Pontifícia Universidade Católica
do Rio de Janeiro

Gustavo Bertagna Peixoto Barbosa

Robust Vision-based Autonomous Crop Row Navigation for Wheeled Mobile Robots in Sloped and Rough Terrains

Dissertação de Mestrado

Dissertation 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. Eduardo Costa da Silva
Co-advisor: Prof. Antonio Candea Leite

Rio de Janeiro
April 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Gustavo Bertagna Peixoto Barbosa

Robust Vision-based Autonomous Crop Row Navigation for Wheeled Mobile Robots in Sloped and Rough Terrains

Dissertation 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 Committe:

Prof. Eduardo Costa da Silva
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Antonio Candea Leite
Co-advisor
Norwegian University of Life Sciences – NMBU

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

Prof. Wouter Caarls
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Vivian Suzano Medeiros
Centro de Pesquisa em Tecnologia de Inspeção – PUC-Rio

Rio de Janeiro, April the 5th, 2022

Gustavo Bertagna Peixoto Barbosa

Undergraduate in Control and Automation Engineering from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio)

Bibliographic data

Bertagna Peixoto Barbosa, Gustavo

Robust Vision-based Autonomous Crop Row Navigation for Wheeled Mobile Robots in Sloped and Rough Terrains / Gustavo Bertagna Peixoto Barbosa; advisor: Eduardo Costa da Silva; co-advisor: Antonio Candea Leite. – Rio de janeiro: PUC-Rio , Departamento de Engenharia Elétrica, 2022.

v., 77 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 Controle e Automação – Teses. 3. Navegação Autônoma de Veículos;. 4. Automação Agrícola;. 5. Controle Não-Linear;. 6. Controle Robusto/Adaptativo;. 7. Simulação em Ambiente Virtual. I. Costa da Silva, Eduardo. II. Candea Leite, Antonio. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 620.11

Acknowledgments

First of all, I would like to thank my supervisors for all the technical and motivational help.

I also thank my family who always supported me to face turbulent times during the making of this material.

I thank my girlfriend for all the support and patience she had with this phase of my life during these years.

I would also like to thank the CNPq agency for the financial support to this research.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior -Brasil (CAPES) -Finance Code 001.

Abstract

Bertagna Peixoto Barbosa, Gustavo; Costa da Silva, Eduardo (Advisor); Candea Leite, Antonio (Co-Advisor). **Robust Vision-based Autonomous Crop Row Navigation for Wheeled Mobile Robots in Sloped and Rough Terrains**. Rio de Janeiro, 2022. 77p. Dissertação de mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

In this work, we present a new application for some robust controllers, such as SMC and STA approaches. The main idea is to perform autonomous navigation in agricultural fields accurately using wheeled mobile robots, equipped with a fixed monocular camera . Here, we consider the existence of uncertainties in the parameters of the robot-camera system and external disturbances caused by high driving velocities, sparse plants, and uneven terrains. First, we design a robust image-based visual servoing approach to deal with model inaccuracies and trajectory perturbations in the image space. In addition, a cascade-based robust control approach is applied, in which the outer vision feedback loop is connected with an inner pose feedback loop to deal with the effects of all disturbances sources. Then, a robust trajectory tracking approach based on the super-twisting algorithm is applied for motion stabilization to ensure the successful execution of row crop following tasks under wheel slippage and vehicle sideslip. ROS-Gazebo platform, an open-source robotics simulator, was used to perform 3D computer simulation using a differential-drive mobile robot and an ad-hoc designed row-crop environment. The effectiveness and feasibility of the robust controllers are evaluated by analyzing numerical simulations and performance metrics, such as: (i) the root-mean square error (RMSE) and (ii) the mean-absolute deviation (MAD). Furthermore, we will see in results, that in general, it is only possible to have stability, using robust controllers.

Keywords

Autonomous Vehicle Navigation; Agricultural Automation; Non-Linear Control; Robust/Adaptive Control; Virtual Environment Simulations

Resumo

Bertagna Peixoto Barbosa, Gustavo; Costa da Silva, Eduardo; Candea Leite, Antonio. **Navegação autônoma em linhas de cultivo baseada em visão robusta para robôs móveis com rodas em terrenos inclinados e acidentados**. Rio de Janeiro, 2022. 77p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Nesse trabalho, nós apresentamos novas aplicações para alguns controladores robustos, tais como as abordagens SMC e STA. O principal objetivo é conseguir executar uma navegação autônoma precisa em campos agrícolas, usando robôs móveis com rodas, equipados com uma câmera monocular fixa. Primeiro, nós projetamos uma abordagem de controle robusto baseado em servo-visão, a fim de lidar com imprecisões do modelo e perturbações da trajetória no espaço da imagem. Além disso, uma abordagem de controle robusto baseada em cascata, é aplicada, na qual, a malha de realimentação externo está conectada com uma malha de realimentação interna para lidar com os efeitos de todas as fontes de perturbação. Desse modo, uma abordagem robusta de rastreamento de trajetória, baseada em super-twisting, é aplicada para estabilização de movimento afim de garantir o sucesso da tarefa de seguir uma linha de cultivo considerando os efeitos de derrapagem das rodas e derrapagem lateral do veículo. A plataforma ROS-Gazebo, um simulador de robótica de código aberto, foi utilizada para realização de simulações computacionais 3D usando um robô móvel do tipo *differential-drive* e um ambiente *ad-hoc* projetado para cultivo em linha. A eficácia e a viabilidade dos controladores robustos são avaliadas analisando simulações numéricas e métricas de desempenho, tais como: (i) o Erro Quadrático Médio (EQM) e (ii) o Desvio Absoluto Médio (DAM). Além disso, nós veremos nos resultados, que em geral, só é possível ter estabilidade, utilizando os controladores rosbustos.

Palavras-chave

Navegação Autônoma de Veículos; Automação Agrícola; Controle Não-Linear; Controle Robusto/Adaptativo; Simulação em Ambiente Virtual

Table of contents

List of figures

List of Table

List of Abreviations

WMRs – Wheeled Mobile Robots
DDMR – Differential-Drive Mobile Robot
IBVS – Image-Based Visual Servoing
IBVS-CC – IBVS Column Controller
IBVS-RC – IBVS Row Controller
rIBVS – Robust Image-Based Visual Servoing
rIBVS-CC – Robust IBVS Column Controller
rIBVS-RC – Robust IBVS Row Controller
STA – Super Twisting Algorithm
rTTC – Trajectory Tracking Control
rCC – Robust Cascade-Based Control
SMC – Sliding Mode Control
FTC – Finite-Time Control
RMSE – Root-Mean Square Error
MAD - Mean-Absolute Deviation
EQM - Erro Quadrático Médio
DAM - Desvio Médio Absoluto

List of Symbols

$\mathcal{L}$ – Best-fit line
$\mathcal{P}$ – Image feature point
$\mathcal{P}^*$ – Desired image feature point
Θ – Image feature angle
$\mathcal{F}_v$ – Image frame
$\mathcal{F}_d$ – Reference framee
θ_v – Coordinate of image feature angle Θ
(x_v, y_v) – Coordinates of the image feature point $\mathcal{P}$
s – Image feature vector
ψ – Tilt offset angle
$\mathcal{F}_c$ – Camera frame
t_x – x-coordinate of the camera's optical center
t_y – y-coordinate of the camera's optical center
t_z – z-coordinate of the camera's optical center
$\mathcal{F}_r$ – Robot frame
$\mathcal{F}_w$ – World frame
q – WMR Generalized coordinates
(x, y) – Cartesian coordinates of a given point of the wheel
θ –Wheel orientation with respect to the x-axis of the world frame $\mathcal{F}_w$
$(\dot{x}, \dot{y})$ – Velocity of the wheel point
v – WMR driving velocity
ω – WMR steering velocity
ω_L – Left wheel speed
ω_R – Right wheel speed
r – Wheels radius
d – Distance between the wheels
V_r – Robot velocity
u – Robot inputs
S – Selection matrix
p_c – Coordinates of the image feature point $\mathcal{P}$ expressed in the camera frame
v_c – Linear velocity of the camera expressed in the world frame
ω_c – Angular velocity of the camera expressed in the world frame
(x_{v0}, y_{v0}) – Camera principal point

f – Focal length of the camera lens

(α_x, α_y) – Scaling factors

ρ – Perpendicular distance between the origin of the image frame and the line

$\dot{s}$– Rate of change of the image features

V_c – Camera spatial velocity

L_s – Interaction matrix

ξ_s – Uncertain motion of the image feature

ξ_x – x-coordinate of the uncertain motion of the image feature

ξ_y – y-coordinate of the uncertain motion of the image feature

ξ_θ – θ-coordinate of the uncertain motion of the image feature

$\xi_{s_\mathcal{C}}$ – Uncertain motion of the image feature for the Column Controller

$\xi_{s_\mathcal{R}}$ – Uncertain motion of the image feature for the Row Controller

A_{cr} – Adjoint matrix

t_{cr} – Translation vector from the origin of the robot frame to the origin of the camera frame

R_{cr} – Rotation matrix of the robot frame with respect to the camera frame

T_{cr} – Homogeneous transformation matrix

T_v – First column of the homogeneous transformation matrix

T_ω – Second column of the homogeneous transformation matrix

ω_i – Angular velocity being $i = \{c, \mathcal{R}\}$

$\mathcal{I}$ – Image plane

$x_v^\star$ – x-coordinate of the desired image

$y_v^\star$ – y-coordinate of the desired image

$\theta_v^\star$ – θ-coordinate of the desired image coordinate

$J_{v_\mathcal{C}}$ – Linear Jacobian matrix for the Column Controller

$J_{\omega_\mathcal{C}}$ – Angular Jacobian matrix for the Column Controller

$J_{v_\mathcal{R}}$ – Linear Jacobian matrix for the Row Controller

$J_{\omega_\mathcal{R}}$ – Angular Jacobian matrix for the Row Controller

$\xi_{s_\mathcal{C}}$ – Uncertain motion of the image feature for the Column Controller

$\Lambda_\mathcal{C}$ – Positive definite gain matrix for the Column Controller

$\Lambda_\mathcal{R}$ – Positive definite gain matrix for the Row Controller

v_d – Desired linear velocity

ω_d – Desired angular velocity

u_d – Desired input velocities

$e_\mathcal{C}$ – Image feature error for the Column Controller

$e_\mathcal{R}$ – Image feature error for the Row Controller

e_x – x-coordinate of the image feature error

e_y – y-coordinate of the image feature error

e_θ – θ-coordinate of the image feature error

$\hat{J}^{\dagger}_{\omega i}$ – Pseudo-inverse of the nominal angular Jacobian matrix with $i=\{c,\mathcal{R}\}$

ν_i – Robust control action with $i=\{c,\mathcal{R}\}$

η_i – Disturbance term with $i=\{c,\mathcal{R}\}$

$\tau_{_i}$ – Robustness terms with $i=\{c,\mathcal{R}\}$

p – Index for the robust approach

$\tau_{_c}$ – Robustness terms for the Column Controller

$\tau_{_\mathcal{R}}$ – Robustness terms for the Row Controller

$\zeta_c, \zeta_\mathcal{R}$ – STA terms for the rIBVS approach

$K_{_c}$ – Positive definite gain matrix for rIBVS-CC

$K_{_\mathcal{R}}$ – Positive definite gain matrix for rIBVS-RC

$\mu_{_c}$ – Positive definite gain matrix for rIBVS-CC

$\mu_{_c}$ – Positive definite gain matrix for rIBVS-RC

$\sigma_{_c}$ – Sliding Surface for rIBVS-CC

$\sigma_{_\mathcal{R}}$ – Sliding Surface for rIBVS-RC

$\Gamma_{_c}$ – Positive definite gain matrix for rIBVS-CC

$\Gamma_{_c}$ – Positive definite gain matrix for rIBVS-RC

W_r – Robot width

L_r – Robot length

H_r – Robot height

H – Image plane height

W – Image plane width

d_f – Front wheels diameter

d_b – Castor wheels diameter

$D(s)$ – Transfer function of the unmodeled dynamics

k – Numerator of transfer function D(s)

λ – Denominator term of the transfer function $D(s)$

Π – Small term to avoid discontinuities in the UVC control law

T_{max} – Simulation time

γ – Slip ratio

γ_L – Slip ratio of the left wheel

γ_R – Slip ratio of the right wheel

v_x – Logitudinal travelling speed of the robot wheels

v_y – Lateral travelling speed of the robot wheels

ν – Linear speed of the wheel

$\dot{\phi}$ - Angular speed of a given wheel (left or right wheel)

κ – Slipping coefficient

β – Slipping angle

δ – Disturbance Term

q_d – Desired trajectory generated by a virtual reference DDMR

$\tilde{q}$ – Cartesian tracking error

x_d – x-coordinate of the desired reference trajectory

y_d – y-coordinate of the desired reference trajectory

θ_d – θ-coordinate of the desired reference trajectory

z – Chained form coordinates vector

z_1 – z_1-coordinate of the chained form system

z_2 – z_2-coordinate of the chained form system

z_3 – z_3-coordinate of the chained form system

T_{zq} – Coordinate transformation matrix system

T_{uv} – Input transformation matrix

(u_1, u_2) – Inputs of the chained form system

(f_1, f_2, f_3) – Bounded disturbances of the chained form system

b – Positive constant gain

α – Non-negative variable gain

α_0 – Positive constant

ϵ – Positive constant which defines the radius of a ϵ-sphere

w_1, w_2 – Robustness terms designed for the rCC approach

ρ_{11}, ρ_{12} – Positives gains for rCC

ρ_{21}, ρ_{22} – Positives gains for rCC

ζ_1, ζ_2 – STA terms for the rCC approach

σ – Sliding surface for the rCC approach

Δ_{si} –Upper bound of a disturbance in the rIBVS system

ι –Upper bound of a disturbance in the rIBVS system

Υ – Upper bound of a disturbance in the rIBVS system

J_{mi} – Lower bound of the term $J_{\omega i}$

J_{mi} – Upper bound of the term $J_{\omega i}$

$V_i(\sigma_i)$ – Lyapunov candidate function

1
Introduction

1.1
Motivation

The growth of agricultural productivity has been driven since the last century, making farms larger and more complex. In this context, precision agriculture is a new trend of farming development around the world. Consequently, agriculture has demanded technological innovation, such as intelligent machinery with decision-making capabilities, genetically modified seeds, and highly accurate sensors for monitoring the cultivation fields [2, 3]. Hence, an important branch of precision agriculture, autonomous navigation technologies are becoming a very important research field, since it can reduce environmental impacts , lower production costs [4], and minimize herbicide waste to weed control [5]. Therefore, agricultural robots are a feasible and affordable solution to increase crop yields, help with labor shortages, and prevent crop losses [6].

Several wheeled mobile robots (WMRs), equipped with sensor packages and different tools, have been developed to carry out various agricultural tasks. For example, spraying robots as ARA (Fig. 1.1 (a)) and AgBotII (Fig. 1.1 (b)) can be autonomously guided to follow crop rows using machine vision, GPS RTK and other sensors. they use robot arms to move micro-spray nozzles attached to their end-effectors to apply herbicides on the detected weed plants selectively [7, 8].

Figure 1.1: (a) ARA Robot navigating in lettuce harvest; (b) AgBotII performs field trials of weed management in Dalby.

Monitoring robots, as TIBA and SoyBot (Fig. 1.2), can navigate autonomously in sugarcane crop tunnels or visually follow crop rows in soybeans and cotton farms for plant health inspection tasks, mainly carrying out dis-

eases and insect pests inventory [9–11]. Autonomous modular robots, as Thorvald II, can be reconfigurable for most agricultural environments and carry out fruit harvesting in strawberry polytunnels, UV-treatment on tomatoes in greenhouses, plant phenotyping in open fields, among others [12].

Figure 1.2: (a) Soybot, a mobile robot designed for inspection of soy and cotton crops ; (b) Soybot monitoring a cotton crop .

1.2 Review of State of the art

To ensure efficient and safe autonomous navigation in agricultural fields, it is crucial to localize the robot accurately, inside or outside the row crops. Currently, the most common approach for localization is to use RTK GNSS/GPS receivers to guide robots along previously planned routes, for survey and data collection with centimeter accuracy. However, RTK GNSS (Real-Time Kinematic) technology still has a high cost especially for large-scale applications, where a fleet of robots has to operate in large cultivation areas [13,14]. LIDAR and laser scanners combined with cameras have been used to generate topological maps of the row crops as an affordable solution for autonomous navigation [15]. However, simultaneous localization and mapping (SLAM) approaches based on range sensors have to face some challenges such as high computational cost for point cloud/image processing and optimization, accumulation of localization errors over time, and localization failure [16,17]. All these drawbacks have contributed for the development of filtering techniques, to improve the accuracy of the robot localization, for example, Extended Kalman Filter (EKF) [18], Monte-Carlo localization (particle filters) [19] and Unscented Kalman Filter (UKF) [20].

Following this trend, several vision-based autonomous navigation techniques for vehicle guidance have been developed using monocular, stereo and RGB-D cameras [21]. An Image-based Visual Servoing (IBVS) approach for robot navigation in urban environments was introduced in [22], where authors design two primitive vision-based controllers: a row controller, for path following; and column controller for path reaching. These controllers have the

advantage of requiringe fewer computing resources. This IBVS approach was also used for autonomous navigation in unknown indoor environments using an electric wheelchair capable of following corridors and passing through doorways [23]. The previous works, however, were applied for mobile robot control only in partially structured scenarios instead of open agricultural fields, which are commonly subject to large variability in soil properties and crop plant density. In [24], the aforementioned IBVS approach was firstly employed for row crop following and two monocular cameras, mounted back and forth, were used to avoid the need for maneuvers in the headlands.

Within this context, the stabilization and trajectory tracking problems for WMRs have been an extensive field of research in artificial intelligence, robotics and control. Except for omnidirectional robots, most WMRs are generally subject to nonholonomic kinematic constraints [25]. Therefore, these robots cannot be stabilized using smooth feedback control laws and cannot reach all the possible paths in the configuration space [26]. Thereby, some authors have proposed significant waypoints regulation and trajectory tracking control approaches for nonholonomic vehicles considering their dynamic models [27, 28]. In addition, a variety of discontinuous switching controllers based on the backstepping technique [29] and sliding-mode control (SMC) approach [30] have been designed to stabilize nonholonomic systems. Because of the undesired chattering effect caused by the discontinuous nature of such robust controllers, some continuous approximation for switching control laws have been proposed [31]. In [32], for example, the finite-time control (FTC) technique and the super-twisting algorithm (STA) approach [33–35] have been combined for achieving posture stabilization of a WMR.

In general, most autonomous navigation systems for open fields applications consider that WMRs are moving on locally flat and stable terrains [36]. Then, the performance of the navigation algorithms may be degraded when operating in row crops with different soil types, slopes, and roughness. For this reason, controlling mobile robots through rough and uneven terrains is one of the most challenging problems for autonomous navigation in agricultural environments. In such cases, external disturbances in robot trajectory may arise, for example, wheel slippage and sideslip, undermining the performance of the motion control algorithm [37]. Therefore, advanced control techniques for WMRs have been proposed to deal with model inaccuracies and trajectory perturbations caused by non-ideal environments. Wang *et al.* have presented a backstepping controller for path-following based on the kinematic model under wheel slippage and vehicle sideslip [38]. Mobayen has designed a recursive terminal SMC approach for trajectory tracking for disturbed non-

holonomic systems [39]. Bessas *et al.* have developed a novel integral SMC approach combined with non-linear time-varying feedback for trajectory tracking of unicycle robots to deal with matched disturbances and attenuating the unmatched ones [40]. In Martínez *et al.* [41], a robust tracking control approach for unicycle-type mobile robots based on SMC approach and attractive ellipsoid method have been studied to reject disturbances on the kinematic equations, considering the constraints of input saturation.

In addition, a variety of robust controllers have been developed to reject perturbations and deal with unmodeled dynamics in visual servoing systems. [42, 43]. To increase the accuracy of visual tracking tasks for uncertain robot manipulators as well as to cope with uncertainties in camera calibration parameters, Oliveira *et al.* [44] have proposed a Sliding Mode Control (SMC) technique based on the unit vector control approach and a switching monitoring function for the image tracking error. Becerra *et al.* [45] have designed an SMC technique to stabilize the pose of mobile robots equipped with an uncalibrated epipolar camera and to deal with singular configurations of the image Jacobian matrix, allowing the vehicle to move directly toward the target. Furthermore, Becerra *et al.* [46] have introduced an STA-based visual servoing system for mobile robots, which uses uncalibrated camera models to successfully perform tracking tasks of online-generated image trajectories with unknown first-order time-derivative.

1.3 Methodology

This Master's thesis presents and discusses two main techniques to perform autonomous navigation in agricultural environments with the presence of disturbances and uncertainties: a robust image-based visual servoing (rIBVS) approach and a robust cascade-based control (rCC) approach. We include the 3D CAD model of the Soybot robot in a row crop agricultural field developed in Gazebo, a 3D Simulation environment, for evaluating these image-based autonomous navigation solutions in different scenarios. Simulation results are included to ilustrate the performance and fesibility of the proposed rIBVS and rCC.

