Pontifícia Universidade Católica
do Rio de Janeiro

**Adalberto Igor de Souza Oliveira**

# Modeling and Kinematic Control of a Mobile Robot for Autonomous Navigation in Agricultural Fields

**Dissertação de Mestrado**

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica .

Advisor: Prof. Antonio Candea Leite

Rio de Janeiro
September 2019

Pontifícia Universidade Católica
do Rio de Janeiro

**Adalberto Igor de Souza Oliveira**

# Modeling and Kinematic Control of a Mobile Robot for Autonomous Navigation in Agricultural Fields

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica . Approved by the Examination Committee.

**Prof. Antonio Candea Leite**
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Marco Antonio Meggiolaro**
Departamento de Engenharia Mecância – PUC-Rio

**Prof. Tiago Roux De Oliveira**
Universidade do Estado do Rio de Janeiro – UERJ

**Dr. Mateus Tonini Eitelwein**
Smart Consultoria Agronômica e Serviços Agrícolas Ltda – Smart Agri

Rio de Janeiro, September the 30th, 2019

**Adalberto Igor de Souza Oliveira**

Undergraduate in Computer Engineering from Faculdade Independente do Nordeste (Vitória da Conquista, Bahia, Brazil)

Bibliographic data

Oliveira, A. I. S.

Modeling and Kinematic Control of a Mobile Robot for Autonomous Navigation in Agricultural Fields / Adalberto Igor de Souza Oliveira; advisor: Antonio Candea Leite. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica , 2019.

v., 117 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica .

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Engenharia de Computação – Teses. 3. Robótica Móvel;. 4. Robótica Agrícola;. 5. Controle Não-Linear;. 6. Controle Robusto;. I. Leite, Antonio C.. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica . III. Título.

CDD: 621.3

To my parents, my sister, my kids and to my pets.

# Acknowledgments

I would like to first thank my advisor for his patience, technical support in difficult moments and lessons learned.

Then, I wish to thank my colleagues for the friendship, good vibrations, joy and team work during the last two years, the department staff (Teo and Bruno) for the support and help with the bureaucratic issues.

At least, I wish to thank The Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

# Abstract

Oliveira, A. I. S.; Leite, Antonio C. (Advisor). **Modeling and Kinematic Control of a Mobile Robot for Autonomous Navigation in Agricultural Fields**. Rio de Janeiro, 2019. 117p. Dissertação de mestrado – Departamento de Engenharia Elétrica , Pontifícia Universidade Católica do Rio de Janeiro.

In the last years, mobile robots have emerged as an alternative solution for increasing the levels of automation and mechanization in agricultural fields. In this context, the key idea of precision agriculture is to optimize the use of production inputs, crop losses, waste of water and to increase the crop production in small areas, in an efficient and sustainable manner. Agricultural robots or AgBots may be autonomous or remotely controlled, being endowed with different types of locomotion apparatus, actuation and sensory systems, as well as specialized tools which enable them to carry out a number of agricultural tasks such as, seeding, pruning, harvesting, phenotyping, monitoring and data collection. In this work, we perform a study on two type of wheeled mobile robots (i.e., differential-drive and car-like) and their application for autonomous navigation in agricultural fields. The modeling and control design is based on classical and advanced approaches, using robust control approaches such as Sliding Mode Control (first order) and Super Twisting Algorithm (second order) to deal with parametric uncertainties and external disturbances, commonly founded in agricultural fields. Verification and validation are carried out by means of numerical simulations in MATLAB and 3D computer simulations in Gazebo. Preliminary experimental tests are included to illustrate the performance and feasibility of the proposed modeling and control methodologies. Concluding remarks and perspectives are presented to summarize the strengths and weaknesses of the proposed solution and to suggest the scope for future improvements.

## Keywords

Mobile Robotics; Agricultural Robotics; Non-Linear Control; Robust Control;

## Resumo

Oliveira, A. I. S.; Leite, Antonio C.. **Modelagem e Controle Cinemático de um Robô Móvel para Navegação Autônoma em Campos Agrícolas**. Rio de Janeiro, 2019. 117p. Dissertação de Mestrado – Departamento de Engenharia Elétrica , Pontifícia Universidade Católica do Rio de Janeiro.

Nos últimos anos, os robôs móveis têm emergido como uma solução alternativa para o aumento do nível de automação e mecanização na agricultura. Neste contexto, o foco da agricultura de precisão é a otimização do uso de insumos, redução de perdas nas lavouras, redução do desperdício de água e melhorar a produtividade em áreas cada vez menores, tornando a produção mais eficiente e sustentável. Os robôs agrícolas, ou AgBots podem ser controlados remotamente ou atuar de forma autônoma, utilizando diferentes sistemas de locomoção, bem como serem equipados com atuadores e sensores que lhes permitem realizar diversas tarefas agrícolas, tais como plantio, colheita, poda, fenotipagem, monitoramento e coleta de dado, entre outros. Neste trabalho será realizado um estudo em robôs móveis com rodas direcionado para os modelos de tração diferencial e no modelo similar ao carro (com atuação do sistema de direção) e suas aplicações em navegação autônoma em ambientes agrícolas. A modelagem e o projeto de controle baseiam-se em técnica clássicas e avançadas, utilizando abordagens de controle robusto por modo deslisante, tanto de primeira como de segunda ordens (*Super Twisting Algorithm*) para lidar com incertezas e interferências externas, comumente encontradas no tipo de ambiente agrícola a que se destina. Teste de verificação e validação são realizados através de simulações numéricas (MATLAB) e em ambiente de virtualização 3D (Gazebo). Testes experimentais preliminares são incluídos para ilustrarem as possibilidades de aplicação das metodologias de controle propostas em um ambiente real. Conclusões a respeito do trabalho são apresentadas, desenvolvendo uma discussão sobre os seus pontos mais relevantes, bem como sobre as perspectivas de melhorias futuras e pontos que ainda podem ser melhor pesquisados.

## Palavras-chave

Robótica Móvel; Robótica Agrícola; Controle Não-Linear; Controle Robusto;

# Table of contents

# List of figures

# List of tables

# List of Abreviations

$m$ – meters  
$rad$ – radians  
$m\,s^{-1}$ – meters per seconds  
$rad\,s^{-1}$ – radians per seconds  
SMC – Sliding Mode Control  
STA – Super Twisting Algorithm  
RGB – Red-Green-Blue color space  
RoI – Region of Interest  
ROS – Robot Operating System  
CAS – Colision Avoidance System  
GPS – Global Position System  
RTK – Real-Time Kinematic  
DC – Direct Current  
IMU – Inertial Measure Unity  
MCE – Embedded Control Module (from Portuguese Módulo de Controle Embarcado)  
RC – Remotely Controlled  
PD – Proportional-Derivate (control)

# 1
# Introduction

In the last decades, robotic systems have been included in the daily activities of people in a increasingly fast and frequent manner. Recent technological advances in communication, actuation, sensory and computer systems have also allowed robots to work safely and efficiently close to human operators. Besides the well-known automotive industries and various industrial applications, it is possible to see the use of different types of robots in a variety of application areas such as:

- *Space:* helping in manipulation, assembling and maintenance tasks in orbit as assistants to astronauts, or performing exploration on remote planets as substitutes for human explorers [1];
- *Construction:* performing offsite prefabrication tasks or a single task at the construction site or integrating a robotized construction site in which multiple robots/machines collaborate to build an entire structure [2];
- *Hazardous Environments:* using teleoperated robots for explosive ordnance disposal, underwater engineering work, fire fighting, search and rescue operations, removing high-level nuclear contamination, reactor decommissioning, and tunneling through rock falls [3].
- *Mining:* carrying out autonomous tasks such as excavation and haulage, mapping and surveying for gathering geological, mineralogical and spatial information, as well as drilling and explosives handling, gases and other material detection [4].
- *Agriculture and Forestry:* automatic vehicle guidance, automatic sensing, handling and processing of crop production, automation of bush trimming, autonomous harvesting, color sorting and grading of crop production, substantial instrumentation and mechanization of livestock procedures, such as automated milking parlours and automated sheep-shearing [5].

Other relevant robotics applications which are growing fast are: medical and computer-integrated surgery [6], rehabilitation and health-care [7], domestic and education [8], and competitions [9, 10].

## 1.1 Motivation

Nowadays, livestock and agriculture are important sectors of the economy, which demand technologies from several areas such as sensor integration, embedded systems, image processing, and communication systems [11, 12]. From the first tools used by the primitive man to the latest agricultural equipment and machinery of the present days, technological advances have contributed to the increase in food production and, in the near future, also to decrease the water consumption, fertilizers and pesticides, reducing the environmental impacts [13].

Some examples are the development of transgenic grains, which are resistant to pests and with higher productivity than the conventional ones [14], as well as fully autonomous tractors, capable of performing tasks predefined by their users, such as planting, spraying and harvesting [15]. Besides the use in large agricultural tasks, robots have also been considered for precision agriculture activities that require small equipment and tools [13]. In this context, agricultural tasks such as crop monitoring, pest control, and plant phenotyping, which require the use of proper methods for soil sample collection and image acquisition can be carried out by using small robots, that may be autonomous or remotely controlled, terrestrial or aerial [16, 17]. In this way, several prototypes of mobile robots have been developed to perform a wide range of repetitive tasks, considering numerous challenges commonly found in the fields [18].

In general, agricultural robots or AgBots usually have intelligent and innovative solutions, based on sensor networks, autonomous or semi-autonomous navigation systems for obstacles detection and avoidance [19], modular mechanical structures for kinematic reconfiguration *in-situ* [20, 21], energy management systems as well as advanced control systems. However, most of these mechanisms are developed in an *ad-hoc* manner to perform specific tasks, having high costs for construction, maintenance and logistics. As an example of a low-cost solution, we can mention the test platform for research and experimentation composed of multiple cooperative vehicles based on the TXT-1 Monster Truck, developed by Tamiya Inc., presented in Cruz *et al.* [22] for outdoor surveillance tasks.

## 1.2 Review of The State of The Art

The agricultural sector has demanded innovative technologies that allow the increase of production, such as autonomous machines, genetically modified seeds, sensors for environmental monitoring, among others. As a result, lower production costs and reduction of environmental impacts have become relevant topics for research.

In cotton and soybeans crops, for instance, most of the production costs are related to pest, weed and disease controls, which are based on visual monitoring with workers previously trained as well as the use of pesticides and herbicides [23]. According to agriculture consultancy companies in Brazil, the expenditure on pesticides and herbicides in the states of Mato Grosso and Bahia, the largest Brazilian cotton producers, can reach up to 34% of the total production cost [24].

In this context, precision agricultural tasks such as seeding, pruning, weeding, harvesting, and phenotyping have been carried out by using several robots, terrestrial or aerial, with different types of locomotion systems, sensor packages and specialized tools [25]. Reduction of inputs, lower operating costs, less waste of water, increasing yields, and sustainable production are some of the benefits achieved by using robotic systems in agricultural fields [26, 27]. In this way, new techniques for crop maintenance and sustainability are being evaluated such as ultraviolet radiation for inactivation of microorganisms [28], laser-based weed control, biological pest control and autonomous mobile robots for visual monitoring of pests [29].

Taking into account that most of the field tasks are tedious and repetitive, the use of autonomous robotic systems becomes as solution to reduce the manpower usage, increase the productivity and optimize the agricultural processes. These mechanisms rely on different navigation systems based on advanced control strategies, planning algorithms and artificial intelligence, to perform monitoring and inspection missions in a safe and efficient manner [18, 30].

The latest developments of the Thorvald (Fig. 1.1 (a)) platform show us how robots can be used for transportation, UV treatment (Fig. 1.1 (b)), phenotyping, grass cutting, soil sampling, precision weeding, soft fruit picking, and controlling mildew with light treatment [31]. In this context, mobile robots with hybrid weeding mechanisms are being manufactured and endowed with vision-based capabilities for online weed detection and classification, allowing for specialized treatment of individual species [32].

It is well-know that crop monitoring, spraying and inspection tasks can be

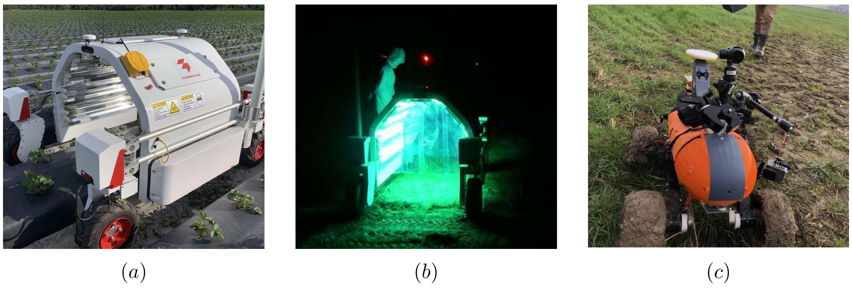

(a) (b) (c)

Figure 1.1: (a) Saga Robotics Thorvald; (b) UV treatment performed by Thorvald; (c) Smart Robot Company Tom with several embedded sensors.

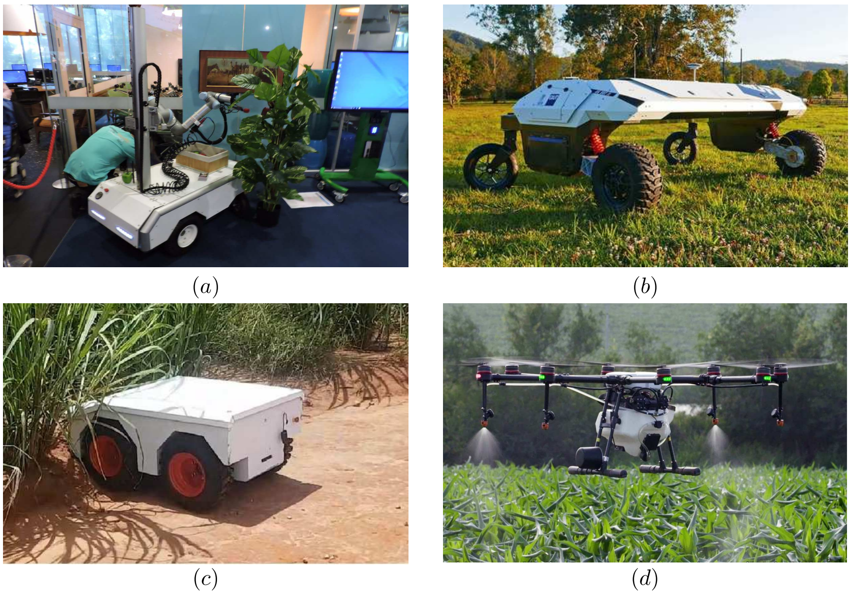

(a) (b) (c) (d)

Figure 1.2: (a) QUT Harvey; (b) QUT AgBot II; (c) TIBA robot; (d) DJI Agras.

carried out by using unmanned aerial vehicles (UAVs) (Fig. 1.2 (a)) with high-resolution sensors and high-performance computing. However, UAVs suffer from low flight autonomy, low loading capacity, high operating costs and are crash-prone, making their use inappropriate in dense vegetation fields [33].

The aforementioned disadvantages suggest the use of wheeled mobile robots as an alternative solution for smart farming, since these are capable of navigating intelligently through the crop rows with high autonomy and carrying a high number of different sensors, as shown in Fig. 1.2 (b) [34, 35]. Moreover, it is possible to mount a robotic manipulator on the robot structure, endowed with an attached camera and specialized tools to visualize the plants from different points of views and collect samples respectively (Fig. 1.2 (c)) [36].

A picking robot that harvests strawberries on tabletop cultivation systems was developed in [37] and mounted on a mobile platform that navigates autonomously through the greenhouse. The mechanical design, systems integration, and field evaluation of a robotic apple harvester based on the combination of sensing, planning, and manipulation capabilities was presented in [38] resulting in low-cost modern orchard system.

Autonomous navigation of crop rows is typically accomplished by using cameras and global positioning system (GPS) units (Fig. 1.1 (c)). Machine vision strategies can also be implemented to detect crop row contours and edges in order to ensure a proper navigation of rows without damaging crops [39]. An automatic guidance system for a four–wheel–steering mobile robot is developed in [34] to perform accurate field operations based on the full compensation of sliding effects during path tracking. In [27], an adaptive steering system based on a model reference adaptive control (MRAC) is designed to control the lateral position of a farm tractor with varying yaw rate properties in order to track a straight path. A skid steering mobile robot (Fig. 1.2 (d)) was developed in [40] to infiltrate into tunnels of dense vegetation that occur in sugarcane plantations, performing mapping and collecting samples, using thermal imaging for autonomous navigation, as well as performing crop mapping and collecting samples.

## 1.3 Methodology

In this work, we address the problem of controlling a class of wheeled mobile robots – differential-drive and car-like robots – capable of performing autonomous navigation missions in agricultural fields. The methodology will be split in three parts: modeling, control design, verification an validation.

### 1.3.1 Kinematic Modeling

Our solution is based on a kinematic approach, combining robust control and computer vision techniques to allow a safe and efficient operation of the mobile robot in row crops. The kinematics-based solution is commonly used for practical applications where the robot dynamics may be neglected when vehicle motions require low speeds and slow accelerations to carry out a given task [41].

In the proposed monitoring and inspection task, the robot needs to move along the crop rows to collect samples (e.g., images) in predefined points (way points). Due to the well-known characteristics of the agricultural fields, such as terrain slope, variety of landform, drainage pattern, weathering and soil type, wheeled mobile robots are not able to perform high speed maneuvers with safety and efficiency, which match with the proposed kinematics-based solution.

### 1.3.2 Control Design

The differential-drive mobile robots can be modeled as an unicycle vehicle [42] and the motion control strategy is based on the combination of a visual servoing approach and an image segmentation method which takes into account the uncertainties commonly found in open fields, such as changes in light conditions [43–45]. A robust control strategy will be also implemented by using the Sliding Mode Control approach due its well-known ability to deal with parametric uncertainties and external disturbances caused by slipping/skidding on the terrain, which implies in loss of accuracy for position estimation based on dead reckoning approach [46].

To overcome these drawbacks, robust control techniques will be investigated and evaluated in simulation environments (e.g., Matlab, Gazebo). To perform autonomous navigation through the crop field, we propose the use of an image-based visual servoing (IBVS) approach [41, 47] wherein the image feature extraction is based on a robust segmentation method [43, 44].

The car-like mobile robot can be modeled as the bicycle vehicle and it is mechanically adapted with the purpose of performing a simple data collection in small size crops due its small dimension. For such a vehicle, we propose a motion control strategy which relies on its kinematic similarity with the unicycle vehicle, to ensure a satisfactory performance for a given stabilization task.

### 1.3.3 Verification and Validation

Preliminary results obtained from experimental tests in a real-world scenario using both mobile robots are presented in order to verify and validate the proposed methodology. The experimental tests are carried out with the differential-drive robot performing monitoring and inspections tasks on a cotton farm, wherein the vehicle must be able to navigate autonomously in the plantation using the crop rows as a reference path. These graphical results are presented in Sec. 3. For the car-like robot, practical assays are performed in the parking area of the PUC-Rio, wherein the vehicle must be able to move toward some waypoints for collecting data. The graphical results of such practical assays are presented in Sec. 4 followed by a brief discussion.

## 1.4 Goals and Objectives

The objectives of this Master's thesis is to develop an efficient and feasible autonomous navigation approach, enabling two classes of wheeled mobile robots to perform monitoring and inspection tasks in agricultural fields. We also seek to carry out experimental tests in order to verify and validate the proposed modeling and control methodology.

As short-term goals, we expect to properly design the electro-mechanical system and integrate several sensory systems in order to better understand the challenges and research opportunities offered by the agricultural environment. We also aim to submit at least one paper for a highly-ranked international conference in the robotics area. As long-term goals, we expect to provide enough knowledge and tools so that other researchers from the group may continue the development of some research topics presented here such as, autonomous navigation, visual servoing, robust control and swarm robotics. We also intend to carry out experimental tests in a real agricultural environment with the mobile robots of our lab, named SoyBot, iCat and TIBA, performing an autonomous navigation task by using a swarm-based control strategy.

## 1.5 Contributions

The main contribution of this Master's thesis is to investigate the combination of different modeling methodologies and control strategies to allow for the safe and efficient operation of two classes of wheeled mobile robots in agricultural fields. Other relevant contribution is to design and develop a 3D simulation environment using Gazebo and to implement a ROS package for autonomous robot navigation in agricultural fields. During the execution of this Master's thesis the following manuscripts were published:

- Oliveira, A. I. S., Carvalho, T. M., Martins, F. F., Leite, A. C., Figueiredo, K. T., Vellasco, M. M. B. R. & Caarls, W., "On the Intelligent Control Design of an Agricultural Mobile Robot for Cotton Crop Monitoring", Proceedings of the 12th International Conference Developments in e-Systems Engineering: Robotics, Sensors and Industry 4.0, Kazan, Russia, 2019.

- Barbosa, W. S., Oliveira, A. I. S., Barbosa, G. P. B., Leite, A. C., Figueiredo, K. T., Vellasco, M. M. B. R. & Caarls, W., "Design and Development of an Autonomous Mobile Robot for Inspection of Soy and Cotton Crops", Proceedings of the 12th International Conference Developments in e-Systems Engineering: Robotics, Sensors and Industry 4.0, Kazan, Russia, 2019.

- Oliveira, A. I. S., Prado, M. G. & Leite, A. C., “Adaptação de um Automodelo para Aplicações de Robótica Móvel na Agricultura”, Anais do XXII Congresso Brasileiro de Automática, João Pessoa (PB), Brasil, 2018.

Most part of the material presented in this Master’s thesis is based on the aforementioned manuscripts.

## 1.6 Organization of the Thesis

This Master’s thesis is organized as follows:

- **Chapter 2** introduces the kinematics modeling and basic control design, based on Cartesian and polar coordinates, for the differential-drive vehicle. Numerical simulations with Matlab are include to illustrate the performance and feasibility of the proposed modeling and control methodology. We also present the design, development and constructive aspects of the SoyBot robot, developed to carry out monitoring and inspection tasks in agricultural fields.
- **Chapter 3** presents the visual servoing approach used for vision-based autonomous navigation. We also introduces the input-output linearization method used to simplify the control design, as well as numerical simulations with the Sliding Mode Control (SMC) approach. Verification and validation of the proposed visual servoing approach are carried out through experimental tests with the SoyBot robot in a cotton crop.
- **Chapter 4** presents the modeling and control design for a car-like mobile robot, named iCat, able to collect samples beneath of the leaves in soy and cotton crops. A kinematic control approach, based on Cartesian coordinates, is used to allow the iCat robot to move toward waypoints in the agricultural field. Numerical simulations in Matlab and 3D visualization in Rviz are included to show the performance and efficiency of the proposed methodology.
- **Chapter 5** presents the final considerations, including conclusions remarks and perspectives for future work.

# 2
# Design and Development of a Differential-Drive Mobile Robot

## 2.1
## Introduction

In this chapter, we present the design and development of an autonomous mobile robot, conceived to perform routine monitoring and inspection tasks on soy and cotton crops. The robot design is similar to a differential-drive vehicle due to its simplicity of construction, modeling and control, taking in account that the differential-drive is the kinematic equivalent of the unicycle-like mobile robot. Results of numerical simulations will be presented to evaluate the proposed methodology and analyzes the behaviour of the system.

## 2.2
## Kinematic Model

The robot configuration $q \in \mathbb{R}^3$ can be characterized by the Cartesian position $(x, y) \in \mathbb{R}^2$ of such contact point with respect to a fixed reference frame $\mathcal{F}_w$ and the orientation angle $\theta \in \mathbb{R}$ with respect to its $x$-axis (Fig. 2.1). Thus, the configuration vector $q$ that defines the robot position and orientation is expressed by

$$q = \begin{bmatrix} x & y & \theta \end{bmatrix}^{\mathrm{T}}. \tag{2-1}$$

According to [41], a unicycle-like is a vehicle with a single orientable wheel, which its configuration can be described by (2-1) and the reference point is the contract point between the ground and the wheel. This wheel is a standard steerable wheel and its pure rolling constraint is expressed in Pfaffian form as

$$\mathbf{a}^{\mathrm{T}}(\mathbf{q})\,\dot{\mathbf{q}} \;=\; \begin{bmatrix} \sin\theta & \cos\theta & 0 \end{bmatrix} \dot{\mathbf{q}} = 0\,, \tag{2-2}$$

entailing that the velocity in the contact point is zero in the direction orthogonal to the sagittal axis of the vehicle, so called the *zero motion line.*

Consider a base of $\mathcal{N}(\mathbf{a}^{\mathrm{T}}(\mathbf{q}))$, formed by all admissible trajectories for the proposed constraint. The dynamic system to defines the kinematic model of the unicycle can be expressed as

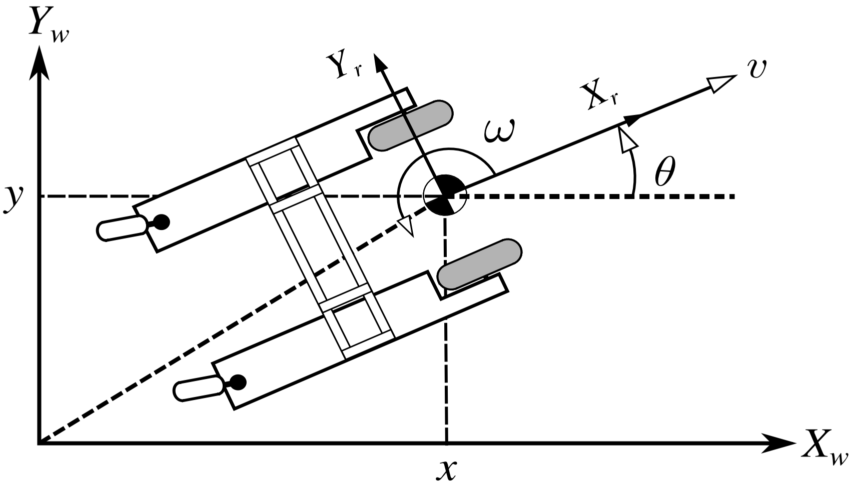


Figure 2.1: Global and robot reference frames.

$$\dot{\mathbf{q}} = \sum_{j=1}^{m} \mathbf{g}_j(\mathbf{q})u_i = G(\mathbf{q})\mathbf{u} \qquad m = n - k, \tag{2-3}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the state vector (2-1), $k$ is the number of Pfaffian constraints and $\mathbf{u} \in \mathbb{R}^m$ is the input vector. In this way, we can define the matrix

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{q}) & \mathbf{g}_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \tag{2-4}$$

which $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$ are in the null space of null space of $\mathbf{a}(\mathbf{q})$ in 2-2. Thus, the kinematic model of the unicycle is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{2-5}$$

where the physical interpretation for $v$ and $\omega$ are respectively the linear or drive velocity of the vehicle and its steering or angular velocity around the vertical axis.

The reference point in the differential-drive robot is the midpoint of the segment joining the two wheels and the linear and angular velocities can be expressed in function of them angular speed $\dot{\phi}_R$ and $\dot{\phi}_L$ as follow:

$$v = \frac{r(\dot{\phi}_R + \dot{\phi}_L)}{2} \qquad\qquad \omega = \frac{r(\dot{\phi}_R - \dot{\phi}_L)}{d}, \tag{2-6}$$

where $r > 0$ is the radius of the wheels, $d > 0$ is the distance between their centers and $\dot{\phi}_R$ and $\dot{\phi}_L$ are the angular velocities of right and left wheels, respectively.

## 2.3
## Motion control

According to [48], despite the variety of application for mobile robots, three generic control problems can mainly considered: path following, stabilization of trajectories and stabilization of fixed postures.

The first task consists of, given a curve $\mathcal{C}$ on the plane and a fixed longitudinal velocity $v_0$, the robot needs to follow the curve, stabilizing at zero its distance to the curve. The second task differs from the previews one in that the longitudinal velocity in not fixed at $v_0$. The geometric curve $\mathcal{C}$ is complemented with a time schedule, parameterized with respect to the time. The goal is to stabilize the position error vector $(x(t) - x_r(t)\,, y(t) - y_r(t))$ at zero, where $(x_r(t)\ ,y_r(t))$ can be formulated as a reference vehicle to be tracked. In the third class of task, the goal is to stabilize at zero the posture vector (or pose) with respect to a reference pose. This kind of task can still be classified as partial Cartesian regulation, when the final orientation in not specified, and posture regulation, where the whole configuration vector are take in account.

The next section presents the proposed solutions to these issues for differential-drive robot. To evaluate the proposed approaches, numerical simulation is presented.

### 2.3.1
### Cartesian Control for Partial Regulation Task

The partial regulation task have a practical interest. It can be used for exploration of an unknown environment, visiting a sequence of Cartesian coordinates (view points), using its on-board sensors. For the case where the sensors ares isotropically distributed on the robot, such rotating lasers or ultrasonic sensors, the orientation is irrelevant [41].

In this section, we consider that the control goal is reach a desired position defined on the Cartesian space, $(x_d, y_d) \in \mathbb{R}^2$, from the current position of the robot $(x_0, y_0) \in \mathbb{R}^2$. The partial regulation problem consist to regulate the robot from its initial position $q$ to a desired position, without considerate the final robot orientation. In this context, the control goal can be defined as:

$$x_d \rightarrow x_0\,, \qquad e_x = x_d - x_0 \rightarrow 0, \tag{2-7}$$

$$y_d \rightarrow y_0\,, \qquad e_y = y_d - y_0 \rightarrow 0, \tag{2-8}$$

where $e_x = x_d - x_0$ and $e_y = y_d - y_0$. The desired orientation angle with respect to axis $x$ given by $\theta_d = \tan^{-1}(e_y, e_x)$. Thus, we can state the follow theorem:

**Theorem 2.1** *Consider the kinematic model of the mobile robot (2-5) and assume that the desired coordinates* $(x_d\,,\,y_d)$ *and* $\theta_d = \tan^{-1}(e_y, e_x)$ *are bounded. Then, the following control laws*

$$v = -k_1[e_x \cos(\theta) + e_y \sin(\theta)], \tag{2-9}$$

$$\omega = k_2[\theta_d - \theta + \pi], \tag{2-10}$$

*where* $k_1 > 0$ *and* $k_2 > 0$ *are the proportional gains, ensure the convergence of the position error* $e_p = [e_x \; e_y]^\mathsf{T}$ *to zero.*

*Proof.* Consider the following Lyapunov-like function candidate

$$2V(e_x\,,e_y) = e_x^2 + e_y^2\,, \tag{2-11}$$

which is an energy function, positive semidefinite at origin for any value of $\theta$. Then, taking its time-derivative along the trajectories of the error system (2-7), given by $\dot{e}_x = \dot{x}_d - \dot{x}_0$ and $\dot{e}_y = \dot{y}_d - \dot{y}_0$ and using the control law (2-9) we obtain $\dot{V}(x, y) = -k_1\,[\,e_x \cos(\theta) + e_y \sin(\theta)\,]^2$ which is negative semi definite. Because $V$ is a positive definite function with non-positive derivative, we have that $V \in \mathcal{L}_\infty$, implying that $e_x\,, e_y \in \mathcal{L}_\infty$ and, therefore, $e_p \in \mathcal{L}_\infty$. In this way, the equilibrium state $e_p$ is uniformly stable. Once $V > 0$ e $\dot{V} \leq 0$ we have that $\lim_{t\to\infty} V(x(t), y(t)) = V_\infty \geq 0$ exist. From $\dot{V}$ we have that $\int_0^\infty \dot{V}(\tau)d\tau = V_0 - V_\infty \geq 0\,$, where $V_0 = V(x(0), y(0))$ implying $x, y \in \mathcal{L}_2$ and, therefore, $e_p \in \mathcal{L}_2$ $\theta \in \mathcal{L}_\infty$ and $e_x\,, e_y \in \mathcal{L}_\infty$ implies that $v, \omega \in \mathcal{L}_\infty$ and, consequently, concludes that $\dot{x}, \dot{y}, \dot{e}_p \in \mathcal{L}_\infty$. Therefore, using the Barbalat's Lemma [41] and knowing that $e_p \in \mathcal{L}_2$ e $\dot{e}_p \in \mathcal{L}_\infty$, implies that $\lim_{t\to\infty} e_p(t) = 0$. ■

To evaluate the proposed approach, the follow numerical simulations was proposed, where the vehicle should perform a displacement along a set of waypoints in the plane $(x\,,y)$ according to Tab. 2.1. As cited bellow, the final orientation was not considerate for this case.

Table 2.1: List of Waypoints

| ***Start*** $(x_0,y_0,\theta_0)$ | ***Stop*** $(x,y,\theta)$ |
|---|---|
| (0,10,0) | (10,10,0) |
| (10,10,0) | $(10,-10,\pi)$ |
| $(10,-10,\pi)$ | $(-10,-10,\pi)$ |
| $(-10,-10,\pi)$ | $(-10,10,0)$ |
| $(-10,10,0)$ | (0,10,0) |

As can be seen in Figs. 2.2 and 2.3, all the waypoints was reached using the proposed controller, with regulation error in $(x\,,y)$ tending to zero, as shown in Fig. 2.4. As expected, the control signal for linear velocity $v$ tends to zero as the robot is close to desired coordinate. For numerical simulation purpose the values of linear and angular control signals were left as computed. In order to prevent damage to motors and driver system, for real application this values need to be limited by the use of saturation therms and reduced gains. This same idea can be extend for all following control approaches. For some cases, a signal inversion can happen because of the use of sinusoidal functions in the controller, as can be seen in Fig.2.5, between $40\,s$ and $50\,s$ of simulation. The angular control signal $\omega$ becomes zero when the robot is pointing to the destination, as can be seen in Fig. 2.5.

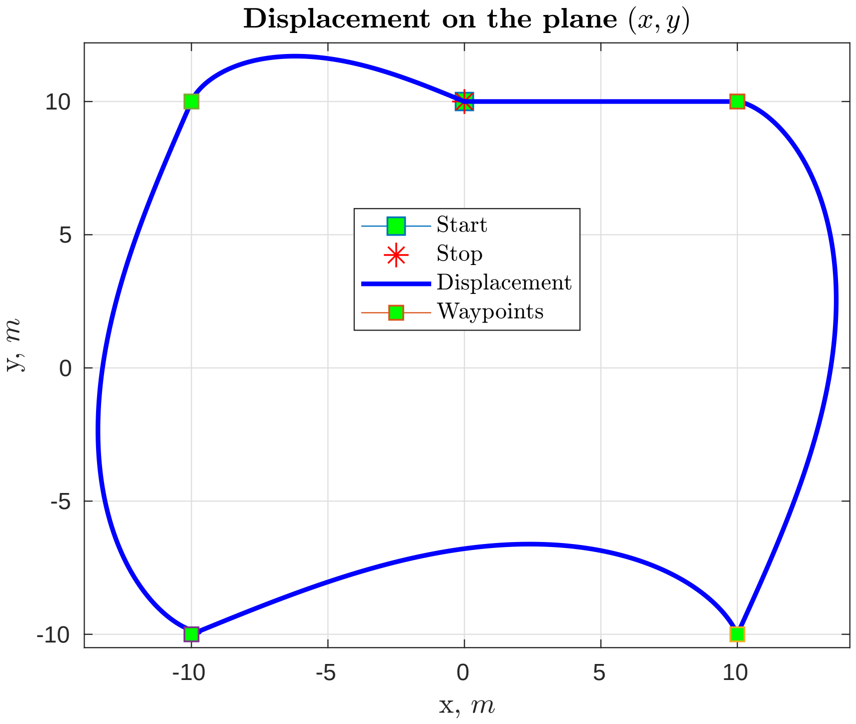


Figure 2.2: Robot displacement (in meters) on the plane ($x$ ,$y$).

### 2.3.2 Polar Coordinates Control for Posture Regulation

In the last section, we presented a kinematic control approach based on Cartesian coordinates which is able to ensure only the position regulation, without considering the final orientation. Indeed, in the numerical simulations it is possible to observe that the robot orientation is not properly controlled. This drawback can be overcome by using a convenient representation in polar coordinates [41, 42].

We can represent the robot kinematics, with respect to the reference frame, by using three coordinates as shown in Fig.(2.6): $\rho$ is the distance

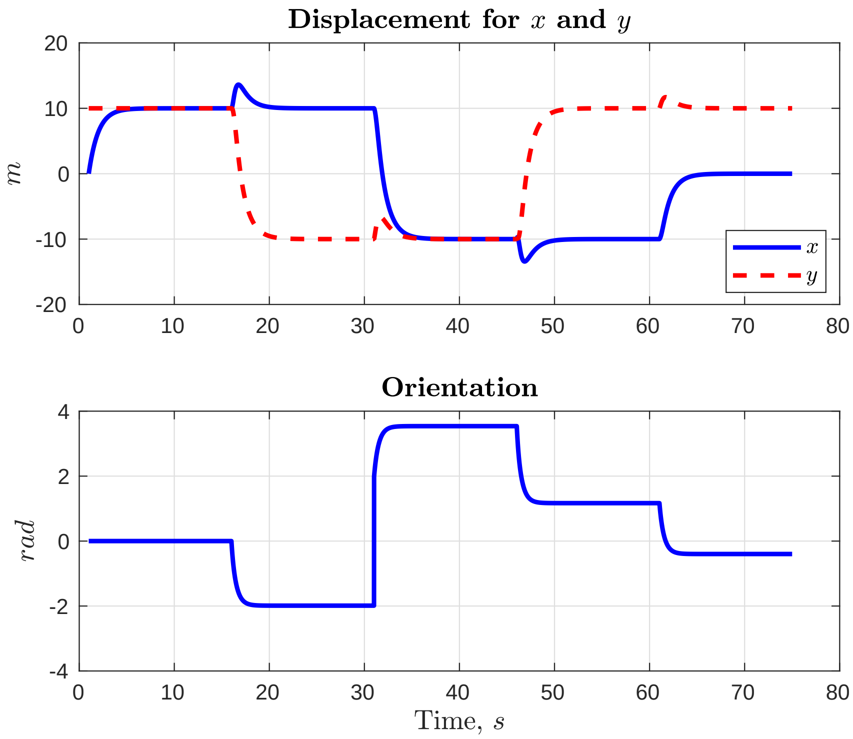


Figure 2.3: Behavior over time for state variables $x$, $y$ (meters by seconds) and $\theta$ (radian by seconds).

