

AN ENERGY EFFICIENT GAIT FOR HUMANOID  
ROBOTS WALKING ON EVEN AND UNEVEN  
TERRAINS

Dissertation

to obtain the degree of Doctor at  
Maastricht University,  
on the authority of the Rector Magnificus,  
Prof. dr. Rianne M. Letschert  
in accordance with the decision of the Board of Deans,  
to be defended in public  
on Wednesday 27 March, 2019 at 14:00 hours

by

Zhenglong SUN

Promotor:

Prof. dr. ir. R.L.M. Peeters

Copromotor:

Dr. ir. ing. N. Roos

Assessment Committee:

Prof. dr. G.B. Weiss (chair)

Prof. dr. K.V. Hindriks (Vrije Universiteit Amsterdam)

Dr. ir. K. Meijer

Prof. dr. F. Thuijsman

Dr. A. Visser (Universiteit van Amsterdam)



*The research reported in this thesis was financially supported by the China Scholarship Council (under the grant No. 2011607008), and the Department of Data Science and Knowledge Engineering, Maastricht University.*

Printed by: Datawyse, Maastricht

ISBN 978-94-6380-280-2

Copyright © 2019 by Zhenglong Sun, Maastricht, The Netherlands.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically photocopying, recoding or otherwise, without prior permission of the author.

*Science is a wonderful thing  
if one does not have to earn one's living at it.*

Albert Einstein



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Humanoid Robots	1
1.2	History and State of the Art	2
1.3	Biped Walking and Its Challenges	6
1.3.1	Stability	8
1.3.2	Energy Efficiency	8
1.4	Problem Statement and Research Questions	9
1.5	Thesis Overview	11
<b>2</b>	<b>Platform Overview</b>	<b>13</b>
2.1	Introduction of NAO Robot	13
2.2	Hardware Specification	13
2.2.1	Configuration	13
2.2.2	Sensors	14
2.2.3	Computer Hardware	17
2.3	Software Architecture	17
2.3.1	The NaoQi Distributed Architecture	20
2.3.2	The Device Control Manager (DCM)	21
2.3.3	Actuator Stiffness	22
2.4	Kinematics	22
2.4.1	Affine Transformations	23
2.4.2	Coordinate Systems	26
2.4.3	Forward Kinematics	28
2.4.4	Inverse Kinematics	32
2.5	Conclusion	35
<b>3</b>	<b>Walking Gait Generation of Humanoid Robots</b>	<b>37</b>
3.1	Introduction	37
3.2	Humanoid Robot Modeling	37
3.2.1	The Inverted Pendulum Model	38
3.2.2	The Linear Inverted Pendulum Model	40
3.3	Stability Criteria	42
3.3.1	Static Stability	42
3.3.2	Dynamical Stability	43
3.4	Walking Gait Generator	46
3.4.1	Passive Dynamic Walking	47
3.4.2	Model Free Walking Gait Generation	48
3.4.3	Model-Based Walking Gait Generation	51
3.4.4	Human-Inspired Approaches	53

3.5	Conclusion . . . . .	54
<b>4</b>	<b>An Inverted Pendulum Model for Gait Analysis</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Kinematic Model in the Sagittal Direction . . . . .	57
4.3	Kinematic Model in the Lateral Direction . . . . .	64
4.4	Experiments . . . . .	67
4.4.1	Validating Assumptions . . . . .	67
4.4.2	Effects of the Force Polices on the Leg-length Policy . . . . .	69
4.5	Conclusion . . . . .	70
<b>5</b>	<b>Gait Controller Design for Flat Floor Walking</b>	<b>73</b>
5.1	Introduction . . . . .	73
5.2	Overview of the Gait Controller Design . . . . .	74
5.2.1	Finite State Machine . . . . .	75
5.2.2	Leg Trajectory Generation . . . . .	75
5.2.3	Body Trajectory Generator . . . . .	80
5.3	Controller Parameters and Fitness Function . . . . .	81
5.4	Algorithm for Learning Controller Parameters . . . . .	83
5.5	Learning Optimal Parameters in the Webots Simulator . . . . .	85
5.6	Experimental Validation . . . . .	89
5.6.1	Gait Analysis . . . . .	89
5.7	Conclusion . . . . .	95
<b>6</b>	<b>Walking on Uneven Terrains</b>	<b>97</b>
6.1	Introduction . . . . .	97
6.2	Uneven Terrains Versus Slopes . . . . .	97
6.3	Adaptive Controller Design . . . . .	98
6.3.1	Analysis of the Single Support Phase . . . . .	99
6.3.2	The Sagittal Controller . . . . .	100
6.3.3	Analysis of the Double Support Phase . . . . .	102
6.3.4	The Lateral Controller . . . . .	102
6.4	Gait Controller . . . . .	103
6.5	Experiments . . . . .	105
6.5.1	Uneven Terrains . . . . .	105
6.5.2	Evaluation of Stability . . . . .	107
6.5.3	Evaluation of Energy Efficiency . . . . .	109
6.6	Conclusion . . . . .	112
<b>7</b>	<b>Conclusions and Future Research</b>	<b>113</b>
7.1	Answers to the Research Questions . . . . .	114
7.2	Answer to Problem Statement. . . . .	117
7.3	Recommendations for Future Research . . . . .	118

---

<b>References</b>	<b>121</b>
<b>Summary</b>	<b>137</b>
<b>Valorization</b>	<b>143</b>
<b>Acknowledgements</b>	<b>145</b>
<b>About the Author</b>	<b>147</b>
<b>List of Publications</b>	<b>149</b>





# 1

## INTRODUCTION

### 1.1. HUMANOID ROBOTS

**H**UMANOID robots, robots with anthropomorphic shape are popular as the research topics. More and more researchers are working on issues such as bipedal locomotion [46, 125, 148], human-robot interaction[50, 87, 113, 130], imitation learning[10, 134], etc. aimed at the application in humanoid robots. These researchers are motivated by the idea of creating a machine that can work in cooperation with humans in the environment designed for humans. Highly specialized industrial robots are widely employed in factories, and robots, like drones and vacuum cleaners, become more and more popular in consumer electronics markets. General humanoid robots possess an even greater potential in future application of robotics.

There are several reasons why we might want to develop robots with humanoid form. An important reason for creating robots in the anthropomorphic form, is the interaction with humans. A robot may gain experience through interactions with humans[68, 108, 109]. If a robot has humanoid form, humans will feel more natural to interact with the robot. For example, if a human-like robot is able to synthesize emotion by eye movements, body language etc., it may use the human body language in its communication with humans. The robot may learn the human body language through a large number of interaction with humans.

A second reason for building a humanoid form robot, stands on firmer ground. The human-like body is suited for daily use in human-inhabited environment where stairs, narrow corridor, door handles and so on are designed for humans. A robot with anthropomorphic shape, which can employ hand-like actuators to manipulate the objects, can take advantage of these human-oriented designs. The benefits of the adaptation to human environments is not limited to the hand. Legs also are paramount for practical application of robots. Compared to the wheeled robot, which are able to move in a high efficient way on even surfaces, humanoid robot with legs are capable of stepping across the obstacles on uneven terrains. Last but not least, humanoid robots can be used as a new platform to test certain level of human intelligence or develop medical prosthesis

and exoskeletons, because of the same biometric way the robot has been built. However, addressing all of the above areas exceeds the current cutting-edge knowledge in robotics.

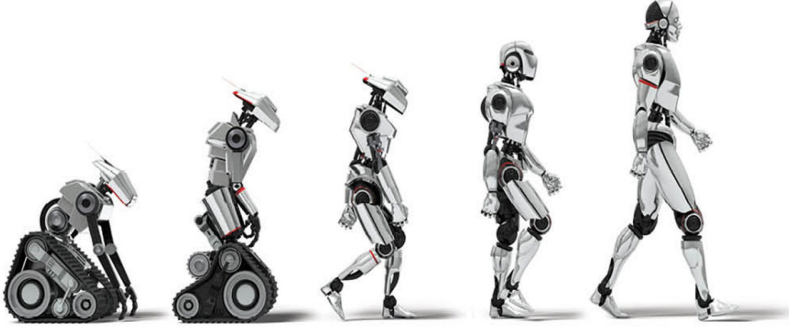


Figure 1.1: The fictional evolution roadmap of robots. (Copyright by Ryan Etter)

## 1.2. HISTORY AND STATE OF THE ART

In the field of humanoid robots, significant progress has been made in many aspects, such as the design of hardware platform and the control of bipedal systems. In this section we highlight several important developments in bipedal robot walking. We mention developments in walking pattern generation, stability control and illustrate this with examples of robots that apply these techniques. Table 1.1 lists some milestones in the development of humanoid robots.

Seven types of robots can be distinguished based on the applied method of stability control and walking pattern generation.

- The static walker
- The fully-actuated dynamic walker
- The passive dynamic walker and the under-actuated dynamic walker
- The walker based on zero dynamics
- The walker based on virtual model control
- The walker based on coupled neural oscillators
- The human-inspired walker

The *static walker* uses a static stability criterion to ensure that the robot does not fall. A static stability criterion requires that the robot's center of mass is always within the

Table 1.1: The information of several milestone humanoid prototypes since 1970

Model	Released year	Features
WABOT-1	1970	The world's first full-scale humanoid robot.
McGeer's passive walker	1990	A biped walker with knees could walk passively on a down slope.
P1	1993	A robot with upper limbs to pick up and carry objects.(Figure.1.2(d))
ASIMO	2000	Running, and hopping on one leg or both legs.
HRP series	2002	A light-weight full-sized biped walking robot.
KHR-1	2004	A programmable bipedal humanoid robot with external disturbances rejection.
Flame	2007	3D robot with elastically actuated joints and the ability to do side-ways foot placement.
Nao	2008	An affordable mass-produced biped robot, the standard platform in SPL.
Atlas	2013	Specialized for mobile manipulation and walking over a wide range of terrain.

area on the ground that supports the foot/feet which is/are on the ground. The static stability guarantees stability if acceleration forces are small enough to be ignored.

The most of static walkers were build in the initial stage of the development of humanoid robots. These robots were heavy and slow due to the limitation of the hardware design, the lack of advanced motors, and the use of a static stability criterion. One of the first bipedal robots in this type is WABOT-1 (Figure.1.2(a)), which was developed in 1970 by the researchers from Waseda University Japan. WHL-11 (Figure.1.2(b)) is another biped robot of this type, developed by Kato et al. [66], which is capable of static walking on a flat surface. The robot needs 10 seconds to finish a step. Despite of the slow motion, the robot made a stable walk of 40 kilometers in 6 months.

An important step in stability control was made 1972 when Vukobratovic proposed the zero moment point (ZMP) stability criterion [157]. The ZMP stability criterion makes it possible to take the accretion forces of a moving robot into account. It allows for the development of robots that can walk with much higher speeds than those using a static stability criterion. An important requirement of the ZMP stability criterion is that the

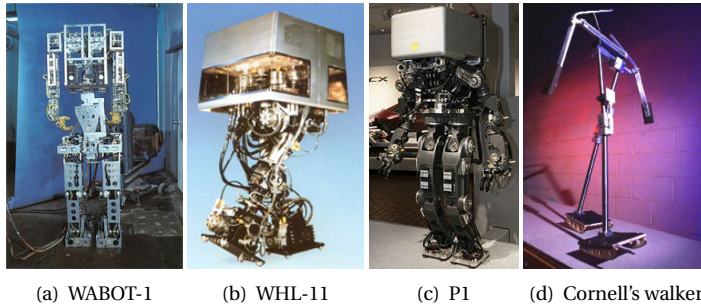


Figure 1.2: Some pioneer prototypes of humanoid platform.

robot has full actuation and control over each degree of freedom. Therefore, such a robot is called *fully-actuated dynamic walker*. The fully-actuated dynamic walker can control its motion along a desired joint trajectory.

Robots with actuation and control of all joints have been developed by researchers, and by companies. For example, the P series and ASIMO (Figure 1.3(a)) [62] developed by HONDA, the Wabian series developed by Waseda University (Figure 1.3(b)), the HRP series [78, 80, 81] built by Japanese National Institute of Advanced Industrial Science and Technology (AIST) (Figure 1.3(c)), KHR series (Figure 1.3(d)) is the latest prototype HUBO2 from Korea Advanced Institute of Science and Technology (KAIST), Atlas developed by Boston Dynamics (Figure 1.3(f)), the robot H6 [117] and H7 (Figure 1.3(g)) [89] from University of Tokyo, Jonnie and Lola (Figure 1.3(h)) [96] from the Technical University of Munich, Sony's Qrio (Figure 1.3(i)) [69], and the Nao (Figure 1.3(j)) robot manufactured by Aldebaran. The latter is the platform used in this research.

The generation of a walking pattern using ZMP stability criterion can be realized by using an accurate physical model of the robots. Such approaches are usually computationally expensive. Another approach to generate a walking pattern uses an abstract model of the robots, such as an inverted pendulum. To compensate for the lack of an accurate physical model, feedback control is used. An important contribution to the walking pattern generation based on feedback control is the linear inverted pendulum mode (LIPM), which was proposed by Kajita and Tan in 1991 [77]. The 3D linear inverted pendulum mode (3D-LIPM) described by Kajita et al. in 2001 [74] is used in several robots such as the Nao robot.

Another parallel development started in 1990 when McGeer published his paper about a *passive dynamic walker*. McGeer showed that it is possible to create a bipedal robot that can walk down a slope without actuators and without stability control [100]. Although the gait of the passive dynamic walker is stable, the robot is not in a stable state during a step. McGeer implemented a passive dynamic walker as 2D walker without [101] and with knees [102]. An advantage of robots based on McGeer's passive dynamic walker is the low energy consumption compared to robot that use the ZMP stability criterion and the absence of a controller ensuring the stability of the robot.

After McGeer, several researchers have presented their work on passive dynamic walking. Goswami et al. [52] studied dynamic characteristics of a simple compass-type biped

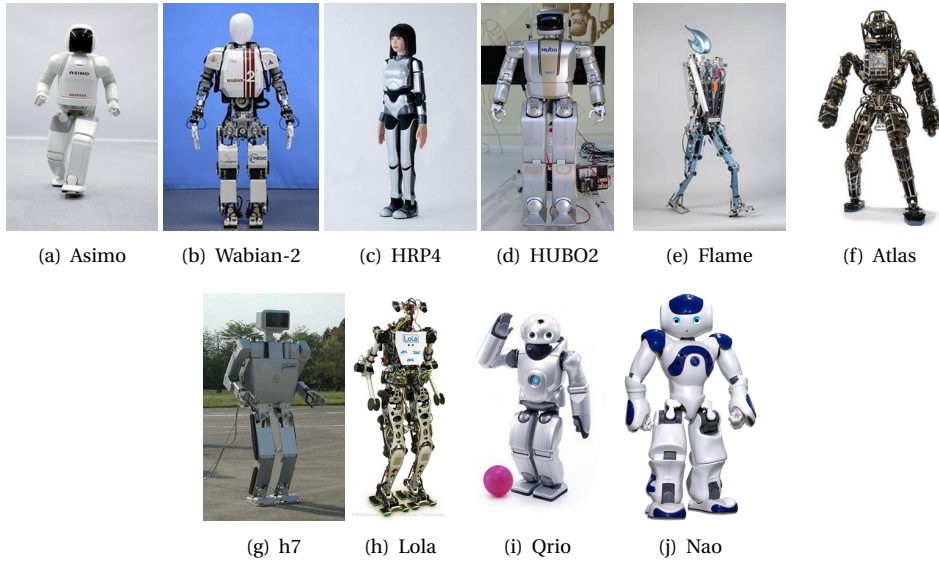


Figure 1.3: Several modern prototypes of humanoid robots developed by companies and institutes.

mechanism on inclined plane as limit cycle. Garcia et al. [45] performed the simulation studies on fully passive models and showed the models have stable cyclic motion. Collins et al. [28] built the first three-dimensional, two-legged passive dynamic walking machine with knees (Figure.1.2(c)). Van der Linde [155] built another prototype, which extended the passive dynamic walker, by introducing pneumatic artificial McKibben muscles. These muscles behave like springs of which the stiffness can be varied.

The success of the passive dynamic walker motivated researcher to develop a partially passive dynamic walker, for which the objective is to use minimal actuation (most of joints are passive) and control to achieve highly efficient walking on flat ground. Examples of machine designs based on passive walkers, extended with low-power actuators as the energy source rather than gravity, are the Cornell biped robot develop at Cornell University [26], and Flame developed at Delft University of Technology (Figure 1.3(e)). The features of these robots are that they use less control hardware with less energy to achieve a more natural gait than fully-actuated counterparts. Though robots based on the passive dynamic walker have drawbacks such as a limited range of walking motions and inadequate robustness of control, they generally excel in the energy efficiency compared to the fully-actuated counterparts.

The *walker based on zero dynamics* uses control theory for non-linear systems to control an under-actuated walker [19, 21, 22]. The reason for studying under-actuated walkers is because (1) fully-actuated walking cannot recover from large disturbances, and (2) human walking has under-actuated phases because rolling over the foot and the toe-off. The walker based on zero dynamics assumes that the robot has point feet, which can be realized by either removing the feet or by using freely rotating ankle joints. To control the resulting non-linear system, an accurate model of the dynamics of the whole

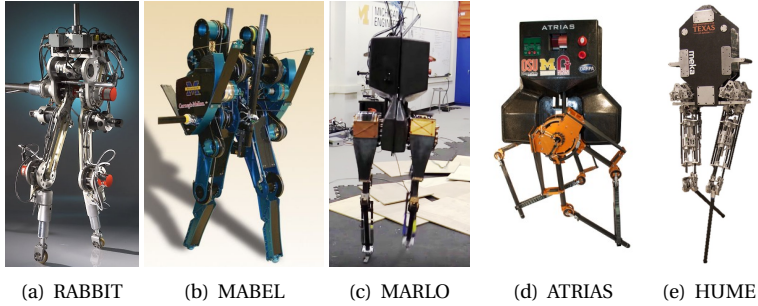


Figure 1.4: Dynamic walkers with pointy feet: (a) RABBIT, (b) MABEL, (c) MARLO, (d) ATRIAS, (e) HUME

robot is needed. A controller based on zero-dynamics is used to control the stability of the model.

Several research institutions have developed this kind of robots, such as RABBIT built as part of the French National Project ROBEA ((Figure 1.4(a)), MABEL engineered in 2009 by researchers at the University of Michigan ((Figure 1.4(b)), MARLO and ATRIAS from Dynamic Legged Locomotion Lab at Oregon State University ((Figure 1.4(c-d)), HUME commissioned by UT Austin ((Figure 1.4(e)), etc.

Beside the four types of walkers described above, there are currently a few other approaches that have been used to build a dynamic walker. The *walker based on virtual model control* [125] is one such an approach. It has been used to build a planar walker<sup>1</sup>. The idea is to add virtual components, consisting of springs and dampers, to the robot model and translate the forces generated by the virtual components to torques by the robots joints. For instance a spring-damper that keeps the robot's body at a specified height and a spring-damper that pulls the robot forward with a constant speed.

Another approach is the *walker based on coupled neural oscillators* [38, 39]. Neural oscillators that produce oscillating signals, are used to control the robot's joints. By coupling the oscillators of different joints, a walking pattern is generated. The approach has been tested on a planar walker.

A last approach that we mention here is the *human-inspired walker*. The human-inspired walker uses observations of human walking patterns to generate walking patterns for robots [2–4, 165, 166]. Moreover, zero dynamics is used in the generation of the walking patterns. The resulting walking patterns have been evaluated on a real robots, such as the Nao robot.

### 1.3. BIPED WALKING AND ITS CHALLENGES

The development of the humanoid robots provides the robot companies and research institution with an effective means to show their technological disposition. It is expected that the human-like form of locomotion (i.e. biped walking) will also prove useful, especially in environments that are designed for humans, which makes a humanoid robot a more versatile machine. The quality of biped walking, such as velocity, flexibility, etc. has

<sup>1</sup>A planar walker is a robot that is stabilized in the lateral direction by some external devise.

a great impact on the applicabilities of humanoid robots, and therefore deserves major research investments.

A biped, a class of walking robots that imitates human locomotion, is an open kinematic chain consisting of two sub-chain legs that connected at a hip joint. When only one leg contacts with ground, the contacting leg is referred to as stance leg and the other one is called the swing leg. The end of leg is called foot, whether it has a foot or not. The single support phase (SSP) is defined as the phase of locomotion where only one foot is on the ground. The double support phase (DSP) is defined as the phase of locomotion where both feet are on the ground. Therefore, biped walking cycle can be defined as alternating phases of single and double support.

Biped walking for humanoid robots is one of the important challenges in robotics. Over the past decades, certain specific aspects of the biped walking research have emerged which have accumulated considerable research interests. Much research and funding has been put into biped robots, which has resulted in several successful machines, but most of them have never been outside a laboratory. This is mainly due to the fact that the control of a biped machine is a very complex issue. We can categorize these difficulties into following challenges:

1. Non-linear dynamics and kinematics:

The highly non-linear dynamics and kinematics of the humanoid robot is one of the major challenges. The non-linear feature of the dynamics and kinematics require the complex models to accurately describe the locomotion, which is not always feasible. Besides, modeling the dynamics is difficult because the dynamics changes in the different support phases, i.e. the single support phase and the double support phase.

2. Robustness in unknown and dynamic environment:

To enable robust locomotion in unknown and dynamic environment, robot needs to identify a collision-free path from the current to the target position. This can be achieved by combining the robot's walking control with a computer vision system capable of recognizing obstacles to choose a viable path, or with inertial and force sensors capable of providing essential data about the robot's real-time state for balance state recovery. This thesis does not address of computer vision. However, to enable autonomous locomotion, the real-time control system must supply information about the robot's state and be able to execute continuously changing commands while walking.

3. Behavioral Complexity:

The motion of Biped walking is a complex set of combined motion sequences, especially in a flexible and highly maneuverable robotics system. The essential motions of the walking, such as turning, acceleration, de-acceleration, stopping, require accurate and real-time motor control, which normally is combined with sensory input and high level cognition.

4. High degrees of freedom:

Humanoid robot is a high DOF system. For example, Honda ASIMO has 34 DOF, and HUBO2 has total 40 DOF. When we wish to implement locomotion behaviour on humanoid robots, controlling high DOF is an important issue.

Besides these challenges, this work will focus on two other difficulties of the biped walking, namely stability and energy efficiency.

### 1.3.1. STABILITY

Humanoid robot employ legs for intermittent contacts with ground to propel and sustain themselves. This makes the biped walking an unstable locomotion and therefore the robustness of the walking strategy is a paramount feature.

There are a few mechanisms that are widely used in stabilizing biped locomotion, such as Foot Rotation Point (FRI) [51], Central Pattern Generator (CPG) based approaches [6, 91, 116, 128], force-torque controlled approaches [147, 148], reflex-like mechanisms [121], etc. The probably most well-known approach to control the walk behaviour and balance of a robot is manipulation of the concept of the Zero Moment Point (ZMP) [157]. The ZMP is a point on the ground where the resultant of the ground reaction force acts (see Subsection 3.3.2). The stability of a locomotion can be guaranteed when the ZMP is within the convex hull of the support area of the robot. The state of the art humanoids such as: ASIMO [133], HRP-4 [81] and HRP-3 [80] make use of the ZMP concept. One of drawbacks of ZMP arises from the need to have the whole foot in contact with a flat surface. Besides, this method requires the full model of the robot and the environment to be known. It is not easy for a humanoid robot to directly manipulate the ZMP. Therefore, control methods have to be developed to indirectly influence the ZMP so that it stays within the stable region.

### 1.3.2. ENERGY EFFICIENCY

Since the biped robot need to carry its energy source and since the capacity of the source is limited, a low energy consumption can directly contribute to a longer working duration. Especially, advanced biped robots consume more energy for motion control, compared to other kind of mobile robots [7]. The necessity of energy efficiency has been recognized by the research community and is an important research topic.

There exists a type of biped walker called passive walker [101] that is able to trade the potential energy at the upper part of the slope for energy losses during walking. The most fundamental cause of the energy losses are impact and friction, primarily between the feet and the ground. Because it needs no energy for stabilization or control, the walker operating passive dynamically can be energy efficient. Inspired by the simple passive biped model, [52] and [28] present their results of low energy gait for more complex robots. To be able to walk over level ground minimal actuation is provided, just enough to overcome friction when walking on level ground [152], like the Delft biped Denise [26].

Another attempt to optimize the energy consumption of the biped robot is the introduction of compliant actuators. Experiments [15] show that in human locomotion the compliance is used to store energy during early stance and then recovering it near the end of stance. This makes it possible for the whole body to operate in an efficiency way



during the locomotion. The compliant joint can store motion energy, reduce control effort and absorb shocks. Similar to human locomotion, it is evident that compliance joint can play an important role in biped robot locomotion.

There are two main methods to achieve compliance in biped robot: passive compliance (physical springs installed on joints) and active compliance (conceptually the same as stiffness control) created by the motors [159]. Dallali et al. [30] developed a humanoid robot COMAN equipped with passive compliance in the major joints of the legs which provide shock protection, locomotion robustness and energy efficiency. Choi et al. [23] discuss the design of a passive variable compliance actuator using leaf springs. Active compliance can be achieved through the joint actuation via a controller. Torque or force feedback can be used to bring compliance to the manipulator. It is mainly software based and can be applied to different applications. Lefebvre et al. [95] contribute a literature survey on active compliant motion.

Some researchers synthesize the energy-efficient walking gait by optimization. To extend the minimum-energy walking method to flat ground and uphill slopes, Channon et al. [16] proposed methods of gait generations by minimizing the cost function of the energy consumption. Westervelt et al. [161] used Bézier polynomials to approximate the actuated joint trajectories. The optimization was to minimize a energy cost function that satisfies several constraints. Chevallereau et al. [20] used four-order polynomials for the joint leg coordinates to avoid the use of the inverse dynamic model. The polynomial coefficients are the optimisation variables for torque cost.

A different approach focusing on minimizing energy consumption is the imitation of human motions. Collins and Ruina [27] assume that the gait of humans is the most energy-efficient way of walking. Srinivasan and Ruina [140] confirmed this hypothesis using a simple model in which the human is a point mass with straight legs that can change in length during a step. Their results show that the dynamically stable inverted pendulum walk without an area of support is the most energy-efficient gait. We will demonstrate in Chapter 4 that this also holds for humanoid robots, despite differences with humans.

## 1.4. PROBLEM STATEMENT AND RESEARCH QUESTIONS

In the previous sections we categorized several challenges about the motion control of the biped walking and described some solutions for resolving these issues. Furthermore, we highlighted several key areas of interest which have been established as the focus points of research. This thesis focuses on two of these key areas, namely stability of biped walking in unknown environments and energy efficiency of the biped locomotion. The following problem statement will guide the research:

**Problem statement** *How to synthesize a biped walking gait with energy efficiency and dynamical stability on a humanoid robot with high degrees of freedom on uneven terrains?*

We identify six research questions which guide the research with regards to the problem statement. The questions address the problems of (1) dynamics model, (2) energy efficiency of locomotion in sagittal and lateral direction, (3) lateral stability, (4) controller design, (5) gait optimization on level ground, (6) gait optimization on uneven terrains.

**Research Question 1:** *Can the planar inverted pendulum model be used to create a bipedal gait with a high energy efficiency?*

To solve the problem of biped locomotion from an analytic perspective, it is advantageous to obtain an elemental representation of the dynamics of the robot described by sets of simple equations. The prevailing approach employed to model the kinematics of the biped locomotion is the (linear) inverted pendulum model. However, this model is used to analyse gait stability but not energy efficiency. Our research aims at using the inverted pendulum model for both gait stability and energy efficiency.

**Research Question 2:** *What is the influence of the double support phase on the biped gait in the sagittal plane?*

The double support phase plays an important role concerning the energy efficiency and the stability of a biped gait. Our research aims at extending planar inverted pendulum model with a double support phase that will have influence on the dynamics of the sagittal locomotion. The extended model is used to analyse the effect of the force generated by the swing leg during the double support phase on the dynamically stable and energy efficient gait.

**Research Question 3:** *What are the requirements of a double support phase with respect to lateral stability of a gait?*

The introduction of double support phase not only influence the sagittal stability but also the later stability. Our research aims at developing a novel approach at addressing the lateral stability issue in the double support phase.

**Research Question 4:** *How to design the controller for gait generation?*

The control of biped locomotion is difficult because biped are high-dimensional dynamical systems. Common approaches to design the locomotion controller include: (a) the use of passive walker as a starting point for the design of active walkers; (b) the use of "zero moment point" control; (c) the use of a fixed control architecture and applying parameter search to find the parameter settings that yield successful walking gaits; and (d) the development of feedback laws based upon insights into balance and locomotion. However, these approaches imposes several issues, for example the contradictory goal including energy efficiency and robustness. Our research aims at developing a gait controller on a Nao robot for walking on level ground that implements the dynamically stable and energy efficient biped gait that was identified using the inverted pendulum model.

**Research Question 5:** *How to optimize the energy efficiency for the walk gait on flat floor?*

Since the inverted pendulum model is an abstraction of a real robot, our controller

has several parameters that needs to be fine tuned. The optimal setting of the parameters is determined by a fitness function that measures both the stability and the energy consumption of the gait. Energy efficiency and stability are sometimes contradictory requirements. Our research aims at optimizing both energy efficiently and stability of a gait on level ground, where the stability has priority over the energy efficiency.

**Research Question 6:** *How to optimize the stability of the walking gait on uneven terrains?*

Uneven terrains are tests-bed for controller robustness since they represent situations in which the robot faces changes in the environment that are unexpected and hard to model. The complexity of the uneven terrains brings great challenges w.r.t. the stability of a biped gait. In order to improve the stability of the proposed gait in outdoor environments, our research aims at optimizing the controller for uneven terrains while keeping the energy cost low and stability high.

## 1.5. THESIS OVERVIEW

This thesis is divided into seven chapters. Chapter 1 provides a brief introduction to the field. It presents the problem statement and research questions that have been posed to approach it. Chapter 2 reviews the hardware features and the electronic architecture of humanoid Nao robot. Moreover, we describe the methods used to solve the forward/inverse kinematic problem. Chapter 3 gives a brief overview of existing research about kinematics models for humanoid robots, stability criteria and various approaches to obtain energy efficient bipedal walking. Chapter 4 answers the first three research questions. We introduce a prevailing model to describe the dynamics for humanoid robots and use the Inverted Pendulum Model (IPM) to investigate the energy consumption in the sagittal plane. Subsequently, we extend the model to the frontal plane.

Chapter 5 answers the fourth and fifth research question. We describes a gait controller with multiple parameters for a 3D full-body humanoid robot. The controller can achieve an energy efficient and stable gait on the level ground after we optimize the controller parameters through Policy Gradient Reinforcement Learning (PGRL). Moreover, it presents the experiments used to evaluate the stability and the energy consumption. The improved energy efficiency is demonstrated through the comparison of our proposed gait with the original gait supplied with the Nao robot. Chapter 6 gives the answer to the sixth research question. Our approach aims to enhance the dynamical stability of biped gait on uneven terrain. Chapter 7 concludes this paper. It also gives possible directions for future work.



# 2

## PLATFORM OVERVIEW

### 2.1. INTRODUCTION OF NAO ROBOT

THE study of the special features of the robotic platform employed is paramount in the design of the gait. Some physical details of different robot platforms (for instance, the special design of joints, the number of degrees of freedom, etc.) should be considered when we implement a walking pattern generator on a real robot. In this chapter, we first review the hardware specifications of the humanoid Nao, which is the robot platform we employed in our experiments, and we analysed the performance of its actuators, sensors and central processor. We also discuss the software architecture of the Nao robot. Moreover, we introduce the basic knowledge about the kinematic chains with multiple degrees of freedom, including the calculations required to solve the forward kinematics problem, the affine transformations and the Denavit-Hartenberg (DH) parameters. Finally, we discuss the inverse kinematics problem of mapping from the three-dimensional space to the joint space.

### 2.2. HARDWARE SPECIFICATION

The robotics platform we used for this work is the humanoid robot Nao version 5, developed by Aldebaran Robotics company. The Nao robot is 57.4 cm tall and weights 5.4 kg. The Nao robot has a body-mass index (BMI) of about  $13.5 \text{ kg/m}^2$ , which means that it is very light compared to other existing robots of the similar size [54]. Table 2.1 lists the BMI values of different popular humanoid robots used for experiment with advanced robotics algorithms. Compared to the heavy robots, the light-weight benefits the performance of robots in many aspects, for instance, high load-to-weight ratio, increased flexibility and low power consumption [94, 164].

#### 2.2.1. CONFIGURATION

The employed Nao robot has total 25 degrees of freedom (DOF) (see Table 2.2). There are 11 DOF in the legs and pelvis, and 14 DOF for the upper part that includes torso, arms and head. Each arm features 2 DOF at the shoulder, 2 DOF at the elbow, 1 DOF at

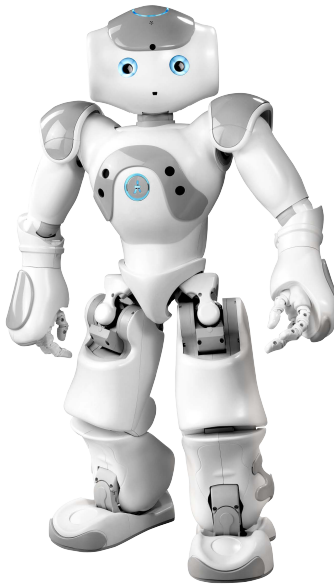


Figure 2.1: Aldebaran Nao humanoid robot

Table 2.1: BMI of different functional humanoid robots

Model	Height(m)	Weight(kg)	BMI(kg/m <sup>2</sup> )
ASIMO[61]	1.3	54.0	32.0
HRP-3[80]	1.6	68.0	26.5
KHR-3[118]	1.25	55	35.2
Nao[55]	0.57	4.5	13.5
KHR-2HV[97]	0.35	1.3	10.9

the wrist and 1 additional DOF on the hand. Each leg has 2 DOF at the ankle, 1 DOF at the knee and 2 DOF at the hip. The head can rotate around yaw and pitch axes. Figure 2.2 shows the kinematics details. In addition, a special mechanism (see Figure 2.3) composed of two coupled joints at each hip equips the pelvis. The rotation axis of these two joints are inclined at 45° towards the body. The special structure requires only one motor to drive the pelvis, which allows to reduce the building cost and save space in the narrow pelvis joint. Another special design of the Nao is its foot sole, which does not feature any passive or active joint (toe) that would enhance the speed and stability of the gait [154].

### 2.2.2. SENSORS

For a Nao to interact with its environment, the robot carries a fully capable multimedia system, which includes four microphones (for voice recognition and sound localization), two loudspeakers (for multilingual text-to-speech synthesis) and two cameras (for computer vision, including facial and shape recognition). Each camera provides a maximum resolution of 1280×960 at 30 frames per second. The robot features an inertial measure-

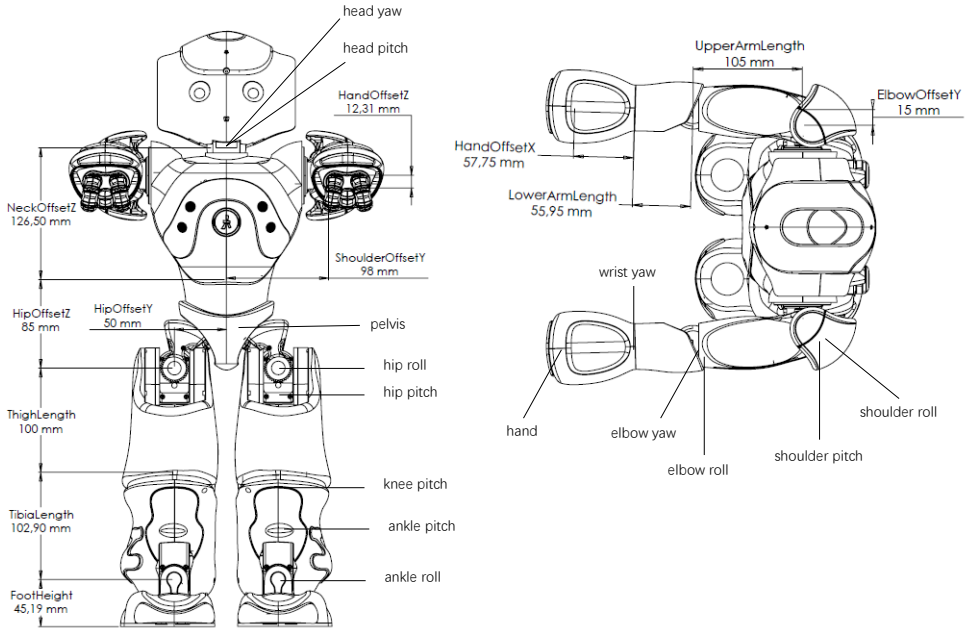


Figure 2.2: Kinematics of a Nao robot.

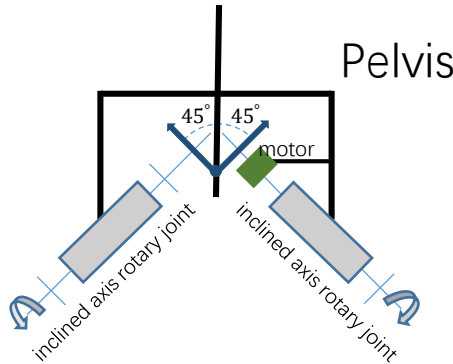


Figure 2.3: Coupled inclined axis rotary joints for the Nao pelvis

ment unit (includes accelerometer, gyrometer) and four ultrasonic sensors that provide Nao with the information of its attitude and position within space. In addition, four force sensitive resistors (FSR) with a working range from 0 N to 25 N are located on the corners of each foot. Two bumpers located at the front of each foot can provide information on collisions of the feet with obstacles. One pair of infra-red sensors (transmitter and receiver) provide the Nao robot the features of remote-control and infra-red communication between two robots (see Table 2.3).

Table 2.2: DOF names and the joints range of the Nao humanoid robot

Body part	DOF	Range(deg)
Head	head yaw	-119.5 to 119.5
	head pitch	-38.5 to 29.5
Left leg	pelvis(incline 45°)	-65.62 to 42.44
	hip roll	-21.74 to 45.29
	hip pitch	-88.00 to 27.73
	knee pitch	-5.29 to 121.04
	ankle pitch	-68.15 to 52.86
	ankle roll	-22.79 to 44.06
Left arm	shoulder pitch	-119.5 to 119.5
	shoulder roll	-18 to 76
	elbow yaw	-119.5 to 119.5
	elbow roll	-88.5 to -2
	wrist yaw	-104.5 to 104.5
Left hand	hand	open and close
Right leg	hip roll	-45.29 to 21.74
	hip pitch	-88.00 to 27.73
	knee pitch	-5.29 to 121.04
	ankle pitch	-67.97 to 53.40
	ankle roll	-44.06 to 22.80
Right arm	shoulder pitch	-119.5 to 119.5
	shoulder roll	-76 to 18
	elbow yaw	-119.5 to 119.5
	elbow roll	2 to 88.5
	wrist yaw	-104.5 to 104.5
Right hand	hand	open and close

**Inertial Measurement Unit** The acceleration and rotational speed is important information to maintain the balance of robot. To measure these two variables, Nao is equipped with an inertial measurement unit that consists of a three-axis accelerometer and a three-axis gyrometer. According to the manufacturer's specifications, the accelerometer has 1% precision in the range of  $\pm 2G$  accelerations and the gyrometer provides 5% precision for an angular speed range of  $\pm 500$  °/s. The main force measured by this accelerometer is the force of gravity, which is a constant value with high magnitude pointing downwards. The inertial forces that apply on the torso caused by the locomotion of the robot can be measured by this accelerometer too.

**Current Sensor** Every motorboard has a current sensor for each motor to measure the absolute value of the electric current. The sensor itself is a shunt resistor. The magnitude of electric current in this thesis is mainly used to calculate the power usage of joints to evaluate the energy cost of a walking gait. Besides, the electric current value of the motor is used to estimate the motor temperature, which is used to protect the motor. The magnitude of the current on the joint is directly related to the motor type and the



Table 2.3: Sensors equipped in the Nao robot

Type	number
Microphone	4
Loudspeaker	2
CMOS video camera	2
Accelerometer	3
Gyrometer	3
Ultrasonic sensor	4
Force sensitive resistor (FSR)	8
Foot bumper	2
Infra-red sensor	2

joint stiffness, which will be discussed in next section.

### 2.2.3. COMPUTER HARDWARE

The Nao is equipped with two processors. An ARM 9 processor located in the torso distributes information to all the actuator and sensor modules. One Intel ATOM Z530 1.6 GHz CPU motherboard with 1 Gb RAM is located in the head. The CPU manages the video, audio, network and other advanced modules. An additional 2 Gb flash memory and 8 Gb Micro SDHC are also available. The user can communicate with the robot through the WiFi 802.11g protocol and the Ethernet port in the back of the head.

Customized circuits based on Microchip 16 bit dsPICs micro-controllers are used for the sensor management, the actuator control and the bus control. Each motor has a micro-controller equipped with magnetic rotary encoders (MRE) that can output the 12 bits precision value (from 0 to 4095) of the angle on the joint, which is mapped to values in  $[0, 2\pi]$ . One micro-controller connected to the ARM9 processor, is dedicated to acquire signal from the IMU housed in the Nao's chest. Four force sensitive resistors (FSR) on each foot also have a micro-controller. Figure 2.4 shows the overall electric architecture of the system.

