mathematics

MDPI

Article

Lie-Group Type Quadcopter Control Design by Dynamics Replacement and the Virtual Attractive-Repulsive Potentials Theory

Simone Fiori [1,*], Luca Bigelli [2] and Federico Polenta [3]

[1] Department of Information Engineering, Marches Polytechnic University, 60121 Ancona, Italy
[2] Graduate School of Mechatronic Engineering, Politecnico di Torino, 10129 Turin, Italy; luca.bigelli@studenti.polito.it
[3] Graduate School of Automation and Control Engineering, Politecnico di Milano, 20133 Milan, Italy; federico.polenta@mail.polimi.it
* Correspondence: s.fiori@staff.univpm.it

Abstract: The aim of the present research work is to design a control law for a quadcopter drone based on the Virtual Attractive-Repulsive Potentials (VARP) theory. VARP theory, originally designed to enable path following by a small wheeled robot, will be tailored to control a quadcopter drone, hence allowing such device to learn flight planning. The proposed strategy combines an instance of VARP method to control a drone's attitude (SO(3)-VARP) and an instance of VARP method to control a drone's spatial location ($\mathbb{R}^3$-VARP). The resulting control strategy will be referred to as double-VARP method, which aims at making a drone follow a predefined path in space. Since the model of the drone as well as the devised control theory are formulated on a Lie group, their simulation on a computing platform is performed through a numerical analysis method specifically designed for these kinds of numerical simulations. A numerical simulation analysis is used to assess the salient features of the proposed regulation theory. In particular, resilience against shock-type disturbances are assessed numerically.

Keywords: VARP control theory; feedback control; Lie group; autonomous quadcopter control; path following; virtual potentials; resilience against shock-type disturbance

Citation: Fiori, S.; Bigelli, L.; Polenta, F. Lie-Group Type Quadcopter Control Design by Dynamics Replacement and the Virtual Attractive-Repulsive Potentials Theory. *Mathematics* **2022**, *10*, 1104. https://doi.org/10.3390/math10071104

Academic Editors: Panagiota Tsompanopoulou, Dimplekumar N. Chalishajar and Asier Ibeas

Received: 26 December 2021
Accepted: 24 March 2022
Published: 29 March 2022

1. Introduction

Quadcopters represent a prominent class of UAVs (Unmanned Aerial Vehicles), that are aircrafts piloted by remote control or onboard computers. In particular, quadcopters represent aerial robots whose control is based on the variation of the speed of their four propellers through the use of an appropriate algorithm, which manages to keep a drone's flight as stable and independent from any external factor as possible. Quadcopters are widely used for professional purposes such as photography, filming, and rescue operations [1–3]. Since quadcopters are easy to build and inexpensive, such aircrafts may also be found on the market for non-professional or semi-professional purposes. As quadcopters are becoming increasingly popular, a noteworthy amount of scientific and technological research has been carried out in order to design methods and algorithms to control their attitude and position.

Most control problems of interest in applied sciences and engineering concern positioning, path planning and obstacles avoidance. Virtual potentials proved effective in solving non-linear control problems. In order to control a particular dynamical system by a virtual-potential-based paradigm, it is necessary to build a virtual potential field, which may be constructed by borrowing ideas from the theory of harmonic functions and Laplace's equations [4,5], artificial gyroscopic forces [6] and stream functions from fluid dynamics [7]. Potential fields have been widely used to control different sorts of vehicles such as cylindrical robots [8], helicopters [9], road vehicles [10], unmanned ground

vehicles [11], and autonomous underwater vehicles [12]. Seminal research works on potential fields theory applied to robot navigation were published by Koditschek and Rimon in the 1990s [13,14]. Their works were based on the observation that the freespace, goal and obstacles may be encoded through a potential function, called *navigation function*, and that a gradient-based controlled system would naturally be able to navigate through the freespace.

The use of potential fields is affected by known drawbacks, as illustrated in several research outputs, such as [15–19]. The aim of the present paper is to design a novel control scheme based on a theory known as *Virtual Attractive-Repulsive Potentials* (VARP) [20,21]. The versatility of such a theory makes it profitable in several different fields, such as biology [20] and vehicle coordination [22]. The idea of using artificial potential functions for robot coordination is well established. Even in the manifold of special orthogonal rotations $\mathrm{SO}(3)$, artificial potential functions were employed for synchronization and control at least as early as 2007 in [23]. Several papers have employed a variation of the Riemannian metric in $\mathrm{SO}(3)$ or other manifolds as a potential function/error function/navigation function for constructing control schemes, see, e.g., [24–27].

The main contribution of the present paper is to propose an extension of the original VARP method to control dynamical systems whose state-space equations are formulated on Lie groups. In particular, in order to control a quadcopter drone, we shall make use of the VARP method twice for two different purposes, namely, for regulating both attitude and location of a quadcopter.

Section 2 of the present paper recalls relevant notations about Lie groups and their prominent properties. This section explains how the VARP control method may be extended to regulate a second-order dynamical system formulated on a Lie group, with special reference to the $\mathrm{SO}(3)$ manifold of three-dimensional rotations, in order to control a quadcopter [28–31]. A mathematical model of a quadcopter drone is recalled from the paper [32]. This section explains in detail the mathematical techniques utilized to design a control field, based on the notions of 'dynamics replacement' and error feedback control. All mathematical steps required to design a VARP-based control equations for a quadcopter drone are reported in details for the benefit of the reader. A particular attention will be dedicated to evaluating the 'control effort' and rotors' speed dynamics, which are essential in analyzing the physical realizability of the proposed control method. A further part of this section will also explain how the devised control law may be time-discretized and numerically integrated by means of a tailored forward Euler-type numerical method to perform numerical simulations.

Section 3 introduces a novel non-linear regulation method based the VARP theory extended to Lie groups. In particular, the introduced regulation method is based on two instances of a VARP-based control field. An instance, termed VARP_1, will serve to stabilize the attitude of a drone during flight. Such instance proves of prime importance to maintain a drone in a proper attitude and to disallow the quadcopter to tip over. A further instance, termed VARP_2, will fix the drone attitude to steer its body toward a pre-defined spatial target. In addition, within this section, it is defined an evolution law to control the vertical positioning of a drone during target approaching.

Section 4 completes the present document, focusing on conclusions and future works.

2. VARP Control Theory on SO(3) Manifold

The present section explains how to extend the VARP principle to control dynamical systems whose state-space possesses the structure of the special orthogonal group $\mathrm{SO}(3)$. The manifold $\mathrm{SO}(3)$ has been chosen because its elements, the special orthogonal matrices, could represent quadcopters attitude.

2.1. Lie Groups, Lie Algebras, Definitions and Properties

Let us recapitulate the following definitions and properties from manifold calculus and matrix Lie group theory from [33]:

Matrix Lie group: A smooth matrix manifold $\mathbb{G}$ that also enjoys the properties of an algebraic group is termed a matrix Lie group. A matrix group is a matrix set endowed with:

1. an associative binary operation, termed *group multiplication* that, for any two elements $g, h \in \mathbb{G}$, returns an element denoted as $gh \in \mathbb{G}$,
2. an *identity element* with respect to the group multiplication, denoted by e, such that $eg = ge = g$ for any element $g \in \mathbb{G}$,
3. an *inversion operation* with respect to group multiplication, denoted by g^{-1}, such that $g^{-1}g = gg^{-1} = e$ for any element $g \in \mathbb{G}$,
4. a *left translation* $L : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ defined by the relationship $L_g(h) := g^{-1}h$.

An instance of matrix Lie group is $\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = RR^\top = I_3, \det(R) = +1\}$, where the symbol $^\top$ denotes matrix transposition and the quantity I_3 represents a 3×3 identity matrix. Such group-manifold is termed *special orthogonal group* and its elements represent rigid rotations in a three-dimensional space.

Tangent bundle and its metrization: Given a point $g \in \mathbb{G}$, the tangent space to $\mathbb{G}$ at g will be denoted as $T_g\mathbb{G}$. The tangent bundle associated with a manifold-group $\mathbb{G}$ is denoted by $T\mathbb{G}$ and plays the role of phase-space. The scalar product of two vectors $u, v \in T_g\mathbb{G}$ is denoted by $\langle u, v\rangle_g$. A smooth function $F : \mathbb{G} \to \mathbb{G}$ induces a linear map $\mathrm{d}F_g : T_g\mathbb{G} \to T_{F(g)}\mathbb{G}$ termed *pushforward map*. For a matrix Lie group, the pushforward map $\mathrm{d}(L_g)_h : T_h\mathbb{G} \to T_{g^{-1}h}\mathbb{G}$ associated with a left translation is $\mathrm{d}(L_g)_h(u) := g^{-1}u$, where $u \in T_h\mathbb{G}$. The symbol $d : \mathbb{G}^2 \to \mathbb{R}_+$ denotes the *Riemannian distance* over the manifold $\mathbb{G}$ associated to the scalar product $\langle\cdot,\cdot\rangle_\cdot : (T\mathbb{G})^2 \to \mathbb{R}$.

Lie algebra: The tangent space $\mathfrak{g} := T_e\mathbb{G}$ to a Lie group at the identity is termed *Lie algebra*. Any Lie algebra is endowed with *Lie brackets*, denoted as $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, and an *adjoint endomorphism* $\mathrm{ad}_u v := [u, v]$. (It interesting to notice that the Lie-bracket operator may be seen as a generalization of vector product.) The Lie algebra associated to the group $\mathrm{SO}(3)$ is $\mathfrak{so}(3) := \{\xi \in \mathbb{R}^{3\times 3} \mid \xi + \xi^\top = 0\}$. On a matrix Lie algebra, Lie brackets coincide with matrix commutator, namely $[u, v] := uv - vu$. In particular, the matrix commutator in $\mathfrak{so}(3)$ is anti-symmetric, namely $[\xi, \eta] + [\eta, \xi] = 0$ for every $\xi, \eta \in \mathfrak{so}(3)$. A pushforward map $\mathrm{d}(L_g)_g : T_g\mathbb{G} \to \mathfrak{g}$ is denoted as $\mathrm{d}L_g$ for brevity. It is convenient to define a basis of $\mathfrak{so}(3) = \mathrm{span}(\chi_x, \chi_y, \chi_z)$ as follows:

$$\chi_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \ \chi_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \ \chi_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{1}$$

Additionally, it pays to define the operator $[\![\cdot]\!] : \mathbb{R}^3 \to \mathfrak{so}(3)$ as:

$$x := \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto [\![x]\!] := \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{2}$$

Canonical operators and the Riemannian distance for the Lie group $\mathrm{SO}(3)$ endowed with the inner product $\langle U, V\rangle_X := \mathrm{tr}(U^\top V)$ take the following expressions:

$$\begin{cases} \text{Exponential map: } \exp_X(V) := X\,\mathrm{Exp}(X^\top V), \\ \text{Logarithmic map: } \log_X Y := X\,\mathrm{Log}(X^\top Y), \\ \text{Riemannian distance: } d(X, Y) := \|\mathrm{Log}(X^\top Y)\|_{\mathrm{F}}, \\ \text{Orthogonal projection to } T_R\mathrm{SO}(3)\text{: } \Pi_R(V) := \frac{1}{2}R(R^\top V - V^\top R), \\ \text{Left translation: } \mathrm{L}_R(V) := R^\top V, \end{cases} \tag{3}$$

where 'Exp' denotes matrix exponential, 'Log' denotes principal matrix logarithm and $\|\cdot\|_{\mathrm{F}}$ denotes a Frobenius matrix norm ($\|U\|_{\mathrm{F}} := \sqrt{\mathrm{tr}(U^\top U)}$).

The Lie group $\mathrm{SO}(3)$ denotes the (continuous) set of all rotations in the three-dimensional Euclidean space $\mathbb{R}^3$, which models the spatial arena where physical objects move. Any spatial rotation may be described either through a set of three angles (termed Euler angles), or through an axis–angle representation [34]. The axis–angle representation will be considered since it parametrizes a rotation in a three-dimensional Euclidean space by two quantities: a unit vector $\hat{n}$ (sometimes referred to as the Euler axis), which indicates the axis of rotation, and an angle φ, which describes the magnitude of the rotation about the chosen axis. Notice that only two parameters (not three) are needed to define the direction of a unit vector $\hat{n}$ rooted at the origin, because its magnitude is constrained to 1. By Rodrigues' rotation formula [35,36], it is possible to determine a rotation matrix as:

$$R_{\hat{n}}(\varphi) := \hat{n}\hat{n}^\top + \cos\varphi\,(I_3 - \hat{n}\hat{n}^\top) - \sin\varphi\,[\![\hat{n}]\!], \tag{4}$$

with a desired angle φ and a desired rotation axis $\hat{n} = [n_1\ n_2\ n_3]^\top$ (which needs to be a unit vector). Rodrigues' rotation formula returns a rotation matrix $R_{\hat{n}}(\varphi)$ whose columns and rows are orthogonal unit vectors and whose transpose matrix is equal to its inverse so that $R^\top = R^{-1}$ and whose determinant is equal to 1.

2.2. VARP Extension to the SO(3) Manifold

The VARP principle may be summarized as follows. Given N particles labeled $i = 1, 2, \ldots, N$, that form an aggregate, the model proposed in [20] that governs the motion of each particle in $\mathbb{R}^3$ reads:

$$\begin{cases} \frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} = \vec{v}_i, \\ m_i \frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} = \alpha \hat{f}_i - \beta \vec{v}_i - \vec{\mathrm{grad}} V, \end{cases} \tag{5}$$

where the constants $m_i > 0$ denote the mass of each particle, while vectors $\vec{x}_i \in \mathbb{R}^3$ and $\vec{v}_i \in \mathbb{R}^3$ denote their location and velocity, respectively. The function V is a virtual attractive-repulsive potential that determines the external forces acting on each particle. Each particle experiences a self-propelling force $\vec{f}_i \in \mathbb{R}^3$ with fixed magnitude $\alpha > 0$. A friction force with coefficient $\beta > 0$ prevents particles from reaching large speeds. Levine et al. verified experimentally that the qualitative behavior of the aggregate is independent of the explicit expression of the interaction potential. As a matter of fact, Levine's model is based on virtual potentials of the kind:

$$V := \sum_{j,\, j\neq i} \mathrm{C_r} \exp\left(-\frac{\left\|\vec{x}_i - \vec{x}_j\right\|}{\ell_\mathrm{r}}\right) - \sum_{j,\, j\neq i} \mathrm{C_a} \exp\left(-\frac{\left\|\vec{x}_i - \vec{x}_j\right\|}{\ell_\mathrm{a}}\right), \tag{6}$$

where the constants $\mathrm{C_a} > 0$, $\mathrm{C_r} > 0$ determine the strength of the attractive and repulsive forces, respectively. Each particle in a Levine's model is subjected to an attraction force that depends only on the distance from the others, characterized by an interaction range $\ell_a > 0$. Such force is responsible for the aggregation of the particles. A shorter-range repulsive force, characterized by an interaction range $\ell_r > 0$, prevents the aggregate from collapsing.

The structure of the virtual potential field depends on the application at hand, which spans different areas, such as identification of traffic congestion situation in cold-climate cities [37] and collision avoidance in quadcopter group formation [38].

On the basis of the above-recalled VARP theory, Nguyen et al. in [21] proposed a control method to drive a small wheeled robot. We propose an extension to the Lie group $\mathrm{SO}(3)$ of the system of equations describing the VARP theory and subsequently a control method to regulate systems evolving on the Lie group $\mathrm{SO}(3)$.

The proposed extension to Levine's model (5) is laid out as follows:

$$\begin{cases} \dot{R} = R\chi, \\ \nabla_{\dot{R}}\dot{R} = \alpha R\hat{\chi} - \mu\dot{R} - \mathrm{grad}_R V, \end{cases} \tag{7}$$

where $R \in \mathrm{SO}(3)$ denotes the attitude of a rotating body, $\dot{R}$ denotes its rotational velocity, $\chi \in \mathfrak{so}(3)$ denotes its angular velocity, $\nabla_{\dot{R}}\dot{R}$ denotes its acceleration (indeed, the symbol ∇ denotes covariant derivation), the symbol $\hat{\chi}$ denotes normalized angular velocity (exhibiting unitary norm) and $\mathrm{grad}_R V$ denotes the Riemannian gradient of the potential function $V : \mathrm{SO}(3) \to \mathbb{R}$. (The covariant derivative of a vector field with respect to a given tangent direction represents a generalization of directional derivative of a vector field and returns a further tangent direction).

The above extension to Levine's model may be written in a more explicit fashion by means of the following mathematical result.

Lemma 1. *On the Lie group* $\mathrm{SO}(3)$*, it holds that* $\nabla_{\dot{R}}\dot{R} = R\dot{\chi}$.

