

Università Politecnica delle Marche



Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica e dell'Automazione

**Design of a Virtual Attractive Repulsive
Potentials-based control method on Riemannian
Manifolds and Lie Groups**

**Progettazione di un metodo di controllo basato sulla teoria dei
Potenziali Virtuali Attrattivi e Repulsivi sviluppata su Manifold
Riemanniani e Gruppi di Lie**

Relatore:
Chiar.mo Prof. Simone Fiori

Candidato:
Federico Polenta

Correlatore:
Prof. Toshihisa Tanaka

Anno Accademico 2019/2020

Design of a Virtual Attractive Repulsive Potentials-based control method on Riemannian Manifolds and Lie Groups

Federico Polenta

Abstract

The aim of this document is to design a control law based on VARP (Virtual Attractive Repulsive Potentials) control theory developed on two manifolds. VARP Theory, initially designed for the control of a dynamic system whose state space is \mathbb{R}^2 , will be applied on a generic second-order dynamic system whose state equations are formulated on the tangent complex of \mathbb{S}^2 . Afterwards, the control method will be tailored for the control of a quadcopter whose state equations are formulated on the $\text{SO}(3)$ Lie group. Combining an instance of VARP method for the control of the drone's attitude (SO3-VARP) and an instance of VARP method for the drone's space-position control (\mathbb{R}^3 -VARP) it will arise the Double VARP (D-VARP) which aims to make the drone move autonomously in the space.

Supervisors: Chiar.mo Prof. Simone Fiori and Dr. Toshihisa Tanaka
Marche Polytechnic University

Progettazione di un metodo di controllo basato sulla teoria dei Potenziali Virtuali Attrattivi e Repulsivi sviluppata su Manifold Riemanniani e Gruppi di Lie

Federico Polenta

Sommario

Lo scopo di questo documento è presentare e testare numericamente la teoria di controllo VARP (Virtual Attractive Repulsive Potentials) sviluppata su due manifold. Nella prima parte del documento, il metodo di controllo verrà applicato su un generico sistema dinamico del secondo ordine le cui equazioni di stato sono formulate sul complesso tangente associato ad un manifold \mathbb{S}^2 . Nella seconda parte del documento, il metodo di controllo verrà applicato su un drone a quattro eliche le cui equazioni sono formulate sul gruppo di Lie $SO(3)$. Combinando un'istanza del metodo VARP per il controllo dell'assetto del drone (SO3-VARP) e un'istanza del metodo VARP per il controllo della sua posizione (\mathbb{R}^3 -VARP) nascerà il Doppio VARP (D-VARP) che avrà lo scopo di far muovere il drone autonomamente nello spazio.

Contents

List of Figures	v
List of Tables	vii
1 Introduction	1
2 VARP Control Theory	3
2.1 Virtual Potentials for swarming models in biology	3
2.2 VARP Control Theory	4
3 VARP Control Theory on \mathbb{R}^2	6
3.1 Kelly robot	6
3.2 VARP Control Theory applied to the Kelly robot	8
3.3 Numerical schemes for the simulation of a dynamic system's evolution . . .	11
3.3.1 Forward Euler Method (fEul)	11
3.3.2 Explicit Fourth Order Runge-Kutta Method (eRK4)	12
3.4 Simulations and results of VARP Method	14
3.4.1 First Test	15
3.4.2 Second Test	18
3.4.3 Third Test	20
3.4.4 Fourth Test	23
3.4.5 Fifth Test	25
4 VARP Control Theory on the Riemannian Manifold \mathbb{S}^2	28
4.1 Generalities on Riemannian Manifolds	28
4.2 Extension of the VARP Principle to Riemannian Manifolds	29
4.3 Euler scheme for the simulation of the M-VARP Control Method	31
4.4 Simulations and results of S2-VARP Method	34
4.4.1 First Test	34
4.4.2 Second Test	35
4.4.3 Third Test	36
4.4.4 Fourth Test	37

4.4.5	Fifth Test	38
5	VARP Control Theory on $\mathbf{SO}(3)$ Manifold	39
5.1	Lie Groups and Algebras definitions and properties	39
5.2	Manifold $\mathbf{SO}(3)$ and his Lie Algebra $\mathfrak{so}(3)$	41
5.3	VARP extension to the $\mathbf{SO}(3)$ Manifold	42
5.4	Quadcopter model	45
5.5	Control field design by <i>dynamics replacement</i> and error feedback	46
5.6	Design of a VARP-based control law for a quadcopter drone	47
5.7	Physical realizability and control effort analysis	49
5.8	Case-Study: Single attractive/repulsive point	51
5.9	Numerical integration of the Initial Value Problem	52
6	Double VARP Control for Quadcopter Motion in the Space	54
6.1	Double VARP Control Theory	54
6.2	Rotors' speed	57
6.3	Simulations and results of D-VARP Method	58
6.3.1	First Test	60
6.3.2	Second Test	63
6.3.3	Third Test	64
6.3.4	Fourth Test	65
6.3.5	Fifth Test	66
6.3.6	Sixth Test	67
6.3.7	Seventh Test	68
6.3.8	Eighth Test	70
6.3.9	Ninth Test	71
7	Conclusions and Future Works	72
	Bibliography	75

List of Figures

2.1	Coordinate system to describe position and velocity of the vehicles.	5
3.1	Kelly vehicle at the Caltech University. Adapted from [1].	6
3.2	Kelly's forces and geometry scheme.	7
3.3	First test - Path and potential field.	16
3.4	First test - Target approaching.	16
3.5	First test - Fans forces, orientation and distance over time.	17
3.6	Second test - Path and potential field.	18
3.7	Second test - Target approaching.	19
3.8	Second test - Fans forces, orientation and distance over time.	19
3.9	Third test - Path and potential field.	21
3.10	Third Test - Target approaching.	21
3.11	Third test - Fans forces, orientation and distance over time.	22
3.12	Fourth test - Path and potential field.	23
3.13	Fourth test - Target approaching.	24
3.14	Fourth test - Fans forces, orientation and distance over time.	24
3.15	Fifth test - Path and potential field.	26
3.16	Fifth test - Target approaching.	26
3.17	Fifth test - Fans forces, orientation and distance over time.	27
4.1	First test - Path planning using a corridor, $\alpha = 0.2$	34
4.2	Second test - Path planning using a corridor, $\alpha = 4$	35
4.3	Third test - Barrier bypassing.	36
4.4	Fourth test - Obstacle avoidance.	37
4.5	Fifth test - Barrier bypassing and target approaching.	38
5.1	Quadcopter. Generated by Yihang Zhang's MATLAB script.	46
6.1	First test - Target approaching from two different perspectives.	61
6.2	First test - Quadcopter's flight overview.	61
6.3	First test - Rotors' speed in the first instants.	62
6.4	Second test - Target approaching from two different perspectives.	63

6.5	Second test - Quadcopter's flight overview.	63
6.6	Third test - Target approaching from two different perspectives.	64
6.7	Third test - Quadcopter's flight overview.	64
6.8	Fourth test - Target approaching from two different perspectives.	65
6.9	Fourth test - Quadcopter's flight overview.	65
6.10	Fifth test - Target approaching from two different perspectives.	66
6.11	Fifth test - Quadcopter's flight overview.	66
6.12	Sixth test - Target approaching from two different perspectives.	67
6.13	Sixth test - Quadcopter's flight overview.	67
6.14	Seventh test - Target approaching from two different perspectives.	69
6.15	Seventh test - Quadcopter's flight overview.	69
6.16	Eighth test - Target approaching from two different perspectives.	70
6.17	Eighth test - Quadcopter's flight overview.	70
6.18	Ninth test - Target approaching from two different perspectives.	71
6.19	Ninth test - Quadcopter's flight overview.	71

List of Tables

3.1	Potentials for the first VARP test.	15
3.2	Parameters for the first VARP test.	15
3.3	Parameters for the second VARP test.	18
3.4	Potentials for the third VARP test.	20
3.5	Parameters for the third VARP test.	20
3.6	Parameters for the fourth VARP test.	23
3.7	Potentials for the fifth VARP test.	25
3.8	Parameters for the fifth VARP test.	25
4.1	Parameters for the first S2-VARP test.	34
4.2	Parameters for the second S2-VARP test.	35
4.3	Parameters for the third S2-VARP test.	36
4.4	Parameters for the fourth S2-VARP test.	37
4.5	Parameters for the fifth S2-VARP test.	38
5.1	Summary table of the OS4 Mini-VTOL quadcopter parameters.	45
6.1	Parameters for D-VARP tests.	59

Introduction

Most of the control problems faced nowadays are based on positioning, on path-planning and obstacles-avoidance. Virtual potentials can be very useful and effective to solve several control problems as they are very versatile and easily applicable. In order to control a particular dynamic system, it is necessary to build a virtual potential field using one of many possible mathematical frameworks. The artificial potential field could be constructed with the use of harmonic functions and Laplace's equations [2, 3], artificial gyroscopic forces [4] and stream functions from fluid dynamics [5].

The aim of this report is to design an artificial field in connection to a control theory known as VARP (Virtual Attractive-Repulsive Potentials). The versatility of this theory makes it straightforward to be used in several different fields, such as biology [6] or vehicle coordination [7]. As shown in this document, the VARP method will be applied in order to control different dynamical systems whose state-space equations belong to different mathematical groups. The high versatility of this theory can be seen in the last part of this document: in order to control a quadcopter, we used the VARP method twice for two different purposes, regulating attitude and the position of a quadcopter drone.

Quadcopters are an important class of UAVs (Unmanned Aerial Vehicles) which are aircrafts piloted by remote control or onboard computers. In particular, quadcopters are 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 the drone as stable and independent from any external factor as possible. Nowadays, quadcopters are widely used for professional purposes such as photography, filming or rescue operation. Since they are easy to build and inexpensive, quadcopters can be found on the market also for non-professional or semi-professional purposes. As quadcopters are used more and more for a wide variety of purposes, much research has been carried out in order to find a method to adequately control their attitude and position.

From the main solutions developed, we recall a trajectory planning method which exploits the differential flatness of the vehicle dynamics considering a six degree of freedom (6DoF) model of the quadrotor [8], a coordinated motion which uses mathematical programming techniques, in particular a combination of linear and integer programming (MILP) [9, 10], randomized algorithms for path planning [11], techniques that use graph searching and

surface covering [12], convex safe regions generations with IRIS and mixed-integer convex optimization [13] and also Fuzzy-based controllers [14, 15] have been developed.

This document is organized as follows: Chapter 2 describes the VARP method as it was introduced in [6] and [1], detailing the peculiarities and the notations used in the following chapters.

Chapter 3 introduces the Kelly, a robot whose state equations live in \mathbb{R}^2 , used by the Caltech University for testing the VARP Principle and explains how it is possible to discretize this control method in order to simulate his effectiveness on Kelly. We will report the tests made on Kelly by varying the VARP parameters, in order to understand their dynamics and behavior. The simulations will be carried out using two different discretizing methods, namely, fEul and eRK4.

Chapter 4 recalls some generalities on Riemannian manifolds and considers a generic second-order dynamic system whose state equations are formulated on the tangent bundle of a manifold M . This chapter illustrates how the VARP method can be extended in order to control a dynamical system whose state equation lives in a \mathbb{S}^2 manifold M-VARP. This control problem will be discretized using a fEul based on Riemannian manifolds, which will permit us to carry out simulations in a \mathbb{S}^2 manifold.

Chapter 5 recalls some notations about Lie groups and their properties. Then, it is explained how the VARP can be extended to a Lie group, the $SO(3)$ manifold, in order to control a quadcopter whose state equations live in $SO(3)$ and its algebra. The quadcopter equations and parameters are taken from [16]. Thus, it is explained the technique used to design the control field, based on dynamic replacement and error feedback. All the steps required to obtain the VARP-based control law for a quadcopter drone are reported, taking into account the physical realizability of this control method. A particular attention will be dedicated to the control effort and rotors speed analysis. Also this control law will be discretized using a fEul based on Lie groups in order to accomplish simulations in the $SO(3)$ manifold.

Chapter 6 details a control theory based on using the VARP control theory twice and concurrently. In particular, we will introduce two instances of VARP. An instance $VARP_1$ will serve to stabilize the attitude of a drone during flight. This instance is very important in order to maintain the drone in a proper attitude and not allow it to tip over. An instance $VARP_2$ will fix the drone attitude in such a way that it is steered toward a pre-defined target in the space. It is also defined an evolution law to control the vertical positioning of the drone during target approaching. At the end of this chapter some tests will be carried out to prove the efficiency of the Double VARP.

Chapter 7 completes this document, focusing on conclusion and future works.

VARP Control Theory

The Virtual Attractive-Repulsive Potential (VARP) control theory is based on the construction of a virtual potential field made of virtual points to which will be assigned both an attractive and a repulsive potential. The points on the virtual field could represent obstacles, target points or even any other object in the considered state-space. In this document we will demonstrate that it is possible to use the VARP control theory for the control of different dynamic systems with different state space. Therefore, the virtual points in the virtual potential field will not necessarily represent 3D space points but could also represent whatever belongs to the particular state-space considered. Such virtual potentials arise in swarming models in biology [6] and from these biological applications the VARP theory get ideas for the coordination and motion planning of a large group of vehicles. We will first introduce the biological model [6] from which the VARP theory arise, then the latter will be reported and explained as it was designed in [1].

2.1 Virtual Potentials for swarming models in biology

Virtual potentials provide a convenient framework for autonomous vehicle control and path planning. Such potentials arise in swarming models in biology, like the discrete particle model proposed by the Levine-Rappel-Cohen group [6] that will be now recalled.

Given N particles labelled $i = 1, \dots, N$, the equations of the model proposed in [6] that control the motion of each particle are:

$$m_i \frac{d\vec{v}_i}{dt} = \alpha \hat{f}_i - \beta \vec{v}_i - \vec{\text{grad}}V, \quad (2.1)$$

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i, \quad (2.2)$$

where m_i is the mass of each particle, x_i and v_i its position and velocity, respectively. Each particle experiences a self propelling force \vec{f}_i with fixed magnitude α . To prevent the particles from reaching large speeds, a friction force with coefficient β was introduced. In addition, each particle is subjected to an attractive force (it depends only on the distance from one particle to the others) which is characterized by an interaction range ℓ_a . This force

is responsible for the aggregation of the particles. To prevent a collapse of the aggregate, a shorter-range repulsive force was introduced with interaction range ℓ_r . The authors of [6] checked that qualitative results are independent of the explicit form of the interaction potential, hence they have chosen an exponentially decaying interaction:

$$V := \sum_{j, j \neq i} C_a \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|}{\ell_a}\right) - \sum_{j, j \neq i} C_r \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|}{\ell_r}\right), \quad (2.3)$$

where C_a , C_r determine the strength of the attractive and repulsive force, respectively. Note that Levine's model is based on only one pair of parameters (C_a and C_r) while in future works we will see that it could be convenient to define a value of C_a and C_r for each of the obstacles and targets.

From this model, designed for collective motion in biology, arises the VARP theory for the control and coordination of a large group of vehicles.

2.2 VARP Control Theory

After a brief recall in the previous section of the Levine-Rappel-Cohen model applied in biology, we can introduce the VARP control theory.

Given an i -th vehicle and N agents labeled $j = 1, \dots, N$, consider the following general coupled equations of motion:

$$\begin{cases} \frac{d\vec{z}_i}{dt} &= \vec{w}_i, \\ m_i \frac{d\vec{w}_i}{dt} &= \alpha \hat{\xi}_i - \beta \vec{w}_i - \text{grad}_{\vec{z}_i} \sum_{j, j \neq i}^N [V_j^r(d(\vec{z}_i, \vec{q}_j)) - V_j^a(d(\vec{z}_i, \vec{q}_j))], \end{cases} \quad (2.4)$$

where:

- $\hat{\xi}$ and $\hat{\zeta}$ are unit vectors referred to a reference frame attached to each i -th vehicle rotated with respect to the inertial reference frame $\hat{x} - \hat{y}$ by an angle θ (see Figure 2.1);
- $\vec{z}_i = (z_{x_i}, z_{y_i})$ denotes the position of the i -th vehicle;
- \vec{w}_i is the velocity of the i -th vehicle;
- m_i is the mass of the i -th vehicle;
- $d(\vec{z}_i, \vec{q}_j) = \|\vec{z}_i - \vec{q}_j\|$ is the distance between the i -th vehicle and the j -th element which is at the position $\vec{q}_j = (q_{x_j}, q_{y_j})$, which could be an obstacle, a target point or another vehicle. In the case of multiple vehicle control, each vehicle other than the i -th, will be considered as an obstacle for the i -th vehicle, therefore the position of the i -th vehicle will be denoted as \vec{z}_i while the position of the j -th obstacle will be denoted as \vec{q}_j ;
- α is the magnitude of the self propulsion force;
- β is the friction coefficient;

- V_j^a e V_j^r are, respectively, the attractive and repulsive potential functions, defined as:

$$V_j^a(\phi) := C_j^a e^{-\phi/\ell_j^a}, \quad V_j^r(\phi) := C_j^r e^{-\phi/\ell_j^r}, \quad (2.5)$$

where C_j^a , C_j^r denote the “magnitude of the potentials”, ℓ_j^a , ℓ_j^r their “characteristic lengths” and ϕ is a real positive variable.

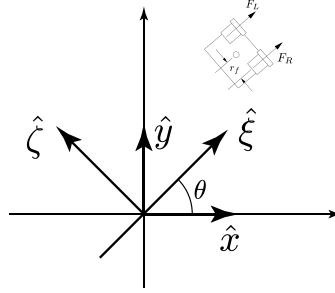


Figure 2.1: Coordinate system to describe position and velocity of the vehicles.

In the biology model proposed in [6], potentials serve to organize a group of self-propelled particles into a mill-like formation. In the VARP theory model attractive potentials have been used to direct vehicles towards way points and attractive/repulsive potentials to keep vehicles avoiding each other and stationary obstacles, namely a point to point controller will arise from this theory.

VARP Control Theory on \mathbb{R}^2

In this chapter, we will describe how the VARP control theory previously introduced could be used in order to control the Kelly robot, a second-order dynamical system whose state space lives in \mathbb{R}^2 . Firstly, we will introduce the Kelly and how VARP has been applied to it. Later, we will describe the numerical schemes used for the simulations, which are reported at the end of the chapter.

3.1 Kelly robot

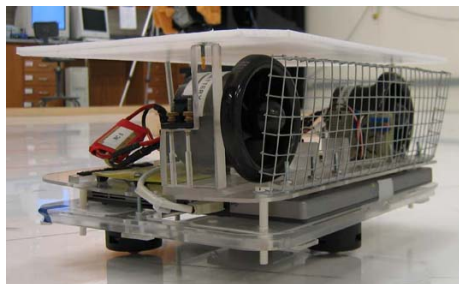


Figure 3.1: Kelly vehicle at the Caltech University. Adapted from [1].

In Figure 3.1 it is possible to see the Kelly, which is the robot used at the *Caltech University* for testing the VARP principle. It is characterized by a plexiglass structure in which are mounted: an onboard micro-controller (that implements the algorithm for the movement control), some onboard sensors and an 802.11b wireless Ethernet card with which robots could interface to each other. Other interesting characteristics are the low-friction omnidirectional casters (which allow the Kelly to move) and two high-performance ducted fans each capable of producing up to 4.5 N of continuous thrust. Caltech tests have been conducted at the MVWT (Caltech Multi-Vehicle Wireless Testbed) which is a platform designed for validating theoretical advances in multiple-vehicle coordination and cooperative control. The MVWT consists of Kelly vehicles with the ability to communicate over a wireless network, an arena for multi-vehicle operations, a lab positioning system using

overhead cameras, and an offboard computer network. The smooth floor where the vehicles maneuver has dimensions of approximately 6.5 m \times 7.0 m. The vehicles are marked with binary symbols on their hats, which the vision system uses to identify each vehicle's position and orientation.

Another interesting characteristic of the Kelly is that it is a nonlinear system whose evolution is described by the following second-order equations:

$$m \frac{du}{dt} = -\mu u + (F_R + F_L) \cos \theta, \quad (3.1)$$

$$m \frac{dv}{dt} = -\mu v + (F_R + F_L) \sin \theta, \quad (3.2)$$

$$J \frac{d\Omega}{dt} = -\psi \Omega + (F_R - F_L) r_f, \quad (3.3)$$

where:

- F_R and F_L are the magnitudes of the forces generated by, respectively, the right and left ducted fans, which are separated by a distance of $2r_f$ as shown in the Figure 3.2;
- m is the mass of the vehicle, in the specific case of the Kelly it is 5.05 kg;
- μ is the linear friction coefficient, in the case of the MVWT platform it is 4.5 kg/s;
- ψ is the angular friction coefficient, in the case of the MVWT it is 0.064 kg m²/s;
- $u = \frac{dz_x}{dt}$ and $v = \frac{dz_y}{dt}$ are the components of the linear velocity of the vehicle, where z_x and z_y are the position coordinates of the vehicle, respectively;
- $\Omega = \frac{d\theta}{dt}$ is the angular velocity with respect to the $\hat{x} - \hat{y}$ reference frame and θ is the orientation of the vehicle (with respect to the same $\hat{x} - \hat{y}$ reference frame).

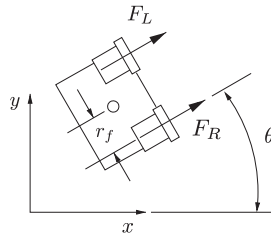


Figure 3.2: Kelly's forces and geometry scheme.

3.2 VARP Control Theory applied to the Kelly robot

Given Kelly's equations of motion (3.1)–(3.3) and the equations which describes the VARP theory (2.4), it is possible to combine them and, after some mathematical steps, to obtain a direct definition of the right and left fan forces.

Writing the second equation of (2.4) with the explicit potential functions we have:

$$f_i = m \frac{d\vec{w}_i}{dt} = m \frac{d^2 \vec{z}_i}{dt^2} = \alpha \hat{\xi}_i - \beta \vec{w}_i - \text{grad}_{\vec{z}_i} \sum_{j, j \neq i}^N \left[-C_j^a e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} + C_j^r e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right]. \quad (3.4)$$

Applying the gradient to the sum we obtain:

$$f_i = m \frac{d\vec{w}_i}{dt} = m \frac{d^2 \vec{z}_i}{dt^2} = \alpha \hat{\xi}_i - \beta \vec{w}_i + \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j}) \hat{x} + (z_{y_i} - q_{y_j}) \hat{y}}{d(\vec{z}_i, \vec{q}_j)} \right]. \quad (3.5)$$

The $\hat{\xi} - \hat{\zeta}$ system is obtained rotating the $\hat{x} - \hat{y}$ system by an angle θ , and they have the following relationship:

$$\hat{x} = \cos \theta_i \hat{\xi}_i - \sin \theta_i \hat{\zeta}_i \quad (3.6)$$

$$\hat{y} = \sin \theta_i \hat{\xi}_i + \cos \theta_i \hat{\zeta}_i \quad (3.7)$$

Replacing equations (3.6) and (3.7) in (3.5) we obtain:

$$f_i = \alpha \hat{\xi}_i - \beta \vec{w}_i + \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j}) \cos \theta_i + (z_{y_i} - q_{y_j}) \sin \theta_i}{d(\vec{z}_i, \vec{q}_j)} \hat{\xi}_i + \left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{-(z_{x_i} - q_{x_j}) \sin \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i}{d(\vec{z}_i, \vec{q}_j)} \hat{\zeta}_i \right], \quad (3.8)$$

therefore the equation (3.8) can be decomposed in its components along the $\hat{\xi}_i$ and $\hat{\zeta}_i$ axes.