## 2.3. SOFTWARE ARCHITECTURE

A high-level description of the software architecture for the Nao robots is shown in Figure 2.5. NaoQi is the Aldebaran Robotics software framework that is a modular and distributed environment that can deal with user customized executable binaries. It provides access to the electronic devices, such as joint encoders and the IMU. It also provide interfaces to modify the joint angles and stiffness. Sensory feedback is handled similarly, allowing users to retrieve data from a variety of sources such as the on-board cameras, FSRs, IMU and the ultrasound microphones. When the NaoQi starts, it loads necessary libraries. A library contains modules which advertise their methods (depicted in Figure 2.6). The bipedal walking and other locomotions are implemented by executing one or more modules in the NaoQi.

The Nao v5 uses NaoQi version 2.1.4. The motion module provided by NaoQi ver-

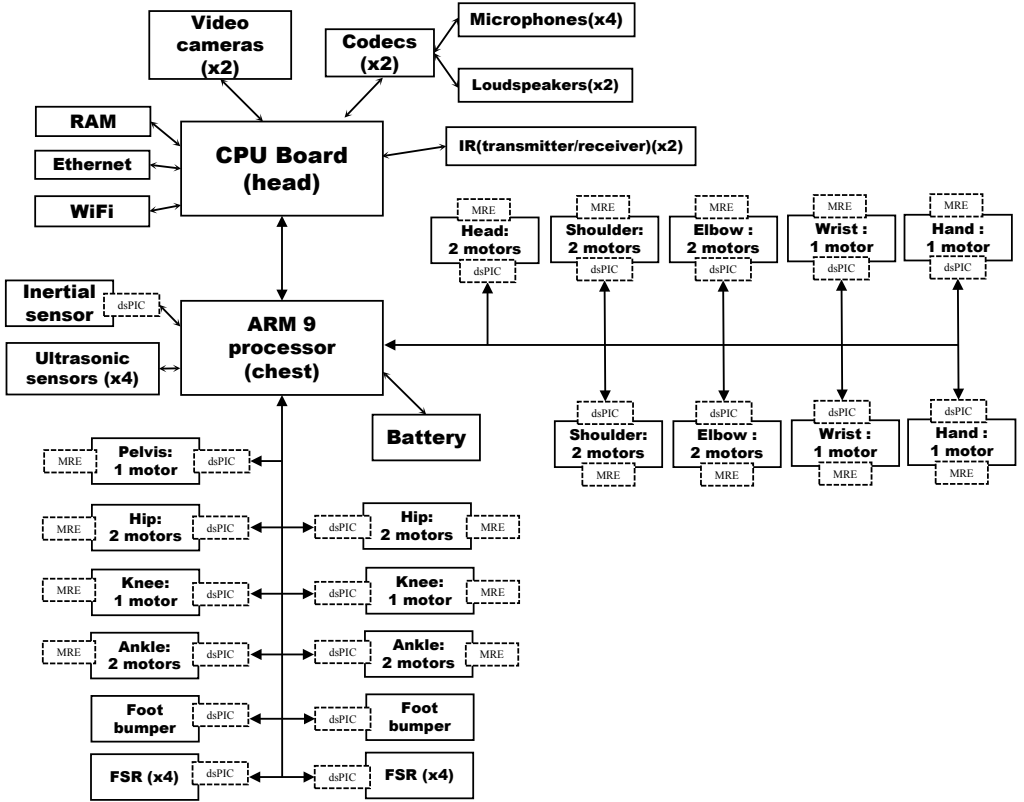


Figure 2.4: Electric architecture of the humanoid Nao robot [54]

sion 2.1.4 allows only limited control of the Nao's gait, which is further reduce in NaoQi version 2.8 provided used by the Nao v6. In the NaoQi system, when the motion module is activated, a Linear Inverted Pendulum gait with user defined speed is the only option. The module does not allow for directly controlling the angles at a high updating rate on every joint. When developing a new gait, one must be able to control every joint directly. Therefore, we created a new walking module to generate the required omni-directional motions. This module uses Lua script as a common development platform to interface with other modules. Low level interactions to the hardware level are implemented as compiled C libraries in conjunction with the Nao's on-board micro-controllers. We utilize C++ for low level interaction with hardware, Lua scripts for proposed walk module and Webots 8.2<sup>1</sup> for simulation purposes. The walk module generates trajectories for the robot's joints based upon desired translational and rotational velocity settings. Moreover, IMU feedback is used to adapt the commanded joint angles and phase of the gait

<sup>1</sup>Webots is a robot simulator developed by Cyberbotics Ltd. It can provide a complete development environment to model, program and simulate Nao robot.

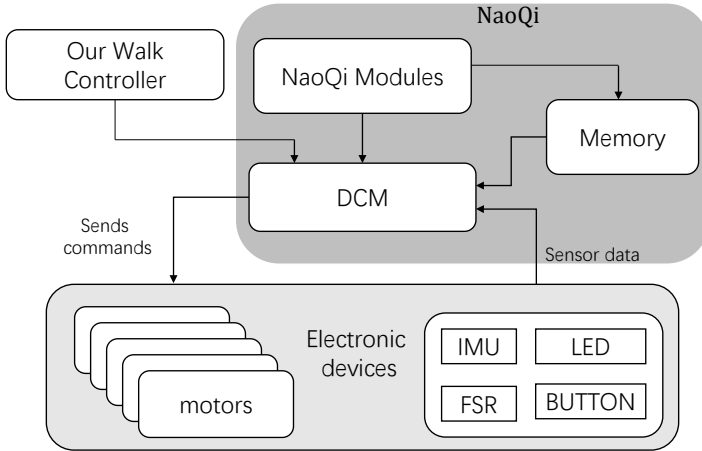


Figure 2.5: The software architecture of the humanoid Nao robot up to NaoQi 2.1.4.

cycle to improve stability during locomotion. In this way, minor disturbances such as carpet imperfections or bumping do not cause the robot to fall over.

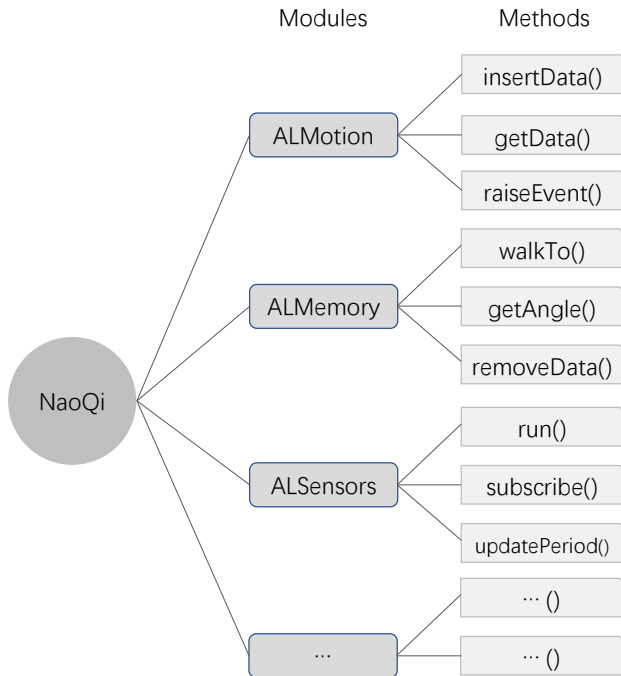


Figure 2.6: NaoQi loading modules to form a tree of methods attached to modules

### 2.3.1. THE NAOQi DISTRIBUTED ARCHITECTURE

NaoQi is the default main software that runs on the robot and controls it. The advantages of the distributed environment are various. It allows users to run behaviours locally or remotely. Robot functionalities such as motion, vision, text-to-speech, etc., can be run on a robot in the same executable or in a standalone executable that interacts with other modules on other computers. Development is made easier in a distributed environment since the same code can be compiled on different platforms and cross-compiled for embedded execution. A distributed environment also allows a user to look at variables and running methods on any real or simulated robot from the programming interfaces.

NaoQi provides many useful functionalities for development. For instance, a user can control a real robot or simulate one with many languages, such as C++, Ruby, Urbi, Python and Matlab. The user can choose whether to compile an application as a dynamic library or as an executable without changing source code. NaoQi is also a multi-platform framework that supports Linux, Windows and macOS.

The architecture is event-driven which can promote the production, detection and reaction to state changes of the robot. For example, when the robot is about to fall, the robot's state changes from "stand" or "walk" to "fall". The NaoQi system architecture may treat this state change as an event which occurrence can be made known to other applications or modules in the architecture if they subscribe to the events in the centralized memory "ALMemory". An event subscription starts the execution of the "Extractor" that generates the event. For example, subscribing to the event **FaceDetected** launches the face detection module. Subscribing to the event **WordRecognized** starts the execution of the speech recognition module.

A module can be either remote or local. If it is remote, it is compiled as an executable file, and can be run outside the robot. Remote modules communicate using the network which limits the speed of accessing local modules (direct memory access for example). Remote modules are easier to use and can be debugged easily on the external computer, but are less efficient in terms of speed and memory usage.

Another type of module is the local module, which is compiled as a library, and can only be used on the robot. Local modules launched in the same process can share variables and call each other's methods without serialization or networking. This allows the fastest possible communication between them. Therefore, they are more efficient than remote modules. Each module contains various methods. Among them, some methods can be called from the other modules. There are two different ways to access other module's methods. If two modules require to access each other's methods, in other words, they ask for a mutual communication, they can establish a connection in Broker-to-Broker way, so that modules from both brokers can talk to each other. The another way is to establish a Proxy to Broker connection which opens a single way of communication between modules. It only allows one module to access the other one and not vice versa. Figure 2.7 illustrates the process of the module communications.

Though NaoQi has many advantages for development (such as it is cross-platform, cross-language, etc.), it is designed for general purpose. Therefore, NaoQi does not provide the functionality of fine-tuning the specific actuator values (angle, angular velocity) in milliseconds which is necessary for developing a new gait. Therefore, we have to design a tailored framework to implement the interactions with the hardware. The in-

teractions are implemented using compiled C libraries in conjunction with custom controllers or NaoQi to allow for the motor position control and motor stiffness setting. Data retrieved from sensors is also possible, allowing users to get data from various built-in sensors such as the IMU, force sensitive resistors, etc.

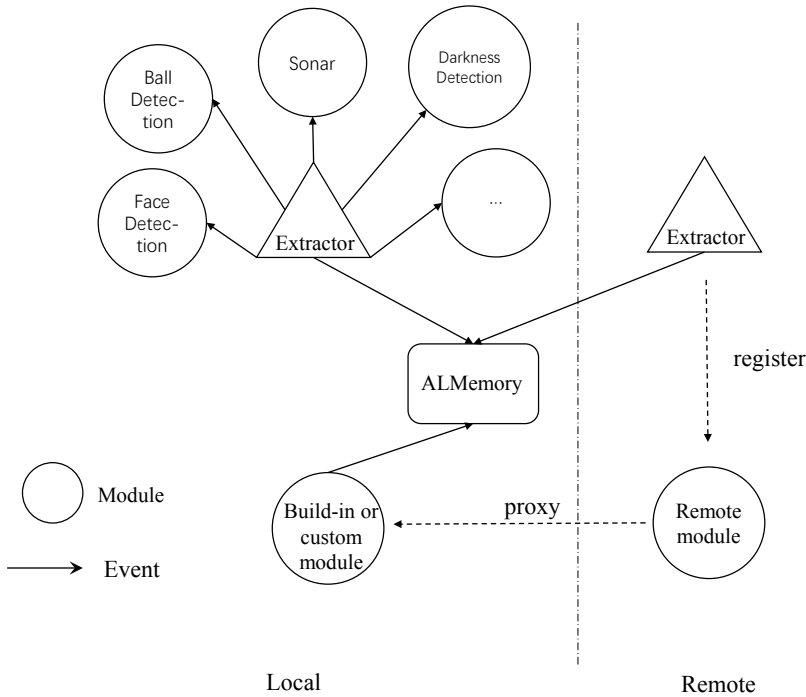


Figure 2.7: The communication process between remote and local modules

### 2.3.2. THE DEVICE CONTROL MANAGER (DCM)

The DCM is Nao's real time module that is linked as a library. It is in charge of the communication with the electronic devices of the robot except the sound sensors and the video cameras. It can be seen as the link between the high level software composed of other modules, and the low level software that is embedded inside the electronic boards. The DCM has two main purposes: sending commands to actuators and updating actuator/sensor values. The DCM gets information from the electronic devices through the motherboard and also accesses devices located in the head. In case of actuators, modules need to send an update request to the DCM using a timed command. A timed command is an order containing a float number to be sent to an actuator and the time at which the order should be executed. More than one timed command can be included in the same request. The time is an absolute value in milliseconds based on the system time. The DCM stores all timed command for each actuator, then at each 10-millisecond cycle it analyses the previous and next orders based on the current time and computes the appropriate command to send, using a linear interpolation.

### 2.3.3. ACTUATOR STIFFNESS

Actuator accuracy in the movements execution mainly relies on the following two factors: stiffness of the joints, and flexibility of the body parts. Since the body parts of Nao robot are made of a relatively rigid plastic material, we will focus on the stiffness of the joints as the dominant factor that determines the accuracy of the movements.

The stiffness of an actuator, which indicates the resistance to deviate from the reference position under the action of a force, is a crucial factor for stabilising the intended limb movement against disturbances. Stiffness of the joint is quantified as a proportion of the max power: 0.0 means 0% and 1.0 means 100% of the full power. In the motor-board, this proportion is directly applied to the maximum current. Setting the Stiffness to 0.5 means that the electric current limitation is reduced to 50%. If the Stiffness is less than 0, the motor is free (no electromagnetic break). The command to set the stiffness is sent to the motor board every DCM cycle time.

The advantage of having stiff joints is that they can track trajectories accurately. For example, the conventional ZMP tracking schemes require precise joint position tracking which is guaranteed in robots with stiff actuation and rigid links. However, stiff actuation does make the system more sensitive to external disturbances (due to collisions, impact, uneven terrain, etc.). Moreover, the stiffness of a joint is directly related to the torque that the motor delivers. Therefore, stiff joints involve a higher energy consumption and require more expensive motors. For some other biped humanoids, the sensitivity to disturbances and the high energy consumption are reduced by introducing compliant joints [1, 65]. Compliant joints reduce the stiffness to allow for smoother movements, and are good at absorbing external or internal disturbances. Hence, the compliance inherently improves the stability performance in terms of impact absorption compared to stiff leg systems. The joint stiffness tuning is a critical element in the performance of compliant systems. A low level of stiffness is preferred by some authors [90]. They choose a stiffness of 0.7 for legs (except ankles) and 0.3 for the head and the arms. The reason for the low value in the head and the arms is, first, they do not need to hold any weight during normal operation of the robot, and second, they are more elevated and thus exposed to greater damage in the event of impact against the ground.

## 2.4. KINEMATICS

To achieve a walking gait with a bipedal robot, not only the hardware and software of the robot but also the kinematics need to be investigated. The kinematics in robotics refers the analytical study of the motion of a robot manipulator. Formulating a suitable kinematics models for a robot is very crucial to analyze the behavior of the robot. There are mainly two different spaces used in kinematics modeling namely, Cartesian space and Quaternion space. The transformation between two Cartesian coordinates systems can be decomposed into rotations and translations where homogeneous transformations based on  $(n + 1) \times (n + 1)$  real matrices (orthonormal matrices) in  $n$  dimensional Cartesian space have been used most often in robotics. Although quaternions constitute an elegant representation for rotation, they have not been used as much as homogeneous transformations by the robotics community. This section also introduces the coordinate system used in depicting the position and orientation of the robot and its manipulator in relation to itself and the environment.

The robot kinematics can be divided into forward kinematics and inverse kinematics. Forward kinematics problem is straightforward and there is no complexity deriving the equations. Hence, there is always a forward kinematics solution of a manipulator. Denavit and Hartenberg [32] showed that a general transformation between two joints requires four parameters. These parameters known as the Denavit-Hartenberg (DH) parameters, have become the standard for describing robot kinematics. Inverse kinematics is a much more difficult problem than forward kinematics. The solution of the inverse kinematics problem is computationally expensive and generally takes a very long time in the real time control of manipulators. Singularities and non-linearities make the inverse kinematics problem more difficult to solve.

### 2.4.1. AFFINE TRANSFORMATIONS

An affine transformation is a mapping described by linear affine equations that transforms points, straight lines and planes. Any affine transformation preserves collinearity (i.e. after the transformation, all points on a line initially still lie on a line, sets of initially parallel lines remain parallel). The affine transformation  $f : X \rightarrow Y$  is of the form  $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ , where  $A$  is a matrix and  $\mathbf{b}$  is a vector. The types of affine transformations are following: translation, scaling, shearing, rotation, expansion, dilation, reflection, etc. All the possible combinations of the above produce an affine transformation as well. The flexibility of affine transformations with respect to describing the position and orientation of objects makes it a very useful tool.

**Augmented Matrix and Homogeneous Coordinates** A general system of  $n$  affine transformations with  $n$  variables can be written as

$$\begin{array}{ccc} x_1 & & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_1 \\ x_2 & & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_2 \\ \vdots & \mapsto & \vdots \\ x_n & & a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_n \end{array} \quad (2.1)$$

where  $x_1, x_2, \dots, x_n$  are the variables,  $a_{11}, a_{12}, \dots, a_{nn}$  are the coefficients of the system, and  $b_1, b_2, \dots, b_n$  are the constant terms.

Let  $A$  be the coefficient matrix of the system and let  $\mathbf{x}$  be the vector of variables of the system:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

In linear algebra, an augmented matrix is a matrix obtained by appending the columns of two given matrices. Given vector  $\mathbf{b}$ , where

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

the augmented matrix  $M = (A|\mathbf{b})$  is written as:

$$M = \left[ \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right]$$

In order to efficiently combine different affine transformations, we can use homogeneous coordinates. Homogeneous coordinates allows us to combine affine transformations, such as translation, rotation, scaling, etc., by multiplying the corresponding augmented matrices.

Homogeneous coordinates means extending a vector or matrix with one extra column, filling the extra space with zeros except the lower corner which has to be a "1". For instance, given the augmented matrix  $M$ , its form of homogeneous coordinates  $\hat{M}$  becomes:

$$\hat{M} = \left[ \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \\ 0 & 0 & \dots & 0 & 1 \end{array} \right]$$

In this way, the affine transformation can be simplified as matrix multiplication.

$$\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} A\mathbf{x} + \mathbf{b} \\ 0 \quad \dots \quad 1 \end{bmatrix} = \hat{M} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \hat{M}\hat{\mathbf{x}}$$

**Translation** A translation is an affine transformation that moves every point by a fixed distance in a specified direction. To translate an object by a vector  $\mathbf{b}$ , each vector  $\mathbf{p}$  in homogeneous coordinates can be multiplied by the following translation matrix:

$$T = \begin{bmatrix} 1 & 0 & 0 & b_1 \\ 0 & 1 & 0 & b_2 \\ 0 & 0 & 1 & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using this, the function form of the translated vector  $\mathbf{p}^* = (p_x^*, p_y^*, p_z^*)^{\text{tr}}$  is  $p_x^* = p_x + b_1$ ,  $p_y^* = p_y + b_2$ ,  $p_z^* = p_z + b_3$ , where  $b_1$ ,  $b_2$ ,  $b_3$  define the distance of translation along the  $x$ ,



$y, z$  axis respectively, can be expressed with matrix multiplication:

$$\hat{\mathbf{p}}^* = \begin{bmatrix} p_x + b_1 \\ p_y + b_2 \\ p_z + b_3 \\ 1 \end{bmatrix} = T\hat{\mathbf{p}} = \begin{bmatrix} 1 & 0 & 0 & b_1 \\ 0 & 1 & 0 & b_2 \\ 0 & 0 & 1 & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Because addition of vectors is commutative, multiplication of translation matrices is therefore also commutative.

**Rotation** A rotation in the  $n$  dimensional space is described by an  $((n+1) \times (n+1))$  matrix  $R$ . In the three-dimensional Cartesian space there are three distinct rotation matrices, each one of them performing a rotation of  $\theta_x, \theta_y, \theta_z$  about the  $x, y, z$  axis respectively, assuming a right-handed coordinate system:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Example** Using homogeneous coordinates, multiple affine transformations can be combined into one. For example, Suppose a rotation  $R_x$  by an angle  $\theta_x$  is performed, followed by a translation  $T$  with vector  $(b_1, b_2, b_3)^T$ , where

$$T = \begin{bmatrix} 1 & 0 & 0 & b_1 \\ 0 & 1 & 0 & b_2 \\ 0 & 0 & 1 & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The result  $Q = R_x T$  of two transformations is simply obtained by a matrix multiplication:

$$Q = \begin{bmatrix} 1 & 0 & 0 & b_1 \\ 0 & \cos\theta_x & -\sin\theta_x & b_2 \\ 0 & \sin\theta_x & \cos\theta_x & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In this thesis, we only use rotations and translations, so we will focus on these two types of elementary affine transformation in terms of 4 corresponding homogeneous coordinates. Additionally, we are invariably working in a three-dimensional Cartesian space and therefore all the definitions and examples from now on will focus on this space.

### 2.4.2. COORDINATE SYSTEMS

A coordinate system is a system which uses one or more numbers (coordinates) to determine the position of geometry items on a manifold. The prototypical example of a coordinate system is the Cartesian coordinate system. In this thesis, the coordinate system refers to a three-dimensional Cartesian coordinate system, in which three perpendicular planes are chosen and the three coordinates of a point are the signed distances to each of the planes. By default, when defining the position of a geometry item, its coordinates are interpreted in the global Cartesian system, which can be thought of as an absolute reference frame. However, in some cases, it may be more convenient to define the coordinates in a system other than the global Cartesian system, where the origin and the orientation differ from those of the global systems.

**Global coordinate system** The global coordinate system is represented by three orthogonal axes that are rigidly connected at a point called the origin of this system which is denoted by  $O$ . The global coordinate system is utilized to represent the global, absolute, or inertial frame of reference. In the present work, the global coordinates system is denoted by  $Oxyz$  which is depicted in Figure 2.8. The origin point  $O$  is set to the vertical projection point of the center of mass on the horizontal ground plane at the starting time.

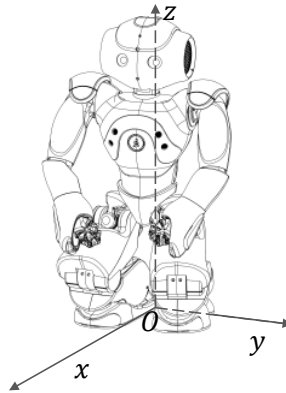


Figure 2.8: Global coordinate system reference frame

**Local coordinate systems** In addition, a local coordinate system is considered to define local properties of points that belong to a body. One specific local system of coordinates used in this thesis is attached to the CoM of the robot body. Moreover, this local coordinate system translates and rotates with the body motion and, consequently, its location and rotation vary with time. The origin of system is always located at the position of the CoM. The directions of the  $x$ ,  $y$  and  $z$  axis w.r.t. the global coordinate system are not fixed. When the Nao robot has applied a pre-defined configuration "StandZero" in which all its joint angles are zero, the direction of the  $x$  axis parallels the line that points from one of Nao's elbows to its wrist on the same arm. The direction of the  $y$  axis parallels the line that points from Nao's right shoulder to its left shoulder. The  $z$  axis is perpendicular to the transversal plane and points upwards, as shown in Figure 2.9.

To perform the kinematic analysis, we rigidly attach a local coordinate system to each link of the robot. Whatever motion the robot executes, the coordinates of each point on the link  $i$  are constant when expressed in its local coordinate system. All local coordinate systems are also described relative to the pose "StandZero". The origin point is located at the joint of the body part. Figure 2.10 shows the positions of the origin denoted by  $(o, R)$  in the local coordinate systems of four different body parts.

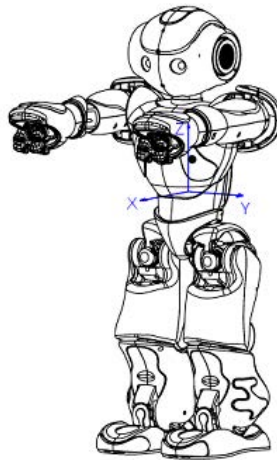


Figure 2.9: Local coordinate system reference frame when the pre-defined configuration "StandZero" is applied on a Nao [138].

**Rotation Direction** A 3D body part maybe able to rotate about three orthogonal axes, as shown in Figure 2.11. These rotations are referred to as pitch, roll and yaw:

- A pitch is a clockwise rotation about  $y$  axis, when viewed from the origin.
- A roll is a clockwise rotation about  $x$  axis, when viewed from the origin.
- A yaw is a clockwise rotation about  $z$  axis, when viewed from the origin.

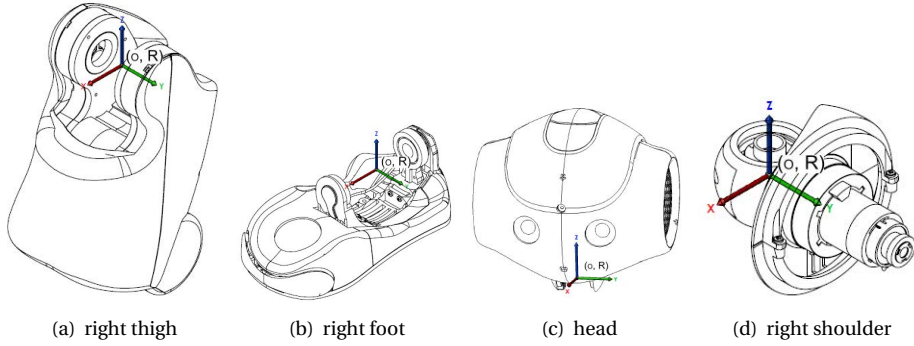


Figure 2.10: The origins (o,R) of local coordinate systems on four different body parts [139].

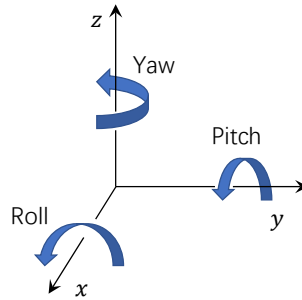


Figure 2.11: Three-dimensional rotation can be described as a sequence of yaw, pitch, and roll rotations.

The Nao robot uses an algorithm to compute its torso orientation from signals acquired with accelerometers and gyrometers. Since gravity applied on the torso is commonly the main force that can be measured by the accelerometer. It can act as an absolute reference. This implies that the signals retrieved from the accelerometer can be used to estimate the torso angle in static mode. By neglecting all the other forces acting on the torso, the roll and pitch angles of the upper body can be estimated in the following way:

$$\text{pitch} = \arcsin\left(\frac{a_x}{G}\right) \quad (2.2)$$

$$\text{roll} = -\arcsin\left(\frac{a_y}{G}\right) \quad (2.3)$$

where  $a_x$  and  $a_y$  are the acceleration in the direction of the  $x$  and  $y$  axis, respectively, and  $G$  is the gravitational constant.

### 2.4.3. FORWARD KINEMATICS

The forward kinematics problem is to determine the position and orientation of the end-effector, given the values for the joint angles of the robot. An end-effector is the device at the end of a robotic limb, designed to interact with the environment. Stated more for-

mally, the forward kinematics define a mapping from the joint space to the 3D Cartesian space.

**Denavit-Hartenberg (DH) Parameters** In order to describe the forward kinematics of a robot mechanism in a systematic manner, one should use a suitable kinematics model. Denavit-Hartenberg method, which was introduced by Jacques Denavit and Richard S. Hartenberg [32], is the most common method for describing robot kinematics. In this method, we can describe the coordinate transformation using only four parameters. These are known as Denavit-Hartenberg (DH) Parameters:  $\alpha$ ,  $r$ ,  $\theta$  and  $d$ , which are the twist angle, link length, joint angle and joint offset, respectively. A reference frame is attached to each joint  $i - 1$  to determine DH parameters w.r.t. the reference frame of its adjacent joint  $i$ :

- The  $z_i$  and  $z_{i-1}$  axis are set to the direction of the joint axis.
- The  $x_i$  axis is parallel to the common normal which means it intersects both the  $z_i$  and the  $z_{i-1}$  axis and is perpendicular to both the  $z_i$  and  $z_{i-1}$  axis.
- The direction of  $x_i$  is derived using the right-hand rule from  $z_{i-1}$  to  $z_i$ .
- the origin of joint  $i$  is at the intersection of the  $x_i$  and  $z_i$  axis.

The DH parameters can be described as followed (see Figure. 2.12):

- $\alpha_i$  (Twist angle): Angle between the orthogonal projections of joint axis,  $Z_i$  and  $Z_{i-1}$  onto a plane normal to the common normal. This is measured as the angle between the axis,  $Z_i$  and  $Z_{i-1}$ , about axis  $X_i$  to be taken positive when rotation is made counter clockwise.
- $r_i$  (Link length): Length of the common normal. This is measured as the distance between the common normals to axis  $Z_i$  and  $Z_{i-1}$  along  $X_i$ .
- $\theta_i$  (Joint angle): Angle between the orthogonal projections of the common normals,  $X_i$  and  $X_{i-1}$ , to a plane normal to the joint axis  $Z_{i-1}$ . Rotation is positive when it is made counter clockwise.
- $d_i$  (Joint offset): Length of the intersections along the joint axis  $Z_i$  to the common normal. This is measured as the distance between  $X_i$  and  $X_{i-1}$  along  $Z_{i-1}$ .

When the Nao robot applies the initial pose "StandZero" (in Figure 2.9), the DH parameters for the rotation of the reference frame on every link are shown in Table 2.4 and Figure 2.13.

**DenavitHartenberg matrix** The DenavitHartenberg convention allows the definition of the movement of links around a common joint axis to be separated into the product of a translation along a line and a rotation around the same line, so that the movement around axis  $Z_{i-1}$

$$[M_{Z_{i-1}}] = T_{Z_{i-1}}(d_i)R_{Z_{i-1}}(\theta_i) \quad (2.4)$$

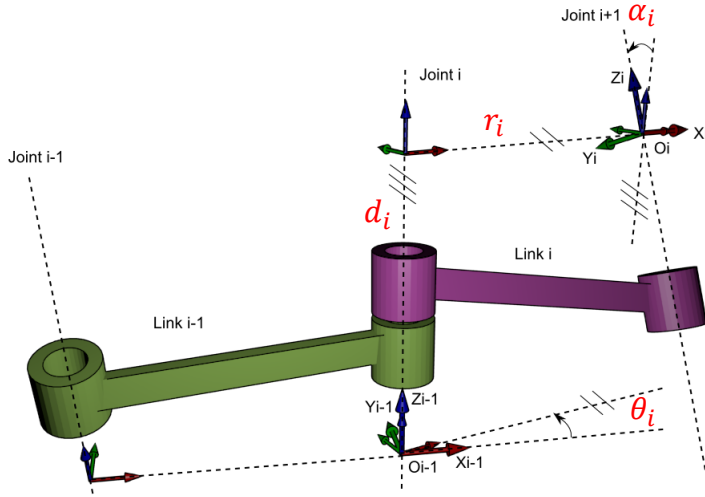


Figure 2.12: The four parameters of classic DH convention are shown in red text, which are  $\alpha_i$ ,  $r_i$ ,  $\theta_i$  and  $d_i$ . We can get the homogeneous matrices that translate the coordinates from reference frame  $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$  to  $O_iX_iY_iZ_i$ . (This figure is the work of wikipedia user "Olllydbg").

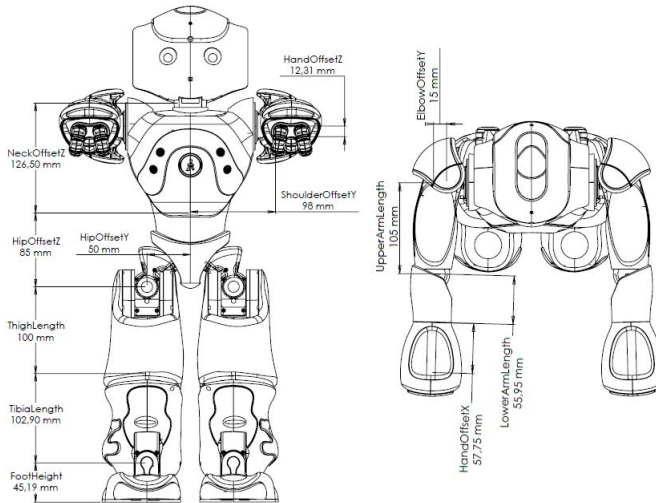


Figure 2.13: Overview of the link lengths of the Nao robot[138].

and the movement around axis  $X_{i-1}$

$$[M_{X_{i-1}}] = T_{Z_{i-1}}(r_i)R_{Z_{i-1}}(\alpha_i) \quad (2.5)$$

With these notation, each link can be described by a transformation from a coordinate system to the next coordinate system, which in case of the Figure 2.12 is the coordinates translation from reference frame  $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$  to  $O_iX_iY_iZ_i$ .

Table 2.4: Denavit-Hartenberg (DH) Parameters for Nao Robots

Joint Names	$\alpha$	r	$\theta$	d
HeadYaw	0	0	0	0
HeadPitch	$-\pi/2$	0	$-\pi/2$	0
LShoulderPitch	$-\pi/2$	0	0	0
LShoulderRoll	$\pi/2$	0	$\pi/2$	0
LElbowYaw	$\pi/2$	0	0	UpperArmLength
LElbowRoll	$-\pi/2$	0	0	0
LWristYaw	$\pi/2$	0	0	LowerArmLength
LHand	0	0	0	0
LHipYawPitch	$-3\pi/4$	0	$-\pi/2$	0
LHipRoll	$-\pi/2$	0	$\pi/4$	0
LHipPitch	$\pi/2$	0	0	0
LKneePitch	$\pi/2$	-ThighLength	0	0
LAnklePitch	0	-TibiaLength	0	0
LAnleRoll	$-\pi/2$	0	0	0
RShoulderPitch	$-\pi/2$	0	0	0
RShoulderRoll	$\pi/2$	0	$\pi/2$	0
RElbowYaw	$\pi/2$	0	0	UpperArmLength
RElbowRoll	$-\pi/2$	0	0	0
RWristYaw	$\pi/2$	0	0	LowerArmLength
RHand	0	0	0	0
RHipYawPitch	$-\pi/4$	0	$-\pi/2$	0
RHipRoll	$-\pi/2$	0	$-\pi/4$	0
RHipPitch	$\pi/2$	0	0	0
RKneePitch	$\pi/2$	-ThighLength	0	0
RAnklePitch	0	-TibiaLength	0	0
RAnleRoll	$-\pi/2$	0	0	0

$$M_{i-1}^i = T_{Z_{i-1}}(d_i)R_{Z_{i-1}}(\theta_i)T_{Z_{i-1}}(r_i)R_{Z_{i-1}}(\alpha_i) \quad (2.6)$$

The matrices of these transformation are:

$$T_{Z_{i-1}}(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{Z_{i-1}}(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{Z_{i-1}}(r_i) = \begin{bmatrix} 1 & 0 & 0 & r_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{Z_{i-1}}(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This gives:

$$M_{i-1}^i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & r_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & r_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The forward kinematics of the end-effector with respect to the base frame is determined by multiplying all of the  $M_{i-1}^i$  matrices.

#### 2.4.4. INVERSE KINEMATICS

Robot manipulators typically need to reach target points or follow trajectories in the three-dimensional space. To make the end-effector reach a point, or to follow a trajectory, we have to specify appropriate angles of the joints in the kinematic chain. Tasks to be performed by a manipulator are in the Cartesian space, whereas actuators work in joint space. Cartesian space includes orientation matrix and position vector. However, joint space is represented by joint angles. The conversion of the position and orientation of a manipulator end-effector from Cartesian space to joint space is called the Inverse Kinematics (IK) problem.

**The General Inverse Kinematics Problem** Assuming there are  $n$  joints and given the joint angle  $q_j$  of each joint, the configuration of the given manipulator is specified by  $\mathbf{q} = (q_1, \dots, q_j, \dots, q_n)^T$ . The general problem of inverse kinematics can be stated as: finding the joint angles at a given configuration so that an end-effector can reach a target position  $\mathbf{s}$ . The position and orientation  $\mathbf{x}$  of an end-effector is a function of the configuration  $\mathbf{q}$ :

$$\mathbf{x} = f(\mathbf{q})$$

The solution of the IK problem is a vector  $\mathbf{q}$ :

$$\mathbf{q} = f^{-1}(\mathbf{x})$$

where  $f$  is a non-linear function, which has an inverse function. Noted that not every function has an inverse. In order to have an inverse function, the function should must be injective. However, generally, given a position and orientation  $\mathbf{x}$ , there may exit multiple  $\mathbf{q}$  values. This fact makes the method of inverting the forward kinematics equation to solve the IK problems not feasible.



The IK problem of the robot manipulators has been studied for many decades [8, 99, 122]. Solving the inverse kinematics is computationally expensive and generally takes a very long time in the real time control of manipulators. Two main solution techniques for the inverse kinematics problem are analytical and numerical methods. In the first type, the joint variables are solved analytically according to given configuration data [34, 35, 70]. In the second type of solution, the joint variables are obtained based on the numerical methods [11, 12, 41]. We first describe the second type and then discuss a special case where the first type is possible.

**Jacobian Matrix** Suppose that all joints undertake a set of infinitely small rotation, represented  $\partial \mathbf{q}$ . Then, at the end-effector, there will be a corresponding set of displacements of the position and orientation  $\partial \mathbf{x}$ . By considering the time derivatives of  $\mathbf{x}$  and  $\mathbf{q}$ , this same relationship can be viewed as a relationship between the velocities  $\dot{\mathbf{x}}$  and  $\dot{\mathbf{q}}$ . The relationship can be described by the Jacobian matrix. Jacobian matrix is the matrix of all first-order partial derivatives of a vector-value function. Suppose  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a function which maps the space defined by vector  $\mathbf{q} \in \mathbb{R}^n$  to the space defined by the vector  $\mathbf{x} \in \mathbb{R}^m$ .

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_m(q) \end{pmatrix} \quad (2.7)$$

As described above, we can consider the corresponding displacement and orientation  $\mathbf{x} = f(\mathbf{q})$ . If write it for each component of  $\mathbf{x}$  and  $\mathbf{q}$ , we can derive the following set of equations for  $\partial x_1, \partial x_2, \dots, \partial x_m$  as functions of  $\partial q_1, \partial q_2, \dots, \partial q_n$ :

$$\begin{aligned} \partial x_1 &= \frac{\partial f_1}{\partial q_1} \partial q_1 + \dots + \frac{\partial f_1}{\partial q_n} \partial q_n \\ &\vdots \\ \partial x_m &= \frac{\partial f_m}{\partial q_1} \partial q_1 + \dots + \frac{\partial f_m}{\partial q_n} \partial q_n \end{aligned}$$

The above equations can be written as follows:

$$\partial \mathbf{x} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \partial \mathbf{q} \quad (2.8)$$

The matrix in the above relationship is called the Jacobian matrix, denoted by  $J$ :

$$J = \begin{bmatrix} \frac{\partial f}{\partial q_1} & \dots & \frac{\partial f}{\partial q_n} \end{bmatrix} = \begin{Bmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{Bmatrix} \quad (2.9)$$

or component-wise:

$$J_{ij} = \frac{\partial f_i}{\partial q_j}$$

A Jacobian in our context defines the dynamic relationship between two different representations of a system: (i) the end-effector's Cartesian position and orientation (which is denoted by the function  $f$ ) (ii) the set of joint variables (which is denoted by  $\mathbf{q}$ ). The Jacobian of the system provides the information about how changes in the joint variables  $\mathbf{q}$  cause changes in position and orientation  $\mathbf{x}$  of the end-effector.

In general, the Jacobian allows to relate corresponding small displacement in different spaces. If we divide both sides of the relationship by small time interval (i.e. differentiate w.r.t time), we obtain a relationship between the velocities of the end-effector in joint space and Cartesian space.

$$J = \frac{\partial f}{\partial \mathbf{q}} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} = \frac{\partial \mathbf{x}}{\partial t} \frac{\partial t}{\partial \mathbf{q}}$$

we get

$$\frac{\partial \mathbf{x}}{\partial t} = J \frac{\partial \mathbf{q}}{\partial t}$$

Therefore,

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}}$$

where  $\dot{\mathbf{x}}$  and  $\dot{\mathbf{q}}$  denote the time derivatives of  $\mathbf{x}$  and  $\mathbf{q}$ .

**Iterative Approximation Solutions to Inverse Kinematics** Among the IK solutions, approximation methods often be used to solve the IK problems numerically. These approximation methods often use Newton's method, which involves the Jacobian matrix (a mapping between some changes in joint space and how they create small changes in Cartesian space) of a manipulator [11, 12, 41]. The most flexible of these methods typically rely on iterative optimization to seek out an approximate solution, due to the difficulty of inverting the forward kinematics equation. We will briefly discuss iterative approximation methods to solve the IK problem.