Proof. Recalling that $R^\top R = I_3$, deriving both sides with respect to time yields:

$$\dot{R}^\top R + R^\top \dot{R} = 0, \tag{8}$$

which gives $\chi^\top + \chi = 0$ with $\dot{R} = R\chi$. Deriving again the Equation (8) with respect to time yields:

$$\ddot{R}^\top R + \dot{R}^\top \dot{R} + \dot{R}^\top \dot{R} + R^\top \ddot{R} = 0,$$

namely:

$$\ddot{R}^\top R + 2\dot{R}^\top \dot{R} + R^\top \ddot{R} = 0. \tag{9}$$

From the property $\dot{R} = R\chi$, it follows that

$$2\dot{R}^\top \dot{R} = -2\chi^2. \tag{10}$$

The covariant derivative $\nabla_{\dot{R}}\dot{R}$ may be computed as the orthogonal projection of the naïve derivative $\ddot{R}$ onto the tangent space $T_R\mathrm{SO}(3)$. (The quantity $\ddot{R}$ is termed *naïve* derivative. Such matrix function may be decomposed into a normal component to the tangent space and a tangent component. Through the orthogonal projector operator defined in (3), the tangent component is retained.) Applying the projection defined in (3) gives:

$$\nabla_{\dot{R}}\dot{R} = \frac{1}{2}R(R^\top \ddot{R} - \ddot{R}^\top R). \tag{11}$$

Plugging in Equations (9) and (10) leads to:

$$\nabla_{\dot{R}}\dot{R} = \frac{R}{2}(R^\top \ddot{R} - 2\chi^2 + R^\top \ddot{R}) = -R\chi^2 + \ddot{R}. \tag{12}$$

Recalling that $\ddot{R} = R\dot{\chi} + \dot{R}\chi = R(\chi^2 + \dot{\chi})$ gives:

$$\nabla_{\dot{R}}\dot{R} = R(\chi^2 + \dot{\chi} - \chi^2) = R\dot{\chi}, \tag{13}$$

which proves the assertion. □

On the basis of the above lemma and on the fact that $\dot{R} = R\chi$, the second equation in (7) may be rewritten as:

$$R\dot{\chi} = \alpha R\hat{\chi} - \mu R\chi - \mathrm{grad}_R V. \tag{14}$$

Applying the left translation operator to both sides allows one to write explicitly an equation for the angular acceleration, namely:

$$\dot{\chi} = \alpha\hat{\chi} - \mu\chi - L_R(\mathrm{grad}_R V), \tag{15}$$

therefore the system (7) may be recast as:

$$\begin{cases} \dot{R}(t) = R(t)\chi(t), \\ \dot{\chi}(t) = \alpha\hat{\chi}(t) - \mu\chi(t) - L_{R(t)}\big(\mathrm{grad}_{R(t)}V(R(t))\big), \end{cases} \tag{16}$$

where:

- the matrix function $R(t) \in \mathrm{SO}(3)$ denotes a rotation matrix;
- the matrix function $\dot{R}(t) \in T_{R(t)}\mathrm{SO}(3)$ denotes the derivative of the matrix R with respect to time, hence velocity;
- the matrix function $\chi(t) \in \mathfrak{so}(3)$ denotes angular velocity;
- the matrix function $\dot{\chi}(t) \in \mathfrak{so}(3)$ denotes the rate at which the angular velocity χ changes, therefore it represents angular acceleration;
- the term $\mu\chi(t) \in \mathfrak{so}(3)$ represents the friction term that opposes to rotation; the constant μ denotes a friction coefficient;
- the term $\alpha\hat{\chi}(t) \in \mathfrak{so}(3)$ denotes a self-propelling term; such forcing term is parallel to χ, while its magnitude coincides with a constant $\alpha > 0$, in fact, whenever $\chi \neq 0$ it holds that $\hat{\chi} := \chi/\|\chi\|_{\mathrm{F}}$, otherwise $\hat{\chi} = 0$. The self-propelling term might cause severe oscillations around the attraction point; notice that, in the absence of any potentials (namely $V = 0$), the self-propelling term causes a velocity drift of amplitude $\|\chi_{\mathrm{drift}}\|_{\mathrm{F}} = \frac{\alpha}{\mu}$;
- $L_{R(t)}\big(\mathrm{grad}_{R(t)}V(R(t))\big) \in \mathfrak{so}(3)$ denotes the gradient of the potential function translated to the algebra.

  The potential function that generalizes the original Levine's function has been chosen as

$$V(R) := \sum_{j=1}^{N}\Big(V_j^{\mathrm{r}}(d^2(R,R_j)) - V_j^{\mathrm{a}}(d^2(R,R_j))\Big), \tag{17}$$

where each $R_j \in \mathrm{SO}(3)$ represents either an unwanted attitude to disallow, or a desired attitude to reach. The left-translated Riemannian gradient of such potential function reads:

$$\begin{aligned} L_R\Big(\mathrm{grad}_R V(R)\Big) &= L_R\left[\mathrm{grad}_R \sum_{j=1}^{N}\Big(V_j^{\mathrm{r}}(d^2(R,R_j)) - V_j^{\mathrm{a}}(d^2(R,R_j))\Big)\right] \\ &= L_R\left[\sum_{j=1}^{N}\Big(\dot{V}_j^{\mathrm{r}}(d^2(R,R_j)) - \dot{V}_j^{\mathrm{a}}(d^2(R,R_j))\Big)\mathrm{grad}_R d^2(R,R_j)\right]. \end{aligned} \tag{18}$$

Notice that each element exhibits both characteristics: in case of an attractive location, the attractive characteristic will be predominant over the repulsive one, or vice versa.

In analogy to the original VARP principle, exponential-type attractive and repulsive potential functions have been chosen as:

$$V_j^{\mathrm{a}}(\phi) := C_j^{\mathrm{a}} e^{-\sqrt{y}/\ell_j^{\mathrm{a}}},\ V_j^{\mathrm{r}}(\phi) := C_j^{\mathrm{r}} e^{-\sqrt{y}/\ell_j^{\mathrm{r}}}, \tag{19}$$

where the constants C_j^{a} and C_j^{r} denote the 'magnitude of the potentials', ℓ_j^{a} and ℓ_j^{r} their 'characteristic lengths' and the symbol y denotes a real, positive variable.

A purely repulsive point with index j is characterized by a coefficient $C_j^{\mathrm{a}} = 0$, while a purely attractive point with index j is characterized by a coefficient $C_j^{\mathrm{r}} = 0$. In [21], it is set $C_j^{\mathrm{r}} = 0$ if a point represents a target while the obstacles are usually characterized by $C_j^{\mathrm{r}} \gg C_j^{\mathrm{a}}$. An explanation could lie in the fact that by setting obstacles, it is possible to build a path to guide a moving agent, hence they should attract an agent but, at the same time, the agent must avoid obstacles. In the case of a target, it is advisable to set $C_j^{\mathrm{r}} \neq 0$ if an agent should approach that particular target without actually reaching it. (To explain such a choice by a practical example, one may consider the case of a small robot whose

goal is to approach a tree while keeping at a reasonable distance from its roots.) In this instance, it is advisable to set $C_j^{\mathrm{a}} \gg C_j^{\mathrm{r}} \neq 0$. On the other hand, in the case of a quadcopter drone, it is generally allowed to set $C_j^{\mathrm{r}} = 0$ since a desired attitude does not represent a physical object to avoid. For different kinds of aircrafts, placing a purely repulsive potential at a certain (disallowed) attitude might prove instrumental to allow maneuvering while avoiding bright objects [39].

From the known property $\mathrm{grad}_R d^2(R,Q) = -2\log_R Q$, for every $R, Q \in \mathrm{SO}(3)$, it immediately follows that:

$$L_R\Big(\mathrm{grad}_R V(R)\Big) = L_R\left[2\sum_{j=1}^{N}\Big(\dot{V}_j^{\mathrm{a}}(d^2(R,R_j)) - \dot{V}_j^{\mathrm{r}}(d^2(R,R_j))\Big)\log_R(R_j)\right]. \tag{20}$$

Computing the derivatives of the sub-potentials with respect to the distance function leads to the detailed expression:

$$L_R\Big(\mathrm{grad}_R V(R)\Big) = L_R\left[\sum_{j=1}^{N}\left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}}\frac{e^{-d(R,R_j)/\ell_j^{\mathrm{r}}}}{d(R,R_j)} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}}\frac{e^{-d(R,R_j)/\ell_j^{\mathrm{a}}}}{d(R,R_j)}\right)\log_R(R_j)\right]. \tag{21}$$

Furthermore, upon recalling the definition given in (3) about the Riemannian distance between two points in the manifold $\mathrm{SO}(3)$, the Equation (21) becomes:

$$L_R\Big(\mathrm{grad}_R V(R)\Big) = L_R\left[\sum_{j=1}^{N}\left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{r}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{a}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}\right)\log_R(R_j)\right]. \tag{22}$$

Applying the left translation operator as defined in (3) to transport the potential term into the Lie algebra $\mathfrak{so}(3)$ yields:

$$L_R\Big(\mathrm{grad}_R V(R)\Big) = \sum_{j=1}^{N}\left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{r}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{a}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}\right)R^\top\log_R(R_j). \tag{23}$$

Plugging in the expression of the logarithmic map for the manifold $\mathrm{SO}(3)$ (as recalled in (3)) gives:

$$R^\top\log_R(R_j) = R^\top R\mathrm{Log}(R^\top R_j) = \mathrm{Log}(R^\top R_j). \tag{24}$$

The final expression for the potential gradient hence reads:

$$L_R\Big(\mathrm{grad}_R V(R)\Big) = \sum_{j=1}^{N}\left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{r}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}}\frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{a}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}\right)\mathrm{Log}(R^\top R_j). \tag{25}$$

2.3. Quadcopter Model

Since the aim of this paper is to develop a Lie-group based control method to regulate the flight of a quadcopter drone, we deemed it appropriate to recall the following quadcopter drone Lie-group type mathematical model from [32]:

$$\begin{cases} \dot{R} &= R\chi, \\ \dot{\chi} &= D^{-1}([\hat{J}_{\mathrm{q}}, \chi^2] + [\beta, \chi] - \dot{\beta} + \tau)D^{-1}, \\ \beta &:= (-\omega_1 + \omega_2 - \omega_3 + \omega_4)J_{\mathcal{R}}\chi_z, \\ \tau &:= br(\omega_4^2 - \omega_2^2)\chi_x + br(\omega_3^2 - \omega_1^2)\chi_y + \gamma(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2)\chi_z, \\ \dot{q} &= v, \\ \dot{v} &= \frac{1}{2}\frac{b}{M_{\mathrm{q}}}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)R\mathrm{e}_z - \bar{g}\mathrm{e}_z - \frac{1}{M_{\mathrm{q}}}\Gamma v. \end{cases} \tag{26}$$

The constant matrix D is defined as:

$$D := \mathrm{diag}\left(\sqrt{\frac{J_y J_z}{J_x}}, \sqrt{\frac{J_x J_z}{J_y}}, \sqrt{\frac{J_x J_y}{J_z}}\right), \tag{27}$$

which contains the principal moments of inertia of the drone. In the above equations, R denotes the attitude of the drone with respect to an inertial reference frame, χ denotes its angular velocity, the quantities ω_i, $i = 1, 2, 3, 4$, denote the angular velocities of the four propellers, the function q denotes the position of the center of mass of the drone, the vector e_z denotes the unit vector $\mathrm{e}_z := [0\ 0\ 1]^\top$, the function τ denotes the mechanical torque exerted by the thrusters on the drone's body and several constants represent the physical features of the drone, such as mass, inertial coefficients and propeller efficiency parameters. A graphical rendering of a small drone modeled by Equation (26) is displayed in the Figure 1.

Figure 1. A small quadcopter drone simulated numerically by the UnivPM team. A reference system attached to the drone has its x-y axes parallel to the plane of the drone and the z axis normal to such a plane.

In order to perform numerical simulations, an OS4-Mini-VTOL quadrotor has been taken as reference model, as reported in [40]. In particular, the parameters values summarized in Table 1 were made use of in the computer codes.

The recalled mathematical model consists of a series of Lie-group type differential equations that need to be solved numerically on a computing platform. Such important aspect will be discussed in Section 2.8.

Table 1. Summary table of the OS4 Mini-VTOL quadcopter parameters (SI units).

Parameter	Symbol	Value
Overall quadrotor mass	M_{q}	0.650 kg
Inertia on x axis	J_x	7.5×10^{-3} kg $\cdot$ m^2
Inertia on y axis	J_y	7.5×10^{-3} kg $\cdot$ m^2
Inertia on z axis	J_z	1.3×10^{-2} kg $\cdot$ m^2
Thrust coefficient	b	3.13×10^{-5} N $\cdot$ s^2
Drag coefficient	γ	7.5×10^{-7} N $\cdot$ m $\cdot$ s^2
Rotor inertia	$J_{\mathcal{R}}$	6×10^{-5} kg $\cdot$ m^2
Arm length	r	0.23 m
Gravitational acceleration	$\bar{g}$	9.81 m $\cdot$ s^{-2}

2.4. Control Field Design by Dynamics Replacement and Error Feedback

The present subsection explains how to design a control field by dynamics replacement and error feedback control through a direct transposition of the VARP theory.

A second-order physical system on a Lie group $\mathbb{G}$ may be formulated as

$$\begin{cases} \dot{R}_{\mathrm{s}} = R_{\mathrm{s}}\chi_{\mathrm{s}}, \\ \dot{\chi}_{\mathrm{s}} = \sigma_{\mathrm{s}}(\chi_{\mathrm{s}}) + u, \end{cases} \tag{28}$$

where $\sigma_{\mathrm{s}} : \mathfrak{g} \to \mathfrak{g}$ denotes a state-transition function and $u \in \mathfrak{g}$ a Lie-algebra control field. The initial conditions for the initial value problem (28) are $R_{\mathrm{s}}(0) = R_{\mathrm{s},0}$ and $\chi_{\mathrm{s}}(0) = \chi_{\mathrm{s},0}$. A control method aims at making the dynamics of the system (28) conform to the desired dynamics of the reference system:

$$\begin{cases} \dot{R}_{\mathrm{m}} = R_{\mathrm{m}}\chi_{\mathrm{m}}, \\ \dot{\chi}_{\mathrm{m}} = \sigma_{\mathrm{m}}(\chi_{\mathrm{m}}), \end{cases} \tag{29}$$

where $\sigma_{\mathrm{m}} : \mathfrak{g} \to \mathfrak{g}$ denotes a further state-transition function. The initial conditions of this system are denoted by $R_{\mathrm{m}}(0) = R_{\mathrm{m},0}$ and $\chi_{\mathrm{m}}(0) = \chi_{\mathrm{m},0}$. The initial state of the reference system (29) would likely differ from the initial state of the system (28). A way to achieve synchronization of the dynamical system (28) to the reference system (29) is to set the control field as

$$u := \sigma_{\mathrm{m}}(\chi_{\mathrm{m}}) - \sigma_{\mathrm{s}}(\chi_{\mathrm{s}}) + \kappa\, \mathrm{L}_{R_{\mathrm{s}}}(\log_{R_{\mathrm{s}}} R_{\mathrm{m}}), \tag{30}$$

where $\kappa > 0$ is a control parameter. The term $\sigma_{\mathrm{m}}(\chi_{\mathrm{m}}) - \sigma_{\mathrm{s}}(\chi_{\mathrm{s}})$ represents an instance of the *principle of dynamics replacement*, namely, it has the purpose of canceling the dynamics of the real-world system and to replace it with the dynamics of the reference system. The term $\log_{R_{\mathrm{s}}} R_{\mathrm{m}}$ represents a feedback error on the Lie-group variables and has the purpose to align such states to one another. In fact, notice that dynamics replacement aligns the angular velocities, which *per se*, does not align the states. In addition, the term $\log_{R_{\mathrm{s}}} R_{\mathrm{m}}$ compensates for possible mismatches in the system model.

2.5. Design of a VARP-Based Control Law for a Quadcopter Drone

The above general-purpose control-design theory may be tailored to the case of interest in the present research work as follows.

In the present setting, the second-order physical system (28) represents a quadcopter drone, namely:

$$\sigma_{\mathrm{s}}(\chi_{\mathrm{s}}) := D^{-1}([\hat{J}_q, \chi_{\mathrm{s}}^2] + [\beta, \chi_{\mathrm{s}}] - \dot{\beta})D^{-1}, \tag{31}$$

while the system model (29) represents the dynamic prescribed by the VARP theory, namely:

$$\sigma_{\mathrm{m}}(\chi_{\mathrm{m}}) := \alpha\,\hat{\chi}_{\mathrm{m}} - \mu\,\chi_{\mathrm{m}} - \mathrm{L}_{R_{\mathrm{m}}}(\nabla_{R_{\mathrm{m}}} V). \tag{32}$$

The only term that can be acted upon to control a drone is the external mechanical torque $\tau \in \mathfrak{so}(3)$, hence, in the present setting, the control field u introduced in (28) reads

$$u := D^{-1}\tau D^{-1} \in \mathfrak{so}(3). \tag{33}$$

According to dynamics replacement principle and error feedback control, we shall set the control field to

$$\begin{aligned} u = {} & \alpha\,\hat{\chi}_{\mathrm{m}} - \mu\,\chi_{\mathrm{m}} - \mathrm{L}_{R_{\mathrm{m}}}(\nabla_{R_{\mathrm{m}}} V) - D^{-1}[\hat{J}_q, \chi_{\mathrm{s}}^2]D^{-1} - D^{-1}[\beta, \chi_{\mathrm{s}}]D^{-1} \\ & + D^{-1}\dot{\beta}D^{-1} + \kappa\,\mathrm{L}_{R_{\mathrm{s}}}(\log_{R_{\mathrm{s}}} R_{\mathrm{m}}). \end{aligned} \tag{34}$$

Henceforth, the VARP control method entails a particular structure of the mechanical torque given by

$$\begin{aligned} \tau_{\mathrm{VARP}} := DuD = {} & \alpha\,D\hat{\chi}_{\mathrm{m}}D - \mu\,D\chi_{\mathrm{m}}D - D\mathrm{L}_{R_{\mathrm{m}}}(\nabla_{R_{\mathrm{m}}} V)D - [\hat{J}_q, \chi_{\mathrm{s}}^2] - [\beta, \chi_{\mathrm{s}}] \\ & + \dot{\beta} + \kappa\,D\mathrm{L}_{R_{\mathrm{s}}}(\log_{R_{\mathrm{s}}} R_{\mathrm{m}})D, \end{aligned} \tag{35}$$

which needs to be generated by the thrusters in order to drive the quadcopter drone along a desired trajectory. Ultimately, the system of equations that describes a VARP-controlled drone is:

$$\begin{cases} \dot{R}_{\mathrm{m}}(t) = R_{\mathrm{m}}(t)\chi_{\mathrm{m}}(t), \\ \dot{\chi}_{\mathrm{m}}(t) = \alpha\,\hat{\chi}_{\mathrm{m}}(t) - \mu\,\chi_{\mathrm{m}}(t) - \mathrm{L}_{R_{\mathrm{m}}(t)}(\nabla_{R_{\mathrm{m}}} V), \\ \mathrm{L}_R(\nabla_R V) = \sum_{j=1}^{N} \left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}} \frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{r}}}}}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}} \frac{e^{\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{a}}}}}{\|\mathrm{Log}(R(t)^\top R_j)\|_{\mathrm{F}}} \right) \mathrm{Log}(R^\top R_j), \\ \dot{R}_{\mathrm{s}}(t) = R_{\mathrm{s}}(t)\chi_{\mathrm{s}}(t), \\ \dot{\chi}_{\mathrm{s}}(t) = D^{-1}([\hat{J}_q, \chi_{\mathrm{s}}^2(t)] + [\beta, \chi_{\mathrm{s}}(t)] - \dot{\beta}(t) + \tau_{\mathrm{VARP}}(t))D^{-1}, \\ \tau_{\mathrm{VARP}} = \alpha\,D\hat{\chi}_{\mathrm{m}}D - \mu\,D\chi_{\mathrm{m}}D - D\mathrm{L}_{R_{\mathrm{m}}}(\nabla_{R_{\mathrm{m}}} V)D - [\hat{J}_q, \chi_{\mathrm{s}}^2] - [\beta, \chi_{\mathrm{s}}] \\ \qquad\qquad + \dot{\beta} + \kappa\,D\mathrm{Log}(R_{\mathrm{s}}^\top R_{\mathrm{m}})D. \end{cases} \tag{36}$$

We now notice that a simplified control scheme would arise from the following two hypotheses:

- It is not necessary to synchronize the attitudes of the two systems perfectly, which implies that the constant κ may be set to zero; in addition, since the reference system is purely abstract, it may be initialized identically to the drone system;
- The rotational speeds of the two systems are close to each other, namely $\chi_{\mathrm{m}} \approx \chi_{\mathrm{s}}$, hence their values may be taken as equal to a common value χ.

