REVISTA INTERDISCIPLINAR DE PESQUISA EM ENGENHARIA

CILAMCE 2016

XXXVII IBERIAN LATIN AMERICAN CONGRESS
ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

Model Predictive Control Applied to a Quadrotor UAV

Tiago Santana Lourenço

tiagosl@gmail.com

Universidade de Brasília

Campus Universitário Darcy Ribeiro, Brasília, 70910-900, Distrito Federal, Brasil

André Murilo de Almeida Pinto

Renato Vilela Lopes

andremurilo@unb.br

rvlopes@unb.br

Universidade de Brasília

Faculdade Gama, Área Especial de Indústria Projeção A Brasília, 72.444-240, Distrito Federal, Brasil

Abstract. *This paper describes and analyzes the performance of two different formulations of model predictive control (MPC) applied to a quadrotor unmanned aerial vehicle (UAV), one that explicitly handles constraints and another that doesn't. The objective of the MPC strategy is to compute an optimal sequence of actions within its prediction horizon to track desired states. The optimization strategies adopted in this work are based on an approximated dynamical model for prediction while imposing a quadratic cost function. One prominent vantage of MPC is its ability to handle constraints inherent to the process, based either on actuators' limitations or security concerns. Through simulations, the impacts of imposing a convex set of constraints is analyzed, regarding the performance and computational effort involved in solving a trajectory tracking problem.*

Keywords: *Optimal control, Predictive Control, Quadrotors, UAVs*

1 INTRODUCTION

Research and development of Unmanned Aerial Vehicles (UAVs), accelerated by recent technological advances, has enabled applications such as search-and-rescue (Faessler et al. 2015), mapping (Scaramuzza et al. 2014, Holz & Behnke 2016) and visual inspection (Burri et al. 2012). Quadrotors are among the most frequently used platforms because of their ability to hover, high maneuverability and simple design. These UAVs can be used both indoors and outdoors, but controlling its flight is rather challenging because of its coupled dynamics and underactuated nature. In addition, the dynamics of the quadrotor are highly non-linear, introducing several uncertainties in its operation.

In the last decade, a considerable number of papers have been published regarding quadrotor dynamic modeling (Santana & Borges 2009, Kim et al. 2010, Powers et al. 2015). Several control strategies have been proposed and tested for these systems, which include: PID, LQR, backstepping and feedback linearization. A broad compilation of control algorithms for quadrotors can be found in (Zulu et al. 2014).

More recently, advanced controllers have been made possible by the boost of computational power in mobile devices. One prominent strategy that has emerged in this context is model predictive control (MPC). Dating back to the late 1970s, MPC was first conceived to control industrial chemical processes (Richalet et al. 1978), which typically have slow dynamics and sampling times measured in seconds or minutes. With the high performance processors available today, MPC has been applied for fast systems, such as the quadrotor.

The strategy of MPC consists in computing an optimal control sequence over a prediction horizon, by minimizing some given cost function expressing the control objective. It relies on the dynamical model of the system to predict its response to the commands. An important feature of MPC is its constraint handling, that keeps the systems states and commands inside a desirable zone of operation. The optimization problem involved depends greatly on the formulations of the cost function and constraints. If the cost function is quadratic and the constraints imposed are a convex set, the resulting optimization will be a quadratic programming problem (QP), which is indeed a common approach in MPC formulations (Wang & Boyd 2010).

MPC has been successfully used for control of UAVs. Lopes et al. (2011) formulated a MPC based on a linearized model of the quadrotor. The proposed controller can follow trajectories while imposing constraints to the orientation of the aircraft (roll and pitch angles). In (Izadi et al. 2011) a fault tolerant MPC controller is proposed, based on the Unscented Kalman Filter for estimating nonlinear parameters. Important remarks are also made about the high computational time necessary for the algorithm. Experimental validation of predictive controllers are presented in (Burri et al. 2012), testing both LQR and MPC controllers. In their experimental setup, proposed for industrial visual inspection, a quadrotor equipped with a low level controller and on-board cameras navigates confined spaces.

In this paper, two different approaches of predictive control are compared through simulations. First, an unconstrained MPC with finite prediction horizon that amounts to a linear feedback control, similar to a LQR. Second, a constrained MPC formulation that yields a QP problem to optimize the future control inputs. In the latter, the constraint handling in state and input variables are demonstrated, but also result in a considerable increase of computational effort. Further, the controllers' performance is discussed, regarding their trajectory tracking and stabilization.

2 DESCRIPTION AND MODELING

2.1 Notation

Here, we define the notation used in the remainder of the paper. These are useful for both the modeling and the MPC formulation.

Vectors and matrices: All vectors are column vectors, and the transpose of a vector or matrix is denoted with a superscript T, e.g. A^T. The notation $diag(d_1, ..., d_n)$ denotes a diagonal matrix with d_i values along the diagonal. The notation $\mathbb{I}_n$ denotes a $n \times n$ identity matrix. Similarly, a $n \times n$ zero matrix is represented by $\mathbb{O}_n$.

