drones

MDPI

Article

A Comparative Study between NMPC and Baseline Feedback Controllers for UAV Trajectory Tracking

Bryan S. Guevara [1,2,†], Luis F. Recalde [2,3,†], José Varela-Aldás [1,3,*,†], Victor H. Andaluz [3,4], Daniel C. Gandolfo [2] and Juan M. Toibero [2]

[1] Centro de Investigaciones de Ciencias Humanas y de la Educación (CICHE), Universidad Indoamérica, Ambato 180103, Ecuador
[2] Instituto de Automática, Universidad Nacional de San Juan—CONICET, Av. San Martín Oeste 1109, San Juan J5400ARL, Argentina
[3] SISAu Research Group, Facultad de Ingeniería, Industria y Producción, Universidad Indoamérica, Ambato 180103, Ecuador
[4] Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas—ESPE, Sangolquí 171103, Ecuador
* Correspondence: josevarela@uti.edu.ec
† These authors contributed equally to this work.

Abstract: Transport, rescue, search, surveillance, and disaster relief tasks are some applications that can be developed with unmanned aerial vehicles (UAVs), where accurate trajectory tracking is a crucial property to operate in a cluttered environment or under uncertainties. However, this is challenging due to high nonlinear dynamics, system constraints, and uncertainties presented in cluttered environments. Hence, uncertainties in the form of unmodeled dynamics, aerodynamic effects, and external disturbances such as wind can produce unstable feedback control schemes, introducing significant positional tracking errors. This work presents a detailed comparative study between controllers such as nonlinear model predictive control (NMPC) and non-predictive baseline feedback controllers, with particular attention to tracking accuracy and computational efficiency. The development of the non-predictive feedback controller schemes was divided into inverse differential kinematics and inverse dynamic compensation of the aerial vehicle. The design of the two controllers uses the mathematical model of UAV and nonlinear control theory, guaranteeing a low computational cost and an asymptotically stable algorithm. The NMPC formulation was developed considering system constraints, where the simplified dynamic model was included; additionally, the boundaries in control actions and a candidate Lyapunov function guarantees the stability of the control structure. Finally, this work uses the commercial simulator DJI brand and DJI Matrice 100 UAV in real-world experiments, where the NMPC shows a reduction in tracking error, indicating the advantages of this formulation.

Keywords: optimal control; nonlinear control; quadrotor UAV; trajectory tracking

check for updates

Citation: Guevara, B.S.; Recalde, L.F.; Varela-Aldás, J.; Andaluz, V.H.; Gandolfo, D.C.; Toibero, J.M. A Comparative Study between NMPC and Baseline Feedback Controllers for UAV Trajectory Tracking. *Drones* **2023**, *7*, 144. https://doi.org/10.3390/drones7020144

Academic Editor: Xiangwei Bu

Received: 31 January 2023
Revised: 16 February 2023
Accepted: 16 February 2023
Published: 20 February 2023

1. Introduction

1.1. Motivation

Nowadays, unmanned aerial vehicles (UAVs) are applied in many fields due to recent technological advances and the ability to perform tasks that humans cannot do. Their fields of application can be industry, transportation, and agriculture [1]. Quadrotor platforms are a type of UAV widely adopted in aerial robotic applications and extensively used by the scientific community due to high maneuverability, vertical take-off, and landing (VTOL). Significant results have been published [2–6], where such systems were employed successfully in applications as remote sensing and environment interaction [7–10], autonomous delivery [11,12], and exploration in unknown environments [13–16].

The extreme agility of quadrotors can be used for critical missions such as rescue, search, and transportation; this kind of application requires a minimal execution time that is only possible at high-speed trajectories, where accurate trajectory tracking and robustness are critical features. These features are still challenging problems for the research community; furthermore, the ability to precisely control quadrotors during agile maneuvers would allow faster trajectories and a wider range of motion that enables the system to be safely closer to obstacles, where slight deviations from the reference trajectories can produce serious accidents. In response to this fact, various controller structures have been proposed. However, only some approaches can handle the issues presented during flights in cluttered environments, such as non-linear dynamics, uncertainties, and limitations in control actions.

This work considers that the problem of quadrotor control in cluttered environments can be divided into two groups: (i) Non-predictive Feedback Controllers, which use formulations of linear and nonlinear control exploiting system model structure to design control laws to guarantee the system stability and execution of the desired tasks. However, these techniques do not handle system constraints such as limitations presented in control actions, ignore future decisions, and formulations are not trivial for complex systems; (ii) Predictive Optimization Methods which uses the model of the system to generate the optimal control policy that guarantees the execution of the desired task minimizing an objective function considering the system and input constraints in a predictive manner. Furthermore, this formulation is capable of fully exploiting the capabilities of the quadrotor and taking advantage of the available computational power. Optimal control approaches have presented notorious developments; specifically, nonlinear model predictive control (NMPC) has experimented with outstanding attention for quadrotor applications due to the last advances in hardware and algorithms, allowing the execution of these formulations with high computational efficiency [17–20].

However, NMPC is computationally expensive compared to two of the baseline feedback controllers, such as inverse differential kinematics and inverse dynamic compensation, where these formulations have presented successful developments in the field of robotics [21–23]. Although these formulations have not achieved quadrotor control under agile trajectories as NMPC proposed in [19,24]; moreover, the high computational cost associated with NMPC could make its application questionable. Therefore, a comparative study of NMPC and Baseline Feedback Controllers is needed to provide insights for future developments, reporting benefits and drawbacks in terms of performance improvement, disturbance rejection, and computation effort applied in quadrotor control under cluttered environments.

1.2. Related Work

This work divides the trajectory tracking controllers into two big groups: *(i) Non-Predictive Formulation*, where this scheme only depends on a single reference set-point at the current time, *(ii) Predictive Formulation* which considers multiple future references into the control scheme at the moment the algorithm is being executed. In the following section, this work presents a short review of the scientific community's latest contributions in trajectory tracking accuracy under these two formulations. For more detailed information about quadrotor controllers, refer to the following works [25,26].

1.2.1. Non-Predictive Controllers

Trajectory tracking controllers have been developed during the last decade by research institutes. The literature developed generally considers slight angle deviations, which guarantee hover or near hover along the execution of the trajectory. For formulations based on linear control methods, refer to the following works [27,28].

However, the usual hover assumptions are not enough for cluttered environments, where nonlinearities appear as one of the first problems to handle. Nonlinear control formulations are valuable tools that allow the design of algorithms capable of mitigating problems associated with cluttered environments, such as nonlinearities and robustness. For this reason, some works use nonlinear feedback control, such as backstepping formulations [29], feedback linearization [30,31], and robust nonlinear control [32–34]. In addition, quaternions are used to avoid singularities presented in the mathematical model using Euler angles formulation of aerial vehicles [35].

Moreover, intelligent control theory has been applied in a wide range of works, where the main formulations use concepts of artificial neural networks [36,37], fuzzy logic systems [38,39], and reinforcement learning techniques [40,41]. Generally, these formulations cannot handle states or control input constraints; furthermore, the reference trajectory needs to be selected carefully, which is not viable in cluttered environments.

1.2.2. Predictive Controllers

MPC has presented outstanding precision and robustness properties among the multiple approaches applied in quadrotor trajectory tracking, so this formulation has been considered in numerous articles to demonstrate some important properties in advanced UAV control. Due to the predictive nature of its formulation is possible to generate the control policy optimizing the reference states and the constraints at the same time, which is beneficial because the majority of systems present limitations in their states and control actions. In the same way, control commands could be generated by minimizing the tracking error in the predicted time horizon by solving constrained optimization problems, with successful results in applications, as shown in [17,18,42,43].

Nevertheless, MPC formulations can be computationally expensive, specifically nonlinear model predictive control NMPC [42,44]. For this reason, linear model predictive control is usually adopted in many works, where the nonlinearities are simplified using linear approximations under small angle variations assumptions [30,45]. Thanks to the recent advances in technology and the new optimization solvers [46], many works started using nonlinear dynamics, exploiting the capabilities of the quadrotors [17,18]. Although NMPC formulations are realizable on modern computers, the authors still require significantly more computational resources than non-predictive schemes such as inverse differential kinematics and inverse dynamic compensation. Hence, NMPC may experiment with numerical convergence problems associated with computational capabilities. Finally, it is necessary to compare these two formulations and realize which conditions present better performance or complications related to tracking accuracy and computational time.

1.3. Contribution of This Work

This work presents a comparative study between *(i) Non-predictive Feedback Controllers*, conformed by inverse differential kinematics and inverse dynamic compensation controllers and *(ii) Predictive Optimization Methods* with the Nonlinear Model Predictive Control formulation. To show the advantages and capabilities of each controller, this work reports comparative experiments considering a wide range of agile trajectories with a particular emphasis on tracking accuracy, computational efficiency, and robustness. The main contributions of this work are:

- Differential Kinematics and Dynamic Formulation: this work develops the differential kinematics and dynamics formulation considering maneuverability velocities as the control input of the system because many commercial aerial vehicles are produced under this structure where the user does not have access to attitude control schemes. Thus, this formulation can be straightforwardly applied to another commercial platform with desired velocities as inputs of the system. To obtain the dynamic model of the system, this work used the formulation of Euler–Lagrange; additionally, to modify this model into the velocities space, it is necessary to introduce the concept of low-level PID in the mathematical formulation.
- *Nonlinear Feedback Formulations:* are generated by inversion of differential kinematics and dynamic compensation techniques. The design of these control structures was

developed using the theory of nonlinear control and linear algebra, guaranteeing minimal computational time with asymptotically and robust behavior.

- *NMPC Formulation:* This work formulates the NMPC considering system and input constraints such as system dynamics, limits in control action, rate of change of control inputs, and a candidate Lyapunov function that guarantees the stability of the control structure. This work uses CasADI to solve the nonlinear optimization problem .
- *Comparative Study:* The experiments are carried out in simulations and real-world environments. The simulation experiments were performed under the commercial simulator of the DJI brand. On the other hand, real-world experiments were conducted using the DJI Matrice 100 platform, which was equipped with all the necessary hardware to run the controllers on the onboard computer. The agile trajectory selected to compare the performances of the controllers is the Lissajous; additionally, to guarantee a good comparison, this work considers low and high velocities through the reference signal where the aerodynamics effects, latency, and uncertainties are inevitable. Finally, with the comparative results is possible to show the benefits of each controller considering their predictive and non-predictive structures; the comparative consider the following metrics: tracking accuracy and average computational time.

This work is organized as follows: Section 2 presents the preliminary materials, such as differential kinematics and dynamic models. The formulation of the quadrotor controllers is presented in Section 3 with a subsection describing the non-predictive (inverse differential kinetics and inverse dynamic compensation and predictive (NMPC) schemes. Section 4 shows the comparative results through simulations and real-world experiments. The discussions and conclusions of this work are presented in Section 5 and Section 6, respectively.

2. Preliminary Materials

This section presents the formal notations used throughout the paper and the model employed to describe the motions of the Quadrotor platform. First, an instantaneous kinematic model was derived, and then the simplified dynamic model was formulated using the Euler–Lagrange. Finally, identification and validation processes are necessary to employ the models in the controller formulations.

2.1. Kinematics Formulation

This work formulates the mathematical representation of the quadrotor, defining the world-fixed inertial frame as $< \mathbb{I} >$ and the frame attached to the body of the system as $< \mathbb{B} >$, which is aligned with the center of mass (CoM), these notations are shown in Figure 1.

This work expresses the pose of the quadrotor using the following definition $\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p}(\boldsymbol{\eta}) & \boldsymbol{\phi}(\boldsymbol{\eta}) \end{bmatrix}^\mathsf{T}$, where $\mathbf{p}(\boldsymbol{\eta}) = \begin{bmatrix} \eta_x & \eta_y & \eta_z \end{bmatrix}$ and $\boldsymbol{\phi}(\boldsymbol{\eta}) = \begin{bmatrix} \eta_\varphi & \eta_v & \eta_\psi \end{bmatrix}$ are the position and the orientation respect $< \mathbb{I} >$; further, due to the low-level controllers based on PID formulations which guarantee the hover position, this work assumes rotations only through the vertical axis $\mathbf{B}_z$ and slight angular deviation along the angles (φ, v) roll and pitch respectively.

Thus, the minimum representation is defined as $\boldsymbol{\eta} = \begin{bmatrix} \eta_x & \eta_y & \eta_z & \eta_\psi \end{bmatrix}^\mathsf{T}$ expressed with more detail by the following equation:

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

where $\boldsymbol{\eta} = \begin{bmatrix} \eta_{x_0} & \eta_{y_0} & \eta_{z_0} \end{bmatrix}^\mathsf{T}$ is CoM position; additionally this work defines the point of interest of the system displacement from the CoM, which can be generally defined as (d_x, d_y, d_z).

