sensors

MDPI

*Article*

# System Identification and Nonlinear Model Predictive Control with Collision Avoidance Applied in Hexacopters UAVs

**Luis F. Recalde [1], Bryan S. Guevara [2], Christian P. Carvajal [2], Victor H. Andaluz [3], José Varela-Aldás [1,4,*] and Daniel C. Gandolfo [2]**

[1] SISAu Research Group, Facultad de Ingeniería y Tecnologías de la Información y Comunicación, Universidad Tecnológica Indoamérica, Ambato 180103, Ecuador; fernandorecalde@uti.edu.ec

[2] Instituto de Automática, Universidad Nacional de San Juan-CONICET, Av. San Martín Oeste 1109, San Juan J5400ARL, Argentina; bguevara@inaut.unsj.edu.ar (B.S.G.); cpcarvajal@inaut.unsj.edu.ar (C.P.C.); dgandolfo@inaut.unsj.edu.ar (D.C.G.)

[3] Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas–ESPE, Sangolquí 171103, Ecuador; vhandaluz1@espe.edu.ec

[4] Department of Electronic Engineering and Communications, University of Zaragoza, 44003 Teruel, Spain

* Correspondence: josevarela@uti.edu.ec or 771638@unizar.es

**Abstract:** Accurate trajectory tracking is a critical property of unmanned aerial vehicles (UAVs) due to system nonlinearities, under-actuated properties and constraints. Specifically, the use of unmanned rotorcrafts with accuracy trajectory tracking controllers in dynamic environments has the potential to improve the fields of environment monitoring, safety, search and rescue, border surveillance, geology and mining, agriculture industry, and traffic control. Monitoring operations in dynamic environments produce significant complications with respect to accuracy and obstacles in the surrounding environment and, in many cases, it is difficult to perform even with state-of-the-art controllers. This work presents a nonlinear model predictive control (NMPC) with collision avoidance for hexacopters' trajectory tracking in dynamic environments, as well as shows a comparative study between the accuracies of the Euler–Lagrange formulation and the dynamic mode decomposition (DMD) models in order to find the precise representation of the system dynamics. The proposed controller includes limits on the maneuverability velocities, system dynamics, obstacles and the tracking error in the optimization control problem (OCP). In order to show the good performance of this control proposal, computational simulations and real experiments were carried out using a six rotary-wind unmanned aerial vehicle (hexacopter—DJI MATRICE 600). The experimental results prove the good performance of the predictive scheme and its ability to regenerate the optimal control policy. Simulation results expand the proposed controller in simulating highly dynamic environments that showing the scalability of the controller.

**Keywords:** system identification; model predictive control; obstacles avoidance; hexacopter UAV; system constraints; optimization

check for updates

**Citation:** Recalde, L.F.; Guevara, B.S.; Carvajal, C.P.; Andaluz, V.H.; Varela-Aldás, J.; Gandolfo, D.C. System Identification and Nonlinear Model Predictive Control with Collision Avoidance Applied in Hexacopters UAVs. *Sensors* **2022**, *22*, 4712. https://doi.org/10.3390/s22134712

Academic Editor: Sindhuja Sankaran

Received: 9 May 2022
Accepted: 18 June 2022
Published: 22 June 2022

## 1. Introduction

### 1.1. Motivation

In recent years, the field of robotics has experienced exponential growth due to fast technological advances. The applications are not restricted to the industrial field, as nowadays robots are equipped with sophisticated and intelligent algorithms that are useful for performing complex task in unstructured and natural environments with a high degree of autonomy. The integration between robots and humans through service robotics has the objective of facilitating activities of daily life or improving repetitive tasks. As such, we can find mobile manipulators that are used to load transportation in structured environments [1,2], robotic systems that help people during their rehabilitation process [3] and aerial robotic vehicles that are useful in rescue or transportation operations [4]. Robot applications can be divided into four main groups: *(i) Learning robots,* which are robotic systems for whom the main objective is to improve learning and stimulation processes [5,6]. *(ii) Construction robots,* which are systems that can be programmed to develop complex repetitive tasks, where the main contributions are the decrease in production time and the prevention of accidents in dangerous workplaces [7,8]. *(iii) Field robots,* which show a variety of improvements in mining industries, agricultural and livestock, increasing the economy of the population [9,10]. *(iv) Military robots,* which are designed to perform exploration, security, rescue, among other applications in hostile environments that can be dangerous for humans, and these applications are highly developed using unmanned aerial vehicles (UAVs) [11].

The use of unmanned aerial vehicles have grown in a wide range of fields and has been extensively studied by the scientific community due to the versatility and the possibility to perform the fully autonomous tasks that humans are unable to. The extremely agility of the UAVs with four or more rotors has attracted attention due to their ability to vertically take off and land in confined or difficult to access spaces, which eases construction and enables having a higher payload capacity, which are useful in critical missions such as transportation, rescue, search, surveillance and disaster assistance. In these applications, the execution time is an extremely important feature due to the task only being executed for high-speed trajectories. To take advantage of UAVs, it is necessary to guarantee the safe execution of the application; however, dynamic environments are spaces with highly probabilities of collisions, where obstacle information is unknown at the beginning of the task, and thus the UAV must be capable of avoiding the obstacles using online-scheme information. Obstacle avoidance is an important feature that guarantees the correct functionality of UAV applications. In this context, research efforts are still needed to safely introduce UAVs as a technological tool with great potential to solve various current problems such as a collision avoidance control structure that can perform high-speed trajectories in dynamic environments.

### *1.2. Related Work*

Depending on the environment, UAV applications can be divided in two groups: *(i) Outdoor environments* where many studies have employed UAVs with onboard sensors, e.g., Mellinger et al. presented collaborative grasping to logistic applications [12]; Bircher et al. performed path planning for structural inspection using UAVs [13]; search and rescue operations were developed by Oetter- shagen et al. in [14]; and an application consisting of tasks related to physical interaction with the environment was developed by Garimella et al. [15]. *(ii) Indoor environments* are the counterpart applications, developed in cluttered indoor experiments, examples of which include the study by Song and Hsu who presented navigation based on factor graph optimization (FGO) [16]; Paredes et al., who developed a hybrid acoustic and optical positioning system for accurate 3D movements [17]; and Sandino et al., who proposed an autonomous navigation system using partial observable and uncertainty sensor measurements [18]. Due to the different types of UAVs, these systems have the ability to perform a wide range of applications that are emerging. The most popular configuration of multi-rotors is the quadrotor UAV, which is applied in a variety of aerial fields, e.g., Radoglou–Grammatiski et al., who presented the compilation of the most important applications for precision agriculture using quadrotors [19]; and Gupta et al., who presented the advances of UAVs and the applications in transportation systems [20]. This configuration only presents four motors attached to the mechanical frame generating a limiting power that restricts the fields of use, especially for rescue and transportation tasks. However, nowadays the design of a multi-rotor with more than four rotors is a rapidly growing industry because multi-rotor configuration has presented important advances due to its high maneuverability and transportation capacity [21,22]. More works were conducted by Belmonte et al., who designed an octocopter for the inspection of mobile cranes [23]; Phong-Nguyen et al., who presented a sliding mode control algorithm with a fuzzy inference system considering variable gains for hexacopters [24]; and Ali et al., who developed adaptive backstepping for the attitude and altitude of coaxial octorotors [25].

The use of more rotors can produce high costs in production, a considerably increased size of the UAV and complex dynamics. Nowadays, it still a challenge to design effective controllers that ensure good performance in multi-rotors at high-speed trajectories. Precise trajectory tracking is an important feature for multi-rotors operating in real-world environments due to the considerable size of the structure and possible collision with obstacles. The main reasons that make this configuration more complex than the others are presented below: *(i) System actuation:* An inherently under-actuated system due to there only being four control inputs over the six degrees of freedom (DOF). *(ii) Dynamics behavior:* high nonlinearities produced by translational and rotational dynamics due to the considerable size of the system. *(iii) Dynamic model mismatch:* Unmodeled dynamics, uncertainties and disturbances produced for high-speed trajectories and the possibility of external forces such as wind in outdoor environments. All of these problems incited the special attention of researchers from different fields. As a consequence, a wide variety of control strategies have been proposed for the problem of trajectory tracking for hexacopters. Initially, some linear control methods were presented, and these works considered small angles' assumptions that are necessary for linear control techniques, and the most common structures were presented by Alaimo et al., who developed a proportional–integral and derivative controller (PID) under linear assumptions in UAVs with an hexacopter configuration [26] and Salim et al. presented optimum linear control that stabilizes the attitude of a micro-hexacopter in indoor environments [27].

Linear models are only valid around the hover position, providing slow movements where the controllers cannot guarantee the convergence to the desired trajectories. In order to improve the performance and to satisfy the requirements of high-speed trajectories, a large number of studies have used the principles of nonlinear control, e.g., Chen et al., who proposed a nonlinear trajectory controller for UAVs based on backstepping and nonlinear observer [28] and Wang et al. presented the sliding mode control (SMC) with a variable structure to generate the control inputs [29]. Recently, many authors have proposed using geometric tracking controllers, e.g., Lee et al. showed the results of trajectory tracking using a nonlinear controller developed in a Euclidean group $SE(3)$ [30]; Mellinger et al. revealed the differential flatness properties in UAVs and the ability to derive the attitude, acceleration and angular rate [31]. The property of differential flatness improves the accurate trajectory tracking under high-speed references and it was demonstrated by [32,33]. In recent years, intelligent control approaches were used to developed advance control systems, and these schemes included artificial neural networks and fuzzy logic controllers that are highly used in multi-rotor UAVs; the results present interesting behaviors under disturbances and parameter changes [34–37].

Unlike multi-rotor control techniques, obstacle avoidance has not been highly developed by the research community. Obstacle avoidance is a relatively new field and most works have used the concept of artificial potential fields, a method which is based on the generation of repulsive and attractive forces that allow the movements of the system around obstacles. The literature presents the extensive applications with UAVs, but these works revealed the problem of a local minimal that cannot guarantee the execution of the task [38,39]. Nowadays, other methods have been proposed to solve the problem of obstacle avoidance, the most popular among which include: *(i) rapidly exploring random tree (RRT)*, a formulation which generates a tree rooted at the initial configuration of the system and using random samples to create edges between feasible points. The work by Achtelik et al., who presented an RRT algorithm with a modification rapidly exploring random belief tree (RRBT), which was used to generate the collision free path using quadrotor UAVs [40]; *(ii) The belief roadmap*, which is a probabilistic version of the roadmap, which uses the Kalman filter for the planning motion, and has the ability to generate trajectories by avoiding obstacles; this structure was presented by [41]. *(iii) The minimum snap trajectory generation* this algorithm includes navigation through corridors whilst considering constraints on velocities and accelerations, the solution to which is based on the generation of optimal trajectories that minimizes a functional cost confirmed by square norm of the snap, an application of which was presented in [31]. All the obstacle avoidance algorithms presented beforehand only consider the prior knowledge of the obstacle in the environment, whilst also requiring high computational times and complex operations.

Taking into account the problems of no-dynamics constraints and only the prior knowledge of environments, the results thus produce unfeasible control actions and problems in dynamic environments with highly probabilities of collisions during the task execution. To fully exploit hexacopters' capabilities and take advantage of the available computation power, optimization-based control techniques are becoming suitable for real-time problems. Model predictive control (MPC) has been used in many fields of robotics due to the accuracy of its tracking trajectories, robust performance and the ability to include constraints in the controller scheme. In function of the field, the general applications can be divided into the following groups:

- Ground mobile robots: where Sani et al. presented nonlinear model predictive control (NMPC) to solve the competitive games between ground robots [42], another work undertaken developed by Subramanian et al. increased the ability to avoid dynamic obstacles [43], and finally, a leader–follower structure control using NMPC through visual information to mobile robots was presented by Ribeiro et al. [44].
- Robotic arms and manipulation: In the last year, this field of research has been the focus of many studies due to the complexity of the systems and the scalability of the NMPC structure. The work of Osman et al. presented a task-space controller based on (NMPC) to control a mobile manipulator 10 (DOF) [45].
- Aerial mobile robots: this field has also been the subject of a large number of studies due to the high use of aerial vehicles in real-world applications, some applications of which were studies Neunert et al., who developed an unconstrained nonlinear predictive model for generation and tracking trajectories for the AscTec Firefly hexacopter [46]; Aoki et al., who presented an NMPC for position and attitude control applied in a hexacopter without three of the six motor configurations [47]; and finally, the contribution of Tzoumanikas et al., a letter which presented the NMPC designed for micro aerial vehicles (MAVs) equipped with a robotic arm [48].

This kind of controller has had a huge impact on the field of robotics thanks to the predictive behavior and ability to introduce system constraints in the controller scheme. The main idea of this controller is to generate the control actions trying to minimize a cost function over a prediction horizon. The cost function is solved using constrained optimization techniques; however, the resolution of constrained optimization problems is computationally expensive. Due to the recent developments in hardware and nonlinear optimization solvers, nonlinear predictive control is computationally applicable in many fields and has received the attention of a hexacopter control field [49–53].

### *1.3. Main Contributions of This Work*

Run NMPC structures are realizable in modern computers and guarantee the execution of complex tasks but still require more computational time than the schemes mentioned beforehand. It is necessary to show performance under tracking accuracy and the ability to extend this technique to dynamic environments in order to add obstacles in the optimization problem. With all the aforementioned controllers and the significant results in many areas, this work presents the nonlinear model predictive control to fast trajectory tracking in dynamic environments applied in hexacopter platforms. The controller best generates the control policy that moves the hexacopter to the reference trajectory while avoiding obstacles through the environment. To construct the approach, this work developed the important contributions presented below:

- System dynamics: a large number of research platforms cannot be controlled through torque commands; as such, this work developed a reduced dynamic model of the hexacopter using general control velocities, which is possible with the incorporation of low-level PID schemes in mathematical representation. The PIDs guarantee the required velocities generated by the high-level controllers. The mathematical representation was developed through an Euler–Lagrange formulation and the data-driven technique known as dynamic mode decomposition (DMDc). Both approaches were identified using experimental data from the (DJI MATRICE 600 PRO); furthermore, this work presents accuracy comparative results between both techniques and selects the best formulation to build the controller scheme.

- NMPC formulation: this includes the differential kinematics and the reduced dynamic representation; furthermore, due to the ability of the NMPC controller, the system constraints can be included in the optimization problem. The constraints included in this work are: control action limits, control rate of change, and the system dynamics and static and dynamic obstacles. The nonlinear optimization problem was solved using CasADI framework and the problem was transformed into a nonlinear programming problem (NLP) form using the direct multiple-shooting method. The CasADI optimization toolkit guarantees a fast convergence with the possibility of extending the solution to onboard hardware implementation.
- Results: the experiments were conducted in real-world outdoor environments, where the system has a model mismatch and external disturbance product by air flows and delay in system communications. The experiments were developed through the hexacopter platform (DJI MATRICE 600 PRO) where multiple reference trajectories were selected in order to verify the performance of the controller. The metrics used in the experiments are the following: tracking accuracy, computational time and disturbance rejection. Additionally, the simulation results show the scalability in simulated high dynamic environments, considering the identified dynamics and obstacles; these results will be used as a start point in future research due to the fact that the hexacopter aerial platform does not have any sensor to measure obstacles during the experiments.

### *1.4. Outline*

This work is structured as follows. Section 2 presents the instantaneous kinematics with the reduced dynamic model and system identification and validation. Furthermore, the controller formulation is presented in this section, which consists of the nonlinear model predictive controller with an obstacles avoidance scheme. Section 3 shows the real-world experiments and simulation results; furthermore, considering the experiment results, this section exhibits a review of the important aspects and the future research direction. The conclusions of the work are presented in Section 4.

## **2. Materials and Methods**

### *2.1. Hexacopter System Preliminaries*

This section presents the instantaneous kinematics and the dynamic model. The dynamic model was formulated in maneuverability velocities space using two different formulations: the Euler–Lagrange and dynamic mode decomposition. After the formulation process, this work presents the validation results in order to verify the performance of each formulation.

Figure 1 shows the hexacopter platform DJI MATRICE 600 PRO, where the world-fixed inertial frame is represented by $< \mathbb{I} >$ with the following unit vectors $\{\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z\}$ and the body-fixed frame attached to hexacopter movements is defined by $< \mathbb{B} >$ with the unit vectors $\{\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z\}$, where the center of mass (CoM) is aligned; furthermore, the hexacopter is configured with six motors in the mechanical frame which allows the movement of the system.