Quadratic norm: The notation $\|\nu\|_Q^2$ denotes the quadratic norm $\nu^T Q \nu$.

Discrete-time state space: Variables that change at each discrete timestep have the time index denoted between parenthesis, e.g. $x(k)$. In the state space model representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y_\alpha(k) &= C_\alpha x(k) \end{aligned}$$

The state vector $x(k)$ has its dimension denoted by n, i.e. $x \in \mathbb{R}^n$. Similarly, the input vector has its dimension denoted by n_u. The state and input matrices are respectively $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n_u}$. Output vectors like $y_\alpha \in \mathbb{R}^{n_\alpha}$ are defined by output matrices like $C_\alpha \in \mathbb{R}^{n_\alpha \times n}$.

Variable sequences: Given a finite prediction horizon of N future steps, a consecutive sequence of a variable $y(k)$ within the prediction horizon ($i \in [1, ..., N]$) is denoted by a $\tilde{y}$. For example, the future control inputs can be written as the vector:

$$\tilde{u} = \begin{bmatrix} u(k) \; u(k+1) \; \dots \; u(k+N-1) \end{bmatrix}^T \in \mathbb{R}^{N \cdot n_u}$$

To isolate any component of this sequence, the following operator is used:

$$u(k+i) = \Pi_i^{(n_u,N)} \tilde{u}$$

where $\Pi_i^{(n,N)}$ extracts the i-th vector of the concatenation of N vectors of dimension n.

2.2 Nonlinear Dynamics

The quadrotor's flight and motion is achieved by the propulsion and torques generated by its rotors. The only actuators of the system are its four motors, one pair spins clockwise while the other spins counterclockwise. This way, when all motors spin with the same velocity, their torques cancel out. Additionally, if all motors reach a certain speed ($\bar{\omega}$), it's possible to generate enough thrust so that the vehicle can hover. In figure 1 the motor velocities (ω_i), thrust forces (f_i) and torques (τ_{M_i}) are represented, along with the inertial ($\mathbb{I}$) and body ($\mathbb{B}$) reference frames.

Assuming the quadrotor to be a rigid body with six degrees of freedom $(x, y, z, \phi, \theta, \psi)$ and identical actuators, the nonlinear dynamical model can be derived by the Euler-Lagrange equations, as shown in (Santana & Borges 2009). Assuming small inclinations (ϕ, θ), the model

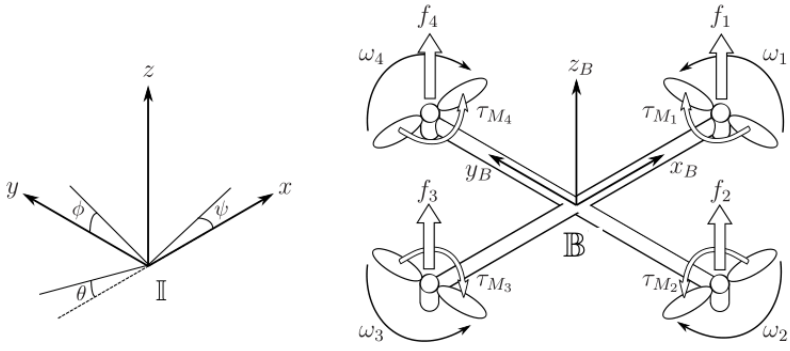

Figure 1: Diagram of quadrotor dynamics with inercial and body reference frames, adapted from (Luukkonen 2011).

can be written:

$$\ddot{x} = \left(C_\psi S_\theta C_\phi + S_\psi S_\phi\right)\frac{U}{m} \qquad \ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\psi}\dot{\theta} + \frac{\tau_x}{I_{xx}}$$

$$\ddot{y} = (S_\psi S_\theta C_\phi - S_\phi C_\psi)\frac{U}{m} \qquad \ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\psi}\dot{\theta} + \frac{\tau_y}{I_{yy}}$$

$$\ddot{z} = -g + (C_\theta C_\phi)\frac{U}{m} \qquad \ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} + \frac{\tau_z}{I_{zz}}$$

Here $C_\alpha = cos(\alpha)$, m is the UAV's mass, I_{jj} denotes de mass moment of inertia along the j-axis and $U = f_1 + f_2 + f_3 + f_4$ is the total thrust. The resulting torques in the body frame are represented by (τ_x, τ_y, τ_z) and are achieved by unbalancing the opposite thrusts and torques. These are modeled as:

$$\tau_x = bL(\omega_4^2 - \omega_2^2),\ \tau_y = bL(\omega_3^2 - \omega_1^2),\ \tau_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

where b and d are the thrust and drag coefficients respectively. The motor's distance from the center of mass (also center of $\mathbb{B}$) is L, assumed to be the same for all motors.

