

Article

A qLPV-MPC Control Strategy for Trajectory Tracking of Quadrotors

Daniel Rodriguez-Guevara ¹, Antonio Favela-Contreras ^{1,*} and Oscar Julian Gonzalez-Villarreal ²

¹ School of Engineering and Sciences, Tecnologico de Monterrey, Ave. Eugenio Garza Sada 2501, Monterrey 64849, Mexico; a01280937@tec.mx

² Centre for Autonomous and Cyber-Physical Systems, Cranfield University, College Road, Cranfield MK43 0AL, UK; oscar.gonzalez-villarreal@cranfield.ac.uk

* Correspondence: antonio.favela@tec.mx

Abstract: This article proposes a model predictive control (MPC) strategy for a quadrotor drone trajectory tracking based on a compact state-space model based on a quasi-linear parameter varying (qLPV) representation of the nonlinear quadrotor. The use of a qLPV representation allows for faster execution times, which can be suitable for real-time applications and for solving the optimization problem using quadratic programming (QP). The estimation of future values of the scheduling parameters along the prediction horizon is made by using the planned trajectory based on the previous optimal control actions. The performance of the proposed approach is tested by following different trajectories in simulation to show the effectiveness of the proposed control scheme.

Keywords: nonlinear control; model predictive control; linear parameter varying; unmanned aerial vehicles; optimal control



Citation: Rodriguez-Guevara, D.; Favela-Contreras, A.; Gonzalez Villarreal, O.J. A qLPV-MPC Control Strategy for Trajectory Tracking of Quadrotors. *Machines* **2023**, *11*, 755. <https://doi.org/10.3390/machines11070755>

Academic Editor: Kwanho You

Received: 16 June 2023

Revised: 13 July 2023

Accepted: 17 July 2023

Published: 19 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Research on unmanned aerial vehicles (UAV) has been increasing in the last two decades. UAVs have many applications such as communications, surveillance, assistance in natural disasters, mapping, delivery of goods, etc. The main reason for the popularity of UAVs in these applications is the size and speed of these vehicles, their range of mobility in harsh conditions, and the advantage of being an unmanned vehicle.

The autonomy of UAVs is crucial for a wide variety of applications without human involvement such as trajectory tracking, disturbance rejection, and collaboration with other UAVs. Therefore, the design of a controller is crucial for the use of UAVs in most applications.

One of the most popular types of UAVs is the quadrotor, a vehicle driven by four propellers. The quadrotor is a 12-state highly nonlinear system controlled only by four control inputs and the speed of each propeller. The translation of the quadrotors is possible by controlling the lift and the pitch and roll angles of the UAV to manipulate the tilt of the drone to manage the direction of translation. They are also characterized for their fast dynamics, being able to change directions quickly and to follow complex paths in the presence of external disturbances caused by the environment.

Trajectory planning and tracking is one of the most important tasks for the autonomy of quadrotors. Several studies have focused on this problem and the design of a controller that can track a desired trajectory efficiently. Some controllers that have been explored for this type of control problem are PID [1–4], LQG [5–7], H₂ and H_∞ [8–10], and fuzzy logic controllers [11–14]. All of these controllers achieve acceptable performance on trajectory tracking. However, most of these controllers rely on advanced tuning techniques rather than an optimal controller design considering the quadrotor's nonlinear dynamics.

Another control strategy widely used and researched for trajectory tracking of quadrotors is model predictive control (MPC). In this strategy, the trajectory of the quadrotor is

predicted along a prediction horizon based on a model as a function of the control actions. Afterward, an optimization problem is conducted online at every sampling time to obtain the optimal set of control actions based on a desired performance expressed as a cost function. One of the main advantages of MPC over other control strategies is the possibility to include constraints in the optimization problem.

In [15], a linear MPC controller for trajectory tracking control of a quadrotor is presented. In this approach, a linear model is used to represent a nonlinear quadrotor by assuming small-angle movements. This is ensured by constraints in the states of the systems and the control inputs. The results showed proper tracking of the altitude of the drone, while the trajectory control in the X and Y coordinates is not explored due to the limitations of the linear representation. Other linear MPC strategies for trajectory tracking are presented in [16–19].

To overcome the limitations of using a linear representation to design a controller for a highly nonlinear system such as a quadrotor, the nonlinear MPC (NMPC) control strategy has been used widely. In this approach, nonlinear dynamics are considered to predict the behavior of the drone along the prediction horizon. In [20], a nonlinear MPC is presented for trajectory tracking of a quadrotor. In this framework, the computational load is a limitation due to the complexity of the cost function involving nonlinear dynamics. To overcome this limitation, a numerical method solution such as the generalized minimal residual method is used. The use of numerical methods to achieve fast execution times results also in the need to add constraints for the stability of the system, which are included as contraction constraints. The results showed appropriate performance, but the stability conditions affect the convergence time. Other studies regarding NMPC strategies in UAVs are presented in [21–25].

Linear parameter varying (LPV) models have been used to deal with nonlinear models as linear systems or quasi-linear systems. In this type of model, the nonlinearities of the system are embedded in scheduling parameters, which are updated at each sampling time depending on the conditions of the system or external factors such as time and disturbances. LPV representations are often used to design controllers without including the complex nonlinear dynamics of most systems.

In quadrotors, LPV models have been used to design linear MPC controllers, which are dependent on the scheduling parameters. In [26], an MPC controller based on a quasi-LPV (qLPV) representation of a quadrotor is presented. In this work, a cascade control strategy is chosen and the quadrotor model is divided into a translational model and an attitude model. The translational model is defined as a linear model, while the attitude model is an LPV model. Afterward, the translational model was controlled using an LQR controller, while the LPV attitude was governed using an LPV-MPC controller. The results showed proper trajectory tracking for a spiral trajectory.

Several studies with MPC strategies for the control of UAVs are presented in [27–30]. In all of previous studies, the model of the quadrotor was divided into two models, due to the highly nonlinear dynamics of the quadrotor system. The main limitation of this kind of approach is that the scheduling parameters are unknown along the prediction horizon and an estimation of the parameters needs to be performed. If the parameters are not bounded, the estimation may result in improper estimation and non-optimal performance.

In this study, a novel qLPV-MPC control strategy for trajectory tracking of quadrotors is presented. The highly nonlinear model dynamics of the quadrotor are presented by a 12-state LPV system with six scheduling parameters depending on the states of the system and the angular velocity of the propellers. Afterward, an LPV-MPC controller is defined as a regular linear MPC depending on the scheduling parameters. The parameter estimation along the prediction horizon is considered via estimation of the states and control actions derived from the previous optimal set of control actions.

This work is structured as follows: Section 2 presents the quadrotor dynamics, including translation and attitude dynamics defined from the Earth frame and the body frame. Section 3 describes the state-space LPV model for the design of the MPC controller. In

Section 4, the Model Predictive Control for trajectory tracking of the controller is presented. Section 5 shows the results of the proposed control strategy for two different trajectories. Finally, Section 6 goes over future work and conclusions.

2. Quadrotor Dynamics

The quadrotor is composed of two main features: the body frame and the propellers. The function of the propellers is to give thrust to the UAV and to steer the quadrotor. The dynamics of the quadrotor are defined by the translation of the body in the XYZ space and the attitude of the body, which is composed of the rotational angles: roll, pitch, and yaw. Figure 1 presents a schematic of the quadrotor.

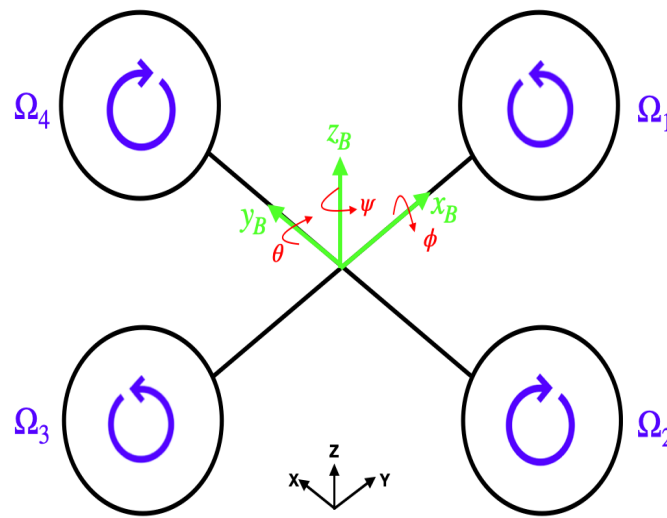


Figure 1. Quadrotor Schematic.

As seen in Figure 1, the propellers rotate in cross directions; if all the propellers rotate at the same speed and there is no external disturbance, the quadrotor will stay in a hovering state. The roll ϕ , pitch θ , and yaw ψ represent the rotational angles along the x_B , y_B and, z_B -axes of the quadrotor respectively. $[\Omega_1, \Omega_2, \Omega_3, \Omega_4]$ represent the velocity of each propeller. Note that the x_B , y_B and, z_B -axes of the quadrotor are different from the XYZ axes defined by the Earth frame shown in black in Figure 1. To define the dynamics of the quadrotor, the equations will be divided into the Earth frame (E-frame) equations and the body frame (B-frame) equations, as shown in [30]. Afterward, both frames will be combined in a compound frame (C-frame).

