Iterated non-Linear Model Predictive Control based on Tubes and Contractive Constraints

M.Murillo[a,*], G.Sánchez[a], L.Giovanini[a]

[a] *Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL/CONICET, Ciudad Universitaria UNL, 4° piso FICH, (S3000) Santa Fe, Argentina.*

Abstract

This paper presents a predictive control algorithm for non-linear systems based on successive linearizations of the non-linear dynamic around a given trajectory. A linear time varying model is then obtained and the non-convex constrained optimization problem is transformed into a sequence of locally convex ones that can be solved using standard quadratic programming techniques. The robustness of the proposed algorithm is addressed adding a convex contractive constraint that forces the cost function to remain constant or to decrease within the current time instant, determining an upper bound for the performance index. To account for linearization errors and to obtain more accurate results an inner iteration loop is also added to the algorithm. Also, a simple methodology to obtain an outer bounding-tube for state trajectories is presented. The convergence of the iterative process and the stability of the closed-loop system are analysed. The simulation results show the effectiveness of the proposed algorithm in controlling a quadcopter type unmanned aerial vehicle.

Keywords: contractive constraint, iterative process, non-linear model predictive control, robust model predictive control

---

*Corresponding author

Email addresses: mmurillo@sinc.unl.edu.ar (M.Murillo), gsanchez@sinc.unl.edu.ar (G.Sánchez), lgiovanini@sinc.unl.edu.ar (L.Giovanini)

1. Introduction

Model predictive control (MPC) refers to a class of algorithms in which models of the plant are used to predict the future behaviour of the system over a prediction horizon. It is formulated by solving an on-line optimization problem. The optimal control input sequence is calculated by minimizing an objective function subject to constraints. Only the first element of the computed optimal control input is applied to the plant according to a receding horizon strategy [1, 2]. Linear MPC has been successfully applied in a variety of cases due to its ability to explicitly incorporate the system model and state/inputs constraints into the control calculation [3–6].

In the last few decades, MPC principles have been extended to non-linear processes yielding to non-linear model predictive control (NMPC). The use of general non-linear programming (NLP) techniques to solve the NMPC problem has been proposed in several works [7–10]. However, the solution methods based on NLP present some drawback. First, these algorithms are computationally demanding, as they require to solve on-line a non-linear optimization problem. Moreover, the constraints introduced by the non-linear model dynamics yields to non-convex optimization problems.

Linearization and linear approximation have been adopted in a variety of works to overcome the computational complexity problem [11, 12]. The main advantage of these methods lie in the fact that the model used in the prediction calculation is a set of local linear approximation of the dynamics of the plant, thus converting the non-linear optimization problem into a set of locally convex ones, as it is done in [13–15]. However, linear predictive control techniques do not automatically ensure the stability of the closed-loop system. This issue has been studied by numerous researchers for many years (see [11, 16] for an overview). One way to address the stability problem is to add a contractive constraint to the optimization problem. This idea was firstly introduced by *Yang* and *Polak* [17] and the stability proof was developed by *De Olivera* and *Morari* [18]. In this approach, the authors propose to add a contractive constraint that forces the system states to decrease at each time step. To the best of our knowledge, there are few works that address the addition of such contractive constraint and also this constraint has only been used to contract the system states.

In this paper we present a novel robust predictive control algorithm for non-linear systems. The proposed algorithm uses a linearization process along pre-defined trajectories that transform the non-convex optimization problem into a set of locally convex ones, which can be solved using the standard quadratic programming (QP) techniques. Here, to address stability and convergence issues, the addition of a set of contractive constraints to the optimization problem is analysed. These constraints force the cost functions to decrease or (at least) to remain constant within the current time instant, thus allowing to take into account disturbances and determining an upper bound of the cost functions value. Moreover, an inner iteration loop is added to the proposed algorithm to account for linearization errors and to obtain more accurate results.

The organization of this paper is as follows: in Section 2 the formulation of the NMPC algorithm with the addition of the contractive constraint is presented. In Section 3 a simple methodology to obtain an outer bounding-tube for state trajectories is analysed. In Section 4 an inner iteration loop is added to the previous algorithm. Simulation results are shown in Section 5. Finally, conclusions are discussed in Section 6.

2. Non-linear Model Predictive Control Formulation

Consider the discrete non-linear system

$$x_{k+1} = f\left(x_k, u_k, d_k\right) \tag{1}$$

where $x_k = x(k) \in \Re^n$, $u_k = u(k) \in \mathcal{U} \subseteq \Re^m$ and $d_k = d(k) \in \mathcal{D} \subseteq \Re^l$ are the state vector, the control input vector and the bounded disturbance vector, respectively, $\mathcal{U}$ is the input constraint set and $f(\cdot)$ is a continuous and differentiable vector function that describes the dynamics of the system.

The non-linear model predictive control problem is formulated as a regulatory problem stated as follows:
For a given[1] disturbance sequence

$$\mathbf{d}_k = \left[d_{k|k}, \cdots, d_{k+N-1|k}\right]^T, \tag{2}$$

find at each time instant k, a control input sequence

$$\mathbf{u}_k = \left[u_{k|k}, \cdots, u_{k+N-1|k}\right]^T, \tag{3}$$

[1]If $\mathbf{d}_k$ is not available, the most common assumption is $\mathbf{d}_{k+i} = \mathbf{d}_{k+i-1}$, $i = 1, \cdots, N$

and predicted state sequence

$$\mathbf{x}_k = \left[x_{k+1|k}, \cdots, x_{k+N|k}\right]^T, \tag{4}$$

over a prediction horizon of N sampling intervals, such that

$$\begin{aligned} &\min_{\mathbf{u}_k \in \mathcal{U}} \ \mathcal{J}(k) \\ &\text{st.} \quad x_{k+1} = f(x_k, u_k, d_k). \end{aligned} \tag{5}$$

The vectors $d_{k+i|k}$, $u_{k+i|k}$ and $x_{k+i|k}$ in Eqs. (2) to (4) represent the disturbance, input and state vectors at time $k+i$ that are predicted using the information available at time k.[2] The optimal solution of the problem (5) is denoted here as:

$$\mathbf{u}_k^* = \left[u_{k|k}^*, \cdots, u_{k+N-1|k}^*\right]^T. \tag{6}$$

Regardless the cost function $\mathcal{J}(k)$ is convex or not, the optimization problem (5) is non-convex due to the non-linearity of the system dynamics, and the computational effort is a major issue in its on-line implementation. If $\mathcal{J}(k)$ is chosen to be a quadratic cost function, then the convexity of (5) can be recovered by approximating the non-linear model (1) with a linear time-varying (LTV) one [19, 20], which can be obtained linearizing the system around a desired state and input trajectory $\mathbf{x}_k^r$, $\mathbf{u}_k^r$, where

$$\mathbf{x}_k^r = \left[x_{k+1|k}^r, \cdots, x_{k+N|k}^r\right]^T, \tag{7}$$

and

$$\mathbf{u}_k^r = \left[u_{k|k}^r, \cdots, u_{k+N-1|k}^r\right]^T. \tag{8}$$

Assuming that a reference perturbation $d_{k+i|k}^r$, $i = 0, \cdots, N-1$ is given or estimated, then the dynamic behavior of the deviation from the desired trajectory can be written as an LTV model

$$\tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{u_{k|k}}\tilde{u}_{k|k} + B_{d_{k|k}}\tilde{d}_{k|k}, \tag{9}$$

where

$$\tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \qquad \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r \ \text{ and } \ \tilde{d}_{k|k} = d_{k|k} - d_{k|k}^r. \tag{10}$$

