotras [24] the following Lemma will be used to prove the stability properties of the problem at hand.

Lemma 2 *The time-derivative of a control Lyapunov function defined as* $V = 2\log\left(1 + |\boldsymbol{\sigma}|^2\right)$ *taken along the trajectories of the MRP kinematics* $\dot{\boldsymbol{\sigma}} = B(\boldsymbol{\sigma})\boldsymbol{\omega}$ *is given by*

$$\dot{V} = \boldsymbol{\sigma}^\mathsf{T}\boldsymbol{\omega}$$

Proof This result is obtained simply by taking the time-derivative of V and replacing the MRP kinematics expression, noting that $\boldsymbol{\sigma}^\mathsf{T}\,[\boldsymbol{\sigma}\times] = \mathbf{0}$. □

The compensated error dynamics are written

$$\dot{\bar{\mathbf{z}}}_1 = \mathbf{B}(\bar{\mathbf{z}}_1)\left(\delta\boldsymbol{\omega} - \mathbf{C}(\bar{\mathbf{z}}_1)\mathbf{B}^{-1}\left(\boldsymbol{\chi}_1\right)\dot{\boldsymbol{\chi}}_1\right) \tag{52}$$

$$\dot{\bar{\mathbf{z}}}_2 = (\mathbf{J}^B)^{-1}\boldsymbol{f}_\omega(\boldsymbol{\sigma}^B_O, \boldsymbol{\omega}^B_{BI}) + \mathbf{m}_\mathrm{c} - \dot{\mathbf{x}}_{2,\mathrm{c}} - \dot{\boldsymbol{\chi}}_2\,. \tag{53}$$

Using the result of Lemma 2, the control Lyapunov function

$$V = 2\log\left(1 + |\bar{\mathbf{z}}_1|^2\right) + \tfrac{1}{2}\,\bar{\mathbf{z}}_2^\mathsf{T}\mathbf{J}^B\bar{\mathbf{z}}_2 \tag{54}$$

has the time-derivative along (52) and (53) given by

$$\begin{aligned} \dot{V} &= \bar{\mathbf{z}}_1^\mathsf{T}\left(\delta\boldsymbol{\omega} - \mathbf{C}(\bar{\mathbf{z}}_1)\mathbf{B}^{-1}\left(\boldsymbol{\chi}_1\right)\dot{\boldsymbol{\chi}}_1\right) + \bar{\mathbf{z}}_2^\mathsf{T}\left(\boldsymbol{f}_\omega(\boldsymbol{\sigma}^B_O, \boldsymbol{\omega}^B_{BI}) + \mathbf{m}_\mathrm{c} - \mathbf{J}^B\dot{\mathbf{x}}_{2,\mathrm{c}} - \mathbf{J}^B\dot{\boldsymbol{\chi}}_2\right)\\ &= \bar{\mathbf{z}}_1^\mathsf{T}\left(\mathbf{z}_2 + \mathbf{B}^{-1}(\boldsymbol{\sigma}^B_O)\boldsymbol{f}_\sigma(\boldsymbol{\sigma}^B_O) - \mathbf{C}(\bar{\mathbf{z}}_1)\mathbf{C}(\boldsymbol{\chi}_1)\boldsymbol{\omega}^O_{OI,\mathrm{r}} + \mathbf{K}_1\boldsymbol{\chi}_1 + \mathbf{x}^0_{2,\mathrm{c}}\right)\\ &\quad + \bar{\mathbf{z}}_2^\mathsf{T}\left(\boldsymbol{f}_\omega(\boldsymbol{\sigma}^B_O, \boldsymbol{\omega}^B_{BI}) - \mathbf{J}^B\dot{\mathbf{x}}_{2,\mathrm{c}} + \mathbf{K}_2\boldsymbol{\chi}_2 + \mathbf{m}^0_\mathrm{c}\right)\,, \end{aligned} \tag{55}$$

which, plugging in the control laws (49) and (50), yields

$$\begin{aligned} \dot{V} &= \bar{\mathbf{z}}_1^\mathsf{T}\left(\mathbf{z}_2 + \mathbf{K}_1\boldsymbol{\chi}_1 - \mathbf{K}_1\left(\bar{\mathbf{z}}_1 + \boldsymbol{\chi}_1\right) - \boldsymbol{\chi}_2\right) + \bar{\mathbf{z}}_2^\mathsf{T}\left(\mathbf{K}_2\boldsymbol{\chi}_2 - \mathbf{K}_2\mathbf{z}_2 - \bar{\mathbf{z}}_1\right)\\ &= -\,\bar{\mathbf{z}}_1^\mathsf{T}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^\mathsf{T}\mathbf{K}_2\bar{\mathbf{z}}_2 < 0\,, \qquad \forall(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2) \neq \mathbf{0}\,, \end{aligned} \tag{56}$$

which, in turn, proves the global asymptotic stability of the error origin equilibrium.

4.2 Immersion and Invariance Adaptive Backstepping Controller

Considering now the presence of a time-variant smooth disturbance torque $\mathbf{m}_\mathrm{d}$ in system (46), the static controller derived in the previous section can be made adaptive through the Immersion and Invariance design method, presented in Sect. 3.1, by using the feedback law

$$\mathbf{m}_\mathrm{c}^0 = \mathbf{m}_{\mathrm{c,BS}}^0 - \hat{\mathbf{m}}_\mathrm{d} - \boldsymbol{\beta}(\mathbf{h}_{BI}^B)\,, \tag{57}$$

where $\mathbf{m}_{\mathrm{c,BS}}^0$ is the non-adaptive Backstepping law (50) and $\boldsymbol{\beta}(\mathbf{h}_{BI}^B)$ is a function yet to be defined. Notice that, since the uncertainty is matched in the dynamics loop, the kinematics feedback law is kept the same.

As earlier described, the I&I procedure starts by defining an *off-the-manifold* coordinate

$$\boldsymbol{\eta} = \hat{\mathbf{m}}_\mathrm{d} - \mathbf{m}_\mathrm{d} + \boldsymbol{\beta}(\mathbf{h}_{BI}^B)\,, \tag{58}$$

which, in this case, has the dynamics

$$\begin{aligned}
\dot{\boldsymbol{\eta}} &= \dot{\hat{\mathbf{m}}}_\mathrm{d} - \dot{\mathbf{m}}_\mathrm{d} + \frac{\partial\boldsymbol{\beta}(\mathbf{h}_{BI}^B)}{\partial\mathbf{h}_{BI}^B}\left(\boldsymbol{f}_\omega(\boldsymbol{\sigma}_O^B, \boldsymbol{\omega}_{BI}^B) + \mathbf{m}_\mathrm{c} + \mathbf{m}_\mathrm{d}\right)\\
&= \dot{\hat{\mathbf{m}}}_\mathrm{d} - \dot{\mathbf{m}}_\mathrm{d} + \frac{\partial\boldsymbol{\beta}(\mathbf{h}_{BI}^B)}{\partial\mathbf{h}_{BI}^B}\left(\boldsymbol{f}_\omega(\boldsymbol{\sigma}_O^B, \boldsymbol{\omega}_{BI}^B) + (\mathbf{m}_\mathrm{c} - \mathbf{m}_\mathrm{c}^0) + \mathbf{m}_\mathrm{c}^0 + \hat{\mathbf{m}}_\mathrm{d} + \boldsymbol{\beta}(\mathbf{h}_{BI}^B) - \boldsymbol{\eta}\right)\\
&= \dot{\hat{\mathbf{m}}}_\mathrm{d} - \dot{\mathbf{m}}_\mathrm{d} + \frac{\partial\boldsymbol{\beta}(\mathbf{h}_{BI}^B)}{\partial\mathbf{h}_{BI}^B}\left(-\mathbf{K}_2\bar{\mathbf{z}}_2 + \mathbf{J}^B\dot{\boldsymbol{\chi}}_2 + \mathbf{J}^B\dot{\mathbf{x}}_{2,\mathrm{c}} - \bar{\mathbf{z}}_1 - \boldsymbol{\eta}\right)\;.
\end{aligned} \tag{59}$$

Defining the estimate update law as

$$\dot{\hat{\mathbf{m}}}_\mathrm{d} = \frac{\partial\beta(\mathbf{h}_{BI}^B)}{\partial\mathbf{h}_{BI}^B}\left(\mathbf{K}_2\bar{\mathbf{z}}_2 - \mathbf{J}^B\dot{\boldsymbol{\chi}}_2 - \mathbf{J}^B\dot{\mathbf{x}}_{2,c} + \bar{\mathbf{z}}_1\right)\;. \tag{60}$$

renders (59)

$$\dot{\boldsymbol{\eta}} = -\dot{\mathbf{m}}_\mathrm{d} - \frac{\partial\boldsymbol{\beta}(\mathbf{h}_{BI}^B)}{\partial\mathbf{h}_{BI}^B}\boldsymbol{\eta}\;. \tag{61}$$

Selecting a control Lyapunov function as

$$V = 2\log\left(1 + |\bar{\mathbf{z}}_1|^2\right) + \tfrac{1}{2}\,\bar{\mathbf{z}}_2^{\mathsf{T}}\mathbf{J}^B\bar{\mathbf{z}}_2 + \tfrac{1}{2}\,\boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{\eta} \tag{62}$$

and choosing

$$\boldsymbol{\beta}(\mathbf{h}_{BI}^B) = \boldsymbol{\Gamma}\mathbf{h}_{BI}^B\,, \qquad \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^{\mathsf{T}} > 0\,, \tag{63}$$

yields

$$
\begin{aligned}
\dot{V} &= -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\mathbf{K}_2\bar{\mathbf{z}}_2 - \bar{\mathbf{z}}_2^{\mathrm{T}}\boldsymbol{\eta} - 2\boldsymbol{\eta}^{\mathrm{T}}\dot{\mathbf{m}}_{\mathrm{d}} - \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\Gamma}\boldsymbol{\eta} \\
&= -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\left(\mathbf{K}_2 - \tfrac{1}{2}\boldsymbol{\Gamma}^{-1}\right)\bar{\mathbf{z}}_2 - \tfrac{1}{2}\left(\bar{\mathbf{z}}_2 + \boldsymbol{\Gamma}\boldsymbol{\eta}\right)^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\left(\bar{\mathbf{z}}_2 + \boldsymbol{\Gamma}\boldsymbol{\eta}\right) \\
&\quad - \tfrac{1}{2}\left(\boldsymbol{\eta} + \boldsymbol{\Gamma}^{-1}\dot{\mathbf{m}}_{\mathrm{d}}\right)^{\mathrm{T}}\boldsymbol{\Gamma}\left(\boldsymbol{\eta} + \boldsymbol{\Gamma}^{-1}\dot{\mathbf{m}}_{\mathrm{d}}\right) + \tfrac{1}{2}\dot{\mathbf{m}}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\dot{\mathbf{m}}_{\mathrm{d}} \\
&\leq -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\bar{\mathbf{K}}_2\bar{\mathbf{z}}_2 + \tfrac{1}{2}\dot{\mathbf{m}}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\dot{\mathbf{m}}_{\mathrm{d}}\ . \qquad (64)
\end{aligned}
$$

For

$$\bar{\mathbf{K}}_2 = \mathbf{K}_2 - \frac{1}{2}\boldsymbol{\Gamma}^{-1} > 0\ , \qquad (65)$$

condition (64) proves the *input-to-state* stability of $(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2)$, and by proper filter design also of $(\mathbf{z}_1, \mathbf{z}_2)$, with respect to $\dot{\mathbf{m}}_{\mathrm{d}}$, for the feedback (49) and (50) with estimate update law (60) and $\boldsymbol{\beta}(\mathbf{h}_{BI}^{B})$ function (63). Hence, for bounded $\dot{\mathbf{m}}_{\mathrm{d}}$, the compact set around the origin to which the pair of tracking error coordinates converge to can be made arbitrarily small by increasing the entries of the gain matrix $\boldsymbol{\Gamma}$.

