Model Predictive Control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft

Renato Vilela Lopes, lopes@lara.unb.br
Pedro Henrique de Rodrigues Quemel e Assis Santana, phrqas@ieee.org
Geovany Araújo Borges, gaborges@unb.br
João Yoshiyuki Ishihara, ishihara@ene.unb.br
Automation and Robotics Laboratory (LARA), Department of Electrical Engineering, University of Brasília, Brasília, DF, Brasil.

Abstract. *In this paper we present a new control approach for a quadrotor mini-helicopter using the model predictive control (MPC) technique. The formulation employed is based on a linear prediction model obtained by linearizing the plant's dynamics around an equilibrium point. The optimal control sequence is implemented in a receding-horizon strategy. The optimization is repeated at each sampling instant, taking into account the new sensor readings. The MPC's ability in handling operational constraints on input and output variables is explored to ensure position control and overall system stabilization with a single layer control design. This control loop with only one predictive controller is the main contribution of this work. Various simulations of a quadrotor show the good performance of the proposed control law and simulation results are compared with conventional linear PID control and nonlinear backstepping control.*

Keywords: *predictive control, quadrotor, constraint relaxation, constrained control.*

1. INTRODUCTION

Developing autonomous miniature flying robots for indoor and outdoor operations is among the great challenges in robotics research (Becker *et al.* (2006)), (Adigbli *et al.* (2007)), (Tayebi and McGilvray (2006)). In fact, great difficulties arise in these vehicles' design for operation in unknown environments, mainly because of the necessity for lighter, smaller systems with cost and power consumption restrictions, which implies performance limitations.

Currently, the interest for vertical take-off and landing (VTOL) vehicles is growing, since their ability to hover flight and their lack of necessity for runways make them well-suited options for supervision, inspection, and work in environments where space is limited and high maneuverability is required (Bouabdallah (2007)). In this context, the quadrotor shows itself to be advantageous in terms of applicability for its simplified mechanics, stable hover flight, and high payloads. However, because quadrotors are nonlinear, multivariable and underactuated mechatronic systems, the problems of stabilization and position control become challenging.

The quadrotor is said to be an underactuated system because its dynamical model has six outputs (x, y, z, ϕ, θ, ψ) and only four independent inputs (each one of its rotors). Therefore, it is not possible to control all of the states at the same time (Madani and Benallegue (2006b)). In this scenario, a good controller should be able to reach a desired Cartesian position (x, y, z) and a desired yaw angle (ψ) while keeping the stabilization of the pitch (θ) and roll (ϕ) angles (Madani and Benallegue (2006a)).

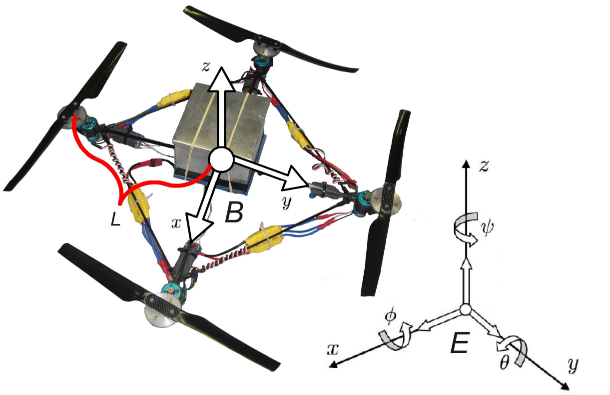
A considerable amount of effort has been invested in controlling quadrotor helicopters and several control strategies have been tested. Feedback linearization method was first used by Mistler *et al.* (2001) to make a quadrotor track a reference trajectory. Bouabdallah *et al.* (2004) compared the performance of two model-based control techniques for stabilization of this kind of helicopter: a classical PID approach, which assumed simplified dynamics, and a modern LQR technique based on a more complete model. Castillo et al. applied nested saturations control to specify a quadrotor's position and stabilize its attitude (Castillo *et al.* (2004a)), (Castillo *et al.* (2004b)). In (Xu and Ozguner (2006)), a sliding mode controller is proposed to stabilize a class of cascaded underactuated systems. Another common approach to quadrotor control is the backstepping technique. Bouabdallah and Siegwart (2005) proposed a backstepping controller using a quadrotor's simplified model and a special decomposition of the control law. Madani and Benallegue used traditional backstepping (Madani and Benallegue (2006a)) and Full State Backstepping (Madani and Benallegue (2006b)) techniques having in mind that the quadrotor can be seen as three interconnected subsystems.

More recently, two papers presented interesting approaches using a model predictive controller (MPC) in the control loop. In Raffo *et al.* (2008), the control structure consists of a MPC to track the reference trajectory and a nonlinear H∞ controller to stabilize the rotational movements. A similar approach is used in (Alexis *et al.* (2010)), where a MPC is designed to establish a robust control law for position, while a feedforward controller performs the aircraft's stabilization. The MPC technique consists of solving a moving-horizon optimal control problem. Such a solution is reiterated at periodic intervals (normally, at each sampling period) based on sensor feedback information (Camacho and Bordons (1999)). The main features that have contributed towards the popularity of predictive controllers, according to Maciejowski (2002) and Vada *et al.* (2001), are the ability to cope with transport delays and large numbers of controlled and manipulated variables, as well as the enforcement of operational constraints, which is of value to reduce the number of emergency stops and system downtime. Because of their constraint handling feature, predictive controllers are a promising option to ensure a safe operation in quadrotor control (Maciejowski (2002)), since MPC is able to increase the flight envelope, without compromising system reliability, even if the actuators employed operate close to their saturation limits.

In all referred papers, the control loop uses two controllers: the MPC for position control and a second controller for stabilization. In this paper, a new approach is presented using a configuration with only one controller. The MPC's ability to incorporate operational constraints on input and output variables is explored to ensure position control and overall system stabilization. A state-space MPC formulation is used, which is based on a linear prediction model obtained from linearization of a quadrotor's nominal physical model around an equilibrium point. Simulation results evaluate the MPC's performance using the complete model of a quadrotor's dynamics.