2.3 Approximate Dynamics

Assuming near hover operation, the quadrotor's dynamic model can be aproximated by a linear one. Using a first order Taylor expansion around an equilibrium point, the model can be rewritten in state space representation. Further, after defining a sampling period of τ_s, the discretized model is:

$$\xi(k+1) = A\xi(k) + Bu(k) \tag{1}$$

Here $\xi = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^T$ is the state vector and $u = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$ are the control inputs. The state and input matrices are A and B, respectively. These matrices can be found in (Lopes et al. 2011), that define the equilibrium point to be $\bar{\xi} = [0\ 0\ 0\ 0\ 10\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and $\bar{u} = [\bar{\omega}\ \bar{\omega}\ \bar{\omega}\ \bar{\omega}]^T$. This point represents the condition of hovering $10m$ high, which is achieved when all rotors spin in the right velocity ($\bar{\omega}\ rad/s$) with thrusts aligned upwards.

3 PREDICTIVE CONTROL FORMULATION

As previously mentioned, the MPC strategy relies on the dynamical model to predict the system's response to inputs within a finite prediction horizon of N future steps. Further, a cost function is defined to evaluate the trajectory tracking of future control inputs $\tilde{u}$. This amounts to an optimization problem that computes the optimal course of actions $\tilde{u}^{opt}$, that minimizes the cost function. However, only the first action of this sequence is actually applied to the system, that is $\Pi_1^{(n_u,N)}\tilde{u}^{opt}$. In the next sampling instant, a new reading of the current state $\xi(k)$ is made and the process repeats.

The complexity of this optimization depends essentially on the formulation of the prediction model, cost function and constraints. With a quadratic cost function and no constraints, the minimization is straightforward and solving for $\tilde{u}^{opt}$ is analytical. Further, introducing a convex set of constraints significantly complicates the solution. This way, $\tilde{u}^{opt}$ is obtained through a quadratic programming problem (QP). Fortunately, QPs have a wide range of solution methods (Wang & Boyd 2010). The definitions necessary for these two approaches, with or without constraints, are presented next.

3.1 Prediction Map

Assuming our quadrotor operates near hover condition during most of its flight, the model in Eq. (1) can be used to predict future states $\xi(k+i)\ \forall\ i \in [1,...,N]$. So assuming that the current state $\xi(k)$ and the sequence of applied commands $\tilde{u}$ are known:

$$\begin{aligned}
\xi(k+1) &= A\xi(k) + Bu(k) \\
\xi(k+2) &= A\Big[A\xi(k) + Bu(k)\Big] + Bu(k+1) = A^2\xi(k) + ABu(k) + Bu(k+1) \\
&\vdots \\
\xi(k+i) &= A^i\xi(k) + \begin{bmatrix} A^{i-1}B & A^{i-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+i-1) \end{bmatrix}
\end{aligned}$$

Finally, the prediction model can be rewritten as

$$\xi(k+i) = \Phi_i\xi(k) + \Psi_i\tilde{u} \tag{2}$$

where the model matrices are

$$\Phi_i = [A^i]\,, \quad \Psi_i = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} \Pi_1^{(n_u,N)} \\ \Pi_2^{(n_u,N)} \\ \vdots \\ \Pi_i^{(n_u,N)} \end{bmatrix} \tag{3}$$

3.2 Cost Function

The control objective is to drive the system's outputs to a desired trajectory. These regulated outputs may be state variables or their linear combinations, thus:

$$y_r(k) = C_r \xi(k) \tag{4}$$

Here $y_r \in \mathbb{R}^{n_r}$ is the regulated output vector defined by the matrix $C_r \in \mathbb{R}^{n_r \times n_r}$. Moreover, the desired trajectory is represented by $y_r^d(k+i) \in \mathbb{R}^{n_r}$.

The cost function attributes a scalar to a given sequence of inputs $\tilde{u}$, expressing how well it tracks the desired trajectory. It can be defined as (Alamir 2013):

$$J(\tilde{u}) = \sum_{i=1}^{N} \|y_r(k+i) - y_r^d(k+i)\|_{Q_y}^2 + \sum_{i=1}^{N} \|u(k+i-1) - \bar{u}\|_{Q_u}^2 \tag{5}$$

where the first sum imposes penalty on the regulated output error, while the second sum penalizes control input deviation from the equilibrium point $\bar{u}$. Further, the weighting matrices $Q_y = Q_y^T > 0$ and $Q_u = Q_u^T > 0$ are chosen according to the regulation objectives.

