



Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica

Research group MACRO - Mechatronics, Control and Robotics

GAIT AND BALANCE KINEMATIC CONTROL FOR A HUMANOID ROBOT BASED ON DUAL QUATERNION ALGEBRA

Ana Christine de Oliveira

Belo Horizonte, Brazil

2015

Ana Christine de Oliveira

GAIT AND BALANCE KINEMATIC CONTROL FOR A HUMANOID ROBOT BASED ON DUAL QUATERNION ALGEBRA

Thesis submitted to the Graduate Program in Electrical Engineering of Escola de Engenharia at the Universidade Federal de Minas Gerais, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Advisor: Bruno Vilhena Adorno

Belo Horizonte, Brazil

2015

*Para José Antônio
e Maria Dinazarde*

Acknowledgements

First of all, I would like to thank my beloved parents, José Antônio and Maria Dinazarde, who have supported me unconditionally in this journey. I have a great pride and admiration for them, for their strength, kindness and persistence.

To my brother and best friend of a life, José Antônio Jr., who had encouraged me at the very beginning to enter in the engineering world, for being my biggest reference when it comes to creative engineering and persistence.

To Marcelino Almeida, for stimulating me to always foster the best results in all aspects of my life. He has always been an enthusiastic partner and has supported me in every single moment of this journey.

A special thanks to my advisor, or as he says, my academic father, Prof. Bruno Adorno, for accepting me as his student and guiding me in this extraordinary world of Master studies. He has contributed in all aspects of this work, not only with theoretical and practical advices and remarks, but also with lessons that helped me to improve my professional skills, and I am very grateful for that. He has an admirable argumentation ability, and is capable of pointing out many relevant issues invisible for the most part of people, and he will always be a great source of inspiration in my academic life.

I would like to acknowledge Prof. Patricia Pena and Prof. Ricardo Takahashi, my undergraduate advisors, for sharing valuable lessons and thoughts with me, and for encouraging me to pursue the academic life. In particular to Prof. Patrícia, thanks for being my female reference in this mannish engineering world, and for supporting me in my academic decisions.

To the members of MACRO, and also cohabitants of LCR: Alex, Antônio, Daniel, Diana, Eduardo, Ernesto, Frederico, Fredy, Gabriela, Heitor, Juan, Jaime, Laysa, Leandro, Lucas, Marcelo, Mariana, Priscilla, Rafael, Rigoberto, and Stella. Thanks for cheering up my daily life, and for sharing laughs, reflections, and frustrations throughout this time. Special thanks to Ernesto, who patiently and kindly helped me with many theoretical, practical, and sometimes, existential doubts.

I acknowledge Prof. Guilherme Raffo and Prof. Leonardo Torres, for accepting to be reviewers of my thesis, and for their keen remarks on the text.

To the PPGEE faculty members, a special acknowledgment, for their educational

excellence that makes the Graduate Program in Electrical Engineering of UFMG one of the bests of Brazil. Their thorough analysis about engineering issues are a source of inspiration to pursue excellence in our works.

I also would like to thank the PPGEE administrative staff, for providing me the infrastructure that I needed to work, specially to Jerônimo Coelho and Prof. Rodney Saldanha.

“Whether you think you can, or you
think you can’t—you’re right.”

Henry Ford

Resumo

Este trabalho apresenta uma nova metodologia baseada na álgebra de quatérnios duais (QDs) para obter o modelo cinemático de um robô humanóide, e propõe uma estratégia de controle que satisfaz as restrições cinemáticas garantindo uma caminhada estável. O método de modelagem apresentado consiste de três etapas: modelagem dos membros do robô, modelagem do centro de massa e modelagem do comportamento cooperativo das pernas, utilizando o Espaço de Cooperação Dual. As vantagens deste novo método de modelagem são a sua compacidade e menor esforço computacional demandado para calcular o modelo, quando comparado com métodos baseados em matrizes de transformação homogêneas (MTH), uma vez que os QDs têm apenas oito parâmetros, enquanto as MTHs têm doze. Além disso, a multiplicação de QDs é mais barata computacionalmente que a multiplicação de MTHs e os seus coeficientes podem ser diretamente utilizados na lei de controle, ao contrário das MTHs. A estratégia de controle foi projetada utilizando a pseudoinversa da matriz Jacobiana, a qual representa a tarefa da caminhada, com um termo de *feed-forward*, e uma tarefa adicional, projetada para manter os braços próximos de uma configuração desejada, foi incluída no espaço nulo desta matriz Jacobiana. O modelo obtido, bem como o controlador proposto, foram validados utilizando o software de simulação de realidade virtual para sistemas robóticos V-REP. Foi verificado que os dados estimados utilizando o modelo e os valores calculados pelo simulador (considerados como dados medidos) se aproximaram bastante, evidenciando que o método de modelagem fornece informações confiáveis. Além disso, o robô controlado pela estratégia proposta foi capaz de executar diferentes movimentos de caminhada com sucesso, além de ser capaz de manter o equilíbrio até mesmo quando os braços estavam se movendo. Este trabalho é a primeira etapa do Projeto Popeye, cujo objetivo geral é construir uma plataforma de testes para um robô humanóide real.

Abstract

In this work, we present a novel method to obtain the kinematic model for a humanoid robot based on dual quaternion (DQ) algebra, and propose a control strategy that fulfills the kinematic constraints for a balanced gait. The modeling method consists of three stages: the robot's limbs modeling, the center of mass modeling, and the legs cooperative behavior modeling using the Cooperative Dual Task-Space Framework. The advantages of the novel modeling method are its compactness and lower computational effort required to calculate the model, when compared with methods based on homogeneous transformation matrices (HTMs), since DQs has only eight parameters whereas HTMs has twelve. In addition, DQs multiplications are less expensive than HTMs multiplications and its coefficients can be directly used in a control law, differently from HTMs. The presented control strategy was designed using the pseudo-inverse of a Jacobian matrix representing the locomotion task with a velocity feed-forward term, and an additional task, designed to keep the arms close to the desired configuration, was included in the null space of this Jacobian matrix. We validated the obtained model and the proposed controller using the virtual reality simulation software for robotic systems V-REP. The data estimated using the model and the one calculated by the simulation software (which we consider as the measured values) were very close, showing that the modeling method provides reliable information. Furthermore, the robot successfully performed different walking motions when controlled by the proposed strategy, and was capable of keeping the balance even when the arms were moving. This work is the first stage of the Popeye Project, whose goal is to build a test framework for a real humanoid robot.

Contents

List of Figures	xiii
List of Tables	xvii
Acronyms	xix
Notation	xxi
1 Introduction	1
1.1 Historical Background	3
1.2 Objectives	4
1.3 Contributions	5
1.4 Structure of the Text	6
2 State of the Art	7
2.1 Modeling of Biped and Humanoid Robots	7
2.2 Walking Pattern	10
2.3 Control of Gait and Balance	13
2.4 Chapter Overview	16
3 Mathematical Background	17
3.1 Basic Concepts in Dual Quaternion Theory	17
3.1.1 Quaternions	17
3.1.2 Dual Quaternions	19
3.2 Dual Quaternions Representing Rigid Motions	21
3.2.1 Rotations and Translations Represented by Quaternions	21
3.2.2 Rigid Motions Represented by Dual Quaternions	23
3.3 Robots Kinematic Modeling	25
3.3.1 Forward Kinematics Model	25
3.3.2 Differential Forward Kinematics Model	26
3.4 Cooperative Dual Task-Space Framework	28

3.5	Chapter Overview	30
4	Kinematic Modeling	31
4.1	Kinematic Modeling of the Humanoid Robot's Limbs	31
4.2	System's Reference Frame	33
4.3	Kinematic Modeling of the Robot's Center of Mass	34
4.4	Legs Coordinated Behavior Modeling	38
4.5	Chapter Overview	40
5	Gait and Balance Control Strategies	41
5.1	Stability Conditions	41
5.2	Reference Trajectories	42
5.2.1	Walking Pattern Generation Method	42
5.2.2	Feet Trajectories	48
	Circular Trajectory	52
5.3	Control Strategies	54
5.4	Controller Stability Proof	57
5.5	Chapter Overview	59
6	Experiments and Results	61
6.1	Platforms Specification	61
6.1.1	Simulation Environment Specification	61
6.1.2	Robot Specification	63
6.2	Model Validation	64
6.3	Center of Mass Control	68
6.4	Gait and Balance Control	69
6.5	Chapter Overview	88
7	Conclusions and Future Works	91
7.1	Overview	91
7.1.1	Kinematic Modeling Method	91
7.1.2	Gait and Balance Control Strategies	92
7.2	Future works	93
	Bibliography	95
A	DH Parameters of the Robot's Limbs	101
B	Remainder Results	103
B.1	Center of Mass Control Results	103
B.2	Walking Control	104

List of Figures

1.1	Recently developed humanoid robots.	4
1.2	Poppy robot.	5
2.1	3DLIMP related to a humanoid robot.	11
3.1	Example of the projection of a point.	22
3.2	Example of a rigid motion represented by the dual quaternion in Example (3.2).	24
3.3	Cooperative variables.	29
4.1	Robot's limbs frames and coordinate transformations.	33
4.2	Estimated CoM of robot's links.	35
4.3	Joints influence in the CoM $\mathbf{c}_{14}^b$	37
4.4	Cooperative variables of the robot's legs.	38
5.1	Support polygon of a humanoid robot.	41
5.2	Cart-table model.	43
5.3	Results of the walking pattern generation method.	47
5.4	Comparison of the ZMP Resultant for different values of Q_x	47
5.5	Example of a cubic Bézier curve.	48
5.6	Pulse function example.	49
5.7	Trajectory generated by (5.11).	50
5.8	Complete position trajectories of the feet generated by (5.12).	51
5.9	Linear velocities of the feet generated by (5.13).	51
5.10	Circular Path of Example (5.4).	54
6.1	Structure of the simulation framework.	63
6.2	Humanoid robot ASTI.	63
6.3	DH convention for ASTI.	64
6.4	Robot configuration during the validation.	65
6.5	CoM and CoM linear velocity validation.	66
6.6	Poses of the limbs end-effectors during validation.	66
6.7	Linear and angular velocities of the limbs end-effectors during validation.	67

6.8	Results obtained from the execution of a circular trajectory.	69
6.9	Results obtained from the execution of the standing up movement.. . . .	70
6.10	Simulation snapshots of the standing up movement.	71
6.11	Results obtained from the execution of the second pattern controlled using the strategy 1.	72
6.12	Feet trajectories obtained from the execution of the second pattern controlled using the strategy 1.	73
6.13	Simulation snapshots from the execution of the second pattern controlled using the strategy 1.	74
6.14	Comparison of the results obtained from the execution of the first pattern controlled using strategies 1 and 2.	75
6.15	Results obtained from the execution of a walking motion controlled using the strategy 2, where the arms were actuated.	77
6.16	Simulation snapshots of a walking motion controlled by the strategy 2, while moving the arms.	78
6.17	Angles of the arms joints during the walking motion controlled using the strategy 3, where the arms were actuated.	79
6.18	Results obtained from the execution of a walking motion of 50 footsteps, controlled using the strategy 3, which includes the arms in the control law.	80
6.19	Simulation snapshots of a walking motion of 50 footsteps, controlled using the strategy 3, which includes the arms in the control law.	81
6.20	Arms joints angles during a walking motion of 50 footsteps, controlled using the strategy 3, which includes the arms in the control law.	82
6.21	Results obtained from the execution of a walking motion of 50 footsteps, controlled using the strategy 4.	83
6.22	Angles of the arms' joints during a walking motion of 50 footsteps, controlled using the strategy 4.	84
6.23	Results obtained from the execution of a walking motion controlled using the strategy 4, where the arms were actuated.	84
6.24	Angles of the arms joints during the walking motion controlled using the strategy 4.	85
6.25	Simulation snapshots of a walking motion controlled by the strategy 4, while moving arms.	86
6.26	Results obtained from the execution of a walking motion in a semi-circle shaped path, controlled using the strategy 4.	87
6.27	Footprints executed in a walking motion in a semi-circle shaped path, controlled using the strategy 4.	88
A.1	Right arm scheme.	101

A.2	Left arm scheme.	101
A.3	Left and right leg scheme.	102
B.1	Results obtained from the execution of a trajectory defined by a sine in the xy plane.	103
B.2	Results obtained from the execution of a trajectory defined by a sine in the xz plane.	104
B.3	Comparison of the results obtained from the execution of the second pattern controlled using strategies 1 and 2.	105
B.4	Comparison of the results obtained from the execution of the third pattern controlled using strategies 1 and 2.	106
B.5	Comparison of the results obtained from the execution of the forth pattern controlled using strategies 1 and 2.	107
B.6	Comparison of the results obtained from the execution of the first pattern controlled using strategies 2 and 4.	108
B.7	Comparison of the results obtained from the execution of the second pattern controlled using strategies 2 and 4.	109
B.8	Comparison of the results obtained from the execution of the third pattern controlled using strategies 2 and 4.	110
B.9	Comparison of the results obtained from the execution of the forth pattern controlled using strategies 2 and 4.	111

List of Tables

6.1	Robot's links masses.	64
6.2	Summary of the proposed control strategies.	72
6.3	Index comparison between strategies 1 and 2.	76
6.4	Control Strategy 4 comparison indexes.	82
A.1	DH Parameters of right arm.	101
A.2	DH Parameters of left arm.	101
A.3	DH Parameters of left and right legs.	102

Acronyms

3D	Three-Dimensional
3DLIPM	Three-Dimensional Linear Inverted Pendulum Mode
3MLIMP	Three-Mass Linear Inverted Pendulum Mode
AIST	National Institute of Advanced Industrial Science and Technology
API	Application Programming Interface
ASIMO	Advanced Step in Innovative Mobility
BIPMAN	Bipedal Walking Machine
CDTS	Cooperative Dual Task Space
CoM	Center of Mass
DFKM	Differential Forward Kinematics Model
DH	Denavit-Hartenberg
DOF	Degrees of Freedom
DQ	Dual quaternion
DSP	Double-Support Phase
FKM	Forward Kinematics Model
GCIPM	Gravity-Compensated Inverted Pendulum Mode
GCoM	Center of Mass Projection on the Ground
HRP	Humanoid Robotics Project
HTM	Homogeneous Transformation Matrix
IK	Inverse Kinematics

IAVU	Integrated Absolute Variation of the Control signal
MACRO	Mechatronics, Control, and Robotics
MAE	Mean Absolute Error
PID	Proportional Integral Derivative
ROS	Robot Operating System
SESC	Statically Equivalent Serial Chain
SSP	Single-Support Phase
SVD	Singular Value Decomposition
UFMG	Universidade Federal de Minas Gerais
WL-3	WASEDA LEG-3
WL-10R	WASEDA LEG-10 REFINED
WL-12	WASEDA LEG-12
ZMP	Zero Moment Point

Notation

Γ	Function representing the trajectory of one foot.
ε	Dual unit.
λ	Scalar gain of the control law.
μ	Step function.
ϕ	Rotation angle.
$\Pi_{\Delta t}$	Pulse function with time duration of Δt .
Ψ_x, Ψ_y	Performance indexes of the preview control method.
B_i^n	Bernstein polynomial of order n .
C_8, C_4	Conjugating matrices.
d_h	Step height.
$\underline{d}$	Dual number.
$\mathcal{D}$	Operator to extract the dual part of dual quaternions.
e	Error between the reference and actual values in the control law.
$\mathcal{F}$	Coordinate system or frame.
$\mathcal{H}$	Set of dual quaternions.
$\mathbb{H}$	Set of quaternions.
$\overset{+}{\mathbf{H}}_4, \bar{\mathbf{H}}_4$	Hamilton operators.
$\overset{+}{\mathbf{H}}, \bar{\mathbf{H}}$	Hamilton operators extended for dual quaternions.
I	Identity matrix.
i_s	Step index.

$\hat{i}, \hat{j}, \hat{k}$	Quaternion (or imaginary) units.
L, M, N, X	Matrices or vectors.
N_s	Number of steps.
J	Jacobian matrix.
J^+	Pseudo-inverse of the Jacobian matrix.
$\hat{J}$	Extended Jacobian matrix.
$\bar{J}$	Jacobian matrix obtained by removing the last row of J (associated with z -axis).
J^A	Jacobian matrix with respect to frame $\mathcal{F}_A$.
$J_{\mathcal{P}}, J_{\mathcal{D}}$	Jacobian matrices related with the primary and dual parts of J .
J_{ori}, J_{pos}	Orientation and position Jacobian matrices.
J_r, J_a	Cooperative Jacobian matrices (relative and absolute).
$\mathbf{n}$	Pure quaternion representing the rotation axis.
$\mathcal{P}$	Operator to extract the primary part of dual quaternions.
$\mathbf{p}_{AB}^A$	Quaternion representing the translation from the frame $\mathcal{F}_A$ to $\mathcal{F}_B$ with respect to $\mathcal{F}_A$.
$\mathbf{p}^A$	Quaternion representing a point with respect to a frame $\mathcal{F}_A$.
q	Joints vector.
$\mathbf{r}_B^A$	Unit quaternion representing a rotation from frame $\mathcal{F}_A$ to $\mathcal{F}_B$ (moving frames notation).
T	Sampling time.
$\mathbf{h}, \mathbf{x}, \mathbf{y}$	Quaternions.
$\underline{\mathbf{h}}, \underline{\mathbf{x}}, \underline{\mathbf{y}}$	Dual quaternions.
$\underline{\mathbf{x}}^*$	Dual quaternion conjugate.
$\underline{\mathbf{x}}^{\{\lambda\}}$	Dual quaternion exponentiation.
$\ \underline{\mathbf{x}}\ $	Dual quaternion norm.
$\underline{\mathbf{x}}_A^B$	Complete pose of a frame $\mathcal{F}_A$ with respect to $\mathcal{F}_B$.

$\underline{\boldsymbol{x}}_r, \underline{\boldsymbol{x}}_a$	Cooperative variables (relative and absolute).
$\text{vec}_4, \text{vec}_8$	Vec operators representing a one-to-one mapping from $\mathbb{H}$ to $\mathbb{R}^4$ or $\mathbb{H}$ to $\mathbb{R}^8$, respectively.
w_L	Preview time window.

1

Introduction

Biped robots are a particular case of legged robots, whose locomotion is achieved by the coordination between two legs. They are more versatile when compared with conventional wheeled robots, since they have higher mobility in human environments thanks to its peculiar characteristic of discontinuous contact with the ground, which allows the locomotion in rough terrains, stairs climbing, and obstacles avoidance. However, biped robots tends to easily tip over in real environments. Therefore, it is necessary to develop a controller that meets all constraints required for a balanced walk.

Humanoid robots, or simply humanoids, are a class of biped robots whose body shape resembles the human body, i.e. with two arms, two legs, one torso, and one head. This type of robots arouses interest on the scientific community since the 1970s, mainly due its similarity with the human body, which makes it easier to interact with human tools and environments, such as door pullers, stairs, automobiles, valves, etc. Humanoids can be used to access and handle tools in harmful and dangerous zones, like in the case of nuclear accidents or buildings with compromised structures (Siciliano & Khatib, 2008).

A curious fact about this type of robots comes from a work published by Mori et al. (2012), where they define the “uncanny valley.” This work shows how human beings react in the presence of human-like entities, and one conclusion is that human beings are familiar with humanoid robots almost as they are with themselves. Thus, another interesting application of humanoids is in assistive tasks, as carrying and supporting elderly or physically disabled people, or in treatments of psychological disorders such as autism.

According to Fukuda et al. (2012), robots locomotion can be classified into two

categories: quasi-static and dynamic locomotion. As defined by Mason (1985, apud Siciliano et al. 2008), in quasi-static robot systems, the effects generated by accelerations and inertial forces are negligible, and, in this case, the system is modeled as a transition between discrete configurations. Furthermore, in this case, it is assumed that the robot is statically stable, i.e. if the locomotion is interrupted the robot keeps a stable posture. The first approach can be used to achieve slow walking-speeds, and it is possible to be performed only on flat surfaces. The second approach allows a better performance, achieving faster walking-speeds, including running, and a flat surface is not required in this case. Thus walking on uneven or rough terrains, as well as stairs climbing, is only possible assuming a dynamic locomotion.

The autonomous locomotion of humanoids is a very challenging subject, and depending on the desired behavior and application, the dynamic or the kinematic approach can be adopted in its modeling and control. This class of robots has a large number of Degrees of Freedom (DOF), and obtaining a dynamic model in this case requires extensive calculations. Furthermore, the dynamic model of this kind of structure can eventually change when interacting with the environment, and the controller must be designed to be robust or adapt to these situations. On the other hand, the kinematic modeling does not suffer from these problems, however, this approach has a limitation regarding the locomotion velocity and the environment structure, since it does not allow fast locomotion and walking on rough terrains. Moreover, in this case, the model does not take into consideration the forces related to the movement.

This work focuses on the kinematic approach to model and control a humanoid robot, and we adopt a representation based on dual quaternion (DQ) algebra. The most commonly used representation within the context of kinematics, robotics, and control systems are the homogeneous transformation matrices (HTMs), since they are a singularity-free representation and can be used to express rigid motions. However, in the last two decades, DQs have gained popularity in this context, because they also represent rigid motions without suffering from representational singularities, and are more compact than HTMs, since the former have only eight parameters whereas the latter have twelve. Moreover, DQ multiplications are less expensive than HTMs multiplications, and its coefficients can be directly used in a control law, which, to the best of our knowledge, has not been done in the case of HTMs (Adorno, 2011).

The remainder of this chapter is organized as follows: Section 1.1 presents an overview about the remarkable events on humanoid robots history; Section 1.2 explains the objectives of this work; Section 1.3 lists its contributions; and Section 1.4 details the structure of this thesis.

1.1 Historical Background

The first known record about humanoid robots was found in Leonardo Da Vinci's notebook, possibly conceived around 1495 and rediscovered in the 1950s by Carlos Pedretti, a professor of the University of California (Rosheim, 2006). The designed mechanism—called “Robot Knight”—was composed of a series of pulleys and cables, and could stand, sit, raise its visor, and maneuver its arms, independently.

The first biped robots with actuated joints were developed by Ichiro Kato and his colleagues from the Department of Mechanical Engineering of the School of Science and Engineering of the University of Waseda in Tokyo. In 1968, they built the lower-limbs mechanical model WASEDA LEG-3 (WL-3), which was electro-hydraulic actuated and was capable to perform a human-like gait, stand and sit. One year later, they designed a biped robot with an anthropomorphic appearance that was pneumatically activated, called WAP-1. The first human-body sized robot was built in 1973, named WABOT-1, which had a limb-control system, an image processing unit and an embedded speech and interaction system. In addition to walking locomotion, this robot was able to carry and manipulate objects with its hands.

The robot WASEDA LEG-10 REFINED (WL-10R) was presented in 1982 by A. Takanishi, who was a collaborator of Ichiro Kato, and his colleagues. It represented a landmark in humanoid robots history as the first robot to execute dynamic locomotion. In 1984, some researchers from the University of Tokyo developed the BIPER Series, which were all statically unstable, but could execute dynamic locomotion. BIPER 1 and 2 were capable to walk only in the sideway direction whereas BIPER 3 could also walk in forward and backward directions. BIPER 3 has only a point contact between the leg and the ground; thus, to stay upright, the robot must keep executing successive steps, continuously.

R. Katoh and M. Mori built in 1984 the BIPMAN (Bipedal Walking Machine) robot, which was a new paradigm in the context of biped robots, since it was constituted by two telescopic legs without knees that could extend and contract to perform the locomotion. In 1985, Jessica Hodgins and Marc Raibert developed a biped robot hydraulically actuated that was able to jump up to 67 cm and run with a speed up to 4 m/s.

Ishiro Kato and his coworkers once again presented an innovation in 1986 with the robot WASEDA LEG-12 (WL-12), which was the first to perform dynamic walking with trunk compensation. In 1990, Tad McGeer developed the first 1 DOF biped robot with articulated legs and no actuation, capable to perform locomotion in a downhill slope, so called “passive walking,” generated by the interaction between gravity and inertia.

After two decades of development and research, the company Honda launched in 2005 the robot ASIMO (Advanced Step in Innovative Mobility), which can run, walk on uneven terrains, smoothly turn walking direction, climb stairs, and grasp objects. Furthermore, it has an advanced system of interaction, allowing it to respond to voice commands, and

also recognize faces and map environments. It is one of the most complete humanoid robot available in the market. Along with ASIMO, some recently presented humanoids are the NAO by AldebaranTM, the Atlas by Boston DynamicsTM, and the HRP (Humanoid Robotics Project) series developed by the National Institute of Advanced Industrial Science and Technology (AIST), like the HRP-4C. These robots are all depicted in Figure 1.1.

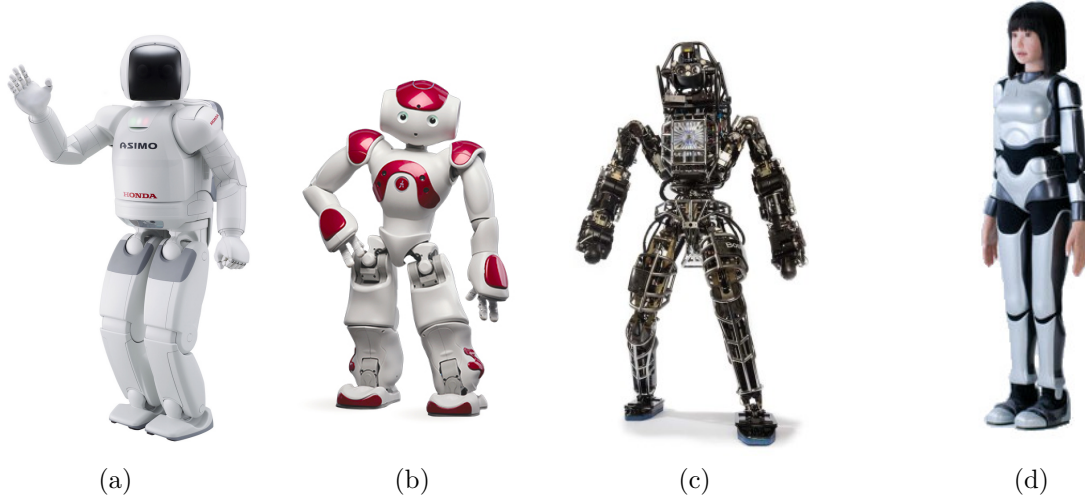


Figure 1.1: Recently developed humanoid robots. From left to right: (1.1a) ASIMO robot¹, (1.1b) NAO robot², (1.1c) Atlas robot³, and (1.1d) HRP-4C robot⁴.

For a more comprehensive survey on the humanoid robots history, see André et al. (2004).

1.2 Objectives

This work was developed within the research group MACRO (Mechatronics, Control, and Robotics) and is part of the Popeye Project, whose main goal is to build at UFMG a humanoid robot for the study of whole-body control techniques and human-robot interaction. This humanoid robot, which is depicted in Figure 1.2, is a result of an initiative of the INRIA Institute, called Poppy Project. It consists of an open-source platform for the creation and assembly of the robot's 3D printed parts, and offers libraries to control the robot's joints and behavior.

The work herein presented is a conceptual part in the development of Popeye Project, focusing on the locomotion, which is a fundamental part to implement a testbed for the humanoid robot. Its objectives are listed below:

¹Source: <http://asimo.honda.com>.

²Source: <https://www.aldebaran.com>.

³Source: <http://www.bostondynamics.com>.

⁴Source: <http://www.aist.go.jp>.

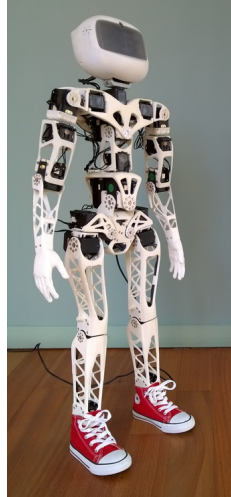


Figure 1.2: Poppy robot⁵.

1. Development of a kinematic model for the whole-body and Center of Mass, for a generic humanoid robot;
2. Development of a gait and balance controller for the robot, based on the work of Park & Lee (2013); Adorno et al. (2010), using the DQ algebra and the Cooperative Dual Task-Space Framework.

1.3 Contributions

Within the scope of UFMG, the contribution of this work relies on the replication and validation of the results found in the literature, using a virtual reality simulation environment. This is an important stage for a better understanding of the consolidated works about humanoid robots, which is fundamental to build a reliable testbed for a humanoid robot.

The scientific contribution of this work consists in the adoption of the DQ algebra in the humanoid's kinematic model conception, since it was not found in the related literature. Moreover, we proposed enhancements in the original control strategy by Park & Lee (2013) so as to reduce the control effort and also to allow wide range arms movements during the walking motion, still keeping the robot balanced.

Some partial results of this work were published in the XII SBAI - *Simpósio Brasileiro de Automação Inteligente* (in English: Brazilian Symposium of Intelligent Automation) (Oliveira & Adorno, 2015).

⁵Source: <https://www.poppy-project.org>.

1.4 Structure of the Text

This thesis is organized into seven chapters, summarized as follows:

Chapter 2 presents some of the most important works on humanoid robots, comprehending modeling methods, walking pattern generators, and gait controllers.

Chapter 3 reviews the mathematical background needed to understand the presented methods and establishes the notation used throughout this dissertation.

Chapter 4 describes the methodology to obtain the humanoid's kinematic model using DQ algebra.

Chapter 5 summarizes the stability conditions for a balanced locomotion, and details the methods to obtain the reference trajectories for the control. Furthermore, the control strategy adopted is presented and some improvements on the techniques present in the literature are proposed.

In Chapter 6, the experimental platforms are presented, the performed trials are described, and the corresponding results are analyzed.

Chapter 7 presents an overview of the entire work and some steps and improvements for future works are proposed.

Finally, Appendix A contains the Denavit Hartenberg parameters of the robot's limbs, and Appendix B presents all results not shown in Chapter 6.

2

State of the Art

This chapter discusses the related works recently developed within the biped robots field, and is organized as follows: Section 2.1 highlights the existing modeling methods and conventions applied specifically to biped robots; Section 2.2 presents the most consolidated walking pattern generation methods; and Section 2.3 discusses some control strategies for gait and balance of biped robots .

2.1 Modeling of Biped and Humanoid Robots

The Denavit-Hartenberg (DH) convention (Hartenberg & Denavit, 1955) is widely adopted in robots kinematic modeling. This convention is used to define frames to each joint, such that the geometric parameters of a robot can be easily used to obtain the robot's Forward Kinematics Model (FKM). The coordinate transformation between the frames of two consecutive joints is represented by a product of four basic transformations, using the DH parameters (see Section 3.3.1). This convention is frequently adopted to obtain the FKM of biped robots' limbs (Kofinas et al., 2014; Ali et al., 2010; Chevallereau et al., 2010; Zannatha & Limon, 2009), and is also used in the work herein described. Toscano et al. (2014) propose a different methodology based on the Screw Theory to obtain the robot's limbs FKMs, by using the concept called "screw displacements" (a rigid motion can be regarded as a combination of a rotation around an axis and a translation along the same axis) to define the relations between the joints of the kinematic chain. This is a methodology concurrent to the DQ algebra, used herein. Since our research group focuses

on DQ algebra, the adoption of a concurrent methodology becomes inconvenient, because it makes difficult to integrate this work with the rest of the ones developed within the group. Furthermore, the DQ representation, as used in our research group, seems more appropriate for integration with low level kinematic controllers.

According to Sentis (2007), a humanoid robot can be regarded as a free-floating system which has a specified number of joints and a base frame describing its position and orientation. He suggests a virtual spherical joint in series with three virtual prismatic joints to represent the robot’s kinematics, and treats the robot as a holonomic system with n actuated joints and six passive DOFs, where ground reaction forces appear at the contact points with the floor. Chevallereau et al. (2010) extend this interpretation and present two methodologies to obtain the generalized coordinates of the robot: the “Kinematically Free Mode,” which also considers the robot as a free-floating system with the base at the hip center, and the “Rooted Kinematic Chain” mode, which considers the existence of a link in contact with the ground where the system’s base is located, which stays stationary during the whole step-motion. In both approaches, some dynamic constraints are associated with the kinematic model, which specify the conditions for a stable locomotion. The first approach is not interesting in the particular case of this work, since it allows the supporting-foot to move during the step motion whereas it is desired to keep it stationary when it comes to define the system’s reference frame.

In the majority of cases, the robot is controlled in the task space, and the joints configuration must be determined from a desired pose of the robot’s end-effectors. This is the inverse of the FKM, and is known as the Inverse Kinematics (IK) problem. Within the context of biped robots, this problem is usually solved by analytical (Zannatha & Limon, 2009; Tolani et al., 2000; Ali et al., 2010; Kofinas et al., 2014) or numerical (Tevatia & Schaal, 2000) means. In analytical methods, the IK is solved by geometry, relating the robot’s generalized coordinates with its geometric features. For instance, Kofinas et al. (2014) solves the IK problem for a specific robot, while Ali et al. (2010) find a closed-form that can be used for humanoids with a specific structure, and they take advantage of this structure in order to simplify the geometry. This method is interesting to determine the IK of specific robots, and since this work has the objective of determining a general solution, it is not applicable in our case. On the other hand, numerical methods use the system’s Jacobian matrix—which relates velocities at the robot’s end-effectors with the joints velocities—to obtain the solution of the IK problem. Tevatia & Schaal (2000), for example, use the pseudo-inverse of the Jacobian matrix together with the desired velocities of the end-effectors, to obtain the solution. Another approach is called “Control-theory based”, which consists of casting the differential kinematic model of the system into a control problem, that is solved also using the pseudo-inverse of the Jacobian matrix. This method is addressed by Sciavicco et al. (2000), and is used in Mansard et al. (2009).

In the case of biped robots, the Center of Mass (CoM) is an important feature, because

it is related with the robot's balance. Plenty of works propose modeling methods relating the robot's CoM with respect to its joints configuration (Choi et al., 2007; Cotton et al., 2009; Boulic et al., 1995; Phillips & Badler, 1991). The method proposed by Choi et al. (2007) consists in building an equivalent FKM for the robot's CoM, and its expression is defined as a weighted sum of the FKMs of the CoM of each link. To obtain the FKM of each link CoM, it is regarded as the end-effector of the corresponding limb, and the links after this one are not considered. In order to solve the IK of the robot's CoM, a Jacobian matrix is defined by taking the first time derivative, with respect to the joints vector, of the expression representing the FKM. Boulic et al. (1995) and Cotton et al. (2009) use equivalent chains to represent the robot's CoM. Boulic et al. (1995) propose the "Augmented Body" associated with a joint, which consists of a rigid body dynamically equivalent to the union of all bodies supported by the joint, for the current state of the system. Likewise, Cotton et al. (2009), defined a concept named "Statically Equivalent Serial Chain (SESC)," which consists of an equivalent chain whose origin is placed at the link in contact with the ground, and whose extremity is coincident with the CoM. It is a one-to-one mapping between the original chain's joints and the equivalent chain's joints. The aim of that work is to build a CoM Jacobian matrix for a biped robot without its dynamic parameters, which is used to solve the IK problem for the robot's CoM. To do that, the configuration of the joints for some stable postures of the robot are recorded and used in a system of linear equations to obtain the parameters of the CoM Jacobian. These methods are very useful when one foot must be stationary on the ground, but they become inconvenient for robot's in locomotion, when the rooted link changes periodically. Phillips & Badler (1991) use a different approach, where they define the lower part of the robot's torso as the estimated CoM of the robot, and constraint variables associated with the ankle, the knee, and the hip joints of the rooted leg are defined. If the CoM approximation is different from the expected, the constraints must be solved again to guarantee a balanced posture. This approach gives a good approximation of the robot's CoM in the case of robots with well-distributed masses among their limbs. However, if the legs are heavier than the upper-body, which frequently happens, the method becomes inefficient, since it will give a poor approximation of the CoM, and, consequently, increasing the computational effort to satisfy the constraints and adjust the posture.

In this work, we adopt the Rooted Kinematic Chain interpretation to build the whole-body kinematic model of the biped robot, using the DH convention to define the frames of the robot's joints. The IK problem is solved using the Control-theory based method aforementioned. Finally, to obtain the CoM kinematic model we adopt a methodology similar to the used by Choi et al. (2007), i.e. to build an equivalent FKM for the robot's CoM, defined as a weighted sum of the FKMs of the CoM of each link.

In all of the aforementioned works, Homogeneous Transformation Matrices (HTMs) are used to represent transformations between frames. In the work presented in this

dissertation, all transformations are represented by dual quaternions. In the context of biped robots, this approach was only found in Park & Lee (2013), which is focused in the control strategy for the locomotion task, and is better detailed in Section 2.3.

2.2 Walking Pattern

The gait of biped robots is one of the most challenging and exciting fields in robotics. The walking motion is inherently an unstable movement, thus, in order to keep the robot balanced, some kinematic and dynamic constraints must be fulfilled during the motion. The gait is a sequence of coordinate and periodic movements, and its complete cycle is composed of two phases:

- Single-support phase (SSP), when one foot is stationary on the ground, while the other foot is swinging towards the next footprint;
- Double-support phase (DSP), when both feet are in contact with the ground.

In the last decades, some methods were proposed to generate the feet trajectories and the hip trajectory for the gait cycle which fulfills the balance constraints, commonly denoted as “walking pattern.” As long as these trajectories are known, the joints configurations to execute the walking motion can be determined.

The Zero Moment Point (ZMP) was introduced by Vukobratović et al. (1970) and is defined as a point where the moments around the x -axis and the y -axis generated by reaction forces and reaction torques are zero. It is a commonly used parameter when it comes to analyze the stability of the robot in dynamic locomotion. The Three-Dimensional Linear Inverted Pendulum Mode (3DLIPM) and its variations are widely used in the literature to generate walking patterns for biped robots in flat terrains (Kajita et al., 2001; Tang & Er, 2008; Feng, 2008), and consists in a dynamic model that relates the robot’s ZMP to its CoM. To summarize, the dynamic behavior of the robot during the SSP is approximated by the dynamics of an inverted pendulum, consisting of a telescopic rod with a point mass whose motion is constrained by a plane. The base of the inverted pendulum is located at the ZMP, and the point mass represents the robot’s global CoM. The 3DLIPM related to a humanoid robot is illustrated in Figure 2.1.

The ZMP is assumed to be close to the foot center, and the ZMP trajectory is defined by the desired footprints, which are determined by a periodic function regarding the desired walking speed, and the time duration of the SSP and the DSP. Thus, the CoM trajectory is determined from the ZMP trajectory using the dynamic equations of the system.

This method has a limitation regarding the robot stability in rough terrains with a very restricted area for the footprints. In this case, the footprints must be specified

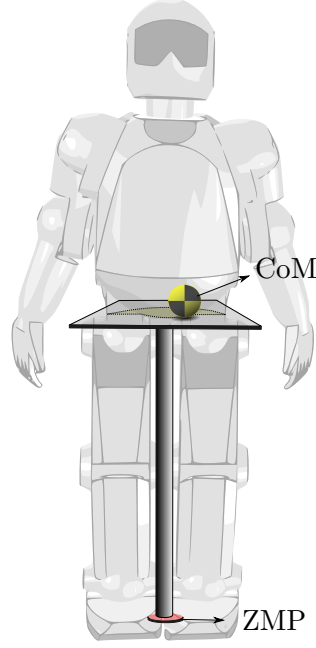


Figure 2.1: 3DLIMP related to a humanoid robot.

according to the terrain conditions. Consequently, the ZMP trajectory is not defined by a known function, and the aforementioned method using the 3DLIPM is not appropriate to obtain the CoM trajectory. Aiming to solve this issue, Kajita et al. (2003) and Kim (2007) proposed methods to determine the CoM trajectory from arbitrary trajectories of the ZMP, also based on 3DLIPM. Kajita et al. (2003) proposed a method based on preview control¹, where the current input of the system, which is the current CoM reference, is calculated using future references of the known ZMP trajectory. They presented a discrete state space system that models the dynamics of the 3DLIPM, and used the associated matrices to calculate the preview control gains, and to determine the CoM reference. Kim (2007), on the other hand, proposed a convolution sum method to find the solution, where the 3DLIPM is written in the frequency domain. Its impulse response is determined, which is non-causal, but stable, and the current CoM reference can be calculated through a discrete convolution between the ZMP trajectory and the system's impulse response, within a finite time window. The convolution result consists of two terms: the first one represents the CoM computed from past information about the ZMP trajectory, and the second term represents the CoM computed from future information about the ZMP trajectory. Adding these two terms, the result is the current CoM reference.

The method proposed by Kajita et al. (2003) is widely used in the context of walking pattern generation for biped robots on account of its efficiency and flexibility. However, the robot's global CoM jerk can reach large values in the presence of disturbances, compromising the robot's stability. With the objective of solving this issue, Wieber (2006)

¹Preview control is one of the precursors theories of the predictive control theory, and this is the terminology defined in the mentioned work.

proposed an extension of this method, including some constraints in the minimization of the performance index, which is calculated as part of the preview control method. These constraints determine a limit for the CoM jerk and a safe region inside the support polygon where the ZMP is allowed to stay. The drawback of this method is the large computational effort required to solve the constrained optimization problem at each time instant.