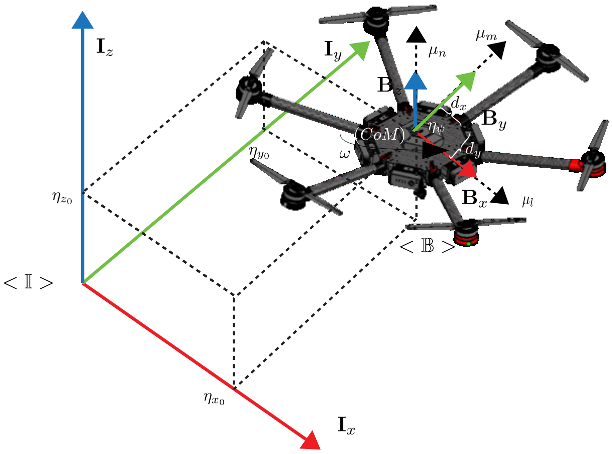

**Figure 1.** Hexacopter platform representation.

### 2.1.1. Kinematics Model

The robot has the ability to move through the inertial frame $< \mathbb{I} >$ and it is enabled to rotate only in yaw $\psi$ defined in the vertical axis $\mathbf{B}_z$, whilst the others' angular rotations roll and pitch $(\phi, \theta)$ are not considered in this work due to this hexacopter platform having a low level flight controller which guarantees the hover position. The position and orientation of the point to be controlled is defined by the vector $\boldsymbol{\eta} = \begin{bmatrix} \eta_x & \eta_y & \eta_z & \eta_\psi \end{bmatrix}^T \in \mathbb{R}^4$ with respect to the frame $< \mathbb{I} >$, and the following equation defines these elements in greater detail:

$$\begin{cases} \eta_x = \eta_{x_0} + d_x \cos(\psi) - d_y \sin(\psi) \\ \eta_y = \eta_{y_0} + d_x \sin(\psi) + d_y \cos(\psi) \\ \eta_z = \eta_{z_0} + d_z \\ \eta_\psi = \psi \end{cases} \tag{1}$$

where the values $(\eta_{x_0}, \eta_{y_0}, \eta_{z_0})$ are the locations of the CoM and $(d_x, d_y, d_z)$ are the distances in the body-fixed frame $< \mathbb{B} >$ to the point of interest $\boldsymbol{\eta}$. In order to know the evolution of the interest point, it is necessary to introduce the concept of the instantaneous kinematic, defining the time derivative as $\dot{\boldsymbol{\eta}} = \frac{\partial \boldsymbol{\eta}}{\partial t}\boldsymbol{\mu}$. With these considerations, the velocity of the point is defined by the vector $\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\eta}_x & \dot{\eta}_y & \dot{\eta}_z & \dot{\eta}_\psi \end{bmatrix}^T \in \mathbb{R}^4$ with respect to the frame $< \mathbb{I} >$; furthermore, the time derivative produces maneuverability velocities $\boldsymbol{\mu}$ attached to the frame $< \mathbb{B} >$. Due to the low-level PIDs controllers, the maneuverability vector is defined by $\boldsymbol{\mu} = \begin{bmatrix} \mu_l & \mu_m & \mu_n & \omega \end{bmatrix}^T \in \mathbb{R}^4$, with the longitudinal velocities $(\mu_l, \mu_m, \mu_n)$ through the axis $(\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z)$ and the angular rate of change $\omega$ over unit vector $\mathbf{B}_z$. Using the definition of instantaneous kinematics, the evolution can be defined by the following equation:

$$\begin{cases} \dot{\eta}_x = \mu_l \cos(\psi) - \mu_m \sin(\psi) - (d_x \sin(\psi) + d_y \cos(\psi))\omega \\ \dot{\eta}_y = \mu_l \sin(\psi) + \mu_m \cos(\psi) + (d_x \cos(\psi) - d_y \sin(\psi))\omega \\ \dot{\eta}_z = \mu_n \\ \dot{\eta}_\psi = \omega \end{cases} \tag{2}$$

the equation presented before can be written in a matrix form, defined as:

$$\begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_z \\ \dot{\eta}_\psi \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & -(d_x \sin(\psi) + d_y \cos(\psi)) \\ \sin(\psi) & \cos(\psi) & 0 & (d_x \cos(\psi) - d_y \sin(\psi)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \tag{3}$$

where the expressions $-(d_x \sin(\psi) + d_y \cos(\psi))$ and $(d_x \cos(\psi) - d_y \sin(\psi))$ represent the additional behavior considering the displacement of the point of interest. To simplify the notation, Equation (3) is expressed in the compact form in (4)

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{J}(\psi(t))\boldsymbol{\mu}(t) \tag{4}$$

where $\mathbf{J}(\psi(t)) \in \mathbb{R}^{4\times 4}$ is the Jacobian matrix which allows the linear mapping between the control maneuverability velocities $\boldsymbol{\mu}$ to the evolution of the point of interest $\dot{\boldsymbol{\eta}}$, Equation (3) is expressed in vector-function form as:

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{f}_k(\psi(t), \boldsymbol{\mu}(t)) \tag{5}$$

Considering that $\ddot{\boldsymbol{\eta}}(t) = \frac{d}{dt}\dot{\boldsymbol{\eta}}$, the following relation was developed:

$$\begin{aligned} \begin{bmatrix} \ddot{\eta}_x \\ \ddot{\eta}_y \\ \ddot{\eta}_z \\ \ddot{\eta}_\psi \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & -(d_x \sin(\psi) + d_y \cos(\psi)) \\ \sin(\psi) & \cos(\psi) & 0 & (d_x \cos(\psi) - d_y \sin(\psi)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\mu}_l \\ \dot{\mu}_m \\ \dot{\mu}_n \\ \dot{\omega} \end{bmatrix} + \dots \\ \begin{bmatrix} -\omega \sin(\psi) & -\omega \cos(\psi) & 0 & -\omega(d_x \cos(\psi) - d_y \sin(\psi)) \\ \omega \cos(\psi) & -\omega \sin(\psi) & 0 & \omega(d_x \sin(\psi) + d_y \cos(\psi)) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \end{aligned} \tag{6}$$

it can be written in compact form as $\ddot{\boldsymbol{\eta}}(t) = \mathbf{J}(\psi)\dot{\boldsymbol{\mu}}(t) + \dot{\mathbf{J}}(\psi, \omega)\boldsymbol{\mu}(t)$, where $\dot{\boldsymbol{\mu}} = \begin{bmatrix} \dot{\mu}_l & \dot{\mu}_m & \dot{\mu}_n & \dot{\omega} \end{bmatrix}^T \in \mathbb{R}^4$ is the vector of maneuverability accelerations, and this formulation will be used in the dynamic model section.

### 2.1.2. Dynamic Model

This section presents the dynamic model, which has been developed considering low-level PIDs controllers that only generate longitudinal movements in $\{\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z\}$ and the angular rotation through $\mathbf{B}_z$. The dynamic model is useful for guaranteeing the stability of the proposed controller in real-world applications.

#### Euler–Lagrange Formulation

One of the approaches to find the mathematical model is the Euler–Lagrange formulation. This formulation is a convenient analytical method for obtaining the dynamic model and studying the physical phenomena of the hexacopter. The Euler–Lagrange formulation uses the total energy of the system confirmed by kinetic and potential energy. The total kinetic energy is defined by the following equation:

$$T(\dot{\boldsymbol{\eta}}) = \frac{1}{2}\dot{\boldsymbol{\eta}}^T \mathbf{M}\, \dot{\boldsymbol{\eta}} \tag{7}$$

where $\mathbf{M} = diag(m, m, m, I)$ is a diagonal matrix confirmed by the mass of the hexacopter $m$ and the moment of inertia $I$ around the vertical axis $\mathbf{B}_z$. On the other hand, the potential energy is the position or configuration of the system with respect to the world-fixed frame $< \mathbb{I} >$, and the potential energy is described by:

$$V(\boldsymbol{\eta}) = mg(\eta_z + d_z) \tag{8}$$

where $g$ represents the gravitational acceleration. The Lagrange formulation is obtained by the subtraction of the kinetic $T(\dot{\boldsymbol{\eta}})$ and potential $V(\boldsymbol{\eta})$ energy expressed as:

$$L = \frac{1}{2}\dot{\boldsymbol{\eta}}^T \mathbf{M}\, \dot{\boldsymbol{\eta}} - mg(\eta_z + d_z) \tag{9}$$

Now, applying the Euler–Lagrange formulation:

$$\frac{d}{dt}\left(\frac{dL}{\dot{\boldsymbol{\eta}}}\right) - \frac{dL}{\boldsymbol{\eta}} = \mathbf{f_I} \tag{10}$$

where $\mathbf{f_I} = \begin{bmatrix} f_{x_I} & f_{y_I} & f_{z_I} & \tau_{\psi_I} \end{bmatrix}^T \in \mathbb{R}^4$ are the generalized forces and torque vector with respect to the frame $< \mathbb{I} >$; with this formulation, it is possible to obtain the nonlinear model by the following equations:

$$\begin{cases} m\ddot{\eta}_x - d_y m\cos(\psi)\ddot{\eta}_\psi - d_x m\sin(\psi)\ddot{\eta}_\psi - d_x m\cos(\psi)\dot{\eta}_\psi{}^2 + d_y m\sin(\psi)\dot{\eta}_\psi{}^2 = f_{x_I} \\ m\ddot{\eta}_y + d_x m\cos(\psi)\ddot{\eta}_\psi - d_y m\sin(\psi)\ddot{\eta}_\psi - d_y m\cos(\psi)\dot{\eta}_\psi{}^2 - d_x m\sin(\psi)\dot{\eta}_\psi{}^2 = f_{y_I} \\ m\ddot{\eta}_z + mg = f_{z_I} \\ I\ddot{\eta}_\psi + d_x{}^2 m\ \ddot{\eta}_\psi + d_y{}^2 m\ \ddot{\eta}_\psi - d_y m\cos(\psi)\ddot{\eta}_x + d_x m\cos(\psi)\ddot{\eta}_y - d_x m\sin(\psi)\ddot{\eta}_x - d_y m\sin(\psi)\ddot{\eta}_y = \tau_{\psi_I} \end{cases} \tag{11}$$

the equations presented in (11) can be written in a matrix form, resulting in:

$$\begin{aligned} \begin{bmatrix} f_{x_I} \\ f_{y_I} \\ f_{z_I} \\ \tau_{\psi_I} \end{bmatrix} &= \begin{bmatrix} m & 0 & 0 & -m(d_y\cos(\psi) + d_x\sin(\psi)) \\ 0 & m & 0 & m(d_x\cos(\psi) - d_y\sin(\psi)) \\ 0 & 0 & m & 0 \\ -m(d_y\cos(\psi) + d_x\sin(\psi)) & m(d_x\cos(\psi) - d_y\sin(\psi)) & 0 & md_x{}^2 + md_y{}^2 + I \end{bmatrix} \begin{bmatrix} \ddot{\eta}_x \\ \ddot{\eta}_y \\ \ddot{\eta}_z \\ \ddot{\eta}_\psi \end{bmatrix} \dots \\ \dots &+ \begin{bmatrix} 0 & 0 & 0 & d_y m\sin(\psi)\dot{\eta}_\psi - d_x m\cos(\psi)\dot{\eta}_\psi \\ 0 & 0 & 0 & -d_y m\cos(\psi)\dot{\eta}_\psi - d_x m\sin(\psi)\dot{\eta}_\psi \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_z \\ \dot{\eta}_\psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ gm \\ 0 \end{bmatrix} \end{aligned} \tag{12}$$

To simplify the notation, this work compacts Equation (12), resulting in the following classical representation:

$$\mathbf{f_I}(t) = \bar{\mathbf{H}}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}(t) + \bar{\mathbf{C}}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}(t) + \bar{\mathbf{g}} \tag{13}$$

where $\bar{\mathbf{H}} \in \mathbb{R}^{4\times 4}$ is the mass and inertia matrix of the hexacopter system, in addition to being positive and symmetric definite; $\bar{\mathbf{C}} \in \mathbb{R}^{4\times 4}$ is the matrix of Coriolis forces and $\bar{\mathbf{g}} \in \mathbb{R}^4$ is known as the gravitational vector. Vectors $\ddot{\boldsymbol{\eta}} = \begin{bmatrix} \ddot{\eta}_x & \ddot{\eta}_y & \ddot{\eta}_z & \ddot{\eta}_\psi \end{bmatrix}$ and $\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\eta}_x & \dot{\eta}_y & \dot{\eta}_z & \dot{\eta}_\psi \end{bmatrix}$ are the acceleration and velocity of the point of interest in the inertia frame $< \mathbb{I} >$.

Due to the fact that the hexacopter platform can be controlled through reference maneuverability velocity commands, this work converts the generalized force and torque inputs of the dynamic model (13) into reference maneuverability velocities as system control inputs. Considering the position dynamic model presented in [54] which considers all DOF in a multirotor:

$$\begin{aligned} \begin{bmatrix} \dot{\mu}_l \\ \dot{\mu}_m \\ \dot{\mu}_n \end{bmatrix} = \begin{bmatrix} 0 & w_z & -w_y \\ -w_z & 0 & w_x \\ w_y & -w_x & 0 \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \end{bmatrix} - \frac{f_{z_I}}{m} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \dots \\ g \begin{bmatrix} \cos(\psi)\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin(\psi)\cos\theta & \cos\psi\cos\theta + \sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \tag{14}$$

it can be written in a compact form as $\dot{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{w} \end{bmatrix}_x \boldsymbol{\mu} + g\mathbf{R}(\theta, \phi, \psi)^T \mathbf{e_3} - \frac{f_z}{m}\mathbf{e_3}$, where $\mathbf{w} = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}$ are the angular velocities associated with the frame $< \mathbb{B} >$, $\begin{bmatrix} \mathbf{w} \end{bmatrix}_x$ represents skew symmetric matrix and finally $\mathbf{R}(\theta, \phi, \psi)$ represents the rotation matrix from the frame $< \mathbb{B} >$ to $< \mathbb{I} >$. For any multicopter, drags applied to rotating blades are in the direction of the body axes; from (14), the position dynamic model considering the aerodynamic drag model is represented as:

$$\begin{bmatrix}\dot{\mu_l}\\ \dot{\mu_m}\end{bmatrix} = \begin{bmatrix}\mu_m\omega_z - \mu_n\omega_y - g\sin\theta - \frac{k_{drag}}{m}\mu_l \\ \mu_n\omega_x - \mu_l\omega_z + g\cos\theta\sin\phi - \frac{k_{drag}}{m}\mu_m\end{bmatrix} \tag{15}$$

where $\dot{\mu_l}$, $\dot{\mu_m}$ are the accelerations in the body frame $< \mathbb{B} >$ and $k_{drag}$ is the drag constant. To include the drag coefficients in the dynamic model, this work split Equation (14) and can be written as:

$$\begin{bmatrix}a_x\\ a_y\end{bmatrix} = \begin{bmatrix}\mu_m\omega_z - \mu_n\omega_y - \frac{k_{drag}}{m}\mu_l \\ \mu_n\omega_x - \mu_l\omega_z - \frac{k_{drag}}{m}\mu_m\end{bmatrix} \tag{16}$$

where $a_x = \dot{\mu_l} + g\sin(\theta)$ and $a_y = \dot{\mu_m} - g\cos(\theta)\sin(\phi)$. Since the low-level controller guarantees hover, the cross terms are ignored, so the specific force satisfies:

$$\begin{bmatrix}f_x\\ f_y\end{bmatrix} = \begin{bmatrix}-k_{drag}\mu_l \\ -k_{drag}\mu_m\end{bmatrix} \tag{17}$$

where the forces produced in the system $\mathbf{f}_{hB} = \begin{bmatrix}f_x & f_y\end{bmatrix}$ are generate by a PD controller for the horizontal plane. The PD controller was designed as:

$$\mathbf{f}_{hB} = k_{pl}(\boldsymbol{\mu}_{h_{ref}} - \boldsymbol{\mu}_h) + k_{dl}(\dot{\boldsymbol{\mu}}_{h_{ref}} - \dot{\boldsymbol{\mu}}_h) \tag{18}$$

where $\boldsymbol{\mu}_h = \begin{bmatrix}\mu_l & \mu_m\end{bmatrix}^T \in \mathbb{R}^2$ is the vector of the horizontal maneuverability velocities, $\boldsymbol{\mu}_{h_{ref}} = \begin{bmatrix}\mu_{lref} & \mu_{mref}\end{bmatrix}^T \in \mathbb{R}^2$ is the reference desired velocity and $k_{pl}$, $k_{dl}$ are positive definite matrices. For the PD controller, this work used the following assumptions if $\dot{\boldsymbol{\mu}}_{h_{ref}} = 0$ then $\lim_{t\to\infty} \|\mathbf{f}_h\| = 0$ and $\lim_{t\to\infty} \|\tilde{\boldsymbol{\mu}}_h\| = 0$, respectively.

A similar concept was applied to the vertical force attached to the body frame $f_{z_B} = f_{ref} + mg$. Designing a PD controller for the vertical thrust as:

$$f_{z_B} = k_{pn}(\mu_{n_{ref}} - \mu_n) + k_{dn}(\dot{\mu}_{n_{ref}} - \dot{\mu}_n) + mg \tag{19}$$

where $k_{pn}$, $k_{dn}$ are positive definite matrices, this work considers the following assumptions if $\dot{\mu}_{n_{ref}} = 0$ then the $\lim_{t\to\infty} \|f_{z_I}\| = 0$ and $\lim_{t\to\infty} \|\tilde{\mu}_n\| = 0$, respectively.