Under the above hypotheses, the control scheme (36) would simplify to

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = D^{-1}([\hat{J}_q, \chi^2] + [\beta, \chi] - \dot{\beta} + \tau_{\mathrm{VARP}})D^{-1}, \\ \tau_{\mathrm{VARP}} := \alpha\,D\hat{\chi}D - \mu\,D\chi D - D\mathrm{L}_R(\nabla_R V)D - [\hat{J}_q, \chi^2] - [\beta, \chi] + \dot{\beta}, \end{cases} \tag{37}$$

where the footers dropped since all descriptive variables pertain to the quadcopter drone (namely, the 's' system). The mathematical model of a controlled drone (37) will be analyzed in the remainder of this paper. The above equations will be referred to as SO(3)-VARP control method in the following developments.

2.6. Physical Realizability and Control Effort Analysis

The control field prescribed by the VARP theory needs to be consistent with the mechanical torque generated by the four thrusters, namely:

- The speed of rotation χ cannot be too large and its component along the vertical axis z needs to stay close to zero (to prevent the quadcopter to yaw along its vertical axis);
- The mechanical torque τ is generated by the fans, which can spin up to a maximum speed (the thrusters are small DC-brushless electrical motors, hence they cannot generate a large thrust); moreover, the velocities ω_1, ω_2, ω_3, ω_4 need not to depart excessively from the hovering velocity, which is denoted as ω_{ss} in [32] and satisfies

$$\omega_{\mathrm{ss}}^2 = \frac{M_{\mathrm{q}}\bar{g}}{2b}. \tag{38}$$

In order to verify that the designed control strategy fulfills the physical realizability of the devised control action, it is advisable to evaluate the three components $\Omega_x := \langle \chi, \chi_x \rangle$, $\Omega_y := \langle \chi, \chi_y \rangle$ and $\Omega_z := \langle \chi, \chi_z \rangle$ of the angular velocity of the drone as well as the three components of the control torque field τ_{VARP}, namely $\tau_x^{\mathrm{V}} := \langle \tau_{\mathrm{VARP}}, \chi_x \rangle$, $\tau_y^{\mathrm{V}} := \langle \tau_{\mathrm{VARP}}, \chi_y \rangle$ and $\tau_z^{\mathrm{V}} := \langle \tau_{\mathrm{VARP}}, \chi_z \rangle$. The components of the control torque are bound to the values generated by the four fans, as described in (26), namely

$$\begin{cases} \tau_x := br(\omega_4^2 - \omega_2^2), \\ \tau_y := br(\omega_3^2 - \omega_1^2), \\ \tau_z := \gamma(\omega_2^2 - \omega_1^2 + \omega_4^2 - \omega_3^2). \end{cases} \tag{39}$$

Notice that such three equations bind four rotation speeds, hence the propeller rotation speeds corresponding to a desired mechanical torque may not be determined uniquely from (39). (More efficient drones in their maneuvering may employ all four rotors for each of the three rotations, although these models might consume more power for the roll and pitch rotations.) To complete the equations set, it is customary to complement the Equation (39) with the hovering condition [32]

$$b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = 2M_{\mathrm{q}}\bar{g}. \tag{40}$$

A fundamental step in the analysis of the physical realizability of the VARP control action is the computation of the propellers speed corresponding to the control field. In order to compute the rotors angular velocities corresponding to the control field τ_{VARP}, it is possible to derive three conditions from the Equation (39):

$$\begin{cases} br(\omega_4^2 - \omega_2^2) = \tau_x^{\mathrm{V}}, \\ br(\omega_3^2 - \omega_1^2) = \tau_y^{\mathrm{V}}, \\ \gamma(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) = \tau_z^{\mathrm{V}}. \end{cases} \tag{41}$$

The system (41) is under-determined, since it totals three equations in four unknowns, therefore it gets complemented by the hovering condition (40). The system of algebraic Equation (41) hence takes the form:

$$\begin{cases} \omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2 = \frac{1}{\gamma}\tau_z^{\mathrm{V}}, \\ \omega_4^2 - \omega_2^2 = \frac{1}{br}\tau_x^{\mathrm{V}}, \\ \omega_3^2 - \omega_1^2 = \frac{1}{br}\tau_y^{\mathrm{V}}, \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \frac{2M_{\mathrm{q}}\bar{g}}{b}. \end{cases} \tag{42}$$

The above system of equations is linear in the squared unknowns ω_1^2, ω_2^2, ω_3^2 and ω_4^2 and admits solutions

$$\begin{cases} \omega_1^2 = \frac{-2\gamma\tau_y^{\mathrm{V}} - br\tau_z^{\mathrm{V}} + 2r\gamma M_{\mathrm{q}}\bar{g}}{4br\gamma}, \\ \omega_2^2 = \frac{br\tau_z^{\mathrm{V}} - 2\gamma\tau_x^{\mathrm{V}} + 2r\gamma M_{\mathrm{q}}\bar{g}}{4br\gamma}, \\ \omega_3^2 = \frac{2\gamma\tau_y^{\mathrm{V}} - br\tau_z^{\mathrm{V}} + 2r\gamma M_{\mathrm{q}}\bar{g}}{4br\gamma}, \\ \omega_4^2 = \frac{2\gamma\tau_x^{\mathrm{V}} + br\tau_z^{\mathrm{V}} + 2r\gamma M_{\mathrm{q}}\bar{g}}{4br\gamma}. \end{cases} \tag{43}$$

The solutions (43) are real-valued, hence physically realizable, in principle, whenever the values taken by the components of the control field do not differ too largely from one another and are absolutely bounded. Furthermore, the more massive a drone, the lighter the effect of the control field components on the propellers velocities. Introducing the expression (38), the above solutions may be rewritten equivalently as

$$\begin{cases} \omega_1^2 = \omega_{\mathrm{ss}}^2 - \frac{1}{2br}\tau_y^{\mathrm{V}} - \frac{1}{4\gamma}\tau_z^{\mathrm{V}}, \\ \omega_2^2 = \omega_{\mathrm{ss}}^2 - \frac{1}{2br}\tau_x^{\mathrm{V}} + \frac{1}{4\gamma}\tau_z^{\mathrm{V}}, \\ \omega_3^2 = \omega_{\mathrm{ss}}^2 + \frac{1}{2br}\tau_y^{\mathrm{V}} - \frac{1}{4\gamma}\tau_z^{\mathrm{V}}, \\ \omega_4^2 = \omega_{\mathrm{ss}}^2 + \frac{1}{2br}\tau_x^{\mathrm{V}} + \frac{1}{4\gamma}\tau_z^{\mathrm{V}}. \end{cases} \tag{44}$$

From the above expressions, it is apparent how the control actions cause a deviation of the rotors velocities from the hovering velocity proportional to the control torques. Such deviations need not be large in order to guarantee the physical realizability of a control action. It is worth noticing how the τ_z^{V} component of the control torque field influences all rotors velocities, as it is responsible for the yawing of the quadcopter drone, while the τ_x^{V} component only affects the rotors placed along the y-axis and the τ_y^{V} component only affects the rotors along the x-axis.

A further observation of interest is that the hovering condition (40) is valid only when a quadcopter is almost horizontal with respect to the inertial reference frame: whenever a drone is tilted, the normal component of the thrust decreases, hence, to balance the gravitational pull, a controller will slightly increase the speed of all rotors. For a tilted quadcopter (namely, whenever $R \neq I_3$), the simple hovering condition (40) generalizes to:

$$\tfrac{b}{2}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)Re_{\mathrm{z}} = M_{\mathrm{q}}\bar{g}e_{\mathrm{z}}. \tag{45}$$

This is a vector equation with only a scalar unknown $\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2$, hence it may be solved by pre-multiplying both sides by $e_{\mathrm{z}}^{\top}$, which leads to

$$\tfrac{b}{2}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = \frac{M_{\mathrm{q}}\bar{g}}{e_{\mathrm{z}}^{\top}Re_{\mathrm{z}}}. \tag{46}$$

Notice that $|R_{(3,3)}| = |e_{\mathrm{z}}^{\top}Re_{\mathrm{z}}| \leq 1$, hence the right-hand side of the condition (46) is larger than (or equal to) $M_{\mathrm{q}}\bar{g}$. In the case that $R \neq I_3$, the system of Equation (41) takes the form:

$$\begin{cases} \omega_4^2 - \omega_2^2 = \frac{1}{br}\tau_x^{\mathrm{V}}, \\ \omega_3^2 - \omega_1^2 = \frac{1}{br}\tau_y^{\mathrm{V}}, \\ -\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 = \frac{1}{\gamma}\tau_z^{\mathrm{V}}, \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \frac{2M_{\mathrm{q}}\bar{g}}{be_{\mathrm{z}}^{\top}Re_{\mathrm{z}}}. \end{cases} \tag{47}$$

Again, the system of equations that relate the desired torques, as calculated by the VARP control method, to the propellers rotation speeds, is linear in the squared unknowns ω_1^2, ω_2^2, ω_3^2 and ω_4^2 and admits solutions

$$\begin{cases} \omega_1^2 = \frac{M_{\mathrm{q}}\bar{g}}{2bR_{(3,3)}} - \frac{\tau_y^{\mathrm{V}}}{2br} - \frac{\tau_z^{\mathrm{V}}}{4\gamma}, \\ \omega_2^2 = \frac{M_{\mathrm{q}}\bar{g}}{2bR_{(3,3)}} - \frac{\tau_x^{\mathrm{V}}}{2br} + frac\tau_z^{\mathrm{V}}4\gamma, \\ \omega_3^2 = \frac{M_{\mathrm{q}}\bar{g}}{2bR_{(3,3)}} + \frac{\tau_y^{\mathrm{V}}}{2br} - \frac{\tau_z^{\mathrm{V}}}{4\gamma}, \\ \omega_4^2 = \frac{M_{\mathrm{q}}\bar{g}}{2bR_{(3,3)}} + \frac{\tau_x^{\mathrm{V}}}{2br} + \frac{\tau_z^{\mathrm{V}}}{4\gamma}. \end{cases} \tag{48}$$

By the relation (38), the above solutions may be rewritten equivalently as

$$\begin{cases} \omega_1^2 = \frac{\omega_{\mathrm{ss}}^2}{R_{(3,3)}} - \frac{\tau_y^{\mathrm{V}}}{2br} - \frac{\tau_z^{\mathrm{V}}}{4\gamma}, \\ \omega_2^2 = \frac{\omega_{\mathrm{ss}}^2}{R_{(3,3)}} - \frac{\tau_x^{\mathrm{V}}}{2br} + \frac{\tau_z^{\mathrm{V}}}{4\gamma}, \\ \omega_3^2 = \frac{\omega_{\mathrm{ss}}^2}{R_{(3,3)}} + \frac{\tau_y^{\mathrm{V}}}{2br} - \frac{\tau_z^{\mathrm{V}}}{4\gamma}, \\ \omega_4^2 = \frac{\omega_{\mathrm{ss}}^2}{R_{(3,3)}} + \frac{\tau_x^{\mathrm{V}}}{2br} + \frac{\tau_z^{\mathrm{V}}}{4\gamma}. \end{cases} \tag{49}$$

Such relationship quantifies the amount of deviation from the hovering speed due to inclination and to desired torque values. The steady-state velocity of the OS4-Mini-VTOL quadcopter is $\omega_{\mathrm{ss}} = \sqrt{\frac{M_{\mathrm{q}}\bar{g}}{2b}} \approx 319$ rad/s. The constants that appear in the expressions (44) take values $2br \approx 1.44 \times 10^{-5}\ \mathrm{N \cdot m \cdot s^2}$ and $4\gamma = 3 \times 10^{-6}\ \mathrm{N \cdot m \cdot s^2}$. The relations (49) will be made use of in the course of numerical simulations to quantify the physical realizability of the devised control action.

In addition to the above discussion, it is worth surveying the value of the *control effort* defined as

$$\varepsilon := \tfrac{1}{2}\sqrt{\langle u, u\rangle}, \tag{50}$$

where $u := D^{-1}\tau_{\mathrm{VARP}}D^{-1}$ and $\langle\cdot,\cdot\rangle$ denotes the inner product in the algebra $\mathfrak{so}(3)$. Notice that the measurement unit of the control efforts results to be the same of an angular acceleration, namely $\mathrm{rad/s^2}$. The control effort quantifies the effort required to the actuators to achieve a desired control action. Since, ultimately, all control actions amount at making a drone take an appropriate attitude, including translational control, the control effort is defined only in terms of (scaled) mechanical torque.

2.7. Case-Study: Single Attractive/Repulsive Point

As a case-study to clarify the subsequent development of a full attitudinal/positional control, let us consider a VARP-controlled drone bound to approach a single target, realized by an attractive point potential. The equations of motion of the controlled quadcopter may be summarized as follows:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = \alpha\,\hat{\chi} - \mu\,\chi - \left[\frac{\mathrm{C}^{\mathrm{r}}}{\ell^{\mathrm{r}}}\exp\left(\frac{-\|\mathrm{Log}(R^\top R^{\mathrm{a}})\|_{\mathrm{F}}}{\ell^{\mathrm{r}}}\right) - \frac{\mathrm{C}^{\mathrm{a}}}{\ell^{\mathrm{a}}}\exp\left(\frac{-\|\mathrm{Log}(R^\top R^{\mathrm{a}})\|_{\mathrm{F}}}{\ell^{\mathrm{a}}}\right)\right]\frac{\mathrm{Log}(R^\top R^{\mathrm{a}})}{\|\mathrm{Log}(R^\top R^{\mathrm{a}})\|_{\mathrm{F}}}, \\ \dot{q} = v, \\ \dot{v} = \frac{b}{2M_{\mathrm{q}}}\left(\sum_{i=1}^4 \omega_i^2\right)R\mathrm{e}_z - \bar{g}\mathrm{e}_z - \frac{1}{M_{\mathrm{q}}}\Gamma v, \end{cases} \tag{51}$$

where $R^{\mathrm{a}} \in \mathrm{SO}(3)$ denotes a desired/undesired attitude. It is important to emphasize that the translational component of motion depends on the attitude R as well as on the sum of squared velocities of the propellers, which is an independent variable, as discussed in Section 2.6.

Since the propeller speeds are almost-unitary fractions of the hovering speed, we set

$$\sum_{i=1}^4 \omega_i^2 = \eta\,\frac{2M_{\mathrm{q}}\bar{g}}{b\,\mathrm{e}_z^\top R\mathrm{e}_z}, \tag{52}$$

where the parameter η represents a sort of 'throttle' coefficient, namely $\eta < 1$ causes a drone to descend, $\eta > 1$ causes vertical ascent, while $\eta = 1$ corresponds to hovering in midair. As a consequence, the equations describing a controlled drone may be rewritten as:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = \left(\dfrac{\alpha}{\|\chi\|_{\mathrm{F}}} - \mu\right)\chi \\ \qquad -\left[\dfrac{C^{\mathrm{r}}}{\ell^{\mathrm{r}}}\exp\left(\dfrac{-\|\mathrm{Log}(R^{\top}R^{\mathrm{a}})\|_{\mathrm{F}}}{\ell^{\mathrm{r}}}\right) - \dfrac{C^{\mathrm{a}}}{\ell^{\mathrm{a}}}\exp\left(\dfrac{-\|\mathrm{Log}(R^{\top}R^{\mathrm{a}})\|_{\mathrm{F}}}{\ell^{\mathrm{a}}}\right)\right]\dfrac{\mathrm{Log}(R^{\top}R^{\mathrm{a}})}{\|\mathrm{Log}(R^{\top}R^{\mathrm{a}})\|_{\mathrm{F}}}, \\ \dot{q} = v, \\ \dot{v} = \bar{g}\left(\dfrac{\eta\, R}{\mathrm{e}_z^{\top}R\mathrm{e}_z} - I_3\right)\mathrm{e}_z - \dfrac{1}{M_{\mathrm{q}}}\Gamma v. \end{cases} \tag{53}$$

Apparently, the term $\bar{g}M_{\mathrm{q}}\left(\dfrac{\eta\, R}{\mathrm{e}_z^{\top}R\mathrm{e}_z} - I_3\right)\mathrm{e}_z$ denotes the net mechanical force that the thrusters are able to exert on the (center of mass of the) drone. Notice that when $R = I_3$, such net force equals $(\eta - 1)M_{\mathrm{q}}\bar{g}\mathrm{e}_z$, coherently with the role taken by the coefficient η.

2.8. Numerical Integration of the Equations of Motion

In order to simulate numerically the flight of a controlled quadcopter, the simplest numerical integration method has been employed, namely, the forward Euler (fEul) method [32,41]. In order to make such methods effective in solving Lie-groups equations, it has been customized coherently with the mathematical structure of the rotation group $\mathrm{SO}(3)$.

Recasting the equations of the controlled system (36) in a compact way, we have:

$$\begin{cases} \dot{R}(t) = \nu(\chi(t), R(t)),\ t \geq 0, \\ \dot{\chi}(t) = \sigma(\chi(t), R(t)), \\ \xi(0) = \chi_0,\ R(0) = R_0, \end{cases} \tag{54}$$

where $\nu : \mathfrak{so}(3) \times \mathrm{SO}(3) \to T\mathrm{SO}(3)$ and $\sigma : \mathfrak{so}(3) \times \mathrm{SO}(3) \to \mathfrak{so}(3)$ represent the right-hand sides of the first and of the second equation in (36), respectively. According to the fEul method, the value of the first-order derivative can be approximated by means of the right-sided incremental ratio, namely it may be written as:

$$\dot{\chi}(t) = \frac{\chi(t+h) - \chi(t)}{h} + \text{approximation error}, \tag{55}$$

where $h > 0$ denotes a step size. Ignoring the approximation error term and introducing a uniform time discretization, we obtain:

$$\frac{\chi_{k+1} - \chi_k}{h} = \sigma(\chi_k, R_k),\ k = 0,\ 1,\ 2,\ 3,\ \ldots, \tag{56}$$

which gives rise to the iteration:

$$\chi_{k+1} = \chi_k + h\sigma(\chi_k, R_k),\ k = 0,\ 1,\ 2,\ 3,\ \ldots. \tag{57}$$

Notice that the above iterations is based on the fundamental observation that the second equation in (54) involves terms in the vector space $\mathfrak{so}(3)$ only.