Furthermore, Kajita et al. (2003) assumes that the robot's mass is concentrated at the CoM, and that the free-leg dynamics can be neglected during the SSP. In reality, the leg mass is a large proportion of the robot's body mass. Consequently, during the SSP, the ZMP moves away from the predefined reference, decreasing the robot's stability. With the objective of obtaining a more stable controller, Park & Kim (1998) proposed a more accurate model named Gravity-Compensated Inverted Pendulum Mode (GCIPM), which is based on the linear inverted pendulum mode, including the dynamics of the free-leg. The robot is modeled by a two-mass system: one representing the free-leg, located at the swinging-foot, and one representing the rest of the robot's body, concentrated at the hip. The proposed method assumes a defined trajectory for the swinging-foot, so that the influence of the free-leg mass in the robot's dynamics can be computed and compensated for by the hip motion.

Another model similar to the 3DLIMP also applied in the literature to generate walking patterns is the Three-Mass Linear Inverted Pendulum Model (3MLIPM) (Galdeano et al., 2013; Feng, 2008). The robot's dynamics is approximated by the 3MLIPM, which consists in a three-link system with a point mass in each link—one representing the torso, and the two others representing the legs—, being a more accurate model than the single mass linear inverted pendulum model. The method proposed by Feng (2008) is similar to the aforementioned ones, where the footprints are determined for a flat terrain, and the trajectory of the masses is defined according to the dynamic equations of the system. Galdeano et al. (2013) presents an extension of this method, where an optimization of the joints trajectory is made with the objective of minimizing the ZMP movement inside the footprint area to improve the robot's stability. The contribution of this extension is the possibility of generating walking patterns with direction changes, which is not possible in the work of Feng (2008).

Interpolation is another method often used in walking pattern generation (Huang et al., 2001; Kajita et al., 1999). According to this method, firstly some kinematic constraints are imposed to the feet, for instance the time duration of the gait cycle, the time duration of the DSP, the position of the highest point of the swinging-foot, and the floor conditions. From these constraints, a third-order periodic spline interpolation is made to generate the feet trajectories. Thus, a series of hip trajectories are generated also using third-order periodic spline interpolation, by varying the distance between the hip and the supporting-foot at the start and at the end of the step motion. Thereafter, the trajectory which maximizes the stability margin, which is defined as the distance between the ZMP and the

support polygon edges, is chosen. Roussel et al. (1998) also presents a method to minimize the energy of the gait cycle, where a dynamic model for each phase of the gait cycle is formulated and a cost function representing the energy injected in the system during the cycle is presented. The aim is to minimize this cost function, given some constraints: the initial and final desired configuration of the robot's joints in the SSP, the final desired configuration of the robot's joints in the DSP, the time duration of the SSP, and the time duration of the whole gait cycle. In a non-periodic walking motion, these method becomes inefficient, since the optimization problem must be solved for each step.

In contrast with the presented ideas, Zarrugh & Radcliffe (1979) propose a walking pattern generation method in the joints level so that the walking motion mimics the human gait. In order to accomplish that, kinematic data is recorded from a human gait, and the displacements of the joints are determined to formulate the walking pattern. This work gives some interesting insights about the walking motion in the robotics field, however it is not very versatile since it does not allow to change the walking directions and to control the locomotion itself.

The choice of the walking pattern generation method is a trade-off between model accuracy and footsteps placement flexibility. In this work, the walking pattern generation method is based on the 3DLIPM, because it is more flexible than others, and allows us to deal with robots whose dynamic parameters are not known, or uncertain. The adopted method is based on the work proposed by Kajita et al. (2003), which is more consolidated in the robotics field, and requires a low computational effort, if compared with the aforementioned works.

2.3 Control of Gait and Balance

Since the feet and CoM reference trajectories are determined by the walking pattern generator, the displacements in the robot's joints space can be calculated by solving the IK problem. However, the humanoid robot tends to tip over easily in real environments, specially under disturbances. Therefore, the implementation of a balance control is of a particular relevance in locomotion.

Most of the existing methods to stabilize the robot's posture are based on the ZMP criterion for dynamic locomotion (Sugihara et al., 2002; Galdeano et al., 2014; Yokoi et al., 2004; Hirai et al., 1998; Li et al., 2012). Sugihara et al. (2002) propose a controller based on the 3DLIMP, which leads the actual ZMP to the desired position given by the walking pattern generator. The controller modifies the robot's CoM using the pseudo-inverse of its Jacobian matrix. The controller presented by Galdeano et al. (2014) also attempts to lead the actual ZMP to the reference. In this strategy, the ZMP error compensation is calculated using a non-linear controller, and projected on a virtual sphere; the result is then added to the CoM reference. This method was implemented in a robot with an

embedded control framework for the CoM, and it allowed the stabilization of the robot's body on a ground with varying slope. A more complex method is proposed by Hirai et al. (1998), and also presented by Yokoi et al. (2004), by using a set of complementary strategies to formulate the balance controller. One of these strategies—denoted as “Body Inclination Control” or “Ground Reaction Force Control”—consists in modifying the feet position/orientation in order to correct the body's posture. The other one—denoted as “ZMP Damping Control” or “Model ZMP Control”—attempts to shift the desired ZMP by adjusting the horizontal position of the robot's torso. Finally, the “Foot Landing Position Control” or “Foot Adjusting Control” has the aim of correcting the relative position of the upper-body and the feet, and can be used to stabilize the robot's body on uneven terrains, where the feet may be inclined. Li et al. (2012) also developed a controller based on a set of different strategies; however, instead of controlling stiff humanoid systems, the intention is to stabilize compliant robots. The three strategies used are: the “Horizontal Compliance Control,” which is used to make the body's compliance regulation by using a PID controller based on the ZMP error to modify the robot's CoM; the “Body Attitude Control” strategy is used to compensate the angular momentum generated by the lower-limbs motion, by rotating the upper-body in the inverse direction; the “Potential Energy Control” stabilizes the robot's potential energy by constraining the CoM to an inclined plane. The drawback of ZMP based control methods is the need of sensors to measure it, or, in the case of using mathematical relations to estimate it, the large estimation errors.

For static locomotion, a common approach for stabilizing the robot's posture is to use control strategies based on kinematic-constraints, in which the ZMP information is not taken in consideration, but the CoM projection on the ground (GCoM). Following this idea, some works propose the adjustment of some walking pattern parameters together with the IK to keep the robot balanced during the locomotion (Hof, 2008; Missura & Behnke, 2011; Graf, 2010). Hof (2008) presents a new concept named “Extrapolated Center of Mass (XcoM),” which combines the robot's CoM projection on the ground and its CoM velocity and simplifies the stability conditions for locomotion. In this method, a feedback controller compensates the foot placements and CoM errors by adjusting the step length or the step time duration. Similarly, Missura & Behnke (2011) propose a control strategy, whose main purpose is lateral disturbance rejection, by adjusting the step time duration and location of the next step after the disturbance, in order to correct the CoM trajectory. With the intention of generating faster gait cycles, Graf (2010) suggested a controller that eliminates the DSP while keeping the robot balanced by adjusting the step time duration. These methods become impractical in cases when the footprints cannot be changed during the walking motion, due to floor constraints for example.

Some works propose control strategies to stabilize the robot's posture in standing configurations (Boulic et al., 1995; Hofmann et al., 2009; Stephens, 2007). Boulic et al. (1995) present a strategy for posture stabilization of articulated bodies by keeping its

GCoM near to the center of the support polygon. The proposed controller uses the pseudo-inverse of the body's CoM Jacobian matrix to regulate the GCoM on the desired position. Furthermore, the limbs end-effectors poses are controlled to follow the reference in the null space of the robot's joints configuration, i.e. taking advantage of the robot's redundancy with respect to the main task, which is the regulation of the GCoM. Hofmann et al. (2009) adopted a strategy based on a Proportional-Derivative controller and the robot's dynamic model that allows the stabilization of a robot in unstable initial conditions, through the control of the CoM, robot's orientation, and swing-foot pose, simultaneously. The controller maneuver the robot's free-leg and torso in order to drive the robot's body to stable postures. Stephens (2007) combines two strategies to control the biped robot's posture, with the feet fixed on the ground, in the presence of disturbances: the ankle strategy and the hip strategy. The first one designs a controller which keeps all joints torques neglected, except the ankle joint. As this behavior can lead the global CoM to unstable regions, the hip strategy is applied, which consists of maneuvering the hip angles in order to incline the torso and lead the CoM to the desired target.

Ames et al. (2015) develop a formal controller synthesis for biped robots, consisting of a two-step approach to generate physically realizable stable walking. By formal synthesis we mean that the author works on formal mathematical axioms to obtain the controller.

In this work, we do not take into consideration the forces involved in the movement. As stated in Chapter 1, the humanoid robots locomotion can be classified into quasi-static and dynamic locomotion, and in quasi-static locomotion the effects of accelerations and inertial forces are negligible. Thus, we assume a quasi-static locomotion herein, even though a dynamic parameter is compensated in our control method (the robot's CoM). As a consequence, the ZMP based control methods are not applicable in our work, since they deal with dynamic locomotion, which is not our focus. Moreover, the methods to control the robot's posture in standing configurations do not address the locomotion control problem. Therefore, they are not appropriate in our case, and were presented just for the sake of information. About the formal method to controller synthesis, although it seems a very promising approach, it deals with hybrid systems, which is beyond the scope of this work.

The work presented in this dissertation is an extension of Park & Lee (2013). The authors use the Cooperative Dual Task-Space Framework (CDTS), proposed by Adorno et al. (2010) (see Section 3.4) to model the coordination between the robot's legs, and to control the robot's gait and balance. The proposed controller consists in a set of three strategies: the first one takes the actual feet poses to the desired references, given by the walking pattern generator, by controlling the relative variable (a pose defining the geometric relation between the feet); the second strategy drives the robot's CoM to the reference also given by the walking pattern generator, by using the pseudo-inverse of the CoM Jacobian matrix; and, finally, the third strategy regulates the robot's posture in an

upright configuration in order to avoid the generation of angular momentum, by controlling the absolute variable (an intermediate pose between the feet).

2.4 Chapter Overview

This chapter presented the most important works related to the biped robots field.

Section 2.1 showed two methods used in the context of biped robots to obtain the FKMs of the robot's limbs: the DH convention, used in this work, and the Screw Theory. Moreover, the two most common methodologies to obtain the whole-body model of the robot were described: the first one considers the robot as a free-floating system, with a base frame describing its position and orientation, and the second one, adopted in this work, considers the robot as a rooted kinematic chain, in which the system's base stays stationary during the whole step-motion, and is located in the link in contact with the ground. Three groups of methods were presented to solve the IK problem in the context of humanoid robots: the analytical methods, the numerical methods, and the control-theory based methods, which we adopted. The methods to model the robot's CoM were classified into two categories: the kinematically equivalent models, used herein, where an equivalent model for the CoM of each part of the robot's body is calculated and used to obtain the global CoM, and the equivalent kinematic chains, where the CoM is the end-effector of an equivalent chain whose base is located in the link in contact with the ground.

Section 2.2 presented the walking pattern generation methods, which can be classified into three groups: the inverted-pendulum-model based methods, the interpolation and optimization methods, and the walking pattern generation in the joints level. We adopted in this work a method based on the inverted pendulum model.

Finally, Section 2.3 discussed the works focused on gait and balance control. Those works were separated in five groups: the ZMP based methods, the on-line walking parameters adjustment methods, the posture stabilization methods, the formal controller synthesis, and CDTs-based method. This work is based on the last method.

3

Mathematical Background

This chapter reviews the basic concepts, definitions, and operations about quaternions and dual quaternions, and establishes the basic notation that will be used throughout this dissertation. Furthermore, the methods to obtain kinematic models and Jacobian matrices of serial robotic systems and cooperative robotic systems are presented. The concepts and notations presented herein are based on the work of Adorno (2011).

3.1 Basic Concepts in Dual Quaternion Theory

3.1.1 Quaternions

Quaternions are algebraic structures first introduced by Hamilton (1844, apud Adorno 2011), and can be regarded as an extension of complex numbers. They are defined as hypercomplex numbers consisting of a real and an imaginary part, whose imaginary part is composed by three imaginary components $\hat{i}$, $\hat{j}$, $\hat{k}$, also called imaginary (or quaternion) units. These quaternion units have the following properties:

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1, \quad (3.1)$$

which imply

$$\hat{i}\hat{j} = -\hat{j}\hat{i} = \hat{k}, \quad \hat{j}\hat{k} = -\hat{k}\hat{j} = \hat{i}, \quad \hat{k}\hat{i} = -\hat{i}\hat{k} = \hat{j}.$$

Given $h_1, h_2, h_3, h_4 \in \mathbb{R}$, the quaternion $\mathbf{h} \in \mathbb{H}$ is defined as

$$\mathbf{h} \triangleq h_1 + h_2\hat{i} + h_3\hat{j} + h_4\hat{k},$$

where the real part is denoted by $\text{Re}(\mathbf{h}) \triangleq h_1$, and the imaginary part is denoted by $\text{Im}(\mathbf{h}) \triangleq h_2\hat{i} + h_3\hat{j} + h_4\hat{k}$, such that $\mathbf{h} = \text{Re}(\mathbf{h}) + \text{Im}(\mathbf{h})$. Notice that complex numbers are a particular case of quaternions by letting $h_3 = h_4 = 0$.

Definition 3.1. Given the quaternions $\mathbf{a} = a_1 + a_2\hat{i} + a_3\hat{j} + a_4\hat{k}$ and $\mathbf{b} = b_1 + b_2\hat{i} + b_3\hat{j} + b_4\hat{k}$, the quaternions sum/subtraction is expressed by

$$\begin{aligned} \mathbf{a} \pm \mathbf{b} &= (a_1 + a_2\hat{i} + a_3\hat{j} + a_4\hat{k}) \pm (b_1 + b_2\hat{i} + b_3\hat{j} + b_4\hat{k}) \\ &= a_1 \pm b_1 + (a_2 \pm b_2)\hat{i} + (a_3 \pm b_3)\hat{j} + (a_4 \pm b_4)\hat{k}. \end{aligned}$$

Definition 3.2. Given the quaternions $\mathbf{a} = a_1 + a_2\hat{i} + a_3\hat{j} + a_4\hat{k}$ and $\mathbf{b} = b_1 + b_2\hat{i} + b_3\hat{j} + b_4\hat{k}$, the quaternion multiplication is expressed by

$$\begin{aligned} \mathbf{ab} &= (a_1 + a_2\hat{i} + a_3\hat{j} + a_4\hat{k}) (b_1 + b_2\hat{i} + b_3\hat{j} + b_4\hat{k}) \\ &= (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4) + \\ &\quad (a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3)\hat{i} + \\ &\quad (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)\hat{j} + \\ &\quad (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)\hat{k}. \end{aligned} \tag{3.2}$$

From (3.1), it is easy to see that the imaginary units do not commute, hence quaternion multiplication is not commutative, i.e. $\mathbf{ab} \neq \mathbf{ba}$.

Definition 3.3. Given the quaternion $\mathbf{a}$, its conjugate is defined as

$$\mathbf{h}^* \triangleq \text{Re}(\mathbf{h}) - \text{Im}(\mathbf{h}).$$

Definition 3.4. Given $\mathbf{a} \in \mathbb{H}$, its norm is defined as

$$\|\mathbf{h}\| \triangleq \sqrt{\mathbf{h}^*\mathbf{h}} = \sqrt{\mathbf{h}\mathbf{h}^*}.$$

Definition 3.5. The $\text{vec}_4 : \mathbb{H} \rightarrow \mathbb{R}^4$ operator performs a one-to-one mapping. Given the quaternion $\mathbf{h} = h_1 + h_2\hat{i} + h_3\hat{j} + h_4\hat{k}$, this operator is defined as (Adorno (2011))

$$\text{vec}_4(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}^T. \tag{3.3}$$

Furthermore, the Hamilton operators $\mathbf{H}_4^+(\cdot)$ and $\mathbf{H}_4^-(\cdot)$ for quaternions are defined as

(Adorno, 2011)

$$\mathbf{H}_4^+(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & -h_4 & h_3 \\ h_3 & h_4 & h_1 & -h_2 \\ h_4 & -h_3 & h_2 & h_1 \end{bmatrix}, \quad \mathbf{H}_4^-(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & h_4 & -h_3 \\ h_3 & -h_4 & h_1 & h_2 \\ h_4 & h_3 & -h_2 & h_1 \end{bmatrix}. \quad (3.4)$$

Given $\mathbf{a}, \mathbf{b} \in \mathbb{H}$, the Hamilton operators satisfy the following condition (Adorno, 2011):

$$\text{vec}_4(\mathbf{ab}) = \mathbf{H}_4^+(\mathbf{a}) \text{vec}_4(\mathbf{b}) = \mathbf{H}_4^-(\mathbf{b}) \text{vec}_4(\mathbf{a}). \quad (3.5)$$

The vec_4 is a linear operator, since it satisfies the superposition principle:

$$\begin{aligned} \text{vec}_4(\mathbf{a} + \mathbf{b}) &= \text{vec}_4 \mathbf{a} + \text{vec}_4 \mathbf{b}, \\ \text{vec}_4(\lambda \mathbf{a}) &= \lambda \text{vec}_4 \mathbf{a}, \quad \forall \lambda \in \mathbb{R}. \end{aligned}$$

Definition 3.6. The conjugating matrix C_4 is defined as (Adorno, 2011)

$$C_4 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \quad (3.6)$$

Given $\mathbf{a} \in \mathbb{H}$, this matrix satisfies the following condition

$$\text{vec}_4(\mathbf{h}^*) = C_4 \text{vec}_4 \mathbf{h}. \quad (3.7)$$

3.1.2 Dual Quaternions

Dual quaternions are a representation which combines the quaternions and dual numbers, introduced by Clifford (1873, apud Adorno, 2011).

Definition 3.7. Given two numbers $d_{\mathcal{P}}$ and $d_{\mathcal{D}}$ belonging to the same field, the dual number $\underline{d}$ is defined as (Selig, 2005, apud Adorno 2011)

$$\underline{d} = d_{\mathcal{P}} + \varepsilon d_{\mathcal{D}}, \quad (3.8)$$

where ε is the dual unit proposed by Clifford (1873), which is nilpotent and has the following properties

$$\begin{aligned} \varepsilon &\neq 0, \\ \varepsilon^2 &= 0. \end{aligned}$$

A dual number is composed by a primary part and a dual part, which can be extracted using the operators $\mathcal{P}(\underline{d})$ and $\mathcal{D}(\underline{d})$, respectively. In (3.8), $\mathcal{P}(\underline{d}) = d_{\mathcal{P}}$ and $\mathcal{D}(\underline{d}) = d_{\mathcal{D}}$.

Definition 3.8. The dual quaternion $\underline{h} \in \mathcal{H}$ is a dual number whose primary part and dual part are quaternions given by $\mathbf{h}_{\mathcal{P}}$, $\mathbf{h}_{\mathcal{D}}$, respectively, and is defined as (Selig, 2005)

$$\underline{h} \triangleq \mathbf{h}_{\mathcal{P}} + \varepsilon \mathbf{h}_{\mathcal{D}}.$$

Definition 3.9. Given the dual quaternions $\underline{a} = \mathbf{a}_{\mathcal{P}} + \varepsilon \mathbf{a}_{\mathcal{D}}$ and $\underline{b} = \mathbf{b}_{\mathcal{P}} + \varepsilon \mathbf{b}_{\mathcal{D}}$, the dual quaternion sum/subtraction is expressed by

$$\begin{aligned} \underline{a} \pm \underline{b} &= (\mathbf{a}_{\mathcal{P}} + \varepsilon \mathbf{a}_{\mathcal{D}}) \pm (\mathbf{b}_{\mathcal{P}} + \varepsilon \mathbf{b}_{\mathcal{D}}) \\ &= (\mathbf{a}_{\mathcal{P}} \pm \mathbf{b}_{\mathcal{P}}) + \varepsilon (\mathbf{a}_{\mathcal{D}} \pm \mathbf{b}_{\mathcal{D}}). \end{aligned}$$

Definition 3.10. Given the dual quaternions $\underline{a} = \mathbf{a}_{\mathcal{P}} + \varepsilon \mathbf{a}_{\mathcal{D}}$ and $\underline{b} = \mathbf{b}_{\mathcal{P}} + \varepsilon \mathbf{b}_{\mathcal{D}}$, the dual quaternion multiplication is expressed by

$$\begin{aligned} \underline{a}\underline{b} &= (\mathbf{a}_{\mathcal{P}} + \varepsilon \mathbf{a}_{\mathcal{D}})(\mathbf{b}_{\mathcal{P}} + \varepsilon \mathbf{b}_{\mathcal{D}}) \\ &= (\mathbf{a}_{\mathcal{P}}\mathbf{b}_{\mathcal{P}}) + \varepsilon (\mathbf{a}_{\mathcal{P}}\mathbf{b}_{\mathcal{D}} + \mathbf{a}_{\mathcal{D}}\mathbf{b}_{\mathcal{P}}). \end{aligned}$$

Equivalently to the quaternion multiplication, the dual quaternion multiplication is not commutative, i.e. $\underline{a}\underline{b} \neq \underline{b}\underline{a}$.

Definition 3.11. Given the dual quaternion $\underline{h} = \mathbf{h}_{\mathcal{P}} + \varepsilon \mathbf{h}_{\mathcal{D}}$, its conjugate is defined as

$$\underline{h}^* \triangleq \mathbf{h}_{\mathcal{P}}^* + \varepsilon \mathbf{h}_{\mathcal{D}}^*.$$

Definition 3.12. Given $\underline{a} \in \mathcal{H}$, its norm is defined as

$$\|\underline{h}\| \triangleq \sqrt{\underline{h}^* \underline{h}} = \sqrt{\underline{h} \underline{h}^*}. \quad (3.9)$$

Definition 3.13. The $\text{vec}_8 : \mathcal{H} \rightarrow \mathbb{R}^8$ operator performs a one-to-one mapping. Given the dual quaternion $\underline{h} = h_1 + h_2\hat{i} + h_3\hat{j} + h_4\hat{k} + \varepsilon (h_5 + h_6\hat{i} + h_7\hat{j} + h_8\hat{k})$, this operator is defined as (Adorno, 2011)

$$\text{vec}_8(\underline{h}) \triangleq \begin{bmatrix} h_1 & \cdots & h_8 \end{bmatrix}^T. \quad (3.10)$$

Furthermore, the Hamilton operators $\overset{+}{H}(\cdot)$ and $\overset{-}{H}(\cdot)$ extended for dual quaternions are obtained from (3.4) as follows:

$$\overset{+}{H}(\underline{h}) \triangleq \begin{bmatrix} \overset{+}{H}_4(\mathbf{h}_{\mathcal{P}}) & 0_{4 \times 4} \\ \overset{+}{H}_4(\mathbf{h}_{\mathcal{D}}) & \overset{+}{H}_4(\mathbf{h}_{\mathcal{P}}) \end{bmatrix}, \quad \overset{-}{H}(\underline{h}) \triangleq \begin{bmatrix} \overset{-}{H}_4(\mathbf{h}_{\mathcal{P}}) & 0_{4 \times 4} \\ \overset{-}{H}_4(\mathbf{h}_{\mathcal{D}}) & \overset{-}{H}_4(\mathbf{h}_{\mathcal{P}}) \end{bmatrix}. \quad (3.11)$$

Given $\underline{a}, \underline{b} \in \mathcal{H}$, these operators satisfy the following condition (Adorno, 2011):

$$\text{vec}_8(\underline{ab}) = \overset{+}{\mathbf{H}}(\underline{a}) \text{vec}_8(\underline{b}) = \bar{\mathbf{H}}(\underline{b}) \text{vec}_8(\underline{a}). \quad (3.12)$$

The vec_8 is a linear operator, since it satisfies the superposition principle:

$$\begin{aligned} \text{vec}_8(\underline{a} + \underline{b}) &= \text{vec}_8 \underline{a} + \text{vec}_8 \underline{b}, \\ \text{vec}_8(\lambda \underline{a}) &= \lambda \text{vec}_8 \underline{a}, \quad \lambda \in \mathbb{R}. \end{aligned}$$

Definition 3.14. The conjugating matrix extended for dual quaternions C_8 is defined as (Adorno, 2011)

$$C_8 \triangleq \begin{bmatrix} C_4 & 0_{4 \times 4} \\ 0_{4 \times 4} & C_4 \end{bmatrix}. \quad (3.13)$$

Given $\underline{a} \in \mathcal{H}$, this matrix satisfies the following condition

$$\text{vec}_8(\underline{h}^*) = C_8 \text{vec}_8 \underline{h}. \quad (3.14)$$

3.2 Dual Quaternions Representing Rigid Motions

A rigid body is completely described in space by its pose, i.e. its position and orientation regarding a specified frame of interest. Similarly, the movement of a rigid body (i.e. rigid motion) is described by a translation and a rotation with respect to the desired frame. In the particular context of robotics, rigid motions are useful to describe the kinematics of different systems, such as aerial and mobile robots, manipulators, etc.

Quaternions are primarily known to represent rotations (Kuipers, 2002, apud Adorno, 2011), and they may also be used to represent translations (Adorno, 2011). The specific class of dual quaternions with unit norm (or unit dual quaternions) represents rigid motions in a very compact way, by combining a rotation quaternion and a translation quaternion. The following subsections present the representation of rigid motions using dual quaternions.

3.2.1 Rotations and Translations Represented by Quaternions

Definition 3.15. Given $p_x, p_y, p_z \in \mathbb{R}$ the coordinates of a point in space with respect to a reference frame $\mathcal{F}_0$. The position of this point can be represented by a pure quaternion¹ $\mathbf{p}^0$ given by

$$\mathbf{p}^0 = p_x \hat{i} + p_y \hat{j} + p_z \hat{k}. \quad (3.15)$$

¹ $\mathbf{h}$ is a pure quaternion if $\text{Re}(\mathbf{h}) = 0$.

Letting p_x, p_y, p_z represent the displacements in x, y , and z axes, respectively, between the frames $\mathcal{F}_0$ and $\mathcal{F}_1$, the translation between these frames expressed in $\mathcal{F}_0$ is represented by $\mathbf{p}_{01}^0$, or simply $\mathbf{p}_1^0$, and is equally given by a pure quaternion similarly to the one defined in (3.15).

Hereafter, the notation of superscripts and subscripts will be used to represent the reference and the current frames, respectively.

Definition 3.16. Let $\mathcal{F}_0$ be a frame that will be rotated of an angle ϕ around a unit norm axis $\mathbf{n} = n_x \hat{i} + n_y \hat{j} + n_z \hat{k}$, and $\mathcal{F}_1$ be the resultant frame after the rotation. This rotation with respect to $\mathcal{F}_0$ is represented by the unit norm quaternion $\mathbf{r}_1^0$ given by

$$\mathbf{r}_1^0 = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \sin\left(\frac{\phi}{2}\right). \quad (3.16)$$

Letting $\phi = 0^\circ$ we obtain a null rotation, which is equal to 1.

Given the quaternion $\mathbf{r}_i^j$ representing a rotation from the frame $\mathcal{F}_j$ to $\mathcal{F}_i$,² and the quaternion $\mathbf{p}^i$ representing a point in $\mathcal{F}_i$, the projection of $\mathbf{p}^i$ in the frame $\mathcal{F}_j$, given by $\mathbf{p}^j$, is expressed by

$$\mathbf{p}^j = \mathbf{r}_i^j \mathbf{p}^i (\mathbf{r}_i^j)^*. \quad (3.17)$$

Example 3.1. Let $\mathbf{r}_1^0$ and $\mathbf{p}_{01}^0$ be quaternions representing, respectively, the rotation and translation from the frame $\mathcal{F}_0$ to $\mathcal{F}_1$, as depicted in Figure 3.1. The point $\mathbf{p}^1$ with respect to $\mathcal{F}_1$, when represented with respect to $\mathcal{F}_0$, is computed by

$$\mathbf{p}^0 = \mathbf{p}_{01}^0 + \mathbf{r}_1^0 \mathbf{p}^1 (\mathbf{r}_1^0)^*.$$

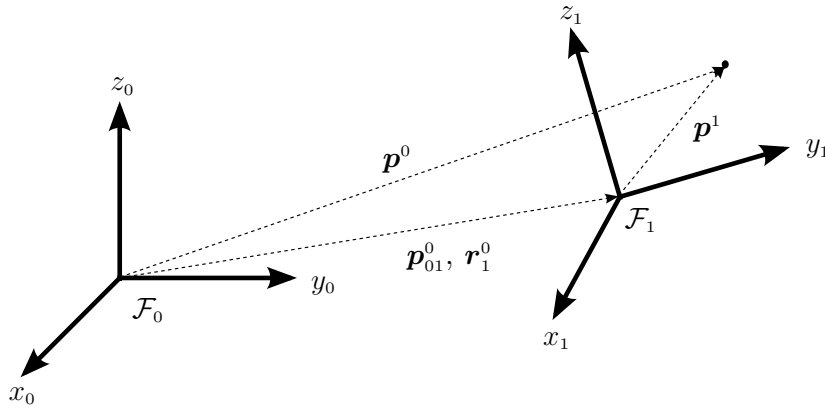


Figure 3.1: Example of the projection of a point.

²Note that the convention of moving frames is being used.

3.2.2 Rigid Motions Represented by Dual Quaternions

Definition 3.17. Given $\mathbf{p}_{01}^0, \mathbf{r}_1^0 \in \mathbb{H}$, defined by (3.15) and (3.16), respectively, representing the position and orientation of the frame $\mathcal{F}_1$ with respect to frame $\mathcal{F}_0$. These variables together express the dual position (or the complete pose) of $\mathcal{F}_1$ with respect to $\mathcal{F}_0$, represented by the unit dual quaternion $\underline{\mathbf{x}}_1^0$, which is defined as (Adorno, 2011)

$$\underline{\mathbf{x}}_1^0 \triangleq \mathbf{r}_1^0 + \frac{1}{2}\varepsilon \mathbf{p}_{01}^0 \mathbf{r}_1^0.$$

Similarly, $\underline{\mathbf{x}}_1^0$ can be regarded as the rigid motion between frames $\mathcal{F}_0$ and $\mathcal{F}_1$, where $\mathbf{p}_{01}^0$ would represent the translation and $\mathbf{r}_1^0$ the rotation.

By inspection, given a unit dual quaternion $\underline{\mathbf{x}} = \mathcal{P}(\underline{\mathbf{x}}) + \varepsilon \mathcal{D}(\underline{\mathbf{x}})$ representing a pose or a rigid motion, the orientation/rotation is obtained by taking the primary part of $\underline{\mathbf{x}}$, as follows

$$\mathbf{r} = \mathcal{P}(\underline{\mathbf{x}}),$$

and the position/translation is extracted by the relation

$$\mathbf{p} = 2 \mathcal{D}(\underline{\mathbf{x}}) \mathcal{P}(\underline{\mathbf{x}})^*. \quad (3.18)$$

Fact 3.1. Given $\underline{\mathbf{x}}_i^j \in \mathcal{H}$ a unit dual quaternion representing the rigid motion from frame $\mathcal{F}_j$ to $\mathcal{F}_i$. The conjugate of $\underline{\mathbf{x}}_i^j$ represents the inverse transformation—i.e. from the frame $\mathcal{F}_i$ to $\mathcal{F}_j$ —as follows

$$(\underline{\mathbf{x}}_i^j)^* = \underline{\mathbf{x}}_j^i. \quad (3.19)$$

Consequently, the multiplication of a unit dual quaternion by its conjugate corresponds to a null transformation as follows

$$\underline{\mathbf{x}}_i^j (\underline{\mathbf{x}}_i^j)^* = (\underline{\mathbf{x}}_i^j)^* \underline{\mathbf{x}}_i^j = 1.$$

Example 3.2. Let $\underline{\mathbf{x}}_1^0 = \mathbf{r}_1^0 + \frac{1}{2}\varepsilon \mathbf{p}_{01}^0 \mathbf{r}_1^0$ be a dual quaternion representing a rigid motion from the frame $\mathcal{F}_0$ to $\mathcal{F}_1$, where $\mathbf{r}_1^0 = \cos(\frac{\phi}{2}) + \mathbf{n} \sin(\frac{\phi}{2})$. The rigid motion represented by $\underline{\mathbf{x}}_1^0$ is depicted in Figure 3.2.

Definition 3.18. Let $\underline{\mathbf{x}}_1^0, \underline{\mathbf{x}}_2^1, \dots, \underline{\mathbf{x}}_n^{n-1} \in \mathcal{H}$ be a sequence of n rigid motions. The resultant pose after these transformations, with respect to the first frame $\mathcal{F}_0$, is represented by the dual quaternion $\underline{\mathbf{x}}_n^0$ as follows

$$\underline{\mathbf{x}}_n^0 = \underline{\mathbf{x}}_1^0 \underline{\mathbf{x}}_2^1 \dots \underline{\mathbf{x}}_n^{n-1}.$$

Definition 3.19. Given the unit dual quaternion $\underline{\mathbf{x}} = \mathbf{r} + \frac{1}{2}\varepsilon \mathbf{p} \mathbf{r}$, with $\mathbf{r} = \cos(\phi/2) + \mathbf{n} \sin(\phi/2)$, $\mathbf{n} = n_x \hat{i} + n_y \hat{j} + n_z \hat{k}$, and $\mathbf{p} = p_x \hat{i} + p_y \hat{j} + p_z \hat{k}$, the logarithm of $\underline{\mathbf{x}}$ is defined as (Adorno (2011))

$$\log \underline{\mathbf{x}} \triangleq \frac{\mathbf{n}\phi}{2} + \varepsilon \frac{\mathbf{p}}{2}. \quad (3.20)$$

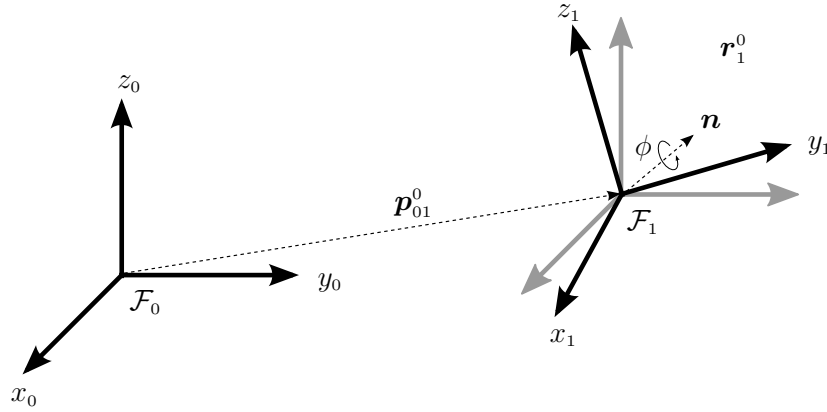


Figure 3.2: Example of a rigid motion represented by the dual quaternion in Example (3.2).

Note that $\log \underline{\mathbf{x}} \in \mathcal{H}$. In summary, applying the log operator in a unit dual quaternion will extract the arguments which describe the rigid motion, i.e. the rotation angle and axis in the primary part, and the translation in the dual part.

Example 3.3. Let the unit dual quaternion $\underline{\mathbf{x}} = \mathbf{r} + \frac{1}{2}\varepsilon\mathbf{p}\mathbf{r}$ be the pose of a rigid body in space. The rotation angle ϕ of the body is given by

$$\phi = 2 \|\mathcal{P}(\log \underline{\mathbf{x}})\|,$$

and the rotation axis is expressed by

$$\mathbf{n} = \frac{\mathcal{P}(\log \underline{\mathbf{x}})}{\|\mathcal{P}(\log \underline{\mathbf{x}})\|}.$$

The amount of displacement of the body with respect to the origin of the reference frame is

$$p = 2 \|\mathcal{D}(\log \underline{\mathbf{x}})\|,$$

and the displacement direction is calculated by

$$\mathbf{p}_u = \frac{\mathcal{D}(\log \underline{\mathbf{x}})}{\|\mathcal{D}(\log \underline{\mathbf{x}})\|},$$

such that $\mathbf{p} = p\mathbf{p}_u$.

Definition 3.20. Given the pure dual quaternion $\underline{\mathbf{g}}$ (i.e., $\text{Re}(\mathcal{P}(\underline{\mathbf{g}})) = \text{Im}(\mathcal{D}(\underline{\mathbf{g}})) = 0$), its exponential is expressed by (Adorno, 2011)

$$\exp \underline{\mathbf{g}} \triangleq \mathcal{P}_{exp}(\underline{\mathbf{g}}) + \varepsilon \mathcal{D}(\underline{\mathbf{g}}) \mathcal{P}_{exp}(\underline{\mathbf{g}}), \quad (3.21)$$

where

$$\mathcal{P}_{exp}(\underline{g}) = \begin{cases} \cos \|\mathcal{P}(\underline{g})\| + \frac{\sin \|\mathcal{P}(\underline{g})\|}{\|\mathcal{P}(\underline{g})\|} \mathcal{P}(\underline{g}) & \text{if } \|\mathcal{P}(\underline{g})\| \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

Definition 3.21. Given $\lambda \in \mathbb{R}$, using (3.20) and (3.21), the unit dual quaternion $\underline{x}$ raised by λ is given by

$$\underline{x}^{\{\lambda\}} \triangleq \exp(\lambda \log \underline{x}). \quad (3.22)$$

This definition will be useful to describe the Cooperative Dual Task-Space Framework in Section 3.4.

Example 3.4. Let the unit dual quaternion $\underline{x} = \underline{r} + \frac{1}{2}\varepsilon \underline{p}\underline{r}$ be the pose of a rigid body in space, with $\underline{r} = \cos(\phi/2) + \underline{n} \sin(\phi/2)$. Recalling (3.20), raising $\underline{x}$ to $1/2$, we obtain

$$\begin{aligned} \underline{x}^{\{1/2\}} &= \exp\left(\frac{1}{2} \log \underline{x}\right) \\ &= \exp\left(\frac{\underline{n}\phi}{4} + \varepsilon \frac{\underline{p}}{4}\right) \\ &= \left(\cos \frac{\phi}{4} + \underline{n} \sin \frac{\phi}{4}\right) + \frac{1}{2}\varepsilon \left[\frac{\underline{p}}{2} \left(\cos \frac{\phi}{4} + \underline{n} \sin \frac{\phi}{4}\right)\right] \\ &= \underline{r}^{\{1/2\}} + \frac{1}{2}\varepsilon \underline{p}' \underline{r}^{\{1/2\}}, \end{aligned}$$

where $\underline{p}' = \underline{p}/2$. By inspection, we can notice that $\underline{x}^{\{1/2\}}$ represents a rigid motion given by half rotation and half translation of $\underline{x}$.

3.3 Robots Kinematic Modeling

3.3.1 Forward Kinematics Model

The Denavit-Hartenberg (DH) convention is useful to obtain the Forward Kinematics Model (FKM) of some robots. As mentioned in Section 2.1, it consists of a convention to define frames for each joint of a serial kinematic chain, such that four geometric parameters are sufficient to establish the rigid transformation between two consecutive joints in the kinematic chain (Spong et al., 2006):

θ : rotation angle in the z axis with respect to the previous joint;

d : displacement in the z axis with respect to the previous joint;

a : displacement in the current x axis;

α : rotation angle in the current x axis.

The four homogeneous transformations in dual quaternion form regarding these parameters are, respectively, given by Adorno (2011)

$$\begin{aligned} \mathbf{r}_{\theta,z} &= \cos\left(\frac{\theta}{2}\right) + \hat{k} \sin\left(\frac{\theta}{2}\right), \\ \underline{\mathbf{p}}_{d,z} &= 1 + \varepsilon \frac{d}{2} \hat{k}, \\ \underline{\mathbf{p}}_{a,x} &= 1 + \varepsilon \frac{a}{2} \hat{i}, \\ \mathbf{r}_{\alpha,x} &= \cos\left(\frac{\alpha}{2}\right) + \hat{i} \sin\left(\frac{\alpha}{2}\right). \end{aligned}$$

The transformation between the frames of two consecutive joints is represented by the dual quaternion $\underline{\mathbf{x}}_{DH}$, which is expressed by

$$\underline{\mathbf{x}}_{DH} = \mathbf{r}_{\theta,z} \underline{\mathbf{p}}_{d,z} \underline{\mathbf{p}}_{a,x} \mathbf{r}_{\alpha,x}. \quad (3.23)$$

Definition 3.22. The FKM of a serial kinematic chain composed of n revolute joints provides the pose of its end-effector with respect to its base, given the current joints configuration. Using dual quaternion algebra, the FKM is represented by a mapping such that $\mathbf{f}: \mathbb{T}^n \rightarrow \mathcal{H}$; that is,

$$\underline{\mathbf{x}}_e = \mathbf{f}(q), \quad (3.24)$$

where q is an n -dimensional vector representing the joints angles vector.

Let the transformation $\underline{\mathbf{x}}_i^{i-1}$ be a dual quaternion relating the pose of joint i with respect to the previous one obtained by (3.23), such that $i \in \{1, \dots, n\}$. Then $\underline{\mathbf{x}}_e$ is expressed by

$$\underline{\mathbf{x}}_e = \underline{\mathbf{x}}_1^0 \underline{\mathbf{x}}_2^1 \dots \underline{\mathbf{x}}_n^{n-1}. \quad (3.25)$$

Note that the reference frame of $\underline{\mathbf{x}}_e$ in this case is $\mathcal{F}_0$, which represents the base of the kinematic chain.

