CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

DEPARTAMENTO DE CONTROL AUTOMÁTICO

"Sobre el Control Robusto de Humanoides y MAVs en Tareas Complejas"

T E S I S

Que presenta

SANTOS MIGUEL OROZCO SOTO

Para obtener el grado de

DOCTOR EN CIENCIAS

EN LA ESPECIALIDAD DE CONTROL AUTOMÁTICO

Director de la Tesis:
Dr. Juan Manuel Ibarra Zannatha

Ciudad de México FEBRERO, 2020

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

AUTOMATIC CONTROL DEPARTMENT

"On Robust Control of Humanoids and MAVs in Complex Tasks"

T H E S I S

That presents

SANTOS MIGUEL OROZCO SOTO

To obtain the degree of

DOCTOR OF SCIENCE

WITH SPECIALITY IN AUTOMATIC CONTROL

Thesis Supervisor:
Dr. Juan Manuel Ibarra Zannatha

México City FEBRUARY, 2020

Resumen

En este trabajo se presenta el diseño y la implementación de técnicas de control robusto para mejorar el comportamiento de robots autónomos, tales como humanoides y micro-vehículos aéreos tetrarrotores. Estos robots son sistemas dinámicos complejos que deben lograr tareas dentro de entornos impredecibles, por lo tanto, están sujetos a perturbaciones desconocidas que podrían afectar su rendimiento o el cumplimiento de sus tareas. Por tal motivo, se implementaron técnicas de control robusto capaces de contrarrestar la incertidumbre de la dinámica y las perturbaciones desconocidas, además de ser mejoradas utilizando técnicas clásicas de control y observación robustas. Los controladores propuestos fueron probados en simulaciones y en humanoides y tetrarrotores reales, mostrando resultados satisfactorios en problemas de regulación y seguimiento, utilizando realimentación inercial y de visión artificial. La tesis resultante de este trabajo de investigación está escrita en idioma inglés, con el objetivo de difundirla internacionalmente; por tal motivo, se presentan a continuación, brevemente, los objetivos, problemática e hipótesis del trabajo de investigación, así como los logros.

Objetivos de Investigación

Objetivo General

El objetivo general de este trabajo de investigación es implementar estrategias de control en lazo cerrado, para contrarrestar perturbaciones desconocidas que afectan el comportamiento de robots humanoides y micro-vehículos aéreos autónomos que interactuan con un entorno no estructurado, utilizando controladores robustos que no requieren un conocimiento preciso de sus modelos dinámicos, ni tampoco the las perturbaciones mencionadas.

Objetivos Específicos

- Desarrollar modelos dinámicos de nuestros humanoides y micro-vehículos aéreos para su posterior uso en el diseño de sistemas de control, utilizando técnicas revisadas en la literatura.
- Diseñar técnicas de control en lazo cerrado para tareas complejas de humanoides y micro-vehículos aéreos basadas en avances recientes en controladores robustos, garantizando estabilidad a pesar de perturbaciones e incertidumbre.
- Implementar los controladores diseñados en humanoides y micro-vehículos aéreos reales, utilizando herramientas especializadas de software para sistemas robóticos.
- Comparar los controladores implementados contra otras técnicas no robustas existentes en la literatura, para mostrar la capacidad de rechazo a perturbaciones e incertidumbre de dichos controladores.

Problemática

Tanto los robots humanoides como los micro-vehículos aéreos son sistemas robóticos que operan dentro de entornos no estructurados, por lo tanto, son susceptibles a ser perturbados aleatoriamente, lo cual puede provocar el incumplimento o cumplimiento incompleto de sus tareas asignadas. Además, es complicado obtener modelos dinámicos precisos de sistemas robóticos reales, incluyendo a los humanoides y micro-vehículos aéreos, los cuales son útiles para el desarrollo de sistemas de control.

Hipótesis

El uso de una estrategia de control capaz de contrarrestar incertidumbre y perturbaciones en linea, que además no requiera el conocimiento preciso de la dinámica del sistema, es una solución apropiada para el cumplimiento de las tareas ejecutadas por los sistemas robóticos mencionados.

Solución propuesta

Para satisfacer los objetivos de investigación, se implementó una técnica de control recientemente aprovechada, llamada control con rechazo activo de perturbaciones (ADRC, por sus siglas en inglés), para controlar sistemas robóticos. Dicha técnica permite el rechazo, en tiempo real, de perturbaciones, por medio de la estimación en linea de incertidumbres dinámicas y dichas perturbaciones, gracias a un observador de estado extendido. El control ADRC no requiere un modelo dinámico preciso de la planta, ni un conocimiento *a priori* de las señales de perturbación, solo sus cotas superiores y el orden aproximado del sistema. Además, se implementaron observadores robustos para mejorar el rendimiento del ADRC. Para el caso de las tareas con múltiples objetivos para robots humanoides, se implementó un controlador cuadrático ponderado, el cuál fue mejorado con una técnica de control robusto que solo requiere de las señales de error para funcionar: el algoritmo de modos deslizantes *super-twisting*.

Logros del trabajo de investigación

El trabajo desarrollado consistió en diseñar e implementar sistemas de control robusto que son útiles para el cumplimiento de tareas complejas de robots humanoides y micro-vehículos aéreos tetrarrotor autónomos. Para cumplir los objetivos de investigación mencionados, se llevaron a cabo las siguientes actividades:

Actividades técnicas

- Revisión de literatura relacionada con los avances en la teoría del control robusto.
- Análisis y desarrollo de técnicas de control robusto.
- Implementación de algoritmos de realimentación y controladores utilizando lenguajes de programación y herramientas de desarrollo de software para robots humanoides y micro-vehículos aéreos.

- Uso de sofware especializado para robótica: *Robotics Operating System* y *mc_rtc*.
- Implementación de software de simulación utilizado a nivel internacional como *Chorenoid*, *RVIZ*, *Gazebo*.
- Asistencia a competencias nacionales de tetrarrotores autónomos.

Logros

- Desarrollo de modelos matemáticos de nuestros robots humanoides.
- Mejora del observador de estado extendido utilizando el enfoque de modos deslizantes.
- Implementación exitosa de los controladores robustos ADRC y modos deslizantes *super-twisting* embebido en un control QP, para el control del equilibrio y caminado de robots humanoides, y su equilibrio sobre un balancín.
- Desarrollo de modelos matemáticos de nuestros tetrarrotores.
- Implementación exitosa del control ADRC para la suspensión, regulación de pose y navegación autónoma de drones tetrarrotor.
- Implementación de técnicas de realimentación visual e inercial para los sistemas de control.

Publicaciones

[1] Orozco-Soto, Santos M., and Juan M. Ibarra-Zannatha. "Motion control of humanoid robots using sliding mode observer-based active disturbance rejection control." 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC). IEEE, 2017.

[2] Orozco-Soto, Santos M., R. S. Núñez-Cruz, and J. M. Ibarra-Zannatha. "Active disturbance rejection control for humanoid stable walking." 2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE, 2016.

[3] Orozco-Soto, Santos M., and Juan Manuel Ibarra Zannatha. "Control con rechazo activo de perturbaciones para el equilibrio de robots humanoides." Research in Computing Science 135 (2017): 159-171.

[4] Orozco-Soto, Santos M., and Juan Manuel Ibarra Zannatha. "Robust walking of humanoid robots using active disturbance rejection control." International Journal of Humanoid Robotics (Submitted).

[5] Orozco-Soto, Santos M., et al. "Active disturbance rejection control for UAV pose regulation." 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE, 2017.

[6] Orozco-Soto, Santos M., et al. "Active Disturbance Rejection Control for UAV Hover using ROS." 2018 XX Congreso Mexicano de Robótica (COMRob). IEEE, 2018.

[7] Orozco-Soto, Santos M., Pablo Vera-Bustamante, and Juan M. Ibarra-Zannatha. "ORB-SLAM Based Active Disturbance Rejection Control for Quadrotor Autonomous Flight." 2018 XX Congreso Mexicano de Robótica (COMRob). IEEE, 2018.

Abstract

This thesis work presents the design and implementation of robust control techniques to improve the behaviour of autonomous robots, such as humanoids and quadrotor micro-aerial vehicles. These robots are complex dynamic systems that must accomplish tasks within unpredictable environments, therefore, they are subjected to unknown disturbances that might affect its performance or its tasks fulfilment. Hence, an innovative robust control strategy that is capable to deal with uncertain dynamics and unknown disturbance is implemented and improved using classical robust control and observation techniques. The proposed controllers were tested in simulation and actual humanoid robots, as well as actual quadrotors, showing successful results in regulation and tracking control problems by means of inertial, force and visual feedback.

Dedicatorias

Dedico esta tesis a mis tías, quienes me han procurado durante toda mi vida; a mis padres, cuyo apoyo y cariño ha sido fundamental para mi desarrollo; a mi esposa, que estuvo conmigo desde el inicio de este trabajo y me continúa apoyando incondicionalmente; a mi hermano y a mi familia, quienes siempre me manifiestan su apoyo bajo cualquier circunstancia; a mi asesor, quién no solo es asesor técnico, sino también es consejero y amigo.

Dedications

I dedicate this thesis to my aunts, who have took care of me during my whole life; to my parents, whose support and love has been fundamental for my development; to my wife, who has been by my side since the beginning of this work, and keeps supporting me unconditionally; to my brother and to my family, who always express their support under any circumstance; to my supervisor, who is not only technical supervisor, but also is advisor and friend.

Agradecimientos

En este texto manifiesto mi agradecimiento a los profesores, personal administrativo y de mantenimiento del Departamento de Control Automático y de todo el CINVESTAV, ya que contribuyeron significativamente a mi formación con su profesionalismo y responsabilidad. También quiero agradecer a mis compañeros, con quienes pasé tiempo compartiendo las instalaciones, clases e interesantes pláticas tanto técnicas como ordinarias. Asimismo, estoy agradecido con mi asesor, quién me dio la oportunidad de estudiar un doctorado dentro de su equipo; el siempre me ofreció su apoyo, oportunidades, y útiles consejos para mi trabajo de investigación y para la vida. También quiero agradecer a mi familia y a mi esposa, ellos siempre me apoyaron incondicionalmente durante mis estudios de doctorado. Finalmente, quiero reconocer al CONACYT por otorgarme una beca y por financiar mis estancias de investigación.

Acknowledgements

Hereby I state my gratefulness to professors, administrative and maintenance staff of the Automatic Control Department, and of the whole Research and Advanced Studies Center, since they contributed significantly in my formation with their professionalism and responsibility. I would also like to thank my classmates, with whom I spent many time sharing the facilities, classes, and both technical and daily interesting talks. Likewise, I am grateful with my supervisor, who gave me the opportunity of studying a Ph. D. within his team; he always offered me his support, opportunities and useful advices for my research work and also for life. I also want to thank my family and my wife, they always supported me unconditionally during my Ph. D. studies. Finally, I acknowledge the National Science and Technology Council for granting me a scholarship and for supporting my research stays.

Contents

1

Introduction

The use of robotic systems is growing so quickly nowadays due to the demand of automation, precision, efficiency and accuracy of many productive tasks. Service robots are now taking more attention from industry and service companies for different purposes than typical industrial applications, and the number of robots that interact within environments designed for human beings is increasing. The general purpose of a control system is to monitor and correct the actions of a determined dynamic system [1]. Since disturbances exist in all practical dynamic systems, the problem of disturbance attenuation or rejection has been a significant topic for many control field researchers, yielding mainly two approaches to deal with disturbances: attenuation methods and rejection techniques [1], for which several theories have been developed. Talking specifically, modern service robots, like humanoids or micro-aerial vehicles (MAVs), are constantly subjected to disturbances inherently to their interaction within environments designed for human beings; in that sense, control systems that enable the robots to perform their tasks correctly and safely are required. The purpose of this research work is to develop automatic control schemes, that deal with the problem of disturbances affecting the successful accomplishment of complex tasks performed by both, humanoids and MAVs. For the case of humanoid robots, the robustness of balance and walking problems were addressed, meanwhile, the robust autonomous flight of quadrotor MAVs problem was tackled. The selected control strategies were two: the active disturbance rejection control technique, which is a modern innovative strategy that takes advantage of canonical forms and the extended-state observer to estimate and cancel out disturbances on-line; and the super-twisting sliding mode control, in order to regulate the position of a see-saw mechanism using a humanoid robot as an actuator. During the research work, an important contribution to the control theory is the improvement of active disturbance rejection technique by means of sliding-mode extended-state observers, yielding successful results for both control of humanoid and MAVs. Another important achievement within the research was the implementation of the mentioned controllers, using new software tools for modern robotic systems such as *mc_rtc*, designed for multi-contact and multitask

optimal control of humanoid robots, and the so-called *Robotics Operating System*, a middleware helpful for multithreading tasks for robots. In addition, computer vision systems and sensor-fusion algorithms were implemented in order to develop the real-time robust controllers for the robots, yielding successful results that were published on international conferences and journals. This thesis is organized as follows: Chapter 1 introduces the reader to the problems of disturbances in control systems for complex tasks of robots. Chapter 2 is an overview of the robust control theory used in the research work. Chapter 3 addresses the modelling of humanoids and quadrotors for control synthesis. Chapter 4 describes the implementation of the robust controllers for different tasks and robots. Finally, conclusions are stated in Chapter 5.

1.0.1 Humanoid Robots as Control Systems

Humanoid robots are called, at least in theory, to develop almost any activity performed by human beings. Nevertheless, in order to humanoids perform such tasks, they need to be able to walk in environments where humans do: flat or uneven terrains with obstacles, climbing ramps or stairs, among other possibilities. Walking means to displace at a moderate speed lifting alternately both feet and landing them in a few distance forward; each step of the humanoid gait is compound by the double support phase and the swing phase. Within the field of modelling and control of dynamic systems, the main target is to develop a stable dynamic walking which must be robust upon disturbances that allows the humanoid robot to move with the nearest agility to the human being. It is almost impossible to copy directly the human motion to the robot because it has many less degrees of freedom (DOF) and actuators than humans [2]. The walking tasks that humanoid robots perform consist, firstly, on the generation of stable walking patterns supplied to the robot and, then, optionally, the use a of a feedback stabilizer to ensure the correct execution of those walking patterns by means of disturbance rejection, as is depicted in Fig. 1.1 [3]. Several disturbance rejection techniques to guarantee humanoid stable walking have been developed and implemented successfully [4] - [5], most of them demand an accurate dynamic model of the robot or a precise knowledge of the disturbance to be rejected, increasing the complexity of the control system design. In this thesis work, the development of robust controllers for humanoid balance and walking inertial and visual feedback techniques is addressed.

1.0.2 Unmanned Air Vehicles from Control Systems Perspective

An unmanned air vehicle (UAV) -or drone- is an aircraft with no on board pilot, that might be controlled remotely by a human operator or by an external computer; nevertheless, sometimes autonomous UAVs have powerful computers on board. Compared to other complex robotic systems, UAVs are simple to build; they are compound

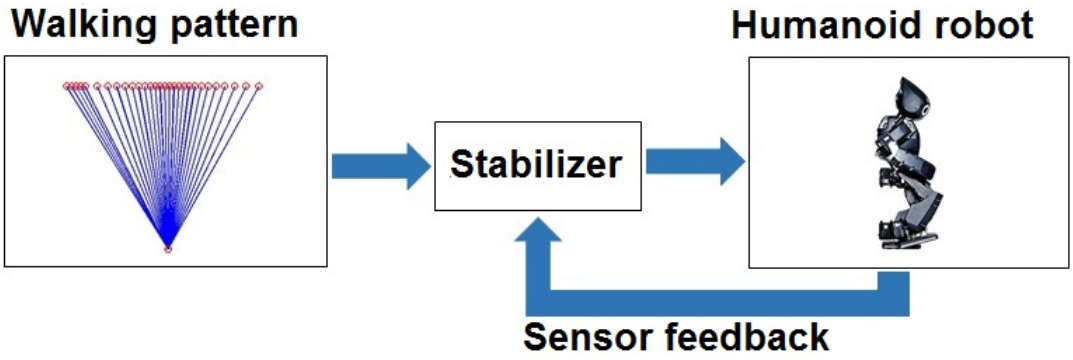

Figure 1.1: Diagram of feedback control of humanoid robots gait.

by a frame, motors, propellers, electronic drivers, speed controllers, the battery, a flight controller and wireless communication electronics [6]. There are several types of drones such as fixed-wing, rotorcrafts, flapping wing, among other varieties [7] shown in Fig. 1.2, and their use is nowadays extended to agriculture, filming, construction, entertainment, evaluating and delivering applications [6]. One of the most popular type of UAV is the rotorcraft, which can also be classified according to the number of rotors and their set up [8], and within this classification, quadrotors have become widely used recently, since they offer better payload capability than other rotorcrafts, in addition to their building and control simplicity, attenuation of gyroscopic effects [9], small size, low cost, agility, manoeuvrability and hovering at fixed point [10], among other advantages. However, quadrotors have many drawbacks from the control systems perspective: highly coupled dynamics [11], underactuation, open-loop instability, difficult parametric identification of dynamic model and noisy measurements are some reasons that led many control system designers to be particularly interested in the development of techniques capable to deal with such control challenges. In this context, many proposals have been developed and reported in literature such as PID-type controllers, adaptive approaches, feedback linearization, backstepping, intelligent control schemes, among many other strategies which have shown successful results in dealing with above mentioned control problems, but also presenting some drawbacks according to the nature of the control solution for the given problems [11] [10] [12] [13].

1.0.3 Disturbance Rejection for Robotic Systems

Most of real-life control systems are affected by unconsidered dynamics, parametric uncertainties of their mathematical models and, of course, external disturbances, which consequently lead to different implementation challenges for control engineers [14], specially robotic systems, which are developed to work within environments designed for human beings. A relatively recent control approach that allows real-time

Figure 1.2: Different types of drones.

disturbance rejection that does not require an accurate dynamic model of the plant or a precise knowledge of the disturbance signals, is the active disturbance rejection control (ADRC) [15], which has become popular among control systems developers and has been implemented in many kind of control applications [16] [17], since it has several advantages such as the ability in dealing with a vast range of uncertainties, great transient response, easy implementation and energy savings, among other interesting benefits [14]; all those capabilities are possible since ADRC is based on two of the best offerings of control theory: canonical forms and state observers [14]. An important feature of ADRC is that uses the state augmentation to estimate parameters and disturbances embedded as an additional state, but it is required to implement an extended-state observer that converges quickly to have a suitable control performance [18]. Sliding observers are preferred by many control designers since they offer similar advantages to those of sliding controllers, such as robustness upon parametric uncertainties, high control (observation) authority and easy implementation; furthermore, chattering inconveniences in sliding observers represent only numerical

implementation matters rather than physical problems [19]. It worth to mention that the robustness of sliding observers against parametric uncertainties and noise measurement is bigger than Luenberger observers and the extended Kalman filter [19], however, sliding observers are susceptible to input disturbances in the plant, but if the mentioned disturbance signals are bounded, they can still be overcome for state tracking [20]. ADRC has already been used successfully in robotics applications like control of mobile robots [21] and also in walking control of humanoid robots, reporting successful simulation results [16],[22]. This thesis work presents the implementation of continuous-time sliding and discrete-time *quasi sliding* observers [23], to implement ADRC in humanoid robots and quadrotors, showing successful simulation and real-time implementation results.

1.1 Research Objectives

1.1.1 Main Objective

The main objective of this research work is to implement closed-loop control strategies, in order to deal with unknown disturbances that affect the performance of humanoid robots and autonomous micro-aerial vehicles (MAVs) that interact with a non structured environment, using robust control approaches that do not require a precise knowledge of their dynamic models, neither the mentioned disturbances.

1.1.2 Specific Objectives

- To develop dynamic models for our humanoids and micro-aerial vehicles for their later use in control systems synthesis, by means of techniques reviewed in the literature.

- To design closed-loop control techniques for complex tasks of humanoids and micro-aerial vehicles based on recent advances in robust controllers, guaranteeing stability despite of uncertainties or disturbances.

- To implement the designed controllers in actual humanoids and micro-aerial vehicles, using specialized software development tools for robotic systems.

- To compare the implemented controllers against other non-robust techniques existing in the literature, showing the disturbances and uncertainties rejection capability.

1.2 Research Problem and Hypothesis

Both humanoid robots and MAVs are robotic systems that work within unstructured environments, therefore, they are susceptible to be disturbed randomly, which may lead to an unsuccessful or incomplete accomplishment of their assigned tasks. Furthermore, it is difficult to obtain an accurate dynamic model of actual robotic systems, including humanoids and MAVs, that can be useful for control systems development.

Hypothesis: The use of a robust control strategy capable to deal with uncertainties and disturbances on-line, and that neither require an accurate knowledge of the system dynamics, is an appropriate solution for the accomplishment of the tasks performed by the mentioned robotic systems.

1.3 Proposed Solution

In order to accomplish the research objectives, a recently exploited control approach called active disturbance rejection control (ADRC) was implemented to control robotic systems. Such technique allows real-time disturbance rejection by means of the on-line estimation of the disturbances and uncertainties by means of an extended-state observer (ESO). The ADRC does not require an accurate dynamic model of the plant or a precise knowledge of the disturbance signals, but only the order of the system and the upper bounds of the disturbances. Furthermore, robust observers were implemented to improve the performance of the ADRC. For the case of multi-objective tasks for humanoid robots, a weight prioritized quadratic programming (QP) control was implemented and enhanced with a robust control technique that only require the error signals to work, that is the case of the super-twisting sliding mode algorithm.

1.4 Achievements of the Research Work

The development of this research work consisted on the design and implementation of robust control systems that are helpful to accomplish complex tasks of humanoid robots and autonomous quadrotor MAVs. In order to fulfil the listed research objectives, the following activities were carried on:

Technical Activities

- Literature review about advances in robust control theory.
- Analysis and development of robust control techniques.
- Implementation of feedback algorithms and controllers using programming languages and development tools for humanoid robots and quadrotor MAVs.

- Use of specialized software for robotics: *Robotics Operating System* and *mc_rtc*.
- Implementation of worldwide-use simulation software such as *Chorenoid*, *RVIZ*, *Gazebo*.
- Attending to national competitions of autonomous quadrotors.

Achievements

- Development of mathematical models of our humanoid robots.
- Improvement of extended-state observer using sliding mode approach.
- Successful implementation of robust controllers ADRC and super-twisting sliding mode embedded in QP control for humanoid robots balance, walking and balancing on a see-saw.
- Development of mathematical models of quadrotors.
- Successful implementation of ADRC for hover, pose regulation and autonomous navigation of quadrotor drones.
- Implementation of inertial and visual feedback techniques for the control systems.

Publications

[1] Orozco-Soto, Santos M., and Juan M. Ibarra-Zannatha. "Motion control of humanoid robots using sliding mode observer-based active disturbance rejection control." 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC). IEEE, 2017.

[2] Orozco-Soto, Santos M., R. S. Núñez-Cruz, and J. M. Ibarra-Zannatha. "Active disturbance rejection control for humanoid stable walking." 2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE, 2016.

[3] Orozco-Soto, Santos M., and Juan Manuel Ibarra Zannatha. "Control con rechazo activo de perturbaciones para el equilibrio de robots humanoides." Research in Computing Science 135 (2017): 159-171.

[4] Orozco-Soto, Santos M., and Juan Manuel Ibarra Zannatha. "Robust walking of humanoid robots using active disturbance rejection control." International Journal of Humanoid Robotics (Submitted).

[5] Orozco-Soto, Santos M., et al. "Active disturbance rejection control for UAV pose regulation." 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE, 2017.

[6] Orozco-Soto, Santos M., et al. "Active Disturbance Rejection Control for UAV Hover using ROS." 2018 XX Congreso Mexicano de Robótica (COMRob). IEEE, 2018.

[7] Orozco-Soto, Santos M., Pablo Vera-Bustamante, and Juan M. Ibarra-Zannatha. "ORB-SLAM Based Active Disturbance Rejection Control for Quadrotor Autonomous Flight." 2018 XX Congreso Mexicano de Robótica (COMRob). IEEE, 2018.

1.5 Collaboration Work

Some collaboration with other national and international institutions in the field of robotics were also developed with the following succesful results.

RWTH Aachen, Germany

Project: Development of fuzzy models and visual technology for spasticity detection.
Publications:

[9] Orozco-Soto, S. M., et al. "Development of a Visual-Inertial Motion Tracking System for Muscular-Effort/Angular Joint-Position Relation to Obtain a Quantifiable Variable of Spasticity." Wearable Robotics: Challenges and Trends. Vol. 22. Springer, 2018. pp. 210–215.

[10] Orozco-Soto, Santos M., et al. "Modelo neurodifuso para el control de un exoesqueleto para rehabilitación de brazo en pacientes con EVC." Research in Computing Science 148 (2019): 267-275.

CINVESTAV, Monterrey, Mexico

Project: Development of a visual-servoing system for a robotic arm in a semi-autonomous BCI.
Publications:

[11] Ramírez-Moreno, M. A., Orozco-Soto, S. M., Ibarra-Zannatha, J. M., and Gutiérrez, D. "Artificial Vision Algorithm for Object Manipulation with a Robotic Arm in a Semi-Autonomous Brain-Computer Interface." Wearable Robotics: Challenges and Trends. Vol. 22. Springer, 2018. pp. 187–191.

Laboratoire d' Informatique, de Robotique et de Microelectronique de Montpellier, Montpellier, France

Project: Multirobot-based robust control of humanoid robots in task space.

Universidad del Cauca, Popayan, Colombia

Project: Control of planar bipeds in joint space.

2

On Robust Control and Observation

2.1 Generalities on Active Disturbance Rejection Control

Active Disturbance Rejection Control (ADRC) is a control strategy that has become popular among control engineers and researchers, since it has the capability to deal with a vast range of uncertainties and disturbances with the advantages of easy implementation and energy savings [14]. The baseline of ADRC is the exploitation of two of the best offerings of modern control theory: the canonical form representation and the state observer [14].

2.1.1 Fundamentals of ADRC

Consider a dynamic system of the form:

$$\begin{aligned} \dot{x} &= f(t, x, u, w) \\ y &= h(t, x, u, w) \end{aligned} \tag{2.1}$$

Where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^p$ is the control inputs vector, $w \in \mathbb{R}^q$ is the disturbances vector, usually unknown, but bounded, and y is the output to be analysed. Independently from its linear or nonlinear nature, the stated system (2.1) can be represented as a cascade of integrators canonical form under some conditions, like by means of a diffeomorphic transformation $z = T(x)$ as follows:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= g(t, x, u, w) \\ y &= h(z) \end{aligned} \tag{2.2}$$

Where $g(t, x, u, w)$ can be considered as a total disturbance that embeds external disturbances, unconsidered dynamics and parametric uncertainties. Therefore, an Extended-State Observer (ESO) can be proposed for state tracking and total disturbance estimation as follows:

$$\begin{aligned} \dot{\hat{z}}_1 &= \hat{z}_2 - l_1(\hat{e}) \\ &\vdots \\ \dot{\hat{z}}_{n-1} &= \hat{z}_n - l_n(\hat{e}) \\ \dot{\hat{z}}_n &= g(\hat{z}, t, u, w) - l_{n+1}(\hat{e}) \end{aligned} \tag{2.3}$$

Note that $l(\hat{e})$ is a vector function which leads $\hat{z} \to z$ asymptotically with an observation error $\hat{e} = \hat{z} - z$. In order to cancel the total disturbance effect, the following controller can be implemented using the information cast from the ESO:

$$u = u_0(t, x, r) - \hat{z} \tag{2.4}$$

Where $u_0(t, x, r)$ is a controller, that considers $x = T^{-1}(z)$, for the n-th order cascade of integrators. The general design procedure for ADRC can be stated in three steps [14]

1) Select the canonical form of the plant in the form of cascade integrators.