The same approach can be extended to the attitude dynamic model presented in [54], which is defined as:

$$\mathbf{J_B} \cdot \dot{\mathbf{w}} = -\mathbf{w} \times (\mathbf{J_B} \cdot \mathbf{w}) + \mathbf{G_B} + \boldsymbol{\tau} \tag{20}$$

where $\boldsymbol{\tau} \triangleq \begin{bmatrix}\tau_x & \tau_y & \tau_\psi\end{bmatrix} \in \mathbb{R}^3$ is the vector of moments in the body axes $< \mathbb{B} >$; $\mathbf{J_B} \in \mathbb{R}^{3\times3}$ includes the moment of inertia and $\mathbf{G_B} \triangleq \begin{bmatrix}G_{B,\phi} & G_{B,\theta} & G_{B,\psi}\end{bmatrix}^T \in \mathbb{R}^3$ represents the gyroscopic torques. Ignoring the term $-\mathbf{w} \times (\mathbf{J_B} \cdot \mathbf{w}) + \mathbf{G_B}$, the simplified attitude dynamic model is defined as: $\mathbf{J_B}\dot{\mathbf{w}} = \tau_\psi$, where a PD controller for angular translations is defined as:

$$\tau_\psi = k_{p\omega}(\omega_{ref} - \omega) + k_{d\omega}(\dot{\omega}_{ref} - \dot{\omega}) \tag{21}$$

where $k_{p\omega}, k_{d\omega} \in \mathbb{R}$ are positive scalars and the following assumptions are true if $\dot{\omega}_{ref} = 0$; then, $\lim_{t\to\infty} \|\tau_\psi\| = 0$ and $\lim_{t\to\infty} \|\tilde{\omega}\| = 0$, respectively.

After all the considerations presented beforehand, this work introduces the general structure of low-level PD controllers combining Equations (18), (19) and (21), finally the structure is presented in Equation (22).

$$\begin{bmatrix}f_{x_B}\\ f_{y_B}\\ f_{z_B}\\ \tau_{\psi_B}\end{bmatrix} = \begin{bmatrix}k_{pl} & 0 & 0 & 0\\ 0 & k_{pl} & 0 & 0\\ 0 & 0 & k_{pn} & 0\\ 0 & 0 & 0 & k_{p\omega}\end{bmatrix}\begin{bmatrix}\mu_{lref}\\ \mu_{mref}\\ \mu_{nref}\\ \omega_{ref}\end{bmatrix} - \begin{bmatrix}k_{pl} & 0 & 0 & 0\\ 0 & k_{pl} & 0 & 0\\ 0 & 0 & k_{pn} & 0\\ 0 & 0 & 0 & k_{p\omega}\end{bmatrix}\begin{bmatrix}\mu_l\\ \mu_m\\ \mu_n\\ \omega\end{bmatrix} - \begin{bmatrix}k_{dl} & 0 & 0 & 0\\ 0 & k_{dl} & 0 & 0\\ 0 & 0 & k_{dn} & 0\\ 0 & 0 & 0 & k_{d\omega}\end{bmatrix}\begin{bmatrix}\dot{\mu}_l\\ \dot{\mu}_m\\ \dot{\mu}_n\\ \dot{\omega}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ mg\\ 0\end{bmatrix} \tag{22}$$

Equation (22) can be written in a compact form as $\mathbf{f_B}(t) = \mathbf{K_p}\boldsymbol{\mu}_{\mathbf{ref}}(t) - \mathbf{K_p}\boldsymbol{\mu}(t) - \mathbf{K_d}\dot{\boldsymbol{\mu}}(t) + \mathbf{g}$, where the values $k_{pl}$, $k_{pm}$ and $k_{p\omega}$ are the proportional values of the PD controller; $k_{dl}$, $k_{dm}$ and $k_{d\omega}$ are the derivative gains; $\boldsymbol{\mu}_{\mathbf{ref}} = \begin{bmatrix} \mu_{lref} & \mu_{mref} & \mu_{nref} & \omega_{ref} \end{bmatrix}^T \in \mathbb{R}^4$ are the reference maneuverability velocities or control vector of the hexacopter; $\boldsymbol{\mu}$ and $\dot{\boldsymbol{\mu}}$ are the real velocities and accelerations generated in the system.

Finally, with all the transformations considered beforehand, Equations (4) and (6) are substituted into a dynamic hexarotor model (12), and equating the expression to (22), a new reduced dynamic model can written as:

$$\begin{bmatrix} \mu_{lref} \\ \mu_{mref} \\ \mu_{nref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} \zeta_1 & 0 & 0 & \zeta_2 \\ 0 & \zeta_3 & 0 & \zeta_4 \\ 0 & 0 & \zeta_5 & 0 \\ b\zeta_6 & a\zeta_7 & 0 & \zeta_8(a^2+b^2)+\zeta_9 \end{bmatrix} \begin{bmatrix} \dot{\mu}_l \\ \dot{\mu}_m \\ \dot{\mu}_n \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \zeta_{10} & \omega\zeta_{11} & 0 & a\omega\zeta_{12} \\ \omega\zeta_{13} & \zeta_{14} & 0 & b\omega\zeta_{15} \\ 0 & 0 & \zeta_{16} & 0 \\ a\omega\zeta_{17} & b\omega\zeta_{18} & 0 & \zeta_{19} \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \tag{23}$$

This new formulation represents the evolution of the hexacopter's general velocities with the reference maneuverability velocities as the input of the system. This representation is useful because commercial aerial platforms can be directly controlled with maneuverability velocities, ignoring the low-level control. Equation (23) can be compactly written as:

$$\boldsymbol{\mu}_{\mathbf{ref}}(t) = \mathbf{H}(\boldsymbol{\zeta}, a, b)\dot{\boldsymbol{\mu}}(t) + \mathbf{C}(\boldsymbol{\zeta}, \boldsymbol{\mu})\boldsymbol{\mu}(t) \tag{24}$$

where $\mathbf{H} = (\mathbf{RK_p})^{-1}((\bar{\mathbf{H}}\mathbf{J} + \mathbf{RK_d})$, $\mathbf{C} = (\mathbf{RK_p})^{-1}(\bar{\mathbf{H}}\dot{\mathbf{J}} + \mathbf{RK_p} + \bar{\mathbf{C}}\mathbf{J})$ , $\mathbf{G} = (\mathbf{RK_p})^{-1}(\mathbf{g} - \mathbf{R}\bar{\mathbf{G}})) = 0$. Thus, $\mathbf{H}(\zeta, a, b) \in \mathbb{R}^{4\times4}$ is a positive definite matrix, which is the new mass and inertia matrix of the hexacopter robot, the new Coriolis and Centripetal matrix is defined by $\mathbf{C}(\boldsymbol{\zeta}, \boldsymbol{\mu}) \in \mathbb{R}^{4\times4}$, $\mathbf{R}$ represents the rotation matrix under the z axis from frame $< \mathbb{B} >$ to frame $< \mathbb{I} >$. The vector of dynamic parameters represents the combination of all the internal values in the hexacopter robot such as the physical, mechanical, electrical and PD values, and the vector is defined by $\boldsymbol{\zeta} = \begin{bmatrix} \zeta_1 & \zeta_2 & \dots & \zeta_{l_d} \end{bmatrix} \in \mathbb{R}^{l_d}$ where $l_d = 19$ is the number of variables that this work needs to identify as the details of each parameters are presented in Appendix A for Equations (A1) and (A2).

### Euler–Lagrange Identification and Validation

This section presents the identification and validation of the dynamic parameters, as this work uses real experimental data to estimate these values in order to use the model in the proposed controller. This work transforms the mathematical model (24) in the linear parametric representation into:

$$\boldsymbol{\mu}_{\mathbf{ref}}(t) = \boldsymbol{\Theta}(\dot{\boldsymbol{\mu}}(t), \boldsymbol{\mu}(t))\boldsymbol{\zeta} \tag{25}$$

where the matrix $\boldsymbol{\Theta}$ confirms the values of velocities and accelerations obtained from real-world experiments. In order to estimate the vector of dynamic values, this work uses snapshot measurements over the time as $l \in [t, t + t_f]$ where $l$ is an instant of measure and $t_f$ is the final time of the experimental information. The identification process was developed through optimization techniques defining $\ell_m(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t), \boldsymbol{\zeta}) : \mathbb{R}^4 \to \mathbb{R}_{\geq 0}$, which is a positive semi-definite function confirmed by:

$$\ell_m(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t), \boldsymbol{\zeta}) = \frac{1}{2}\overbrace{||\boldsymbol{\mu}_{\mathbf{ref}}(l|t) - \boldsymbol{\Theta}(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t))\boldsymbol{\zeta}||^2_{\mathbf{Q}_m}}^{Model\ identification\ cost} \tag{26}$$

*Model identification cost:* this function was formulated using the subtraction between the reference maneuverability $\boldsymbol{\mu}_{\mathbf{ref}}(l|t)$ velocities applied in the hexacopter and the system measurements defined by the matrix $\boldsymbol{\Theta}(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t))$; the operator $|| \cdot ||$ is known as the Euclidean norm; and $\mathbf{Q}_m \in \mathbb{R}^{4\times4}_{>0}$ is a positive definite weighting matrix.

With Equation (26), it is possible to generate the functional cost or performance index over the experimental data, which is formulated as follows:

$$V_m(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t), \boldsymbol{\zeta}) = \int_t^{t+t_f} \ell_m(\dot{\boldsymbol{\mu}}(\tau|t), \boldsymbol{\mu}(\tau|t), \boldsymbol{\mu}_{\mathbf{ref}}(\tau|t), \boldsymbol{\zeta})d\tau \tag{27}$$

Considering the performance index function presented in (27), the optimization structures are defined in (28), which was solved using the sequential quadratic programming (SQP) technique.

$$\mathcal{P}_m : P_m(\dot{\boldsymbol{\mu}}(t), \boldsymbol{\mu}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) = \min_{\boldsymbol{\zeta}} \quad V_m(\dot{\boldsymbol{\mu}}(l|t), \boldsymbol{\mu}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t), \boldsymbol{\zeta}) \tag{28}$$

This work used the experimental information of the hexacopter platform DJI MATRICE 600 PRO to identify the dynamic parameters. The aerial system has a low-level flight controller called A3, which is confirmed by three global positioning systems (GPS) and three inertial measurement unit systems (IMUs). Furthermore, this system provides highly precise measurements from the system states. The identified parameters are presented in Table 1; furthermore, for the identification and validation process, this work used a different kind of reference velocity signals.

**Table 1.** Dynamic parameters of the hexacopter aerial platform.

| System | Dynamic Parameters | | |
|---|---|---|---|
| DJI MATRICE 600 PRO | $\zeta_1 = 2.11$ | $\zeta_2 = -0.005$ | $\zeta_3 = 1.8$ |
| | $\zeta_4 = 3.17$ | $\zeta_5 = 1.78$ | $\zeta_6 = 0.39$ |
| | $\zeta_7 = -0.003$ | $\zeta_8 = -0.03$ | $\zeta_9 = 0.006$ |
| | $\zeta_{10} = 0.02$ | $\zeta_{11} = 0.002$ | $\zeta_{12} = 0.06$ |
| | $\zeta_{13} = 0.70$ | $\zeta_{14} = 0.02$ | $\zeta_{15} = -0.05$ |
| | $\zeta_{16} = -0.01$ | $\zeta_{17} = -0.005$ | $\zeta_{18} = -0.01$ |
| | $\zeta_{19} = 0.831$ | | |

Figure 2 shows the results of the identification process, where the signals $\mu_l$, $\mu_m$, $\mu_n$ and $\omega$ were obtained by real-world experiments; and $\mu_{lm}$, $\mu_{mm}$, $\mu_{nm}$ and $\omega_m$ are the estimated values of the dynamic model using the optimization formulation presented previously. The presence of noise in the signals is an important factor due to the fact that it probably causes problems in the identification process; this work uses a filter defined as: $\lambda/(s+\lambda)$ with $\lambda = 1$ to eliminate noise in the real measurements. The filter was applied to the system measurements and the reference maneuverability velocities.

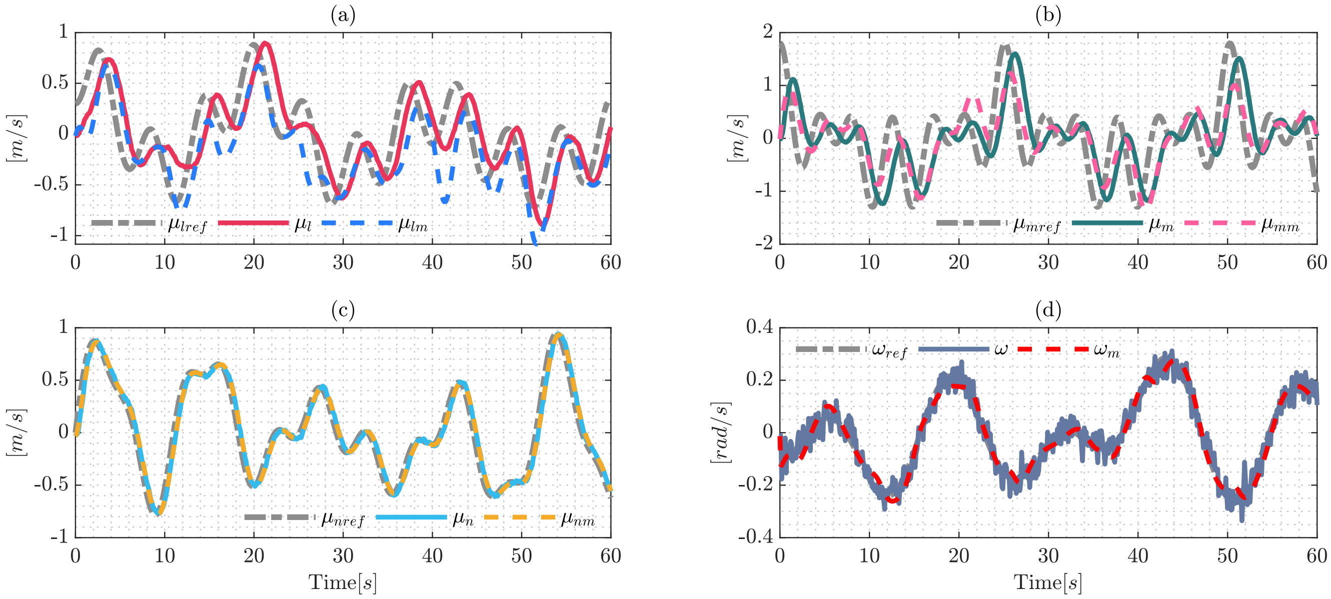

**Figure 2.** Signals of the identification process using the Euler-Lagrange formulation with the optimization techniques (SQPs). The results are confirmed by: (**a**) identification of the reference velocity $\mu_{lref}$; (**b**) shows the results over the reference velocity $\mu_{mref}$; (**c**) identification over the upper longitudinal velocity $\mu_{nref}$; and (**d**) shows the results of the reference angular velocity $\omega_{ref}$.

The validation process of the dynamic parameters is a very important factor; hence, the model needs to estimate the behavior of the system with different signals in order to represent the real dynamics of the system. The validation signals are shown in Figure 3, the validation process shows the performance of the estimate dynamic model, which will be useful for the comparative section.

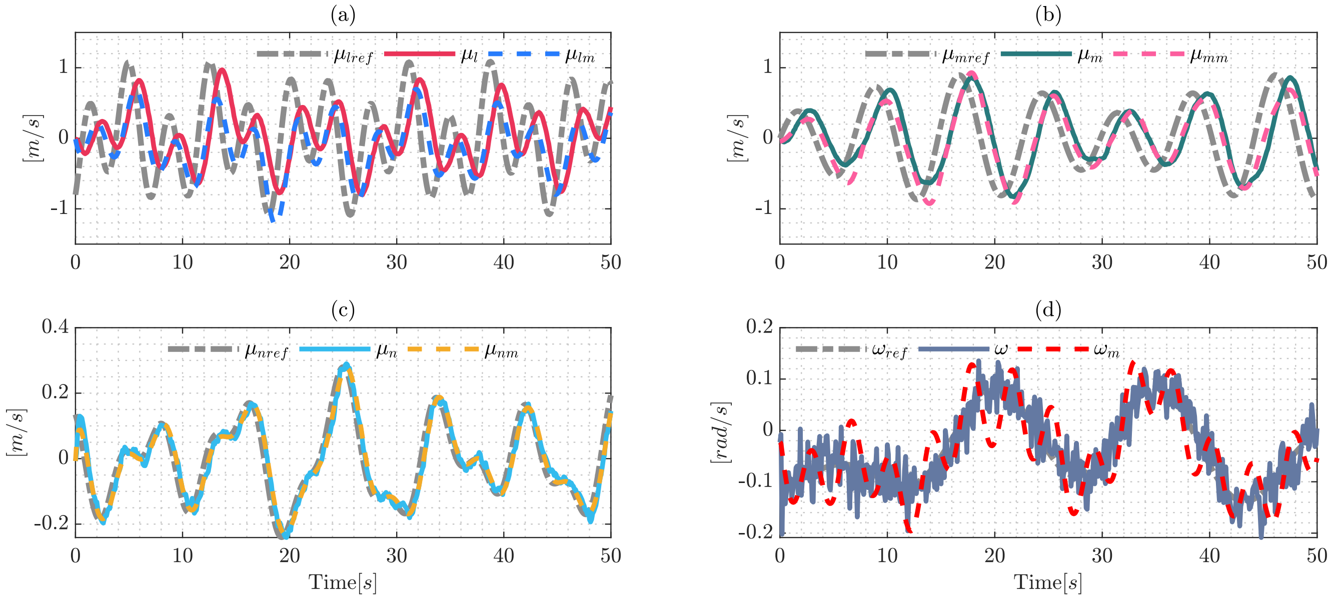

**Figure 3.** Validation process signals using the Euler–Lagrange formulation with the optimization technique (SQP). The results are confirmed by: (**a**) validation over the reference velocity $\mu_{lref}$; (**b**) shows the results over the reference velocity $\mu_{mref}$; (**c**) shows the validation over the longitudinal velocity $\mu_{nref}$; and (**d**) shows the validation results of the reference angular velocity $\omega_{ref}$.

