Technical University of Crete, Greece
School of Electronic and Computer Engineering

Predictive Control for Stable Dynamic Locomotion of Real Humanoid Robots

Stylianos Piperakis

Thesis Committee
Associate Professor Michail G. Lagoudakis (ECE)
Professor Athanasios P. Liavas (ECE)
Associate Professor Aggelos Bletsas (ECE)

Chania, July 2014

Πολυτεχνειο Κρητης

Σχολη Ηλεκτρονικων Μηχανικων και Μηχανικων Υπολογιστων

Προβλεπτικός Έλεγχος για Ευσταθές Δυναμικό Βάδισμα Πραγματικών Ανθρωποειδών Ρομπότ

Στυλιανός Πιπεράκης

Εξεταστική Επιτροπή

Αναπληρωτής Καθηγητής Μιχαήλ Γ. Λαγουδάκης (ΗΜΜΥ)

Καθηγητής Αθανάσιος Π. Λιάβας (ΗΜΜΥ)

Αναπληρωτής Καθηγητής Άγγελος Μπλέτσας (ΗΜΜΥ)

Χανιά, Ιούλιος 2014

Abstract

Robust stable omnidirectional locomotion for humanoid robots is a crucial problem and an active research area nowadays. In general, biped locomotion relies on distinct gait phases, during which it must be ensured that the sum of the forces acting on the robot do not result in a loss of balance. To generate stable walking patterns, the need of a stability measure is evident to ensure upright locomotion. State-of-the-art work on this problem uses the Zero Moment Point (ZMP) as a criterion to measure stability. The ZMP approach is a formal representation of the problem, which makes full use of sensor information commonly available on humanoid robots and allows for rigorous solutions to be constructed. This thesis presents a complete formulation of the challenging task of stable humanoid robot omnidirectional walk, based on the Cart and Table model for approximating the robot dynamics. For the control task, two novel approaches are proposed: (i) Preview Control augmented with the inverse system for negotiating strong disturbances and uneven terrain and (ii) Linear Model-Predictive Control (LMPC) approximated by an orthonormal basis for computational efficiency coupled with constraints for improved stability. For the generation of smooth feet trajectory, a new approach based on rigid body interpolation is proposed, enhanced by adaptive step correction. Finally, we present a sensor fusion approach for sensor-based state estimation and an effective solution to sensors' noise, delay, and bias issues, as well as to errors induced by the simplified dynamics and actuation imperfections. The proposed formulation is applied on a real Aldebaran Nao humanoid robot, where it achieves real-time onboard execution and yields smooth and stable gaits.

Περίληψη

Το σθεναρό και ευσταθές πολυκατευθυντικό βάδισμα για ανθρωποειδή ρομπότ είναι ένα κρίσιμο πρόβλημα και ένας ενεργός τομέας έρευνας στις μέρες μας. Σε γενικές γραμμές, το βάδισμα δίποδων ρομπότ βασίζεται σε διακριτές εναλλασσόμενες φάσεις, στη διάρκεια των οποίων θα πρέπει να εξασφαλισθεί ότι το άθροισμα των δυνάμεων που ενεργούν στο ρομπότ δεν οδηγούν σε απώλεια της ισορροπίας του. Για τη δημιουργία ευσταθών προτύπων βαδίσματος, είναι προφανής η ανάγκη ενός κριτηρίου ευστάθειας που εξασφαλίζει την όρθια στάση κατά το βάδισμα. Η σύγχρονη τεχνολογία σχετικά με το πρόβλημα αυτό χρησιμοποιεί το κριτήριο Zero Moment Point (ZMP) ως μέτρο ευστάθειας. Η προσέγγιση ZMR προσφέρει μια αξιωματική αναπαράσταση του προβλήματος, η οποία εκμεταλλεύεται πλήρως τις πληροφορίες των αισθητήρων που συμπεριλαμβάνονται συνήθως σε ανθρωποειδή ρομπότ και επιτρέπει την κατασκευή μαθηματικά αυστηρών λύσεων. Η παρούσα μεταπτυχιακή εργασία παρουσιάζει μια πλήρη διατύπωση του δύσκολου προβλήματος του ευσταθούς πολυκατευθυντικού βαδίσματος για ανθρωποειδή ρομπότ, βασισμένη στο μοντέλο Cart and Table για την προσέγγιση της δυναμικής του ρομπότ. Για το πρόβλημα του ελέγχου, προτείνονται δύο νέες προσεγγίσεις: (α) έλεγχος προεπισκόπησης (Preview Control) επαυξημένος με το αντίστροφο σύστημα για την διαπραγμάτευση ισχυρών διαταραχών και της παρουσίας ανωμάλου εδάφους και (β) γραμμικός προβλεπτικός έλεγχος μοντέλου (Linear Model-Predictive Control) που προσεγγίζεται από μια ορθοκανονική βάση για υπολογιστική αποδοτικότητα σε συνδυασμό με ανισοτικούς περιορισμούς για βελτιωμένη ευστάθεια. Για την παραγωγή ομαλών τροχιών για τα πέλματα, προτείνεται μια νέα προσέγγιση βασισμένη σε διαφορική γεωμετρία, εμπλουτισμένη με προσαρμοστική διόρθωση βήματος. Τέλος, παρουσιάζουμε μια προσέγγιση σύνθεσης μετρήσεων των αισθητήρων για την εκτίμηση της κατάστασης που δίνει αποτελεσματική λύση στα προβλήματα θορύβου, καθυστερήσεων, πόλωσης, καθώς και στα σφάλματα που προκαλούνται από την προσεγγιστική δυναμική και τις ατέλειες των επενεργητών. Η προτεινόμενη προσέγγιση εφαρμόζεται σε ένα πραγματικό ανθρωποειδές ρομπότ Aldebaran Nao, όπου επιτυγχάνει εκτέλεση σε πραγματικό χρόνο και αποδίδει ομαλό και ευσταθές βάδισμα.

Acknowledgements

First I would like to thank my supervisor, Michael G. Lagoudakis, for accepting me as his student and member of the Kouretes team, giving me the opportunity to make this work possible, for his insistence on the highest standards, and especially for his input into the paper we co-authored.

In addition I would like to thank Prof. A.P Liavas for the inspiration his course provided me with.

Throughout the course of my M.Sc, I benefited from many interesting discussions about programming techniques and algorithms from Emmanouil Orfanoudakis and Eleytherios Chatzilaris, which eventually allowed my theories to be implemented on a Nao humanoid robot. In addition I thank my office colleagues Nikolaos Kofinas, Charilaos Akasiadis and Angelos Aggelidakis for enduring me and the noise I caused by the time Nao started to walk. Furthermore my teammates in Kouretes Evangelos Michelioudakis, Nikolaos Kargas and Nikolaos Pavlakis for the great times we had when participating to Robocup SPL competitions.

I would like to express my deepest thanks to Pagonitsa for her love and support throughout the highs and the lows of this work. Also my regards to my close friends Mole, Mits van Mav, Jizzmaster, Aerial, Beleaver and Spiderman.

Finally I want to thank my family for supporting me in all possible ways and providing me with what I needed to succeed.

Contents

List of Figures

LIST OF FIGURES

List of Algorithms

Chapter 1

Introduction

There is a driving force more powerful than steam, electricity and atomic energy: the Will. Albert Einstein (1879–1955)

Like never before, nowadays technology can bring imagination to life. The question is: "what ideas will we conceive?" For decades, popular culture has been enthralled with the possibility of robots that act and look like humans. We are promised by movies, science fiction, and television that humanoids will cook for us, clean for us, become our best friends, teach our children, and even fall in love with us. So, where are they? The forerunners are already here. Recently, the media have covered a surprising number of new humanoid robots emerging on the commercial market. Like many new technologies, these early generations of commercially-available humanoids are costly curiosities, useful for entertainment, but little of anything else. Yet, in due time, they will accomplish a wide variety of tasks in homes, battlefields, nuclear plants, government installations, factory floors, and even space stations.

Humanoids will exhibit emotion, forge relationships, make decisions, and develop as they learn through interaction with the environment. Robots that can incrementally acquire new knowledge from autonomous interactions with the environment will accomplish tasks by means their designers did not explicitly implement and will adapt to unanticipated circumstances of unstructured environments. Already, humanoid robots can autonomously perform task decomposition necessary to carry out high-level, complex commands given through gesture and speech. Humanoids can adapt and orchestrate

existing capabilities as well as create new behaviors using a variety of machine learning techniques. In fact, some researchers claim to have implemented a first stab at the "seed", which will allow robot intelligence to develop indefinitely. As they adapt to their own unique experiences with the world, we will look out upon a population where no two humanoids are exactly alike.

Humanoids may prove to be the ideal robot design to interact with people. After all, humans tend to naturally interact with other human-like entities; the interface is hardwired in our brains. Their bodies will allow them to seamlessly blend into environments already designed for humans. Historically, we humans have adapted to the highly-constrained modality of monitor and keyboard to access most of today's technology. In the future, technology will adapt to us. Undoubtedly, humanoids will change the way we interact with machines and will impact how we interact with and understand each other.

Humanoid robotics also offers a unique research tool for understanding the human brain and body. Already, humanoids have provided revolutionary new ways for studying cognitive science. Using humanoids, researchers can embody their theories and take them to task at a variety of levels.

Thus, we can conclude that the goal is to make humanoid robots, which look like us, but in no way replace us. We should not forget that they are only machines built to aid us in our daily tasks, entertain us, work with us, and work for us. To accomplish such tasks in human environments, a humanoid robot must be equipped with a variety of skills and abilities. Such a fundamental skill is locomotion.

1.1 Humanoid Robot Locomotion

For humanoids to exploit the way in which we have structured our environment, they will need to have legs. They must be able to walk up stairs and steep inclines and walk over rough, uneven terrain. It is easily understood that a humanoid robot, unable to move and change its position in space, is useless. The problem is that walking is not simply a periodic mechanical movement of the legs, but a full-body, carefully-planned course of actions taking place in real-time.

Robust stable omnidirectional locomotion for humanoid robots is a crucial problem and an active research area nowadays. The best approaches for humanoid robot locomo-

tion look closely at the dynamics of the human body for insight, however such approaches are computationally expensive and cannot be implemented on many modern humanoid robots. In practice, concentrated-mass models are used to approximate the true complex dynamics in order to achieve on-board execution.

In general, biped locomotion relies on distinct gait phases, during which it must be ensured that the sum of the forces acting on the robot do not result in a loss of balance. During the Single-Support (SS) phase, the gait relies on balancing on one leg (support leg), while the other one (swing leg) moves towards a planned foot location. When this is accomplished, it enters the Double-Support (DS) phase, where the weight of the robot is shifted from the support to the swing leg, the legs switch roles, and the process repeats. The key question which naturally arises throughout these gait phases is the following: "How can we guarantee that humanoids will stably locomote like humans even in unknown environments?".

1.2 Thesis Contribution

This thesis presents a complete formulation for dynamic humanoid robot locomotion based on the Cart and Table model for approximating the complex non linear robot dynamics. For the walking pattern generation task, two novel approaches are proposed: (i) an approach based on the use of preview control augmented with the inverse system for encountering strong disturbances and the presence of uneven terrain and (ii) an approach based on using the Linear Model Predictive Control (LMPC) approximated by an orthonormal basis, which offers faster computation, since the dimensionality of the problem is significantly reduced. Furthermore, the second approach enables the use of constraints in locomotion for increased stability. An exterior point method is utilized for solving the constrained LMPC problem, which yields fast and accurate results in practice, satisfying our objectives for efficiency and computational performance.

While the walking patterns is the corner stone for solving the humanoid robot locomotion task, it is equally important to plan smooth feet trajectories that can achieve the desired gait, while abiding to the physical constraints of the robot. For the task of generating smooth feet trajectories, we present a new approach based on rigid body interpolation in the Special Euclidean (SE(3)) group yielding two kinds of trajectories: (i)

the geodesic curves and (ii) the minimum jerk curves, both obtained from the minimization of functionals based on left-invariant Riemannian metrics. Our approach improves upon common Bezier and spline interpolations and is further enhanced by adaptive step corrections to the generated trajectories. Thus, smooth feet movements can be achieved in practice, making the locomotion more human-like.

Moving from theory to practice, however, introduces many implementation issues. Sensor readings on a humanoid robot are typically corrupted by noise, delay, and bias, making the control task inviable. Proper actuation is also non-trivial, even if analytical inverse kinematic procedures are used; this is due to the imperfect nature of the robot's joints and also due to use of simplified (or approximate) dynamics. All these issues are addressed effectively in this thesis and smooth human-like locomotion is achieved, as demonstrated by our implementation on the Aldebaran Nao humanoid robot. The outcome of this work covers the needs of our robotic team Kouretes, competing in the Standard Platform League (SPL) of the RoboCup competition.

1.3 Thesis Outline

This thesis consists of nine chapters, including this introduction. In Chapter 2, an answer to the question "what is a humanoid robot?" is given and many worth-mentioning humanoid robot projects from around the world are highlighted. Chapter 3 is dedicated to the criteria determining locomotion stability. Three different criteria for stable walking are presented. The most popular and widely accepted is the Zero Moment Point (ZMP), which is extensively presented and discussed in this chapter. In addition, the Foot Rotation Indicator (FRI) point, as well as the Contact Wrench Cone (CWC), are also presented. These criteria generalize and extend the ZMP criterion to walking scenarios where strong disturbances or uneven terrain must be negotiated. In Chapter 4, the problem of humanoid robot gait generation is analyzed and discussed. In particular, humanoid robots are classified as active and passive walkers, whereby active walkers are further categorized as static and dynamic walkers and the generation of the corresponding walking gaits is described. We also emphasize the need for simplifying humanoid robot dynamics to design, analyze, and control online their walking gaits. For this purpose, two concentrated-mass models are presented: the 3-Dimensional Linear Inverted Pendulum model (3D-LIPM) and the Cart and Table model. In Chapter 5, the proposed control

estimation scheme for humanoid robot locomotion is presented. First, a brief description of the Linear Quadratic Regulator (LQR) is given, followed by the Luenberger Observer needed for the state estimation task. Furthermore, the Preview control is extensively presented and applied to both the 3D-LIPM and to the Cart and Table model. Next, the novel control schemes are presented: (i) the preview controller for the Cart and Table model, augmented with its inverse system, making use of the auxiliary ZMP and (ii) the Linear Model Predictive Control (LMPC) with the use of an orthonormal basis for approximation, obtained through unconstrained or constrained optimization. In Chapter 6, interpolation methods for generating feet reference trajectories are presented. First, the well-known Cubic Bezier and Cubic-B spline interpolation methods are briefly outlined. Then, a new interpolation method is introduced, based on the fact that the foot is a rigid body. After a brief introduction to Lie groups and Lie algebra, two kinds of feet trajectory generation are described. In Chapter 7, the implementation of all proposed methods on a real humanoid robot, namely the Nao humanoid robot of Aldebaran Robotics, is discussed. In Chapter 8, various experiments are conducted with the Nao robot and results are shown and discussed. Finally, Chapter 9 concludes the thesis and outlines possible future work.

1.4 Notation

In this thesis, the standard mathematical notation is used. The matrices are denoted with bold capital letters, i.e $\mathbf{A}$. Row vectors are denoted with bold letters in lower case followed by a transpose symbol, i.e $\mathbf{b}^\top$. Column vectors are denoted with bold letters in lower case, i.e $\mathbf{c}$.

Chapter 2

Humanoid Robots

Intelligence is the ability to adapt to change. Stephen Hawking (1942–)

2.1 What is a Humanoid Robot?

Humanoid robotics includes a rich diversity of projects, where perception, processing and action are embodied in a recognizably anthropomorphic form in order to emulate some subset of the physical, cognitive, and social dimensions of the human body and experience. Humanoid robotics is not an attempt to recreate humans. The goal is not, nor should it ever be, to make machines that can be mistaken or intentionally used for replacing real human beings. Rather, the goal is to create a new kind of tool, fundamentally different from anything we have yet seen, which is designed to work with and for humans. Humanoids will interact socially with people in typical, everyday environments. We already have robots that carry out tedious, repetitive labor for specialized environments and tasks. Instead, humanoids will be designed to act safely alongside humans, extending our capabilities in a wide variety of tasks and environments.

At present, humanoid robotics is not a clearly-defined field, but rather an underlying impulse driving collaborative efforts that crosscut many disciplines. Mechanical, electrical, and computer engineers, roboticists, computer scientists, artificial intelligence researchers, psychologists, physicists, biologists, cognitive scientists, neurobiolo-

Figure 2.1: The HRP-4C humanoid robot presented in an open demonstration

gists, philosophers, linguists and artists all contribute and lay claim to the diverse humanoid projects around the world (see Figure 2.1 for an example). Inevitably, some projects choose to emphasize the form and mechanical function of the humanoid body. Others may focus on the software to animate these bodies. There are projects that use humanoid robots to model the cognitive or physical aspects of humans. Other projects are more concerned with developing useful applications for commercial use in service or entertainment industries. At times, there are deep ideological and methodological differences. For example, some researchers are most interested in using the human form as a platform for machine learning and online adaptation, while others claim that machine intelligence is not necessary.

Defining a humanoid robot is a lot like defining what it means to be human. Most likely, you'll know one when you see it, and yet have trouble putting the characteristics on paper. The physical constitution of the body is clearly crucial. Not surprisingly, some have chosen to define a humanoid robot as any robot with two arms, two legs and a human-like head. Unfortunately, such a definition says nothing about the ability of this robot to receive information, process it, and respond. Moreover, many humanoid robotics projects spend the majority of their efforts on a portion of the body, such as the head, the legs, or the arms. Rather than distinguishing humanoids by their physical construction, we choose to identify several complementary research areas, which, thus

far, have stood out as distinct emphases. Eventually, a fully-fledged humanoid robot will incorporate work from each of the areas below.

Perception This area includes computer vision, as well as a great variety of other sensing modalities, including visual, auditory, taste, smell, sonar, infrared, haptic, tactile, and a range of motion sensors. It also includes implementation of unconscious physiological mechanisms, such as the vestibulo-ocular reflex, which allows humans to track visual areas of interest while moving. Lastly, this area includes the attentional, sensor fusion, and perceptual categorization mechanisms, which roboticists implement to filter stimulation and coordinate sensing. Estimation of the current state of the world, for example localization of self and other objects in space, is typically posed as a high-level perception problem.

Human-robot interaction This area includes the study of human factors related to the tasking and control of humanoid robots. How can we communicate efficiently, accurately, and conveniently with humanoids? Another concern is that many humanoids are, at least for now, large and heavy. How can we insure the safety of humans who interact with them? Much work in this area is focused on coding or training mechanisms that allow robots to pick up visual cues, such as gestures and facial expressions, that guide interaction. Lastly, this area considers the ways in which humanoids can be profitably and safely integrated into everyday life.

Learning and adaptive behavior For robots to be useful in everyday environments, they must be able to adapt existing capabilities to cope with environmental changes. Eventually, humanoids will learn new tasks on the fly by appropriately sequencing existing behaviors. A spectrum of machine learning techniques will be used, including supervised learning, where a human trainer interacts with the humanoid, and reinforcement learning, where a built-in critic is used to direct autonomous learning. Learning will not only allow robust, domain-independent behavior, but will also facilitate design and development by hiding the complexity of task decomposition from the user. Humanoids should be told what to do rather than how to do it.

Legged locomotion For humanoids to exploit the way in which we have structured our environment, they will need to have legs. They must be able to walk up stairs and steep inclines and walk over rough, uneven terrain. The problem is that walking is not simply a forward-backward periodic mechanical movement of the legs, but a full-body orchestrated course of action that must take place in real-time. The best approaches look closely at the dynamics of the human body for insight, however such approaches are computationally expensive and cannot be implemented on many modern humanoid robot. Therefore, researchers turned to approaches where the humanoid robot dynamics are simplified to concentrated-mass models, which can be controlled in real time achieving efficient performance in practice. To this end, predictive controllers have been employed for controlling the robot gait, yielding stable, smooth, and human-like locomotion.

Arm control and dexterous manipulation Researchers around the world are working on dexterous robot tasks, including catching balls, juggling, chopping fruits, performing tele-surgery, and many more. From a mechanical point of view, robot arms have come a long way, even in the last year or so. Once large and heavy with noisy, awkward hydraulics, some humanoids now have sleek, compliant limbs with high strength-to-weight ratios. While mechanical innovation will and should continue to make progress, the real hard problem is how to move from brittle, hard-coded dexterity toward adaptive control, where graceful degradation can be realized. Currently, a humanoid body typically functions as a whole and, consequently, small errors in even one joint can drastically degrade the performance of the entire body.

Research Issues Will humanoid research propel robotics to great heights, channeling ideas from diverse fields towards an ultimate goal? Or, will the quest to model ourselves prove to be agonizingly complex and none can figure it out? Are we the best models of intelligence?. Although cognitive neuroscience will continue to contribute much to our self-understanding, we by no means fully appreciate the internal processes that actually produce our intelligence. Most likely, I believe, we will never fully understand or recreate everything that it means to be human. As the limits of our self-understanding expand, humanoid robots may simply follow our continuously changing conception of what we "think" we are.

2.2 The Problem of Legged Locomotion

A humanoid robot is a complex dynamical system and, from a control point of a view, this introduces many difficulties, especially in the task of controlling its locomotion. The main control problems to be anticipated are fundamentally stability problems: the prevention of fall under any body posture, the capacity of the mechanism to absorb impact during foot landing, the adaptation of stepping on any surface, and the attitude for maintaining the reference biped body orientation. Adequate sensors and control algorithms should be employed to obtain stable walking motion from the reference patterns. The combination of inertial measurement units, such as accelerometers and gyroscopes, and force sensors are used to feed back external disturbances due to terrain irregularities, structural imperfections, and inertial and gravity effects by measuring the actual attitude and forces on each foot and hand while standing and walking, as well as while performing some (collaborative) task by lifting, holding, carrying, or supporting a load. Alternatively, structural compliances should be mounted on the body in order to mechanically reduce the impact effects (due to working precision, design and terrain imperfections), normally in the soles of the feet. Furthermore, suitable reference walking patterns are required in order to distribute the biped mass during a walking motion to maintain stability (by taking into account the dynamics effects) and reduce the impacts. These walking patterns are the starting point of the mechanical design and the control system of the biped robot, because they define the number of degrees of freedom and working angles of each joint, which are the inputs of the kinematics and dynamics models and the references of the control system. This area is the focus of this thesis.

Thus, the locomotion control problem of a biped robot could be summarized as follows:

1. **Maintaining Stability**
   Stability is maintained by controlling a non-physical degree of freedom, which is called the Zero Moment Point (ZMP). This criterion was proposed thirty years ago by Professor Miomir Vukobratovic [1]. The dynamics of the whole body are taken into account and the criterion affirms that the biped does not fall down, if the ZMP is maintained inside the convex hull during the walking motion. Therefore, it is necessary to have a control loop for the ZMP to obtain stable walking motion. To compute the ZMP online, force sensors are used; this information is fed back to the respective control loop.

2. **Absorbing the Impact and Adapting to Uneven Terrain**
   Imperfections or changes in the walking surface, as well as changing the support foot while walking, drastically cause force variations on the landing foot. These forces stress the biped structure. Thus, the compliance control loop should be implemented to adapt the biped to these changes. Furthermore, the compliance structural design should be developed in order to cushion the impacts. The force sensors are used to feed back the external reaction on each foot.

3. **Maintaining Upright Position**
   During the walking gait, the dynamics and gravitational effects cause tipping torques, which may cause the biped to fall down. Furthermore, structural imperfections cause significant flexing on some joints. As the reference patterns include the attitude that the biped should walk upright at all times, a control attitude loop must be implemented for the upper body to assist stable walking motion. Suitable gyroscopes and accelerometers are used to compute the actual robot attitude and it is fed back to the biped control system, which allows the biped to walk upright.

2.3 Humanoid Robot Projects

Humanoid research has already begun to accelerate. While only a few institutions are fully dedicated to the creation of humanoid robots, a large number of projects around the world have met with encouraging success in particular areas. This section highlights endeavors in legged locomotion, arm control and dexterous manipulation, robot-human interaction, service robots, learning and adaptive behavior, perception, and anthropopathic (emotive) robots. These categories certainly should and do overlap. Robust arm control is, of course, impossible without perception. Legged locomotion for rough terrain usually requires a panoply of control and estimation techniques. One of the most encouraging things for the world of humanoid robotics is the increased collaboration and the growing community behind the various projects and research areas. The brief descriptions below attempt to give some insight into how these projects promote the development of the field.

2.3.1 Early Endeavors

With the rise of the computer, people immediately began to envision the potential for encoding human intelligence into textual programs, but soon discovered that static programs and rule-based logic cannot capture the true essence of human intelligence. Early attempts to create artificial intelligence produced information-processing machines that operated on high-level human concepts, but had difficulty relating those concepts to actions and perceptions in the external world. Estranged from perception and action, such intelligence derived meaning only as an extension of the human creator or user.

Once embodied in real robots, such programs were confounded by noisy and all-too-often inconsistent data streaming in and out from a host of real-world sensors and actuators. Intricate path-planning routines allowed robots to optimally traverse their internal environments, but were rendered meaningless as soon as the robot, inevitably, became disoriented. This correspondence problem hindered robots' ability to generalize knowledge and adapt behavior, resulting in hard-coded functionality applicable only to highly structured, specialized tasks, such as factory automation. Most roboticists forsook the goal of human-like cognition entirely and focused on creating functional, high-utility agents.

Nonetheless, as roboticists continued, mostly from a mechanical point of view, to develop new robotic tools for a variety of purposes, they gained a new respect for the human body as a platform that remains unmatched for versatility and adaptability. Accepting what they believed to be one of the greatest engineering challenges of all time, a few intrepid mechanical and electrical engineers began to build the world's first humanoid robots. In 1973, the construction of a human-like robot was started at the Waseda University in Tokyo under the direction of the late Ichiro Kato. He and his group developed WABOT-1, shown in Figure 2.2, the first full-scale anthropomorphic robot in the world. It consisted of a limb-control system, a vision system and a conversation system. WABOT-1 was able to communicate with a person in Japanese and to measure distances and directions to the objects using external receptors, artificial ears and eyes, and an artificial mouth. The WABOT-1 walked with its lower limbs and was able to grip and transport objects with touch-sensitive hands. At the time, it was estimated that the WABOT-1 had the mental level of a one-and-half-year-old child. In 1985, Kato and his research

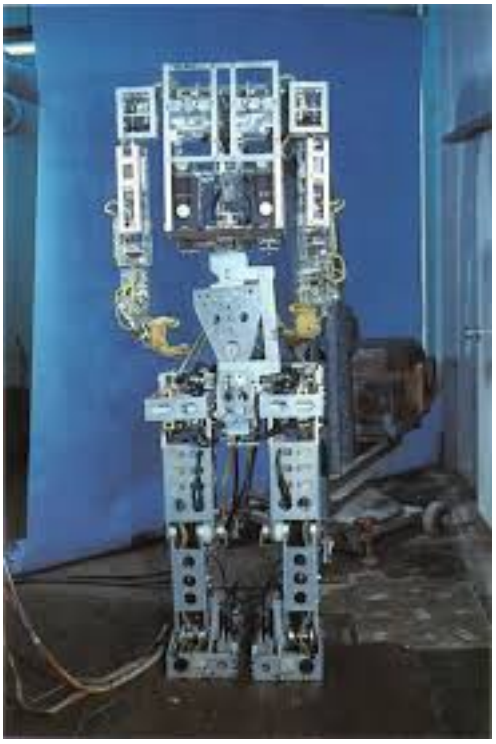

Figure 2.2: The WABOT-1 humanoid robot

group at Waseda University built WASUBOT, a humanoid musician (WAseda SUmitomo roBOT), developed with Sumitomo Electric Industry Ltd. WASUBOT could read a musical score and play a repertoire of 16 tunes on a keyboard instrument. Figure 2.3 shows WASUBOT playing the piano. Since these early successes, the Japanese electronics and automotive industries have played a key role in the emergence of humanoids by creating and developing robots capable of walking over uneven terrain, kicking a soccer ball, climbing stairs, and performing dexterous tasks, such as using a screwdriver and juggling. At the present time, we have full-scale humanoid robots that roughly emulate the physical dynamics and mechanical dexterity of the human body.

It remains to be seen to what extent we can breathe life into these creations. Ever since Karel Capek's play "Rossum's Universal Robots" captured the public's imagination in 1921, popular film, fiction and television have ingrained in our minds the possibility of intelligent, anthropomorphic robots that may eventually eliminate and replace mankind. Is it possible that we will eventually find ourselves surpassed or even displaced by our own creations? Or, will humanoid robots, despite our best efforts, remain little more

Figure 2.3: The WASUBOT humanoid robot

than smart appliances? As we examine many of the top humanoid projects in the world, this chapter presents the reader with a great diversity of projects that take us closer to an answer.

2.3.2 Honda Robots

In 1986, Honda Mo. Co. Inc. started to develop the ASIMO robot [9], which was eventually presented in 2000. ASIMO, whose evolution is shown in Figure 2.4, is currently one of the most advanced humanoid robots; it can walk like a human in any direction, climb stairs, and recognize faces, voice and gesture commands, etc. The version presented in 2005 could additionally run up to a speed of 6 Km/h and it was used as museum's guide and as a host in the Honda headquarters.

In December 1996, Honda publicly presented a humanoid robot with two legs and two arms. The research and development of this robot was initiated 24 years ago in 1984 [10]. The keywords behind this project were "intelligence" and "mobility" and the main direction and thoughts were "to coexist and collaborate with a human, to perform things that a human is unable to do and to create a mobility which brings additional value to human society". Put it another way, researchers aimed at developing a new type

Figure 2.4: The evolution of ASIMO humanoid robots

of robot to meet consumer needs, not simply a robot for a special limited purpose. They first planned a practical wheeled robot with two arms and a video camera installed on the upper body for recognition research, which they thought would be very convenient for studying intelligence as judgment and recognition research. However, as they gave careful thought to the meaning of a consumer type robot which they initially intended to develop, they came to the conclusion that it does not meet one of the keywords, "mobility". Then they looked into a type of consumer robot which will better meet their initial objective.

If they were to look at a "Domestic Robot" for an example of a type of robot that consumers may use, it would be necessary for that robot to walk around the furniture and walk up and down the staircase inside of a house. They found that a human with two legs is best suited for such movements. At the same time, if they were able to develop a two-legged (biped) robot technology, they believed that the robot should be able to move around the majority of ground environments, including rough terrain. Consequently, they reached a conclusion that the configuration of the lower part of the robot would be better, if it had a two-legged mobile mechanism, which can walk like human, than if it were the wheeled type and, therefore, decided to develop a robot by concentrating efforts

on that objective. Once they established their direction, the next step was how to realize it. They conducted a study on two-legged walking mechanisms by first analyzing actual human walking, in fact by taking themselves as models. The following 7 topics for the humanoid leg's mechanism were extensively studied.

1. Effectiveness of leg joints related to walking.
2. Locations of leg joints.
3. Movable extent of leg joints.
4. Dimension, weight, and center of gravity of a leg.
5. Torque placed on leg joints during walking.
6. Sensors related to walking.
7. Grounding impact on leg joints during walking.

The next step was to create a humanoid robot. They defined the functions of this humanoid robot as follows. The robot should be of such a type that it can automatically perform a certain type of work within a known environment and perform an uncertain type of work with assistance from a human operator within an unknown environment. The first experimental humanoid robot had an overall length of 1915 mm and a weight of 185 kg.

Honda researchers first concentrated their study on how to realize the coordinated movement of legs and arms and, therefore, the computers for image processing, action planning, electric power supply, etc., were not installed on the first robot. Through experimentation with this robot, they studied a coordinated movement of the robot to perform tasks, like turning a switch on and off, grasping and turning a door knob, and holding and carrying an object.

In the next stage, Honda researchers developed a wireless humanoid robot, as shown in Figure 2.5, which was eventually publicly revealed by Honda, as mentioned earlier. The overall length was 1820 mm with a weight of 210 kg. Computers, a motor driver, batteries as power source, and a transmitter were installed inside the robot. The main functional specifications were the ones listed below.

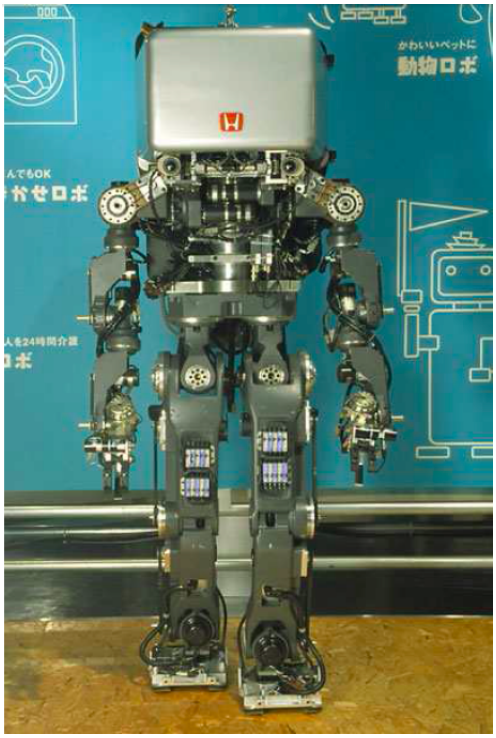

Figure 2.5: The P1 Honda humanoid robot

Mobility Performance

1. Be able to move around on normal flat surfaces, e.g. plastic tiles, paved road, grazing, etc.
2. Be able to pass through a narrow opening, whose width is 850 mm.
3. Be able to step over steps of height 200 cm and cross over steps and mounds 150 mm high and 150 mm long.
4. Be able to walk up and down the staircase of common buildings (steps 200 mm high and 220 mm deep) at a normal human speed.

Working Ability

1. Be able to grasp and hold an object that weights about 5 kg.
2. Be able to perform light work using a tool such, as a wrench, by remote control.

Figure 2.6: The latest ASIMO humanoid robot

The next development was divided into a short-term and long-term plan. For the short-term plan, emphasis was placed on hardware improvement, namely on the following objectives:

1. A smaller, more compact robot
2. Mobility performance improvement
3. Operability performance improvement

For the long-term plan, ASIMO researchers believe that increasing physical versatility by way of mobility improvement and environmental adaptability, which is made possible through hardware and software technology advancement, as well as improving autonomous mobility without detailed human instructions, are the important areas of focus. Furthermore, they also hope to develop technologies so that the humanoid robot can function not only as a machine, but blend in our social environment, interact with people, and play more important roles in our society. Under these efforts, the ASIMO humanoid robot was finally born in 2000. Five years later, the New ASIMO was presented (Figure 2.6)

Figure 2.7: The HRP-2P humanoid robot

with more abilities, such as running up to 6 km/h, performing human cooperation and interaction tasks, etc.

2.3.3 The HRP project

The University of Tokyo and Kawada Industries started with the H series of humanoid robots up to version H7. Next, the Advanced Institute of Science and Technologies (AIST) with Kawada Industries, initiated the Humanoid Robot Project (HRP), supported by the Japanese government, and developed the HRP-2L, HRP-2P, HRP-2, HRP-3P and HRP-3 humanoid robots (Figures 2.7 and 2.8). These humanoid robots have a wide range of joint motion and can lie down and stand up [11]. The HRP-3 is water-resistant and can walk on low-friction surfaces. The latest model, HRP-4C, shown in Figure 2.9, is closer to human nature, since it comes with a Japanese woman's face.

The HRP project was introduced as a question in the discussion on humanoid appli-

Figure 2.8: The HRP-2, HRP-3 and HRP-3P humanoid robots (from left to right)

cations: "What is the effectiveness of a human-shaped robot?". There are many people who claim that most robot applications do not not require a human shape and that the functions are more important than the shape [12]. However, when considering the practical meaning of humanoid research, one should recognize the importance of the human shape. There are at least three benefits a "human-shaped" robot brings:

1. A human shape is required for some tasks in certain domains and that creates new opportunities for robot applications in such areas.

2. A human shape produces emotional feelings useful for friendly communication between a robot and a human.

3. A human shape is one of the best shapes for a remotely-controlled robot, because it makes easier the control for the operator.

There are some businesses which require human-shaped devices and machines, such as entertainment business, prothesis and orthosis, commercial advertisements, and so on.

Figure 2.9: The HRP-4C humanoid robot

In those areas, human-shape robotic technologies have traditionally been pursued. Humanoid technology will have the possibility of being a powerful new tool to be activated in those areas.

The behavior of a humanoid seems to produce some kinds of feelings, which make people imagine it is friendly to humans. It is a special effect, which people typically do not experience, when they observe the behavior of a conventional industrial robot. Such a feature is thought to be very effective for constructing a service robot, especially one which works in a human-living environment. In telerobotics, one of the most important problems to be solved is to design a robot, which can be controlled easily by a human operator. The human shape is one of the best shapes for remotely-controlled robots. In the history of telerobotics research, the human-shaped robot has already been investigated and humanoid technology will likely provide an efficient tool for constructing an excellent telerobotic system.

As mentioned above, humanoid technology has several possibilities for practical applications. Therefore, the HRP project in Japan is promoted with the emphasis on the application aspect. The objective of the project is to develop a safe and reliable, human-friendly robot system capable of carrying out complicated tasks and supporting humans

within human living and working environments. The leader of the project is Prof. Hirochika Inoue and the sub-leader Prof. Susumo Tachi, both of whom are affiliated with the University of Tokyo. The New Energy and Industrial Technology Development Organization (NEDO) is responsible for the administration of the project along with the Ministry of Economy, Trade and Industry (METI), while the Manufacturing Science and Technology Center (MSTC) serves as the secretary of the project.

The project is run on a new scheme called a platform-based approach. The platform uses a humanoid prototype for common uses in the project. The platform is developed at phase one and it is utilized by contributors of the project as an infrastructure for research and development at phase two. The approach is an antithesis of the ordinary way of robotics projects, where elemental technologies are developed first and they are integrated in the final phase of the projects. The project was run from 1998 to 2002 for five years, consisting of phase one for the first two years and phase two for the last three years. The total budget of the project is expected to be about 5 billon JPY.

The platform consists of a humanoid robot, a remote control cockpit to control the robot and an equivalent virtual robot. Honda RD Co. Ltd. has produced the humanoid robot platform (the well known P3). Kawasaki Heavy Industries, Ltd., Matsushita Electric Works, Ltd., Fanuc, Ltd., and the University of Tokyo have developed the remote control cockpit system. By using the developed cockpit system, they can obtain realistic kinesthetic sensation of a humanoid robot's motion. Fujitsu, Hitachi, the University of Tokyo, Electrotechnical Laboratory (ETL), and Mechanical Engineering Laboratory (MEL), where ETL and MEL are currently a part of the Intelligent Systems Institute, AIST, have developed the virtual robot as the software counterpart of the platform robot. It has the equivalent dynamics and geometric model of the humanoid robot as well as that of its working environment.

In phase two (2000–2002), research and development were carried out on the applications of humanoid robots with consideration given to needs of industries in which such robots might be used. Improvement and addition of elemental technologies were carried out using the platforms developed in phase one. The applications included maintenance tasks of industrial plants, teledriving of construction machines, human care, security services and construction work. In selecting those applications, how the human shape is effective for each application was deeply considered.

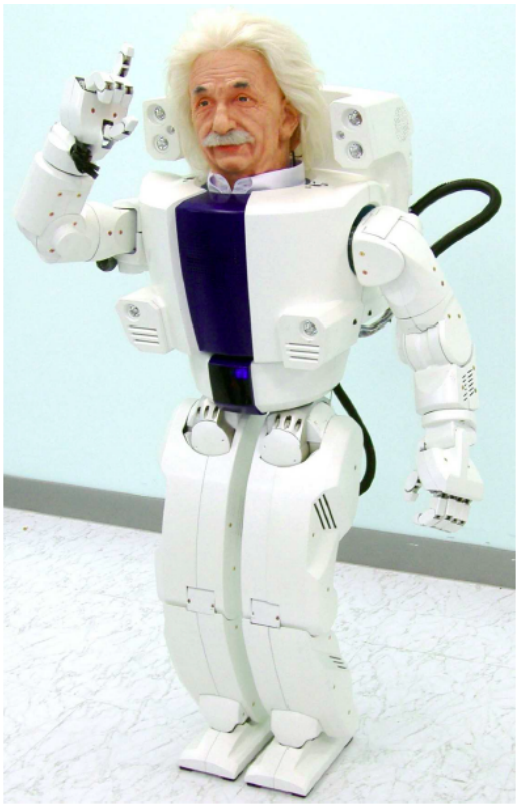

Figure 2.10: The Alber HUBO humanoid robot

2.3.4 The Einstein-HUBO humanoid project

From its preview versions, KHR-1 to KHR-3, Albert HUBO is an android robot. It is composed of a head, which is modeled after Dr. Albert Einstein's and HUBO's body. The development period took about 3 months, and it was finished in November, 2005. The head part was developed by Hanson Robotics. Its skin is a special material, Frubber, often used in the Hollywood [13].

The head has 35 joints, so it can impersonate various facial expressions using independent movements of eyes and lips. It has 2 CCD cameras to do vision recognition. Also, the body of Albert HUBO can perform all the HUBO's performances, so it is possible to express more natural features and movements. In the body, there are lithium polymer batteries which can get about two and a half hours of operating time. By using a remote network, it is possible to access the Albert HUBO from an external computer. Albert HUBO (Figure 2.10) was announced first at the 2005 APEC Summit in Busan, Korea.

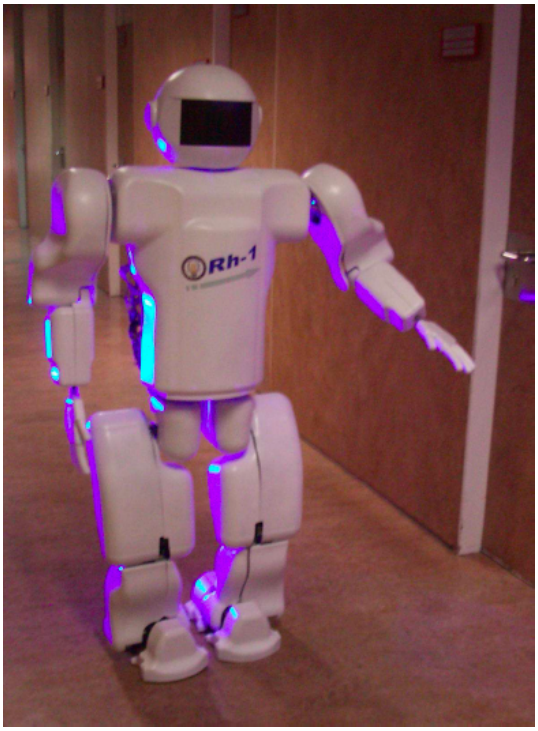

Figure 2.11: The Rh1 humanoid robot

It was praised by many world leaders, such as the US president, the Japanese Prime Minister, and so on.

2.3.5 The Rh-1 humanoid project

Inspired by this active area, the robotics lab of the University Carlos III of Madrid launched the humanoid robot project, the Rh-0 in its first phase (2002-2004), the Rh-1 (2005-2007) and finally the Rh-2 (2007-2009). The Rh-1 humanoid robot (Figure 2.11) is 1.45 mm tall, weighs 48.5 kg, has 21 DOFs, can walk at about 0.8 Km/h [14], recognizes faces, and responds to voice commands.

The Control level is divided into 3 layers represented as a controller centered on its own task, such as external communications, motion controller's network supervision, or general control. At on the Device level, each servo drive not only closes the servo loop, but also calculates and performs trajectories online, synchronizes with other devices, and can execute different motion programs located in its memory. These kinds of devices

are located near the motors, gaining the benefit of less wiring that is one of the requirements for energy efficiency. They are light-weight and require less effort in cabling. Advanced and commercially available motion controllers were utilized in order to reduce development time and cost. Continuous evolution and improvements in electronics and computing have already made it possible to reduce the industrial controller's size for using it in the humanoid development project. Furthermore, it has the advantage of applying well supported and widely used devices from the industrial control field, and brings the commonly used and well supported standards into a humanoid robot development area.

On the Control level, the Main controller is a commercial PC/104+ single board computer because of its small size and low energy consumption. It was used instead of a DSP controller, because it has a different peripheral interface, such as Ethernet and RS-232, and an easy programming environment. Also, there is a great variety of additional extension modules for the PC/104+ bus, like CAN-bus, digital and analog input-output, and PCMCIA cards. Selecting criteria were fast CPU speed, low consumption, and availability of expansion interfaces. The Main Controller provides general synchronization, updates sensory data, calculates the trajectory, and sends it to the servo controllers of each joint. It also supervises data transmission for extension boards, like the Supervisory Controller and the ZMP Estimation Controller via PC/104+ bus. The Communication Supervisory Controller uses a network bus to reliably connect distributed intelligent motion controllers with the Main Controller.

According to the Server-Client model, the humanoid robot is controlled by the passive Server, which waits for requests and upon their receipt, processes them and then serves replies for the Client. On the other hand, the Server controls all Control Agents which reside in the CAN bus network. In that case, the Control Server is no longer a slave. It is a network master for Control Agents which perform their operations (motion control or sensing) and reply for the Server.

2.3.6 The Robonaut project

Robonaut is a humanoid robot designed by the Robot Systems Technology Branch at NASA's Johnson Space Center in a collaborative effort with DARPA (Figure 2.12). The Robonaut project as presented by Bluethmann et.al [15], seeks to develop and demonstrate a robotic system that can function as an Extravehicular Activity (EVA) astronaut

Figure 2.12: The NASA Robotnaut humanoid robot

equivalent. Robonaut jumps generations ahead by eliminating the robotic scars (e.g. special robotic grapples and targets) and specialized robotic tools of traditional on-orbit robotics. However, it still keeps the human operator in the control loop through its telepresence control system. Robonaut is designed to be used for EVA tasks, i.e., those which were not specifically designed for robots.

The challenge is to build machines that can help humans work and explore in space. Working side by side with humans, or going where the risks are too great for people, machines like Robonaut will expand our ability for construction and discovery. Central to that effort is a capability they call dexterous manipulation, embodied by an ability to use ones hand to do work, and our challenge has been to build machines with dexterity that exceeds that of a suited astronaut. The resulting robotic system, called Robonaut, is the product of collaboration between NASA and DARPA, supporting the hard work of many Johnson Space Center (JSC) engineers that are determined to meet these goals.

Researchers are using a humanoid shape to meet NASA's increasing requirements for Extravehicular Activity (EVA, or spacewalks). Over the past five decades, space flight hardware has been designed for human servicing. Space walks are planned for most of the assembly missions for the International Space Station and they are a key contingency

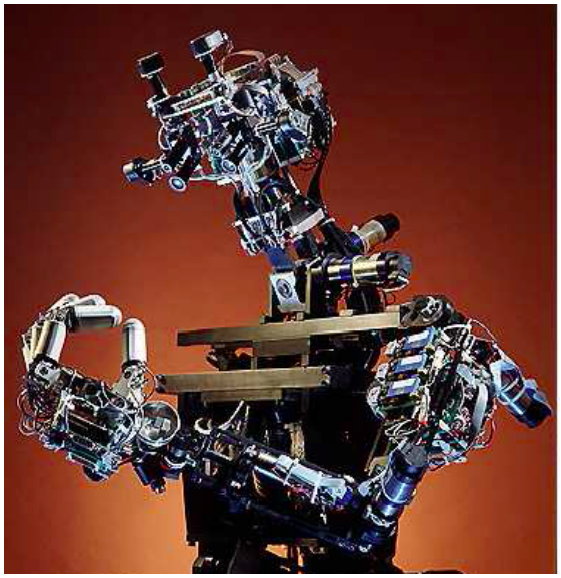

Figure 2.13: The COG humanoid robot

for resolving in-orbit failures. Combined with a substantial investment in EVA tools, this accumulation of equipment requiring a humanoid shape and an assumed level of human performance presents a unique opportunity for a humanoid system.

While the depth and breadth of human performance is beyond the current state of the art in robotics, NASA targeted the reduced dexterity and performance of a suited astronaut as Robonaut's design goals, specifically using the work envelope, ranges of motion, strength and endurance capabilities of space walking humans.

2.3.7 The COG project

To explore issues of developmental structure, physical embodiment, integration of multiple sensory and motor systems, and social interaction, an upper-torso humanoid robot, called COG (Figure 2.13), has been constructed by Brooks et al. [16]. The robot has twenty-one degrees of freedom and a variety of sensory systems, including visual, auditory, vestibular, kinesthetic, and tactile senses. It has a variety of implemented visual motor routines (smooth-pursuit tracking, binocular vergence, and vestibular-ocular and opto-kinetic reflexes), orientation behaviors, motor control techniques, and social behaviors (pointing to a visual target, recognizing joint attention through face and eye finding, imitation of head nods, and regulating interaction through expressive feedback). It also

further outlinse a number of areas for future research that will be necessary to build a complete embodied system.

As Prof. Brooks of MIT states, avoiding flighty anthropomorphism, you can consider COG to be a set of sensors and actuators, which try to approximate the sensory and motor dynamics of a human body. Except for legs and a flexible spine, the major degrees of motor freedom in the trunk, head, and arms are all there. Sight exists in the form of video cameras. Hearing and touch are on the drawing board. Proprioception in the form of joint position and torque is already in place; a vestibular system is on the way. Hands are being built and a system for vocalization is also in the works. COG is a unique hardware platform, which seeks to bring together each of the many subfields of Artificial Intelligence into one unified, coherent, functional whole.

2.3.8 Aldebaran Nao Humanoid Robot Project

The hardware platform that is currently used in the RoboCup Standard Platform League (SPL) is Nao, an integrated, programmable, medium-sized humanoid robot developed by Aldebaran Robotics in Paris, France. The robot's development began with the launch of Project Nao [17] in 2004. In August 2007, Nao officially replaced Sony's Aibo quadruped robot in the RoboCup SPL. In the past few years, Nao has evolved over several designs and several versions.

Nao (version 3.3), shown in Figure 2.14, is a 58cm, 5kg humanoid robot. The Nao robot carries a fully-capable computer on-board with an x86 AMD Geode processor at 500 MHz, 256 MB SDRAM, and 2 GB flash memory running an Embedded Linux distribution. It is powered by a 6-cell Lithium-Ion battery which provides about 30 minutes of continuous operation and communicates with remote computers via an IEEE 802.11g wireless or a wired ethernet link.

