Pontifícia Universidade Católica
do Rio de Janeiro

Guilherme Neri de Souza

Sliding Mode Control for Single- and Multi-Legged Robots

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Programa de Pós Graduação em Engenharia Elétricada PUC-Rioin partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor : Prof. Wouter Caarls
Co-advisor: Prof. Antonio Candea Leite

Rio de Janeiro
October 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Guilherme Neri de Souza

Sliding Mode Control for Single- and Multi-Legged Robots

Dissertation presented to the Programa de Pós–graduação em Programa de Pós Graduação em Engenharia Elétricada PUC-Rioin partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Prof. Wouter Caarls
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 – DETEL/UERJ

Prof. Thiago Boaventura Cunha
Universidade de São Paulo – DEM/USP

Rio de Janeiro, October the 22nd, 2020

Guilherme Neri de Souza

Bachelor's degree in Electrical Engineering, emphasis on Electronics, Department of Electronics and Telecommunications (DETEL), at the Rio de Janeiro State University (UERJ).

Bibliographic data

Souza Neri, Guilherme

Sliding Mode Control for Single- and Multi-Legged Robots / Guilherme Neri de Souza; advisor: Wouter Caarls; co-advisor: Antonio Candea Leite. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2020.

v., 129 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 Elétrica – Teses. 3. Robôs Móveis com Pernas;. 4. Controle por Modo Deslizante;. 5. Robô Saltitante. 6. Robô Multi-pernas. 7. Analise de Estabilidade. I. Caarls, W.. II. Leite, Antonio C.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 620.11

To my parents, for their support
and encouragement.

Acknowledgments

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

I would like to first thank Professor Antonio C. Leite, for the technical guidance and supervision, and also for believing in me during the difficult moments and encouraging me to push my limits.

I wish to thank Professor Wouter Caarls, for the valuable tips and teachings in reinforcement learning and ROS that somehow have contributed to my professional growth and the development of this work.

I also wish to thank my colleagues Adalberto Igor Souza de Oliveira, Gustavo Bertagna Peixoto Barbosa, Yessica Rosas Cuevas, Francisco Lostalo Cruz and Franklin Cardeñoso Fernadez, for their patience, technical support, and funny moments at PUC-Rio, and also for making the workplace motivating and fun.

I also wish to thank my professors in the State University of Rio de Janeiro for their dedication to ensuring quality education despite all the adversities faced.

I also wish to thank the Pontifical Catholic University for their dedication to ensuring quality education.

Finally, I wish to thank my family and friends for their care and emotional support, especially within my last months of graduate studies.

Abstract

Souza Neri, Guilherme; Caarls, W. (Advisor); Leite, Antonio C. (Co-Advisor). **Sliding Mode Control for Single- and Multi-Legged Robots**. Rio de Janeiro, 2020. 129p. Dissertação de mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

In the last years, legged mobile robots have increased the interest of the robotics community because such mechanisms have higher versatility compared to wheeled and aerial mobile robots. These characteristics make robot with legs a viable solution for rescue and monitoring operations in irregular terrains and difficult to access locations. Although single-legged or multi-legged mechanisms can transverse any terrain, some of their disadvantages are higher complexity in modelling and control design and higher power consumption. In this work, the author considers the problem of modelling and robust control design for a class of legged mobile robots using the sliding mode control approach. A comparative study between a planning algorithm based on Fourier techniques and sliding mode controllers is presented for the stabilization problem of a hopping robot in flight phase. The author also proposes the stabilization of the posture of multi-legged mobile robots such as, hexapod and biped robot, using two different control approaches, the Cartesian regulation control and the sliding mode control. The Lyapunov stability theory is used to demonstrate the stability properties of the closed-loop control systems. Numerical simulations in MATLAB simulation software and computer simulations in Gazebo, an open-source 3D robotic simulator, are included to illustrate the performance and feasibility of the propose methodology.

Keywords

Legged Mobile Robots Sliding Mode Control Hoppping Robot Hexapod Robot Stability Analysis

Resumo

Souza Neri, Guilherme; Caarls, W.; Leite, Antonio C.. **Controle por Modos Deslizantes de Robôs com Uma e Múltiplas Pernas**. Rio de Janeiro, 2020. 129p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Nos últimos anos, os robôs móveis com pernas têm despertado o interesse da comunidade robótica, pois tais mecanismos apresentam maior versatilidade em relação aos robôs móveis de rodas e aéreos. Neste trabalho, o autor considera o problema de modelagem e projeto de controle robusto para uma classe de robôs móveis com pernas usando a abordagem de controle por modos deslizantes. Um estudo comparativo entre um algoritmo de planejamento baseado em técnicas de Fourier e controladores via modo deslizante é apresentado para o problema de estabilização de um robô móvel saltitante na fase de vôo. O autor também propõe a estabilização da postura de robôs móveis multipernas, como hexapod e robô bípede, utilizando duas abordagens de controle diferentes, o controle de regulação Cartesiana e o controle via modos deslizantes. A teoria de estabilidade de Lyapunov é usada para demonstrar as propriedades de estabilidade dos sistemas de controle em malha-fechada. Simulações numéricas em ambiente de simulação MATLAB e simulações computacionais em Gazebo, um simulador robótico 3D de código aberto, são incluídas para ilustrar o desempenho e a viabilidade da metodologia proposta.

Palavras-chave

Robôs Móveis com Pernas; Controle por Modo Deslizante; Robô Saltitante Robô Multi-pernas Analise de Estabilidade

Table of contents

List of figures

List of tables

List of Abreviations

EHS – Environment, Health and Safety IROS –International Conference on Intelligent Robots and
CASE – Conference on Automation Science and Engineering
ICRA – International Conference Robotic and Automation
SMC – Sliding mode Control
PID – Proportional Integral Derivative
ICRA – IEEE Conference Robotic and Automation
DARPA–The Defense Advanced Research Projects Agency
NASA – National Aeronautics and Space Administration
DMC – Divergent Component of Motion
ZMP – Zero Moment Point
CoM – Center of Mass
LQR – Linear Quadratic Regulator
CHOMP – Covariant Hamiltonian Optimization for Motion Planning
ROS – Robot Operating System

1
Introduction

Robotics is a field of engineering that is gaining a lot of prominence today, with the growing demand of several sectors such as industrial, medical and agricultural for autonomy, safety and sustainability. In addition, along with technological advances over the years, these sectors are looking for ways to modernize in order to reduce costs, increase production and improve EHS conditions [5–7].One of the advantages of using robots is their application in regions where human presence for long periods is not viable, such as radioactive areas, locations with adverse climatic conditions and in space exploration [8,9]. Currently, there is also a tendency to use robots in hospitals to perform minimally invasive surgeries, [10] and smart farming systems to carry out precision agriculture tasks, wherein efficiency and accuracy are essential requirements [11, 12].

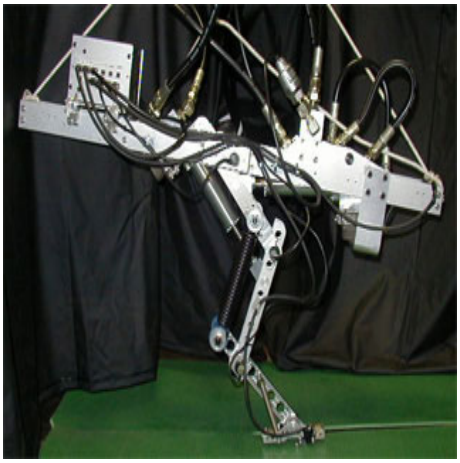
These technological advances and the demand for the use of robots have also led to the development of several control techniques for robot manipulators, such as visual servoing [13] and force control [14]. Another example is the use of robust control algorithms for wheeled robots and artificial intelligence, such as neural networks and reinforcement learning, for legged robots [15–17]. In this context, one of the research topics that has increased the interest of academic community in the area of field robotics is the use of humanoids and bio-inspired legged robots with four, six or eight legs, to carry out search and rescue missions in natural or man-made disaster areas [18, 19]. These complex mechanisms employ advanced control strategies and artificial intelligence algorithms to reproduce the locomotion abilities of humans and animals to stand, balance, walk, jump and run, enabling their use on uneven, slippery and obstacle-prone terrains [20]. There is no necessity for legged mobile robots to follow the desired leg trajectory. Sometimes the leg can not follow the desired leg trajectory because of, risk of collision between the legs and obstacles in the terrain. In this situation, an error in the leg trajectory avoids the collision with obstacles. The equilibrium condition of biped robots is that the ZMP is within the support polygon, and in the case of quadruped and hexapod robots, the CoM must be within the support polygon. In both cases, even if the legs do not follow the desired leg trajectories, if the equilibrium

conditions are satisfied, the robot will continue to move. This characteristic presented is an advantage that would justify its use in complex terrains.

1.1 Motivation

Before the invention of digital computers, the development of machines with legs was carried out employing electromechanical systems without any type of closed-loop control. This situation changed in the 1960s with the development of a digitally controlled robot that culminated in the late 1970s and during the 1980s in advances in the construction of these types of robots. In recent years has been an increase in legged mobile robots researches it occurs because of their potential to accomplish exploration task in harsh environments and difficult-to-access locations, which human beings and wheeled mobile robots are not capable access [21–23]. Their capably to avoid complex obstacles and to adapt the gait allow the legged robot execute task such as up the stairs, walking in slope terrains are the advantages over wheeled mobile robots.

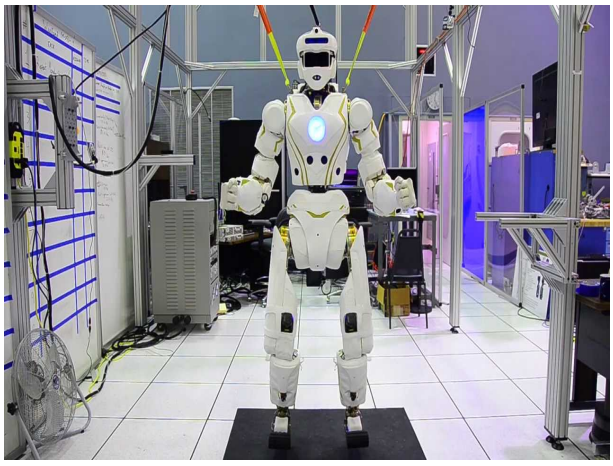
In this context there are many types of a legged mobile robot being used for research, the hopping robots such as Kenken [24], the Mowgli, a bipedal and jumping robot [25], the Handle from Boston Dynamics [26], the Bionic kangaroo from Festo [27] and, the two-wheeled jumping robot Ascento [28]. In the biped case, we have humanoids such as Darwin [29] and Walkyrie [30]. In the quadruped robot, we have HyQ [31] robot, the mini cheetah [32] and ANYmal robot [33]. The hexapod such as PhantomX is a well-known example of a hexapod.



1.1(a): The Kenken hopping robot [24]

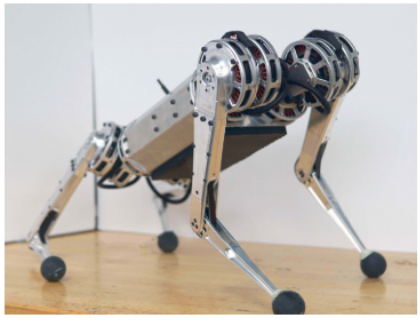
1.1(b): Kangaroo robot from Festo [27]

1.1(c): The DawinOP2

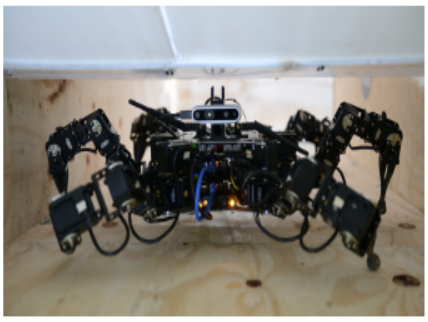


1.1(d): Walkyrie humanoid robot [30]

Figure 1.1: Legged mobile robots

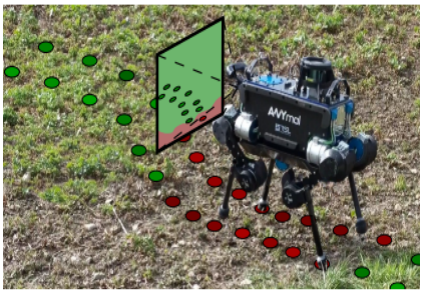


1.2(a): The MIT mini Cheetah robot [32]



1.2(b): The hexapod weaver [34]

1.2(c): Handle from Boston Dynamics [26]



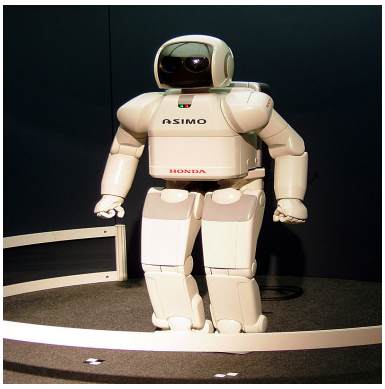
1.2(d): The ANYmal robot [33]

Figure 1.2: Legged mobile robots.

The USA government has been interesting about legged mobile robots to use in military applications, the American government finance researches in the legged mobile robots, the DARPA had a program dedicated to developing semi-autonomous legged mobile robots. The Rhex is a hexapod created by five universities (The University of Michigan, McGill University, Carnegie Mellon University, University of California, Princeton University, Cornell University and the University of Lahore) financed by DARPA. The Bigdog is an example of a quadruped robot financed by DARPA. It was created in 2005 by the Boston Dynamics with Foster-Miller, the NASA Jet Propulsion Laboratory, and the Harvard University Concord Field Station. The Bigdog is a pack-mule to serve to accompany soldiers in rough terrain where the vehicle can not access. Japan has prominence in a humanoid robot research an example is the Asimo produced by Honda the name is a tribute to the Russian writer Isaac Asimov. The Qrio created by Sony and present in the RoboCup in 2004 has another example of Japanese humanoid robot. The Actroid is a Japanese humanoid robot created by the Osaka University which has the human appearance and mimics functions such as speaking, blinking and breathing.

1.3(a): Big dog robot

1.3(b): The Rhex hexapod



1.3(c): The Asimo robot created by Honda

1.3(d): The Qrio created by Sony

1.3(e): The Actroid

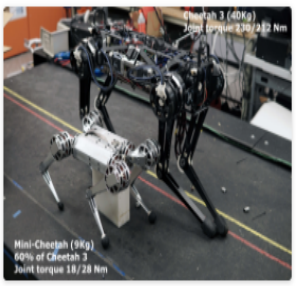
Figure 1.3: Legged mobile robots and applications (source: Wikipedia)

Legged mobile robots is a research theme which has been widely investigated in the last year and resulted in many publications in highly ranked robotics conferences such as ICRA, IROS and CASE. For instance, a full day workshop on legged robots was held in 2019 ICRA (Montreal, Canada) named Towards Real-World Deployment of Legged Robots, showing the scientific relevance of the such mechanisms. The following robot will run demos during the workshop: Spot Mini (Boston Dynamics), Laikago (Unitree), Mini Cheetah (MIT), ANYmal (ANYbotics) and GR Vision (Ghost Robotics).

1.4(a): Spot Mini from Boston Dynamics

1.4(b): Laikago from Unitree

1.4(c): Ghost robot GR Vision



1.4(d): MIT mini cheetah Unitree

1.4(e): ANY robotics ANYmal robot

Figure 1.4: Legged mobile robots [1]

1.2 Review of the State of the Art

Although the legged mobile robots have advantages, there are many challenges such as balance and design gait robot. The number of legs influence in the balance in the biped case, the stability condition is satisfied if the zero moment point is on the support polygon [35]. In the hexapod case, most stability criteria use some variation of Conservative Support Polygon [2]. In this context, another chalenge is to design a sequence of leg for each situation, terrain and task desired. In this context, legged robots are currently a topic of great interest in robotics, not only to reconcile balance and coordination of the legs but also to control the position and orientation of these robots in space and execute a task such as up the stairs, jumping the obstacles.

In Higa *et al.* [36], the authors propose to analyze how the articular position of the robot's leg affects the passivity condition and shows the importance of this aspect in the stability of the impedance controller. By analyzing the linear model of a robotic leg using its Nyquist graph and the Z width, the author can determine whether the joint configuration within the workspace is suitable for interacting with the environment or with people. In He *et al.* [37], the authors prose to use a backstepping control in a hopping robot in-flight phase, considering the non-holonomy in the joints configurations the authors transform the system in the chained form extended and apply the joint position control. The disadvantage is to use the extended chained form being the sliding mode control allow the same task using a simple chained form.

In Buchanan*et al.* [34], the authors present a deformable bounding box - which is an abstraction of a robot model - combined with mapping and planning strategies that enable the robot to change its shape and navigate in confined spaces. According to the paper, the mapping is achieved by using the robot-centric-multi-elevation maps generated with distance sensors carried by the robot. Finally, the authors propose the use of the CHOMP, an optimization algorithm which creates smooth trajectories while avoiding obstacles. The method is validated in the simulation and implemented in the hexapod weaver.

In Wellhausen*et al.* [33] proposes to collect data from robot interact terrain and associated with images. A neural network is trained using the sparse data acquired in teleoperation experiments the objective is to generate a prediction the terrain propriety. According to the paper, the training data is generated, projecting the foothold position from the robot trajectory into on-board camera images. The informations about the terrain are used in the autonomous navigation and the validation in the ANYmal robot.

In [38] the concepts about legged mobile robots, uses joints and spatial coordinates to represent the robot and describe the dynamic model of the robot. The dynamic model of the robot has the actuated part, represented by the joint coordinates and the underacted part, represented by the spatial coordinates. The book analyzes walking stability using forces and moments and a mobile leg robot and introduces the pressure centre in context.

In Caron*et al.* [39] proposes to control a humanoid robot to climb a stair, the humanoid robot is approximate a linear inverted pendulum and implemented two controllers: the DCM (Divergent Component of Motion) feedback control which seconds the author this control computes the desired wench to compensate the deviation walking. This part has the following inputs: the desired DCM and ZMP, the desired contacts forces and CoM, the estimated DCM and ZMP and the output controller is the distributed foot wrench. The second control is the whole-body admittance control this controller, allows the controlled-position robot to generate the desired contact wrench. The whole-body admittance control implements the feedback force control, the DCM output and the measured foot wrench and the desired kinematic target generates the robot's trajectory. The author validates the control implemented in an HRP4 humanoid robot.

In Klemm *et al.*, 2019 [28] the authors introduce the Ascent, a wheeled biped robot and assumes the following hypothesis: the link dynamics is neglected, there is no friction or hysteresis at the joints, the friction between the wheel and the floor is simplified, the motor dynamics are neglected, the time delay is not modeled, the links and the bodies are rigid. The authors consider a fixed leg geometric configuration and, model the Ascent robot as a two-wheeled inverted pendulum. Using the Lagrangean formulation the authors find the dynamic equation that describes the robot. The control strategy proposed in the paper for the Stabilizing Control is the LQR and the jump controller is the heuristic feed-forward controller based in the human jumping. The PID controller is used to control the leg extraction and the joint position in the flight phase. The author validates the controllers in simulation in GAZEBO and implements the controller in the real Ascento robot.

In Buhijara*et al.* [40], the authors present the difficulties to analyse the legged mobile robots, them present, the dynamic equation of legged mobile robot and split into two parts an actuated part and underactuated part. To deal with the locomotion problem the authors proposed the Alternating Direction Method of Multipliers, as claimed by the paper, the idea behind the method is to exploit the splitting between cost terms recursively, allowing to solver a problem simpler, than the original one. The algorithm presented assumes

the dependence on the upstream content planner to generate feasible contact planning. The algorithm is validated in the HRP2 humanoid.

In Rekleitis *et al.* [2], proposes to obtain the optimal leg sequence considering the external forces and the inclined terrain, to solve the problem the authors present the Leg Sequence Selection Algorithm, this algorithm has the following inputs: the external forces and the sloping terrain, the desired locomotion mode and the gait. The algorithm output is the best leg sequence if there is, the Leg Sequence Algorithm reconfigure the robot pose to ensure the stability and try again. To validate the algorithm, it implemented in the hexapod Hexaterra and Phantomx.

In Villarreal*et al.* [31] proposes a dynamic foothold using the visual feedback. The method uses to adjust the foot landing position using the only-board computers and sensors. A convolutional neural network is used to adapt the land position, the validation of the proposed method using simulations and implementation of the method in the HyQ quadruped robot.

In Semini *et al.* [41], the author wants to obtain quadruped robots using pneumatic actuators that have robust self-correction and aspects related to manipulation. They give an overview of the design of two new quadruped robots with hydraulic actuators: the HyQ2Max, an improved version of HyQ and, the centaur-style robot that combines the mobility platform of the HyQ2Max with a new hydraulic manipulator arm. The authors focus on the concept of the design of the mechanism of a new compact hydraulic arm.

In Chen *et al.* [42] presents the challenges of locomotion in hostile environments such as rugged locations and with many rocks, thus justifying the motivation for using robots with legs instead of robots with wheels. The authors propose to control the posture of a six-legged robot using a nonsingular fast terminal sliding mode (NFTSM). NFTSM is a control technique that uses a continuous sliding surface based on position error. The authors prove the stability and robustness using Lyapunov theory, verification the control technique through simulations and its validation through experiments on the quadrupedal robot ZJU Walker.

In Gehring *et al.* [43] the author proposes to control a quadruped robot. Through a model analogous to that of the inverted pendulum, the desired locations for the position of the legs are discovered and through a PD controller, it is used to control the movement of the legs and forces in order to control the position and orientation of the body. There is an external mesh capable of controlling horizontal, vertical and angular speeds, with this, the author simulates the robustness of the control when walking in complex terrain. Control validation takes place through experiments on a StarlETH robot

(Springy Tetrapod with Articulated Robotic Legs).

In Sakaino *et al.* [44], the authors have developed an SMC approach based on position control for a hopping robot, ensuring contact stability and motion accuracy even in the presence of high impact forces. The proposed solution leads to a straightforward controller design and an easier tuning of the feedback gains, in contrast to the variable compliance control.

In Liou *et al.* [45] the authors have designed a single-legged hopping robot with a pneumatic cylinder. Based on the energy analysis, the hopping cycle and height for the vertical hopping are evaluated. Then, a second-order SMC-based approach is used to control the mass flow rate for upper and lower chambers of the cylinder, regulating the hopping height under matched disturbances.

In Thomas *et al.* [46] the author proposes to control non-holonomic systems in chain form using Super-Twisting Algorithm combined with Finite-Time Control. The author's idea is to divide the system into two subsystems and control each subsystem separately and then, by means of a switch, a change in the control law takes place. The car only performed the stability test, thus lacking detailed proof of the robustness to disturbances or noise. Also, a curiosity is that the relationship between switching time T and system stability has not been rigorously tested. The verification of the controller takes place through numerical simulations with a differential drive.

In Murray *et al.* [47] the author proposes the use of sinusoidal to perform the control of non-holonomic systems, one of the examples presented by the author is a hopping robot with a prismatic joint in the flight phase. The advantage of this method is its simplicity but, its limitation is the amplitude of the input signal since the model of the robot has high order terms that generally harmonics that deteriorate the functioning of the robot.The author varificate the method by using numerical simulations.

In Abbasi *et al.* [48] the author uses super-twisting to control non-holonomic systems. The method consists of dividing the system into two subsystems and then carrying out the control action of each subsystem separately using two discontinuous control laws. The author performs numerical simulations with several non-holonomic systems such as differential drive and bicycles, for instance.

Table 1.1: Overview of the Review of the State-of-the Art.

Author	Robotic system	Modelling	Challenges	Control Strategy	Robotic Task	Verification and Validation
Higa *et al.*, 2019 [36]	Leg of a Quadruped Robot	Line	Impedance Control	Human-robot Interaction	Numerical Simulation	
He *et al.* [37]	Hopping robot	Dynamic	Control a non-holonomic system	Backstepping	Stabilization	Numerical simulation
Buchanan*et al.*,2019 [34]	hexapod robot	Free-modelling	Avoid obstacles		Tracking	Simulations and implementation in a real robot
Wellhausen*et al.* [33]	Quadruped robot	Free-modelling	To deal with the proprieties terrain	Neural network	Autonomous navigation	Implmentation in a real robot
Handbook of Robotic [38]	Legged mobile robots	Dynamic	Balance and walking			
Caron*et al.*, 2019 [39]	Humanoid robot	Linear inverted pendulum, Dynamic	Walking and balance	DCM and whole-body admittance control	Climbing a stair	Implementate the control in a HRP2 humanoid
Klemm *et al.*, 2019 [28]	Biped wheeled robot	Two-Wheeled Inverted pendulum Dynamic	Balance and walking	LQR and PID	Walking and jumping obstacles	Simulation in GaZEBO and implementation in a real robot
Buhijara*et al.* [40]	Humanoid robot	Dynamic	Solve a system with many DOFs	Alternating Direction Method of Multipliers	Robot locomotion	Implementation in a HRP-2
Rekleitis *et .al* [2]	Hexapod robot	Free-modelling	To deal with external forces and slope terrain	Leg Sequence Selection Algorithm	Select the gait and walking	Implementation algorithm in in hexapod robots
Villareal *et al.* [31]	Quadruped robot	Free-modelling	To meet a fast response for control action in dynamic locomotion	Convolution neural Network	Adjust the foot landing position	Implementation in the HyQ robot
Semini *et al.* [31]	Quadruped robot	Free-modelling	Create a robust quadruped robot with a hydraulic actuator		Walking in rough and slope terrain	Implementation in the HyQ and HyQ2Max robot
Chen *et al.* [42]	Quadruped robot	Kinematic	Balance and the complex inverse kinematic	Nonsingular fast terminal slidingmode	Control de posture	Implementation in ZJU Walker robot
Gehring *et al.* [43]	Quadruped robot	Dynamic	Balance and the complex inverse kinematic	PD controller	Control de posture	Implementation in a StarIETH robot
Sakaino *et al.* [43]	Hopping robot	Dynamic and Kinematic	Balance and stability	Sliding Mode Controlr	Tracking a specific trajectory	Implementation in a hopping robot
Liou *et al.* [45]	Hopping robot	Dynamic	Balance and stability	Second- Order Sliding Mode Control	Tracking a specific trajectory	Implementation in a hopping robot
Thomas *et al.* [46]	Wheeled robot	Kinematic	Deal with nonholomic constraint	Super-Twisting andr Finite-Time Control	Regulation posture	Numerical simulations
Murray *et al.* [47]	Hopping robot	Kinematic	Deal with nonholomic constraint	Planing (Fourier techiniquies)	Regulation posture	Numerical simulations
Abbasi *et al.* [48]	Whelled robot	Kinematic	Deal with nonholomic constraint	Super-Twisting Algorithm	Regulation posture	Numerical simulations

1.3 Methodology

First, the hopping robot model with a prismatic joint will be developed. Then we will use planning to direct the robot joints to the desired position. We will simulate the Matlab for the simplified model and the complete model and, these results will be commented on and analysed. Then, mathematical will be developed to prove the stability and robustness of three robust control algorithms: First-Order SMC, STA + SMC and STA + FTA. The numerical simulations should be performed in Matlab considering high order terms, unmodeled dynamics and measurement noise. We will calculate and analyse the errors metrics (RMSE) and control signals (MAD). We will adopt the same methodology for an articulated hopping robot in the flight phase. Finally, to finish the study of the hopping robot, the hopping robot model will be developed in the stance phase. Then, two controllers will be proposed: the first is the computed torque control and the second and the SMC with robustness.

After the study on the hopping robot, studies on hexapod robots will be carried out. The first step will be developed, a cascade structure to control the hexapod position and orientation.

Then, it will be designing the gait so that the robot can move around considering the conditions of balance and speed of the robot.

Next, a robot approach as a differential drive will be proposed, then two types of controllers for the robot, a Cartesian and another SMC, will be designed. After the development of the controllers, graphical simulations will be carried out in the Gazebo and comment and analyse the results.

1.4 Contribution

The contribution of this master's thesis is control techniques for nonholonomic systems. These two methods consist of a first-order slider control without chattering. The second controller is the combination of a conventional slider control with control by second-order sliding modes. For this, theorems have been proposed that provide stability and robustness to drivers in the shape of the chain. The nonholonomic system that will be used for a driver analysis is a jumping robot in the flight phase. We propose to control the position and orientation of a hexapod robot by using a differential drive approach and the SMC.

1.5 Goals and Objectives

The objectives of the thesis are to make a comparative study between controls for a hopping robot in the flight phase.

Perform a comparison between classic control techniques such as computed torque and SMC.

Control the position of a hexapod robot Phantomx using a Cartesian controller and control the posture of the same robot using SMC.

1.6 Organization of the Thesis

The documentation of this Master thesis is organized according to the following chapters:

- In chapter 2, the hopping robot model with a prismatic joint in the flight phase will be presented and, we will use planning by Fourier techniques for steering. Next, we propose a comparative study between robust controllers for hopping robot control. Then, the same comparative study is applied in an articulated hopping robot in the flight phase. Finally, we will compare the computed torque and SMC for hopping robot control in the stance phase and demonstrate the advantages and disadvantages of the SMC controller.
- In Chapter 3, we will introduce a cascade control strategy for a hexapod position and orientation. Then, we will use the differential drive approach and propose the Cartesian controller for position control and the SMC for position and orientation control. Then, we will simulate the controllers in Gazebo and comments and analysis the results.
- In Chapter 4, the author's general analysis of the chapters and the presentation of future works and new challenges to be overcome in the study of legged robots will be presented.
- In the Appendix, we will present the statistics on robots with legs, authors and institutions that work with this type of robot, possible applications in agriculture. It will also be presented basic concepts about the theory of control by sliding modes implementations carried out in ROS as nodes and topics used.

2
Hopping Robot: Modelling and Control Design

A hopping robot consists of a simple one-legged robot that moves through jumping. Its locomotion mode imitates animals such as kangaroo and, the simplicity in the hopping robot model make it, a start point to study the modelling, control design and motion planning of legged robot. In this context, there are many types of hopping robots have been used for research, for instance, the biologically-inspired robot based on the kangaroo walking such as Kenken [24],the Mowgli, a bipedal and jumping robot [25], and the Bionic Kangaroo robot from Festo [27].Another type of hopping robot widely investigated is the robot with a prismatic joint on the leg such as the simplified hopping robot [47] and the Ascent, a two-wheeled jumping robot [28].

The spring-loaded inverted pendulum (SLIP) shown in Fig. 2.1 has emerged as a simple model for studying dynamic locomotion of single-legged mobile robots. The leg consists of a massless spring connected to a body which represents the centre of mass of the walking mechanism. In the stance phase, the spring is compressed and decompressed describing a given trajectory and, in the flight phase the gravity is the only acceleration acting in the body and, the centre of mass describes a ballistic trajectory [49].

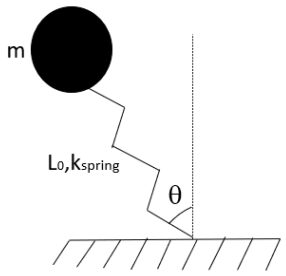


Figure 2.1: Representation of a Spring Loaded Inverted Pendulum.

The kinematic modelling of the hopping robot can be divided into two stages: in the flight phase, the robot is flying and does not touch the ground; in the stance phase, the robot is standing and in contact with the ground. In the flight phase, all joint motions satisfy the angular momentum conservation and, the system dynamics can be expressed in the chained form [47]. In the stance phase, the contact force can be used to model the interaction between

the robot and the ground and the Lagrange's equations are used to describe the system dynamics.

The challenge in the flight phase is to regulate the robot posture when the mechanism touches the grounds to prevent its fall. On the other hand, in the stance phase, the challenge is to ensure the robot balance and satisfactory performance during the tracking of specific joints trajectories. In this context, since legged mobile robots are inherently unstable systems, several control strategies and artificial intelligence algorithms have been used to solve the balance issues during locomotion.

The fuzzy logic control is a possible technique which to control the robot in flight and stance phases [50] it has the advantages that is not necessary information about the model, the disadvantages is the accuracy increase the number of rules, for instance disadvantage is an exponential increase in the number of rules as the accuracy of the controller increases. The artificial intelligence such as neural network [51] makes possible ensure the hopping robot desired trajectory, on the other hand, this technique have disadvantages such as the weight adjust of the neural network and the data available.

In this chapter, we proposed to control planar hopping robot in-flight phase using Sliding Mode Control and Super Twisting Algorithm with Finite-Time Control and control the joint position of a hopping robot in stance phase using robust control.

2.1 Modeling and Control Design of a Prismatic Hopping Robot

The hopping robot consists of a rigid body attached to an actuated leg which can rotate and extend. The robot configuration is given by the triplet $q=[\psi \;\; L \;\; \theta]^{\mathsf{T}}$ where ψ is the leg angle, L is the leg extension, and θ is the body angle of the robot (Fig. 2.2). The moment of inertia of the body is denoted by I and we assume that the leg mass m is concentrated at the foot. The upper leg length is denoted by d with L being interpreted as a prismatic joint which represents the extension of the leg past this point.