[2] When it clearly refers to current time k, the time dependency at which the information is available will be omitted, i.e. $(\cdot)_{k+i|k} = (\cdot)_{k+i}$

The matrices $A_{k|k}$, $B_{u_{k|k}}$ and $B_{d_{k|k}}$, are the Jacobian matrices of the discrete non-linear system (1), and they are defined as follows

$$A_{k|k} = \left.\frac{\partial f(x_k,u_k,d_k)}{\partial x_k}\right|_{(*)}, \quad B_{u_{k|k}} = \left.\frac{\partial f(x_k,u_k,d_k)}{\partial u(k)}\right|_{(*)}, \quad B_{d_{k|k}} = \left.\frac{\partial f(x_k,u_k,d_k)}{\partial d(k)}\right|_{(*)}, \tag{11}$$

where $(*)$ stands for (x_k^r, u_k^r, d_k^r). In terms of the LTV system (9), the following quadratic objective function $\mathcal{J}(k)$, commonly used in the literature, is adopted:

$$\mathcal{J}(k) \ = \sum_{i=0}^{N-1} \left[\tilde{x}_{k+i|k}^T Q \tilde{x}_{k+i|k} + \tilde{u}_{k+i|k}^T R \tilde{u}_{k+i|k} \right] + \tilde{x}_{k+N|k}^T P_{k|k} \tilde{x}_{k+N|k}, \tag{12}$$

where $Q, R, P_{k|k}$ are positive definite matrices; $P_{k|k}$ is the terminal weight matrix that is chosen so as it satisfies the Lyapunov equation

$$P_{k|k} - A_{k|k}^T P_{k|k} A_{k|k} = Q. \tag{13}$$

As a result, the non-convex optimization problem (5) can be rewritten as a convex optimization problem as follows:

$$\begin{aligned} &\min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \ \mathcal{J}(k) \\ &\text{st.} \ \left\{ \begin{array}{l} \tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{uk|k}\tilde{u}_{k|k} + B_{dk|k}\tilde{d}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \\ \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r, \\ \tilde{d}_{k|k} = d_{k|k} - d_{k|k}^r. \end{array} \right. \end{aligned} \tag{14}$$

In Algorithm 1 the NMPC receding horizon control technique is summarized.

Algorithm 1: NMPC Algorithm

Given $Q, R > 0$, $x_{k|k}$ the initial condition.

Step 1: Obtain the linearization trajectory $\mathbf{x}_k^r$, $\mathbf{u}_k^r$ using as initial condition $\mathbf{u}_k^0 = [u^*_{k|k-1}, u^*_{k+1|k-1}, \cdots, u^*_{k+N-2|k-1}, 0]^T$ and estimate d_{k+i} for $i = 0, \cdots, N-1$

Step 2: Obtain the LTV system (9) and $P_{k|k}$ solving (13)

Step 3: Compute the optimal control input sequence $\tilde{\mathbf{u}}_k^*$ solving (14)

Step 4: Update $\mathbf{u}_k^* \leftarrow \mathbf{u}_k^r + \tilde{\mathbf{u}}_k^*$

Step 5: Apply $u_{k|k} = u^*_{k|k}$ to the system

Step 6: Move the horizon forward to the next sampling instant $k \leftarrow k+1$ and go back to **Step 1**

Linearization techniques are the most straightforward ways to adapt linear control methods to non-linear control problems. In absence of perturbations and linearization errors, the Algorithm 1 will guarantee the closed-loop stability.

Assumption 1. The LTV system (9) is stabilizable for $\mathbf{u}_k \in \mathcal{U}$.

Assumption 2. The prediction horizon N is chosen sufficiently long.

Assumption 3. There are no perturbations, i.e. $d_{k+i} = 0$, $i = 0, \cdots, N-1$.

Theorem 1. *Let assumptions 1 - 3 hold. If the optimization problem (14) solved using Algorithm 1 is feasible, then the origin is an exponentially stable equilibrium point.*

Proof. See Appendix 8.A. □

Although assumption 1 establishes that the prediction horizon N should be long enough, for engineering applications this horizon should be actually chosen as small as possible in order to reduce the workload of online calculation. Consequently, the stability of the system should be ensured using a different argument (see for instance [15, 16, 18]). Moreover, if disturbances are present Theorem 1 might not be satisfied because the contractivity of the cost function cannot be guaranteed at the successive time instants. To address this problem, we propose to add the convex contractive constraint

$$\mathcal{J}(k) \leq \mathcal{J}_0(k), \tag{15}$$

to the optimization problem (14), where $\mathcal{J}_0(k)$ denotes the cost function evaluated for the initial solution

$$\mathbf{u}_k^0 = [u^*_{k|k-1}, u^*_{k|k-1}, \cdots, u^*_{k+N-2|k-1}, 0]^T. \tag{16}$$

at iteration k. Note that this constraint forces the cost function to remain constant or to decrease within the current time instant, thus determining an upper bound for $\mathcal{J}(k)$. Then, the new optimization problem can be stated as follows:

$$\begin{aligned} &\min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \mathcal{J}(k) \\ &\text{st.} \quad \begin{cases} \tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{k|k}\tilde{u}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x^r_{k|k}, \\ \tilde{u}_{k|k} = u_{k|k} - u^r_{k|k}, \\ \mathcal{J}(k) \leq \mathcal{J}_0(k). \end{cases} \end{aligned} \tag{17}$$

As the contractive constraint (15) is defined at the current time instant, if any perturbation occur the value of $\mathcal{J}(k)$ can increase (only at time k) but then it is forced to decrease or to remain constant. The optimization problem (17) can be seen as a multi-objective problem, where the constraint (15) is used to guarantee the stability of the closed-loop system and $\mathcal{J}(k)$ is used to measure the performance of the closed-loop system.

Theorem 2. *If the optimization problem (17) solved using Algorithm 1 is feasible, then the closed-loop system is stable.*

Proof. See Appendix 8.B. □

Remark 1. Note that in the absence of perturbations, the constraint (15) guarantees the contractivity of the cost function at successive time instants, i.e.

$$\mathcal{J}^*(k) \leq \mathcal{J}(k) \leq \mathcal{J}_0(k) \leq \mathcal{J}^*(k-1) \leq \mathcal{J}(k-1). \tag{18}$$

Remark 2. As the stability of the system is guaranteed, the prediction horizon N can be reduced, consequently lowering the workload of online calculation (see for instance the simulation example of Section 5.1).

Remark 3. The addition of the constraint (15) is equivalent to the addition of an input constraint on $\mathbf{u}_k$, hence if the system is stabilizable with $\mathbf{u}_k \in \mathcal{U}$, then the initial feasibility is guaranteed and using the argument of recursive feasibility, the contractive constraint (15) does not affect original feasibility [16].