2.1. E-Frame Dynamics

The E-frame dynamics are related to the movement of the center of gravity (COG) of the quadrotor with respect to an arbitrary origin set on a point of the Earth. These movements are the translation of the drone and are defined by the position, velocity, and acceleration of the quadrotor and the roll, pitch, and yaw movements. The set of equations used to describe the translation of the quadrotor is shown in Equation (1):

$$\begin{cases} \dot{x} = \cos \theta \cos \psi u + (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi)v + (\cos \phi \cos \theta \cos \psi + \sin \phi \sin \psi)w \\ \dot{y} = \cos \theta \sin \psi u + (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi)v + (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)w \\ \dot{z} = -\sin \theta u + \sin \phi \sin \theta v + \cos \phi \cos \theta w \\ \dot{\phi} = p + \sin \phi \tan \theta q + \cos \phi \tan \theta r \\ \dot{\theta} = \cos \phi q - \sin \phi r \\ \dot{\psi} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r \end{cases} \quad (1)$$

in which $[u, v, w]$ is the velocity vector, while $[p, q, r]$ is the angular velocity vector with respect to the B-frame. As seen in Equation (1), this representation is not enough to model the whole quadrotor because the previous equation is dependent on parameters defined in the body frame.

2.2. B-Frame Dynamics

In the B-frame, the origin is located at the COG of the quadrotor and moves with it. This type of representation is useful to see the effects that the rotation of each propeller generates at the quadrotor. Equation (2) shows the set of equations needed to represent these dynamics, including the gravitational term, the gyroscopic effect, and the input effect on these dynamics.

$$\begin{cases} \dot{u} = vr - wq + \sin \theta g \\ \dot{v} = wp - ur - \sin \phi \cos \theta g \\ \dot{w} = uq - vp - \cos \phi \cos \theta g + \frac{U_1}{m} \\ \dot{p} = \frac{1}{J_{xx}} [-(J_{zz} - J_{yy})qr - J_{pq}\Omega_p + U_2] \\ \dot{q} = \frac{1}{J_{yy}} [(J_{zz} - J_{xx})pr - J_{pq}\Omega_p + U_3] \\ \dot{r} = \frac{1}{J_{zz}} [-(J_{yy} - J_{xx})pq + U_4] \end{cases} \quad (2)$$

in which g is the gravitational constant; m is the mass of the quadrotor; $[J_{xx}, J_{yy}, J_{zz}]$ are the inertia tensor; and $[U_1, U_2, U_3, U_4]$ are the auxiliary control inputs being functions of the speed of each rotor and defined by Equation (3)

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl(\Omega_4^2 - \Omega_2^2) \\ U_3 = bl(\Omega_3^2 - \Omega_1^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases} \quad (3)$$

where b represents the thrust factor, l is the distance from the center of the quadrotor to the propeller, and d is the drag factor. The overall velocity of the propellers is defined as $\Omega_p = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$.

Similar to the E-frame, the B-frame is not enough to understand all the dynamics of the quadrotor due to the dependence on parameters represented in the E-frame such as the roll, pitch, and yaw values. In order to build a model that includes all the dynamics of both the E-frame and the B-frame, a compound frame (C-frame) needs to be developed.

2.3. C-Frame Dynamics

In order to define a compound frame, a relation between the B-frame and the E-frame needs to be obtained. The relation of these two frames is represented by Equation (4).

$$\dot{\zeta} = J_{\Theta} v \quad (4)$$

in which $\zeta = [x, y, z, \phi, \theta, \psi]^T$ is the E-frame vector; $v = [u, v, w, p, q, r]^T$ is the B-frame vector; and J_{Θ} is a transformation matrix defined as:

$$J_{\Theta} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & 0 & 0 & 0 \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & 0 & 0 & 0 \\ -\sin \theta & \sin \phi \sin \theta & \cos \phi \cos \theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & 0 & 0 & 0 & \cos \phi & -\sin \phi \\ 0 & 0 & 0 & 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}$$

With this transformation matrix and Equations (1)–(4), the equations defining the dynamics of the quadrotor can be derived. These equations are shown in Equation (5).

$$\begin{cases} \ddot{x} = \frac{1}{m} [\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)] U_1 \\ \ddot{y} = \frac{1}{m} [\cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi)] U_1 \\ \ddot{z} = \frac{1}{m} [\cos(\phi) \cos(\theta)] U_1 - g \\ \dot{p} = \frac{1}{J_{xx}} [-(J_{zz} - J_{yy})qr - J_p q \Omega_p + U_2] \\ \dot{q} = \frac{1}{J_{yy}} [(J_{zz} - J_{xx})pr - J_p p \Omega_p + U_3] \\ \dot{r} = \frac{1}{J_{zz}} [-(J_{yy} - J_{xx})pq + U_4] \end{cases} \quad (5)$$

3. LPV State Space Model of the Quadrotor

In order to build the MPC cost function, an LPV-state-space model with scheduling variable-dependent matrices \mathbf{A} and \mathbf{B} is defined. Using Equations (1)–(5), a nonlinear state-space model to track all the variables for the movement of the quadrotor can be derived. The nonlinear state-space model for the quadrotor is shown in Equation (6).

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \frac{1}{m} [\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)] U_1 \\ \dot{v}_y = \frac{1}{m} [\cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi)] U_1 \\ \dot{v}_z = \frac{1}{m} [\cos(\phi) \cos(\theta)] U_1 - g \\ \dot{\phi} = p + \sin(\phi) \tan(\theta) q + \cos(\phi) \tan(\theta) r \\ \dot{\theta} = \cos(\phi) q - \sin(\phi) r \\ \dot{\psi} = \frac{\sin(\phi)}{\cos(\theta)} q + \frac{\cos(\phi)}{\cos(\theta)} r \\ \dot{p} = \frac{1}{J_{xx}} [-(J_{zz} - J_{yy})qr - J_p q \Omega_p + U_2] \\ \dot{q} = \frac{1}{J_{yy}} [(J_{zz} - J_{xx})pr - J_p p \Omega_p + U_3] \\ \dot{r} = \frac{1}{J_{zz}} [-(J_{yy} - J_{xx})pq + U_4] \end{cases} \quad (6)$$

As seen in the previous state-space system, the quadrotor is highly nonlinear; therefore, in order to treat the system as a linear state-space system, an LPV representation with six different scheduling variables depending directly on the states of the system is derived. The scheduling variables are presented in Equation (7).

$$\begin{cases} \rho_1 = \phi \\ \rho_2 = \theta \\ \rho_3 = \psi \\ \rho_4 = p \\ \rho_5 = q \\ \rho_6 = \Omega_p \end{cases} \quad (7)$$

The scheduling variables were chosen to represent the nonlinear state-space system of the quadrotor in an LPV state-space model using the least amount of scheduling variables needed to model the nonlinear dynamics of the quadrotor. With Equations (6) and (7), a quasi-LPV state-space system can be used to compute the MPC cost function in a linear form. The qLPV-SS with matrices \mathbf{A} and \mathbf{B} being scheduling parameter-dependent is shown in Equation (8).

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6) \mathbf{x}(t) + \mathbf{B}(\rho_1, \rho_2, \rho_3) \begin{bmatrix} \mathbf{u}(t) \\ g \end{bmatrix} \quad (8)$$

where $\mathbf{x}(t) = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r]^T$, $\mathbf{u}(t) = [U_1, U_2, U_3, U_4]^T$, and matrices \mathbf{A} and \mathbf{B} are defined as follows:

$$\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \epsilon\rho_2 & \epsilon\rho_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \epsilon\rho_1 & 0 & \epsilon\rho_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \epsilon\rho_1 & \epsilon\rho_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \sin(\rho_1) \tan(\rho_2) & \cos(\rho_1) \tan(\rho_2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cos(\rho_1) & -\sin(\rho_1) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sin(\rho_1)}{\cos(\rho_2)} & \frac{\cos(\rho_1)}{\cos(\rho_2)} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{J_p \rho_6}{J_{xx}} & -\frac{(J_{zz} - J_{yy})\rho_5}{J_{xx}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{J_p \rho_6}{J_{yy}} & 0 & \frac{(J_{zz} - J_{xx})\rho_4}{J_{yy}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-(J_{yy} - J_{xx})\rho_4}{J_{zz}} & 0
\end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{m} [\cos(\rho_1) \sin(\rho_2) \cos(\rho_3) + \sin(\rho_1) \sin(\rho_3)] & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{m} [\cos(\rho_1) \sin(\rho_2) \sin(\rho_3) - \sin(\rho_1) \cos(\rho_3)] & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{m} [\cos(\rho_1) \cos(\rho_2)] & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{J_{xx}} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{J_{yy}} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{J_{zz}} & 0 & 0
\end{bmatrix}$$

The value of ϵ is defined as a very low constant (e.g., $\epsilon = 1 \times 10^{-3}$) and is included for the system to be controllable and to avoid having three zero rows in the \mathbf{A} matrix. The parameter ϵ is therefore added for the MPC to identify that, by changing the attitude angles, the acceleration on the XYZ plane will be affected, but the predicted trajectory will not be significantly affected due to the small value of ϵ . The inclusion of the six different scheduling variables on the qLPV state-space allows for expressing the highly nonlinear quadrotor system into a compact quasi-linear state-space system, which is updated at each sampling time depending on the values of the states and the speed of the propellers. This compact qLPV-SS representation simplifies the computation of the future states of the system along the prediction horizon and improves the construction of the MPC cost function without splitting the system into several sub-systems. The gravitational constant is treated as an uncontrollable input or disturbance, which is constant at every sampling instant. In the next section, the proposed qLPV-SS presented in (8) will be used to compute the MPC cost function and to solve the optimization problem to obtain the optimal set of control actions.