1.3.1 Robust Image-Based Visual Servoing Approach

The rIBVS approach is developed to increase the accuracy of autonomous navigation tasks in agricultural fields performed by wheeled mobile robots, equipped with a single monocular camera. This method seeks to deal with

model inaccuracies and disturbances due to the weak camera calibration or trajectory perturbations, that appear at high driving velocities. First, an image segmentation technique based on the combination of Hough transform and RANSAC [47], is applied to extract the line features of the crop row in the image space. Such algorithm can be an alternative solution to the machine learning-based methods, due to their well-known requirements regarding the complexity of learning models, computational capacity and dataset quality. Then, a sliding mode control approach is used to include a robustness term into the classic IBVS approach based on two primitive row and column controllers in the image space [22].

1.3.2 Robust Cascade-Based Control Approach

In addition, a robust cascade-based control (rCC) approach for WMRs, with an embedded fixed monocular camera, was developed to carry out autonomous navigation tasks in row crop fields accurately. The designed controller comprises an outer vision feedback loop connected via a cascade control strategy with an inner pose feedback loop, with the aim of rejecting all sources of disturbance. First, the robust image-based visual servoing (rIBVS) approach, based on the STA method, is designed to deal with the uncertainties mentioned in the first objective. Then, a robust trajectory tracking control (rTTC) approach, based on the SMC method, is implemented to provide motion stabilization in the presence of wheel slippage and vehicle sideslip, due to variations in terrain roughness and slope. We employ the Lyapunov stability theory to verify the stability and robustness properties of the overall closed-loop system.

We include the 3D CAD model of the Soybot robot in a row crop agricultural field developed in Gazebo, a 3D simulation environment, for evaluating the image-based autonomous navigation solutions in different scenarios. Simulation results are included to illustrate the performance and feasibility of the proposed robust vision-based controller (rIBVS) and the robust cascade-based controller (rCC).

1.4 Goals and Objectives

The objective of this Master's thesis is to design robust controllers aiming at performing autonomous navigation tasks in different row crop fields acurrately, with a monocular camera. We also seek to carry out simulation tests on the ROS-Gazebo platform to verify and validate the proposed control

approaches. Thereby, a virtual agricultural environment has been created containing irregularities in the terrain and sparse plants, as well as, slopes on the ground. In addition, we hope to apply the designed controllers using real robots, in a challenging agricultural environment, to experimentally validate the theory and simulations developed in the present work.

1.5 Contributions

The first contribution of this Master's Thesis is to apply the vision-based controller, based on column and row primitives, to navigate a wheeled mobile robot in agricultural fields. Originally, these approaches were designed for mobile robots to ride in urban environments only. Thus, it was necessary to use a segmentation algorithm, based on the combination of RANSAC and rough transform algorithms to find the best line that represents the crop row. Then, we modified the column and row primitive controllers to include a robustness term into the original control laws to ensure successful row crop and following tasks. Furthermore, we consider the existence of uncertainties in the parameters of the robot-cameras system and external disturbances caused by high driving velocities, sparse plants, and uneven terrains. In addition, we developed a cascade-based robust control approach to deal with the effects of all disturbances sources. First, we design a robust image-based visual servoing (rIBVS) to deal with model inaccuracies and trajectory perturbations in image space. Secondly, a robust trajectory tracking control is designed for motion stabilization to guarantee the successful performance of r ow crop following under wheel slippage and vehicle sideslip. Another relevant contribution is the development of a 3D agricultural environment to simulate all the behaviors described above. The research conducted along this mater's thesis, allowed the publication of the following articles:

- G. B. P. Barbosa, E. C. Silva and A. C. Leite, "Robust Image-based Visual Servoing for Autonomous Row Crop Following with Wheeled Mobile Robots", 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France, 2021.
- G. B. P. Barbosa, E. C. Silva and A. C. Leite, "Vision-based Autonomous Crop Row Navigation for Wheeled Mobile Robots using Super-twisting Sliding Mode Control", 2021 European Conference on Mobile Robots (ECMR), Bonn, Germany, 2021.

1.6 Dissertation Structure

This Master's thesis is organized as follows:

– **Chapter 2** presents a brief explanation of the main concepts of the robust controllers used in this dissertation, such as: Sliding Mode Control (SMC), Unit Vector Control (UVC) and Super Twisting Algorithm (STA) approaches.

– **Chapter 3** introduces a robust image-based visual servoing (rIBVS) approach for WMRs with a single monocular camera embedded, to carry out autonomous navigation in row crop fields. First, we present the kinematic modeling of the WMR, the camera model and the IBVS system. Next, we describe the vision-based controller, based on column and row primitives. Lastly, we make small changes to these approaches to include the robustness term. To illustrate the effectiveness and feasibility of the proposed vision-based control methodology, 3D computer simulations are executed in the R0S-Gazebo simulator using the 3D CAD model of the SoyBot Robot.

– **Chapter 4** shows a robust cascade-based controller (rCC) approach for mobile robots equipped with a fixed monocular camera. This controller comprises an outer vision feedback loop formed by the rIBVS approach, with an inner pose feedback loop. The purpose of this approach is to deal with all sources of disturbance. Thereby, we consider that a WMR has to navigate autonomously in agricultural fields in the presence of external disturbances caused by the slope and roughness of the terrain. Then, we introduce a coordinate transformation of the kinematic model into the chained form, that faclitates solving problems of planning and control of nonholonomic systems. In addition, a robust trajectory tracking control (rTTC) approach based on the SMC method is implemented to provide motion stabilization in the presence of the disturbances caused by imperfections in the terrain. For checking the robustness and stability of the closed-loop system, 3D computer simulations are executed in the ROS-Gazebo platform, using a differential-drive mobile robot and an ad-hoc designed row crop environment.

– **Chapter 5** describes the conclusion of the developed approaches. Next, we hightlight potential future works that we consider as the most relevant next steps of the reserach.

2
Main Concepts

In this chapter we describe a motivation of using robust controllers, and also present a brief description of the control techniques applied in this dissertation, such as: SMC, UVC and STA approaches.

2.1
Motivation

Generally, in any control problem there are discrepancies between the real model and the mathematical model used to design the control law [31]. These discrepancies can arise from the following reasons: external disturbances, uncertain plant parameters and unmodeled dynamics (underestimation of system order). Thereby, developing controllers that perform the desired task in the presence of inaccuracies/disturbances is a great challenge. The two major approaches to deal with these problems are adaptive and robust controllers [1]. As will be seen in the next sections, the typical structure of robust controllers is composed of a nominal part, similar to a feedback linearization and an additional term for dealing with inaccuracies/disturbances.

2.2
Sliding Mode Control (SMC)

One of the main robust control approaches is so-called sliding control methodology. This technique makes it possible to simplify n^{th}-order problems into an equivalent first-order problem. Therefore, using Sliding Mode Control (SMC) approach in systems with disturbances and inaccuracies, makes possible to achieve the desired performance during the task. However, it is necessary to apply an extremely high control activity.

2.2.1
Sliding Surface

Let's consider the following single-input dynamic system:

$$x^{(n)} = g(\mathbf{x}) + c(\mathbf{x})u\,, \tag{2-1}$$

where $x \in \mathbb{R}$ is the output of interest, $u \in \mathbb{R}$ is the control input and $\mathbf{x} = (x, \dot{x}, ..., x^{(n-1)}) \in \mathbb{R}^{n\times 1}$ is the state vector. In Eq. (2-1), the non-linear function $g(\mathbf{x})$ is not exactly known, but its upper bound and the gain function $c(\mathbf{x})$ are known functions of $\mathbf{x}$. The control problem consists of making the state vector $\mathbf{x}$ to track a specific time-varying state $\mathbf{x}_d = (x_d, \dot{x}_d, ..., x_d^{(n-1)}) \in \mathbb{R}^{n\times 1}$ in presence of model imprecision on $g(\mathbf{x})$ and $c(\mathbf{x})$. In addition, to perform the mentioned trajectory tracking task using the finite control u, the initial condition $\mathbf{x}(0)$ must be defined by:

$$\mathbf{x}(0) = \mathbf{x}_d(0)\,. \tag{2-2}$$

Now, let us define the trajectory tracking error as:

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d\,, \tag{2-3}$$

where $\mathbf{x} \in \mathbb{R}^{n\times 1}$ is the tracking error vector. In turn, considering a time-varying surface $S(t)$ in the state space $\mathbb{R}^{n\times 1}$ by the equation $s(\mathbf{x}, t) \in \mathbb{R}$:

$$s(\mathbf{x}, t) = (\frac{d}{dt} + \lambda)^{n-1}\tilde{x}\,, \tag{2-4}$$

where $\lambda \in \mathbb{R}$ is a strictly positive constant and n is the system order. Thus, given the initial condition defined by Eq. (2-2), the trajectory tracking problem is equivalent to that of stay on the surface $S(t)$ for all $t > 0$, since $s(\mathbf{x}, t) = 0$ represents a linear differential equation with a unique solution $\tilde{x} = 0$. Therefore, the n-dimensional stabilization problem is reduced to keeping $s(\mathbf{x}, t)$ at 0. The scalar $s(\mathbf{x}, t)$ represents a true measure of tracking performance, since the bounds on $s(\mathbf{x}, t)$ can be directly translated into bounds on the tracking error vector $\tilde{x}$. This problem can be solved by choosing the control law u, such that outside of S, satisfies:

$$\frac{1}{2}\frac{d}{dt}s(\mathbf{x}, t)^2 \leq -\Omega|s(\mathbf{x}, t)|\,, \tag{2-5}$$

where Ω is a strictly positive constant. In Eq. (2-5), $s(\mathbf{x}, t)^2$ represents the squared distance to the surface, which decreases along all system trajectory. Thereby, all trajectories are constrained to point to the surface $S(t)$, which means once on the surface, the system remains on the surface. Thus, the sliding condition given by Eq. (2-5) makes the surface an invariant set. Furthermore, we can deal with the perturbations and uncertainties in the model by keeping the surface an invariant set. Another point is that once the sliding surface is reached, the trajectories are defined by:

$$(\frac{d}{dt} + \lambda)^{n-1}\tilde{x} = 0\,. \tag{2-6}$$

Then, with Eq. 2-6, we can confirm the statement that an n-order problem can be replaced by a first-order one. In addition, satisfying the sliding condition, guarantees that if the condition described by Eq. 2-2 is not exactly verified, the surface $S(t)$ will be reached in less time than $|s(t=0)/\Omega|$ as can be seen in [1]. Now, the robust controller design is divided into two steps. The first one is a feedback control law u to verify the sliding condition described in Eq. (2-5). However, due to modeling imprecision and disturbances, we design a discontinuous control law in the second step to achieve robustness. At this step, since the implementation of the associated control switchings is necessarily imperfect, the chattering phenomenon appears.

2.2.2 Filippov's Construction of the Equivalent Dynamics

The movement of the system on the sliding surface can receive an interesting geometry interpretation as an "average" of the system dynamics on both sides of the surface. The dynamic in sliding mode is given by:

$$\dot{s} = 0\,. \tag{2-7}$$

By solving Eq. (2-7) to find the control input u, we obtain the equivalent control u_{eq}. This term can be interpreted as a continuous control law that to keeps $s(\mathbf{x},t)=0$, if the dynamics were known. Now, we consider the following system:

$$\ddot{x} = f + u\,. \tag{2-8}$$

Then, calculating the sliding surface for second-order system according to Eq. (2-4), we have:

$$s = \dot{\tilde{x}} + \lambda\tilde{x}\,, \tag{2-9}$$

Next, taking the time-derivative of the Eq. (2-9) and applying Eq. (2-8), we can found:

$$\dot{s} = f + u - \ddot{x}_d + \lambda\dot{\tilde{x}}\,. \tag{2-10}$$

The equivalent control u_{eq} taking into account if the dynamic of f were known and using Eq. (2-10) to calculate $s(\dot{\mathbf{x}},t)=0$ is given by:

$$u_{eq} = -f + \ddot{x}_d - \lambda\dot{\tilde{x}}\,, \tag{2-11}$$

geometrically, the equivalent control u_{eq} can be built

$$u_{eq} = \alpha u_+ + (1-\alpha)u_-\,. \tag{2-12}$$

Therefore, u_{eq} can be written as a convex combination of the u values on both sides of the surface $S(t)$ The α value can be calculated using Eq. (2-7). that corresponds to requiring system trajectories tangent to the surface. This idea, created by Filippov [1] is illustrated in the Fig. 2.1 where f_+, f_- and f_{eq} are

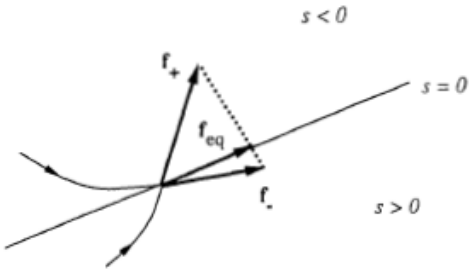

Figure 2.1: Filippov's construction of the equivalent dynamics in sliding mode [1].

given by:

$$\begin{cases} f_+ = \begin{bmatrix} \dot{x} & f + u_+ \end{bmatrix}^T , \\ f_- = \begin{bmatrix} \dot{x} & f + u_- \end{bmatrix}^T , \\ f_{eq} = \begin{bmatrix} \dot{x} & f + u_{eq} \end{bmatrix}^T . \end{cases} \tag{2-13}$$

The sliding mode on the surface corresponds to a limiting behavior. As control switchings occur infinitely fast, the formal solution α of Eq. (2-7) and Eq. (2-12) can be interpreted as the average "residence time" of the trajectory on the side $s(\mathbf{x}, t) > 0$.

2.2.3 A Basic Example

Considering a second-order system given by:

$$\ddot{x} = f + u \,, \tag{2-14}$$

where $u \in \mathbb{R}$ is the control input, $x \in \mathbb{R}$ is the output of interest, and $f \in \mathbb{R}$ s not exactly known but can be estimated as $\hat{f} \in \mathbb{R}$. The estimation error on f is assumed to be bounded by some known function $F(x, \dot{x}) \in \mathbb{R}$, described by

$$|\hat{f} - f| \leq F \,. \tag{2-15}$$

To give a better example, let the system be:

$$\ddot{x} + a(t)\dot{x}^2 \cos 3x = u \,, \tag{2-16}$$

where a(t) is unknown but obeys

$$1 \leq a(t) \leq 2 \,. \tag{2-17}$$

In addition, we have defined that $\hat{f} = -1.5x^2 \cos 3x$ and $F = 0.5\dot{x}^2 |\cos 3x|$. Now, to the system track $x(t) = x_d(t)$, we have a sliding surface $s = 0$, according to Eq. (2-4) given by

$$s = \dot{\tilde{x}} + \lambda \tilde{x} \,. \tag{2-18}$$

Then, taking the time-derivative of Eq. (2-18), we found:

$$\dot{s} = f + u - \ddot{x}_d + \lambda\dot{\tilde{x}}\,. \tag{2-19}$$

The best approximation for the feedback control law $\hat{u}$ that would achieve $\dot{s}=0$ is given by:

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\hat{x}}\,. \tag{2-20}$$

The next step is to add to $\hat{u}$ a discontinuous term to satisfy the sliding condition, defined in Eq. (2-5) despite the uncertainty on f

$$u = \hat{u} - ksgn(s)\,. \tag{2-21}$$

Chosing k large enough, we can ensure that the sliding condition is verified. Applying Eq. (2-21) in Eq. (2-19), we can found

$$\frac{1}{2}\frac{d}{dt}s^2 = (f - \hat{f})s - k|s|\,. \tag{2-22}$$

To guarantee the sliding condition, we have the following value for k:

$$k = F + \Omega\,. \tag{2-23}$$

2.3 Unit Vector Control (UVC)

Once the SMC approach is based on the *sign* function, on the other hand the UVC approach [25], [44] is given by:

$$u = -K\frac{s(\mathbf{x},t)}{|s(\mathbf{x},t)|}\,, \tag{2-24}$$

where $K \in \mathbb{R}$ is a positive constant gain, and $s(\mathbf{x},t)$ is the sliding surface defined in Eq. (2-4). Another difference between the SMC and UVC approaches is that sliding mode for the first one can take place in any individual switching surface before their intersection $x=0$. However, the sliding mode for the UVC approach happens only at the origin.

2.4 Super Twisting Algorithm (STA)

In many cases the high-frequency switching control described in Sections 2.2 and 2.3 is impractical. Thereby, continuous control is necessary. In this section we introduce a controller that follows this approach so-called Super Twisting Algorithm (STA), a second-order sliding mode control. First, let us rewrite Eq.(2-10):

$$\dot{s} = u - \phi\,, \tag{2-25}$$

where $\phi \in \mathbb{R}$ is a disturbance term given by:

$$\phi = -f + \ddot{x}_d - \lambda\dot{\tilde{x}}\,, \tag{2-26}$$

Now the STA control law [31] will be defined as:

$$\begin{cases} u = c|s|^{0.5} sign(s) + w\,, \\ \dot{w} = b\, sign(s)\,, \end{cases} \tag{2-27}$$

where $c \in \mathbb{R}$ and $b \in \mathbb{R}$ are positive gains calculated as:

$$c = 1.5\sqrt{C} \quad b = 1.1C\,, \tag{2-28}$$

assuming that $|\dot{\phi}| \leq C$. The dynamic of the sliding surface s described in Eq. (2-4), applying the control law defined in Eq. (2-28), is given by:

$$\begin{cases} u = c|s|^{0.5} sign(s) + w = \phi\,, \\ \dot{w} = b\, sign(s)\,, \end{cases} \tag{2-29}$$

Thereby, the term w defined in Eq. (2-29) becomes equal to the disturbance ϕ described in Eq. (2-26), in finite time. Therefore, s-dynamic is given by:

$$\dot{s} = c|s|^{0.5} sign(s)\,. \tag{2-30}$$

Consequently, $s \to 0$ in finite time. For the STA approach, the following properties hold:

– The STA approach given by Eq. (2-29) is a second-order sliding mode control, since it drives both $s, \dot{s} \to 0$ in finite time.

– The STA approach given by Eq. (2-29) is continuous since both terms $c|s|^{0.5} sign(s)$ and $w = b \int sign(s)dt$ are continuous. Thus, the high-frequency switching term $sign(s)$ is "hidden" under the integral.

2.5 Final Remarks

In this chapter, we have presented a motivation to apply robust control, and we also have provided details of the approaches that will be used in this work. In the next chapter, we will develop robust image-based visual servoing (rIBVS) approaches, based on SMC and STA, mentioned in this chapter, for mobile robots using a single monocular camera. This methods will be designed to perform autonomous navigation tasks in agricultural fields with high accuracy.

3
Robust Image-based Visual Servoing for Autonomous Row Crop Following

In this chapter, we will introduce robust image-based visual servoing (rIBVS) approaches based on SMC and STA for WMRs, endowed with a single monocular camera to increase the accuracy of the autonomous navigation task in agricultural fields. These approaches can handle model inaccuracies and external disturbances caused by weak camera calibration and sparse plant distributions, as well as trajectory perturbations that arise at high driving velocities. Then, robust controller approaches are designed to add robustness terms to the classic IBVS approach comprising column and row visual primitive controllers [22]. To evaluate the effectiveness and feasibility of the proposed vision-based control methodology, we performed tests with two different techniques of the developed rIBVS. The first is based on SMC approach, and then the STA approach is applied to mitigate the effects of chattering and increase the accuracy of the task performed. 3D Computer simulations are executed in ROS-Gazebo simulator using the 3D CAD model of the Soybot Robot, a differential-drive mobile robot (DDMR) navigating autonomously in an ad-hoc developed row crop agricultural environment.

3.1
Problem Formulation

Here, we consider the problem of row crop following by using a fixed monocular camera attached to the top of a wheeled mobile robot (WMR). Most crops in open fields or gardens are planted in rows or straight lines, either single or multiple rows, mainly to enhance maximum yields and simplify the farm logistics. We assume that the task of interest consists of autonomously guiding the robot over a single crop row, keeping its wheels in the midpoint between two adjacent rows A differential-drive mobile robot, called Soybot [10] designed to carry out plant-health monitoring tasks in soybeans and cotton crops, is used for autonomous visual navigation in the presence of uncertainties and disturbances in the camera-robot system's parameters. During the row crop following, RGB images are continuously captured by the vision sensor. An image segmentation method, based on the combination of RANSAC and

Hough transform algorithms [47], is then applied to find the best-fit line $\mathcal{L}$ to represent the crop row and extract line features. Then, the following 2D image feature set $\mathcal{L}=\{\mathcal{P},\Theta\}$ is computed in the image frame $\mathcal{F}_v$: (i) $\theta_v \in (-\pi,\pi]$ is the coordinate of image feature angle Θ, denoting the angle between the line and the negative y_v-axis; (ii) $(x_v, y_v)\in\mathbb{R}^2$ are the coordinates of the image feature point $\mathcal{P}$, denoting the bottom (top) pixel row or left (right) pixel column. Therefore, the image feature set can be denoted by the image feature vector $s=(x_v, y_v, \theta_v)\in\mathbb{R}^{3\times 1}$.

Afterward, a robust IBVS approach is designed to enable the robot to visually reach and follow a path on the ground defined by the crop rows [24]. An uncalibrated camera is mounted with a tilt offset angle $\psi\in(-\frac{\pi}{2},\frac{\pi}{2}]$ with respect to the x-axis of the camera frame $\mathcal{F}_c$. In addition, the camera's optical center is positioned at the coordinates $(t_x, 0, t_z)\in\mathbb{R}^3$ with respect to the robot frame $\mathcal{F}_r$, as shown in Fig. 3.1.

