DOI: 10.48011/asba.v2i1.1643

Explicit Model Predictive Control for a Tiltrotor UAV in Cargo Transportation Tasks ⋆

Richard Andrade * **Antonio Ferramosca** ** **Julio E. Normey-Rico** *** **Guilherme V. Raffo** ****

* *Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil*
** *CONICET - UTN FRRQ, Reconquista, Argentina*
*** *Department of Automation and Systems, Federal University of Santa Catarina, Florianópolis, SC, Brazil*
**** *Department of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil*

Abstract: This paper deals with the trajectory tracking problem of a tilt-rotor unmanned aerial vehicle carrying a suspended load. An explicit model predictive control (eMPC) based on multi-parametric optimization is used to derive optimal control laws which could be implemented in an embedded system. The eMPC is designed based on the nominal linearized error model of the system, which is obtained around a generic trajectory. The optimal control problem (OCP) is solved taking into account input and state constraints. Additionally, a terminal cost is considered to guarantee stability. Euler-Lagrange formulation is used to derive the multibody non-linear dynamic model. Numerical experiments are performed to evaluate the proposed controller when the system is affected by constant disturbances at different instants of time and parametric uncertainties.

Keywords: Tilt-rotor UAV, Load transportation, Explicit MPC.

1. INTRODUCTION

Over the past few years, autonomous aircraft have been widely involved in civilian applications, where usually is required load transportation of medicines, food, supplies, and sensors. In this context, multi-rotor unmanned aerial vehicles (UAVs) are suitable to perform these tasks, considering their capabilities of vertical take-off and landing, hovering, and high maneuverability. However, in some missions, it is necessary to achieve long distances with higher speeds while maintaining the benefits of a multi-rotor aircraft. A proper UAV architecture to cover these requirements is derived from the tiltrotor configuration, which can perform transition among two flight modes: rotary-wing and fixed-wing.

In the literature, some works have used tiltrotor UAVs for suspended load transportation due to the aforementioned advantages. In de Almeida and Raffo (2015), a nonlinear cascade control strategy is proposed to solve the suspended load transportation problem using a tiltrotor unmanned aerial vehicle. The latter work was extended in Raffo and Almeida (2018) by using a two-level cascade controller, in which an input-output feedback linearization control law with dynamic extension was designed to control the attitude, altitude and tilting mechanisms. Thus, the outer-loop control law was in charge of guiding the aircraft and reducing the load swing. In Rego and Raffo (2019) the cargo transportation problem is formulated from the load point of view, in which the load is required to track a reference trajectory. To perform the suspended load trajectory tracking a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ discrete-time controller with pole placement constraints is designed.

Recently, Model Predictive Control (MPC) is being used to control fast dynamic systems due to its ability to explicitly deal with operational constraints in MIMO systems. For example, in Santos et al. (2017), an MPC strategy is proposed for path tracking of the suspended load with stabilization of the tiltrotor UAV when parametric uncertainties and external disturbances affect the load. However, the main drawback of MPC is the high computational demand to calculate the optimal control sequence, preventing its implementation in fast mechatronic embedded systems, which need to be controlled in the range of milliseconds.

In order to deal with this problem, some control techniques were developed. In Bemporad et al. (2002), the explicit model predictive control (eMPC) algorithm was proposed, in which a group of optimal control affine functions is determined off-line and explicitly. Consequently, it can be implemented analytically in an embedded system and deal

⋆ This work was in part supported by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3, FAPESP (São Paulo Research Foundation), Brazil 2014/50851-0. This work was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil (Finance Code 88887.136349/2017-00, and 001), CNPq, Brazil (grant numbers 313568/2017-0 and 426392/2016-7), and FAPEMIG, Brazil (grant number APQ-03090-17).

with fast dynamic systems. Nonetheless, as the solution needs to be stored, it demands a lot of memory that depends on the system dimension. Tøndel et al. (2003) extend the theoretical results of Bemporad et al. (2002), improving the efficiency of the previous algorithm, by analyzing several related properties of the polyhedral partition geometry with the active constraints' combination at the optimum. In Kouramas et al. (2011), a method employing dynamic and multi-parametric programming techniques was developed to disassemble the MPC formulation into a set of smaller stage optimization sub-problems, considering optimal control variables for each stage as a function of the system states. Furthermore, Pistikopoulos (2012) presented recent advances and applications of multi-parametric programming and eMPC, together with the concept of MPC on-a-chip. In Voelker et al. (2013), a MPC using multi-parametric programming is presented based on a simultaneous explicit design methodology for a constrained moving horizon estimation.

In order to reduces the computational complexity while maintaining the controller performance, Lee and Chang (2018) designed an eMPC to perform a double-lane-change (DLC) maneuver for discrete linear time-variant (LTV) systems. Besides, the reduction of the number of critical regions is used as an alternative to decrease the high memory demand of the eMPC, as presented in Kvasnica and Fikar (2010), in which the regions where the control law becomes saturated are eliminated by using the neighbors unsaturated regions. In Maddalena et al. (2019b), a region elimination procedure is presented based on the premise of knowing the initial state or the limits of the initial condition. This algorithm assumes that an initial condition spreads over n regions of the original polyhedral partition. On the other hand, in Maddalena et al. (2019a), a neuronal network is trained and the training weights are used to compute a new explicit affine control laws, which results in a lower number of regions with an approximated behavior to the original controller.

In this paper, the trajectory tracking problem of a tiltrotor UAV, ensuring swing-load free transportation, is solved with an explicit linear model predictive controller subjected to state and control constraints. This controller is based on a non-incremental linearized nominal error model of the states when following the reference state trajectory. In order to ensure constant disturbance rejection, an integral action of the position state error is also included. Moreover, a terminal cost is formulated for this eMPC using linear matrix inequalities (LMIs). Finally, the explicit solutions are stored using three dimensional matrices, which permit to search and compute the control law by parallel computation in order to reduce the computational time of the eMPC.

The rest of the paper is organized as follows: Section 2 develops the dynamic model of the tiltrotor UAV with suspended load; in Section 3 the non-linear model is linearized around a desired trajectory. The explicit MPC formulation is described in Section 4, while implementations aspects of the formulation are presented in Section 5. Numerical results are depicted in Section 6. Finally, conclusion comments are presented.

