International Journal of Control, Automation, and Systems (2015) 13(2):361-370 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-013-0416-y http://www.springer.com/12555

Generalized Nonlinear Optimal Predictive Control using Iterative State-space Trajectories: Applications to Autonomous Flight of UAVs

Marina H. Murillo*, Alejandro C. Limache, Pablo S. Rojas Fredini, and Leonardo L. Giovanini

Abstract: Model Predictive Control (MPC) is a modern technique that, nowadays, encapsulates different optimal control techniques. For the case of non-linear dynamics, many possible variants can be developed which can lead to new control algorithms. In this manuscript a novel generic control system method is presented. This method can be applied to control, in an optimal way, different systems having non-linear dynamics. Particularly, in this paper, the proposed technique is presented in the context of developing a control system for autonomous flight of UAVs. This technique can be used for different types of aerial vehicles having any type of generic non-linear dynamics. The presented method is based on the use of iteratively defined optimal candidate state-space trajectories in global state-space. The method uses a generalized linearization process which, opposite to standard methods, does not need to be predefined in a certain equilibrium state but instead it is performed along any arbitrary state. The technique allows the inclusion of constraints with ease. The presented technique is used as a centralized control system unit that is able to control the full aircraft dynamics without the need of decoupling the system in different reduced modes. The technique is tested by making a Cessna 172 airplane model to perform the following autonomous unmanned maneuvers: climbing at constant speed to a desired altitude, heading change to a desired flight direction, and, coordinate turn.

Keywords: Non-linear model predictive control, optimization, state-space trajectories prediction, unmanned aerial vehicles.

1. INTRODUCTION

In the last decades, navigation and control techniques such as feedback control [1-3] have led to modern control techniques such as model predictive control (MPC) [4-7] and others [8-11]. With MPC one can work with multi-variable discrete state space models [12]. MPC allows the treatment of physical systems constraints and fits well in the case of aircraft dynamics [13-16] where control inputs as well as states variables must meet certain limits.

MPC is formulated by solving an on-line optimal control problem [17,18]. Control inputs are determined with a refresh rate equal to a given discretization period ΔT_s. The optimal control sequence valid for a given time control horizon t^{h_c} is calculated by minimizing an objective function subject to state and inputs constraints during a prediction time horizon t^{h_p}. The control sequence is assumed to remain constant for $t^{h_p} \geq t \geq t^{h_c}$.

The obtained optimal control input corresponding to the first sampling instant is then applied to the system and the calculation is restarted displacing the prediction horizon forward to the next sampling instant ($t^p \leftarrow t^p + \Delta T_s$). The computation of new predictions at each sampling rate compensate for unmeasured disturbances. A review of MPC can be found in [19,20].

This paper presents a novel non-linear optimal predictive control algorithm that can be used for general non-linear system dynamics [21]. The method uses an iterative state-space trajectories linearization process to find the optimal discrete control inputs. The mathematical algorithm is presented in the application context of controlling unmanned aerial vehicles. Both, the automatic control operation and the non-linear predictive control method are presented. The presented technique, named *Iterative Non-linear Model Predictive Control* (INL-MPC), is used as an automatic control system unit which allows an unmanned aircraft to fly and perform different autonomous maneuvers, such as, climbing, descend, change of heading and coordinated turns, in an optimal way. For this specific application, the UAV state-space representation consists of twelve differential equations that describe the behavior and dynamics of a complete six degrees of freedom (6-DOF) aircraft [2]. In Section 2, the non-linear aircraft model is presented. In Section 3, the theoretical linearization approach on generalized state-space trajectories is presented. In Section 4, the proposed non-linear MPC

Manuscript received September 4, 2013; revised June 6, 2014; accepted July 21, 2014. Recommended by Associate Editor Chang Kyung Ryoo under the direction of Editor PooGyeon Park.

Marina H. Murillo, Alejandro C. Limache, and Pablo S. Rojas Fredini are with the Research Center of Computational Methods (CIMEC), CONICET, Santa Fe, Argentina (e-mails: marinah murillo@gmail.com, alejandrolimache@gmail.com, srojasfredini @santafe-conicet.gov.ar)

Leonardo L. Giovanini is with the Research Center for Signals, Systems and Computational Intelligence (SINC), FICH, Universidad Nacional del Litoral, Santa Fe, Argentina (e-mail: llgiovanini@googlemail.com).

* Corresponding author.

method is described. In Section 5, a group of selected automatic control maneuvers are presented using a Cessna 172 aircraft model as the unmanned vehicle. The following maneuvers are tested: 1) Climbing at constant speed, 2) Change of heading direction of the flight path maintaining constant speed and altitude and 3) Coordinated turn at constant speed and altitude.

The proposed control system has been implemented in the flight simulator *Excalibur* for its evaluation. This simulator has been developed at the Research Center of Computational Methods (CIMEC) [22]. The use of the flight simulator facilitates the testing of different control systems and it enables to verify that the desired maneuvers are performed in real time. The control module is implemented as an independent subsystem.

2. NON-LINEAR SYSTEM DYNAMICS

A general representation of the non-linear dynamics of an arbitrary system is given by the following first-order differential equation:

$$\dot{\mathbf{x}} = \bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}), \tag{1}$$

where $\mathbf{x}$ is the vector of system's states, $\mathbf{u}$ is the vector of system's inputs or controls, $\dot{\mathbf{x}}$ is the vector of state rates and $\bar{\mathbf{f}}$ is a vector function that depends on the system being modeled.

2.1. Non-linear aircraft dynamics

In the context of applications to developing control systems for autonomous flight of unmanned aircrafts, the full order aircraft system is represented by the following state vector:

$$\begin{aligned}\mathbf{x} &= [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}]^T \\ &= [v_t\ \alpha\ \beta\ \phi\ \theta\ \psi\ p\ q\ r\ x_{\mathrm{N}}\ y_{\mathrm{E}}\ h]^T\end{aligned} \tag{2}$$

of dimension $N_s = 12$ whose components are given by:

- v_t, α and β are the true airspeed, the angle of attack and the sideslip angle, respectively.
- ϕ, θ and ψ are the Euler angles that define the roll, pitch and yaw orientation of the Body frame with respect to the North-East-Down (NED) reference frame.
- p, q and r are the components of the aircraft's angular velocity vector in Body axes ($\hat{\mathbf{x}}_{\mathrm{Body}}$, $\hat{\mathbf{y}}_{\mathrm{Body}}$, $\hat{\mathbf{z}}_{\mathrm{Body}}$), respectively.
- x_{N}, y_{E} and $z_{\mathrm{D}} = -h$ are the components of the aircraft's CG position with respect to the NED reference frame.

The vector $\mathbf{u}$ is assumed here to be given by the following $N_i = 4$ aircraft control variables:

$$\mathbf{u} = [u_1\ \ u_2\ \ u_3\ \ u_4]^T = [thtl\ \ \delta_e\ \ \delta_a\ \ \delta_r]^T, \tag{3}$$

where δ_e is the elevator deflection in degrees, δ_a is the aileron deflection in degrees, δ_r is the rudder deflection in degrees and $thtl$ is the position of the throttle column whose value is between zero and one.