2) Consider the different part from the canonical form as a total disturbance.

3) Reduce the system to a cascade of integrators by estimating and compensating the total disturbance via the ESO.

2.1.2 ADRC for Disturbed Linear Systems

Consider the following LTI disturbed dynamic system:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew \\ y &= Cx \end{aligned} \tag{2.5}$$

Where $x \in \mathbb{R}^n$ is the state, $w \in \mathbb{R}^p$ is the disturbance vector, usually unknown but bounded, and $E \in \mathbb{R}^{n \times p}$ is the disturbance mapping matrix to each degree of freedom of (2.5). The central idea of linear ADRC is to estimate the unknown generalized disturbance by means of an extended state observer (ESO), so that, the following extended-state representation using $\xi = [x, \dot{x}, f(t, x, u, w)]^T$ is implemented [24]:

$$\begin{aligned} \dot{\xi} &= A_e \xi + B_e u \\ y &= C_e \xi \end{aligned} \tag{2.6}$$

where $A_e \in \mathbb{R}^{(n+q)\times(n+q)}$, $B_e \in \mathbb{R}^{(n+q)}$ and $C_e \in \mathbb{R}^{1\times(n+q)}$, q is the order of the augmented state [24]. It worth to remark that A_e is in controllable canonical form.

Hence, the following ESO for (2.6) is proposed:

$$\begin{array}{c} \dot{\hat{\xi}} = A_e\hat{\xi} + B_e u + L(y - \hat{y}) \\ y = C_e\xi \end{array} \tag{2.7}$$

the observer gains can be tuned using only the observation bandwidth ω_0 as follows [24]:

$$L = \begin{bmatrix} 3\omega_0 & 3\omega_0^2 & \omega_0^3 \end{bmatrix} \tag{2.8}$$

Note that each entry of L is of n-th order; thus, if $(A_e - LC_e)$ is Hurwitz, $\hat{\xi}_1, ..., \hat{\xi}_{n-1}$ will approximate $\xi_1, ..., \xi_{n-1}$ and $\hat{\xi}_n$ will approximate the generalized disturbance $f(t, x, u, w)$; therefore, the estimated generalized disturbance can be used in a general controller of the form:

$$u = \bar{B}^{-1}(u_0 - \hat{\xi}_n) \tag{2.9}$$

Where $\bar{B}$ is an estimated of B and u_0 is any regulator for a chain of integrators, such as a PID type controller. Notice that a precise knowledge of the plant or the disturbances is not required, but only the order of the system and the upper bound of the disturbances acting on it.

2.1.3 Feedback Linearization of Nonlinear Systems using the Extended State Observer

Consider the following nonlinear dynamic system:

$$\begin{array}{c} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{array} \tag{2.10}$$

where x is the state vector, y is the output, u is the control signal, $f(x)$ and $g(x)$ are the n-dimensional vectors of continuous real-valued functions of x with partial derivatives of any order. It is well known from literature that there exists a diffeomorphic transformation such that [14]:

$$z = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} \tag{2.11}$$

Furthermore, 2.10 can be exactly linearized using the state feedback control law:

$$u = \frac{1}{b(z)}\left(-a(z) + v\right) \tag{2.12}$$

Where $a(z) = L_f^n h(x)$ and $b(z) = L_g L_f^{n-1} h(x)$. The result of exact feedback linearization is the following cascade of integrators:

$$\begin{array}{c} \dot{z} = Az + Bv \\ y = Cx \end{array} \tag{2.13}$$

Where [14]:

$$A = \begin{bmatrix} 0 & 1 & & \\ & 0 & & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T, C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

In order to perform such exact linearization technique, an accurate knowledge of the parameters of (2.10) is required. Nevertheless, the exact feedback linearization can be still accomplished by means of an ESO, without an accurate knowledge of the parameters of 2.10; consider the following cascade of integrators:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= \bar{B}u \end{aligned} \tag{2.14}$$

as a canonical form of (2.10), where $\bar{B}$ is an estimation of $B(x) = L_g L_f^{n-1} h(x)$. The remaining part of the system that is different from the system is:

$$F(x) + (B(x) - \bar{B})u \triangleq x_{n+1} \tag{2.15}$$

Where $F(x) = L_f^n h(x)$; furthermore, if the system is affected by external disturbances, (2.15) can be rewritten as:

$$F(x) + (B(x) - \bar{B})u + w \triangleq x_{n+1} \tag{2.16}$$

x_{n+1} is an extended-state of the system (2.14), for which an ESO can be implemented to estimate such extended state as follows [14]:

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 - \mathcal{G}_1(\hat{x}_1 - x_1) \\ &\vdots \\ \dot{\hat{x}}_n &= x_{n+1} - \mathcal{G}_n(\hat{x}_1 - x_1) + \bar{B}u \\ \dot{\hat{x}}_{n+1} &= -\mathcal{G}_{n+1}(\hat{x}_1 - x_1) \end{aligned} \tag{2.17}$$

Where $\mathcal{G}_i(\hat{x}_1 - x_1)$ is a nonlinear vector function such that $\hat{x} \to x$ as $t \to \infty$. This on-line estimation of the "total disturbance" (2.16) is used by the following ADRC to perform the state-feedback linerization:

$$u = -\bar{B}^{-1}\hat{x}_{n+1} + u_0 \tag{2.18}$$

Where u_0 is a linear controller to regulate (2.14). The success of (2.17) depends on the information of the sensor that measures x_1 and on the robustness of the ESO, therefore, in this thesis work, a proposal to improve the observed state tracking is implemented using robust sliding mode observers.

2.1.4 On the Convergence of the Extended-State Observer

Consider the following nonlinear multi-input multi-output (MIMO) uncertain disturbed system [25]:

$$\begin{aligned}
&x_1^{(n_1)}(t) = f_1(x_1(t), ..., x_1^{(n_1-1)}(t), ..., x_m^{(n_m-1)}(t), w_1(t)) + g_1(u_1(t), ..., u_k(t)) \\
&x_2^{(n_2)}(t) = f_2(x_1(t), ..., x_1^{(n_1-1)}(t), ..., x_m^{(n_m-1)}(t), w_2(t)) + g_2(u_1(t), ..., u_k(t)) \\
&\qquad\vdots \\
&x_m^{(n_m)}(t) = f_m(x_1(t), ..., x_1^{(n_1-1)}(t), ..., x_m^{(n_m-1)}(t), w_m(t)) + g_m(u_1(t), ..., u_k(t)) \\
&y_i(t) = x_i(t), \quad i = 1, 2, ..., m
\end{aligned} \tag{2.19}$$

Transforming (2.19) into a first-order system described by m number of subsystems of first-order differential equations [25]:

$$\begin{aligned}
&\dot{x}_{i,1}(t) = x_{i,2}(t) \\
&\dot{x}_{i,2}(t) = x_{i,3}(t) \\
&\qquad\vdots \\
&\dot{x}_{i,n_i}(t) = f_m(x_{1,1}(t), ..., x_{1,n_1}(t), ..., x_{m,n_m}(t), w_i(t) + g_i(u_1, u_2, ..., u_k) \\
&y_i(t) = x_{i,1}(t), \quad i = 1, 2, ..., m
\end{aligned} \tag{2.20}$$

Where $x_{i,j}(t) = x_i^{(j-1)}(t), \quad j = 1, 2, ..., n_i$. An ESO composed of m subsystem for (2.20) can be expressed as:

$$\begin{aligned}
&\dot{\hat{x}}_{i,1}(t) = \hat{x}_{i,2}(t) + \varepsilon^{n_i-1}\phi_{i,1}\left(\frac{x_{i,1}(t) - \hat{x}_{i,1}(t)}{\varepsilon^{n_i}}\right) \\
&\dot{\hat{x}}_{i,2}(t) = \hat{x}_{i,3}(t) + \varepsilon^{n_i-2}\phi_{i,2}\left(\frac{x_{i,1}(t) - \hat{x}_{i,1}(t)}{\varepsilon^{n_i}}\right) \\
&\qquad\vdots \\
&\dot{\hat{x}}_{i,n_i}(t) = \hat{x}_{i,n+1}(t) + \phi_{i,n_i}\left(\frac{x_{i,1}(t) - \hat{x}_{i,1}(t)}{\varepsilon^{n_i}}\right) + g_i(u_1, u_2, ..., u_k) \\
&\dot{\hat{x}}_{i,n_i+1}(t) = \frac{1}{\varepsilon}\phi_{i,n_i+1}\left(\frac{x_{i,1}(t) - \hat{x}_{i,1}(t)}{\varepsilon^{n_i}}\right), \qquad i = 1, 2, ..., m
\end{aligned} \tag{2.21}$$

The convergence of (2.21) is demonstrated on [25], and relies on the following assumptions:

Assumption (A1). For every $i = 1, 2, ..., m$, all u_i, w_i and the solution of (2.20) are bounded. Additionally, $g_i \in \mathcal{C}(\mathbb{R}^k, \mathbb{R})$ and $f_i \in \mathcal{C}^1(\mathbb{R}^{n_1+..+n_m+1}, \mathbb{R})$.

Assumption (A2). For every $i = 1, 2, ..., m$, there exists continuously differentiable, positive defined, radially unbounded functions $V_i, W_i : \mathbb{R}^{n_i+1} \to \mathbb{R}$ satisfying:

$$\sum_{l=1}^{n_i} \frac{\partial V_i}{\partial y_l}(y_{l+1} - \phi_{i,l}(y_1)) - \frac{\partial V_i}{\partial y_{n_i+1}}\phi_{i,n_i+1}(y_1) \leq -W_i(y) \tag{2.22}$$

Therefore, the following theorem ensures the convergence of ESO:

Theorem 2.1 *Considering assumptions (A1) and (A2), for any initial values of (2.20) and (2.21):*

i) For any $t > 0$:

$$\lim_{\varepsilon \to 0} |x_{i,j}(t) - \hat{x}_{i,j}(t)| = 0 \tag{2.23}$$

ii) For any $\varepsilon \in (0, 1)$:

$$|x_{i,j}(t) - \hat{x}_{i,j}(t)| \leq K_{i,j} \varepsilon^{n_i + 2 - j} \tag{2.24}$$

where $K_{i,j}$ are positive constants independent of ε, but depending on the initial values [25].

Theorem 2.1 is valid for both MIMO and SISO nonlinear or linear systems [18, 25].

2.2 Sliding Mode Observers

Let be the following dynamic system in companion form:

$$x^{(n)} = f(t, x, u, w) \tag{2.25}$$

Note that the parameters of $f(t, x, u)$ are not accurately known, but the mismatch $|\Delta f| = |f - \hat{f}|$ and w are upper bounded. The observation problem is to lead an estimated state $\hat{x}$ to track a specific state x despite model imprecisions and disturbances. Hence, the following sliding mode observer (SMO) for (2.25) based on the measurement of x_1 is proposed:

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 - (l_1 \hat{e}_1 + k_1 sign(\hat{e}_1)) \\ &\vdots \\ \dot{\hat{x}}_n &= \hat{f} - (l_n \hat{e}_1 + k_n sign(\hat{e}_1)) \end{aligned} \tag{2.26}$$

Where $\hat{e}_1 = x_1 - \hat{x}_1$ is the observation error of the single measured variable, $\hat{f}$ is an estimated of $f(t, x)$ and the constants l_i are chosen similarly to a classical Luenberger observer to ensure asymptotic error decay of a linearized representation. The resulting error dynamics can be expressed as [19]:

$$\begin{aligned} \dot{\hat{e}}_1 &= -l_1 \hat{e}_1 + \hat{e}_2 - k_1 sign(\hat{e}_1) \\ &\vdots \\ \dot{\hat{e}}_n &= -l_n \hat{e}_1 + \hat{f} - k_n sign(\hat{e}_1) \end{aligned} \tag{2.27}$$

A suitable alternative for the observer tuning is to choose $k_n \geq |\hat{f}|$, and keep a constant ratio with k_1; furthermore, the gains l_i determine the dynamic performance

of the observer in order to reach the following sliding patch [19]:

$$\left[\lambda I_{n-1} - \begin{pmatrix} -(l_2 + k_2/k_1) & 1 & 0 & \cdots & 0 \\ -(l_3 + k_3/k_1) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -(l_n + k_n/k_1) & 0 & 0 & \cdots & 0 \end{pmatrix}\right] \tag{2.28}$$

Where λI_{n-1} is the matrix of the $n-1$ poles of the linear part of the observer. In order to get a successful performance of the observer (2.26), it is recommended to pre-filter the measurement x_1 using, for example, a first-order filter with significant larger bandwidth than the frequency content of x_1, in addition, the sampling rate should be chosen to be consistent with the pre-filtering stage [19]. Note that multiple measurements can be included into the SMO to enhance the state tracking performance.

2.2.1 Some Aspects on Discrete-Time Variable Structure Systems

Even sliding mode control and observation algorithms are performed by hardware, normally they are designed and considered as continuous-time systems thanks to nowadays high-speed computers; however, for relatively slowly-sampled systems, it is convenient to consider a discrete-time analysis for a suitable control (or observer) design and performance. Consider a discrete-time plant:

$$x(k+1) = Ax(k) + bu(k) \tag{2.29}$$

When a discrete-time variable structure control (or observer) is applied to the plant (2.29), it earns the following properties [23]:

P1) The states will perform a trajectory from an initial condition towards a switching plane (or surface) and cross it in finite time.

P2) Once the trajectory crosses the plane, it will cross it again repeatedly every successive sampling period, resulting in a zigzag motion about the switching plane.

P3) The zigzag motion amplitude size is non-increasing and the trajectory remains within a band given by:

$$\{x| - \Delta < s(x) < \Delta\} \tag{2.30}$$

Where 2Δ is the width of the band.

The switching motion of a discrete-time variable structure system satisfying P2 and P3 is called *Quasi-sliding mode*, and the band (2.30) is known as *Quasi-sliding mode band*. A discrete-time variable structure system satisfies a *reaching condition* when

it has the properties P1, P2 and P3 [23]. Since the above properties are not always satisfied, a reaching law must be firstly specified to guarantee their fulfilment. This reaching law should govern the dynamics of a stable switching function $S(x) = 0$ [23]; a convenient reaching law may be:

$$s(k+1) - s(k) = -qT_s s(k) - \epsilon T_s sign(s(k)), \quad 1 - qT_s > 0 \tag{2.31}$$

Where $T_s > 0$ is the sample period, $\epsilon > 0$ and $q > 0$ are constant design parameters that affect the reaching mode response. Note that the choice of T_s is restricted since it guarantees the fulfilment of P1, and the presence of the $sgn(\cdot)$ term guarantees P2 and P3 [23].

2.2.2 Sliding Mode Extended-State Observers

Consider the sliding mode observer (2.26); according to the equivalent control method, an "average" dynamics of the system can be achieved from the high-frequency switching using a low-pass filter as follows:

$$\hat{f}(t, x, u, w) = \rho(\mathsf{sign}(\hat{e})) \tag{2.32}$$

Where $\rho(\mathsf{sign}(\hat{e}))$ is the low-pass filter. Nevertheless, information cast from (2.32) might not be as accurate as the total disturbance provided from an ESO. Hence, in order to enhance the ESO with robustness, an sliding mode part can be implemented as follows:

$$\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 + \varepsilon^{n-1}\left(\frac{l_1\hat{e}_1 + \mathsf{sign}(\hat{e}_1)}{\varepsilon^n}\right) \\
&\vdots \\
\dot{\hat{x}}_n &= \hat{x}_{n+1} + \left(\frac{l_n\hat{e}_1 + \frac{k_n}{k_1}\mathsf{sign}(\hat{e}_1)}{\varepsilon^n}\right) + u \\
\dot{\hat{x}}_{n+1} &= \varepsilon^{-1}\left(\frac{l_{n+1}\hat{e}_1 + \frac{k_{n+1}}{k_1}\mathsf{sign}(\hat{e}_1)}{\varepsilon^n}\right)
\end{aligned} \tag{2.33}$$

Invoking Theorem 2.1 and guaranteeing that the eigenvalues of the sliding patch (2.28), the convergence of the sliding-mode extended-state observer (SMESO) is ensured, either a robust total disturbance estimation and, consequently, a successful disturbance rejection part of an ADRC.

2.3 Quadratic Programming Control

Task-Space control of robotic systems is not trivial to solve, since it involves several kinematic, contacts, dynamics among other kinds of constraints that must be computed almost simultaneously, due to the interaction of the robots with their environment and other robots. In order to unify all tasks in a single optimization problem subject to constraints, the Quadratic Programming (QP) control has been proposed. Such controller reduces to the solution of a QP where multiple objective tasks are accomplished, minimizing the acceleration error of each task using the following cost function [26]:

$$\min_{x} \sum_{k=1}^{M} w_k \|\ddot{\Theta}_k - \ddot{\Theta}_k^d\| \tag{2.34}$$

where $x \in M$ is the constraint variables space and Θ_k, Θ_k^d are the acceleration of the tasks and desired accelerations respectively; hence, the solution of (2.34) lead to the derivation of the gains of linear control techniques in task space. QP control has been used widely in humanoid robots control, since the complex tasks involve multiple objectives to fulfill in real time [27]. Nevertheless, the QP is based on the models of the robots and tasks, therefore, it would be some mismatches, inaccuracies or disturbances that might affect the control performance or even the tasks accomplishment.

2.4 Super Twisting Sliding Mode Control

Sliding-mode controllers (SMC) are well known by their robustness upon parametric uncertainty of the controlled systems and against bounded disturbances [28], yielding successful regulation and tracking performances. Some improvements to the SMC have been developed to reduce *chattering*; among more popular is the Super-Twisting SMC (STSMC), which is a kind of second-order sliding-mode algorithm firstly designed to compensate Lipschitz disturbances exactly, ensuring finite time convergence [29]. This controller has been proposed in order to substitute discontinuous controllers by continuous ones, reducing the chattering effect of the sliding modes. In addition, finite time convergence, and tuning methods have been developed using geometric, homogeneity and strict-Lyapunov approaches to ensure the stability of such controllers [29]. The classical STSMC is expressed as:

$$\begin{aligned} u(t) &= u_1(t) + u_2(t), \\ u_1(t) &= -k_1|\sigma|^{1/2}\mathtt{sign}(\sigma), \\ u_1(t) &= -k_2\mathtt{sign}(\sigma) \end{aligned} \tag{2.35}$$

Where $k_1, k_2 > 0$. The controller (2.35) is robust against a disturbance $w(t)$ with bounded derivative $|\dot{w}(t)| \leq L$, and do not require the derivative of the state variable σ for its implementation [29].

3

Modelling of Humanoids and Quadrotors

3.1 Modelling of Humanoid Robots

A humanoid robot is a highly nonlinear system with coupled complex dynamics, which is also compound by multiple DOF, yielding an arborescent kinematic chain. Therefore, accurate modelling of humanoid robots results a very complex task. Even though the equations of motion of the humanoids can be developed using the Euler-Lagrange methodology, coarse graining methods are also popular to approach their dynamics behavior in order to have a useful representation for control strategies synthesis. In this section, most common mathematical representations of humanoid robots kinematics and dynamics are presented. First, the problem of walking pattern generation is addressed in order to generate the Cartesian trajectories that the body of the humanoid must follow to perform its gait. Therefore, the inverse kinematics is required to transform the Cartesian trajectories into desired joint trajectories, hence, a geomteric approach was followed to derive the equations. Finally, different approaches were used to yield the humanoids dynamics depending on the control target, from very simple and approximated models, to complex Lagrangian and spatial vector based approaches.

3.1.1 Walking Pattern Generation using the Three-Dimensional Linear Inverted Pendulum Model

Consider the humanoid robot shown in Fig. 3.1, where it can be observed that such three-dimensional linear inverted pendulum (3D LIPM) model is compound by the total Center of Mass (CoM) of the robot attached to a massless prismatic joint, that rotates freely about a supporting point representing the ankle joint during the single support phase [16]. The motion of the CoM at x and y directions is constrained by the plane $z = k_x x + k_y y + z_c$; since the motion is consider purely horizontal,

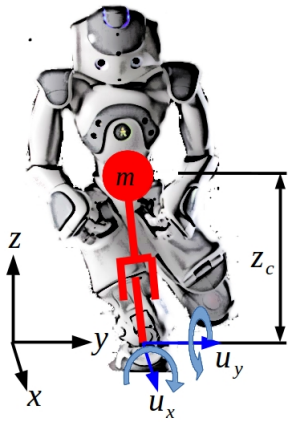

Figure 3.1: Humanoid represented as a 3D LIPM system.

then the slopes are $k_x \equiv k_y \equiv 0$. Hence, the 3D LIP dynamics is given by:

$$\begin{aligned} \ddot{x} &= \tfrac{g}{z_c} x + \tfrac{u_x}{m z_c} \\ \ddot{y} &= \tfrac{g}{z_c} y + \tfrac{u_y}{m z_c} \end{aligned} \tag{3.1}$$

Where g is the gravity acceleration, (x, y) is the CoM motion Cartesian coordinate and (u_x, u_y) are the control inputs incoming from *roll* and *pitch* torques from the ankle, however, such torques are very limited to stabilize the whole humanoid, so they are not considered [16]. Therefore, in order to perform the walking pattern, the 3D LIP dynamics (3.1) is integrated yielding a *walking primitive* taking advantage of its saddle point behavior.

Definition 3.1 ***Walking Primitive.*** *A walking primitive is a piece of a 3D linear inverted pendulum trajectory, which is symmetric about y axis and it is defined within the period* $[T_0, T_{sup}]$*, where* T_{sup} *is the duration of the supporting phase.*

A *walking primitive* describes the Cartesian trajectory of the CoM from a point $(-\bar{x}, \bar{y})$ to $(\bar{x}, \bar{y})$. A set of walking primitives is called a *walking trajectory*, and can be appreciated in Fig. 3.2, where both a single *walking primitive* and a *walking trajectory* are displayed; for each step of the humanoid walking trajectories, the following initial conditions to solve (3.1) must be stated:

- Initial position in x:
$$\bar{x}^{(n)} = S_x^{(n+1)}/2 \tag{3.2}$$
- Initial position in y:
$$\bar{y}^{(n)} = (-1)^n S_y^{(n+1)}/2 \tag{3.3}$$
- Initial velocity in x:
$$\bar{v}_x^{(n)} = \bar{x}(C+1)/T_c S \tag{3.4}$$

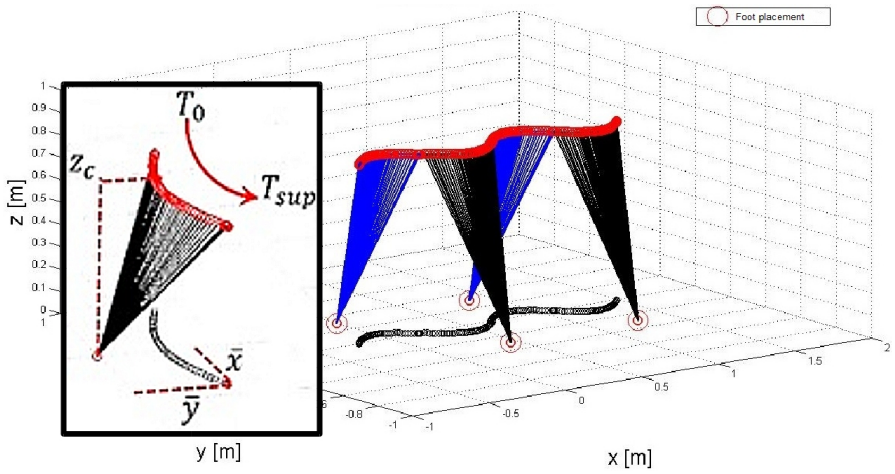

Figure 3.2: Walking primitive with its parameters and walking trajectory. The change of color means the suportting leg switching.