From the scalar product between (3.8) and the unit vector $\hat{\xi}_i$ we obtain:

$$f_{i,x}^{\parallel} = \alpha \hat{\xi}_i - \beta \vec{w}_i + \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j}) \cos \theta_i + (z_{y_i} - q_{y_j}) \sin \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right] \hat{\xi}_i. \quad (3.9)$$

The equation (3.9), whose right-hand side is the component of the force along the $\hat{\xi}_i$ direction, can be decomposed in two components, the first along the \hat{x} direction (3.10), and the second along the \hat{y} direction (3.11):

$$f_{i,x}^{\parallel} = \alpha \cos \theta_i - \beta u_i - \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \left(\frac{(z_{x_i} - q_{x_j}) \cos^2 \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i \sin \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right) \right], \quad (3.10)$$

$$f_{i,y}^{\parallel} = \alpha \sin \theta_i - \beta v_i - \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \left(\frac{(z_{x_i} - q_{x_j}) \cos \theta_i \sin \theta_i + (z_{y_i} - q_{y_j}) \sin^2 \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right) \right]. \quad (3.11)$$

From the scalar product between (3.8) and the unit vector $\hat{\zeta}_i$ we obtain:

$$f_i^{\perp} = - \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \right. \quad (3.12)$$

$$\left. \left(\frac{-(z_{x_i} - q_{x_j}) \sin \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right) \right] \hat{\zeta}_i \quad (3.13)$$

which is the component of the force along the $\hat{\zeta}_i$ direction.

Equation (3.12) denotes the force which is responsible for the change of the angular momentum. Therefore, we can write:

$$-r_f f_i^{\perp} = -r_f \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \left(\frac{-(z_{x_i} - q_{x_j}) \sin \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right) \right]. \quad (3.14)$$

The right-hand side member of equation (3.14) is the magnitude of the mechanical torque, which could be calculated as the product between the position vector of the force r_f and the component (3.12) of the force which is perpendicular to Kelly's direction.

However, the equation (3.14) is not directly physically interpretable. In fact, from the Figure 3.2 it is possible to note that the velocity \vec{w} is directed along the $\hat{\xi}$ direction and it is null along $\hat{\zeta}$. Nevertheless, the perpendicular component $\left. \frac{d\vec{w}}{dt} \right|_{\perp}$ (projected along $\hat{\zeta}$) of the robot acceleration is related to its rotation; that is why it could be convenient to define a virtual representation of our problem, based on an interpretation of the physical model, since r_f , as in [1], is not defined as a vector, but as a distance. We can define a distance r_f , with a direction consistent to the fact that $\left. \frac{d\vec{w}}{dt} \right|_{\perp}$ is directed along $\hat{\zeta}$. In fact, if we multiply $\left. \frac{d\vec{w}}{dt} \right|_{\perp}$ with the mass, we will obtain a force along the $\hat{\zeta}$ direction, which is responsible for the Kelly's rotation. We could think to decompose this force along the $\hat{\zeta}$ direction in two components, both of them with a magnitude equal to the half of the magnitude of the force. Consequently the two virtual forces, with a moment arm r_f , establish a torque which cause the rotation of the system.

Comparing the last equations with Kelly's equations of motion, we obtain the relationship between VARP theory and Kelly's parameters. Particularly, by matching (3.1) and (3.10), we obtain:

$$-\mu u_i + (F_{R,i} + F_{L,i}) \cos \theta_i = \alpha \cos \theta_i - \beta u_i + \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \left(\frac{(z_{x_i} - q_{x_j}) \cos^2 \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i \sin \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right) \right]. \quad (3.15)$$

The parameter β could be chosen arbitrarily, hence we will impose $\beta = \mu$. Furthermore, simplifying $\cos \theta_i$ both in the left and the right side of the equation, we obtain:

$$F_{R,i} + F_{L,i} = \alpha - \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j}) \cos \theta_i + (z_{y_i} - q_{y_j}) \sin \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right]. \quad (3.16)$$

Then, comparing (3.3) with (3.14) and taking $\psi = 0$ (this condition will not affect on the results) we obtain:

$$F_{R,i} - F_{L,i} = - \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{-(z_{x_i} - q_{x_j}) \sin \theta_i + (z_{y_i} - q_{y_j}) \cos \theta_i}{d(\vec{z}_i, \vec{q}_j)} \right]. \quad (3.17)$$

From (3.16) and (3.17) we have:

$$F_{R,i} = \frac{\alpha}{2} - \frac{1}{2} \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j})(\cos \theta_i - \sin \theta_i) + (z_{y_i} - q_{y_j})(\sin \theta_i + \cos \theta_i)}{d(\vec{z}_i, \vec{q}_j)} \right], \quad (3.18)$$

$$F_{L,i} = \frac{\alpha}{2} - \frac{1}{2} \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j})(\cos \theta_i + \sin \theta_i) + (z_{y_i} - q_{y_j})(\sin \theta_i - \cos \theta_i)}{d(\vec{z}_i, \vec{q}_j)} \right]. \quad (3.19)$$

Collecting the equal terms in the equations (3.18) and (3.19) we can define:

$$F_{1,i} = -\frac{1}{2} \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{x_i} - q_{x_j})}{d(\vec{z}_i, \vec{q}_j)} \right], \quad (3.20)$$

$$F_{2,i} = -\frac{1}{2} \sum_{j, j \neq i}^N \left[\left(\frac{C_j^a}{\ell_j^a} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^a}} - \frac{C_j^r}{\ell_j^r} e^{-\frac{d(\vec{z}_i, \vec{q}_j)}{\ell_j^r}} \right) \frac{(z_{y_i} - q_{y_j})}{d(\vec{z}_i, \vec{q}_j)} \right]. \quad (3.21)$$

In the end, we obtain two equations for the right and the left forces of the fans.

$$F_{R,i} = \frac{\alpha}{2} + (\cos \theta_i - \sin \theta_i) F_{1,i} + (\sin \theta_i + \cos \theta_i) F_{2,i}, \quad (3.22)$$

$$F_{L,i} = \frac{\alpha}{2} + (\cos \theta_i + \sin \theta_i) F_{1,i} + (\sin \theta_i - \cos \theta_i) F_{2,i}. \quad (3.23)$$

These equations could be numerically implemented in order to simulate Kelly's dynamic evolution.

3.3 Numerical schemes for the simulation of a dynamic system's evolution

The aim of this section is to apply two numerical schemes, namely the *forward Euler method* and the *explicit Runge-Kutta method of the fourth order* to approximately solve the differential equations that govern the motion of the Kelly robot.

3.3.1 Forward Euler Method (fEul)

The forward Euler method (hereafter denoted as fEul) is a numerical integration method used to approximate the solutions of an ordinary differential equation starting from an initial value, namely an *initial value problem* (IVP). This method is based on two key ideas, namely the uniform time-discretization of the state variable and the approximation of its first derivative by means of the right-side incremental ratio. Uniform time-discretization is a procedure that affords switching from a continuous time function to a discrete one. In other words, since we are unable to evaluate the value of a function $x(t)$ at any time, we define a uniform succession of time-steps, separated by a reasonably short interval $h > 0$ termed *stepsize*, which are denoted by $t_0 = 0$, $t_1 = t_0 + h$, $t_2 = t_1 + h$, and so forth, at which we will try to approximate the values of the function.

Recalling a generic IVP:

$$\begin{cases} \dot{x}(t) = f(t, x(t)), t \geq t_0, \\ x(t_0) = x_0, \end{cases} \quad (3.24)$$

we can write from the definition of first-order derivative

$$\dot{x}(t) = \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h}. \quad (3.25)$$

According to the fEul, for reasonably small values of h , the first-order derivative \dot{x} can be approximated as the incremental ratio

$$\dot{x}(t) \approx \frac{x(t+h) - x(t)}{h}. \quad (3.26)$$

Since we have discretized the time variable, we can denote with x_n the value of the function $x(t_n)$ at the n -th step, and with x_{n+1} the value at the following step. With this notation we can write the equation (3.26) for a generic t_n as:

$$f(t_n, x_n) \approx \frac{x_{n+1} - x_n}{h}. \quad (3.27)$$

From the equation above, we can write the iterative equation which is crucial in the fEul method:

$$x_{n+1} = x_n + hf(t_n, x_n). \quad (3.28)$$

We will apply the fEul method to approximate the solutions to equations (3.1), (3.2) and (3.3) which will be discretized and then used for the numerical simulations. In the following

mathematical steps, we will only apply the method to the function (3.1), because for the (3.2) and (3.3) the procedure is the same.

For the sake of notation conciseness, hereafter we shall assume the presence of a single Kelly robot, so as to drop the index “i”. Starting from u_n and x_n , we will apply fEul to the Kelly’s differential equation (3.1) in order to calculate u_{n+1} and x_{n+1} . The differential equation is a second order equation in the variable x , thus, with $u = \dot{x}$, we have:

$$m\ddot{x} = -\mu\dot{x} + (F_R + F_L) \cos \theta. \quad (3.29)$$

At $t = 0$, the Kelly has a position x_0 and a velocity u_0 along x . The sampling step is denoted as h . We have two IVPs:

$$\text{IVP}_1 = \begin{cases} \dot{u} = -\frac{\mu}{m}u + \frac{(F_R+F_L)\cos\theta}{m}, \\ u(t_0) = u_0. \end{cases}, \quad \text{IVP}_2 = \begin{cases} \dot{x} = u, \\ x(t_0) = x_0. \end{cases} \quad (3.30)$$

From the second IVP, applying fEul, we have:

$$x_{n+1} = x_n + hu_n. \quad (3.31)$$

From the first IVP, applying fEul, we have:

$$u_{n+1} = u_n + h \left[-\frac{\mu}{m}u_n + \frac{(F_{R,n} + F_{L,n}) \cos \theta_n}{m} \right]. \quad (3.32)$$

For the equation (3.2) we have:

$$v_{n+1} = v_n + h \left[-\frac{\mu}{m}v_n + \frac{(F_{R,n} + F_{L,n}) \sin \theta_n}{m} \right]. \quad (3.33)$$

For the equation (3.3) we have:

$$\theta_{n+1} = \theta_n + h\Omega_n, \quad (3.34)$$

$$\Omega_{n+1} = \Omega_n + h \left[-\frac{\psi}{J}\Omega_n + \frac{(F_{R,n} - F_{L,n})r_f}{J} \right]. \quad (3.35)$$

Using fEul iteratively, we can approximate the values of the functions calculating them on each sampling step: the more the sampling step is small, the more the approximation will be accurate.

3.3.2 Explicit Fourth Order Runge-Kutta Method (eRK4)

Runge-Kutta is a class of methods used for the numerical integration of initial value problems. Differently from the fEul, the Runge-Kutta scheme is a fourth-order method, hence it aims to make the increment as accurate as possible. From the equation (3.28) we can notice that with the fEul method the value of the x variable at the successive step is calculated adding to the current one an *increment* of $hf(t_n, x_n)$. The eRK4 scheme is based on four partial increments that are:

$$k_{1,n} = hf(t_n, x_n), \quad (3.36)$$

$$k_{2,n} = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_{1,n}), \quad (3.37)$$

$$k_{3,n} = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_{2,n}), \quad (3.38)$$

$$k_{4,n} = hf(t_n + h, x_n + k_{3,n}). \quad (3.39)$$

Each increment will be “weighted” with a coefficient defined by the method and, adding all of them together, they lead to a complete step, that is:

$$x_{n+1} = x_n + \frac{1}{6} [k_{1,n} + 2k_{2,n} + 2k_{3,n} + k_{4,n}]. \quad (3.40)$$

Applying the eRK4 method to Kelly’s equations of motion (in particular, we consider (3.1)), we obtain the increments:

$$k_{1,n} = h \left[-\frac{\mu}{m} u_n + \frac{(F_{R,n} + F_{L,n}) \cos \theta_n}{m} \right], \quad (3.41)$$

$$k_{2,n} = h \left[-\frac{\mu}{m} \left(u_n + \frac{k_{1,n}}{2} \right) + \frac{(F_{R,n} + F_{L,n}) \cos \theta_n}{m} \right], \quad (3.42)$$

$$k_{3,n} = h \left[-\frac{\mu}{m} \left(u_n + \frac{k_{2,n}}{2} \right) + \frac{(F_{R,n} + F_{L,n}) \cos \theta_n}{m} \right], \quad (3.43)$$

$$k_{4,n} = h \left[-\frac{\mu}{m} (u_n + k_{3,n}) + \frac{(F_{R,n} + F_{L,n}) \cos \theta_n}{m} \right], \quad (3.44)$$

and the resolving equation (complete step) will be:

$$u_{n+1} = u_n + \frac{1}{6} [k_{1,n} + 2k_{2,n} + 2k_{3,n} + k_{4,n}]. \quad (3.45)$$

Applying the equations (3.36)–(3.40) to the remaining Kelly’s equations of motion, (3.2) and (3.3), the components of the velocity along \hat{y} and the angular velocity could be calculated.

3.4 Simulations and results of VARP Method

In this section we will introduce five different tests in order to check if the VARP theory applied to the Kelly works properly. The tests carried out in this section, in addition to verifying that the VARP theory works correctly, will focus on how VARP parameters affect Kelly's behavior. Furthermore, the potentials have been positioned in such a way as to form corridors, barriers and cages, in order to direct the Kelly in a definite path. Therefore, for each test, the parameters and the potentials with the relative positions and characteristics will be provided.

For each test three figures will be provided: the first represents the path taken by the Kelly on the $x - y$ plane. The path is represented by a blue circle, which are printed in the image every sampling instant $t = 0.12s$. The more the circles are spaced, the more the Kelly is moving fast. The black crosses represent the obstacles, the green cross indicates the starting point, the blue cross represents the target while the red cross indicates the arrival point which is the position of the Kelly in the last instant of simulation. Moreover, the first figure shows a 3D representation that helps us to understand the shape of the potential field. In this 3D graph the Kelly's path will be projected in order to understand how the robot moves in the potential field. In this representation, the target corresponds to a well in the potential field, while the obstacles correspond to bulges. The sharpness of this bulges and wells depends, respectively, on how the obstacles and targets parameters are defined. In particular, the greater the parameters L_a and L_r , the broader wells and bulges are. The greater the parameters C_a and C_r , and the larger wells and bulges are respectively in depth and in height. The second figure shows in detail the target approaching of the Kelly. Kelly's path is represented by a blue line, the arrival point by a black cross, while the target by a black diamond. In the third figure there are four panels that will be useful to understand how the Kelly behaves while it completes its path. These four panels show the trend of the fans forces, the orientation of the Kelly, and the distance from the target as the time changes.

3.4.1 First Test

In this test, we used the following potentials:

Potentials							
Potentials names	X	Y	ℓ_a	ℓ_r	C_a	C_r	Note
P01	8.5	11	1	2	0.6	3	Repulsive
P02	10	6	1	3	0.5	1.5	Repulsive
P03	1	4	1	1	0.5	0.5	Neutral
P04	4	2	1	1	0.4	1	Repulsive
P05	6	4	1	1	0.5	0.8	Repulsive
P06	5	8	1	1	0.5	0.7	Repulsive
P07	8	7	1	2	0.9	1.5	Repulsive
P08	7	10	1	1	1	0.3	Attractive
P09	3	7	1	1	0.7	2	Repulsive
P10	10	10	5	1	10	0	Attractive, Target

Table 3.1: Potentials for the first VARP test.

The remaining parameters are defined as follows:

VARP Parameters		
Parameter	Value	Unit
α	0.2	N
μ	4.5	$\frac{\text{Kg}}{\text{s}}$
ψ	0.064	$\frac{\text{Kg m}^2}{\text{s}}$
β	4.5	$\frac{\text{Kg}}{\text{s}}$
ω_1	0	$\frac{\text{rad}}{\text{s}}$
θ_1	$\pi/6$	rad

Table 3.2: Parameters for the first VARP test.

In this test we used a low value for α in order to allow Kelly to follow an almost-optimal path. Indeed, as we will see from the next tests, as the α increases the Kelly will be more and more independent from the action of the potentials and influenced to follow its own motion. As the Figure 3.3 shows, the obstacles are placed in such a way to form a corridor towards the target, and we can notice that the Kelly is able to avoid them in a proper way. The target has a moderated value of C_a , and it is possible to see that the closer Kelly is to the target, the faster it is. This can be noticed by the fact that the blue dots are more distant from each other in the area next to the target and also from the trend of the curve that represent the distance to the target, in Figure 3.5. Although the Kelly comes very close to the target, it never stops on it because there is a self-propelling coefficient α which is not equal to zero. Nevertheless the behaviour of the Kelly is satisfying and we can say that the VARP is working properly. In Figure 3.3 we can see a graphic representation of how the potential field is made. The Figure 3.4 shows how the Kelly is unable to approach the target and remain in its proximity for the whole the time. This is due to the action of

its α parameter, which, as previously mentioned, corresponds to a term of self-propulsion which causes the Kelly to move continuously, despite having reached the target. Figure 3.5 shows the forces of the Kelly's fans, its orientation and the distance from the target over time. It is possible to see that the forces exerted by the fans are greater when the Kelly is in proximity to the target: as seen in Figure 3.4, Kelly tends to wander around the target once it has been reached and this implies a great force required from the fans. The Kelly orientation panel also points out how the robot wanders around the target. From the distance panel, we can see how initially the Kelly approaches the target in a linear way, and then its trend changes becoming steeper. Also in this graph we can see that the Kelly does not stop and does not position itself at a certain distance from the target once the latter is reached but, on the contrary, it moves continuously.

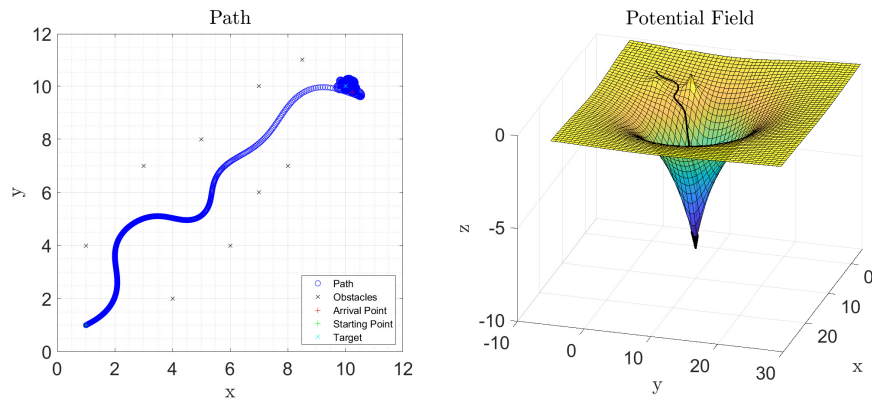


Figure 3.3: First test - Path and potential field.

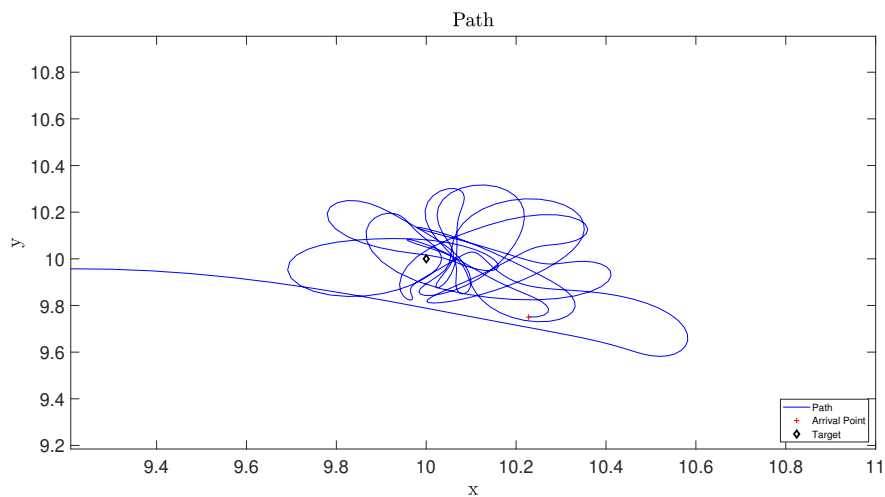


Figure 3.4: First test - Target approaching.

3.4.1. First Test

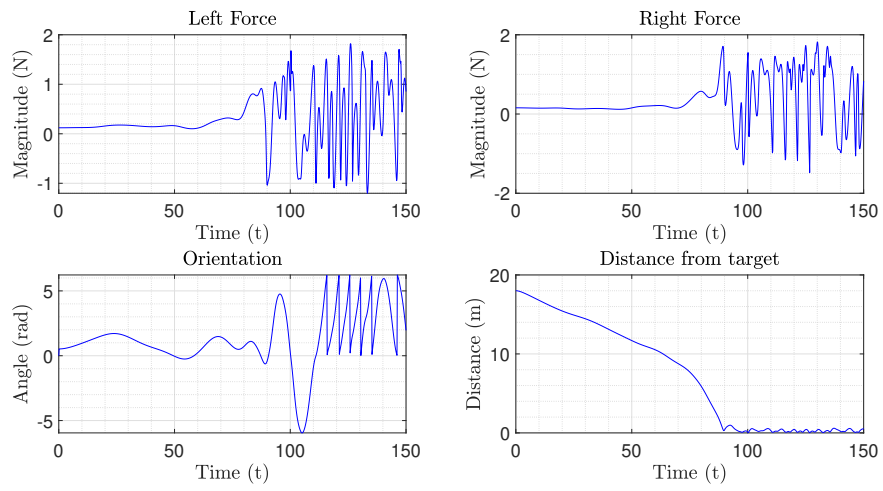


Figure 3.5: First test - Fans forces, orientation and distance over time.

3.4.2 Second Test

For this test, we used the same potentials shown in Table 3.1. The other parameters are:

VARP parameters		
Parameter	Value	Unit
α	1.5	N
μ	4.5	$\frac{\text{Kg}}{\text{s}}$
ψ	0.064	$\frac{\text{Kg m}^2}{\text{s}}$
β	4.5	$\frac{\text{Kg}}{\text{s}}$
ω_1	0	$\frac{\text{rad}}{\text{s}}$
θ_1	$\pi/6$	rad

Table 3.3: Parameters for the second VARP test.