In addition, it is necessary to consider the numerical integration of the first equation in (54). The solution of the first equation in (54) may be approximated by a fEul method (upon uniform time discretization) as

$$\dot{R}(t) = \frac{R(t+h) - R(t)}{h} + \text{approximation error} \ \Rightarrow\ R_{k+1} = R_k + h\nu(\chi_k, R_k), \tag{58}$$

however, such iteration will quickly drive the rotation matrix sequence R_k out of the rotation group, which is not a vector space. The reason for such difficulty is that it is not legitimate to treat a curved manifold as a linear space and, in particular, summation is not allowed. Simple summation must be replaced by the exponential map, which takes a pair $(R, V) \in T\mathrm{SO}(3)$ into a point in the $\mathrm{SO}(3)$ manifold. Ultimately, the Equation (58) should be written as:

$$R_{k+1} = \exp_{R_k}(h\nu(\chi_k, R_k)) = R_k \mathrm{Exp}(h\, R_k^\top \nu(\chi_k, R_k)). \tag{59}$$

The complete set of equations used in the numerical simulations thus reads:

$$\begin{cases} R_{k+1} = R_k \mathrm{Exp}(h\chi_k), \\ \chi_{k+1} = \chi_k + \alpha h\hat{\chi}_k - \mu h\chi_k - hL_{R_k}(\nabla_{R_k} V(R_k)), \\ L_R(\nabla_R V(R)) = \sum_{j=1}^{N} \left(\frac{C_j^{\mathrm{r}}}{\ell_j^{\mathrm{r}}} \frac{\exp\left(\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{r}}}\right)}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} - \frac{C_j^{\mathrm{a}}}{\ell_j^{\mathrm{a}}} \frac{\exp\left(\frac{-\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}}{\ell_j^{\mathrm{a}}}\right)}{\|\mathrm{Log}(R^\top R_j)\|_{\mathrm{F}}} \right) \mathrm{Log}(R^\top R_j), \\ k = 0,\ 1,\ 2,\ 3, \ldots. \end{cases} \tag{60}$$

We mention that there exist a number of possible approaches to integrate numerically differential equations formulated on manifold tangent bundles or Lie-groups tangent bundles.

3. Double-VARP Control of a Quadcopter

The SO(3)-VARP control method has been designed specifically to regulate a quadcopter's attitude, namely, given a set of desired/undesired attitudes, the SO(3)-VARP method ultimately provides an expression τ_{VARP} for the torque that the propellers need to produce in order to steer a quadcopter. In fact, from the quadcopter model (26), it is readily noticed that the SO(3)-VARP control field will affect only the evolution of the attitude R and of the angular velocity χ of the quadcopter. The devised SO(3)-VARP method does not influence directly the position and the linear velocity of a drone (although the control method developed so far influences indirectly the spatial trajectory of a drone through its attitude). In order to steer a quadcopter over a desired spatial trajectory, it is hence necessary to regulate its translational dynamics, represented by the variables q and v, by influencing directly its linear acceleration.

The aim of the present section is to illustrate how to exploit the notion of VARP regulation to achieve position control applied to quadcopter guidance.

3.1. Double VARP Control Theory

The main proposal is to use two instances of the VARP regulation theory, namely, two controllers that act concurrently to achieve two goals:

- A first instance (referred to as VARP_1) will serve to stabilize the attitude of a drone during flight; the purpose of this instance of VARP controller is to make sure the tilt of the drone keeps limited, namely that the body of the drone will always keep in an almost-horizontal orientation, hence stabilizing its flight mode;
- A second instance (referred to as VARP_2) will regulate the tilting of the drone in such a way that it is steered toward a predefined target-point in space, regardless of its initial position, orientation and overall rotation speed.

To what concern positional regulation, from the last equation in the quadcopter model (26), it is possible to notice that the thrust exerted by the thrusters on the drone's body is always normal to its x-y plane, therefore, in order to steer the quadcopter toward a desired direction, the total thrust must be directed toward the target and, consequently, the drone's body must be tilted (i.e., rolled and pitched) along an appropriate direction. In other terms, the z axis of the drone must be heading to a desired direction.

To summarize the following development, a VARP controller is used to induce an appropriate tilting of a drone's body at every instant, and a mechanical torque is calculated that tends to align the z axis of the drone to a steering direction. The mechanical torque

is, in fact, computed by a *cross product* between the steering direction provided by the controller and the direction of the z axis of the drone. In addition, special attention should be paid to fill up the vertical gap between the drone's actual position and the target. Such goal may be achieved independently by setting the throttle coefficient proportional to such a gap.

The equations of controlled motion inspired by the SO(3)-VARP principle may be laid out as follows:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = -\mu\,\chi + \tau_1^{\mathrm{V}} + \tau_2^{\mathrm{V}}, \\ \dot{q} = v, \\ \dot{v} = \bar{g}\left(\dfrac{\eta\, R}{\mathrm{e}_z^\top R \mathrm{e}_z} - I_3\right)\mathrm{e}_z - \dfrac{1}{M_{\mathrm{q}}}\Gamma v, \end{cases} \tag{61}$$

where the self-propelling term has been eliminated, as it was deemed to be detrimental (namely, we set $\alpha = 0$), and $\tau_1^{\mathrm{V}} \in \mathfrak{so}(3)$ and $\tau_2^{\mathrm{V}} \in \mathfrak{so}(3)$ denote the mechanical torques corresponding to two concurring VARP controllers. In order to complete the equations, we need to specify the mathematical laws to iteratively compute the quantities η, τ_1^{V}, τ_2^{V}.

In the following, we shall denote by $q_{\mathrm{t}} \in \mathbb{R}^3$ the set point representing the desired positional target. The displacement $\epsilon \in \mathbb{R}^3$ between the current location of the (center of mass of the) drone and the set point is defined as

$$\epsilon := q_{\mathrm{t}} - q. \tag{62}$$

The z component of the displacement will be denoted as $\epsilon_z := \epsilon^\top \mathrm{e}_z$.

Vertical positioning: In order to achieve vertical positioning, it is necessary to establish an evolution law for the coefficient η. The following law was exploited:

$$\eta = \exp(C_\eta\, \epsilon_z), \tag{63}$$

where $C_\eta > 0$ is a coefficient that determines the sensitivity of vertical control action to the vertical displacement. When $\epsilon_z > 0$, the drone is located below the set point, hence it must ascend, therefore $\eta > 1$; when $\epsilon_z < 0$, the drone is located above the set point, hence it must descend, therefore $\eta < 1$.

Attitude stabilization (VARP$_1$): The controller VARP$_1$ must output a control pseudo-torque τ_1^{V} that tends to keep the drone as horizontal as possible (notice that $D\tau_1^{\mathrm{V}}D$ denotes indeed a mechanical torque). Therefore, it may be set up as a SO(3)-VARP controller without repulsive attitudes and with an attractive attitude set to the identity matrix I_3. Since a stabilization control action needs not be excessively stringent at target attitude, it is not necessary to choose a sharp potential as in Nguyen et al. [21], hence we relaxed the potential function to a Gaussian one, namely, we have defined

$$V_1(R) := \frac{1}{2} C_1 \ell_1 \exp\left(\frac{-d^2(R, I_3)}{\ell_1}\right), \tag{64}$$

where $d : \mathrm{SO}(3) \times \mathrm{SO}(3) \to \mathbb{R}$ denotes a Riemannian distance function. Notice that the intensity and action-range coefficients have been written in a slightly different way compared to Section 2, which facilitates the implementation of the control action. The corresponding torque term hence reads:

$$\tau_1^{\mathrm{V}} := L_R(\mathrm{grad}_R V_1) = -C_1 \exp\left(\frac{-\|\mathrm{Log}(R)\|_{\mathrm{F}}^2}{\ell_1}\right)\mathrm{Log}(R), \tag{65}$$

where we have used the fact that $\mathrm{Log}(R^\top) = -\mathrm{Log}(R)$.

Positional control (VARP$_2$): The main idea carried out in this section consists in defining a mechanical torque to steer a drone toward a set point. The controller VARP$_2$

must output a control pseudo-torque. Therefore, it may be set up as a $\mathbb{R}^3$-VARP without repulsion points and with an attraction point set to the desired location q_t. Again, we relaxed the sharp Laplacian potential function to a rounder Gaussian one, namely, we defined

$$V_2(\epsilon) := \frac{1}{2} C_2 \ell_2 \exp\left(\frac{-\|\epsilon\|^2}{\ell_2}\right). \tag{66}$$

The corresponding pseudo-torque is defined as

$$\tau_2^\mathrm{V} := [\![(R\,\mathrm{e}_z) \wedge (-\mathrm{grad}_\epsilon V_2)]\!] = C_2 \exp\left(\frac{-\|\epsilon\|^2}{\ell_2}\right) [\![(R\,\mathrm{e}_z) \wedge \epsilon]\!], \tag{67}$$

where $\wedge$ denotes cross vector product. The analytic form of the torque $[\![(R\,\mathrm{e}_z) \wedge (-\mathrm{grad}_\epsilon V_2)]\!]$ has the effect of orientating the z axis of the drone toward the positional displacement ϵ.

The complete set of equations describing the controlled drone read:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = -\mu\,\chi - C_1 \exp\left(\dfrac{-\|\mathrm{Log}(R)\|_\mathrm{F}^2}{\ell_1}\right) \mathrm{Log}(R) \\ \qquad\quad + C_2 \exp\left(\dfrac{-\|q_\mathrm{t} - q\|^2}{\ell_2}\right) [\![(R\,\mathrm{e}_z) \wedge (q_\mathrm{t} - q)]\!], \\ \dot{q} = v, \\ \dot{v} = \bar{g}\left(\dfrac{e^{C_\eta\,(q_\mathrm{t}-q)^\top \mathrm{e}_z}}{\mathrm{e}_z^\top R \mathrm{e}_z} R - I_3\right)\mathrm{e}_z - \dfrac{1}{M_\mathrm{q}}\Gamma v, \end{cases} \tag{68}$$

which may be implemented by a numerical method (see Section 2.8) once initial conditions $R(0) = R_0$, $\chi(0) = \chi_0$, $q(0) = q_0$ and $v(0) = v_0$ and a target position q_t are provided. Notice that the initial conditions may be arbitrarily given as long as they are compatible with the physics of the drone (for instance, the axis $R_0\mathrm{e}_z$ may not be excessively tilted away from vertical, and the initial rotational speed χ_0 may not exceed the drone's maximal acceptable rotation speed).

It is worth underlining that, unlike previous works on artificial potentials, the pseudo-torque τ_2^V, as defined in Equation (67), is not derived directly from the gradient of a potential, hence its expression is more involved. This fact alone makes a Lyapunov-type analysis of convergence more involved than in previous research studies.

3.2. Rotors' Speed, Real-Time Guidance and GPS Aid

As it was discussed in Section 2.6, it is important to make sure that the control field resulting from the devised application of the VARP theory be consistent with the mechanical torque that can effectively be generated by the propellers.

As a first step in this analysis, it is necessary to define the torque control law that will identify the system (37) to the system of equations (61). This torque control law, which results in an expression of the function τ_VARP, is as follows:

$$\tau_\mathrm{VARP} := -\mu D\chi D + D\tau_1^\mathrm{V} D + D\tau_2^\mathrm{V} D - [\hat{J}_q, \chi^2] - [\beta, \chi] + \dot{\beta}. \tag{69}$$

On the basis of the control torque expression and on the physical realizability discussion carried out in Section 2.6, it is possible to calculate the rotors' speed through Equation (48). Rooting the squared values of the rotors' speed from (48) results in rotors' angular speeds that make a drone follow a desired trajectory according to the proposed double VARP control method. The obtained rotors' speeds are expressed in radians per second, although it is more common to express them in RPM (revolutions per minute). In order to convert radians per second to RPM we will use the relation:

$$\omega_{\text{RPM}} = \frac{30}{\pi}\omega_{\text{rad}}, \tag{70}$$

where ω_{RPM} denotes the value of angular speed in RPM while ω_{rad} denotes the value in radians per second.

To what concern the feasibility of real-time control, we quote Lopez and McInnes [42]: "Since the guidance is entirely analytical, it is believed that this methodology may be suitable for the real-time, autonomous guidance [...] using a minimum of onboard computational power".

Further to the discussion on physical realizability, we would like to underline that the control algorithm presupposes full access to system's state, including position and orientation. For indoor applications such information may be accessed by vision-based methods combined with inertial sensing, such as visual odometry or SLAM using 2D/3D cameras, laser range finders or tether's sensory feedback [43]. For outdoor applications, GPS (combined with other localization methodologies through information fusion) may be take advantage of [44].

3.3. Numerical Simulation Results

Comprehensive numerical simulations based on the controlled mathematical model (68) were performed by taking ideal initial conditions (namely, linear and angular velocities close to zero and initial attitude R_0 equal to the identity matrix I_3) as well as random initial conditions.

The features of the devised control theory will be displayed through the following graphical illustrations:

- **Target approaching panels**: Two 3D graphs that show the same simulation result from two different perspectives; the blue line represents the trajectory followed by the center of mass of the quadcopter, while the black dashed line denotes the ideal trajectory, which directly links the initial location to the target location; the start and arrival points of the quadcopter have been marked with an open red circle, while the target has been marked by a magenta cross mark; the start and target point have been labeled to distinguish a point from another;
- **Control effort panel**: In order to analyze the effort required to the thrusters by the control law, control effort values collected during a simulation are represented in a dedicated panel;
- **Torque components panel**: This panel summarizes the values taken by the torque components that the quadcopter rotors exert on its frame in order to tilt the drone as prescribed by the double VARP control method, namely, the components of the array τ_{VARP} defined in Equation (69);
- **Distance-to-target panel**: In order to verify that the devised control method is able to take a drone to the target, values of the Euclidean distance between the center of mass of a drone and the location of the target have been collected and displayed;
- **Inclination-to-vertical panel**: In order to visualize the inclination of the drone (combined pitching and rolling), a numerical index, which we shall refer to as 'inclination to vertical $\theta(t)$', has been defined as the angle between the inertial vertical axis e_z and the drone's body vertical axis which, in the inertial reference frame, is represented by $R\mathrm{e}_z$; such angle is thus calculated as $\theta := \arccos(\mathrm{e}_z^\top R\mathrm{e}_z)$;
- **Rotors' speed panels**: Two rotors speed panels serve to display the RPM values of the propeller's speed taken during a simulation. The panel on the left-hand side shows the values recorded during the whole simulation (although occasionally it will be truncated for a better visualization) while the panel on the right-hand side displays the values of rotors' speeds within specific time-windows in order to emphasize the difference between the speeds at which different rotors spin on a drone's frame.

The total time-span of the simulation is $[0,\ t_f]$, where the value of t_f in each simulation is specified in the figures.

3.4. Target Point above a Drone's Initial Location

Figure 2 shows the dynamics of a controlled quadcopter starting from favorable conditions.

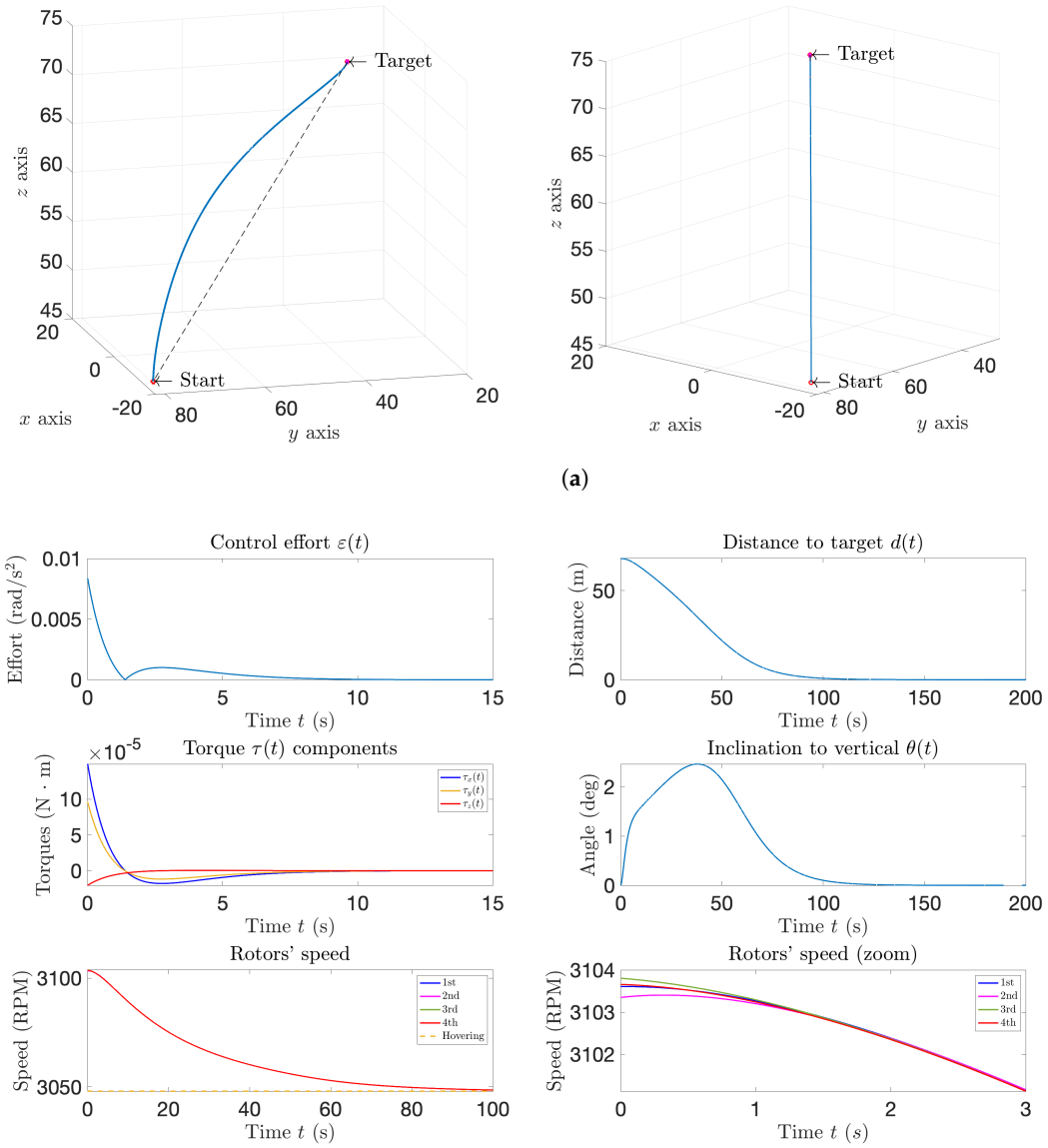


Figure 2. Experiment: Target point *above* the start point, favorable initial conditions. In the RPM panel, the yellow dashed line represents the steady-state hovering speed value ($RPM_{ss} = 3048$ from [32]). Below the panels are reported the values of VARP parameters used in the simulation. Notice that different panels present different time-spans in order to better illustrate the trend of the curves during the simulation. (**a**) Target approaching trajectory; (**b**) Control figures.