Newton-Raphson is an iterative process. It starts from some initial guess  $\mathbf{q}$ . If this  $\mathbf{q}$  is a solution of  $f(\mathbf{q}) = \mathbf{x}$ , the process is completed. If it is not, Newton-Raphson calculates a step  $\Delta \mathbf{q}$  to a new guess  $\mathbf{q}'$ . The process then continues with  $\mathbf{q}'$  as the new initial guess. Since  $J$  is in general not square or non-invertible, in the update equation  $\Delta \mathbf{q}_k = J^+(\mathbf{x} - f(\mathbf{q}_k))$  ( $\mathbf{q}_k$  is the joint angles at current iteration) Moore-Penrose pseudoinverse  $J^+$  is used to invert the Jacobian  $J$  of the system. This equation is derived by considering a

linear approximation of  $f(\mathbf{q})$ . Such an approximation around  $\mathbf{q}_k$  is given by  $f(\mathbf{q}_k + \Delta\mathbf{q}) \approx f(\mathbf{q}_k) + J\Delta\mathbf{q}$ . Setting this approximation equal to  $\mathbf{x}$ , we find  $\mathbf{x} = f(\mathbf{q}_k) + J\Delta\mathbf{q}$ . From this equation,  $\Delta\mathbf{q}$  can be found by solving  $J\Delta\mathbf{q} = \mathbf{x} - f(\mathbf{q}_k)$ .

The cyclic coordinate descent (CCD) method [158, 160] attempts to minimize position and orientation errors by transforming one joint variable at a time. The method states that, starting from the end-effector inward towards the manipulator base, each joint must be transformed in order to move the end-effector as close as possible to the target. This procedure is repeated until a satisfactory solution is obtained. The computational cost for each joint is low and therefore a solution can be formulated very quickly.

A heuristic method called Forward And Backward Reaching Inverse Kinematics (FABRIK) [9] uses the previously calculated positions of the joints to find the updates in a forward and backward iterative mode. It minimizes the system error by adjusting each joint angle one at a time. The method starts from the last joint of the chain and works forwards, adjusting each joint along the way. Thereafter, it works backward in the same way, in order to complete a full iteration. The method instead of using angle rotation, treats finding the joint locations as a problem of finding a new point on a line, which passed through two adjacent joint positions. Thus, it converges in fewer iterations, has low computational cost and produces visually realistic poses.

**Analytical Solutions to Inverse Kinematics** Geometric Algebra can be used to analytically solve an inverse kinematic problem; see for instance [171]. In this thesis however, we do not need a general inverse kinematics method to compute the joint angles, for the ankle, the knee and the hip joints. The angles of the hips can be computed using the desired angle of the upper-body (which is usually the vertical position) and the information of the IMU. The angle of the knee of the stance leg is directly prescribed by gait controller proposed in this thesis. The robot is allowed to rotate freely around the ankle joints in the proposed gait. So, for such a policy, we need not compute the angles of the ankle joints. However, given the information of the position on upper body provided by the IMU, it is possible to analytically compute desired angles of the ankle, even if the robot walks on an uneven terrain. The joint positions of the swing leg can also be computed analytically given the *step height* (one of the control parameter described in Chapter 5). Hence, general inverse kinematic methods are not needed to implement the robot gait proposed in this thesis.

## 2.5. CONCLUSION

We have reviewed the hardware features and the electronic architecture of humanoid Nao robot in the previous sections. The embedded software architecture including NaoQi and DCM was also discussed. We also introduce the design of a software framework with the functionality of fine-tuning the actuator angle and velocity in milliseconds. Moreover, with goal of kinematics modeling, we introduced the affine transformation, coordinate systems, forward and inverse kinematics solutions. In our cases, however, the inverse kinematic problem of computing the angles on the hip and knee joints can be solved analytically.



# 3

## WALKING GAIT GENERATION OF HUMANOID ROBOTS

### 3.1. INTRODUCTION

**H**UMANOID robots are a type of robots that mimic the appearance of the human body. These robots employ hand-like actuators to grasp objects, are equipped with sensors to perceive the environment in their surroundings and they propel themselves with two legs. Those activities involve several scientific challenges. For instance, performing grasping and walking requires perception, planning and precise control of actuators. In this work, we focus on locomotion control. Since humanoid robots employ two legs to move themselves, they depend on alternately moving one leg to propel and maintain the body upright. When set up in a naive way, the bipedal walking is highly unstable and a fall of a robot might have severe consequences for the physical integrity of robot. Hence robustness of walking is of significance for operating humanoid robot.

In this chapter we review the literature concerning locomotion of humanoid robots. To begin with, we will describe the different simplified models that are usually employed to describe the dynamics of humanoid robots. This is followed by a review of the criteria that are used to ensure the locomotion balance. The next section discusses the main approaches to generate walking patterns. Looking at those methods for two-legged robots, four fundamentally different approaches can be observed: the passive dynamic walking approach, model-free approaches, model-based approaches and the biology-inspired approach. The final section gives the conclusion of this chapter.

### 3.2. HUMANOID ROBOT MODELING

Humanoid robots consist of various parts with different shapes and mass distributions. To achieve the goal of generating a stable locomotion, it is important to abstract a simple and accurate representation of the dynamics of the real robot, which can be described by simple equations. Next we will introduce several prevailing approaches employed to model the kinematics of the humanoid robot.

### 3.2.1. THE INVERTED PENDULUM MODEL

The inverted pendulum model [72, 74] represents a bipedal robot by a point mass and a single telescopic leg. The point mass  $m$  represents the whole body of the robot located at the Center of Mass (CoM). The telescopic leg represents the stance leg of the robot. The swing leg is ignored in this abstract model because the model focuses on the single support phase (SSP) of a step. The inverted pendulum model represents the foot as a point located in the origin of the coordinate system. The  $x$ -axis points to the forward moving direction of the robot, the  $y$ -axis points to the left of the robot, and the  $z$ -axis points vertically upward. The  $x$  and  $z$ -axis determine the *sagittal* plane. The  $y$  and  $z$ -axis determine the *frontal* plane.

The position vector  $\mathbf{p} = (x, y, z)^{\text{tr}}$  of the CoM can be expressed in terms of a vector of state variables  $\mathbf{q} = (\theta_r, \theta_p, r)^{\text{tr}}$  depicted in Figure 3.1. The state variables are essentially a polar coordinate system where  $r$  is the length of the telescopic leg,  $\theta_r$  is the angle between the leg and its projection on sagittal plane,  $\theta_p$  is the angle between the leg and its projection in the frontal plane. The relation between the state variables  $\mathbf{q} = (\theta_r, \theta_p, r)^{\text{tr}}$  and the position  $\mathbf{p} = (x, y, z)^{\text{tr}}$  of the CoM is given by:

$$x = rS_p, y = rS_r, z = rD, \quad (3.1)$$

where  $S_p = \sin\theta_p$ ,  $S_r = \sin\theta_r$ ,  $D = \sqrt{1 - S_r^2 - S_p^2}$ .

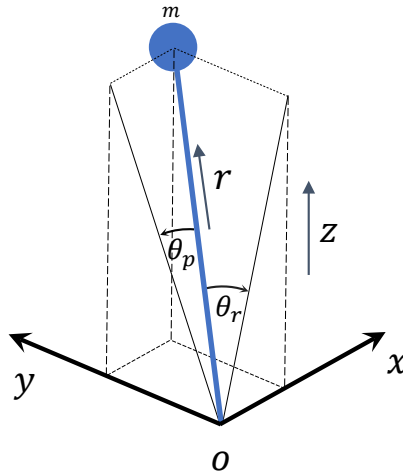


Figure 3.1: The 3D pendulum Model.

In order to determine the movement of the CoM, we need to know the all forces that are applied on the CoM. In this model, a robot can apply force on the CoM using the ankle torques and radial force of the telescopic leg. The radial force of the telescopic leg is caused by changing the length of the telescopic leg. The latter is the combined effect of the robot's ankle, knee and hip joints.

Let  $\mathbf{T} = (\tau_r, \tau_p, f)^{\text{tr}}$  be the vector describing the angular torques and the radial force on the CoM, corresponding to the state variables  $\mathbf{q} = (\theta_r, \theta_p, r)^{\text{tr}}$ . Moreover, let  $\mathbf{F}$  be the

corresponding force applied to the CoM. We can determine the power  $P$  that is put into the system using the force  $\mathbf{F}$  and the velocity  $\dot{\mathbf{p}}$  of the CoM in the Cartesian coordinates system:

$$P = \mathbf{F} \cdot \dot{\mathbf{p}} \quad (3.2)$$

We can also calculate the power  $P$  using  $\mathbf{T}$  and the state variables  $\mathbf{q} = (\theta_r, \theta_p, r)^{\text{tr}}$ :

$$P = \mathbf{T} \cdot \dot{\mathbf{q}} \quad (3.3)$$

Since the power is the same in both equations:

$$\begin{aligned} \mathbf{T} \cdot \dot{\mathbf{q}} &= \mathbf{F} \cdot \dot{\mathbf{p}} \\ \mathbf{T} \cdot \dot{\mathbf{q}} &= \mathbf{F} \cdot (J(\mathbf{q})\dot{\mathbf{q}}) \\ \mathbf{T}^{\text{tr}} \dot{\mathbf{q}} &= \mathbf{F}^{\text{tr}} J(\mathbf{q}) \dot{\mathbf{q}} \\ \mathbf{T}^{\text{tr}} &= \mathbf{F}^{\text{tr}} J(\mathbf{q}) \\ \mathbf{T} &= J(\mathbf{q})^{\text{tr}} \mathbf{F} \\ \mathbf{F} &= (J(\mathbf{q})^{\text{tr}})^{-1} \mathbf{T} \end{aligned} \quad (3.4)$$

where  $J(\mathbf{q})$  is the Jacobian of the state variable w.r.t. the Cartesian coordinates:  $\dot{\mathbf{p}} = J(\mathbf{q})\dot{\mathbf{q}}$ . Note that we can divide the third equation by  $\dot{\mathbf{q}}$  because  $\dot{\mathbf{q}}$  is independent of  $\mathbf{q}$  for all values.

We compute the Jacobian  $J(\mathbf{q})$  as follows:

$$J(\mathbf{q}) = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} = \begin{pmatrix} 0 & rC_p & S_p \\ rC_r & 0 & S_r \\ -rC_r S_r / D & -rC_p S_p / D & D \end{pmatrix} \quad (3.5)$$

where  $C_r = \cos \theta_r$ ,  $C_p = \cos \theta_p$ . We will use this result to compute the effect of the ankle torques and radial force on the dynamics of a robot.

According to Newton's second law:  $\mathbf{F} = m\mathbf{a}$ . In case of the inverted pendulum model,  $\mathbf{a} = \ddot{\mathbf{p}}$ . The force  $\mathbf{F}$  is the combined result of gravity  $(0, 0, -mg)^{\text{tr}}$ , the ankle torques  $\theta_r$  and  $\theta_p$  and the radial force  $f$  generated by the length changes of the telescopic leg. We use Equation (3.4) to transform the torques and radial force to a force vector in the Cartesian coordinate system. Therefore,  $\mathbf{F} = m\mathbf{a}$  becomes:

$$(J(\mathbf{q})^{\text{tr}})^{-1} \begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} = m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} \quad (3.6)$$

where  $m$  is the total mass of the model,  $g$  is the gravitational constant. The inverse Jacobian in Equation (3.6) can be eliminated by multiplying both sides of the equation with the matrix  $J(\mathbf{q})^{\text{tr}}$ :

$$m \begin{pmatrix} 0 & rC_r & -rC_r S_r / D \\ rC_p & 0 & -rC_p S_p / D \\ S_p & S_r & D \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} - mg \begin{pmatrix} -rC_r S_r / D \\ -rC_p S_p / D \\ D \end{pmatrix} \quad (3.7)$$

If we multiply the first row of Equation (3.7) by  $D/C_r$ , we can get

$$m(rD\ddot{y} - rS_r\ddot{z}) = \frac{D}{C_r}\tau_r + rS_rmg \quad (3.8)$$

If we translate  $rD$  and  $rS_r$  to Cartesian coordinates, we get an equation that describes the dynamics along the  $y$ -axis:

$$m(z\ddot{y} - y\ddot{z}) = \frac{D}{C_r}\tau_r + mgy \quad (3.9)$$

In the same way we get an equation that describes the dynamics along the  $x$ -axis:

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p}\tau_p + mgx \quad (3.10)$$

These equations are often used to analyze the movements of a bipedal robot.

### 3.2.2. THE LINEAR INVERTED PENDULUM MODEL

One of the most popular models for analyzing the gait of a robot is the Linear Inverted Pendulum Model (LIPM) [77]. The LIPM restricts the movement of the CoM in the inverted pendulum model to an arbitrarily defined plane, which normally is parallel to the walking surface. Restricting the movement of the CoM to a plane provides a solution of the non-linear dynamics Equation (3.7) of Inverted Pendulum Model by providing analytic solutions based on an acceptable approximation.

The first step is to limit the movement of the CoM to a plane:

$$ax + by + cz = d$$

We reformulate this equation as:

$$-\frac{a}{c}x - \frac{b}{c}y - z = -\frac{d}{c}$$

We define  $k_x = -\frac{a}{c}$ ,  $k_y = -\frac{b}{c}$  and  $z_c = \frac{d}{c}$ . This enables us to describe the plane by the normal vector  $(k_x, k_y, -1)$ . We reformulate the above equation as:

$$z = k_x x + k_y y + z_c \quad (3.11)$$

The second derivative of Equation (3.11) is also used:

$$\ddot{z} = k_x \ddot{x} + k_y \ddot{y} \quad (3.12)$$

Substituting the constraints (3.11) and (3.12) into Equation (3.9) and (3.10) yields:

$$\ddot{y} = \frac{g}{z_c}y - \frac{k_x}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c}u_r \quad (3.13)$$

$$\ddot{x} = \frac{g}{z_c}x + \frac{k_y}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c}u_p \quad (3.14)$$



where  $u_r$  and  $u_p$  are virtual inputs (that exist only in software). The relations between  $\tau_r$  and  $u_r$ , and between  $\tau_p$  and  $u_p$  are represented by following equations:

$$\tau_r = \frac{C_r}{D} u_r \quad (3.15)$$

$$\tau_p = \frac{C_p}{D} u_p \quad (3.16)$$

For a robot walking on a flat ground, the horizontal constraint plane is obtained by setting  $k_x = 0, k_y = 0$ . In this case, Equations (3.13) and (3.14) become:

$$\ddot{y} = \frac{g}{z_c} y + \frac{1}{m z_c} u_r \quad (3.17)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} u_p \quad (3.18)$$

where  $z_c$  denotes the prescribed height of the CoM.

For a robot walking on a slope,  $k_x \neq 0$  or  $k_y \neq 0$ . In this case, we can also derive Equations (3.17) and (3.18) by introducing an additional constraint. If we multiply Equation (3.13) by  $x$  and subtract the result of multiplying Equation (3.14) by  $y$ , we obtain:

$$x\ddot{y} - \ddot{x}y = -\frac{k_x x}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{x}{m z_c} u_r - \frac{k_y y}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{y}{m z_c} u_p$$

Multiplying this equation by  $z_c$  and grouping the term  $x\ddot{y} - \ddot{x}y$  results in:

$$(x\ddot{y} - \ddot{x}y)(k_x x + k_y y + z_c) = \frac{1}{m} u_r - \frac{1}{m} u_p$$

Since  $z = k_x x + k_y y + z_c$ , we get:

$$x\ddot{y} - \ddot{x}y = \frac{1}{m z} (u_r x - u_p y) \quad (3.19)$$

We can make the left-hand-side of this equation equal to 0 by introducing the following constraint:

$$u_r x - u_p y = 0 \quad (3.20)$$

Substituting  $x\ddot{y} - \ddot{x}y = 0$  into Equations (3.13) and (3.14) also results in Equations (3.17) and (3.18). Hence, constraint (3.20) enables us to use the same dynamics, described by Equations (3.17) and (3.18), for a robot walking level ground and on a slope.

Finally, if the input torques  $u_r$  and  $u_p$  are set to zero, the non-linear dynamics equations describing the motion of the CoM become independent linear equations which determine a trajectory in 3D space:

$$\ddot{y} = \frac{g}{z_c} y \quad (3.21)$$

$$\ddot{x} = \frac{g}{z_c} x \quad (3.22)$$

### 3.3. STABILITY CRITERIA

To design joint motions and controllers for a biped robot, there should be some criteria to check if the designed motions and controllers will maintain the biped stability; i.e., do they make sure that the biped robot does not fall or tip over. There are two main kinds of stability criteria that have been defined for biped robots: *static* and *dynamic* stability criteria. Both types of criteria make use of the concept of a *support polygon*.

A support polygon is the convex hull around the contact area of an object with the ground. In case of the robot, it is the convex hull around the stance foot in the single support phase, and the convex hull around both feet in the double support phase. The support polygon does not exist if both feet are off the ground (for instance, while running or jumping) or if the contact area is degenerated to a point or a line (i.e., the rigid foot rotates about an axis or point and the mechanism as a whole is pivoting).

Statically balanced robots keep the vertical projection of the Center of Mass within the support polygon in order to maintain postural stability. This is sufficient if the robot moves slowly and all the inertial forces are negligible. Contrary to statically balanced robots, the inertial effects have to be taken into account for dynamically balanced robots in the different control strategies. However, the boundary between static and dynamic balancing is very loose and dynamic gaits are referred to as not statically balanced at all times. The robot is not statically balanced when phases can be distinguished were the vertical projection of the center of mass is outside the area of support during walking. For biped robots having single-point feet, purely static balance during motion is also impossible.

A general problem of the term "dynamic balance" is that within the robotics community one does not agree about the definition and that consequently different concepts exist to judge if the robot's dynamic stability is guaranteed or not, such as zero moment point [157], foot rotation indicator [51], tumble stability criterion [170], feasible solution of wrench (FSW) [149], Adios ZMP [63] etc.

#### 3.3.1. STATIC STABILITY

Static stability is the simplest stability criterion. McGhee and Frank proposed the first static stability criterion for a walking robot at a constant speed on flat ground in 1968 [103]. They claim that a walking robot is statically stable if the vertical projection of the CoM is inside the support polygon that is formed by the contact points between the robot and ground.

Static stability defines a *static stability margin*, which is the smallest distance between the projected CoM and the edge of the support polygon. The greater the stability margin, the more stable the robot is. This criterion was extended to uneven terrain and slightly modified to reduce the calculation complexity by McGhee and Iswandhi [104].

Figure 3.2 illustrates the concept of static stability using a table. The balance of the table is maintained as long as the CoM is completely within the support polygon (blue area) that is set by the ground contact points. The dashed black arrow shows the stability margin, which is the shortest distance to the edge of the support polygon.

Projection based criteria do not take the height of the CoM into consideration. For practical purposes, a higher CoM is less stable given the same stability margin. The Energy Stability Margin, introduced by Messuri [105], addresses this problem by con-

sidering the potential energy of the robot and represents a more reliable static stability margin.

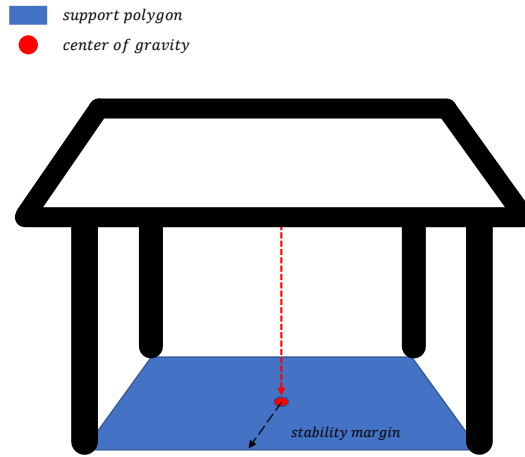


Figure 3.2: Example of support polygon and stability margin

Static stability criteria are only applicable if the CoM moves at a constant speed. Because the early generations of walking machines were huge and composed of heavy parts, they could only walk at low velocities. During the slow motion of heavy robots, some inertial effects and other dynamic components (friction, elasticity, etc.) can safely be neglected. This makes it possible to implement a statically stable walk on such robots. When it comes to fast movement, the velocity, the amount of internal angular momentum, friction, etc. are necessary conditions to maintain the balance. But static stability criterion does not take them into consideration. Therefore, even though this criterion can be applied to moving robots, for advanced robots with higher velocity, the static stability criterion becomes infeasible.

### 3.3.2. DYNAMICAL STABILITY

Static stability criteria are not suitable to assess the stability of fast moving robots, especially robots with a human-like walking gait. Human-like walking consists of a periodic orbit of a stable phase that alternates with an unstable phase from the perspective of static stability criteria. Human-like walking is a complex, though highly energy efficient way of biped locomotion, since "falling" (during the unstable phase) due to gravity, ensures the forward movement. In the unstable phase, the vertically projected CoM resides outside the support polygon. Nevertheless, the human gait is stable and allows for continuous and fast movements. This kind of motion or gait is called dynamically stable.

Dynamic stability requires different criteria to measure the stability margin of a walking robot. One of the most popular criteria in use today is the Zero Moment Point criterion [75, 76, 142, 157]. This criterion uses the notion of the Zero Moment Point (ZMP), which has been known for about fifty years, but is in no way outdated.

The ZMP specifies the point within the support polygon where the line through CoM

of the resulting force vector  $\mathbf{F}_{com}$  of all the forces working on the robot, crosses the ground surface. At this point, the ground reaction force  $\mathbf{F}_{gr}$  compensates  $\mathbf{F}_{com}$ , i.e.  $\mathbf{F}_{com} + \mathbf{F}_{gr} = 0$ . The ground reaction force  $\mathbf{F}_{gr}$  on the horizontal level ground consists of a vertical component that avoids sinking into the ground and a horizontal component that is the result of ground friction of the foot. The point where the ground reaction force  $\mathbf{F}_{gr}$  is working is called the Center of Pressure (CoP). Because the ground reaction force  $\mathbf{F}_{gr}$  cancels  $\mathbf{F}_{com}$  at the CoP, if it is inside the support polygon, there is no moment that can rotate the robot in the sagittal and the frontal plane, which could let the robot fall. Hence, the name Zero Moment Point. If, however, the line through CoM of  $\mathbf{F}_{com}$  crosses the ground surface outside the support polygon, the CoP moves to the edge of the support polygon and there is an uncompensated moment, which makes the robot rotate around the CoP. Figure 3.3 gives an illustration of the two cases. Note that the figure assumes that all the joints are stiff (fully actuated). Without fully actuated joints, the CoP changes and the robot may rotate around unactuated joints.

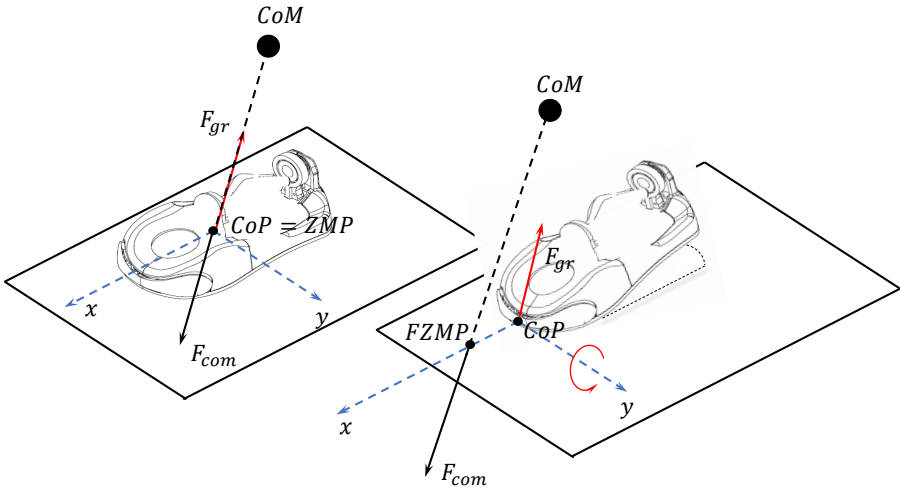


Figure 3.3: On the left, in the situation when the foot are firmly on the ground, the ground reaction force cancel out the gravitational force at the CoP or ZMP, which lies inside the support polygon. In this case, the robot is in the dynamic stability. On the right, when the heel lifts at some height, the line through CoM of  $\mathbf{F}_{com}$  crosses the ground surface outside the support polygon, the CoP moves to the edge of the support polygon and there is an uncompensated moment, which makes the robot rotate around the CoP.

The dynamics of the robot may also cause a moment that could rotate the robot in the horizontal plane, for instance by moving the swing leg. There is no point on the ground where this moment is 0 unless ensured by the dynamics of the robot. To avoid a rotation in the horizontal plane, the foot or feet must have sufficient rotational friction with the ground.

The ZMP is used as a *dynamic stability criterion*. This ZMP criterion guarantees that we can control a robot as if the robot is solidly connected to the ground. Here, we assume that the foot in the single support phase or the feet in the double support phase have sufficient grip to avoid slipping; i.e., the friction can compensate the horizontal component

of  $\mathbf{F}_{com}$ , and can compensate the moment in the horizontal plane. If the projection of  $\mathbf{F}_{com}$  on the ground surface lies outside the support polygon, the ZMP is no longer defined. In this case the projection point is denoted as the Fictitious ZMP (FZMP) [168] or the Foot Rotation Indicator<sup>1</sup> (FRI) [51]. Since the FZMP lies outside the support polygon, in a fully actuated robot, there is a moment w.r.t. the edge of the support polygon, and the robot starts rotating around the CoP, which then lies on the edge, and falls.

The (F)ZMP can be computed by considering the gravitational, inertial, coriolis and centrifugal forces on all parts of the robot. Real-time accurate computation of the (F)ZMP is not easy, and simplified interpretations are often used. One example of this is the definition of Huang et al. [29], as shown in following equations:

$$y_{zmp} = \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g) y_i - \sum_{i=1}^n m_i \dot{y}_i z_i - \sum_{i=1}^n I_{i,x} \ddot{\theta}_{i,x}}{\sum_{i=1}^n m_i (\ddot{z}_i + g)} \quad (3.23)$$

$$x_{zmp} = \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g) x_i - \sum_{i=1}^n m_i \dot{x}_i z_i - \sum_{i=1}^n I_{i,y} \ddot{\theta}_{i,y}}{\sum_{i=1}^n m_i (\ddot{z}_i + g)} \quad (3.24)$$

where

- $(x_{zmp}, y_{zmp}, 0)$  are the ZMP coordinates in the Cartesian coordinate system,
- $(x_i, y_i, z_i)$  is the center of mass of link  $i$ ,
- $m_i$  is the mass of link  $i$ ,
- $I_{i,x}$  and  $I_{i,y}$  are the  $x$  and  $y$  components of the inertia moment of link  $i$ ,
- $\theta_{i,x}$  and  $\theta_{i,y}$  are the angular velocities around the axes  $x$  and  $y$  of link  $i$ ,
- and  $g$  is the gravitational acceleration.

Note that, in this set-up, the origin  $(0,0,0)$  of the coordinate system lies on the ground surface.

Although the real time computation of the ZMP is difficult for a robot with many links, it is easy for the linear inverted pendulum model (LIPM). Of course the LIPM is a simplification that ignores aspects such as the angular moments of the robot's links, and therefore, the LIPM can only be used to approximate the location of the ZMP. An often-used approach for calculating the ZMP is the cart-table model [72]; see Figure 3.4. We assume a weightless table on which a cart with mass  $m$  is driving. The cart may accelerate using *an engine that drives the wheels*. So, if the cart accelerates in one direction  $\mathbf{a}$ , the wheels execute a force  $\mathbf{F}$  on the table in the opposite direction; i.e.,  $\mathbf{F} = -m\mathbf{a}$ . The point where the line through the CoM of the force  $\mathbf{F} + (0,0,mg)^{\text{tr}}$  crosses the ground surface is the ZMP.

$$\frac{y_{zmp} - y}{z_c} = \frac{F_y}{mg}$$

$$\frac{x_{zmp} - x}{z_c} = \frac{F_x}{mg}$$

<sup>1</sup>The Foot Rotation Indicator is defined both inside and outside the support polygon. The FRI corresponds with the ZMP inside the support polygon.

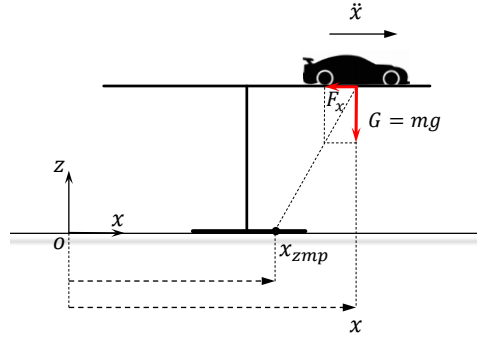


Figure 3.4: The cart-table model

We can rewrite these equations as:

$$y_{zmp} = y - \frac{z_c \ddot{y}}{g} \quad (3.25)$$

$$x_{zmp} = x - \frac{z_c \ddot{x}}{g} \quad (3.26)$$

If we substitute Equations (3.21) and (3.22) in the above equations, we find the ZMP for the LIPM without ankle torques:  $\mathbf{p}_{zmp} = (0, 0, 0)^T$ . Note that we put the origin of the coordinate system in LIPM at the contact point of the telescopic leg; i.e., the location of the ankle joint of the stance leg. The location of the ZMP is not surprising since the CoM moves in a horizontal plane and therefore does not fall. This is the result of the increasing length of the telescopic leg. Without torques on the ankle joint the tangential force cause an unrestricted rotation around the ankle joint.

The ZMP does not stay at the ankle joint if torques are applied. By substituting Equations (3.17) and (3.18) in Equations (3.26) and (3.25), we get:

$$y_{zmp} = \frac{1}{mg} u_r \quad (3.27)$$

$$x_{zmp} = \frac{1}{mg} u_p \quad (3.28)$$

Kajita et al. [72] use this ZMP approximation of the LIPM to design a gait controller for a bipedal robot.

### 3.4. WALKING GAIT GENERATOR

Walking gait generation is employed to calculate the trajectory of the joint angles that generate the motions. Beside realizing a gait, the objectives of walking gait generation include robustness, energy efficiency, movement velocity, versatility, etc. of the gait. In this section we review several approaches of walking gait generation that can be used to generate joint trajectories guaranteeing the dynamical postural stability of the humanoid robot. Especially, we review model-based approaches using the ZMP criterion.

### 3.4.1. PASSIVE DYNAMIC WALKING

There exists a type of biped walker with a simple architecture [28, 100] that is able to walk down a slope; see Figure 3.5. The walker alternates its support from one leg to the other and does not require any energy source to propel other than the power that comes from the potential energy gained by moving down a slope. It is able to trade the potential energy at the upper part of the slope for energy losses during walking. The most fundamental cause of the energy losses are impact and friction, primarily between the feet and the ground. Because it needs no energy for stabilization or control, the walker operating passive dynamically can be energy efficient.



Figure 3.5: A passive dynamic walker by Collins [28].

To understand how a stable gait is achieved, simplified models of such systems have been studied in detail. The simplest mathematical model of passive dynamic walking is the rimless wheel which has rigid, massless spokes and a single point mass at the center of the wheel. The non-linear dynamics and stability of the rimless wheel were studied extensively by Coleman and Ruina [25]. If we take only two legs from the rimless wheel, place a pin joint between them creating a rudimentary hip, and allow the legs to swing freely, then we can obtain the compass gait [42, 52, 53]. For this case, the legs are modeled by links with point masses in the center of mass of a link. These point masses are responsible for swinging a leg forward during a step.

The modern implementation of passive dynamic walking was invented by McGeer who published a paper [100] introducing this concept and implemented it both in silico (by simulation) and as walking machines built of bars and hinges. McGeer's walkers have human-like gaits and are energy efficient, and are inherently stable with respect to small disturbances. The legs of the dynamic walker described by McGeer [100] do not have knee joints. To further investigate the dynamics of the human-like walker, McGeer published a second paper which described a passive dynamic walker with knees [101, 102]. However, McGeer's machines (Figure 3.6) are human-looking only when viewed from the side. From the front, they look more like a person walking on crutches because they were built with four legs to keep the motion stable in the frontal plane. Collins

[28] built the first three-dimensional, two-legged passive dynamic walking machine with knees.

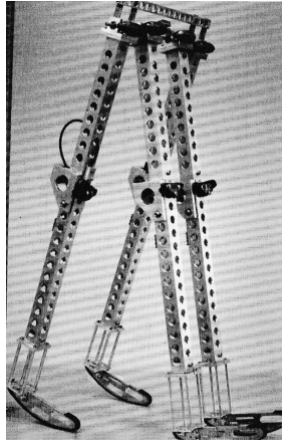


Figure 3.6: McGeer's experimental passive biped

The passive dynamic walker is a simple and energy efficient mechanism that enables bipedal walking. One of its limitations is the difficulty to execute complex motions. Since the hardware design of passive dynamic walking is oriented to optimize the energy consumption, it is not usually possible to perform different tasks.

### 3.4.2. MODEL FREE WALKING GAIT GENERATION

A conventional procedure to design a walking gait generator is to apply a physical abstract model and employ a suitable dynamic stability criterion to generate the trajectories of joint angles that guarantee a stable movement. However, it is also possible to create the walking gait generator without any physical model. This approach is called "model-free". In this section, we will discuss several methods to generate a model-free walking gait that aims to mimic the natural human step.

#### DYNAMICS CONTROL

The idea of dynamics control is to put the intelligence not in the control of the robot but in the mechanical design. These robots are carefully designed mechanical systems with tuned dynamics so they can achieve a dynamic gait despite the lack of any control. They can be viewed as passive dynamic walkers, extended with actuators that enable them to walk on a flat ground, such as the works of Collins et al. [28], Van der Linde [155] and Takuma [151].

These works give very nice results regarding energy consumption and the walking motion looks very natural. However, they lack the versatility of trajectory controlled robots. A major disadvantage is that there is not a strategy to build and control such a robot, but this mainly depends on the experience and skills of the researcher. Many parameters have to be tuned by hand, so from a practical point of view the size and weight of the robot must be adapted to this.



### CENTRAL PATTERN GENERATOR

The motion of humans and other animals are controlled by rhythmic patterned signals with sensory feedback. These signals are generated by neural network called Central Pattern Generator (CPG) [36, 67]. Since humanoid robot resembles the structure of the human, it is natural to control the gait of a humanoid robot in a similar way by the rhythmic signals synthesis.

CPG methods used to control the biped locomotion on humanoid robots are often inspired by Taga's work on neuro-mechanical simulations. Taga successfully applied a CPG controller for an 8-link simulated planar biped model [148]. A pair of CPGs, modeled by an ANN (artificial neural network), controlled the muscles of the trunk and the left and right hip, knee and ankle joints. Once the model had been trained, it not only produced a gait under normal conditions, but it also adapted to environmental perturbations such as uneven terrain or increased carrying load. Taga also demonstrated that the speed of walking could be controlled by a single parameter which drove the neural oscillators, and the step cycle could be maintained by a rhythmic input to the oscillators [146, 147]. Miyakoshi et al. extended Taga's work from 2D to 3D and also simplified the CPG control mechanism [110].

Morimoto et al. [111] show that a humanoid robot can step and walk using simple sinusoidal joint trajectories with their phase adjusted by a coupled oscillator model. The control approach was successfully applied in a hydraulic humanoid robot developed by SARCOS and the small humanoid QRIO. Other examples of CPG controlled biped locomotion include [5, 39, 58, 129, 135].

A Central Pattern Generator offers several advantages for the control of locomotion in robots [67].

1. The purpose of CPG is to produce stable rhythmic patterns which provides robustness against perturbations. That is, the system can rapidly return to the normal rhythmic behaviour after perturbations happen.
2. CPG-based approaches are ideally suited to integrate sensory feedback signals [84].
3. CPGs usually offer a good foundation for learning and optimization algorithms.

CPG-based approaches also have inherent drawbacks [67]. One of these is the huge exploration cost to find a proper combination of shapes and rhythms for a particular locomotion problem. A second drawback is that a solid theoretical foundation for describing CPGs is yet missing. For instance, it is very difficult to prove the stability of the complete CPG-robot system.

### FUZZY CONTROL AND LEARNING

This group of methods consist of neural networks, fuzzy logic techniques, genetic algorithms, etc. or combinations of these methods. The characteristics of the methods are the absence of a model and the high tolerance against imprecision and uncertainty.

**Fuzzy Control** Some investigators in the field of biped locomotion use fuzzy control for biped gait synthesis and for balance control.

Murakami et al. [112] verified the validity of using fuzzy control for controlling the gait of a biped dynamic walking robot.

Zhou et al. [173] presents a general fuzzy reinforcement learning (FRL) method for biped dynamic balance control. A neuro-fuzzy network architecture makes it possible to incorporate intuitive knowledge about human balancing and knowledge about how a gait is evaluated into the FRL agent in order to accelerate the learning process. The knowledge is used to initialise the action network, the critic network and (or) the evaluation feedback module.

Jha et al. [71] made a gait generator using a fuzzy logic controller whose rule base was optimized off-line, using a genetic algorithm.

Zhou et al. [172] proposed an adaptive-network-based fuzzy inference system (ANFIS) control strategy based on a hierarchy consisting of (1) a walking planning level, (2) a gait generating level, and (3) a joint control level. This hierarchical architecture does not require detailed kinematics knowledge or dynamic biped models. The effectiveness of the proposed ANFIS joint control was verified through a 5-link biped robot simulation.

Choi et al. [24] applied fuzzy posture control against disturbance and to keep balance on a slope. A fuzzy algorithm that makes use of the obtained ZMP trajectory information controls the posture of biped walking robot.

**Genetic Algorithms** Many researchers have proposed methods that involve the genetic algorithm as an optimization tool to generate a gait for the biped robot.

Cheng et al. [17] proposed to formulate the design of the biped controller of a gait as a parameter search problem, and to apply a genetic algorithm to obtain the optimal design. They use a 5-links bipedal model for their approach.

Arakawa [7] applied a genetic algorithm to generate off-line the joint trajectories of biped locomotion minimizing the energy consumption. He aims to generate more natural motions by considering the dynamic effect.

Capi et al. [13] used a real number genetic algorithm as an optimization tool for trajectory generation with minimum consumed energy. The minimum torque change cost function and constant vertical position trajectories are used in order to compare the results and verify the effectiveness of this method.

Kho et al. [82] used genetic algorithms to minimize the consumed energy by searching for the optimal locations of the center mass of every link of a biped robot. The learning control scheme consists of a feed-forward learning rule and linear feedback control inputs for the stabilization of the learning system.

Park et al. [120] employed a genetic algorithm to search for an optimal locomotion pattern, and at the same time, the optimal locations of the mass center of every link of the biped robot.

**Neural Networks** Researchers in the field of biped locomotion have studied the use of neural network learning for on-line gait modification, which is proved as a feasible method to generate new biped gaits.

Miller [106] developed biped control strategies that do not require detailed kinematic or dynamic models, based on a fixed gait generator, PID controllers, and neural network learning. Kun et al. [92] proposed to improve the biped performance by using CMAC

neural networks that are responsible for the adaptive control of side-to-side and front-to-back balance, as well as for maintaining good foot contact.

Sabourin et al. [131] used neural networks twice to learn the joint trajectories of a gait. The goal of the use of these neural networks is to increase the robustness of gait in the case of external disturbances.

The work of Geng et al. [48] demonstrated that stable biped walking can be achieved by combining the physical properties of the walking robot with a small, reflex-based neuronal network governed mainly by local sensor signals.

Nakanishi et al. proposed a learning method that can learn biped locomotion from human demonstrations [116]. Other research in this field was performed by Aoi and Tsuchiya [6], Komatsu and Usui [86] and Matsubara et al. [37].

### 3.4.3. MODEL-BASED WALKING GAIT GENERATION

Model-based walking gait generation methods generate trajectories for the different joints based on a model of the robot. A joint trajectory-tracking controller has to follow the desired trajectory. Many of such controllers use the Linear Inverted Pendulum Model described in Subsection 3.2.2, and the Zero Moment Point criterion described in Subsection 3.3.2, to generate the joint trajectories.

Sugihara [143] divides model-based trajectory generation methods in "trajectory replaying" and "real-time generation", or roughly speaking in off-line and on-line techniques.

#### TRAJECTORY REPLAYING

The first category prepares a joint-motion trajectory in advance, and applies it to the real robot with little on-line modification [60, 114, 167]. These methods divide the problem into two sub-problems, namely, planning and control. Numerous off-line techniques mainly focus on the aspect of optimization of a certain criteria, such as motion stability.

Kim et al. [83] first designed a gait for a biped robot that enables walking on flat floors. To enhance the motion stability on an uneven and inclined floor, they developed six decoupled on-line controllers and added them to the previous gait pattern generator. The resulting gait pattern generator enables a biped humanoid robot to adapt to the floor conditions.

There is also research in this category mainly focusing on the optimization of energy consumption. Chevallereau and Aoustin focused on optimal cyclic gaits for a walking and running biped robot without actuated ankles [20]. They use polynomial functions for the joint leg coordinates to avoid the use of the inverse dynamic model. The polynomial coefficients are the optimisation variables for various criteria (maximal forward velocity, minimal torque, and minimal energy) to ensure cyclic motion of the biped.

Denk et al. [33] developed a systematic method for generating databases of walking primitives for humanoid robots allowing step length adaptation, direction changes and stepping over obstacles. They demonstrated that these walking primitives can be computed efficiently by optimal control techniques using direct collocation methods.