As shown in Table 3.3, we used an higher value for α . Figure 3.6 displays how the Kelly is more subjected to an higher self-propelling coefficient. As previously said, the higher α is, the less Kelly's motion is affected by the presence of attractive and repulsive potentials. In this particular test, the Kelly doesn't follow the best path because firstly it turns due to repulsive potentials and secondly it tends to go straight due to the self-propelling term. Therefore, it does not maintain the right orientation towards the target. Moreover, we can see that Kelly passes very close to the potential "P08" because in that point there is an high attraction to the target as well as a strong self-propelling term. This could be a problem if the repulsive potential "P08" represents a physical obstacle, but it could be easily avoided changing the value of his C_r . Figure 3.7 shows also that an higher value of α means more oscillation around the target once the Kelly is close to it. We can note from the Figure 3.8 how the Kelly manages to reach the target in a time equal to 25% with respect to the previous one.

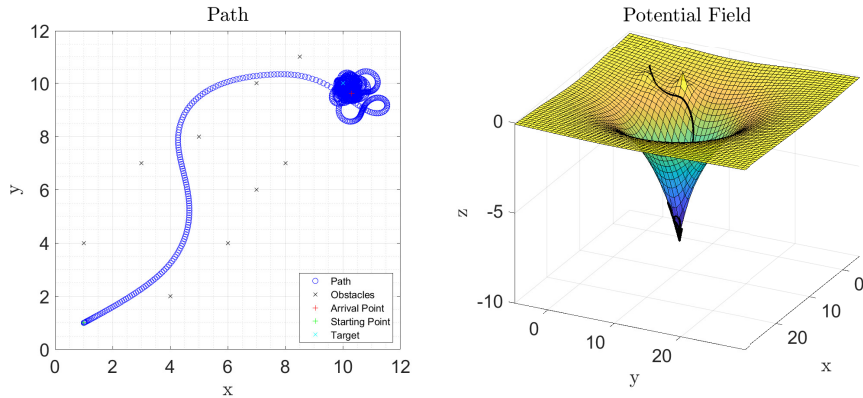


Figure 3.6: Second test - Path and potential field.

3.4.2. Second Test

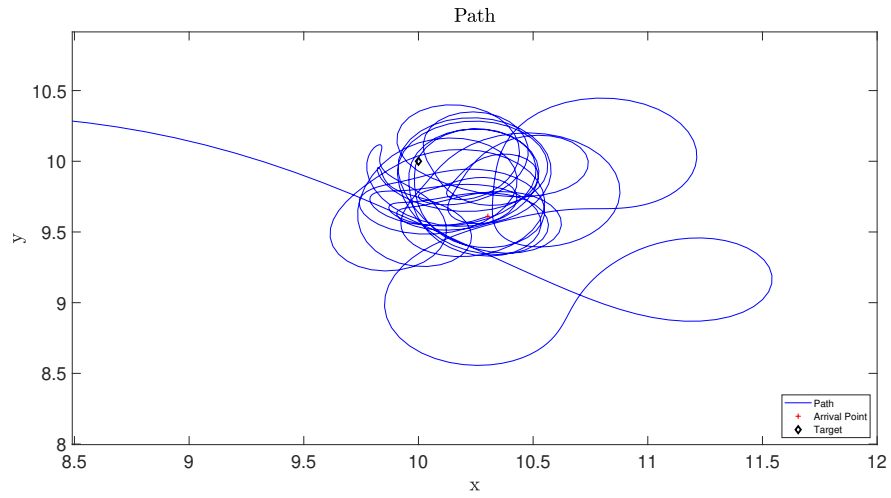


Figure 3.7: Second test - Target approaching.

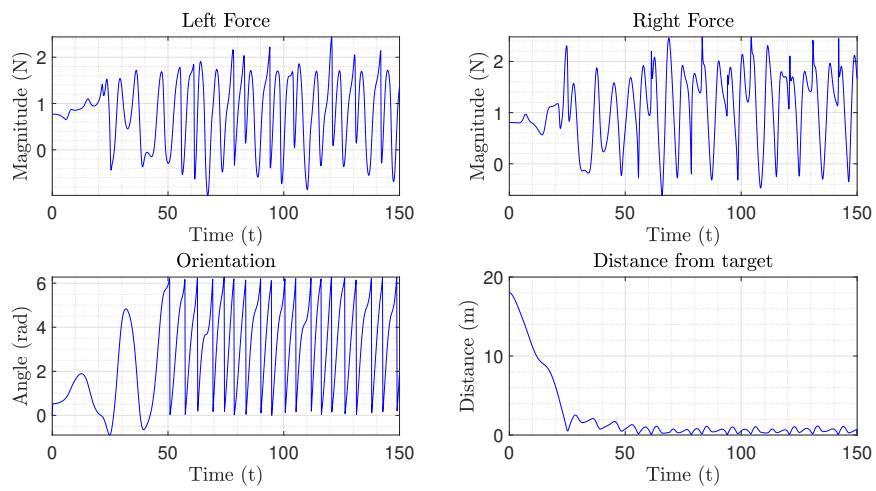


Figure 3.8: Second test - Fans forces, orientation and distance over time.

3.4.3 Third Test

In this test, we used the following potentials:

Potentials							
Potentials names	X	Y	ℓ_a	ℓ_r	C_a	C_r	Note
P01	6	6	0.1	0.5	0.1	10	Repulsive
P02	6	10	0.1	0.5	0.1	10	Repulsive
P03	5	8	0.1	0.5	0.1	10	Repulsive
P04	7	8	0.1	0.5	0.1	10	Repulsive
P05	7	4	0.1	0.5	0.1	10	Repulsive
P06	8	6	0.1	0.5	0.1	10	Repulsive
P07	5	5	0.1	6	0.1	10	Repulsive
P08	9	4	0.1	0.5	0.1	10	Repulsive
P09	5	12	0.1	0.5	0.1	10	Repulsive
P10	12	12	10	1	30	0	Attractive, Target

Table 3.4: Potentials for the third VARP test.

The remaining parameters are defined as follows:

VARP parameters		
Parameter	Value	Unit
α	0.2	N
μ	4.5	$\frac{\text{Kg}}{\text{s}}$
ψ	0.064	$\frac{\text{Kg m}^2}{\text{s}}$
β	4.5	$\frac{\text{Kg}}{\text{s}}$
ω_1	0	$\frac{\text{rad}}{\text{s}}$
θ_1	$\pi/6$	rad

Table 3.5: Parameters for the third VARP test.

In this test we created a barrier between the Kelly's starting position and the target so as to see if the robot manages to avoid it. Figure 3.9 shows that the Kelly is able to avoid the barrier, which is composed by several repulsive potentials with a low value of ℓ_r . We chose a low value because otherwise the Kelly would be affected too much by the barrier. Indeed, by choosing a low value for ℓ_r , the repulsive potentials have influence only in their close proximity. The only repulsive potential that has an higher value of ℓ_r is "P07", which is located in the front tip of the pseudo-triangular barrier. We chose an higher value because it allows the Kelly to stay far enough from the barrier. Thus, by appropriately adjusting the position and magnitude of the positive and repulsive potentials, we can not only avoid or approach a single point, but also a barrier or a wall. This is very meaningful because this feature permits to solve several real problems. For example, this particular version of VARP, developed in \mathbb{R}^2 , can be easily applied to a robot vacuum cleaner, which needs to avoid artificial barriers such as walls, steps, tables, furniture and to reach target points as charging stations, dirty areas and parking spots. Figure 3.9 also shows us that the

field of potentials is more complex than the previous ones. In fact, the potential barrier is represented by a mountain range in the 3D graph. It is interesting to note from Figure 3.11 that Kelly initially remains at a considerable distance from the target, this is because it is held back by the potential barrier, which partially counteracts the attraction of the target. Despite this, given the strong attraction, the Kelly manages to approach the target slowly and then, once the barrier is crossed, it continues towards it with greater speed. This also suggests that this VARP feature could also be used in the opposite way of what was mentioned above. We may request that the Kelly never reach the target, as its achievement could seriously jeopardize its functioning. Indeed, taking the robot vacuum cleaner as an example, we could prevent it from reaching a dirty area if this action involves crossing steps or a jump that could damage it.

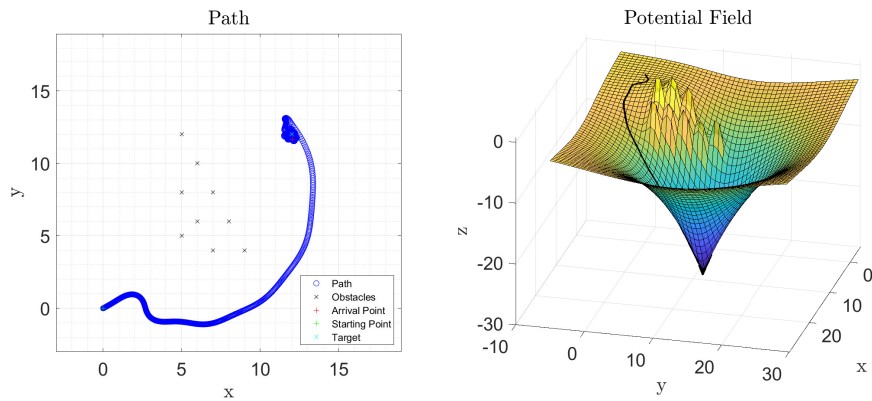


Figure 3.9: Third test - Path and potential field.

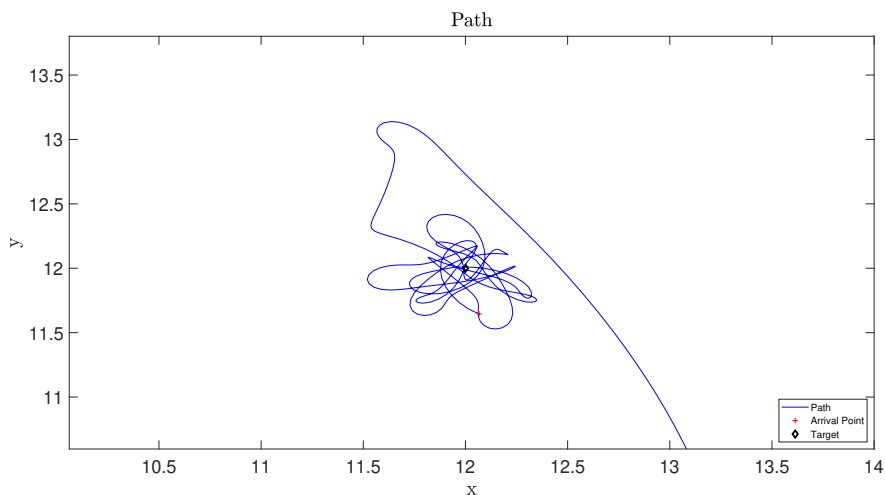


Figure 3.10: Third Test - Target approaching.

3.4.3. Third Test

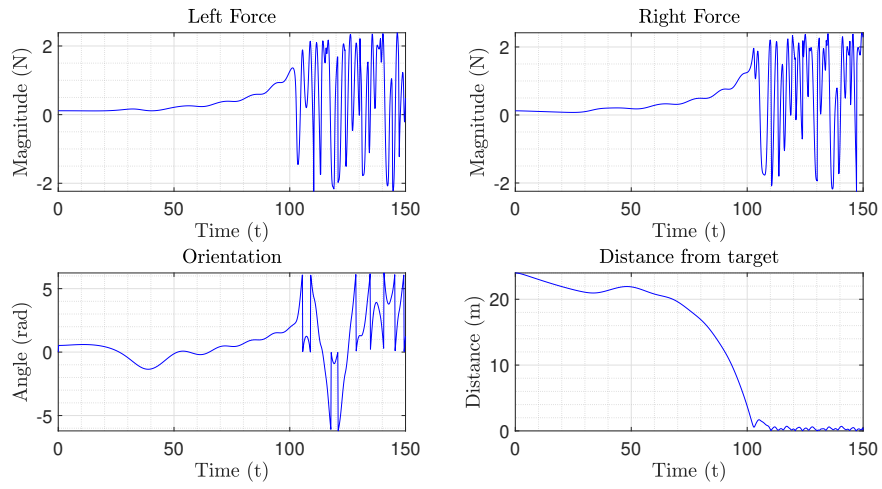


Figure 3.11: Third test - Fans forces, orientation and distance over time.

3.4.4 Fourth Test

In this test, we used the same potentials shown in Table 3.4, in order to build an identical barrier.

The remaining parameters are defined as follows:

VARP parameters		
Parameter	Value	Unit
α	2.2	N
μ	4.5	$\frac{\text{Kg}}{\text{s}}$
ψ	0.064	$\frac{\text{Kg m}^2}{\text{s}}$
β	4.5	$\frac{\text{Kg}}{\text{s}}$
ω_1	0	$\frac{\text{rad}}{\text{s}}$
θ_1	$\pi/6$	rad

Table 3.6: Parameters for the fourth VARP test.

As shown in Table 3.6, the value α has been increased compared with the previous test. This test shows very clearly how the α parameter is fundamental in determining Kelly's behavior. In fact, by increasing the α value compared to the third test, we can see how the Kelly gets closer to the barrier, after which it is repelled. This continuous effect of repulsion and thrust involves a non-linear trend of the Kelly, which feels subject to stronger forces than in the previous test. Despite the less linear trend, the Kelly manages to reach the goal in half the time compared to the previous test. In conclusion, we can say that increasing the α parameter allows the Kelly to reach the goal in a shorter time, but also implies a possible collision with obstacles and also a possible oscillation and a consequent non-linear trend. This oscillation can also be seen from the graph which shows the distance from the target as time changes, in Figure 3.14.

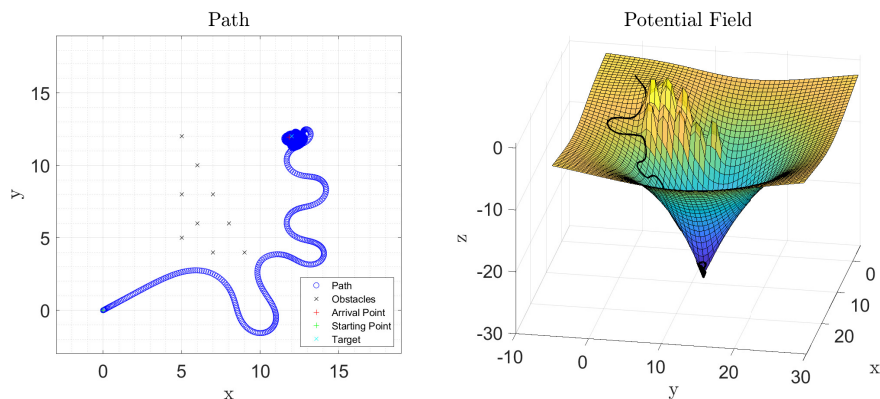


Figure 3.12: Fourth test - Path and potential field.

3.4.4. Fourth Test

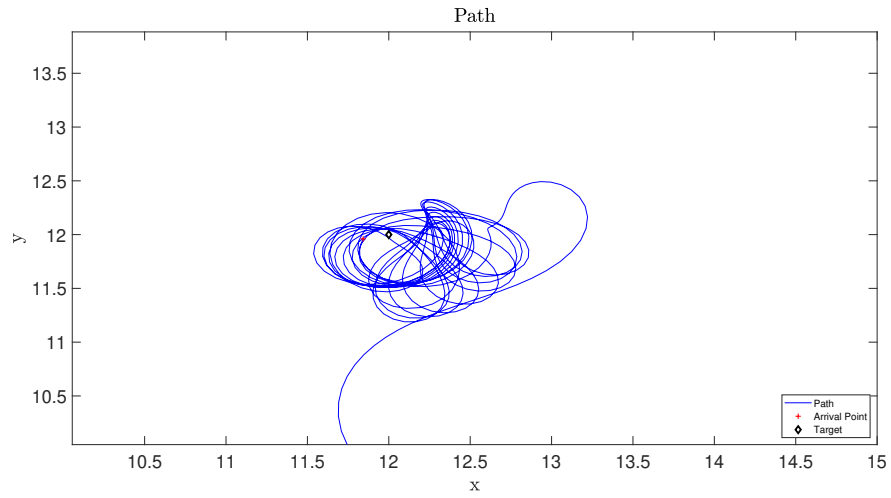


Figure 3.13: Fourth test - Target approaching.

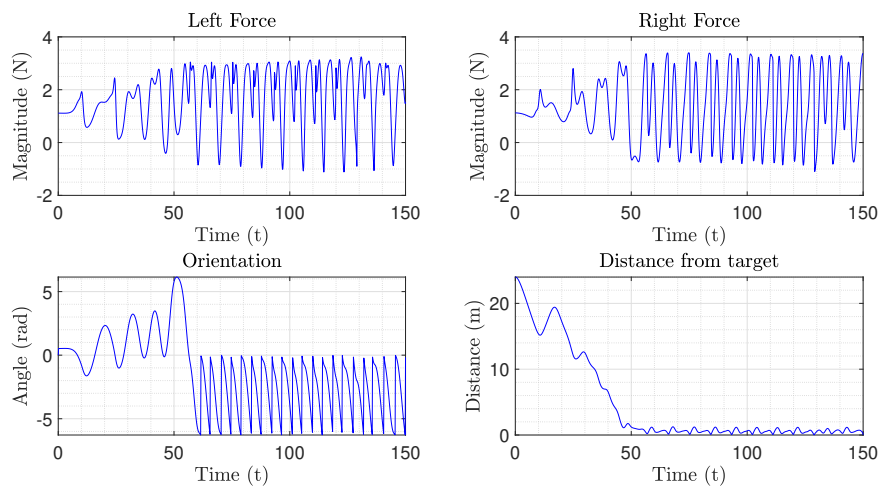


Figure 3.14: Fourth test - Fans forces, orientation and distance over time.

3.4.5 Fifth Test

In this test, we used the following potentials:

Potentials							
Potentials names	X	Y	ℓ_a	ℓ_r	C_a	C_r	Note
P01	4	8	0.1	0.9	0.1	17	Repulsive
P02	4	0	0.1	0.9	0.1	17	Repulsive
P03	8	4	0.1	0.9	0.1	17	Repulsive
P04	0	4	0.1	0.9	0.1	17	Repulsive
P05	6	6	0.1	0.9	0.1	17	Repulsive
P06	2	6	0.1	0.9	0.1	17	Repulsive
P07	2	2	0.1	0.9	0.1	17	Repulsive
P08	7	5	0.1	0.9	0.1	17	Repulsive
P09	5	7	0.1	0.9	0.1	17	Repulsive
P10	3	7	0.1	0.9	0.1	17	Repulsive
P11	1	5	0.1	0.9	0.1	17	Repulsive
P12	1	3	0.1	0.9	0.1	17	Repulsive
P13	3	1	0.1	0.9	0.1	17	Repulsive
P14	12	12	70	1	30	0	Attractive, Target

Table 3.7: Potentials for the fifth VARP test.

The remaining parameters are defined as follows:

VARP parameters		
Parameter	Value	Unit
α	0.2	N
μ	4.5	$\frac{\text{Kg}}{\text{s}}$
ψ	0.064	$\frac{\text{Kg m}^2}{\text{s}}$
β	4.5	$\frac{\text{Kg}}{\text{s}}$
ω_1	0	$\frac{\text{rad}}{\text{s}}$
θ_1	$\pi/6$	rad

Table 3.8: Parameters for the fifth VARP test.

In this test we designed a diamond-shaped cage in order to study if Kelly is able to escape through an opening. The Kelly's starting position is located in the cage center and the gap is located on one side of the rhombus. Figure 3.15 shows that Kelly is able to escape from the cage. Also this feature is very important in real situations because there are several control problems concerning the escape through a gap, such as exiting a parking lot or leaving a room through a door. This is a further evidence that VARP can be used for a wide variety of purposes. Figure 3.15 is interesting because it shows how the potential cage could be represented using a 3D graph. Also in this case, the panel relating to the distance from the target in Figure 3.17 shows how the Kelly does not initially approach the target as the cage of repulsive potentials tends to make the robot remain in position.

3.4.5. Fifth Test

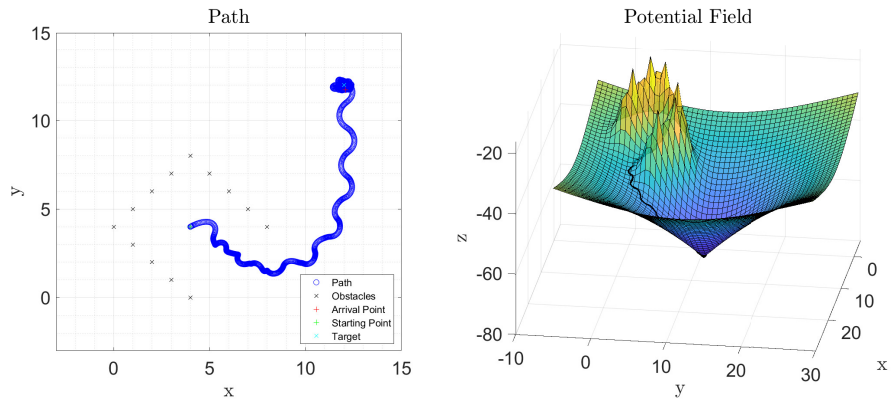


Figure 3.15: Fifth test - Path and potential field.

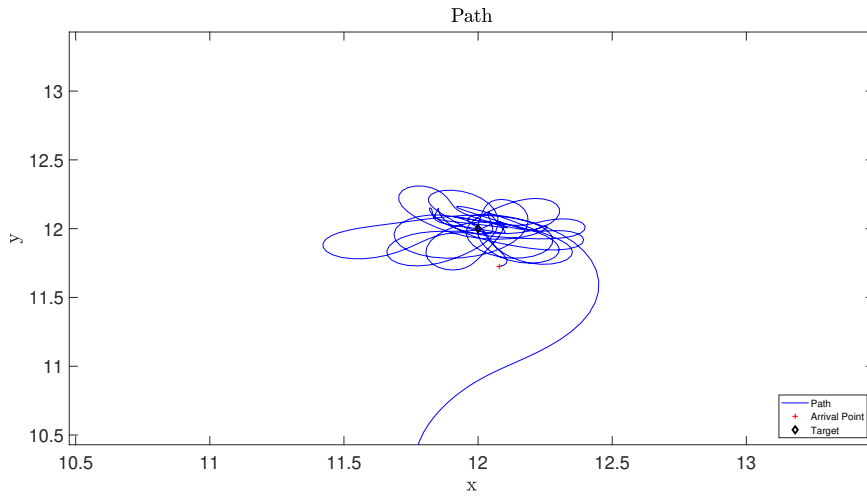


Figure 3.16: Fifth test - Target approaching.