- Initial velocity in y:

$$\bar{v}_y^{(n)} = \bar{y}(C-1)/T_c S \tag{3.5}$$

Where $S_x^{(n+1)}$ is the length of the step, $S_y^{(n+1)}$ is the width of the step $T_c = \sqrt{z_c/g}, C = \cosh(T_{sup}/T_c), S = \sinh(T_{sup}/T_c)$ and z_c is the desired CoM height along the trajectory. Finally, the initial conditions to solve (3.1) are:

$$\mathbf{x}_0^{(n)} = (-\bar{x}^{(n)}, \bar{v}_x^{(n)}, \bar{y}^{(n)}, -\bar{v}_y^{(n)}) \tag{3.6}$$

It worths to consider the following remarks for a suitable walking pattern (trajectory) generation [16]:

Remark 3.1 *In order to perform a continuous CoM trajectory, and consequently, continuous joint trajectories, the length of steps should be based on the size of the legs of the humanoid robot.*

Remark 3.2 *The walking primitives only depend on the initial conditions of position and velocity, since the 3D LIP moition is generated intrinsically by its unstable saddle-point nature, so that, no control inputs supplied by the ankles are required.*

Hence, integrating (3.1) along $[-T_{sup}/2, T_{sup}/2]$ with initial conditions (3.6) yields the following hip (CoM) trajectory of the humanoid robot to perform a step:

$$\begin{bmatrix} x_h(k) & y_h(k) & z_c(k) \end{bmatrix} \in \mathbb{R}^{k\times 3} \tag{3.7}$$

Where k is the number of samples within the integration time interval. In order to compute the Cartesian trajectory of the swing ankle, the following expressions can be implemented:

$$x_a(k) = \bar{x}\sin\left(\frac{\pi k T}{T_{sup}}\right) \tag{3.8}$$

$$z_a(k) = \frac{H_s}{2}\left(1 + \sin\left(\frac{2\pi kT}{T_{sup}} + \frac{\pi}{2}\right)\right) \tag{3.9}$$

Where H_s is the maximum height of the swing foot and kT is the time at sample k. Fig. 3.3 shows both Sagittal and frontal planes of a humanoid robot, where the step parameters mentioned above can be appreciated. The general algorithm for walking pattern generation using the 3D LIP approach is as follows:

Algorithm 3.1 *3D LIP Walking Pattern Generation*

1. *State kinematic parameters of the robot.*
2. *Determine step parameters:* $z_c, T_{sup}, S_x^{(n+1)}, S_y^{(n+1)}$.
3. *Compute the initial conditions (3.6).*
4. *Obtain the hip trajectories by integrating (3.1) along* $[-T_{sup}/2, T_{sup}/2]$ *with initial conditions (3.6).*
5. *Obtain the swing ankle trajectories solving (3.8) and (3.9).*
6. *Compute joint trajectories using the inverse kinematics of legs.*
7. *IF any trajectory is discontinuous: Go to 2.*

3.1.2 Inverse Kinematics for Humanoid Robots Walking: A Geometric Approach

In last section, the problem of developing Cartesian trajectories for the hip and the swing ankle for biped gait was addressed. Such Cartesian trajectories must be transformed into joint trajectories that are going to be tracked by the local joint controllers, by means of the inverse kinematics equations. The problem of determining the most appropriate joint motions for humanoid robots under a given set of motion-task constraints is a difficult task [30]. Humanoid roboticists have developed analytical geometric methods [3, 31], but some others prefer computational approaches such as Extended Jacobian integration [32], multiobjective optimization [27], or prioritized tasks [33]. In this work, the geometric approach presented in [31] was preferred, since the only desired task for the humanoid is to walk.
Therefore, the swing leg desired angles will be firstly computed. Consider Fig. 3.3 c), the first desired joint angle that is computed is the right ankle *roll* by means of the expression:

$$q_1^d = \frac{\pi}{2} + \tan^{-1}\left(\frac{y_h - \bar{y}}{z_c}\right) \tag{3.10}$$

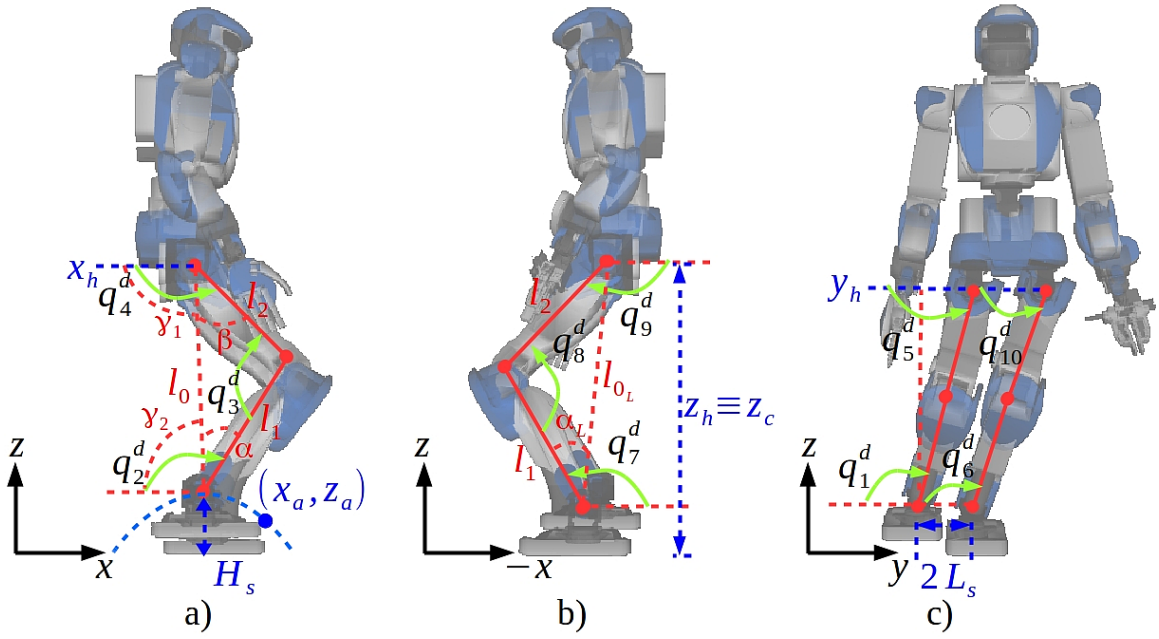

Figure 3.3: Sagittal and frontal planes of a humanoid robot with the useful geometric parameters to derive the inverse kinematics.a) Right-side Sagittal plane. Both step parameters and geometric parameters of the robot can be appreciated. b) Left-side Sagittal plane showing step and geometric parameters. c) Frontal plane, where both step and geometric parameters are depicted.

Where $\bar{y}$ can be computed using (3.3). Continuing with the swing leg, behold Fig. 3.3 a) to compute the ankle *pitch* as follows:

$$q_2^d = -\left(\pi - \alpha - \gamma_2\right) \tag{3.11}$$

Where $\alpha = \cos^{-1}\left(\frac{l_1^2+l_0^2-l_2^2}{2l_0l_1}\right)$, $\gamma_2 = \tan^{-1}\left(\frac{x_h-x_a}{z_c-z_a}\right) + \frac{\pi}{2}$ and $l_0 = \sqrt{\left(x_h - x_a\right)^2 + \left(z_c - z_a\right)^2}$. Notice that x_h, x_a, z_h and z_a are time-depending trajectories, hence the joint angles must be computed for each value of such variables at each sample within the trajectory period. Then, the swing knee angle is given by:

$$q_3^d = \cos^{-1}\left(\frac{l_2^2 + l_1^2 - l_0^2}{2l_1l_2}\right) \tag{3.12}$$

The swing-leg hip *pitch* can be computed using:

$$q_4^d = -\left(\pi - \beta - \gamma_1\right) \tag{3.13}$$

Where $\beta = \pi - \cos^{-1}\left(\frac{l_2^2+l_1^2-l_0^2}{2l_1l_2}\right)$ and $\gamma_1 = \pi - \gamma_2$. Then, observing Fig. 3.3 c), the swing-leg hip *roll* can be obtained by:

$$q_5^d = \pi - q_1^d \tag{3.14}$$

Continuing with the support leg, from the frontal plane can be observed that:

$$q_6^d = q_1^d \tag{3.15}$$

Considering now Fig. 3.3 b), the support-leg knee angle is derived by:

$$q_8^d = \cos^{-1}\left(\frac{l_{0_L}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \tag{3.16}$$

Where $l_{0_L} = \sqrt{x_h^2 + z_h^2}$. In addition, support-leg hip *pitch* can be obtained by means of:

$$q_9^d = -\left(-\alpha_L + \cos^{-1}\left(\frac{l_1^2 - l_{0_L}^2 - l_2^2}{-2l_{0_L} l_2}\right)\right) \tag{3.17}$$

Where $\alpha_L = \tan^{-1}\left(\frac{x_h}{z_h}\right)$. Hence, the support-leg ankle *pitch* can be computed as follows:

$$q_7^d = q_8^d - q_9^d \tag{3.18}$$

Finally, from Fig. 3.3 c) it can be seen that:

$$q_{10}^d = q_5^d \tag{3.19}$$

3.1.3 Humanoids Dynamics: Cart-Table ZMP Approach

This model considers the total mass of the humanoid concentrated in a cart that runs over a massless table, as is depicted in Fig. 3.4 where the constant height of the cart z_c and the generalized position and acceleration of the cart can be appreciated. Even though the foot of the table is significantly small, if the cart moves at a determined acceleration, it is possible to reach instant balance, hence $\ddot{x}$ can be considered as the control input. In addition, its linear displacement x determines the center of pressure that acts from the floor at the sole(s) of the humanoid, consequently, the zero-moment point (ZMP) can be computed in order to have feedback for a suitable balance condition [3].

Definition 3.2 ***Zero Moment Point.***
Is the point where the horizontal components of the moment of the reaction forces between the ground with the foot (feet) of the humanoid is 0. Therefore, the ZMP of the robot can be represented mathematically only for x axis as follows:

$$p_x = x - \frac{z_c}{g}\ddot{x} \tag{3.20}$$

Where p_x is the distance of the ZMP respect to an arbitrary coordinate frame.

In order to guarantee that the robot keeps standing (or balanced), the ZMP must exist inside the support polygon formed by its soles, leading to the following definition:

Definition 3.3 ***Balance of a Humanoid Robot.*** *A humanoid robot is said to be balanced if the its ZMP exists inside the support polygon formed by its sole(s), i.e:*

$$||(p_x, p_y)|| \leq \sup S_p \tag{3.21}$$

Where p_y is the ZMP computed for y axis respect to an arbitrary frame and S_p is a convex hull formed by the sole of a single foot or the soles of the feet.

In the Figure 3.4, the point p_x, is also represented, and it is helpful to determine if the robot is balanced if the ZMP moment $\tau_{ZMP} = 0$ [3]. Now, a state-space representation for the humanoid robot based on (3.20) can be written as:

$$\begin{gathered}\frac{d}{dt}\begin{bmatrix} p_x \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -1/T_{sens} & 1/T_{sens} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_x \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} -z_c/g \\ 0 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_x & x & \dot{x} \end{bmatrix}^T \end{gathered} \tag{3.22}$$

Where T_{sens} is the time constant of the sensors that computes the ZMP located at the soles of the robot. The implementation of a controller using this model requires that the actuators are capable to supply position signals, since the control input is in acceleration units. In addition, it is also necessary a correct measurement of the ZMP using force sensors at the soles of the humanoid to have a reliable feedback. Note that (3.22) is a non-minimum phase plant, so that, some considerations must be taken into account while designing the control system.

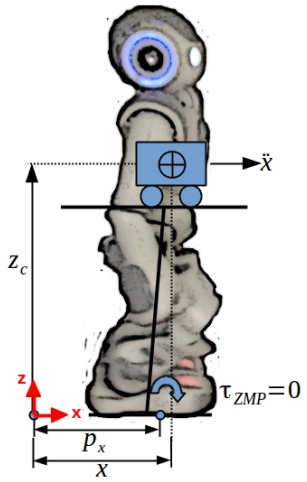

Figure 3.4: Humanoid represented as a cart-table system.

3.1.4 Biped Dynamics for Single-Support Phase using Euler-Lagrange Methodology

Biped walking of humanoid robots is divided generally in two phases: Double-Support Phase (DSP), when both feet of the robot are in contact with the ground, and Single-Support Phase (SSP), when feet are alternatively swinging to perform a step. In order to develop a control strategy that guarantees the SSP stability, the dynamics equations of can be implemented. One of the most used methodology to derive the dynamics equations for robotic systems is the Euler-Lagrange approach, which is based in the difference of kinetic and potential energies. Such methodology has been widely used for manipulators compound by several DOF with successful results. For the case of the humanoid robots dynamics, deriving the equations of motion using this energy-based technique can be more difficult than for manipulators, since humanoids are compound by more DOF in arborescent kinematic chains. Nevertheless, in order to obtain the humanoid's equations of motion using the Euler-Lagrange methodology, some authors perform an approximation of the upper limbs, head and trunk of the humanoid into a single link called *torso*, therefore, the lower limbs are modeled as a planar biped from 3 DOF [34], up to 7 DOF [35], but yielding equations of motion from such approximations is still challenging. To simplify the development of the equations, most authors have followed a convention to measure the angles of the links respect to the y axis of the world frame. In Fig. 3.5, an example of a 5-link planar biped using such convention is presented, where the angles considered respect to the y axis can be appreciated [36]. The resulting equations of motion using the mentioned approach have the following form:

$$D(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + G(\theta) = E\tau \tag{3.23}$$

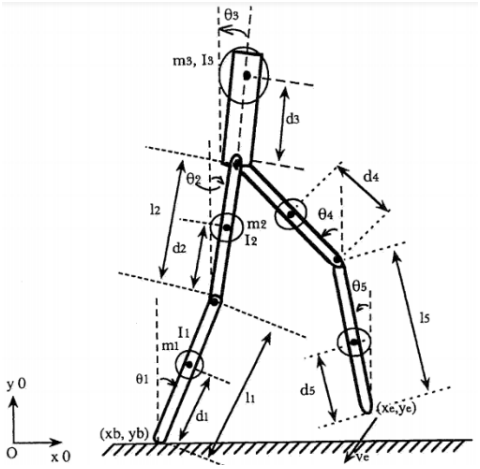

Figure 3.5: 5-link planar biped robot model used in [36].

Where $D(\theta) \in \mathbb{R}^{n\times n}$ is the inertial forces matrix, $h(\theta, \dot{\theta}) \in \mathbb{R}^n$ is the centripetal and Coriolis forces vector, $G(\theta) \in \mathbb{R}^n$ is the gravitational forces vector, τ is the control inputs vector and $E \in \mathbb{R}^{n\times n-1}$ is a mapping matrix such that:

$$E = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix}$$

However, some later transformation are required to map $\theta \rightarrow q$, in order to control the local generalized coordinates. In this thesis work, the direct equations of motion were obtained considering the local joint angles as is depicted in Fig. 3.6, where the joint angle convention and the dynamics parameters can be appreciated. In this sense, and for the derivation of the SSP dynamics, the position vectors of the links are given by:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} l_{c1}s_1 \\ l_{c1}c_1 \end{bmatrix}, \quad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + l_{c2}s_{12} \\ l_1 c_1 + l_{c2}c_{12} \end{bmatrix}, \quad \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + l_2 s_{12} + l_{c3}s_{123} \\ l_1 c_1 + l_2 c_{12} + l_{c3}c_{123} \end{bmatrix},$$

$$\begin{bmatrix} x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + l_2 s_{12} + l_3 s_{123} + l_{c4}s_{1234} \\ l_1 c_1 + l_2 c_2 + l_3 c_{123} + l_{c4}c_{1234} \end{bmatrix}, \quad \begin{bmatrix} x_5 \\ y_5 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + l_2 s_{12} + l_{c5}s_{125} \\ l_1 c_1 + l_2 c_{12} + l_{c5}c_{125} \end{bmatrix} \tag{3.24}$$

In order to derive the dynamics of the biped robot for SSP, only Sagittal-plane motion

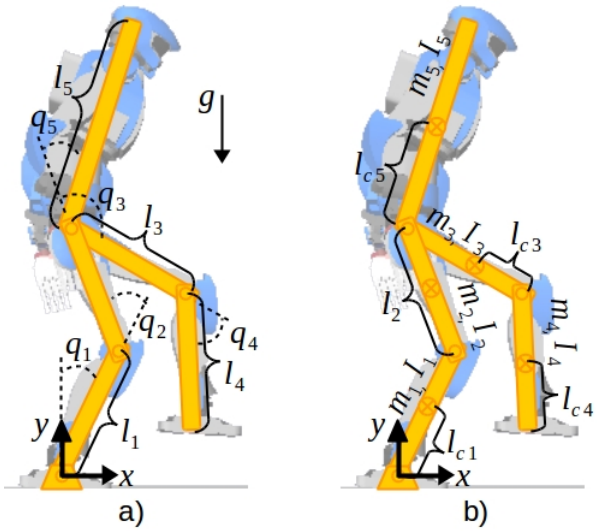

Figure 3.6: 5-link planar biped robot model with dynamics parameters used in this thesis work. a) Displays the joint angle convention and the link lengths. b) Shows the lengths of the links from the joint to their CoM, their masses m_i and their inertia moments I_i; g is the gravity acceleration.

with parallel feet to the ground must be assumed. Therefore, the Lagrangian of the system is given by:

$$\mathcal{L} = \mathcal{K} - \mathcal{U} \tag{3.25}$$

Where the potential energy of the whole system can be expressed as:

$$\mathcal{U} = \sum_{i=1}^{5} m_i g y_i \tag{3.26}$$

And the kinetic energy is:

$$\mathcal{K} = \sum_{i=1}^{5} \frac{1}{2} m_i v_i^2 + \sum_{i=1}^{5} \frac{1}{2} I_i \omega_i^2 \tag{3.27}$$

Where $v_i = \begin{bmatrix} \dot{x}_i & \dot{y}_i \end{bmatrix}^T$ are the linear velocities and ω_i are the angular velocities. After manipulating:

$$\frac{d}{dt}\left\{\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right\} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau \tag{3.28}$$

The equations of motion of the biped robot in SSP can be written as:

$$M(q)\ddot{q} + b(q, \dot{q}) = \tau \tag{3.29}$$

Where $M(q) \in \mathbb{R}^{5\times5}$ is the inertial forces matrix whose entries are:
$M_{11} =$
$m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + m_3 l_1^2 + m_3 l_2^2 + m_3 l_{c3}^2 + m_4 l_1^2 + m_4 l_2^2 + m_4 l_3^2 + m_4 l_{c4}^2 + m_5 l_1^2 + m_5 l_2^2 + m_5 l_{c5}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + 2m_3 l_1 l_2 \cos(q_2 + q_3) + 2m_3 l_2 l_{c3} \cos(q_3) + 2m_4 l_1 l_3 \cos(q_2 + q_3) + 2m_4 l_1 l_{c4} \cos(q_2 + q_3 + q_4) + 4m_4 l_2 l_3 \cos(q_3) + 2m_4 l_2 l_{c4} \cos(q_3 + q_4) + 2m_4 l_3 l_{c4} \cos(q_4) + 2m_5 l_1 l_2 \cos(q_2 + q_5) + 2m_5 l_2 l_{c5} \cos(q_5) + I_1$

$M_{12} =$
$m_2 l_{c2}^2 + m_3 l_2^2 + m_3 l_{c3}^2 + m_4 l_3^2 + m_4 l_{c4}^2 + m_5 l_2^2 + m_5 l_{c5}^2 + m_2 l_1 l_{c2} \cos(q_2) + m_3 l_1 l_2 \cos(q_2 + q_3) + 2m_3 l_2 l_{c3} \cos(q_3) + m_4 l_1 l_2 \cos(q_2) + m_4 l_1 l_3 \cos(q_2 + q_3) + m_4 l_2 l_3 \cos(q_3) + m_4 l_1 l_{c4} \cos(q_2 + q_3 + q_4) + 2m_4 l_2 l_{c4} \cos(q_3 + q_4) + 2m_4 l_3 l_{c4} \cos(q_4) + m_5 l_1 l_2 \cos(q_2 + q_5) + 2m_5 l_2 l_{c5} \cos(q_5)$

$M_{13} =$
$m_3 l_{c3}^2 + m_4 l_3^2 + m_4 l_{c4}^2 + m_3 l_1 l_2 \cos(q_2 + q_3) + 2m_3 l_2 l_{c3} \cos(q_3) + m_4 l_1 l_3 \cos(q_2 + q_3) + m_4 l_1 l_{c4} \cos(q_2 + q_3 + q_4) + m_4 l_2 l_3 \cos(q_3) + 2m_4 l_3 l_{c4} \cos(q_4)$

$M_{14} = m_4 l_{c4} + m_4 l_1 l_{c4} \cos(q_2 + q_3 + q_4) + m_4 l_3 l_{c4} \cos(q_4)$

$M_{15} = m_5 l_{c3}^2 + m_5 l_1 l_2 \cos(q_2 + q_5) + 2m_5 l_2 l_{c5} \cos(q_5)$

$M_{21} = M_{12}$

$M_{22} =$
$m_2 l_{c2}^2 + m_3 l_2^2 + m_3 l_{c3}^2 + m_4 l_3^2 + m_4 l_{c4}^2 + m_5 l_2^2 + m_5 l_{c5}^2 + 2m_3 l_2 l_{c3} \cos(q_3) + 2m_4 l_3 l_{c4} \cos(q_4) +$
$2m_5 l_2 l_{c5} \cos(q_5) + I_2$

$M_{23} = m_3 l_{c3}^2 + m_4 l_3^2 + m_4 l_{c4}^2 + m_4 l_3 l_4 \cos(q_3 + q_4) + m_4 l_2 l_3 \cos(q_3) + 2m_4 l_3 l_{c4} \cos(q_4)$

$M_{24} = m_4 l_{c4}^2 + m_4 l_3 l_{c4} \cos(q_4)$

$M_{25} = m_5 l_{c5}^2 + m_5 l_2 l_{c5} \cos(q_5)$

$M_{31} = M_{13}$

$M_{32} = M_{23}$

$M_{33} = m_3 l_{c3}^2 + m_4 l_3^2 + m_4 l_{c4}^2 + 2m_4 l_3 l_{c4} \cos(q_4) + I_3$

$M_{34} = m_4 l_{c4}^2 + m_4 l_3 l_{c4} \cos(q_4)$

$M_{35} = m_5 l_{c5}^2 + m_5 l_2 l_{c5} \cos(q_5)$

$M_{41} = M_{14}$

$M_{42} = M_{24}$

$M_{43} = M_{34}$

$M_{44} = m_4 l_{c4}^2 + I_4$

$M_{45} = 0$

$M_{51} = M_{15}$

$M_{52} = M_{25}$

$M_{53} = M_{35}$

$M_{54} = M_{45}$

$M_{55} = m_5 l_{c5}^2 + I_5$

$b(q) \in \mathbb{R}^5$ is the bias vector including centripetal, Coriolis and gravitational forces,

which is compound by:

$$
\begin{aligned}
b_1 = \\
&-m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) \left[2\dot{q}_1 + \dot{q}_2\right] \\
&-m_3 l_1 l_2 \left(\dot{q}_2 + \dot{q}_3\right) \sin(q_2 + q_3) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right] \\
&-m_4 l_1 l_2 \dot{q}_2 \sin(q_2) \left[2\dot{q}_1 + \dot{q}_2\right] \\
&-m_4 l_1 l_3 \left(\dot{q}_2 + \dot{q}_3\right) \sin(q_2 + q_3) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right] \\
&-m_4 l_1 l_{c4} \left(\dot{q}_2 + \dot{q}_3 + \dot{q}_4\right) \sin(q_2 + q_3 + q_4) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4\right] \\
&-m_4 l_2 l_3 \dot{q}_3 \sin(q_3) \left[4\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right] \\
&-m_4 l_2 l_{c4} \left(\dot{q}_3 + \dot{q}_4\right) \sin(q_3 + q_4) \left[2\dot{q}_1 + 2\dot{q}_2\right] \\
&-m_4 l_3 l_{c4} \dot{q}_4 \sin(q_4) \left[2\dot{q}_1 + 2\dot{q}_2 + 2\dot{q}_3 + \dot{q}_4\right] \\
&-m_5 l_1 l_2 \left(\dot{q}_2 + \dot{q}_5\right) \sin(q_2 + q_5) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_5\right] \\
&+ \left[m_1 l_{c1} + m_2 l_1 + m_3 l_1 + m_4 l_1 + m_5 l_1\right] g \sin(q_1) \\
&+ \left[m_2 l_{c2} + m_3 l_2 + m_4 l_2 + m_5 l_2\right] g \sin(q_1 + q_2) \\
&+ \left[m_3 l_{c3} + m_4 l_3\right] g \sin(q_1 + q_2 + q_3) \\
&+ m_4 l_{c4} g \sin(q_1 + q_2 + q_3 + q_4) \\
&+ m_5 l_{c5} g \sin(q_1 + q_2 + q_5)
\end{aligned}
$$

$$
\begin{aligned}
b_2 = \\
&-m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \sin(q_2) \\
&-m_3 l_1 l_2 \left(\dot{q}_1 \dot{q}_2 + \dot{q}_1 \dot{q}_3\right) \sin(q_2 + q_3) \\
&-m_3 l_2 l_{c3} \dot{q}_3 \sin(q_3) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right] \\
&-m_4 l_1 l_2 \dot{q}_1 \dot{q}_2 \sin(q_2) \\
&-m_4 l_1 l_3 \dot{q}_1 \left(\dot{q}_2 + \dot{q}_3\right) \sin(q_2 + q_3) \\
&-m_4 l_1 l_{c4} \dot{q}_1 \left(\dot{q}_2 + \dot{q}_3 + \dot{q}_4\right) \sin(q_2 + q_3 + q_4) \\
&-m_4 l_2 l_3 \dot{q}_3 \sin(q_3) \left[4\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right] \\
&-m_4 l_2 l_{c4} \dot{q}_3 \sin(q_3) \left[\dot{q}_1 + 2\dot{q}_2 + \dot{q}_3 +\right] \\
&-m_4 l_3 l_{c4} \dot{q}_4 \sin(q_4) \left[2\dot{q}_1 + 2\dot{q}_2 + 2\dot{q}_3 + \dot{q}_4\right] \\
&-m_5 l_1 l_2 \left(\dot{q}_1 \dot{q}_2 + \dot{q}_1 \dot{q}_5\right) \sin(q_2 + q_5) \\
&-m_5 l_2 l_{c5} \dot{q}_5 \sin(q_5) \left[2\dot{q}_1 + \dot{q}_2 + \dot{q}_5\right] \\
&+ \left[m_2 l_{c2} + m_3 l_2 + m_4 l_2 + m_5 l_2\right] g \sin(q_1 + q_2) \\
&+ \left[m_3 l_{c3} + m_4 l_3\right] g \sin(q_1 + q_2 + q_3) \\
&+ m_4 l_{c4} g \sin(q_1 + q_2 + q_3 + q_4) \\
&+ m_5 l_{c5} g \sin(q_1 + q_2 + q_5)
\end{aligned}
$$

$$
\begin{aligned}
b_3 = \\
&-m_3 l_1 l_2 \dot{q}_1 \left(\dot{q}_2 + \dot{q}_1 \dot{q}_3\right) \sin(q_2 + q_3) \\
&-m_3 l_2 l_{c3} \dot{q}_3 \sin(q_3) \left[\dot{q}_1 + \dot{q}_2\right] \\
&-m_4 l_1 l_3 \dot{q}_1 \left(\dot{q}_2 + \dot{q}_3\right) \sin(q_2 + q_3) \\
&-m_4 l_1 l_{c4} \dot{q}_1 \left(\dot{q}_2 + \dot{q}_3 + \dot{q}_4\right) \sin(q_2 + q_3 + q_4) \\
&-m_4 l_2 l_3 \dot{q}_3 \sin(q_3) \left[\dot{q}_1 + \dot{q}_2\right] \\
&-m_4 l_2 l_{c4} \dot{q}_2 \sin(q_3) \left[\dot{q}_3 + \dot{q}_4 +\right] \\
&-m_4 l_3 l_{c4} \dot{q}_4 \sin(q_4) \left[2\dot{q}_1 + 2\dot{q}_2 + 2\dot{q}_3 + \dot{q}_4\right]
\end{aligned}
$$

$+\left[m_3 l_{c3} + m_4 l_3\right] g \sin(q_1 + q_2 + q_3)$
$+m_4 l_{c4} g \sin(q_1 + q_2 + q_3 + q_4)$
$+m_5 l_{c5} g \sin(q_1 + q_2 + q_5)$

$b_4 =$
$-m_4 l_1 l_{c4} \dot{q}_1 \left(\dot{q}_2 + \dot{q}_3 + \dot{q}_4\right) \sin(q_2 + q_3 + q_4)$
$-m_4 l_3 l_{c4} \dot{q}_4 \sin(q_4) \left[\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right]$
$+m_4 l_{c4} g \sin(q_1 + q_2 + q_3 + q_4)$

$b_5 =$
$-m_5 l_1 l_2 \dot{q}_1 \left(\dot{q}_2 + \dot{q}_1 \dot{q}_5\right) \sin(q_2 + q_5)$
$-m_5 l_2 l_{c5} \dot{q}_5 \sin(q_5) \left[\dot{q}_1 + \dot{q}_2\right]$
$+m_5 l_{c5} g \sin(q_1 + q_2 + q_5)$

The dynamic model (3.29) has the following properties [37]:

- **Property 1**. The inertial forces matrix $M(q)$ is positive definite and its inverse can always be calculated.
- **Property 2**. The nonlinear bias velocity-dependent torque $b(q, \dot{q})$ can be expressed as:
$$b(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) \tag{3.30}$$
Even $C(q, \dot{q}) \in \mathbb{R}^{5\times 5}$ is not unique, $b(q, \dot{q})$ is unique [38].
- **Property 3**. There exists a skew-symmetric matrix $S^{\times}(q, \dot{q})$ that determines the passivity property such that $\dot{q}^T S^{\times}(q, \dot{q})\dot{q} = 0$. This matrix is defined by:
$$S^{\times}(q, \dot{q}) = \frac{1}{2}\dot{M}(q) - C(q, \dot{q}) \tag{3.31}$$
- **Property 4**. Since the humanoiod robot is actuated only by rotational joints, the gravitational torques vector $g(q)$ is Lipschitz, i.e., it exists a constant $k_g > 0$ such that [38]:
$$||g(x) - g(y)|| \leq k_g ||x - y||, \qquad \forall x, y \in \mathbb{R}^5 \tag{3.32}$$
A possible bound of k_g may be:
$$k_g \geq n \left[\max_{i,j,q} \left|\frac{\partial g_i(q)}{\partial q_j}\right|\right] \tag{3.33}$$
Furthermore, k_g satisfy:
$$k_g \geq \|\frac{\partial g(q)}{\partial q}\| \geq \lambda_{Max}\left\{\frac{\partial g(q)}{\partial q}\right\} \tag{3.34}$$
Where λ_{Max} is the maximum eigenvalue.