Nao RoboCup edition has 21 degrees of freedom; 2 in the head, 4 in each arm, 5 in each leg and 1 in the pelvis (there are two pelvis joints which are coupled together on one servo and cannot move independently). Nao, also, features a variety of sensors. Two cameras are mounted on the head in vertical alignment providing non-overlapping views of the lower and distant frontal areas, but only one is active each time and the view can be switched from one to the other almost instantaneously. Each camera is a 640×480 VGA devise operating at 30fps. Four sonars (two emitters and two receivers)

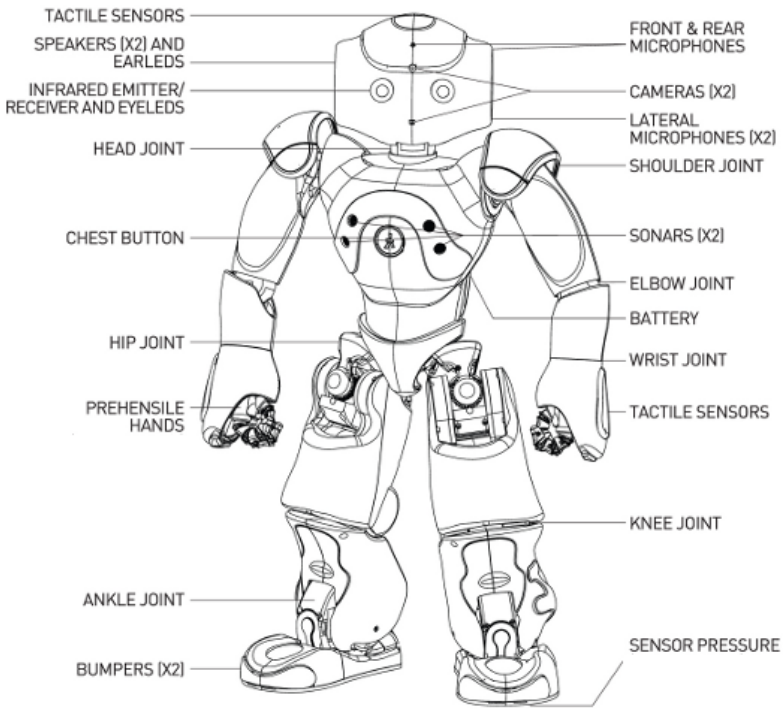

Figure 2.14: Aldebaran Nao v3.3 (Academic edition) components

on the chest allow Nao to sense obstacles in front of it. In addition, the Nao has a rich inertial unit, with one 2-axis gyroscope and one 3-axis accelerometer, in the torso that provides real-time information about its instantaneous body movements. Two bumpers located at the tip of each foot are simple ON/OFF switches and can provide information on collisions of the feet with obstacles. Finally, an array of force sensitive resistors on each foot delivers feedback of the forces applied to the feet, while encoders on all servos record the actual values of all joints at each time.

Aldebaran Robotics has equipped Nao with both embedded and desktop software [18] to be used as a base for further development (Figure 2.15). The embedded software, running on the motherboard located in the head of the robot, includes an embedded GNU/Linux distribution and NAOqi, the main proprietary software that runs on the robot and controls it. Nao's desktop software includes Choregraphe, a visual programming application which allows the creation and the simulation of animations and behaviors for the robot before the final upload to the real Nao, and Telepathe which provides

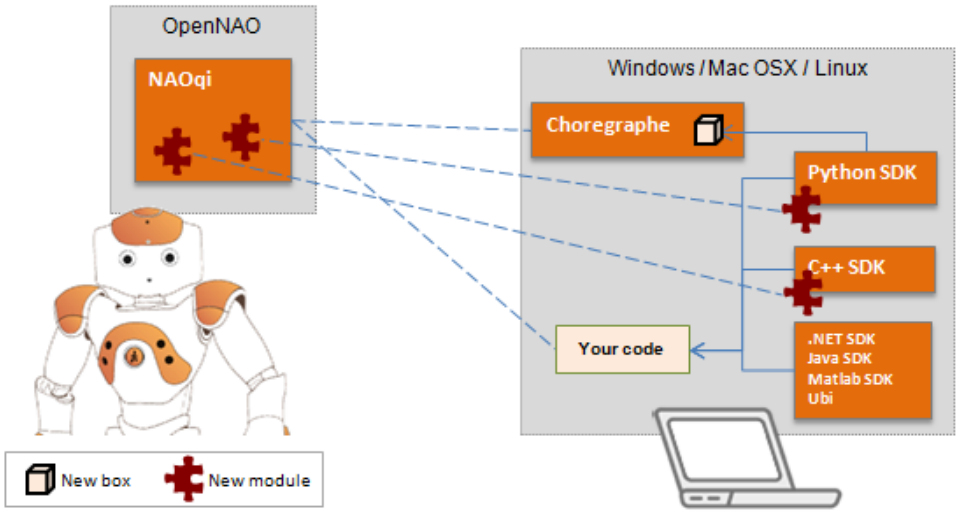

Figure 2.15: Embedded and desktop software for the Nao robot

elementary feedback about the robot's hardware and a simple interface to accessing its camera settings.

As far as the NAOqi framework is concerned, it is cross-platform, cross-language, and provides introspection which means that the framework knows which functions are available in the different modules and where. It provides parallelism, resources, synchronization, and events. NAOqi, also, allows homogeneous communication between different modules (motion, audio, video), homogeneous programming, and homogeneous information sharing. Software can be developed in C++, Python, and Urbi. The programmer can state which libraries have to be loaded when NAOqi starts via a preference file called `autoload.ini`. The available libraries contain one or more modules, which are typically classes within the library and each module consists of multiple methods (Figure 2.16).

2.3.9 DARwIn OP: Open Platform Humanoid Robot

DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence - Open Platform), shown in Figure 2.17, is an affordable, miniature humanoid-robot platform with advanced computational power, sophisticated sensors, high payload capacity, and dynamic motion ability to enable many exciting research, education, and outreach activities. Sponsored by the National Science Foundation (NSF) in the United States, DARwIn-OP has been

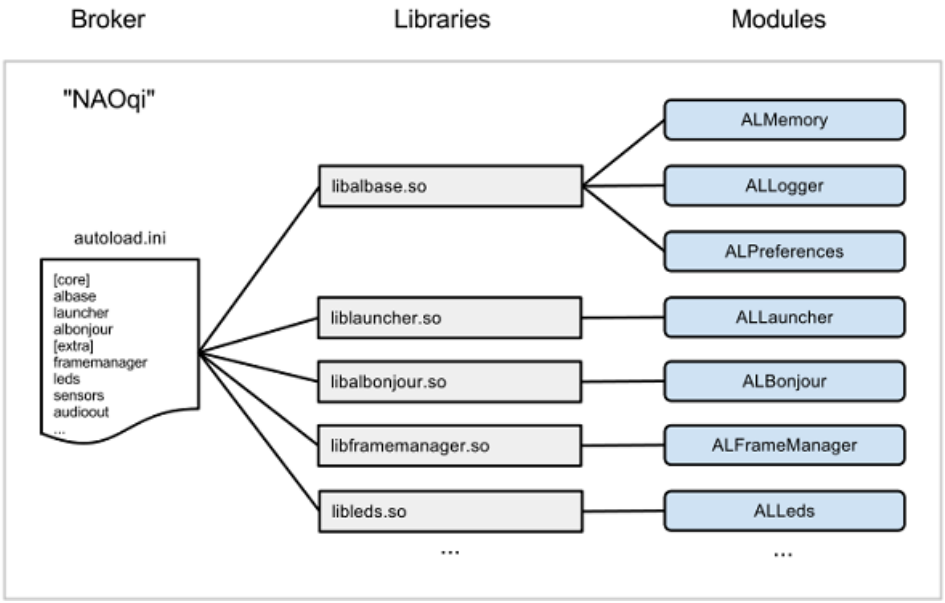

Figure 2.16: The NAOqi process

developed by RoMeLa at Virginia Tech in collaboration with the University of Pennsylvania, Purdue University and Robotis Co., based on the award-winning DARwIn series of humanoid robots in development since 2004. In July 2011, Team DARwIn competed in the Kid size League at RoboCup in Istanbul, Turkey winning the first place among 24 international teams.

DARwIn-OP is 454.5mm tall humanoid robot, weighs 2.9kg, and has 20 DOF (2-DOF head pan/tilt, 3-DOF arms, and 6-DOF legs), equipped with 20 MX-28 actuators with metallic gears and 3 Mbit/s high-speed Dynamixel buses for joint control. In addition, it has a built-in 1.6 GHz Intel Atom Z530 (32 bit) on-board and a 4 GB flash SSD. Also it has a 3-axis gyroscope, a 3-axis accelerometer, 3 buttons, and 2 detection microphones. DARwIn-OP is a true open platform where users are encouraged to modify it in both hardware and software, and various software implementations are possible (C++, Python, LabVIEW, MATLAB, etc.) The open source hardware is not only user-serviceable, thanks to its modular design, but can also be fabricated by the user. Publicly open CAD files for all of its parts and instructions manuals for fabrication and assembly are available on-line for free.

A number of DARwIn-OP units have been fabricated and built by Robotis Co. for

Figure 2.17: The DARwIn-OP humanoid robot

distribution to 11 partner universities (including major research universities, RUI institutions, a women's college, and two local high schools), which utilize them in their classroom teaching and projects, as well as outreach activities. The objective is to introduce DARwIn-OP to the humanoid robotics community, to broaden the DARwIn-OP project, and form a user community; train the users for use in research, education, and outreach activities; disseminate results of the usage of DARwIn-OP in the classroom; and to obtain feedback from the users for future improvements.

2.3.10 Entertainment Humanoid Robots

Sony had developed the SDR series of humanoid robots and finally in 2000 presented the "QRIO" (Figure 2.18), which stands for Quest for cuRIOsity. The robot comes with high mobility ranges and there were future commercial uses in mind, but the production was terminated a few years ago (2006). QRIO was promoted as an advanced entertainment humanoid robot and served as Sony's corporate ambassador, who embodies the company's vision of dreams, entertainment and technology around the world. A product of

Figure 2.18: The Sony QRIO humanoid robot

cutting-edge artificial intelligence and dynamic technology, it served as Sony's technology platform to promote advancements by combining Sony's expertise in these technologies. It is a small biped entertainment robot that aims to live with people in the future, makes life fun and makes people happy. QRIO is curious about many things in the world –for example, technology, microelectronics, and robotics to the rhythm of hip-hop. The everyday occurrences in people's lives are magical wonders to QRIO. The entertainment robot's features are autonomy, singing in a simple clear voice, dancing in a fluid motion, the ability to memorize and detect up to 10 individual faces, recognition of speech, as well as the ability to learn and memorize new words, short and long- term memory (can temporarily memorize objects and individuals, and through continued interaction, commit faces, names and emotions to its long-term memory), expressing an array of emotions through speech and body language, and walk on flat, irregular and tilted surfaces. This humanoid robot incorporates other Sony technology, such as steady-shot technology sensors to provide the robot a posture sensor acting like the inner-ear balance feature (from video camera technology), effective actuators to support motion, delicate and smooth movements from the manufacturing technology of precise machinery assembly (from CD

Figure 2.19: The Fujitsu HOAP-3 humanoid robot

player technology), and a jam detecting sensor which is a basic safety technology (from the past AIBO robot technology).

Another well-known humanoid robot for entertainment is Fujitsu Automation's HOAP-3 (Figure 2.19). Short in stature at 60cm and weighing in at 8.8kg, it is equipped with image recognition and various sensors. With a 1.1GHz Pentium M processor that runs on RT-Linux inside, HOAP-3 is a prime target for home-brew application developers. Fujitsu Automation has announced that the internal interface information of the hardware/software for the unit will be released. Simulation software for pre-confirmation of control programs is included for those who are into self-programming. A camera, microphone, speaker, expression LED, audio recognition function, voice synthesis function, and image recognition function have been added in contrast to HOAP-2. This makes it viable for research involving human communication and hand-eye coordination. HOAP-3's distance sensor, grip sensor, and acceleration sensor are controllable through a wired or wireless LAN.

2.4 RoboCup

The RoboCup competition is an international annual robotics competition inspired by Hiroaki Kitano in 1993 [19] and established in 1997. Its goal is to promote the research fields of artificial intelligence, multi-agent systems, and robotics. Participating teams focus on developing fully autonomous agents capable of operating into dynamic environments. The official goal of the project as stated by the RoboCup Federation is as follows:

> *"By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, complying with the official rule of the FIFA, against the winner of the most recent World Cup".*

While this ambitious objective remains open, RoboCup has since expanded into other relevant application domains based on the needs of modern society. Today, RoboCup covers the following themes.

RoboCup Soccer: The main focus of the RoboCup competitions is the game of football/soccer, where the research goals concern cooperative multi-robot and multi-agent systems in dynamic adversarial environments. All robots in this league are fully autonomous.

RoboCup Rescue: The intention of the RoboCup Rescue division is to promote research and development of highly mobile, dexterous, and fully- or semi- autonomous robots for search and rescue missions.

RoboCup @Home: This division aims at designing autonomous and naturally interactive assistant robots that can help people in their daily lives at home and in public.

RoboCup @Work: It is a new competition in RoboCup that targets the use of robots in work-related scenarios. It aims to foster research and development that enables use of innovative mobile robots equipped with advanced manipulators for current and future industrial applications.

RoboCup Junior: It is designed to introduce RoboCup to primary and secondary school children, as well as undergraduates who do not have the resources to get involved in the senior leagues of RoboCup. The focus of the Junior League lies on education.

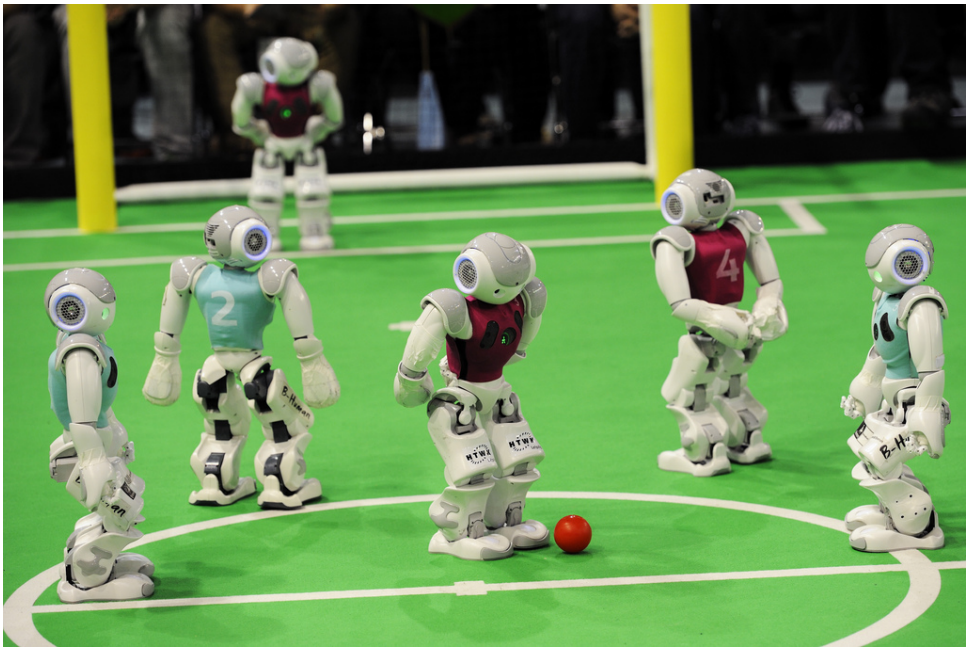

Figure 2.20: Standard Platform League at RoboCup 2013

2.4.1 Standard Platform League

RoboCup Soccer consists of five different leagues (Humanoid, Middle Size, Simulation, Small Size, and Standard Platform). In the Standard Platform League (SPL) all the teams use identical robots, the Aldebaran Nao humanoid robot. The teams are prohibited to make any changes to the hardware of the robot, neither can they interact with the robots during the games, therefore they concentrate on algorithm design and software development aiming at developing fully autonomous robots. The only interaction allowed is among the robots in the field through the wireless network and between the robots and the Game Controller, a computer that broadcasts information about the state of the game (score, time, penalties, etc.).

Currently, the SPL games are conducted on a field with dimensions $6m \times 9m$ (Figure 2.20). The field consists of a green carpet marked with white lines and two yellow goals. The appearance of the field is similar to a real soccer field, but it is scaled to the size of the robots. The ball is an orange street hockey ball. Each team consists of five robots, one goal keeper and four field players. Each robot wears a colored jersey shirt

as team marker; jerseys are blue for one team and red for the other. The total game time is 20 minutes and is broken in two halves; each half lasts 10 minutes. There are strict rules about player pushing, ball holding, leaving the field, etc. enforced by human referees; violation of theses rules results in player penalizations. The complete rules of the SPL games are stated in detail in the RoboCup Standard Platform League (Nao) Rule Book [20].

2.5 RoboCup SPL Team Kouretes

Team Kouretes is the RoboCup team of the Technical University of Crete and currently the only RoboCup SPL team founded in Greece. The team was founded in 2006 and participates in the main RoboCup competition ever since in various leagues (Four-Legged, Standard Platform, MSRS, Webots), as well as in various local RoboCup events (German Open, Mediterranean Open, Iran Open, RC4EW, RomeCup) and RoboCup exhibitions (Athens Digital Week, Micropolis, Schoolfest). Distinctions of the team include: 2nd place in MSRS at RoboCup 2007; 3rd place in SPL-Nao, 1st place in SPL-MSRS, among the top 8 teams in SPL-Webots at RoboCup 2008; 1st place in RomeCup 2009; 6th place in SPL-Webots at RoboCup 2009; 2nd place in SPL at RC4EW 2010; and 2nd place in SPL Open Challenge Competition at RoboCup 2011 (joint team Noxious-Kouretes). In the RoboCup 2012 competition, the team succeeded to proceed to the second round-robin round and rank among the top-16 SPL teams in the world. Recently, the team participated in AutCup 2012, in RoboCup Iran Open 2013, and in the RoboCup 2013 competition in Eindhoven (Figure 2.21). The members of the team are senior undergraduate and postgraduate ECE students of the Technical University of Crete working on their diploma thesis on RoboCup-related topics.

Kouretes started developing their own robotic software framework in 2008 and the code is constantly growing and gets maintained ever since. The team's publicly-available code repository[1] includes a custom software architecture, a custom communication framework, a graphical application for behavior specification, and modules for object recognition, state estimation, obstacle avoidance, behavior execution, and team coordination, which are briefly described below.

[1] https://github.com/kouretes/Monas

Figure 2.21: Team Kouretes at RoboCup 2013 in Eindhoven, The Netherlands

2.5.1 Monas Software Architecture

Monas [21] is a flexible software architecture which provides an abstraction layer from the hardware platform and allows the synthesis of complex robot software as XML-specified Monas modules, Provider modules, and/or Statechart modules. Monas modules, the so-called agents, focus on specific functionalities and each one of them is executed independently at any desired frequency completing a series of activities at each execution. The base activities, that an agent may consist of, are described briefly below:

- `Vision` [22] is a light-weight image processing method for humanoid robots, via which Kouretes team has accomplished visual object recognition. The vision module determines the exact camera position in the 3-dimensional space and subsequently the view horizon and the sampling grid, so that scanning is approximately uniformly projected over the ground (field). The identification of regions of interest on the pixels of the sampling grid follows next utilizing an auto-calibrated color recognition scheme. Finally, detailed analysis of the identified regions of interest seeks potential matches for corresponding target objects. These matches are eval-

uated and filtered by several heuristics, so that the best match (if any) in terms of color, shape, and size for a target object is finally extracted. Then, the corresponding objects are returned as perceived, along with an estimate of their current distance and bearing. Currently the detectable objects are the ball and the field goals.

- `LocalWorldState` is the end product of this thesis; it is responsible for estimating and providing the local belief of each robot and operates on the outcomes of the `Vision` activity.

- `SharedWorldModel` [23] is the activity which combines the local beliefs of all robots to create a common and shared estimation of the current state of the world (robot poses within the field, location of the ball, etc.) consistent with these local beliefs. The communication between the robots is accomplished through our communication framework, Narukom.

- `PathPlanning` [24] is the activity which accomplishes path planning with obstacle avoidance by first building a local, polar, obstacle occupancy map, which is updated constantly with real-time sonar information, taking into consideration the robot's locomotion. Afterwards, an A* search algorithm is used for path planning, the outcome of which suggests an obstacle-free path for guiding the robot to a desired destination. The way-points of the planned path are finally translated into walk commands to guide the robot along the path.

- `Behavior` is the activity which implements the desired robotic behavior. It operates on the outcomes of the `Vision`, `LocalWorldState`, `ObstacleAvoidance` and `SharedWorldModel` activities and decides which one is the most appropriate action to be executed next (walk, kick, etc.). Locomotion actions are passed to the `PathPlanning` activity for obstacle-free navigation, while motion actions are sent to the `MotionController` activity for execution. Behavior also includes a separate module which is responsible for determining a team strategy and assigning a role to each robot in the team [25]. This coordination mechanism dynamically selects the most appropriate roles depending on the robots' belief about the global world state, provided by the `SharedWorldModel` activity.

- `HeadController` manages the movements of the robot head (camera). It receives the desired commands from the `Behavior` activity.
- `MotionController` is used for managing and executing robot locomotion commands and special actions.
- `RobotController` handles external signals on the game state.
- `LedHandler` controls the robot LEDs (eyes, ears, chest button, feet).

Provider modules accomplish the complete decoupling of the robotic hardware by collecting and filtering measurements from the robot sensors and cameras and forming them as messages in order to be utilized as input data by any interested Monas agents. Each provider module can be executed independently and at any desired frequency.

Custom Forward and Inverse Kinematics [26, 27], designed specifically for the Nao humanoid robot, have been implemented as an independent software library optimized for speed and efficiency. The library is currently being used in other team projects, such as omni-directional walk engine and dynamic kick engine.

Statechart modules, which offer an alternative intuitive graphical specification of robot behavior, have also been integrated into Monas [28]. Kouretes Statechart Engine (KSE) [29, 30] is our own graphical tool for designing and editing statecharts for robot behavior. Statecharts are automatically transformed into code and are executed on-board using a generic multi-threaded statechart engine, which provides the required concurrency and meets the real-time requirements of the activities on each robot.

Robot communication is accomplished through Narukom [31], the communication framework developed for the needs of the team's code and it is based on the publish/-subscribe messaging pattern. Narukom supports multiple ways of communication, including local communication among the Monas modules, the Providers modules, and the Statechart modules that constitute the robot software, and remote communication via multicast connection among multiple robot nodes and among robot and external computer nodes. The information that needs to be communicated between nodes is formed as messages which are tagged with appropriate topics and host IDs.

KMonitor [32] is a debugging tool created specifically for the Monas architecture that takes advantage of the modularity of Kouretes code and helps in finding errors or verifying

that newly implemented features work correctly. It also allows for the easy creation of colortables, the transmission of remote commands over the network, etc.

2.6 What does the Future Hold?

While the humanoid projects presented in this chapter, along with many annual robotic competitions and research conferences held worldwide, are important steps in the right direction, functional results come slowly. Like the human infants, contemporary humanoids are inefficient at most tasks and require intensive training. One of the implications of this research is that to create human-like adaptability and versatility, it may be necessary to introduce an element of human inability and inconsistency. We have already manufactured machines with the ability to perform a task in exactly the same way, time after time. While such behavior can be useful for some tasks, it is brittle and will fail as soon as the wind changes. The future will bring humanoids designed to take part in the drama of chaos, inconsistency, and error we know fondly as the real world. Such humanoids will not be hindered by complexity and complication, but will embrace it and thrive on it.

Many of the attributes we consider uniquely-human come not from our strength, reliability, or precision with which we execute tasks. In fact, we often do quite poorly in these areas. This is not a coincidence, but a trace of our adaptability and intelligence. Optimality brings stasis and hinders versatility. It is a concept that has little to do with the flux of change in our real world. Unlike classroom computer science, the algorithms of human intelligence are neither provable nor constant. When we move from theory and virtual worlds into the real-world arena of noisy sensors, humidity and slippery floors, even fundamental tasks, such as locomotion, become agonizingly hard. Such stochasticities in the environment makes the controlling task non-trivial.

Therefore, one could say that the most "human" feature is our amazing ability to be fault-tolerant and cope in inconsistent environments. This ability derives, at least in part, from our ability to recognize imperfection and even exploit it, using the arbitrary fluctuations in ourselves and our environment to drive learning, creativity, humor and inventiveness. Humanoids will not inherit these human attributes, but rather reinforce them, allowing us to eventually answer what it means to be human.

Humans have always been eager to project emotion into machines. Our imaginations seem to have little trouble rising to the task. Recently, Hasbro Inc. has enlisted the help

of top roboticists at iRobot to develop robot infants that can be sold as toys. Complete with numerous servo motors and sensors, these inexpensive robots can simulate body functions, such as feeding and digestion, interact with humans, and even simulate real infant language learning, progressing from proto-language babble to the use of real words. The dolls are able to move their limbs and facial degrees of freedom in realistic ways to communicate a wide spectrum of desires and emotions in response to human attention. Fortunately for the designers, modeling realistic infant behavior is a decidedly easier task than adult capabilities and already includes an element of awkward imprecision. While many attempts to create adult humanoids have floundered, infant behavior may not be such a long shot. It seems that the first kind of humanoids to enter our homes will be animatronic infants.

As the number of humanoids increases, the collective population of humanoids will learn, develop, and perhaps eventually reproduce themselves more effectively. Unlike cars or televisions that improve along a linear, highly-controlled course, humanoids will be the ultimate point in self-accelerating technology. Moore's law holds at least in part, because we can use new chips to design and engineer their successors. Likewise, robotics is a self-enabling technology. Robotic tools will make the humanoids, we ourselves could never make. Once we have a large population of self-motivated agents attending to separate tasks, these agents will negotiate, exchanging tasks and resources in mutually beneficial ways. Humanoids will comprise a new distributed infrastructure, comprised not only of information, but also real-world action.

The ability for a machine to design and manufacture another machine is real. In regards to the notion of artificial life, we are faced with the prospect of something that seems to echo the phenomenon of biological reproduction. At the present time, a fabrication machine is available, where a simple humanoid robot can be manufactured. Even if humanoids do not reproduce themselves, imagine the utility of a humanoid that can evolve and fabricate its own tools on the fly to be perfectly suited for the task at hand. On the other hand, there are currently severe limitations. A fabrication machine that can handle larger, more complex robots may be, for a time, prohibitively expensive. Also, more complex behavior will require more accurate real-world modeling and richer simulation. The increased demand in computational power and time may limit the complexity of robots that can be evolved.

Although the application of Moore's law to silicon microchips will taper off some time around 2020, there are possibilities on the horizon for optical computers that switch photons instead of electrons; DNA computers that use sequences of bases to encode information; molecular computers that use molecules as logic gates; and, quantum computers that encode information by manipulating the rotation of individual atoms. Some who champion the potential of nanobot technology are convinced that within a century we will use molecule-sized robots to infiltrate and understand every pertinent function of the brain well enough to replicate each in an artificial medium. It is quite possible that such predictions overestimate our skills, while at the same time underestimate the organ that produces it. On the other hand, the computational power of computers is increasing exponentially, while the human brain is fixed at around 100 trillion connections. The somewhat shocking realization is that, if someday AI can reach a level comparable to human intelligence, there is no reason why it will not continue to sail past it.

While we cannot ignore this theoretical possibility, such predictions fail to realize that humans and humanoids may evolve along separate trajectories, such that it may never make sense to equate human and humanoid intelligence. Most likely, humans and humanoids will continue to be good at different things. Digital computers can already operate more than 10 million times faster than the electrochemical processes in our brains. The structure of the brain, however, more than makes up for this ostensible shortcoming. The power of the human brain arises not from the sheer number of bytes it can store or from the speed with which its electrochemical processes operate. Unlike the linear, digital computer, the brain is a massively-distributed parallel processor, where information is stored implicitly as associations between neurons. Thus, the fact that computers are moving quickly toward faster and larger computation does not necessarily mean they are moving closer toward human intelligence. The brain's capacity to not only learn, but also be aware of and able to direct its own learning, may result from recursive mappings between local collectives of neurons. Recent research indicates that this phenomenon of re-entry provides the brain with optimal complexity, a perfect balance between the tractability of a highly integrated, well-ordered structure and the infinite potential of desultory variation.

As computing does become faster and more pervasive, it remains to be seen whether humanoids can become crucial arbiters of the emerging New World. Some have argued that distributed computing will sweep away the need for humanoid robots. According to

this reasoning, computing will not need to be centralized in a single sophisticated agent, but will rather reside throughout the environment in every object around us. No doubt the role of humanoids will evolve alongside the changing lives of the humans they serve. Most likely, these changes will accentuate, rather than remove the need for intelligent agents that can mediate between humans and the increasingly complex, technological world we are creating.

2.7 Ethical Considerations

The world's population continues to steadily grow. Someone could ask why we would want to make a machine that looks, thinks, and emotes like a human, when we have plenty of humans already. Especially we should consider that those machine will gain expertise in certain domains, making them suitable for particular jobs. This matter is of great importance, since, in the world we live in, many of us do not have jobs or good places to stay, struggling to pull through and live well. Therefore, it is vital to re-emphasize that humanoids cannot and will not ever replace humans. Computers and humans are good at fundamentally different things. One should consider that calculators did not replace mathematicians. They did change though drastically the way mathematics was taught. For example, the ability to mentally multiply large numbers, although impressive, is no longer a highly-valued human capability. Calculators have not stolen from us part of what it means to be human, but have rather freed our minds, so we can spent our effort in more worthy tasks. By the time robots arrived, a change in the the society, industry and economy was noticed and humanoids are expected to affect this change even more.

The question arises "why should we have intelligent, emotion-exhibiting humanoids?". Emotion is often considered a debilitating, irrational characteristic. Why not keep humanoids, like calculators, merely as useful gadgetry easing our daily schedule? One could say that, if we do want humanoids to be truly reliable and useful, they must be able to adapt, learn, and develop. This is probably correct, since it is impossible to hard-code high-utility, general-purpose behavior, and, therefore, humanoids must play some role as arbiters of their own development. Then, another rational question naturally arises, concerning the future of humanoid robotics: "How we can motivate such development?". Speaking in purely utilitarian terms, emotion is the implementation of a motivational system that propels us to work, improve, reproduce, and survive. In reality, many of

our human "weaknesses" actually serve powerful biological purposes. Thus, if we want useful, human-like robots, we will have to give them some motivational system. We may choose to call this system "emotion" or we may reserve that term for ourselves and assert that humanoids are merely simulating emotion using algorithms whose output controls facial degrees of freedom, tone of voice, body posture, and other physical manifestations of emotion.

As it is evident by the most significant humanoid projects worldwide, most likely two distinct types of humanoids will arise in the future. In the first type, there will be those robots that we simply want to do work, such as daily tasks over and over again with no drip of emotion. In the other type, there will be the "emotional" humanoids, which will respond and interact with us and our emotions, rather than only be a non-stop working labor. Many argue that no robot really feels or knows anything that we have not (directly or indirectly) told them to feel or know. From this point of view, it seems unnecessary to give a second though to our treatment of humanoids. We could just say they are not "real", they are only machines.

At their dawn, all technologies seem artificial, and upset the perceived natural way of things. For example, with the rise of the Internet, a new "virtual world" was discovered. Nowadays, it is hard to distinguish between a "real world" and that "virtual world". To someone who spends 10 hours a day logged into Internet chat rooms or play cooperative online games, the so-called "virtual world" is as real to them, as anything else in their lives. Likewise, the interactions humans will have with humanoids will be real, because we make them be such. To this end, I believe this question is hard to answer. Even many years from now, someone will be confused by the question "Does a robot have real intelligence?". Intelligence is whatever intelligence does. Ultimately, the most relevant issue is not whether a robot's emotion or intelligence can be considered "real", but rather the fact that, real or not, it will have an effect on us and our daily lives.

The real danger is not that humanoids will make us mad with power or that humanoids will themselves become super intelligent and take over the world as in K. Capek's "Rossum's Universal Robots". But, as the Internet had negative consequences, so may have the machines. Will humans start interacting more and more with machines and less with each other? Already many humans spent large percentages of their life time interacting with machines, for work, education, entertainment, and other purposes. Furthermore, many of us prefer to forge and maintain relationships via e-mail, chat rooms,

and instant messenger rather than in person. Despite promises that the Internet will take us anywhere, we find ourselves, hour after hour, glued to our chairs. We supposedly live in a world with no borders. Yet, while at this time we should be coming closer together, it seems we are growing further apart. I believe humanoids may accelerate this trend.

If it is hard to imagine how humans could develop an emotional connection to a robot, consider what the effects would be of systematically imparting knowledge, personality, and intentions to a robot over a sustained period of time. It may well be that much of the software for intelligent humanoid robot control is developed under an Open Source paradigm, which means that thousands or even millions of developers will be able to modify the software of their own or other people's robots. Source code aside, humanoids will be given the ability to develop and learn in response to the input they receive. Could a cruel master make a cruel humanoid? Will people begin to see their robots as a reflection of themselves? As works of art? As valuable tools? As children? If humanoids learn "bad behavior", whom should we hold responsible? The manufacturer? The owner? The robot itself? Or the surrounding environment as a collective whole?

Isaac Asimov believed that robots should be invested with underlying rules that govern all behavior. Although generations of readers have admired and enjoyed Asimov's ability to depict the theoretical interplay of these rules, it may be that such encompassing, high-level rules are simply impracticable from a software engineering perspective. Robot intelligence is the emergent effect of layered, low-level mappings from sensing to action. Already, software developers are often unable to predict the emergent effect of these behaviors, when subjected to a non-Markovian (i.e. real-world) environment.

Whatever else it may be, technological progress flows with a swift current. The Internet continues to grow with little oversight, offering an incredible wealth of information and services, while at the same time presenting a new and devastating opportunity for fraud, theft, disruption of commerce and dissemination of misinformation. Humanoids pose a grave threat for the very reason that they will be of great service. As our technologies become more complex, more pervasive and more dangerous, we will be ever more likely to employ the aid of humanoids. They will not come in to work late, get tired, or demand profit sharing, promotion, etc. Although they will never be perfect, humanoids may someday prove more reliable than their creators.

Most likely, humanoids will never rise up and take control from our hands. Instead, we may give it to them, one home, one factory, one nuclear facility at a time until "pulling the

plug" becomes at first infeasible and then eventually unthinkable. Even now, imagine the economic chaos, if we were to disable the Internet. We are steadily replacing the natural world with the products of our own minds and hands. As we continue to disrupt and manipulate the existing state of our world, the changes we make require successive intervention. Technologies engender and demand new technologies. Once unleashed, it is difficult to revoke a technology without incurring profound economic, social, and psychological consequences. Rather, the problems that arise from new technologies are often met with more complex and daring technologies.

Yet, no matter how quickly technological progress seems to unfold, foresight and imagination will always play key roles in altering the society. We cannot hand up responsibility by calling the future inevitable. Neither we nor our creations stand outside the natural world, but rather we are all an integral part of it. We have designed humanoids to model and extend aspects of ourselves and, if we fear them, it is because we fear ourselves.

Chapter 3

Locomotion Stability Criteria

I did have strange ideas during certain periods of time. John F. Nash (1928–)

Humanoid robot locomotion, is a crucial and active research field worldwide. Since the origins of this research area, the dynamic stability criteria for obtaining a stable walking gait has always been the main point of interest. The most popular and widely-accepted criterion, the Zero Moment Point (ZMP), has been proposed by Professor Miomir Vukobratovic and his coworkers in 1970 [1]. The ZMP has been the base for many research projects on biped gait planning, analysis, and control. In Section 3.1 a thorough presentation for the ZMP is given. Although ZMP has proven very successful even with the most advanced humanoid robots, other researchers have proposed new criteria for dynamically stable locomotion. Dr. Ambarish Goswami in 1999 proposed the Foot Rotation Indicator (FRI) point, which extends the ZMP criterion, in the sense that it is not necessary for the ZMP to remain inside the convex hull to obtain a stable dynamic gait. The FRI is briefly presented in Section 3.2. Furthermore, in 2006 Dr. Hirohisa Hirukawa and his coworkers from the National Institute of Advanced Industrial Science and Technology (AIST) in Tsukuba, Japan proposed a more robust stability criterion which would prove useful for walking over rough or uneven terrain, the Contact Wrench Cone (CWC). This criterion is presented in Section 3.3, where it is stated that, if the sum of the gravity and inertia wrench applied to the COM of the robot is inside a polyhedral convex cone of the contact wrench between the feet of the robot and its environment, the gait is stable. Although FRI and CWC criteria have yielded some interesting results in robotic applications, this

work will be based on the ZMP criterion, mainly because it is the most well-studied in the last 30 years by many researchers and has led to successful results with the most advanced humanoid robots worldwide.

3.1 Zero Moment Point (ZMP)

In this section the ZMP concept will be presented in detail for the reader to fully understand its notion [33, 34]. Theoretical studies covering different aspects have been accompanied by extensive simulation work and limited physically-realized systems, from the simplest cases of planar mechanisms to the Honda, Kawada, and Sony humanoid robots, the most advanced biped locomotion robots designed up to now. Irrespectively of their structure and number of degrees of freedom (DOFs) involved, the basic characteristics of all biped locomotion systems [35] are: (i) the possibility of rotation of the overall system about one of the foot edges caused by strong disturbances, which is equivalent to the appearance of an unpowered (passive) DOF, (ii) gait repeatability (symmetry), which is related to regular gait only, and (iii) regular interchangeability of single- and double-support phases. During walking, two different situations arise in succession: the statically stable double-support phase, in which the robot mechanism is supported on both feet simultaneously, and the statically unstable single-support phase, in which only one foot of the mechanism is in contact with the ground, while the other is being transferred from the previous to the next step position. Thus, the locomotion mechanism changes its configuration during a single walking cycle from an open to a closed kinematic chain. All these circumstances have to be taken into account in artificial gait synthesis [36].

All of the biped mechanism joints are powered and are directly controllable, except for the contact between the foot and the ground (which can be considered as an additional passive DOF), where the interaction of the mechanism and the environment takes place. This contact is essential for the walk realization, because of the position of the mechanism with respect to the environment depends on the relative position of the foot/feet with respect to the ground.

The foot cannot be controlled directly, but only in an indirect way, by ensuring the appropriate dynamics of the mechanism above the foot. Thus, the overall indicator of the mechanism behavior is the point where the influence of all forces acting on the mechanism can be replaced by a single force. This point is termed the ZMP. Recognition

of the significance and the role of the ZMP in the artificial biped walk was a turning point in gait analysis, planning, and control. The seminal method for gait synthesis (semi-inverse method) was proposed by Vukobratovic and Juric [37]. It is noted that, despite the limitations of the synthesized motion, this method remained for a long time the only procedure available for biped gait synthesis. The ZMP is also indispensable in biped control for establishing the practically-unavoidable feedback with respect to dynamic ground reaction forces.

Apart from the realization of the relative motion of the mechanism's links, the most important task of a locomotion mechanism during the gait is to preserve its dynamic balance, which is achieved by that ensuring the whole area of the foot, and not only some edge, is in contact with the ground. The foot relies freely on the support and the only contact with the environment is realized via the friction force and vertical force of the ground reaction.

Let us consider the locomotion mechanism in the single-support phase shown in Figure 3.1(a), with the whole foot being on the ground. To facilitate the analysis we can neglect the part of the mechanism above the ankle of the support foot (point **A**) and replace its influence by force $\mathbf{F_A}$ and moment $\mathbf{M_A}$ (Figure 3.1(b)), whereby the weight of the foot itself acts at its center of mass (point **G**). The foot also experiences the ground reaction at point **P**, whose action keeps the whole mechanism in equilibrium. In general, the total ground reaction consists of three components of the force $\mathbf{R} = \begin{bmatrix} R_x & R_y & R_z \end{bmatrix}^\top$ and the moment $\mathbf{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^\top$.

Since the friction force acts at the point of contact of the foot with the ground and the foot on the ground is at rest, those components of force **R** and the moment **M** that act in the horizontal plane will be balanced by friction. Therefore, the horizontal reaction force (R_x, R_y) represents the friction force that balances the horizontal component of the force $\mathbf{F_A}$, whereas the vertical reaction moment M_z represents the moment of friction reaction forces (Figure 3.1(c)) that balances the vertical component of moment $\mathbf{M_A}$ and the moment inducted by force $\mathbf{F_A}$. Thus, if we assume that the foot-floor contact experiences no sliding, static friction will compensate for the horizontal force components (R_x, R_y) and the vertical reaction torque M_z. The vertical reaction force R_z represents the ground reaction that balances vertical forces. It remains to consider the balancing of the horizontal component of the foot load moment. However, due to the unidirectional nature of the ground reaction force induced by the foot contact, which is always oriented

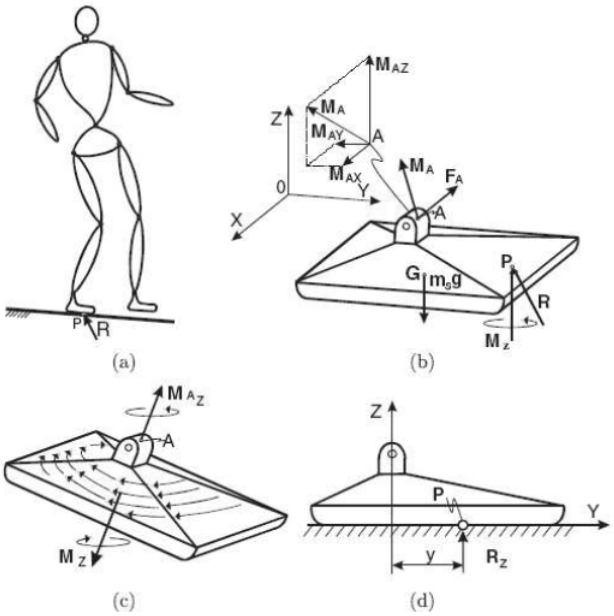

Figure 3.1: Biped mechanism and forces acting on its sole [1].

upwards, horizontal components of all active moments can be compensated for only by changing the position of the reaction force $\mathbf{R}$ within the support polygon. Therefore, the horizontal component of moment $\mathbf{M_A}$ will shift the reaction to the appropriate position to balance the additional load. This is illustrated in Figure 3.1(d), where, for the sake of simplicity, we present a simple planar case in the $y-z$ plane. Moment $M_{\mathbf{A}_x}$ is balanced by shifting the acting point of force R_z, which induces a moment whose intensity is determined by the distance y. It is necessary to emphasize that throughout the above, if the reaction force stays within the area covered by the foot, the increase in the ankle moment will be compensated for by changing the position of this force and no horizontal components of the moments M_x and M_y will exist. This is the reason only the M_z component appears at point $\mathbf{P}$ in Figure 3.1(b).

However, if the real support polygon is not large enough to encompass the appropriate position of force $\mathbf{R}$ to balance the action of external moments, force $\mathbf{R}$ will act at some foot edge and the uncompensated part of the horizontal component of the reaction moment will cause the mechanism's rotation about the foot edge, which can result in the mechanism's overturning. Therefore, we can say that the necessary and sufficient

condition for the locomotion mechanism to be in dynamic equilibrium is:

$$\begin{aligned} M_x &= 0 \\ M_y &= 0 \end{aligned} \tag{3.1}$$

at the point **P** on the sole, where the ground reaction force is acting. Since both components relevant to the realization of dynamic balance are equal to zero, a natural choice for naming this point was Zero Moment Point. From a different point of view, the reaction of the ground due to the foot resting on it can always be reduced to force **R** and the vertical component of moment M_z; point **P** on which the reaction force is acting represents ZMP.

Definition 3.1 (Zero Moment Point)**.** *The point on the plane around which the horizontal rotation momenta vanish is called the Zero Moment Point (ZMP).*

Now, a question can be naturally posed: "given the mechanism dynamics, what should the ZMP position be to ensure dynamic equilibrium?" It should be noted that in view of the fact that the entire mechanism is supported on the foot, a prerequisite for the mechanism's dynamic equilibrium is that the foot rests fully on the floor. Thus, to answer the previous question, let us state the static equilibrium equations for the supporting foot (Figure 3.1(b)):

$$\mathbf{R} + \mathbf{F}_A + m_s \boldsymbol{g} = 0 \tag{3.2}$$

$$\overrightarrow{OP} \times \mathbf{R} + \overrightarrow{OG} \times m_s \boldsymbol{g} + \mathbf{M_A} + \mathbf{M}_z + \overrightarrow{OA} \times \mathbf{F_A} = \mathbf{0} \tag{3.3}$$

where m_s is the foot mass, $\overrightarrow{OP}$, $\overrightarrow{OG}$, and $\overrightarrow{OA}$ are the radius vectors from the origin of the global coordinate frame $\mathbf{O}_{xyz}$ to the ground reaction force acting point **P**, foot mass center **G**, and ankle joint **A** respectively. If we place the origin of $\mathbf{O}_{xyz}$ at point **P** and project Eq.(3.3) onto the z-axis, then the vertical component of the ground reaction moment is actually the ground friction moment

$$\mathbf{M}_z = \mathbf{M}_{fr} = -\left(\mathbf{M}_\mathbf{A}^z + \left(\overrightarrow{OA} \times \mathbf{F_A}\right)^z\right) \tag{3.4}$$

Generally, M_z is not zero and can be reduced to zero only by appropriate changes in the mechanism dynamics. Next, projecting Eq.(3.3) onto the horizontal plane xy gives

$$\left(\overrightarrow{OP} \times \mathbf{R}\right)^{xy} + \overrightarrow{OG} \times m_s \boldsymbol{g} + \mathbf{M}_\mathbf{A}^{xy} + \left(\overrightarrow{OA} \times \mathbf{F_A}\right)^{xy} = \mathbf{0} \tag{3.5}$$

The position of the ground location force acting point **P** can be directly computed by Eq.(3.5). This equation represents the mathematical expression of the foot in equilibrium and gives the position of the ZMP that will ensure dynamic equilibrium for the overall underlying mechanism.

One may also consider the inverse question: "given an arbitrary motion of the mechanism, will the mechanism remain in dynamic equilibrium?" To answer such a question we have to consider the relationship between the computed position of **P** and the support polygon. If the position of **P**, computed by Eq.(3.5), is within the support polygon, then the mechanism is dynamically stable. Therefore, we can state the following:

Definition 3.2 (Dynamic Stability)**.** *If the ZMP remains within the convex hull created by the foot/feet, then the humanoid robot is dynamically stable.*

In real humanoid robots, point **P** cannot exist outside the support polygon, since, in that case, the reaction force **R** cannot act on the system at all. As already mentioned, in order to ensure dynamic equilibrium, the point **P** that satisfies Eq.(3.5) must be within the support polygon. If we suppose for a moment that point **P** lies outside the support polygon and considering the fact that the position of **P** was obtained from the condition $M_x = M_y = 0$, we conclude that it is a fictitious ZMP (FZMP).

Let's get a deeper insight. It is clear from Eqs.(3.2),(3.3) that the ZMP position depends on the mechanism dynamics (i.e. on $\mathbf{F_A}$ and $\mathbf{M_A}$). In a situation where the mechanism dynamics change and point **P** approaches the support polygon edge (in either single-support or double-support phase), **P** will remain the ZMP, only if no additional moments are acting on it. However, if an additional moment appears, the locomotion mechanism would start to rotate about the foot edge and the mechanism will eventually tip over. In such a situation, the acting point of ground reaction force would be on the foot edge, but this point would not be the ZMP any more, since the conditions $M_x = 0$ and $M_y = 0$ are not satisfied simultaneously.

To further clarify the meaning of the ZMP outside the support polygon (FZMP), let us recollect the problem we posed above: determining the ZMP position, given a motion of the mechanism. In the case of a real walking mechanism equipped with Force Sensitive Resistor (FSR) sensors on the soles, information about the ZMP position can be obtained by measuring forces acting at the contact of the ground and the mechanism [35]. However, if the biped gait is investigated using a dynamic model, the ZMP position can instead be

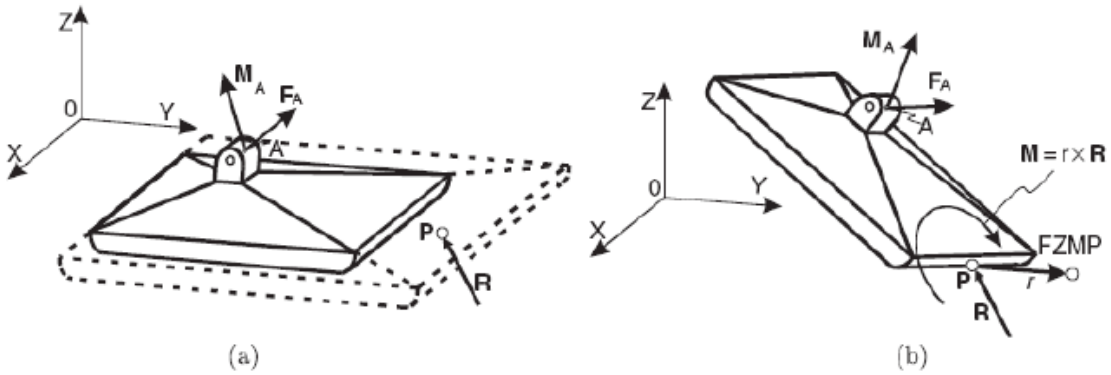

Figure 3.2: Illustration of the determination of the ZMP position [1]: (a) Step 1, and (b) Step 2.

computed analytically. For a given mechanism motion, the force and moment at the ankle joint ($\mathbf{F_A}$ and $\mathbf{M_A}$) can be obtained from the model of the mechanism dynamics and all elements in Eq.(3.5) except from $\overrightarrow{OP}$ will be known. The procedure for determining the position of the ZMP can be described in two steps:

1. **Step 1**. Compute $\overrightarrow{OP}$ from Eq.(3.5) (see Figure 3.1). Let us call the obtained position of **P** the *computed ZMP position*. Note again that at this moment we do not know whether the position of point **P** will be within the real support polygon or outside it.

2. **Step 2**. The computed ZMP position is just a candidate to be a regular ZMP and its position should be compared with the real support polygon size. If the computed ZMP is outside the support polygon, this means that the ground reaction force acting on **P** is actually on the edge of the support polygon and the mechanism rotation about the support polygon edge will be initiated by the unbalanced moment, whose intensity depends on the distance from the support polygon edge to the computed position of ZMP, i.e. the distance to the FZMP position.

These steps are illustrated in Figure 3.2. In Step 1, we obtain an answer to the question concerning the ZMP location for the given dynamic state of the mechanism, not taking into account the real support polygon size (Figure 3.2(a)), whereas in Step 2, we answer whether, regarding the real support polygon size, the mechanism is dynamically stable and where the regular ZMP location is (provided it exists). If this is not the case, the

ground reaction force acting point will be on the edge of the support polygon and the distance from it to the computed ZMP position is proportional to the intensity of the perturbation moment that acts on the foot, as shown in Figure 3.2(b).