#### REAL-TIME GENERATION

The walking patterns generated by trajectory replaying methods substantially limit the mobility of humanoid robots despite several decades of study. They can neither cope

with sudden contacts with the unknown environment, nor respond to unexpected commands such as an emergency stop. In order to enhance the high-mobility of robots, the humanoid motion should be generated in real-time in accordance with the dynamics. The methods in this category generate joint trajectories in real-time, while using actual robot feedback information in accordance with a pre-determined goal of the motion [44, 126, 150] (Figure 3.7). This kind of method executes planning and control in a unified way. Although real-time generation is more promising than trajectory replaying from the viewpoint of high mobility, these methods commonly suffer from a large amount of computations which have to be performed in real time.

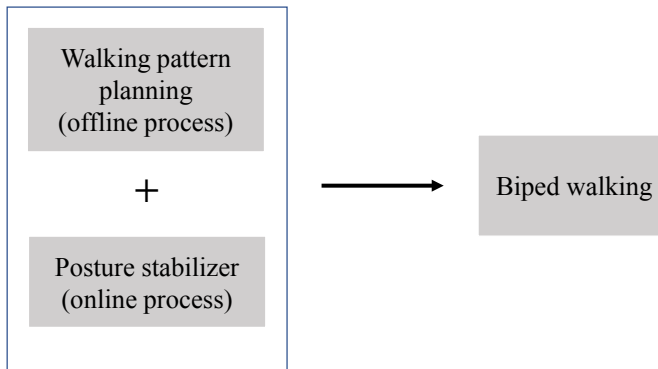


Figure 3.7: The realization of biped walking with realtime generation.

Sugihara et al. [143] have developed a real-time motion generation method that controls the center of mass (CoM) by indirect manipulation of the zero moment point (ZMP). The real-time response of the method provides humanoid robots with high-mobility. The method consists of four subsystems, namely, the referential ZMP planning, the ZMP manipulation, the COM velocity decomposition to joint angles, and local control of joint angles.

Kajita et al. [73] introduced the Three-Dimensional Linear Inverted Pendulum Mode (3D-LIPM) to represent the dynamics of a biped robot. Under this model, the geometric nature of trajectories are discussed. To stabilize the walking motion, a sensor feedback module is used to reduce the error between the desired ZMP trajectory and the actual ZMP.

Another approach introduced by Kajita et al. [72] generates a walking gait pattern by using a preview control of the zero-moment point. In order to track the ZMP well, a preview of future ZMP positions is used. The preview controller reduces the tracking error by considering the future ZMP positions. The output of the preview controller are the required accelerations  $a_x$  and  $a_y$  of the CoM. The actual position  $(x, y)$  of the CoM is determined by solving a discretized version of a system of differential equations using

the outputs  $a_x$  and  $a_y$  of the preview controller as input.

$$\frac{d}{dt} \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} a_y \quad (3.29)$$

$$\frac{d}{dt} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} a_x \quad (3.30)$$

The current ZMP position is subsequently estimated using Equations (3.25) and (3.26).

$$y_{\text{zmp}} = \begin{pmatrix} 1 \\ 0 \\ -z_c/g \end{pmatrix}^{tr} \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \end{pmatrix} \quad (3.31)$$

$$x_{\text{zmp}} = \begin{pmatrix} 1 \\ 0 \\ -z_c/g \end{pmatrix}^{tr} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} \quad (3.32)$$

This approach and several refinements have been applied to generate a walking gait pattern in several robots [40, 43, 59].

Nagasaka et al. [115] proposed an integrated motion control method to make high mobility movement on a bipedal humanoid. This method generates dynamically consistent motion patterns in real-time based on the concept of the dynamics filter, which assures that the force and the moment generated by the robot can equilibrate with that caused by the environment.

#### 3.4.4. HUMAN-INSPIRED APPROACHES

Instead of mimicking the neuro-biological approaches used by humans to generate the walking patterns, it is also possible to mimic the walking patterns themselves. A problem with this technique is that the anthropomorphic robot mechanisms are not equipped with as many degrees of freedom as the real human, and the mass distribution are different. Therefore, if raw data of motions captured from humans are directly replayed in humanoid robots, the resulting gaits are poorly balanced. For this reason, human-inspired methods are usually restricted to the creation of animations. However, it has also been applied to real humanoid robots [2, 4, 165].

Mita et al. [107] recorded human data and applied a tracking control to the human gait trajectories. Unfortunately, measuring the angle trajectories during human walking for a wide range of step lengths and walking speeds is difficult and time consuming [14]. Moreover a humanoid robot does not necessarily have the same kinematic and dynamical properties (e.g. link dimensions, number of DOF, number of actuators, etc.) as a human individual. Therefore, the recorded data from humans have to be processed to fit the robot's specifications.

Ames [3, 166] utilized human data and output functions (or virtual constraints) that appear to be intrinsic to human walking in order to formally design controllers that probably result in stable robotic walking.

In the work of Sinnet et al.[137], a human-inspired method for achieving bipedal robotic walking is proposed in which a hybrid model of a human is used in conjunction with experimental walking data to obtain a multi-domain hybrid system. Note that interesting biologically inspired locomotion controllers for biped locomotion can also be constructed based on CPGs ([47, 49]).

### 3.5. CONCLUSION

In this chapter, we introduced the literature concerning biped walking gait generation on humanoid robots. First we enumerated the different simplified models that are usually employed to describe the dynamics of humanoid robots. One important model is the Inverted Pendulum Model, which represents the body as point mass that is supported by the legs and rotates around the ankle joint. The conceptual and mathematical simplicity of the inverted pendulum model made it be employed as a base for many other models. An often-used case is the 3D Linear Inverted Pendulum Model, which represents the non-linear dynamics of the CoM by means of second order linear differential equations. However, the simulation study showed that the walking gait generated by this model has inherent disadvantage in terms of energy efficiency, since the legs have to keep bent in order to constrain the CoM at a certain height during the walking. Therefore, it might be interesting to investigate the performance of gait based on this model with fully-stretched legs.

After defining the prevailing dynamics model of the humanoid robot, stability criteria were discussed. The static stability criterion requires the vertical projection of the CoM of the robot to lie within the area of support. This criterion can only be applied on static or low-mobility systems. Since this criterion neglects inertial forces, it is not suited for high-mobility systems. The inclusion of inertial forces leads to dynamic stability criteria, which are mostly based on the concept of the Zero Moment Point. The ZMP specifies the point on the ground where moment acting on the robot equals zero. The dynamic criteria specify that, in order to be able to control the movements of the robot, the ZMP must lie within the support polygon.

The last section discussed the main types of walking gait generation. Looking at those methods for two-legged robots, four fundamentally different approaches could be observed: the passive dynamic walking approach, model-free approaches, model-based approaches and the biology-inspired approach. Passive dynamic walking approaches enable the robot to walk down a slope without any energy source except the one from its potential energy. Model free approaches generate the walking gait generator without any physical model. Model-based approaches employ analytic equations to generate walking patterns. Biologically inspired approaches try to transfer results from human motion capture and analysis, bio-mechanics, or neuro-scientific research to the control system. In this work, we applied one of the model-based approaches to analyse the dynamics of the walking gait.

# 4

## AN INVERTED PENDULUM MODEL FOR GAIT ANALYSIS

### 4.1. INTRODUCTION

ONGOING research on bipedal walking in the past decades resulted in legged robots with impressive versatility. Bipedes such as ASIMO or HRP-2 can walk, climb and even run. Apart from versatility, desirable properties of a humanoid robot are low energy consumption and gait adaptation to uneven terrains.

Collins and Ruina [27] assumed that the gait of humans is the most energy-efficient way of walking. Srinivasan and Ruina [140] confirmed this hypothesis using a simple model in which the human is a point mass with straight legs that can change in length during a step. The energy consumption of the gait is defined as the positive work of increasing the length of the stance leg. Srinivasan and Ruina use this model to identify the most energy-efficient gait. Their results show that the stance leg should have a fixed leg-length almost till the end of a step, and should increase in length at the end of a step. This corresponds to a human who's stance leg has a stretched knee joint and the stance foot flat on the ground during the single support phase, and a toe-off during the

---

This chapter is an updated and expanded version of the following publications:

1. **Z. Sun**, and N. Roos: An energy efficient gait for a Nao robot, BNAIC 2013: *Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands*, pp. 323-329, 2013. Delft University of Technology (TU Delft).
2. **Z. Sun**, and N. Roos: Dynamic Lateral Stability for an Energy Efficient Gait, BNAIC 2014: *Proceedings of the 26th Benelux Conference on Artificial Intelligence, Nijmegen, the Netherlands*, 2013, pp. 234-242. Radboud University.
3. **Z. Sun**, and N. Roos: An energy efficient dynamic gait for a Nao robot, *Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2014, pp. 267-272.
4. **Z. Sun**, and N. Roos: A Dynamically Stable Gait Without Support Polygons, *under review*, 2017.

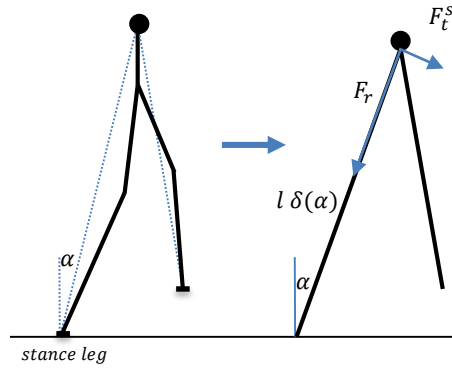


Figure 4.1: The abstract Inverted Pendulum Model. The left part of the figure is a 5-link model with point feet and imaginary massless virtual legs (dashed line). The right side of the figure is the Inverted Pendulum Model with telescopic virtual legs that we used to represent the physical structure of a biped robot. The definitions of angle  $\alpha$ , a radial force  $F_r$  and a tangential force  $F_t^s$  are shown in the right side of figure as well.

double support phase. So, there is no torque on the knee-joint of the stance leg during a step. In comparison to human walking, energy efficiency of modern walking robots is mostly inferior [140]. A promising way to enhance energy efficiency of locomotion is to use compliant elements, such as springs embedded in joints. Work at the actuators is reduced by energy storage and release through springs. But this method can not be implemented on standard robot platform, such as the Nao robot.

In this thesis, we demonstrate that a similar model as described by Srinivasan and Ruina can be applied for identifying an energy efficient gait for humanoid robots such as a Nao robot, despite differences with humans. First, unlike muscles, motors of a robot do not behave like springs. Second, our experiments with a Nao robot show that the energy consumption of a motor depends more on the torques of joints than on the work that is done. Therefore, torques on the joints need to be considered. Third, a human need not bend the knee of the stance leg while walking because (s)he can toe-off using the foot and the calf muscle. In this way the human can increase the length of the leg without torque on the knee joint. A robot such as a Nao robot cannot toe-off, because it can not bend its foot. This difference makes it possible to ignore the torque in the human model, but not in models of certain robots such as a Nao robot.

The model that we first consider is a planar kinematic chain abstracted from a Nao robot, which has a complex body with irregular shape and mass distribution. We assume that the robot can be represented as a point mass located in the CoM position which is connected to two identical legs at a single joint hip. The feet are represented as two points, as depicted in the left side of Figure 4.1. All motions are assumed to take place in the two-dimensional sagittal plane and consist of successive phases of single support (meaning the stance leg touches the walking surface and the swing leg does not) and double support (the swing leg and the stance leg are both in contact with the walking surface).



To reduce the total energy cost, we set the stiffness on both ankle joints to zero. Thus, the stance leg of the robot can freely rotate around the ankle joints, and the area of support reduces to a point. Apart from reducing the energy cost, zero ankle stiffness also allows the robot to adapt to an uneven floor. Many researchers [56, 119, 141, 169] have presented planar walkers with zero ankle torques that can keep their balance. So, reducing the ankle torque in order to increase the energy efficiency should be feasible.

To analyze the energy consumption, we further simplified the model to an Inverted Pendulum Model with telescopic legs ([144, 145]). This model, which is based on the work of [140], ignores the physical structure of both legs, and use an imaginary massless segment, the *virtual leg*, connecting the point mass to a ground contact point instead. Furthermore, this model allows the length of the virtual support leg to vary during a step. A leg-length policy  $\delta : [-\frac{\pi}{2}, \frac{\pi}{2}] \rightarrow [0, 1]$  determines how much the virtual support leg will be shortened as a function of the angle between the stance leg and the vertical axis. The shortening of the stance leg is realized by bending the knee joint. When the body is supported by a single leg, the model can be simplified to an inverted pendulum as shown in the right side of Figure 4.1.

In the following sections, the Inverted Pendulum Model is used to analyse the dynamics and to identify a low energy cost walking gait. The kinematic model in sagittal plane is discussed in Section 4.2, while the dynamics of the model in frontal plane is described in Section 4.3. Section 4.4 presents the performed experiments and discusses the results. Section 4.5 gives the conclusions of this chapter and answers the research questions.

## 4.2. KINEMATIC MODEL IN THE SAGITTAL DIRECTION

In this section, a simplified kinematic model (the Inverted Pendulum Model) is used to analyze the relation between the gait of a robot and its energy consumption. The kinematics of this model consist of a continuous-time single support phase and an instantaneous double support phase. The single support phase (SSP), or swing phase, begins when the swing foot leaves the ground and ends at heel-touch.

If we were to know the force produced by the leg, we could apply Newton's second law to derive the second order equation for the movement of the mass  $m$ . Since the ground generates a reaction force which can become large at the moment a stretched leg impacts with the ground, and since the length of the leg is bounded by  $l$ , we choose a different approach. We assume a leg-length policy  $\delta(\alpha)$  where  $\delta : [-\frac{\pi}{2}, \frac{\pi}{2}] \rightarrow [0, 1]$  and where  $\alpha$  is the angle between the stance leg and the vertical axis and  $l\delta(\alpha)$  is the actual length of the leg, as shown in Figure 4.1. We use the leg-length policy to determine the radial force  $F_r$  on the leg and a tangential force  $F_t^s$  perpendicular to  $F_r$ . Note that  $F_t^s$  works perpendicular to the telescopic leg while the path of the mass  $m$  need not be perpendicular to the leg because the length of the leg may change. The superscript  $s$  in  $F_t^s$  refers to the **single** support phase. Using the force  $F_t^s$ , we can determine the second derivative of the position of  $x_t$  w.r.t. the time  $t$  which is given by:

$$F_t^s = ma = m \frac{d^2 x_t}{dt^2}$$

$F_t^s$  is determined by the component of the gravity working perpendicular to the stance

leg:  $mg \sin \alpha$  and the friction:  $b \frac{dx_t}{dt}$ . The air friction and the ankle joint friction (if an ankle is present) reduce the forward speed of a walking robot. The air friction is quadratic in the speed but can be assumed linear because of the low walking speed. The ankle joint friction depends on the connected gearbox and motor. This friction is also assumed to be linear in the speed. The constant  $b$  captures both types of friction.

$$\frac{d^2 x_t}{dt^2} + \frac{b}{m} \frac{dx_t}{dt} - g \sin \alpha = 0 \quad (4.1)$$

We can transform this equation into a change of the angle  $\alpha$  using:

$$d\alpha = \frac{dx_t}{\delta(\alpha)l}$$

Therefore,

$$\begin{aligned} \frac{dx_t}{dt} &= \delta(\alpha)l \frac{d\alpha}{dt} \\ \frac{d^2 x_t}{dt^2} &= l \frac{d\delta(\alpha)}{dt} \frac{d\alpha}{dt} + \delta(\alpha)l \frac{d^2 \alpha}{dt^2} \\ &= l \frac{d\delta(\alpha)}{d\alpha} \left( \frac{d\alpha}{dt} \right)^2 + \delta(\alpha)l \frac{d^2 \alpha}{dt^2} \end{aligned}$$

This results in:

$$\frac{d^2 \alpha}{dt^2} + \frac{1}{\delta(\alpha)} \frac{d\delta(\alpha)}{d\alpha} \left( \frac{d\alpha}{dt} \right)^2 + \frac{b}{m} \frac{d\alpha}{dt} - \frac{g}{\delta(\alpha)l} \sin \alpha = 0 \quad (4.2)$$

To use this second order differential equation, we reformulate it as a system of first order differential equations.

$$\begin{aligned} \frac{d\omega}{dt} &= -\frac{1}{\delta(\alpha)} \frac{d\delta(\alpha)}{d\alpha} \omega^2 - \frac{b}{m} \omega + \frac{g}{\delta(\alpha)l} \sin \alpha \\ \frac{d\alpha}{dt} &= \omega \end{aligned} \quad (4.3)$$

The length of the stance leg at the beginning and at the end of a step, is denoted by  $l_b$  and  $l_e$  respectively. They need not be the same. We therefore need to know the angles  $\alpha$  at the beginning and at the end of a step, denoted by  $\alpha_b$  and  $\alpha_e$  respectively, given a fixed step length  $\theta_1$ . The leg-length policy should describe the changes in the leg length between these angles. To determine the angles  $\alpha_b$  and  $\alpha_e$ , we apply the cosine rule. Hence,

$$\alpha_b = \arcsin \frac{\theta_1^2 + l_b^2 - l_e^2}{2\theta_1 l_b} \quad \alpha_e = \arcsin \frac{\theta_1^2 + l_e^2 - l_b^2}{2\theta_1 l_e}$$

#### FORCE ON THE STANCE LEG

The solution of the above differential equations also enables us to determine the radial force on the stance leg. This force together with the leg-length policy  $\delta(\alpha)$  determines

the energy consumption of the knee joint of the stance leg. The radial force consists of a gravitational component and a component needed to accelerate the mass in the direction of the radius. The former is a reaction force equal to:  $mg \cos \alpha$ , and the latter is determined by the second derivative of the leg-length policy:

$$\begin{aligned}\frac{dl\delta(\alpha)}{dt} &= l \frac{d\delta(\alpha)}{d\alpha} \frac{d\alpha}{dt} \\ \frac{d^2l\delta(\alpha)}{dt^2} &= l \left( \frac{d^2\delta(\alpha)}{d\alpha^2} \left( \frac{d\alpha}{dt} \right)^2 + \frac{d\delta(\alpha)}{d\alpha} \frac{d^2\alpha}{dt^2} \right)\end{aligned}$$

So,

$$F_r = mg \cos \alpha + ml \left( \frac{d^2\delta(\alpha)}{d\alpha^2} \left( \frac{d\alpha}{dt} \right)^2 + \frac{d\delta(\alpha)}{d\alpha} \frac{d^2\alpha}{dt^2} \right) \quad (4.4)$$

When the swing foot impacts with the ground at the end of a step, the direction in which the mass  $m$  is moving may change. We assume an *inelastic* collision of the swing foot with the ground. This implies that the speed of the mass  $m$  in the radial direction of the swing leg is becomes equal to the radial speed  $v_{b,r}$  of the stance leg at the beginning of a step. To ensure a constant walking speed, the tangential speed of the swing leg at the end of a step must be the same as the tangential speed  $v_{b,t}$  of the stance leg at the beginning of a step. To compute the tangential speed of the swing leg at the end of a step, we first compute the tangential and radial speed of the mass  $m$  w.r.t. the stance leg, and the corresponding speeds in the Cartesian coordinate system.

$$\begin{aligned}v_t &= l\delta(\alpha) \frac{d\alpha}{dt} & v_r &= \frac{dl\delta(\alpha)}{dt} = l \frac{d\delta(\alpha)}{d\alpha} \frac{d\alpha}{dt} \\ v_x &= v_t \cos(\alpha) + v_r \sin(\alpha) & v_z &= v_t \sin(\alpha) + v_r \cos(\alpha)\end{aligned}$$

The robot is walking at a constant speed if the following equation holds:

$$v_{t,b} = (v_{z,e} \sin(\alpha_b) + v_{x,e} \cos(\alpha_b))$$

Note that  $\alpha_b$  specifies the angle of the swing leg at the end of step.

The law of momentum conservation applies to the impact of the swing leg with the ground. Therefore, the sum of the moments before and after the foot impacts with the ground must be 0. This does not imply that the impulse generated by the reaction force of the leg when it impacts with the ground is 0. The inelastic collision implies that the impulse must stop the movement in the radial direction of the swing leg (the *new* stance leg). The impulse generated by the reaction force of the new stance leg is determined by the change in speed of  $m$  in the direction of the new stance leg:

$$I_r = v_{r,b}m - v_{r,e}m$$

Here  $I_r$  denotes the impulse in the radial direction of the new stance leg,  $v_{r,b}$  denotes the radial speed of the new stance leg at the beginning of a step, and  $v_{r,e}$  denotes the radial speed of the swing leg at the end of a step. After the foot impacts with the ground, the

speed in the radial direction of the new stance leg is completely determined by leg-length policy because of the inelastic collision; i.e., no bounce occurs:

$$v_{r,b} = \frac{dl\delta(\alpha)}{dt}(t_b)$$

So, the impulse produced by the leg becomes:

$$I_r = m \left( \frac{dl\delta(\alpha)}{dt}(t_b) - (v_{x,e} \sin(\alpha_b) + v_{z,e} \cos(\alpha_b)) \right)$$

The impulse also equals:  $I_r = \int F_i dt$ . In an ideal situation, the impact time with the ground is infinitely small implying an infinitely large reaction force produced by the leg on the mass  $m$ . In practice material always bends or compresses somewhat. This increases the impact time and thereby reduces the reaction force of the leg. Although the force will now be finite, even if we would know the impact time, we cannot calculate it. We therefore make the assumption that the reaction force is constant during the impact. This assumption makes it possible to calculate the force on the leg for a given impact time  $t_{imp}$ .

$$F_i = \frac{I_r}{t_{imp}} \quad (4.5)$$

#### OPTIMAL LEG-LENGTH POLICY

To identify the leg-length policy that minimizes the energy consumption of a robot, we make use of the fact that the robot has to bend the knee in order to shorten the leg. The energy consumption is observed to be proportional to the torque of these joints in experiments with a Nao robot (see the stand-up experiment described in subsection: 4.4.1). So a stretched leg requires no energy while a largely bent leg requires a maximum amount of energy. We will use this observation to determine the energy consumption in the model. The torque on the knee joint is determined by the radial force and arm of this

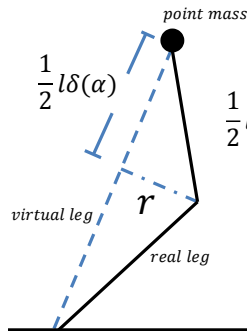


Figure 4.2: The arm  $r$  determines the knee torque

force w.r.t. the knee joint (see Figure 4.2). The arm is given by:  $r = \frac{1}{2}l\sqrt{1 - \delta(\alpha)^2}$ .

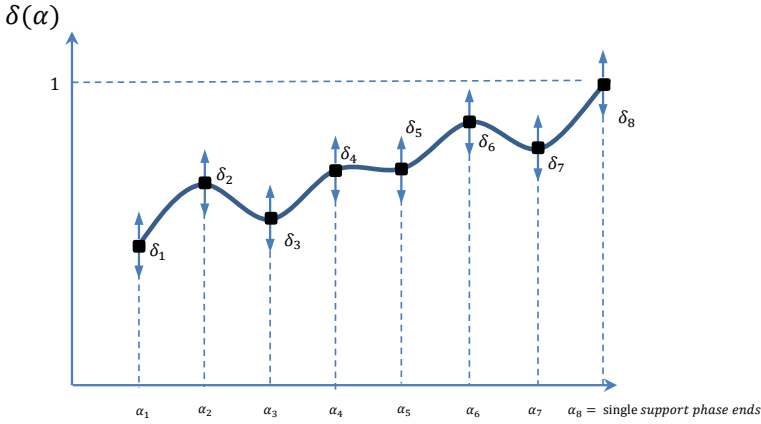


Figure 4.3: CoM trajectory in single support phase controlled by leg-length policy  $\delta(\alpha)$

The radial force  $F_r$  acting on the mass  $m$  can become high when the leg impacts with the ground. So, keeping the leg stretched implying  $r = 0$  minimizes the energy consumption. Also no energy is provided to the system, and therefore, because of the friction, the robot will not continue walking. To provide energy, the leg must bend and stretch again. The energy is used to provide the necessary torque and the positive work of lifting the mass  $m$  when stretching the leg. Ignoring the positive work and assuming an impact time  $t_{imp}$ , we define the energy consumption as:

$$E \propto \int_0^{t_{imp}} \frac{1}{2} l \sqrt{1 - \delta(\alpha)^2} (F_r^s + F_i) dt + \int_{t_{imp}}^{t_s} \frac{1}{2} l \sqrt{1 - \delta(\alpha)^2} F_r^s dt \quad (4.6)$$

Here,  $t_s$  is the duration time of a step.

We can derive the energy consumption  $E_{ic}$  of an inelastic collision with an infinitely small duration from Equations 4.5 and 4.6.

$$\begin{aligned} E_{ic} &= \lim_{t_{imp} \rightarrow 0} E \\ &\propto \frac{1}{2} l \sqrt{1 - \delta(\alpha_b)^2} I_r + \int_0^{t_s} \frac{1}{2} l \sqrt{1 - \delta(\alpha)^2} F_r^s dt \end{aligned} \quad (4.7)$$

The above described model has been used to identify an energy efficient gait by evaluating different leg-length policies. We systematically generated different leg-length policies that can be described by cubic splines. We divided the SSP into 7 intervals and chose a leg-length for the beginning of each interval plus the end of the last interval. The cubic splines described the leg-length trajectory through 8 points, see Figure 4.3. We investigated all possible policies with a resolution of 7 intervals, computed their energy costs, made a comparison, then found an optimal policy that minimizes the power usage. The optimal leg-length policy that was identified, starts the single support phase with a slightly bent stance leg. Subsequently, as the CoM keeps moving forward, the

stance leg gradually stretches until its full length. The stance leg remains stretched till the end of the step. Figure 4.4 shows the leg-length policy  $\delta(\alpha)$  as a function of the angle  $\alpha$  from the beginning till the end of the step and Figure 4.5 shows the realization using the 5-link model.

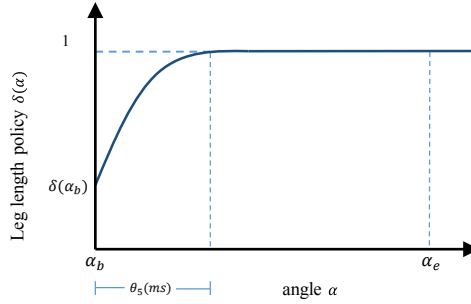


Figure 4.4: The optimal leg-length policy

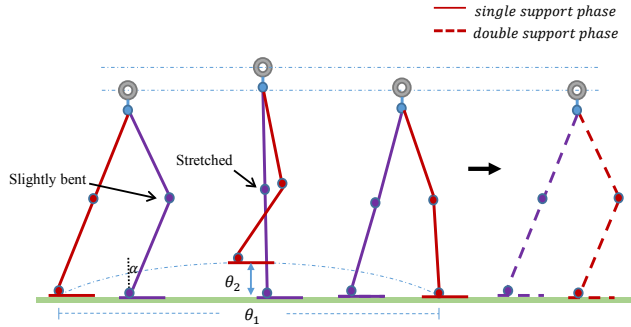


Figure 4.5: The kinematics of the sagittal motion

It is important to note that the ankle joints are underactuated in the realization shown in Figure 4.5. This implies that no stability criteria such as the ZMP criterion are used in the model, and that the robot *falls* forward till the swing leg takes over the support of the robot. As shown in Figure 4.5, the swing leg may touch the ground before it takes over the support of the robot. Since the swing leg only takes over the support of the robot at the end of a step, the knee joint of the swing leg is underactuated during the double support phase in this planar model.

#### DOUBLE SUPPORT PHASE EXTENSION

In the double support phase (DSP) the robot has to shift its weight from the stance leg to the swing leg. In order to be balanced in the lateral direction at the end of the double support phase, the robot must put force on the swing leg to stop the lateral movement in time. This force also influences the movement in the sagittal plane.

We need a way to describe the influence of the swing leg on the sagittal movement of the CoM in the DSP. This cannot be done by simply applying a leg-length policy for the

swing leg in the double support phase. Given the step length  $\theta_1$ , the leg-length policy  $\delta(\alpha)$  of the stance leg and the angle  $\alpha$ , the length of the swing leg in the DSP is fixed. Prescribing the length of the swing leg by a policy, the two legs and the ground form a rigid triangle in which the mass  $m$  can no longer move freely. We therefore chose to let the mass  $m$  move freely given the leg-length policy of the stance leg and use a *force policy* for the swing leg in the double support phase. The force prescribed by the force policy does not only influence the lateral movement but also the sagittal movement. In this Subsubsection we investigate the latter influence.

We extended the 2 dimensional planar model described in the previous subsection with a double support phase. We stop the single support phase at some point of a step; i.e., at the angle  $\alpha_d = \alpha_b + \theta_9(\alpha_e - \alpha_b)$  where  $\alpha_b$  and  $\alpha_e$  are the beginning and the end angle of the stance leg during a step, respectively. ( $\theta_9$  is one of the controller's parameters which determines how long the single support phase lasts in one single step, as further defined in Section: 5.3.) From this moment the swing leg will also be on the ground, therefore influencing the movement of the mass  $m$ .

The force policy  $\gamma(\alpha')$  where  $\gamma: [-\frac{1}{2}\pi, \frac{1}{2}\pi] \rightarrow [0, 1]$  and where  $\alpha'$  is the angle of the swing leg with the vertical axis in the sagittal plane, will be expressed as a percentage of the force  $F_r$  on the stance leg, as depicted in Figure 4.6.  $F_r$  is the force on the stance leg caused by gravity and by the changes in the length of the stance leg. The force  $F_s^d$  generated by the swing leg causes a force  $F_o^d$  in the opposite direction of  $F_r$  and a force  $F_p^d$  perpendicular to the stance leg. The force policy  $\gamma(\alpha')$  determines  $F_o^d = \gamma(\alpha')F_r$ , and thereby it also determines the force generated by the swing leg  $F_s^d$  as well as the force  $F_p^d$ . Note that the angle  $\alpha$  of the stance leg with the vertical axis, the leg-length policy  $\delta(\alpha)$ , and the step length  $\theta_1$ , together determine the angle  $\alpha'$  of the swing leg with the vertical axis. Also note that the force policy guarantees that the stance leg stays on the ground till the DSP finishes. This means that the value of the force policy must be less than or equal to one.

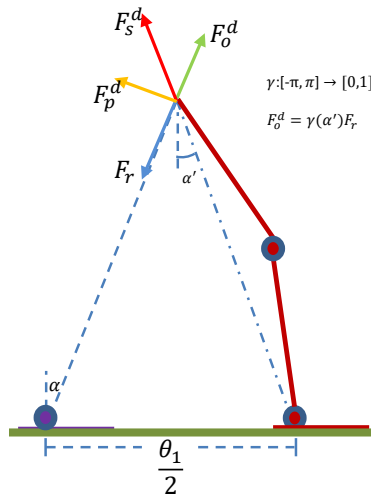


Figure 4.6: Force policy in double support phase

Taking the force policy into account, we can derive the following differential equations for the movement of the mass  $m$  in the sagittal plane during the DSP:

$$\frac{d^2\alpha}{dt^2} + \frac{1}{\delta(\alpha)} \frac{d\delta(\alpha)}{d\alpha} \left( \frac{d\alpha}{dt} \right)^2 + \frac{b}{m} \frac{d\alpha}{dt} - \frac{g \sin \alpha}{\delta(\alpha)l} + \frac{\gamma(\alpha') \tan(\alpha + \alpha') F_r}{ml\delta(\alpha)} = 0 \quad (4.8)$$

where  $F_r$  is defined in Equation (4.4).

After extending our model with the double support phase and the force policy for the swing leg during the DSP, we repeated the simulations. The results of these simulations showed that the shape of the leg-length policy did not change, the stance (rear) leg stays fully stretched ( $\delta(\alpha) = 1$ ) in the DSP. So, in Equation (4.8), the term  $\frac{d\delta(\alpha)}{d\alpha} = 0$ . Then we get:

$$\frac{d^2\alpha}{dt^2} + \frac{b}{m} \frac{d\alpha}{dt} - \frac{\sin \alpha - \gamma(\alpha') \tan(\alpha + \alpha') \cos \alpha}{l} g = 0 \quad (4.9)$$

The simulation in subsection 4.4.2 also showed that the optimal force policy puts no force on the swing leg till it becomes the new stance leg. Since shifting the weight from the stance leg to the swing leg in the DSP requires a force generated by the swing leg, we also evaluated the effects of different force policies on the leg-length policy. The experiments showed that the shape of the leg-length policy (shown in Figure 4.4) does not change. The robot still starts with a slightly bent leg which is subsequently stretched and remains stretched till the end of the step.

### 4.3. KINEMATIC MODEL IN THE LATERAL DIRECTION

For a simple forward step, it is insufficient to only consider 2D dynamics in the sagittal direction. To achieve the lateral stability, we designed a lateral controller to regulate the CoM lateral movement during the double support phase. Our idea is to use the force generated by the swing leg and the upper body tilt to regulate the lateral component of CoM velocity. Since the sagittal and frontal planes are *orthogonal*, we can analyze the lateral component of the forces working on the CoM independently of the sagittal component of the forces working on the CoM. Therefore, we restrict the equations to the frontal plane.

In order to introduce the equations describing the movement of the CoM in the frontal plane, we first define a variable  $\beta$  which is the angle between the stance leg and the vertical axis in the frontal plane. When the total force resulting from gravity and inertia generates rotation around the contact point between the sole of the stance leg and the ground, the angle  $\beta$  varies from 0 to  $\beta_e$ , as illustrated in Figure 4.7.

We assume that during the single support phase, the robot is perfectly balanced in the lateral direction. Therefore, at the beginning of the DSP, the CoM is vertically above the center of the stance foot ( $\beta = 0$ ), and in the frontal plane there is no torque to make the CoM rotate around the sole of the stance leg. Note that the heel strike of the swing foot does not disturb the sagittal movement because at the start of the DSP both the ankle and knee joint of the swing leg are underactuated. For the same reason, the heel strike introduces no lateral torque that may change the moving direction of the CoM. In order to balance the CoM in the lateral direction above the swing foot (the next stance foot) at the end of the DSP, we need to generate a torque  $\tau$  rotating the CoM from 0 to



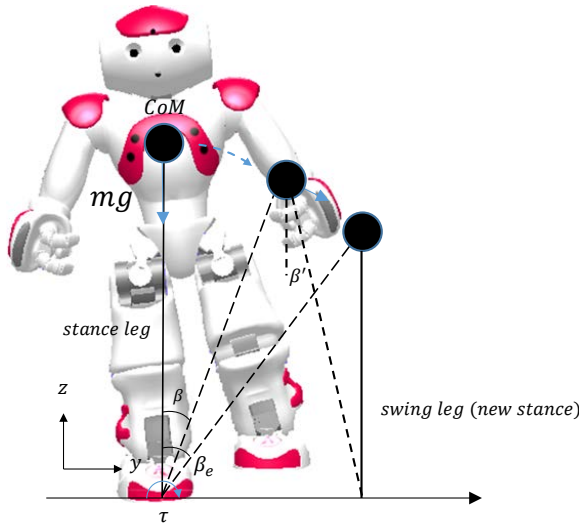


Figure 4.7: The frontal plane during double support phase

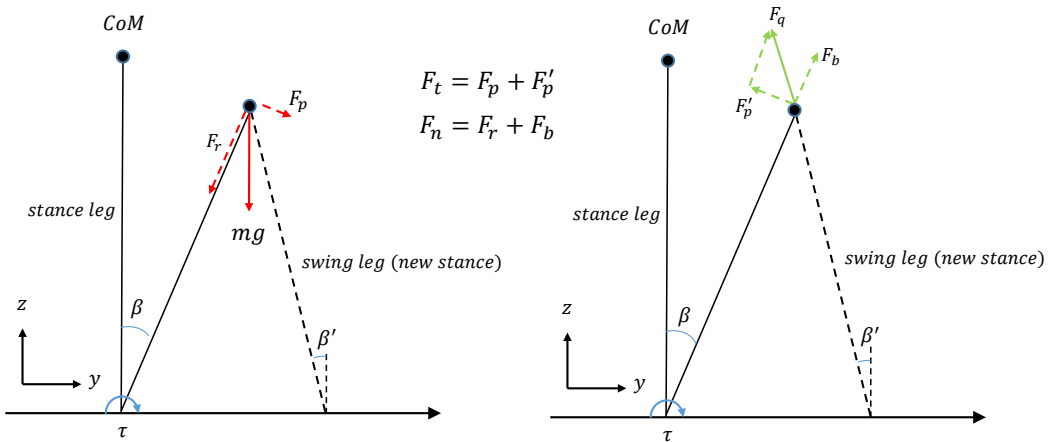


Figure 4.8: (a) Force on stance leg in frontal plane and (b) Forces on swing leg in frontal plane

$\beta_e$ . In order to generate the torque  $\tau$ , we manipulate the upper body to bend slightly inwards at the angle  $\theta_8$  for 100 ms. The bending  $\theta_8$  disrupts the balance enabling gravity to create a torque  $\tau > 0$ . We manipulate the force generated by the swing leg to control the rotation of the CoM with a non-zero angular velocity  $\beta$  and to stop the rotation at the position ( $\beta = \beta_e$ ) where the robot can put its whole body weight on the new stance leg and keep it stable. The problem is to control the torque  $\tau$  appropriately. Our method to mitigate this problem, is controlling the force generated by the swing leg by means of a force policy  $\gamma : [-\frac{1}{2}\pi, \frac{1}{2}\pi] \rightarrow [0, 1]$ . Unlike in the previous subsection, here we define the force policy with respect to the angle  $\beta$  of the telescopic leg with the vertical axis in the

frontal plane.

Compared to the height of the CoM, the step length is small (less than 5% of CoM height). Therefore we assume that in the DSP, the length of the stance leg can be considered as fixed in the frontal plane. That is, for the stance leg, we ignore the effect of the CoM moving forward in the sagittal plane. Given this assumption, we analyze the forces on the CoM in the frontal plane.

We break down the gravity into two components consisting of a radial force  $F_r$  along the stance leg and  $F_p$  that is perpendicular to the stance leg (see Figure 4.8(a)). We manipulate the force policy  $\gamma(\beta)$  of the swing leg to control the force  $F_q$  generated by the swing leg on the CoM. We also break down  $F_q$  into two components:  $F_b$  which is directed opposite to  $F_r$ , and another force  $F'_p$  which is opposite to  $F_p$  (see Figure 4.8(b)). Therefore, in the radial direction, there is the combined radial force  $F_n = F_r + F_b$ . Since the length of the stance leg  $l$  is fixed, the radial force  $F_n$  has no effect on the movement of the CoM. However, the combined force  $F_t = F_p + F'_p$  generates a torque  $\tau$  around the contact point between the sole of the stance leg and the ground. The torque  $\tau$  rotates the CoM, shifting the weight from the stance leg to the swing leg.

At the beginning of the DSP, ideally, the force policy  $\gamma(\beta)$  imposes *no* force on the swing leg. Therefore the torque  $\tau$  generated by the slightly inward bending  $\theta_8$  of the upper body, is needed to start the lateral movement of the CoM. (In our experiments, the bending lasts for 100 ms.) As  $\beta$  increases, the force policy controls the force  $F_q$  to gradually decrease the  $\dot{\beta}(t)$ , and stops the CoM movement when  $\beta = \beta_e$ . The force  $F_q$  causes the force  $F'_p$  which decelerates the movement of the CoM. When  $\beta = \beta_e$ , the force  $F_t$ , the torque  $\tau = F_t \cdot l$  and the angular velocity  $\dot{\beta}$  become 0, therefore the CoM stabilizes above the swing leg (the new stance leg). We generate a force policy by regulating the knee stiffness of the swing leg, as a function of the angle  $\beta$ . The shape of the force policy is determined by means of a Quadratic Bezier curve, as illustrated in Figure 4.9. This Quadratic Bezier curve is defined by 3 points in the interval of the DSP. The start point and the end point are fixed, so we start with no force generated by the swing leg and stop with the full weight of the robot on the swing leg, after which it becomes the new stance leg. We assume a smooth transition between these two points which is determined by the middle point  $\theta_7$  of the Quadratic Bezier curve. So we have to determine the optimal point  $\theta_7$ . The optimization of the point  $\theta_7$  will be discussed in the Chapter 5.

To summarize, the controller manipulates the upper body to bend slightly inwards at angle  $\theta_8$  to trigger the CoM movement. At first the knee joint of the swing leg is underactuated and the forces  $F_q$  and  $F'_p$  are 0. Consequently,  $F_t = F_p > 0$  leads to an acceleration of the angular velocity of CoM around the point-foot of the stance leg, which forces the underactuated knee joint of the swing to bend. Next, the stiffness of the knee joint of the swing leg is increased in order to stop the rotation when the CoM reaches its end position ( $\beta = \beta_e$ ). When the CoM reaches its end position, the lateral rotational velocity  $\dot{\beta}$  and acceleration  $\ddot{\beta}$  become 0.

Since the point-foot of the swing leg is in front of the point-foot of the stance leg in the sagittal plane, the increasing stiffness of the knee joint of swing leg will also influence the forward speed in the sagittal plane. We showed in Subsection 4.2 that this does not influence the shape of the leg-length policy.