Following [2], in the case of arbitrary aerial vehicles, the vector function $\bar{\mathbf{f}}(\mathbf{x}, \mathbf{u})$ is given by:

$$\bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} (u\dot{u} + v\dot{v} + w\dot{w})/v_t \\ (u\dot{w} - w\dot{u})/(u^2 + w^2) \\ (\dot{v}v_t - v\dot{v}_t)/(v_t^2 \cos\beta) \\ p + \tan\theta(q\sin\phi + r\cos\phi) \\ q\cos\phi - r\sin\phi \\ (q\sin\phi + r\cos\phi)/\cos\theta \\ (c_1 r + c_2 p)q + c_3 M_x + c_4 M_z \\ c_5 pr - c_6(p^2 - r^2) + c_7 M_y \\ (c_8 p - c_2 r)q + c_4 M_x + c_9 M_z \\ ub_{11} + vb_{21} + wb_{31} \\ ub_{12} + vb_{22} + wb_{32} \\ -ub_{13} - vb_{23} - wb_{33} \end{bmatrix}, \tag{4}$$

where

- b_{ij} are the components of the rotation matrix B_B defined in page 37, Eq. (1.4-10) of [2]
- c_i are coefficients related to the aircraft's inertia moments and are defined in page 80, Eq. (2.4-6) of [2]
- $\dot{u} = rv - qw - g\sin\theta + F_x/m$
- $\dot{v} = -ru + pw + g\sin\phi\cos\theta + F_y/m$
- $\dot{w} = qu - pv + g\cos\phi\cos\theta + F_z/m$
- $\dot{v}_t = (u\dot{u} + v\dot{v} + w\dot{w})/v_t$
- $F_x = F_x(\mathbf{x}, \mathbf{u})$, $F_y = F_y(\mathbf{x}, \mathbf{u})$, $F_z = F_z(\mathbf{x}, \mathbf{u})$ are the total forces (aerodynamic plus propulsion forces) acting on the UAV expressed in Body Frame coordinates. These forces explicitly depend on the state and control variables **x**, **u**.
- $M_x = M_x(\mathbf{x}, \mathbf{u})$, $M_y = M_y(\mathbf{x}, \mathbf{u})$, $M_z = M_z(\mathbf{x}, \mathbf{u})$ are the total moments (aerodynamic plus propulsion moments) with respect to the CG position acting on the UAV, expressed in the Body Frame system. These moments also explicitly depend on the state and control variables.

The aerodynamic and propulsion forces and moments depend on the unmanned aircraft being modeled.

2.2. Predicted state-space trajectories

Now, consider the airplane at certain time t^0, and assume it is in an arbitrary state $(\mathbf{x}^0, \mathbf{u}^0)$. Now assume that the airplane is going to be controlled by a series of h_p consecutive piecewise constant controls (See Fig. 1)

$$\mathbf{U}_p = \left[\mathbf{u}^1 \mathbf{u}^2 \cdots \mathbf{u}^k \cdots \mathbf{u}^{h_p}\right]^T \tag{5}$$

during a time period $[t^0, t^{h_p}]$. Each control $\mathbf{u}^k$ in (5) is applied during a time interval $[t^{k-1}, t^k]$ of length ΔT_s. The discrete times are defined as $t^k = t^0 + k\Delta T_s$.

Notation: From now on superscripts to denote the discrete values of variables at different discrete times t^k will be used. For example, the value of variable $y(t)$ at time t^k will be denoted by y^k.

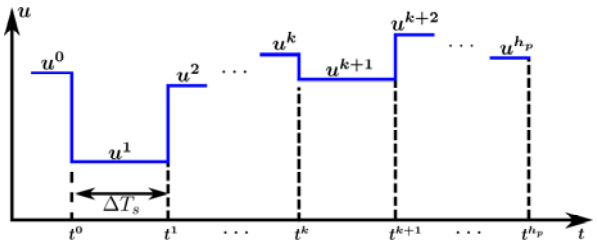


Fig. 1. Time discretization scheme.

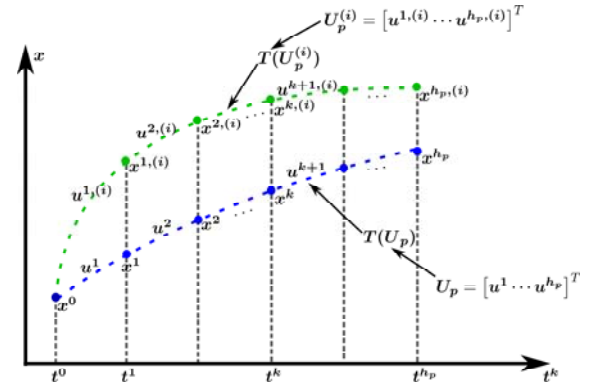


Fig. 2. Predicted state-space trajectories.

With the airplane at the initial state $\mathbf{x}^0$ and given the first control input $\mathbf{U}_p(1)=\mathbf{u}^1$, (1) can be integrated using for example a Runge-Kutta integration algorithm, to find the airplane's new state $\mathbf{x}^1$ at time t^1. Iteratively, the subsequent states $\mathbf{x}^k$ associated to the control sequence $\mathbf{U}_p$ defined in (5) can be determined to find the vector of predicted states:

$$\widehat{\mathbf{X}}=\left[\mathbf{x}^1\mathbf{x}^2\cdots\mathbf{x}^k\cdots\mathbf{x}^{h_p}\right]^T. \tag{6}$$

The above defines an airplane state-space trajectory $T(\mathbf{U}_p)$, as shown in Fig. 2. Of course, a different control sequence $\mathbf{U}_p^{(i)}$:

$$\mathbf{U}_p^{(i)}=\left[\mathbf{u}^{1,(i)}\mathbf{u}^{2,(i)}\cdots\mathbf{u}^{k,(i)}\cdots\mathbf{u}^{h_p,(i)}\right]^T \tag{7}$$

will generate a different state-space trajectory $T(\mathbf{U}_p^{(i)})$ with its corresponding vector of predicted states $\widehat{\mathbf{X}}^{(i)}$:

$$\widehat{\mathbf{X}}^{(i)}=\left[\mathbf{x}^{1,(i)}\mathbf{x}^{2,(i)}\cdots\mathbf{x}^{k,(i)}\cdots\mathbf{x}^{h_p,(i)}\right]^T. \tag{8}$$

This is also shown in Fig. 2.

3. GENERALIZED LINEARIZATION, CONSTRAINTS AND TIME-DISCRETIZATION

3.1. Linearization

The non-linear state equation (1) can be linearized by expanding $\bar{\mathbf{f}}(\mathbf{x},\mathbf{u})$ into first order (or into a higher order [23]) Taylor series around an arbitrary operating point $(\mathbf{x}_e,\mathbf{u}_e)$ ([2,17]):

$$\dot{\mathbf{x}}\approx\bar{\mathbf{f}}(\mathbf{x}_e,\mathbf{u}_e)+\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{x}}|_{(\mathbf{x}_e,\mathbf{u}_e)}(\mathbf{x}-\mathbf{x}_e)+\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{u}}|_{(\mathbf{x}_e,\mathbf{u}_e)}(\mathbf{u}-\mathbf{u}_e), \tag{9}$$

where $\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{x}}$ and $\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{u}}$ are the system's Jacobian matrices evaluated at $(\mathbf{x}_e,\mathbf{u}_e)$.

The linearization point $(\mathbf{x}_e,\mathbf{u}_e)$ is an arbitrary point, physically realizable, defined by a flying state $\mathbf{x}_e$ in which the position of the controls are specified by the control vector $\mathbf{u}_e$. Note that $\bar{\mathbf{f}}(\mathbf{x}_e,\mathbf{u}_e)$ can be calculated for every $(\mathbf{x}_e,\mathbf{u}_e)$ using (4) and can take nonzero values. Then, if

$$\begin{aligned}&\mathbf{A}=\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{x}}|_{(\mathbf{x}_e,\mathbf{u}_e)},\quad \mathbf{B}=\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{u}}|_{(\mathbf{x}_e,\mathbf{u}_e)},\\&\mathbf{d}=\bar{\mathbf{f}}(\mathbf{x}_e,\mathbf{u}_e)-\mathbf{A}\mathbf{x}_e-\mathbf{B}\mathbf{u}_e\end{aligned} \tag{10}$$

are defined, equation (9) can be written in compact form as

$$\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t)+\mathbf{d}. \tag{11}$$

The general solution of (11) is

$$\begin{aligned}\mathbf{x}(t)=e^{\mathbf{A}(t-t_e)}\mathbf{x}(t_e)+\int_{t_e}^{t}e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau\\+\int_{t_e}^{t}e^{\mathbf{A}(t-\tau)}\mathbf{d}d\tau.\end{aligned} \tag{12}$$