3.3.2 Differential Forward Kinematics Model

Consider a serial kinematic chain composed of n revolute joints. Using dual quaternion algebra, the Differential Forward Kinematics Model (DFKM) provides the relationship between the end-effector generalized velocities and the joints velocities, which is represented by

$$\text{vec}_8 \dot{\underline{\mathbf{x}}}_e = J \dot{q}, \quad (3.26)$$

where $\dot{\underline{\mathbf{x}}}_e$ represents the time derivative of the end-effector pose (i.e., the generalized velocities), $\dot{q}$ is an n -dimensional vector representing the joints velocities vector and $J \in \mathbb{R}^{8 \times n}$ is the analytical Jacobian matrix obtained algebraically, deriving (3.25) with respect to q (Adorno, 2011). The Jacobian matrix can be discriminated into two parts as

follows

$$J = \begin{bmatrix} J_{\mathcal{P}} \\ J_{\mathcal{D}} \end{bmatrix},$$

where $J_{\mathcal{P}}$ and $J_{\mathcal{D}}$ are related with the primary and dual parts of $\underline{\dot{\mathbf{x}}}_e$, respectively, and satisfy the relations

$$\text{vec}_4(\mathcal{P}(\underline{\dot{\mathbf{x}}}_e)) = J_{\mathcal{P}}\dot{\mathbf{q}}, \quad \text{vec}_4(\mathcal{D}(\underline{\dot{\mathbf{x}}}_e)) = J_{\mathcal{D}}\dot{\mathbf{q}}. \quad (3.27)$$

Let the end-effector pose be represented by the dual quaternion $\underline{\mathbf{x}}_e = \mathbf{r}_e + \frac{1}{2}\varepsilon\mathbf{p}_e\mathbf{r}_e$. The orientation Jacobian matrix of this kinematic chain, represented by $J_{ori} \in \mathbb{R}^{4 \times n}$, relates the joints velocities with the time derivative of end-effector orientation, denoted by $\dot{\mathbf{r}}_e$. Recalling (3.27), we obtain J_{ori} by inspection, as follows:

$$\begin{aligned} \text{vec}_4 \dot{\mathbf{r}}_e &= \text{vec}_4(\mathcal{P}(\underline{\dot{\mathbf{x}}}_e)), \\ &= J_{\mathcal{P}}\dot{\mathbf{q}}, \\ &= J_{ori}\dot{\mathbf{q}}. \end{aligned} \quad (3.28)$$

The position Jacobian matrix $J_{pos} \in \mathbb{R}^{4 \times n}$ relates the joints velocities with the end-effector linear velocity, denoted by $\dot{\mathbf{p}}_e$. Using (3.12), (3.14), (3.18), and (3.27), J_{pos} is determined by expanding the following equation (Adorno et al., 2010):

$$\begin{aligned} \text{vec}_4 \dot{\mathbf{p}}_e &= \text{vec}_4 \left[\frac{d}{dt} (2 \mathcal{D}(\underline{\mathbf{x}}_e) \mathcal{P}(\underline{\mathbf{x}}_e)^*) \right], \\ &= \text{vec}_4 (2 \mathcal{D}(\underline{\dot{\mathbf{x}}}_e) \mathcal{P}(\underline{\mathbf{x}}_e)^*) + \text{vec}_4 (2 \mathcal{D}(\underline{\mathbf{x}}_e) \mathcal{P}(\underline{\dot{\mathbf{x}}}_e)^*), \\ &= \text{vec}_4 (2 \mathcal{D}(\underline{\dot{\mathbf{x}}}_e) \mathcal{P}(\underline{\mathbf{x}}_e)^*) + \text{vec}_4 (2 \mathcal{D}(\underline{\mathbf{x}}_e) \mathcal{P}(\underline{\dot{\mathbf{x}}}_e)^*), \\ &= 2\bar{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{x}}_e^*)) \text{vec}_4(\mathcal{D}(\underline{\dot{\mathbf{x}}}_e)) + 2\overset{+}{\mathbf{H}}(\mathcal{D}(\underline{\mathbf{x}}_e)) \text{vec}_4(\mathcal{P}(\underline{\dot{\mathbf{x}}}_e)^*), \\ &= \underbrace{\left[2\bar{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{x}}_e^*)) J_{\mathcal{D}} + 2\overset{+}{\mathbf{H}}(\mathcal{D}(\underline{\mathbf{x}}_e)) C_4 J_{\mathcal{P}} \right]}_{J_{pos}} \dot{\mathbf{q}}. \end{aligned} \quad (3.29)$$

Let the dual quaternion $\underline{\mathbf{x}}_0^1$ be the constant homogeneous transformation from frame $\mathcal{F}_0$ to $\mathcal{F}_1$. Given a serial kinematic chain, whose DFKM with respect to $\mathcal{F}_0$ is given by $\text{vec}_8 \underline{\dot{\mathbf{x}}}_e = J\dot{\mathbf{q}}$, using (3.12), the DFKM of this chain regarding $\mathcal{F}_1$ is derived as follows:

$$\begin{aligned} \text{vec}_8 \underline{\dot{\mathbf{x}}}_e^1 &= \text{vec}_8 (\underline{\mathbf{x}}_0^1 \underline{\dot{\mathbf{x}}}_e) \\ &= \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_0^1) \text{vec}_8 \underline{\dot{\mathbf{x}}}_e \\ &= \underbrace{\overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_0^1) J}_{J^1} \dot{\mathbf{q}}, \end{aligned}$$

where J^1 is the Jacobian matrix of the serial kinematic chain with respect to $\mathcal{F}_1$. As a convention, hereafter the superscript in Jacobian matrices will denote the frame used to obtain the matrix, other than the normally considered serial chain base.

3.4 Cooperative Dual Task-Space Framework

The Cooperative Dual Task-Space (CDTS) was proposed by Adorno et al. (2010), where a set of control strategies developed to perform two-arm manipulation were presented, whose aim is to simplify the task definition. This framework was derived from the Cooperative Task-Space, which defines four variables in terms of absolute and relative position and orientation with an intrinsic physical meaning.

Generally speaking, this framework can be applied to describe a system composed by two kinematic chains working cooperatively, aiming to represent the geometrical relationships between them. Similarly to the Cooperative Task-Space (Chiacchio et al., 1996), the state of the system is completely determined by the absolute and relative position and orientation between the chains' end-effectors, and these four variables are clustered into two dual quaternions: one representing the relative pose $\underline{\mathbf{x}}_r$ and the other representing the absolute pose $\underline{\mathbf{x}}_a$. The relative pose $\underline{\mathbf{x}}_r$ provides the relative configuration between the two end-effectors, and $\underline{\mathbf{x}}_a$ is an intermediate pose between them. These variables are called "cooperative variables," as they are often used in cooperative systems.

Definition 3.23. Let the dual quaternions $\underline{\mathbf{x}}_1$ and $\underline{\mathbf{x}}_2$ denote the poses of two chains' end-effectors regarding a common reference frame. The cooperative variables for these chains are defined as (Adorno et al., 2010)

$$\underline{\mathbf{x}}_r \triangleq \underline{\mathbf{x}}_1^* \underline{\mathbf{x}}_2, \quad (3.30)$$

$$\underline{\mathbf{x}}_a \triangleq \underline{\mathbf{x}}_1 \underline{\mathbf{x}}_r^{\{1/2\}}, \quad (3.31)$$

where $\underline{\mathbf{x}}_r, \underline{\mathbf{x}}_a \in \mathcal{H}$. The relative pose is given with respect to the first end-effector's pose and the absolute pose is expressed with respect to the chosen reference frame. In this work, the absolute variable is a pose in the exact center between the two feet poses, since the relative variable is raised by $1/2$. Thus, the absolute pose is not biased by any of the feet poses.

Example 3.5. The dual quaternions $\underline{\mathbf{x}}_1$ and $\underline{\mathbf{x}}_2$ represent the end-effectors of a cooperative system, expressed with respect to a reference frame $\mathcal{F}_0$. Let $\underline{\mathbf{x}}_a$ and $\underline{\mathbf{x}}_r$ be the cooperative variables calculated according to (3.30) and (3.31), such that $\underline{\mathbf{x}}_r = \mathbf{r} + \frac{1}{2}\varepsilon\mathbf{p}\mathbf{r}$, with $\mathbf{r} = \cos(\phi/2) + \hat{k} \sin(\phi/2)$ and $\mathbf{p} = p\hat{j}$. The cooperative variables are represented in Figure 3.3.

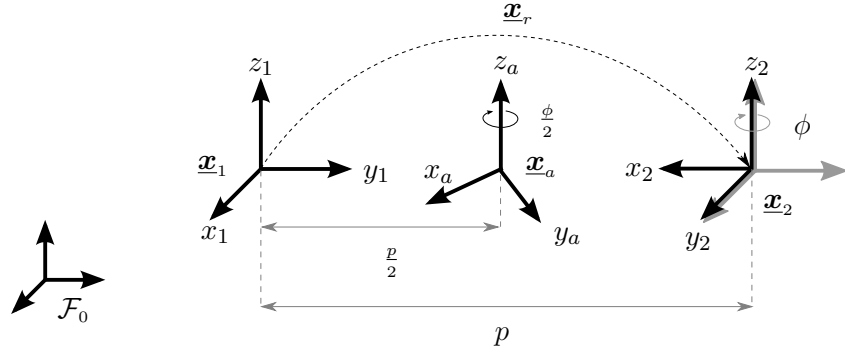


Figure 3.3: Cooperative variables.

Definition 3.24. Consider two serial kinematic chains working cooperatively, the first one composed of n_1 revolute joints, and the other composed of n_2 revolute joints. Let $\underline{x}_1, \underline{x}_2 \in \mathcal{H}$ be the end-effectors poses of these chains expressed in a common reference frame $\mathcal{F}_0$, and $\underline{x}_r, \underline{x}_a \in \mathcal{H}$ be the cooperative variables regarding these chains.

Furthermore, let $J_1 \in \mathbb{R}^{8 \times n_1}$ and $J_2 \in \mathbb{R}^{8 \times n_2}$ be the Jacobian matrices of the two chains with respect to $\mathcal{F}_0$, and $q_1 \in \mathbb{R}^{n_1}$ and $q_2 \in \mathbb{R}^{n_2}$ the joints vectors of the chains, such that $\text{vec}_8 \dot{\underline{x}}_1 = J_1 \dot{q}_1$ and $\text{vec}_8 \dot{\underline{x}}_2 = J_2 \dot{q}_2$.

Taking the first derivative of (3.30), and using (3.12), (3.13), (3.14), and (3.27), the DFKM for the relative pose is derived as follows (Adorno et al., 2010):

$$\begin{aligned}
 \text{vec}_8 \dot{\underline{x}}_r &= \text{vec}_8 (\dot{\underline{x}}_1^* \underline{x}_2 + \underline{x}_1^* \dot{\underline{x}}_2) \\
 &= \text{vec}_8 (\dot{\underline{x}}_1^* \underline{x}_2) + \text{vec}_8 (\underline{x}_1^* \dot{\underline{x}}_2) \\
 &= \bar{H}(\underline{x}_2) \text{vec}_8 (\dot{\underline{x}}_1^*) + \overset{+}{H}(\underline{x}_1^*) \text{vec}_8 \dot{\underline{x}}_2 \\
 &= \bar{H}(\underline{x}_2) C_8 J_1 \dot{q}_1 + \overset{+}{H}(\underline{x}_1^*) J_2 \dot{q}_2 \\
 &= \underbrace{\left[\bar{H}(\underline{x}_2) C_8 J_1 \quad \overset{+}{H}(\underline{x}_1^*) J_2 \right]}_{J_r} \dot{q}, \tag{3.32}
 \end{aligned}$$

where $\dot{q} \in \mathbb{R}^{(n_1+n_2)}$ is a vector representing the joints velocities of both chains— $\dot{q}_1$ and $\dot{q}_2$ —given by

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix},$$

and $J_r \in \mathbb{R}^{8 \times (n_1+n_2)}$ is the Jacobian matrix for the relative pose, hereafter called the “relative Jacobian matrix.”

Likewise, the DFKM for the absolute pose is derived from (3.31) as follows (Adorno

et al., 2010):

$$\begin{aligned}
\text{vec}_8 \dot{\underline{\mathbf{x}}}_a &= \text{vec}_8 (\dot{\underline{\mathbf{x}}}_1 \underline{\mathbf{x}}_r^{\{1/2\}} + \underline{\mathbf{x}}_1 \dot{\underline{\mathbf{x}}}_r^{\{1/2\}}) \\
&= \text{vec}_8 (\dot{\underline{\mathbf{x}}}_1 \underline{\mathbf{x}}_r^{\{1/2\}}) + \text{vec}_8 (\underline{\mathbf{x}}_1 \dot{\underline{\mathbf{x}}}_r^{\{1/2\}}) \\
&= \bar{\mathbf{H}}(\underline{\mathbf{x}}_r^{\{1/2\}}) \text{vec}_8 \dot{\underline{\mathbf{x}}}_1 + \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_1) \text{vec}_8 (\dot{\underline{\mathbf{x}}}_r^{\{1/2\}}) \\
&= \bar{\mathbf{H}}(\underline{\mathbf{x}}_r^{\{1/2\}}) J_1 \dot{\mathbf{q}}_1 + \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_1) J_{r/2} \dot{\mathbf{q}} \\
&= \underbrace{\left\{ \left[\bar{\mathbf{H}}(\underline{\mathbf{x}}_r^{\{1/2\}}) J_1 \quad \mid \quad 0_{8 \times n_2} \right] + \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_1) J_{r/2} \right\}}_{J_a} \dot{\mathbf{q}}, \tag{3.33}
\end{aligned}$$

where $J_a \in \mathbb{R}^{8 \times (n_1 + n_2)}$ is the Jacobian matrix for the absolute pose, hereafter called the “absolute Jacobian matrix,” and $J_{r/2}$ satisfies the relation $d(\underline{\mathbf{x}}_r^{\{1/2\}})/dt = J_{r/2} \dot{\mathbf{q}}$ (Adorno et al., 2010).

The matrices J_r and J_a are denoted as “cooperative Jacobian matrices,” as they are related to the cooperative variables.

The main advantages of using the CDTS is that the coordination between two kinematic chains working cooperatively can be completely described using only two variables. Moreover, the cooperative variables explicitly take into account the kinematic constraints of the system, which are imposed by the task.

3.5 Chapter Overview

This chapter reviewed the basic concepts, definitions, and operations about quaternions, dual quaternions, and the CDTS.

Section 3.1 presented the properties, definitions, and operations with quaternions and dual quaternions.

Section 3.2 detailed the representation of rotations and translations using quaternions, and rigid motions using dual quaternions. Furthermore, some useful operators were presented to extract and modify the parameters of the dual quaternion representing a pose.

Section 3.3 showed the methodology to obtain the FKM and DFKM of a kinematic chain, and Section 3.4 presented the concepts and definitions related to the CDTS.

4

Kinematic Modeling

This chapter presents the kinematic modeling method for a humanoid robot using the dual quaternion (DQ) algebra and the Cooperative Dual Task-Space (CDTS). The chapter is organized as follows: in Section 4.1 the robot's limbs kinematic models are obtained; Section 4.2 discusses the choice of the system's reference frame during the walking motion; in Section 4.3 the method to obtain the kinematic model of the robot's CoM is presented; and in Section 4.4 the kinematic model that describes the coordinated behavior of the robot's legs is shown.

4.1 Kinematic Modeling of the Humanoid Robot's Limbs

A humanoid robot is composed, in general, by four limbs—two arms and two legs—connected by a torso that, in some cases, is articulated. One way to obtain the Forward Kinematics Models (FKMs) of these limbs is to use the parameters defined by the Denavit-Hartenberg (DH) convention. Hereafter, the legs end-effectors will be called feet, and the arms end-effectors will be called hands.

Consider a humanoid robot with a non-articulated torso and n limbs, which are labeled from 1 to n . The i -th limb has $k(i)$ links (and we define that $k(0) = 0$) and $\mathcal{F}_i$ is a frame attached to its base. The FKM of the limb i provides the end-effector pose with respect

to the frame $\mathcal{F}_i$, and, according to definition 3.24, it is represented by

$$\underline{\mathbf{x}}_{ei}^i = \underline{\mathbf{f}}(q_i),$$

where the subscript ei represents the end-effector of limb i , and $q_i = [q_{i0} \ q_{i1} \ \cdots \ q_{ik(i)}]^T$ is its joints vector. Recalling the vec_8 operator defined in (3.10) and the representation of the Differential Forward Kinematics Model (DFKM) in (3.26), the DFKM of limb i is given by

$$\text{vec}_8 \dot{\underline{\mathbf{x}}}_{ei}^i = J_{ei} \dot{q}_i,$$

where J_{ei} is the Jacobian matrix of the limb i with respect to frame $\mathcal{F}_i$.¹

For the purpose of controlling the robot motions, the end-effectors poses of all limbs are represented in a common frame, defined as $\mathcal{F}_b$, which is attached to the torso's geometric center so as to avoid the calculation of the robot's body velocity during the motion.

The coordinate transformation between $\mathcal{F}_b$ and the base of the i -th limb, $\mathcal{F}_i$, is constant and determined by the geometry of the robot's body. This transformation² is defined as

$$\underline{\mathbf{x}}_i^b \triangleq \mathbf{r}_i^b + \frac{1}{2} \varepsilon \mathbf{p}_{bi}^b \mathbf{r}_i^b, \quad (4.1)$$

where $\mathbf{r}_i^b$ and $\mathbf{p}_{bi}^b$ are quaternions representing a rotation and a translation from $\mathcal{F}_b$ to $\mathcal{F}_i$, respectively.

Figure 4.1 illustrates the frames $\mathcal{F}_b$ and $\mathcal{F}_i$, and also the coordinate transformations $\underline{\mathbf{x}}_i^b$ and $\underline{\mathbf{x}}_{ei}^i$ for a specific humanoid robot with four limbs.

The end-effector of the i -th limb expressed with respect to $\mathcal{F}_b$ is given by

$$\underline{\mathbf{x}}_{ei}^b = \underline{\mathbf{x}}_i^b \underline{\mathbf{x}}_{ei}^i. \quad (4.2)$$

Recalling the relation defined in (3.12), the DFKM of this limb with respect to $\mathcal{F}_b$ is derived as follows

$$\begin{aligned} \text{vec}_8 \dot{\underline{\mathbf{x}}}_{ei}^b &= \text{vec}_8 (\underline{\mathbf{x}}_i^b \dot{\underline{\mathbf{x}}}_{ei}^i) \\ &= \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_i^b) \text{vec}_8 \dot{\underline{\mathbf{x}}}_{ei}^i \\ &= \underbrace{\overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_i^b) J_{ei}^i}_{J_{ei}^b} \dot{q}_i. \end{aligned} \quad (4.3)$$

where J_{ei}^b represents the Jacobian matrix of the limb i expressed in $\mathcal{F}_b$.

¹The Jacobian matrix can be algebraically calculated from DQ algebra (Adorno, 2011).

²Note that we consider a moving coordinate system.

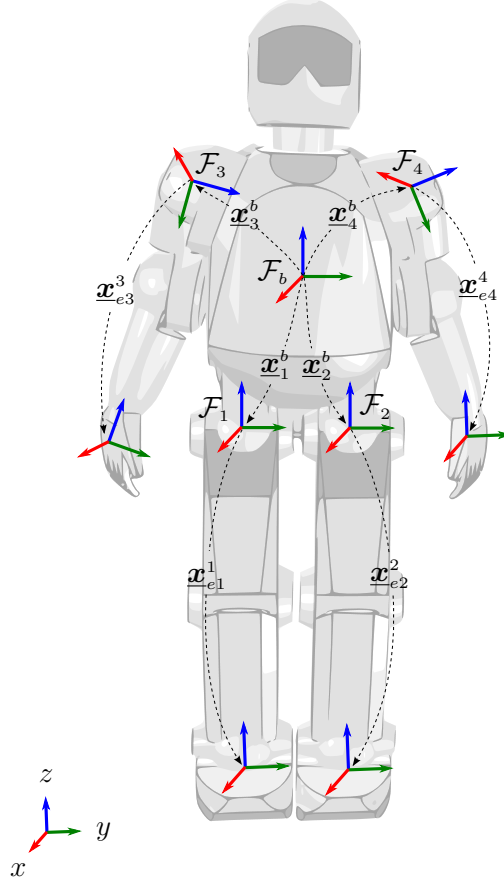


Figure 4.1: Robot's limbs frames and coordinate transformations.

4.2 System's Reference Frame

During the gait cycle, at least one foot keeps contact with the ground, which is called the “supporting-foot,” and the foot performing the step motion is called the “swinging-foot.” Thus, to keep the system's reference frame fixed during the step motion, it must be attached to the supporting-foot in the single-support phases, and in any of the feet in the double-support phase. Consequently, the system's reference frame is exchanged whenever the supporting-foot is switched.

Let the frames attached to the right and left foot be represented by $\mathcal{F}_{e1}$ and $\mathcal{F}_{e2}$, respectively. We wish to represent the end-effector pose of the i -th limb with respect to the current reference frame. As it changes constantly from $\mathcal{F}_{e1}$ and $\mathcal{F}_{e2}$, for the sake of simplicity, the reference frame is represented by $\mathcal{F}_{ej}$, where $j \in \{1, 2\}$. Using (4.2), we have that

$$\underline{\mathbf{x}}_{ei}^{ej} = \underline{\mathbf{x}}_b^{ej} \underline{\mathbf{x}}_{ei}^b, \quad (4.4)$$

where $\underline{\mathbf{x}}_b^{ej} = (\underline{\mathbf{x}}_{ej}^b)^*$, as shown in (3.19).

When i equals j , the dual quaternion $\underline{\mathbf{x}}_{ei}^{ej}$ represents a frame with respect to itself, which clearly leads to the origin of this coordinate system, represented by 1, since $\underline{\mathbf{x}}_{ei}^{ei} = (\underline{\mathbf{x}}_{ei}^b)^* \underline{\mathbf{x}}_{ei}^b = \|\underline{\mathbf{x}}_{ei}^b\|^2 = 1$. Considering this, from now on we will consider that $i \neq j$.

Let $q_i \in \mathbb{R}^{n_i}$ and $q_j \in \mathbb{R}^{n_j}$, represent the joints vectors of the i -th and j -th limbs, respectively. Using (3.12) and (3.14), the DFKM of the i -th limb with respect to the current reference frame $\mathcal{F}_{ej}$ is derived as follows

$$\begin{aligned}
\text{vec}_8 \dot{\underline{\mathbf{x}}}_{ei}^{ej} &= \text{vec}_8 (\dot{\underline{\mathbf{x}}}_b^{ej} \underline{\mathbf{x}}_{ei}^b + \underline{\mathbf{x}}_b^{ej} \dot{\underline{\mathbf{x}}}_{ei}^b) \\
&= \text{vec}_8 (\dot{\underline{\mathbf{x}}}_b^{ej} \underline{\mathbf{x}}_{ei}^b) + \text{vec}_8 (\underline{\mathbf{x}}_b^{ej} \dot{\underline{\mathbf{x}}}_{ei}^b) \\
&= \bar{\mathbf{H}}(\underline{\mathbf{x}}_{ei}^b) \text{vec}_8 (\dot{\underline{\mathbf{x}}}_b^{ej}) + \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_b^{ej}) \text{vec}_8 \dot{\underline{\mathbf{x}}}_{ei}^b \\
&= \bar{\mathbf{H}}(\underline{\mathbf{x}}_{ei}^b) C_8 J_{ej}^b \dot{q}_j + \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_b^{ej}) J_{ei}^b \dot{q}_i \\
&= \underbrace{\left\{ \left[\overset{+}{\mathbf{H}}(\underline{\mathbf{x}}_b^{ej}) J_{ei}^b \quad \vdots \quad 0_{8 \times n_j} \right] + \left[0_{8 \times n_i} \quad \vdots \quad \bar{\mathbf{H}}(\underline{\mathbf{x}}_{ei}^b) C_8 J_{ej}^b \right] \right\}}_{J_{ei}^{ej}} \dot{q}_{iej} \quad (4.5)
\end{aligned}$$

where $q_{iej} = \begin{bmatrix} q_i^T & q_j^T \end{bmatrix}^T$, and J_{ei}^{ej} is the Jacobian matrix of the i -th limb with respect to $\mathcal{F}_{ej}$.

4.3 Kinematic Modeling of the Robot's Center of Mass

The robot's Center of Mass (CoM) is an important variable to be controlled and is directly related to the robot's balance. When a body is under an uniform gravitational field—the gravitational pull near the surface of the Earth—its CoM will be coincident with its Center of Gravity, which is a term also adopted in related works. Even though the CoM is a dynamic parameter, we can define an equivalent kinematic model to estimate its position regarding the current robot's joints configuration.

Assuming that the robot's links are rigid, their CoM can be determined by their geometry and mass distribution. In order to simplify, we can consider that the links are symmetric and have a spatially uniform mass distribution, thus the CoM will be located in the geometric center of the link. Under those circumstances, the estimated CoM of the l -th link is located at half distance between the two consecutive joints l and $(l - 1)$, and the estimated CoM of the last link of a limb is located at half distance between the last joint and the limb end-effector. Figure 4.2 illustrates the estimated CoM of a humanoid robot and its links. As one may notice, the subscripts of the CoMs of the links are not in sequence in this figure, since the origin of the frames defined to some joints are coincident, and the links between them were disregarded in this illustration.

The aim of the kinematic modeling of the robot's CoM is to express the relation between the robot's CoM linear velocity and its joints velocities. The method adopted to obtain the kinematic model of the robot's CoM is similar to the proposed by Choi et al. (2007), and consists of obtaining equivalent FKMs for the CoM of each link.

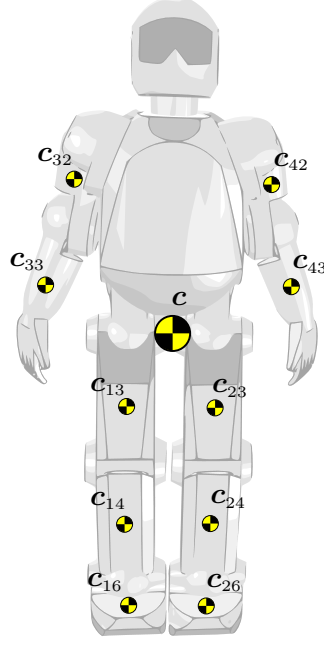


Figure 4.2: Estimated CoM of robot's links.

The FKM of each link CoM is given by

$$\mathbf{c}_{il}^i = \mathbf{f}(q_{il}),$$

where $\mathbf{c}_{il}^i$ is the CoM of the l -th link located at the i -th limb with respect to the frame $\mathcal{F}_i$, and $q_{il} \in \mathbb{R}^{n_{il}}$ is the vector of the joints that have an effect on this link. Recalling the vec_4 operator defined in (3.3), and the position Jacobian matrix derived in (3.29), the DFKM for this link CoM is given by

$$\text{vec}_4 \dot{\mathbf{c}}_{il}^i = J_{cil}^i \dot{q}_{il},$$

where $J_{cil}^i \in \mathbb{R}^{4 \times n_{il}}$ is the CoM Jacobian matrix expressed in frame $\mathcal{F}_i$.

The CoM of a specific link is known regarding the frame attached to its respective limb-base. However, in order to obtain the robot's global CoM, the CoM of all links must be expressed in a common frame, which we defined previously as $\mathcal{F}_b$. Using Example 3.1, the CoM of the l -th link in the i -th limb expressed with respect to $\mathcal{F}_b$ is given by

$$\mathbf{c}_{il}^b = \mathbf{p}_{bi}^b + \mathbf{r}_i^b \mathbf{c}_{il}^i \mathbf{r}_b^i.$$

Recalling the relation defined in (3.5), and assuming that $\mathbf{p}_{bi}^b$ is a rigid transformation that

does not change with the limbs motion, the DFKM is derived as follows:

$$\begin{aligned}
 \text{vec}_4 \dot{\mathbf{c}}_{il}^b &= \text{vec}_4 (\mathbf{r}_i^b \mathbf{c}_{il}^i \mathbf{r}_b^i) \\
 &= \mathbf{H}_4^+(\mathbf{r}_i^b) \bar{\mathbf{H}}_4(\mathbf{r}_i^{b*}) \text{vec}_4 \dot{\mathbf{c}}_{il}^i \\
 &= \underbrace{\mathbf{H}_4^+(\mathbf{r}_i^b) \bar{\mathbf{H}}_4(\mathbf{r}_i^{b*}) J_{cil}}_{J_{cil}^b} \dot{q}_{il},
 \end{aligned}$$

where J_{cil}^b is the CoM Jacobian matrix expressed in frame $\mathcal{F}_b$.

Given that the mass of each link is concentrated on its CoM, the computation of the robot's global CoM consists in obtaining the CoM of a system of particles. Let M be the robot's total mass and m_{il} be the mass of l -th link in the i -th limb. Thus, the robot's global CoM with respect to $\mathcal{F}_b$ is given by

$$\mathbf{c}^b = \frac{1}{M} \sum_{i=1}^n \sum_{l=1}^{k(i)} m_{il} \mathbf{c}_{il}^b, \quad (4.6)$$

where n is the number of limbs and $k(i)$ is the number of links in the i -th limb. The DFKM for the robot's global CoM is derived as follows:

$$\begin{aligned}
 \text{vec}_4 \dot{\mathbf{c}}^b &= \text{vec}_4 \left(\frac{1}{M} \sum_{i=1}^n \sum_{l=1}^{k(i)} m_{il} \dot{\mathbf{c}}_{il}^b \right), \\
 &= \frac{1}{M} \sum_{i=1}^n \sum_{l=1}^{k(i)} m_{il} \text{vec}_4 \dot{\mathbf{c}}_{il}^b, \\
 &= \underbrace{\frac{1}{M} \sum_{i=1}^n \sum_{l=1}^{k(i)} m_{il} \hat{J}_{cil}^b}_{J_{com}^b} \dot{q}_{wb}.
 \end{aligned} \quad (4.7)$$

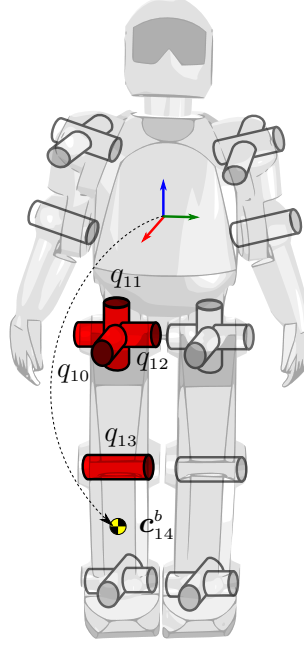
where $q_{wb} = [q_1^T \ q_2^T \ \dots \ q_n^T]^T$ is the whole-body joints angles vector, and $\hat{J}_{cil}^b$ is the position Jacobian matrix relating the whole-body joints configuration and the CoM linear velocity of l -th link in the i -th limb. The CoM at this link with respect to $\mathcal{F}_b$ is affected by joints $q_{i0}, q_{i1}, \dots$, and $q_{i(l-1)}$, which represent all joints anterior to the l -th link, and located at the i -th joint. Figure (4.3) illustrates this fact, where we highlight the joints that influence $\mathbf{c}_{14}^b$, i.e. q_{10}, q_{11}, q_{12} , and q_{13} .

Given these points, $\hat{J}_{cil}^b$ is defined as

$$\hat{J}_{cil}^b \triangleq \begin{bmatrix} 0_{4 \times n_{Ail}} & \vdots & J_{cil}^b & \vdots & 0_{4 \times n_{Pil}} \end{bmatrix},$$

where $n_{Ail} = \sum_{a=1}^{i-1} k(a)$ represent the number of joints in all limbs anterior to the i -th limb, and $n_{Pil} = \sum_{a=i}^n k(a) - (l-1)$ represents the number of joints posterior to the l -th link in the i -th limb, added to the number of joints in all limbs posterior to the i -th limb.

Given the dual quaternion $\underline{\mathbf{x}}_{ej}^b = \mathbf{r}_{ej}^b + \frac{1}{2} \varepsilon \mathbf{p}_{ej}^b \mathbf{r}_{ej}^b$ obtained from (4.2), the robot's global

Figure 4.3: Joints influence in the CoM $\mathbf{c}_{14}^b$.

CoM expressed in the current reference frame $\mathcal{F}_{ej}$ is given by

$$\mathbf{c}^{ej} = \mathbf{p}_b^{ej} + \mathbf{r}_b^{ej} \mathbf{c}^b \mathbf{r}_{ej}^b,$$

Let $n_{Aj} = \sum_{a=1}^{j-1} k(a)$ be the number of joints located in all limbs anterior to limb j , and $n_{Pj} = \sum_{a=j+1}^n k(a)$ be the number of joints located in all limbs posterior to limb j , and J_{b-pos}^{ej} and J_{b-ori}^{ej} be the position and orientation Jacobian matrices related to $J_b^{ej} = C_8 J_{ej}^b$, and J_{ej-ori}^b the orientation Jacobian matrix related to J_{ej}^b , derived in (3.29) and (3.28), respectively. Using the relations defined in (3.5), (3.7), and (4.7), the DFKM of the robot's global CoM expressed in $\mathcal{F}_{ej}$ is derived as follows:

$$\begin{aligned} \text{vec}_4 \dot{\mathbf{c}}^{ej} &= \text{vec}_4 (\dot{\mathbf{p}}_b^{ej} + \dot{\mathbf{r}}_b^{ej} \mathbf{c}^b \mathbf{r}_{ej}^b + \mathbf{r}_b^{ej} \dot{\mathbf{c}}^b \mathbf{r}_{ej}^b + \mathbf{r}_b^{ej} \mathbf{c}^b \dot{\mathbf{r}}_{ej}^b), \\ &= \text{vec}_4 (\dot{\mathbf{p}}_b^{ej}) + \bar{\mathbf{H}}_4 (\mathbf{c}^b \mathbf{r}_{ej}^b) \text{vec}_4 (\dot{\mathbf{r}}_b^{ej}) + \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej}) \bar{\mathbf{H}}_4 (\mathbf{r}_{ej}^b) \text{vec}_4 \dot{\mathbf{c}}^b + \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej} \mathbf{c}^b) \text{vec}_4 \dot{\mathbf{r}}_{ej}^b, \\ &= J_{b-pos}^{ej} \dot{q}_j + \bar{\mathbf{H}}_4 (\mathbf{c}^b \mathbf{r}_{ej}^b) J_{b-ori}^{ej} \dot{q}_j + \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej}) \bar{\mathbf{H}}_4 (\mathbf{r}_{ej}^b) J_{com}^b \dot{q}_{wb} + \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej} \mathbf{c}^b) J_{ej-ori}^b \dot{q}_j, \\ &= \left\{ \begin{bmatrix} 0_{4 \times n_{Aj}} & | & J_{b-pos}^{ej} + A + B & | & 0_{4 \times n_{Pj}} \end{bmatrix} + C \right\} \dot{q}_{wb}, \\ &= J_c^{ej} \dot{q}_{wb}, \end{aligned} \tag{4.8}$$

where $A = \bar{\mathbf{H}}_4 (\mathbf{c}^b \mathbf{r}_{ej}^b) J_{b-ori}^{ej}$, $B = \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej} \mathbf{c}^b) J_{ej-ori}^b$, and $C = \bar{\mathbf{H}}_4^+ (\mathbf{r}_b^{ej}) \bar{\mathbf{H}}_4 (\mathbf{r}_{ej}^b) J_{com}^b$.

Being $\mathbf{c}^{ej} = c_x \hat{i} + c_y \hat{j} + c_z \hat{k}$, consequently, $\text{vec} \dot{\mathbf{c}}^{ej} = \begin{bmatrix} 0 & \dot{c}_x & \dot{c}_y & \dot{c}_z \end{bmatrix}^T$. Thus, (4.8) can be simplified as follows:

$$\begin{bmatrix} \dot{c}_x \\ \dot{c}_y \\ \dot{c}_z \end{bmatrix} = J_{com} \dot{q}_{wb},$$

where J_{com} is obtained removing the first row of matrix J_c^{ej} . The superscript ej will be omitted in this matrix for the sake of simplicity. The matrix J_{com} can be separated in two parts

$$J_{com} = \begin{bmatrix} J_{com_{UB}} & J_{com_{LB}} \end{bmatrix}, \quad (4.9)$$

where $J_{com_{UB}}$ is related to the arms joints and $J_{com_{LB}}$ to the legs joints.

4.4 Legs Coordinated Behavior Modeling

A humanoid is a biped robot whose locomotion is performed by the coordination of its legs. In order to obtain a kinematic model for the coordinated behavior between the legs, we apply the CDTS presented in Section 3.4, which allows the representation of the geometric relation between the feet regarding the legs-joints configuration.

Consider the poses of the feet, expressed with respect to the reference frame, represented by the dual quaternions $\underline{x}_{e1}^{ej}$ and $\underline{x}_{e2}^{ej}$, as shown in Fig. 4.4.

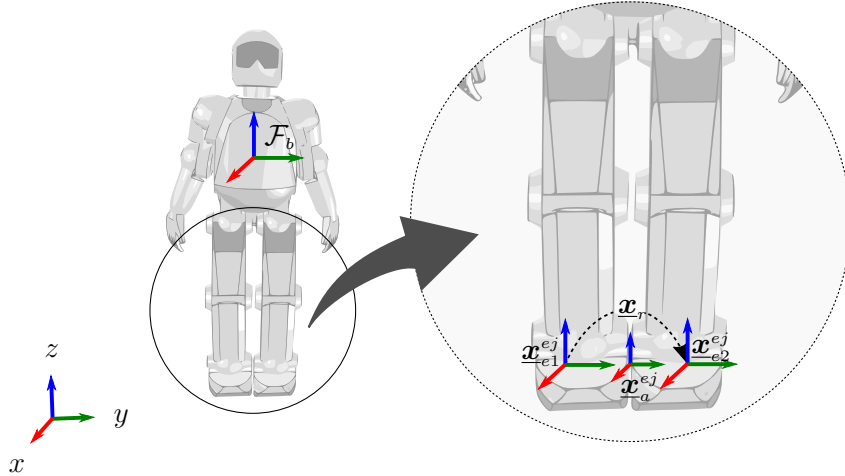


Figure 4.4: Cooperative variables of the robot's legs.

Firstly, we desire to obtain the geometric relation between the swinging-foot and the supporting-foot. This geometric relation can be captured by the relative variable $\underline{x}_r$, which is computed according to (3.30). More specifically, $\underline{x}_r$ is computed as

$$\underline{x}_r = (\underline{x}_{e1}^{ej})^* \underline{x}_{e2}^{ej}. \quad (4.10)$$

The relative pose does not have a superscript, since we assume it represents a geometric relation, not a pose with respect to a frame.

According to (3.32), the DFKM for the relative pose can be obtained by

$$\text{vec}_8 \dot{\underline{x}}_r = J_r \dot{q}_{legs},$$

where $q_{legs} = \begin{bmatrix} q_1^T & q_2^T \end{bmatrix}^T$, with q_1 and q_2 representing the joints vectors of the respective legs, and J_r is the relative Jacobian matrix, calculated as

$$J_r = \begin{bmatrix} \bar{\mathbf{H}}(\underline{\mathbf{x}}_{e2}^{ej}) C_8 J_{e1}^{ej} & \mid & \mathbf{H}^+(\underline{\mathbf{x}}_{e1}^{ej*}) J_{e2}^{ej} \end{bmatrix}. \quad (4.11)$$

The robot's body posture is also an important variable to be modeled, as better explained in Section 5.1, and it can be defined as the orientation of the torso with respect to the gravitational vector. If we represent the absolute pose with respect to the frame attached to $\mathcal{F}_b$, we have the geometric relation between the robot's torso and the feet. Assuming that the feet are always flat on the floor, the orientation of the absolute pose with respect to $\mathcal{F}_b$ captures the robot's body posture.

According to (3.31), the absolute pose for the biped robot with respect to the reference frame $\mathcal{F}_{ej}$ is obtained by

$$\underline{\mathbf{x}}_a^{ej} = \underline{\mathbf{x}}_{e1}^{ej} \underline{\mathbf{x}}_r^{\{1/2\}}. \quad (4.12)$$

Since the coordinate transformation between $\mathcal{F}_{ej}$ and $\mathcal{F}_b$ is given by (4.2), then, the absolute pose with respect to $\mathcal{F}_b$ is computed as

$$\underline{\mathbf{x}}_a^b = \underline{\mathbf{x}}_{ej}^b \underline{\mathbf{x}}_a^{ej}.$$

Let n_1 be the dimension of vector q_1 and n_2 be the dimension of vector q_2 . Using the absolute Jacobian matrix derived in (3.33), and the relation defined in (3.5), the DFKM of the absolute variable with respect to $\mathcal{F}_b$ is derivated as follows:

$$\begin{aligned} \text{vec}_8 \dot{\underline{\mathbf{x}}}_a^b &= \text{vec}_8 (\dot{\underline{\mathbf{x}}}_{ej}^b \underline{\mathbf{x}}_a^{ej} + \underline{\mathbf{x}}_{ej}^b \dot{\underline{\mathbf{x}}}_a^{ej}) \\ &= \bar{\mathbf{H}}(\underline{\mathbf{x}}_a^{ej}) \text{vec}_8 \dot{\underline{\mathbf{x}}}_{ej}^b + \mathbf{H}^+(\underline{\mathbf{x}}_{ej}^b) \text{vec}_8 \dot{\underline{\mathbf{x}}}_a^{ej} \\ &= \bar{\mathbf{H}}(\underline{\mathbf{x}}_a^{ej}) J_{ej}^b \dot{q}_j + \mathbf{H}^+(\underline{\mathbf{x}}_{ej}^b) J_a^{ej} \dot{q}_{legs}. \end{aligned}$$

This DFKM varies depending on the current supporting-foot, i.e. the value of j . Given $j=1$, the DFKM is given by

$$\text{vec}_8 \dot{\underline{\mathbf{x}}}_a^b = \underbrace{\left\{ \left[\bar{\mathbf{H}}(\underline{\mathbf{x}}_a^{e1}) J_{e1}^b \mid 0_{8 \times n_2} \right] \right\}}_{J_a^b} + \mathbf{H}^+(\underline{\mathbf{x}}_{e1}^b) J_a^{e1} \dot{q}_{legs}, \quad (4.13)$$

otherwise, the DFKM is defined as

$$\text{vec}_8 \dot{\underline{\mathbf{x}}}_a^b = \underbrace{\left\{ \left[0_{8 \times n_1} \mid \bar{\mathbf{H}}(\underline{\mathbf{x}}_a^{e2}) J_{e2}^b \right] \right\}}_{J_a^b} + \mathbf{H}^+(\underline{\mathbf{x}}_{e2}^b) J_a^{e2} \dot{q}_{legs}. \quad (4.14)$$

4.5 Chapter Overview

This chapter presented the kinematic modeling method for a humanoid robot, which consisted of three stages: the limbs modeling, the CoM modeling, and the modeling of the legs coordinated behavior.