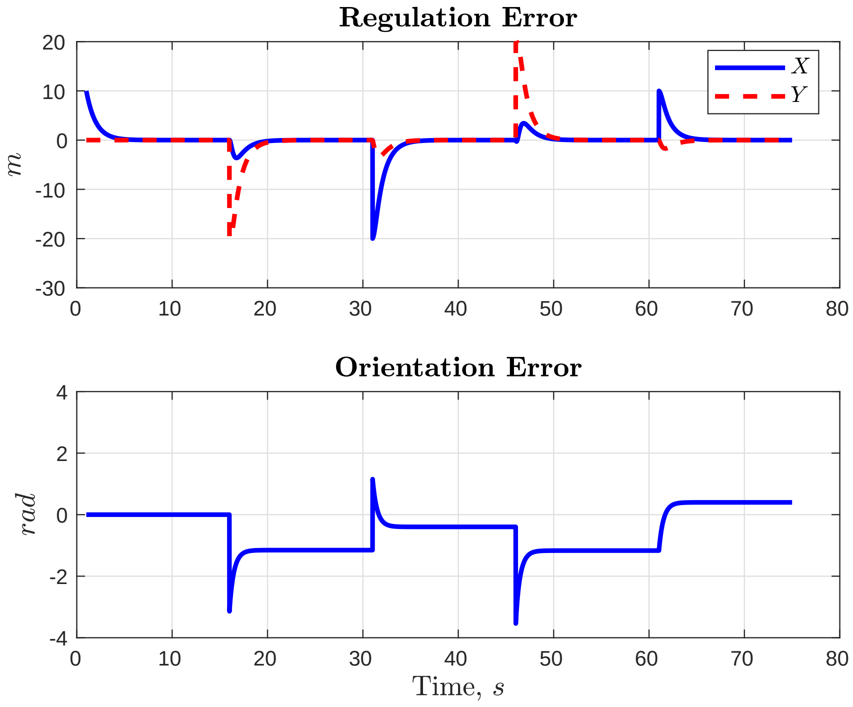


Figure 2.4: Position and angular error.

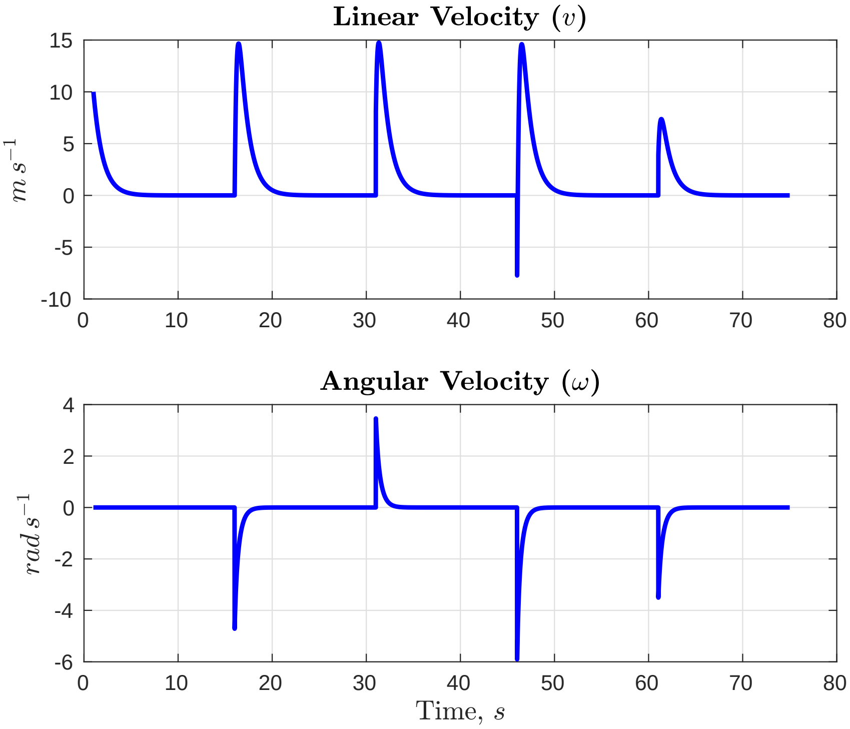


Figure 2.5: Control signal for $v$ and $\omega$.

between the origins of the robot frame $\mathcal{F}_R$ (the robot position) and goal frame $\mathcal{F}_G$ (the goal position). In general, the robot frame $\mathcal{F}_R$ is attached to the center of the axle of the wheels; $\alpha$ is the angle between the $x$-axis of the robot frame $\mathcal{F}_R$ and the vector $\hat{x}$ defined along the distance connecting the robot position to the goal position; $\beta$ is the angle between the the vector $\hat{x}$ and the $x$-axis of the goal frame $\mathcal{F}_G$.

From Fig. 2.6, if $\alpha \in \mathcal{I}$ where $\mathcal{I} \in (-\pi/2, \pi/2\,]$, the following coordinate transformation can be used [42]:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}\,, \tag{2-12}$$

$$\alpha = -\theta + \mathrm{atan2}(\Delta y, \Delta x)\,, \tag{2-13}$$

$$\beta = \theta + \alpha\,, \tag{2-14}$$

and the robot kinematic model expressed in this new coordinates is given by:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} -\rho\,\cos(\alpha) & 0 \\ \sin(\alpha) & -\rho \\ -\sin(\alpha) & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{2-15}$$

In the next sections, two control design approaches based on linear and nonlinear control laws are presented as well as corresponding numerical simulations to illustrate their performance and feasibility.

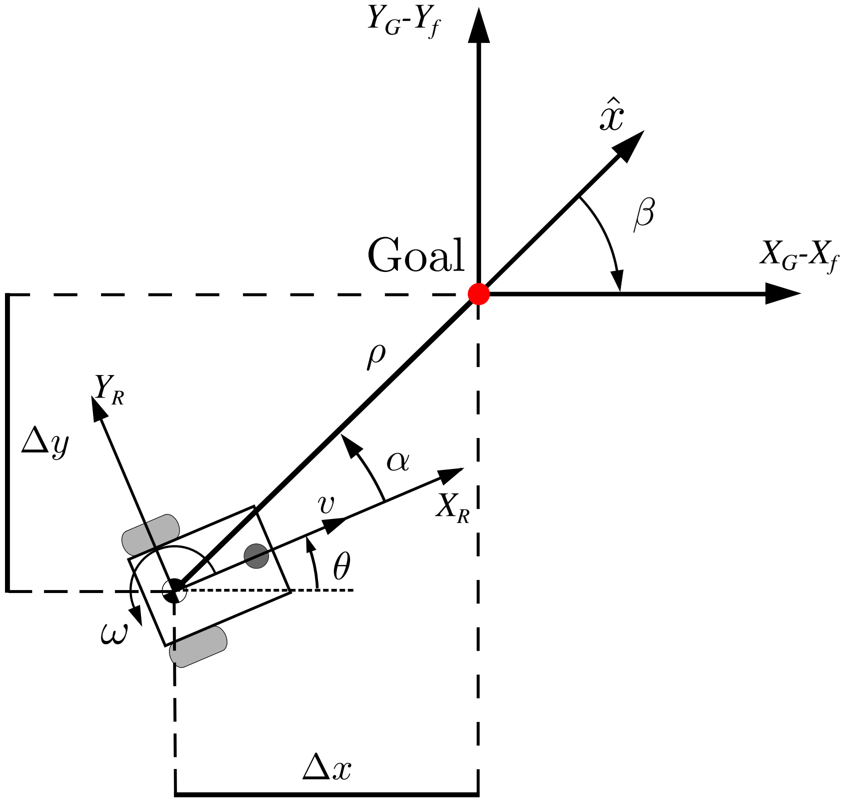


Figure 2.6: Polar coordinates representation for the differential-drive robot.

#### 2.3.2.1 Polar Coordinates with Non-linear Control

Consider, without loss of generality, that we want to stabilize the robot to the origins of the reference coordinate system. For this case, we can define the desired position as $\begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$. Based on the transformation (2-12), we can use the following control law ( [41]):

$$v = k_1 \rho \cos\alpha, \tag{2-16}$$

$$\omega = k_2 \alpha + k_1 \frac{\sin\alpha\cos\alpha}{\alpha}(\alpha + k_3\beta). \tag{2-17}$$

with $k_1 > 0$ and $k_2 > 0$. We can easily note that (2-16) is a proportional control law, as proposed in (2-9), where the linear velocity tends to zero as the robot is close to the destination. For the steering velocity, a second term for pointing error correction was added. The stability can be verified using the follow Lyapunov function:

$$V = \frac{1}{2}(\rho^2 + \alpha^2 + k_3\beta^2), \tag{2-18}$$

whose the close-loop system is given by

$$\begin{aligned} \dot{V} &= \dot{\rho}\rho + \dot{\alpha}\alpha + k_3\dot{\beta}\beta \\ \dot{V} &= -k_1\rho^2\cos^2\alpha - k_2\alpha^2 \ \leq\ 0\,, \qquad \forall\, k_1,\, k_2\ \geq\ 0 \end{aligned} \tag{2-19}$$

Because (2-19) is negative semidefinite at the origin, with $\dot{V}\ \leq\ 0$, it is possible to guarantee that the position error, defined by $\rho$ and $\alpha$, tends to zero as the time $t$ tends to infinite for any configuration. Therefore, $V$ tends to a limit value and system state is limited. In view of Barbalat lemma, it can be inferred that $\dot{V}$ tends to zero, likewise $\rho$ and $\alpha$ due to $\ddot{V}$ is bounded [41].

To evaluate this approach, numerical simulation was performed for initial conditions proposed in Tab. 2.1, with $k_\rho = 1$, $k_\alpha = 2.5$ and $k_\beta = 1$ as control gains [41].

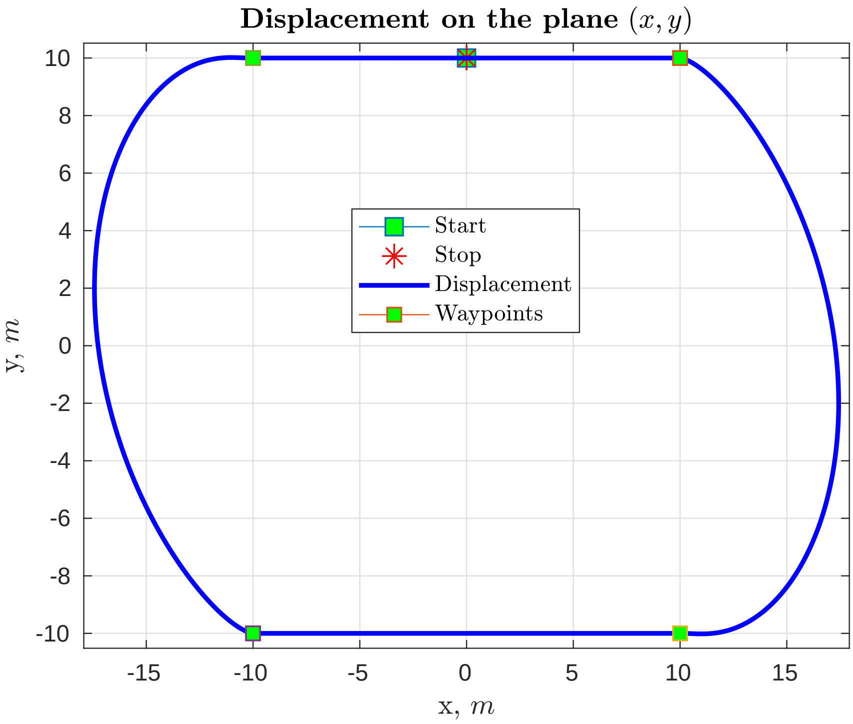


Figure 2.7: Robot displacement in the plane $(x\,, y)$.

As can be seen in Figs. 2.7 and 2.8, all the waypoints was reached, as in the Cartesian approach. However, because the polar coordinates controller allow the orientation control, the path traveled was more smooth that first case. This behavior is very useful in task such line follow or crop follow in field applications. At least, we can see that the state variables of polar system ($\alpha\,,\beta$ and $\rho$) tends to zero at the same time as the original system variables ($x\,,y$ and $\theta$) approaches to the desired coordinates, as can be seen in Fig. 2.11.

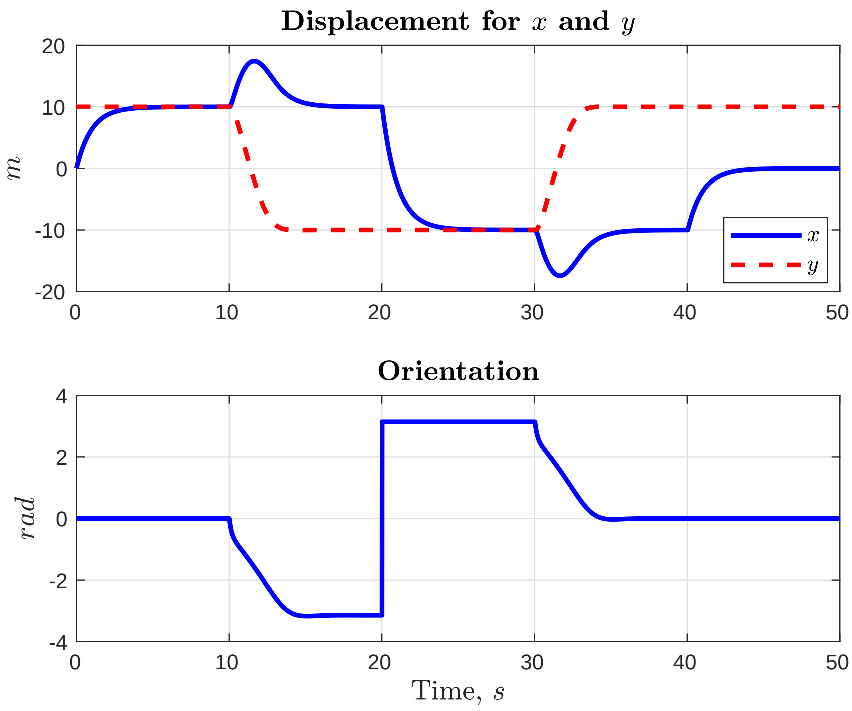


Figure 2.8: Behavior over time for state variables $x$, $y$ and $\theta$.

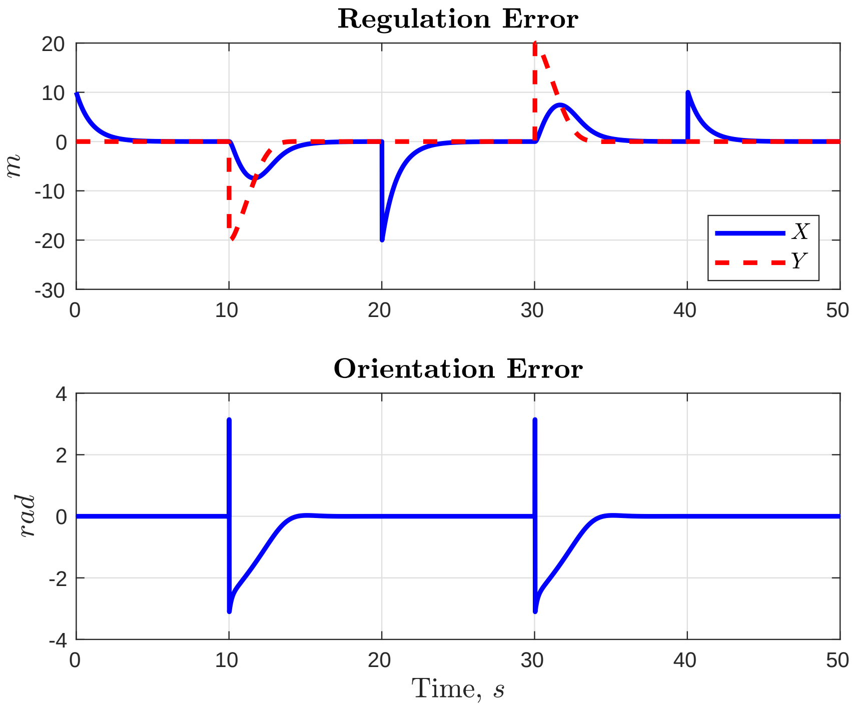


Figure 2.9: Position and angular error.

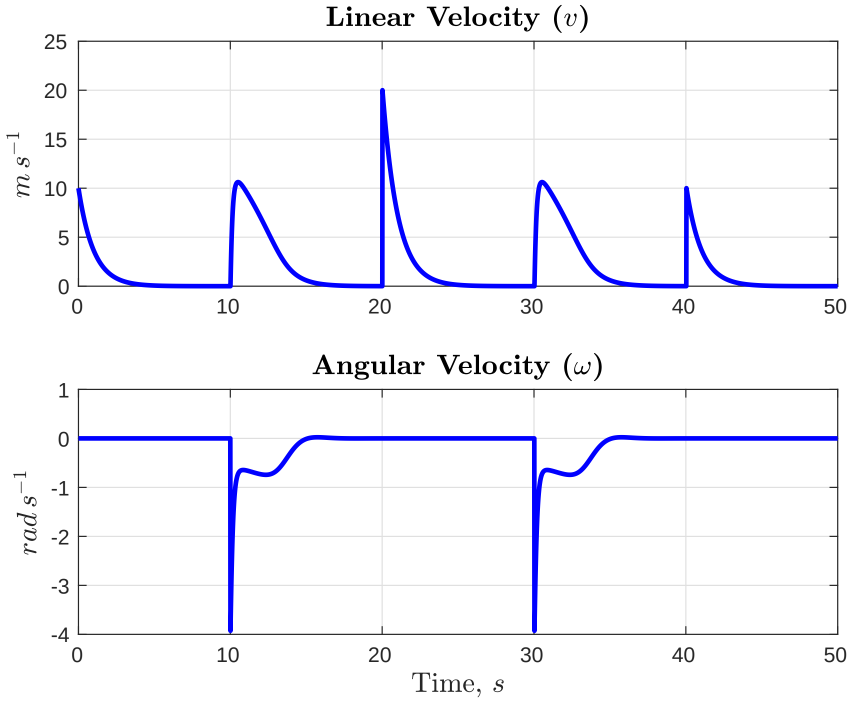


Figure 2.10: Control signal for $v$ and $\omega$.

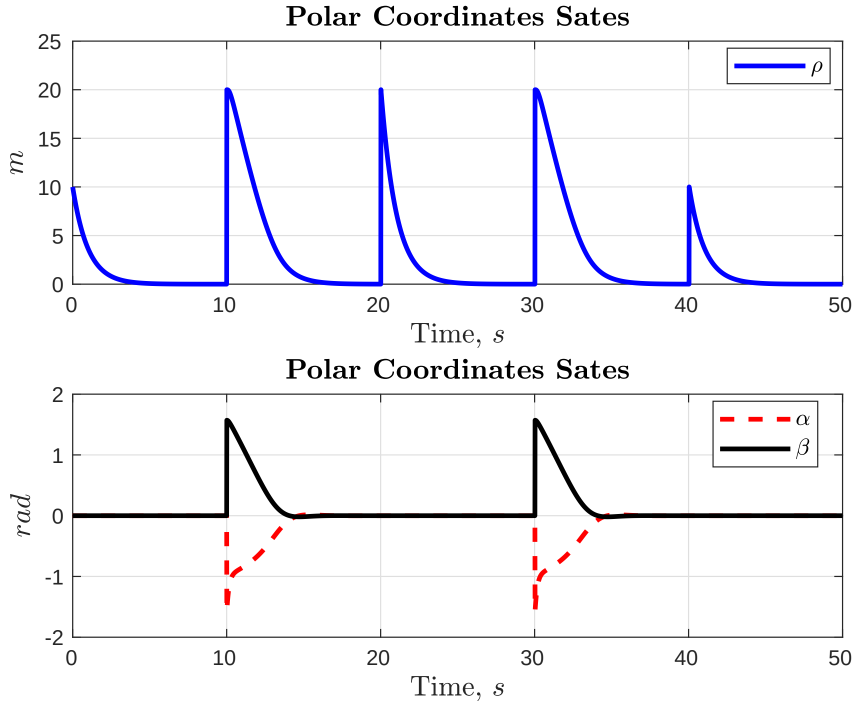


Figure 2.11: State variable for polar system.

### 2.3.2.2
### Polar Coordinates with Linearized Control

Consider, without loss of generality, that the goal coordinate is the system origin, as it can be seen in Fig. 2.6. Although the system present a singularity (or discontinuity) in $\rho = 0$ and the coordinate transformation is not defined at $\Delta x = \Delta y = 0$, the Brockett's theorem which ensures the smooth stabilization still holds. [42]. Consider the following linear control law

$$v = k_\rho \rho\,, \tag{2-20}$$

$$\omega = k_\alpha \alpha + k_\beta \beta\,, \tag{2-21}$$

applied to the control system (2-15). Notice that this control low is equivalent to the proposed in (2-16) and (2-17) for values of $\alpha$ near to zero, which implies in $\cos(\alpha) = 1$ and $\sin(\alpha) == \alpha$ Then, the closed-loop system can be expressed by

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \rho \sin\alpha \end{bmatrix}. \tag{2-22}$$

Defining $\alpha \in I_1 = (-\pi/2\,, \pi/2]$ and expressing $\alpha$ and $\beta$ in the range $(-\pi\,, \pi)$, this system does not have any singularity at $\rho = 0$ and has a unique equilibrium point at $\begin{bmatrix} \rho & \alpha & \beta \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$, driving the robot to this point. The sign of $v$ is always constant and positive due to the fact that $\alpha(0) \in \mathcal{I}$. Otherwise, it is always negative. This implies that the robot performs maneuvers always in a single direction, without reversing its motion. Notice that, the control law (2-21) is locally stable since $\alpha \in \mathcal{I}$ for all $t > 0$. Rewriting the system and linearizing around the origin ($\cos\phi = 1$ and $\sin\phi = \phi$), we have

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & k_\rho - k_\alpha & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}, \tag{2-23}$$

that can be written in matrix form as

$$\dot{\mathbf{z}} = A\,\mathbf{z}\,. \tag{2-24}$$

The condition for locally and exponentially stability is that the real part of all eigenvalues of matrix $A$ are negative. Thus, the characteristic polynomial of matrix $A$, denoted by $\Delta(\lambda)$ is given by

$$\Delta(\lambda) = (\lambda + k_\rho)[\,\lambda^2 + (k_\alpha - k_\rho)\lambda - k_\rho k_\beta\,] = 0\,,$$

and, according to Routh-Hurwitz stability criteria, for all roots have negative real part, implies that

$$k_\rho > 0\,, \qquad k_\alpha > k_\rho\,, \qquad k_\beta < 0\,. \tag{2-25}$$

The evaluation of the proposed approach can be seen in the results of numerical simulation, presented as follow. For this case, was used the values $k_\rho = 3$, $k_\alpha = 8$ and $k_\beta = -1.5$ and the values of Tab.2.1 as destination.

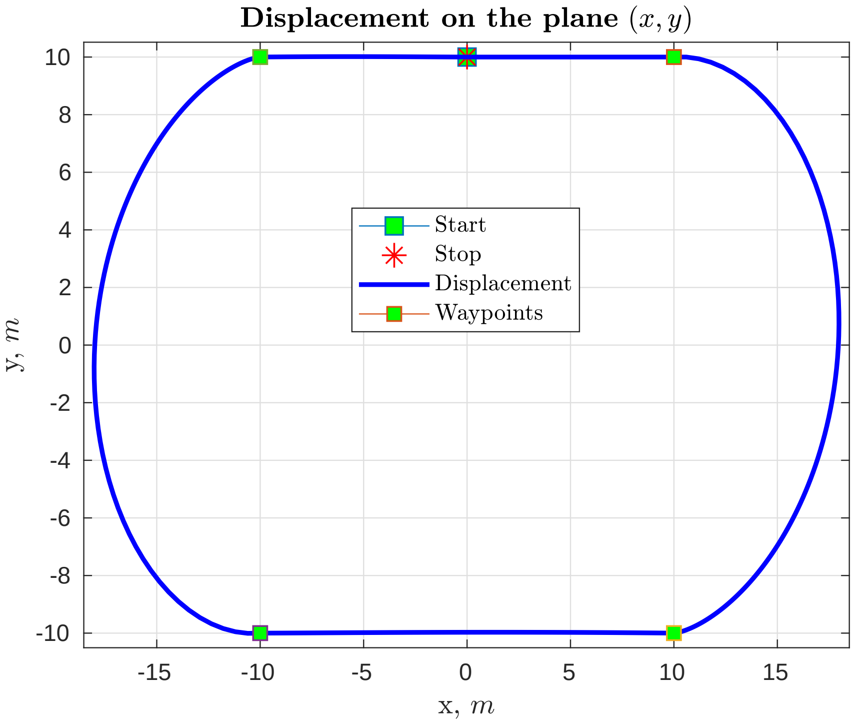


Figure 2.12: Robot displacement in the plane $(x\ ,y)$.

As long as the error in the polar state variables tends to zero (Fig. 2.16, the robot reach the desired pose $(x_d,\, y_d,\, \theta_d)$, as can seen in Figs. 2.12 and 2.13. and the pose regulation error tends to zero (Fig. 2.14).

## 2.4 Robot Modeling

The design and construction of a wheeled mobile robot for agriculture purposes require previous knowledge of the crop field characteristics (e.g., crop type and density, type of terrain, distance between rows, plant height and width within rows, crop stage, etc.) as well as the type of inspection (e.g., insects, weeds and diseases). The first specification to be met is to know the characteristics of the terrain such as its slope, variety of landform, drainage pattern, weathering and soil type.

The soil type, for instance, is used to define the power supply, the maximum motor torque and the tire tread. In this work, according to the customer specifications, the soil is basically composed of clay and sand (Fig. 2.17), in

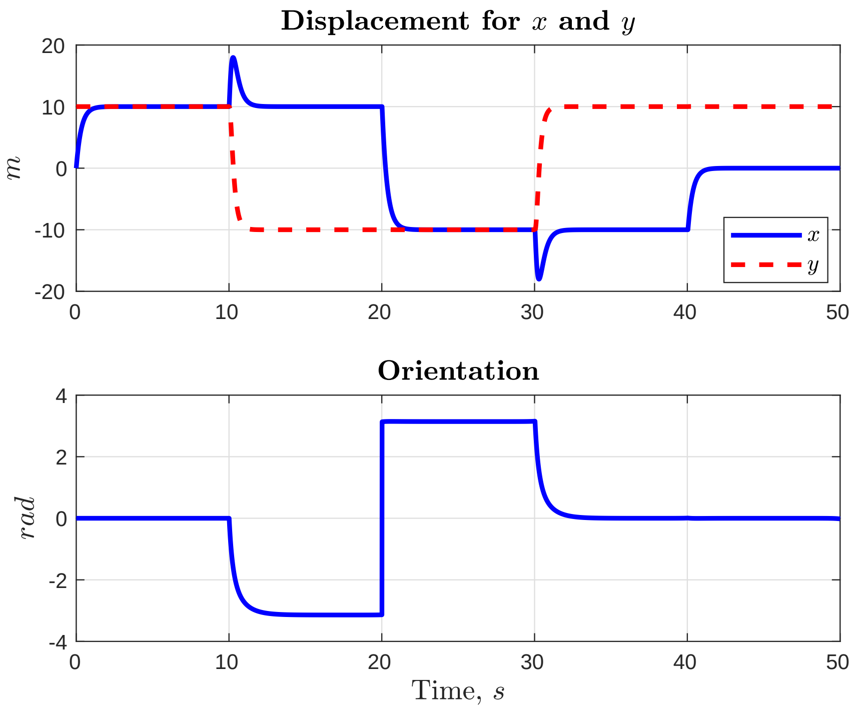


Figure 2.13: Behavior over time for state variables $x$, $y$ and $\theta$.

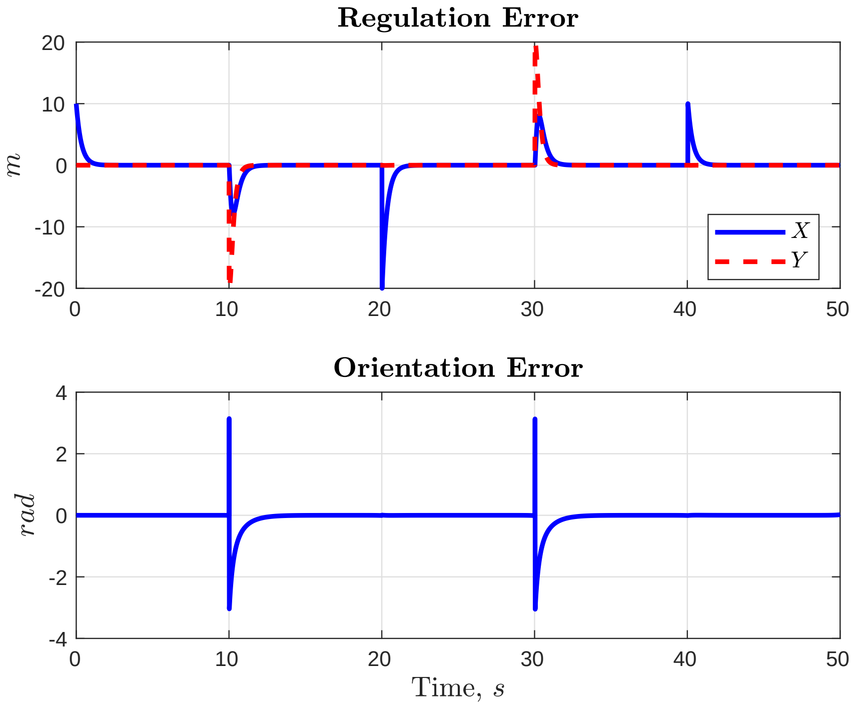


Figure 2.14: Position and angular error.

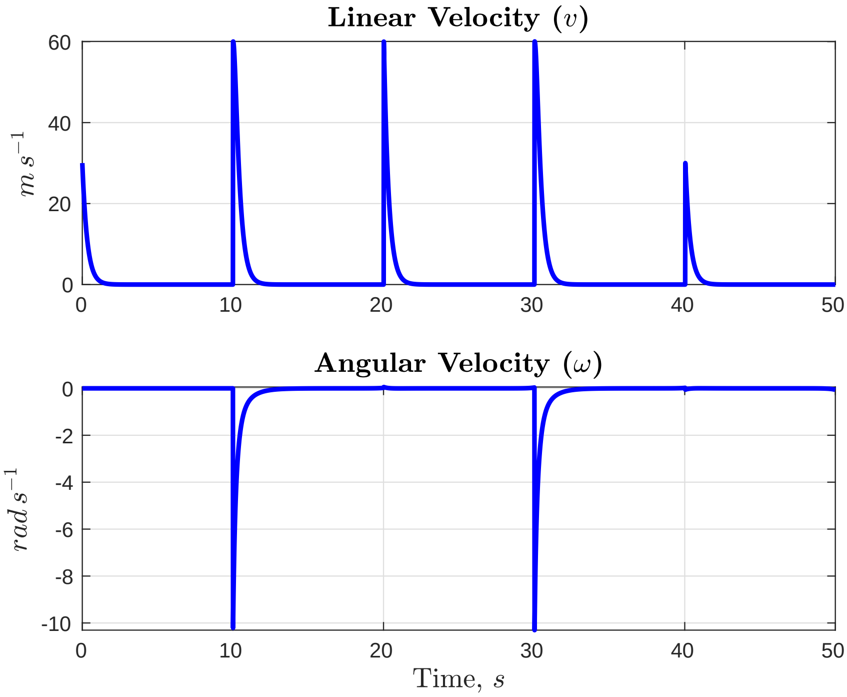


Figure 2.15: Control signal for $v$ and $\omega$.

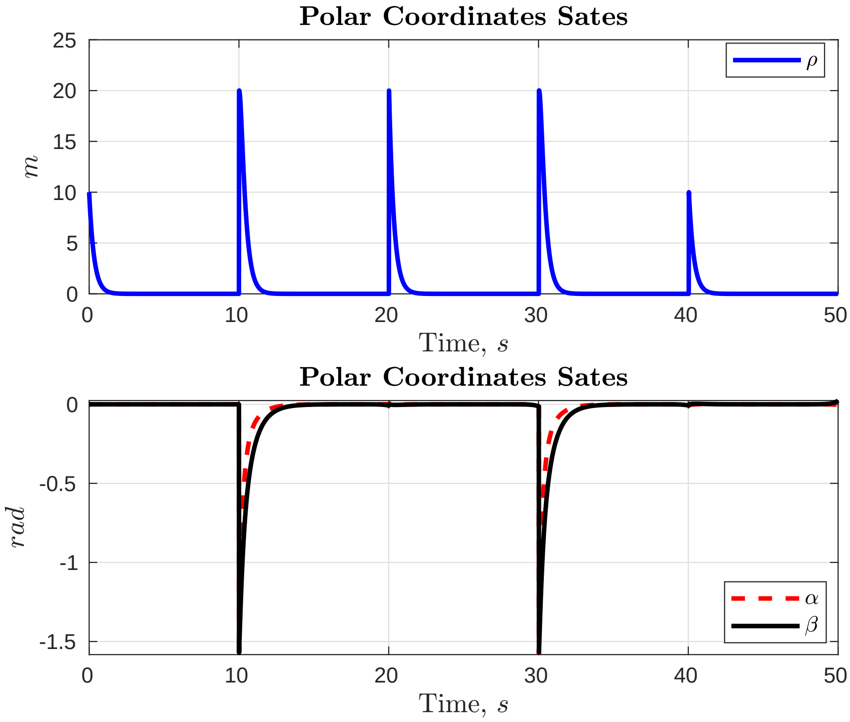


Figure 2.16: State variable for linearized polar system.

which it is relatively easy to get stuck, demanding a thick tread tire and a large diameter wheel.

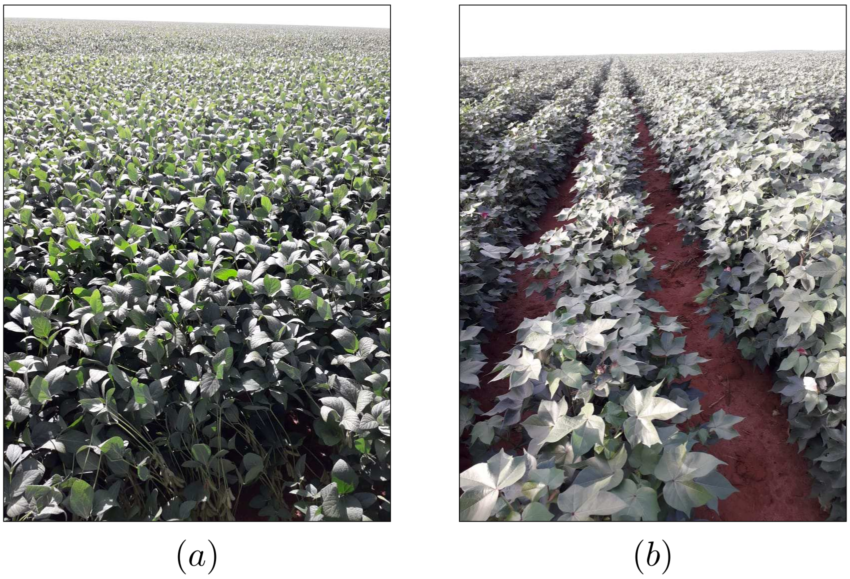


Figure 2.17: Row crops on clay soil for (a) soybeans and (b) cotton.

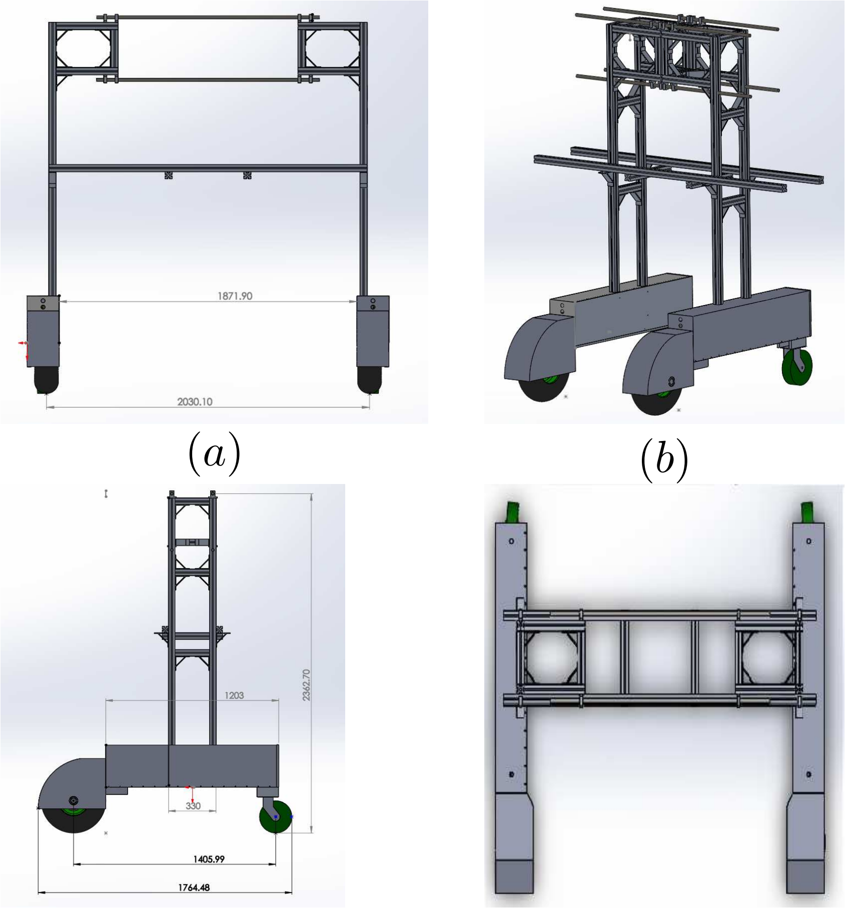
Other relevant specifications for the robot mechanical design are the distance between rows, the plant height and width within rows and the crop stage. In order to perform M&I tasks on different types of crops with different planting characteristics, it is mandatory to design a multipurpose mobile robot whose mechanical structure can be easily reconfigured on the field (Fig. 2.18). In this work, we consider the row width (W), the minimum and maximum plant heights (H) and the slope of the terrain as shown in Table 2.4.