With all the previously presented results and the validation of the dynamic parameters, this work formulates the identified model using Euler-Lagrange as a vector-function presented as:

$$\dot{\boldsymbol{\mu}}(t) = \mathbf{f}_{SQP}(\boldsymbol{\zeta}, \boldsymbol{\mu}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \tag{29}$$

Dynamic Mode Decomposition Formulation

This section presents the second approach of the dynamic model formulation which, due to the hexacopter platform, only has internal low-level PIDs; and the system tries to stay close to the hover-position with small angular variations in roll and pitch angles $(\phi, \theta)$. Furthermore, this kind of hexacopter platforms cannot be controlled with the torques commands, and the control inputs at the system are the reference maneuverability velocities $\boldsymbol{\mu}_{\mathbf{ref}}$. Considering the small angle assumptions and the inputs in the system, the model can hence be formulated with a linear approximation for time-varying systems resulting in:

$$\dot{\boldsymbol{\mu}}(t) = \mathbf{A}\boldsymbol{\mu}(t) + \mathbf{B}\boldsymbol{\mu}_{\mathbf{ref}}(t) \tag{30}$$

where $\boldsymbol{\mu}$ and $\dot{\boldsymbol{\mu}}$ are the vectors of the maneuverability velocities and accelerations; $\boldsymbol{\mu}_{\mathbf{ref}}$ is the reference maneuverability velocity vector also known as the control vector; matrices $\mathbf{A} \in \mathbb{R}^{4\times4}$ and $\mathbf{B} \in \mathbb{R}^{4\times4}$ represent the unforced system and the contribution of the control vector, respectively.

One of the emerging techniques to identify systems by experimental information is that of dynamic mode decomposition (DMD) [55–57], where the objective is to approximate the matrices $\mathbf{A}$ and $\mathbf{B}$. The DMD algorithm proposes the construction of snapshot measurements $s$ and the formation of augmented matrices, resulting in a new formulation of the system as follows:

$$\dot{\boldsymbol{\chi}} \approx \mathbf{A}\boldsymbol{\chi} + \mathbf{B}\boldsymbol{\Gamma} \tag{31}$$

where the new matrices are defined below:

$$\dot{\chi} = \begin{bmatrix} | & | & & | \\ \dot{\mu}_1 & \dot{\mu}_2 & \dots & \dot{\mu}_s \\ | & | & & | \end{bmatrix}$$

$$\chi = \begin{bmatrix} | & | & & | \\ \mu_1 & \mu_2 & \dots & \mu_s \\ | & | & & | \end{bmatrix}$$

$$\mathbf{\Gamma} = \begin{bmatrix} | & | & & | \\ \mu_{\mathbf{ref}_1} & \mu_{\mathbf{ref}_2} & \dots & \mu_{\mathbf{ref}_s} \\ | & | & & | \end{bmatrix}$$

$\dot{\chi}$ and $\chi$ are the augmented matrices confirmed by the maneuverability accelerations and velocities, respectively; and $\mathbf{\Gamma}$ is the matrix of the reference velocities applied in the system over the $s$ number of snapshots.

For estimation purposes, the system can be formulated as:

$$\dot{\chi} \approx \mathbf{\Phi\Omega} \tag{32}$$

where $\mathbf{\Phi} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \in \mathbb{R}^{4\times(4+4)}$ is the matrix confirmed by unknown values of $\mathbf{A}$ and $\mathbf{B}$; and $\mathbf{\Omega} = \begin{bmatrix} \chi & \mathbf{\Gamma} \end{bmatrix}^T \in \mathbb{R}^{(4+4)\times s}$ is the data matrix obtained from the system. This work finds the best values of the unknown matrix using the Frobenius norm $||\dot{\chi} - \mathbf{\Phi\Omega}||_F$, defining the values in the following equation:

$$\mathbf{\Phi} \approx \dot{\chi}\mathbf{\Omega}^\dagger \tag{33}$$

One of the best methods to find the solution of the Frobenius norm is the singular value decomposition (SVD); therefore, applying this method to the augmented data matrix resulting as $\mathbf{\Omega} \approx \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^T$, where $\bar{\mathbf{U}}$, $\bar{\mathbf{\Sigma}}$ and $\bar{\mathbf{V}}$ are matrices with a truncation value $\bar{r}$, with these considerations presented above, the estimation is defined by the following equation:

$$\mathbf{\Phi} \approx \dot{\chi}\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{U}}^T \tag{34}$$

To approximate the values of the matrices $\mathbf{A}$ and $\mathbf{B}$, the Equation (34) can be written as (35), considering the splitting values of the singular vector $\bar{\mathbf{U}}$ .

$$\begin{bmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}} \end{bmatrix} \approx \begin{bmatrix} \dot{\chi}\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{U}}_1^T & \dot{\chi}\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{U}}_2^T \end{bmatrix} \tag{35}$$

The values of the matrix $\mathbf{A}$ can be approximated using the split matrix $\bar{\mathbf{U}}_1 \in \mathbb{R}^{4\times\bar{r}}$ and the values of $\mathbf{B}$ are approximated using the values of $\bar{\mathbf{U}}_2 \in \mathbb{R}^{4\times\bar{r}}$.

### Dynamic Mode Decomposition Identification and Validation

This section presents the identification and validation process of the dynamic model using the DMD formulation. The values of matrices $\dot{\chi}$, $\chi$ and $\mathbf{\Gamma}$ were constructed by measure snapshots information of the aerial platform DJI MATRICE 600 PRO and the application of Algorithm 1, which represents all the necessary steps to identify the values of unknown matrices.

The approximation values of the unknown matrix are presented in Tables 2 and 3, where, $a_{ij}$ and $b_{ij}$ are the individual elements of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$, respectively.

The signals used for the identification process are presented in Figure 4, which are the same as those used for the identification process employing Euler–Lagrange formulation, due to the fact that both techniques need to estimate the values under the same conditions.

**Algorithm 1** Identification process Using DMD

```
**Input:** DJI MATRICE 600 PRO measurements $(\dot{\chi}, \chi, \mathbf{\Gamma})$ and truncation value $\bar{r}$.
**Output:** Matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$.
    **Ident** $(\dot{\chi}, \chi, \mathbf{\Gamma}, \bar{r})$ **do**
        $\mathbf{\Omega} \leftarrow \begin{bmatrix} \chi & \mathbf{Y} \end{bmatrix}^T$
        $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow SVD(\mathbf{\Omega})$
        $(\bar{\mathbf{U}}, \bar{\mathbf{\Sigma}}, \bar{\mathbf{V}}) \leftarrow Trunc(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}, \bar{r})$
        $(\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2) \leftarrow Split(\bar{\mathbf{U}})$
        $\bar{\mathbf{A}} \leftarrow \dot{\chi}\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{U}}_1^T$
        $\bar{\mathbf{B}} \leftarrow \dot{\chi}\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{U}}_2^T$
    **end Ident**
    **return** $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$
```

**Table 2.** Approximation values of unforced matrix.

| Matrix | Approximated Values | | | |
|---|---|---|---|---|
| $\bar{\mathbf{A}}$ | $a_{11} = -0.5827$ | $a_{12} = -0.0721$ | $a_{13} = 0.0587$ | $a_{14} = -0.0280$ |
| | $a_{21} = 0.0214$ | $a_{22} = -0.4770$ | $a_{23} = 0.0731$ | $a_{24} = -0.1840$ |
| | $a_{31} = 0.0152$ | $a_{32} = 0.0102$ | $a_{33} = -3.5696$ | $a_{34} = 0.0064$ |
| | $a_{41} = 0.0377$ | $a_{42} = -0.0113$ | $a_{43} = -0.0058$ | $a_{44} = -8.2424$ |

**Table 3.** Approximate values of control actuation matrix.

| Matrix | Approximated Values | | | |
|---|---|---|---|---|
| $\bar{\mathbf{B}}$ | $b_{11} = 0.8048$ | $b_{12} = -0.0376$ | $b_{13} = -0.0793$ | $b_{14} = -0.1797$ |
| | $b_{21} = 0.0634$ | $b_{22} = 0.8527$ | $b_{23} = -0.0783$ | $b_{24} = 0.3821$ |
| | $b_{31} = -0.0202$ | $b_{32} = -0.0046$ | $b_{33} = 3.5776$ | $b_{34} = 0.0607$ |
| | $b_{41} = -0.0251$ | $b_{42} = -0.0029$ | $b_{43} = -0.0288$ | $b_{44} = 8.1294$ |

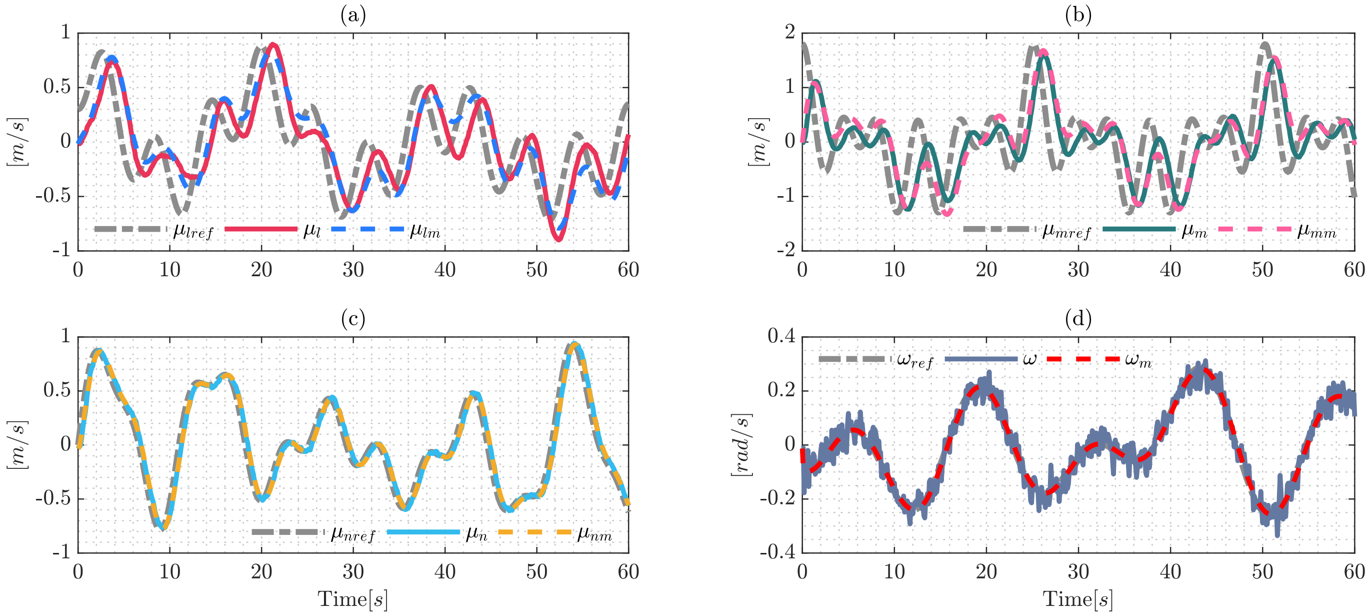

**Figure 4.** Identification process using dynamic mode decomposition formulation (DMD): (**a**) results over $\mu_{lref}$; (**b**) identification on $\mu_{mref}$; (**c**) validation over $\mu_{nref}$; and (**d**) shows the results on $\omega_{ref}$.

The estimated model displays good behavior due to the fact it replicates the signals obtained from experimental data. The noise in the angular maneuverability velocity $\omega$ is an important factor, however, dynamic mode decomposition and the robustness of the

Frobenius norm can deal with this noise and the application of filters is not necessary in the identification process.

In the same way presented in the validation process using the Lagrange formulation, this section presents the validation process under DMD formulation, the results of which are shown in Figure 5, which shows the good performance of the proposed technique.

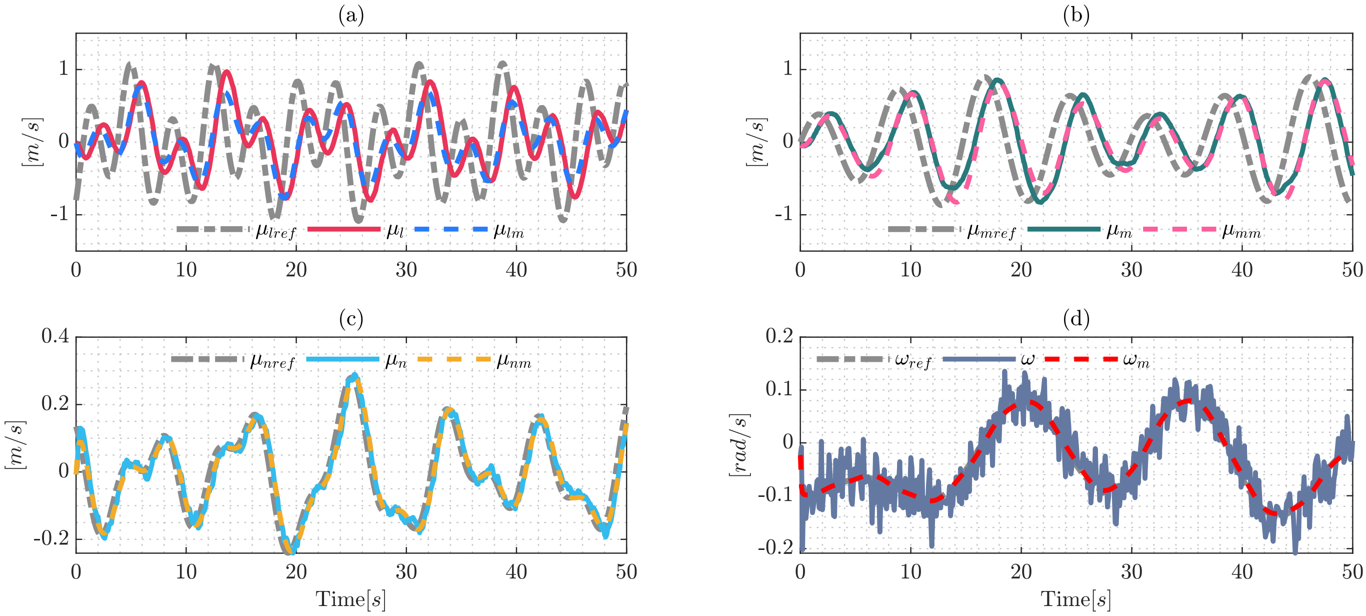

**Figure 5.** Validation process using dynamic mode decomposition (DMD). The results are: (**a**) validation over the reference velocity $\mu_{lref}$; (**b**) shows the results through velocity $\mu_{mref}$; (**c**) validation over the longitudinal velocity $\mu_{nref}$; and (**d**) shows the validation results of the reference angular velocity $\omega_{ref}$.

Finally, considering the approximation values of the unknown matrices, the dynamic model can be written in a vector-function form as:

$$\dot{\boldsymbol{\mu}}(t) = \mathbf{f}_{DMD}(\boldsymbol{\Gamma}, \boldsymbol{\mu}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \tag{36}$$

### 2.1.3. Accuracy Comparative Results

This section presents the comparative results between the Euler–Lagrange (29) and dynamic mode decomposition (36) formulations. This work uses the integral square error (ISE) to show the performance of each formulation and selects the best one for the nonlinear model predictive controller structure.

Figure 6 shows the ISE of each formulation, where the model for frontal maneuverability velocity $\mu_{lm}$ presents a considerable error using the Euler–Lagrange formulation—with a value of precisely 1.26—whilst on the other hand, DMD results show a better performance with a value of 0.56. The proposed formulations show similar results with respect to the lateral velocity $\mu_{mm}$, and the values using the Euler–Lagrange and DMD are 0.35 and 0.16, respectively. The results for the estimated upper maneuverability velocity $\mu_{nm}$ are really close with the following values 0.0104 and 0.0100. Finally, the ISE of the estimated models for the angular maneuverability velocity are 0.104 and 0.038 for the Euler–Lagrange and DMD formulations, respectively.