Section 4.1 showed the method to obtain the FKM of each link using DQ algebra, where the DH convention was used to define the limbs geometric parameters. Furthermore, the end-effectors poses were represented with respect to a common frame $\mathcal{F}_b$.

Section 4.2 defined the system's reference frame during the walking motion, which is coincident with the pose of the supporting foot. Then, the limbs end-effectors are represented in the reference frame.

Section 4.3 detailed the method to obtain the kinematically equivalent model of the CoM, where an equivalent model for the CoM of each link was used to calculate the global CoM model.

Finally, Section 4.4 presented the CDTS applied to the robot's legs, and the cooperative variables were calculated, which describe the relation between the feet, and can represent the kinematic constraints of locomotion in a compact way.

5

Gait and Balance Control Strategies

This chapter discusses the gait control of the humanoid robot. The chapter is organized as follows: Section 5.1 presents the stability conditions for a balanced gait; in Section 5.2 the methods to obtain the reference trajectories are explained; in Section 5.3 the control strategies are developed; and in Section 5.4 the stability proof of these controllers is derived.

5.1 Stability Conditions

In the context of humanoid robots, balance is a term to describe the dynamics of the robot's posture to prevent falling, as defined by Winter (1995). Herein we consider that the robot is stable if it is balanced.

An important concept to understand the stability conditions of biped robots locomotion is the “support polygon.” They are stable regions characterized by the convex hull of the feet contact-points with the ground. Figure 5.1 illustrates this concept, where the support polygon of a humanoid robot is highlighted in three different views.

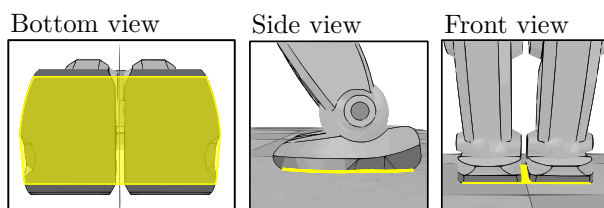


Figure 5.1: Support polygon of a humanoid robot.

As mentioned in Chapter 1, robot locomotion can be classified into quasi-static locomotion and dynamic locomotion. The stability condition for quasi-static locomotion is that the CoM projection on the ground (GCoM) must lie within the support polygon. The dynamic stability condition does not require that the GCoM be within the support polygon, but the Zero Moment Point (ZMP)¹ must be. This condition guarantees that rotations around the feet edges will be neglected, consequently the feet soles do not detach from the ground, which can cause a loss of balance.

As stated in Section 2.3, we assume a quasi-static locomotion in this work. In this case, the walking speed must be low, the feet sizes must be large and the robot must have strong ankle joints to allow that the effects of inertial forces be negligible. The trajectories that guarantee the robot's stability are shown in Section 5.2.

5.2 Reference Trajectories

The gait and balance control for the quasi-static locomotion must be performed in such a way that the CoM must stay within the support polygon during the whole gait cycle. In order to guarantee a stable gait cycle, the CoM and the feet reference trajectories must be carefully formulated to fulfill the stability condition.

Assuming that the robot's dynamic parameters are not precisely known, the walking pattern can be generated by a method based on the inverted pendulum model, as mentioned in Section 2.2. Herein, the method chosen to generate the reference trajectories is the one proposed by Kajita et al. (2003), which is based on the inverted pendulum model and uses a ZMP reference trajectory, which can be regarded as the central points of the desired footprints. Even though this method has a high complexity involved in its implementation and understanding, which may be unnecessary in this work, we consider that the method has great relevance in the context of biped robots. Since this work is an initial stage of a larger project, we believe that using a well-consolidated methodology pays off the complexity.

In the following subsection, the walking pattern generation method is detailed.

5.2.1 Walking Pattern Generation Method

A biped robot can be modeled according to the Three-Dimensional Linear Inverted Pendulum Mode (3DLIPM), as proposed by Kajita et al. (2003). In this model, the robot dynamics is simplified by a cart-table model, which consists of a pedestal table whose mass is negligible, and a running cart of mass m moving on its surface, representing the robot's global CoM. The cart-table model is illustrated in Figure 5.2.

¹Recalling from Section 2.2, the ZMP is defined as a point where the moments around the x -axis and the y -axis (parallel to the flat ground) generated by reaction forces and reaction torques are zero.

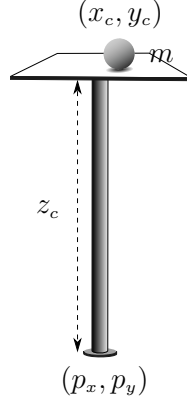


Figure 5.2: Cart-table model.

The table's foot is smaller than the table itself, and according to the static stability condition, the table keeps upright if the cart stay inside the support polygon. On the other hand, the dynamic condition for stability states that the ZMP must be inside the support polygon. If the cart accelerates with a proper rate, then the ZMP stays inside the edges of the table foot, thus the table keeps upright for a while. Applying this idea in the biped robot locomotion, we must calculate the proper CoM acceleration for a given ZMP trajectory so as to keep the robot balanced.

In the cart-table model, the mass should move along a constraint plane. Considering that the constraint plane is horizontal, the dynamics of this system is represented by (Kajita et al., 2001)

$$\ddot{y}_c = \frac{g}{z_c}(y_c - p_y), \quad (5.1)$$

$$\ddot{x}_c = \frac{g}{z_c}(x_c - p_x), \quad (5.2)$$

where (x_c, y_c) is the location of the cart on the table, (p_x, p_y) is the location of the ZMP on the floor, g is the gravitational acceleration, and z_c is the distance between the table and the floor, which is a constraint parameter. In our case, z_c is a constant value, calculated by the distance between the CoM and the floor at the beginning of the gait. In order to control the ZMP, p_x and p_y must be outputs of the system. Thus, (5.1) and (5.2) are rewritten as

$$p_y = y_c - \frac{z_c}{g}\ddot{y}_c,$$

$$p_x = x_c - \frac{z_c}{g}\ddot{x}_c.$$

The ZMP reference trajectory is determined by the desired footprints trajectory, and is represented by (p_x^{ref}, p_y^{ref}) . The walking pattern generation method proposed by Kajita et al. (2003) consists of generating the CoM trajectory such that the resulting ZMP trajectory follows the given reference. The ZMP changes suddenly during the gait cycle, and can be

modeled with a step function. However, the CoM motion must be smooth so as to avoid large accelerations of the robot's body. The CoM reference in the current time instant must be calculated also considering future inputs of the ZMP. To accomplish this, Kajita et al. (2003) proposes a preview control² method, as explained in the sequel.

In this method, new variables u_x and u_y are defined as follows:

$$\frac{d}{dt}\ddot{x}_c = u_x,$$

$$\frac{d}{dt}\ddot{y}_c = u_y,$$

and the relation between the ZMP and the CoM can be modeled through a state space representation, as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \\ \ddot{x}_c \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ \dot{x}_c \\ \ddot{x}_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \\ p_x &= \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x_c \\ \dot{x}_c \\ \ddot{x}_c \end{bmatrix}, \end{aligned}$$

and similarly for y_c and its time derivatives.

The system is discretized with a sampling time T . For the x -axis, the discretized system is given by

$$\begin{aligned} x(k+1) &= Lx(k) + Mu_x(k), \\ p_x(k) &= Nx(k), \\ x(k) &\triangleq \begin{bmatrix} x_c(kT) \\ \dot{x}_c(kT) \\ \ddot{x}_c(kT) \end{bmatrix}, \\ u_x(k) &\triangleq u_x(kT), \\ p_x(k) &\triangleq p_x(kT), \end{aligned}$$

where

$$L = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, M = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, N = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix}.$$

²Preview control is one of the precursors theories of the predictive control theory, and this is the terminology defined in the mentioned work and, consequently, used herein.

For the y -axis, the reasoning is analogous.

A performance index Ψ_x is defined as

$$\Psi_x = \sum_{i=k}^{\infty} \{Q_e e_x(i)^2 + \Delta x^T(i) Q_x \Delta x(i) + R \Delta u_x^2(i)\}, \quad (5.3)$$

where $e_x(i) = p_x(i) - p_x^{ref}(i)$, $\Delta x(i) = x(k) - x(k-1)$ and $\Delta u_x(i) = u_x(k) - u_x(k-1)$. The weighting matrices are such that $Q_e > 0$, $R > 0$ and $Q_x \geq 0$. The index Ψ_y regarding y -axis is obtained equivalently, being $Q_y = Q_x$.

Considering a preview time window, or a prediction horizon, w_L , the first control actions values from the control histories that minimize Ψ_x and Ψ_y are given by

$$u_x(k) = -G_I \sum_{i=0}^k e_x(i) - G_x x(k) - \sum_{j=1}^{N_L} G_p(j) p_x^{ref}(k+j) \quad (5.4)$$

$$u_y(k) = -G_I \sum_{i=0}^k e_y(i) - G_y y(k) - \sum_{j=1}^{N_L} G_p(j) p_y^{ref}(k+j) \quad (5.5)$$

where $N_L = w_L/T$ is the number of points of the ZMP reference trajectory used in the control, and G_I , $G_x = G_y$, and $G_p(j)$ are the gains of integral action, state feedback, and preview action, respectively. These gains are obtained solving an algebraic Riccati equation, as described in Katayama et al. (1985). They define the matrices $\tilde{M}$, $\tilde{I}$, $\tilde{F}$, $\tilde{Q}$, and $\tilde{L}$, which are given by

$$\tilde{M} \triangleq \begin{bmatrix} NM \\ M \end{bmatrix}, \quad \tilde{I} \triangleq \begin{bmatrix} 1 \\ M \end{bmatrix}, \quad \tilde{F} \triangleq \begin{bmatrix} NL \\ L \end{bmatrix}, \quad \tilde{Q} \triangleq \begin{bmatrix} Q_e & 0 \\ 0 & Q_x \end{bmatrix}, \quad \tilde{L} \triangleq \begin{bmatrix} \tilde{I} & \tilde{F} \end{bmatrix}.$$

The matrix $\tilde{K} \geq 0$ is the solution of the following discrete-time algebraic Riccati equation

$$\tilde{K} = \tilde{L}^T \tilde{K} \tilde{L} - \tilde{L}^T \tilde{K} \tilde{M} [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{L} + \tilde{Q}. \quad (5.6)$$

We use the algorithm proposed by Arnold & Laub (1984), implemented in the Matlab function “*dare*,” to solve this algebraic Riccati equation. Thus, it is possible to compute the gains G_I , G_x , and G_p using the following relations

$$G_I = [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{I}, \quad (5.7)$$

$$G_x = [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{F}. \quad (5.8)$$

Note that $G_y = G_x$. The gain G_p depends on the chosen preview time window, and is

recursively calculated as follows

$$\begin{aligned} G_p(1) &= -G_1, \\ G_p(j) &= [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{X}(j-1), \end{aligned}$$

where $\tilde{X}$ is given by

$$\begin{aligned} \tilde{X}(1) &= -L_c^T \tilde{K} \tilde{I}, \\ \tilde{X}(j) &= L_c^T \tilde{X}(j-1), \end{aligned}$$

with $L_c = \tilde{L} - \tilde{M}[R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{L}$ representing the closed-loop matrix of the system. Algorithm 5.1 summarizes this method in a pseudocode.

Algorithm 5.1 Pseudocode of the preview-control method.

Require: $w_L, T, z_c, Q_e, Q_x, R, p_x^{ref}, p_y^{ref}, Num_iterations$

- 1: $L, M, N \leftarrow Calculate_Basic_Matrices(T, z_c);$
- 2: $\tilde{L}, \tilde{M}, \tilde{I}, \tilde{F}, \tilde{Q} \leftarrow Calculate_Ricatti_Matrices(L, M, N, Q_e, Q_x);$
- 3: $\tilde{K} \leftarrow dare(\tilde{L}, \tilde{M}, \tilde{Q}, R);$
- 4: $G_i \leftarrow [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{I};$
- 5: $G_x \leftarrow [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{F};$
- 6: $G_y \leftarrow G_x;$
- 7: $L_c \leftarrow \tilde{L} - \tilde{M} [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{K} \tilde{L};$
- 8: $\tilde{X}(1) \leftarrow -L_c^T \tilde{K} \tilde{I};$
- 9: $N_L \leftarrow w_L/T;$
- 10: **while** $j < N_L$ **do**
- 11: $\tilde{X}(j) \leftarrow -L_c^T \tilde{X}(j-1);$
- 12: $G_p(j) \leftarrow [R + \tilde{M}^T \tilde{K} \tilde{M}]^{-1} \tilde{M}^T \tilde{X}(j-1);$
- 13: **end while**
- 14: $k \leftarrow 0;$
- 15: $x(0) \leftarrow 0;$
- 16: $y(0) \leftarrow 0;$
- 17: **while** $k < Num_iterations$ **do**
- 18: $p_x(k) \leftarrow Nx(k);$
- 19: $p_y(k) \leftarrow Ny(k);$
- 20: $e_x(k) \leftarrow p_x(k) - p_x^{ref}(k);$
- 21: $e_y(k) \leftarrow p_y(k) - p_y^{ref}(k);$
- 22: $u_x(k) \leftarrow -G_i \sum_{i=1}^k e_x(i) - G_x x(k) - \sum_{j=1}^{N_L} G_p(j) p_x^{ref}(k+j);$
- 23: $u_y(k) \leftarrow -G_i \sum_{i=1}^k e_y(i) - G_y y(k) - \sum_{j=1}^{N_L} G_p(j) p_y^{ref}(k+j);$
- 24: $x(k+1) \leftarrow Lx(k) + Mu_x(k);$
- 25: $y(k+1) \leftarrow Ly(k) + Mu_y(k);$
- 26: **end while**

return x, y, p_x, p_y

Example 5.1. Considering $Q_e = 1$, $R = 10^{-6}$ and $Q_x = 0$ (which applies for both, x -axis and y -axis), the dependence of the preview gain G_p on the length of the preview time

window w_L is depicted in Figure 5.3a. We can notice that giving information of far future is not needed, since the preview gain becomes very small for a choice of w_L larger than 1.5s. Thus, choosing $w_L = 1.5$ s, a ZMP trajectory defined by a number of footsteps $N_s = 6$, the step duration $t_s = 1$ s, and a constant step length of 0.2m towards x -axis positive direction, the generated CoM trajectory is depicted in Figure 5.3b.

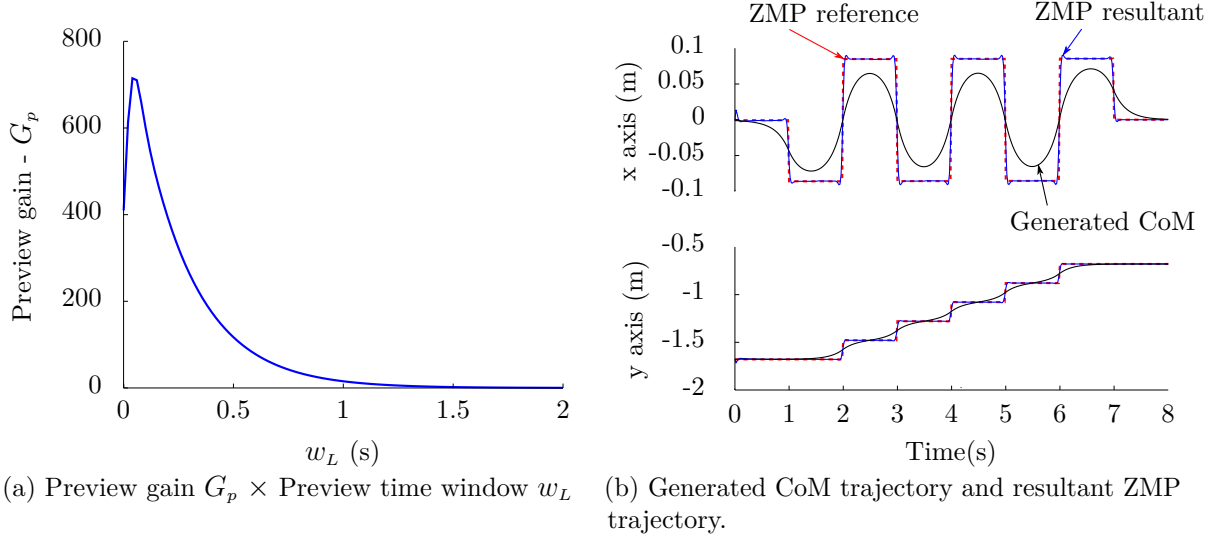


Figure 5.3: Results of the walking pattern generation method.

The parameter Q_x is related to the smoothness of the the state-space system's output (the resulting ZMP trajectory defined by (p_x, p_y)), such that the greater the value of Q_x , the smoother is the output. Figure (5.4) shows the comparison between the resultant ZMP trajectories for $Q_x = 0$ and $Q_x = I_{3 \times 3}$, where $I_{3 \times 3} \in \mathbb{R}^{3 \times 3}$ is the identity matrix. As one may notice, the trajectory is smoother in the second case, as expected. Since one desires that the ZMP follows the defined footsteps, it must change suddenly whenever the supporting-foot switches, and must not be defined by a smooth trajectory. Thus, the value of Q_x is set equal to zero, as shown in the Example (5.1).

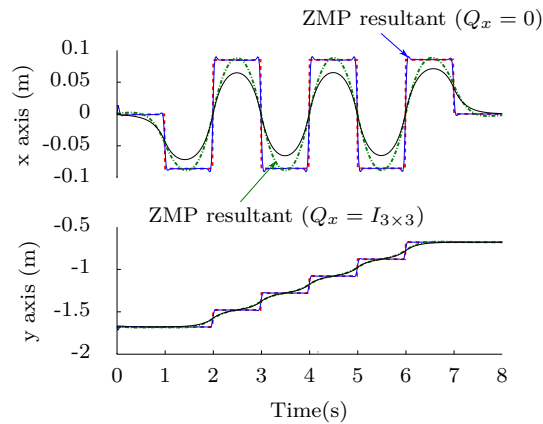


Figure 5.4: Comparison of the ZMP Resultant for different values of Q_x .

The CoM trajectory generated by this method consists of a set of points parameterized in time, given by (x_c, y_c) . In order to calculate the CoM reference velocity, we approximated the trajectory by a piecewise polynomial form of the cubic spline interpolated data, and obtained the first derivative with respect to time for this approximated function.

5.2.2 Feet Trajectories

In the walking pattern generation method, the desired footprints are defined in order to approximate the ZMP trajectory, which is used to generate the CoM trajectory. These footprints are also used as a reference to generate the trajectory of the swinging-foot in each step, while the supporting-foot is kept stationary.

The swinging-foot path is determined by a Bézier curve, as proposed by Alcaraz-Jimenez et al. (2013). The reason for this choice is that Bézier curves provide flexibility to generate complex curves, allowing different forms, depending on the choice of its parameters. A Bézier curve of order n with respect to time k is defined by $n + 1$ points, and is given by

$$\mathbf{b}(k) = \sum_{i=0}^n \mathbf{p}_i B_i^n(k),$$

where $\mathbf{b}(k)$ and $\mathbf{p}_i$ are pure quaternions, and $\mathbf{p}_i$ represents the defined curve points. Moreover, $B_i^n(k)$ is a Bernstein polynomial of order n , computed by

$$B_i^n(k) = \frac{n!}{(n-k)!k!} k^i (1-k)^{n-i}.$$

The swinging-foot path is determined by a quadratic Bézier curve, and the three points which characterize this curve are depicted in Figure 5.5. The points $\mathbf{p}_0$ and $\mathbf{p}_2$ are the start position and the end position of the step regarding the reference frame, respectively, which we assume to be known a priori. Moreover, $\mathbf{p}_1$ is defined by a parameter d_h associated with the desired step height, and is calculated by $\mathbf{p}_1 = \mathbf{p}_2 + d_h \hat{k}$.

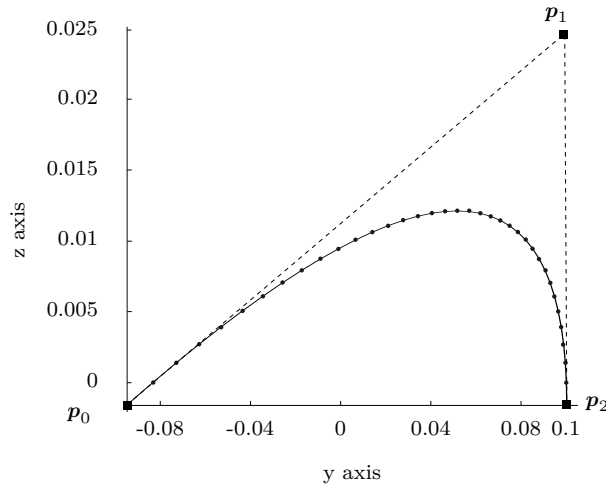


Figure 5.5: Example of a cubic Bézier curve.

The quadratic Bézier curve is written as

$$\mathbf{b}(k) = \mathbf{p}_0(1 - k)^2 + \mathbf{p}_1 2k(1 - k) + \mathbf{p}_2 k^2, \quad 0 \leq k \leq 1. \quad (5.9)$$

The first derivative of $\mathbf{b}$ with respect to k is given by

$$\frac{d\mathbf{b}(k)}{dk} = -\mathbf{p}_0 2(1 - k) + \mathbf{p}_1 2(1 - 2k) + 2\mathbf{p}_2 k, \quad 0 \leq k \leq 1,$$

which represents the reference velocity for the swinging-foot.

Let N_s be the total number of footsteps, and $t_0(i_s)$ and $t_s(i_s)$ be functions that provide the initial instant and the time duration of each step i_s , respectively, where $i_s \in \mathbb{N}$ and $i_s \leq N_s$. In order to obtain a curve parameterized in time to execute the footstep in a defined time period, we must substitute k by

$$\tau(t, i_s) = \frac{(t - t_0(i_s))}{t_s(i_s)} \quad (5.10)$$

in (5.9), where t is the current time instant.

Consider the step function $\mu(t)$, defined as

$$\mu(t) \triangleq \begin{cases} 1, & \text{if } t \geq 0, \\ 0, & \text{if } t < 0, \end{cases}$$

and the pulse function $\Pi_{\Delta t}(t)$, which is defined as

$$\Pi_{\Delta t}(t) \triangleq \mu(t) - \mu(t - \Delta t),$$

where Δt is the time duration of the pulse. Figure 5.6 illustrates an example of a pulse function with time duration of 2s, defined as $\Pi_2(t - 1)$, which is given by the summation of the step functions $\mu(t - 1)$ and $\mu(t - 3)$.

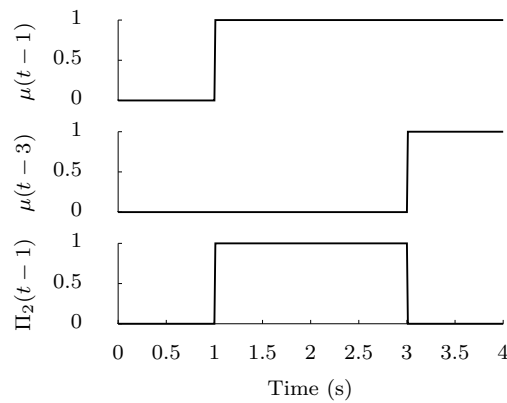


Figure 5.6: Pulse function example.

Let $\mathbf{b}_{i_s}(\tau(t, i_s))$ represents the trajectory of the swinging-foot in the i_s -th step, deter-

mined by the start and end positions of the step i_s and the step height d_h . We can represent a foot position trajectory over one gait cycle with the following expression:

$$\gamma(t, i_s) = \mathbf{b}_{i_s}(\tau(t, i_s)) \Pi_{t_s(i_s)}(t - t_0(i_s)) + \mathbf{b}_{i_s}(1) (1 - \Pi_{t_s(i_s)}(t - t_0(i_s))), \quad t_0(i_s) \leq t \leq t_0(i_s + 1) \quad (5.11)$$

where the first term represents the swing movement and the second term represents the stationary position of the foot after reaching the floor.

The first derivative of (5.11) with respect to time, neglecting the discontinuous transitions in $\Pi_{t_s(i_s)}$, is given by

$$\frac{\partial \gamma(t, i_s)}{\partial t} = \frac{d\mathbf{b}_{i_s}(\tau(t, i_s))}{dt} \frac{1}{t_s(i_s)} \Pi_{t_s(i_s)}(t - t_0(i_s)), \quad t_0(i_s) \leq t \leq t_0(i_s + 1).$$

As the supporting-foot changes at the time instant $t_0(i_s)$, we consider that $\partial \gamma / \partial t = 0$ when $t = t_0(i_s)$.

Based on the above mentioned considerations, the function representing the complete position trajectory of one foot is given by a summation of the trajectories calculated for each gait cycle.

$$\Gamma(t) = \sum_{i_s=1}^{N_s} \gamma(t, i_s), \quad (5.12)$$

$$\frac{d\Gamma(t)}{dt} = \sum_{i_s=1}^{N_s} \frac{\partial \gamma(t, i_s)}{\partial t}. \quad (5.13)$$

In summary, these trajectories depend on the desired footprints, the initial time instant, and the time duration of each footprint, that are defined a priori. Moreover, we can vary the step height by modifying the parameter d_h .

Example 5.2. For a step length of 0.2m towards the y -axis positive direction, $t_0 = 3s$, $t_s = 1s$, and $d_h = 0.05m$, the trajectory generated by the function $\Gamma(t)$, regarding an inertial frame, is depicted in Figure 5.7.

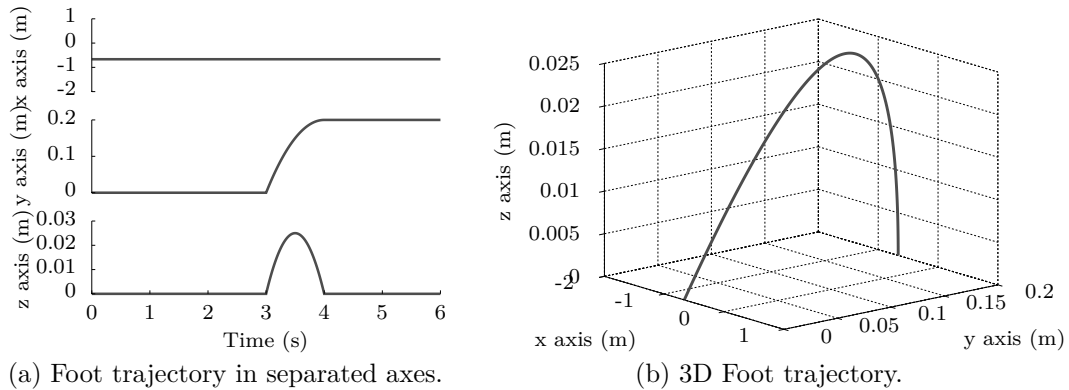


Figure 5.7: Trajectory generated by (5.11).

Example 5.3. Given the desired footprints trajectory defined by a number of steps $N_s = 6$, a constant step length of 0.2m towards the y -axis positive direction, $t_0(i_s) = i_s$, and $t_s(i_s) = 1s$, the complete position trajectories of the feet regarding an inertial frame, are illustrated in Figure 5.8. The linear velocities of the feet are represented in Figure 5.9.

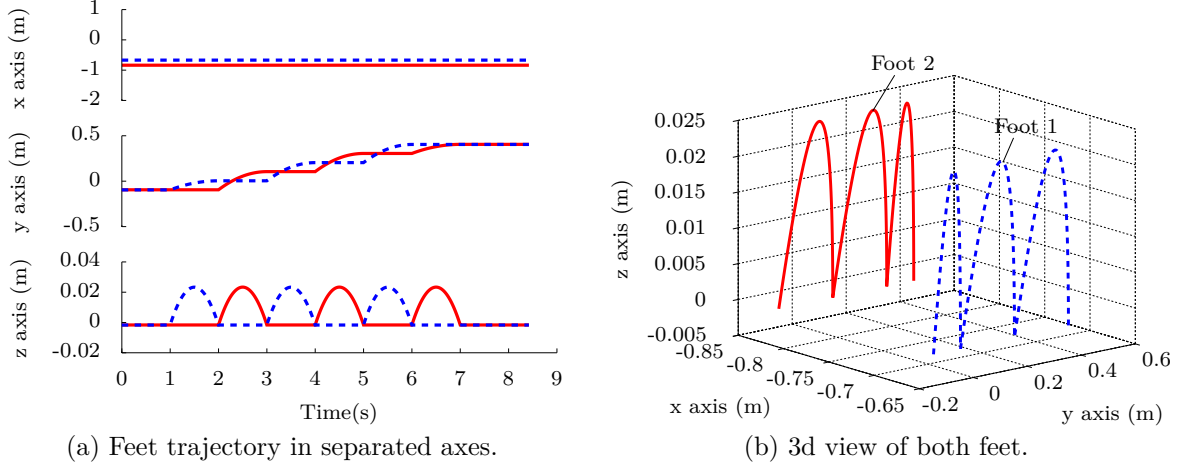


Figure 5.8: Complete position trajectories of the feet generated by (5.12). In this case, the robot walks towards $y > 0$ from a $y < 0$ initial position. The *dashed* blue line represents the left foot trajectory, the *solid* red line represents the right foot trajectory.

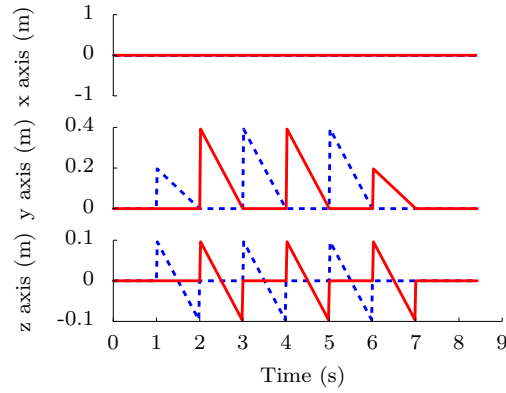


Figure 5.9: Linear velocities of the feet generated by (5.13). The *dashed* blue line represents the left foot trajectory, the *solid* red line represents the right foot trajectory.

In this work, we assume that the feet stays parallel to the ground plane during the entire movement. Thus, the orientation trajectory of one foot in one gait cycle is defined by a desired angular velocity $\dot{\theta}(t)$ around z -axis (normal to the ground), which can vary during the swing movement and is zero during the supporting phase. The orientation trajectory is

$$\mathbf{r}_f(t) = \cos\left(\frac{\theta(t)}{2}\right) + \hat{k} \sin\left(\frac{\theta(t)}{2}\right).$$

The complete trajectory of this foot, with respect to the reference frame, represented

by dual quaternions, is given by

$$\underline{\mathbf{x}}_f(t) = \mathbf{r}_f(t) + \frac{1}{2}\varepsilon\mathbf{\Gamma}(t)\mathbf{r}_f(t). \quad (5.14)$$

Taking the first derivative of (5.14) with respect to the time

$$\dot{\underline{\mathbf{x}}}_f(t) = \dot{\mathbf{r}}_f(t) + \frac{1}{2}\varepsilon\left(\dot{\mathbf{\Gamma}}(t)\mathbf{r}_f(t) + \mathbf{\Gamma}(t)\dot{\mathbf{r}}_f(t)\right),$$

where $\dot{\mathbf{r}}_f(t)$ represents the reference angular velocity of the foot, and is given by

$$\dot{\mathbf{r}}_f(t) = -\frac{\dot{\theta}(t)}{2}\sin\left(\frac{\theta(t)}{2}\right) + \hat{k}\frac{\dot{\theta}(t)}{2}\cos\left(\frac{\theta(t)}{2}\right).$$

Circular Trajectory

In order to obtain trajectories with varying direction, we define a circular path for the footprints, from which we obtain the CoM trajectory, using the method described in Subsection 5.2.1, and the feet trajectories, using the method described in Subsection 5.2.2.

Assuming that the initial poses of the feet are known and given by $\underline{\mathbf{x}}_{e1_i}$ and $\underline{\mathbf{x}}_{e2_i}$ with respect to the inertial frame, the initial absolute variable between the feet is calculated by

$$\underline{\mathbf{x}}_{a_i} = \underline{\mathbf{x}}_{e1_i}\underline{\mathbf{x}}_{r_i}^{\{1/2\}},$$

where $\underline{\mathbf{x}}_{r_i} = \underline{\mathbf{x}}_{e1_i}^*\underline{\mathbf{x}}_{e2_i}$. Being r_w the radius of the trajectory, and s_l the step length, the step angle ϕ_l is calculated by $\phi_l = \arctan(s_l/r_w)$. The absolute variable after each footstep i_s is obtained iteratively as follows:

$$\underline{\mathbf{x}}_a[i_s] = \underline{\mathbf{x}}_a[i_s - 1]\mathbf{r}_{\phi_l}\underline{\mathbf{h}}_{s_l},$$

where $\underline{\mathbf{x}}_a[0] = \underline{\mathbf{x}}_{a_i}$, $\mathbf{r}_{\phi_l} = \cos(\phi_l/2) + \hat{k}\sin(\phi_l/2)$, and $\underline{\mathbf{h}}_{s_l} = 1 + \varepsilon(1/2)(s_l\hat{j})$. Let d_f be the distance between the feet, and the right foot be the starting foot. Considering that the poses $\underline{\mathbf{x}}_{e1_i}$ and $\underline{\mathbf{x}}_{e2_i}$ have the y -axis pointed to the forward walking direction, the footprints path is obtained as follows:

$$\underline{\mathbf{x}}_{f1}[i_s] = \begin{cases} \underline{\mathbf{x}}_a[i_s]\underline{\mathbf{h}}_{p1}, & \text{if } i_l \text{ is odd,} \\ \underline{\mathbf{x}}_{f1}[i_s - 1], & \text{if } i_l \text{ is even,} \end{cases} \quad (5.15)$$

$$\underline{\mathbf{x}}_{f2}[i_s] = \begin{cases} \underline{\mathbf{x}}_a[i_s]\underline{\mathbf{h}}_{p2}, & \text{if } i_l \text{ is even,} \\ \underline{\mathbf{x}}_{f2}[i_s - 1] & \text{if } i_l \text{ is odd,} \end{cases} \quad (5.16)$$

where

$$\begin{aligned}\mathbf{h}_{p_1} &= 1 + \frac{1}{2}\varepsilon \left(-\frac{d_f}{2}\hat{i} + s_l\hat{j} \right), \\ \mathbf{h}_{p_2} &= 1 + \frac{1}{2}\varepsilon \left(\frac{d_f}{2}\hat{i} + s_l\hat{j} \right).\end{aligned}$$

Furthermore, the orientation trajectory for both feet in one gait cycle is given by

$$\mathbf{r}_f(t, i_s) = \cos\left(\frac{\phi_l\tau(t, i_s)}{2}\right) + \hat{k} \sin\left(\frac{\phi_l\tau(t, i_s)}{2}\right), \quad (5.17)$$

where $\tau(t, i_s)$ is defined in (5.10).

Note that, in the case of a varying walking direction, the robot's body orientation must change during the walking motion in order to keep the torso pointed in the walking forward direction. The body must rotate only around z -axis, following the rotation performed by the swinging foot during one gait cycle, given by

$$\mathbf{r}_b(t) = \cos\left(\frac{\phi_l N_s t}{2}\right) + \hat{k} \sin\left(\frac{\phi_l N_s t}{2}\right),$$

where N_s is the total number of footsteps.

The reference trajectory for the body with respect to $\mathcal{F}_0$ during one gait cycle is given by

$$\mathbf{x}_{b_d}^0(t) = \mathbf{x}_{b_i}^0 \mathbf{r}_b(t),$$

where $\mathbf{x}_{b_i}^0$ is the initial pose of the robot's torso with respect to the inertial frame $\mathcal{F}_0$.

In the control strategy, the robot's body orientation is represented by the primary part of the absolute variable with respect to the frame $\mathcal{F}_b$. Assuming that the pose of the current supporting foot with respect to $\mathcal{F}_0$, $\mathbf{x}_{e_j}^0$, is known, the reference trajectory for the absolute variable is given by

$$\mathbf{x}_{a_d}^b(t) = \left(\mathbf{x}_{b_d}^0(t)\right)^* \mathbf{x}_{e_j}^0 \mathbf{x}_a^{e_j}, \quad (5.18)$$

where $\mathbf{x}_a^{e_j}$ is given by (4.12).

Taking the first derivative of (5.18) with respect to time

$$\dot{\mathbf{x}}_{a_d}^b(t) = \left(\dot{\mathbf{x}}_{b_d}^0(t)\right)^* \mathbf{x}_{e_j}^0 \mathbf{x}_a^{e_j} + \left(\mathbf{x}_{b_d}^0(t)\right)^* \mathbf{x}_{e_j}^0 \dot{\mathbf{x}}_a^{e_j},$$

where

$$\dot{\mathbf{x}}_{b_d}^0(t) = \mathbf{x}_{b_i}^0 \dot{\mathbf{r}}_b(t),$$

and $\dot{\mathbf{r}}_b(t)$ is given by

$$\dot{\mathbf{r}}_b(t) = -\frac{\phi_l N_s}{2} \sin\left(\frac{\phi_l N_s t}{2}\right) + \hat{k} \frac{\phi_l N_s}{2} \cos\left(\frac{\phi_l N_s t}{2}\right).$$

Example 5.4. Given the radius of the trajectory $r_w = 1\text{m}$, the step length $s_l = 0.05\text{m}$, and the number of steps $N_s = 64$. Assuming that the initial absolute variable of the robot is at the origin of the inertial frame, and the orientations of the feet coincide with the orientation of the inertial frame, the path generated using (5.15) and (5.16) is illustrated in Figure (5.10).

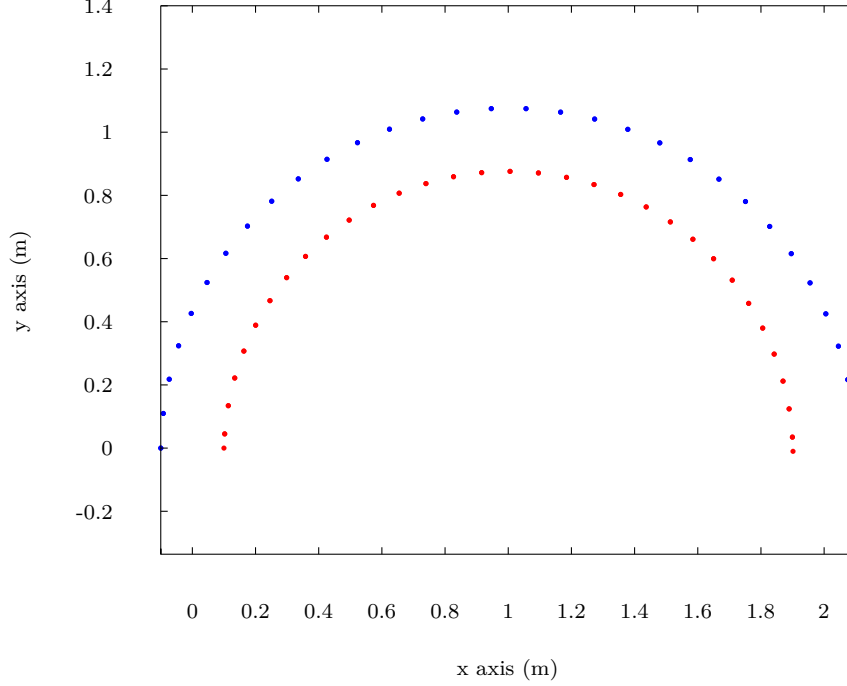


Figure 5.10: Circular Path of Example (5.4). The *red dots* represent the path generated for the right foot, and the *blue dots* represent the path generated for the left foot.

5.3 Control Strategies

We must control the feet poses and the robot's global CoM to follow the references generated by the methods described in Section 5.2. As long as the reference frame is attached to the current supporting-foot, tracking the feet poses is equivalent to track only the swinging-foot pose. Furthermore, to avoid fluctuations of the angular momentum about the CoM, what can lead the GCoM out of the support polygon, the robot's body posture must be kept aligned to the gravitational vector during the motion.

A control strategy for gait and balance must satisfy, simultaneously, three objectives:

1. Tracking the components in x and y axes of the robot's global CoM ;
2. Tracking of the swinging-foot pose;
3. Regulation of the robot's body orientation.

These objectives require two, six and three DOF, respectively, thus the complete task requires eleven DOF. In the case of a walking motion with a varying direction, the third control objective becomes a tracking task, rather than regulation.