When the hopping robot is in the flight phase (e.g., free floating), the law of conservation of angular momentum implies that moving the legs causes the central body to rotate [47]. In the case that the angular momentum is zero, such a conservation law can be interpreted as a nonholonomic constraint on the hopping robot [47]. In this case, the conservation of the angular momentum give us the following constraint equation:

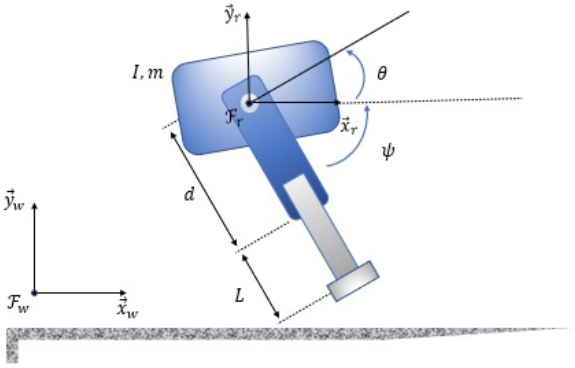


Figure 2.2: A simple hopping robot, its frames and configuration parameters.

$$\sum_{i=1}^{N} \frac{d\mu_i}{dt} = \mathcal{I}\,\dot{\omega} = 0\,, \tag{2-1}$$

where $\dot{\omega} \in \mathbb{R}$ is the angular acceleration of the body, $\mathcal{I}$ is the moment of inertia and $\mu_i \in \mathbb{R}$ is the i-th angular momentum. Therefore, the angular momentum of the hopping robot in flight phase can be given by:

$$I\,\dot{\theta} + m(L+d)^2(\dot{\theta}+\dot{\psi}) = 0\,. \tag{2-2}$$

As in the hopping robot it is possible to command the leg angle and the leg extension directly, from the kinematic control approach we can choose their velocities as the system inputs. Then, we set $\dot{\psi} = u_1$ and $\dot{L} = u_2$ to obtain:

$$\begin{aligned} \dot{\psi} &= u_1\,, \\ \dot{L} &= u_2\,, \\ \dot{\theta} &= \frac{-m(L+d)^2}{I + m(L+d)^2}\,u_1\,. \end{aligned} \tag{2-3}$$

Notice that, from Eq. (2-3), the rotational dynamics of the body angle θ is highly nonlinear and depends on the leg extension L. Then, to find a linear approximation for $\dot{\theta}$, we can expand the right-hand side of the last term of Eq. (2-3) using a Taylor series about $L = L^*$, obtain:

$$\dot{\theta} = -k_\psi \dot{\psi} - k_u\,(L - L^*)\,u_1 + f(L)\,, \tag{2-4}$$

with

$$k_\psi = \frac{m(L^*+d)^2}{I + m(L^*+d)^2}\,, \qquad k_u = \frac{2m(L^*+d)I}{(I + m(L^*+d)^2)^2}\,,$$

and $f(L) = O(L^2)\,u_1$ in the Eq. (2-4) denotes the quadratic and high-order terms in L, which can be neglected if L is close to L^*, the term $O(L^2)$ is a

notation for high order term adopted in [47]. To transform the system in the chained form we use the following relationship:

$$\alpha = \theta + k_\psi \, \psi \, . \tag{2-5}$$

Here, without loss of generality, we assume that $L^* = 0$. Therefore, the motion dynamics of the hopping robot system expressed in the chained form is given by:

$$\begin{aligned} \dot{\psi} &= u_1 \, . \\ \dot{L} &= u_2 \, . \\ \dot{\alpha} &= -k_u \, L \, u_1 + \ f(L) \, . \end{aligned} \tag{2-6}$$

Notice that, Eq. (2-2) is a single non-holonomic constraint [47] expressed in terms of the angular velocities $\dot{\psi}$ and $\dot{\theta}$. Thus, the corresponding control system in Eq. (2-6) has two inputs and three configuration variables minus one constraint.

2.2 Planning Algorithm using Fourier Techniques

The hopping robot in the flight phase is a nonholonomic system described by the equation Eq. (2-6), notice that the variable $\dot{\theta}$ difficult the analysis of problem. A solution to simplify the analysis of problem is to represent $\dot{\theta}$ in a Taylor series expansion and, neglect the high order terms which result in:

$$\dot{\psi} = u_1 \, , \tag{2-7}$$

$$\dot{L} = u_2 \, , \tag{2-8}$$

$$\dot{\alpha} = -k_u L \, u_1 \, .$$

The equation Eq. (2-7) is in chained form, in [47] the authors, propose an algorithm to deal with kind of system. The algorithm consists in steer ψ and L to desired values. Then, the following sinusoids are used:

$$u_1 = a_1 \sin(\omega t) \, , \tag{2-9}$$

$$u_2 = a_2 \cos(\omega t) \, . \tag{2-10}$$

By choice, the period used is 1 second, and the last motion does not affect the final values of variables ψ and L. Since $L = (a_2/2\pi)\sin(\omega \, t)$, the high order

terms $f(L)$ can be expanded in a Fourier series:

$$f\left(\frac{a_2}{2\pi} sin(2\,\pi\,t)\right) = \beta_1\,\sin(2\,\pi\,t) + \beta_1\,\sin(4\,\pi\,t) + ... \tag{2-11}$$

Integrating $\dot{\alpha}$ over one period and noting that only the component in fundamental frequency contributes to the net motion

$$\alpha(1) - \alpha(0) = \int_0^1 (\beta_1\,\sin^2(2\,\pi\,t) + \beta_1\,\sin^2(4\,\pi\,t) + ...)dt \tag{2-12}$$

$$= \frac{1}{2} a_1 \beta_1\,. \tag{2-13}$$

where β_1 is a function of a_2, which can be solved numerically for values of a_1 and a_2 to achieve the net change in α.

Considering Eq 2-7, α can solver analytically, using the following expression:

$$\alpha = \alpha\,(0) - k_u\,a_1\,a_2 \int_0^1 \sin^2\omega t\,dt\,, \tag{2-14}$$

$$\alpha = \alpha\,(0) - \pi n \frac{k_u\,a_1\,a_2}{\omega}\,. \tag{2-15}$$

The equation Eq 2-14 allows calculating the angle α and hence θ, the problem in the approximation is the harmonics are not considering.

2.2.1 Simulations and results

In this section is simulated the method proposed in [47], the objective is to check where the method development fails. The parameters used in the simulation are shown in table 2.1:

Table 2.1: Parameters simulations of hopping robot

Parameters	Value	Unit
Mass	1.0	kg
Momentum of Inertia	1.0	$kg\,m^2$
Distance leg (d)	1.0	m
Initial leg angle (ψ_i)	0	rad
Desired leg angle (ψ_d)	0	rad
Initial body angle (θ_i)	0	rad
Desired body angle (θ_d)	–	rad
Initial leg length (L_i)	0	m
Desired leg length (L_d)	0	m
a_1	1	–
a_2	1	–
Input Frequency (ω)	1	$rad\,s^{-1}$

In the first place, we check the standard deviation, the variance and the mean for the angle θ do not consider the non-linearity in θ. We variate the amplitudes a_1 and a_2 and, check the error in each situation the result are in the table 2.2:

Table 2.2: Simulation results neglecting the harmonic effect

Amplitude a_1	Amplitude a_2	e_θ
0.5	1.0	$-2.585\,10^{-5}$
0.6	1.2	$-0.3712\,10^{-5}$
0.8	1.5	$-0.6201\,10^{-5}$
1.0	1.0	$-0.5168\,10^{-5}$
1.2	0.5	$-0.3101\,10^{-5}$
0.7	0.4	$-0.1447\,10^{-5}$
0.3	0.5	$-0.7752\,10^{-5}$
1.2	1.5	$-0.9302\,10^{-5}$
1.5	1.0	$-0.7752\,10^{-5}$
0.4	0.4	$-0.8268\,10^{-5}$
Standard Deviation	Mean	Variance
$7.1624\,10^{-06}$	$-7.1575\,10^{-06}$	$5.130\,10^{-11}$

Analysing the table 2.3 notice that, the standard deviation is close to zero, it means that the error does not disperse about the mean. The method has limitations in the use because, the amplitude of ψ and L influence the system behaviour, for the low amplitudes the non-linearity does not affect much the performance and the approximation proposed to α can be used.

Using the non-linearity in the analyse the standard deviation is far to the mean, it means that the approximation fails in this case, to low amplitudes the error is close to zero because the harmonics are not much excited by the control applied. The result can see the table 2.2.

Table 2.3: Simulation results considering the harmonic effect

Amplitude a_1	Amplitude a_2	e_θ
0.5	1.0	-0.15551
0.6	1.2	-0.2228
0.8	1.5	-0.3699
1.0	1.0	-0.3102
1.2	0.5	-0.1873
0.7	0.4	-0.0875
0.3	0.5	-0.0468
1.2	1.5	-0.5548
1.5	1.0	-0.4623
0.4	0.4	-0.050
Standard Deviation	Mean	Variance
0.1767	-0.2434	0.0312

Notice in Fig.2.4a-b and Fig.2.4a-b the state ψ and L does not change at the end of the input signal cycle. The graphics in Fig. 2.3.c and Fig. 2.4.c show the difference between the simplified model (neglect high order terms) and complete model(considering the high order terms), in Fig.2.3.$c\theta$ assume value close to desired. In Fig 2.4.c the error in θ increase, it is proved that the method has limitations. In both cases, we consider the same control signal in Fig . 2.5(a)(b) the control signal is bounded.

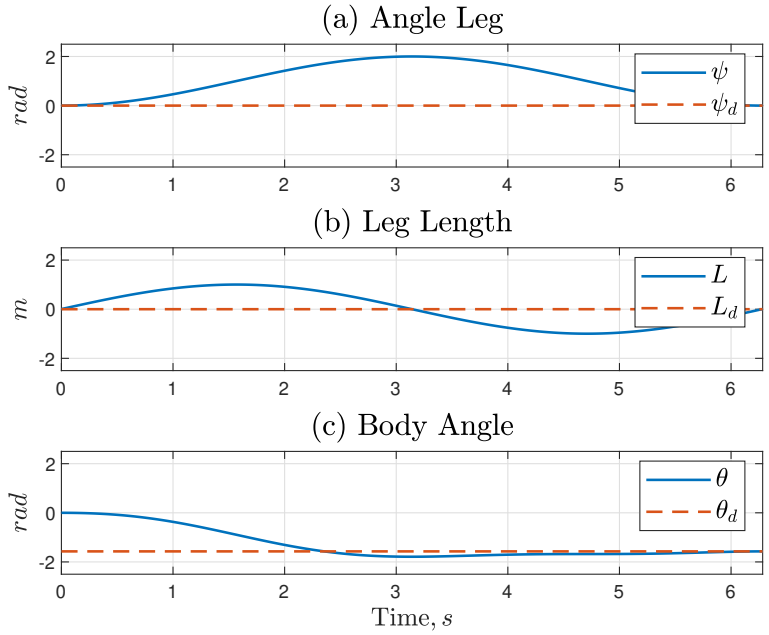


Figure 2.3: Motion planing of hopping robot to approximate model:(a) Leg Angle, (b) Leg Length, (c) Angle Body

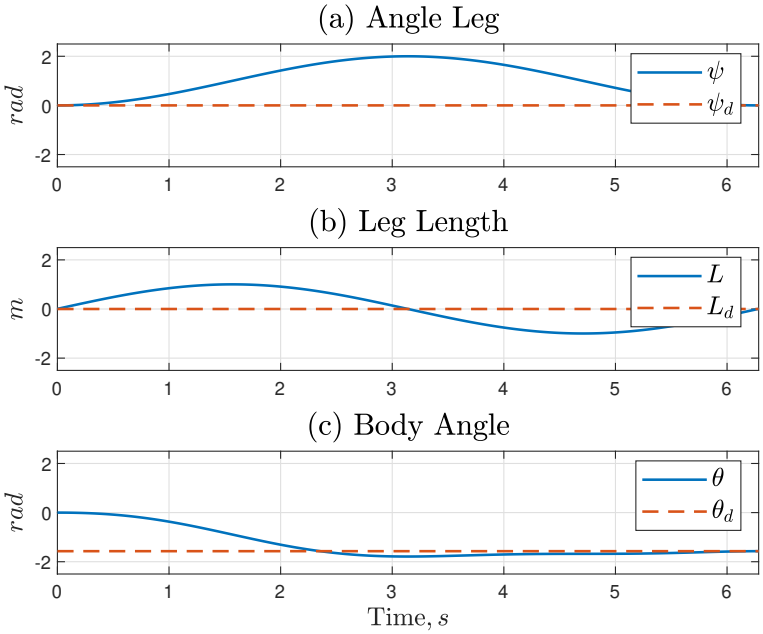


Figure 2.4: Motion planing of hopping robot to considering the high order terms in the model: (a) Leg Angle, (b) Leg Length, (c) Angle Body

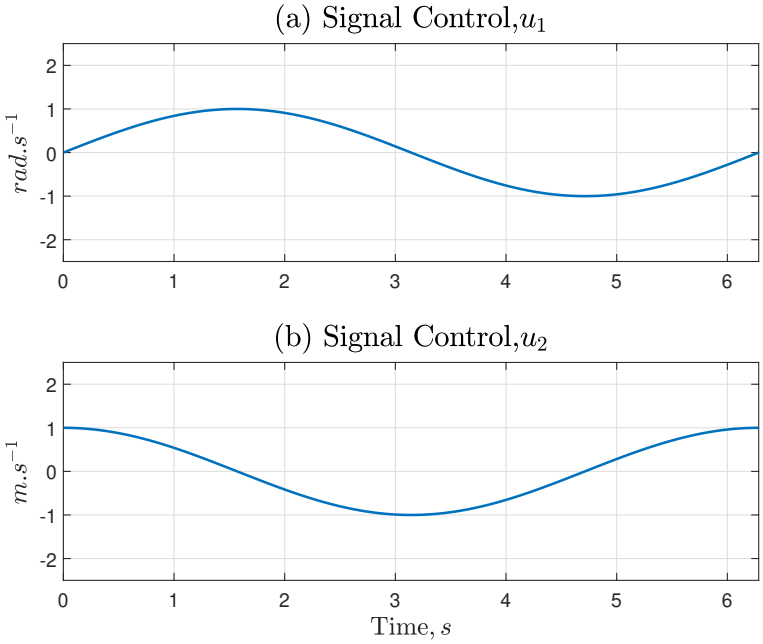


Figure 2.5: Control Signal Proposed for the Hopping Robot:(a) Angular velocity, u_1, (b) Linear Velocity, u_2

2.2.2 Robust Control Design

Consider the stabilization problem of a hopping robot in a flight phase. In this case, the control goal is to reorient the body of robot while its is in free floating and bring the current leg rotation and extension (ψ, L) to a constants desired final values, denoted by (ψ_d, L_d). Then, the control goal can be simply described by:

$$\psi \to \psi_d\,, \qquad e_\psi = \psi - \psi_d \to 0\,, \tag{2-16}$$

$$L \to L_d\,, \qquad e_L = L - L_d \to 0\,, \tag{2-17}$$

where $e_\psi \in \mathbb{R}$ and $e_L \in \mathbb{R}$ are respectively the errors in the leg rotation and extension. Defining $e_\alpha := \alpha - \alpha_d$ where, α_d is a know value and taking the first time-derivative of such errors we obtain:

$$\begin{aligned} \dot{e}_\psi &= u_1\,. \\ \dot{e}_L &= u_2\,. \\ \dot{e}_\alpha &= -k_u\, e_L u_1 + f(L)\,. \end{aligned} \tag{2-18}$$

where $u_1 \in \mathbb{R}$ and $u_2 \in \mathbb{R}$ are the velocity control signals to be designed. In the next section, we derive three controllers for a one-legged hopping robot in flight phase subject to unmatched perturbations. We use three algorithms: Algorithm I (First-Order SMC), Algorithm II (STA+SMC) and Algorithm III (STA+FTC).

2.2.3 Algorithm I: First-order SMC

Here the key idea is to design a robust control algorithm to steer the full nonlinear system in the presence of external disturbances caused by the quadratic and high order terms $f(L, u_1)$. Once we control the ψ and L states directly, we first steer such states to their desired values by using the following control laws:

$$u_1 = -e_\psi + \gamma\, e_L\, |s|^p \operatorname{sgn}(s)\,, \tag{2-19.1}$$

$$u_2 = -e_L - \gamma\, e_\psi\, |s|^p \operatorname{sgn}(s)\,, \tag{2-19.2}$$

where $p \in \mathbb{R}$ is a constant parameter such that $0 < p < 1$, $\gamma > 0$ is the control gain, and $s \in \mathbb{R}$ is the proposed sliding surface given by:

$$s = -2k_u^{-1}\, e_\alpha - e_\psi\, e_L\,, \tag{2-20}$$

which ensures that we can steer α to its desired value. The sliding surface s is designed based on studies on first-order SMC approach of non-holonomic systems described in the chained form.

It is worth mentioning that motion planning algorithms, such as Fourier techniques, could also be used for steering a class of controllable nonholonomic systems, which can be represented by the canonical chained form. However, when sinusoidal inputs are used for steering, the existence of any additional nonlinearities in the structure of chained systems may cause perturbations (e.g., zero frequency components) in their previous coordinates which can lead to a performance degradation for regulation and tracking tasks. This motivates us to design a robust control algorithm for stabilizing a class of nonholonomic systems represented in the canonical chained form subject to external disturbances. Then, we can state the following theorem to stabilish the stability and robustness properties of the proposed first-order SMC-based controller:

Teorema 2.1 *Consider a nonholonomic system described by Eq. (2-18) and the stabilizing control laws given by Eq. (2-19.1) and Eq. (2-19.2), with the sliding surface defined by Eq. (2-20). Assume that the desired values* (ψ_d, L_d) *and the perturbation term* $f(L, u_1)$ *are uniformly bounded. Then, the following stability properties hold: (i) all signals of the overall closed-loop system are bounded; (ii)* $\lim_{t\to\infty} e_\psi(t), e_L(t) = 0$*; (iii)* $\lim_{t\to\infty} s(t) = 0$ *and, consequently,* $\lim_{t\to\infty} e_\alpha(t) = 0$.

Prova. The asymptotic stability of the errors e_ψ and e_L can be demonstrated by using the following Lyapunov-like function: $2V_r(e_\psi, e_L) = e_\psi^2 + e_L^2$, which is only positive semi-definite at the origin, because it is zero in all configurations such that $e_\psi = e_L = 0$, regardless of the value of the angle e_α. Taking the time-derivative of V_r, substituting Eq. (2-18) and using the control laws given by Eq. (2-19.1) and Eq. (2-19.2) we have: $\dot{V}_r(e_\psi, e_L) = -(e_\psi^2 + e_L^2) \leq 0$, which is semi-definite negative at the origin. This means that $V_r \in \mathcal{L}_\infty$ and, consequently, $e_\psi \in \mathcal{L}_\infty$ and $e_L \in \mathcal{L}_\infty$. Therefore, the error state vector (e_ψ, e_L) converges to a manifold $\Omega = \{e \in \mathbb{R}^3 \mid e_\psi = e_L = 0\}$. Now, let us use the sliding mode control to ensure that e_α converges to zero. Then, taking the time-derivative of s from

Eq. (2-20) as:

$$\dot{s} = -e_L u_1 + e_\psi u_2 + 2K_u^{-1} f(L, u_1)\,, \tag{2-21}$$

and replacing Eq. (2-19.1) and Eq. (2-19.2) into Eq. (2-21) yields:

$$\dot{s} = -\gamma(e_L^2 + e_\psi^2)\,|s|^p\,\mathrm{sgn}(s) + 2k_u^{-1}\,f(L, u_1)\,. \tag{2-22}$$

The next step is to show that the sliding condition $s\dot{s} \leq -\beta|s|$ holds, for some $\beta > 0$. Thus, we choose the following Lyapunov candidate function $2V_s(s) = s^2$, and taking its time-derivative yields:

$$\begin{aligned}\dot{V}_s(s) = &-\gamma(e_L^2 + e_\psi^2)\,|s|^{p+1} \\ &- 2k_u^{-1}\,O(L^2)\,[\,e_\psi s - e_L\,|s|^{p+1}\,]\,.\end{aligned} \tag{2-23}$$

We have shown that e_ψ and e_L converge to zero as $f(L, u_1)$ is a bounded function. Then, we can conclude that the terms $O(L^2)e_\psi$ and $O(L^2)e_L$ in Eq. (2-23) vanish. Therefore, $\dot{V}_s(s)$ becomes negative definite which implies that $\dot{V}_s, s \in \mathcal{L}_\infty$ and sliding condition holds. In this case, we conclude that $s \to 0$ and, consequently, $e_\alpha \to 0$. ■

2.2.4
Algorithm II: SMC + STA

Although conventional first-order SMC approaches can guarantee stability and robustness properties under external disturbances, these controllers are subject to chattering phenomenon which can deteriorate the performance of the system response. However, our proposed first-order SMC approach is capable of reducing or eliminating the vibration effects caused by non-ideal switching. The high-frequency components can also be attenuated by using the so-called Super-twisting algorithm (STA) approach [48, 52], a second-order sliding mode approach. Here, we consider the hierarchical control approach designing two discontinuous control laws to firstly stabilize the errors e_L and e_α, and secondly the error e_ψ. Then, we first design the discontinuous control law u_2 based on the first-order SMC approach as:

$$u_2 = \begin{cases} 0\,, & t \geq T\,, \\ -\lambda^{-1}(k_u e_L - k_1\,|s|^\rho\,\mathrm{sgn}(s))\,, & T_0 < t < T\,, \end{cases} \tag{2-24}$$

where $k_1 > 0$ is the proportional gain and $s \in \mathbb{R}$ is the sliding surface given by:

$$s = -e_\alpha + \lambda\, e_L\,, \tag{2-25}$$

with $\lambda \in \mathbb{R}$ being the slope of the sliding surface.

Now, let us design the discontinuous control law u_1 based on the STA approach as:

$$u_1 = \begin{cases} -k_2\,|e_\psi|^\rho \operatorname{sgn}(e_\psi) + z\,, & t \geq T\,, \\ 1\,, & T_0 < t < T\,, \end{cases} \tag{2-26}$$

with $\dot{z} = -k_3\,|e_\psi|^\rho \operatorname{sgn}(e_\psi)$, where $k_2 > 0$ and $k_3 > 0$ are the proportional gains, $\rho \in (0, 1/2]$ is a constant parameter, which attenuates the chattering phenomena, $T \in \mathbb{R}^+$ is the switching time, which is responsible to switch the control laws, given by Eq. (2-24) and Eq. (2-26), and $T_0 \geq 0$ is the initial time. It is worth noticing that, the switching time T can be chosen empirically according to the value of maximum error norm for e_α (around 0.08) and robot flight time T_f (less than $2\,s$), defined after exhaustive numerical simulations.

Then, we can state the following theorem to stabilish the stability and robustness properties of the proposed SMC plus STA based controller:

Teorema 2.2 *Consider a nonholonomic system described by Eq. (2-18), the stabilizing control laws given by Eq. (2-24) and (2-26) with the sliding surface defined by Eq. (2-25). Assume that the desired values (ψ_d, L_d) and the perturbation term $f(L, u_1)$ are uniformly bounded, and the switching time is $T \geq 0$. Then, the following stability properties hold: (i) all signals of the overall closed-loop system are bounded; (ii) $\lim_{t\to T} e_L(t)$, $\lim_{t\to T} e_\alpha(t) = 0$ and, consequently, $\lim_{t\to T} s(t) = 0$; (iii) $\lim_{t\to\infty} e_\psi(t) = 0$.*

Prova. The first step is to show that the error subsystem $\mathcal{S}_1 : \{e_L, e_\alpha\}$ converges to zero. Thus, consider the following Lyapunov candidate function: $2V_1(s) = s^2$. Computing its time-derivative, replacing the time-derivative of the sliding surface given by Eq. (2-25) and using Eq. (2-18) yields: $\dot{V}_1(s) = s(k_u e_L u_1 - f(L, u_1) + u_2)$. Then, when $T_0 \leq t \leq T$ we have $u_1(t) = 1$ and substituting Eq. (2-24) into $\dot{V}_1(s)$, we have:

$$\dot{V}_1(s) = s\,[\,-f(L, u_1) - k_1|s|^\rho \operatorname{sgn}(s)\,]\,. \tag{2-27}$$

Now, we must choose k_1 such that $\dot{V}_1(s)$ is negative semi-definite, ensuring the sliding condition $s\,\dot{s} \leq -\beta|s|$ for some $\beta > 0$. Then, for $k_1 \geq |f(L, u_1)|$ implies that $s \to 0$ and, hence, $\dot{e}_L = -k_u e_L$. Therefore, we conclude that $e_L \to 0$ and, consequently, $e_\alpha \to 0$.

Now, the next step is to show the convergence of the error subsystem $\mathcal{S}_2 : \{e_\psi\}$ to zero, using a suitable Lyapunov-like function $V_2(e_\psi, z)$ given by:

$$V_2(e_\psi, z) = k_3 \frac{|e_\psi|^{\rho+1}}{\rho+1} + \frac{1}{2} z^2 \,, \tag{2-28}$$

Then, taking its time-derivative, we have:

$$\dot{V}_2(e_\psi, z) = k_3 \, |e_\psi|^{\rho-1} \, \dot{e}_\psi \, \mathrm{sgn}(e_\psi) + z\dot{z} \,. \tag{2-29}$$

When $t \geq T$, we have $u_2(t) = 0$ and the error subsystem $\mathcal{S}_1$ remains at its equilibrium. Substituting Eq. (2-26) into Eq. (2-29) yields:

$$\dot{V}_2(e_\psi, z) = -k_2 \, k_3 \, |e_\psi|^{2\rho} \,. \tag{2-30}$$

Therefore, $\dot{V}_2(e_\psi, z)$ is negative semi-definite, which implies that $e_\psi, z \in \mathcal{L}_\infty$ and, consequently, that $e_\psi \to 0$ and $z \to 0$. ■

2.2.5
Algorithm III: STA + FTC

In this section, we develop a robust control algorithm based on a suitable combination of STA approach and FTC technique for steering the hopping robot in its flight phase under unmatched perturbations. The control design is analogous to that one introduced in Thomas *et al.* [46] for posture stabilization of a unicycle mobile robot. Here, we consider the hierarchical control approach designing two discontinuous control laws to firstly stabilize the error e_ψ, and secondly the errors e_L and e_α. Then, we first design the discontinuous control law u_1 based on the STA approach as:

$$u_1 = \begin{cases} -k_4 |e_\psi|^\rho \mathrm{sgn}(e_\psi) + z \,, & t \geq T \,, \\ 1 \,, & T_0 < t < T \,, \end{cases} \tag{2-31}$$

with $\dot{z} = -k_5 \, \mathrm{sgn}(e_\psi)$, where $k_4 > 0$ and $k_5 > 0$ are the proportional gains. Now, let us design the discontinuous control law u_2 based on the FTC technique as:

$$u_2 = \begin{cases} 0 \,, & t \geq T \,, \\ k_u \, k_6 |e_\alpha|^{\alpha_1} \, \mathrm{sgn}(e_\alpha) + \kappa \,, & T_0 < t < T \,, \end{cases} \tag{2-32}$$

with $\kappa = -k_7 \, |e_L|^{\alpha_2} \, \mathrm{sgn}(e_L)$, where $k_6 > 0$ and $k_7 > 0$ are the proportional gains, and $\alpha_1, \alpha_2 \in (0, 1/2]$ are constant parameters that attenuate the chattering phenomena. Then, we can state the following theorem to establish the stability and robustness properties of the proposed STA plus FTC based controller:

Teorema 2.3 *Consider the nonholonomic system described by Eq. (2-18), the stabilizing control laws given by Eq. (2-31) and Eq. (2-32). Assume that the desired values* (ψ_d, L_d) *and the high order terms* $f(L, u_1)$ *are uniformly bounded, and the switching time is* $T \geq 0$*. Then, the following stability properties hold: (i) all signals of the overall closed-loop system are bounded; (ii)* $\lim_{t\to\infty} e_\psi(t)$*; (iii)* $\lim_{t\to T} e_L(t)=0$ *and, consequently,* $\lim_{t\to T} e_\alpha(t)=0$*.*

Prova. The first step is to show that the error subsystem $\mathcal{S}_1 : \{e_L, e_\alpha\}$ converges to zero. Thus, consider the following Lyapunov candidate function:

$$V_1(e_L, e_\alpha) = k_6 \frac{|e_\alpha|^{\alpha_1+1}}{\alpha_1+1} + \frac{1}{2}e_L^2 \,. \tag{2-33}$$

Then, taking its time-derivative, we have:

$$\dot{V}_1(e_L, e_\alpha) = k_6 \, |e_\alpha|^{\alpha_1} \, \dot{e}_\alpha \, \mathrm{sgn}(e_\alpha) + e_L \, \dot{e}_L \,. \tag{2-34}$$

Then, when $T_0 \leq t \leq T$ we have $u_1(t) = 1$, and substituting Eq. (2-32) into Eq. (2-34), we have: $\dot{V}_2(e_L, e_\alpha) = -k_7 \, |e_L|^{\alpha_2+1} + k_6 \, |e_\alpha|^{\alpha_1} \, \mathrm{sgn}(e_\alpha) \, f(L, u_1)$. Notice that, the second term in the left-hand side of $\dot{V}_1$ has indefinite sign. Then, we can show that the error subsystem $\mathcal{S}_1$ converges to zero only without the perturbation term $f(L, u_1)$. Indeed, as shown in [53], $\dot{V}_1$ is a weak Lyapunov function and finite time convergence can be proven only by using a generalization of LaSalle's invariance principle for discontinous systems. Moreover, it is not possible to demonstrate robustness properties, as shown in [46]. However, as proven in [53], the following Lyapunov function $V_1(\zeta) = \zeta^\mathsf{T} P \zeta$ where $\zeta = [\, |e_\alpha|^{1/2} \mathrm{sgn}(e_\alpha) \;\; e_L \,]^\mathsf{T}$ with $P = P^\mathsf{T} > 0$ can be used to shown that $e_L \to 0$ and $e_\alpha \to 0$ in finite time. This demonstration will be omitted here for the sake of space-saving. Now, the next step is to show that the error subsystem $\mathcal{S}_2 : \{e_\psi\}$ converges to zero. Thus, consider the following Lyapunov candidate function:

$$V_2(e_\psi, z) = k_5 |e_\psi| + \frac{1}{2}z^2 \,. \tag{2-35}$$

Then, taking its time-derivative, we have:

$$\dot{V}_2(e_\psi, z) = k_5 \, \dot{e}_\psi \, \mathrm{sgn}(e_\psi) + z \, \dot{z} \,. \tag{2-36}$$

When $t \geq 0$, we have $u_2(t) = 0$ and the subsystem $\mathcal{S}_1$ remains at its equilibrium. Substituting Eq. (2-31) into Eq. (2-36) give us

$$\dot{V}_2(e_\psi, z) = -k_4 \, k_5 \, |e_\psi|^\rho \,.$$

Therefore, $\dot{V}_2(e_\psi, z)$ is negative semi-definite, which implies that $e_\psi, z \in \mathcal{L}_\infty$ and, consequently, that $e_\psi \to 0$ and $z \to 0$. ■

2.3 Numerical Simulations

In this section, we present a comparative study among the proposed robust controllers for stabilizing the single-legged hopping robot during its flight phase in the presence of high order terms, unmodeled dynamics and measurement noise. The robot parameters used in the numerical simulations are shown in Table 2.4:

Table 2.4: Hopping robot parameters for numerical simulation.

Parameters	Value	Unit
Mass, m	1.0	Kg
Moment of inertia, I	1.0	$kg\,m^2$
Upper leg length, d	1.0	m
Initial leg angle, ψ_0	$\pi/4$	rad
Desired leg angle, ψ_d	$\pi/3$	rad
Initial body angle, θ_0	$\pi/5$	rad
Desired body angle, θ_d	-1.837	rad
Initial leg length, L_0	0.0	m
Desired leg length, L_d	0.6	m
Minimum leg length, L_{min}	0.0	m
Maximum leg length, L_{max}	1.7	m
Flight phase duration, T_f	2.0	s

The numerical simulations were implemented using ad-hoc developed Matlab scripts and functions (R2019b), running on the Windows 10 Enterprise, 64-bit OS using a Intel(R) Core(TM) i5-8250U CPU @ 1.6GHz, 8GB DDR4 RAM. For all case studies, we use the Euler integration method with a sampling rate of $h = 10^{-3}\,s$ and a simulation time of $T_s = 4\,s$.

2.3.1 Case I: High-order Terms

In this simulation, the following parameters were chosen for each algorithm: (i) Algorithm I, first-order SMC: $k_u = 1.7$, $\gamma = 11.0$ and $p = 0.5$; (ii) Algorithm II, SMC + STA: $k_1 = 15.0$, $k_2 = k_3 = 25.0$, $\rho = 0.5$, $\beta = 10.0$ and $T = 1.6\ s$; (iii) Algorithm III, STA + FTC: $k_4 = 15.5$, $k_5 = 10.5$, $k_6 = 10.5$, $k_7 = 5.5$, $\rho = 0.5$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $T = 1.65\,s$.

The plots depicted in Fig. 2.6 and Fig. 2.7 show the behavior over time for the robot states and control signals, under the influence of high-order terms.

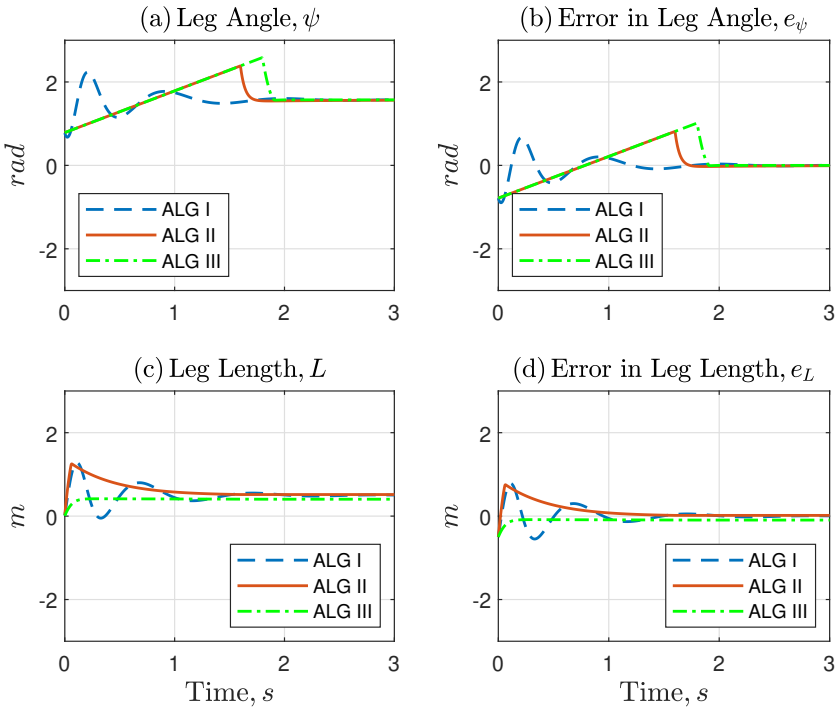


Figure 2.6: Case I, state variables under the effect of high-order terms: (a) leg angle, ψ; (b) leg angle error, e_ψ; (c) leg length, L; (d) leg length error, e_L.