3. Robust Non-linear Model Predictive Control

The design of robust control algorithms have been studied for many years because such algorithms have the ability to handle system parametric and structural uncertainties (modeled as bounded disturbances) during the system operation. One possible way of accounting for robustness in NMPC algorithm consists in evaluating at each sampling instant all the possible system state trajectories for a given (or estimated) disturbance. This can be done solving an optimization problem that considers the different states trajectories, i.e.:

$$\min_{\tilde{\mathbf{u}}_k^l \in \mathcal{U}} \mathcal{J}(k)$$
$$\text{st.} \quad \begin{cases} \tilde{x}^l_{k+1|k} = A^l_{k|k}\tilde{x}^l_{k|k} + B^l_{k|k}\tilde{u}^l_{k|k}, \\ \tilde{x}^l_{k|k} = x^l_{k|k} - x^r_{k|k}, \\ \tilde{u}^l_{k|k} = u^l_{k|k} - u^r_{k|k}, \\ \mathcal{J}^l(k) \leq \mathcal{J}^l_0(k), \end{cases} \tag{19}$$

where $l = 1, \cdots, m$ stands for the different system realizations regarding the given disturbance. As a result, it can be thought that each state trajectory defines an edge of a time varying polytope [21, 22]. This polytope can be used to generate a tube which actually contains all the possible state trajectories. Tubes have been widely used to bound uncertainties [21, 23–25]. However, the determination of an exact tube for non-linear systems is very difficult.

In this work, the LTV system (9) is obtained by a first order Taylor series expansion. To measure the deviation between the LTV system and the non-linear one, the second order Taylor remainder is used to bound these linearization errors. Instead of obtaining the sequence of all state trajectories $\mathbf{x}^l_k$, $l = 1, \cdots, m$, we propose to use the Taylor remainder to compute state trajectory sequence with the worst uncertainty $\mathbf{x}^{\triangle}_k$. This trajectory can then be used to determine an outer bounding-tube that contains all the state trajectories. Finally, this tube is used to guarantee the stability of the closed-loop system. The proposed procedure is explained below.

The non-linear system (1) can be approximated exactly with an LTV model if the second order Taylor reminder $R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k)$[3] is added to the RHS of (9)

$$\tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{k|k}\tilde{u}_{k|k} + R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k). \tag{20}$$

From equation (20) it can be seen that the term $R_1(\cdot)$ acts as an additive disturbance. This term can be maximized[4] in order to obtain $\mathbf{x}_k^{\triangle}$, which is the state trajectory sequence with the worst uncertainty. Once $\mathbf{x}_k^{\triangle}$ is obtained, its associated cost $\mathcal{J}^{\triangle}(k)$ can be computed. Then, the stability condition for the whole problem can be established if this cost function is forced to decrease. This can be done adding the following contractive constraint

$$\mathcal{J}^{\triangle}(k) \leq \mathcal{J}_0^{\triangle}(k), \tag{21}$$

to the optimization problem (17). Finally the proposed robust control problem to be solved is

$$\min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \mathcal{J}(k)$$
$$\text{st.} \begin{cases} \tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{k|k}\tilde{u}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x^r_{k|k}, \\ \tilde{u}_{k|k} = u_{k|k} - u^r_{k|k}, \\ \mathcal{J}(k) \leq \mathcal{J}_0(k), \\ \mathcal{J}^{\triangle}(k) \leq \mathcal{J}_0^{\triangle}(k), \end{cases} \tag{22}$$

where $\mathcal{J}_0^{\triangle}(k)$ denotes the cost functions $\mathcal{J}^{\triangle}(k)$ evaluated for the initial condition $\mathbf{u}_k^0$.

By including the contractive constraint (21) the stability of the origin is guaranteed as $\mathcal{J}^{\triangle}(k)$ is forced to decrease (or to remain constant). Moreover, as $\mathcal{J}^{\triangle}(k)$ is pushed to zero it actually contract the nominal cost. Following similar arguments as that used in the proof of Theorem 2 and in (18), it can be shown that if the optimization problem (22) solved using Algorithm 1 is feasible, then the origin is an exponentially stable equilibrium point.

Remark 4. Note that as $\mathcal{J}_0^{\triangle}(k)$ is a relaxed upper bound (see for instance Figs. 2(c) and 4(b)), then it is surely bigger than $\mathcal{J}^{\triangle}(k)$, thus not affecting the feasibility of the optimization problem (22).

[3] $R_1(\cdot)$ can be obtained as in [26]

[4] If there is no information about $\mathbf{d}_k$ a given value can be assumed

Remark 5. Following the same arguments used in **Remark 3**, it can be deduced that the addition of the contractive constraint (21) in the optimization problem (22) does not affect original feasibility.

It is worth comparing the proposed approach with those described in [24] and [21]. In our work an outer bounding-tube is obtained by simply maximizing $R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k)$ and then computing the state trajectory $\mathbf{x}_k^{\triangle}$. Within this bounding-tube lie all the perturbed system trajectories [23, 27]. Additionally, the stability condition is guaranteed just by the inclusion of the constraint (21). The procedure proposed in [24] by Cannon *et al.* is more complex. The authors solve a multi-parametric optimization problem with many constraints yielding a high computational burden (even for a simple state-space model) and the impossibility to solve the algorithm in real time. On the other hand, in [21] Langson *et al.* propose a method for robust MPC of linear constrained systems with uncertainties. They use (convex) compact polytopes and (convex) closed polyhedrons, which are difficult to handle when there are several resulting regions. Moreover, the tube is defined as a sequence of sets of states and associated time-varying control input law. This is time demanding as they compute all the possible state trajectories to define the tube.

4. Iterated Robust Non-linear Model Predictive Control

When non-linear systems are linearized, linearization errors may appear and they could be large if linearization trajectories are far from the system operating point. To account for these errors, we propose to include an iterative technique [15, 28] in Algorithm 1 in order to improve the performance of the closed-loop system. The proposed iteration works as follows: at each sampling instant, the non-linear system is linearized along a predefined linearization trajectory. The optimal control input sequence is computed and then it is checked if the breaking loop condition is satisfied. If it is not the case, the linearization trajectory is re-computed using the new control input sequence. The non-linear system is re-linearized and the control input sequence is re-computed. This loop is followed until the convergence condition is satisfied. As a result, a more accurate optimal control input sequence $\mathbf{u}_k^*$ is then obtained. In Algorithm 2 the proposed iterated robust NLMPC technique is summarized.

Algorithm 2: Iterated Robust NMPC Algorithm

Given $Q, R > 0$, $x_{k|k}$ the initial condition, q the iteration index.

Step 1: Initialize $\mathbf{u}_k^q = [u^*_{k|k-1}, u^*_{k+1|k-1}, \cdots u^*_{k+N-2|k-1}, 0]^T$

Step 2: Obtain the linearization trajectory $\mathbf{x}_k^q$, $\mathbf{u}_k^q$

Step 3: Obtain the LTV system (9) and $P^q_{k|k}$ solving (13)

Step 4: Compute the optimal control input sequence $\tilde{\mathbf{u}}_k^{*,q}$ solving (22)

Step 5: Update $\mathbf{u}_k^{*,q} \leftarrow \mathbf{u}_k^q + \tilde{\mathbf{u}}_k^{*,q}$

Step 6: ***if*** $\left\|\mathbf{u}_k^{*,q} - \mathbf{u}_k^{*,q-1}\right\|_\infty \leq \epsilon$

$\mathbf{u}_k^* \leftarrow \mathbf{u}_k^{*,q}$,
$k \leftarrow k+1$
$q \leftarrow 0$

else

$q \leftarrow q+1$
Update $\mathbf{u}_k^q = \mathbf{u}_k^{*,q-1}$
Go back to **Step 2**

end

Step 7: Apply $u_{k|k} = u^*_{k|k}$ to the system and go back to **Step 1**

As the optimization problem to be solved in Algorithm 2 includes the contractive constraints (15) and (21), the stability of the algorithm is guaranteed. Consequently, the iteration process can be stopped at any time, thus improving the online computational burden.