3.4.5. Fifth Test

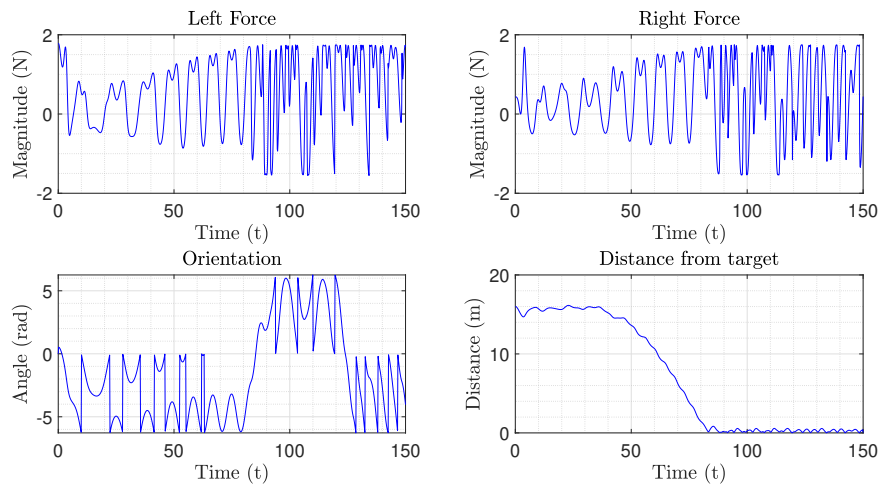


Figure 3.17: Fifth test - Fans forces, orientation and distance over time.

VARP Control Theory on the Riemannian Manifold \mathbb{S}^2

The aim of this chapter is to design the VARP principle to control second-order dynamic systems whose state space is a Riemannian manifold. This method will be called *M-VARP* as it represents an extension of the original VARP control method to a generic Riemannian manifold. Successively, the VARP principle will be extended, implemented and simulated for the control of a system whose state space is the Riemannian manifold \mathbb{S}^2 (we will denote this method as *S2-VARP*). As in the previous case, we will consider the presence of a single agent, therefore the control could not be considered “cooperative”.

4.1 Generalities on Riemannian Manifolds

Let \mathbb{M} denote a Riemannian manifold. At a point $x \in \mathbb{M}$, the tangent space to the manifold \mathbb{M} is denoted as $T_x\mathbb{M}$. The symbol $T\mathbb{M}$ denotes the tangent bundle defined as $T\mathbb{M} := \{(x, v) \mid x \in \mathbb{M}, v \in T_x\mathbb{M}\}$.

A Riemannian manifold \mathbb{M} is endowed with a bilinear, positive-definite form $\langle \cdot, \cdot \rangle_x : T_x\mathbb{M} \times T_x\mathbb{M} \rightarrow \mathbb{R}$. A local metric $\langle \cdot, \cdot \rangle_x$ also defines a local norm $\|v\|_x := \sqrt{\langle v, v \rangle_x}$, for $v \in T_x\mathbb{M}$. The Riemannian gradient of a function $\psi : \mathbb{R} \rightarrow \mathbb{M}$ evaluated at the point $x \in \mathbb{M}$ is denoted as $\text{grad}_x \psi$. The Riemannian gradient is associated to a specific metric.

A *manifold exponential map* $\exp : T\mathbb{M} \rightarrow \mathbb{M}$ applies as $y = \exp_x(v)$; it maps a pair $(x, v) \in T_x\mathbb{M}$ to a point y on the manifold. The exponential map “shifts” a point x along the geodesic curve in the direction of v to get to the point y .

Its inverse ‘log’ is defined only locally and is termed *manifold logarithm*. Given points $x, y \in \mathbb{M}$, a manifold logarithm computes a tangent vector $v = \log_x(y) \in T_x\mathbb{M}$ such that $\exp_x(v) = y$.

Given two points $x, y \in \mathbb{M}$ connectable by a geodesic arc, their *Riemannian distance* is denoted by $d(x, y)$. On a Riemannian manifold, the distance between two nearby points

may be evaluated by:

$$d(x, y) = \|\log_x(y)\|_x. \quad (4.1)$$

A fundamental result of the calculus on manifolds states that the Riemannian gradient of a squared distance function reads:

$$\text{grad}_x d^2(x, y) = -2 \log_x(y), \quad (4.2)$$

wherever the logarithm is defined.

The covariant derivative of a vector field $w_x \in T_x\mathbb{M}$ in the direction of a vector $v \in T_x\mathbb{M}$ is denoted as $\nabla_v w$. We assume \mathbb{M} to be endowed with a *metric connection* (namely, that the covariant derivative of the metric tensor is identically zero).

The *parallel transport* operator $P : T\mathbb{M}^2 \rightarrow T\mathbb{M}$ maps a tangent vector $v \in T_x\mathbb{M}$ at a given point $x \in \mathbb{M}$ into a tangent vector $w \in T_y\mathbb{M}$ at another given point $y \in \mathbb{M}$ and it is denoted as $w = P^{x \rightarrow y}(v)$. The parallel transport moves the tangent vector v from $T_x\mathbb{M}$ to $T_y\mathbb{M}$ along the geodesic curve that connects the point x to the point y keeping the tangency. Parallel transport and covariant derivation are closely related to one another, in particular, covariant derivation may be expressed in terms of parallel transport as follows:

$$\nabla_v w = \lim_{h \rightarrow 0} \frac{P^{\gamma(h) \rightarrow x}[w_{\gamma(h)}] - w_x}{h}, \quad (4.3)$$

where γ denotes any smooth curve such that $\gamma(0) = x \in \mathbb{M}$ and $\dot{\gamma}(0) = v \in T_x\mathbb{M}$. Such relationship leads to a numerical approximation of the covariant derivative at a point x .

4.2 Extension of the VARP Principle to Riemannian Manifolds

The first element to consider is a generic second-order dynamic system whose state equations are formulated on the tangent bundle of a manifold \mathbb{M} :

$$\begin{cases} \dot{x}(t) = v(t), \\ \nabla_{v(t)} v(t) = -\mu v(t) + u(t), \end{cases} \quad (4.4)$$

where the following notation has been used:

- $x(t) \in \mathbb{M}$ denotes the state of the dynamic system (it could be thought of as the position of a point mass on the manifold surface at the time t);
- $v(t) \in T_{x(t)}\mathbb{M}$ denotes the tangent vector to the trajectory $x(t)$ at the time t (it could be thought of as the velocity of the point mass on the manifold surface);
- $\nabla_v v$ represents the covariant derivative of the vector field v with respect to itself (it could be thought of as the acceleration of the point mass on the manifold surface); if $\nabla_v v = 0$ the acceleration is zero, thus the point mass will follow a uniform geodesic trajectory;

- $\mu > 0$ is the friction coefficient (therefore, the term $-\mu v$ could be thought of as a friction force which brakes the motion of the point mass);
- $u(t)$ is the control signal at the time t (it could be thought of as a force whose purpose is to move the point mass on the manifold); the control signal is a vector field which is tangent to the manifold at the state $x(t)$.

Now, extending the VARP principle to manifolds, we have:

$$\begin{cases} \dot{x}(t) = v(t), \\ \nabla_{v(t)} v(t) = -\mu v(t) + \alpha \hat{v}(t) - \text{grad}_{x(t)} V, \\ V(x) = \sum_{j=1}^N (V_j^r(d^2(x, q_j)) - V_j^a(d^2(x, q_j))), \end{cases} \quad (4.5)$$

where:

- $\alpha \hat{v}$, with $\alpha > 0$, represents the *self-propelling term* from [1], where $\hat{v} := v/\|v\|$ if $v \neq 0$ otherwise $\hat{v} = 0$;
- $V(x)$ is the total potential which depends on the state x ; V_j^r and V_j^a denotes respectively the repulsive and attractive component of the potential, as they were already defined on the VARP principle for Euclidean spaces in the Section 2.2; note that the components are indexed with j : this indexing does not appear explicitly in [1]. In the present case, each obstacle and target has his own coefficient;
- q_j terms, which are N , represent obstacles and targets positions on the manifold \mathbb{M} ;
- the function $d(\cdot, \cdot)$ denotes the Riemannian distance (or geodesic distance) between two points on the manifold; on the equations it will appear as d^2 because this, generally, simplifies the calculation of the derivative;

Combining together the equations (4.4) and (4.5), the control field has been designed as

$$u := \alpha \hat{v}(t) - \text{grad}_x \sum_{j=1}^N (V_j^r(d^2(x, q_j)) - V_j^a(d^2(x, q_j))) = \quad (4.6)$$

$$= \alpha \hat{v}(t) - \sum_{j=1}^N (\dot{V}_j^r(d^2(x, q_j)) - \dot{V}_j^a(d^2(x, q_j))) \text{grad}_x d^2(x, q_j). \quad (4.7)$$

Recalling the notable relation (4.2) for Riemannian manifolds, we have:

$$u = \alpha \hat{v}(t) - 2 \sum_{j=1}^N (\dot{V}_j^a(d^2(x, q_j)) - \dot{V}_j^r(d^2(x, q_j))) \log_x(q_j). \quad (4.8)$$

Coherently with what was expected, it results that $u \in T_x \mathbb{M}$.

As for the VARP principle for smooth spaces (see (2.5)), exponential attractive and repulsive potential functions have been chosen:

$$V_j^a(\phi) := C_j^a e^{-\sqrt{\phi}/\ell_j^a}, \quad V_j^r(\phi) := C_j^r e^{-\sqrt{\phi}/\ell_j^r}, \quad (4.9)$$

where C_j^a and C_j^r denote the “magnitude of the potentials”, ℓ_j^a and ℓ_j^r their “characteristic lengths” and ϕ is a real positive variable (as in the previous case). Their derivative with respect to ϕ could be written, respectively, as:

$$\frac{dV_j^a(\phi)}{d\phi} = -\frac{C_j^a}{2\ell_j^a\sqrt{\phi}}e^{-\sqrt{\phi}/\ell_j^a}, \quad \frac{dV_j^r(\phi)}{d\phi} = -\frac{C_j^r}{2\ell_j^r\sqrt{\phi}}e^{-\sqrt{\phi}/\ell_j^r}, \quad (4.10)$$

therefore the control signal (4.8) is:

$$u(t) = \alpha \hat{v}(t) - \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-d(x(t), q_j)/\ell_j^r}}{d(x(t), q_j)} - \frac{C_j^a}{\ell_j^a} \frac{e^{-d(x(t), q_j)/\ell_j^a}}{d(x(t), q_j)} \right) \log_{x(t)}(q_j). \quad (4.11)$$

The system to simulate numerically will be:

$$\begin{cases} \dot{x}(t) = v(t), \\ \nabla_{v(t)} v(t) = -\mu v(t) + \alpha \hat{v}(t) - \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-d(x(t), q_j)/\ell_j^r}}{d(x(t), q_j)} - \frac{C_j^a}{\ell_j^a} \frac{e^{-d(x(t), q_j)/\ell_j^a}}{d(x(t), q_j)} \right) \log_{x(t)}(q_j). \end{cases} \quad (4.12)$$

4.3 Euler scheme for the simulation of the M-VARP Control Method

It pays to explain how it was possible to use the Euler method on the manifold to simulate the evolution of a controlled system.

Recalling the system (4.12) that has to be controlled, and writing it in a compact form as:

$$\begin{cases} \dot{x}(t) = v(t), \\ \nabla_{v(t)} v(t) = \sigma(x(t), v(t), u(t)), \\ x(0) = x^0, \quad v(0) = v^0 \text{ (Initial Conditions)}, \end{cases} \quad (4.13)$$

where $\sigma : T\mathbb{M}^2 \rightarrow T\mathbb{M}$ is the function on the right-hand side of the second equation in (4.12) that maps a point x and two tangent vectors $v, u \in T_x\mathbb{M}$ into a tangent vector in $T_x\mathbb{M}$.

To solve numerically the system of differential equations (4.13), namely to simulate the controlled system (4.12), we introduce three discrete sequences that arise from the time discretization with a step size $h > 0$, that are:

- The sequence $x^k \in \mathbb{M}$, with $k = 0, 1, 2, 3, \dots$, represent the discretized version of the state $x(t)$, namely x^k denotes a numerical approximation of $x(kh)$.
- The sequence $v^k \in \mathbb{M}$, with $k = 0, 1, 2, 3, \dots$, represent the discretized version of the velocity $v(t)$, namely v^k denotes a numerical approximation of $v(kh)$.
- The sequence $u^k \in \mathbb{M}$, with $k = 0, 1, 2, 3, \dots$, represent the discretized version of the control signal $u(t)$, namely u^k denotes a numerical approximation of $u(kh)$, that is:

$$u^k := \alpha \hat{v}^k - \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-d(x^k, q_j)/\ell_j^r}}{d(x^k, q_j)} - \frac{C_j^a}{\ell_j^a} \frac{e^{-d(x^k, q_j)/\ell_j^a}}{d(x^k, q_j)} \right) \log_{x^k}(q_j). \quad (4.14)$$

Note that the time index has been placed in the high position (k) to avoid confusing it with the obstacle/target index which is in the lower position (j). Once a time discretization of the variables has been made, we can proceed applying the forward Euler method (fEul). To express the fEul method, the exponential map and the parallel transport operators recalled in Section 4.1 will be applied.

It is important to remind that the parallel transport operator is fundamental in the calculation of the covariant derivative. Let us denote with u_x a vector field on TM (in each point x of the manifold, u_x defines a tangent vector at the x point itself) of which we want to calculate the covariant derivative and v_x another vector field along which we want to calculate the covariant derivative. Then it turns out that:

$$\nabla_{v_x} u_x = \lim_{h \rightarrow 0} \frac{u_{\exp_x(hv_x)} - \mathbf{P}^{x \rightarrow \exp_x(hv_x)}(u_x)}{h}. \quad (4.15)$$

Indeed, the covariant derivative $\nabla_{v_x} u_x$ represents the velocity with which the vector field u changes, moving from x in the direction of v . Let us consider the numerical solution of the first equation in (4.13) that is:

$$\frac{dx(t)}{dt} = v(t), \quad (4.16)$$

which is an IVP on the tangent bundle TM . The algorithm that implements the numeric solution with the fEul method could be written as:

$$x^{k+1} = \exp_{x^k}(h v^k), \quad k = 0, 1, 2, 3, 4, \dots, \quad (4.17)$$

with x^0 and v^0 known from the initial conditions.

Let's consider now the numeric solution of the second equation of (4.13), that is:

$$\nabla_{v(t)} v(t) = \sigma(x(t), v(t), u(t)), \quad (4.18)$$

which is a different kind of IVP on the tangent bundle TM . The relation (4.15), applied to the vector field v with respect to itself, could be numerically approximate as

$$\nabla_v v \approx \frac{v^{k+1} - \mathbf{P}^{x^k \rightarrow x^{k+1}}(v^k)}{h}. \quad (4.19)$$

Therefore, the algorithm that implements the numeric solution by means of the fEul method is:

$$v^{k+1} = \mathbf{P}^{x^k \rightarrow x^{k+1}}(v^k + h\sigma(x^k, v^k, u^k)), \quad k = 0, 1, 2, 3, 4, \dots, \quad (4.20)$$

with x^0 and v^0 known from the initial conditions. Therefore, the numerical method that will be used is:

$$\begin{cases} x^{k+1} = \exp_{x^k}(h v^k), \\ u^k = \alpha \hat{v}^k - \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} e^{-d(x^k, q_j)/\ell_j^r} - \frac{C_j^a}{\ell_j^a} e^{-d(x^k, q_j)/\ell_j^a} \right) \log_{x^k}(q_j), \\ \sigma^k = -\mu v^k + u^k, \\ v^{k+1} = \mathbf{P}^{x^k \rightarrow x^{k+1}}(v^k + h \sigma^k), \\ k = 0, 1, 2, 3, 4, \dots, \end{cases} \quad (4.21)$$

with x^0 and v^0 known from the initial conditions. Equivalently, the numerical algorithm may be written as:

$$\begin{cases} x^{k+1} = \exp_{x^k}(h v^k), \\ u^k = \alpha \hat{v}^k - \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} e^{-d(x^k, q_j)/\ell_j^r} - \frac{C_j^a}{\ell_j^a} e^{-d(x^k, q_j)/\ell_j^a} \right) \log_{x^k}(q_j), \\ v^{k+1} = \mathbb{P}^{x^k \rightarrow x^{k+1}}((1 - h\mu)v^k + h u^k), \\ k = 0, 1, 2, 3, 4, \dots, \end{cases} \quad (4.22)$$

with the hypothesis that $v^k \neq 0$. Let us recall that, in the case of hyper-sphere ($\mathbb{M} := \mathbb{S}^{n-1}$), the canonical operators and the Riemannian distance required for the numerical implementation have the following expressions:

$$\begin{cases} \exp_x(v) = \begin{cases} x \cos(\|v\|) + v\|v\|^{-1} \sin(\|v\|), & \text{if } v \neq 0, \\ x, & \text{otherwise,} \end{cases} \\ d(x, y) = |\arccos(x^\top y)|, \\ \log_x y = \frac{d(x, y)}{\sin d(x, y)} (I_n - x x^\top) y, \\ \mathbb{P}^{x \rightarrow y}(v) = \left[I_n - \frac{(y+x)y^\top}{1+x^\top y} \right] v. \end{cases} \quad (4.23)$$

In this specific case the space state is the hyper-sphere \mathbb{S}^2 , therefore in the expressions (4.23), we will consider $n = 3$.

4.4 Simulations and results of S2-VARP Method

In this section we will introduce five different tests in order to check how the S2-VARP control method works on the unitary sphere \mathbb{S}^2 . For each test, we will provide the S2-VARP parameters that have been used and a 3D representation of the \mathbb{S}^2 manifold. In this representation, the red line represents the path of the dynamical system, the black star indicates the starting point, the green crosses represent the obstacles and the black diamond indicates the target.

4.4.1 First Test

In this test we built a corridor made up of obstacles, in order to guide the system along a predefined path. This type of problem, known as path planning, is a typical control issue in many real situations. As you we can see from the Figure 4.1, the system is able to reach the target without any problem. The parameters used in this test are shown in Table 4.1.

S2-VARP parameters			
Parameter	Value	Parameter	Value
α	0.2	$L_{r,o}$	0.1
μ	5.05	$C_{a,t}$	5
$C_{a,o}$	0	$C_{r,t}$	0
$C_{r,o}$	0.1	$L_{a,t}$	0.5
$L_{a,o}$	0.1	$L_{r,t}$	0.1

Table 4.1: Parameters for the first S2-VARP test.

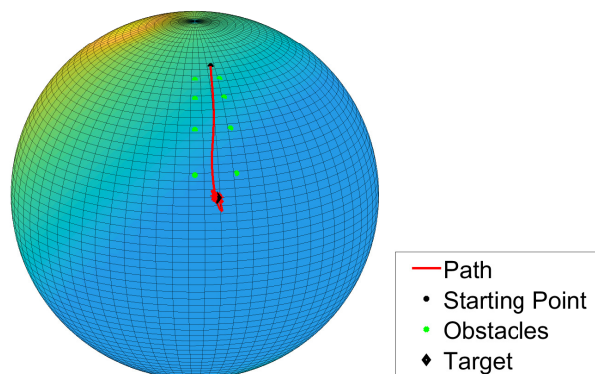


Figure 4.1: First test - Path planning using a corridor, $\alpha = 0.2$.

4.4.2 Second Test

In this test we have increased the α parameter, so as to show how even in the case of M-VARP the increase of this parameter implies a greater oscillation around the target or a possible collision with one of the obstacles, as shown in Figure 4.2. The parameters used in test are as follows:

S2-VARP parameters			
Parameter	Value	Parameter	Value
α	4	$L_{r,o}$	0.1
μ	5.05	$C_{a,t}$	5
$C_{a,o}$	0	$C_{r,t}$	0
$C_{r,o}$	0.1	$L_{a,t}$	0.5
$L_{a,o}$	0.1	$L_{r,t}$	0.1

Table 4.2: Parameters for the second S2-VARP test.

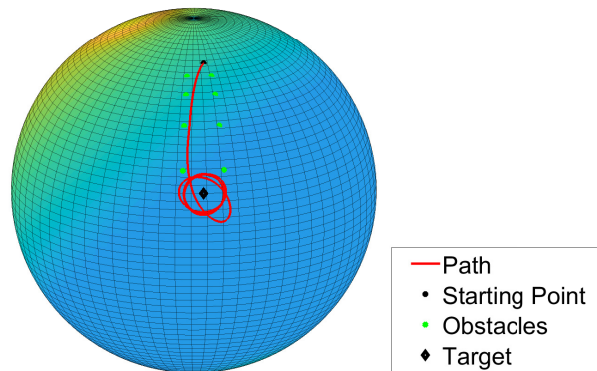


Figure 4.2: Second test - Path planning using a corridor, $\alpha = 4$.

4.4.3 Third Test

In this test we built a barrier, composed of obstacles, along the geodesic between the starting point and the target. It can be seen from the Figure 4.3 how the system is able to bypass the barrier along a wider path. The parameters are reported in Table 4.3.

S2-VARP parameters			
Parameter	Value	Parameter	Value
α	0.2	$L_{r,o}$	0.1
μ	5.05	$C_{a,t}$	5
$C_{a,o}$	0	$C_{r,t}$	0
$C_{r,o}$	0.5	$L_{a,t}$	0.5
$L_{a,o}$	0.1	$L_{r,t}$	0.1

Table 4.3: Parameters for the third S2-VARP test.

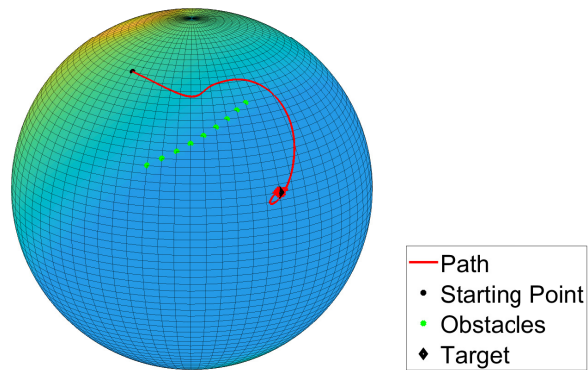


Figure 4.3: Third test - Barrier bypassing.

4.4.4 Fourth Test

Similar to what we did in the previous test, we placed a single obstacle exactly along the geodesic between the starting point and the target. Also in this case the system bypasses the obstacle and manages to reach the target, as in the Figure 4.4. Table 4.4 shows the parameters of this test.

S2-VARP parameters			
Parameter	Value	Parameter	Value
α	0.2	$L_{r,o}$	0.07
μ	5.05	$C_{a,t}$	5
$C_{a,o}$	0	$C_{r,t}$	0
$C_{r,o}$	2	$L_{a,t}$	0.5
$L_{a,o}$	0.5	$L_{r,t}$	0.1

Table 4.4: Parameters for the fourth S2-VARP test.

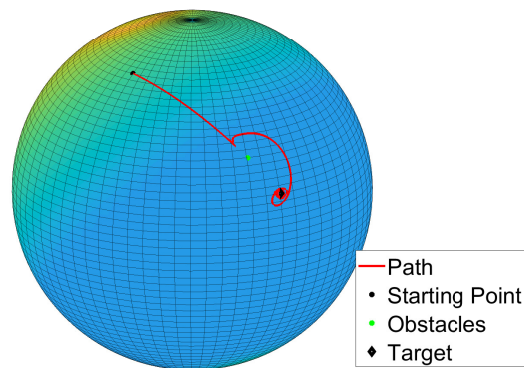


Figure 4.4: Fourth test - Obstacle avoidance.