4.3 Adaptive Backstepping Controller with Tuning Functions

In order to compare the performance of the I&I adaptive design a "traditional" adaptive Backstepping controller with tuning function estimation is here developed. This type of control law compensates for the presence of the disturbance term using the feedback law

$$\mathbf{m}_{\mathrm{c}}^{0} = \mathbf{m}_{\mathrm{c,BS}}^{0} - \hat{\mathbf{m}}_{\mathrm{d}}\ . \qquad (66)$$

where $\mathbf{m}_{\mathrm{c,BS}}^{0}$ is the non-adaptive Backstepping control law (50).

In the definition of the control Lyapunov function the estimation error $\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}} = \mathbf{m}_{\mathrm{d}} - \hat{\mathbf{m}}_{\mathrm{d}}$ is accounted for as

$$V = 2\log\left(1 + |\bar{\mathbf{z}}_1|^2\right) + \tfrac{1}{2}\bar{\mathbf{z}}_2^{\mathrm{T}}\bar{\mathbf{z}}_2 + \tfrac{1}{2}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\ , \qquad \boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\mathrm{T}} > 0\,. \qquad (67)$$

The time-derivative of such Lyapunov function is

$$
\begin{aligned}
\dot{V} &= -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\mathbf{K}_2\bar{\mathbf{z}}_2 + \bar{\mathbf{z}}_2^{\mathrm{T}}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}} + \boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\dot{\mathbf{m}}_{\mathrm{d}} - \boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\dot{\hat{\mathbf{m}}}_{\mathrm{d}} \\
&= -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\mathbf{K}_2\bar{\mathbf{z}}_2 + \boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{T}\left(\bar{\mathbf{z}}_2 - \boldsymbol{\Lambda}^{-1}\dot{\hat{\mathbf{m}}}_{\mathrm{d}}\right) - \tfrac{1}{2}\left(\dot{\mathbf{m}}_{\mathrm{d}} - \boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\right)^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\left(\dot{\mathbf{m}}_{\mathrm{d}} - \boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\right) \\
&\quad + \tfrac{1}{2}\dot{\mathbf{m}}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\dot{\mathbf{m}}_{\mathrm{d}} + \tfrac{1}{2}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}} \\
&\leq -\bar{\mathbf{z}}_1^{\mathrm{T}}\mathbf{K}_1\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2^{\mathrm{T}}\mathbf{K}_2\bar{\mathbf{z}}_2 + \tfrac{1}{2}\dot{\mathbf{m}}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\dot{\mathbf{m}}_{\mathrm{d}} + \tfrac{1}{2}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\,, \qquad (68)
\end{aligned}
$$

where inequality (68) is achieved through the choice of the estimate update law

$$\dot{\hat{\mathbf{m}}}_{\mathrm{d}} = \mathrm{Proj}\left(\boldsymbol{\Lambda}\bar{\mathbf{z}}_2\right)\ . \qquad (69)$$

Notice that, in contrast to the I&I design case, here the pair of tracking errors is *input-to-state* stable with respect to both $\dot{\mathbf{m}}_\mathrm{d}$ and $\boldsymbol{\delta}\mathbf{m}_\mathrm{d}$, meaning that, in addition to the smoothness condition of the disturbance torque, the estimation error has to remain bounded. This is achieved by using the Projection operator (Proj) on the estimate update law (69), which is here defined as done by [18]. Also in contrast with the I&I framework is the fact that the estimate $\hat{\mathbf{m}}_\mathrm{d}$ is not guaranteed to converge to actual $\mathbf{m}_\mathrm{d}$, rather providing an *integrative* action to the rate control.

5 Simulation and Performance Analysis

In this section the designed controllers are tested and compared in the presence of two different disturbance torques, separately applied, one produced by a saturated uncontrolled reaction wheel and one originated by an oscillating payload part. The disturbance dynamics model for each of this cases can be found in [23].

For fairness of comparison, the controller gains $\mathbf{K}_1$ and $\mathbf{K}_2$ were defined equally for the three laws, as $\mathbf{K}_1 = 0.4\mathbf{I}$ and $\mathbf{K}_2 = 5\mathbf{I}$. The adaptive gains were matched, as $\boldsymbol{\Gamma} = 20\mathbf{I}$ and $\boldsymbol{\Lambda} = 20\mathbf{I}$. The laws are here referred to as: S-BS, the (static) non-adaptive Backstepping controller of Sect. 4.1; I&I-BS, the Immersion and Invariance controller in Sect. 4.2; and TF-BS, the Tuning Functions adaptive Backstepping law of Sect. 4.3. The attitude reference signal used includes a slew, a steady-pointing period and a sweeping (or scanning) motion, as shows Fig. 1. The performance is assessed through the evaluation of two error quantities: the absolute pointing error, $\theta_e = 4\arctan\left(\|\boldsymbol{\delta\sigma}\|\right)$, and the norm of the angular velocity tracking error, $\omega_e = \|\boldsymbol{\omega}^B_{BI} - \mathbf{x}_{2,\mathrm{c}}\|$. The project requirements for the attitude controller's performance are $\theta_e \leq 30$ arcsec and $\omega_e \leq 1$ arcsec/s. The spacecraft has the coarse inertia matrix $\mathbf{J}^B = \mathsf{diag}\left([3.083,\ 3.083,\ 2.083]\right)\,\mathrm{kg\,m}^2$, and flies in an approximately circular orbit of 650 km altitude, with orbital rate n_o of 1.073 m rad/s. The control laws run at a rate of 10 Hz.

Nominal Performance

In a disturbance-free scenario all three controllers succeed in maintaining the tracking errors bellow the required levels (Fig. 2). Table 1 shows the Root-Mean-Square (RMS) pointing error. The non-adaptive and the I&I laws achieve a similar pointing

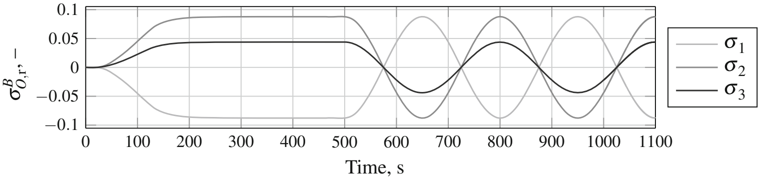


Fig. 1 MRP attitude reference trajectory

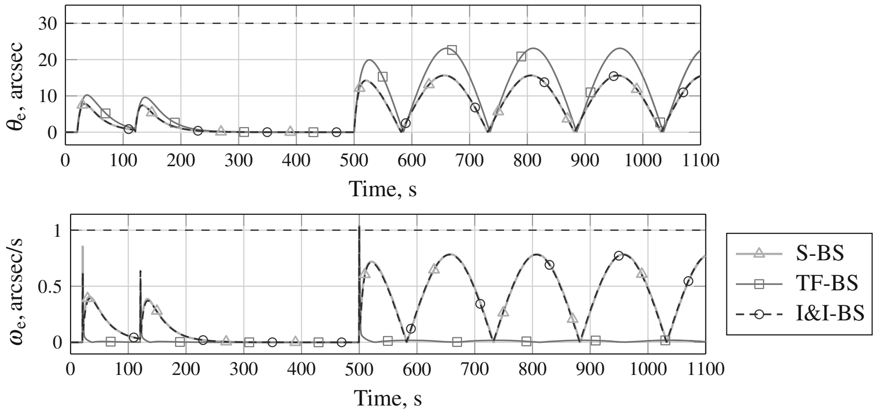


Fig. 2 Pointing and Rate tracking errors in nominal case

Table 1 Nominal case pointing error RMS

	S-BS	TF-BS	I & I-BS
$\theta_{\mathrm{e,RMS}}$, arcsec	8.161	11.899	8.154

Table 2 Nominal case estimation error RMS

	TF-BS	I&I-BS
$\|\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\|_{\mathrm{RMS}}$, μNm	43.203	10.137

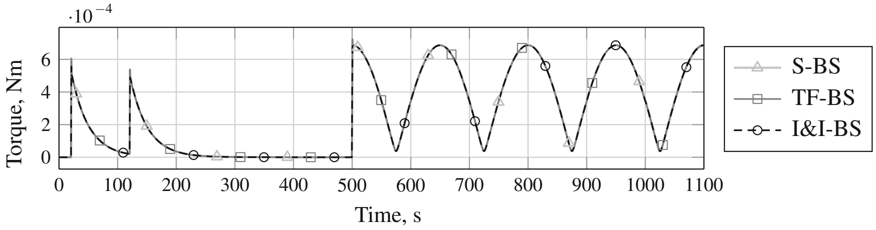


Fig. 3 Actuation torque in nominal case

performance while the tuning functions one lags slightly behind. This is explained by the *integrative* action of the adaptation law which promotes rate-tracking, compromising pointing accuracy. This is also reflected on the high RMS of the estimated perturbation error in comparison to the I&I (Table 2). The actuation history is very similar among the three controllers (Fig. 3), being differences only perceived when computing the total workload (Table 3). This reveals a marginally higher effort by the TF-BS law, certainly explained by the close tracking of the rate command.