Theorem 3. *The iteration loop of Algorithm 2 converges to the optimal value.*

Proof. See Appendix 8.C. □

5. Simulation Examples

In this section simulation examples are shown. Using the quadcopter model described in Appendix D and the iterated robust NMPC technique of Section 4 two autonomous maneuvers are performed. To evaluate the performance of the proposed controller, simulations with different horizons are also performed.

5.1. *First Example: Climbing Up, Moving Forward and Landing with Colored Wind Gusts*

The first maneuver to be tested is the following: first the quadcopter starts climbing up with an altitude rate $\dot{h} = 0.15\,[\mathrm{m/\,sec}]$. At approximately $t = 10\,[\mathrm{sec}]$, the vehicle starts moving forward along the x-axis. When $t = 20\,[\mathrm{sec}]$, the quadcopter reaches the desired altitude $h_{sp} = 3\,[\mathrm{m}]$ and it keeps moving forward for about 5 seconds longer. When $t = 25\,[\mathrm{sec}]$, the vehicle starts a landing maneuver. Finally, after 10 seconds, the quadcopter is back in the ground. It is assumed that the quadcopter flies immersed in colored wind. The forces generated by these wind gusts act at the quadcopter CG position and they vary randomly between $-1.0\,[\mathrm{N}]$ and $1.0\,[\mathrm{N}]$, as it can be seen in Figure 1:

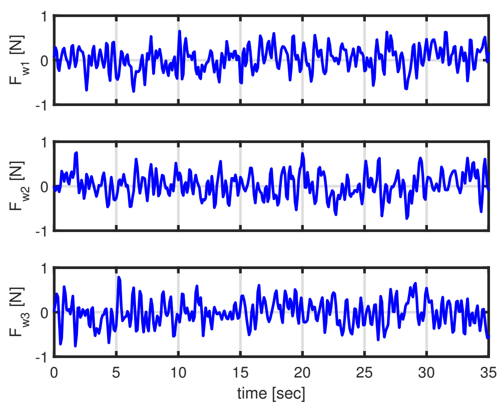


Figure 1: Evolution of colored wind gusts

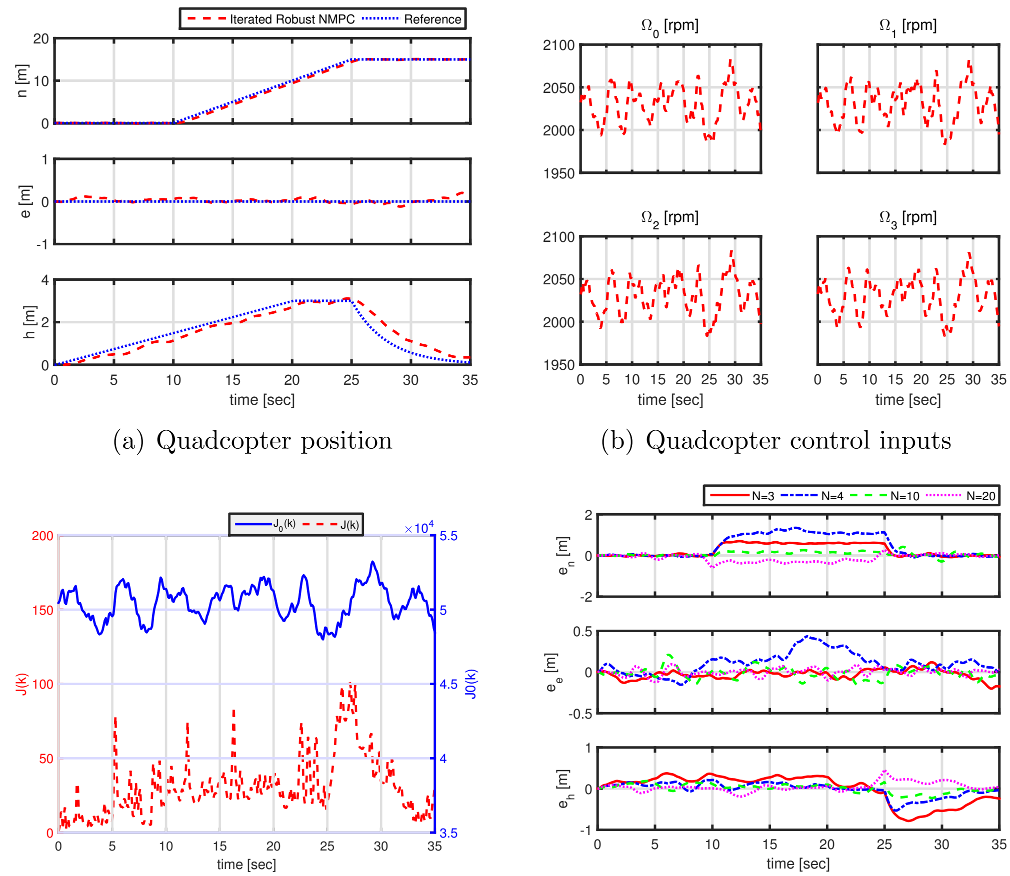
The robust NMPC controller was designed using an horizon $N = 3$ and sampling period $T_s = 0.1\,[\mathrm{sec}]$. The weight matrices Q, R and $P_{k|k}$ were defined as

$$\begin{gathered} Q = \mathrm{diag}(10, 1, 100, 10, 0, 10, 0, 10, 0, 10, 0, 10) \\ R = \mathrm{diag}(0.1, 0.1, 0.1, 0.1) \end{gathered} \tag{23}$$

while $P_{k|k}$ was computed at each sampling interval using (13). Fig. 2(a) shows the quadcopter position[5]. It can be seen that the vehicle starts climbing while moving forward. It reaches the desired altitude and continues

[5]The x-axis points to the north (n), the y-axis points to the east (e) and the z-axis points down (h=-z)

moving along the positive x-axis. Finally, it lands in the ground successfully. Fig. 2(b) depicts the evolution of the computed optimal control inputs. The



(a) Quadcopter position

(b) Quadcopter control inputs

(c) Evolution of $\mathcal{J}(k)$ with initial guess $\mathcal{J}_0(k)$ for $N = 3$

(d) Position errors for different horizons

Figure 2: Climbing up, moving forward and landing maneuver with colored wind gusts

obtained values are physically realizable for a quadcopter. Also, the variation of the four control inputs are similar in shape and in magnitude, which allow to maintain the quadcopter at a stable flight. From Fig. 2(c) it can be seen clearly that the proposed contractive constraint $\mathcal{J}_0(k)$ acts as an upper bound for the cost function $\mathcal{J}(k)$. This constraint is never active because the aim of including $\mathcal{J}_0(k)$ in (22) is to limit the searching space of optimal solutions. It should be noted that despite the value of N was very short, the proposed maneuver was performed successfully. The adopted value in fact corresponds to shortest horizon possible that can be used in a receding horizon control scheme with this quadcopter model (the number of unstable modes plus one [29]). Fig. 2(d) shows the errors in the quadcopter position when larger values of N are used. As it can be seen, the differences between the simulations are small. This is very advantageous as the computational burden of the robust NMPC scheme is reduced if shorter horizons are used.