Such properties are useful for control systems design.

3.1.5 Humanoids Dynamics using Spatial Vector Algebra

A rigid body has six degrees of motion freedom, therefore, in literature there exists a tool that can represent mathematically such DOF by means of 6D vectors: the *spatial vector algebra* [39]. Such technique has been widely used for robot kinematics and dynamics applications with successful results [40], since it reduces the volume of algebra yielding efficient, clear and compact solutions in comparison to use 3D-vectors equations.

Definition 3.4 ***Spatial Vector.*** *A spatial vector is a 6D vector that provides a complete description of the state of motion of a rigid body, or the forces acting upon it, in much the same way that a Euclidean vector provides a complete description of the state of motion of a particle, or the forces acting upon it. In particular, a spatial vector combines the linear and angular aspects of rigid-body motions* $\hat{\boldsymbol{m}} \in M^6$ *or forces* $\hat{\boldsymbol{f}} \in F^6$ *into a single quantity.*

A body-fixed point is a point in a fixed location relative to a rigid body. When the body moves, the point moves with it. The set of vector fields that describe every possible velocity of a rigid body moving in 3D space forms a 6D vector space [39]. The spatial velocity of a rigid body B moving from an origin O to a point P can be expressed as:

$$\hat{\mathbf{v}} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \nu_{O_x} & \nu_{O_y} & \nu_{O_z} \end{bmatrix}^T \tag{3.35}$$

Where $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ is the angular velocity and $\nu_O = \begin{bmatrix} \nu_{O_x} & \nu_{O_y} & \nu_{O_z} \end{bmatrix}^T$ is the linear velocity. Notice that equation (3.35) defines a vector field, the velocity field of a body B [41]. Likewise, a force acting on a rigid body at a point O can be represented by:

$$\hat{\mathbf{f}} = \begin{bmatrix} n_{O_x} & n_{O_y} & n_{O_z} & f_x & f_y & f_z \end{bmatrix}^T \tag{3.36}$$

Where $n = \begin{bmatrix} n_{O_x} & n_{O_y} & n_{O_z} \end{bmatrix}^T$ is the moments vector at the point o and $f = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$ is the forces vector.

Plücker Coordinate Transforms

Consider the motion from coordinate frames A to B with displacement r and rotation E represented graphically in Fig. 3.7. Such motion can be computed in terms of the Plücker transform using the following equation:

$${}^B\mathbf{X}_A = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \tilde{\mathbf{r}}^T & \mathbf{1} \end{bmatrix} \tag{3.37}$$

Where:

$$\tilde{\mathbf{r}} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

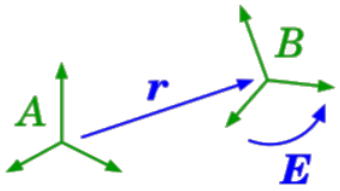

Figure 3.7: Representation of a motion from coordinate frames A to B with displacement r and rotation E.

For the case of force vectors representation, the following expression is useful:

$$ {}^{B}\mathbf{X}_{A}^{*} = ({}^{B}\mathbf{X}_{A})^{-T} \tag{3.38} $$

Spatial Acceleration

If a rigid body has a velocity $\begin{bmatrix}\omega^T & \nu_O^T\end{bmatrix}^T$ respect to an origin O, its spatial acceleration is given by [41]:

$$ \hat{\mathbf{a}} = \dot{\hat{\mathbf{v}}} = \frac{d}{dt}\begin{bmatrix}\omega \\ \nu_O\end{bmatrix} = \lim_{\delta t \to 0} \frac{1}{\delta t}\begin{bmatrix}\omega(t+\delta t) - \omega(t) \\ \nu_O(t+\delta t) - \nu_O(t)\end{bmatrix} = \begin{bmatrix}\dot{\omega} \\ \ddot{r} - \omega \times \dot{r}\end{bmatrix} \tag{3.39} $$

Spatial Inertia

The spatial momentum of a rigid body is a spatial force vector (wrench) describing its linear and angular momentum.

$$ \overrightarrow{h}_O = \overrightarrow{h}_c + \overrightarrow{c} \times \overrightarrow{h} \tag{3.40} $$

The spatial intertia of a rigid body is a function of its mass, center of mass (CoM) and rotational inertia about its CoM, that maps spatial velocity to momentum.

$$ I_O = \begin{bmatrix}\bar{I}_O & m\overrightarrow{c}\times \\ m\overrightarrow{c}\times^T & m\end{bmatrix} \tag{3.41} $$

Where $\bar{I}_O = \bar{I}_C - m\overrightarrow{c} \times \overrightarrow{c}\times$, m is the mass, C is the CoM and $\bar{I}_C$ is the inertia at CoM [39].

Humanoid Dynamics with Contacts

The equation of motion of a humanoid robot can be expressed as follows [42]:

$$ M\ddot{q} + b = u + \sum_{i=1}^{N_E} J_{E_i}^T f_{E_i} \tag{3.42} $$

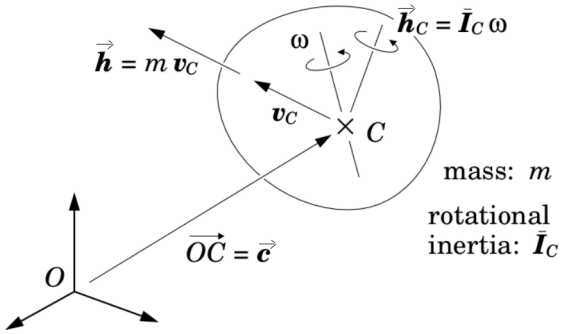

Figure 3.8: Moving rigid body whose centre of mass coincides with the fixed point C. Its mass is m, and its rotational inertia is $\bar{I}_c$ [41].

where $q \in \mathbb{R}^n$ is the generalized coordinate of the robot, $M = M(q) \in \mathbb{R}^{n\times n}$ is the inertial forces matrix, $b = b(q, \dot{q}) \in \mathbb{R}^n$ is the nonlinear bias force including the generalizaed centrifugal, Coriolis and gravitational forces, $J_{E_i} = J_{E_i}(q) \in \mathbb{R}^{n\times 3}$ is the Jacobian matrix of the ith contact point and $f_{E_i} \in \mathbb{R}^3$ is the ith contact force; furthermore, n is the size of q and N_E is the number of contact points [42]. Fig. 3.9 shows an example of a humanoid robot in contact with the environment, where all the contact point forces are illustrated.

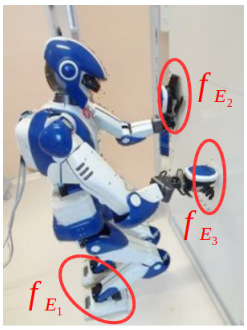

Figure 3.9: Humanoid robot in contact with the environment using are 3 contact points, consequently, there are 3 contact forces affecting the robot dynamics.

Humanoid Dynamics in Spatial Coordinates

Humanoid robots interact with the environment using their end links; such end-links motion and force controllers are specified in spatial coordinates [37]. Consider the eq. (3.9), the relation between the inertial forces of the robot and the end-link spatial coordinates is given by:

$$\mathbb{M}_e^{-1}(q) = J(q)M_q^{-1}(q)J^T(q) \tag{3.43}$$

Where $M_q(q) = M(q)$ and $\mathbb{M}_e^{-1}(q)$ is the operational space inertia matrix [37]. Therefore, considering the static force relation $\tau_f = J^T F^m$, F^m is the spatial force related to the mobility rather than contacts, the equation of motion of humanoids in spatial coordinates for operational space can be expressed as:

$$F^m = \mathbb{M}_e(q)\dot{\nu} + C_e(q, \dot{q}) + \mathcal{G}_e(q) \tag{3.44}$$

Where $C_e(q, \dot{q}) = J^T b_q - \mathbb{M}_e(q)\dot{J}\dot{q}$ is the nonlinear velocity-dependent force and $\mathcal{G}_e(q) = J^T G_q$ is the gravity force, both are expressed as end-link wrenches [37].

Multirobot Formalism

Consider a system of n robots, which can be actual robots or passive articulated mechanisms modeled as a kinematic tree structure using [43]:

$$M_i(q_i)\ddot{q}_i + b_i(q_i, \dot{q}_i) = J_i^T f_{E_i} + S_i\tau_i \tag{3.45}$$

Where S_i is an actuation mapping matrix and f_{E_i} are contact forces, as in (3.42), but also applied to the ith robot, and can be decomposed as follows [43]:

$$f_i = (f_i^0, f_i^-, -f_i^+) \tag{3.46}$$

Where f_i^0 stacks the forces applied by the fixed environment on the robot, f_i^- are the forces applied by the robots $j < i$ on robot i, and f_i^+ stacks the forces applied by the robot i on robots $j > i$, hence, the equations of motion for a whole system of robots can be written as [43]:

$$M_i(q_i)\ddot{q}_i + b_i(q_i, \dot{q}_i) = J_{i,0}^T f_i^0 + J_{i,-}^T f_i^- - J_{i,+}^T \Psi_i F^- + S_i\tau_i \tag{3.47}$$

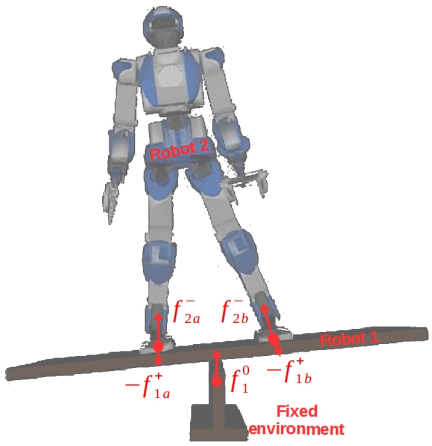

Figure 3.10: Humanoid robot in contact with a seesaw fixed to the environment and the forces acting to each robot. Humanoid is the Robot 2, the seesaw is the Robot 1, and the base of the seesaw is the fixed environment, since it is attached to the floor.

Where $J_{i,0}$, $J_{i,-}$ and $J_{i,+}$ are matrices obtained by extracting the columns of f^0, f^- and f^+ from f, F^- is the vector containing all f_i^-, and $\Psi = \Pi \otimes I_3$ is a permutation matrix, $\otimes$ is the Kronecker product. Fig. 3.10 shows an example of the force interaction between robots.

3.2 Modelling of Quadrotors

Consider the Fig. 3.11, where a UAV airframe, its generalized coordinates and the forces acting on it are depicted. The coordinate frame of the rigid UAV body (or navigation frame) is denoted by $\{B\}$, and the world reference frame is $\{W\}$. Note that the x axis of the navigation frame is in the frontal motion direction, y axis is in the lateral motion direction, and z is the vertical motion axis with positive direction pointing to the ground according to aerodynamics convention. Furthermore, the *roll*, *pitch* and *yaw* angular motions represented by ϕ, θ and ψ respectively can be appreciated.

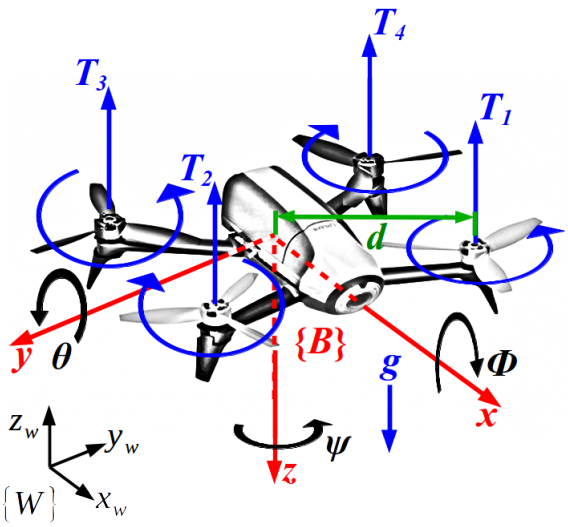

Figure 3.11: Quadrotor navigation and coordinates frames.

Rotor Dynamics

Each rotor that compounds the UAV has an angular velocity ω_i which produces a vertical force or thrust given by [44]:

$$T_i = C_{T_i} \rho A_{r_i} r_i^2 \omega_i^2 \tag{3.48}$$

Where, for the i-th rotor, C_{T_i} is the thrust coefficient that depends on the chord length and geometric profile of the blades, ρ is the air density, A_{r_i} is the rotor disk area and r_i is the radius. However, for simplicity, some authors lump such parameters into a single parameter model as follows:

$$T_i = k_T \omega_i^2 \tag{3.49}$$

Where $k_T > 0$ can be determined by experimental thrust tests [44]. In addition, the torque applied to each propeller by the actuator is opposed by aerodynamic drag such that [45]:

$$Q_i = k_Q \omega_i^2 \tag{3.50}$$

Note that k_Q depends on the same parameters as k_T.

Airframe Dynamics

Consider the i-th rotor thrust (3.49), the total upward or hover thrust applied to the airframe is given by [44]:

$$T = \sum_{i=1}^{4} |k_T \omega_i^2| \tag{3.51}$$

By Newton's Second Law, the equations of translational motion of the quadrotor are given by[45]:

$$m\ddot{\xi} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} - {}^{W}\mathbf{R}_B \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \tag{3.52}$$

Where $\xi = [x, y, z]^T$ is the linear displacement coordinates vector, m is the total mass of the quadrotor, g is the gravity acceleration and ${}^{W}\mathbf{R}_B$ is the rotation matrix from the world coordinate frame $\{W\}$ to the navigation frame $\{B\}$, which can be expressed in terms of the *roll, pitch* and *yaw* angles as follows [46]:

$${}^{W}\mathbf{R}_B = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi s_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \tag{3.53}$$

In addition to translational forces, also rotational moments affect the airframe; pairwise differences in rotor thrusts cause the vehicle to rotate [45]. The torque about x axis (or *roll* torque) is given by:

$$\tau_\phi = d(T_4 - T_2) = dk_T(\omega_4^2 - \omega_2^2) \tag{3.54}$$

Where d is the distance between the center of mass of the vehicle and the center of any rotor, which can also be appreciated in Fig. 3.11. The torque about y axis (or *pitch* torque) can be obtained by:

$$\tau_\theta = d(T_1 - T_3) = dk_T(\omega_1^2 - \omega_3^2) \tag{3.55}$$

Now consider the aerodynamic drag (3.50) caused by the i-th rotor, the total reaction torque about z axis is expressed as:

$$\tau_\psi = k_T(Q_1 - Q_2 + Q_3 - Q_4) = k_T(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{3.56}$$

Note that the rotational motion of the airframe is expressed by the following 2^{nd} order dynamics [45]:

$$\tau = J\ddot{\eta} + \beta\dot{\eta} \tag{3.57}$$

Where $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ is the moments vector, $\eta = [\phi, \theta, \psi]^T$ is the rotational motion generalized coordinates vector, $\beta \approx 0$ is the aerodynamic damping coefficient [45], and the inertia tensor is:

$$J = \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{pmatrix} \tag{3.58}$$

According to [45], the products of inertia (elements outside the main diagonal of J) are zero if the object's mass distribution is symmetrical with respect to the coordinate frame, which is the case of the UAV of this work, hence, only the main diagonal elements are considered as $J_1 = J_{xx}$, $J_2 = J_{yy}$ and $J_3 = J_{zz}$. Considering ϕ and θ very small, such that $\cos(\phi) \approx 1, \sin(\phi) \approx \phi$, and the same for θ, the dynamics of the quadrotor airframe is given by:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ J_3\ddot{\psi} \\ J_2\ddot{\theta} \\ J_1\ddot{\phi} \end{bmatrix} = \begin{bmatrix} T(c_\psi s_\theta c_\phi + s_\psi s_\phi) \\ T(s_\psi s_\theta c_\phi - c_\psi s_\phi) \\ Tc_\theta c_\phi - mg \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} T(\theta + \psi\phi) \\ T(\psi\theta - \phi) \\ T - mg \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \tag{3.59}$$

It worth to remark that the quadrotor used in this work, *Parrot Bebop*, includes internal controllers $T, \tau_\phi, \tau_\theta$ that compensates the nonlinearities of the model and the gravity, yielding the following model:

$$\begin{aligned} m\ddot{x} &= \bar{u}_x \\ m\ddot{y} &= \bar{u}_y \\ m\ddot{z} &= \bar{u}_z \\ J_3\ddot{\psi} &= \bar{u}_\psi \end{aligned} \tag{3.60}$$

where $\bar{u}_{(\cdot)}$ are the control inputs.

4

Robust Control of Humanoid Robots and MAVs

4.1 Active Disturbance Rejection Control for Humanoid Robust Balance

The balance control problem is based on keeping the ZMP inside the support polygon formed by the soles of the humanoid that are in contact with the ground, so that, the robot remains balanced even in presence of disturbances.

Control System Design

Considering motion only in x direction and the sensors time constant $T_{sens} = 1$ for simplicity, the cart-table model (3.22) requires to be transformed into a canonical *cascade of integrators* form as is suggested from ADRC theory. Therefore, consider the controllability matrix of (3.22) given by:

$$M_C = \begin{bmatrix} -z_c/g & z_c/g & 1 - z_c/g \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.1}$$

Note that (4.1) is full rank, so that, it is invertible, and, consequently, there exists:

$$M_C^{-1} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \tag{4.2}$$

From row q_3, the inverse transformation matrix is built as follows:

$$T_C^{-1} = \begin{bmatrix} q_3 & q_3 A & q_3 A^2 \end{bmatrix}^T \tag{4.3}$$

Hence, inverting (4.3) yields the corresponding transformation matrix:

$$T_C = \begin{bmatrix} 1 & 0 & -z_c/g \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{4.4}$$

Therefore, the corresponding transformed system is:

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (u_C + w_x) \qquad y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} z \tag{4.5}$$

Where $z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T = T_C^{-1} \begin{bmatrix} p_x & x & \dot{x} \end{bmatrix}^T$, w_x is the disturbance affecting the ZMP dynamics and u_C is a controller $u_C = \tilde{K}z$ that works on the new coordinates, then, a return of coordinates must be carried out for the original state variables by means of:

$$u_0 = \tilde{K} T_C^{-1} \begin{bmatrix} p_x & x & \dot{x} \end{bmatrix}^T \tag{4.6}$$

Note that $\tilde{K}$ can be designed using a classical technique such as Ackermann's formula or LQR optimal approach. Once the system is represented in a canonical form, an extended state variables set $\xi = \begin{bmatrix} z_1 & z_2 & z_3 & w_x \end{bmatrix}^T$ is put into effect according to the ADRC theory, hence, the extended-state representation of (4.5) is given by:

$$\dot{\xi} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_C \qquad y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \xi \tag{4.7}$$

Then, the following SMO is proposed:

$$\dot{\hat{\xi}} = \begin{pmatrix} \hat{\xi}_2 - (l_1 \hat{e}_1 + sign(\hat{e}_1)) \\ \hat{\xi}_3 - (l_2 \hat{e}_1 + sign(\hat{e}_1)) \\ \hat{\xi}_4 - \hat{\xi}_3 - (l_3 \hat{e}_1 + sign(\hat{e}_1)) + u_C \\ -(l_4 \hat{e}_1 + sign(\hat{e}_1)) \end{pmatrix} \tag{4.8}$$

Where $e_i = \xi_i - \hat{\xi}_i$ is the observation error for the $i-$th variable, $\hat{\xi}_4 = \xi_1 - \hat{\xi}_1$ is the disturbance reconstruction and l_i are the observer gains. Finally, the total controller for the ZMP is given by:

$$u = \tilde{K} T_C^{-1} \begin{bmatrix} p_x & x & \dot{x} \end{bmatrix}^T - T_C^{-1} \hat{\xi}_4 \tag{4.9}$$

However, some tradeoffs have to be taken into account for the non-minimum phase plant (3.22); this kind of systems with a faster response will have more undershoot and vice versa, in addition, for a non-minimum phase system with slow poles it is hard to get a fast response, because the observer bandwidth has an upper limit, hence, increasing system bandwidth might lead the system close to instability [47]. Some techniques have been developed to cancel the effect of non-minimum phase for humanoid robots like using preview control [3], or using the torso dynamics as a phase

compensator [48]. In this thesis work, it was preferred to have an slower response to minimize the undershoot by means of a LQR gain $\tilde{K}$ selection approach presented below. Additionally, it worth to mention that another control scheme must be considered for y motion, which is independent from x motion controller.

Control System in Discrete Time

Some commercial humanoid robots are equipped with force sensors in the soles, like NAO robot of *Aldebaran Robotics*, which is helpful to compute the ZMP in order to robust the walking task via a stabilizer. However, accessing the data of several sensors and running some other algorithms such as computing the robot pose or performing a control scheme can take significant time, which might worth to consider relatively high sample periods, and consequently, to carry out a discrete-time analysis. Therefore, consider the model (3.22) to design a ZMP based control to balance a humanoid robot and a canonical form of such model given by (4.5); hence, a discrete-time representation of the transformed system using the zero-order hold method can be constructed as follows [49]:

$$\begin{aligned} z(k+1) &= \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} z(k) + \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T \end{bmatrix} (u_C(k) + w_x(k)) \\ y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} z(k) \end{aligned} \tag{4.10}$$

Where T_s is the sampling period. Note that (4.10) keeps the *cascade of unit delays* canonical form. Once the canonical form requirement is fulfilled as is requested from the ADRC theory, the following extended-state representation that considers $\xi(k) = \begin{bmatrix} z_1(k) & z_2(k) & z_3(k) & w_x(k) \end{bmatrix}^T$ is given:

$$\begin{aligned} \xi(k+1) &= \begin{bmatrix} 1 & T_s & T_s^2/2 & T_s^3/6 \\ 0 & 1 & T_s & T_s^2/2 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \xi(k) + \begin{bmatrix} T_s^4/8 \\ T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix} u_C(k) \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \xi(k) \end{aligned} \tag{4.11}$$

Now, in order to continue with the design of the ADRC, the following *Quasi Sliding Mode Observer* is proposed for the disturbances detection based on the measurement

of ZMP:

$$\dot{\hat{\xi}}(k) = \begin{pmatrix} \hat{\xi}_1(k) + T_s\hat{\xi}_2(k) + \frac{T_s^2}{2}\hat{\xi}_3(k) + \frac{T_s^3}{6}\hat{\xi}_4(k) + \frac{T_s^4}{8}u_C(k) - l_1 sign(\hat{e}_1(k)) \\ \hat{\xi}_2(k) + T_s\hat{\xi}_3(k) + \frac{T_s^2}{2}\hat{\xi}_4(k) + \frac{T_s^3}{6}u_C(k) - l_2 sign(\hat{e}_1(k)) \\ \hat{\xi}_3(k) + T_s\hat{\xi}_4(k)\frac{T_s^2}{2}u_C(k) - l_3 sign(\hat{e}_1(k)) \\ \xi_1(k) - \hat{\xi}_1(k) + T_s u_C(k) - l_4 sign(\hat{e}_1(k)) \end{pmatrix} \tag{4.12}$$

Where $\hat{e}_i(k) = \xi_i(k) - \hat{\xi}_i(k)$ is the $i-$th observation error. Note that the observation error $\hat{e}_1(k)$ is not required in the observer, since the disturbance is propagated along all states from the input transfer vector as control inputs does. The observer gains l_i are chosen based on the bandwidth of the observer and on the sample period T_s as is suggested in (2.31). The auxiliary control input $u_C(k) = \tilde{K}z(k)$ is designed as the following discrete LQR porblem. The target of control system is to minimize the following ZMP error cost function:

$$J = \sum_{j=1}^{\infty} \left\{ Q(p_{x_j}^d(k) - p_{x_j}(k))^2 + Ru_{C_j}^2(k) \right\} \tag{4.13}$$

Where $Q > 0 \in \mathbb{R}^{3\times 3}$ and $R > 0$ are weight constants. Reducing J means solving the following Riccati equation:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \tag{4.14}$$

In order to guarantee stability in the sense of Lyapunov. Solving (4.14) yields the gains using:

$$\tilde{K} = R^{-1}B^T S \tag{4.15}$$

Finally, the discrete-time ADRC based in *Quasi Sliding Mode Observer* is given by:

$$u(k) = \tilde{K}T_{C_D}^{-1}\begin{bmatrix} p_x(k) & x(k) & \dot{x}(k) \end{bmatrix}^T - T_{C_D}^{-1}\hat{\xi}_4(k) \tag{4.16}$$

Where T_{C_D} is the corresponding transformation to return to the original coordinates but in discrete time. Note that the same tradeoffs for minimum-phase systems like control (4.9) have to be considered when designing this controller, in addition to a low-pass prefiltering stage for ZMP measurements.