Table 3 Total workload in nominal case

	S-BS	TF-BS	I&I-BS
Workload, mJ	1.5103	1.5104	1.5103

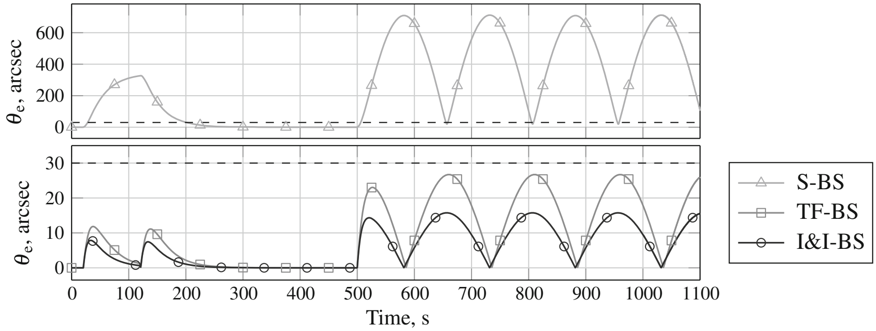


Fig. 4 Pointing error in the uncontrolled reaction-wheel case

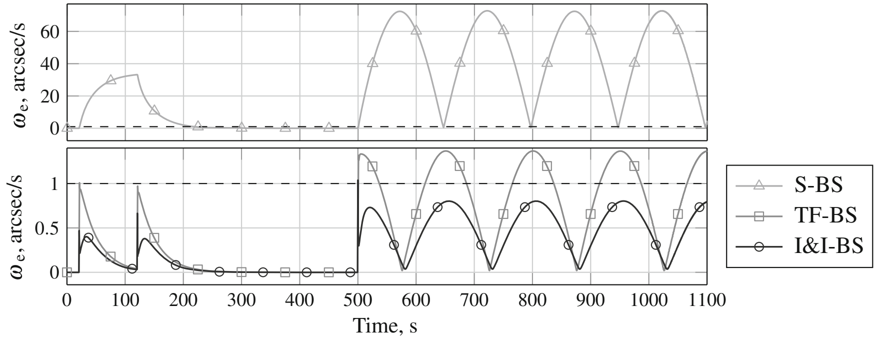


Fig. 5 Rate tracking error in the uncontrolled reaction-wheel case

Uncontrolled Reaction Wheel

Wheel anomalies are one of the most common GNC-related failures. The Iridium 42 mission, for instance, failed due to a faulty tachometer in one of the wheels, which lead to unknown real actuation [19]. In this test, the effect of an uncontrolled saturated (redundant) reaction wheel is modelled. A 1 kg wheel is assumed stuck at 4200 rpm producing a strong gyroscopic moment unknown to the control system. Figures 4 and 5 show that the non-adaptive control cannot cope with such a strong disturbance. The adaptive laws, on the other hand, are quite successful in this case. While the TF-BS controller sees a 54% pointing degradation (Table 4), the I&I-BS law is seemingly undisturbed, maintaining virtually the same performance as in the nominal scenario. This is explained by the extremely low disturbance estimation error by this controller (Table 5), which is roughly one order of magnitude better than that of the TF law. The

Table 4 Pointing error RMS in the uncontrolled reaction-wheel case

$\theta_{\mathrm{e,RMS}}$	S-BS	TF-BS	I&I-BS
Value, arcsec	384.370	18.271	8.210
Ratio to nominal	47.099	1.536	1.007

Table 5 Estimation error RMS in the uncontrolled reaction-wheel case

$\|\boldsymbol{\delta}\mathbf{m}_{\mathrm{d}}\|_{\mathrm{RMS}}$	TF-BS	I&I-BS
Value, μNm	148.957	15.915
Ratio to $\|\mathbf{m}_{\mathrm{d}}\|_{\mathrm{RMS}}$	0.044	0.005

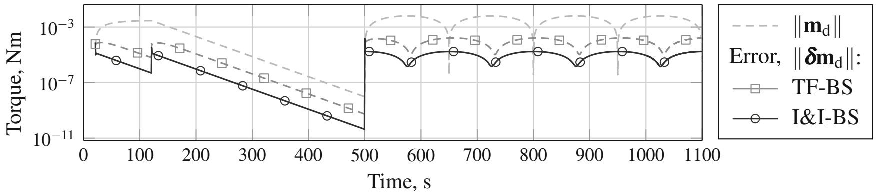


Fig. 6 Total disturbance and estimation error norms in the uncontrolled reaction-wheel case

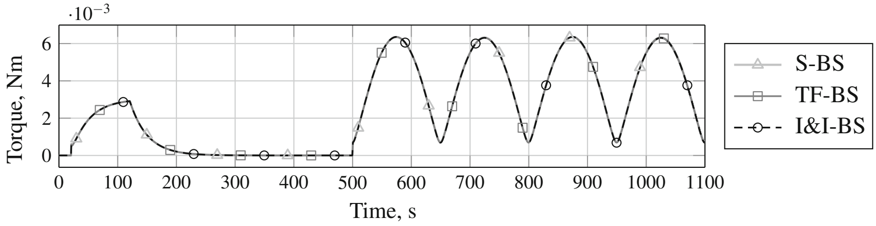


Fig. 7 Actuation torque in an uncontrolled reaction-wheel case

same can be seen in Fig. 6, where the estimation errors of the adaptive laws are plot against the total disturbance (in norm). Note that, in this case, the TF-BS controller does not manage to properly track the rate signal, violating the requirement. As Fig. 7 shows, again, the laws' control actions are nearly indistinguishable. The total workload (Table 6) reveals that the adaptive schemes require slightly less effort. The I&I-BS law is the least demanding here.

Table 6 Total workload in the uncontrolled reaction-wheel case

	S-BS	TF-BS	I&I-BS
Workload, mJ	18.7416	18.7251	18.7237

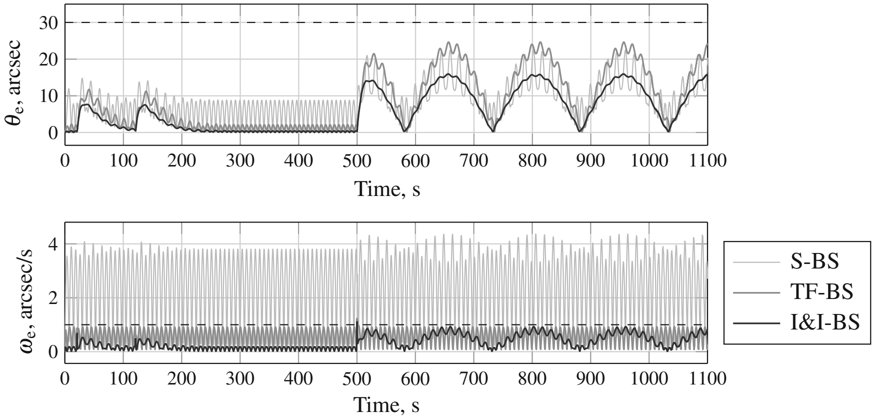


Fig. 8 Pointing and Rate tracking errors in a moving payload case

Moving Payload

Many spacecraft instruments/payloads have moving parts. An example is the moving mirror of NASA's Geostationary Imaging Fourier Transform Spectrometer [7].

In the present case, a smaller spectrometer mirror is considered and assumed to be unaccounted for in the control design. The piece, a 0.5 cm × 5 cm × 5 cm cuboid of 30 g, oscillates about its own center of mass. Its angular orientation is modelled as a sinusoidal function with 15 s of period and peak-to-peak amplitude of 90 deg.

While the required pointing is achieved by the three controllers (Fig. 8), although with obvious oscillations on the part of S-BS and TF-BS, rate tracking proves to be the main difficulty. Only the adaptive schemes comply with the rate requirement. Among these the I&I is clearly the least disturbed, again, roughly maintaining its nominal performance (Table 7). As Table 8 shows, the estimation deteriorated for both adaptive laws with respect to the previous scenario. As before, the I&I observer yielded about one order of magnitude lower estimation error than the TF law (Fig. 9).

Table 7 Pointing error RMS in a moving payload case

$\theta_{e,\mathrm{RMS}}$	S-BS	TF-BS	I&I-BS
Value, arcsec	10.280	12.210	8.161
Ratio to nominal	1.260	1.026	1.001

Table 8 Estimation error RMS in a moving payload case

$\|\boldsymbol{\delta}\mathbf{m}_d\|_{\mathrm{RMS}}$	TF-BS	I&I-BS
Value, μNm	83.858	12.990
Ratio to $\|\mathbf{m}_d\|_{\mathrm{RMS}}$	0.535	0.083

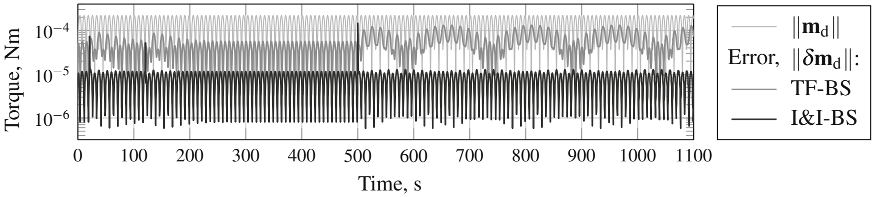


Fig. 9 Total disturbance and estimation error norms in a moving payload case

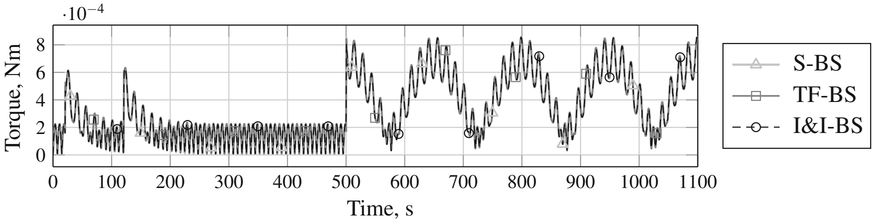


Fig. 10 Actuation torque in a moving payload case

Table 9 Total workload in a moving payload case

	S-BS	TF-BS	I&I-BS
Workload, mJ	1.6411	1.6394	1.6357

Figure 10 shows that, as in the two previous cases, control action is very similar among the three controllers. The total workload, displayed on Table 9, reveals that once again the I&I-BS demanded a notch less control action.

6 Conclusion

The I&I adaptive control framework, combined with nonlinear Backstepping, has been used to develop an attitude controller for a micro-satellite. A novel *input-to-state* stability result for a class of I&I controlled strict-feedback systems with time-varying uncertainties has been introduced. This allowed a rather general modelling of the rotational dynamics uncertainty. Results for two different severely disturbed cases clearly revealed superior robustness of the I&I controller in comparison to a baseline standard Backstepping law and to a "traditional" Tuning Functions adaptive scheme. In fact, the I&I control law managed to virtually keep its nominal performance in all scenarios, having in the disturbance-free case matched the baseline (full-information) law. The tight tracking of the disturbance torque by the I&I estimator greatly contributed to this result. Although the differences in control torque were marginal in all cases, the proposed I&I law demanded the least control effort.