5.2. Second Example: Spiral Motion with Controlled Yaw Angle

The second maneuver to be tested is a spiral descend motion with controlled yaw angle. For this case, the robust NMPC controller was also designed using an horizon $N = 3$ and sampling period $T_s = 0.1\,[\text{sec}]$. The weight matrices Q, R and $P_{k|k}$ were defined as

$$\begin{aligned} Q &= \mathrm{diag}(100, 1, 100, 10, 0, 10, 0, 10, 0, 10, 0, 10) \\ R &= \mathrm{diag}(0.1, 0.1, 0.1, 0.1) \end{aligned} \tag{24}$$

while $P_{k|k}$ was computed at each sampling interval using (13).

The proposed maneuver, in addition of the spiral descend motion, also controls the quadcopter yaw angle in such a way that the quadcopter x-axis is always aligned with the circumference radius, and as a result the vehicle always ‘looks’ at the center of the spiral. This maneuver would result very useful, for example, if one would use a quadcopter with a fixed-mounted camera to inspect a tower. As it can be seen in Fig. 3, the desired maneuver was performed successfully. The quadcopter achieved the spiral descend motion while the *yaw* angle was controlled in order the quadcopter ‘looks’ at the center of the spiral. Fig. 4(a) shows the evolution of the computed control inputs. It can be seen that the propellers which are opposite to each other have a similar variation. Control inputs practically vary at the beginning and at the end of the maneuver, staying constant while the quadcopter is performing the spiral descent. Fig. 4(b) depicts both the cost function $\mathcal{J}(k)$ and its upper bound $\mathcal{J}_0(k)$. It shows that when the spiral descend is being performed, the cost is constant and when the quadcopter reaches the ground, $\mathcal{J}(k)$ effectively tends to zero.

5.3. Comparison between iterated robust NMPC and classical NMPC techniques

Here, the proposed iterated robust NMPC technique is compared with the classical NMPC technique presented in [28]. To test the performance

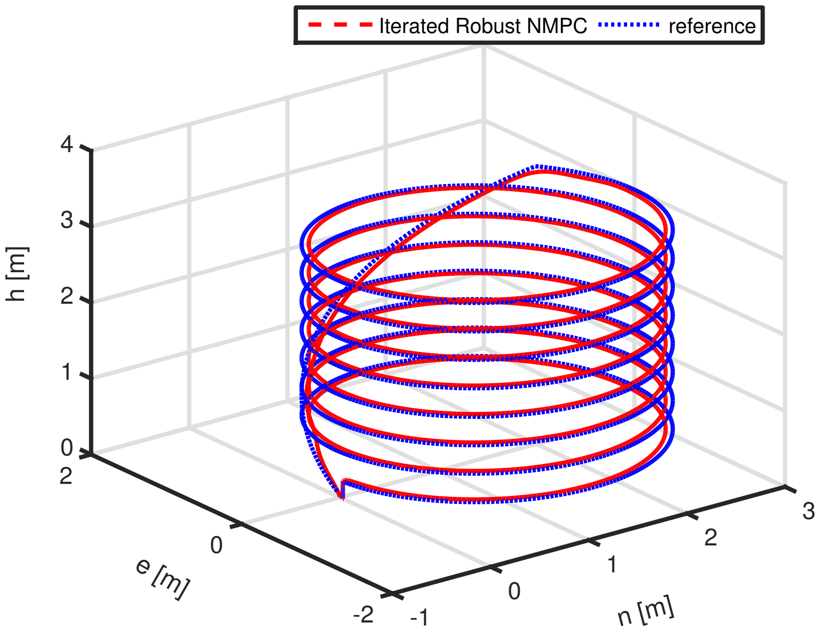
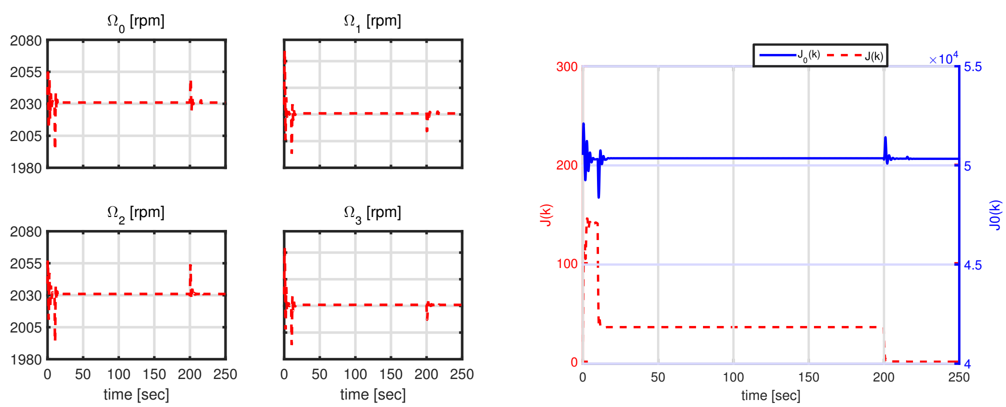


Figure 3: Evolution of the quadcopter position



(a) Quadcopter control inputs

(b) Evolution of $\mathcal{J}(k)$ with initial guess $\mathcal{J}_0(k)$ for $N = 3$

Figure 4: Spiral motion with controlled yaw angle

of our algorithm, we simulated the maneuver presented in Section 5.1 using both algorithms. In Figure 5 it can be seen the errors in the quadcopter position. The results suggest that when the value of N is maintained and the contractive contraints are not added to the optimization problem, then the errors in the quadcopter position are increased. However, in order to obtain a similar response as the one obtained with the iterated robust NMPC, we had to use the NMPC technique with a larger value of N, thus increasing the online computational workload. On the other hand, the proposed algorithm could be executed within a maximum of three iterations but as the stability of the closed-loop system is guaranteed, the iteration loop could have been stopped with fewer iterations, thus reducing even more the online computational workload.

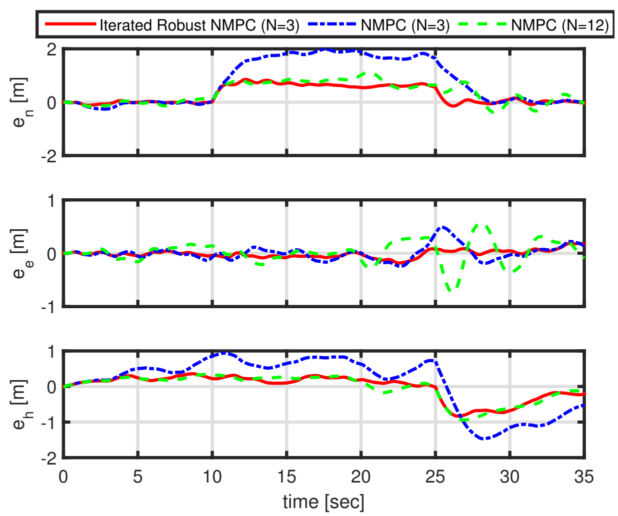


Figure 5: Comparison with the standard NMPC technique

Additionally, we have also compared our algorithm with the one proposed by *Cannon et al.* [24]. We performed the same maneuver as before using both algorithms. Similar results were obtained when a large horizon was used with Cannon's algorithm. Moreover, we found that the online computational burden for this algorithm is three or four times higher than that obtained with our algorithm.

Consequently, because of the presented results the iterated robust NMPC algorithm may be an useful tool for real time simulations as it allows to obtain acceptable responses at lower computational burden.

