Nonlinear Model Predictive Control with Constraint Satisfactions for a Quadcopter

Ye Wang[1], **Andres Ramirez-Jaime**[2], **Feng Xu**[3] **and Vicenç Puig**[1]

[1] Advanced Control Systems (SAC) Research Group at Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Universitat Politècnica de Catalunya-BarcelonaTech (UPC), C/. Llorens i Artigas 4-6, 08028 Barcelona, Spain

[2] Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1, No. 18A-10, Colombia

[3] Center of Intelligent Control and Telescience, Graduate School at Shenzhen, Tsinghua University, 518055 Shenzhen, P.R. China

E-mail: `ywang@iri.upc.edu`, `af.ramirez236@uniandes.edu.co`, `xu.feng@sz.tsinghua.edu.cn`, `vicenc.puig@upc.edu`

Abstract. This paper presents a nonlinear model predictive control (NMPC) strategy combined with constraint satisfactions for a quadcopter. The full dynamics of the quadcopter describing the attitude and position are nonlinear, which are quite sensitive to changes of inputs and disturbances. By means of constraint satisfactions, partial nonlinearities and modeling errors of the control-oriented model of full dynamics can be transformed into the inequality constraints. Subsequently, the quadcopter can be controlled by an NMPC controller with the updated constraints generated by constraint satisfactions. Finally, the simulation results applied to a quadcopter simulator are provided to show the effectiveness of the proposed strategy.

1. Introduction

The quadcopter has become one of the most popular unmanned aerial vehicles (UAVs), which attracts considerable attention in both academia and industry. Due to its simple structure and flexible flight ability, the quadcopter is playing a significant role in applications such as video surveillance, agricultural service, mapping, goods delivery and other military applications [1, 2, 3].

The motion of a quadcopter relies on the rotation of four propellers to generate thrusts and to control its attitude and position. In general, a simultaneous increasing/decreasing of angular speed of four propellers will make the quadcopter fly upward/downward. When keeping the vertical component of total thrust from four propellers equal to the gravity of quadcopter, the quadcopter can be controlled to fly forward/backward or /leftward/rightward by increasing the angular speed of one propeller and simultaneous decreasing the angular speed of the propeller on its centrosymmetric position, respectively. Additionally, simultaneous increasing the speed of one pair of propellers centrosymmetric to each other and decreasing the speed of the other pair can control the quadcopter to rotate in horizontal plane. However, due to the nonlinear dynamics and high degrees of couplings between different control channels, to achieve high-performance control of quadcopter is still an issue full of challenges.

The control problem of quadcopter has been widely investigated by using a variety of approaches such as proportional-integral-derivative (PID) control, backstepping control, feedback linearization, linear quadratic regulator and model predictive control (MPC). In [4, 5, 6], the PID control was used to control the quadcopter with the objective of achieving robustness and fault tolerance. The application of backstepping control and feedback linearization to the quadcopter could be found in [7, 8, 9, 10]. Related to the topic of the current paper, we can also find MPC control of quadcopter in [2, 11, 12], where a common feature of these works is that they are based on the linear MPC techniques.

Commonly, the quadcopter is normally considered as a system with high nonlinearities and fast dynamics. Although it is possible to achieve some control objectives by using a linearized model or a group of linearized models to approximate the nonlinear dynamics of quadcopter, we have to face a degree of performance degradation. Motivated by this point, we propose to directly use the nonlinear MPC technique to control the quadcopter. In this case, we can not only simultaneously obtain better performance but also make use of the inherent capability of MPC, i.e., explicit constraint handling, which is very helpful to deal with the constraints on states, inputs and outputs. Additionally, turning to the constraint-handling capability, MPC is quite convenient to be used for tolerance of faults in actuators (i.e., motors of quadcopter) by adjusting its input constraints, which is also of vital important interest of many works [13, 14].

The main contribution of this paper is to present a nonlinear MPC (NMPC) strategy combined with constraint satisfactions for a quadcopter. Paritial nonlinear constraints and modeling errors are updated through the procedure of constraint satisfactions. Then, the NMPC strategy is implemented by solving a nonlinear optimization problem through the nonlinear programming technique. As a result, simulation results by applying the proposed NMPC strategy to a quadcopter simulator implemented in MATLAB environment are shown in order to verify the effectiveness of the proposed MPC strategy. Besides, comparison results show the performance of the proposed MPC strategy.

The remainder of this paper is organized as follows: The nonlinear dynamics and control-oriented model of the quadcopter are briefly introduced in Section 2. The constraint satisfaction for the quadcopter control is addressed in Section 3. In Section 4, the NMPC design for the quadcopter is presented. The simulation results as well as comparisons results with different uncertain cases based on a quadcopter simulator are shown in Section 5. Finally, some conclusions are written in Section 6.