References

1. Astolfi A, Ortega R (2001) Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems. In: IFAC symposium on nonlinear control system design
2. Atassi AN, Khalil HK (1999) A separation principle for the stabilizationof a class of nonlinear systems. IEEE Trans Autom Control 44(9):1672–1687
3. Farrell JA, Polycarpou M, Sharma M (2003) Adaptive backstepping with magnitude, rate, and bandwidth constraints: aircraft longitude control. In: Proceedings of the annual American control conference
4. Farrell JA et al (2009) Command filtered backstepping. IEEE Trans Autom Control 54:1391–1395
5. Gill E et al (2010) Atmospheric aerosol characterization with the Dutch-Chinese fast formation flying mission. Acta Astronautica 66:1044–1051
6. Guo J et al (2009) Status of the FAST mission: micro-satellite formation flying for technology, science and education. In: 60th international astronautical congress
7. Juang J (2004) Instrument attitude precision control. Technical report. Langley Research Center, NASA, Virginia
8. Junkins JL, Singla P (2004) How nonlinear is it? a tutorial on nonlinearity of orbit and attitude dynamics. J Astronaut Sci 52:7–60
9. Kanellakopoulos I, Kokotović P, Morse AS (1991) Systematic design of adaptive controllers for feedback linearizable systems. IEEE Trans Autom Control 36(11):1241–1253
10. Karagiannis D, Astolfi A (2008) Nonlinear adaptive control of systems in feedback form: an alternative to adaptive backstepping. Syst Control Lett 57:733–739
11. Karagiannis D, Astolfi A (2005) Nonlinear observer design using invariant manifolds and applications. In: 44th IEEE conference on decision and control
12. Karagiannis D, Carnevale D, Astolfi A (2008) Invariant manifold based reduced-order observer design for nonlinear systems. IEEE Trans Autom Control 53(11):2602–2614
13. Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Englewood Cliffs
14. Kokotović P, Arcak M (2001) Constructive nonlinear control: a historical perspective. Autimatica 37:637–662
15. Krstić M, Kanellakopoulos I, Kokotović V (1995) Nonlinear and adaptive control design. Wiley-Interscience, New Jersey
16. Li C, Ma G (2007) Adaptive backstepping control for attitude tracking of a spacecraft. In: IEEE international symposium on industrial electronics
17. Maessen D et al (2009) Mission design of the Dutch-Chinese FAST micro-satellite mission. In: 7th IAA symposium on small satellites for earth observation
18. Pomet J, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Trans Autom Control 37:729–740
19. Robertson B, Stoneking E (2003) Satellite GN&C anomaly trends. In: 26th annual AAS guidance and control conference
20. Schaub H, Junkins JL (1996) Stereographic orientation parameters for attitude dynamics: a generalization of the Rodrigues parameters. J Astronaut Sci 44:1–19
21. Shuster MD (1993) A survey of attitude representations. J Astronaut Sci 41(4):439–517
22. Sonneveldt L et al (2010) Immersion and invariance based nonlinear adaptive flight control. In: AIAA guidance, navigation and control conference and exhibit
23. Trigo GF (2011) Robust and adaptive nonlinear attitude control of a spacecraft: a comparison of backstepping-based designs. MA thesis. Delft University of Technology
24. Tsiotras P (1994) New control laws for the attitude stabilization of rigid bodies. In: IFAC symposium on automatic control in aerospace
25. van Oort ER et al (2010) Full envelope modular adaptive control of a fighter aircraft using orthogonal least-squares. J Guid Navig Dyn 33(5):1461–1472
26. Wie B (2008) In: Schetz JA (ed) Space vehicle dynamics and control, 2nd edn. AIAA
27. Wiesel WE (1997) Spaceflight dynamics, 2nd edn. Irwin/McGraw-Hill
28. Yip PP, Hedrick JK (1998) Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems. Inter J Control 71:959–979

Terrain Relative Navigation for Planetary Landing Using Stereo Vision Measurements Obtained from Hazard Mapping

Svenja Woicke and Erwin Mooij

1 Introduction

The task of precision landings on other planets, moons, and asteroids has been receiving an increasing amount of attention in recent years. For instance, NASA's Mars2020 lander, scheduled to land on Mars in 2020, is supposed to fly with a terrain-relative navigation (TRN) capability, as it was found that 9 out of 10 potential landing sites would require TRN for a safe landing [3]. This need is, of course, driven by the desire to land in regions, which are more complex and contain more hazards than those considered for its predecessor, the Mars Science Laboratory lander.

Desiring the capabilities to land in regions, which contain hazards, thus requiring hazard avoidance, will always lead to the necessity of using TRN. This being said, it can be concluded that if a TRN system is used, a hazard detection and avoidance (HDA) system will likely also be in place. HDA can have various different appearances, for this research we assume a set-up where hazard detection is done on-board, in real-time and autonomously. Such a system will generate surface digital elevation models (DEMs), as these are needed to compute slope and roughness of the local terrain in the landing region. The same kind of data is used as input in some TRN systems for their localisation efforts.

In this research we are trying to make use of the HDA DEMs as an input to the TRN system. This is an effort to partially combine these two systems and decrease the computational burden by reusing data, which is already generated. Since HDA data is only available at a very late stage of the lander's descent towards the surface, it is attempted to increase the localisation relative to the image, and therefore a landing site selected in the images. This means that our algorithm is not capable of performing a pin-point landing with respect to an inertially defined landing site. But it can perform an accurate landing with respect to a landing site selected in the HDA

S. Woicke (✉) · E. Mooij
Faculty of Aerospace Engineering, Delft University of Technology,
Klyverweg 1, 2629 HS Delft, The Netherlands
e-mail: s.woicke@tudelft.nl

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_39

DEMs. This is a capability that would be necessary for a descent system with active hazard avoidance. This is sometimes also refereed to as hazard-relative navigation.

Most terrain-relative navigation developments so far focused on methods that can locate the vehicle absolute to known surface features, for example, as presented in Ref. [15]. This results in accurate inertial localisation of the lander and enables precise landing relative to an inertially defined landing site. However, this kind of navigation can only be performed at higher altitudes as a-priori maps are not available at sufficient resolution to use this algorithm at lower altitudes. Therefore, it does not enable hazard relative navigation at lower altitudes when hazards become visible.

However, there are also a few of studies that did investigate TRN at lower altitudes in the absence of mapped surface features. In Ref. [12] such an approach is introduced, which uses measurements of both a-priori known and unknown features, depending on the current altitude. Monocular vision (a single camera) is used to select these features. A sounding-rocket test of this algorithm was performed and delivered promising results. Still, this method does not render the camera pose observable at lower altitudes, but only rotation and velocity can be retrieved from the successive images. The results show that position and velocity error increase when switching to the tracking of only unknown features. Therefore, it can be concluded that a different approach should be used at lower altitudes.

An approach comparable to this method is presented in Ref. [6]. Filter-wise the methods are the same, but two different techniques to identify the landmarks are used and compared in this work. In this case, only absolute localisation is performed with respect to landmarks identified in an a priori map. However, it is assumed that the filter can run down to altitudes of 10 m, which would require a-priori surface maps of very high resolution.

A more simultaneous localisation and mapping (SLAM)-like approach for TRN is presented in Ref. [7], however, the paper presents the algorithm only at a conceptual level and does not provide any results, neither from simulations nor from experiments. This method still used known features in combination with line-of-sight measurements towards these features. SLAM is employed in this approach, as this makes the velocity measurable using this method. A sparse-weight Kalman filter was chosen to handle the feature positions in a more effective manner inside the filter.

Reference [5] introduces a hazard-detection method, which only focuses on boulders, and does thus not reconstruct full DEMs. Still, it is an interesting reference, as it does include the locations of these boulders in an Extended Kalman Filter (EKF) for estimation of updates hazard maps during later stages of a descent. However, the main focus is hazard-map prediction and improvement by using the boulder measurements in combination with the EKF, rather than using this information for navigation purposes. Therefore, the influence of this approach on the localisation performance is not presented, even though a positive effect is to be expected.

In the robotics community plenty of solutions to the SLAM problem exist. However, these systems have different constraints and requirements. Often their motion is considerably slower than that of a landing vehicle. Moreover, in robotics the modelling of the system's dynamics is usually highly simplified and not as well modelled as for landing applications, since they are usually also less predictable. Also the utilization of revisiting points, which were already imaged at a later stage, so-called loop closure, is something that is frequently studied for robotic systems. However, loop closers do not and also cannot occur during a landing. Moreover, the robotics community turned away from the Kalman-filter approaches for localisation, but rather uses different methods for localisation and trajectory estimation. Yet, for space-flight applications Kalman filters are still the most common choice with a lot of flight heritage. For example, Ref. [13] presents a stereo-based SLAM algorithm. However, the presented algorithm does not use a Kalman-filter based approach, but bundle adjustment. Reference [9] uses yet another method for trajectory estimation, namely splines.

We will follow a SLAM-like approach, which means that the selected and tracked surface features are not only used for measurement purposes, but are also added to the navigation state. This does not only improve the navigation performance, but also improves the localisation accuracy of these surface features. This information might be interesting for other purposes, such as map building or hazard detection. However, at the current stage of our research this data is not exploited yet.

The input DEMs are computed using a stereo-vision (SV) based hazard-detection algorithm, based on previous work [16]. Stereo vision was chosen over other camera-based methods as in [17] we were able to show that stereo vision is the most suitable approach for detecting hazards during the final phase before touch-down. In that work it was shown that at low altitudes both stereo-from-motion and shape-from-shading based algorithms are outperformed by stereo vision, therefore we choose this method in this work. However, DEMs from any other hazard-detection method could be used as an input for this method as well, here it would also be possible to use DEMs from active sensors such as a lidar.

The paper is set-up as follows, first, in Sect. 2 we will give a short background on the stereo-vision based hazard-detection algorithm used for creation of the DEMs. After this the TRN algorithm is discussed in Sect. 3. Results of the algorithm in combination with a performance analysis are presented in Sect. 4. The paper closes with conclusions and recommendations in Sect. 5.

2 Stereo Vision Surface Mapping

As mentioned above, the stereo-based mapping algorithm used in this work is the same as used for hazard-map construction [16]. For completeness, a quick summary of the stereo-vision algorithm is cited. For a more detailed discussion the reader is referred to the reference.