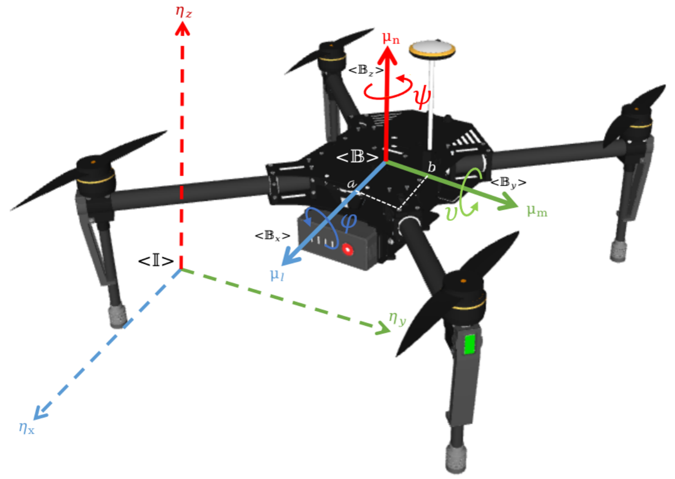

Figure 1. DJI Matrice 100 UAV platform representation.

To formulate the controller schemes is necessary to know the evolution of the system over time; in fact of this, the instantaneous kinematics model of the quadrotor is defined by the vector $\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\eta}_x & \dot{\eta}_y & \dot{\eta}_z & \dot{\eta}_\psi \end{bmatrix}^\mathsf{T}$ with the following matrix form representation:

$$\begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_z \\ \dot{\eta}_\psi \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & -\rho_1 \\ \sin(\psi) & \cos(\psi) & 0 & \rho_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \tag{2}$$

where linear and angular velocities are defined by $\boldsymbol{\mu} = \begin{bmatrix} \mu_l & \mu_m & \mu_n & \omega \end{bmatrix}^\mathsf{T}$, additionally $\rho_1 = d_x \sin(\psi) + d_y \cos(\psi)$ and $\rho_2 = d_x \cos(\psi) - d_y \sin(\psi)$ represent the behavior generated by displacement of the point of interest. A compact formulation can be defined by the following equation:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_k(\psi)\boldsymbol{\mu} \tag{3}$$

where the linear mapping between the evolution of the point of interest $\dot{\boldsymbol{\eta}}$ and the control maneuverability velocities $\boldsymbol{\mu}$ can be defined by the analytical Jacobian matrix $\mathbf{J}_k(\psi)$.

For the simplified dynamic section based on maneuverability velocities [24], it is necessary to define the time derivative of (3) expressed in its compact form as:

$$\ddot{\boldsymbol{\eta}} = \mathbf{J}_k(\psi)\dot{\boldsymbol{\mu}} + \dot{\mathbf{J}}_k(\psi, \omega)\boldsymbol{\mu} \tag{4}$$

where $\dot{\boldsymbol{\mu}} = \begin{bmatrix} \dot{\mu}_l & \dot{\mu_m} & \dot{\mu}_n & \dot{\omega} \end{bmatrix}^\mathsf{T}$ is the acceleration vector of the quadrotor and $\dot{\mathbf{J}}(\psi, \omega)$ is the time derivative of the analytic Jacobian defined as

$$\dot{\mathbf{J}}(\psi, \omega) = \begin{bmatrix} -\omega \sin(\psi) & -\omega \cos(\psi) & 0 & -\omega(d_x \cos(\psi) - d_y \sin(\psi)) \\ \omega \cos(\psi) & -\omega \sin(\psi) & 0 & \omega(d_x \sin(\psi) + d_y \cos(\psi)) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2.2. Quadrotor Dynamics

The formulation of a mathematical model that defines the precise behavior of the quadrotor flight is obtained through the Euler–Lagrange equations, which describe the evolution of a mechanical system subject to holonomic constraints. With these considerations, the Euler–Lagrange can be systematically defined using the total kinetic and potential energy present in the system. The Euler–Lagrange equations [47,48] are expressed as follows:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{\eta}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\eta}} = \boldsymbol{\tau} \tag{5}$$

where $\mathcal{L} = \mathcal{K} - \mathcal{U}$ represents the *Lagrangian* as a difference between the kinetic energy and the potential energy and $\boldsymbol{\tau} = \begin{bmatrix} f_{x_I} & f_{y_I} & f_{z_I} & \tau_{\psi_I} \end{bmatrix}^{\mathsf{T}}$ represents the forces and torques applied in each degree of freedom. The total kinetic energy of the system $\mathcal{K} = \mathcal{K}_v + \mathcal{K}_\omega$ is defined as the sum of energies associated with translational and rotational velocities.

The translational term contributes to the translational kinetic energy of the system, defined as

$$\mathcal{K}_v = \frac{1}{2} m \dot{\boldsymbol{\eta}}^{\mathsf{T}} \dot{\boldsymbol{\eta}} \tag{6}$$

where m represents the mass of the quadrotor; additionally, the rotational kinetic energy is expressed:

$$\mathcal{K}_\omega = \frac{1}{2} \dot{\boldsymbol{\omega}}^{\mathsf{T}} \mathbf{I}_\omega \dot{\boldsymbol{\omega}} \tag{7}$$

where $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathsf{T}}$ is the angular velocity and $\mathbf{I}_\omega$ is the inertia matrix expressed in a fixed coordinate system $< \mathbb{I} >$. Then the angular velocity can be expressed as a function of the rotational velocity in $\boldsymbol{\phi}$ as

$$\boldsymbol{\omega} = \mathbf{R}^{\mathsf{T}}(\boldsymbol{\phi}) \mathbf{J}_\omega(\boldsymbol{\phi}) \dot{\boldsymbol{\phi}} \tag{8}$$

where $\mathbf{R}$ is a rotation matrix that transforms the vectors of the coordinate system in the body frame $< \mathbb{B} >$ to the absolute inertial coordinate system $< \mathbb{I} >$; $\mathbf{J}_\omega$ represents the geometric Jacobian which is related to the analytical Jacobian $\mathbf{J}_\omega$ by the following equation:

$$\mathbf{J}_\omega(\boldsymbol{\phi}) = \mathbf{T}_A(\boldsymbol{\phi}) \mathbf{J}_k(\psi) \tag{9}$$

where

$$\mathbf{T}_A(\boldsymbol{\phi}) = \begin{bmatrix} I & 0 \\ 0 & T_\omega(\boldsymbol{\phi}) \end{bmatrix} \; \mathbf{T}_\omega(\boldsymbol{\phi}) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi)\sin(v) \\ 0 & \cos(\varphi) & \sin(\varphi)\sin(v) \\ 1 & 0 & \cos(\varphi) \end{bmatrix} \tag{10}$$

Similarly, we can formulate $\boldsymbol{\omega} = \mathbf{T}_\omega(\boldsymbol{\phi})\dot{\boldsymbol{\phi}}$ to formulate the total kinetic energy of the system as a function of generalized coordinates and the relationship between analytical and geometric Jacobian [47], fulfilling that:

$$\mathcal{K} = \frac{1}{2} \dot{\boldsymbol{\eta}}^{\mathsf{T}} \left[m \mathbf{J}_k^{\mathsf{T}}(\psi) \mathbf{J}_k(\psi) + \mathbf{J}_\omega^{\mathsf{T}}(\boldsymbol{\phi}) \mathbf{R}(\boldsymbol{\phi}) \mathbf{I} \mathbf{R}^{\mathsf{T}}(\boldsymbol{\phi}) \mathbf{J}_\omega(\boldsymbol{\phi}) \right] \dot{\boldsymbol{\eta}} \tag{11}$$

Finally, with the assumptions presented above, the total kinetic energy of the system can be written compactly as

$$\mathcal{K} = \frac{1}{2} \dot{\boldsymbol{\eta}}^{\mathsf{T}} \mathbf{M}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} \tag{12}$$

where $\mathbf{M}(\boldsymbol{\eta})$ is the inertia matrix, which is symmetric and positive definite. The potential energy of the quadrotor arises only due to gravity defined by

$$\mathcal{U} = m \mathbf{g}^{\mathsf{T}} \boldsymbol{\eta} \tag{13}$$

where $\mathbf{g} = \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix}^{\mathsf{T}}$ is the vector of gravitational accelerations in the absolute coordinate system $< \mathbb{I} >$. Thus, the Lagrangian for the quadrotor can be written as

$$\mathcal{L}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \frac{1}{2} \dot{\boldsymbol{\eta}}^{\mathsf{T}} \mathbf{M}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} - m \mathbf{g}^{\mathsf{T}} \boldsymbol{\eta} \tag{14}$$

Then applying the partial derivatives and Euler–Lagrange's constraint (5), the dynamic model of the quadrotor can be written in matrix form as

$$\begin{bmatrix} f_{x_I} \\ f_{y_I} \\ f_{z_I} \\ \tau_{\psi_I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & -m(d_y\cos(\psi) + d_x\sin(\psi)) \\ 0 & m & 0 & m(d_x\cos(\psi) - d_y\sin(\psi)) \\ 0 & 0 & m & 0 \\ -m(d_y\cos(\psi) + d_x\sin(\psi)) & m(d_x\cos(\psi) - d_y\sin(\psi)) & 0 & m{d_x}^2 + m{d_y}^2 + I \end{bmatrix} \begin{bmatrix} \ddot{\eta}_x \\ \ddot{\eta}_y \\ \ddot{\eta}_z \\ \ddot{\eta}_\psi \end{bmatrix} \dots$$
$$\dots + \begin{bmatrix} 0 & 0 & 0 & -m\dot{\eta}_\psi(d_x\cos(\psi) - d_y\sin(\psi)) \\ 0 & 0 & 0 & -m\dot{\eta}_\psi(d_y\cos(\psi) + d_x\sin(\psi)) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_z \\ \dot{\eta}_\psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ gm \\ 0 \end{bmatrix} \tag{15}$$

Therefore, the dynamic model of the quadrotor can be written in compact form:

$$\mathbf{M}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{16}$$

where, the Coriolis effects $\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ and gravitational forces $\mathbf{g}(\boldsymbol{\eta})$ can be represented by the following matrices: $\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \dot{\mathbf{M}} - \frac{1}{2}\dot{\boldsymbol{\eta}}^{\mathsf{T}}\frac{\partial \mathbf{M}}{\partial \boldsymbol{\eta}}$ and $\mathbf{g}(\boldsymbol{\eta}) = \frac{\partial \mathcal{U}}{\partial \boldsymbol{\eta}}$.

2.3. Simplified Dynamic Model Based on Maneuvering Velocities

This section presents the following premise; the quadrotor remains in a hover position performed by PD dynamic controller with gravity compensation. The forces $\mathbf{f}_{t_B} = \begin{bmatrix} f_{x_B} & f_{y_B} & \tau_{\psi_B} \end{bmatrix}$ govern the attitude PD controller of the quadrotor; furthermore these forces can be also defined as

$$\mathbf{f}_{t_B} = \mathbf{K}_{pt}(\boldsymbol{\mu}_{t_{ref}} - \boldsymbol{\mu}_t) + \mathbf{K}_{dt}(\dot{\boldsymbol{\mu}}_{t_{ref}} - \dot{\boldsymbol{\mu}}_t) \tag{17}$$

where $\boldsymbol{\mu}_t = \begin{bmatrix} \mu_l & \mu_m & \omega \end{bmatrix}^{\mathsf{T}}$ is the velocity vector that defines the attitude of the quadrotor in the horizontal plane; $\boldsymbol{\mu}_{t_{ref}} = \begin{bmatrix} \mu_{lref} & \mu_{mref} & \omega_{ref} \end{bmatrix}^{\mathsf{T}}$ is the reference horizontal attitude vector; $(\dot{\boldsymbol{\mu}}_t, \dot{\boldsymbol{\mu}}_{t_{ref}})$ are the time derivatives of $\boldsymbol{\mu}_t$ and $\boldsymbol{\mu}_{t_{ref}}$, respectively; and $\mathbf{K}_{pt}$, $\mathbf{K}_{dt}$ are diagonal positive definite gain matrices.

Furthermore, this work uses the following assumption $\dot{\boldsymbol{\mu}}_{t_{ref}} = 0$ if $\lim_{t\to\infty}\|\mathbf{f}_{t_B}\| = 0$ and $\lim_{t\to\infty}\|\tilde{\boldsymbol{\mu}}_t\| = 0$, where $\tilde{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_{t_{ref}} - \boldsymbol{\mu}_t$. Now, considering the vertical thrust force $f_{l_B} = f_{z_B} + mg$ that opposes the force of gravity, the altitude PD controller is defined in (18) under the assumption that $\lim_{t\to\infty}\|f_{z_I}\| = 0$ and $\lim_{t\to\infty}\|\tilde{\mu}_n\| = 0$ if $\dot{\mu}_{n_{ref}} = 0$, considering that $\tilde{\mu}_n = \mu_{n_{ref}} - \mu_n$.

$$f_{l_B} = k_{pl}(\mu_{n_{ref}} - \mu_n) + k_{dl}(\dot{\mu}_{n_{ref}} - \dot{\mu}_n) + mg \tag{18}$$

where $\mu_{n_{ref}}$ is the reference vertical velocity; $\dot{\boldsymbol{\mu}}_n$ and $\dot{\boldsymbol{\mu}}_{n_{ref}}$ are the time derivatives of μ_n and $\mu_{n_{ref}}$, respectively; and k_{pl}, k_{dl} are positive gain scalars that define the proportional and derivative gains of the PD controller.