6. Conclusion

In this paper, a robust non-linear model predictive control technique was presented. The proposed technique is based on the linearization of non-linear systems along pre-defined state trajectories and the minimization of a constrained objective function. To guarantee the stability of the closed-loop system we add to the optimization problem a contractive constraint that forces the cost function to decrease (or to remain constant) at the current time instant. This stability can be also guaranteed even with uncertainties. As the stability of the system is guaranteed, the inclusion of this constraint allows to reduce the prediction horizon to its minimum value, thus lowering the computational workload. This may be useful when controlling non-linear systems with fast dynamics such as a quadcopter. The robustness of the proposed NMPC algorithm is achieved by using the Taylor reminder to compute the state trajectory associated to the worst uncertainty. This trajectory can then be used to determine an outer bounding-tube that contains all the system state trajectories. The proposed methodology to obtain the outer bounding-tube for state trajectories seems to be simpler and less computationally demanding. To account for linearization errors and to improve the performance of the closed-loop system we have included an iteration loop in the robust NMPC algorithm, yielding to the iterated robust NMP algorithm.

The iterated NMPC algorithm was used as a central unit that can control a full quadcopter model without the need of decoupling the non-linear system. To evaluate the performance of this algorithm, we have performed the simulation of two autonomous maneuvers, which were performed both successfully. Also, the results were compared with those obtained using larger horizons, having no significant differences between the short horizon adopted and the larger ones. Finally, we have performed a comparison between iterated robust NMPC algorithm and classical NMPC. The results obtained suggest that the proposed algorithm can achieve a similar response to the classical NMPC but using a shorter prediction horizon, thus having a lower computational workload than classical NMPC.

7. Acknowledgments

The authors wish to thank the *Universidad Nacional de Litoral* (with CAID 501201101 00529 LI) and the *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) from Argentina, for their support.

8. Appendices

A. Proof of Theorem 1

Proof. First it is shown that the input and the state converge to the origin, and then it will be shown that the origin is an stable equilibrium point for the close-loop system. The combination of convergence and stability gives asymptotic stability.

Convergence. Convergence of the state and input to the origin can be established by showing that the sequence of cost values is non-increasing.

Let the cost function $\mathcal{J}(k)$ be given by (12), with Q, R and $P_{k|k}$ positive definite matrices; $P_{k|k}$ satisfies the Lyapunov equation.

Let $\mathbf{u}_k^* = [u_{k|k}^*, u_{k+1|k}^*, \cdots, u_{k+N-1|k}^*]^T$ be the optimal control input sequence computed at time k. Assuming that only exists inputs constraints, then the control input sequence $\hat{\mathbf{u}}_{k+1} = [u_{k+1|k}^*, u_{k+2|k}^*, \cdots, u_{k+N-1|k}^*, 0]^T$ is feasible at time $k+1$. As $P_{k|k}$ satisfies the Lyapunov equation, then the cost function (12) approximates exactly the infinite cost problem. Then, evaluating $\mathcal{J}(k)$ for both $\mathbf{u}_k^*$ and $\hat{\mathbf{u}}_{k+1}$, and assuming that there are no perturbations nor linearization errors, it can be shown that

$$\hat{\mathcal{J}}(k+1) - \mathcal{J}^*(k) = -x_{k|k}^T Q x_{k|k} - u_{k|k}^* R u_{k|k}^*, \tag{25}$$

where $\hat{\mathcal{J}}(i)$ and $\mathcal{J}^*(i)$ denote the values of the cost function for $\hat{\mathbf{u}}_i$ and $\mathbf{u}_i^*$, respectively. As the RHS of (25) is semi-negative definite, then

$$\hat{\mathcal{J}}(k+1) \leq \mathcal{J}^*(k). \tag{26}$$

But $\hat{\mathbf{u}}_{k+1}$ is a feasible but sub-optimal sequence, then it can be said that $\mathcal{J}^*(k+1) \leq \tilde{\mathcal{J}}(k+1)$, and consequently

$$\mathcal{J}^*(k+1) \leq \mathcal{J}^*(k) \quad \forall k. \tag{27}$$

This shows that the sequence of optimal cost values $\{\mathcal{J}^*(k)\}$ decreases along closed-loop trajectories of the system. The cost is bounded below by zero and thus has a nonnegative limit. Therefore as $k \to \infty$ the difference of optimal cost $\Delta\mathcal{J}^*(k+1) = \mathcal{J}^*(k+1) - \mathcal{J}^*(k) \to 0$. Because Q and R are positive definite, as $\Delta\mathcal{J}^*(k+1) \to 0$ the states and the inputs must converge to the origin $x_k \to 0$ and $u_k \to 0$ as $k \to \infty$.

Stability. To prove that the origin is asymptotically stable, from (27) it is clear that the sequence of optimal costs $\{\mathcal{J}^*(k)\}$ is non-increasing, which implies $\mathcal{J}^*(k) \leq \mathcal{J}^*(0)\,\forall k > 0$. At time $k = 0$, the cost function can be written as

$$\mathcal{J}(0) = x_0^T P_0 x_0, \tag{28}$$

where P_0 satisfies the Lyapunov equation $P_k - A_k^T P_k A_k = Q$, $Q > 0$. From the definition of cost function, it can be written that

$$x_k^T Q x_k \leq \mathcal{J}^*(k), \tag{29}$$

then,

$$x_k^T Q x_k \leq \mathcal{J}^*(k) \leq \mathcal{J}^*(0) \leq \mathcal{J}(0) = x_0^T P_0 x_0, \tag{30}$$

which implies

$$x_k^T Q x_k \leq x_0^T P_0 x_0 \quad \forall k. \tag{31}$$

Since Q and P_0 are positive definite it follows that

$$\lambda\min(Q)\left\|x_k\right\|^2 \leq \lambda\max(P_0)\left\|x_0\right\|^2 \quad \forall k, \tag{32}$$

where $\lambda\min(\cdot)$ and $\lambda\max(\cdot)$ denote the min-max eigenvalue of the corresponding matrix. Finally it can be written that

$$\|x_k\| \leq \sqrt{\frac{\lambda\max(P_0)}{\lambda\min(Q)}}\,\|x_0\| \quad \forall k > 0. \tag{33}$$

Thus, the closed-loop is stable. The combination of convergence and stability implies that the origin is asymptotically stable equilibrium point of the closed-loop system. □

B. Proof of Theorem 2

Proof. As the cost function (12) is locally convex at each sampling instant and only linear inputs constraints are considered, the optimization problem of Algorithm 1 is locally convex.

Let the control input sequence $\mathbf{u}_k^0$ be a feasible solution at time k defined as:

$$\mathbf{u}_k^0 = [u^*_{k|k-1}, u^*_{k+1|k-1}, \cdots, u^*_{k+N-2|k-1}, 0]^T. \tag{34}$$

At time k, let $\mathbf{u}_k$ be a feasible convex combination of $\mathbf{u}_k^*$ and $\mathbf{u}_k^0$, i.e.

$$\mathbf{u}_k = \alpha \mathbf{u}_k^* + (1-\alpha)\mathbf{u}_k^0, \quad \text{with}\, 0 \leq \alpha \leq 1. \tag{35}$$

As $\mathcal{J}(k)$ is a locally convex function, it can be easily shown that

$$\begin{aligned} \mathcal{J}(k) &= \alpha \mathcal{J}^*(k) + (1-\alpha)\mathcal{J}_0(k), \\ &= \alpha\left(\mathcal{J}^*(k) - \mathcal{J}_0(k)\right) + \mathcal{J}_0(k), \end{aligned} \tag{36}$$

as $0 \leq \alpha \leq 1$ and $\mathcal{J}^*(k)$ is the optimal value of the cost function at time k, then

$$\alpha\left(\mathcal{J}^*(k) - \mathcal{J}_0(k)\right) \leq 0, \tag{37}$$

and consequently

$$\mathcal{J}(k) \leq \mathcal{J}_0(k), \tag{38}$$

This shows that at each time instant k the cost function $\mathcal{J}(k)$ is non-increasing, thus the resulting closed-loop is stable.