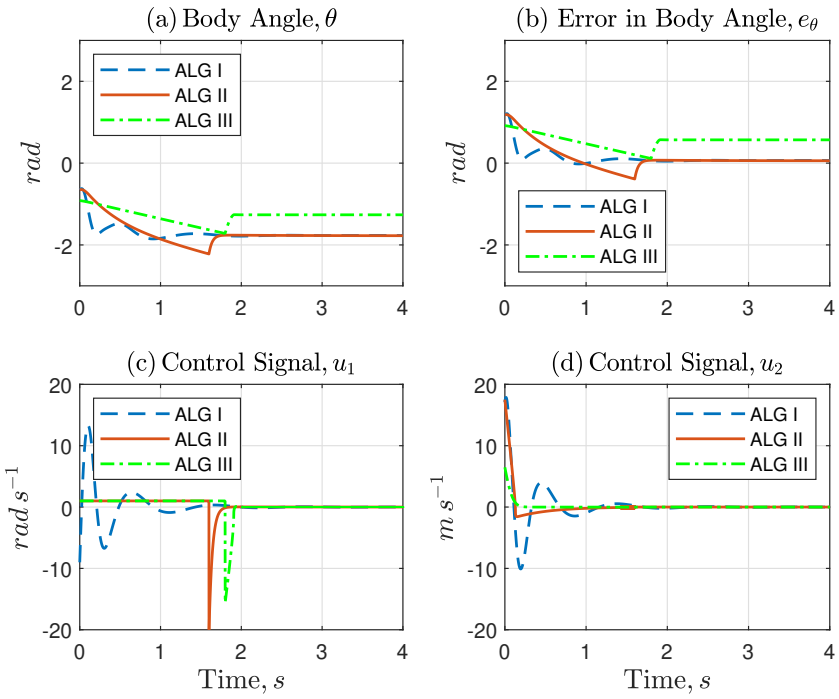


Figure 2.7: Case I, state variables and inputs under the effect of high-order terms: (a) body angle, θ; (b) body angle error, e_θ; (c)-(d) control signals, u_1 and u_2.

We can observe in Fig. 2.6(a) that the error e_ψ goes to zero for all algorithms. On the other hand, the errors e_L and e_θ are driven to zero only for Algorithms I and II, as depicted in Fig. 2.6(b) and Fig. 2.7(b) respectively. Notice that, Algorithm III is only capable of leading the errors e_L and e_θ to a small residual set, due to its weak disturbance rejection characteristics in a short period of time.

We can also observe that the effect of high-order terms in the robot model is suppressed in Algorithms I and II since all error states, in Fig. 2.6(b)-(d) and Fig. 2.7(b), converge to zero in less than 2 s. From Fig. 2.7(c)-(d), it is possible to see that the control signals u_1 and u_2 remain bounded, despite the occurrence of small oscillations in Algorithm I, and the peak due to the switching of control laws in Algorithms II and III respectively. The effect of the switching time T for the convergence of the errors is more evident in the control signal u_1, for Algorithms II and III, as shown in Fig. 2.7(c).

2.3.2 Case II: Unmodeled Dynamics

In this simulation, the following parameters were chosen for each algorithm: (i) Algorithm I, first-order SMC: $k_u = 2.5$, $\gamma = 3.45$ and $p = 0.5$; (ii) Algorithm II, SMC + STA: $k_1 = 8.0$, $k_2 = 1.6$, $k_3 = 5.6$, $\rho = 0.5$, $\beta = 5.0$ and $T = 1.4$ s; (iii) Algorithm III, STA + FTC: $k_4 = 15.5$, $k_5 = 10.5$, $k_6 = 10.5$, $k_7 = 5.5$, $\rho = 0.5$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $T = 1.65\,s$. The unmodeled dynamics was implemented using the following transfer function: $H(s) = (c/\lambda)[1/(s+\omega)]$, where $c = 1$ is the gain constant, $\omega = 1/\lambda$ is the cutoff frequency with $\lambda = 0.1$ and $\omega = 10$ $rad\,s^{-1}$.

The plots depicted in Fig. 2.8 and Fig. 2.9 show the behavior over time for the robot states and control signals, under the influence of unmodeled dynamics. We can observe in Fig. 2.8(a) that the error e_ψ goes to zero only for Algorithms I and II, in contrast to the sustained oscillations obtained by using Algorithm III. On the other hand, the error e_L converges to zero for all algorithms despite of its poor transient behavior, as shown in Fig.2.8(d).

In Fig. 2.9(b), we can verify that Algorithms I and II are able to bring the error e_θ to a small residual set, whereas Algorithm III has an oscillatory response and a longer convergence time. From Fig. 2.9(c)-(d), it is possible to see that the control signals u_1 and u_2 remain bounded, despite the occurrence of high amplitude oscillations in Algorithm III and the multiples peaks due to the switching of control laws in Algorithm II.

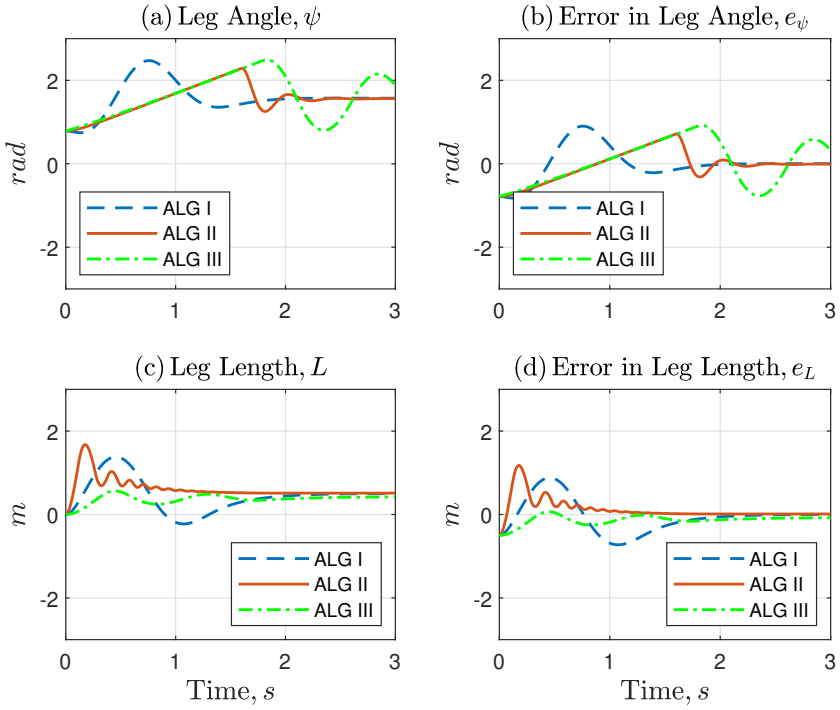


Figure 2.8: Case II, state variables under the effect of unmodeled dynamics: (a) leg angle, ψ; (b) leg angle error, e_ψ; (c) leg length, L; (d) leg length error, e_L.

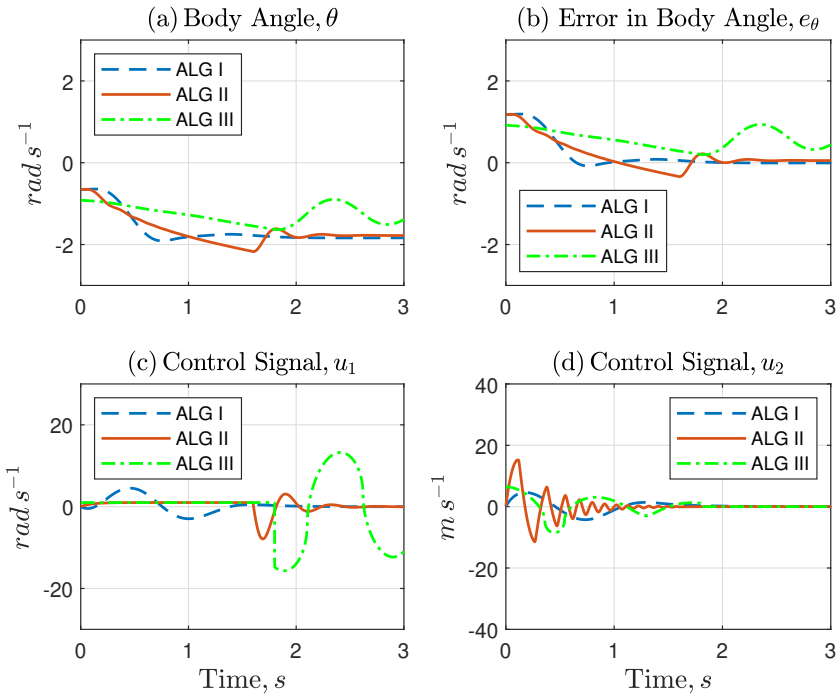


Figure 2.9: Case II, state variables and inputs under the effect of unmodeled dynamics: (a) body angle, θ; (b) body angle error, e_θ; (c)-(d) control signals, u_1 and u_2.

2.3.3 Case III: High-order Terms, Unmodeled Dynamics, and Measurement Noise

Here, we consider the worst-case scenario for the hopping robot during the flight phase, including the influence of high-order terms, unmodeled dynamics and measurement noise in the numerical simulations, as depicted in Fig. 2.10 and Fig. 2.11. In this simulation, the following parameters were chosen for each algorithm: (i) Algorithm I, first-order SMC: $k_u = 2.5$, $\gamma = 15$ and $p = 0.5$; (ii) Algorithm II, SMC + STA: $k_1 = 15.0$, $k_2 = 1.6$, $k_3 = 5.6$, $\rho = 0.5$, $\beta = 5.0$ and $T = 1.5\ s$; (iii) Algorithm III, STA + FTC: $k_4 = 15.5$, $k_5 = 10.5$, $k_6 = 10.5$, $k_7 = 5.5$, $\rho = 0.5$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $T = 1.65\ s$. We use the `awgn` function from Matlab to add white Gaussian noise to the robot configuration q and with signal-to-noise ratio (SNR) of 60 dB.

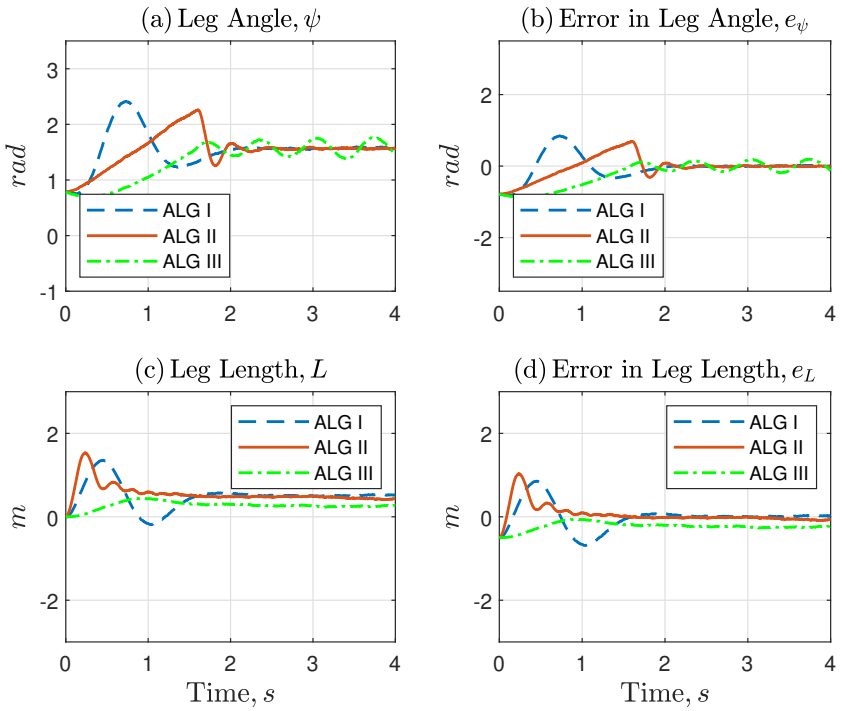


Figure 2.10: Case III, state variables under the effect of high-order terms, unmodeled dynamics and measurement noise: (a) leg angle, ψ; (b) leg angle error, e_ψ; (c) leg length, L; (d) leg length error, e_L.

We can observe in Fig. 2.10(b)-(d) that Algorithms I and II were able to drive the errors e_ψ and e_L to zero in less than 3 s, in contrast to the corresponding oscillatory behavior and non-zero steady-state error provided by Algorithm III. From Fig. 2.11(b), it is possible to see that only Algorithms I and II can stabilize the error e_θ satisfactorily. Finally, Fig. 2.11(c)-(d) show us the oscillatory behavior in control signals, u_1 and u_2, caused by Algorithms

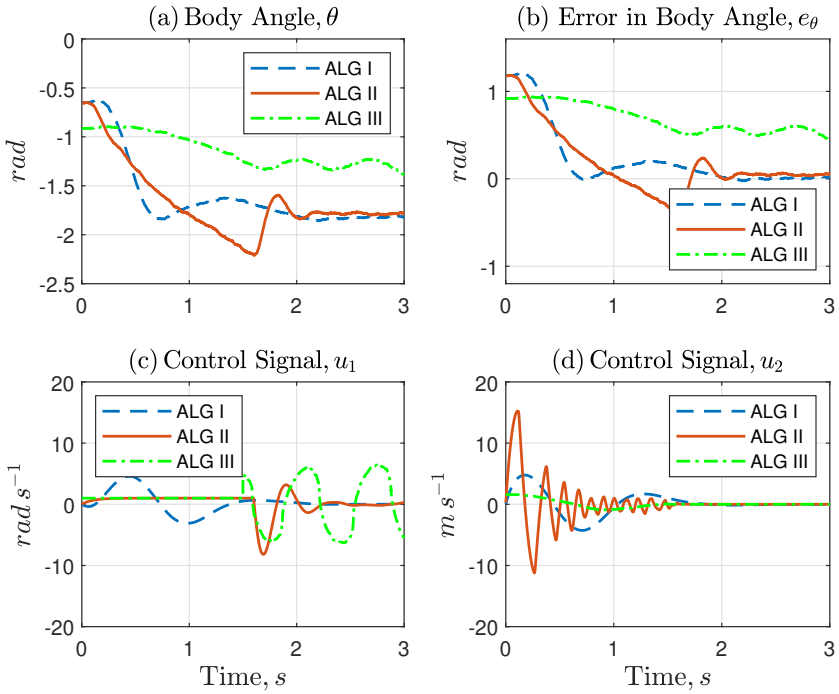


Figure 2.11: Case III, state variables and inputs under the effect of high-order terms, unmodeled dynamics and measurement noise: (a) body angle, θ; (b) body angle error, e_θ; (c)-(d) control signals, u_1 and u_2.

III and II respectively. Notice that, on the contrary, Algorithm I provides a faster and smoother convergence to zero in finite time.

Table 2.3.3 shows the performance metrics used to evaluate the numerical simulations, representing a quantitative analysis for the effectiveness of the robust controllers. The metrics used are the mean average deviation (MAD) of the input signals and the root mean square error (RMSE) of the error states applied for the CASE 3. It can be seen that Algorithms I and II can provide smaller RMSE values for leg angle error e_ψ and body angle error e_θ, compared to Algorithm III. All algorithms are capable of delivering small RMSE values for the leg length error e_L. On the other hand, smaller MAD values for control signals u_1 and u_2 can be obtained only with Algorithm I.

Table 2.5: Performance Metrics of the Robust Control Algorithms.

CASE III	RMSE			MAD	
Variables	e_ψ	e_L	e_θ	u_1	u_2
Algorithm I	0.3372	0.3113	0.3021	0.2061	0.1838
Algorithm II	0.3154	0.2372	0.2240	0.8214	1.1595
Algorithm III	0.4136	0.2431	0.5328	3.3167	0.3068

2.3.4 Discussion and Analysis

In this section, we have introduced a robust control design based on sliding mode approaches for stabilizing a single-legged hopping robot during the flight phase in the presence of model inaccuracies and external disturbances. A comparative study among three promising solutions has been carried out to evaluate their robustness and performance: Algorithm I, based on the first-order SMC approach with chattering attenuation; Algorithm II, based on the combination of SMC and STA approaches; Algorithm III, based on the combination of the STA approach and FTC technique. The Lyapunov stability theory has been used to evaluate the stability properties of the proposed switching control laws based on smooth discontinuous function. Numerical simulations and performance metrics have been included to demonstrate the effectiveness and feasibility of the sliding mode algorithms for legged robots.

After evaluating the results, it is clear the only Algorithms I and II were able to cope with perturbation effects caused by high-order terms, unmodeled dynamics, and measurement noise during the flight phase. Indeed, Algorithm III requires a longer time to achieve finite-time convergence to a small residual set of order $O(L)$ independent of the initial conditions. A major challenge in the design and implementation of Algorithms II and III is to determine to what extent the switching time between the control laws affects the convergence and robustness properties of the algorithms, provided that the constant T is not

considered rigorously in the stability analysis. Although Algorithm I does not suffer from such a drawback, the sliding surface design is not straightforward.

Since hopping robots, at the flight and stance phases, behave as nonholonomic and under-actuated systems, sliding mode approaches based on switching control laws may become a key idea for balance and motion stabilization purposes.

2.4 Sliding Mode Control for an Articulated Robot in Flight Phase

The articulated hopping robot is a model that approximated a real leg. The dynamic equations are similar to a robot manipulator considering which the base is mobile. To simplify the model, we consider that the entire mass of L_3 is concentrated in the joint that joins links 2 and 3, making it so that we do not need to calculate more parameters which would result in a complex model.

In the stance phase, the contact forces are actuating in the robot and, the angle between the robot and, the floor must be within friction cone. In the flight phase, their nonholonomic constraint is the angular momentum conservation. The articulated hopping robot in Fig.(2.12) has the following dynamic equation:

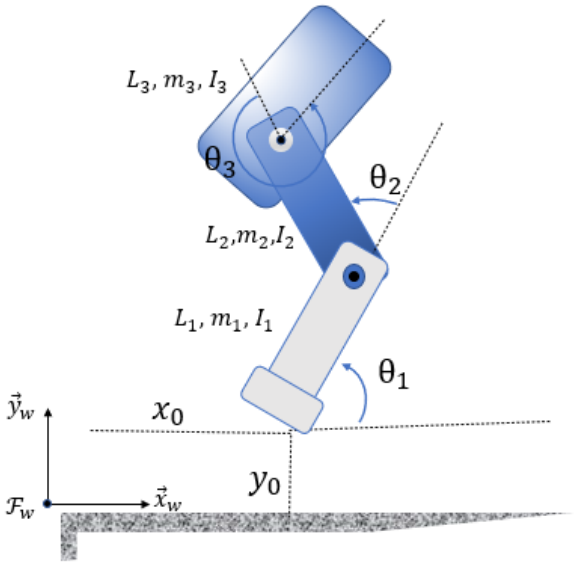


Figure 2.12: An articulated hopping robot, its frames and configuration parameters.

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B\tau\,, \tag{2-37}$$

where $H(q) \in \mathbb{R}^{5\times5}$ in the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{5\times1}$ is the matrix which represent the Coriolis's and centrifugal forces, $G(q) \in \mathbb{R}^{5\times1}$ represent the gravitational terms, $B \in \mathbb{R}^{5\times2}$is the selection matrix which, mapping the inputs in the equations and, $q \in \mathbb{R}^{5\times1}$ are the coordinate of system, the vector $q = [\ x \quad y \quad \theta_1 \quad \theta_2 \quad \theta_3\ ]^T$, note that the coordinates q have two parts, a linear part related to the foot position and, an angular part related to the joint position. The dynamics can be simplified by rewriting the equation in the function of joints variables then, the Eq. (2-37) is rewritten the following form:

$$\bar{h}_{11}\ddot{\theta}_1 + \bar{h}_{12}\ddot{\theta}_2 + \bar{h}_{13}\ddot{\theta}_3 + \bar{c}_1 + \bar{g}_1 = 0\,, \tag{2-38}$$
$$\bar{h}_{21}\ddot{\theta}_1 + \bar{h}_{22}\ddot{\theta}_2 + \bar{h}_{23}\ddot{\theta}_3 + \bar{c}_2 + \bar{g}_2 = \tau_1\,, \tag{2-39}$$
$$\bar{h}_{31}\ddot{\theta}_1 + \bar{h}_{32}\ddot{\theta}_2 + \bar{h}_{33}\ddot{\theta}_3 + \bar{c}_3 + \bar{g}_3 = \tau_2\,, \tag{2-40}$$

where:

$$\bar{h}_{11} = h_{33} - h_{11}^{-1}(h_{13}h_{31}) - h_{22}^{-1}(h_{23}h_{32})\,, \tag{2-41}$$
$$\bar{h}_{12} = \bar{h}_{21} = h_{34} - h_{11}^{-1}(h_{14}h_{31}) - h_{22}^{-1}(h_{24}h_{32})\,, \tag{2-42}$$
$$\bar{h}_{22} = h_{44} - h_{11}^{-1}(h_{14}h_{41}) - h_{22}^{-1}(h_{24}h_{42})\,, \tag{2-43}$$
$$\bar{h}_{13} = \bar{h}_{31} = h_{35} - h_{11}^{-1}(h_{15}h_{31}) - h_{22}^{-1}(h_{25}h_{32})\,, \tag{2-44}$$
$$\bar{h}_{23} = \bar{h}_{32} = h_{45} - h_{11}^{-1}(h_{15}h_{41}) - h_{22}^{-1}(h_{25}h_{42})\,, \tag{2-45}$$
$$\bar{h}_{33} = h_{55} - h_{11}^{-1}(h_{15}h_{51}) - h_{22}^{-1}(h_{25}h_{52})\,, \tag{2-46}$$
$$\bar{c}_1 = c_3 - h_{11}^{-1}(h_{31}c_1) - h_{22}^{-1}(h_{32}c_2)\,, \tag{2-47}$$
$$\bar{c}_2 = c_4 - h_{11}^{-1}(h_{41}c_1) - h_{22}^{-1}(h_{42}c_2)\,, \tag{2-48}$$
$$\bar{c}_3 = c_5 - h_{11}^{-1}(h_{51}c_1) - h_{22}^{-1}(h_{52}c_2)\,, \tag{2-49}$$
$$\bar{g}_1 = g_3 - h_{11}^{-1}(h_{31}g_1) - h_{22}^{-1}(h_{32}g_2)\,, \tag{2-50}$$
$$\bar{g}_2 = g_4 - h_{11}^{-1}(h_{41}g_1) - h_{22}^{-1}(h_{42}g_2)\,, \tag{2-51}$$
$$\bar{g}_3 = g_5 - h_{11}^{-1}(h_{51}g_1) - h_{22}^{-1}(h_{52}g_2)\,. \tag{2-52}$$

The nonholonomic constraint is $\mu = \mu_0$ where, μ is the angular momentum and μ_0 is a constant that represents the initial angular momentum we consider $\mu_0 = 0$. In the Lagrangean equation, we note that the kinetic energy does not depend on θ_1 then we have :

$$\frac{\partial T}{\partial \dot{\theta}_1} - \frac{\partial U}{\partial \theta_1} = 0\,. \tag{2-53}$$

The kinetic rotation energy is given by the Eq .(2-54) [37]:

$$\begin{aligned} T_{rotational} &= \frac{1}{2}\dot{\bar{q}}^T \bar{H} \dot{\bar{q}}^T \,, \\ &= \bar{h}_{11}\dot{\theta}_1^2 + \bar{h}_{12}\dot{\theta}_1\dot{\theta}_2 + \bar{h}_{13}\dot{\theta}_1\dot{\theta}_3 + \bar{h}_{23}\dot{\theta}_2\dot{\theta}_3 + \bar{h}_{22}\dot{\theta}_2^2 + \bar{h}_{33}\dot{\theta}_3^2 \,, \end{aligned} \tag{2-54}$$

where $T_{rotational} \in \mathbb{R}$ is the kinetic energy of rotation, $\dot{\bar{q}} = [\ \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3\ ]^T$ is the joint velocity. Differentiate the Eq. (2-54) in relation $\dot{\theta}_1$ we have:

$$\mu = \bar{h}_{11}\dot{\theta}_1 + \bar{h}_{12}\dot{\theta}_2 + \bar{h}_{13}\dot{\theta}_3 \,, \tag{2-55}$$

Suppose that $\dot{\theta}_2 = u_2$ and $\dot{\theta}_3 = u_1$ we have the following drift less system:

$$\dot{\theta}_1 = -h_{11}^{-1}h_{12}u_2 - h_{11}^{-1}h_{13}u_1 \,, \tag{2-56}$$

$$\dot{\theta}_2 = u_2 \,, \tag{2-57}$$

$$\dot{\theta}_3 = u_1 \,. \tag{2-58}$$

The vector fields $g_1 = \left[\ -h_{11}^{-1}h_{12} \quad 1 \quad 0\ \right]^T$ and $g_2 = \left[\ -h_{11}^{-1}h_{13} \quad 0 \quad 1\ \right]^T$ allows to write the system in the following form:

$$\dot{q} = g_1 u_2 + g_2 u_1 \,. \tag{2-59}$$

Here, we can define the regulation error as $e := q - q_d$ where q_d is the desired robot configuration. For the regulation case, we can assume without loss of generality that $\dot{q}_d = 0$. Thus, the error dynamics can be written as:

$$\dot{e} = g_1 u_1 + q_2 u_2 \,. \tag{2-60}$$

The error system Eq .(2-60) can be transformed into the chained form [37, 47] in order to apply the SMC approach. For this, the following condition must be satisfied:

$$dy_1.\Delta_1 = 0 \,, \tag{2-61}$$

$$dy_1.g_1 = 1 \,, \tag{2-62}$$

$$dy_2.\Delta_2 = 0 \,, \tag{2-63}$$

where,

$$\Delta_0 = span\{g_1, g_2, ad_{g_2}g_1\}\,, \tag{2-64}$$

$$\Delta_1 = span\{g_2, ad_{g_2}g_1\}\,, \tag{2-65}$$

$$\Delta_2 = span\{g_2\}\,, \tag{2-66}$$

$$ad_{g_2}g_1 = \begin{bmatrix} -\dfrac{\partial}{\partial\theta_1}\left(\dfrac{h_{13}}{h_{11}}\right) & 0 & 1 \end{bmatrix}^T . \tag{2-67}$$

The condition are satisfied choosing the following terms:

$$y_1 = e_3\,, \tag{2-68}$$

$$y_2 = e_1 + \int \frac{h_{12}}{h_{11}} de_2\,, \tag{2-69}$$

and we have the following coordinates transform $z = \phi(\theta)$ and the input transform $v = \beta(\theta)u$:

$$z_1 = y_1 = e_3\,, \tag{2-70}$$

$$z_2 = L_{g_1}y_2 = -\frac{h_{13}}{h_{11}}\,, \tag{2-71}$$

$$z_3 = y_2 = e_1 + \int \frac{h_{12}}{h_{11}} de_2\,, \tag{2-72}$$

$$v_1 = u_1\,, \tag{2-73}$$

$$v_2 = (L^2_{g_1}y_2)u_1 + (L_{g_1}L_{g_2}y_2)u_2\,, \tag{2-74}$$

which results in:

$$\dot{z}_1 = u_1\,, \tag{2-75}$$

$$\dot{z}_2 = u_2\,, \tag{2-76}$$

$$\dot{z}_3 = z_2u_1\,, \tag{2-77}$$

$$v_1 = u_1\,, \tag{2-78}$$

$$v_2 = -\frac{\partial}{\partial e_{\theta_2}}\left(\frac{h_{13}}{h_{11}}\right), \tag{2-79}$$

Notice that, the motion dynamics of the articulated hopping robot can be described by the sub-optimal chained form Eq.(2-75) which was previously used in the Section 2.2.2. Therefore, the controllers based controllers designed in the previous section can also be used. In this section, we reproduce the control techniques adopted for the hopping robot with a prismatic joint. We will use the theorems already demonstrated and develop numerical simulations

in Matlab considering unmodeled dynamics and noise. We will calculate the metrics of error and control signals.

2.4.1 Algorithm I : First-Order SMC

The idea in this section is to control the joints of an articulated robot hooping in the flight phase through an SMC controller, the choice of this controller is due to its robustness to external matched and unmatched disturbances and parametric uncertainties. We can design the control laws that guarantee the control of a system in the chained form, we developed the following control law:

$$u_1 = -\,k\,z_1 + \gamma z_2\,\mathrm{sgn}(s)|s|^p\,, \tag{2-80}$$

$$u_2 = -\,k\,z_2 - \gamma z_1\,\mathrm{sgn}(s)|s|^p\,, \tag{2-81}$$

where $k \in \mathbb{R}$ is a proportional gain, $p \in \mathbb{R}$ is a constant parameter such that $0 \leq p \leq 1/2$, $\gamma > 0$ is the control gain and, $s \in \mathbb{R}$ is the sliding surface given by:

$$s = -2\,z_3 - z_1\,z_2\,. \tag{2-82}$$

According to the Theorem 2.1 we can prove that the $\lim_{t\to\infty} z_1, z_2, s = 0$ and hence $\lim_{t\to\infty} z_3 = 0$, it means that $\lim_{t\to\infty} e_1, e_2, = 0$ and hence $\lim_{t\to\infty} e_3 = 0$. Although they have advantages, sliding control has disadvantages such as, the chattering phenomenon that occurs due to imperfections in the switching of the controller. This effect can cause serious problems such as heating or instability in unmodeled dynamics at high-frequencies. There are ways to mitigate the chattering as shown in Eq (2-80) where the term p causes the discontinuity in the control law to be smoothed.

2.5 Algorithm II: SMC + STA

Although conventional first-order SMC can guarantee stability and robustness properties under external disturbances, these are subject to chattering, which can deteriorate the performance of the system response. The chattering phenomenon can be attenuated by using the Super Twisting Algorithm (STA) [48, 52].

The key idea is to divide the system in the chained form into two subsystems, then using two discontinuous control laws to stabilize the subsystems. First, the errors e_1 and e_2 are stabilized and secondly, e_3 is stabilized. Although

conventional first-order SMC can guarantee stability and robustness properties under external disturbances, these are subject to chattering, which can deteriorate the performance of the system response. The chattering phenomenon can be attenuated by using the Super Twisting Algorithm (STA). The key idea is to divide the system in the chained form into two subsystems, then using two discontinuous control laws to stabilize the subsystems. First, the errors e_1 and e_2 are stabilized and secondly, e_3 is stabilized. Then, let us design the first discontinuous control u_1 is based on the STA approach as:

$$u_1 = \begin{cases} -k_1 \left|e_3\right|^{\rho} \operatorname{sgn}(e_3) + w\,, & t \geq T\,, \\ 1\,, & T_0 < t < T\,, \end{cases} \tag{2-83}$$

with $\dot{w} = -k_2 \left|e_1\right|^{\rho} \operatorname{sgn}(e_1)$, where $k_2 > 0$ and $k_3 > 0$ are the proportional gains, $\rho \in (0, 1/2]$ is a constant parameter, which attenuates the chattering phenomena, $T \in \mathbb{R}^+$ is the switching time, which is responsible to switch the control laws, given by Eq. (2-83), and $T_0 \geq 0$ is the initial time. It is worth noting that the value of T is chosen empirically according to the error norm that is tolerable (0.08) and the duration of the flight phase T_f. The control we design the second discontinuous control law based on the SMC approach as :

$$u_2 = \begin{cases} 0\,, & t \geq T\,, \\ -\lambda e_2 - k_3 \left|s\right|^{\rho} \operatorname{sgn}(s)\,, & T_0 < t < T\,, \end{cases} \tag{2-84}$$

where $k_1 > 0$ is the proportional gain and $s \in \mathbb{R}$ is the sliding surface given by:

$$s = -\lambda e_1 + e_2\,, \tag{2-85}$$

with $\beta \in \mathbb{R}$ being the slope of the sliding surface. Using the Theorem 2.2, we prove that $\lim_{t\to\infty} e_3(t) = 0$, and $\lim_{t\to T} s(t) = 0$ and, consequently, $\lim_{t\to T} e_1, e_2 = 0$.

2.5.1 Algorithm III: STA+FTC

Here, a robust control algorithm was developed based on the combination of the Super Twisting Algorithm approach and the Finite-Time Controller (FTC) technique to direct the joints of the hopping robot in the flight phase to the desired values. The control technique is similar to that one introduced by Thomas *et al.* [46] to stabilize the position and orientation of a differential drive. The control approach consists of designing two discontinuous control laws to first stabilize e_1 and e_2 and then stabilize e_3.The control approach consists of designing two discontinuous control laws to first stabilize e_1 and e_2

and then stabilize e_3. So the first u_1 discontinuous control law is based on the STA approach as:

$$u_1 = \begin{cases} -k_4|e_3|^{\rho}\text{sgn}(e_3) + w\,, & t \geq T\,, \\ 1\,, & T_0 < t < T\,, \end{cases} \tag{2-86}$$

with $\dot{w} = -k_5\,\text{sgn}(e_3)$, where $k_4 > 0$ and $k_5 > 0$ are the proportional gains. Now we design the second u_2 discontinuous control law based on the FTC technique as:

$$u_2 = \begin{cases} 0\,, & t \geq T\,, \\ -k_6|e_1|^{\alpha_1}\,\text{sgn}(e_1) + \kappa\,, & T_0 < t < T\,, \end{cases} \tag{2-87}$$

with $\kappa = -k_7\,|e_2|^{\alpha_2}\,\text{sgn}(e_2)$, where $k_6 > 0$ and $k_7 > 0$ are the proportional gains, and $\alpha_1, \alpha_2 \in (0, 1/2]$ are constant parameters that attenuate the chattering phenomena. Based on the Theorem 2.3 we can prove that $\lim_{t\to\infty} e_3$ and $\lim_{t\to T} e_1, e_2 = 0$.

2.5.2 Numerical Simulations

The purpose of this subsection is to make a comparative study between the proposed robust controllers to stabilize an articulated hopping robot in the flight phase in the presence of unmodeled dynamics and noise. The simulation parameters used in numerical simulations are shown in Table 2.6

The numerical simulations were implemented using ad-hoc developed Matlab scripts and functions (R2019b), running on the Windows 10 Enterprise, 64-bit OS using a Intel(R) Core(TM) i5-8250U CPU @ 1.6GHz, 8GB DDR4 RAM. For all case studies, we use the Euler integration method with a sampling rate of $h = 10^{-3}\,s$, a simulation time of $T_s = 20\,s$ the numeric integration method implemented is Euler.

Table 2.6: Hopping robot parameters for numerical simulation.