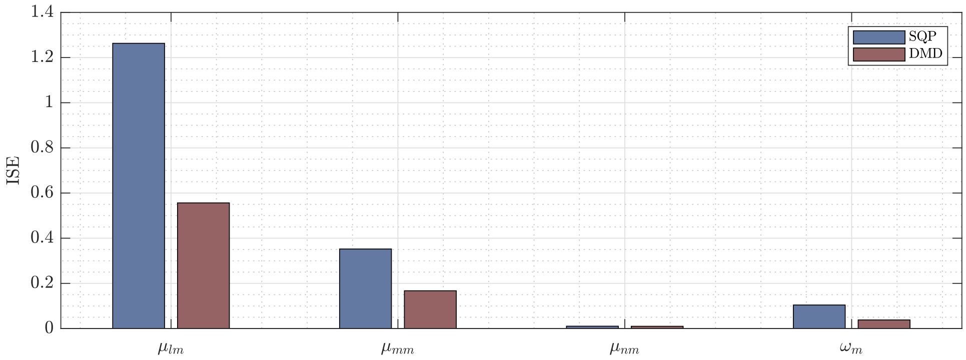

**Figure 6.** Integral square error of the identified models using Euler–Lagrange with sequential quadratic programming (SQP) and dynamic mode decomposition (DMD) formulation.

With the previously presented results, this work uses the DMD estimated model to construct the proposed controller. Combining Equations (5) and (36), the general mathematical representation is defined as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \tag{37}$$
$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \;:=\; \begin{bmatrix} \mathbf{f}_k(\psi(t), \boldsymbol{\mu}(t)) \\ \mathbf{f}_{DMD}(\boldsymbol{\Gamma}, \boldsymbol{\mu}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \end{bmatrix}$$

where $\mathbf{x} = \begin{bmatrix}\boldsymbol{\eta}^T & \boldsymbol{\mu}^T\end{bmatrix}^T \in \mathbb{R}^8$ is the generalized vector of the system states that is confirmed by $\boldsymbol{\eta}$, which is the pose of the point of interest in the hexacopter and $\boldsymbol{\mu}$ is the maneuverability velocities.

### 2.2. Control Methodology

This section presents the formulation of the proposed controller, where the main objective is the trajectory tracking over the desired reference trajectory using an hexacopter platform and due to the dynamic effects and the important size of the system, this work includes obstacle avoidance to improve the behavior of the system in dynamic environments. This work presents the nonlinear model predictive controller (NMPC); furthermore, the constraints included in the optimization problem were: generalized system dynamics, bounded maneuverability velocities, rate of change of control velocities and obstacles in the dynamic environment. The main objectives of the proposed controller are: to solve the control problem associated with the trajectory tracking subject to the system and the environment and system constraints. The control problem is solved by the scheme shown in Figure 7, where the structure uses the system information and possible obstacles in the environment to generate an optimal control policy under the system constraints.

#### 2.2.1. Nonlinear Model Predictive Control

In order to solve the NMPC proposed scheme, this work formulates the generalized dynamics (37) as a prediction, resulting in:

$$\dot{\mathbf{x}}(l|t) = \mathbf{f}(\mathbf{x}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t)) \tag{38}$$

where $l \in [t, t+T]$ is the instant value evolution between the initial time $t$ and the prediction horizon $T$. Due to the prediction behavior of this scheme, the NMPC formulation requires index performances over the prediction horizon $T$ that guarantees the solution of the control problem and the system constraints. The generation of the performance functions are presented below.

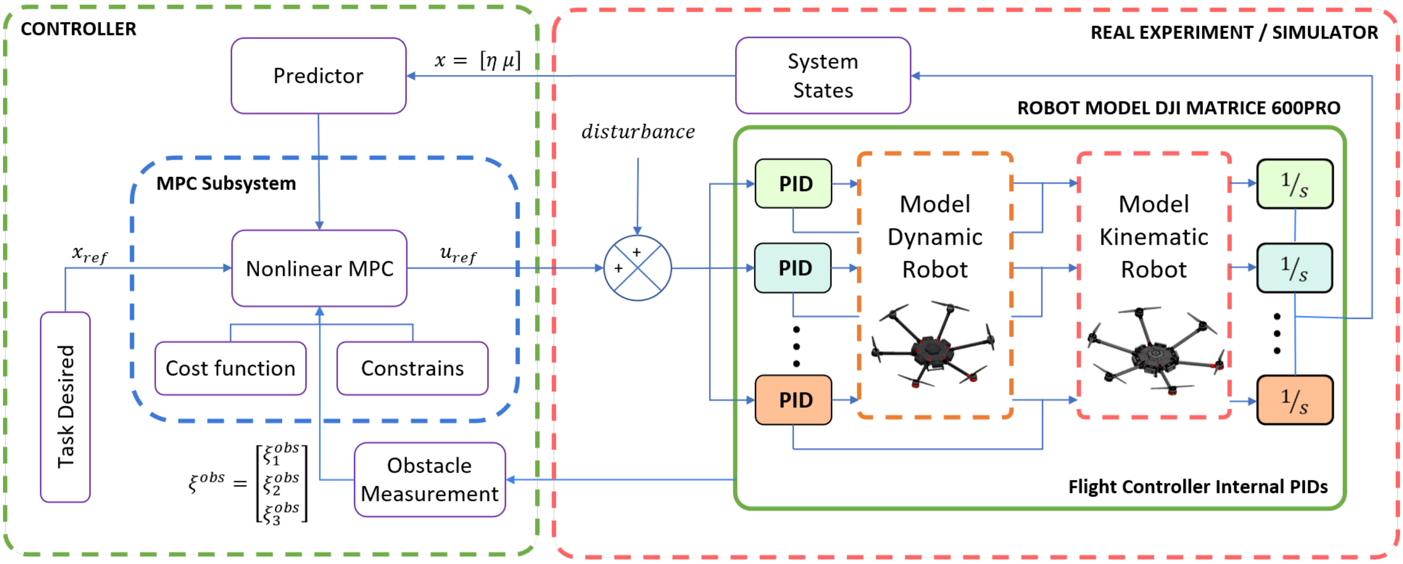

**Figure 7.** Proposed controller scheme.

### 2.2.2. Tracking Trajectory Formulation

This subsection presents the generation of the cost function over the evolution instants $l$ considering the following property: $\ell_t(\mathbf{x}(l|t), \boldsymbol{\eta}_{\mathbf{ref}}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t)) : \mathbb{R}^4 \times \mathbb{R}^4 \rightarrow \mathbb{R}_{\geq 0}$ to formulate the function as:

$$\ell_t(\mathbf{x}(l|t), \boldsymbol{\eta}_{\mathbf{ref}}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t)) = \overbrace{\frac{1}{2}||\boldsymbol{\eta}_{\mathbf{ref}}(l|t) - \mathbf{W_t x}(l|t)||^2_{\mathbf{Q}_t}}^{Trajectory\ stage\ cost} + \underbrace{\frac{1}{2}||\boldsymbol{\mu}_{\mathbf{ref}}(l|t)||^2_{\mathbf{Q}_u}}_{Control\ stage\ cost} \tag{39}$$

*Trajectory Stage Cost:* this part of the function is structured by the control error between the desired trajectory $\boldsymbol{\eta}_{\mathbf{ref}} = \begin{bmatrix} \eta_{x_{ref}} & \eta_{y_{ref}} & \eta_{z_{ref}} & \eta_{\psi_{ref}} \end{bmatrix}^T \in \mathbb{R}^4$ and the measurements of the system $\mathbf{x}$; the constant matrix $\mathbf{W_t} \in \mathbb{R}^{4\times 8}$ produces linear mapping between all the states of the system and the outputs $\eta_x$, $\eta_y$, $\eta_z$ and $\eta_\psi$; the Euclidean norm is defined by the operator $||\cdot||$; finally, $\mathbf{Q}_t \in \mathbb{R}^{4\times 4}_{>0}$ is a constant positive definite matrix.

*Control stage cost:* this part of the cost function defines the variations between the maneuverability control velocities $\boldsymbol{\mu}_{\mathbf{ref}}(l|t))$ considering that $\mathbf{Q}_u \in \mathbb{R}^{4\times 4}_{>0}$ is a positive definite weighting matrix.

The cost function at the last step of the prediction horizon $T$ was formulated with following property: $\ell_T(\mathbf{x}(t+T|t), \boldsymbol{\eta}_{\mathbf{ref}}(t+T|t)) : \mathbb{R}^4 \rightarrow \mathbb{R}_{\geq 0}$; with this consideration, the cost function is defined as:

$$\ell_T(\mathbf{x}(t+T|t), \boldsymbol{\eta}_{\mathbf{ref}}(t+T|t)) = \overbrace{\frac{1}{2}||\boldsymbol{\eta}_{\mathbf{ref}}(t+T|t) - \mathbf{W}\mathbf{x}(t+T|t)||^2_{\mathbf{Q}_T}}^{Terminal\ Trajectory\ cost} \tag{40}$$

*Terminal trajectory cost:* this term considers the control error between the reference trajectory at the last step of the prediction horizon $T$, where $\mathbf{Q}_T \in \mathbb{R}^{4\times 4}_{>0}$ is a positive definite weighting matrix.

This work uses Equations (39) and (40) to generate the performance index over the prediction horizon, which was developed in the continuous time formulation, where $\tau$ is the interval analyzed in the integral operation, and the performance index is defined as:

$$V(\mathbf{x}(l|t), \boldsymbol{\eta}_{\mathbf{ref}}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t)) = \int_t^{t+T} \ell_t(\mathbf{x}(\tau|t), \boldsymbol{\eta}_{\mathbf{ref}}(\tau|t), \boldsymbol{\mu}_{\mathbf{ref}}(\tau|t))d\tau + \ell_T(\mathbf{x}(t+T|t), \boldsymbol{\eta}_{\mathbf{ref}}(t+T|t)) \tag{41}$$

The performance index function (41) guarantees the trajectory tracking propriety of the controller. The tuning process of the positive definite matrices were developed in the simulations in order to prevent accidents in real-world experiments.

### 2.2.3. Maneuverability Velocities Constraints

Due to the hexacopter platform having maximum maneuverability velocities, this work proposes control input constraints on the NMPC structure, in addition to further constraints on the successive difference of the maneuverability velocities which were included to prevent aggressive behaviors. The input constraints are defined in (42) in order to maintain the control signals generated by the proposed structure under the limits of the hexacopter platform

$$\mathcal{U} = \{\boldsymbol{\mu}_{\mathbf{ref}} \in \mathcal{R}^4 : \boldsymbol{\mu}_{\mathbf{ref}_{min}} \leq \boldsymbol{\mu}_{\mathbf{ref}} \leq \boldsymbol{\mu}_{\mathbf{ref}_{max}}; |\boldsymbol{\mu}_{\mathbf{ref}}(j+1|t) - \boldsymbol{\mu}_{\mathbf{ref}}(j|t)| \leq \Delta\boldsymbol{\mu}_{\mathbf{ref}_{max}}\} \tag{42}$$

where $\boldsymbol{\mu}_{\mathbf{ref}_{min}} \in \mathbb{R}^4$ and $\boldsymbol{\mu}_{\mathbf{ref}_{max}} \in \mathbb{R}^4$ are the vectors of lower and upper limits in maneuverability velocities; to the successive difference, this works considers $j \in [t, t + (T-1)]$ and $\Delta\boldsymbol{\mu}_{\mathbf{ref}_{max}}$ as the vector of the maximum rate of change between the maneuverability velocities.

### 2.2.4. Obstacle Constraints

This section presents the obstacle constraints in order to track the desired trajectory and avoid obstacles in the dynamic environments. The obstacles can be defined as $\boldsymbol{\xi}^{obs}(i|z) = \begin{bmatrix} \eta_x^{obs} & \eta_y^{obs} & \eta_z^{obs} \end{bmatrix}^T \in \mathbb{R}^3$ like a position vector with respect to the inertia frame $< \mathbb{I} >$, where $i \in [z, z+Z]$ considering the analyzed obstacle as $z$ and $Z$ the number of obstacles detected by the system, the representation of the obstacles is shown in Figure 8.

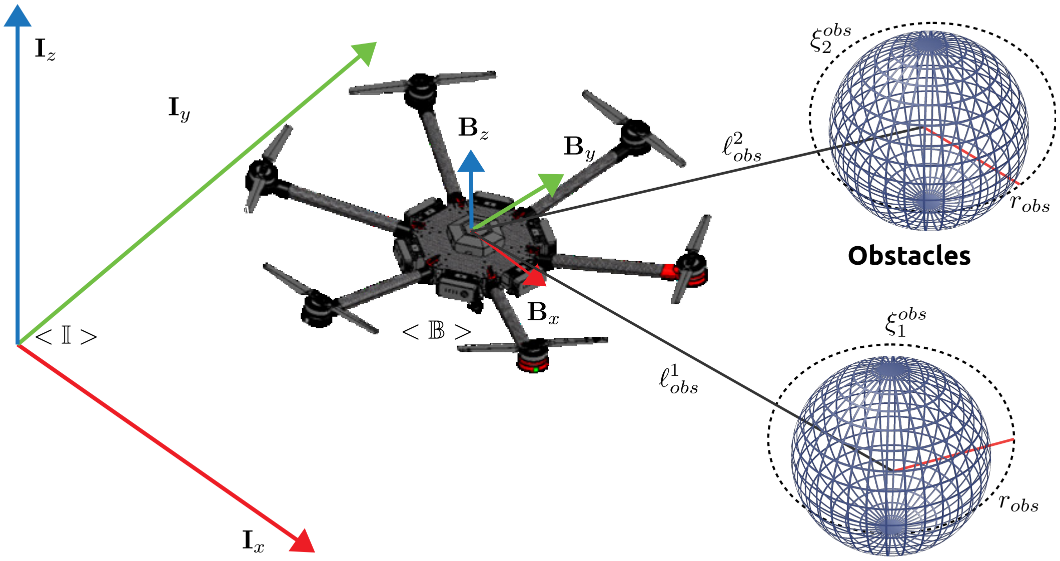

**Figure 8.** Representation of the hexacopter and obstacles.

The function of the distance between obstacles and the hexacopter has the following property: $\ell_{obs}(\mathbf{x}(l|t), \boldsymbol{\xi}^{obs}(i|z)) : \mathbb{R}^4 \to \mathbb{R}_{\geq 0}$, and the equation is formulated as:

$$\ell_{obs}(\mathbf{x}(l|t), \boldsymbol{\xi}^{obs}(i|z)) = \overbrace{||\boldsymbol{\xi}^{obs}(i|z) - \mathbf{W_o}\mathbf{x}(l|t)||^2}^{Distance\ to\ Obstacles} \tag{43}$$

where $\mathbf{W_o} \in \mathbb{R}^{3x8}$ is a constant matrix that produces linear mapping between all the states of the system and the outputs $\eta_x$, $\eta_y$ and $\eta_z$. With the previously presented considerations, this work formulates the system states constraints in (44), and where the safe region was defined as $r_{obs} \in \mathbb{R}_{>0}$, this value includes the safely region and the rotating blades' positions.

$$\mathcal{O} = \{\mathbf{x} \in \mathcal{R}^4 : r_{obs} - \ell_{obs}(\mathbf{x}(l|t), \boldsymbol{\xi}^{obs}(i|z)) \leq 0\} \tag{44}$$

### 2.2.5. Nonlinear Model Predictive Control Formulation

Considering the performance index presented before (41) and the system states (44) and control (42) constraints, this work can formulate the optimization problem as:

$$\mathcal{P} : P(\boldsymbol{\eta}_{\mathbf{ref}}(t)) = \min_{\boldsymbol{\mu}_{\mathbf{ref}}(l|t), \mathbf{x}(l|t)} \int_{t}^{t+T} \ell_t(\mathbf{x}(\tau|t), \boldsymbol{\eta}_{\mathbf{ref}}(\tau|t), \boldsymbol{\mu}_{\mathbf{ref}}(\tau|t))d\tau + \ell_T(\mathbf{x}(t+T|t), \boldsymbol{\eta}_{\mathbf{ref}}(t+T|t)) \tag{45a}$$

$$\text{subject to: } \dot{\mathbf{x}}(l|t) - \mathbf{f}(\mathbf{x}(l|t), \boldsymbol{\mu}_{\mathbf{ref}}(l|t)) = 0 \tag{45b}$$

with the following constrains:

$$\mathbf{x}(l|t) \in r_{obs} - \ell_{obs}(\mathbf{x}(l|t), \boldsymbol{\xi}^{obs}(i|z)) \leq 0 \tag{45c}$$

$$\boldsymbol{\mu}_{\mathbf{ref}}(l|t) \in \boldsymbol{\mu}_{\mathbf{ref}_{min}} \leq \boldsymbol{\mu}_{\mathbf{ref}} \leq \boldsymbol{\mu}_{\mathbf{ref}_{max}}; |\boldsymbol{\mu}_{\mathbf{ref}}(j+1|t) - \boldsymbol{\mu}_{\mathbf{ref}}(j|t)| \leq \Delta\boldsymbol{\mu}_{\mathbf{ref}_{max}} \tag{45d}$$

$\mathcal{P}$ is the function that guarantees the correct trajectory tracking and obstacle avoidance of the hexacopter system. The optimal control problem (45a) was converted into a nonlinear programming formulation (NLP) using the direct multiple shooting method, where the control maneuverability velocities $\boldsymbol{\mu}_{\mathbf{ref}}(l|t)$ and the system states $\mathbf{x}(l|t)$ are the decision variables of the optimization problem. A multiple shooting technique is more computationally efficient when it is compared with other discretization formulation, e.g., single shooting, more information in [58]. The system model was considered as the optimization constrain defined by (45b), the system and control constraints formulated in (45c) and (45d), respectively. With these considerations, this work minimizes the nonlinear programming problem using CasADI as a nonlinear optimization framework [59].