2. System Description and Modelling

The object of study in this work is a quadrotor helicopter, which consists of four propulsion rotors, usually arranged in a cross configuration, fixed at the ends of a rigid body, as shown in Figure 1. The illustrated quadrotor is currently being developed in our laboratory and its physical parameters (see Fig. 1) are used in this work's numerical results.



Symbol	Definition	Value
m	mass	$4.0\ kg$
g	local gravity	$9.81\ \frac{m}{s^2}$
I_{xx}	x inertia	$0.033\ kgm^2$
I_{yy}	y inertia	$0.033\ kgm^2$
I_{zz}	z inertia	$0.066\ kgm^2$
L	half wingspan	$0.5\ m$
b	thrust coefficient	$2.64e{-4}\ Ns^2$
d	drag coefficient	$7.5e{-7}\ Nms^2$

Figure 1: *Quadrotor and its coordinate systems and parameters.*

The helicopter's dynamics can be described by a nonlinear 12^{th} order model with states corresponding to Cartesian positions x, y and z (in meters); the attitude angles ϕ (pitch), θ (roll), and ψ (yaw) in radians; and their respective rates $(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$. The manipulated variables used for control are the four propellers' rotational speeds. The dynamical model's equations can be obtained by the Lagrange-Euler formalism, as presented in (Santana and Braga (2008)) and (Santana and Borges (2009)). The system can be described by the following ODE:

$$\dot{\xi} = f(\xi, u) \tag{1}$$

where $\xi = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^T$ is the state vector and $u = [\Omega_1\ \Omega_2\ \Omega_3\ \Omega_4]^T$ is the control input (in radians per second), where Ω_i is the rotation imposed on the i-th motor. Owing to the extension of the resulting equations, they are omitted here. For a complete development, please refer to (Santana and Braga (2008)).

The helicopter's movements stem from changes in the propellers' rotations. Vertical movement is made by simultaneously increasing (up) or decreasing (down) the rotors' speeds. Longitudinal motions are achieved by means of changes in the front and rear rotors, while lateral displacements are performed using the speed of the right and left propellers. Yaw movement is obtained from unbalances in the counter-torque between each pair of propellers, i.e., accelerating the two clockwise-turning rotors while decelerating the counter-clockwise ones yields a counter-clockwise movement, and vice-versa.

The quadrotor's dynamics was approximated by a linear model using a first-order Taylor expansion around an equilibrium point. In this work, it is used the following open-loop unstable equilibrium point $\overline{\xi} = [0\ 0\ 0\ 0\ 10\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and $\overline{u} = [192.8\ 192.8\ 192.8\ 192.8]^T$, which correspond to a hover flight condition at a 10m height. In this condition, the linear model is characterized by the matrices A_c and B_c.

$$\dot{\xi} = A_c \xi + B_c u \tag{2}$$

For digital implementation, however, this model had to be discretized. The sampling period is adopted as $T_s = 50$ms and a zero-order-hold was included at the controller's output. After linearization and discretization, is obtained a model of the form

$$\begin{aligned} \xi(k+1) &= A\xi(k) + Bu(k) \\ y(k) &= C\xi(k) \end{aligned} \tag{3}$$

By defining the state and output vetors as $\xi = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^T$ and $y = [x\ y\ z\ \phi\ \theta\ \psi]^T$ and considering a quadrotor with parameters specified in Fig. 1, the model matrices are the following:

$$A = \begin{bmatrix} 1.0 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0123 & 0.0002 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4905 & 0.0123 & 0 & 0 \\ 0 & 0 & 1.0 & 0.05 & 0 & 0 & -0.0123 & -0.0002 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & -0.4905 & -0.0123 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.0002 & 0 & 0.0002 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.0002 & 0 & -0.0002 \\ 0.0001 & 0.0001 & 0.0001 & 0.0001 \\ 0.0021 & 0.0021 & 0.0021 & 0.0021 \\ 0 & -0.0012 & 0 & 0.0012 \\ 0 & -0.0472 & 0 & 0.0472 \\ -0.0012 & 0 & 0.0012 & 0 \\ -0.0472 & 0 & 0.0472 & 0 \\ 0 & 0 & 0 & 0 \\ 0.0001 & -0.0001 & 0.0001 & -0.0001 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

3. Predictive control formulation

Figure 2 presents the min elements of the discrete-time predictive control formulation adopted in this work. In this figure, $u(k) \in \mathbb{R}^p$, $y(k) \in \mathbb{R}^q$, and $r(k) \in \mathbb{R}^p$ denote, respectively, the manipulated variables (plant's inputs), the controlled variables (plant's outputs), and the reference signals. The plant's model (Eq. 3) is employed to calculate output predictions ($\hat{y}(k+i|k)$) up to N steps in the future, where N is termed "Prediction Horizon". Such predictions are determined on the basis of the state ($x(k)$) measured at the present time (k-th sampling instant) and they are also dependent on the applied control sequence $\Delta\hat{u}(k+i-1|k)$. The increments in control are denoted by $\Delta\hat{u}(k+i-1|k) = \hat{u}(k+i-1|k) - \hat{u}(k+i-2|k)$ and the hat symbol ($\hat{}$) denotes that the optimal control sequence at time k can still change.

The optimization algorithm is aimed at determining the sequence of future control increments $\Delta\hat{u}(k-1+i|k), i = 1, ..., M$, that minimize the cost function specified for the problem subject to constraints on the plant inputs and outputs. The value of M ("Control Horizon") is typically smaller than N, and the optimization assumes that $\Delta\hat{u}(k-1+i|k) = 0$ for $M < i \leq N$. The control is implemented in a receding horizon manner, where only the first element of the optimized control sequence ($u^*(k)$) is applied to the plant and the optimization is repeated at the next sampling instant, on the basis of fresh measurements.