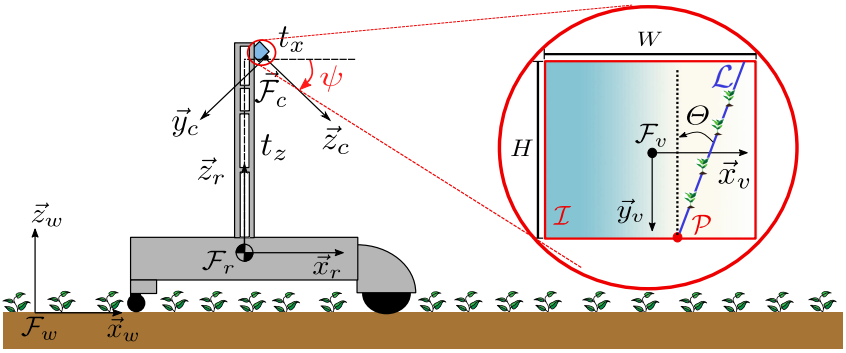

Figure 3.1: The Soybot robot and its coordinate frames: $\mathcal{F}_v$ is the image frame and $s=(x_v, y_v, \theta_v)\in\mathbb{R}^3$ are the image features vector. Side view of the robot: $\mathcal{F}_r$ is the robot frame, $\mathcal{F}_w$ is the world frame, $\mathcal{F}_c$ is the camera frame with a camera tilt offset ψ.

For designing the rIBVS controller for real-world agricultural applications, the following assumptions are considered: (A1) the robot is located outside the crop row, so that both column and row primitives may be included into the control design; (A2) the image processing algorithm uses a region of interest (RoI) to find the best-fit line denoting the crop row to be monitored; (A3) the mobile robot can travel with high driving velocities inside the crop field; and (A4) the terrain is rough and has plants sparsely distributed.

Then, a robustness term will be included in the classic IBVS controller, based on row and column primitives [22], to cope with the external perturbations caused by assumptions (A3) and (A4). The new rIBVS controller is split into twofold: a robust IBVS column controller (rIBVS-CC), to reach out the crop rows and maneuver the robot; (ii) a robust IBVS row controller (rIBVS-RC), to keep the robot following the path after reaching it.

3.2 Kinematic Robot Model

The unicycle robot is a rigid body with a single orientable wheel and its configuration is described by the generalized coordinates $q = (x, y, \theta) \in \mathbb{R}^{3\times1}$, wherein $(x, y) \in \mathbb{R}^{2\times1}$ is the Cartesian coordinates of a given point of the wheel and $\theta \in \mathbb{R}$ is the wheel orientation with respect to the x-axis of the world frame $\mathcal{F}_w$. WMRs are subject to kinematic constraints, integrable or not (nonholonomic constraints), which reduce the mobility of the mechanical structure. For instance, the motion of a unicycle robot is characterized by a pure rolling constraint given by:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0\,, \tag{3-1}$$

which means that the point velocity $(\dot{x}, \dot{y}) \in \mathbb{R}^{2\times1}$ is zero in the direction orthogonal to the sagittal axis of the wheel [25]. In this context, the kinematic model of the unicycle robot is expressed in terms of the null space of the kinematic constraint as:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix} v \; + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix} \omega\,, \tag{3-2}$$

where the robot inputs $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ are the driving and steering velocities respectively. However, a single wheeled robot has serious problems of balance in static conditions. That is why there are vehicles kinematically equivalent to a unicycle robot that use two or more wheels to provide higher mechanical stability, such as the DDMR. Its configuration is described by the same generalized coordinates of the unicycle robot, where $(x, y) \in \mathbb{R}^{2\times1}$ is the Cartesian coordinates of the midpoint between the two wheel centers and θ is the common orientation of the fixed wheels (Fig. 3.2).

Hence, the kinematic model in Eq. (3-2) is also applied to the DDMR and the robot inputs $(v, \omega) \in \mathbb{R}^{2\times1}$ can be expressed as a function of the angular speeds $(\omega_L, \omega_R) \in \mathbb{R}^{2\times1}$ of the left and right wheels respectively, as given by:

$$v = \frac{r\,(\omega_R + \omega_L)}{2}\,, \qquad \omega = \frac{r\,(\omega_R - \omega_L)}{d}\,, \tag{3-3}$$

where $r > 0$ is the wheels radius and $d > 0$ is a distance between their centers [25].

Then, the robot velocity $V_r = (v, 0, 0, 0, 0, \omega) \in \mathbb{R}^{6\times1}$ can be related to the robot inputs $u = (v, \omega) \in \mathbb{R}^{2\times1}$ using the following relationship:

$$V_r = S\,u\,, \qquad S = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^{\mathsf{T}}\,, \tag{3-4}$$

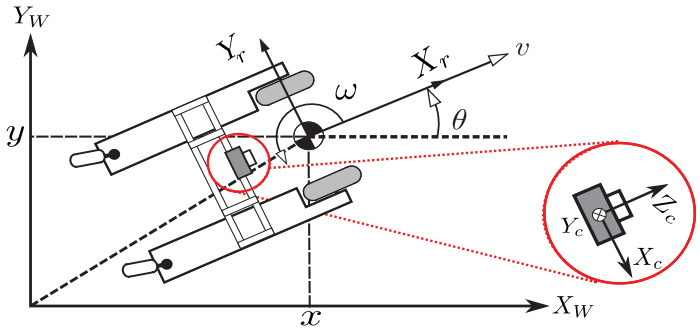

Figure 3.2: Generalized coordinates $q = (x, y, \theta) \in \mathbb{R}^{3\times1}$ and the robot inputs $u = (v, \omega) \in \mathbb{R}^{2\times1}$ for a differential-drive mobile robot.

where $S \in \mathbb{R}^{6\times2}$ is a selection matrix.

3.3 Camera Model

Consider a pin-hole camera observing a line $\mathcal{L}$ with a set of image features $(\mathcal{P}, \Theta)$, as described in Section 3.1. Then, let $p_c = (x_c, y_c, z_c) \in \mathbb{R}^{3\times1}$ be the coordinates of the image feature point $\mathcal{P}$ expressed in the camera frame $\mathcal{F}_c$. The velocity of $\mathcal{P}$ expressed in frame $\mathcal{F}_c$ is given by:

$$\dot{p}_c = -\omega_c \times p_c - v_c \,, \tag{3-5}$$

where $v_c \in \mathbb{R}^{3\times1}$ and $\omega_c \in \mathbb{R}^{3\times1}$ are respectively the linear and angular velocities of the camera expressed in the world frame $\mathcal{F}_w$ [48]. The image feature point $\mathcal{P}$ can be expressed in the image frame $\mathcal{F}_v$ with its projected coordinates $(x_v, y_v) \in \mathbb{R}^{2\times1}$, using the following relationship:

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} \alpha_x & 0 \\ 0 & \alpha_y \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_{v0} \\ y_{v0} \end{bmatrix} \,, \tag{3-6}$$

where $(x_{v0}, y_{v0}) \in \mathbb{R}^{2\times1}$ is the camera principal point, $f > 0$ is the focal length of the camera lens, and $\alpha_x, \alpha_y > 0$ are the scaling factors. A line feature $\mathcal{L}$ can be parameterized using a pair (ρ, θ_v) described by:

$$y_v = -x_v \, \tan(\theta_v) + \rho / \cos(\theta_v) \,, \tag{3-7}$$

where $\rho \in [-\rho_{min}, \rho_{max}]$ is the perpendicular distance between the origin of the image frame $\mathcal{F}_v$ and the line. The rate of change of the image features vector $s = (x_v, y_v, \theta_v) \in \mathbb{R}^{3\times1}$ is related to the camera spatial velocity $V_c = (v_c, \omega_c) \in \mathbb{R}^{6\times1}$, as given by [48]:

$$\dot{s} = L_s \, V_c + \xi_s \,, \tag{3-8}$$

where $L_s = [\, L_x \;\; L_y \;\; L_\theta \,]^\mathsf{T} \in \mathbb{R}^{3\times6}$ is the so-called interaction matrix, and $\xi_s = (\xi_x \,, \xi_y \,, \xi_\theta) \in \mathbb{R}^{3\times1}$ denotes the uncertain motion of the image feature, which

can be considered as a bounded perturbation. In general, this disturbance term must be estimated to compensate the target motion or it can be neglected, when the robot is moving with low velocities and saccelerations [48]. Our solution consists in designing a robust controller that explicitly deals with model inaccuracies and trajectory perturbations in the image space caused by high driving velocities and uneven terrains.

3.4 Camera Robot-System Model

The camera velocity $V_c = (v_c, \omega_c) \in \mathbb{R}^{6\times1}$ is related to the robot velocity $V_r \in \mathbb{R}^{6\times1}$ by using the adjoint matrix $A_{cr} \in \mathbb{R}^{6\times6}$ as expressed by:

$$V_c = A_{cr}\, V_r\,, \qquad A_{cr} = \begin{bmatrix} R_{cr} & [t_{cr}]_\times R_{cr} \\ 0_{3\times3} & R_{cr} \end{bmatrix}\,, \tag{3-9}$$

where $t_{cr} = (0, t_z, -t_x) \in \mathbb{R}^{3\times1}$ is the translation vector from the origin of the robot frame $\mathcal{F}_r$ to the origin of the camera frame $\mathcal{F}_c$ and $[t_{cr}]_\times$ is a skew-symmetric matrix. The rotation matrix $R_{cr}(\psi) \in \mathbb{SO}(3)$ of the robot frame $\mathcal{F}_r$ with respect to the camera frame $\mathcal{F}_c$ is computed as $R_{cr}(\psi) = R_x(\psi)\, R_y\,(\pi/2)\; R_x\,(-\pi/2)$. The robot inputs u can be related with the camera velocity V_c by using the homogeneous transformation matrix $T_{cr} \in \mathbb{R}^{6\times2}$, which is calculated by substituting Eq. (3-4) into Eq. (3-9) as:

$$T_{cr} = A_{cr}\, S\,, \tag{3-10}$$

where $T_{cr} = [\, T_v \;\; T_\omega \,] \in \mathbb{R}^{6\times2}$ can be split in two parts, with $T_v \in \mathbb{R}^{6\times1}$ and $T_\omega \in \mathbb{R}^{6\times1}$ being its first and second columns respectively. Substituting Eq. (3-10) into Eq. (3-8) results in the following IBVS system:

$$\dot{s} = L_s\, T_{cr}\, u + \xi_s\,. \tag{3-11}$$

3.5 Classic Image-Based Visual Servoing Approach

In the row crop following task, it is possible to identify two common practical situations based on the robot location regarding the crop rows: (i) inner, the robot is inside the crop rows; (ii) outer, the robot is outside the crop rows. In [22], authors have developed an IBVS approach using the analysis of the location of the image feature point $\mathcal{P}$ in the image plane. Then, the proposed IBVS strategy is divided into two parts or primitives: a row controller and a column controller. Therefore, in the block diagram represented in Fig. 3.3, the task planning stage will choose the primitive controller depending on the inspection of the image feature point $\mathcal{P}$ through the image feature

vector s. Afterwards, the selected controller will calculate the angular velocity ω_i, being $i = \{c, \mathcal{R}\}$, for IBVS Column and Row Controllers respectively. Lastly, the angular velocity ω_i is concatenated with the linear velocity v to form the input u of the robot. In the next subsections, we briefly present the IBVS control design and a sequence of images representing the control action for both row and column cases. Notice that, in Figures 3.4 and 3.5, the red line denotes the current crop row orientation whereas the blue line denotes the desired path tangent orientation. Both angles are taken to be positive if the lines rotation are counter-clockwise about y_v-axis of the image frame $\mathcal{F}_v$.

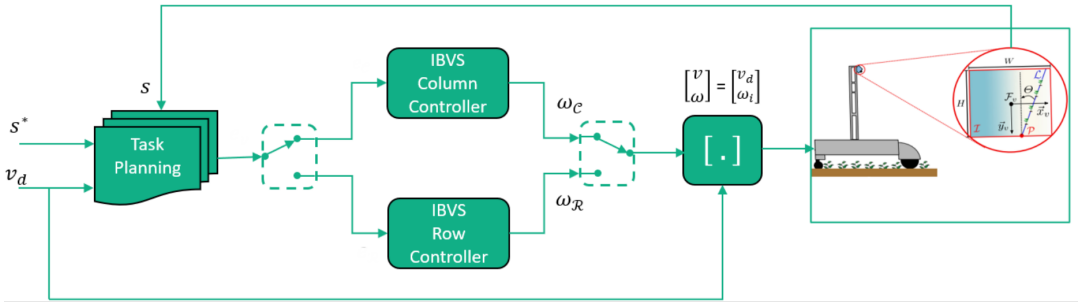

Figure 3.3: Block Diagram representing the two primitive controllers (Column and Row).

3.5.1 IBVS Column Controller (IBVS-CC)

The IBVS Column Controller (IBVS-CC) approach is applied for reaching the crop row or during maneuvers at the headlands, when the image feature point $\mathcal{P}$ is located at the right (or left) pixel column of the image plane $\mathcal{I}$. Here, we assume that the control goal is to drive the current image coordinates (y_v, θ_v) to the desired values $y_v^\star = H/2$ and $\theta_v^\star = 0$, under the path following constraint $x_v^\star = \pm W/2$ and $\dot{x}_v^\star = 0$. An application example of the IBVS-CC approach is depicted in Fig. 3.4(a)-(d), where it can be observed a sequence of motions in which the robot pose is successfully controlled. In such a case, the current $\mathcal{P}$ moves to the bottom left corner of the image plane $(y_v = H/2)$. Afterwards, the IBVS row controller should be applied so that $\mathcal{P}$ can achieve the desired $\mathcal{P}^\star$ located at the bottom center of the image plane.

Due to the velocity constraint $\dot{x}_v = 0$, Eq. (3-11) can be expressed as:

$$\dot{s}_c = \begin{bmatrix} \dot{y}_v \\ \dot{\theta}_v \end{bmatrix} = J_{v_c}\, v + J_{\omega_c}\, \omega + \xi_{s_c}\,, \tag{3-12}$$

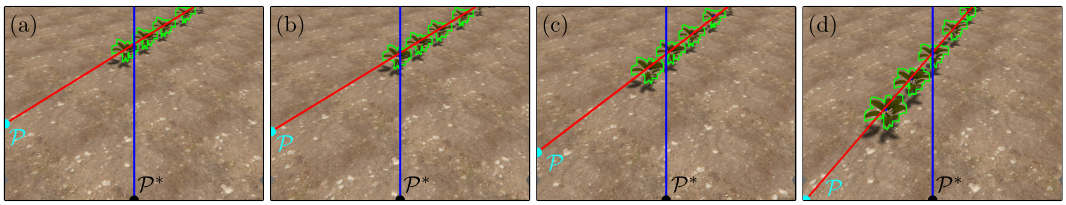

Figure 3.4: An application example of the IBVS-CC approach: The red and blue lines denote respectively the visually estimated crop row and the path tangent orientation with respect to the y_v-axis, in image frame $\mathcal{F}_v$.

where $\xi_{s_\mathcal{C}} \in \mathbb{R}^{2\times1}$ is a perturbation term, with $J_{v\mathcal{C}} \in \mathbb{R}^{2\times1}$ and $J_{\omega\mathcal{C}} \in \mathbb{R}^{2\times1}$ being respectively the linear and angular Jacobian matrices that result from the multiplication of the matrices corresponding to L_s and T_{cr}, as shown below:

$$J_{v\mathcal{C}} = \begin{bmatrix} L_y \\ L_\theta \end{bmatrix} T_v \,, \quad J_{\omega\mathcal{C}} = \begin{bmatrix} L_y \\ L_\theta \end{bmatrix} T_\omega \,. \tag{3-13}$$

Then, the IBVS system, described in Eq. (3-12) with $\xi_{s_\mathcal{C}} \approx 0$, can be controlled using the following control law:

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_d \\ -J^{\dagger}_{\omega\mathcal{C}}(\Lambda_c e_c + J_{v\mathcal{C}}\, v_d) \end{bmatrix} , \tag{3-14}$$

where $\Lambda_c \in \mathbb{R}^{2\times2}$ is a positive definite gain matrix and $v_d \in \mathbb{R}$ is the desired linear velocity. The image feature error $e_c = (e_y, e_\theta) \in \mathbb{R}^{2\times1}$ for the column controller is given by:

$$e_y = y_v - y_v^* \,, \qquad e_\theta = \theta_v - \theta_v^* \,. \tag{3-15}$$

The stability properties and convergence analysis for the IBVS-CC approach, given by Eq. (3-14), can be found in [22].

3.5.2 IBVS Row Controller (IBVS-RC)

The IBVS Row Controller (IBVS-RC) is applied for row following task at the vicinity of the crop row, when the image feature point $\mathcal{P}$ is located at the bottom (or top) pixel row of the image plane $\mathcal{I}$. Here, we assume that the control goal is to drive the current image coordinates (x_v, θ_v) to the desired values $x_v^\star = 0$ and $\theta_v^\star = 0$, under the path following constraint $y_v^\star = H/2$ and $\dot{y}_v^\star = 0$. An application example of the IBVS-RC approach is depicted in Fig. 3.5(a)-(d), where it can be observed a sequence of motions in which the robot pose is successfully controlled. In such a case, the current $\mathcal{P}$ moves to the bottom center of the image plane ($x_v = 0$) and Θ becomes vertical ($\theta_v = 0$). Then, the row crop (red line) aligns with the center of the image (blue line)

and the offset between the lines tends to zero.

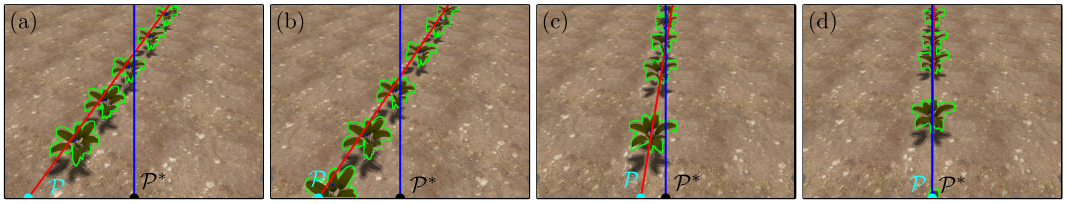

Figure 3.5: An application example of IBVS-RC approach: $\mathcal{P}$ (cyan) and $\mathcal{P}^*$ (black) points denotes respectively the current and desired image feature points, in image space $\mathcal{I}$.

Due to the kinematic constraint $\dot{y}_v = 0$, Eq. (3-11) can be expressed as:

$$\dot{s}_{\mathcal{R}} = \begin{bmatrix} \dot{x}_v \\ \dot{\theta}_v \end{bmatrix} = J_{v_{\mathcal{R}}}\, v + J_{\omega_{\mathcal{R}}}\, \omega + \xi_{s_{\mathcal{R}}}\,, \tag{3-16}$$

where $\xi_{s_{\mathcal{R}}} \in \mathbb{R}^{2\times 1}$ is a perturbation term, with $J_{v_{\mathcal{R}}} \in \mathbb{R}^{2\times 1}$ and $J_{\omega_{\mathcal{R}}} \in \mathbb{R}^{2\times 1}$ being respectively the linear and angular Jacobian matrices that result from the multiplication of the matrices corresponding to L_s and T_{cr}, as shown below:

$$J_{v\mathcal{R}} = \begin{bmatrix} L_x \\ L_\theta \end{bmatrix} T_v\,, \quad J_{\omega\mathcal{R}} = \begin{bmatrix} L_x \\ L_\theta \end{bmatrix} T_\omega\,. \tag{3-17}$$

Then, the IBVS system, described in Eq. (3-16) with $\xi_{s_{\mathcal{R}}} \approx 0$, can be controlled using the following control law:

$$\begin{bmatrix} v_{\mathcal{R}} \\ \omega_{\mathcal{R}} \end{bmatrix} = \begin{bmatrix} v_d \\ -J^{\dagger}_{\omega\mathcal{R}}(\Lambda_{\mathcal{R}}\, e_{\mathcal{R}} + J_{v\mathcal{R}}\, v_d) \end{bmatrix}\,, \tag{3-18}$$

where $\Lambda_{\mathcal{R}} \in \mathbb{R}^{2\times 2}$ is a positive definite gain matrix. The image feature error $e_{\mathcal{R}} = (e_x, e_\theta) \in \mathbb{R}^{2\times 1}$ for the IBVS row controller is given by:

$$e_x = x_v - x_v^*\,, \qquad e_\theta = \theta_v - \theta_v^*\,. \tag{3-19}$$

The stability properties and convergence analysis for the IBVS-RC approach, given by Eq. (3-18), can be found in [22].