The concept of ZMP has been properly comprehended by researchers, it is widely used, and it is frequently cited. It can be noted that, although being essentially correct, all the ZMP definitions differ significantly in the extent of their detail. To clarify things, Vukobratovic and his coworkers pointed out another important issue, namely the difference between the Center of Pressure (CoP) and the ZMP. It is quite important to make a clear distinction between the two notions, which must not be regarded as identical, in general. The pressure between the foot and the ground can always be replaced by a force acting at the CoP. If this force balances all active forces acting on the mechanism during the motion (inertia, gravitation, Coriolis and centrifugal forces and moments), its acting point is indeed the ZMP. Thus, in the case of a dynamically stable gait, the CoP and the ZMP coincide. When the gait is not dynamically stable, the ZMP does not exist and the mechanism tips about the foot edge. To make the ZMP notion and its relationship to the CoP perfectly clear, Vukobratovic and his coworkers illustrated these observations using three characteristic cases for a non-rigid foot in contact with the ground, as sketched out in Figure 3.3. In a stable gait, the ZMP coincides with the CoP (Figure 3.3(a)). In the case of a disturbance that brings the acting point of the ground reaction force to the foot edge, the perturbation moment will cause rotation of the biped system about the foot edge. In that case, we can only talk about the FZMP, whose distance from the foot edge represents the intensity of the perturbation moment (Figure 3.3(b)). However, it is possible to realize the biped motion, for example on the toe tips with special shoes having a pinpoint area (balletic motion), while keeping the ZMP position within the pinpoint area (Figure 3.3(c)). Although the ZMP now coincides with CoP, it is not a regular gait and the biped should be specially trained to perform it. Here, it is necessary to be reminded that the task of deriving a model of nominal dynamics for a humanoid robot is concerned with satisfying a certain number of dynamic connections. This is in fact the so-called mixed type of task, where the link's motion and the driving torques are both partially known and their complements are sought. In the case of investigating the dynamics of biped structure, the motion of the links performing a given type of gait is known, while the known moments are equal to zero. The latter follows from the equilibrium conditions holding for a selected point within the support polygon and for the joints of passive links.

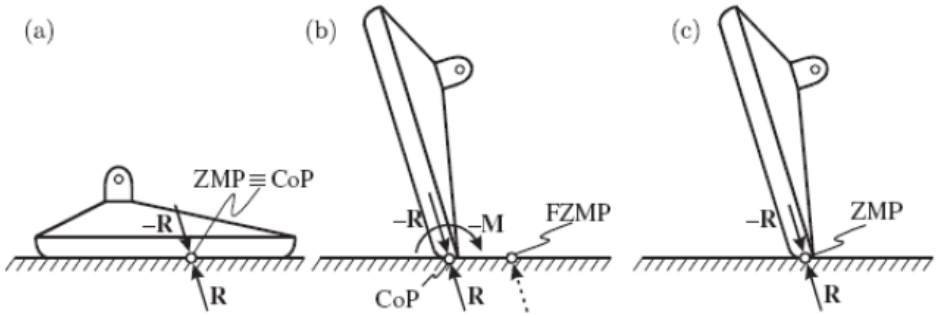

Figure 3.3: Possible relations between the ZMP and the CoP for a non-rigid foot: (a) dynamically stable gait, (b) unbalanced gait, and (c) tiptoe dynamic balance [1].

Therefore, there are two types of zero-moment points. Both of them serve to form the model of nominal dynamics of the humanoid robot, but those within the support polygon are practically unavoidable in gait synthesis, as well as for the overall control of a dynamically balanced gait.

Viewing the ZMP as being identical to the CoP is not correct, as the ZMP can exist at other points in the system, e.g. at the shoulder joints, if we consider arms as freely-swinging pendulums with no actuators at the joints. In summary, the ZMP coincides with the CoP only during a dynamically stable gait. Furthermore, the FZMP never coincides with the CoP, because the CoP cannot, naturally, exist outside the support polygon.

3.2 Foot-Rotation Indicator (FRI) Point

The Foot-Rotation Indicator (FRI) point was introduced by Dr. A. Goswami [2]. In his paper, he views the entire biped robot as a general n-segment extended rigid-body kinematic chain, as shown in Figure 3.4(left). To formally introduce the FRI point, we first treat the biped robot as a system and determine its response to external force/torque. We may employ Newton's or d' Alembert's principle for this purpose. The external forces acting on the robot are the resultant ground reaction force/torques, $\mathbf{R}$ and $\mathbf{M}$, acting at the CoP (denoted by $\mathbf{P}$) and the gravity, as shown in Figure 3.4(right). The equation for rotational dynamic equilibrium[1] is obtained by noting that the sum of the external

[1]We deal with rotational equilibrium only and do not discuss translational equilibrium or "sliding", assuming that the foot/ground friction is sufficiently large to prevent it.

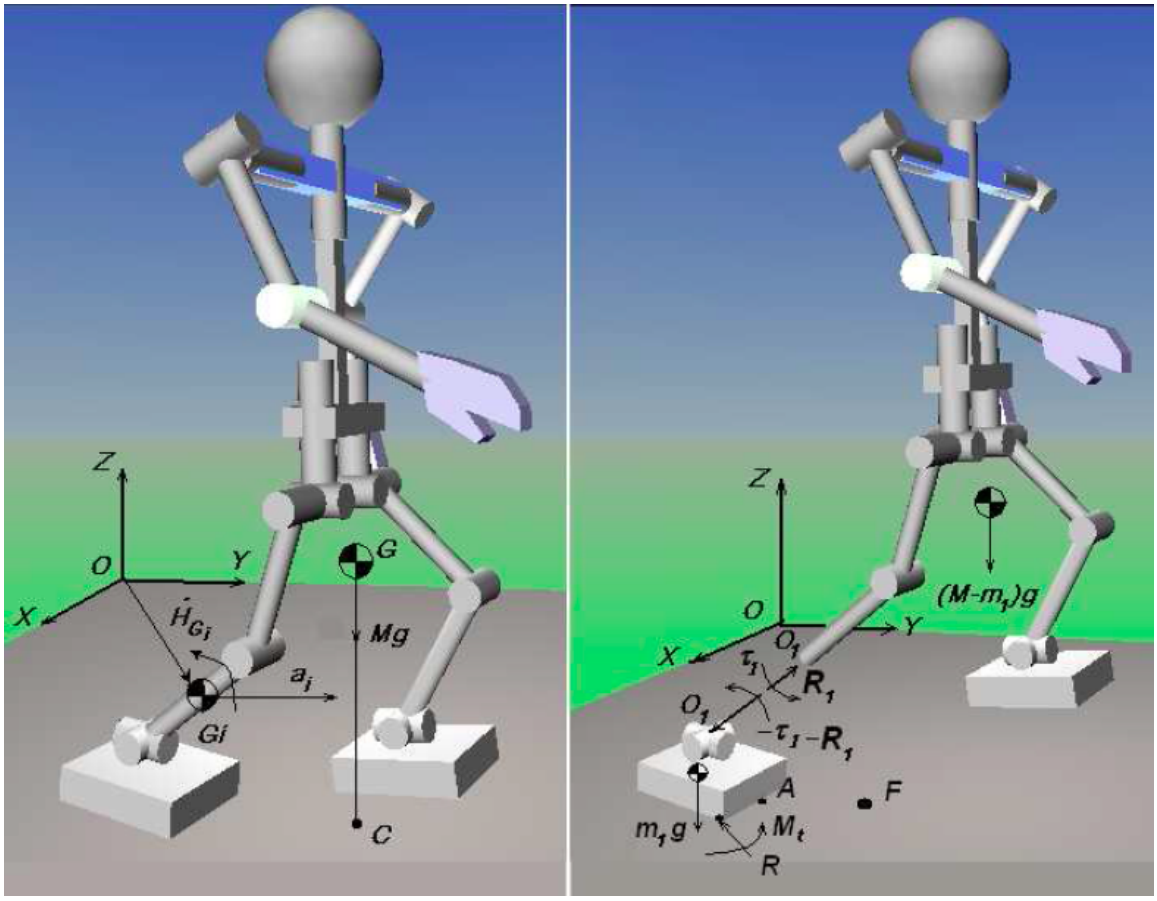

Figure 3.4: The sketch of a 3D extended rigid-body biped robot (left) and a view with its support foot artificially disconnected from the shank to show the intervening forces (right) [2].

moments on the robot, computed either at its Center of Mass Ground projection (GCoM) or at any stationary reference point, is equal to the sum of the rates of change of angular momenta of the individual segments about the same reference point. Taking moments at the origin **O**, we have

$$\mathbf{M} + \overrightarrow{OP} \times \mathbf{R} + \sum_{i=1}^{n} \overrightarrow{OG_i} \times m_i \boldsymbol{g} = \sum_{i=1}^{n} \dot{\mathbf{H}}_{G_i} + \sum_{i=1}^{n} \overrightarrow{OG_i} \times m_i \mathbf{a}_i \tag{3.6}$$

where m_i is the mass, G_i is the CoM location, a_i is the CoM linear acceleration, and H_{G_i} is the angular momentum about the CoM of the i-th link of the robot.

Let's now focus on the dynamic behavior governing the humanoid robot in the single-

support phase. As seen in Figure 3.4, an important aspect of this approach is to treat the single-support foot as the focus of attention. Since the foot is the only robot part interacting with the ground, it is subjected to joint forces, gravity forces, and the ground-reaction forces. Viewed from the foot, the dynamics of the rest of the robot may be completely represented by the ankle force/torque $-R_1$ and $-\tau_1$ (where negative signs are used by convention). Figure 3.4(right) artificially disconnects the support foot from the shank to clearly show the active forces at that joint. The dynamics equation of the single-support foot is:

$$\mathbf{M} + \overrightarrow{OP} \times \mathbf{R} + \overrightarrow{OG_1} \times m_1\boldsymbol{g} - \tau_1 - \overrightarrow{OO_1} \times \mathbf{R}_1 = \dot{\mathbf{H}}_{G_1} + \overrightarrow{OG_1} \times m_1\mathbf{a}_1 \tag{3.7}$$

which, at equilibrium, reduces to:

$$\mathbf{M} + \overrightarrow{OP} \times \mathbf{R} + \overrightarrow{OG_1} \times m_1\boldsymbol{g} - \tau_1 - \overrightarrow{OO_1} \times \mathbf{R}_1 = \mathbf{0} \tag{3.8}$$

Recall Eq.(3.8) can be expressed with respect to any other stationary reference point. Out of these, the CoP represents a special point, where Eq.(3.8) reduces to

$$\mathbf{M} + \overrightarrow{PG_1} \times m_1\boldsymbol{g} - \tau_1 - \overrightarrow{PO_1} \times \mathbf{R}_1 = \mathbf{0} \tag{3.9}$$

Considering only the horizontal (xy) vector components of Eq.(3.9), we may write

$$\left(\tau_1 + \overrightarrow{PO_1} \times \mathbf{R}_1 - \overrightarrow{PG_1} \times m_1\boldsymbol{g}\right)^{xy} = \mathbf{0} \tag{3.10}$$

Since $\mathbf{M}$ is tangential to the foot/ground surface, its vector direction is normal to that surface and does not contribute to this equation (we ignore foot rotation about the normal ground, as it does not contribute to a loss of balance). In the presence of an unbalanced torque on the foot, Eq.(3.10) is not satisfied for any point within the support polygon. One may, however, still find a point $\mathbf{F}$ outside the support boundary that satisfies Eq.(3.10)

$$\left(\tau_1 + \overrightarrow{FO_1} \times \mathbf{R}_1 - \overrightarrow{FG_1} \times m_1\boldsymbol{g}\right)^{xy} = \mathbf{0} \tag{3.11}$$

Point $\mathbf{F}$ is called the FRI point [2, 3] and is formally defined as

Definition 3.3 (Foot Rotation Indicator Point)**.** *The point on the foot/ground contact surface, within or outside the convex hull of the foot-support area, at which the resultant moment of the force/torque imposed on the foot is normal to the surface, is the Foot Rotation Indicator (FRI) point.*

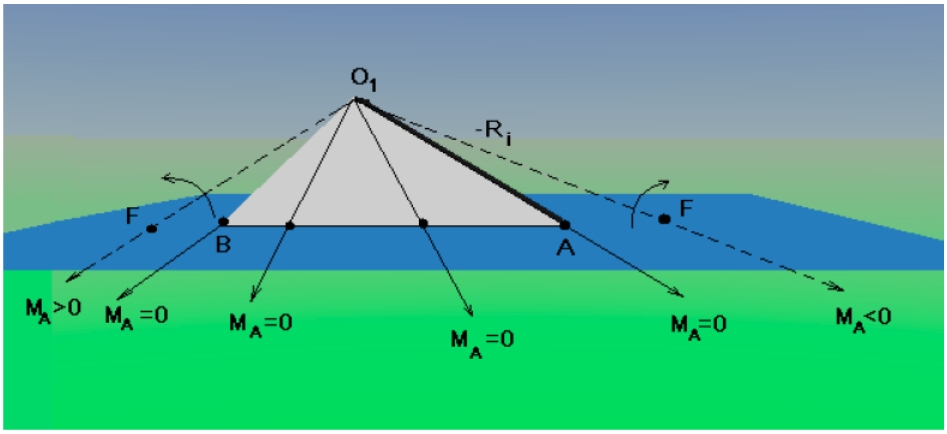

Figure 3.5: Condition for foot rotation, when $\boldsymbol{\tau}_1 = 0$. The figure sketches different lines of action of the force $\mathbf{R}_1$ applied to the robot foot by the rest of the robot at ankle joint $\mathbf{O}_1$ [2].

By the term "force/torque imposed", we mean the force and torque at the ankle joint, other external forces, plus the weight of the foot, but not the ground-reaction forces. We may identify the imposed forces as the acting forces, in contrast to the reaction forces from the ground, which are the constraining forces.

An intuitive understanding of the FRI point is obtained by setting $\boldsymbol{\tau}_1 = 0$ and $m_1 = 0$ in Eq.(3.11). In this case, $\mathbf{F}$ is simply the point on the ground where the line of action of $\mathbf{R}_1$ penetrates the ground, as shown in Figure 3.5. The case of the unactuated ankle joint was considered by Lee and Raibert in 1991 [38] to analyze the hoof rotation in a monopod.

It is important to note that the location of the ankle joint and the geometry of the support-polygon boundary are the only important features of the foot that are relevant to our discussion. The actual physical shape of the foot is not important. See Figure 3.6 for a graphical illustration of this fact.

Explicit expressions for the coordinates of $\mathbf{F}$, $\overrightarrow{OF} = (OF_x, OF_y, OF_z = 0)$, are obtained by computing the dynamics of the robot minus the foot at $\mathbf{F}$:

$$\mathbf{M} + \overrightarrow{OP} \times \mathbf{R} + \sum_{i=2}^{n} \overrightarrow{OG_i} \times m_i \boldsymbol{g} = \sum_{i=2}^{n} \dot{\mathbf{H}}_{G_i} + \sum_{i=2}^{n} \overrightarrow{OG_i} \times m_i \mathbf{a}_i \tag{3.12}$$

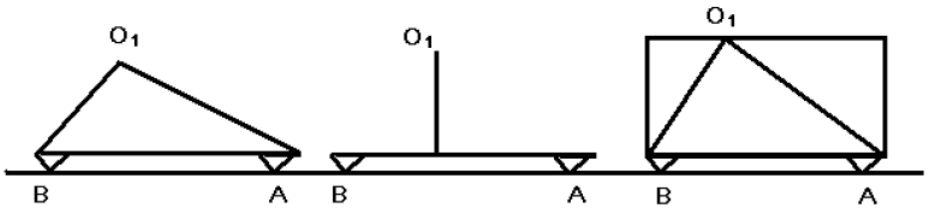

Figure 3.6: The locations of key points, namely the ankle-joint location ($\mathbf{O}_1$) and the support-polygon boundary ($\mathbf{A}$ and $\mathbf{B}$), not its overall geometry, are relevant to the behavior of the foot. The three examples of the robot foot shown above have identical behavior, although their geometries are very different [2].

Using Eq.(3.11) and considering only the horizontal components, we have:

$$\left(\overrightarrow{FG_1} \times m_1\boldsymbol{g} + \sum_{i=2}^{n} \overrightarrow{FG_i} \times m_i(\boldsymbol{g} - \mathbf{a}_i)\right)^{xy} = \left(\sum_{i=2}^{n} \dot{\mathbf{H}}_{G_i}\right)^{xy} \tag{3.13}$$

Noting that $FG_i = FO + OG_i$ and $OF = -FO$, Eq.(3.13) may be rewritten as

$$\left(\sum_{i=2}^{n} \overrightarrow{OF} \times m_i\left(\mathbf{a}_i - \boldsymbol{g}\right) - \overrightarrow{OF} \times m_1\boldsymbol{g}\right)^{xy} = \left(-\overrightarrow{OG_1} \times m_1\boldsymbol{g} + \sum_{i=2}^{n} \dot{\mathbf{H}}_{G_i} + \sum_{i=2}^{n} \overrightarrow{OG_i} \times m_i\left(\mathbf{a}_i - \boldsymbol{g}\right)\right)^{xy} \tag{3.14}$$

Carrying out the operation, we may finally obtain

$$\overrightarrow{OF}_x = \frac{m_1\overrightarrow{OG}_{1y}\boldsymbol{g} + \sum_{i=2}^{n} m_i\overrightarrow{OG}_{iy}\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)}{m_1\boldsymbol{g} + \sum_{i=2}^{n} m_i\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)} - \frac{\sum_{i=2}^{n} m_i\overrightarrow{OG}_{iz}\mathbf{a}_{iy} + \sum_{i=2}^{n} \dot{\mathbf{H}}_{G_{ix}}}{m_1\boldsymbol{g} + \sum_{i=2}^{n} m_i\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)} \tag{3.15}$$

$$\overrightarrow{OF}_y = \frac{m_1\overrightarrow{OG}_{1x}\boldsymbol{g} + \sum_{i=2}^{n} m_i\overrightarrow{OG}_{ix}\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)}{m_1\boldsymbol{g} + \sum_{i=2}^{n} m_i\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)} - \frac{\sum_{i=2}^{n} m_i\overrightarrow{OG}_{iz}\mathbf{a}_{ix} + \sum_{i=2}^{n} \dot{\mathbf{H}}_{G_{iy}}}{m_1\boldsymbol{g} + \sum_{i=2}^{n} m_i\left(\mathbf{a}_{iz} + \boldsymbol{g}\right)} \tag{3.16}$$

Some useful properties of the FRI point which may be exploited in gait planning include the following:

- The FRI point indicates the "occurrence" of foot rotation, as already described.
- The location of the FRI point indicates the "magnitude" of the unbalanced moment on the foot. The total moment $\mathbf{M}_{\mathbf{A}}^{I}$ due to the impressed forces about a point $\mathbf{A}$ on the support-polygon boundary (Figure 3.4, right) is

$$\mathbf{M}_{\mathbf{A}}^{I} = \overrightarrow{AF} \times \left(m_1\boldsymbol{g} - \mathbf{R}_1\right) \tag{3.17}$$

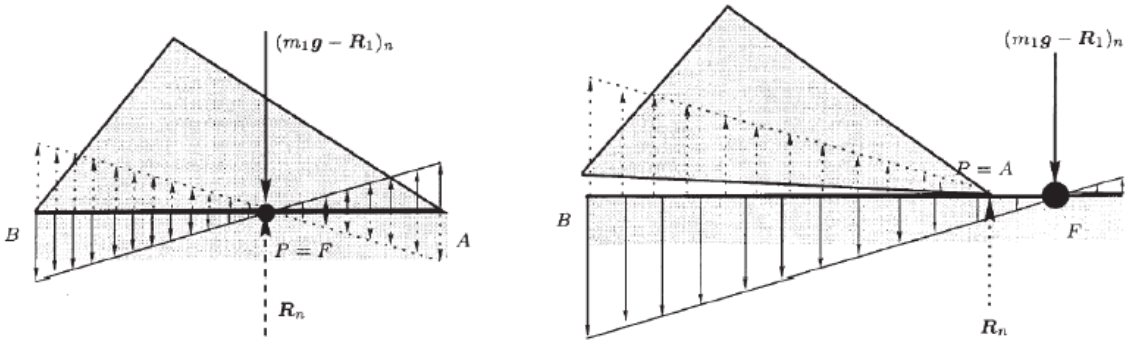

Figure 3.7: The magnitudes of the moments at different points is proportional the length of the arrows. Clockwise (i.e., negative) moments are shown by upward-pointing arrows. In the image on the left the moments are precisely compensated, whereas in the image on the right they are not. The subscript n denotes the normal component of a force [3].

which is proportional to the distance between **A** and **F**. If **F** is situated inside the support polygon, $\mathbf{M}_\mathbf{A}^I$ is counteracted by the moment due to **R** and is precisely compensated; see Figure 3.7 (left) for a planar example. Otherwise, $\mathbf{M}_\mathbf{A}^I$ is the uncompensated moment that causes the foot to rotate; see Figure 3.7 (right).

- The FRI point indicates the "direction" of foot rotation. This is derived from Eq.(3.17), assuming that $m_1\boldsymbol{g} - \mathbf{R}_1$ is directed downwards.

- The FRI point indicates the "stability margin" of the robot, which may be quantified as the minimum distance of the support-polygon boundary from the current location of the FRI point within the footprint. Conversely, when the FRI point is outside the footprint, this minimum distance is a measure of instability of the robot. An imminent foot rotation will be indicated by a motion of the FRI point towards the support-polygon boundary.

3.3 Contact Wrench Cone (CWC)

In 2006, Dr. Hirukawa and coworkers of the AIST proposed a universal stability criterion of the foot contact of humanoid robots [4]. The proposed method checks whether the sum of the gravity and the inertia wrench applied to the CoM of the robot is inside

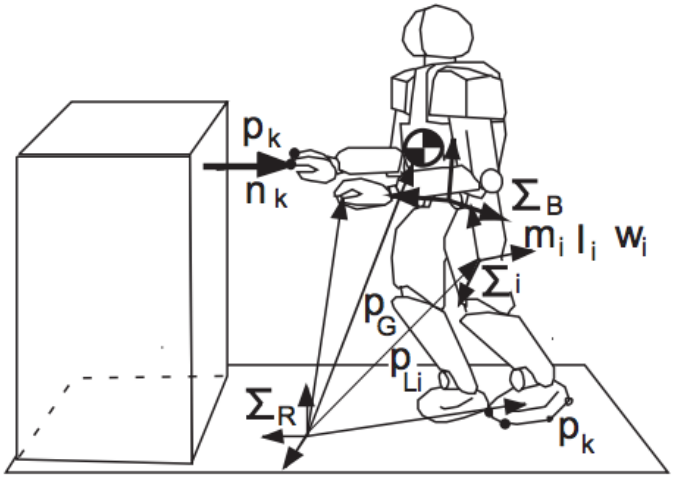

Figure 3.8: Coordinate Frames [4]

the polyhedral convex cone of the contact wrench between the feet of the robot and its environment, as shown in Figure 3.8, and determine stability. The criterion can be used to determine the strong stability of the foot contact when a robot walks on an arbitrary terrain (even rough terrain) and/or when the hands of the robot are in contact with it under the sufficient friction assumption. This procedure is equivalent to checking whether the ZMP is inside the support polygon of the feet when the robot walks on a horizontal plane with sufficient friction. Finally, when the friction follows a physical law, the criterion can also be used to determine whether the foot contact is weakly stable. Therefore, the proposed criterion can be used to judge the behavior of the ZMP in more general cases.

Figure 3.8 illustrates a legged robot whose hands may be in contact with the environment, where Σ_R is the reference coordinate frame, Σ_B is the coordinate frame fixed to the waist of the humanoid robot, and Σ_{L_i} is the coordinate frame fixed to the CoM of the i-th link of the robot with $i = 1, \ldots, N$. Next, let $p_{L_i} = \begin{bmatrix} x_{L_i} & y_{L_i} & z_{L_i} \end{bmatrix}^\top$ be the origin of Σ_{L_i}, $\mathbf{p}_B = \begin{bmatrix} x_B & y_B & z_B \end{bmatrix}^\top$ the origin of Σ_B with respect to Σ_R, $\mathbf{p}_k$ with $k = 1, \ldots, K$ be the vertices of the support polygons of the hands and feet, and $\mathbf{p}_G = \begin{bmatrix} x_G & y_G & z_G \end{bmatrix}^\top$ be the position of the CoM of the robot. In addition, let $\boldsymbol{f}_k$ be the force applied to the robot at $\mathbf{p}_k$, $\mathbf{n}_k$ be the unit normal vector at $\mathbf{p}_k$ pointed to the robot, and let $\mathbf{I}_i$ and $\boldsymbol{\omega}_i$ denote the inertia tensor and angular velocity of the i-th link with respect to the reference

coordinate frame Σ_R. Then, the sum $\boldsymbol{f}_G$ of the gravity and the force of inertia applied to the robot and the sum $\boldsymbol{\tau}_G$ of the moments about the CoM of the robot with respect to the reference coordinate frame Σ_R are given by

$$\boldsymbol{f}_G = M\left(\boldsymbol{g} - \ddot{\mathbf{p}}_G\right) \tag{3.18}$$

$$\boldsymbol{\tau}_G = \mathbf{p}_G \times M\left(\boldsymbol{g} - \ddot{\mathbf{p}}_G\right) - \dot{\boldsymbol{\mathcal{L}}} \tag{3.19}$$

where M is the total mass of the robot, $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\top$ the gravitational vector, and $\boldsymbol{\mathcal{L}} = \begin{bmatrix} \mathcal{L}_x & \mathcal{L}_y & \mathcal{L}_z \end{bmatrix}^\top$ the angular momentum of the robot with respect to the CoM defined by

$$\boldsymbol{\mathcal{L}} = \sum_{i=1}^{N} \left\{ m_i \left(\mathbf{p}_{L_i} - \mathbf{p}_G\right) \times \dot{\mathbf{p}}_{L_i} + \mathbf{I}_i \boldsymbol{\omega}_i \right\} \tag{3.20}$$

with m_i being the mass of the i-th link of the robot. Next, let $\boldsymbol{f}_C$ be the contact force, which can be applied from the environment to the robot with respect to the reference coordinate frame Σ_R, and $\boldsymbol{\tau}_C$ the corresponding moment. These are given by

$$\boldsymbol{f}_C = \sum_{k=1}^{N} \sum_{l=1}^{L} \epsilon_k^l \left(\mathbf{n}_k + \mu_k \mathbf{t}_k^l\right) \tag{3.21}$$

$$\boldsymbol{\tau}_C = \sum_{k=1}^{N} \sum_{l=1}^{L} \epsilon_k^l \mathbf{p}_k \times \left(\mathbf{n}_k + \mu_k \mathbf{t}_k^l\right) \tag{3.22}$$

where the friction cone at $\mathbf{p}_k$ is approximated by an L-polyhedral cone, $\mathbf{t}_k^l$ is a unit tangent vector to make $\mathbf{n}_k + \mu_k \mathbf{t}_k^l$ be the l-th edge of the polyhedral cone, μ_k is the friction coefficient at $\mathbf{p}_k$, and ϵ_k^l a non-negative scalar, which gives the magnitude of the force of the l-th edge of the approximated friction cone at the k-th contact point. The set of $(\boldsymbol{f}_C, \boldsymbol{\tau}_C)$ forms a polyhedral convex cone in the space of the contact force and torque and is called *a polyhedral convex cone of the contact wrench*.

We are now ready to determine the stability of locomotion. The contact between the robot and the environment is strongly stable, when it is guaranteed that the contact is stable within $(\boldsymbol{f}_G, \boldsymbol{\tau}_G)$. The contact is weakly stable, when the contact may or may not be stable within $(\boldsymbol{f}_G, \boldsymbol{\tau}_G)$. The contact is strongly unstable, when it is not weakly stable. Strong stability cannot always be determined, since the contact force is indeterminate, in general. The contact is always weakly stable, when the motion of the robot is feasible, as discussed below. Let us assume that sufficient friction exists at the contact. The

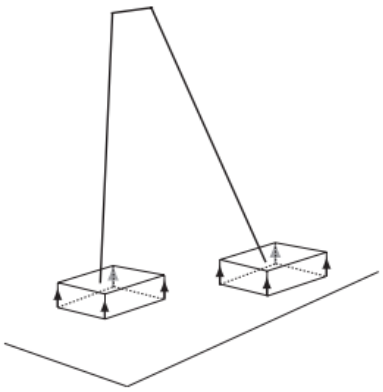

Figure 3.9: Two feet on a horizontal plane [4]

assumption implies that an arbitrary friction force can be generated at every contact point independently of the normal force at that point and it can be written as

$$\boldsymbol{f}_C = \sum_{k=1}^{K} \left(\epsilon_k^0 \mathbf{n}_k + \sum_{l=1}^{4} \epsilon_k^l \mathbf{t}_k^l \right) \tag{3.23}$$

$$\boldsymbol{\tau}_C = \sum_{k=1}^{K} \mathbf{p}_k \times \left(\epsilon_k^0 \mathbf{n}_k + \sum_{l=1}^{4} \epsilon_k^l \mathbf{t}_k^l \right) \tag{3.24}$$

where $\mathbf{t}_k^l$ with $l = 1, \ldots, 4$ are the unit tangent vectors at $\mathbf{p}_k$ whose non-negative k linear combination spans the horizontal plane. The authors conclude with the following stability criterion:

Theorem 3.1 (Strong Stability Criterion)**.** *If* $(-\boldsymbol{f}_G, -\boldsymbol{\tau}_G)$ *is an internal element of the polyhedral convex cone of the contact wrench given by Eqs.(3.21),(3.22), then the contact is strongly stable within* $(\boldsymbol{f}_G, \boldsymbol{\tau}_G)$*.*

Let us consider a case, where a biped robot walks on a horizontal plane with sufficient friction, as shown in Figure 3.9. Then, the horizontal elements of $\boldsymbol{f}_G$ and $\boldsymbol{\tau}_G$ about the z-axis should always be balanced by the contact force and torque:

$$M\ddot{x}_G = \sum_{k=1}^{K} \left(\epsilon_k^1 - \epsilon_k^2 \right) \tag{3.25}$$

$$M\ddot{y}_G = \sum_{k=1}^{K} \left(\epsilon_k^3 - \epsilon_k^4 \right) \tag{3.26}$$

$$Mx_G\ddot{y}_G - My_g\ddot{x}_G + \dot{\mathcal{L}}_z = \sum_{k=1}^{K}\left\{\left(\epsilon_k^3 - \epsilon_k^4\right)x_k - \sum_{k=1}^{K}\left(\epsilon_k^1 - \epsilon_k^2\right)y_k\right\} \tag{3.27}$$

The polyhedral convex cone of the contact wrench is the direct product of the linear subspace given by the right-hand side of Eqs.(3.25),(3.26),(3.27) and a polyhedral convex cone in the complement of the subspace, and therefore strong stability can be determined by checking if $(\boldsymbol{f}_G, \boldsymbol{\tau}_G)$ is inside the polyhedral convex cone in the complement subspace. The relationship in the complement subspace can be written as

$$M\left(\ddot{z}_G + g\right) = \sum_{k=1}^{K}\epsilon_k^0 \tag{3.28}$$

$$M\left(\ddot{z}_G + g\right)y_G - M\ddot{y}_G z_G + \dot{\mathcal{L}}_x = \sum_{k=1}^{K}\epsilon_k^0 y_k - z_0\sum_{k=1}^{K}\left(\epsilon_k^3 - \epsilon_k^4\right) \tag{3.29}$$

$$-M\left(\ddot{z}_G + g\right)x_G + M\ddot{x}_G z_G + \dot{\mathcal{L}}_y = -\sum_{k=1}^{K}\epsilon_k^0 x_k - z_0\sum_{k=1}^{K}\left(\epsilon_k^1 - \epsilon_k^2\right) \tag{3.30}$$

where z_0 is the height for the horizontal floor. Note that the second terms of the right-hand sides in Eqs.(3.29),(3.30) are independent of the positions of the contact points, since the z coordinate of all the contact points is z_0. Then, we can set $z_0 = 0$, without loss of the generality, and we obtain

$$M\left(\ddot{z}_G + g\right)y_G - M\ddot{y}_G z_G + \dot{\mathcal{L}}_x = \sum_{k=1}^{K}\epsilon_k^0 y_k \tag{3.31}$$

$$-M\left(\ddot{z}_G + g\right)x_G + M\ddot{x}_G z_G + \dot{\mathcal{L}}_y = -\sum_{k=1}^{K}\epsilon_k^0 x_k \tag{3.32}$$

From Eqs.(3.28),(3.31),(3.32), $(-\boldsymbol{f}_G, -\boldsymbol{\tau}_G)$ is an internal element of the polyhedral convex cone of the contact wrench, given by Eqs.(3.21),(3.22), if Eqs.(3.31),(3.32) hold for at least three positive ϵ_k^0 and then the contact is strongly stable by the strong stability theorem. Figure 3.10 illustrates the support polygon of the robot and the corresponding intersection of the polyhedral convex cone, given by the right-hand sides of Eqs.(3.21),(3.31),(3.32), with the plane $f_z = M(\ddot{z}_G + g)$. The set of $((\boldsymbol{\tau}_c)_x, (\boldsymbol{\tau}_c)_y)$ is the dual polygon of the support polygon, since x_k and y_k are exchanged in the right-hand sides of Eqs.(3.31),(3.32) with the minus sign in Eq.(3.32).

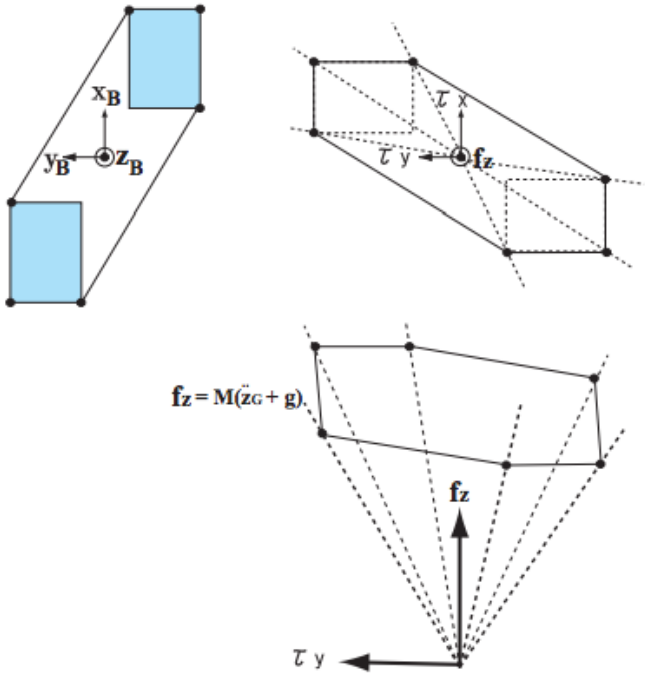

Figure 3.10: Support polygon and intersection of the polyhedral convex cone [4]

Let us consider the same contact stability using the ZMP. The ZMP $= (x_0, y_0)$ can be given by

$$x_0 = \frac{Mx_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{x}_G - \dot{\mathcal{L}}_y}{M\left(\ddot{z}_G + g\right)} \tag{3.33}$$

$$y_0 = \frac{My_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{y}_G - \dot{\mathcal{L}}_x}{M\left(\ddot{z}_G + g\right)} \tag{3.34}$$

The ZMP is an internal point of the support polygon defined by the feet, if

$$x_0 = \sum_{k=1}^{K} \lambda_k x_k \tag{3.35}$$

$$y_0 = \sum_{k=1}^{K} \lambda_k y_k \tag{3.36}$$

$$\sum_{k=1}^{K} \lambda_k = 1 \tag{3.37}$$

$$\lambda_k \geq 0 \tag{3.38}$$

and at least three λ_k's are positive. Let $z_0 = 0$; then, from Eqs.(3.33),(3.34) we obtain

$$\frac{Mx_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{x}_G - \dot{\mathcal{L}}_y}{M\left(\ddot{z}_G + g\right)} = \sum_{k=1}^{K} \lambda_k x_k \tag{3.39}$$

$$\frac{My_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{y}_G - \dot{\mathcal{L}}_x}{M\left(\ddot{z}_G + g\right)} = \sum_{k=1}^{K} \lambda_k y_k \tag{3.40}$$

Next, we can prove that the presented criterion is equivalent to the ZMP in this case. Let $\epsilon = \sum_{k=1}^{K} \epsilon_k^0$. Substituting Eq.(3.28) into Eqs.(3.31),(3.32), we get

$$\frac{Mx_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{x}_G - \dot{\mathcal{L}}_y}{M\left(\ddot{z}_G + g\right)} = \sum_{k=1}^{K} \frac{\epsilon_k^0}{\epsilon} x_k \tag{3.41}$$

$$\frac{My_G\left(\ddot{z}_G + g\right) - M\left(z_G - z_0\right)\ddot{y}_G - \dot{\mathcal{L}}_x}{M\left(\ddot{z}_G + g\right)} = \sum_{k=1}^{K} \frac{\epsilon_k^0}{\epsilon} y_k \tag{3.42}$$

It is trivial to see that Eq.(3.37) should be identical with Eq.(3.36), since $\sum_{k=1}^{K} \frac{\epsilon_k^0}{\epsilon}$. This proves that the proposed criterion is equivalent to the ZMP, when a humanoid robot walks on a horizontal plane with sufficient friction. Thus, from this example, it is evident that the ZMP criterion is a special case of the CWC criterion. A key question is whether the CWC criterion offers any advantage compared to the ZMP criterion. The CWC criterion does not need the division to find the ZMP in Eq.(3.33) and therefore its computation is more numerically stable, especially when the vertical contact force is small. The trajectory of the ZMP can be plotted more comprehensively, since it is a point on a plane. The CWC criterion requires an intersection plane of $\boldsymbol{f}_z$ to be plotted on a plane, as shown in Figure 3.10.

Finally, when the sufficient friction assumption is lifted, it is not possible to determine strong stability in general and the presented theorem does not hold [4]. When motion patterns of a legged robot are planned, the generated patterns may not be feasible in the physical world. So, the weak stability criterion can be used to check if the planned motions are feasible, but it does not state if the contact is stable. An alternative idea is to judge if $(-\boldsymbol{f}_G, -\boldsymbol{\tau}_G)$ should be included in a proper subset of the polyhedral convex cone of the contact wrench. Then, the contact is likely to be stable within a margin, but there is no guarantee that it is stable within the entire polyhedral convex cone of the contact wrench. The idea is summarized as follows:

Theorem 3.2 (Weak Stability Criterion)**.** *If* $(-\boldsymbol{f}_G, -\boldsymbol{\tau}_G)$ *is an element of a proper subset of the polyhedral convex cone of the contact wrench given by Eqs.(3.21), (3.22), the contact is called sufficiently weakly stable to* $(\boldsymbol{f}_G, \boldsymbol{\tau}_G)$*.*

3.4 Conclusion

In the previous sections, three popular stability criteria, which received attention by many researchers worldwide, were presented. First, the well-known Zero Moment Point (ZMP) criterion, which is most often applied in humanoid robots, was fully presented. Dynamic stability of a humanoid robot is maintained, while the ZMP remains inside the convex hull created by the support foot/feet. The concept of ZMP has played and will continue to play an essential role in both theoretical considerations and practical development of humanoid robots and biped locomotion. Next, the Foot Rotation Indicator (FRI) point was briefly presented, providing a deeper insight into the humanoid robot stability and leading to the following major result: the farther away the FRI point is from the support boundary, the larger the unbalanced moment on the foot and the greater the instability. The distance between the FRI point and the nearest point on the polygon boundary is a useful indicator of the static stability margin of the foot. Finally, to cover research related to humanoid-human cooperation for handling ordinary tasks in human environments, the Contact Wrench Cone (CWC) universal stability criterion was briefly described. This criterion is oriented towards maintaining stability with multi-contact points on any terrain, possibly while carrying objects. The CWC criterion checks if the sum of the gravity and the inertia wrench applied to the CoM of a humanoid robot is inside the polyhedral convex cone of the contact wrench between the feet of the robot and its environment. This criterion can be used to determine the strong stability of the foot contact even when a robot walks on an arbitrary terrain, other than the horizontal plane and/or when the hands of the robot are in contact with the terrain, under the assumption that sufficient friction exists at the point of contact. The CWC criterion from a theoretical basis seems to be the strongest stability criterion and it is better for critical cases, i.e. at uneven terrain. In the case of humanoid locomotion on flat terrain, it is demonstrated from the last thirty years of related research, that the ZMP criterion is robust enough, theoretically and experimentally. This is the main reason for adopting the ZMP criterion in this thesis.

Chapter 4

Humanoid Robot Gait Generation

The worthwhile problems are the ones you can really solve or help solve, the ones you can really contribute something to. ... No problem is too small or too trivial if we can really do something about it.

Richard Phillips Feynman (1918–1988)

In this chapter, the humanoid robot gait generation task will be introduced and discussed. Over the years, various methods for handling such a task were proposed. One of them relies on motion without the use of active joints, that is only with joints that cannot be controlled directly, namely passive walking. In passive walking, the field of gravity is the force which actuates the robot, forcing it to walk. A passive walk starts with an impulse and the legged robot walks on a sloped terrain. Interesting results have been obtained, while researching this class of legged robots, the passive walkers, but all lead to the following conclusion: in passive walking, the gait cannot be controlled directly, meaning the legged robot cannot change its walking direction or stop walking. Obviously, such walkers are not suitable for human environments. Therefore, research turned into making use of legged robot's active joints. In this case, predefined walking patterns are introduced, followed, and most often corrected online, so that the legged robot walks stably in any direction, on any terrain, over any step length and performs tasks always within mechanical and physical limits. Therefore, this class of walkers was named active walkers, and active walking is still the core philosophy of every humanoid robot locomotion engine. These concepts are presented in Section 4.1. In Section 4.2, the notions of static

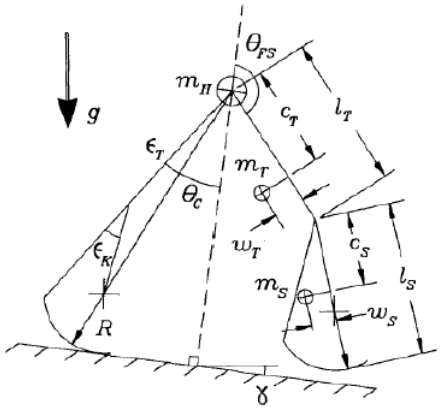

Figure 4.1: A chain of four rigid, inanimate links walking down an incline [5]

and dynamic gait are presented and contrasted, followed by concentrated-mass models in Section 4.3, where the robot dynamics are taken into account. Popular dynamic models suitable for pattern generation are the Inverted Pendulum model (Section 4.3.1) and the Cart and Table model (Section 4.3.2), where the ZMP criterion is used to determine the gait stability, while suitable human-like walking patterns can be obtained.

4.1 Passive and Active Walking

In this section, the ideas behind the passive and active walkers are briefly presented and discussed. We focus on active walking with concentrated-mass models, which lie at the core of this research, since they provide simplicity and effectiveness, without compromising gait stability.

4.1.1 Passive Walkers

In this section, an important class of humanoid robots is defined, based on their gait generation method, namely the *passive walkers*. The passive walking approach was first presented by McGeer [5]. He demonstrated that a four-bar mechanism in the shape of the skeleton of the lower half of the human body could walk under only the gravitational effect down a slight incline. This research showed that, as long as the lengths and masses of the various components of the mechanism were correctly tuned, a simple pendulum

motion is enough to produce human-like walking. Also, curved feet were used to complete the control by self-correcting for any mild perturbations. Some authors call it the *cyclic walking* and it could all be summarized into a single idea: "Let the gravity do the walking". Passive walking is in general implemented on underactuated mechanisms, where not all joints are directly controllable, and semi-underactuated legged robots.

This approach is also know as McGeer's theory of passive dynamic bipedal locomotion, where the gait is simply a "natural repetitive motion of a dynamical system". The advantage of this approach is that the legged robot consumes minimal energy and requires no computer control for walking on sloped terrains. The disadvantage, however, is that it is good for nothing else. Finally, it should be noted that passive walking can also work with legged robot having knees (see Figure 4.1), although it was originally introduced for legged robots having only straight legs. In such case a more natural human-like walking gait can be obtained by passive interaction of gravity and inertia. The physics are more or less the same as in straight-legged walking, but the knee-jointed form has two advantages. First, it offers a simple solution to the problem of foot clearance during the recovery phase. Second, in some cases it is more stable [5].

4.1.2 Active Walkers

In this section, we will define the *active walkers*, where we will consider all the humanoid robots, whose gait generation is based on the idea of active walking. Walking in this case is called active in the sense that each biped robot's joint is actuated by a motor and a motor-driver. A key difference from passive walking is that while swinging, the swing leg does not fall on the landing phase, therefore the impact force is considerably reduced, so the motion is more stable.

From our point of view, for generating active walking motion of biped robots, suitable patterns should be developed taking into account the dynamics and stability governing each step the robot takes. In order to generate such motion patterns, distributed-mass and concentrated-mass model can be used. The popular concentrated-mass models based on the Inverted Pendulum and the Cart and Table will be described in detail in the next sections, mainly because both models have been successfully used by researchers on many humanoid robots to obtain stable walking motion. The Inverted Pendulum model is based on the motion of a free ball within a plane in space, which follows the inverted pendulum

dynamics. Such behavior is observed in the human center of mass (CoM) motion during walking. The Cart and Table model deals with controlling the motion of a cart on a flat table by optimizing the cart's jerk. The behavior is similar to the Inverted Pendulum motion, but this method also reflects the dynamics of non-linearity during the act of walking.

In all cases, the ZMP reference pattern has been constructed a priori (see Appendix A). For the given ZMP reference trajectories, the dynamic models will try to track the walking pattern and output a smooth trajectory for the CoM of the robot. Next, the generated CoM trajectories along with the feet trajectory are fed to the inverse kinematics procedure, producing suitable joint trajectories which will achieve the predefined walking pattern.

Active walking motion is still an open area of research, and many proposals have been made, some of them only in simulation and others on real humanoid robot platforms of various sizes.

4.2 Static and Dynamic Gait

In this section, we underline the difference between the static and dynamic gait in the humanoid robot walking motion. In humanoid robotics literature, static gaits correspond to low-speed gaits, while dynamic gaits to human-like walking with higher speeds.

4.2.1 Static Gait

The idea behind the static gait is that the CoM must always lie inside the support foot, while the robot is taking a step (Figure 4.2). Therefore, the motion produced this way corresponds to low-speed walking. In static gait, the whole-body dynamics of the humanoid robot do not help the stability during the walking motion, because the CoM must always stay close to the middle of the support polygon defined by the foot/feet. During walking with a static gait, the range of motion of the CoM on the plane is increased considerably. In general, for static gaits on flat terrain, the CoM trajectory is almost the same as the ZMP trajectory. Graf et al. [39] presented such a walking framework, implemented on the NAO humanoid robot [18] using feedback from sensors to improve stability.

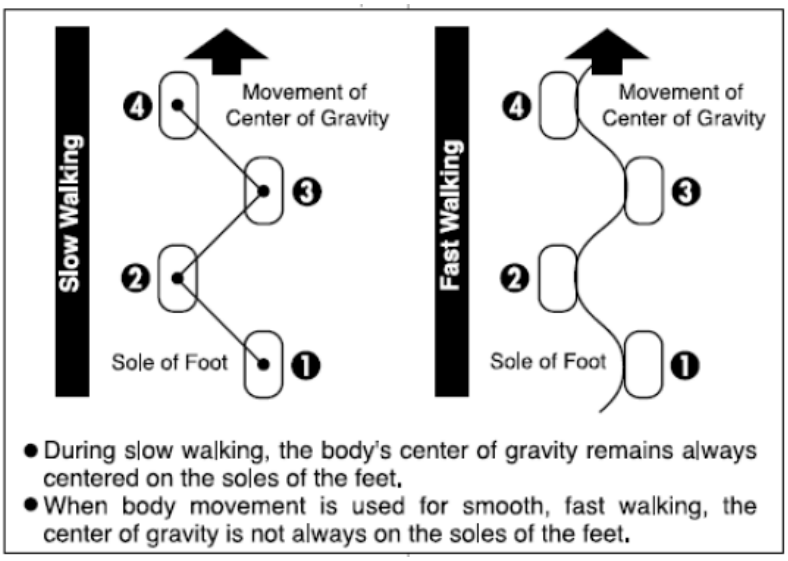

Figure 4.2: Static and Dynamic Gait examples [6]

4.2.2 Dynamic Gait

In the dynamic gait, although the CoM is not required to lie inside the support polygon while the robot is taking a step, the ZMP must be inside the support polygon to maintain the overall stability. This kind of gait is also called fast walking, since higher-speed gaits can be achieved compared to the static one.

When walking is faster, the CoM could be close or even outside the boundary of the support polygon, thus it is more human-like motion. Smooth patterns must be generated in order to reduce the inertial effects, which are higher in comparison to the static gait. Paradoxically, the inertial effects help to maintain biped stability during the steps, because the CoM acceleration relocates the ZMP closer to the middle of the support polygon. In this case, the CoM acceleration must be controlled, so that the ZMP does not overcome the support polygon. Some techniques reduce the jerk with high-order polynomials, but the disadvantage of high computation time does not allow real-time applications of gait generation, so the gait is pre-planned and the online biped control cancels external disturbances. Other techniques optimize the jerk, which reduces the computation time and it is possible to develop real-time walking patterns. The online control reduces external disturbances, such as gravity, terrain and structural imperfections of the mechanism. Dynamic walking algorithms has been successfully implemented

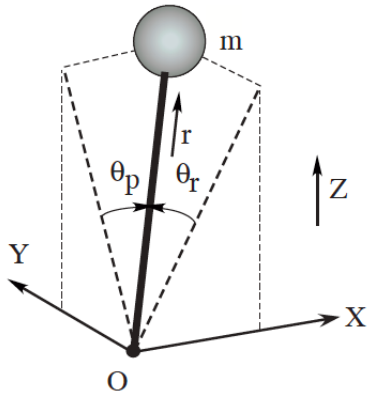

Figure 4.3: The 3D Inverted Pendulum model [7]

in the majority of humanoid robots today (i.e. ASIMO, HUBO, HRP-2, Nao), but it still remains an open subject for research. Many dynamic gait generation methods are based on simplified dynamic models, such as the Inverted Pendulum model [40] and the Cart and Table model [41], where it was observed that these models can generate smooth CoM patterns and a more natural motion for the humanoid robot.