□

C. Proof of Theorem 3

Proof. At iteration $q = 1$ let $\mathbf{u}_k^1$ be a feasible convex combination of $\mathbf{u}_k^*$ and $\mathbf{u}_k^0$, i.e.

$$\mathbf{u}_k^1 = \alpha \mathbf{u}_k^* + (1-\alpha)\mathbf{u}_k^0, \quad \text{with}\, 0 \leq \alpha \leq 1. \tag{39}$$

As the iterated cost function

$$\mathcal{J}^q(k) = \sum_{i=0}^{N-1}\left[\tilde{x}_{k+i|k}^{q^T} Q \tilde{x}_{k+i|k}^q + \tilde{u}_{k+i|k}^{q^T} R \tilde{u}_{k+i|k}^q\right] + \tilde{x}_{k+N|k}^{q^T} P_{k|k}^q \tilde{x}_{k+N|k}^q, \tag{40}$$

is a locally convex function, then following a similar reasoning to the proof of Theorem 2, it can be easily shown that

$$\mathcal{J}^1(k) \leq \mathcal{J}^0(k), \tag{41}$$

The same argument can be repeated at subsequent iteration to show that

$$\mathcal{J}^{q+1}(k) \leq \mathcal{J}^q(k), \quad q \geq 0, \tag{42}$$

This shows that the sequence $\{\mathcal{J}^q(k)\}$ is non-increasing. As the cost function is quadratic, it is bounded below by zero and thus has a non-negative limit. Therefore, as $q \to \infty$ the difference of cost $\Delta\mathcal{J}^q(k) = \mathcal{J}^{q+1} - \mathcal{J}^q \to 0$, and as a result $\mathcal{J}^q(k) \to \mathcal{J}^*(k)$. □

D. Non-linear Quadcopter Model

The quadcopter state vector x is defined as:

$$x = [n\,e\,h\,u\,v\,w\,\phi\,\theta\,\psi\,p\,q\,r]^T, \tag{43}$$

where n, e and $h = -z$ are the coordinates of the quadcopter CG position, u, v and w are the components of the quadcopter velocity vector, ϕ, θ and ψ are the Euler angles that define the roll, pitch and yaw movements and p, q and r are the components of the quadcopter angular velocity vector. The quadcopter control inputs vector u is defined as:

$$u = [\Omega_0\,\Omega_1\,\Omega_2\,\Omega_3]^T, \tag{44}$$

where Ω_i denotes the absolute angular speed of the i-th rotor.

Defining c_α, s_α, and t_α as the notation representing $\cos(\alpha)$, $\sin(\alpha)$ and $\tan(\alpha)$, respectively, for a generic angle α, the 6-degrees of freedom (6-DOF) non-linear dynamic of a quadcopter can be represented by the following set of differential equations:

$$\dot{x} = \begin{bmatrix} u\,\mathrm{c}_\theta\,\mathrm{c}_\psi + v(\mathrm{s}_\phi\,\mathrm{s}_\theta\,\mathrm{c}_\psi - \mathrm{c}_\phi\,\mathrm{s}_\psi) + w(\mathrm{c}_\phi\,\mathrm{s}_\theta\,\mathrm{c}_\psi + \mathrm{s}_\phi\,\mathrm{s}_\psi) \\ u\,\mathrm{c}_\theta\,\mathrm{s}_\psi + v(\mathrm{s}_\phi\,\mathrm{s}_\theta\,\mathrm{s}_\psi + \mathrm{c}_\phi\,\mathrm{c}_\psi) + w(\mathrm{c}_\phi\,\mathrm{s}_\theta\,\mathrm{s}_\psi - \mathrm{s}_\phi\,\mathrm{c}_\psi) \\ u\,\mathrm{s}_\theta - v\,\mathrm{s}_\phi\,\mathrm{c}_\theta - w\,\mathrm{c}_\phi\,\mathrm{c}_\theta \\ rv - qw - g\,\mathrm{s}_\theta - \frac{\mu}{m}u - \frac{C\,A_x\rho}{2m}u\,|u| \\ pw - ru + g\,\mathrm{s}_\phi\,\mathrm{c}_\theta - \frac{\mu}{m}v - \frac{C\,A_y\rho}{2m}v\,|v| \\ qu - pv + g\,\mathrm{c}_\phi\,\mathrm{c}_\theta - \frac{b}{m}(\Omega_0^2 + \Omega_1^2 + \Omega_2^2 + \Omega_3^2) - \frac{C\,A_z\rho}{2m}w\,|w| \\ p + q\,\mathrm{s}_\phi\,\mathrm{t}_\theta + r\,\mathrm{c}_\phi\,\mathrm{t}_\theta \\ q\,\mathrm{c}_\phi - r\,\mathrm{s}_\phi \\ q\,\mathrm{s}_\phi \sec\theta + r\,\mathrm{c}_\phi \sec\theta \\ \frac{I_y - I_z}{I_x}qr + \frac{d\,b\sqrt{2}}{2I_x}(-\Omega_0^2 - \Omega_1^2 + \Omega_2^2 + \Omega_3^2) - \frac{k\rho A}{I_x}p + \frac{J_r}{I_x}q(\Omega_0 - \Omega_1 + \Omega_2 - \Omega_3) \\ \frac{I_z - I_x}{I_y}pr + \frac{d\,b\sqrt{2}}{2I_y}(\Omega_0^2 - \Omega_1^2 - \Omega_2^2 + \Omega_3^2) - \frac{k\rho A}{I_y}q - \frac{J_r}{I_y}p(\Omega_0 - \Omega_1 + \Omega_2 - \Omega_3) \\ \frac{I_x - I_y}{I_z}pq + \frac{\epsilon}{I_z}(\Omega_0^2 - \Omega_1^2 + \Omega_2^2 - \Omega_3^2) - \frac{k\rho A}{I_z}r + \frac{J_r}{I_z}(\dot{\Omega}_0 - \dot{\Omega}_1 + \dot{\Omega}_2 - \dot{\Omega}_3) \end{bmatrix}, \tag{45}$$