The first control strategy presented here is based on the work by Park & Lee (2013). According to this work, the first control objective is accomplished using a lower-body Jacobian matrix associated with the GCoM, defined as $\bar{J}_{com_{LB}}$. Since $J_{com_{LB}}$, shown in (4.7), is associated with the components in x , y and z axes, then $\bar{J}_{com_{LB}}$ is obtained removing the third row of $J_{com_{LB}}$. In a similar way, we can obtain $\bar{J}_{com_{UB}}$ and $\bar{J}_{com}$.

Using the reference poses defined a priori for the feet, a reference trajectory parameterized in time is obtained for the relative variable, described in Section 4.4. The relative Jacobian matrix J_r , given by (4.11), is used to fulfill the second control objective. Finally, the third control objective is satisfied with the absolute Jacobian matrix expressed in a frame attached to the robot's body J_a^b , shown in (4.13) and (4.14).

This control strategy is given by

$$\dot{q}_1 = J_1^+ \lambda e(t), \quad (5.19)$$

where J_1^+ is the pseudoinverse—computed by the Singular Value Decomposition (SVD)—of J_1 , which is defined as

$$J_1 = \begin{bmatrix} J_r \\ J_{a-ori}^b \\ \bar{J}_{com_{LB}} \end{bmatrix}. \quad (5.20)$$

In this equation, J_{a-ori}^b is the orientation Jacobian matrix related to J_a^b , λ is a scalar gain, and e is the error between the reference and actual values, i.e.

$$e(t) = X_d(t) - X = \begin{bmatrix} \text{vec}_8 \underline{x}_{r_d}(t) \\ \text{vec}_4 \mathcal{P}(\underline{x}_{a_d}^b(t)) \\ c_{x_d}(t) \\ c_{y_d}(t) \end{bmatrix} - \begin{bmatrix} \text{vec}_8 \underline{x}_r \\ \text{vec}_4 \mathcal{P}(\underline{x}_a^b) \\ c_x \\ c_y \end{bmatrix}, \quad (5.21)$$

where the variables $\underline{x}_{r_d}$, $\underline{x}_{a_d}^b$, c_{x_d} and c_{y_d} represent the desired values of the respective variables.

The control strategy presented in (5.19) is designed for regulation tasks. For tracking tasks, the expected behavior of this type of controller is a delayed response regarding the reference.

In the next control strategy, a feed-forward term is added in order to eliminate the delay in tracking tasks. The controller is given by

$$\dot{q}_2 = J_1^+ \left(\dot{X}_d(t) + \lambda e(t) \right), \quad (5.22)$$

where

$$\dot{X}_d(t) = \begin{bmatrix} \text{vec}_8(\dot{\mathbf{x}}_{rd}(t)) \\ \text{vec}_4(\mathcal{P}(\dot{\mathbf{x}}_{ad}^b(t))) \\ \dot{x}_d(t) \\ \dot{y}_d(t) \end{bmatrix}. \quad (5.23)$$

The third control strategy considers the arms joints in the control law, by substituting the Jacobian matrix $\bar{J}_{com_{LB}}$ by $\bar{J}_{com}$ in (5.20), that is, the whole body is considered. This control strategy can be written as

$$\dot{q}_3 = J_2^+ (\dot{X}_d(t) + \lambda e(t)), \quad (5.24)$$

where

$$J_2 = \begin{bmatrix} J_r \\ J_{a-ori}^b \\ \bar{J}_{com} \end{bmatrix}.$$

Considering a humanoid robot whose legs have six actuated DOF each, the system is redundant for the balance and gait control, since this task demands eleven DOF. However, for a humanoid robot with 5 DOF legs, this system is under-actuated. Under this condition, the third control strategy can be useful, because the addition of the arms joints in the control law increases the actuated DOF of the system.

Including the arms joints in the control law can make the system redundant for the locomotion task. Thus it is possible to include additional tasks operating in the null space of the Jacobian matrix of the locomotion task, given by J_2 . This additional task is described by a function $f(q)$, which is differentiable:

$$\begin{aligned} \frac{df}{dt} &= \frac{\partial f}{\partial q} \dot{q} \\ &= J_f \dot{q}, \end{aligned} \quad (5.25)$$

where J_f is the Jacobian matrix representing this additional task.

Including the secondary task in the control strategy given by (5.24), we have the fourth strategy, defined as

$$\dot{q}_4 = J_2^+ (\dot{X}_d(t) + \lambda e(t)) + (I - J_2^+ J_2) J_f^+ \lambda_2 e_f(t), \quad (5.26)$$

where $e_f(t) = f_d(q) - f(q)$ represents the difference between the desired and the current value of $f(q)$, and λ_2 is a scalar positive gain. We desire that $f_d(q) = 0$, meaning that the secondary task had been reached, then (5.26) can be rewritten as follows

$$\dot{q}_4 = J_2^+ (\dot{X}_d(t) + \lambda e(t)) - (I - J_2^+ J_2) J_f^+ \lambda_2 f(q). \quad (5.27)$$

The stability proofs of these strategies are shown next.

5.4 Controller Stability Proof

We want to prove the stability of the control law in (5.24).

Consider the SVD representation of the Jacobian matrix J , whose dimension is $m \times n$, given by $J = U\Sigma V^T$, in such a way that

$$\begin{aligned} V &= \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}, \\ U &= \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}, \\ \Sigma &= \left[\begin{array}{c|c} \Sigma_r & 0 \\ \hline 0 & 0 \end{array} \right] = \left[\begin{array}{ccc|c} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_r & 0 \\ \hline & & 0 & 0 \end{array} \right], \end{aligned}$$

where $V \in O(n)$ ³, whose columns are the eigenvectors of $J^T J$, and $U \in O(m)$, whose columns are the eigenvectors of $J J^T$. Furthermore, $\text{rank}(J) = r$, and the elements of the diagonal of Σ are the nonzero singular values of J , given by $\sigma_i = \sqrt{\lambda_i}$, being λ_i an eigenvalue of $J J^T$, and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$.

The pseudoinverse of matrix J is computed as

$$J^+ = V\Sigma^+U^T,$$

where

$$\Sigma^+ = \left[\begin{array}{c|c} \Sigma_r^{-1} & 0 \\ \hline 0 & 0 \end{array} \right].$$

Since Σ_r is an $r \times r$ matrix, and $\text{rank}(\Sigma_r) = r$, thus Σ_r is invertible and is calculated as

$$\Sigma_r^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_r} \end{bmatrix}.$$

Given these facts, we want to determine the result of $J J^+$. Thus,

$$\begin{aligned} J J^+ &= (U\Sigma V^T)(V\Sigma^+U^T), \\ &= U\Sigma\Sigma^+U^T, \end{aligned} \tag{5.28}$$

³ $O(n)$ represents an orthogonal group of order n , such that $M \in O(n) \Leftrightarrow M \in \mathbb{R}^{n \times n}$, $\det M = \pm 1$, and $M^{-1} = M^T$.

because V is orthogonal, thus $V^T V = I$. Furthermore

$$\begin{aligned}\Sigma \Sigma^+ &= \left[\begin{array}{c|c} \Sigma_r & 0 \\ \hline 0 & 0 \end{array} \right] \left[\begin{array}{c|c} \Sigma_r^{-1} & 0 \\ \hline 0 & 0 \end{array} \right], \\ &= \left[\begin{array}{c|c} \Sigma_r \Sigma_r^{-1} & 0 \\ \hline 0 & 0 \end{array} \right], \\ &= \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array} \right].\end{aligned}$$

Considering $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$, where $\dim(U_1) = m \times r$ and $\dim(U_2) = m \times (m - r)$, substituting U in (5.28), we obtain

$$\begin{aligned}J J^+ &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array} \right] \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}, \\ &= U_1 U_1^T.\end{aligned}$$

Fact 5.1. (See Fact 3.7.26 of Bernstein (2005)) Considering a matrix A , such that $A \in \mathbb{R}^{n \times m}$. The matrix AA^T is always a positive semi-definite matrix. Besides that, AA^T is a positive definite matrix iff A is right invertible, i.e.

$$\exists A^R \text{ such that } A^R = A^T (AA^T)^{-1}.$$

This condition is true when AA^T is invertible, which is true iff A is full row rank.

Corollary 5.1. The matrix U_1 , whose dimension is $m \times r$, is not full row rank since $m > r$. Thus, $U_1 U_1^T$ is not positive definite, but, from Fact (5.1), it is still a positive semi-definite matrix.

Taking the first derivative of (5.21), we have that

$$\dot{e}(t) = \dot{X}_d(t) - \dot{X}. \quad (5.29)$$

Given the DFKM defined in (3.26) represented by

$$\dot{X} = J\dot{q}, \quad (5.30)$$

substituting (5.30) in (5.29), we have that

$$\dot{e}(t) = \dot{X}_d(t) - J\dot{q}. \quad (5.31)$$

As aforementioned, we wish to prove stability for a tracking task, when $\dot{X}_d$ is time-varying, which is described in (5.23). We propose the following candidate Lyapunov

function

$$V = \frac{1}{2}e^T e.$$

Taking the first derivative of this Lyapunov function and substituting $\dot{e}$ given by (5.31)

$$\begin{aligned}\dot{V} &= \frac{1}{2}(\dot{e}^T e + e^T \dot{e}), \quad \dot{e}e = e^T \dot{e} \\ &= e^T \dot{e} = e^T (\dot{X}_d - \dot{X}) = e^T (\dot{X}_d - Ju),\end{aligned}$$

where we assume the single integrator model, that is, $u \triangleq \dot{q}$.

For the choice $u = J^+ (\dot{X}_d + \lambda e)$, we obtain

$$\dot{V} = e^T [\dot{X}_d - JJ^+ (\dot{X}_d + \lambda e)].$$

We assume that⁴ $\dot{X}_d \in R(J)$, which is a reasonable assumption, because $\dot{X}_d$ is generated arbitrarily and the desired trajectory must be kinematically feasible. Furthermore, according to Section 6.1 of Bernstein (2005), if $J \in \mathbb{R}^{n \times m}$, then $J^+ \in \mathbb{R}^{m \times n}$, and J^+ satisfies the following condition:

$$JJ^+J = J, \tag{5.32}$$

Then, using (5.30) and (5.32):

$$\dot{X}_d = J\dot{q}_d = JJ^+J\dot{q}_d = JJ^+\dot{X}_d,$$

for $\dot{q}_d \in \mathbb{R}^n$. Thus,

$$\begin{aligned}\dot{V} &= e^T [\dot{X}_d - JJ^+\dot{X}_d - \lambda JJ^+e] \\ &= -\lambda e^T U_1 U_1^T e.\end{aligned}$$

Using Corollary (5.1), $U_1 U_1^T \geq 0$. Thus, from Lyapunov's stability theory (See Theorem 4.1 of Khalil (2002)), the system is stable, for $\lambda > 0$. It guarantees that the robot will be stable during the gait, but residual errors can exist during the motion. To prove the asymptotic stability of this system, one can apply the Barbalat Lemma (See Subsection 4.5.2 of Slotine & Li (1991)), which is not done in this work.

5.5 Chapter Overview

This chapter presented the methods to obtain the reference trajectories for a balanced walking motion, and the gait control strategies.

⁴The range space of J is defined as $R(J) \triangleq \{J\dot{q} \mid \dot{q} \in \mathbb{R}^n\}$.

Section 5.1 discussed the stability conditions of a humanoid robot during the locomotion, which requires that the GCoM always be within the support polygon.

Section 5.2 presented the methods to generate the reference trajectories for the robot's CoM and the feet, given the desired footprints, which are based on the inverted pendulum model and Bézier curves, respectively. The method to obtain the desired footprints for circular trajectories was also shown.

Section 5.3 detailed four control strategies, which fulfill three control objectives: tracking of the relative variable, regulation (or tracking) of the absolute variable, and tracking of the GCoM. The first control strategy was based on an existent work, the second control strategy included a feed-forward term, the third control strategy included the arms in the tracking of the GCoM, and the fourth control strategy included a secondary task in the null space of the Jacobian matrix of locomotion, which was possible thanks to the redundancy of the system with respect to the locomotion task.

Section 5.4 showed the stability proof for the control strategies presented. The asymptotic stability was not proven.

6

Experiments and Results

The modeling and control strategies presented in this dissertation were implemented and tested in a simulation environment, and this chapter shows and discusses the obtained results. The chapter is organized as follows: in Section 6.1 the platforms used in the simulations are specified; Section 6.2 details the model validation and its results; and Section 6.3 and Section 6.4 show the results obtained from the experiments with the robot executing the control strategies presented.

6.1 Platforms Specification

6.1.1 Simulation Environment Specification

We intend to validate the methods proposed in this work in a realistic environment, in order to have a better understanding about the expected results when a real robot is used in the future. Thus, the experiments were executed in the software V-REP (Rohmer et al., 2013), which consists of a virtual reality simulation software for robotic systems. This software has some calculation modules, including the dynamics module and the collision detection module, better explained as follows:

- **Dynamics module:** computes rigid body dynamics and interactions using the Bullet Physics Library¹ or the Open Dynamics Engine (ODE)². In this work, we

¹Source: <http://www.bulletphysics.org>.

²Source: <http://www.ode.org>.

arbitrarily choose the ODE, since we did not investigate the accuracy of the results presented by both methods.

- **Collision detection module:** computes interference between any shape or collection of shapes, and is fully independent from the collision response calculation algorithm of the dynamics module.

In addition, the joints in V-REP have dynamic features, including saturation in the output torques.

In this framework, all objects and models can be individually controlled using, for instance, embedded routines, plugins, ROS (Robot Operating System) nodes, or external applications connected through Application Programming Interfaces (APIs). The APIs are provided by the developers and offer a list of useful functions available for C/C++, Python, Java, Lua, MATLAB®, Octave, or Urbi.

In our case, the controllers were implemented in an external MATLAB routine connected with V-REP through the API for MATLAB, with a sampling time of 20ms. The communication between the controller and the environment is made through sockets using a client-server structure, being the external application the client and the simulation environment the server. The structure of the simulation framework used in the trials is represented in Figure 6.1.

In V-REP, the smaller the simulation time step, the slower is the evolution of the simulated environment, since the graphics must be updated in each simulation step, requiring a great amount of graphics processor effort. Since there is no guarantees regarding the communication sampling time, we choose a simulation time step in V-REP of 2ms, in order to have a slow evolution of the environment. Note that the simulation time does not correspond to the real time in this case.

Moreover, the open-source library DQ Robotics³ was used to represent and operate on dual quaternions.

The simulations were executed under the following specifications:

- Processor Intel Core I3 CPU M380 @ 2.53GHz and 4,00GB Memory RAM;
- Operating System Microsoft Windows® version 7 64-bit;
- MATLAB version R2012a 64-bit;
- V-REP version PRO EDU 3.1.3 32-bit;
- Graphics processor Intel (R) HD Graphics 512 MB.

³Source: <http://dqrobotics.sourceforge.net/>

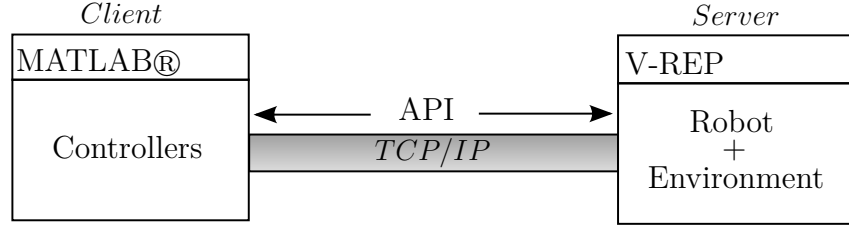


Figure 6.1: Structure of the simulation framework.

6.1.2 Robot Specification

The humanoid robot used in the experiments is a native model offered by V-REP called ASTI, which is depicted in Figure 6.2. This robot has 20 revolute joints: 6 in each leg, 3 in each arm, and 2 in the head.

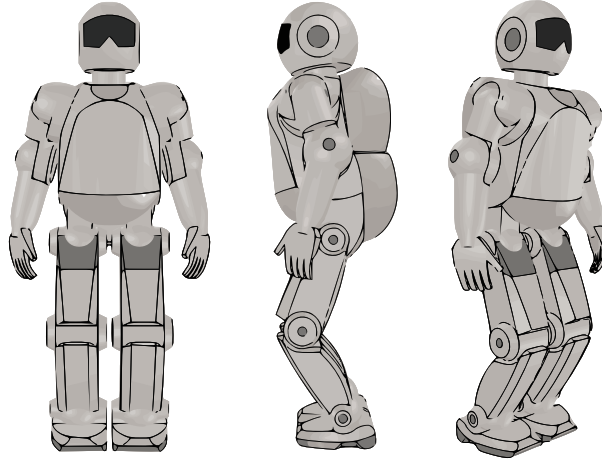


Figure 6.2: Humanoid robot ASTI.

The joints convention for each limb is defined in Figure 6.3. According to this convention, the robot's legs are represented by the same FKM, and the robot's arms FKMs are quite similar. The Denavit-Hartenberg (DH) parameters were extracted for each limb separately, and are presented in Tables A.1, A.2, and A.3 in Appendix A.

In addition to the DH parameters, rigid transformations are required to achieve the defined end-effectors poses. The transformation for the robot's hands consists of a translation of 27.9 cm in the x -axis followed by a rotation of $\pi/2$ in the y -axis; and the transformation for the robot's feet consists of a rotation of $-\pi/2$ in the y -axis followed by a rotation of $-\pi/2$ in the z -axis, and then, a translation of -8.80 cm in the z -axis. These transformations, given by the dual quaternions $\underline{\mathbf{x}}_H^E$ and $\underline{\mathbf{x}}_F^E$, respectively, are defined as

$$\begin{aligned}\underline{\mathbf{x}}_H^E &= \left(\cos\left(\frac{\pi}{4}\right) + \hat{j} \sin\left(\frac{\pi}{4}\right) \right) + \frac{1}{2}\varepsilon \left[0.279\hat{i} \left(\cos\left(\frac{\pi}{4}\right) + \hat{j} \sin\left(\frac{\pi}{4}\right) \right) \right] \\ \underline{\mathbf{x}}_F^E &= \mathbf{r}_1 \mathbf{r}_2 \underline{\mathbf{p}},\end{aligned}$$

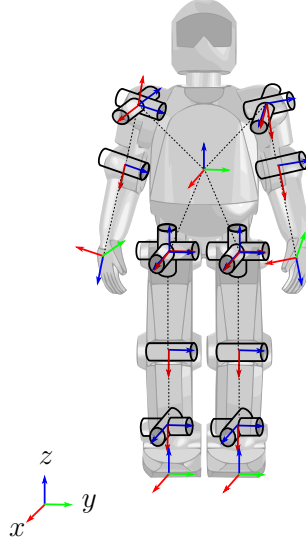


Figure 6.3: DH convention for ASTI.

where

$$\begin{aligned}\mathbf{r}_1 &= \cos\left(-\frac{\pi}{4}\right) + \hat{j} \sin\left(-\frac{\pi}{4}\right) \\ \mathbf{r}_2 &= \cos\left(-\frac{\pi}{4}\right) + \hat{k} \sin\left(-\frac{\pi}{4}\right) \\ \underline{\mathbf{p}} &= 1 + \frac{1}{2}\varepsilon\left(-0.088\hat{k}\right)\end{aligned}$$

The masses of the robot's limbs are defined in the model provided by V-REP, and are listed in Table 6.1. The head mass is 2 kg, and the torso mass is 11 kg.

Part	Mass (kg)
Upper-arm	7
Lower-arm	2
Upper-leg	8
Lower-leg	6
Foot	5

Table 6.1: Robot's links masses.

6.2 Model Validation

In order to verify the reliability of the robot's kinematic model, a validation methodology was executed, which consists of capturing kinematic data from the robot in an arbitrary movement, and comparing the actual to the estimated data.

With the aim of keeping the reference frame stationary during the procedure, which is attached to the robot's torso, we fixed the robot in a suspended pole, as depicted

in Figure 6.4. Since the head joints are not actuated in this work, the head was kept stationary with respect to the torso.

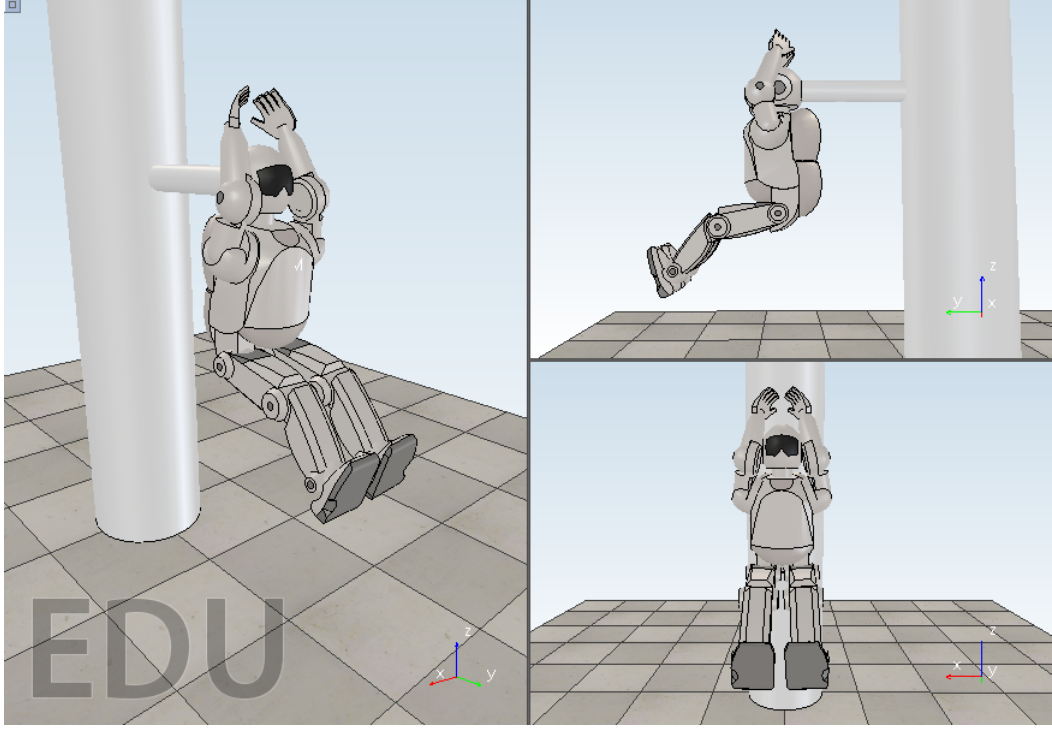


Figure 6.4: Robot configuration during the validation.

The monitored information was:

- Poses of each limb end-effector;
- Linear and angular velocities of each limb end-effector;
- Robot's Center of Mass (CoM);
- Linear velocity of the CoM.

This information was directly captured in V-REP with respect to an inertial frame, and is regarded as the actual data. The joints angles and velocities were also captured, and used in the Forward Kinematics Models (FKMs) and Differential Forward Kinematics Models (DFKMs) to obtain the estimated data.

In Figure 6.5, the estimated and actual CoM and CoM linear velocity are shown. Furthermore, the actual and estimated poses trajectories, linear velocities evolution and angular velocities evolution of the limbs end-effectors are depicted in Figures 6.6 and 6.7, respectively. The poses are represented by the coefficients of the dual quaternions, the linear velocities are represented separately in x , y , and z axes as well as the angular velocities.

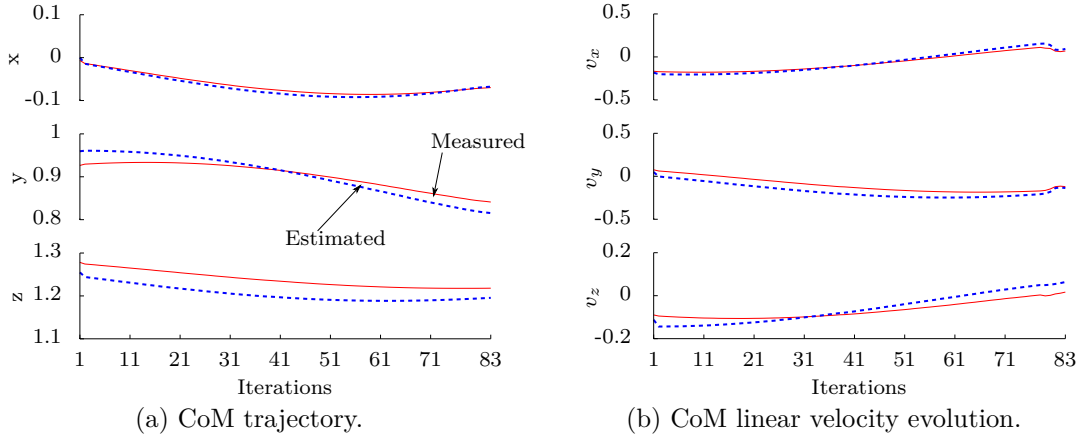


Figure 6.5: CoM and CoM linear velocity validation. The *solid red line* represents the measured data and the *dashed blue line* represents the estimated data.

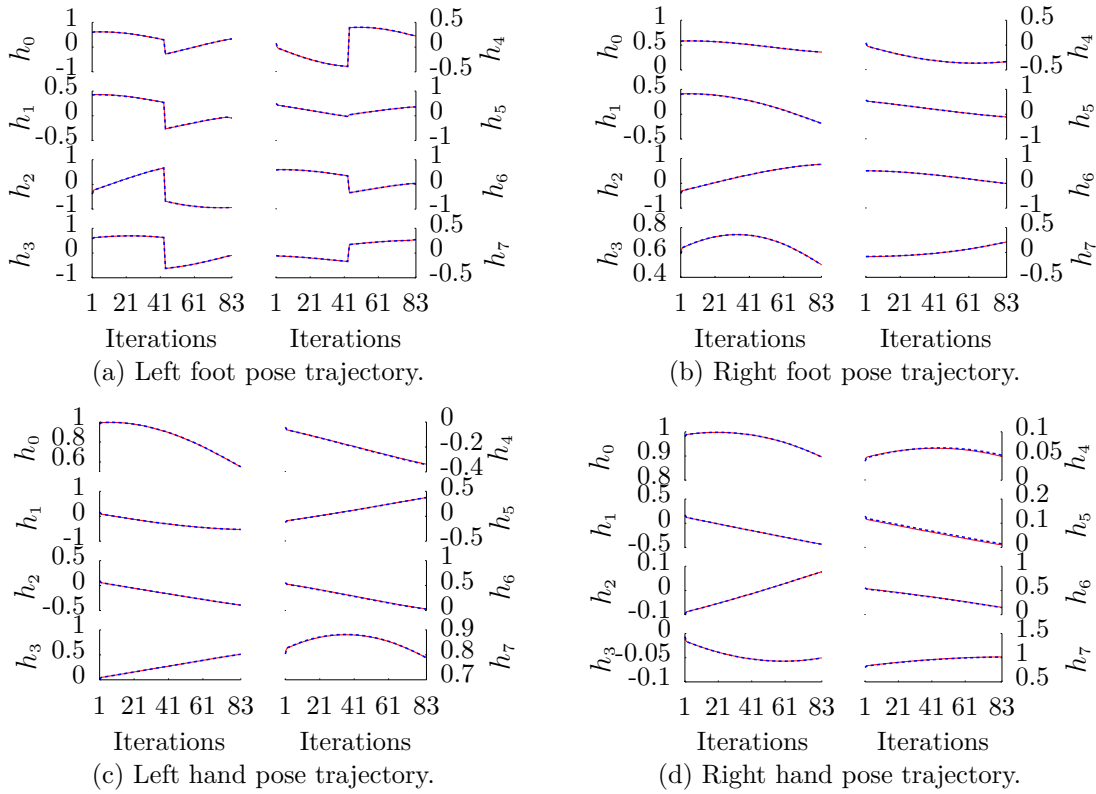


Figure 6.6: Poses of the limbs end-effectors during the model validation. The *solid red line* represents the measured data and the *dashed blue line* represents the estimated data.

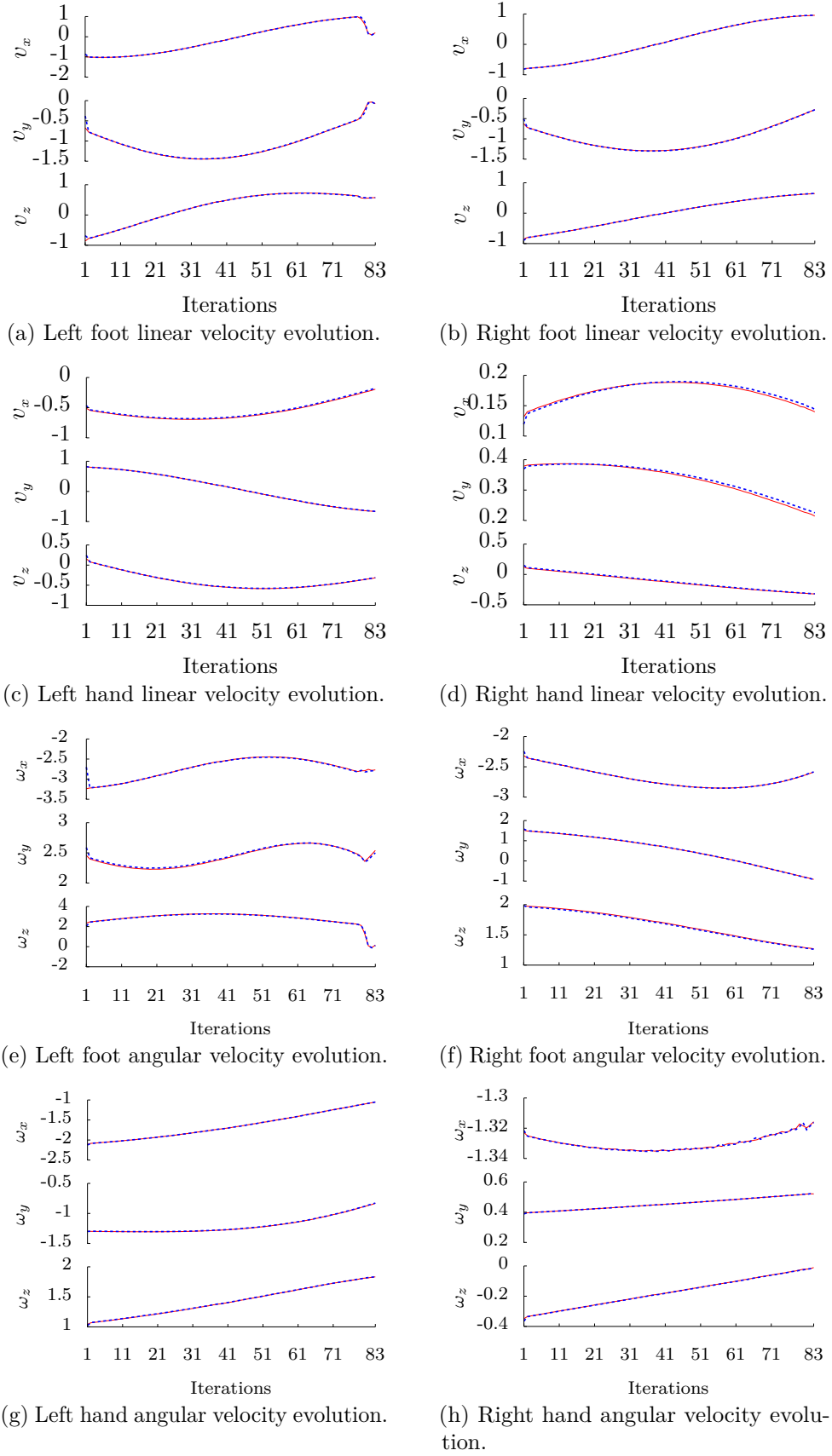


Figure 6.7: Linear and angular velocities of the limbs end-effectors during the model validation. The *solid red line* represents the measured data and the *dashed blue line* represents the estimated data.

By observing the poses trajectories and velocities evolution of the limbs end-effectors, we can notice that the actual and estimated data are almost coincident, indicating a good reliability of the limbs FKMs and DFKMs. Regarding the robot's CoM and CoM linear velocity, the actual and estimated data are close, but slightly different. Recalling that we assumed that the CoM of a link is located in its geometric center, the observed deviation is probably a consequence of this approximation. However, we can notice that, even though the links are not absolutely symmetric, the FKM for the CoM still generates an estimation that is very close to the actual values.

6.3 Center of Mass Control

The aim of the first experiment performed is to verify the accuracy of the CoM tracking. In this experiment, the robot's feet are kept stationary and a set of different trajectories are defined for the CoM, and the control strategy adopted is given by (5.19), with the difference that the CoM Jacobian matrix includes the z -axis, i.e. J_{comLB} , in order to track the CoM, not only the GCoM. These trajectories are defined with respect to the inertial frame used by V-REP as default, and are designed in such a way that its projection onto the ground be within the robot's support polygon—which is known a priori, because the feet poses do not change.

Three different trajectories were tested to observe the robot's stability close to the support-polygon edges, as well as to check the occurrence of self-collisions during the motion: a circle in xy -plane, a sine in xy -plane, and a sine xz -plane. The results obtained from the execution of the circular trajectory are presented in Figure 6.8, where the robot's CoM trajectory with respect to the inertial frame, the complete pose of the relative variable, and the orientation—i.e. the primary part of the dual quaternion—of the absolute variable are depicted. The results obtained from the execution of the remaining trajectories are shown in Figures B.1 and B.2 of Appendix B.

Standing up is an important movement to be executed by humanoid robots, in order to reach a stable configuration after squatting or falling down on the floor. In order to verify if the robot is able to execute this movement, it was set in a squatting posture, and a vertical line was defined as the CoM reference trajectory. The control strategy adopted in this case was given by (5.24), where all joints are actuated, with the difference that the CoM Jacobian matrix includes the z -axis, i.e. J_{com} . The robot successfully stands up, as shown in the results depicted in Figure 6.9, where all controlled variables followed the given references. Figure 6.10 shows some snapshots of the simulation to illustrate the executed movement.

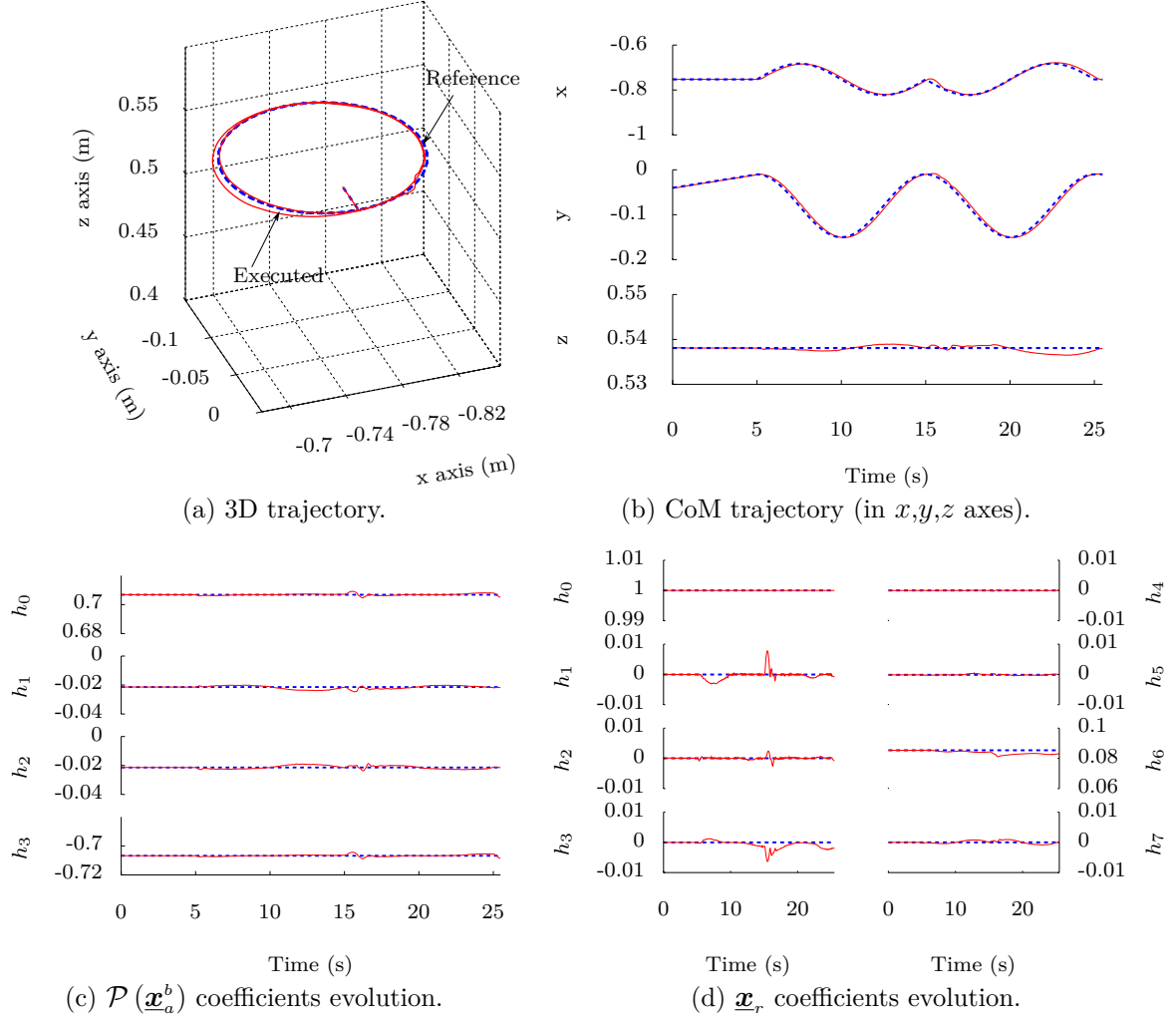


Figure 6.8: Results obtained from the execution of a circular trajectory. The *solid red line* represents the measured values and the *dashed blue line* represents the reference.

6.4 Gait and Balance Control

Some walking patterns were designed in order to verify the robot's behavior during the locomotion. Recalling the parameters defined in the Subsection 5.2.2, d_h is the step height, N_s is the number of footsteps, and t_s is the time duration of the step.

Four walking patterns are defined, as follows:

1. Step length = 5cm, $t_s = 0.5$ s;
2. Step length = 10cm, $t_s = 1$ s;
3. Step length = 20cm, $t_s = 1$ s;
4. Step length = 30cm, $t_s = 1.5$ s.

Furthermore, in all cases, $d_h = 5$ cm and $N_s = 20$, and all patterns describe a locomotion in a straight line in the forward direction. In the experiments performed hereafter, we choose

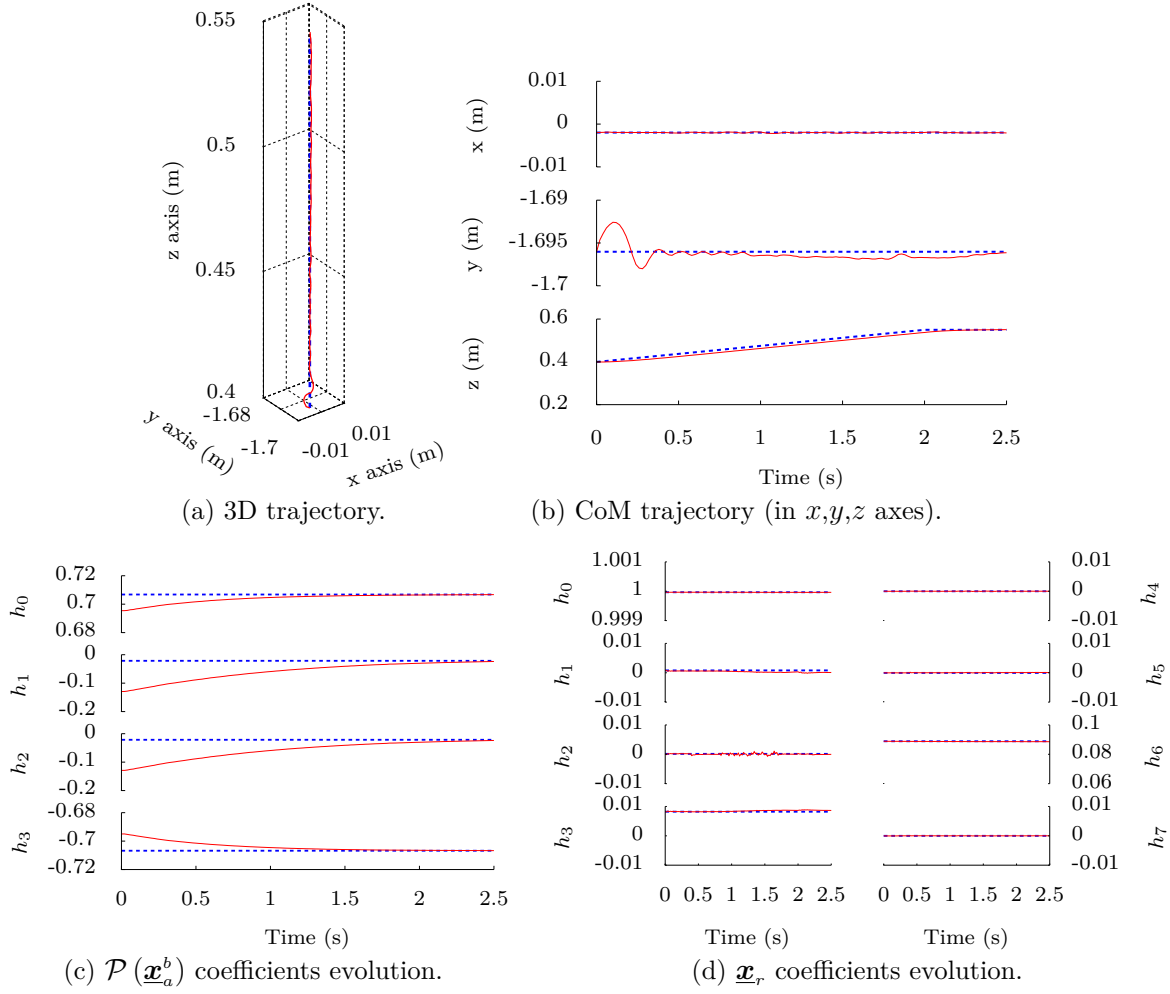


Figure 6.9: Results obtained from the execution of the standing up movement. The *solid red line* represents the measured values and the *dashed blue line* represents the reference.