3.2. Linearization along a predicted state-space trajectory

As can be seen in subsection 2.2, different control inputs sequences define different state-space trajectories. Then, assume that, at time t^0, the airplane is at a known state $(\mathbf{x}^0,\mathbf{u}^0)$ and that during the prediction horizon $[t^0,t^{h_p}]$ it will closely move along a predicted state-space trajectory $T(\mathbf{U}_p^{(i)})$ which is defined by the vector of control inputs $\mathbf{U}_p^{(i)}$ and predicted states $\widehat{\mathbf{X}}^{(i)}$ given in (7) and (8). Now, proceed to linearize (1), in each time-interval $[t^k,t^{k+1}]$, using as linearization points $(\mathbf{x}_e,\mathbf{u}_e)$, the states and control inputs $(\mathbf{x}^{k,(i)},\mathbf{u}^{k,(i)})$ associated to the predicted state-space trajectory $T(\mathbf{U}_p^{(i)})$. Then, for each time interval $[t^k,t^{k+1}]$, the linearized matrices given in (10) become:

$$\begin{aligned}&\mathbf{A}^k=\mathbf{A}^{k,(i)}=\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{x}}|_{\mathbf{x}^{k,(i)},\mathbf{u}^{k,(i)}},\\&\mathbf{B}^k=\mathbf{B}^{k,(i)}=\frac{\partial\bar{\mathbf{f}}}{\partial\mathbf{u}}|_{\mathbf{x}^{k,(i)},\mathbf{u}^{k,(i)}},\\&\mathbf{d}^k=\mathbf{d}^{k,(i)}=\bar{\mathbf{f}}(\mathbf{x}^{k,(i)},\mathbf{u}^{k,(i)})-\mathbf{A}^k\mathbf{x}^{k,(i)}-\mathbf{B}^k\mathbf{u}^{k,(i)}.\end{aligned} \tag{13}$$

Now, these matrices can be used in (12) to predict the state $\mathbf{x}^{k+1}$ at time t^{k+1} given the state $\mathbf{x}^k$ at time t^k and the applied constant control $\mathbf{u}^{k+1}$ (see Fig. 1), then:

$$\mathbf{x}^{k+1}=\widetilde{\mathbf{A}}^k\mathbf{x}^k+\widetilde{\mathbf{B}}^k\mathbf{u}^{k+1}+\widetilde{\mathbf{G}}^k\mathbf{d}^k, \tag{14}$$

where

$$\widetilde{\mathbf{A}}^k=e^{\mathbf{A}^k\Delta T_s},\quad \widetilde{\mathbf{G}}^k=\int_0^{\Delta T_s}e^{\mathbf{A}^k\tau}d\tau,\quad \widetilde{\mathbf{B}}^k=\widetilde{\mathbf{G}}^k\mathbf{B}^k \tag{15}$$

are the discretized versions of the linear system matrices (13). Equation (14) becomes the discrete state-space representation of the linearized airplane model.

Iterating the model described by (14) along the consecutive times t^k with $k = 0,1,\ldots,h_p - 1$, it can be seen that the vector of predicted states, defined in (6), can be calculated as:

$$\widehat{\mathbf{X}} = \mathbf{P}\mathbf{x}^0 + \mathbf{H}_u\mathbf{U}_p + \mathbf{H}_g\mathbf{D}. \tag{16}$$

Matrices **D**, **P**, $\mathbf{H}_u$ and $\mathbf{H}_g$ are defined in the Appendix.

3.3. Constraints

In discrete time, the constraints in control inputs, control inputs rates and states can be expressed as follows:

$$\mathbf{u}_m^k \le \mathbf{u}^k \le \mathbf{u}_M^k \qquad k = 1,\cdots,h_p, \tag{17}$$

$$\dot{\mathbf{u}}_m^k \le \dot{\mathbf{u}}^k \le \dot{\mathbf{u}}_M^k \qquad k = 1,\cdots,h_p, \tag{18}$$

and

$$\mathbf{x}_m^k \le \mathbf{x}^k \le \mathbf{x}_M^k \qquad k = 1,\cdots,h_p, \tag{19}$$

where $(\mathbf{u}_m^k, \mathbf{u}_M^k)$ represent the min-max constraints on the controls, $(\dot{\mathbf{u}}_m^k, \dot{\mathbf{u}}_M^k)$ represent the min-max constraints on the controls rates and $(\mathbf{x}_m^k, \mathbf{x}_M^k)$ represent the min-max constraints on the states.

Equation (17) can be written in matrix form as:

$$\begin{bmatrix} \mathbf{I}_p \\ -\mathbf{I}_p \end{bmatrix} \mathbf{U}_p \le \begin{bmatrix} \mathbf{U}_M \\ -\mathbf{U}_m \end{bmatrix}, \tag{20}$$

where $\mathbf{U}_M$ and $\mathbf{U}_m$ are vectors that contain the upper and lower limits of the control inputs respectively and $\mathbf{I}_p$ is an identity matrix of dimension $(N_i \times h_p) \times (N_i \times h_p)$.

Similarly, evaluating for $k = 1,\cdots,h_p$, equation (18) can be written as follows:

$$\dot{\mathbf{U}}_m \le \frac{\Delta\mathbf{U}_p}{\Delta T_s} \le \dot{\mathbf{U}}_M, \tag{21}$$

where $\dot{\mathbf{U}}_M$ and $\dot{\mathbf{U}}_m$ are vectors that contain the upper and lower limits of the control inputs rates, respectively, and where:

$$\Delta\mathbf{U}_p = \mathbf{E}\mathbf{U}_p + \mathbf{U}_0, \tag{22}$$

where **E** and $\mathbf{U}_0$ are defined in the Appendix.

The control inputs rates constraints (21) can be expressed in matrix form as follows:

$$\begin{bmatrix} \mathbf{E} \\ -\mathbf{E} \end{bmatrix} \mathbf{U}_p \le \begin{bmatrix} \dot{\mathbf{U}}_M \\ -\dot{\mathbf{U}}_m \end{bmatrix} \Delta T_s + \begin{bmatrix} -\mathbf{U}_0 \\ \mathbf{U}_0 \end{bmatrix}. \tag{23}$$

Using (14), the states constraints (19) become:

$$\begin{bmatrix} \mathbf{H}_u \\ -\mathbf{H}_u \end{bmatrix} \mathbf{U}_p \le \begin{bmatrix} \mathbf{X}_M \\ -\mathbf{X}_m \end{bmatrix} + \begin{bmatrix} -\mathbf{P} \\ \mathbf{P} \end{bmatrix} \mathbf{x}^0 + \begin{bmatrix} -\mathbf{H}_g \\ \mathbf{H}_g \end{bmatrix} \mathbf{D} \tag{24}$$

where $\mathbf{X}_M$ and $\mathbf{X}_m$ are vectors that contain the upper and lower limits of the states, respectively.

Finally, putting together (20), (23) and (24), the restrictions can be grouped as

$$\mathbf{A}_{\text{ineq}}\mathbf{U}_p \le \mathbf{b}_{\text{ineq}}. \tag{25}$$

Matrices $\mathbf{A}_{\text{ineq}}$ and $\mathbf{b}_{\text{ineq}}$ are defined in the Appendix.

4. ITERATIVE NON-LINEAR MODEL PREDICTIVE CONTROL (INL-MPC)

The proposed INL-MPC algorithm is a general algorithm extending classical MPC capabilities in order to control non-linear systems. The proposed method reduces the problem of non-linear optimal control of a non-linear system to a series of iterative, *easily solvable*, quadratic optimization suproblems (later defined in (40)). In order to achieve this, it uses a generalized linearization process which is performed along iteratively defined candidate state-space trajectories in global state-space, while classical MPC generally uses a linearized model in one or more certain equilibrium states. The presented method does not require the definition of equilibrium states. The INL-MPC method can be seen as an iterative succession of time-varying linear MPC methods.

4.1. Quadratic objective function

MPC uses the prediction of future behavior of a system in order to determine a sequence of optimal control inputs which minimize an objective function or cost function $J(\mathbf{x}, \mathbf{u})$. The cost function J penalizes the errors between predicted states **x** and desired states $\mathbf{x}_{sp}$. It can also penalize changes in control inputs **u**.