4.3 Gait Dynamic Models

In this section, we will introduce dynamic models capable of producing stable walking patterns for robust dynamic locomotion. We will focus on the well-known concentrated-mass models, which simplify the whole body dynamics to a CoM motion by concentrating the robot mass into a single point. One may wonder: "Why not derive whole body dynamics and use them in obtaining accurate walking patterns and maybe also try to control those dynamics to reject disturbances encountered while walking?". The answer to such a question is of course the issues of model complexity and computational time. Nowadays, humanoid robot's embedded systems are quite limited and the operations mentioned in the above question cannot be carried out in real-time. Therefore, researchers turned to simplified dynamic models, which are suitable for onboard real-time planning and control.

4.3.1 Inverted Pendulum Model

When a humanoid robot is supporting its body on one leg, its dominant dynamics can be represented by a single inverted pendulum, which connects the supporting foot and the CoM of the robot, as proposed by Kajita et al. [7]. Such a model consists of a point mass and a massless telescopic leg, as shown in Figure 4.3. The position of the point mass is assumed to be at the CoM of the humanoid robot. Let that position be $\boldsymbol{p} = (c_x, c_y, c_z)$ in Cartesian coordinates, which can be alternatively specified by a state vector $\boldsymbol{q} = (\theta_r, \theta_p, r)$ in spherical coordinates

$$c_x = rS_p \tag{4.1}$$

$$c_y = -rS_r \tag{4.2}$$

$$c_z = rD \tag{4.3}$$

where $S_r = \sin\theta_r$, $S_p = \sin\theta_p$, and $D = \sqrt{1 - S_r^2 - S_p^2}$. Let (τ_r, τ_p, f) be the actuator torque and force associated with the state vector $\boldsymbol{q}$. With these inputs, the 3D Inverted pendulum dynamic model in Cartesian coordinates is given by

$$m \begin{bmatrix} \ddot{c}_x \\ \ddot{c}_y \\ \ddot{c}_z \end{bmatrix} = \left(\boldsymbol{J}^\top\right)^{-1} \begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{4.4}$$

where m is the point mass of the pendulum and g the acceleration of gravity. The structure of the Jacobian $\boldsymbol{J}$ is given by

$$\boldsymbol{J} = \frac{\partial p}{\partial q} = \begin{bmatrix} 0 & rC_p & S_p \\ -rC_r & 0 & -S_r \\ -\frac{rC_rS_r}{D} & -\frac{rC_pS_p}{D} & D \end{bmatrix} \tag{4.5}$$

with $C_r = \cos\theta_r$, $C_p = \cos\theta_p$. Multiplying by $\boldsymbol{J}^\top$ from the left in Eq.(4.4) gives

$$m \begin{bmatrix} 0 & -rC_r & -\frac{rC_rS_r}{D} \\ rC_p & 0 & -\frac{rC_pS_p}{D} \\ S_p & -S_r & D \end{bmatrix} \begin{bmatrix} \ddot{c}_x \\ \ddot{c}_y \\ \ddot{c}_z \end{bmatrix} = \begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} - mg \begin{bmatrix} -\frac{rC_rS_r}{D} \\ -\frac{rC_pS_p}{D} \\ D \end{bmatrix} \tag{4.6}$$

Using the first row of Eq.(4.6) with Eqs.(4.1),(4.2),(4.3), we can obtain a second-order differential equation for the motion along the y-axis

$$m(-c_z\ddot{c}_y + c_y\ddot{c}_z) = \frac{D}{C_r}\tau_r - mgc_y \tag{4.7}$$

and similarly for the x-axis

$$m\left(c_z\ddot{c}_x - c_x\ddot{c}_z\right) = \frac{D}{C_p}\tau_p + mgc_x \tag{4.8}$$

The dynamic model presented can be used in producing different kinds of motions. We will narrow these classes down by applying constraints, which simplify its dynamic model and limit the motion of the pendulum, making it suitable for walking. The first constraint limits the motion of the pendulum within a plane defined by the normal vector $(k_x, k_y, -1)$ intersecting the z-axis at h_c. Under this constraint, c_z is now given by

$$c_z = k_x c_x + k_y c_y + h_c \tag{4.9}$$

For a robot walking on uneven terrain, the normal vector should match the slope of the ground and c_z should be the expected average distance of the center of the robot's mass from the ground. We then use as constraint the second derivative of Eq.(4.9)

$$\ddot{c}_z = k_x\ddot{c}_x + k_y\ddot{c}_y \tag{4.10}$$

Substituting these constraints into Eqs.(4.7),(4.8), we obtain the dynamics of the pendulum under the constraints

$$\ddot{c}_x = \frac{g}{h_c}c_x + \frac{k_y}{h_c}\left(c_x\ddot{c}_y - \ddot{c}_x c_y\right) + \frac{1}{mh_c}u_p \tag{4.11}$$

$$\ddot{c}_y = \frac{g}{h_c}c_y - \frac{k_x}{h_c}\left(c_x\ddot{c}_y - \ddot{c}_x c_y\right) - \frac{1}{mh_c}u_r \tag{4.12}$$

where u_r, u_p are new virtual inputs, which are introduced to compensate for the input nonlinearity

$$\tau_r = \frac{C_r}{D}u_r \tag{4.13}$$

$$\tau_p = \frac{C_p}{D}u_p \tag{4.14}$$

Now consider the case of the robot walking on a flat terrain. In this case, we can define the horizontal constraint plane as $(k_x, k_y, -1) = (0, 0, -1)$ and we obtain

$$\ddot{c}_x = \frac{g}{h_c}c_x + \frac{1}{mh_c}u_p \tag{4.15}$$

$$\ddot{c}_y = \frac{g}{h_c}c_y - \frac{1}{mh_c}u_r \tag{4.16}$$

Next, consider the case of walking on a slope, where $k_x \neq 0$ and/or $k_y \neq 0$. Multiplying Eqs.(4.10),(4.11) by c_x, c_y respectively and adding them, we obtain

$$c_x \ddot{c_y} - \ddot{c_x} c_y = \frac{-1}{mc_z}(u_r c_x + u_p c_y) \tag{4.17}$$

Therefore, we can reduce the case of an inclined constraint plane to the dynamics of the flat plane, given by Eqs.(4.15),(4.16), by introducing the following additional constraint on the inputs

$$u_r c_x + u_p c_y = 0 \tag{4.18}$$

Eqs.(4.15),(4.16) are independent linear equations. The only parameter governing those dynamics is h_c. Note that despite the original nonlinear dynamics, the obtained constrained dynamics are linear without introducing any approximations. This model is called the Three-Dimensional Linear Inverted Pendulum Model (3D-LIPM). Kajita and Tani introduced a two-dimensional version of the dynamic model in 1991 [42].

Nature of the 3D-LIPM Next, we will examine the nature of trajectories generated by the 3D-LIPM with zero inputs ($u_r = u_p = 0$). In this case, the linear dynamics become

$$\ddot{c_x} = \frac{g}{h_c} c_x \tag{4.19}$$

$$\ddot{c_y} = \frac{g}{h_c} c_y \tag{4.20}$$

For any given initial condition, these equations determine a trajectory in the three-dimensional space under the influence of gravity, as shown in Figure 4.4, obtained by solving Eqs.(4.19),(4.20).

Geometry of the 3D-LIPM Trajectory Motion along the x and y axes is governed by Eqs.(4.19),(4.20) respectively. Integrating each equation, we obtain a time-invariant parameter, named the *orbital energy* E [42]

$$E_x = -\frac{g}{2h_c} c_x^2 + \frac{1}{2}\dot{c_x}^2 \tag{4.21}$$

$$E_y = -\frac{g}{2h_c} c_y^2 + \frac{1}{2}\dot{c_y}^2 \tag{4.22}$$

The orbital energy E' on a rotated-by-θ coordinate plane is obtained as follows

$$E'_x = -\frac{g}{2h_c}(C_\theta c_x + S_\theta c_y)^2 + \frac{1}{2}(C_\theta \dot{c}_x + S_\theta \dot{c}_y)^2 \tag{4.23}$$

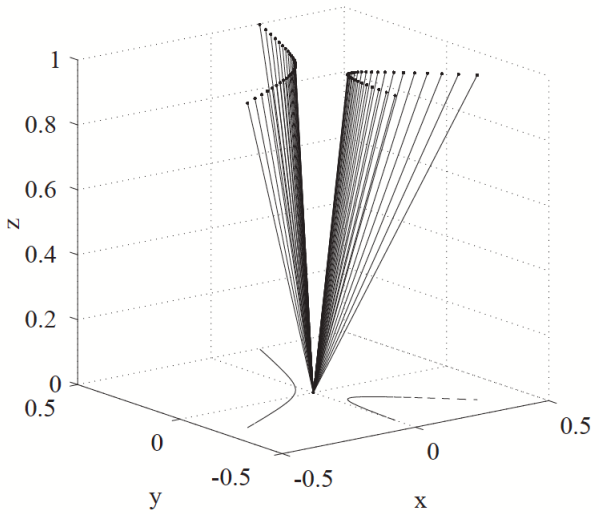

Figure 4.4: Free response of the 3D-LIPM [7]

$$E'_y = -\frac{g}{2h_c}(-S_\theta c_x + C_\theta c_y)^2 + \frac{1}{2}(-S_\theta \dot{c}_y + C_\theta \dot{c}_y)^2 \tag{4.24}$$

The total energy of the system does not change while rotating the coordinate frames

$$E'_x + E'_y = E_x + E_y = const \tag{4.25}$$

When the rotated y-axis coincides with the axis of symmetry of the motion projected on the xy plane, E'_x and E'_y become maximum and minimum respectively. Therefore, we can compute the axis of symmetry by solving the variation of the orbital energy with respect to the rotation angle θ

$$\frac{\partial E'_x}{\partial \theta} = A(S_\theta^2 - C_\theta^2) + BS_\theta C_\theta = 0 \tag{4.26}$$

where

$$A = \frac{g}{h_c} c_x c_y - \dot{c}_x \dot{c}_y \tag{4.27}$$

$$B = \frac{g}{h_c}(c_x^2 - c_y^2) - (\dot{c}_x^2 - \dot{c}_y^2) \tag{4.28}$$

When $B \neq 0$, the solution is

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2A}{B}\right) \tag{4.29}$$

whereas, when $A = B = 0$, the solution becomes

$$\theta = atan2(c_y, c_x) \tag{4.30}$$

and, finally, when $A \neq 0, B = 0$, the solution is

$$\theta = \frac{\pi}{4} \tag{4.31}$$

If the y-axis already coincides with the axis of symmetry, θ must be zero. Ignoring the case $B = 0$, which rarely happens, Eq.(4.29) implies that A must be zero as well and, therefore, Eq.(4.27) leads to the following condition

$$\left(\frac{g}{h_c}\right) c_x c_y - \dot{c}_x \dot{c}_y = 0 \tag{4.32}$$

Using this equation, we can calculate the geometric shape of the 3D-LIPM, by substituting Eqs.(4.21),(4.22) in Eq.(4.32) to obtain the final quadratic form

$$\frac{g}{2h_c E_x} c_x^2 + \frac{g}{2h_c E_y} c_y^2 + 1 = 0 \tag{4.33}$$

Since $E_x > 0$ and $E_y < 0$, Eq.(4.33) forms a hyperbolic curve.

Controllable Dynamic Model Next, we will try to control the motion of the 3D-LIPM, assuming walking over a flat plane, as presented by Czarnetzki et al. [43]. In this case, the motion is constrained to the horizontal plane, as explained above. The dynamics governing such a motion are given by Eqs.(4.15),(4.16). Let's focus on the motion along the x-axis. According to this model, the position z_x of the ZMP on the ground can be easily derived as

$$z_x(t) = -\frac{\tau_r(t)}{mg} \tag{4.34}$$

Substituting Eq.(4.34) into Eq.(4.15), we can obtain the following equation, called the ZMP equation

$$z_x(t) = c_x(t) - \frac{h_c}{g} \ddot{c}_x(t) \tag{4.35}$$

It is obvious that, for a constant CoM height h_c, the ZMP position depends on the position and the acceleration of the CoM in that axis. The same procedure can be followed for the y-axis component, which can be handled separately. The simplified dynamic model

can be used for planning and controlling stable walking. Unfortunately, the dynamic model presented, does not take into account the support polygon and it may result in a FZMP, or an equivalent FRI point, as defined in Chapter 3, which is an indication of an unbalanced momentum resulting in gait instability. Therefore, the need of effective control to maintain dynamic stability is mandatory. As seen before, the movement of a humanoid robot to track a reference ZMP trajectory is reduced to obtaining a suitable CoM pattern from the dynamic model given by Eq.(4.35). The ZMP position z_x is both the output of the dynamic model and the target of the control procedure. Therefore, a suitable state vector is the following

$$\mathbf{x}(t) = \begin{bmatrix} c_x(t) \\ \dot{c}_x(t) \\ z_x(t) \end{bmatrix} \tag{4.36}$$

Then, the dynamic model in Eq.(4.35) can be represented in its equivalent state-space form

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{h_c} & 0 & -\frac{g}{h_c} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_x(t) \tag{4.37}$$

$$z_x(t) = \begin{bmatrix} 1 & 0 & -\frac{h_c}{g} \end{bmatrix} \mathbf{x}(t) \tag{4.38}$$

where $v_x(t) = \dot{z_x}(t)$ and can be considered as the change in ZMP z_x according to a reference ZMP z_x^{ref}. Discretizing Eqs.(4.37),(4.38) with sampling period T_s and with state vector

$$\mathbf{x}(k) = \begin{bmatrix} c_x(kTs) \\ \dot{c}_x(kTs) \\ z_x(kTs) \end{bmatrix} \tag{4.39}$$

the discrete-time state space will be given by

$$\mathbf{x}(k+1) = \overbrace{\begin{bmatrix} 1 & T_s & 0 \\ \frac{g}{h_c}T_s & 1 & -\frac{g}{h_c}T_s \\ 0 & 0 & 1 \end{bmatrix}}^{\mathbf{A}} \mathbf{x}(k) + \overbrace{\begin{bmatrix} 0 \\ 0 \\ T_s \end{bmatrix}}^{\mathbf{b}} v_x(k) \tag{4.40}$$

$$z_x(k) = \overbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}^{\mathbf{c}^\top} \mathbf{x}(k) \tag{4.41}$$

4.3.2 Cart and Table Model

The dynamics of the robot can be modelled as a point-mass cart on a massless table, a model proposed by Kajita et al. [41], known as the Cart and Table model (Figure 4.5). This model gives a convenient representation for stability, as the ZMP must always lie inside the table's base and the cart must accelerate with a proper rate to keep the table upright. The ZMP exists inside the table base and since the moment around the ZMP must be zero, we can obtain

$$\tau_x(t) = mg\left(c_x(t) - z_x(t)\right) - m\ddot{c}_x(t)h_c = 0 \tag{4.42}$$

leading to

$$z_x(t) = c_x(t) - \frac{h_c}{g}\ddot{c}_x(t) \tag{4.43}$$

Defining a new variable $u_x(t)$ as the jerk of the cart (time derivative of the acceleration)

$$\dddot{c}_x(t) = u_x(t) \tag{4.44}$$

and assuming $u_x(t)$ as the input of Eq.(4.42) and state vector

$$\mathbf{x}(t) = \begin{bmatrix} c_x(t) \\ \dot{c}_x(t) \\ \ddot{c}_x(t) \end{bmatrix} \tag{4.45}$$

we can translate the second-order differential equation governing the Cart and Table system into a state-space representation

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x(t) \tag{4.46}$$

$$z_x(t) = \begin{bmatrix} 1 & 0 & -\frac{h_c}{g} \end{bmatrix} \mathbf{x}(t) \tag{4.47}$$

We can generate walking patterns by treating the ZMP control as a servomechanism control problem. Let $c_x(t)$ be the position of the CoM (cart) at time t along axis x, expressed at an arbitrary inertial frame. At discrete intervals (assuming a sampling period T_s), we can define the discrete-time state $\mathbf{x}(k)$, which includes the position, the velocity, and the acceleration of the cart at discrete time k

$$\mathbf{x}(k) = \begin{bmatrix} c_x(kT_s) \\ \dot{c}_x(kT_s) \\ \ddot{c}_x(kT_s) \end{bmatrix} \tag{4.48}$$

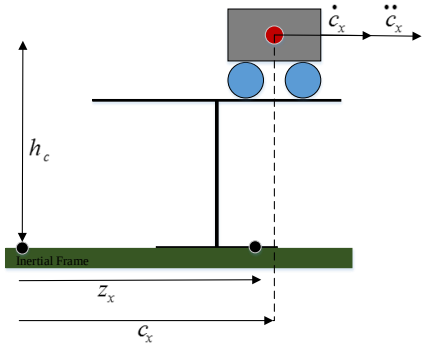

Figure 4.5: The Cart and Table Model

Assuming stability, the mass is constrained on the upright table at height h_c. If $u_x(k) = \dddot{c}_x(kT_s)$ is the control input (jerk of the cart) at discrete time k, then the state-space system dynamics are formed as

$$\mathbf{x}(k+1) = \overbrace{\begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}}^{\mathbf{A}} \mathbf{x}(k) + \overbrace{\begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}}^{\mathbf{b}} u_x(k) \tag{4.49}$$

The ZMP of the system under gravitational acceleration g is

$$z_x(k) = \overbrace{\begin{bmatrix} 1 & 0 & -h_c/g \end{bmatrix}}^{\mathbf{c}^\top} \mathbf{x}(k) \tag{4.50}$$

Assuming the motion is not coupled on the two axes, this one-dimensional model can express the dynamics on the x and y axes independently.

4.3.3 3D-LIPM and Cart and Table Model Comparison

In this section, we will compare the two models presented for the task of walking pattern generation. It is true that the 3D-LIPM simplifies the CoM motion obtained, but there is no direct relationship to the ZMP reference trajectory. This problem introduces a discontinuity at the moment where the gait changes from the SS to the DS phase. The discontinuity created is more observable in high velocities, where the CoM's jerk is an important factor. This situation could be improved by using high-order splines. On the contrary, in the Cart and Table model the jerk is usually optimized within some

performance index, resulting in smoother actions maintaining the continuity no matter the change of the phase, resulting in higher-speed gaits. Furthermore, there is a unique relationship between the ZMP reference trajectory corresponding to a CoM trajectory. In addition, as stated by Arbulú [44], the non-linear dynamics of the cart help in realizing a better ZMP motion than the 3D-LIPM does.

For the walking pattern generation task with simplified dynamic models, a few solutions have been proposed. Based on the Preview Control introduced by Katayama et al. [45] in 1985 and established by Sheridan [46], Kajita et al. [41, 47] proposed the ZMP Preview Control, which will be used in this research. There are other solutions by Choi et al. [48] in 2004 and Kim et al. [49] in 2007. The latter reduces considerably the error of the tracked ZMP compared to the previous two solutions. Therefore, at high velocities it is a more stable walking pattern generator. In this research, the Cart and Table model will be used confirming the advantages over the 3D-LIPM in practice.

4.4 Conclusion

In this chapter, a classification of humanoid robots based on their locomotion style was given. The concept of passive walking was first introduced, where it was highlighted that there is no active control on humanoid robot joints and only gravity is responsible for the locomotion. The advantage of this approach is that the legged robot consumes minimal energy and requires no computer control for walking on slope terrains. The disadvantage, however, is that this approach is not good for anything else. Next, the concept of active walking was presented, where each biped robot's joint is actuated by a motor and a motor-driver. This concept seems to be more suitable to generate motion patterns for robots working in human environments. Furthermore, the active gait can be categorized into static and dynamic gait. In static gaits, the biped robot's CoM must lie always inside the support foot, while the robot is taking a step, yielding low-speed gaits. In dynamic gaits, there is no such restriction, but, to maintain stability, the ZMP must be inside the support polygon. This is also called fast walking, since higher speed gaits can be achieved compared to the static one. Furthermore, the need for simplified dynamic models was highlighted. Two well-known concentrated-mass models, which have been tested on some biped and humanoid robots, have been detailed in this chapter. The 3-Dimension Linear Inverted Pendulum Model (3D-LIPM) and the Cart and Table model. These simplified

dynamic models allow online computations in real-time applications, effectively modeling the humanoid robot dynamics and generating walking patterns. Finally, these models were compared, concluding to the Cart and Table model being more suitable for humanoid robot locomotion.

Chapter 5

Control and Estimation Schemes for Locomotion

If I have seen further it is by standing on the shoulders of giants.
Isaac Newton (1642–1727)

In the field of control theory there are both traditional and modern techniques for regulating dynamic systems. In the first category, two main approaches were well studied in the past: the frequency-response techniques and the root-locus technique, both based on the transfer function of the system. [50]. Both methods are very effective, but are largely based on trial and error and require a significant amount of experience. Even when an acceptable design is completed, the question remains as to whether a "better" design could be found. In the second category, design techniques using the so-called state-space of the dynamic systems are employed [51]. The pole-placement technique, a modern design technique, is based on the state-space model of the plant, rather than on the transfer function, as required by the traditional methods. In this technique, we assume that we know the exact locations required for the closed-loop transfer-function poles and we can realize these locations, at least in linear models. Of course, for a dynamic system, the regions in which the pole locations can be placed are limited.

In the pole-placement technique, we assume that we know the poles locations that yield the "best" control system. In this chapter, we develop a different technique that yields the "best" control system for a linear system. This technique is an optimal design

technique and assumes that we can write a mathematical function, which is called the cost function or the performance index, which leads to "best" control if minimized, hence the term optimal. However, in most cases, the choice of the cost function involves some trial and error; that is, we are not sure of the exact form the cost function should take.

In this chapter, first the Linear Quadratic Regulator (LQR) is briefly presented (Section 5.1), followed by the Luenberger observer for estimating the state vector in Section 5.2, as proposed by Luenberger [52]. In Section 5.3 the Preview Control is comprehensively covered, as proposed by Katayama et al. [45]. The Preview Control is then applied to both the 3D-LIPM and the Cart and Table model, augmented with an observer in Sections 5.4 and 5.5 respectively. After confirming that the Cart and Table model is more suitable for humanoid robot locomotion, compared to the 3D-LIPM, it will be fully adopted throughout this research.

Next, two novel control scheme are presented. In Section 5.5.3, the Preview Controller equipped with the Auxiliary ZMP for the Cart and Table model is presented, as a way of adapting online the generated walking patterns in the presence of disturbances and/or uneven terrain. Furthermore, in Section 5.6, a novel Linear Model Predictive Control (LMPC) framework for Linear Time-Invariant (LTI) dynamic systems is presented and extended also to Multiple-Input Multiple-Output (MIMO) systems. This control scheme aims at reducing significantly the dimension of the optimization problem, resulting in faster solutions. In Section 5.7, the novel LMPC scheme is adopted for the walking pattern generator task with the Cart and Table model. Furthermore, exploiting the dimensionality reduction of the optimization problem, constraints on the ZMP are employed for improving the gait stability while walking. This problem is formulated as a Quadratic Program (QP) with linear inequality constraints, which can be solved online for real-time fast controlling schemes due to the dimensionality reduction claimed.

5.1 Linear Quadratic Regulator

In this section, an optimal control for Linear Time-Invariant (LTI) systems is derived, adopting the Linear Quadratic Regulator (LQR) method for discrete-time systems [51]. Given a discrete-time state space

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{5.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \tag{5.2}$$

where $k = kT_s$, $k = 0, 1, \ldots$, T_s is the sampling time constant, $\mathbf{x}(k) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(k) \in \mathbb{R}^r$ is control vector, and $\mathbf{y}(k) \in \mathbb{R}^p$ is the output vector to be controlled. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are constant matrices with dimension $n \times n$, $n \times r$, and $p \times n$ respectively.

The infinite horizon LQR is computed by optimizing the additive quadratic cost function. In the discrete-time case, the problem is formulated as

$$J = \sum_{i=0}^{\infty} \mathbf{x}(i)^\top \mathbf{Q}\mathbf{x}(i) + \mathbf{u}(i)^\top \mathbf{R}\mathbf{u}(i) \tag{5.3}$$

Note that the matrix $\mathbf{Q}$ is a $n \times n$ positive semidefinite matrix and the matrix $\mathbf{R}$ is a $r \times r$ positive definite matrix. The optimal control is given by the following theorem.

Theorem 5.1 (Optimal LQR Control)**.** *The optimal control minimizing the cost given in Eq.(5.3) for the system dynamics in Eq.(5.1) is*

$$\mathbf{u}(k) = -\mathbf{K}_{\text{LQR}}\mathbf{x}(k) \tag{5.4}$$

where K_{LQR} is the LQR gain matrix

$$\mathbf{K}_{\text{LQR}} = \left[\mathbf{R} + \mathbf{B}\mathbf{P}\mathbf{B}^\top\right]^{-1} \mathbf{B}^\top \mathbf{P}\mathbf{A} \tag{5.5}$$

with $\mathbf{P}$ being the solution of the algebraic Riccati equation (ARE)

$$\mathbf{P} = \mathbf{A}^\top \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{A}^\top \mathbf{P}\mathbf{B}\left[\mathbf{B}^\top \mathbf{P}\mathbf{B} + \mathbf{R}\right]^{-1} \mathbf{B}^\top \mathbf{P}\mathbf{A} \tag{5.6}$$

For such stabilizing gain to exist, allowing the dynamic system to be controlled, the following condition, stated as theorem below, must be satisfied.

Theorem 5.2 (Controllability)**.** *The controllability matrix, $\mathbb{C}_n$, of the system or equivalently of the pair $(\mathbf{A}, \mathbf{B})$ must be full rank. The controllability matrix is given by*

$$\mathbb{C}_n = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \ldots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \tag{5.7}$$

However, when the matrix $\mathbb{C}_n$ is not full rank, meaning the system has state variables that cannot be controlled, it can still be stabilized, if the pair $(\mathbf{A}, \mathbf{B})$ is stabilizable, e.g. the uncontrollable states are stable.

5.2 Luenberger State Observer

The main requirement for obtaining an LQR solution is that all the states are available for feedback, that is, our system is fully observable and also our sensors do not introduce noise in the readings. In the real world, these two conditions are not always true. To deal with these issues, we introduce the Luenberger observer for estimating the state variables vector in an optimal way, originally proposed by Lunberger et al. [52]. It is proven that, under some hypothesis about the system and the sensors noise, the optimal observer (or optimal estimator) is a linear dynamic system with a specific structure given by

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \tag{5.8}$$

where $\hat{\mathbf{x}}$ denotes the state estimate, obtained by the observer, and $\mathbf{L}$ is the observer gain and has dimensions $n \times p$. As it can be easily seen, an observer comprises a real-time simulation of the system, driven by the same input as the system and by a correction term derived from the difference between the actual output of the system and the predicted output derived from the observer.

Let the error vector be $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$. Given this definition, the dynamics of the error are determined by the following error model:

$$\begin{aligned}
\tilde{\mathbf{x}}(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \\
&= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) - \mathbf{A}\hat{\mathbf{x}}(k) - \mathbf{B}\mathbf{u}(k) - \mathbf{L}(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)) \\
&= \mathbf{A}(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) - \mathbf{L}(\mathbf{C}\mathbf{x}(k) - \mathbf{C}\hat{\mathbf{x}}(k)) \\
&= (\mathbf{A} - \mathbf{L}\mathbf{C})\,\tilde{\mathbf{x}}(k)
\end{aligned}$$

Hence, the error dynamics have a characteristic equation given by

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{L}\mathbf{C})\,| = 0 \tag{5.9}$$

In general, $\tilde{\mathbf{x}}(0) \neq 0$, so we have to find a suitable observer gain $\mathbf{L}$, which makes $\tilde{\mathbf{x}} \to 0$ as $k \to \infty$ for any initial condition $\tilde{\mathbf{x}}(0)$. This can realized using the LQR method, essentially solving an estimation problem as the dual of a control problem. When obtaining a suitable $\mathbf{L}$, if $\tilde{\mathbf{x}} \to 0$ as $k \to \infty$, then $\hat{\mathbf{x}}(k) \to \mathbf{x}(k)$ as $k \to \infty$, that is, the state estimates eventually converge to their actual values. A key point here is that the estimation error does not depend on what the control inputs are, allowing us to solve the estimation problem independently from the control problem, a result referred to as the *separation theorem*.

A condition for such a suitable $\mathbf{L}$ to exist is that the system must be fully observable or, equivalently, that the pair $(\mathbf{C}, \mathbf{A})$ is observable.

Theorem 5.3 (Observability)**.** *A system is fully observable when the observability matrix $\mathbb{O}_n$ is full rank, where $\mathbb{O}_n$ is given by*

$$\mathbb{O}_n = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{5.10}$$

When the pair $(\mathbf{C}, \mathbf{A})$ is not observable, there exist unobservable state variables. However, if the pair $(\mathbf{C}, \mathbf{A})$ is detectable, e.g. all the unobserved states are stable, a suitable $\mathbf{L}$ can still be obtained.

5.3 Linear Preview Control

In many practical control system designs, it is required that the outputs or the state variables track without steady-state error the reference signals in presence of unmeasured disturbances. Conventional methods fail, when we require the state variables or the outputs to have a desired transient response, as in our case, where we want the ZMP trajectory to track the reference one, while the CoM trajectory has already started changing. Therefore, we have to adapt our control strategy to take into account this previewable demand. Katayama et al. [45] proposed an optimal controller subject to such a previewable demand.

Consider the discrete time state-space

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{E}\mathbf{w}(k) \tag{5.11}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \tag{5.12}$$

where $\mathbf{w}(k) \in \mathbb{R}^q$ is the inaccessible constant disturbance. The matrix $\mathbf{E}$ is a constant matrix with dimension $n \times q$. It is assumed that the rank of $\mathbf{B}$ is r, the rank of $\mathbf{C}$ is p, and the rank of $\mathbf{E}$ is q. Let $\mathbf{y}_{\text{ref}}(k)$ be the $p \times 1$ reference output vector, for which we assume that there exists a constant vector $\bar{\mathbf{y}}_{\text{ref}}$, such that

$$\lim_{k \to \infty} \mathbf{y}_{\text{ref}}(k) = \bar{\mathbf{y}}_{\text{ref}} \tag{5.13}$$

This implies that the reference vector is an arbitrary time-varying function, except that it reaches a steady state. We will further assume that the reference is previewable in the sense that at each discrete time k, N_p future values $\mathbf{y}_{\text{ref}}(k+1), \ldots, \mathbf{y}_{\text{ref}}(k+N_p)$ as well as the present and past values of the reference are available. The future values of the desired output beyond discrete time $k+N_p$ are approximated by $\mathbf{y}_{\text{ref}}(k+N_p)$, namely

$$\mathbf{y}_{\text{ref}}(k+i) = \mathbf{y}_{\text{ref}}(k+N_p), \quad i = N_p+1, \ldots \tag{5.14}$$

We seek a controller which satisfies the following

- In the steady state, the output $\mathbf{y}(k)$ tracks the reference vector $\mathbf{y}_{\text{ref}}(k)$ in the presence of disturbance $\mathbf{w}(k)$.
- The obtained closed-loop system is asymptotically stable and exhibits acceptable transient responses.

In order to satisfy those requirements, it is desired to introduce integrators to eliminate the tracking error $\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{y}_{\text{ref}}(k)$. Therefore, we must design a type one servomechanism for the dynamic system described by Eqs.(5.11),(5.12), such that the asymptotic regulation occurs, that is, $\mathbf{e}(k) \to 0$ as $k \to \infty$, while keeping the transient responses satisfactory in some sense. To this end, we employ the Linear Quadratic Integral (LQI) technique.

Let the incremental state vector be $\mathbf{\Delta x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$ and the incremental control vector be $\mathbf{\Delta u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$. It is well known that the integral action of the controller is introduced by including the incremental control in the performance index. Therefore, we seek the optimal controller $\mathbf{u}(k)$, such that the performance index

$$J_{\text{p}} = \sum_{i=k}^{\infty} \mathbf{e}(i)^{\top}\mathbf{Q}_e\mathbf{e}(i) + \mathbf{\Delta x}(i)^{\top}\mathbf{Q}_{\mathbf{x}}\mathbf{\Delta x}(i) + \mathbf{\Delta u}(i)^{\top}\mathbf{R}\mathbf{\Delta u}(i) \tag{5.15}$$

is minimized at each time k, where $\mathbf{Q}_e$ and $\mathbf{R}$ are $p \times p$ and $r \times r$ symmetric positive definite matrices respectively, $\mathbf{Q}_{\mathbf{x}}$ is an $n \times n$ symmetric non-negative definite matrix, and i denotes the dummy time index. The term $\mathbf{e}(i)^{\top}\mathbf{Q}_e\mathbf{e}(i)$ stands for the loss due to tracking error, and the terms $\mathbf{\Delta x}(i)^{\top}\mathbf{Q}_{\mathbf{x}}\mathbf{\Delta x}(i)$ and $\mathbf{\Delta u}(i)^{\top}\mathbf{R}\mathbf{\Delta u}(i)$ represent loss due to incremental state and control vectors respectively. Therefore, we can conclude that the physical interpretation of J_p is to achieve the asymptotic regulation without excessive rate

of change in the state and control vectors. The quadratic term for the rate of change in the state vector makes the design technique more flexible allowing us to directly regulate the transient responses of the state variables.

5.3.1 Optimal Preview Control Derivation

To derive the preview control, we need to make use of an augmented state-space that includes the future information of the reference signal as well as the error $\mathbf{e}(i)$, the incremental state vector $\mathbf{\Delta x}(i)$, and the incremental control vector $\mathbf{\Delta u}(i)$. From Eq.(5.11), the incremental state can be formulated as

$$\mathbf{\Delta x}(i+1) = \mathbf{A\Delta x}(i) + \mathbf{B\Delta u}(i), \quad i = k, k+1, \ldots \tag{5.16}$$

where we note that the incremental disturbance $\mathbf{\Delta w}(i)$ does not appear, because of the assumption that the disturbance is a step function. Also, we can notice from Eq.(5.12) and Eq.(5.16) that the tracking error satisfies

$$\mathbf{e}(i+1) = \mathbf{e}(i) + \mathbf{CA\Delta x}(i) + \mathbf{CB\Delta u}(i) - \mathbf{\Delta y}_{\text{ref}}(i+1), \quad i = k, k+1, \ldots \tag{5.17}$$

where the incremental reference vector is defined as

$$\mathbf{\Delta y}_{\text{ref}}(i) = \mathbf{y}_{\text{ref}}(i) - \mathbf{y}_{\text{ref}}(i-1) \tag{5.18}$$

Combining Eqs.(5.16),(5.17) into a single state-space representation we obtain

$$\begin{bmatrix} \mathbf{e}(i+1) \\ \mathbf{\Delta x}(i+1) \end{bmatrix} \begin{bmatrix} \mathbf{I}_p & \mathbf{CA} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{e}(i) \\ \mathbf{\Delta x}(i) \end{bmatrix} + \begin{bmatrix} \mathbf{CB} \\ \mathbf{B} \end{bmatrix} \mathbf{\Delta u}(i) + \begin{bmatrix} -\mathbf{I}_p \\ 0 \end{bmatrix} \mathbf{\Delta y}_{\text{ref}}(i+1) \tag{5.19}$$

where $i = k, k+1, \ldots$, and $\mathbf{I}_p$ denotes the $p \times p$ unit matrix.

Since N_p future references $\mathbf{y}_{\text{ref}}(i)$, $i = k+1, \ldots, k+N_p$ are available at time k, the relevant information on the incremental reference can be summarized as the $pN_p \times 1$ vector

$$\mathbf{x}_{\text{ref}}(k) = \left[\mathbf{\Delta y}_{\text{ref}}(k+1)^\top, \ldots, \mathbf{\Delta y}_{\text{ref}}(k+N_p)^\top\right]^\top \tag{5.20}$$

It follows from the assumption given in Eq.(5.14) that $\mathbf{x}_{\text{ref}}(i)$ satisfies

$$\mathbf{x}_{\text{ref}}(i+1) = \mathbf{A}_{\text{ref}}\mathbf{x}_{\text{ref}}(i), \quad i = k, k+1, \ldots \tag{5.21}$$

where $\mathbf{A}_{\text{ref}}$ is

$$\mathbf{A}_{\text{ref}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_p & & \mathbf{0} \\ & \mathbf{0} & \ddots & \mathbf{0} \\ & & \ddots & \mathbf{I}_p \\ \mathbf{0} & & & \mathbf{0} \end{bmatrix} \tag{5.22}$$

Now, we can define the $(p + n + pN_p) \times 1$ augmented state vector

$$\tilde{\mathbf{x}}(i) = \begin{bmatrix} \mathbf{e}(i)^\top & \boldsymbol{\Delta}\mathbf{x}(i)^\top & \mathbf{x}_{\text{ref}}(i)^\top \end{bmatrix}^\top \tag{5.23}$$

with dynamics

$$\tilde{\mathbf{x}}(i+1) = \left[\begin{array}{cc|ccc} \mathbf{I}_p & \mathbf{CA} & -\mathbf{I}_p & \mathbf{0} & \dots \; \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{0} & \dots \; \mathbf{0} \\ \hline \multicolumn{2}{c|}{\mathbf{0}} & \multicolumn{3}{c}{\mathbf{A_{ref}}} \end{array}\right] \tilde{\mathbf{x}}(i) \tag{5.24}$$

Changing variable in the performance index, given by Eq.(5.15), we can obtain the following performance index J_p, expressed in terms of the augmented state vector $\tilde{\mathbf{x}}$

$$J_p = \sum_{i=k}^{\infty} \tilde{\mathbf{x}}^\top(i) \begin{bmatrix} \mathbf{Q}_e & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tilde{\mathbf{x}}(i) + \boldsymbol{\Delta}\mathbf{u}(i)^\top \mathbf{R} \boldsymbol{\Delta}\mathbf{u}(i) \tag{5.25}$$

Therefore, the optimal controller can by derived by solving the LQR problem, given by Eq.(5.25), subject to the dynamic constraint, given by Eq.(5.24). For simplicity in the derivation, we will define the following matrices

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{CB} \\ \mathbf{B} \end{bmatrix}, \tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0} \end{bmatrix}, \tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{A} \end{bmatrix}, \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_e & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_x \end{bmatrix}, \tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{I}} & \tilde{\mathbf{F}} \end{bmatrix} \tag{5.26}$$

Katayama et al. [45] solved this problem, stating the following theorem.

Theorem 5.4. *The optimal control obtained by minimizing the cost given in Eq.(5.25) is*

$$\mathbf{u}^o(k) = -\mathbf{G}_I \mathbf{e}(k) - \mathbf{G}_x \mathbf{x}(k) - \sum_{l=1}^{N_p} \mathbf{G}_p \mathbf{y}_{\text{ref}}(k+l) \tag{5.27}$$

where

$$\mathbf{G}_I = \left[\mathbf{R} + \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{I}} \tag{5.28}$$

$$\mathbf{G}_x = \left[\mathbf{R} + \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{F}} \tag{5.29}$$

$$\mathbf{G}_d(1) = -\mathbf{G}_I \tag{5.30}$$

$$\mathbf{G}_d(l) = \left[\mathbf{R} + \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{X}}(l-1), \quad l = 2, \ldots, N_p \tag{5.31}$$

and where the $(p+n) \times (p+n)$ *matrix* $\tilde{\mathbf{K}}$ *is the non-negative definite solution of the ARE*

$$\tilde{\mathbf{K}} = \tilde{\mathbf{A}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{A}} - \tilde{\mathbf{A}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}} \left[\mathbf{R} + \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{A}} + \tilde{\mathbf{Q}} \tag{5.32}$$

Furthermore, the $(p+n) \times p$ *matrices* $\tilde{\mathbf{X}}(l)$ *are given by*

$$\tilde{\mathbf{X}}(l) = \tilde{\mathbf{A}}_c^\top \tilde{\mathbf{X}}(l-1), \quad l = 2, \ldots, N_p \tag{5.33}$$

$$\tilde{\mathbf{X}}(1) = -\tilde{\mathbf{A}}_c^\top \tilde{\mathbf{K}} \tilde{\mathbf{I}} \tag{5.34}$$

where the $\tilde{\mathbf{A}}_c$ *is the closed-loop matrix defined by*

$$\tilde{\mathbf{A}}_c = \tilde{\mathbf{A}} - \tilde{\mathbf{B}} \left[\mathbf{R} + \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{K}} \tilde{\mathbf{A}} \tag{5.35}$$

It should be noted that the optimal controller $\mathbf{u}^o(k)$ of Eq.(5.27) consists of three terms; the first term represents the integral action of the tracking error, the second term represents the state feedback, and the third term is the feedforward or preview action based on the local future information on the reference vector. If we set $N_p = 0$, then the preview action in Eq.(5.27) disappears and the optimal control becomes

$$\mathbf{u}^o(k) = -\mathbf{G}_i \sum_{i=0}^{k} \mathbf{e}(i) - \mathbf{G}_x \mathbf{x}(k) \tag{5.36}$$

Moreover, when $N_p = 1$ and because of $\mathbf{G}_d(1) = -\mathbf{G}_i$, the optimal control becomes

$$\mathbf{u}^o(k) = -\mathbf{G}_i \sum_{i=0}^{k} [\mathbf{y}(i) - \mathbf{y}_{\text{ref}}(i+1)] - \mathbf{G}_x \mathbf{x}(k) \tag{5.37}$$

This is a state feedback controller with integral action and feedforward actions.

Let $\mathbf{v}(k)$ be the discrete integral of tracking error, $\mathbf{e}(k)$

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{e}(k) \tag{5.38}$$

$$= \frac{z}{z-1} \mathbf{e}(k) \tag{5.39}$$

Thus, it follows from Eqs.(5.27),(5.38) that the optimal controller is

$$\mathbf{u}^o(k) = -\mathbf{G}_i\mathbf{v}(k) - \mathbf{G}_x\mathbf{x}(k) - \sum_{l=1}^{N_p} \mathbf{G}_d(l)\mathbf{y}_{\text{ref}}(k+l) \tag{5.40}$$

Using Eqs.(5.11),(5.12),(5.38), we can obtain

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \mathbf{CAx}(k) + \mathbf{CBu}(k) + \mathbf{CEw}(k) - \mathbf{y}_{\text{ref}}(k+1) \tag{5.41}$$

Next, combining Eq.(5.11) with Eq.(5.41)

$$\begin{bmatrix}\mathbf{v}(k+1)\\ \mathbf{x}(k+1)\end{bmatrix} = \tilde{\mathbf{A}}\begin{bmatrix}\mathbf{v}(k)\\ \mathbf{x}(k)\end{bmatrix} + \tilde{\mathbf{B}}\mathbf{u}(k) + \tilde{\mathbf{E}}\mathbf{w}(k) - \tilde{\mathbf{I}}\mathbf{y}_{\text{ref}}(k+1) \tag{5.42}$$

where

$$\tilde{\mathbf{E}} = \begin{bmatrix}\mathbf{CE}\\ \mathbf{E}\end{bmatrix} \tag{5.43}$$

and substituting Eq.(5.40) into Eq.(5.42), we obtain

$$\begin{bmatrix}\mathbf{v}(k+1)\\ \mathbf{x}(k+1)\end{bmatrix} = \tilde{\mathbf{A}}_c\begin{bmatrix}\mathbf{v}(k)\\ \mathbf{x}(k)\end{bmatrix} + \tilde{\mathbf{B}}\sum_{l=1}^{N_p}\mathbf{G}_d(l)\mathbf{y}_{\text{ref}}(k+l) + \tilde{\mathbf{E}}\mathbf{w}(k) - \tilde{\mathbf{I}}\mathbf{y}_{\text{ref}}(k+1) \tag{5.44}$$

From Eq.(5.44) we can observe that the closed-loop characteristics is determined by the state feedback and the integral action, so that the stability of the overall system is independent of the preview action. In addition, we note that the control is independent of the disturbance matrix $\mathbf{E}$; thus, the exact knowledge of the disturbance matrix is not necessary for designing the optimal preview controller.

5.4 3D-LIPM Preview Control

In this section, we will apply the linear preview control technique to humanoid robot gait control. The main problem in dynamic locomotion is to compute the robot's body movement that will achieve a preplanned ZMP reference trajectory. To calculate this movement online and allow real-time control, a simplified dynamic model is used. As mentioned previously, the ZMP cannot be achieved correctly, given its current target value alone, but the CoM needs to start moving prior to the ZMP, hence the incorporation of some future course of the ZMP is necessary. Therefore, in this section we will use the

3D-LIPM, originally presented in Section 4.3.1, and we will derive the preview controller. We will notice the loss of accuracy appearing during the walking-phase changes. The 3D-LIPM Preview Control has been extensively studied. Czarnetzki et al. [43] proposed a control scheme using an observer allowing feedback through the FSRs, while Urbahn et al. [53] used the Preview Controller on the 3D-LIPM with feedback from the FSRs and the joint encoders. Both schemes were implemented on the Nao humanoid robot.

5.4.1 3D-LIPM Preview Controller Derivation

Movement of the robot's body to achieve a given ZMP reference trajectory is reduced to planning the CoM trajectory for each axis, resulting in two 3D-LIPMs (one for each axis). The state space of the 3D-LIPM is given by Eqs.(4.40),(4.41), where the state vector $\mathbf{x} \in \mathbb{R}^3$, $v_x \in \mathbb{R}$, and the output $z_x \in \mathbb{R}$. Therefore, the derivation of the preview controller is significant simplified, since the system is Single-Input Single-Output (SISO).

The performance index J_p can be defined as

$$J_p = \sum_{i=k}^{\infty} \{Q_e \left[z_x(i) - z_x^{\text{ref}}(i)\right]^2 + R\Delta v_x^2(i)\} \tag{5.45}$$

where Q_e and R are non-negative weights, for the penalties due to tracking error and incremental control respectively. The physical interpretation of J_p is to achieve regulation without an excessive rate of change in the control signal. The preview controller minimizing Eq.(5.45) is given by

$$v_x(k) = -\mathbf{K}\mathbf{x}(k) + \sum_{l=1}^{N_p} G_d(i) z_x^{\text{ref}}(k+l) \tag{5.46}$$

where $\mathbf{K}$ and $\mathbf{G}_d(l)$ are calculated as

$$\mathbf{K} = \left[R + \mathbf{B}^\top \mathbf{P}\mathbf{B}\right]^{-1} \mathbf{B}^\top \mathbf{P}\mathbf{A} \tag{5.47}$$

$$G_d(l) = \left[R + \mathbf{B}^\top \mathbf{P}\mathbf{B}\right]^{-1} \mathbf{B}^\top \left[(\mathbf{A} - \mathbf{B}\mathbf{K})^T\right]^{l-1} \mathbf{c} Q_e \quad l = 1, \ldots, N_p \tag{5.48}$$

where the matrix $\mathbf{P}$ is the unique stabilizing solution of the following algebraic Riccati

$$\mathbf{P} = \mathbf{A}^\top \mathbf{P}\mathbf{A} + \mathbf{c} Q_e \mathbf{c}^\top - \mathbf{A}^\top \mathbf{P}\mathbf{B} \left[R + \mathbf{B}^\top \mathbf{P}\mathbf{B}\right]^{-1} \mathbf{B}^\top \mathbf{P}\mathbf{A} \tag{5.49}$$

Substituting the state feedback gain in the 3D-LIPM state space

$$\mathbf{x}(k+1) = \tilde{\mathbf{A}}\mathbf{x}(k) + \mathbf{B}u(k) \tag{5.50}$$

where

$$\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK} \tag{5.51}$$

and the system input $u(k)$ is calculated by the reference ZMP trajectory z_x^{ref} and the preview gains

$$u(k) = \sum_{l=1}^{N_p} G_d(i) z_x^{\text{ref}}(k+l) \tag{5.52}$$

Furthermore, the control u differs from v_x, as it only contains the necessary action to control the system in order to track the ZMP reference trajectory. One may observe that the performance index J_p is taken over an infinite horizon, but the controller takes into account only a finite number of steps N_p, which results in a deviation from the optimal solution. Therefore, it becomes clear that the larger N_p is, the better the approximation of the optimal solution and the smaller the tracking error is. For small N_p, the CoM fails to start moving early enough, resulting in large ZMP tracking errors and incapability of walking in practice. A large N_p instead allows nearly optimal solution, resulting in a ZMP trajectory z_x closely tracking the reference z_x^{ref}. When N_p becomes larger than a threshold, which can be system-dependent, performance does not improve anymore and the tracking error does not decrease further.

5.4.2 3D-LIPM Augmented with an Observer

In the previous section, we assumed that the walking patterns generated by the 3D-LIPM with the preview controller are sufficient for stable walking in the absence of disturbances, the presence of an accurate model and full state information. These assumptions, however, do not hold on a real humanoid robot. First, the differences between the 3D-LIPM and the multi-body model of a real humanoid can be significant, depending on the performed motions of the robot. Especially in the SS phase, where the swinging leg induces an additional momentum, which has an impact on the ZMP location. The shorter the SS phase, the faster the leg has to move to reach the next step position resulting in a large momentum acting on the robot's body. Other disturbances, not included in the model, may be external forces acting on the robot due to collisions, uneven walking terrain, and

also material properties of the robot, such as flexibility in parts or insufficient motor reaction.

Without the observer's feedback, the preview controller alone is an open-loop system depending on a perfect model to achieve stable walking. This would be adequate, if all disturbances could be known and modeled in advance. In the real world, the disturbances and the state vector, needed in the state feedback term of the preview controller, cannot be measured directly in most cases. Therefore, it becomes necessary to estimate these quantities from the system input and the measured sensor data, in order to achieve satisfactory performance in real humanoid robots. This can be done using a Luenberger observer.

The design of the observer consists of two main problems: estimating the state vector and managing a stabilizing feedback. To this end, an observer gain matrix $\mathbf{L}_{\text{obs}}$ is used to reason from differences between the expected output and the measured value. A solution for $\mathbf{L}_{\text{obs}}$ can be derived after integrating the observer into the control system. We will assume that humanoid robot is equipped with joint encoders for computing a measurement of the robot's CoM and also FSRs in the feet for computing a CoP measurement, in every discrete-time k. This assumption is true, since nowadays these sensors are available in all modern humanoid robots.

