25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

# COB-2019-1515
# PARAMETRIZED MODEL PREDICTIVE CONTROL TO QUADROTOR

**Tiago Santana Lourenço**
**Victor Hugo Santos Lima**
**André Murilo de Almeida Pinto**
**Renato Vilela Lopes**
Universidade de Brasília - UNB
tiagosl@gmail.com, victorhslima@hotmail.com, andremurilo@unb.br and rvlopes@unb.br

***Abstract.*** *In this paper, a model predictive controller (MPC) is proposed for a quadrotor, an unmanned aerial vehicle. A simplified prediction model is obtained by linearizing the plant's dynamics around an equilibrium point. The MPC's ability in handling operational constraints on input and output variables is explored to ensure the system stays around the equilibrium point. A control parametrization is also proposed that substantially reducing the computational cost of the quadratic programming through of an exponential linear combination applied command vector* $\mathbf{u}$*. Finally, the simulation result shows that the technique used is faster, when compared to the controller without parametrization, keeping the system stable and within the limits imposed by the constraints.*

***Keywords:*** *UAV, quadrotor, MPC, Parametrized Control.*

## 1. INTRODUCTION

The development of unmanned aerial vehicles (UAVs), leveraged by recent technological advances, has enabled the use of these platforms for several applications and through the interest of researchers in the control area. The quadrotor, a type of helicopter with four propellers, has been frequently studied for monitoring, search and rescue applications (Faessler *et al.*, 2016). In outdoor or indoor environments, this UAV model stands out for vertical take-off and landing capability and mechanical simplicity, as its actuators are only four fixed engines. Controlling your flight, however, raises one of this due to its coupled dynamics and sub-actualized nature. In addition, their dynamic model is highly non-linear introducing even more uncertainties in missions (Zulu and John, 2014).

There are several studies about the dynamics of a quadrotor and its mathematical modeling, such as those of (Kim *et al.*, 2009) and (Santana and Borges, 2009). A review of prominent control algorithms can be found in (Zulu and John, 2014), including PID, LQR, Sliding Mode, Backstepping, Adaptive and Optimal Control.

Model Predictive Control or MPC (Model Predictive Controller) is particularly interesting because it allows the imposition of constraints, delimiting a zone of operation to the states and commands of the system. In this way, the limits of actuators or operational safety are respected. The purpose of the MPC is to a sequence of commands within a defined prediction horizon so that the outputs of the system approximate desired references. This is done by minimizing a cost function, which assigns a scalar to possible command sequences. In the case where the cost function is quadratic and the imposed constraints are linear, this minimization becomes quadratic programming, which is a convex optimization problem (Wang and Boyd, 2010).

In (Raffo *et al.*, 2010), it is proposed a control method that uses the MPC to track the trajectory allied to an H to stabilize its orientation. MPC is used to stabilize and track trajectories in four-wheelers, using a discrete linear mathematical model for the prediction and imposing restrictions on the roll and pitch angles (Lopes *et al.*, 2012). The MPC is used in a fault-tolerant controller, which also relies on an Unscented Kalman Filter (UKF) for parameter estimation (Izadi *et al.*, 2011). In the latter, important caveats are also made regarding the high calculation time of the proposed algorithms, which represent an obstacle to its real-time application.

Computation time is a recognized obstacle in MPC application, keeping it limited to slow dynamics systems in general, where sampling time can be measured in seconds or minutes (Wang and Boyd, 2010). In this context, this work proposes an MPC for stabilization and trajectory tracking associated with a technique that drastically reduces its computational cost: the parameterization of the control profile. Simulations will evaluate the performance and reduction of the computational cost of a MPC associated with this technique.

This article is organized as follows: Section 2 gives a description of the system and its dynamic modeling; Then, in section 3, the MPC formulation and the optimization problem; In section 4, the exponential parameterization of the command sequence is defined; the results of the simulations and an analysis of the calculation times are presented in section 5; Finally, in section 6, the conclusions are made.

## 2. DYNAMIC MODELING AND SYSTEM DESCRIPTION

The flight of quadrotors results from the force and momentum of their rotors, by controlling the speeds of the four engines $(\omega_1, \omega_2, \omega_3, \omega_4)$. The rotors are the only actuators in this system and, at each instant, exert a propulsion force $f_i$ and a torque $\tau M_i$ in the body of the UAV, as shown in Fig. 1. Each pair of opposing rotors rotates in the same direction so that, when all spin at the same speed, the torques cancel. On the other hand, when the torques don't cancel, angular acceleration is induced around its vertical axis, allowing the control of the yaw $(\psi)$. The longitudinal movement is obtained by varying the velocities $\omega_1$ and $\omega_3$, while the lateral movement is obtained by varying the velocities $\omega_2$ and $\omega_4$.