To generate pictures of a planet's surface using the principle of stereo vision, two cameras need to be attached to the lander with a known baseline (Fig. 1). This baseline is constrained by the lander's dimensions and is assumed to be limited to 2–3 m, based on current lander designs. In this paper, a 2 m baseline is assumed, as this represents a more challenging condition, while still being realistic. The two cameras will generate two input images for the SV algorithm. Stereo vision makes use of the fact that when taking two pictures of the same scene from different viewpoints the projections of the same ground points will appear at different image positions. Finding the change in image position for a given ground point can be used to compute the distance towards this point from triangulation. Linking a pixel in the left image to the pixel representing the same ground point in the right image is called image matching. Comparing the two different locations results in a value for the shift of that pixel. This shift is the so-called disparity, d. In image matching a cost function is used to determine which two pixels are the "perfect" match. The proposed algorithm is using the sum of squared difference (SSD) as cost function:

$$\mathrm{SSD} = \sum_{u,v} (I_1(u,v) - I_2(u+d,v))^2 \tag{1}$$

where $I_1(u,v)$ and $I_2(u,v)$ are the pixel intensities at the pixel positions u, v for the left and the right image, respectively. The algorithm computes non-integer disparities, $\hat{d}_{\min}$, using a parabolic fit through the point of minimum disparity, $d_{\min}$:

$$\hat{d}_{\min} = d_{\min} - \frac{C(d_{\min}-1) - C(d_{\min}+1)}{4C(d_{\min}) - 2C(d_{\min}-1) - 2C(d_{\min}+1)} \tag{2}$$

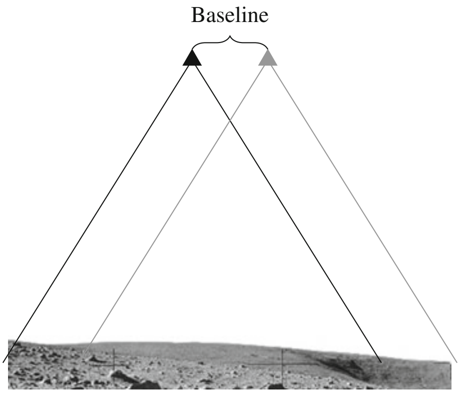
where C is a cost function; as described above the cost function used is the SSD. Knowing the disparity, the distance from the cameras to the surface can be computed using

$$z(d) = \frac{fb}{d} \tag{3}$$

where both the focal length f and the baseline b are fixed by the design, and the disparity d was found from image matching.

Finding the match for each pixel in the left image and computing the distances from the camera to the surface for each of them will lead to a dense DEM. To remove

Fig. 1 Stereo vision



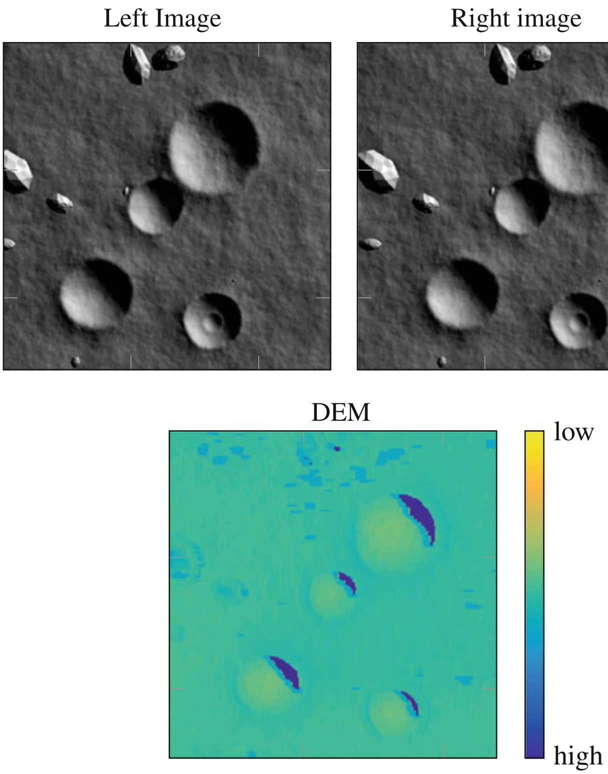


Fig. 2 Stereo DEM and stereo input

noise and outliers from the DEMs the resulting maps are filtered using a median filter before they are made available for further processing of either the hazard detection or the TRN function.

Figure 2 gives an example of an input data set and the resulting stereo DEM. It can be seen that the craters and boulders are well reconstructed and are thus clearly visible in the DEM. Furthermore, one can see that the terrain cannot be reconstructed in the shaded regions of the craters, which is always the case for camera-based methods and can only be overcome by using an active sensor. All in all, the shown DEM supports the conclusions drawn in [16]: stereo vision can construct the scene sufficiently well to perform successful hazard detection.

3 TRN Algorithm

For the task of state estimation an error-state Kalman filter (ESKF), also called indirect Kalman filter, was chosen. This filter estimates the error in the state rather than the full state. This turns the non-linear problem of a common Extended Kalman Filter

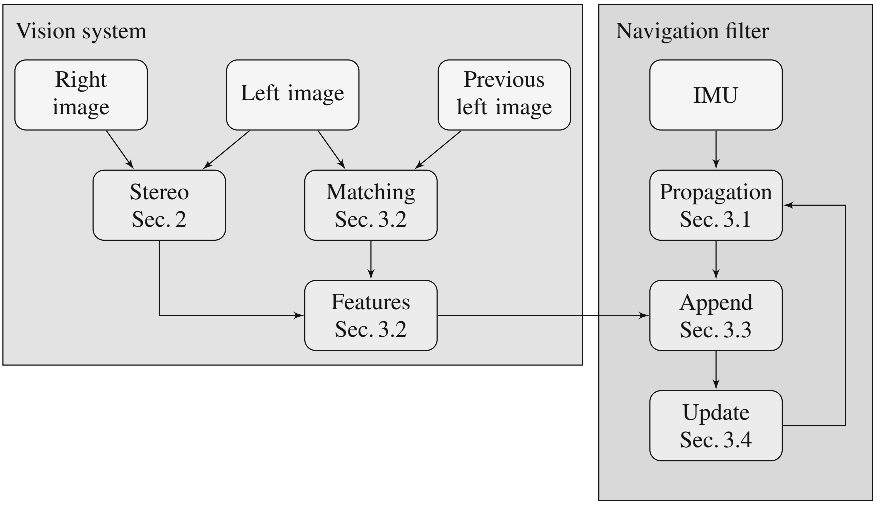


Fig. 3 System set-up

(EKF) into a linear problem, provided that the error is small. Therefore, no linearisation are necessary in the ESKF opposed to an EKF. The ESKF avoids the problem of the over-parametrisation of a quaternion for attitude representation as well, as the attitude is represented by an error descriptor, which is a minimal representation. Reference [11] compares the EKF to an ESKF for attitude estimation. It was found that the ESKF is in general more robust than the EKF. Since both Refs. [6, 12] make use of a similar approach for state estimation, we also selected the ESKF. In Ref. [11] a more detailed discussion of the benefits of an ESKF over a standard EKF is given.

Figure 3 shows the set-up of the full TRN system. The vision system and the navigation filter are indicated as distinct sub-groups. The former uses two stereo images as well as one image taken at the last measurement step as an input. The stereo pair is used for depth estimation (Block **Stereo**), while the image from the previous measurement step and left image of the current stereo pair are used for feature matching and tracking (Block **Matching** and **Features**). In the navigation loop an IMU is used for state propagation (Block **Propagation**). The IMU is modelled using a bias and an additive white noise for both the accelerometers and the gyroscopes. Misalignment, scale and other error sources are not considered yet, but could easily be added to the system. The measured acceleration and angular rate are thus described by

$$\boldsymbol{a}(t)_m = \boldsymbol{a}(t) + \boldsymbol{b}_a(t) + \boldsymbol{n}_a(t) \tag{4}$$

$$\boldsymbol{\omega}(t)_m = \boldsymbol{\omega}(t) + \boldsymbol{b}_g(t) + \boldsymbol{n}_g(t) \tag{5}$$

where $\boldsymbol{n}$ are the white noises and $\boldsymbol{b}$ are the biases. The bias is described by a random walk process:

$$\dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{ba} \tag{6}$$

$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{bg} \tag{7}$$

The bias in discrete time can be found by integrating these equations.

Once a measurement is recorded new features are added to the state (Block **Append**). Re-observed features are used in the update step to update the state (Block **Update**). The navigation loop is executed until touchdown or the filter is stopped. In this section, each of the blocks will be discussed in more detail.

3.1 Propagation

The true state consists of 16 elements

$$\boldsymbol{x} = [\boldsymbol{r}^T \boldsymbol{v}^T \boldsymbol{q}^T \boldsymbol{b}_g^T \boldsymbol{b}_a^T] \tag{8}$$

with:

- $\boldsymbol{r}$ the position of the lander in the global reference frame, G, which is a non-rotating frame fixed to the main body's center. The dimension is [3 × 1]
- $\boldsymbol{v}$ the velocity of the lander in the global reference frame. The dimension is [3 × 1]
- $\boldsymbol{q}$ the quaternion representing the rotation from the global to the body reference frame, $(\boldsymbol{q}_G^B)$. The dimension is [4 × 1]
- $\boldsymbol{b}_g$ the IMU gyroscope measurement bias, also called drift. The dimension is [3 × 1]
- $\boldsymbol{a}_g$ the IMU accelerometer measurement bias. The dimension is [3 × 1]

The error state is defined as

$$\delta\boldsymbol{x} = [\delta\boldsymbol{r}^T \delta\boldsymbol{v}^T \delta\boldsymbol{\theta}^T \delta\boldsymbol{b}_g^T \delta\boldsymbol{b}_a^T] \tag{9}$$

To avoid the non-minimal representation of the full quaternion the ESKF uses the three-dimensional error $\delta\boldsymbol{\theta}$ in place of the error quaternion $\delta\boldsymbol{q}$. The error quaternion is defined as:

$$\delta\boldsymbol{q} = [\delta\boldsymbol{q}^T \delta q] \tag{10}$$

from the conversion of the axis-angle representation to quaternions we know

$$\delta\boldsymbol{q} = [\boldsymbol{k} \sin \delta\boldsymbol{\theta}/2; \cos \delta\boldsymbol{\theta}/2] \tag{11}$$

By assumption the error $\delta\boldsymbol{\theta}$ is small, thus small-angle approximations can be used and Eq. (11) can be simplified to:

$$\delta\boldsymbol{q} \simeq [\delta\boldsymbol{\theta}/2; 1] \tag{12}$$

The attitude is thus represented by 3 parameters in the error state, therefore the error state has one element less than the true state and is now a minimal representation opposed to the over-parametrized quaternion. It should be noted that other methods exist to avoid the over-parametrization of the quaternion, however, this is the approach commonly used for ESKFs.

To retrieve the full state from the error state, the errors are simply added to the state estimate, $\hat{\boldsymbol{x}}$, which is obtained from state propagation, and is explained in more detail later in this section.

$$\boldsymbol{x} = \hat{\boldsymbol{x}} + \delta\boldsymbol{x} \tag{13}$$

However, the true quaternion and its relation the the error quaternion and the estimate is not additive, but has to be computed by a quaternion multiplication:

$$\boldsymbol{q} = \delta\boldsymbol{q} \otimes \hat{\boldsymbol{q}} \tag{14}$$

The state estimate uses $\boldsymbol{q}$, while the update (see Sect. 3.4) is using $\boldsymbol{\theta}$, this has to be taken into account when merging the predicted state with the error retrieved from the update (see Sect. 3.4). Here, Eq. (12) can be used.