2. TILTROTOR DYNAMIC MODEL

As presented in Figure 1, the mechanical system is composed by four rigid bodies, where for mathematical purposes, six references axis are defined, with $\mathcal{C}_i$ being four axes rigidly fixed to the center of mass of the bodies, frame $\mathcal{B}$ is fixed to the UAV geometric center, and the inertia frame $\mathcal{J}$ is fixed to the Earth.

The vector $\boldsymbol{\xi} = [x\ y\ z]'$ [1] describes the translational position of frame $\mathcal{B}$ with respect to $\mathcal{J}$. Besides, the rotational position of frame $\mathcal{B}$ with respect to $\mathcal{J}$ is parametrized by the roll, pitch and yaw angles, $\boldsymbol{\eta} = [\phi\ \theta\ \psi]'$, known as Euler angles and using the ZYX convention around the local axes. The vector $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2]'$ gives the angular position of the suspended load, while the angles α_r and α_l represent the angular position, with respect to frame $\mathcal{B}$, of the right and left rotors, respectively. In this context, the generalized coordinates vector is defined as $\boldsymbol{q} = [\boldsymbol{\xi}\ \boldsymbol{\eta}\ \alpha_r\ \alpha_l\ \boldsymbol{\gamma}]'$.

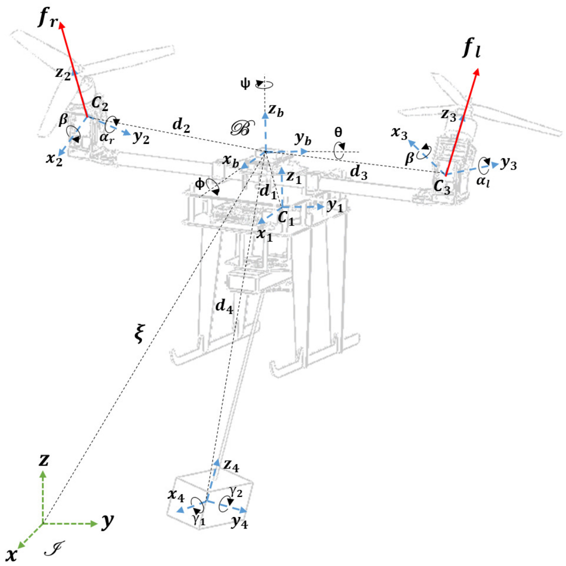

Figure 1. The tiltrotor UAV with suspended load description, in which the reference frames, kinematic parameters, control inputs and generalized coordinates are illustrated.

The non-linear model of the tiltrotor UAV with suspended load is obtained via Euler-Lagrange formulation, which is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{F_{drag}}, \qquad (1)$$

where the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is derived from the sum of the kinetic energies of all bodies, given by $\boldsymbol{K} = \frac{1}{2}\dot{\boldsymbol{q}}'\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$. The Coriolis and centripetal forces matrix $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is obtained from the inertia matrix using the Chirstoffel symbols of the first kind. Further, the gravitational force vector is computed as $\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{q}}$, with $\boldsymbol{P}$ being the potential energies summation of the aircraft's bodies. Finally, $\boldsymbol{F}(\boldsymbol{q})$ is the generalized forces/torques vector, and $\boldsymbol{F_{drag}}$ is the viscous friction force vector present in the tilting mechanisms and the load joint.

[1] The prime $'$ notation denotes the transpose operator.

The generalized forces/torques vector can be expanded as $\boldsymbol{F}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}(t)$. The control input vector, $\boldsymbol{u}(t) = [f_r\ f_l\ \tau_r\ \tau_l]'$, is composed by f_r and f_l, the right and left thrust forces generated by the propellers, and τ_r and τ_l, the right and left torques applied to the servomotors. In the same context, the viscous friction vector can be assumed as $\boldsymbol{F_{drag}} = -\boldsymbol{\Psi}\cdot\dot{\boldsymbol{q}}$, where $\boldsymbol{\Psi}$ is a semidefinite constant matrix.

In order to obtain the state space representation of the non-linear model of the system, $\dot{\boldsymbol{x}}_{\boldsymbol{s}}(t) = f(\boldsymbol{x_s}(t), \boldsymbol{u}(t))$, the state space vector can be defined as $\boldsymbol{x_s}(t) \triangleq \left[\boldsymbol{q}'\ \dot{\boldsymbol{q}}'\right]'$.

The system's equilibrium point ($\boldsymbol{u}^*$,$\boldsymbol{q}^*$,$\dot{\boldsymbol{q}}^*$) can be found solving the following algebraic equation

$$\begin{bmatrix} \dot{\boldsymbol{q}}^* \\ \boldsymbol{M}^{-1}(\boldsymbol{B}(\boldsymbol{q}^*)\boldsymbol{u}^* - [\boldsymbol{C}(\boldsymbol{q}^*, \dot{\boldsymbol{q}}^*) + \boldsymbol{\Psi}]\dot{\boldsymbol{q}}^* - \boldsymbol{G}(\boldsymbol{q}^*)) \end{bmatrix} = 0. \tag{2}$$

A more detailed description about the tiltrotor UAV with suspended load is available in Raffo and Almeida (2018).

3. LINEARIZED MODEL

In this section, a linear time-varying model is obtained by performing a linearization of the non-linear model of the tiltrotor UAV with suspended load around a generic desired trajectory. The UAV with a suspended load is an underactuated system since it has ten degrees of freedom and only four control signals. Therefore, it is only possible to perform trajectory tracking of four degrees of freedom, and the remaining ones must be stabilized around the equilibrium point given by (2).