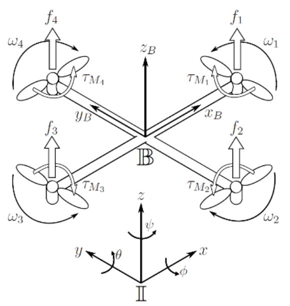


Figure 1: Adapted of (Luukkonen, 2011)

.

For mathematical modeling, the quadrotor is considered a rigid body with a reference $(\mathbb{B})$ fixed to its center of mass, which is equidistant from the four rotors. The generalized coordinates of the system are $q = (x, y, z, \phi, \theta, \psi)$, where $(x, y, z)$ represent the displacement of the center of mass of the UAV relative to the inertial frame $\mathbb{I}$ and $(\phi, \theta, \psi)$ are the angles of Euler that represent the orientation of $\mathbb{B}$ with respect to $\mathbb{I}$. The differential equations are developed using the Euler-Lagrange formalism, as shown in detail by (Santana and Borges, 2009). The obtained nonlinear model can be written, for simplicity, as:

$$\dot{\xi}_c = f(\xi_c, u_c) \tag{1}$$

where $\boldsymbol{\xi}_c = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^T$ is the state vector of the state system and $\boldsymbol{u_c} = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$ is the vector of commands, being $\omega_i$ the angle velocity of i-th rotor. In this work, will be use MPC formulation based on a linearized and discretized model, which will result in a convex optimization. This model is obtained through a Taylor expansion of Eq. (1) around a point of equilibrium $\left(\bar{\xi}, \bar{u}\right)$. Then, by defining a sampling time $\tau_s$, the discrete model is written in the states space:

$$\boldsymbol{\xi}(k+1) = \mathbf{A}\boldsymbol{\xi}(k) + \mathbf{B}\mathbf{u}(k) \tag{2}$$

where $\xi \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n \times n}$ is the state matrix and $B \in \mathbb{R}^{n \times n_u}$ is the input matrix. For the quadrotor, $n = 12$ and $n_u = 4$. Matrices can be found in (Lopes *et al.*, 2012).

The MPC based on a linear model invokes a convex optimization problem. The disadvantage is that the prediction will only be reliable for a certain region of operation around the equilibrium point $\left(\bar{\xi}, \bar{u}\right)$. Without adequate prediction, MPC stability is severely impaired. This gives more importance to the stipulation of constraints, which are formulated in such a way that the system remains in an operating zone close to $\left(\bar{\xi}, \bar{u}\right)$, here adopted as $\bar{\boldsymbol{\xi}} = [0\ 0\ 0\ 0\ 10\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and $\bar{\boldsymbol{u}} = [\bar{\omega}\ \bar{\omega}\ \bar{\omega}\ \bar{\omega}]^T$. This coincides with the condition of hovering at 10 m in height, standing in relation to an inertial reference. For such, the UAV remains horizontal with the rotors rotating at a constant $\bar{\omega}$.

## 3. MODEL PREDICTIVE CONTROL

The prediction model adopted is allied to the formulation of a cost function to evaluate which control sequences best meet the control objectives. This sets up an optimization problem whose solution is an optimal sequence of commands, which approximates the states of a desired reference within the prediction horizon. However, only the first command in the sequence is applied during the acquisition period of the sensors. With a state vector measurement, the optimization problem is solved again and the process repeats.

In this work, the prediction is based on the linearized model of Eq. (2) applied until the end of the prediction horizon. This means that, since one has a sequence of commands $\widetilde{\boldsymbol{u}}$ and the current state $\boldsymbol{\xi}\,(k)$, one can predict the sequence of future states $\widetilde{\boldsymbol{\xi}}$. These sequences follow the following definition:

$$\tilde{u} \triangleq \begin{pmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{pmatrix}, \tilde{\xi} \triangleq \begin{pmatrix} \xi(k) \\ \xi(k+1) \\ \vdots \\ \xi(k+N) \end{pmatrix}$$

where $N$ represents the prediction horizon, $\widetilde{u} \in \mathbb{R}^{N \cdot n_u}$ and $\widetilde{\xi} \in \mathbb{R}^{N \cdot n}$. Thus, by applying Eq. (2) $N$ times, the prediction can be expressed by the equation:

$$\tilde{\boldsymbol{\xi}} = \begin{pmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_n \end{pmatrix} \boldsymbol{\xi}(k) + \begin{pmatrix} \boldsymbol{\Psi}_1 \\ \vdots \\ \boldsymbol{\Psi}_n \end{pmatrix} \tilde{\boldsymbol{u}}, \tag{3}$$

where the matrices $\boldsymbol{\Phi_i}$ and $\boldsymbol{\Psi_i}$ are constants and depend only on the model, it can be shown that they are:

$$\Phi_i = [A^i] \in \mathbb{R}^{n \times n} \tag{4}$$

$$\Psi_i = [A^{i-1}B, \ldots, AB, B] \begin{pmatrix} \prod_1^{(n_u,N)} \\ \prod_2^{(n_u,N)} \\ \vdots \\ \prod_i^{(n_u,N)} \end{pmatrix} \in \mathbb{R}^{n \times i \cdot n_u} \tag{5}$$