Parameters	Value	Unit
Mass of Link 1, m_1	5.0	Kg
Moment of inertia of Link 1, I_1	1.0	$kg\,m^2$
Length of of Link 1, L_1	0.4	m
Center of Mass of Link 1, L_{c1}	0.2	m
Mass of Link 2, m_2	8.0	rad
Moment of inertia of Link 2, I_2	1.0	$Kg.m^2$
Length of of Link 2, L_2	0.7	m
Center of Mass of Link 2, L_{c2}	0.35	m
Mass of Link 3, m_3	30.0	m
Moment of inertia of Link 3, I_3	1.0	$Kg.m^2$
Length of of Link 3, L_3	0.8	m
Center of Mass of Link 3, L_{c3}	0.0	m
Initial Position of Joint 1, q_{01}	$\pi/3$	rad
Initial Position of Joint 2, q_{02}	$\pi/2$	rad
Initial Position of Joint 3, q_{03}	$-\pi/2$	rad
Desired Position of Joint 1, q_{d1}	$\pi/2$	rad
Desired Position of Joint 2, q_{d2}	1.0	rad
Desired Position of Joint 3, q_{d3}	$-2\pi/3$	rad
Flight phase duration, T_f	4.0	s

2.5.2.1 First Case: Unmodeled Dynamics

In this simulation, the following parameters were chosen for each algorithm: (i) Algorithm I, first-order SMC:$k = 10$ $\gamma = 600.0$ and $p = 0.5$; (ii) Algorithm II, SMC + STA: $k_1 = 10.5$, $k_2 = 10.03$,$k_3 = 15.04$, $\rho = 0.5$, $\beta = 1.0$ and $T = 1.5\ s$; (iii) Algorithm III, STA + FTC: $k_4 = 15.04$, $k_5 = 1.0$, $k_6 = 50.0$, $k_7 = 14.0$, $\rho = 0.5$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $T = 4.65\,s$. The unmodeled dynamics was implemented using the following transfer function: $H(s) = (c/\lambda)[1/(s+\omega)]$, where $c = 1$ is the gain constant, $\omega = 1/\lambda$ is the cutoff frequency with $\lambda = 0.1$ and $\omega = 10\ rad\,s^{-1}$.

The plots depicted in 2.13 and 2.14 show the behavior over time for the robot states and control signals, under the influence of unmodeled dynamics. We can see that the error e_1 and e_2 are driven to zero in algorithms I, II and III despite its poor transient as shown in Fig.2.13.(b) and Fig.2.13.(d) On the other hand, in Fig.2.14(b) we can verify that Algorithms I and II are capable bring e_3 to zero, whereas Algorithm III has a small oscillatory response.From Fig 2.14 (c)-(d) it is possible to observe that the control signals u_1 and u_2 remain limited despite the high-frequency oscillations observed in the control signals and the presence of peaks due to switching as in Algorithm III.

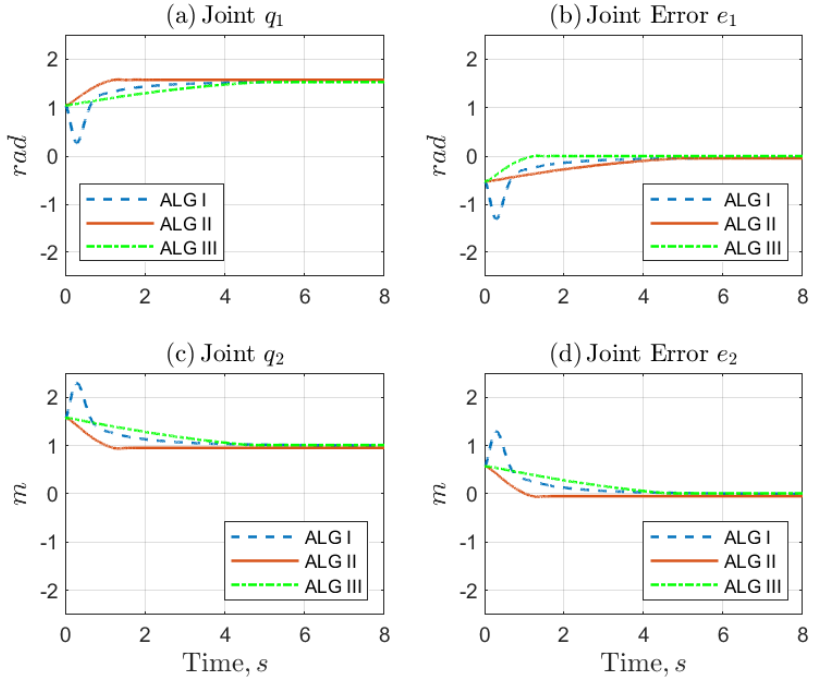


Figure 2.13: Case II, state variables under the effect of unmodeled dynamics: (a) Joint Position, q_1; (b) Joint Position Error, e_3; (c) Joint Position, e_2; (d) Joint Position Error, e_2.

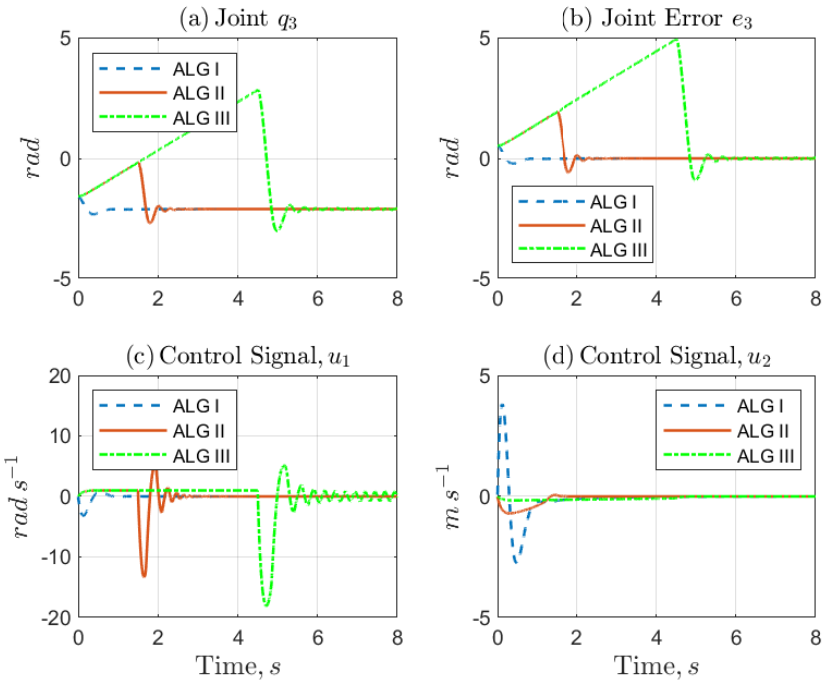


Figure 2.14: Case II, state variables and inputs under the effect of unmodeled dynamics: (a) Joint Position, q_3; (b) Joint Position Error, e_3; (c)-(d) control signals, u_1 and u_2.

2.5.2.2 Case II: Unmodeled Dynamics, and Measurement Noise

Here, we consider the worst-case scenario for the hopping robot during the flight phase, where we include unmodeled dynamics and noise in the measurement in the numerical simulation. In this simulation, the following parameters were chosen for each algorithm: (i) Algorithm I, first-order SMC: $k = 10$, $\gamma = 600.0$ and $p = 0.5$; (ii) Algorithm II, SMC + STA: $k_1 = 10.5$, $k_2 = 10.03$, $k_3 = 15.04$, $\rho = 0.5$, $\lambda = 1.0$ and $T = 1.5\ s$; (iii) Algorithm III, STA + FTC: $k_4 = 15.5$, $k_5 = 1.0$, $k_6 = 50.0$, $k_7 = 14.0$,$\rho = 0.5$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $T = 4.5\,s$. We use the `awgn` function from Matlab to add white Gaussian noise to the robot configuration q and with signal-to-noise ratio (SNR) of 60 dB.

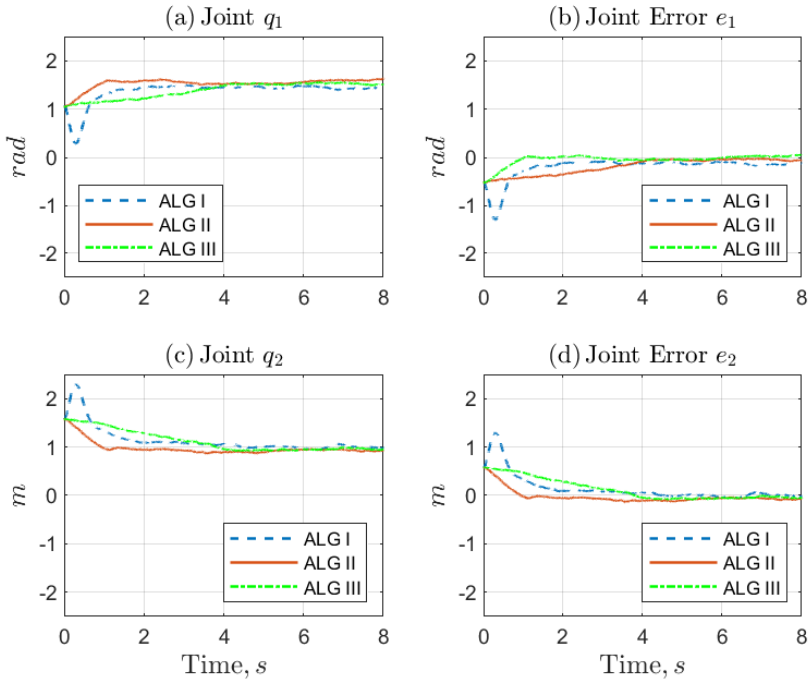


Figure 2.15: Case II, state variables under the effect of unmodeled dynamics and measurement noise: (a) Joint Position, q_1; (b) Joint Position Error, e_3; (c) Joint Position, e_2; (d) Joint Position Error, e_2.

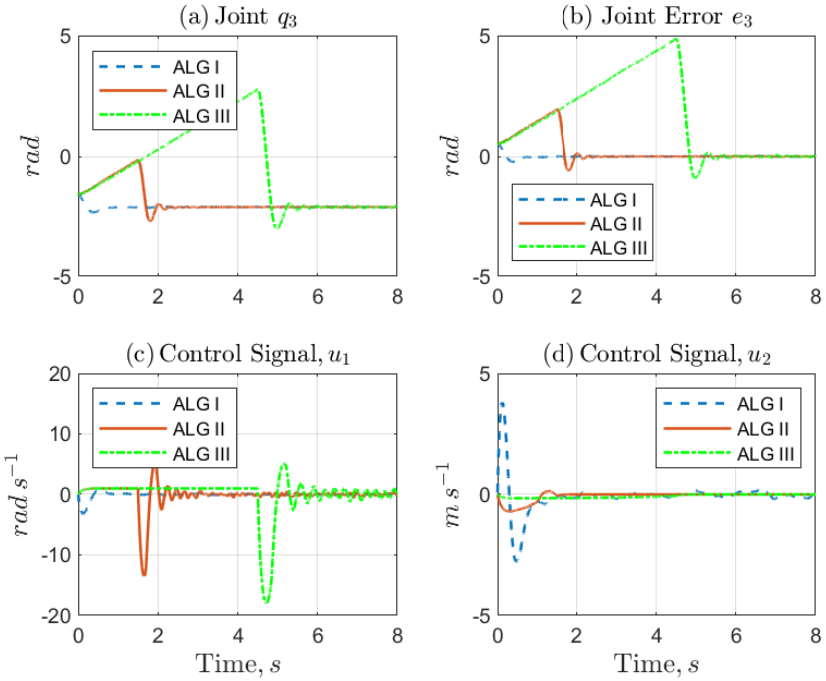


Figure 2.16: Case II, state variables and inputs under the effect of unmodeled dynamics and measurement noise: (a) Joint Position, q_3; (b) Joint Position Error, e_3; (c)-(d) control signals, u_1 and u_2.

We can see in Fig .2.15(b) and Fig .2.15(d) that Algorithms I and II were able to direct e_1, e_2 and e_3 to zero in less than $4s$, in contrast to the small oscillatory response in permanent regime than in e_3. Finally, Fig 2.16 (c) shows a small oscillatory behaviour in the u_1 control signal caused by Algorithms II and III respectively, observed that in Fig. 2.16 (d), the same behaviour is not observed in the control signal u_2. Comparing the algorithms used, we observe that all control signals are limited. The algorithm I presents a smooth convergence to zero infinite time and, Algorithm II presents a faster finite-time convergence.

Table 2.5.2.2 shows the performance of the metrics to evaluate the numerical simulations, representing a quantitative analysis for the effectiveness of the robust controllers. The metrics used are the mean absolute deviation (MAD) of the input signals and, the root means square error (RMSE) of the state error for case III. It can be seen that smaller MAD values for the control signals u_1 and u_2 can be provides only with Algorithm I .

Table 2.7: Performance Metrics of the Robust Control Algorithms.

CASE III	RMSE			MAD	
Variables	e_1	e_2	e_3	u_1	u_2
Algorithm I	0.1905	0.19314	0.04261	0.1000	0.1587
Algorithm II	0.0704	0.0886	0.3710	0.9599	0.0452
Algorithm III	0.14748	0.14943	1.4760	0.959	0.0452

2.5.3 Discussion and Analysis

In this subsection, we introduce a robust control based on the sliding mode approach to stabilize a hopping robot during the flight phase in the presence of unmodeled dynamics and noise in the measurement. A comparative study among three promising solutions has been carried out to evaluate their robustness and performance: Algorithm I, based on the first-order SMC approach with chattering attenuation; Algorithm II, based on the combination of SMC and STA approaches; Algorithm III, based on the combination of the STA approach and FTC technique. The Lyapunov stability theory has been used to evaluate the stability properties of the proposed switching control laws based on smooth discontinuous function. Numerical simulations and performance metrics have been included to demonstrate the effectiveness and feasibility of the sliding mode algorithms for legged robots.

After evaluating the results, it is clear that only algorithms I and II can bring the joint position error to zero in less than $2s$ in the presence of unmodeled dynamics and measurement noise. Algorithms II and III need a longer time to ensure that all joints positions errors converge to zero. A major challenge in the design and implementation of Algorithms II and III is to determine to what extent the switching time between the control laws affects the convergence and robustness properties of the algorithms provided that the constant T is not considered rigorously in the stability analysis. Although Algorithm I does not suffer from such a drawback, the sliding surface design is not straightforward.

Another challenge in numerical simulation is to define the parameters in the hopping robot so that there are no problems when simulating because we have a numerical poor conditioning problem because according to the choice of the parameters, the inverse matrix transformation of the chained form generates a larger or smaller control signal u_2 undermining the control action this is the reason why the proportional gain k in Algorithm I is higher.

2.6 Articulated Hopping Robot in Stance Phase

Now, we consider the articulated hopping robot shown in the Fig. 2.17, generalizing the motion control problem for flight and stance phases as a single case involving the presence and absence of contact forces actuating in the robot foot, instead of being addressed as two separate and distinct cases. The stance phase is the situation where, there are contact forces in the leg which affect of the system dynamic, the controller must be deal with this forces and to ensures the desired joint position. Considering in the same figure that, the body does not rotate and the CoM is close to the body, it means, $m_1 << m_3$ and $m_2 << m_3$ and the center of mass body $L_{c3} = 0$. Let coordinates vector

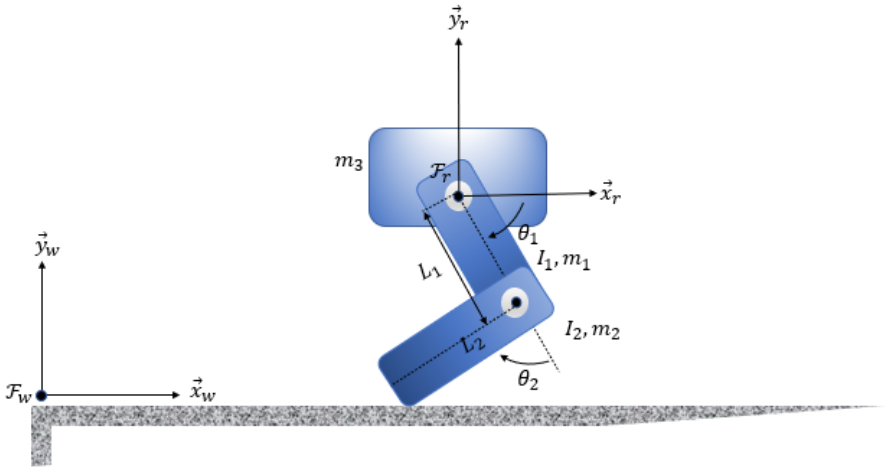


Figure 2.17: Articulated Hopping robot in stance phase

$q = \begin{bmatrix} x_r & y_r & \theta_1 & \theta_2 \end{bmatrix}^T$ which represents the position of thigh and the joints angles. The dynamic of systems is given by the following equation.

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B\tau + J_e^T\lambda\,, \tag{2-88}$$

where $q \in \mathbb{R}^4$ is the coordinates, $H(q) \in \mathbb{R}^{4\times4}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{4\times1}$ is the Coriolis and centrifugal matrix, $G(q) \in \mathbb{R}^{4\times1}$ is the gravitational and potential terms, $\lambda \in \mathbb{R}^{2\times1}$ is the Lagrange multiplier vector, which represents the forces and moments of contact at the end of the leg if $\lambda = 0$ means that no forces or moments are acting on the robot which means that the hopping robot is in the flight phase, if $\lambda \neq 0$ means that contact forces are acting on the end of the robot so it is in the stance phase and $J_e \in \mathbb{R}^{2\times4}$ is Jacobian of the leg.

The extremity position of hopping robot is giving by:

$$x_e = x_r + L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2)\,, \tag{2-89}$$

$$y_e = y_r - L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2)\,, \tag{2-90}$$

and their velocity:

$$\dot{p}_e = J_e \dot{q}\,, \tag{2-91}$$

where $\dot{p}_e = \begin{bmatrix} \dot{x}_e & \dot{y}_e \end{bmatrix}^T$ is the foot velocity.

$$J_e = \begin{bmatrix} 1 & 0 & -L_1 c_1 - L_2 c_{12} & -L_2 c_{12} \\ 0 & 1 & -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \end{bmatrix}. \tag{2-92}$$

The Lagrange multiplier vector is found using the following constrains in stance phase:

$$J_e \dot{q} = 0\,, \tag{2-93}$$

deriving the constraint:

$$\begin{aligned} &J_e \ddot{q} + \dot{J}_e \dot{q} = 0\,, \\ &J_e H^{-1}(q)(B\tau + J_e^T \lambda - C(q,\dot{q})\dot{q} - G(q)) + \dot{J}_e \dot{q} = 0\,, \\ &\lambda = (J_e (H(q))^{-1} J_e^T)^{-1}(C(q,\dot{q})\dot{q} + G(q) - B\tau - \dot{J}_e \dot{q})\,. \end{aligned}$$

The system can split into two parts a passive part that can not control directly by actuators such as, x_r and y_r and an active part which are control directly using actuators such as θ_1 and θ_2. The system can be written in passive and active terms:

$$H_{pp}\ddot{q}_p + H_{pa}\ddot{q}_a + C_p \dot{q}_p + G_p = J_p^T \lambda\,, \tag{2-94}$$

$$H_{ap}\ddot{q}_p + H_{aa}\ddot{q}_a + C_a \dot{q}_a + G_a = \tau + J_a^T \lambda\,, \tag{2-95}$$

where the suffix **p** represents passive and **a** active.

2.6.1 Computed Torque Controller for a Hopping Robot in Stance Phase

The classical linear controller is an alternative solution for the joint control of hopping robot in Fig.2.17. In this case, we choose a system input which cancels all nonlinear terms and linearize the error system in closed-loop,

whose desired response can be defined by by choosing suitable gain matrices K_p and K_d. Considering Eq.2-94 we choose the following control input τ as:

$$\tau = (H_{aa} - H_{ap}H_{pp}^{-1}H_{pa})\,u + H_{ap}H_{pp}^{-1}(C_p\dot{q}_p + G_p)\,, \\ -C_a\dot{q}_a - G_a - (J_a^T - H_{ap}H_{pp}^{-1}J_p^T)\lambda\,. \tag{2-96}$$

Now, we choose the linear control law u such as:

$$u = \ddot{q}_{ad} - K_d(\dot{q}_a - \dot{q}_{ad}) - K_p(q_a - q_{ad})\,, \tag{2-97}$$

where $q_{ad} \in \mathbb{R}^2$ is the desired value for the active joints, $\dot{q}_{ad}, \ddot{q}_{ad} \in \mathbb{R}^2$ its desired velocity and acceleration, $K_d \in \mathbb{R}^{2\times2}$ and $K_p \in \mathbb{R}^{2\times2}$ are positive definitive matrices. Then, Eq.2-94 becomes:

$$\ddot{q}_p = H_{pp}^{-1}(J_p^T\lambda - H_{pa}u - C_p\dot{q}_p - G_p)\,, \tag{2-98}$$

$$\ddot{q}_a = u\,. \tag{2-99}$$

Selecting the derivative and proportional gain matrices to be $K_d = \mathrm{diag}\{2\zeta_1\omega_1\,, 2\zeta_2\omega_2\}$ and $K_p = \mathrm{diag}\{\omega_1^2\,, \omega_2^2\}$, where ζ_i and ω_i for $i = 1, 2$ are respectively the damping factor and the natural frequency, we ensure that the error system converges to zero during the flight time and the stance time.

2.6.2 Sliding Mode Control for a Hopping Robot

Although the PD controller ensures the stability properties for the active joints, we can show that such a controller does not have a satisfactory performance in the presence of parametric uncertainties and external disturbances. The sliding mode control approach can deal with this problem in the system in Eq.2-94. Considering the sliding surface:

$$s = \dot{e}_a + K_s e_a\,, \tag{2-100}$$

where K_s is a positive definitive matrix. We need prove that the sliding surface is asymptotically stable :

$$\begin{aligned} 2V(s) =& s^2 \,, \\ \dot{V}(s) =& s\dot{s} \\ =& s(\ddot{q}_a - \ddot{q}_{ad}) + K_s\dot{e}_a \\ =& s(H_{aa} - H_{ap}H_{pp}^{-1}H_{pa})^{-1}(\tau + (J_a^T - H_{ap}H_{pp}^{-1}J_{pp}^T)\lambda \,, \\ & - C_a\dot{q}_a - G_a - H_{ap}H_{pp}^{-1}(C_p\dot{q}_p + G_p)) \,. \end{aligned} \tag{2-101}$$

Choosing the following control law:

$$\begin{aligned} \tau =& (H_{aa} - H_{ap}H_{pp}^{-1}H_{pa})u - (J_a^T - H_{ap}H_{pp}^{-1}J_{pp}^T)\lambda + C_a\dot{q}_a \,, \\ & + G_a + H_{ap}H_{pp}^{-1}(C_p\dot{q}_p + G_p) \,. \\ =& su \,. \end{aligned} \tag{2-102}$$

If $u = \beta sign(s)|s|^{1/2}$ results in:

$$\dot{V}(s) = -\beta|s|^{3/2} \,. \tag{2-103}$$

The above equation ensures the existence of sliding mode then, when the surface is reach we have:

$$\dot{e}_a = -K_a e_a \,. \tag{2-104}$$

The error e_a converges to zero and its convergence depends on value of K_a gain.

2.6.3 Robust Control for a Hopping Robot

The PD and SMC controllers in the previously subsections does not to deal with uncertainties and disturbances in the system, in the practice we find situations where, the uncertainties in robot parameters or in the model,disturbances, the terrain features affect the robot performance in this case, it is necessary that, the controller to be robust in this situations, to reach this aim, we use the robust control. In the sliding mode control if the controller reach the sliding surface even with the uncertainties in the model and disturbance we say that the controller is robust. Considering the system

in Eq. (2-94) writing the system the following form:

$$H_a\ddot{q}_a - n = \tau\,, \tag{2-105}$$

where

$$H_a = (H_{aa} - H_{ap}H_{pp}^{-1}H_{pa})\,, \tag{2-106}$$
$$n = (J_a^T - H_{ap}H_{pp}^{-1}J_{pp}^T)\lambda + C_a\dot{q}_a + G_a + H_{ap}H_{pp}^{-1}(C_p\dot{q}_p + G_p)\,. \tag{2-107}$$

The matrix H_a^{-1} is lower and upper bounded:

$$H_{min} \leq ||H_a^{-1}|| \leq H_{max} < \infty\,. \tag{2-108}$$

Then, considering the following control law [54]:

$$\tau = \hat{H}_a u + \hat{n}\,. \tag{2-109}$$

Adding and subtracting u and multiplying per $\hat{H}_a^{-1}$ we have:

$$\ddot{q}_a = (I - H_a^{-1}\hat{H}_a)u + H_a^{-1}\tilde{n} + u\,, \tag{2-110}$$
$$\ddot{q}_a = u + \eta\,. \tag{2-111}$$

where the uncertain η and the error estimation n are:

$$\eta = (I - H_a^{-1}\hat{H}_a)u + H_a^{-1}\tilde{n}\,, \tag{2-112}$$
$$n = \hat{n} - n\,. \tag{2-113}$$

According the following assumptions in [54]:

Assumption 1 : $||\ddot{q}_{ad}|| \leq \ddot{Q}_{ad} < \infty \quad \forall \ddot{q}_{ad}$

Assumption 2 : $||I - H_a^{-1}\hat{H}_a|| \leq \alpha < 1$

Assumption 3 : $||\tilde{n}|| \leq \phi < \infty$

The matrix H_a^{-1} is lower and upper bounded:

$$H_{min} \leq ||H_a^{-1}|| \leq H_{max} < \infty\,, \quad \forall t\,. \tag{2-114}$$

In [54] is given a possible matrix $\hat{H}$ that satisfied the assumption 2 this matrix is given by:

$$\hat{H} = \frac{2}{H_{max} + H_{min}} \,. \tag{2-115}$$

Results in:

$$||H_a^{-1}\hat{H}_a - I|| \leq \frac{H_{max} - H_{min}}{H_{max} + H_{min}} = \alpha < 1 \,. \tag{2-116}$$

Using the sliding surface and the Lyapunov candidate function in the Eq.(2-100) and Eq.(2-101) we have:

$$2\,V(s) = s^T s \tag{2-117}$$

$$\begin{aligned} 2\,\dot{V}(s) =& s^T\dot{s} + \dot{s}^T s \\ =& s^T(u + \eta - \ddot{q}_{ad} + K_a\dot{e}) \,. \end{aligned} \tag{2-118}$$

Choosing the following u:

$$u = \ddot{q}_{ad} - K_s\dot{e} - \omega \,. \tag{2-119}$$

Replacing the Eq. (2-119) into Eq. (2-117):

$$\begin{aligned} \dot{V}(s) =& s^T\dot{s} \\ =& s^T(-\omega + \eta) \\ =& s^T(\eta - \omega) \,. \end{aligned} \tag{2-120}$$

In Eq. (2-120), we note that the robustness term ω ensures that the surface converges to zero despite the uncertain η if $|\eta| < |\omega|$ we propose the following robustness term ω:

$$\omega = \rho\, sign(s) \,, \tag{2-121}$$

and, then, $\dot{V}(s)$ takes the form:

$$\dot{V}(s) \leq (\eta - \omega)s < 0 \,, \tag{2-122}$$

$$\leq |\eta - \omega||s| < 0 \,,$$

$$\leq |\eta||s| - |\omega||s| < 0 \,. \tag{2-123}$$

If the robustness term ρ is greater that η the function is negative semi definitive the minimum value of robustness term is given by :

$$\begin{aligned} |\eta| &= ||(I - H_a^{-1}\hat{H}_a)u + H_a^{-1}\tilde{n}||\,, \qquad (2\text{-}124)\\ &\leq ||(I - H_a^{-1}\hat{H}_a)||.||u|| + ||H_a^{-1}\tilde{n}||\,,\\ &\leq ||(I - H_a^{-1}\hat{H}_a)||.||u|| + ||H_a^{-1}||.||\tilde{n}||\,. \qquad (2\text{-}125) \end{aligned}$$

Considering the assumption (1),(2) and (3) we have:

$$\begin{aligned} |\eta| &\leq \alpha||u|| + ||H_{max}||.||\phi||\,,\\ &= \alpha||\ddot{Q}_{max}|| + \lambda||\dot{e}|| + ||\beta||||H_{max}||.||\phi||\,. \qquad (2\text{-}126) \end{aligned}$$

Hence the robustness term is given by:

$$\begin{aligned} \rho &\geq \alpha||u|| + ||H_{max}||.||\phi||\,,\\ &= \alpha||\ddot{Q}_{max}|| + \lambda||\dot{e}|| + \rho + ||H_{max}||.||\phi||\,,\\ \rho &\geq (1-\alpha)^{-1}(||\ddot{Q}_{max}|| + \lambda||\dot{e}|| + ||H_{max}||.||\phi||)\,. \qquad (2\text{-}127) \end{aligned}$$

2.6.4 Numerical Simulation

In this subsection, we present a comparative study between two controllers: computed torque controller and SMC. The objective is to stabilize a hopping robot instance phase in the presence of parametric uncertainties and external disturbances. The robot parameters are described in Table 2.8:

Table 2.8: Simulation parameters

Parameters	Value	Unit
Body mass(m_3)	10.0	Kg
Mass of link 1 (m_1)	1.0	Kg
Length of link 1 (L_1)	0.8	m
Center of mass of link 1 (L_{c1})	0.5	m
Constant torsion spring of joint 1 (k_{s1})	25	$N.m/rad$
Momentum of Inertia of body of link 1 (I_1)	1	$Kg.m^2$
Mass of link 2 (m_2)	1.0	Kg
Length of link 2 (L_2)	0.5	rad
Center of mass of link 2 (L_{c2})	0.25	m
Momentum of Inertia of body of link 2 (I_2)	0.5	$Kg.m^2$
Constant torsion spring of joint 2 (k_{s1})	25	$N.m/rad$

The numerical simulations were implemented using ad-hoc developed Matlab scripts and functions (R2019b), running on the Windows 10 Enterprise, 64-bit OS using Intel (R) Core (TM) i5-8250U CPU @ 1.6GHz, 8GB DDR4 RAM. For all case studies, we use the Euler integration method with a sampling rate of $h = 10^{-4}\, s$ and a simulation time of $T_s = 5\, s$. In this simulation, we use the following gains for the computed torque control: $k_p = 100$ and $k_d = 20$ and for the SMC control we will use the following parameters: $K_a = 10I$, where I is the identity matrix, $\beta = 100$. The external disturbance is of the type $d = [0.1\sin(t)\ 0.1\cos(t)]^T$.

The plots depicted in Fig. 2.18(a), Fig. 2.18(b) and 2.20(c) show the behaviour over time for robot state and control signals under parametric uncertainties and external disturbances. We can observe in Fig.2.18(a) and Fig.2.18(b) that the robot reaches the equilibrium point ($x_d = 0m\,, y_d = 1.3m$) for the SMC controller, the computed torque control can not bring the CoM coordinates to the equilibrium point. In Fig. 2.18(c) and Fig. 2.18(d), it can be seen that the computer torque control was not able to guarantee the robot joints reach the desired positions, represented in the graphics by the green dotted line. The SMC control can ensure that the joint position error goes to approximately zero.

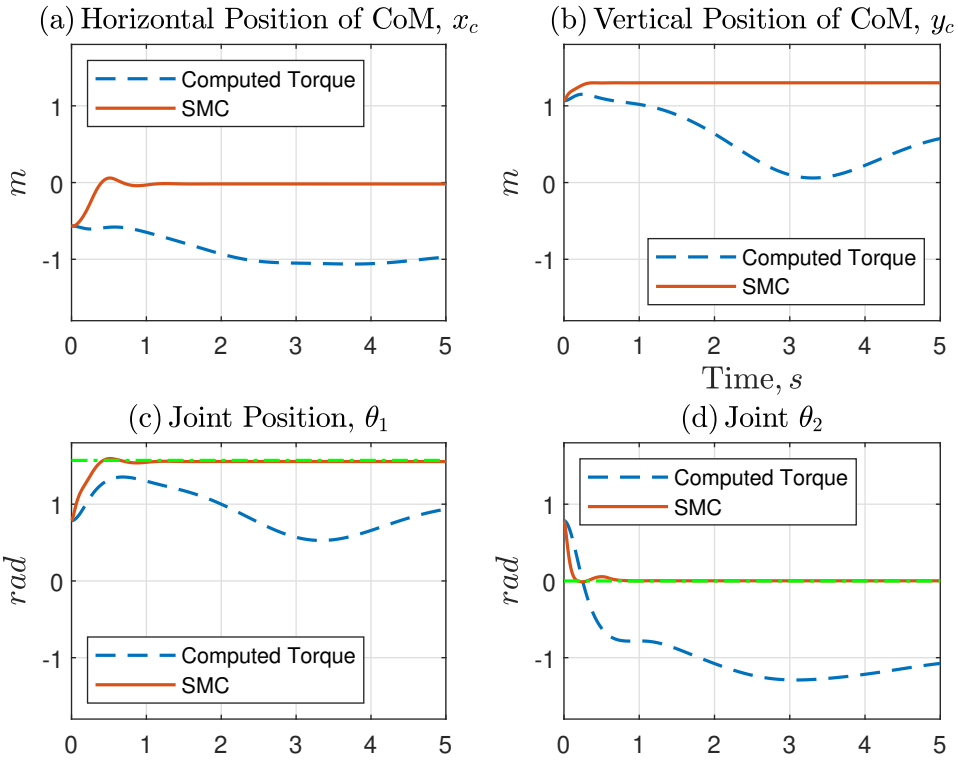


Figure 2.18: Hopping robot in stance phase under parametric uncertainties and external disturbances: (a) Horizontal position of CoM, x_c; (b) Vertical position of CoM, y_c; (c) Joint Position, θ_1; (d) Joint Position, θ_2.

We can see in Fig. 2.19(a) and Fig. 2.19(b) that computed torque control does not guarantee that the CoM velocity converges to zero. It has an oscillatory response. In the SMC controller, the CoM velocity in the x and y axis have a high overshooting and faster response. In Fig.2.19(c) and Fig.2.19(d) we can see that the joints velocities in the SMC controller converge to zero, but peaks are observed during the transient. In the computed torque control, there is a smooth response in the joint speeds. However, there is no convergence in the joints velocity.