As it may be readily seen from the panel about angle of inclination, at the beginning of the simulation, the drone tilts toward the desired direction and starts to orientate towards the target point. Such behavior may also be noticed from the RPM panel on the right-hand side of the figure, which displays the values of propellers' speed during the first 3 s of the simulation. The speeds of the four propellers take slightly different values in order to tilt the drone's frame. As soon as the drone reaches the target, it tilts back to the hovering attitude, namely the inclination angle θ tends to zero.

Since the target point is initially located over the start point, overall rotors' speed is initially larger than the hovering value, in order to overcome the gravitational pull, as the RPM panel on the left-hand side shows. As the drone approaches the target, speed's values will progressively decrease to the hovering speed. It is interesting also to notice that, from the RPM panel on the left-hand side, it is difficult to distinguish the curves pertaining to different thrusters since the speed of rotation of propellers needs to be almost equal to one another, otherwise the drone will change its attitude.

From the panels that display the values taken by the control effort index and by the torque components, it is readily seen that only in the first 15 s of simulation large values of such indexes are observed since the control algorithm increases propellers' speed in order to change the quadcopter attitude and steer its body toward the target. From the panels that display the values of the distance from target, as expected, it is possible to see that the drone get progressively closer to the target until reaching it.

Results displayed in Figure 3 were obtained on an experiment that is similar to the previous one, although initial conditions were generated at random.

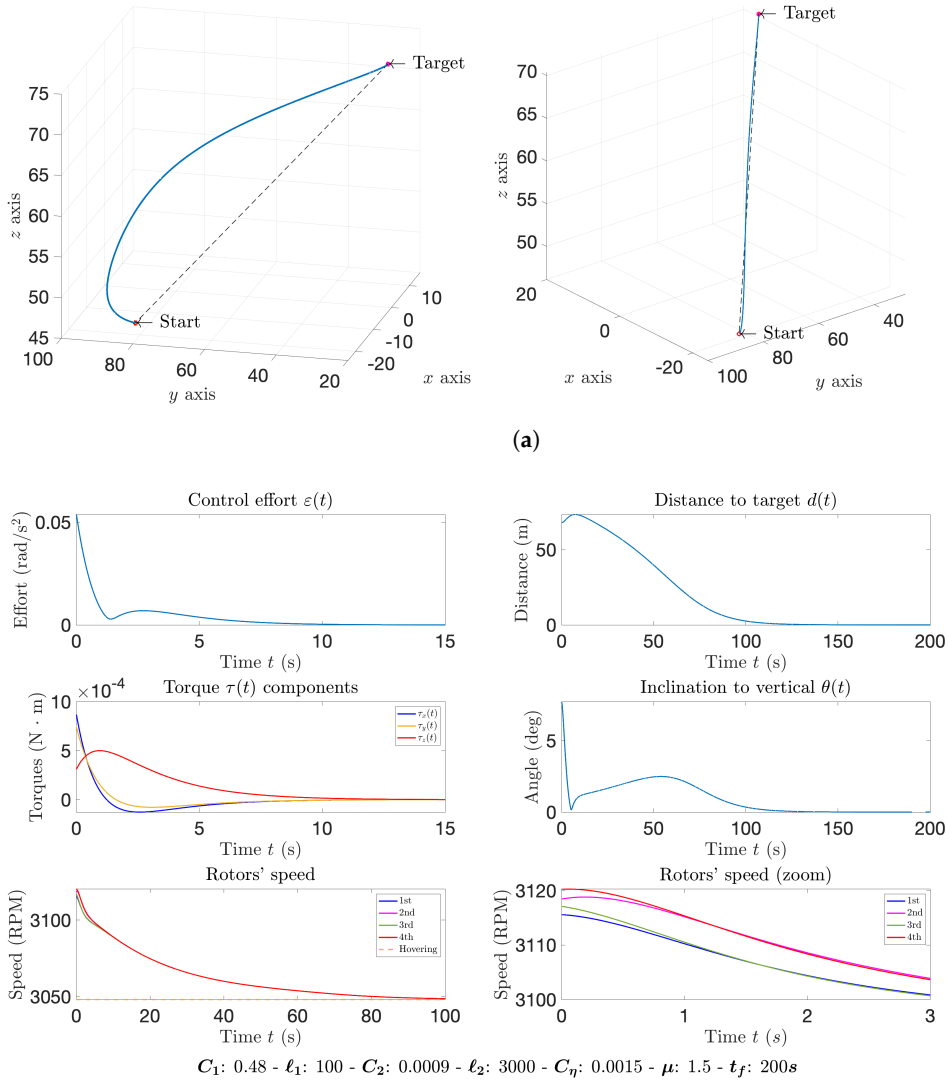


Figure 3. Experiment: Target point *above* the start point, random initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.

In this case, from the RPM graphs, it is readily seen that the drone was not initially directed towards the target since, in the first part of the simulation, the propellers' speeds

look considerably different from one another. As a consequence, the quadcopter drone varied its attitude to perform a more complex trajectory. Notice that the RPM graph on the right side was truncated to illustrate the values of the indexes over the first 10 s of simulation.

The above-discussed observations are confirmed by the results showed in the inclination-to-vertical panel: The drone was initially tilted in one direction, then it reduces its inclination and finally tilts back to the correct attitude in order to fly towards the target point. As soon as the drone has assumed the correct attitude, propellers' speeds become approximately equal to each other in order to make the drone move upward.

The unfavorable initial attitude could also be noticed from the target approaching panel on the right-hand side, where the quadcopter trajectory is not perfectly superimposed to the ideal one, but it exhibits a small 'hump' to the left and to the right.

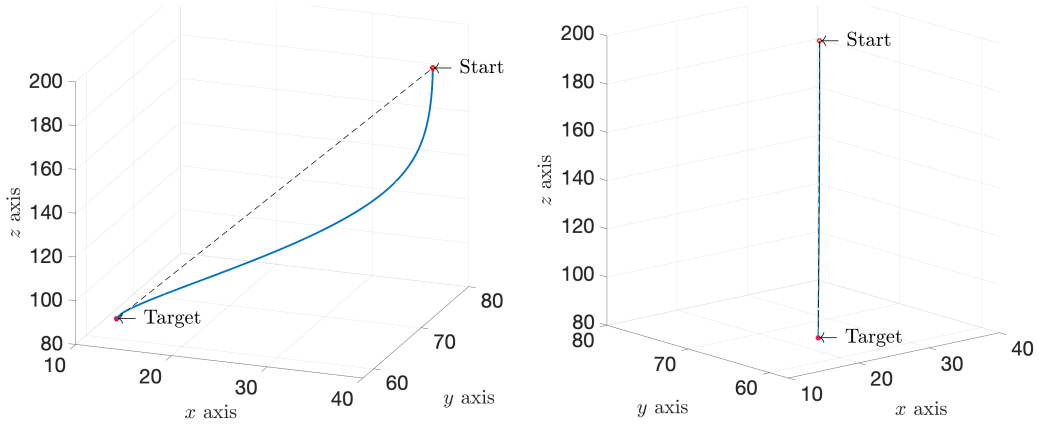
Even in this case, the curves displayed in the panels about control effort and torque components show larger values only during the first part of the simulation, when the drone needs to vary the speed of propellers and to modify its attitude. From the RPM graph on the right-hand side, it can be noticed that, with respect to the previous simulation, the rotors' speed curves look considerably different since the drone needs to adapt from a random attitude to the correct one.

3.5. Target Point below a Drone's Initial Location

The following two simulations, whose results are illustrated in the Figures 4 and 5, were obtained by placing a quadcopter below the target. As in the previous subsection, a favorable as well as a random initial attitude were simulated.

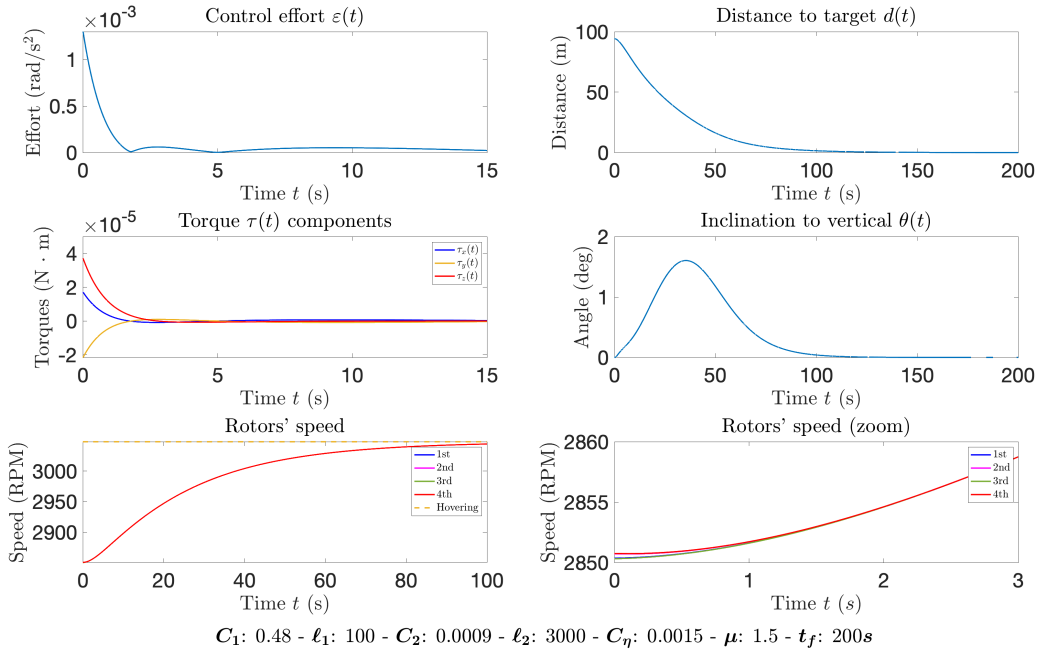
In one such case, a drone needs to move downward to approach the target hence, at the beginning of the simulation, the RPMs of the propellers take lower values than the hovering speed. As the quadcopter approaches the target, the propellers' speed tends to approach the hovering speed.

Even in this case, the trajectory followed by the quadcopter shows a small 'hump' on the left side since the initial attitude of the quadcopter is not favorable to move toward the target. Such a phenomenon may be noticed even from the RPM panel on the right side: In the simulation with favorable conditions, the speeds slightly differ from each other, while in the simulation results corresponding to random initial conditions, the RPMs look considerably different from one another during the first segment of the simulation.



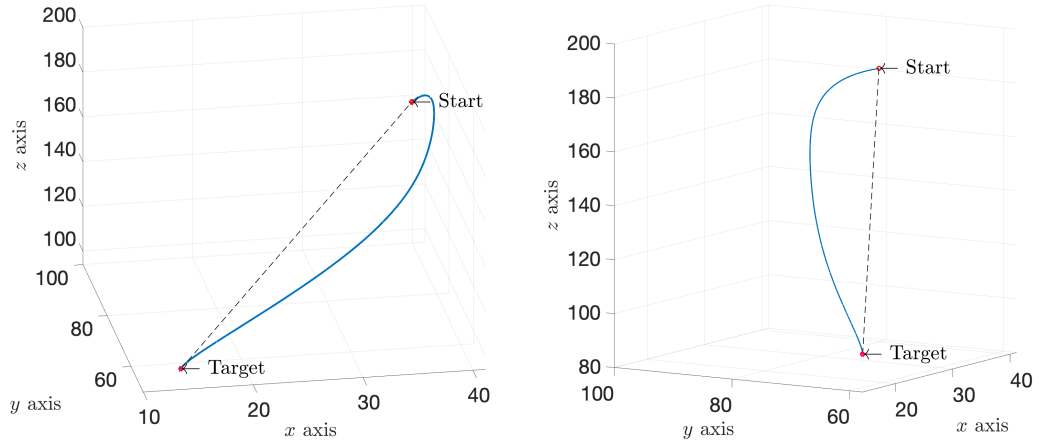
(**a**)

Figure 4. *Cont.*



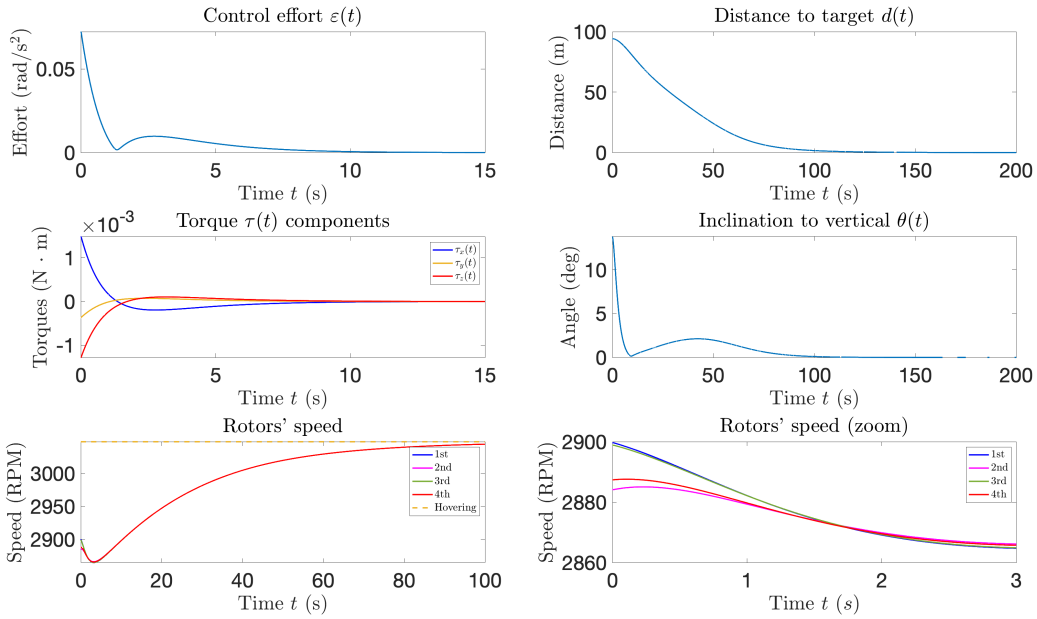
(**b**)

Figure 4. Experiment: Target point *below* the start point, favorable initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.



(**a**)

Figure 5. *Cont.*



(**b**)

Figure 5. Experiment: Target point *below* the start point, random initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.

3.6. Target Point at Same Quota as Drone's Initial Location

In the simulation results illustrated in Figure 6, the starting point and target are located at the same altitude, therefore the quadcopter needs only to slide horizontally to reach the target.

When favorable initial conditions are set, the drone follows the expected trajectory. In particular, it is possible to see from the target approaching graph on the right side that the drone moves slightly along the vertical direction, although such displacement is negligible (of the order of a few centimeters). Control effort and torque components panels show definitely low values, which corresponds to the fact that the quadcopter only needs to tilt slightly in order to fly toward the target. In order to tilt the drone's body, the propellers' RPM values reported in the panels are slightly different from one another during the first 5 s of the simulation, then the RPMs approach the hovering value. It is also interesting to notice that the rotors' speeds do not equate the hovering speed, but they differ from it for less than a half revolution per minute.

From Figure 7, the behavior of the quadcopter when starting off with random initial conditions is readily inspected.

It is indeed interesting to see that, in this case, the quadcopter slightly moves backward since it was clearly tilted in an unfavorable direction and then, as soon as it reaches a favorable attitude, it starts to fly towards the target point.

It may also be noticed from the RPM panel that, in order to vary the attitude of the quadcopter, propellers' speeds need to significantly differ from one another during the first part of the simulation, then approaching the hovering speed. Compared to the results obtained in the previous simulation, control effort and torque components panels show larger values during the first part of the simulation due to the initial necessary change of direction.