Due to the above, the general structure of the hover controller based on altitude and attitude control is expressed as

$$\mathbf{f_B} = \mathbf{K_p}\boldsymbol{\mu}_{\mathbf{ref}} - \mathbf{K_p}\boldsymbol{\mu} - \mathbf{K_d}\dot{\boldsymbol{\mu}} + \mathbf{g} \tag{19}$$

where $\mathbf{g} = \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix}$; $\mathbf{K_p} \in \mathbb{R}^{4\times4}$ is a diagonal matrix of proportional gains and $\mathbf{K_d} \in \mathbb{R}^{4\times4}$ is a diagonal matrix of derivative gains.

Finally, substitute Equations (3) and (4) into the expression resulting from equating the dynamic models (16) and (19), then the new simplified dynamic model of the quadrotor is obtained, which considers the manipulability velocities as input of the system:

$$\begin{bmatrix} \mu_{lref} \\ \mu_{mref} \\ \mu_{nref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} \pi_1 & 0 & 0 & d_x\omega\pi_2 \\ 0 & \pi_3 & 0 & d_y\omega\pi_4 \\ 0 & 0 & \pi_5 & 0 \\ d_x\omega\pi_6 & d_y\omega\pi_7 & 0 & \pi_8 \end{bmatrix} \begin{bmatrix} \dot{\mu}_l \\ \dot{\mu}_m \\ \dot{\mu}_n \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \pi_9 & 0 & 0 & d_x\omega\pi_{10} \\ 0 & \pi_{11} & 0 & d_y\omega\pi_{12} \\ 0 & 0 & \pi_{13} & 0 \\ d_y\omega^2\pi_{14} & d_x\omega^2\pi_{15} & 0 & d_x^2\omega^2\pi_{16} + d_y^2\omega^2\pi_{17} + \pi_{18} \end{bmatrix} \begin{bmatrix} \mu_l \\ \mu_m \\ \mu_n \\ \omega \end{bmatrix} \tag{20}$$

This new simplified dynamic model can be applied to any commercial quadrotor UAV with maneuverability velocities defined as inputs of the system. The following equation expresses the simplified dynamic model expressed in a compact form:

$$\boldsymbol{\mu}_{\mathbf{ref}} = \bar{\mathbf{M}}(\boldsymbol{\mu},\boldsymbol{\pi})\dot{\boldsymbol{\mu}} + \bar{\mathbf{C}}(\boldsymbol{\mu},\boldsymbol{\pi})\boldsymbol{\mu} \tag{21}$$

where $\bar{\boldsymbol{M}}(\boldsymbol{\mu},\boldsymbol{\pi}) = (\boldsymbol{RK_p})^{-1}((\boldsymbol{MJ} + \boldsymbol{RK_d})$ is the new matrix of mass and inertia; $\bar{\boldsymbol{C}}(\boldsymbol{\mu},\boldsymbol{\pi}) = (\boldsymbol{RK_p})^{-1}(\boldsymbol{M\dot{J}} + \boldsymbol{RK_p} + \boldsymbol{CJ})$ is the new matrix of centripetal and Coriolis forces; finally, $\boldsymbol{\pi} = \begin{bmatrix} \pi_1 & \pi_2 & .. & \pi_{18} \end{bmatrix}$ is the vector of dynamic parameters that includes all the physical phenomena that change the quadrotor's dynamics.

2.4. *Identification and Validation*

This section shows the identification and validation of the dynamic model presented in (21), which due to the important property of dynamic systems equation, can be formulated in a linear parameter representation [48] as $\boldsymbol{\mu}_{\mathbf{ref}}(t) = \mathbf{Y}(\dot{\boldsymbol{\mu}},\boldsymbol{\mu})\boldsymbol{\pi}$, where $\mathbf{Y}(\dot{\boldsymbol{\mu}},\boldsymbol{\mu})$ is a regressor matrix, and $\boldsymbol{\pi}$ is the vector of linear parameters.

The identification process establishes the relation between the mathematical formulation and the real-world system, allowing the use of the model in control structures. This process uses the measurements obtained from the aerial vehicle DJI Matrice 100, and due to the presence of noise during the flight measurements, this work applies a filter in the identification experiment defined by $\bar{\boldsymbol{\mu}}_{\mathbf{ref}} = \frac{\lambda}{(s+\lambda)}\boldsymbol{\mu}_{\mathbf{ref}}$ and $\bar{\boldsymbol{\mu}} = \frac{\lambda}{(s+\lambda)}\boldsymbol{\mu}$. The identification process takes the filters signals and generates the following matrices:

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{Y}_1(\dot{\bar{\boldsymbol{\mu}}}_1,\bar{\boldsymbol{\mu}}_1) \\ \vdots \\ \mathbf{Y}_s(\dot{\bar{\boldsymbol{\mu}}}_s,\bar{\boldsymbol{\mu}}_s) \end{bmatrix} \qquad \boldsymbol{\Omega} = \begin{bmatrix} \bar{\boldsymbol{\mu}}_{\mathbf{ref_1}} \\ \vdots \\ \bar{\boldsymbol{\mu}}_{\mathbf{ref_s}} \end{bmatrix} \tag{22}$$

where s is the number of measurements used during the identification process, thus, the system can be written as follows:

$$\boldsymbol{\Phi}\pi = \boldsymbol{\Omega} \tag{23}$$

Finally, this work uses the method of least squares to identify the system parameters defining the following equation:

$$\boldsymbol{\pi} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{\Omega} \tag{24}$$

where $\boldsymbol{\Phi}^{\dagger}$ is defined as $(\boldsymbol{\Phi}^{\intercal}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\intercal}$.

Table 1 show the values of the parameters obtained using the least squares method,

Table 1. Parameters of the aerial vehicle DJI Matrice 100.

Parameters			
$\pi_1 = 2.11$	$\pi_2 = -0.005$	$\pi_3 = 1.8$	$\pi_4 = 3.17$
$\pi_5 = 1.78$	$\pi_6 = 0.39$	$\pi_7 = -0.003$	$\pi_8 = -0.03$
$\pi_9 = 0.006$	$\pi_{10} = 0.02$	$\pi_{11} = 0.002$	$\pi_{12} = 0.06$
$\pi_{13} = 0.70$	$\pi_{14} = 0.02$	$\pi_{15} = -0.05$	$\pi_{16} = -0.01$
$\pi_{17} = -0.005$	$\pi_{18} = -0.01$		

Figure 2a shows the evolution of the velocities estimated by the dynamic model $(\mu_{lm}, \mu_{mm}, \mu_{nm}, \omega_m)$ presented in (21) and developed by the fourth order Runge-Kutta method. Finally, Figure 2b compares the real velocities and the velocities generated by the dynamic model with a new set of reference velocities to verify the accuracy of the expected behavior.

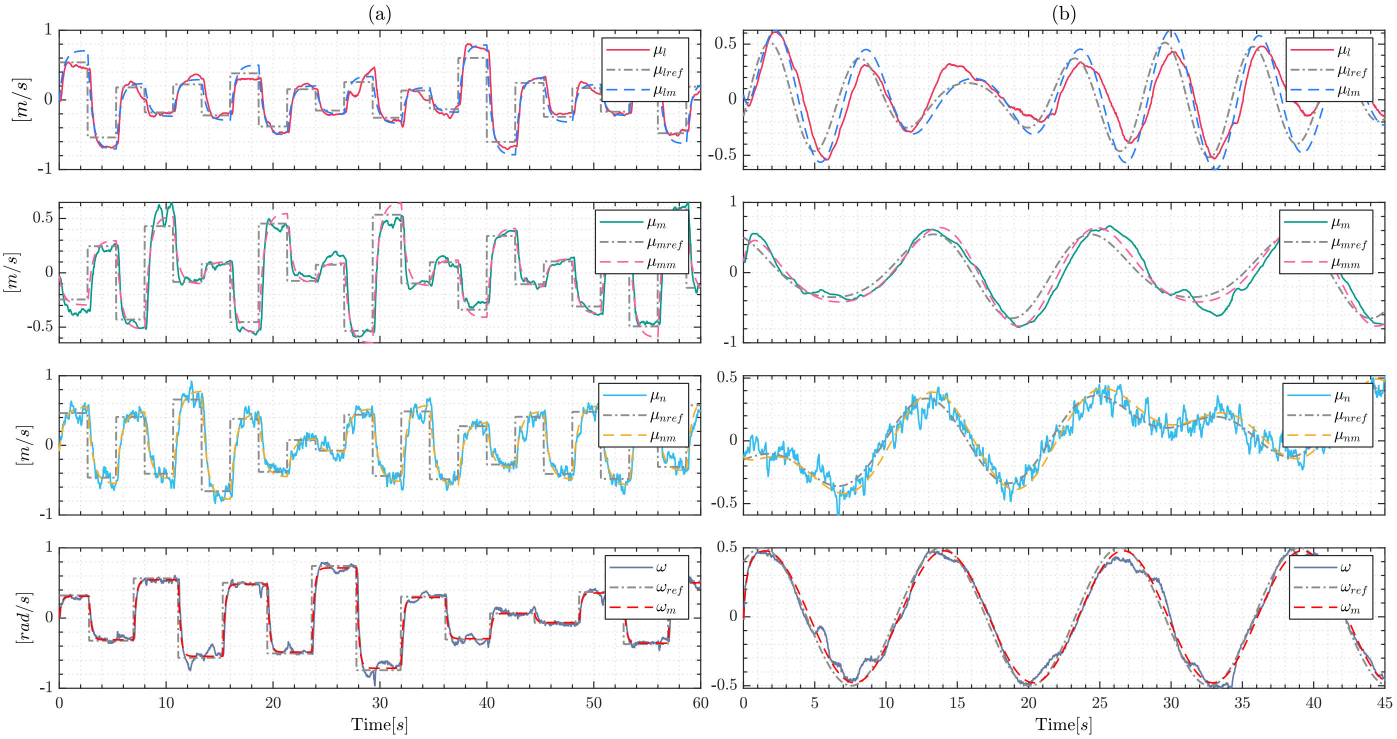

Figure 2. Identification and validation, (**a**) shows the identification process under step signals, and (**b**) presents the validation of the proposed dynamic model considering smooth signals.

3. Controllers Design

This work presents the formulation of the *(i) Non-predictive Feedback Controllers*, conformed by inverse differential kinematics and inverse dynamic compensation controllers [49] and *(ii) Predictive Optimization Methods* with the non-linear model predictive control formulation.

3.1. Kinematic Controller

The kinematic controller tries to calculate the instantaneous velocities of the quadrotor to track the desired trajectory at the same time that the error in the steady state converges asymptotically to zero. The general scheme of the kinematic controller implemented is shown in Figure 3.

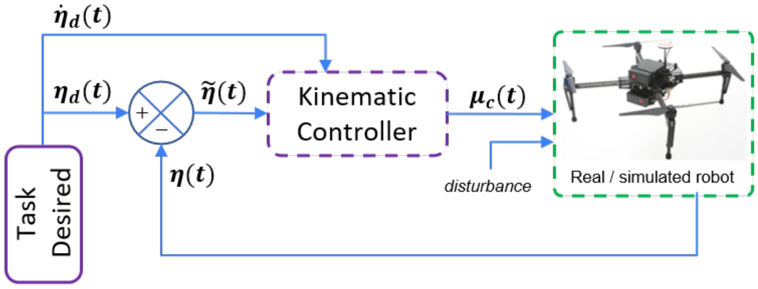

Figure 3. General scheme of the kinematic controller.

The control law generates the desired velocities for trajectory tracking:

$$\boldsymbol{\mu}_c = \mathbf{J}_k^{\dagger}(\psi)\left[\dot{\boldsymbol{\eta}}_d + \mathbf{K}_2 tanh(\mathbf{K}_2^{-1}\mathbf{K}_1\tilde{\boldsymbol{\eta}})\right] \tag{25}$$

where $\dot{\boldsymbol{\eta}}_d$ are the velocities associated with the desired trajectory; $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$ are the position errors; additionally, $\tanh(\cdot)$ is a saturation function where $\mathbf{K}_1 > 0$ is a diagonal matrix defines the slopes of the saturation function and $\mathbf{K}_2 > 0$ is a diagonal matrix that represents the maximum and minimum velocities generated by the controller. The use of $\tanh(\cdot)$ in the control formulation enables the consideration of limits in control actions, which is a remarkable feature in developing a comparative study.