The linear error model is computed using a generic reference trajectory. The reference state vector is defined as $\boldsymbol{x_{sr}}(t) = [x_r(t)\ y_r(t)\ z_r(t)\ \phi^*\ \theta^*\ \psi_r(t)\ \alpha_r^*\ \alpha_l^*\ \gamma_1^*\ \gamma_2^*\ \dot{x}_r(t)\ \dot{y}_r(t)\ \dot{z}_r(t)\ 0\ 0\ 0\ 0\ 0\ 0\ 0]'$, and the reference control vector is computed by solving $\boldsymbol{u_r}(t) = \boldsymbol{B}(\boldsymbol{q_r})^{\#}(\boldsymbol{M}(\boldsymbol{q_r})\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + [\boldsymbol{C}(\boldsymbol{q_r}, \dot{\boldsymbol{q}}_{\boldsymbol{r}}) + \gamma]\boldsymbol{q_r} + \boldsymbol{G}(\boldsymbol{q_r}))$, assuming a feasible reference trajectory. [2]

Then, the linear dynamic model of the error is obtained using the first order Taylor series, with the error state vector $\tilde{\boldsymbol{x}}_{\boldsymbol{s}}(t) = \boldsymbol{x_s}(t) - \boldsymbol{x_{sr}}(t)$ and the error control vector $\tilde{\boldsymbol{u}}(t) = \boldsymbol{u}(t) - \boldsymbol{u_r}(t)$. Consequently, the system is written as

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}_{\boldsymbol{s}}(t) &= \boldsymbol{A}(t)\tilde{\boldsymbol{x}}_{\boldsymbol{s}}(t) + \boldsymbol{B}\tilde{\boldsymbol{u}}(t) \\ \tilde{\boldsymbol{y}}_{\boldsymbol{s}}(t) &= \boldsymbol{C}\tilde{\boldsymbol{x}}_{\boldsymbol{s}}(t), \end{aligned} \tag{3}$$

where $\boldsymbol{A}(t) = \left.\frac{\partial f(\boldsymbol{x_s}, \boldsymbol{u})}{\partial \boldsymbol{x_s}}\right|_{\substack{\boldsymbol{x_s}=\boldsymbol{x_{sr}} \\ \boldsymbol{u}=\boldsymbol{u_r}}}$, $\boldsymbol{B} = \left.\frac{\partial f(\boldsymbol{x_s}, \boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x_s}=\boldsymbol{x_{sr}} \\ \boldsymbol{u}=\boldsymbol{u_r}}}$, and $\boldsymbol{C} = \begin{bmatrix} I_{3\times3} & 0_{3\times21} \\ 0_{1\times3} & [0\ 0\ 1\ 0 \cdots\ 0] \end{bmatrix}$.

System (3) is a linear time-varying model (LTV) due to the fact that $\boldsymbol{A}(t)$ varies with the time-varying reference trajectory.

Moreover, the state vector is extended with the integral action of the position error in x, y, z and ψ, in order to achieve offset free path tracking and rejection of constant disturbances. Thus, the augmented system is rewritten as

$$\dot{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(t) = \boldsymbol{A_a}(t)\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(t) + \boldsymbol{B_a}\tilde{\boldsymbol{u}}(t), \tag{4}$$

with $\boldsymbol{A_a}(t) = \begin{bmatrix} \boldsymbol{A}(t) & 0 \\ \boldsymbol{C} & 0 \end{bmatrix}$, $\boldsymbol{B_a} = \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix}$, and the augmented error state vector given by

$$\tilde{\boldsymbol{x}}_a(t) = \left[\tilde{\boldsymbol{x}}_{\boldsymbol{s}}(t)' \int(\tilde{x}(t)) \int(\tilde{y}(t)) \int(\tilde{z}(t)) \int(\tilde{\psi}(t))\right]'. \tag{5}$$

Finally, in order to apply the explicit model predictive formulation, the Euler approximation is used to discretize the system (4), defining the discrete state matrix as $\boldsymbol{A_z}(kT) = \boldsymbol{I} - \boldsymbol{A_a}(t)\cdot T$ and the discrete control matrix as $\boldsymbol{B_z} = \boldsymbol{B_a}\cdot T$, with T being the sampling time. Hence, the discrete time-varying model is written as

$$\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k+1) = \boldsymbol{A_z}(k)\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) + \boldsymbol{B_z}\tilde{\boldsymbol{u}}(k). \tag{6}$$

For notational simplicity, the discrete terms $a(kT)$ will be written as $a(k)$.

4. EXPLICIT MODEL PREDICTIVE CONTROL FORMULATION FOR THE TILT-ROTOR UAV

In this section, the proposed eMPC is developed, which is an extension of the MPC presented in Andrade et al. (2016), using multi-parametric optimization. For the eMPC design proposes, it is considered the nominal error linear model (i.e. $\boldsymbol{A_z}(k) = \bar{\boldsymbol{A}}_{\boldsymbol{z}}$), which is given by

$$\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k+1) = \bar{\boldsymbol{A}}_{\boldsymbol{z}}\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) + \boldsymbol{B_z}\tilde{\boldsymbol{u}}(k). \tag{7}$$

Consider the prediction of the discrete model at the instant j, $\hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+j)$, $j = 1, 2..., N$, with N being the prediction horizon, and M the control horizon. The prediction error vector, $\widehat{\boldsymbol{x}}$, and the predicted control error vector, $\widehat{\boldsymbol{u}}$, are defined as

$$\widehat{\boldsymbol{x}} = \begin{bmatrix} \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+1) \\ \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+2) \\ \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+3) \\ \vdots \\ \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+N) \end{bmatrix}, \quad \widehat{\boldsymbol{u}} = \begin{bmatrix} \hat{\tilde{\boldsymbol{u}}}(k) \\ \hat{\tilde{\boldsymbol{u}}}(k+1) \\ \hat{\tilde{\boldsymbol{u}}}(k+2) \\ \vdots \\ \hat{\tilde{\boldsymbol{u}}}(k+M-1) \end{bmatrix}. \tag{8}$$

The predicted error is computed recursively by $\widehat{\boldsymbol{x}} = \boldsymbol{P}\hat{\boldsymbol{u}} + \boldsymbol{Q}\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k)$, where matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are given by

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{B_z} & \dots & 0 \\ \bar{\boldsymbol{A}}_{\boldsymbol{z}}\boldsymbol{B_z} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{A}}_{\boldsymbol{z}}{}^{N-1}\boldsymbol{B_z} & \dots & \bar{\boldsymbol{A}}_{\boldsymbol{z}}{}^{N-M}\boldsymbol{B_z} \end{bmatrix}, \ \boldsymbol{Q} = \begin{bmatrix} \bar{\boldsymbol{A}}_{\boldsymbol{z}} \\ \bar{\boldsymbol{A}}_{\boldsymbol{z}}{}^2 \\ \vdots \\ \bar{\boldsymbol{A}}_{\boldsymbol{z}}{}^N \end{bmatrix}.$$