## 3. Results and Discussions

This section presents the simulations and experimental results to validate the proposed controller for the hexacopter platform DJI MATRICE 600 PRO. The experiments were performed considering: *(i) Real-world experiment:* the results presented in this section were developed using the nonlinear model predictive controller and environment obstacles are not considered in this experiments due to the hexacopter platform not having the required sensor for real-world measurements. *(ii) Simulation experiment:* simulation experiments show the effectiveness of the proposed controller in highly dynamic simulation environments with simulations of the dynamic behavior of the hexacopter in order to make it more realistic and prove the scalability of the controller.

### *3.1. Real-World Experiments*

Several experiments on trajectory tracing were performed in order to demonstrate the performance of the proposed controller. In order to demonstrate the effectiveness of the optimization structure, the controller was implemented at onboard computer of the hexacopter through the software Matlab, additionally the hexacopter platform is shown in Figure 9 with the following hardware features Intel processor i7-7700HQ and CPU 2.80 GHz × 8. The controller was developed considering the optimal control problem (45a), the evolution of the identified model was developed using the fourth-order Runge–Kutta method, with a sample time of $t_s = 0.1[\text{s}]$ and the final time of the experiment is defined as $t_f = 100[\text{s}]$.

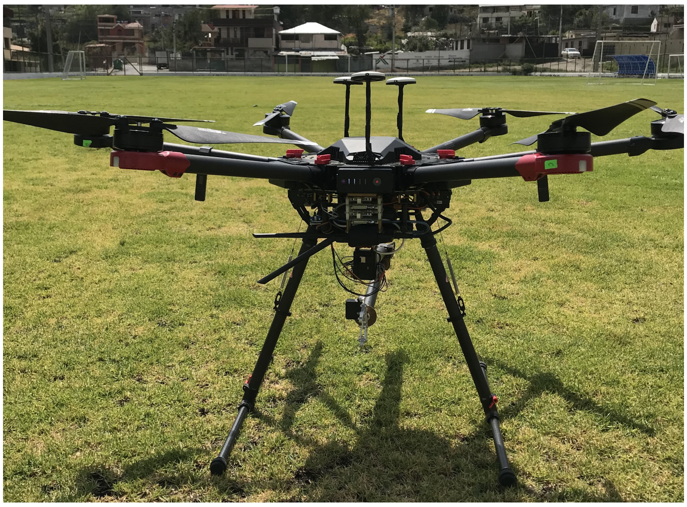

**Figure 9.** Hexacopter platform used in a real-world experiment.

The experiment was performed in the city of Ambato, in the province of Tungurahua, Ecuador, and started at 09:36 on 12 January 2022. The wind velocity at the time of the experiment was approximately 10.1 km/h, as shown in Figure 10.

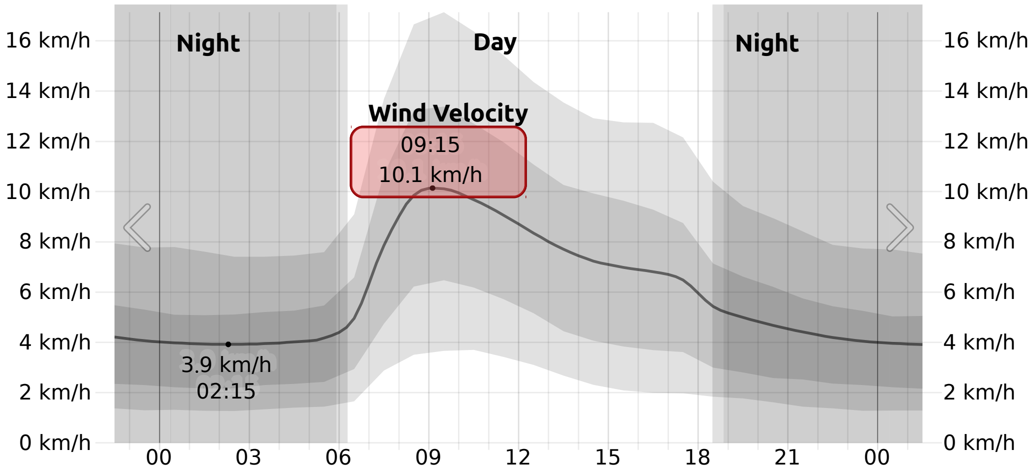

**Figure 10.** Wind velocity during experiment.

The desired reference trajectory for the hexacopter and the initial conditions are defined in Table 4, where $(\eta_{x_o}, \eta_{y_o}, \eta_{z_o}, \eta_{\psi_o})$ and $(\mu_{l_o}, \mu_{m_o}, \mu_{n_o}, \omega_o)$ are the initial position and maneuverability velocities of the hexacopter, respectively.

**Table 4.** Desired reference trajectory in a real-world experiment.

| Initial Positions | Initial Maneuverability Velocities | Reference Trajectory |
|---|---|---|
| $\eta_{x_o} = 4.18[\text{m}]$ | $\mu_{l_o} = 0.01[\text{m/s}]$ | $\eta_{x_{ref}} = 8\sin(0.1t)[\text{m}]$ |
| $\eta_{y_o} = 1.88[\text{m}]$ | $\mu_{m_o} = -0.04[\text{m/s}]$ | $\eta_{y_{ref}} = 6\sin(0.2t) + 1[\text{m}]$ |
| $\eta_{z_o} = 3.97[\text{m}]$ | $\mu_{n_o} = -0.02[\text{m/s}]$ | $\eta_{z_{ref}} = 0.35\sin(0.5t) + 7[\text{m}]$ |
| $\eta_{\psi_o} = 1.05[\text{rad}]$ | $\omega_o = -0.02[\text{rad/s}]$ | $\eta_{\psi_{ref}} = 0[\text{rad}]$ |

To implement the proposed controller, the constant values of positive definite matrices are defined in Table 5, and were obtained through a variety of experiments in order to improve the performance of the controller scheme.

**Table 5.** Proposed values for the NMPC scheme in a real-world experiment.

| Parameters | Values | Parameters | Values |
|---|---|---|---|
| $\mathbf{Q}_T$ | $diag(1) \in \mathbb{R}^{4\times4}$ | $\mathbf{Q}_t$ | $diag(1) \in \mathbb{R}^{4\times4}$ |
| $\mathbf{Q}_u$ | $diag(0.005) \in \mathbb{R}^{4\times4}$ | $T$ | 2 [s] |
| $\boldsymbol{\mu}_{\mathbf{ref}_{min}}$ | $-[1,1,1,1.5]$[m/s,rad/s] | $\boldsymbol{\mu}_{\mathbf{ref}_{max}}$ | $[1,1,1,1.5]$[m/s,rad/s] |
| $\Delta\boldsymbol{\mu}_{\mathbf{ref}_{max}}$ | 0.05[m/s,rad/s] | | |

The results of the experiment are illustrated in Figures 11–14. Figure 11 shows the movement of the hexacopter platform based on real information over the experiment. The hexacopter tracks the desired reference trajectory; however, the results present a small control error in the trajectory due to the wind velocity that acts as an external disturbance. The reference trajectory is defined as $\boldsymbol{\eta}_{\mathbf{ref}} = \begin{bmatrix} \eta_{x_{ref}} & \eta_{y_{ref}} & \eta_{z_{ref}} & \eta_{\psi_{ref}} \end{bmatrix}$ and the hexacopter system states during the experiment were confirmed by $\boldsymbol{\eta}$.

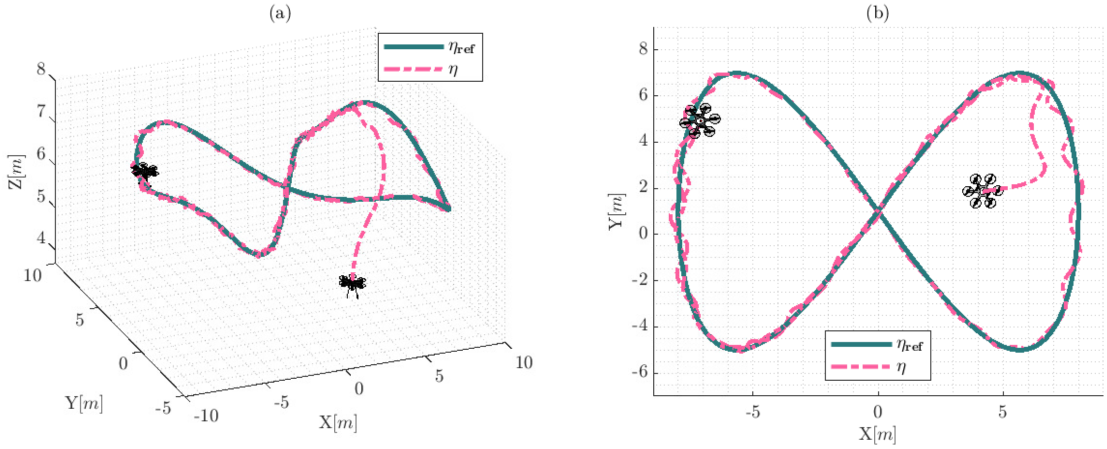

**Figure 11.** Movements of the hexacopter based on the real-world experimentation: (**a**) represents the behavior of the hexacopter platform using the 3D isometric perspective $\{\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z\}$; and (**b**) shows the upper view $\{\mathbf{I}_x, \mathbf{I}_y\}$ of the experimental information.

The previously presented results show the performance of the controller that guarantees the execution of the reference trajectory over different axes.

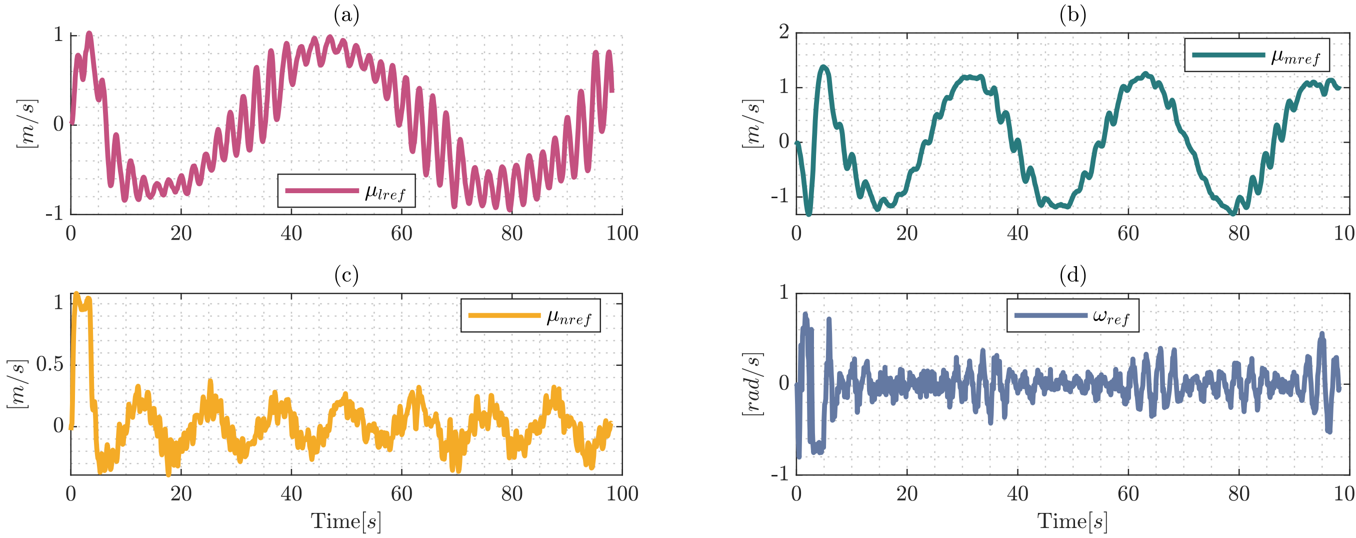

**Figure 12.** Control signals generated by the proposed control during the real-world experiment, where (**a**) represents the control velocity over the axis $\{\mathbf{I}_x\}$; (**b**) describes the evolution of velocity in $\{\mathbf{I}_y\}$; finally (**c**) and (**d**) represent the evolution of the upper and angular control velocity, respectively.

The control signals represented in Figure 12 are generated by the proposed control scheme, the external disturbances produce a miss match in the model but the controller generates the adequate smooth control actions.

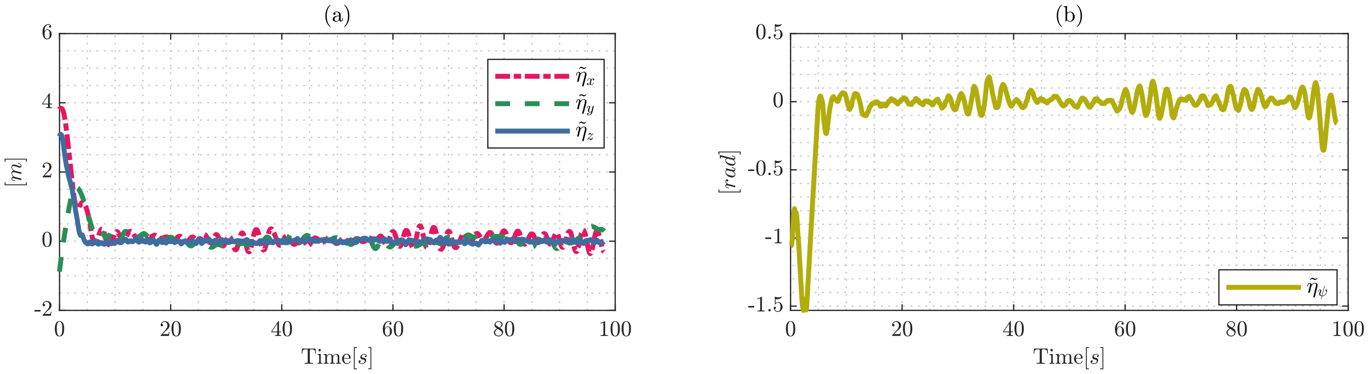

**Figure 13.** Control errors during a real outdoor experiment: (**a**) represents the error in the work-space $\{\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z\}$; and (**b**) shows the results with respect to the desired angular position.

Figure 13 shows the control errors of the proposed controller, where $\tilde{\eta}_x, \tilde{\eta}_y, \tilde{\eta}_z$ and $\tilde{\eta}_\psi$ are the results from the steady state error that asymptotically converge to values close to zero, i.e., since these position errors are bounded and different from zero $\boldsymbol{\eta}_{\mathbf{ref}}(t) - \mathbf{W_t x}(t) = 0 \in R^3$, it is achieved that the errors in the steady state are $|\tilde{\eta}| < 0.15[\mathrm{m}]$. In addition, the control error maintains a dependency on the external disturbances in the outdoor environments, and external disturbances are the product of the wind velocity which was approximately 10 km/h.

The time required to solve the optimal control problem is shown in Figure 14, and the computation time stays always below the sample time $t_s = 0.1$ [s], guaranteeing the efficient computation of the proposed control considering the time horizon of $T = 2$ [s] in the optimal control problem.

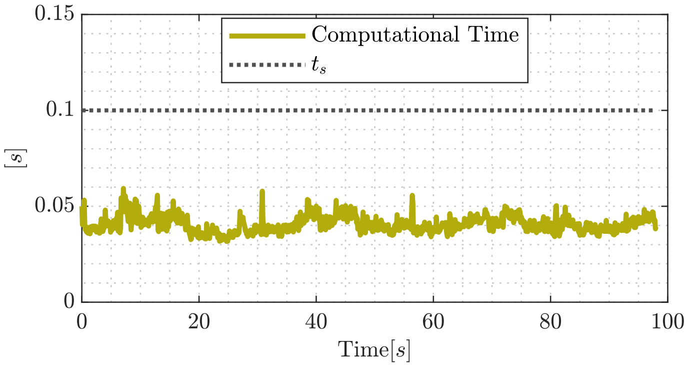

**Figure 14.** Computational time during the real-world experimental results.

### 3.2. Simulation Experiments

This section presents the simulation result of the proposed controller. In order to improve the results, this experiment was performed using the identified dynamics in Equation (36) with the identified parameters presented in Tables 2 and 3 . Furthermore, a Gaussian noise with the following consideration in all the states $\boldsymbol{\eta}_n \sim \mathcal{N}(-0.05, 0.05)$ and $\boldsymbol{\mu}_n \sim \mathcal{N}(-0.01, 0.01)$, where $\boldsymbol{\eta}_n$ and $\boldsymbol{\mu}_n$ are the additive noise in position and velocity states, was identified.

The desired reference trajectories and initial values over the frame $< \mathbb{I} >$ are defined in Table 6.