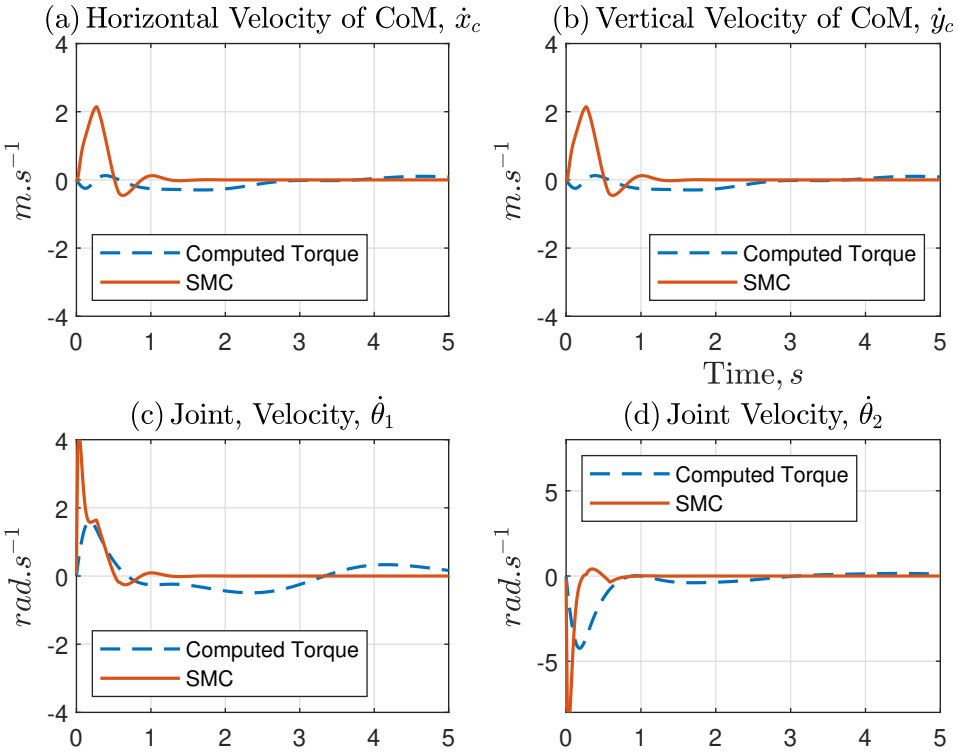


Figure 2.19: Hopping robot in stance phase under parametric uncertainties and external disturbances: (a) Horizontal velocity of CoM, x_c; (b) Vertical velocity of CoM, y_c; (c) Joint velocity, $\dot{\theta}_1$; (d) Joint velocity, $\dot{\theta}_2$.

We can see in Fig.2.20(a) and Fig.2.20 (b) that the control signals τ_1 and τ_2 are limited. The computed torque control has a greater smoothness in the control signals τ_1 and τ_2 small oscillations, whereas the control signals τ_1 and τ_2 have a peak in the instant initials and after overshooting the signals remain constant.

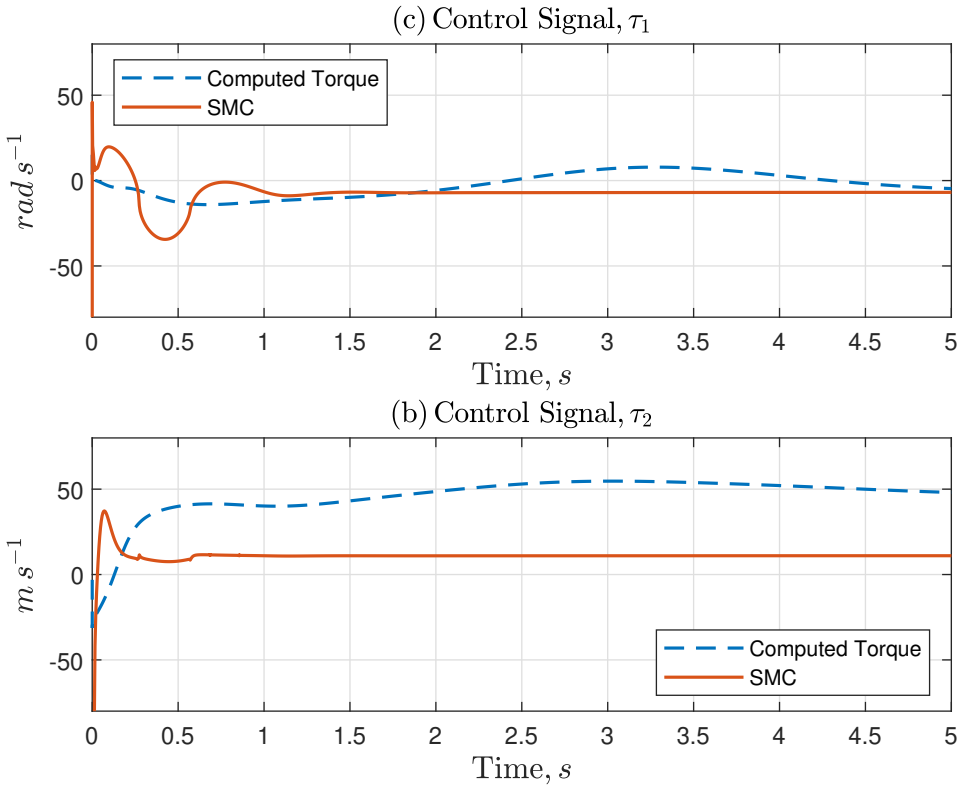


Figure 2.20: Caption

2.6.4.1 Discussion and Analysis

When performing the hopping robot simulations in the support phase using the computed torque controller and SMC, we noticed that the SMC control deals with parametric uncertainties and external disturbances. Despite presenting a faster response, we have as a consequence peaks in the control signals τ_1 and τ_2 due to the necessity to accelerate the body then, that it can reach the equilibrium position quickly. We can also observe that the parameter *lambda* is very important to define whether the robot is in the stance or flight phase. The presented model allows controlling both the stance and flight phases for that it is necessary to change only the initial conditions for each case. It is important to remember that sometimes it is not desired that the norm of the error is null in terrains with irregularities. It is desirable to have an error in the trajectory so that the robot does not become unbalanced.

3
Hexapod Robot: Modelling and Control Design

Although hexapods have advantages as a large balance compared to bipeds, there are several challenges in the legged robot area as balance, navigation in complex terrains and the definition of the sequence of legs so that the robot can move. In the case of these robots, the trajectory to be followed by the legs does not necessarily have to be the same as desired errors in the tracking can occur when, for example, we have an obstacle in the middle of the path in this case, following the desired trajectory becomes impossible. Even if there is an error in the tracking, the robot moves according to desired navigating the terrain.

The solutions to robot leg problems consist of using algorithms that allow robots with legs to move forward, moving to the side and rotate. These algorithms can be artificial intelligence techniques [42], sliding mode control [31]. Define a trajectory so that the robot can navigate in environments with obstacles with the CHOMP algorithm [34]. In this chapter, two controllers will be designed: a Cartesian controller and, SMC controller, which can solve the robot's position problem. However, only the SMC controller can guarantee orientation control despite presenting the chattering phenomenon.

The hexapod robot is an example of a legged mobile robot, which has many applications such as navigate in confined spaces and walking in complex and rough terrains. Examples of hexapod robot are the PhantomX AX Metal Hexapod Mark III from the Trossen robotics in Fig.3.1, the Lauron I created by Forschungszentrum Informatik (FZI) in Fig.3.2 and the Rhex in the Fig.3.3 a hexapod created by five universities (The University of Michigan,McGill University,Carnegie Mellon University,University of California,Princeton University,Cornell University and University of Lahore) financed by DARPA (Defense Advanced Research Projects Agency).

Figure 3.1: Phantomx AX Metal Mark III

Figure 3.2: Hexapod Lauron I source: Wikipedia

Figure 3.3: Rhex 1.1 hexapod source:Wikipedia

3.1 Gait and Stability

The gait planning is the most import part of the legged robot mobile locomotion, influences in the legged robot mobile balance and stability. Besides that, the gait planning is necessary to execute tasks such as climb stairs, walking in the rough terrains and avoid obstacles.

The stability is the main criterion to design the gait most stability criteria are based in the Conservative Support Polygon [2]. The idea behind the Conversation Polygon is shown in Fig 3.4, the point C represent the projection of the centre of mass in the plane, the rectangle of dimensions P and Q represent the leg workspace, notice that the CoM is within support polygon it means the robot is stable.

The leg sequence is designed considering the task which the robot must to

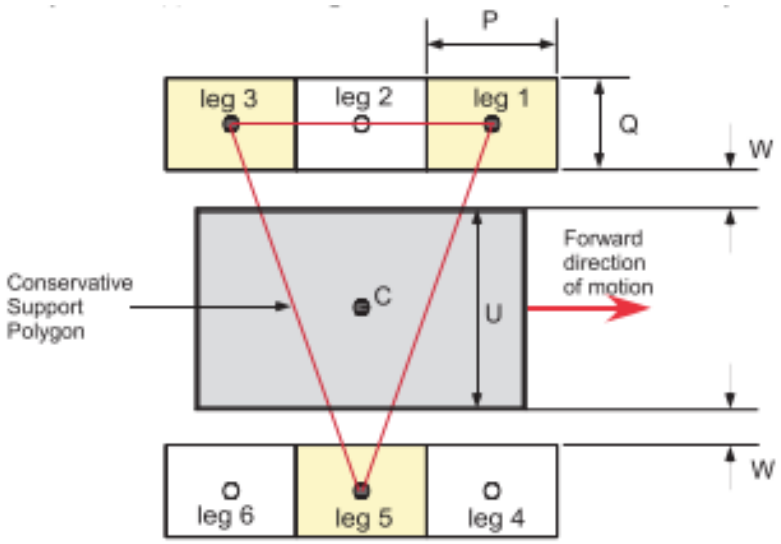


Figure 3.4: Kinematic parameters of hexapod robot [2]

execute such as, walking in slope terrains, walking in rough terrains for stance, the energy consumption, velocity and the hexapod balance. The tripod is a gait the consists in each gait cycle, three leg are in the ground (stance) and others three leg are in the air (swing). Although the advantage of this gait is the velocity, it has the more unstable than wave and quadruped for instance. The tripod sequence is seen in the Fig .3.5 note that, during the gait cycle the CoM is within the support polygon then the tripod gait, in this case, is stable.

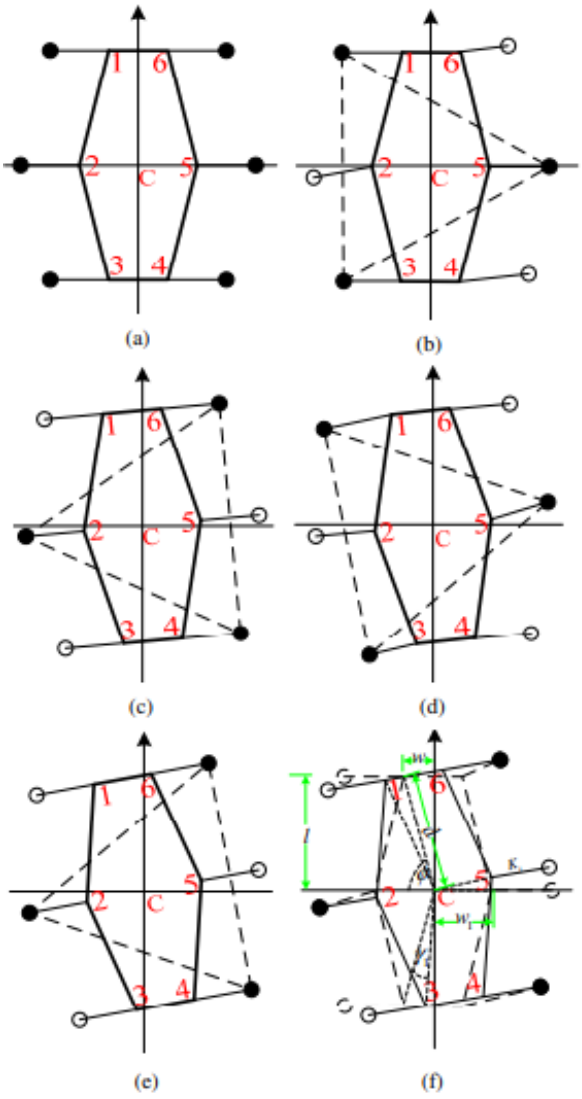


Figure 3.5: The hexapod tripod gait [3]

3.1.1 Control Diagram

The block diagram in the Fig. 3.6 shows the structure which make the robot move, the first part is the joint position controller which can be a PID, sliding mode control, adaptive control, robust control, fuzzy or artificial intelligence techniques such as, reinforcement learning, genetic algorithm for instance. The gait planning is the part which generate the desired joints position and execute a task such as walking forward and backward, rotate, walking side for instance, this algorithm can be inverse kinematic, differential kinematic, fuzzy, reinforcement learning ,neural networks or any gait planning algorithm. The control algorithm is the part responsible for the position and orientation control this part avoid the planning joints necessity.

The cascade control strategy can be applied to control the posture of the

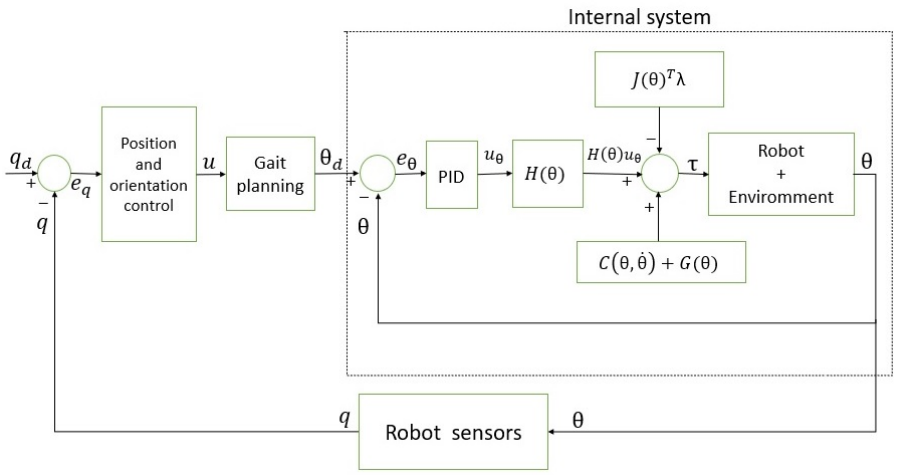


Figure 3.6: Block diagram: cascade control strategy for the hexapod robot

hexapod robot and the performance of the controller would depends on the type of motion carried out by the center of mass (e.g., differential drive, omni-directional, etc.). Another advantage of cascade control is the use of sensor force to implement an orientation control based of the force feedback in order to avoid slippage on the terrain.

3.2 Differential Drive Approach

An approach proposed to the legged mobile robot in Fig.3.7 is modelled by a differential drive if the robot has a kinematic constraint in robot y axis (the crab gait not implemented). The only signals control in the hexapod robot are the linear velocity in the x robot axis and the angular velocity w in the robot z axis. In this case, the kinematic constraint is given by:

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

The modelling of the robot in space is given by the following equation:

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

The velocity in the y axis depends on the linear velocity v and the angular velocity w the equation can decomposed in two vectors g_1 and g_2 given by the

following form:

$$g_1 = \begin{bmatrix} cos(\theta) \\ sin(\theta) \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{3-3}$$

In Eq.3-2 written using the vector g_1 and g_2 results in:

$$\dot{q} = g_1 v + g_2 \omega \,. \tag{3-4}$$

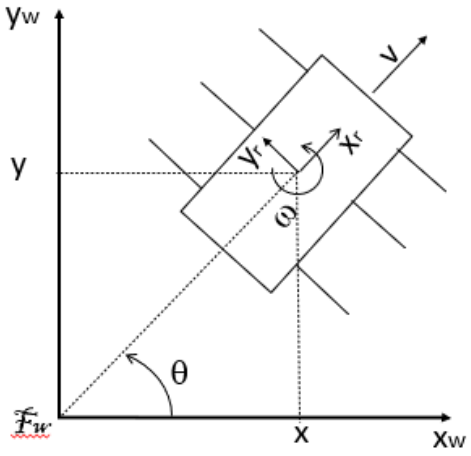


Figure 3.7: Kinematic modeling of the legged mobile robot.

3.2.1 Cartesian Control

Consider that the control goal for a legged mobile robot is to drive the current robot position $p \in \mathbb{R}^2$ to a desired constant position $p_d \in \mathbb{R}^2$, regardless of the robot orientation θ. There are practical situations, named Cartesian regulation, where the objective is simply to reach a desired position no matter the final orientation. For instance, a legged mobile robot exploring an agricultural field must reach a sequence of waypoints and use its onboard sensors (e.g., cameras, LIDAR, or ultrasound) to perceive the characteristics of the environment. If the sensors are attached to the robot structure in such a way that their readings are invariant with respect to direction, then the robot

orientation is not relevant. Then, let us consider the following control goal:

$$p \to p_d\,, \qquad e := p_d - p \to 0\,, \tag{3-5}$$

where the $e \in \mathbb{R}^2$ is the Cartesian position error defined as:

$$e = [\ e_x \quad e_y\ ]^T\,. \tag{3-6}$$

The desired position is the origin the Cartesian system in this case we adopt the following control law:

$$v = -K_1(x\cos\theta + y\sin\theta)\,, \tag{3-7}$$

$$w = -K_2(atan2(e_y, e_x) - \theta + \pi)\,, \tag{3-8}$$

where $K_1 > 0$ and $K_2 > 0$. The driving velocity v is proportional to projection in the Cartesian axis of the displacement vector formed by the current robot position and the desired robot position, the steering velocity ω is proportional to the difference between the vector displacement angle and the robot orientation. Using the following Lyapunov like function:

$$2\,V(e_x, e_y) = x^2 + y^2\,, \tag{3-9}$$

Derivating the Lyapunov like function we have:

$$\dot{V}(e_x, e_y) = x\,\dot{x} + y\,\dot{y}\,, \tag{3-10}$$

$$= x\,v\cos\theta + y\,v\sin\theta\,, \tag{3-11}$$

$$= -K_1(x\ \cos\theta\,(x\cos\theta + y\sin\theta) + y\ \sin\theta(x\cos\theta + y\sin\theta))\,, \tag{3-12}$$

$$= -K_1(x\ \cos\theta + y\ \sin\theta)^2\,. \tag{3-13}$$

The function $\dot{V}$ is a negative semi definite function then we prove that the robot position converges to origin, using the Barbalat's Lemma we have:

$$\ddot{V} = -2\,K_1(x\ \cos\theta + y\ \sin\theta)^2 - 2\,K_1\,(y\ \cos\theta - x\ \sin\theta)\,\omega\,. \tag{3-14}$$

$\ddot{V}$ is bounded then, the position error converges to zero, the Lyapunov like does not provides any information about the orientation so we do not say anything about it, we conclude that, the orientation assume any value.

3.2.2 Sliding Mode Control

The Cartesian control is a usable controller when the goal is the position control, in situations where we need to control not only the position but also the orientation the Cartesian control can not satisfy this necessity. An example is a situation where the camera is fixed in the robot in this situation the orientation control is important a solution proposed for this problem to use the sliding mode control. The purpose of using the SMC control for position and orientation control and its ability to deal with unmatched disturbances is not possible with the Backstepping Control. The Polar Control ensures position and orientation control but, the singularity present in its control law can be a problem when we want to take the robot to a specific position and orientation and the it can not to deal with unmodeled dynamics. The chattering phenomenon is a limitation in the SMC controller which, degrades the controller accuracy and may excite unmodeled dynamics, which undermines the system performance and may cause instability [55, 56]. To use sliding control as proposed in the section, first, we can use the chained form transformation, we use the transformation in Eq .3-15 [54]:

$$z_1 = e_\theta \,, \tag{3-15}$$

$$z_2 = (x - x_d)\,\cos\theta + (y - y_d)\,\sin\theta \,,$$

$$z_3 = (x - x_d)\,\sin\theta - (y - y_d)\,\cos\theta \,, \tag{3-16}$$

$$\tag{3-17}$$

and the following input transform:

$$u_1 = \omega \,, \tag{3-18}$$

$$z_2 = u_2 + z_3\, u_1 \,, \tag{3-19}$$

which results in:

$$\dot{z}_1 = u_1 \,, \tag{3-20}$$

$$\dot{z}_2 = u_2 \,, \tag{3-21}$$

$$\dot{z}_3 = z_2\, u_1 \,. \tag{3-22}$$

The desired coordinates are $q_d = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$, using the following control law:

$$u_1 = -k\, z_1 - \alpha\, z_2\, sign(s)\, |s|^p\,, \tag{3-23}$$

$$u_2 = -k\, z_2 + \alpha\, z_1\, sign(s)\, |s|^p\,, \tag{3-24}$$

where $k \in \mathbb{R}$ is a proportional control gain, $\alpha \in \mathbb{R}$ is a robustness control gain, $0 < p \leq 1/2$ is a term responsible for attenuating the chattering effect, $s \in \mathbf{R}$ is the sliding surface defined by:

$$s = -2\, z_3 + z_1\, z_2\,, \tag{3-25}$$

We can prove that the system reach the desired position and orientation, using the following theorem below:

Teorema 3.1 *Consider a nonholonomic system (3-2) and the stabilizing control law in (3-23) with sliding surface (3-25). Then, the following stability properties hold: (i) all signals of the overall closed-loop system are bounded; (ii)* $\lim_{t\to\infty} z_1(t), z_2(t) = 0$*; (iii)* $\lim_{t\to\infty} s(t) = 0$ *and, consequently,* $\lim_{t\to\infty} z_3(t) = 0$.

Prova. The first step is to use the following Lyapunov like function and prove the e_{z1} and e_{z2} stability:

$$2V(z_1, z_2) = z_1^2 + z_2^2\,, \tag{3-26}$$

$$\dot{V} = z_1\, u_1 + z_1\, u_2\,, \tag{3-27}$$

$$\dot{V} = -k\, (z_1^2 + z_2^2)\,, \tag{3-28}$$

Now we have prove that the sliding surface can reached ,using the following Lyapunov function:

$$2V(s) = s^2\,, \tag{3-29}$$

$$\dot{V} = s\, \dot{s}\,, \tag{3-30}$$

$$= s\, (2\, z_2\, u_1 - z_2 u_1 - z_1\, u_2)\,, \tag{3-31}$$

$$= s(z_2\, u_1 - z_1\, u_2) \tag{3-32}$$

$$= -\alpha(z_1^2 + z_2^2) sign(s) s |s|^p \tag{3-33}$$

$$= -\alpha(z_1^2 + z_2^2) |s|^{p+1}\,. \tag{3-34}$$

■

Notice that, in Eq .3-23, the proportional term ensures the convergence of z_1 and z_2 to zero whereas the robustness term is responsible for the convergence

of s and hence z_3 to zero. The Lyapunov like function proves that z_1 and z_2 converge to zero and, notice that the second part of control law u_1 and u_2 does not affect the stability prove and ensures which z_3 converges to zero but, if the α is a high value, the system is not as asymptotically stable. Choosing the Lyapunov function for the surface: The Lyapunov $V(s)$ shows that the sliding surface converges to zero and, the Lyapunov like function shows that z_1 and z_2 converges to zero then z_3 converges to zero.

3.2.3 Simulation and Results

Here we run the simulations in the Gazebo, using a hexapod PhantomX robot. The graphics simulations were implemented using ad-hoc developed Gazebo scripts and functions, running on the Ubuntu 18, Linux Enterprise, 64-bit OS using Intel (R) Core (TM) i5-8250U CPU @ 1.6GHz, 8GB DDR4 RAM.

The purpose of the simulations is to check the position and orientation control of the Cartesian and SMC controls. The joints of the legs are controlled by means of an internal PID controller that had its gain adjusted to the following values: $k_p = 100$, $k_d = 10$ and $k_I = 0.01$. We can see the Gazebo interface in Fig 3.8.

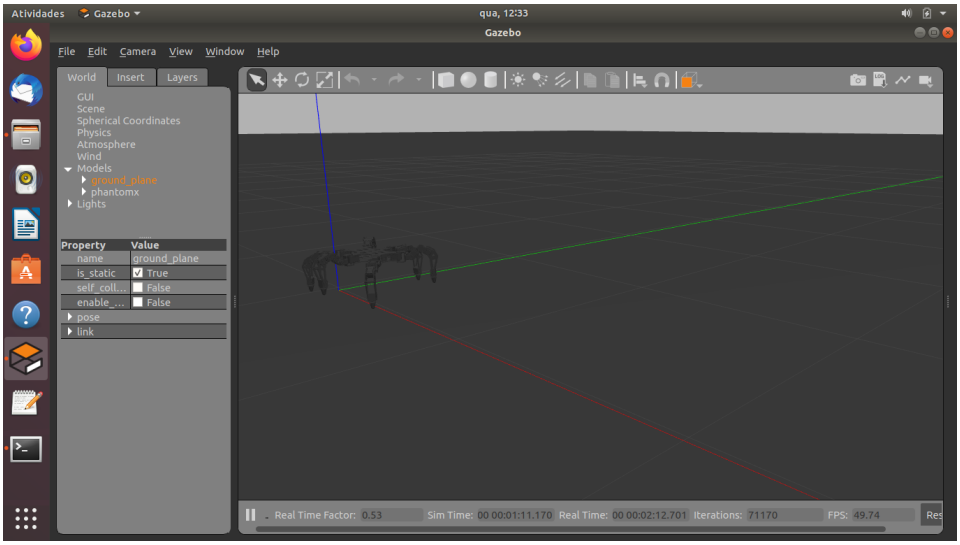


Figure 3.8: PhantomX Hexapod in Gazebo Interface

The gait used for locomotion of the hexapod is the TRIPOD the choice of this sequence of legs is because, in addition to desiring faster locomotion, it also allows the robot to always be stable because the robot's centre of mass is always inside the support polygon.

The desired joints positions for the robot to perform its trajectory are known and the robot itself already has a PID controller for the control of the joints that have already been configured with the defined gains. In the case of robots with legs, following the desired trajectory of the robot's legs is not relevant, because even if there is an error in tracking, we can still guarantee the robot's locomotion. To make the robot move, the sequence of legs was associated with the commands to perform translation and rotation of the robot so that it can move in the plane. When the control loop is closed, the

designed controller performs position control in the case of Cartesian control and position and orientation control in the case of SMC control.

In the figure 3.9, you can see the robot moving frame by frame towards the origin in the Gazebo.The slowness in the robot trajectory in Gazebo was due to the low relationship between the simulation time and the real-time. This relationship occurs because of the low computational capability of the computer used in the simulation. Therefore, we will observe the slow convergence in the graphics.

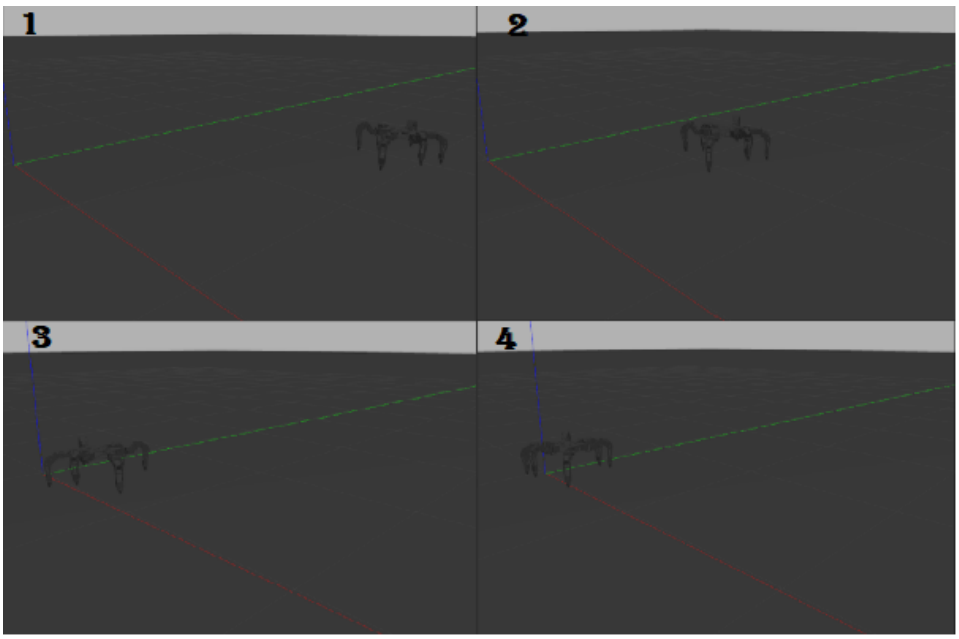


Figure 3.9: Hexapod Robot Moving Frame to Frame

The purpose of this section is to validate the Cartesian control and the SMC controller. We will simulate the SMC controller under unmodeled dynamic and noise measurement in odometry.

3.2.4 Cartesian Controller

We tried using a Cartesian control to take the robot to its origin. We tested the controller for the four quadrants of the plane. To do this, we adjust the controller's earnings to $k_1 = k_2 = 0.4$. The key idea of testing the controller for quadrants is to see if it can reach the origin of the Cartesian plane from different quadrants or if there are quadrants where malfunctions occur.

3.2.4.1
Case I: First Quadrant

Here we check the Cartesian controller in the first quadrant, we chose $x = 2m$, $y = 2m$ and $\theta = 0rad$ as initial position and orientations. As noted in Fig.3.10 the robot's trajectory reaches the origin of the plane. In Fig.3.11. We observe that the position errors converge to zero. On the other hand, the orientation error converges to values between 2 and $3rad$ and the control signals v has a large overshooting and has a poor transient and w converges to zero faster, both control signals are bounded.

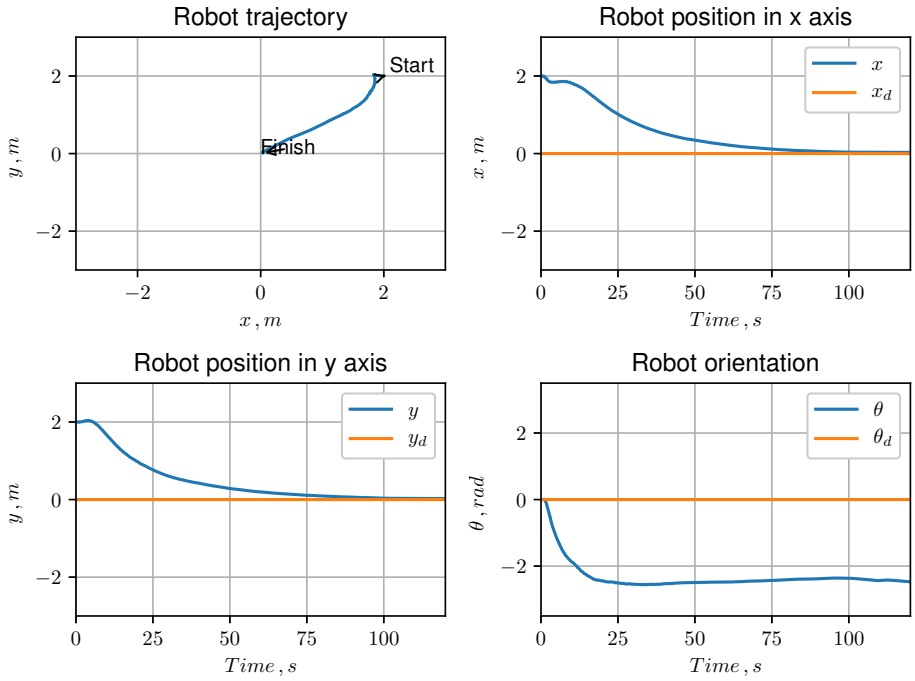


Figure 3.10: Position and Orientation of Hexapod in First Quadrant

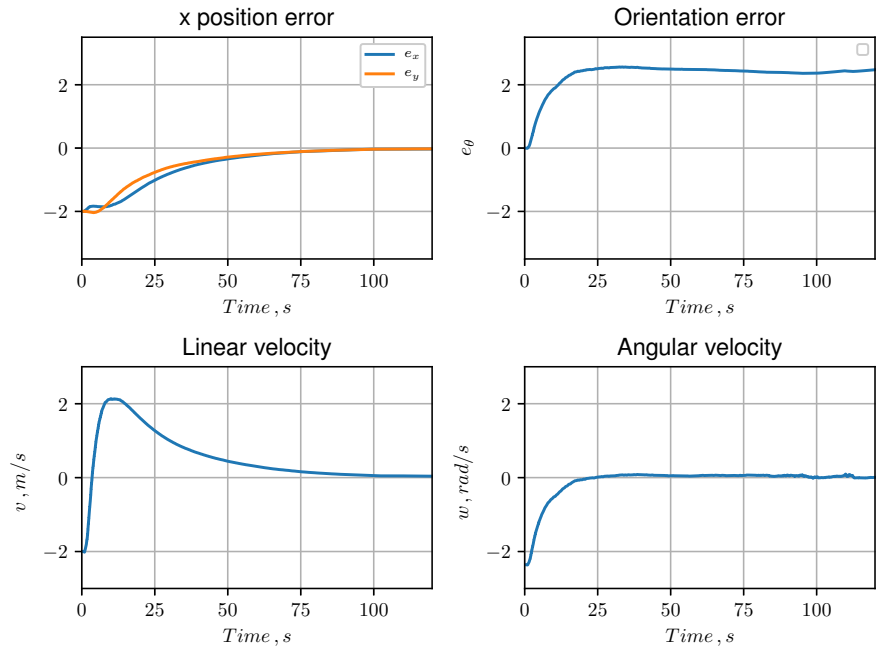


Figure 3.11: Errors Coordinates and Control of Hexapod in First Quadrant

3.2.4.2 Case II: Second Quadrant

Now, we check the Cartesian controller in the second quadrant, we chose $x = -2m$, $y = 2m$ and $\theta = 0rad$ as initial position and orientations. As noted in Fig.3.12 the robot's trajectory reaches the origin of the plane. In Fig.3.13. We observe that the position errors converge to zero. On the other hand, the orientation error converges to values between -3 and $-2rad$ and the control signals v has a small overshooting and a poor transient w has a faster convergence and both control signals are bounded.