A quite general continuous-time form of the objective function J can be written as:

$$\begin{aligned} J(\mathbf{x},\mathbf{u}) = \int_{t^0}^{t^{h_p}} \Big[& (\mathbf{x}_{sp} - \mathbf{x})^T \mathbf{Q}_x (\mathbf{x}_{sp} - \mathbf{x}) \\ & + (\dot{\mathbf{x}}_{sp} - \dot{\mathbf{x}})^T \mathbf{Q}_{\dot{x}} (\dot{\mathbf{x}}_{sp} - \dot{\mathbf{x}}) + (\mathbf{u}_{sp} - \mathbf{u})^T \mathbf{R}_u (\mathbf{u}_{sp} - \mathbf{u}) \\ & + (\dot{\mathbf{u}}_{sp} - \dot{\mathbf{u}})^T \mathbf{R}_{\dot{u}} (\dot{\mathbf{u}}_{sp} - \dot{\mathbf{u}}) \Big] dt, \end{aligned} \tag{26}$$

where $\mathbf{Q}_x$ and $\mathbf{Q}_{\dot{x}}$ are positive semidefinite matrices and $\mathbf{R}_u$ and $\mathbf{R}_{\dot{u}}$ are positive definite matrices and $\mathbf{x}_{sp}$, $\dot{\mathbf{x}}_{sp}$, $\mathbf{u}_{sp}$ and $\dot{\mathbf{u}}_{sp}$ specify the setpoints of the states variables, states variables rates, control inputs and control inputs rates respectively, and are defined as follows:

$$\mathbf{x}_{sp} = [v_{t_{sp}}\ \alpha_{sp}\ \beta_{sp}\ \phi_{sp}\ \theta_{sp}\ \psi_{sp}\ p_{sp}\ q_{sp}\ r_{sp}\ x_{N_{sp}}\ y_{E_{sp}}\ h_{sp}]^T, \tag{27}$$

$$\dot{\mathbf{x}}_{sp} = [\dot{v}_{t_{sp}}\ \dot{\alpha}_{sp}\ \dot{\beta}_{sp}\ \dot{\phi}_{sp}\ \dot{\theta}_{sp}\ \dot{\psi}_{sp}\ \dot{p}_{sp}\ \dot{q}_{sp}\ \dot{r}_{sp}\ \dot{x}_{N_{sp}}\ \dot{y}_{E_{sp}}\ \dot{h}_{sp}]^T, \tag{28}$$

$$\mathbf{u}_{sp} = [thtl_{sp}\ \delta_{e_{sp}}\ \delta_{a_{sp}}\ \delta_{r_{sp}}]^T, \tag{29}$$

$$\dot{\mathbf{u}}_{sp} = [\dot{thtl}_{sp}\ \dot{\delta}_{e_{sp}}\ \dot{\delta}_{a_{sp}}\ \dot{\delta}_{r_{sp}}]^T. \tag{30}$$

In this work, it will be enough to consider a reduced cost function penalizing states deviations and rate-changes in control inputs, only. As a consequence, matrices $\mathbf{Q}_{\dot{x}}$ and $\mathbf{R}_u$ are assumed to be zero. Additionally, control inputs with minimum motion are desired, so $\dot{\mathbf{u}}_{sp}$ is set equal zero. With these assumptions, the discrete form of (26) becomes:

$$J=\sum_{k=1}^{h_p}[(\mathbf{x}_{sp}^k-\mathbf{x}^k)^T\tilde{\mathbf{Q}}_x^k(\mathbf{x}_{sp}^k-\mathbf{x}^k) +(\mathbf{u}^k-\mathbf{u}^{k-1})^T\tilde{\mathbf{R}}_{\dot{u}}^k(\mathbf{u}^k-\mathbf{u}^{k-1})], \tag{31}$$

where

$$\tilde{\mathbf{Q}}_x^k=\Delta T_s\mathbf{Q}_x^k \quad \text{and} \quad \tilde{\mathbf{R}}_{\dot{u}}^k=\frac{\mathbf{R}_{\dot{u}}^k}{\Delta T_s}. \tag{32}$$

Using (22), (31) can be written in matrix form as follows:

$$J(\hat{\mathbf{X}},\mathbf{U}_p)=(\mathbf{X}_{sp}-\hat{\mathbf{X}})^T\breve{\mathbf{Q}}_x(\mathbf{X}_{sp}-\hat{\mathbf{X}})+ +(\mathbf{E}\mathbf{U}_p+\mathbf{U}_0)^T\breve{\mathbf{R}}_{\dot{u}}(\mathbf{E}\mathbf{U}_p+\mathbf{U}_0), \tag{33}$$

where $\breve{\mathbf{Q}}_x$, $\breve{\mathbf{R}}_{\dot{u}}$ and $\mathbf{X}_{sp}$ are defined in the Appendix.

Using (16) into (33) and discarding constant terms, the objective function (33) can be written as a function of the control sequence only, as follows:

$$J(\mathbf{U}_p)=\mathbf{g}^T\mathbf{U}_p+\mathbf{U}_p^T\mathbf{H}\mathbf{U}_p, \tag{34}$$

where

$$\mathbf{g}=2\left[\mathbf{H}_u^T\breve{\mathbf{Q}}_x(\mathbf{P}\mathbf{x}^0+\mathbf{H}_g\mathbf{D}-\mathbf{X}_{sp})+\mathbf{E}^T\breve{\mathbf{R}}_{\dot{u}}\mathbf{U}_0\right] \tag{35}$$

and

$$\mathbf{H}=\mathbf{H}_u^T\breve{\mathbf{Q}}_x\mathbf{H}_u+\mathbf{E}^T\breve{\mathbf{R}}_{\dot{u}}\mathbf{E}. \tag{36}$$

From (34), it can be seen that the control problem has been reduced to the one of finding an optimal control sequence $\mathbf{U}_p^*$ that minimizes a quadratic objective function subject to the constraints defined in (25).

4.2. How MPC works

In MPC, the control values are assumed to remain constant once the time control horizon t^{h_c} has been reached. This means that the values $\mathbf{U}_p(k)$ of the control sequence $\mathbf{U}_p$ remain equal to $\mathbf{u}^{h_c}$ for $k=h_c+1, \cdots, h_p$. As a consequence, it is possible to express the complete control sequence $\mathbf{U}_p$ in terms of the effective control sequence $\mathbf{U}_c$ as follows:

$$\mathbf{U}_p=\breve{\mathbf{T}}\mathbf{U}_c, \tag{37}$$

where the effective control sequence vector is given by

$$\mathbf{U}_c=\left[\mathbf{u}^1\,\mathbf{u}^2\cdots\mathbf{u}^{h_c}\right]^T \tag{38}$$

and where matrix $\breve{\mathbf{T}}$ is given by

$$\breve{\mathbf{T}}=\begin{bmatrix}\mathbf{I}_c\\ \mathbf{V}\end{bmatrix}, \tag{39}$$

where matrices $\mathbf{I}_c$ and $\mathbf{V}$ are defined in the Appendix.

Note that, while $\mathbf{U}_p$ is a vector of dimension $h_p\times N_i$ containing the complete control sequence, $\mathbf{U}_c$ is a vector of dimension $h_c\times N_i$ containing the controls applied during the control time horizon $(k=1,\cdots,h_c)$.

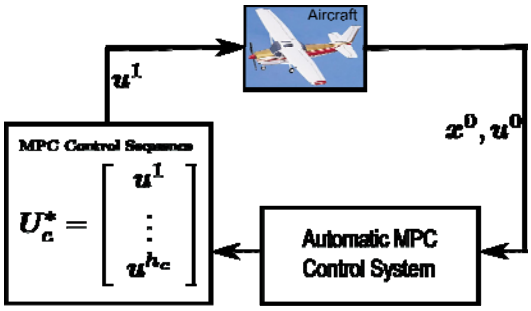


Fig. 3. Control loop scheme.

The last step in this formulation consists in expressing the cost function $J(\mathbf{U}_p)$ and the constraints in terms of the effective control sequence $\mathbf{U}_c$. This can be done straightforwardly by replacing (37) into (34) and (25). Finally, the minimization problem can be written in terms of $\mathbf{U}_c$ as:

$$\min_{\mathbf{U}_c} J(\mathbf{U}_c)=\mathbf{g}^T\breve{\mathbf{T}}\mathbf{U}_c+\mathbf{U}_c^T\breve{\mathbf{T}}^T\mathbf{H}\breve{\mathbf{T}}\mathbf{U}_c \quad st. \quad \mathbf{A}_{\text{ineq}}\breve{\mathbf{T}}\mathbf{U}_c\le\mathbf{b}_{\text{ineq}}. \tag{40}$$

The solution of the minimization problem of (40) gives the optimal control input sequence $\mathbf{U}_c^*\to\mathbf{U}_p^*$ which will command the airplane towards the desired maneuver.