Kinematic Controller Stability Analysis

This work demonstrates the stability of the proposed controller considering that $\boldsymbol{\mu} \equiv \boldsymbol{\mu}_c$; thus, the closed-loop equation of the system is defined as

$$\dot{\tilde{\boldsymbol{\eta}}} = -\mathbf{K_2} \tanh\left(\mathbf{K_2^{-1}K_1}\tilde{\boldsymbol{\eta}}\right) \tag{26}$$

A Lyapunov candidate function based on the control errors is defined as follows:

$$\mathbf{V}_k = \frac{1}{2}\tilde{\boldsymbol{\eta}}^{\mathsf{T}}\boldsymbol{\Lambda}\tilde{\boldsymbol{\eta}} \tag{27}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times4}$ is a positive definite diagonal matrix. The time derivative of the candidate Lyapunov function along the trajectory of the system is defined by

$$\dot{\mathbf{V}}_k(\tilde{\boldsymbol{\eta}}) = -\tilde{\boldsymbol{\eta}}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{K_2} \tanh\left(\mathbf{K_2^{-1}K_1}\tilde{\boldsymbol{\eta}}\right) \tag{28}$$

According to the Lyapunov criterion, if $\mathbf{K}_1 > 0$ and $\mathbf{K}_2 > 0$ then the stability of the system is guaranteed because $\dot{\mathbf{V}}_k(\tilde{\boldsymbol{\eta}}) < 0$, which means that the error converges to zero asymptotically $\tilde{\boldsymbol{\eta}} \to 0$ since $t \to \infty$.

3.2. Dynamic Controller

The quadrotor presents dynamic effects related to inertia movements; this generates velocities errors $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu_c} - \boldsymbol{\mu}$ with respect to the instantaneous velocities generated by the kinematic controller. To reduce this effect, Figure 4 shows the general control scheme where the dynamic compensation generates reference velocities $\boldsymbol{\mu}_{ref}$ that compensate for the effects generated by the inertia movements.

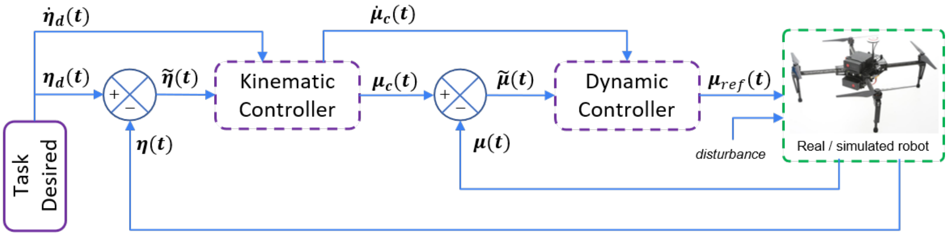

Figure 4. General scheme of the dynamic controller.

The implemented dynamic compensation control law is indicated in (29).

$$\boldsymbol{\mu}_{\mathbf{ref}} = \bar{\mathbf{M}}(\chi, a, b)\sigma + \bar{\mathbf{C}}(\chi, \boldsymbol{\mu})\boldsymbol{\mu} \tag{29}$$

where σ defines the dynamic compensation auxiliary variable, which is defined as follows:

$$\sigma = \dot{\boldsymbol{\mu}}_{\boldsymbol{c}} + \mathbf{K_4} \tanh\left(\mathbf{K_4^{-1}K_3}\tilde{\boldsymbol{\mu}}\right) \tag{30}$$

$\dot{\boldsymbol{\mu}}_{\boldsymbol{c}}$ defines the acceleration of the kinematic control reference $\boldsymbol{\mu}_c$; $\mathbf{K_3}$ is a matrix that weight the convergence of the velocity control errors and $\mathbf{K_4}$ is a diagonal matrix that delimits instantaneous velocities.

Dynamic Controller Stability Analysis

This work uses the Lyapunov criterion to guarantee the stability of the proposed dynamic compensation, combining Equations (29) and (21) is possible to obtain the following closed-loop expression:

$$\dot{\tilde{\boldsymbol{\mu}}} = -\mathbf{K_4} \tanh\left(\mathbf{K_4^{-1}K_3}\tilde{\boldsymbol{\mu}}\right) \tag{31}$$

Therefore, the Lyapunov candidate function is proposed as follows:

$$\mathbf{V}_d = \frac{1}{2}\tilde{\mu}^{\mathsf{T}}\mathbf{\Gamma}\tilde{\mu} \tag{32}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{4x4}$ is a positive definite diagonal matrix. The time derivative of the Lyapunov candidate function along the system's trajectories can be defined as

$$\dot{\mathbf{V}}_d(\tilde{\mu}) = -\tilde{\mu}^{\mathsf{T}}\mathbf{\Gamma}\mathbf{K_4}\tanh\left(\mathbf{K_4^{-1}}\mathbf{K}_3\tilde{\mu}\right) \tag{33}$$

According to the Lyapunov stability criterion, the stability of the system is guaranteed because $\dot{\mathbf{V}}_d(\tilde{\mu}) < 0$, that is, $\tilde{\mu} \to 0$ since $t \to \infty$ as long as $\mathbf{K}_3 > 0$ and $\mathbf{K}_4 > 0$.

3.3. NMPC Controller

This work formulates the NMPC modifying the cost function into nonlinear programming formulation (NLP) to generate the optimal velocities that keep the quadrotor on the desired trajectory considering the kinematics and dynamics constraints along the entire finite horizon of prediction; in addition, the kinematics, dynamics, and control actions that include the rate of change of the control velocities are defined as constraints. Finally, this work considers a Lyapunov candidate function as a hard constraint to guarantee the stability of the system. The general scheme of the proposed controller is shown in Figure 5.

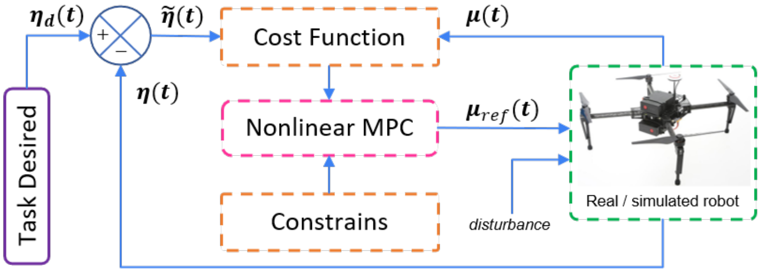

Figure 5. General scheme of the NMPC.

The prediction of the generalized nonlinear kinematic-dynamical system is defined as

$$\dot{x}(l|t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) \tag{34}$$

$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\mu}_{\mathbf{ref}}(t)) := \boldsymbol{A}\boldsymbol{x}(l|t) + \boldsymbol{B}\boldsymbol{\mu}_{ref}(l|t)$$

$$A = \begin{bmatrix} \mathbf{0}_{4\times4} & \mathbf{J}(\psi) \\ \mathbf{0}_{4\times4} & \bar{\mathbf{M}}^{-1}\bar{\mathbf{C}} \end{bmatrix},\ B = \begin{bmatrix} \mathbf{0}_{4x4} \\ \bar{\mathbf{M}}^{-1} \end{bmatrix}$$

where $\boldsymbol{x}(l|t) = \begin{bmatrix} \boldsymbol{\eta}^{\mathsf{T}} & \boldsymbol{\mu}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{X}$ and $\boldsymbol{\mu}_{\mathbf{ref}}(l|t) \in \mathbb{U}$ are the state and input of the system, and $l \in [t, t+T]$ is the evolution of the instantaneous value throughout the finite prediction interval T considering the instant measure t. Thus, this work uses an intermediate cost function $\boldsymbol{\ell}_t$ defined as

$$\boldsymbol{\ell}_t(\tilde{\boldsymbol{\eta}}(l|t), \boldsymbol{\mu}(l|t)) = \frac{1}{2}(\tilde{\boldsymbol{\eta}}(l|t)^{\mathsf{T}}\mathbf{Q}\tilde{\boldsymbol{\eta}}(l|t) + \boldsymbol{\mu}(l|t)^{\mathsf{T}}\mathbf{R}\boldsymbol{\mu}(l|t)) \tag{35}$$

In the last instant of time, the final prediction cost function $\boldsymbol{\ell}_f$ is defined as

$$\boldsymbol{\ell}_f(\tilde{\boldsymbol{\eta}}(t+T)) = \frac{1}{2}\tilde{\boldsymbol{\eta}}(t+T)^{\mathsf{T}}\mathbf{Q}\tilde{\boldsymbol{\eta}}(t+T) \tag{36}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are an arbitrary positive definite gain matrix for error and input system, respectively. The MPC is defined as the solution to the nonlinear optimal control problem (NOCP):

$$\min_{\boldsymbol{\eta}(.),\boldsymbol{\mu}(.)} \int_{t}^{t+T} \boldsymbol{\ell}_t(\tilde{\boldsymbol{\eta}}(l|t), \boldsymbol{\mu}(l|t))dt + \boldsymbol{\ell}_f(\tilde{\boldsymbol{\eta}}(t+T)) \tag{37}$$

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

$$\boldsymbol{x}(0) = \boldsymbol{x}_{\mathbf{0}} \tag{39}$$

$$|\boldsymbol{\mu}_{\mathbf{ref}}(i+1|t) - \boldsymbol{\mu}_{\mathbf{ref}}(i|t)| \leq \Delta\boldsymbol{\mu}_{\mathbf{ref}} \tag{40}$$

$$\boldsymbol{\mu}_{\mathbf{ref}}(l|t) \in [\boldsymbol{\mu}_{\mathbf{ref}_{min}}, \boldsymbol{\mu}_{\mathbf{ref}_{max}}] \tag{41}$$

$$\boldsymbol{\mu}_{\mathbf{ref}}(l|t) \in \mathbb{U} \quad \forall t \in [0, N-1] \tag{42}$$

$$\boldsymbol{x}(l|t) \in \mathbb{X} \quad \forall t \in [0, N] \tag{43}$$

$$h_f(\tilde{\boldsymbol{\mu}}) \leq 0 \tag{44}$$

where the NOCP (37) is solved using the direct multiple shooting method and considering the initial conditions (39). Equation (38) defines the system dynamics considered as a constraint. Equations (40)–(42) define the inputs constraints, where $\Delta\boldsymbol{\mu}_{\mathbf{ref}}$, $\boldsymbol{\mu}_{\mathbf{ref}_{min}}$ and $\boldsymbol{\mu}_{\mathbf{ref}_{max}}$ represents the rate of change, the maximum and minimum values of the control actions, respectively. Equation (43) defines the state constraints.

Finally, Equation (44) considers the time derivative of the Lyapunov candidate function (33) defined as the constraint $h_f(\tilde{\boldsymbol{\mu}}) = -\tilde{\boldsymbol{\mu}}^{\top}\boldsymbol{\Gamma}\mathbf{K_4}\tanh\left(\mathbf{K_4^{-1}K_3}\tilde{\boldsymbol{\mu}}\right)$ that guarantees the system's stability.

4. Results

This section illustrates the simulation and real-world experimental results to compare each controller's performance. The experiments were performed considering the following conditions: *(i) Simulations Experiments*, this section uses the simulation software provided by DJI with the nominal model without disturbances; *(ii) Real World Experiments* the considerations for this section include a the Lissajous trajectory with low and high velocities where aerodynamics effects, latency, and uncertainties are inevitable guaranteeing a fair comparative study; additionally, the aerial vehicle used during the experiments was DJI Matrice 100 by DJI. The comparative study considers the following metrics: tracking accuracy, computational time, and robustness. Throughout the experiments, this work uses $\boldsymbol{\eta}$ and $\boldsymbol{\eta}_d$ to represent the movement of the DJI Matrice 100 aerial robot and the desired trajectory, respectively.

4.1. Simulation Experiments Results

This section illustrates the results of the controllers developed in Section 3; simulation experiments allow the user to tune the parameters presented in each controller and avoid problems in real-world experiments. The reference trajectory $\boldsymbol{\eta}_d = \begin{bmatrix} \eta_{x_d} & \eta_{y_d} & \eta_{z_d} & \eta_{\psi_d} \end{bmatrix}^{\top}$, initial conditions $\boldsymbol{\eta}_o = \begin{bmatrix} \eta_{x_o} & \eta_{y_o} & \eta_{z_o} & \eta_{\psi_o} \end{bmatrix}^{\top}$ $\boldsymbol{\mu}_o = \begin{bmatrix} \mu_{l_o} & \mu_{m_o} & \mu_{n_o} & \omega_o \end{bmatrix}^{\top}$ conducted in these experiments are presented in Table 2. Finally, to measure the trajectory tracking performance, this work defined the control error as $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$ considering only positional error; orientation errors were not considered.

Table 2. Reference trajectory and initial conditions used in simulation experiments.

Initial Positions [m]-[rad]	Initial Velocities [m/s]-[rad/s]	Reference Trajectory [m]-[rad]
$\eta_{x_o} = 0$	$\mu_{l_o} = 0$	$\eta_{x_d} = 3\sin(0.4t) + 3$
$\eta_{y_o} = 0$	$\mu_{m_o} = 0$	$\eta_{y_d} = 3\sin(0.8t)$
$\eta_{z_o} = 1$	$\mu_{n_o} = 0$	$\eta_{z_d} = 1.5\sin(0.2t) + 8$
$\eta_{\psi_o} = 0$	$\omega_o = 0$	$\eta_{\psi_d} = 0$

4.1.1. Inverse Differential Kinematics

In this section, the inverse differential kinematics (25) is implemented, where the desired trajectory and initial conditions are shown in Table 2. Additionally, this control structure uses the controller parameters presented in Table 3. The control parameters were selected considering the limits in the control maneuverability velocities $\mathbf{K}_2$ and the fair pondering of the control error $\mathbf{K}_1$.

Table 3. Proposed values for the inverse differential kinematics controller.