As outlined in Mayne et al. (2000), in order to compute an optimal control action and guarantee stability of the closed-loop in a finite horizon, the cost function of the optimization problem can be chosen as

$$J(\tilde{\boldsymbol{u}}(k,j)) = \sum_{i=1}^{N-1} \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+i)'\, \boldsymbol{\Sigma_\rho}\, \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+i) + \tag{9}$$

$$\sum_{j=0}^{M-1} \hat{\tilde{\boldsymbol{u}}}(k+j)'\, \boldsymbol{\Sigma_\lambda}\, \hat{\tilde{\boldsymbol{u}}}(k+j) + \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+N)'\, \boldsymbol{L}\, \hat{\tilde{\boldsymbol{x}}}_{\boldsymbol{a}}(k+N),$$

where the state weighting matrix is given by $\boldsymbol{\Sigma_\rho} = \mathrm{diag}(\rho_1, \rho_2, \rho_3, ..., \rho_\mathrm{n})$, and the control weighting matrix by $\boldsymbol{\Sigma_\lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_\mathrm{m})$.

The matrix $\boldsymbol{L}$ of the terminal cost function is computed by an LQR control problem using LMIs, where the discrete

[2] # means the pseudo inverse operator of a matrix.

LTV model (6) can be represented as a linear parameter varying system. Since the matrix $\boldsymbol{A_z}(k)$ varies in time due to the evolution of the reference acceleration $\ddot{x}_r$, $\ddot{y}_r$, $\ddot{z}_r$, matrices A_i with $i = 1, \ldots, 3$, are computed at the vertices of the polytope obtained assuming the maximum and minimum values. Then, the stability condition will be guaranteed inside this polytope by the solution of the following LMI (Andrade et al. (2016)):

$$
\begin{aligned}
&\min \quad \kappa \\
&\text{subject to :} \\
&\begin{bmatrix} \kappa & 0 \\ 0 & M \end{bmatrix} > 0 \\
&\begin{bmatrix} -M & M \cdot C_z^{'} + Y^{'} \cdot D_z^{'} & M \cdot A_i^{'} + Y^{'} \cdot B^{'} \\ C_z \cdot M + D_z \cdot Y & -I & 0 \\ A_i \cdot M + B \cdot Y & 0 & -M \end{bmatrix} < 0 \\
&M > 0.
\end{aligned}
\tag{10}
$$

where matrices C_z and D_z must satisfy $C_z * D_z = 0$, $C_z * C_z' = \Sigma_\rho$, and $D_z * D_z' = \Sigma_\lambda$. Besides, the matrix M is symmetric and $M = L^{-1}$, which leads to $K = Y * M^{-1}$.

Therefore, the matrix form of equation (9) is given by

$$
\boldsymbol{J}(\widehat{\boldsymbol{u}}) = \widehat{\boldsymbol{x}}' \, \boldsymbol{W_y} \, \widehat{\boldsymbol{x}} + \widehat{\boldsymbol{u}}' \, \boldsymbol{W_u} \, \widehat{\boldsymbol{u}}, \tag{11}
$$

where $\boldsymbol{W_u}$ is an input weighting block diagonal matrix with M copies of the matrix $\boldsymbol{\Sigma_\lambda}$, and the state weighting block diagonal matrix $\boldsymbol{W_y}$ is given by $\boldsymbol{W_y} = \text{blkdiag}\,(\boldsymbol{\Sigma_\rho}, \boldsymbol{\Sigma_\rho}, \ldots, \boldsymbol{\Sigma_\rho}, \boldsymbol{L})$. The explicit solution of the above MPC formulation is obtained by applying a multi-parametric optimization algorithm to the following optimization problem

$$
\begin{aligned}
V(\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k)) = \quad & \min_{\widehat{\boldsymbol{u}}} \Big\{ \widehat{\boldsymbol{x}}' \, \boldsymbol{W_y} \, \widehat{\boldsymbol{x}} + \widehat{\boldsymbol{u}}' \, \boldsymbol{W_u} \, \widehat{\boldsymbol{u}} \Big\} \\
\text{s. t.:} \quad & \widehat{\boldsymbol{x}} = \boldsymbol{P}\hat{\boldsymbol{u}} + \boldsymbol{Q}\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k), \\
& \widehat{\boldsymbol{u}}_{\boldsymbol{min}} \le \boldsymbol{u} \le \widehat{\boldsymbol{u}}_{\boldsymbol{max}}, \\
& \widehat{\boldsymbol{x}}_{\boldsymbol{min}} \le \boldsymbol{x} \le \widehat{\boldsymbol{x}}_{\boldsymbol{max}},
\end{aligned}
\tag{12}
$$

in which $V(\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k))$ is the optimal cost and also used as a Lyapunov function for stability analysis. The vectors $\widehat{\boldsymbol{u}}_{\boldsymbol{min}}$, $\widehat{\boldsymbol{u}}_{\boldsymbol{max}}$, are the input bounds considering physical limitations of actuators as

$$
\widehat{\boldsymbol{u}}_{\boldsymbol{min}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_{\boldsymbol{min}} \\ \tilde{\boldsymbol{u}}_{\boldsymbol{min}} \\ \vdots \\ \tilde{\boldsymbol{u}}_{\boldsymbol{min}} \end{bmatrix}, \; \widehat{\boldsymbol{u}}_{\boldsymbol{max}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_{\boldsymbol{max}} \\ \tilde{\boldsymbol{u}}_{\boldsymbol{max}} \\ \vdots \\ \tilde{\boldsymbol{u}}_{\boldsymbol{max}} \end{bmatrix}, \tag{13}
$$

with $\tilde{\boldsymbol{u}}_{\boldsymbol{max}} = \boldsymbol{u_{max}} - \boldsymbol{u_{rmax}}$ and $\tilde{\boldsymbol{u}}_{\boldsymbol{min}} = \boldsymbol{u_{min}} - \boldsymbol{u_{rmin}}$. Note that $\boldsymbol{u_{max}}$ and $\boldsymbol{u_{min}}$ are the upper and lower bounds of the actuators, respectively, while $\boldsymbol{u_{rmax}}$ and $\boldsymbol{u_{rmin}}$ are the upper and lower values of the control reference. Besides, state constraints, $\widehat{\boldsymbol{x}}_{\boldsymbol{min}}$ and $\widehat{\boldsymbol{x}}_{\boldsymbol{max}}$, are considered to take into account a trajectory tracking into a confined environment, which are given by