Implementation

The designed control system (4.16) was implemented using a Nao humanoid robot. Consider the equation (3.20) that represents the ZMP of a humanoid. Even x is the variable of interest, it is frequently not available as a direct measurement; nevertheless, $\ddot{x}$ can be obtained from an inertial measurement unit (IMU) attached at the humanoid

torso as is displayed in Fig. 4.1, and x can be integrated from such measurement. Additionally, the ZMP can be computed as follows [50]:

$$zmp = \frac{1}{\sum_1^4 F_{Lfsr_i}} \begin{bmatrix} \sum_1^4 F_{Lfsr_i} dL_{x_i} \\ \sum_1^4 F_{Lfsr_i} dL_{y_i} \end{bmatrix} + \frac{1}{\sum_1^4 F_{Rfsr_i}} \begin{bmatrix} \sum_1^4 F_{Rfsr_i} dR_{x_i} \\ \sum_1^4 F_{Rfsr_i} dR_{y_i} \end{bmatrix} \tag{4.17}$$

Where F_{Lfsr_i} is the force measured at the $i-th$ force sensor (FSR) of the left foot, F_{Rfsr_i} represents the same force at the right foot, dL_{y_i} is the force location at y axis of the $i-th$ FSR at the left foot respect to an arbitrary frame, dL_{x_i} is the same location but at x axis, dR_{y_i} and dR_{x_i} are the right foot counterparts. It worth to mention that (3.20) represents only the x axis motion. In Fig. 4.2, the concept of location of FSR to compute the ZMP is illustrated; furthermore, the arbitrary reference frame from where the distances are computed can be appreciated. The NAO humanoid parameters are shown in Table 4.1. The DLQR parameters obtained from (4.14), which consider the lowest energy consumption are:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

$$R = 1e-3 \tag{4.19}$$

$$P = \begin{bmatrix} 480.026 & 234.299 & 22.775 \\ 234.299 & 1244.482 & 46.328 \\ 22.775 & 46.328 & 34.337 \end{bmatrix} \tag{4.20}$$

$$K = \begin{bmatrix} 6.2917(g/zc)T_s & 37.7014T_s & 32.5359T_s \end{bmatrix} \tag{4.21}$$

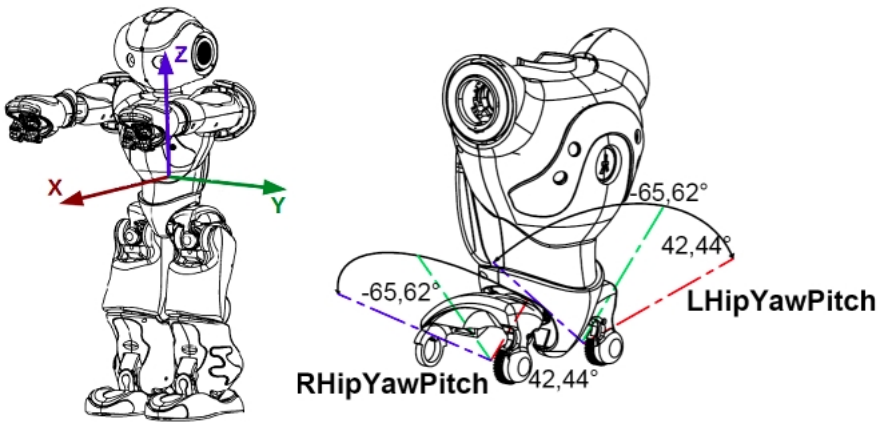

Figure 4.1: Diagram representing the IMU attached at Nao humanoid torso and hip joints used as control inputs.

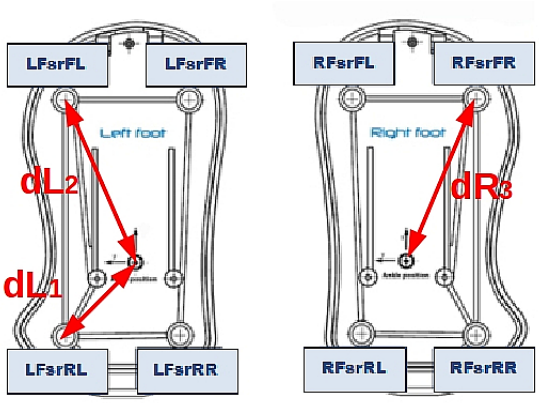

Figure 4.2: FSRs attached to humanoid feet that help the ZMP estimation.

Table 4.1: Parameters of the NAO humanoid robot.

Parameter	**Symbol**	**Value**	**Units**
Mass of the robot	m	5.182530	Kg
CoM Height	z_c	0.335	m
Gravity acceleration	g	9.81	m/s^2
Sample period	T	100	ms

The stable poles of LQR are given by:

$$z = \begin{bmatrix} 0.96887 \\ 0.99889 + 0.00044i \\ 0.99889 - 0.00044i \end{bmatrix} \tag{4.22}$$

The whole control system (4.16) was programmed in *Python* language using the hip actuators showed in Fig. 4.1.

Results

In order to test the proposed ADRC on the Nao platform, a target ZMP was set in $p_x^{ref} = 0.02m$ respect to the IMU frame. The experiment consisted on leading the ZMP of the robot to the target using the hip actuators. Once the ZMP position was regulated, external forces at the torso are applied in frontal direction as is displayed in Fig. 4.3 a), and also shoves from the back were applied as can be appreciated in Fig. 4.3 b), in order to disturb the ZMP position and hence the humanoid should keep balanced. This experiment was excecuted for 20 seconds using both the DLQR and the ADRC controllers.
Fig. 4.4 a) shows the results of the developed experiment using the DLQR control, where it can be observed that the oscilations of the ZMP after shoves are of bigger

amplitude and higher frequency than those depicted in Fig. 4.5 a), which are the results of the implementation of ADRC. In addition, in Figs. 4.4 b) and 4.5 b), the disturbances detected by the Quasi-sliding mode extended-state observer are presented; it worths to mention that the DLQR do not use such detected disturbance. Figs. 4.4 c) and 4.5 c) show the x position computed from the double integration of the IMU signals and also the observed $\hat{x}$, where a successful tracking from the robust observer can be appreciated.

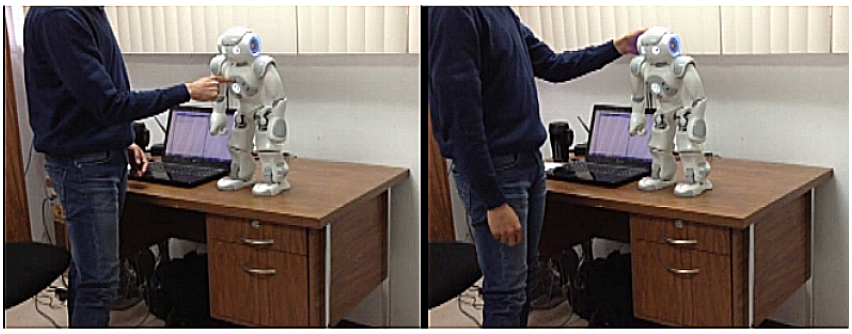

Figure 4.3: Disturbances applied to the humanoid. a) Front disturbance. b) Back disturbance. Video available at `https://www.youtube.com/watch?v=B9_J0H3KX0I`.

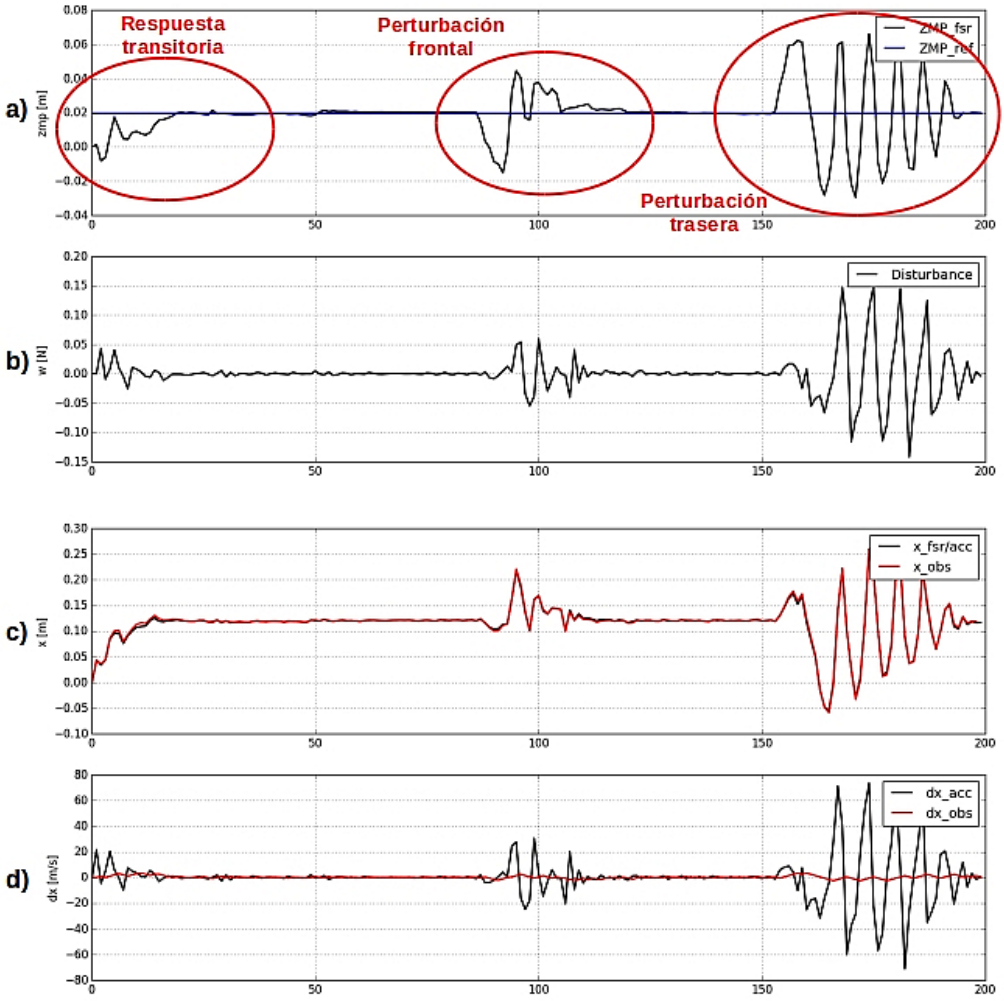

Figure 4.4: Results with DLQR. a) Controlled ZMP. b) Detected disturbance $w_x(k)$. c) Computed and observed positions. d) Computed and observed velocities.

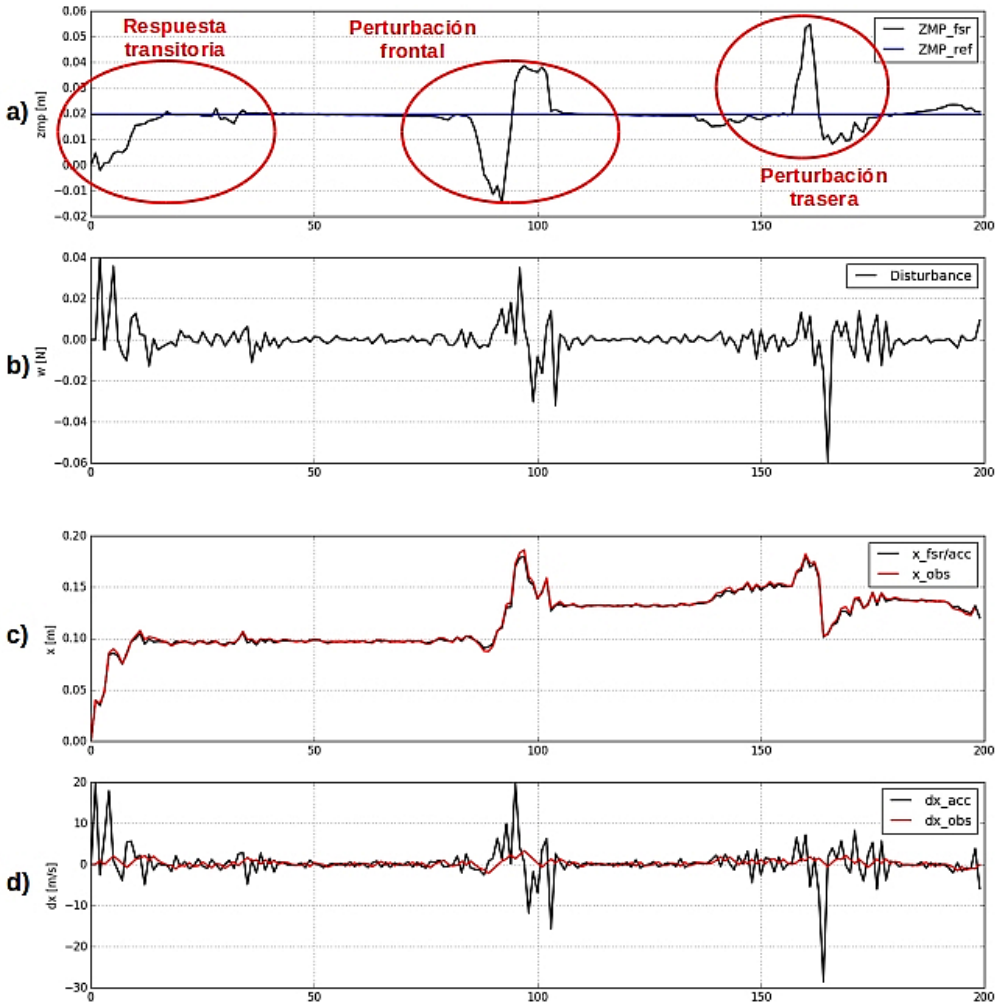

Figure 4.5: Results with ADRC. a) Controlled ZMP. b) Detected disturbance $w_x(k)$. c) Computed and observed positions. d) Computed and observed velocities.

4.2 Active Disturbance Rejection Control for Humanoid Robust Walking

The walking problem of a biped robot is to follow set of joint trajectories within a period of time in order to perform a step (or a set of steps), keeping the robot itself balanced in both SSP and DSP, even in presence of disturbances. Then, consider the humanoid dynamics in SSP (3.29) represented as the following system subjected to disturbances:

$$\dot{x} = f(x) + g(x)u + w \tag{4.23}$$

Where w is a set of unknown piecewise-smooth bounded external disturbances evolving in Ω, which is a compact subset of $\mathbb{R}^p$. In order to track the desired joint trajectories, a robust controller must be applied.

Definition 4.1 ***Robust state feedback stabilizing control*** *[51].*
A static state feedback $u = k(x)$, with k a smooth function mapping $\mathbb{R}^n \to \mathbb{R}$ and $k(0) = 0$ is said to be a local robust state feedback stabilizing control for system (4.23) if the origin $x(0)$ of the closed-loop system

$$\dot{x} = f(x) + g(x)k(x) + w$$

is locally uniformly asymptotically stable for any $w \in \Omega$. It is said to be a globally stabilizing robust control if the origin is globally uniformly asymptotically stable for any $w \in \Omega$.

Theorem 4.1 *Assume that the nominal system (f, g) of (4.23) is globally state feedback linearizable and Ω is a compact subset of $\mathbb{R}^p$, therefore, there exists a global static state feedback locally stabilizing control [51].*

Theorem 4.2 *Assume that the system (4.23) is globally state feedback linearizable, and that there exists an Extended-State Observer (ESO) that can estimate a total disturbance $\hat{f} + w$. Therefore, there exists a robust state feedback stabilizing control that use the information cast from the ESO to exaclty linearize (4.23)[14].*

Generation of the Desired Joint Trajectories

Firstly, a walking pattern generation technique must be implemented to later develop the trajectories of the joints using the inverse kinematics of the legs, hence, the 3D LIP approach is used considering the step parameters shown in Table 4.2 for equations (3.2), (3.3), (3.4) and (3.5). The CoM is named, in this work, indistinctly as hip, only for the case of 3D LIP motion. The resulting walking primitive for x motion can be appreciated in Fig. 4.6, where both desired position and velocity trajectories of the hip in the Sagittal plane can be appreciated. Furthermore, Fig. 4.7 shows the position and velocity desired trajectories for the hip in y direction. Additionally, considering

the equations (3.8) and (3.9), the desired position trajectories of the ankle in x and z directions are depicted in Fig. 4.8. The Cartesian-space trajectories of the hip and the ankle are displayed in Fig. 4.9.

Table 4.2: Step parameters for 3D LIP walking pattern generation.

Parameter	**Symbol**	**Value**	**Units**
Height of the CoM	z_c	0.65	m
Half step width	H_s	0.1	m
Half step length	D_s	0.15	m
Support phase time	T_{sup}	0.5	s

Now consider the inverse kinematics equations from (3.10) to (3.19) and the Fig. 3.3. Solving such equations using the Cartesian trajectories yield the desired joint trajectories for the humanoid legs. The parameters of the robot used for the inverse kinematics solution are listed in Table 4.3. Fig. 4.10 shows the resulting desired joint trajectories for the support-leg joints in Sagittal plane. The desired joint trajectories for the swing-leg are depicted in Fig. 4.11, and finally, the desired trajectories of the ankles and hips for the frontal plane joints are presented in Fig. 4.12. Those desired joint trajectories feed the robust control input that must ensure the asymptotic tracking of such joint trajectories, that consequently yield the accomplishment of the walking task.

Table 4.3: Humanoid robot kinematic parameters for inverse kinematics solution.

Parameter	**Symbol**	**Value**	**Units**
support shank	l_1	0.34	m
support thigh	l_2	0.36	m
swing thigh	l_3	0.36	m
swing shank	l_4	0.34	m

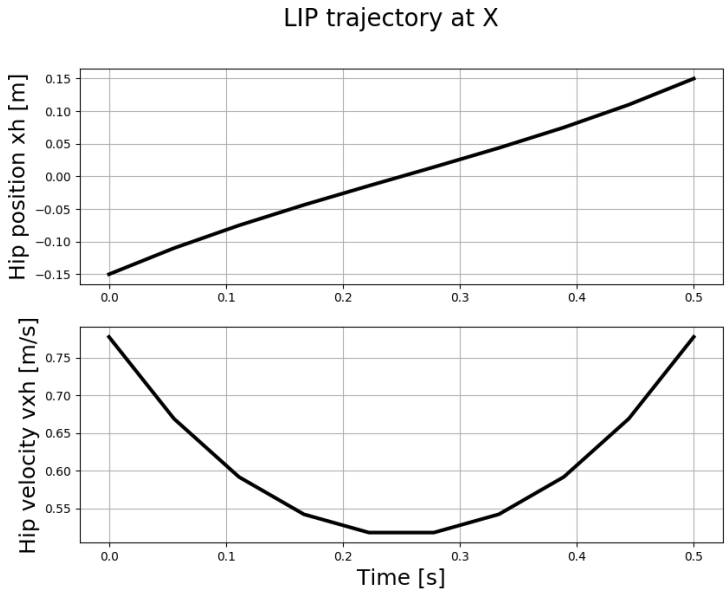

Figure 4.6: Desired position and velocity trajectories of the hip (CoM) in x direction.

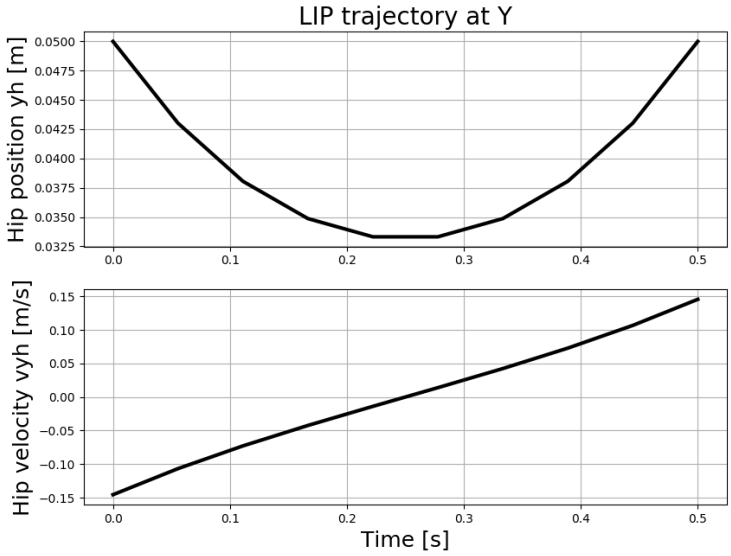

Figure 4.7: Desired position and velocity trajectories of the hip (CoM) in y direction.

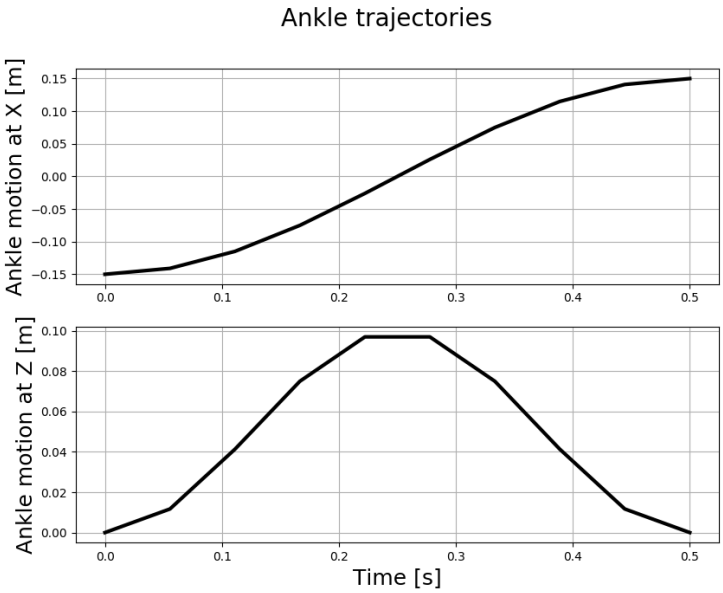

Figure 4.8: Desired position trajectories of the ankle in x and z directions.

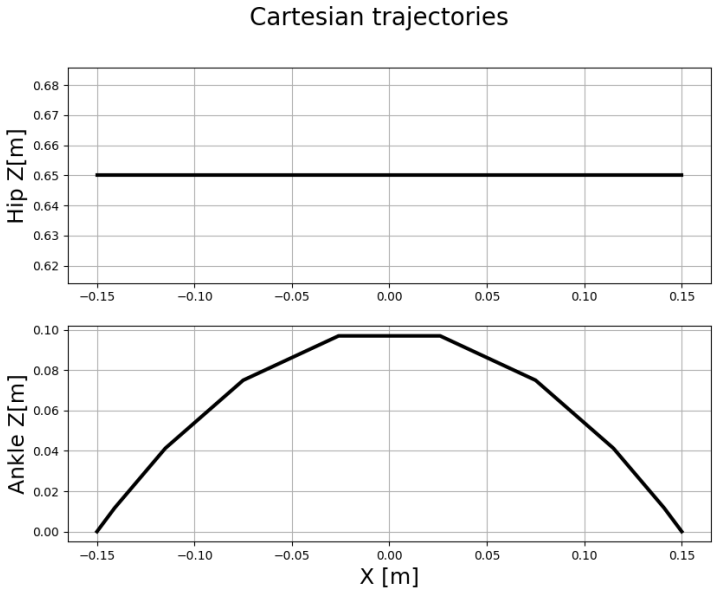

Figure 4.9: Desired Cartesian trajectories for the hip (CoM) and the ankle.

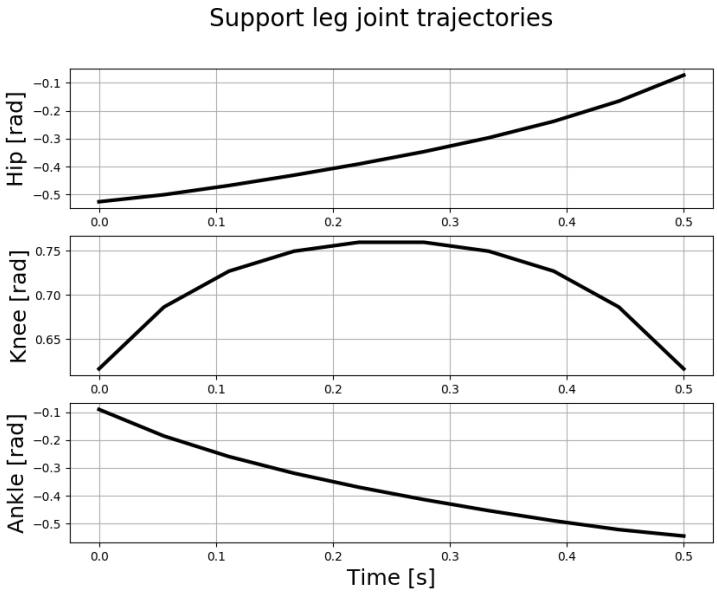

Figure 4.10: Desired position trajectories for the hip, knee and the ankle *pitch* angles for the support leg.

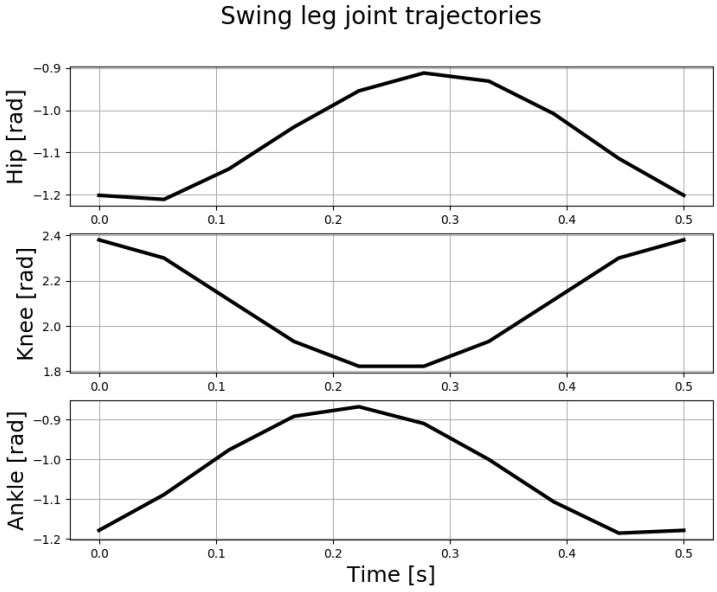

Figure 4.11: Desired position trajectories for the hip, knee and the ankle *pitch* angles for the swing leg.

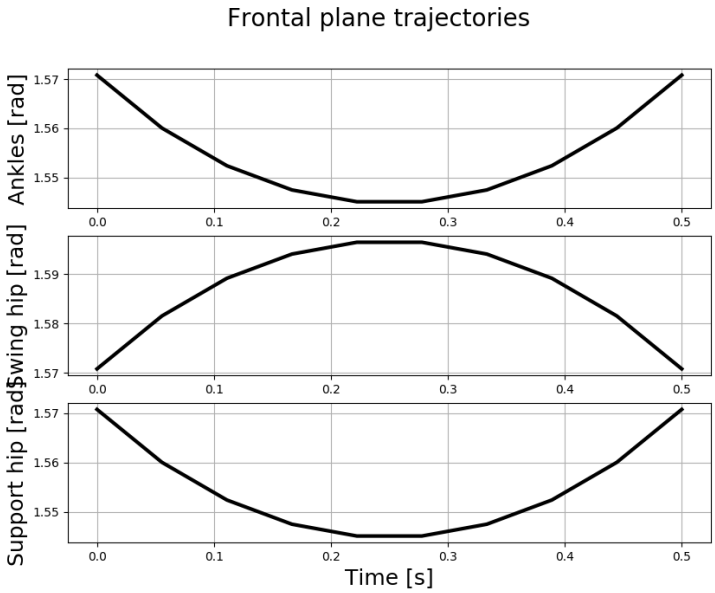

Figure 4.12: Desired position trajectories for both ankles *roll* angles, swing hip and support hip *roll* angles.