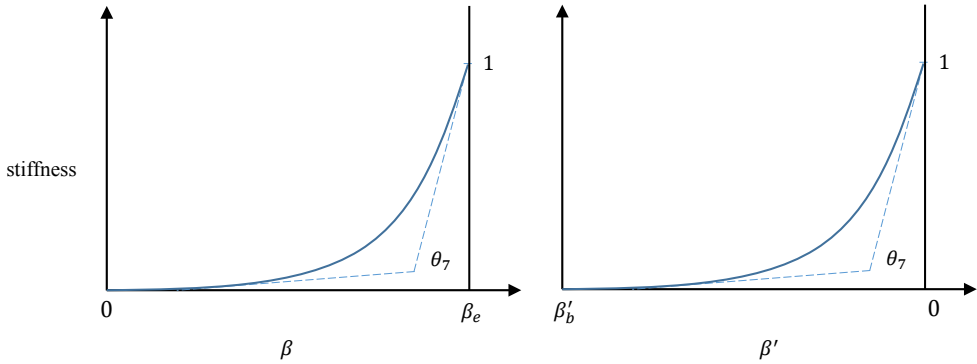


Figure 4.9: Stiffness over angle  $\beta$  and  $\beta'$  by Quadratic Bezier Curves

## 4.4. EXPERIMENTS

In this section, we introduce the experiments which address the assumptions that were used in the Inverted pendulum model described in this chapter and evaluate the effects of different force polices on the leg-length policies.

### 4.4.1. VALIDATING ASSUMPTIONS

We need to validate two assumptions through experiments, namely,

- the assumption that the knee and ankle pits joints are the main contributors of the energy consumption, and
- the assumption that the torque and not the work done is the main factor determining the energy consumption of a joint.

We evaluate these assumptions by measuring the currents to the hip, knee and ankle joints. Given a constant voltage, the currents are linearly proportional to the power consumptions of the joints.

#### POWER CONSUMPTION OF JOINTS

We designed an experiment to find the relation between stiffness and maximum drawn current on each joint. In this experiment, a free joint movement with different stiffness values is conducted. We make the robot march in one spot without moving forward with the different joint stiffness. The test was repeated for stiffness values between 0.8 and 1.0 with increments of 0.05. The same experiment was done five times for each stiffness value. Averaged data over five experiments were used to find the average current for that specific joint.

The current in each joint of the left leg for different stiffness values is shown in Table 4.1. The data shows the joints that are responsible for most of the energy consumption. With the same stiffness value, the largest contributor is the knee joint which uses about 35.2% ~ 40.4% of the whole body energy consumption. It is a large improvement in efficiency if we improve the energy cost in the knee joint through reducing the knee flexion.

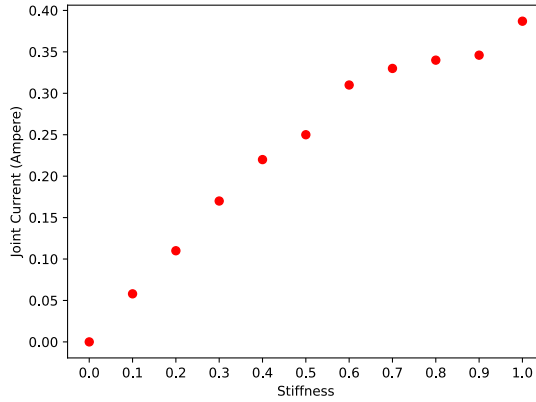


Figure 4.10: The averaged current drawn by the ankle pitch joint as a function of the joint's stiffness.

The ankle pitch joint is the next largest contributor to the whole body energy consumption. Figure 4.10 shows that the current in the ankle pitch joint is positively related to the joint stiffness, which implies we can improve the energy efficiency by decreasing the ankle pitch stiffness. Since we did not consider the feet as part of the Inverted Pendulum model, theoretically in the real robot, we can set the ankle pitch stiffness to 0 in order to improve energy efficiency without influencing the walking stability. As discussed in Subsection 5.2.2, it is necessary to set the ankle pitch stiffness to 0.1 in order to assure that the stance foot and the swing foot in the double support phase, stay flat on the ground and have sufficient grip to propel the Nao robot. The ankle roll stiffness is set to 0.

Table 4.1: Comparison of electric current (Ampere) on Nao(v5)'s joints with different stiffness

		Current (Ampere) for each joint of Nao					
		Knee	Ankle Roll	Ankle Pitch	Hip Pitch	Hip Roll	Hip Yaw
Stiffness	1.00	0.546	0.065	0.387	0.165	0.175	0.212
	0.95	0.536	0.054	0.369	0.143	0.168	0.186
	0.90	0.523	0.045	0.346	0.131	0.152	0.157
	0.85	0.517	0.043	0.342	0.102	0.143	0.131

#### POWER CONSUMPTION AND JOINT TORQUE

The goal of this experiment is to validate the assumption that energy consumption is positively related to the joint torque on the knee joint. In this experiment, we changed the position of the Nao from standing upright (knee angle =  $0^\circ$ ) to a position where the angle is around  $90^\circ$ . We kept the Nao for a while in this position after which we let the Nao move back to the upright position. During this experiment, we monitored the current of a knee joint, as shown in Figure 4.11.

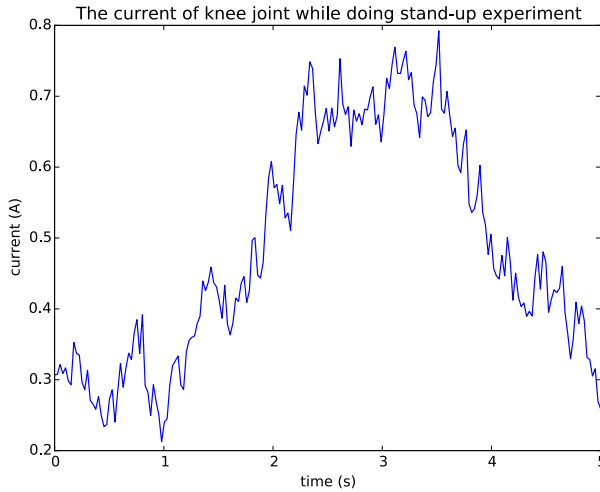


Figure 4.11: The current of knee joint while doing stand-up experiment

When standing upright it is around 0.3A. When bending the knees, the current gradually increases. In the fixed bent position, the current is 0.7A. Finally, when the Nao moves back to the upright position, the current gradually drops from 0.7A to 0.3A. Based on this experiment, we conclude that the torque determines energy consumption and that the positive work can be ignored.

#### 4.4.2. EFFECTS OF THE FORCE POLICES ON THE LEG-LENGTH POLICY

This subsection presents the experiment to evaluate the effects of different force polices on the leg-length policies. In Subsection 4.2, a force policy was introduced in order to determine the influence of the swing leg on the sagittal plane in the double support phase. The feasible force policy should satisfy the following constraint:

1. With the force policy, the energy consumption of the gait should be minimal.
2. The angle of the knee joint at the end of the DSP should be equal to the one of the same joint at the beginning of the SSP.

As discussed in Section 4.2, the energy consumed by the knee joint of the stance leg is used to provide the necessary torque, and the positive work of lifting the mass  $m$  when stretching the leg. In the DSP, the torque generated by the knee joint of the swing leg determines the force policy. This torque increases the energy cost of a step.

The force policy can reduce the impulse generated by the heel strike of the swing leg, but also influences the movement in the sagittal plane. We therefore evaluated different combinations of leg-length policies and force policies. Similar to the way the optimal leg-length policy was found, we generated different force policies that were described by cubic splines. We first divided the double support phase into 5 time intervals and chose

a force policy  $\gamma(\alpha')$ , i.e. a value between 0 and 1, for the beginning of each interval and for the end of the last interval. The cubic splines described the force policy trajectory through 6 points. We investigated all possible combinations of a leg-length policy and a force policy to find the optimal force policy minimizing the energy cost. The optimal leg-length policy did not change (see Figure 4.4) and the optimal force policy puts no force on the swing leg till the end of the step (see Figure 4.12).

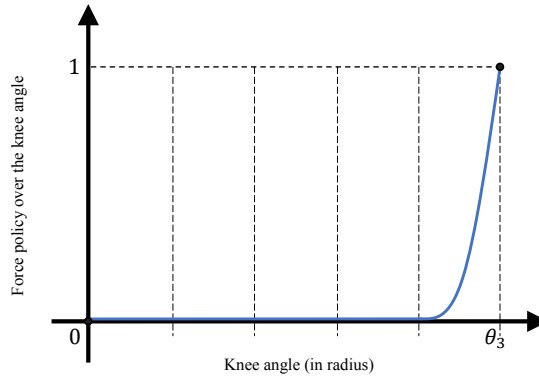


Figure 4.12: The optimal force policy

We also investigated the effects of different force policies on the optimal leg-length policy. From Figure 4.13, we can observe that, despite the different force policies applied in the DSP, the experiments showed that the shape of the leg-length policy (shown in Figure 4.4) does not change. Independent of the force policy, the robot still starts the single support phase with a slightly bent leg which is subsequently stretched and remains stretched till the end of the step. This does not mean that the force policy has no impact on the optimal leg-length policy. Figure 4.13 shows that the corresponding leg-length policies have a different starting length  $\lambda(\alpha_b)$ . Moreover, the time needed to stretch the leg varies slightly due to the different force policies, but the differences are so small that for most practical purposes they can be neglected.

## 4.5. CONCLUSION

In this chapter, we have investigated the possibility of creating an energy efficient gait without an area of support. Here, the absence of an area of support means that the ankle joint is free to turn in any direction while the foot is on the ground. For simplification purposes, the walking motion is considered to be a steady walking pattern, consisting of successive single support phases separated by a double support phase. These double support phases are used for the exchange of support between the two feet. In the sagittal direction the robot is falling forward till the foot of the swing leg touches the ground. In the lateral direction, the robot balances above the stance foot in the single support phase, and falls towards the new stance foot in the double support phase. The falling towards the new stance foot is stopped by putting a force on the new stance leg. Furthermore, we validated our assumptions about the main factors determining the energy consumption of a gait by conducting experiments on a real Nao robot. With these confirmed assump-

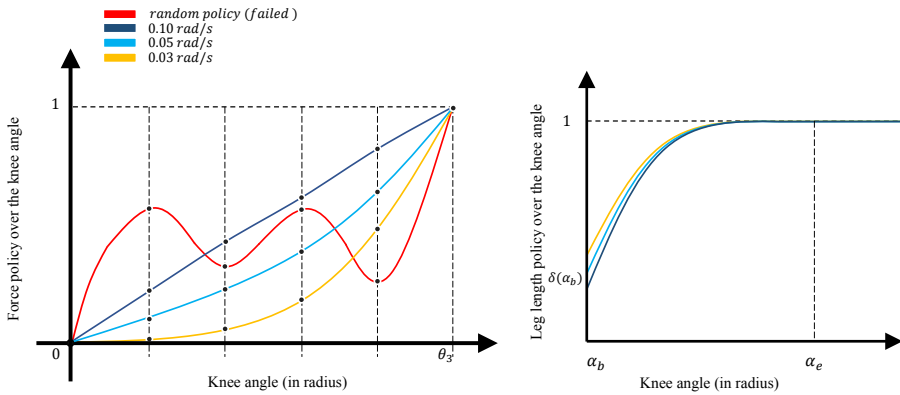


Figure 4.13: The different force policies with different initial speed and corresponding optimal leg-length policies

tions, we further conducted experiments to obtain the optimal leg-length policy and the optimal force policy which satisfy the kinematic model and minimize the energy consumption. Lastly, we established that the inclusion of our proposed force policy has no significant effect on the leg-length policy.

**Research Question** The results of this chapter provide us with partial answers on research questions 1 to 3.

1. *Can the planar inverted pendulum model be used to create a bipedal gait with a high energy efficiency?*

The biped gait based on a conventional Linear Inverted Pendulum Model restrains the robot's CoM to a horizontal plane, hence the legs have to bend most of the time when the robot is walking. In Subsection 4.4.1 we observed that the energy consumption of the joint motor is mainly related to the applied torque. It implies the possibility that energy efficiency of the biped gait can be optimized. We proposed a leg-length policy to describe the change of the virtual leg length and optimized the policy to find one that results in a walking gait with less energy cost. The optimal leg-length policy starts the single support phase with a slightly bent stance leg. Subsequently, as the CoM keeps moving forward, the stance leg gradually stretches until its full length. The stance leg remains stretched till the end of the step. Figure 4.4 shows the leg-length policy  $\delta(\alpha)$  as a function of the angle  $\alpha$  from the beginning till the end of the step and Figure 4.5 shows the realization using a 5-link model.

2. *What is the influence of the double support phase on the energy efficient gait ?*

A complete biped walking cycle is composed of two phases: a double support phase and a single support phase. During the double support phase, both feet are in contact with the ground. Many studies on gait planning with the Inverted Pendulum Model ignore the double Support Phase [72, 74, 79, 142] or assume that the double support phase is instantaneous [88, 136, 174]. In such cases, the robot's

CoM must be transferred from the stance leg to the swing leg instantly after the single support phase ends. This phenomenon makes the smooth transition of the CoM have an additional acceleration, which increases the inertial force that induces further instability. Besides, from the viewpoint of energy efficiency, the double support phase is important since, during the double support phase, the impact force generated by the swing leg striking the ground may introduce a torque on the knee joint of the swing leg which will increase the energy consumption. We extended the Inverted Pendulum Model with a double support phase and proposed a force policy to control the lateral movement of the CoM. Experiment showed the proposed force policy has no significant effect on the shape of the leg-length policy.

## 4

3. *What are the requirements of a double support phase w.r.t. the lateral stability of a gait?*

The robot should be perfectly balanced in the lateral direction during the SSP; i.e., no wagging like a penguin. Therefore, at the beginning of the DSP, the CoM is vertically above the center of the stance foot. During the double support phase, the CoM is supposed to transfer from the stance leg to the swing leg. At the end of the DSP, the projection of inertial and gravitational forces on the floor should be in the middle of the swing foot w.r.t. the lateral direction. This is necessary because the support polygon of the foot reduces to a point due to zero stiffness on the ankle joint. If a small disturbance occurs, the robot may overshoot or undershoot its stable end point, balancing above the swing foot. Therefore, a force control method is required. In this chapter, our method is to manipulate the force on the swing leg by means of a force policy which is generated by regulating the knee stiffness of the swing leg. To guarantee the smoothness of CoM transition and accurate timing control, the shape of the force policy is determined by a Quadratic Bezier curve, as illustrated in Figure 4.9.



# 5

## GAIT CONTROLLER DESIGN FOR FLAT FLOOR WALKING

IN this chapter, we describe a controller which implements a walking gait on level ground for Nao robots using the results of the previous chapters. Because of the differences between the abstract model and the Nao, several parameters of the controller need to be fine-tuned. We present the gait controller and its parameters, the learning algorithm that we applied, the evaluation criterion used by the learning algorithm, and the results of the learning process.

### 5.1. INTRODUCTION

In the previous chapter we have introduced a method for creating a low energy cost biped gait without involving support polygon. This gait is derived from the Inverted Pendulum Model and extended with a leg-length policy and a force policy that can be optimized to improve the gait's performance in stability and energy efficiency. To be able to establish the optimal policies and relevant control parameters, it is necessary to apply a feasible optimization method which leads to research question 4: "How to design the controller for gait generation?" and research question 5: "How to optimize the energy efficiency for the walking gait on a flat floor?".

---

This chapter is an updated and expanded version of the following publications:

1. **Z. Sun**, and N. Roos: An energy efficient gait for a Nao robot," in BNAIC 2013: *Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands*, pp. 323-329, 2013. Delft University of Technology (TU Delft).
2. **Z. Sun**, and N. Roos: Dynamic Lateral Stability for an Energy Efficient Gait," in BNAIC 2014: *Proceedings of the 26th Benelux Conference on Artificial Intelligence, Nijmegen, the Netherlands*, 2013, pp. 234-242. Radboud University.
3. **Z. Sun**, and N. Roos: An energy efficient dynamic gait for a Nao robot," in *Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2014, pp. 267-272.

This chapter aims to answer these questions by first introducing the structure and the operation of the gait controller in Section 5.2. We propose a parameter set for the gait controller (in Section 5.3). Those parameters are essential to the primary objective of determining the gait’s performance with respect to stability and energy efficiency. Next, in Section 5.4, we describe the policy gradient reinforcement learning algorithm used to search for a set of parameters that guarantee a stable and low energy cost gait. Section 5.5, describes the learning of the optimal set of parameter values in a simulation environment. Section 5.6 presents the experimental evaluation on a real Nao robot and discusses its results, and Section 5.7 gives the conclusion of this chapter and answers the research questions.

## 5.2. OVERVIEW OF THE GAIT CONTROLLER DESIGN

A block-diagram of the controller is shown in Figure 5.1. In the following description, we review the operation of this controller.

5

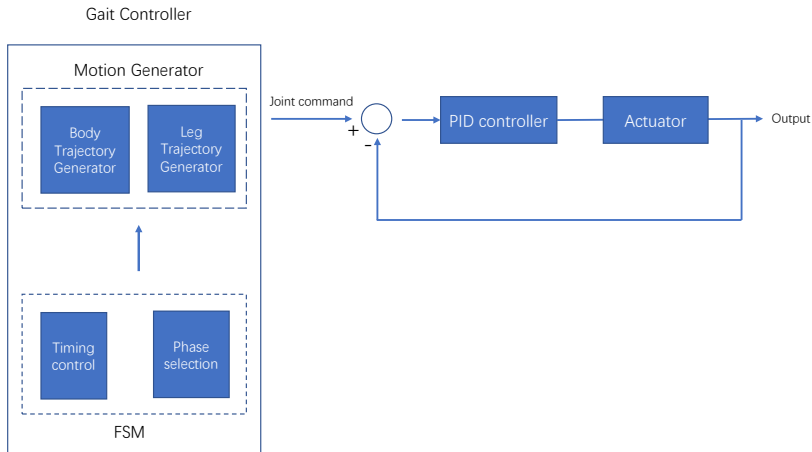


Figure 5.1: The operation overview of the gait controller

The gait controller on the left part of the Figure 5.1 consists of joint-level controllers (i.e., the motion generator), which are supervised by a high-level control structure (i.e., a Finite State Machine) that provides the appropriate joint functionality. Individual joint angles are computed by the motion generator, whose structure is organized into a body trajectory generator and a leg trajectory generator. The time-control of the motion generator is implemented by a finite state machine. The finite state machine operates on the time scale of the locomotion phases, i.e., the single support phase and the double support phase. The right part of the Figure 5.1 shows the control process of the Nao robot that implements the joint commands using a proportional–integral–derivative (PID) controller around each joint.

### 5.2.1. FINITE STATE MACHINE

A finite state machine represents an effective action selection mechanism widely used in the domain of robotics [98, 162]. In this work, we present a finite state machine (FSM) which guides the motion generator to produce the various trajectories that create desired motions. The FSM represents a sequence of motion states which are deterministic and the transitions are observable. The FSM allows motion states to be implemented independently of their successor states. As shown in Figure 5.2, the FSM consists of two motion states (single support phase and double support phase) and two transitions between the motion states. Based on the condition when the transition occurs, FSMs can be classified into two types: event-triggered, and time-triggered [163]. The transition in an event-triggered FSM will occur whenever an event is raised or an input is provided. Time-triggered FSM means that states transition occurs after some time interval.

In our case (i.e. a walk on the flat ground), the FSM used to guide the motion generator is time-triggered. The timer determines the duration of the motion states and the motions need to be executed within the set time interval. We choose a time-triggered approach because the duration of the same motion state in each step is identical and we want to learn the optimal duration of each state. In the next chapter about the walking gait on uneven terrain, these timers are replaced by observable events (i.e. "swing foot strikes the ground" and "CoM balances above the new stance leg") because the durations of the motion states are variable and dependent on the terrain.

The motion states are defined as follows:

- "Single support phase": In this state, the stance foot is on the ground, while the swing foot is in the air.
- "Double support phase": During this state, two feet are placed on the ground. The CoM transfers from stance leg to the swing leg (new stance leg) in both sagittal and lateral direction.

The impact of the swing foot, i.e. the heel-strike, with the ground is inelastic, because the torques on the knee and ankle joints of the swing leg are set to 0. This implies that during the instantaneous state transition, there is no slip and no bounce happening. When the impact takes place, the robot configuration remains the same and the angular momentum of the robot on the impact point is conserved.

### 5.2.2. LEG TRAJECTORY GENERATION

The gait identified in the previous section uses the abstract model of a robot consisting of a point mass and two telescopic legs. To implement the gait on a robot such as the Nao as defined in Chapter 4, we need to implement the leg-length and the force policy by controlling the leg joints of the Nao robot. Moreover, we need to generate a trajectory for the swing foot (making a step), and we need to address the control of the ankle of the stance leg.

#### LEG-LENGTH POLICY

The leg-length policy of the virtual telescopic stance leg, shown in Figure 5.3, must be implemented by controlling the knee joint of the actual stance leg of the Nao robot. Since

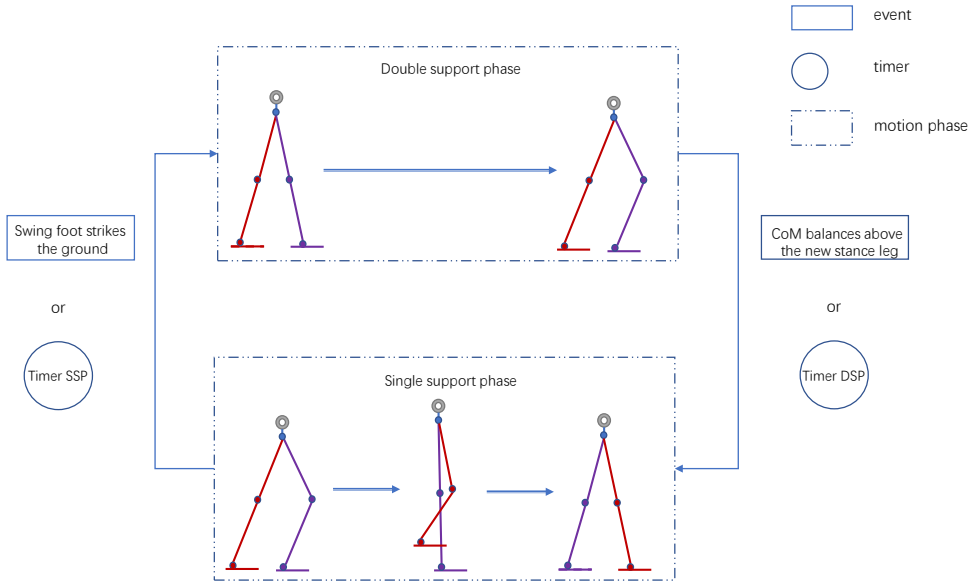


Figure 5.2: The motion states and transitions of the FSM

the stance foot is always flat on the ground, we need not consider the ankle joint. Moreover, we assume that the upper body is in a vertical position. This implies that we only need to control the knee joint of the stance leg. The body trajectory generator determines the angle of the hip pitch joint.

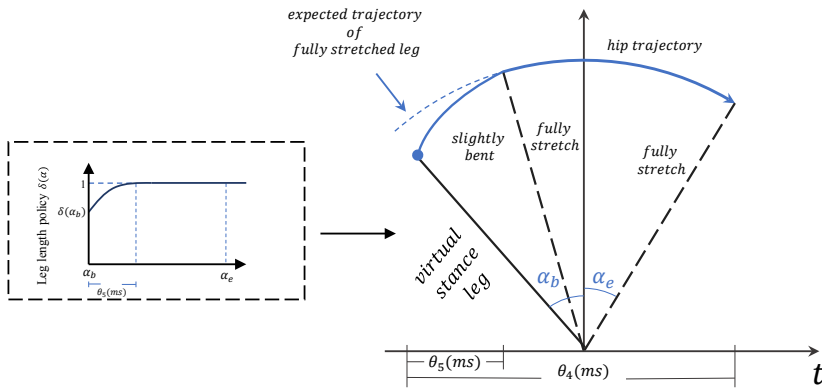


Figure 5.3: The mapping from the leg-length policy to the joints

Figure 5.4 shows the angles that we use to implement the leg-length policy. The leg-length policy changes the angle  $\lambda_{kst}$  of the knee joint from the initial angle  $\theta_3$  at the start of the single support phase to 0 in  $\theta_5$  seconds during the single support phase. Both  $\theta_3$  and  $\theta_5$  are control parameters of the controller, which can be adapted and will be

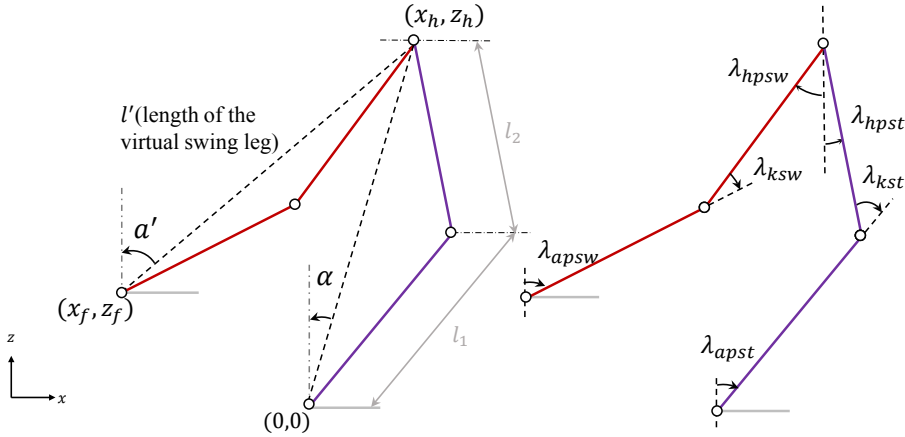


Figure 5.4: The definition of a virtual leg and key angles of a biped model in the sagittal plane

optimized; see Sections 5.3 and 5.4. We have chosen to use sinusoid to generate a smooth transition of the leg length from the beginning of the single support phase to the end:

$$\lambda_{kst} = \frac{1}{2}\theta_3 \left( \cos\left(\frac{t - t_{ssp}}{\theta_5}\pi\right) + 1 \right) \quad (5.1)$$

where  $t$  with  $0 \leq t - t_{ssp} \leq \theta_5$  is the current time point, and  $t_{ssp}$  is the time that the single support phase of the current step starts.

Note that we do not implement the leg-length policy  $\delta(\alpha)$  by measuring the angle  $\alpha$  of the virtual telescopic leg with the vertical axis and deriving the knee-angle from the leg-length  $l\delta(\alpha)$ . Instead, we use a time interval  $\theta_5$  to control the stretching of the leg from an initial angle  $\theta_3$  to a fully stretch leg. Obviously, the time interval will depend on the walking speed and therefore, needs to be optimized for a given speed; see Section 5.4.

#### FORCE POLICY

The force policy is applied during the double support phase of a step. This phase starts at the heel strike of the swing foot. Although the Nao robot has sensors in the feet that can be used to identify the heel strike, we use a timer instead. The timer determines when the heel strike must occur, and therefore when the double support must start. The time at which the double support phase starts is determined by the parameter  $\theta_9$ , which denotes the proportion of the total step time specified by the parameter  $\theta_4$ . So, the double support phase starts  $\theta_4\theta_9$  seconds after the start of the single support phase. Both  $\theta_4$  and  $\theta_9$  are parameters that need to be optimized; see Section 5.4. The advantage of this approach is that we can search for the optimal proportion between the single and the double support phase.

The force policy of the swing leg during the double support phase is implemented by setting the target angle and the stiffness of the knee joint as a function of the angle  $\beta'$  between the swing leg and the vertical axis in the frontal plane. The angle  $\beta'$  changes

from  $\beta'_b$  to 0 during the double support phase. The Nao robot has a build-in function `ALMotionProxy::getPosition` that can give the position and angle of joints w.r.t. the `FRAME_TORSO` of which the origin lays near the CoM. In our implementation of the gait, we use this function to compute  $\beta'$ . The initial angle  $\beta'_b$  can be determined by computing  $\beta'$  at the start of the double support phase.

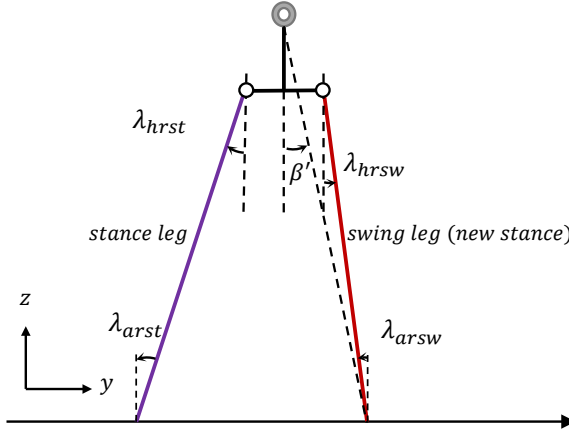


Figure 5.5: The definition of key angles of a biped model in the frontal plane

On a Nao robot, the target angle and stiffness of a joint can be adapted every 20 ms. We set, in every cycle of 20 ms, the target angle of the knee joint equal to the currently observed angle. This means that we try to lock the knee joint in its current position. At the same time we set the stiffness using the Quadratic Bezier Curve shown in Figure 4.9. When  $\beta'$  approximates the value 0, the stiffness approximates the value 1. The parameter  $\theta_7$  determines the shape of the Bezier curve.

#### FOOT TRAJECTORY OF THE SWING FOOT IN SSP

During the single support phase, the foot of the swing leg has to move from its old position to a new position. The time available for the move is  $\theta_4\theta_9$  seconds. The other parameters that are relevant, are the step length  $\theta_1$  and the step height  $\theta_2$ . Based on these parameters, a foot trajectory, shown in Figure 5.6, has to be generated for the swing foot. We compute this trajectory by specifying the  $x$  and  $z$  position of the swing foot in a local coordinate system  $O_{st}XY$ , where the position of the stance foot is the origin  $O_{st}(0,0)$  (see Figure 5.4), the  $x$ -axis points in the walking direction, the  $z$ -axis points upward. We use a sinusoid to generate a smooth trajectory.

$$x_f(t) = -\frac{\theta_1}{2} \cos\left(\frac{t-t_{ssp}}{\theta_4\theta_9}\pi\right) \quad (5.2)$$

$$z_f(t) = \frac{\theta_2}{2} \left[1 - \cos\left(\frac{t-t_{ssp}}{\theta_4\theta_9}2\pi\right)\right] \quad (5.3)$$

where  $t$  with  $0 \leq t - t_{ssp} \leq \theta_4\theta_9$  is the current time point, and  $t_{ssp}$  is the time that the single support phase of the current step starts.

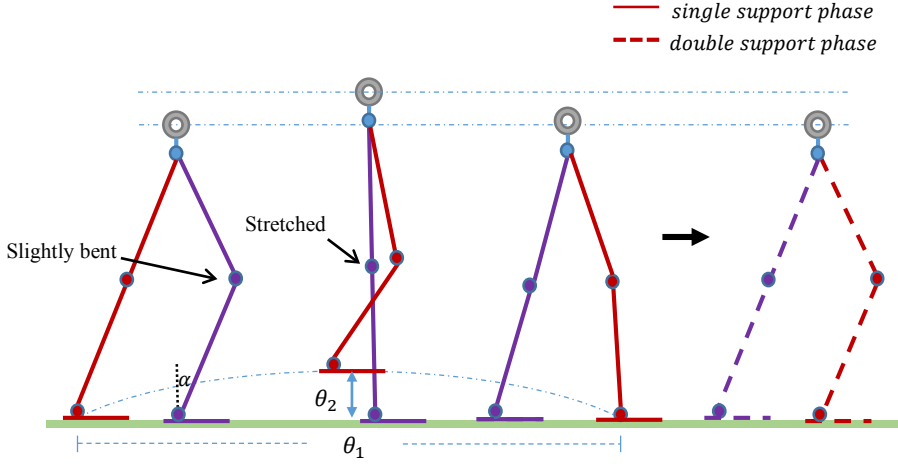


Figure 5.6: The sagittal motion of the resulting walking gait with telescopic legs

To compute the angles of the swing leg generating this trajectory, we first compute the location of the hip joint in the local coordinate system.

$$x_h(t) = l_1 \sin(\lambda_{\text{apst}}(t)) + l_2 \sin(\lambda_{\text{apst}}(t) - \lambda_{\text{kst}}(t)) \quad (5.4)$$

$$z_h(t) = l_1 \cos(\lambda_{\text{apst}}(t)) + l_2 \cos(\lambda_{\text{apst}}(t) - \lambda_{\text{kst}}(t)) \quad (5.5)$$

where  $\lambda_{\text{apst}}$  is the ankle pitch of the stance leg and  $\lambda_{\text{kst}}$  is the angle of the knee joint. Next we compute the length  $l'$  and angle  $\alpha'$  of the virtual swing leg; see Figure 5.4.

$$l'(t) = \sqrt{(x_h(t) - x_f(t))^2 + (z_h(t) - z_f(t))^2} \quad (5.6)$$

$$\alpha'(t) = \arctan\left(\frac{x_h(t) - x_f(t)}{z_h(t) - z_f(t)}\right) \quad (5.7)$$

Finally, we compute the angles of the hip and knee joint of the swing leg.

$$\lambda_{\text{ksw}}(t) = \pi - \arccos\left(\frac{l_1^2 + l_2^2 - (l'(t))^2}{2l_1l_2}\right) \quad (5.8)$$

$$\lambda_{\text{hpsw}}(t) = \alpha'(t) - \arccos\left(\frac{(l'(t))^2 + l_2^2 - l_1^2}{2l'(t)l_2}\right) \quad (5.9)$$

The swing foot is kept parallel to the ground surface during the single support phase. The appropriate ankle angle can be computed by the summation of the joint angles on the kinematic chain from the vertical body to the swing foot.

#### ANKLE CONTROL OF THE STANCE LEG

The gait that we developed in Chapter 4, does not require a support polygon. The feet are reduced to points around which the robot can rotate freely. The ability to rotate freely

around the contact point of the telescopic leg with the ground is an important feature of this gait. On a real robot such as a Nao, the contact point with the ground corresponds with the ankle joint. At the ankle joint the robot can rotate in any direction. The ability to rotate *freely* around the ankle joint can be implemented in two ways. We can simulate free rotation around the ankle joint by computing the angles that correspond with a free rotation, and use these angles together with full stiffness of the ankle joint. An advantage is that it gives a support polygon, which improves the stability of the gait. A disadvantage is the energy consumption of the stiff ankle joints. The second option is setting the ankle stiffness to 0. An advantage of this option is the reduction of the energy consumption in the ankle joints, and the ability to adapt to small bumps in the walking surface. A possible disadvantage is the stability of the gait. Of the two options we chose the second because of the advantages, and because we wished to check whether a stable gait is possible without a support polygon.

Setting the stiffness of the ankle joint to 0 changes the joint into an underactuated joint. As a consequence, the robot is mainly supported by a small area under the ankle joint. Experiments with the Nao robot showed that setting the stiffness in the sagittal direction to 0 is not possible. With 0 stiffness the Nao robot starts walking on its heels and cannot propel itself because of lack of grip. Further experiments showed that setting the stiffness of the ankle pitch to 0.1 ensures that the foot of the stance leg and the foot of the swing leg during the double support phase of the Nao robot are flat on the ground and provide sufficient grip. The ankle pitch stiffness of 0.1 and ankle roll stiffness of 0 ensure that the ankle joints can still rotate freely in any direction and are therefore essentially underactuated joints.

5

### 5.2.3. BODY TRAJECTORY GENERATOR

In the previous section, we assumed that the body of the Nao is in a vertical position. We need a controller that assures that this assumption holds. This controller determines the angles of the hip pitch and roll joints of the stance leg.

Walking on flat ground, it is easy to determine the proper angles. Since we can use the ground surface as a reference, we can determine the proper angles of the hip pitch and roll joints of the stance leg using the angles of the ankle and knee joints.

$$\lambda_{\text{hpst}} = -\lambda_{\text{apst}} - \lambda_{\text{kst}} \quad (5.10)$$

$$\lambda_{\text{hrst}} = -\lambda_{\text{arst}} \quad (5.11)$$

where  $\lambda_{\text{apst}}$  is the ankle pitch angle of the stance leg,  $\lambda_{\text{kst}}$  is the knee angle of the stance leg, and  $\lambda_{\text{arst}}$  is the ankle roll angle of the stance leg.

The body of a robot is not always in a vertical position. As we saw in Chapter 4, at the start of every double support phase, the Nao robot must lean slightly inward in order to disturb its balance. Moreover, when the robot starts walking, it must lean slightly forward in order to generate a forward motion of its body. Without first generating a forward motion of the body, the robot will fall. Note that the ankle stiffness is set to 0.1 and therefore the forward leaning will enable gravity to cause a forward motion. The forward leaning is determined by the Torso Pitch Inclination parameter  $\theta_6$ , and has a duration of 200 ms. The inward leaning is determined by the Torso Roll Inclination parameter  $\theta_8$



and has a duration of 100 ms. During the forward and inward leaning, the angles of the hip joint are adapted (see Figure 5.7):

$$\lambda_{\text{hpst}}^* = \lambda_{\text{hpst}} + \theta_6 \quad (5.12)$$

$$\lambda_{\text{hrst}}^* = \lambda_{\text{hrst}} - \theta_8 \quad (5.13)$$

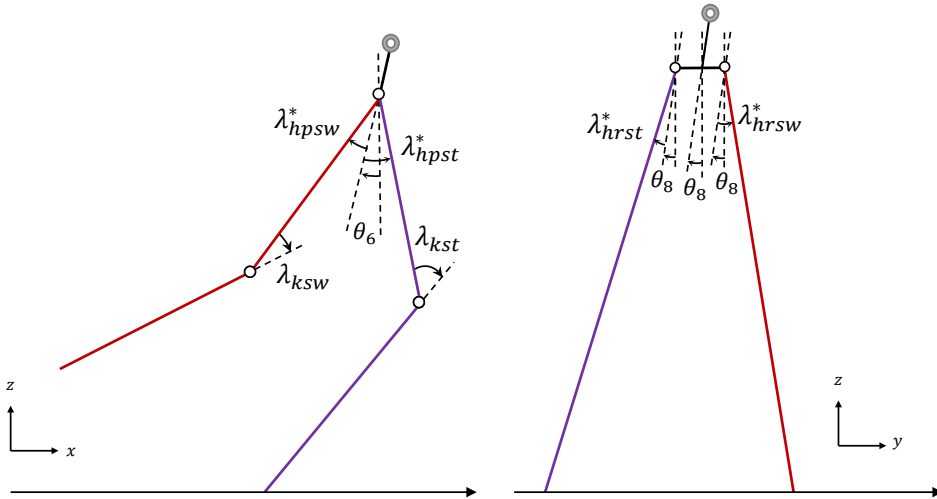


Figure 5.7: The definition of the joint angles when the torso inclines

### 5.3. CONTROLLER PARAMETERS AND FITNESS FUNCTION

This section presents the parameters of a gait controller that realizes the leg-length policy described in Section 4 and the lateral movement of the CoM in the DPS. We identified 9 parameters that are essential in controlling a dynamic gait:

- *Step Length* ( $\theta_1$ ): Defines the distance over which the Nao moves in a single step (sagittal).
- *Step Height* ( $\theta_2$ ): Defines the maximal altitude between the ground and a lifted foot. A high step height requires a faster movement of the swing leg, which may cause instabilities. A low step height increases the possibility of tripping and limits the step length.
- *Knee Bending* ( $\theta_3$ ): Defines the maximum bending of the swing leg at the beginning of the double support phase which determines the value of  $\delta(\alpha_b)$ , see Figure 4.4. This parameter determines the sagittal velocity and the energy cost.
- *Step Time* ( $\theta_4$ ): Defines how long a single step lasts. This parameter determines the sagittal walking velocity.

- *Stretch Time* ( $\theta_5$ ): Defines how long it takes for the stance leg to stretch from  $\theta_3$  (angle of bent knee) to its full length at the beginning of the single support phase, see Figure 4.4.
- *Torso Pitch Inclination* ( $\theta_6$ ): Defines the maximum angle that the torso leans in the sagittal direction at the beginning of the first step. If positive, it will move the center of mass (CoM) in the sagittal direction. If it is not set appropriately, a fall will occur. In our experiments, the inclination lasts for 200 ms.
- *Quadratic Bezier point* ( $\theta_7$ ): Defines the magnitude of the middle points in Quadratic Bezier Curves, see Figure 4.9, which determines the force policy of the swing leg (introduced in Section 4).
- *Torso Roll Inclination* ( $\theta_8$ ): Defines the maximum angle that the torso leans in the lateral direction. If positive, it will move the center of mass (CoM) towards the swing leg in the frontal plane as discussed in Section 4.
- *Proportion of single support duration* ( $\theta_9$ ): Defines how long the single support phase lasts in one single step. The single support phase duration equals the product of this parameter and the step time  $\theta_4$ .

All parameters except  $\theta_1$  (the step length) will be optimized in the experiments. We do not consider the optimization of the step length, because we determine the parameters for a fixed walking velocity. We manually set a different walking velocity  $v$  in each experiment and determined the optimal *Step Time*  $\theta_4$ . The corresponding step length is given by:  $\theta_1 = v\theta_4$ .