The operator $\prod_i^{(j,M)}$ removes the i-th vector $\boldsymbol{x}(k+i) \in \mathbb{R}^j$ from any concatenation of $M$ vectors $\widetilde{x} \in \mathbb{R}^{M \cdot j}$. So that $\boldsymbol{x}\,(k+i) = \prod_i^{(j,M)} \widetilde{\boldsymbol{x}}$.

The optimum sequence of commands $\widetilde{\boldsymbol{u}}^{opt}$ is the solution of an optimization problem which, by minimizing a cost function, seeks to approximate the states of the system $\boldsymbol{\xi}(k+i)$ of its desired trajectory $\widetilde{\boldsymbol{\xi}}^d(k+i)$ to $i = [1, \ldots, N]$. However, since it is not desirable to assign trajectories to all n states of the system, it is convenient to define a vector of regulated outputs $y_r \in \mathbb{R}^{n_r}$:

$$\boldsymbol{y_r}(k+i) = \boldsymbol{C_r}\boldsymbol{\xi}(k+i), \tag{6}$$

where $C_r \in \mathbb{R}^{n_r \times n}$. To this set of outputs are imposed the desired references and $\boldsymbol{y_r^d}(k)$ for trajectory tracking.

### 3.1 The Cost Function

The objective of the control is that the regulated states $\boldsymbol{y_r}(k)$ approach the maximum of their desired trajectories $\boldsymbol{y_r^d}(k)$, also minimizing the deviation of the applied commands $\boldsymbol{u}\,(k)$ with respect to their desired operating point $\bar{\boldsymbol{u}}$. The cost function that must be minimized to achieve this goal is defined as:

$$J(\tilde{u}) \triangleq \sum_{i=1}^{N} \parallel y_r(k+i) - y_r^d(k+i) \parallel_{Q_y}^2 + \sum_{i=1}^{N} \parallel u(k+i-1) - \tilde{u} \parallel_{Q_u}^2, \tag{7}$$

where the notation $\|x\|_Q^2$ represents the scalar defined by $\|x\|_Q^2 \triangleq x^T Q x$ for any vector $x \in \mathbb{R}^j$ and a matrix $Q \in \mathbb{R}^{j \times j}$. Therefore, the matrices $Q_y$ and $Q_u$ are the weighting matrices and defined according to the control objectives. Substituting Eq. (3) and (6) into (7) and expanding the norms, we can rewrite cost function in standardized quadratic format:

$$J(\tilde{u}) = \frac{1}{2}\tilde{u}^T H \tilde{u} + F^T(k)\tilde{u} \tag{8}$$

where,

$$F(k) = F_1\xi(k) + F_2\tilde{y}_r^d + F_3\tilde{u}, \tag{9}$$

where the matrices $H$, $F_1$, $F_2$ and $F_3$ are constant and defined as:

$$H \triangleq 2\sum_{i=1}^{N} \left[ \Psi_i^T C_r^T Q_y C_r \Psi_i + \left( \prod_i^{(n_u,N)} \right)^T Q_u \left( \prod_i^{(n_u,N)} \right) \right] \tag{10}$$

$$F_1 \triangleq 2\sum_{i=1}^{N}\left[\Psi_i^T C_r^T Q_y C_r \Phi_i\right] \tag{11}$$

$$F_2 \triangleq -2\sum_{i=1}^{N}\left[\Psi_i^T C_r^T Q_y \prod_{i}^{(n_r,N)}\right] \tag{12}$$

$$F_3 \triangleq 2\sum_{i=1}^{N}\left[\left(\prod_{i}^{(n_u,N)}\right)^T Q_u\right] \tag{13}$$