2. Nonlinear Dynamics of the Quadcopter

2.1. Quadcopter Dynamic Description

The motion of the quadcopter is generated by the lift forces produced by the rotating propeller blades, where the translations and rotations are realized by manipulating the difference between velocities of four rotors. The quadcopter has 6 degrees of freedom (attitude and position) according to the earth-fixed frame. The position of this helicopter is described by the position vector $[x, y, z]^T$ and its attitude is defined by three Euler angles (pitch, roll and yaw) as $[\phi, \theta, \psi]^T$, where these three angles are assumed to satisfy the following constraints [15]:

$$-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}, \tag{1a}$$

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \tag{1b}$$

$$-\pi \leq \psi \leq \pi. \tag{1c}$$

Consider the earth-fixed frame as inertial frame and body-fixed frame with the origin in the mass center of the quadcopter. It is further assumed that the body of quadcopter is rigid and symmetric. The dynamics of the quadcopter can be characterized by a set of continuous-time differential equations as follows [2]:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2, \tag{2a}$$
$$\ddot{\theta} = \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega_r + b_2U_3, \tag{2b}$$
$$\ddot{\psi} = \dot{\phi}\dot{\theta}a_5 + b_3U_4, \tag{2c}$$
$$\ddot{z} = -g + (\cos\phi\cos\theta)\, U_1/m, \tag{2d}$$
$$\ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\, U_1/m, \tag{2e}$$
$$\ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\, U_1/m, \tag{2f}$$

with

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \Omega_r \end{bmatrix} = \begin{bmatrix} b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left(\Omega_2^2 - \Omega_4^2\right) \\ b\left(-\Omega_1^2 + \Omega_3^2\right) \\ d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{bmatrix} \tag{3}$$

and the parameters of the quadcopter are defined as

$$\begin{aligned} a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}},\ a_2 = \frac{J_r}{I_{xx}},\ a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}}, \\ a_4 &= \frac{J_r}{I_{yy}},\ a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}}, \\ b_1 &= \frac{\ell}{I_{xx}},\ b_2 = \frac{\ell}{I_{yy}},\ b_3 = \frac{1}{I_{zz}}. \end{aligned}$$

where Ω_1, Ω_2, Ω_3 and Ω_4 denote the speeds of four rotors, respectively, I_{xx}, I_{yy} and I_{zz} are the moments of inertial of the quadcopter around three body axes, respectively, J_r is the propeller rotational moment of inertia, ℓ denotes the arm length of the quadcopter, b and d represent thrust and drag coefficients. Besides, the constraints on the rotor angular speeds can be formulated by

$$0 \leq \Omega_i \leq \Omega_i^{max}, \quad i = 1, 2, \ldots, 4, \tag{4}$$

where Ω_i^{max} denotes the maximum angular speed of the i-th rotor.

It is noticed that the dynamic model in (2) for designing the MPC controller is a simplified one, where some modeling errors are ignored, such as the wind resistances. In the simulation, the uncertain simulator model including some modeling errors will be used.

2.2. Control-oriented Model of the Quadcopter

Considering the dynamic equations of the quadcopter in (2), the variables of system states and inputs are chosen as

$$\mathbf{x} \triangleq \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} & x & \dot{x} & y & \dot{y} \end{bmatrix}^T, \tag{5a}$$
$$\mathbf{u} \triangleq [\Omega_1\ \ \Omega_2\ \ \Omega_3\ \ \Omega_4]^T. \tag{5b}$$

Then, by means of the Euler discretization method, the discrete-time control-oriented model of the quadcopter can be generally formulated as

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k), \tag{6a}$$
$$\mathbf{y}(k) = \mathbf{G}(\mathbf{x}(k)) + \mathbf{v}(k), \tag{6b}$$

where $\mathbf{x}(k)$ and $\mathbf{u}(k)$ denote vectors of system states and inputs defined in (5) at time instant k, respectively. $\mathbf{F}(\cdot)$ denotes the nonlinear mapping function describing the system dynamics in (2) and (3). $\mathbf{G}(\cdot)$ represents the measurement function based on the characteristics of the available sensors. Besides, $\mathbf{w}(k)$ and $\mathbf{v}(k)$ denote the vectors of the system disturbances of modeling errors and the measurement noise of sensors at time instant k, respectively.

The quadcopter operates under some constraints, which can be written as follows:

$$\mathbf{x}_{\min} \leq \mathbf{x}(k) \leq \mathbf{x}_{\max}, \tag{7}$$
$$\mathbf{0} \leq \mathbf{u}(k) \leq \mathbf{u}_{\max}, \tag{8}$$

where $\mathbf{x}_{\min}$ denotes the vector of minimum values of the system states, $\mathbf{x}_{\max}$ and $\mathbf{u}_{\max}$ denote vectors of maximum values of system states and inputs.

2.3. Partial Nonlinearity Embeddings

Depending on (3), it is clear that the selected input of the quadcopter Ω_i is hidden in $[U_1, U_2, U_3, U_4, \Omega_r]^T$. Hence, the virtual inputs of the quadcopter are chosen as

$$\widetilde{\mathbf{u}} \triangleq [U_1, U_2, U_3, U_4, \Omega_r]^T . \tag{9}$$

Therefore, the nominal and uncertain control-oriented model of the quadcopter (6) can be reformulated with the virtual inputs as

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \widetilde{\mathbf{u}}(k)) + \mathbf{w}(k), \tag{10a}$$
$$\mathbf{y}(k) = \mathbf{G}(\mathbf{x}(k)) + \mathbf{v}(k). \tag{10b}$$

3. Constraint Satisfactions for the Quadcopter Control

In this section, the constraint-satisfaction approach is used to find the suitable bounds of the defined virtual inputs (9) in order to guarantee the consistency (3). The utilized constraint-satisfaction approach relies on an interval-based method, which has been popularly applied in the robotics [16]. The interval-based constraint-satisfaction approach can be implemented by a procedure of the forward and backward propagations to refine the value range from wide intervals to short ones including eliminating the impossible values violated from some defined equality and inequality constraints.