Table 2.2: Planting distances for soy and cotton crops in Brazil.

| | **Planting Distances** | | | |
|---|---|---|---|---|
| **Crop Type** | ***W (m)*** | ***min. H (m)*** | ***max. H (m)*** | ***slope (deg)*** |
| Soy | 0,45 | 0.08 | 1.80 | 20 |
| Cotton | 0.75 | 0.08 | 1.20 | 20 |

Based on the specifications of cultivation and soil previously discussed, we develop the 3D model of the robot prototype by using a *SolidWorks* CAD (Computer Aided Design) environment. After the last phase of development, the resulting mechanical design has an inverted "U" shaped frame and allows robot movements between crop rows as well as top, bottom and side views of the plants.

As it can be seen, while each of its lateral columns moves between the rows, resting on the chassis, its upper part moves above the plantation carrying both the monitoring equipment and all the embedded electronics that allows



$(a)$ $(b)$

$(c)$ $(d)$

Figure 2.18: Standard views of the robot structure: (a) front; (b) perspective; (c) side; (d) top.

its operation. The robot has been developed with the ability to be reconfigured in order to adapt its structure to different types of crops, in particular, row crops. The frame consists of two mechanically equivalent and symmetrical substructures, allowing the manual adjustment of its width.

This unique characteristic gives the robot the flexibility to operate in crops with smaller spacing between rows (e.g., soybeans), or in crops with larger spacing (e.g., cotton). In addition to the lateral extension capability, the robot has a transverse structure with manually adjustable height, where it is possible to install monitoring and inspection devices such as HD cameras and range sensors. For future applications, a lightweight robot arms can be attached to this structure for collecting samples as well as installing actuation systems to enable the remote height reconfiguration.

The minimum height of this structure with respect to the ground is the base of the robot chassis (approx. 0.4 $m$) and the maximum height is reached when the frame is elevated until the top of the structural column (approx. 1.9 $m$), as shown in Fig. 2.18.

### 2.4.1 Embedded Electronics, Sensing and Control

In this section we present detailed information about the design of the embedded electronics, sensory and control systems. To provide the necessary power to move the vehicle and support all electronic devices, two Lithium batteries 25 $Ah$, 48 $V$ DC with Samsung ICR18650-22FU cells (2200 $mAh$, 3.6 $V$), were mounted in a parallel configuration providing a total capacity of 2400 $Wh$, which allows the robot to operate continuously for approximately 8 hours. The block diagram presented in Fig. 2.19 shows the energy distribution scheme in the robotic system.

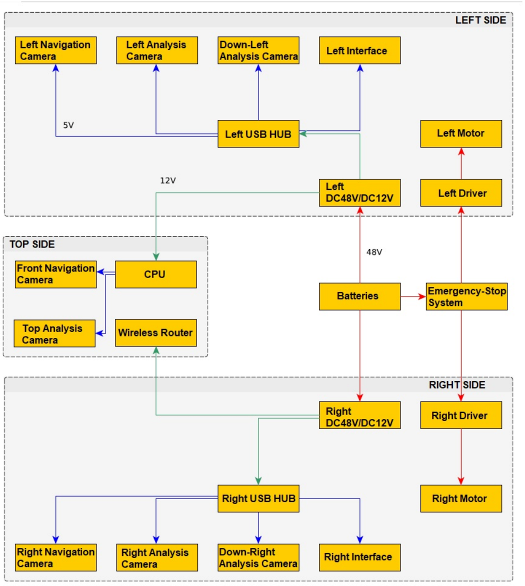


Figure 2.19: Diagram of energy distribution.

To provides power to secondary electronics devices, such as cameras and microcomputers, was used a 48-12 $V$ DC-DC. The robot has an emergency stop system, composed by six push buttons fixed in its fairing, that cut the power of the motor driver. The Fig. 2.20 shows the circuit diagram of the Emergency Stop Device (ESD).

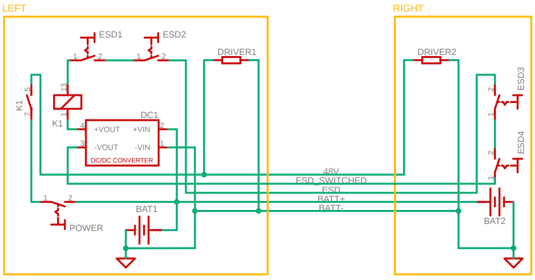


Figure 2.20: Electrical diagram of Emergency Stop Device.

The Robot Operating System (ROS) framework, version **Melodic Morenia**, is used to manage the communication, actuation, sensing and control systems. It runs on the Linux OS Ubuntu 18.04 LTS by using an Intel NUC7i5BNK Mini PC, Intel Core i5-7260U 2.2GHz, 16GB DDR4 RAM. The ROS packages to visually control the robot motion were developed ad-hoc in the Python programming language. Two custom-designed boards, each one with an ATMEGA32U4-AU, 8bit AVR Microcontroller attached, were built to allow the low-level interface between the NUC and the motor power drivers. The boards receive an emulated encoder signal from the motor driver generated by hall sensors embedded in the motor shell, which are used to calculate wheel speeds (Fig. 2.21).

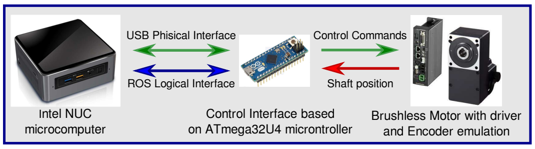
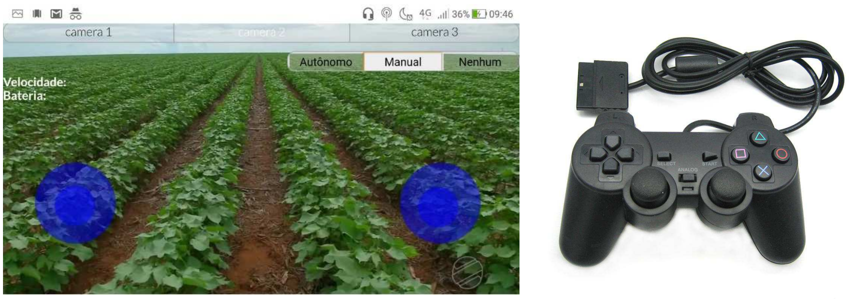

USB Phisical Interface
ROS Logical Interface
Control Commands
Shaft position
Intel NUC microcomputer
Control Interface based on ATmega32U4 micronntroller
Brushless Motor with driver and Encoder emulation

Figure 2.21: Diagram of the motor-driver interface.

The robot motion is controlled by using the visual feedback provided by three WebCams Logitech C270 HD (720p and 30 fps): two located at the robot fenders and one fixed on the top of the robot frame. The camera on top can be used when the crop height is low, whereas the cameras on the fender are suitable for navigation in high crops where the top view has a uniform image. Along with this three cameras, five high resolution cameras (model 13M, from ELP) was used for data collection to monitoring diseases and pests. The mobile robot can also be remotely controlled by using two solutions implemented in the ROS framework: one uses a standard USB joystick and another uses a web application based on a JavaScript program which can be accessed through a mobile device.

Two USB hubs and a Wi-Fi router were used to implement a communication network where a Graphical User Interface (GUI) is connected. The GUI allows the operator to visualize the environment using the robot cameras, monitor the telemetry data (e.g., battery status, wheel speeds, control inputs and outputs) and select the operation mode (e.g., manual or autonomous) as shown in Fig. 2.22. For crop monitoring and inspection, five cameras were fixed at the robot structure at different locations, collecting images to be analyzed subsequently.

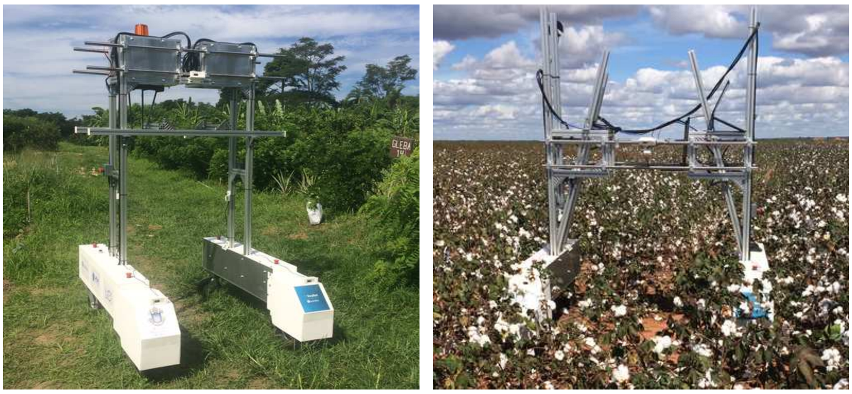


(a) (b)

Figure 2.22: Devices for remote control: (a)App for robot control in JavaScript.; (b) PS2-like USB joystick.

### 2.4.2 Preliminary Fields Tests

Considering the robot structure sizing (e.g., static, fatigue and damage tolerance analysis) according to 3D CAD models and satisfying all design specifications defined by the type of terrain and plant height/width within rows, the mechanical design of the robot was finally concluded, presenting an inverted "U" shaped rectangular structure mounted on two driving cars, as shown in Fig. 2.23.



(a) (b)

Figure 2.23: Preliminary field tests of the SoyBot robot: (a) Experimental field from Embrapa Agrobiology, Rio de Janeiro; (b) Soy and Cotton farm, west region of Bahia state, Brazil.

In the preliminary tests of translational and rotational motions, it was observed that the robot has suffered a slight torsion around an axis perpendicular to its sagittal axis. This effect occurs only when the mechanical structure is mounted in the configuration of maximum dimension (i.e., higher height and wider width) as shown in Fig. 2.24. However, this effect does not occur when the mechanical structure is mounted at the minimum dimension configuration.

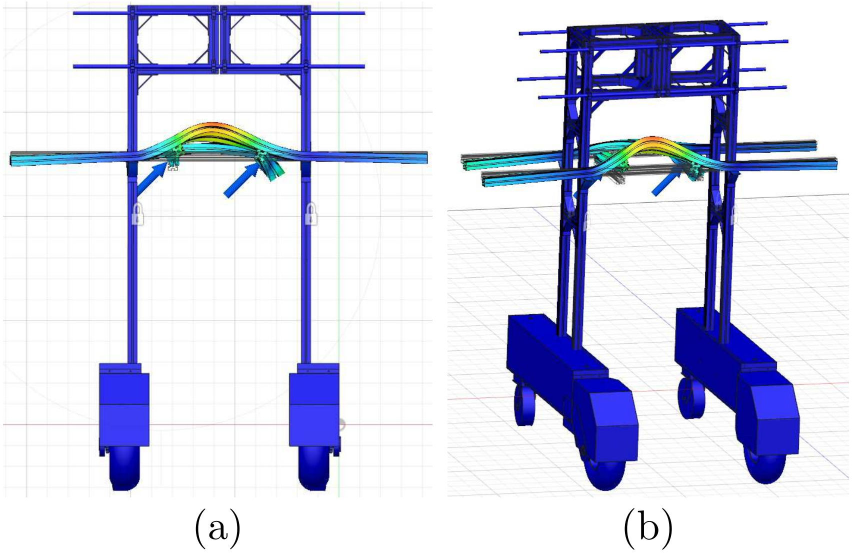


Figure 2.24: Simulation of efforts and vibrations in the robot mechanical structure: (a) buckling effect; (b) soft deformation.

In this context, with the aim to evaluate efforts and vibrations in the robot structure, two simulations in *Autodesk Fusion 360* were developed. In the first simulation, two forces at the middle of the structure with 50 $kN$ of magnitude and 45 degrees of inclination with the $z$ axis were considered. The result was a torsonial buckling at the horizontal bars, but the structure does not have apparent collapse. In the second simulation, again two forces at the middle of the structure with 500 $N$ of magnitude (maximum load of project) and 45 degrees of inclination with the $z$-axis and other two horizontal forces with 500 $N$ of magnitude (maximum load of project) to simulate the motor forces were considered. The result was only a soft deformation at the base of one side of the robot. This torsion effect was quite reduced during the tests *in-situ* by fixing transverse bars in the robot structure. So, the robot was able to move in a straight line and perform small maneuvers on different terrains satisfactorily, without twisting its structure. In the next section, we will introduce the field tests in autonomous mode using visual servoing approach. Moreover, we will discourse about the application of a robust control technique for tracking tasks.

## 2.5 Concluding Remarks

This chapter presented some methodologies for robot modeling besides some robot models, such as unicycle (e.g. differential drive) and bicycle (e.g. car-like robot).Several kinematic-based controllers were proposed and numerical simulations was presented to validate the feasibility of the control schemes. According to the simulations, linear controllers can be used to control a non-linear robot since the system remains in the linear regions (e.g. low velocities and small acceleration). However, when non-linear controllers are used, the system presents better response due to this kind of approach being able to handle all state space, including linear and non-linear regions.

We also present the design, development and constructive aspects of a differential drive mobile robot named SoybBot, able to navigate autonomously in agricultural environments such as soybean and cotton crops, performing monitoring and inspection task. Detailed information about the mechanical system, embedded electronics, communication protocols, actuation and sensor systems, as well as the control system were presented and extensively discussed throughout the sections.

Preliminary field tests were carried out to verify and validate the performance and feasibility of the designed mechanical structure of the SoyBot robot. During these tests it was found some structural issue due to the height of the robot, that was solved by changes the center of gravity of the robot and reinforcing its side structures.

In the next section, we will present the proposed control approach for autonomous navigation using image based visual servoing. Experimental results ara presented in order to validate the proposed control approach.

# 3
# Advanced Control Design for an Agricultural Mobile Robot

In this chapter, we cover the control design of the SoyBot wheeled mobile robot, presented in Chapter 2. This robot is capable of performing autonomous navigation tasks in agricultural fields, and its control methodology is based on the kinematic approach due to its known ability to ensure satisfactory performance when robot movements are performed at slow speeds and slow accelerations. The control strategy for trajectory stabilization and tracking purposes is based on a visual servo algorithm and considers the straight line tracking problem using a constant linear velocity in Cartesian space and position error in image space. Besides that, We propose a robust control approach based on the sliding mode control technique in order to perform regulation tasks in the presence of uncertainties, a situation commonly encountered in field applications, and a homogeneous robot swarm formation control proposed that It can be used to perform tasks in extensive crops such as cotton, soybeans, sugar cane, among others. The numerical simulation results are presented to illustrate the effectiveness and feasibility of the sliding mode approach. Simulation results using the Gazebo robot simulator are presented to validate the proposed control approach for this task. Finally, experimental tests using the SoyBot mobile agricultural robot are performed to verify and validate the visual servo control approach for cotton crop monitoring.

## 3.1
## Problem Statement

In the agricultural industry, automation and mechanization allow for the cultivation of gigantic areas, which demands minimally structured environments, such as plain terrain, georeferencing, and alignment of the plantation (Fig. 2.17). This type of infrastructure contributes to the use of autonomous machinery as well as mobile robotic system, such rovers or drones, that can use the row crops as guide for localization and navigation purposes [49, 50].

In this context, the control problem consists in guiding a mobile robot through the plantation using crop rows as a reference path, in the same way as a line follower robot (e.g., AGV systems). To perform this task, the proposed approach is to use a single camera as an input sensor [51], providing the desired orientation to the mobile robot, while the forward velocity remains constant. The line following task is useful in data collection routines, such as soil sample collection, image data set creation, environmental conditions monitoring, or in monitoring tasks such as weed detection [50], where the robot has to move through the plantation in an autonomous mode. In this case, the robot task is to collect images to compose a data set of pests and diseases in cotton and soybean crops.

Due to the robot model have already been introduced, in the next sections we will present the proposed control scheme for autonomous navigation using the visual servoing approach.

## 3.2 Visual Servoing for Autonomous Navigation

### 3.2.1 Camera-Robot System Model

Consider a pin-hole camera moving with the body velocity $(V, \Omega) \in \mathbb{R}^6$ and observing an arbitrary point $\mathcal{P}$ located at the agricultural environment with camera frame coordinates $p_c = (x_c, y_c, z_c) \in \mathbb{R}^3$. The velocity of the point $\mathcal{P}$ with respect to the camera frame $\mathcal{F}_c$ is given by:

$$\dot{p}_c = -\Omega \times p_c - V\,, \tag{3-1}$$

where $V = (v_x, v_y, v_z) \in \mathbb{R}^3$ and $\Omega = (\omega_x, \omega_y, \omega_z) \in \mathbb{R}^3$ are the linear and angular velocities expressed in the camera frame, or camera spatial velocities. The point $\mathcal{P}$ can be represented in the image space using the following relationship:

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} \alpha_x & 0 \\ 0 & \alpha_y \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_{v0} \\ y_{v0} \end{bmatrix}\,, \tag{3-2}$$

with $(x_v, y_v) \in \mathbb{R}^2$ are the coordinates of the point $\mathcal{P}$ expressed in the image frame $\mathcal{F}_v$, $f > 0$ is the focal length of the camera lens, $\alpha_x, \alpha_y > 0$ are the camera scaling factors, $(x_{v0}, y_{v0}) \in \mathbb{R}^2$ are the coordinates of the principal point.

Here, without loss of generality, we assume that: (A1) the mobile robot operates in the $xy$-plane and rotates only about $z$-axis of the world frame $\mathcal{F}_w$; (A2) the camera is attached to the robot structure so that its optical axis is parallel to the sagittal axis of the vehicle, as can be seen in Fig. 3.1, that shows how the camera is attached to the robot chassis as well as the system frames: world $\mathcal{F}_w$; robot $\mathcal{F}_r$; and camera $\mathcal{F}_c$. In this context the relationship between the pixel motion and the camera spatial velocities can be obtained applying (3-1) into the time-derivative of (3-2), leading to:

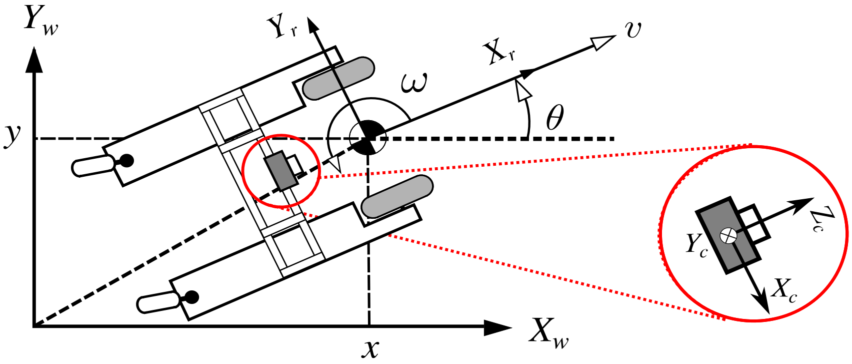


Figure 3.1: Representation of the differential-drive mobile robot, its generalized coordinates and the world, robot and camera frames.

$$\begin{bmatrix} \dot{x}_v \\ \dot{y}_v \end{bmatrix} = \begin{bmatrix} -\dfrac{f'}{z_c} & 0 & \dfrac{\bar{x}_v}{z_c} & \dfrac{\bar{x}_v\,\bar{y}_v}{f'} & -\dfrac{f'^2+\bar{x}_v^2}{f'} & \bar{y}_v \\ 0 & -\dfrac{f'}{z_c} & \dfrac{\bar{y}_v}{z_c} & \dfrac{f'^2+\bar{y}_v^2}{f'} & -\dfrac{\bar{y}_v\,\bar{y}_v}{f'} & -\bar{x}_v \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \qquad (3\text{-}3)$$

where $\bar{x}_v = x_v - x_{v0}$, $\bar{y}_v = y_v - y_{v0}$ and $f' = f\alpha$, assuming a square shape pixel $(\alpha_x = \alpha_y = \alpha)$.

As can be seen in the Fig. 3.1, the camera motion is constrained by the robot motion constrain, that allows motions only in the x-axis direction and around the z-axis (linear and angular velocities respectively). The robot spatial velocities are projected to the image frame through the adjoint matrix [47], which relates both camera and robot velocities. This transformation is given by:

$$\nu_c = \begin{bmatrix} R_{cr} & [\mathbf{r}_{cr}]_\times\, R_{cr} \\ \mathbf{0}_{3\times3} & R_{cr} \end{bmatrix} \nu_r \; = \; Ad(\xi_{cr})\nu_r\,, \qquad (3\text{-}4)$$

where $R_{cr} \in SO(3)$ and $r_{cr} \in \mathbb{R}^3$ are the rotation matrix and translational vector from the camera frame to the robot frame, $\nu_c\,,\ \nu_r \in \mathbb{R}^6$ are the spatial velocities vector of the camera and robot respectively. The translational vector is given by $r_{cr} = \begin{bmatrix} 0 & 0 & z_c \end{bmatrix}^\mathsf{T}$ due to the origins of the camera and robot frames are aligned. The camera have rotations around z-axis and x-axis of $\theta_z = \theta_x = -\pi/2\,\mathrm{rad}$, defining the rotation matrix as:

$$R_{cr} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}. \qquad (3\text{-}5)$$

and the adjoint matrix as:

$$Ad(\xi_{cr}) = \begin{bmatrix} 0 & -1 & 0 & z_c & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & z_c & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} . \tag{3-6}$$

Finally, the dynamical system that relates the robot input control to the pixel behavior in the vision frame is given by:

$$\begin{bmatrix} \dot{x}_v \\ \dot{y}_v \end{bmatrix} = \begin{bmatrix} \dfrac{\bar{x}_v}{z_c} & \dfrac{\left(f'^2 + \bar{x}_v^2\right)}{f'} \\ \dfrac{\bar{y}_v}{z_c} & \dfrac{\bar{x}_v\,\bar{y}_v}{f'} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} . \tag{3-7}$$

### 3.2.2 Adaptive Algorithm for Image Segmentation

For the row crop navigation task, the information that needs to be extracted from the RGB image, captured by the camera, is the position of an arbitrary point $\mathcal{P}$ (midpoint) located between two rows where the vehicle is moving. From this information, the robot is able to perform alignment maneuvers while moving through the plantation. One way to obtain such information is to use an image segmentation technique and extract the centroid of the region of interest (RoI).

The proposed navigation task in open fields demands a robust segmentation algorithm in order to deal with changing light conditions. A robust solution proposed by [43] uses a combination of two techniques: an RGB color space transformation to the OHTA color space and an adaptive threshold algorithm based on the image histogram analysis. The OHTA color space is obtained from a linear conversion between the RGB color space into an orthogonal color feature set (or channels) $I_i$ and $I'_i$ for $i\!=\!1,2,3$, which is given by:

$$\begin{aligned} I_1 &=(R+G+B)/2\,, & I'_1 &=R-G\,, \\ I_2 &=(R-B)/2\,, & I'_2 &=R-B\,, \\ I_3 &=(2G-R-B)/4\,, & I'_3 &=(2G-R-B)/2\,. \end{aligned} \tag{3-8}$$

Notice that, the original RGB color space uses three color channels (RGB) per image feature, while OHTA color space uses a grayscale channel which reduces the computational cost.

The next step consists of creating a binary image, where is applied the Otsu method, based on an image histogram analysis for dynamic class separation. The key idea is to compute the threshold value for binarization based on the quotient between the biggest value of between-class variance and the sum of this value and the inner-class smallest value (collectively variance).

The solution proposed by [43] was used for fruit detection, where the objective was to extract features from ripe fruits in complex backgrounds. For this case, the channels of the OHTA color space, $I_2$, $I_1'$ and $I_2'$, was chosen due to the predominance of the red colors tones. For the row crop navigation task, the predominant color tones are yellow (floor and ground) and green (leaf and plant). The channels that presented the best results was $I_1'$ and $I_3'$. To improve the segmentation process, it was applied a Gaussian Blur filter [52] for smoothing and noise reduction into the OHTA channels before the binarization process using the Otsu method.

The arbitrary point $\mathcal{P}$ is given by the centroid coordinates $(x_v, y_v)$ of the biggest area in the binary image (see Fig. 3.2(c)), on which is computed its displacement $(e_v)$ with respect to the center of the image $(x_{vd}=0)$. The steps of the image feature extraction process are shown in Fig. 3.2.

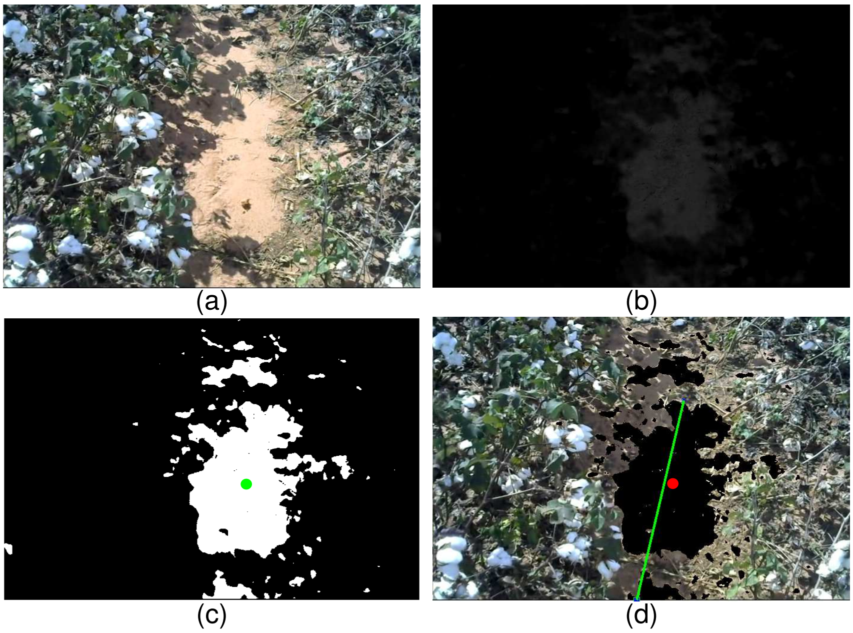


Figure 3.2: Steps of crop row alignment: (a) raw image; (b) OHTA channel $I_1'$; (c) Binary image (mask) and its centroid; (d) estimated robot misalignment.

### 3.2.3 Algorithm for Orientation Control

Consider that the control goal is to keep the centroid coordinates $(x_v, y_v)$ of same interest region, obtained through the use of an appropriate image feature extraction technique, aligned with a desired point $(x_{vd}, y_{vd})$ in the image frame while the linear velocity $v$ of the robot remains constant. Here, without loss of generality, we consider that during the robot motions through the row crop, there is no motion of point $\mathcal{P}$ along the $y$-axis in the image frame $\mathcal{F}_v$ and its depth coordinate $z_c$ remains constant. Under these assumptions, the control system presented in (3-7) takes the form:

$$\dot{x}_v = \frac{\bar{x}_v}{z_c}\, v_d + \frac{\left(f'^2 + \bar{x}_v^2\right)}{f'}\, \omega\,. \qquad (3\text{-}9)$$

The visual servoing problem is to ensure that the mobile robot visually follows the image target, keeping its orientation aligned to the row crop. In this context, the control goal can be simply defined by:

$$x_v \to x_{vd}\,, \qquad e_v = x_{vd} - x_v \to 0\,, \qquad (3\text{-}10)$$

where $e_v \in \mathbb{R}$ is the image error in $x$-axis of the image frame $\mathcal{F}_v$ and $x_{vd}$ is assumed to be constant or zero. Then, in order to guarantee the stabilization of the image error $e_v$ to zero, we propose the following control law:

$$\omega = \left(\frac{f'}{f'^2 + \bar{x}_v^2}\right)\left(k_v\, e_v - \frac{\bar{x}_v}{z_c}\, v_d\right)\,, \qquad k_v > 0\,, \qquad (3\text{-}11)$$

where $k_v > 0$ is the proportional gain. Notice that, all necessary signals to implement the control law (3-11) are available since $x_v$ is measured by the camera and the camera parameters such as $f'$ and $z_c$ are known.

Then, taking the time-derivative of (3-10) and substituting (3-9), we obtain the following image error equation:

$$\dot{e}_v = -\left[\frac{\bar{x}_v}{z_c}\, v_d + \frac{\left(f'^2 + \bar{x}_v^2\right)}{f'}\, \omega\right]\,. \qquad (3\text{-}12)$$

Now, we can state the following theorem to summarize the main result of this section:

**Theorem 3.1** *Consider the vision-based control system (3-7). Under the assumptions (A1)-(A2) and considering that the desired position $x_{vd}$ is bounded, the following control law (3-11) where $k_v > 0$ is the proportional gain, ensures the asymptotic convergence of the image error $e_v$ to zero, that is,* $\lim_{t\to\infty} e_v(t) = 0$.

*Proof.* Consider the following Lyapunov-like function candidate $2V(e_v) = e_v^2$. Then, taking its time-derivative along the trajectories of the error system (3-12) and using the control law (**??**) we obtain $\dot{V}(e_v) = -k_v e_v^2 < 0$ which is negative definite. Since $V > 0$ and $\dot{V} < 0$ for $\forall e_v \neq 0$ implies that $V, e_v \in \mathcal{L}_\infty$. Because $V(e_v) \to \infty$ as $|e_v| \to \infty$ and $\dot{V} < 0$ implies that $V(e_v) \leq V(0)$.

Therefore, the equilibrium at the origin $e_v = 0$ is globally asymptotically stable and $\lim_{t\to\infty} e_v(t) = 0$. ■

## 3.3 Robust Control for Tracking Task

According to [46], the Sliding Mode Control (SMC) is a special mode in variable structure systems (VSS), where the idea is based on the introduction of a custom-designed function, named the sliding variable. As soon as the properly designed sliding variable becomes to equal to zero, it defines the sliding manifold or the sliding surface. In this section, we propose a SMC approach to achieve stabilization for a differential drive robot, based on coordinate transformation to chained form [53] and Numerical simulation are presented for validate the approach.

### 3.3.1 Robust Control for Differential Drive Robot Using Sub-Optimal Chained Form Transformation

Consider the differential model system in (2-5) and the coordinate transformation to sub-optimal chained form given by:

$$\begin{cases} z_1 = \theta \,, \\ z_2 = x\cos(\theta) + y\cos(\theta) \,, \\ z_3 = x\sin(\theta) - y\cos(\theta) \,, \end{cases} \tag{3-13}$$

leading to a dynamic system given by:

$$\begin{cases} \dot{z}_1 = u_1 \,, \\ \dot{z}_2 = u_2 \,, \\ \dot{z}_3 = z_2 u_1 \,, \end{cases} \tag{3-14}$$

where system inputs can be computed using the follow transformation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} z_3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} . \tag{3-15}$$

This well-known transformation can be used to design a robust control to perform a regulation task that can handle parametric uncertainties and external perturbations (e.g. wheel radius uncertainties, slip component). Thus, the following theorem can be proposed:

**Theorem 3.2** *Consider the kinematic drive model system in (2-5) and (2-6), the coordinates transformation (3-15) and the sub-optimal chained form system*

*(3-14). Then, the following control law:*

$$u_1 = -z_1 - \alpha\, z_2\, sign(\sigma), \tag{3-16}$$

$$u_2 = -z_2 + \alpha\, z_1\, sign(\sigma), \tag{3-17}$$

*where $\alpha > 0$ and $\sigma = 2z_3 - z_1\, z_2$ is the sliding surface, ensure the stabilization of system states at zero.*

*Proof.* Consider the Lyapunov-like candidate function $2V(z_1\,, z_2) = (z_1^2 + z_2^2)\,$, which its time derivative along the system trajectories is given by $\dot{V} = z_1 u_1 + z_2 u_2$. The close-loop system using the control laws (3-16) and (3-17) leads to $\dot{V} = -(z_1^2 + z_2^2) = 2V \leq 0$, bringing the states $z_1$ and $z_2$ to zero. Due to the sufficient sliding condition being defined by $\sigma\dot{\sigma} < -\beta\,|\sigma|\,, \beta > 0$ [54], with $\dot{\sigma} = -2\,\alpha V\,\mathrm{sign}(\sigma)$ implies in $\sigma\dot{\sigma} = -2\,\alpha V \sigma\,\mathrm{sign}(\sigma) < -\beta\,|\sigma|$. Therefore, as $V$ is a positive definite function, the absolute value of $\sigma$ decreases and converging to zero in finite time if $V(0) > |\sigma(0)|$ and $\dot{V} > \dot{\sigma}\,\mathrm{sign}(\sigma)$. With the sliding condition satisfied, all the system trajectories converge to the sliding surface $\sigma = 0$ and $z_1 z_2 = 2z_3$, implying in the convergence to $z_3$ to zero, ensuring the local stability of origin. ■

In order to validate this approach we can rewrite the system (2-5) as function of wheel speed and taking into account the slip coefficient:

$$\dot{x} = \hat{r}[\cos(\theta) + \gamma \sin(\theta)]v_m\,, \tag{3-18}$$

$$\dot{y} = \hat{r}[\sin(\theta) + \gamma \cos(\theta)]v_m\,, \tag{3-19}$$

$$\dot{\theta} = \hat{r}\,\omega_m, \tag{3-20}$$

where $\hat{r} = r + \delta_r$ is the uncertain wheel radius, with $r$ and $\delta_r$ being the wheel radius and the uncertainty in its value, respectively, $v_m = (\dot{\phi}_L + \dot{\phi}_R)/2$ and $\omega_m = (\dot{\phi}_L - \dot{\phi}_R)/d$ are the new control inputs and $\gamma$ the slip coefficient. The simulation results consider wheel radius uncertainties of $0.15\,\%$ and slip coefficient $\gamma = 0.2$. Fig. 3.3 shown trajectories from several initial position with same initial orientation $\theta_0 = 0$. As can be seen, the robot achieve the goal starting from all coordinates. In Fig. 3.4 we can see that the position and orientation error tends to zero as the coordinates in chained form tends to zero.

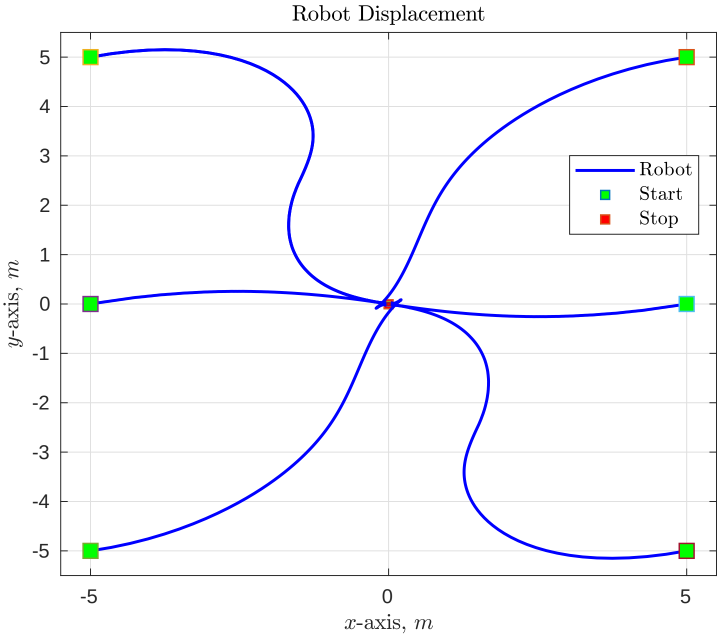


Figure 3.3: Displacement of the robot in $(x\,,y)$ plane from several initial conditions. For this case all conditions start with zero orientation.

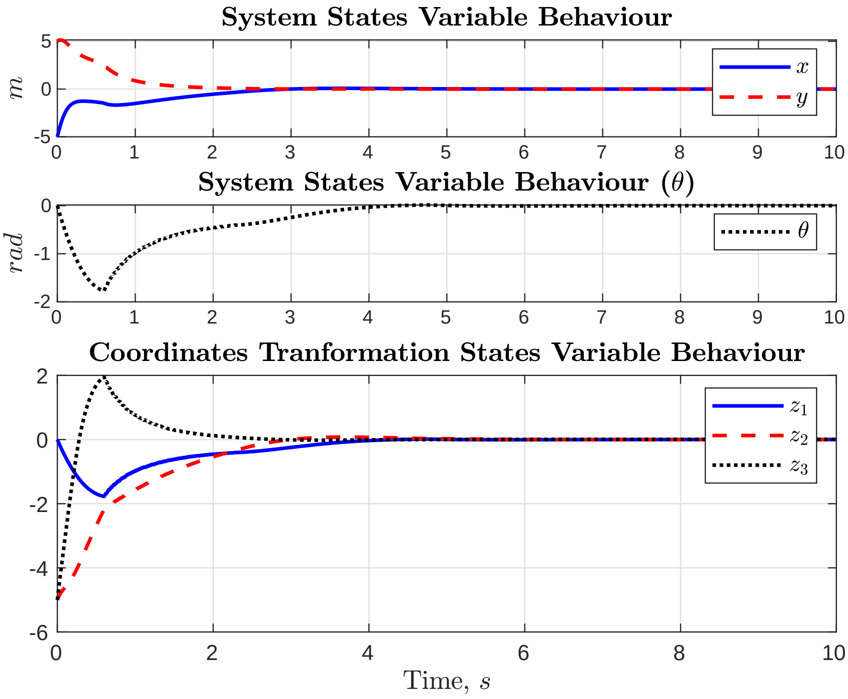


Figure 3.4: Behaviour of the system state variables for Cartesian and chained form coordinates.

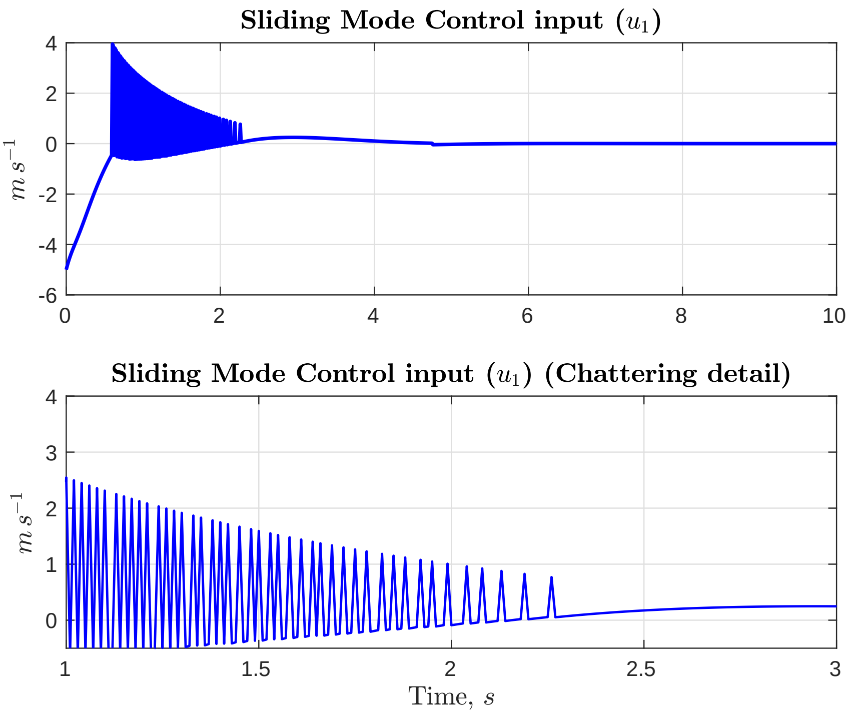


Figure 3.5: Input variable $u_1$ for chained form system. At the bottom we can see the shattering effect due to the switching effect.