Let $\hat{\mathbf{x}}(k)$ be the estimate of the state vector $\mathbf{x}$ based on the measurements of the CoM c_x^{ENC} from the joint encoders and the ZMP z_x^{FSR} from the FSRs.

$$\mathbf{y}_{\text{m}}(k) = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\mathbf{C}_{\text{m}}} \mathbf{x}(k) = \begin{bmatrix} c_x^{\text{ENC}}(k) \\ z_x^{\text{FSR}}(k) \end{bmatrix} \tag{5.53}$$

Then, the discrete-time observer dynamics are given by

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}v_x(k) - \mathbf{L}_{\text{obs}}\left[\mathbf{y}_{\text{mes}} - \mathbf{C}_{\text{m}}\hat{\mathbf{x}}(k)\right] \tag{5.54}$$

To find a suitable $\mathbf{L}_{\text{obs}}$, the LQR method is applied. The observer gain $\mathbf{L}_{\text{obs}}$ is determined so that the matrix

$$\tilde{\mathbf{A}}_L = \mathbf{A} - \mathbf{L}_{\text{obs}}\mathbf{C}_{\text{m}} \tag{5.55}$$

is asymptotically stable.

The observer-based preview controller for the 3D-LIPM is given by

$$u(k) = -G_i \sum_{i=0}^{k} \left[\mathbf{c}\hat{\mathbf{x}}(i) - z_x^{\text{ref}}(i)\right] - \mathbf{G}_x \hat{\mathbf{x}}(k) - \sum_{l=1}^{N_p} G_d(l) z_x^{\text{ref}}(k+l) \tag{5.56}$$

and consists of an integral action on the tracking error, a proportional state feedback, and a preview action based on the future reference signal. The gains G_i, $\mathbf{G}_x$, and G_d are chosen to optimize the performance index

$$J_p = \sum_{i=k}^{\infty} \left\{ Q_e \left[z_x(i) - z_x^{\text{ref}}(i)\right]^2 + \Delta\mathbf{x}(k)^\top \mathbf{Q}_x \Delta\mathbf{x}(k) + R\Delta v_x(i) \right\} \tag{5.57}$$

where $\Delta\mathbf{x}$ is the incremental state vector. Similarly to Eq.(5.45), both the tracking error and excessive changes in state and control are penalized with weight Q_e, $\mathbf{Q}_x$, and R respectively, so that the controller obtained by minimizing the performance index in Eq.(5.56) will tend to achieve a smooth regulation of the system.

Following the preview control derivation presented in the previous section, we define the following matrices for simplicity

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{cB} \\ \mathbf{B} \end{bmatrix}, \quad \tilde{\mathbf{I}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{cA} \\ \mathbf{A} \end{bmatrix}$$
$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_e & 0 \\ 0 & \mathbf{Q}_x \end{bmatrix}, \quad \tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{I}} & \tilde{\mathbf{F}} \end{bmatrix} \tag{5.58}$$

Then, the optimal preview control gains are given by

$$G_i = \left[R + \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{I}} \tag{5.59}$$

$$\mathbf{G}_x = \left[R + \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{F}} \tag{5.60}$$

$$G_d(l) = -\left[R + \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \left[\tilde{\mathbf{A}}_c^\top\right]^{l-1} \tilde{\mathbf{P}} \tilde{\mathbf{I}}, \quad l = 1, 2, \ldots, N_p \tag{5.61}$$

where $\tilde{\mathbf{A}}_c$ is the closed-loop matrix defined as

$$\tilde{\mathbf{A}}_c = \tilde{\mathbf{A}} - \tilde{\mathbf{B}} \left[R + \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{A}} \tag{5.62}$$

Finally, $\tilde{\mathbf{P}}$ is obtained by solving the matrix ARE

$$\tilde{\mathbf{P}} = \tilde{\mathbf{A}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{A}} - \tilde{\mathbf{A}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}} \left[R + \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{B}}\right]^{-1} \tilde{\mathbf{B}}^\top \tilde{\mathbf{P}} \tilde{\mathbf{A}} + \tilde{\mathbf{Q}} \tag{5.63}$$

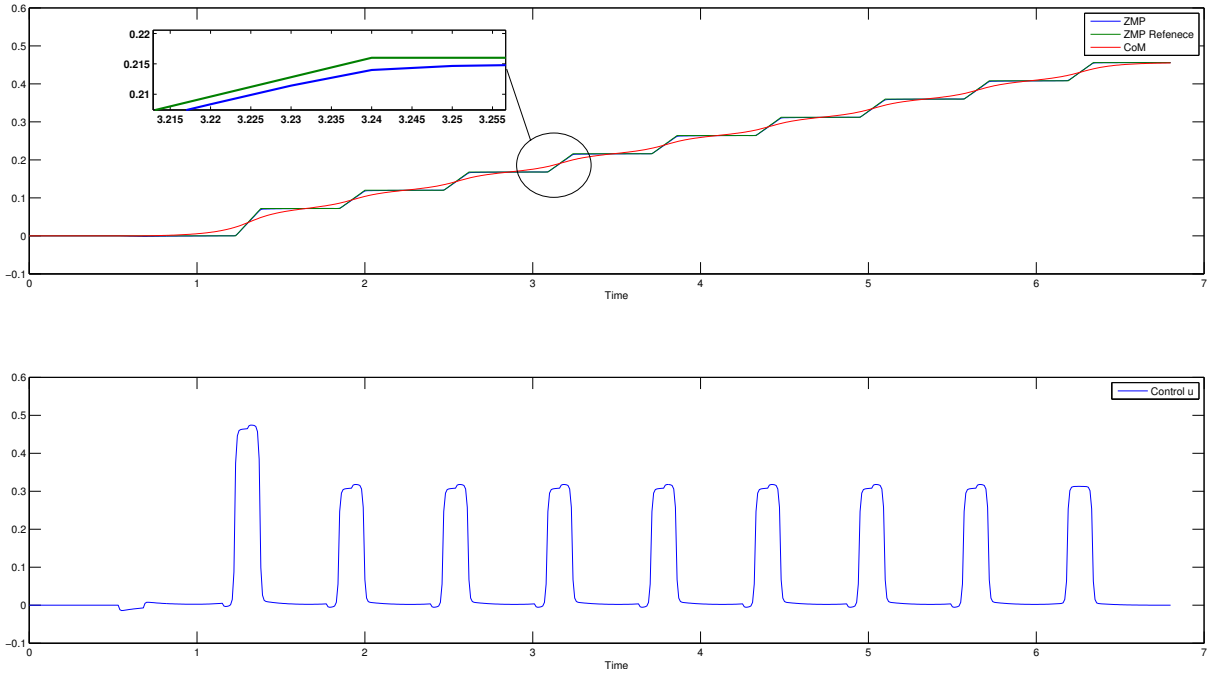

Figure 5.1: 3D-LIPM Preview control x-axis response

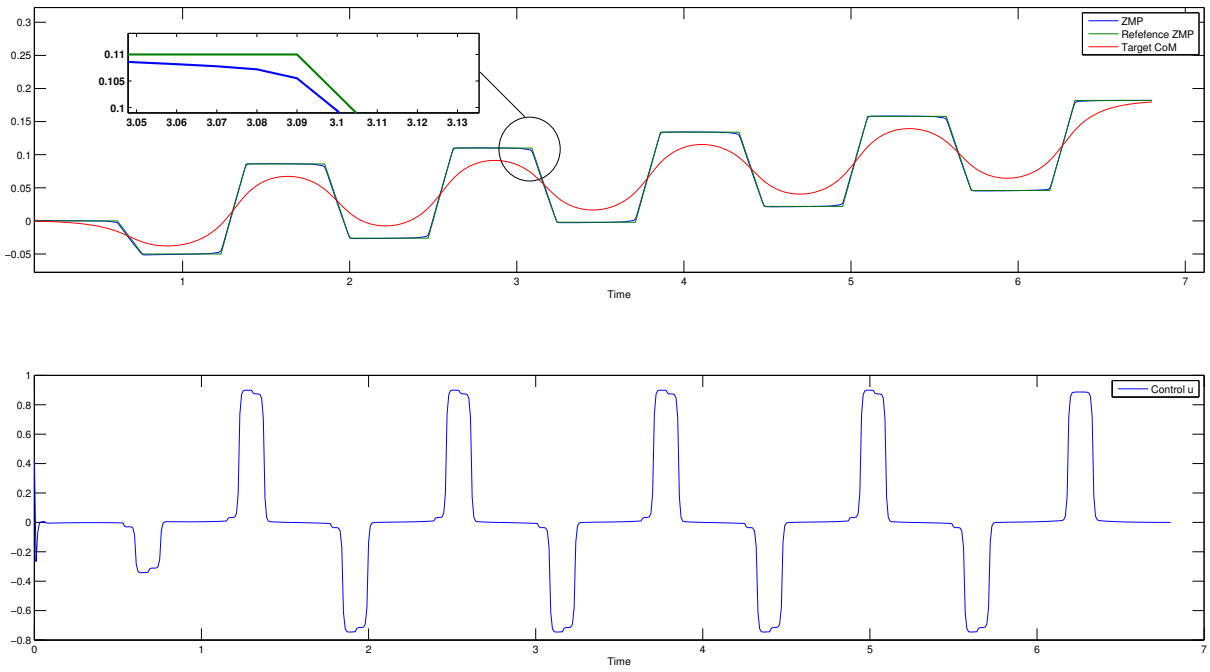

Figure 5.2: 3D-LIPM Preview Control y-axis response

Figures 5.1 and 5.2 show the 3D-LIPM Preview control response on axes x and y respectively. In the zoom-out windows, one may notice the tracking error, when the walking phase changes from SS to DS and vice versa.

5.5 Cart And Table Model Preview Control

In this section, we will use the Cart and Table model, originally presented in Section 4.3.2, for representing the humanoid robot dynamics. To generate walking patterns suitable for stable dynamic locomotion of humanoid robots, we need to make use of the optimal Preview controller. As mentioned already, the CoM must start moving early enough for the ZMP to track correctly the reference signal, resulting in dynamically stable gaits. Kajita et al. [41] first proposed the Preview control on the Cart and Table model, simulated on an HRP-2. Strom et al. [54] proposed a Preview control scheme, augmented with an observer, which allowed feedback with the FSRs. This approach was implemented on a Nao humanoid robot.

5.5.1 Cart And Table Model Preview Controller Derivation

Since the Cart and Table model dynamics are linear, and furthermore the Cart and Table model is a SISO system, the derivation of the optimal Preview control is similar to the one presented in the previous section. The Cart and Table model dynamics are given by Eqs.(4.49),(4.50). Let us again focus our derivation in the x-axis, of course similarly procedure can be carried out for the derivation in the y-axis. The performance index to be minimized is given

$$J_{\mathrm{p}} = \sum_{i=k}^{\infty} Q_e e(i)^2 + \mathbf{\Delta x}(i)^{\top} \mathbf{Q_x} \mathbf{\Delta x}(i) + R\Delta u_x^2(i) \tag{5.64}$$

where $e(k) = z_x(k) - z_x^{\mathrm{ref}}(k)$, $\mathbf{\Delta x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$ is the incremental state vector, $\Delta u_x(k) = u_x(k) - u_x(k-1)$ is the incremental input, $\mathbf{Q_x}$ is a 3×3 symmetric non-negative definite matrix, $Q_e > 0$, and $R > 0$. When the ZMP reference can be previewed for N_p steps in the future, the optimal Preview control which minimizes the performance index in Eq.(5.15) is given by

$$u_x(k) = -G_i \sum_{i=0}^{k} e(k) - \mathbf{G_x^{\top}} \mathbf{x}(k) - \sum_{j=1}^{N_p} G_p(j) z_x^{\mathrm{ref}}(k+j) \tag{5.65}$$

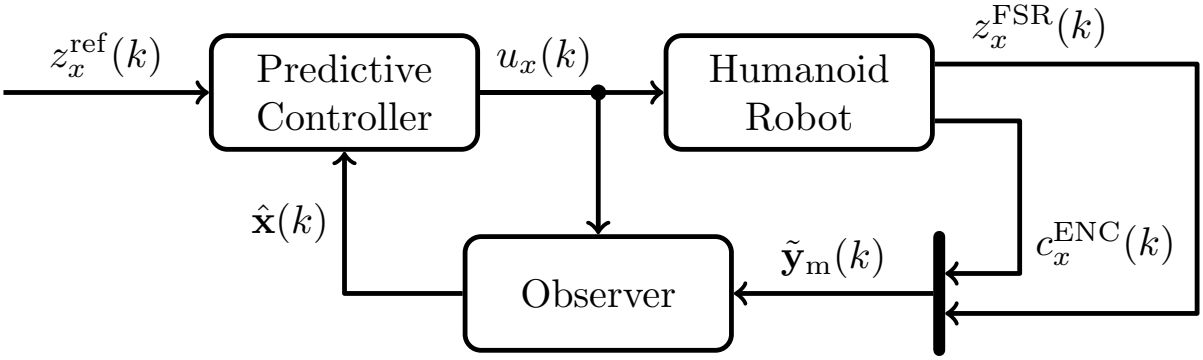

Figure 5.3: Cart and Table Model augmented with an Observer

The state-feedback gain vector $\mathbf{G_x}$, the prediction gains $G_p(j)$, and the integral gain G_i are computed from the system matrices $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}^\top$ and using Eqs.(5.59)-(5.63).

As shown in Figures 5.4, 5.5, the ZMP reference trajectory is very closely tracked by the ZMP trajectory realized by the system dynamics, even when the phase changes from SS to DS and vice versa.

5.5.2 Cart and Table Model Augmented with an Observer

For the control task we did assume that the state $\mathbf{x}$ is known at every sampling time k, which in practice is not always true. Therefore, we chose to augment the control system with a Luenberger observer to accomplish the state feedback task. We present a sensor fusion approach (Figure 5.3) for the Cart and Table model based on sensors commonly available on humanoid robots. We assume again that the humanoid robot is equipped with encoders at all joints, which can be used to measure the real CoM c_x^{ENC}, and pressure-sensitive FSR at each foot, which can be used to measure the ZMP z_x^{FSR} that coincides with the CoP during stable walk. Then, the measurement $\mathbf{y}_{\text{m}}$ sent to the observer is:

$$\mathbf{y}_{\text{m}}(k) = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & -h_c/g \end{bmatrix}}^{\mathbf{C}_{\text{m}}} \mathbf{x}(k) = \begin{bmatrix} c_x^{\text{ENC}}(k) \\ z_x^{\text{FSR}}(k) \end{bmatrix} \tag{5.66}$$

and the observer dynamics over the estimated state $\hat{\mathbf{x}}$ are:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{b}u_x(k) + \mathbf{L}_{\text{obs}}\big(\mathbf{y}_{\text{m}}(k) - \mathbf{C}_{\text{m}}\hat{\mathbf{x}}(k)\big) \tag{5.67}$$

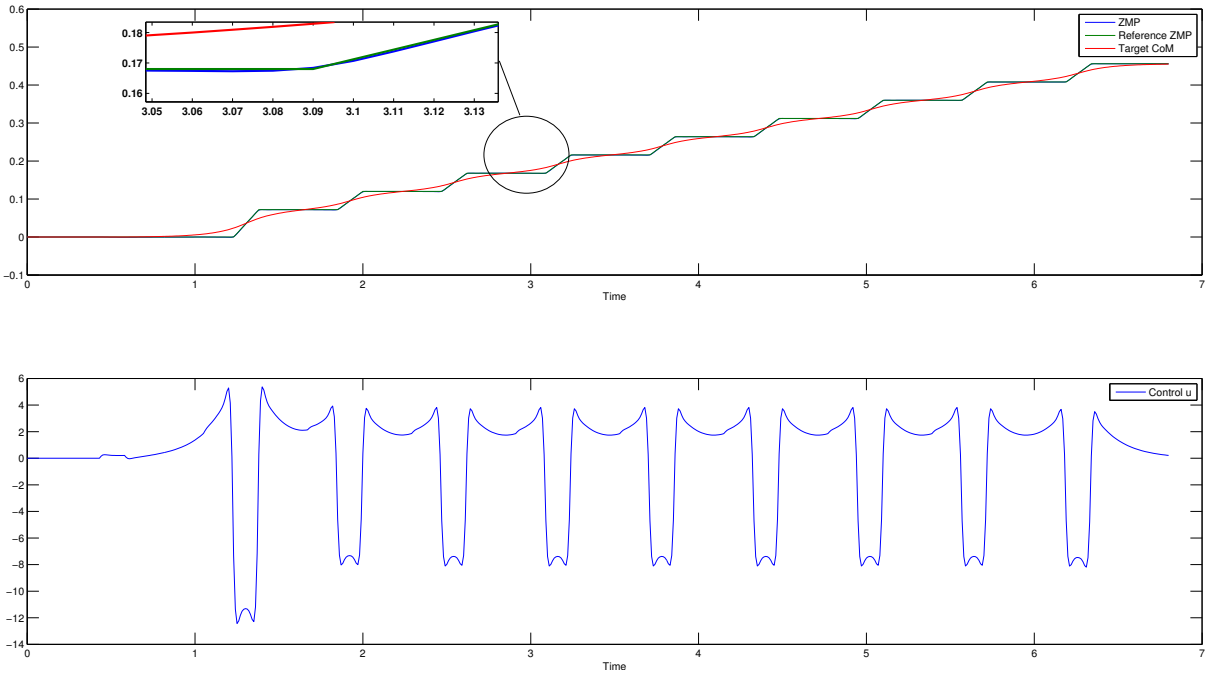

Figure 5.4: Cart and Table Preview Control x-axis response

To find a suitable observer gain $\mathbf{L}_{\text{obs}}$, the discrete-time infinite-horizon LQR is adopted with matrices $(\mathbf{A} - \mathbf{b}\mathbf{G}_{\mathbf{x}}^{\top})^{\top}$, $\mathbf{C}_{\text{m}}^{\top}$, $\mathbf{Q}_{\text{obs}}$, and $\mathbf{R}_{\text{obs}}$. The observer can compensate for errors in the model and for disturbances encountered while walking, but unfortunately cannot resolve effectively the sensor noise, delay, and bias, which we will effectively address in chapter 7 .

5.5.3 Cart and Table Preview Control with Auxiliary ZMP

To account for dynamic changes in the walking pattern, we make use of the auxiliary ZMP on the Cart and Table Model, originally introduced for the 3D-LIPM by Kajita et al. [47]. The auxiliary ZMP modifies the instantaneous behavior of the walking robot and as a result the long-term parameters, namely walking cycle and step width, to ensure compliance with the system dynamics.

The output z_x of the pattern generator, which is used to control the humanoid robot, gives a ZMP consistent with the robot dynamics as given by Eq.(4.49) and is guaranteed to be arbitrarily close to the reference ZMP by appropriately selecting the weights of Eq.(5.15). Ideally, the reference ZMP z_x^{ref} should be modified online by an auxiliary ZMP

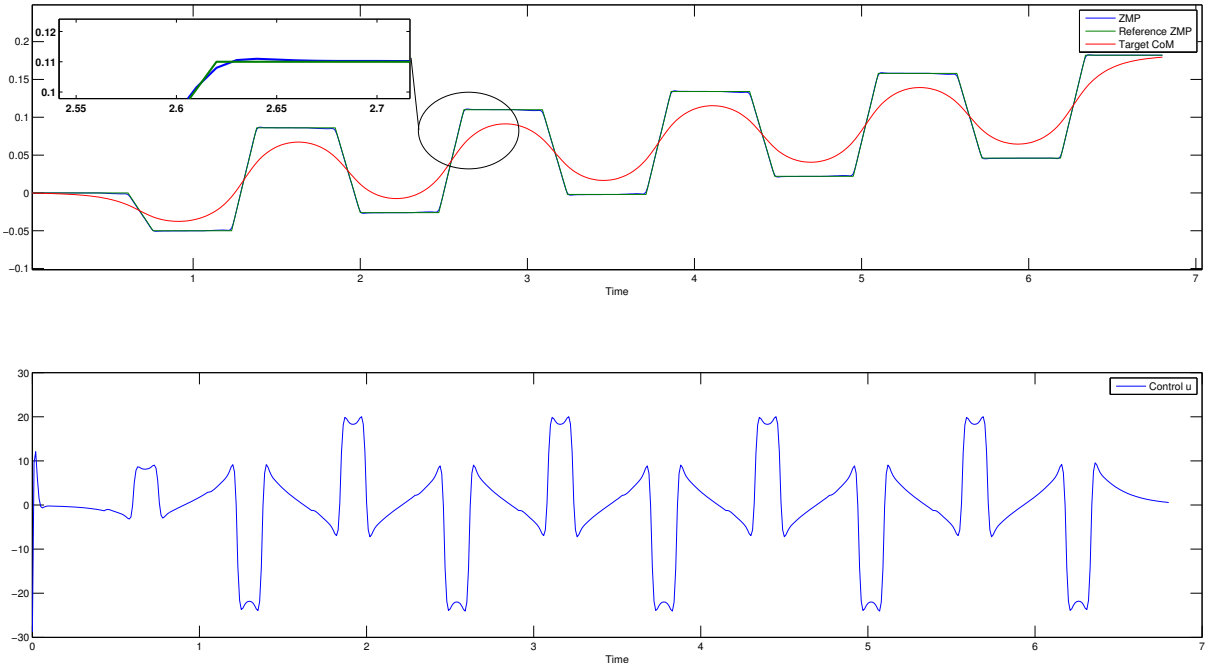

Figure 5.5: Cart and Table Preview Control y-axis response

z_x^{aux} to counter disturbances and/or the presence of uneven terrain, noted by a tracking error ϵ:

$$z_x(k) = z_x^{\text{ref}}(k) + z_x^{\text{aux}}(k) + \epsilon \tag{5.68}$$

Practically, such a modification can be achieved online with the use of an inverse system as proposed by Sain et al. [55]. Let $u_x(k) = v_x(k) - \mathbf{G}_{\mathbf{x}}^{\top}\mathbf{x}(k)$, where v_x is provided by the inverse system. Substituting into Eq.(4.49):

$$\mathbf{x}(k+1) = \overbrace{(\mathbf{A} - \mathbf{b}\mathbf{G}_{\mathbf{x}}^{\top})}^{\bar{\mathbf{A}}}\mathbf{x}(k) + \mathbf{b}v_x(k) \tag{5.69}$$

The state space for the inverse Cart and Table model is derived by considering Eq.(4.50) one time sample in the future:

$$z_x(k+1) = \mathbf{c}^{\top}\mathbf{x}(k+1) = \mathbf{c}^{\top}\bar{\mathbf{A}}\mathbf{x}(k) + \mathbf{c}^{\top}\mathbf{b}v_x(k)$$

Then, by solving for v_x (assuming $\mathbf{c}^{\top}\mathbf{b} \neq 0$)

$$v_x(k) = \overbrace{-(\mathbf{c}^{\top}\mathbf{b})^{-1}\mathbf{c}^{\top}\bar{\mathbf{A}}}^{\mathbf{c}_{\text{inv}}^{\top}}\mathbf{x}(k) + \overbrace{(\mathbf{c}^{\top}\mathbf{b})^{-1}}^{\mathbf{d}_{\text{inv}}} z_x(k+1)$$

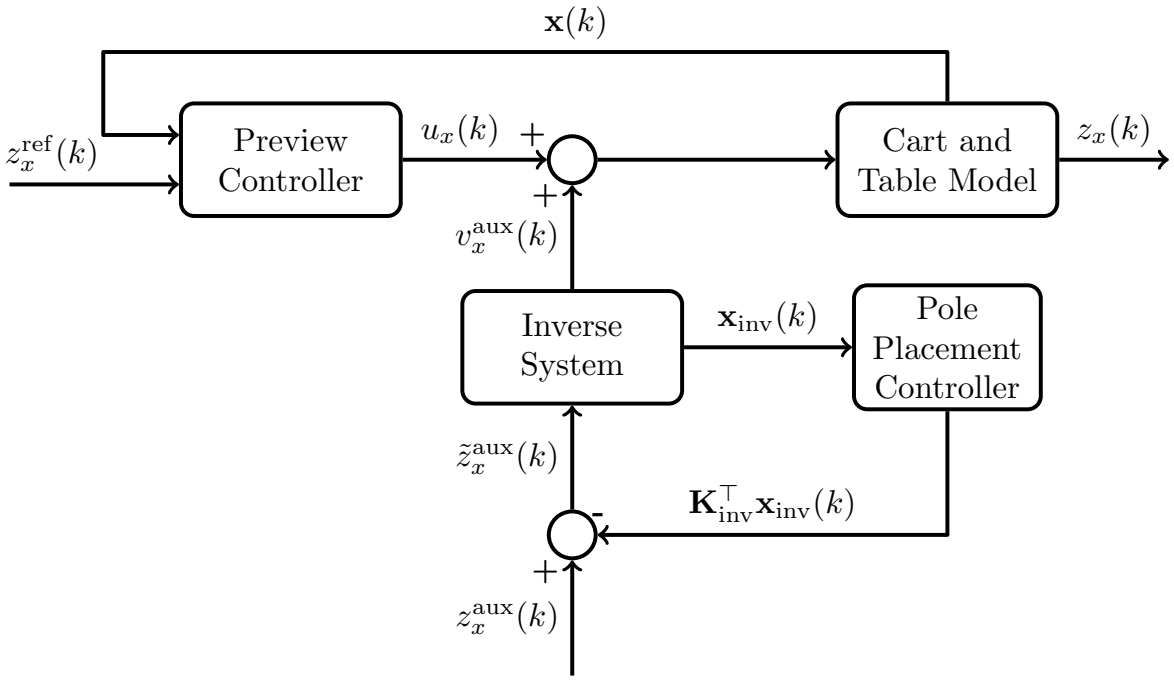

Figure 5.6: Preview Controller with inverse system control scheme

and substituting into Eq.(5.69)

$$\mathbf{x}(k+1) = \overbrace{(\bar{\mathbf{A}} - \mathbf{b}(\mathbf{c}^{\top}\mathbf{b})^{-1}\mathbf{c}^{\top}\bar{\mathbf{A}})}^{\mathbf{A}_{\text{inv}}}\mathbf{x}(k) + \overbrace{\mathbf{b}(\mathbf{c}^{\top}\mathbf{b})^{-1}}^{\mathbf{b}_{\text{inv}}} z_x(k+1)$$

the dynamics of the inverse Cart and Table model become:

$$\mathbf{x}_{\text{inv}}(k+1) = \mathbf{A}_{\text{inv}}\mathbf{x}_{\text{inv}}(k) + \mathbf{b}_{\text{inv}} z_x^{\text{aux}}(k) \tag{5.70}$$

$$v_x^{\text{aux}}(k) = \mathbf{c}_{\text{inv}}^{\top}\mathbf{x}_{\text{inv}}(k) + \mathbf{d}_{\text{inv}} z_x^{\text{aux}}(k) \tag{5.71}$$

The inverse Cart and Table model is unstable, therefore a discrete-time pole-placement controller was employed for the stabilization of the system, replacing z_x^{aux} in Eqs.(5.70),(5.71) by:

$$\tilde{z}_x^{\text{aux}}(k) = z_x^{\text{aux}}(k) - \mathbf{K}_{\text{inv}}^{\top}\mathbf{x}_{\text{inv}}(k) \tag{5.72}$$

where $\mathbf{K}_{\text{inv}}$ is the pole-placement gain vector. The auxiliary ZMP z_x^{aux} in Eq.(5.72) can be computed at each discrete time k using information obtained by the sensors of the humanoid robot [47]. The proposed control scheme is shown in Figure 5.6. Furthermore

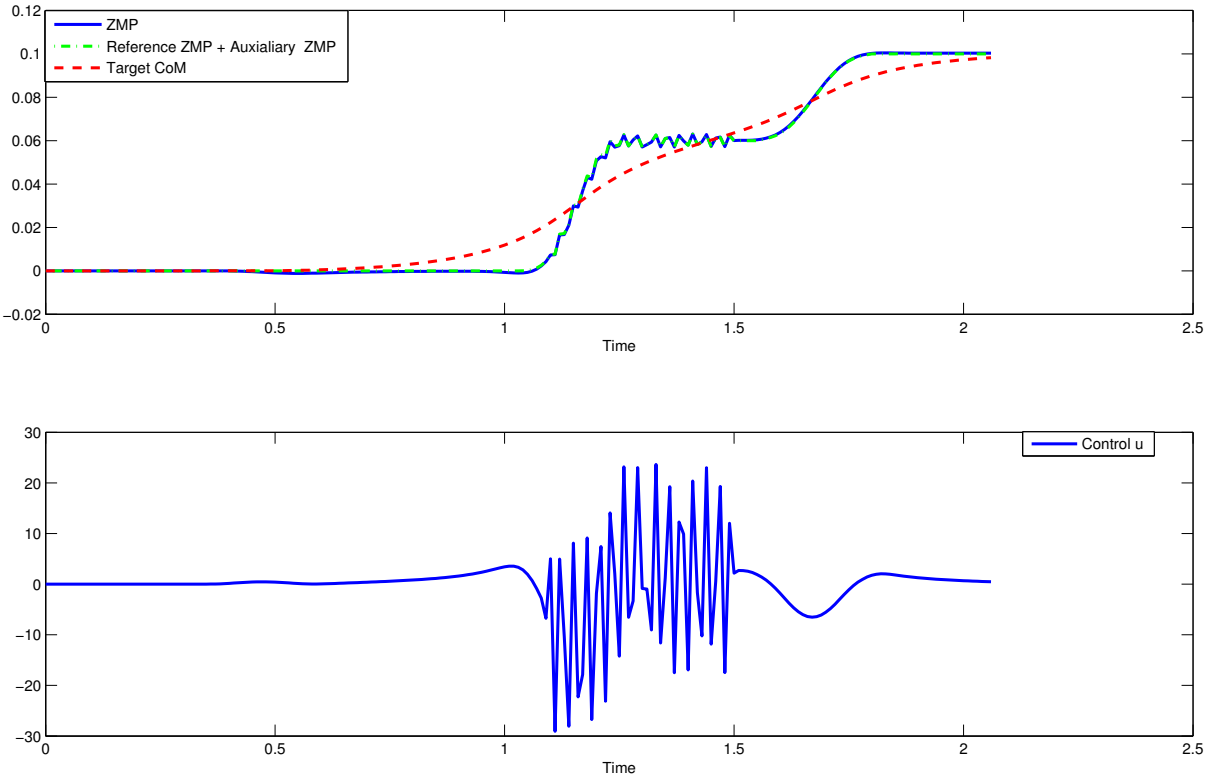

Figure 5.7: x-axis response of the Preview Controller with the inverse system

Figure 5.7 shows how the ZMP is correctly realized and tracked with the assistance of some given auxiliary ZMP input under disturbance (from 1.1 to 1.5sec) in the x-axis for three consecutive steps of straight walk.

5.6 Linear Model Predictive Control

The mathematical background needed for the conventional Linear Model Predictive Control (LMPC), along with the approximated LMPC with orthonormal basis functions are extensively presented by Piperakis [56], where also the Multiple-Input Multiple-Output (MIMO) case is outlined and applied in a real 2-DOF helicopter. In addition Maciejowski [57] also grants an extensive presentation of the LMPC framework when constraints are taken into consideration.

In this section an introduction to the LMPC approximated with Laguerre functions is given. A former approach by Wang [58], assumes that the reference signal remains

constant for the duration of the prediction horizon N_p. In general and in our specific problem, this assumption does not hold, therefore an extension to the already existing LMPC framework is mandatory.

In the first sections, the orthonormal Laguerre basis set and its application to LMPC are short presented. In addition the unconstrained solution is derived for any arbitrary SISO LTI dynamic system. Furthermore, an extension to MIMO dynamic systems is given in Section 5.6.6, generalizing this novel LMPC framework. Finally in Section 5.7 this novel framework is successfully applied in the Cart and Table model.

5.6.1 The Laguerre basis set

For notational simplicity the SISO case is examined, however the results can be generalized in the MIMO case [56, 58]. The control vector that is optimized in a predictive control framework is the $\mathbf{\Delta U}$, defined by

$$\mathbf{\Delta U}(k) = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_c-1) \end{bmatrix}^\top \tag{5.73}$$

where the dimension of the control vector is N_c, namely the control horizon. At time k, any element within $\mathbf{\Delta U}$ can be represented using the discrete δ-function in conjunction with $\mathbf{\Delta U}$.

$$\Delta u(k+i) = \begin{bmatrix} \delta(i) & \delta(i-1) & \dots & \delta(i-N_c+1) \end{bmatrix} \mathbf{\Delta U} \tag{5.74}$$

where δ is the kronecker function, $\delta(i) = 1$, if $i = 0$; and $\delta(i) = 0$ if $i \neq 0$. The δ function acts like a pulse, and the function $\delta(i-d)$ shifts the center of the pulse forward as the index d increases. From this expression it is clear that pulse operators are used in capturing the control trajectory, if we consider the vector $\mathbf{\Delta U}$ as the coefficient vector. Thus, $\Delta u(k+i)$ for $i = 0, 1, \dots, N_c-1$ can be approximated by a discrete polynomial function. Wang [58] proposed to use a set of discrete-time Laguerre functions to approximate the sequence $\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_c-1)$. The reason for using Laguerre functions is that they satisfy a difference equation which can simplify the computations needed.

5.6.2 Discrete-time Laguerre Networks

We will start our analysis from the frequency domain, where an interesting state-space realization of the Laguerre networks can be obtained. Furthermore, a compact expression

of the Laguerre functions in the time domain can be then derived using the inverse z-transformation.

The z-transforms of N discrete-time Laguerre functions leads to N Laguerre networks in the frequency domain:

$$\begin{aligned}\Gamma_1(z) &= \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}} \\ \Gamma_2(z) &= \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}}\frac{z^{-1}-\alpha}{1-\alpha z^{-1}} \\ &\vdots \\ \Gamma_N(z) &= \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}}\left(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}\right)^{N-1}\end{aligned}$$

where α is the pole of the networks, with $0 \leq \alpha < 1$ for stability of the network. The free parameter α, is selected by the designer, therefore it is also called the scaling factor. The Laguerre networks are well known for satisfying the orthonormality property. In the frequency domain, this orthonormality property is expressed in terms of the orthonormal functions Γ_m, with $m = 1, 2, \ldots$, as

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_m(e^{j\omega})\Gamma_m(e^{j\omega})^* d\omega = 1 \tag{5.75}$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_m(e^{j\omega})\Gamma_n(e^{j\omega})^* d\omega = 0 \;\; m \neq n \tag{5.76}$$

where $\Gamma_m(e^{j\omega})^*$ denotes the conjugate of $\Gamma_n(e^{j\omega})$.

5.6.3 Discrete-time Laguerre functions

The discrete-time Laguerre functions are obtained through the inverse z-transform of the Laguerre networks. However, taking the inverse z-transform of the Laguerre networks does not lead to compact expression of the Laguerre functions in the discrete-time domain. Another way, more straightforward, is to find these discrete-time functions based on a state-space realization of the networks

$$\Gamma_k(z) = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}}\Gamma_{k-1}(z) \tag{5.77}$$

with initial state $\Gamma_1(z) = \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}}$.
If $l_i(k)$ is the inverse z-transform of $\Gamma_i(z)$, the set of discrete-time Laguerre functions can be expressed in vector form as

$$\mathbf{L}(k) = \begin{bmatrix} l_1(k) & l_2(k) & \cdots & l_N(k) \end{bmatrix}^\top \tag{5.78}$$

Due to Eq.(5.77), the set of discrete-time Laguerre functions satisfies the discrete-time state space

$$\mathbf{L}(k+1) = \mathbf{A}_l(\alpha)\mathbf{L}(k) \tag{5.79}$$

with initial condition

$$\mathbf{L}(0) = \sqrt{1-\alpha^2}\begin{bmatrix} 1 & -\alpha & \cdots & (-1)^{N-1}\alpha^{N-1} \end{bmatrix}^\top \tag{5.80}$$

where $\mathbf{A}_l$ is a specific lower-triangular $N \times N$ matrix [58].
One important aspect of the Laguerre basis is the orthonormality property, given in the discrete-time domain by Eq.(5.81). This property plays an essential role in the optimization procedure of the LMPC.

$$\sum_{k=0}^{\infty} l_i(k) l_i(k) = 1 \tag{5.81}$$

$$\sum_{k=0}^{\infty} l_i(k) l_j(k) = 0 \ , \ i \neq j \tag{5.82}$$

5.6.4 LMPC Approximated with Laguerre Basis

To derive the LMPC, a modified state space is used. By enhancing the state-space model with an embedded integrator [57], the modified state vector becomes:

$$\mathbf{x}_e(k) = \begin{bmatrix} \mathbf{\Delta x}(k)^\top & y(k) \end{bmatrix}^\top \tag{5.83}$$

The augmented state-space with the embedded integrator is:

$$\mathbf{x}_e(k+1) = \overbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0}^\top \\ \mathbf{c}^\top\mathbf{A} & 1 \end{bmatrix}}^{\mathbf{A}_e} \mathbf{x}_e(k) + \overbrace{\begin{bmatrix} \mathbf{b} \\ \mathbf{c}^\top\mathbf{b} \end{bmatrix}}^{\mathbf{b}_e} \Delta u(k) \tag{5.84}$$

$$y(k) = \overbrace{\begin{bmatrix} \mathbf{0}^\top & 1 \end{bmatrix}}^{\mathbf{c}_e^\top} \mathbf{x}_e(k) \tag{5.85}$$

At discrete time k, the incremental control signal trajectory $\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+m), \ldots$ is regarded as the impulse response of a stable dynamic system. Thus, a set of Laguerre functions, $l_1(m), l_2(m), \ldots, l_N(m)$, are used to capture this dynamic response with the corresponding Laguerre coefficients $\eta_1, \eta_2, \ldots, \eta_N$, which will be determined by the optimization procedure.

For an arbitrary future time sample m, with k being the start of the moving horizon window, the incremental control input will be given by:

$$\Delta u(k+m) = \sum_{j=1}^{N} \eta_j(k) l_j(m) = \mathbf{L}(m)^\top \boldsymbol{\eta}(k) \tag{5.86}$$

In this design framework, there is no explicit control horizon parameter N_c, as in the conventional LMPC approach [57]. Instead, the number of Laguerre functions N along with parameter α determine the complexity of the incremental control trajectory. Note that for $\alpha = 0$ and $N = N_c$, the conventional LMPC approach is recovered.

When using Eq.(5.86) in the modified state-space model Eq.(5.84), the prediction of the future state variable $\mathbf{x}_e(k+m|k)$ at sampling time m based on $\mathbf{x}_e(k)$ is given by:

$$\mathbf{x}_e(k+m|k) = \mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top \boldsymbol{\eta}(k) \tag{5.87}$$

and the predicted output at sampling time m is given by:

$$y(k+m|k) = \mathbf{c}_e^\top \mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{c}_e^\top \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top \boldsymbol{\eta}(k) \tag{5.88}$$

5.6.5 Unconstrained LMPC Derivation

In the conventional LMPC approach [57], the performance index to be minimized is given by:

$$\underset{\mathbf{\Delta U}(k)}{\text{minimize}} \;\; J_\text{u} = \|\mathbf{Y}^\text{ref}(k) - \mathbf{Y}(k)\|_2^2 + \|\mathbf{\Delta U}(k)\|_\mathbf{R}^2 \tag{5.89}$$

where $\mathbf{R}$ is a diagonal positive definite matrix and

$$\begin{aligned}
\mathbf{Y}^\text{ref}(k) &= \begin{bmatrix} y^\text{ref}(k+1) & y^\text{ref}(k+2) & \cdots & y^\text{ref}(k+N_p) \end{bmatrix}^\top \\
\mathbf{Y}(k) &= \begin{bmatrix} y(k+1|k) & y(k+2|k) & \cdots & y(k+N_p|k) \end{bmatrix}^\top \\
\mathbf{\Delta U}(k) &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \cdots & \Delta u(k+N_c-1) \end{bmatrix}^\top
\end{aligned}$$

Replacing each term $\Delta u(k+m)$ as in Eq.(5.86) and using the orthonormality of the Laguerre functions, the performance index Eq.(5.89) becomes equivalent to:

$$\underset{\boldsymbol{\eta}(k)}{\text{minimize}} \quad J_{\text{u}} = \|\mathbf{Y}^{\text{ref}}(k) - \mathbf{Y}(k)\|_2^2 + \|\boldsymbol{\eta}(k)\|_{\mathbf{R_L}}^2 \tag{5.90}$$

where $\mathbf{R_L}$ is a positive definite matrix based on matrix $\mathbf{R}$.

Theorem 5.5 (Optimal Laguerre Coefficients)**.** *The optimal Laguerre coefficients* $\boldsymbol{\eta}$ *for the minimization problem given in Eq.(5.90) are:*

$$\boldsymbol{\eta}(k) = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{R_L}\right)^{-1} \boldsymbol{\Phi}^\top \left(\mathbf{Y}^{\text{ref}}(k) - \mathbf{T}\mathbf{x}_e(k)\right) \tag{5.91}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{c}_e^\top \mathbf{A}_e \\ \mathbf{c}_e^\top \mathbf{A}_e^2 \\ \vdots \\ \mathbf{c}_e^\top \mathbf{A}_e^{N_p} \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(N_p) \end{bmatrix}, \quad \phi(m) = \sum_{i=0}^{m-1} \mathbf{c}_e^\top \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top$$

Proof. Using Eq.(5.88) and denoting the convolution appearing therein as

$$\phi(m) = \sum_{i=0}^{m-1} \mathbf{c}_e^\top \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top$$

$\mathbf{Y}(k)$ takes the form

$$\mathbf{Y}(k) = \overbrace{\begin{bmatrix} \mathbf{c}_e^\top \mathbf{A}_e \\ \mathbf{c}_e^\top \mathbf{A}_e^2 \\ \vdots \\ \mathbf{c}_e^\top \mathbf{A}_e^{N_p} \end{bmatrix}}^{\mathbf{T}} \mathbf{x}_e(k) + \overbrace{\begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(N_p) \end{bmatrix}}^{\boldsymbol{\Phi}} \boldsymbol{\eta}(k) \tag{5.92}$$

Substituting Eq.(5.92) in Eq.(5.90) gives the standard quadratic form

$$J = \boldsymbol{\eta}(k)^\top (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{R_L}) \boldsymbol{\eta}(k) + 2(\mathbf{T}\mathbf{x}_e(k) - \mathbf{Y}^{\text{ref}}(k))^\top \boldsymbol{\Phi} \boldsymbol{\eta}(k) + \text{const}$$

Taking the gradient of J with respect to $\boldsymbol{\eta}(k)$ yields

$$\nabla_{\boldsymbol{\eta}} J = 2(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{R_L}) \boldsymbol{\eta}(k) + 2\boldsymbol{\Phi}^\top (\mathbf{T}\mathbf{x}_e(k) - \mathbf{Y}^{\text{ref}}(k)) \tag{5.93}$$

Then, the optimal solution can be obtained by solving Eq.(5.93) for $\boldsymbol{\eta}(k)$

$$\boldsymbol{\eta}(k) = \left(\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \mathbf{R}_{\mathbf{L}}\right)^{-1}\boldsymbol{\Phi}^{\top}\left(\mathbf{Y}^{\mathrm{ref}}(k) - \mathbf{T}\mathbf{x}_e(k)\right) \tag{5.94}$$

where the optimal coefficient vector $\boldsymbol{\eta}$ can be obtained by inverting an $N \times N$ matrix, instead of an $N_c \times N_c$ as in the conventional approach. □

Upon obtaining the optimal coefficients $\boldsymbol{\eta}(k)$, the receding horizon control law [58] is realized as:

$$\Delta u(k) = \mathbf{L}(0)^{\top}\boldsymbol{\eta}(k) \tag{5.95}$$

5.6.6 Extension to MIMO

We can extend the formulation presented in the previous section in the case of MIMO LTI systems. Assume again that, $\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^r, \mathbf{y} \in \mathbb{R}^p$, the modified state vector becomes

$$\mathbf{x}_e(k) = \begin{bmatrix}\boldsymbol{\Delta}\mathbf{x}(k)^{\top} & \mathbf{y}(k)^{\top}\end{bmatrix}^{\top} \tag{5.96}$$

The augmented state-space with the embedded integrator is

$$\mathbf{x}_e(k+1) = \overbrace{\begin{bmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{CA} & \mathbf{I}_p\end{bmatrix}}^{\mathbf{A}_e}\mathbf{x}_e(k) + \overbrace{\begin{bmatrix}\mathbf{B}\\ \mathbf{CB}\end{bmatrix}}^{\mathbf{B}_e}\boldsymbol{\Delta}\mathbf{u}(k) \tag{5.97}$$

$$\mathbf{y}(k) = \overbrace{\begin{bmatrix}\mathbf{0} & \mathbf{I}_p\end{bmatrix}}^{\mathbf{C}_e}\mathbf{x}_e(k) \tag{5.98}$$

The idea is to assign in each input a set of Laguerre basis capturing the response. A different Laguerre pole a, can be used to influence the decay rate of each incremental control signal. For instance, if we want a specific incremental control to decay fast, then we choose its pole location to be zero. Thus, the description can be extended to MIMO with full flexibility in the choice of a and N parameters. Let

$$\boldsymbol{\Delta}\mathbf{u}(k) = \begin{bmatrix}\Delta u_1(k) & \Delta u_2(k) & \dots & \Delta u_r(k)\end{bmatrix}^{\top} \tag{5.99}$$

and the input matrix $\mathbf{B}_e$ can be partitioned as

$$\mathbf{B}_e = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_r\end{bmatrix} \tag{5.100}$$

where $\mathbf{b}_i$ is the i-th column of the matrix $\mathbf{B}_e$. We then express the i-th control signal $\Delta u_i(k)$ by choosing a pole a_i and order N_i such that

$$\Delta u_i(k+m) = \mathbf{L}_i(m)^\top \boldsymbol{\eta}_i(k) \tag{5.101}$$

where $\boldsymbol{\eta}_i$ and $\mathbf{L}_i$ are the Laguerre network description of the i-th control, with

$$\mathbf{L}_i(m) = \begin{bmatrix} l_1^i(m) & l_2^i(m) & \dots & l_{N_i}^i(m) \end{bmatrix}^\top \tag{5.102}$$

Based on the partition of the input matrix $\mathbf{B}_e$ and given state variable information at $\mathbf{x}_e(k)$, the prediction of the future state at time m can be written as

$$\begin{aligned} \mathbf{x}_e(k+m|k) &= \mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{A}_e^{m-i-1} \begin{bmatrix} \mathbf{b}_1\mathbf{L}_1(i)^\top & \mathbf{b}_2\mathbf{L}_2(i)^\top & \dots & \mathbf{b}_r\mathbf{L}_r(i)^\top \end{bmatrix} \boldsymbol{\eta}(k) \\ &= \mathbf{A}_e^m \mathbf{x}_e(k) + \boldsymbol{\phi}(m)^\top \boldsymbol{\eta}(k) \end{aligned} \tag{5.103}$$

Furthermore, the prediction of the output $\mathbf{y}$ at time m is

$$\begin{aligned} \mathbf{y}(k+m|k) &= \mathbf{C}_e\mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{C}_e\mathbf{A}_e^{m-i-1} \begin{bmatrix} \mathbf{b}_1\mathbf{L}_1(i)^\top & \mathbf{b}_2\mathbf{L}_2(i)^\top & \dots & \mathbf{b}_r\mathbf{L}_r(i)^\top \end{bmatrix} \boldsymbol{\eta}(k) \\ &= \mathbf{C}_e\mathbf{A}_e^m \mathbf{x}_e(k) + \boldsymbol{\phi}(m)^\top \boldsymbol{\eta}(k) \end{aligned} \tag{5.104}$$

where the parameter vector $\boldsymbol{\eta}$ and the matrix $\boldsymbol{\phi}^\top$ consist of individual coefficient vectors given by

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1^\top & \boldsymbol{\eta}_2^\top & \dots & \boldsymbol{\eta}_r^\top \end{bmatrix}^\top \tag{5.105}$$

$$\boldsymbol{\phi}(m)^\top = \sum_{i=0}^{m-1} \mathbf{C}_e\mathbf{A}_e^{m-i-1} \begin{bmatrix} \mathbf{b}_1\mathbf{L}_1(i)^\top & \mathbf{b}_2\mathbf{L}_2(i)^\top & \dots & \mathbf{b}_r\mathbf{L}_r(i)^\top \end{bmatrix} \tag{5.106}$$

In this point we have to note that

$$\boldsymbol{\phi}(m)_r^\top = \sum_{i=0}^{m-1} \mathbf{C}_e\mathbf{A}_e^{m-i-1}\mathbf{b}_r\mathbf{L}_r(i)^\top \tag{5.107}$$

The convolution summation in the MIMO case is decomposed into computing the subsystems, and the computed results are put together block by block to form a multi-input structure.

Similar to the SISO case, the cost function for the MIMO case is defined as

$$\underset{\boldsymbol{\Delta}\mathbf{U}(k)}{\text{minimize}} \;\; J_\text{u} = \|\mathbf{Y}^\text{ref}(k) - \mathbf{Y}(k)\|_2^2 + \|\boldsymbol{\Delta}\mathbf{U}(k)\|_\mathbf{R}^2 \tag{5.108}$$

where $\mathbf{R}$ is a diagonal positive definite matrix of appropriate dimension and

$$\begin{aligned}\mathbf{Y}^{\text{ref}}(k) &= \begin{bmatrix}\mathbf{y}^{\text{ref}}(k+1)^\top & \mathbf{y}^{\text{ref}}(k+2)^\top & \cdots & \mathbf{y}^{\text{ref}}(k+N_p)^\top\end{bmatrix}^\top \\ \mathbf{Y}(k) &= \begin{bmatrix}\mathbf{y}(k+1|k)^\top & \mathbf{y}(k+2|k)^\top & \cdots & \mathbf{y}(k+N_p|k)^\top\end{bmatrix}^\top \\ \boldsymbol{\Delta}\mathbf{U}(k) &= \begin{bmatrix}\boldsymbol{\Delta}\mathbf{u}(k)^\top & \boldsymbol{\Delta}\mathbf{u}(k+1)^\top & \cdots & \boldsymbol{\Delta}\mathbf{u}(k+N_c-1)^\top\end{bmatrix}^\top\end{aligned}$$

As before, by replacing each component of the vector $\boldsymbol{\Delta}\mathbf{u}(k+m)$ as in Eq.(5.101) and using the orthonormality property, we can obtain the equivalent cost in terms of the Laguerre coefficients $\boldsymbol{\eta}$

$$\underset{\boldsymbol{\eta}(k)}{\text{minimize}} \quad J_{\text{u}} = \|\mathbf{Y}^{\text{ref}}(k) - \mathbf{Y}(k)\|_2^2 + \|\boldsymbol{\eta}(k)\|_{\mathbf{R_L}}^2 \tag{5.109}$$

The optimal coefficients are given again by the Eq.(5.91), but the matrices $\boldsymbol{\Phi}$ and $\mathbf{T}$ are defined as

$$\mathbf{T} = \begin{bmatrix}\mathbf{C}_e\mathbf{A}_e \\ \mathbf{C}_e\mathbf{A}_e^2 \\ \vdots \\ \mathbf{C}_e\mathbf{A}_e^{N_p}\end{bmatrix} \quad \boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{\phi}(1) \\ \boldsymbol{\phi}(2) \\ \vdots \\ \boldsymbol{\phi}(N_p)\end{bmatrix} \tag{5.110}$$

where the $n \times (N_1 + N_2 + \ldots + N_r)$ matrices $\boldsymbol{\phi}$ are given by Eq.(5.106) and the optimal Laguerre coefficients are a vector of dimension $(N_1+N_2+\ldots+N_r)$ as shown in Eq.(5.105). Finally the receding horizon control can be realized as

$$\boldsymbol{\Delta}\mathbf{u}(k) = \begin{bmatrix}\mathbf{L}_1(0)^\top & \mathbf{o}_2^\top & \ldots & \mathbf{o}_r^\top \\ \mathbf{o}_1^\top & \mathbf{L}_2(0)^\top & \ldots & \mathbf{o}_r^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{o}_1^\top & \mathbf{o}_2^\top & \ldots & \mathbf{L}_r(0)^\top\end{bmatrix} \tag{5.111}$$

where $\mathbf{o}_i^\top$, $i = 1, 2, \ldots, r$ represents a zero row vector with identical dimension to $\mathbf{L}(0)_i^\top$.