3.1. Constraint-Satisfaction Approach

As introduced in [16], a constraint-satisfaction approach on sets can be formulated as a 3-tuple $\mathcal{H} = (\mathcal{Z}, \mathcal{D}, \mathcal{C})$, where

- $\mathcal{Z} = \{z_1, z_2, \cdots, z_n\}$ is a finite set of variables.
- $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_n\}$ is the set of domains of the variables.
- $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$ is a finite set of constraints, where each constraint c_i is specified by the pair $(\mathcal{Z}_i, \mathcal{R}_i)$, in which $\mathcal{Z}_i$, that is called as the constraint scope, is a subset of $\mathcal{Z}$ and $\mathcal{R}_i$, that is called as constraint relation, is a relation indicating the allowed combination of values for the variables belonging to $\mathcal{Z}_i$.

The contractors are the useful tools for solving the constraint-satisfaction problem [16], [17]. Solving a constraint-satisfaction problem consists in finding all variable assignments such that all constraints are satisfied. The variable value assignment $(\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_n) \in \mathcal{D}$ is a solution of $\mathcal{H}$ if all constraints in $\mathcal{C}$ are satisfied. The set of all solution points of $\mathcal{H}$ is called the global solution set and denoted by $\mathcal{S}(\mathcal{H})$. The variable $z_i \in \mathcal{Z}$ is consistent in $\mathcal{H}$ if and only if $\forall \hat{z}_i \in \mathcal{D}_i$,

$$\exists(\hat{z}_1 \in \mathcal{D}_1, \cdots, \hat{z}_n \in \mathcal{D}_n)$$

such that $(\hat{z}_i, \cdots, \hat{z}_n) \in \mathcal{S}(\mathcal{H})$.

The solution of a constraint-satisfaction problem is said to be globally consistent if and only if every variable is consistent. A variable is locally consistent if and only if it is consistent with respect to all directly connected constraints. Thus, the solution of the constraint-satisfaction problem is said to be locally consistent if all variables are locally consistent. An algorithm for finding an approximation of the solution set of a constraint-satisfaction problem can be found in [16].

3.2. Constraint Satisfactions for the Quadcopter Control

As aforementioned, partial nonlinearities in (3) are embedded into the control-oriented model of the quadcopter. Therefore, $\mathbf{u}(k)$ is replaced by $\widetilde{\mathbf{u}}(k)$ in (9). The constraints on $\mathbf{u}(k)$ and Equation (3) are transformed into the new constraints on $\widetilde{\mathbf{u}}(k)$. The constraint-satisfaction approach for the quadcopter control along the MPC prediction horizon H_p is implemented by using the following algorithm.

Algorithm 1 Constraint Satisfaction applied to the Quadcopter

```
1: for k := 1 to H_p do
2:     X_k ⇐ [0, x_max]
3:     U_{k−1} ⇐ [0, u_max]
4:     Ũ_{k−1} ⇐ [0, ũ_max]
5: end for
```

6: $\mathcal{V} \Leftarrow \overbrace{\mathbf{x}(1), \mathbf{x}(2), \dots \mathbf{x}(H_p)}^{X}, \overbrace{\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(H_p-1)}^{U}, \overbrace{\widetilde{\mathbf{u}}(0), \widetilde{\mathbf{u}}(1), \dots, \widetilde{\mathbf{u}}(H_p-1)}^{\widetilde{U}}$

7: $\mathcal{D} \Leftarrow \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{H_p}, \mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_{H_p-1}, \widetilde{\mathcal{U}}_0, \widetilde{\mathcal{U}}_1, \dots, \widetilde{\mathcal{U}}_{H_p-1}$

8: $\mathcal{C} \Leftarrow$ Equation (3)

9: $\mathcal{H} \Leftarrow \mathcal{V}, \mathcal{D}, \mathcal{C}$

10: $\mathcal{S} = solve(\mathcal{H})$

11: Implement the new constraint $\mathcal{S}$ into the MPC optimization problem.

4. Nonlinear Model Predictive Control of the Quadcopter

4.1. Closed-loop NMPC Scheme

The closed-loop NMPC scheme is shown in **Figure 1**. The constraint-satisfaction approach presented in Algorithm 1 is solved off-line and the updated constraints can be found. A path planner is usually required to generate a reference trajectory in order to let the quadcopter track it, where the reference trajectory contains not only the position vector $[x, y, z]^T$ but also the attitude of the quadcopter ψ. In this paper, the NMPC controller is divided into two layers: In the upper layer, a nonlinear optimization problem is proposed in order to control the quadcopter to track the given references generated by the path planner and subsequently the virtual inputs $\widetilde{\mathbf{u}}(k)$ can be obtained. In the lower layer, the actual input can be computed by solving a set of algebraic equations. Besides, when there are no enough sensors available in the quadcopter to feedback the values of the system states, an observer for the quadcopter is also required, for instance, the zonotopic extended Kalman filter proposed in [18]. Finally, a quadcopter simulator is used to test the effectiveness of the proposed NMPC controller.