$$
\widehat{\boldsymbol{x}}_{\boldsymbol{min}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_{\boldsymbol{min}} \\ \tilde{\boldsymbol{x}}_{\boldsymbol{min}} \\ \vdots \\ \tilde{\boldsymbol{x}}_{\boldsymbol{min}} \end{bmatrix}, \; \widehat{\boldsymbol{x}}_{\boldsymbol{max}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_{\boldsymbol{max}} \\ \tilde{\boldsymbol{x}}_{\boldsymbol{max}} \\ \vdots \\ \tilde{\boldsymbol{x}}_{\boldsymbol{max}} \end{bmatrix}, \tag{14}
$$

with $\tilde{\boldsymbol{x}}_{\boldsymbol{max}} = \boldsymbol{x_{max}} - \boldsymbol{x_{rmax}}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{min}} = \boldsymbol{x_{min}} - \boldsymbol{x_{rmin}}$, being $\boldsymbol{x_{max}}$ and $\boldsymbol{x_{min}}$ the space limits of the environment, while $\boldsymbol{x_{rmax}}$ and $\boldsymbol{x_{rmin}}$ are the maximum and minimum values of the reference trajectory. Finally, we can transform the optimization problem (12) into a parametric quadratic programming (pQP) as

$$
\begin{aligned}
&\min_{\widehat{\boldsymbol{u}}} \Big\{ \frac{1}{2} \, \widehat{\boldsymbol{u}}' \, \boldsymbol{H} \, \widehat{\boldsymbol{u}} + \tilde{\boldsymbol{x}}_{\boldsymbol{a}}' \, \boldsymbol{F} \, \widehat{\boldsymbol{u}} + \frac{1}{2} \, \tilde{\boldsymbol{x}}_{\boldsymbol{a}}' \, \boldsymbol{Y} \, \tilde{\boldsymbol{x}}_{\boldsymbol{a}} \Big\} \\
&s.t. \qquad \boldsymbol{G} \, \widehat{\boldsymbol{u}} \; \le \; \boldsymbol{W} + \boldsymbol{E} \, \tilde{\boldsymbol{x}}_{\boldsymbol{a}}
\end{aligned}
\tag{15}
$$

where $\boldsymbol{H} = \boldsymbol{P}'\boldsymbol{W_y}\boldsymbol{P} + \boldsymbol{W_u}$, $\boldsymbol{F} = \boldsymbol{Q}'\boldsymbol{W_y}\boldsymbol{P}$, $\boldsymbol{Y} = \boldsymbol{Q}'\boldsymbol{W_y}\boldsymbol{Q}$, $\boldsymbol{H} \ge 0$, and $\tilde{\boldsymbol{x}}_{\boldsymbol{a}}$ is considered as a parametric variable. Also, in order to place the constraints in the pQP form, the inequality constraints of problem (12) can be rewritten as

$$
\underbrace{\begin{bmatrix} \boldsymbol{I_m} \\ -\boldsymbol{I_m} \\ \boldsymbol{P} \\ -\boldsymbol{P} \end{bmatrix}}_{\boldsymbol{G}} \le \underbrace{\begin{bmatrix} \widehat{\boldsymbol{u}}_{\boldsymbol{max}} \\ -\widehat{\boldsymbol{u}}_{\boldsymbol{min}} \\ \widehat{\boldsymbol{x}}_{\boldsymbol{max}} \\ -\widehat{\boldsymbol{x}}_{\boldsymbol{min}} \end{bmatrix}}_{\boldsymbol{W}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{Q} \\ \boldsymbol{Q} \end{bmatrix}}_{\boldsymbol{E}} \tilde{\boldsymbol{x}}_{\boldsymbol{a}} \tag{16}
$$

where $\boldsymbol{I_m}$ is an identity matrix of appropriate dimensions. According with Jones and Morari (2006), in order to apply the multi-parametric optimization, the problem (15) is solved using the linear complementary programming (LCP) formulation, where the pQP problem is placed into the standard form using the Lagrange dual problem

$$
\begin{aligned}
&\min_{\boldsymbol{\mu}} \Big\{ \frac{1}{2} \, \boldsymbol{\mu}' \, \boldsymbol{H_d} \, \boldsymbol{\mu} + (\boldsymbol{F_d}\tilde{\boldsymbol{x}}_{\boldsymbol{a}} + \boldsymbol{c_d})' \boldsymbol{\mu} \Big\} \\
&s.t. \qquad \boldsymbol{G_d} \, \boldsymbol{\mu} \; \ge \; \boldsymbol{W_d} + \boldsymbol{E_d} \, \tilde{\boldsymbol{x}}_{\boldsymbol{a}} \\
&\qquad\qquad \boldsymbol{\mu} \ge 0.
\end{aligned}
$$

where $\boldsymbol{\mu}$ is the dual variable, $\boldsymbol{H_d}$ is a positive semi-definite matrix, and matrices $\boldsymbol{F_d}, \boldsymbol{G_d}, \boldsymbol{E_d}$ and vectors $\boldsymbol{W_d}, \boldsymbol{c_d}$ are derived from the Lagrange dual method applied to the problem (15).

Then, it is possible to write the problem in the pLCP form, applying the Karush-Kuhn-Tucker(KKT) conditions, as

$$
\begin{gathered}
\begin{bmatrix} \upsilon \\ \sigma \end{bmatrix} - \begin{bmatrix} \boldsymbol{H_d} & -\boldsymbol{G_d}' \\ \boldsymbol{G_d} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_d} \\ -\boldsymbol{E_d} \end{bmatrix} \tilde{\boldsymbol{x}}_{\boldsymbol{a}} + \begin{bmatrix} \boldsymbol{c_d} \\ -\boldsymbol{W_d} \end{bmatrix} \\
\begin{bmatrix} \upsilon \\ \sigma \end{bmatrix}' \begin{bmatrix} \boldsymbol{\mu} \\ \lambda \end{bmatrix} = 0, \quad \upsilon, \sigma, \boldsymbol{\mu}, \lambda \ge 0
\end{gathered}
$$