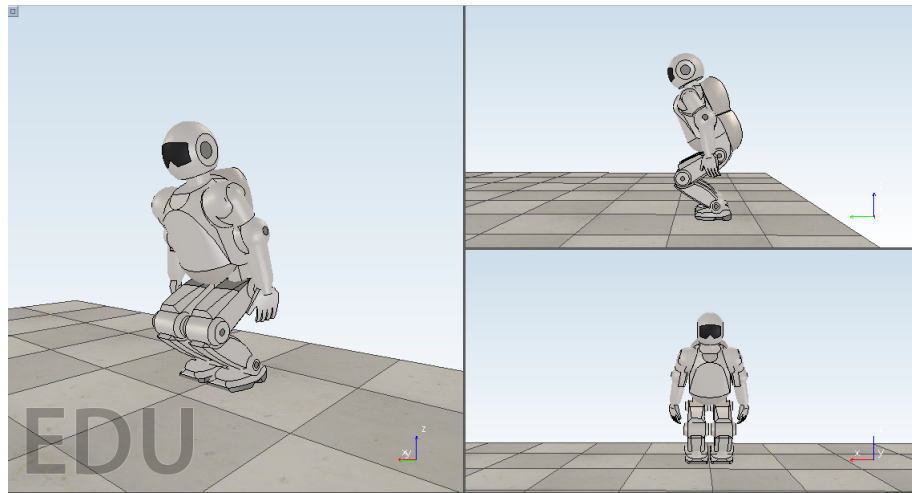
the right foot as the starting foot. Moreover, the controller gain λ was chosen by try and error, and was set to 0.4.

In order to provide a better understanding to the reader, the control strategies presented in Chapter 5 are summarized in Table 6.2.

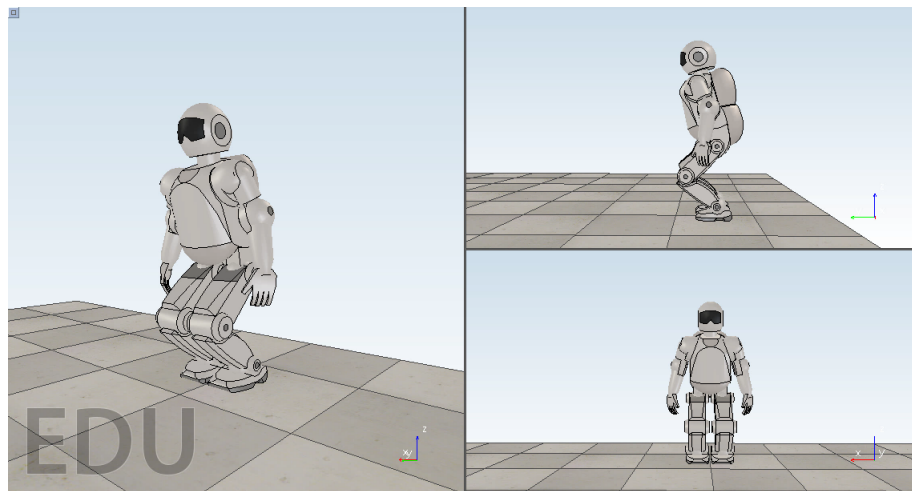
In the first experiment, the adopted control strategy is given by (5.19), which is labeled herein as “strategy 1.” The results obtained from the execution of the second walking pattern are presented in Figure 6.11, where the robot’s CoM trajectory with respect to the inertial frame, the complete pose of the relative variable, and the orientation of the absolute variable are depicted. Since the z -axis of the CoM is not controlled, only its measured values are shown for the sake of information.

The relative variable represents the feet trajectories in a very compact way, however, the interpretation of the executed movement is not obvious through its analysis. Thus, the feet trajectories with respect to the inertial frame are also depicted in Figure 6.12.

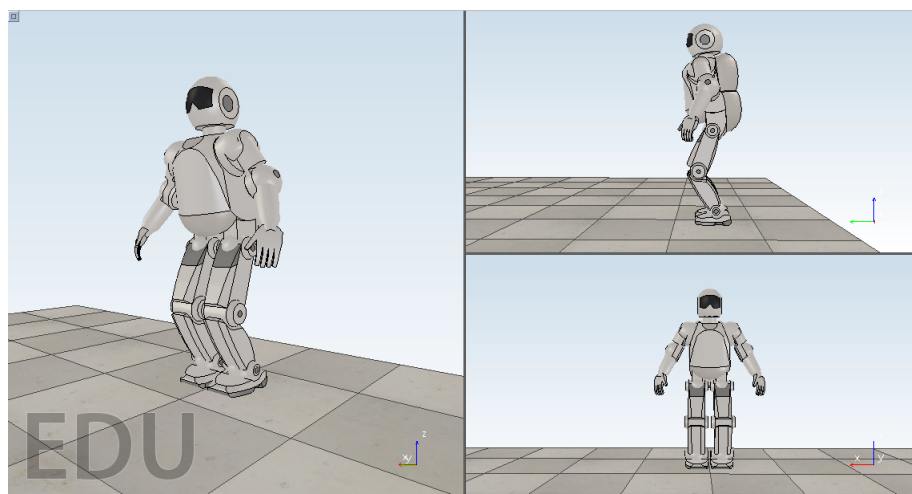
These results show that the robot is able to execute a balanced locomotion when controlled by the strategy 1, since all controlled variables follow the references. In



(a) Simulation snapshot at 0 s.



(b) Simulation snapshot after 1 s.



(c) Simulation snapshot after 2.5 s.

Figure 6.10: Simulation snapshots of the standing up movement.

Control strategy	Description	Control law
1	Tracking of the GCoM and the relative variable, and regulation of the absolute variable orientation	$\dot{q}_1 = J_1^+ \lambda e(t)$
2	Addition of a feed-forward term	$\dot{q}_2 = J_1^+ \left(\dot{X}_d(t) + \lambda e(t) \right)$
3	Inclusion of the arms in the control law	$\dot{q}_3 = J_2^+ \left(\dot{X}_d(t) + \lambda e(t) \right)$
4	Inclusion of a secondary control law to keep the arms close to the body	$\dot{q}_4 = J_2^+ \left(\dot{X}_d(t) + \lambda e(t) \right) - (I - J_2^+ J_2) J_f^+ \lambda_2 f(q)$

Table 6.2: Summary of the proposed control strategies.

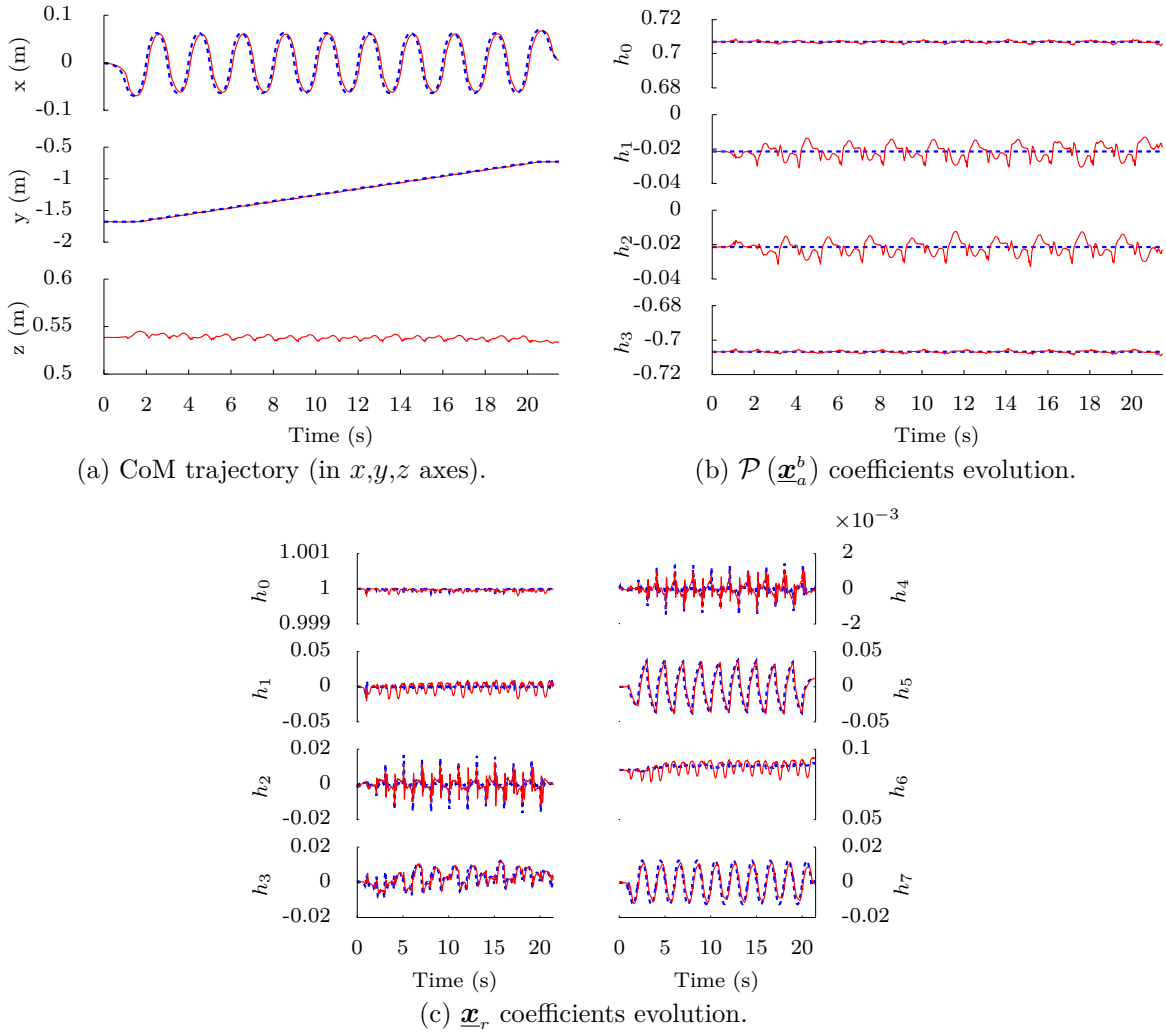


Figure 6.11: Results obtained from the execution of the second pattern controlled using the strategy 1. The *dashed blue line* represents the reference, and the *solid red line* represents the measured values.

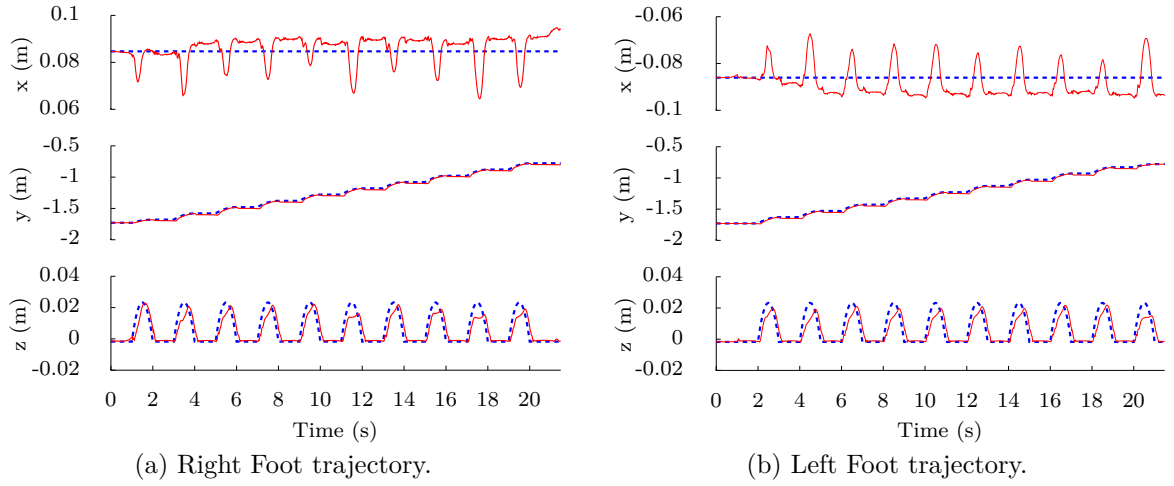
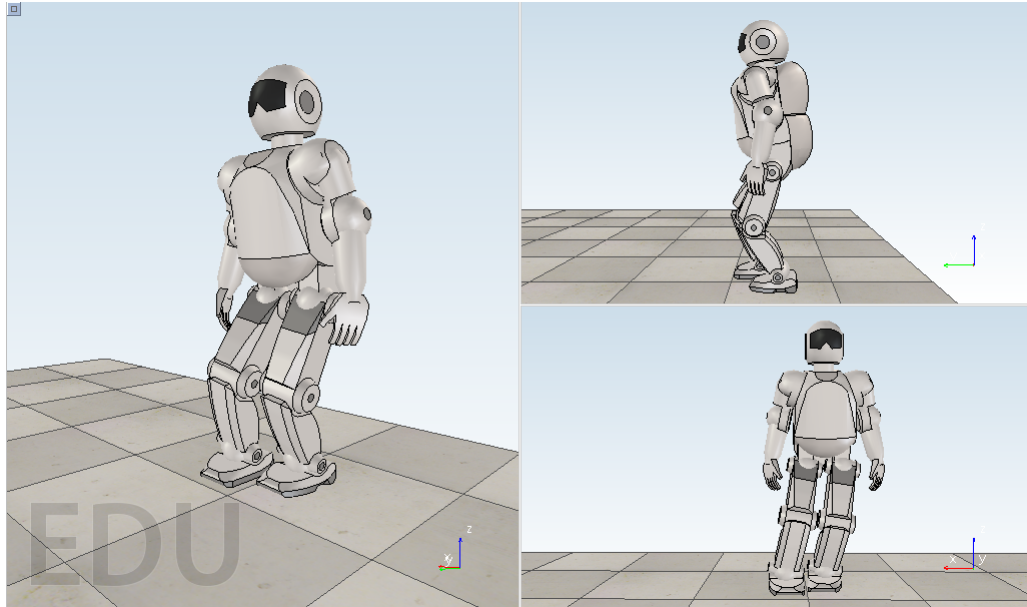


Figure 6.12: Feet trajectories (in x, y, z axes) obtained from the execution of the second pattern controlled using the strategy 1. The *dashed blue line* represents the reference, and the *solid red line* represents the measured values.

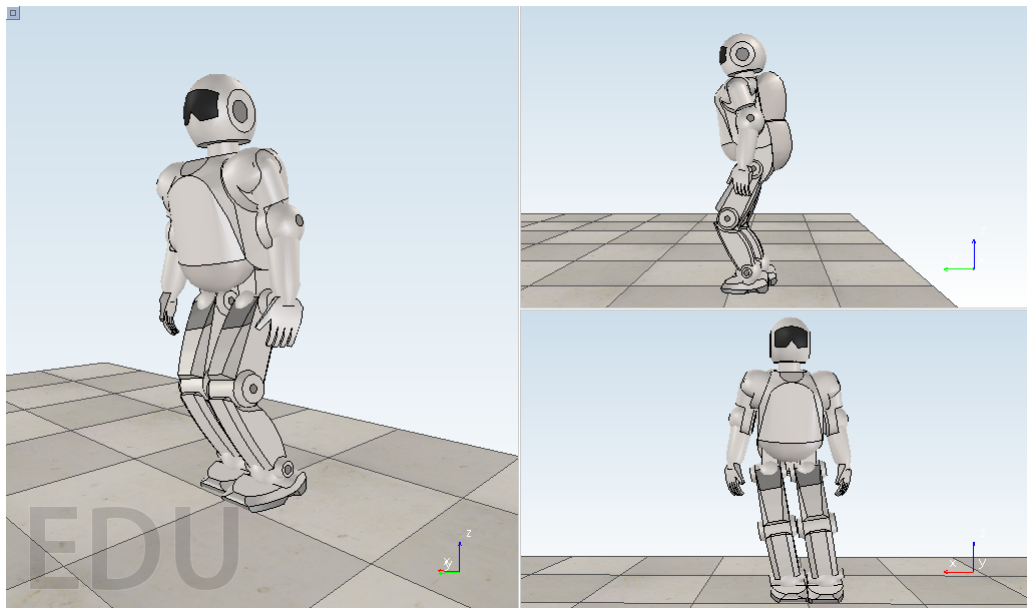
Figure 6.13, snapshots of the walking motion during two SSP are depicted, where we can notice that the robot's body is correctly oriented, and the feet are parallel to the ground during the swing movement, as expected.

In Figure 6.12, we can notice periodic deviations in the x -axis of the feet trajectories, which might be caused by the interaction between the robot and the environment. The dynamics of this system changes constantly as the supporting-foot switches, and the kinematic model does not capture this variation, thus, it can represent a disturbance to the system. The controller actuates in order to correct the trajectory, bringing the swinging foot closer to the reference after the disturbance. However, it is not enough to nullify the error in the swinging-foot trajectory within the step time duration, which accumulates over the trajectory. Moreover, as expected, we can observe a delay response of the system with respect to the reference trajectory, more evident in Figures 6.11a and 6.12. Aiming to reduce this delay, a feed-forward term is included in the control strategy, as defined in (5.22), labeled herein as “strategy 2.”

In Figure 6.14, the results obtained from the execution of the first walking pattern using the strategy 2 are compared with the results obtained using the strategy 1. Since the reference of the relative variable is calculated on-the-fly with respect to the current supporting-foot pose, it does not make sense to compare the evolution of $\underline{x}_r$, since the reference will not be the same for both executions. Thus, instead of representing the relative variable evolution, the feet trajectories are shown. We can notice that the delay is significantly reduced when the feed-forward term is included, which becomes more evident when the chart is expanded. The results obtained from the execution of the remaining walking patterns using strategies 1 and 2 are shown in Figures B.3, B.4, and B.5 of Appendix B.



(a) Simulation snapshot after 1.18 s (step with right foot).



(b) Simulation snapshot after 2.13 s (step with left foot).

Figure 6.13: Simulation snapshots from the execution of the second pattern controlled using the strategy 1.

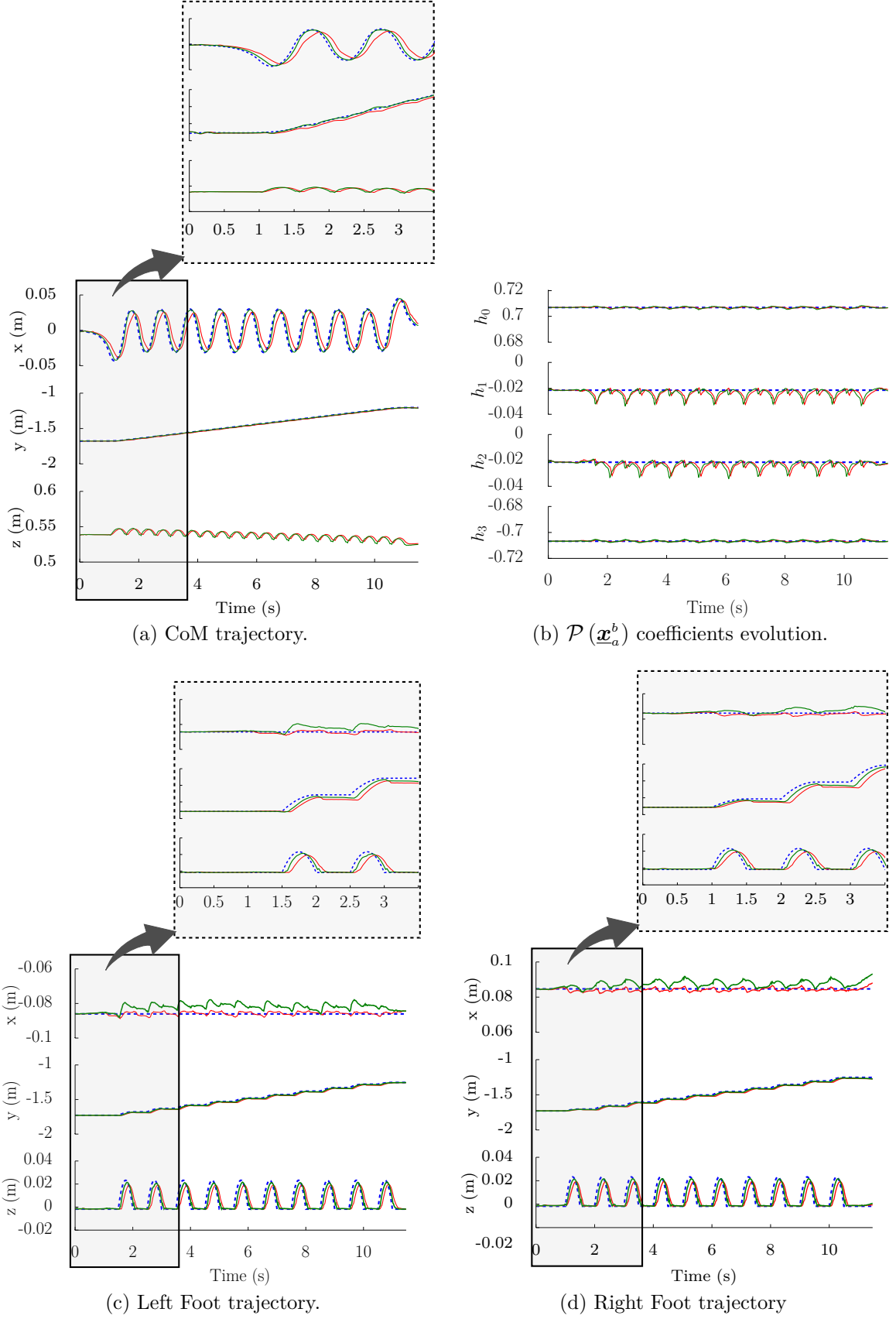


Figure 6.14: Comparison of the results obtained from the execution of the first pattern controlled using strategies 1 and 2. The *dashed blue line* represents the reference, the *solid red (darker) line* represents the measured values of the system controlled by strategy 1, and the *solid green (lighter) line* represents the measured values of the system controlled by strategy 2.

Aiming to compare the performance of both controllers quantitatively, two indexes are used: the Mean Absolute Error (MAE) and the Integrated Absolute Variation of the Control signal (IAVU), which are calculated according to (6.1) and (6.2).

$$MAE = \frac{1}{n} \sum_{i=1}^n \|\hat{x}(i) - x(i)\| = \frac{1}{n} \sum_{i=1}^n \|e(i)\|, \quad (6.1)$$

$$IAVU = \int_0^{time} \left\| \frac{d\dot{q}(t)}{dt} \right\| dt, \quad (6.2)$$

where $\hat{x}(i)$ is the reference and $x(i)$ is the measured value of a variable of interest, and $\dot{q}$ is the control signal. In all cases, the control signal $\dot{q}$ represents the reference velocities generated to all joints, even though some of them are not actuated in some cases. In those cases, the value of the control signal for those particular joints is always 0. The *MAE* index is calculated for the global error as well as for the error in each controlled variable—i.e. the CoM projection on the ground, the orientation of $\underline{x}_a^b$, and $\underline{x}_r$ —separately.

In Tables 6.3a and 6.3b, the calculated indexes for both strategies 1 and 2 are presented regarding the four walking patterns aforementioned. We can notice that, in most part of the cases, the MAE indexes computed for the strategy 2 are lower. On the other hand, the control effort, represented by the IAVU index, is greater for the strategy 2.

Walking Pattern	1	2	3	4
<i>MAE</i>	0.0200	0.0195	0.0200	0.0230
<i>MAE_{CoM}</i>	0.0141	0.0135	0.0129	0.0144
<i>MAE_{$\underline{x}_a^b$}</i>	0.0130	0.0116	0.0136	0.0155
<i>MAE_{$\underline{x}_r$}</i>	0.0038	0.0049	0.0038	0.0047
<i>IAVU</i>	1.0763×10^4	1.5701×10^4	1.5904×10^4	1.7826×10^4

(a) Control Strategy 1.

Walking Pattern	1	2	3	4
<i>MAE</i>	0.0117	0.0104	0.0157	0.0164
<i>MAE_{CoM}</i>	0.0047	0.0067	0.0094	0.0101
<i>MAE_{$\underline{x}_a^b$}</i>	0.0094	0.0066	0.0113	0.0111
<i>MAE_{$\underline{x}_r$}</i>	0.0040	0.0029	0.0031	0.0035
<i>IAVU</i>	2.0752×10^4	1.9741×10^4	1.9787×10^4	2.1166×10^4

(b) Control Strategy 2.

Table 6.3: Index comparison between strategies 1 and 2.

In order to verify the robustness of the system controlled by the strategy 2, we executed a walking motion defined by the pattern 3, while the robot's arms moves from an upright configuration to a neutral configuration.

Figure 6.15 shows the results obtained in this experiment, where the robot's CoM trajectory with respect to the inertial frame, the complete pose of the relative variable, and the orientation of the absolute variable are depicted. Moreover, in Figure 6.16, some

simulation snapshots taken during the experiment are presented, and Figure 6.17 shows the evolution of the joints angles of the robot's arms.

We observe that the robot tips over in the first seconds of the experiment, which occurs because the movement of the arms represents a disturbance that the controller is not able to reject, and the system becomes unstable. With the aim of preventing this behavior, we propose to include the arms joints in the GCoM tracking, as presented in Section 5.3. This control strategy is given by (5.24), and is labeled herein as “strategy 3.”

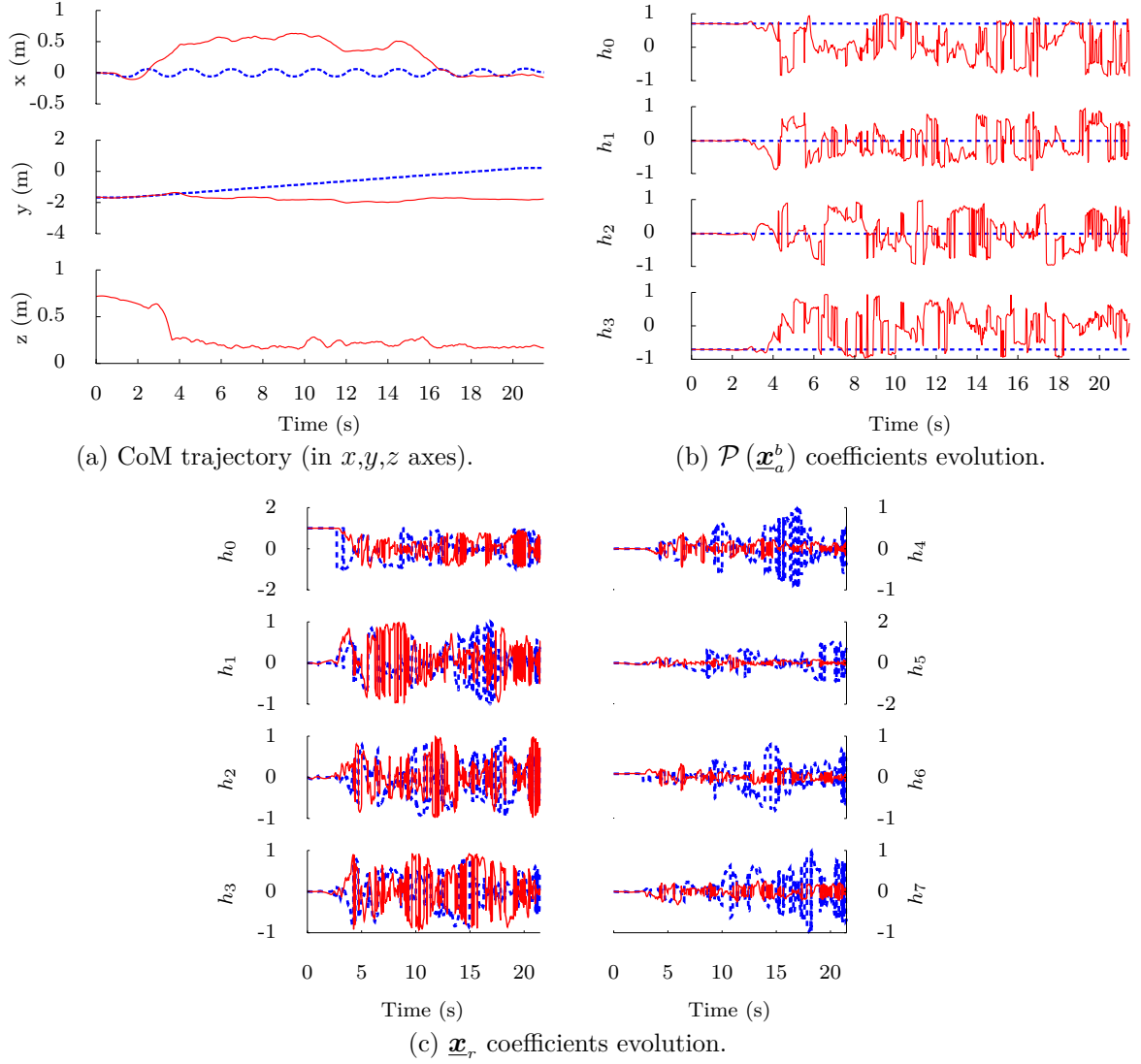
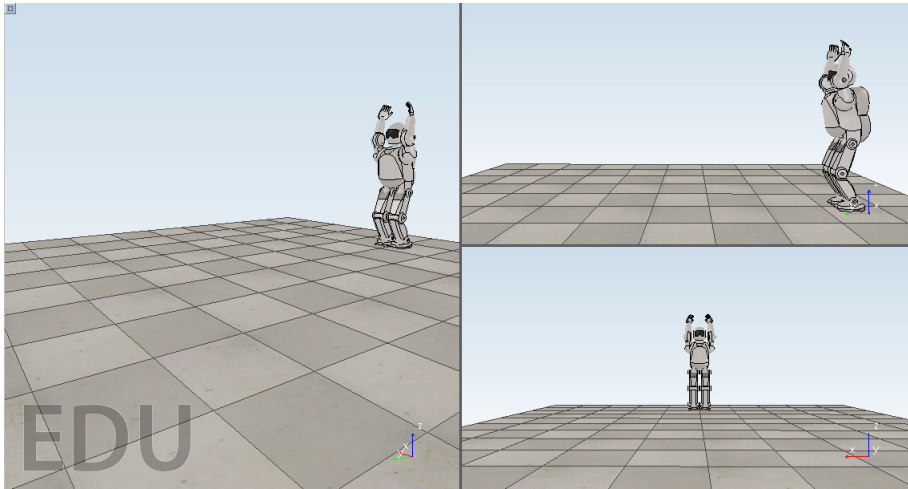
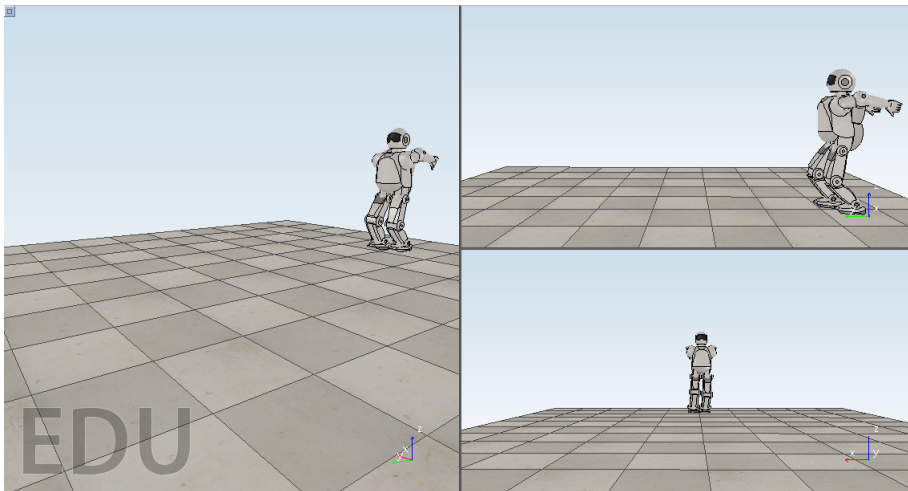


Figure 6.15: Results obtained from the execution of a walking motion controlled using the strategy 2, where the arms were actuated. The *dashed blue line* represents the reference and the *solid red line* represents the measured values.

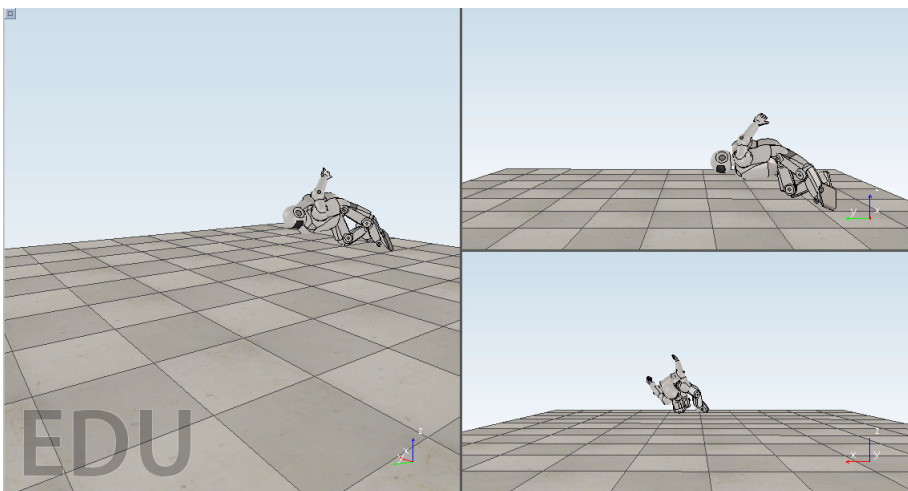
Since no reference trajectory is defined to the arms end-effectors, the controller can generate unnatural movements of the arms. This occurs because the control signal is calculated using the pseudo-inverse of the Jacobian Matrix, which gives the minimum norm solution for the current time instant, regardless the movement executed in the past. As the system's dynamics is unknown, these unnatural movements can generate large



(a) Simulation snapshot at 0 s.



(b) Simulation snapshot after 2 s.



(c) Simulation snapshot at 3 s.

Figure 6.16: Simulation snapshots of a walking motion controlled by the strategy 2, while moving the arms.

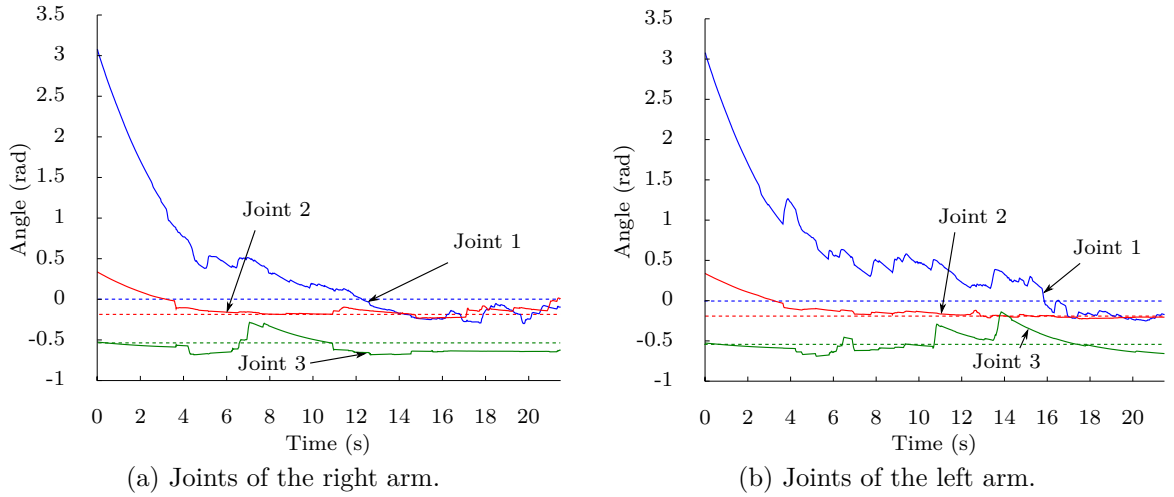


Figure 6.17: Angles of the arms joints during the walking motion controlled using the strategy 2, where the arms were actuated. The *dashed lines* represent the references and the *solid lines* represent the measured values.

disturbances after walking for a long period. To verify the behavior of the system under these unnatural movements, an experiment was performed using the strategy 3, setting the number of steps $N_s = 50$.

Figure 6.18 shows the results obtained in this experiment, where the robot's CoM trajectory with respect to the inertial frame, the complete pose of the relative variable, and the orientation of the absolute variable are depicted. Furthermore, in Figure 6.19, some simulation snapshots taken during the experiment are presented, and the angles of the arms joints are shown in Figure 6.20.

As we observe in Figures 6.19 and 6.20, the robot's arms move towards backward direction during the experiment. The angular momentum around the CoM increases as the arms move, which generates rotations of the robot's body. These rotations represent disturbances for the controlled system, more specifically in the absolute variable, and depending on the magnitude of the disturbances, the controller is not able to correct the body orientation. In Figure 6.18b, we can notice periodic deviations in the coefficients of $\mathcal{P}(\underline{x}_a^b)$, which increases over time, until the robot tips over.

In order to prevent this behavior, it is desirable to keep the arms close to the neutral position while it contributes to the CoM tracking. Since the humanoid robot is redundant, it is possible to include a secondary task operating in the null space of the Jacobian matrix of the locomotion task, using the strategy 4 given by (5.27), where $f(q)$ is a function that leads the robot's joints to angles defining a desired configuration, which, in this case, is the initial configuration. This function is given by

$$f(q) = \sum_{i=1}^n \frac{1}{2} [q(i) - q_d(i)]^2,$$

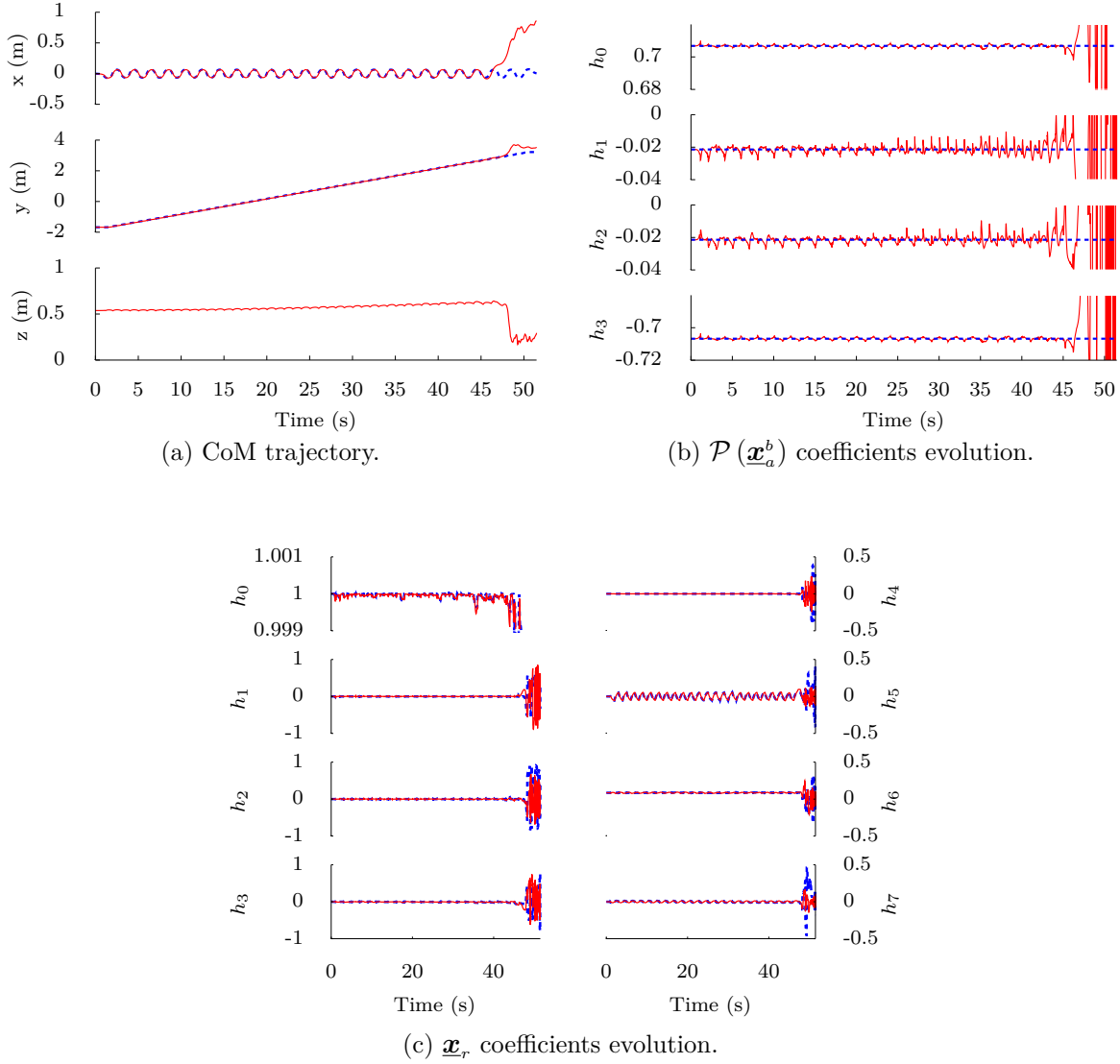
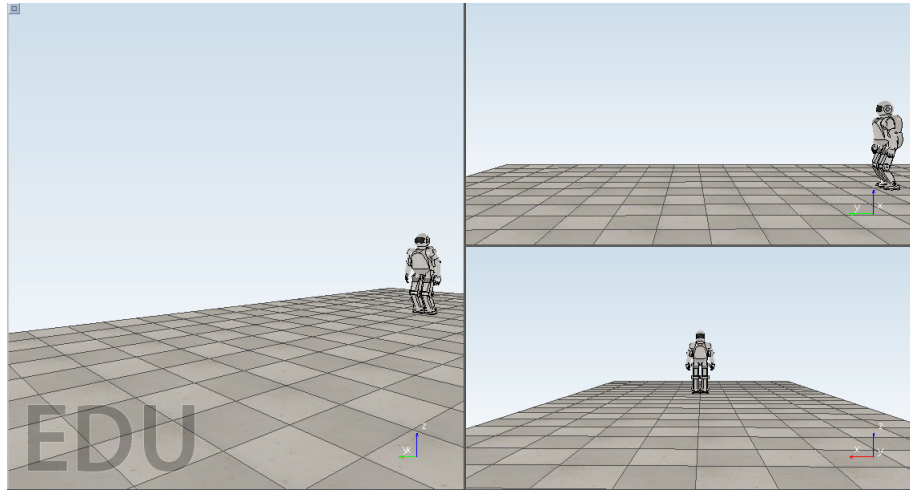


Figure 6.18: Results obtained from the execution of a walking motion of 50 footsteps, controlled using the strategy 3. The *dashed blue line* represents the reference and the *solid red line* represents the measured values.