Parameters	Values	Parameters	Values
$\mathbf{K}_1$	$diag(1) \in \mathbb{R}^{4\times4}$	$\mathbf{K}_2$	$diag(2\ 2\ 2\ 10) \in \mathbb{R}^{4\times4}$

The results of this experiment are illustrated in Figure 6a,b; the quadrotor follows the desired trajectory; furthermore, the evolution of the system has been associated with the Euclidean norm $\|\tilde{\boldsymbol{\eta}}\|$ of the control error, which is helpful to illustrate the deviation of the system during the execution of the desired trajectory.

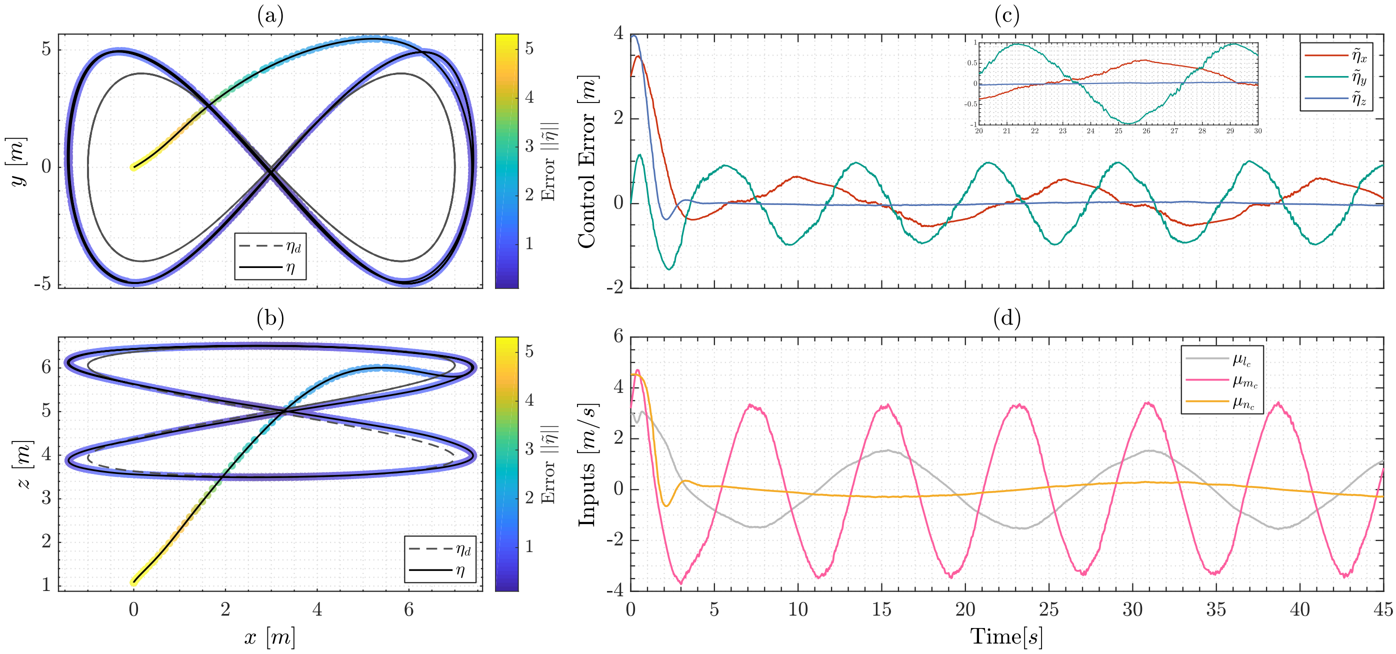

Figure 6. Simulation tracking results considering a Lissajous trajectory using the inverse differential kinematic controller; where (**a**) shows the evolution of the system in the plane $\{\mathbf{I}_x, \mathbf{I}_y\}$ and (**b**) considering the plane $\{\mathbf{I}_x, \mathbf{I}_z\}$; finally (**c**,**d**) illustrate the control error and actions, respectively.

However, this formulation shows a steady state error associated with the lack of dynamics in the control structure Figure 6c. A close-up of the control errors shows that these values are bounded, achieving a final characteristic error with a max value of $\tilde{\eta}_x < 0.6\ [m]$, $\tilde{\eta}_y < 1\ [m]$ and $\tilde{\eta}_z < 0.1\ [m]$ respectively for each axis.

Finally, the control actions $(\mu_{l_c}, \mu_{m_c}, \mu_{n_c})$ are shown in Figure 6d, which were generated by the presented controller and guaranteed smooth values; however, these signals cannot ensure the convergence of the system to the reference trajectory.

4.1.2. Inverse Dynamic Compensation

The implementation of the inverse dynamic compensation (29) is presented in this section; moreover, to obtain fair comparative results, this work uses the same desired trajectory and initial condition defined in Table 2. Additionally, the parameters of this control structure are defined in Table 4. The control parameters of the dynamic compensation controller were selected as identity matrices; because of this, the control error values do not change, and we can ensure a fair comparison.

Table 4. Proposed values for the inverse dynamic compensation controller.

Parameters	Values	Parameters	Values
$\mathbf{K}_3$	$diag(1) \in \mathbb{R}^{4\times4}$	$\mathbf{K}_4$	$diag(1) \in \mathbb{R}^{4\times4}$

The results using the inverse dynamic compensation are presented in Figure 7a,b; this controller shows a better performance with more accuracy than the results presented in Figure 6 due to the controller can include the dynamics of the system in the control formulation. Additionally, the movement of the quadrotor platform based on measures of the system introduces the Euclidean norm showing the control error values along the trajectory.

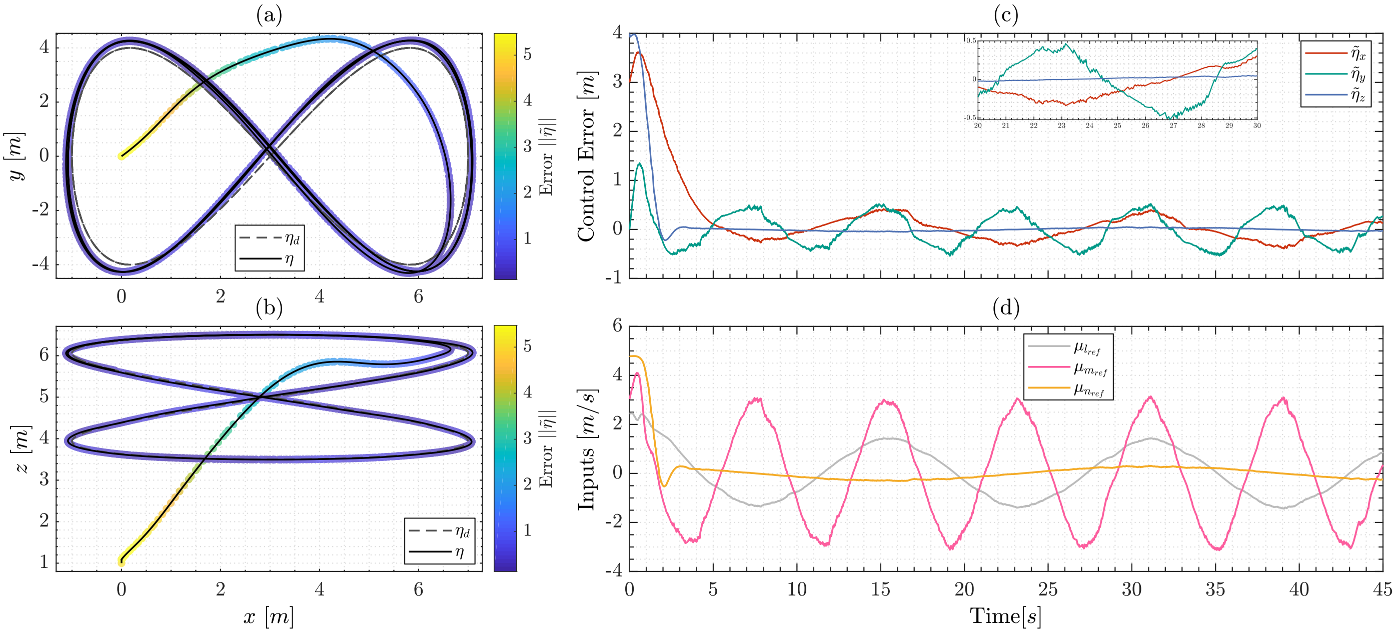

Figure 7. Simulation tracking results considering a Lissajous trajectory using the inverse dynamic compensation controller; where (**a**) shows the evolution of the system in the plane $\{\mathbf{I}_x, \mathbf{I}_y\}$ and (**b**) over the plane $\{\mathbf{I}_x, \mathbf{I}_z\}$; finally (**c**,**d**) represent the control error and actions, respectively.

A close-up of the control errors (see Figure 7c) shows boundaries with maximum values of $\tilde{\eta}_x < 0.3\ [m]$, $\tilde{\eta}_y < 0.5[m]$ and $\tilde{\eta}_z < 0.1\ [m]$ for each axis, respectively. The control signals generated by the controller $(\mu_{l_{ref}}, \mu_{m_{ref}}, \mu_{n_{ref}})$ are presented in Figure 7d; where these signals show a non-smooth behavior compared to the formulation presented before.

4.1.3. Nonlinear Model Predictive Control

This section presents the implementation of the NMPC controller proposed in (37); additionally, this work uses direct multiple shooting and nonlinear programming formulation to include the internal states $\boldsymbol{\eta}$ and the control actions $\boldsymbol{\mu}_{\mathbf{ref}}$ as optimization variables guarantying computational efficiency. This work uses CasADI [46] to solve the optimization problem guaranteeing real-time solutions.

This work defines the values of the positive define matrices presented in Table 5; these values were obtained through experiments to improve the performance of the controller scheme. Additionally, the boundaries in the control actions $(\boldsymbol{\mu}_{\mathbf{ref}_{min}}, \boldsymbol{\mu}_{\mathbf{ref}_{max}})$ are similar to those considered in the inverse kinematic controllers; moreover, the matrices $\mathbf{Q}$ and $\mathbf{R}$ guarantee a fair comparison and smooth control variables under this control formulation.

Table 5. Proposed values for the NMPC scheme.

Parameters	Values	Parameters	Values
$\mathbf{Q}$ $\boldsymbol{\mu}_{\mathbf{ref}_{min}}$ $T[s]$	$diag(0.9) \in \mathbb{R}^{4\times4}$ $-[2.4, 2.4, 2.4, 11.5]$ 1	$\mathbf{R}$ $\boldsymbol{\mu}_{\mathbf{ref}_{max}}$	$diag(0.1) \in \mathbb{R}^{4\times4}$ $[2.4, 2.4, 2.4, 11.5]$

The movements of the quadrotor using this formulation are shown in Figure 8a and Figure 8b; a close-up of the control errors (see Figure 8c) indicates boundaries with maximum values of $\tilde{\eta}_x < 0.3\ [m]$, $\tilde{\eta}_y < 0.2[m]$ and $\tilde{\eta}_z < 0.1\ [m]$, guaranteeing that the aerial vehicle tracks the desired reference trajectory more accurately than the control schemes proposed in (25) and (29), demonstrating the advantages of the formulation compared with non-predictive formulations.

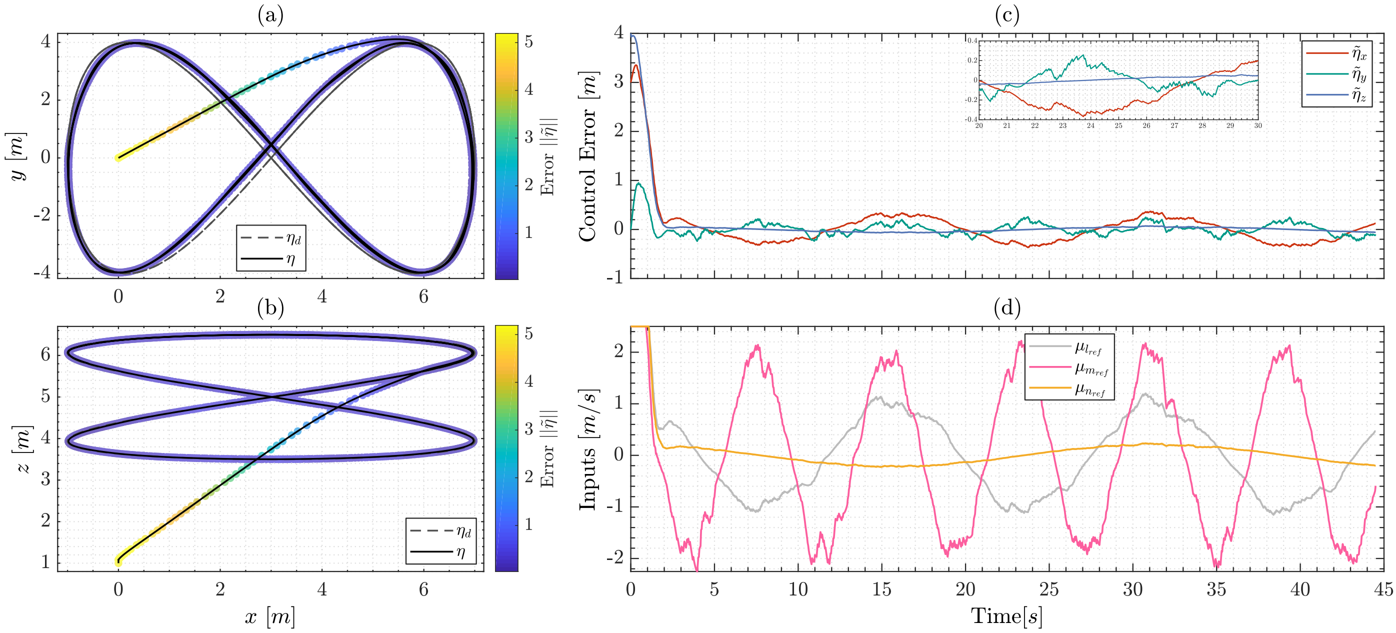

Figure 8. Simulation tracking results considering a Lissajous trajectory using the NMPC controller; where (**a**) shows the evolution of the system in the plane $\{\mathbf{I}_x, \mathbf{I}_y\}$ and (**b**) over the plane $\{\mathbf{I}_x, \mathbf{I}_z\}$; finally (**c**,**d**) represent the control error and actions, respectively.