3.6 Robust Image-Based Visual Servoing (rIBVS)

In this section, we shortly present the robust control design based on the SMC and STA approaches for the IBVS column and row controllers, introduced in the previous section. Here, let us introduce the following useful notation: $\lceil \cdot \rfloor^p = |\cdot|^p \operatorname{sgn}(\cdot)$ for $0 < p \leq 1$, where $\operatorname{sgn}(\cdot)$ is the sign function. Considering the model inaccuracies and trajectory perturbations in the designing of the robust

IBVS approach, the linear and angular velocities, v and ω, can be written respectively as:

$$v = v_d\,, \qquad \omega = \hat{J}^{\dagger}_{\omega i}\,\nu_i\,, \tag{3-20}$$

where $\hat{J}^{\dagger}_{\omega i} \in \mathbb{R}^{1\times 2}$ is the pseudo-inverse of the nominal angular Jacobian matrix and $\nu_i \in \mathbb{R}^{2\times 1}$ is the robust control action to be designed with $i = \{c,\mathcal{R}\}$. Now, we obtain the following perturbed IBVS system:

$$\dot{s}_i = \nu_i + \eta_i\,, \tag{3-21}$$

where the disturbance term η_i is given by:

$$\eta_i = (J_{\omega i}\hat{J}^{\dagger}_{\omega i} - I)\,\nu_i + J_{vi}\,v_d\, + \xi_{si}\,. \tag{3-22}$$

Notice that, the effect of the disturbance term η_i increases for high values of the desired driving velocity v_d and, consequently, the IBVS system may perform very poorly or become unstable. To cope with performance and instability issues caused by system's inaccuracies and perturbations, an alternative solution consists of using adaptive and robust control techniques due to their remarkable capability of dealing with parametric uncertainties and external disturbances [31]. In the block diagram shown in the Fig. 3.6, it is possible to see the addition of the robustness terms $\tau_i \in \mathbb{R}^{2\times 1}$, being $i = \{c, \mathcal{R}\}$, to the primitive controllers, aiming at dealing with disturbances caused by term η_i.

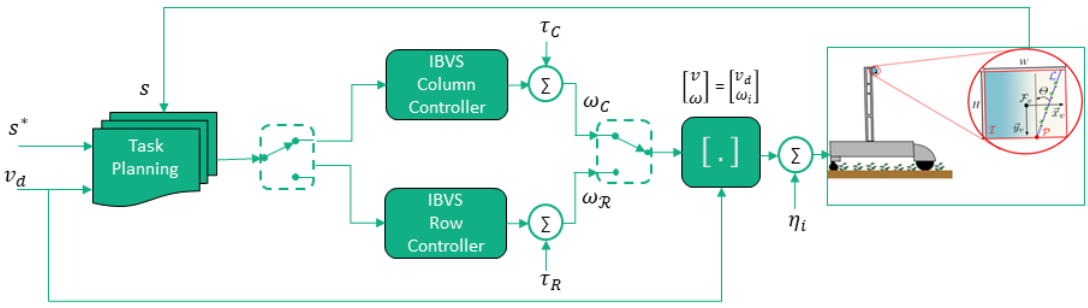

Figure 3.6: Block Diagram representing the robust control design for the IBVS column and row controller.

3.6.1
rIBVS Column Controller (rIBVS-CC)

In the design of the rIBVS-CC approach, we consider the following robustness term $\tau_c \in \mathbb{R}^{2\times 1}$ given by:

$$\tau_c = K_c\,\lceil \sigma_c \rfloor^{p}\, + \zeta_c\,, \qquad \dot{\zeta}_c = -\mu_c \mathrm{sgn}(\sigma_c)\,, \tag{3-23}$$

where $K_c \in \mathbb{R}^{2\times2}$ and $\mu_c \in \mathbb{R}^{2\times2}$ are positive definite gain matrices, and $\sigma_c \in \mathbb{R}^{2\times1}$ being the sliding surface defined as $\sigma_c = e_c + \Gamma_c \int_0^t e_c(\tau)\, d\tau$, where $\Gamma_c \in \mathbb{R}^{2\times2}$ is a positive definite matrix. Hence, the rIBVS-CC approach can be implemented by applying Eq. (3-23) to Eq. (3-20) and adding a proportional term $\Lambda_c e_c$, as shown below:

$$\nu_c = \Lambda_c\, e_c + \tau_c\, . \tag{3-24}$$

3.6.2 rIBVS Row Controller (rIBVS-RC)

In the design of the rIBVS-RC approach, we consider the following robustness term $\tau_{\mathcal{R}} \in \mathbb{R}^{2\times1}$ given by:

$$\tau_{\mathcal{R}} = K_{\mathcal{R}} \lceil \sigma_{\mathcal{R}} \rfloor^p + \zeta_{\mathcal{R}}\, , \qquad \dot{\zeta}_{\mathcal{R}} = -\mu_{\mathcal{R}} \mathrm{sgn}(\sigma_{\mathcal{R}})\, , \tag{3-25}$$

where $K_{\mathcal{R}} \in \mathbb{R}^{2\times2}$ and $\mu_{\mathcal{R}} \in \mathbb{R}^{2\times2}$ are positive definite gain matrices, and $\sigma_{\mathcal{R}} \in \mathbb{R}^{2\times1}$ the sliding surface defined as $\sigma_{\mathcal{R}} = e_{\mathcal{R}} + \Gamma_{\mathcal{R}} \int_0^t e_{\mathcal{R}}(\tau) d\tau$, where $\Gamma_{\mathcal{R}} \in \mathbb{R}^{2\times2}$ is a positive definite matrix. Hence, the rIBVS-RC approach can be implemented by applying Eq. (3-25) to Eq. (3-20) and adding a proportional term $\Lambda_{\mathcal{R}} e_{\mathcal{R}}$, as shown below:

$$\nu_{\mathcal{R}} = \Lambda_{\mathcal{R}}\, e_{\mathcal{R}} + \tau_{\mathcal{R}}\, . \tag{3-26}$$

The following theorem can be stated for establishing for the stability properties of the rIBVS approach:

Theorem 3.1 *Consider the IBVS system defined by Eq. (3-21) under the disturbance term given by Eq. (3-22), assumed to be bounded. Under the robust IBVS row and column controllers given by Eq. (3-26) and Eq. (3-24) respectively, the following properties hold: (i) all systems signals are bounded; (ii)* $\sigma_c(t), \sigma_{\mathcal{R}}(t) \to 0$*; (iii)* $e_x(t), e_\theta(t) \to 0$*; and (iv)* $e_y(t), e_\theta(t) \to 0$ *in finite time.*

Proof 1 *: For a proof of the rIBVS approach based on SMC, please, see Appendix A. Furthermore, for a proof of the rIBVS approach based on STA, please, consult [46]*

3.7 3D Computer Simulation

In this section, we present 3D computer simulations and case studies with the rIBVS approach, based on SMC, developed for autonomous visual navigation with a wheeled mobile robot in row crop fields. An example of a

3D agricultural environment with non-ideal plants and terrain conditions was created in Gazebo platform, as shown in Fig. 3.7.

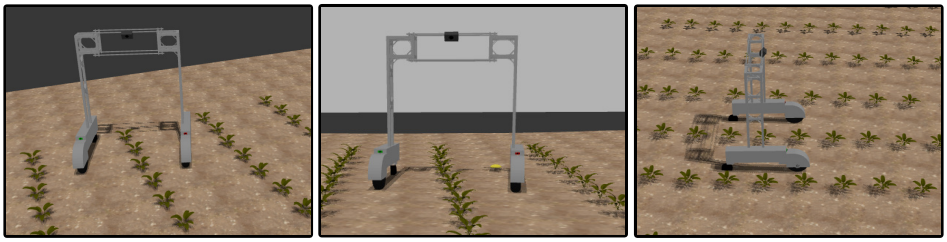

Figure 3.7: Soybot robot navigating in the 3D agricultural environment created in the Gazebo platform, with sparse plants and rough terrains.

In real-world scenarios, row crop fields occur in rough and uneven terrains, and plants are sparsely distributed. The WMR used for autonomous navigation tests was the Soybot, a differential-drive mobile robot, ad-hoc designed for plant health monitoring in soybeans and cotton crops [10], equipped with a monocular camera attached at the top of the mechanical structure. The robot and the world (i.e., the crop field) models are described in a SDF file, containing all their physical and geometrical characteristics. The visual and collision geometry of the robot are given in STL format developed in *SolidWorks* CAD. Moreover, the camera and motor plugins are also given in the SDF file, that sends their data to the user via ROS topic [49]. In this simulation, we use the diff drive plugin, that takes the x component of the linear velocity and z component of the angular velocity to compute the velocities sent to the two wheels of a differential drive wheel base. The simulated field is created using a tool called *heightmap*, which converts a 2D grayscale image (Fig.3.8(a)) into a 3D RGB format (Fig.3.8(b)). This 3D model can be easily imported into the Gazebo platform [50]. To emulate the crop plants, we use a simple 3D model in STL format, called *acanthus*.

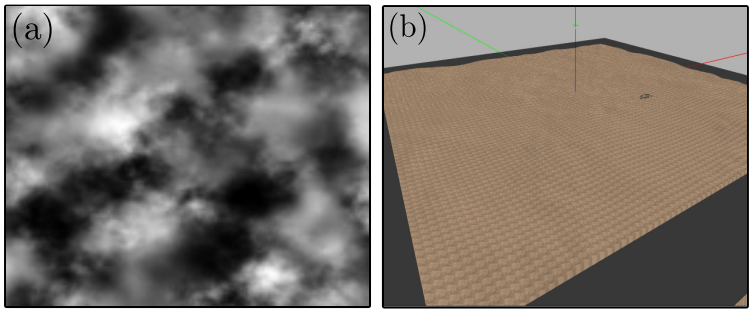

Figure 3.8: (a) 2D grayscale representing the heightmap used to create (b) the 3D environment used to simulate this comparative study.

3.7.1 Robot-Camera Parameters

Here, we present the parameters of the camera-robot system used in the simulated experiments. The dimensions of the robot, that is, its width, length and height, are given respectively by $(W_r, L_r, H_r) = (1.2, 1.9, 2.0)\ m$. The diameter of the front wheels is $d_f = 40\ cm$, and the castor wheels located at the robot back have $d_b = 20\ cm$ of diameter. The camera used in the simulation was similar to a Logitech C270 HD webcam (720p, 30fps), with a focal length of $f = 8\ mm$, and scale factors of $\alpha_x = 69.4\ pixel\, mm^{-1}$ and $\alpha_y = 52.1\ pixel\, mm^{-1}$ respectively. The RGB images captured by the camera have a dimension of 640×480 pixels resolution. These images are used for the navigation task after the line feature extraction.

The camera is placed in a vertical and horizontal coordinates, relative to the robot's centre of mass, given by $(t_x, t_y, t_z) = (0.16, 0.0, 1.4)\ m$, with a tilt angle offset of $\psi = \pi/4\ rad$. The communication between the sensors and the plugins was implemented using the Robot Operating System (ROS), release Kinetic Kame, running on Linux OS Ubuntu 16.04 LTS using an Intel Core i5-8250U 1.6GHz, 8GB DDR4 RAM. The ROS nodes for implementing the autonomous navigation task were developed in Python with OpenCV libraries. In the simulation tests, we considered an uncertainty of 20% in all the intrinsic camera calibration parameters and in the following extrinsic parameters: ψ, t_x, t_z. To verify the robustness properties of the proposed controller, we also measured noise in the image feature and unmodeled dynamics with high- or low-frequency, to emulate the effects of actuator dynamics and Coulomb friction. The unmodeled dynamics is given by the transfer function $D(s) = k/\lambda s + 1$, where $k = 1$ and $\lambda \in [0.1, 0.5]$.

3.7.2 Results of Comparative Study I

Here, we will present a comparative study[1] among the classic IBVS controller, given by Eq. (3-14) and Eq. (3-18), the rIBVS controller, given by Eq. (3-24) and Eq. (3-26) and the IBVS approach based on the Unit Vector Control (UVC) described in [44] with a small term $\Pi = 10^{-3}$ summed to avoid discontinuities in the control law. We also consider a high driving velocity, as $v_d = 1.5\ ms^{-1}$, to evaluate the performance of the IBVS controllers in situations where the robot needs to move faster along the row crop field, relaxing the low-speed assumption used in [22]. Once the robot starts outside of the crop row,

[1] The video clip representing this comparative study can be found at the following link: https://youtu.be/nXVUzyDoKoQ.

the image feature error e_v will have high values. Thereby, we need to apply the IBVS Column Controller approach with small values in the gain matrix Λ_c, for the robot to reach the crop row smoothly. If the gains are too high, the robot will perform the maneuver abruptly, which may cause instability in the system, since the robot's input values may be too large. On the other hand, when the robot needs to stay above the row, the image feature error e_v will have lower values. Furthermore, the performance of the control approach in this case, needs to be more precise, so that the robot does not reach neighboring rows. Therefore, the gain matrix $\Lambda_\mathcal{R}$ for the IBVS Row Controller approach will have bigger values compared to IBVS Column Controller approach. For the IBVS-CC approach, the gains were chosen in the range between diag(0.1,0.2) and diag(0.5,0.6), while, for the IBVS-RC approach, the values were selected in the range between diag(0.5,1.0) and diag(5,10). In addition, the robust controllers gains K_c and $K_\mathcal{R}$ were varied from 0.01I to 0.5I for both approaches, aiming to identify their best values. Table 3.1 shows the combination of gains that presented best results in each case. The gains μ_c, $\mu_\mathcal{R}$, Γ_c and $\Gamma_\mathcal{R}$ have no values in the Table 3.1, as all controllers in this comparative study are of first order and the sliding surfaces σ_i, being $i = \{c, \mathcal{R}\}$, are equal to the image feature vector e_v.

Table 3.1: Control parameters for the IBVS approaches.

Parameters	classic IBVS	UVC	rIBVS (SMC)
Λ_c	diag(0.3, 0.6)	diag(0.15, 0.3)	diag(0.3, 0.6)
$\Lambda_\mathcal{R}$	diag(2.5, 5.0)	diag(2.0, 5.0)	diag(2.5, 5.0)
$K_c, K_\mathcal{R}$	-	$0.2I$, $0.02I$	$0.2I$, $0.02I$
$\mu_c, \mu_\mathcal{R}$	-	-	-
$\Gamma_c, \Gamma_\mathcal{R}$	-	-	-

Other parameter is the simulation time $T_{max} = 60$ s. At the beginning of the experiments, the robot is located outside the row crop. The column and row controllers are then activated to enable the robot to reach the crop row and align to it. Figures 3.9(a)-(b) and 3.9(c)-(d) depict the behavior over time of the image errors and image coordinates in x- and y-axis of the image frame $\mathcal{F}_v$ respectively, whereas Fig. 3.9(e)-(f) depicts the image error e_θ and the image coordinate θ_v. We can observe that all controllers are able to drive the image coordinate y_v to the desired value $H/2$, and the robot successfully reaches the crop row. However, it is possible to notice the degraded performance obtained with the IBVS approach for image coordinates x_v and θ_v. Therefore, the robot motion is impaired by unwanted oscillations during the crop-row following,

caused by external disturbances in the field and parametric uncertainties in the camera-robot system, which can damage neighboring plants and even the robot. On the other hand, as can be seen in Fig. 3.10(a)-(d), the rIBVS and UVC approaches are capable of mitigating such perturbations and inaccuracies, allowing the robot to carry out the autonomous navigation task in a safe and efficient manner.

Finally, Fig. 3.11(a)-(b) shows the robot inputs (v, ω) and Fig. 3.11(c)-(d) depicts the robot wheel speeds (ω_R, ω_L), where it can be observed the remarkable performance of the rIBVS and UVC approaches in contrast to the oscillatory behavior of the IBVS approach.

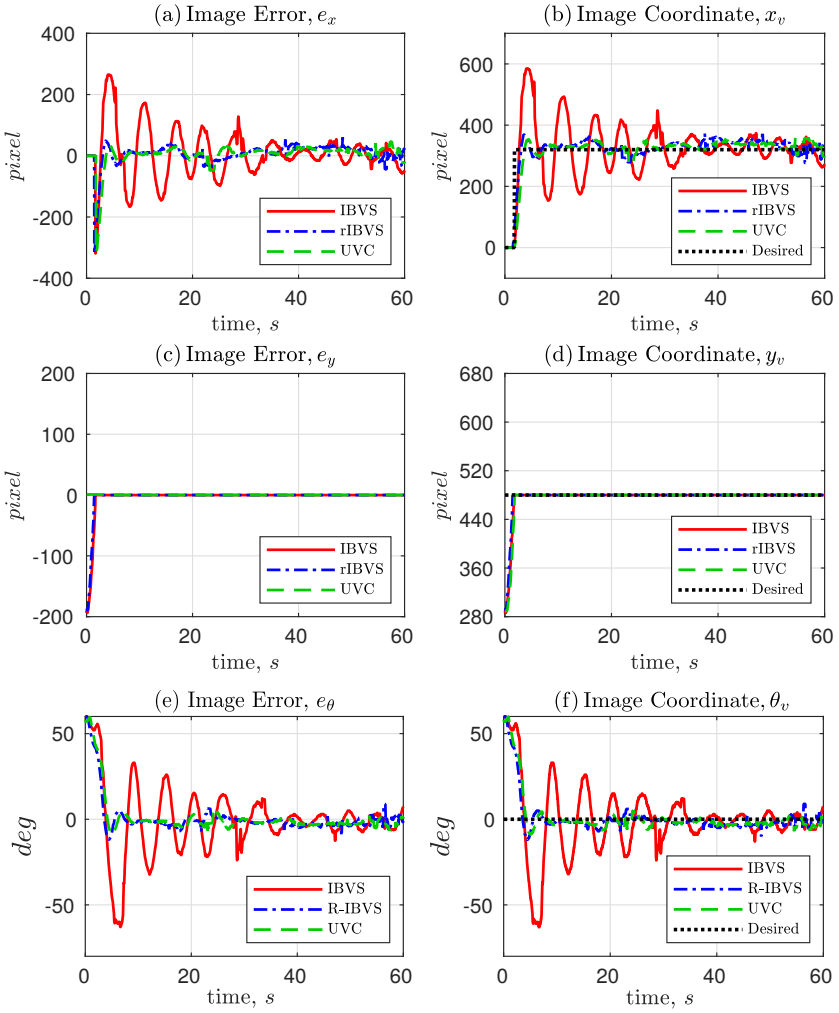

Figure 3.9: Comparative study among the IBVS, rIBVS and UVC approaches: (a)-(b) image error, e_x and image coordinate, x_v; (c)-(d) image error, e_y and image coordinate, y_v. (e)-(f) image error, e_θ and image coordinate, θ_v

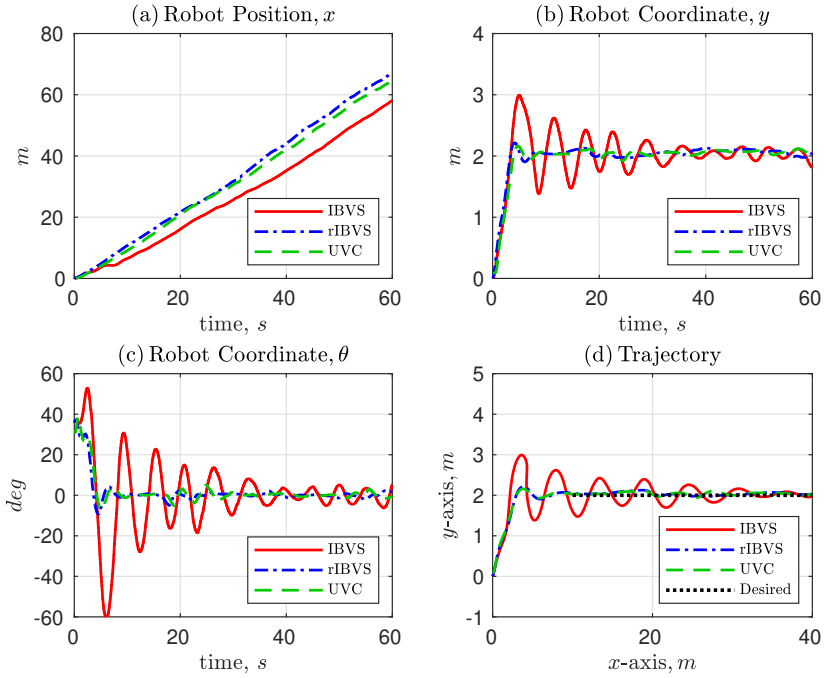

Figure 3.10: Comparative study among the IBVS, rIBVS and UVC approaches: (a)-(c) robot coordinates, x_r, y_r and θ_r; (d) trajectory of the robot navigating along the crop row.

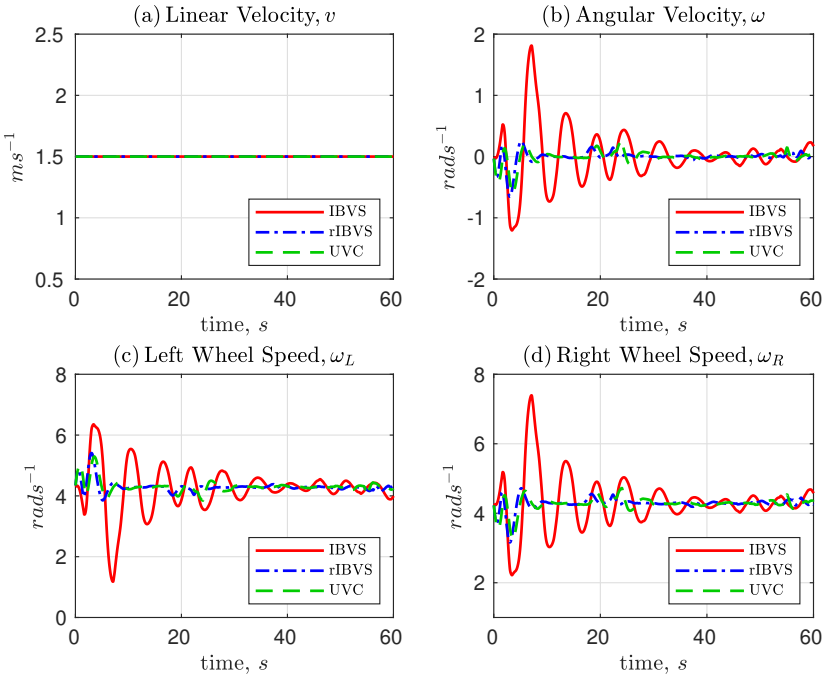

Figure 3.11: Comparative study among the IBVS, rIBVS and UVC approaches: (a)-(b) linear and angular velocities; (c)-(d) left and right wheel speeds.