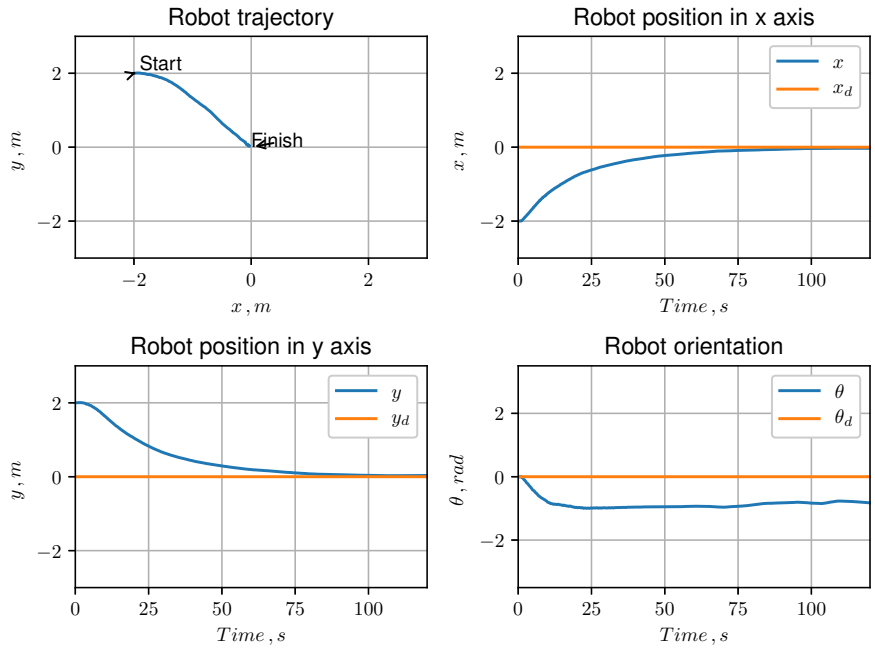


Figure 3.12: Position and Orientation of Hexapod in Second Quadrant

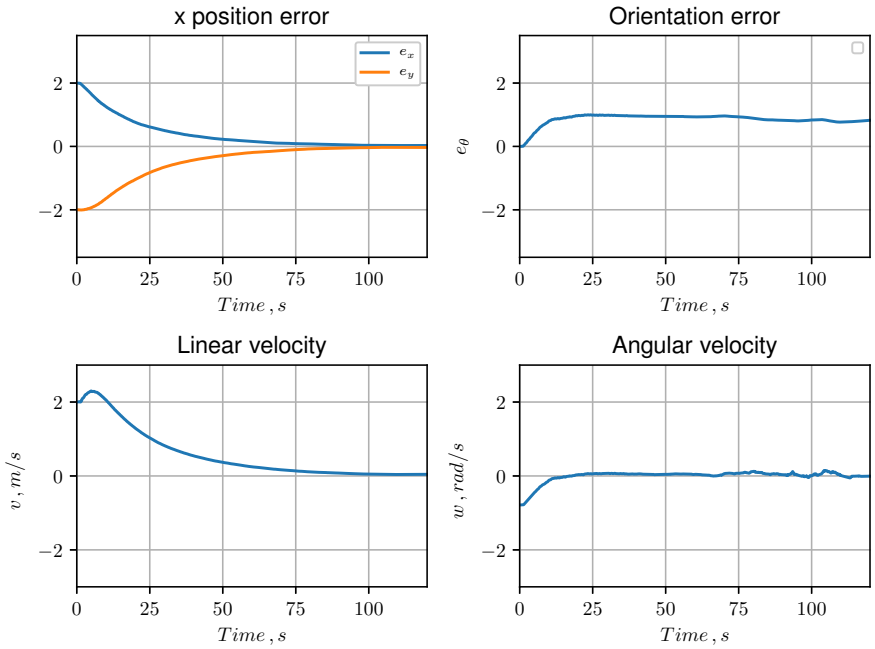


Figure 3.13: Errors Coordinates and Control of Hexapod in Second Quadrant

3.2.4.3 Case III: Third Quadrant

Here we check the Cartesian controller in the third quadrant, we chose $x = -2m$, $y = -2m$ and $\theta = 0rad$ as initial position and orientations. As noted in Fig.3.10 the robot's trajectory reaches the origin of the plane. In Fig.3.11. We observe that the position errors converge to zero. On the other hand, the orientation error converges to values between 2 and 3 and the control signals v has an small overshooting and a slow convergence and w has an faster convergence and both control signals are bounded.

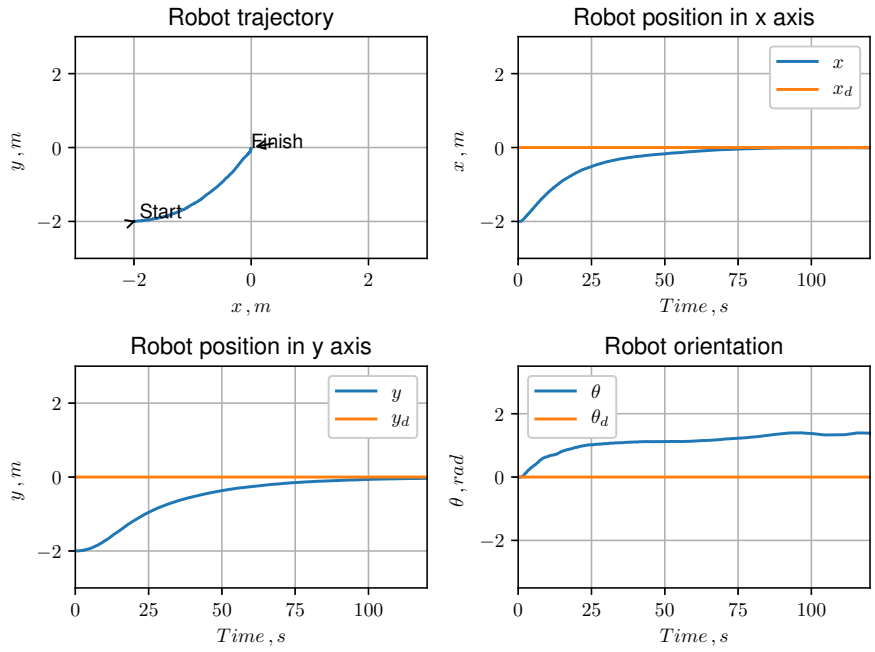


Figure 3.14: Position and Orientation of Hexapod in Third Quadrant

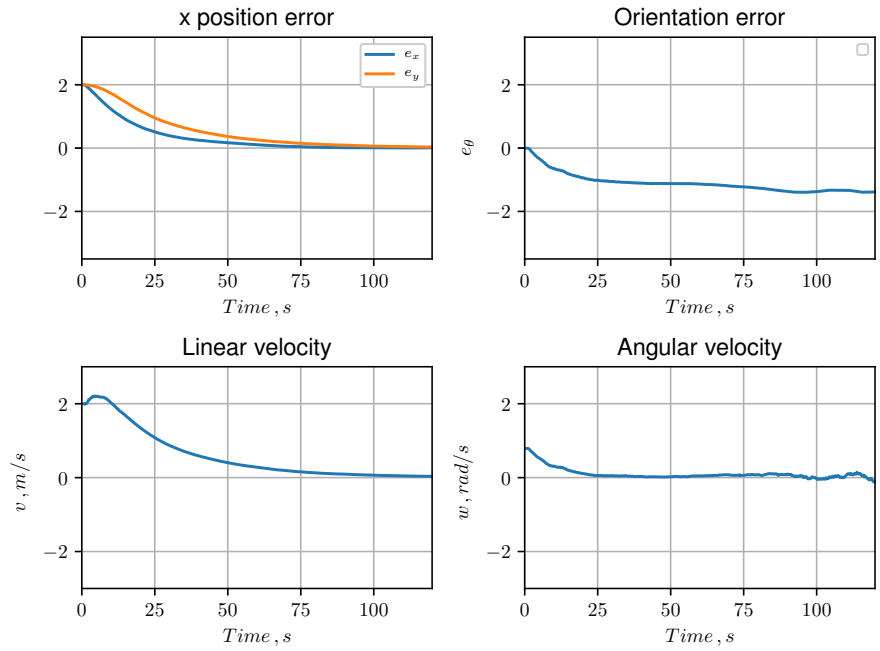


Figure 3.15: Errors Coordinates and Control of Hexapod in Third Quadrant

3.2.4.4 Case IV: Fourth Quadrant

The next step is check the Cartesian controller in the fourth quadrant, we chose $x = 2m$, $y = -2m$ and $\theta = 0rad$ as initial position and orientations. As noted in Fig.3.16 the robot's trajectory reaches the origin of the plane. In Fig.3.17. We observe that the position errors converge to zero. On the other hand, the orientation error converges to values between 2 and $3rad$ and the control signals v has a small overshooting a low convergence and w has a faster convergence and both control signals are bounded.

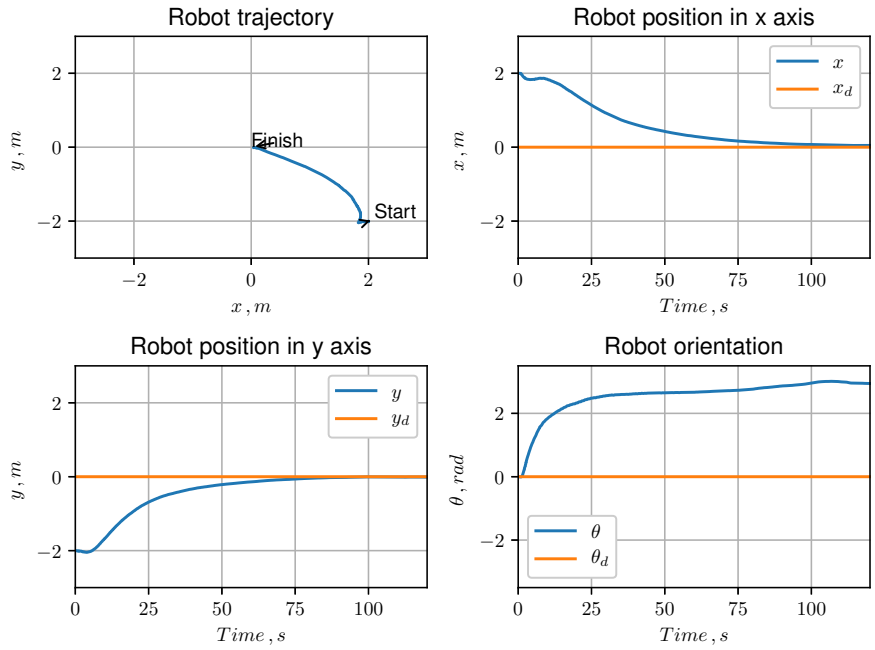


Figure 3.16: Position and Orientation of Hexapod in the Fourth Quadrant

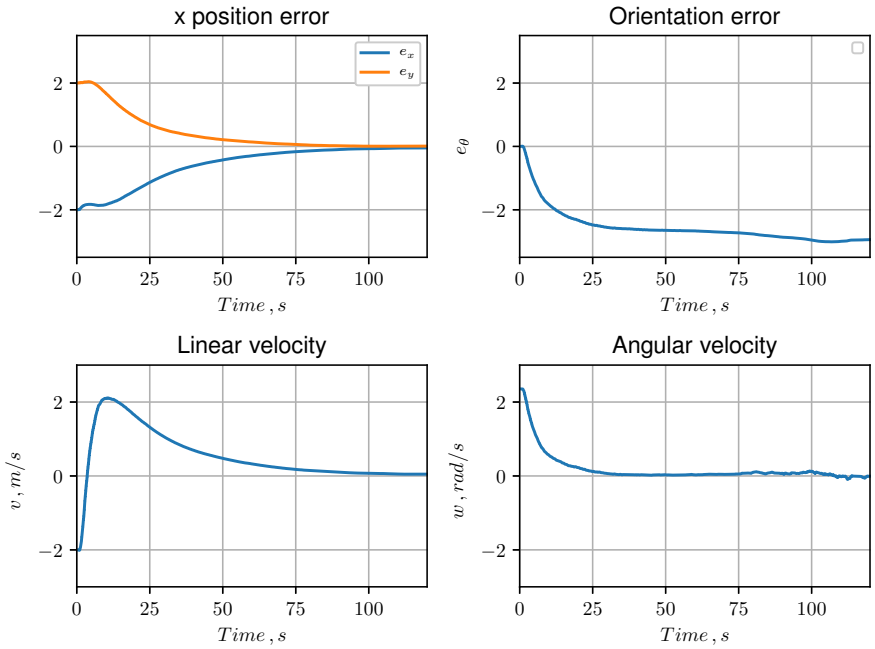


Figure 3.17: Error Coordinates and Control of Hexapod in Fourth Quadrant

3.2.4.5 Discussion and Analysis

We can observe in the simulations that the Cartesian control can not control the robot orientation. However, if the problem is only of position control, the Cartesian controller drives the robot to the desired position. The control signal v has a high overshooting when the robot is in the first and fourth quadrants and low overshooting when in the second and third quadrants we notice that the control signal acts for a long time on the robot. On the other hand, w presents a faster response when it stops actuating quickly on the robot. That is the reason why explains why the robot's trajectory is a straight line towards the origin.

3.2.5 Sliding Mode Controller

Here we analyze the results of the SMC controller in the position and orientation control of the PhantomX hexapod. We simulate the controller, choosing the initial conditions in each quadrant of the Cartesian plane to check the controller performance, unmodeled dynamics and, inherent noise from the odometry sensor are considered in the simulation. We use the following unmodel dynamic:

$$H(s) = \frac{k_s}{1 + s\lambda}\,, \tag{3-35}$$

where, $k_s = 1$, $\lambda = 0.2$, the cut-off frequency is $\omega = 1/\lambda = 5rad/s$, $k = 0.1$ and $\alpha = 0.2$.

3.2.5.1 Case I: First Quadrant

Here we simulate the robot in the first quadrant, we choose $x = 2m$, $y = 2m$ and $\theta = 0rad$ as initial conditions. In Fig. 3.18, it is possible to observe that the trajectory has a discontinuity that occurs when the robot stops and adjusts its orientation. Notice that the robot reaches the position x and y in the Cartesian plane. When observing the orientation, we perceive an oscillation and a poor transient. However, we observe that the orientation converges to the desired value. In Fig. 3.19 it is observed that the control signals v and w are bounded even with peaks and oscillations, and the sliding surface also presents an oscillatory transient response.

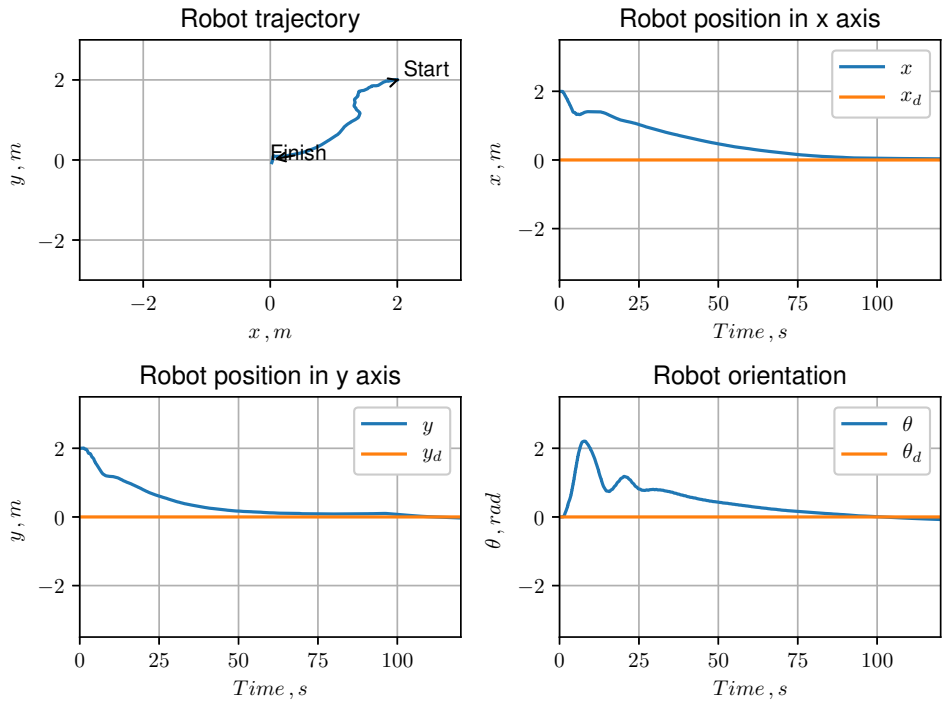


Figure 3.18: Position and orientation in the first quadrant: Unmodeled Dynamic and noise measurement in odometry

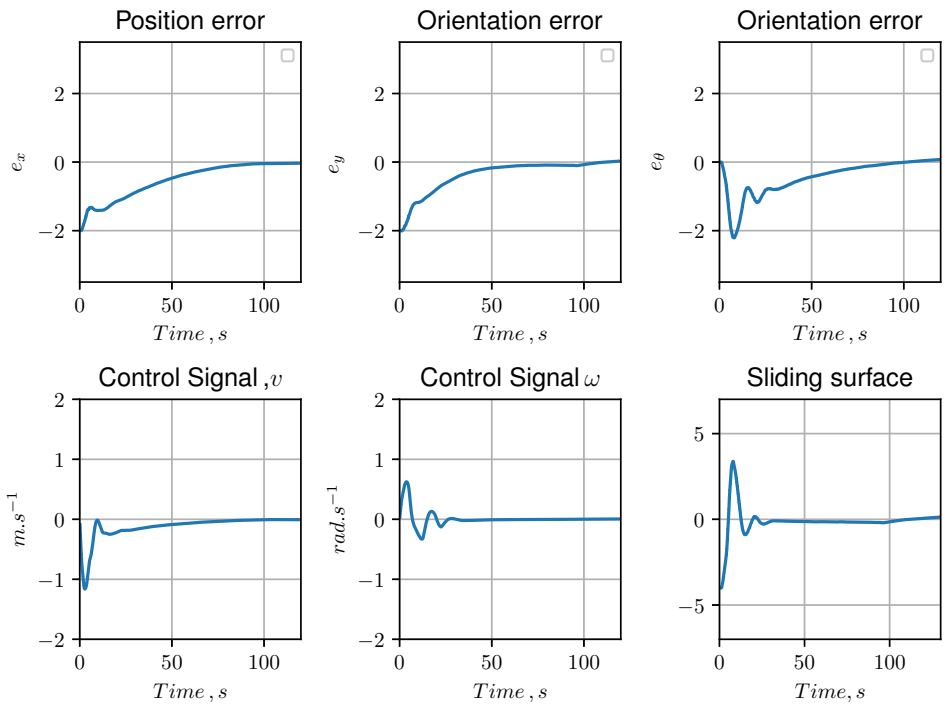


Figure 3.19: Position and orientation Errors, control signals and sliding surface in the first quadrant: Unmodeled dynamic and noise measurement in odometry

3.2.5.2 Case II: Second Quadrant

Now, we simulate the hexapod in the second quadrant it can be seen in Fig. 3.20 the trajectory is discontinuous, which means that the robot is stopped for a long time this can be seen in the position graphs where x and y remain constant for a while. The orientation also converges to zero, but there is an oscillation in the response due to unmodeled dynamics. In Fig. 3.21 we observe that the control signals v and ω and the sliding surface are bounded and have oscillate during the transient regime.

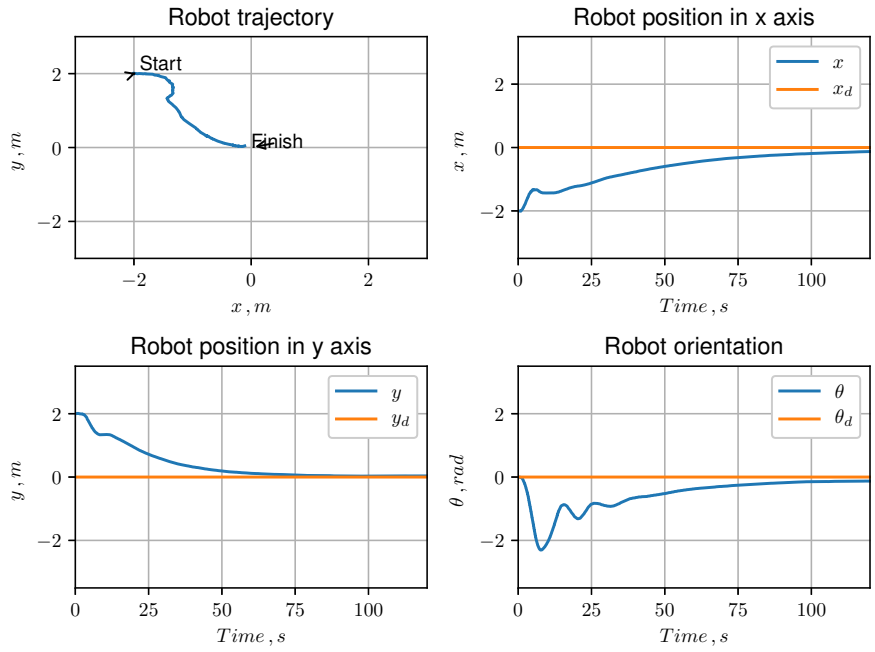


Figure 3.20: Position and orientation in the second quadrant: Unmodeled Dynamic and noise measurement in odometry

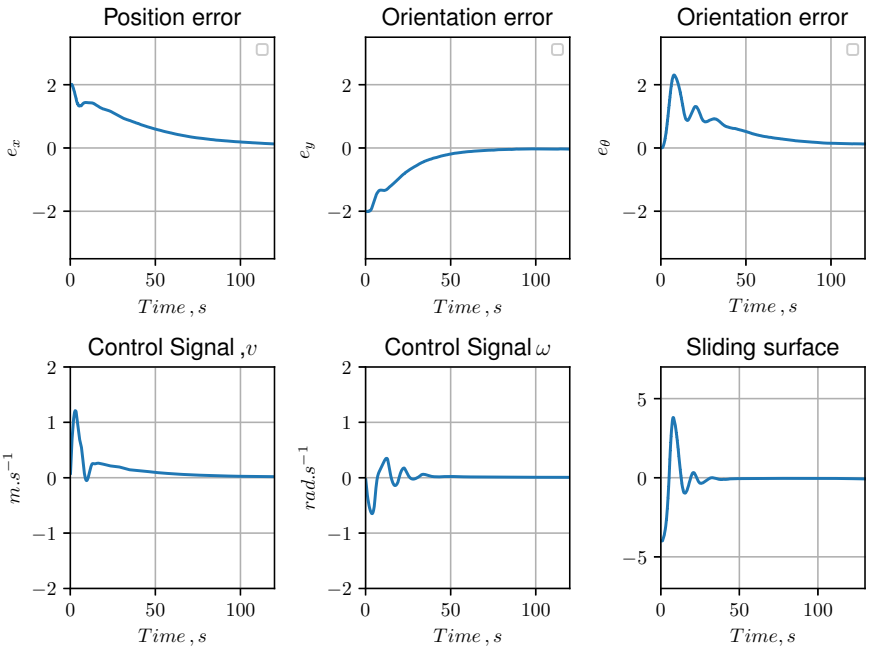


Figure 3.21: Position and orientation Errors, control signals and sliding surface in the second quadrant: Unmodeled dynamic and noise measurement in odometry

3.2.6 Case III: Third Quadrant

The next step is simulate the hexapod in the third quadrant it can be seen in Fig. 3.22 the trajectory is discontinuous, which means that the robot is stopped for a long time this can be seen in the position graphs where x and y remain constant for a while. The orientation also converges to zero, but there is an oscillation in the response due to unmodulated dynamics. In Fig. 3.23 we observe that the control signals v and ω and the sliding surface are bounded and has oscillate during the transient regime.

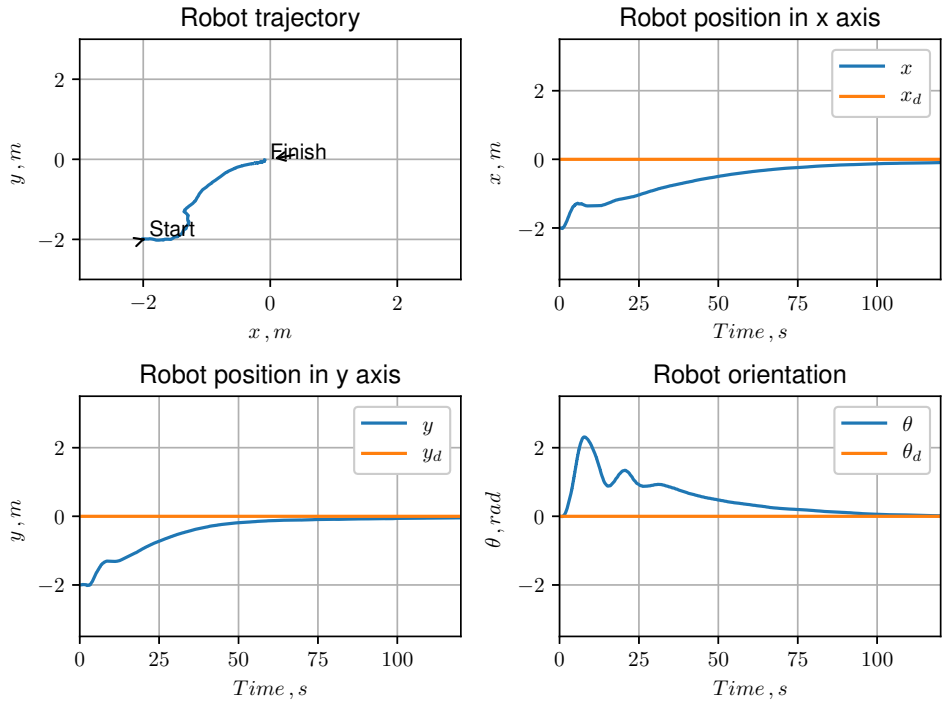


Figure 3.22: Position and orientation in the third quadrant: Unmodeled Dynamic and noise measurement in odometry

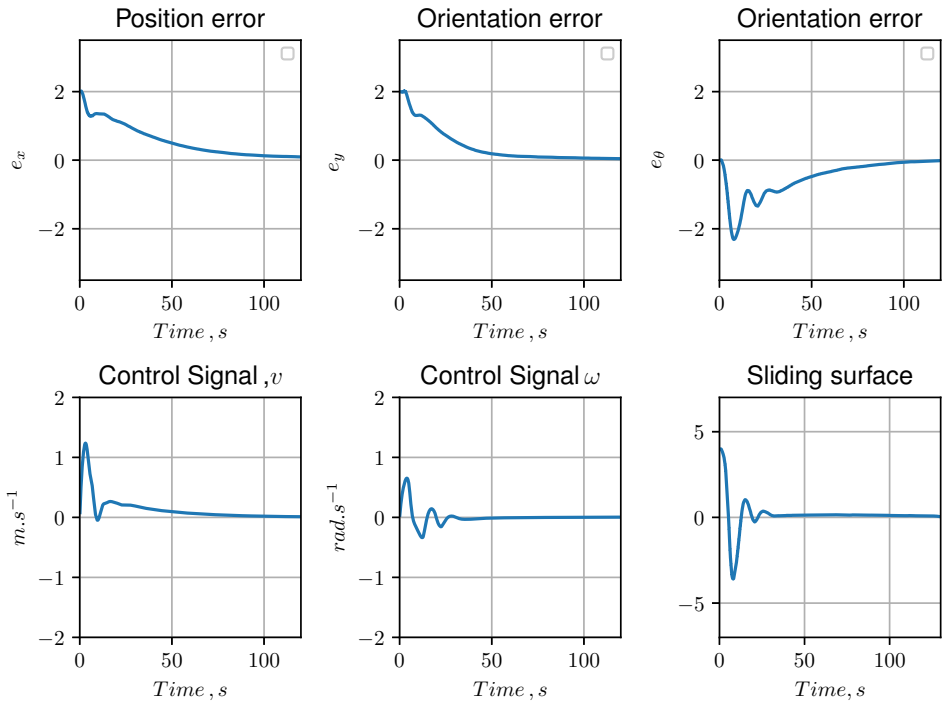


Figure 3.23: Position and orientation Errors, control signals and sliding surface in the third quadrant: Unmodeled dynamic and noise measurement in odometry

3.2.6.1 Case IV: Fourth Quadrant

Here we simulate the hexapod in the fouth quadrant it can be seen in Fig. 3.24 the trajectory is discontinuous, which means that the robot is stopped for a long time this can be seen in the position graphs where x and y remain constant for a while. The orientation also converges to zero, but there is an oscillation in the response due to unmodulated dynamics. In Fig. 3.25 we observe that the control signals v and ω and the sliding surface are bounded and has oscillate during the transient regime.

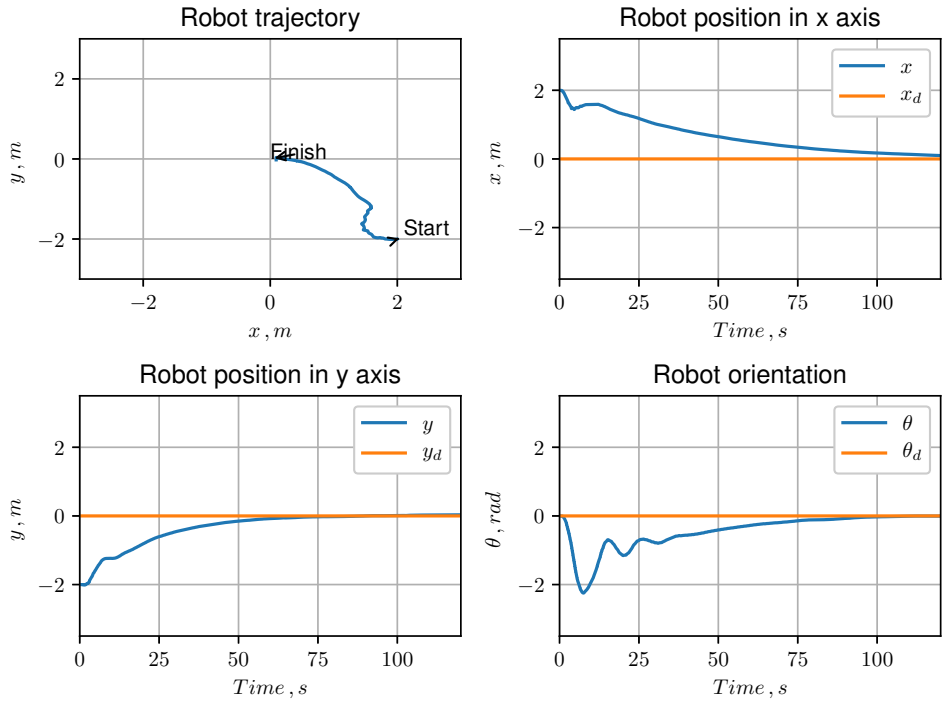


Figure 3.24: Position and orientation in the fourth quadrant: Unmodeled Dynamic and noise measurement in odometry

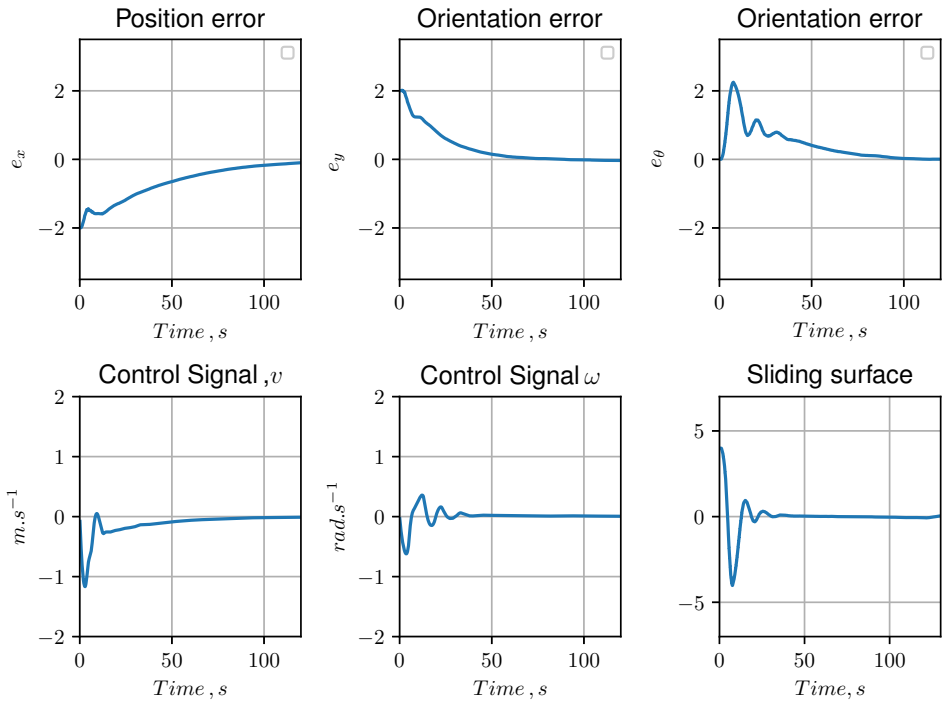


Figure 3.25: Position and orientation Errors, control signals and sliding surface in the fourth quadrant: Unmodeled dynamic and noise measurement in odometry

3.2.7 Discussion and Analysis

When testing the Cartesian and SMC controllers for the hexapod it was noticed that both manage to guarantee the position control of the robot, but only the SMC was also able to guarantee the orientation control, which depending on the task to be performed may be useful or not. Both the mathematical statements and the graphs clearly show the limitations of Cartesian control, since the orientation of the robot can take on any value. On the other hand, the SMC control for being able to guarantee control of both position and orientation also has limitations regarding its use, such as chattering which, in addition to causing warming, can cause stability problems when dealing with unmodeled dynamics, a solution to mitigate chattering is to use a term $|s|^p$ by multiplying sign(s). It can be seen, that even with the measurement noise in the odometry and with unmodeled dynamics, the hexapod managed to reach the desired position with the desired orientation the gains also greatly influenced the SMC. Due to the simplicity of the model used for the robot, the SMC is not capable to perform tasks such as climbing stairs because, the model does not have information about roughness and inclination terrain, which makes the control action impossible, which is a

practical limitation.

The use of both controllers depends on the objective to be achieved, if we only want to control the position, the Cartesian controller can meet this objective, now if the objective is orientation control, the SMC controller can meet this objective.