Finally, the control signals generated by the NMPC formulation $(\mu_{l_{ref}}, \mu_{m_{ref}}, \mu_{n_{ref}})$ are shown in Figure 8d; the control values are bounded considering the specifications in Table 5, this is one of an essential feature of this controller due to the systems incorporate limitation in control actions and internal system states.

4.2. Comparative of Simulations Experiments

This section presents comparative results where each controller was evaluated using the information presented in the previous section. In the simulation experiments, this work does not consider unmodeled dynamics and uncertainties to ensure that each controller has the accurate formulation to control the system.

In the following comparison, this work uses the root mean square error (RMSE) of the positional error as a metric to compare each controller; additionally, we use each controller's average computational time to compare the required time of each formulation to generate the solution.

Figure 9a compares the Euclidean norm of each controller during the execution of the desired task, where $\|\tilde{\boldsymbol{\eta}}\|_{Kinematic}$, $\|\tilde{\boldsymbol{\eta}}\|_{Dynamic}$ and $\|\tilde{\boldsymbol{\eta}}\|_{NMPC}$ illustrated the evolution of the control error, demonstrating the better performance of the NMPC formulation. Figure 9b shows the RMSE of each controller, the NMPC has a better performance, especially in the Y axis, where the trajectory presents the more aggressive behavior.

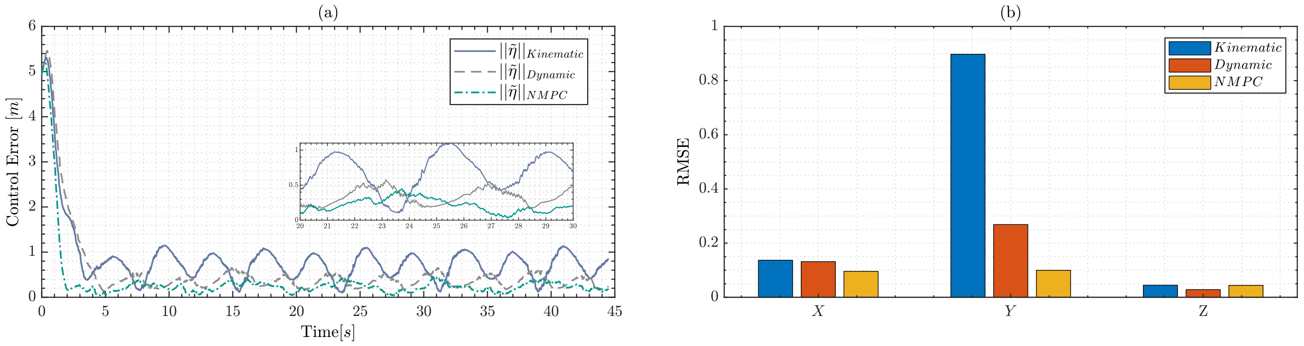

Figure 9. Comparative results of the proposed controllers during the simulation experiments; where (**a**) illustrates the norm of the control error during the simulation of each controller (**b**) shows the RMSE of each controller along the simulation.

Simulation experiments were carried out in the simulator provided by DJI; moreover, the desktop computer presents the following hardware features: Intel processor i7-7700HQ and CPU 2.80 GHzx8. The control structures were developed in the onboard computer of the aerial vehicle (Jetson nano 4 GB core), where the programming selected programming language is Python due to the famous general-purpose structure. Table 6 compares the average computational time for each controller. As expected, the NMPC formulation requires more time to compute the control signal; however, the excessive computational time compared with the other formulation does not mean that this controller cannot be implemented with a sufficiently high frequency.

Table 6. Average computational time for each controller in simulations.

	Kinematic	**Dynamic**	**NMPC**
Avg. dt [ms]	0.054	0.141	17.54

4.3. Real-World Experiments Results

This work develops many experiments using the commercial aerial platform DJI Matrice 100 to compare the performance of each controller proposed in Section 3.

Each controller was implemented on the onboard computer; a more detailed explanation of the hardware in the aerial vehicle is presented in Figure 10 considering the onboard computer Jetson nano 4 GB core module.

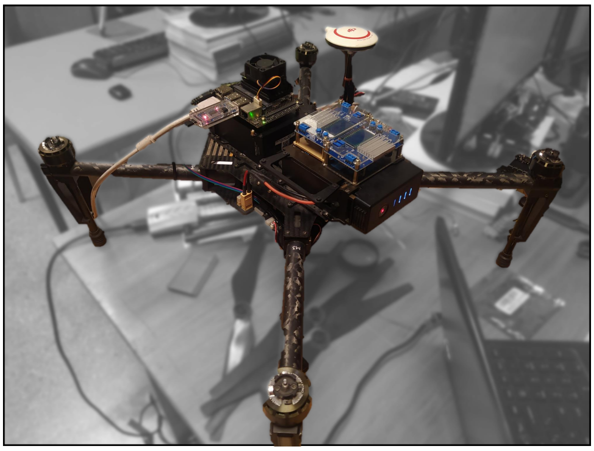

Figure 10. DJI Matrice 100 Experimental Platform.

The desired trajectory and initial conditions used during the experimental results are presented in Table 7; due to the extra hardware presented in the aerial vehicle, such as the on-board computer, the selected trajectory presents lower velocities than the trajectory in Table 2 to maintain the system's integrity.

Table 7. Reference trajectory and initial conditions used in real-world conditions.

Initial Positions [m]-[rad]	Initial Velocities [m/s]-[rad/s]	Reference Trajectory [m]-[rad]
$\eta_{x_o} = 2$ $\eta_{y_o} = 0$ $\eta_{z_o} = 1$ $\eta_{\psi_o} = 0$	$\mu_{l_o} = 0$ $\mu_{m_o} = 0$ $\mu_{n_o} = 0$ $\omega_o = 0$	$\eta_{x_d} = 3\sin(0.16t) + 3$ $\eta_{y_d} = 3\sin(0.32t)$ $\eta_{z_d} = 1.5\sin(0.2t) + 8$ $\eta_{\psi_d} = 0$

4.3.1. Inverse Differential Kinematics

This section presents the results using the controller proposed in Equation (25); we consider the same control parameters presented in Table 3 to present a fair comparison. Figure 11a,b show the evolution of the quadrotor during the experiment; furthermore, the control error is presented in Figure 11c with boundaries where the maximum values are $\tilde{\eta}_x < 0.2\ [m]$, $\tilde{\eta}_y < 0.5\ [m]$ and $\tilde{\eta}_z < 0.5\ [m]$. Finally, the control actions $(\mu_{l_c}, \mu_{m_c}, \mu_{n_c})$ are shown in Figure 11d, which present a smooth behavior.

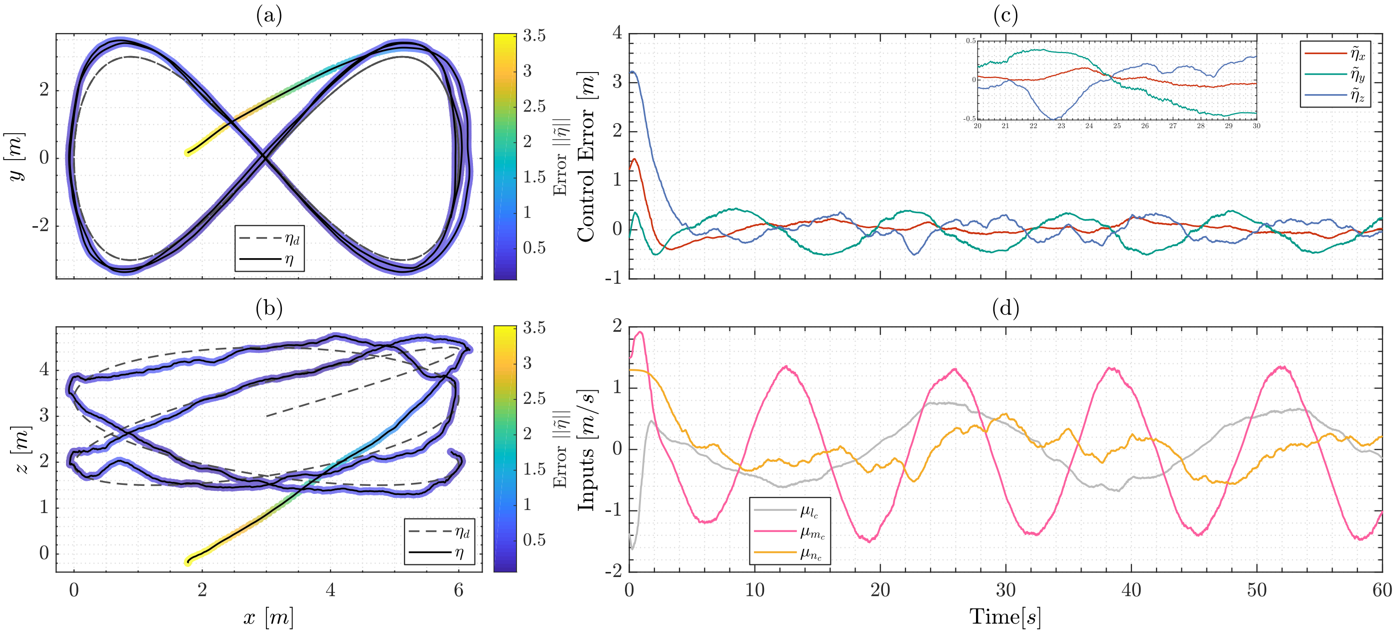

Figure 11. Real-world tracking results considering the inverse differential kinematic controller; where (**a**,**b**) show the evolution of the system; finally (**c**,**d**) represent the control error and actions, respectively.

4.3.2. Inverse Dynamic Compensation

The section presents the implementation of inverse dynamic compensation (29) in real-world experiments; additionally, to get fair comparative results, this work uses the same control parameters defined in Table 4.

The results using the inverse dynamic compensation are presented in Figure 12a,b; where the results show a better performance with more accuracy than those presented in Figure 11. Finally, Figure 12c shows the control error results, where these values present boundaries with maximum values of $\tilde{\eta}_x < 0.3\ [m]$, $\tilde{\eta}_y < 0.3\ [m]$ and $\tilde{\eta}_z < 0.35\ [m]$, finally the control signals $(\mu_{l_{ref}}, \mu_{m_{ref}}, \mu_{n_{ref}})$ are presented in Figure 12d.

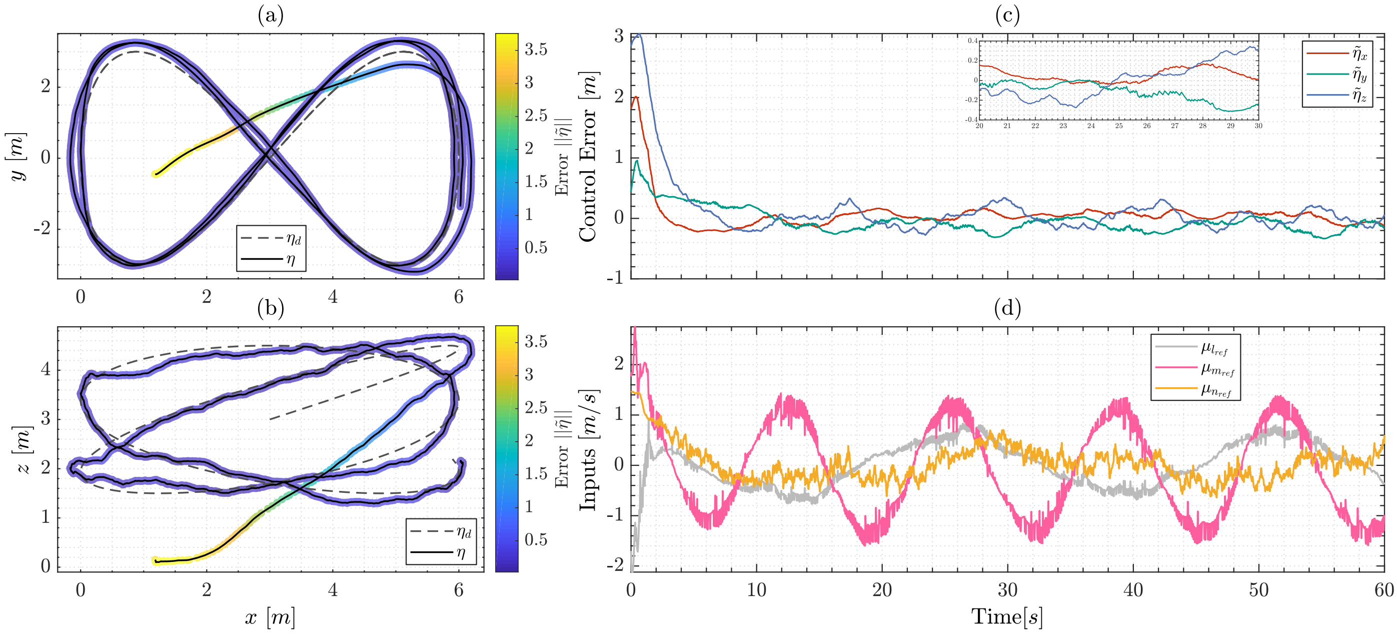

Figure 12. Real-world tracking results considering the inverse dynamic compensation; where (**a**,**b**) show the evolution of the system; finally (**c**,**d**) represent the control error and actions.