Control System Design

Consider that the humanoid robot dynamics (4.23), which may be subjected to disturbances and its parameters are unknown, is desired to follow the smooth trajectories vector q^d. The error signals of such tracking problem can be expressed as follows:

$$\begin{aligned} z_1 &= q^d - q \\ z_2 &= \dot{q}^d - \dot{q} \end{aligned} \tag{4.24}$$

Rewritting (4.24) as a first-order vector differential equation gives:

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ \ddot{q}^d - M(q)^{-1}\left(\tau - b(q,\dot{q}) + w\right) \end{bmatrix} \tag{4.25}$$

Note that $\ddot{q}^d$ is available from data; then, the control problem of trajectory tracking is a stabilizing problem of (4.25)[52]. Hence, invoking the Theorem 4.1, the feedback controller that stabilizes (4.25) is:

$$\tau = \hat{M}(q)(u - \ddot{q}^d) + \hat{b}(q,\dot{q}) \tag{4.26}$$

Where $\hat{M}(q)$ is a estimate value of $M(q)$ and $\hat{b}(q,\dot{q}) \triangleq (b(q,\dot{q}) - w)$ is considered as a total disturbance. In order to accomplish the stabilization of (4.25) by means of

the feedback linearization controller (4.26), an ADRC approach can be implemented, so that, an accurate knowledge of the parameters of (4.23), neither the disturbances affecting the system are required. Therefore, the following robust ESO is implemented to estimate the disturbances on-line:

$$\begin{cases} \dot{\hat{z}}_1 = \hat{z}_2 + \varepsilon \left(\dfrac{(z_1 - \hat{z}_1) + \mathtt{sign}(z_1 - \hat{z}_1)}{\varepsilon^2} \right) \\ \dot{\hat{z}}_2 = \hat{z}_3 + \left(\dfrac{(z_1 - \hat{z}_1) + \mathtt{sign}(z_1 - \hat{z}_1)}{\varepsilon^2} \right) + (u - \ddot{q}^d) \\ \dot{\hat{z}}_3 = \dfrac{1}{\varepsilon} \left(\dfrac{(z_1 - \hat{z}_1) + \mathtt{sign}(z_1 - \hat{z}_1)}{\varepsilon^2} \right) \end{cases} \tag{4.27}$$

Where $0 < \varepsilon < 1$ [25]. Hence, the different part from the canonical form of the cascade of integrators is the total disturbance given by [14]:

$$(M(q) - \hat{M}(q))(u - \ddot{q}^d) + \hat{b}(q, \dot{q}) \triangleq \hat{z}_3 \tag{4.28}$$

Therefore, the ADRC controller that accomplishes the robust feedback linearization based on Theorems 4.1 and 4.2 is:

$$\tau = \hat{M}(q)(u - \ddot{q}^d) + \hat{z}_3 \tag{4.29}$$

Where $u = -Kz$ is a stabilizer to achieve a satisfactory behavior of the remaining cascade of integrators.

Implementation

In order to evaluate the designed robust controller (4.29), simulation experiments were put into effect, comparing it with a feedback linearization control to demonstrate the ADRC disturbance capability. The control and the observation algorithms were programmed in Python language using the *odeint* function to solve the differential equations numerically. The parameters used are listed in Table 4.4 [53]. Furthermore, a disturbance vector $w = [0, -50, 0, -50, -60]^T\ [Nm]$ was applied to the biped robot for 200 ms, and also a -50 % variation of the D matrix was applied when using the ADRC, simulating parametric uncertainty.

Results

In Fig. 4.13, the behaviour of the joints of the biped robot in SSP using feedback linearization control are presented, and the error signals produced by the controller are depicted in Fig. 4.14. Notice that errors approximate asymptotically to 0. When the disturbance vector is applied, the tracking performance fails, as can be appreciated in Fig. 4.15; furthermore, error signals do not keep close to 0 while the disturbance

is applied. ADRC performance is similar to feedback linearization in absence of disturbances, as is depicted in Fig. 4.17 and Fig. 4.18, where the correct tracking and the asymptotic approximation to 0 of error signals can be appreciated. In contrast, Fig. 4.19 a successful tracking in presence of disturbances, which is reinforced with the error signals behaviour presented in Fig. 4.20, which remain close to 0 even in presence of disturbances and parametric uncertainty. A visualization of the robot walking can be watched using the link of Fig. 4.21.

Table 4.4: Humanoid robot dynamic parameters for numerical simulation.

Parameter	**Symbol**	**Value**	**Units**
Length of link 1 (support shank)	l_1	0.34	m
Length of link 2 (support thigh)	l_2	0.36	m
Length of link 3 (swing thigh)	l_3	0.36	m
Length of link 4 (swing shank)	l_4	0.34	m
Length of link 5 (trunk)	l_5	0.96	m
Length from joint 1 to CoM of link 1	l_{c1}	0.114	m
Length from joint 2 to CoM of link 2	l_{c2}	0.123	m
Length from joint 3 to CoM of link 3	l_{c3}	0.123	m
Length from joint 4 to CoM of link 4	l_{c4}	0.114	m
Length from joint 3 to CoM of link 5	l_{c5}	0.624	m
Mass of link 1	m_1	2.0812	Kg
Mass of link 2	m_2	2.0648	Kg
Mass of link 3	m_3	2.0648	Kg
Mass of link 4	m_4	2.0812	Kg
Mass of link 5	m_5	9.2110	Kg
Inertia of link 1	I_1	0.0164	Kg·m^2
Inertia of link 2	I_2	0.0103	Kg·m^2
Inertia of link 3	I_3	0.0103	Kg·m^2
Inertia of link 4	I_4	0.0164	Kg·m^2
Inertia of link 5	I_5	0.1054	Kg·m^2

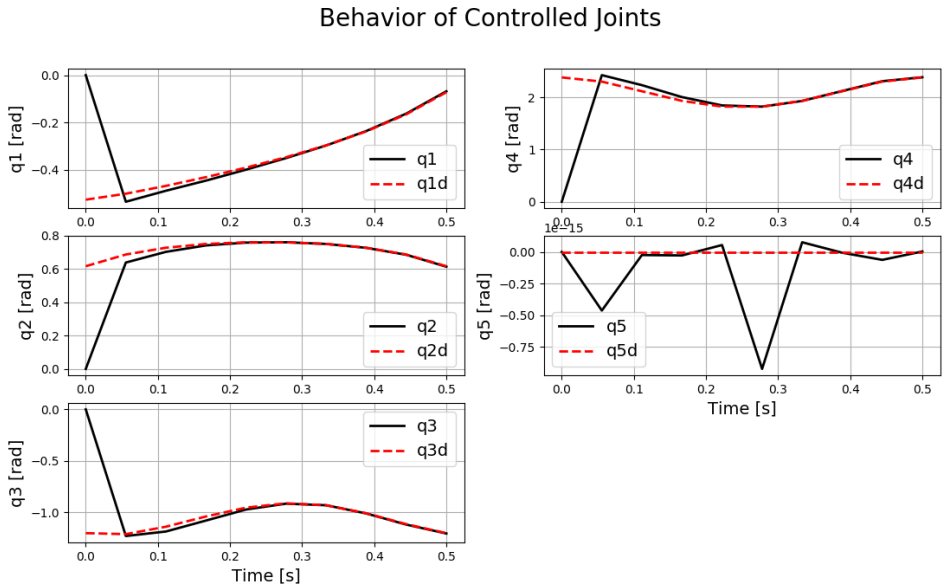

Figure 4.13: Behavior of the controlled joints while performing a step using feedback linearization control.

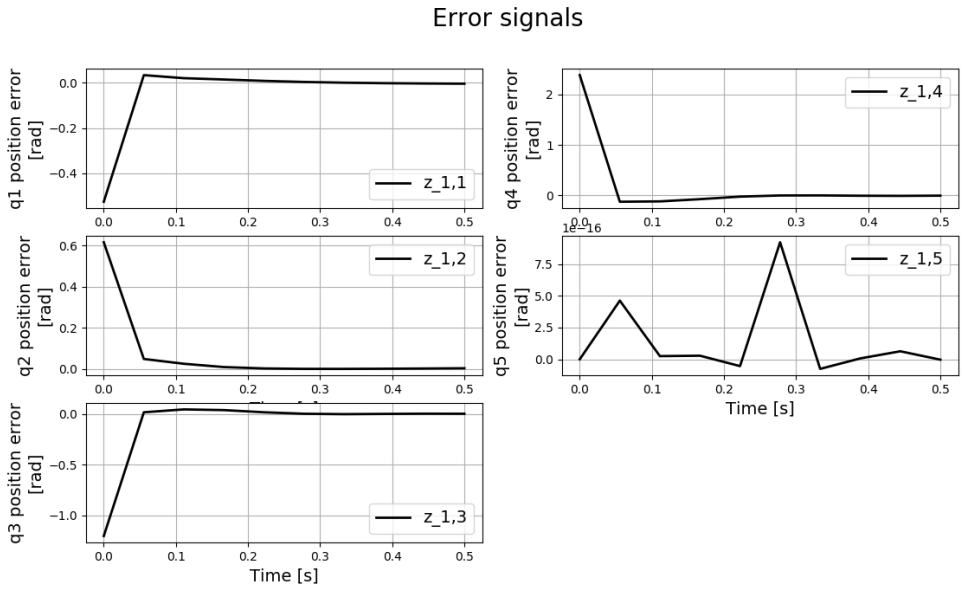

Figure 4.14: Error signals of the joints using feedback linearization control. All errors approximate asymptotically to 0.

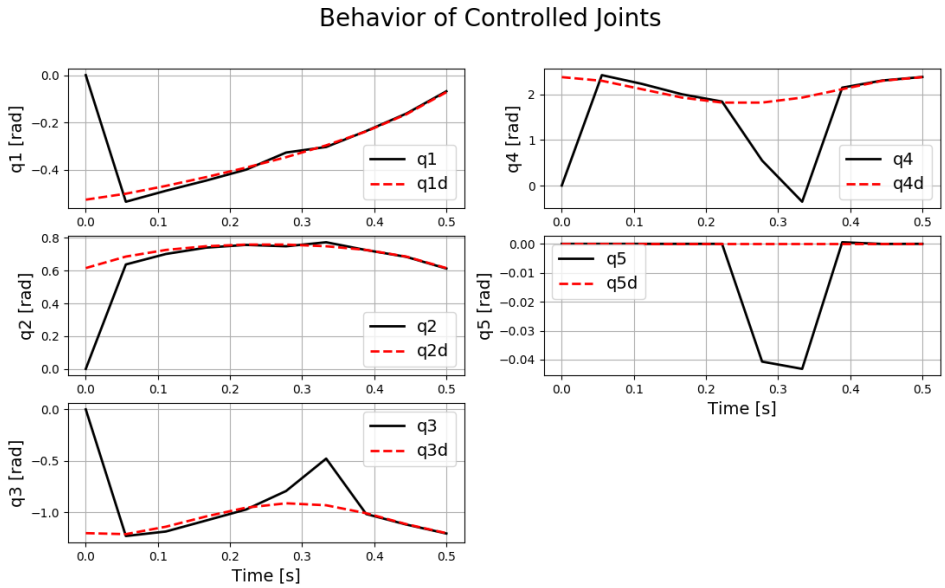

Figure 4.15: Behavior of the controlled joints while performing a step using feedback linearization control in presence of disturbances.

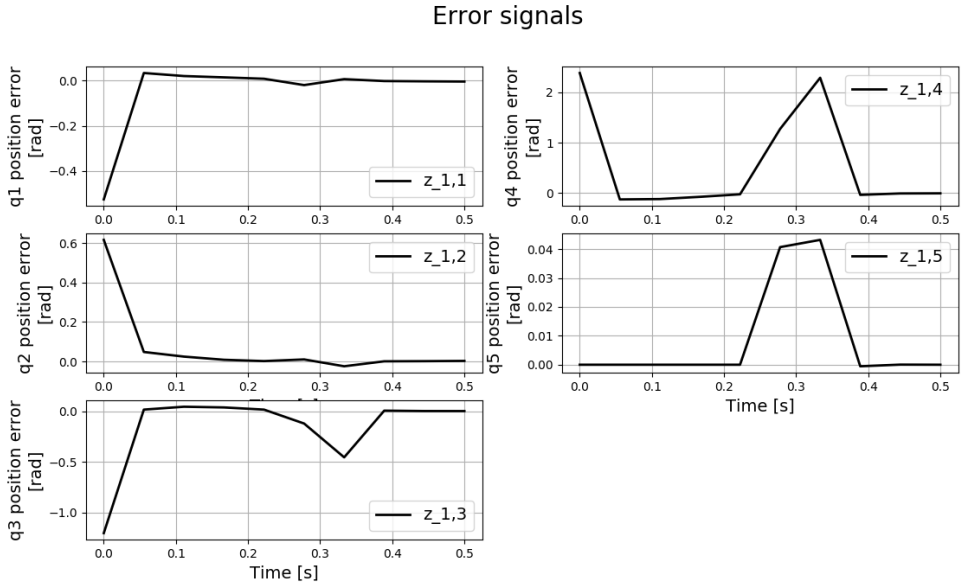

Figure 4.16: Error signals of the joints using feedback linearization control in presence of disturbances. Some errors do not keep close to 0.

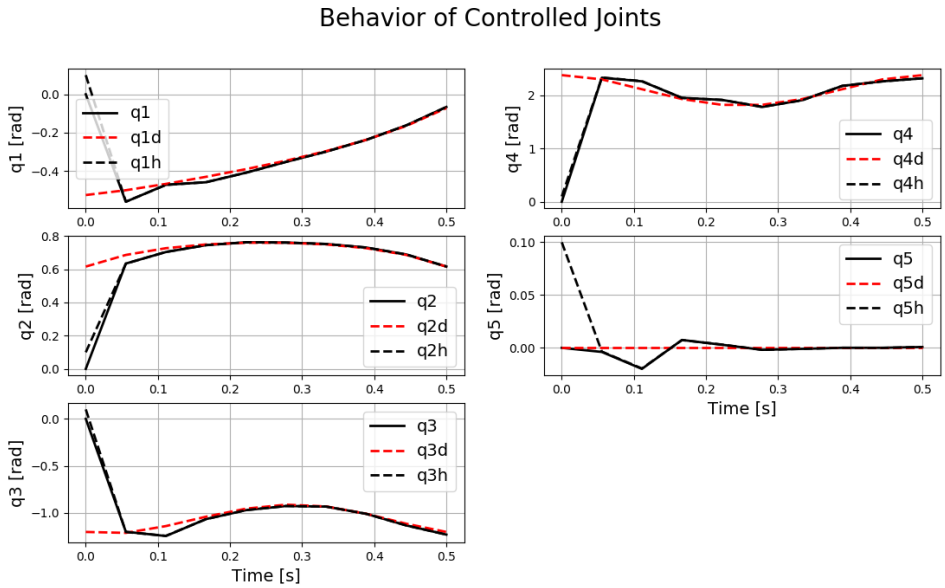

Figure 4.17: Behavior of the controlled joints while performing a step using ADRC.

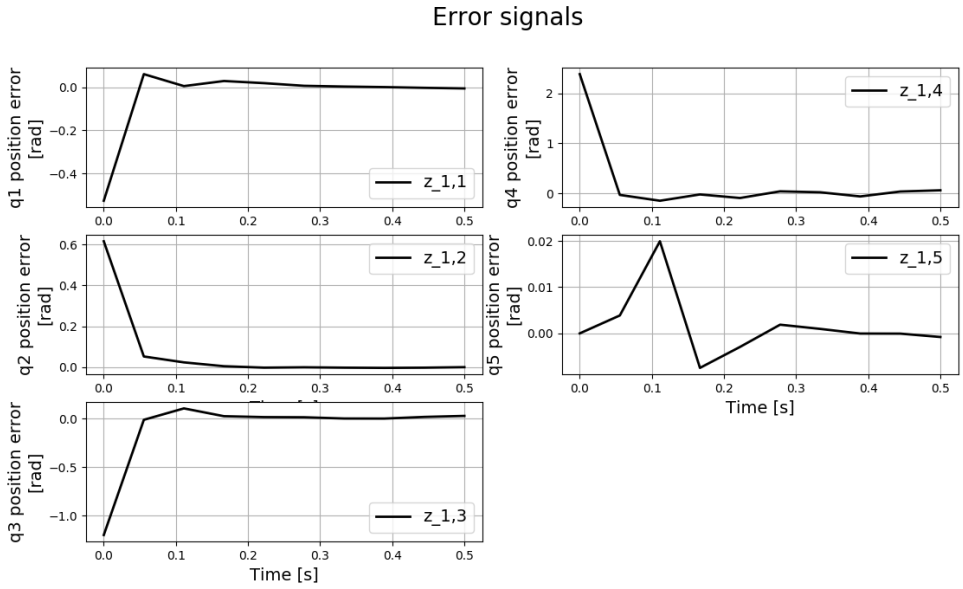

Figure 4.18: Error signals of the joints using ADRC. All errors approximate asymptotically to 0.

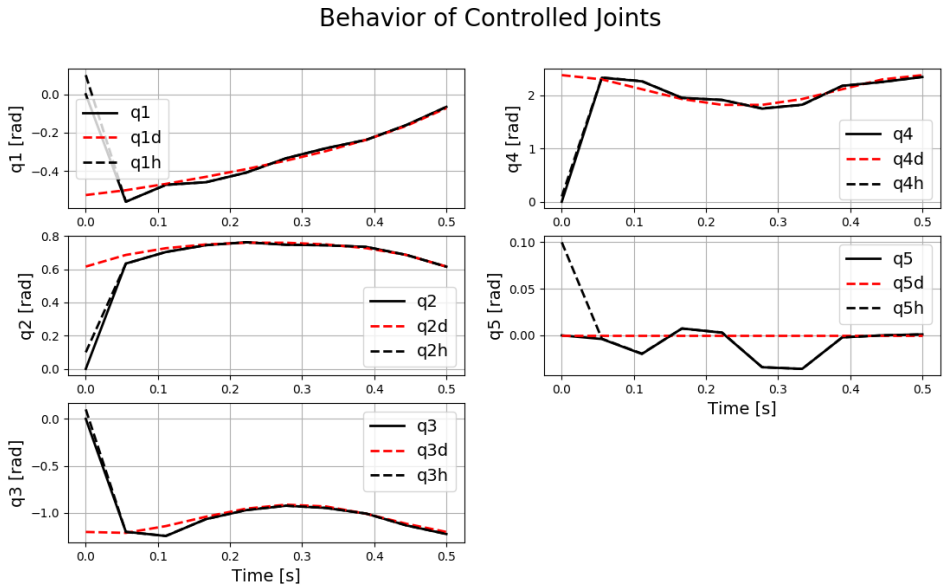

Figure 4.19: Behavior of the controlled joints while performing a step using ADRC in presence of disturbances.

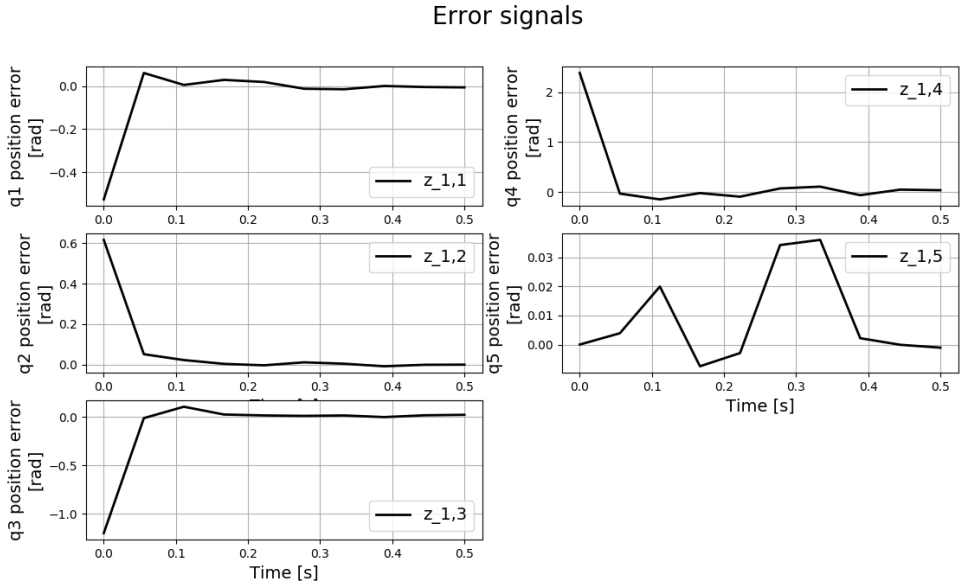

Figure 4.20: Error signals of the joints using ADRC in presence of disturbances. All errors keep close to 0.

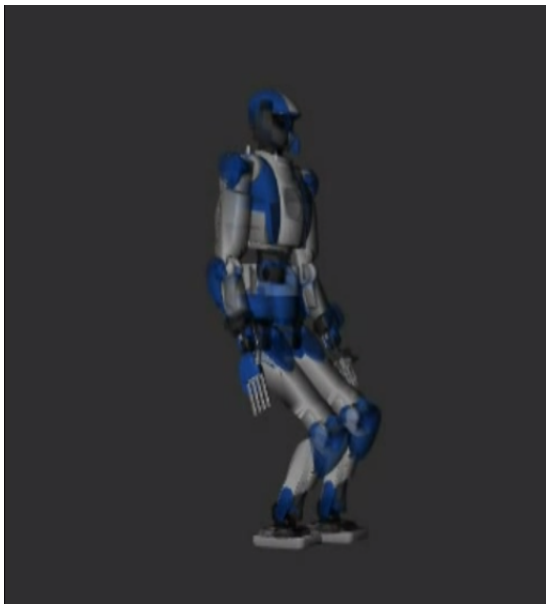

Figure 4.21: Visualization of the walking trajectories using the *RViz* simulator. Video available at `https://www.youtube.com/watch?v=WildzQwH9M0`.

4.3 Super Twisting Sliding Mode Control Embedded within QP Control for a Seesaw Actuated by Humanoid

QP control has been proposed in the robotics to solve the control problem of multi-body systems with floating bases subject to friction, contacts and motion limitations, appearing particularly to humanoid robots due to the complexity of their tasks, which normally involve the performing of several activities with each limb [43]. Such QP control target is to minimize the error of each weight prioritized task that must be performed by the humanoid [26], nevertheless, the task accomplishment is not robust. In order to enhance somoe tasks within the QP control, robust control strategies can be implemented. In this thesis work, the complex task of controlling a seesaw system using a humanoid robot standing on it is presented. The control targets are the following:

1. Control the seesaw angle by means of a torque signal applied on it by a humanoid robot.
2. Control the torque applied by the humanoid and measure it using the force sensors at its feet to accomplish 1.
3. Control the CoM of the robot to supply forces on its feet, keeping itself balanced over the seesaw.

A complete description of the humanoid-seesaw system is presented in Fig. 4.22, where the control variables and the contacts can be appreciated.

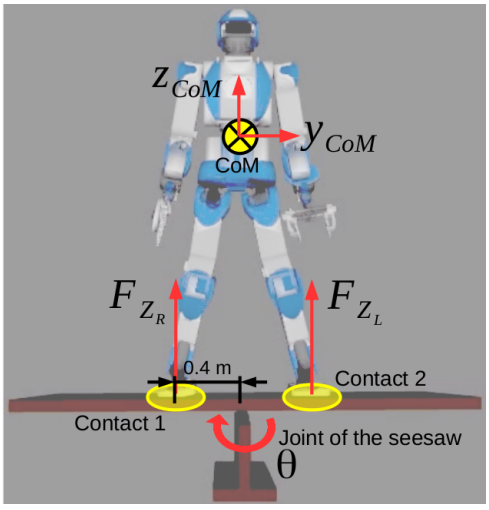

Figure 4.22: HRP4 humanoid balancing on a seesaw.

Control System Design

Consider the following system of 2 robots:

$$M_i(q_i)\ddot{q}_i + b_i(q_i, \dot{q}_i) = J_{f_i}^T + S_i\tau_i \tag{4.30}$$

Where $i = 1, 2$, are the humanoid and the seesaw indices respectively. Such system must perform the following tasks:

T1. **Posture Task**. Related to the local joints position control of the servomotors actuating the humanoid.

T2. **CoM Task**. Related to the Cartesian position control of the CoM of the humanoid.

T3. **Seesaw Posture Task**. Related to the local position control of the joint of the seesaw, which is underactuated.

T4. **End-Effector Task**. Related to the Cartesian pose control of the an end-effector of the humanoid, in this case, the right foot.

Therefore, the following QP control is implemented to accomplish such tasks [43]:

$$\min_{\ddot{q},\tau,\lambda} \sum_{k=1}^{M} w_k \|\ddot{\Theta}_k - \ddot{\Theta}_k^d\| \tag{4.31}$$

Subject to:

C1. Dynamics (4.30)

C2. The contact constraints between robots:

$$\lambda = (\lambda^-, \lambda^0) \geq 0 \tag{4.32}$$

C3. The contact constraints between the seesaw and the fixed environment: $J_0 = 0$

C4. The humanoid and seesaw kinematic constraints:

$$q_{min} \leq q \leq q_{max} \tag{4.33}$$

$$\dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max} \tag{4.34}$$

Where Θ_k are the tasks $T_1, ..., T_4$ listed above. The robot dynamics (4.30), the QP controller (4.31), the tasks and constraints are performed computationally. However, the QP control is not robust, at least respecting to the seesaw position regulation, and

either the robot balance. Hence, the following Super Twisting-Sliding Mode Control algorithmfor the position regulation of the seesaw is implemented:

$$\tau_{ss} = K_{ss}e_{ss} + \frac{1}{4}K_{ss}\dot{e}_s s - \mathcal{W} \tag{4.35}$$

Where the error signal is given by:

$$e_{ss} = q_{ss}^d - q_{ss} \tag{4.36}$$

The sliding state variable is given by:

$$\sigma = e_{ss} \tag{4.37}$$

And the components of the STSMC are:

$$\mathcal{W} = -K_{\mathcal{W}}|s|^{\frac{1}{2}}\texttt{sign}\,(\sigma) + \mathcal{V} \tag{4.38}$$

$$\dot{\mathcal{V}} = -K_{\mathcal{V}}\texttt{sign}\,(\sigma) \tag{4.39}$$

Notice that it was decided to use the controller (4.35), since no algebraic, neither numeric knowledge about the seesaw is available, but only the numerical computation of its dynamics embbeded within the QP solver. Continuing with the control system design, in order to achieve the Control Target 2, a torque τ_{ss} must be supplied to the seesaw provided by the humanoid. The control system developed to supply such torque signal to the seesaw by means of the humanoid is a Difference of Force controller of the form:

$$u_{\Delta_{CoM}} = \texttt{sat}\left|K_p e_f + K_d\dot{e}_f + K_i\int_0^t e_f(\mathcal{T})d\mathcal{T}\right| \tag{4.40}$$

Where $e_f = \Delta_f^f - \Delta_f$ is the error signal and $\Delta_f^d = \tau_{ss}/d_f$ is the desired difference of forces required to supply a torque τ_{ss}. d_f is the distance from the joint of the seesaw to the center of the contact of each foot of the humanoid. The difference of forces is computed using:

$$\Delta_f = F_{zR} - F_{zL} \tag{4.41}$$

Where F_{zR} and F_{zL} are the measurements of the force sensors at the soles of the HRP4 humanoid in z direction. The control signal $u_{\Delta_{CoM}}$ is the displacement of the CoM of the humanoid in y direction in order to project forces at the feet, as is depicted in Fig. 4.22. The controller (4.40) is saturated since large quick displacements of the CoM might cause the ZMP of the humanoid get outside the support polygon causing it to fall. Finally, the controller (4.40) supplies the input of the CoM task of the QP controller (4.31).