For comparative study purposes, we use two performance metrics as shown in Table 3.2: (i) the root mean square error (RMSE) of the image feature errors e_x, e_y and e_θ; (ii) the mean-absolute deviation (MAD) of the control signals ω, ω_L and ω_R. Analyzing the RMSE, we notice that the rIBVS approach shows the smallest values for the image features coordinates compared to IBVS and UVC approaches. Hence, the rIBVS approach presents a better transient behavior with fewer oscillations since the RMSE metrics penalizes more the most significant errors. In addition, Table 3.2 indicates that the rIBVS approach has lower average values for the control signals, that may be relevant for saving battery. Notice that, the rIBVS and UVC approaches are able to generate almost the same control effort for the robot wheels than the IBVS approach, even in the presence of disturbances and uncertainties.

Table 3.2: Performance metrics: RMSE for image feature errors and MAD for control signals.

RMSE	IBVS	rIBVS (SMC)	UVC	unit
x_v	27.76	12.65	12.89	cm
y_v	4.91	4.60	6.18	cm
θ_v	18.40	11.35	12.27	deg
MAD	IBVS	rIBVS (SMC)	UVC	unit
ω	0.261	0.0430	0.0533	$rad\,s^{-1}$
ω_L	0.448	0.0737	0.0914	$rad\,s^{-1}$
ω_R	0.448	0.0737	0.0914	$rad\,s^{-1}$

3.7.3 Results of Comparative Study II

In this simulation, we decrease the spacing between the rows, and increase terrain irregularities. Therefore, we will present a comparative study [2] among the classic IBVS controller described in [22, 24], the IBVS controller based on the unit vector control (UVC) approach described in [44] with a small term $\Pi = 10^{-3}$ summed to avoid discontinuities in the control law, and the proposed rIBVS controller based on the STA approach, given by Eq. (3-23) and Eq. (3-25) with sliding surfaces σ_c and $\sigma_{\mathcal{R}}$. The choice of gain matrices values Λ_c, $\Lambda_{\mathcal{R}}$, K_c and $K_{\mathcal{R}}$ follows the same reasoning described in Subsection 3.7.2. On the other hand, for this case, we consider the gains μ_c, $\mu_{\mathcal{R}}$, Γ_c and $\Gamma_{\mathcal{R}}$, that were not used in the first comparative study (Subsection 3.7.2). These gains were chosen small, since they are multiplied by integral terms, which make

[2] The video clip representing this comparative study can be found at the following link: `https://youtu.be/Sswm8KLGKlk`.

the system unstable, if chosen with high values. These tested values for the gains μ_c and $\mu_{\mathcal{R}}$ were between $0.005I$ and $0.02I$. In addition, the gains Γ_c and $\Gamma_{\mathcal{R}}$ were chosen in the range between $0.0001I$ and $0.001I$. The control gains that presented best results are shown in Table 3.3. The simulation time is $T_{max} = 30\ s$. Morever, like the previous case study, we also consider high driving velocities for the robot, such as $v_d = 1.5\ ms^{-1}$, to evaluate the performance of all IBVS controllers in cases where the robot needs to move faster across the row crop field, relaxing the low-speed assumption used in [22, 24].

Table 3.3: Control parameters for the IBVS approaches.

Parameters	classic IBVS	UVC	rIBVS (STA)
Λ_c	$\text{diag}(0.3, 0.6)$	$\text{diag}(0.15, 0.3)$	$\text{diag}(0.15, 0.3)$
$\Lambda_{\mathcal{R}}$	$\text{diag}(2.5, 5.0)$	$\text{diag}(1.0, 1.5)$	$\text{diag}(1.0, 1.5)$
$K_c, K_{\mathcal{R}}$	-	$0.3I$, $0.05I$	$0.3I$, $0.08I$
$\mu_c, \mu_{\mathcal{R}}$	-	-	$0.01I$, $0.005I$
$\Gamma_c, \Gamma_{\mathcal{R}}$	-	-	$0.001I$, $0.001I$

At the start of the experiments, the robot is located outside the row crop field. Then, the column and row controllers are activated to allow the robot to reach the crop row and align to it. The time history of the image errors and image coordinates in x- and y-axis of the image frame $\mathcal{F}_v$ are depicted in Figures 3.12(a)-(b) and 3.12(c)-(d) respectively. Furthermore, Fig. 3.12(e)-(f) reciprocally depicts the behavior over time of the image error e_θ and the image coordinate θ_v.

We can observe the oscillating behavior and poor performance obtained with the classic IBVS approach for image coordinates x_v and θ_v. Thereby, the robot motion may be affected by these unwanted oscillations during the crop row following tasks, caused by external disturbances in the field and parametric uncertainties in the robot-camera system, damaging neighboring plants and even the vehicle. On the other hand, we can see that in the steady-state regime all IBVS controllers can drive the image coordinate y_v to the desired value $H/2$, and the robot successfully reaches the crop row, despite the poor performance of the classic IBVS approach.

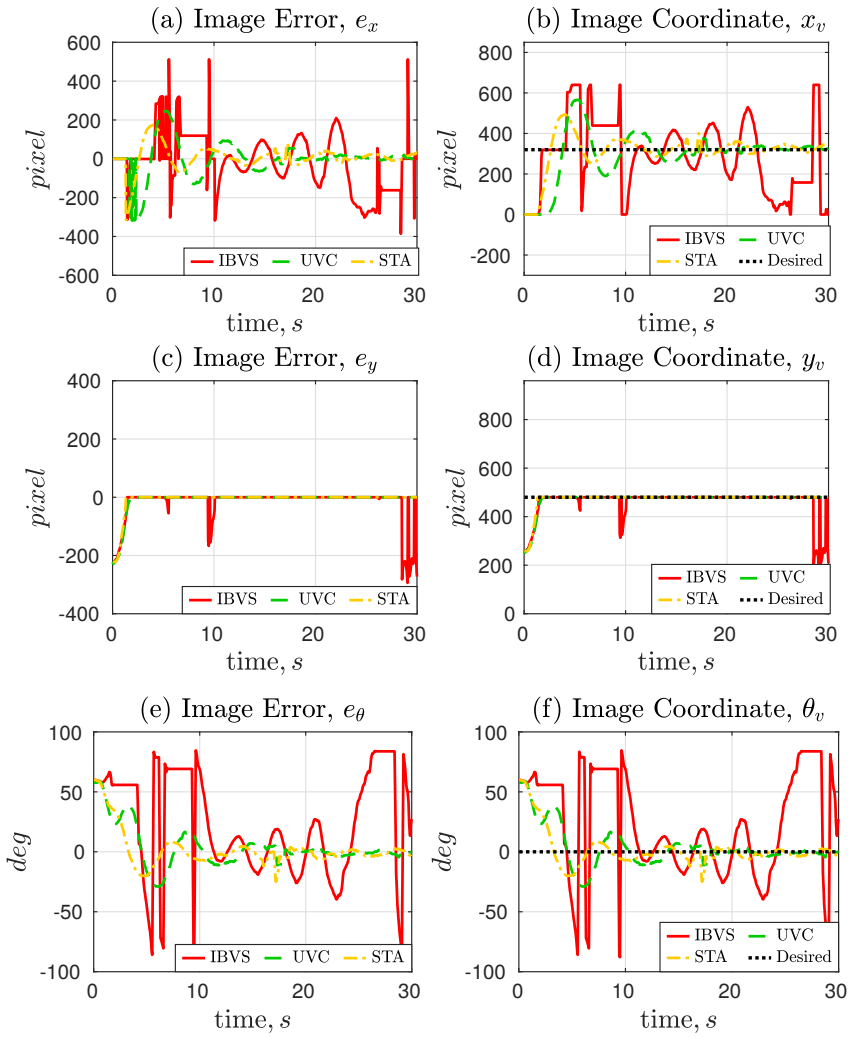

Figure 3.12: Comparative study among the IBVS, UVC and STA approaches: (a)-(b) image error, e_x and image coordinate, x_v; (c)-(d) image error, e_y and image coordinate, y_v; (e)-(f) image error, e_θ and image coordinate, θ_v.

In Fig. 3.13(a)-(d), we can notice that the UVC and STA approaches are the only ones capable of dealing with such modeling inaccuracies and trajectory perturbations, allowing the robot to carry out the autonomous navigation task safely and efficiently.

Finally, Fig. 3.14(a)-(b) shows the time history of the robot inputs (v, ω), whereas Fig. 3.14(c)-(d) depicts the time history of the robot wheel speeds (ω_L, ω_R) respectively. Thereby, we can observe the remarkable performance of the UVC and STA approaches in contrast to the oscillatory behavior of the classic IBVS approach.

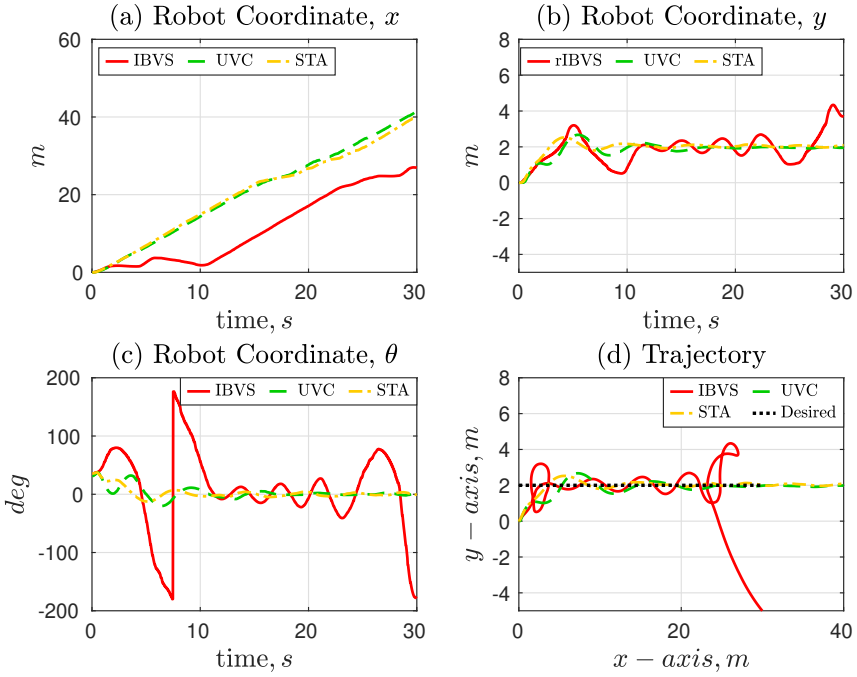

Figure 3.13: Comparative study among the IBVS, UVC and STA approaches: (a)-(c) robot coordinates, x_r, y_r and θ_r; (d) trajectory of the robot navigating along the crop row.

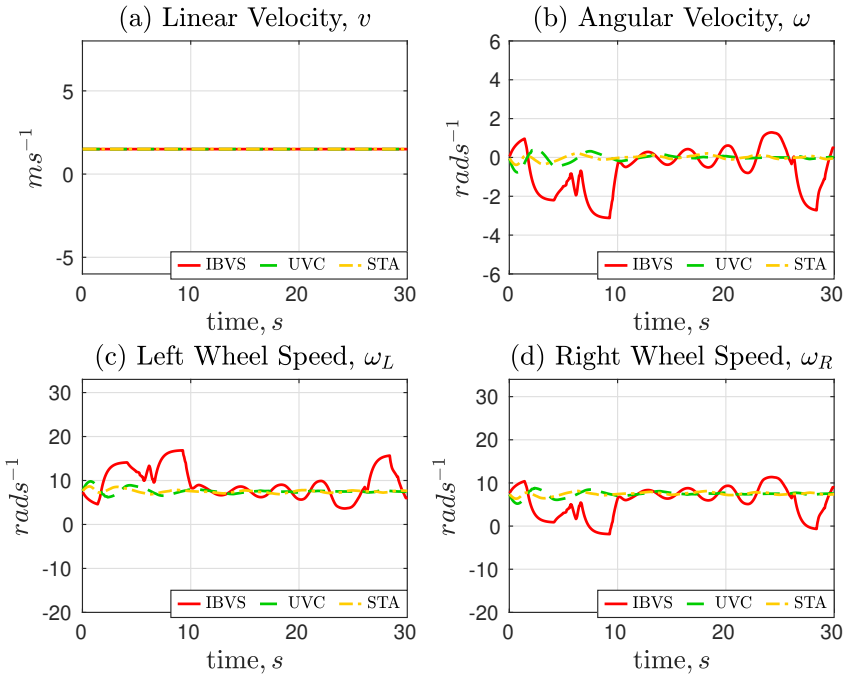

Figure 3.14: Comparative study among the IBVS, UVC and STA approaches: (a)-(b) linear and angular velocities; (c)-(d) left and right wheel speeds.

For comparative study purposes, we use two performance metrics as shown in Table 3.4: (i) the root-mean-square error (RMSE) of the image features coordinates x_v, y_v and θ_v; (ii) the mean-absolute deviation (MAD) of the robot input ω, and the wheel speeds ω_L and ω_R. Evaluating the RMSE, we observe that the STA approach shows the lowest values for all image features coordinates compared to the classic IBVS and UVC approaches. Therefore, the STA approach presents a better transient behavior with less oscillations, since the RMSE metrics penalizes the most significant errors. Moreover, Table 3.4 indicates that the rIBVS approach has lower mean values for the control signals, which can be relevant to save battery during operations. Finally, we note that the UVC and STA approaches generate almost the same control effort for the robot wheels, which is much less than the classic IBVS approach.

Table 3.4: Performance metrics: RMSE and MAD.

RMSE	classic IBVS	UVC	rIBVS (STA)	Unit
x_v	37.09	24.01	16.40	cm
y_v	16.49	6.48	5.98	cm
θ_v	32.61	13.47	11.00	deg
MAD	classic IBVS	UVC	rIBVS (STA)	Unit
ω	0.6	0.08	0.06	$rad\,s^{-1}$
ω_L, ω_R	1.711	0.244	0.197	$rad\,s^{-1}$

3.7.4 Results of Comparative Study III

In this section, we present a comparative study among the rIBVS controllers based on SMC and STA approaches. Here, we consider similar conditions described in subsection 3.7.3, such as, the same terrain and linear velocity $v_d = 1.5ms^{-1}$. Thereby, the WMR moves faster across the row crop field, relaxing the low speed assumption [22, 24]. In addition, the control gains of the applied techniques were chosen following the same logic mentioned in subsections 3.7.2 and 3.7.3. The gains presented in Table 3.5 lead to the best results in simulations carried out for this section. The simulation time is $T_{max} = 30s$.

Table 3.5: Control parameters for the rIBVS approaches.

Parameters	rIBVS (SMC)	rIBVS (STA)
$\Lambda_{\mathcal{C}}$	$\text{diag}(0.3, 0.3)$	$\text{diag}(0.15, 0.3)$
$\Lambda_{\mathcal{R}}$	$\text{diag}(1.0, 1.5)$	$\text{diag}(1.0, 1.5)$
$K_{\mathcal{C}}, K_{\mathcal{R}}$	$0.4I$, $0.08I$	$0.3I$, $0.05I$
$\mu_{\mathcal{C}}, \mu_{\mathcal{R}}$	-	$0.01I$, $0.005I$
$\Gamma_{\mathcal{C}}, \Gamma_{\mathcal{R}}$	-	$0.001I$, $0.001I$

As in previous experiments described in subsections 3.7.2 and 3.7.3, the robot is placed outside the row crop field at the beginning of the tests. Then, the column and row primitives are applied, enabling the WMR to achieve the crop row and align to it. Figures 3.15(a)-(b) and 3.15(c)-(d) show the performance over time of the image errors and image coordinates in x- and y-axis of the image frame $\mathcal{F}_v$ respectively. Next, Fig. 3.15(e)-(f) presents the image error e_θ and the image coordinate θ_v. Analyzing these figures, we notice that both approaches are capable of taking the image coordinates x_v, y_v and θ_v to the desired values. However, the results shown in Fig. 3.15(a)-(b) and Fig. 3.15(e)-(f), indicate that the rIBVS controller based on STA approach has a smoother response, than the rIBVS controller based on SMC approach. The results shown in Fig 3.16(a)-(d), confirm that both controllers are able to mitigate the parametric uncertainties in the camera-robot system and the external disturbances caused by the uneven terrain. Thereby, the two approaches can drive the robot to the crop row, and keep it aligned. Moreover, we can also verify that STA approach has a more attenuated steady-state oscillatory behavior compared to the rIBVS controller based on the SMC approach. Therefore, STA approach can better deal with the disturbances acting on the robot.

Finally, Fig. 3.17(a)-(b) presents the time history of the robot inputs (v, ω) and Fig. 3.17(c)-(d) depicts the robot wheel speeds (ω_L, ω_R) respectively. The inputs of both controllers seem very similar if we take into account the graphs shown in Fig. 3.17(a)-(d). Then, a more quantitative performance analysis will be made using the metrics described in Table 3.6, aiming at evidencing the differences between the two analyzed approaches.

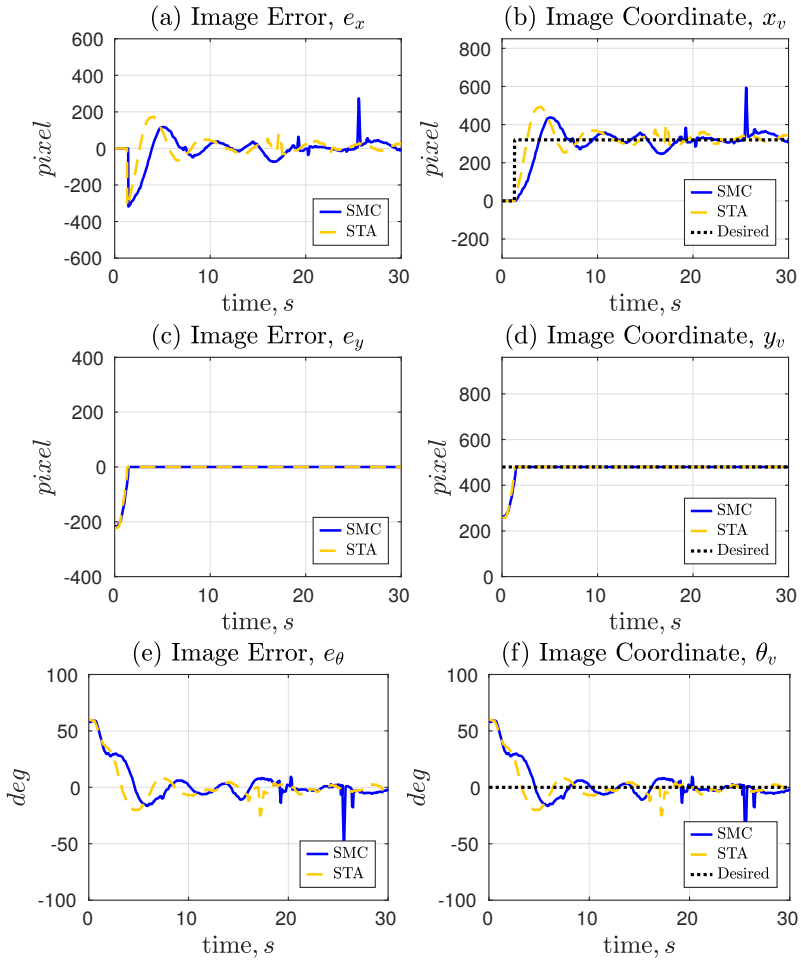

Figure 3.15: Comparative study between the rIBVS controllers based on SMC and STA approaches: (a)-(b) image error, e_x and image coordinate, x_v; (c)-(d) image error, e_y and image coordinate, y_v; (e)-(f) image error, e_θ and image coordinate, θ_v.

Now, for comparative study purposes, we calculate two performance metrics as depicted in Table 3.6: (i) the root-mean-square error (RMSE) of the image features x_v,y_v and θ_v; and (ii) the mean absolute deviation (MAD) of the robot input ω and the wheel speeds ω_L and ω_R. Looking at the RMSE, we see that the rIBVS controller based on STA approach has lower values than the rIBVS controller based on SMC approach for all image features coordinates. Thereby, the first method has a smaller oscillatory behavior, and does not have so many accentuated values of perturbations, since the RMSE metrics penalizes the most significant errors. In addition, we can note that STA approach presents lower MAD values for the control signals than the SMC approach, meaning that the first technique is more efficient for battery operations.