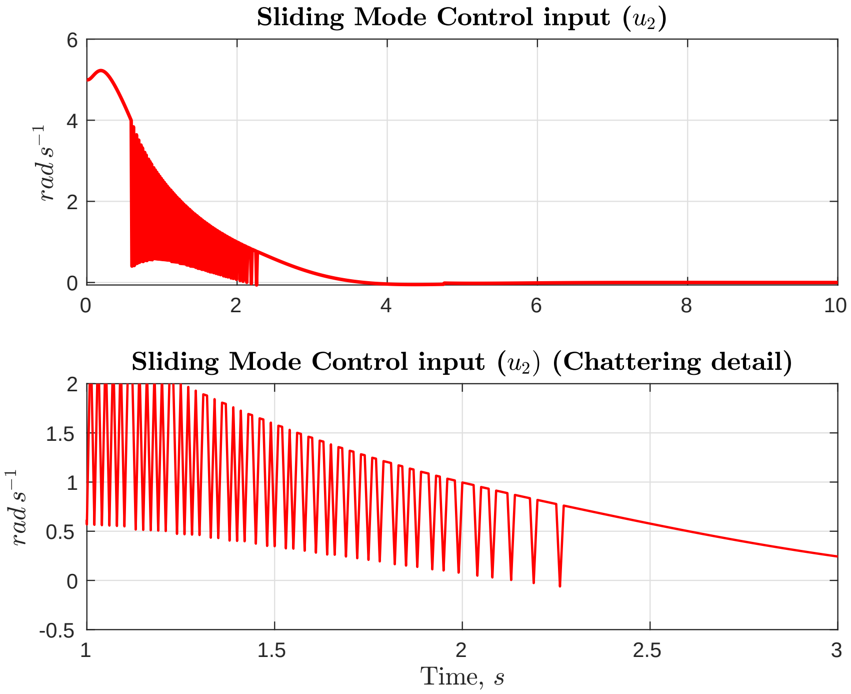


Figure 3.6: Input variable $u_2$ for chained form system. At the bottom we can see the shattering effect due to the switching effect.

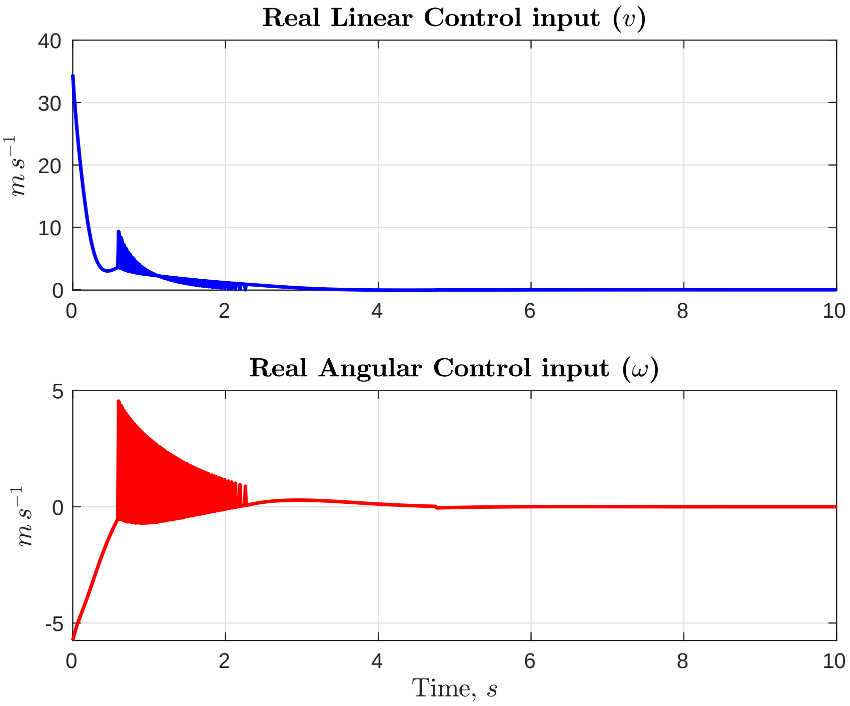


Figure 3.7: Linear velocity input variable $v$ for Cartesian system. At the bottom we can see the shattering effect due to the switching effect.

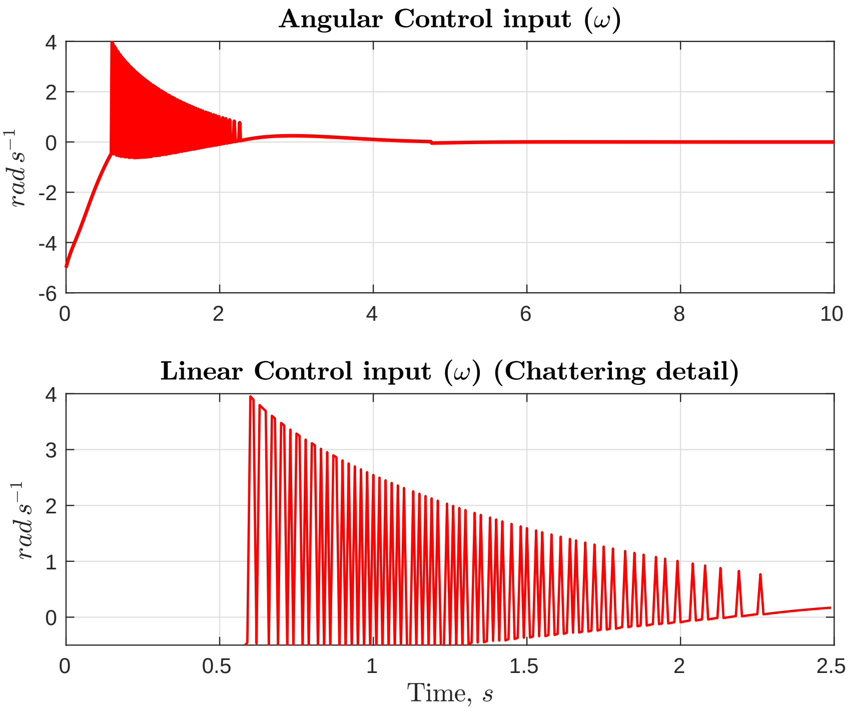


Figure 3.8: Angular velocity input variable $\omega$ for Cartesian system. At the bottom we can see the shattering effect due to the switching effect.

### 3.3.2 Robust Control for Differential Drive Robot Using Optimal Chained Form Transformation

Consider the differential drive model system in (2-5). We can propose a coordinate transformation to the chained form given by:

$$\begin{cases} z_1 = x\cos(\theta) + y\sin(\theta)\,, \\ z_2 = \theta\,, \\ z_3 = -\theta[\,x\cos(\theta) + y\sin(\theta)\,] + 2[\,x\cos(\theta) - y\sin(\theta)\,]\,, \end{cases} \tag{3-21}$$

which leads to the dynamic system in a chained form, given by:

$$\begin{cases} \dot{z}_1 = u_1\,, \\ \dot{z}_2 = u_2\,, \\ \dot{z}_3 = z_1 u_2 - z_2 u_1\,, \end{cases} \tag{3-22}$$

where system inputs can be computed using the follow transformation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & x\sin(\theta) - y\cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{3-23}$$

Under this condition, we can propose the follow theorem:

**Theorem 3.3** *Consider the kinematic model of a differential-drive mobile robot given by (2-5) and (2-6), the coordinates transformation (3-21) and the chained form system (3-22). Then, the following control laws:*

$$u_1 = -z_1 + \alpha\, z_2\, sign(\sigma)\,, \tag{3-24}$$

$$u_2 = -z_2 - \alpha\, z_1\, sign(\sigma)\,, \tag{3-25}$$

*where $\alpha > 0$ is the sliding gain and $\sigma = z_3$ is the sliding surface ensure the asymptotic stability of the system states to zero.*

*Proof.* Consider the following Lyapunov-like candidate function $2V(z_1\,, z_2) = z_1^2 + z_2^2$. Then, taking its time-derivative along the system (3-22) and using the control laws (3-24) and (3-25), we obtain $\dot{V}(z_1\,, z_2) = -z_1^2 - z_2^2 \leq 0$, which is negative definite. Since $V > 0$ and $\dot{V} < 0$, for $\forall\, z_1\,, z_2 \neq 0$, implies that $V, z_1, z_2 \in \mathcal{L}_\infty$. Therefore, the equilibrium point $z_1 = z_2 = 0$ is globally asymptotically stable and $\lim_{t\to\infty} z_1(t) = 0$ and $\lim_{t\to\infty} z_2(t) = 0$. Consider that $\sigma = z_3 = 0$ is a sliding surface and the sliding condition given by $\dot{\sigma}\sigma < -\beta\,|\sigma|$, with $\beta > 0$. Taking the following Lyapunov-like function candidate $2V(\sigma) = \sigma^2$ its time derivative along the system trajectories and applying (3-24) and (3-25), we obtain $\dot{V}(\sigma) = -\alpha\,(z_1^2 + z_2^2)\,\text{sign}(\sigma) < -\beta|\sigma|$. Because $\sigma \in \mathcal{L}_\infty$, and $V$

is positive for all $\sigma \neq 0$, if the sliding condition is hold, the system converges to zero in finite time and $\lim_{t\to\infty} \sigma(t) = 0$.

■

To validate this approach, the following simulation was performed, where the robot needs to reach the system origin with a desired orientation, using a gain value $\alpha = 1$. In this case, we used the system presented in (3-18) and the same values for radius uncertainty ans slip coefficient. Fig. 3.9 shows the trajectories of the robot in the plane, starting from several initial conditions. Fig. 3.10 shows the behaviour of the state variables for Cartesian and chained form, from the initial condition $q_0 = [-5\,,5\,,0]^{\mathsf{T}}$. As we can see, all the system state variables converge to the origin in finite time, stabilizing the robot. Figs. 3.11 and 3.12 shows the input signal from Sliding Mode Controller, while Figs. 3.13 and 3.14 presents the input signal for Cartesian model, before the input transformation (3-23).

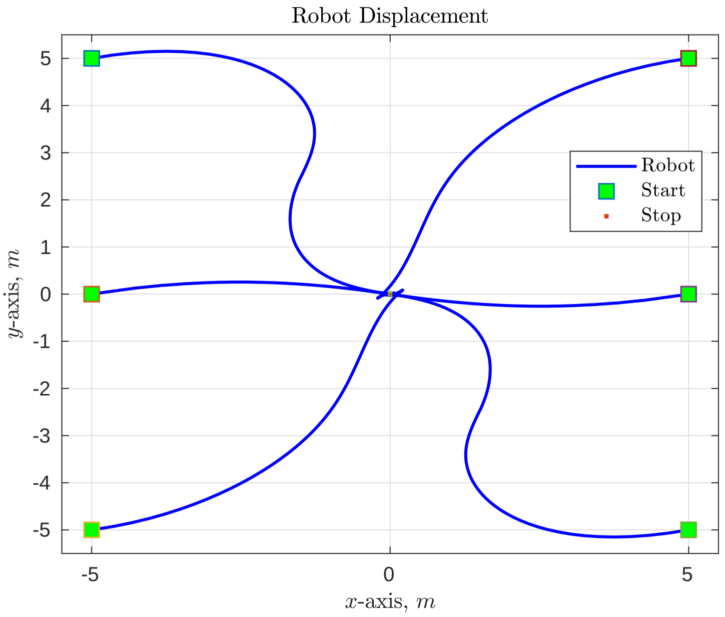


Figure 3.9: Displacement of the robot in $(x\,,y)$ plane from several initial conditions. For this case all conditions start with zero orientation.

### 3.3.3 Chattering Reduction with Quasi-SM

The chattering effect can be observed when the SMC approach is applied. This effect is not desirable in many application such as aircraft or mobile robots. Many techniques can be used to avoid this effect and maintain the controller robustness and its characteristics of insensitivity to bounded model uncertainties and external disturbance. One of this techniques presented in [46]

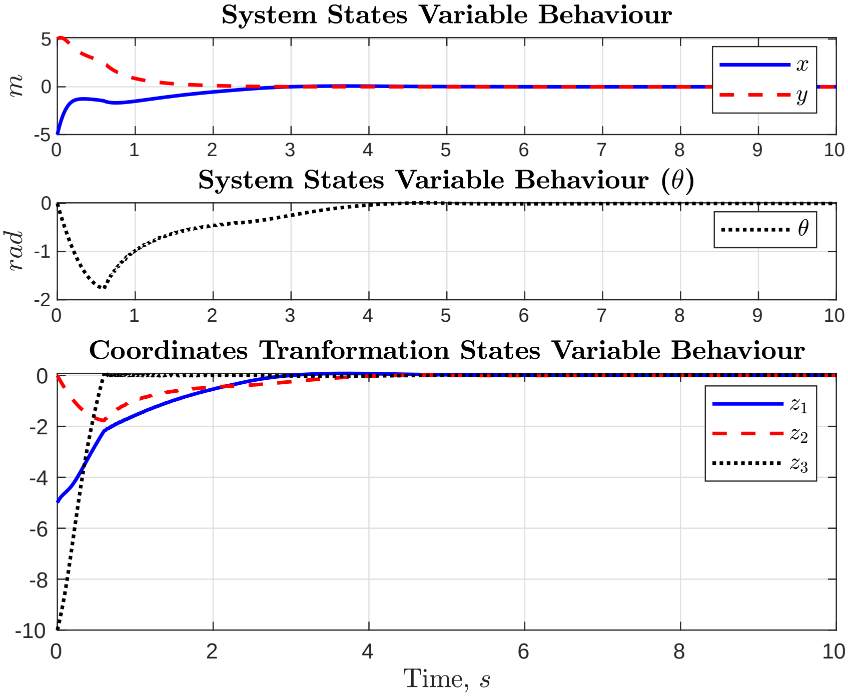


Figure 3.10: Behaviour of the system state variables for Cartesian and chained form coordinates.

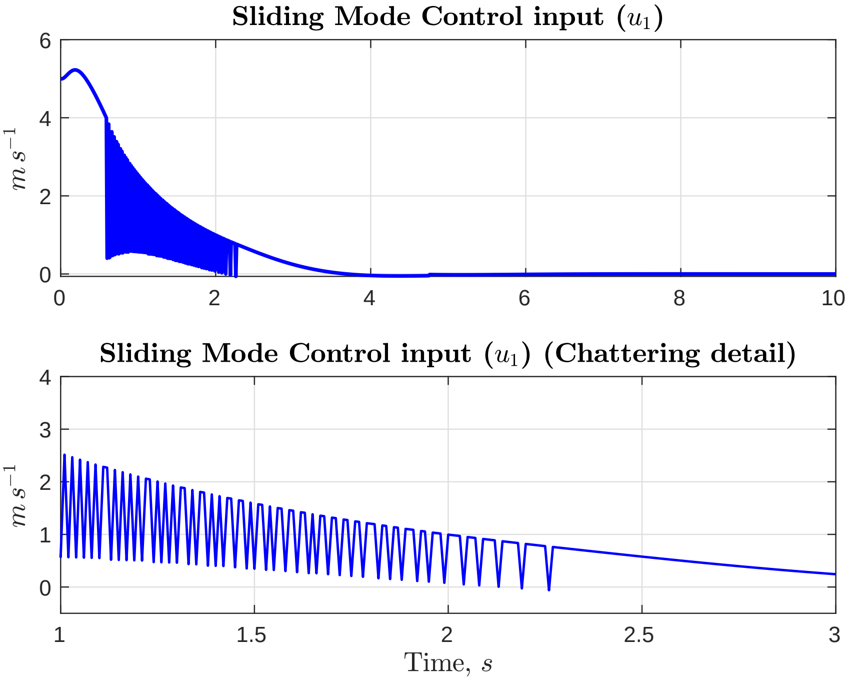


Figure 3.11: Input variable $u_1$ for chained form system. At the bottom we can see the shattering effect due to the switching effect.

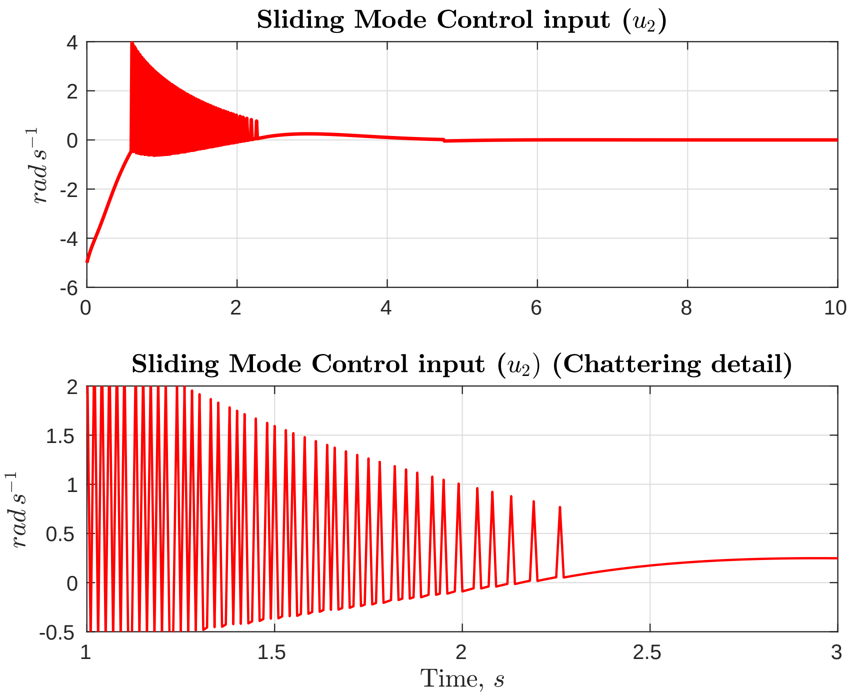


Figure 3.12: Input variable $u_2$ for chained form system. At the bottom we can see the shattering effect due to the switching effect.

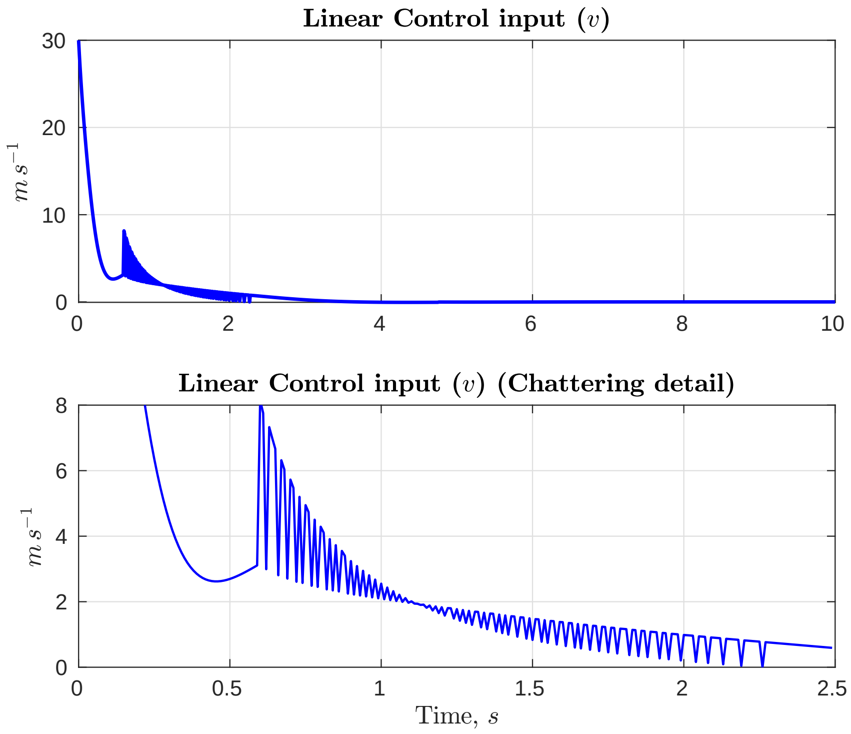


Figure 3.13: Linear velocity input variable $v$ for Cartesian system. At the bottom we can see the shattering effect due to the switching effect.

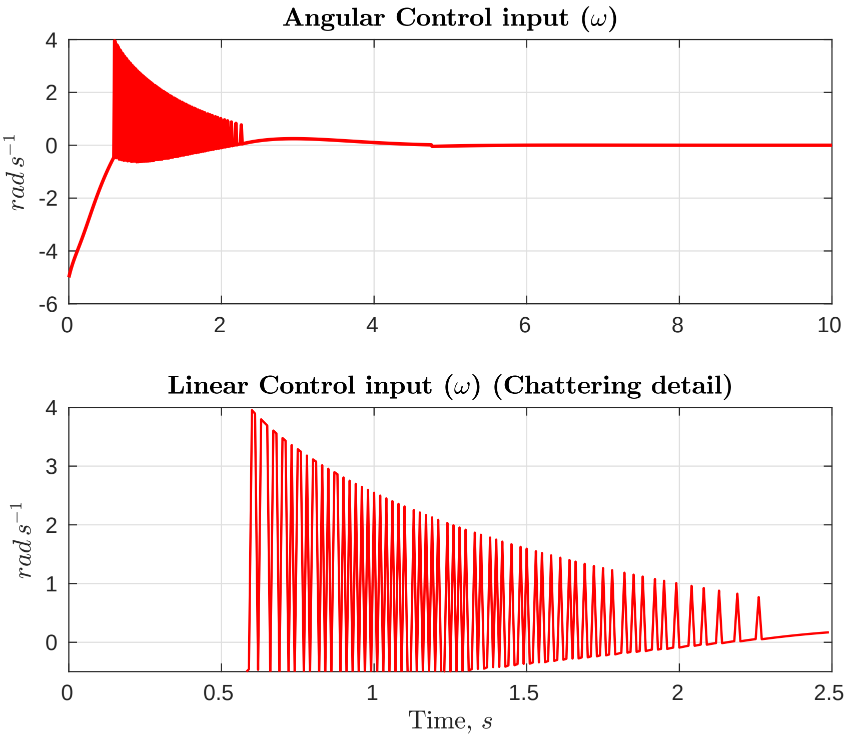


Figure 3.14: Angular velocity input variable $\omega$ for Cartesian system. At the bottom we can see the shattering effect due to the switching effect.

is the *Quasi-Sliding Mode* chattering elimination method, where the signal function sign($\sigma$) can be replaced by a sigmoid function

$$\text{sign}(\sigma) = \frac{\sigma}{|\sigma| + \epsilon}, \tag{3-26}$$

where $\epsilon$ is a small positive scalar, such that $\lim_{t\to 0}(\sigma/(|\sigma| + \epsilon)) = \text{sign}(\sigma)$ selected to trade-off the requirement to maintain the performance and ensure a smooth control action [46]. The sigmoid function for some values of $\epsilon$ can be seen in Fig. 3.15. For $\epsilon = 0$ the function becomes to the original signal function.

According to [46] the price paid to obtain a smooth control function is the loss of finite-time convergence of the sliding variable to zero. The sigmoid function provides convergence to the vicinity of the origin due to the effect of the system disturbance, which can lead to a loss of accuracy and making it impossible to proof the stability of the controller.

In order to get best convergence and chattering reduction, we select $\epsilon = 0.01$ by trial and error method for the propose application. As can be seen in Fig. 3.16 , all the state variable converge to zero in presence of same uncertainties and disturbance proposed in Sec. 3.3.1. The control signals shown in Figs. 3.17 to 3.20 do not have the chattering effect, while the Figs. 3.21 and 3.22 shown the smooth control signal without chattering effects.

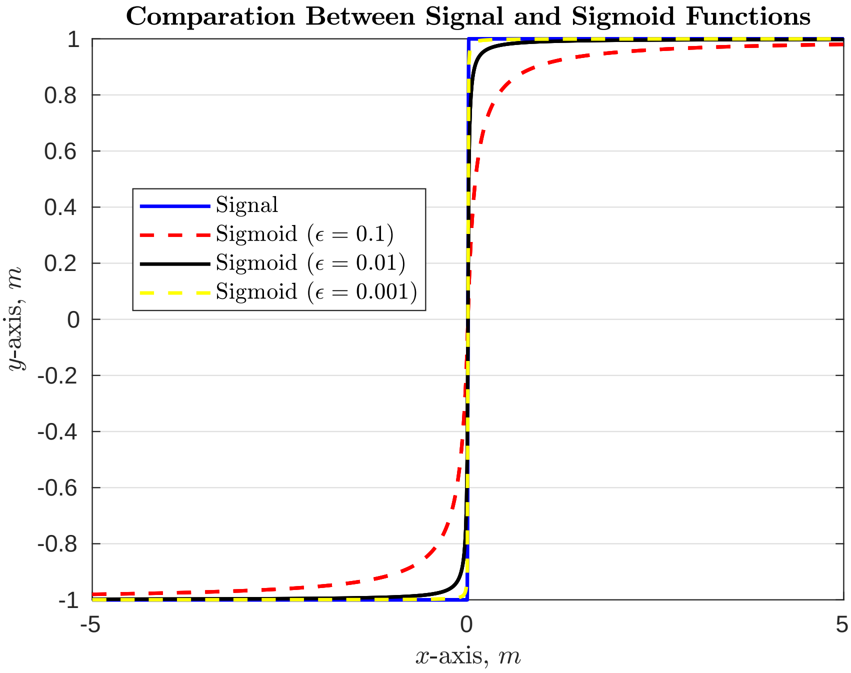


Figure 3.15: Comparison between the Signal Function and the Sigmoid Function for different values of $\epsilon$.

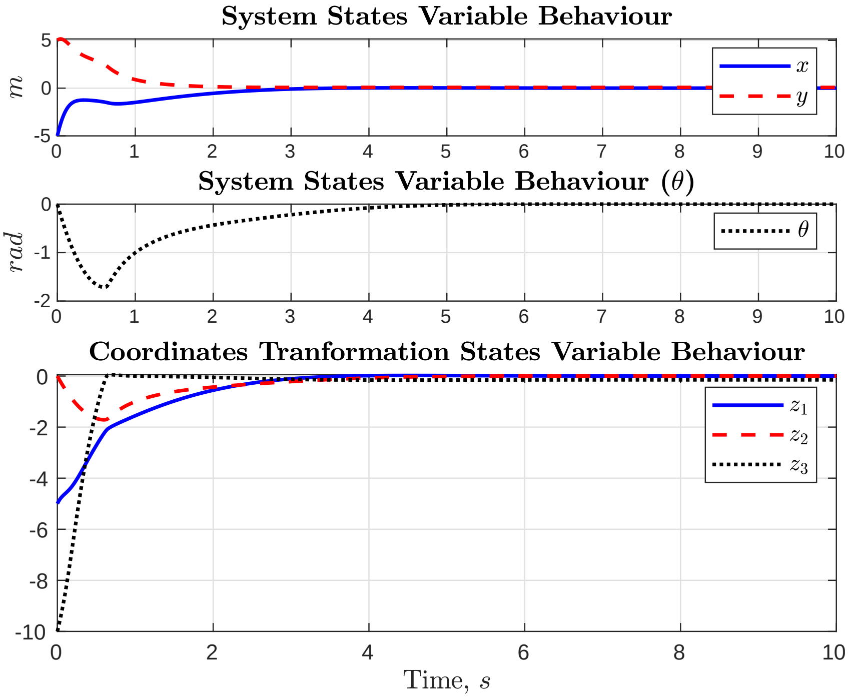


Figure 3.16: Behaviour of the system state variables for Cartesian and chained form coordinates using sigmoid function.

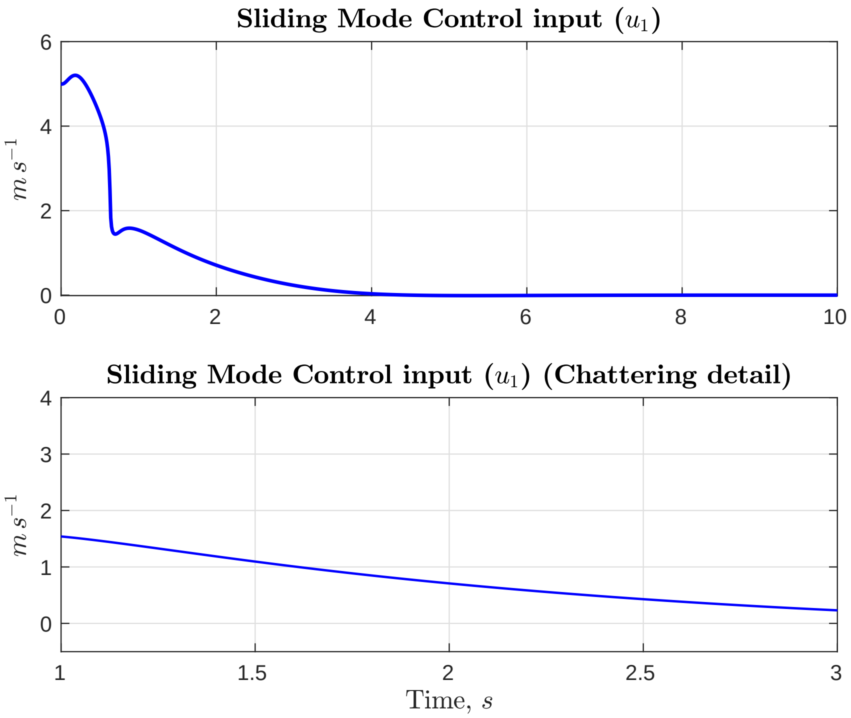


Figure 3.17: Input variable $u_1$ for chained form system. At the bottom we can see that the shattering effect was vanished due to the use of sigmoid function.

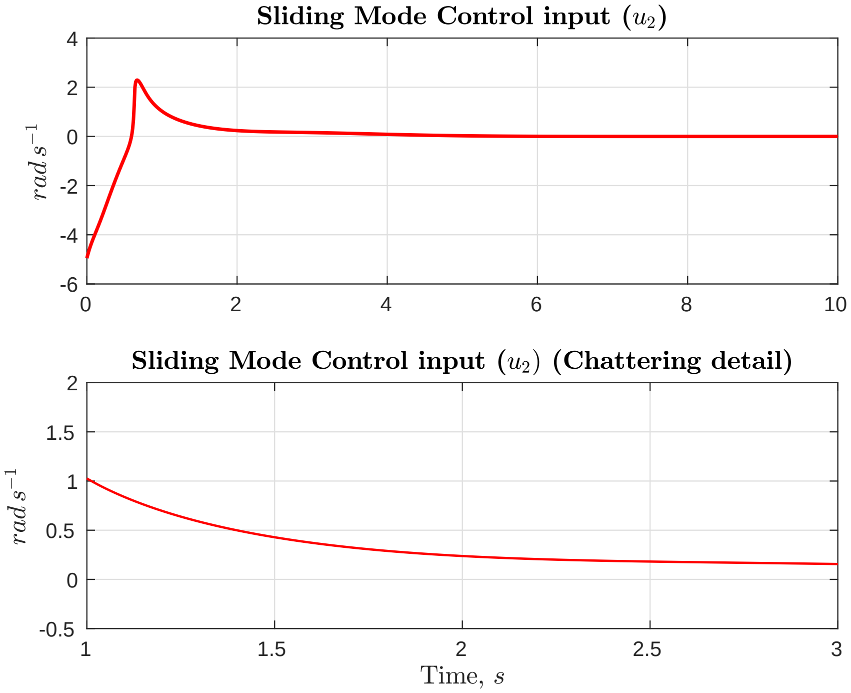


Figure 3.18: Input variable $u_2$ for chained form system. At the bottom we can see that the shattering effect was vanished due to the use of sigmoid function.

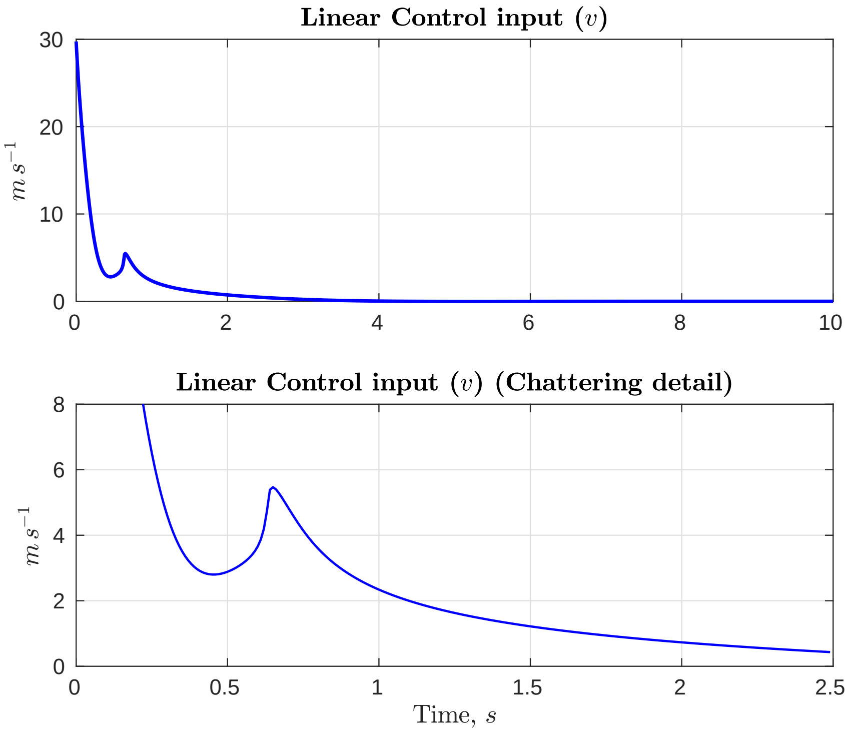


Figure 3.19: Linear velocity input variable $v$ for Cartesian system. At the bottom we can see that the shattering effect was vanished due to the use of sigmoid function.

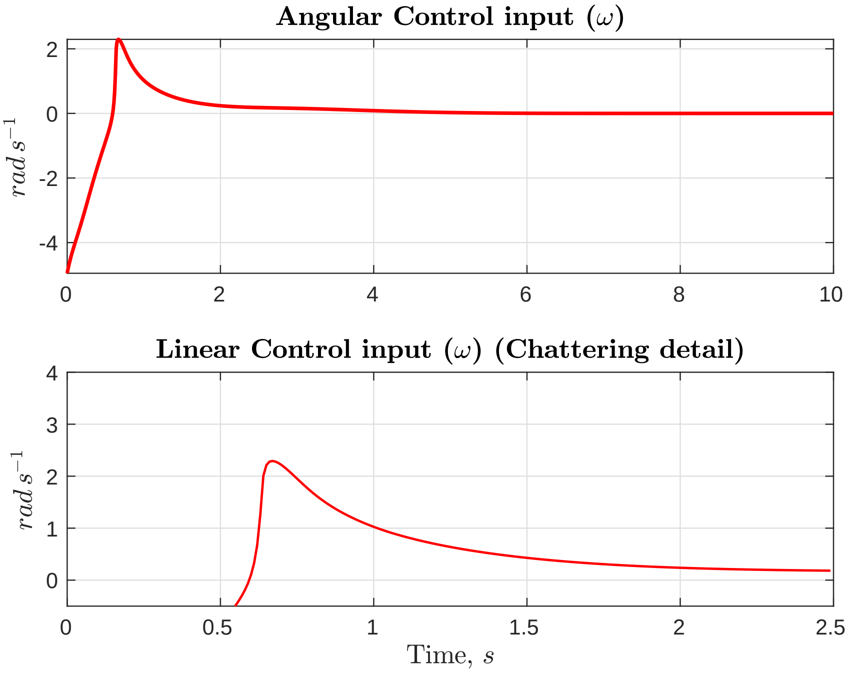


Figure 3.20: Angular velocity input variable $\omega$ for Cartesian system. At the bottom we can see that the shattering effect was vanished due to the use of sigmoid function.

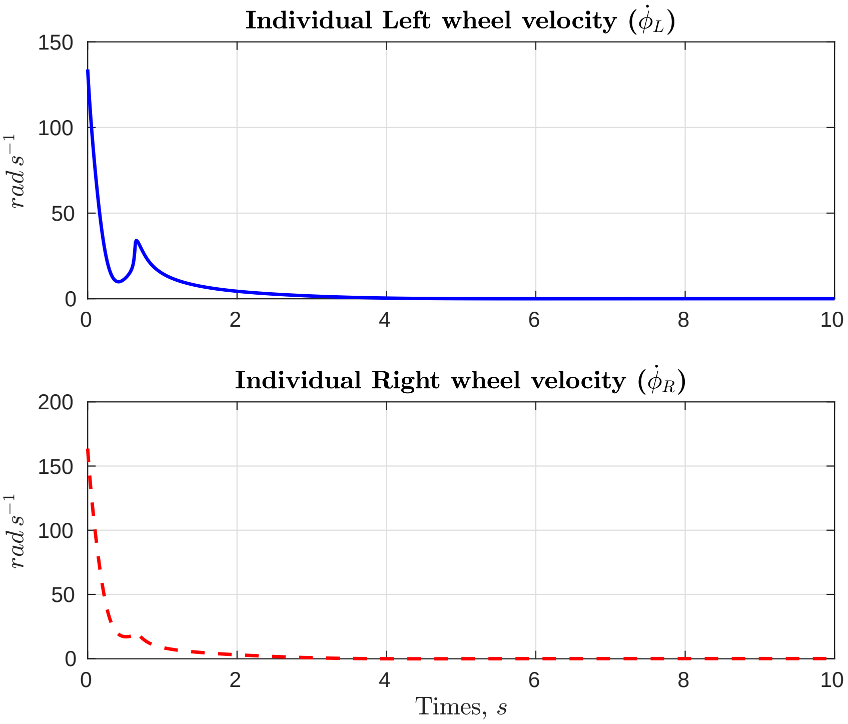


Figure 3.21: Individual velocity per wheel. We can see the smooth control signal.