4.4.5 Fifth Test

In this test we built a barrier that partially contains the target. Figure 4.5 shows that the system is able to reach the target through the side without any obstacles. The parameters used in this test are shown in Table 4.5.

S2-VARP parameters			
Parameter	Value	Parameter	Value
α	0.2	$L_{r,o}$	0.3
μ	5.05	$C_{a,t}$	30
$C_{a,o}$	0	$C_{r,t}$	0
$C_{r,o}$	0.4	$L_{a,t}$	10
$L_{a,o}$	0.5	$L_{r,t}$	0.1

Table 4.5: Parameters for the fifth S2-VARP test.

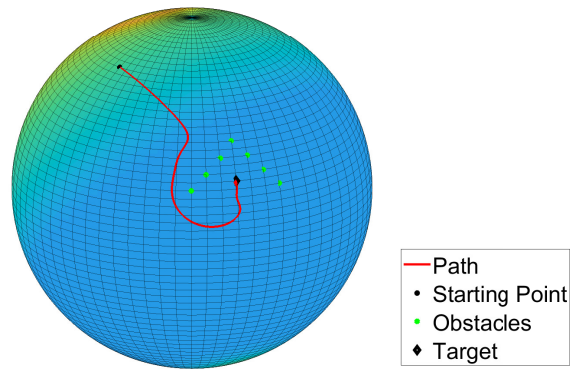


Figure 4.5: Fifth test - Barrier bypassing and target approaching.

VARP Control Theory on $SO(3)$ Manifold

In this chapter we aim to apply the VARP Principle for the control of dynamic systems whose state space is the Manifold $SO(3)$. The Manifold $SO(3)$ has been chosen because its elements, known as special orthogonal matrices, could represent quadcopters attitude. Thus the final target of our work is to perform a quadcopter attitude control with the VARP Principle which was initially designed for the control of vehicles which could maneuver in a smooth surface.

5.1 Lie Groups and Algebras definitions and properties

Let us recapitulate the following definitions and properties:

Matrix Lie group: A smooth matrix manifold \mathbb{M} that is also 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* which, for any two elements $g, h \in \mathbb{M}$, is denoted by gh and endowed with the property of closure,
2. an *identity element* with respect to the multiplication, denoted by e , such that $eg = ge = g$ for any $g \in \mathbb{M}$,
3. an *inversion operation*, denoted by g^{-1} , with respect to multiplication, such that $g^{-1}g = gg^{-1} = e$ for any $g \in \mathbb{M}$.

A *left translation* $L : \mathbb{M} \times \mathbb{M} \rightarrow \mathbb{M}$ is defined as $L_g(h) := g^{-1}h$.

An instance of matrix Lie group is $SO(3) := \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = RR^T = I_3, \det(R) = +1\}$, where the symbol T denotes matrix transposition and the quantity I_3 represents a 3×3 identity matrix.

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

Lie algebra: The tangent space $\mathfrak{g} := T_e\mathbb{M}$ to a Lie group at the identity is termed *Lie algebra*. The Lie algebra is endowed with *Lie brackets*, denoted as $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$, and an *adjoint endomorphism* $\text{ad}_\xi \eta := [\xi, \eta]$. The Lie algebra associated to the group $\text{SO}(3)$ is $\mathfrak{so}(3) := \{\xi \in \mathbb{R}^{3 \times 3} \mid \xi + \xi^\top = 0\}$. On a matrix Lie algebra, the Lie brackets coincide with matrix commutator, namely $[\xi, \eta] := \xi\eta - \eta\xi$. The matrix commutator in $\mathfrak{so}(3)$ is an anti-symmetric bilinear form, namely $[\xi, \eta] + [\eta, \xi] = 0$. A pushforward map $d(L_g)_g : T_g\mathbb{M} \rightarrow \mathfrak{g}$ is denoted as dL_g for brevity.

It is convenient to define a basis of $\mathfrak{so}(3) = \text{span}(\chi_x, \chi_y, \chi_z)$ as follows:

$$\chi_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \chi_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad \chi_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (5.1)$$

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

$$x := \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto [\![x]\!] := \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \quad (5.2)$$

We briefly recall the definition of the canonical operators and the Riemannian distance for the Manifold $\text{SO}(3)$ endowed with the inner product $\langle U, V \rangle_X := \text{tr}(U^\top V)$:

$$\left\{ \begin{array}{l} \text{Exponential map : } \exp_X(V) := X \text{Exp}(X^\top V), \\ \text{Logarithmic map : } \log_X Y := X \text{Log}(X^\top Y), \\ \text{Riemannian distance : } d(X, Y) := \|\text{Log}(X^\top Y)\|_F, \\ \text{Orthogonal projection to } T_R\text{SO}(3) : \Pi_R(V) := \frac{1}{2}R(R^\top V - V^\top R), \\ \text{Left translation : } L_R(V) := R^\top V, \end{array} \right. \quad (5.3)$$

where ‘Exp’ denotes matrix exponential and ‘Log’ denotes principal matrix logarithm.

5.2 Manifold $\mathbf{SO}(3)$ and his Lie Algebra $\mathfrak{so}(3)$

As previously said, the $\mathbf{SO}(3)$ manifold is an instance of Lie group and it is known as the *Special Orthogonal Matrices Group* in 3 dimensions (hence the name $\mathbf{SO}(3)$). It is the group of all rotations in the three-dimensional Euclidean space \mathbb{R}^3 which models the spatial arena in which physical objects evolve. By definition, a rotation about the origin is a transformation that preserves the origin, the Euclidean distance (hence it is an isometry), and orientation (i.e. handedness of space). Rotations are not commutative, that is, the order in which rotations are composed makes a difference. For example, an R rotation of 90° in the $x - y$ plane followed by an S rotation of 90° in the $y - z$ plane is not the same as the rotation S followed by R . Therefore, the manifold $\mathbf{SO}(3)$ is a non-Abelian group.

Any spatial rotation could be described either with a set of three angles (namely Euler angles), or with the axis-angle representation. 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 called the Euler axis) which indicates the axis of rotation, and an angle φ describing the magnitude of the rotation about the chosen axis. Note 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 (it must be equal to 1). By Rodrigues' rotation formula, with a desired angle φ and a desired rotation axis $\hat{n} = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}^\top$ (which must be a unit vector) it is possible to determine a rotation matrix as:

$$R_{\hat{n}}(\varphi) = \hat{n}\hat{n}^\top + \cos \varphi (I_3 - \hat{n}\hat{n}^\top) - \sin \varphi \llbracket \hat{n} \rrbracket. \quad (5.4)$$

The Rodrigues' rotation formula compute a rotation matrix $R_{\hat{n}}(\varphi)$ which has all the properties required by the Lie group $\mathbf{SO}(3)$. Each element in $\mathbf{SO}(3)$ is a 3×3 orthogonal matrix, namely a square matrix 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.

Our aim is to use $\mathbf{SO}(3)$ matrices to represent quadcopter attitudes because, recalling the model designed in [16], we can notice that the dynamic evolution of the quadcopter is described by equations which involve a rotation matrix R .

5.3 VARP extension to the $\mathbf{SO}(3)$ Manifold

The system of equations (4.5) could be written for the $\mathbf{SO}(3)$ Manifold as:

$$\begin{cases} \dot{R} = R\chi, \\ \nabla_{\dot{R}}\dot{R} = \alpha R\hat{\chi} - \mu\dot{R} - \text{grad}_R V, \end{cases} \quad (5.5)$$

but $\nabla_{\dot{R}}\dot{R} = R\dot{\chi}$ (the proof follows) and $\dot{R} = R\chi$, therefore the second equation reads:

$$R\dot{\chi} = \alpha R\hat{\chi} - \mu R\chi - \text{grad}_R V. \quad (5.6)$$

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

$$\dot{\chi} = \alpha\hat{\chi} - \mu\chi - L_R(\text{grad}_R V), \quad (5.7)$$

therefore the system (5.5) will be:

$$\begin{cases} \dot{R}(t) = R(t)\chi(t), \\ \dot{\chi}(t) = \alpha\hat{\chi}(t) - \mu\chi(t) - L_{R(t)}(\text{grad}_{R(t)} V(R(t))), \end{cases} \quad (5.8)$$

where:

- $R(t) \in \mathbf{SO}(3)$ denotes a rotation matrix;
- $\dot{R}(t) \in T_{R(t)}\mathbf{SO}(3)$ denotes the derivative of the matrix R with respect to time, hence velocity;
- $\chi \in \mathfrak{so}(3)$ is the angular velocity; here $\mathfrak{so}(3)$ denotes the Lie algebra associated to the Lie group $\mathbf{SO}(3)$, which is supposed to be endowed with the canonical inner product $\langle \chi, \psi \rangle := \text{tr}(\chi^\top \psi)$, with associated norm $\sqrt{\langle \chi, \chi \rangle} =: \|\chi\|_{\mathbb{F}}$;
- $\dot{\chi} \in \mathfrak{so}(3)$ is the rate at which the angular velocity χ changes, therefore it represents angular acceleration;
- $\mu\chi(t) \in \mathfrak{so}(3)$ represents the friction term that opposes to the change of angular velocity. μ denotes the friction coefficient;
- $\alpha\hat{\chi}(t) \in \mathfrak{so}(3)$ denotes the self-propelling term. As in the equations (4.5) it has the same direction of χ , hence the self-propelling coefficient α is multiplied with the norm of the matrix χ taking only its direction. Frobenius norm will be used as it is the associated norm to the canonical inner product of the Lie algebra $\mathfrak{so}(3)$. If $\chi \neq 0$ it will be $\hat{\chi} := \chi/\|\chi\|_{\mathbb{F}}$, otherwise $\hat{\chi} = 0$. The self-propelling term must be handled with care because it might cause severe oscillations around the attraction point, since it forces the system to move away from it causing the controller to counter its action. Notice that, in the absence of any potentials (namely $V = 0$), the self-propelling term causes a velocity drift $\dot{\chi}$ of amplitude $\|\chi_{\text{drift}}\|_{\mathbb{F}} = \frac{\alpha}{\mu}$;
- $L_{R(t)}(\text{grad}_{R(t)} V(R(t))) \in \mathfrak{so}(3)$ denotes the gradient of the potential function translated to the algebra.

As in Section 4.2, we can write:

$$\begin{aligned} L_R\left(\text{grad}_R V(R)\right) &= L_R\left[\text{grad}_R \sum_{j=1}^N (V_j^r(d^2(R, R_j)) - V_j^a(d^2(R, R_j)))\right] = \\ &= L_R\left[\sum_{j=1}^N (\dot{V}_j^r(d^2(R, R_j)) - \dot{V}_j^a(d^2(R, R_j))) \text{grad}_x d^2(R, R_j)\right], \end{aligned} \quad (5.9)$$

where $R_j \in \mathbf{SO}(3)$ is the attractive or repulsive element, which could be a desired or undesired attitude. Note that each element exhibits both characteristics despite it could actually be an attractive or repulsive point: if it is attractive, the attractive characteristic will be predominant with respect to the repulsive one.

A purely repulsive point with index j is characterized by a coefficient $C_j^a = 0$, while a purely attractive point with index j is characterized by a coefficient $C_j^r = 0$. In the article by Bertozzi *et al.* [1], it is set $C_j^r = 0$ if the point is a target while the obstacles usually has $C_j^r \gg C_j^a$. An explanation could lie in the fact that by setting obstacles it is possible to build a path for the system, hence they must attract the system but at the same time the system must avoid them. In the case of a target, it is possible to set $C_j^r \neq 0$ if we want the system to avoid the target, namely, it must stay near it but not over it. For example, in the case of Kelly robot, we would like the Kelly to reach a tree but it cannot be in the same position of it, namely we have to set $C_j^a \gg C_j^r \neq 0$. On the other hand, in the case of a drone, we can consider to set $C_j^r = 0$ since the desired attitude does not represent a physical object. From the property (4.2), we can write:

$$L_R\left(\text{grad}_R V(R)\right) = L_R\left[2 \sum_{j=1}^N (\dot{V}_j^a(d^2(R, R_j)) - \dot{V}_j^r(d^2(R, R_j))) \log_R(R_j)\right]. \quad (5.10)$$

The attractive and repulsive potentials and their derivatives have the expressions (4.9)–(4.10), respectively, hence we can write:

$$L_R\left(\text{grad}_R V(R)\right) = L_R\left[\sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-d(R, R_j)/\ell_j^r}}{d(R, R_j)} - \frac{C_j^a}{\ell_j^a} \frac{e^{-d(R, R_j)/\ell_j^a}}{d(R, R_j)}\right) \log_R(R_j)\right]. \quad (5.11)$$

Recalling the definition in (5.3) for the Riemannian distance between two points in the manifold $\mathbf{SO}(3)$, the equation (5.11) becomes:

$$L_R\left(\text{grad}_R V(R)\right) = L_R\left[\sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^r}}}{\|\text{Log}(R^\top R_j)\|_F} - \frac{C_j^a}{\ell_j^a} \frac{e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^a}}}{\|\text{Log}(R^\top R_j)\|_F}\right) \log_R(R_j)\right]. \quad (5.12)$$

The left translation operator (defined in (5.3)) was used to transport the potential term in the Lie algebra $\mathfrak{so}(3)$:

$$L_R\left(\text{grad}_R V(R)\right) = \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} \frac{e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^r}}}{\|\text{Log}(R^\top R_j)\|_F} - \frac{C_j^a}{\ell_j^a} \frac{e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^a}}}{\|\text{Log}(R^\top R_j)\|_F}\right) R^\top \log_R(R_j). \quad (5.13)$$

Applying the definition of the logarithmic map for the manifold $\mathbf{SO}(3)$ (see (5.3)), we have:

$$R^\top \log_R(R_j) = R^\top R \text{Log}(R^\top R_j) = \text{Log}(R^\top R_j), \quad (5.14)$$

where “Log” denotes the principal matrix logarithm. Finally, we have:

$$L_R \left(\text{grad}_R V(R) \right) = \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} e^{\frac{-\|\text{Log}(R^\top R_j)\|_F}{\ell_j^r}} - \frac{C_j^a}{\ell_j^a} e^{\frac{-\|\text{Log}(R^\top R_j)\|_F}{\ell_j^a}} \right) \text{Log}(R^\top R_j). \quad (5.15)$$

Let us prove how to get equation (5.6) from equation (5.5). Starting from one of the properties of the $\mathbf{SO}(3)$ manifold, namely $R^\top R = I_3$, the derivative of its two members with respect to time is:

$$\dot{R}^\top R + R^\top \dot{R} = 0, \quad (5.16)$$

which gives $\chi^\top + \chi = 0$ with $\dot{R} = R\chi$. Deriving again the equation (5.16) 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:

$$\ddot{R}^\top R + 2\dot{R}^\top \dot{R} + R^\top \ddot{R} = 0, \quad (5.17)$$

where it results that:

$$2\dot{R}^\top \dot{R} = -2\chi^2. \quad (5.18)$$

Now, since it is possible to calculate the covariant derivative as the projection on the tangent space $T_R \mathbf{SO}(3)$ of the naive* derivative \ddot{R} , we can write:

$$\nabla_{\dot{R}} \dot{R} = \frac{R}{2} (R^\top \ddot{R} - \ddot{R}^\top R) \quad (5.19)$$

where it was used the projector operator defined in (5.3).

Matching the equations (5.16)–(5.17) as $\ddot{R}^\top R = -R^\top \ddot{R} - 2\dot{R}^\top \dot{R}$ we have:

$$\nabla_{\dot{R}} \dot{R} = \frac{R}{2} (R^\top \ddot{R} - 2\chi^2 + R^\top \ddot{R}) = -R\chi^2 + \ddot{R}, \quad (5.20)$$

therefore, it results:

$$\nabla_{\dot{R}} \dot{R} = \ddot{R} - R\chi^2. \quad (5.21)$$

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

$$\nabla_{\dot{R}} \dot{R} = R(\chi^2 + \dot{\chi} - \chi^2) = R\dot{\chi} \Rightarrow \nabla_{\dot{R}} \dot{R} = R\dot{\chi} \quad (5.22)$$

and applying the left translation to the above result, we finally have:

$$\dot{\chi} = L_R(\nabla_{\dot{R}} \dot{R}) \quad (5.23)$$

as it was written above.

* \ddot{R} is denoted as *naive* derivative since it is calculated as the difference between two vectors: the result will not belong to the tangent space at R . It has two components: one normal to the tangent space and the other parallel to the tangent space. Through the projector operator defined in (5.3) it is possible to keep the parallel component and it results that this component is equal to the covariant derivative.

5.4 Quadcopter model

We recall a quadcopter drone model from [16]:

$$\begin{cases} \dot{R} &= R\chi, \\ \dot{\chi} &= D^{-1}([\hat{J}_q, \chi^2] + [\beta, \chi] - \dot{\beta} + \tau)D^{-1}, \\ \dot{\beta} &:= (-\omega_1 + \omega_2 - \omega_3 + \omega_4)J_{\mathcal{R}}\chi_z, \\ \dot{\tau} &:= br(\omega_4^2 - \omega_2^2)\chi_x + br(\omega_3^2 - \omega_1^2)\chi_y + \gamma(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2)\chi_z, \\ \dot{q} &= v, \\ \dot{v} &= \frac{1}{2} \frac{b}{M_q} (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) R e_z - \bar{g} e_z - \frac{1}{M_q} \Gamma v, \end{cases} \quad (5.24)$$

where R denotes the attitude of the drone with respect to an inertial reference frame, χ denotes its angular velocity, the ω_i , $i = 1, 2, 3, 4$, denote the angular velocities of the four propellers, q denotes the position of the center of mass of the drone, e_z denotes the unit vector $e_z := [0 \ 0 \ 1]^T$, τ denotes the mechanical torque exerted by the propellers on the drone's body and several constants represent the physical features of the drone, such as mass, inertial coefficients and propeller efficiency parameters.

In order to perform numerical simulations, an OS4-Mini-VTOL quadrotor has been taken as reference model, as reported in [17], therefore the parameters shown in Table 5.1 were made use of.

Parameter	Symbol	Value
Overall quadrotor mass	M_q	650 g
Inertia on x axis	J_x	$7.5 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
Inertia on y axis	J_y	$7.5 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
Inertia on z axis	J_z	$1.3 \times 10^{-2} \text{ kg} \cdot \text{m}^2$
Thrust coefficient	b	$3.13 \times 10^{-5} \text{ N} \cdot \text{s}^2$
Drag coefficient	γ	$7.5 \times 10^{-7} \text{ N} \cdot \text{m} \cdot \text{s}^2$
Rotor inertia	$J_{\mathcal{R}}$	$6 \times 10^{-5} \text{ kg} \cdot \text{m}^2$
Arm length	r	0.23 m
Gravitational acceleration	\bar{g}	$9.81 \text{ m} \cdot \text{s}^{-2}$

Table 5.1: Summary table of the OS4 Mini-VTOL quadcopter parameters.

The drone model consists of a series of differential equations that need to be solved numerically on a computing platform. Again a Euler method, tailored to the $\text{SO}(3)$ case, will be made use of.



Figure 5.1: Quadcopter. Generated by Yihang Zhang's MATLAB script.

5.5 Control field design by *dynamics replacement* and error feedback

In this section, we explain how to design a control field by dynamics replacement and error feedback control through a direct transposition of VARP theory.

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

$$\begin{cases} \dot{R}_s = R_s \chi_s, \\ \dot{\chi}_s = \sigma_s(\chi_s) + u, \end{cases} \quad (5.25)$$

where $\sigma_s : \mathfrak{g} \rightarrow \mathfrak{g}$ denotes a state-transition function and $u \in \mathfrak{g}$ a Lie-algebra control field. The initial conditions of this system are $R_s(0) = R_{s,0}$ and $\chi_s(0) = \chi_{s,0}$. The aim of control is to make the dynamics of the (5.25) conform to the desired dynamics of the model-system

$$\begin{cases} \dot{R}_m = R_m \chi_m, \\ \dot{\chi}_m = \sigma_m(\chi_m), \end{cases} \quad (5.26)$$

where $\sigma_m : \mathfrak{g} \rightarrow \mathfrak{g}$ denotes a state-transition function. The initial conditions of this system are $R_m(0) = R_{m,0}$ and $\chi_m(0) = \chi_{m,0}$. The initial state of the model-system (5.26) would likely differ from the initial state of the system (5.25). A way to achieve such result is to set the control field as

$$u := \sigma_m(\chi_m) - \sigma_s(\chi_s) + \kappa L_{R_s}(\log_{R_s} R_m), \quad (5.27)$$

where $\kappa > 0$ is a control parameter. The term $\sigma_m(\chi_m) - \sigma_s(\chi_s)$ implements the *principle of dynamics replacement*, namely, it has the purpose to cancel the dynamics of the real-world system and replace it with those of the model system. The term $\log_{R_s} R_m$ 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 substitution aligns the angular velocities, which *per se*, does not align the states. In addition, this term compensates for possible mismatches in the drone model.

5.6 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 (5.25) represents a quadcopter drone, namely:

$$\sigma_s(\chi_s) := D^{-1}([\hat{J}_q, \chi_s^2] + [\beta, \chi_s] - \dot{\beta})D^{-1}, \quad (5.28)$$

while the system model (5.26) represents the dynamic prescribed by the VARP theory, namely:

$$\sigma_m(\chi_m) := \alpha \hat{\chi}_m - \mu \chi_m - L_{R_m}(\nabla_{R_m} V). \quad (5.29)$$

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 (5.25) reads

$$u := D^{-1}\tau D^{-1} \in \mathfrak{so}(3). \quad (5.30)$$

According to the control design theory introduced at the beginning of this chapter, we shall set the control field to

$$\begin{aligned} u = & \alpha \hat{\chi}_m - \mu \chi_m - L_{R_m}(\nabla_{R_m} V) - D^{-1}[\hat{J}_q, \chi_s^2]D^{-1} - D^{-1}[\beta, \chi_s]D^{-1} + \\ & + D^{-1}\dot{\beta}D^{-1} + \kappa L_{R_s}(\log_{R_s} R_m). \end{aligned} \quad (5.31)$$

Therefore, the VARP control method yields a particular value of the mechanical torque

$$\begin{aligned} \tau_{\text{VARP}} := & DuD = \alpha D\hat{\chi}_mD - \mu D\chi_mD - DL_{R_m}(\nabla_{R_m} V)D - [\hat{J}_q, \chi_s^2] - [\beta, \chi_s] + \\ & + \dot{\beta} + \kappa DL_{R_s}(\log_{R_s} R_m)D, \end{aligned} \quad (5.32)$$

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

$$\left\{ \begin{array}{l} \dot{R}_m(t) = R_m(t)\chi_m(t), \\ \dot{\chi}_m(t) = \alpha \hat{\chi}_m(t) - \mu \chi_m(t) - L_{R_m(t)}(\nabla_{R_m} V), \\ L_R(\nabla_R V) = \sum_{j=1}^N \left(\frac{C_j^r}{\ell_j^r} e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^r}} - \frac{C_j^a}{\ell_j^a} e^{-\frac{\|\text{Log}(R^\top R_j)\|_F}{\ell_j^a}} \right) \text{Log}(R^\top R_j), \\ \dot{R}_s(t) = R_s(t)\chi_s(t), \\ \dot{\chi}_s(t) = D^{-1}([\hat{J}_q, \chi_s^2(t)] + [\beta, \chi_s(t)] - \dot{\beta}(t) + \tau_{\text{VARP}}(t))D^{-1}, \\ \tau_{\text{VARP}} = \alpha D\hat{\chi}_mD - \mu D\chi_mD - DL_{R_m}(\nabla_{R_m} V)D - [\hat{J}_q, \chi_s^2] - [\beta, \chi_s] + \\ \quad + \dot{\beta} + \kappa D\text{Log}(R_s^\top R_m)D. \end{array} \right. \quad (5.33)$$

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 master is a purely algorithmic system, it may be initialized to the initial conditions of the slave (drone) system.