The following equations describe the evolving state in continuous time:

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \tag{15}$$

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{a}(t) \tag{16}$$

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\boldsymbol{q}(t) \tag{17}$$

$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{bg}(t) \tag{18}$$

$$\dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{ba}(t) \tag{19}$$

where $\boldsymbol{a}(t)$ is the acceleration, $\boldsymbol{n}_{bg}$ and $\boldsymbol{n}_{ba}$ are the noises of the gyroscope and the accelerometer respectively, $\boldsymbol{\omega}$ is the angular rate and $\boldsymbol{\Omega}$ is defined as:

$$\boldsymbol{\Omega}\boldsymbol{\omega} = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{20}$$

$$[\boldsymbol{\omega}\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{21}$$

State propagation is done based on the measurements of an IMU, namely the gyroscope and the accelerometer. The gyroscope measurement is given by

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{\omega}_G + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{22}$$

where $\boldsymbol{R}(\boldsymbol{q})$ is the rotation matrix equivalent to the rotation described by $\boldsymbol{q}$, $\boldsymbol{n}_g$ is a zero mean white Gaussian process noise and $\boldsymbol{\omega}_G$ is the planets rotation. The accelerometer measurement is given by

$$\boldsymbol{a}_m = \boldsymbol{R}(\boldsymbol{q})(\boldsymbol{a} - \boldsymbol{g} + 2[\boldsymbol{\omega}\times]\boldsymbol{v} + [\boldsymbol{\omega}\times]^2\boldsymbol{r}]) + \boldsymbol{b}_a + \boldsymbol{n}_a \tag{23}$$

where $\boldsymbol{g}$ is the gravitational acceleration in the local frame. Note that a is the full acceleration of the body and does therefore includes $\boldsymbol{g}$. Thus, it is necessary to subtract $\boldsymbol{g}$!

The state estimate can be described using the equations below, note that these equations are in continuous time, however, for readability the argument (t) is omitted for in the following

$$\dot{\hat{\boldsymbol{r}}} = \hat{\boldsymbol{v}} \tag{24}$$

$$\dot{\hat{\boldsymbol{v}}} = \boldsymbol{R}(\hat{\boldsymbol{q}})(\boldsymbol{a_m} - \hat{\boldsymbol{b}}_a) - 2[\boldsymbol{\omega}_m\times]\hat{\boldsymbol{v}} - [\boldsymbol{\omega}_m\times]^2\hat{\boldsymbol{r}} + \boldsymbol{g} \tag{25}$$

$$\dot{\hat{\boldsymbol{q}}} = \frac{1}{2}\Omega(\boldsymbol{\omega_m} - \hat{\boldsymbol{b}}_g - \boldsymbol{R}(\boldsymbol{\omega}_G))\hat{\boldsymbol{q}} \tag{26}$$

$$\dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{0} \tag{27}$$

$$\dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{0} \tag{28}$$

The linearised system model for the error state is given by

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}\delta\boldsymbol{x} + \boldsymbol{G}\boldsymbol{n}_{IMU} \tag{29}$$

where the system matrix $\boldsymbol{F}$ and the system noise matrix $\boldsymbol{G}$ are computed based on Eqs. (24)–(28). $\boldsymbol{n}_{IMU}$ is a vector composed of all IMU noises present, $\boldsymbol{n}_a$, $\boldsymbol{n}_{ba}$, $\boldsymbol{n}_g$ and $\boldsymbol{n}_{bg}$. Note, that the ESKF update is performed using the error state and not the full state as in a normal EKF. Therefore, the Jacobian elements $\boldsymbol{F}$ and $\boldsymbol{G}$ have to be calculated with respect to the error state $\delta\boldsymbol{x}$ and not the full state.

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -[\boldsymbol{\omega}_m\times]^2 & -2[\boldsymbol{\omega}_m\times] & -\boldsymbol{R}(\hat{\boldsymbol{q}})(\boldsymbol{a}_m - \hat{\boldsymbol{b}}_a) & \boldsymbol{0} & -\boldsymbol{R}(\hat{\boldsymbol{q}}) \\ \boldsymbol{0} & \boldsymbol{0} & -[\boldsymbol{\omega}_G\times] & -\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{30}$$

The $\boldsymbol{G}$ matrix is

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{R}(\hat{\boldsymbol{q}}) & \boldsymbol{0} \\ -\boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix} \tag{31}$$

Until now, all equations are given in continuous time. However, the IMU measurements are only available at discrete time steps δt. Thus, as for every normal EKF, the state propagation in discrete time (of the full state, the error state is not propagated, as the expected value is zero) is done by numerical integration of Eqs. (24)–(28). In our work we use a Euler integrator for this task. Moreover, the covariance matrix has to be propagated. Note, that this propagation is linked to the error state and not the full state. The propagation of the discrete time $\boldsymbol{P}_{k|k-1}$ is done by numerical integration of the continuous time equation $\dot{\boldsymbol{P}}(t)$ using the previous covariance $\boldsymbol{P}_{k-1|k-1}$ as initial conditions.

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{F}(t)\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{F}^T(t) + \boldsymbol{G}(t)\boldsymbol{Q}(t)\boldsymbol{G}^T(t) \tag{32}$$

The state is propagated with these equations until a measurement is recorded.

3.2 Measurement

To update the predictions based on the IMU, to estimate the biases and to get a more precise estimate of the full state, a measurement has to be recorded.

The measurement is performed by combining the stereo maps as presented in Sect. 2 with a feature matching and tracking algorithm. These features are tracked over time during the descent and are included in the state, as will be explained in Sect. 3.3. From the stereo DEMs measurements from the camera to the selected features are available. To ensure re-observation of features over time, a rather large number of features is extracted, the total number is not limited by the algorithm. The stereo measurements can be translated to the 3-D location of the features in the global reference frame. Matching and extracting the features over multiple images is done using the SURF feature descriptor [2]. Observed features are then added to the state in a SLAM fashion. Namely, they are added to the state and are also propagated in the propagation step. Therefore, positions of re-observed features can be compared to their previously measured position in the update step. It is expected that this will further improve state estimation. Figure 4 shows two consecutive images, taken at different altitudes and the features matched in it.

The discrete time measurement model, as a function of the true state, is described by

$$z_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{33}$$

The expected measurement can be described by the following equation.

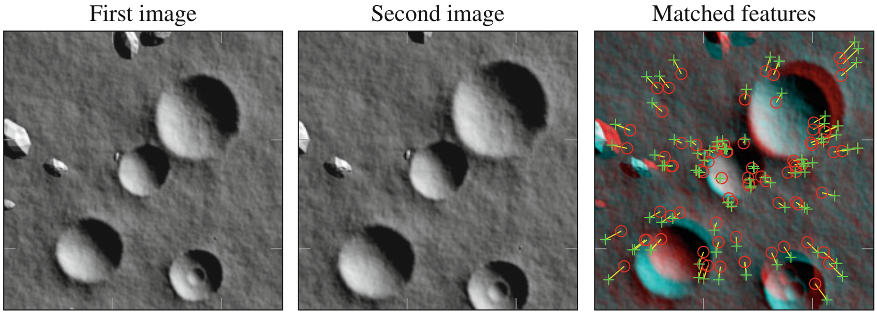


Fig. 4 Matched features over two consecutive images. The first image is taken at 107 m altitude while the second, lower, image is taken at 100 m altitude. The camera is facing directly downwards

$$\hat{z}_k = \underbrace{\begin{bmatrix} 1/res & & \\ & 1/res & \\ & & 1 \end{bmatrix}}_{\text{Real2Mes}} \boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{G},\boldsymbol{k}}^{\hat{\boldsymbol{B}}})(\hat{\boldsymbol{r}}_{g,\text{Feature},k} - \hat{\boldsymbol{r}}_{g,\text{Body},k}) \tag{34}$$

where $\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})$ and $\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{G},\boldsymbol{k}}^{\boldsymbol{B}})$ describe the rotation from the body to the camera frame and from the global to the body frame, respectively. The matrix Real2Mes maps a point from the 3D world on the 2D image plane. The resolution, res, is given by

$$res = \frac{\tan\frac{FOV}{2}(x_{3,k} - \text{radius})}{2 \cdot \text{image size}} \tag{35}$$

The resolution is thus linked to the camera's position above the ground, which can be derived from the third element of the state, the vehicle's z-position. Figure 5 depicts the measurements and the different frames involved. It can be seen that the measurement is recorded in the camera frame, while the camera (and thus the camera frame) is fixed in with respect to the body frame of the spacecraft. The location of the spacecraft (and thus the body frame) is expressed with respect to the global frame. Ultimately, the goal is to express the feature locations measured in the camera frame in the global frame. In this research it is further assumed that the surface of the planet is flat and that the rotation of the body can be neglected. Since only the very last part of a descent is simulated, which only lasts a couple of seconds, this assumption is valid - if the rotational rate of the body is sufficiently small.

Since the update, as discussed in Sect. 3.4, is done using the error state, also the measurement matrix $\boldsymbol{H}$ needs to be computed based on the error state, rather than the nominal state. This means that also the Jacobian, of the measurement function $\boldsymbol{h}$, has to be computed with respect to error state:

$$\boldsymbol{H} \triangleq \frac{\partial \boldsymbol{h}}{\partial \delta \boldsymbol{x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}} = \boldsymbol{H}_x \boldsymbol{X}_{\delta \boldsymbol{x}} \tag{36}$$

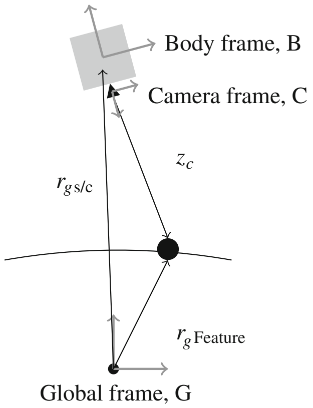


Fig. 5 Frame definitions

Here, the first part $\boldsymbol{H}_x$ is the "normal" Jacobian with respect to the full state, as also used in a standard EKF formulation. And $\boldsymbol{X}_{\delta \boldsymbol{x}}$ is the Jacobian of the state with respect to the error state. As discussed previously, all elements of the error state, apart from the term for the angles, are defined as sums. For example, the position (the first three elements of the state) is defined as $\boldsymbol{r} + \delta \boldsymbol{r}$. Therefore, the Jacobian element of the position with respect to the error in position becomes

$$\boldsymbol{X}_{\delta \boldsymbol{x}, \delta r} = \frac{\partial (\boldsymbol{r} + \delta \boldsymbol{r})}{\partial \delta \boldsymbol{r}} = \mathbf{1} \tag{37}$$