where $\sigma = \boldsymbol{G_d} - \boldsymbol{E_d}\tilde{\boldsymbol{x}}_{\boldsymbol{a}}$, and λ, υ are Lagrange multipliers vectors. The control inputs are given by the solution of the following multi-parametric LCP problem:

$$
\begin{aligned}
\text{find} \quad & w, z \\
s.t. \quad & w - Mz = q + \mathcal{Q}\tilde{\boldsymbol{x}}_{\boldsymbol{a}} \\
& w'z = 0 \\
& w, z \ge 0 \\
& \tilde{\boldsymbol{x}}_{\boldsymbol{a}} \in \mathcal{X}.
\end{aligned}
$$

where $w = [\upsilon \;\, \sigma]'$, $z = [\boldsymbol{\mu} \;\, \lambda]'$, $q = [\boldsymbol{c_d} \;\; -\boldsymbol{W_d}]$, $\mathcal{Q} = [\boldsymbol{F_d} \;\; -\boldsymbol{E_d}]'$ and $M = \begin{bmatrix} \boldsymbol{H_d} & -\boldsymbol{G_d}' \\ \boldsymbol{G_d} & 0 \end{bmatrix}$.

By using the algorithm presented in Jones and Morari (2006), the solution is a set of explicit affine functions in the form

$$
\tilde{\boldsymbol{u}}(k) = \begin{cases} \boldsymbol{K}_1\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) + \boldsymbol{c_{R1}} & if \quad \tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) \in CR^1 \\ \boldsymbol{K}_2\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) + \boldsymbol{c_{R2}} & if \quad \tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) \in CR^2 \\ \quad \vdots & \qquad\qquad \vdots \\ \boldsymbol{K}_s\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) + \boldsymbol{c_{Rs}} & if \quad \tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) \in CR^s \end{cases} \tag{17}
$$

where s is the number of critical regions derived from the optimization problem, matrix $\boldsymbol{K_s}$ and vector $\boldsymbol{c_s}$ are the control gains related to the critical region CR^s. Since (17) provides the error control signal at instant k, the applied control input $\boldsymbol{u}(k)$ is given by

$$\boldsymbol{u}(k) = \boldsymbol{u_r}(k) + \tilde{\boldsymbol{u}}(k). \tag{18}$$

5. IMPLEMENTATION ASPECTS

The explicit MPC was formulated using the multi-parametric toolbox (MPT3)(Herceg et al. (2013)). First, problem (15) is formulated using YALMIP, then it is converted into an MPT format and finally, it is solved using the algorithm of Jones and Morari (2006). Thus, the parametric optimization generated 3761 critical regions for the trajectory traking problem of the tilt-rotor UAV carrying a suspended load. The regions CR^i are characterized by Polyhedrons in the form of $\boldsymbol{A_R}\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k) \leq \boldsymbol{b_R}$. As presented in Figure 2, the regions are stored in a multi-dimensional matrix $\boldsymbol{CR} \in \mathbb{R}^{59\times25\times3761}$, where each slide of the matrix contains the region information which is defined as $\boldsymbol{CR}_j = [\boldsymbol{A_{Rj}}\ \boldsymbol{b_{Rj}}]$, being $\boldsymbol{A_{Rj}} \in \mathbb{R}^{59x24}$ and $\boldsymbol{b_{Rj}} \in \mathbb{R}^{59x1}$, for $j = 0, 1, .., s$. In the same way, the control laws are stored in a matrix $\boldsymbol{K_R} \in \mathbb{R}^{4\times25\times3761}$, where each slide is composed by $\boldsymbol{K_{Rj}} = [\boldsymbol{K}_j\ \boldsymbol{c_{Rj}}]$, being $\boldsymbol{K}_j \in \mathbb{R}^{4x24}$ and $\boldsymbol{c_{Rj}} \in \mathbb{R}^{4x1}$ for $j = 0, 1, .., s$.

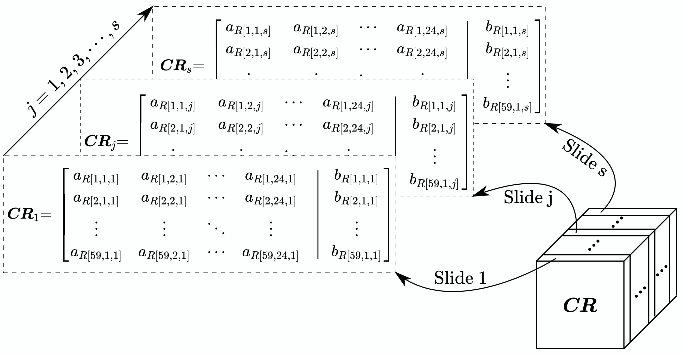

Figure 2. Critical Regions Multi-dimensional Matrix.

In order to compute the control law at each sample time, it is necessary to find the critical region where the actual state belongs.

Algorithm 1: Control Law Calculation

Input: $\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k)$
Output: $\tilde{\boldsymbol{u}}(k)$
`/* Performing Parallel Execution */`
# pragma omp parallel for;
for $j \leftarrow 0$ **to** s **by** 1 **do**
  $\boldsymbol{x_c} = [\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k)\ \ -1]$;
  $aux = \boldsymbol{CR}_j \cdot \boldsymbol{x_c}(k)$;
  if $aux \leq \varepsilon$ **then**
    $\boldsymbol{x_c} = [\tilde{\boldsymbol{x}}_{\boldsymbol{a}}(k)\ 1]$;
    $\tilde{\boldsymbol{u}}(k) = \boldsymbol{K_{Rj}} \cdot \boldsymbol{x_c}(k)$;
    break;

As presented into Algorithm 1, a concurrent program accesses each slide and performs an operation to determine if the state belongs to that region, if it is not, the program continues looking for some valid critical region. Note that, as the program uses parallel programming, the access is performed by multiple threads at the same time, so the search time is reduced.

Furthermore, for the parallel programming, the *mtimex* c library was used and compiled with OpenMP to perform multi-thread operations. Also, the whole system was simulated using MATLAB and a core i7 platform with 12GB RAM memory.

6. NUMERICAL RESULTS

With the objective of verifying the performance of the proposed eMPC formulation, some numerical experiments are performed [3]. The simulations consider the non-linear model of a tiltrotor UAV with suspended load (Almeida et al. (2014)). In addition, constant aerodynamic forces and moments are applied to the 10DOF at different time instants, in order to verify the disturbance rejection capability, as in Andrade et al. (2016).