- The rotational speeds of the two systems are quite close to each other, namely $\chi_m \approx \chi_s$, hence their values may be taken as equal to a common value χ .

Under the above hypotheses, the control scheme would simplify to

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = D^{-1}([\hat{J}_q, \chi^2] + [\beta, \chi] - \dot{\beta} + \tau_{\text{VARP}})D^{-1}, \\ \tau_{\text{VARP}} := \alpha D\hat{\chi}D - \mu D\chi D - DL_R(\nabla_R V)D - [\hat{J}_q, \chi^2] - [\beta, \chi] + \dot{\beta}, \end{cases} \quad (5.34)$$

where the footers dropped since all quantities pertain to the quadcopter drone.

5.7 Physical realizability and control effort analysis

It is necessary to make sure that the control field prescribed by the VARP theory will be consistent with the mechanical torque generated by the four propellers, namely:

- the velocity of rotation χ cannot be too high and its component along z must be almost zero;
- the mechanical torque τ is generated by the fans, which can rotate up to a maximum limit of velocity (the propellers are small DC-brushless electrical motors, hence they cannot generate a large thrust). Moreover, the velocities $\omega_1, \omega_2, \omega_3, \omega_4$ are bound to not differ excessively from the hovering velocity, which is denoted as ω_{ss} in [16] and satisfies

$$\omega_{ss}^2 = \frac{M_q \bar{g}}{2b}. \quad (5.35)$$

In order to check whether the designed control strategy fulfills the physical realizability of the control action, we shall analyze the value of 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 $\tau_x^V := \langle \tau_{\text{VARP}}, \chi_x \rangle$, $\tau_y^V := \langle \tau_{\text{VARP}}, \chi_y \rangle$ and $\tau_z^V := \langle \tau_{\text{VARP}}, \chi_z \rangle$. The components of the control torque are bound to the values generated by the four fans, as described in (5.24), namely

$$\begin{cases} \tau_x := br(\omega_4^2 - \omega_2^2), \\ \tau_y := br(\omega_3^2 - \omega_1^2), \\ \tau_z := \gamma(\omega_2^2 - \omega_1^2 + \omega_4^2 - \omega_3^2). \end{cases} \quad (5.36)$$

We further recall the hovering condition $b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = 2M_q \bar{g}$. In addition, it is worth surveying the value of the *control effort*

$$\varepsilon := \frac{1}{2} \sqrt{\langle u, u \rangle}, \quad (5.37)$$

that quantifies the effort required to the actuators to achieve a desired control action.

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 equations (5.36):

$$\begin{cases} br(\omega_4^2 - \omega_2^2) = \tau_x^V, \\ br(\omega_3^2 - \omega_1^2) = \tau_y^V, \\ \gamma(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) = \tau_z^V. \end{cases} \quad (5.38)$$

The system (5.38) is under-determined, since it totals three equations in four unknowns, therefore a fourth constraint needs to be added by considering the drone in a hovering condition. The added constraint is:

$$\frac{b}{2}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = M_q \bar{g} \quad (5.39)$$

and the system (5.38) can be updated as

$$\begin{cases} \omega_4^2 - \omega_2^2 = \frac{1}{br}\tau_x^V, \\ \omega_3^2 - \omega_1^2 = \frac{1}{br}\tau_y^V, \\ -\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 = \frac{1}{\gamma}\tau_z^V, \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \frac{2M_q\bar{g}}{b}. \end{cases} \quad (5.40)$$

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

$$\begin{cases} \omega_1^2 = \frac{-2\gamma\tau_y^V - br\tau_z^V + 2r\gamma M_q\bar{g}}{4br\gamma}, \\ \omega_2^2 = \frac{br\tau_z^V - 2\gamma\tau_x^V + 2r\gamma M_q\bar{g}}{4br\gamma}, \\ \omega_3^2 = \frac{2\gamma\tau_y^V - br\tau_z^V + 2r\gamma M_q\bar{g}}{4br\gamma}, \\ \omega_4^2 = \frac{2\gamma\tau_x^V + br\tau_z^V + 2r\gamma M_q\bar{g}}{4br\gamma}. \end{cases} \quad (5.41)$$

The solutions (5.41) 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 the drone, the lighter the effect of the control field components on the propellers velocities. Introducing the expression (5.35), the above solutions may be rewritten as

$$\begin{cases} \omega_1^2 = \omega_{ss}^2 - \frac{1}{2br}\tau_y^V - \frac{1}{4\gamma}\tau_z^V, \\ \omega_2^2 = \omega_{ss}^2 - \frac{1}{2br}\tau_x^V + \frac{1}{4\gamma}\tau_z^V, \\ \omega_3^2 = \omega_{ss}^2 + \frac{1}{2br}\tau_y^V - \frac{1}{4\gamma}\tau_z^V, \\ \omega_4^2 = \omega_{ss}^2 + \frac{1}{2br}\tau_x^V + \frac{1}{4\gamma}\tau_z^V. \end{cases} \quad (5.42)$$

From the above expressions it is apparent how the control actions cause a deviation of the rotors velocities from the hovering velocity. Such deviations need to be small 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 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 (5.39) 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, it is necessary to increase slightly the speed of the rotors. For a tilted quadcopter (namely $R \neq I_3$), the hovering condition generalizes to:

$$\frac{b}{2}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)Re_z = M_q\bar{g}e_z. \quad (5.43)$$

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_z^\top , which leads to

$$\frac{b}{2}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = \frac{M_q\bar{g}}{e_z^\top Re_z}. \quad (5.44)$$

Notice that $|R_{(3,3)}| = |e_z^\top R e_z| \leq 1$, hence the right-hand side of the condition (5.44) is larger than (or equal to) $M_q \bar{g}$. In the case of $R \neq I_3$ the system (5.38) can be updated as:

$$\begin{cases} \omega_4^2 - \omega_2^2 = \frac{1}{br} \tau_x^V, \\ \omega_3^2 - \omega_1^2 = \frac{1}{br} \tau_y^V, \\ -\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 = \frac{1}{\gamma} \tau_z^V, \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \frac{2M_q \bar{g}}{be_z^\top R e_z}. \end{cases} \quad (5.45)$$

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

$$\begin{cases} \omega_1^2 = \frac{M_q \bar{g}}{2bR_{(3,3)}} - \frac{\tau_y^V}{2br} - \frac{\tau_z^V}{4\gamma}, \\ \omega_2^2 = \frac{M_q \bar{g}}{2bR_{(3,3)}} - \frac{\tau_x^V}{2br} + \frac{\tau_z^V}{4\gamma}, \\ \omega_3^2 = \frac{M_q \bar{g}}{2bR_{(3,3)}} + \frac{\tau_y^V}{2br} - \frac{\tau_z^V}{4\gamma}, \\ \omega_4^2 = \frac{M_q \bar{g}}{2bR_{(3,3)}} + \frac{\tau_x^V}{2br} + \frac{\tau_z^V}{4\gamma}; \end{cases} \quad (5.46)$$

introducing the (5.35) the system could be rewritten as

$$\begin{cases} \omega_1^2 = \frac{\omega_{ss}^2}{R_{(3,3)}} - \frac{\tau_y^V}{2br} - \frac{\tau_z^V}{4\gamma}, \\ \omega_2^2 = \frac{\omega_{ss}^2}{R_{(3,3)}} - \frac{\tau_x^V}{2br} + \frac{\tau_z^V}{4\gamma}, \\ \omega_3^2 = \frac{\omega_{ss}^2}{R_{(3,3)}} + \frac{\tau_y^V}{2br} - \frac{\tau_z^V}{4\gamma}, \\ \omega_4^2 = \frac{\omega_{ss}^2}{R_{(3,3)}} + \frac{\tau_x^V}{2br} + \frac{\tau_z^V}{4\gamma}. \end{cases} \quad (5.47)$$

The steady-state velocity of the OS4-Mini-VTOL quadcopter is $\omega_{ss} = \sqrt{\frac{M_q \bar{g}}{2b}} \approx 319$ rad/s. The constants that appear in the expressions (5.42) take values $2br \approx 1.44 \times 10^{-5}$ [N][m][s²] and $4\gamma = 3 \times 10^{-6}$ [N][m][s²].

5.8 Case-Study: Single attractive/repulsive point

As a case-study, let us consider a VARP-controlled drone with a single attraction point. The equations of motion may be summarized as follows:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = \alpha \hat{\chi} - \mu \chi - \left[\frac{C^r}{\ell^r} \exp\left(-\frac{\|\text{Log}(R^\top R^a)\|_F}{\ell^r}\right) - \frac{C^a}{\ell^a} \exp\left(-\frac{\|\text{Log}(R^\top R^a)\|_F}{\ell^a}\right) \right] \frac{\text{Log}(R^\top R^a)}{\|\text{Log}(R^\top R^a)\|_F}, \\ \dot{q} = v, \\ \dot{v} = \frac{b}{2M_q} \left(\sum_{i=1}^4 \omega_i^2 \right) R e_z - \bar{g} e_z - \frac{1}{M_q} \Gamma v, \end{cases} \quad (5.48)$$

where $R^a \in \text{SO}(3)$ 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 5.7. For example, one may well set

$$\sum_{i=1}^4 \omega_i^2 = \eta \frac{2M_q \bar{g}}{be_z^\top R e_z}, \quad (5.49)$$

where the coefficient η has the meaning of ‘throttle’, namely $\eta < 1$ causes a drone to descend, $\eta > 1$ causes vertical ascent, while $\eta = 1$ corresponds to hovering in midair. The value assigned to the coefficient η must be compatible with the physics of the drone.

In this case, the equations of motion may be rewritten as:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = \left(\frac{\alpha}{\|\chi\|_F} - \mu \right) \chi - \left[\frac{C^r}{\ell^r} \exp\left(\frac{-\|\text{Log}(R^\top R^a)\|_F}{\ell^r} \right) + \right. \\ \quad \left. - \frac{C^a}{\ell^a} \exp\left(\frac{-\|\text{Log}(R^\top R^a)\|_F}{\ell^a} \right) \right] \frac{\text{Log}(R^\top R^a)}{\|\text{Log}(R^\top R^a)\|_F}, \\ \dot{q} = v, \\ \dot{v} = \bar{g} \left(\frac{\eta R}{e_z^\top R e_z} - I_3 \right) e_z - \frac{1}{M_q} \Gamma v, \end{cases} \quad (5.50)$$

where the term $\bar{g}M_q \left(\frac{\eta R}{e_z^\top R e_z} - I_3 \right) e_z$ denotes the net mechanical force that the propellers are able to exert on the (center of mass of the) drone. Notice that when $R = I_3$, such net force equals $(\eta - 1) M_q \bar{g} e_z$, which further clarifies the role of the coefficient η .

5.9 Numerical integration of the Initial Value Problem

In order to simulate numerically the above control scheme, we will apply the most widely used method for the numerical integration of first order dynamical system equations, the forward Euler method.

Writing again the controlled system in (5.33) in a more compact way we have:

$$\begin{cases} \dot{R}(t) = R(t)\chi(t), \\ \dot{\chi}(t) = \sigma(\chi(t), R(t)), \\ \xi(0) = \chi_0, \quad R(0) = R_0, \end{cases} \quad (5.51)$$

where $\sigma : \mathfrak{so}(3) \times \text{SO}(3) \rightarrow \mathfrak{so}(3)$ is the right-hand side of the second equation in (5.33) and it takes a matrix $R \in \text{SO}(3)$, a matrix $\xi \in \mathfrak{so}(3)$ and all the other constant coefficients of the VARP control method.

According to the fEul method, the value of the first-order derivative can be approximated by means of the right-side incremental ratio:

$$\dot{\chi}(t) = \frac{\chi(t+h) - \chi(t)}{h} + \varepsilon(t). \quad (5.52)$$

Ignoring the error term $\varepsilon(t)$ and applying a uniform time discretization, we obtain:

$$\frac{\chi_{k+1} - \chi_k}{h} = \sigma(\chi_k, R_k), \quad k = 1, 2, 3, \dots, \quad (5.53)$$

which gives rise to the iteration:

$$\chi_{k+1} = \chi_k + h\sigma(\chi_k, R_k), \quad k = 1, 2, 3, \dots \quad (5.54)$$

All the previous terms live in the same space, namely $\mathfrak{so}(3)$. We have:

$$\sigma(\chi_k, R_k) = \alpha\hat{\chi}_k - \mu\chi_k - L_{R_k}(\nabla_{R_k}V(R_k)). \quad (5.55)$$

Then we can consider the numerical integration of the first equation of (5.51) which is

$$\dot{R}(t) = R(t)\chi(t) := \nu(R(t), \chi(t)). \quad (5.56)$$

Note that $R \in \text{SO}(3)$, $\dot{R} \in T_R\text{SO}(3)$ and $\chi \in \mathfrak{so}(3)$. The solution of (5.56) could be approximated by fEul and discretized as

$$\dot{R}(t) = \nu(R(t), \chi(t)) = \frac{R(t+h) - R(t)}{h} + \varepsilon(t) \Rightarrow R_{k+1} = R_k + h\nu(R_k, \chi_k) \quad (5.57)$$

but it is not legitimate to make the sum operation on the right-hand side of the arrow because $R_k \in \text{SO}(3)$ and $\nu(R_k, \chi_k) \in T_{R_k}\text{SO}(3)$. Since they live in different spaces we can use the exponential map, which is an operator that maps a pair $(R, V) \in T\text{SO}(3)$ into a point in the $\text{SO}(3)$ manifold. Ultimately, the equation (5.57) should be written as:

$$R_{k+1} = \exp_{R_k}(h\nu(R_k, \chi_k)) \Rightarrow R_{k+1} = R_k \text{Exp}(h R_k^\top \nu(R_k, \chi_k)). \quad (5.58)$$

The equations that we have to implement numerically are:

$$\begin{cases} R_{k+1} = R_k \text{Exp}(h R_k^\top \nu(R_k, \chi_k)), \\ \chi_{k+1} = \chi_k + h\sigma(\chi_k, R_k), \\ k = 0, 1, 2, 3, \dots, \end{cases} \quad (5.59)$$

where:

$$\begin{cases} \nu(R_k, \xi_k) = R_k \chi_k, \\ \sigma(\chi_k, R_k) = \alpha\hat{\chi}_k - \mu\chi_k - L_{R_k}(\nabla_{R_k}V(R_k)), \end{cases} \quad (5.60)$$

therefore the complete equations are:

$$\begin{cases} R_{k+1} = R_k \text{Exp}(h R_k^\top R_k \chi_k) \Rightarrow R_{k+1} = R_k \text{Exp}(h \chi_k), \\ \chi_{k+1} = \chi_k + \alpha h \hat{\chi}_k - \mu h \chi_k - h L_{R_k}(\nabla_{R_k}V(R_k)), \\ L_R(\nabla_R V(R)) = \sum_{j=1}^N \left[\left(\frac{C_j^r \exp\left(\frac{-\|\text{Log}(R^\top R_j)\|_F}{\ell_j^r}\right)}{\|\text{Log}(R^\top R_j)\|_F} \right) + \right. \\ \left. - \frac{C_j^a \exp\left(\frac{-\|\text{Log}(R^\top R_j)\|_F}{\ell_j^a}\right)}{\ell_j^a \|\text{Log}(R^\top R_j)\|_F} \right] \text{Log}(R^\top R_j), \\ k = 0, 1, 2, 3, \dots \end{cases} \quad (5.61)$$

Double VARP Control for Quadcopter Motion in the Space

The SO3-VARP has been designed in the previous chapter for the control of the quadcopter's attitude: given a desired attitude, the VARP principle will provide the torque τ_{VARP} that the propellers must produce in order to steer the drone into the desired attitude.

Recalling the quadcopter model (5.24) it is possible to notice that the SO3-VARP will affect only the dynamic evolution of the attitude R and the angular velocity χ of the quadcopter.

Since the quadcopter is a physical system which could maneuver in the 3D space, we must deal also with the last two equations, namely \dot{q} and \dot{v} , which respectively represent the quadcopter speed and acceleration in the 3D space.

Implementing the equations of speed and position it is possible to notice that with only the SO3-VARP, the drone will be wandering in the space. We aim to direct the quadcopter towards a defined target in space. In this section it will be explained how to exploit the notion of VARP regulation to achieve position control applied to quadcopter guidance.

6.1 Double VARP Control Theory

Recalling the equations (5.24), the main idea is to use two instances of the VARP regulation theory. Namely, two controllers will work concurrently to achieve two goals:

- An instance (VARP₁) will serve to stabilize the attitude of a drone during flight. The purpose of this instance of VARP controller will be to make sure the tilt of the drone keeps limited, namely that the $x - y$ plane of the drone will always stay in an almost-horizontal orientation, hence stabilizing its flight mode;
- An instance (VARP₂) will regulate the tilting of the drone in such a way that it is steered toward a pre-defined target point in space, in spite of its initial position, tilting and rotation speed.

To what concern positional regulation, from the last equation in the quadcopter model (5.24), it is possible to notice that the thrust of the drone will be always normal to its $x - y$ plane, therefore, in order to steer the quadcopter toward a desired direction, the 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. Particularly the z axis of the drone must be aligned to a desired direction.

A VARP controller is used to calculate a desired attitude at every instant, and a mechanical torque is calculated that tends to align the z axis of the drone to the 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 given to fill up the vertical gap between the drone's actual position and the target, which may be achieved independently by setting the throttle coefficient proportional to such gap.

The equations of motion may be summarized as follows:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = -\mu\chi + \tau_1^V + \tau_2^V, \\ \dot{q} = v, \\ \dot{v} = \bar{g} \left(\frac{\eta R}{e_z^\top R e_z} - I_3 \right) e_z - \frac{1}{M_q} \Gamma v, \end{cases} \quad (6.1)$$

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

In the following, we shall denote as $q_t \in \mathbb{R}^3$ the set point. 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

$$\epsilon := q_t - q. \quad (6.2)$$

The z component of the displacement will be denoted as $\epsilon_z := \epsilon^\top e_z$.

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:

$$\eta = \exp(C_\eta \epsilon_z), \quad (6.3)$$

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

Attitude stabilization (VARP₁): The controller VARP₁ must output a control torque τ_1^V that tends to keep the drone as horizontal as possible. Therefore, it may be set up as a SO(3)-VARP without repulsion points and with an attraction point set to the identity 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 Bertozzi *et al.* [1], hence we relaxed the potential function to a Gaussian one, namely, we have defined

$$V_1(R) := \frac{1}{2}C_1\ell_1 \exp\left(\frac{-d^2(R, I_3)}{\ell_1}\right), \quad (6.4)$$

where $d : \text{SO}(3)^2 \rightarrow \mathbb{R}$ denotes a Riemannian distance. Notice that the coefficients have been written in a slightly different way, which facilitates the implementation of the control action. We thus have:

$$\tau_1^V := L_R(\text{grad}_R V_1) = -C_1 \exp\left(\frac{-\|\text{Log}(R)\|_{\mathbb{F}}^2}{\ell_1}\right) \text{Log}(R), \quad (6.5)$$

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

Positional control (VARP₂): The main idea carried out in this section consists in defining a mechanical torque to steer a drone toward a set point. The controller VARP₂ must output a control 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

$$V_2(\epsilon) := \frac{1}{2}C_2\ell_2 \exp\left(\frac{-\|\epsilon\|^2}{\ell_2}\right). \quad (6.6)$$

The corresponding mechanical torque is defined as

$$\tau_2^V := \llbracket (R e_z) \wedge (-\text{grad}_\epsilon V_2) \rrbracket = C_2 \exp\left(\frac{-\|\epsilon\|^2}{\ell_2}\right) \llbracket (R e_z) \wedge \epsilon \rrbracket, \quad (6.7)$$

where \wedge denotes cross vector product. The analytical form of the mechanical torque $\llbracket (R e_z) \wedge (-\text{grad}_\epsilon V_2) \rrbracket$ has the effect to orientate the z axis of the drone toward the positional displacement ϵ .

The complete set of equation describing the controlled drone read:

$$\begin{cases} \dot{R} = R\chi, \\ \dot{\chi} = -\mu\chi - C_1 \exp\left(\frac{-\|\text{Log}(R)\|_{\mathbb{F}}^2}{\ell_1}\right) \text{Log}(R) + \\ \quad + C_2 \exp\left(\frac{-\|q_t - q\|^2}{\ell_2}\right) \llbracket (R e_z) \wedge (q_t - q) \rrbracket, \\ \dot{q} = v, \\ \dot{v} = \bar{g} \left(\frac{e^{C_\eta (q_t - q)^\top e_z}}{e_z^\top R e_z} R - I_3 \right) e_z - \frac{1}{M_q} \Gamma v, \end{cases} \quad (6.8)$$

which may be implemented by a numerical method (see Section 5.9) 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 $R_0 e_z$ 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).

6.2 Rotors' speed

As it was discussed in the Section 5.7, it is important to make sure that the control field resulting from the VARP theory is consistent with the mechanical torque generated by the four propellers.

First of all we must define the control law that will be used in the system (5.34), namely we must define the τ_{VARP} law. The control law could be designed as follow:

$$\tau_{\text{VARP}} := -\mu D\chi D + D\tau_1^V D + D\tau_2^V D - [\hat{J}_q, \chi^2] - [\beta, \chi] + \dot{\beta}. \quad (6.9)$$

Placing the control law in (5.34), we obtain the equations showed in (6.1).