The same holds true for all these derivatives, but $\boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta}$, since the error link to the attitude is not described by a sum, but a quaternion multiplication. The Jacobian $\boldsymbol{X}_{\delta \boldsymbol{x}}$ is

$$\boldsymbol{X}_{\delta \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I}_{6\times 6} & & \\ & \boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta} & \\ & & \boldsymbol{I}_{6+3n \times 6+3n} \end{bmatrix} \tag{38}$$

where n is the number of features in the state and $\boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta}$ is given by:

$$\boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{39}$$

The next step is to compute the Jacobian element $\boldsymbol{H}_x$. For Eq. (34) it is known that the measurement is only a function of the spacecraft position $\boldsymbol{r}$, the feature position $\boldsymbol{r}_{\mathrm{Feature}}$, the resolution and $\boldsymbol{q}$. Therefore, all elements but the Jacobian elements with respect to these state variables, $\boldsymbol{H}_r$, $\boldsymbol{H}_{r,\mathrm{Feature}}$ and $\boldsymbol{H}_q$, are zero. The Jacobian of the measurement function with respect to the spacecraft position is:

$$\boldsymbol{H}_r = \boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}}) \begin{bmatrix} -1/res & 0 & \boldsymbol{r}_1 u \\ 0 & -1/res & \boldsymbol{r}_2 u \\ 0 & 0 & -1 \end{bmatrix} \tag{40}$$

$$u = \delta \frac{1}{res} = \frac{\text{image size} \tan(\text{FOV}/2)}{(\tan(\text{FOV}/2)\hat{x}_3 - \tan(\text{FOV}/2)\text{radius})^2} \tag{41}$$

Next, the Jacobian with respect to the current feature's position is

$$\boldsymbol{H}_{r,F} = \text{Real2Mes} \cdot \boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}}) \tag{42}$$

where the transformation Real2Meas, which projects the coordinates of a feature point from the 3D camera reference frame on the 2D image plane, is given by

$$\text{Real2Mes} = \begin{bmatrix} 1/\text{res} & & \\ & 1/\text{res} & \\ & & 1 \end{bmatrix} \tag{43}$$

And lastly the Jacobian with respect to the quaternion is:

$$\begin{aligned} \boldsymbol{H}_q &= 2 \cdot \text{Real2Mes} \cdot \boldsymbol{R}(\widehat{\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}}})\{\hat{\boldsymbol{q}}(\bar{d}^T)\boldsymbol{I}_3 + \hat{\boldsymbol{q}}(\bar{\boldsymbol{d}})^T - \\ &\quad \bar{\boldsymbol{d}}\hat{\boldsymbol{q}}^T + 2\hat{q}[(\bar{\boldsymbol{d}})\times]\hat{q}(\bar{\boldsymbol{d}}) + [\bar{\boldsymbol{d}}\times]\hat{\boldsymbol{q}}\} \qquad (44) \\ \bar{\boldsymbol{d}} &= \hat{\boldsymbol{r}}_F - \hat{\boldsymbol{r}}_B \qquad (45) \end{aligned}$$

where $\boldsymbol{H}_q$ was derived from the attitude matrix $R(\boldsymbol{q}) = ((q^2 - \boldsymbol{q}^T\boldsymbol{q})\mathbf{1}_3 + 2\boldsymbol{q}\boldsymbol{q}^T - 2q[\boldsymbol{q}\times]$.

3.3 Covariance and State Augmentation

In SLAM a key element is that observed map points, or features, are added to the state. Therefore, after every measurement the state has to be augmented: if a feature/point in space is observed, it is added to the state, i.e., *the state vector is augmented*. This means that the features have to be propagated. However, in our case features are not dynamic and therefore their locations do not change in the propagation step, but only change when updating. Here, it should be noted that this is only the case, because the final phase is so fast that it is assumed that the rotation of the planet is negligible. If a very fast rotating body would be the target or the algorithm would be applied for a longer period, the rotation might have to be accounted for and thus the features would need to be propagated as well.

When the state is augmented, the covariance has to be augmented as well. This is depicted in Fig. 6. Here, it is important to note that the covariances of the added

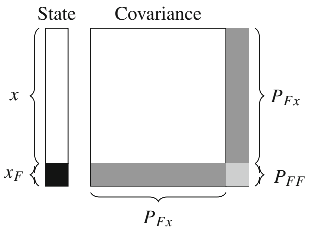


Fig. 6 State and covariance augmentation

features are not independent. This is because feature locations are computed from the measurement, using information stored in the state.

For each measured feature, the feature's position in the global reference frame is appended to the state using the inverse measurement matrix, which is the inverse of the measurement function given in Eq. (34):

$$\boldsymbol{r}_{Feature} = \hat{\boldsymbol{r}}_{Body} + [Real2Mes \cdot \boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}})]^{-1}\boldsymbol{z} \tag{46}$$

From this equation, the Jacobian elements with respect to the spacecraft state and the measurement z are computed, $\boldsymbol{J}_z$, since the inverse measurement function is a function of the state and the measurement. Apart from $\boldsymbol{z}$, the inverse measurement function is only a function of the position and the quaternion, the Jacobian with respect to all other elements are zero. Therefore only $\boldsymbol{J}_z$, $\boldsymbol{J}_{x,P_{\text{body}}}$ and $\boldsymbol{J}_{x,\theta}$ need to be computed.

$$\boldsymbol{J}_z = (\text{Real2Mes}\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}}))^{-1} \tag{47}$$

$$\boldsymbol{J}_{x,P_{\text{body}}} = \boldsymbol{I}_3 \tag{48}$$

The Jacobian $\boldsymbol{J}_{x,\theta}$ is more difficult to compute as the equation is a function of $\boldsymbol{q}$ while the state contains $\delta\boldsymbol{\theta}$. In principle the same approach as for the Jacobian $\boldsymbol{H}_{\delta\boldsymbol{x}}$ is followed, see Eq. (36): Therefore,

$$\boldsymbol{J}_{x,\theta} = \boldsymbol{H}_q \boldsymbol{X}_{\delta\theta} \tag{49}$$

$$\boldsymbol{H}_q = 2\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})^{-1}\left[\hat{\boldsymbol{q}}^T\boldsymbol{z}\boldsymbol{I}_3 + \hat{\boldsymbol{q}}\boldsymbol{z}^T - \boldsymbol{z}\hat{\boldsymbol{q}}^T + 2\hat{q}[\boldsymbol{z}\times]\,\hat{q}\boldsymbol{z} + [\boldsymbol{z}\times]\hat{\boldsymbol{q}}\right] \tag{50}$$

where $[\boldsymbol{q}^T q]$ are the elements of the inverse of $\boldsymbol{q}_{\boldsymbol{G}}^{\boldsymbol{B}}$.

The new augmented covariance can be computed from, see [1]

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{P}_{Fx}^T \\ \boldsymbol{P}_{Fx} & \boldsymbol{P}_{FF} \end{bmatrix} \tag{51}$$

with

$$\boldsymbol{P}_{FF} = \boldsymbol{J}_x \boldsymbol{P}_{xx} J_x^T + \boldsymbol{J}_z \boldsymbol{R} \boldsymbol{J}_z^T \tag{52}$$

$$\boldsymbol{P}_{Fx} = \boldsymbol{J}_x \boldsymbol{P}_{xx} \tag{53}$$

If a feature is not re-observed or falls outside of the area imaged, it will be removed from the state. This means that the related elements will also be deleted from $\boldsymbol{P}$.

3.4 State Update

After state augmentation the state update can then be performed using the following equations:

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k \tag{54}$$

From $\boldsymbol{S}_k$ the Kalman gain can be computed

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k \boldsymbol{S}_k^{-1} \tag{55}$$

Using the Kalman gain, the update of the error state can be computed. This does not involve a prediction of the error state as opposed to the update steps of an EKF.

$$\delta \hat{\boldsymbol{x}}_k = \boldsymbol{K}_k (z_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k)) \tag{56}$$

Knowing the estimate of the error state, the estimate of the full state can be updated.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \delta \hat{\boldsymbol{x}}_k \tag{57}$$

The final step is to update the predicted covariance:

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + (\boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k) \tag{58}$$

After this the filter continues propagation of the state estimate until a new measurement is recorded.

Here, one has to be careful when merging the attitude error with quaternion in the nominal state. The value in the error state is the three-dimensional $\delta\boldsymbol{\theta}$, which has to be translated to the four-dimensional $\delta\boldsymbol{q}$ first. Measurements from other sources could also be included during the update step, for example, from a star tracker or altimeter, however, this is not necessary for the goal of this work. The main purpose of this algorithm is to limit the horizontal errors. Adding, for example, an altimeter to the sensor suit will not increase the accuracies in horizontal directions, but only improve the knowledge of the vertical distance towards the landing site. This information, however, will not decrease the size of the hazard-relative landing ellipse.

4 Analysis of Results

The algorithm presented in the previous sections was implemented and a Monte Carlo analysis was performed to investigate the performance of the algorithm. The simulation set-up as well as the results are presented in this section.

4.1 Set-Up

This research focuses on the final phase of a descent, therefore a trajectory from 120 m down is simulated. This represents the phase during which stereo-vision based HDA is possible, as concluded in [16]. An E-guidance-based trajectory is flown, as E-guidance [4] is a guidance law, which could potentially be used in an HDA scenario due to its ability to select a specific landing site. Guidance laws used in combination with an HDA/TRN system have to be more flexible than current guidance systems. HDA might require a change in the commanded landing location, while it is also necessary to keep the surface/landing site in the field of view of the respective sensors. This results in the need for constraining the spacecraft's orientation.

The scene used has a maximum elevation difference of ≈5 m, while the largest boulder in the scene has a vertical size of ≈2 m. Also multiple smaller features of the size of 0.5–1.5 m are located in the landing region. Figure 4 given in Sect. 2 shows an example of the scenes used.

As discussed in Sect. 1, the system is not capable of reducing the inertial error in the localization, as features are not matched to features with inertially known positions. The flown trajectory is presented in Fig. 7. Both the 3-D trajectory as well as the X, Y and Z components as a function of time are given.

As mentioned in Sect. 2, a 2 m stereo baseline is used, based on a sensitivity study performed during earlier work (see [16]). As an imaging sensor a 512 × 512 pixel camera is assumed. The camera's field of view is 30°. The surface images are generated using PANGU [14], therefore only a simple pinhole camera-model is employed. The IMU is modelled based on the data presented in [8], the used data is summarized in Table 1. The measurements are taken at a frequency of 2 Hz.

Artificially created images do, of course, have some drawbacks. Noise is only present when added manually, and other imaging effects, such as motion blur or distortion, are not present (or have to be modelled artificially). To this end it is always recommended to test algorithms also with images obtained trough real sensors. A possibility to perform such a test is, for example, the Testbed for Robotic Optical Navigation at the German Aerospace Center in Bremen [10]. It is planned for the near future to use this testbed for testing the proposed algorithm.