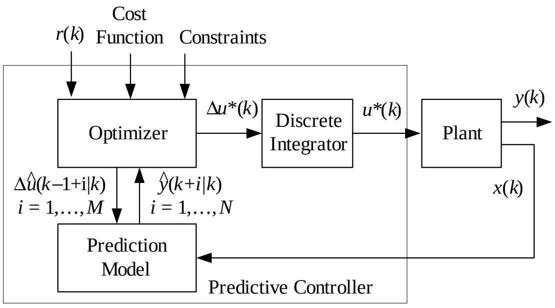


Figure 2: Predictive control loop employing state feedback. The optimal control at instant k is denoted by $u^*(k)$.

The following cost function, which penalizes tracking errors at q plant outputs and control variations at p plant inputs, was adopted:

$$J(\Delta\hat{U}) = \sum_{j=1}^{q}\sum_{i=1}^{N} \mu_j[\hat{y}_j(k+i|k) - r_j(k+i)]^2 + \sum_{l=1}^{p}\sum_{i=1}^{M} \rho_l[\Delta\hat{u}_l(k-1+i|k)]^2 \tag{5}$$

where $\rho_l > 0; l = 1, \dots, p$ and $\mu_j \geq 0; j = 1, \dots, q$. The design parameter ρ may be adjusted to achieve a compromise between minimizing the output tracking error and minimizing variations on the control signal. Decreasing ρ tends to increase the speed of the closed-loop response at the cost of a larger control effort and a greater sensitivity to measurement noise.

The use of control variations in the cost function formulation aims to provide the controller with an integral action, allowing offset-free tracking (Maciejowski (2002)).

By defining an output weighting matrix W_y as

$$W_y = \begin{bmatrix} \Sigma(\mu) & 0_{q\times q} & \cdots & 0_{q\times q} \\ 0_{q\times q} & \Sigma(\mu) & \cdots & 0_{q\times q} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{q\times q} & 0_{q\times q} & \cdots & \Sigma(\mu) \end{bmatrix}, \text{where } \Sigma(\mu) = \begin{bmatrix} \mu_1 & 0 & \cdots & 0 \\ 0 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_q \end{bmatrix}$$

and an input weighting matrix W_u in a similar manner (using weights ρ_l), the cost function can be rewritten in the form

$$J(\Delta\hat{U}) = (\hat{Y} - R)W_y(\hat{Y} - R)^T + \Delta\hat{U}W_u\Delta\hat{U}^T \tag{6}$$

where

$$R = \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+N) \end{bmatrix}, \; \hat{Y} = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix}, \; \Delta\hat{U} = \begin{bmatrix} \Delta\hat{u}(k|k) \\ \vdots \\ \Delta\hat{u}(k+M-1|k) \end{bmatrix} \tag{7}$$

The relation between $\hat{Y}$ and $\Delta\hat{U}$ can be expressed by a prediction equation based on an incremental state-space model. By assuming a linearized model of the form showed in Eq. 3, the incremental model can be written as

$$\begin{aligned} \Delta x(k+1) &= A\Delta x(k) + B\Delta u(k) \\ \Delta y(k) &= C\Delta x(k) \end{aligned} \tag{8}$$

Therefore, a prediction equation for Δy can be written as (Maciejowski (2002))

$$\Delta\hat{Y} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CA^{N-M}B \end{bmatrix} \Delta\hat{U} + \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} \Delta x(k) = P\Delta\hat{U} + Q\Delta x(k) \tag{9}$$

It can be easily seen that $\Delta\hat{Y}$ and $\hat{Y}$ can be related as

$$\hat{Y} = \begin{bmatrix} I_q & 0 & \cdots & 0 \\ I_q & I_q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_q & I_q & I_q & I_q \end{bmatrix} \Delta\hat{Y} + \begin{bmatrix} I_q \\ I_q \\ \vdots \\ I_q \end{bmatrix} y(k) = T_N^{I_q}\Delta\hat{Y} + \Gamma_N^{I_q} y(k) \tag{10}$$

By using the identity in (9), it follows that $\hat{Y} = T_N^{I_q}P\Delta\hat{U} + T_N^{I_q}Q\Delta x(k) + \Gamma_N^{I_q}y(k) = G\Delta\hat{U} + F$, where $G = T_N^{I_q}P$ and $F = T_N^{I_q}Q\Delta x(k) + \Gamma_N^{I_q}y(k)$. Therefore, the cost function (6) can be rewritten as

$$J(\Delta\hat{U}) = (G\Delta\hat{U} + F - R)^T W_y (G\Delta\hat{U} + F - R) + \Delta\hat{U}^T W_u \Delta\hat{U} = \frac{1}{2}\Delta\hat{U}^T \mathcal{G}\Delta\hat{U} + f^T\Delta\hat{U} + c \tag{11}$$

where $(1/2)\mathcal{G} = G^T W_y G + W_u$, $f^T = 2(F-R)^T W_y G$ and $c = (F-R)^T W_y (F-R)$.

It can, thus, be seen that the cost is a quadratic function of the optimization variables $\Delta\hat{U}$. In the absence of constraints, the control sequence ΔU^* that minimizes the cost is given by $\Delta U^* = (G^T W_y G + W_u)^{-1} G^T W_y (R - F)$.

If restrictions of the form $u_{min} \leq \hat{u}(k-1+i|k) \leq u_{max}$, $i \in \{1, \dots, M\}$, and $y_{min} \leq \hat{y}(k+i|k) \leq y_{max}$, $i \in \{1, \dots, N\}$, are to be satisfied on the manipulated and controlled variables, the minimization of the cost is subject to the following linear constraints on $\Delta\hat{U}$ (Camacho and Bordons (1999)):

$$\begin{bmatrix} T_M^{I_p} \\ -T_M^{I_p} \\ G \\ -G \end{bmatrix} \Delta\hat{U} \leq \begin{bmatrix} Bl_M[u_{max} - u(k-1)] \\ Bl_M[u(k-1) - u_{min}] \\ Bl_N[y_{max}] - F \\ F - Bl_N[y_{min}] \end{bmatrix} \tag{12}$$

where $T_M^{I_p}$ is a lower block-triangular matrix of identities and $Bl_N[\bullet]$ is an operator that stacks N copies of a column vector. The set of constraints (12) can be rewritten as $S\Delta\hat{U} \leq b$. Therefore, the optimization problem to be solved at each sampling period is one of Quadratic Programming (quadratic cost, linear constraints), for which efficient numerical algorithms are available (Maciejowski (2002)).