### 3.2 The Constraints

The imposition of restrictions is an important advantage of MPC's strategy. As was pointed out above, it is important to keep the system close to the equilibrium point, in a region that holds the linear mathematical model valid. This is done by stipulating constraints on a set of system outputs. These restricted outputs are defined as $y_c(k) = C_c\xi(k) \in \mathbb{R}^{n_c}$, where $C_c \in \mathbb{R}^{n_c\times n}$. These outputs must satisfy the set of restrictions $y_c^{min} \leq y_c(k+i) \leq y_c^{max}$, for $i = [1, \ldots, N]$, being able to be written:

$$\begin{pmatrix} y_c^{min} \\ \vdots \\ y_c^{min} \end{pmatrix} \leqslant \begin{pmatrix} C_c\prod_1^{n,N} \\ \vdots \\ C_c\prod_N^{n,N} \end{pmatrix}\tilde{\xi} + \begin{pmatrix} y_c^{max} \\ \vdots \\ y_c^{max} \end{pmatrix}, \tag{14}$$

where $y_c^{min} \in \mathbb{R}^{n_c}$ and $y_c^{max} \in \mathbb{R}^{n_c}$ are respectively their minimum and maximum values determined. Substituting Eq. (3) into Eq. (14) and doing some manipulations, the inequalities can be rewritten in the system of $2N \cdot n_c$ linear inequalities:

$$\underbrace{\begin{pmatrix} C_c\Psi_1 \\ \vdots \\ C_c\Psi_N \\ -C_c\Psi_1 \\ \vdots \\ -C_c\Psi_N \end{pmatrix}}_{A_{ineq}^{(1)}}\tilde{u} \leqslant \underbrace{\begin{pmatrix} -C_c\Phi_1 \\ \vdots \\ -C_c\Phi_N \\ C_c\Phi_1 \\ \vdots \\ C_c\Phi_N \end{pmatrix}}_{G_1^{(1)}}\xi(k) + \underbrace{\begin{pmatrix} y_c^{max} \\ \vdots \\ y_c^{max} \\ -y_c^{min} \\ \vdots \\ -y_c^{min} \end{pmatrix}}_{G_1^{(3)}} \tag{15}$$

Then, restrictions are imposed on the rate of change of the commands in the form of the inequalities: $\delta_{min} \leq u(k+i) - u(k+i-1) \leq \delta_{max}$, where $\delta_{min}$ and $\delta_{max}$ are the variation limits of the command vector in a sampling period $\tau_s$. Using $\mathbb{I}_{n_u} \in \mathbb{R}^{n_u\times n_u}$ as the identity matrix and $\mathbb{O}_{n_u} \in \mathbb{R}^{n_u\times n_u}$ as the null matrix, we rewrite these inequalities for the entire prediction horizon as:

$$\underbrace{\begin{pmatrix} \mathbb{I}_{n_u} & \mathbb{O}_{n_u} & \cdots & \mathbb{O}_{n_u} & \mathbb{O}_{n_u} \\ -\mathbb{I}_{n_u} & \mathbb{I}_{n_u} & \mathbb{O}_{n_u} & \cdots & \mathbb{O}_{n_u} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbb{O}_{n_u} & \mathbb{O}_{n_u} & \cdots & -\mathbb{I}_{n_u} & \mathbb{I}_{n_u} \\ -\mathbb{I}_{n_u} & \mathbb{O}_{n_u} & \mathbb{O}_{n_u} & \cdots & \mathbb{O}_{n_u} \\ \mathbb{I}_{n_u} & -\mathbb{I}_{n_u} & \mathbb{O}_{n_u} & \cdots & \mathbb{O}_{n_u} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbb{O}_{n_u} & \mathbb{O}_{n_u} & \cdots & \mathbb{I}_{n_u} & -\mathbb{I}_{n_u} \end{pmatrix}}_{A_{ineq}^{(2)}}\tilde{u} \leqslant \underbrace{\begin{pmatrix} \mathbb{I}_{n_u} \\ \mathbb{O}_{n_u} \\ \vdots \\ \mathbb{O}_{n_u} \\ -\mathbb{I}_{n_u} \\ \mathbb{O}_{n_u} \\ \vdots \\ \mathbb{O}_{n_u} \end{pmatrix}}_{G_2^{(2)}} u(k-1) + \underbrace{\begin{pmatrix} \delta_{max} \\ \delta_{max} \\ \vdots \\ \delta_{max} \\ -\delta_{min} \\ -\delta_{min} \\ \vdots \\ -\delta_{min} \end{pmatrix}}_{G_3^{(2)}} \tag{16}$$