To measure the performance of this formulation, the mean square error (MSE) of the states and the total variation of the control signals metrics are compared with the on-line MPC proposed in Andrade et al. (2016). In this context, the equilibrium point, predefined reference trajectory, MPC tuning parameters, and constraints intervals are the same as defined in Andrade et al. (2016). Additionally, the parameter ε of Algorithm 1 is settled at $1e^{-8}$.

Three different scenarios are considered: Case 1) the aircraft is simulated with nominal parameters using the set of explicit affine functions computed in the last section; Case 2) contemplates a parametric uncertainty of +30% the mass of the tiltrotor; and Case 3) the mass is assumed −30% than the nominal value. The aircraft take-off position is $\boldsymbol{q}(0) = [0.4\ 0.1\ 1\ 0_{1x7}]$.

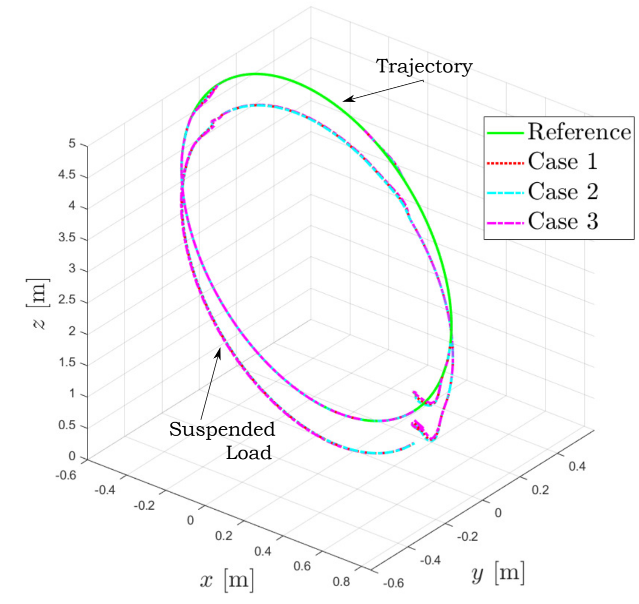

Figure 3. Trajectory performed by the tiltrotor with the suspended load using eMPC strategy.

Figure 3 illustrates the tiltrotor UAV and the suspended load trajectories performed during the simulation. Notice

[3] https://youtu.be/BBboZWrj8zU

that the eMPC has the capability to perform path tracking with null error and reject constant disturbances affecting the system. Consequently, the controller has achieved good results in the path tracking problem for all tested cases. The time response of Euler angles is shown in Figure 4. Observe that the yaw angle also presents null error when the system is disturbed, due to the integral action considered in the error of this variable. On the other side, since it is not required reference tracking in the error of the roll and pitch angles, these variables stabilize around different equilibrium points, which vary with the magnitude of the uncertainties affecting the system. Besides, one can notice that these angular positions remain stable through the entire trajectory tracking, fulfilling the imposed requirement. In Figure 5, it is depicted the behavior of the load's and servomotors' angles.

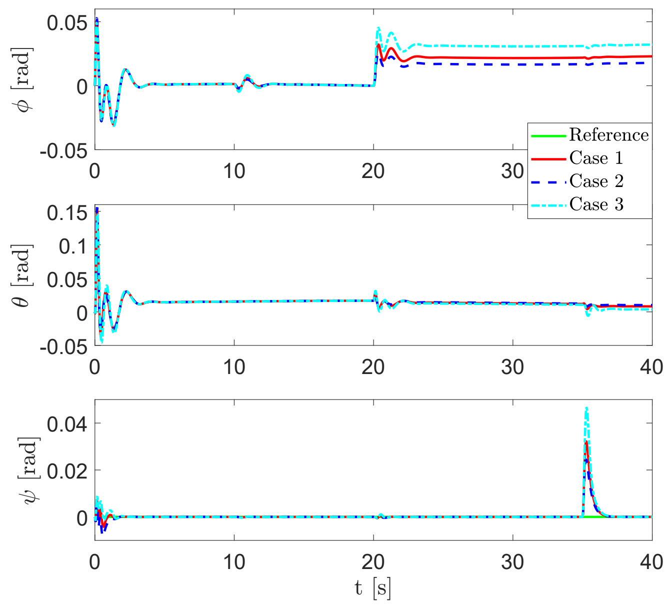

Figure 4. Euler angles of the tiltrotor using eMPC strategy

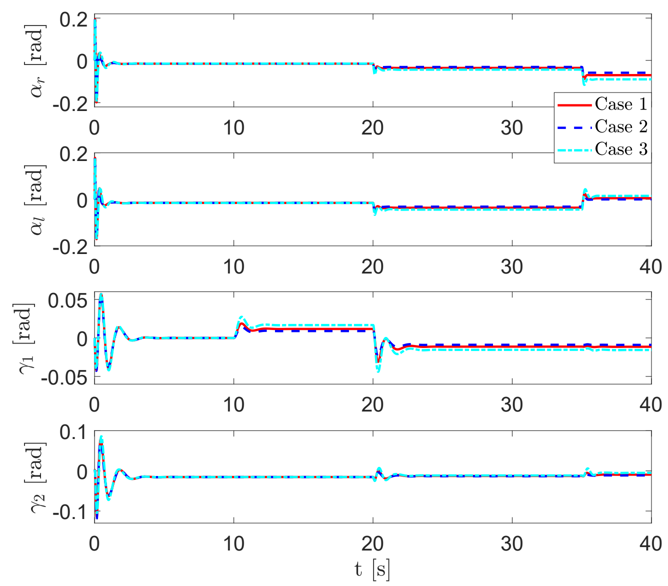

Figure 5. Servo-motor angles using eMPC strategy.

The control signals applied to the UAV throughout the trajectory tracking are illustrated in Figure 6. Analysing Case 1, the value of the thrust forces are close to the equilibrium point. However, for Case 2 the forces are greater, as the mass of the aircraft was increased. Contrary, in Case 3, the magnitude of the forces goes down, since system's mass is lower than the value used to design the controller. Note that the proposed controller was able to deal with the considered parametric uncertainty.