After substituting the Eqs. (2) and (4) in (5) and expanding the norms, the cost function can be rewritten as:

$$J(\tilde{u}) = \frac{1}{2}\tilde{u}^T H \tilde{u} + [F_1 \xi(k) + F_2 \tilde{y}_r^d + F_3 \bar{u}]^T \tilde{u} \tag{6}$$

where the following definitions were introduced:

$$H \triangleq 2\sum_{i=1}^{N} \left[\Psi_i^T C_r^T Q_y C_r \Psi_i + (\Pi_i^{(n_u,N)})^T Q_u (\Pi_i^{(n_u,N)}) \right] \tag{7}$$

$$F_1 \triangleq 2\sum_{i=1}^{N} \left[\Psi_i^T C_r^T Q_y C_r \Phi_i \right] \tag{8}$$

$$F_2 \triangleq -2\sum_{i=1}^{N} \left[\Psi_i^T C_r^T Q_y \Pi_i^{(n_r,N)} \right] \tag{9}$$

$$F_3 \triangleq 2\sum_{i=1}^{N} \left[(\Pi_i^{(n_u,N)})^T Q_u \right] \tag{10}$$

$$\tag{11}$$

3.3 Constraints

Constraint handling is a major advantage of MPC strategies. They can cope with the system's inherent limitations like actuator saturation, operational restrictions or security concerns. Here, the constraints are imposed as convex sets of the system's inputs and outputs.

Once again, the constrained output vetor is composed of state variables or their linear combinations:

$$y_c(k) = C_c \xi(k) \tag{12}$$

where the matrix C_c defines which variables will be bounded. So for all future instants in the prediction horizon, i.e. $i \in [1, ..., N]$:

$$y_c^{min} \leq y_c(k+i) \leq y_c^{max} \tag{13}$$

After substituting Eqs. (2) and (12) in (13) and rearranging, the output bounds can be rewritten as the following inequalities:

$$\underbrace{\begin{bmatrix} C_c\Psi_1 \\ \vdots \\ C_c\Psi_N \\ -C_c\Psi_1 \\ \vdots \\ -C_c\Psi_N \end{bmatrix}}_{A_{ineq}^{(1)}} \tilde{u} \leq \underbrace{\begin{bmatrix} -C_c\Phi_1 \\ \vdots \\ -C_c\Phi_N \\ C_c\Phi_1 \\ \vdots \\ C_c\Phi_N \end{bmatrix}}_{G_1^{(1)}} \xi(k) + \underbrace{\begin{bmatrix} y_c^{max} \\ \vdots \\ y_c^{max} \\ -y_c^{min} \\ \vdots \\ -y_c^{min} \end{bmatrix}}_{G_3^{(1)}} \tag{14}$$

Next, the input constraints are defined to handle actuator limitations. Firstly, by imposing restrictions the rate of change:

$$\delta_{min} \leq u(k+i) - u(k+i-1) \leq \delta_{max} \tag{15}$$

This can be rewritten using notation defined in Section 2.1 and rearranged as following:

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

The inequities in Eqs. (14) and (16) can be combined and rewritten in more compact form:

$$A_{ineq}\tilde{u} \leq G_1\xi(k) + G_2 u(k-1) + G_3 \tag{17}$$

where the following definitions were introduced, based on the highlighted terms.

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

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

Finally, one more set of bounds is imposed to take actuator saturation into account.

$$u_{min} \leq u(k+i) \leq u_{max} \tag{20}$$

4 Unconstrained MPC

The unconstrained approach is based on the prediction horizon of Eq. (2) and cost funtion of Eq. (5). Without restrictions, minimization consists in taking the gradient of the cost function equal to zero. So as long the matrix H is positive definite ($H > 0$), a global minimum is found by (Alamir 2013):

$$H\tilde{u} + F_1\xi(k) + F_2\tilde{y}_r^d + F_3\bar{u} = 0 \tag{21}$$

Therefore, the optimal sequence can be isolated and given by the following linear feedback law:

$$\tilde{u}^{opt} = -H^{-1}\big[F_1\xi(k) + F_2\tilde{y}_r^d + F_3\bar{u}\big] \tag{22}$$

Considering that the MPC only applies the first action of this sequence, the feedback law can be rewriten:

$$u^{opt}(k) = K_N\xi(k) + G_N\tilde{y}_r^d + L_N\bar{u} \tag{23}$$

where the constant gains K_N, G_N and L_N are given by:

$$K_N = -\Pi_1^{(n_u,N)} \cdot H^{-1}F_1 \tag{24}$$

$$G_N = -\Pi_1^{(n_u,N)} \cdot H^{-1}F_2 \tag{25}$$

$$L_N = -\Pi_1^{(n_u,N)} \cdot H^{-1}F_3 \tag{26}$$

This approach is similar to a LQR controller, as discussed in detail by Alamir (2013). In fact, if we take $C_r = \mathbb{I}_{n\times n}$ and N sufficiently high, the gain K_N converges to that of the LQR controller. In this case, the unconstrained MPC can be seen as an LQR controller combined with the feedforward terms G_N and L_N, to track the trajectory $\tilde{y}_r^d$ and stabilize the commands in the equilibrium point $\bar{u}$.