The proposed INL-MPC control algorithm operates on-line as indicated below.

First, the current airplane's state $\mathbf{x}^0$ and current control inputs $\mathbf{u}^0$ are measured, then the optimal control sequence $\mathbf{U}_c^*$ is computed by solving (40). Then, only the control input $\mathbf{u}^1=\mathbf{U}_c^*(1)$ corresponding to the first sampling interval is applied to the aircraft. The control loop operation is shown in Fig. 3.

4.3. Algorithm

It assumed that the control algorithm is in a control unit embedded within the airplane. The control unit runs with a refresh rate equal to the discretization period ΔT_s; therefore, as shown in Fig. 1, the aircraft controls remain constant at $\mathbf{u}^k$ until it is refreshed to a new value $\mathbf{u}^{k+1}$. The algorithm and the control procedure using the present INL-MPC technique are given in Table 1.

5. APPLICATIONS

This section presents three aircraft's automatic maneuvers done with the presented INL-MPC technique:

1) Climb to a desired altitude at constant speed
2) Change of heading in the flight path at constant speed and altitude
3) Perform a coordinated turn at constant speed and altitude

For the presented applications it was assumed that the UAV is a Cessna 172 airplane. The state vector function $\bar{\mathbf{f}}(\mathbf{x},\mathbf{u})$ is given in (4). The necessary aerodynamic and propulsion forces and moments, together with the characteristic data of a Cessna 172 airplane were included in [24]. Sampling period of $\Delta T_s=0.5$ sec, prediction hori-

Table 1. INL-MPC algorithm.

Algorithm
Step 1: Initialize the variables *tol* and *maxIter* which indicate the accepted tolerance to break the iteration loop and the maximum number of iterations respectively. Set the iteration index i =1.
Step 2: Initialize the control inputs sequence $\mathbf{U}_p^{(i)}$ of (7) with the current control input $\mathbf{u}^0$ and consider it remains constant along the predictive interval $[t^0, t^{h_p}]$.
Step 3: Given the current airplane state $\mathbf{x}^0$ and the control input sequence $\mathbf{U}_p^{(i)}$ find the vector of predicted states $\widehat{\mathbf{X}}^{(i)}$ defined in (8) associated to the state-space trajectory $T(\mathbf{U}_p^{(i)})$.
Step 4: Linearize the non-linear aircraft system along the state-space trajectory $T(\mathbf{U}_p^{(i)})$ to calculate the state-space system matrices as defined in (13). Then, compute the prediction matrices defined in (A.1)-(A.3).
Step 5: Calculate the objective function's matrices defined in Eqs. (35) and (36) and the constraints matrices given in (A.5) so as to set the matrices of the minimization problem (40).
Step 6: Compute the optimal control input $\mathbf{U}_c^*$ with a suitable quadratic program algorithm. Recall that the control inputs for $k > h_c$ remain constant so that the complete control input sequence can be calculated as $\mathbf{U}_p^* = \bar{\mathbf{T}}\mathbf{U}_c^*$.
Step 7: *If,* $\left\|\mathbf{U}_p^* - \mathbf{U}_p^{(i)}\right\| \geq tol$ and $i \leq maxIter$ increment the iteration index $i \leftarrow i+1$, update the control input sequence $\mathbf{U}_p^{(i)} = \mathbf{U}_p^*$ for the next iteration and go back to **Step 3**.
Else, break the iteration loop and go to **Step 8**.
Step 8: Apply the first control input $\mathbf{u}^1 = \mathbf{U}_p^*(1)$ to the airplane and move the time horizon one step ahead to the next sampling step $t^0 \leftarrow t^0 + \Delta T_s$.

zon $h_p = 20$ and control horizon $h_c = 17$ are adopted. Both, h_p and h_c have been selected empirically, increasing the horizons beyond those values produced no significant changes in the system response and would produce an unnecessary increase in computing times.

In order to properly weight physical states of different order of magnitude, the weights $\widetilde{\mathbf{Q}}_x^k$ are re-normalized taking into account typical values $\mathbf{x}_{Typ}$ of states deviations. The generic typical deviation for the state component $\mathbf{x}(i)$ is noted as $\mathbf{x}_{Typ}(i)$. Then, the values of the weighting matrices (32) are selected as:

$$\widetilde{\mathbf{Q}}_x^k(i,i) = \frac{\mathbf{Q}_x^k(i,i)\Delta T_s}{max\{\mathbf{x}(i) - \mathbf{x}_{sp}(i), \mathbf{x}_{Typ}(i)\}^2}. \tag{41}$$

Furthermore, $\mathbf{Q}_x^k(i,j) = 0$ and $\mathbf{R}_{\dot{u}}^k(i,j) = 0$ for $i \neq j$. Typical values of states deviations are selected as:

$$\begin{aligned} &v_{t_{Typ}} = 5.0\frac{\text{m}}{\text{sec}}, \quad \alpha_{Typ} = 1.0\,\text{deg}, \quad \beta_{Typ} = 1.0\,\text{deg}, \\ &\phi_{Typ} = 5.0\,\text{deg}, \quad \psi_{Typ} = 5.0\,\text{deg}, \quad z_{Typ} = 100.0\,\text{m}. \end{aligned} \tag{42}$$

The weights $\mathbf{Q}_x^k(i,i)$ and $\mathbf{R}_{\dot{u}}^k(i,i)$ are defined for each example. Their weighting values have been selected empirically by simply weighting more the states and inputs according to the desired airplane maneuver.

For the three examples, the following inputs constraints have been used: $thtl_m = 0.0$, $thtl_M = 1.0$, $\delta_{em} = -20.0$ deg, $\delta_{eM} = 20.0$ deg, $\delta_{am} = -20.0$ deg, $\delta_{aM} = 20.0$ deg, $\delta_{rm} = -15.0$ deg, $\delta_{rM} = 15.0$ deg.

The inputs rates constraints used in the examples are: $\dot{thtl}_m = -0.2$ 1/sec, $\dot{thtl}_M = 0.2$ 1/sec, $\dot{\delta}_{em} = -2.0$ deg/sec, $\dot{\delta}_{eM} = 2.0$ deg/sec, $\dot{\delta}_{am} = -2.0$ deg/sec, $\dot{\delta}_{aM} = 2.0$ deg/sec, $\dot{\delta}_{rm} = -2.0$ deg/sec, $\dot{\delta}_{rM} = 2.0$ deg/sec.

Finally, the states constraints used are: $v_{t_m} = 10.0$ m/sec, $v_{t_M} = 76.0$ m/sec, $\alpha_m = -16.0$ deg, $\alpha_M = 16.0$ deg, $\beta_m = -15.0$ m, $\beta_M = 15.0$ deg, $h_m = 0.0$ m, $h_M = 4000.0$ m.

5.1. First example: climbing maneuver at constant speed

The proposed INL-MPC technique is applied to the aircraft, initially flying at an altitude $h = 1000.0$ m and speed $v_t = 45.0$ m/sec, to perform a climbing maneuver to a new altitude $h = 1500.0$ m while keeping a constant speed. So the elements of the setpoint vector are selected as: $\mathbf{x}_{sp}(1) = v_{t_{sp}} = 45.0$, $\mathbf{x}_{sp}(3) = \beta_{sp} = 0.0$, $\mathbf{x}_{sp}(4) = \phi_{sp} = 0.0$, $\mathbf{x}_{sp}(6) = \psi_{sp} = 0.0$ and $\mathbf{x}_{sp}(12) = h_{sp} = 1500.0$. To perform the climbing maneuver, the optimization weights are chosen as follows: $\mathbf{Q}_x^k(1,1) = 10.0$, $\mathbf{Q}_x^k(3,3) = 10.0$, $\mathbf{Q}_x^k(4,4) = 10.0$, $\mathbf{Q}_x^k(6,6) = 10.0$, $\mathbf{Q}_x^k(12,12) = 10.0$ and $\mathbf{R}_{\dot{u}}^k(1,1) = 0.1$, $\mathbf{R}_{\dot{u}}^k(2,2) = 0.1$, $\mathbf{R}_{\dot{u}}^k(3,3) = 0.1$, $\mathbf{R}_{\dot{u}}^k(4,4) = 0.1$, with $k = 1, \cdots, h_p$. Note that only the states involved in the climbing maneuver have been weighted: velocity v_t, sideslip angle β, Euler angles ϕ and ψ and height h. The Euler angles are weighted in order to maintain the straight and level flight condition.