Implementation

The presented control strategy was implemented using the *mc-rtc* framework, which is a super set of spatial vector algebra-based functions that are helpful to compute

Table 4.5: Parameters of the control system proposed.

Parameter	Symbol	Value
Proportional gain of STSMC	K_{ss}	1600
Gain of STSMC	$K_{\mathcal{V}}$	15
Gain of STSMC	$K_{\mathcal{W}}$	15
Proportional gain of PID	K_p	0.005
Derivative gain of PID	K_d	0.001
Integral gain of PID	K_i	0.0005
Saturation limits of PID	$\texttt{sat}\lvert\cdot\rvert$	± 0.015

dynamics and control of robotic systems. The robot models are computed by means of computational graphs that are involved in a task-optimiztion problem subjected to constraints. The proposed controller was programmed using C++ and it was implemented in a virtual model of the HRP4 humanoid robot. The parameters used for the presented control scheme are listed in Table 4.5. The full algorithm was implemented as a Finite-State Machine control (FSMC) detailed as follows:

Algorithm 4.1 *Finite-State Machine Control.*

1. *Move the CoM over the left foot.*
   - ***Tasks (weight,stiffness):*** $\Theta_1(1000, 5), \Theta 2(1000, 5), \Theta 3(1000, 1000), \Theta 4(0, 0)$
   - ***Target:*** $CoM_y = 0.12$
   - ***Condition:*** $||\Theta_1|| \leq 0.02, ||\dot{\Theta}_1|| \leq 0.0$
2. *Move the right foot to the right.*

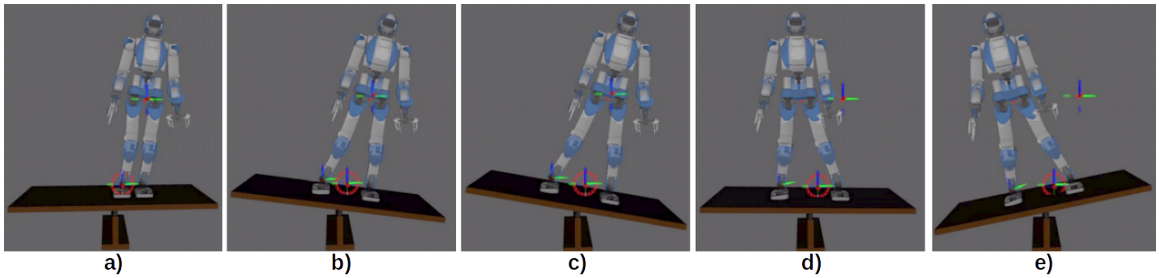

Figure 4.23: States of the FSMC: a) Move CoM Left. b) Move Rigth Foot Right. c) Move Rigth Foot Dowm. d) Move CoM Center. e) Force Control. Video available at `https://youtu.be/mmBlAmVddoU`.

- ***Tasks (weight,stiffness):***
  $\Theta_1(1000, 5), \Theta 2(1000, 5), \Theta 3(1000, 0.1), \Theta 4(1000, 2)$
- ***Target:*** $(x, y, z) = (0.0, -0.4, 0.1)$
- ***Condition:*** $||\Theta_4|| \leq 0.06, ||\dot{\Theta}_4|| \leq 0.0$

3. Move the right foot down.

- ***Tasks (weight,stiffness):***
  $\Theta_1(1000, 5), \Theta 2(1000, 5), \Theta 3(1000, 0.1), \Theta 4(1000, 2)$
- ***Target:*** $(x, y, z) = (0.0, -0.4, 0.06)$
- ***Condition:*** $||\Theta_4|| \leq 0.01 \otimes F_{zR} > 20, ||\dot{\Theta}_4|| \leq 0.0$

4. Move the CoM to the center.

- ***Tasks (weight,stiffness):***
  $\Theta_1(1000, 5), \Theta 2(1000, 5), \Theta 3(1000, 0.1), \Theta 4(0, 0)$
- ***Target:*** $CoM_y = -0.1$
- ***Condition:*** $||\Theta_1|| \leq 0.02, ||\dot{\Theta}_1|| \leq 0.0$

5. Force control.

- ***Tasks (weight,stiffness):***
  $\Theta_1(1000, 5), \Theta 2(1000, 5), \Theta 3(1000, 0.1), \Theta 4(0, 0)$

The complete states of the FSMC are dected in Fig. 4.23, where those states sequence can be appreciated.

Results

The behavior of the controlled multirobot system using the STSMC embedded within the QP control are presented below. Fig. 4.24 shows the behavior of the seesaw controlled by the humanoid using a target of $q_{ss} = 0\,[rad]$. Notice that the angle regulation is accomplished. In Fig. 4.25, the behavior of the seesaw using a control target of $q_{ss} = 0.1\,[rad]$ is presented, where it can be appreciated a steady state error at position regulation. Difference of forces control tracks the trajectory correctly. Another experiment was performed using a target of $q_{ss} = -0.15\,[rad]$, where it can be noticed that the position regulation is achieved with significatively low steady-state error and the control of difference of forces tracks the trajectory asymptotically.

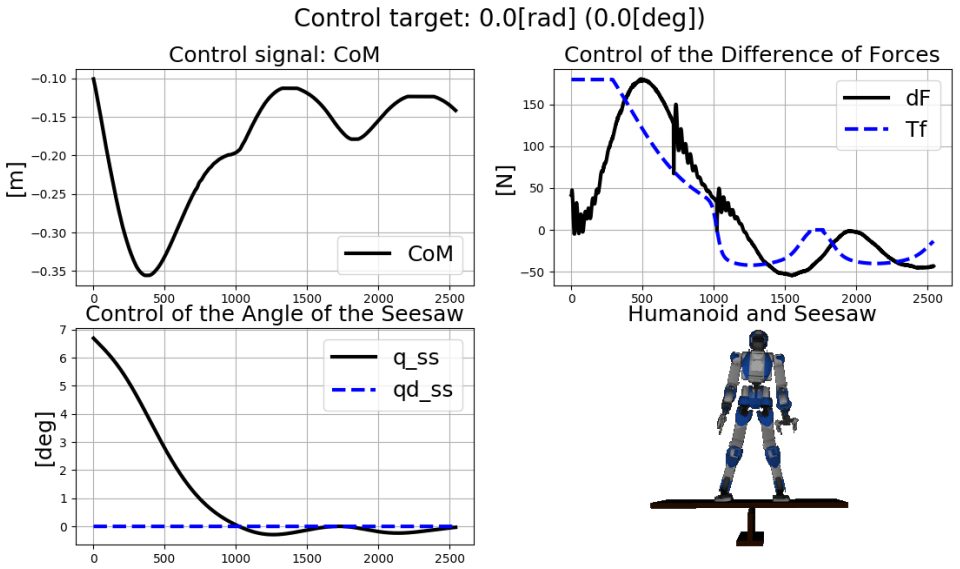

Figure 4.24: Behavior of the multirobot system with the proposed STSMC with a target position of 0.0 $[rad]$.

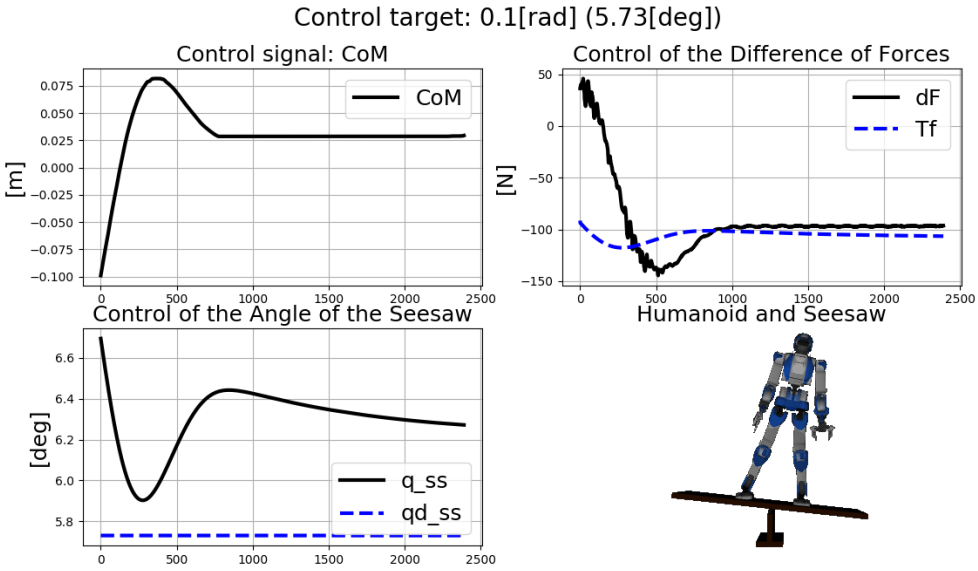

Figure 4.25: Behavior of the multirobot system with the proposed STSMC with a target position of 0.1 $[rad]$.

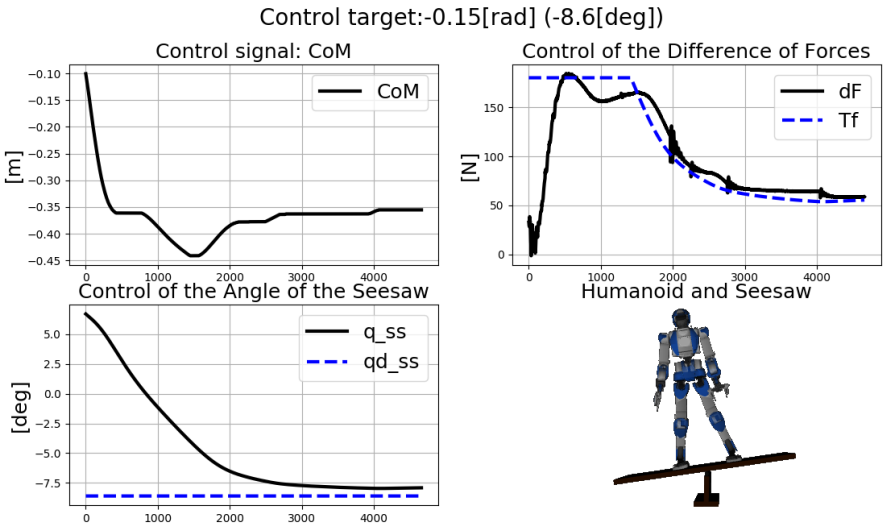

Figure 4.26: Behavior of the multirobot system with the proposed STSMC with a target position of $-0.15\,[rad]$.

4.4 ADRC for Quadrotor Autonomous Flight

Control System Design

Consider the dynamics of a quadrotor airframe given by (3.60), affected by disturbances, and the *Parrot Bebop* quadrotor internal control system that simplifies the dynamics as follows:

$$\begin{aligned} m\ddot{x} &= \bar{u}_x + w_x \\ m\ddot{y} &= \bar{u}_y + w_y \\ m\ddot{z} &= \bar{u}_z + w_z \\ J_3\ddot{\psi} &= \bar{u}_\psi + w_\psi \end{aligned} \tag{4.42}$$

Hence, *Parrot Bebop* dynamics can be considered as the following linear system with disturbances:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/J_3 \end{bmatrix} \begin{bmatrix} u_x + w_x \\ u_y + w_y \\ u_z + w_z \\ u_\psi + w_\psi \end{bmatrix} \tag{4.43}$$

Recalling to the central idea of ADRC, such equations can be represented in the form (2.6) using: $\xi = \left[x, \dot{x}, y, \dot{y}, z, \dot{z}, \psi, \dot{\psi}, w^T\right]^T$, $w = [w_x, w_y, w_z, w_\psi]$ and:

$$A_e = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/J_3 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with $y = C_e\xi$. Hence, an ESO of the form (2.7) is implemented to estimate the disturbances, using the bandwidth $\omega_0 = 200\pi\,[rad/s]$, which is based on the frequency of the sensors measurements. Then, the observer gain matrix is given by:

$$L = \begin{bmatrix} 3\omega_0 & 0 & 0 & 0 \\ 3\omega_0^2 & 0 & 0 & 0 \\ 0 & 3\omega_0 & 0 & 0 \\ 0 & 3\omega_0^2 & 0 & 0 \\ 0 & 0 & 3\omega_0 & 0 \\ 0 & 0 & 3\omega_0^2 & 0 \\ 0 & 0 & 0 & 3\omega_0 \\ 0 & 0 & 0 & 3\omega_0^2 \\ \omega_0^2 & 0 & 0 & 0 \\ 0 & \omega_0^2 & 0 & 0 \\ 0 & 0 & \omega_0^2 & 0 \\ 0 & 0 & 0 & \omega_0^2 \end{bmatrix} \tag{4.44}$$

In order to enhance and robust the ESO performance, a sliding mode term was added to the Luenberger observer, giving the finally form of:

$$\dot{\hat{\xi}} = A_e\hat{\xi} + B_e u + L(y - \hat{y}) + sign(y - \hat{y}) \tag{4.45}$$

Then, the estimated disturbances are given by:

$$\begin{aligned}
\hat{\xi}_9 &= \omega_0^2(\xi_1 - \hat{\xi}_1) + sign(\beta_5(\xi_1 - \hat{\xi}_1)) \\
\hat{\xi}_{10} &= \omega_0^2(\xi_3 - \hat{\xi}_3) + sign(\beta_6(\xi_3 - \hat{\xi}_3)) \\
\hat{\xi}_{11} &= \omega_0^2(\xi_5 - \hat{\xi}_5) + sign(\beta_7(\xi_5 - \hat{\xi}_5)) \\
\hat{\xi}_{12} &= \omega_0^2(\xi_7 - \hat{\xi}_7) + sign(\beta_8(\xi_7 - \hat{\xi}_7))
\end{aligned} \tag{4.46}$$

Substituting the value of the observer bandwidth, the eigenvalues of $A - LC$ considering only the linear part of the observer are:

$$\begin{aligned}
l_{1,3,5,7} &= -942 + 544j \\
l_{2,4,6,8} &= -942 - 544j \\
l_{9,10,11,12} &= -394785
\end{aligned} \tag{4.47}$$

Which ensures the convergence of the observer and the reconstruction of the disturbances.
Now consider the disturbed representation of quadrotor (4.43). Since it is a *cascade of integrators*, all of its poles are 0. Then, a state-feedback controller to firstly stabilize the system was designed using Ackerman's formula and the mass and inertia of the quadrotor as $m = 0.48\,[Kg]$ and $J = 0.01\,[Kgm^2]$, which yields:

$$K = \begin{bmatrix} 1.08 & 0 & 0 & 0 \\ 1.44 & 0 & 0 & 0 \\ 0 & 1.08 & 0 & 0 \\ 0 & 1.44 & 0 & 0 \\ 0 & 0 & 1.08 & 0 \\ 0 & 0 & 1.44 & 0 \\ 0 & 0 & 0 & 0.0225 \\ 0 & 0 & 0 & 0.03 \end{bmatrix}^T \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \psi \\ \dot{\psi} \end{bmatrix} \tag{4.48}$$

Then, the eigenvalues of $A - BK$ are:

$$p_{1,\ldots,8} = -1.5 \tag{4.49}$$

So now the closed-loop system is stable. In order to track a reference signal, the pose errors $e_x = x - x_d, e_y = y - y_d, e_z = z - z_d$ and $e_\psi = \psi - \psi_d$ were introduced within the final controller in addition to the disturbance rejection part, yielding:

$$\begin{aligned}
u = -\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & J_3 \end{bmatrix} K \left[e_x, \dot{x}, e_y, \dot{y}, e_z, \dot{z}, e_\psi, \dot{\psi}\right]^T - \\
\left[\hat{\xi}_9, \hat{\xi}_{10}, \hat{\xi}_{11}, \hat{\xi}_{12}\right]^T
\end{aligned} \tag{4.50}$$

4.4.1 Hover control using inertial feedback

Inertial feedback using the IMU

The quadrotor used in this thesis work is equipped with proprioceptive sensors such as a 9-DOF inertial measurement unit (IMU), an ultrasonic sensor for height measurement up to 16 ft, a pressure sensor that complements the last one for heights beyond 16 ft, a vertical stabilization camera and a global navigation satellite system (GNSS) chipset (GPS + GNSS). The Bebop 2 is also equipped with an embedded full-HD 14 Mpx fish-eye camera, which is helpful as an exteroceptive sensor for simultaneous localization and mapping, obstacle detection and control applications. This quadrotor reaches up to 60 km/h horizontal speed and up to 21 km/h vertical, furthermore, it resists head winds up to 63 km/h. Another important feature is the included hover regulation, which is useful because the gravity compensation can be omitted during control design.

Implementation

The middleware *Robotics Operating System* (ROS) was developed in 2007 by Stanford Artificial Intelligence Laboratory (SAIL) to create functionalities that can be shared and used in other robots or by a single robot to save computational resources performed by hardware [46]. In this thesis work, ROS was implemented to perform the control algorithms, which is helpful to execute simultaneous tasks that imply multiple sensors reading and some security functions such as forcing the quadrotor to hover or land in case of autonomous behavior fail. The control system and the sliding mode extended state observer algorithms were developed using C++ language including the ROS libraries and functions. The ultrasonic sensor was used for height hovering measurement. The ROS nodes interaction is depicted in Fig. 4.27.

Results

The experiments to test the proposed control strategy consisted in giving a 0.6 m reference altitude to the quadrotor after its first default 1 m hover stabilization.Once the UAV is stable, some shoves were applied downwards and then its response was observed during 20 seconds, as is illustrated in Fig. 4.28. The behavior of the quadrotor during both ADRC and PD control experiments is presented in Fig. 4.29 and 4.30; notice that the behavior measurements include the take-off, default hover, controlled hover, disturbed hover and landing stages and the samples in x axis scale is 10. It worth to mention that the feedback linearization part for gravity and attitude compensation is included in Bebop internal control; nevertheless, the altitude regulation can be still implemented externally. Another important remark is that at such desired altitude, the wind gusts generated by the rotors bounce against the ground and return upwards affecting the hovering stabilization. Note in Fig. 4.29 that the quadrotor hover controlled by PD is significantly affected by vortex effect presenting oscillations

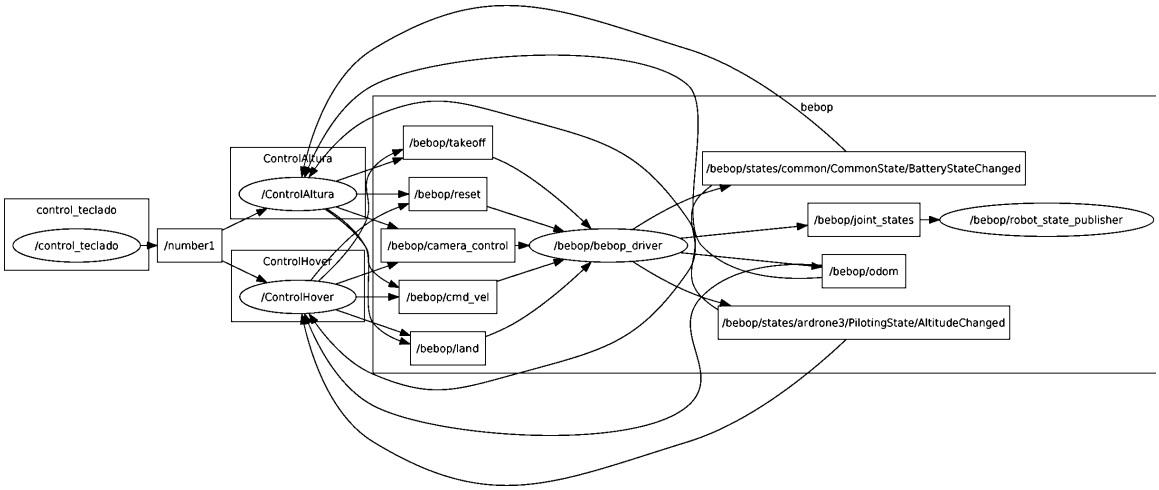

Figure 4.27: ROS *rqt plot* showing the nodes and topics of the implemented control system for the quadrotor autonomous flight using inertial feedback.

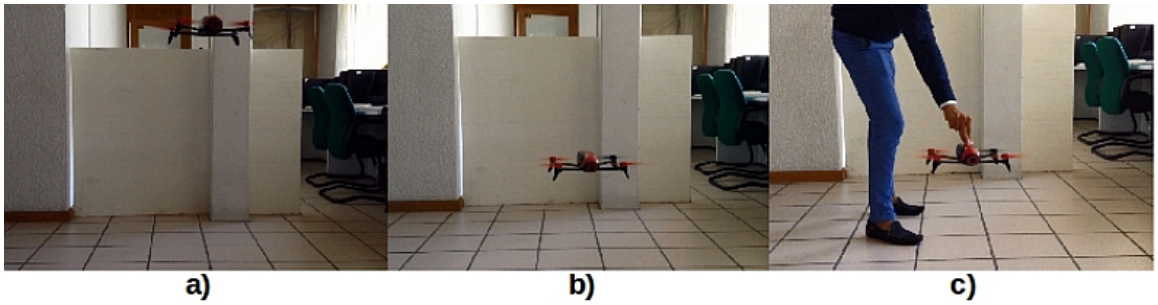

Figure 4.28: Hover control stages: a) Initial hovering at 1 m. b) Height regulation to 0.6 m. c) Applied disturbances. Video available at `https://youtu.be/x_wLSkgdnag`.

during the controlled hover stage of up to 15%. Furthermore, after the vertical downwards shoves the quadrotor presents an overshoot of up to 40% and a recuperation before about 5000 samples or 1.5 seconds. ADRC results, are depicted in Fig. 4.30, where can be observed that the controlled hover presents smaller oscillations of up to 3%, which in comparison with PD controller, shows better performance. In addition, when the downwards shoves are applied to the quadrotor, the maximum overshoot is of 20% with the same recovery time. These quantitative results are presented also in Table 4.6, which includes overshoot and oscillations amplitude peak percentages and approximated recovery time.

Table 4.6: Comparison between PD and ADRC hover controllers.

Controller	**Oscillation peak time**	**Maximum overshoot**	**Recovery time**
PD	15%	40%	≈ 1.5s
ADRC	3%	20%	≈ 1.5s

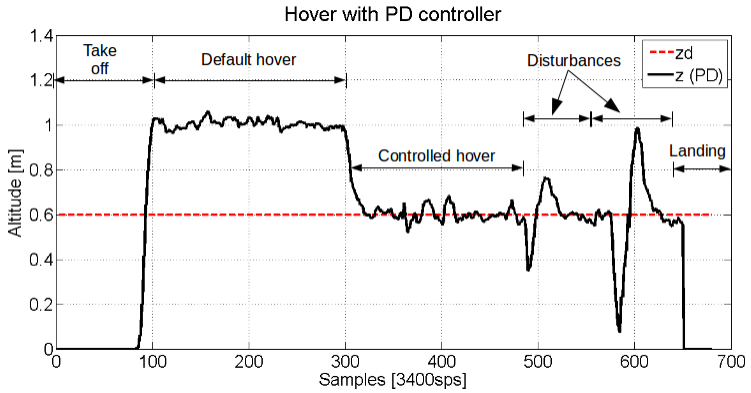

Figure 4.29: Quadrotor hover with PD control.

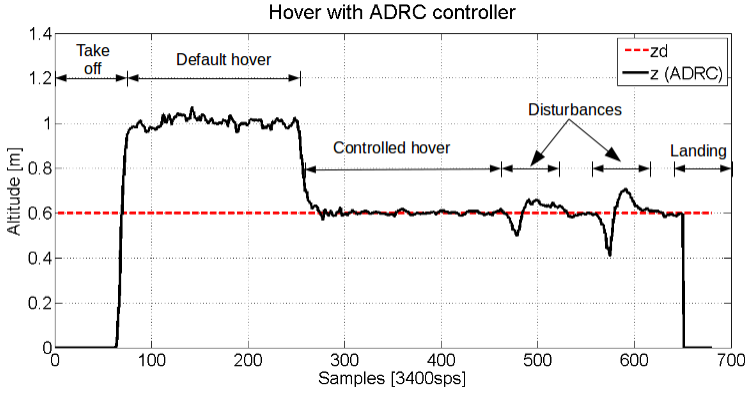

Figure 4.30: Quadrotor hover with ADRC control.

4.4.2 Pose regulation using inertial feedback

Implementation

The successful hover control results encouraged to the implementation of ADRC now for pose regulation. So that, an experiment consisting on programming to the quadrotor a target 4 DOF pose $\left[\xi^{ref}, \psi^{ref}\right]^T = [0.5, 0.2, 0.5, 28.5]^T$ from an initial pose $[\xi_0, \psi_0]^T = [0, 0, 1, 0]^T$ using a PD controller and the proposed ADRC was put into effect.

Results

In Fig. 4.31, the behavior of the generalized coordinates is presented during a 25 seconds experiment. Note that x and y positions are never reached with PD control, however, the quadrotor controlled by the ADRC achieves the goal position. Furthermore, position in z with ADRC presents less-amplitude overshoots than PD. The downward peaks in both cases are not a real measurements; during some experiments, inaccuracy of the ultrasonic sensor was realized, so that, an alternative technique should be put into effect to enhance the accuracy of height measurements. In the case of ψ, ADRC also produces less amplitude oscillations and also faster recovery. The 3D behavior of the quadrotor is presented in Fig. 4.32, where it can be appreciated in blue line that the quadrotor does not achieve the target pose, represented with a red dot, with the PD controller. However, such pose is reached with the ADRC, which is represented with the black line.