4.2. The Upper Level: Nonlinear Optimization Problem

The MPC prediction model of the quadcopter is chosen by the nominal model of (10) without considering system disturbances and measurement noise. The constraints on system states $\mathbf{x}(k)$,

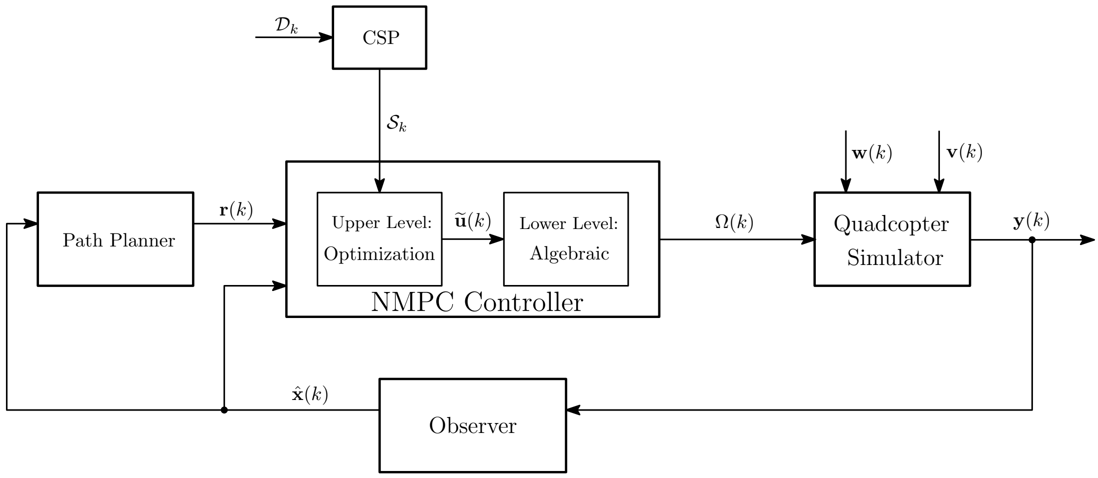


Figure 1. The closed-loop simulation scheme of the NMPC strategy with constraint satisfactions for the quadcopter

the virtual inputs $\widetilde{\mathbf{u}}(k)$ at time instant k can be denoted as

$$\mathbf{0} \leq \mathbf{x}(k+i \mid k) \leq \mathbf{x}_{\max}, \quad i = 1, 2, \ldots, H_p, \tag{11a}$$
$$\mathbf{0} \leq \widetilde{\mathbf{u}}(k+j \mid k) \leq \widetilde{\mathbf{u}}_{\max}, \quad j = 0, 1, \ldots, H_p - 1. \tag{11b}$$

The control objective of the quadcopter is mainly towards controlling the quadcopter to track the given position references and attitude references both generated by the path planner. Therefore, the cost function for the NMPC controller mainly includes two parts: a tracking part of the minimization of the deviations between x, y, z and ψ with their references x^r, y^r, z^r and ψ^r and a smoothness part with the minimization of the slew rate of the virtual inputs, which can be defined as

$$\begin{aligned} L(\mathbf{x}, \widetilde{\mathbf{u}}) \triangleq \sum_{i=1}^{H_p} \Big(& \lambda_x \ell_x \left(x_9(k+i \mid k), x^r(k+i) \right) \\ & + \lambda_y \ell_y \left(x_{11}(k+i \mid k), y^r(k+i) \right) \\ & + \lambda_z \ell_z \left(x_7(k+i \mid k), z^r(k+i) \right) \\ & + \lambda_\psi \ell_\psi \left(x_5(k+i \mid k), \psi^r(k+i) \right) \Big) \\ + \sum_{j=0}^{H_p-1} & \lambda_{\Delta u} \ell_{\Delta u} \left(\widetilde{\mathbf{u}}(k+j \mid k) \right), \end{aligned} \tag{12}$$

with

$$\ell_x \left(x_9(k+i \mid k) \right) = \left\| x_9(k+i \mid k) - x^r(k+i) \right\|_2, \tag{13a}$$
$$\ell_y \left(x_{11}(k+i \mid k) \right) = \left\| x_{11}(k+i \mid k) - y^r(k+i) \right\|_2, \tag{13b}$$
$$\ell_z \left(x_7(k+i \mid k) \right) = \left\| x_7(k+i \mid k) - z^r(k+i) \right\|_2, \tag{13c}$$
$$\ell_\psi \left(x_5(k+i \mid k) \right) = \left\| x_5(k+i \mid k) - \psi^r(k+i) \right\|_2, \tag{13d}$$

and

$$\ell_{\Delta u} \left(\widetilde{\mathbf{u}}(k+j \mid k) \right) = \left\| \Delta \widetilde{\mathbf{u}}(k+j \mid k) \right\|_2, \tag{14a}$$
$$\Delta \widetilde{\mathbf{u}}(k+j \mid k) = \widetilde{\mathbf{u}}(k+j \mid k) - \widetilde{\mathbf{u}}(k+j-1 \mid k), \tag{14b}$$

where λ_x, λ_y, λ_z, λ_ψ and $\lambda_{\Delta u}$ denote the weights for each objective and $\|\cdot\|_2$ denotes the 2-norm operator.

The selected observer, here the zonotopic extended Kalman filter in [18], can be written as

$$\hat{\mathbf{x}}(k) = \chi(\mathbf{y}(k)), \tag{15}$$

where $\hat{\mathbf{x}}(k)$ denotes the estimated full states at time instant k and $\chi(\cdot)$ represents the model-based observer function based on the model of the quadcopter in (2).