5.7 Cart and Table Linear Model Predictive Control

In this section, the LMPC approximated by Laguerre basis functions is used for generating walking patterns with the Cart and Table model. In addition, this framework can be also easily applied in the 3D-LIPM for the same task. JJ.Alcaraz-Jimenez et al. [59] have used the conventional LMPC with the Cart and Table model. Furthermore, P-B.Wieber [60] used the conventional LMPC with constraints on the gait. To this end,

A.Herdt et al. [61] added an automatic foot stepping feature, and D.Dimitrov et al [62] proposed an optimized solver for the constrained case. The choice of constrained LMPC, results in computationally heavy solutions, which are potentially inapplicable in practice due to typically limited computational resources for onboard execution. To this end, a sparse formulation has been proposed by D.Dimitrov et al [63], which results in larger optimization problems, but offers faster solutions. Unfortunately, all of these constrained LMPC approaches have been tested only in simulation.

5.7.1 LMPC using Laguerre Functions for the Cart and Table Model

As before, to derive the LMPC, a modified state space is used. The augmented state vector used, is given by

$$\mathbf{x}_e(k) = \begin{bmatrix} \mathbf{\Delta x}(k)^\top & z_x(k) \end{bmatrix}^\top \tag{5.112}$$

The Cart and Table model with the embedded integrator is

$$\mathbf{x}_e(k+1) = \overbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0}^\top \\ \mathbf{c}^\top\mathbf{A} & 1 \end{bmatrix}}^{\mathbf{A}_e} \mathbf{x}_e(k) + \overbrace{\begin{bmatrix} \mathbf{b} \\ \mathbf{c}^\top\mathbf{b} \end{bmatrix}}^{\mathbf{b}_e} \Delta u_x(k) \tag{5.113}$$

$$z_x(k) = \overbrace{\begin{bmatrix} \mathbf{0}^\top & 1 \end{bmatrix}}^{\mathbf{c}_e^\top} \mathbf{x}_e(k) \tag{5.114}$$

Let's use a set of Laguerre functions, $l_1(m), l_2(m), \ldots, l_N(m)$, with the corresponding Laguerre coefficients $\eta_{x_1}, \eta_{x_2}, \ldots, \eta_{x_N}$ to capture the dynamic response of the incremental control signal trajectory $\Delta u_x(k), \Delta u_x(k+1), \ldots, \Delta u_x(k+m), \ldots$, at discrete time k.

For an arbitrary future time sample m, with k being the start of the moving horizon window, the incremental control input will be given by:

$$\Delta u_x(k+m) = \sum_{j=1}^{N} \eta_{x_j}(k) l_j(m) = \mathbf{L}(m)^\top \boldsymbol{\eta}_x(k) \tag{5.115}$$

When using Eq.(5.115) in the modified state-space model Eq.(5.113), the prediction of the future state variable $\mathbf{x}_e(k+m|k)$ at sampling time m based on $\mathbf{x}_e(k)$ is given by:

$$\mathbf{x}_e(k+m|k) = \mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top \boldsymbol{\eta}_x(k) \tag{5.116}$$

and the predicted output at sampling time m is given by:

$$z_x(k+m|k) = \mathbf{c}_e^\top \mathbf{A}_e^m \mathbf{x}_e(k) + \sum_{i=0}^{m-1} \mathbf{c}_e^\top \mathbf{A}_e^{m-i-1} \mathbf{b}_e \mathbf{L}(i)^\top \boldsymbol{\eta}_x(k) \tag{5.117}$$

5.7.2 Cart and Table Unconstrained LMPC

The performance index to be minimized is given by

$$\underset{\boldsymbol{\Delta}\mathbf{U}_x(k)}{\text{minimize}} \;\; J_\text{u} = \|\mathbf{Z}_x^\text{ref}(k) - \mathbf{Z}_x(k)\|_2^2 + \|\boldsymbol{\Delta}\mathbf{U}_x(k)\|_\mathbf{R}^2 \tag{5.118}$$

where $\mathbf{R}$ is a diagonal positive definite matrix and

$$\begin{aligned}
\mathbf{Z}_x^\text{ref}(k) &= \begin{bmatrix} z_x^\text{ref}(k+1) & z_x^\text{ref}(k+2) & \cdots & z_x^\text{ref}(k+N_p) \end{bmatrix}^\top \\
\mathbf{Z}_x(k) &= \begin{bmatrix} z_x(k+1|k) & z_x(k+2|k) & \cdots & z_x(k+N_p|k) \end{bmatrix}^\top \\
\boldsymbol{\Delta}\mathbf{U}_x(k) &= \begin{bmatrix} \Delta u_x(k) & \Delta u_x(k+1) & \cdots & \Delta u_x(k+N_c-1) \end{bmatrix}^\top
\end{aligned}$$

Replacing each term $\Delta u_x(k+m)$ as in Eq.(5.115) and using the orthonormality of the Laguerre functions, the performance index Eq.(5.118) becomes equivalent to:

$$\underset{\boldsymbol{\eta}_x(k)}{\text{minimize}} \;\; J_\text{u} = \|\mathbf{Z}_x^\text{ref}(k) - \mathbf{Z}_x(k)\|_2^2 + \|\boldsymbol{\eta}_x(k)\|_{\mathbf{R_L}}^2 \tag{5.119}$$

Using Eq.(5.117) and $\mathbf{Z}_x(k)$ takes the form:

$$\mathbf{Z}_x(k) = \overbrace{\begin{bmatrix} \mathbf{c}_e^\top \mathbf{A}_e \\ \mathbf{c}_e^\top \mathbf{A}_e^2 \\ \vdots \\ \mathbf{c}_e^\top \mathbf{A}_e^{N_p} \end{bmatrix}}^{\mathbf{T}} \mathbf{x}_e(k) + \overbrace{\begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(N_p) \end{bmatrix}}^{\boldsymbol{\Phi}} \boldsymbol{\eta}_x(k) \tag{5.120}$$

Using the Theorem 5.5, the analytic solution to the unconstrained problem is:

$$\boldsymbol{\eta}_x(k) = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{R_L}\right)^{-1} \boldsymbol{\Phi}^\top \left(\mathbf{Z}_x^\text{ref}(k) - \mathbf{T}\mathbf{x}_e(k)\right) \tag{5.121}$$

Upon obtaining the optimal coefficients $\boldsymbol{\eta}_x(k)$, the receding horizon control law given by Eq.(5.95) is realized as:

$$\Delta u_x(k) = \mathbf{L}(0)^\top \boldsymbol{\eta}_x(k) \tag{5.122}$$

The proposed control scheme is shown in Figure 5.8. The same procedure can be followed for the y-axis. Figure 5.9 shows the x-axis response for a few omnidirectional steps.

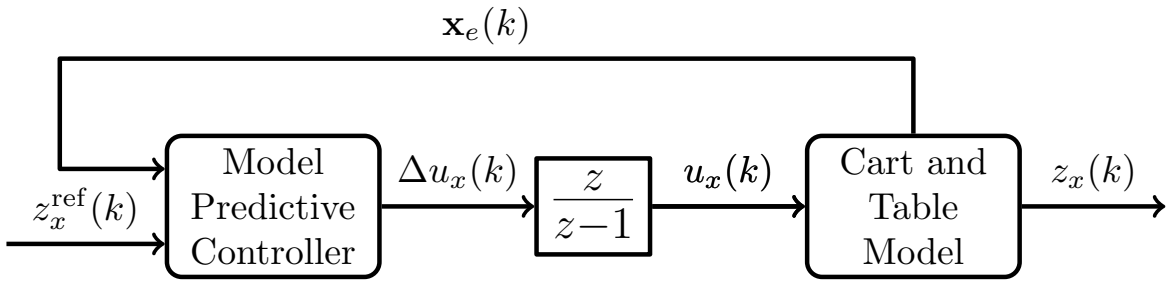

Figure 5.8: LMPC control scheme

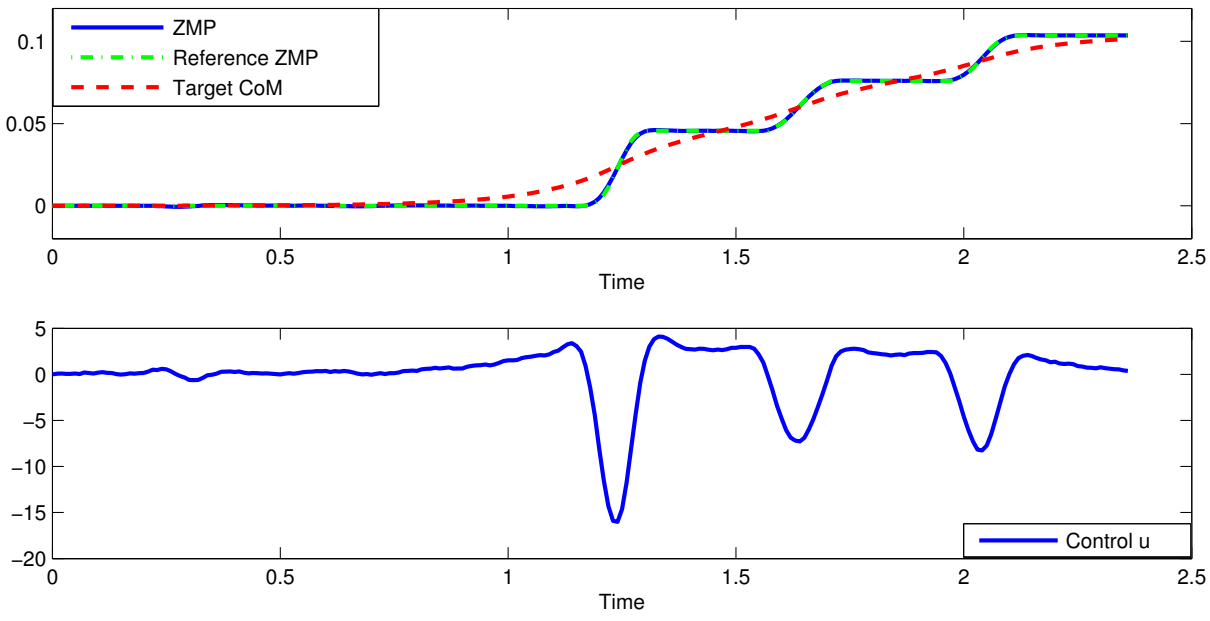

Figure 5.9: Unconstrained LMPC response for omindirectional steps (x-axis)

5.7.3 Cart and Table Constrained LMPC

A key feature of the LMPC is the ability to handle hard constraints in the design. In our case we will introduce constraints on the output of the system, namely the ZMP realized by the system dynamics will be constrained to lie in the support polygon defined separately in SS and DS phase. When walking omnidirectionally, the x and y axes cannot be decoupled in the optimization procedure, because of the coupling imposed by the constraints. Therefore, the complete problem must be solved once every sampling time, leading to a MIMO LTI dynamic system. The performance index to be minimized takes the following form:

$$J_{\mathrm{c}} = \boldsymbol{\eta}(k)^{\top}\mathbf{H}\,\boldsymbol{\eta}(k) + \mathbf{f}^{\top}(k)\boldsymbol{\eta}(k) \tag{5.123}$$

where, using the same Laguerre basis for both inputs u_x, u_y,

$$\boldsymbol{\eta}(k) = \begin{bmatrix}\boldsymbol{\eta}_x(k)\\ \boldsymbol{\eta}_y(k)\end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix}\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \mathbf{R}_{\mathbf{L}} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \mathbf{R}_{\mathbf{L}}\end{bmatrix}$$

$$\mathbf{f}(k) = \begin{bmatrix}2\left(\mathbf{T}\mathbf{x}_e(k) - \mathbf{Z}_x^{\mathrm{ref}}(k)\right)^{\top}\boldsymbol{\Phi}\\ 2\left(\mathbf{T}\mathbf{y}_e(k) - \mathbf{Z}_y^{\mathrm{ref}}(k)\right)^{\top}\boldsymbol{\Phi}\end{bmatrix}$$

During the SS phase, the support polygon is defined by the supporting foot and remains unchanged, whereas, during the DS phase, it is the convex hull of the feet and cannot be easily defined. Therefore, at each DS sampling time k, it is approximated as a moving support foot centered at $\left(z_x^{\mathrm{ref}}(k), z_y^{\mathrm{ref}}(k)\right)$ with orientation changing from the previous support foot to the next one using some interpolation method, as shown in Figure 5.10. Therefore, the constraints on the ZMP for both phases take the form:

$$\begin{bmatrix}\mathbf{I}\\ -\mathbf{I}\end{bmatrix}\mathbf{R}_z(\theta_i)^{\top}\begin{bmatrix}z_x(k+m|k) - z_x^{\mathrm{ref}}(k+m)\\ z_y(k+m|k) - z_y^{\mathrm{ref}}(k+m)\end{bmatrix} \leq \frac{1}{2}\begin{bmatrix}s_x\\ s_y\\ s_x\\ s_y\end{bmatrix}$$

for $m = 1, \ldots, N_p$, where $\mathbf{I}$ is the 2×2 identity matrix, s_x and s_y are the length and width of the foot respectively, and $\mathbf{R}_z(\theta_i) \in SO(2)$, is a rotation matrix of angle θ_i about the vertical z-axis at the i-th moment of the prediction horizon, while $SO(2)$ stands for the 2-dimensional Special Orthogonal group. Using Eq.(5.88) we can express the constraints

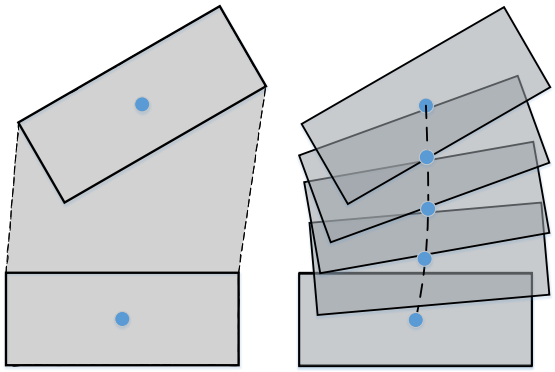

Figure 5.10: DS convex hull approximated by a series of SS polygons

in terms of the Laguerre coefficients, forming a Quadratic Program (QP) with linear constraints (dropping the index k for clarity):

$$\begin{aligned} \underset{\boldsymbol{\eta}}{\text{minimize}} \quad & J_{\text{c}} = \boldsymbol{\eta}^{\top}\mathbf{H}\,\boldsymbol{\eta} + \mathbf{f}^{\top}\boldsymbol{\eta} \\ \text{subject to} \quad & \mathbf{A}_{\text{c}}\,\boldsymbol{\eta} \leq \mathbf{b}_{\text{c}} \end{aligned} \tag{5.124}$$

where

$$\mathbf{A}_{\text{c}} = \begin{bmatrix} \mathbf{E}\Phi & \mathbf{F}\Phi \\ -\mathbf{F}\Phi & \mathbf{E}\Phi \\ -\mathbf{E}\Phi & -\mathbf{F}\Phi \\ \mathbf{F}\Phi & -\mathbf{E}\Phi \end{bmatrix}$$

$$\mathbf{b}_{\text{c}} = \begin{bmatrix} \mathbf{S}_x - \mathbf{E}(\mathbf{T}\mathbf{x}_e^x(k) - \mathbf{Z}_x^{\text{ref}}(k)) - \mathbf{F}(\mathbf{T}\mathbf{x}_e^y(k) - \mathbf{Z}_y^{\text{ref}}(k)) \\ \mathbf{S}_y + \mathbf{F}(\mathbf{T}\mathbf{x}_e^x(k) - \mathbf{Z}_x^{\text{ref}}(k)) - \mathbf{E}(\mathbf{T}\mathbf{x}_e^y(k) - \mathbf{Z}_y^{\text{ref}}(k)) \\ \mathbf{S}_x + \mathbf{E}(\mathbf{T}\mathbf{x}_e^x(k) - \mathbf{Z}_x^{\text{ref}}(k)) + \mathbf{F}(\mathbf{T}\mathbf{x}_e^y(k) - \mathbf{Z}_y^{\text{ref}}(k)) \\ \mathbf{S}_y - \mathbf{F}(\mathbf{T}\mathbf{x}_e^x(k) - \mathbf{Z}_x^{\text{ref}}(k)) + \mathbf{E}(\mathbf{T}\mathbf{x}_e^y(k) - \mathbf{Z}_y^{\text{ref}}(k)) \end{bmatrix}$$

$$\mathbf{S}_x = \tfrac{1}{2}\begin{bmatrix} s_x & \dots & s_x \end{bmatrix}^{\top} \qquad \mathbf{S}_y = \tfrac{1}{2}\begin{bmatrix} s_y & \dots & s_y \end{bmatrix}^{\top}$$

$$\mathbf{E} = \begin{bmatrix} \cos(\theta_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \cos(\theta_{N_p}) \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \sin(\theta_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sin(\theta_{N_p}) \end{bmatrix}$$

We will utilize a penalty method [64] for solving the QP shown in Eq.(5.124), which will be short presented in the next section.

5.7.4 Penalty Methods

A penalty method replaces a constrained optimization problem by a series of unconstrained problems, whose solutions ideally converge to the solution of the original constrained problem. The unconstrained problems are formed by adding a term, called a penalty function, to the performance index that consists of a penalty parameter multiplied by a measure of violation of the constraints. The measure of violation is nonzero when the constraints are violated and is zero in the region where constraints are not violated. An important difference with the logarithmic barrier methods (S.Boyd et al. [65]) is that in this case the iterates are forced to remain interior to the feasible domain and the barrier is in place to bias the iterates to remain away from the boundary of the feasible region, here there is not such requirement.

Assume the convex optimization problem, with $f(\mathbf{x}) \in domf$ and $f_i(\mathbf{x}) \in domf_i$ with $i = 1, \ldots, m$

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0 \;\; i = 1, \ldots, m \end{aligned} \tag{5.125}$$

where $f(\mathbf{x})$ and $f_i(\mathbf{x})$ for $i = 1, \ldots, m$ are convex functions.
The general case of a Penalty method is given by

$$\text{minimize} \quad f(\mathbf{x}) + \frac{1}{t}\sum_{i=1}^{m} P(f_i(\mathbf{x})) \tag{5.126}$$

where t is a positive constant called the penalty coefficient and $P(f_i(\mathbf{x}))$ is the penalty function with $P(f_i(\mathbf{x})) \geq 0$ for every $\mathbf{x} \in domf$ and $P(f_i(\mathbf{x})) = 0$ only for every $\mathbf{x}$ satisfying the constraints. A commonly chosen penalty function is

$$p(f_i(\mathbf{x})) = \max(0, f_i(\mathbf{x}))^2 \tag{5.127}$$

The generic Penalty method is described in algorithm 1, where the Newton step $\mathbf{\Delta x_{Nt}} = (\nabla^2 J)^{-1}\nabla J$, the Newton decrement $\lambda(\mathbf{x_k}) = \nabla J^\top (\nabla^2 J)^{-1}\nabla J$, ϵ_{inner} and ϵ_{outer} is the inner's loop and the outer's loop accuracy, respectively.

In our case, the performance index to be minimized is given by Eq.(5.124), using the penalty method the index to be minimized is

$$J_0 = J_\mathrm{c} + \frac{1}{t}\sum_{i=1}^{4N_p} \max\left(0, \left(\mathbf{a}_{\mathrm{c}_i}^\top \boldsymbol{\eta} - b_{\mathrm{c}_i}\right)\right)^2 \tag{5.128}$$

Algorithm 1 Penalty Method

while (TRUE) **do**
  while (TRUE) **do**
    1.Compute Newton Step $\mathbf{\Delta x_{Nt}}$
    if $\lambda(\mathbf{x_k}) < \epsilon_{inner}$ **then** break
    end if
    2. Backtracking Line Search
    3. $\mathbf{x_{k+1}}$:=$\mathbf{x_k}$+$\tau_{bt}\mathbf{\Delta x_{Nt}}$
  end while
  if $(m/t) > \epsilon_{outer}$ **then** break
  end if
  4. $t = \mu t$
end while
return x

where $\mathbf{a}_{c_i}^{\top}$ is the i-th row of $\mathbf{A_c}$ and b_{c_i} the i-th element of $\mathbf{b_c}$. Assuming the problem is feasible, Algorithm 2 finds a solution efficiently by iteratively identifying violated.

Algorithm 2 Naive approach with Quadratic Penalty

while (TRUE) **do**
  while (TRUE) **do**
    compute Newton Step $\mathbf{\Delta \eta_{Nt}}$
    if $\lambda(\boldsymbol{\eta}_\mathbf{k}) < \epsilon_{inner}$ **then** break
    end if
    Backtracking Line Search
    $\boldsymbol{\eta}_{\mathbf{k+1}}$:=$\boldsymbol{\eta}_\mathbf{k}$+$\tau_{bt}\mathbf{\Delta \eta_{Nt}}$
  end while
  if $(4N_p/t) > \epsilon_{outer}$ **then** break
  end if
  4. $t = \mu t$
end while
return $\boldsymbol{\eta}$

If we exploit the fact that active constraints just add a quadratic term to the performance index, an analytical solution of the inner loop can be obtained. Thus, the

computations are less, since there is no need to compute the Newton decrement anymore. This approach is shown in algorithm 3.

Algorithm 3 Analytic Approach with Quadratic Penalty

while (TRUE) **do**
  compute $\boldsymbol{\eta} = -(2\mathbf{H})^{-1}\mathbf{f}$
  find i such that $\mathbf{a}_{c_i}^{\top}\boldsymbol{\eta} - b_{c_i} > 0$ (violated constraint)
  if i exists **then** $\mathbf{H} = \mathbf{H} + \frac{1}{t}\mathbf{a}_{c_i}\mathbf{a}_{c_i}^{\top}$ and $\mathbf{f} = \mathbf{f} - \frac{1}{t}(2b_{c_i}\mathbf{a}_{c_i})$
  else break
  end if
  if $(4N_p/t) > \epsilon_{outer}$ **then return** $\boldsymbol{\eta}$
  end if
  $t = \mu t$
end while

Finally we noticed that the use of a quadratic penalty term was not mandatory. Therefore, we replaced the quadratic penalty with a linear one

$$J_0 = J_c + \frac{1}{t}\sum_{i=1}^{4N_p} \max\left(0, \left(\mathbf{a}_{c_i}^{\top}\boldsymbol{\eta} - b_{c_i}\right)\right) \tag{5.129}$$

In such a way, the complexity of the algorithm is reduced significantly. If a constraint is active, then a linear term is added to the performance index. Thus, for computing the optimal solution in every time step the hessian $\mathbf{H}$ is not need to be inverted, only a matrix multiplication is needed after correctly adapting the linear term $\mathbf{f}$. This approach is demonstrated on algorithm 4 and was found very effective in practice. Upon obtaining the optimal vector $\boldsymbol{\eta}$ for sampling time k, we can apply the receding horizon law (Eq.(5.111)), and compute the incremental optimal control for the discrete time k as follows:

$$\Delta u(k) = \begin{bmatrix} \mathbf{L}(0)^{\top} & \mathbf{0}^{\top} \\ \mathbf{0}^{\top} & \mathbf{L}(0)^{\top} \end{bmatrix} \boldsymbol{\eta}(k) \tag{5.130}$$

The constrained LMPC and the system response are shown in Figure 5.11, when a disturbance is encountered in the y-axis.

Algorithm 4 Analytic Approach with Linear Penalty

while (TRUE) **do**
  compute $\boldsymbol{\eta} = -(2\mathbf{H})^{-1}\mathbf{f}$
  find i such that $\mathbf{a}_{c_i}^{\top}\boldsymbol{\eta} - b_{c_i} > 0$ (violated constraint)
  if i exists **then** $\mathbf{f} = \mathbf{f} + \frac{1}{t}a_{c_i}$
  else break
  end if
  if $(4N_p/t) > \epsilon_{outer}$ **then return** $\boldsymbol{\eta}$
  end if
  $t = \mu t$
end while

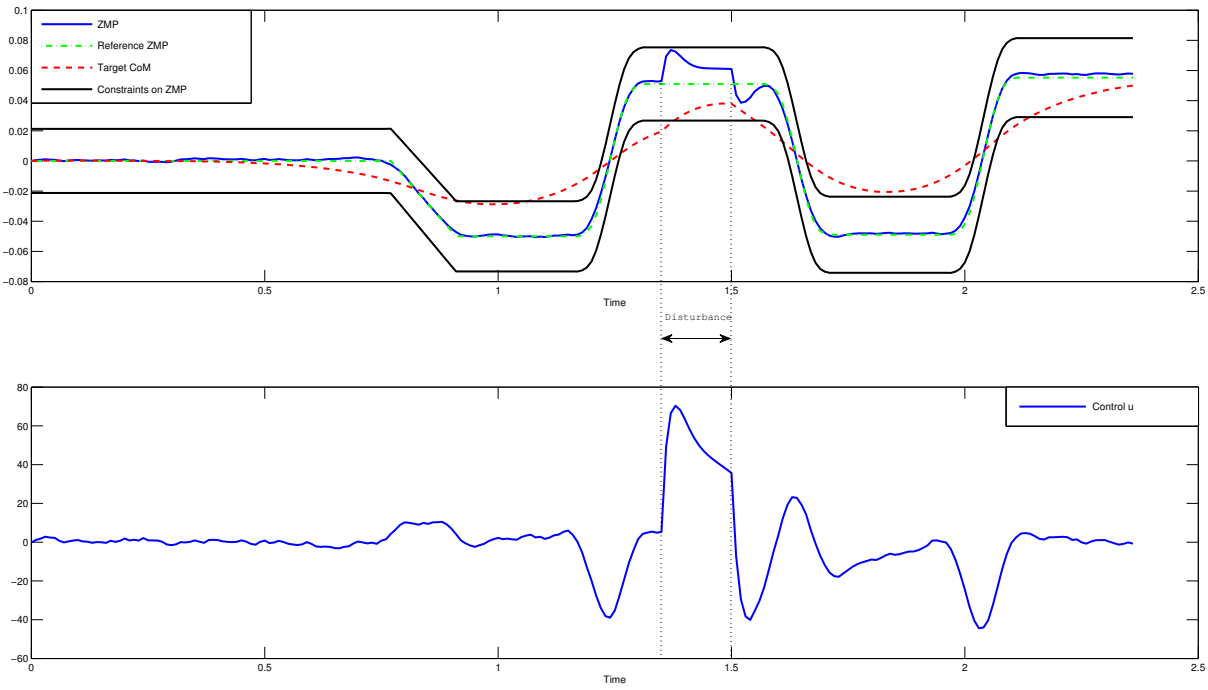

Figure 5.11: Constrained LMPC response for a constant disturbance (y-axis).

5.8 Conclusion

In this chapter a variety of control methods were presented and later applied in the walking pattern generation task using simplified dynamic models. The state of the art LQR technique was abstractly presented, followed by the preview control allowing future

reference information into the control design framework. For the state estimation task, the Luenberger observer was presented, and an optimal state estimator was derived by treating the estimation problem as the dual of a control problem. Next the preview control for the 3D-LIPM was derived augmented with an observer. The same procedure was followed for the Cart and Table model, where also the concept of the Auxiliary ZMP was introduced, enchanting the preview control scheme with the ability to overcome disturbances and/or the presence of the uneven terrain by correcting online the walking patterns. Furthermore the LMPC of the Cart and Table model approximated by an orthonormal basis set was proposed. In such a way the dimensionality of the optimization problem was reduced significantly allowing faster computation, as it is necessary for real-time control systems. In addition constraints were applied on the ZMP during the gait, reinforcing the overall dynamic stability. Thus, strong disturbances can be endured and walking over rough ground becomes viable. Finally, the constrained LMPC problem was efficiently solved with a Penalty method yielding fast and accurate results.

Chapter 6

Feet Trajectory Planning

As long as Algebra and Geometry were separate subjects, their progress was slow and their uses limited, but when these two sciences united, they have lent each other their forces, and have since marched together towards perfection.

Joseph Louis Lagrange (1736–1813)

In this chapter, we will address the problem of feet trajectory generation. Once the CoM patterns are determined, a proper feet trajectory must be generated to achieve a smooth gait. The problem of finding smooth motions that interpolate a given set of positions and orientations is very frequently encountered in the field of robotics. Therefore, we seek techniques for planning smooth reference feet trajectory. Mainly, three methods will be presented and discussed. In Section 6.1, the cubic Bezier interpolation is outlined. Further, in Section 6.2, the spline interpolation method is briefly presented, focusing in the cubic B-spline method. In Section 6.3, rigid body interpolation techniques are presented. First, a brief introduction to Lie groups and Lie algebra, along with some important notions of differential geometry, is given, then the geodesic curves for rigid bodies are computed. However, the geodesic curves are not suitable trajectories for smooth human-like walking. To this end, the minimal jerk curves are generated, which yield smooth feet trajectory. All methods discussed, are applied to a simple gait example. Finally, an efficient and computationally inexpensive method for adaptively correcting online the reference feet trajectory is outlined.

6.1 Cubic Bezier Interpolation

We are interested in planning reference trajectories for the feet in the x, y, z axes with respect to an inertial reference frame and also a trajectory for the yaw angle of each foot. For the feet trajectory in the x and y axes along with the yaw angle trajectory, a simple linear interpolation can yield an acceptable walking performance in practice. However, for the foot trajectory in the z axis, we need a smoother trajectory. For this purpose, we will adopt the Bezier curves. Bezier is one of the most influential polynomials and an important tool for interpolation. The Bezier interpolating curve always lies within a convex hull and it never oscillates wildly away from the control points. Bezier polynomial has several applications in the fields of engineering, science, and technology, such as highway or railway route designing, networks, computer-aided design systems, animation, environment design, robotics, communications, and many other disciplines, because it is easy to compute and is also very stable. One of the main approaches to robot motion is through the use of cubic Bezier spline functions.

Let $\mathbf{p}_i = (x_i, y_i)$ and $i = 0, 1, \ldots, n$ be the control points of the Bezier curve. The Bezier curve of degree n is given by

$$\mathbf{p}(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{p}_i \tag{6.1}$$

with $t \in [0, 1]$ and $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$, $i = 0, 1, 2, \ldots, n$, being the Bernstein polynomials of degree n. For example, using the cubic Bezier interpolation method with four control points, $\mathbf{p}_0 = [0, 0]$, $\mathbf{p}_1 = \left[0.02, \frac{1}{3}T_{ss}\right]$, $\mathbf{p}_2 = \left[0.02, \frac{2}{3}T_{ss}\right]$, $\mathbf{p}_3 = [0, T_{ss}]$, where T_{ss} is the duration of the SS phase, we can generate a smooth trajectory $\mathbf{d}_z^{\text{ref}}$ for the swinging leg, which yields a maximum height of 0.015m.

$$\mathbf{d}_z^{\text{ref}}(t) = (1-t)^3\mathbf{p}_0 + 3(1-t)^2 t\mathbf{p}_1 + 3(1-t)t^2\mathbf{p}_2 + t^3\mathbf{p}_3 \tag{6.2}$$

A walking gait produced by cubic Bezier interpolation is shown in Figure 6.1, where the smooth trajectories of the feet are obvious.

6.2 Spline Interpolation

As presented in the previous section, Bezier curves can yield a smooth trajectory for the swinging leg in the SS phase. However in practice, the foot does not make contact with the

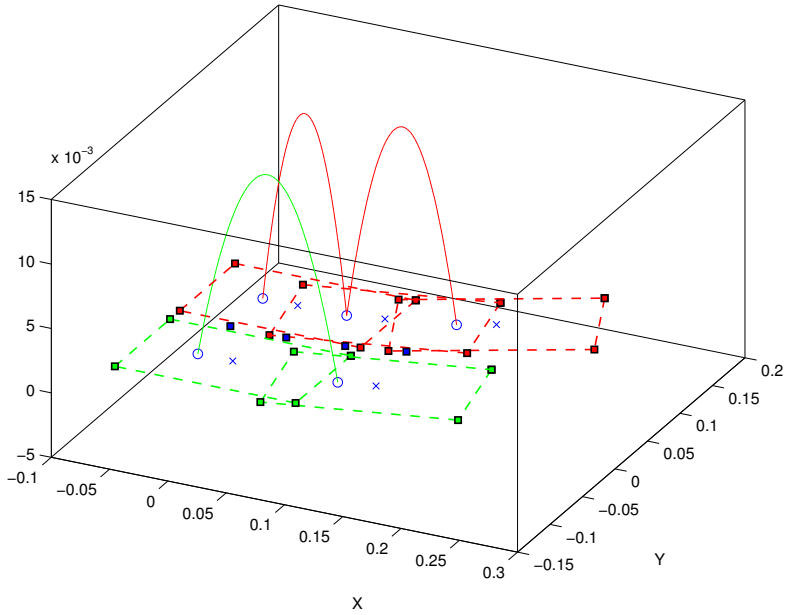

Figure 6.1: Feet trajectory using Bezier interpolation with four control points

ground smoothly. To compensate for this phenomenon, the cubic B-spline interpolation method was used. Working with five control points, a smoother foot motion was achieved, both at the start and at the end of the SS phase. We will first give a brief description of splines and then use it for interpolating the foot trajectory in the z-axis.

Given a tabulated function $f_k = f(x_k)$, $k = 0, ..., N,$, a spline is a polynomial between each pair of tabulated points, but one whose coefficients are determined "slightly" non-locally. The non-locality is designed to guarantee global smoothness in the interpolated function up to some order of derivative.

6.2.1 Cubic Spline

Cubic splines are very popular, as they produce an interpolated function, which is continuous through to the second derivative. Splines tend to be stabler than fitting a polynomial through the $N + 1$ points, with less possibility of wild oscillations between the tabulated points. In general, if the function to be approximated is smooth, then the cubic spline will do better than any piecewise linear interpolation.

In each interval $[x_k, x_{k+1}]$, we can fit a straight line through the points (x_k, f_k) and

(x_{k+1}, f_{k+1}) using the formula

$$f = af_k + bf_{k+1} \tag{6.3}$$

where the coefficients a, b are given by

$$a = \frac{x_{k+1} - x}{x_{k+1} - x_k}, \quad b = 1 - a = \frac{x - x_k}{x_{k+1} - x_k} \tag{6.4}$$

The problem is that with the linear function, the first derivative is not continuous at the boundary between two adjacent intervals, while we want the second derivative to be continuous, even at the boundary.

Now suppose, that in addition to the tabulated values of f_i, we also have tabulated values for the function's second derivatives, that is, a set of numbers f_i''. Then, within each interval $[x_k, x_{k+1}]$, we can add to the right-hand side of Eq.(6.3) a cubic polynomial, whose second derivative varies linearly from a value f_k'' on the left to a value f_{k+1}'' on the right. Doing so, we will have the desired continuous second derivative. If we also construct the cubic polynomial to have zero values at x_k and x_{k+1}, then adding it into f does not spoil the agreement with the tabulated functional values f_k and f_{k+1} at the endpoints x_k and x_{k+1}. Therefore, Eq.(6.3) is replaced by

$$f = af_k + bf_{k+1} + cf_k'' + df_{k+1}'' \tag{6.5}$$

where coefficients a, b are defined as in Eq.(6.4) and c, d are given by

$$c = \frac{1}{6}(a^3 - a)(x_{k+1} - x_k)^2, \quad d = \frac{1}{6}(b^3 - b)(x_{k+1} - x_k)^2 \tag{6.6}$$

At this point, we have to note that the coefficients a, b are linearly dependent on x, while c, d have cubic dependence on x (through a and b). We can check that f'', the second derivative of the interpolated function, is indeed continuous. The first derivative is given by

$$f' = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} - \frac{3a^2 - 1}{6}(x_{k+1} - x_k)f_k'' + \frac{3b^2 - 1}{6}(x_{k+1} - x_k)f_{k+1}'' \tag{6.7}$$

and the second derivative by

$$f'' = af_k'' + bf_{k+1}'' \tag{6.8}$$

Since $a = 1$, $b = 0$ at x_k and $a = 0$, $b = 1$ at x_{k+1} respectively, Eq.(6.8) shows that f'' is just the tabulated second derivative at the end points of the interval, and also that f''

will be continuous across the boundary between two intervals, in this case $[x_{k-1}, x_k]$ and $[x_k, x_{k+1}]$.

For the formulation presented above, we required that the f''_k for $k = 0, \ldots, N$ are known, but usually this is not the case. To obtain the values of f''_k's, we need to evaluate Eq.(6.7) for $x = x_k$ in the interval $[x_{k-1}, x_k]$ and set it to be equal with the same equation evaluated for $x = x_k$, but in the interval $[x_k, x_{k+1}]$. With some rearrangement, this results to

$$\frac{x_k - x_{k-1}}{6} f''_{k-1} + \frac{x_{k+1} - x_{k-1}}{3} f''_k + \frac{x_{k+1} - x_k}{6} f''_{k+1} = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} - \frac{f_k - f_{k-1}}{x_k - x_{k-1}} \tag{6.9}$$

These are $N - 1$ linear equations in the $N + 1$ unknown functions f''_i, $i = 0, \ldots, N$. Therefore, there is a two-parameter family of possible solutions. For a unique solution, we need to specify further conditions, typically taken as boundary conditions at x_0 and x_N. The most common choice is to set f''_0 and f''_N equal to zero, then the so-called natural cubic spline is obtained.

6.2.2 Cubic B-Splines

In the previous section, we started with a functional form for the interpolation formula (Eq.(6.5)) and had to use the constraint that f' is continuous at interval boundaries to solve for f''. In this section, we will provide an approach based on a set of cubic polynomials defined on some sub-intervals of $[x_0, x_N]$, which are by construction continuous through to second derivative at the boundaries of intervals. They will form a set of basis functions, since linear combinations of these functions will also satisfy the continuity properties at the boundaries between adjacent intervals. To construct the cubic spline over the whole range $[x_0, x_N]$, we will simply need to match the sum of the basis functions with tabulated values of f_i at the interpolating nodes x_i, $i = 0, \ldots, N$.

The B-splines are the basis functions that satisfy the continuity conditions mentioned. For node x_0, the corresponding basis B_0 is defined by

$$B_0(x) = \begin{cases} 0 & \text{if } x \le x_0 - 2h \\ 2h + (x - x_0) & \text{if } x_0 - 2h \le x \le x_0 - h \\ -2h - 3(x - x_0) & \text{if } x_0 - h \le x \le x_0 \\ -2h + 3(x - x_0) & \text{if } x_0 \le x \le x_0 + h \\ 2h - (x - x_0) & \text{if } x_0 + h \le x \le x_0 + 2h \\ 0 & \text{if } x \ge x_0 + 2h \end{cases} \tag{6.10}$$

where $h = x_{k+1} - x_k = \frac{x_N - x_0}{N}$ is the width between interpolating nodes, which in this case is assumed to be equal. The cubic spline function $S(x)$ over a region $[x_0, x_N]$ is then written as the linear combination of the bases B_k

$$S(x) = \sum_{k=-1}^{N+1} a_k B_k(x) \tag{6.11}$$

The sum is from -1 to $N+1$ since B_{-1} is nonzero in the interval $[x_0, x_1]$, and B_{N+1} is nonzero in the interval $[x_{N-1}, x_N]$, so we need to take into account these functions.

The parameters a_k, $k = 0, \ldots, N$, are given by solving the following matrix equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 1 & 4 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 4 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \\ a_N \end{bmatrix} = \frac{1}{h^3} \begin{bmatrix} f(x_0) \\ 6f(x_1) \\ 6f(x_2) \\ \vdots \\ 6f(x_{N-1}) \\ f(x_N) \end{bmatrix} \tag{6.12}$$

This set of equations is tridiagonal and can be solved in $O(N)$ operations by a tridiagonal algorithm. The last two coefficients to completely determine $S(x)$ are given by

$$a_{-1} = 2a_0 - a_1 \tag{6.13}$$

$$a_{N+1} = 2a_N - a_{N-1} \tag{6.14}$$

In our case, we are using 5 points for computing a smooth trajectory for the swinging foot in the z-axis. The points selected were $(0, 0)$, $(T_{ss}/4, S_z/2)$, $(T_{ss}/2, S_z)$, $(3T_{ss}/4, S_z/2)$, $(T_{ss}, 0)$. The coefficients a_k for $k = -1, \ldots, 6$ were found according to Eqs.(6.12),(6.13), $h = T_{ss}/5$, and the reference trajectory is given by

$$d_z^{\text{ref}}(t) = \sum_{i=-1}^{6} a_i B_i(t) \tag{6.15}$$

A gait example is shown in Figure (6.2), where a smooth foot take off and a smooth foot landing is achieved.

6.3 Rigid Body Interpolation

All previous interpolation methods treat the foot as a single point object and solve each axis separately to compute a trajectory. We can treat the feet as rigid bodies and

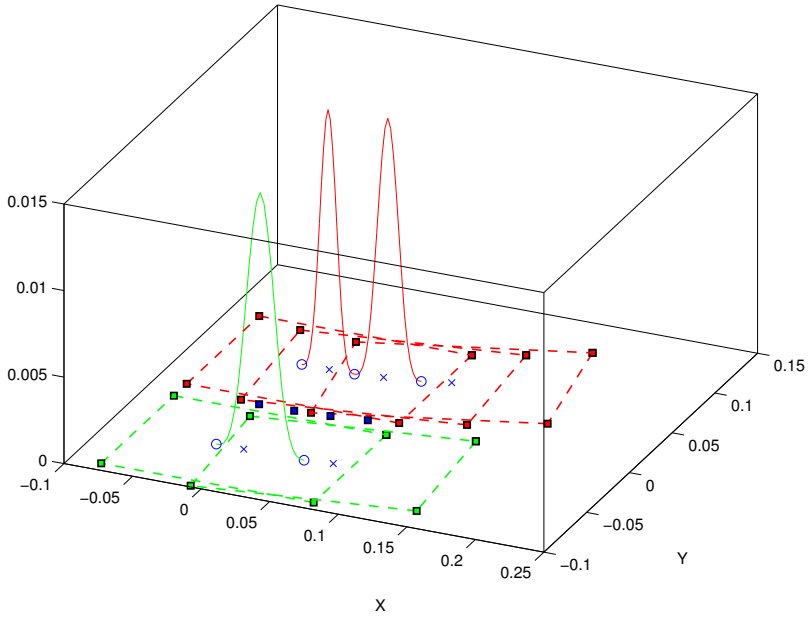

Figure 6.2: Feet trajectory using Cubic B-Spline interpolation with five points

generate trajectories in $SE(3)$, between an initial and a final position and orientation. Such rigid body interpolation methods have been proposed by M. Zefran et al. [66]. The methods presented below can be used in a variety of applications, not only for walking, i.e. to plan movements between any end-effector starting position and orientation and a desired position and orientation. In robotics, smooth trajectories are desired, since the electro-mechanical system is limited by its actuator size and its control bandwidth, thus it cannot produce large velocities and accelerations and because movements with high accelerations and/or jerks can excite the structural natural frequencies of the system. Trajectories generated will maximize an appropriate measure of smoothness, in the form of an integral cost function. Depending on the chosen integrand, boundary conditions on the derivatives of the desired order can be enforced. Two interpolation methods on $SE(3)$ will be presented, the geodesic and the minimum-jerk trajectories.

6.3.1 Lie Groups and Lie Algebra

Consider a rigid body moving in free space. Assume an inertial reference frame F fixed in space and a frame M fixed to the moving rigid body, as shown in Figure 6.3. At each time instance, the configuration of the rigid body can be described by a homogeneous

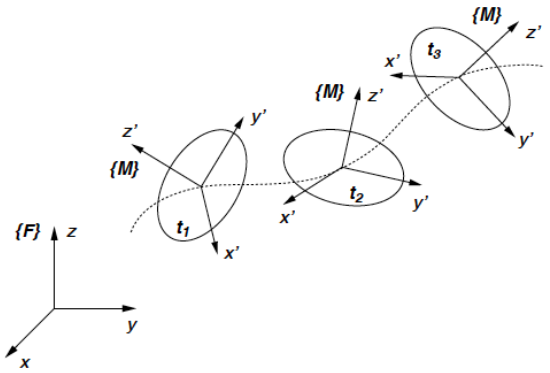

Figure 6.3: A rigid body moving in ambient space [8]

transformation matrix, corresponding to the displacement from frame F to frame M. The set of all such transformation matrices $\mathbf{A}$ is called the Special Euclidian group of rigid body transformations in the three dimensions, $SE(3)$.

$$SE(3) = \left\{ \mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R} \in \mathbb{R}^{3\times3}, \mathbf{d} \in \mathbb{R}^3, \mathbf{R} = \mathbf{R}^\top, det(\mathbf{R}) = 1 \right\} \tag{6.16}$$

The $SE(3)$ group is also a Lie Group, since it is a closed subset of $GL(4, \mathbb{R})$, and therefore the tangent space at the group identity has the structure of a Lie algebra. The Lie algebra of $SE(3)$, denoted by $se(3)$, is given by:

$$se(3) = \left\{ \mathbf{S} = \begin{bmatrix} \mathbf{\Omega} & \boldsymbol{v} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{\Omega} \in \mathbb{R}^{3\times3}, \boldsymbol{v} \in \mathbb{R}^3, \mathbf{\Omega} = -\mathbf{\Omega}^\top \right\} \tag{6.17}$$

Each 3×3 skew-symmetric matrix $\mathbf{\Omega}$ can be uniquely identified by a vector $\boldsymbol{\omega} \in \mathbb{R}^3$, so that for an arbitrary vector $\mathbf{x} \in \mathbb{R}^3$

$$\mathbf{\Omega x} = \boldsymbol{\omega} \times \mathbf{x} \tag{6.18}$$

with $\times$ being the vector cross product operation in $\mathbb{R}^3$. Physically $\boldsymbol{\omega}$ is the angular velocity of the rigid body, corresponding to the angular velocities about the x-axis, the y-axis, and z-axis of the moving frame M, while $\boldsymbol{v}$ corresponds to the translational velocity of the origin of frame M along the same axes. Thus, an element $\mathbf{S} \in se(3)$, called twist, can be uniquely identified by the pair of velocities $(\boldsymbol{\omega}, \boldsymbol{v})$. The Lie algebra $se(3)$ is isomorphic to the set of all twists.

Given a motion of a rigid body, described by a curve $\mathbf{A}(t) : [-a, a] \mapsto SE(3)$, a twist $\mathbf{S}(t)$ can be associated to the tangent vector $\dot{\mathbf{A}}(t)$ at each point t by:

$$\mathbf{S}(t) = \mathbf{A}(t)^{-1}\dot{\mathbf{A}}(t) = \begin{bmatrix} \mathbf{R}^\top\dot{\mathbf{R}} & \mathbf{R}^\top\dot{\mathbf{d}} \\ \mathbf{0} & 0 \end{bmatrix} \tag{6.19}$$

The computed twists consist of velocities that physically correspond to the angular velocity of the rigid body and translational velocity of the point on the rigid body that is coincident with the origin of the frame M, both expressed in the frame M. Such a twist is said to be obtained by the left translation of the tangent vector $\dot{\mathbf{A}}$ and does not depend on the choice of the inertial frame F.

Since $se(3)$ is a vector space, any elements can be expressed as a 6×1 vector of components corresponding to a chosen basis. A set of basis twists, which correspond to instantaneous rotations about and instantaneous translations along the Cartesian axes x, y, and z, given by

$$\mathbf{L}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{L}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.20}$$

will be used as a basis for $se(3)$. This twist basis has the property that any twist $\mathbf{S} \in se(3)$ is expressed using the components of vectors $(\boldsymbol{\omega}, \boldsymbol{v})$ as coefficients.

A smooth assignment of a tangent vector to each point of $SE(3)$ is called a *vector field*. One possible way to define a vector field, $\mathbf{X}$, at an arbitrary element $\mathbf{A} \in SE(3)$ is

$$\mathbf{X}(\mathbf{A}) = \hat{\mathbf{S}}(\mathbf{A}) = \mathbf{AS} \tag{6.21}$$

where $\mathbf{S} \in se(3)$ is a twist. Such vector fields are called left-invariant vector fields. We use the notation $\hat{\mathbf{S}}$ to indicate that the vector field was obtained by left-translating the Lie algebra element $\mathbf{S}$. There is a one-to-one correspondence between the left-invariant vector fields and elements of $se(3)$. Since $\mathbf{L}_1$, $\mathbf{L}_2$, ..., $\mathbf{L}_6$ are basis for $se(3)$, a choice of basis for the set of the left-invariant vector fields is $\hat{\mathbf{L}}_1$, $\hat{\mathbf{L}}_2$, ..., $\hat{\mathbf{L}}_6$, which can be obtained by Eq.(6.21). Therefore, at any point in $SE(3)$, the corresponding vector fields $\hat{\mathbf{L}}_1$, $\hat{\mathbf{L}}_2$,

..., $\hat{\mathbf{L}}_6$ are linearly independent, since they form a basis. So, any vector field $\mathbf{X}$ can be expressed as

$$\mathbf{X} = \sum_{i=1}^{6} X^i \hat{\mathbf{L}}_i \tag{6.22}$$

where the coefficients X^i vary over the manifold. If they are assumed to be constant, then the vector field $\mathbf{X}$ is called left-invariant. For each vector field $\mathbf{X}$, there exists an associated vector pair $(\boldsymbol{\omega}, \boldsymbol{v})$, defined by

$$\boldsymbol{\omega} = \begin{bmatrix} X^1 \\ X^2 \\ X^3 \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} X^4 \\ X^5 \\ X^6 \end{bmatrix} \tag{6.23}$$

Having defined the manifold, we are interested in carrying out a dynamic analysis over the moving manifold. To this end, we need to introduce an appropriate metric.