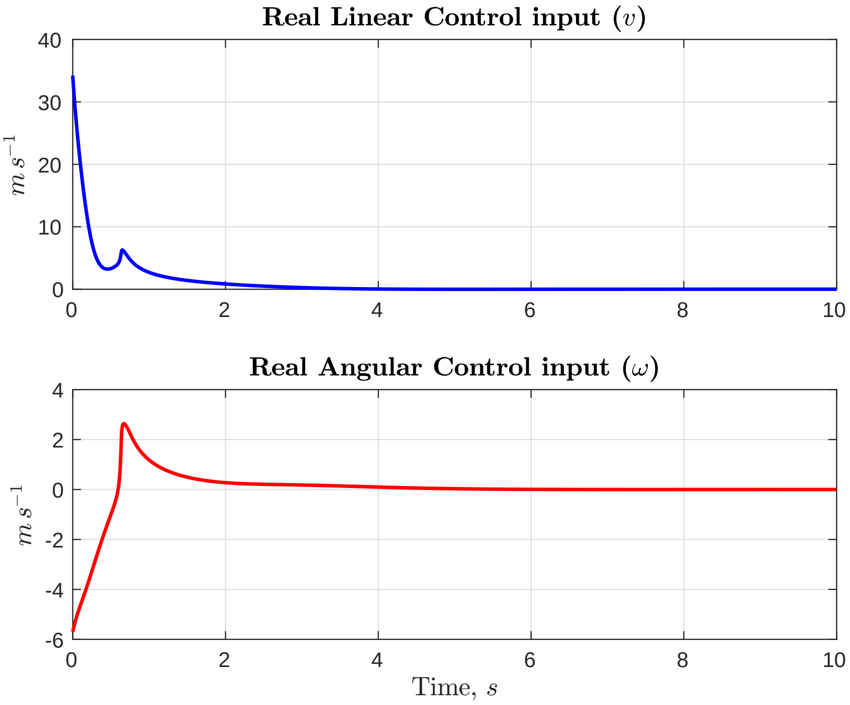


Figure 3.22: Real linear and angular velocities input for Cartesian system after conversion from wheel signal. We can see the smooth control signal.

## 3.4 Formation Control for Robot Swarm

### 3.4.1 Problem Statement

Consider that we want to perform a monitoring task in a large crop area, such as soil sample collection, pest infestation level or weed detection. This type of activity is commonly performed by a group of human workers, who inspect a sample of the total number of plants, assessing the health of the leaves, the existence of pests and the overall development of the inspected individual. With this in mind, the proposed problem is to collect images to compose a pest identification dataset using a five-member homogeneous robot group that will traverse the crop under a specific formation. This particular task comes from the demand for real-world imaging of cotton and soy farms, since for image classification algorithms a large amount of data is required.

To perform this task, we need to establish some assumption:

- Plantation needs to be minimally structured, such as flat ground and crop row alignment, which is commonly found on large-scale plantations; (Fig. 2.17);
- Agents need to have a kinematic similarity. For this case, all robots are of the differential drive model;
- Robots cannot collide with other members of the formation or with the plantation;
- The group will align outside the plantation in a predefined formation and enter the crop line, maintaining their position for as long as the task is being performed. No change in formation can take place after the mission begins;
- The formation will be guided by a predefined leader, who will dictate the velocity and direction of the group;
- There is a communication support that allows the exchange of information between robots.

Based on these assumptions, we can propose a formation control approach, which will be presented in the next section.

### 3.4.2 Swarm Structure

In the sense of the classification proposed by [55], this work is situated in the group of *intentionally cooperative systems*, in a weakly cooperative solution. The system presented is composed of a group of homogeneous mobile robots performing an extensive task (e.g. environment mapping, monitoring, data set creation) that requires task coordination periods, followed by operational independence. Its control architecture has a hybrid philosophy, combined with high level local control, allowing the definition of global objectives. To achieve this, we use the concept of networked robots, where coordination and cooperation take place over a network connection, according to IEEE definitions. [55].

Based on the definitions proposed by [55] and [56], the proposed $\Sigma$ dynamic mobile robot network is a group of $N$ agents, each of which is characterized by a unique identifier $i \in I = 1, 2, , \ldots, N$, with a state vector $x_i \in X_i \subset \mathbb{R}^n$ and a control input vector $u_i \in U_i \subset \mathbb{R}$, related by a $f_i : X_i \times U_i \to TX_i$, specifying the dynamics function: $\dot{x}_i = f_i(x_i, u_i)$. This group shares a $\mathcal{Q}$ configuration space and has sensing and communication graphs, which can be formed statically or dynamically, depending on the physical proximity of a pair of agents. In this case, we have a number $N = 5$ (five) agents, using a state vector $x_i \in \mathbb{R}^3$, a control entry $u_i \in \mathbb{R}^2$, a system dynamics provided by (2-5), and only one communication graph is used to exchange information between agents.

The communication support for implementing the robot network was based on the robotic framework ROS - robotic operating system [1]. This system allows the unique identification of the robot, the definition of the formation format and the creation of the communication graph. The ROS (or just ROS) framework has been used in a number of researches applied to swarm control, such as surveillance, locating and tracking objects [57], expanding telecommunications network in disaster areas [58] and indoor micro drone swarms [59]. Using ROS, all robot information can be shared across the network, including telemetry, position, control signal, or sensor data. This information is used to maintain training and avoid collisions with other teammates.

The proposed swarm control approach, that can be developed using tools provided by the framework ROS, aims to maintain group formation during crop line displacement. To perform this task, we propose the use of an individual reference frame strategy, in which each agent is connected to a specific reference

[1] `http://www.ros.org/`

frame. Then, This frame is then connected to the leading robot, which can be given a desired path to follow, an overall goal to be achieved or remotely controlled by a human operator. Because a physical leader is subject to external risks, the key idea is the use of a virtual robot as leader. With this approach the system becomes more reliable and robust due to the fact that a virtual robot can not suffer physical damages due to environmental conditions or intensive use.

As of [56], a formation is a $N$ vehicle network, interconnected through a collection of $S = s_{i\,j}$ nodes, specified by:

$$s_{ij}(t) := \left[\ell_{ij}(t) \quad \psi_{ij}(t)\right]^T, \tag{3-27}$$

where $i, j \in I$ i a pair of robots, $\ell_{ij}$ the separation distance between them and $\psi_{ij}$ its bearing angle. Using these concepts, we can create a new coordinate system, attached and referenced to a leader, where each frame has a pose (distance and bearing angle) relative to it. Each member of the formation, which is referenced to a common world or global reference, is correlated to a specific position in coordinate system of the formation through a coordinate transformation structure, which produces a transformation tree between the global reference and the formation coordinate system. Fig. 3.23 shows the transformation tree, where can be seen the global reference and the formation structure attached to the formation leader (in this case, the robot #1 `agricultural_01`).

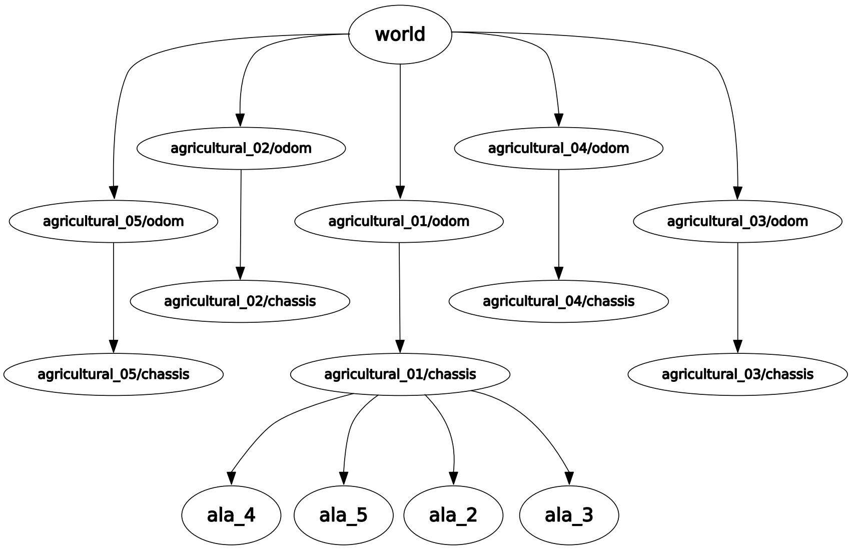


Figure 3.23: Communication chart of the system, where you can see the transformation structure with global and local references.

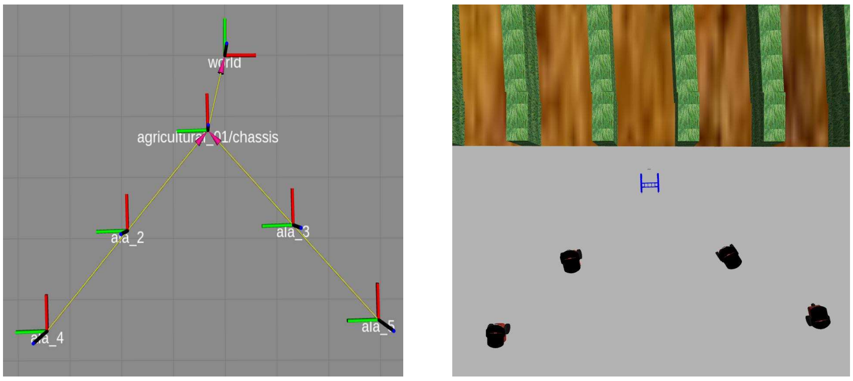
A global formation control system enables group management from a central point, which can be a human supervisor or other management software.

With this approach, the format of the formation can be changed during operation by using a specific control publication. Also, because the architecture uses a team leader, the overall goal needs to be communicated to only one agent. The next section will explain the formation control approach used to maintain group formation.

The formation control proposed by [56] can be considered a coordinate transformation towards polar coordinates, where the distance between teammates and the final orientation of the robot can be obtained based on the concepts presented in Sec. 2.3.2.

Thus, we want that the group perform displacement while maintaining a predefined formation using a networked robot group. All members of the group have their position related to a global reference, and each one is related to a specific formation position, called $ala_i$, with $i \in I = \{1, .., 5\}$. These positions were defined relative to the leader, with a fixed distance $\ell_i$ and a rolling angle $\psi_i$, published as a transformation topic (`\tf`) over the network.

At each robot's internal control node a coordinate transformation is performed between its current position and the reference frame. This operation generates a pose, which can be used as a goal for a posture regulation task using the polar approach. Since all five robots are similar, the same type of node can run on all robots. To do so, simply modify your identification, which should be unique in the group. In this way the robots will take their own position in the desired formation, as shown in Fig. 3.24



$(a)$ $(b)$

Figure 3.24: (a) Frame structure in "V-shape" formation viewed in Rviz; (b) robots performing the formation in Gazebo simulation.

## 3.4.3 Collision Avoidance System

The Collision Avoidance System (CAS) is used to prevent collisions with other teammates and keep the robot in line with the crop line by working in two different ways, using different detection systems. The CAS acts on the linear and angular velocities to prevent frontal collisions and maneuvers to avoid lateral collisions. This system uses the artificial potential function approach, defined by [60] and [41], as a total potential obtained by overlapping repulsive ($U_r$) and an attractive ($U_a$) potentials, given by:

$$U_t(q) = U_a(q) + U_r(q). \tag{3-28}$$

To create attractive potential, we used the distance to the goal ($\rho$ in 2-15), which is always positive and zero at $q = q_g$. The repulsive potential was created using the current position of the other robots in the formation. The communication graph provides all data about teammates. Using this data, the distance and bearing angle can be calculated to compose an influence vector, used to create a potential barrier around each robot that is at a limit distance. From this idea we can propose [60]:

$$U_r = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{D(q)} - \frac{1}{T^*}\right)^2, & D(q) \leq T^*, \\ 0, & D(q) > T^*, \end{cases} \tag{3-29}$$

where $D(q) \in \mathbb{R}$ is the distance between robots, $T^* \in \mathbb{R}$ is the limit of influence and $\eta \in \mathbb{R}$ is a gain in the repulsive gradient. Due to the fact that the system has an angular coordinate, the Khatib and Chaatila [42] Extended Potential Field method was used. The repulsive function takes into account the orientation of one robot relative to another, which reduces repulsive force and corrects alignment by performing the deviation maneuver when applied to the angular control input. Thus, the potential repulsive function is given by:

$$U_r(q) = \sum_{i=1}^{n} U_{r,i}(q)\,, \tag{3-30}$$

with $n \in \mathbb{R}$ is the number of robot in the formation. To keep the robot aligned to the crop row and prevent collision with plants is used a Image Based Visual Servoing Control (IBVS Control) presented in Sec. 3.2. Based on this premises, the follow control law can be proposed that allow both the formation control and the obstacle avoidance:

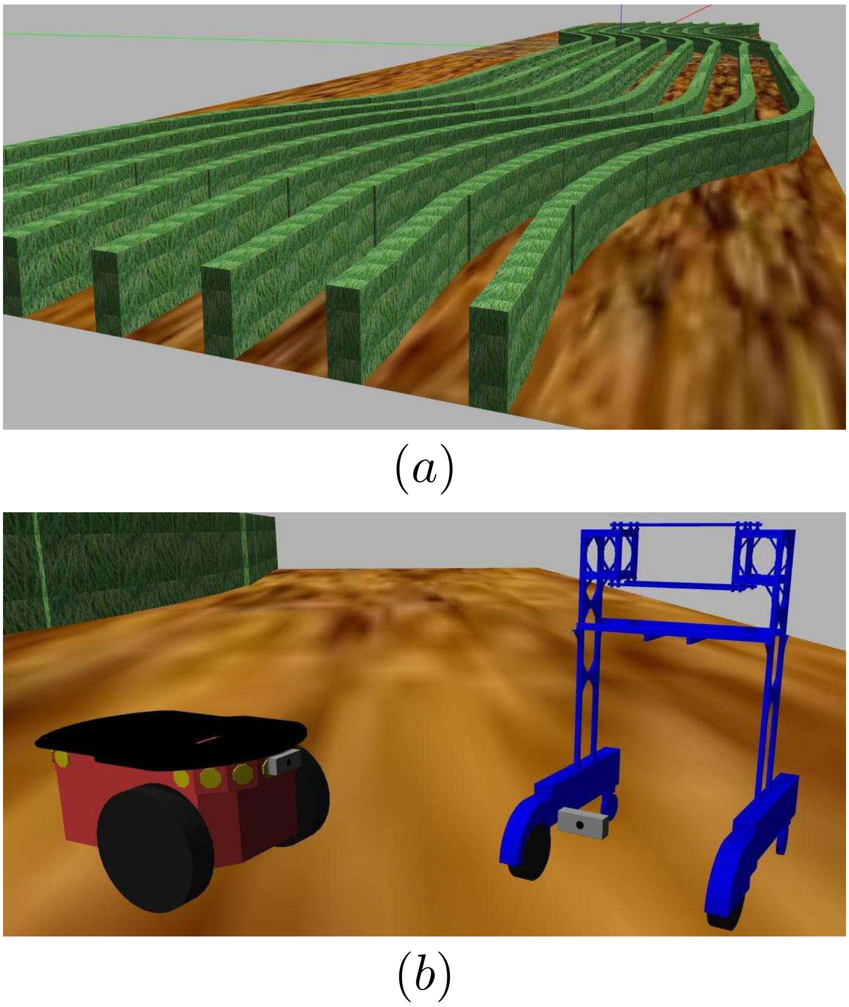
$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} U_{t_{linear}} \\ U_{t_{angular}} + \omega_v \end{bmatrix}, \tag{3-31}$$

where $U_{t_{linear}}$ is the total potential obtained by (3-28) guiding the robot to the goal, $U_{t_{angular}}$ is the total potential obtained by (3-29) applied to the orientation and $\omega_v \in \mathbb{R}$ is the control signal provided by the visual servoing control. The proposed experimental setup for validate this approach is presented in next section.

### 3.4.4 Simulation Tests and Results

This section provides the experimental tests performed to validate the proposed approach. All proposed experiments were performed in the Gazebo robot simulator virtual environment, integrated with the ROS (Robot Operating System) framework, described in detail below.

The experiments were performed in Gazebo environment, version 7, together with the ROS framework, Kinetic Kame version. A 2000 $m^2$ (100m x20m) agricultural field simulating the actual color of the soil was simulated. To simulate the crop rows, we used vegetation patterned cubic structures, arranged along the field, composing a set of seven lines. The upper side of Fig. 3.25 shows the proposed virtual environment used in the validation tests.



$(a)$

$(b)$

Figure 3.25: The proposed virtual testing environment:crop row (a); (b)the robots Pioneer 2DX (left) and the SoyBot (rigth).

We use two types of robot models for the experiments. The leading robot was inspired by the SoyBot robot itself (Fig. 3.25 (b) right), scaled for simulation reasons. This robot has been defined as the team leader, navigating through the crop lines with constant linear velocity using his on-boar camera via subscribed topic `/camera_1/image_raw` to perform the line following task, with the proposed controller in (3-2). This robot does not use any position information from other group members and is unable to avoid collisions with other robots. Followers were inspired by the Pioneer 2DX commercial model (Fig. 3.25 (b) left), which is available on the simulator as standard. Each of these robots can perform regulation tasks, used to reach their position in formation, avoid collisions with other teammates, and follow the lines when inside the plantation using computer vision. To perform these tasks, the agent knows the current position of the teammates by subscribing to each other's topic `/odom`, their position in formation, defined by the topics `/tf` and `/tf_static` and use the camera image via the topic `/camera_1/image_raw`. This outline can be seen in Fig. 3.26.

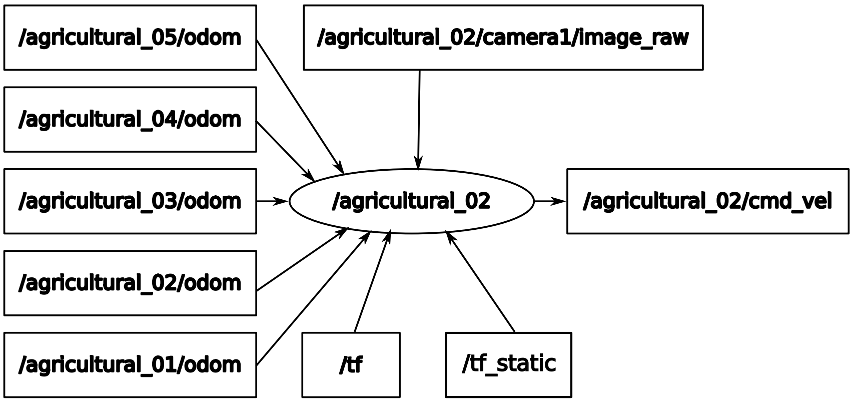


Figure 3.26: Active nodes used for robot guidance.

The first task proposed consists of a formation task, in which the group needs to perform an alignment in a predefined form, such as row (in-line), column and V-shape. The first in formation displacement is started from random initial conditions and directed to the desired in line formation. The initial condition for the next formation is the previous state and no obstacles were placed on the field in this test. Two laps was performed, named *Lap 1* and *Lap 2*. The Fig. 3.27 shows the displacement in each formation, while Figs. 3.28 and 3.30 shows the control signal ($v$ and $\omega$) for each displacement and the results are shown in the Figs. The purpose of this test was evaluate the formation control for different conditions using the proposed polar control approach, along with the Collision Avoidance System to avoid collision with other robots.

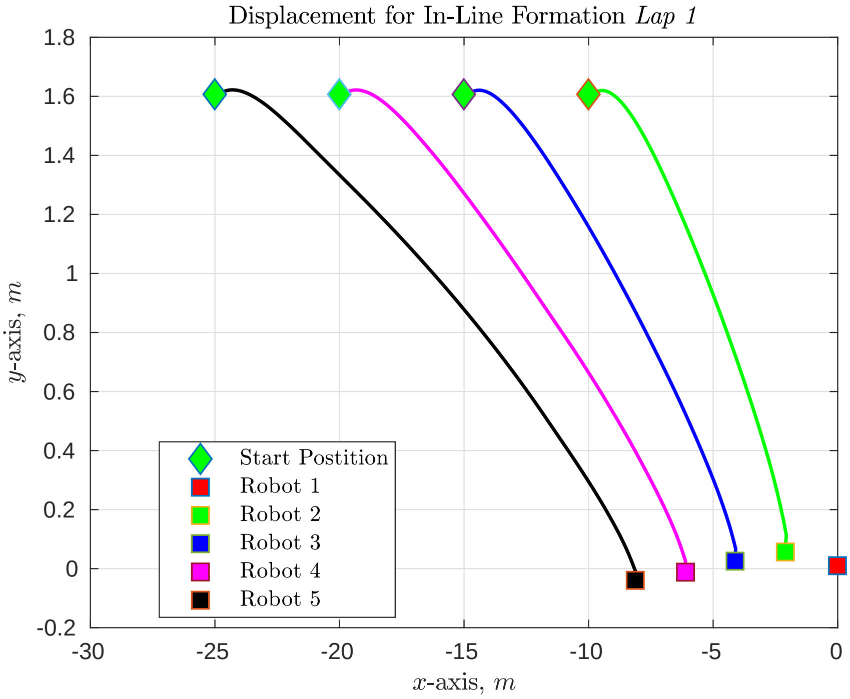


Figure 3.27: In-line formation for *Lap 1*. At the beginning of the formation, the leader (red) remains stationary while the other members assume their positions.

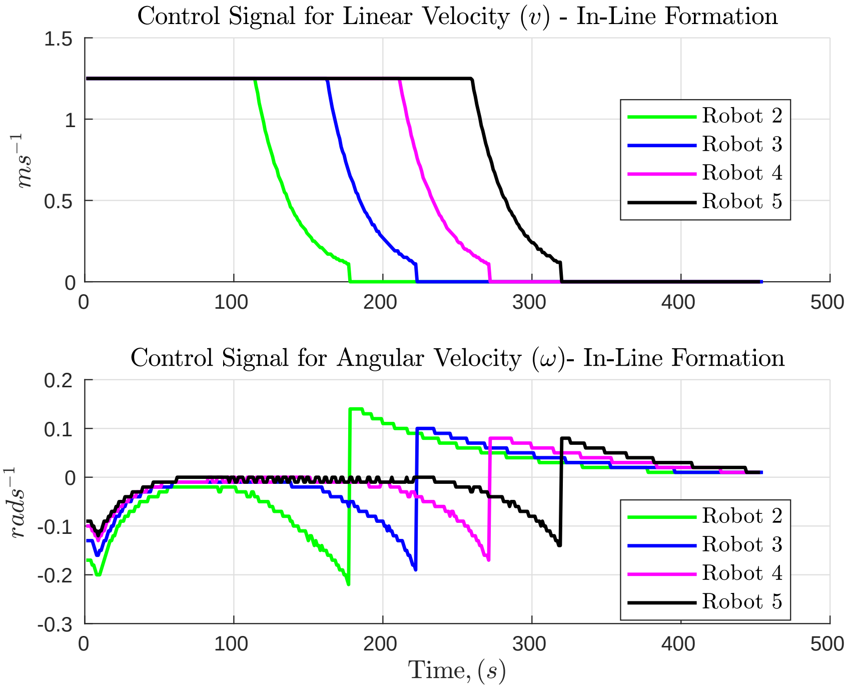


Figure 3.28: Control Signals for in-line formation for *Lap 1*.

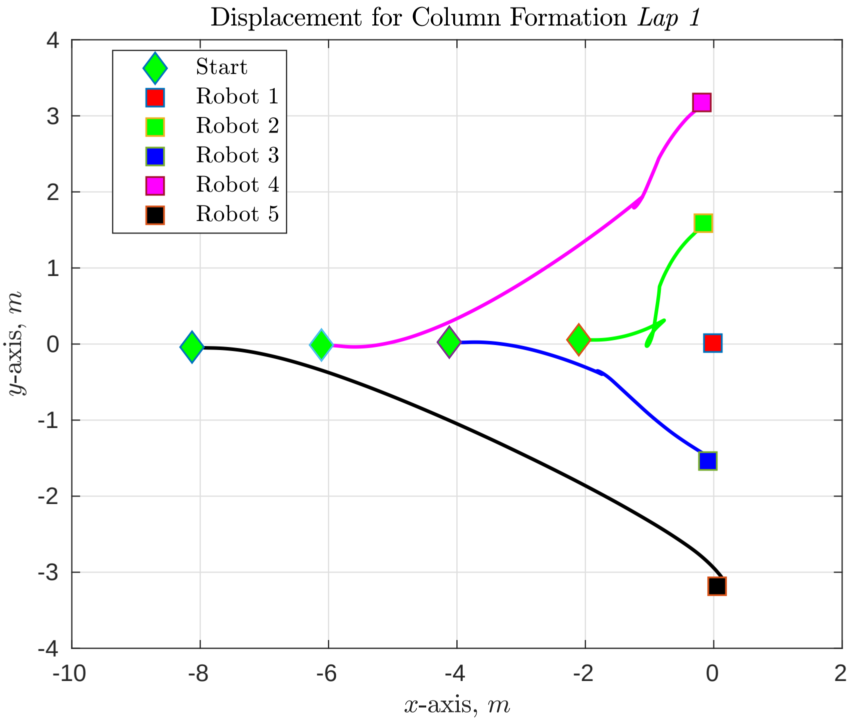


Figure 3.29: Column formation for *Lap 1*.

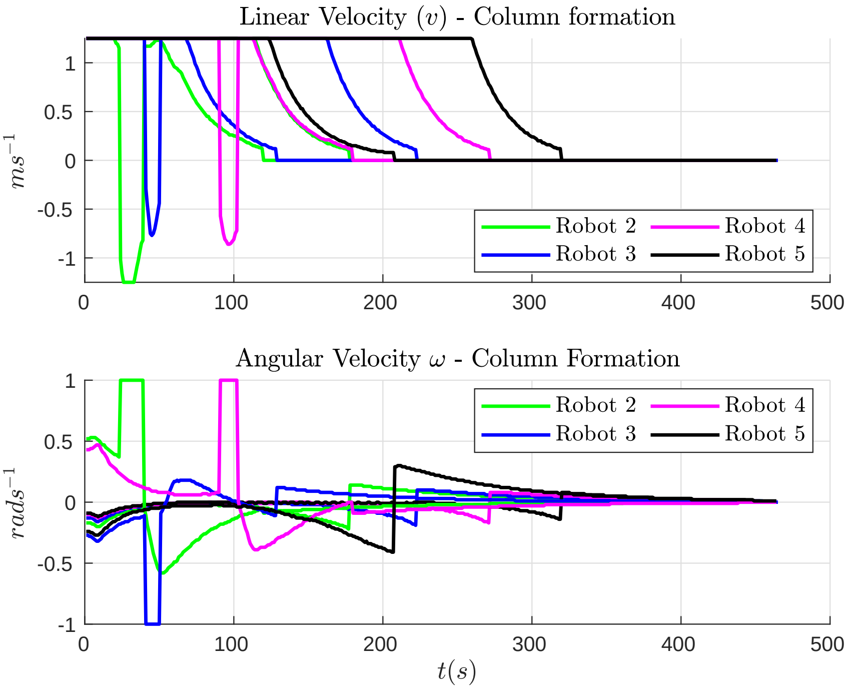


Figure 3.30: Control Signals for column formation for *Lap 1*.

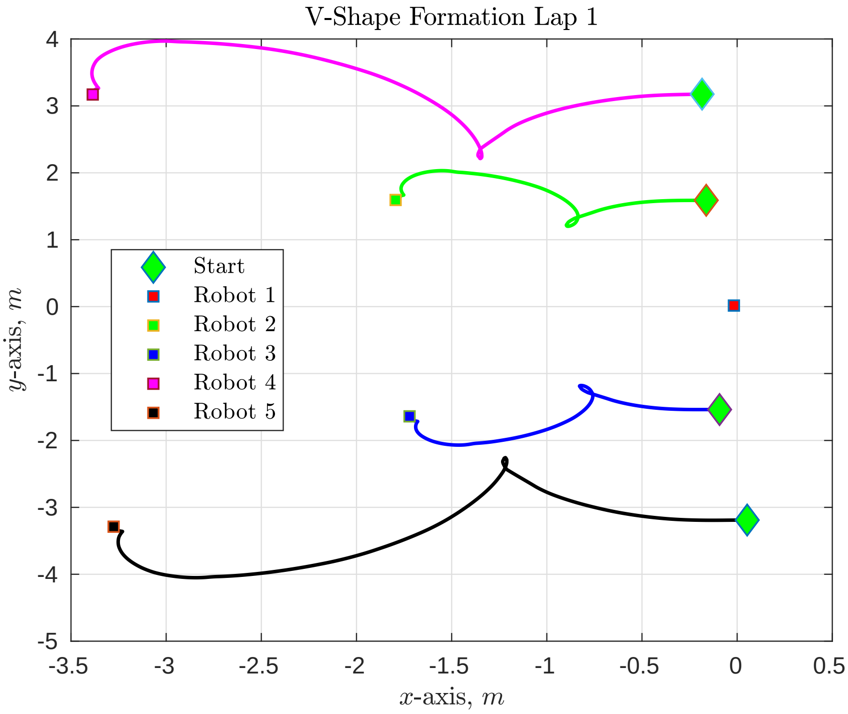


Figure 3.31: V-Shape formation for *Lap 1*.

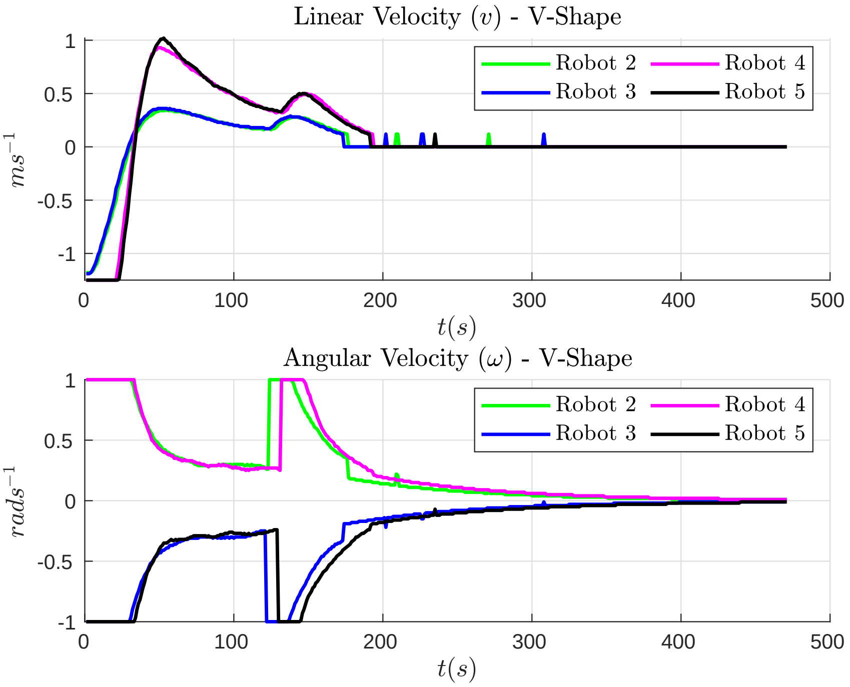


Figure 3.32: Control Signals for V-Shape formation for *Lap 1*.

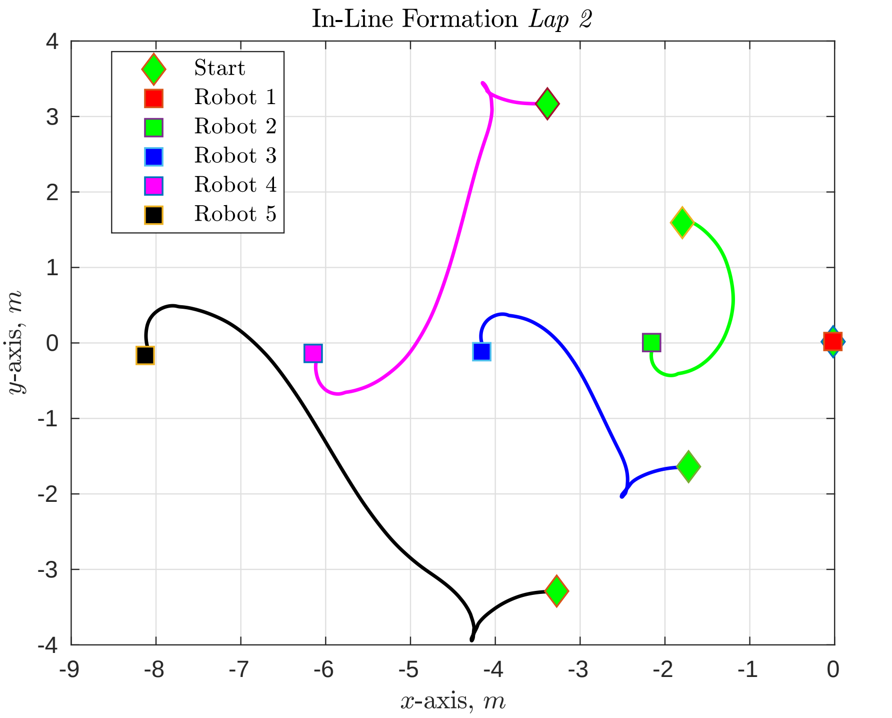


Figure 3.33: In-line formation for *Lap 2*, from V-Shape formation.

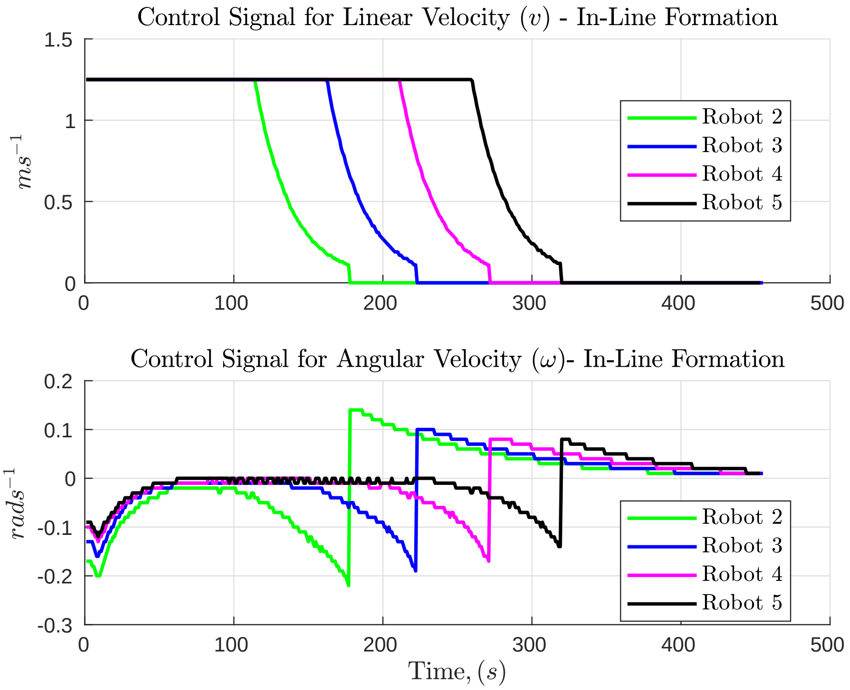


Figure 3.34: Control Signals for in-line formation for *Lap 2*.

### 3.4.4.1 Displacement in Formation

This second test consists of a group movement, in which a fixed linear and angular velocity is applied to the team leader and teammates must follow him at the same velocity. Robots are started from an initial column formation condition and must achieve V-shape during movement, maintaining distance and position during travel. The lead velocities for the first experiment were $0.35\,m\,s^{-1}$ and $0.1\,rads^{-1}$ for linear and angular velocities respectively, describing a circular path; in the second experiment, a linear velocity of $0.35\,m\,s^{-1}$ and angular velocity 0 was applied to the team leader, describing a straight path. Again, no obstacles were placed in the field test. The objective of this test was to validate the formation movement approach, its ability to maintain formation during a displacement and the CAS action in formation movement tasks. The displacement of the group can be seen in Fig. 3.35 (a) for circular trajectory and (b) for straight path. Control signals for both trajectories are shown in Fig. 3.36 from (a) to (d).

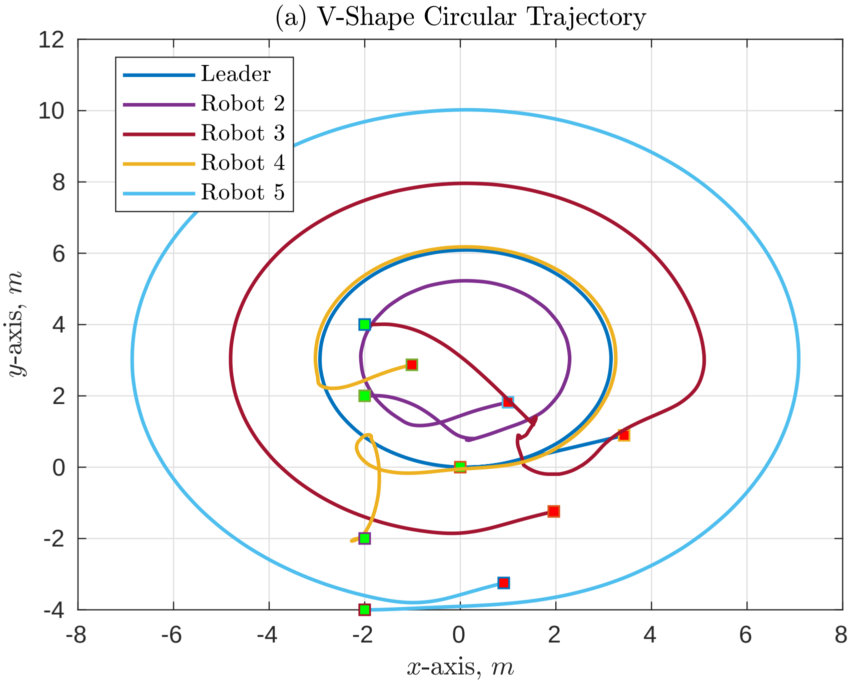


Figure 3.35: Formation displacement in motion in a circular movement.