To generate a walking gait that is energy efficient and stable, we considered a fitness function that concerns the total energy cost and the stability over a certain distance of forward walk. The fitness function is inversely proportional to energy cost which is expressed by

$$C = \frac{E}{mgD} \quad (5.14)$$

where  $C$  is the *cost of transport* [127, 156],  $E$  is the total amount of energy consumed completing the forward walk,  $m$  is the mass of the whole robot and  $D$  is the walking distance. The energy is measured using the voltage and current readings from the on-board battery and motors. The value of  $C$  is then used to calculate the fitness  $\mathcal{F}_c$  using:

$$\mathcal{F}_c = \frac{\frac{43.2t}{mgD}}{C+1} = \frac{43.2}{(C+1)mgv} \quad (5.15)$$

where  $v$  is the average forward velocity for the robot to finish the distance  $D$ , 43.2 is the maximum power consumption of the robot's joints. This fitness function was chosen to make  $\mathcal{F}_c$  maximal when the efficiency is the highest.  $\mathcal{F}_s$  is the fitness function evaluating the motion stability.

$$\mathcal{F}_s = \frac{\text{distance}}{D} \quad (5.16)$$

where *distance* is the travelled distance before the robot is falling,  $D$  is the whole walking distance that robot needs to complete.

For the optimization of energy efficiency and the stability of a gait, the following general fitness function was selected:

$$\mathcal{F} = \lambda \cdot \mathcal{F}_c + (1 - \lambda) \mathcal{F}_s \quad (5.17)$$

The parameter  $\lambda$  of this fitness function is a scaling parameter. In our experiments, we gave the stability of the gait priority over the gait's energy cost. Through the experiments, we have tested different weight of the stability  $\lambda$  and confirm  $\lambda = 0.3$  is a proper value in our case. We use  $\mathcal{F}(\boldsymbol{\theta})$  to denote the fitness of the gait determined by the vector  $\boldsymbol{\theta}$  of control parameter values.

## 5.4. ALGORITHM FOR LEARNING CONTROLLER PARAMETERS

We use a form of policy gradient reinforcement learning to automatically search the set of possible parameters with the goal of finding a stable and low energy cost walk. Assuming that the policy is differentiable with respect to each of the parameters, we can estimate the policy's gradient in parameter space, and then follow it towards a local optimum. But since we do not know anything about the true functional form of the policy, we cannot calculate the gradient exactly in symbolic form. Therefore, after investigating the policy search algorithms proposed by Saidouni et al. [132], Chernova et al. [18], Tedrake et al. [153] and Deisenroth et al. [31], we chose to use a Policy Gradient Reinforcement Learning (PGRL) method presented by Kohl et al. [85] to optimize the gait parameters. Though Peters et al. [124] concluded this PGRL method has a theoretically slower convergence rate among other PGRL methods, this PGRL method was the only applicable method among those we investigated because we cannot calculate the gradient exactly.

Kohl's method [85] is among the finite-difference methods. In PGRL, each  $N$  dimensional parameter vector  $\boldsymbol{\theta}$  is considered as an open loop policy that can be executed by the robot. The algorithm randomly generates  $B$  test policies  $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^B$  around a policy  $\boldsymbol{\theta}$ , which is to be improved. The parameter  $\theta_j^i$  of the test policy  $\boldsymbol{\theta}^i$  is set randomly to either  $\theta_j - \epsilon_j$ , to  $\theta_j$ , or to  $\theta_j + \epsilon_j$ , for all  $1 \leq i \leq B$  and  $1 \leq j \leq N$ , where  $N$  is the number of parameters. The disturbance  $\epsilon_j$  is a small constant value. Each test policy  $\boldsymbol{\theta}^i$  evaluated and the corresponding value of the fitness function  $\mathcal{F}(\boldsymbol{\theta}^i)$  (Equation 5.17) is determined. The test policies  $\boldsymbol{\theta}^i$  are grouped into three categories for each parameter  $1 \leq j \leq N$ :  $S_j^-$ ,  $S_j^0$  and  $S_j^+$ , depending on whether the  $j$ th parameter of  $\boldsymbol{\theta}^i$  is modified by  $-\epsilon$ , 0 or  $+\epsilon$ . For each  $S_j^x$  with  $x \in \{-, 0, +\}$ , we calculate the average score of the fitness values:  $Avg_j^-$ ,  $Avg_j^0$  and  $Avg_j^+$  where

$$Avg_j^x = \frac{i}{|S_j^x|} \sum_{\boldsymbol{\theta}^i \in S_j^x} \mathcal{F}(\boldsymbol{\theta}^i) \quad (5.18)$$

These three average values estimate the benefit of altering the  $j$ th parameter by  $+\epsilon_j$ , 0,

$-\epsilon_j$ . An adjustment vector  $\mathbf{A}$  of size  $N$  is calculated where

$$A_j = \begin{cases} 0, & \text{if } Avg_j^0 \geq Avg_j^+ \text{ and} \\ & Avg_j^0 \geq Avg_j^- \\ Avg_j^+ - Avg_j^-, & \text{otherwise} \end{cases}$$

If  $\mathbf{A} = \mathbf{0}$ , a possible local optimum is encountered. In all other cases, the adjustment  $\mathbf{A}$  is normalized to get a direction vector of length 1. Next, the policy  $\boldsymbol{\theta}$  is adapted using the normalized direction vector. Since each parameter  $\theta_j$  in  $\boldsymbol{\theta}$  may have a different range, and since of de choice of  $\epsilon_j$  will depend on the range, the normalized direction for parameter  $j$  is multiplied by  $\epsilon_j$  and by a scaling factor  $\eta$ .  $\eta$  determines the accuracy and the search speed of algorithm.

$$\theta_j \leftarrow \theta_j + \epsilon_j \frac{A_j}{|\mathbf{A}|} \eta \quad (5.19)$$

PGRL continues the search using the adjusted policy  $\boldsymbol{\theta}$  in the next iteration.

---

**Algorithm 1** Policy Gradient Reinforcement Learning algorithm

---

**Input:**  $\boldsymbol{\theta}$  (initial policy),  $\boldsymbol{\epsilon}$  (disturbance vector),  $\eta$  (scaling factor)

**Output:**  $\boldsymbol{\theta}^*$  (optimal policy)

```

1: while  $\mathbf{A} \neq \mathbf{0}$  do
2:   generate  $B$  random perturbations of  $\boldsymbol{\theta}$ :  $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^B$ 
3:   run experiments to determine the fitness values:  $\mathcal{F}(\boldsymbol{\theta}^1), \mathcal{F}(\boldsymbol{\theta}^2), \dots, \mathcal{F}(\boldsymbol{\theta}^B)$ 
4:   for  $j = 1$  to  $N$  do
5:      $S_j^+ = \{\boldsymbol{\theta}^i \mid \theta_j^i = \theta_j + \epsilon_j\}$ 
6:      $Avg_j^+ \leftarrow \frac{i}{|S_j^+|} \sum_{\boldsymbol{\theta}^i \in S_j^+} \mathcal{F}(\boldsymbol{\theta}^i)$ 
7:      $S_j^0 = \{\boldsymbol{\theta}^i \mid \theta_j^i = \theta_j + 0\}$ 
8:      $Avg_j^0 \leftarrow \frac{i}{|S_j^0|} \sum_{\boldsymbol{\theta}^i \in S_j^0} \mathcal{F}(\boldsymbol{\theta}^i)$ 
9:      $S_j^- = \{\boldsymbol{\theta}^i \mid \theta_j^i = \theta_j - \epsilon_j\}$ 
10:     $Avg_j^- \leftarrow \frac{i}{|S_j^-|} \sum_{\boldsymbol{\theta}^i \in S_j^-} \mathcal{F}(\boldsymbol{\theta}^i)$ 
11:    if  $Avg_j^0 \geq Avg_j^+$  and  $Avg_j^0 \geq Avg_j^-$  then
12:       $A_j \leftarrow 0$ 
13:    else
14:       $A_j \leftarrow Avg_j^+ - Avg_j^-$ 
15:    end if
16:     $\theta_j^* \leftarrow \theta_j + \epsilon_j \frac{A_j}{|\mathbf{A}|} \eta$ 
17:  end for
18: end while
19: return  $\boldsymbol{\theta}^*$ 

```

---

## 5.5. LEARNING OPTIMAL PARAMETERS IN THE WEBOTS SIMULATOR

To identify the optimal gait parameters and validate the gait's performance, we uploaded the controller of our proposed gait together with an implementation of the policy gradient algorithm into the Webots simulator. We used a relatively elementary hand-tuned gait as a starting policy for the policy gradient algorithm. Though a bad starting policy may lead to a simulation failure or a local optimum, we did not deliberately An overview of the learning loop is shown in Figure 5.8. The loop models the gait controller and learning process. At each time step, individual joint angles are computed by the motion generator, which structure is organized into two motion states using a finite state machine. The right part of the loop defines the learning process, which outputs the learned control policy. Each policy  $\theta$  represents a group of control parameters. PGRL algorithm generates  $B$  test policies  $\theta^i$  around  $\theta$ . Each new policy was evaluated by letting a robot walk at a constant distance of 75 cm while measuring the energy consumption and the walking distance without falling. These values are used by the fitness function  $\mathcal{F}$  to evaluate the quality of each test policy  $\theta^i$ . After evaluating all  $B$  test policies, the policy  $\theta$  is updated using Equation 5.18. We set the number of generated test policies  $B$  equal to 10, and set scaling factor  $\eta$  equal to 3. The used  $\epsilon$  values are given in Table 5.1.

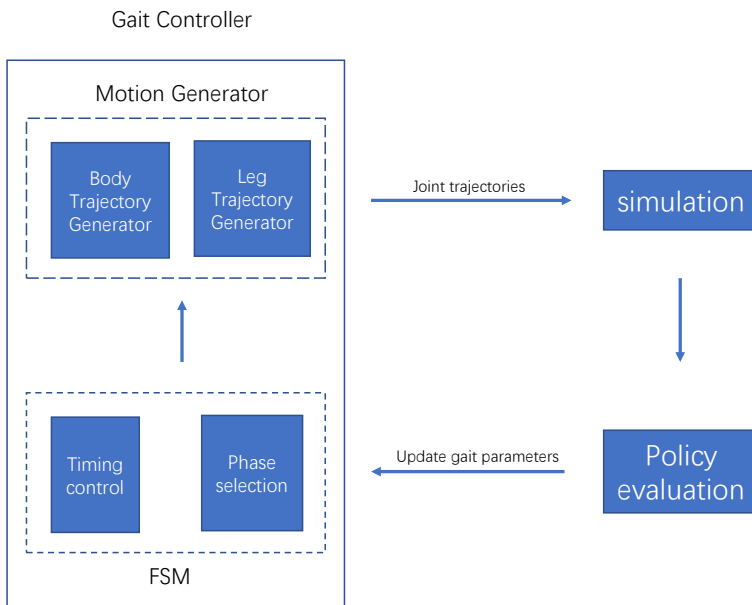


Figure 5.8: The overview of the policy gradient reinforcement learning process

Note that the electric current of joint motors cannot be directly obtained in the simulator. Since the electric current of a motor is usually considered approximately proportional to the torque produced by this motor, we retrieved the force feedback to get an approximation of the current consumed by the motor.

### CONVERGENCE AND LEARNING TIME

The policy gradient algorithm converged to the parameter set  $P$  shown in Table 5.1. The algorithm needed 1200 iterations to converge. Each iteration required on average 10 seconds of computation time on a standard PC with CPU: Intel Core i5-2400 3.10GHz, and operating system: Windows 7 Enterprise. The computation time includes the evaluation of each of the 1200 iterations in the *Webots* simulator, in which the Nao robot needed to walk for 75 cm in real time (no accelerations of the simulation).

Table 5.1: Learned Parameters set  $P$

Symbol	Parameter	$\epsilon$	Learned Value
$\theta_1$	step length	0.1	3.6(cm)
$\theta_2$	step height	0.02	3.24(cm)
$\theta_3$	knee bending	0.1	14.2(°)
$\theta_4$	step time	25	600(ms)
$\theta_5$	stretch time	6	78(ms)
$\theta_6$	torso pitch inclination	0.1	8.9 (°)
$\theta_7$	Quadratic Bezier point	0.1	(0.9*DSP time, 0.2)
$\theta_8$	torso roll inclination	0.1	6.5 (°)
$\theta_9$	proportion	0.1	0.8
v	velocity		6 (cm/s)

The algorithm presented here will converge to a local optimum. Therefore, the set of parameters  $P$  shown in Table 5.1 need not be the global optimal choice of parameters. In order to investigate whether the results could be a global optimum, we repeated the learning experiment 500 times, each time starting from a randomly generated parameter vector  $x^T$  with the same velocity. Figure 5.9 shows the convergence curves of the parameter  $\theta_3$  (knee bending) in five experiments. The other parameters show similar convergence results. In every experiment the parameters converge to the same optimal values. The results of the experiments indicate that the local optimum we have in Table 5.1 is likely the global optimum. Therefore, the parameter set  $P$  most likely results in the globally most energy efficient gait.

The total time needed to repeat the learning experiment 500 times in order to verify whether we identified a global optimum, was 3 weeks of computation time.

### ROBUSTNESS OF THE OPTIMAL PARAMETERS VALUES

Optimization of a set of parameters under a set of constraints may result in a solution that is not robust under small changes. For instance, the optimal energy efficiency may result in a gait that is on the boundary of being stable. A small change in the parameter values can make the gait unstable. Such an optimal set of parameter values does result in gait that is robust in real world applications.

To investigate the robustness of the identified optimal parameter values, we investigated how much each parameter value can vary without making the gait unstable. The method we used to identify the boundaries of each parameter value is by adding a value  $k\epsilon$  with  $k \in \mathbb{Z}$  to a parameter while keeping the other parameters fixed at their optimal values. In this way we determined the boundary of the interval that guarantees a stable

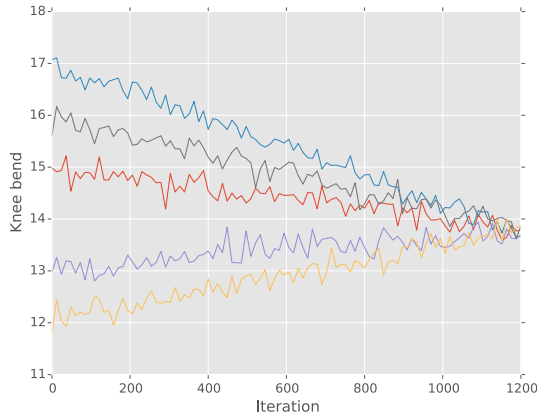


Figure 5.9: The convergence curves of gait parameter *knee bending*

gait. The Table 5.2 shows the epsilon value that we used for each parameter as well as the upper and lower bound of each parameter value.

When we compare the optimal parameter values listed in Table 5.1 with the boundaries listed in Table 5.2, we see that only two parameters are critical, namely  $\theta_7$ , the Quadratic Bezier point, and  $\theta_9$ , the proportion of the single support duration. The latter parameter value  $\theta_9$  is determined by the walking speed of the Nao robot and the time to fall towards the new stance leg in the double support phase. So, this parameter is determined by the movement of the robot, and not one that we can actually control. The same holds for the former parameter value  $\theta_7$ , which is determined by the requirement that the lateral movement is stopped when the CoM is above the new stance feet.

Table 5.2: The boundary of the parameter space for each control parameter of the walking gait at a speed of 6 cm/s.

Symbol	Parameter	$\epsilon$	Lower bound	Upper bound
$\theta_1$	step length	0.1	0.0(cm)	4.5(cm)
$\theta_2$	step height	0.02	0.18(cm)	4.24(cm)
$\theta_3$	knee bending	0.1	12.6(°)	14.8(°)
$\theta_4$	step time	25	375(ms)	725(ms)
$\theta_5$	stretch time	6	24(ms)	96(ms)
$\theta_6$	torso pitch inclination	0.1	2.3 (°)	10.6 (°)
$\theta_7$	Quadratic Bezier point	0.1	(0.9*DSP time, 0.2)	(0.9*DSP time, 0.2)
$\theta_8$	torso roll inclination	0.1	4.6 (°)	9.5 (°)
$\theta_9$	proportion	0.1	0.8	0.8

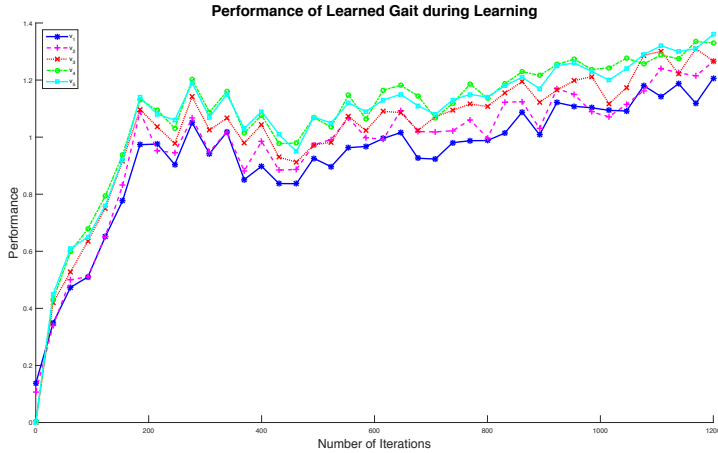


Figure 5.10: The performance of the walking gait during learning iterations

## 5

### PARAMETERS AND ENERGY CONSUMPTION OF DIFFERENT WALKING SPEEDS

We carried out walking experiments for nine different velocities  $v_i$  between 2 cm/s and 10 cm/s. Each experiment generates an optimized set of parameter values, which determines a walking gait. We kept the parameter  $\theta_2$  constant in the experiments to ensure it does not influence the energy consumption of the gait at different walking speeds. Table 5.3 specifies the optimal parameter values identified by the learning algorithm for each of the nine walking speeds. Figure 5.10 shows the learning behavior for the walking speeds of 2, 3, 4, 5 and 6 cm/s.

Table 5.3: The optimized parameters of gaits in different velocities

Velocity	$\theta_1(cm)$	$\theta_2(cm)$	$\theta_3$	$\theta_4(ms)$	$\theta_5(ms)$	$\theta_6$	$\theta_7$	$\theta_8$	$\theta_9$
2 cm/s	2.3	3.24	$6.2^\circ$	1150	53	$4.3^\circ$	(0.9,0.2)	$6.8^\circ$	0.7
3 cm/s	2.9	3.24	$8.7^\circ$	980	59	$5.6^\circ$	(0.9,0.2)	$6.8^\circ$	0.7
4 cm/s	3.2	3.24	$11.5^\circ$	800	67	$6.1^\circ$	(0.9,0.2)	$6.8^\circ$	0.7
5 cm/s	3.4	3.24	$12.3^\circ$	680	71	$7.5^\circ$	(0.9,0.2)	$6.8^\circ$	0.8
6 cm/s	3.6	3.24	$14.2^\circ$	600	78	$8.9^\circ$	(0.9,0.2)	$6.5^\circ$	0.8
7 cm/s	3.7	3.24	$15.6^\circ$	530	81	$9.2^\circ$	(0.9,0.2)	$6.5^\circ$	0.8
8 cm/s	3.8	3.24	$16.7^\circ$	480	83	$9.4^\circ$	(0.9,0.2)	$6.3^\circ$	0.9
9 cm/s	3.8	3.24	$16.7^\circ$	430	83	$9.6^\circ$	(0.9,0.2)	$6.3^\circ$	0.9
10 cm/s	3.8	3.24	$16.7^\circ$	380	83	$9.6^\circ$	(0.9,0.2)	$6.3^\circ$	0.9

We computed the energy consumption of each gait. Since the Webots simulator cannot provide the energy consumption of the joint, we estimate the energy consumption by assuming that the energy consumption is proportional to torques of the joints. Moreover, since the individual results of the estimated energy consumption are not accurate enough, we repeated the experiments five times to obtain the average energy cost under the same walking velocity. The results are shown Figure 5.11. The values of the energy



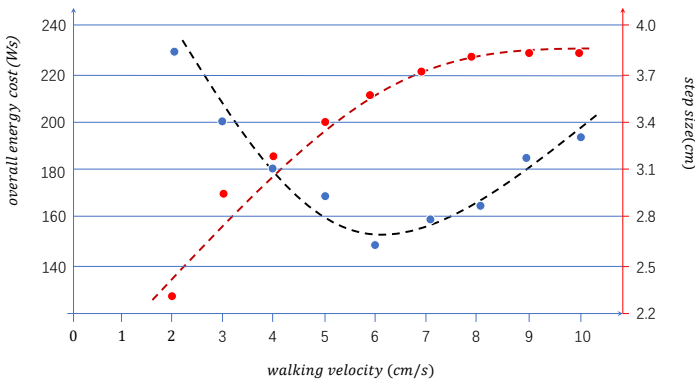


Figure 5.11: The estimated energy consumption (the blue dots) for walking a distance of 75 cm, and the corresponding optimal step length (the red dots).

consumption shown on the lefthand side of the figure are calibrated using the energy consumption of a real Nao robot while walking at 6 cm/s (see the next section).

Figure 5.11 shows the total energy needed to walk a distance of 75 cm as well as the corresponding optimal step size of the gait. The figure shows that the most energy efficient walking speed for traversing a specified distance is 6 cm/s. The accompanying video material<sup>1</sup> shows the Nao robot walking on flat ground with our proposed gait controller at a speed of 6 cm/s.

If we compute the power consumption for the different walking speeds, we see that walking with a speed of 2 cm/s reduces the power consumption with 49% compared to walking with a speed of 6 cm/s (6.1 W for 2 cm/s versus 11.9 W for 6 cm/s). However, the total energy consumption for walking a distance of 75 cm increases with 54 %.

Since a Nao robot has a limited battery capacity, it is important to minimize the energy consumption needed to perform a task that requires walking. In that case, a walking speed of 6 cm/s is optimal. This corresponds to a step length of 3.6 cm. In some applications, such as robot soccer, speed is more important, provided the robot does not drain its battery before the end of the game. Figure 5.11 shows how a higher speed increases the energy consumption. A walking speed of 10 cm/s increases the overall energy consumption with 30 %.

## 5.6. EXPERIMENTAL VALIDATION

In this section, we present the real world experiments on an Aldebaran Nao robot, which address the various aspects of the proposed gait on a real Nao shown in Figure 5.12.

### 5.6.1. GAIT ANALYSIS

The main aspects of the new gait that we analyzed, are the impact on the stability and the energy consumption of the robot. Other aspects of interest are the movement of the knee joint while implementing the leg-length policy, and the ground projection of the

<sup>1</sup><https://project.dke.maastrichtuniversity.nl/robotlab/wp-content/uploads/naowalk.mp4>

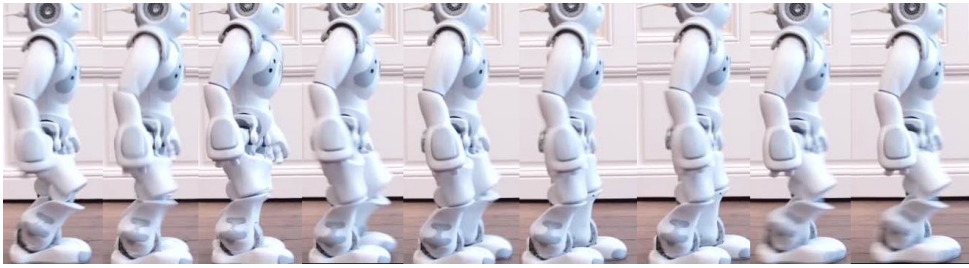


Figure 5.12: A sequence of photos of the walking biped Nao robot walk in our proposed gait

forces working on the Nao while walking.

### ACCELERATION FORCES

As introduced in Chapter 2, the Nao robot contains an IMU located at its chest that is composed of 3-axis accelerometers and 3-axis gyrometers. During the operation of the Nao, there are two different types of forces that can be measured by the accelerometers. The first one is the force of gravity, which has a high magnitude and a constant value pointing downwards. The second group consists of the inertial forces on the torso due to the locomotion movements of the robot. Their magnitudes depend on the smoothness of the locomotion.

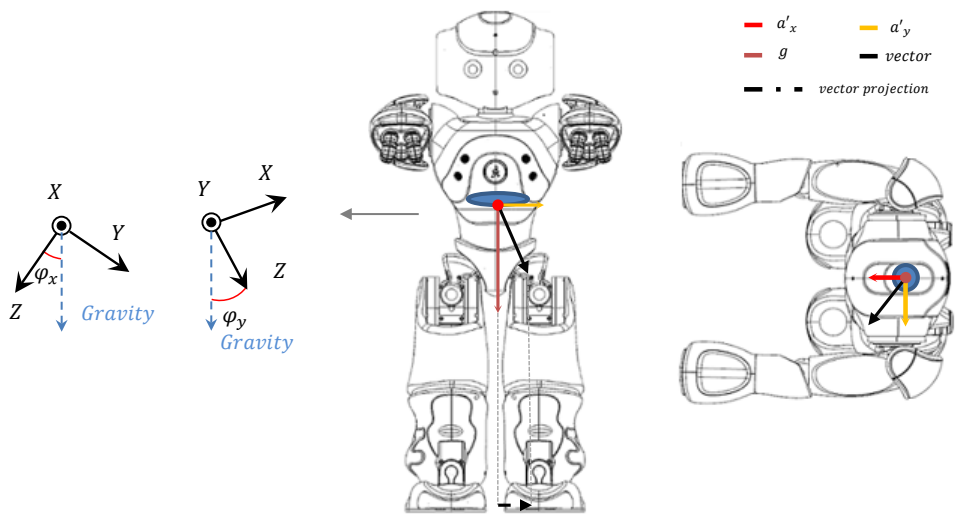


Figure 5.13: Aldebaran Nao: The robotic platform we employed in this thesis and its acceleration vector

Since gravity is commonly the main force that can be measured at the torso, the signals provided by the accelerometers and gyrometers can be employed to estimate torso orientation. With the torso orientation, we determine the acceleration  $a'_x$  along the orientation of  $x$  axis in the world coordinate system and the acceleration  $a'_y$  along the ori-

entation of  $y$  axis in the world coordinate system.

$$a'_x = a_z \sin \varphi_y + a_x \cos \varphi_y \quad (5.20)$$

$$a'_y = -a_z \sin \varphi_x + a_y \cos \varphi_x \quad (5.21)$$

where  $a_x$ ,  $a_y$  and  $a_z$  are the accelerometer measurements in the  $x$ ,  $y$  and  $z$  axis respectively.  $\varphi_x$  and  $\varphi_y$  are the estimated torso roll angle and pitch angle respectively, based on the accelerometer and gyroscope measurements. In the direction along  $z$  axis of the world coordinate system, the acceleration on the robot equals gravitational acceleration  $9.81 \text{ (m/s}^2\text{)}$ , which can not be changed by robot movement (see Figure 5.13). To describe the dynamics of the robot, the trajectory of the robot acceleration vector is evaluated in the axial plane, neglecting acceleration in the  $z$  axis of the world coordinate system. We employed this 2D trajectory to measure the torso oscillation, see Figure 5.14.

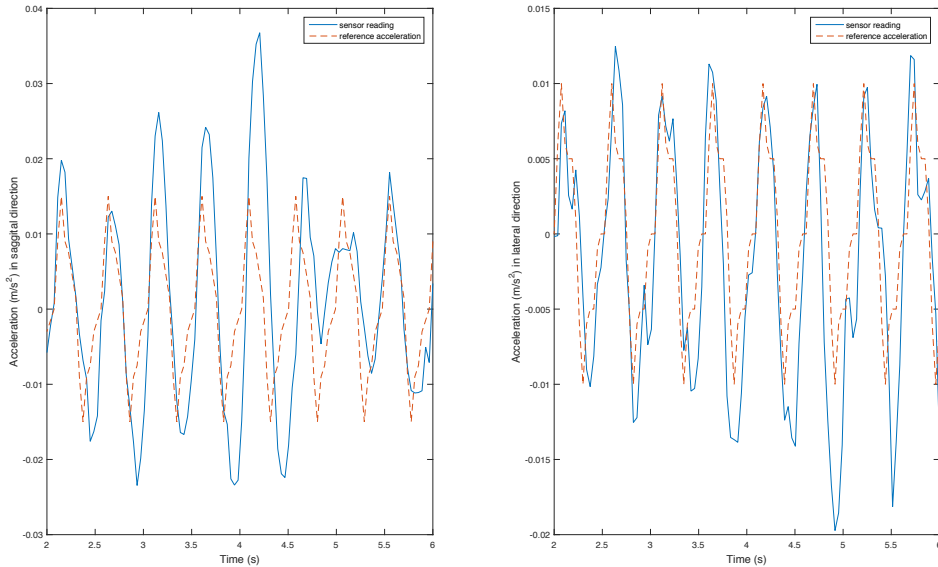


Figure 5.14: The comparison between the projected acceleration vector in the sagittal and frontal plane on a real robot and the reference projected acceleration vector in the simulator.

For comparison, we introduced a reference trajectory for the acceleration vector in Figure 5.14. We obtained this reference trajectory by simulating the robot's movement with identical controller parameters in *Webots*. In the simulator, we removed all the external factors which could potentially disturb the robot's stability and there is no noise in the data retrieved from the IMU.

We can observe in the Figure 5.14 that the accelerations in the sagittal and lateral direction are sometimes twice as high as the reference accelerations that we identified using the *Webots* simulator. Possible causes of these differences are the wooden plank floor on which the Nao was walking and the internal joint movements (the joints, motors

and gearboxes). Despite the higher accelerations that sometimes appear, the controller of the new gait can still generate a stable walking pattern.

### ENERGY CONSUMPTION

We evaluated the energy consumption of the hip, knee and ankle joints of the Nao robot by comparing it with the energy consumption of the standard gait which Aldebaran ships with the Nao robot. We determined the power consumption by measuring the currents drawn by the joints. Given a fixed voltage, the currents are linearly proportional to the supplied power. Figure 5.15 shows the average electric current of the joints for the two gaits.

For a step length of 3.6 cm, the current at the knee joint reduces by 32.7% (from 0.52 A to 0.35 A). This reduction is caused by the stance-leg being fully stretched during 88 % of a step. The knee joint of a stretched leg requires less power because the radial forces on the stance-leg result in none torque on the knee joint. The bent knee joint of the standard gait must always maintain a torque to handle the radial forces on the stance leg, and therefore, consumes more power.

5

The new gait does not require ankle stiffness for being dynamically stable while walking. This made it possible to reduce the ankle roll stiffness to 0 and the ankle pitch stiffness to 10% of full stiffness. Because of this stiffness reduction, the current of the ankle joint decreases by 83.3% (from 0.30 A to 0.05 A). In the standard gait the ankle stiffness is essential for maintaining the dynamic stability while walking. Reducing the ankle stiffness to 10 % of the full stiffness will lead to a fall of the Nao robot for the standard gait.

There is an increase of the current by 25% (from 0.12 A to 15 A) in the hip joints. This increase of power consumption is caused by the torso inclination. The hip actuators consume slightly more energy to execute the torso inclination.

The reduction in power consumption of the knee and ankle joints outweighs the increase in power consumption in the hip joints. The power consumption of the new gait is still 41% less than for the standard gait of the Nao provided by Aldebaran. The overall current of whole body of a Nao for each walk with step length 3.6 cm is shown in Figure 5.16.

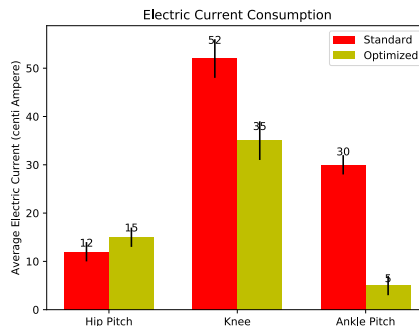


Figure 5.15: Comparison of the measured current for three joints (hip, knee, ankle).



Figure 5.16: The whole body current of a Nao after walking 0.75 meter with the step length 3.6 cm. The lines represent the unfiltered current values from two different walking gait. Note that the mean value of our proposed walk is clearly lower than that of the Aldebaran walking gait

### MOVEMENT OF THE KNEE JOINT

An essential aspect of the new gait proposed in this thesis is the leg-length policy. We investigated the realization of the leg-length policy by measuring the angle of the stance leg during a step. Figure 5.17 shows the realized joint angle during a step. We can clearly observe that the leg stretches in the beginning of the single support phase and remains stretched during the remainder of the single support phase and during the double support phase.

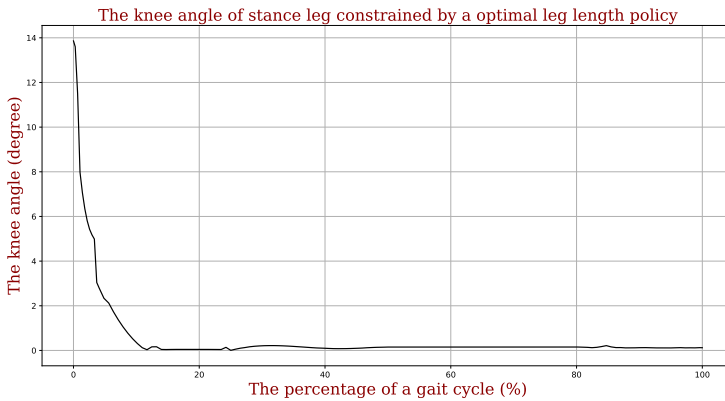


Figure 5.17: The knee angle for the stance leg implementing the leg-length policy

### GROUND PROJECTIONS OF FORCES

A criterion such as the ZMP requires that the point where the vector of all the forces working on the CoM crosses the walking surface, lies within the support polygon. If the point lies within the support polygon, it is called the Zero Moment point (ZMP) or the Center of Pressure (CoP). If it lies outside the support polygon, the ZMP is not defined and the robot is considered to be unstable according to the ZMP stability criterion, see Section 3.3. We are therefore interested in the vector projection of all the forces applied on the CoM.

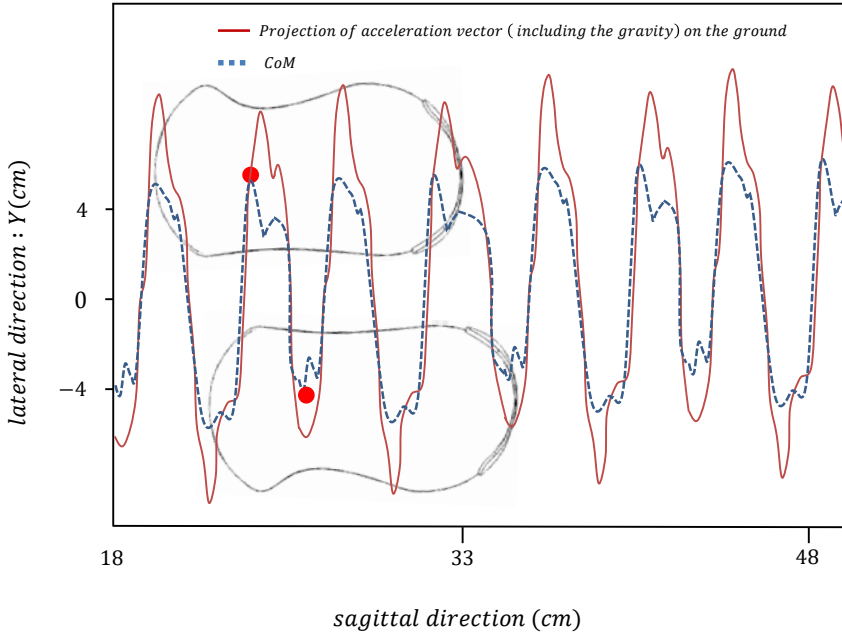


Figure 5.18: The trajectory of the point where the acceleration vector crosses the ground surface. The acceleration vector is retrieved from the IMU by sampling its values at a rate of 120 Hz. We average over 4 samples to smooth the curve. The red dots denote the locations to the ankle joints and represent the support points. Also the vertical ground projection of the CoM is shown. Note that the feet are shown at 10% of their actual sizes.

The IMU of the Nao robot enables us to measure the acceleration vector of the torso, and indirectly of the CoM. According to Newton's Laws of Motion, the direction of the acceleration of an object is the same as the direction of the applied force. Since we can measure the joint angles of the Nao, we can compute where the acceleration vector, and therefore the force vector, crosses the ground surface. We call this the ground projection of the acceleration vector. Figure 5.18 shows the movement of the projection of the acceleration vector on the ground surface. Because of the ankle pitch stiffness of 0.1 and the ankle roll stiffness of 0, the ankle joint is essentially underactuated. This implies that the weight of the robot is supported by only a small area under the ankle joint. The red dots in Figure 5.18 denote the centers of these small areas. To be stable according to the ZMP criterion, the projection of the acceleration vector should stay near a red dot except during the double support phase. Clearly, this is not the case for the proposed gait. This implies that there is an uncompensated torque working on the CoM of the robot, and therefore the robot starts to rotate around the underactuated ankle joint. This, however, is a feature of the proposed gait. The reason that the robot does not fall is because the swing foot will touch down before the robot falls in the sagittal direction and the force policy stops the fall in the lateral direction. Therefore the gait is dynamically stable.

## 5.7. CONCLUSION

In this thesis, we have investigated the possibility of creating a dynamically stable and energy efficient gait without an area of support. Here, the absence of an area of support means that the ankle joint is free to turn in any direction while the foot is on the ground. In the sagittal direction the robot is falling forward till the foot of the swing leg touches the ground. In the lateral direction, the robot balances above the stance foot in the single support phase, and falls towards the new stance foot in the double support phase. The falling towards the new stance foot is stopped by putting a force on the new stance leg.

In the previous chapter, we determined a new gait for a robot by analyzing how the length of the stance leg should change during a step in order to minimize the energy consumption. This analysis was carried out with an abstract model of a robot which needs to be fine-tuned for a real robot. In this chapter, we described a controller that realizes the energy efficient gait on a Nao robot. This controller has 9 parameters, which have been determined through reinforcement learning using the *Webots* simulator of the Nao robot. The resulting gait was subsequently evaluated on a real Nao robot.

Experiments with the new gait on the Nao robot have shown that no area of support is needed to create a stable gait. Moreover, an overall reduction of the energy consumption by 41% was realized with respect to the standard gait of the Nao robot provided by Aldebaran.

The absence of an area of support allows the robot to adapt its foot to an uneven floor. However, an uneven floor may influence the falling in the sagittal and lateral direction during a step and thereby the robot's stability. In the next chapter, we will extend the controller with a feedback loop in order to compensate for the effects of walking on an uneven terrain.

**Research Question** The results of this chapter provide us with answers to research questions 4 and 5.

### 4. *How to design the controller for gait generation ?*

The control of biped locomotion is difficult because biped robots are high-dimensional dynamical systems. We have developed a simple controller that can be used to generate various biped gaits. The controller has 9 control parameters that determine the generated gait. The parameters reflect the detail with which we model the gait. The parameters can be grouped into four categories: (a) sagittal motion (b) lateral motion (c) stability (d) time control.

The most important parameters for biped gait are Knee Bending ( $\theta_3$ ) and the Quadratic Bezier point ( $\theta_7$ ). The resulting leg-length policy and force policy mainly depend on these two parameters. Other parameters also make significant contributions to generate a robust balanced gait, such as the Torso Pitch Inclination ( $\theta_6$ ) which helps the torso to lean forward a bit, the Step Height ( $\theta_2$ ) and the Step Time ( $\theta_4$ ), which determine the lift of the swing foot upwards and forwards for a fixed distance in a fixed duration of time.

### 5. *How to optimize the energy efficiency for the walking gait on a flat floor ?*

Given the parametrized controller, the problem of finding a desired motion was reduced to assigning a set of values to the control parameters. A desired motion is defined through the choice of a fitness function: one that measures the travelled distance without falling or the total energy consumption when the maximum distance is reached without falling. We can search for the values of the control parameters that lead to an optimal value of the fitness function.

The optimal set of parameters was determined through policy gradient reinforcement learning. To ensure that the identified set of parameter values corresponds to a global optimum, the trial simulations were restarted with random initial parameter values. In all cases, the reinforcement learning algorithm converged to the same set of parameter values. The optimized gait was implemented and analysed on a real Nao robot to validate the results found in the Webots simulation environment.



# 6

## WALKING ON UNEVEN TERRAINS

### 6.1. INTRODUCTION

ACCOMPLISHING tasks in complex or unknown environments is one of the most relevant applications for a biped humanoid robot. Therefore, it is required to have robustness and terrain-adaptive properties in the walking controller. Though the method presented in previous chapters shows the effectiveness of using a combination of a leg-length policy, a force policy and a policy gradient reinforcement learning method to obtain an energy-efficient bipedal walk gait on level ground, that gait has its limitation. The stability of the gait can not be guaranteed when external disturbances occur. This means that any disturbance such as a push or stepping on uneven terrain may jeopardize its balance. For example, if the robot is standing on a slope or stepping on a bump, its feet on the ground are not on the same altitude, this may cause the CoM undershoot or overshoot the balance position when switching the stance leg. To enhance the walking stability especially on uneven terrain and with unknown disturbances, we adapt the controller introduced in Chapter 5.

### 6.2. UNEVEN TERRAINS VERSUS SLOPES

Since a step on uneven terrain causes a difference in altitude for the two feet of the biped robot, we assume that each step of the walk on an uneven ground can be viewed as a step on a (virtual) slope (see Figure 6.1). The angle of the slope is determined by the difference in altitude of both feet when they are firmly placed on the ground. In the sagittal

---

This chapter is an updated and expanded version of the following publication:

1. **Z. Sun**, and N. Roos: A Controller for Improving Lateral Stability in a Dynamically Stable Gait," in *AIAA: 12th IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2016. pp. 371-383. Springer.
2. **Z. Sun**, and N. Roos: "Dynamically stable walk control of biped humanoid on uneven and inclined terrain," *Neurocomputing*, 2018, vol. 280, pp. 111-122. Elsevier.