Definition 6.1 (Riemannian Metric)**.** *A Riemannian metric on $SE(3)$ can be defined as an inner product of elements in $se(3)$ [67]. Assuming two elements $\mathbf{S}_1, \mathbf{S}_2 \in se(3)$, the inner product at the identity element $\mathbf{I} \in SE(3)$ is given by*

$$< \mathbf{S}_1, \mathbf{S}_2 >_{\mathbf{I}} = \mathbf{s}_1^{\top} \mathbf{W} \mathbf{s}_2 \tag{6.24}$$

where $\mathbf{W}$ is positive definite matrix and the vectors $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^6$ and are given by

$$\mathbf{s}_1 = \mathbf{S}_1^{\vee} \tag{6.25}$$

$$\mathbf{s}_2 = \mathbf{S}_2^{\vee} \tag{6.26}$$

where $\vee$ is the vee operation of a twist $\mathbf{S} \in se(3)$.

Definition 6.2 (Left-Invariant Riemannian Metrics)**.** *If $\mathbf{V}_1$ and $\mathbf{V}_2$ are tangent vectors at an arbitrary point $\mathbf{A} \in SE(3)$, the inner product $< \mathbf{V}_1, \mathbf{V}_2 >_{\mathbf{A}}$ in the tangent space $T_{\mathbf{A}}SE(3)$ can be defined by*

$$< \mathbf{V}_1, \mathbf{V}_2 >_{\mathbf{A}} = < \mathbf{A}^{-1}\mathbf{V}_1, \mathbf{A}^{-1}\mathbf{V}_2 >_{\mathbf{I}} \tag{6.27}$$

The metric obtained in such a way is called left-invariant Riemannian metric. A left-invariant metric is independent of the choice of the inertial reference frame.

A left-invariant Riemannian metric was proposed by Park et al. [68], where the matrix $\mathbf{W}$ is given by

$$\mathbf{W} = \begin{bmatrix} \alpha\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \beta\mathbf{I} \end{bmatrix} \tag{6.28}$$

and α and β are positive scalars, which act like scaling factors for angular and linear velocities. Furthermore, additional choices of Riemannian metrics for rigid body kinematics have been proposed by Zefran et al. [69], which can lead to different results. The application of a Reimannian metric to a manifold leads to the following definition.

Definition 6.3 (Riemannian Manifold)**.** *A manifold equipped with a Riemannian metric is called a Riemannian manifold.*

One can easily imagine that the motion of a rigid body can be represented by a curve, $\mathbf{A}(t)$ on $SE(3)$. The velocity at an arbitrary point is the tangent vector to the curve at that point. In order to obtain other kinematic quantities, such as acceleration and jerk, and engage a dynamic analysis, it is necessary to differentiate the velocity vector field along the curve. If the manifold $SE(3)$ is embedded in the space of all 4×4 matrices, differentiation can be carried out in the 4×4 Euclidean space. If not, we have to define the covariant derivative of a vector field, which is a derivative that is intrinsic to $SE(3)$, and results do not depend on the ambient space. To define a covariant derivative, the notion of affine connection or the Levi-Civita connection (if we are working on a Riemannian manifold) needs to be introduced [70]. An affine connection in the abstract notion provides a way to compare tangent vectors that lie in different tangent spaces.

6.3.2 Affine Connection and Covariant Derivative

The motion of a rigid body, as mentioned, is represented by a curve, $\mathbf{A}(t)$, on $SE(3)$. The velocity at an arbitrary point is tangent to the curve at that point. In order to obtain the acceleration and other dynamic quantities, we need to be able to differentiate a vector field along the curve. At each point $\mathbf{A} \in SE(3)$, the value of a vector field belongs to the tangent space $T_{\mathbf{A}}SE(3)$ and, to differentiate a vector field along a curve, we must be able to compare vectors from the tangent spaces at different points on the curve. The issue in $SE(3)$ as opposed to $\mathbb{R}^n$, is that vectors cannot be translated to a common origin, because on curved spaces no canonical way of translating vectors exists and must be explicitly defined. To differentiate vector fields along a curve, it suffices

to define how to transport a vector along the curve from one tangent space to another. This process of translation of vectors along a curve is called *parallel transport* and was introduced by the Italian mathematician Levi-Civita.

Parallel transport is not unique. However, given a parallel transport, we can define the derivative of a vector field along a curve. Let $\mathbf{X}$ be a vector field along $\mathbf{A}(t)$ and let $\mathbf{X}(t)$ stand for $\mathbf{X}(\mathbf{A}(t))$. Denote by $\mathbf{X}^{t_0}(t)$ the parallel transport of vector $\mathbf{X}(t)$ to point $\mathbf{A}(t_0)$.

Definition 6.4 (Covariant Derivative)**.** *The covariant derivative of the vector field* $\mathbf{X}$ *along the curve* $\mathbf{A}(t)$ *is given by*

$$\frac{D}{dt}\mathbf{X}|_{t_0} = \lim_{t \to t_0} \frac{\mathbf{X}^{t_0}(t) - \mathbf{X}(t_0)}{t - t_0} \tag{6.29}$$

By taking the covariant derivative along integral curves of a vector field $\mathbf{Y}$, we obtain a covariant derivative of the vector field $\mathbf{X}$ with respect to the vector field $\mathbf{Y}$. We can, therefore, express a relation between the vector fields $\mathbf{X}$ and $\mathbf{Y}$.

Definition 6.5 (Affine Connection)**.** *Let the operator* ∇ *be the affine connection relating two vector fields* $\mathbf{X}$*,* $\mathbf{Y}$*, defined as*

$$\nabla_{\mathbf{Y}}\mathbf{X}|_{\mathbf{A}_0} = \frac{D}{dt}\mathbf{X}|_{t_0} \tag{6.30}$$

where $\frac{D}{dt}\mathbf{X}$ *is taken along the integral curve of* $\mathbf{Y}$ *passing through* $\mathbf{A}_0 = \mathbf{A}(t_0)$*.*

In order to compute the covariant derivative $\nabla_{\mathbf{Y}}\mathbf{X}$, it suffices to know the covariant derivatives of the basis vector fields. Thus, we first express the vector fields as a combination of the basis, using the Einstein summation notation

$$\mathbf{X} = X^i \hat{\mathbf{L}}_i$$
$$\mathbf{Y} = Y^i \hat{\mathbf{L}}_i$$

where the basis $\hat{\mathbf{L}}_i$ is a covariant basis and the components X^i, Y^i form a contravariant tensor. Then, the expression for $\nabla_{\mathbf{Y}}\mathbf{X}$ is given by

$$\nabla_{\mathbf{Y}}\mathbf{X} = \frac{dX^i}{dt}\hat{\mathbf{L}}_i + X^i Y^j \nabla_{\hat{\mathbf{L}}_j}\hat{\mathbf{L}}_i \tag{6.31}$$

where $\frac{d}{dt}$ is the derivative along the integral curve of $\mathbf{Y}$.

Definition 6.6 (Levi-Civita Connection)**.** *Given a Riemannian manifold, there exists a unique connection, called the Levi-Civita connection, which is compatible with the metric and symmetric.*

Assume three vector fields $\mathbf{X}$, $\mathbf{Y}$,and $\mathbf{Z}$. If ∇ is the Levi-Civita connection for any of these vector fields, it can be shown that it satisfies the following two properties

$$\mathbf{X} < \mathbf{Y}, \mathbf{Z} >=< \nabla_{\mathbf{X}}\mathbf{Y}, \mathbf{Z} > + < \mathbf{Y}, \nabla_{\mathbf{X}}, \mathbf{Z} > \quad \text{compatibility with the metric} \tag{6.32}$$

$$\nabla_{\mathbf{X}}\mathbf{Y} - \nabla_{\mathbf{Y}}\mathbf{X} = [\mathbf{X}, \mathbf{Y}] \quad \text{symmetry} \tag{6.33}$$

It also satisfies the following identity

$$\begin{aligned} < \mathbf{Z}, \nabla_{\mathbf{X}}\mathbf{Y} >= \frac{1}{2}\{&\mathbf{Y} < \mathbf{X}, \mathbf{Z} > + \mathbf{X} < \mathbf{Z}, \mathbf{Y} > - \mathbf{Z} < \mathbf{X}, \mathbf{Y} > + \\ &< [\mathbf{Z}, \mathbf{Y}], \mathbf{X} > + < [\mathbf{Z}, \mathbf{X}], \mathbf{Y} > + < [\mathbf{X}, \mathbf{Y}], \mathbf{Y} >\} \end{aligned} \tag{6.34}$$

The velocity, of the rigid body moving along the curve $\mathbf{A}(t)$ is given by the tangent vector field $\mathbf{V}(t) = \frac{d\mathbf{A}(t)}{dt}$. Given a connection, we can now define the acceleration and higher derivatives of the velocity. The acceleration is the covariant derivative of the velocity along the curve

$$\frac{D}{dt}\mathbf{V} = \nabla_{\frac{d\mathbf{A}}{dt}}\mathbf{V} = \nabla_{\mathbf{V}}\mathbf{V} \tag{6.35}$$

In the same way, one can derive an expression for the jerk of the curve. This is given by $\nabla_{\mathbf{V}}\nabla_{\mathbf{V}}\mathbf{V}$. At this point, we have to note that the velocity does not depend on the choice of the connection, but unfortunately we cannot claim the same proposition for the acceleration and the jerk. Once a left-invariant metric is introduced on $SE(3)$, the corresponding Riemannian connection can be directly derived from Eq.(6.34).

Assume that we are using the left-invariant metric in Eq.(6.28) and let $\mathbf{X} = X^i\hat{\mathbf{L}}_i$ and $\mathbf{Y} = Y^i\hat{\mathbf{L}}_i$ be two arbitrary vector fields with the corresponding vector pairs $(\boldsymbol{\omega}_x, \boldsymbol{v}_x)$ and $(\boldsymbol{\omega}_y, \boldsymbol{v}_y)$. The Riemannian connection corresponding to the metric in Eq.(6.28) is given by

$$\nabla_{\mathbf{X}}\mathbf{Y} = \left(\frac{d\boldsymbol{\omega}_y}{dt} + \frac{1}{2}\boldsymbol{\omega}_x \times \boldsymbol{\omega}_y, \frac{d\boldsymbol{v}_y}{dt} + \boldsymbol{\omega}_x \times \boldsymbol{v}_y\right) \tag{6.36}$$

At this point we have to note that Eq.(6.36) is independent of the choice of the scalars α and β.

Moving on, to obtain smooth curves in $SE(3)$, we need to define another tool, namely the Riemannian curvature.

Definition 6.7 (The Riemann-Cristoffel Tensor)**.** *The curvature tensor of a Riemannian manifold is a correspondence* $\boldsymbol{R}$ *that associates to a pair of vector fields* $\mathbf{X}$, $\mathbf{Y}$ *a mapping*

$$\boldsymbol{R}(\mathbf{X},\mathbf{Y})\mathbf{Z} = \nabla_{\mathbf{Y}}\nabla_{\mathbf{X}}\mathbf{Z} - \nabla_{\mathbf{X}}\nabla_{\mathbf{Y}}\mathbf{Z} + \nabla_{[\mathbf{X},\mathbf{Y}]}\mathbf{Z} \tag{6.37}$$

Assume again the three vector fields $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ on $SE(3)$ associated with the vector pairs $(\boldsymbol{\omega}_x, \boldsymbol{v}_x)$, $(\boldsymbol{\omega}_y, \boldsymbol{v}_y)$ and $(\boldsymbol{\omega}_z, \boldsymbol{v}_z)$, respectively. The Riemannian curvature corresponding to the Riemannian connection, given by Eq.(6.36), is

$$\boldsymbol{R}(\mathbf{X},\mathbf{Y})\mathbf{Z} = \left(\frac{1}{4}(\boldsymbol{\omega}_x \times \boldsymbol{\omega}_y) \times \boldsymbol{\omega}_z, 0\right) \tag{6.38}$$

If $\mathbf{V}$ is the velocity associated with the motion $\mathbf{A}(t)$ of a rigid body and $(\boldsymbol{\omega}, \boldsymbol{v})$ is the corresponding vector pair, adopting the metric in Eq.(6.28) and using Eq.(6.36), the acceleration will be given by

$$\nabla_{\mathbf{V}}\mathbf{V} = (\dot{\boldsymbol{\omega}}, \dot{\boldsymbol{v}} + \boldsymbol{\omega} \times \boldsymbol{v}) \tag{6.39}$$

which is the usual expression for the acceleration of a rigid body in $\mathbb{R}^3$, as shown in Murray et al. [71]. The expression for the third derivative of motion, the jerk, is given by

$$\nabla_{\mathbf{V}}\nabla_{\mathbf{V}}\mathbf{V} = \left(\frac{d\boldsymbol{\omega}}{dt} + \frac{1}{2}\boldsymbol{\omega} \times \dot{\boldsymbol{\omega}}, \frac{d(\dot{\boldsymbol{v}} + \boldsymbol{\omega} \times \boldsymbol{v})}{dt} + \boldsymbol{\omega} \times (\dot{\boldsymbol{v}} + \boldsymbol{\omega} \times \boldsymbol{v})\right) \tag{6.40}$$

Once again, this is the same expression as the usual jerk expression [71].

6.3.3 Geodesic Curves

Next, we will consider trajectories between a starting and a final position and orientation, which minimize an integral cost, while satisfying additional boundary conditions. First, we will focus on the geodesic curves or the shortest-path curves. To generate necessary conditions for the optimal trajectories satisfying such costs, one needs to be familiar with the calculus of variations on manifolds [67, 72]. We will skip the derivation to present only the necessary conditions on how to obtain a geodesic curve. The minimal geodesic curve on $SE(3)$ is the solution that minimizes the energy functional given by

$$J = L_E = \int_a^b < \mathbf{V}, \mathbf{V} > dt \tag{6.41}$$

Using properties of the Riemannian connection and the variation of L_E, we can obtain the geodesic equation.

Theorem 6.1 (Geodesic curve)**.** *Let* $\mathbf{A}(t)$ *be a curve on a Riemannian manifold with the metric defined by Eq.(6.28). If* $\mathbf{A}(t)$ *minimizes the functional* L_E*, given by Eq.(6.41), it satisfies the following differential equation*

$$\nabla_{\mathbf{V}}\mathbf{V} = 0 \tag{6.42}$$

where $\mathbf{V} = \frac{d\mathbf{A}(t)}{dt}$.

To solve Eq.(6.42) and find the geodesics on $SE(3)$, we express $\mathbf{V}$ as a linear combination of the left invariant vectors fields, $\hat{\mathbf{L}}_1, \ldots, \hat{\mathbf{L}}_6$, according to Eq.(6.22). This leads to the following proposition.

Proposition 6.1. *If* $\mathbf{A}(t)$ *is the geodesic for the left-invariant metric in Eq.(6.28), the vector pair* $(\boldsymbol{\omega}, \boldsymbol{v})$*, corresponding to the velocity of the vector field* $\mathbf{V} = \frac{d\mathbf{A}(t)}{dt}$*, must satisfy the equations*

$$\frac{d\boldsymbol{\omega}}{dt} = 0 \tag{6.43}$$

$$\frac{d\boldsymbol{v}}{dt} = -\boldsymbol{\omega} \times \boldsymbol{v} \tag{6.44}$$

Note again that this result is independent of the scalars α, β. Solving Eq.(6.42) for an initial and a final configuration $\mathbf{T}_\mathrm{i}, \mathbf{T}_\mathrm{f} \in SE(3)$ respectively leads to the following proposition.

Proposition 6.2. *Given the initial and final foot rotation matrices,* $\mathbf{R}_\mathrm{i}, \mathbf{R}_\mathrm{f} \in SO(3)$*, and the initial and final foot positions in the three-dimensional space,* $\mathbf{d}_\mathrm{i}, \mathbf{d}_\mathrm{f} \in \mathbb{R}^3$*, the geodesic reference trajectory at time* t *is given by*

$$\mathbf{R}^{\mathrm{ref}}(t) = \mathbf{R}_\mathrm{i} e^{t\hat{\boldsymbol{\omega}}_0} \tag{6.45}$$

$$\mathbf{d}^{\mathrm{ref}}(t) = t(\mathbf{d}_\mathrm{f} - \mathbf{d}_\mathrm{i}) + \mathbf{d}_\mathrm{i} \tag{6.46}$$

where $\hat{\boldsymbol{\omega}}_0 \in so(3)$ *can be determined from the initial and final rotation and position as follows*

$$\hat{\boldsymbol{\omega}}_0 = \log(\mathbf{R}_\mathrm{i}^\top \mathbf{R}_\mathrm{f}) \tag{6.47}$$

Operators log and e stand for the logarithmic and exponential map respectively [71]. The outcome is an affine transformation $\mathbf{T}^{\mathrm{ref}}(t) \in SE(3)$

$$\mathbf{T}^{\mathrm{ref}}(t) = \begin{bmatrix} \mathbf{R}^{\mathrm{ref}}(t) & \mathbf{d}^{\mathrm{ref}}(t) \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{6.48}$$

which is subsequently expressed into the robot local coordinate frame and, after taking into account the target CoM position, is fed into inverse kinematics for computing joint angles. An example of omnidirectional walk, using the geodesic curves, is shown in Figure 6.4.

6.3.4 Minimum-Jerk Curves with Homogeneous Boundary Conditions

We will follow a similar procedure to obtain the minimum-jerk curve between two configurations in $SE(3)$ [66]. The minimum-jerk curve is obtained by minimizing the integral of the norm of the Cartesian jerk, provided that the appropriate boundary conditions are given. In particular, the minimum-jerk curve is well-defined, when the initial and the final velocities and accelerations are specified. It is obvious that such trajectories are particularly useful in robotics, where one is generally able to control the acceleration of the end effector of a robot, and therefore the position and the velocity, since the electromechanical actuators cannot produce sudden changes in the acceleration. In addition, it is important to note that, according to Flash et al. [73], humans plan trajectories that minimize such an integral measure of the jerk to achieve a smooth motion between points in space.

The minimum-jerk cost functional is given by

$$J = L_J = \int_a^b < \nabla_{\mathbf{V}} \nabla_{\mathbf{V}} \mathbf{V}, \nabla_{\mathbf{V}} \nabla_{\mathbf{V}} \mathbf{V} > dt \tag{6.49}$$

where, as mentioned before, $\mathbf{V} = \frac{d\mathbf{A}(t)}{dt}$. The necessary conditions for the solution of Eq.(6.49) are stated in the following theorem

Theorem 6.2 (Minimal Jerk Curves)**.** *Let $\mathbf{A}(t)$ be a curve on a Riemannian manifold with the metric defined by Eq.(6.28) that satisfies the boundary conditions and let $\mathbf{V} = \frac{d\mathbf{A}(t)}{dt}$. If $\mathbf{A}(t)$ minimizes the functional L_J given by Eq.(6.49), then it satisfies the following equation*

$$\nabla^5_{\mathbf{V}} \mathbf{V} + \mathbf{R}(\mathbf{V}, \nabla^3_{\mathbf{V}} \mathbf{V})\mathbf{V} - \mathbf{R}(\nabla_{\mathbf{V}} \mathbf{V}, \nabla^2_{\mathbf{V}} \mathbf{V})\mathbf{V} = 0 \tag{6.50}$$

In general, the rotational components of the necessary condition of Eq.(6.50) cannot be solved analytically. However, when the initial and the final velocities and accelerations are prescribed to be zero, an analytical solution can be obtained.

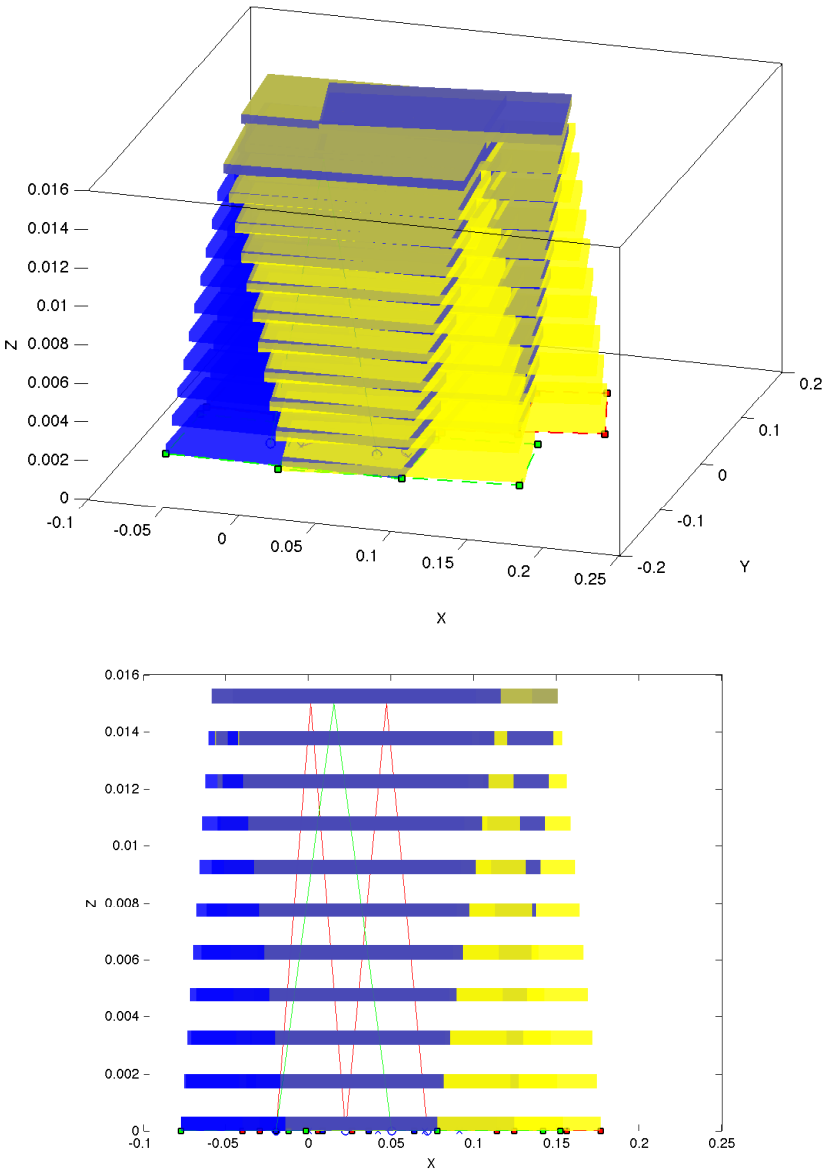

Figure 6.4: Feet trajectory with geodesic curves (top: 3D view, bottom: lateral view).

Proposition 6.3. *Given the initial and final foot rotation matrices,* $\mathbf{R}_\mathrm{i}, \mathbf{R}_\mathrm{f} \in SO(3)$*, and the initial and final foot positions in the three-dimensional space,* $\mathbf{d}_\mathrm{i}, \mathbf{d}_\mathrm{f} \in \mathbb{R}^3$*, provided that the initial and final velocities and accelerations are zero, the minimum-jerk reference trajectory at time* t *can be computed as*

$$\mathbf{R}^{\mathrm{ref}}(t) = \mathbf{R}_\mathrm{i} e^{p(t)\hat{\omega}_0} \tag{6.51}$$

$$\mathbf{d}^{\mathrm{ref}}(t) = p(t)(\mathbf{d}_\mathrm{f} - \mathbf{d}_\mathrm{i}) + \mathbf{d}_\mathrm{i} \tag{6.52}$$

where $p(t) = 6t^5 - 15t^4 + 10t^3$ *and* $\hat{\omega}_0 \in so(3)$ *can be determined from the initial and final rotation and position as follows*

$$\hat{\boldsymbol{\omega}}_0 = \log(\mathbf{R}_\mathrm{i}^\top \mathbf{R}_\mathrm{f}) \tag{6.53}$$

The outcome is an affine transformation $\mathbf{T}^{\mathrm{ref}}(t) \in SE(3)$

$$\mathbf{T}^{\mathrm{ref}}(t) = \begin{bmatrix} \mathbf{R}^{\mathrm{ref}}(t) & \mathbf{d}^{\mathrm{ref}}(t) \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{6.54}$$

The requirement of zero velocity and acceleration at the boundaries is easily satisfied and is also desirable in humanoid walk, since at the boundaries the foot either rests on the floor (zero velocity and acceleration) or it has reached the maximum height during the swing and reverses directions along the z-axis (again, zero velocity and acceleration). An example of omnidirectional walk, using the minimum-jerk curves, is shown in Figure 6.5. Comparing this trajectory to the geodesic trajectory in Figure 6.4, it is evident that the path followed by the rigid body is the same. However, the minimum-jerk trajectory must start and end with zero velocity and zero acceleration. Indeed, the velocity starts from zero with zero acceleration, rises to a peak, and then decreases to zero with zero acceleration.

6.4 Adaptive Stepping

If a disturbance is sensed during a SS phase, the placement of the swinging foot can be adaptively corrected, so as to increase the area of the support polygon in the upcoming DS phase and, therefore, improve stability of the gait. At sampling instant k the future ZMP error in the x-axis (similarly in the y-axis) is estimated from the observer as

$$z_x^e(k+1) = \hat{z}_x(k+1) - z_x^{\mathrm{ref}}(k+1) \tag{6.55}$$

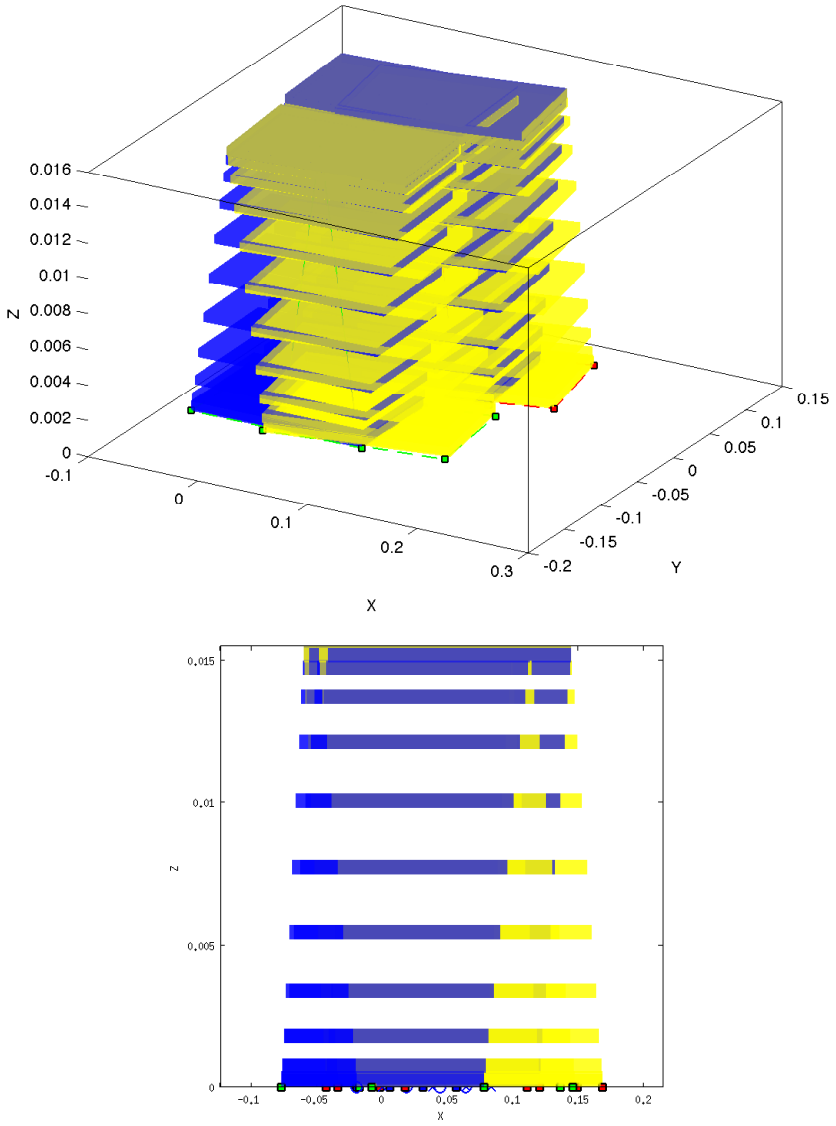

Figure 6.5: Feet trajectory with minimum-jerk curves (top: 3D view, bottom: lateral view).

Then, an appropriate, online modification to the foot trajectory of the swinging leg, $d_x(k+1)$, can be achieved by

$$d_x(k+1) = d_x^{\text{ref}}(k+1) + K_p z_x^e(k+1) \tag{6.56}$$

where K_p is a scalar gain determined by experimentation.

6.5 Conclusion

In this chapter, we focused on generating a smooth feet trajectory, since it is vital for achieving smooth human-like walking in practice. First, the cubic Bezier interpolation method was presented and discussed. This method generated smooth trajectories, however, the take-off and the landing of the feet were not smooth, which in practice resulted in situations where the humanoid robot "hit" the floor with force. To compensate for this phenomenon, the cubic spline was presented and the cubic B-spline interpolation method was adopted. This method yielded smooth feet trajectories, with smooth take-off and landing phases in practice. Furthermore, the concept of treating the foot as a rigid object was proposed, interpolating in the SE(3) group. After a short trip in the world of differential geometry and Riemannian manifolds, the geodesic curves were derived corresponding to the shortest-path trajectories. These trajectories are not suitable for smooth walking, thus the minimum-jerk curves with homogeneous boundary conditions were further derived, yielding smooth human-like walking in practice. The advantage of this method is that, by interpolating in the SE(3) group, we generate affine transformation matrices for the feet, which can be used by the inverse kinematic procedure in a straightforward way. Finally, a simple method for adaptive online stepping was presented, allowing online modification to the reference feet trajectory generated, when disturbances or uneven terrain is sensed.

Chapter 7

Implementation

I do not fear computers. I fear the lack of them.

Isaac Asimov (1920–1992)

This chapter addresses important issues present in practice. The control and estimation schemes presented in Chapter 5 along with the interpolation methods in Chapter 6 have been fully implemented on the Aldebaran Nao humanoid robot (Section 2.3.8) for the needs of our RoboCup team *Kouretes*. We used Nao version 3.3, which is a 58cm, 5kg humanoid robot, equipped with an x86 AMD Geode processor at 500 MHz and 256 MB SDRAM. Furthermore, it has 21 degrees of freedom; 2 in the head, 4 in each arm, 5 in each leg, and 1 in the pelvis. All joints are position-controlled, using closed-loop PID controllers and encoders. It also features an Inertial Measurement Unit (IMU) in the torso and 4 FSR at each foot. In Section 7.1, we utilize a discrete-time Kalman filter for dissipating the noise the FSR readings impose in the ZMP estimation. Furthermore, the significant delay appearing in the measurements is addressed. In Section 7.2, another important issue is anticipated, namely the non-systematic bias the FSR impose in the ZMP estimation. Approaches, where the Kalman filter is used for its estimation, had no practical impact, due to the loss of observability. Therefore, the Luenberger observer is used, effectively estimating the ZMP bias. Finally, in Section 7.3, the actuation error issue is addressed. Such an error, if not corrected, can have negative effects on the walking gait, as time progresses.

7.1 ZMP Estimation using FSR

The FSR measurements come with significant delay, practically measured up to $50ms$, and are corrupted by noise. These issues compromise the stability of the proposed control schemes. To compensate for the delay and noise and acquire a noise-free reading for the ZMP in each axis, when at least one foot is in contact with the ground, a discrete-time Kalman filter with feedback from the Cart and Table model in the corresponding axis is employed. Assuming a delay of d sampling instants, the filter dynamics in the x-axis are:

$$\hat{z}_x^{\mathrm{FSR}}\big(k-d\big|k-d-1\big) = \hat{z}_x^{\mathrm{FSR}}\big(k-d-1\big|k-d-1\big) + \Delta\hat{z}_x(k-d) \tag{7.1}$$

$$\tilde{y}(k-d) = z_x^{\mathrm{FSR}}(k-d) - \hat{z}_x^{\mathrm{FSR}}\big(k-d\big|k-d-1\big) \tag{7.2}$$

$$\hat{z}_x^{\mathrm{FSR}}\big(k-d\big|k-d\big) = \hat{z}_x^{\mathrm{FSR}}\big(k-d\big|k-d-1\big) + K(k-d)\tilde{y}(k-d) \tag{7.3}$$

where $\hat{z}_x^{\mathrm{FSR}}$ is the filter's estimate of the ZMP, z_x^{FSR} is the ZMP reading provided by the FSR, $\tilde{y}$ is the innovation and K is the optimal Kalman gain, which takes into account the process and measurement noise. The input of the filter is the incremental ZMP estimate $\Delta\hat{z}_x(k) = \hat{z}_x(k) - \hat{z}_x(k-1)$ provided by the observer. This way, observations are integrated immediately into the filter's estimate at the correct sampling instant. As a result, the filter's estimate is also delayed by d sampling instants. Given the fact that the Kalman filter is an optimal predictor, we use it to obtain the estimate d sampling instants in the future. In such a way the estimate $\hat{z}_x^{\mathrm{FSR}}(k|k-d)$ becomes synchronized and can be properly used for feedback.

7.2 Elimination of ZMP Bias

Another important issue encountered is a systematic bias of the ZMP readings, due to sensor tolerances. This bias varies over time as each particular sensor loses and gains contact periodically. Thus, Nao tends to hobble in the direction of the bias and eventually tips over. Our initial attempt to estimate the bias with the Kalman filter presented above had no practical impact, due to the lack of observability. To overcome this issue, the state space of the Cart and Table model was augmented with a ZMP bias state, so that the system observer can estimate it since observability is maintained. The augmented state vector is given by:

$$\widetilde{\mathbf{x}}(k) = \begin{bmatrix} c_x(kT_s) & \dot{c}_x(kT_s) & \ddot{c}_x(kT_s) & z_x^{\mathrm{bias}}(kT_s) \end{bmatrix}^\top \tag{7.4}$$

and the augmented Cart and Table model takes the form:

$$\widetilde{\mathbf{x}}(k+1) = \overbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}}^{\widetilde{\mathbf{A}}} \widetilde{\mathbf{x}}(k) + \overbrace{\begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}}^{\widetilde{\mathbf{b}}} u_x(k) \tag{7.5}$$

$$z_x(k) = \overbrace{\begin{bmatrix} \mathbf{c}^\top & 1 \end{bmatrix}}^{\widetilde{\mathbf{c}}^\top} \widetilde{\mathbf{x}}(k) \tag{7.6}$$

Therefore, the augmented observer dynamics are given by:

$$\widetilde{\mathbf{y}}_{\mathrm{m}}(k) = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -h_c/g & 1 \end{bmatrix}}^{\widetilde{\mathbf{C}}_{\mathrm{m}}} \widetilde{\mathbf{x}}(k) = \begin{bmatrix} c_x^{\mathrm{ENC}}(k) \\ \hat{z}_x^{\mathrm{FSR}}(k|k-d) \end{bmatrix}$$

$$\widehat{\widetilde{\mathbf{x}}}(k+1) = \widetilde{\mathbf{A}}\widehat{\widetilde{\mathbf{x}}}(k) + \widetilde{\mathbf{b}}u_x(k) + \widetilde{\mathbf{L}}_{\mathrm{obs}} \left(\widetilde{\mathbf{y}}_{\mathrm{m}}(k) - \widetilde{\mathbf{C}}_{\mathrm{m}}\widehat{\widetilde{\mathbf{x}}}(k) \right)$$

where $\hat{z}_x^{\mathrm{FSR}}(k|k-d)$ is the filter's estimate, which serves as a measurement of $\hat{z}_x(k|k-d) + z_x^{\mathrm{bias}}(k|k-d)$. This approach proved to be very effective and converged to the bias values in both axes. The final implemented control and estimation scheme is shown in Fig. 7.1.

7.3 Actuation Error Correction

Despite the use of the analytical inverse kinematics for the Nao humanoid robot, proposed by N.Kofinas et. al [74], a given target $\mathbf{T}^{\mathrm{ref}}(k)$ for a foot is never reached without error on a real robot. Let $\mathbf{T}^{\mathrm{m}}(k)$ be the actual position reached as measured by the joint encoders with respect to an inertial frame. The actuation error can be computed as

$$\mathbf{E}(k) = \left(\mathbf{T}^{\mathrm{m}}(k)\right)^{-1} \mathbf{T}^{\mathrm{ref}}(k-1) \tag{7.7}$$

Assuming this error is roughly constant for consecutive sampling instances, we can adjust the next target to account for it using a damping method. In particular, the Euler angles and the displacement are extracted from $\mathbf{E}(k)$, they are scaled by a damping factor $0 \leq \gamma \ll 1$, a new transformation matrix $\mathbf{E}_\gamma(k)$ is reconstructed using the damped values, and the corrected next target is expressed

$$\mathbf{T}(k+1) = \mathbf{T}^{\mathrm{ref}}(k+1)\mathbf{E}_\gamma(k) \tag{7.8}$$

In such a way a more accurate tracking of the preplanned feet trajectory can be achieved, which can be further used for odometry purposes.

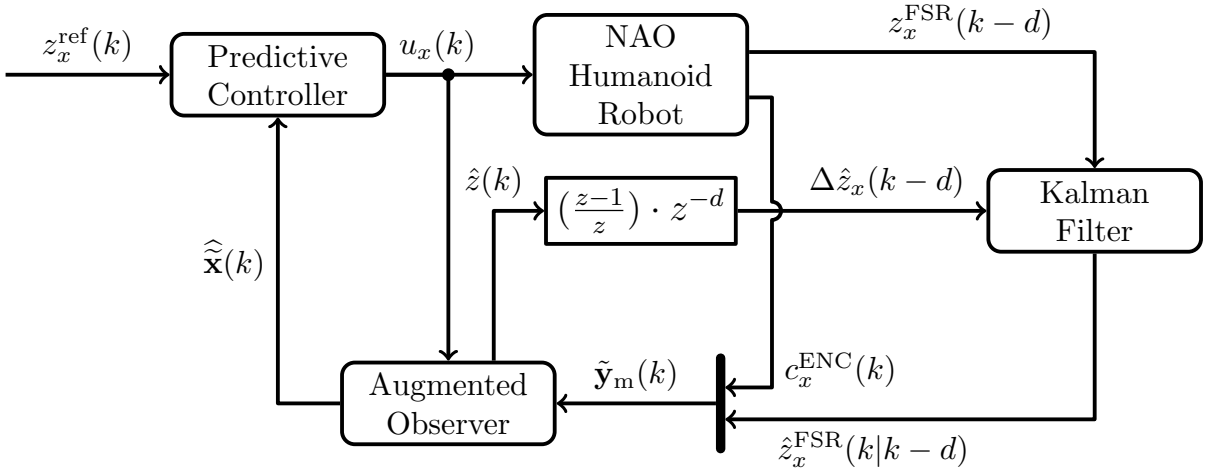

Figure 7.1: Implemented control and estimation scheme on the Nao robot

7.4 Conclusion

This chapter was dedicated to issues arising, when crossing the gap from theory to practice. The most important issues were successfully addressed, leading to online control of the gait. First, the noise the FSR imposed along with the delay appearing in the readings were addressed by a discrete-time linear Kalman filter. In such a way, a noise-free measurement of the ZMP, synchronized with the on board execution time, can be obtained. However, as observed in practice, this was not enough. The ZMP measurement comes with a significant bias value. Using the Kalman filter for estimating the bias value yielded no results, since the observability is lost. This bias is not systematic, since it depends on the number of FSR that are in contact with the ground at the time the measurement is taken. This effect did compromise the stability of the overall control scheme and often led to a fall. To compensate for this phenomenon, we augmented the Luenberger observer with the bias state and we let the observer estimate it online. This approach proved very effective in practice. Finally, the actuation errors induced by the robot's actuators and/or the inverse kinematics were anticipated, thus a more accurate gait was obtained.

Chapter 8

Experimental Results

I have had my results for a long time: but I do not yet know how I am to arrive at them.

Carl-Friedrich Gauss (1777–1855)

In this chapter, experimental results are presented for the proposed control and estimation schemes. First, a straight gait is demonstrated by both control approaches on the Nao robot, with the constrained LMPC yielding smoother and faster gaits, while keeping the ZMP tracking error smaller compared to the response generated by the preview controller. In addition, in both approaches the elimination of noise, delay, and bias is obvious. Furthermore, the bias estimate by the observer is illustrated and briefly discussed. Next, the Nao robot is heavily disturbed, while the constrained LMPC is used for countering the disturbances, and the effectiveness of the constraints on the ZMP is visualized and discussed. Then, walking to the side (side-steps), as well as walking and stopping, are considered. Finally, the Nao robot is forced to walk over uneven terrain. The gait demonstrated is omnidirectional and dynamically stable, validating our claims.

8.1 Straight Gait

The walk approach is demonstrated for a simple 30-step straight gait with an average speed of $14cm/sec$ using both preview control (Fig. 8.2) and constrained LMPC (Fig. 8.4)

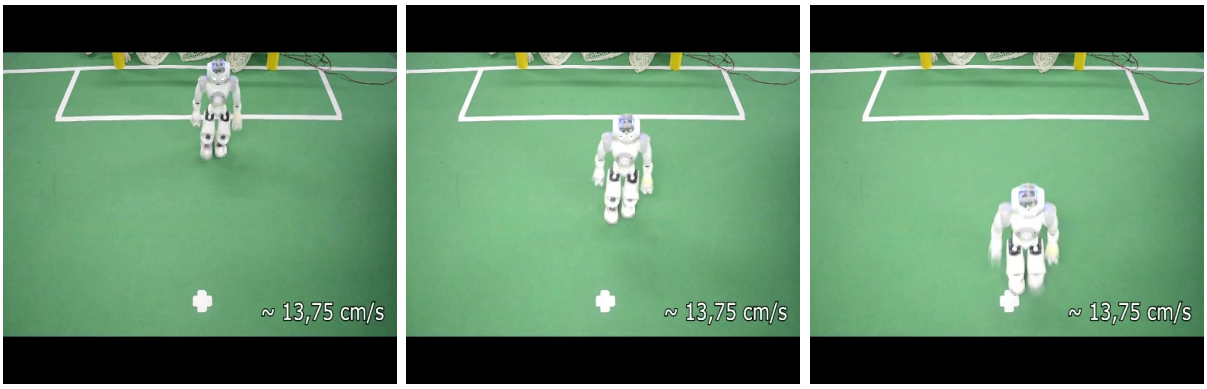

Figure 8.1: Straight gait using preview control on Nao

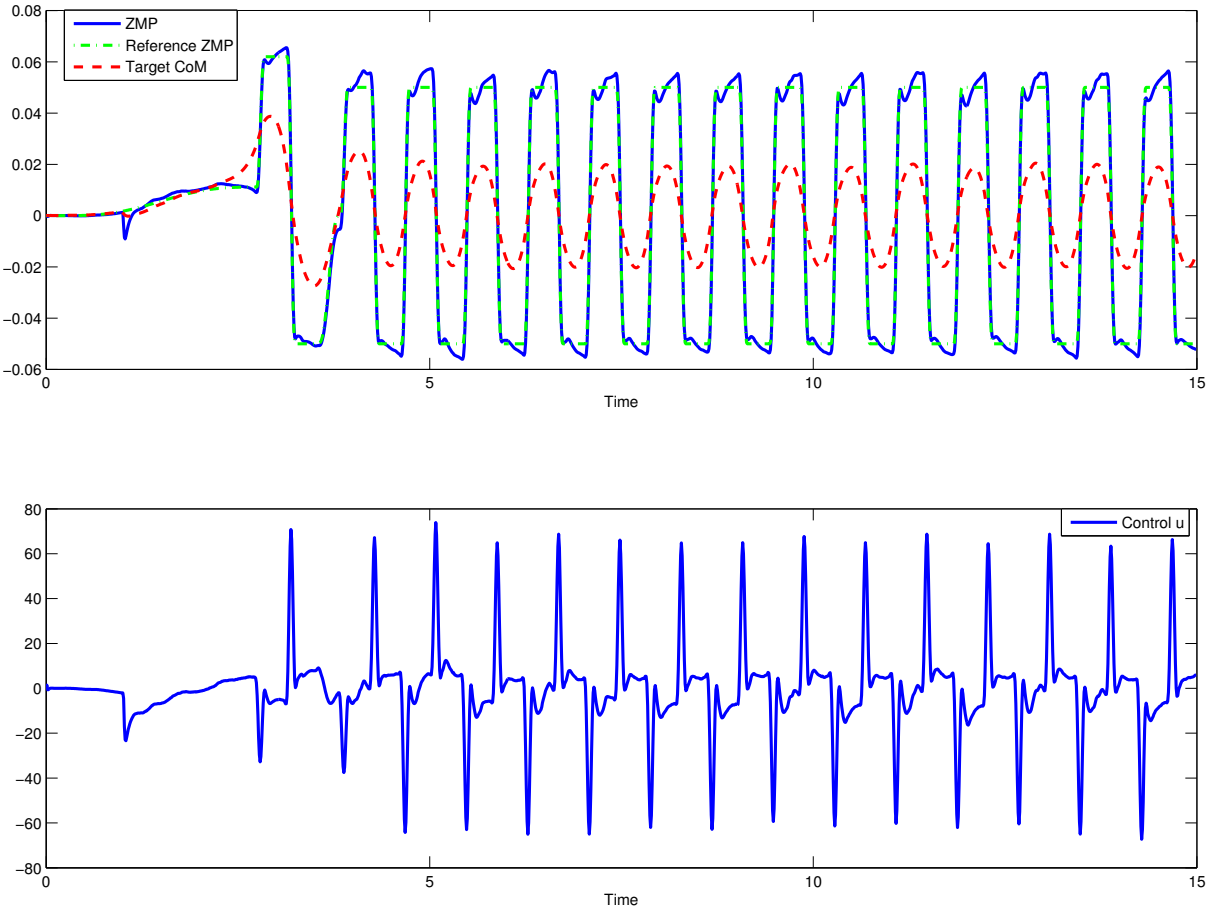

Figure 8.2: y-axis response for straight gait using preview control on Nao

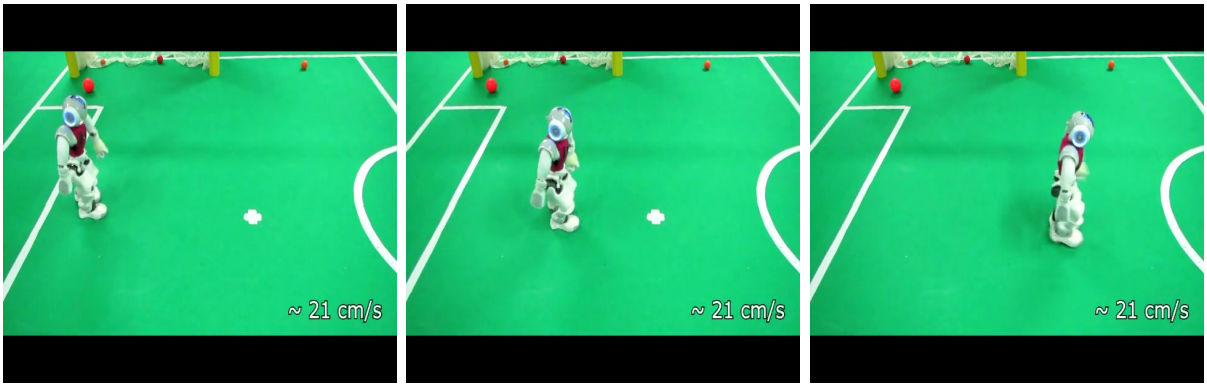

Figure 8.3: Straight gait using constrained LMPC on Nao

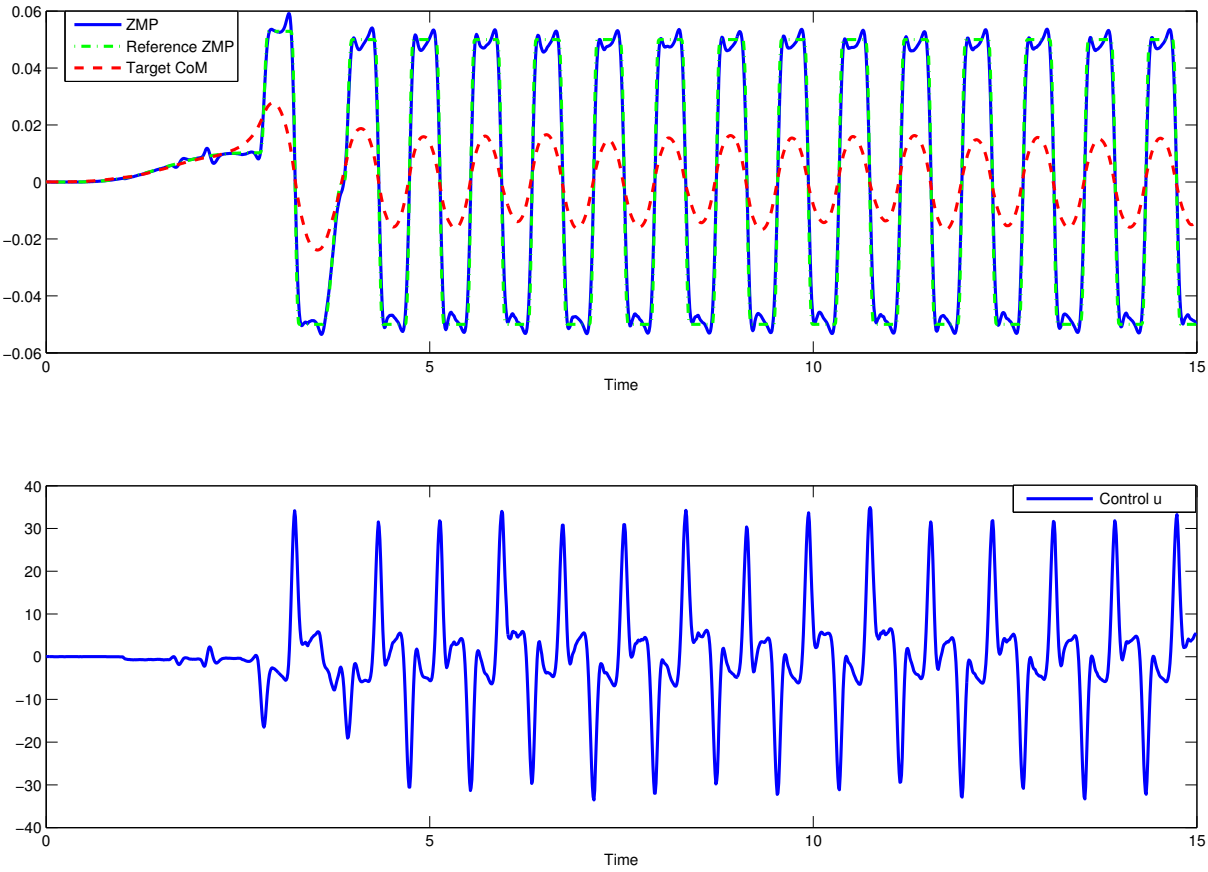

Figure 8.4: y-axis response for straight gait using constrained LMPC on Nao

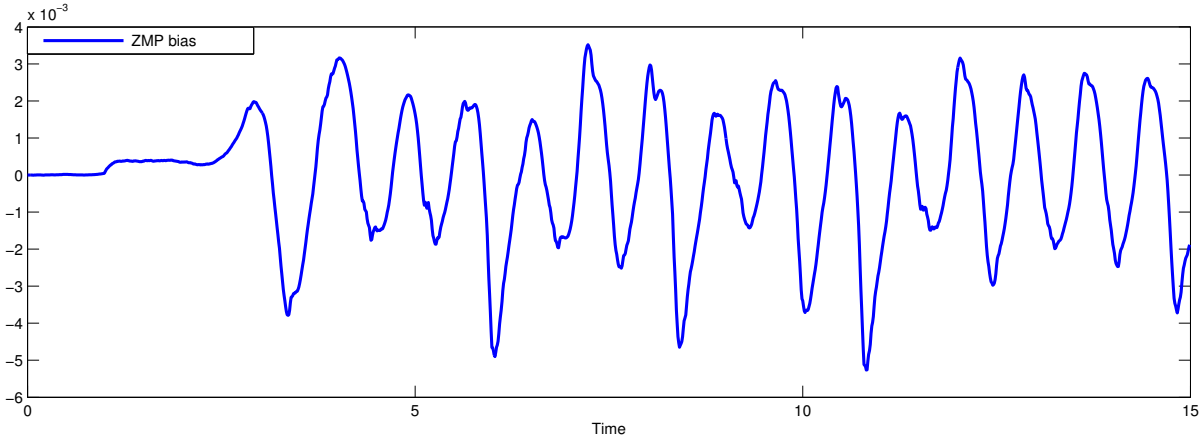

Figure 8.5: Estimated ZMP bias in the y-axis for straight gait

computed on board on a real Nao robot. In the experimentation, we used a sampling period of $T_s = 10ms$, a delay of $d = 5$, a prediction horizon of $N_p = 100$ for both predictive controllers, a set of $N = 15$ Laguerre functions, and $\alpha = 0.8$. In both cases, the reference ZMP is closely tracked by the estimated ZMP with the constrained LMPC yielding more accurate tracking, while keeping the magnitude of the jerk smaller. Both controllers operate at the sampling period T_s. The constrained LMPC is solved sufficiently fast for the $10ms$ period, requiring approximately $5 - 6ms$ of CPU time. Additionally, there are no signs of ZMP delays, noise, and bias, evidenced in similar approaches [53] and also in our initial experiments, where these issues were not explicitly addressed. Finally, the most significant difference between the two approaches can be observed in the magnitude of the control signal; the constrained LMPC results in a smoother CoM trajectory and, thus, smoother and much faster walk (measured up to $42cm/sec$ on Nao).