### 3.4.4.2 Displacement into Crop Rows

This last test consists of the movement of the group in the crop lines, where the shape of the formation needs to be maintained during the movement, while the robot remains aligned to the center of the crop row, avoiding

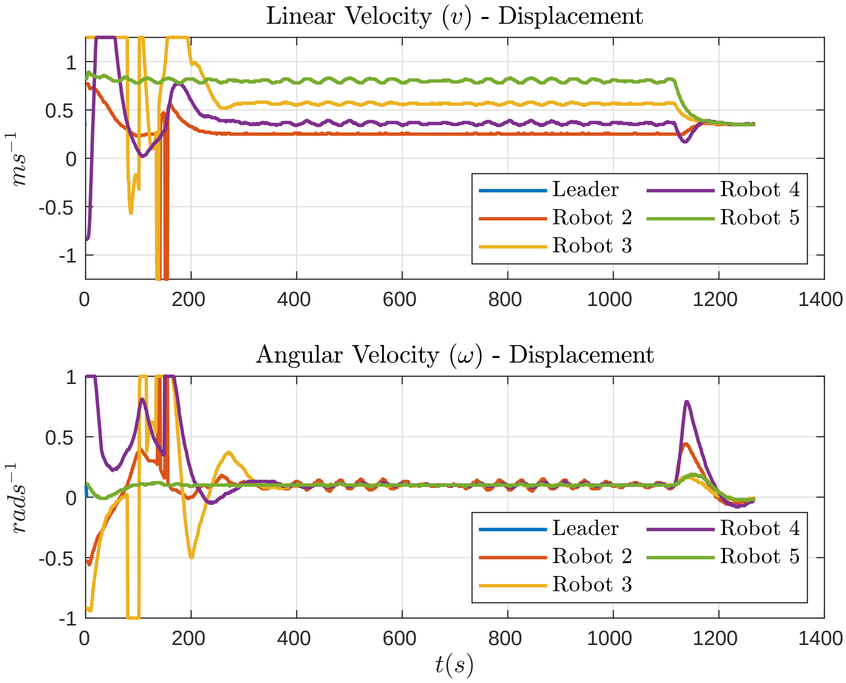


Figure 3.36: Control Signals during circular movement for linear and angular velocities.

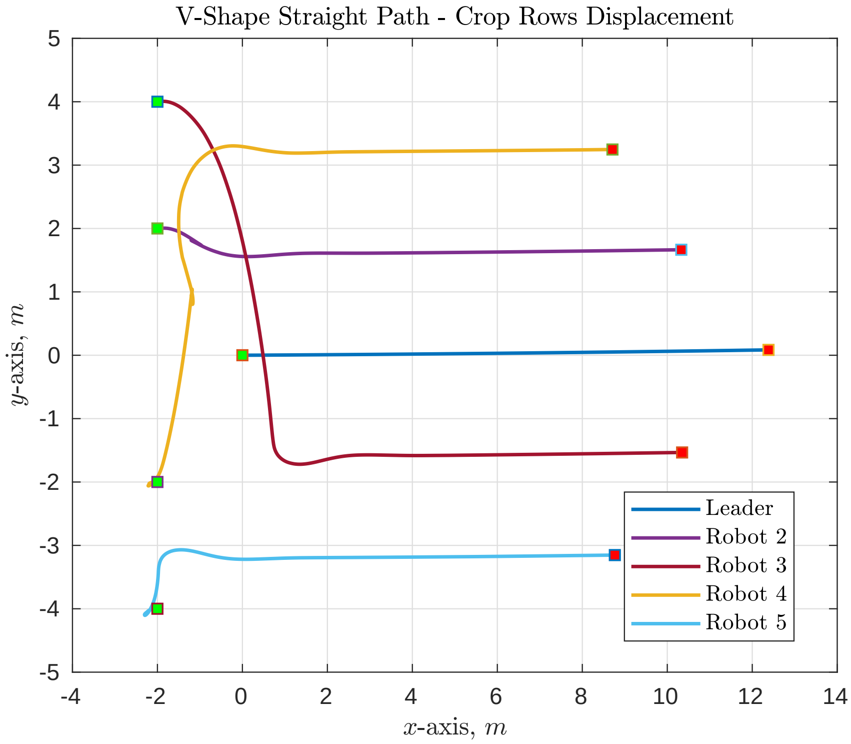


Figure 3.37: Formation displacement in motion in a straight line movement.

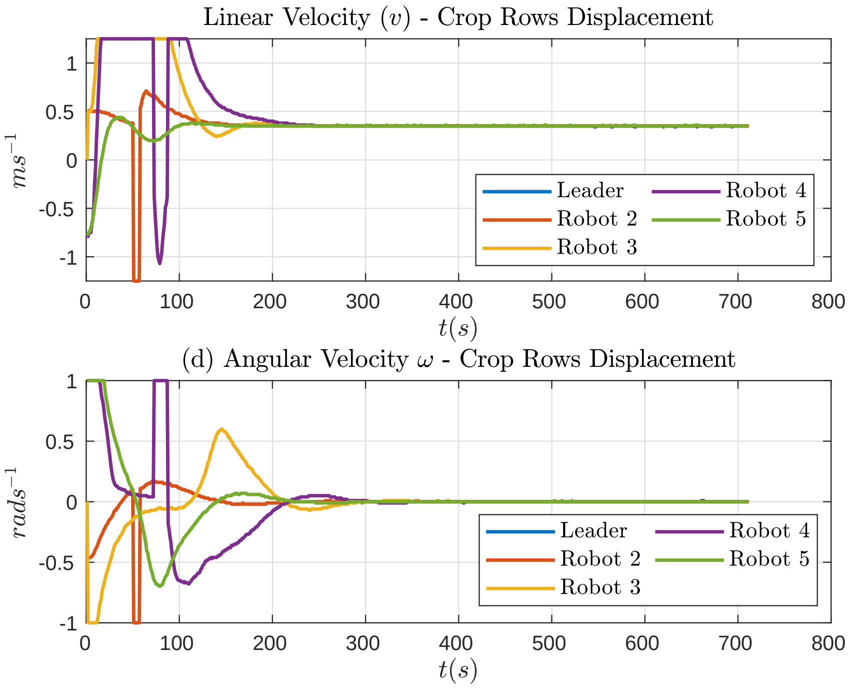


Figure 3.38: Control Signals during displacement straight line movement for linear and angular velocities.

collisions with the plantation. For this task, the group leader enters the plantation using visual servoing control to correct his orientation towards the plantation, moving at a constant linear velocity of $0.35ms^{1}$. Followers need to perform the same task, using CAS to avoid collisions with the crop and other robots, maintaining the shape of the formation until the end. In addition to the assessments proposed below, this test aims to validate the prevention of crop collisions and the proposed alignment behavior in the latter part of the Sec. 3.4.3. The Fig. 3.39 shows the robots displacement during the task and the Fig. 3.40 shows the control signal of all the robot and Fig. 3.41 shows the formation evolution from initial random position to displacement in formation into the plantation.

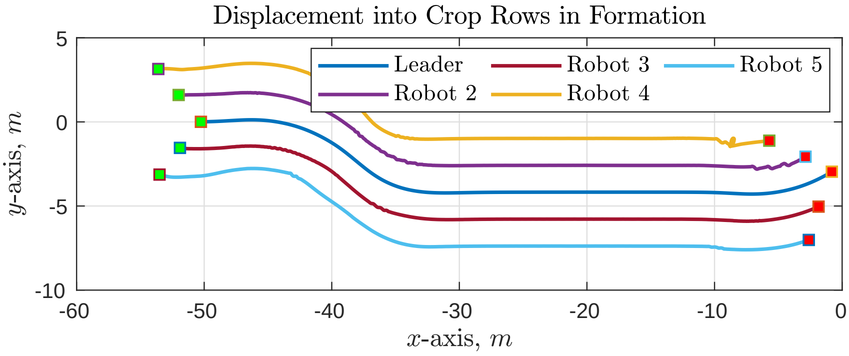


Figure 3.39: Formation displacement into the crop rows. The leader (center) and the Alas.

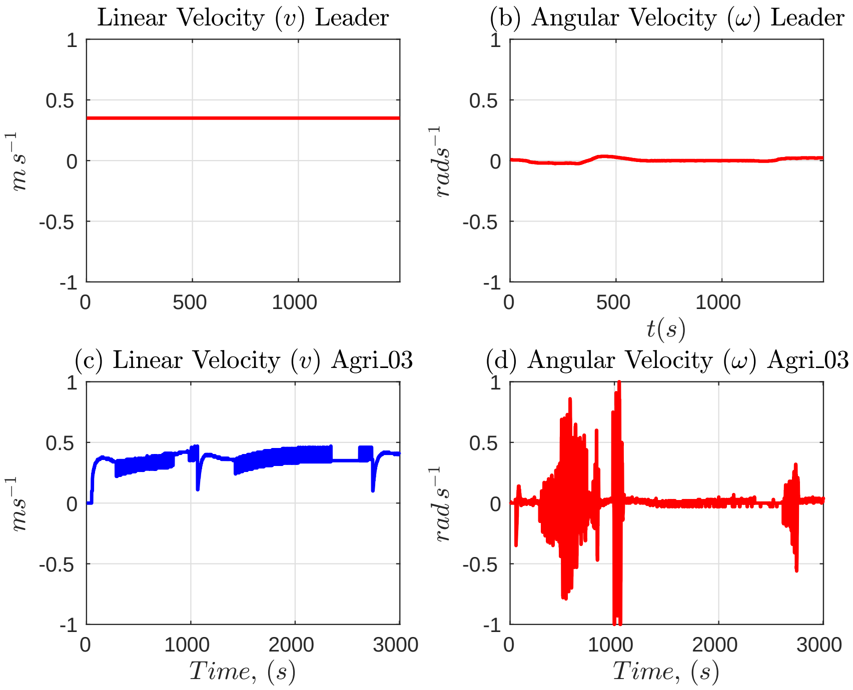
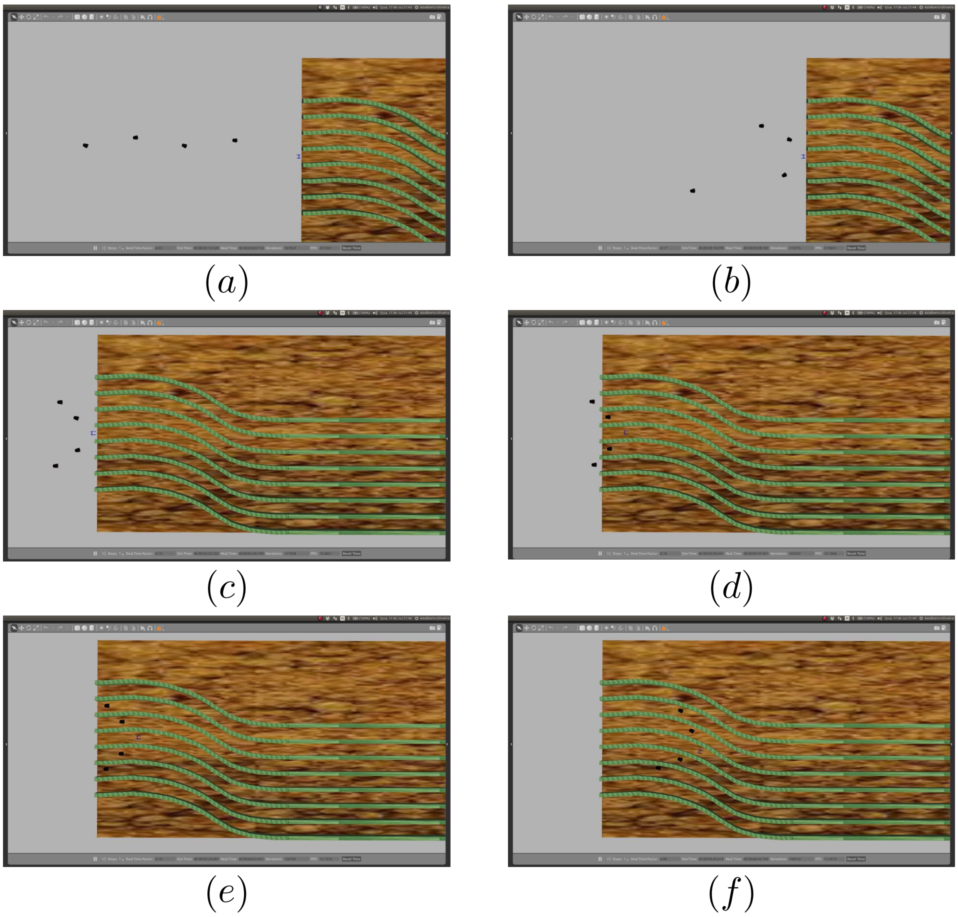


Figure 3.40: Control Signals during displacement: (a) and (b) team leader; (c) and (d) Ala_3 (Agricultural_03).


$(a)$ $(b)$

$(c)$ $(d)$

$(e)$ $(f)$

Figure 3.41: Evolution of the formation: (a) initial condition; (b) starting formation; (c) robots in formation; (d) to (f) penetration into the plantation.

## 3.5 Experimental Results

In this section we present the experimental results for visual servoing control applied to the differential driver mobile robot in a real environment. The experiments were performed in a cotton farm, which is a structured environment as described bellow, with flat terrain and aligned row crops with a clear path between them (Fig. 2.17).

The robot used for the tests, the SoyBot (presented in Section 2.4), was developed by PUC-Rio in partnership with Solinftec [2]. The kinematic model of the robot is equal to the differential-drive robot and unicycle models described Sec. 2.2. As previously mentioned, the robot is equipped with three WebCams Logitech C270 HD (720p and 30 fps) for image-based navigation, whose frames are aligned to the robot frame $\mathcal{F}_r$: one on top, pointing to the ground, and two above each wheel drive, pointing forward.

The results correspond to one lap, with elapsed time of 433 seconds (about 7 minutes) and traveling distance of approximately 30 meters. At the beginning of the experiments, both world and robot frames, $\mathcal{F}_w$ and $\mathcal{F}_r$, are assumed to be coincident and the initial robot configuration $q(0)$ is assumed to be zero.

Since the $x$-axis of the robot frame $\mathcal{F}_r$ is not aligned to the row crops, the mobile robot needs to perform alignment maneuvers during the navigation task. Figures 3.42(a)-(f) show the behavior over time of such maneuvers from the robot point of view by using only the camera mounted on top of the robot structure, with a narrow field of view (FoV), which allows only the row crop visualization.

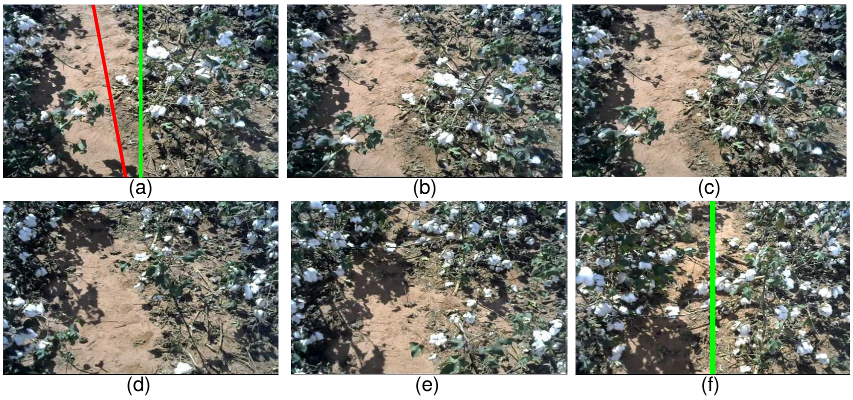


Figure 3.42: Top view of the alignment maneuvers performed by the SoyBot robot.

[2] https://solinftec.com/pt-br/

This behaviour can be seen in Fig. 3.43(a) wherein the robot moves along the row crop in a straight line with heading angle around 0.2 $rad$, as shown in Fig. 3.43(c). As expected, the robot was able to perform the autonomous navigation task following the rows satisfactorily.

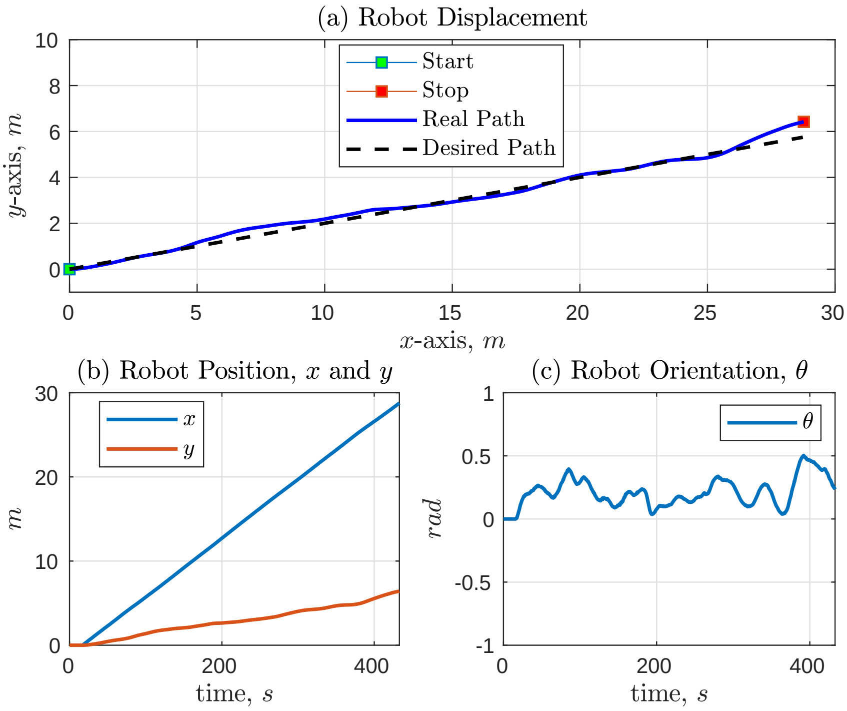


Figure 3.43: (a) Robot displacement; (b) robot position in $x$- and $y$-axis; (c) robot orientation, $\theta$.

Fig. 3.44 shows the control signals $(v, \omega)$ applied to the robot and its estimated velocities based on the readings of the wheel encoders. In Fig. 3.45, it can be seen the velocity commands applied to each wheel drive controller and the estimated wheel speeds based on the encoder readings.

## 3.6 Concluding Remarks

In this chapter, we have proposed a control system for a differential-drive mobile robot by using an image-based visual servoing approach, which is able to autonomously drive the vehicle through the row crops. We also have presented the robot modelling and an image segmentation technique based on a color space transformation with an adaptive threshold binarization method. Numerical simulations show the feasibility of the SMC approach for tracking tasks. The Lyapunov control theory was used to project the controller for tracking problem based on visual servoing and to proven its stability. Experimental results, obtained with the SoyBot robot navigating in

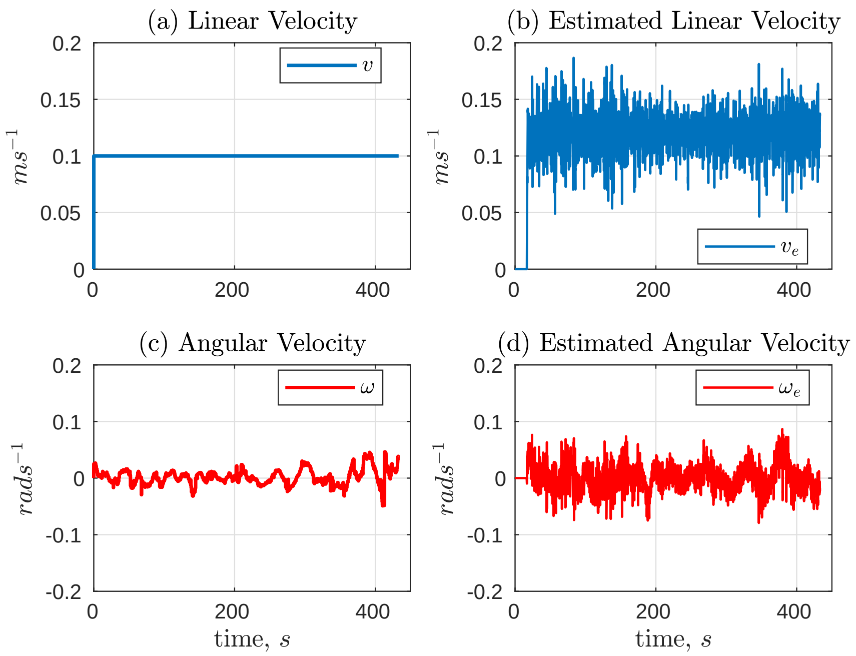


Figure 3.44: (a)-(c) Linear and angular velocities; (b)-(d) estimated linear and angular velocities.

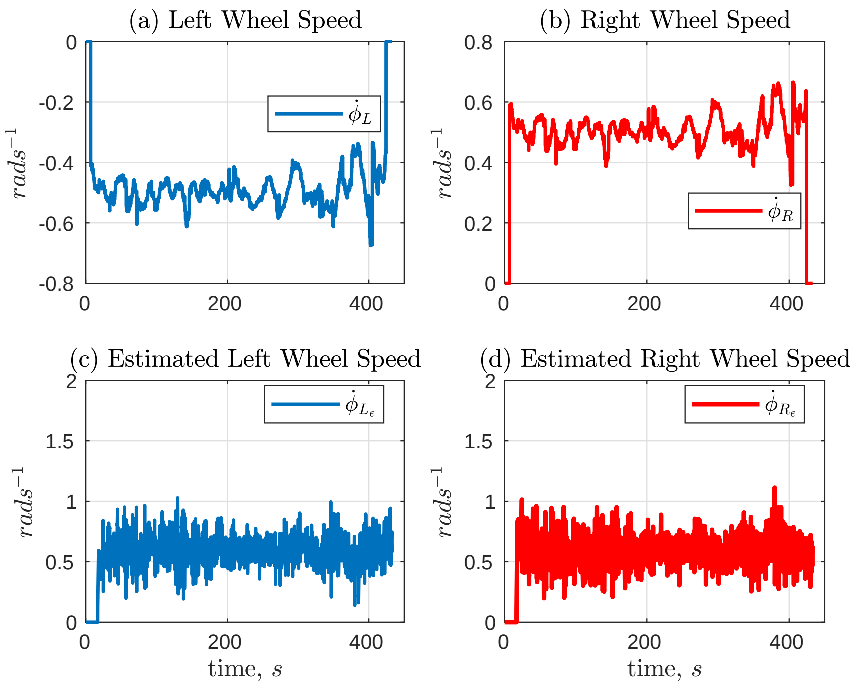


Figure 3.45: (a)-(b) Left and right wheel speeds; (c)-(d) estimated left and right wheel speed.

a real cotton field, were presented in order to validate the proposed control methodology.

In the problem of visual navigation, the use of a single image feature was good enough for navigation in an ideal condition such as a clear path, without occlusions or segmentation mismatch. To handle these issues, more features can be used (e.g. line extraction) as a reference for the control algorithm. Along with this action, stereo cameras with infrared abilities (night vision) or GPS with Real-time Kinematic (RTK) can be used to generates new references for the algorithm. With these upgrades, the robot can be able to perform data collection in hard conditions such night time, allowing to know the behavior of some types of pests.

For the swarm problem, the *Centralized Architecture* proposed here can lead to lost of coordination due to problems with the leader robot or communication loss. To deal with these issues other approaches can be applied such as hierarchical, decentralized or hybrid control architectures [55]. Beside that, the embedded camera can be used to find others members of the formation, estimating its position for collision avoidance and keep the formation in the same way as [56].

In the next chapter, we will discuss about the modeling and control of the car-like robot for position stabilization using the kinematic similarity with the differential-drive model and a control scheme for trajectory tracking based on input/output linearization was proposed. Experimental results are presented to validate the stabilization control approach and numerical simulation was used to proof the controller feasibility.

# 4
# Modeling, Control Design and Prototyping of a Car-like Mobile Robot

## 4.1
## Introduction

In this chapter, we will summarize the process of technological adaptation of a commercial vehicle, the Ground Pounder 1:10 Remote-Controlled (RC) Monstertruck produced by Redcat Racing [61] into the **iCat**: Inteligent Customizable Agricultural Truck, a robot for agricultural field tasks. The objectives are: (i) to develop the mathematical modeling of the vehicle to project controls laws that meet the demands of the interest tasks; (ii) to introduce the electronic design of the devices used to control the vehicle; (iii) to develop programs for integrate the devices devices with the computation control; (iv) perform experimental tests in controlled environment to evaluate, qualitatively and quantitatively, the performance of the proposed robotic system. It is intended to validate the operation of the robot in an external location, with irregular terrain, similar to an agricultural environment.

The choice of the Redcat Ground Pounder model is due to the fact that, in addition to having low cost and desirable off-road vehicle characteristics, the equipment is a small scale model, allowing the use of dynamic similarity for larger scale project validation. Besides that, scale model offer cost reduction with maintenance and logistic during evaluations test, as well the reduce of risks of accidents.

The mathematical model based on the car-like vehicle is used to design control laws for regulation task, using the similarity to the unicycle-like model, and for tracking task, using input/output approximation. Numerical simulations are presented to validate the approaches and experimental results are included to illustrate the feasibility of the proposed methodology.

## 4.2
## Bicycle: the Car-like Mobile Robot

We can characterize a bicycle as a vehicle having an orientable wheel and a fixed one, arranged in a way that the sagittal axis of the rear wheel is aligned to the rotational axis of the front wheel, separated by a distance $l$. The generalized coordinate system is depicted in Fig. 4.1, where $(x\,, y) \in \mathbf{R}^2$ is the Cartesian position of the contact point of the front wheel, $\theta \in \mathbf{R}$ is the angular difference between $\mathcal{F}_w$ and $\mathcal{F}_w$ and $\phi \in \mathbb{R}$ is the steering angle of the front wheel. The point called $ICR$ is the *instantaneous centre of rotation*, whose position depends on the configuration of $\mathbf{q}$.

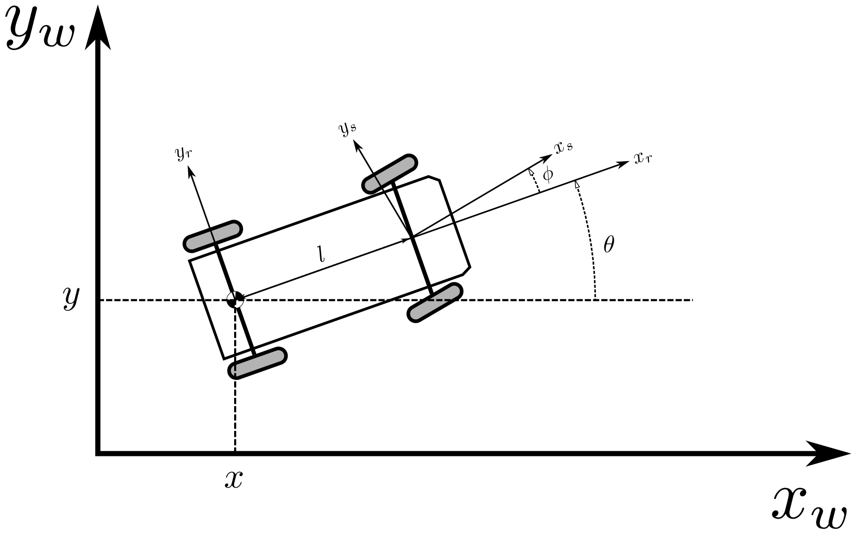


Figure 4.1: Global and robot reference frames for car-like model.

To find the kinematic model for the bicycle we need to compute two constraints, one by each wheel:

$$\dot{x}_f \sin(\theta + \phi) - \dot{y}_f \cos(\theta + \phi) = 0 \tag{4-1}$$

$$\dot{x} \sin\theta - \dot{y} \cos\theta = 0, \tag{4-2}$$

where $(x_f\,,y_f)$ is the Cartesian coordinate of the contact point of the front wheel, can be replaced by the rigid body constraint given by

$$\begin{aligned} x_f &= x + l\cos\theta, \\ y_f &= y + l\sin\theta, \end{aligned}$$

which leads (4-1) to

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - l\dot{\theta}\cos\phi = 0. \tag{4-3}$$

The matrix associated to the Pfaffian constraint is

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \sin\theta & \sin(\theta + \phi) \\ -\cos\theta & -\cos(\theta + \phi) \\ 0 & -l\cos\phi \\ 0 & 0 \end{bmatrix}, \tag{4-4}$$

and one solution for the null space of $\mathbf{A}(\mathbf{q})$ is given by:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos\theta\cos\phi\\ \sin\theta\cos\phi\\ \sin\phi/l\\ 0\end{bmatrix} u_1 + \begin{bmatrix}0\\0\\0\\1\end{bmatrix} u_2. \tag{4-5}$$

Whereas the system has *rear-wheel drive*, we can set $u_1 = v/\cos\phi$ and $u_2 = \omega$, where $v$ is the liner velocity and $\omega$ is the steering velocity, obtain

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos\theta\\ \sin\theta\\ \tan\phi/l\\ 0\end{bmatrix} v + \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \omega. \tag{4-6}$$

### 4.2.1 Bicycle and Unicycle Approximation

The kinematic model for the bicycle robot in (4-6) was derived so that the first two equations to match with the unicycle model presented in (2-5). Because of the similarity between the systems, a control law design for the unicycle can be used in the bicycle, take in account the system constraints. The first one is that the front wheel should never be orthogonal to the rear wheel. This condition would causes a indefiniteness condition in $\tan(\pi/2)$. A physical interpretation is that the rear wheel would be force a motion in the constraint direction of the front wheel, results in a null displacement. Generally, in a four wheeled vehicle, the mechanical structure prevents this situation, limiting the steering angle.

The key idea is to show that the unicycle can be considered as a subsystem of the bicycle. Assuming that the steering angle is controlled by an internal mesh with high gain, the behaviors of the bicycle and the unicycle become approximately equivalents. Thus, from (2-5), consider the follow non-linear function $f(\cdot)$ with two inputs and one output *i.e.* $(\phi, v)\mapsto\overline{\omega}$, such that:

$$\overline{\omega}(t) := f(v,\phi) = (v/\ell)\,\tan(\phi)\,. \tag{4-7}$$

From (4-7), it is possible to define the follow non-linear function $g(\cdot)$ with two inputs and one output, *i.e.* $(v^*, \omega*)\mapsto\overline{\phi}$, such that:

$$\overline{\phi}(t) := g(v^*, \omega*) = \tan^{-1}(\ell\omega^*, v^*)\,, \tag{4-8}$$

where is assumed that $v^*$ e $\omega^*$ are continuous and arbitrary signals and $\overline{\phi}$ is a bounded signal. Thus, the follow control law is defined to the bicycle steering wheel:

$$\omega := K\,[\,\overline{\phi}(t) - \phi(t)\,]\,, \qquad K > 0\,. \tag{4-9}$$

For a high value of gain $K$ (*i.e.*, $K >> 1$) we have $\overline{\phi}(t) \approx \phi(t)$ and, from (4-7) and (4-8) we have $\overline{\omega}(t) \approx \omega(t)$.

Therefore, ensuring a high gain around the steering angle dynamic, we conclude that the control law projected to unicycle $(v^*, \omega^*)$ can be used to control the bicycle using the follows velocity relationships:

$$v = v^*\,, \qquad \omega = K\,[\,\tan^{-1}(\ell\omega^*, v^*) - \phi\,]\,, \tag{4-10}$$

that satisfying the follow conditions: (i) $K >> 1$; (ii) $v^*(t) > 0$; (iii) $\omega^*(t)/v^*(t) \in \mathcal{L}_\infty$. For the cases where is desired a zero linear velocity, it should be ensured that the angular velocity of unicycle tends to zero faster than the linear velocity [62].

## 4.3 Motion Control

### 4.3.1 Partial Regulation Task for Car-Like Robot

Consider that we want to perform a partial regulation task using a bicycle-like or car-like robot. We consider, for instance, that the task is to visit a list of waypoints to collect data (e.g. image, sensing or mapping) such that the final orientation is not relevant. Thus, we can use the bicycle and unicycle approximation presented in Sec. 4.2.1.

To this case, we can consider the problem of partial regulation presented in Sec. 2.3.1. Applying the relationship (4-9) to design a control law to the car-like robot, we can assign the linear control law (2-9) for $v$ and use (4-10) to define the steering velocity of direction wheel as presented below. In this way, we have the following control law:

$$v := v^*\,, \tag{4-11}$$

$$\omega = K[\text{atan2}(l\omega^*, v^*)\; - \;\phi]\,, \tag{4-12}$$

with $v^*$ and $\omega^*$ being obtained from the control design of the unicycle model as

$$v^* = k_1[x\cos(\theta) + y\sin(\theta)]\,, \tag{4-13}$$

$$\omega^* = k_2[\text{atan2}(y, x) - \theta]\,, \tag{4-14}$$

where $l$ is the distance between the front and rear wheels, $\omega^*$ and $v^*$ are respectively the angular and linear velocity from the bicycle law and $\phi$ is the current angular position of the direction wheel. The proof of this approach was presented to the unicycle model, given that the same control law and conditions were used. To evaluate the controller, was performed numerical simulations presented bellow and experimental tests will be presented in Sec. 4.5.

For the numerical simulations, Tab. 2.1 once again was used, aiming to create a baseline to compare the proposed controller and applications. The gains ($k2$ ,$k2$) was the same used to unicycle control (Sec. 2.3.1) and $K = 5$ to ensuring a high gain around the steering angle dynamic. The steering angle was limited to satisfy the assumption that $\phi$ is has a limited range, for this case, between $\pm\pi/12\,rad$ and the length of $l = 0.35\,cm$. This values was used to math with the characteristic set of the real robot, used in the experimental tests. Figs. 4.2 and 4.3 shows the displacement of the robot and the behaviour of the state variables for the proposed case.

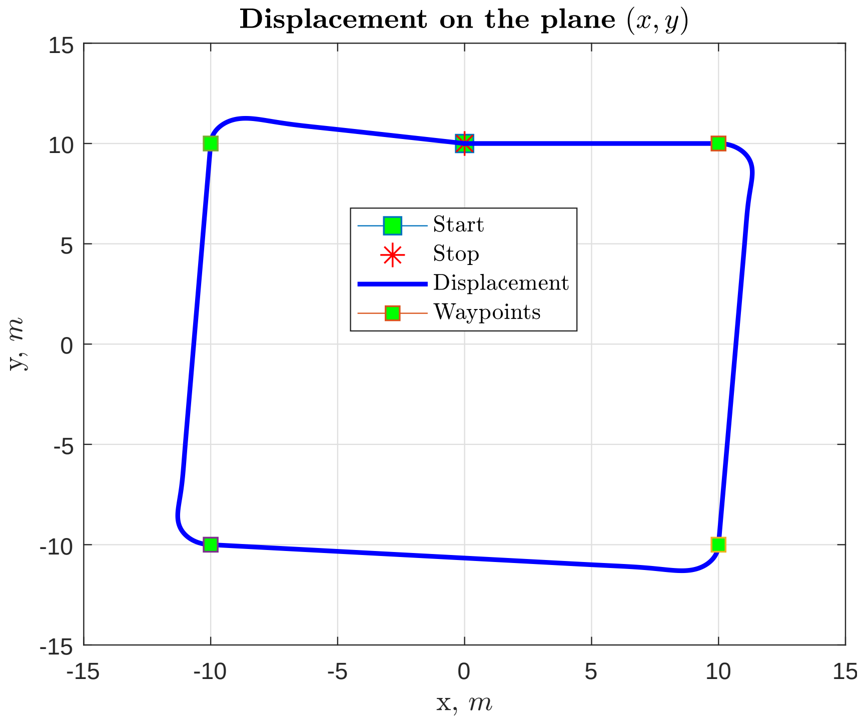


Figure 4.2: Robot displacement in the plane ($x$ ,$y$).

As can be seen, the robot was visited all the waypoints with zero regulation error, as expected (Fig. 4.4). The system dynamics tends to zero as the robot approach to destination, which includes the robot angular velocity and the steering velocity. This results proof the approach feasibility and its extension to a real case is natural.

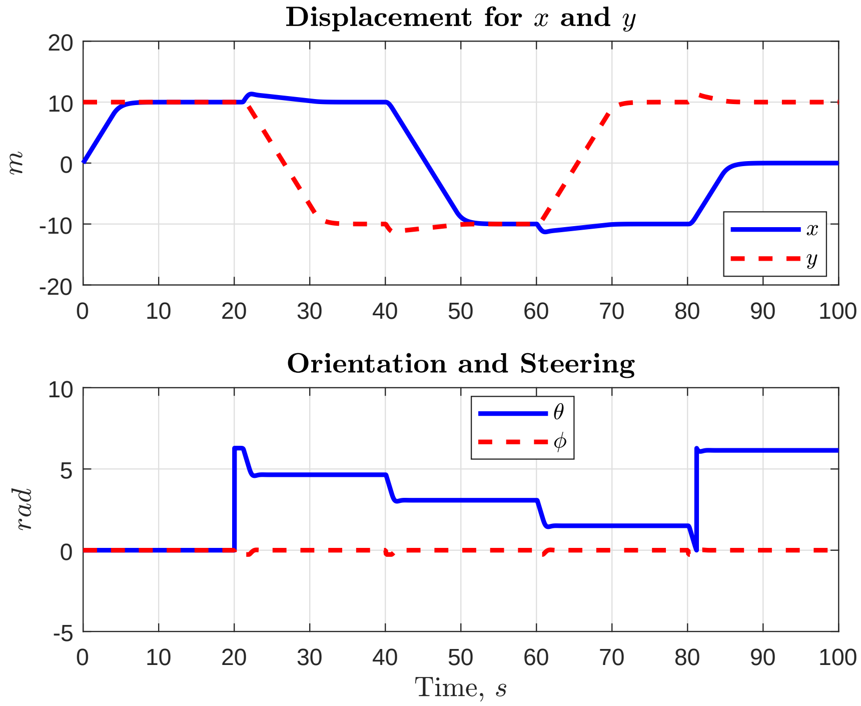


Figure 4.3: Behaviour of state variables $x$, $y$, $\theta$ and $\phi$.

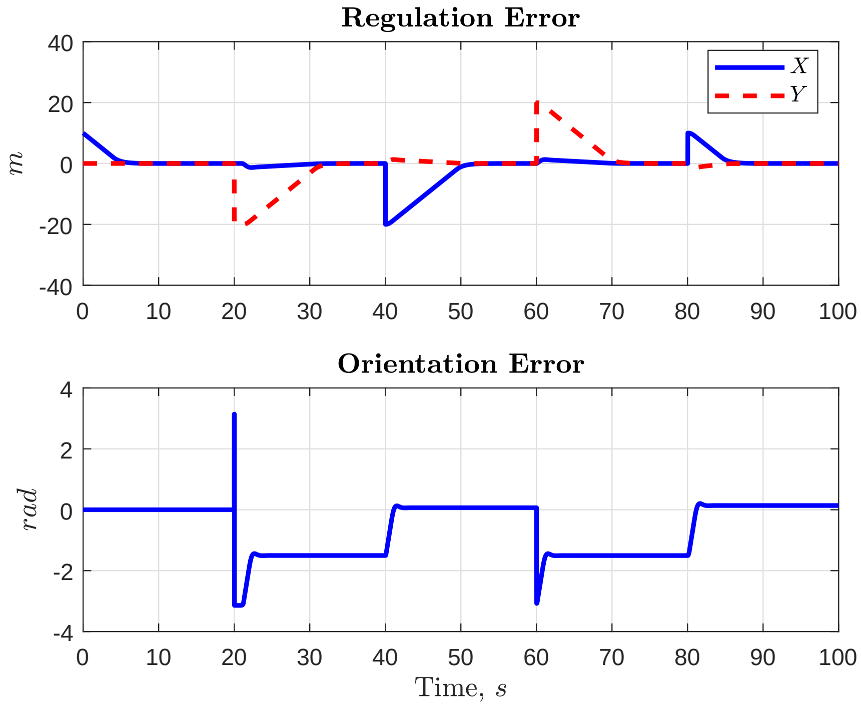


Figure 4.4: Position and angular error for tracking.