Inequalities (15) and (16) can be combined and rewritten as:

$$A_{ineq}\tilde{u} \leqslant \underbrace{G_1\xi(k) + G_2 u(k+1) + G_3}_{B_{ineq}(k)} \tag{17}$$

where the following definitions have been used, from the terms in (15) and (16):

$$A_{ineq} \triangleq \begin{pmatrix} A_{ineq}^{(1)} \\ A_{ineq}^{(2)} \end{pmatrix}; G_1 \triangleq \begin{pmatrix} G_1^{(1)} \\ \mathbb{O}_{(2Nn_u)\times n} \end{pmatrix} \tag{18}$$

$$G_2 \triangleq \begin{pmatrix} \mathbb{O}_{(2Nn_c)\times n_u} \\ G_2^{(2)} \end{pmatrix}; G_3 \triangleq \begin{pmatrix} G_3^{(1)} \\ G_3^{(2)} \end{pmatrix} \tag{19}$$

Finally, constraints are imposed on the command vector in the form: $u_{min} \leq u\,(k+i) \leq u_{max}$, where $u_{min} \in \mathbb{R}^{n_u}$ and $u_{max} \in \mathbb{R}^{n_u}$ delimit the minimum and maximum commands applicable by the actuators. Rewriting for all instants $k + i$ for $i \in [1, \ldots, N]$, obtain:

$$\tilde{u}_{min} \leqslant \tilde{u} \leqslant \tilde{u}_{max} \tag{20}$$

The constraints imposed on the system commands are important because, in addition to keeping the operation close to equilibrium point as discussed above, they also serve to ensure that the commands planned by the controller respect the limits of the system actuators. With this, the controller will search for solutions that are within a set of sequences of commands that are realizable by the actuators.

### 3.3 Quadratic Programming

The quadratic cost function (8) together with the linear constraints (17) and (20), define a quadratic programming problem (QP). This optimization problem consists of calculating a $\tilde{\boldsymbol{u}}^{\boldsymbol{opt}}$ vector that minimizes the cost function and respects the restrictions imposed, that is:

$$\tilde{\boldsymbol{u}}^{opt}(k) \triangleq argmin : \left[\frac{1}{2}\tilde{\boldsymbol{u}}^T \boldsymbol{H} \tilde{\boldsymbol{u}} + \boldsymbol{F}^T(k)\tilde{\boldsymbol{u}}\right] \tag{21}$$

There is a considerable repertoire of efficient methods for solving QPs, such as those presented in (Wang and Boyd, 2010).

The calculation time for the QP solution is directly impacted by the vector size $\widetilde{u} \in \mathbb{R}^{N \cdot n_u}$. This generates a conflict of choice since an increase in the N prediction horizon, while reinforcing the condition of closed-loop stability, requires a greater computational effort. It is important to note that the conditions for predictive control stability go beyond an appropriate choice of the prediction horizon. An extensive discussion on the topic can be found in (Mayne *et al.*, 2000). A possible solution to the constraints arising from the high computational cost is to reduce the number of variables needed to define the $\widetilde{\boldsymbol{u}}$ command sequence. In this work this is done through a parameterization of the control sequence, as proposed in (Alamir, 2013).

## 4. EXPONENTIAL PARAMETRIZATION

The parameterization of control allows to express the sequence $\widetilde{\boldsymbol{u}}$ through a vector of reduced size $p \in \mathbb{R}^{n_p}$. This can be done in several ways, in the exponential parameterization each component of the command vector $\boldsymbol{u}$ is parameterized by a linear combination of exponentials:

$$\boldsymbol{u_j}(k+i) \triangleq \sum_{l=1}^{n_e^j} \left[e^{\lambda_l^j \cdot (i\tau_s)}\right] \cdot p_l^j,\, \lambda_l^j \triangleq \frac{-2}{\alpha^j[(l-1)\beta+1]}, \tag{22}$$

where $u_j\ \forall\ j \in [1, \ldots, n_u]$ j-th component of the command vector and $n_e^j$ e is the number of exponentials chosen to parameterize $u_j$. The component $\lambda_l^j$ is defined as:

$$\lambda_l^j \triangleq \frac{-2}{\alpha^j[(l-1)\beta+1]}, \tag{23}$$

where $\alpha^j$ is the settling time of the j-th actuator and $\beta > 1$ is a constant. Notice that, with this, the command is completely determined by the parameter $p_l^j$ that composes a new vector:

$$p \triangleq \begin{pmatrix} p^1 \in \mathbb{R}^{n_e^1} \\ \vdots \\ p^{n_u} \in \mathbb{R}^{n_e^{nu}} \end{pmatrix} \in \mathbb{R}^{n_p} \tag{24}$$

The size of the vector $\boldsymbol{p}$ depends on the number of exponentials chosen for each actuator $\left(n_e^j\right)$, so that $n_p = \sum_{j=1}^{n_u} n_e^j$.