In general, the NMPC strategy in the upper layer can be implemented by solving the following nonlinear optimization problem $\mathbb{P}_q(k)$:

Problem 1 (NMPC-Quadcopter Problem)

$$\min_{\widetilde{\mathbf{u}}^*(k|k),\ldots,\widetilde{\mathbf{u}}^*(k+H_p-1|k)} L(\mathbf{x}, \widetilde{\mathbf{u}}), \tag{16a}$$

subject to

$$\begin{aligned}
&\mathbf{x}(k+i+1 \mid k) = F(\mathbf{x}(k+i \mid k), \widetilde{\mathbf{u}}(k \mid k)), && \text{(16b)}\\
&\mathbf{0} \le \mathbf{x}(k+i \mid k) \le \mathbf{x}_{\max}, && \text{(16c)}\\
&\mathbf{0} \le \widetilde{\mathbf{u}}(k+i \mid k) \le \widetilde{\mathbf{u}}_{\max}, && \text{(16d)}\\
&\mathbf{x}(k \mid k) = \hat{\mathbf{x}}(k), && \text{(16e)}\\
&\hat{\mathbf{x}}(k) = \chi(\mathbf{y}(k)). && \text{(16f)}
\end{aligned}$$

After solving the nonlinear optimization problem $\mathbb{P}_q(k)$ at time step k, a series of the virtual inputs $\widetilde{\mathbf{u}}^*(k \mid k)$, … , $\widetilde{\mathbf{u}}^*(k + H_p - 1 \mid k)$ can be obtained. By means of the receding horizon approach, the first value $\widetilde{\mathbf{u}}^*(k \mid k)$ is used as the optimal control action $\widetilde{\mathbf{u}}^*(k)$ at this time step.

4.3. The Lower Level: Algebraic Problem

As introduced in the previous subsection, the virtual input can be found by solving the nonlinear optimization problem. Then, the actual input can be computed after the virtual input obtained by means of the algebraic problem.

Depending on (3), the virtual input $\widetilde{\mathbf{u}}(k)$ and actual input $\mathbf{u}(k)$ has the obvious algebraic relationship, which can be written as follows:

$$\widetilde{\mathbf{u}}(k) = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ b & 0 & -b & 0 \\ -d & d & -d & d \end{bmatrix} \mathbf{u}^2(k). \tag{17}$$

And then the actual input at time instant k can be computed by

$$\mathbf{u}(k) = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ b & 0 & -b & 0 \\ -d & d & -d & d \end{bmatrix}^{-\frac{1}{2}} \widetilde{\mathbf{u}}^{\frac{1}{2}}(k). \tag{18}$$

The solution $\mathbf{u}(k)$ of (17) satisfies the nonlinear optimization problem proposed in the upper layer as well as (18). Therefore, the solution $\mathbf{u}(k)$ can be used for the actual input at time instant k.

5. Simulation Results

In this section, the proposed NMPC controller combined with constraint satisfactions is tested using continuous-time quadcopter simulator implemented in MATLAB. In order to evaluate the inherent robustness of the NMPC controller, the simulator model includes the uncertainties in (6), which are not explicitly considered in the prediction model in the NMPC controller.

5.1. Simulator Model

Taking into account the uncertainties discussed before, the mathematical model of the simulator can be formulated as follows:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2 + \omega_1, \tag{19a}$$
$$\ddot{\theta} = \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega_r + b_2U_3 + \omega_2, \tag{19b}$$
$$\ddot{\psi} = \dot{\phi}\dot{\theta}a_5 + b_3U_4 + \omega_3, \tag{19c}$$
$$\ddot{z} = K_a^z\dot{z} - g + (\cos\phi\cos\theta)\,U_1/m, \tag{19d}$$
$$\ddot{x} = K_a^x\dot{x} + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\,U_1/m, \tag{19e}$$
$$\ddot{y} = K_a^y\dot{y} + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\,U_1/m, \tag{19f}$$

where $\omega_1, \omega_2, \omega_3 \in \mathbb{R}$ denote the system disturbances that can be regarded as the Gaussian white noise following some certain distributions, respectively. K_a^x, K_a^y and K_a^z denote the aerodynamic friction coefficients for the axes x, y and z, respectively. All the parameters of the quadcopter are obtained from [2].

The measurement outputs of the quadcopter simulator are selected as

$$\mathbf{y} = \begin{bmatrix}\dot{x} & \dot{y} & \dot{z} & \dot{\psi}\end{bmatrix}^T. \tag{20}$$

5.2. Results

The NMPC optimization problem is solved by using the nonlinear programming technique implemented through *YALMIP* toolbox [19] and *IPOPT* solver implemented in *OPTI* toolbox [20]. The constraint-satisfaction algorithm for the quadcopter control is implemented using the *Interval Peeler* software [21]. The continuous-time quadcopter simulator is built by means of the function of *ode* in MATLAB.

In this work, the reference trajectory is assumed to be available generated by a path planner. The weights for each objective are set as $\lambda_x = 10$, $\lambda_y = 10$, $\lambda_z = 10$, $\lambda_\psi = 10$ and $\lambda_{\Delta u} = 0.1$. The sampling time is chosen as $0.1s$. The NMPC prediction horizon H_p is chosen equal to 10.