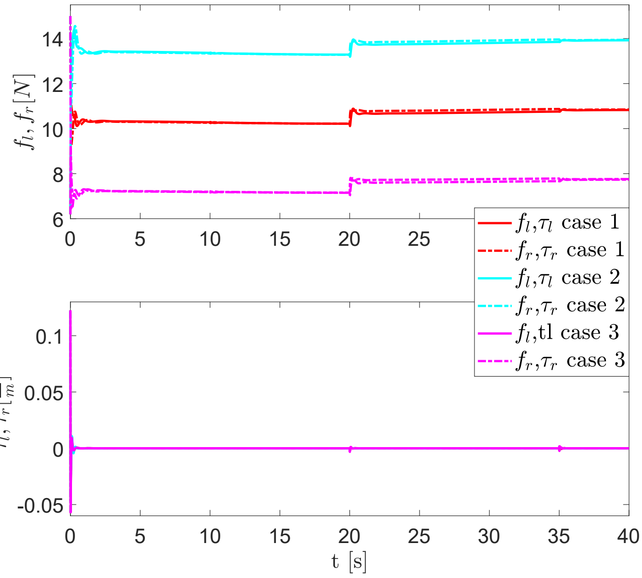

Figure 6. Control Signals applied to the aircraft using eMPC strategy.

Observing Table 1, in average, the on-line MPC presents a 12% better tracking trajectory performance than proposed eMPC, as highlighted by the MSE index. Also, the torques control signals of online MPC present smoother behavior than eMPC. Contrary, the force control signals generated by the eMPC are smoother than the on-line MPC, as presented in Table 2. In average, the eMPC control signal forces and torques are 89.27% lower and 24.13% bigger than the on-line MPC, respectively. In Table 3, the computational time of each technique is evaluated. The worst time used to compute the control signal by the on-line MPC was 11.16% greater than the eMPC.

Table 1. MSE of the outputs.

States	$eMPC$	MPC
x	0.0011	0.0011
y	11.0740	4.0519
z	3.5430	4.0452
ϕ	0.2186	1.3088
θ	0.6149	0.1265
ψ	1.8602	0.9616
α_r	2.7339	2.0276
α_l	2.9695	2.6242
γ_1	2.6769	2.6650
γ_2	2.0806	2.2941

7. CONCLUSION

In this work an explicit model predictive control was developed to solve the trajectory tracking problem for a tiltrotor UAV while carrying a suspended load. The

Table 2. Total Variation of the control signal.

Controllers	$eMPC$	MPC
fr	1195.23	9906.84
fl	929.98	9892.71
tl	0.40	0.30
tr	0.43	0.33

Table 3. Computational Time Spent on the control low calculation

time (ms)	$eMPC$	MPC
max	12.5	14.07
$average$	4.2	6.8
min	3.8	4.3

multi-parametric formulation generated 3761 critical regions. The eMPC was derived from the nominal linearized error model of the system, which was augmented with an integral action in the tracking position for performing the trajectory tracking when constant disturbances and dynamic uncertainties are presented. In order to consider real demands of actuators and states, the controller explicitly considers inputs and outputs constraints. Also, it was added a terminal cost to the OCP, aiming at ensuring stability and reducing the prediction horizon. The numerical results demonstrate that the control laws derived from the parametric optimization stabilize the UAV with the suspended load and present a successful trajectory tracking with good disturbance rejection. Moreover, using the eMPC with its implementation through parallel programming allowed to reduce in 11.16% the computational time in comparison with an on-line MPC, sacrificing in 12% the trajectory tracking performance. As future work, the formulation will be extended using robust criteria in order to guarantee robust stability and achieve better performance against uncertainties.

REFERENCES

Almeida, M.M., Donadel, R., Raffo, G.V., and B.Becker, L. (2014). Full control of a tiltrotor uav for load transportation. In *Anais do XX Congresso Brasileiro de Automática*, 2097–2104.

Andrade, R., Raffo, G.V., and Normey-Rico, J.E. (2016). Model predictive control of a tilt-rotor uav for load transportation. In *2016 European Control Conference (ECC)*, 2165–2170.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

de Almeida, M.M. and Raffo, G.V. (2015). Nonlinear control of a tiltrotor uav for load transportation. *IFAC-PapersOnLine*, 48(19), 232–237.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *2013 European Control Conference*, 502–510. Zürich, Switzerland.

Jones, C.N. and Morari, M. (2006). Multiparametric linear complementarity problems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 5687–5692.

Kouramas, K.I., Faísca, N.P., Panos, C., and Pistikopoulos, E.N. (2011). Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming. *Automatica*, 47(8), 1638–1645.

Kvasnica, M. and Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. *Proceedings of the IEEE Conference on Decision and Control*, 5270–5275.

Lee, J. and Chang, H.J. (2018). Explicit model predictive control for linear time-variant systems with application to double-lane-change maneuver. *PLoS ONE*, 13(12), 1–17.

Maddalena, E.T., Moraes, C.G.d.S., Waltrich, G., and Jones, C.N. (2019a). A Neural Network Architecture to Learn Explicit MPC Controllers from Data. URL `http://arxiv.org/abs/1911.10789`.

Maddalena, E.T., Galvão, R.K.H., and Afonso, R.J.M. (2019b). Robust region elimination for piecewise affine control laws. *Automatica*, 99, 333–337.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789 – 814.

Pistikopoulos, E.N. (2012). From multi-parametric programming theory to MPC-on-a-chip multi-scale systems applications. *Computers and Chemical Engineering*, 47, 57–66.

Raffo, G.V. and Almeida, M.d. (2018). A Load Transportation Nonlinear Control Strategy Using a Tilt-Rotor UAV. *Journal of Advanced Transportation*, 2018, 1–20.

Rego, B.S. and Raffo, G.V. (2019). Suspended load path tracking control using a tilt-rotor UAV based on zonotopic state estimation. *Journal of the Franklin Institute*, 356(4), 1695–1729.

Santos, M.A., Rego, B.S., Raffo, G.V., and Ferramosca, A. (2017). Suspended Load Path Tracking Control Strategy Using a Tilt-Rotor UAV. *Journal of Advanced Transportation*, 2017, 1–22.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489–497.

Voelker, A., Kouramas, K., and Pistikopoulos, E.N. (2013). Simultaneous design of explicit/multi-parametric constrained moving horizon estimation and robust model predictive control. *Computers and Chemical Engineering*, 54, 24–33.