4. Model Predictive Control Based on the LPV Representation of the Quadrotor

To build an MPC based on an LPV model, the prediction of the future states along the prediction horizon N_p is formulated considering not only the state matrices \mathbf{A} and \mathbf{B} but also the variations in the scheduling variables along the prediction horizon. Therefore, matrices \mathbf{A} and \mathbf{B} are not considered constant along the prediction horizon, and the values of these matrices need to be considered for each step of the horizon in order to consider the nonlinear behavior of the quadrotor. Thus, Equation (9) presents the i -steps ahead calculation of the states of the quadrotor.

$$\mathbf{x}(k+i|k) = \prod_{j=0}^{i-1} \mathbf{A}_d(\rho_i(k+j)) \mathbf{x}(k) + \left(\sum_{s=1}^{i-1} \left(\prod_{l=s}^{i-1} \mathbf{A}_d(\rho_i(k+l)) \right) \mathbf{B}_d(\rho_i(k+s-1)) \begin{bmatrix} \mathbf{u}(k+s-1) \\ g \end{bmatrix} \right) + \mathbf{B}_d(\rho_i(k+i-1)) \begin{bmatrix} \mathbf{u}(k+i-1) \\ g \end{bmatrix} \quad (9)$$

where $\mathbf{A}_d(\rho_i)$ and $\mathbf{B}_d(\rho_i)$ are the discrete matrices obtained from the continuous matrices \mathbf{A} and \mathbf{B} presented in Section 3 using sampling time T_s and zero-order-hold (ZOH) discretization. These matrices are updated at every sampling instant k as the set of scheduling parameters (ρ_i) are also updated at each iteration. The prediction of the future states is performed along the prediction horizon, from k to $k + N_p$; thus, Equation (9) can be expressed as a compact matrix form to calculate the values of the states in the prediction horizon, as shown in Equation (10).

$$\mathbf{X} = \Phi \cdot \mathbf{x}(k) + \Psi \cdot \mathbf{U} \quad (10)$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(k+1|k) \\ \mathbf{x}(k+2|k) \\ \vdots \\ \mathbf{x}(k+N_p|k) \end{bmatrix} \quad (11)$$

$$\Phi(k) = \begin{bmatrix} \mathbf{A}_d(\rho_i(k)) \\ \prod_{j=0}^1 (\mathbf{A}_d(\rho_i(k+j))) \\ \vdots \\ \prod_{j=0}^{N_p-1} (\mathbf{A}_d(\rho_i(k+j))) \end{bmatrix} \quad (12)$$

$$\Psi(k) = \begin{bmatrix} \mathbf{B}_d(\rho_i(k)) & 0_{n_x \cdot n_u} & \dots & 0_{n_x \cdot n_u} \\ \mathbf{A}_d(\rho_i(k+1)) \mathbf{B}_d(\rho_i(k)) & \mathbf{B}_d(\rho_i(k+1)) & \dots & 0_{n_x \cdot n_u} \\ \mathbf{A}_d(\rho_i(k+2)) \mathbf{A}_d(\rho_i(k+1)) \mathbf{B}_d(\rho_i(k)) & \mathbf{A}_d(\rho_i(k+2)) \mathbf{B}_d(\rho_i(k+1)) & \dots & 0_{n_x \cdot n_u} \\ \vdots & \vdots & \ddots & \vdots \\ (\prod_{i=1}^{N_p-1} \mathbf{A}_d(\rho_i(k+1))) \mathbf{B}_d(\rho_i(k)) & (\prod_{i=1}^{N_p-1} \mathbf{A}_d(\rho_i(k+2))) \mathbf{B}_d(\rho_i(k+1)) & \dots & \mathbf{B}_d(\rho_i(k+N_p-1)) \end{bmatrix} \quad (13)$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}(k) \\ g \\ \mathbf{u}(k+1) \\ g \\ \vdots \\ \mathbf{u}(k+N_p-1) \\ g \end{bmatrix} \quad (14)$$

where $\mathbf{X} \in \mathbb{R}^{N_p \cdot n_x}$, $\Phi \in \mathbb{R}^{N_p \times n_x \cdot n_x}$, $\Psi \in \mathbb{R}^{N_p \cdot n_x \times N_p \cdot n_u}$ and $\mathbf{U} \in \mathbb{R}^{N_p \cdot (n_u+1)}$, where n_x represents the number of states and n_u represents the number of controllable inputs of the quadrotor. With Equation (10), the computation of the future values of the states of the quadrotor is performed, and a cost function for trajectory tracking can be designed to make the quadrotor follow a specified trajectory and to minimize the effort of the propellers. The proposed cost function is the following:

$$J = (\mathbf{X}(k) - \mathbf{X}_r(k))^T Q (\mathbf{X}(k) - \mathbf{X}_r(k)) + (\mathbf{U}(k) - \mathbf{U}(k-1))^T R (\mathbf{U}(k) - \mathbf{U}(k-1)) \quad (15)$$

where $Q \in \mathbb{R}^{n_x \cdot N_p \times n_x \cdot N_p}$ and $R \in \mathbb{R}^{n_u \cdot N_p \times n_u \cdot N_p}$ are weighting matrices, \mathbf{X}_r is the trajectory reference vector along the prediction horizon, and $\mathbf{U}(k-1)$ is the previous set of optimal control actions. For this application, the trajectory reference vector only includes the desired trajectory in the XYZ coordinates and the yaw angle reference. The other states are considered free states and are only bonded by the constraint limits. The values of the

scheduling variables along the prediction horizon are calculated by using Equation (10) from the previous iteration. Having the future values of the states along the prediction horizon allows us to represent the nonlinear behavior of the quadrotor correctly in the MPC algorithm. The objective of the MPC strategy is to minimize the cost function (15) at every sampling instant k subject to the constraints shown in (16) and (17).

$$\mathbf{U}_{min} \leq \mathbf{U} \leq \mathbf{U}_{max} \quad (16)$$

$$\mathbf{X}_{min} \leq \mathbf{X} \leq \mathbf{X}_{max} \quad (17)$$

where $\mathbf{U}_{min}, \mathbf{U}_{max} \in \mathbb{R}^{(n_u+1) \cdot N_p}$ and $\mathbf{X}_{min}, \mathbf{X}_{max} \in \mathbb{R}^{n_x \cdot N_p}$ are the defined limits for the controlled inputs and the states, respectively. The gravitational term g included in the control vector \mathbf{U} is constrained to be equal to 9.81 at every iteration. Thus, the optimization problem to solve at each sampling instant k is presented in Equation (18).

$$\min_{\mathbf{U}} J \text{ s.t. (16) and (17)} \quad (18)$$

Solving the optimization problem (18) results in the optimal set of control actions \mathbf{U} along the prediction horizons; therefore, a conversion from the control actions to the actual speeds of the propellers that are input to the system needs to be performed. Using Equation (3), the values of the speed of each propeller can be calculated as shown in Equation (19).

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (19)$$

To obtain the optimal velocities of each propellers $[\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$, the absolute value of the square root of each square velocity is computed. Figure 2 presents the qLPV-MPC control strategy for the quadrotor using a compact 12-state qLPV system.

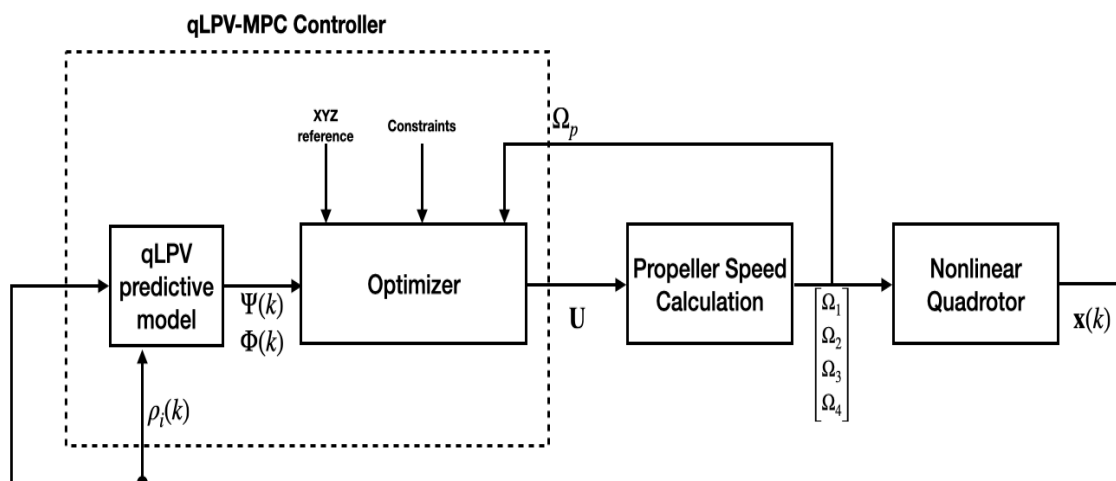


Figure 2. Block diagram of the proposed qLPV-MPC control strategy for a nonlinear quadrotor.