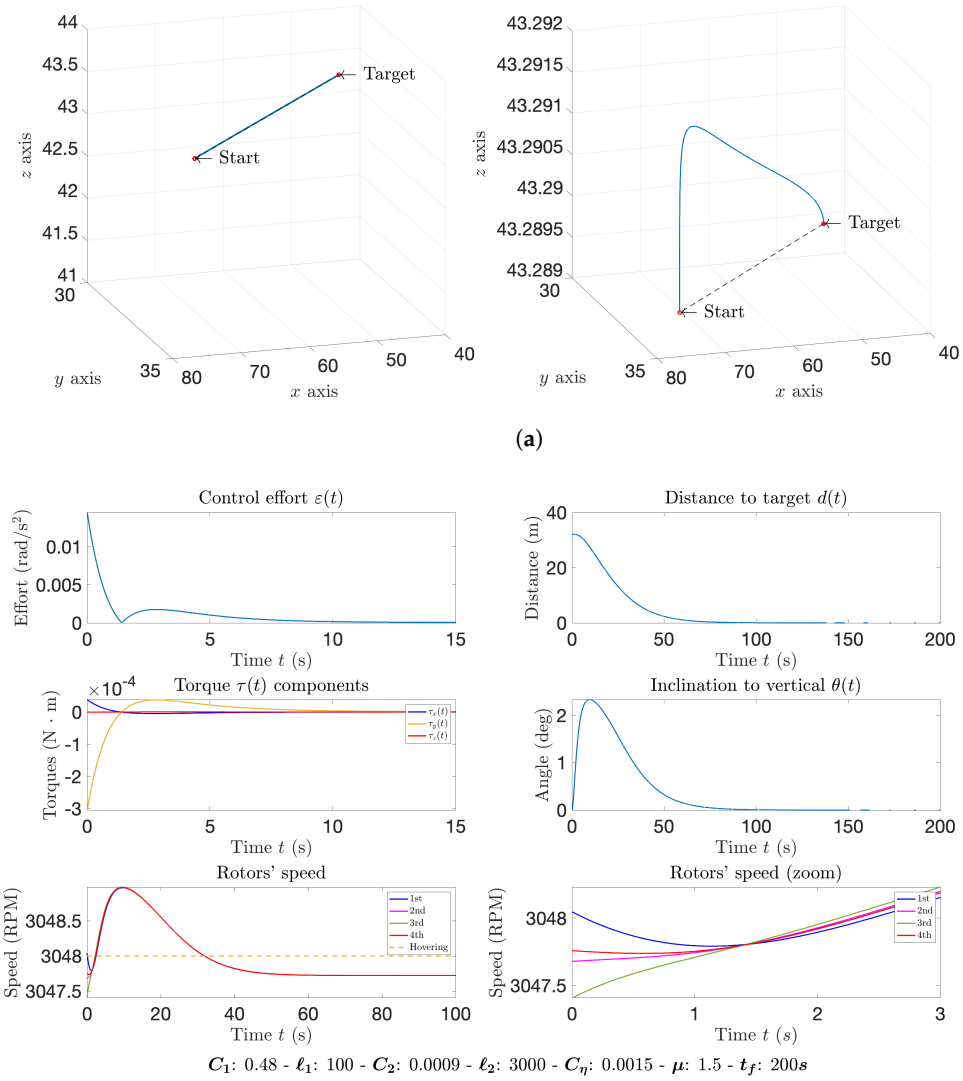
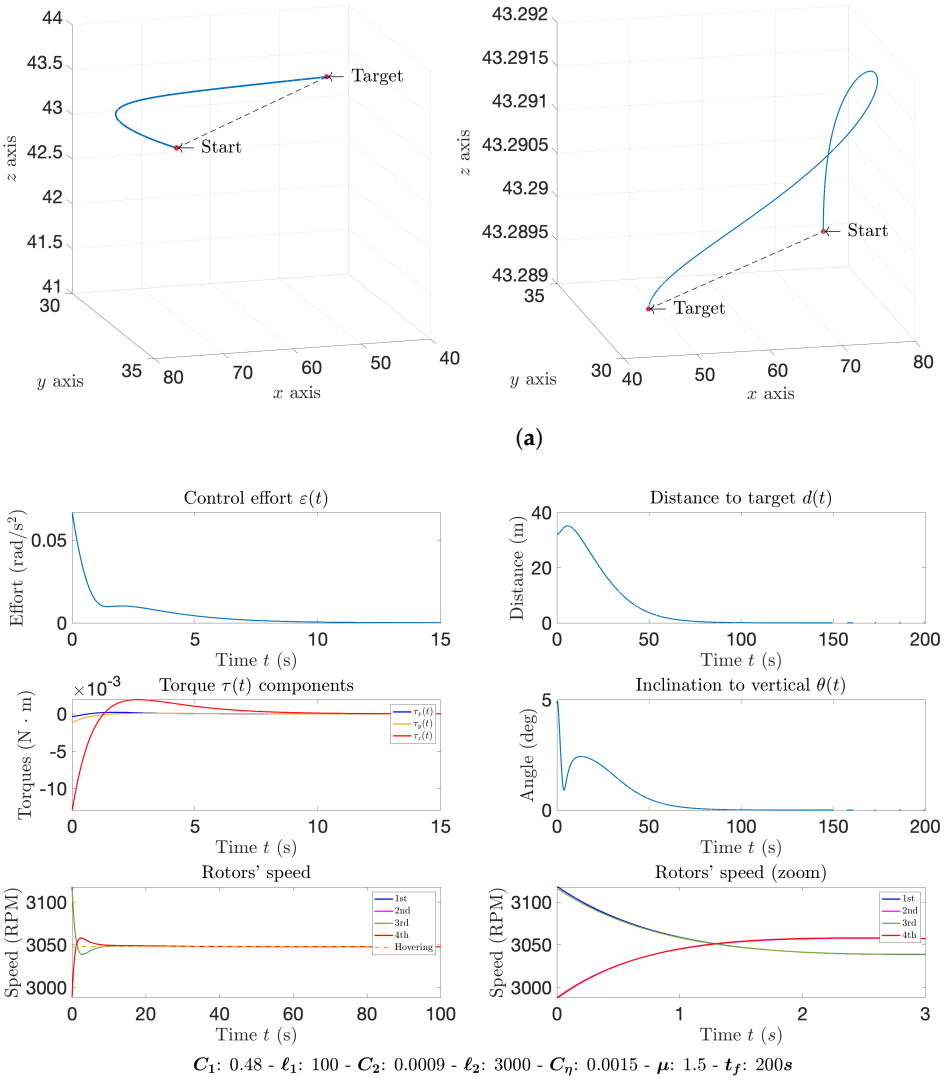


Figure 6. Experiment: *Horizontal* displacement, favorable initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.

The inclination-to-vertical panels tell that, in the simulation obtained with favorable initial conditions, the drone will tilt directly toward the right attitude (from $\theta = 0$) in order to fly toward the target and then, as soon as the drone approaches the target location, inclination goes back to zero. In the simulation pertaining to random initial conditions, the panel about angle of inclination shows that the controlled drone needs to take a countermeasure in order to reach the right attitude starting from an unfavorable one, after that it can navigate seamlessly towards the target.

The plots that report the distance from target show that in both simulations the drone manages to reach the target even in the case that it just needs to cover a purely horizontal displacement.



(**a**)

(**b**)

Figure 7. Experiment: *Horizontal* displacement, random initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.

3.7. Target and Drone's Initial Location Separated by a Vertical Displacement

In the simulation results illustrated by Figures 8 and 9, the start point locates above the target (with no horizontal displacement), hence the quadcopter just needs to descend vertically, which ideally may be achieved by decreasing the speed of rotors for a short period of time.

In the simulation corresponding to favorable initial conditions, whose results are illustrated in Figure 8, it can be seen from the inclination graph that the tilting of the quadcopter takes an almost null value, since no tilting is necessary to move vertically. For this reason, at the beginning of the simulation the RPMs of the propellers take close values to one another and such common value is lower than the hovering speed. As the quadcopter approaches the target, spinning velocities come closer to the hovering speed.

Even in this case (as in Section 3.6), the trajectory generated by the controller only slightly differs from the ideal trajectory. This effect may be noticed from the panel on the right side where a larger scale for the y axis has been utilized (the particular shape of such trajectory may only be appreciated using a larger scale for the axes since the discrepancy to the ideal trajectory is of the order of millimeters).

A further interesting aspect emerging from the discussed simulation result is that the control effort takes quite low values as opposed to the simulation results displayed so far. This effect may be explained by noticing that the drone does not need to vary its attitude, but only to slow down or speed up rotors' spinning. The same observation holds about the torque components that take quite low values compared to the previous simulations.

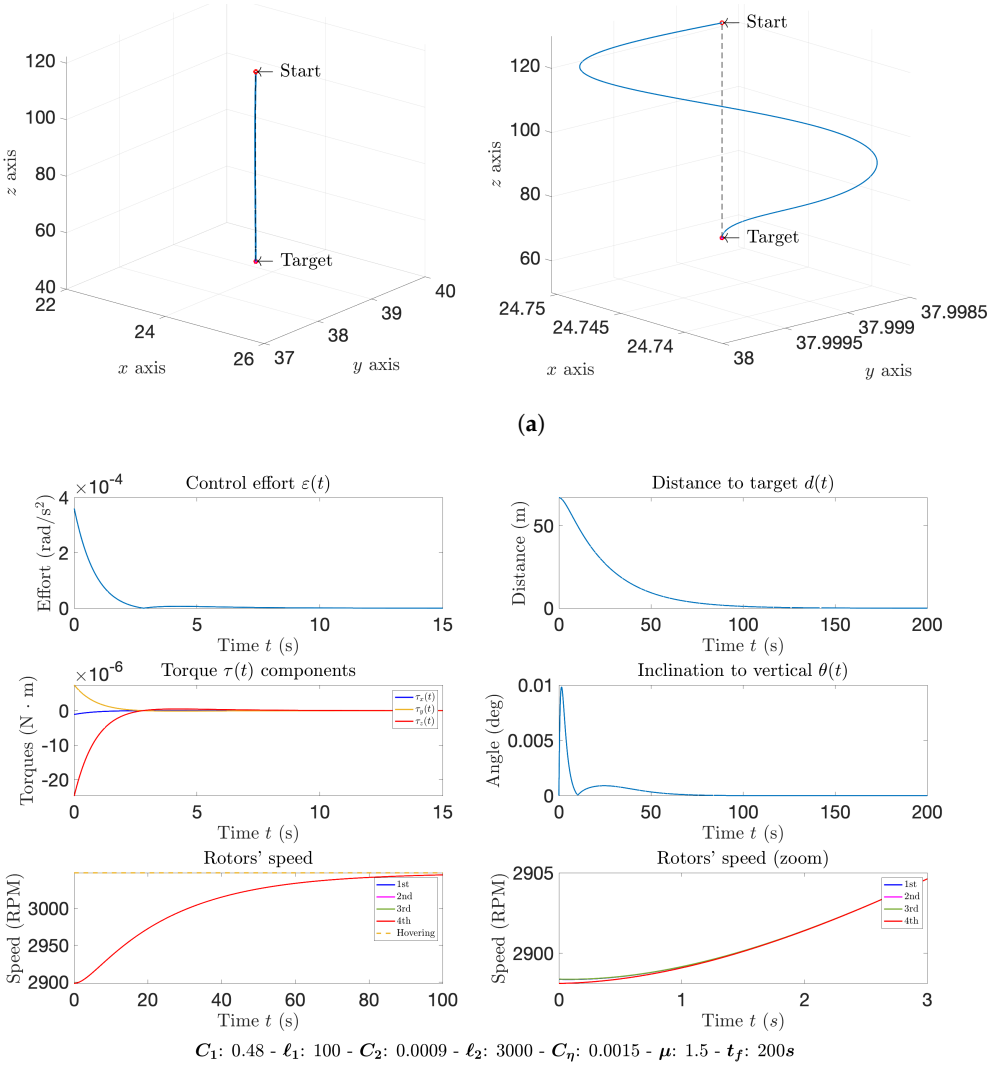


Figure 8. Experiment: Vertical ascension, favorable initial conditions. (**a**) Target approaching trajectory.; (**b**) Control figures.

Figure 9 displays numerical results on the same control endeavor obtained with random initial conditions.

In such case, it emerges from the RPMs plot that in the first fragment of simulation the controlled drone needs to vary its attitude, hence rotors' speeds take considerably different values from one propeller to another. After the first 10 s, such speeds become almost equal to one another and, as the drone gets closer to the target, the spinning velocities increase until the hovering speed is reached.

The inclination plot shows that the drone starts off its flight from an unfavorable initial attitude, therefore at first it tilts to the hovering attitude in order to move downward.

In this case, the control effort and torque components reach higher values with respect to the previous simulation. From this consideration, it is readily inferred how a change of attitude requires a significant effort to the propelling rotors.

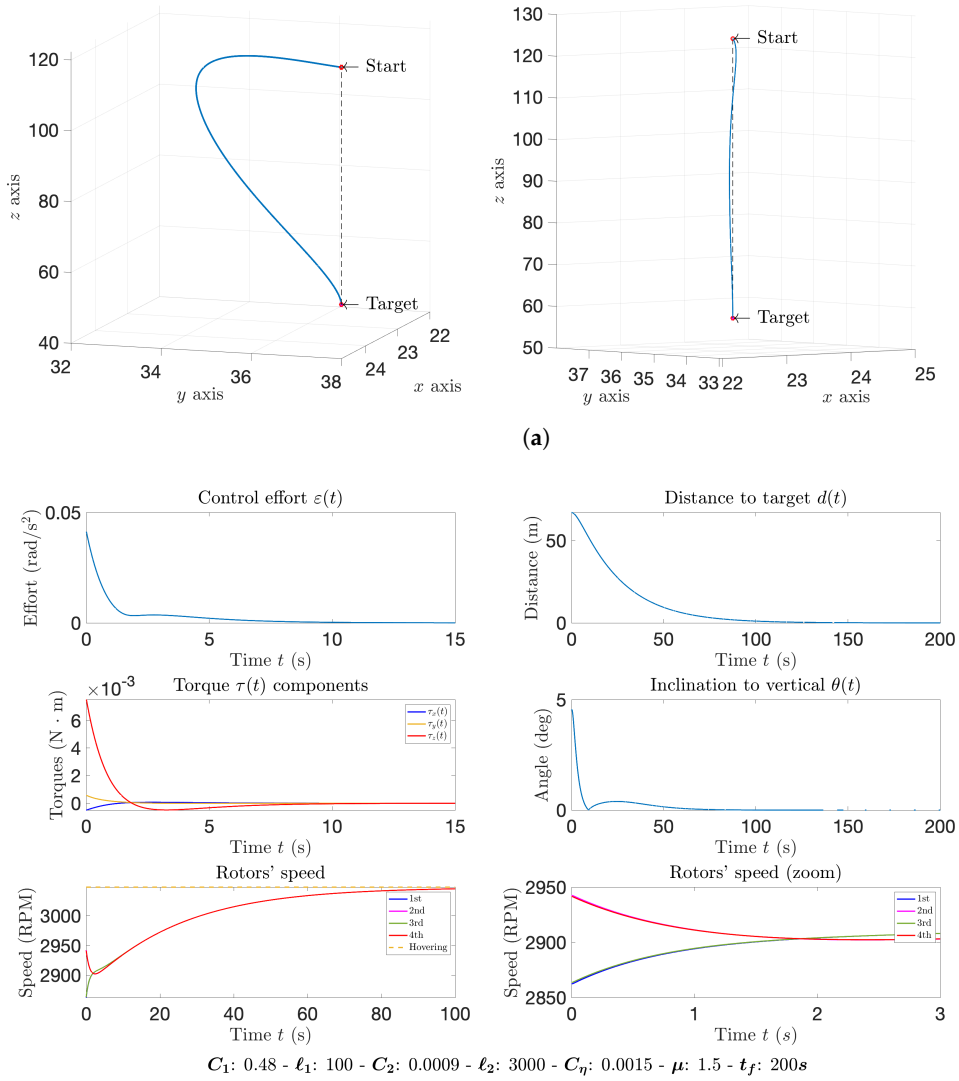


Figure 9. Experiment: Vertical ascension, random initial conditions. (**a**) Target approaching trajectory; (**b**) Control figures.

The distance-from-target plot shows that the drone manages to correctly reach the target in both simulations.

3.8. Flight under Impulsive Disturbance

Figures 10–12 show results of simulations performed to experiment with quadcopter navigation disturbed by impulsive forces. In order to effect such simulations, the acceleration equation from the quadcopter model (26) has been modified as follows:

$$\dot{v} = \frac{1}{2}\frac{b}{M_q}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)Re_z - \bar{g}e_z - \frac{1}{M_q}\Gamma v + \frac{F_{\text{imp}}}{M_q}, \tag{71}$$

where F_{imp} is a three-dimensional vector that represents an impulsive force acting on the drone. Denoting by $U(t)$ a unit-step function, an impulsive disturbance is represented by

$$F_{\text{imp}}(t) := \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} (U(t - T_{\text{hit}}) - U(t - T_{\text{hit}} - \Delta T_{\text{hit}})), \tag{72}$$

where f_x, f_y and f_z denote three force components randomly drawn from the interval $[-3.25,\ +3.25]$ (N), T_{hit} denotes the inception time of such disturbance and ΔT_{hit} denotes its duration. Notice that the values of the force components are chosen so that the entries of $\frac{F_{\text{imp}}}{M_{\text{q}}}$ range approximately in $[-5,\ +5]$ (m/s^2), which is approximately half of gravitational acceleration.

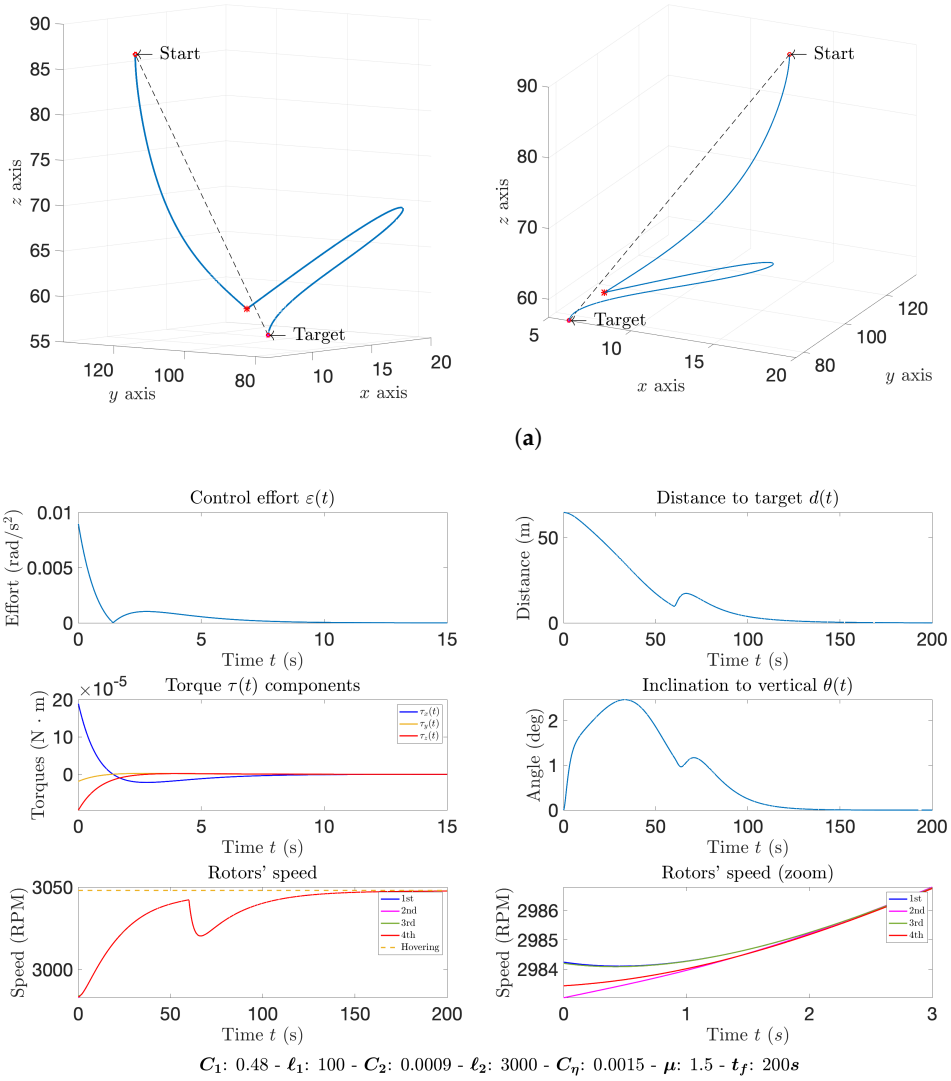


Figure 10. Experiment: Single impulsive hit. On the target approaching plot, a red asterisk denotes the point on the trajectory where the disturbance hits. (**a**) Target approaching trajectory; (**b**) Control figures.