The ZMP bias estimated by the augmented observer over time is shown in Fig. 8.5. The bias changes smoothly during the cycle (..., left SS, DS, right SS, DS, ...) of support phases. Note that for this particular robot there is no symmetry between left and right leg ZMP biases.

8.2 Disturbance Rejection

In our next experiment, the disturbance rejection property of the constrained LMPC is illustrated. A standing Nao robot is "pushed" for approximately $1.5s$ starting at time $2s$.

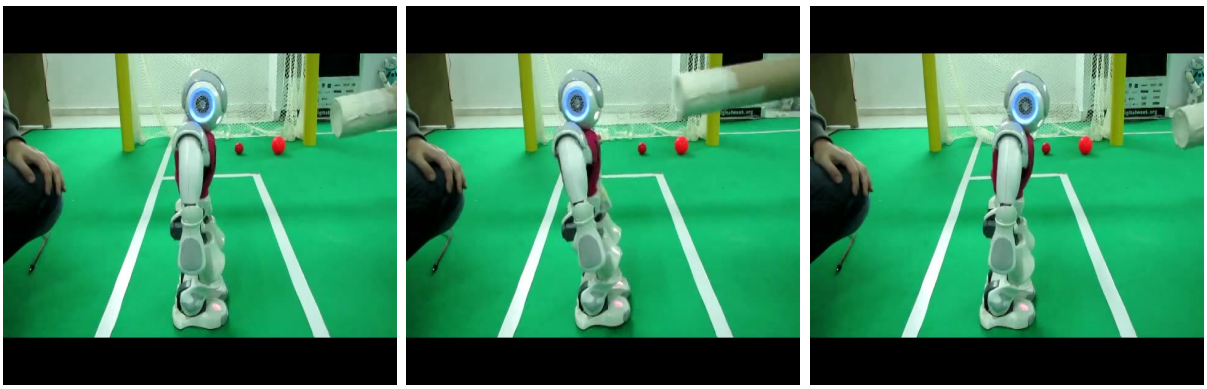

Figure 8.6: Disturbance rejection using constrained LMPC on Nao

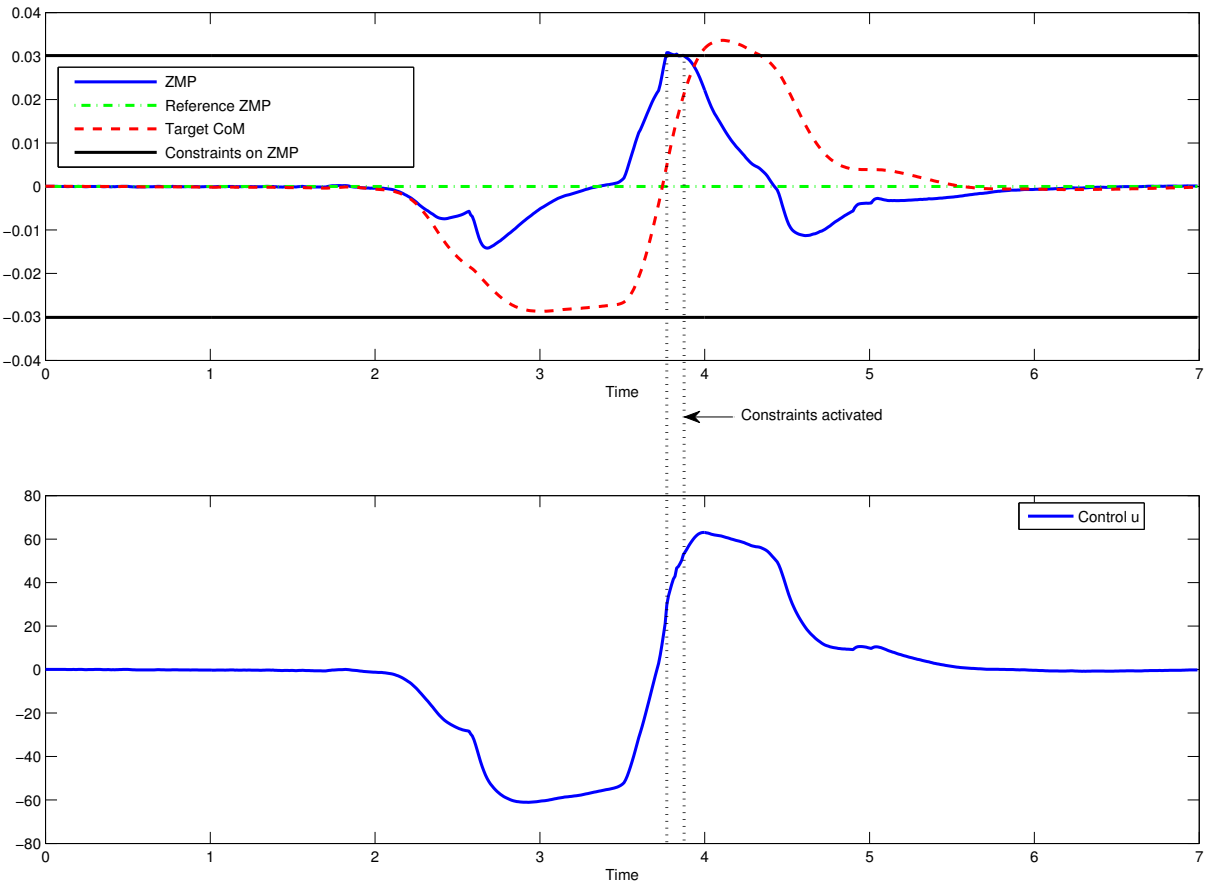

Figure 8.7: Constrained LMPC x-axis response when disturbing a standing Nao

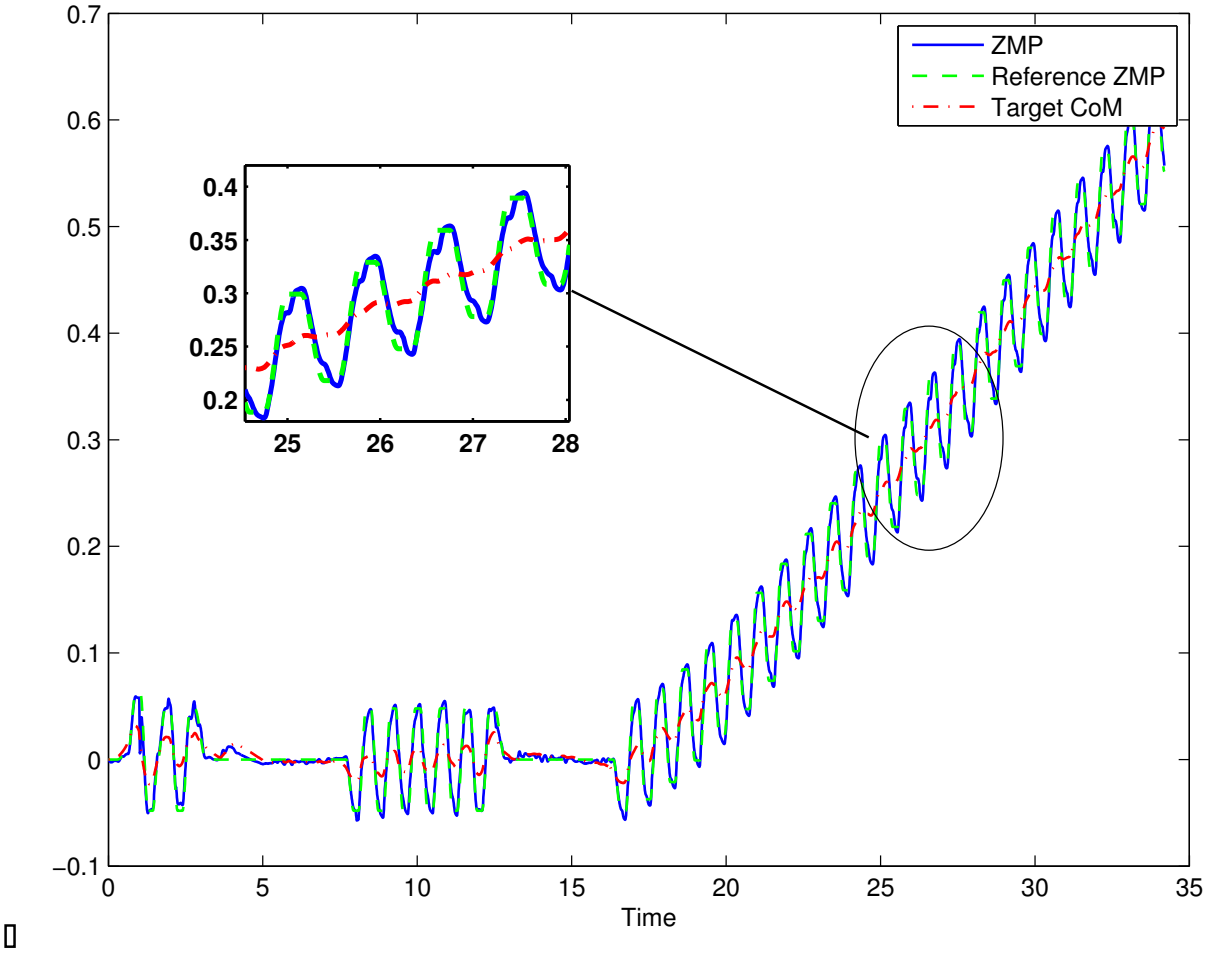

Figure 8.8: y-axis response for a side-stepping gait using constrained LMPC on Nao

The attempt to counter the disturbance makes the estimated by the observer ZMP "hit" the constraints, nevertheless without violating them; soon thereafter, both the estimated ZMP and the CoM are driven to equilibrium. The x-axis response of this experiment is shown in Fig. 8.7.

8.3 Side Steps

Since the constrained LMPC yields better results in practice, we will fully adopt this control method for our next two experiments. In this section, walking to the side along the y-axis (side steps) is demonstrated. In Figure 8.8 the response in the y-axis is shown,

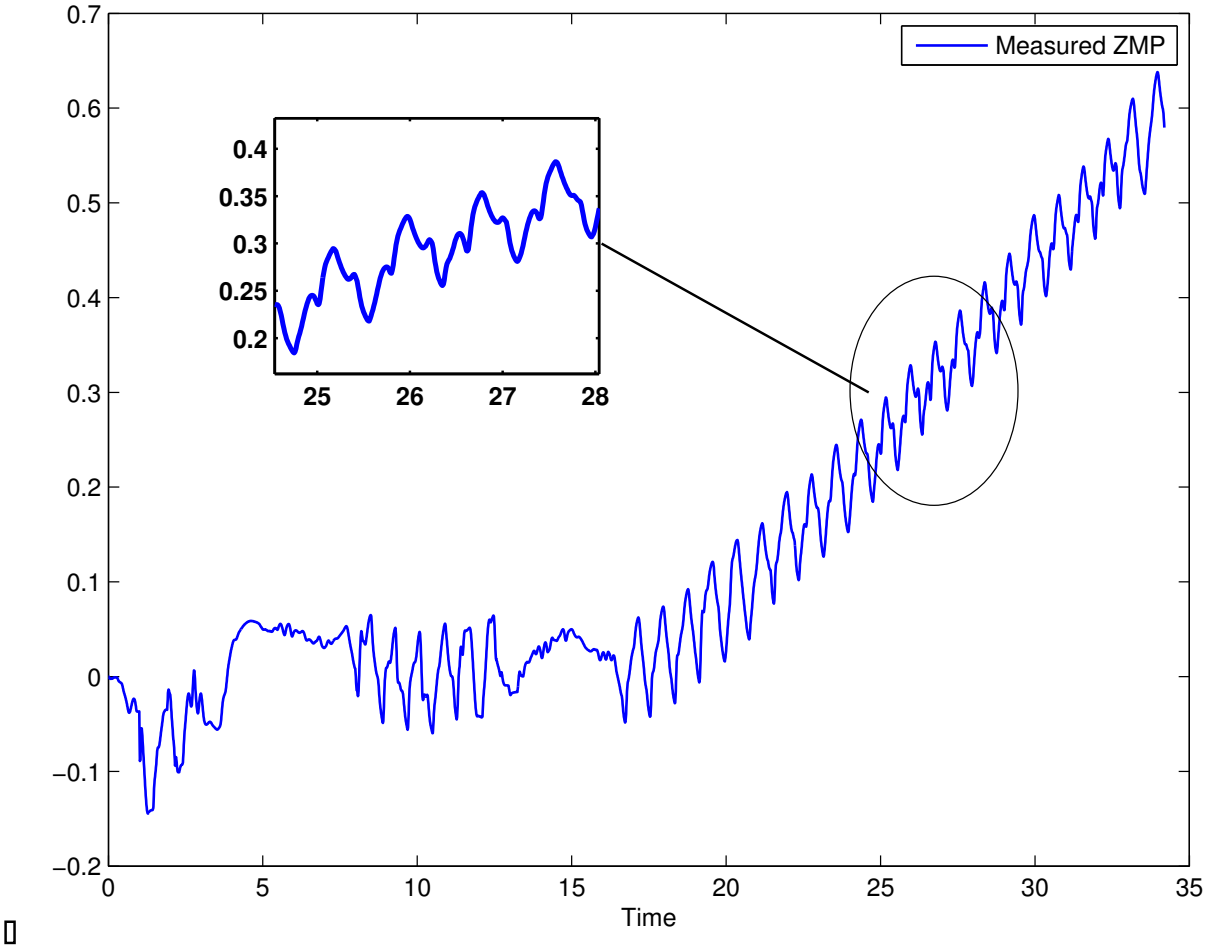

Figure 8.9: y-axis ZMP from the Kalman filter for the side-stepping gait

while in Figure 8.9 the response of the Kalman filter in the same axis is presented.

It is obvious that the ZMP obtained from the Kalman filter is not suitable for walking, because it fails to closely track the reference ZMP. However, when fed to the observer, it is reshaped and the reference is closely tracked.

8.4 Walking and Stopping

Furthermore, a gait, where only a few steps are planned and then Nao has to come to a complete stop, is demonstrated. In Figure 8.10, the y-axis response is shown, where in the start a disturbance is sensed. The ZMP suddenly peaks until it "hits" the constraint

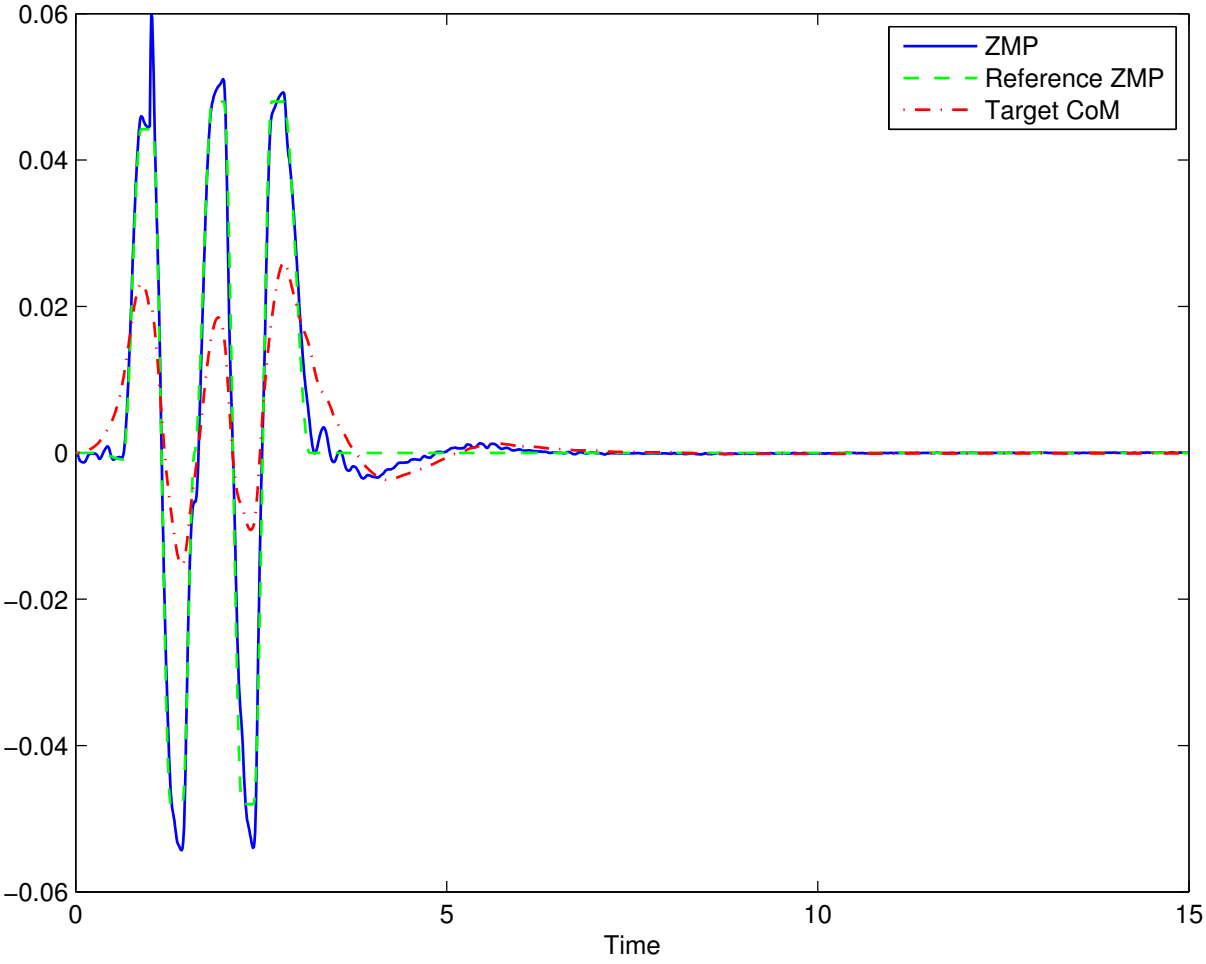

Figure 8.10: y-axis response for a side stepping gait using constrained LMPC on Nao

and returns. After the steps, Nao manages to fully balance itself and stop. In Figure 8.11, the ZMP bias on the y-axis for the previous gait is shown; one may see that, after Nao stops and balances, the bias converges to a constant value.

8.5 Omnidirectional Gait on Uneven Terrain

Finally, the preview control and constrained LMPC are used for an omnidirectional gait, where uneven terrain is encountered. This kind of gait is of most interest, since a humanoid robot must be able to walk in every direction and without tipping over, when stepping over rough ground. In this experiment, the Nao robot achieved a smoother gait,

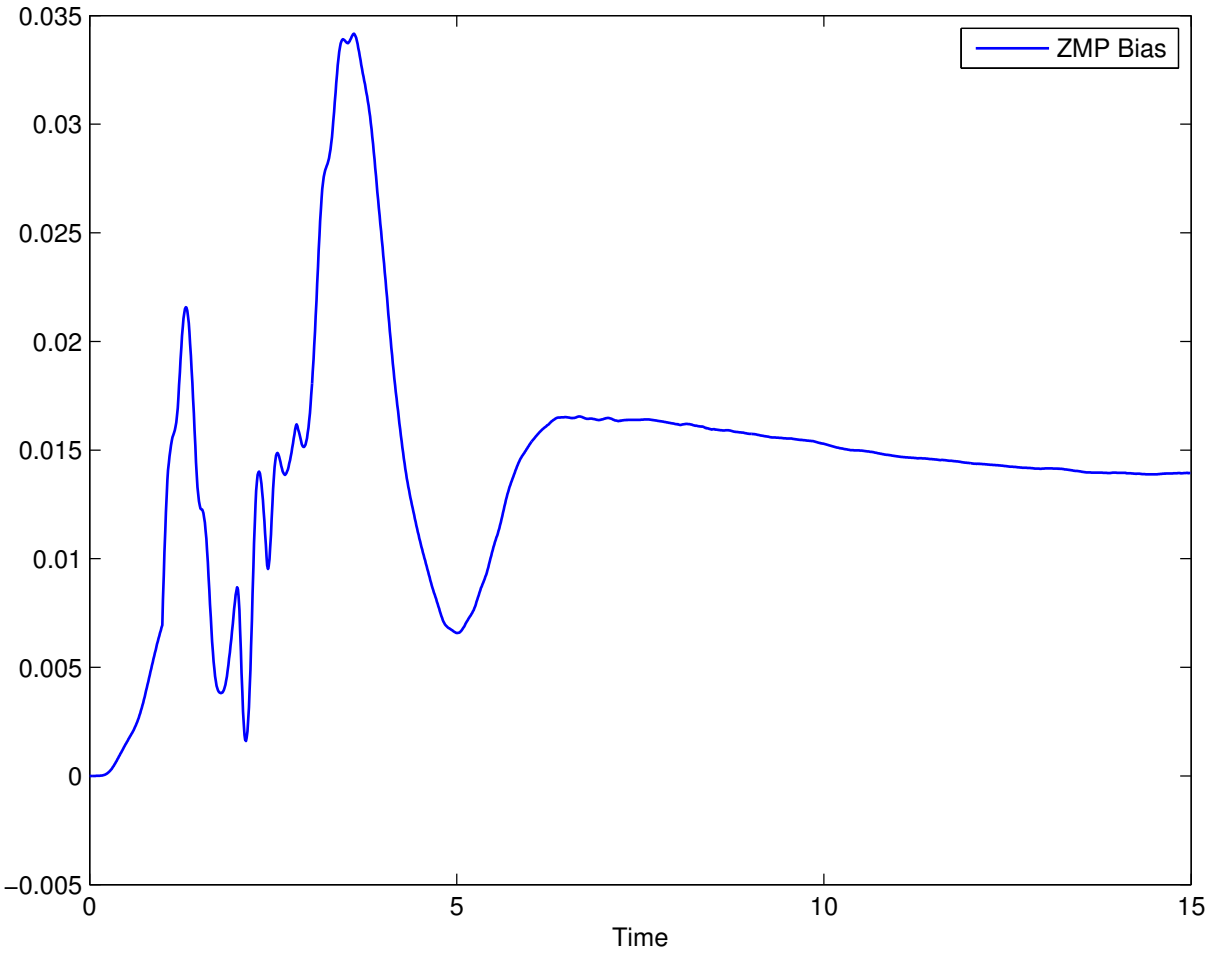

Figure 8.11: y-axis ZMP bias for the previous gait

while transiting from the flat terrain to the uneven terrain and vice versa, when using the constrained LMPC as opposed to the performance achieved by the preview control in the same situation. An illustration of this experiment is shown in Figure 8.12, where the Nao robot stably walks over a cable of pencil size, then climbs over a carpet, where it turns by 90 degrees, and finally climbs down the carpet.

8.6 Conclusion

In this chapter, the control and estimation schemes presented were demonstrated in practice, proving our claims for humanoid dynamic stable locomotion. In the experiments

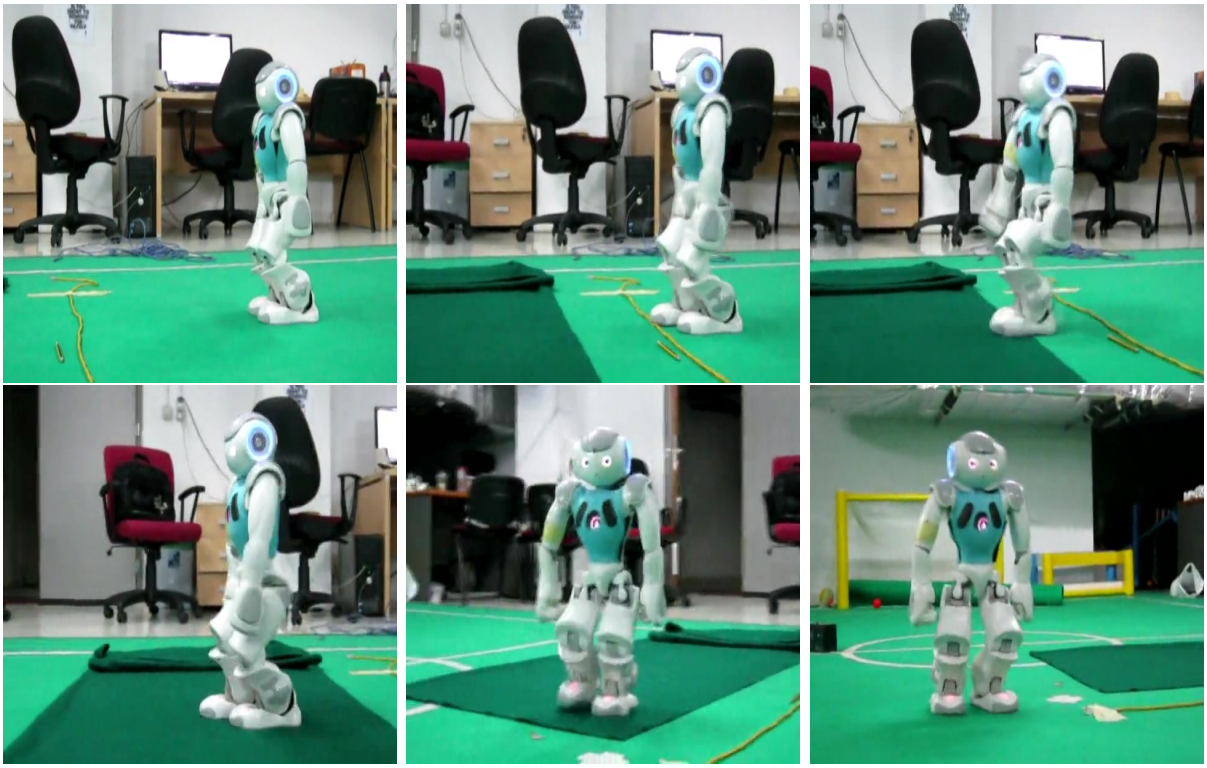

Figure 8.12: Omnidirectional walk on uneven terrain using constrained LMPC on Nao

performed, the Nao humanoid robot was used and both predictive controllers were tested and compared. First, we demonstrated a straight gait, where the constrained LMPC achieved higher speed and smoother locomotion, compared to the preview control. Next, a standing Nao was heavily disturbed, with the constrained LMPC yielding a smooth response, while maintaining balance. Side steps, as well as walk and stop, were further demonstrated with the constrained LMPC. Finally, the Nao was forced to walk omnidirectionally on uneven terrain. In this experiment, the constrained LMPC yielded smooth transitions from flat to uneven terrain and vice versa, maintaining the dynamic stability. Although with the preview control, the balance was lost for a short period of time and Nao started to tremble, the gait was successfully completed without falling to the ground.

Chapter 9

Conclusion and Future Work

All truths are easy to understand once they are discovered; the point is to discover them.

Galileo Galilei (1564 –1642)

9.1 Conclusion

Dynamic locomotion of real humanoid robots is far from being a solved problem; research is still, and will be, carried on for a long time. The quest for achieving human-like behavior in the way humanoid robot locomote, is not straightforward and seems to have a long way to go, until it is considered accomplished. I would like to believe that this work is a step to the right direction.

This thesis presented a variety of methods for controlling the humanoid robot locomotion and for perceiving the environment in a way where disturbances and rough terrain can be sensed and counteracted. First, the well-studied in literature preview control is presented using the Cart and Table model for simplifying the complex non-linear humanoid robot dynamics and ease the computation. The walking patterns generated this way are dynamically stable and can achieve high-speed gaits in practice.

Next, the control system is augmented with a Luenberger observer for the state estimation task, which is of great importance for the state feedback used in the control

loop. In addition, the concept of the auxiliary ZMP was proposed for the Cart and Table model, enhancing the preview control scheme in such a way that disturbances and the presence of uneven terrain can online change the walking patterns accordingly, thus dynamic stability can be maintained.

Furthermore, a novel LMPC framework was proposed, which allows dimensionality reduction of the optimization problem, yielding faster solutions. This approach can be adopted in fast-controlling schemes, such as the real-time controlling of the humanoid robot locomotion. In this LMPC framework, constraints on the gait can be used for reinforcing the dynamic stability of the gait, enduring strong disturbances. The benefit of the constraints is that walking over uneven terrain is accomplished in a more efficient way, compared to the preview control approach. Also, the walking patterns generated are more energy-efficient and smoother, yielding locomotion closer to the human one.

Next interpolation methods were proposed for designing smooth feet trajectory that would grant a smoother overall gait. The methods presented first were the cubic Bezier and cubic B-spline interpolation, which are typically used in planning trajectories. In addition, a new approach was presented, where the foot is treated as a rigid object and trajectories in the SE(3) group are generated. First, the geodesic curves were obtained, consisting of the shortest-path trajectories. Then, the minimum-jerk curves with homogeneous boundary conditions were derived, which were found to be suitable for the locomotion task.

An attempt to apply these control schemes in practice yielded many issues. Sensor readings on a humanoid robot introduced noise, delay, and bias, making the control task inviable. Proper actuation was also non-trivial, even though an analytical inverse kinematics procedure was used. These issues were successfully handled and finally human-like locomotion was achieved on a Nao humanoid robot, fully utilizing the methods proposed.

9.2 Future Work

This complete formulation of humanoid robot walk can be upgraded in a numerous ways. Since the base framework is successfully established, extending it is mandatory for pursuing a locomotion as close as possible to the human one. Below some suggestions are listed. To begin with, we plan to make use of the Intertial Measurement Unit (IMU) readings in the auxiliary ZMP computation, enhancing our preview control scheme with

the approach presented. In addition, we can extend the LMPC framework to make use of online modifications to the CoM height; in such a way, even smoother gaits will be generated. Also, we can include the foot adaptation in the optimization procedure of the constrained LMPC; in such a way, an optimal foot adaptation can be derived. A tilt angle controller of the upper body can be developed for maintaining the upright position of the humanoid robot, using the IMU measurements, and reinforcing the balance while walking. In addition, the simplified dynamics can be changed in such a way that the rotational inertia effects or the Coriolis forces acting on the humanoid robot can be encountered, yielding smoother performance, while omnidirectionally walking. Finally, the ZMP reference trajectory can be improved, using one of the interpolation techniques proposed and obtaining a smoother trajectory, which in practice will allow even faster gaits.

References

[1] Vukobratovic, M., Borovac, B.: Zero-Moment Point - Thirty Five Years of its Life. International Journal of Humanoid Robotics **1**(1) (2004) 157–173

[2] Goswami, A.: Postural Stability of Biped Robots and the Foot Rotation Indicator (FRI) Point. Robotics Research **18** (1999) 523–533

[3] Goswami, A.: Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 1. (1999) 47–52

[4] Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., Morisawa, M.: A Universal Stability Criterion of the Foot Contact of Legged Robots - Adios ZMP. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (2006) 1976–1983

[5] McGeer, T.: Passive walking with knees. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 3. (1990) 1640–1645

[6] Honda Motor Co. Inc.: Inc. Corporate Affairs & Communications American Honda Motor Co. Technical Information. (2003)

[7] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: A real-time pattern generator for biped walking. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 1. (2002) 31–37

[8] Belta, C., Kumar, V.: An Efficient, Geometric Approach to Rigid Body Motion Interpolation. Technical report, University of Pennsylvania, USA (2000)

[9] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent ASIMO: system overview and integration. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Volume 3. (2002) 2478–2483

[10] Hirai, K.: Current and future perspective of Honda humamoid robot. In: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Volume 2. (1997) 500–508

[11] Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., Isozumi, T.: Humanoid robot HRP-2. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 2. (2004) 1083–1090

[12] Inoue, H.: Overview of the humanoid project of METI. In: Proceedings of the 32rd International Symposium on Robotics. (2000) 1478–1482

[13] HUBO Robotics Laboratory: Hubo lab http://hubolab.kaist.ac.kr.

[14] Arbulú, M., Pardos, J., Cabas, L., Staroverov, P., Kaynov, D., Peréz, C., Rodriguez, M., Balaguer, C.: Rh-0 humanoid full size robot's control strategy based on the Lie logic technique. In: Proceedings of the 5th IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS). (2005) 271–276

[15] Bluethmann, W., Ambrose, R., Diftler, M., Huber, E., Fagg, A., Rosenstein, M., Platt, R., Grupen, R., Breazeal, C., Brooks, A., Lockerd, A., Peters, R., Jenkins, O., Mataric, M., Bugajska, M.: Building an autonomous humanoid tool user. In: Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS). Volume 1. (2004) 402–421

[16] Brooks, R.: The Cog Project. Journal of the Robotics Society of Japan, Special Issue (Mini) on Humanoid **15**(7) (1997) 1478–1482

[17] Gouaillier, D., Blazevic, P.: A mechatronic platform, the Aldebaran Robotics humanoid robot. In: Proceedings of the 32nd IEEE Annual Conference on Industrial Electronics (IECON). (2006) 4049–4053

[18] Aldebaran Robotics: Nao documentation (2012) Only available online: www.aldebaran-robotics.com/documentation.

[19] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H.: Robocup: A challenge problem for AI. AI Magazine **18**(1) (1997) 73–85

[20] RoboCup SPL Technical Committee: RoboCup Standard Platform League (NAO) rule book (2013) Only available online: www.tzi.de/spl/pub/Website/Downloads/Rules2013.pdf.

[21] Paraschos, A.: A flexible software architecture for robotic agents. Diploma thesis, Technical University of Crete, Greece (2010)

[22] Orfanoudakis, E.: Reliable object recognition for the RoboCup domain. Diploma thesis, Technical University of Crete, Greece (2011)

[23] Pavlakis, N.: Cooperative global game state estimation for the RoboCup standard platform league. Diploma thesis, Technical University of Crete, Greece (2013)

[24] Kyranou, I.: Path planning for NAO robots using an egocentric polar occupancy map. Diploma thesis, Technical University of Crete, Greece (2012)

[25] Michelioudakis, E.: Dynamic multi-robot coordination for the RoboCup standard platform league. Diploma thesis, Technical University of Crete, Greece (2013)

[26] Kofinas, N., Orfanoudakis, E., Lagoudakis, M.G.: Complete analytical inverse kinematics for NAO. In: Proceedings of the 13th International Conference on Autonomous Robot Systems and Competitions (ROBOTICA). (2013)

[27] Kofinas, N.: Forward and inverse kinematics for the NAO humanoid robot. Diploma thesis, Technical University of Crete, Greece (2012)

[28] Paraschos, A., Spanoudakis, N.I., Lagoudakis, M.G.: Model-driven behavior specification for robotic teams. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2012)

[29] Topalidou-Kyniazopoulou, A., Spanoudakis, N.I., Lagoudakis, M.G.: A CASE tool for robot behavior development. In: RoboCup 2012: Robot Soccer World Cup XVI. Volume 7500 of Lecture Notes in Computer Science. Springer (2013) 225–236

[30] Topalidou-Kyniazopoulou, A.: A CASE (computer-aided software engineering) tool for robot-team behavior-control development. Diploma thesis, Technical University of Crete, Greece (2012)

[31] Vazaios, E.: Narukom: A distributed, cross-platform, transparent communication framework for robotic teams. Diploma thesis, Technical University of Crete, Greece (2010)

[32] Karamitrou, M.: KMonitor: global and local state visualization and monitoring for the Robocup SPL league. Diploma thesis, Technical University of Crete, Greece (2012)

[33] Vukobratovic, M., Borovac, B., Potkonjak, V.: Zmp: A review of some basic misunderstandings. Humanoid Robotics **2** (2006) 153–175

[34] Vukobratović, M., Herr, H.M., Borovac, B., Raković, M., Popovic, M., Hofmann, A., Jovanović, M., Potkonjak, V.: Biological principles of control selection for a humanoid robot's dynamic balance preservation. Humanoid Robotics **5**(4) (2008) 639–678

[35] Vukobratović, M.: Biped Locomotion. Springer-Verlag New York, Inc. (1990)

[36] Borovac, B., Vukobratović, M., Surla, D.: An approach to biped control synthesis. Robotica **7**(3) (1989) 231–241

[37] Vukobratovic, M., Juricic, D.: Contribution to the Synthesis of Biped Gait. IEEE Transactions on Biomedical Engineering **BME-16**(1) (1969) 1–6

[38] Lee, W., Raibert, M.: Control of hoof rolling in an articulated leg. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 2. (1991) 1386–1391

[39] Graf, C., Hartl, E., Rofer, T., Laue, T.: A Robust Closed-Loop Gait for the Standard Platform League Humanoid. In: Proceedings of the 4th Workshop on Humanoid Soccer Robots in Conjuction with IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS). (2009) 30–37

[40] Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H.: The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Volume 1. (2001) 239–246

[41] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of Zero-Moment Point. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 2. (2003) 1620–1626

[42] Kajita, S., Tani, K.: Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (1991) 1405–1411

[43] Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems **57**(8) (2009) 839–845

[44] Arbulú, M.: Stable locomotion of humanoid robots based on mass concentrated model. Ph.d. thesis, Universidad Carlos III de Madrid, Spain (2008)

[45] Katayama, T., Ohki, T., Inoue, T., Kato, T.: Design of an optimal controller for a discrete-time system subject to previewable demand. International Journal of Control **41**(3) (1985) 677–699

[46] Sheridan, T.: Three models of preview control. IEEE Transactions on Human Factors in Electronics **HFE-7**(2) (1966) 91–102

[47] Kajita, S., Morisawa, M., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K., Hirukawa, H.: Biped walking pattern generator allowing auxiliary ZMP control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2006) 2993–2999

[48] Choi, Y., You, B.J., Oh, S.R.: On the stability of indirect ZMP controller for biped robot systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Volume 2. (2004) 1966–1971

[49] Kim, J.H.: Walking pattern generation of a biped walking robot using convolution sum. In: Proceedings of the 7th IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS). (2007) 539–544

[50] Dorf, R.C., Bishop, R.H.: Modern Control Systems. 9th edn. Prentice-Hall (2000)

[51] Phillips, C., Nagle, H.: Digital control system analysis and design. Prentice-Hall (1984)

[52] Luenberger, D.: An introduction to observers. IEEE Transactions on Automatic Control **16**(6) (1971) 596–602

[53] Urbann, O., Tasse, S.: Observer based biped walking control, a sensor fusion approach. Autonomous Robots **35**(1) (2013) 37–49

[54] Strom, J., Slavov, G., Chown, E.: Omnidirectional walking using ZMP and preview control for the NAO humanoid robot. In: RoboCup 2009: Robot Soccer World Cup XIII. Volume 5949 of Lecture Notes in Computer Science. Springer (2010) 378–389

[55] Sain, M., Massey, J.: Invertibility of linear time-invariant dynamical systems. IEEE Transactions on Automatic Control **14**(2) (1969) 141–149

[56] Piperakis, S.: Flying vehicle model predictive control. Diploma thesis, University of Patras, Greece (2012)

[57] Maciejowski, J.: Predictive Control: With Constraints. Prentice Hall (2002)

[58] Wang, L.: Model Predictive Control System Design and Implementation Using MATLAB. 1st edn. Springer (2009)

[59] Alcaraz-Jiménez, J., Herrero-Pérez, D., Martínez-Barberá, H.: Robust feedback control of ZMP-based gait for the humanoid robot Nao. The International Journal of Robotics Research **32**(9–10) (2013) 1074–1088

[60] Wieber, P.B.: Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. In: Proceedings of the 6th IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS). (2006) 137–142

[61] Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., Diehl, M.: Online Walking Motion Generation with Automatic Foot Step Placement. Advanced Robotics **24**(5-6) (2010) 719–737

[62] Dimitrov, D., Wieber, P.B., Stasse, O., Ferreau, H., Diedam, H.: An optimized Linear Model Predictive Control solver for online walking motion generation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (2009) 1171–1176

[63] Dimitrov, D., Sherikov, A., Wieber, P.B.: A sparse model predictive control formulation for walking motion generation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2011) 2292–2299

[64] Luenberger, D., Ye, Y.: Linear and Nonlinear Programming. Springer (2008)

[65] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)

[66] Zefran, M., Kumar, V.: Planning of smooth motions on SE(3). In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Volume 1. (1996) 121–126

[67] do Carmo, M.P.: Riemannian Geometry. Birkhäuser Boston (1992)

[68] Park, F.C., Brockett, R.W.: Kinematic Dexterity of Robotic Mechanisms. International Journal of Robotic Research **13** (1994) 1–15

[69] Zefran, M., Kumar, V., Croke, C.: Choice of Riemannian Metrics for Rigid Body Kinematics. In: Proceedings of the ASME Design Engineering Technical Conference and Computers in Engineering Conference. (1996) 1–11

[70] Lee, J.: Introduction to Topological Manifolds. Springer (2011)

[71] Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press (1994)

[72] Noakes, L., Heinzinger, G., Paden, B.: Cubic splines on curved spaces. IMA Journal of Mathematical Control and Information **6**(4) (1989) 465–473

[73] Flash, T., Hogans, N.: The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. Journal of Neuroscience **5** (1985) 1688–1703

[74] Kofinas, N., Orfanoudakis, E., Lagoudakis, M.G.: Complete analytical forward and inverse kinematics for the nao humanoid robot. Intelligent and Robotic Systems (2014) 1–14

Appendix A

The Step-Planner

An autonomous robotic agent should not be aware how its low-level functionality works; algorithms using sensors and actuators should be encapsulated. Therefore, the locomotion must be based on a high-level decision; such decisions should correspond to questions like "where to locomote?" and "how fast?". Thus, a medium stage is needed, translating desired gait velocity into feet placement from which suitable reference ZMP and feet trajectories can be obtained.

This appendix provides a formulation for computing where the foot should be placed in the $x - y$ plane when an input velocity vector is provided. The formulation is based on an empirical rule derived by observation of human locomotion biomechanics. When we take a step, our pelvis tends to approximately lie in the middle of the convex hull created by the two feet. Thus, if we further took an imaginary half-step with the other leg in the same direction, our pelvis would align with the swing foot. In such a way, a humanoid robot could empirically infer where to place the swing foot, according to the average speed it decided to move with.

The pelvis position and orientation (x_p, y_p, θ_p) after taking a step with step duration T_{step} and with desired average velocity (u_x, u_y, ω_z), where u_x, u_y are the translational velocities in the horizontal plane and ω_z is the rotational velocity about the z-axis, perpendicular to the horizontal plane, will be

$$x_p = u_x T_{step}$$
$$y_p = u_y T_{step}$$
$$\theta_p = \omega T_{step}$$

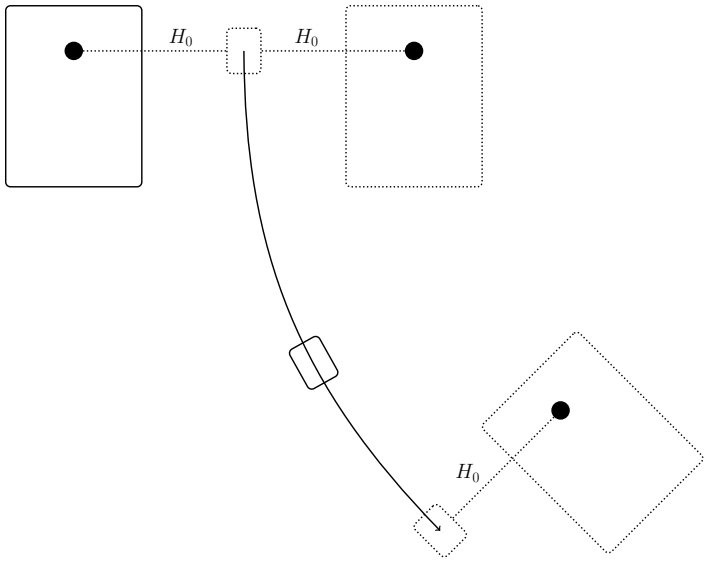

Figure A.1: Translating the desired velocity vector into steps.

In such a way, the imaginary pelvis location will be

$$x_{ip} = 1.5 u_x T_{step}$$
$$y_{ip} = 1.5 u_y T_{step}$$
$$\theta_{ip} = 1.5 \omega T_{step}$$

After having computed the imaginary pelvis position and orientation, we can determine the foot location, since the foot must be parallel to the imaginary pelvis. This procedure is demonstrated in Figure A.1.

Thus, having determined the feet location in the $x - y$ plane, the reference feet trajectory can be obtained with the methods presented in Chapter 6. Furthermore, the ZMP reference signals needed in Chapter 5 can be obtained by the following three rules:

1. If no stepping occurs, the ZMP must lie in the middle of the convex hull created by the feet.

2. If the humanoid is in the SS phase, the ZMP must lie in the center of the support foot.

3. When the humanoid is transitioning from SS to DS, the ZMP must move from the previous support to the next support foot.

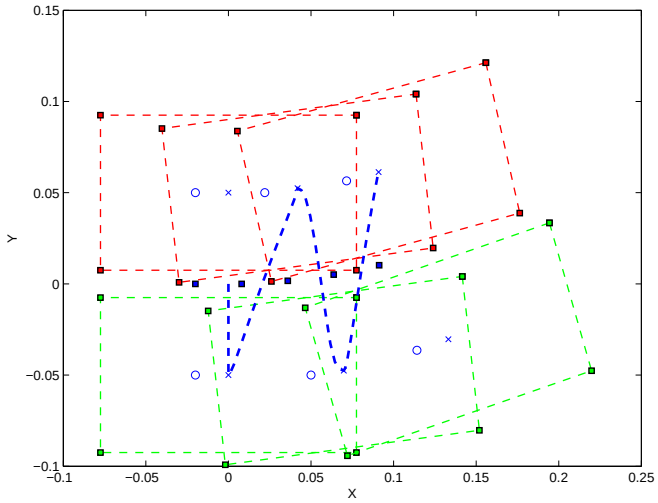

Figure A.2: A gait of 4 consecutive steps with velocity $u_x = 2.0$, $u_y = 0$ and $\omega_z = 0.5$; the blue line indicates the ZMP reference trajectory, the circles the step centers, the x's the ankle positions, and the blue cubes the pelvis positions.

In Figure A.2, the reference ZMP signal in the $x - y$ plane for a predefined gait is shown, while, in Figure A.3, the ZMP signal is decoupled in its x and y components.

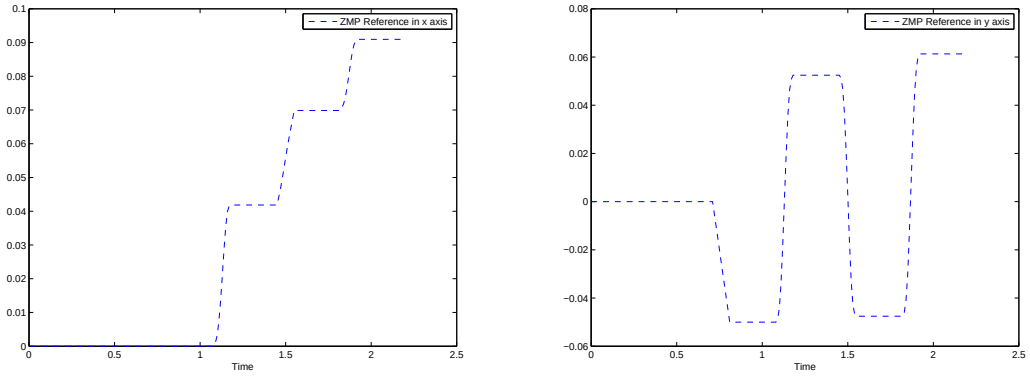

Figure A.3: The reference ZMP of the previous gait is decoupled in x and y axes