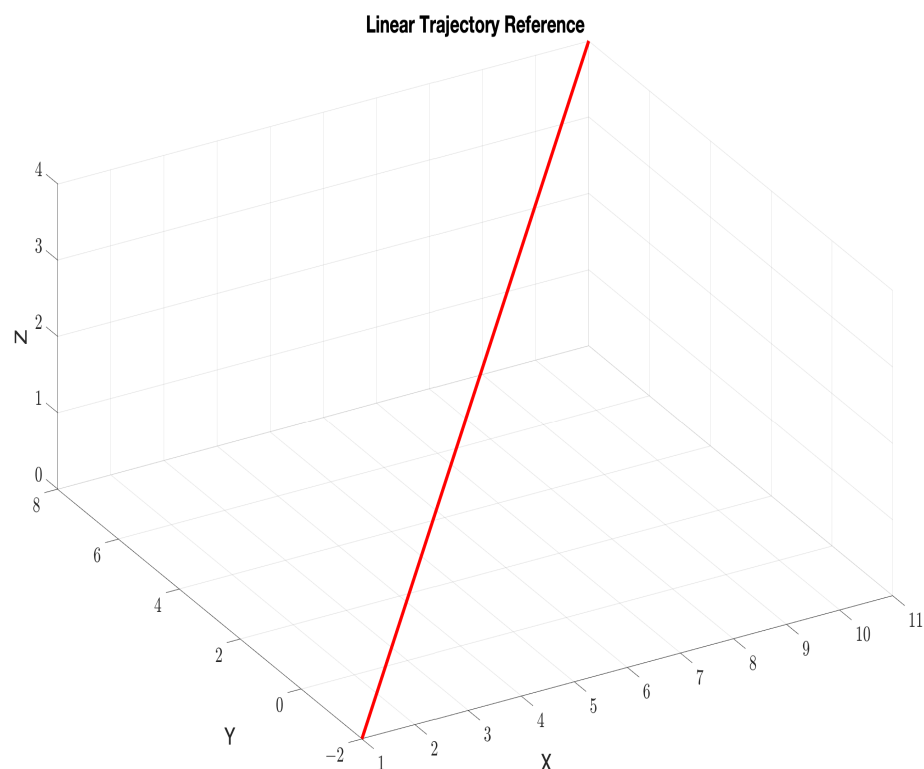
5. Results and Discussion

In order to test the performance of the proposed qLPV-MPC control strategy, a simulation of a quadrotor following two different trajectories will be performed. In the first test, a linear trajectory will be tested, while in the second test, the quadrotor will follow a spiral trajectory. In both tests, the quadrotor will start at a different starting point and will need to track and follow the trajectory. Table 1 shows the specifications of the quadrotor as presented in [26].

Table 1. Constant Values of the quadrotor.

Variable	Value	Units
g	9.8	m/s^2
m	0.698	kg
J_{xx}	0.0034	$\text{kg} \cdot \text{m}^2$
J_{yy}	0.0034	$\text{kg} \cdot \text{m}^2$
J_{zz}	0.006	$\text{kg} \cdot \text{m}^2$
J_p	1.302×10^{-6}	$\text{kg} \cdot \text{m}^2$
b	7.6184×10^{-9}	$\text{N} \cdot \text{s}^2$
d	2.6839×10^{-9}	$\text{N} \cdot \text{s}^2$
l	0.171	m

The linear trajectory and the spiral trajectory are shown in Figures 3 and 4, respectively. The simulations were performed in the Matlab-Simulink (Matlab R2019b) software environment running on a Macbook Air with a dual-core Intel Core i5 1.8 GHz processor and 8 GB of RAM with 1600 MHz DDR3. The sampling time was set to $T_s = 10$ ms to be able to measure and detect the fast dynamics of the quadrotor in any flying situation. The weighting matrix \mathbf{Q} was designed in order to penalize the states more regarding the XYZ position and velocity at the first prediction steps. Weighting matrix \mathbf{R} is designed as an identity matrix in order to penalize all the control actions equally. The optimization was performed using the Matlab function *fmincon* with the solver of sequential quadratic programming (SQP) to manage the cost function as a Quadratic Programming (QP) problem.

**Figure 3.** Linear trajectory set as the reference for the first test.

A prediction horizon of $N_p = 5$ was defined after several simulations testing the performance for different prediction horizons. Increasing the prediction horizon will not improve the performance of the proposed control strategy significantly but will increase the optimization times, making real-time implementation impossible. Figure 5 presents the results of the trajectory tracking for the proposed qLPV-MPC for the linear reference trajectory. Figures 6–8 show the values of the XYZ reference coordinates individually and the tracking carried out by the quadrotor with the qLPV-MPC strategy.

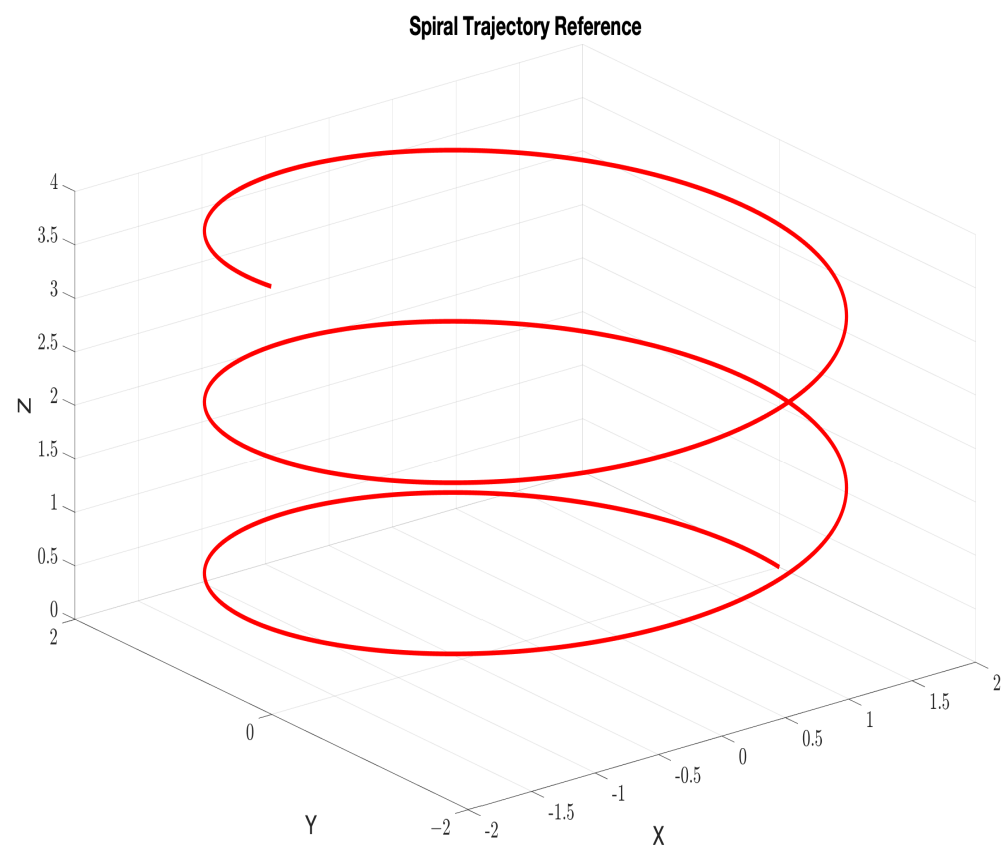


Figure 4. Spiral trajectory set as the reference for the second test.

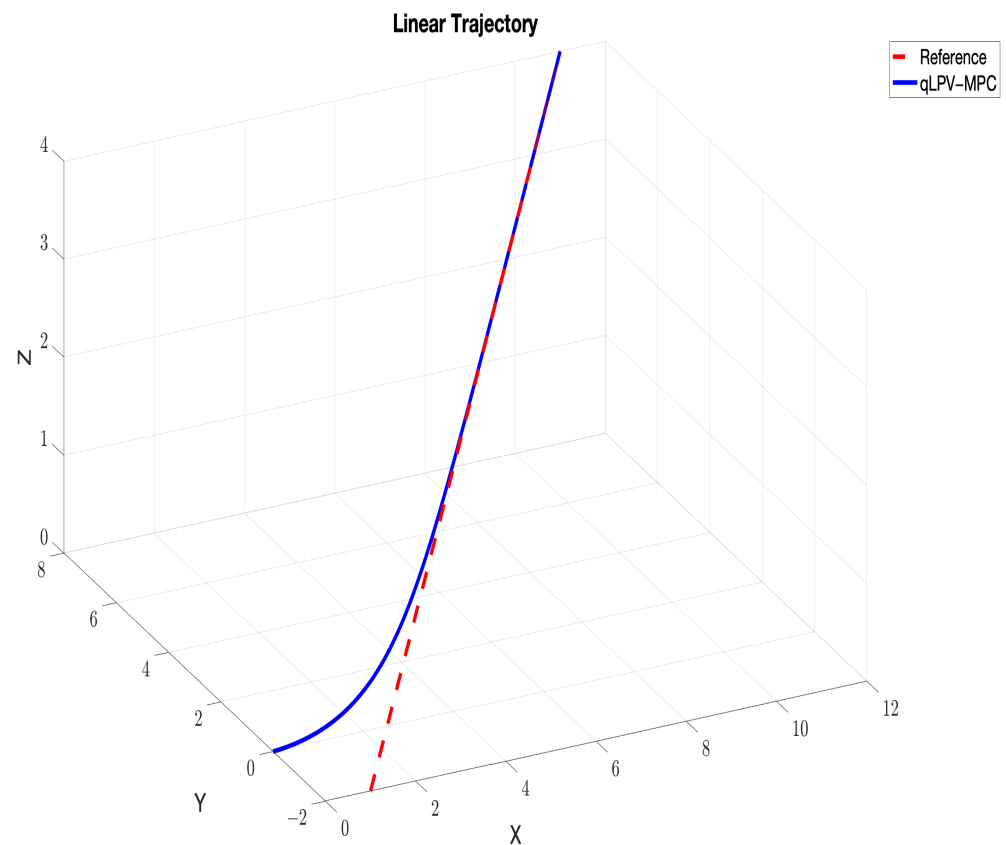


Figure 5. Tracking of the linear trajectory using the qLPV-MPC.