It is also important to calculate the rotors' speed. In order to do this we can take again the system (5.38) which is under-determined. This time we must use as last condition the equation (5.49). The final system to compute the speed of rotors will be:

$$\begin{cases} br(\omega_4^2 - \omega_2^2) = \tau_x^V, \\ br(\omega_3^2 - \omega_1^2) = \tau_y^V, \\ \gamma(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) = \tau_z^V, \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 = \eta \frac{2M_q \bar{g}}{b e_z^+ Re_z}. \end{cases} \quad (6.10)$$

The above system of equations is linear in the unknowns ω_1^2 , ω_2^2 , ω_3^2 and ω_4^2 and admits the same solutions of the system (5.46). Rooting the square values of the rotors' speed from (5.46) it is possible to calculate rotors' angular speeds to make the drone follow the trajectory computed by the Double VARP control method. The rotors' speeds just obtained are expressed in radians but, in the field of quadcopters, it is common to express them in RPM (revolutions per minute). In order to convert radians to RPM we will use:

$$\omega_{\text{RPM}} = \frac{30}{\pi} \omega_{\text{rad}}, \quad (6.11)$$

where ω_{RPM} is the value of angular speed in RPM and ω_{rad} is the value in radians computed by means the control method.

6.3 Simulations and results of D-VARP Method

In this section we will test how D-VARP can be effective in controlling attitude and position of the drone introduced in Section 5.4.

As already explained above, the two instances of D-VARP act simultaneously and must be balanced in such a way that one does not prevail over the other. To do this it is necessary that the parameters of the two instances are well balanced. As a matter of fact, in the event that the VARP₁ instance prevails over the VARP₂ instance, the drone would tend to maintain an attitude too close to the identity matrix I_3 . This would cause a predominantly vertical flight and the drone would struggle to reach the target in an acceptable time. On the contrary, in the event that the VARP₂ instance prevails over the VARP₁ instance, the drone would tend to assume risky positions, which could also lead to overturning. Indeed, in the event that the drone is required to reach a target placed horizontally with respect to its starting position, the drone would have no control over the attitude's safety and could reach an attitude such that it overturns.

Furthermore, also the C_η coefficient relative to vertical positioning is fundamental to obtain a flight similar to the ideal one. Indeed, a not balanced C_η parameter could cause a too early or too late vertical positioning. In the event that the vertical positioning is anticipated, the drone will reach the target altitude in the first moments of its flight and then assume the correct attitude that will allow it to move horizontally. This imbalance between coefficients would therefore lead to a flight path very different from the optimal one, as the drone would move first vertically, then horizontally but not simultaneously. Similarly, if the vertical positioning is delayed, the drone would move first horizontally, then vertically. Thus, in this section, contrary to what was done in previous tests, we won't modify the parameters of the two D-VARP instances, because we managed to find a set of universal parameters that are almost optimal for each test. Indeed, after several tests, we have noticed that this set of parameters is efficient in almost all cases; this is an evident feature given that many sets of parameters work for a small variety of situations. Instead of focusing on evaluating sets of parameters, we will focus mainly on the quadcopter's ability to reach the target and remain in balance under several conditions. Indeed, we will see how the drone will behave under ideal and not ideal initial conditions, under disturbing circumstances, such as gusts of wind, and in the case in which the path is composed of intermediate targets. Editing D-VARP parameters involves, as we can imagine, the precision of the drone in following the minimum path and the speed with which it reaches the target. The set of parameters found, as shown in Table 6.1, guarantees satisfactory speed and precision.

As mentioned earlier, two scenarios for the departure of the drone can be distinguished: optimal and non-optimal conditions. By optimal conditions we mean the situation in which the drone is at the initial instant with a translational and rotational speed equal to zero and with an initial attitude almost equal to the I_3 matrix: this means that the drone is motionless and upright. These are defined as optimal because it is uncommon that the drone is in these conditions when it is assigned a target to reach. Furthermore, these conditions mean that the drone is as facilitated as possible to reach a certain attitude, necessary to reach the target. By non-optimal conditions, however, we mean situations

D-VARP Parameters	
Parameter	Value
C_1	2
ℓ_1	500
C_2	$2.6 \cdot 10^{-3}$
ℓ_2	$6 \cdot 10^3$
C_η	$7.5 \cdot 10^{-4}$

Table 6.1: Parameters for D-VARP tests.

very close to the real ones, in which the drone is at a non-negligible speed and with a different attitude from the identity matrix I_3 . Non-optimal conditions often imply a more difficult control of the drone, especially in the initial instants, as the latter is forced to make a greater effort to find the right attitude which allows to reach the target. However, in the most fortunate cases, there are non-optimal conditions that can benefit the drone, as they facilitate the achievement of the target. We will illustrate the behavior of the drone during its flight using through the following figures:

- **Target approaching:** through the use of two 3D graphs, the drone path will be reported from two different points of view. The black dashed line connects the starting point with the target. It represents the shortest path which the drone can travel (henceforth we will call it as *minimum path*). The solid blue line represents the path taken by the drone. The starting point is represented with a green dot, the target with a green square, and the quadcopter's arrival point with a blue diamond. More details about this figure will be provided in the related tests;
- **Distance to target:** this panel indicates the remaining distance from the target, as the time t changes. If this distance $d := \|q_t - q\|$ reaches zero, it means that the drone is located in the target position;
- **Inclination to vertical:** in order to understand how much the drone tilts during its flight, this panel indicates its inclination with respect to the vertical axis. This inclination will be represented with: $\theta := \arccos[e_z^\top Re_z]$;
- **Torque components:** this panel reports the torque that the rotors must exert in order to allow the drone to tilt in the right way and reach the target. These torque components are provided by the D-VARP control method;
- **Control effort:** this graph report the effort required to the rotors by the control law;
- **Rotors' speed:** by using this graph, the rotors' speed are reported as time t changes. We expect that the rotor speeds tend to hovering value as the drone approaches the target. The rotors' speed will be indicated with four lines of different colors. For the first test a more detailed overview of these speeds will be provided. In particular, this detail will be used to understand the speed trend in the first moments of flight;
- **η coefficient:** this graph shows the trend of the η coefficient over time. According to Equation (6.3), we expect this coefficient to have an exponential trend.

6.3.1 First Test

In this test, we set the target to a lower altitude with respect to the starting position. This test was carried out considering ideal initial conditions. As we can see from Figure 6.1, the drone follows a path that is very close to the minimum one, which means that the parameters used for this simulation are effective. Choosing a worse set of parameters would cause a slow flight or a path very different from the minimum one, which is represented by the dashed line. From Figure 6.1 we can see that already from the starting instant the drone follows a different path from the optimal one, this is due to the transient time during which the drone settles itself into the correct attitude that allows it to reach the target.

In Figure 6.2 we can see the panels which give us an idea of how the drone behaves during its flight. As we could also imagine from the previous figure, the distance panel shows us that the drone is able to get closer to the target gradually. By focusing on the drone's inclination, we can see that the latter is more tilted in the initial moments of its flight. This happens because in the initial instants the drone tilts until it reaches a desired attitude, which allows to reach the target. Once the target is reached, the inclination of the drone is zero; this means that the drone is upright with respect to the ground. As shown by the panel, the drone's attitude gets closer to the I_3 matrix as the quadcopter approaches the target. The torque values, as shown by the relative graph, tend to zero very quickly, particularly in the first moments of flight. Furthermore, the quantities assumed by these values are very small. The same considerations also apply to the control effort panel: the control effort values remain very small and tend to zero quickly. These last values are larger in the first moments of flight because the drone has to change attitude quickly. We can also notice, as we could have imagined, that the η coefficient increases with an exponential trend, in accordance with the Equation (6.3). From the rotors' speed panel we can notice how the velocities increase until the hovering value (RPM_{ss}) is reached. Once the hovering value has been reached, the drone is in balance and in proximity to the target. In Figure 6.3 we can see the trend of rotors' speed in the first instants of simulation. We can note that in these instants rotors' speed values are slightly different from each other: this difference between the values involves the inclination of the drone. In the next tests this graph will not be reported for redundancy reasons, however the behavior of the drone remains similar to that shown in this test.

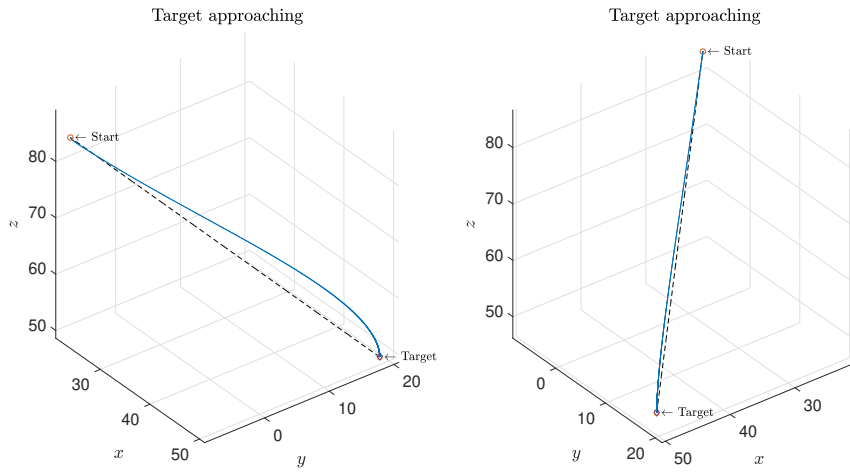


Figure 6.1: First test - Target approaching from two different perspectives.

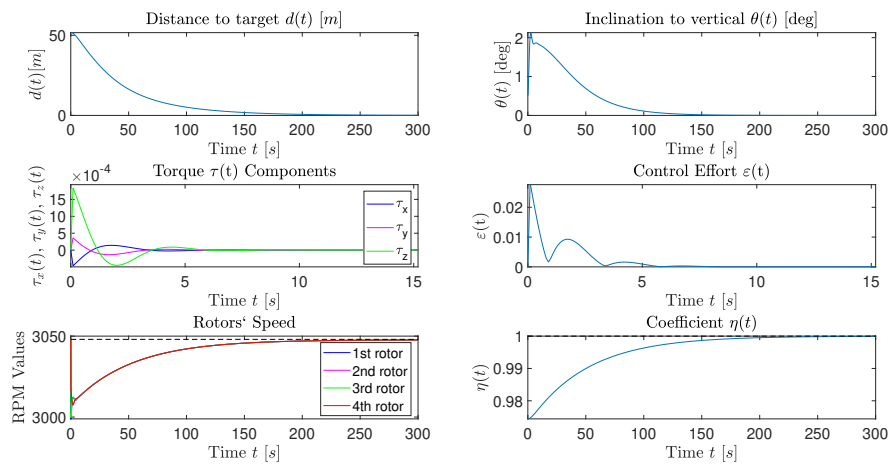


Figure 6.2: First test - Quadcopter's flight overview.

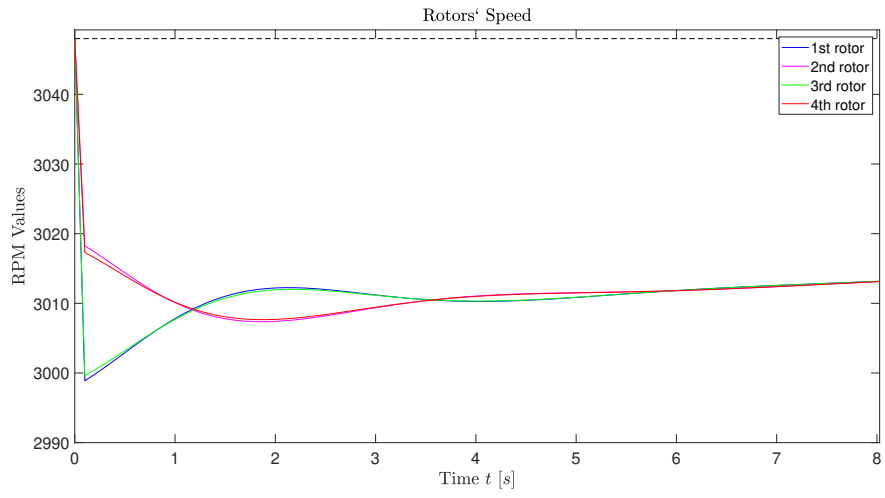


Figure 6.3: First test - Rotors' speed in the first instants.

6.3.2 Second Test

This test was carried out considering the initial conditions as optimal. In this case, the drone is at a lower altitude than the target's one. The results obtained are very similar to those of the previous test, with the exception of the η coefficient which in this case has a decreasing trend. Furthermore, the rotor speeds must be greater than the hovering speed while the quadcopter is approaching the target: this is because the starting point has a lower altitude with respect to the target, so it must overcome the altitudinal gap increasing the speed of the rotors above the hovering value. Figures 6.4 and 6.5 show the results of this test.

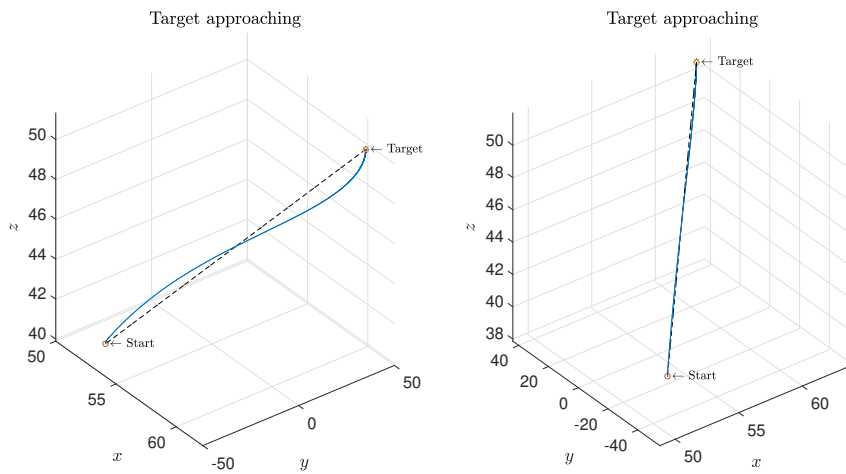


Figure 6.4: Second test - Target approaching from two different perspectives.

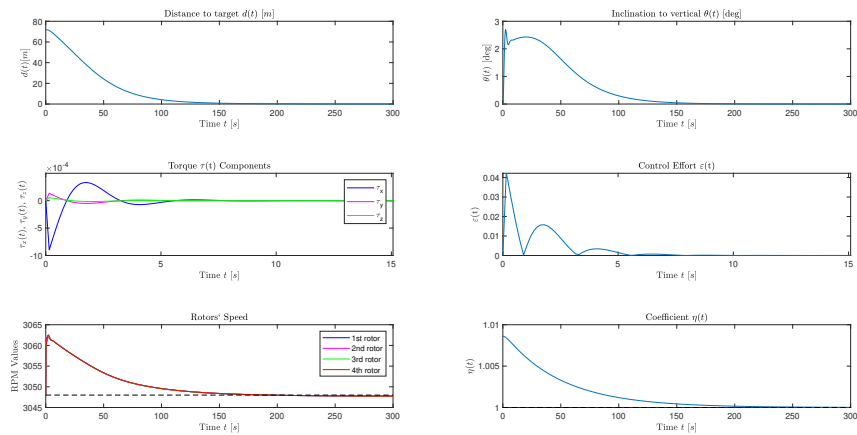


Figure 6.5: Second test - Quadcopter's flight overview.

6.3.3 Third Test

In this test we positioned the target exactly below the starting position, in such a way as to make the quadcopter follow a vertical path. As we can see from Figure 6.6, the quadcopter follows a path that can be considered completely vertical, since the movement along the \hat{x} and \hat{y} directions is of the order of millimeters. Figure 6.7 shows an overview about the drone's flight. It is interesting to notice from the "Inclination to vertical" panel how the drone maintains itself in a vertical position throughout its flight, since the maximum inclination is in the order of hundredths of a degree. Furthermore, we can also notice that the control effort and the torque values are two orders of magnitude smaller than in the previous tests; this was predictable since, as previously mentioned, these two values depend on how different the desired attitude is from the identity matrix I_3 .

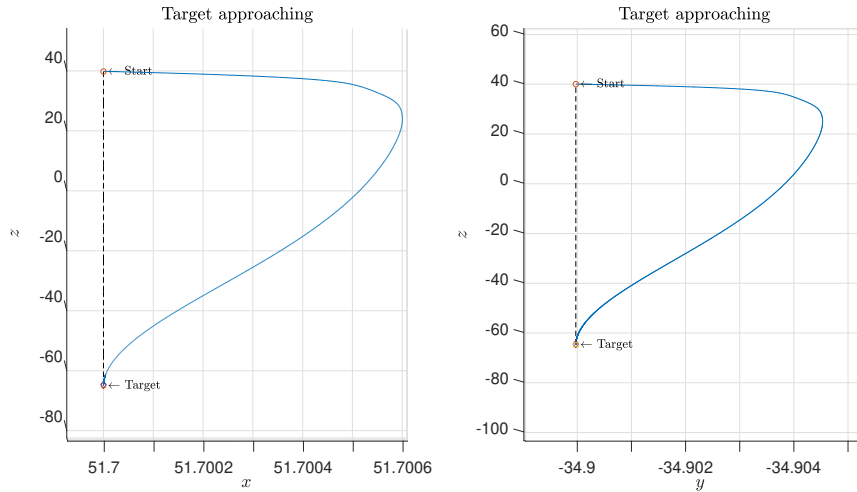


Figure 6.6: Third test - Target approaching from two different perspectives.

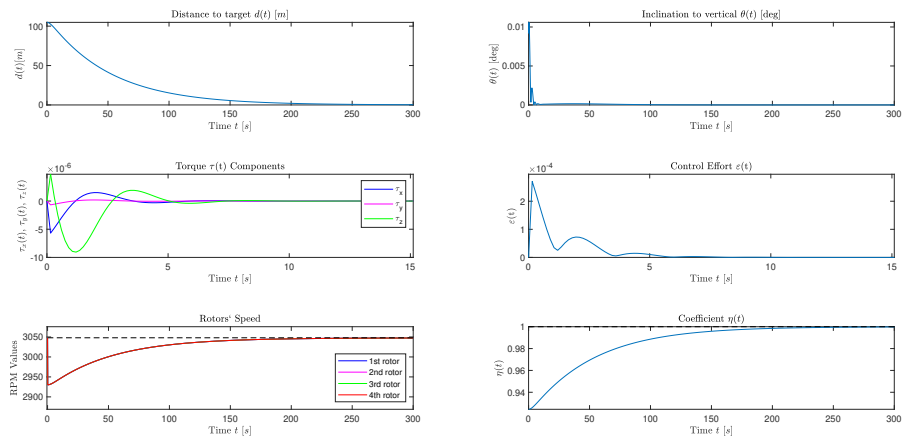


Figure 6.7: Third test - Quadcopter's flight overview.

6.3.4 Fourth Test

Similarly to what was done in the previous test, in this simulation we positioned the target exactly above the starting position, in such a way as to make the quadcopter follow a vertical path. Also in this case, as Figure 6.8 shows, the drone follows a vertical path, with only small movements along the \hat{x} and \hat{y} directions of the order of millimeters. The considerations made regarding Figure 6.7 are also valid for Figure 6.9. This test was carried out considering optimal initial conditions.

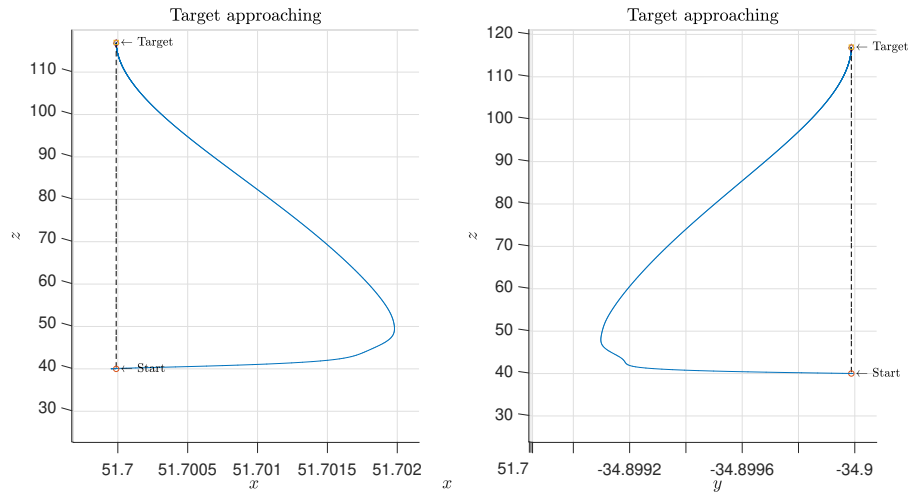


Figure 6.8: Fourth test - Target approaching from two different perspectives.

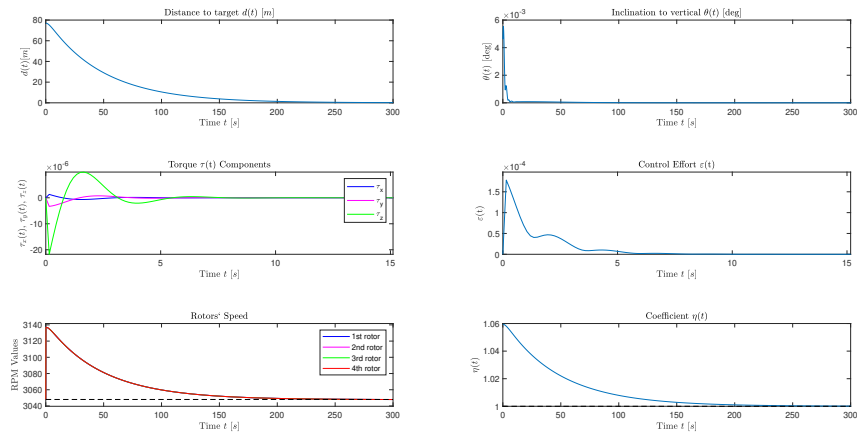


Figure 6.9: Fourth test - Quadcopter's flight overview.

6.3.5 Fifth Test

In this test we positioned the target at the same altitude as the starting point, in order to make the drone perform a horizontal flight. This test was carried out considering optimal initial conditions. Figure 6.10 shows the path traveled by the drone, which can be considered horizontal, given that the altitude variation is in the order of tenths of a millimeter. Particularly worthy of note is the fact that, as shown in Figure 6.11, the rotors of the drone assume a speed value almost equal to the hovering one: $\text{RPM}_{\text{ss}} \approx 3048$. Furthermore, as we might have expected, the value of η is approximately equal to 1.