4.3.3. Nonlinear Model Predictive Control

This section presents the implementation of the proposed NMPC controller (37) in real-world experiments; due to uncertainties and model mismatch, this section demonstrates the robustness and precision of the NMPC.

The evolution of the quadrotor using NMPC is shown in Figure 13a,b; although of the better performance of this control scheme a close-up of the control errors (see Figure 13c) shows boundaries with maximum values of $\tilde{\eta}_x < 0.1\ [m]$, $\tilde{\eta}_y < 0.15[m]$ and $\tilde{\eta}_z < 0.2\ [m]$. The predictive control formulation guarantees that the quadrotor tracks the desired reference trajectory more accurately than the control schemes proposed in (25) and (29). The NMPC formulation uses the same control parameters shown in Table 5.

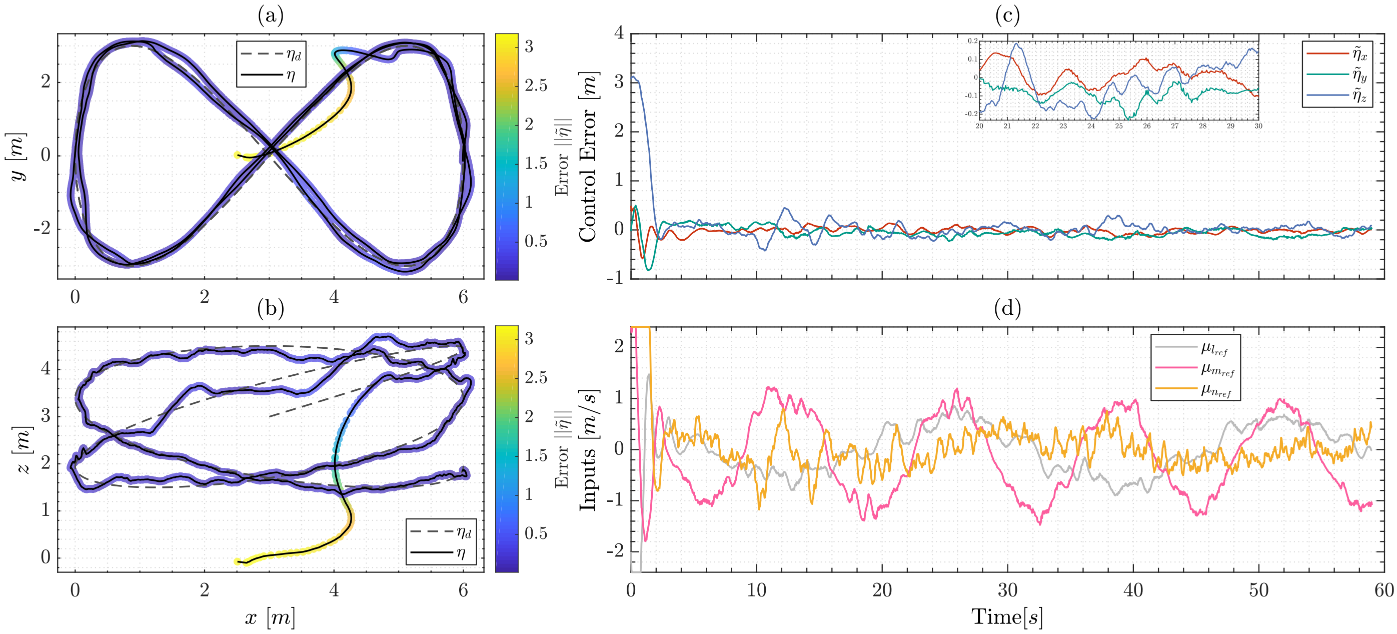

Figure 13. Real-world tracking results considering the NMPC controller; where (**a**,**b**) show the evolution of the system; finally (**c**,**d**) represent the control error and actions, respectively.

Finally, the control signals generated by the optimal policy of the optimization problem $(\mu_{l_{ref}}, \mu_{m_{ref}}, \mu_{n_{ref}})$ are shown in Figure 13d; although of the disturbances and uncertainties the control values are still bounded from the disturbances and uncertainties, it is an essential characteristic of this controller that generates robustness properties in the system.

4.4. Comparative of Real-World Experiments

The metrics presented in Section 4.2 were analyzed in this Section considering the RMSE of the control error vector. To demonstrate the efficacy of each controller during the experiments, the system experimented with disturbances produced by external forces such as wind, a better representation of this external disturbance is shown in Figure 14. Finally, we use each controller's average computational time to compare the required time of each formulation.

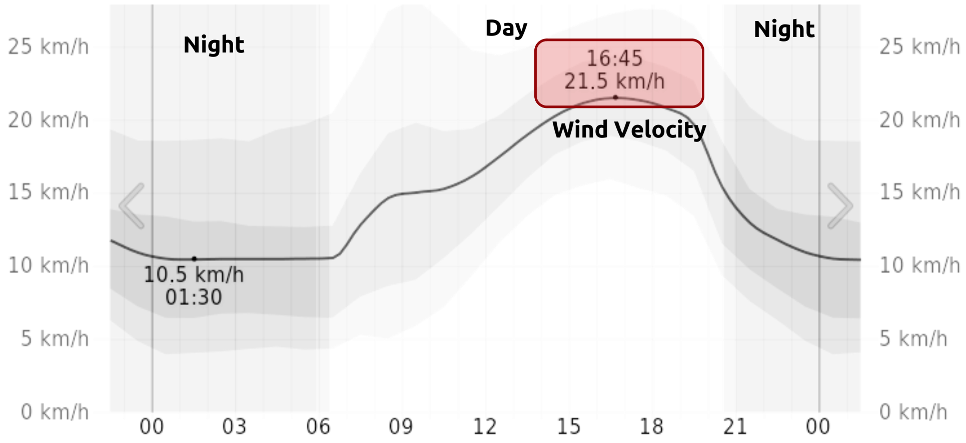

Figure 14. Wind velocity during the real-world experiments.

Figure 15a compares the Euclidean norm of each controller during the real-world experimental results; the NMPC presents a better performance associated with more accuracy than the non-predictive formulations.

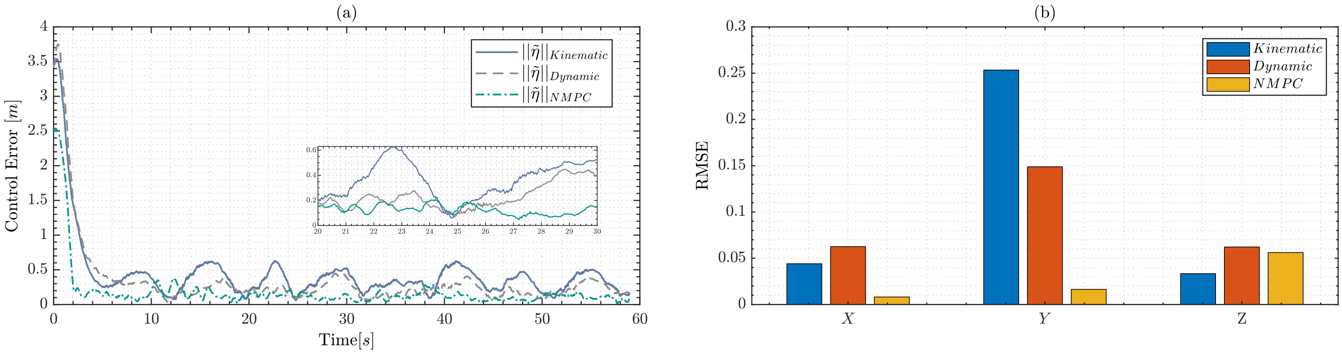

Figure 15. Comparative results of the proposed controllers during the real-world experiments; where (**a**) illustrates the Euclidean norm of the control error during the execution of each controller (**b**) shows the RMSE of each controller.

Figure 15b shows the RMSE of each controller; due to the disturbances and model mismatch presented during the experiment, the performance of each controller was affected; however, NMPC remains a better performance specifically in the Y and X axis where the trajectory presents an aggressive behavior.

Real-world experiments were carried out using the aerial vehicle Matrice 100 by DJI, which incorporates the onboard computer Jetson nano 4GB with the following hardware features Quad-core ARM Cortex-A57 MPCore processor. Additionally, the optimization problem was solved through multiple shooting schemes and using IPOPT, short for "Interior Point Optimizer executed in CasADi [46].

As expected, the NMPC formulation requires more time to compute the control signal (see Table 8); however, this control formulation can be implemented with a sufficiently high frequency to achieve the agile trajectories used in this work.

Table 8. Average computational time for each controller.

	Kinematic	**Dynamic**	**NMPC**
Avg. dt [ms]	0.075	0.26	25.36

5. Discussion

NMPC controllers have recently demonstrated their validity in multirotor air vehicles, but only some authors have presented their performance in simulation, and real experimentation [17]. In this work, an NMPC control has been selected as the best option based on trajectory tracking, where the NMPC is robust in positioning errors and works better at high velocities, and the small radius of curvature of the reference trajectories [50]. On the other hand, baseline controllers have always shown adequate performance for multirotor trajectory tracking control [26]. However, the contrast between these two controllers is still limited in the literature; in [51], the authors compare a predictive contour control for time-optimal with respect to state-of-the-art and with respect to a professional pilot, evidencing better performance for the predictive control.

Non-predictive baseline feedback controllers such as inverse differential kinematics and inverse dynamic compensation have demonstrated accuracy and computational efficiency [52]. This justifies the comparison of our NMPC proposal with this class of controllers that are widely used for UAVs. In this case, higher trajectory-following accuracy is achieved with the NMPC control, which is more evident in real-world experimentation than in simulation, possibly due to the effect of extrinsic components of the system. Regarding the UAV used in this study, the DJI Matrice 100 UAV has proven to be a viable option for evaluating advanced control algorithms [53]. This is because it allows testing both in simulation and experimentation.

Perturbations are an essential component in the modeling and control of a UAV system, and simulation results have demonstrated the efficiency and robustness of nonlinear controllers [29,32]. However, only some works with experimental results analyze disturbances in NMPC controllers. In [45], the authors explore the performance of an MPC controller against aerodynamic disturbances in trajectory tracking. In [42], an NMPC control is used for trajectory tracking applied to micro aerial vehicles, demonstrating good disturbance rejection capacity, stepped response, tracking performance, and slightly less computational effort than the linear MPC control.

Although the mean computation times of our NMPC control are higher than the baseline controllers, with 17.54 ms in the simulation and 25.36 ms in experimentation, these times are lower than the results of [54], which present an NMPC which enables real-time solutions with a sampling time of 50 ms to avoid collisions with dynamic obstacles with UAVs. Some works have achieved similar times in manipulator control with NMPC but have only been evaluated in simulation environments, leaving the controller performance in real environments in doubt [55]. Other works have achieved shorter computation times in NMPC control of electric vehicles, but these results are based on hardware loop tests [56]. In any case, our findings contribute to applying NMPC controllers in UAVs with precision registration in trajectory tracking and computation times that examine the control system's efficiency.

6. Conclusions

This work developed the instantaneous kinematics and simplified dynamic model of the aerial vehicle Matrice 100, based on linear and angular velocities as inputs of the aerial vehicle. Because of this, the non-predictive control formulations considered in this work are the inverse differential kinematics and the inverse dynamic compensation; on the other hand, the predictive formulation selected is the NMPC. This work compares the control structures presented before considering the tracking of agile trajectories; simulations and real-world experiments were performed to evaluate the performance in terms of accuracy and computational efficiency, which are helpful features that could benefit future works.