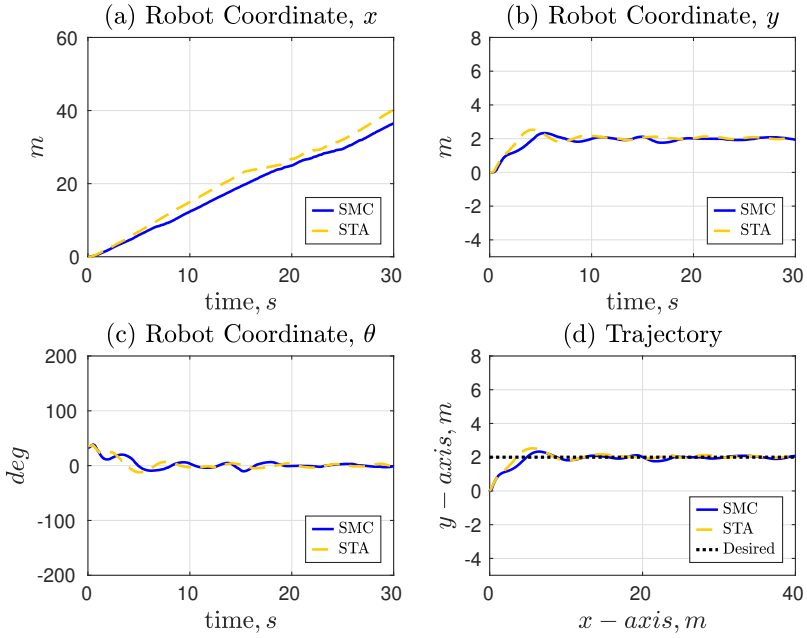

Figure 3.16: Comparative study between the rIBVS controllers based on SMC and STA approaches: (a)-(c) robot coordinates, x_r, y_r and θ_r; (d) trajectory of the robot navigating along the crop row.

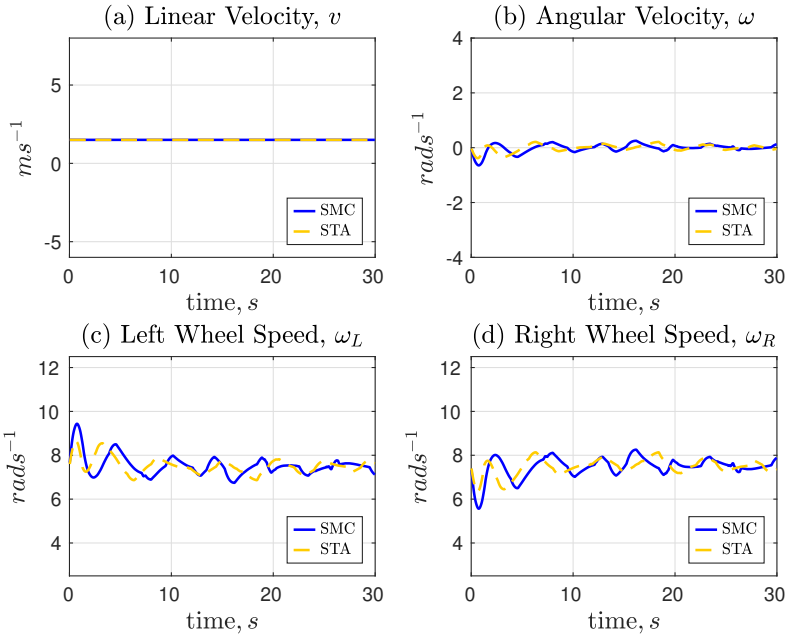

Figure 3.17: Comparative study between the rIBVS controllers based on SMC and STA approaches: (a)-(b) linear and angular velocities; (c)-(d) left and right wheel speeds.

Table 3.6: Performance metrics: RMSE and MAD.

RMSE	rIBVS (SMC)	rIBVS (STA)	Unit
x_v	26.29	16.40	cm
y_v	8.91	5.98	cm
θ_v	20.92	11.00	deg
MAD	rIBVS (SMC)	rIBVS (STA)	Unit
ω	0.103	0.06	$rad\,s^{-1}$
ω_L, ω_R	0.309	0.197	$rad\,s^{-1}$

3.8 Final Remarks

In this chapter, we present a robust image-based visual servoing (rIBVS) approach for wheeled mobile robots (WMRs) to execute autonomous visual navigation tasks in row crop fields using a single monocular camera. The main idea is to include a robustness term to the classic IBVS approach to deal with model inaccuracies caused by camera miscalibration and trajectory perturbations, due to the robot's high driving velocities. In the first comparative study, we considered three control strategies, the classic IBVS controller with row and column primitives, and two robust controllers based on a first-order sliding mode control approach. Then, in the second comparative study, we applied the rIBVS approach, based on a second-order sliding mode control approach, to deal with the added difficulties in relation to the last test. Finally, in the third comparative study, we considered the rIBVS controller based on the SMC approach developed in the first comparative study and the rIBVS controller based on the STA approach developed in the second comparative study to analyze which one had the best performance. The performance of the controllers has been evaluated using the RMSE and MAD metrics applied to the image errors and control signals The 3D Computer simulations with a DDMR were carried out in a row crop field buit on the ROS-Gazebo platform The virtual agricultural environment has been created to include irregularities in the terrain and sparse plants, emulating robot navigation for monitoring row crop fields. In the next chapter, we will design a robust controller capable of dealing with wheel slippage and sideslip owing to autonomous navigation on rough and uneven terrains.

4
Cascade Robust Controller for Row Crop Following with Wheeled Mobile Robots in Sloped and Rough Terrains

Here, in this chapter, we will introduce a cascade-based robust control approach for wheeled mobile robots (WMRs), equipped with a fixed monocular camera, to perform autonomous navigation tasks in row crops accurately. Here, we consider the existence of uncertainties in the parameters of the robot-camera system, and external disturbances caused by high driving velocities, sparse plants and uneven terrains. First, we design a robust image-based visual servoing (rIBVS) approach based on super-twisting algorithm (STA) to deal with model inaccuracies and trajectory perturbations in the image space. Then, a robust trajectory tracking (rTTC) approach based on STA is applied for motion stabilization to ensure the successful execution of row crop following tasks under wheel slippage and vehicle sideslip. The effectiveness and feasibility of the proposed robust-cascade control (rCC) methodology are evaluated by analyzing performance metrics. obtained from 3D computer simulations executed in ROS-Gazebo platform, an open-source robotics simulator, using a differential-drive mobile robot (DDMR) and an ad-hoc designed row-crop environment.

4.1
Problem Formulation

Here, we consider the problem of row crop following by using a fixed monocular camera attached to the top of the SoyBot [10], a differential-drive mobile robot. The task of interest is to guide the robot over a single crop row in a rough and sloped terrain, with the wheels in the midpoint between two adjacent rows. Furthermore, we also consider that the camera is not calibrated, set up with a tilt offset angle $\psi \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ with respect to the x-axis of the camera frame $\mathcal{F}_c$. Just like in the previous chapter, the camera's optical center is positioned at coordinates $(t_x, 0, t_z) \in \mathbb{R}^3$ with respect to the robot frame $\mathcal{F}_r$, as shown in Fig. 4.1.

The designed control approach for performing this activity is based on a robust-cascade control (rCC) method, with two control loops: an external, comprising the rIBVS approach, and an internal, which is composed of an

rTTC approach. The design details of the rIBVS control approach were detailed in Chapter 3. On this occasion, the rIBVS approach is responsible for generating a virtual reference DDMR, with the same kinematic constraints as the real DDMR. Thereby, the second control loop, which consists of the rTTC approach, needs to take the robot generalized coordinates $q = (x, y, z)$ to the desired trajectory with coordinates $q_d = (x_d, y_d, \theta_d) \in \mathbb{R}^{3\times1}$. During this task, the robot coordinates q are measured by odometry. Before designing the rTTC control law, we need to perform a coordinate transformation of the kinematic model into the chained form [25], which is more convenient to solve trajectory tracking problems.

To make the problem even more real, beyond assumptions considered in Chapter 3, we have two more for the rCC approach: (A5) the terrain is sloped, and (A6) the robot is affected by external disturbances due to wheel slippage and sideslip caused by rough and sloped terrain. Then, the rIBVS is applied to deal with the disturbances caused by assumptions mentioned in the previous chapter. In addition, the rTTC is designed based on the STA approach for trajectory tracking to cope with perturbations caused by assumptions (A5) and (A6).

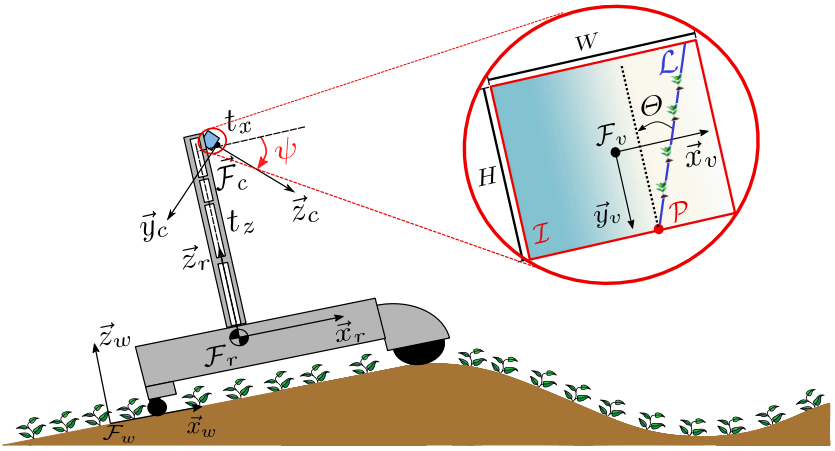

Figure 4.1: The representation of SoyBot robot travelling along a sloped terrain, and its coordinate frames: $\mathcal{F}_v$ is the image frame and $s = (x_v, y_v, \theta_v) \in \mathbb{R}^3$ is the image features vector. Side view of the robot: $\mathcal{F}_r$ is the robot frame, $\mathcal{F}_w$ is the world frame, $\mathcal{F}_c$ is the camera frame with a camera tilt offset ψ.

4.2 External Disturbances

Here, we also consider that a WMR has to navigate autonomously in agricultural fields in the presence of external disturbances caused by the slope and roughness of the terrain. In such a case, WMRs may be affected by wheel slippage and sideslip while traversing or steering on rough terrains. In general,

wheel slippage occurs either when the vehicle moves on a loose soil or a sloped terrain. Let v_x and v_y be the longitudinal and lateral travelling speeds of the robot wheels. The wheel slippage in the longitudinal direction, denoted by the slip ratio $\gamma \in \mathbb{R}$, is given by:

$$\gamma = \begin{cases} (\nu - v_x)/\nu\,, & |\nu| \geq |v_x|\,, \\ (\nu - v_x)/v_x\,, & |\nu| < |v_x|\,, \end{cases} \tag{4-1}$$

with $\nu \in \mathbb{R}$ being the linear speed of the wheel, where $r > 0$ is the wheel radius and $\dot{\phi}$ is the angular speed of a given wheel (i.e., left or right wheel) such that $\nu = r\dot{\phi}$. Notice that, the slip ratio γ will take a value in the range of $(-1, 1]$. In such cases, when $|\nu| \geq |v_x|$ the vehicle is moving uphill (driving). Otherwise, when $|\nu| < |v_x|$, the vehicle is moving downhill (braking). Here, we also assume that: (A7) The wheel speeds do not change abruptly during the autonomous maneuver and, therefore, the slip ratio of the right and left wheels, $\gamma_{_R}$ and $\gamma_{_L}$, are approximately the same ($\gamma_{_R} \approx \gamma_{_L} \approx \gamma$). Then, we can rewrite the slip ratio γ as a single slipping coefficient κ given by:

$$\kappa = \begin{cases} (1 - \gamma)\,, & |\nu| \geq |v_x|\,, \\ (1 + \gamma)^{-1}\,, & |\nu| < |v_x|\,, \end{cases} \tag{4-2}$$

Consequently, the driving and steering velocities v and ω can be related to the angular speeds ω_R and ω_L of the right and left wheels, considering the slipping coefficient κ, as given by:

$$v = \frac{\kappa r(\omega_R + \omega_L)}{2}\,, \quad \omega = \frac{\kappa r(\omega_R - \omega_L)}{d}\,. \tag{4-3}$$

In addition, we consider a slip effect in the lateral direction, denoted by the slip angle $\beta \in [0, \pi/2)$, given by:

$$\beta = \begin{cases} \operatorname{atan}(v_y/v_x)\,, & v_x \neq 0\,, \\ (\pi/2)\operatorname{sgn}(v_y)\,, & v_x = 0\,. \end{cases} \tag{4-4}$$

Now, we can rewrite the nonholonomic constraint given by Eq. (3-1), as a function of the slip angle β [37]:

$$\dot{x}\sin(\theta - \beta) - \dot{y}\cos(\theta - \beta) = 0\,. \tag{4-5}$$

In this context, the kinematic model of the DDMR can be expressed in terms of the slipping coefficient κ and slip angle β as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \kappa\cos(\theta - \beta) \\ \kappa\sin(\theta - \beta) \\ 0 \end{bmatrix} v \; + \; \begin{bmatrix} 0 \\ 0 \\ \kappa \end{bmatrix} \omega\,. \tag{4-6}$$

Finally, using addition and subtraction formulas of trigonometric identities, we obtain the following robot kinematic model which depends only on the robot orientation θ:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \kappa(\cos\theta + \delta\sin\theta) \\ \kappa(\sin\theta - \delta\cos\theta) \\ 0 \end{bmatrix} v_x + \begin{bmatrix} 0 \\ 0 \\ \kappa \end{bmatrix} \omega\,, \tag{4-7}$$

where $\delta = \tan\beta$ is a disturbance term and $v_x = v\cos\beta \in \mathbb{R}$ is the longitudinal velocity of the DDMR. Figure 4.2 shows a representation of the kinematic model of the DDMR including the slipping coefficient κ and slip angle β.

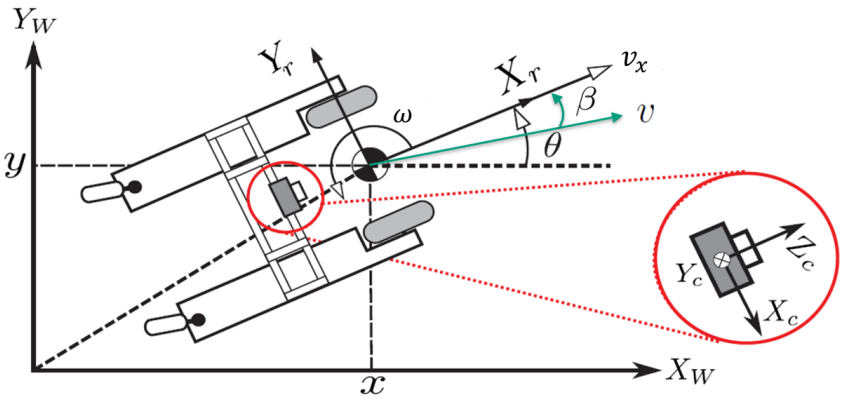

Figure 4.2: Schematic of the kinematic model of a DDMR, the generalized coordinates $q = (x, y, \theta)$ and the robot inputs $u = (v, \omega)$ under the effects of wheel slippage κ and sideslip β.

4.3 Robust Kinematic Controller Design

In this section, we present a robust kinematic control design based on the STA approach for trajectory tracking in the presence of external disturbances, due to wheel slippage and vehicle sideslip caused by rough and sloped terrains [37]. Here, the key idea is to design a robust cascade controller in the configuration space to ensure the accomplishment of the following control goal:

$$q \to q_d(t)\,, \qquad \tilde{q} = q - q_d(t) \to 0\,, \tag{4-8}$$

where $\tilde{q} = (\tilde{x}, \tilde{y}, \tilde{\theta}) \in \mathbb{R}^{3\times 1}$ is the Cartesian tracking error and $q_d(t) \in \mathbb{R}^{3\times 1}$ is the desired trajectory generated by a virtual reference DDMR with the same kinematic constraints of the real DDMR and, therefore, feasible in practice. Feasible reference trajectories must satisfy the motion of a reference frame $\mathcal{F}_d$ rigidly attached to the virtual reference DDMR. Then, let us define the desired reference trajectory $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$ as smooth time functions that

serve as solution to the kinematic model:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \kappa(\cos\theta_d + \delta \sin\theta_d) \\ \kappa(\sin\theta_d - \delta \cos\theta_d) \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 0 \\ 0 \\ \kappa \end{bmatrix} \omega_d \,, \tag{4-9}$$

where $u_d = (v_d, \omega_d) \in \mathbb{R}^{2\times 1}$ are the desired input velocities provided by the rIBVS approach described by Eq. (3-24) and Eq. (3-26) respectively. Then, the tracking problem can be formulated as determine a feedback control $\mathbf{v} = (v, \omega)$ for asymptotically stabilizing the reference trajectory $q_d(t)$ such that the Cartesian tracking error $\tilde{q} := (x - x_d,\, y - y_d,\, \theta - \theta_d)$ asymptotically goes to zero. Figure Fig. 4.3 shows a block diagram that represents in more detail how the previously mentioned control approach is designed. As depicted in the schematic, the rIBVS approach is responsible for dealing with disturbances in the image space, represented by η_i. On the other hand the rTTC approach must reject perturbations caused by wheel slippage and vehicle sideslip. In addition, to find the reference trajectory q_d, the Euler method is used to integrate $\dot{q}_d$ obtained after applying the desired inputs u_d in the virtual robot. In Fig. 4.3, we can also see the coordinate transformations T_{qz} and T_{uv} that will be described in the next subsections.

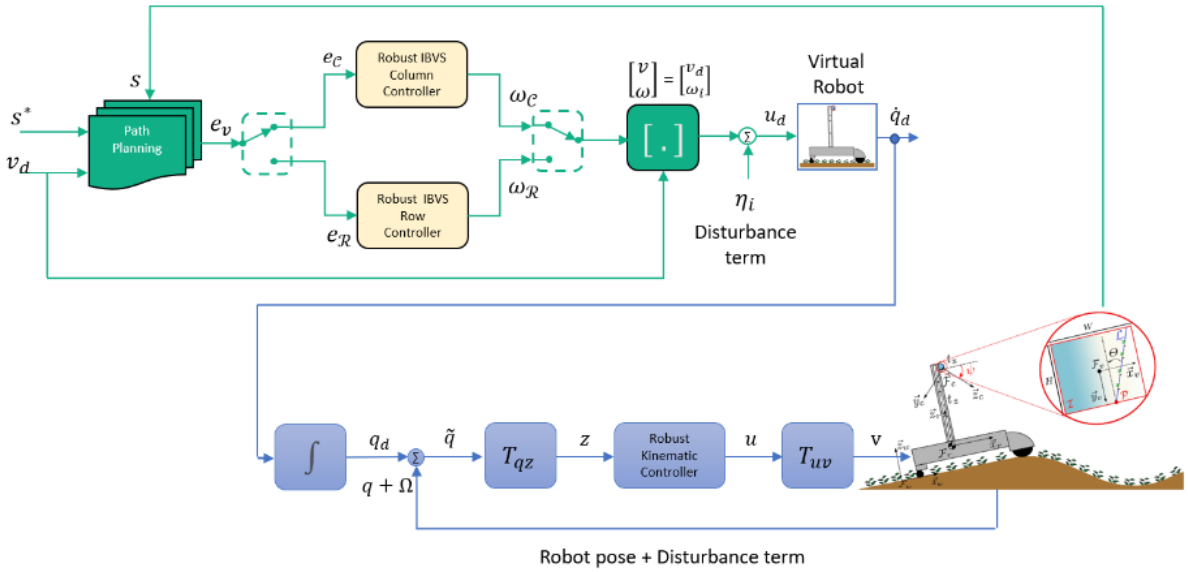

Figure 4.3: Block diagram representing the Robut Cascade-Based Controller (rCC) approach.

4.3.1 Kinematic Model Transformation

Here, we will introduce the coordinate transformations of the kinematic model into the chained form, which is more convenient for solving planning and control problems efficiently by using systematic procedures [36]. Equations in the chained form can be easily integrable in closed-form under appropriate inputs, simplifying the control design and the subsequent stability analysis [25].

Then, the tracking problem can be reformulated as follows: to find the stabilizing control laws $u = (u_1, u_2) \in \mathbb{R}^{3\times 1}$ based on nonlinear switching functions for the resulting chained system under disturbances, which asymptotically stabilizing the Cartesian tracking error $\tilde{q}$ at zero. Let us consider the following change of coordinates and control variables $(\tilde{x}, \tilde{y}, \tilde{\theta}, v, \omega) \mapsto (z_1, z_2, z_3, u_1, u_2)$ via coordinate and input transformations:

$$z = T_{zq}(q)\,\tilde{q}\,, \qquad u = T_{uv}(q)\,\mathbf{v}\,, \tag{4-10}$$

where

$$T_{zq} = \begin{bmatrix} 0 & 0 & 1 \\ \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \end{bmatrix}, \quad T_{uv} = \begin{bmatrix} 0 & 1 \\ 1 & -z_3 \end{bmatrix}, \tag{4-11}$$

with $z = (z_1, z_2, z_3) \in \mathbb{R}^{3\times 1}$ being the chained form coordinates whereas $T_{zq} \in \mathbb{R}^{3\times 3}$ and $T_{uv} \in \mathbb{R}^{2\times 2}$ are respectively the coordinate and input transformation matrices.