4. MPC applied to a quadrotor helicopter

As mentioned in section 1., quadrotor's dynamical model has six outputs $(x, y, z, \phi, \theta, \psi)$ and only four independent inputs. The approach used in this work was to choose four of the state vector's variables to be controlled. The MPC's ability in handling constraints on output variables was explored to keep other state variables stabilized. The idea consists of imposing "fictitious" restrictions (constraints that are not intrinsic to the system) on two variables that are not controlled, in order to restrict their variation to a small range, thus keeping the helicopter stable.

In order to track the desired trajectory, a possible combination of controlled outputs can be (x, y, z, ψ). For this situation, the output weights were defined as $\mu = [1\ 1\ 1\ 0\ 0\ 10]$. The weights for the pitch and roll angles were fixed at zero because no external setpoint is imposed on these variables. The larger weight for the yaw angle informs the controller that tracking this variable's reference is more important than controlling the other three Cartesian positions.

Thus, the two additional variables ϕ and θ are not controlled, but restrictions on their amplitudes are imposed in order to limit the excursion of these signals to a specified range. A possible set of constraints can be: $\phi_{max} = 5°$, $\phi_{min} = -5°$, $\theta_{max} = 5°$ and $\theta_{min} = -5°$. No formal criterion was used in setting the constraint values to limit the signals range, so they may be changed as needed.

The restrictions imposed on the attitude angles make the system slower. If necessary, higher limits can be imposed on the angles or adjustments can be made on the values of control parameters (ρ) in order to get a faster control loop. Indeed, decreasing ρ tends to increase the speed of the closed-loop response at the cost of a larger control effort and a greater sensitivity to measurement noise. In this study, the control weights were set to $\rho = [0.01\ 0.01\ 0.01\ 0.01]$ and no restrictions were imposed on other output variables.

5. Simulation Results

5.1 Methodology

All simulations were carried out using the Simulink environment in Matlab® R2010a. The dynamical system was described using nonlinear differential equations and approximated by a 4th order Runge-Kutta integrator with constant $T_s/10$. A specific Matlab S-function was written to implement the predictive control law and the Quadratic Programming problem was solved by using the *quadprog* function in Matlab's (*Optimization Toolbox*).

The simulations were divided into three parts. The initial part of the study consisted of determining appropriate values for the prediction and control horizons. In these cases, the simulations were performed using a 10m step reference in the direction of x by keeping the elevation at a fixed operating level ($z = 10m$).

The second part was directed to the study of the control loop's stability. In order to illustrate the importance of imposing restrictions on roll and pitch angles, two different scenarios were investigated. First, no constraints were placed on roll and pitch angles (which is equivalent to using relaxed constraints in the quadratic programming algorithm). Second, a lower and an upper bound for the roll and pitch angles were set to $[-5°, 5°]$, as mentioned in section in Section 4. The experiments consisted of the application of 10m steps references in the three displacement variables x, y, and z. Then, to show the controller's stability when constraints are imposed on the two mentioned angles, the helical path

$$x_r(t) = 2\cos(0.2t), \;\; y_r(t) = 2\sin(0.2t), \;\; z_r(t) = 0.2t, \;\; \psi_r = 0°,$$

was defined as the reference trajectory.

Finally, a comparison was made between this work's predictive control structure and two traditional techniques for quadrotor control, namely the PID control technique proposed in (Castillo *et al.* (2004a)) and the backstepping controller proposed in (Bouabdallah and Siegwart (2005)). Since (Castillo *et al.* (2004a)) performs only systems stabilization and attitude control, the simulation used a step reference of 10m in z displacement variable for comparison with the MPC. References imposed on other MPC controlled variables were: $x_r(t) = 0$, $y_r(t) = 0$, and $\psi_r = 0°$. The following initial conditions were considered: $x_0 = 0$, $y_0 = 0$, $\theta_0 = -4°$, $\phi_0 = -4°$, $\psi_0 = 10°$, and all null speeds. For comparison with the backstepping controller, the maneuver was carried out by using a 10m step reference in all three displacement variables, considering the same quadrotor's initial conditions as the previous case.

In all simulations with constraints handling, the restrictions were imposed on the propellers' rotations, which were limited to the range of 0 to 4500 RPM.

5.2 Adjustment of prediction and control horizons

In order to investigate the effect of changes on the prediction horizon, M was set to 5 and five values of N were tested ($N = 20, 30, 50, 70, 90$). Setting $N = 20$ (or smaller) rendered the control loop unstable. It can be argued that, with a small prediction horizon, the controller cannot "see" the future effect of its actions and, as a result, the control decisions are not adequate. Figure 3.a presents the x response for $N = 30, 50, 70, 90$. As can be seen, a small prediction horizon ($N = 30$) tends to produce a closed-loop response with low damping, which is in agreement with the previous discussion. Since the differences between $N = 70$ and $N = 90$ are minor, computational requirements would favour the selection of $N = 70$.

After fixing the prediction horizon at $N = 70$, three additional tests were carried out by varying the control horizon M ($M = 3, 5, 10$). Figure 3.b shows the results. As can be seen, the best result is obtained with $M = 5$.