The trajectory used for the comparison is known as the Lissajous trajectory, which presented an aggressive behavior with velocity variations where the controllers needed to demonstrate accuracy and efficiency. The real-world experiments showed the accuracy properties of the NMPC against disturbances; specifically, at the time of the experimentation, the system presented an average wind the velocity of approximately 21.5 km/h; additionally, the comparative results showed the superiority of the NMPC formulation clearly expressed in the RMSE calculation. Therefore, this work reports the advantages and disadvantages of each formulation analyzing accuracy; because of this, this work suggests the NMPC as the best option based on the comparison under trajectory tracking, which mainly works better at high reference velocities. However, although the accuracy property demonstrated by the NMPC compared with the feedback control approaches, the computational time is highly demanding using the onboard computers. Because of this, it could be unfeasible for miniature air vehicles.

Future works will develop considering soft constraints and Sequential quadratic programming (SQP) to improve the computational time of the NMPC formulation.

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

Funding: The APC was funded by Universidad Indoamérica (Ecuador), funding code INV-0014-013-002.

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 Bryan S. Guevara.

Acknowledgments: The authors would like to thank the Universidad Indoamérica; Universidad de las Fuerzas Armadas ESPE; 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 "Educación 4.0 aplicada a la enseñanza de competencias STEAM" developed by Universidad Indoamérica.

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

References

1. Kendoul, F. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *J. Field Robot.* **2012**, *29*, 315–378. [CrossRef]
2. Mu, B.; Zhang, K.; Shi, Y. Integral Sliding Mode Flight Controller Design for a Quadrotor and theA pplication in a Heterogeneous Multi-Agent System. *IEEE Trans. Ind. Electron.* **2017**, *64*, 9389–9398. [CrossRef]
3. Liang, X.; Fang, Y.; Sun, N.; Lin, H. A novel energy-coupling-based hierarchical control approach for unmanned quadrotor transportation systems. *IEEE ASME Trans. Mechatronics* **2019**, *24*, 248–259. [CrossRef]

4. Xu, L.X.; Ma, H.J.; Guo, D.; Xie, A.H.; Song, D.L. Backstepping Sliding-Mode and Cascade Active Disturbance Rejection Control for a Quadrotor UAV. *IEEE ASME Trans. Mechatronics* **2020**, *25*, 2743–2753. [CrossRef]
5. Lu, Q.; Ren, B.; Parameswaran, S. Uncertainty and Disturbance Estimator-Based Global Trajectory Tracking Control for a Quadrotor. *IEEE ASME Trans. Mechatronics* **2020**, *25*, 1519–1530. [CrossRef]
6. Serrano, M.E.; Gandolfo, D.C.; Scaglia, G.J. Trajectory tracking controller for unmanned helicopter under environmental disturbances. *ISA Trans.* **2020**, *106*, 171–180. [CrossRef]
7. Ou, J.; Guo, X.; Lou, W.; Zhu, M. Quadrotor Autonomous Navigation in Semi-Known Environments Based on Deep Reinforcement Learning. *Remote Sens.* **2021**, *13*, 4330. [CrossRef]
8. Xu, B.; Wang, W.; Falzon, G.; Kwan, P.; Guo, L.; Sun, Z.; Li, C. Livestock classification and counting in quadcopter aerial images using Mask R-CNN. *Int. J. Remote Sens.* **2020**, *41*, 8121–8142. [CrossRef]
9. 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.* **2016**, *40*, 1059–1078. [CrossRef]
10. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding horizon path planning for 3D exploration and surface inspection. *Auton. Robot.* **2018**, *42*, 291–306. [CrossRef]
11. Loianno, G.; Scaramuzza, D. Special issue on future challenges and opportunities in vision-based drone navigation. *J. Field Robot.* **2020**, *37*, 495–496. [CrossRef]
12. Rajendran, S.; Srinivas, S. Air taxi service for urban mobility: A critical review of recent developments, future challenges, and opportunities. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *143*, 102090. [CrossRef]
13. Hanover, D.; Foehn, P.; Sun, S.; Kaufmann, E.; Scaramuzza, D. Performance, precision, and payloads: Adaptive nonlinear MPC for quadrotors. *IEEE Robot. Autom. Lett.* **2022**, *7*, 690–697. [CrossRef]
14. Mohta, K.; Watterson, M.; Mulgaonkar, Y.; Liu, S.; Qu, C.; Makineni, A.; Saulnier, K.; Sun, K.; Zhu, A.; Delmerico, J.; et al. Fast, autonomous flight in GPS-denied and cluttered environments. *J. Field Robot.* **2018**, *35*, 101–120. [CrossRef]
15. Foehn, P.; Kaufmann, E.; Romero, A.; Penicka, R.; Sun, S.; Bauersfeld, L.; Laengle, T.; Cioffi, G.; Song, Y.; Loquercio, A.; et al. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Sci. Robot.* **2022**, *7*, 6259. [CrossRef] [PubMed]
16. Loquercio, A.; Kaufmann, E.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Learning high-speed flight in the wild. *Sci. Robot.* **2021**, *6*. [CrossRef] [PubMed]
17. 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]
18. Foehn, P.; Romero, A.; Scaramuzza, D. Time-Optimal Planning for Quadrotor Waypoint Flight. *Sci. Robot.* **2021**, *6*. [CrossRef]
19. Torrente, G.; Kaufmann, E.; Fohn, P.; Scaramuzza, D. Data-Driven MPC for Quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3769–3776. [CrossRef]
20. Kamel, M.; Stastny, T.; Alexis, K.; Siegwart, R. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. *Stud. Comput. Intell.* **2017**, *707*, 3–39. [CrossRef]
21. Acosta Núñez, J.F.; Ortiz, V.H.A.; Peces, G.G.d.R.; Salas, J.G. Energy-Saver Mobile Manipulator Based on Numerical Methods. *Electronics* **2019**, *8*, 1100. [CrossRef]
22. 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]
23. Gandolfo, D.C.; Salinas, L.R.; Brandão, A.; Toibero, J.M. Stable Path-Following Control for a Quadrotor Helicopter Considering Energy Consumption. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 1423–1430. [CrossRef]
24. 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. [CrossRef]
25. Nascimento, T.P.; Saska, M. Position and attitude control of multi-rotor aerial vehicles: A survey. *Annu. Rev. Control* **2019**, *48*, 129–146. [CrossRef]
26. Lee, H.; Kim, H.J. Trajectory tracking control of multirotors from modelling to experiments: A survey. *Int. J. Control. Autom. Syst.* **2017**, *15*, 281–292. [CrossRef]
27. Martins, L.; Cardeira, C.; Oliveira, P. Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor. *IFAC-PapersOnLine* **2019**, *52*, 176–181. [CrossRef]
28. Shakeel, T.; Arshad, J.; Jaffery, M.H.; Rehman, A.U.; Eldin, E.T.; Ghamry, N.A.; Shafiq, M. A Comparative Study of Control Methods for X3D Quadrotor Feedback Trajectory Control. *Appl. Sci.* **2022**, *12*, 9254. [CrossRef]
29. 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]
30. Shen, Z.; Tsuchiya, T. Singular Zone in Quadrotor Yaw–Position Feedback Linearization. *Drones* **2022**, *6*, 84. [CrossRef]
31. Martins, L.; Cardeira, C.; Oliveira, P. Feedback Linearization with Zero Dynamics Stabilization for Quadrotor Control. *J. Intell. Robot. Syst. Theory Appl.* **2021**, *101*, 1–17. [CrossRef]
32. Kidambi, K.B.; Fermuller, C.; Aloimonos, Y.; Xu, H. Robust Nonlinear Control-Based Trajectory Tracking for Quadrotors under Uncertainty. *IEEE Control Syst. Lett.* **2021**, *5*, 2042–2047. [CrossRef]
33. Bhargavapuri, M.; Sahoo, S.R.; Kothari, M.; Abhishek. Robust nonlinear control of a variable-pitch quadrotor with the flip maneuver. *Control Eng. Pract.* **2019**, *87*, 26–42. [CrossRef]

34. Li, Z.; Ma, X.; Li, Y. Robust tracking control strategy for a quadrotor using RPD-SMC and RISE. *Neurocomputing* **2019**, *331*, 312–322. [CrossRef]
35. Pliego-Jiménez, J. Quaternion-based adaptive control for trajectory tracking of quadrotor unmanned aerial vehicles. *Int. J. Adapt. Control Signal Process.* **2021**, *35*, 628–641. [CrossRef]
36. Muthusamy, P.k.; Garratt, M.; Pota, H.R.; Muthusamy, R. Realtime Adaptive Intelligent Control System for Quadcopter UAV with Payload Uncertainties. *IEEE Trans. Ind. Electron.* **2021**, *69*, 1641–1653. [CrossRef]
37. Guan, X.; Lou, S.; Li, H.; Tang, T. Intelligent control of quad-rotor aircrafts with a STM32 microcontroller using deep neural networks. *Ind. Robot* **2021**, *48*, 700–709. [CrossRef]
38. Al-Mahturi, A.; Santoso, F.; Garratt, M.A.; Anavatti, S.G. Self-Learning in Aerial Robotics Using Type-2 Fuzzy Systems: Case Study in Hovering Quadrotor Flight Control. *IEEE Access* **2021**, *9*, 119520–119532. [CrossRef]
39. Sun, C.; Liu, M.; Liu, C.; Feng, X.; Wu, H. An Industrial Quadrotor UAV Control Method Based on Fuzzy Adaptive Linear Active Disturbance Rejection Control. *Electronics* **2021**, *10*, 376. [CrossRef]
40. Jiang, Z.; Lynch, A.F. Quadrotor motion control using deep reinforcement learning. *J. Unmanned Veh. Syst.* **2021**, *9*, 234–251. [CrossRef]
41. Pi, C.H.; Ye, W.Y.; Cheng, S. Robust Quadrotor Control through Reinforcement Learning with Disturbance Compensation. *Appl. Sci.* **2021**, *11*, 3257. [CrossRef]
42. Kamel, M.; Burri, M.; Siegwart, R. Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles. *IFAC-PapersOnLine* **2017**, *50*, 3463–3469. [CrossRef]
43. 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 2021, Rotterdam, The Netherlands, 29 June–2 July 2020; pp. 1556–1563. [CrossRef]
44. Gros, S.; Zanon, M.; Quirynen, R.; Bemporad, A.; Diehl, M. From linear to nonlinear MPC: Bridging the gap via the real-time iteration. *Int. J. Control* **2020**, *93*, 62–80. [CrossRef]
45. Alexis, K.; Nikolakopoulos, G.; Tzes, A. On Trajectory Tracking Model Predictive Control of an Unmanned Quadrotor Helicopter Subject to Aerodynamic Disturbances. *Asian J. Control* **2014**, *16*, 209–224. [CrossRef]
46. 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.* **2019**, *11*, 1–36. [CrossRef]
47. Kahn, M.E.; Roth, B. The Near-Minimum-Time Control Of Open-Loop Articulated Kinematic Chains. *J. Dyn. Syst. Meas. Control* **1971**, *93*, 164–172. [CrossRef]
48. Lloyd, S.; Irani, R.; Ahmadi, M. A numeric derivation for fast regressive modeling of manipulator dynamics. *Mech. Mach. Theory* **2021**, *156*, 104149. [CrossRef]
49. Herrera, D.; Roberti, F.; Carelli, R.; Andaluz, V.; Varela, J.; Ortiz, J.; Canseco, P. Modeling and Path-Following Control of a Wheelchair in Human-Shared Environments. *Int. J. Humanoid Robot.* **2018**, *15*, 1850010. [CrossRef]
50. Bai, G.; Meng, Y.; Liu, L.; Luo, W.; Gu, Q.; Liu, L. Review and comparison of path tracking based on model predictive control. *Electronics* **2019**, *8*, 1077. [CrossRef]
51. Romero, A.; Sun, S.; Foehn, P.; Scaramuzza, D. Model predictive contouring control for time-optimal quadrotor flight. *IEEE Trans. Robot.* **2022**, *38*, 3340–3356. [CrossRef]
52. Amin, R.; Aijun, L.; Shamshirband, S. A review of quadrotor UAV: Control methodologies and performance evaluation. *Int. J. Autom. Control* **2016**, *10*, 87–103. [CrossRef]
53. Tripathi, V.K.; Yogi, S.C.; Kamath, A.K.; Behera, L.; Verma, N.K.; Nahavandi, S. A Disturbance Observer-Based Intelligent Finite-Time Sliding Mode Flight Controller Design for an Autonomous Quadrotor. *IEEE Syst. J.* **2022**, *16*, 1649–1660. [CrossRef]
54. Lindqvist, B.; Mansouri, S.S.; Agha-mohammadi, A.A.; Nikolakopoulos, G. Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6001–6008. [CrossRef]
55. Reinhold, J.; Baumann, H.; Meurer, T. Constrained-Differential-Kinematics-Decomposition-Based NMPC for Online Manipulator Control with Low Computational Costs. *Robotics* **2023**, *12*, 7. [CrossRef]
56. Ju, F.; Zong, Y.; Zhuang, W.; Wang, Q.; Wang, L. Real-Time NMPC for Speed Planning of Connected Hybrid Electric Vehicles. *Machines* **2022**, *10*, 1129. [CrossRef]

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