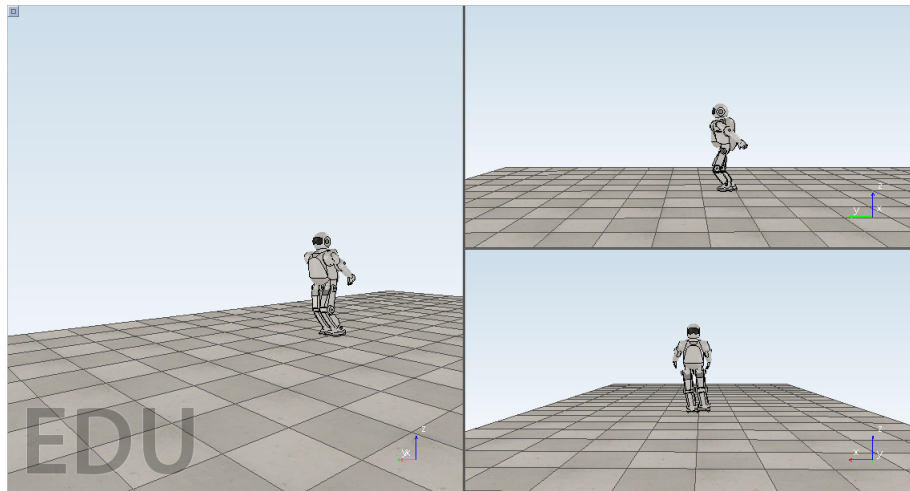
where $q = q_{wb}$ represents the whole-body joints vector, n is the total number of joints, which in the particular case of ASTI robot is 18, since the head joints were not actuated, and q_d is the vector of the desired angles. The Jacobian matrix obtained from (5.25) is given by

$$J_f = \begin{bmatrix} q(1) - q_d(1) \\ \vdots \\ q(n) - q_d(n) \end{bmatrix}^T.$$

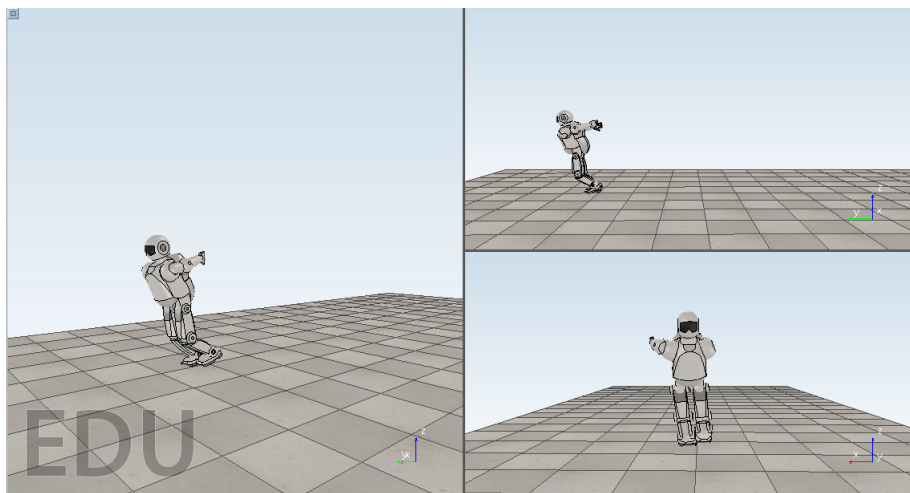
The last experiment was performed again using the strategy 4, with $\lambda_2 = 0.04$. The results are depicted in Figure 6.21. In Figure 6.21b, we notice that the deviations in $\mathcal{P}(\underline{x}_a^b)$ do not increase over time and that all variables followed the given references during the entire experiment. Moreover, the arms joints are kept around the desired angles, as shown



(a) Simulation snapshot at 0 s.



(b) Simulation snapshot after 25 s.



(c) Simulation snapshot after 45 s.

Figure 6.19: Simulation snapshots of a walking motion of 50 footsteps, controlled using the strategy 3, which includes the arms in the control law.

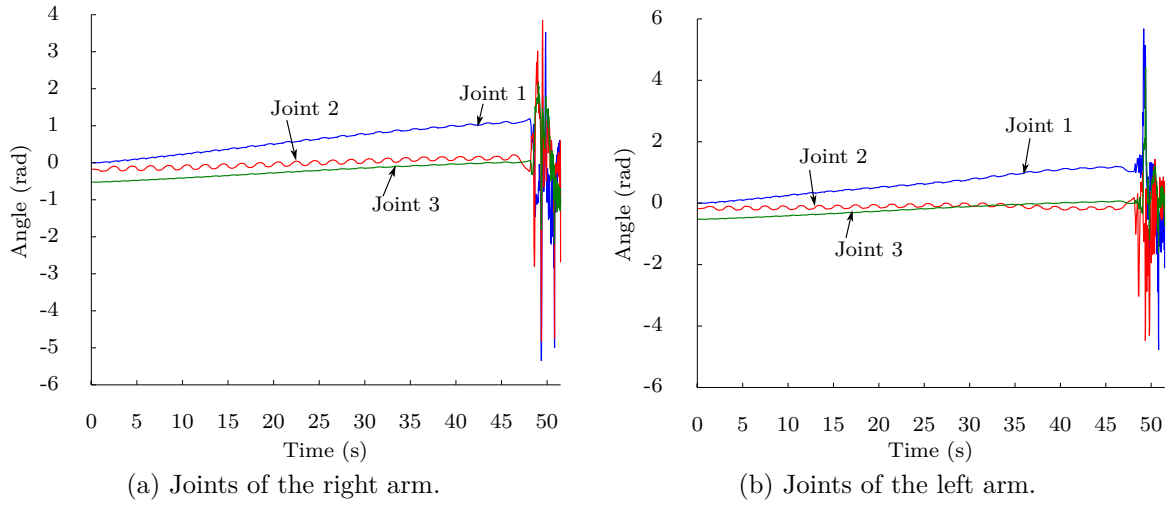


Figure 6.20: Arms joints angles during a walking motion of 50 footsteps, controlled using the strategy 3, which includes the arms in the control law.

in Figure 6.22. .

With the aim of comparing the performances of the strategy 2 and the strategy 4, the indexes MAE and IAVU were calculated for the strategy 4 regarding the four walking patterns aforementioned, which are shown in Table 6.4.

Walking Pattern	1	2	3	4
MAE	0.0164	0.0105	0.0212	0.0165
MAE_{CoM}	0.0068	0.0075	0.0125	0.0111
$MAE_{\underline{x}_a^b}$	0.0131	0.0062	0.0152	0.0104
$MAE_{\underline{x}_r}$	0.0052	0.0022	0.0041	0.0026
$IAVU$	2.0316×10^4	1.8739×10^4	1.7633×10^4	1.9012×10^4

Table 6.4: Control Strategy 4 comparison indexes.

We notice that there is no significant enhancement in the MAE indexes regarding the strategy 2, however, the control effort of the strategy 4 is smaller. It is an interesting fact, because in the second case all joints are actuated, whereas in the first case only the legs joints are actuated, indicating that the effort on the legs joints is lower when the strategy 4 is adopted. The low control effort is a big advantage in the particular case of autonomous robots, since it represents a low energy expenditure. Furthermore, large control efforts might generate mechanical stress in the joints, reducing the durability of the robot.

The charts comparing the trajectories performed using the strategy 2 and the strategy 4, for the four patterns, are presented in Figures B.6, B.7, B.8, and B.9 of Appendix B.

The experiment consisting of the execution of a walking motion, while the arms move from an upright configuration to the neutral configuration using strategy 1, depicted in Figure 6.15, was repeated using the strategy 4. The results are shown in Figures 6.23, 6.24, and 6.25. We notice that, using the strategy 4, all controlled variables follow the

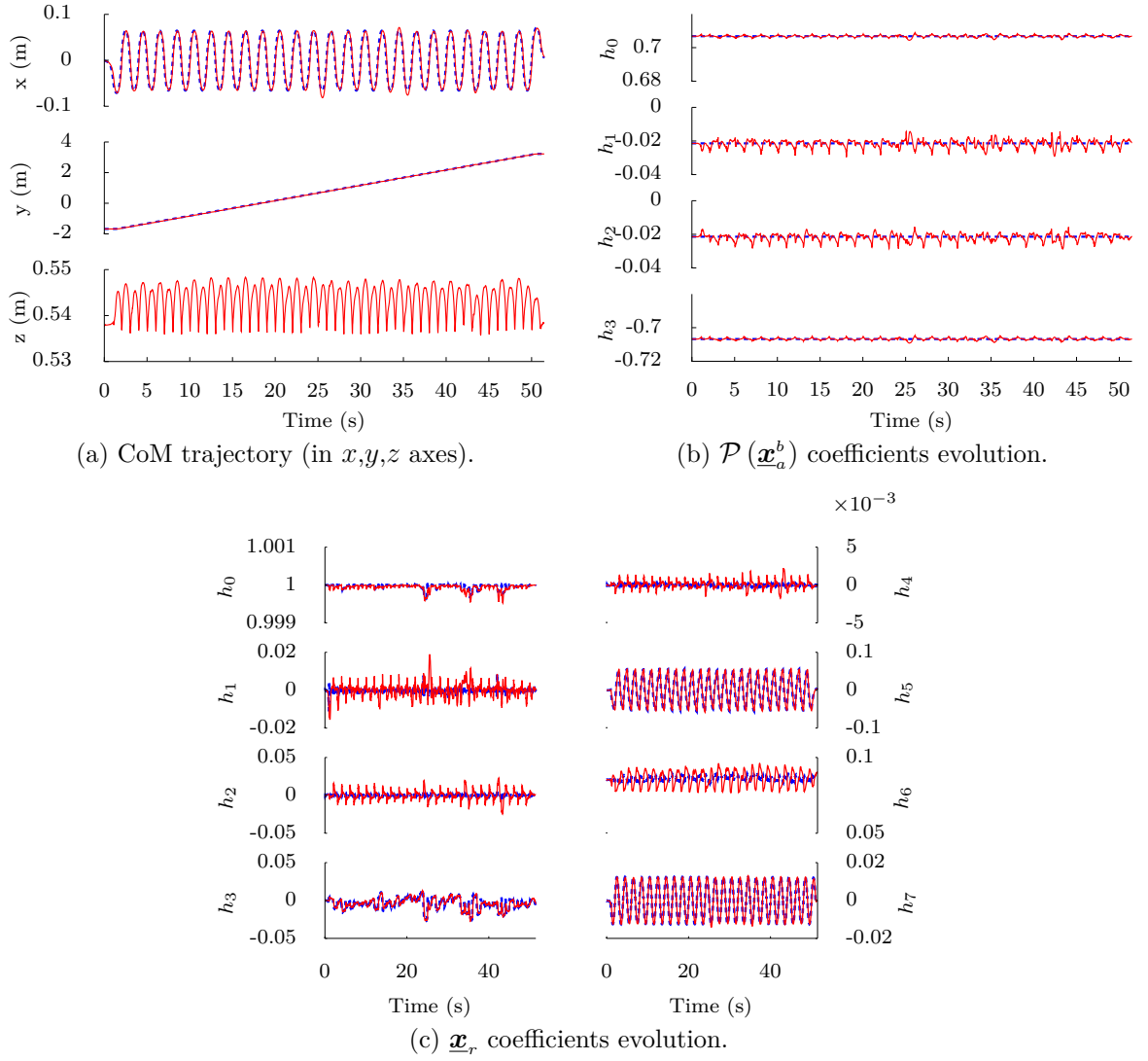


Figure 6.21: Results obtained from the execution of a walking motion of 50 footsteps, controlled using the strategy 4, which includes the arms and has a secondary the control law. The *dashed blue line* represents the reference and the *solid red line* represents the measured values.

references, and the system is able to correct the disturbances generated by the arms motion. Furthermore, comparing Figures 6.17 and 6.24, the arms joints execute smoother trajectories to reach the final configuration.

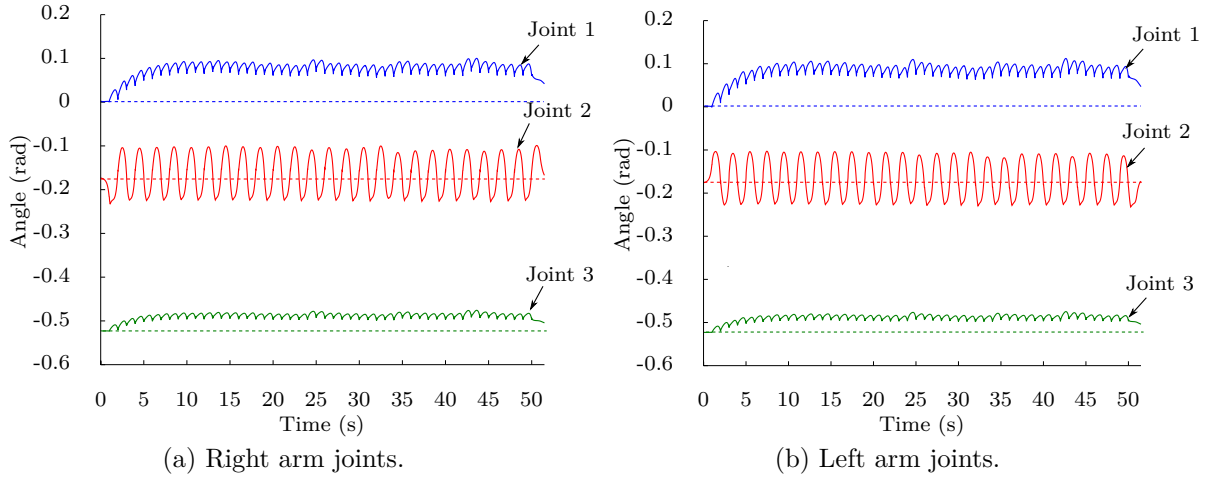


Figure 6.22: Angles of the arms' joints during a walking motion of 50 footsteps, controlled using the strategy 4, which includes the arms and has a secondary the control law. The *dashed lines* represent the neutral configuration and the *solid lines* represent the measured values.

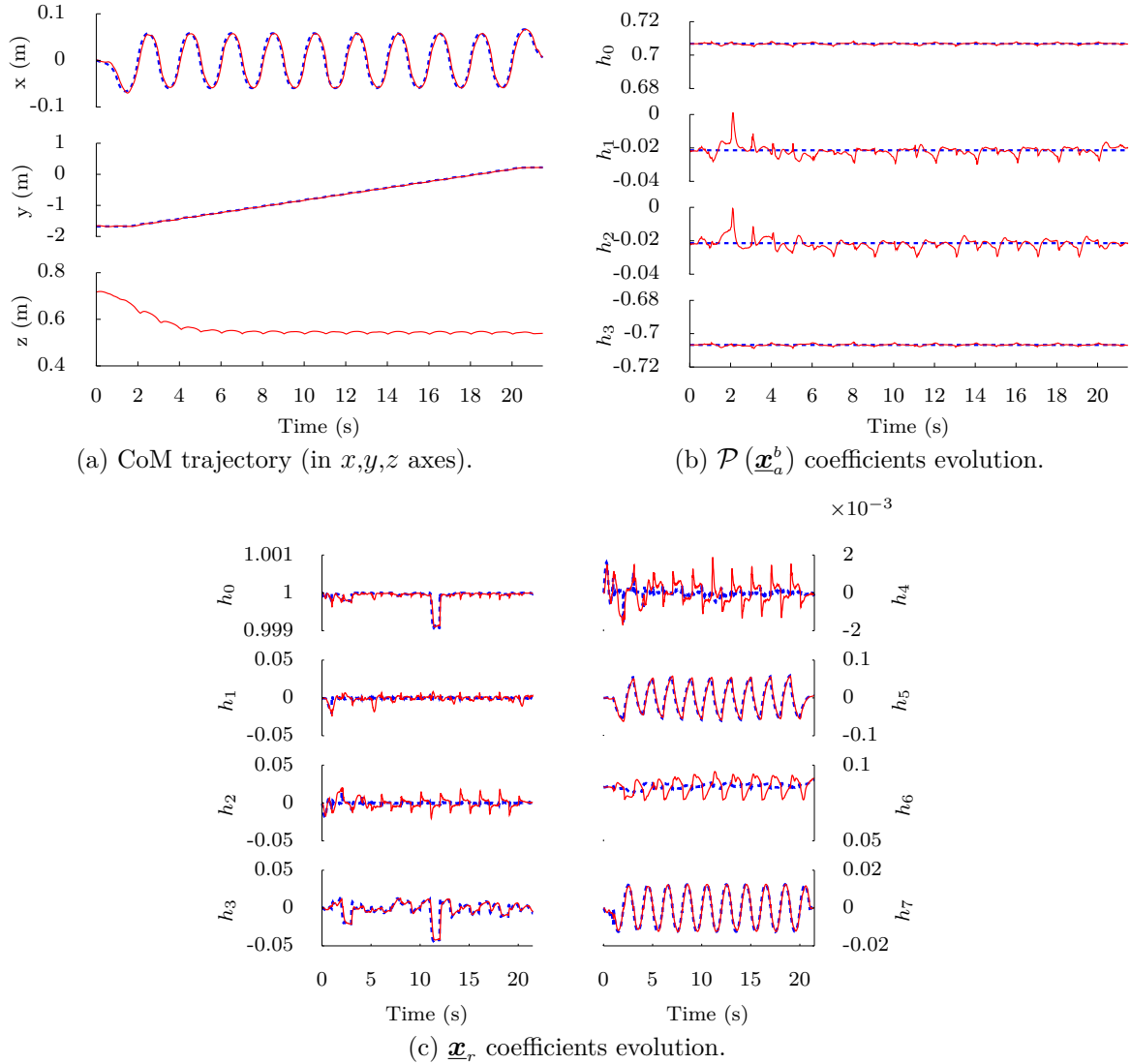
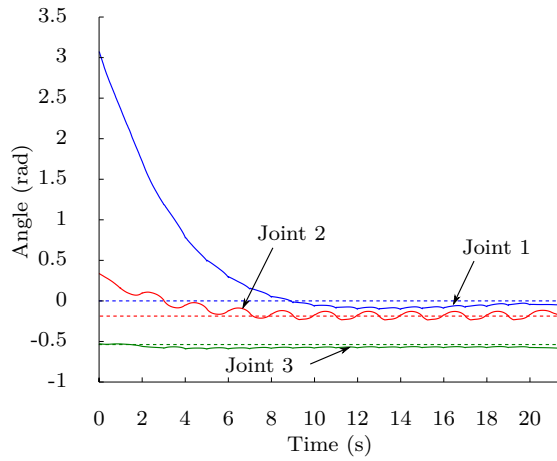
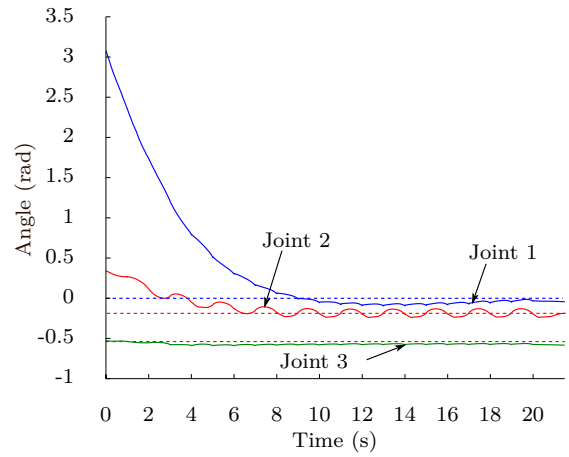


Figure 6.23: Results obtained from the execution of a walking motion controlled using the strategy 4, which includes the arms and has a secondary the control law, where the arms were actuated. The *dashed blue line* represents the reference and the *solid red line* represents the measured values.

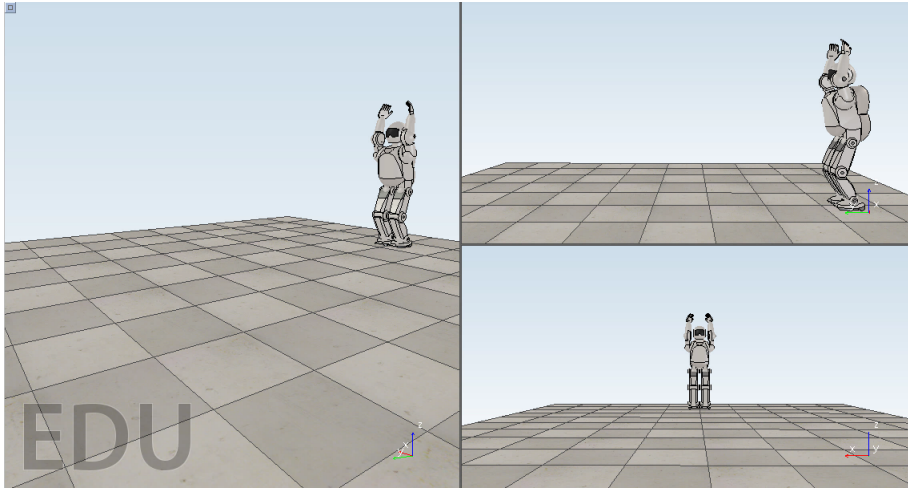


(a) Right arm.

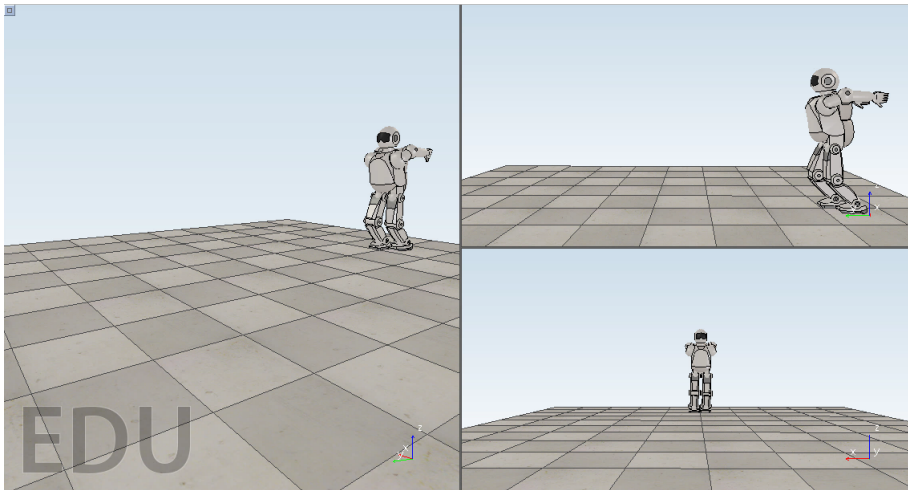


(b) Left arm.

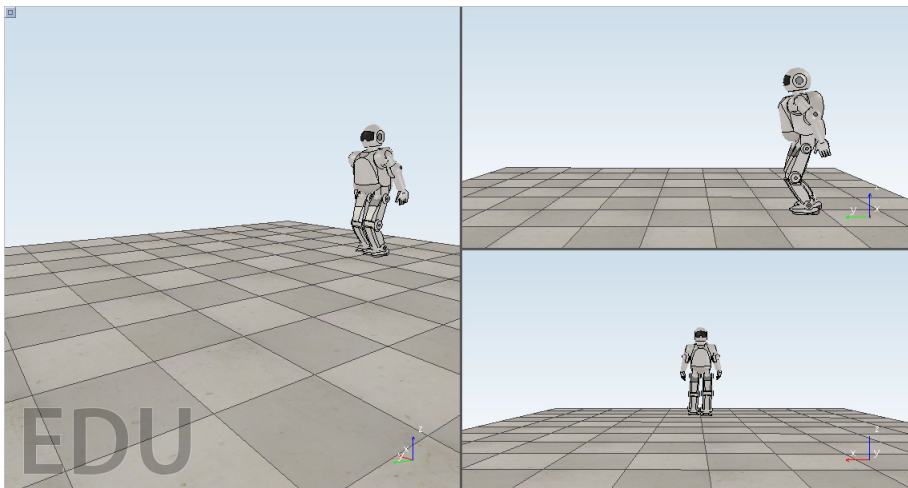
Figure 6.24: Angles of the arms joints during the walking motion controlled using the strategy 4. The *dashed lines* represent the references and the *solid lines* represent the measured values.



(a) Snapshot of simulation at 0 s.



(b) Snapshot of simulation after 2 s.



(c) Snapshot of simulation after 5 s.

Figure 6.25: Simulation snapshots of a walking motion controlled by the strategy 4, while moving arms.

The ability to change the walking direction is an important feature since it allows the

robot to reach various locations in an environment, using path planning methods. Thus, in the final experiment, we tested a walking motion in a circular path, which is defined by the radius, the number of footsteps, and the step length.

We defined a semi-circle shaped path with radius of 1m, 65 footsteps, and a step length of 5cm. The results obtained in these experiments are depicted in Figure 6.26, where we notice that the robot successfully followed the given references. Furthermore, in Figure 6.27, the footprints executed by the robot during the locomotion are shown, making it clear the successful execution of the semi-circle shaped path.

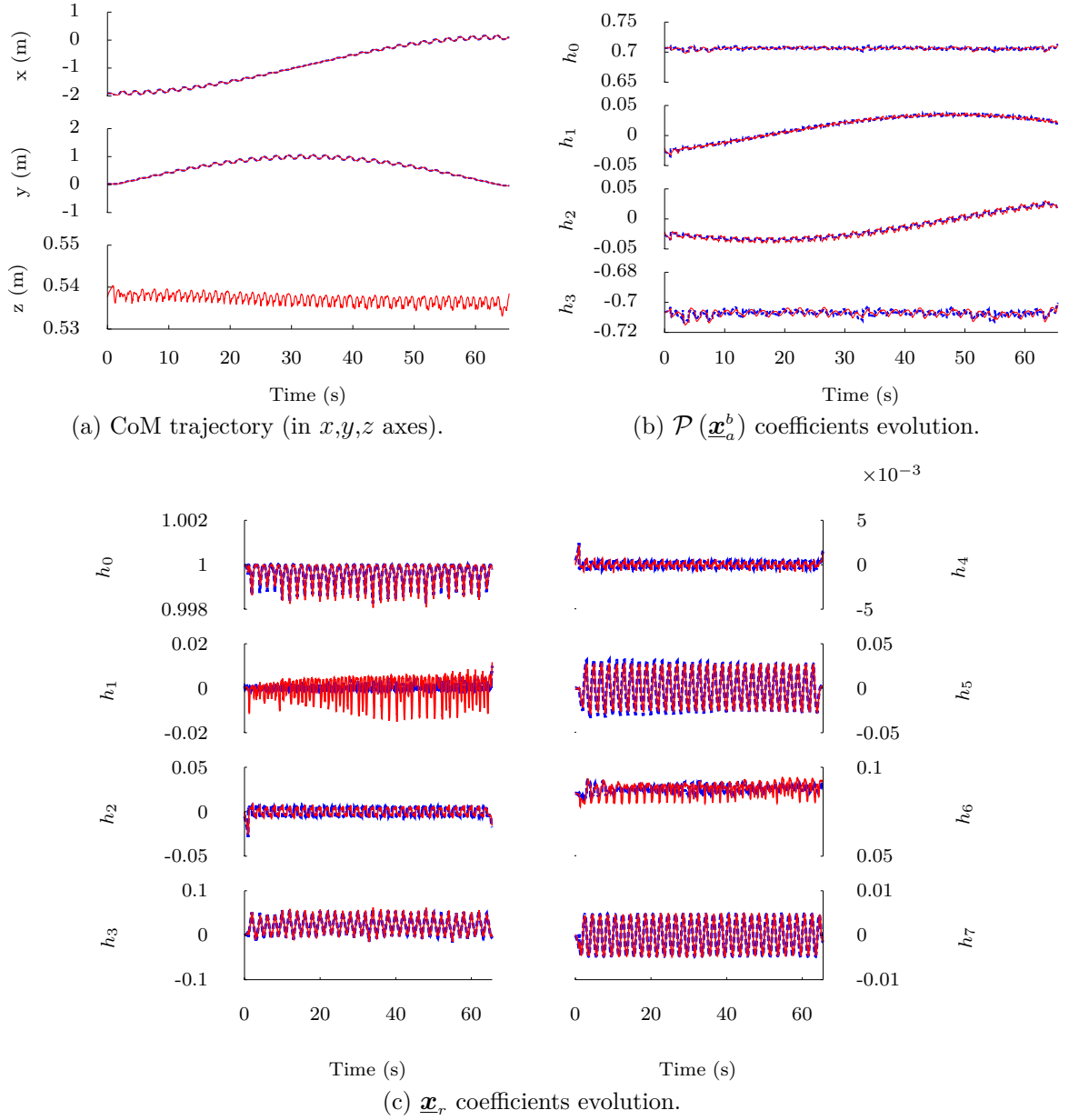


Figure 6.26: Results obtained from the execution of a walking motion in a semi-circle shaped path, controlled using the strategy 4. The *dashed blue line* represents the reference and the *solid red line* represents the measured values.

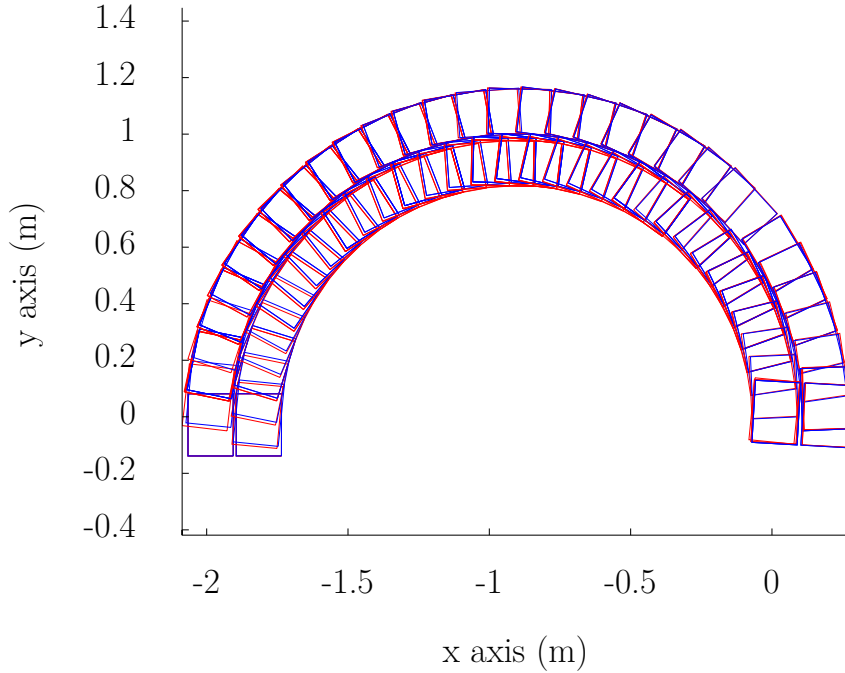


Figure 6.27: Footprints executed in a walking motion in a semi-circle shaped path, controlled using the strategy 4. The *blue (darker) line* represents the desired footprints and the *red (lighter) line* represents the actual footprints.

6.5 Chapter Overview

This chapter presented the experiments performed in this work and discussed the obtained results.

Section 6.1 detailed the specification of the simulation environment where the experiments were executed, including the features of the computer, and the specification of the robot used.

Section 6.2 presented the methodology adopted to validate the models calculated using the proposed methods. The results showed that the information estimated using the model was very close from the information read from the simulation environment.

Section 6.3 showed the experiments performed to verify the accuracy of the CoM tracking, by letting the robot's feet fixed, and defining trajectories for the CoM. The actual CoM successfully followed the given reference.

Section 6.4 presented the experiments involving the walking motion, using the proposed control strategies. In the first experiment, we verified that the robot successfully executed the defined walking motion, using the first control strategy (without feed-forward term). However, an offset between the reference and the actual trajectory was observed. The second experiment applied the second control strategy (with feed-forward term), and we noticed a decrease of the aforementioned offset. In the third experiment, we verified that the robot was not able to execute a walking motion while moving its arms simultaneously, concluding that this behavior represented a disturbance in the system. In the fourth

experiment, we verified the robot's behavior after including the arms joints in the tracking of the GCoM, and we noticed an undesired movement of the arms, which makes the system unstable after a long walking motion. Thus, in the fifth experiment, we verified the effect of adding a secondary law in the null space of the Jacobian matrix of locomotion, and the robot executed the walking motion without the aforementioned large movement of the arms. In the sixth experiment, we repeated the third experiment using the fourth control strategy, and verified that the robot's successfully executed the walking motion while moving its arms. In the seventh experiment, we showed that the robot was able to walk with varying direction using the proposed methods.

7

Conclusions and Future Works

This chapter summarizes the work presented in this master thesis. Furthermore, some shortcomings and proposals for future works are presented.

7.1 Overview

This work presented a kinematic modeling method for a humanoid robot based on dual quaternion (DQ) algebra and proposed a control strategy based on the work of Park & Lee (2013), which fulfills the kinematic constraints for a balanced gait.

7.1.1 Kinematic Modeling Method

The modeling method consisted of three stages, as follows:

1. **The limbs modeling**, where the Denavit-Hartenberg (DH) convention was used to define the limbs geometric parameters, and the limbs Forward Kinematics Models (FKMs) were obtained by using DQ algebra. Furthermore, the limbs Differential Forward Kinematics Models (DFKMs) were calculated by taking the first derivative of the FKMs with respect to the respective limb joints vector.
2. **The center of mass (CoM) modeling**, where an equivalent model for the CoM of each link was used to calculate the global CoM model.

3. **The legs cooperative behavior modeling**, where the CDTS was used to define the cooperative variables, which describe the relation between the feet, and can represent the kinematic constraints of locomotion in a compact way.

All models were obtained algebraically, without using any software for symbolic computation. Thus, one can easily apply the method to different humanoid robots, simply changing the geometric parameters.

To validate the robot's kinematic model, arbitrary commands were sent to the joints and the kinematic data was captured. The actual poses and velocities of each end-effector were directly read from the simulated environment, as well as the global CoM and its linear velocity. Using the limbs FKMs and DFKMs, and also the CoM kinematic model, the estimated values for these variables were computed. Comparing both estimated and actual data, we noticed that the estimated values were very close to the actual values, consequently the modeling method was considered accurate, providing reliable information.

7.1.2 Gait and Balance Control Strategies

The presented control strategy was designed using the pseudo-inverse of a Jacobian matrix representing the locomotion task, which is obtained by stacking the Jacobian matrices of the cooperative variables and the CoM. The reference trajectories for the walking motion were defined by using the method proposed by Kajita et al. (2003), where the desired feet poses are defined a priori and used to calculate the CoM trajectory.

In order to verify the accuracy of the CoM tracking, some simulations were performed, where the robot's feet were kept stationary, and a set of different trajectories were defined for the CoM. The obtained results showed that the actual CoM successfully followed the given references, while the body posture was kept balanced.

Four walking patterns were defined and some experiments were performed to compare and analyze the presented control strategies without feed-forward. The results showed that the robot executed a balanced walking motion, when controlled using the original control strategy. However, as expected, an offset between the reference and the actual trajectories was observed. Thus, we proposed the inclusion of a feed-forward term in the control strategy, which resulted in a substantial decrease in the offset. Two indexes were used to compare the control strategies: the Mean Absolute Error (MAE) and the Integrated Absolute Value of the Control Signal (IAVU). We verified that the error over the trajectory is lower for the controller with the feed-forward term, whereas the control effort is greater.

When the system is subject to a substantial disturbance, created by the movement of the robot's arms during the walking motion, we verified that the controller was not able to stabilize the system, and the robot tipped over. Therefore, we proposed to use the entire CoM Jacobian matrix, including the arms joints in the control strategy. Moreover,

an additional task was included to operate in the null space of the Jacobian matrix of the locomotion task, in order to keep the arms near to a neutral position, so as to avoid the appearance of angular momentum created by unnatural movements of the arms. Comparing the control strategies with and without the arms joints actuation, we did not verify significant improvements regarding the errors over the trajectory. However, the control effort was lower for the case where the arms were actuated, even though the number of actuated joints had been larger. It indicated that the effort on the legs joints was lower when the arms had been included in the GCoM tracking, as well as the global control effort. Besides the reduction of the energy consumption, another advantage of the low control effort might be the reduction of mechanical stress in the joints.

The complete control strategy was also tested in a curvilinear trajectory, and the simulated results showed that the robot was able to perform the locomotion successfully.

In summary, regarding the proposed objectives of this work, namely the development of the kinematic model and the gait and balance controller for a humanoid robot using the DQ algebra, both were fulfilled. Furthermore, some improvements over the original control strategy proposed by Park & Lee (2013) were proposed and validated in simulation.

7.2 Future works

This work is part of a larger project whose goal is to build a testbed for a real humanoid robot. The main objective of this stage was to introduce the methodologies to the kinematic modeling and control of the gait and balance of the robot. However, there are some shortcomings that might be improved in the future:

- **Improving the understanding about the simulation software:** the simulation environment has its own collision detection module, besides the specific features of each object, and we cannot assure that some observed behaviors, like the disturbance in the x -axis during the walking motion, occurs only due the interaction between the robot and the environment. Thus, a further study about the modules of the software must be done. One way to do this is to perform specific experiments in the software, whose results are well-known, like dropping objects, for example, and comparing the obtained results with the expected ones.
- **Implementation and validation of the presented control strategies in a real robot:**¹ all the tests and experiments were executed in simulation, but results obtained using real robots are essential to validate conceptual works.
- **Proving the asymptotic stability of the proposed control strategies:** the stability proof of the proposed control strategies was mathematically derived in

¹A project to build the robot is under development in the research group MACRO.

Section 5.4, however the asymptotic stability was not proven. We suggest to use the Barbalat Lemma to do it.

- **Determining the time response of the joints, and possibly reducing it by adjusting the joint controller gains:** the system's performance is influenced by the physical limitations, like the maximum torques that each joint can handle. Furthermore, since each joint has its own embedded controller, the whole controlled system can be regarded as a cascade control, where the locomotion controller is at the external loop, and the joint controller is at the internal loop. Consequently, the stability of this system also depends on the time response of the joints. The simulation environment do not provide much information about the joint dynamic model, but a black-box modeling approach can be used to determine the time response of the joints.
- **Improving the control strategies:** the presented control strategies were not designed to guarantee the rejection of disturbances, and it is an important improvement to be considered in the future.
- **Implementing in another programming language:** it is desirable to use the controllers in a real-time operating system, or at least to have a reasonable latency time, to allow its implementation in a real robot. Thus, we suggest the implementation of the controllers in another programming language instead of MATLAB, like C++, for instance.
- **Working with dynamic walking instead of the quasi-static approach:** the quasi-static locomotion does not allow the robot to walk on rough terrains and limits the walking velocity. Working with the dynamic approach gives the possibility to execute more complex tasks, like stairs climbing, running, and walking on a sloping floor, for example.

Bibliography

- Adorno, B. V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration*. PhD thesis.
- Adorno, B. V., Fraisse, P., & Druon, S. S. (2010). Dual position control strategies using the cooperative dual task-space framework. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3955–3960).
- Alcaraz-Jimenez, J. J., Herrero-Perez, D., & Martinez-Barbera, H. (2013). Robust feedback control of ZMP-based gait for the humanoid robot Nao. *The International Journal of Robotics Research*, 32(9-10), 1074–1088.
- Ali, M. a., Park, H. a., & Lee, C. S. G. (2010). Closed-form inverse kinematic joint solution for humanoid robots. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 704–709).
- Ames, A. D., Tabuada, P., Schürmann, B., Rungger, M., & Grizzle, J. W. (2015). First Steps toward Formal Controller Synthesis for Bipedal Robots. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control* (pp. 209–218).: ACM.
- André, C., Bezerra, D., & Zampieri, D. E. (2004). Biped Robots : the State of Art. In *International Symposium on History of Machines and Mechanism* (pp. 371–389). Springer Netherlands.
- Arnold, W. F. & Laub, A. J. (1984). Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations. *Proceedings of the IEEE*, 72(12), 1746–1754.
- Bernstein, D. S. (2005). *Matrix Mathematics*, volume 1. Princeton University Press.
- Boulic, R., Mas, R., & Thalmann, D. (1995). Inverse Kinetics for Center of Mass Position Control and Posture Optimization. (1).
- Chevallereau, C., Bessonnet, G., Abba, G., & Aoustin, Y. (2010). *Bipedal Robots: Modeling, Design and Walking Synthesis*.