**Table 6.** Desired reference trajectory in the simulation experiment.

| Initial Positions | Initial Maneuverability Velocities | Reference Trajectory |
|---|---|---|
| $\eta_{x_o} = 4.18[\mathrm{m}]$ | $\mu_{l_o} = 0[\mathrm{m/s}]$ | $\eta_{x_{ref}} = 8\sin(0.1t)[\mathrm{m}]$ |
| $\eta_{y_o} = 1.88[\mathrm{m}]$ | $\mu_{m_o} = 0[\mathrm{m/s}]$ | $\eta_{y_{ref}} = 6\sin(0.2t) + 1[\mathrm{m}]$ |
| $\eta_{z_o} = 3.97[\mathrm{m}]$ | $\mu_{n_o} = 0[\mathrm{m/s}]$ | $\eta_{z_{ref}} = 0.35\sin(0.5t) + 7[\mathrm{m}]$ |
| $\eta_{\psi_o} = 1.05[\mathrm{rad}]$ | $\omega_o = 0[\mathrm{rad/s}]$ | $\eta_{\psi_{ref}} = 0[\mathrm{rad}]$ |

The desired values of the reference trajectory are the same as the proposed in real-world experiments; therefore, the results present similar behavior. The controller constant values are the same as those presented in Table 5. However, the safe region distance was defined as $r_{obs} = 0.4[\mathrm{m}]$.

The trajectory described by the hexacopter platform and the reference trajectory are presented in Figure 15. The results show that the hexacopter tracks the desired reference trajectory, in addition to the avoidance obstacles property being demonstrated due to the system following the reference and trying to maintain the distance to obstacles considering the safety radius $r_{obs}$. The three obstacles were positioned over the reference trajectory, two of which were positioned at the corners are dynamic obstacles and they have movements during the experiment.

The values of the control errors presented in Figure 16 tend to increase when the hexacopter avoids the obstacles satisfying the system constraints, a phenomenon which is represented in the rectangles representing obstacles close to the hexarotor. Given that the errors in steady state $\tilde{\eta}_x, \tilde{\eta}_y, \tilde{\eta}_z$ and $\tilde{\eta}_\psi$ symptomatically converge towards zero in the presence of the Gaussian noise applied in the simulation, the results show the robustness under this type of disturbance and the ability to keep the system under the reference trajectory, i.e., the position errors are bounded and are different from zero $\boldsymbol{\eta}_{\mathbf{ref}}(t) - \mathbf{W_t x}(t) = 0$, achieving a final characteristic error with a max $|\tilde{\eta}| < 0.9[\mathrm{m}]$ when the obstacles are close to the hexarotor and $|\tilde{\eta}| < 0.1[\mathrm{m}]$ without obstacles near the system reference trajectory.

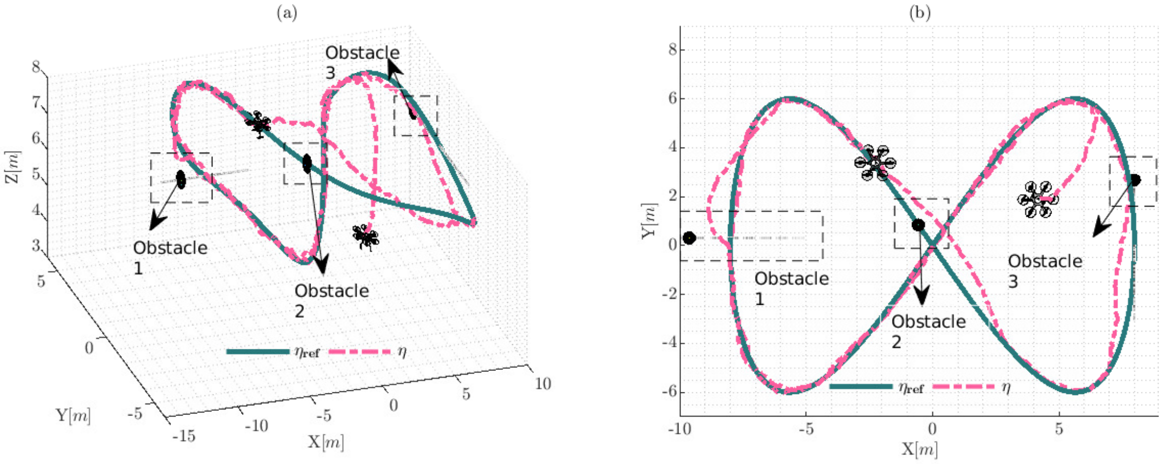

**Figure 15.** Movements of the hexacopter during the simulation experimentation: (**a**) represents the behavior of the hexacopter platform using the 3D isometric perspective $\{\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z\}$; and (**b**) shows the upper view of the experimental information; furthermore, the black spheres represent the static and dynamic obstacles. The velocity of the dynamic obstacles was approximately $0.2\cos(0.1t)[\mathrm{m/s}]$ over $\mathbf{I}_x$ for the first obstacle and approximately $0.6\cos(0.2t)[\mathrm{m/s}]$ over $\mathbf{I}_y$ for the third obstacle. The second obstacle was considered static and the position is static during the experiment.

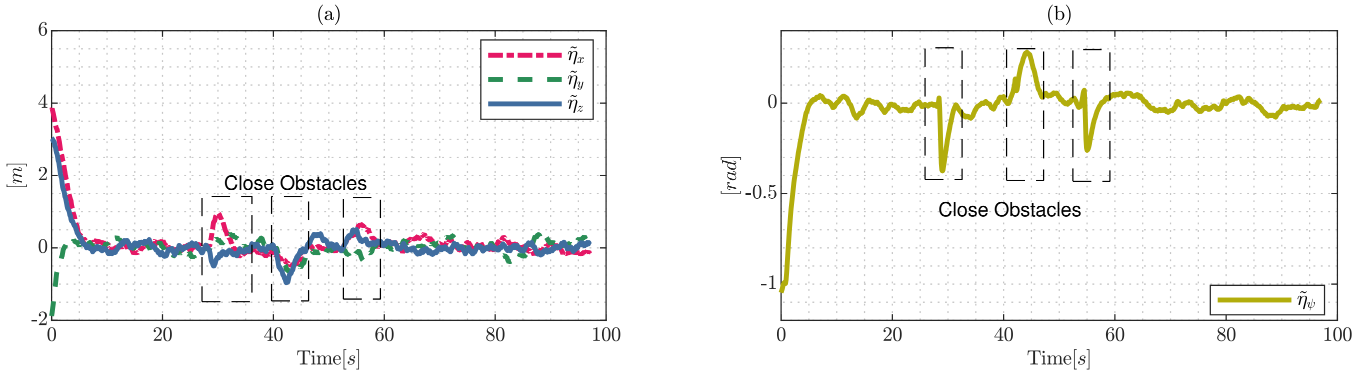

**Figure 16.** Control errors during the simulated experiment: (**a**) represents the error in the work-space; and (**b**) shows the error with respect to angular displacements.

The control signals are presented in Figure 17, which are generated by the proposed control scheme. Taking into consideration the dynamics of the robotic system and the simulated environmental conditions, the behavior of the reference velocities are close to the dynamic model values. Furthermore, the values are presented as smooth curves with the presence of peaks when the obstacles are close to the hexacopter which are emphasized by the rectangles shown in Figure 17. The behavior of the control actions prevents collisions between the hexacopter and the obstacles during the simulation.

The time required to solve the proposed controller and the Euclidean distance are presented in Figure 18, and the computational time is an important factor to demonstrate its ability to generate an adequate control policy that guarantees the system constraints. The computational time presents peaks produced by the obstacles that are close to the hexacopter during the trajectory whilst the controller finds a sub-optimal solution to guarantee that the computational time remains under the sample time; however, this problem can be efficiently solved using another low-level language to implement the proposed controller or with a more powerful computer, which is therefore a solvable technological issue. On the other hand, the distance to each obstacle shows that the proposed controller respects the safely radius, guaranteeing the avoidance property of the system.

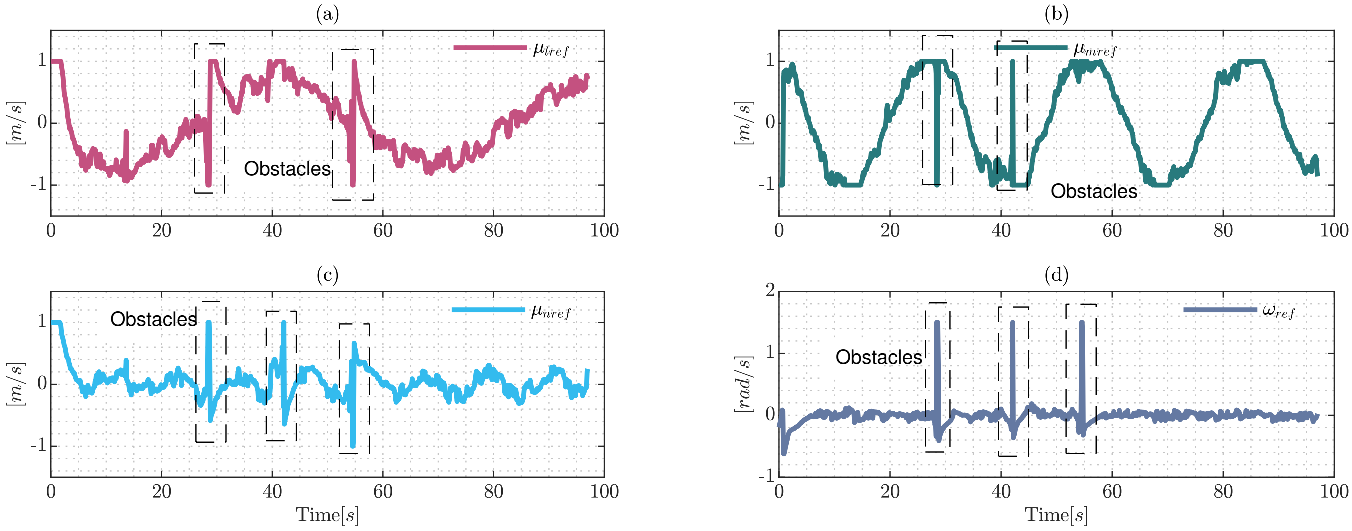

**Figure 17.** Control signals generated by the proposed control and maneuverability velocities were generated by the dynamic approximation, where (**a**) represents the maneuverability velocities over the axis $\{\mathbf{I}_x\}$; (**b**) describes the evolution of velocities in $\{\mathbf{I}_y\}$; finally, (**c**) and (**d**) represent the evolution of the upper and angular velocity, respectively.

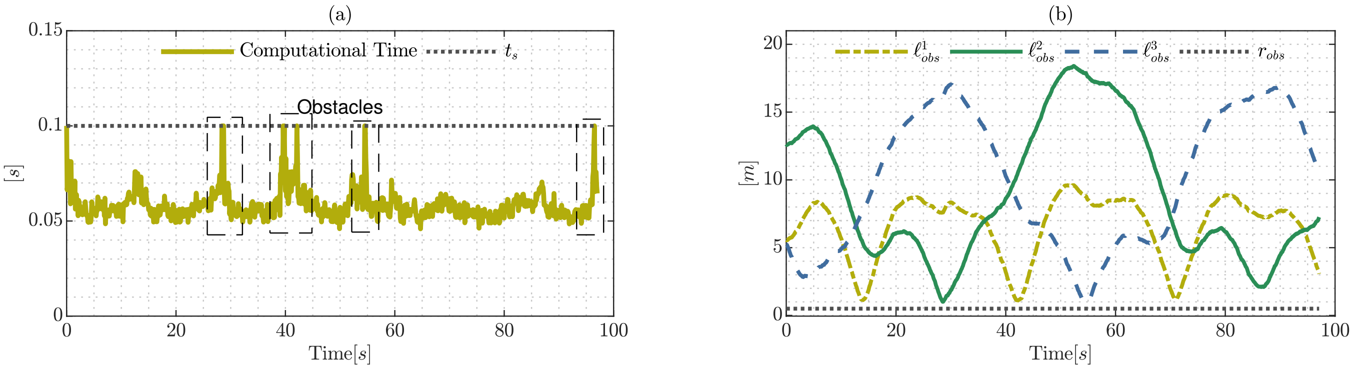

**Figure 18.** Computational time and distance to obstacles: (**a**) represents the time to solve the optimization problem; and (**b**) shows the distance to the obstacles, where $\ell^1_{obs}$ $\ell^2_{obs}$ and $\ell^3_{obs}$ are the respective distances and $r_{obs}$ is the radius of security considering the dimensions of the hexarotor frame.

### *3.3. Discussion*

The experiments presented in the last section show that the nonlinear model predictive scheme solves the control problem associated with the trajectory tracking subject to the system and environment constraints. Specifically, the real-world experiments showed the robustness of the proposed controller due to the excessive wind velocity during the experiment which acted as an external disturbance. Despite the excessive wind velocity, the controller generates the adequate smooth control actions guaranteeing that the steady state error that converges towards values close to zero with the dependency of the external disturbances. The use of CasADI such as an optimization framework guarantees the fact that the computational time always stays below the sample time, which is one of main problems in NMPC schemes due to the highly computational time that these solutions required, as similar computing times were presented in [45].

On the other hand, the simulation results show the ability of the proposed scheme to generates the adequate control signals guaranteeing the corrected tracking trajectories while the system avoids obstacles during the simulation. The presence of Gaussian noise provides uncertainty in the measurements demonstrating the power of the controller as it maintains the distance to obstacles in consideration of the safety radius. One of the main differences presented in this experiment was the control error, which tends to increase when the hexacopter avoids obstacles satisfying the system constraints. Furthermore, the computational time presents peaks produced by the hard constraints included in the optimization problem, and despite the hard constraint, the controller finds a sub-optimal solution to guarantee a low computational time. Other works have used the concept of soft constraints, low-level programming language and embeddable optimization methods such as (SQP) in order to increase the convergence of the optimization algorithm and reduce the computational time [60–64]. This work uses the concept of soft constraints and the approximation of the nonlinear problem in order to improve the computational time and the future implementation on a single on-board PC.

## 4. Conclusions

This work carried out the identification of a dynamic system using the dynamic mode decomposition with control technique to develop the optimal control problem for trajectory tracking with obstacle avoidance, a decision which was made due to the precision of the comparative results between the Euler–Lagrange and the dynamic mode decomposition formulations. The optimal control problem was translated into a nonlinear programming problem (NLP) using the multiple shooting technique and considering the control actions and systems states as decision variables, which is a technique that improves the computational time and the convergence of the algorithm. The nonlinear model predictive control (NMPC) generates the maneuverability velocities policy that allows the hexacopter platform to track the reference trajectory while avoiding obstacles presented in the environment.

The NMPC formulation was developed in consideration of the reference desired trajectory and constraints in the system such as the bounded control actions, the generalized dynamics and static and dynamic obstacles. Furthermore, the NMPC problem was solved through the nonlinear programming framework CasADI, which solves high-dimensional optimization problems and has advantages that can be expanded to a low-level programming language to improve the computational time.

The simulation results show how the hexacopter avoids both static and dynamic obstacles whose locations vary over time while executing the trajectory tracking task, and always maintaining a safe distance defined as a radius of repulsion. Since there is no sensor for obstacle measurements in the hexacopter platform, the real experimental tests were not carried out with obstacle avoidance; however it was verified that although the experimental tests were carried out during the period of time in which the average wind speed was approximately 10 km/h, the behavior of the control law on the hexacopter system fulfills the desired task, so that the control errors in stable state converge closer to zero. In addition, it can be observed that the steady state error asymptotically converges towards values closer to zero under conditions wherein the control actions are saturate when they reach the established restrictions. On the other hand, the computational time of the proposed control algorithm is maintained under 100 ms in real experimentation due to the optimization technique used to solve the control problem. During the simulation tests, it is clearly verified that the computational time remains under 100 ms; however, when obstacle avoidance occurs, this time increases but remains under the specified sample time with a sup-optimal solution. Other works use the concept of soft constraints to guarantee the rapid convergence of the optimization algorithm, as this work will use the same concept to improve the computational time and the possible expansion to visual-servoing systems under an MPC structure.

**Author Contributions:** Conceptualization, L.F.R., V.H.A. and B.S.G.; methodology, L.F.R., C.P.C. and V.H.A.; software, L.F.R., C.P.C. and B.S.G.; validation, L.F.R., C.P.C. and B.S.G.; formal analysis, L.F.R., V.H.A. and D.C.G.; investigation, L.F.R.; resources, L.F.R., V.H.A. and J.V.-A.; data preparation, C.P.C. and D.C.G.; writing—original draft preparation, L.F.R., B.S.G. and C.P.C.; writing—review and editing, J.V.-A., V.H.A. and D.C.G.; visualization, L.F.R. and B.S.G.; supervision, V.H.A., D.C.G. and J.V.-A.; project administration, V.H.A., C.P.C. and J.V.-A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The APC was funded by Universidad Tecnológica Indoámerica (Ecuador).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** If anyone wants to obtain the original data of this paper, please contact with Luis F. Recalde.