The parameterization defined in Eq. (22) can be concatenated for all instants $(k+i)\ \forall\, i \in [1,\dots,N]$, obtaining the expression that generates a sequence of $\widetilde{\boldsymbol{u}}$ commands from the vector $\boldsymbol{p}$. This allows us to rewrite the QP in terms of the vector $\boldsymbol{p}$, such as:

$$\boldsymbol{p^{opt}}(k) \triangleq argmin : \left[\frac{1}{2}\boldsymbol{p}^T\boldsymbol{H_r}\boldsymbol{p} + \boldsymbol{F_r}^T(k)\boldsymbol{p}\right],\ subject\ to : A_r p \leqslant B_r(k) \tag{25}$$

This reformulation reduces the size of the solution from QP de $N \cdot n_u$ to $n_p$, which can drastically reduce the solution time of the problem without reducing the prediction horizon $N$.

## 5. SIMULATION RESULTS

The performance of the predictive controller, with and without parametrization, is investigated and compared through simulations. The nonlinear model (expressed in Equation (1)) is used to simulate more accurately the real dynamics of the quadrotor. Thus, for the same trajectory and operational restrictions, the calculation times are compared, as well as their performance in tracing the trajectory.

The QP solution is obtained by qpOASES, a software package available in (Ferreau *et al.*, 2007–2017). QpOASES applies a strategy recently proposed by (Ferreau *et al.*, 2014), which makes it particularly suitable for applications with predictive control. The simulation was run on a Windows 10 PC with an Intel Core i7 2,4 GHz and 16 GB RAM. The compilation was done in the Simulink environment of Matlab 2014a. Defining a sampling time $\tau_s = 0,05\,s$ and a prediction horizon $N = 50$. The regulated outputs of the quadrotor are $x,\ y,\ z$ and $\psi$, so that:

$$C_r \triangleq \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The restricted outputs are the roll and pitch angles ($\phi$ and $\theta$), defined by:

$$C_c \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The weighting matrices are:

$$Q_y \triangleq \begin{pmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix}, Q_u \triangleq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

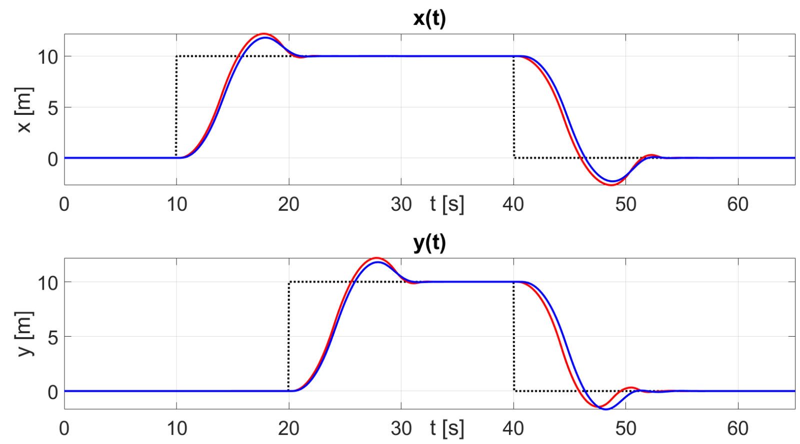
A restriction on the amplitude of these angles is imposed so that the UAV remains close to the equilibrium point of flight regime discussed above. For the simulations, $y_c^{min} = [-5° \ \ -5°]^T$ and $y_c^{max} = [5° \ \ 5°]^T$ are imposed. Since the motors of the quadrotor are unable to reverse their direction of rotation, they are imposed: $u_{min} = [0\,0\,0\,0]^T$ and $u_{max} = [\omega_{max}\,\omega_{max}\,\omega_{max}\,\omega_{max}]^T$, where $\omega_{max} = 250\ rad/s$ is the maximum angular speed that the rotors must attain. For the MPC with parameterization of the control sequence, two exponentials were used for all control variables, $n_j^e = 2$ for all $j = [1,\dots,4]$. The settling times of the motors are $\alpha^j = 0,1\ s\ \forall\, j = [1,\dots,4]$.

Figures 2a present the results of the outputs $x$ and $y$ with the application of the step path of 10 meters in 10 and 20 seconds respectively and returning to 0 in 40 seconds. In Fig. 2b present the results of the outputs $z$ and $\psi$, applying in $z$ a step from 10 meters to 0 in 40 seconds and yaw angle always at zero. The controller MPC without parametrization (red lines) and MPC Parametrized (blue lines), for the same trajectory (dotted lines), present very similar performances and reaching the trajectory just above 10 seconds.