5 Constrained MPC

The constrained MPC, based on the same prediction model of Eq. (2) and cost function of Eq. (5), goes further by introducing the set of constraints expressed in Eqs. (17) and (20). The resulting optimization problem is convex as long as the matrix H is definite positive. Thus, the QP and can be formally written as:

$$\begin{aligned}\tilde{u}^{opt}(k) \triangleq argmin : &\left[\frac{1}{2}\tilde{u}^T H\tilde{u} + F^T(k)\tilde{u}\right] \\ subject\ to : &A_{ineq}\tilde{u} \leq B_{ineq}(k) \\ &\tilde{u}^{min} \leq \tilde{u} \leq \tilde{u}^{max}\end{aligned} \tag{27}$$

where the following definitions are introduced:

$$F(k) = F_1\xi(k) + F_2\tilde{y}_r^d + F_3\bar{u} \tag{28}$$

$$B_{ineq}(k) = G_1\xi(k) + G_2u(k-1) + G_3 \tag{29}$$

As previously mentioned, QPs have a considerable number of solution methods and software libraries such as the ones presented by Mattingley & Boyd (2012) and Ferreau et al. (2014).

6 SIMULATION RESULTS

Numerical simulations were carried out in order to observe the performance of both MPC strategies for the trajectory tracking problem. The values of the model parameters used in the simulations were extracted from Lopes et al. (2011) which can be seen in Table 1.

Table 1: Quadrotor model parameters.

Symbol	Definition	Value
m	mass	4.0 Kg
g	local gravity	9.81 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
$\bar{\omega}$	Rotor velocity for hovering	192.8 rad/s

The reference trajectory is defined as consecutive $10\ m$ long steps in the x, y and z axis, keeping ψ a constant.

$$x^d(t) = \begin{cases} 10 & \text{if } 10 < t < 40 \\ 0 & \text{if } t < 10 \text{ or } t > 40 \end{cases} \quad ; \quad y^d(t) = \begin{cases} 10 & \text{if } 20 < t < 40 \\ 0 & \text{if } t < 20 \text{ or } t > 40 \end{cases}$$

$$z^d(t) = \begin{cases} 10 & \text{if } 0 < t < 40 \\ 0 & \text{if } t > 40 \end{cases} \tag{30}$$

The following initial conditions were considered: $[x, y, z] = [0, 0, 0]\, m$, $[\phi, \theta, \psi] = [0, 0, \frac{\pi}{6}]\, rad$ and all null speeds. Considering a response time of $\tau_r = 5\ s$ for the actuators, the trajectory is filtered by:

$$\mathcal{F}\big(x(k)\big) = x(k) + e^{\frac{-3\tau_s}{\tau_r}} \cdot \big(x(k-1) - x(k)\big) \tag{31}$$

Finally, a sampling time $\tau_s = 50\ ms$ is considered for both applications.

6.1 Unconstrained MPC

In this first approach, the prediction horizon is set to $N = 100$ steps, which is equivalent to $5\ s$ with the given sampling time. Furthermore, we take the entire state vector (ξ) as regulated variables so: $C_r = \mathbb{I}_n$. This way, the controller can track the trajectory set in Eq. (30) and the remaining regulated states are given a null set point. This allows us to stipulate the kind of response the quadrotor is going to give, through adjustments in the weighting matrices Q_y and Q_u. These are set as diagonal matrices:

$$Q_y = diag(K_y) \in \mathbb{R}^{n \times n}; \quad Q_u = diag(K_u) \in \mathbb{R}^{n_u \times n_u} \tag{32}$$

where the main diagonal entries are:

$$K_y = \left[\, 1\ 1\ 1\ 1\ 10\ 1\ 1\ 1\ 1\ 1\ 10\ 1 \right]^T; \qquad K_u = \left[10^{-3}\ 10^{-3}\ 10^{-3}\ 10^{-3}\right]^T \tag{33}$$

Notice these give a lower priority to the control input deviation and higher costs for deviations in z and ψ.

Figure 2 shows the quadrotor's simulated positions and attitude along with their reference trajectories. The system tracks well the trajectory while keeping ψ stabilized in zero.