Besides, three scenarios are provided in the simulations defined as follows:

- *Deterministic scenario*:

$$\omega_1 = \omega_2 = \omega_3 = 0,$$
$$K_a^z = K_a^x = K_a^y = 0,$$

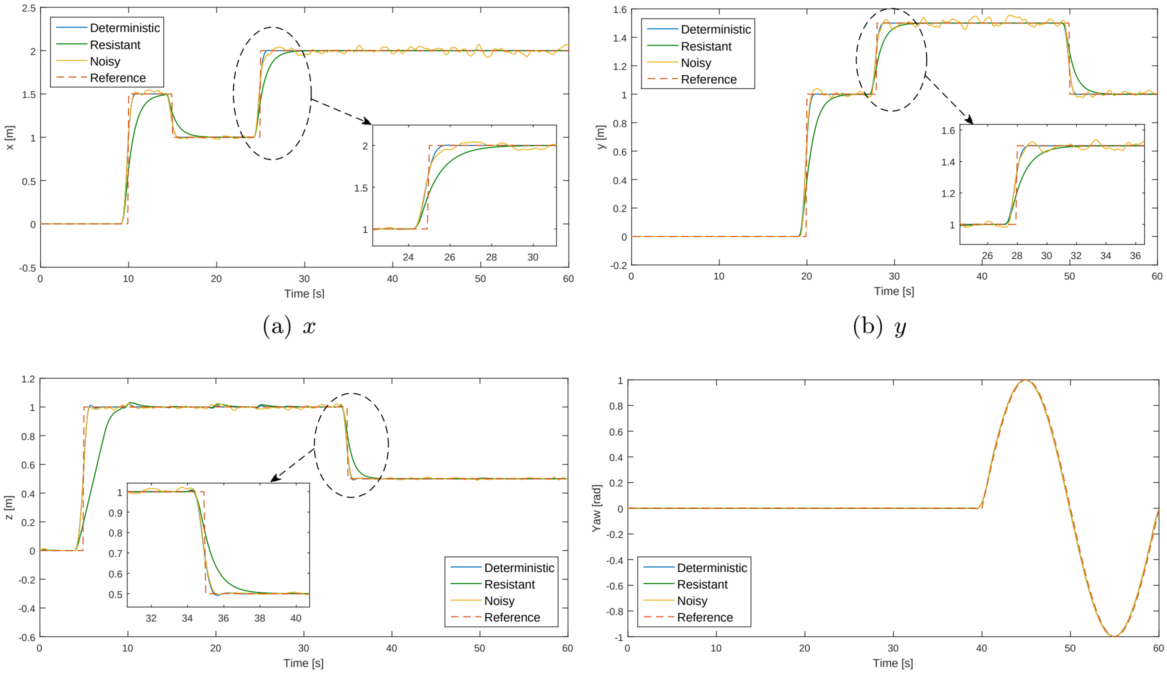
- *Resistant scenario*:

$$\omega_1 = \omega_2 = \omega_3 = 0,$$
$$K_a^z = 4.36, K_a^x = 2.91, K_a^y = 2.91,$$

- *Noisy scenario*:

$$\omega_1, \omega_2, \omega_3 \sim \mathcal{N}(0, 0.05),$$
$$K_a^z = K_a^x = K_a^y = 0,$$

  where $\mathcal{N}(0, 0.05)$ denotes the Gaussian distribution with the zero mean and the variance of 0.05.



(a) x (b) y

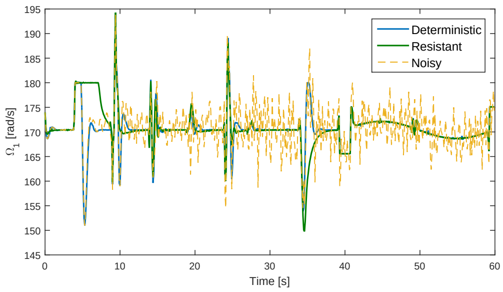
(c) z (d) ψ

Figure 2. Simulation results of x, y, z and ψ for three scenarios

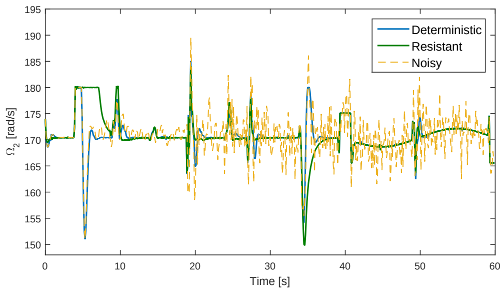
The selected simulation results with the three scenarios are shown in **Figure 2** and **Figure 3**. Four references for x, y, z and ψ are generated by the path planner in red dashed lines. It can be seen that the control objective of tracking the given references is well achieved for all the scenarios. The deterministic scenario presents the case that the simulator model is the same as the prediction model in the controller. In this case, the real trajectories on each axis and angle can follow the given references with a fast response, which means that the MPC is able to find feasible solutions with the mathematical model of the quadcopter. For the resistant scenario, the continuous resistant forces are included in order to simulate the quadcopter is operated in an environment with the effect of the wind. Consequently, as shown in the simulation results, the real trajectories of the quadcopter have some delays to reach the references compared to the deterministic scenario. In terms of the noisy scenario, additive noise following a Gaussian distribution with zero mean and variance of 0.05 is assumed, which is mainly generated from sensors located in the quadcopter. Therefore, there exist small deviations around the given the references on each axis. Besides, four actual control inputs Ω_1, Ω_2, Ω_3 and Ω_4 are plots. For the deterministic and resistant cases, these inputs are quite smooth compared to the noisy case.