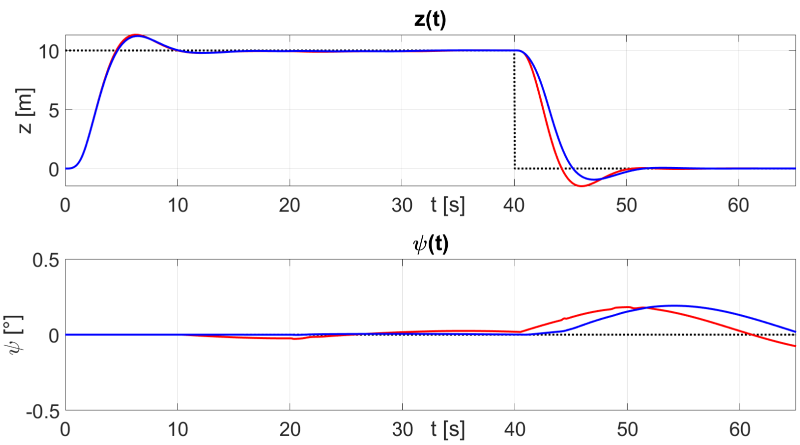
Next, Fig. 3 presents the roll and pitch angles with the same line configuration shown in the previous figure. Here, the restrictions can be observed as being respected and the behavior in both algorithms are again quite similar.

Figure 4 shows the calculation times of each iteration during the simulation of each controller. Here, the differential of the MPC with the parameterization of the control profile is evident, presenting an average calculation time of $3,7\,ms$, while the MPC without parametrization took on average $1,3\,s$ to complete an iteration. As can be seen, where step rungs occur at 10, 20 and 40 seconds, there is a high rate of change in calculation time on both controllers, but the parameterized controller has considerably shorter working time than the controller without parameterization.

Finally, in Fig. 5, the commands applied during the simulation are presented, where again the restrictions imposed are respected and both controllers present a similar profile.




(a) Position $x$ and $y$.




(b) Position $z$ and angle $\psi$.

Figure 2: Results obtained by the MPC algorithms(blue line - Parametrized MPC, red line - MPC and dotted line - tracking).

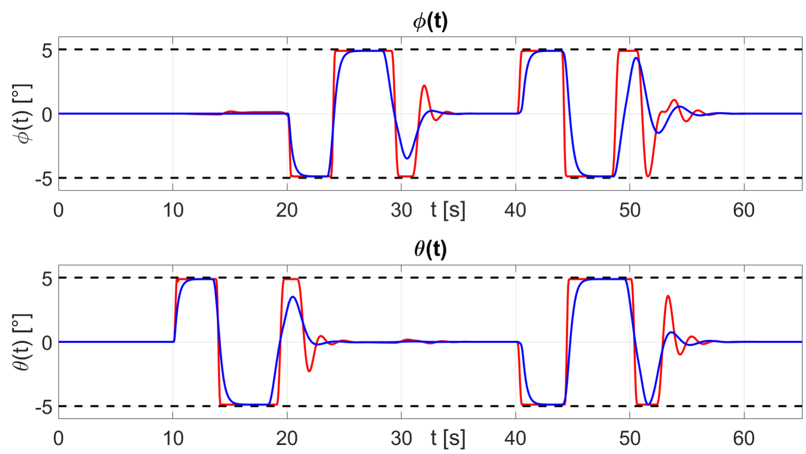



Figure 3: Angle $\phi$ and $\theta$. Results obtained by the MPC algorithms(blue line - Parametrized MPC, red line - MPC and dashed line - constraint).

.

## 6. CONCLUSION

In this paper, the exponential parametrization to MPC was presented to reduce the size of the solution for QP with the algorithm of parametrization proposed by (Alamir, 2013) and applied to the quadrotor, which has a coupled dynamic and

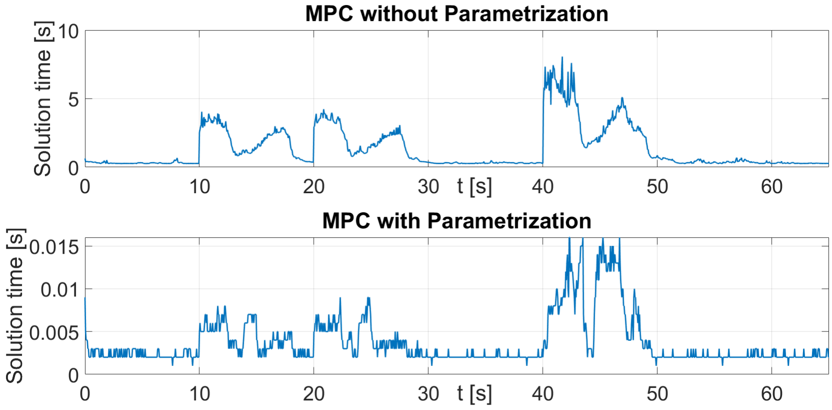



Figure 4: Results of the solution times of each iteration.