where $\dot{x}$ is the time derivative of Eq. (43), $g = 9.81\,[\mathrm{m/\,sec}]$ is the acceleration of gravity, $m = 1\,[\mathrm{kg}]$ is the mass of the quadcopter, $I_x = 8.1 \cdot 10^{-3}\,[\mathrm{Nm\,sec^2}]$, $I_y = 8.1 \cdot 10^{-3}\,[\mathrm{Nm\,sec^2}]$ and $I_z = 14.2 \cdot 10^{-3}\,[\mathrm{Nm\,sec^2}]$ are the body moment of inertia around the x, y and z axis, respectively, $\mu = 1 \cdot 10^{-5}\,[\mathrm{kg/\,sec}]$ is the rotor drag coefficient, $C = 3 \cdot 10^{-4}$ is a dimensionless friction constant, $A_x = 0.05\,[\mathrm{m^2}]$, $A_y = 0.05\,[\mathrm{m^2}]$ and $A_z = 0.1\,[\mathrm{m^2}]$ are the projections of the vehicle area on the yz, xz and xy planes of the B-Frame, respectively, $\rho = 1.2\,[\mathrm{kg/m^3}]$ is the air density, $b = 54.2 \cdot 10^{-6}\,[\mathrm{N\,sec^2}]$ is the aerodynamic contribution of thrust, $d = 0.24\,[\mathrm{m}]$ is the distance between the center of the quadcopter and the center of a propeller, $k = 1 \cdot 10^{-5}\,[\mathrm{m^3/\,sec}]$ is a frictional constant, $A = 0.2\,[\mathrm{m^2}]$ is the vehicle area, $J_r = 104 \cdot 10^{-6}\,[\mathrm{Nm\,sec^2}]$ is the rotational inertia of a propeller and $\epsilon = 1.1 \cdot 10^{-6}\,[\mathrm{Nm\,sec^2}]$ is a yaw drag factor.

9. References

[1] J. Maciejowski, Predictive control: with constraints, Prentice Hall, 2002.

[2] J. Rawlings, D. Mayne, Model Predictive Control: Theory and Design, Nob Hill Pub., 2009.

[3] K. Alexis, G. Nikolakopoulos, A. Tzes, Model predictive quadrotor control: attitude, altitude and position experimental studies, Control Theory Applications, IET 6 (12) (2012) 1812–1827. doi:10.1049/iet-cta.2011.0348.

[4] S. Vazquez, J. Leon, L. Franquelo, J. Rodriguez, H. Young, A. Marquez, P. Zanchetta, Model predictive control: A review of its applications in power electronics, Industrial Electronics Magazine, IEEE 8 (1) (2014) 16–31. doi:10.1109/MIE.2013.2290138.

[5] A. Marasco, S. Givigi, C. Rabbath, Model predictive control for the dynamic encirclement of a target, in: American Control Conference (ACC), 2012, 2012, pp. 2004–2009. doi:10.1109/ACC.2012.6315602.

[6] M. Abdolhosseini, Y. Zhang, C. Rabbath, An efficient model predictive control scheme for an unmanned quadrotor helicopter, Journal of Intelligent & Robotic Systems 70 (1-4) (2013) 27–38. doi:10.1007/s10846-012-9724-3.

[7] Y. Lee, B. Kouvaritakis, M. Cannon, Constrained receding horizon predictive control for nonlinear systems, Automatica 38 (12) (2002) 2093–2102. doi:10.1016/S0005-1098(02)00133-4.

[8] M. Diehl, H. Ferreau, N. Haverbeke, Efficient numerical methods for nonlinear mpc and moving horizon estimation, Nonlinear Model Predictive Control (2009) 391–417doi:10.1007/978-3-642-01094-1_32.

[9] R. Lopez-Negrete, F. J. DAmato, L. T. Biegler, A. Kumar, Fast nonlinear model predictive control: Formulation and industrial process applications, Computers & Chemical Engineering 51 (2013) 55 – 64. doi:http://dx.doi.org/10.1016/j.compchemeng.2012.06.011.

[10] L. Biegler, X. Yang, G. Fischer, Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization, Journal of Process Control 30 (2015) 104 – 116. doi:http://dx.doi.org/10.1016/j.jprocont.2015.02.001.

[11] M. Morari, J. H. Lee, Model predictive control: past, present and future, Computers & Chemical Engineering 23 (4-5) (1999) 667–682. doi:10.1016/S0098-1354(98)00301-9.

[12] A. Brooms, B. Kouvaritakis, Successive constrained optimization and interpolation in non-linear model based predictive control, International journal of control 73 (4) (2000) 312–316. doi:10.1080/002071700219669.

[13] W. C. Li, L. T. Biegler, A multistep, newton-type control strategy for constrained, nonlinear processes, in: American Control Conference, 1989, IEEE, 1989, pp. 1526–1527.

[14] N. De Oliveira, Newton-type algorithms for nonlinear constrained chemical process control, Ph.D. thesis, Carnegie Mellon University (1994).

[15] N. De Oliveira, L. Biegler, An extension of newton-type algorithms for nonlinear process control, Automatica 31 (2) (1995) 281–286. doi:http://dx.doi.org/10.1016/0005-1098(94)00086-X.

[16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814. doi:10.1016/S0005-1098(99)00214-9.

[17] T. Yang, E. Polak, Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty, International Journal of Control 58 (4) (1993) 875–903. doi:10.1080/00207179308923033.

[18] S. De Oliveira Kothare, M. Morari, Contractive model predictive control for constrained nonlinear systems, IEEE Transactions on Automatic

Control 45 (6) (2000) 1053–1071. doi:10.1109/9.863592.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=863592

[19] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, H. Tseng, A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems, in: Decision and Control, 2007 46th IEEE Conference on, IEEE, 2007, pp. 2980–2985. doi:10.1109/CDC.2007.4434137.

[20] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, D. Hrovat, Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation, International journal of robust and nonlinear control 18 (8) (2007) 862–875. doi:10.1002/rnc.1245.

[21] W. Langson, I. Chryssochoos, S. Raković, D. Mayne, Robust model predictive control using tubes, Automatica 40 (1) (2004) 125–133. doi:10.1016/j.automatica.2003.08.009.

[22] F. A. Cuzzola, J. C. Geromel, M. Morari, An improved approach for constrained robust model predictive control, Automatica 38 (7) (2002) 1183 – 1189. doi:http://dx.doi.org/10.1016/S0005-1098(02)00012-2.
URL http://www.sciencedirect.com/science/article/pii/S0005109802000122

[23] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, P. Falugi, Tube-based robust nonlinear model predictive control, International Journal of Robust and Nonlinear Control 21 (11) (2011) 1341–1353. doi:10.1002/rnc.1758.

[24] M. Cannon, J. Buerger, B. Kouvaritakis, S. Rakovic, Robust Tubes in Nonlinear Model Predictive Control, IEEE Transactions on Automatic Control 56 (8) (2011) 1942–1947. doi:10.1109/TAC.2011.2135190.

[25] S. Yu, C. Maier, H. Chen, F. Allgöwer, Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems, Systems and Control Letters 62 (2) (2013) 194–200. doi:10.1016/j.sysconle.2012.11.004.
URL http://dx.doi.org/10.1016/j.sysconle.2012.11.004

[26] J. A. Martínez, Model order reduction of nonlinear dynamic systems using multiple projection bases and optimized state-space sampling, Ph.D. thesis, University of Pittsburgh (2009).

[27] S. Rakovic, D. Mayne, Set robust control invariance for linear discrete time systems, in: Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on, 2005, pp. 975–980. doi:10.1109/CDC.2005.1582284.

[28] J. R. Cueli, C. Bordons, Iterative nonlinear model predictive control. Stability, robustness and applications, Control Engineering Practice 16 (9) (2008) 1023–1034. doi:10.1016/j.conengprac.2007.11.003.

[29] J. B. Rawlings, K. R. Muske, The stability of constrained receding horizon control, Automatic Control, IEEE Transactions on 38 (10) (1993) 1512–1516. doi:10.1109/9.241565.