In Fig. 4, it can be seen the evolution of the aircraft altitude and speed when the climbing maneuver indicated by the INL-MPC control commands is performed. Note how the INL-MPC automatic control system allowed the aircraft to reach the desired altitude while maintaining the speed practically constant. In Fig. 5, it is shown the evolution of throttle and elevator inputs, respectively. The commanded throttle increases so as the airplane starts climbing. When the airplane is near the desired

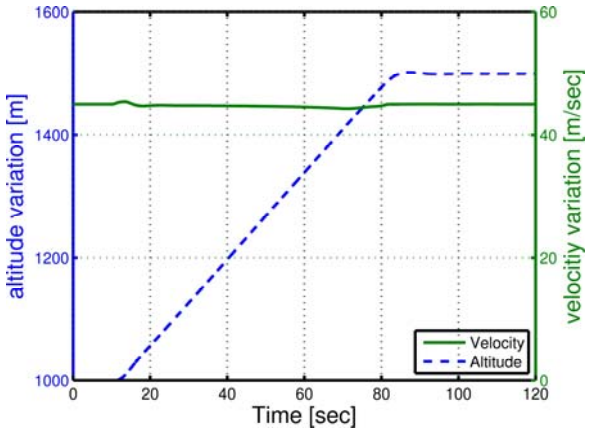


Fig. 4. Climbing maneuver.

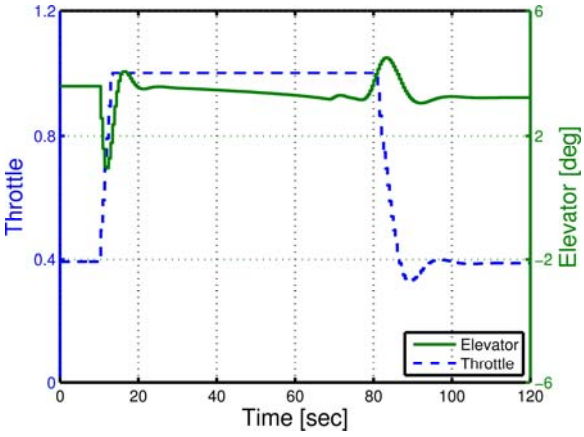


Fig. 5. Control inputs - climbing maneuver.

altitude the throttle is reduced to prevent the aircraft continuing climbing. The elevator moves to produce a variation in the angle of attack so as to maintain constant speed.

5.2. Second example: change of heading in the flight path at constant speed and altitude

Consider that the unmanned airplane is again initially flying at an altitude $h = 1000.0$ m and speed $v_t = 45.0$ m/sec. The INL-MPC algorithm is applied to the UAV to perform a change of 90 deg in the direction of the flight path, while keeping constant both altitude and speed. Thus, the desired values of the setpoint vector are configured as follows: $\mathbf{x}_{sp}(1) = v_{t_{sp}} = 45.0$, $\mathbf{x}_{sp}(3) = \beta_{sp} = 0.0$, $\mathbf{x}_{sp}(6) = \psi_{sp} = \frac{\pi}{2}$ and $\mathbf{x}_{sp}(12) = h_{sp} = 1000.0$. The state weights are chosen as follows: $\mathbf{Q}_x^k(1,1) = 10.0$, $\mathbf{Q}_x^k(3,3) = 1.0$, $\mathbf{Q}_x^k(6,6) = 1.0$, $\mathbf{Q}_x^k(12,12) = 10.0$. Only the states involved in this maneuver have been weighted: velocity v_t, sideslip angle β, yaw angle ψ and height h. The input weight are chosen as follows: $\mathbf{R}_u^k(1,1) = 0.1$, $\mathbf{R}_u^k(2,2) = 0.1$, $\mathbf{R}_u^k(3,3) = 0.1$, $\mathbf{R}_u^k(4,4) = 0.1$; with $k = 1, \cdots, h_p$. The evolution of the aircraft's position trajectory is shown in Fig. 6.

As it can be seen, the INL-MPC automatic control system allowed the aircraft to perform the ninety-degrees

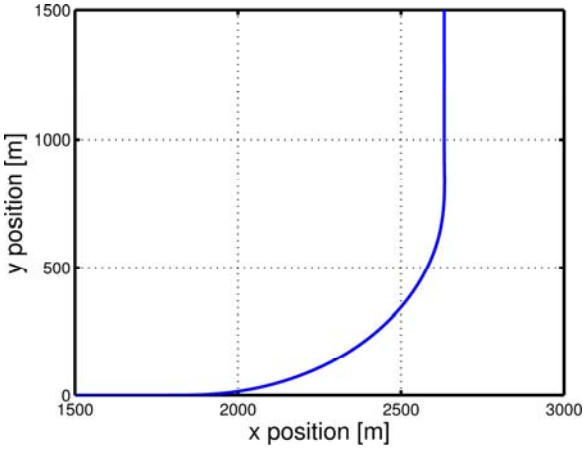
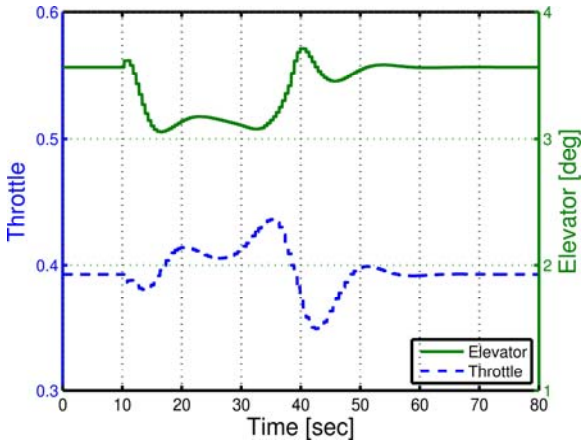
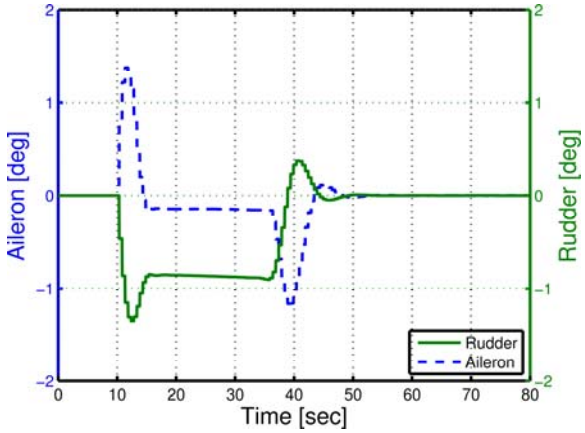


Fig. 6. Change of heading.



(a) Control inputs: throttle and elevator.



(b) Control inputs: aileron and rudder.

Fig. 7. Control inputs - change of heading.

heading change. The maneuver was performed keeping constant altitude and speed. The evolution of the four control inputs which are involved in the maneuver are shown in Figs. 7(a) to 7(b). The throttle and the elevator move jointly to maintain constant values of altitude and velocity. The aileron and rudder produce a change in roll and yaw moments and as a result the airplane performs the change of heading in the flight path.