The real trajectory in 3-dimensional space ($[x, y, z]^T$) is plotted in **Figure 4**. The quadcopter is controlled to do a set of motions including taking off, moving along x, y and z axis and turning the head of the quadcopter through changing ψ. As shown in **Figure 4**, the quadcopter is able to reach the end point from a given start point for the deterministic and resistant cases. For the noisy scenario, the quadcopter arrives at an area around the target place (the end point) since there are small deviations for the each axis. Hence, the tracking accuracy for the quadcopter full-state control is quite significant. As a result, the quadcopter controlled by the NMPC controller is able to track the given references.

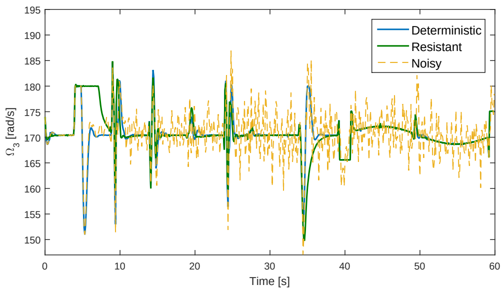
In reality, the quadcopter could suffer some unexpected disturbances at a certain time. Hence, it is necessary to test the behavior of the designed NMPC controller under these disturbances. From **Figure 5**, two peak disturbances with the same magnitude of 0.5 in $\dot{z}$ around two different



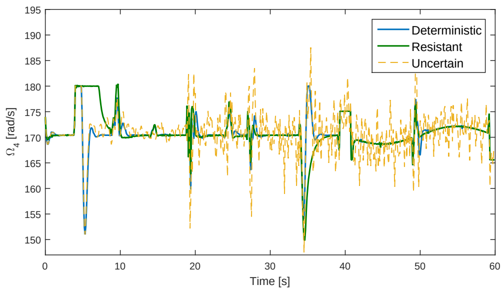
(a) Ω_1



(b) Ω_2



(c) Ω_3



(d) Ω_4

Figure 3. Simulation results of Ω_1, Ω_2, Ω_3 and Ω_4 for three scenarios

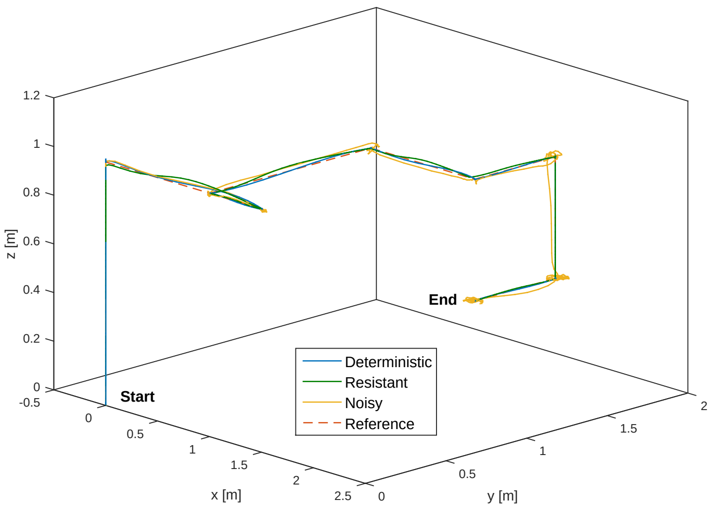
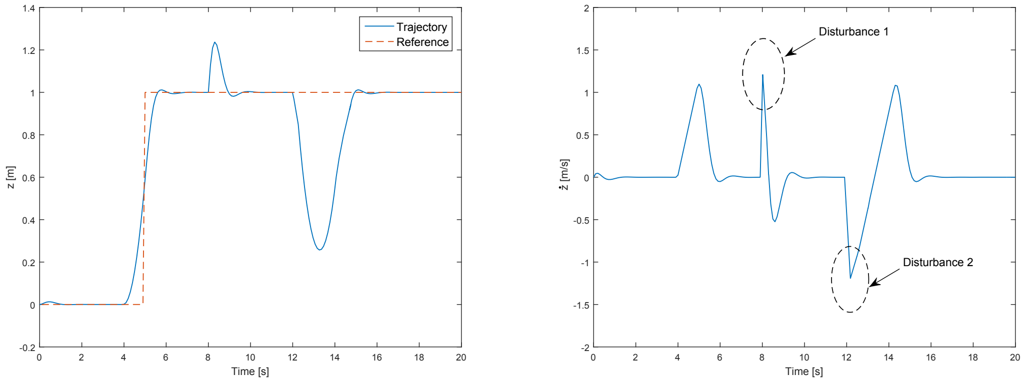


Figure 4. Real trajectory and reference for three scenarios

directions in z axis are tested. The NMPC controller is able to deal with these disturbances within some transient periods. In particular, the tracking objective can be guaranteed by using the proposed NMPC strategy as well.

6. Conclusions

In this paper, an NMPC strategy combined with constraint satisfactions is studied for a quadcopter to track the given references. From the simulation results, the NMPC controller



(a) z (b) $\dot{z}$

Figure 5. Result of adding disturbances: k=80, a disturbance towards positive z axis and k=120, a disturbance towards negative z axis

is able to provide quite accurate tracking results and the inherent robustness of the NMPC controller is able to deal with uncertainties.