After taking the time-derivative of z from Eq. (4-10), and using Eq. (4-7) and Eq. (4-9), we can write the perturbed chained system as:

$$\dot{z}_1 = u_1 - \omega_d + f_1\,, \tag{4-12.1}$$

$$\dot{z}_2 = u_2 - v_d \cos z_1 + f_2\,, \tag{4-12.2}$$

$$\dot{z}_3 = z_2\, u_1 - v_d \sin z_1 + f_3\,, \tag{4-12.3}$$

where $f_1 \in \mathbb{R}$, $f_2 \in \mathbb{R}$, and $f_3 \in \mathbb{R}$ are bounded disturbances given by:

$$f_1 = [\kappa - 1]\,(\omega - \omega_d)\,, \tag{4-13.1}$$

$$f_2 = [\kappa - 1]\,(v - \omega z_3 - v_d \cos z_1) + \kappa\,\delta\, v_d \sin z_1\,, \tag{4-13.2}$$

$$f_3 = [\kappa - 1]\,(\omega z_2 - v_d \sin z_1) + \kappa\,\delta\,(v - v_d \cos z_1)\,. \tag{4-13.3}$$

Notice that, for the case of autonomous navigation in flat terrains, where $\kappa = 1$ and $\delta = 0$, the tracking problem for the perturbed chained system, given by Eq. (4-12.1)-Eq. (4-12.3), is simplified for the conventional tracking problem for chained form systems.

4.3.2
Robust Trajectory Tracking

For asymptotically stabilizing the perturbed chained form system described by Eq. (4-12.1)-Eq. (4-12.3), we design a new robust tracking controller based on the STA approach and the stabilizing control laws proposed in [30] with additional robustness terms, as shown below:

$$u_1 = -\alpha z_1 - b z_2 |\sigma|^{0.5} \mathrm{sgn}(\sigma) + w_1 + \omega_d \,, \tag{4-14.1}$$

$$u_2 = -\alpha z_2 + b z_1 |\sigma|^{0.5} \mathrm{sgn}(\sigma) + w_2 + v_d \cos z_1 \,, \tag{4-14.2}$$

where $\sigma = (2z_3 - z_1\, z_2)$ is the sliding surface, $b \in \mathbb{R}^+$ is a positive constant gain and $\alpha \in \mathbb{R}^{0+}$ is a non-negative variable gain define as a function of the coordinates z_1 and z_2:

$$\alpha(z_1, z_2) = \begin{cases} \alpha_0 \,, & \text{per} \quad z_1^2 + z_2^2 > \varepsilon^2 \,, \\ 0 \,, & \text{per} \quad z_1^2 + z_2^2 \leq \varepsilon^2 \,, \end{cases} \tag{4-15}$$

with $\alpha_0 \in \mathbb{R}$ being a positive constant and $\varepsilon \in \mathbb{R}$ is a positive constant which defines the radius of a ε-sphere of the z_1, z_2-space origin. The robustness terms $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$, designed according to the super-twisting algorithm, are given by:

$$w_1 = -\rho_{11} \, |z_1|^{0.5} \, \mathrm{sgn}(z_1) + \zeta_1 \,, \tag{4-16.1}$$

$$w_2 = -\rho_{21} \, |z_2|^{0.5} \, \mathrm{sgn}(z_2) + \zeta_2 \,, \tag{4-16.2}$$

with $\dot{\zeta}_1 = -\rho_{12} \, \mathrm{sgn}(z_1)$ and $\dot{\zeta}_2 = -\rho_{22} \, \mathrm{sgn}(z_2)$, where $\rho_{11}, \rho_{12} \in \mathbb{R}^+$ and $\rho_{21}, \rho_{22} \in \mathbb{R}^+$ are positive gains. Then, we can state the following theorem:

Theorem 4.1 *Consider the chained form system defined by Eq. (4-12.1)-Eq. (4-12.3) under the disturbance terms given by Eq. (4-13.1)-Eq. (4-13.3) assumed to be bounded. Under the robust SMC-based controllers given by Eq. (4-14.1) and Eq. (4-14.2) respectively, the following properties hold: (i) all systems signals are bounded; (ii)* $z_1(t), z_2(t) \to 0$*; (iii)* $\sigma(t), z_3(t) \to 0$*; and (iv)* $\tilde{q}(t) \to 0$*. (iv)* $\tilde{x}(t), \tilde{y}(t) \to 0$*; (v)* $\tilde{\theta}(t) \to 0$ *in finite time.*

4.4
3D Computer Simulation

Here, we present 3D computer simulations and results of the rCC approach designed for autonomous visual navigation in row crop fields, using the SoyBot [10] equipped with a monocular camera. An example of a 3D agricultural environment with sloped and rough terrain, as well as sparsely

distributed plants, was created in ROS-Gazebo platform, as shown in Fig. 4.4. The WMR and the world (i.e., the crop field) models, as explained in chapter 3, are described in an SDF file, which contains all their physical and geometrical characteristics. The environment has multiple crop-rows created choosing a simple plant 3D model, called *acanthus*. The *heightmap* image, which was converted into the 3D model used for the simulations of this chapter, and the environment itself may be visualized in Fig. 4.5(a) and Fig. 4.5(b), respectively.

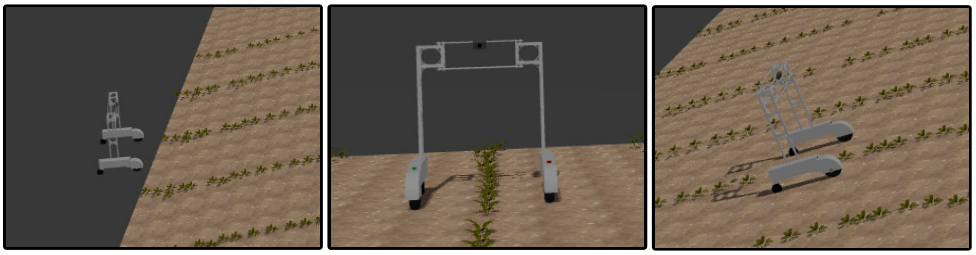

Figure 4.4: SoyBot robot navigating in the 3D agricultural environment created in Gazebo platform, with sparse plants and rough terrains.

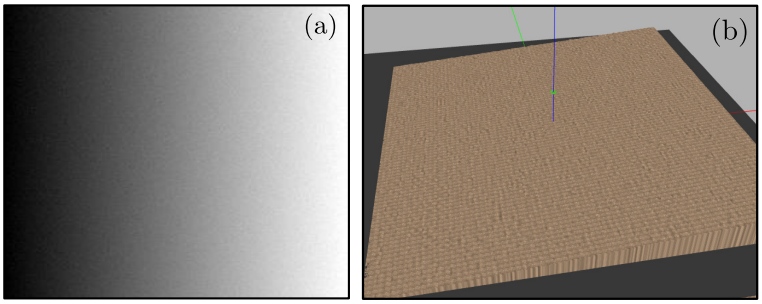

Figure 4.5: (a) 2D grayscale representing the heightmap used to create (b) the 3D environment used to simulate this comparative study.

The parameters of the camera-robot system used in the simulations were previously shown in subsection 3.7.1 The ROS Nodes developed for the simulation were written using Python with OpenCV libraries. The communication between the sensors and the plugin was configured using ROS version Melodic Morenia, running on the operating Linux Ubuntu 18.04, using an Intel Core i5-8250U 1.6 GHz, 8GB DDR4 RAM. In the simulated tests , we considered 20% of uncertainty in the extrinsic and intrinsic camera calibration parameters, a slope angle of the terrain of approximately $\pi/9$ *rad*. In addition, we assume a slipping coefficient $\kappa = 0.6$ and a disturbance term $\delta = 0.52$ to validate the robustness of the rCC approach.

Here, we will show a comparative study among the rCC approach and the pure rIBVS approach, given by Eq. (3-24) and Eq. (3-26). The methods were applied taking into account that the robot is climbing a sloped terrain with roughness. Moreover, we consider a high driving velocity $v_d = 1.5\ ms^{-1}$,

relaxing the low-speed assumption used in [22]. Other parameters are: rate of control loop $h = 30\,Hz$ for a simulation time of $T_{max} = 40\,s$. The control parameters used in the simulations are presented in Table 4.1. The gains Λ_c, $\Lambda_\mathcal{R}$, K_c, $K_\mathcal{R}$, μ_c and $\mu_\mathcal{R}$ are chosen as described in subsection 3.7.3, as well as, the tested ranges. The gain α_0 was evaluated in the interval $[0.1, 20.5]$, where very low values makes variable z_1 and z_2 not to reach an acceptable value of convergence. On the other hand, values near to the upper bound, cause the system to approach instability. The gain b has been selected in such a way that the sliding surface σ converge to about 10^{-3}. Small values of b, makes this gain result unachievable, and on the other hand, high values become the system unstable. The variable ε was selected so that z_1 and z_2 reach values with order of magnitude 10^{-3}. Thereby, it is possible to ensure that the real robot tracks the virtual robot with acceptable accuracy. The robust gains ρ_{11} and ρ_{21} were increased until the controller could deal with the uncertainties and make the system perform the task satisfactorily. Finally the gains ρ_{12} and ρ_{22} were chosen small, since high values can make the system unstable, because they are multipled by integral terms. Moreover, the tested range for these gains is $[0.01, 0.1]$. Table.4.1 shows the control paramaters for the rIBVS and rCC approaches used to produce the simulation results shown in this section.

Table 4.1: Control parameters for the rIBVS and rCC approaches.

Parameters	rIBVS (STA)	rCC
Λ_c	$\text{diag}(0.3, 0.6)$	$\text{diag}(0.3, 0.6)$
$\Lambda_\mathcal{R}$	$\text{diag}(2.5, 5.0)$	$\text{diag}(2.5, 5.0)$
K_c	$0.2I$	$0.2I$
$K_\mathcal{R}$	$0.2I$	$0.2I$
$\mu_c, \mu_\mathcal{R}$	$0.05I$	$0.05I$
α_0	-	6.5
b	-	7.5
ε	-	8×10^{-4}
ρ_{11}	-	0.5
ρ_{21}	-	4.5
ρ_{12}, ρ_{22}	-	0.05

Figures 4.6-4.8 present the main results of various numerical simulations for autonomous visual navigation task in a sloped terrain with roughness. Figures 4.6(a)-(f) show the evolution over time of the image errors and coordinates. At the beginning, we can observe that both rIBVS and rCC approaches are able to make the robot reach the crop row, since, the image coordinate y_v is driven to the desired value $H/2$. However, when the robot is moving up

the sloped terrain, the rIBVS approach is not capable of keeping the robot aligned with the crop row, which can be visualized by the behavior of the image coordinates x_v and θ_v, and their respective errors. On the other hand, the rCC approach is able to reject the disturbances in the kinematic model of the DDMR, as well as the perturbations and inaccuracies in the image space.

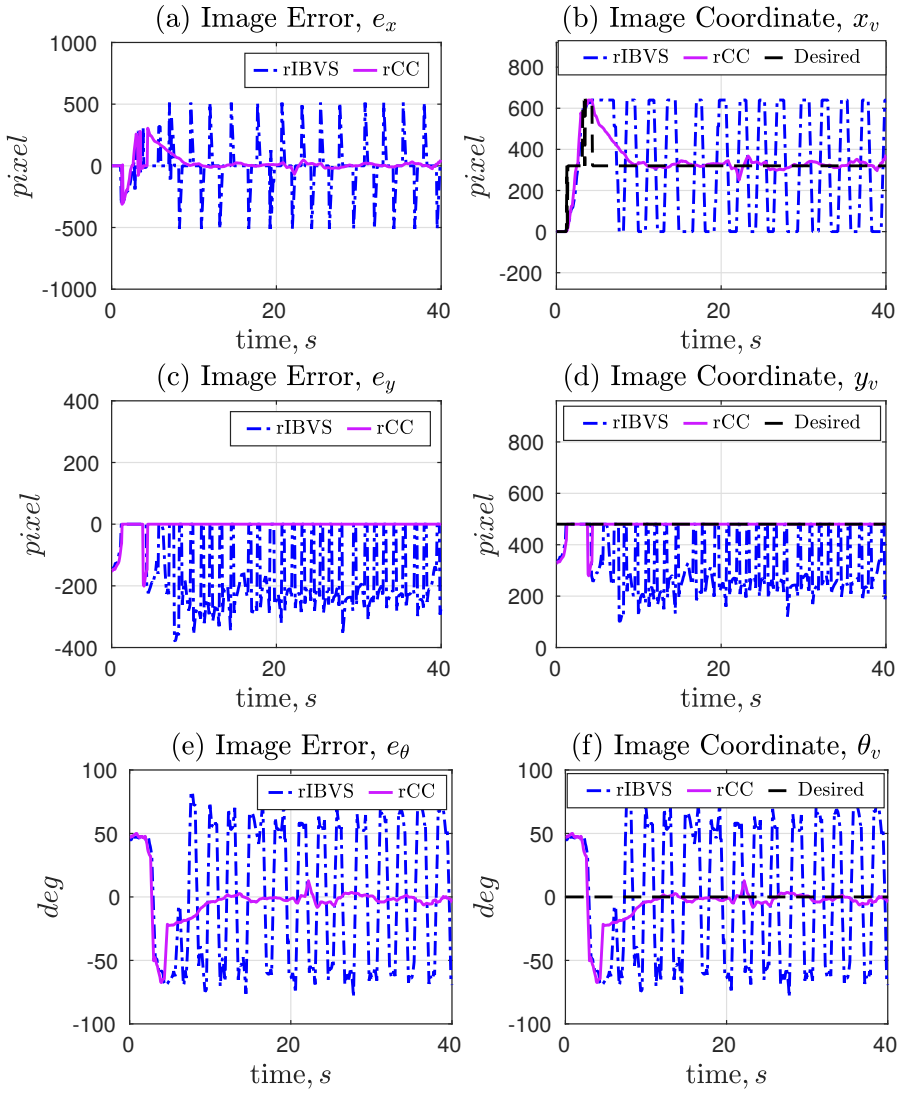

Figure 4.6: Comparative study among the rIBVS,and rCC approaches: (a)-(b) image error, e_x and image coordinate, x_v; (c)-(d) image error, e_y and image coordinate, y_v; (e)-(f) image error, e_θ and image coordinate, θ_v

Therefore, the robot performs the desired task, as can be seen in Fig. 4.7(a)-(d). Finally, the real robot inputs (v, ω) are presented in Fig 4.8(a)-(b) and the robot wheel speeds (ω_L, ω_R) are depicted in Fig 4.8(c)-(d). In these graphics, we realize that the real robot can track the desired robot, with the rCC approach, since the inputs of the real robot achieve the inputs of the desired robot.

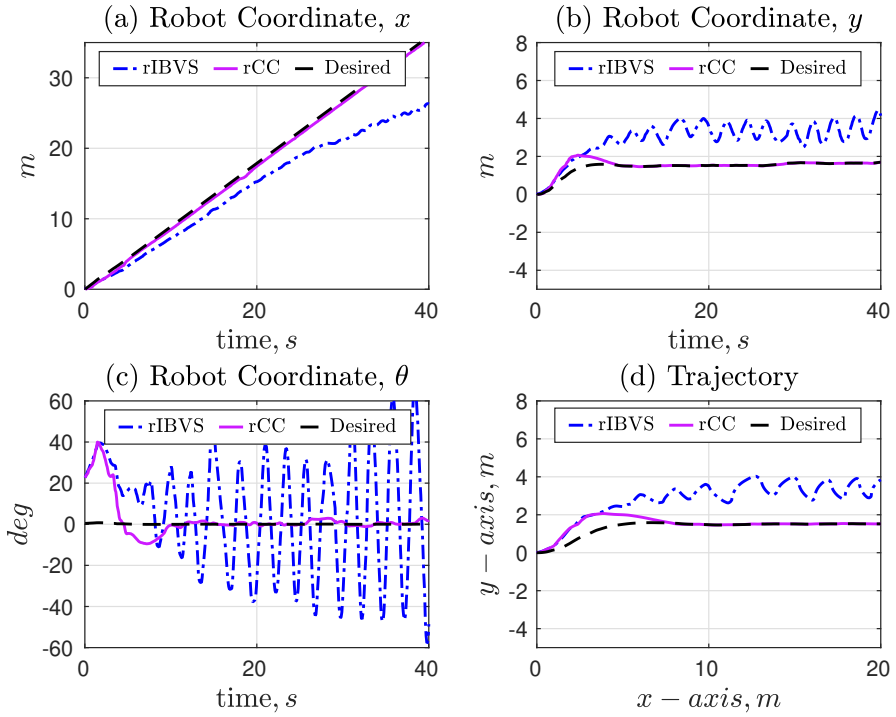

Figure 4.7: Comparative study among the rIBVS and rCC approaches: (a)-(c) robot coordinates, x_r, y_r and θ_r; (d) trajectory traveled by the robot navigating along the crop row.

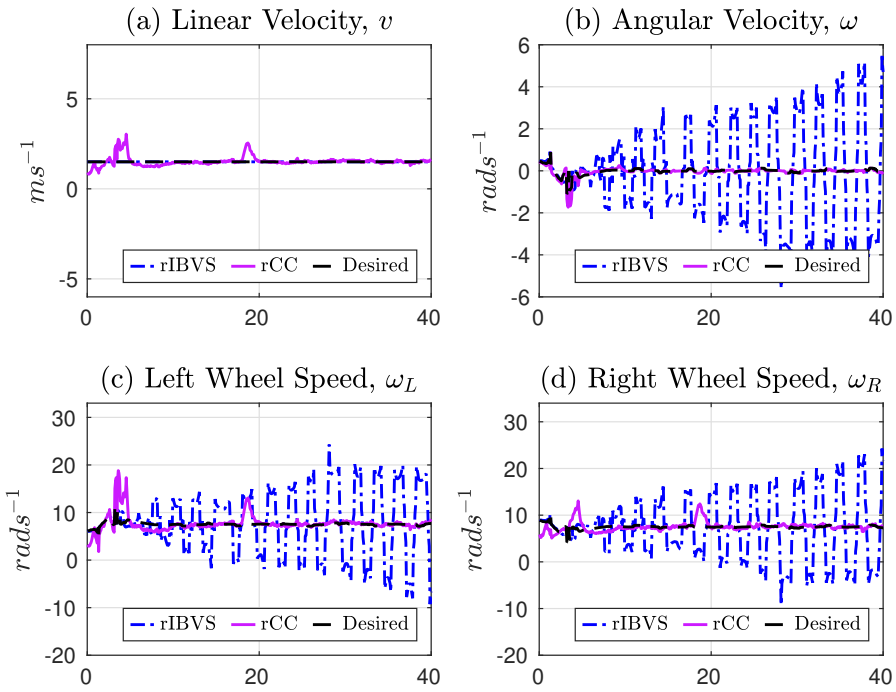

Figure 4.8: Comparative study among the rIBVS and rCC approaches: (a)-(b) robot inputs; (c)-(d) left and right wheel speeds.

In addition, we use two performance metrics for a comparative study, as presented in Table 4.2: (i) the mean-absolute error (MAE) of the image features e_x, e_y, and e_θ; (ii) the root mean squared error (RMSE) of the image features e_x, e_y and e_θ; and (iii) the mean-absolute deviation (MAD) of the control signals v, ω, ω_L and ω_R. The metrics MAE and MAD were chosen to analyze the transient and steady regimes with the same weights. In addition, to facilitate the comparison with the performance of the rIBVS controller applied to the case approached in chapter 2, we include the RMSE values in Table 4.2. Looking at the MAE measure, we realize that the rCC approach achieves much smaller values than the rIBVS approach, which fails to make the system asymptotically stable. It is also possible to do this analysis by the RMSE measure. However, RMSE are higher compared to MAE, as MAE places greater weights on the transient regime of the system.

Furthermore, notice that Table 4.2 shows that the rCC approach also leads to much has lower MAD metrics for the control signals, which may be interesting for saving battering. Notice that, since, the linear velocity v sent to the robot is equal to the desired one and this velocity is not measured in the rIBVS approach, the MAD value is 0 in this case.

Table 4.2: Performance metrics: MAE, RMSE and MAD.

MAE	rIBVS	rCC	unit
x_v	31.11	12.78	cm
y_v	38.78	1.94	cm
θ_v	49.47	7.56	deg
RMSE	rIBVS	rCC	unit
x_v	63.82	26.20	cm
y_v	47.86	8.20	cm
θ_v	53.25	20.87	deg
MAD	rIBVS	rCC	unit
v	0	0.147	$m\,s^{-1}$
ω	2.18	0.105	$rad\,s^{-1}$
ω_L	6.53	0.621	$rad\,s^{-1}$
ω_R	6.53	0.946	$rad\,s^{-1}$

4.5 Final Remarks

In this chapter, it was developed a robust cascade-based control (rCC) methodology for wheeled mobile robots to perform autonomous navigation in agricultural fields using a single monocular camera, considering the effects of

all disturbances sources. The key idea is to connect a robust image-based visual servoing (rIBVS) approach, with a robust trajectory tracking (rTTC) to handle uncertainties caused by camera miscalibration, and external disturbances due to high driving velocities, sparse plants, and changes in the slope and roughness of the terrain First, the rIBVS approach is designed to reject inaccuracies and perturbations in the image space, and afterwards, the rTTC is applied to ensure successful execution of row crop following tasks under wheel slippage and vehicle sideslip. A comparative study has been conducted considering two control techniques, the pure rIBVS controller, presented in chapter 3, and the rCC controller, described in this chapter, both based on the super twisting algorithm (STA) approach. The controllers performance has been analyzed by looking at the MAE and MAD metrics applied to the image features and control signals. The 3D computer simulations are carried out in a row-crop field created in ROS-Gazebo platform, using a differential-drive mobile robot (DDMR). The agricultural environment has been built to incorporate slope and roughness in the terrain, as well as, sparse plants, making the scenario more realistic.