5.3. Third example: coordinated turn at constant speed and altitude

The last autonomous maneuver to be performed is a coordinate turn at constant speed and altitude. Once again, the airplane is initially flying at an altitude $h = 1000.0$ m and speed $v_t = 45.0$ m/sec. The INL-MPC control technique is applied to the unmanned aircraft to perform a coordinated turn while keeping constant speed and altitude. So, the setpoint vector is configured as follows: $\mathbf{x}_{sp}(1) = v_{t_{sp}} = 45.0$, $\mathbf{x}_{sp}(3) = \beta_{sp} = 0.0$, $\mathbf{x}_{sp}(4) = \phi_{sp} = \frac{15.0\pi}{180.0}$ and $\mathbf{x}_{sp}(12) = h_{sp} = 1000.0$. The state weights are chosen as follows: $\mathbf{Q}_x^k(1,1) = 10.0$, $\mathbf{Q}_x^k(3,3) = 1.0$, $\mathbf{Q}_x^k(4,4) = 1.0$, $\mathbf{Q}_x^k(12,12) = 10.0$. Only the states that concern involved in this maneuver have been weighted: velocity v_t, sideslip angle β, roll angle ϕ and height h. The input weights are chosen as follows: $\mathbf{R}_u^k(1,1) = 0.1$,

$\mathbf{R}_{\tilde{u}}^{k}(2,2) = 0.1$, $\mathbf{R}_{\tilde{u}}^{k}(3,3) = 0.1$, $\mathbf{R}_{\tilde{u}}^{k}(4,4) = 0.1$; with $k = 1,\cdots,h_p$. Form Fig. 8 it can be seen that the resulting airplane position trajectory clearly corresponds to a coordinated turn. The maneuver was performed keeping constant altitude and speed. The evolution of the four control inputs are shown in Figs. 9(a) to 9(b). The throttle and elevator inputs move together to maintain both altitude and velocity at constant values. It can be seen that the aileron and the rudder reach a final position that makes the roll angle to go to the desired value.

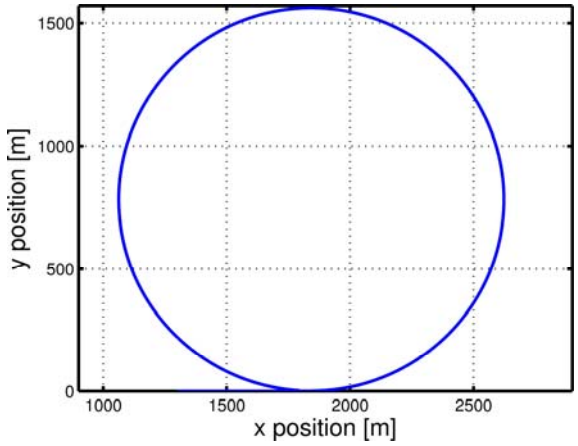
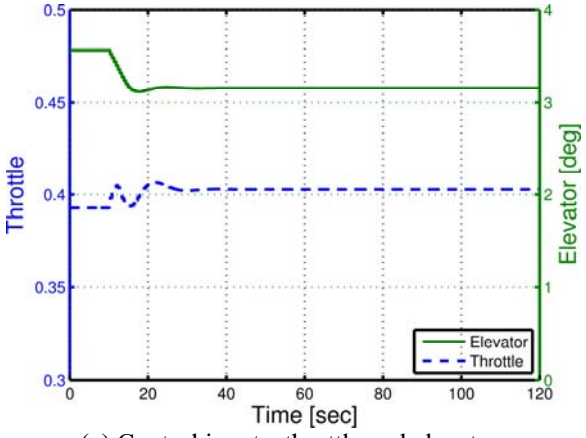
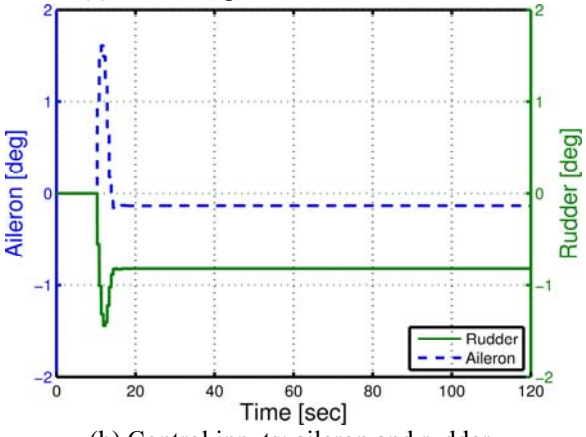


Fig. 8. Coordinated turn maneuver.



(a) Control inputs: throttle and elevator.



(b) Control inputs: aileron and rudder.

Fig. 9. Control inputs - coordinated turn maneuver.

5.4. Additional comments

For the presented application, the well-posedness of each iterative minimization sub-problem, defined in (40), has been evaluated and tested. Meaning that at each iteration a *correct convex* sub-problem is being solved. Because of the convexity, the termination of the optimization problem can be guaranteed. As a consequence, the resulting optimization problems of the INL-MPC algorithm are always well-posed.

The proposed algorithm has several advantages: 1) It allows the inclusion of generic non-linear systems, 2) The algorithm can be used as a centralized unit, which is very important as it can take into account the dynamic couplings between different reduced modes, 3) The proposed method reduces the problem of non-linear optimal control of a non-linear system to a series of iterative, *easily solvable*, quadratic optimization sub-problems, and 4) With the proposed method constraints can be handled with ease, also, state and control inputs rates can be penalized, allowing to take into account different physical limitations of real systems.

Regarding the selection of the sampling period ΔT_s, it must be mentioned that, as the maximum natural frequency of the modeled system is always below $f_{max} = 0.71\,\text{Hz}$, the sampling theorem requires that $\Delta T_s \leq 0.7$ sec, so $\Delta T_s = 0.5$ sec was chosen.

A few additional comments regarding the implementation of the INL-MPC are given below. The UAV model and the INL-MPC have been programmed in C++, using the LTensor Library for matrix computations [25], they run in parallel in independent computing units. Both the UAV model and the INL-MPC algorithm run in *real-time*. At the point of solving the quadratic optimization problem (40) and Step 6 of Algorithm given in Table 1 a call to Matlab's *quadprog* function (accessed from C++ as a dynamic library) is made. Within this function, the *Interior Point* algorithm was chosen. The maximum computing time of the INL-MPC implementation is $\Delta t_{\text{cpu}} = 0.3$ sec (with three iterations in the trajectory optimization loop). This is always less than the control refresh rate $\Delta T_s = 0.5$ sec, guaranteeing a real-time simulation and control.

6. CONCLUSIONS

In this article, an iterative non-linear MPC (INL-MPC) technique for controlling general non-linear systems is presented. The INL-MPC algorithm is based on a recursive linearization of the non-linear system's dynamics along iteratively defined state-space trajectories. The proposed method reduces the problem of non-linear optimal control of a non-linear system to a series of iterative, easily solvable, quadratic optimization sub-problems. The proposed technique allows the inclusion of constraints with ease. General constraints have been addressed. Finally, the control algorithm's performance is demonstrated through the realization of three different autonomous maneuvers on a full 6-degree of freedom, Cessna 172 aircraft model. In all cases the autonomous maneuvers were performed successfully.

APPENDIX A

Matrix definitions:

$$\mathbf{D}=\begin{bmatrix}\mathbf{d}^0\\ \mathbf{d}^1\\ \mathbf{d}^2\\ \vdots\\ \mathbf{d}^{h_p-1}\end{bmatrix},\quad \mathbf{P}=\begin{bmatrix}\widetilde{\mathbf{A}}^0\\ \widetilde{\mathbf{A}}^1\widetilde{\mathbf{A}}^0\\ \widetilde{\mathbf{A}}^2\widetilde{\mathbf{A}}^1\widetilde{\mathbf{A}}^0\\ \vdots\\ \widetilde{\mathbf{A}}^{h_p-1}\cdots\widetilde{\mathbf{A}}^0\end{bmatrix}, \tag{A.1}$$

$$\mathbf{H}_u=\begin{bmatrix}\widetilde{\mathbf{B}}^0 & \mathbf{0} & \cdots & \mathbf{0}\\ \widetilde{\mathbf{A}}^1\widetilde{\mathbf{B}}^0 & \widetilde{\mathbf{B}}^1 & \cdots & \mathbf{0}\\ \widetilde{\mathbf{A}}^2\widetilde{\mathbf{A}}^1\widetilde{\mathbf{B}}^0 & \widetilde{\mathbf{A}}^2\widetilde{\mathbf{B}}^1 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \widetilde{\mathbf{A}}^{h_p-1}\cdots\widetilde{\mathbf{A}}^1\widetilde{\mathbf{B}}^0 & \widetilde{\mathbf{A}}^{h_p-1}\cdots\widetilde{\mathbf{A}}^2\widetilde{\mathbf{B}}^1 & \cdots & \widetilde{\mathbf{B}}^{h_p-1}\end{bmatrix}, \tag{A.2}$$