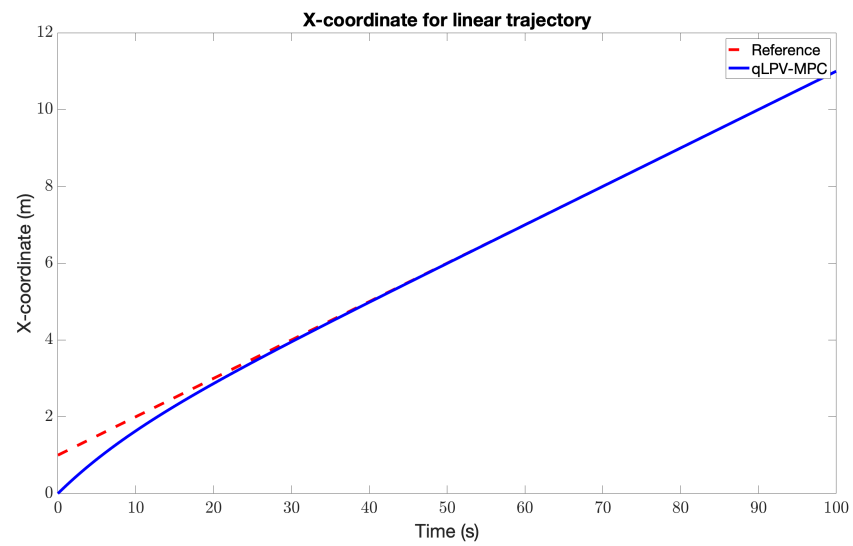


Figure 6. Tracking of the x-coordinate using the qLPV-MPC for linear reference.

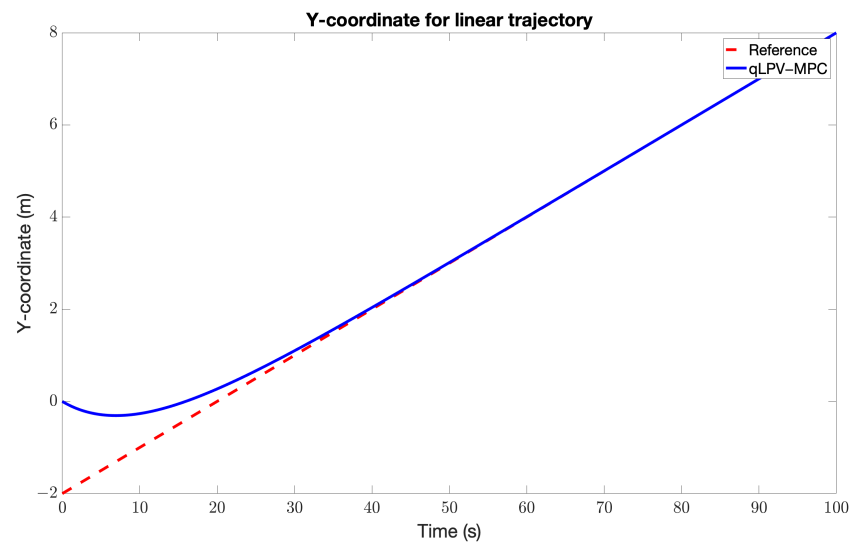


Figure 7. Tracking of the y-coordinate using the qLPV-MPC for linear reference.

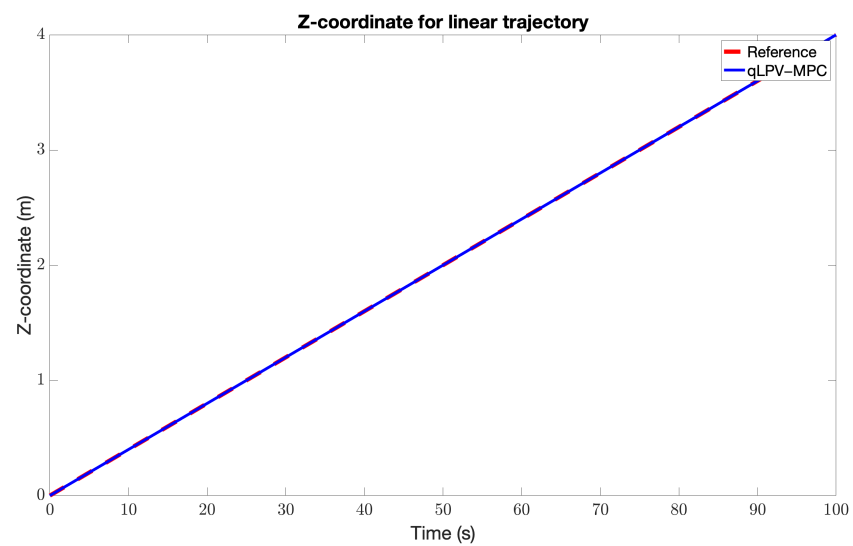


Figure 8. Tracking of the z-coordinate using the qLPV-MPC for linear reference.

Figure 9 shows the behavior of the linear velocity in each coordinate and the angular position of the quadrotor during the tracking of the linear trajectory, while Figures 10 and 11 show the control actions chosen and the required speed for each propeller.

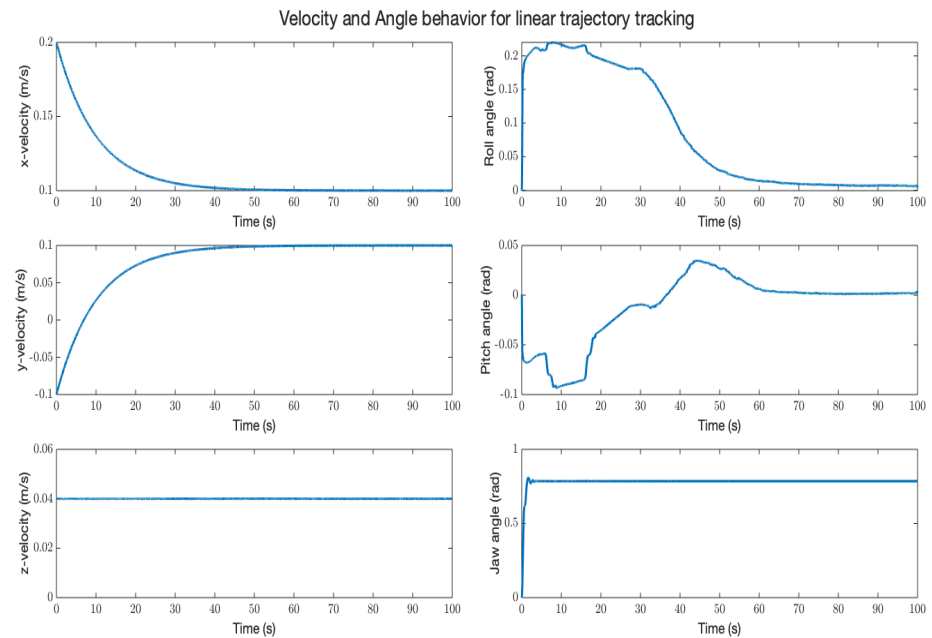


Figure 9. Behavior of the linear velocity and angular position of the quadrotor for tracking the linear trajectory.

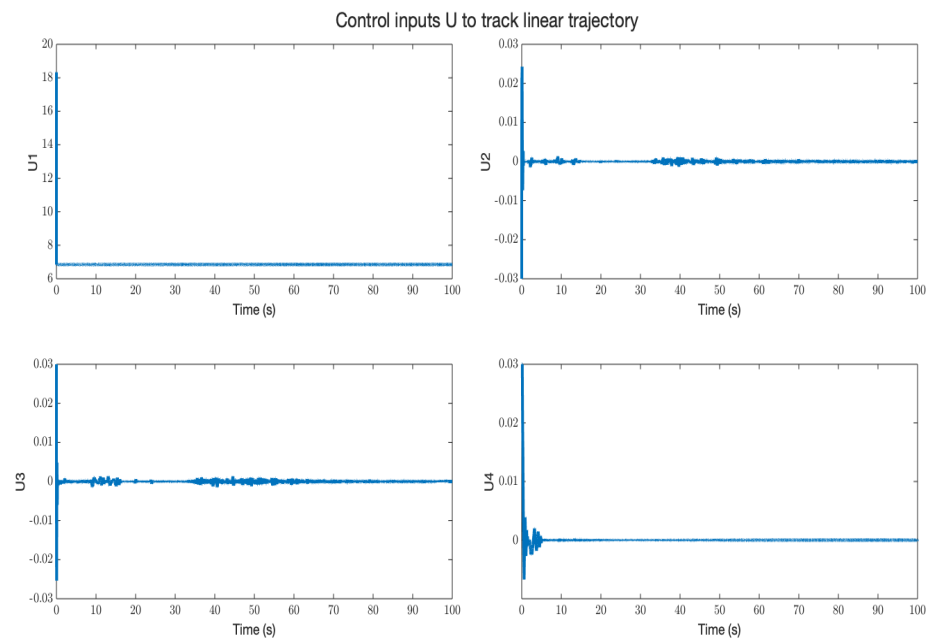


Figure 10. Control inputs U for linear trajectory tracking.