4
Concluding Remarks and Perspectives

In this work we have addressed the modeling and control design, based on the sliding mode control approach, for a class of legged mobile robots. Two kinematic models for the hopping robot, in flight and stance phases, was considered: a single mechanism with revolute and prismatic joints and a complex one with only revolute joints. Two robust controllers based on sliding modes were investigated: a first-order sliding mode control with attenuated chattering effect and a combination of finite-time and super-twisting algorithms. The dynamic model of the hopping was also studied for the stance phase. Two different controllers were implemented: a computed-torque based controller and a first-order sliding mode control with attenuated chattering effect. Finally, the modeling and control of a multi-legged robot was also considered using a cascade control strategy for a hexapod robot. Two controllers were studied to evaluate the robot behavior for regulation tasks: the Cartesian regulation control and the first-order sliding mode control. The Lyapunov stability theory was employed to demonstrate the stability and robustness properties of the sliding mode controllers. Numerical simulations in Matlab were carried out to evaluate the performance and feasibility of the proposed controllers for stabilization of the hopping robots, in both phases. The Gazebo, an open-source 3D robotics simulator, were used to run graphical simulations and analyse the motion sequence for the legs of the hexapod robot. The Robot Operating System (ROS) framework was adopted to develop the ROS nodes, which were responsible for robot motion, posture control and data acquisition.

4.1
Analysis of the Results

After all, we conclude that the sliding mode control proposed to solve the first hopping robot problem, worked even considering the high order terms present in the model. As for super-twisting and planning using Fourier analysis fails to deal with non linearity, as in the case of the first hopping robot presented in Chapter 2.

The approach of the articulated hooping robot was not very simple because, when carrying out the transformation in the chained form, the

robot parameters appear in the equations it was also not very advantageous. After all, the fact that the parameters appear in the chained form makes it difficult to simulate the problem from a computational point of view. The difficulties in simulation happen because the input transformation of chained form depends on the robot parameters, then depending on the adjustment in the robot parameters yields problems in simulation. The controllers proposed for the bouncing robot presented in the support phase are able, within their limitations, to make the joints reach the desired position, the advantage of the model in the control design is to allow the control of the phases of stance and flight and to detecting one of the phases without the necessity of a force sensor in the foot, the disadvantage is the difficulty in thinking about the trajectory in space because it is difficult to describe the CoM trajectory in the Cartesian plane using only the joint space and vice versa.

The choice of the cascade control strategy proved to be useful for the control of the hexapod it managed to guarantee the position and orientation of the robot. The simulations in the Gazebo proved that the control strategy managed to get the robot to reach the desired position and managed to guarantee the orientation of the robot in the case of SMC.

4.2 Future Works

There is still a lot to be explored on robots with legs, for future work we intend to:

- Perform force control so that the robot can walk on inclined places or climb stairs. The aim of the idea is to explore the advantages of robots with legs.
- Include a camera to the robot and use artificial intelligence techniques for image processing.
- Simulate the hexapod presented in an agricultural environment of the Gazebo, as this is a very interesting application and shows an application of the subject.
- Build a robot with real legs and implement the codes developed for the hexapod simulation. As it is a very interesting topic, the construction of a real robot would be a way to apply the developed knowledge.
- Deepening in the analysis and study of bipedal robots, therefore, there is still much to be explored in this theme.

Bibliography references

[1] **ICRA 2019 3rd Full-Day Workshop on Legged Robots**. https://icra2019wslocomotion.wordpress.com/. Accessed: 2020-10-05.

[2] REKLEITIS, G.; VIDAKIS, M. ; PAPADOPOULOS, E.. **Optimal leg sequencing for a hexapod subject to external forces and slopes**. Proceedings - IEEE International Conference on Robotics and Automation, p. 6302–6308, 2019.

[3] DUAN, X.; CHEN, W.; YU, S. ; LIU, J.. **Tripod gaits planning and kinematics analysis of a hexapod robot**. In: 2009 IEEE INTERNATIONAL CONFERENCE ON CONTROL AND AUTOMATION, p. 1850–1855. IEEE, 2009.

[4] **Why ROS Topic should get XMLRPC address:port but ROS Service doesn't?** https://answers.ros.org/question/12882/why-ros-topic-should-get-xmlrpc-addressport-but-ros-service-doesnt/. Accessed: 2020-10-05.

[5] HÄGELE, M.; NILSSON, K.; PIRES, J. N. ; BISCHOFF, R.. **Industrial Robotics**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1385–1422. Springer International Publishing, 2016.

[6] VAN DER LOOS, H. M.; REINKENSMEYER, D. J. ; GUGLIELMELLI, E.. **Rehabilitation and Health Care Robotics**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1685–1728. Springer International Publishing, 2016.

[7] 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.

[8] TREVELYAN, J.; HAMEL, W. R. ; KANG, S.-C.. **Robotics in Hazardous Applications**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1521–1548. Springer International Publishing, 2016.

[9] YOSHIDA, K.; WILCOX, B.; HIRZINGER, G. ; LAMPARIELLO, R.. **Space Robotics**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1423–1462. Springer International Publishing, 2016.

[10] TAYLOR, R. H.; MENCIASSI, A.; FICHTINGER, G.; FIORINI, P. ; DARIO, P.. **Medical Robotics and Computer-Integrated Surgery**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1657–1684. Springer International Publishing, 2016.

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

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

[13] 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.

[14] VILLANI, L.; DE SCHUTTER, J.. **Force Control**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 195–220. Springer International Publishing, 2016.

[15] BEETZ, M.; CHATILA, R.; HERTZBERG, J. ; PECORA, F.. **AI Reasoning Methods for Robotics**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 329–356. Springer International Publishing, 2016.

[16] PETERS, J.; LEE, D. D.; KOBER, J.; NGUYEN-TUONG, D.; BAGNELL, J. A. ; SCHAAL, S.. **Robot Learning**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 357–398. Springer International Publishing, 2016.

[17] BILLARD, A. G.; CALINON, S. ; DILLMANN, R.. **Learning from Humans**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1995–2014. Springer International Publishing, 2016.

[18] FITZPATRICK, P.; HARADA, K.; KEMP, C. C.; MATSUMOTO, Y.; YOKOI, K. ; YOSHIDA, E.. **Humanoids**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1789–1818. Springer International Publishing, 2016.

[19] IIDA, F.; IJSPEERT, A. J.. **Biologically Inspired Robotics**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 2015–2034. Springer International Publishing, 2016.

[20] WIEBER, P.-B.; TEDRAKE, R. ; KUINDERSMA, S.. **Modeling and Control of Legged Robots**. In: Siciliano, B.; Khatib, O., editors, SPRINGER HANDBOOK OF ROBOTICS, p. 1203–1234. Springer International Publishing, 2016.

[21] SCHWARZ, M.; RODEHUTSKORS, T.; SCHREIBER, M. ; BEHNKE, S.. **Hybrid driving-stepping locomotion with the wheeled-legged robot momaro**. 2016 IEEE International Conference on Robotics and Automation (ICRA), p. 5589–5595, 2016.

[22] RAIBERT, M.; BLANKESPOOR, K.; NELSON, G. ; PLAYTER, R.. **BigDog, the Rough-Terrain Quadruped Robot**. IFAC Proceedings Volumes: 17th IFAC World Congress, p. 10822–10825, 2008.

[23] BERMUDEZ, F. L. G.; JULIAN, R. C.; HALDANE, D. W.; ABBEEL, P. ; FEARING, R. S.. **Performance Analysis and Terrain Classification for a Legged Robot over Rough Terrain**. In: 2012 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, p. 513–519, Vilamoura, Portugal, 2012.

[24] HYON, S.; MITA, T.. **Development of a biologically inspired hopping robot - "kenken"**. In: PROCEEDINGS OF THE 2002 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 3984–3991, Washington DC, USA, 2002.

[25] NIIYAMA, R.; NAGAKUBO, A. ; KUNIYOSHI, Y.. **Mowgli: A bipedal jumping and landing robot with an artificial musculoskeletal system**. In: PROCEEDINGS 2007 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 2546–2551, Roma, Italy, 2007.

[26] GUIZZO, E.; ACKERMAN, E.. **Boston Dynamics Officially Unveils its Wheel-Leg Robot: "Best of Both World"**. In: IEEE SPECTRUM, 2017.

[27] GRAICHEN, K.; HENTZELT, S.; HILDEBRANDT, A.; KÄRCHER, N.; GAISSERT, N. ; KNUBBEN, E.. **Control Design for a Bionic Kangaroo**. Control Engineering Practice, p. 106–117, 2015.

[28] KLEMM, V.; MORRA, A.; SALZMANN, C.; TSCHOPP, F.; BODIE, K.; GULICH, L.; KÜNG, N.; MANNHART, D.; PFISTER, C.; VIERNEISEL, M.;

WEBER, F.; DEUBER, R. ; SIEGWART, R.. **Ascento: A two-wheeled jumping robot**. In: PROCEEDINGS OF THE 2019 INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 7515–7521, Montreal QC, Canada, 2019.

[29] WILLIAMS, R. L.. **Darwin-op humanoid robot kinematics**. In: INTERNATIONAL DESIGN ENGINEERING TECHNICAL CONFERENCES AND COMPUTERS AND INFORMATION IN ENGINEERING CONFERENCE, p. 1187–1196, Chicago,USA, 2012.

[30] JORGENSEN, S. J.; HOLLEY, J.; MATHIS, F.; MEHLING, J. S. ; SENTIS, L.. **Thermal Recovery of Multi-Limbed Robots with Electric Actuators**. IEEE Robotics and Automation Letters, p. 1077–1084, 2019.

[31] VILLARREAL, O.; BARASUOL, V.; CAMURRI, M.; FOCCHI, M.; FRANCESCHI, L.; PONTIL, M.; CALDWELL, D. G. ; SEMINI, C.. **Fast and Continuous Foothold Adaptation for Dynamic Locomotion through Convolutional Neural Networks**. IEEE International Conference on Robotics and Automation, 2018.

[32] KATZ, B.; CARLO, J. D. ; KIM, S.. **Mini cheetah: A platform for pushing the limits of dynamic quadruped control**. Proceedings - IEEE International Conference on Robotics and Automation, p. 6295–6301, 2019.

[33] WELLHAUSEN, L.; DOSOVITSKIY, A.; RANFTL, R.; WALAS, K.; CADENA, C. ; HUTTER, M.. **Where should i walk(predicting terrain properties from images via self-supervised learning**. IEEE Robotics and Automation Letters, p. 1509–1516, 2019.

[34] BUCHANAN, R.; BANDYOPADHYAY, T.; BJELONIC, M.; WELLHAUSEN, L.; HUTTER, M. ; KOTTEGE, N.. **Walking posture adaptation for legged robot navigation in confined spaces**. IEEE Robotics and Automation Letters, p. 2148–2155, 2019.

[35] VUKOBRATOVIĆ, M.; BOROVAC, B.. **Zero-moment point—thirty five years of its life**. International journal of humanoid robotics, p. 157–173, 2004.

[36] HIGA, F. Y.; LAHR, G. J.; CAURIN, G. A. ; BOAVENTURA, T.. **Joint kinematic configuration influence on the passivity of an impedance-controlled robotic leg**. Proceedings - IEEE International Conference on Robotics and Automation, p. 9516–9522, 2019.

[37] HE, G.; GENG, Z.. **Exponentially Stabilizing an One-Legged Hopping Robot with Non-SLIP Model in Flight Phase**. Mechatronics, 19(3):364–374, 2009.

[38] SICILIANO, B.; KHATIB, O.. **Springer handbook of robotics**. Springer, Berlin,Germany, 2nd edition, 2016.

[39] CARON, S.; KHEDDAR, A. ; TEMPIER, O.. **Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control**. Proceedings - IEEE International Conference on Robotics and Automation, 2019-May:277–283, 2019.

[40] BUDHIRAJA, R.; CARPENTIER, J. ; MANSARD, N.. **Dynamics consensus between centroidal and whole-body models for locomotion of legged robots**. In: 2019 INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), p. 6727–6733. IEEE, 2019.

[41] SEMINI, C.; GOLDSMITH, J.; REHMAN, B. U.; FRIGERIO, M.; BARASUOL, V.; FOCCHI, M. ; CALDWELL, D. G.. **Design overview of the hydraulic quadruped robots**. In: THE FOURTEENTH SCANDINAVIAN INTERNATIONAL CONFERENCE ON FLUID POWER, p. 20–22. sn, 2015.

[42] CHEN, G.; JIN, B. ; CHEN, Y.. **Nonsingular fast terminal sliding mode posture control for six-legged walking robots with redundant actuation**. Mechatronics, 50:1–15, 2018.

[43] GEHRING, C.; COROS, S.; HUTTER, M.; BLOESCH, M.; HOEPFTINGER, M. A. ; SIEGWART, R. Y.. **Control of dynamic gaits for a quadrupedal robot**. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2013: 6-10 MAY 2013, KARLSRUHE, GERMANY, p. 3287–3292. IEEE, 2013.

[44] SAKAINO, S.; OHNISHI, K.. **Sliding Mode Control Based on Position Control for Contact Motion Applied to Hopping Robot**. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY, p. 170–175, 2006.

[45] LIOU, H.; HO, M.. **Hopping Control of a Pneumatic Single-Legged Robot using Sliding Mode Control**. In: INTERNATIONAL AUTOMATIC CONTROL CONFERENCE, p. 1–6, 2019.

[46] THOMAS, M.; BANDYOPADHYAY, B. ; VACHHANI, L.. **Posture stabilization of unicycle mobile robot using finite time control techniques**. IFAC-PapersOnLine, 49(1):379–384, 2016.

[47] MURRAY, R. M.; LI, Z. ; SASTRY, S. S.. **A Mathematical Introduction to Robotic Manipulation**. CRC Press, Inc., 1994.

[48] ABBASI, W.; SHAH, I.; UR REHMAN, F. ; UD DIN, S.. **Stabilization of Nonholonomic System in Chained Form via Super Twisting Sliding Mode Control**. In: 13TH INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES, p. 1–5, 2017.

[49] LIU, G.-H.; LIN, H.-Y.; LIN, H.-Y.; CHEN, S.-T. ; LIN, P.-C.. **A Bio-inspired Hopping Kangaroo Robot with an Active Tail**. Journal of Bionic Engineering, 11(4):541–555, 2014.

[50] WEI, Q.; LUO, M.; ZHAO, J. ; GUO, F.. **Trajectory Planning and Control for Hopping Robot at the Stance Phase**. In: PROCEEDINGS OF THE 2017 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION, p. 1608–1613, Takamatsu, Japan, 2017.

[51] NAIK, K. G.; MEHRANDEZH, M. ; BARDEN, J. M.. **Control of a One-legged Hopping Robot using a Hybrid Neuro-PD Controller**. In: PROCEEDINGS OF THE 2006 CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING, p. 1530–1533, Ottawa ON, Canada, 2006.

[52] FRIDMAN, L.; MORENO, J. ; IRIARTE, R.. **Sliding Modes after the First Decade of the 21st Century**, volumen 412. Springer-Verlag Berlin Heidelberg, 2011.

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

[54] SICILIANO, B.; SCIAVICCO, L.; VILLANI, L. ; ORIOLO, G.. **Robotics: Modelling, Planning and Control**. Springer-Verlag London Limited, 2nd edition, 2009.

[55] UTKIN, V.. **Discussion Aspects of High-order Sliding Mode Control**. IEEE Transactions on Automatic Control, 61(3):829–833, 2015.

[56] KHALIL, H. K.; GRIZZLE, J. W.. **Nonlinear systems**, volumen 3. Prentice hall Upper Saddle River, NJ, 2002.

[57] UTKIN, V. I.. **Sliding modes and their applications in variable structure systems**. Mir, Moscow, 1978.

[58] OLIVEIRA, T. R.; PEIXOTO, A. J. ; HSU, L.. **Sliding mode control of uncertain multivariable nonlinear systems with unknown control direction via switching and monitoring function**. IEEE Transactions on Automatic Control, 55(4):1028–1034, 2010.

[59] OLIVEIRA, T. R.; PEIXOTO, A. J. ; HSU, L.. **Sliding mode control of uncertain multivariable nonlinear systems with unknown control direction via switching and monitoring function**. IEEE Transactions on Automatic Control, 55(4):1028–1034, 2010.

[60] BOIKO, I.; FRIDMAN, L.. **Analysis of chattering in continuous sliding-mode controllers**. IEEE transactions on automatic control, 50(9):1442–1446, 2005.

[61] BENALLEGUE, A.; MOKHTARI, A. ; FRIDMAN, L.. **High-order sliding-mode observer for a quadrotor uav**. International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, 18(4-5):427–440, 2008.

[62] SLOTINE, J.-J. E.; LI, W. A.. **Applied Nonlinear Control**. Prentice Hall, Inc., 1991.

[63] **HumaRobotics**. `https://github.com/HumaRobotics`. Accessed: 2020-07-20.

5
Appendix

5.1
Relevance of the Research Theme

In this section, we illustrate the relevance of the research on legged mobile robots according to the following aspects: authors,affiliation,country,document type, source and area of knowledge. The following keywords and their combinations were used to obtain the statistical results: legged mobile robots; nonlinear robust control; sliding mode control; outdoor environments; agricultural fields.

5.1.1
Legged mobile robots

Figure 5.1 shows the number of publications per authors using "legged mobile robots" as a keyword search, where it can be seen that Hutter, M. (57), Siegwart, R. (22) and Gehring, C. (18) are the most prominent authors. Notice that, in Figure 5.2, the most relevant institutions working with legged mobile robots are Carnegie Mellon University, Harbin Institute of Technology and ETH Zurich, counting more than 120 publications each.

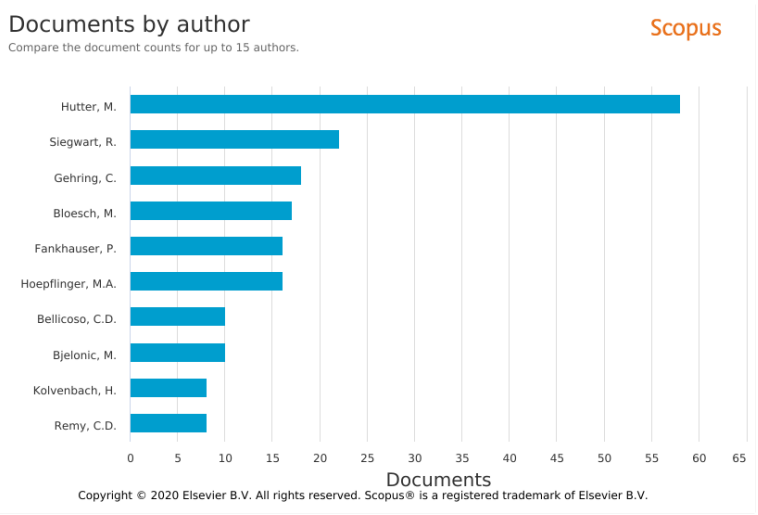


Figure 5.1: Number of publication per authors: legged mobile robots

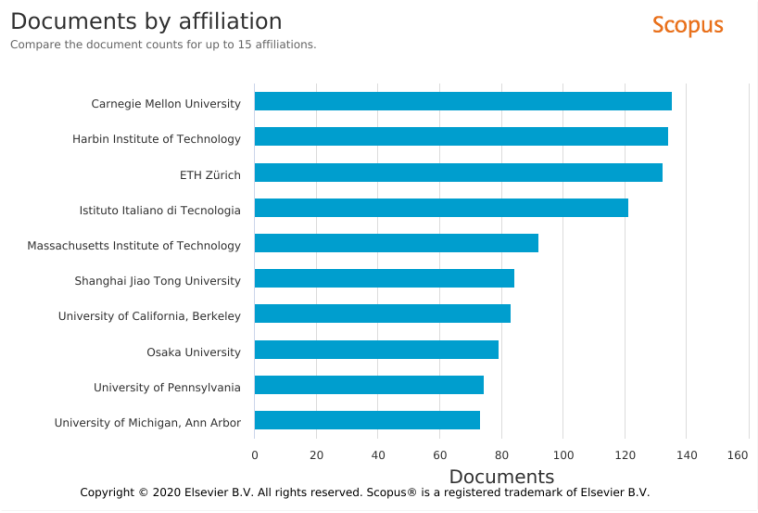


Figure 5.2: Number of publications per Affiliation:legged mobile robots

The figure 5.3 shows that in the last years has been an increase in the publications about legged mobile robots since 2017 in 2019 the number of documentations is the biggest in the nine years.

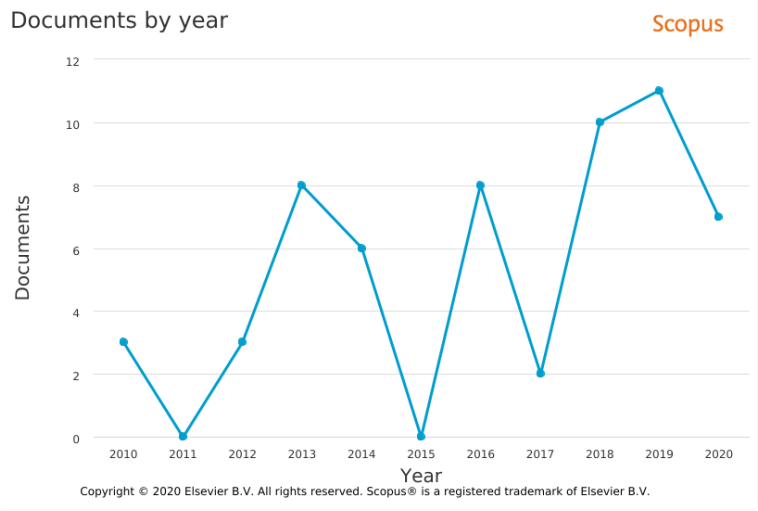


Figure 5.3: Number of publications per year:legged mobile robots

The USA is the leader in the publication about legged mobile robots such as in the graphic in the figure (5.4), Japan is the second country that has documents published about legged mobile robots. In the USA case, the American government use the legged mobile robot in the military application and financed many studies in this field. Japan is in second place, in publications about legged mobile robots, the country has a breakthrough in research with humanoid robots the examples are the Asimo, the Qrio and the Actroid.

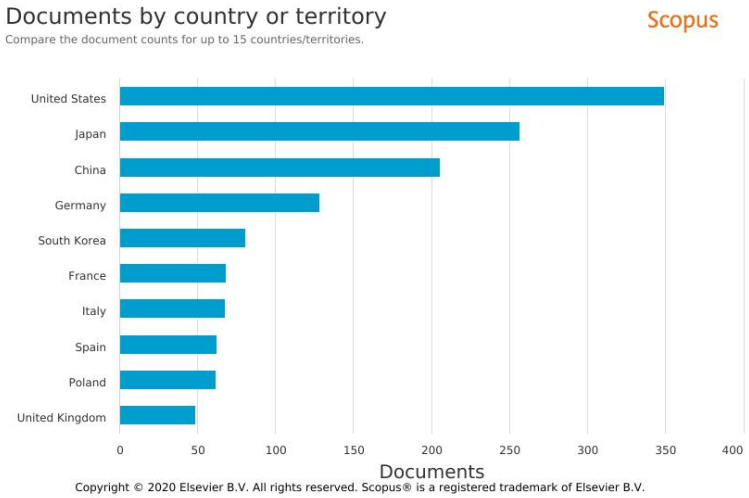


Figure 5.4: Number of publications per country:legged mobile robots

According to 5.5, most documents about legged mobile robots are scientific papers, in 2019 the ICRA had a section about legged mobile robots, in 5.6 show the type of paper in the IEEE,

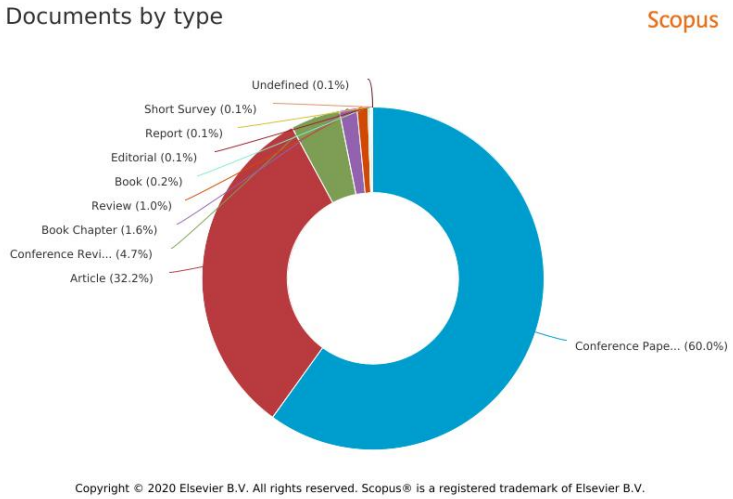


Figure 5.5: Number of publications per document type:legged mobile robots

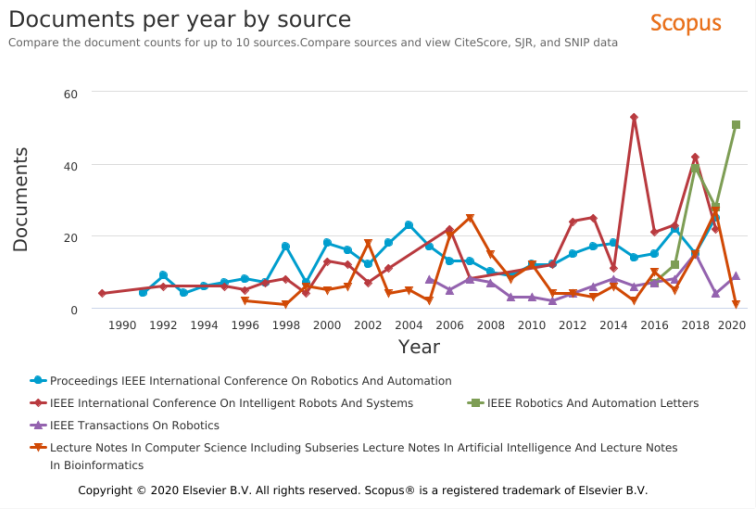


Figure 5.6: Number of publications per source:legged mobile robots

According to figure 5.7 area of knowledge that publishes about legged mobile robots are engineering and computer science a science computation both areas has been combined to project legged robots mobiles using artificial intelligence such as, reinforcement learning, neural networks, control such as robust control, adaptative control, PID for instance.

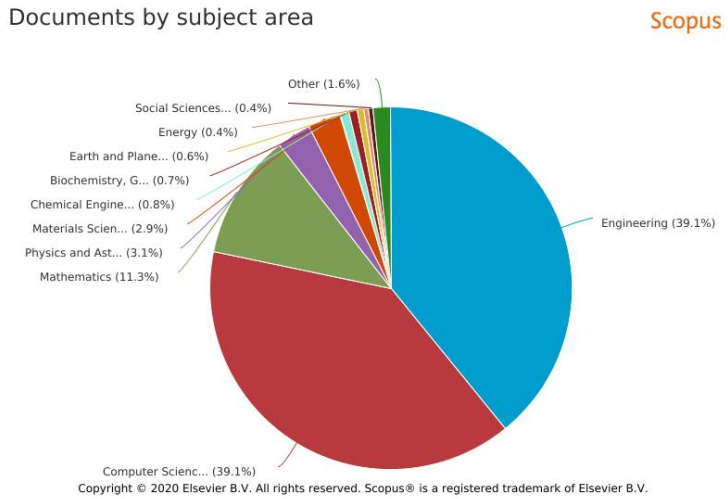


Figure 5.7: Number of publications per subject:legged mobile robots

5.1.2
Legged mobile robots in agriculture

Agriculture is an important activity for the humanity, since the antiquity the human beings need feed and obtain their food to survive, the discovery of agriculture was an contribution for this objective, it contribute that the capacity to human beings fixed them in a land.

The agriculture has a much importance in the economic activity, in Brazil for instance since its discovery in 1500's to the present this activity moving the economy starting to the "pau-brasil" and sugar cane in the Brazil Colony, the coffee in the Brazil Empire to the Brazil Republic, start the mechanization of agriculture in the start of 1960's.

The Industrial Revolution caused a demand for an increase in the agricultural productivity,chasing the new methods for modernize the crop production. The control techniques are in the context, allow the design mechanism, which has accuracy in execution tasks just as, sowing, planting, pest monitoring, harvest.

The combination of the keywords legged mobile robots and agriculture results in graphics in this section, this research shows the applications about legged mobile robots in agriculture. The graphics bellow associate the documents by year, affiliation, author, country, document type and source.

The legged mobile robots has been gaining prominence in robotic, but second the graphic 5.8, note the publication about legged mobile robot in agriculture yet has been in beginning, the graphics shows that the publication has been an increase from 2015 but, the number of document about the themes is three that means the theme has been a low increase. All the Institutes in the graphic has the same number of publications about the legged robots applications in agriculture.

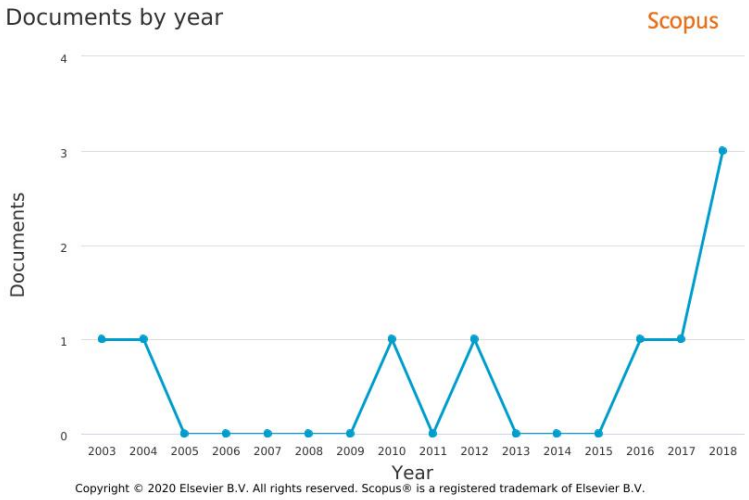


Figure 5.8: Publications about legged mobile application in agriculture

Note in figure 5.9 that all the author published the same quantity of papers in this theme and, the behaviour repeat about affiliations second the figure 5.10. A reason for this index can be mean that the application to legged mobile robots in agriculture is a theme that does not totally exploited.

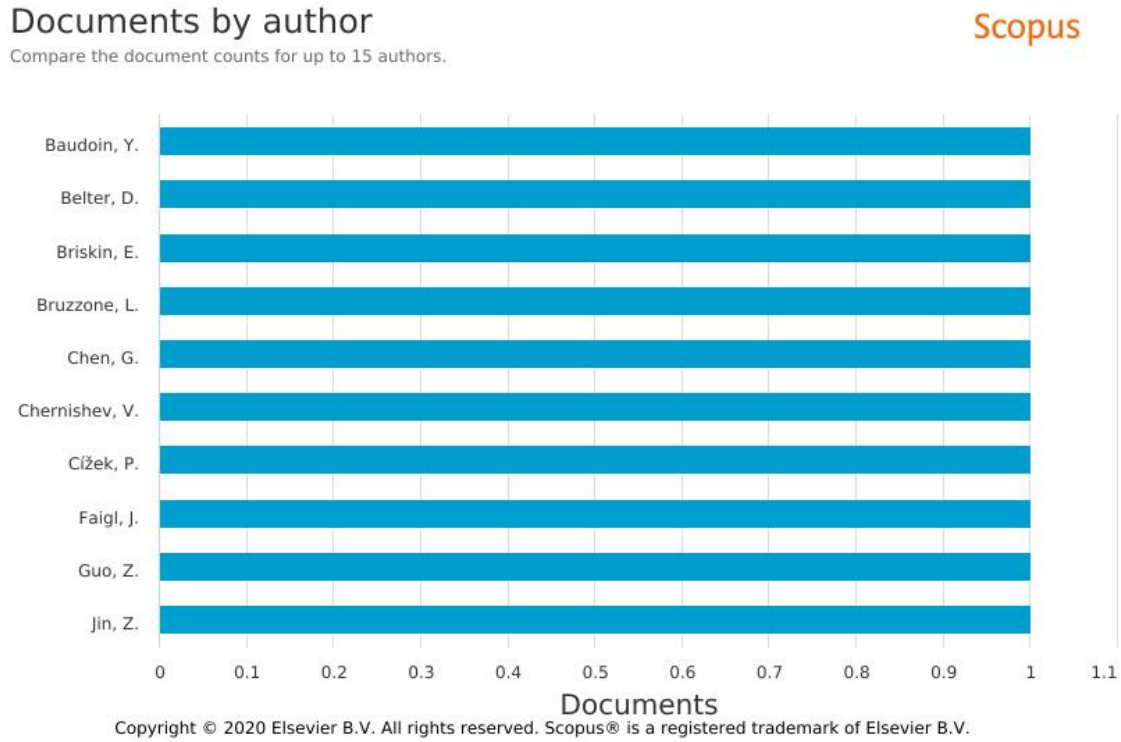


Figure 5.9: Number of publications per Author:legged mobile robots in agriculture

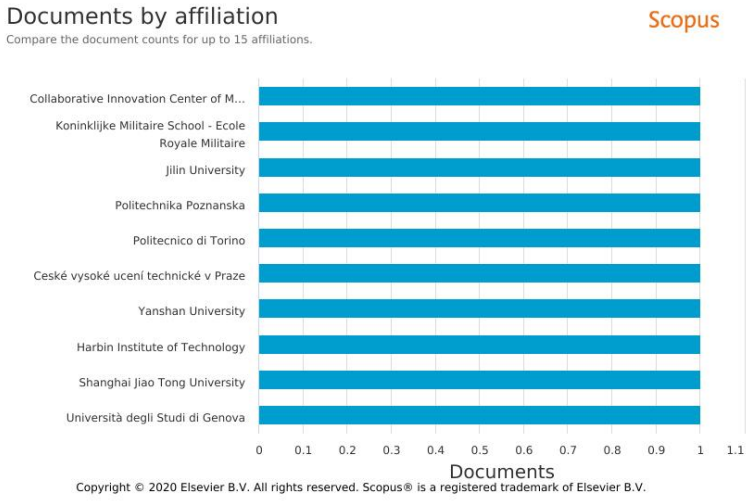


Figure 5.10: Number of publications per Author:legged mobile robots in agriculture

Analysing the graphic in figure 5.12, China has four documents about legged mobile robots and, all the other country has one publication.