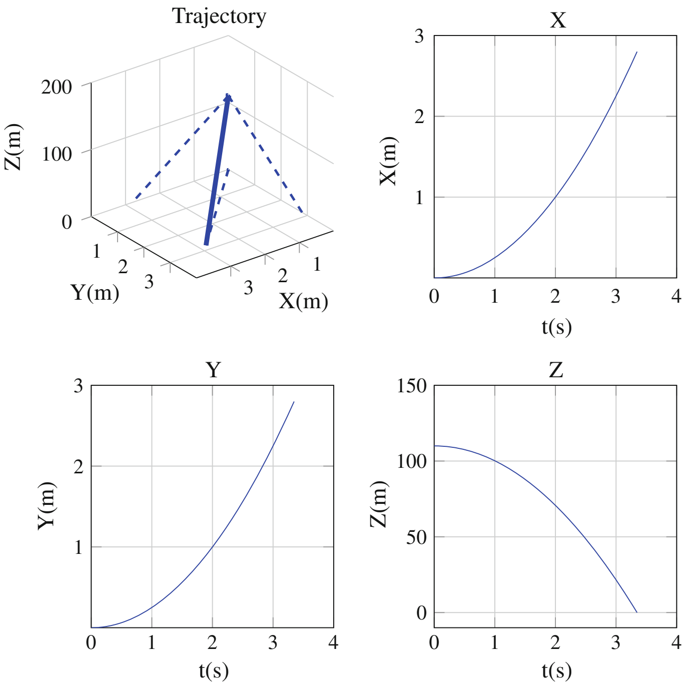


Fig. 7 Nominal trajectory

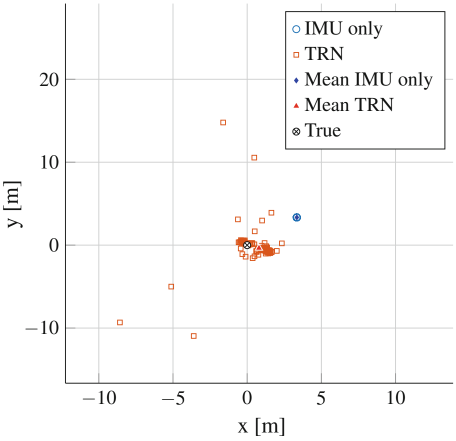
Table 1 IMU parameters based on [8]

Error	Value	Unit
Gyro bias/drift	0.02	deg/h
Gyro scale factor	1.6	PPM
Gyro angle random walk	0.00005	deg/$\sqrt{s}$
Accelerometer bias	30	μg
Accelerometer scale factor	66	PPM
Accelerometer velocity random walk	20	m/s/$\sqrt{s}$

4.2 Results

A Monte Carlo simulation with 200 runs was executed to compare the performance of the algorithm with a pure IMU-propagation system, therefore the random contributions in the IMU change for each run of the algorithm. Moreover, the selected and tracked features change for each run of the algorithm, as they are selected randomly. This serves two purposes: (1) to show that the algorithm is robust and independent of which features are used for tracking and (2) to show that more accurate navigation

Fig. 8 Monte Carlo results



can be performed by tracking features over time, as opposed to pure dead-reckoning as usually performed during the very last phase of the descent.

The results are shown in Fig. 8. Here, it can be concluded that the stereo-based ESKF SLAM is capable of limiting the additional relative error accumulated over the simulated last phase of the descent. Thus, the algorithm is capable of enabling hazard avoidance manoeuvres from a localisation point of view. It can be seen that the TRN results have a larger dispersion than the pure dead reckoning results, however, the results are almost always closer to the truth value than the pure propagation based on the IMU measurements only. However, on a total of 200 runs, 5 extreme outliers are generated when using the TRN system, for which the result is further away from the truth than the dead-reckoning solution. Therefore 97.5% off all runs produce usable results. The few outliers generated are linked to the performance of the vision systems itself, and therefore for future work it is recommended to investigate the feature-extraction and mapping system in more detail.

Currently, between 50 and 100 features are tracked and re-observed. The number of features tracked depends on the number of features that can be matched and gets lower at lower altitudes, as features tend to leave the field of view faster. The number of features tracked is not actively influenced in the algorithm, and simply all features matched are also used for tracking. This is due to the fact that features are not solely used as a measurement, but are fed into the filter to improve the measured feature locations themselves.

Figures 9 and 10 show the results of a single run for the error in position and velocity estimation. From Fig. 9 it can be concluded that the TRN algorithm keeps the error in x and y direction bounded, so the error does not grow as is the case for the pure IMU propagation. In z direction, it is moreover able to not only counteract the error growth, but also to move closer towards the true solution. This is the case as the measurement in z-direction measures the full distance toward the ground, while this is not the case for the x- and y-directions. On the contrary, the velocity error is

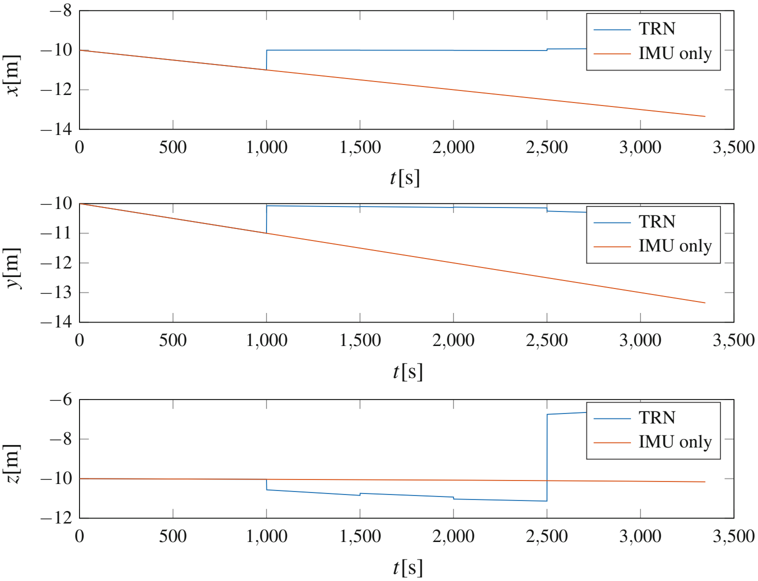


Fig. 9 Spacecraft positions error for a single run

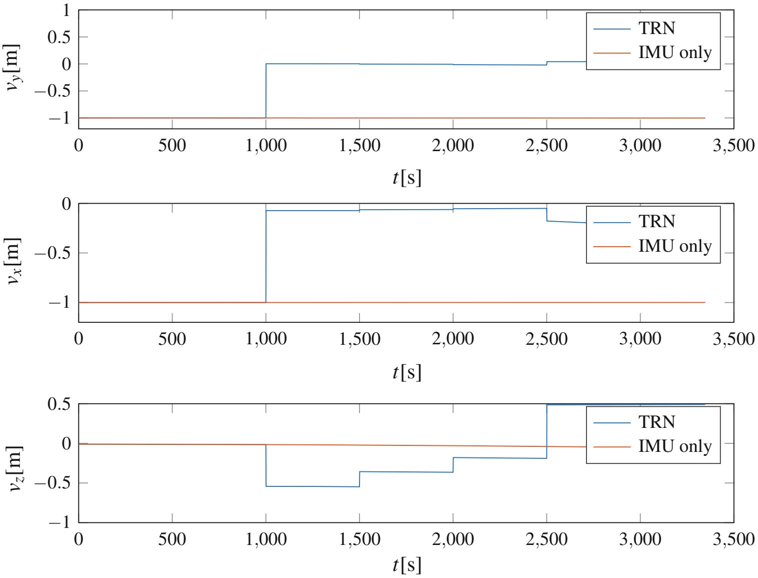


Fig. 10 Spacecraft velocities error for a single run

smaller in x- and y-directions directions than in the z direction, as the velocity can be more accurately estimated in the image plane rather than orthogonal to this plane.

Concluding, these two figures show that the algorithm does perform as expected, however, it can also be seen that no convergence is reached. This comes as no surprise as only 5 measurements are used inside the filter. So few measurements can not lead to a filter convergence.

5 Conclusions and Recommendations

We have shown that it is possible to use stereo-based hazard maps as an input for terrain relative navigation and thereby limit the relative landing error in the final phase of a planetary landing. The method is robust and produces less than 3% outliers. The few outliers encountered are linked to the vision system and not to the filter itself.

This system can therefore enable hazard-relative navigation when touching down in terrains, which are not inherently safe. Moreover, this would reduce the landing ellipse size.

Recommendations

To further investigate the outlier behaviour as presented and discussed in Sect. 4 a detailed analysis of an outlier case will be performed. Furthermore, it is advisable to compare the performance of the stereo-vision based filter with a pure monocular vision-system.

References

1. Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (slam): part ii. IEEE Robot Automat Mag 13(3):108–117
2. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: European conference on computer vision, Springer, Berlin, pp 404–417
3. Chen A, Mischna M (2015) Edl landing site engineering assessment. In: Mars 2020 landing site workshop 2, NASA/JPL, California, pp 1184–1186
4. Cherry GW (1964) E Guidance- a general explicit optimizing guidance law for rocket-propelled spacecraft. MIT Instrumentation Laboratory, Cambridge
5. Crane ES, Rock SM (2012) Influence of trajectory on accuracy of hazard estimation during lunar landing. In: AIAA guidance, navigation, and control conference. AIAA, Minneapolis, MN
6. Delaune J, Le Besnerais G, Sanfourche M, Plyer A, Farges JL, Bourdarias C, Voirin T, Piquereau A (2011) Camera-aided inertial navigation for pinpoint planetary landing on rugged terrains. In: International planetary probe workshop
7. Frapard B, Polle B, Flandin G, Bernard P, Vétel C, Sembely X, Mancuso S (2003) Navigation for planetary approach and landing. Eur Space Agency Publ 516:159–168

8. Geller DK, Christensen D (2009) Linear covariance analysis for powered lunar descent and landing. J Spacecr Rocket 46(6):1231–1248
9. Kerl C, Stuckler J, Cremers D (2015) Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In: Proceedings of the IEEE international conference on computer vision, pp 2264–2272
10. Krüger H, Theil S (2010) Tron-hardware-in-the-loop test facility for lunar descent and landing optical navigation. IFAC Proc Vol 43(15):265–270
11. Madyastha VK, Ravindra VC, Mallikarjunan S, Goyal A (2011) Extended kalman filter vs. error state kalman filter for aircraft attitude estimation. In: AIAA GNC
12. Mourikis AI, Trawny N, Roumeliotis SI, Johnson AE, Ansar A, Matthies L (2009) Vision-aided inertial navigation for spacecraft entry, descent, and landing. IEEE Trans Robot 25(2):264–280
13. Mur-Artal R, Tardos JD (2016) Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. arXiv:1610.06475
14. Parkes S, Martin I, Dunstan M, Matthews D (2004) Planet surface simulation with pangu. In: Eighth international conference on space operations, pp 1–10
15. Verweld MJ (2013) Relative optical navigation for a lunar lander mission. In: Advances in aerospace guidance, navigation and control. Springer, Berlin, pp 661–679
16. Woicke S, Mooij E (2016) A stereo-vision hazard-detection algorithm to increase planetary lander autonomy. Acta Astronautica 122:42–62
17. Woicke S, Mooij E (2016) Passive hazard detection for planetary landing. In: AIAA guidance, navigation, and control conference, p 1133