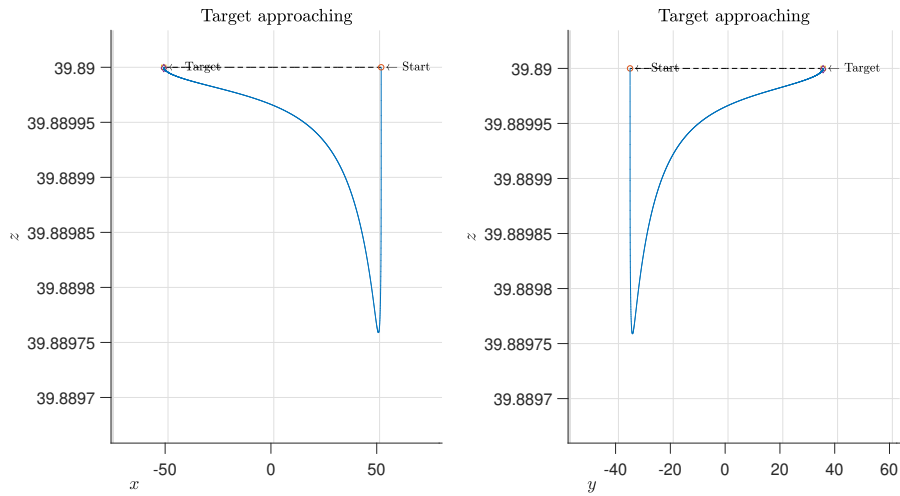


Figure 6.10: Fifth test - Target approaching from two different perspectives.

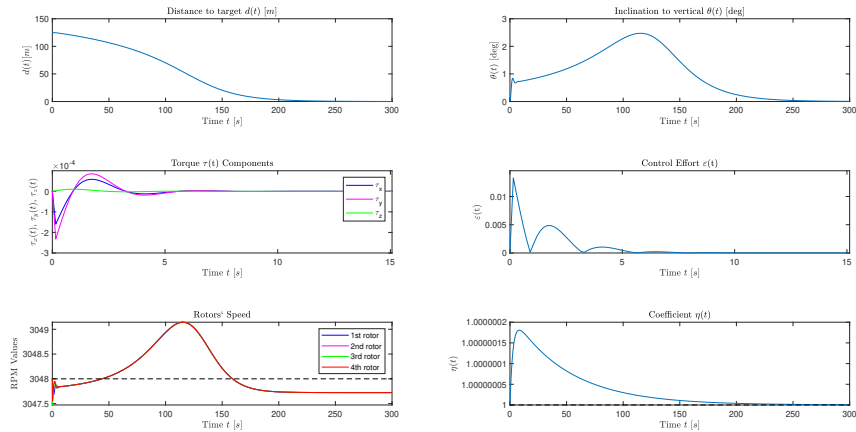


Figure 6.11: Fifth test - Quadcopter's flight overview.

6.3.6 Sixth Test

This test was done considering non-optimal initial conditions. Indeed, the drone has a starting speed and inclination not favorable to reaching the target. It can be seen from Figure 6.12 how the drone is initially moving along a direction which does not coincide with the one connecting the target with the initial position. From Figure 6.13 it is interesting to notice how the drone is initially tilted by approximately 20 degrees with respect to the attitude corresponding to the identity matrix I_3 . This also causes increased control effort and torque. Despite the unfavorable initial conditions, the drone manages to reach the target.

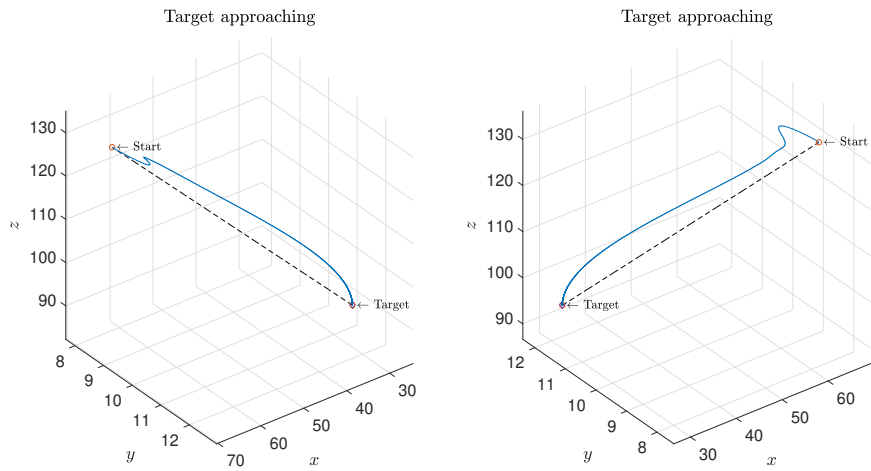


Figure 6.12: Sixth test - Target approaching from two different perspectives.

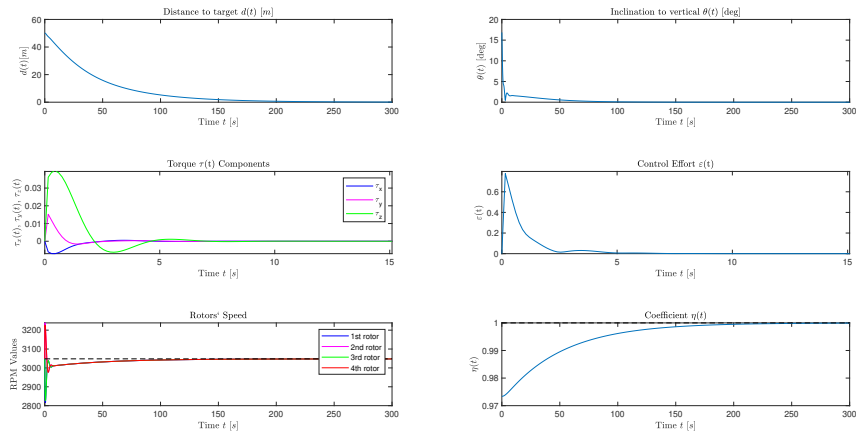


Figure 6.13: Sixth test - Quadcopter's flight overview.

6.3.7 Seventh Test

In this test we will study what happens if the quadcopter is hit by impulsive forces, namely, recalling the acceleration equation from the quadcopter model (5.24), we have considered:

$$\dot{v} = \frac{1}{2} \frac{b}{M_q} (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) Re_z - \bar{g}e_z - \frac{1}{M_q} \Gamma v + \frac{F_{\text{imp}}}{M_q}, \quad (6.12)$$

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

$$F_{\text{imp}}(t) := \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} (U(t - T_{\text{hit}}) - U(t - T_{\text{hit}} - \Delta T_{\text{hit}})), \quad (6.13)$$

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 of a disturbance and ΔT_{hit} denotes its duration. Notice that the values of the force components are chosen so that the entries of $\frac{F_{\text{imp}}}{M_q}$ range approximately in $[-5, 5]$ (m/s²), which is approximately half of gravitational acceleration.

This test is useful to simulate the presence of a gust of wind during the flight of the drone. This type of disturbance is predictable in a real situation, as drones in general perform their tasks outdoors. In this case we tried to simulate a mild gust of wind with this property: $f_x = f_y = f_z = 0.5$ N, $T_{\text{hit}} = 75$ s, $\Delta T_{\text{hit}} = 1.5$ s. The choice of T_{hit} was made in such a way that the gust of wind occurs just before the drone reaches the target, as we can see from the Figure 6.14. Despite the disturbance, the drone manages to reach the target thanks to the fact that D-VARP is effective even in unfavorable situations, as seen in previous tests in which the initial conditions were not optimal. The effect of this disturb can also be seen from the panels in Figure 6.15: the distance to the target slightly increases, the inclination of the drone and the speed of its rotors change as soon as the disturb occurs. It is interesting to notice that the drone, due to the disturbance, switch from a lower to a higher altitude with respect to the target. This can also be seen from the rotor's speed and η panels.

As we can imagine, we can also subject the drone to repetitive gusts of wind: depending on how unfavorable the external conditions are, the drone will be able to reach the target in times that will be longer than those seen with favorable conditions. Subjecting the drone to external forces also means requiring the rotors to make a considerable effort: it may therefore happen that the rotors do not have the necessary power to overcome external forces, making it impossible to control the drone.

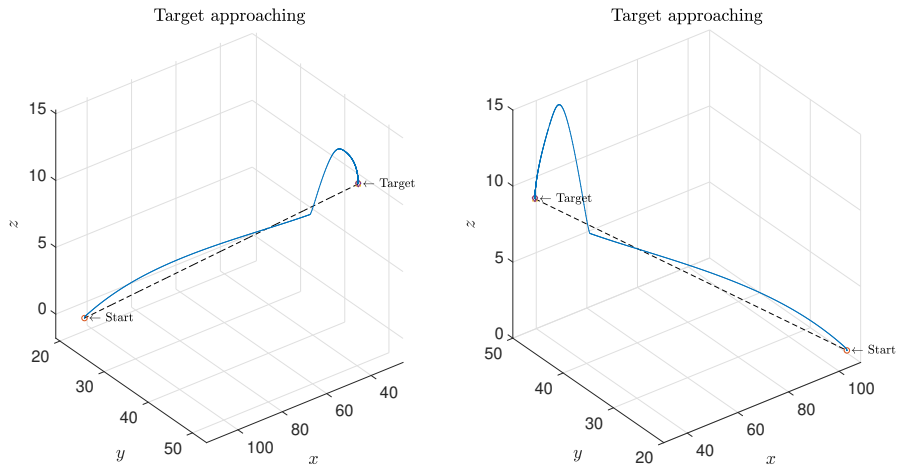


Figure 6.14: Seventh test - Target approaching from two different perspectives.

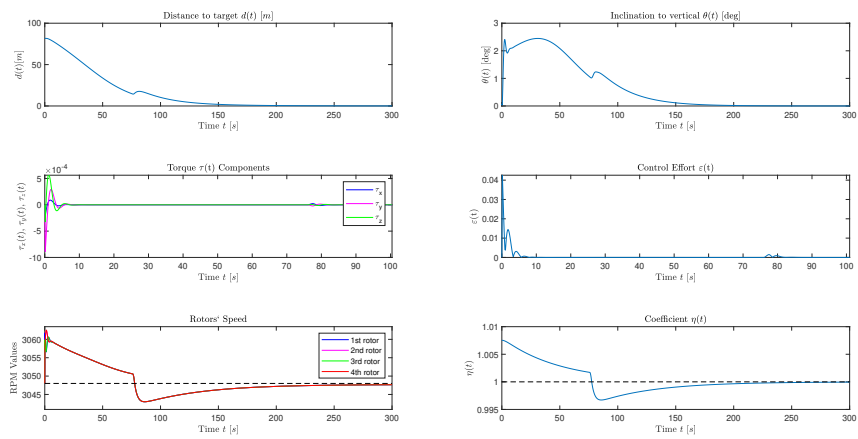


Figure 6.15: Seventh test - Quadcopter's flight overview.

6.3.8 Eighth Test

In this test we have positioned three targets which must be reached by the quadcopter in the proper order. As we can see from Figure 6.16, the drone follows a path that is very similar to the minimum one. This feature of the drone of following an almost minimal path can be very useful for many purposes. For example, if we want to pass the drone inside a curved tunnel, we could locate intermediate targets between the entrance and the exit so that the drone does not hit the walls of the tunnel. From the Figure 6.17 we can see an overview about the drone's flight: it is important to note that the control effort suddenly increases in correspondence with the target change.

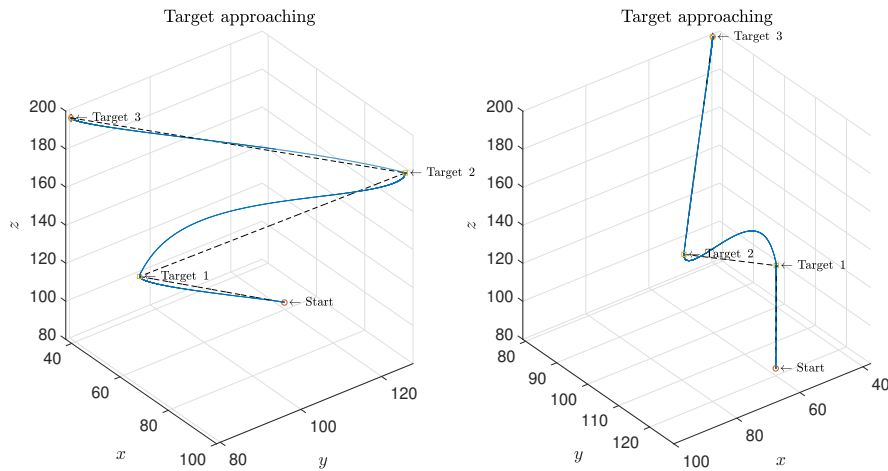


Figure 6.16: Eighth test - Target approaching from two different perspectives.

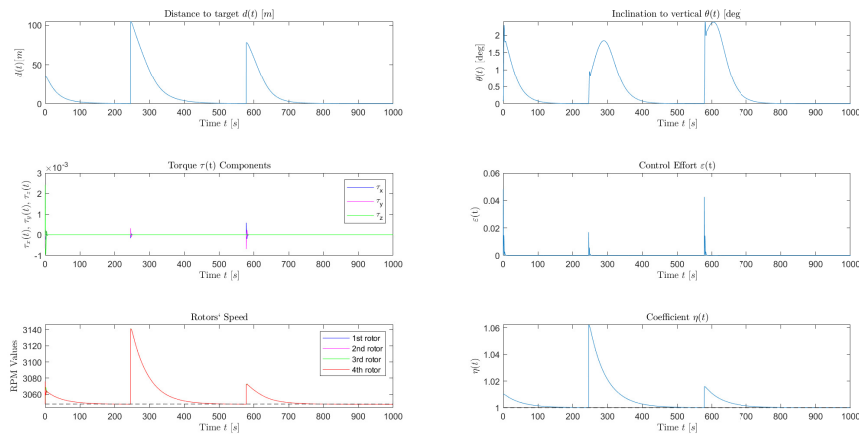


Figure 6.17: Eighth test - Quadcopter's flight overview.

6.3.9 Ninth Test

Similarly to what we did in the previous test, we placed four intermediate targets between the final target and the initial position. The results, shown in Figures 6.18 and 6.19, are consistent with the previous ones, as we might have expected. This technique of locating intermediate targets can also be used for obstacles-avoidance: in the event that there is an obstacle between the position of the quadcopter and the target, the drone can reach intermediate targets in order to circumvent the obstacle.

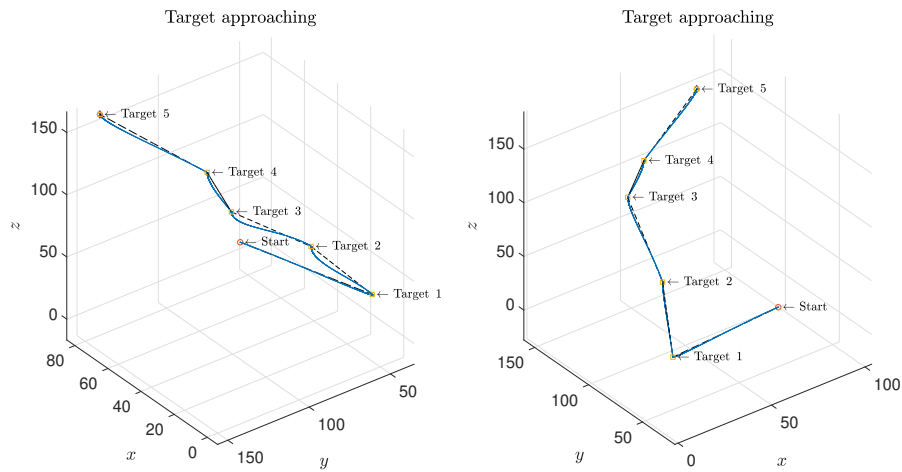


Figure 6.18: Ninth test - Target approaching from two different perspectives.

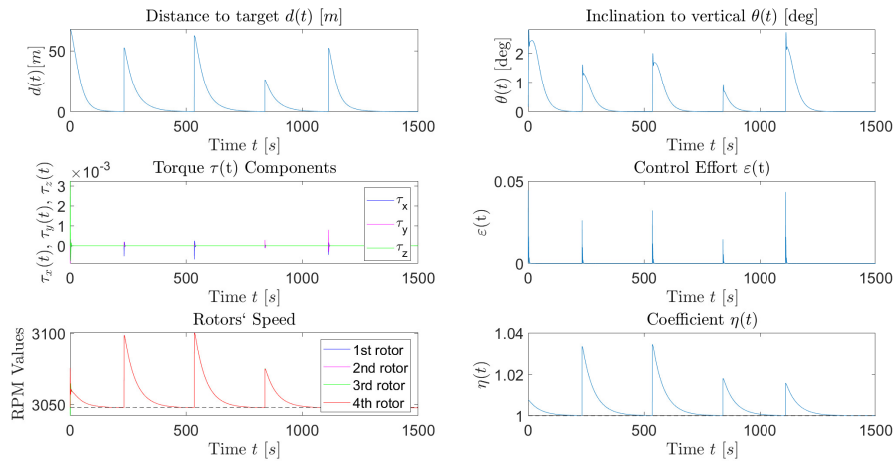


Figure 6.19: Ninth test - Quadcopter's flight overview.

Conclusions and Future Works

In this work we have extended the VARP theory developed in [1] into several state spaces in order to control different dynamic systems. In the first part of the document we were concerned with verifying whether the VARP theory could be useful for the control of a vehicle. This control method has proved effective in the control of a simple robot, the Kelly, and also of a generic dynamical system developed in a \mathbb{S}^2 manifold. Given the effectiveness of this method, we proceeded to extend this theory to a Lie group, namely $SO(3)$, in such a way as to control a drone whose equations belong to the same state space. From the tests carried out in Section 6.3 we have proven the effectiveness of the Double VARP, which is a control method that uses two instances of the VARP developed in $SO(3)$ to control the quadcopter.

The test results point out good insights for possible applications and future works. Regarding VARP method, we have seen how it can be easily applied to commonly used robots, such as robot vacuum cleaners. VARP theory is naturally very suitable for these types of applications. It is also possible to think, for these types of usage, to define dynamic potentials that mutate position and characteristics as time changes. This development can be interesting as robots usually fulfill their tasks in dynamic environments, thus it is evident the usefulness of employ dynamic potentials in order to control the robot even in the case of unexpected events that occur during the completion of the their duties.

The fact that the VARP theory can also be extended to the sphere, as we have seen in Chapter 4 makes it very interesting in those purposes where control in a spherical surface is required. As an example, we can think of possible applications in the medical field, such as in the field of robot assisted surgery. The use of S2-VARP could be useful for example in the control of a robot that deals with performing delicate surgical operations on the brain. Indeed we can think of the cranial surface or other parts of the human body as spherical or pseudo-spherical surfaces. Works has already been done on the computational anatomy, which aims to map the organs of the human body using mathematical frameworks [18, 19]. Based on what emerged from the tests in Section 6.3, we can think of applying this control method to a drone in order to confirm the simulations. The application of this control method could be carried out through the use of position and gyroscopic sensors, in order to track the attitude of the drone and its position. As already mentioned in Subsection

6.3.9, Double VARP can be used, after an improvement, for purposes such as obstacle avoidance. Indeed, we think of circumnavigating a large obstacle, such as a mountain or a building, using intermediate targets that outline the perimeter of the obstacle. Another possible improvement that can be implemented to the Double VARP could be to make the method cooperative, in such a way as to control multiple drones, making them move in the same space avoiding collisions.

In general, VARP technique can potentially be used for many purposes. Given a state space in which the control problem lives, we can think of assigning attractive potentials to the desired states, and repulsive potentials to each state that is dangerous for the control that is set. For example, we could think of applying the VARP technique to control the flow regulated by a valve, where an attractive potential is assigned to the desired flow rate, while a repulsive potential is assigned to the quantity of flow capable of breaking the valve itself.

Acknowledgements

The present research work was carried out while I was undertaking an internship at the Tokyo University of Agriculture and Technology (TUAT, Koganei Campus, Japan) in the period January-March 2020. First of all, I would like to thank Prof. Simone Fiori for making the internship in Japan possible and for giving me such an interesting and satisfying topic to study. I am extremely grateful for the help he has provided me in carrying out this work. I would also like to sincerely thank Dr. Toshihisa Tanaka for giving me the opportunity to do this work in his laboratory as well as his research team who welcomed me in the best way.

Bibliography

- [1] B. Nguyen, Y.-L. Chuang, D. Tung, C. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. Bertozzi, and R. Murray, “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)), pp. 1084 – 1089, June 2005.
- [2] C. Connolly and R. Grupen, “The application of harmonic functions to robotics,” *Journal of Robotic Systems*, vol. 10, pp. 931–946, 10 1993.
- [3] C. I. Connolly, J. B. Burns, and R. Weiss, “Path planning using Laplace’s equation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2102–2106, 1990.
- [4] D. Chang, S. Shadden, J. Marsden, and R. Olfati-Saber, “Collision avoidance for multiple agent systems,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 539–543, 2003.
- [5] S. Waydo and R. Murray, “Vehicle motion planning using functions,” in *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 2484–2491, September 2003.
- [6] H. Levine, W.-J. Rappel, and I. Cohen, “Self-organization in systems of self-propelled particles,” *Physical Review E*, vol. 63, p. 017101, June 2000.
- [7] E. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials and coordinated control of groups,” *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 2968–2973, 2001.
- [8] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, “A prototype of an autonomous controller for a quadrotor uav,” in *2007 European Control Conference (ECC)*, pp. 4001–4008, 2007.
- [9] J. De Waen, H. T. Dinh, M. H. Cruz Torres, and T. Holvoet, “Scalable multirotor uav trajectory planning using mixed integer linear programming,” in *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2017.

- [10] T. Schouwenaars, B. De Moor, E. Feron, and J. How, “Mixed integer programming for multi-vehicle path planning,” in *2001 European Control Conference (ECC)*, pp. 2603–2608, 2001.
- [11] J. Barraquand, L. Kavraki, J.-C. Latombe, R. Motwani, T.-Y. Li, and P. Raghavan, “A random sampling scheme for path planning,” *The International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.
- [12] T. Lozano-Pérez, *Spatial planning with polyhedral models*. PhD thesis, Massachusetts Institute of Technology (MIT), June 1980.
- [13] R. Deits and R. Tedrake, “Efficient mixed-integer planning for uavs in cluttered environments,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–49, 2015.
- [14] D. Domingos, G. Camargo, and F. Gomide, “Autonomous fuzzy control and navigation of quadcopters,” *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 73–78, 2016. Special issue about the 4th IFAC Conference on Intelligent Control and Automation Sciences (ICONS 2016).
- [15] M. Fadholi, Suhartono, P. Sasongko, and Sutikno, “Autonomous pole balancing design in quadcopter using behaviour-based intelligent fuzzy control,” in *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 1–6, 2018.
- [16] S. Fiori, “Model formulation over Lie groups and numerical methods to simulate the motion of gyrostats and quadrotors,” *Mathematics*, vol. 7, no. 10, 2019.
- [17] M. Becker, R. Sampaio, S. Bouabdallah, V. de Perrot, and R. Siegwart, “In-flight collision avoidance controller based only on OS4 embedded sensors,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 34, no. 3, pp. 295–297, 2012.
- [18] U. Grenander and M. I. Miller, “Computational anatomy: An emerging discipline,” *Quarterly of Applied Mathematics*, vol. 56, no. 4, pp. 617–694, 1998.
- [19] U. Grenander, M. I. Miller, and S. C. Joshi, “On the geometry and shape of brain submanifolds,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 08, pp. 1317–1343, 1997.