In the numerical implementation, the duration of a disturbance has been set as $\Delta T_{\text{hit}} := h$, namely the sampling interval (as well as the shortest interval of time allowable in a time-sampled setting). In addition, all simulations described in the present subsection have been performed by choosing favorable initial conditions.

In the simulation whose outcomes are summarized in Figure 10, only one impulsive force acts on the drone. From the target approaching plots it is readily witnessed how, despite the drone has been pushed away by the impulsive force, the controller manages to get the drone on track to a viable journey to the target. At time T_{hit}, when the disturbance happens, the plots show a sudden change in the trend: the distance-to-target panel shows a sudden rise since the drone has been pushed away from its undisturbed trajectory, from the inclination values it is readily noticed that, due to the action of the forcing nuisance,

the drone varies its attitude, and from the control effort plot and torque components panel, it is inferred that the controller requests a supplementary effort to the rotors in order to stabilize the drone and steer it back toward the target.

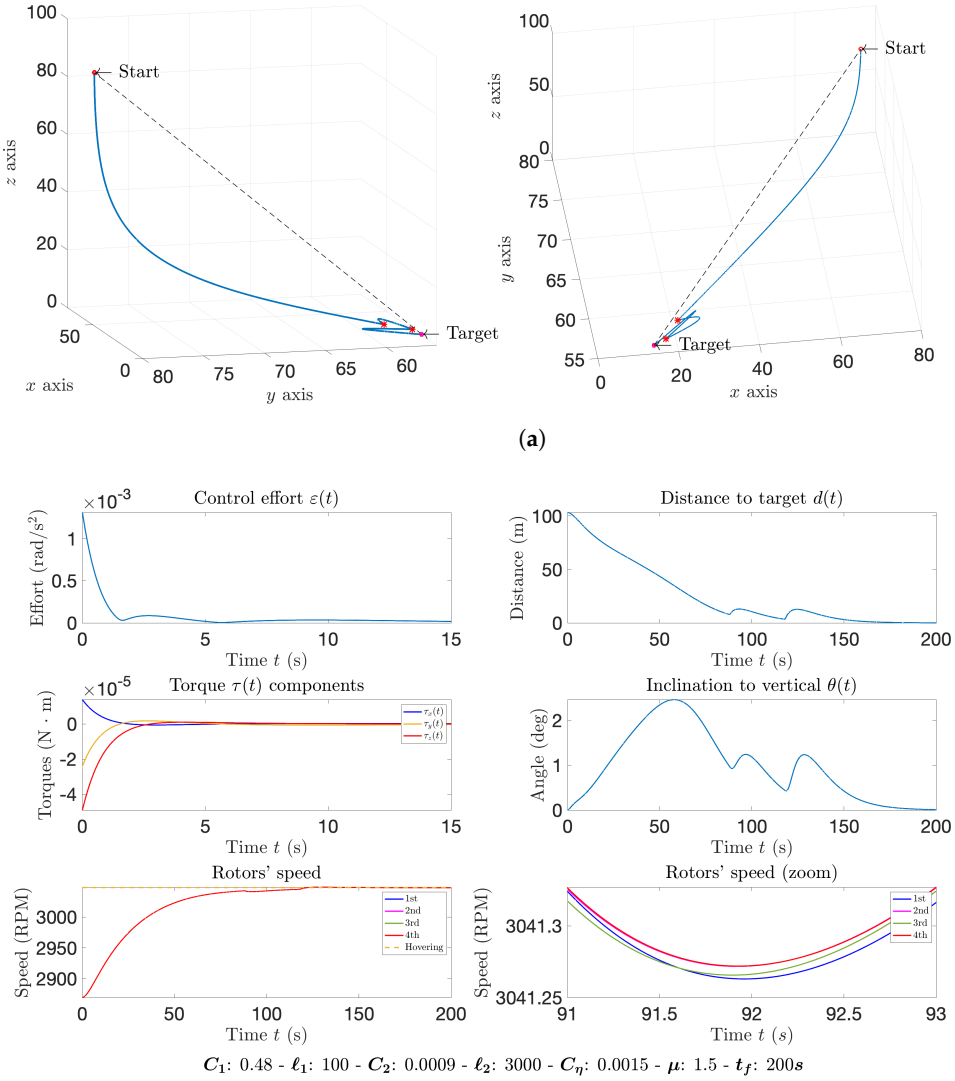


Figure 11. Experiment: Two impulsive forces. The red asterisks on the quadcopter trajectory represent the impact points. (**a**) Target approaching trajectory; (**b**) Control figures.

In the simulation whose outcomes are summarized in Figure 11, two impulsive forces act upon a quadcopter at two distinct instants. According to the obtained results, even in this case the drone is able to reach the target, as illustrated by the distance plot. In this case, since the drone is pushed away from its planned trajectory two times, all the plots exhibit two peaks that occur precisely when the forcing nuisance happens. Nonetheless, rotors' spinning velocity seems to be only slightly influenced by the disturbances, as, in general, small deviations in their values suffice to compensate a drone's attitude.

In a further interesting simulation, whose results are displayed in Figure 12, an impulsive disturbance hits after a drone has already reached the target. The displayed graphs show that the drone gets indeed pushed away from the target, but it eventually manages to make a comeback. The panel showing the values of the torques exerted by the thrusters on the body of the drone illustrates how such quantities are only slightly influenced by the disturbances (as well as the control effort). Indeed, a slight variation of the torques suffices to compensate the trajectory of a drone under the action of impulsive disturbances.

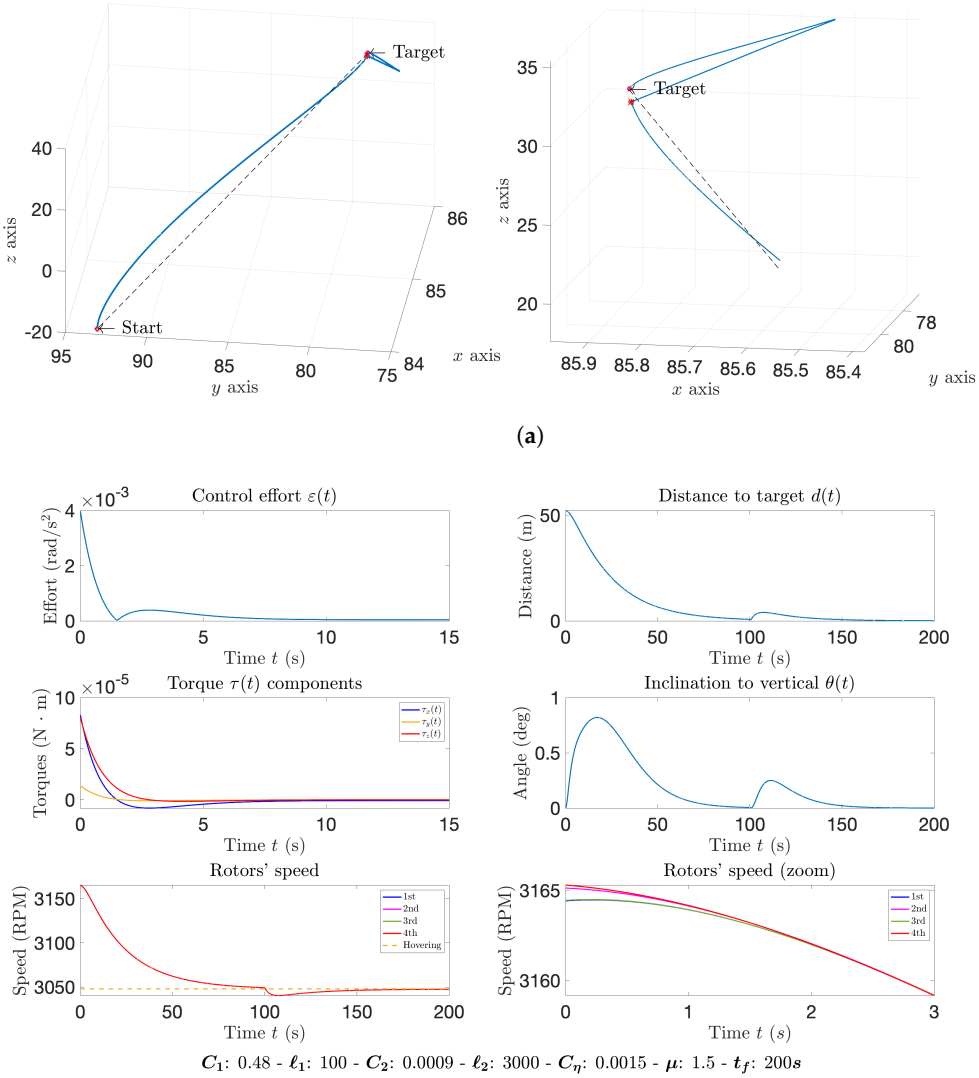
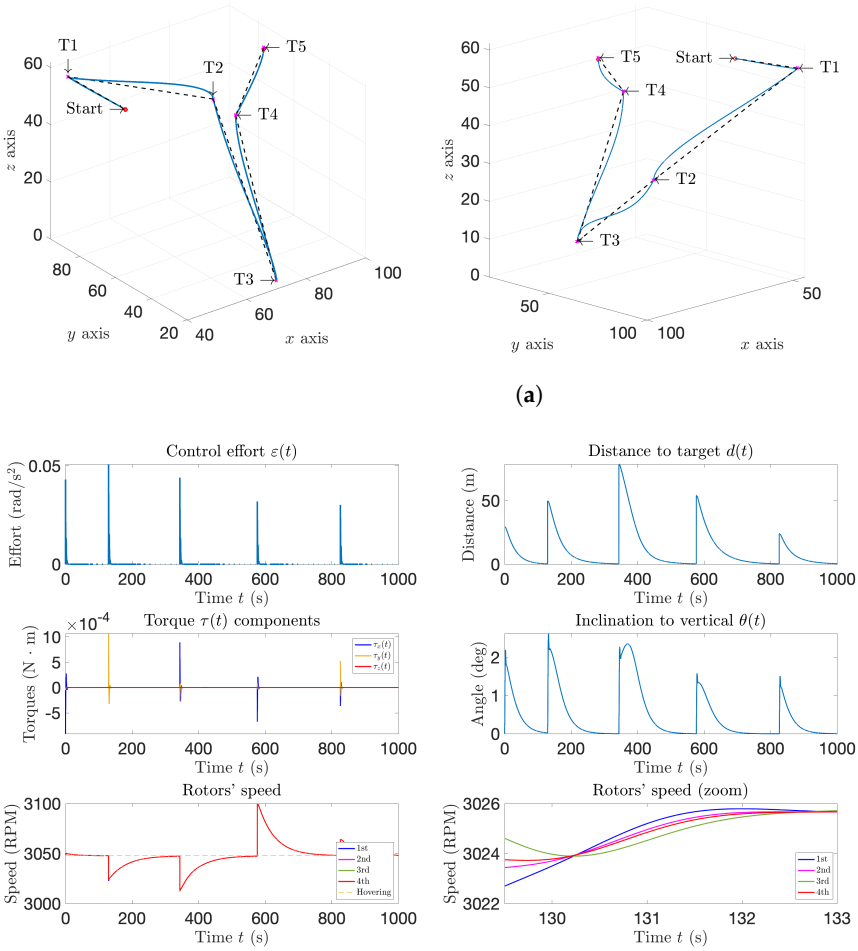


Figure 12. Experiment: Single impulsive force acting after the drone has reached the target. The hit point on the target-approaching panel is marked by a red asterisk. (**a**) Target approaching trajectory; (**b**) Control figures.

3.9. Chasing Shifting Targets

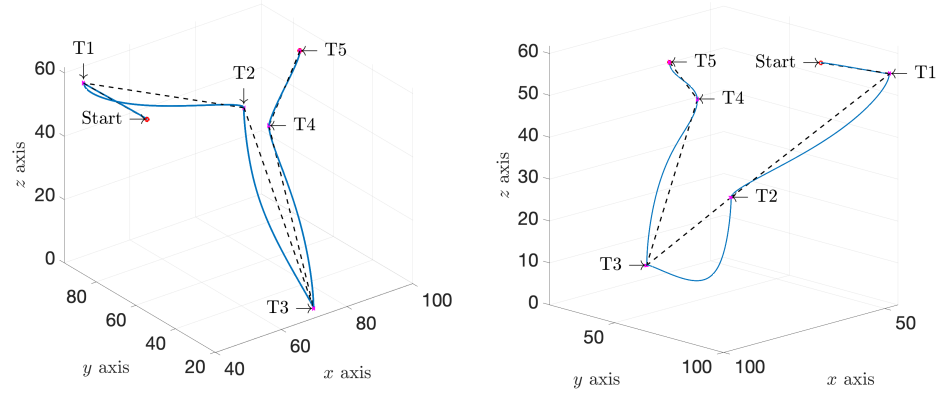
The final simulations in the present comprehensive series of experiments concern chasing a moving target. Two kinds of test have been performed: in the first kind of test, the target shifts after a drone gets sufficiently close to it, while in the second kind of test, the target shifts randomly in time even if the drone has not yet reached it. Both kinds of experiments have been repeated by varying the VARP parameters values. Initial conditions are favorable, namely initial angular speed and linear speed close to zero and initial attitude equal to the identity matrix.

In the simulations whose results are illustrated in Figures 13 and 14, the target shifts as soon as the quadcopter comes to a distance less than 30 cm from its current location. With the first set of VARP parameters, the quadcopter better follows the ideal trajectory, but it takes a long time to reach each target, as shown in Figure 13. Figure 14 shows the results of a simulation obtained with a second set of parameters. It is worth noting that the control figures present several peaks that occur as soon as a new target position manifests. In this case, the drone follows a wider trajectory compared to the ideal one, but it needs considerably lesser time to reach each target compared to the previous simulation.



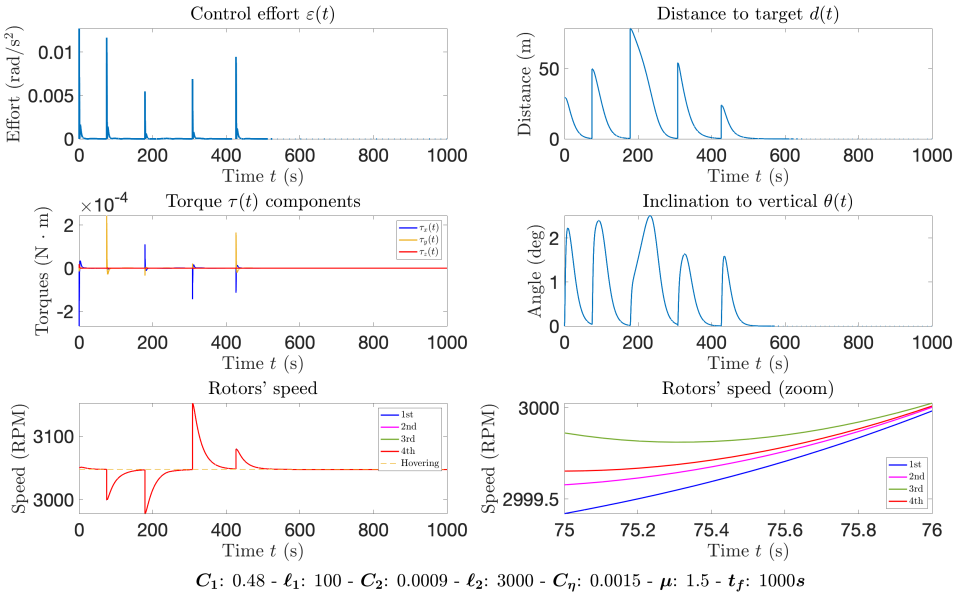
(**b**)

Figure 13. Experiment: Multiple shifting targets, first set of VARP parameters. In the target approaching plots, the dashed black lines represent the ideal trajectory which directly links the start point (current target) to the next target and all the targets have been labeled and marked by a magenta-colored cross. (**a**) Target approaching trajectory; (**b**) Control figures.