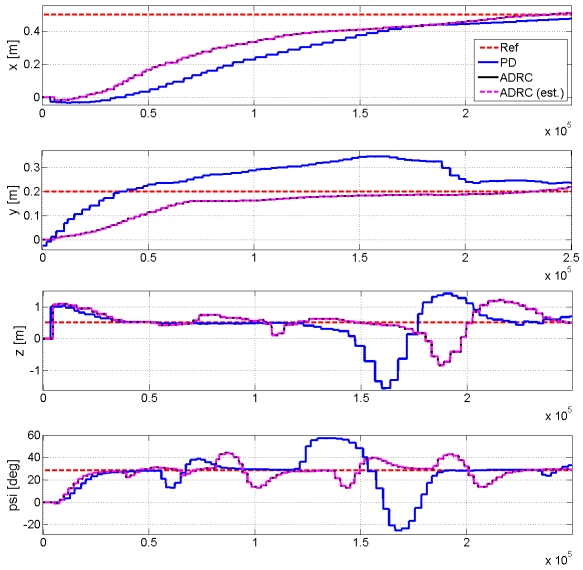

Figure 4.31: Generalized coordinates behavior with both PD and ADRC control schemes. The samples were acquired during 25 seconds.

4.4.3 Pose regulation using visual feedback

Visual feedback: ORB-SLAM

ORB-SLAM is a feature-based monocular video SLAM (VSLAM) algorithm that works in indoor and outdoor environments by means of *Oriented FAST* and *Rotated BRIEF* (ORB) feature extraction technique [54]. This algorithm is compound by the following main stages: tracking, local mapping and loop closing, and it provides the camera pose matrix and a point-cloud map of the environment. In Fig. 4.33, a

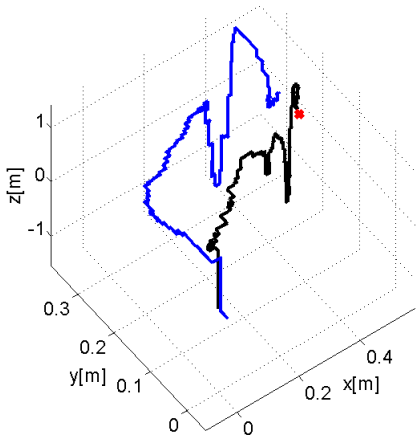

Figure 4.32: 3D view of quadrotor behavior and target pose with both PD and ADRC control schemes.

couple of frames obtained from the *Parrot Bebop* camera are presented, where the detected features in each frame are shown; such features are useful for visual SLAM task activities. It was decided to use ORB-SLAM it this thesis work, since it presents the fastest and most accurate performance for visual localization among all reported techniques in literature [55], in addition to the possibility of using it for indoor or outdoor navigation; furthermore, it is able to be performed using ROS, so that, the control system and vision nodes can be executed in parallel threads.

Figure 4.33: Examples of feature detection using ORB-SLAM.

Coordinate Frames Transformations

Within the presented control strategy, ORB-SLAM works as a pose feedback for the closed-loop system, however, some transformations should be applied to match the ORB-SLAM reference frame with the airframe coordinates. In Fig. 4.34, the considered coordinate frames are depicted. Note that the first frame OS is provided directly by ORB-SLAM as a local frame representing the camera pose. Since the target is to control the global vehicle's pose with the camera on-board respect to a global world frame, a transformation from local coordinates to global world coordinates should be applied to yield the new frame GOS. Such frame is then represented in a conventional form used in robotics literature, named GR. Finally, the standard frame representation for UAVs, but in global coordinates is GQ. The corresponding transformations are given by:

$$\begin{bmatrix} x_{GQ} \\ y_{GQ} \\ z_{GQ} \end{bmatrix} = R_y(\theta_{OS}) R_x(-90^o) R_z(-90^o) R_x(180^o) \begin{bmatrix} x_{OS} \\ y_{OS} \\ z_{OS} \end{bmatrix} \tag{4.51}$$

Where θ_{OS} is the orientation angle about y_{OS} provided by ORB-SLAM pose matrix. Hence, for a simple writting let $\left[x_{GQ}, y_{GQ}, z_{GQ}\right] = [x, y, z]$, and the orientation yaw angle about z_{GQ} is represented as ψ. The scale of the ORB-SLAM information was computed using the inertial sensors data as in [56].

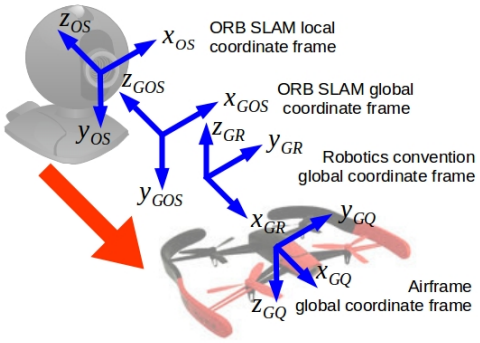

Figure 4.34: Coordinate frames transformation.

Results

The autonomous navigation algorithm selected in this work is the Dead Reckoning (DR), which consists on estimating and advancing from a current position to another previously determined; it is frequently used in mobile robots navigation for its simplicity, and generally consists in a list of desired straight forward motions with orientation targets. A disadvantage of DR is that the position errors are cumulative in open loop, however, since the ORB-SLAM supplies the pose of the quadrotor respect to objects detected in a map, the mentioned drawback can be effectively overcome. The presented controller is implemented to send $[x, y, z, \psi]$ commands to the

quadrotor in order to ensure the DR targets completion. The presented ORB-SLAM based ADRC was implemented using ROS middleware using the nodes depicted in Fig. 4.35, where also the interaction of such developed nodes and their topics can be appreciated. ROS is helpful to compute such nodes or programs in parallel threads, which enhances the control system performance. Furthermore, a block diagram of the control system is displayed in Fig. 4.36; notice that the DR targets supplied as references and the ORB-SLAM feedback signal feed the ADRC, wich consequently commands the quadrotor motion. In order to test the developed controller, the following target pose was commanded:

$$\xi_d = \begin{bmatrix} 6m & 0.3m & 1m & -90^o \end{bmatrix} \tag{4.52}$$

A frame sequence of the *Parrot Bebop* on-board camera during the commanded flight is presented in Fig. 4.37, where the ORB-SLAM features detected at the outdoor environment can be appreciated, and also the trajectory performed by the quadrotor. The behavior of the controlled height is depicted in Fig. 4.38 a); notice that the height reference is missed about $\pm 2cm$ due to wind disturbances, but it is compensated by ADRC. Fig. 4.38 b) presents the distance control behavaior along x axis, where an overshoot of about $30cm$ can be observed, but it was compensated successfully by the ADRC. In Fig. 4.38 c), the orientation behavior is presented first, keeping a $0rad$ angle, and once it reaches the desired x position, a $-\pi/2rad$ target orientation was performed. All the pose measurements of the experiment are depicted in Fig. 4.38 d), where the constant height can be appreciated; furthermore it can be noted from the scaled x motion that once the quadrotor reaches the desired distance, both orientation and y controlled motions are performed. Finally, the desired trajectory in xy plane is presented in Fig. 4.39, where, in the part a), it can be observed that the quadrotor reached the target pose successfully despite the wind disturbances intrinsic to an outdoor environment; in addition, the part b) of the figure depicts the trajectory computed by the ORB-SLAM using the frames, which is consistent with the trajectory displayed in the part a).

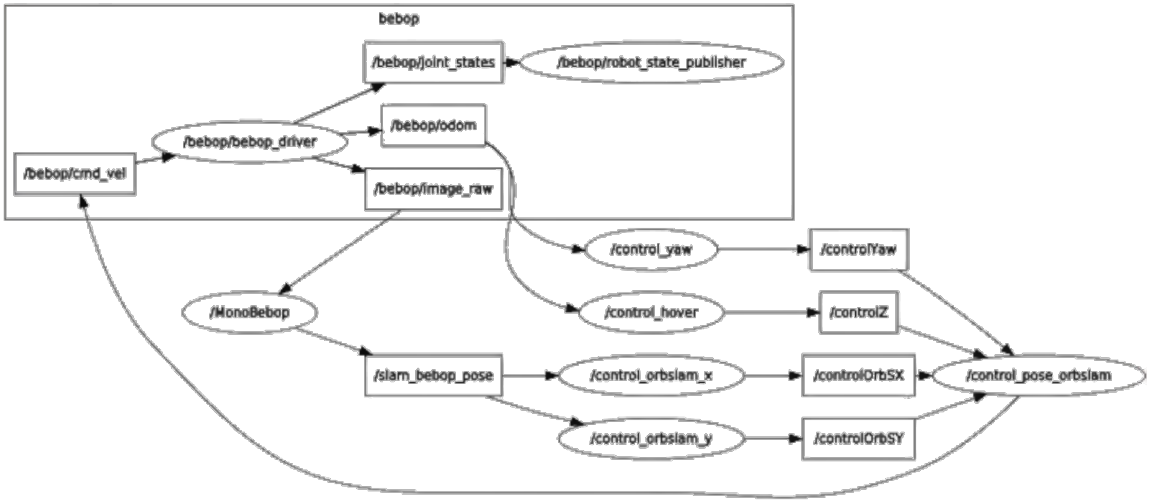

Figure 4.35: ROS *rqt plot* showing the nodes and topics of the implemented control system for the quadrotor autonomous flight using visual feedback.

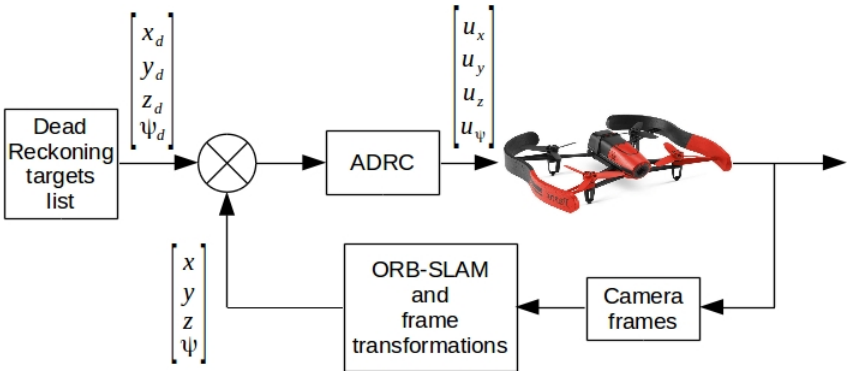

Figure 4.36: Block diagram of the controller for quadrotor autonomous flight.

Figure 4.37: Some ORB-SLAM frames captured by *Parrot Bebop* camera during the performance of the experiment.

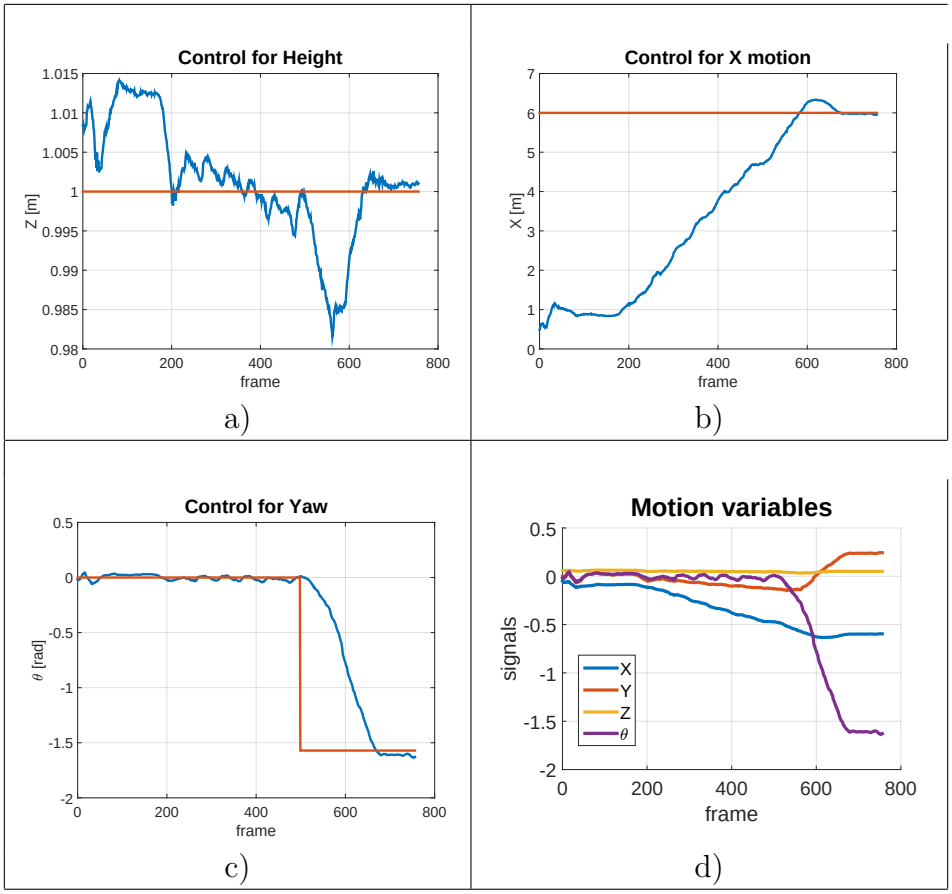

Figure 4.38: Controlled variables of the *Parrot Bebop* quadrotor using ORB-SLAM based ADRC.

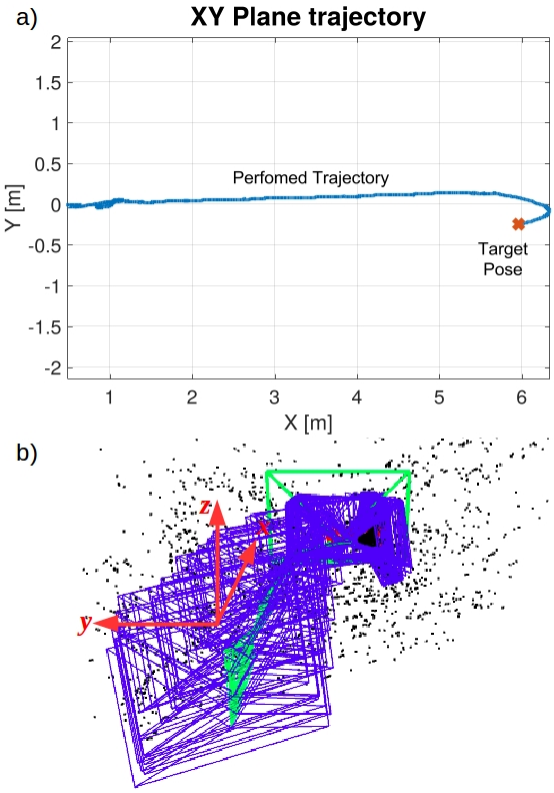

Figure 4.39: Performed trajectory: a) xy plane with target pose. b) ORB-SLAM frames.

5

Conclusions

The aim of this research work is to develop closed-loop controllers based on recent robust control strategies, in order to deal with the disturbances affecting autonomous robotic systems. The work is mainly focused on humanoid robots and quadrotor MAVs, that are robotic systems that share their environments with humans and other natural facts that inherently yield disturbances on their behavior whilst performing an assigned task.

Firstly, mathematical models of our humanoid robots and MAVs were developed, using approaches reviewed in the literature. It can be concluded that those models were useful for control systems synthesis and simulations, since the gains of the controllers obtained theoretically were up to $\pm 8\%$ alike than the used in implementation.

Another important contribution of this work, from the control theory perspective, is the implementation of the sliding-mode approach in order to improve the extended-state observer technique, improving the performance of the Luenberger-ESO based ADRC. An example of the improvement for humanoid robots motion control is detailed in [49], where the observation error was reduced from 7.04 % to 1.01 %, so there is an improvement of the observation technique.

The ADRC was preferred among other robust control schemes since it estimate and reject disturbances on line. Such innovative ADRC, enhanced with sliding mode observers, were implemented on humanoid robots complex tasks such as balance, walking showing successful simulation and implementation results. The ADRC improved the balance of a humanoid robot despite the applied shoves, reducing the ZMP oscilation amplitude up to 40 %, which means a less risk of falling rather than when the humanoid is controlled by an LQR. Such result is consistent with the research objective of accomplishing the task of balance even in presence of disturbances.

For the case of biped walking task, a comparison of the ADRC against a feedback linerization was put into effect. Both controllers led the error signals to 0 asymptotically without disturbances. In presence of disturbances, it was demonstrated the capability of the ADRC to reject them, by keeping the error less than 0.04 $[rad]$ close to 0. In contrast, some errors with feedback linearization exceeded the value of 2

[*rad*]. Thus, another research objective was accomplished.
For the control of a seesaw task of a humanoid robot, a Super-Twisting SMC was implemented, since the model of the system dynamics was totally unknown and it only requires the error signals to be developed. The developed controller was programmed as a meta-task within a QP control solver and it was tested in a highly realistic simulator. The position error of the seesaw was up to 0.77 [deg], from which can be concluded that the Super-Twisting SMC accomplished the completion of the tasks, even for a highly uncertain system, as is requested by the research objectives.
The autonomous flight of quadrorotor MAVs is a challenge from control systems perspective, since the measurements are noisy, sometimes inaccurate and it is a highly susceptible-to-disturbances system. In this work, both inertial and vision-based feedback techniques were implemented to perform the ADRC algorithm to deal with disturbances affecting the hover and autonomous flight of a quadrotor. Such controller was compared against a PD, in order to show the disturbance rejection capability. The ADRC showed less oscillation peak time, less maximum overshoot and the same recovery time than the PD for hover regulation. Whilst performing pose regulation, the PD could not reach the target pose. In both cases, it can be concluded that the ADRC performance was useful to reject disturbances. During the autonomous navigation in outdoors environment, the quadrotor reached the target pose, confirming that the ADRC rejected the environment disturbances, as was expected in the hypothesis.
From all the developed work and the successful results it can be generally concluded that the improvement of the ADRC using sliding-mode observers is a powerful tool to control complex robotic systems, since it is simple to implement and do not only deal with model inaccuracy, but also with actual unknown disturbances that affect the robots in real time. As future work, it would be interesting to compare the proposed robust control techniques against other robust approaches to evaluate their robustness and performance.

References

[1] L. Guo and S. Cao, "Anti-disturbance control theory for systems with multiple disturbances: A survey," *ISA transactions*, vol. 53, no. 4, pp. 846–849, 2014.

[2] S. M. Orozco-Soto, A. A. Ortiz-Olvera, and J. M. Ibarra-Zannatha, "Sobre la generación de patrones de caminado de robots humanoides," in *Proc. of the XVIII Congreso Mexicano Robótica*, (Los Cabos, Mexico), pp. 1–8, AMRob, 2015.

[3] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to humanoid robotics*, vol. 101. Springer, 2014.

[4] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1405–1411, IEEE, 1991.

[5] J. Hill and F. Fahimi, "Active disturbance rejection for walking bipedal robots using the acceleration of the upper limbs," *Robotica*, vol. 33, no. 2, pp. 264–281, 2015.

[6] B. Canis, "Unmanned aircraft systems (uas): Commercial outlook for a new industry," *Washington: Congressional Research Service*, 2015.

[7] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521(7553), pp. 460–466, 2015.

[8] El-Sayed and F. Ahmed, *Fundamentals of Aircraft and Rocket Propulsion*. Springer, 2016.

[9] P. M. S. Bouabdallah and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Proc. of Int. Conf. on Robotics and Automation*, pp. 4393–4398, IEEE, 2004.

[10] L. S. L. Li and J. Jin, ""survey of advances in control algorithms of quadrotor unmanned aerial vehicle," in *Proc. of 16th Int. Conf. on Communication Technology (ICCT)*, pp. 107–111, IEEE, 2015.

[11] Y. Li and S.Song, "A survey of control algorithms for quadrotor unmanned helicopter," in *Proc. of 5th Int. Conf. on Advanced Computational Intelligence (ICACI)*, pp. 365–369, IEEE, 2012.

[12] R. L. J. Li and H. Zheng, "Quadrotor modeling and control based on linear active disturbance rejection control," in *Proc. of 35th Chinese Control Conference (CCC)*, pp. 10651–10656, IEEE, 2016.

[13] H. X. Q. Zhanq and Y. Chen, "Active disturbance rejection control for pitch angle of uavs," in *Proc. of 7th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp. 82–86, IEEE, 2015.

[14] Y. Huang and W. Xue, "Active disturbance rejection control: methodology and theoretical analysis," *ISA transactions*, vol. 53, no. 4, pp. 963–976, 2014.

[15] J. Han, "From pid to active disturbance rejection control," *IEEE transactions on Industrial Electronics*, vol. 56(3), pp. 900–906, 2009.

[16] S. M. Orozco-Soto, R. Núñez-Cruz, and J. Ibarra-Zannatha, "Active disturbance rejection control for humanoid stable walking," in *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6, IEEE, 2016.

[17] Q. Zheng and Z. Gao, "On practical applications of active disturbance rejection control," in *Proceedings of the 29th Chinese Control Conference*, pp. 6095–6100, IEEE, 2010.

[18] D. Yoo, S. S.-T. Yau, and Z. Gao, "On convergence of the linear extended state observer," in *2006 IEEE conference on computer aided control system design, 2006 IEEE international conference on control applications, 2006 IEEE international symposium on intelligent control*, pp. 1645–1650, IEEE, 2006.

[19] J.-J. Slotine, J. K. Hedrick, and E. A. Misawa, "On sliding observers for nonlinear systems," in *1986 American Control Conference*, pp. 1794–1800, IEEE, 1986.

[20] H. Sira-Ramírez and V. F. Batlle, "Robust σ–δ modulation-based sliding mode observers for linear systems subject to time polynomial inputs," *International Journal of Systems Science*, vol. 42, no. 4, pp. 621–631, 2011.

[21] H. Sira-Ramírez, C. López-Uribe, and M. Velasco-Villa, "Linear observer-based active disturbance rejection control of the omnidirectional mobile robot," *Asian Journal of Control*, vol. 15, no. 1, pp. 51–63, 2013.

[22] J. Arcos-Legarda, J. Cortes-Romero, and A. Tovar, "Active disturbance rejection control based on generalized proportional integral observer to control a bipedal robot with five degrees of freedom," in *2016 American Control Conference (ACC)*, pp. 3928–3933, IEEE, 2016.

[23] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time variable structure control systems," *IEEE transactions on Industrial Electronics*, vol. 42, no. 2, pp. 117–122, 1995.

[24] W. Tan and C. Fu, "Linear active disturbance-rejection control: analysis and tuning via imc," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2350–2359, 2015.

[25] B.-Z. Guo and Z.-L. Zhao, "On convergence of non-linear extended state observer for multi-input multi-output systems with uncertainty," *IET Control Theory & Applications*, vol. 6, no. 15, pp. 2375–2386, 2012.

[26] K. Bouyarmane and A. Kheddar, "On weight-prioritized multitask control of humanoid robots," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1632–1647, 2017.

[27] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 277–283, IEEE, 2019.

[28] I. Nagesh and C. Edwards, "A multivariable super-twisting sliding mode approach," *Automatica*, vol. 50, no. 3, pp. 984–988, 2014.

[29] J. A. Moreno, "Lyapunov approach for analysis and design of second order sliding mode algorithms," in *Sliding Modes after the first decade of the 21st Century*, pp. 113–149, Springer, 2011.

[30] D. Nenchev, "Differential kinematics," in *Humanoid Robotics: A Reference*, pp. 675–721, Springer, 2019.

[31] S. Feng and Z. Sun, "A simple trajectory generation method for biped walking," in *2008 10th International Conference on Control, Automation, Robotics and Vision*, pp. 2078–2082, IEEE, 2008.

[32] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 294–299, IEEE, 2000.

[33] S.-i. An and D. Lee, "Prioritized inverse kinematics: Desired task trajectories in nonsingular task spaces," *arXiv preprint arXiv:1910.10300*, 2019.

[34] F. Asano, "Stealth walking of 3-link planar underactuated biped," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4118–4124, IEEE, 2017.

[35] C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin, *Bipedal robots: Modeling, design and walking synthesis*. John Wiley & Sons, 2013.

[36] S. Tzafestas, M. Raibert, and C. Tzafestas, "Robust sliding-mode control applied to a 5-link biped robot," *Journal of Intelligent and Robotic Systems*, vol. 15, no. 1, pp. 67–133, 1996.

[37] D. N. Nenchev, A. Konno, and T. Tsujita, *Humanoid Robots: Modeling and Control*. Butterworth-Heinemann, 2018.

[38] R. Kelly and V. Santibáñez, *Control de movimiento de robots manipuladores*. Pearson educación, 2003.

[39] R. Featherstone, "A beginner's guide to 6-d vectors (part 1)," *IEEE robotics & automation magazine*, vol. 17, no. 3, pp. 83–94, 2010.

[40] R. Featherstone, "A beginner's guide to 6-d vectors (part 2)[tutorial]," *IEEE robotics & automation magazine*, vol. 17, no. 4, pp. 88–99, 2010.

[41] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2008.

[42] T. Sugihara, "Contact dynamics," in *Humanoid Robotics: A Reference*, pp. 785–809, Springer, 2019.

[43] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2018.

[44] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[45] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011.

[46] S. M. Orozco-Soto, J. M. Ibarra-Zannatha, A. J. Malo-Tamayo, and A. Cureño-Ramírez, "Active disturbance rejection control for uav pose regulation," in *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6, IEEE, 2017.

[47] S. Zhao and Z. Gao, "Active disturbance rejection control for non-minimum phase systems," in *Proceedings of the 29th Chinese control conference*, pp. 6066–6070, IEEE, 2010.

[48] H. Dallali, M. Brown, and B. Vanderborght, "Using the torso to compensate for non-minimum phase behaviour in zmp bipedal walking," in *Advances in Robotics Research*, pp. 191–202, Springer, 2009.

[49] S. M. Orozco-Soto and J. M. Ibarra-Zannatha, "Motion control of humanoid robots using sliding mode observer-based active disturbance rejection control," in *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, pp. 1–8, IEEE, 2017.

[50] S. M. Orozco-Soto and J. M. Ibarra-Zannatha, "Control con rechazo activo de perturbaciones para el equilibrio de robots humanoides.," *Research in Computing Science*, vol. 135, pp. 159–171, 2017.

[51] R. Marino and P. Tomei, *Nonlinear control design: geometric, adaptive and robust*. Prentice Hall International (UK) Ltd., 1996.

[52] M. W. Spong and H. Sira-Ramirez, "Robust control design techniques for a class of nonlinear systems," in *1986 American Control Conference*, pp. 1515–1522, IEEE, 1986.

[53] J. Jovic, F. Philipp, A. Escande, K. Ayusawa, E. Yoshida, A. Kheddar, and G. Venture, "Identification of dynamics of humanoids: Systematic exciting motion generation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2173–2179, IEEE, 2015.

[54] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[55] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[56] S. García, M. E. López, R. Barea, L. M. Bergasa, A. Gómez, and E. J. Molinos, "Indoor slam for micro aerial vehicles control using monocular camera and sensor fusion," in *2016 international conference on autonomous robot systems and competitions (ICARSC)*, pp. 205–210, IEEE, 2016.