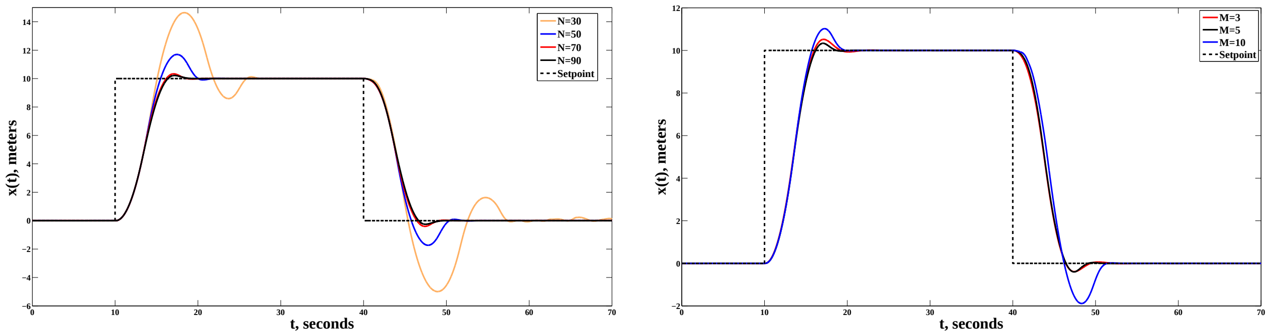


Figure 3: Adjustment of prediction and control horizons. (a) Effect of changes in the prediction horizon N for a control horizon set to $M = 5$. (b) Effect of changes in the control horizon M for a prediction horizon set to $N = 70$.

5.3 Stability analysis

The simulation results for the controller with and without output constraints handling are shown in Figure 4. The dashed lines on the graphs of x, y, and z are the reference signals. Horizontal full lines on the graphs of the pitch and roll angles represent the constraints imposed on these variables. As can be seen, the control loop is stable when MPC is operating under constraints (red curves), but becomes unstable in the absence of them. It can be argued that, without output constraints handling, the controller can apply control signals that yield roll and pitch angles with large amplitudes. In these situations, the helicopter begins to operate in adverse conditions, so the controller becomes unable to find a feasible quadratic programming problem that brings the quadrotor back to a safe operating zone. This fact, combined with the system's fast dynamics, causes instability of the control loop. It should be noticed that the control loop remains stable for the initial 40s of simulation. This happened because the reference steps were applied to displacement variables at different time instants. Although the amplitudes of the roll and pitch angles were high (close to 60° and $-60°$), they did not happen simultaneously. At time $t = 40$s, however, the reference signals were changed in the three variables all at the same time, requiring the controller to choose high values for all angles and forcing the helicopter to operate in adverse regions, causing instability. Thus, when constraints are imposed, they ensure that the helicopter does not enter these regions, ensuring the control loop's stability.

Imposing restrictions on the system variables has the additional effect of extending the linear model's validity, which can be seen as another explanation for the stabilization phenomenon mentioned before. In order to see that, recall that the linear model used for the controller's prediction steps is obtained by a first-order Taylor series expansion around the equilibrium point $\overline{\xi} = [0\ 0\ 0\ 0\ 10\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$. Therefore, if one wishes to keep the discrepancy between the linear model and the system's dynamics to a minimum, it is important to ensure that the quadrotor's orientation will stay close to that of $\overline{\xi}$, which corresponds to ϕ, θ, and ψ close to 0 (zero). The price paid for stability is a slower response for the quadrotor, as shown in Figure 4. If necessary, soft constraints can be imposed on the roll and pitch angles in order to improve the system's response time.

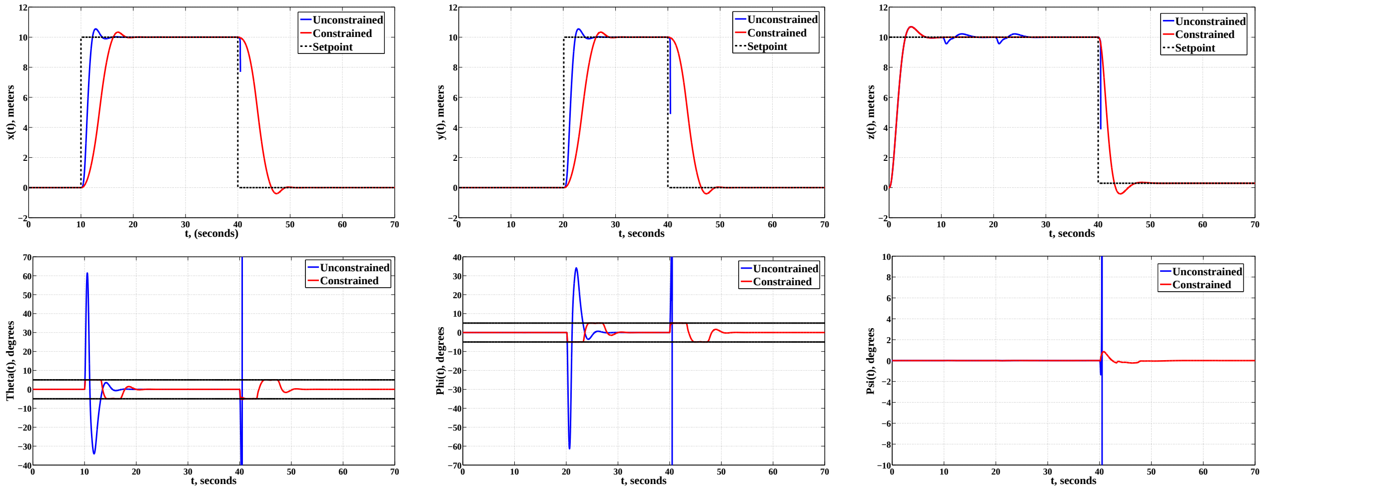


Figure 4: Stability analysis of MPC's control loop for a quadrotor controller.