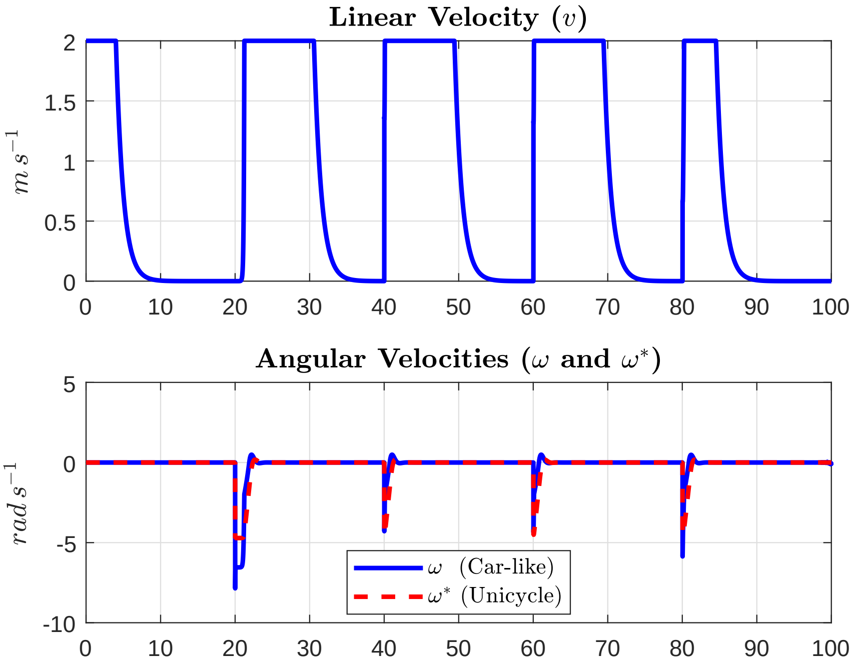


Figure 4.5: Control signal for $v$, $\omega$ and $\omega^*$.

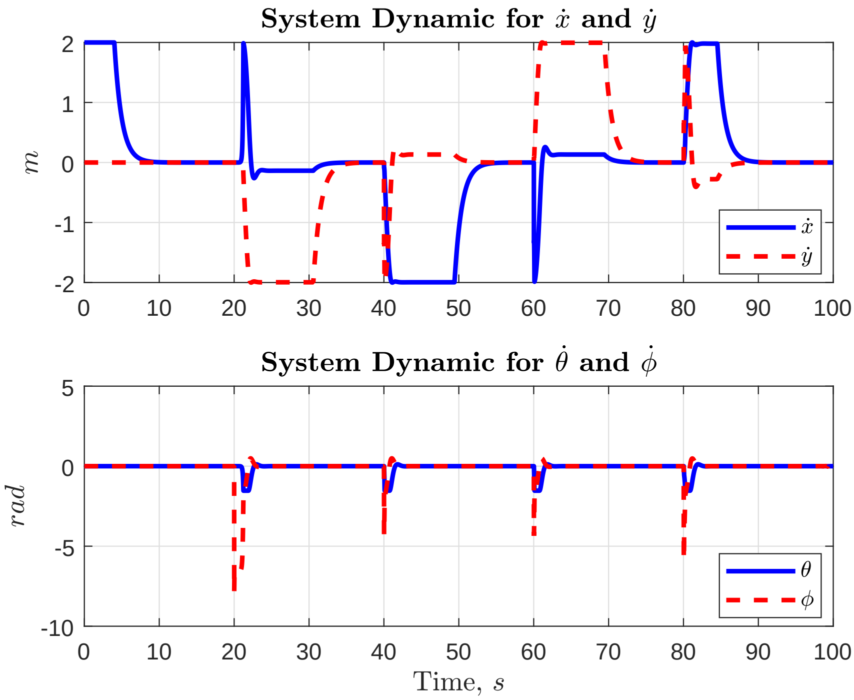


Figure 4.6: Dynamic for the system state variables.

### 4.3.2 Input/Output Linearization Method for Tracking Control

The trajectory tracking problem consists in the asymptotic stabilization of a desired Cartesian trajectory, denoted by the curve $\mathcal{C}$ (see Fig. 4.7), with the robot starting from an initial configuration $q_0 = [\, x_0 \;\; y_0 \;\; \theta_0 \,]^\mathsf{T}$. This problem is of particular interest for the case where the robot needs to move along a desired trajectory or path, such crop rows, using a preliminary planning step (e.g., image-based planning, GPS maps). A systematic approach to design a tracking controller is based on the input/output linearization via feedback [41, 48].

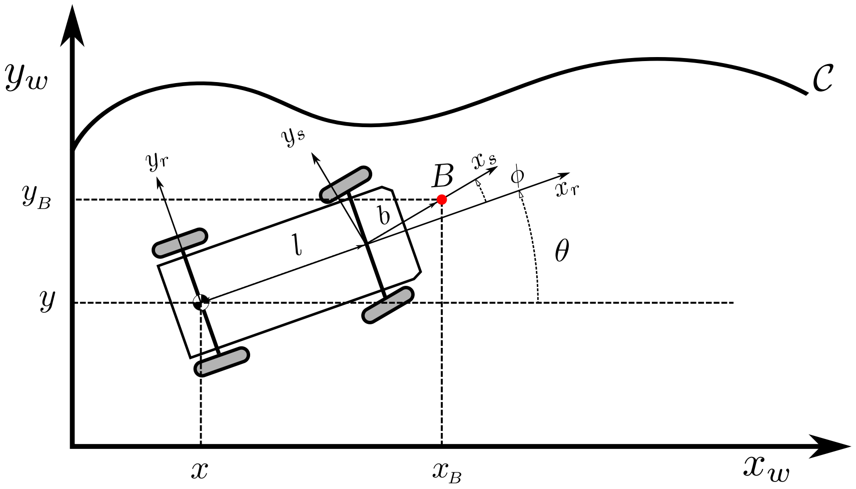


Figure 4.7: Global and robot reference frames for the bicycle model.

Consider the representation of a car-like mobile robot depicted in Fig. 4.7, that has a point of interest **B** attached to a line defined along the $x$-axis of the steering wheel frame $\mathcal{F}_s$, and located at a distance $|b|$ from its origin. The position coordinates of the point **B** with respect to the frame $\mathcal{F}_w$ can be expressed by:

$$x_B = x + l\cos(\theta) + b\cos(\theta + \phi)\,, \tag{4-15}$$

$$y_B = y + l\sin(\theta) + b\sin(\theta + \phi)\,, \tag{4-16}$$

where $(x\,, y) \in \mathbb{R}^2$ is the Cartesian coordinates of the mobile robot and $(y_1,\, y_2) \in \mathbb{R}^2$ is the system outputs related to point $\mathcal{B}$. The time-derivative of (4-15) and (4-16) are given in matrix form by:

$$\begin{bmatrix} \dot{x}_B \\ \dot{y}_B \end{bmatrix} = \begin{bmatrix} \cos(\theta) - \tan(\phi)\sin(\theta) - \dfrac{b}{l}\tan(\phi)\sin(\theta+\phi) & -b\sin(\theta+\phi) \\ \sin(\theta) + \tan(\phi)\cos(\theta) + \dfrac{b}{l}\tan(\phi)\cos(\theta+\phi) & b\cos(\theta+\phi) \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{4-17}$$

and the simplified control system is given by:

$$\dot{\mathbf{p}} = T(\theta,\phi)\mathbf{v}\,, \tag{4-18}$$

where $\dot{\mathbf{p}} = [\,\dot{x}_B \;\; \dot{y}_B\,]^{\mathsf{T}}$, $\mathbf{v} = [\,v \;\; \omega\,]^{\mathsf{T}}$ and the transformation matrix T$(\theta,\phi)$ is given by:

$$T(\theta,\phi) = \begin{bmatrix} \cos(\theta) - \tan(\phi)\sin(\theta) - \dfrac{b}{l}\tan(\phi)\sin(\theta+\phi) & -b\sin(\theta+\phi) \\ \\ \sin(\theta) + \tan(\phi)\cos(\theta) + \dfrac{b}{l}\tan(\phi)\cos(\theta+\phi) & b\cos(\theta+\phi) \end{bmatrix}.$$

To guarantee that the transformation matrix $T(\theta,\phi)$ is non-singular, its determinant given by

$$\begin{aligned} \det(T) = {} & b\cos(\phi+\theta)\,[\cos(\theta) - \tan(\phi)\sin(\theta)] + \\ & b\sin(\phi+\theta)\,[\,\sin(\theta) + \cos(\theta)\tan(\phi)\,]\,, \end{aligned} \tag{4-19}$$

must be nonzero. Based on this assumption, we can use the following input transformation:

$$\mathbf{v} = T^{-1}(\theta,\phi)\,\mathbf{u}\,, \tag{4-20}$$

where $\mathbf{v}=[v\;\omega]^{\mathsf{T}}$ and $\mathbf{u}=[u_1\;u_2]$. Applying this transformation into the control system (4-18), we obtain the linearized system:

$$\dot{\mathbf{p}} = \mathbf{u}. \tag{4-21}$$

Consider that the problem is to track a desired reference trajectory $\mathbf{p}_d(t) = [\,x_d(t)\;\; y_d(t)\,]^{\mathsf{T}}$, admissible for the kinematic model (4-6). The control goal can be defined by:

$$x_B \rightarrow x_d(t)\,, \qquad e_1(t) = x_d - x_B \rightarrow 0\,, \tag{4-22}$$

$$y_B \rightarrow y_d(t)\,, \qquad e_2(t) = y_d - y_B \rightarrow 0\,, \tag{4-23}$$

where $\mathbf{e}=[\,e_1\; e_2\,]^{\mathsf{T}}$ is the tracking error. The following linear controller can be proposed:

$$\mathbf{u} = \dot{\mathbf{p}}_d + K\,(\mathbf{p}_d - \mathbf{p})\,, \tag{4-24}$$

and choosing the gain matrix $K \in \mathbb{R}^{2\times 2}$ as a positive definite matrix, the exponential convergence of the tracking error to zero is guaranteed. In this case, the orientation is not controlled due to the fact that this control scheme is based on the output error rather than the state error [41].

The performance of the proposed control approach has be validated by numerical simulations presented as follows. The desired trajectory $\mathbf{p}_d$ was generated by the interpolation of the waypoints, shown in Tab. 4.1, using a cubic polynomial function. The selected gain matrix was chosen as $K = 30\mathbf{I}$.

Table 4.1: List of waypoints for trajectory generation.

| Desired configuration | | |
|---|---|---|
| $x_d\,(m)$ | $y_d\,(m)$ | $\theta_d$ (rad) |
| 0 | 0 | $\pi/2$ |
| 0 | 8 | $\pi/2$ |
| 5 | 8 | $-\pi/2$ |
| 5 | 0 | $-\pi/2$ |
| 10 | 0 | $\pi/2$ |
| 10 | 8 | $\pi/2$ |
| 15 | 8 | $-\pi/2$ |
| 15 | 0 | $-\pi/2$ |
| 20 | 0 | $\pi/2$ |
| 20 | 8 | $\pi/2$ |
| 25 | 8 | $-\pi/2$ |
| 25 | 0 | $-\pi/2$ |
| 0 | 10 | 0 |

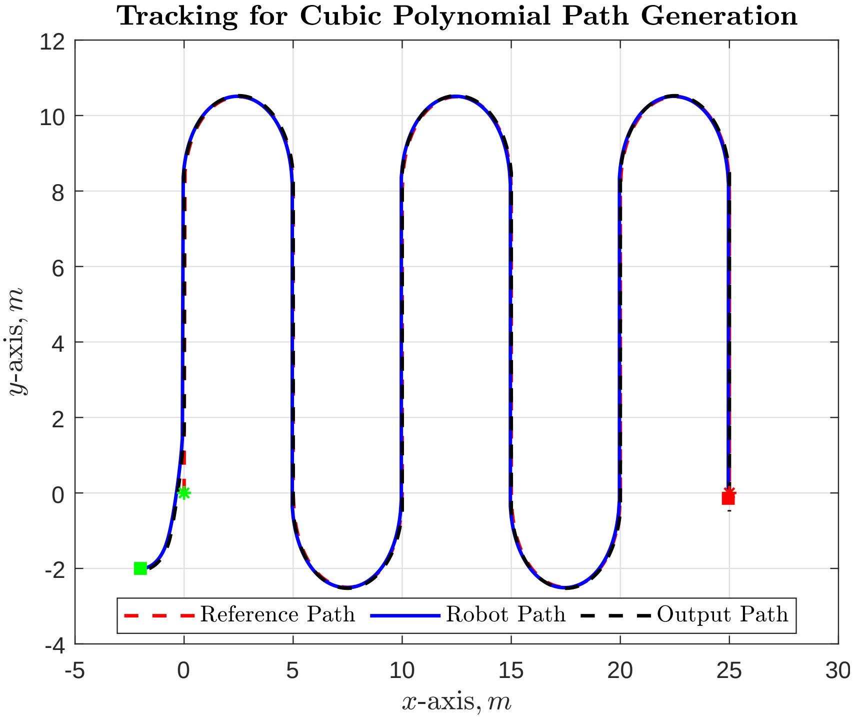


Figure 4.8: Displacement in the $(x\,,y)$ plane.

Consider for instance that the reference velocities $\dot{x}_d$ and $\dot{y}_d$ are not available for measurement, or these signals are not continuously differentiable. From a practical point of view, these signals can be considered as bounded

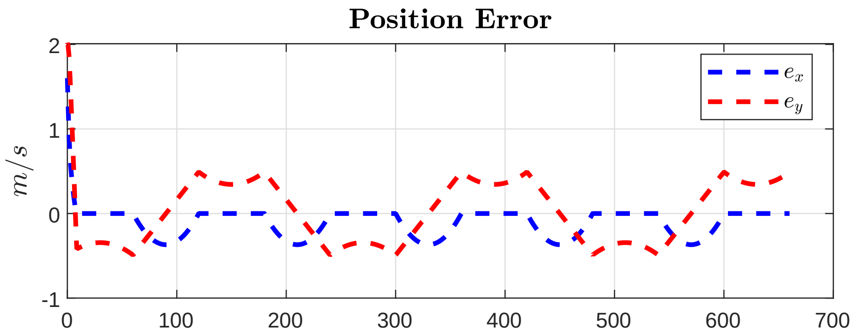


Figure 4.9: System error for tracking.

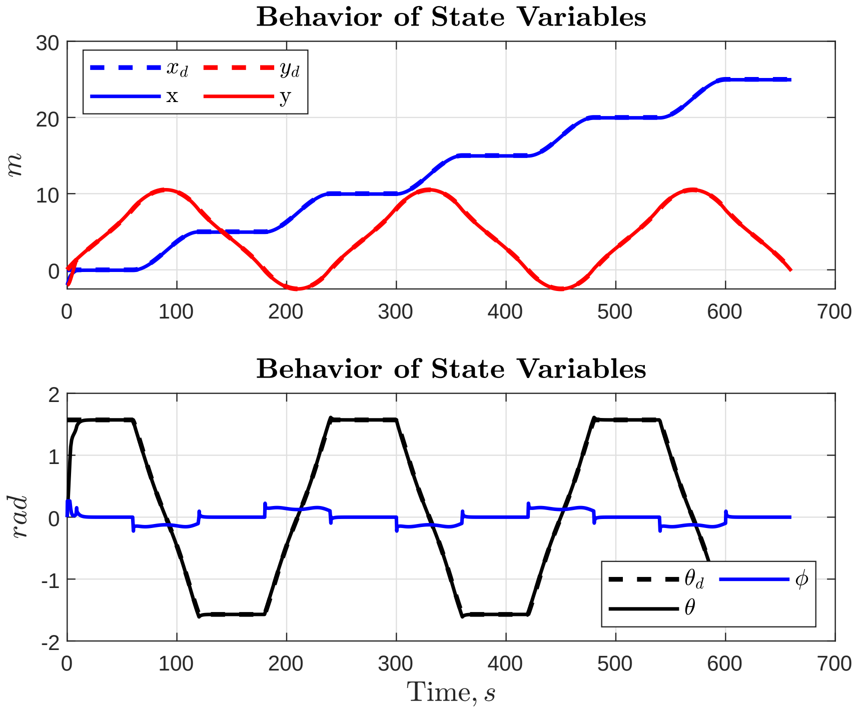


Figure 4.10: Behaviour of System States.

external disturbances. In this case, a control scheme based on the robust control techniques, such as Sliding Mode Control approach (SMC) [46] and Super-Twisting Algorithm (STA) [63,64], can be applied. From (4-21), we can observe the the control system has relative degree one which means the error dynamics is governed by a first order equation. So, we can propose the use of two sliding surfaces based on the error components (4-22) and (4-23) as well as their integral terms. Thus, the proposed sliding surfaces are given by:

$$\sigma_i = e_i + \alpha_i \int_0^t e_i(\tau)\, d\tau \, . \tag{4-25}$$

and the structure of the STA controller is given by:

$$u_i = \beta_i |\sigma_i|^{1/2} \text{sign}(\sigma_i) + w_i \, , \tag{4-26}$$

$$\dot{w}_i = -\lambda_i \, \text{sign}(\sigma_i) \, , \tag{4-27}$$

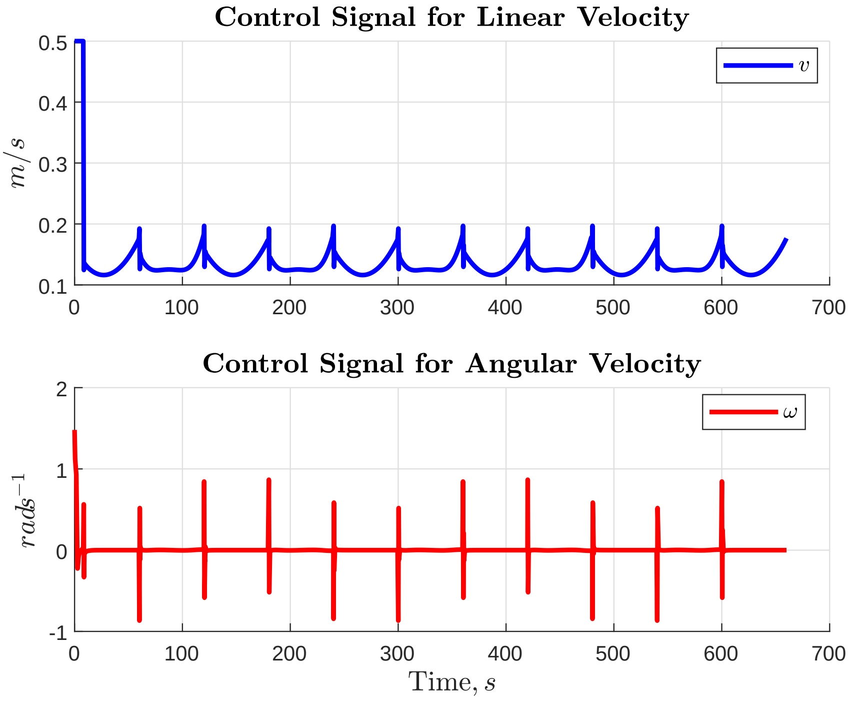


Figure 4.11: Control signal for linear ($v$) and steering ($\omega$) velocities.

where $\alpha_i$ is the slope of the sliding surface, $\beta_i > 0$ and $\lambda_i > 0$ are scalar gains, for $i = 1, 2$.

The robustness of proposed tracking control scheme and its convergence analysis can be performed by using the following Lyapunov-like candidate function:

$$V(\sigma_1\,,\sigma_2) = \lambda_1\,|\sigma_1| + \frac{1}{2}\,w_1^2 + \lambda_2\,|\sigma_2| + \frac{1}{2}\,w_2^2\,. \tag{4-28}$$

Taking its time-derivative along the system trajectories we have:

$$\dot{V}(\sigma_1, \sigma_2) = \lambda_1|\dot{\sigma}_1| + w_1\dot{w}_1 + \lambda_2|\dot{\sigma}_2| + w_2\dot{w}_2\,. \tag{4-29}$$

Then, considering that $|\dot{\sigma}_i| = \dot{\sigma}_i\,\text{sign}(\sigma_i)$ for $i = 1, 2$ and based on the definition of the sliding surfaces $\sigma_i$ (4-25) we obtain:

$$\begin{aligned} \dot{V}(\sigma_1, \sigma_2) \;&=\; \lambda_1\,[\,e_x + \dot{x}_d - u_1\,]\;\text{sign}(\sigma_1) + w_1\dot{w}_1 + \\ &\qquad \lambda_2\,[\,e_y + \dot{y}_d - u_2\,]\;\text{sign}(\sigma_2) + w_2\dot{w}_2\,. \end{aligned} \tag{4-30}$$

Substituting the STA controller (4-26) into (4-30) yields:

$$\begin{aligned} \dot{V}(\sigma_1, \sigma_2) \;&=\; -\lambda_1\,\left[\beta_1\,|\sigma_1|^{1/2} - (e_x + \dot{x}_d)\,\text{sign}(\sigma_1)\right] \\ &\qquad -\lambda_2\,\left[\beta_2\,|\sigma_2|^{1/2} - (e_y + \dot{y}_d)\,\text{sign}(\sigma_2)\right]\,. \end{aligned} \tag{4-31}$$

If we assume $\sup_{t\geq 0}|\dot{x}_d| < x_M < \infty$ for $\forall\dot{x}_d$ and $\sup_{t\geq 0}|\dot{y}_d| < y_M < \infty$ for $\forall\dot{y}_d$, where $x_M$ and $y_M$ are bounded constants, we can show that $\dot{V}(\sigma_1, \sigma_2)$

is negative definite provided that $\beta_1 \geq x_M$ and $\beta_2 \geq y_M$. Notice that the last assumption is practically satisfied, since any planned reference trajectory cannot require infinite velocities.

In order to validate the proposed approach, numerical simulations are presented using the same initial conditions (Tab. 4.1) as before. We choose the sloop of the sliding surface as $\alpha = (0.5\,,\,0.5)$, and gains $\beta = (0.5\,,\,0.5)$ and $\lambda = (0.0025\,,\,0.0025)$. These gains were obtained empirically, through trial and error method. The set of simulation presents the results under the assumption that the reference values $\dot{x}_d$ and $\dot{y}_d$ are not available. As can be seen, the system error converges in finite time, The first set of simulations presents the results under the assumption that the value of the reference velocities $\dot{x}_d$ and $\dot{y}_d$ are available, while in the second set these values are not available.

We can see that the controller is able to keep the robot on the path, tracking in both cases with zero-tending error.

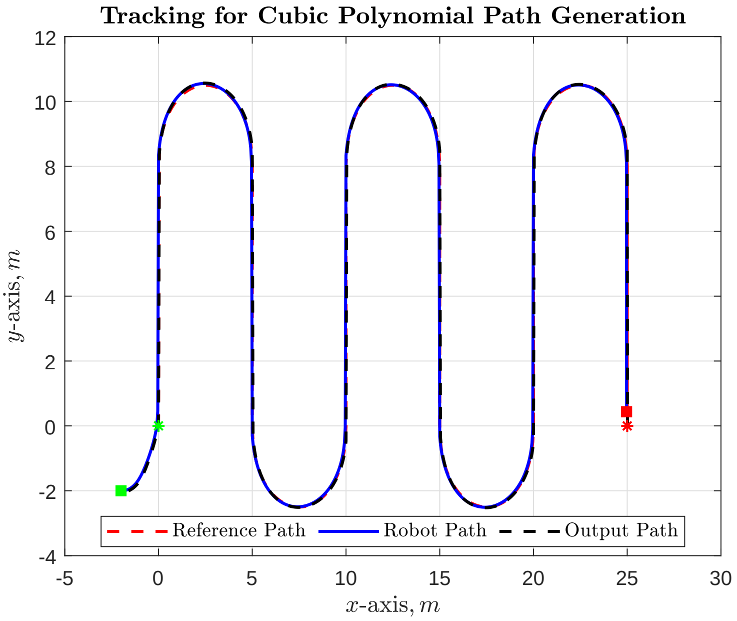


Figure 4.12: Displacement in the $(x\,,y)$ plane using STA without reference velocities $\dot{x}_d$ and $\dot{y}_d$.

## 4.4 The iCat Robot

The Inteligent Customizable Agricultural Truck - iCat (Fig. 4.17, is a car-like robot platform developed from a commercial remote-control vehicle Ground Pounder 1:10 Monster Truck, produced by RedCat Racing [61]. This

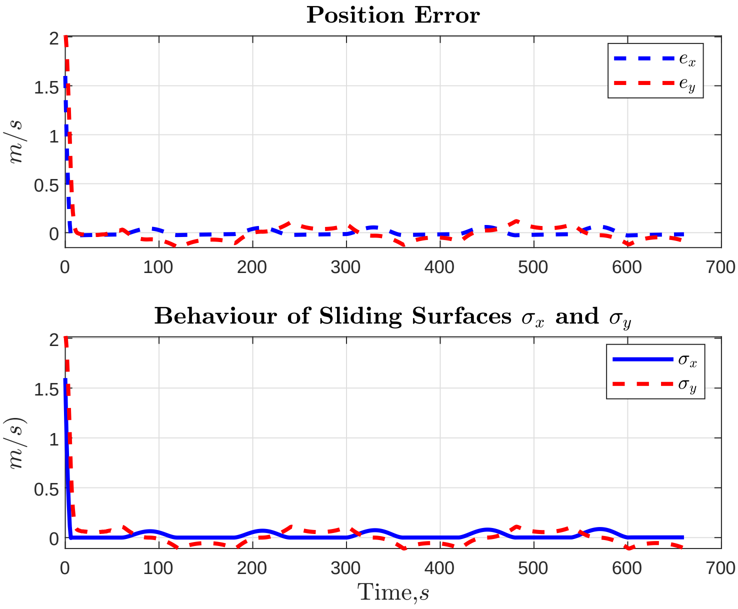


Figure 4.13: System error for tracking using STA without reference velocities $\dot{x}_d$ and $\dot{y}_d$.

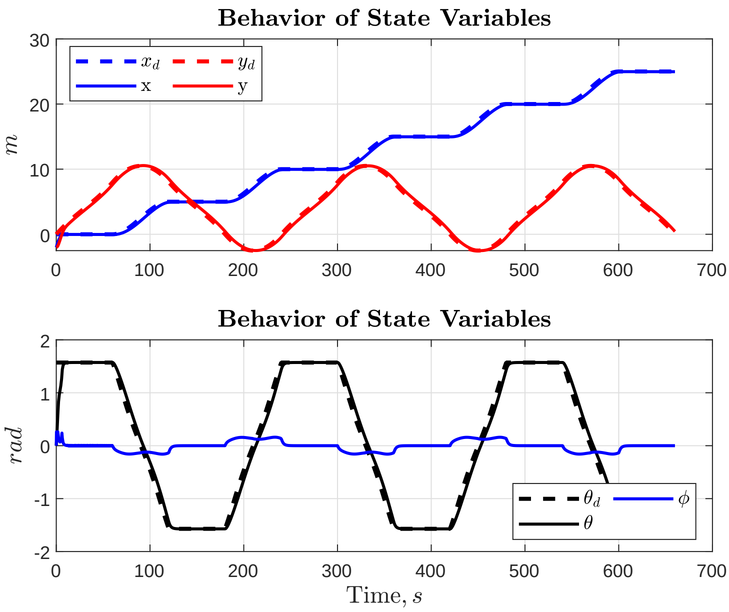


Figure 4.14: Behavior of System States using STA without reference velocities $\dot{x}_d$ and $\dot{y}_d$.

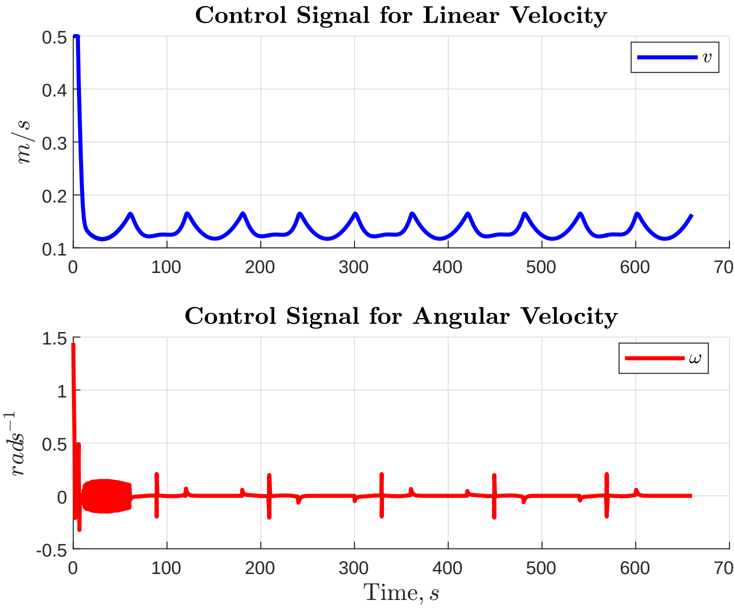


Figure 4.15: Control signal for linear ($v$) and steering ($\omega$) velocities using STA without reference velocities $\dot{x}_d$ and $\dot{y}_d$.

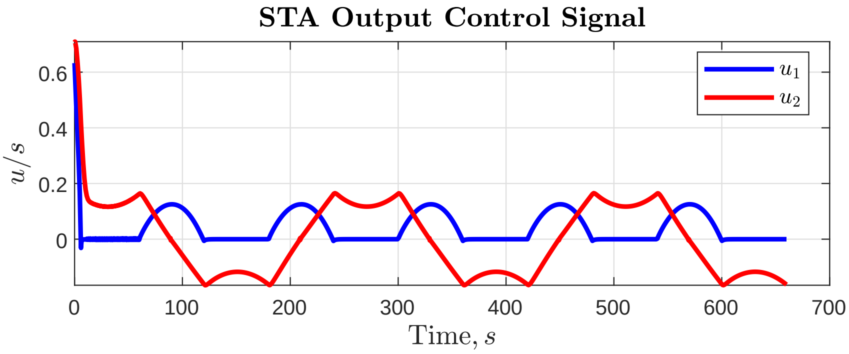
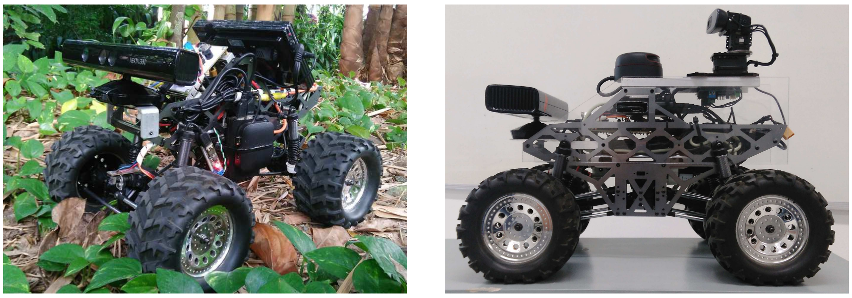


Figure 4.16: Control signal from STA controller.

kind of adaptation is a recurrent approach for research in mobile robotic for providing a low-cost and safe solution [22, 65].



($a$) ($b$)

Figure 4.17: The iCat robot: (a) perspective views in field application; (b) details of embedded sensors.

The iCat Robot has desirable features for precision agriculture such low-cost of acquisition, maintenance and logistic and its off-road design (e.g. water proof parts, suspension system, endurance,) allows the field use. Moreover, the reduce scale size reduce the risk of accidents during operation and transport.

The robot mechanical structure have the follow dimensions: 336,5 mm of wide, 457,2 mm of length and 260,4 mm of height, with 4 (four) wheel with diameter of 149,2 mm by 82,6 mm of scrolling bandwidth. The distance between the wheels ($l$) is 350mm and steering range of approx ±27 degrees ($\pi/12$ rad). In the original configuration 2 (two) servomotors was used for steering the front and read wheel, but only one is used in the final version. An electric DC motor drives the wheels, supplied by 12V DC batteries.

The robot have an embedded control module (MCE) for proprioceptive sensors reading (e.g. encoders, IMU) and control of the actuators, such driver motor and directional wheel steering servomotor, an ARMv3 Cortex-A5 microcomputer Raspberry PI board, for high-level control, external communication and exteroceptive sensors connection. The current version is equipped with a Microsoft Kinect sensor, a Logitech c270 RGB camera with pan-tilt arm, an RPLidar planar laser sensor for mapping and obstacle avoidance applications and a GPS module. It was used the Robot Operating System framework (ROS) is used to provide the communication infrastructure, running on Linux OS Ubuntu Mate 16.

The software design consists of MCE firmware development, using the C programming language, and the ROS nodes running into the embedded computer and the remote computer, both using Python programming language. The customized control package was developed using Python programming language. Fig. 4.18 presents the logical and physical architecture of the system. The Control package, that running into the Raspbery PI, perform the communication between the MCE and the ROS. This drive request telemetry data from the vehicle and control the linear velocity and the directional wheel steering angle by the action of the drive motor and direction servomotor, respectively. The command package, that running into the remote computer, manages the high-level control. The node receives the desired goal coordinates and, base on the selected control law, sends the linear velocity and steering angle values to the control node. All control data is stored in a log file.

## 4.5 Experimental Results: iCat

The experimental tests was performed in a controlled environment with wide area, where the robot need to perform a regulation task to a desired coordinate with respect to its initial position. The system origin is at the place

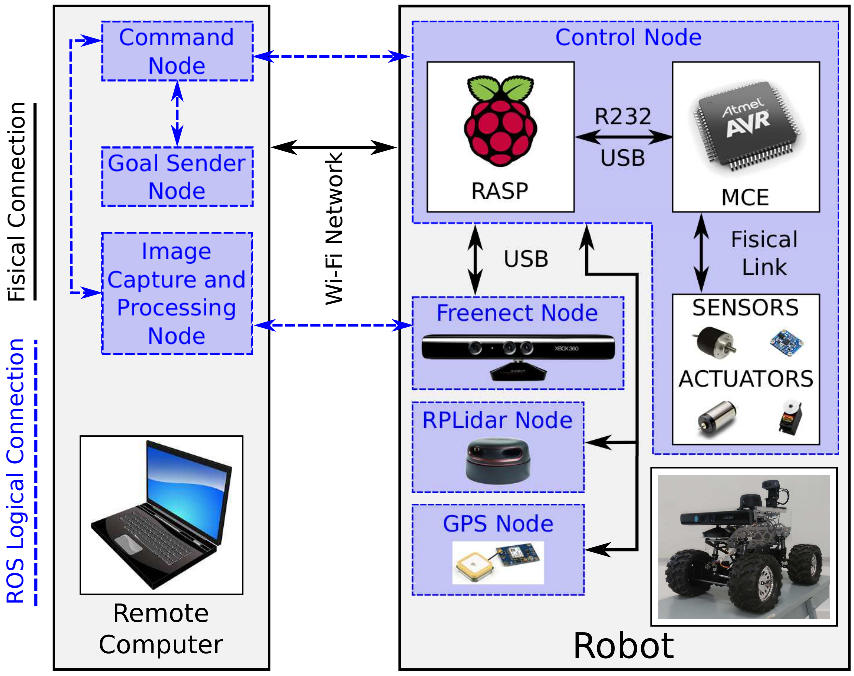


Figure 4.18: System architecture of the iCat robot.

where the robot was turned on and all coordinate are related to this point. Two sets of tests were performed, with non-sequential laps arriving from the system origin (0,0) to goal coordinate (7,0) in the foremost, with three laps, and to (7,−3) in the latest, with two laps. In order to prevent accidents and mechanical damage, limiters was used into the controller for linear velocity ($v_{max}=0.6\,m\,s^{-1}$) and maximum steering angle ($\pm\,0.2\,rad$).

In Fig. 4.19 we can see the robot displacement at the real environment. Fig. 4.21 shows the robot displacement during the regulation task using Matlab. Fig. 4.20 shows a three dimensional model of the robot using Rviz, a native ROS software which allow emulate the robot operate conditions on line.

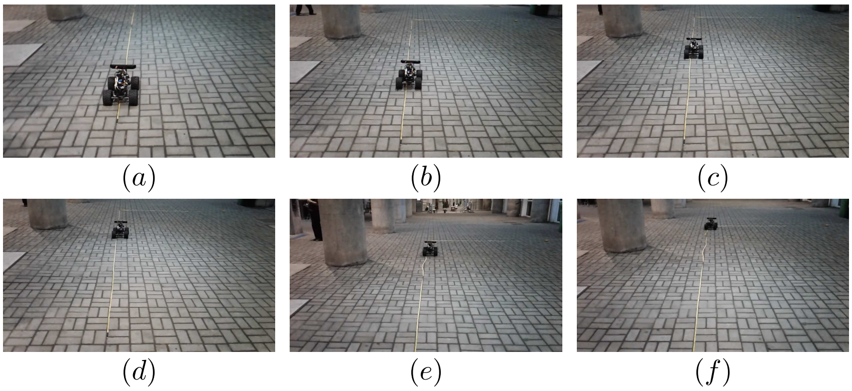
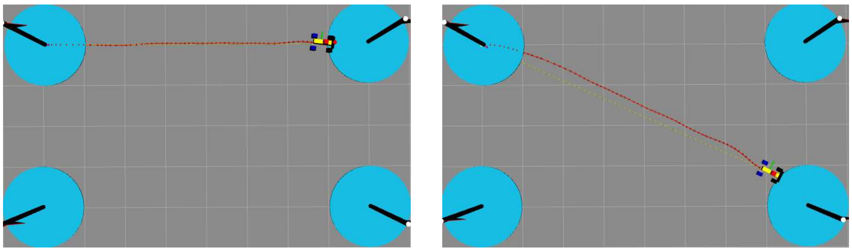

$(a)$ $(b)$ $(c)$
$(d)$ $(e)$ $(f)$

Figure 4.19: Robot displacement in Experiment 1.