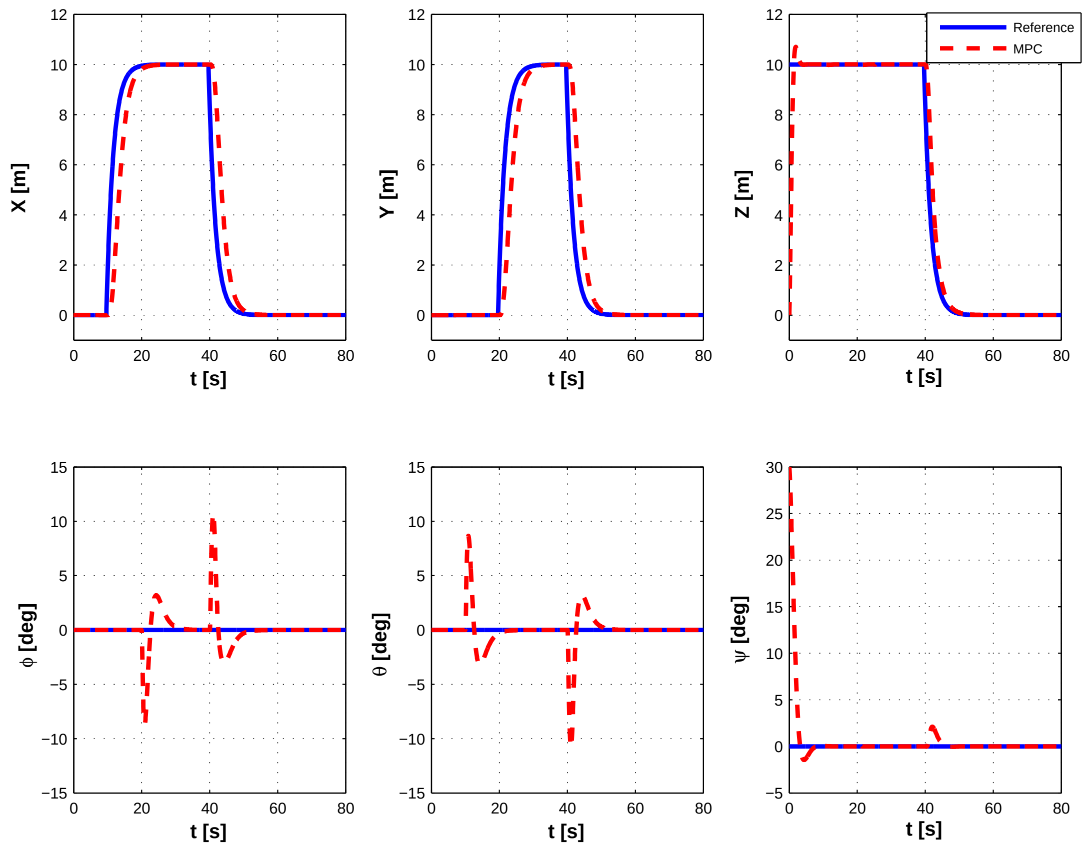

Figure 2: Time evolution of the position and attitude variables along with their respective desired trajectories for the unconstrained MPC simulation.

The control inputs, i.e. the rotor speeds, are depicted in Fig.3 .

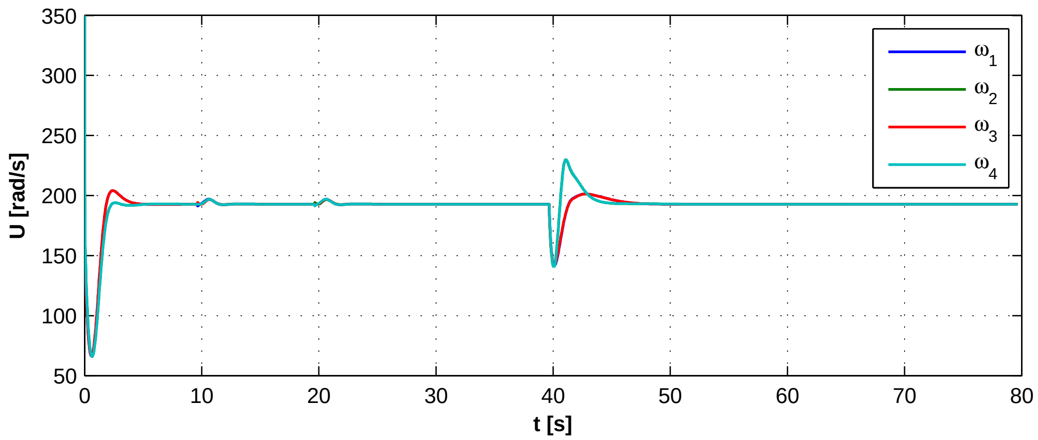

Figure 3: Applied control inputs.

6.2 Constrained MPC

In this more elaborate strategy, the prediction horizon is set to $N = 50$ steps. The regulated output vector is defined as $y_r = \begin{bmatrix} x \; y \; z \; \psi \end{bmatrix}$. To evaluate these outputs' and the control inputs' deviations, the following weighting matrices are stipulated:

$$Q_y = diag(10, 10, 100, 100) \; ; \quad Q_u = diag(0.01, 0.01, 0.01, 0.01) \tag{34}$$

To enforce more stability to the flight, constraints are imposed to its attitude and control inputs. These should keep the quadrotor in a near-hover operation, in which the prediction map (defined in *Section* 3.1) maintains better accuracy. The constrained outputs are thus $y_c = \begin{bmatrix} \phi & \theta \end{bmatrix}^T$. The following bounds are applied:

$$y_c^{min} \begin{bmatrix} -5 \\ -5 \end{bmatrix} deg; \quad y_c^{max} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} deg; \quad \delta^{min} = \begin{bmatrix} -10 \\ -10 \\ -10 \\ -10 \end{bmatrix} rad/s; \quad \delta^{max} = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} rad/s \tag{35}$$