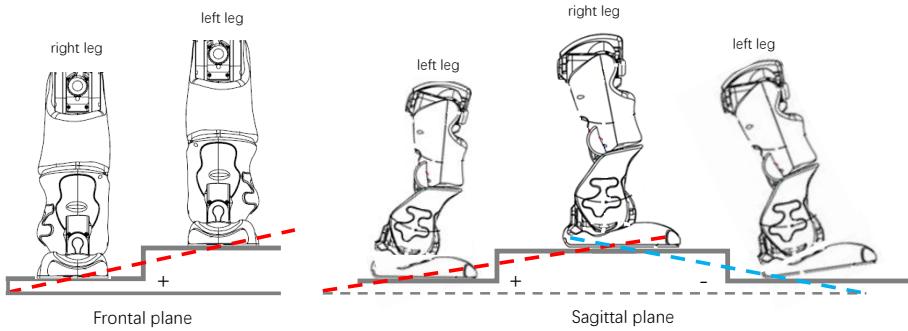


Figure 6.1: The illustration of the assumption: a complex terrain can be modeled as the discrete slopes.

direction, this corresponds to walking up or down a flat slope. In the lateral direction, this corresponds to walking perpendicular to the slope direction at a constant height. We define the slope angle to be positive if left foot is landed higher than the right foot in the frontal plane, or the new support foot is landed higher than the rear foot in the sagittal plane. Hence, the complex terrain with obstacles, including bumps, pits and slopes, can be modeled as slopes of variable angles.

To solve the balancing problem on uneven terrain, the gait controller should be adaptive to the various slopes by changing the control parameters. We will learn the optimal control parameters for different slope angles in the sagittal and lateral direction. In this way, we identify the optimal control parameters for specific differences in the height of the feet. The analysis of the identified control parameters will subsequently show that we can adapt the control parameters for the single and the double support phase independently.

## 6

### 6.3. ADAPTIVE CONTROLLER DESIGN

We designed a controller for walking on a flat horizontal ground surface in the previous chapter. We need to modify the control parameters of this controller when the robot is walking on a slope. Before we can use the controller for walking on a slope, we need to adapt one aspect. The controller described in Chapter 5 assumes that the body is normally in a vertical position. This position is guaranteed by the adapting the hip pitch and yaw joints using information about the joints on the kinematic chain from the body to the stance foot (Equations (5.10) and (5.11)). When walking on an uneven terrain, we can no longer assume the stance foot is in a horizontal position, and therefore, this solution is no longer applicable.

To ensure a vertical position of the body, except at start of the first step, and except at the start of the double support phase, information from the IMU will be used. The IMU measures the acceleration and the angular rotation in three dimensions. Since gravity produces the largest acceleration vector, it is possible to estimate the position of the robot's body (see Subsection 2.4.2). This makes it possible to design a control loop that ensures a vertical position of the robot's body. Figure 6.2 illustrates the architecture of the control system which consists of an Inertial Measurement Unit (IMU), a PID con-

troller and a hip joint. The input of the control system is the required angle of the upper body. The actual angle of the torso is measured by an IMU. The difference between the required and the actual angle is used by the PID controller to adapt the output angles of the hip joint.

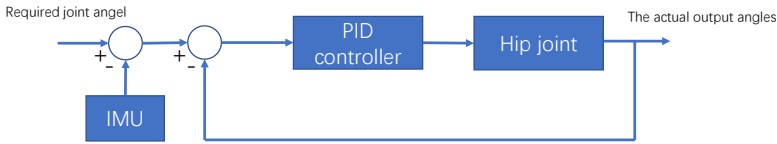


Figure 6.2: The structure of closed-loop control systems for body motion.

### 6.3.1. ANALYSIS OF THE SINGLE SUPPORT PHASE

A total of 9 control parameters serve to define the walking gaits (see Section 5.3). The parameters relevant for the single support phase are the step height, the knee bending, the torso inclination, the knee stretching time and the duration of the single support phase. When designing a controller that can handle disturbances influencing the sagittal stability, we must determine proper control parameters which guarantee a stable walk on an uneven terrain in the sagittal direction. Moreover, as we mentioned in the previous section, the robot's single step on the uneven terrain can be assumed to be one on a slope with a certain angle. So after the robot lands the swing foot on the ground, according to the height difference of the feet, we can categorize the step into two types: the uphill or downhill. If the altitude of the landed swing foot is higher than the altitude of the stance foot, the step is determined as an uphill step. If the altitude of the landed swing foot is lower than the altitude of the stance foot, the step is a downhill step.

We designed a series of experiments in the simulator *Webots* to obtain the optimal control parameters while walking on a specified slope type. In the experiments, the same reinforcement learning algorithm as described in Section 5.4, is used to find the proper control parameters that can generate a stable walking gait on different slopes. The experiments require that the robot walks on slopes where the slope angle varies from  $-0.17$  rad to  $0.17$  rad in the sagittal plane, which corresponds to  $-10^\circ$  to  $10^\circ$ . We repeatedly let the robot walk 10 steps while running the reinforcement learning algorithm. We do not run the learning algorithm for fixed walking speeds of the robot as we did in the previous chapter. Instead, we choose to fix the step size  $\theta_1$  at 3.6 cm and the step time  $\theta_4$  at 650 ms. These two parameter values determine the most energy efficient speed of 6 cm/s on a horizontal flat ground. Since the step size specification is based on a horizontal ground surface, the actual step size on a slope may differ because the swing foot may hit the ground earlier or later depending on the slope angle. This implies that uphill walking reduces the speed while downhill walking increases the speed.

Table 6.1 shows the learned optimal control parameters of the gait for different slope angles  $p$  and the corresponding feet height difference  $h = \theta_1 \sin(p)$ . As the slope angle increases from negative to positive, the parameters  $\theta_2$  (*Step Height*),  $\theta_4$  (*Step Time*) and  $\theta_9$  (*Ratio*) remain constant, while  $h$  (feet height difference),  $\theta_3$  (*Knee Bending*),  $\theta_5$  (*Stretch Time*),  $\theta_6$  (*Torso Pitch Inclination*) and  $\theta_8$  (*Torso Roll Inclination*) depend on the

Table 6.1: Experimentally learned control parameters relevant for the single support phase on different slopes in the sagittal direction.  $p$  is the angle of the slopes.  $h$  is the height different of the feet, which is determined by Equation  $h = \theta_1 \sin(p)$ .  $h$  is positive in an uphill step, negative in a downhill step. Parameter  $\theta_1, \theta_2, \theta_4$  and  $\theta_9$  are fixed. Parameter  $\theta_3, \theta_5, \theta_6, \theta_7, \theta_8$  need to be optimized according to the slope angle.

$p$	$h(\text{cm})$	$\theta_2(\text{cm})$	$\theta_3$	$\theta_4(\text{ms})$	$\theta_5(\text{ms})$	$\theta_6$	$\theta_7$	$\theta_8$	$\theta_9$
-10°	-0.63	3.24	8.5°	650	64	0°	(0.92 0.28)	9.5°	0.8
-7°	-0.44	3.24	9.2°	650	68	1.4°	(0.91 0.27)	7.5°	0.8
-5°	-0.31	3.24	11.7°	650	70	4.1°	(0.91 0.26)	7.1°	0.8
-3°	-0.19	3.24	13.7°	650	75	5.7°	(0.90 0.22)	6.9°	0.8
0°	0.00	3.24	14.2°	650	78	8.9°	(0.90 0.20)	6.5°	0.8
3°	0.19	3.24	14.4°	650	76	10.2°	(0.86 0.16)	5.2°	0.8
5°	0.31	3.24	15.2°	650	75	13.5°	(0.85 0.14)	3.4°	0.8
7°	0.44	3.24	15.7°	650	74	14.9°	(0.82 0.14)	2.2°	0.8
10°	0.63	3.24	16.1°	650	73	16.3°	(0.78 0.12)	1.5°	0.8

slope angle. The control parameters  $\theta_3, \theta_5$  and  $\theta_6$  regulate the locomotion in the single support phase. We must design a sagittal controller that adjusts these variable parameters in order to maintain a stable gait on unknown slopes. The Torso Roll Inclination  $\theta_8$  is a parameter of the double support phase. Its value only depends on the difference in step height. We will address the control of this parameter in Subsection 6.3.3 where we address the double support phase.

6

### 6.3.2. THE SAGITTAL CONTROLLER

The main task of the sagittal controller is to generate a leg-length policy that is able to adapt to unknown slopes. As shown in Figure 4.4, the leg-length policy is determined by two parameters:  $\theta_3$  (*Knee Bending*) and  $\theta_5$  (*Stretch Time*). The parameter  $\theta_3$  determines the shortening  $\delta(\alpha_b)$  (shown in the Figure 4.4) of the virtual telescopic leg at the beginning of the single support phase. This parameter only depends on the difference in step height and on the step time. The parameter  $\theta_3$  is determined by the bending of the knee of the swing leg at the end of the double support phase. So, the only parameter of the leg-length policy that can be controlled, is  $\theta_5$ .

A good value for the parameter  $\theta_5$  depends on the slope angle  $p$ . The only information that we have about the slope angle is the difference in step height of the two feet before a step. We could use the latter to control the parameter  $\theta_5$ . A disadvantage of this choice is the inability to compensate for disturbances during a step; e.g., strong wind, collisions with other robots, etc., which may have a similar effect as walking on a slope. Moreover, the robot cannot adapt the leg-length policy during a step. Therefore, we chose  $\alpha$  (the angle between a virtual leg and its vertical line, see Figure 4.1), angular speed  $\dot{\alpha}$ , angular acceleration  $\ddot{\alpha}$  and  $\theta_3$  as inputs for our sagittal controller. Note that the leg-length policy is a function of the angle  $\alpha$  for a fixed slope angle  $p$ . Different slope angles result in different angular speeds  $\dot{\alpha}$  and different angular accelerations  $\ddot{\alpha}$  for the same angle  $\alpha$ . These inputs are therefore used to implicitly identify the slope angle  $p$ . Finally, in order to adapt the leg-length policy during the single support phase, the controller should directly output the knee angle of the stance leg. Therefore, we decided to

use a neural network, shown on the left side of Figure 6.4, to generate the trajectory of the angle on the stance knee joint (see Figure 6.3) using  $\alpha$ ,  $\dot{\alpha}$ ,  $\ddot{\alpha}$  and  $\theta_3$  as inputs. To collect the training data for the neural network, we let the robot walk on a slope using the controller described in the previous chapter and using the learned parameter values for the slope, and we sampled the knee angle  $\lambda_{kst}$  and the inputs  $\alpha$ ,  $\dot{\alpha}$ ,  $\ddot{\alpha}$  and  $\theta_3$  (see Figure 6.3). The resulting data was used to train the neural network using back propagation.

In the Figure 6.3,  $\lambda_{kst}$  is the angle of knee joint on the stance leg. Angle  $\lambda_{kst}^b$  is the value of the knee angle at the beginning of the step. Angle  $\alpha$  is the angle between the stance leg and the vertical axis.  $\alpha_b$  and  $\alpha_e$  denote the value of  $\alpha$  at the beginning and at the end of a step, respectively.  $\lambda_{kst}^i$  is the value of the  $i_{th}$  sampled angle of the knee joint.

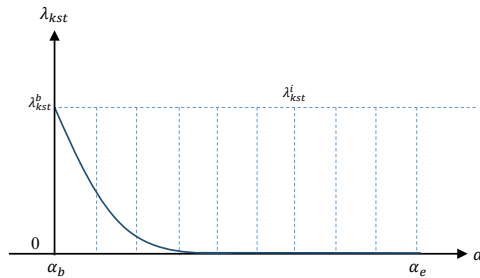


Figure 6.3: The trajectory of a knee joint on the stance leg and its sampled value.

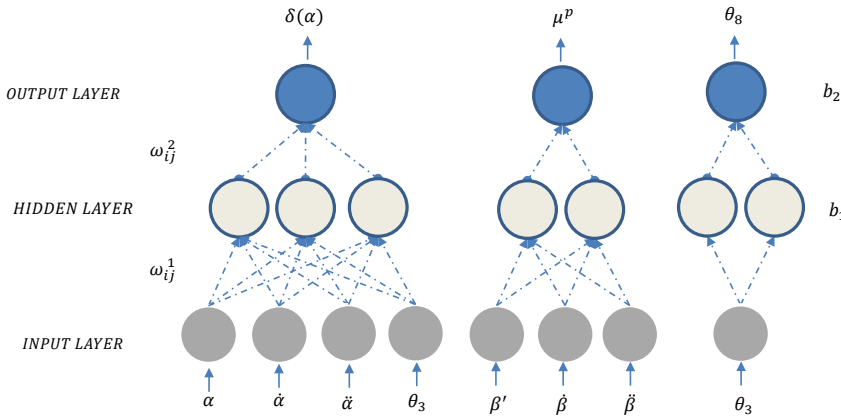


Figure 6.4: The neural networks used to generate a stable walk on uneven terrains. The neural network on the left has four inputs to outputs the sampled angle of the knee joint, which implements the leg-length policy  $\delta(\alpha)$ . The middle one has three inputs and its output is the sampled knee stiffness of swing leg  $\mu_p$ . The right most neural network uses control parameter  $\theta_3$  as the input and outputs the control parameter  $\theta_8$ .

### 6.3.3. ANALYSIS OF THE DOUBLE SUPPORT PHASE

When designing the controller that can handle disturbances during the double support phase, we must determine the control parameters of the gait which guarantee the robot's lateral stability on an uneven terrain. As for the sagittal controller, we choose to fix the step size  $\theta_1$  at 3.6 cm and the step time  $\theta_4$  at 650 ms, which corresponds to the most energy efficient gait on a horizontal flat surface. We let the robot walk perpendicular to the slope direction. Since the left and right feet are at different heights, the control parameters for the left and right leg might be different. Therefore, the control parameters specified in Section 5.3, are split into two parameter sets: for the left leg  $\theta_i^L$  and for the right leg  $\theta_i^R$  respectively. We used the controller described in Chapter 5 to learn the optimal set of the control parameters for walking on different slopes.

We designed a series of experiments in the simulator Webots to obtain the optimal control parameters while walking on a specified slope type. In the experiments, the same policy reinforcement learning method used in Subsection 5.4, was used to find proper control parameters that can generate a stable walking gait for different slopes. The experiments required the robot to stand on slopes for which the tilt angle varied from -0.09 rad to 0.09 rad in the robot's frontal plane, which corresponds to  $-5^\circ$  to  $5^\circ$ . Since in frontal plane, slopes with negative tilt angles are opposite to those with positive angles, the results of left leg on slopes with negative angle are identical to the results of right leg on slopes with positive angle and vice versa. Therefore, we decided to run the experiments on slopes for which the tilt angle varies only from 0.00 rad to 0.09 rad in robot's frontal plane (see Figure 6.1). We repeatedly let the robot walk for 5 seconds while running the reinforcement learning algorithm. After the result of reinforcement learning converged while the robot's movement became stable, the control parameters were recorded.

Table 6.2 shows the control parameters of the gait for the different lateral slopes  $q$  and corresponding feet height difference  $k = d \tan(q)$ . Here,  $d$  denotes the distance between the centers of both feet in the lateral direction, which is 9.5 cm in our experiments. As the slope angle  $q$  increases from  $0^\circ$  to  $5^\circ$ , the parameters  $\theta_2$  (*Step Height*),  $\theta_4$  (*Step Time*), and  $\theta_9$  (*Ratio*) were kept constant, while  $k$  (feet height difference),  $\theta_3$  (*Knee Bending*),  $\theta_5$  (*Stretch Time*),  $\theta_6$  (*Torso Pitch Inclination*),  $\theta_7$  (*Quadratic Bezier point*) and  $\theta_8$  (*Torso Roll Inclination*) depend on the slope angle. The parameters  $\theta_3$  and  $\theta_5$  overlap with the parameters controlled by the sagittal controller. Table 6.1 and Table 6.2 show that the same height difference between the two feet ( $h$  in the Table 6.1 and  $k$  in the Table 6.2) results in the same values for the parameters  $\theta_3$  and  $\theta_5$ .

The lateral controller only needs to adapt the parameters  $\theta_7$  and  $\theta_8$ . The parameter  $\theta_8$  also played a role when walking uphill or downhill. Tables 6.1 and 6.2 show that the same height difference between the two feet ( $h$  in the Table 6.1 and  $k$  in the Table 6.2) results in almost the same value for the parameter  $\theta_8$ . Therefore,  $\theta_8$  does not depend on the type of slope. We designed a lateral controller for the double support phase to adjust the parameters  $\theta_7$  and  $\theta_8$  in order to maintain the gait's stability on unknown slopes.

### 6.3.4. THE LATERAL CONTROLLER

The main task of the lateral controller is to ensure that the robot is balanced during the double support phase. The lateral stability is determined by the parameters  $\theta_7$  (*Quadratic Bezier point*) and  $\theta_8$  (*Torso Roll Inclination*). Both parameters depend on the lateral slope

Table 6.2: Experimentally learned control parameters relevant for the double support phase on different slopes in the lateral direction

$q$	k (cm)	$\theta_2^L$ (cm)	$\theta_3^L$	$\theta_4^L$ (ms)	$\theta_5^L$ (ms)	$\theta_6^L$	$\theta_7^L$	$\theta_8^L$	$\theta_9^L$
0°	0.00	3.24	14.2°	650	78	8.9°	(0.90,0.20)	6.5°	0.8
1°	0.17	3.24	14.5°	650	76	10.1°	(0.86 0.16)	5.5°	0.8
2°	0.33	3.24	15.3°	650	75	13.5°	(0.84 0.14)	3.5°	0.8
3°	0.50	3.24	15.5°	650	74	15.1°	(0.80 0.12)	2.0°	0.8
4°	0.66	3.24	16.1°	650	72	16.8°	(0.78 0.12)	1.5°	0.8
5°	0.83	3.24	16.4°	650	70	17.3°	(0.76 0.12)	1.5°	0.8
$q$	k (cm)	$\theta_2^R$ (cm)	$\theta_3^R$	$\theta_4^R$ (ms)	$\theta_5^R$ (ms)	$\theta_6^R$	$\theta_7^R$	$\theta_8^R$	$\theta_9^R$
0°	0.00	3.24	14.2°	650	78	8.9°	(0.90,0.20)	6.5°	0.8
1°	-0.17	3.24	13.5°	650	74	5.8°	(0.90 0.22)	6.7°	0.8
2°	-0.33	3.24	11.7°	650	70	4.0°	(0.91 0.26)	7.1°	0.8
3°	-0.50	3.24	9.6°	650	68	1.3°	(0.92 0.28)	8.5°	0.8
4°	-0.66	3.24	8.5°	650	65	0.0°	(0.92 0.29)	9.5°	0.8
5°	-0.83	3.24	8.2°	650	65	0.0°	(0.94 0.29)	9.5°	0.8

angle.

The knee angle  $\lambda_{\text{ksw}}$  of the swing leg at the end of the double support phase, which is equal to  $\theta_3$ , encodes information about the height difference between the two feet. We can use the information to control the parameter  $\theta_8$  using the right most neural network displayed in Figure 6.4. We used the information from Table 6.2 to train this neural network using back propagation.

We could also use a neural network to set the parameter  $\theta_7$  using  $\theta_3$  as input. A disadvantage of this choice is the inability to compensate for disturbances such as wind, which may have a similar effect as walking on a slope, because the robot cannot adapt the force policy during a step. Therefore, we chose angle  $\beta'$ , angular velocity  $\dot{\beta}$  and angular acceleration  $\ddot{\beta}$  as inputs for our controller. Angle  $\beta'$  is the angle between the swing leg and the vertical axis in the frontal plane,  $\beta$  is the angle between the stance leg and the vertical axis in the frontal plane, as illustrated in Figure 4.7. Note that the force policy is a function of the angle  $\beta'$  for a fixed slope angle. Different slope angles result in different angular speeds  $\dot{\beta}$  and different angular accelerations  $\ddot{\beta}$  for the same angle  $\beta'$ . These inputs are therefore used to implicitly identify the lateral slope angle.

At the end of the double support phase,  $(\beta', \dot{\beta}, \ddot{\beta})$  must become equal to  $(0, 0, 0)$ . So, in the neighborhood of  $(0, 0, 0)$  the value of  $\theta_7$  is not well defined. Therefore,  $\theta_7$  cannot be adapted by a neural network that uses  $(\beta', \dot{\beta}, \ddot{\beta})$  as inputs. Instead of  $\theta_7$ , the trajectory of stiffness will be directly determined by a neural network controller. The sampled stiffness values from the trajectories defined by different values of  $\theta_7$  can be seen in Figure 6.5.

## 6.4. GAIT CONTROLLER

In the previous section, we investigated how our controller described in Chapter 5 must be adapted to enable walking on uneven terrains. The main adaptations are:

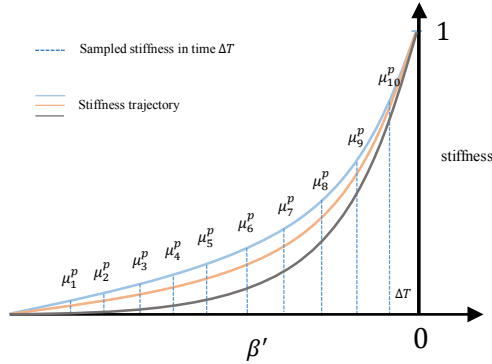


Figure 6.5: Sampled stiffness of the knee joint on the swing leg over angle  $\beta'$

- The IMU is used to ensure that the body is normally in a vertical position, except when the robot starts walking ( $\theta_6$ ) and at the start of each double support phase ( $\theta_8$ ).
- The leg-length policy is generated by a neural network instead of a function implementing a sinusoid.
- The *Torso Roll Inclination*  $\theta_8$  is set by a neural network that uses the knee bending of the swing leg at the end of the double support phase as input.
- The trajectory of the knee stiffness is generated by a neural network instead of a function implementing the Quadratic Bezier curve.

The design of the resulting controller is shown in Figure 6.6.

Beside these changes to the controller described in Chapter 5, two additional changes have been made. The finite state machine of the original controller is time-based. Timers determine the length of the single and the double support phase. On an uneven terrain without fixed slope angles, the duration of the single and the double support phase can no longer be known. Therefore, the time-based finite state machine is replaced by an event-based finite state machine. We use two events to trigger state transitions, the heel-strike and the moment that  $\beta'$  becomes 0. The heel-strike causes the transition from the single to the double support phase, and the moment that  $\beta'$  becomes 0 causes the transition from the double to the single support phase.

The neural networks use  $(\alpha, \dot{\alpha}, \ddot{\alpha})$ ,  $\theta_3$  and  $(\beta', \dot{\beta}, \ddot{\beta})$  respectively, as inputs. So, the gait controller needs to determine these values. Since the body is kept in a vertical position using the IMU, we can compute the position of the stance foot and of the swing foot in the double support phase using forward kinematics with the joint angle as input. This enables us to also compute  $\alpha$  and  $\beta'$ . The Nao robot that we use in the experiments, has a built-in function `ALMotionProxy::getPosition` that implements these computations and returns the value of  $\alpha$  and  $\beta'$  as outputs. The remaining inputs of the neural networks can be determined using the IMU. The gyroscope of the IMU gives information about  $\dot{\alpha}$  and  $\dot{\beta}$  and the accelerometer gives information about  $\ddot{\alpha}$  and  $\ddot{\beta}$ . Note that these values



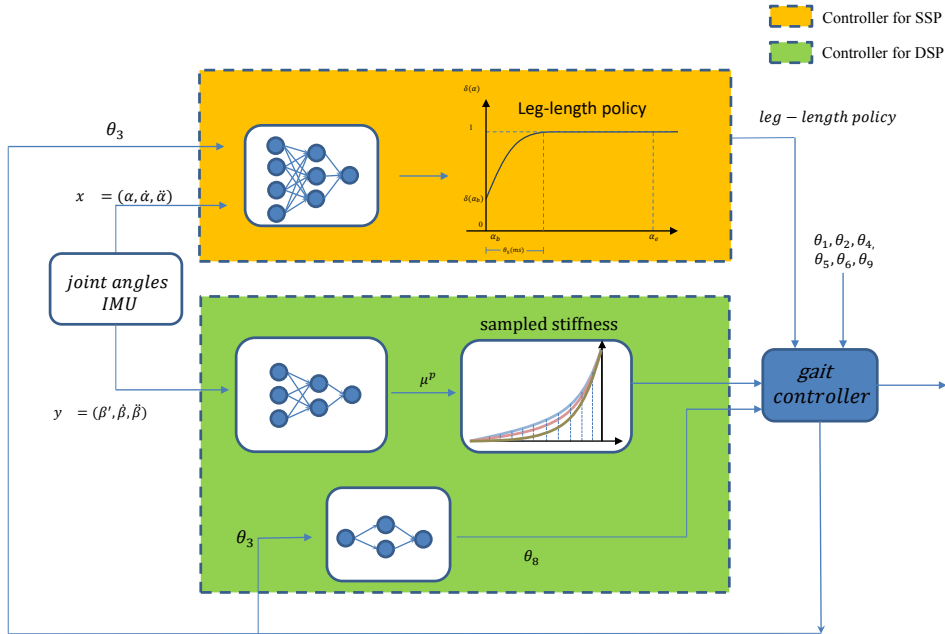


Figure 6.6: Structure of the controller for the single and the double support phase. The top neural network outputs the sampled values of the stance leg's knee angle, the middle neural network outputs the sampled values of the swing leg's knee joint stiffness, and the bottom neural network determines *Torso Roll Inclination*

can also be derived by monitoring the changes in the angles  $\alpha$  and  $\beta'$  respectively, or the changes in the angles  $\lambda_{\text{hpst}}$  and  $\lambda_{\text{hrsw}}$  respectively.

## 6.5. EXPERIMENTS

We implemented three neural networks to control the robot's stability by adjusting the controller parameters to unknown slope. All of the neural networks has a simple structure with one hidden layer and use Sigmoid as the activation function. The back-propagation method has been applied for training multi-layer feed-forward networks. In the Figure 6.4, the first neural network has four inputs and its output is the sampled angle of the knee angle, which implements the leg-length policy  $\delta(\alpha)$ . The second one has three inputs and its output is the sampled knee stiffness of swing leg  $\mu_p$ . The last neural network is the most simple one, which uses control parameter  $\theta_3$  as the input and outputs the control parameter  $\theta_8$ . With the trained networks, we implemented a controller that enables a Nao robot walk on uneven terrains.

### 6.5.1. UNEVEN TERRAINS

Uneven terrains provide a tests-bed for controller robustness since the robot faces changes in the environment that are unexpected, such as sudden changes in inclination. To evaluate the robustness of the gait controller, we used a simulation environment that

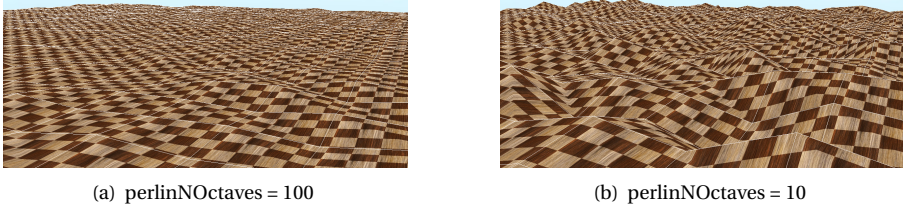


Figure 6.7: The uneven terrains generated in the Webots with the parameter *perlinNOctaves* in 100 and 10.

can generate multiple classes of terrain obstacles that include bumps, pits and slopes. Specifically, we used the commercial simulator *Webots*<sup>1</sup> that uses the Open Dynamics Engine (ODE) for physics simulation. The simulator enables us to make the terrain obstacles arbitrarily difficult. We chose environments that are challenging while remaining viable. All of the terrains are generated randomly based on Perlin noise [123] and the parameters specified in Table 6.3. Figures 6.7(a) and 6.7(b) show two *Webots* world models.

In the experiment, we generate random uneven terrains by introducing Perlin noise, which is a type of gradient noise, on the parameters *xDimension* and *zDimension*. These two parameters indicate the number of points in the grid height array in the *x* and *z* direction. The grid can provide a certain degree of irregularity of uneven terrains that we expect in the nature environment. The roughness of the terrain is controlled by the parameter *perlinNOctaves*. Figure 6.7(a) and 6.7(b) demonstrate the two uneven terrain generated based on the different *perlinNOctaves*.

Table 6.3: The parameters used to generate uneven terrains in *Webots*.

Type	parameter	Value
SFVec3f	translation	0 0 0
SFRotation	rotation	0 1 0 0
SFVec3f	size	50 5 50
SFInt32	<i>xDimension</i>	50
SFInt32	<i>zDimension</i>	50
SFInt32	randomSeed	0.5
SFBool	flatCenter	TRUE
SFBool	flatBounds	TRUE
SFInt32	<i>perlinNOctaves</i>	$-2^{31}$ to $2^{31}$

When the robot walks on the uneven floor and a new step begins with the single support phase, the neural network in the sagittal controller takes the values  $\theta_3$  (*Knee Bending*),  $\alpha$  retrieved from the joint sensors and  $(\dot{\alpha}, \ddot{\alpha})$  from the IMU as the input vector and outputs the sampled angle of the knee angle on the stance leg. When the double support phase begins, the first neural network in the lateral controller uses  $(\beta', \dot{\beta}, \ddot{\beta})$  as input and outputs the value of the knee stiffness  $\mu_p$  on the swing leg, which guarantees

<sup>1</sup>A robot simulator developed by Cyberbotics. <https://cyberbotics.com/>

that the CoM rotation around ankle stops at the balance point. The second neural network in the lateral controller takes the value of  $\theta_3$  (*Knee Bending*) as input at the end of the double support phase and outputs the value of  $\theta_8$ , which is used to determine the angle of the torso that leans in the lateral direction during the next double support phase. Together with the other fixed parameters, the gait controller generates updated joint commands to avoid CoM undershoot or overshoot caused by uneven terrains and other disturbances.

### 6.5.2. EVALUATION OF STABILITY

In order to validate the gait's dynamic stability on different randomly generated uneven terrains, we performed a series of experiments with a Nao robot in the simulation environment *Webots*. We implemented the proposed controller and validated the result by letting the robot walk a constant distance (0.5 meters). The step size was fixed to 3.6 cm.

**Simulated terrain (*perlinNOctaves* = 100)** The first terrain, in Figure 6.7(a) represent a relatively easy terrain for locomotion. This terrain was generated with the parameters in Table 6.3 and *perlinNOctaves*= 100.

Figure 6.8 shows the vertical projection of the CoM on the ground. Each foot shown in this figure is 10 times smaller than a real foot and *does not* represent the area of support during the SSP. Because of the low stiffness on the support ankle joint, the ankle can move freely in the sagittal and the lateral direction. Therefore, the area of support of a foot is constrained to a small area formed by the location of the ankle joint, instead of the foot area that contacts the ground. Each red dot in Figure 6.8 indicates the position of ankle joint in the contact area of a foot during one step. The figure shows that, in the lateral direction, the CoM trajectory does not overshoot or undershoot the support points (red dots) on uneven grounds, which proves that the proposed gait is stable on this terrain.

**Simulated terrain (*perlinNOctaves* = 10)** To generate the second terrain, shown in Figure 6.7(b), the parameter *perlinNOctaves* was set to 10, making it more prone to the existence of more hills and higher inclination.

Figure 6.9 shows the vertical projection of the CoM on the ground. Also in this figure, each foot is 10 times smaller than a real foot and *does not* represent the area of support during the SSP. Because of the low stiffness on the support ankle joint, the area of support of a foot is constrained to a small area formed by the location of the ankle joint indicated by the each red dot in Figure 6.9. Therefore, stability criteria such as the ZMP criterion are not applicable. The figure shows that, in the lateral direction, the CoM trajectory does not overshoot or undershoot the support points (red dots) on uneven grounds. Especially overshooting the support point would make it more difficult to keep the balance. The experiment demonstrates that the proposed gait is stable on this type of terrains. The controller presented in Chapter 5 cannot generate a stable walk on this type of terrain. This shows that the controller proposed in this chapter is essential for walking on uneven terrains.

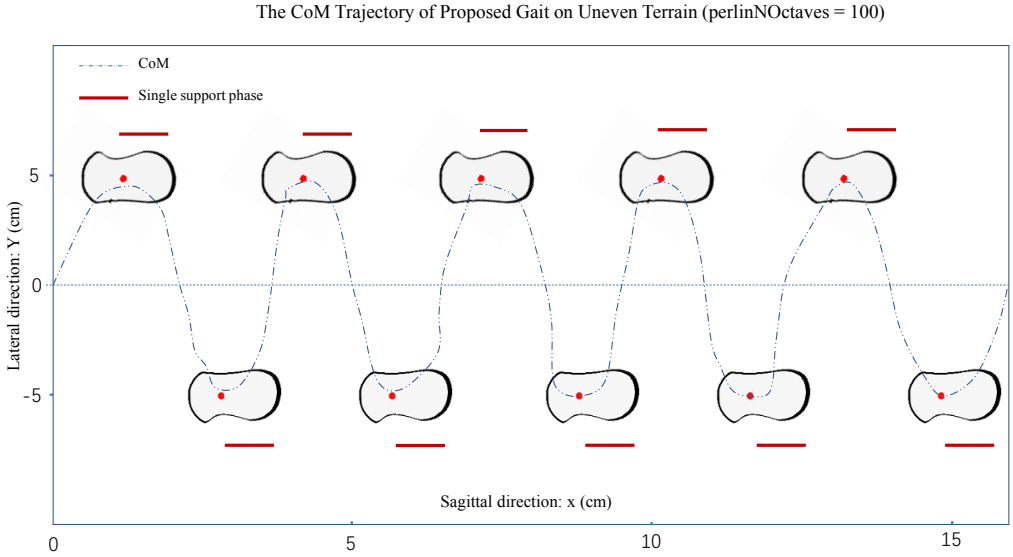


Figure 6.8: The trajectory of the CoM projection on the uneven terrain (perlinNOctaves = 100). The sample rate retrieved from IMU is 120 Hz, and each 4 samples are averaged to smooth the curve. The red dots denote the locations to the ankle joints. Note that the feet are shown at 10% of their actual size.

6

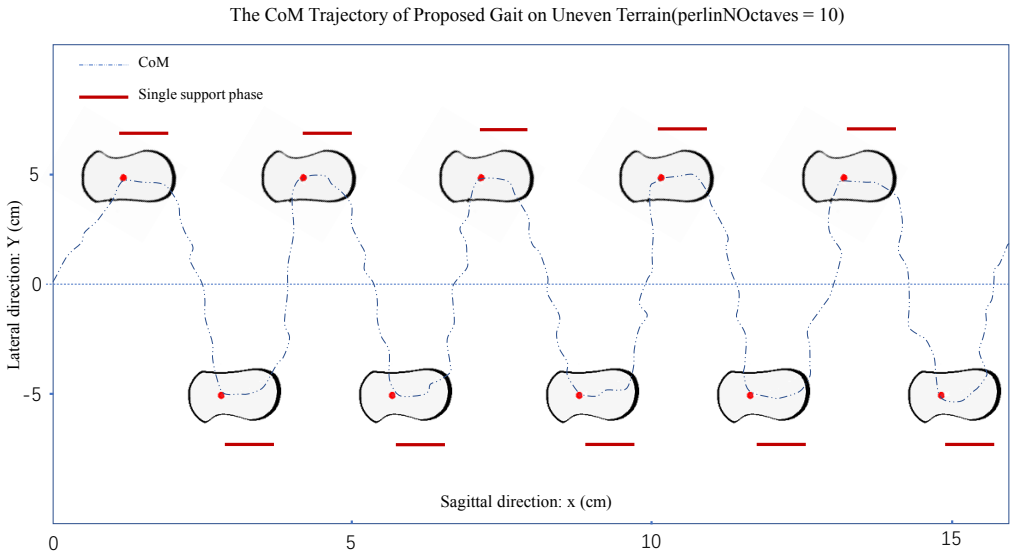


Figure 6.9: The trajectory of the CoM projection on the uneven terrain (perlinNOctaves = 10). The sample rate retrieved from IMU is 120 Hz, and each 4 samples are averaged to smooth the curve. The red dots denote the locations to the ankle joints. Note that the feet are shown at 10% of their actual size.

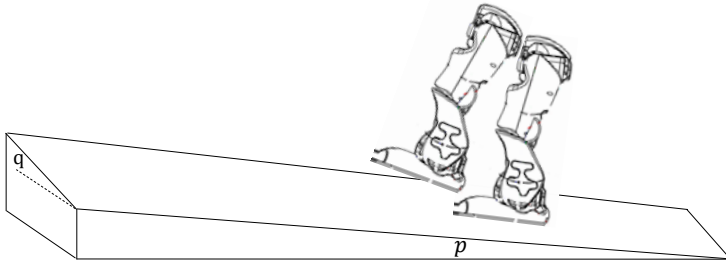


Figure 6.10: The proposed walk gait on different slopes in the simulation.

**Side-way Stability on different slopes** We analyzed the roll angles trajectories of left and right leg when robot walks on slope with angle  $0.02 \text{ rad} (\approx 1^\circ)$ ,  $0.05 \text{ rad} (\approx 3^\circ)$  and  $0.09 \text{ rad} (\approx 5^\circ)$  in both the sagittal and lateral direction (see Figure 6.10). Figure 6.11 shows that left foot is higher than the right one on the slope because the joint of right leg rotates less than the joint on left leg to let the CoM approach its balance point. Moreover, under the different slope angles, the duration of one step does not change, which means that our proposed gait can make a stable walk on different slopes without the loss of walking velocity. The reason why we constrain the slope angle within  $0.09 (\approx 5^\circ)$  is because the behavior of the robot becomes unstable when it walks on steeper slopes. We observed that the chance of a collision between the front of the Nao's big feet and slope surface increases as the slope angle increases. The collision undermines the dynamics of the gait and may cause the robot tip over. To avoid this situation, the walking pattern of swing leg needs to be changed and the robot should be able to observe in some way the angle of the surface. For example, given information of terrain, the foot sole can be pre-parallelled with slope to avoid a collision before landing on the ground. This solution is problematic because the slope angle may change and visual feedback is a challenging problem. Another problem is the Nao's smooth sole, which do not provide sufficient grip when the slope angle becomes too large. So, after many tests, we choose  $5^\circ$  as the maximal slope angle in the demo to ensure the success rate of uphill walking.

The accompanying video material<sup>2</sup> shows the Nao robot walking on an uneven terrain with our proposed gait controller in the simulator Webots, which implies that our controller can handle the altitude differences of the foot placement and adjusts the control parameters to maintain balance.

### 6.5.3. EVALUATION OF ENERGY EFFICIENCY

To evaluate the energy efficiency of the proposed gait, we ran trials with the Nao humanoid robot in the simulator environment. In the first experiment, the average electric currents that flow through the leg joints, are recorded and compared for two gaits: the Aldebaran gait and our new gait, while walking a constant distance on the flat ground. The electric current in the joints for sagittal movement (*HipPitch*, *KneePitch* and *AnklePitch*) and lateral movement (*HipRoll* and *AnkleRoll*) are recorded separately. Figure

<sup>2</sup><https://youtu.be/owHiGQm8WSg>

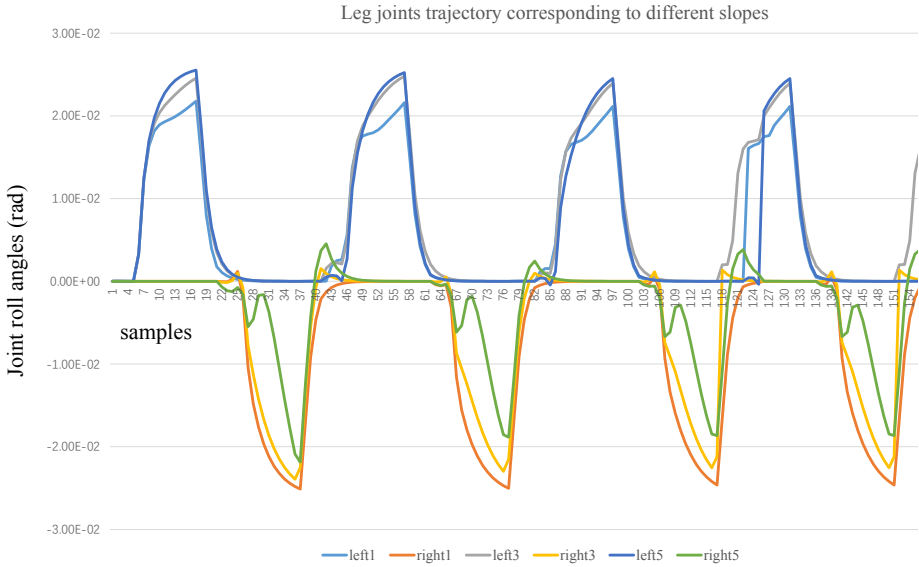


Figure 6.11: The roll angle trajectory of both legs under different slope in 1, 3, 5 degree.