- Chiacchio, P., Chiaverini, S., & Siciliano, B. (1996). Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *Journal of dynamic systems, measurement, and control*, 118(4), 691–697.
- Choi, Y., Kim, D., Oh, Y., & You, B.-j. J. (2007). Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion. *IEEE Transactions on Robotics*, 23(6), 1285–1293.
- Clifford, W. K. (1873). *A preliminary sketch of biquaternions*.
- Cotton, S., Murray, A., & Fraisse, P. (2009). Estimation of the center of mass using statically equivalent serial chain modeling. In *ASME 2009 IDETC/CIE* (pp. 1–7).
- Feng, S. (2008). A simple trajectory generation method for biped walking. , *Robotics and Vision, 2008. ICARCV 2008.*, (December), 17–20.
- Fukuda, T., Hasegawa, Y., Sekiyama, K., & Aoyama, T. (2012). *Multi-Locomotion Robotic Systems*, volume 81. Springer.
- Galdeano, D., Chemori, A., Fraisse, P., Galdeano, D., Chemori, A., Fraisse, P., & Pid, A. N. (2014). A Nonlinear PID Stabilizer With Spherical Projection for Humanoids : From Concept to Real-time Experiments To cite this version : .
- Galdeano, D., Chemori, A., Krut, S., & Fraisse, P. (2013). Optimal pattern generator for dynamic walking in humanoid robotics. *2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013*, (pp. 1–6).
- Graf, C. (2010). A Closed-loop 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid. (pp. 18–22).
- Hamilton, W. R. (1844). II. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine Series 3*, 25(163), 10–13.
- Hartenberg, R. S. & Denavit, J. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Transaction ASME J Appl Mech*, 22, 215–221.
- Hirai, K., Hirose, M., Haikawa, Y., & Takenaka, T. (1998). The development of Honda humanoid robot. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 2, 1321–1326.
- Hof, A. L. (2008). The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human Movement Science*, 27(1), 112–25.
- Hofmann, A., Popovic, M., & Herr, H. (2009). Exploiting angular momentum to enhance bipedal center-of-mass control. In *Proceedings - IEEE International Conference on Robotics and Automation* (pp. 4423–4429).: Ieee.

- Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., & Tanie, K. (2001). Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, 17(3), 280–289.
- Kajita, S., Kanehiro, F., Kando, K., Yokoi, K., & Hirukawa, H. (2001). The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 1, 239–246.
- Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)* (pp. 1620–1626).: IEEE.
- Kajita, S., Koyachi, N., Kaneko, K., Yokoi, K., Arai, H., Komoriya, K., & Tanie, K. (1999). A high stability, smooth walking pattern for a biped robot. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1(May), 65–71.
- Katayama, T., Ohki, T., Inoue, T., & Kato, T. (1985). Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control*, 41(3), 677–699.
- Khalil, H. K. (2002). *Nonlinear Systems, Third Edition*.
- Kim, J.-h. (2007). Walking pattern generation of a biped walking robot using convolution sum. *2007 7th IEEE-RAS International Conference on Humanoid Robots*, (pp. 539–544).
- Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. G. (2014). Complete Analytical Forward and Inverse Kinematics for the NAO Humanoid Robot. *Journal of Intelligent & Robotic Systems*, 77(2), 251–264.
- Kuipers, J. B. (2002). *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press.
- Li, Z., Vanderborght, B., Tsagarakis, N. G., Colasanto, L., & Caldwell, D. G. (2012). Stabilization for the compliant humanoid robot COMAN exploiting intrinsic and controlled compliance. *Proceedings - IEEE International Conference on Robotics and Automation*, (August), 2000–2006.
- Mansard, N., Stasse, O., Evrard, P., & Kheddar, a. (2009). A versatile Generalized Inverted Kinematics implementation for collaborative working humanoid robots: The

- Stack Of Tasks. *Advanced Robotics, 2009. ICAR 2009. International Conference on*, (8), 1–6.
- Mason, M. (1985). On the scope of quasi-static pushing. In *Proc. 1985 Third Int. Symp. on Robotics*.
- Missura, M. & Behnke, S. (2011). Lateral capture steps for bipedal walking. In *IEEE-RAS International Conference on Humanoid Robots*, number October.
- Mori, M., MacDorman, K. F., & Kageki, N. (2012). The uncanny valley. *IEEE Robotics and Automation Magazine*, 19(2), 98–100.
- Oliveira, A. C. & Adorno, B. V. (2015). Balance Control of a Humanoid Robot Based on the Cooperative Dual Task-Space Framework. In *Anais do XII Simpósio Brasileiro de Automação Inteligente* (pp. 1–6). Natal.
- Park, H. A. & Lee, C. S. G. (2013). Cooperative-Dual-Task-Space-based whole-body motion balancing for humanoid robots. In *2013 IEEE International Conference on Robotics and Automation* (pp. 4797–4802).: IEEE.
- Park, J. & Kim, K. (1998). Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 4(May), 3528–3533.
- Phillips, C. B. & Badler, N. I. (1991). Interactive behaviors for bipedal articulated figures.
- Rohmer, E., Singh, S. P. N., & Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. *IEEE International Conference on Intelligent Robots and Systems*, (pp. 1321–1326).
- Rosheim, M. (2006). *Leonardo's Lost Robots*.
- Roussel, L., Canudas-De-Wit, C., & Goswami, a. (1998). Generation of energy optimal complete gait cycles for biped robots. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 3.
- Sciavicco, L., Siciliano, B., & Sciavicco, B. (2000). *Modelling and Control of Robot Manipulators*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2nd edition.
- Selig, J. M. (2005). *Geometric fundamentals of robotics*. Springer.
- Sentis, L. (2007). *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. PhD thesis.
- Siciliano, B. & Khatib, O. (2008). *Handbook of Robotics*. Springer.

- Slotine, J.-J. E. & Li, W. (1991). *Applied Nonlinear Control*, volume 62. Prentice Hall.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*.
- Stephens, B. (2007). Integral control of humanoid balance. In *IEEE International Conference on Intelligent Robots and Systems* (pp. 4020–4027).: Ieee.
- Sugihara, T., Nakamura, Y., Inoue, H., Sugihar, T., Inoue, H., Hiko Nakamur, Y., & Inoue, H. (2002). Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control. (May), 1404–1409.
- Tang, Z. & Er, M. J. (2008). Humanoid 3D gait generation based on inverted pendulum model. *22nd IEEE International Symposium on Intelligent Control, ISIC 2007. Part of IEEE Multi-conference on Systems and Control*, (October), 339–344.
- Tevatia, G. & Schaal, S. (2000). Inverse kinematics for humanoid robots. *Proceedings 2000 ICRA Millennium Conference IEEE International Conference on Robotics and Automation Symposia Proceedings Cat No00CH37065*, 1(Icra), 294–299.
- Tolani, D., Goswami, a., & Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5), 353–388.
- Toscano, G. S., Simas, H., & Castelan, E. B. (2014). Screw-based modeling of a humanoid biped robot. In *Anais do XX Congresso Brasileiro de Automática* (pp. 944–951). Belo Horizonte.
- Vukobratović, M., Frank, A. A., & Juricić, D. (1970). On the stability of biped locomotion. *IEEE transactions on bio-medical engineering*, (1), 25–36.
- Wieber, P.-b. (2006). Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, (pp. 137–142).
- Winter, D. (1995). Human balance and posture control during standing and walking. *Gait & Posture*, 3(4), 193–214.
- Yokoi, K., Kanehiro, F., Kaneko, K., Kajita, S., Fujiwara, K., & Hirukawa, H. (2004). Experimental Study of Humanoid Robot HRP-1S. *The International Journal of Robotics Research*, 23(4), 351–362.
- Zannatha, J. & Limon, R. (2009). Forward and Inverse Kinematics for a Small-Sized Humanoid Robot. *2009 International Conference on Electrical, Communications, and Computers*, (pp. 111–118).
- Zarrugh, M. & Radcliffe, C. (1979). Computer generation of human gait kinematics. *Journal of Biomechanics*, 12(2), 99–111.

A

DH Parameters of the Robot's Limbs

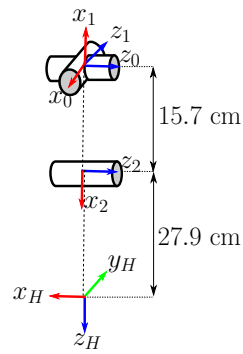


Table A.1: DH Parameters of right arm.

Link	$a(m)$	$\alpha(rad)$	$d(m)$	$\theta(rad)$
1	0	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
2	0.157	$\frac{\pi}{2}$	0	π
3	0	0	0	0

Figure A.1: Right arm scheme.

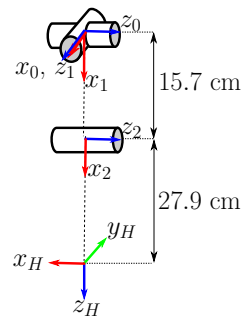


Table A.2: DH Parameters of left arm.

Link	$a(m)$	$\alpha(rad)$	$d(m)$	$\theta(rad)$
1	0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$
2	0.157	$-\frac{\pi}{2}$	0	0
3	0	0	0	0

Figure A.2: Left arm scheme.

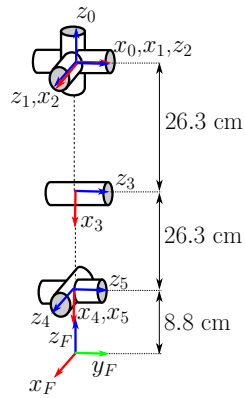


Figure A.3: Left and right leg scheme.

Table A.3: DH Parameters of left and right legs.

Link	$a(m)$	$\alpha(rad)$	$d(m)$	$\theta(rad)$
1	0	$\frac{\pi}{2}$	0	0
2	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
3	0.263	0	0	0
4	0.263	$\frac{\pi}{2}$	0	0
5	0	$-\frac{\pi}{2}$	0	0
6	0	0	0	0

B

Remainder Results

B.1 Center of Mass Control Results

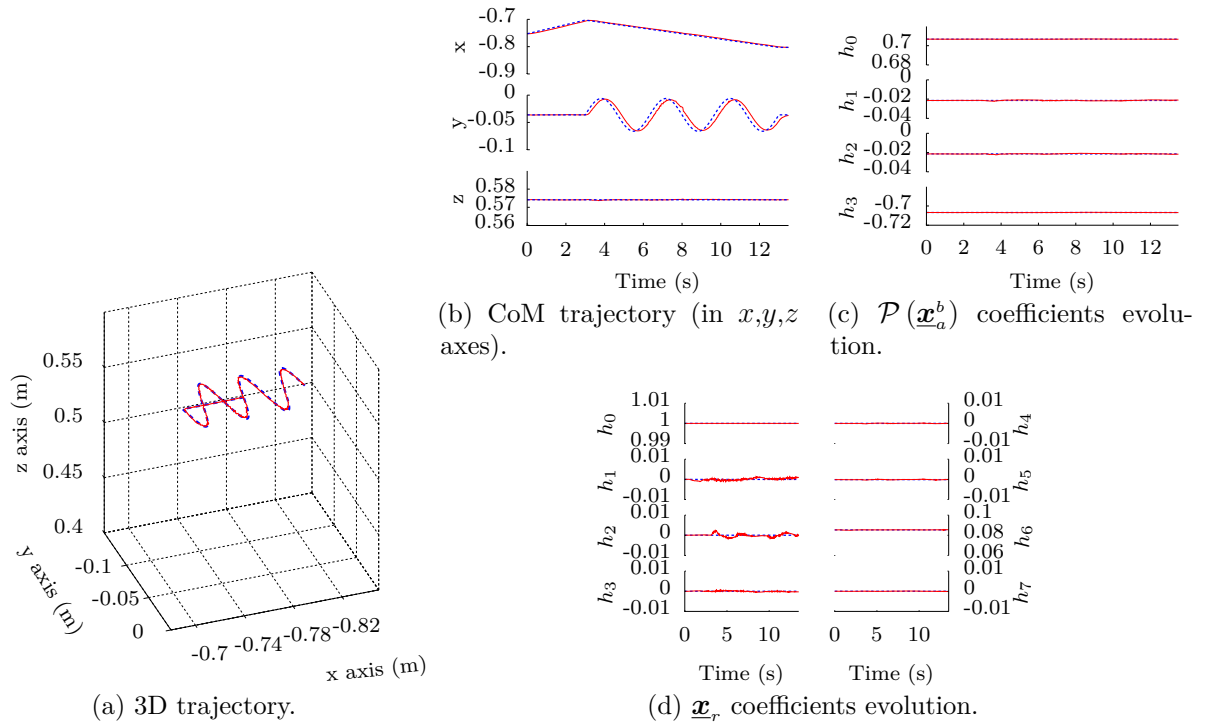


Figure B.1: Results obtained from the execution of a trajectory defined by a sine in the xy plane. The *solid red line* represents the measured values and the *dashed blue line* represents the reference.

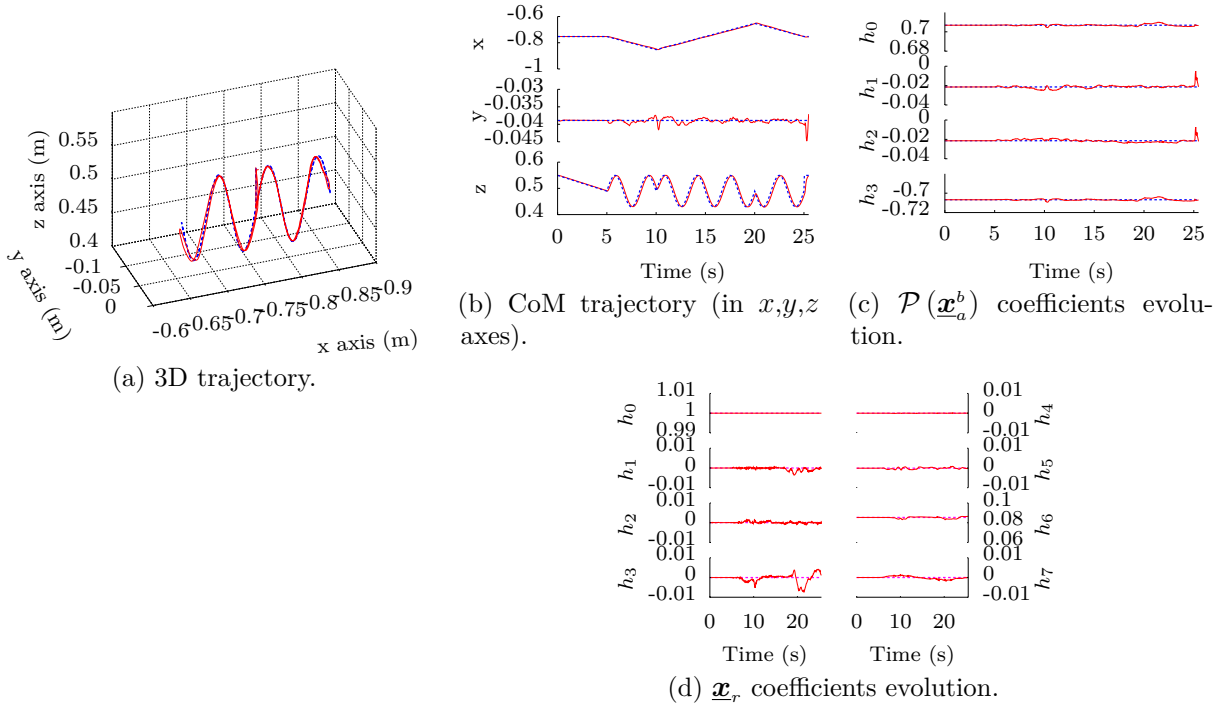
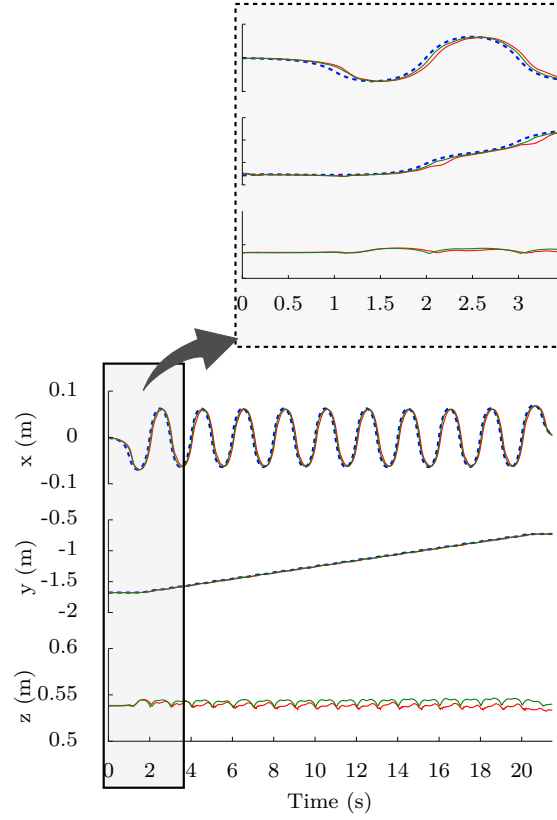
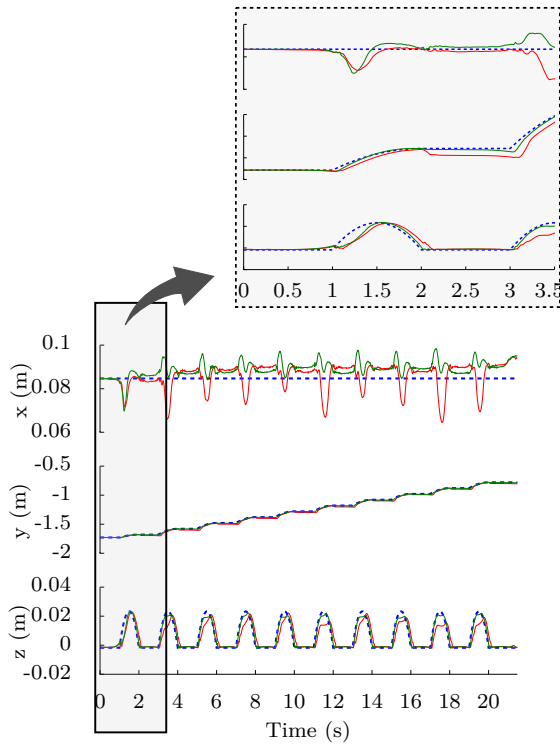


Figure B.2: Results obtained from the execution of a trajectory defined by a sine in the xz plane. The *solid red line* represents the measured values and the *dashed blue line* represents the reference.

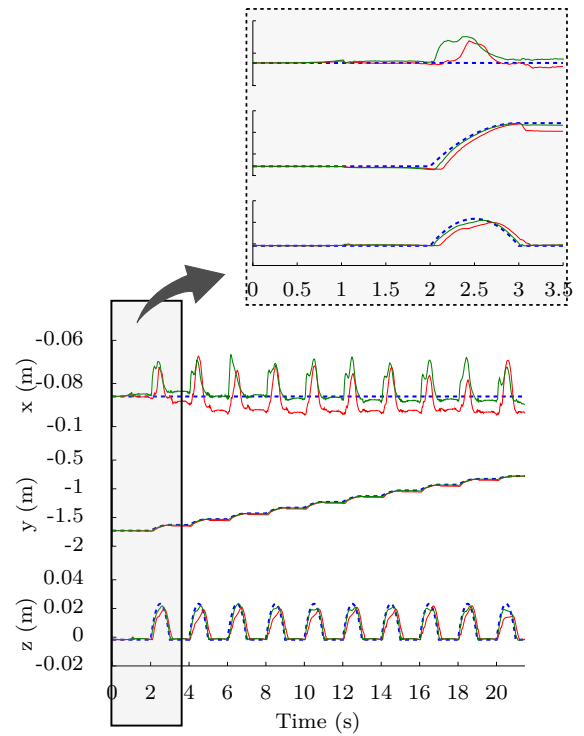
B.2 Walking Control

Recalling the walking patterns used in the experiments:

1. Step length = 5cm, $t_s = 0.5$ s;
2. Step length = 10cm, $t_s = 1$ s;
3. Step length = 20cm, $t_s = 1$ s;
4. Step length = 30cm, $t_s = 1.5$ s.

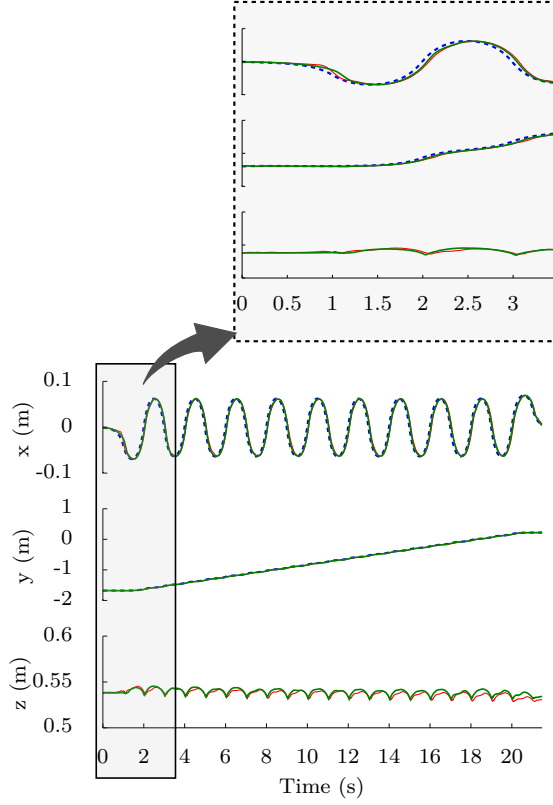
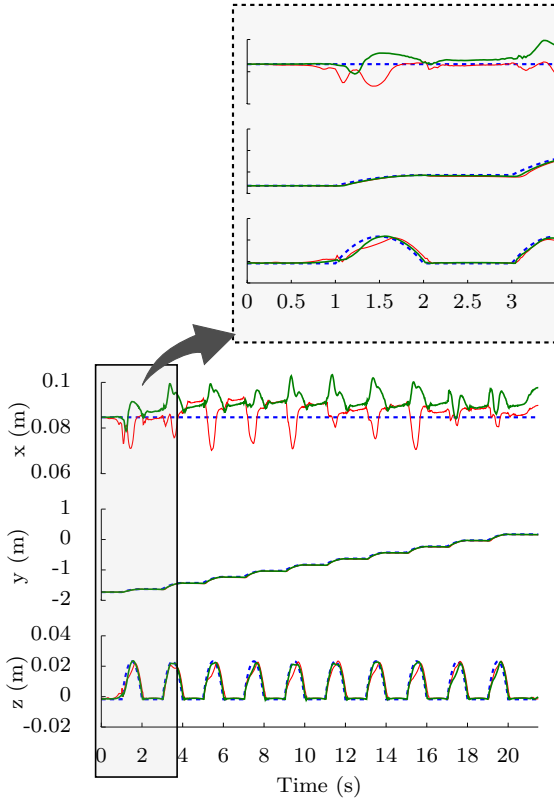
(a) CoM trajectory (in x, y, z axes).

(b) Right foot trajectory.

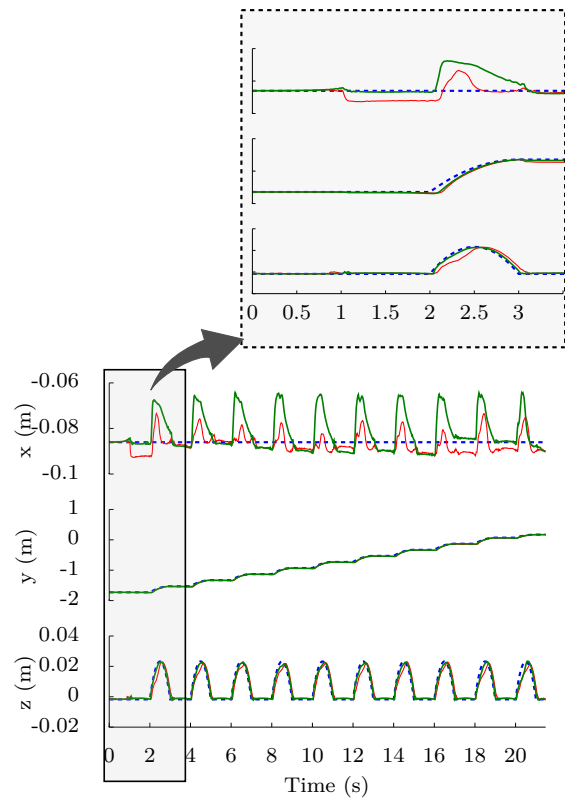


(c) Left foot trajectory.

Figure B.3: Comparison of the results obtained from the execution of the second pattern controlled using strategies 1 and 2. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 1, and the *solid green line* represents the measured values of the system controlled using the strategy 2.

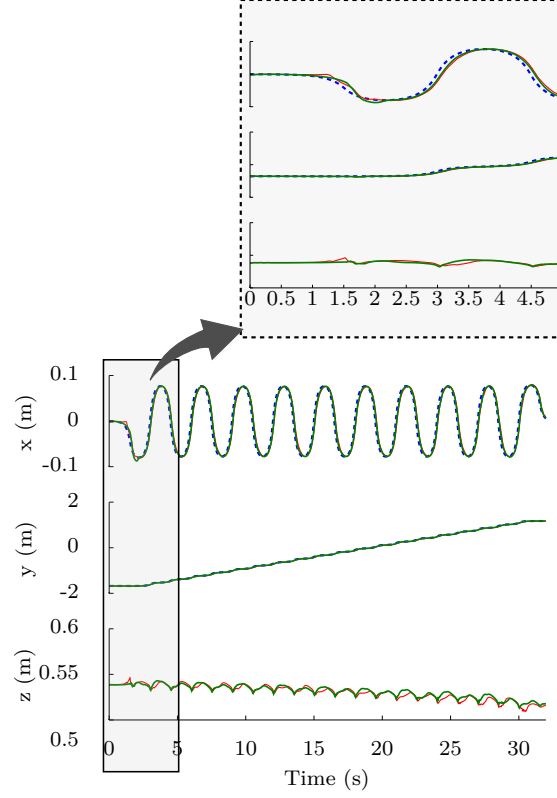
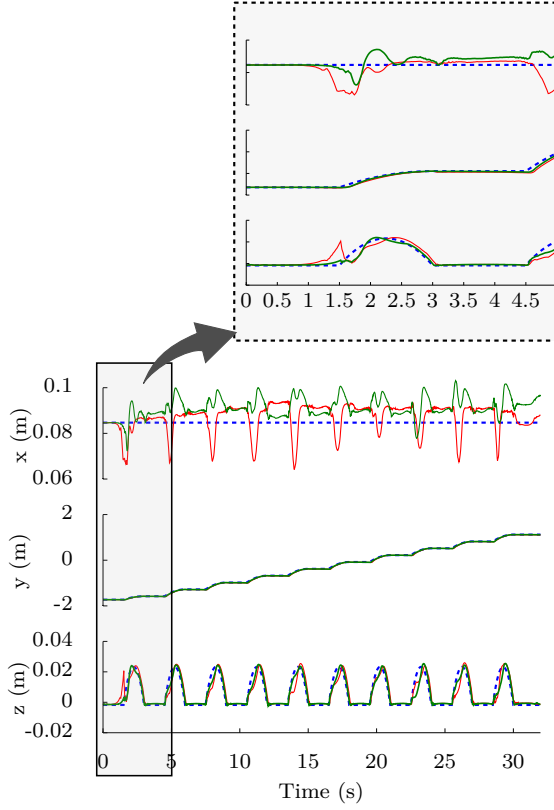
(a) CoM trajectory (in x, y, z axes).

(b) Right foot trajectory.

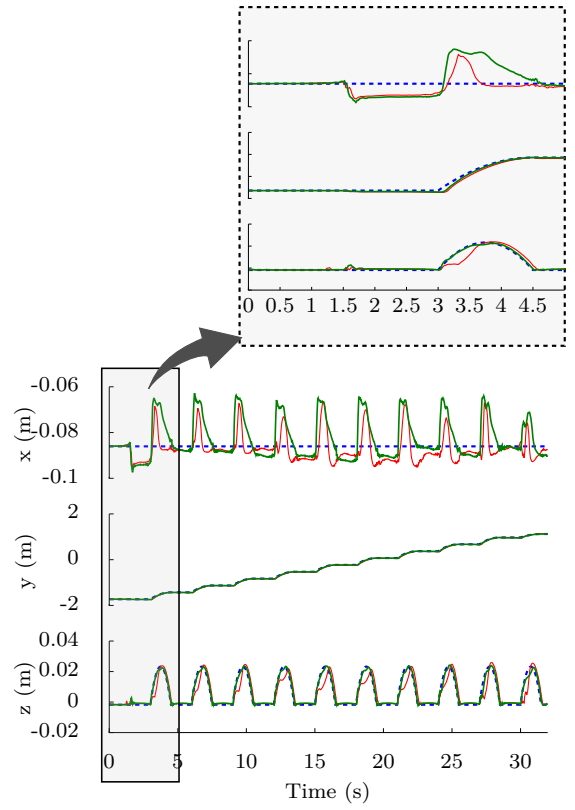


(c) Left foot trajectory.

Figure B.4: Comparison of the results obtained from the execution of the third pattern controlled using strategies 1 and 2. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 1, and the *solid green line* represents the measured values of the system controlled using the strategy 2.

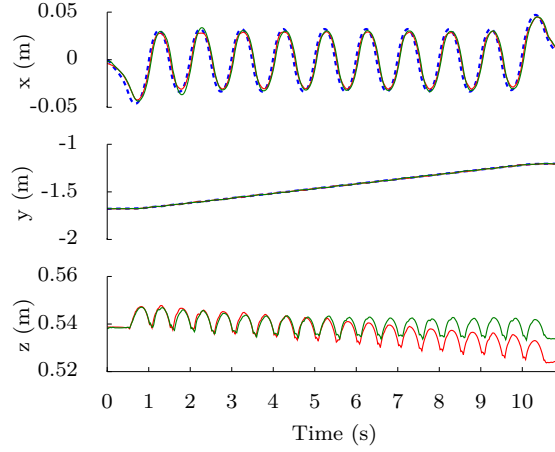
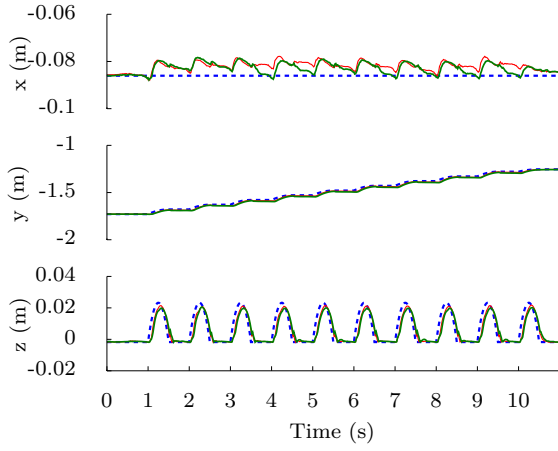
(a) CoM trajectory (in x, y, z axes).

(b) Right foot trajectory.

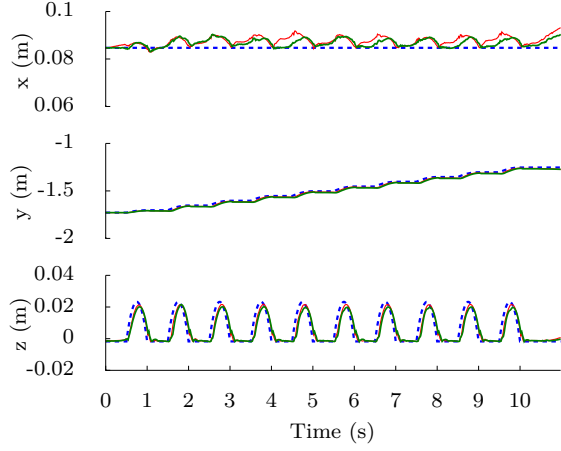


(c) Left foot trajectory.

Figure B.5: Comparison of the results obtained from the execution of the fourth pattern controlled using strategies 1 and 2. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 1, and the *solid green line* represents the measured values of the system controlled using the strategy 2.

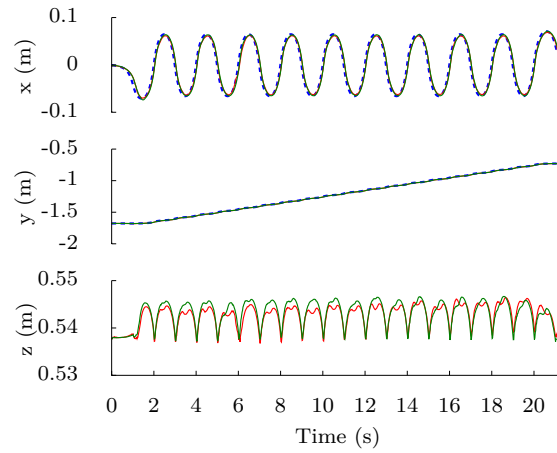
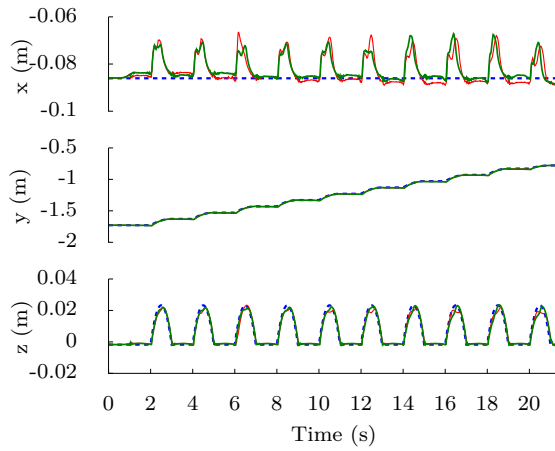
(a) CoM trajectory (in x, y, z axes).

(b) Left foot trajectory.

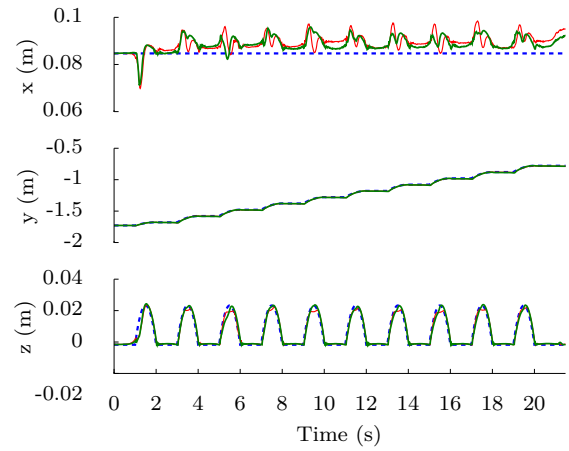


(c) Right foot trajectory.

Figure B.6: Comparison of the results obtained from the execution of the first pattern controlled using strategies 2 and 4. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 2, and the *solid green line* represents the measured values of the system controlled using the strategy 4.

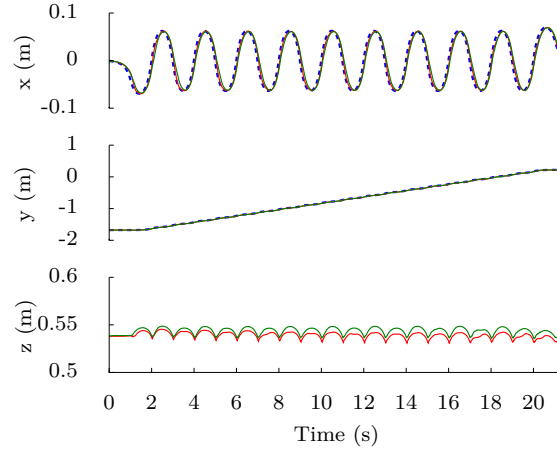
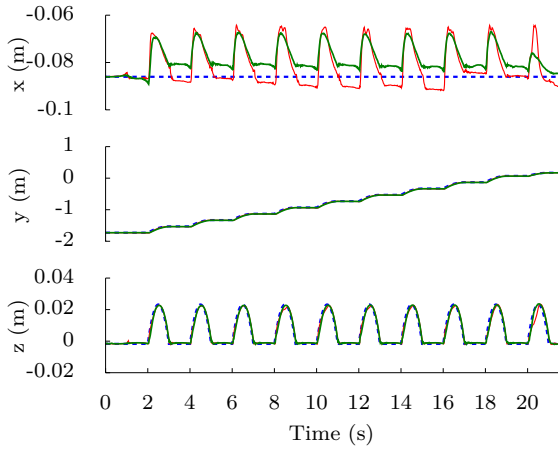
(a) CoM trajectory (in x, y, z axes).

(b) Left foot trajectory.

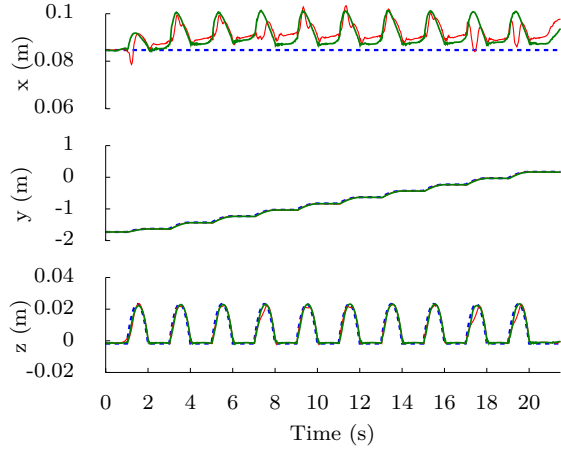


(c) Right foot trajectory.

Figure B.7: Comparison of the results obtained from the execution of the second pattern controlled using strategies 2 and 4. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 2, and the *solid green line* represents the measured values of the system controlled using the strategy 4.

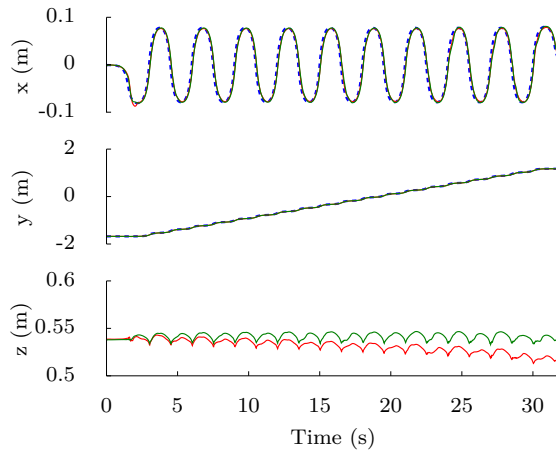
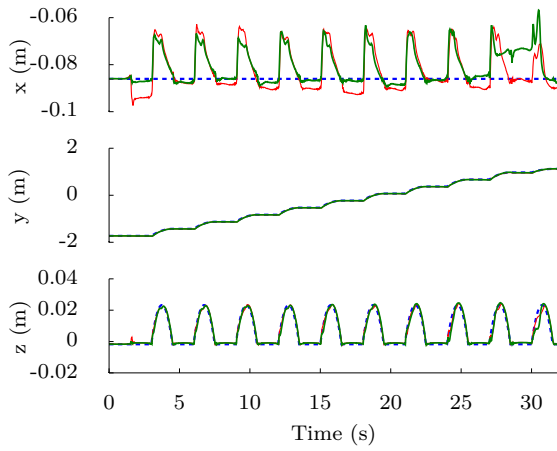
(a) CoM trajectory (in x, y, z axes).

(b) Left foot trajectory.

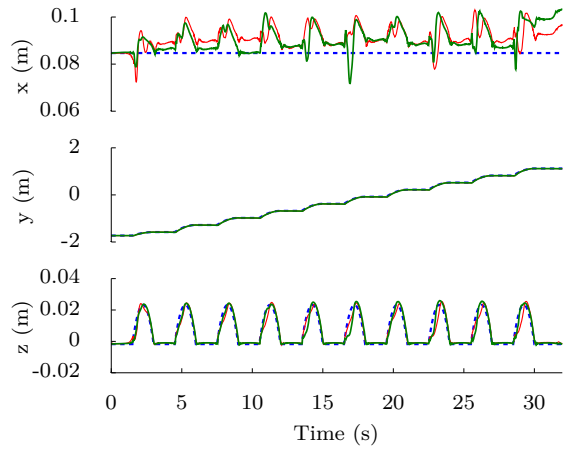


(c) Right foot trajectory.

Figure B.8: Comparison of the results obtained from the execution of the third pattern controlled using strategies 2 and 4. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 2, and the *solid green line* represents the measured values of the system controlled using the strategy 4.

(a) CoM trajectory (in x, y, z axes).

(b) Left foot trajectory.



(c) Right foot trajectory.

Figure B.9: Comparison of the results obtained from the execution of the fourth pattern controlled using strategies 2 and 4. The *dashed blue line* represents the reference, the *solid red line* represents the measured values of the system controlled using the strategy 2, and the *solid green line* represents the measured values of the system controlled using the strategy 4.