Furthermore, to guarantee the control actions are feasible to the actuators, its minimum and maximum values are set as:

$$u_{min} = \begin{bmatrix} 0 \; 0 \; 0 \; 0 \end{bmatrix}^T \; ; \quad u_{max} = \begin{bmatrix} 250 \; 250 \; 250 \; 250 \end{bmatrix}^T rad/s \tag{36}$$

To solve the QP problem involved, as stated in Eq. (27), we use the open-source software package qpOASES, that is available in (Ferreau et al. 2007–2015). This solver uses an active-set method that is particularly suitable for MPC applications, as described in (Ferreau et al. 2014).

Figure 4 shows the position and attitude of the simulated flight. Once again, the controller is able to track the trajectory while keeping ψ stabilized in zero. Furthermore, it can be seen that the roll and pitch angles are kept within the defined bounds.

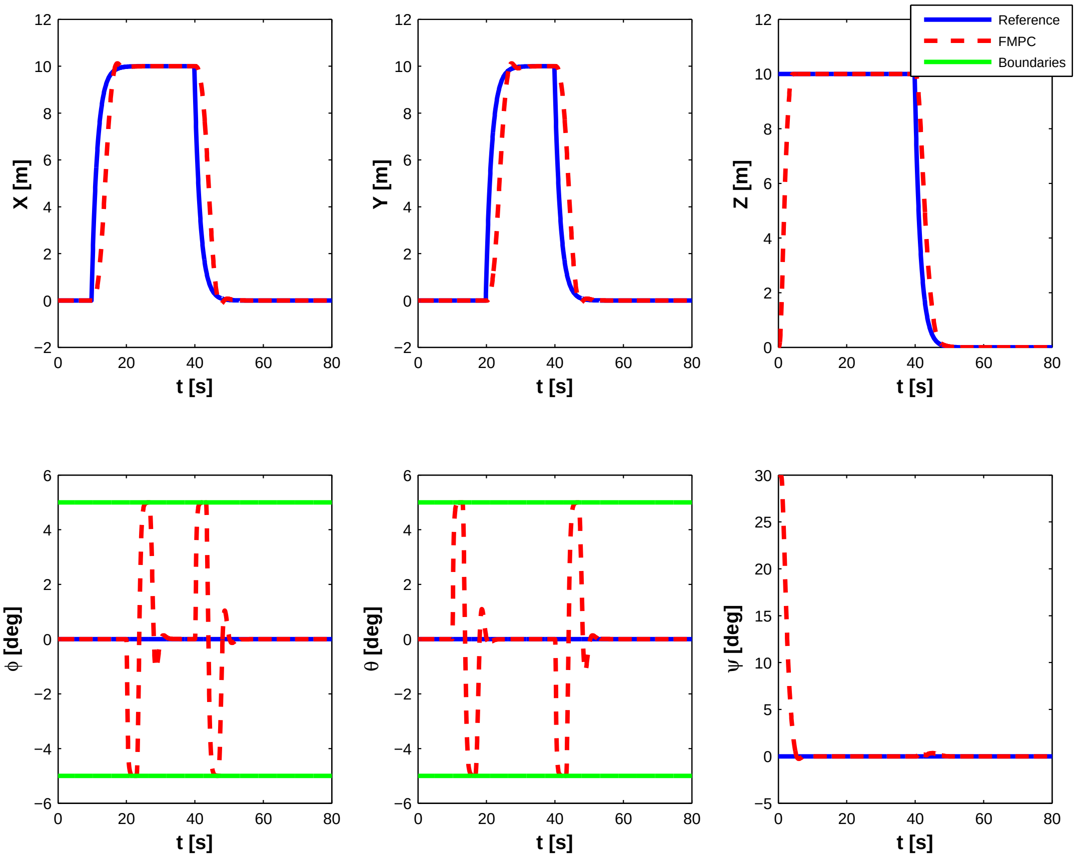

Figure 4: Time evolution of the position and attitude variables along with their respective desired trajectories and bounds for the constrained MPC simulation.

Figure 5 displays the time evolution of the control inputs applied, where its possible to observe the upper bound being enforced during the takeoff.