On the other hand, the nonlinear optimization problem is not easy to be solved compared with the linear one requiring more time to find a solution. Taking into account the continuous hardware and software development in the computer science field, the nonlinear programming technique is expected to find the solution that satisfies the required real-time constraints allowing the real implementation. Thus, as future research, an important direction is to improve nonlinear optimization algorithm by some possible ways, such as properly relaxing the nonlinear constraints and updating the initial guess of the nonlinear solver. Besides, the MPC strategy is normally considered as a flexible control strategy and a quite good framework for fault-tolerant control because the system reconfiguration after faults appeared in the control system is easy to implement in the MPC constraint settings. Hence, as another future work, the fault-tolerant capability of the proposed control strategy will be investigated.

Acknowledgement
This work has been partially funded by the Spanish Government (MINECO) through the project CICYT ECOCIS (ref. DPI2013-48243-C2-1-R), by MINECO and FEDER through the project CICYT HARCRICS (ref. DPI2014-58104-R), by the Natural Science Foundation of Guangdong (No.2015A030313881), by the National Natural Science Foundation of China (No.61673239) and the Shenzhen Key Lab of Space Robotic Technology and Telescience.

References
[1] Alexhs K, Nikolakopoulos G and Tzes A 2011 *International Review of Automatic Control* **4**
[2] Alexis K, Nikolakopoulos G and Tzes A 2012 *IET Control Theory Applications* **6** 1812 – 1827
[3] Valavanis K and Vachtsevanos G 2015 *Handbook of Unmanned Aerial Vehicles* (Springer)
[4] Garciá R, Rubio F and Ortega M 2012 Robust PID control of the quadrotor helicopter *IFAC Conference on Advances in PID Control* (Brescia, Italy)
[5] Lippiello V, Ruggiero F and Serra D 2014 Emergency landing for a quadrotor in case of a propeller failure: A PID based approach *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (Montreal, Canada) pp 1 – 7
[6] Sadeghzadeh I, Mehta A, Chamseddine A and Zhang Y 2012 Active fault tolerant control of a quadrotor UAV based on gainscheduled PID control *25th IEEE Canadian Conference on Electrical Computer Engineering (CCECE)* (Hokkaido, Japan) pp 1 – 4

[7] Benallegue A, Mokhtari A and Fridman L 2006 Feedback linearization and high order sliding mode observer for a quadrotor UAV *International Workshop on Variable Structure Systems (VSS)* (Alghero, Italy) pp 365 – 372

[8] Qingtong W, Honglin W, Qingxian W and Mou C 2015 Backstepping-based attitude control for a quadrotor UAV using nonlinear disturbance observer *34th Chinese Control Conference (CCC)* (Hangzhou, China) pp 771 – 776

[9] Rashad R, Aboudonia A and El-Badawy A 2015 Backstepping trajectory tracking control of a quadrotor with disturbance rejection *XXV International Conference on Information, Communication and Automation Technologies (ICAT)* (Sarajevo, Bosnia and Herzegovina) pp 1 – 7

[10] Shulong Z, Honglei A, Daibing Z and Lincheng S 2014 A new feedback linearization LQR control for attitude of quadrotor *13th International Conference on Control Automation Robotics Vision (ICARCV)* (Singapore) pp 1593 – 1597

[11] Bavili R E, Khosrowjerdi M J and Vatankhah R 2015 *Control Engineering and Applied Informatics* **17** 68 – 76

[12] Deori L, Garatti S and Prandini M 2015 A model predictive control approach to aircraft motion control *American Control Conference (ACC)* (Chicago, USA) pp 2299 – 2304

[13] Drak A, Noura H, Hejase M and Younes Y A 2015 Sensor fault diagnostic and fault-tolerant control for the altitude control of a quadrotor UAV *8th IEEE GCC Conference and Exhibition (GCCCE)* pp 1 – 5

[14] López-Estrada F, Ponsart J, Theilliol D, Zhang Y and Astorga-Zaragoza C 2015 *Journal of Intelligent & Robotic Systems* 1 – 15

[15] Derafa L, Madani T and Benallegue A 2006 Dynamic modelling and experimental identification of four rotors helicopter parameters *IEEE International Conference on Industrial Technology (ICIT)* (Mumbai, India) pp 1834 – 1839

[16] Jaulin L, Kieffer M, Didrit O and Walter E 2001 *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics* (Springer)

[17] Jaulin L 2012 *Computing* **94** 297 – 311

[18] Wang Y and Puig V 2016 Zonotopic extended Kalman filter and fault detection of discrete-time nonlinear systems applied to a quadrotor helicopter *3rd International Conference on Control and Fault-Tolerant Systems (SysTol)* (Barcelona, Spain) pp 361 – 366

[19] Löfberg J 2004 YALMIP: A toolbox for modeling and optimization in MATLAB URL `http://users.isy.liu.se/johanl/yalmip`

[20] Currie J and Wilson D 2012 *OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User* URL `http://www.i2c2.aut.ac.nz/Wiki/OPTI`

[21] Baguenard X 2005 *Interval Peeler Solver* URL `https://www.ensta-bretagne.fr/jaulin/demo.html`