.

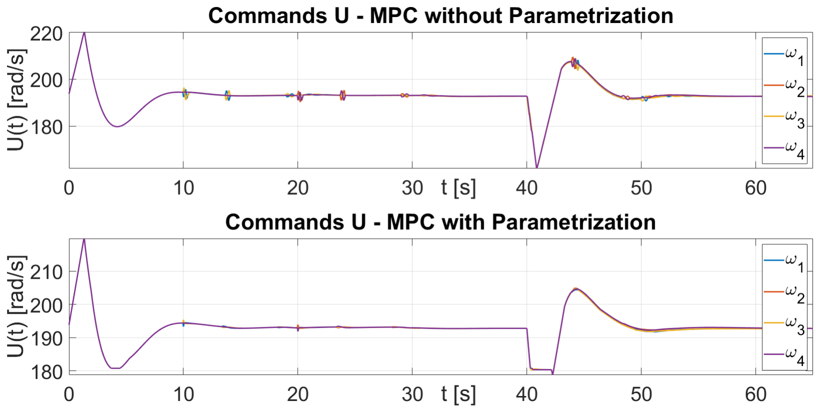



Figure 5: The commands applied by the controllers.

.

underactuated. The results show that this technique reduces the peak solution time around 7,5 s without parametrization for 15 ms peaks when parametrizing the MPC and reducing around 99, 7% average time, satisfying trajectory tracking in $x, y, z$ and $\psi$ and obeying the restrictions imposed on the angles $\phi$ and $\theta$ as well as the rotor angular speeds. Thus, through simulated results, suggest that this technique keeps the system stable and an alternative to embedded systems.

## 7. ACKNOWLEDGEMENTS

The authors thank the National Council for Scientific and Technological Development (CNPq), the Federal District Research Support Foundation (FAP-DF), University of Brasília - Gama College (UnB-FGA) and CAPES for their partial support to research activities.

## 8. REFERENCES

Alamir, M., 2013. *A Pragmatic Story of Model Predictive Control: Self-Contained Algorithms and Case-Studies*. CreateSpace Independent Publishing Platform, 1st edition.

Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M. and Scaramuzza, D., 2016. "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle". *J. Field Robotics*, Vol. 33, pp. 431–450.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G. and Diehl, M., 2014. "qpoases: a parametric active-set algorithm for quadratic programming". *Mathematical Programming Computation*, Vol. 6, No. 4, pp. 327–363.

Ferreau, H., Potschka, A. and Kirches, C., 2007–2017. "qpOASES webpage". http://www.qpOASES.org/.

Izadi, H.A., Zhang, Y. and Gordon, B.W., 2011. "Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation". *IFAC Proceedings Volumes*, Vol. 44, No. 1, pp. 6343 – 6348.

Kim, J., Kang, M.S. and Park, S., 2009. "Accurate modeling and robust hovering control for a quad–rotor vtol aircraft". *Journal of Intelligent and Robotic Systems*, Vol. 57, No. 1, p. 9. ISSN 1573-0409. doi:10.1007/s10846-009-9369-z. URL `https://doi.org/10.1007/s10846-009-9369-z`.

Lopes, R., Santana, P., Borges, G. and Ishihara, J., 2012. "Model predictive control applied to tracking and attitude stabilization of a vtol quadrotor aircraft". In *ABCM Symposium Series in Mechatronics*. Vol. 5.

Luukkonen, T., 2011. *Modelling and control of quadcopter*. Independent research project in applied mathematics, Espoo, Finland.

Mayne, D., Rawlings, J., Rao, C. and Scokaert, P., 2000. "Constrained model predictive control: Stability and optimality". *Automatica*, Vol. 36, No. 6, pp. 789 – 814.

Raffo, G., G. Ortega, M. and Rubio, F., 2010. "An integral predictive/nonlinear $H_\infty$ control structure for a quadrotor helicopter". *Automatica*, Vol. 46, pp. 29–39.

Santana, P. and Borges, G., 2009. "Modelagem e controle de quadrirrotores". In *IX Simpósio Brasileiro de Automação Inteligente (SBAI 2009)*. Brasília, Brazil.

Wang, Y. and Boyd, S., 2010. "Fast model predictive control using online optimization". *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 2, pp. 267–278.

Zulu, A. and John, S., 2014. "A review of control algorithms for autonomous quadrotors". *Open Journal of Applied Sciences*, Vol. 04, pp. 547–556.

## 9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.