(**a**)

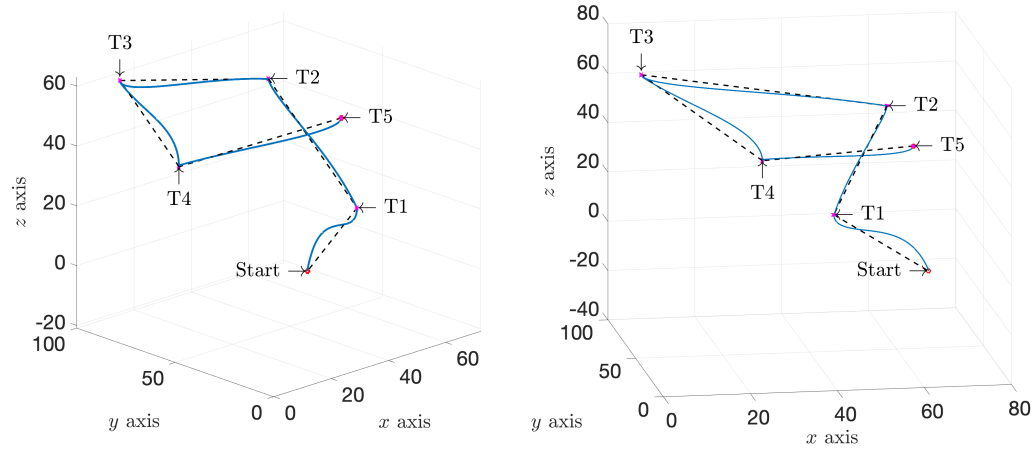
Figure 14. *Cont.*



(**b**)

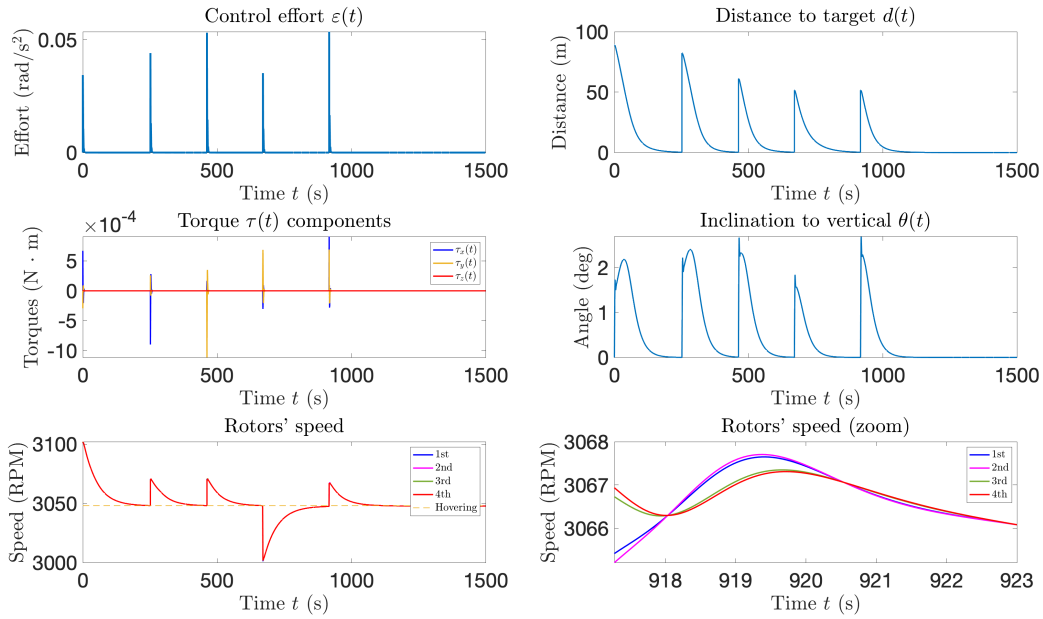
Figure 14. Experiment: Multiple shifting targets, second set of VARP parameters. (**a**) Target approaching trajectory; (**b**) Control figures.

The Figures 15 and 16 report the results of the second kind of test in which the target changes randomly through time. It is particularly interesting to analyze the behavior of the control algorithm between the start point and the target T1. It is readily noticed that as the quadcopter drone was heading toward the target T1, the latter suddenly shifted to a new location T2. Such an event causes the quadcopter to change its heading toward the new target. The peaks in the torque components (and hence in the control effort) are considerably higher than the average values of such figures, although very short in duration.



(**a**)

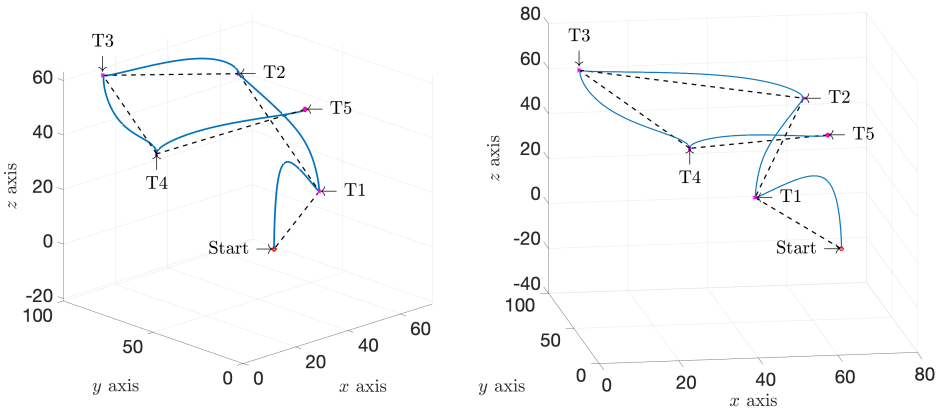
Figure 15. *Cont.*



(**b**)

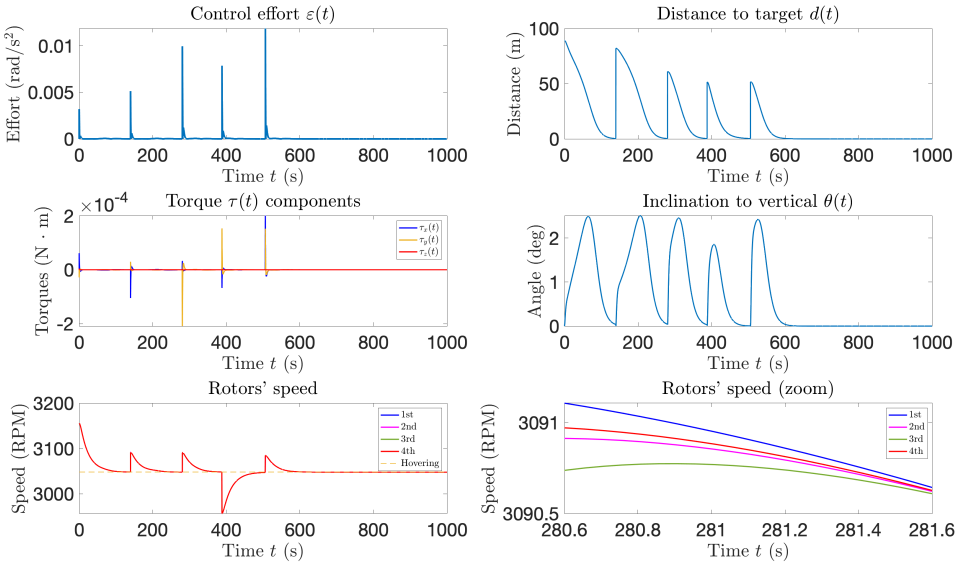
Figure 15. Experiment: Targets shifting randomly in time, first set of VARP parameters. (**a**) Target approaching trajectory; (**b**) Control figures.

As opposed to the previous tests, where a target shifted only *after* the quadcopter got sufficiently close to it, in this case the drone does not necessarily reach all the targets, but it strives to approach the one that is currently plugged into the virtual potential field. Such interesting simulation shows that even in the case of a shifting target the control method is effective. Such feature looks profitable in those applications where a drone is commanded to suddenly shift its destination.



(**a**)

Figure 16. *Cont.*



(**b**)

Figure 16. Experiment: Targets shifting randomly in time, second set of VARP parameters. (**a**) Target approaching trajectory; (**b**) Control figures.

4. Conclusions and Future Work

The VARP control theory that was taken as a reference in the present work, developed in [21], was originally designed to control a dynamical system representing a real-world wheeled robot. As we have seen in the present work, this theory may be extended to control dynamical systems evolving on larger-dimensional, possibly curved, state spaces.

The purpose of the present endeavor was to apply the VARP principle in the design of a novel quadcopter's attitude control law. Since the attitude of a quadcopter may be described by means of elements of the SO(3) manifold, we have generalized the VARP principle equations to control a second-order dynamical system whose state-space representation insists on the Riemannian manifold SO(3).

On the basis of a quadcopter model drawn from [32], the devised SO(3)-VARP control theory has been used to control the attitude of a quadcopter: given a desired attitude, the control law will provide a Lie-algebra-type control field to make a drone tilt toward the desired attitude. In particular, since the only term that can be acted upon to control a drone is the external mechanical torque, the VARP control method yields a particular value of the mechanical torque that the thrusters must produce in order to tilt the drone toward the desired attitude.

In order to achieve complete rotational and positional control of a quadcopter to make it fly from a start point to a desired target point, we have developed a *double VARP* control method. This contribution to the theory of Lie-group control class has been scrutinized through a comprehensive series of numerical experiments performed through specifically-tailored numerical algorithms.

The results of numerical tests have confirmed that the double VARP control method is able to efficiently steer a drone toward a desired target point in space enabling autonomous movement control. Indeed, results of simulations showed that the devised control law is able to drive a drone toward a target even in the event that it gets hit impulsively or if there happens a sudden shift of the target. This latter aspect looks interesting and promising in view of future developments since it could make it possible to drive a drone toward a pre-planned path defined by declaring successive target points in the virtual potential field.

The VARP control method is characterized by a set of parameters $(C_a, C_r, \ell_a, \ell_r)$, whose values influence its performance. From the commented simulations, it emerged that the used set of parameters are suitable in a wide range of situations.

The double VARP control method introduced in the present paper only utilizes target-points both in the $SO(3)$-VARP subsystem and in the $\mathbb{R}^3$-VARP subsystem. Future developments could provide a more involved virtual potential field which could include repulsive points, in order to represent obstacles in the space of maneuver and to make the devised control strategy capable of obstacle avoidance.

Building a more involved and informed virtual potential field could also lead to making the double VARP method *cooperative*, namely capable of controlling a fleet of two or more quadcopters flying coordinately.

A further important aspect to be developed in future endeavors concerns a formal proof of convergence of the devised control algorithm, perhaps based on Lyapunov-type analysis.

Author Contributions: Conceptualization, S.F., L.B. and F.P.; writing—original draft preparation, S.F., L.B. and F.P.; writing—review and editing, S.F.; supervision, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to gratefully thank Toshihisa Tanaka (TUAT—Tokyo University of Agriculture and Technology) for having hosted the authors L.B. and F.P. during an internship in January–March 2020 and for having invited the author S.F. as a visiting professor at the TUAT in February–March 2020.

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

References

1. Divan, A.; Kumar, A.S.; Kumar, A.J.; Jain, A.; Ravishankar, S. Fire detection using quadcopter. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 1–5. [CrossRef]
2. Rousseau, G.; Maniu, C.S.; Tebbani, S.; Babel, M.; Martin, N. Quadcopter-performed cinematographic flight plans using minimum jerk trajectories and predictive camera control. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 2897–2903. [CrossRef]
3. Selvarajan, A.; Yogaraju, H. Design and development of a quadcopter for landmine detection. In Proceedings of the 2020 IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, 27–29 September 2020; pp. 80–85. [CrossRef]
4. Connolly, C.; Grupen, R. The application of harmonic functions to robotics. *J. Robot. Syst.* **1993**, *10*, 931–946. [CrossRef]
5. Connolly, C.I.; Burns, J.B.; Weiss, R. Path planning using Laplace's equation. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; Volume 3, pp. 2102–2106.
6. Chang, D.; Shadden, S.; Marsden, J.; Olfati-Saber, R. Collision avoidance for multiple agent systems. In Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; pp. 539–543.
7. Waydo, S.; Murray, R.M. Vehicle motion planning using stream functions. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 2484–2491.
8. Chen, Y.B.; Luo, G.C.; Mei, Y.S.; Yu, J.Q.; Su, X.L. UAV path planning using artificial potential field method updated by optimal control theory. *Int. J. Syst. Sci.* **2016**, *47*, 1407–1420. [CrossRef]
9. Paul, T.; Krogstad, T.R.; Gravdahl, J.T. Modelling of UAV formation flight using 3D potential field. *Simul. Model. Pract. Theory* **2008**, *16*, 1453–1462. [CrossRef]
10. Rasekhipour, Y.; Khajepour, A.; Chen, S.; Litkouhi, B. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1255–1267. [CrossRef]
11. Shimoda, S.; Kuroda, Y.; Iagnemma, K. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2828–2833. [CrossRef]
12. Cao, X.; Chen, L.; Guo, L.; Han, W. AUV global security path planning based on a potential field bio-inspired neural network in underwater environment. *Intell. Autom. Soft Comput.* **2021**, *27*, 391–407. [CrossRef]
13. Koditschek, D.; Rimon, E. Robot navigation functions on manifolds with boundary. *Adv. Appl. Math.* **1990**, *11*, 412–442. [CrossRef]

14. Rimon, E.; Koditschek, D. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.* **1992**, *8*, 501–518. [CrossRef]
15. Bigelli, L.; Polenta, F.; Fiori, S. Virtual attractive-repulsive potentials control theory: A review and an extension to Riemannian manifolds. *Symmetry* **2022**, *14*, 257. [CrossRef]
16. Lee, M.C.; Park, M.G. Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), Kobe, Japan, 20–24 July 2003; Volume 2, pp. 735–740. [CrossRef]
17. Wang, M.; Su, Z.; Tu, D.; Lu, X. A hybrid algorithm based on Artificial Potential Field and BUG for path planning of mobile robot. In Proceedings of the 2013 International Conference on Measurement, Information and Control, Harbin, China, 16–18 August 2013; Volume 2, pp. 1393–1398. [CrossRef]
18. Ren, J.; McIsaac, K.A.; Patel, R.V. Modified Newton's method applied to potential field-based navigation for mobile robots. *IEEE Trans. Robot.* **2006**, *22*, 384–391. [CrossRef]
19. Li, G.; Yamashita, A.; Asama, H.; Tamura, Y. An efficient improved artificial potential field based regression search method for robot path planning. In Proceedings of the 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, 5–8 August 2012; pp. 1227–1232. [CrossRef]
20. Levine, H.; Rappel, W.J.; Cohen, I. Self-organization in systems of self-propelled particles. *Phys. Rev. E* **2000**, *63*, 017101. [CrossRef]
21. Nguyen, B.; Chuang, Y.L.; Tung, D.; Hsieh, C.; Jin, Z.; Shi, L.; Marthaler, D.; Bertozzi, A.; Murray, R. Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the Caltech MVWT. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 1084–1089.
22. Leonard, E.; Fiorelli, E. Virtual leaders, artificial potentials and coordinated control of groups. In Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, FL, USA, 4–7 December 2001; pp. 2968–2973.
23. Nair, S.; Leonard, N. Stable synchronization of rigid body networks. *Netw. Heterog. Media* **2007**, *2*, 595–624. [CrossRef]
24. Kulumani, S.; Taeyoung, L. Constrained geometric attitude control on SO(3). *Int. J. Control Autom. Syst.* **2017**, *15*, 2796–2809. [CrossRef]
25. Mayhew, C.; Teel, A. Synergistic potential functions for hybrid control of rigid-body attitude. In Proceedings of the 2011 IEEE American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 875–880. [CrossRef]
26. Bullo, F.; Murray, R. Tracking for fully actuated mechanical systems: A geometric framework. *Automatica* **1999**, *35*, 17–34. [CrossRef]
27. Koditschek, D. The application of total energy as a Lyapunov function for mechanical control systems. *Contemp. Math.* **1989**, *97*, 131–157.
28. Fiori, S.; Cervigni, I.; Ippoliti, M.; Menotta, C. Extension of a PID control theory to Lie groups applied to synchronising satellites and drones. *IET Control Theory Appl.* **2020**, *14*, 2628–2642. [CrossRef]
29. Fiori, S.; Del Rossi, L. Minimal control effort and time Lie-group synchronisation design based on proportional-derivative control. *Int. J. Control* **2022**, *95*, 138–150. [CrossRef]
30. Simha, A.; Tallam, M.; Shankar, H.N.; Muralishankar, R.; Simha, H.N.L.N. Adaptive attitude control of the spherical drone on SO(3). In Proceedings of the 2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 August 2016; pp. 90–94. [CrossRef]
31. Yang, S.; Pei, H. The solution of drone attitudes on Lie groups. In Proceedings of the 2020 International Conference on Advanced Robotics and Mechatronics (ICARM), Shenzhen, China, 18–21 December 2020; pp. 276–281. [CrossRef]
32. Fiori, S. Model formulation over Lie groups and numerical methods to simulate the motion of gyrostats and quadrotors. *Mathematics* **2019**, *7*, 935. [CrossRef]
33. Tarsi, A.; Fiori, S. Lie-group modeling and numerical simulation of a helicopter. *Mathematics* **2021**, *9*, 2682. [CrossRef]
34. Roberson, R.E.; Schwertassek, R. *Dynamics of Multibody Systems*; Springer: Berlin/Heidelberg, Germany, 1988.
35. Cheng, H.; Gupta, K.C. An historical note on finite rotations. *J. Appl. Mech.* **1989**, *56*, 139–145. [CrossRef]
36. Rodrigues, O. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J. Math. Pures Appl.* **1840**, *5*, 380–440.
37. Pei, Y.; Cai, X.; Song, K.; Liu, R.; Li, J. Identification method of main road traffic congestion situation in cold-climate cities based on potential energy theory and GPS data. *Symmetry* **2022**, *14*, 227. [CrossRef]
38. Huang, Y.; Tang, J.; Lao, S. UAV group formation collision avoidance method based on second-order consensus algorithm and improved artificial potential field. *Symmetry* **2019**, *11*, 1162. [CrossRef]
39. Hablani, H.B. Attitude commands avoiding bright objects and maintaining communication with ground station. *J. Guid. Control Dyn.* **1999**, *22*, 759–767. [CrossRef]
40. Becker, M.; Sampaio, R.; Bouabdallah, S.; de Perrot, V.; Siegwart, R. In-flight collision avoidance controller based only on OS4 embedded sensors. *J. Braz. Soc. Mech. Sci. Eng.* **2012**, *34*, 295–297. [CrossRef]
41. Ogunrinde, R.B.; Fadugba, S.E.; Okunlola, J.T. On some numerical methods for solving initial value problems in ordinary differential equations. *IOSR J. Math. (IOSRJM)* **2012**, *1*, 25–31.

42. Lopez, I.; McInnes, C.R. Autonomous rendezvous using artificial potential function guidance. *J. Guid. Control Dyn.* **1995**, *18*, 237–241. [CrossRef]
43. Xiao, X.; Fan, Y.; Dufek, J.; Murphy, R. Indoor UAV localization using a tether. In Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA, 6–8 August 2018; pp. 1–6. [CrossRef]
44. Outamazirt, F.; Fu, L.; Lin, Y.; Abdelkrim, N. A new SINS/GPS sensor fusion scheme for UAV localization problem using nonlinear SVSF with covariance derivation and an adaptive boundary layer. *Chin. J. Aeronaut.* **2016**, *29*, 424–440. [CrossRef]