In order to illustrate the control loop's stability, the quadrotor's behavior following a helical trajectory in the 3D plane (xyz) is shown in Figure 5. It can be seen that the controller is able to make the helicopter follow the desired trajectory (offset-free tracking) and, at the same time, not violate the restrictions imposed on the angles. It can also be noticed from Figures 4 and 5 that the controller is able, in both cases, to take the yaw angle (ψ) to zero, as previously specified.

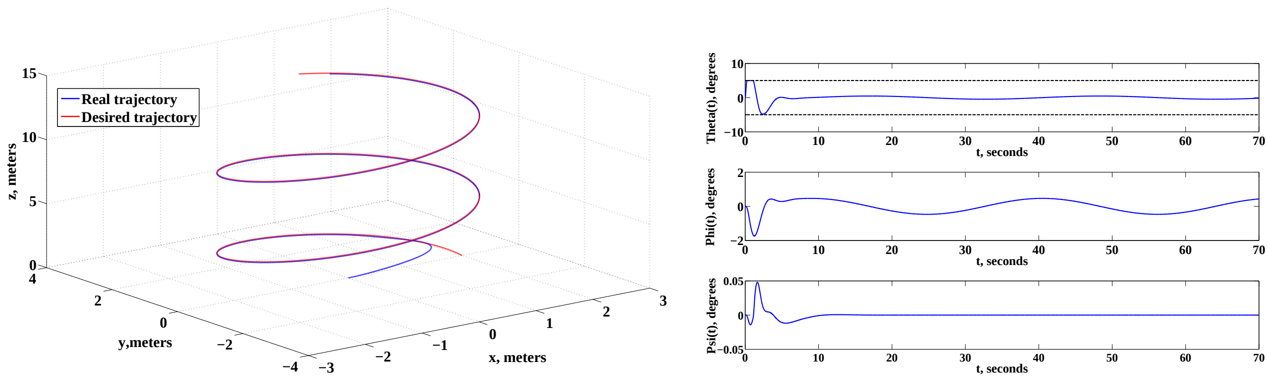


Figure 5: The quadrotor tracks well the desired helical path.

5.4 Comparison with PID controllers

Figure 6 shows the performance comparison between the MPC and PID controllers. First, note that PID control could not keep the helicopter stationary on the xy plane. This was an expected behavior, since this controller performs only stabilization and altitude control. It was also observed that both controllers were able to stabilize the system. However, the MPC is slower and its response is less damped, which is the price paid to keep the helicopter stationary on the xy plane. Additionally, the higher oscillations can be assigned to the constraints placed on roll and pitch angles. Indeed, the controller applies a control signal on the plant and observes the increase in the angles' values. To avoid restrictions violations, it applies a corrective signal to reduce these values. Since the system dynamics is fast, the resulting effect is the low damping observed. Thus, a smoother response can be obtained by increasing the maximum limits for the angles.

In Figure 6, note that the MPC's performance is superior to that of PID in elevation control (z), since it exhibits a lower response time and almost the same value of overshoot. Another advantage of the MPC is null steady-state error for all variables, while the PID presents a small error in the four controlled variables (θ,ϕ,ψ,z). In general, one can state that MPC's performance was better when compared to PID. Even though the observed behaviors in stabilization and altitude control were not much different, MPC could handle two

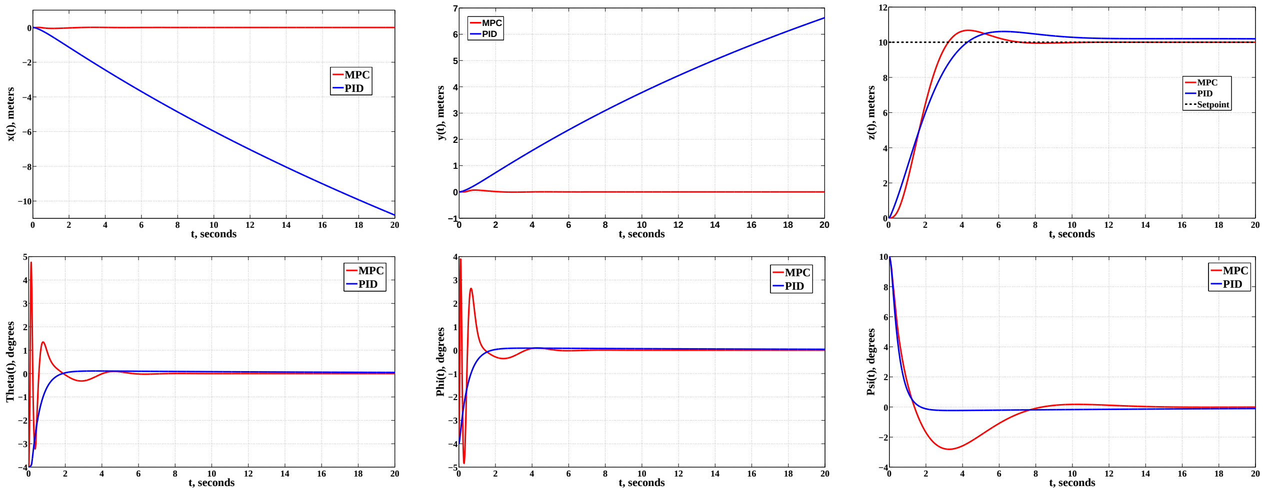


Figure 6: Results obtained with MPC and PID controllers.

additional variables (x and y), keeping the quadrotor's position totally controlled.

5.5 Comparison with a Backstepping controller

Figure 7 shows the performance comparison between MPC and a backstepping controller (BC). In the graphs presented, it can be seen that both controllers could stabilize the system and take the quadrotor to the desired position. It is also clear that the BC's performance is better than MPC's, since the three displacement variables reach stability faster and with almost the same overshoot as the one observed for the MPC. In can be argued, however, that the BC's superior performance is achieved by means of greater stresses over the actuators, as shown in Fig. 8. Because of the the fast varying rotation regime required by the BC, a propulsion system (motors and propellers) with very fast dynamics becomes necessary. In addition, Fig. 7 shows that the BC takes the quadrotor to very steep orientations in order to quickly move the aircraft to the desired position, a potentially dangerous situation if operating in the presence of external disturbances (wind, for example).