Figure 12 presents the results of the trajectory tracking for the proposed qLPV-MPC for the linear reference trajectory. Figures 13–15 show the values of the XYZ reference coordinates individually and the tracking performed by the quadrotor with the qLPV-MPC strategy.

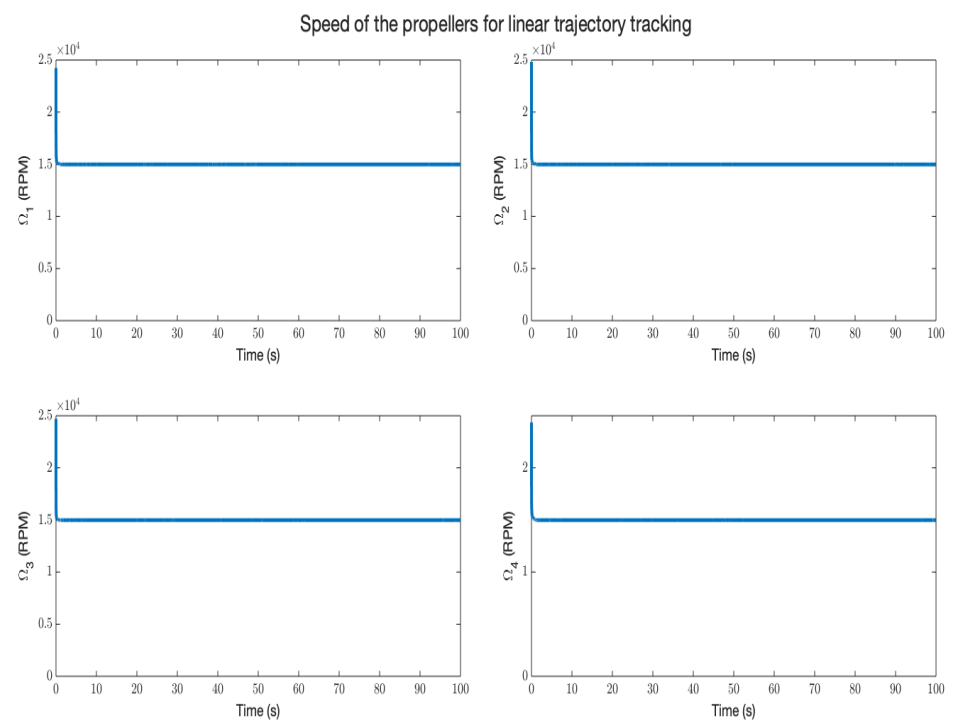


Figure 11. Speed of the propellers Ω for linear trajectory tracking.

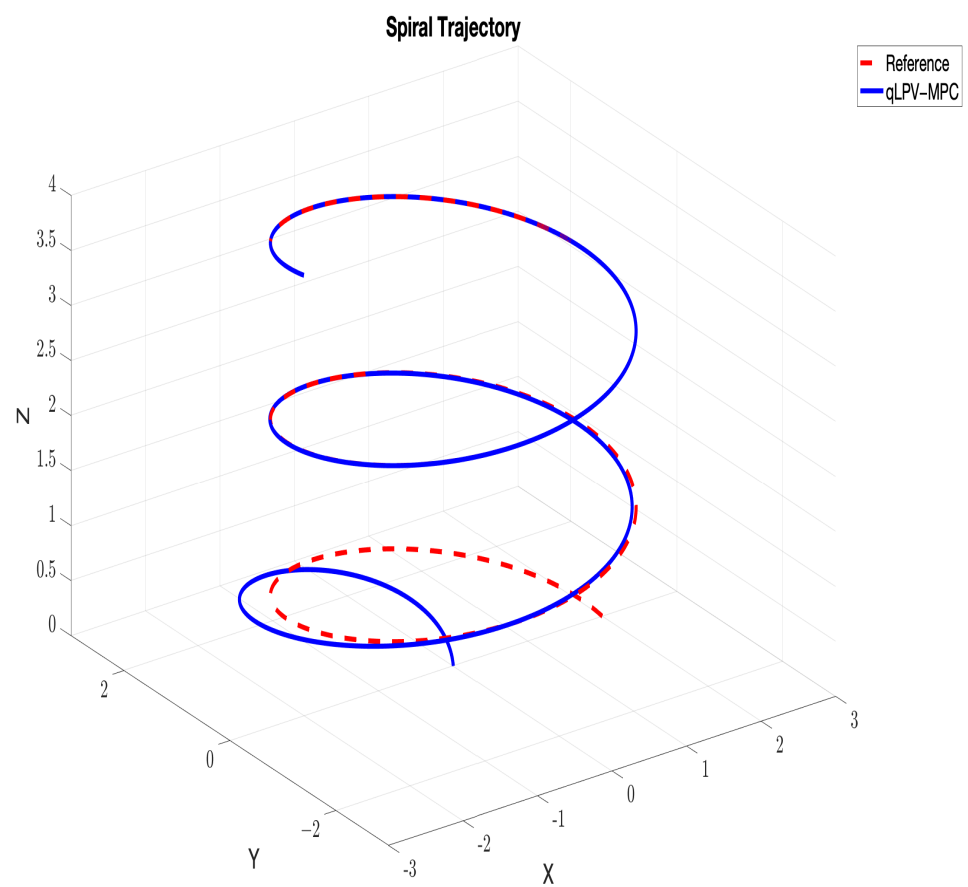


Figure 12. Tracking of the spiral trajectory using the qLPV-MPC.

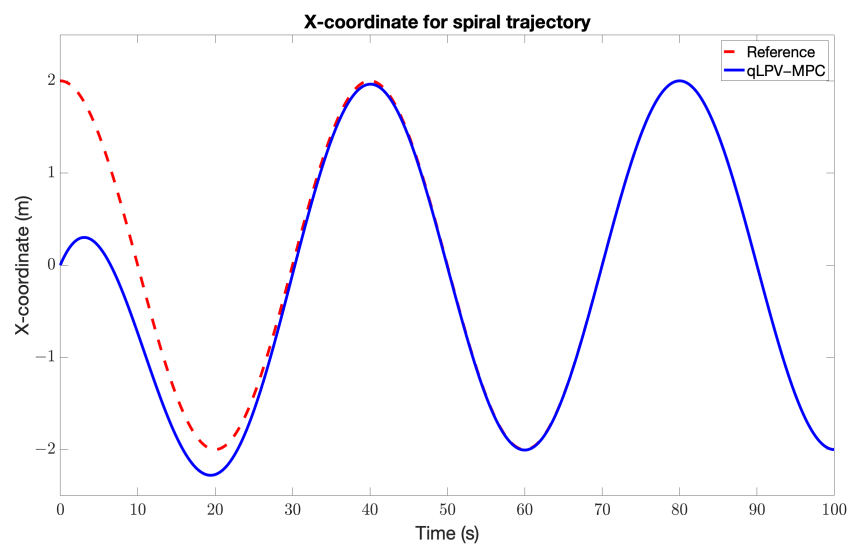


Figure 13. Tracking of the x-coordinate using the qLPV-MPC for spiral reference.

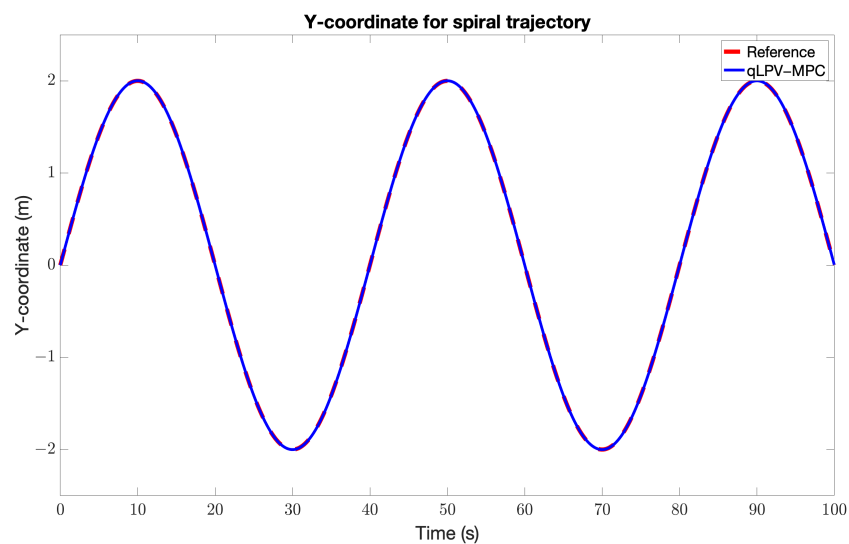


Figure 14. Tracking of the y-coordinate using the qLPV-MPC for spiral reference.