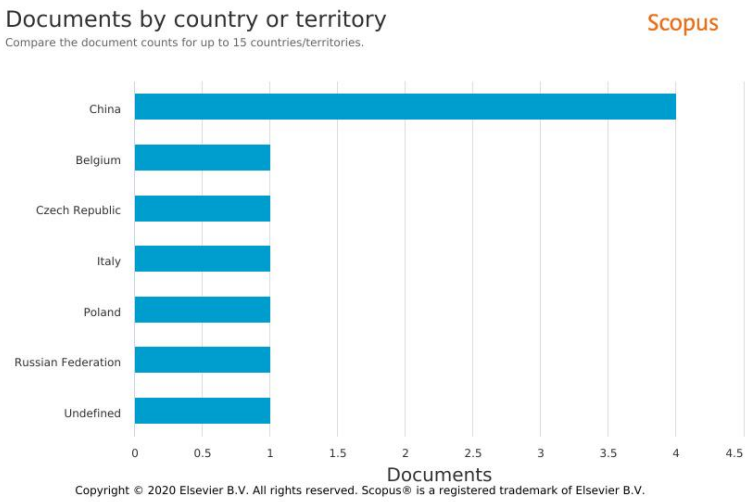


Figure 5.11: Number of publications per Country:legged mobile robots in agriculture

The most documents about the legged mobile robots in agricultural applications has been published by four knowledge areas engineering, agricultural and biologic science, computer science and chemistry.

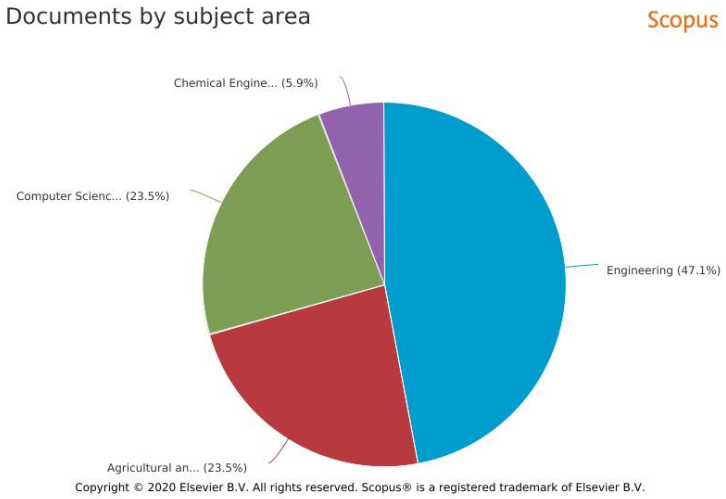


Figure 5.12: Number of publications per Subject area:legged mobile robots in agriculture

The most document about legged mobile in agriculture are articles (55.6%), conference papers corresponding to 22.2% and, the knowledge areas that leaders in publications in this field are the engineering, science computer and agricultural and biological sciences. The IFAC has published 3 documents in 2013 and Nongye Gongcheng Xuebao Transactions Of Chinese Society Of Agricultural Engineering has two papers published.

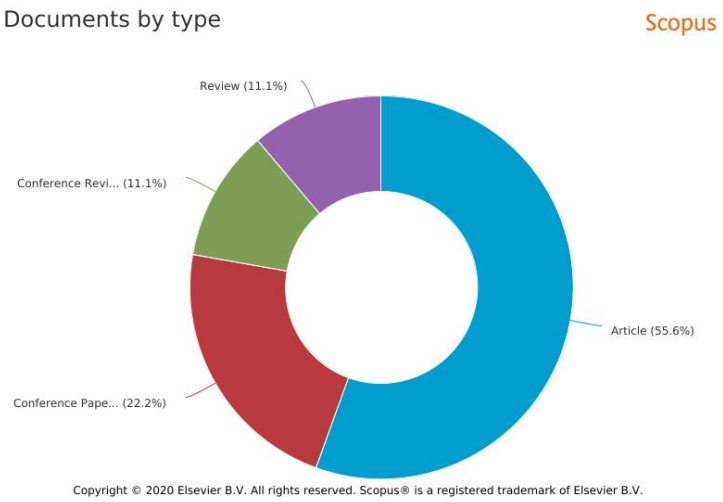


Figure 5.13: Number of publications per Doctype:legged mobile robots in agriculture

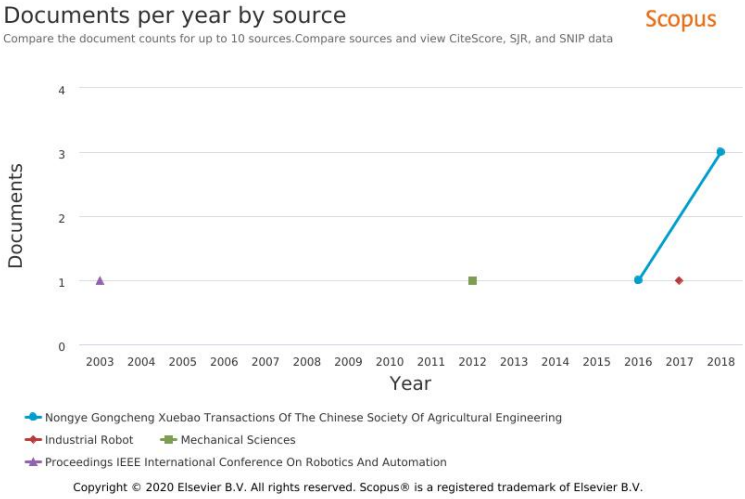


Figure 5.14: Number of publications per Source:legged mobile robots in agriculture

5.1.3 Sliding mode control

In the last years has been an increase attention in systems witch the action control and disturbances have discontinuity. The problem of discontinuous control is resumed in to select a sliding surface for the function control to have a discontinuity [56, 57], this is objective of sliding mode control.

There are many works in sliding mode control, in [58] the authors using sliding mode control in the monitoring function in visual serving to deal with the uncertain in the camera, in [59] the authors propose control an uncertain nonlinear system and unknown direction using sliding mode control. In [60] the author analysis the chattering in continuous sliding mode controllers. In [61] the authors propose to

use a feedback linearization and a high order sliding mode control observation are used in a quadrirotor UAV control.

The sliding mode control allow construct a control law that ensures the robustness of system using the Slotine example [62], to prove the robustness and stability of system with bounded uncertainties.

Considering the following system:

$$\ddot{x} = f(x) + b(x)u\,. \tag{5-1}$$

$$\tag{5-2}$$

The bounded uncertainties on $f(x)$ and $b(x) \neq 0$ satisfied the following conditions adopted in [62]:

$$0 < b_{min} \leq b \leq b_{max}\,, \tag{5-3}$$

$$0 < |f - \hat{f}| \leq F \leq b_{max}\,, \tag{5-4}$$

$\hat{f}$ is an estimate of function f. Considering the estimation of b, $\hat{b}$ is the geometric mean given by [62]:

$$\hat{b} = (b_{min}\, b_{max})^{1/2}\,. \tag{5-5}$$

According [62] the bound in Eq.5-3 can be written the following form:

$$\beta^{-1} \leq \frac{\hat{b}}{b} \leq \beta\,, \tag{5-6}$$

or

$$\beta^{-1} \leq \frac{b}{\hat{b}} \leq \beta\,, \tag{5-7}$$

Given the following sliding surface:

$$s = \dot{\tilde{x}} + \lambda\,\tilde{x}\,. \tag{5-8}$$

Derivating the sliding surface in Eq.5-8 results in:

$$\dot{s} = \ddot{x} - \ddot{x}_d + \lambda\,\dot{\tilde{x}}\,, \tag{5-9}$$

$$= f - b\,u - \ddot{x}_d + \lambda\,\dot{\tilde{x}}\,, \tag{5-10}$$

Defining the control u as:

$$u = \hat{b}^{-1}\hat{u} - ksign(s)\,, \tag{5-11}$$

and choosing the following k which satisfies the sliding condition.

$$k \geq \beta(F+\eta) + (\beta-1)|\hat{u}|\,, \tag{5-12}$$

Replacing Eq.5-11 into Eq.5-9:

$$\dot{s} = f - b\,(\hat{b}^{-1}\hat{u} - k\,sign(s)) - \ddot{x}_d + \lambda\,\dot{\tilde{x}}\,, \tag{5-13}$$

where,

$$\hat{u} = \hat{f} - \ddot{x}_d - \lambda\tilde{x} \tag{5-14}$$

which results in the equation bellow:

$$\dot{s} = f + b\,(\hat{b}^{-1}(\hat{f} - \ddot{x}_d - \lambda\tilde{x}) - k\,sign(s)) - \ddot{x}_d + \lambda\,\dot{\tilde{x}}\,, \tag{5-15}$$

$$= (f - b\,\hat{b}^{-1}\hat{f}) + (1 - b\hat{b})(-\ddot{x}_d + \lambda\tilde{x}) - b\,\hat{b}^{-1}\,k\,sign(s)\,. \tag{5-16}$$

Note that the k is chosen to ensure the sliding condition, choosing the a Lyapunov function $V(s)$ can to prove that the system is reach the sliding surface and the state error vector converges to zero. The Lyapunov function is given by:

$$2\,V(s) = s^2\,, \tag{5-17}$$

$$\dot{V}(s) = s\,\dot{s} \tag{5-18}$$

$$= s((f - b\,\hat{b}^{-1}\hat{f}) + (1 - b\hat{b}^{-1})(-\ddot{x}_d + \lambda\tilde{x})) - b\,\hat{b}^{-1}\,k\,s\,sign(s)\,. \tag{5-19}$$

Note that $V(s)$ is negative definite if the gain k is given by:

$$k \geq |\hat{b}b^{-1}f - \hat{f} + (\hat{b}b^{-1} - 1)(\lambda\,\dot{\tilde{x}} - \ddot{x}_d)| + \eta\,\hat{b}b^{-1}\,, \tag{5-20}$$

$$\geq (\hat{b}b^{-1} - 1)|\hat{f} - \ddot{x}_d + \lambda\dot{\tilde{x}}| + \hat{b}b^{-1}|f - \hat{f}| + \eta\,\hat{b}b^{-1}\,, \tag{5-21}$$

$$\geq (\hat{b}b^{-1} - 1)|\hat{u}| + \hat{b}b(F+\eta)\,, \tag{5-22}$$

$$k \geq (\beta-1)|\hat{u}| + \beta(F+\eta)\,. \tag{5-23}$$

When the sliding surface converges s to zero we have:

$$\dot{\tilde{x}} = -\lambda\,\tilde{x}\,, \tag{5-24}$$

and the error $\tilde{x}$ converges to zero.

The advantages of sliding mode control are to deal with uncertain in the model and parameters and structure variable systems, that is the reason why has been increasing the research in these control techniques. In the Fig. 5.15 shows the phase portrait in the sliding mode control in ideal situation, when the surface is reached the variable $\dot{\tilde{x}}$ converges to zero the slope is given by angular coefficient λ.

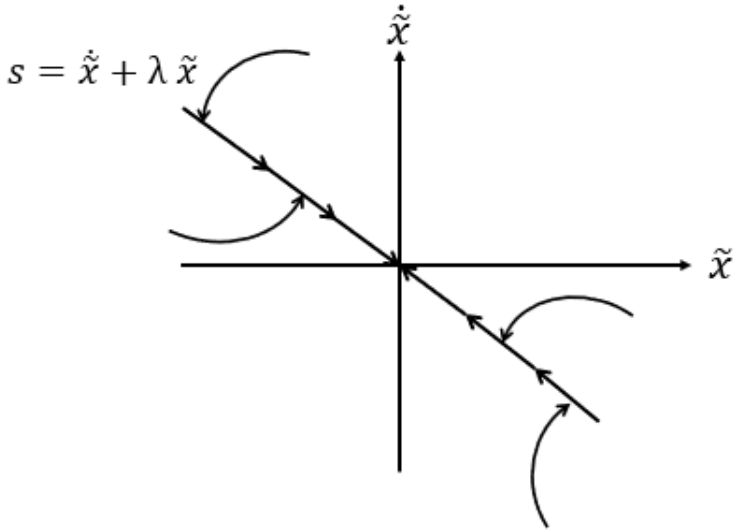


Figure 5.15: Phase portraint in sliding mode control

In the practice the imperfection in the devices and delays cause in the sliding mode control the chattering phenomenal. During the delay between the function $sign(s)$ changes and the time control switches the trajectory cross the sliding surface into the opposite region, when the control switches the trajectory reverses its direction creating an oscillation. The chattering is a problem in sliding mode control because results in a low control accuracy,high heat losses in the electrical power systems, and a wear in mechanical systems. It excites the unmodeled high-frequency dynamics, impairing the performance of the systems and causing instability. [56]

There are method to eliminate or reducing the chattering in sliding mode control, in [56] is proposed two methods, the first is split the control in a continuous part and a switching this method reduce the amplitude of switching, the second is replace the function $sign(s)$ by the saturation function $u = -\beta(x)\, sat(s/\epsilon)$ other example to eliminate the chattering is using the function $sign(s)|s|^p$ where $0 \leq p \leq 1/2$.

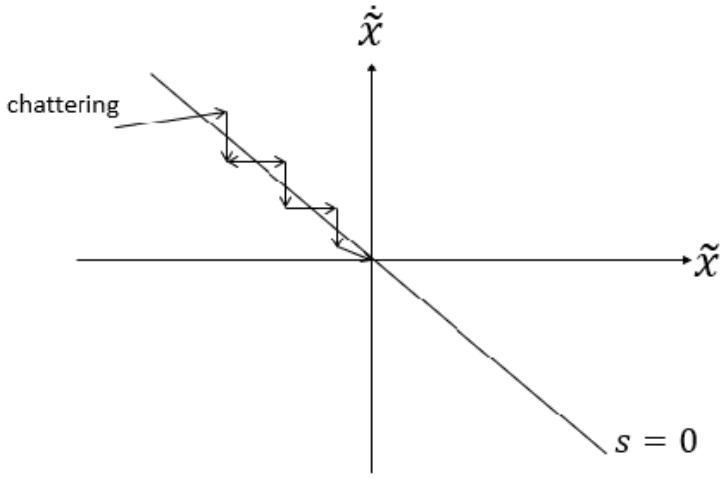


Figure 5.16: Chattering phenomenal in sliding mode control

5.2 Forward kinematic of leg

The hexapod leg is robotics manipulators where the base is body robot then, it is possible to find the leg position and velocity using the forward and inverse kinematic,in the figure (5.17) shows the frames in the hexapod leg. The importance of frame is, find the homogeneous transformation and then, describe the leg position and velocity.

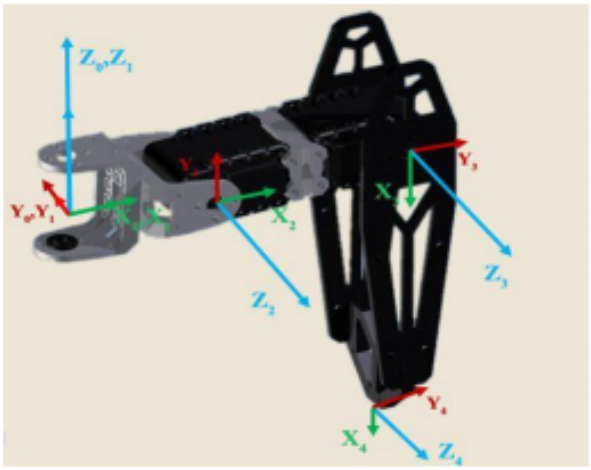


Figure 5.17: Hexapod Leg and the Axis According Denavit-Hartenberg Criteria

The transformation of hexapod robot using the Denavit-Hartenberg for an

hexapod leg is given by:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5-25}$$

Where α_i,θ_i,a_i and d_i are the Denavit-Hartenberg parameters i is the link index.

The Denavit-Hatenberg parameters for a hexapod robot are in the table 5.2:

link	a_1	d_i	α_i	θ_i
1	L_1	0	$\pi/2$	θ_1
2	L_2	0	0	θ_2
3	L_3	0	0	θ_3

Table 5.1: Denavit-Hatenberg parameters

Using the parameters we have following Homogenious Matrix Transformation:

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & L_1\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & L_1\sin\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5-26}$$

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5-27}$$

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5-28}$$

The transformation the base to foot is given by:

$$T_3^0 = T_1^0 T_2^1 T_3^2\,, \tag{5-29}$$

The homogeneous transformation matrix T_3^0 is given by:

$$T_3^0 = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & c_1(L_1 + L_2c_2 + L_3c_{23}) \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1(L_1 + L_2c_2 + L_3c_{23}) \\ s_{23} & c_{23} & 0 & L_2s_2 + L_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5-30}$$

For simplicity, we adopted the following notation $\sin\theta_i = s_i$, $\sin(\theta_i + \theta_j) = s_{ij}$, $\cos\theta_i = c_i$, $\cos(\theta_i + \theta_j) = c_{ij}$ In the last column, the third term is the foot position then the position x ,y and z in relation to base is given by:

$$\begin{aligned} x &= c_1(L_1 + L_2c_2 + L_3c_{23}), \\ y &= s_1(L_1 + L_2c_2 + L_3c_{23}), \\ z &= L_2s_2 + L_3s_{23}. \end{aligned} \tag{5-31}$$

The linear velocity and angular of leg is given by the following equation:

$$\dot{p}_{foot} = J_L(q)\dot{q}, \tag{5-32}$$

$$\omega_{foot} = J_A(q)\dot{q}. \tag{5-33}$$

Where $J_L \in \mathbb{R}^{3\times3}$ and $J_A \in \mathbb{R}^{3\times3}$ are respectively the linear and angular Jacobian of leg .

In general, the Jacobian is given by the following equation:

$$J = \begin{bmatrix} \vec{z}_0 \times (\vec{p}_3 - \vec{p}_0) & \vec{z}_1 \times (\vec{p}_3 - \vec{p}_1) & \vec{z}_2 \times (\vec{p}_3 - \vec{p}_2) \\ \vec{z}_0 & \vec{z}_1 & \vec{z}_2 \end{bmatrix} \tag{5-34}$$

$\vec{p}_0$,$\vec{p}_1$, $\vec{p}_2$ and, $\vec{p}_3$ are the vectors that represent the point position and $\vec{z}_0$, $\vec{z}_1$ and, $\vec{z}_2$ are the unit vector in the angular velocity direction. The vectors position $\vec{p}_0$, $\vec{p}_1$, $\vec{p}_2$ and, $\vec{p}_3$ are:

$$\vec{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \vec{p}_1 = \begin{bmatrix} L_1c_1 \\ L_1s_1 \\ Z_1 \end{bmatrix} \tag{5-35}$$

$$\vec{p}_2 = \begin{bmatrix} L_1c_1 + L_2c_{12} \\ L_1c_1 + L_2s_{12} \\ Z_1 + L_2s_2 \end{bmatrix} \quad \vec{p}_3 = \begin{bmatrix} c_1(L_1 + L_2c_2 + L_3c_{23}) \\ s_1(L_1 + L_2c_2 + L_3c_{23}) \\ Z_1 + L_2s_2 + L_3s_{23} \end{bmatrix} \tag{5-36}$$

The vectors $\vec{z}_0$, $\vec{z}_1$ and, $\vec{z}_2$ are:

$$\vec{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \vec{z}_1 = \begin{bmatrix} s_1 \\ -s_1 \\ 0 \end{bmatrix} \vec{z}_2 = \begin{bmatrix} s_1 \\ -s_1 \\ 0 \end{bmatrix} \tag{5-37}$$

The Jacobian matrix for a hexapod leg is given by:

$$J = \begin{bmatrix} s_1(L_1 + L_2c_2 + L_3c_{23}) & -c_1(L_2s_2 + L_3s_{23}) & -L_3c_1s_{23} \\ c_1(L_1 + L_2c_2 + L_3c_{23}) & -s_1(L_2s_2 + L_3s_{23} & -L_3s_1s_{23} \\ 0 & L_2c_2 + L_3c_{23} & -L_3c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{5-38}$$

The Jacobian describes the linear and angular velocities and the joints variables in this case, note that the velocity component ω_x and ω_y are not zero and the relation between them is given by:

$$\frac{\omega_x}{\omega_y} = -\frac{\sin\theta_1}{\cos\theta_1} \tag{5-39}$$

5.3 Inverse kinematic of hexapod leg

The inverse kinematic is the part responsible for converters the trajectory of legs in the Cartesian in the desired joints trajectories. In the figure (5.18) is shown the lateral and superior visions using the cosine law, it is possible to calculate the inverse kinematic.

The according to figure(5.18) the angle θ_1 is given by the following equation:

$$\theta_1 = tan^{-1}\left(\frac{Y}{X}\right) . \tag{5-40}$$

Using the cosine law it is possible calculate the β and γ.

$$L_3^2 = L_2^2 + D^2 - 2L_2Dcos(\beta)\,, \tag{5-41}$$

$$L_3^2 - L_2^2 - D^2 = -2L_2Dcos(\beta)\,, \tag{5-42}$$

$$cos(\beta) = \frac{L_2^2 + D^2 - L_3^2}{2L_2D}\,, \tag{5-43}$$

$$\beta = cos^{-1}\left(\frac{L_2^2 + D^2 - L_3^2}{2L_2D}\right) . \tag{5-44}$$

$$\tag{5-45}$$

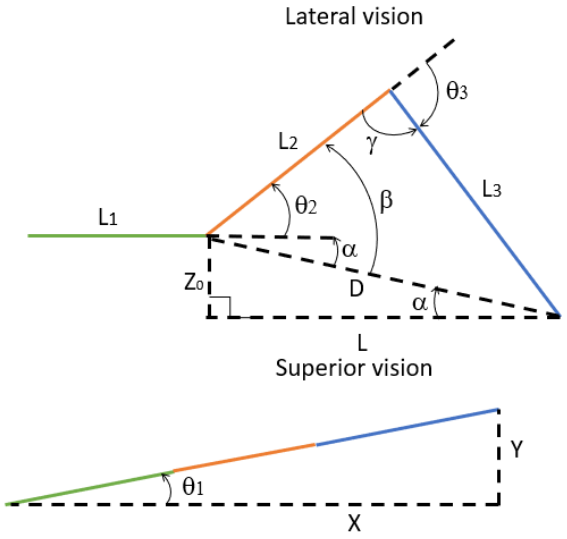


Figure 5.18: Inverse kinematic of hexapod leg

$$
\begin{aligned}
D^2 &= L_2^2 + L_3^2 - 2L_2L_3cos(\gamma)\,, \\
L_3^2 + L_2^2 - D^2 &= 2L_2L_3cos(\gamma)\,, \\
cos(\gamma) &= \frac{L_2^2 + L_3^2 - D^2}{2L_2L_3}\,, \\
\gamma &= cos^{-1}\left(\frac{L_2^2 + L_3^2 - D^2}{2L_2L_3}\right)\,.
\end{aligned}
$$

The angle α is given by the following equation:

$$\alpha = sin^{-1}\left(\frac{Z_0}{\sqrt{Z_0^2 + L^2}}\right)\,. \tag{5-46}$$

Obtained the angles α,β, and γ it is possible calculate the angles θ_2 and θ_3:

$$\theta_2 = \beta - \alpha\,, \tag{5-47}$$

$$\theta_2 = cos^{-1}\left(\frac{L_2^2 + D^2 - L_3^2}{2L_2L_3}\right) - sin^{-1}\left(\frac{Z}{\sqrt{Z^2 + L^2}}\right)\,. \tag{5-48}$$

$$
\begin{aligned}
\theta_3 &= \gamma - \pi\,, \\
\theta_3 &= cos^{-1}\left(\frac{L_2^2 + D^2 - L_3^2}{2L_2L_3}\right) - \pi\,.
\end{aligned} \tag{5-49}
$$

5.4 Dynamic model of a leg

The dynamic model of manipulator is an important to simulate the leg motion , it allow analysis the effects in the leg, the forces and torques actuating in the hexapod leg. The dymanic equation can be obtained using two methods, the first method is the Newton-Euler formulation which consists in describe the diagram forces and torques in the body and apply the Newton's law. The second method is the Lagrane's formulation which consists in the energy concept, calculating the kinetic and potential energy and apply the Lagrangian mechanic equations. In two cases, the dynamic equation that describes the motion of i-th leg can be obtained by the Euler-Lagrange formalism as:

$$H_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q} + G_i(q) = \tau + J_e^T(q)\lambda\underline{f_{wrech}}\,, \tag{5-50}$$

where $H(q) \in \mathbb{R}^{3\times3}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{3\times3}$ is the Coriolis and centrifugal matrix terms, $G(q) \in \mathbf{R}^{3\times1}$ is the gravitational terms, $\tau \in \mathbb{R}^{3\times1}$ is the joint torques, $J_e \in \mathbb{R}^{6\times3}$ is the leg Jacobian and $\lambda \in \mathbf{R}^{6\times1}$ is the generalized contact forces and moments.

The elements of the dynamic model for the i-th leg are given by:

$$
\begin{aligned}
h_{11} &= m_1\, l_{c1}^2 + I_{zz1} + m_2\,(l_1^2 + 2\,l_1\,l_{c2}\,c_2 + l_{c2}^2) + I_{zz2}\,, \\
&\quad + m_3(l_1^2 + 2\,l_1\,l_2\,c_2 + 2\,l_1\,l_2\,c_2 + 2\,l_1\,l_{c3}\,c_3 + l_2^2\,c_2^2 + l_{c3}\,c_{23}^2) + I_{zz3}\,. \\
h_{12} &= h_{13} = h_{21} = 0\,, \\
h_{22} &= m_2\,l_{c2}^2 + I_{xx2}\,s_1^2 + I_{yy2}\,c_1^2 + I_{yy3}\,c_1^2 + m_3\,(l_1^2 + 2\,l_1\,l_2\,c_2 + 2\,l_1\,l_{c3}\,c_3)\,, \\
&\quad + m_3\,(l_2^2\,c_2^2 + l_{c3}\,c_{23}^2) + I_{xx3}\,s_1^2 + I_{yy3}\,c_1^2\,, \\
h_{23} &= m_3\,(l_2\,l_{c3}\,c_3 + l_{c3}^2) + I_{xx3}\,s_1^2 + I_{yy3}\,c_1^2\,, \\
h_{31} &= 0\,, \\
h_{32} &= h_{23}\,, \\
h_{33} &= m_3\,l_{c3}^2 + I_{yy3}\,s_1^2 + I_{xx3}\,c_1^2\,, \\
c_{111} &= 0\,, \\
c_{112} &= -2\,m_2\,l_1\,l_{c2}\,s_2 - m_3\,l_{c3}\,c_{23}\,s_{23}\,, \\
c_{113} &= -2\,m_3\,l_{c3}\,c_{23}\,s23\,, \\
c_{121} &= 0\,, \\
c_{122} &= (I_{xx2} - I_{xx3})\,s_1\,c_1 - (I_{yy2} - I_{yy3})\,s_1\,c_1\,, \\
c_{123} &= -(I_{xx3} - I_{yy3})\,s_1\,c_1\,, \\
c_{131} &= 0\,, \\
c_{132} &= 2\,s_1\,c_1\,(I_{xx3} - I_{yy3})\,, \\
c_{131} &= 0\,,
\end{aligned}
\tag{5-51}
$$

$$
\begin{aligned}
c_{211} &= m_2\, l_1\, l_{c2}\, s_2 + m_3\left(l_1\, l_{c3}\, c_3 + l_2\, s_2\, c_2 + l_{c3}^2 s_{23}\, c_{23}\right), \\
c_{212} &= 0\,, \\
c_{213} &= 0\,, \\
c_{221} &= 2\left(I_{xx2} + I_{xx3}\right) s_1\, c_1 + 2\left(I_{yy2} - I_{yy3}\right) s_1\, c_1\,, \\
c_{222} &= -2\, m_3\left(l_1\, l_2\, s_2 + l_2^2\, s_2\, c_2 + l_{c3}^2\, s_{23}\, c_{23}\right), \\
c_{223} &= -2\, m_3\left(l_1\, l_{c3}\, s_3 + l_{c3}\, s_{23}\, c_{23}\right), \\
c_{231} &= 2\left(I_{xx}\, c_1 - I_{yy}\, s_1\right), \\
c_{232} &= 0\,, \\
c_{233} &= -2\, m_3\, l_2\, l_{c3}\, s_3\,, \\
c_{331} &= 2\, s_1\, c_1\left(I_{xx3} - I_{yy3}\right), \\
c_{332} &= \frac{1}{2}(m_3\, l_2\, l_{c3}\, s_3)\,, \\
c_{333} &= 0\,, \\
g_1 &= 0\,, \\
g_2 &= m_2\, g\, l_{c2}\, c_2 + m_3 z, g\, l_{c3}\, c_{23}\,, \\
g_3 &= m_3\, g\, l_{c3}\, c_{23}\,.
\end{aligned}
\tag{5-52}
$$

Notice that, the use of subscript i in each element of the dynamic model was omitted here for simplicity of notation.

5.5 ROS implementation

ROS is a framework that allows the development of robots and, has several tools and libraries allowing, a hardware abstraction, device drivers, simplifying the robot design. This framework has the following features :

- Communication pier-to-pier.

- Multi language : ROS allow that the programmer write codes in python,$C++$, Java for instance.

- Basic tools : ROS has a many tools which allow monitoring the sensor and visualize the robot structure. An example is the rqt graph which to visualize the graph and the rqt plot which allow to visualize the signals in the sensors and actuators for instance.

– Free software

There are basic concepts of ROS the concepts are:

– Node: Node is a process, it publish(send) a message to a topics or subscribe(receive) a message.The nodes is responsible for control the position and orientation robot for instance, actuating in the joints, an example is the $joint\_state\_publisher$ and $robot\_state\_publisher$

– Topics: Topics contains the message published by a specific node. Topics are buses where nodes exchange messages.

– Message: Message is a type of data comprising typed field an examples of message in ROS.

– Service: Service is defined by a pair of messages, one for the request an other for reply.

The ROS Master handles the information, every time a node needs information from a topic the Master will locate the topic. The communication scheme for ROS is shown in Fig 5.19 the Talker node is responsible for making the information available (publishing in a topic) and the Listener request the information (Subscribing in a topic).

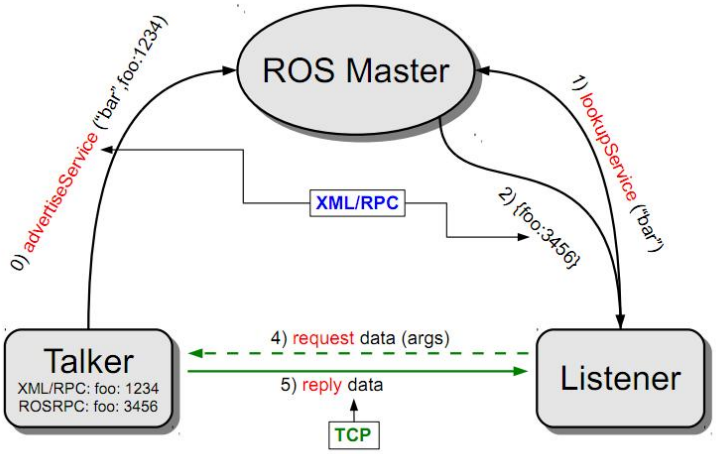


Figure 5.19: Communication scheme for ROS [4]

A software in ROS is organized in packages, a package is the smallest unit of a software organization, for instance, if we think in a particular robot its software in ROS, this robot has a package which can be downloaded in the internet repository.

The Gazebo is a 3D simulator it provides an environment which allows creating a scenario and, simulate the environmental conditions such as illumination, gravity, inertia, etc. We use the Gazebo to simulate the controller developed for the hexapod robot, combining with the ROS using the dynamic model URDF in the phantomx_gazebo package downloaded in the HumaRobotics [63] website and based in the hexapod Phatomx we create a python code which corresponds the note responsible for the hexapod gait. The structure of nodes and tópics are shown in the Fig .5.20. the $gazebo$ and $robot\_state\_publisher$ are the only nodes that have not been implemented by the author.

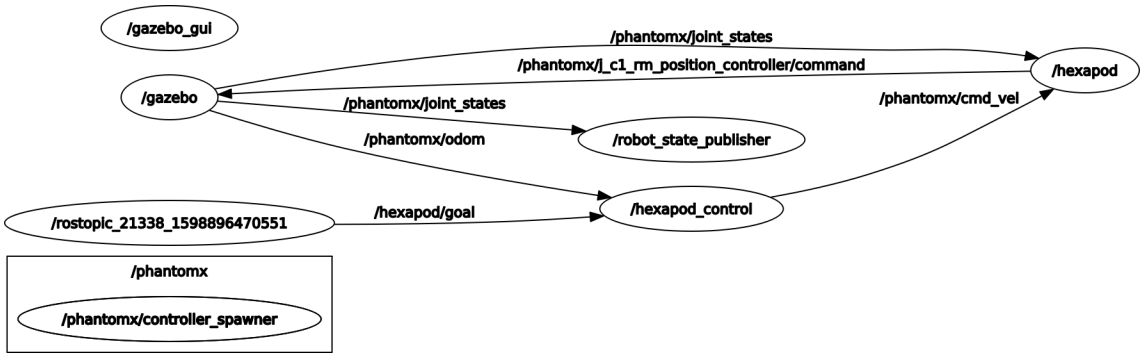


Figure 5.20: Nodes and topics complete

We describe all the nodes and topics used in this work the nodes used are:

- $hexapod$: This the node responsible for define the gait and make the hexapod move.
- $hexapod\_control$: This node responsible for control the position and orientation hexapod.
- $robot\_sate\_publisher$: Publish the state robot in tf topic
- $rostopic\_21338\_1598896470551$: Responsible for publish the desired position orientation the the topic $/hexapod/goal$.

and the topics:

- $/phantomx/odom$: This topic has a message about the position and orientation to the world frame.
- $/phantomx/joint\_states$: Contains the messages about the joint position, velocities and efforts.
- $/hexapod/goal$: Contains a typed message Pose2D and the information about the position and orientation desired.
- $/phantomx/j\_c1\_rm\_position\_/controller/command$: Contains a typed Float64 and the information about the desired hexapod leg joints.
- $/phantomx/cmd\_vel$: Contains a message in the Twist format and the information about the linear and angular hexapod velocity.