The MPC, on the other hand, naturally incorporates operational and physical constraints in its control law, which helps to ensure a safer operation. Additionally, in indoor applications and other situations where a known flight environment is used, the MPC may include restrictions on the output variables in order to avoid collisions. An indication of this can be seen in Fig. 9, where an asymmetric set of actuators is used to stabilize the quadrotor. In this situation, three motors can reach a maximum rotation of 4500 RPM, while motor 3 is limited to 2700 RPM. Since the MPC takes this restriction explicitly into account when searching for the optimal control input, it is able to circumvent this limitation and yield the control performance shown in Fig. 9. Nevertheless, similar results are not found for the BC, since its more agressive control approach makes it unable to recover from the actuators' saturation.

6. Conclusions

In this paper, a model predictive controller was presented to solve the stabilization and path tracking problems for a quadrotor helicopter. Concerning related work in quadrotor control using model predictive techniques, the proposed scheme's main contribution lies in a control loop having only one controller. The approach used is to control four output variables (x, y, z, and ψ), assuring position control, while imposing restrictions on the other two output variables (θ and ϕ) and exploring the MPC's ability to handle constraints in order to ensure the system's stability. The results obtained suggest that the approach presented here may be an alternative to the control of unstable and underactuated systems. The MPC's performance was compared with two other techniques widely used in quadrotor control, namely PID and Backstepping control. Our approach showed a better performance when compared to the PID controller proposed in (Castillo *et al.* (2004a)), but could not achieve the same speed of convergence as the one shown by the BC proposed in (Bouabdallah and Siegwart (2005)). Despite its slower response, the MPC does not require actuators as fast as the ones necessary for the BC. Moreover, MPC's less agressive control scheme, along with its natural ability to incorporate physical and operational constraints, contributes to safer operating conditions.

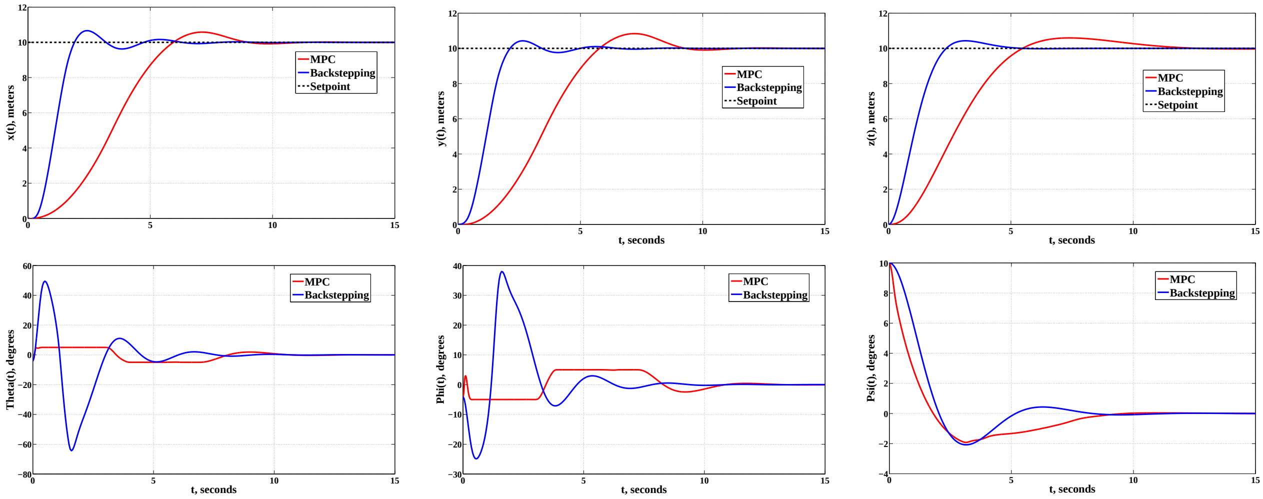


Figure 7: Results obtained with MPC and Backstepping controllers.

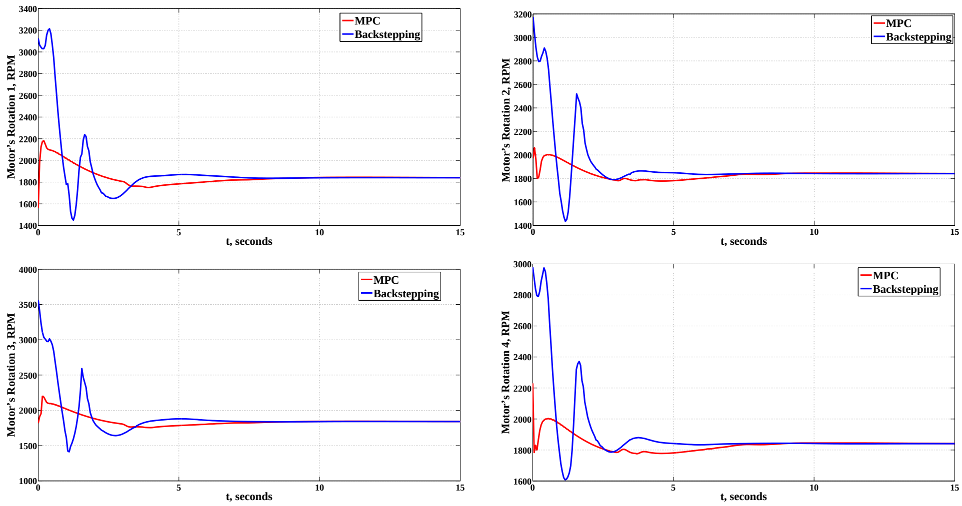


Figure 8: Motors'Rotations.