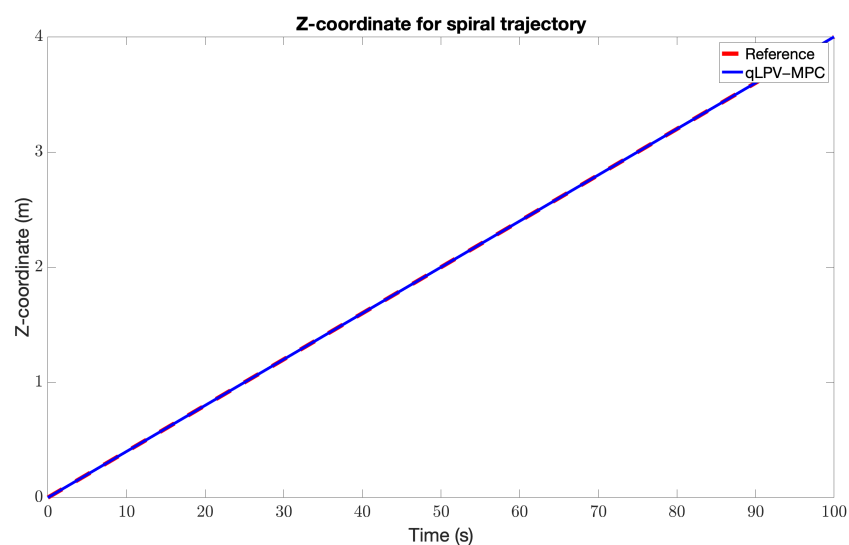


Figure 15. Tracking of the z-coordinate using the qLPV-MPC for spiral reference.

Figure 16 shows the behavior of the linear velocity in each coordinate and the angular position of the quadrotor during the tracking of the linear trajectory, while Figures 17 and 18 show the control actions chosen and the required speed for each propeller.

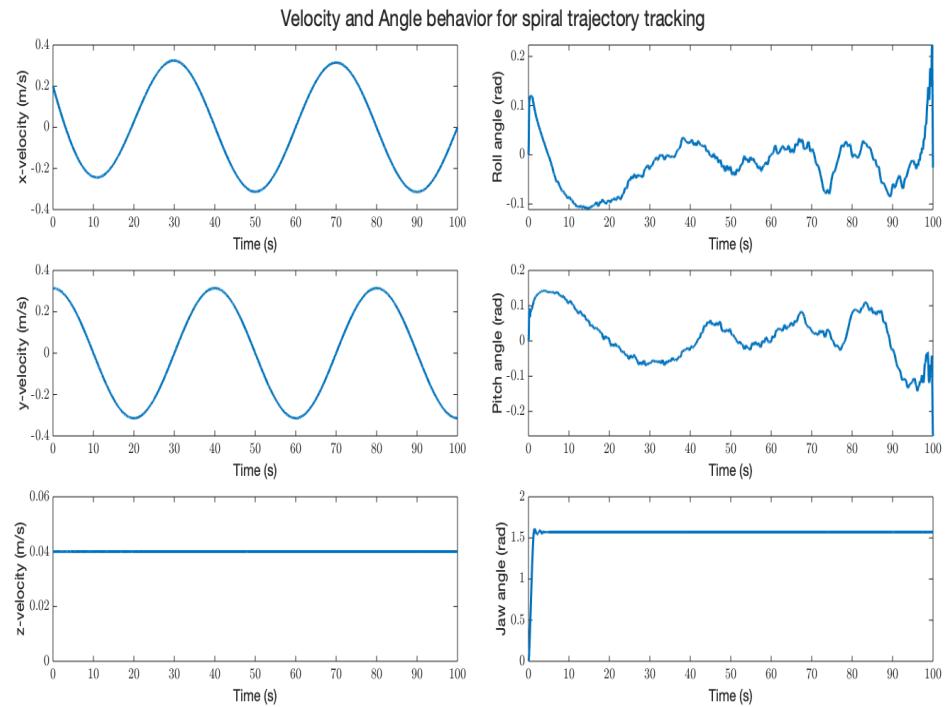


Figure 16. Behavior of the linear velocity and angular position of the quadrotor for tracking the spiral trajectory.

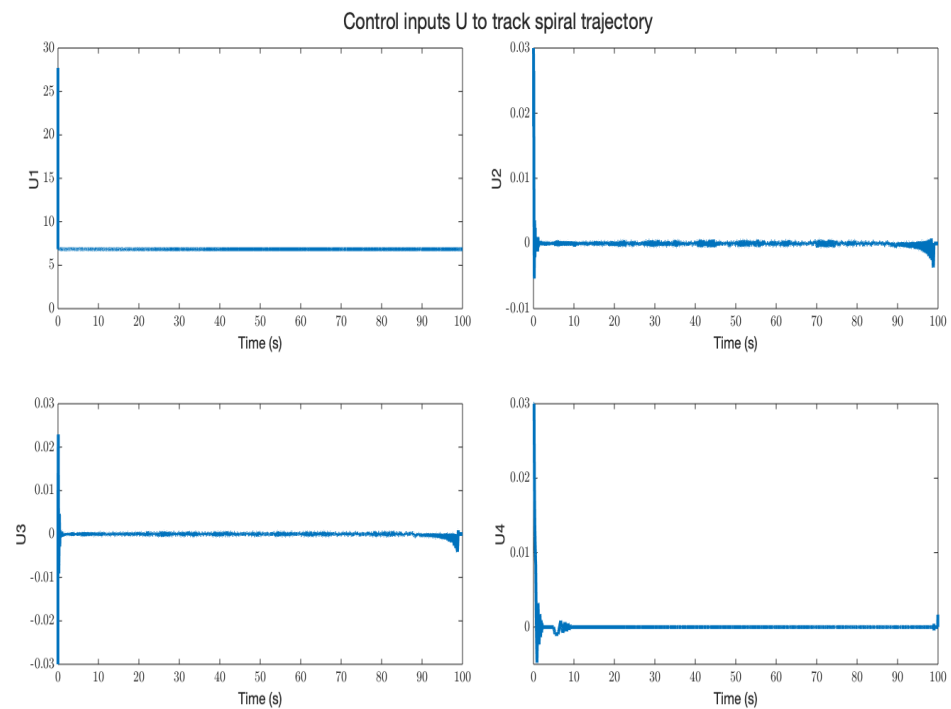


Figure 17. Control inputs U for spiral trajectory tracking.

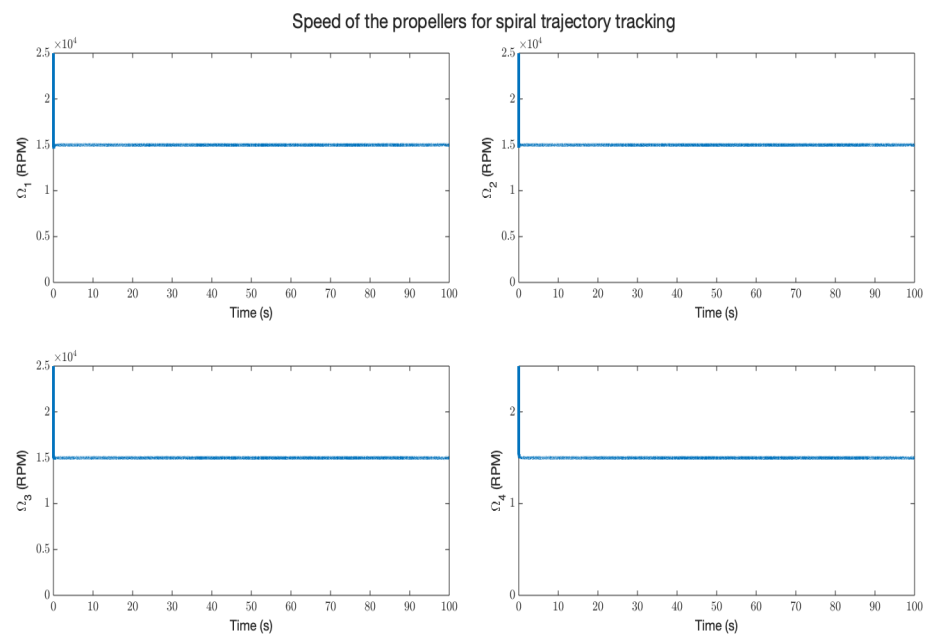


Figure 18. Speed of the propellers Ω for spiral trajectory tracking.

The control actions U present a small jitter, as shown in Figures 10 and 17, as the nonlinearities of the system rapidly change the dynamics in the qLPV system; however, the speed of the rotors Ω , which are the physical control actions, do not present this behavior and make small variations to guide the drone to the desired trajectory. The performance of the proposed qLPV-MPC control for trajectory tracking of the quadrotor proved to be appropriate for tracking both the linear and the spiral trajectory even when starting at a different initial point. The qLPV-MPC plans an optimal trajectory for the quadrotor to be incorporated into the setpoint trajectory in the XYZ coordinates without trying to reach a point in the trajectory that will be unfeasible due to the constraints in both the states and the inputs, resulting in a smooth transition to the desired trajectory. As a result of this smooth transition, the quadrotor attack angles remain at small values, in order to avoid the instability of the UAV. It is also important to note that when the starting point of the quadrotor is also the starting point of the desired trajectory, the path tracking is performed perfectly, and the reference curves and the actual trajectory overlap each other even with nonlinear changes in the reference. This is an advantage of the MPC control strategy over other classical controllers that react over the tracking error.