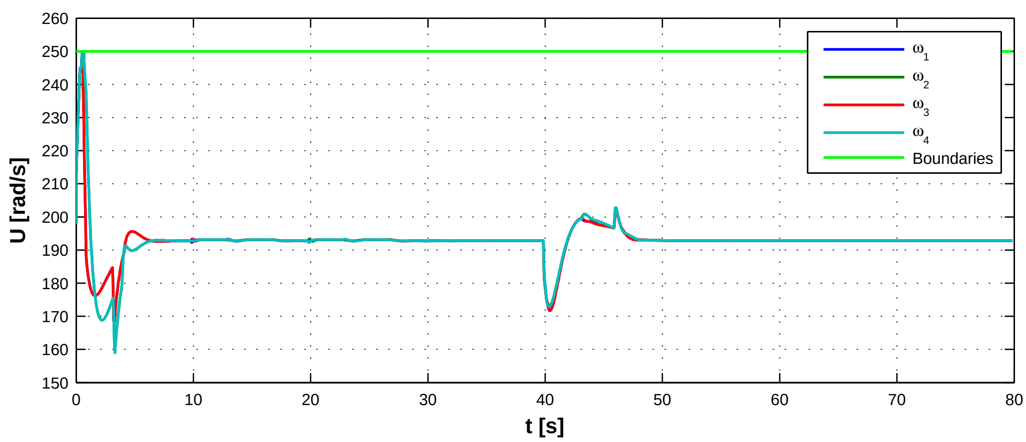

Figure 5: Applied control inputs along with their bounds.

7 CONCLUSION

In this paper, two strategies of MPC were discussed and simulated for a trajectory tracking problem, the main difference between the two being the constraint handling. Both controllers are able to track the given trajectory. However, the constrained MPC is able to perform while keeping the quadrotor in a set of constraints that guarantees more reliability from the prediction map, since it is capable of keeping the UAV closer to horizontal and only plans feasible future control actions. Consequently, the constrained MPC yields smoother control profiles and keeps the velocities feasible for the rotors. On the other hand, the more sophisticated formulation that handles constraints amounts to a QP problem, which requires considerable computational effort. Therefore, real-time application of the constrained strategy requires significantly more processing power. So even though both strategies seem adequate for trajectory tracking with quadrotors, their feasibility depends on other factors. Therefore, the necessity to enforce constraints to the process has to be carefully assessed to deal with the real-time implementation challenges that arise with the higher computational cost required.

Bibliography

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

Burri, M., Nikolic, J., Hurzeler, C., Caprari, G. & Siegwart, R. (2012), Aerial service robots for visual inspection of thermal power plant boiler systems, *in* 'Applied Robotics for the Power Industry (CARPI), 2012 2nd International Conference on', IEEE, pp. 70–75.

Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M. & Scaramuzza, D. (2015), 'Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle', *Journal of Field Robotics* .

Ferreau, H., Kirches, C., Potschka, A., Bock, H. & Diehl, M. (2014), 'qpOASES: A parametric active-set algorithm for quadratic programming', *Mathematical Programming Computation* **6**(4), 327–363.

Ferreau, H., Potschka, A. & Kirches, C. (2007–2015), 'qpOASES webpage', http://www.qpOASES.org/.

Holz, D. & Behnke, S. (2016), Mapping with micro aerial vehicles by registration of sparse *3D* laser scans, *in* 'Intelligent Autonomous Systems 13', Springer, pp. 1583–1599.

Izadi, H. A., Zhang, Y. & Gordon, B. W. (2011), Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation, *in* 'Proc. 2011 Proceedings of the 18th IFAC World Congress', pp. 6343–6348.

Kim, J., Kang, M.-S. & Park, S. (2010), 'Accurate modeling and robust hovering control for a quad–rotor *VTOL* aircraft', *Journal of Intelligent and Robotic Systems* **57**(1-4), 9–26.

Lopes, R. V., Santana, P., Borges, G. & Ishihara, J. (2011), Model predictive control applied to tracking and attitude stabilization of a vtol quadrotor aircraft, *in* '21st International Congress of Mechanical Engineering'.

Luukkonen, T. (2011), 'Modelling and control of quadcopter', *Independent research project in applied mathematics, Espoo* .

Mattingley, J. & Boyd, S. (2012), 'Cvxgen: A code generator for embedded convex optimization', *Optimization and Engineering* **13**(1), 1–27.

Powers, C., Mellinger, D. & Kumar, V. (2015), Quadrotor kinematics and dynamics, *in* 'Handbook of Unmanned Aerial Vehicles', Springer, pp. 307–328.

Richalet, J., Rault, A., Testud, J. & Papon, J. (1978), 'Model predictive heuristic control: Applications to industrial processes', *Automatica* **14**(5), 413–428.

Santana, P. & Borges, G. (2009), 'Modelagem e controle de quadrirrotores', *IX Simpósio Brasileiro de Automaçao Inteligente (SBAI 2009)* **9**, 1–6.

Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M. W., Chli, M., Chatzichristofis, S. A., Kneip, L. et al. (2014), 'Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in *GPS*-denied environments', *Robotics & Automation Magazine, IEEE* **21**(3), 26–40.

Wang, Y. & Boyd, S. (2010), 'Fast model predictive control using online optimization', *Control Systems Technology, IEEE Transactions on* **18**(2), 267–278.

Zulu, A., John, S. et al. (2014), 'A review of control algorithms for autonomous quadrotors', *Open Journal of Applied Sciences* **4**(14), 547.