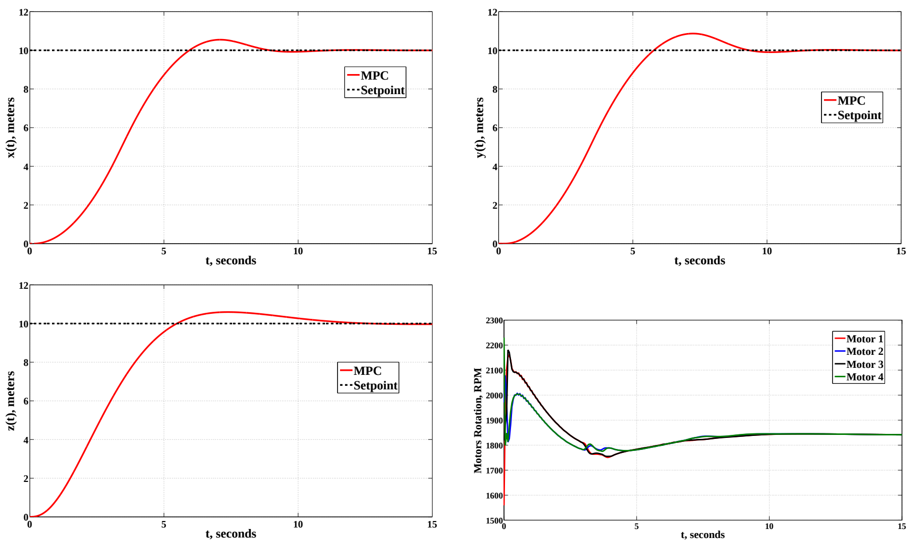


Figure 9: Quadrotor with asymmetric set of actuators.

7. ACKNOWLEDGEMENTS

The authors are funded by research grants 132787/2009-1, 304999/2008-3, and 310852/2006-4 from the National Council for Scientific and Technological Development (CNPq). The authors would also like to acknowledge the financial support of CAPES.

8. REFERENCES

Adigbli, P., Grand, C., Mouret, J.B. and Doncieux, S., 2007. "Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques". In *3rd US-European Competition and Workshop on Micro Air Vehicle Systems (MAV07) & European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*. Toulouse, France.

Alexis, K., Nikolakopoulos, G. and Tzes, A., 2010. "Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts". In *18th Mediterranean Conference on Control & Automation Congress Palace Hotel, Marrakech, Morocco*. pp. 1461–1466.

Becker, M., Bouabdallah, S. and Siegwart, R., 2006. "Desenvolvimento de um controlador de desvio de obstáculos para um mini-helicóptero quadri-rotor autônomo - 1^afase: Simulação". CBA - Congresso Brasileiro de Automática (CBA 2006), Vol. 1, pp. 1201–1206.

Bouabdallah, S., 2007. *Design and control of quadrotors with application to autonomous flying*. Ph.D. thesis, Échole Polytechnique Fédérale de Lausanne.

Bouabdallah, S., Noth, A. and Siegwart, R., 2004. "PID vs LQ control techniques applied to an indoor micro quadrotor". Proceedings of the 2004 IEEE International Conference on Intelligent Robots and Systems, Vol. 3, pp. 2451–2456.

Bouabdallah, S. and Siegwart, R., 2005. "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor". Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2259–2264.

Camacho, E.F. and Bordons, C., 1999. *Model Predictive Control*. Springer-Verlag, London.

Castillo, P., Dzul, A. and Lozano, R., 2004a. "Real-time stabilization and tracking of a four-rotor mini rotorcraft". *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 4, pp. 510–516.

Castillo, P., Lozano, R. and Dzul, A., 2004b. "Stabilization of a mini-rotorcraft having four rotors". Proceedings of the 2004 IEEE International Conference on Industrial Technology, Vol. 3, pp. 1543–1548.

Maciejowski, J.M., 2002. *Predictive Control with Constraints*. Prentice Hall, Harlow, England.

Madani, T. and Benallegue, A., 2006a. "Backstepping control for a quadrotor helicopter". In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. pp. 3255–3260.

Madani, T. and Benallegue, A., 2006b. "Control of a quadrotor mini-helicopter via full state backstepping technique". In *Proceedings of the 45th IEEE Conference on Decision & Control*. pp. 1515–1520.

Mistler, V., Benallegue, A. and M'Sirdi, N.K., 2001. "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback". Proceedings of the 2001 IEEE International Workshop on Robot and Human Interactive Communication, pp. 586–593.

Raffo, G.V., Ortega, M.G. and Rubio, F.R., 2008. "MPC with nonlinear H∞ control for path tracking of a quad-rotor helicopter". 17th World Congress International Federation of Automatic Control (IFAC-08).

Santana, P.H.R.Q.A. and Borges, G.A., 2009. "Modelagem e controle de quadrirrotores". In *IX Simpósio Brasileiro de Automação Inteligente (SBAI 2009)*. Vol. 9, pp. 1–6. Available at `http://www.lara.unb.br/~phsantana/data/files/articles/2009_Santana-Borges_SBAI_Modelagem_e_Controle_de_Quadrirrotores.pdf`.

Santana, P.H.R.Q.A. and Braga, M.A., 2008. "Concepção de um veículo aéreo não-tripulado do tipo quadrirrotor." Technical report, Universidade de Brasília. Available at `http://www.lara.unb.br/~phsantana/data/files/other/Santana_Braga_Conception_Quadrotor.pdf`.

Tayebi, A. and McGilvray, S., 2006. "Attitude stabilization of a VTOL quadrotor aircraft". In *IEEE Transactions on Control Systems Technology*. Vol. 14, pp. 562–571.

Vada, J., Slupphaug, O., Johansen, T.A. and Foss, B.A., 2001. "Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability". *Automatica*, Vol. 37, pp. 1835–1843.

Xu, R. and Ozguner, U., 2006. "Sliding mode control of a quadrotor helicopter". In *Decision and Control, 2006 45th IEEE Conference on*. pp. 4957–4962.

9. Responsibility notice