As can be seen, the regulation task was accomplished with an expected error due to the use of a threshold to define a valid region of $0.4\,m$ around the



(*a*) (*b*)

Figure 4.20: Robot displacement on the plane using Rviz: (a) Experiment 1; (b) Experiment 2.

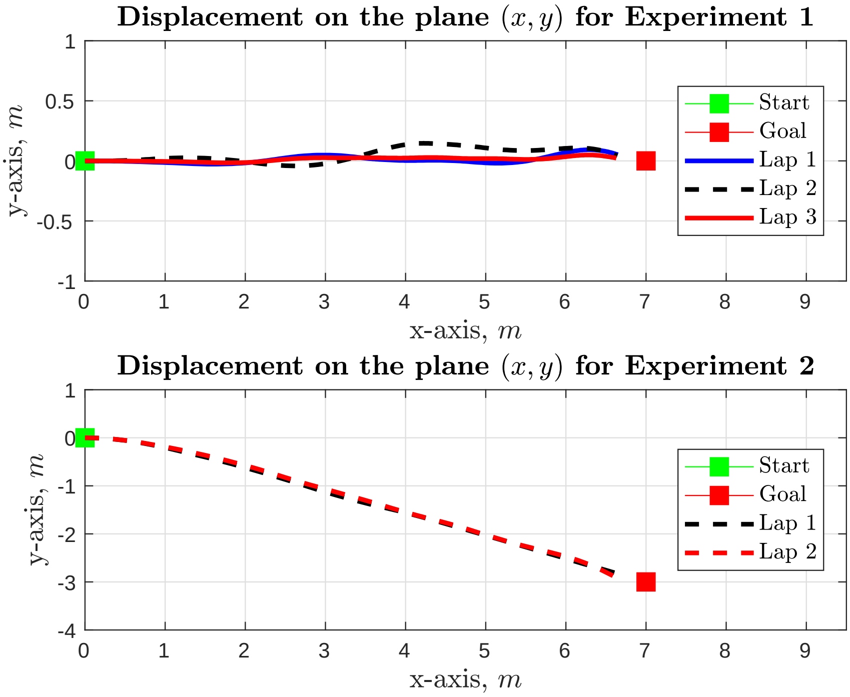


Figure 4.21: Robot displacement on the plane ($x$,$y$) in both experiments.

goal coordinate. The regulation error tend to zero as the vehicle approach to the goal coordinates, comparing Figs. 4.22 and 4.23 for experiment 1 and 4.26 and 4.27 for experiment 2. The control signal for linear velocity tends to zero along with the error due to the proposed proportional controller. Figs. 4.25 and 4.25 presents this behavior.

For the first experiment, the angular control remains almost constant, due to the trajectory to the goal coordinate to be a straight line. In fact, the value of *theta* remains near to zero. The disturbance observed in $\phi$ happen because of the noisy steering sensor, which produces value out of the limit range. In other hand, the steering control signal remains balded by the used limiter (Figs. 4.25 and 4.29).

For the second experiment, the vehicle needs to perform a maneuver to align with the destination. This action can be found in Fig. 4.29, where the controller sends a steering signal. Thus, the orientation of the vehicle turns to a desire heading angle and remains constant, as the steering signal. Note that the steering signal present a peak of action but the orientation of the vehicle suffered small changes. This happen because the car-like angular velocity is coupled with the linear one and this signal was tending to zero.

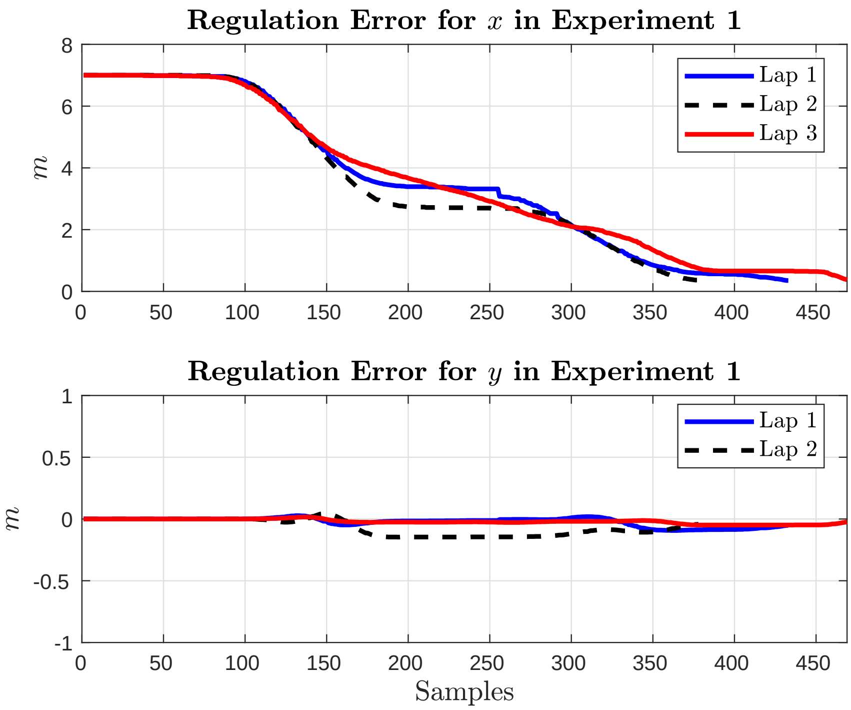


Figure 4.22: Regulation error for variable $x$ and $y$ in Experiment 1.

## 4.6 Concluding Remarks

In this chapter a technological adaptation of a RC car-like model for agricultural robotic applications was presented. Electronic device were built to integrate the sensors and manage programs developed for vehicle control. The use of the ROS framework allowed a rapid integration between embedded sensors, as well as the compatibility with others sensors that could be further incorporate to the system.

A control methodology based on the kinematic model of using the similarity between the unicycle-like and car-like models was used, based on a simplification of kinematic model of the bicycle. As can be seen, this approach is good enough for fast steering dynamics such the iCat robot used in this work.

It was presented some approach for tracking task, based on Input/Output (In/Out) Linearization technique, using proportional-derivative controller (PD) and a robust controller using Super Twisting Algorithm (STA). The

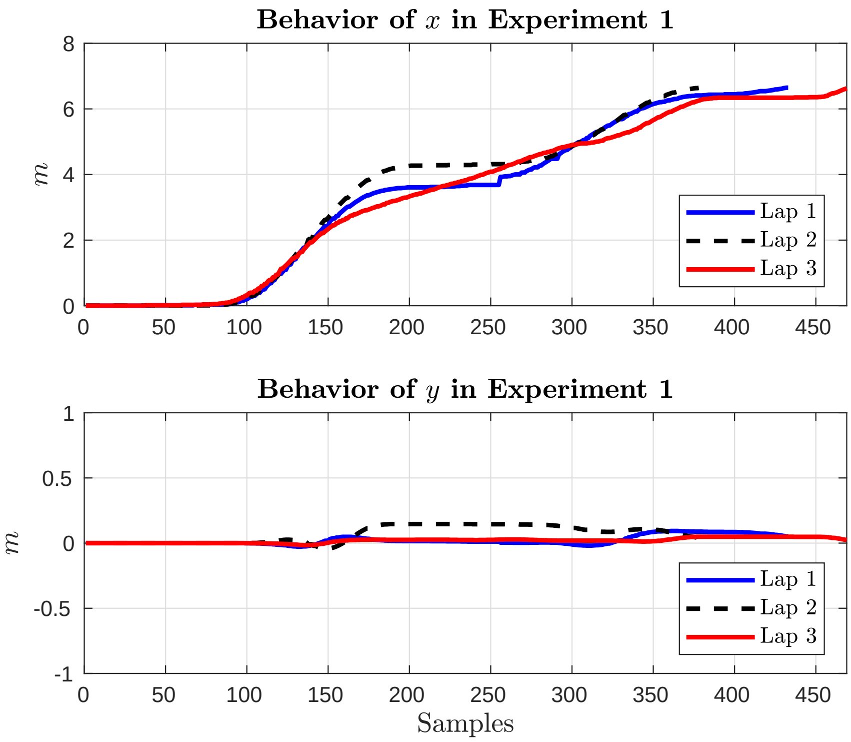


Figure 4.23: Behavior of state variable $x$ and $y$ in Experiment 1.

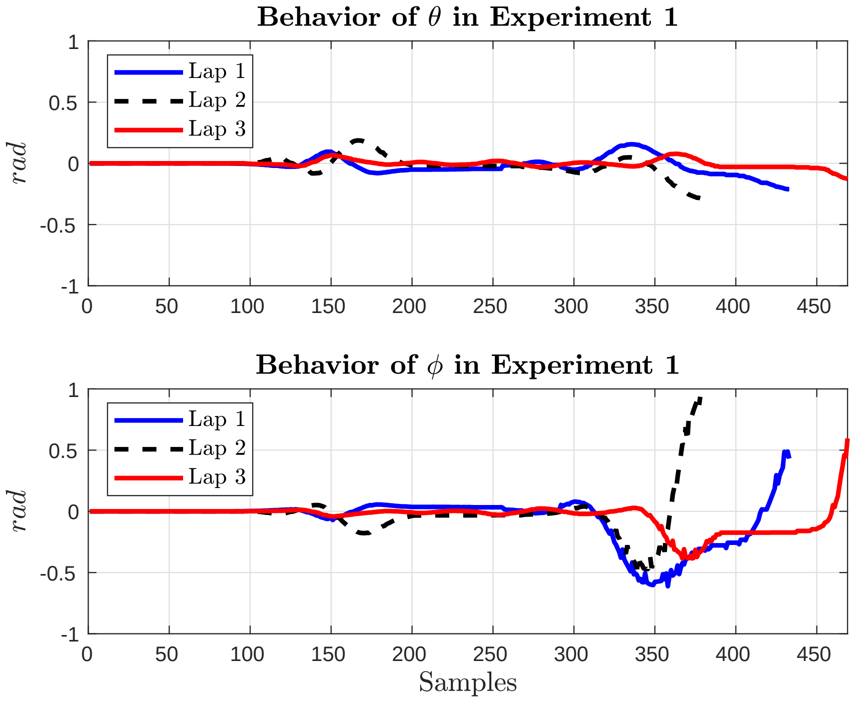


Figure 4.24: Behavior of state variable $\theta$ and $\phi$ in Experiment 1.

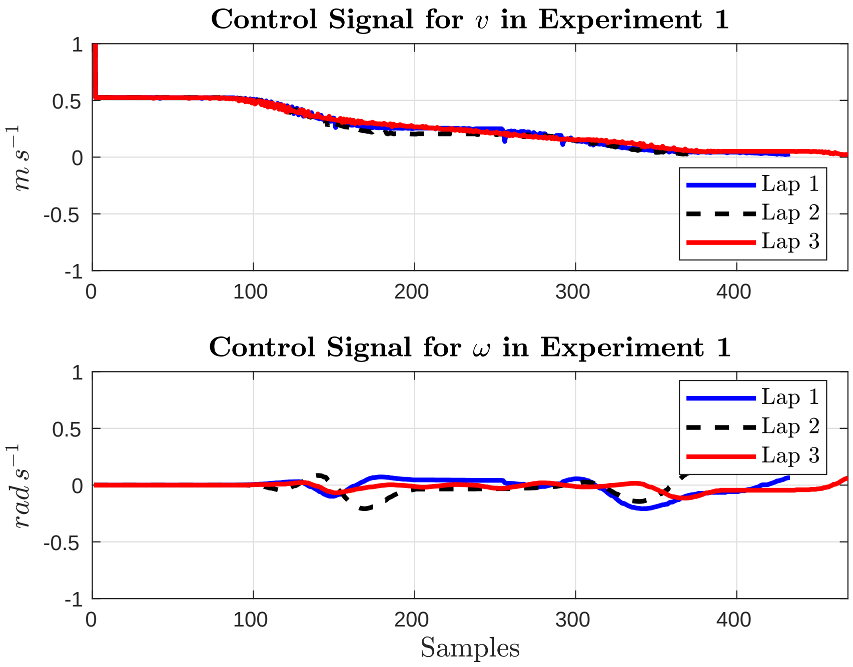


Figure 4.25: Control signal for $v$ and $\omega$ in Experiment 1.

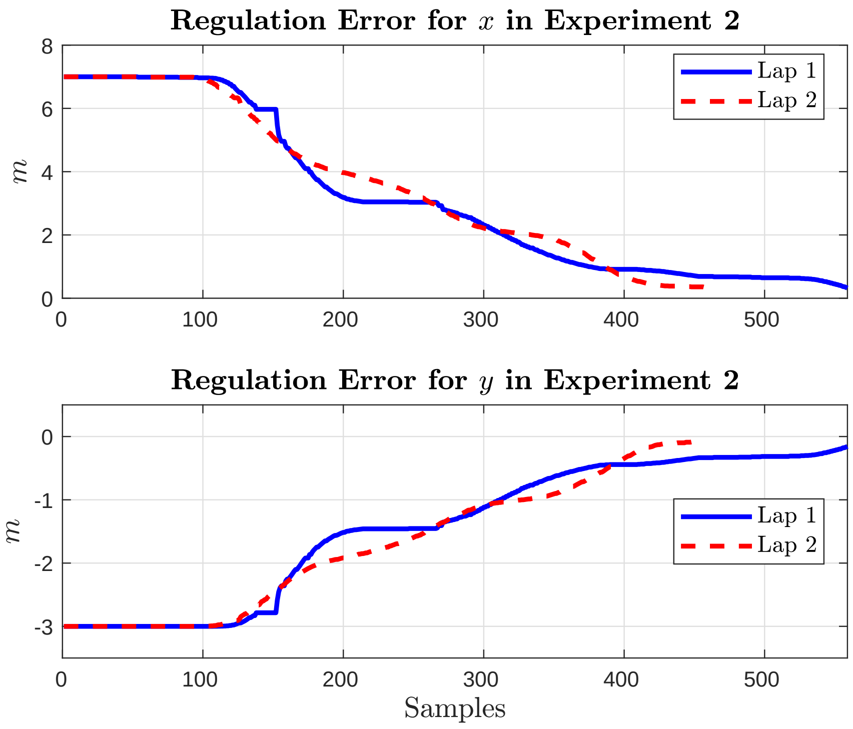


Figure 4.26: Regulation error for variable $x$ and $y$ in Experiment 2.

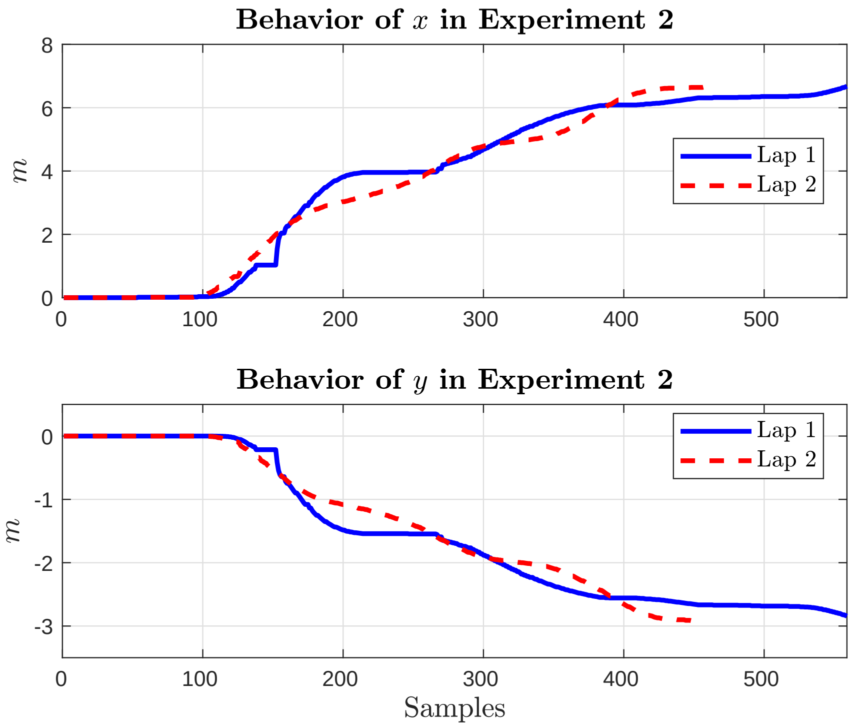


Figure 4.27: Behavior of state variable $x$ and $y$ in Experiment 2.

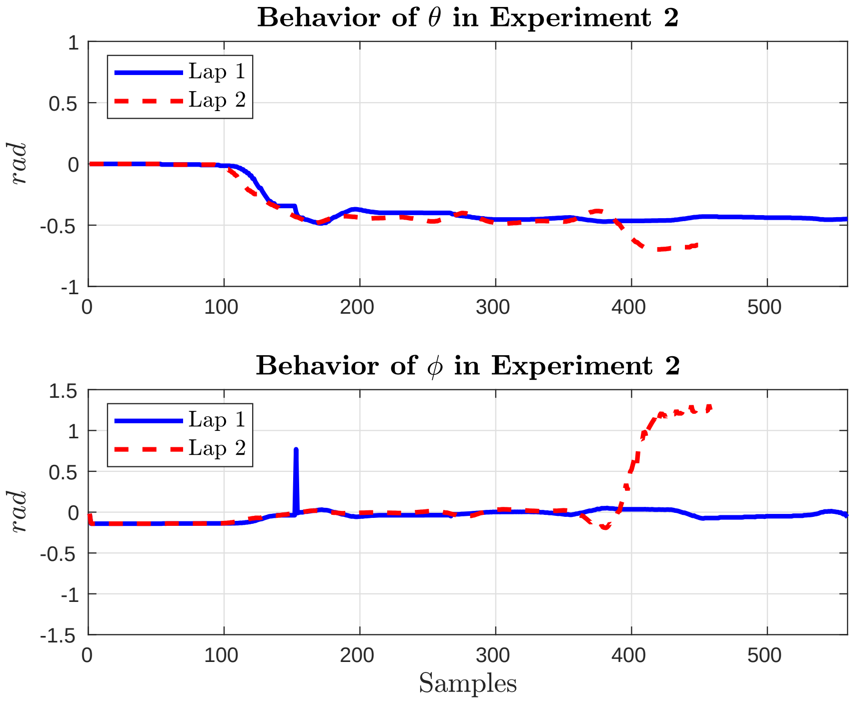


Figure 4.28: Behavior of state variable $\theta$ and $\phi$ in Experiment 2.

Figure 4.29: Control signal for $v$ and $\omega$ in Experiment 2.

In/Out technique can be applied in tracking tasks for the cases where the robot orientation is not required. For the case where the position and velocity references are available and perfectly known, the PD control is the best control approach. However, if the system needs more robustness or some reference data is not available (e.g. desired velocity), the STA is able to control the robot in tracking tasks, ensuring finite time convergence, as can be seen in the presented simulations.

# 5
# Concluding Remarks and Perspectives

In this section, we present the concluding remarks about modeling, control design, constructive aspects and prototyping of wheeled mobile robots for agricultural fields. We have also included a brief discussion of short and long terms perspectives for future work, following the ideas presented here.

## 5.1
## Concluding Remarks

In this work, we have presented a modeling and control design methodology for a class of wheeled mobile robots, capable of performing monitoring and inspection tasks on agricultural fields. We also have shown the details about the electro-mechanical design, constructive aspects, development of embedded electronics, software architecture, communication infra-structure and implementation issues for such robots. The ROS framework has been used to provide communication protocols, sensors integration, robot-motor-driver interface and graphical user interface for robot teleoperation.

The modeling and control design is based on a kinematic approach due its well-know satisfactory performance when the robot motions are executed with low speed and slow accelerations. Different control strategies have been implemented in order to ensure the asymptotic convergence of the tracking error to zero in the presence of external disturbances: the differential-drive mobile robot, named SoyBot, uses a sliding mode control (SMC) approach based on a proper coordinates transformation to chained forms; the car-like mobile robot, named iCat, employs a robust input-output linearization approach based on the super-twisting algorithm for tracking tasks.

Numerical simulations in MATLAB and 3D computer simulations in Gazebo/Rviz were presented to illustrate the performance and feasibility of the proposed methodology. Verification and validation of such methodology were carried out through experimental tests in agricultural fields. In these experiments, a visual servoing approach has been used to align the SoyBot robot to the crop rows, while maintains its linear velocity constant, enabling the autonomous navigation for monitoring and inspection tasks. On the other hand, a kinematic-based control approach has been used to move the iCat robot toward waypoints for simple data collection.

## 5.2 Perspectives

The following research topics may be addressed as future work according to the theoretical and practical developments presented here:

- To develop a rigorous stability analysis based on the Lyapunov stability theory for the robust control techniques introduced in Chapters 3 and 4, which are based on sliding mode control approach and super-twisting algorithm;

- To investigate other collision avoidance algorithms in order to avoid the local minima problem which occurs very often in Artificial Potential Fields algorithms. Some promising solutions may be considered such as the ramdomized best-first algorithm, which is developed for evading local minima, and navigation functions, which are based on artificial potentials that have no local minima.

- To consider the robot dynamics in the control design approach, allowing for the execution of agricultural tasks with fast speeds and rapid accelerations.

  The computation of the dynamic model of a mobile robot is similar to the robot manipulator case. The main difference is the presence of nonholonomic constraints on the generalized coordinates. An important consequence of nonholonomy is that exact linearization of the dynamic model via feedback is no longer possible, in contrast to the kinematic model. Here, we consider an $n$-dimensional mechanical system subject to $k<n$ kinematic constraints in the Pfaffian form $A^{\mathsf{T}}(q)\,\dot{q}=0$ where $A(q)\in\mathbb{R}^{k\times n}$ is the matrix characterizing the kinematic constraints, $q\in\mathbb{R}^n$ is the generalized coordinates.

  As usual, let us define the Lagrangian $\mathcal{L}$ of the mechanical system as the difference between its kinetic $\mathcal{T}$ and potential energy $\mathcal{U}$ as:

$$\mathcal{L}(q,\dot{q}) = \mathcal{T}(q,\dot{q}) - \mathcal{U}(q) = \frac{1}{2}\dot{q}^{\mathsf{T}}M(q)\dot{q} - \mathcal{U}(q)\,, \tag{5-1}$$

  where $M(q)=M^{\mathsf{T}}(q)>0$ is the inertia matrix of the mechanical system. In this case, the Lagrange equations take the form:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{q}}\right)^{\mathsf{T}} - \left(\frac{\partial\mathcal{L}}{\partial q}\right)^{\mathsf{T}} = S(q)\,\tau + A(q)\,\lambda\,, \tag{5-2}$$

  where $S(q)\in\mathbb{R}^{n\times m}$ is an matrix mapping the $m=n-k$ external inputs $\tau$ to generalized coordinates $q$, $\lambda\in\mathbb{R}^m$ is the vector of Lagrange multipliers.

The dynamic model of the constrained mechanical system is expressed as:

$$\begin{aligned} M(q)\ddot{q} + n(q,\dot{q}) &= S(q)\,\tau + A(q)\,\lambda\,, & (5\text{-}3)\\ A^{\mathsf{T}}(q)\,\dot{q} &= 0\,, & (5\text{-}4)\end{aligned}$$

where

$$n(q,\dot{q}) = \dot{M}(q)\,\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}(\dot{q}^{\mathsf{T}}M(q)\dot{q})\right)^{\mathsf{T}} + \left(\frac{\partial\mathcal{U}(q)}{\partial q}\right)^{\mathsf{T}}\,. \qquad (5\text{-}5)$$

# Bibliography references

[1] YOSHIDA, K.; WILCOX, B.; HIRZINGER, G. ; LAMPARIELLO, R.. **Space robotics**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1423–1462. Springer, 2016.

[2] SAIDI, K. S.; BOCK, T. ; GEORGOULAS, C.. **Robotics in construction**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1493–1520. Springer, 2016.

[3] TREVELYAN, J.; HAMEL, W. R. ; KANG, S.-C.. **Robotics in hazardous applications**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1521–1548. Springer, 2016.

[4] MARSHALL, J. A.; BONCHIS, A.; NEBOT, E. ; SCHEDING, S.. **Robotics in mining**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1549–1576. Springer, 2016.

[5] BERGERMAN, M.; BILLINGSLEY, J.; REID, J. ; VAN HENTEN, E.. **Robotics in agriculture and forestry**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1463–1492. Springer, 2016.

[6] TAYLOR, R. H.; MENCIASSI, A.; FICHTINGER, G.; FIORINI, P. ; DARIO, P.. **Medical robotics and computer-integrated surgery**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1657–1684. Springer, 2016.

[7] VAN DER LOOS, H. M.; REINKENSMEYER, D. J. ; GUGLIELMELLI, E.. **Rehabilitation and health care robotics**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1685–1728. Springer, 2016.

[8] PRASSLER, E.; KOSUGE, K.. **Domestic robotics**. Springer handbook of robotics, p. 1253–1281, 2008.

[9] NARDI, D.; ROBERTS, J.; VELOSO, M. ; FLETCHER, L.. **Robotics competitions and challenges**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1759–1788. Springer, 2016.

[10] LIMA, C. E. O.; DE ARAÚJO, F. M. A.; DA SILVA JÚNIOR, M. B. ; ROCHA FILHO, A. E.. **Megabots Team: A Team for the Category Sumo Autonomous 3 kg**. Journal of Mechanics Engineering and Automation 6, p. 254–259, 2016.

[11] TREVISAN, R.; DE SV JUNIOR, N.; PORTZ, G.; EITELWEIN, M. ; MOLIN, J.. **Use of crop height and optical sensor readings to predict mid-season cotton biomass**. In: PRECISION AGRICULTURE'15, p. 1983–1991. Wageningen Academic Publishers, 2015.

[12] EITELWEIN, M. T.. **Sensoriamento proximal de solo para a quantificação de atributos químicos e físicos**. PhD thesis, Universidade de São Paulo, 2017.

[13] BILLINGSLEY, J.; VISAL, A. ; DUNN, M.. **Robotics in Agriculture and Forestry**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1065–1077. Springer Berlin-Heidelberg, 2008.

[14] EDAN, Y.. **Food and Agriculture Robotics**. In: Nof, S. Y., editor, Handbook of Industrial Robotics, p. 1143–1155. John Wiley & Sons, Inc., 2007.

[15] NOGUCHI, N.; ZHANG, Q.; HAN, S. ; REID, J. F.. **Autonomous Agricultural Tractor with an Intelligent Navigation System**. In: PROC. OF THE 4TH IFAC SYMPOSIUM ON INTELLIGENT AUTONOMOUS VEHICLES, p. 197–202, Sapporo, Japan, 2001.

[16] HERNÁNDEZ, J. D.; SANZ, D.; BARRIENTOS, J.; VALENTE, J.; DEL CERRO, J. ; BARRIENTOS, A.. **Non Invasive Moisture Measurement in Agricultural Fields using a Rolling Spherical Robot**. In: PROC. OF THE FIRST INTERNATIONAL CONFERENCE ON ROBOTICS AND ASSOCIATED HIGH-TECHNOLOGIES AND EQUIPMENT FOR AGRICULTURE, p. 229–234, Pisa, Italy, 2012.

[17] PURI, V.; NAYYAR, A. ; RAJA, L.. **Agriculture Drones: A Modern Breakthrough in Precision Agriculture**. Journal of Statistics and Management Systems, 20(4):507–518, 2017.

[18] BECHAR, A.; VIGNEAULT, C.. **Agricultural Robots for Field Operations: Concepts and Components**. Biosystems Engineering, 149:94–111, 2016.

[19] FREITAS, G. M.; HAMNER, B.; BERGERMAN, M. ; SINGH, S.. **A Practical Obstacle Detection System for Autonomous Orchard Vehicles**. In: PROC. OF THE IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, p. 3391–3398, Vilamoura, Portugal, 2012.

[20] FREITAS, G. M.; LIZARRALDE, F.; HSU, L. ; DOS REIS, N. R. S.. **Kinematic Reconfigurability of Mobile Robots on Irregular Terrains**. In: PROC. OF THE IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 1340–1345, Kobe, Japan, 2009.

[21] GRIMSTAD, L.; FROM, P. J.. **Thorvald II - a Modular and Reconfigurable Agricultural Robot**. In: PROC. OF THE 20TH IFAC WORLD CONGRESS, p. 4588–4593, Toulouse, France, Jul 2017.

[22] CRUZ, D.; MCCLINTOCK, J.; PERTEET, B.; ORQUEDA, O. A.; CAO, Y. ; FIERRO, R.. **Decentralized cooperative control-a multivehicle platform for research in networked embedded systems**. IEEE control systems, 27(3):58–78, 2007.

[23] BILLINGSLEY, J.; VISALA, A. ; DUNN, M.. **Robotics in agriculture and forestry**. Springer handbook of robotics, p. 1065–1077, 2008.

[24] PARRA, J. R. P.. **Biological control in Brazil: an overview**. Scientia Agricola, 71(5):420–429, 2014.

[25] EDAN, Y.; HAN, S. ; KONDO, N.. **Automation in Agriculture**. In: Nof, S., editor, SPRINGER HANDBOOK OF AUTOMATION, p. 1095–1128. Springer, Berlin, Heidelberg, 2009.

[26] BERGERMAN, M.; BILLINGSLEY, J.; REID, J. ; VAN HENTEN, E.. **Robotics in Agriculture and Forestry**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1463–1492. Springer-Verlag, Berlin, Heidelberg, 2016.

[27] DERRICK, J. B.; BEVLY, D. M.. **Adaptive Steering Control of a Farm Tractor with Varying Yaw Rate Properties**. Journal of Field Robotics, 26(6-7):519–536, 2009.

[28] KEYSER, M.; MŰLLER, I. A.; CILLIERS, F. P.; NEL, W. ; GOUWS, P. A.. **Ultraviolet radiation as a non-thermal treatment for the inactivation of microorganisms in fruit juice**. Innovative Food Science & Emerging Technologies, 9(3):348–354, 2008.

[29] EDAN, Y.; HAN, S. ; KONDO, N.. **Automation in agriculture**. In: SPRINGER HANDBOOK OF AUTOMATION, p. 1095–1128. Springer, 2009.

[30] BECHAR, A.; VIGNEAULT, C.. **Agricultural Robots for Field Operations. Part 2: Operations and Systems**. Biosystems Engineering, 153:110–128, 2017.

[31] GRIMSTAD, L.; FROM, P. J.. **Thorvald ii-a modular and reconfigurable agricultural robot**. IFAC-PapersOnLine, 50(1):4588–4593, 2017.

[32] BAWDEN, O.; KULK, J.; RUSSELL, R.; MCCOOL, C.; ENGLISH, A.; DAYOUB, F.; LEHNERT, C. ; PEREZ, T.. **Robot for Weed Species Plant-specific Management**. Journal of Field Robotics, 34(6):1179–1199, 2017.

[33] BARRIENTOS, A.; COLORADO, J.; CERRO, J. D.; MARTINEZ, A.; ROSSI, C.; SANZ, D. ; VALENTE, J.. **Aerial Remote Sensing in Agriculture: A Practical Approach to Area Coverage and Path Planning for Fleets of Mini Aerial Robots**. J. Field Robot., 28(5):667–689, 2011.

[34] CARIOU, C.; LENAIN, R.; THUILOT, B. ; BERDUCAT, M.. **Automatic Guidance of a Four-Wheel-Steering Mobile Robot for Accurate Field Operations**. Journal of Field Robotics, 26(6-7):504–518, 2009.

[35] TABILE, R. A.; GODOY, E. P.; PEREIRA, R. R. D.; TANGERINO, G. T.; PORTO, A. J. V. ; INAMASU, R. Y.. **Design and Development of the Architecture of an Agricultural Mobile Robot**. Engenharia AgrÃcola, 31:130–142, 2011.

[36] BAK, T.; JAKOBSEN, H.. **Agricultural Robotic Platform with Four Wheel Steering for Weed Detection**. Biosystems Engineering, 87(2):125–136, 2004.

[37] PRETER, A. D.; ANTHONIS, J. ; BAERDEMAEKER, J. D.. **Development of a Robot for Harvesting Strawberries**. In: PROC. OF THE 6TH IFAC CONFERENCE ON BIO-ROBOTICS, p. 14–19, Beijing, China, 2018.

[38] SILWAL, A.; DAVIDSON, J. R.; KARKEE, M.; MO, C.; ZHANG, Q. ; LEWIS, K.. **Design, Integration, and Field Evaluation of a Robotic Apple Harvester**. J. Field Robot., 34(6):1140–1159, 2017.

[39] ENGLISH, A.; ROSS, P.; BALL, D. ; CORKE, P.. **Vision based Guidance for Robot Navigation in Agriculture**. In: PROC. OF THE IEEE INT. CONF. ON ROBOT. AND AUTOM., p. 1693–1698, Hong Kong, China, 2014.

[40] XAUD, M. F.; LEITE, A. C.; BARBOSA, E. S.; FARIA, H. D.; LOUREIRO, G. S. ; FROM, P. J.. **Robotic tankette for intelligent bioenergy agriculture: Design, development and field tests**. arXiv preprint arXiv:1901.00761, 2019.

[41] SICILIANO, B.; SCIAVICCO, L.; VILLANI, L. ; ORIOLO, G.. **Robotics: Modelling, Planning and Control**. Springer-Verlag London, 2009.

[42] SIEGWART, R.; NOURBAKHSH, I. R. ; SCARAMUZZA, D.. **Introduction to Autonomous Mobile Robots**. MIT Press (MA), 2nd edition, 2011.

[43] WEI, X.; JIA, K.; LAN, J.; LI, Y.; ZENG, Y. ; WANG, C.. **Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot**. Optik-International Journal for Light and Electron Optics, 125(19):5684–5689, 2014.

[44] OHTA, Y.-I.; KANADE, T. ; SAKAI, T.. **Color information for region segmentation**. Computer graphics and image processing, 13(3):222–241, 1980.

[45] OTSU, N.. **A threshold selection method from gray-level histograms**. IEEE transactions on systems, man, and cybernetics, 9(1):62–66, 1979.

[46] SHTESSEL, Y.; EDWARDS, C.; FRIDMAN, L. ; LEVANT, A.. **Sliding mode control and observation**. Springer, 2014.

[47] CORKE, P.. **Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised**, volumen 118. Springer, 2017.

[48] SAMSON, C.; MORIN, P. ; LENAIN, R.. **Modeling and Control of Wheeled Mobile Robots**. In: Siciliano, B.; Khatib, O., editors, Springer Handbook of Robotics, p. 1235–1266. Springer International Publishing, Cham, 2016.

[49] ENGLISH, A.; ROSS, P.; BALL, D. ; CORKE, P.. **Vision based Guidance for Robot Navigation in Agriculture**. In: PROC. OF THE IEEE INT. CONF. ON ROBOT. AND AUTOM., p. 1693–1698, Hong Kong, China, 2014.

[50] FONTAINE, V. AND CROWE, T. G.. **Development of Line-detection Algorithms for Local Positioning in Densely Seeded Crops**. Can. Biosyst. Eng., 48:7.19–7.29, 2006.

[51] ÅSTRAND, B.; BAERVELDT, A.-J.. **A vision based row-following system for agricultural field machinery**. Mechatronics, 15(2):251–269, 2005.

[52] GONZALEZ, R. C.; WOODS, R. E.. **Digital Image Processing**. Pearson Prentice Hall, 3rd edition, 2009.

[53] MURRAY, R. M.. **A mathematical introduction to robotic manipulation**. CRC press, 2017.

[54] EDWARDS, C.; SPURGEON, S.. **Sliding mode control: theory and applications**. Crc Press, 1998.

[55] PARKER, L. E.; RUS, D. ; SUKHATME, G. S.. **Multiple mobile robot systems**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1335–1384. Springer, 2016.

[56] CRUZ, D.; MCCLINTOCK, J.; PERTEET, B.; ORQUEDA, O. A.; CAO, Y. ; FIERRO, R.. **Decentralized cooperative control-a multivehicle platform for research in networked embedded systems**. IEEE Control Systems Magazine, 27(3):58–78, 2007.

[57] MA'SUM, M. A.; JATI, G.; ARROFI, M. K.; WIBOWO, A.; MURSANTO, P. ; JATMIKO, W.. **Autonomous quadcopter swarm robots for object localization and tracking**. In: MHS2013, p. 1–6. IEEE, 2013.

[58] ALVISSALIM, M. S.; ZAMAN, B.; HAFIZH, Z. A.; MA'SUM, M. A.; JATI, G.; JATMIKO, W. ; MURSANTO, P.. **Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area**. In: 2012 PROCEEDINGS OF SICE ANNUAL CONFERENCE (SICE), p. 2256–2261. IEEE, 2012.

[59] HÖNIG, W.; AYANIAN, N.. **Flying multiple UAVs using ROS**. In: ROBOT OPERATING SYSTEM (ROS), p. 83–118. Springer, 2017.

[60] CHOSET, H. M.; HUTCHINSON, S.; LYNCH, K. M.; KANTOR, G.; BURGARD, W.; KAVRAKI, L. E. ; THRUN, S.. **Principles of robot motion: theory, algorithms, and implementation**. MIT press, 2005.

[61] REDCAT. **Instruction Manual, 1/10 Electric Monster Truck, Model No.: Ground Pounder**. Redcat Racing, http://www.redcatracing.com/, 2017. Last access: Mar, 10th 2018.

[62] FRANCIS, B. A.; MAGGIORE, M.. **Models of Mobile Robots in the Plane**. In: FLOCKING AND RENDEZVOUS IN DISTRIBUTED ROBOTICS. SPRINGERBRIEFS IN ELECTRICAL AND COMPUTER ENGINEERING, p. 7–23. Springer, Cham, 2016.

[63] MORENO, J. A.; OSORIO, M.. **Strict Lyapunov functions for the super-twisting algorithm**. IEEE transactions on automatic control, 57(4):1035–1040, 2012.

[64] HAMERLAIN, F.; FLOQUET, T. ; PERRUQUETTI, W.. **Experimental tests of a sliding mode controller for trajectory tracking of a car-like mobile robot**. Robotica, 32(1):63–76, 2014.

[65] KOUROS, G.; PETROU, L.. **PANDORA Monstertruck: A 4WS4WD Car-like Robot for Autonomous Exploration in Unknown Environments**. In: PROC. OF THE 12TH IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS, p. 974–979, Siem Reap, Cambodia, 2017.