5
Conclusion and Future Works

In this chapter, we present the concluding remarks about the design of the Robust Image-based Visual Servoing and the Cascade Robust Controller approaches for row crop following with wheeled mobile robots. In addition, we added a brief discussion of terms perspectives for future works.

5.1
Concluding Remarks

In this work, we have proposed robust approaches for wheeled mobile robots to perform row crop following using a singles monocular camera. In Chapter 3, we designed a robust image-based visual servoing (rIBVS) approach based on STA, for WMR. The main idea is to include a robustness term in the vision-based control law to handle model inaccuracies caused by camera miscalibration and trajectory perturbations in the image space, due to high driving velocities, sparse plants, and uneven terrains. Three comparative studies have been carried out to validate the developed strategy. The first one considers three control techniques, the classic IBVS controller with row and column primitives, and two robust controllers based on the sliding mode control approach. Then, the second comparative study is carried out by replacing the SMC-based rIBVS with the STA-based rIBVS. Finally, the third comparative study compares the performance between the SMC-based rIBVS and the STA-based rIBVS. The performance of the controllers has been evaluated using the RMSE and MAD metrics applied to the image errors and control signals. Analyzing the results shown in Tables 3.2 and 3.4, we see that both rIBVS controllers present smaller RMSE for all image features coordinates, compared with the classic IBVS and UVC approaches. Therefore, the rIBVS approach has a better performance on the transient regime with fewer oscillations, since the RMSE metrics penalize more the most significant errors. These behaviors are also visualized in the graphs of Figs.3.9 and 3.12. Thereby, it is noted that for comparative study I, the system is oscillatory for the classic iBVS, and for the second comparative study, this method makes the system unstable. In addition, Tables 3.2 and 3.4 show that the rIBVS techniques have lower MAD metrics for the control signals, which can be

relevant to save battery during the operations. In the last comparative study presented in subsection 3.7.4, we compared the rIBVS approaches developed for the two previously cited comparative studies. Then, analyzing the RMSE metrics described in Table 3.6, we note that the rIBVS controller based on the SMC approach reaches higher values than the rIBVS controller based on the STA approach, for all image features coordinates. Therefore, the rIBVS controller based on the STA approach has a smaller oscillatory behavior and does not have so many accentuated perturbations, since the RMSE metrics penalize the most significant errors. Moreover, the technique that has the lowest MAD metrics for the control signals, according to the results present in Table 3.6, is the rIBVS controller based on STA, indicating that this method is more energy efficient. The stability and robustness properties of the rIBVS controllers have been analyzed using the Lyapunov stability theory. Afterward, in Chapter 4, we present the development of a robust cascade-based control approach, taking into account all disturbances sources. To achieve this goal, we connected the rIBVS approach, with robust trajectory tracking. Therefore, it is possible to deal with uncertainties in the image space, and perturbations caused by wheel slippage and vehicle sideslip. A comparative study has been conducted considering two control strategies based on the super twisting algorithm approach, the pure rIBVS and rCC controllers. The performance has been analyzed measuring the MAE, RMSE, and MAD metrics applied to the image features and control signals. Observing the results shown in Table 4.2, we realize that the rCC approach reaches significantly lower MAE and RMSE metrics compared with rIBVS approach, which fails to make the system asymptotically stable. This analysis can also be done by looking at the graphs presented in section 4.4. It is also possible to notice that rCC approach has smaller MAD metrics than rIBVS approach for angular velocity ω and wheel speeds ω_L and ω_R. This may be interesting for saving battering. In both Chapters, 3D computer simulations are carried out in a row-crop field created in the ROS-Gazebo platform, using a differential-drive mobile robot named Soybot. We built the agricultural environment inserting slope and roughness in the terrain, as well as, sparse plants.

5.2 Future Works

The following topics will be approached in future works of this research:

– Develop a stability proof based on Lyapunov theory for the Robust Image-Based Visual Servoing (rIBVS) approach developed in Chapter 3.

– Produce a stability proof based on Lyapunov theory for the Cascade Robust Controller approach developed in Chapter 4.

– Apply optimization methods to choose the control gains of the approaches described in Chapters 3 and 4.

– Design a novel control technique to deal with uncertainties and perturbations in the dynamic model of the mobile robot.

– Extend the Cascade Robust Controller approach, so that we can relax the assumption (A7), which mentions that the wheel speeds do not change abruptly during autonomous maneuvers.

– Carry out experimental tests with real robots and environments, to proceed with the experimental validation of the strategies developed in this work.

In addition, we will need to deal with some challenges that arise when we navigate autonomously in real environments, which are caused by:

– Variation in lighting during the execution of the task, since outdoor environments suffer from changes in sunlight during the day.

– Perturbations caused by navigation in types of terrain that were not tested in Gazebo Platform, such as sandy terrains.

– Wear on the physical components of the real robot over time.

– Obstacles during the navigation task in agricultural environments, such as wild animals, workers and others.

– Gain adjustments for the rIBVS and rCC approaches, so that the task is performed with high accuracy.

Bibliography references

[1] SLOTINE, J.-J. E.; LI, W. ; OTHERS. **Applied Nonlinear Control**. Prentice hall Englewood Cliffs, NJ, 1991.

[2] 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 International Publishing, 2016.

[3] Billingsley, J., editor. Burleigh Dodds Science Publishing, 2019.

[4] GAO, X.; LI, J.; FAN, L.; ZHOU, Q.; YIN, K.; WANG, J.; SONG, C.; HUANG, L. ; WANG, Z.. **Review of Wheeled Mobile Robots' Navigation Problems and Application Prospects in Agriculture**. IEEE Access, 6:49248–49268, 2018.

[5] BURGOS-ARTIZZU, X. P.; RIBEIRO, A.; GUIJARRO, M. ; PAJARES, G.. **Real-time Image Processing for Crop/weed Discrimination in Maize Fields**. Computers and Electronics in Agriculture, 75(2):337–346, 2011.

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

[7] 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.

[8] FENNIMORE, S. A.; CUTULLE, M.. **Robotic Weeders Can Improve Weed Control Options for Specialty Crops**. Pest Management Science, 75(7):1767–1774, 2019.

[9] XAUD, M. F. S.; LEITE, A. C. ; FROM, P. J.. **Thermal Image Based Navigation System for Skid-Steering Mobile Robots in Sugarcane Crops**. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 1808–1814, Montreal, QB, Canada, 2019.

[10] BARBOSA, W. S.; OLIVEIRA, A. I. S.; BARBOSA, G. B. P.; 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**. In: 12TH INTERNATIONAL CONFERENCE ON DEVELOPMENTS IN ESYSTEMS ENGINEERING, p. 557–562, 2019.

[11] 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**. In: 12TH INTERNATIONAL CONFERENCE ON DEVELOPMENTS IN ESYSTEMS ENGINEERING, p. 563–568, Kazan, Russia, 2019.

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

[13] KANAGASINGHAM, S.; EKPANYAPONG, M. ; CHAIHAN, R.. **Integrating Machine vision-based Row Guidance with GPS and Compass-based Routing to achieve Autonomous Navigation for a Rice Field Weeding Robot**. Precision Agriculture, 21:831–855, 2020.

[14] WINTERHALTER, W.; FLECKENSTEIN, F.; DORNHEGE, C. ; BURGARD, W.. **Localization for Precision Navigation in Agricultural Fields—Beyond Crop Row Following**. Journal of Field Robotics, p. 1–23, 2020.

[15] KONOLIGE, K.; NÜCHTER, A.. **Range Sensing**, 2016.

[16] LE, T. D.; PONNAMBALAM, V. R.; GJEVESTAD, J. G. O. ; FROM, P. J.. **A Low-cost and Efficient Autonomous Row-following Robot for Food Production in Polytunnels**. Journal of Field Robotics, 37(2):309–321, 2020.

[17] STACHNISS, C.; LEONARD, J. J. ; THRUN, S.. **Simultaneous Localization and Mapping**. In: SPRINGER HANDBOOK OF ROBOTICS, p. 1153–1176. 2016.

[18] CLAUS, B.; KEPPER IV, J. H.; SUMAN, S. ; KINSEY, J. C.. **Closed-Loop One-Way-Travel-Time Navigation Using Low-Grade Odometry for Autonomous Underwater vehicles**. Journal of Field Robotics, 35(4):421–434, 2018.

[19] DURRANT-WHYTE, H.; HENDERSON, T. C.. **Multisensor Data Fusion**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 867–896. Springer International Publishing, 2016.

[20] DENG, F.; YANG, H.-L. ; WANG, L.-J.. **Adaptive Unscented Kalman Filter Based Estimation and Filtering for Dynamic Positioning with Model Uncertainties**. International Journal of Control, Automation and Systems, 17(3):667–678, 2019.

[21] BONIN-FONT, F.; ORTIZ, A. ; OLIVER, G.. **Visual Navigation for Mobile Robots: A Survey**. Journal of Intelligent and Robotic Systems, 53:263–298, 2008.

[22] CHERUBINI, A.; CHAUMETTE, F. ; ORIOLO, G.. **Visual Servoing for Path Reaching with Nonholonomic Robots**. Robotica, 29(7):1037—-1048, 2011.

[23] PASTEAU, F.; NARAYANAN, V. K.; BABEL, M. ; CHAUMETTE, F.. **A Visual Servoing Approach for Autonomous Corridor Following and Doorway Passing in a Wheelchair**. Robotics and Autonomous Systems, 75:28–40, 2016.

[24] AHMADI, A.; NARDI, L.; CHEBROLU, N. ; STACHNISS, C.. **Visual Servoing-based Navigation for Monitoring Row-Crop Fields**. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 4920–4926, 2020.

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

[26] BROCKETT, R. W.; OTHERS. **Asymptotic Stability and Feedback Stabilization**. Differential Geometric Control Theory, 27(1):181–191, 1983.

[27] DIXON, W.; DAWSON, D. M.; ZERGEROGLU, E. ; BEHAL, A.. **Nonlinear Control of Wheeled Mobile Robots**. Springer Verlag London, 2001.

[28] KOZŁOWSKI, K.; PAZDERSKI, D.. **Modeling and Control of a 4-Wheel Skid-Steering Mobile Robot**. International journal of applied mathematics and computer science, 14:477–496, 2004.

[29] FU, J.; CHAI, T.; SU, C.-Y. ; JIN, Y.. **Motion/Force Tracking Control of Nonholonomic Mechanical Systems via Combining Cascaded Design and Backstepping**. Automatica, 49(12):3682–3686, 2013.

[30] BLOCH, A.; DRAKUNOV, S.. **Stabilization and Tracking in the Nonholonomic Integrator via Sliding Modes**. Systems & Control Letters, 29(2):91–99, 1996.

[31] SHTESSEL, Y.; EDWARDS, C.; FRIDMAN, L. ; LEVANT, A.. **Sliding Mode Control and Observation**. Springer Science+Business Media, 2014.

[32] THOMAS, M.; BANDYOPADHYAY, B. ; VACHHANI, L.. **Posture Stabilization of Unicycle Mobile Robot Using Finite Time Control Techniques**. IFAC-PapersOnLine, 49(1):379–384, 2016.

[33] MORENO, J. A.; OSORIO, M.. **A Lyapunov Approach to Second-Order Sliding Mode Controllers and Observers**. In: 2008 47TH IEEE CONFERENCE ON DECISION AND CONTROL, p. 2856–2861, Cancun, Mexico, 2008.

[34] MORENO, J. A.. **A linear framework for the robust stability analysis of a Generalized Super-Twisting Algorithm**. In: 2009 6TH INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING, COMPUTING SCIENCE AND AUTOMATIC CONTROL (CCE), p. 1–6.

[35] MORENO, J. A.; OSORIO, M.. **Strict Lyapunov Functions for the Super-twisting Algorithm**. IEEE Transactions on Automatic Control, 57(4):1035–1040, 2012.

[36] SAMSON, C.; MORIN, P. ; LENAIN, R.. **Motion Control of Wheeled Mobile Robots**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1235–1266. Springer International Publishing, 2016.

[37] NAGATANI, K.; ISHIGAMI, G. ; OKADA, Y.. **Modeling and Control of Robots on Rough Terrain**, p. 1267–1284. Springer International Publishing, 2016.

[38] WANG, D.; LOW, C. B.. **Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective**. IEEE Transactions on Robotics, 24(3):676–687, 2008.

[39] MOBAYEN, S.. **Finite-Time Tracking Control of Chained-Form Nonholonomic Systems with External Disturbances Based on Recursive Terminal Sliding Mode Method**. Nonlinear Dynamics, 80(1):669–683, 2015.

[40] BESSAS, A.; BENALIA, A. ; BOUDJEMA, F.. **Integral Sliding Mode Control for Trajectory Tracking of Wheeled Mobile Robot in Presence of Uncertainties**. Journal of Control Science and Engineering, 2016, 2016.

[41] MARTÍNEZ, E. A.; RÍOS, H. ; MERA, M.. **Robust Tracking Control Design for Unicycle Mobile Robots with Input Saturation**. Control Engineering Practice, 107:104676, 2021.

[42] PARK, B. S.; YOO, S. J.; PARK, J. B. ; CHOI, Y. H.. **Adaptive Neural Sliding Mode Control of Nonholonomic Wheeled Mobile Robots With Model Uncertainty**. IEEE Transactions on Control Systems Technology, 17(1):207–214, 2009.

[43] MUÑOZ-BENAVENT, P.; GRACIA, L.; SOLANES, J. E.; ESPARZA, A. ; TORNERO, J.. **Sliding Mode Control for Robust and Smooth Reference Tracking in Robot Visual Servoing**. International Journal of Robust and Nonlinear Control, 28(5):1728–1756, 2018.

[44] OLIVEIRA, T. R.; LEITE, A. C.; PEIXOTO, A. J. ; HSU, L.. **Overcoming Limitations of Uncalibrated Robotics Visual Servoing by Means of Sliding Mode Control and Switching Monitoring Scheme**. Asian Journal of Control, 16(3):752–764, 2014.

[45] BECERRA, H. M.; LÓPEZ-NICOLÁS, G. ; SAGÜÉS, C.. **A Sliding-mode-control Law for Mobile Robots based on Epipolar Visual Servoing from Three Views**. IEEE Transactions on Robotics, 27(1):175–183, 2011.

[46] BECERRA, H. M.; HAYET, J.-B. ; SAGÜÉS, C.. **A Single Visual-servo Controller of Mobile Robots with Super-twisting Control**. Robotics and Autonomous Systems, 62(11):1623–1635, 2014.

[47] WINTERHALTER, W.; FLECKENSTEIN, F. V.; DORNHEGE, C. ; BURGARD, W.. **Crop Row Detection on Tiny Plants with the Pattern Hough Transform**. IEEE Robotics and Automation Letters, 3(4):3394–3401, 2018.

[48] CHAUMETTE, F.; HUTCHINSON, S. ; CORKE, P.. **Visual Servoing**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 841–866. Springer International Publishing, 2016.

[49] MANHÃES, M. M. M.; SCHERER, S. A.; VOSS, M.; DOUAT, L. R. ; RAUSCHENBACH, T.. **UUV Simulator: A Gazebo-based package**

for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE MONTEREY, p. 1–8, 2016.

[50] ABBYASOV, B.; LAVRENOV, R.; ZAKIEV, A.; YAKOVLEV, K.; SVININ, M. ; MAGID, E.. **Automatic Tool for Gazebo World Construction: from a Grayscale Image to a 3D Solid Model**. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 7226–7232, 2020.

A
Stability and Robustness Analysis of rIBVS Approach

Here, we present the stability and robustness analysis of the proposed rIBVS controllers. Then, the following assumptions for the perturbation term η_i given by Eq. (3-22) hold for all image features vector s_i for $i=\{c,\mathcal{R}\}$ as:

$$sup_{\,t\geq 0}\,\|\xi_{si}\| < \Delta_{si} < \infty, \quad \forall \xi_{si}\,, \tag{A-1}$$

$$\|J_{\omega i}\hat{J}^{\dagger}_{\omega i} - I\| \leq \iota \leq 1\,, \tag{A-2}$$

$$\|J_{vi}v_d\| \leq \Upsilon < \infty\,. \tag{A-3}$$

The assumption given by Eq. (A-1) is satisfied because in Theorem 4.1 was assumed that ξ_{si} is bounded. To prove the assumption made in Eq. (A-2), we consider the boundedness of the norm of $J_{\omega i}$ given by the following inequalities:

$$0 < J_{mi} \leq \|J_{\omega i}\| \leq J_{Mi} < \infty\,, \tag{A-4}$$

where J_{mi} and J_{Mi} are the lower and upper bounds respectively. To satisfy the Eq. (A-2), we can choose $\hat{J}^{\dagger}_{\omega i}$ as:

$$\hat{J}^{\dagger}_{\omega i} = 2I/(J_{Mi} + J_{mi})\,. \tag{A-5}$$

Then, using Eq. (A-4) and Eq. (A-5) in the assumption given by Eq. (A-2), the following inequality can be found:

$$\|J_{\omega i}\hat{J}^{\dagger}_{\omega i} - I\| \leq (J_{Mi} - J_{mi})/(J_{Mi} + J_{mi}) = \iota \leq 1\,. \tag{A-6}$$

Finally, the last assumption given by Eq. (A-3) is satisfied, since the parameters that define J_{vi} and the velocity v_d are also assumed to be bounded. For the stabilization of the rIBVS system in closed-loop (i.e., the regulation problem), the image error equation is taken from Eq. (3-21) as:

$$\dot{s}_i = \dot{e}_i = \nu_i + \eta_i\,, \tag{A-7}$$

where $e_i \in \mathbb{R}^2$ is a generic image error which can be defined in terms of image features errors e_c and $e_{\mathcal{R}}$. Now, to demonstrate the stability and robustness properties of the rIBVS system, we choose the Lyapunov function candidate:

$$2V_i(\sigma_i) = \sigma_i^{\mathsf{T}}\sigma_i\,, \tag{A-8}$$

which must satisfy the following properties:

$$(P1) \quad V_i(\sigma_i) > 0\,, \tag{A-9}$$

$$(P2) \quad \dot{V}_i(\sigma_i) = \sigma_i^T \dot{\sigma}_i \leq -\alpha\,|\sigma_i|\ , \tag{A-10}$$

$$(P3) \quad \dot{V}_i(\sigma) = \sigma_i^{\mathsf{T}} \dot{\sigma}_i\,. \tag{A-11}$$

Notice that, $\sigma_i \in \mathbb{R}^2$ is a generic sliding surface which can be defined in terms of $\sigma_{\mathcal{R}}$ and $\sigma_{\mathcal{C}}$. Now, taking the time-derivative of $V(\sigma_i)$ and using Eq. (A-7), we obtain:

$$\dot{V}_i(\sigma_i) = \sigma_i^{\mathsf{T}}\,(\nu_i + \eta_i)\,. \tag{A-12}$$

Then, by choosing $\nu_i = -\Lambda\,\sigma_i - K_i \lceil \sigma_i \rfloor^p$, with $\Lambda_i = \Lambda_i^{\mathsf{T}} > 0$, yields:

$$\dot{V}(\sigma) = -\sigma^{\mathsf{T}} \Lambda \sigma + \sigma^T (\eta - K \lceil \sigma \rfloor^p)\,, \tag{A-13}$$

which gives:

$$\sigma_i^{\mathsf{T}} (\eta_i - K_i \lceil \sigma_i \rfloor^p) \leq \|\sigma_i\|\,(\|\eta_i\| - K_i)\,. \tag{A-14}$$

To fulfill the sliding condition given by Property (P2), which makes the sliding surface an invariant set, the robustness gain matrix K_i must satisfy:

$$K_i \geq \|\eta_i\|\ , \qquad \forall s_i, s_{di}\,, \tag{A-15}$$

which, in view of Eq. (A-1)-Eq. (A-3), implies that

$$\|\eta_i\| \leq \|J_{\omega i} \hat{J}_{\omega i}^{\dagger} - I\|\,(\|\Lambda_i\|\,\|e_i\| + \|\tau_i\|) + \|J_{vi} v_d\| + \|\xi_{si}\|\ .$$

Then, setting:

$$K \geq \frac{\beta + \Delta_{si} + \iota\,(\|\Lambda_i\|\,\|e_i\|)}{1 - \iota}\,, \tag{A-16}$$

results in:

$$\dot{V}_i(\sigma_i) = -\sigma_i^{\mathsf{T}} \Lambda_i \sigma_i + \sigma_i^T\,(\eta_i - K_i \lceil \sigma_i \rfloor^p)\ < 0\,, \tag{A-17}$$

which implies that $\dot{V}_i(\sigma_i)$ is negative definite for all $\sigma_i \neq 0$. Therefore, we conclude that $\dot{V}_i(\sigma_i) \in \mathcal{L}_\infty$ and $\sigma_i, e_i \in \mathcal{L}_\infty$ and, consequently, $\lim_{t\to\infty} \sigma_i(t), e_i(t) = 0$ [25]. Notice that, from Eq. (3-20), we can also conclude that $\omega \in \mathcal{L}_\infty$ provided that $\nu_i \in \mathcal{L}_\infty$, which proves the property (i) from Theorem 4.1. Finally, we can apply these results to the rIBVS system and conclude that $\sigma_{\mathcal{C}}, \sigma_{\mathcal{R}} \to 0$ and $e_x, e_y, e_\theta \to 0$, which prove the properties (ii), (iii) and (iv) from Theorem 4.1.