**Acknowledgments:** The authors would like to thank the Universidad de las Fuerzas Armadas ESPE; Universidad Tecnológica Indoamérica; SISAu Research Group, and the Research Group ARSI, for the support and development of this work. The results of this work are part of the research project Advanced Control of Unmanned Aerial Vehicles developed by Universidad de las Fuerzas Armadas ESPE, and of the project "Control Cooperativo de Robots con Óptimo Consumo de Recursos" developed by Universidad Tecnológica Indoamérica.

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

## Appendix A

The dynamic parameters developed during the system identification of the hexarotor aerial platform are as follows:

$$\begin{bmatrix} \mu_{lref} \\ \mu_{mref} \\ \mu_{nref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} \frac{K_{dl}+m}{K_{pl}} & 0 & 0 & -\frac{2bm}{K_{pl}} \\ 0 & \frac{K_{dm}+m}{K_{pm}} & 0 & \frac{2am}{K_{pm}} \\ 0 & 0 & \frac{K_{dn}+m}{K_{pn}} & 0 \\ -\frac{bm}{K_{pw}} & \frac{am}{K_{pw}} & 0 & \frac{2m(a+b)+I+K_{dw}}{K_{pw}} \end{bmatrix} \begin{bmatrix} \dot{\mu}_l \\ \dot{\mu}_m \\ \dot{\mu}_n \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 1 & -\frac{m\omega}{K_{pl}} & 0 & -\frac{2am\omega}{K_{pl}} \\ \frac{m\omega}{K_{pm}} & 1 & 0 & -\frac{2bm}{K_{pm}} \\ 0 & 0 & 1 & 0 \\ \frac{am\omega}{K_{pw}} & \frac{bm\omega}{K_{pw}} & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \tag{A1}$$

$$\begin{aligned} \zeta_1 = \frac{K_{dl}+m}{K_{pl}}, \zeta_2 = -\frac{2bm}{K_{pl}}, \zeta_3 = \frac{K_{dm}+m}{K_{pm}}, \zeta_4 = \frac{2am}{K_{pm}}, \zeta_5 = \frac{K_{dn}+m}{K_{pn}}, \\ \zeta_6 = -\frac{bm}{K_{pw}}, \zeta_7 = \frac{am}{K_{pw}}, \zeta_8 = \frac{2m(a+b)}{K_{pw}}, \zeta_9 = \frac{I+K_{dw}}{K_{pw}}, \zeta_{10} = 1, \\ \zeta_{11} = -\frac{m\omega}{K_{pl}}, \zeta_{12} = -\frac{2am\omega}{K_{pl}}, \zeta_{13} = \frac{m\omega}{K_{pm}}, \zeta_{14} = 1, \zeta_{15} = -\frac{2bm}{K_{pm}}, \\ \zeta_{16} = 1, \zeta_{17} = \frac{am\omega}{K_{pw}}, \zeta_{18} = \frac{bm\omega}{K_{pw}}, \zeta_{19} = 1, \end{aligned} \tag{A2}$$

## References

1. Recker, T.; Heilemann, F.; Raatz, A. Handling of large and heavy objects using a single mobile manipulator in combination with a roller board. *Procedia CIRP* **2021**, *97*, 21–26. [CrossRef]
2. Ortiz, J.S.; Molina, M.F.; Andaluz, V.H.; Varela, J.; Morales, V. Coordinated Control of a Omnidirectional Double Mobile Manipulator. *Lect. Notes Electr. Eng.* **2018**, *449*, 278–286. [CrossRef]
3. Ortiz, J.S.; Palacios-navarro, G.; Andaluz, V.H.; Recalde, L.F. Three-Dimensional Unified Motion Control of a Robotic Standing Wheelchair for Rehabilitation Purposes. *Sensors* **2021**, *21*, 3057. [CrossRef] [PubMed]
4. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 1–18. [CrossRef]
5. Jiménez-Moreno, R.; Pinzón-Arenas, J.O.; Pachón-Suescún, C.G. Assistant robot through deep learning. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 1053–1062. [CrossRef]
6. Yolcu, V.; Demirer, V. A Review on the Studies about the Use of Robotic Technologies in Education. *SDU Int. J. Educ. Stud.* **2017**, *4*, 127–139.
7. Petersen, K.H.; Napp, N.; Stuart-Smith, R.; Rus, D.; Kovac, M. A review of collective robotic construction. *Sci. Robot.* **2019**, *4*. [CrossRef]
8. Pan, M.; Linner, T.; Pan, W.; Cheng, H.; Bock, T. A framework of indicators for assessing construction automation and robotics in the sustainability context. *J. Clean. Prod.* **2018**, *182*, 82–95. [CrossRef]
9. Chebrolu, N.; Lottes, P.; Schaefer, A.; Winterhalter, W.; Burgard, W.; Stachniss, C. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *Int. J. Robot. Res.* **2017**, *36*, 1045–1052. [CrossRef]
10. Kuric, I.; Bulej, V.; Saga, M.; Pokorny, P. Development of simulation software for mobile robot path planning within multilayer map system based on metric and topological maps. *Int. J. Adv. Robot. Syst.* **2017**, *14*. [CrossRef]
11. Shaw, I.G. Robot Wars: US Empire and geopolitics in the robotic age. *Secur. Dialogue* **2017**, *48*, 451–470. [CrossRef] [PubMed]
12. Mellinger, D.; Shomin, M.; Michael, N.; Kumar, V. Cooperative Grasping and Transport Using Multiple Quadrotors. *Springer Tracts Adv. Robot.* **2013**, *83*, 545–558. [CrossRef]
13. Bircher, A.; Kamel, M.; Alexis, K.; Burri, M.; Oettershagen, P.; Omari, S.; Mantel, T.; Siegwart, R. Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Auton. Robot.* **2015**, *40*, 1059–1078. [CrossRef]
14. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding horizon next-best-view planner for 3D exploration. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1462–1468. [CrossRef]
15. Garimella, G.; Kobilarov, M. Towards model-predictive control for aerial pick-and-place. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 4692–4697. [CrossRef]
16. Song, Y.; Hsu, L.T. Tightly coupled integrated navigation system via factor graph for UAV indoor localization. *Aerosp. Sci. Technol.* **2021**, *108*, 106370. [CrossRef]
17. Paredes, J.A.; Álvarez, F.J.; Aguilera, T.; Villadangos, J.M. 3D Indoor Positioning of UAVs with Spread Spectrum Ultrasound and Time-of-Flight Cameras. *Sensors* **2017**, *18*, 89. [CrossRef]

18. Sandino, J.; Vanegas, F.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments. *Remote Sens.* **2020**, *12*, 3386. [CrossRef]
19. Radoglou–Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020**, *172*, 107148. [CrossRef]
20. Gupta, A.; Afrin, T.; Scully, E.; Yodo, N. Advances of UAVs toward Future Transportation: The State-of-the-Art, Challenges, and Opportunities. *Future Transp.* **2021**, *1*, 326–350. [CrossRef]
21. Chen, Y.; Zhang, G.; Zhuang, Y.; Hu, H. Autonomous Flight Control for Multi-Rotor UAVs Flying at Low Altitude. *IEEE Access* **2019**, *7*, 42614–42625. [CrossRef]
22. Vu, N.A.; Dang, D.K.; Le Dinh, T. Electric propulsion system sizing methodology for an agriculture multicopter. *Aerosp. Sci. Technol.* **2019**, *90*, 314–326. [CrossRef]
23. Belmonte, N.; Staulo, S.; Fiorot, S.; Luetto, C.; Rizzi, P.; Baricco, M. Fuel cell powered octocopter for inspection of mobile cranes: Design, cost analysis and environmental impacts. *Appl. Energy* **2018**, *215*, 556–565. [CrossRef]
24. Nguyen, N.P.; Kim, W.; Moon, J. Super-twisting observer-based sliding mode control with fuzzy variable gains and its applications to fully-actuated hexarotors. *J. Frankl. Inst.* **2019**, *356*, 4270–4303. [CrossRef]
25. Ali, R.; Peng, Y.; Iqbal, M.T.; Ul Amin, R.; Zahid, O.; Khan, O.I. Adaptive backstepping sliding mode control of coaxial octorotor unmanned aerial vehicle. *IEEE Access* **2019**, *7*, 27526–27534. [CrossRef]
26. Alaimo, A.; Artale, V.; Milazzo, C.L.R.; Ricciardello, A. PID Controller Applied to Hexacopter Flight. *J. Intell. Robot. Syst.* **2013**, *73*, 261–270. [CrossRef]
27. Dayana Salim, N.; Derawi, D.; Abdullah, S.S.; Mazlan, S.A.; Zamzuri, H. PID plus LQR attitude control for hexarotor MAV in indoor environments. In Proceedings of the IEEE International Conference on Industrial Technology, Busan, Korea, 26 February–1 March 2014; pp. 85–90. [CrossRef]
28. Chen, F.; Lei, W.; Zhang, K.; Tao, G.; Jiang, B. A novel nonlinear resilient control for a quadrotor UAV via backstepping control and nonlinear disturbance observer. *Nonlinear Dyn.* **2016**, *2*, 1281–1295. [CrossRef]
29. Wang, Q.; Zhang, H.; Han, J. Research on Trajectory Planning and Tracking of Hexa-copter. *MATEC Web Conf.* **2018**, *173*, 02008. [CrossRef]
30. Lee, T.; Leok, M.; McClamroch, N.H. Geometric tracking control of a quadrotor UAV on SE(3). In Proceedings of the IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010; pp. 5420–5425. [CrossRef]
31. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2520–2525. [CrossRef]
32. Tal, E.; Karaman, S. Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 1203–1218. [CrossRef]
33. Faessler, M.; Franchi, A.; Scaramuzza, D. Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. *IEEE Robot. Autom. Lett.* **2018**, *3*, 620–626. [CrossRef]
34. Santoso, F.; Garratt, M.A.; Anavatti, S.G. A self-learning TS-fuzzy system based on the C-means clustering technique for controlling the altitude of a hexacopter unmanned aerial vehicle. In Proceedings of the 2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 12–14 October 2017; pp. 46–51. [CrossRef]
35. Dierks, T.; Jagannathan, S. Output feedback control of a quadrotor UAV using neural networks. *IEEE Trans. Neural Netw.* **2010**, *21*, 50–66. [CrossRef]
36. Ferdaus, M.M.; Pratama, M.; Anavatti, S.G.; Garratt, M. A Generic Self-Evolving Neuro-Fuzzy Controller Based High-Performance Hexacopter Altitude Control System. In Proceedings of the SMC 2018: 2018 IEEE International Conference on Systems, Man, and Cybernetics, Miyazaki, Japan, 7–10 October 2018; pp. 2784–2791. [CrossRef]
37. ul Amin, R.; Aijun, L.; Khan, M.U.; Shamshirband, S.; Kamsin, A.; ul Amin, R.; Aijun, L.; Khan, M.U.; Shamshirband, S.; Kamsin, A. An adaptive trajectory tracking control of four rotor hover vehicle using extended normalized radial basis function network. *MSSP* **2017**, *83*, 53–74. [CrossRef]
38. Nieuwenhuisen, M.; Droeschel, D.; Schneider, J.; Holz, D.; Labe, T.; Behnke, S. Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In Proceedings of the ECMR 2013: 2013 European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 7–12. [CrossRef]
39. Rezaee, H.; Abdollahi, F. Adaptive artificial potential field approach for obstacle avoidance of unmanned aircrafts. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Kaohsiung, Taiwan, 11–14 July 2012. [CrossRef]
40. Achtelik, M.W.; Weiss, S.; Chli, M.; Siegwart, R. Path planning for motion dependent state estimation on micro aerial vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3926–3932. [CrossRef]
41. He, R.; Prentice, S.; Roy, N. Planning in information space for a quadrotor helicopter in a GPS-denied environment. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1814–1820. [CrossRef]
42. Sani, M.; Robu, B.; Hably, A. Pursuit-evasion game for nonholonomic mobile robots with obstacle avoidance using NMPC. In Proceedings of the MED 2020: 2020 28th Mediterranean Conference on Control and Automation, Saint-Raphaël, France, 15–18 September 2020; pp. 978–983. [CrossRef]

43. Subramanian, S.; Nazari, S.; Alvi, M.A.; Engell, S. Robust NMPC Schemes for the Control of Mobile Robots in the Presence of Dynamic Obstacles. In Proceedings of the MMAR 2018: 2018 23rd International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 27–30 August 2018; pp. 768–773. [CrossRef]
44. Ribeiro, T.T.; Fernandez, R.O.; Conceicao, A.G. NMPC-based Visual Leader-Follower Formation Control for Wheeled Mobile Robots. In Proceedings of the IEEE 16th International Conference on Industrial Informatics, INDIN 2018, Porto, Portugal, 18–20 July 2018; pp. 406–411. [CrossRef]
45. Osman, M.; Mehrez, M.W.; Yang, S.; Jeon, S.; Melek, W. End-Effector Stabilization of a 10-DOF Mobile Manipulator using Nonlinear Model Predictive Control. *IFAC-PapersOnLine* **2020**, *53*, 9772–9777. [CrossRef]
46. Neunert, M.; De Crousaz, C.; Furrer, F.; Kamel, M.; Farshidian, F.; Siegwart, R.; Buchli, J. Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1398–1404. [CrossRef]
47. Aoki, Y.; Asano, Y.; Honda, A.; Motooka, N.; Ohtsuka, T. Nonlinear Model Predictive Control of Position and Attitude in a Hexacopter with Three Failed Rotors. *IFAC-PapersOnLine* **2018**, *51*, 228–233. [CrossRef]
48. Tzoumanikas, D.; Graule, F.; Yan, Q.; Shah, D.; Popovic, M.; Leutenegger, S. Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing. *arXiv* **2020**, arXiv:2006.02116.
49. Bicego, D.; Mazzetto, J.; Carli, R.; Farina, M.; Franchi, A. Nonlinear Model Predictive Control with Enhanced Actuator Model for Multi-Rotor Aerial Vehicles with Generic Designs. *J. Intell. Robot. Syst. Theory Appl.* **2020**, *100*, 1213–1247. [CrossRef]
50. Foehn, P.; Romero, A.; Scaramuzza, D. Time-Optimal Planning for Quadrotor Waypoint Flight. *Sci. Robot.* **2021**, *6*. [CrossRef]
51. Torrente, G.; Kaufmann, E.; Fohn, P.; Scaramuzza, D. Data-Driven MPC for Quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3769–3776. [CrossRef]
52. Small, E.; Sopasakis, P.; Fresk, E.; Patrinos, P.; Nikolakopoulos, G. Aerial navigation in obstructed environments with embedded nonlinear model predictive control. In Proceedings of the 2019 18th European Control Conference, ECC 2019, Naples, Italy, 25–28 June 2019; pp. 3556–3563. [CrossRef]
53. Nguyen, H.; Kamel, M.; Alexis, K.; Siegwart, R. Model Predictive Control for Micro Aerial Vehicles: A Survey. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021; pp. 1556–1563. [CrossRef]
54. Quan, Q. *Introduction to Multicopter Design and Control*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–384. [CrossRef]
55. Zhang, H.; Rowley, C.W.; Deem, E.A.; Cattafesta, L.N. Online dynamic mode decomposition for time-varying systems. *SIAM J. Appl. Dyn. Syst.* **2017**, *18*, 1586–1609. [CrossRef]
56. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [CrossRef]
57. Proctor, J.L.; Brunton, S.L.; Kutz, J.N. Dynamic Mode Decomposition with Control. *SIAM J. Appl. Dyn. Syst.* **2016**, *15*, 142–161. [CrossRef]
58. Albersmeyer, J.; Diehl, M. The Lifted Newton Method and Its Application in Optimization. *SIAM J. Optim.* **2010**, *20*, 1655–1684. [CrossRef]
59. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2018**, *11*, 1–36. [CrossRef]
60. Farshidian, F.; Kamgarpour, M.; Pardo, D.; Buchli, J. Sequential Linear Quadratic Optimal Control for Nonlinear Switched Systems. *IFAC-PapersOnLine* **2016**, *50*, 1463–1469. [CrossRef]
61. Farshidian, F.; Neunert, M.; Winkler, A.W.; Rey, G.; Buchli, J. An Efficient Optimal Planning and Control Framework For Quadrupedal Locomotion. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 93–100. [CrossRef]
62. Giftthaler, M.; Farshidian, F.; Sandy, T.; Stadelmann, L.; Buchli, J. Efficient Kinematic Planning for Mobile Manipulators with Non-holonomic Constraints Using Optimal Control. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 3411–3417. [CrossRef]
63. Verschueren, R.; Frison, G.; Kouzoupis, D.; Frey, J.; Duijkeren, N.V.; Zanelli, A.; Novoselnik, B.; Albin, T.; Quirynen, R.; Diehl, M. acados: A modular open-source framework for fast embedded optimal control. *Math. Program. Comput.* **2019**, *14*, 147–183. [CrossRef]
64. Gaertner, M.; Bjelonic, M.; Farshidian, F.; Hutter, M. Collision-Free MPC for Legged Robots in Static and Dynamic Scenes. In Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 8266–8272. [CrossRef]