$$\mathbf{H}_g=\begin{bmatrix}\widetilde{\mathbf{G}}^0 & \mathbf{0} & \cdots & \mathbf{0}\\ \widetilde{\mathbf{A}}^1\widetilde{\mathbf{G}}^0 & \widetilde{\mathbf{G}}^1 & \cdots & \mathbf{0}\\ \widetilde{\mathbf{A}}^2\widetilde{\mathbf{A}}^1\widetilde{\mathbf{G}}^0 & \widetilde{\mathbf{A}}^2\widetilde{\mathbf{G}}^1 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \widetilde{\mathbf{A}}^{h_p-1}\cdots\widetilde{\mathbf{A}}^1\widetilde{\mathbf{G}}^0 & \widetilde{\mathbf{A}}^{h_p-1}\cdots\widetilde{\mathbf{A}}^2\widetilde{\mathbf{G}}^1 & \cdots & \widetilde{\mathbf{G}}^{h_p-1}\end{bmatrix}, \tag{A.3}$$

$$\mathbf{E}=\begin{bmatrix}\mathbf{I}_{N_i} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ -\mathbf{I}_{N_i} & \mathbf{I}_{N_i} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i} & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I}_{N_i} & \mathbf{I}_{N_i}\end{bmatrix} \text{ and } \mathbf{U}_0=\begin{bmatrix}-\mathbf{u}^0\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}, \tag{A.4}$$

where $\mathbf{I}_{N_i}$ is an identity matrix of dimension $N_i\times N_i$.

$$\mathbf{A}_{\text{ineq}}=\begin{bmatrix}\mathbf{I}_p\\ -\mathbf{I}_p\\ \mathbf{E}\\ -\mathbf{E}\\ \mathbf{H}_u\\ -\mathbf{H}_u\end{bmatrix},\quad \mathbf{b}_{\text{ineq}}=\begin{bmatrix}\mathbf{U}_M\\ -\mathbf{U}_m\\ \dot{\mathbf{U}}_M\,\Delta T_s-\mathbf{U}_0\\ -\dot{\mathbf{U}}_m\,\Delta T_s+\mathbf{U}_0\\ \mathbf{X}_M-\mathbf{P}\mathbf{x}^0-\mathbf{H}_g\mathbf{D}\\ -\mathbf{X}_m+\mathbf{P}\mathbf{x}^0+\mathbf{H}_g\mathbf{D}\end{bmatrix}, \tag{A.5}$$

$$\breve{\mathbf{Q}}_x=\begin{bmatrix}\widetilde{\mathbf{Q}}_x^1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \widetilde{\mathbf{Q}}_x^2 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \widetilde{\mathbf{Q}}_x^{h_p}\end{bmatrix}, \tag{A.6}$$

$$\breve{\mathbf{R}}_{\dot{u}}=\begin{bmatrix}\widetilde{\mathbf{R}}_{\dot{u}}^1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \widetilde{\mathbf{R}}_{\dot{u}}^2 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \widetilde{\mathbf{R}}_{\dot{u}}^{h_p}\end{bmatrix},$$

and

$$\mathbf{X}_{sp}=\begin{bmatrix}\mathbf{x}_{sp}^1\,\mathbf{x}_{sp}^2\cdots\,\mathbf{x}_{sp}^{h_p}\end{bmatrix}^T, \tag{A.7}$$

$$\mathbf{I}_c=\begin{bmatrix}\mathbf{I}_{N_i} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{N_i} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i}\end{bmatrix}$$

$$\mathbf{V}=\begin{bmatrix}\mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_i}\end{bmatrix}, \tag{A.8}$$

where $\mathbf{I}_{N_i}$ and $\mathbf{I}_c$ are an identity matrices of dimension $N_i\times N_i$ and $(N_i\times h_c)\times(N_i\times h_c)$, respectively. $\mathbf{V}$ is a matrix of dimension $(N_i\times(h_p-h_c))\times(N_i\times h_c)$.

REFERENCES

[1] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, Wiley New York, 2007.

[2] B. Stevens and L. Lewis, *Aircraft Control and Simulation*, John Wiley & Sons Inc., 1992.

[3] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, DARcorporation, 2001.

[4] A. Bemporad and M. Morari, "Robust model predictive control: a survey, in robustness in identification and control," *Lecture Notes in Control and Information Sciences*, vol. 245, pp. 207-226, 1999.

[5] B. Kouvaritakis and M. Cannon, *Nonlinear Predictive Control: Theory and Practice*, IET, 2001.

[6] P. Falcone, M. Tufo, F. Borrelli, H. Asgari, and H. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," *Proc. of the 46th IEEE Conf. on Decision and Control*, pp. 2980-2985, 2007.

[7] Z. Yang, X. Qi, and G. Shan, "Simulation of flight control laws design using model predictive controllers," *Proc. of International Conference on Mechatronics and Automation*, pp. 4213-4218, 2009.

[8] R. Anderson, E. Bakolas, D. Milutinović, and P. Tsiotras, "Optimal feedback guidance of a small aerial vehicle in a stochastic wind," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4,

pp. 975-985, 2013.
[9] J. Wilburn, J. Cole, M. Perhinschi, and B. Wilburn, "Comparison of a fuzzy logic controller to a potential field controller for real-time UAV navigation," *Proc. of AIAA Guidance, Navigation, and Control Conference*, AIAA, 2012.
[10] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36-44, 2010.
[11] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, "Path following for small unmanned aerial vehicles using L1 adaptive augmentation of commercial autopilots," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 550-564, 2010.
[12] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp. 1195-1207, 2011.
[13] P. Gibbens and E. Medagoda, "Efficient model predictive control algorithm for aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1909-1915, 2011.
[14] I. Prodan, R. Bencatel, S. Olaru, J. Sousa, C. N. Stoica, and S. I. Niculescu, "Predictive control for autonomous aerial vehicles trajectory tracking," *Nonlinear Model Predictive Control*, vol. 4, no. 1, pp. 508-513, 2012.
[15] T. Keviczky and G. Balas, "Receding horizon control of an F-16 aircraft: a comparative study," *Control Engineering Practice*, vol. 14, no. 9, pp. 1023-1033, 2006.
[16] K. Yang, Y. Kang, and S. Sukkarieh, "Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle," *International Journal of Control, Automation and Systems*, vol. 11, no. 1, pp. 65-74, 2013.
[17] J. Maciejowski, *Predictive Control: with Constraints*, Prentice Hall, 2002.
[18] C. Jones, A. Domahidi, M. Morari, S. Richter, F. Ullmann, and M. N. Zeilinger, "Fast predictive control: real-time computation and certification," in *Nonlinear Model Predictive Control*, vol. 4, no. 1, pp. 94-98, 2012.
[19] M. Morari and J. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, no. 4-5, pp. 667-682, 1999.
[20] J. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, pp. 415-424, 2011.
[21] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*, Springer, 2011.
[22] A. Limache, P. Rojas, and M. Murillo, "Diseño de un moderno simulador de vuelo en tiempo real," 2010.
[23] N. Slegers, J. Kyle, and M. Costello, "Nonlinear model predictive control technique for unmanned air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1179-1188, 2006.
[24] https://drive.google.com/file/d/0B88B-nVTQvRAa0pncDFfV21McGc/edit?usp=sharing,.
[25] https://code.google.com/p/ltensor/,.

Marina H. Murillo received her B.S. degree in Electrical Engineering from Universidad Nacional de Rosario in 2008. She received a CONICET scholarship to start her Ph.D. studies in 2010. Her research interests include control theory and development of navigation & control systems for unmanned aerial vehicles.

Alejandro C. Limache received his Ph.D. degree in Aerospace Engineering from Virginia Polytechnic Institute & State University (Virginia Tech) in 2000. His research interests include real-time simulation, multiphysics based on particle meshless methods, control theory and development of navigation & control systems for unmanned aerial vehicles.

Pablo S. Rojas Fredini received his B.S. degree in Computer Science from Universidad Nacional del Litoral in 2009. He received a CONICET scholarship to start his Ph.D. studies in 2009. His research interests include real time simulation, high performance and parallel algorithms, computer graphics, virtual reality and videogames.

Leonardo L. Giovanini received his Ph.D. degree in Engineering, Computational Mechanics mention from Universidad Nacional del Litoral in 2000. His research interests include modelling and analysis of complex systems, multiagents systems, parameter estimation and fault detection, control and estimation using receding horizons techniques.