The proposed controller also exhibits fast execution times, with an average time of 6.14 ms per optimization each iteration, and a worst-case time of 8.76 ms using the Matlab function 'fmincon'. Therefore, the proposed qLPV-MPC allows real-time implementation for the quadrotor with a sampling time of $T_s = 10$ ms.

6. Conclusions

In this research work, a novel qLPV-MPC control strategy for a quadrotor based on a compact 12-state-space system is presented. The MPC controller is based on a qLPV system with six different scheduling parameters being states of the nonlinear quadrotor system in order to allow real-time implementation by solving the optimization problem in a short time. The prediction of the states along the prediction horizon is used in the next iterations to update the prediction matrices Φ and Ψ to model the nonlinear behavior of the quadrotor system by means of the scheduling variables. The proposed qLPV-MPC control strategy was tested by following two different trajectories, a linear trajectory, and a spiral trajectory. The results showed proper tracking of both trajectories while maintaining small roll, pitch, and yaw angles to maintain the quadrotor in a stable state. The execution times averaged

6.14 ms per optimization of each iteration which allows real-time implementation of the control strategy for the chosen sampling time of 10 ms. Future research work will focus on the robustness conditions of the proposed algorithm in the presence of wind disturbance and obstacle avoidance when following a desired path. An experimental test using a real quadrotor is also considered for future work.

Author Contributions: All authors (D.R.-G., A.F.-C., and O.J.G.-V.) have contributed as follows: Conceptualization, D.R.-G., A.F.-C., and O.J.G.-V.; Methodology, D.R.-G., A.F.-C., and O.J.G.-V.; Software, D.R.-G. and O.J.G.-V.; Validation, D.R.-G., A.F.-C., and O.J.G.-V.; Formal analysis, D.R.-G., A.F.-C., and O.J.G.-V.; Investigation, D.R.-G., A.F.-C., and O.J.G.-V.; Writing—original draft preparation, D.R.-G. and A.F.-C.; Writing—review and editing, D.R.-G., A.F.-C., and O.J.G.-V.; supervision, A.F.-C.; project administration A.F.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors thank Consejo Nacional de Ciencia y Tecnología (CONACyT) and Tecnológico de Monterrey for the financial support. Additionally, thanks go to the Nano-sensors from the School of Engineering and Sciences of Tecnológico de Monterrey for the support given to develop this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Moreno-Valenzuela, J.; Pérez-Alcocer, R.; Guerrero-Medina, M.; Dzul, A. Nonlinear PID-type controller for quadrotor trajectory tracking. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 2436–2447. [\[CrossRef\]](#)
- Abdelhay, S.; Zakriti, A. Modeling of a quadcopter trajectory tracking system using PID controller. *Procedia Manuf.* **2019**, *32*, 564–571. [\[CrossRef\]](#)
- Idres, M.; Mustapha, O.; Okasha, M. Quadrotor trajectory tracking using PID cascade control. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2017; Volume 270, p. 012010.
- El Hamidi, K.; Mjahed, M.; El Kari, A.; Ayad, H. Neural network and fuzzy-logic-based self-tuning PID control for quadcopter path tracking. *Stud. Inform. Control* **2019**, *28*, 401–412. [\[CrossRef\]](#)
- Minh, L.D.; Ha, C. Modeling and Control of Quadrotor MAV Using Vision-Based Measurement. In *Proceedings of the International Forum on Strategic Technology (IFOST)*, Ulsan, Republic of Korea, 13–15 October 2010; pp. 70–75.
- Fessi, R.; Bouallègue, S. LQG controller design for a quadrotor UAV based on particle swarm optimisation. *Int. J. Autom. Control* **2019**, *13*, 569–594. [\[CrossRef\]](#)
- Bouselima, E.; Ichlal, D.; Mammari, S. Quadrotor Control and Actuator Fault Detection: LQG Versus Robust H_{∞} observer. In *Proceedings of the 2019 4th Conference on Control and Fault Tolerant Systems (SysTol)*, Casablanca, Morocco, 18–20 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 86–91.
- Emam, M.; Fakharian, A. Attitude tracking of quadrotor UAV via mixed H_2/H_{∞} controller: An LMI based approach. In *Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED)*, Athens, Greece, 21–24 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 390–395.
- Guo, M.; Su, Y.; Gu, D. Mixed H_2/H_{∞} tracking control with constraints for single quadcopter carrying a cable-suspended payload. *IFAC-PapersOnLine* **2017**, *50*, 4869–4874. [\[CrossRef\]](#)
- Jasim, W.; Gu, D. H_{∞} path tracking control for quadrotors based on quaternion representation. In *Advances in Autonomous Robotics Systems, Proceedings of the 15th Annual Conference, TAROS 2014, Birmingham, UK, 1–3 September 2014*; Proceedings 15; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 72–84.
- Kayacan, E.; Maslim, R. Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions. *IEEE/ASME Trans. Mechatron.* **2016**, *22*, 339–348. [\[CrossRef\]](#)
- Prayitno, A.; Indrawati, V.; Utomo, G. Trajectory tracking of AR. Drone quadrotor using fuzzy logic controller. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2014**, *12*, 819–828. [\[CrossRef\]](#)
- Zhang, C.; Zhou, X.; Zhao, H.; Dai, A.; Zhou, H. Three-dimensional fuzzy control of mini quadrotor UAV trajectory tracking under impact of wind disturbance. In *Proceedings of the 2016 International Conference on Advanced Mechatronic Systems (ICAMEchS)*, Melbourne, VIC, Australia, 30 November–3 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 372–377.

14. Zhou, L.; Zhang, J.; Dou, J.; Wen, B. A fuzzy adaptive backstepping control based on mass observer for trajectory tracking of a quadrotor UAV. *Int. J. Adapt. Control Signal Process.* **2018**, *32*, 1675–1693. [\[CrossRef\]](#)
15. Ganga, G.; Dharmana, M.M. MPC controller for trajectory tracking control of quadcopter. In Proceedings of the 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Kollam, India, 20–21 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
16. Eskandarpour, A.; Sharf, I. A constrained error-based MPC for path following of quadrotor with stability analysis. *Nonlinear Dyn.* **2020**, *99*, 899–918. [\[CrossRef\]](#)
17. Benotsmane, R.; Reda, A.; Vásárhelyi, J. Model Predictive Control for Autonomous Quadrotor Trajectory Tracking. In Proceedings of the 2022 23rd International Carpathian Control Conference (ICCC), Sinaia, Romania, 29 May–1 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 215–220.
18. Abdolhosseini, M.; Zhang, Y.M.; Rabbath, C.A. Trajectory tracking with model predictive control for an unmanned quad-rotor helicopter: Theory and flight test results. In *Intelligent Robotics and Applications, Proceedings of the 5th International Conference, ICIRA 2012, Montreal, QC, Canada, 3–5 October 2012*; Springer: Berlin/Heidelberg, Germany, 2012; Part I 5, pp. 411–420.
19. Huang, S.; Teo, R.S.H.; Tan, K.K. Collision avoidance of multi unmanned aerial vehicles: A review. *Annu. Rev. Control* **2019**, *48*, 147–164. [\[CrossRef\]](#)
20. Wang, D.; Pan, Q.; Shi, Y.; Hu, J.; Zhao, C. Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment. *IEEE Trans. Cybern.* **2021**, *51*, 5057–5068. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Liu, C.; Lu, H.; Chen, W.H. An explicit MPC for quadrotor trajectory tracking. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 4055–4060.
22. Kapnopoulos, A.; Alexandridis, A. A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme. *Aerosp. Sci. Technol.* **2022**, *127*, 107725. [\[CrossRef\]](#)
23. Zhao, C.; Wang, D.; Hu, J.; Pan, Q. Nonlinear model predictive control-based guidance algorithm for quadrotor trajectory tracking with obstacle avoidance. *J. Syst. Sci. Complex.* **2021**, *34*, 1379–1400. [\[CrossRef\]](#)
24. Guevara, B.S.; Recalde, L.F.; Varela-Aldás, J.; Andaluz, V.H.; G.; olfo, D.C.; Toibero, J.M. A Comparative Study between NMPC and Baseline Feedback Controllers for UAV Trajectory Tracking. *Drones* **2023**, *7*, 144. [\[CrossRef\]](#)
25. Stastny, T.J.; Dash, A.; Siegart, R. Nonlinear mpc for fixed-wing uav trajectory tracking: Implementation and flight experiments. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Grapevine, TX, USA, 9–13 January 2017; p. 1512.
26. Misin, M.; Puig, V. LPV MPC control of an autonomous aerial vehicle. In Proceedings of the 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphael France, 15–18 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 109–114.
27. Cavanini, L.; Ippoliti, G.; Camacho, E.F. Model predictive control for a linear parameter varying model of an UAV. *J. Intell. Robot. Syst.* **2021**, *101*, 57. [\[CrossRef\]](#)
28. Singh, B.K.; Kumar, A. Model predictive control using LPV approach for trajectory tracking of quadrotor UAV with external disturbances. *Aircr. Eng. Aerosp. Technol.* **2023**, *95*, 607–618. [\[CrossRef\]](#)
29. Qu, S.; Zhu, G.; Su, W.; Swei, S.S.M. LPV Model-based Adaptive MPC of an eVTOL Aircraft During Tilt Transition Subject to Motor Failure. *Int. J. Control Autom. Syst.* **2023**, *21*, 339–349. [\[CrossRef\]](#)
30. Rodríguez Hernández, X. LPV Predictive Control of a Quadrotor. Master's Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.