6.12 shows how much the sagittal motion and lateral motion contribute to the total energy consumption. In the hip joints, the average currents for lateral motion (*HipRoll*) are almost the same (0.095 A in the Aldebaran gait, 0.09 A in the proposed new gait), but average currents (0.07 A) in the *HipPitch* for the sagittal motion in the proposed new gait is 17.6% less than that in the Aldebaran gait (0.085 A). The reason is that, compared to Aldebaran gait, we use a larger and proper step size in the proposed gait, which makes the robot cover the same distance in fewer steps. In the knee joint, the average current (0.25 A) in the proposed gait is 28.6% less than the one in the Aldebaran gait (0.35 A). Regulated by the optimal leg-length policy, the proposed gait keeps the stance leg stretched most of the step time, which has the advantage in energy efficiency over Aldebaran gait. Since the knee joint has only one degree of freedom (*KneePitch*) in forward direction, it does not have lateral motion, and thus no lateral energy cost. The ankle joint uses a stiffness of 10% in the sagittal direction (*AnklePitch*), and no stiffness in the lateral direction (*AnkleRoll*) in the proposed gait. These stiffness values reduce the energy consumption of the ankle joint from 0.10 A in the standard gait to 0.03 A in the proposed gait. This corresponds to a reduction of 57.1% in the sagittal direction and 100% in the lateral direction. The total energy reduction in the ankle joint is 70% on level ground. The overall energy consumption in the Webots simulator of leg joints of our new gait reduces by 31% compared to the Aldebaran gait on the flat ground. As described in Chapter 5, we implemented the new gait on a real Nao robot and conducted the same experiment. The results show that, compared to the Aldebaran gait of the Nao on the flat floor, the overall energy consumption of the gait reduces by 41%. The disparity between the energy costs in the simulator and the real Nao robot is caused by the different ways of energy measurement on both platforms. For example, on the real Nao robot, the electric current

of joints can be read directly from the joint sensor. Whereas, in the simulation environment, the energy consumption is estimated using the torque applied on the joints.

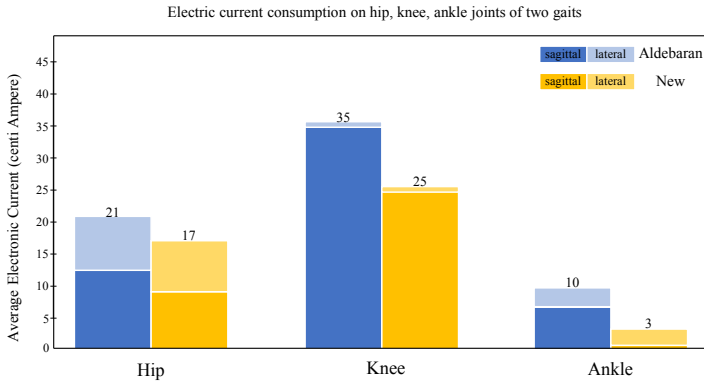


Figure 6.12: The electric current consumption on hip, knee, ankle joints of proposed new gait and Aldebaran gait.

The next experiment evaluated the energy consumption of the gait proposed in this chapter w.r.t. the increasing slope angles up to  $5^\circ$  in the sagittal and frontal plane (see Figure 6.10). In the experiment, the Nao robot is placed on a slope of angle  $p$  both in sagittal and frontal plane. In this way, the energy cost of sagittal motion and lateral motion can be recorded at the same time. We let the robot walk a constant distance on the slope of 1, 3, 5 and 7 degree separately, and record the average electric currents in the joints for sagittal movement (*HipPitch*, *KneePitch*, *AnklePitch*) and lateral movement (*HipRoll*, *AnkleRoll*). Figure 6.13 shows that more energy is consumed in the joints for the sagittal and the lateral motion as the angle of slope increases. Notably, our new gait on the slopes requires less energy for all slope angles than the Aldebaran gait on the flat ground. This indicates that the proposed gait has better performance not only w.r.t. the stability but also w.r.t. the energy efficiency on uneven terrains.

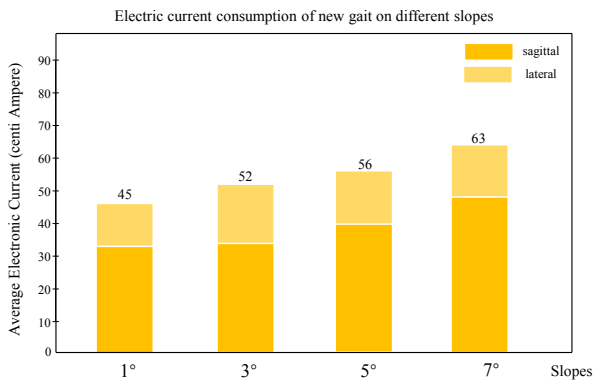


Figure 6.13: The electric current of the proposed walk gait on different slopes

## 6.6. CONCLUSION

In this chapter, we investigate how we can improve the sagittal and lateral stability of the gait when a Nao robot walks on an uneven terrain. We first make an assumption that the single step on uneven terrain can be simply assumed to be the one on the (virtual) slope with certain angle. This assumption views various terrain classes as slopes and therefore, transforms the balance control problems on uneven terrain into those on slopes. Since in the single support phase, the robot can turn freely around the ankle joint, a bump or a hole on the walking surface may disrupt the sagittal stability. Moreover, during the double support phase the robot may overshoot or undershoot its stable end point, namely, balancing above the new stance foot. We presents a feedback controller based on the feed-forward neural networks that adapts the gait parameters in order to ensure the robot's stability while it walks on an uneven terrain. The controller enables the development of leg length policy and force policy that enables highly dynamic terrain adaptive locomotion.

**Research question** The results of this chapter provide us with the answer on research question 6.

### 6. *How can we improve the stability of the walking gait on uneven terrains?*

We developed a feedback controller based on three small feed-forward neural networks. One neural network generates the leg-length policy based on the angle, angular speed and angular acceleration of the virtual telescopic stance leg. Another neural network generates the force policy based on the angle, angular speed and angular acceleration of the virtual telescopic swing leg. The last neural network set the Torso Roll Inclination using the knee angle of the swing leg at the end of the double support phase as an indication of the height difference of the two feet. The evaluation of the new controller showed that it enables a stable walk on an uneven terrain.



# 7

## CONCLUSIONS AND FUTURE RESEARCH

**R**ESearch in robotics has come a long way in the last few decades, and is moving away from being confined to industrial robots with precisely programmed motion plans to include adaptive mobile robots that can robustly operate in a variety of scenarios. Mobile robots are roughly classified into two groups: wheeled robots and legged robots. Wheeled robots have the advantage of moving fast and are mainly used for translation of long distances, such as cars. However, the mobility of the wheeled robot is limited to flat surfaces or easy terrains. Legged robots are able to move on discrete and uneven terrains, such as rubble, wreckage, ruins. In particular, one type of legged robots with anthropomorphic shape (humanoid robots), which walks on its two legs (i.e, biped walk), only needs a small contact area to the ground. Therefore, they are expected to maneuver into the sphere of our daily life and were easily adapt to environments designed for human beings.

The synthesis of biped gait on a humanoid robot is a core challenge within the field of mobile robot locomotion. This thesis investigates a novel way of biped gait synthesis on a humanoid platform, the technologies used for balance control of the biped walking gait on uneven terrains, and for the optimization of the energy consumption. This has led us to the following problem statement:

**Problem statement:**

*How to synthesize a biped walking gait with energy efficiency and dynamical stability on a humanoid robot with high degrees of freedom on uneven terrains ?*

To tackle the given problem statement, we focused in the performed research on two aspects: (1) a biped locomotion model that enables us to analyze the energy cost as a function of the type of the gait, and (2) control methods that implement the identified gait with low energy cost on a real robot. The first aspect is addressed by using an

inverted pendulum model that allows us to investigate the different walking behavior. The second aspect is addressed by first designing a controller that generates an energy-efficient gait, and then, extending the controller to enable walking on uneven grounds. These two aspects were further refined into six tasks. Each of them below corresponds to a research question formulated in Chapter 1.

1. The model used to identify an energy-efficient biped walking gait.
2. The influence of double support phase on the gait.
3. Lateral stability.
4. Controller design.
5. Gait optimization on level ground.
6. Gait optimization on uneven terrain.

In the following section, we summarize our findings to the corresponding research questions. We then discuss them in the larger context of the overall problem statement in Section 7.2. Finally, we present promising directions of future research in Section 7.3.

## 7.1. ANSWERS TO THE RESEARCH QUESTIONS

**The model used to identify an energy-efficient biped walking gait** The (Linear) Inverted Pendulum Model is the prevailing model that is used to give an elementary representation of the dynamics of the biped gait. This model represents a robot that has complex body parts, as a point mass located in the CoM position, with two telescopic legs, and the feet represented as points. With this model, the dynamics of the biped gait can be described as a set of simple physics equations. However, the model is used to analyse the dynamics and stability of a gait, but not the energy consumption. This issue has led us to the following research question.

### Research Question 1:

*Can the planar inverted pendulum model be used to create a bipedal gait with a high energy efficiency?*

The research question is addressed in Chapter 4. To answer this question, we carried out experiments to assess the effect of different joints on energy consumption, the result of which show the energy consumption on knee joints make up about 35.2% ~ 40.4% of that of whole body and the output torque of joints is directly proportional to its energy consumption. We then proposed an Inverted Pendulum model with telescopic legs which allows the length of the virtual leg (i.e., the distance between the center of mass and one of the ankles) to vary during a step. Moreover, based on this model, we proposed a leg-length policy which determines how much the virtual leg will be relatively shortened and indirectly regulates the output torque on knee joints, as described in Section 4.2.

Regarding to the energy efficiency, we optimize the leg-length policy and evaluate the effect of the optimal leg-length policy (in Figure 4.4) on the energy cost of the biped gait generated by our proposed model. Here we observe that the application of the optimal leg-length policy is beneficial for the energy efficiency of a biped walk gait. In a further experiment, we testify the potential benefit of utilizing the low stiffness on ankle joints and investigate the effect of introduction of under-actuated ankle joints on the stability of walk on flat surface. We observe that it is necessary to set the ankle pitch stiffness to 0.1 in order to assure the soles have sufficient grip to propel the Nao robot.

**The influence of the double support phase on the gait** Many studies on biped robots using the Inverted Pendulum Model ignore the double support phase [72, 74, 79, 142] or assume that the double support phase is instantaneous [88, 136, 174]. In this thesis, we extend the Inverted Pendulum Model with a double support phase taking into consideration the stability and energy efficiency. During the double support phase, the robot's weight gradually transfers from the stance leg to the swing leg, rather than instantaneously at the end of the single support phase, which improves the stability of the gait. Besides, from the viewpoint of energy efficiency, the double support phase is important, since the force on the swing leg influences the energy cost. The extended double support phase on the walk gait has lead us to the following research question.

**Research Question 2:**

*What is the influence of the double support phase on the biped gait in the sagittal plane?*

With the intent of answering this research question, we take the first step towards investigating the influence of the double support phase by introducing a force analysis in subsection 4.2. We conclude that the forces on legs in double support phase determine the impulse generated by the heel strike of the swing leg, and also influence the movement in the sagittal plane. We then proposed a force policy that determines the force that the swing leg (i.e., stance leg in the next step) applies on the mass during the double support phase. A feasible force policy has two goals: Firstly, the policy must guarantee that the stance leg stays on the ground till the double support phase finishes. Secondly, the policy must ensure that the CoM of robot balances above the new stance foot at the end of the double support phase. We investigate the effect of different force policies in the double support phase and the responding leg-length policies. The results show the force policy can regulate the force of swing leg to generate a cyclic walk after introducing the double support phase into the model. Moreover, the applied force policy does not change the shape of the optimal leg-length policy, which means have no significant influence on the gait in the single support phase.

**Lateral stability** During the double support phase, the robot switches the stance leg, which may influence the lateral stability of a gait. This has lead us to formulating the following research question.

**Research Question 3:**

*What are the requirements of a double support phase with respect to lateral stability of a*

*gait?*

The research question is addressed also in Section 4.3. We made the following requirements for the double support phase that (1) at the beginning of double support phase, the CoM is vertically above the center of the stance foot in the lateral direction. (2) During the double support phase, the CoM is supposed to transfer from the stance leg to the swing leg. (3) At the end of double support phase, the CoM must be balanced above the center of the old swing foot in the lateral direction.

To achieve a smooth CoM transition and accurate timing control, we proposed that the shape of the force policy is determined by a Quadratic Bezier curve, as illustrated in Figure 4.9. The method we proposed is (1) disturbing the balance by slightly bending the robot body inward. (2) controlling the force generated by the swing leg by means of a force policy which is generated by regulating the knee stiffness of the swing leg.

**Controller Design** The results of answering the previous research questions revealed the model derived from the Inverted Pendulum Model and extended with a leg-length policy and a force policy in the single support phase and the double support phase respectively. The implementation of the model to generate a biped gait has led us to the following research question.

#### **Research Question 4:**

*How to design the controller for gait generation?*

This research question is addressed in Chapter 5. The proposed controller consists of motion generators providing the appropriate joint functionality, which are supervised by a Finite State Machine as an action selector. Individual joint angles are computed by the motion generator, whose structure is organized into a body trajectory generator and a leg trajectory generator. The controller has 9 control parameters fine-tuning the gait. These parameters are optimized by searching for values that optimize stability and minimize energy cost. The parameters reflect the detail with which we model the gait. The parameters can be grouped into four categories: (a) sagittal motion (b) lateral motion (c) stability (d) time control.

With this controller, we implement the leg-length and the force policy by controlling the leg movement of the biped walker. Moreover, this controller generates the trajectory for the swing foot to stride in single support phase and for ankle joints of the stance leg.

**Gait optimization on level ground** The proposed controller can generate different biped gaits by changing the parameter values. However, finding the stable walk gait with the least energy cost is still challenging. To assess the energy consumption and the stability of the walk gaits, a suitable criterion is required. Besides, the controller of walking bipeds involves high-dimensional states and parameters with unknown correlation between each other, which leads to the following research question.

#### **Research Question 5:**

*How to optimize the energy efficiency for the walk gait on a flat floor ?*

This research question is addressed in Sections 5.4 and 5.5. The evaluation of the energy efficiency of the walking gait requires a criterion. Hence, we proposed a fitness function, which measures the total energy cost and the stability of the gait over a certain distance of forward walk. We explore the values of the control parameters that lead to the maximal value of the fitness function.

The optimal set of parameters is determined through policy gradient reinforcement learning. To improve the identified set of parameter approximates to a global optimum, the learning process is repeated many times with randomized initial parameters. In all cases, the reinforcement learning algorithm converged to the same set of parameter values.

We applied the optimal parameter on the gait controller to generate a new biped gait. We then evaluated the energy consumption of the new gait by comparing with the energy consumption of the standard gait which Aldebaran ships with the Nao robot. The results revealed that the energy consumption of the new gait is 41% less than for the standard gait of the Nao provided by Aldebaran.

**Gait Optimization on Uneven Terrains** The developed gait has its limitations because the stability of the new gait can not be guaranteed when external disturbances occur. This means that any disturbance such as a push or stepping on uneven terrains may jeopardize its balance. This leads to the following research question.

#### Research Question 6:

*How to optimize the stability of the walking gait on uneven terrains ?*

The research question is addressed in Chapter 6. We assume a single step on an uneven terrain can be modeled as one on a (virtual) slope with certain unknown angle. This assumption enables us to transform the balance control problem on uneven terrains into one on slopes. The parameters of the controller for level ground can be optimized for specific slopes in the sagittal and the lateral directions. We present a feedback controller based on three feed-forward neural networks, that adapts the gait parameters in order to ensure the robot's stability while walking on an uneven terrain. The neural networks directly determine the leg-length policy and the force policy, which enables continuous adaptation of the gait to the changing terrain conditions. The resulting gait, evaluated on an Aldebaran Nao robot, (1) can handle small disruptions caused by an uneven terrain, and (2) looks more like a human gait.

## 7.2. ANSWER TO PROBLEM STATEMENT

After answering the research questions, we are now able to provide an answer to the problem statement.

#### Problem statement:

*How to synthesize a biped walking gait with energy efficiency and dynamical stability on a humanoid robot with many degrees of freedom on uneven terrains ?*

Taking the answers of the research questions into account, there is a way in synthesizing a biped walk gait for a humanoid robot, the walk of which has a good performance in stability and energy efficiency on and a level ground and uneven terrains.

First, a variant of the Inverted Pendulum Model can be exploited to describe the dynamics of the proposed walk gait, where the energy efficiency can be achieved using a leg-length policy to regulate the stance leg. Second, to achieve a three-dimensional walker, a double support phase is essential. Our proposed approach can address the consequent lateral stability issue. Third, our gait controller with multiple parameters can generate various walk gaits. A stable gait with the least energy cost on level ground can be achieved by exploring the parameter space using a policy gradient reinforcement learning that maximizes the value of a fitness function, which concerns both the stability and energy efficiency. Fourth, our proposed neural network controllers improve upon the performance of stability with regard to the three-dimensional walk of the robot in the presence of an uneven terrain.

### 7.3. RECOMMENDATIONS FOR FUTURE RESEARCH

We identify several key areas which should be investigated to improve the applicability of the presented work. The following part describes possible future work directions for Chapters 4, 5 and 6.

**Chapter 4** It is interesting to extend the inverted pendulum model with a "toe off" phase. A "toe off" phase may improve the energy efficiency of a gait [93]. The "toe off" phase is a subphase of the double support phase, in which the heel of the rear foot rises while the toes remain on the ground. This motion can extend the leg length while the leg is already stretched, and therefore requires almost no torque on the knee joint. Human beings and the humanoid robots with toes can easily and naturally carry out this motion, but the Nao robot used in this research cannot because it has flat feet.

Another interesting extension of the inverted pendulum model is the addition of a small mass to the swing leg. Such an extension may enable us to better analyze a kicking motion, which is important for robot soccer.

**Chapter 5** We proposed a gait controller that generates a leg-length policy and the handles lateral movement of the CoM in the double support phase. To improve the versatility of the robot locomotion, such as allowing the robot to step over an obstacle as well as to climb stairs, we can explore higher and longer steps. The height and length of the steps might be adjusted based on the recognized obstacles. Moreover, we may investigate the possibility to place the feet more in line (i.e. the placement of swing foot is closer to the stance foot) during walking, as humans are doing. This will reduce lateral movement, which may improve the gait's stability.

**Chapter 6** We propose two directions for future work. The first is to conduct stability validation experiments in the real environment, which makes the results more objective. In this thesis, the validation experiments have been carried out only in the

simulation environment, because the roughness of the terrain can be more easily quantified and adjusted there than in a real experiment. Though the results show the good performance of the walking gait on simulated rugged terrain, we must conduct the validation experiments with a real robot in the outdoor environment to obtain a more objective evaluation on the performance of the proposed gait. The second direction of future work is cross-platform compatibility. In this thesis, the effectiveness of the proposed method for generating a biped gait has been verified on the Nao robots. However, the simple structure of the controller and the limited use of sensor information make it possible to apply the proposed approach to other humanoid robot platforms. An interesting continuation of this work will be to pack the basic locomotion skills of the controller as "downloadable" libraries that can be shared between different humanoid robot platforms.





# REFERENCES

## REFERENCES

- [1] Abbas Abdolmaleki, Nima Shafii, Luis Paulo Reis, Nuno Lau, Jan Peters, and Gerhard Neumann. Omnidirectional walking with a compliant inverted pendulum model. In *Advances in Artificial Intelligence–IBERAMIA 2014*, pages 481–493. Springer, 2014.
- [2] Aaron D. Ames. First steps toward automatically generating bipedal robotic walking from human data. In *Robot Motion and Control*, volume 2, pages 89–116. Springer, 2012.
- [3] Aaron D. Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, 2014.
- [4] Aaron D. Ames, Eric A. Cousineau, and Matthew J. Powell. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 135–144. ACM, 2012.
- [5] Shinya Aoi and Kazuo Tsuchiya. Stability analysis of a simple walking model driven by a nonlinear oscillator. In *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4450–4455. IEEE, 2004.
- [6] Shinya Aoi and Kazuo Tsuchiya. Locomotion control of a biped robot using nonlinear oscillators. *Autonomous Robots*, 19(3):219–232, 2005.
- [7] Takemasa Arakawa and Toshio Fukuda. Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization. In *Proceedings of the 1996 IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1495–1500. IEEE, 1996.
- [8] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37(6):35–58, 2018.
- [9] Andreas Aristidou and Joan Lasenby. FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011.
- [10] Giovanni Buccino, Stefan Vogt, Afra Ritzl, Gereon R. Fink, Karl Zilles, HansJoachim Freund, and Giacomo Rizzolatti. Neural circuits underlying imitation learning of hand actions: an event-related fMRI study. *Neuron*, 42(2):323–334, 2004.

- [11] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.
- [12] Adrian A. Canutescu and Roland L. Dunbrack. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12(5):963–972, 2003.
- [13] Genci Capi, ShinIchiro Kaneko, Kazuhisa Mitobe, Leonard Barolli, and Yasuo Nasu. Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. *Robotics and Autonomous Systems*, 38(2):119–128, 2002.
- [14] Genci Capi, Yasuo Nasu, Leonard Barolli, and Kazuhitsa Mitobe. Real time gait generation for autonomous humanoid robots: A case study for walking. *Robotics and Autonomous Systems*, 42(2):107–116, 2003.
- [15] G.A. Cavagna, F.P. Saibene, and R. Margaria. Mechanical work in running. *Journal of Applied Physiology*, 19(2):249–256, 1964.
- [16] P.H. Channon, S.H. Hopkins, and D.T. Pham. Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 10(02):165–172, 1992.
- [17] M.Y. Cheng and C.S. Lin. Genetic algorithm for control design of biped locomotion. *Journal of Field Robotics*, 14(5):365–373, 1997.
- [18] Sonia Chernova and Manuela Veloso. An evolutionary approach to gait learning for four-legged robots. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2562–2567. IEEE, 2004.
- [19] Christine Chevallereau, Gabriel Abba, Yannick Aoustin, Franck Plestan, Eric Westervelt, Carlos Canudas de Wit, and Jessy Grizzle. RABBIT: a testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.
- [20] Christine Chevallereau and Yannick Aoustin. Optimal reference trajectories for walking and running of a biped robot. *Robotica*, 19(05):557–569, 2001.
- [21] Christine Chevallereau, Jessy W Grizzle, and ChingLong Shih. Asymptotically stable walking of a five-link underactuated 3-D bipedal robot. *IEEE Transactions on Robotics*, 25(1):37–50, 2009.
- [22] Christine Chevallereau, Jessy W. Grizzle, and ChingLong Shih. Steering of a 3D bipedal robot with an underactuated ankle. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1242–1247. IEEE, 2010.
- [23] Junho Choi, Seonghun Hong, Woosub Lee, and Sungchul Kang. A variable stiffness joint using leaf springs for robot manipulators. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 4363–4368. IEEE, 2009.
- [24] Kyu Cheon Choi, Hyun Jeong Lee, and Min Cheol Lee. Fuzzy posture control for biped walking robot based on force sensor for ZMP. In *Proceedings of the 2006 SICE-ICASE International Joint Conference*, pages 1185–1189. IEEE, 2006.

- [25] Michael J. Coleman, Anindya Chatterjee, and Andy Ruina. Motions of a rimless spoked wheel: a simple three-dimensional system with impacts. *Dynamics and Stability of Systems*, 12(3):139–159, 1997.
- [26] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.
- [27] Steven H. Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1983–1988, 2005.
- [28] Steven H. Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.
- [29] Eric Cuevas, Daniel Zaldívar, and Raúl Rojas. Walking trajectory control for a biped robot. Technical Report B-04-18, Freie Universität Berlin, Institut für Informatik, 2004.
- [30] Houman Dallali, Mohamad Mosadeghzad, Gustavo A. Medrano-Cerda, Nicolas Docquier, Petar Kormushev, Nikos Tsagarakis, Zhibin Li, and Darwin Caldwell. Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach. In *Proceedings of the 2013 IEEE International Conference on Mechatronics*, pages 598–603. IEEE, 2013.
- [31] Marc Peter Deisenroth, Roberto Calandra, André Seyfarth, and Jan Peters. Toward fast policy search for learning legged locomotion. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1787–1792. IEEE, 2012.
- [32] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22:215–221, 1955.
- [33] Joachim Denk and Günther Schmidt. Synthesis of walking primitive databases for biped robots in 3D environments. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 1343–1349. IEEE, 2003.
- [34] Rosen Diankov. *Automated construction of robotic manipulation programs*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 2010.
- [35] Rosen Diankov and James Kuffner. Openrave: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, Carnegie Mellon University, 2008.
- [36] Jacques Duysens and Henry WAA Van de Crommert. Neural control of locomotion; part 1: The central pattern generator from cats to humans. *Gait & Posture*, 7(2):131–141, 1998.

- [37] Gen Endo, Jun Morimoto, Takamitsu Matsubara, Jun Nakanishi, and Gordon Cheng. Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot. *The International Journal of Robotics Research*, 27(2):213–228, 2008.
- [38] Gen Endo, Jun Morimoto, Jun Nakanishi, and Gordon Cheng. An empirical exploration of a neural oscillator for biped locomotion control. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 3, pages 3036–3042. IEEE, 2004.
- [39] Gen Endo, Jun Nakanishi, Jun Morimoto, and Gordon Cheng. Experimental studies of a neural oscillator for biped locomotion with QRIO. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 596–602. IEEE, 2005.
- [40] Johannes Engelsberger, Christian Ott, and Alin Albu-Schäffer. Three-dimensional bipedal walking control based on divergent component of motion. *IEEE Transactions on Robotics*, 31(2):355–368, 2015.
- [41] Adrien Escande, Nicolas Mansard, and Pierre Brice Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 3733–3738. IEEE, 2010.
- [42] B. Espiau and A. Goswami. Compass gait revisited. In *Proceedings of the IFAC Symposium on Robot Control*, pages 839–846, 1994.
- [43] Siyuan Feng, Eric Whitman, X. Xinjilefu, and Christopher G. Atkeson. Optimization-based full body control for the DARPA robotics challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.
- [44] Yasutaka Fujimoto and Atsuo Kawamura. Simulation of an autonomous biped walking robot including environmental force interaction. *IEEE Robotics & Automation Magazine*, 5(2):33–42, 1998.
- [45] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The simplest walking model: stability, complexity, and scaling. *Journal of Biomechanical Engineering*, 120(2):281–288, 1998.
- [46] S.M. Gatesy and A.A. Biewener. Bipedal locomotion: effects of speed, size and limb posture in birds and humans. *Journal of Zoology*, 224(1):127–147, 1991.
- [47] Tao Geng, Bernd Porr, and Florentin Wörgötter. Fast biped walking with a reflexive controller and real-time policy searching. In *Advances in Neural Information Processing Systems*, pages 427–434, 2006.
- [48] Tao Geng, Bernd Porr, and Florentin Wörgötter. A reflexive neural network for dynamic biped walking control. *Neural Computation*, 18(5):1156–1196, 2006.

- [49] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London B: Biological Sciences*, 270(1529):2173–2183, 2003.
- [50] Michael A. Goodrich and Alan C. Schultz. Human–robot interaction: a survey. *Foundations and Trends in Human–Computer Interaction*, 1(3):203–275, 2008.
- [51] Ambarish Goswami. Postural stability of biped robots and the foot-rotation indicator (FRI) point. *The International Journal of Robotics Research*, 18(6):523–533, 1999.
- [52] Ambarish Goswami, Bernard Espiau, and Ahmed Keramane. Limit cycles and their stability in a passive bipedal gait. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, volume 1, pages 246–251. IEEE, 1996.
- [53] Ambarish Goswami, Benoit Thuilot, and Bernard Espiau. A study of the passive gait of a compass-like biped robot: Symmetry and chaos. *The International Journal of Robotics Research*, 17(12):1282–1301, 1998.
- [54] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The nao humanoid: a combination of performance and affordability. *Computing Research Repository(CoRR) abs/0807.3223*, 2008.
- [55] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. Mechatronic design of NAO humanoid. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 769–774. IEEE, 2009.
- [56] Yasuhisa Hasegawa, Takemasa Arakawa, and Toshio Fukuda. Trajectory generation for biped locomotion robot. *Mechatronics*, 10(1):67–89, 2000.
- [57] Norman C. Heglund and C. Richard Taylor. Speed, stride frequency and energy cost per stride: how do they change with body size and gait? *Journal of Experimental Biology*, 138(1):301–318, 1988.
- [58] Rodolphe Héliot and Bernard Espiau. Multisensor input for cpg-based sensory—motor coordination. *IEEE Transactions on Robotics*, 24(1):191–195, 2008.
- [59] Ayonga Hereid, Eric A. Cousineau, Christian M. Hubicki, and Aaron D. Ames. 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pages 1447–1454. IEEE, 2016.
- [60] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, volume 2, pages 1321–1326. IEEE, 1998.

- [61] Masato Hirose and Kenichi Ogawa. Honda humanoid robots development. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 365(1850):11–19, 2007.
- [62] R. Hirose and T. Takenaka. Development of the humanoid robot ASIMO. *Honda Research and Development Technical Review*, 13(01):1–6, 2001.
- [63] Hirohisa Hirukawa, Shizuko Hattori, Kensuke Harada, Shuuji Kajita, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Mitsuharu Morisawa. A universal stability criterion of the foot contact of legged robots-adios ZMP. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 1976–1983. IEEE, 2006.
- [64] Alan Hreljac and Philip E. Martin. The relationship between smoothness and economy during walking. *Biological Cybernetics*, 69(3):213–218, 1993.
- [65] Sang-Ho Hyon, Joshua G. Hale, and Gordon Cheng. Full-body compliant human-humanoid interaction: balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, 23(5):884–898, 2007.
- [66] Tomiharu Yoshida Ichiro Kato, Masakatsu Fujie and Ken Ichiryu. Development of legged walking robots. *Hitachi Review*, 68(10):25–30, 1985.
- [67] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.
- [68] S. Ikemoto, H. B. Amor, T. Minato, B. Jung, and H. Ishiguro. Physical human-robot interaction: Mutual learning and adaptation. *IEEE Robotics Automation Magazine*, 19(4):24–35, Dec 2012.
- [69] Tatsuzo Ishida, Yoshihiro Kuroki, and Jinichi Yamaguchi. Mechanical system of a small biped entertainment robot. In *Proceedings of the 2003 IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1129–1134. IEEE, 2003.
- [70] Oleg Ivlev and Axel Gräser. An analytical method for the inverse kinematics of redundant robots. *Intelligent Automation and Active Systems*, 2(6):7, 1997.
- [71] Rahul Kumar Jha, Balvinder Singh, and Dilip Kumar Pratihari. On-line stable gait generation of a two-legged robot using a genetic-fuzzy system. *Robotics and Autonomous Systems*, 53(1):15–35, 2005.
- [72] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626. IEEE, 2003.

- [73] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kazuhito Yokoi, and Hirohisa Hirukawa. A realtime pattern generator for biped walking. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 1, pages 31–37. IEEE, 2002.
- [74] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 239–246. IEEE, 2001.
- [75] Shuuji Kajita, Kenji Kaneko, Mitsuharu Morisawa, Shinichiro Nakaoka, and Hirohisa Hirukawa. ZMP-based biped running enhanced by toe springs. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 3963–3969. IEEE, 2007.
- [76] Shuuji Kajita, Mitsuharu Morisawa, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Hirohisa Hirukawa. Biped walking pattern generator allowing auxiliary ZMP control. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2993–2999. IEEE, 2006.
- [77] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1405–1411. IEEE, 1991.
- [78] Noriyuki Kanehira, T.U. Kawasaki, Shigehiko Ohta, T. Ismumi, Tadahiro Kawada, Fumio Kanehiro, Shuuji Kajita, and Kenji Kaneko. Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development. In *Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 2455–2460. IEEE, 2002.
- [79] Fumio Kanehiro, Hirohisa Hirukawa, and Shuuji Kajita. Openhrp: Open architecture humanoid robotics platform. *The International Journal of Robotics Research*, 23(2):155–165, 2004.
- [80] Kenji Kaneko, Kensuke Harada, Fumio Kanehiro, Go Miyamori, and Kazuhiko Akachi. Humanoid robot HRP-3. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2471–2478. IEEE, 2008.
- [81] Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Kazuhiko Akachi, Go Miyamori, Atsushi Hayashi, and Noriyuki Kanehira. Humanoid robot HRP-4-humanoid robotics platform with lightweight and slim body. In *Proceedings of the 2011 IEEE International Conference on Intelligent Robots and Systems*, pages 4400–4407. IEEE, 2011.
- [82] Jae Won Kho, Dong Cheol Lim, and Tae Yong Kuc. Implementation of an intelligent controller for biped walking robot using genetic algorithm. In *Proceedings of the 2006 IEEE International Symposium on Industrial Electronics*, volume 1, pages 49–54. IEEE, 2006.

- [83] Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *Journal of Intelligent & Robotic Systems*, 48(4):457–484, 2007.
- [84] Hiroshi Kimura, Seiichi Akiyama, and Kazuaki Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, 7(3):247–258, 1999.
- [85] Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, volume 3, pages 2619–2624, 2004.
- [86] Tadashi Komatsu and Makiko Usui. Dynamic walking and running of a bipedal robot using hybrid central pattern generator method. In *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation*, volume 2, pages 987–992. IEEE, 2005.
- [87] Kazuhiro Kosuge and Yasuhisa Hirata. Human-robot interaction. In *Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics*, pages 8–11. IEEE, 2004.
- [88] Shunsuke Kudoh and Taku Komura. C/sup 2/ continuous gait-pattern generation for biped robots. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1135–1140. IEEE, 2003.
- [89] James Kuffner, Satoshi Kagami, Koichi Nishiwaki, Masayuki Inaba, and Hirochika Inoue. Online footstep planning for humanoid robots. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 932–937. IEEE, 2003.
- [90] Jason Kulk and James Welsh. A low power walk for the NAO robot. In *Proceedings of the 2008 Australasian Conference on Robotics & Automation*, pages 1–7, 2008.
- [91] Andrew L. Kun and W. Thomas Miller. Control of variable speed gaits for a biped robot. *IEEE Robotics & Automation Magazine*, 6(3):19–29, 1999.
- [92] Andrija Kun and W. Thomas Miller. Adaptive dynamic balance of a biped robot using neural networks. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, volume 1, pages 240–245. IEEE, 1996.
- [93] Arthur D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124(1):113–120, 2002.
- [94] Jørgen Christian Larsen and Kasper Stoy. Energy efficiency of robot locomotion increases proportional to weight. *Procedia Computer Science*, 7:228–230, 2011.
- [95] Tine Lefebvre, Jing Xiao, Herman Bruyninckx, and Gudrun De Gerssem. Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499, 2005.



- [96] Sebastian Lohmeier, Thomas Buschmann, Heinz Ulbrich, and Friedrich Pfeiffer. Modular joint design for performance enhanced humanoid robot LOLA. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, volume 1, pages 88–93. IEEE, 2006.
- [97] Megumi Masuda and Shohei Kato. Motion rendering system for emotion expression of human form robots based on laban movement analysis. In *Proceedings of the 2010 IEEE International Symposium on Robot and Human Interactive Communication*, pages 324–329. IEEE, 2010.
- [98] Maja J. Matarić. *The robotics primer*. MIT Press, 2007.
- [99] J. M. McCarthy. *Introduction to Theoretical Kinematics*. MIT Press, Cambridge, MA, USA, 1990.
- [100] Tad McGeer. Stability and control of two-dimensional biped walking. Technical Report CSS-ISTR 88-0, Center for Systems Science, Simon Fraser University, 1988.
- [101] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.
- [102] Tad McGeer. Passive dynamic biped catalogue, 1991. In *Experimental Robotics II*, pages 463–490. Springer, 1993.
- [103] Robert B. McGhee and Andrew A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3:331–351, 1968.
- [104] Robert B. McGhee and Geoffrey I. Iswandhi. Adaptive locomotion of a multilegged robot over rough terrain. *IEEE transactions on Systems, Man, and Cybernetics*, 9(4):176–182, 1979.
- [105] Dominic Messuri and Charles Klein. Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion. *IEEE Journal on Robotics and Automation*, 1(3):132–141, 1985.
- [106] W. Thomas Miller. Real-time neural network control of a biped walking robot. *IEEE Control Systems*, 14(1):41–48, 1994.
- [107] Tsutomu Mita, Toru Yamaguchi, Toshio Kashiwase, and Taro Kawase. Realization of a high speed biped using modern control theory. *International Journal of Control*, 40(1):107–119, 1984.
- [108] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita. Adapting robot behavior for human–robot interaction. *IEEE Transactions on Robotics*, 24(4):911–916, Aug 2008.
- [109] Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. *Journal of the Robotics Society of Japan*, 24(7):820–829, 2006.

- [110] Seiichi Miyakoshi, Gentaro Taga, Yasuo Kuniyoshi, and Akihiko Nagakubo. Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world. In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 84–89. IEEE, 1998.
- [111] Jun Morimoto, Gen Endo, Jun Nakanishi, S. Hyon, Gordon Cheng, Darrin Bentevegna, and Christopher G. Atkeson. Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 1579–1584. IEEE, 2006.
- [112] Shuta Murakami, Eiji Yamamoto, and Kouji Fujimoto. Fuzzy control of dynamic biped walking robot. In *Proceedings of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, volume 1, pages 77–82. IEEE, 1995.
- [113] Robin R. Murphy, Tatsuya Nomura, Aude Billard, and Jennifer L. Burke. Human-robot interaction. *IEEE robotics & automation magazine*, 17(2):85–89, 2010.
- [114] Ken'ichiro Nagasaka, Hirochika Inoue, and Masayuki Inaba. Dynamic walking pattern generation for a humanoid robot based on optimal gradient method. In *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 908–913. IEEE, 1999.
- [115] Kenichiro Nagasaka, Yoshihiro Kuroki, Shinya Suzuki, Yoshihiro Itoh, and Jinichi Yamaguchi. Integrated motion control for walking, jumping and running on a small bipedal entertainment robot. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 4, pages 3189–3194. IEEE, 2004.
- [116] Jun Nakanishi, Jun Morimoto, Gen Endo, Gordon Cheng, Stefan Schaal, and Mitsuo Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.
- [117] Koichi Nishiwaki, Tomomichi Sugihara, Satoshi Kagami, Fumio Kanehiro, Masayuki Inaba, and Hirochika Inoue. Design and development of research platform for perception-action integration in humanoid robot: H6. In *Proceedings of the 2000 IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 1559–1564. IEEE, 2000.
- [118] Ill Woo Park, Jung Yup Kim, Jung-ho Lee, and Jun Ho Oh. Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO). In *Proceedings of the 2005 5th IEEE-RAS International Conference on Humanoid Robots*, pages 321–326. IEEE, 2005.
- [119] Jong Hyeon Park. Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy Sets and Systems*, 134(1):189–203, 2003.
- [120] Jong Hyeon Park and Moosung Choi. Generation of an optimal gait trajectory for biped robots using a genetic algorithm. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 47(2):715–721, 2004.

- [121] Jong Hyeon Park and Ohung Kwon. Reflex control of biped robot locomotion on a slippery surface. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, volume 4, pages 4134–4139. IEEE, 2001.
- [122] Richard P. Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. The MIT Press, 1981.
- [123] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.
- [124] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE, 2006.
- [125] Jerry Pratt, Chee Meng Chew, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.
- [126] Jerry E. Pratt and Gill A. Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 36, pages 739–748, 1998.
- [127] V. Radhakrishnan. Locomotion: dealing with friction. *Proceedings of the National Academy of Sciences of the United States of America*, 95(10):5448–5455, 1998.
- [128] Ludovic Righetti, Jonas Buchli, and Auke Jan Ijspeert. Dynamic hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, 216(2):269–281, 2006.
- [129] Ludovic Righetti and Auke Jan Ijspeert. Design methodologies for central pattern generators: an application to crawling humanoids. In *Proceedings of Robotics: Science and systems*, pages 191–198, 2006.
- [130] Erika Rogers. Human-robot interaction. *Berkshire Encyclopedia of Human-Computer Interaction*, 17(2):328–332, 2004.
- [131] Christophe Sabourin and Olivier Bruneau. Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. *Robotics and Autonomous Systems*, 51(2):81–99, 2005.
- [132] Tarik Saidouni and Guy Bessonnet. Generating globally optimised sagittal gait cycles of a biped robot. *Robotica*, 21(02):199–210, 2003.
- [133] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2478–2483. IEEE, 2002.
- [134] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

- [135] Jiang Shan and Fumio Nagashima. Neural locomotion controller design and implementation for humanoid robot HOAP-1. In *Proceedings of the 20th Annual Conference of the Robotics Society of Japan*, pages 11–14, 2002.
- [136] Maki Shibuya, Tomoyuki Suzuki, and Kouhei Ohnishi. Trajectory planning of biped robot using linear pendulum mode for double support phase. In *IEEE 2006-32nd Annual Conference on Industrial Electronics*, pages 4094–4099. IEEE, 2006.
- [137] Ryan W. Sinnet, Matthew J. Powell, Rajiv P. Shah, and Aaron D. Ames. A human-inspired hybrid control approach to bipedal robotic walking. In *the 18th World Congress of the International Federation of Automatic Control*, volume 18, pages 6904–6911, 2011.
- [138] SoftBank Robotics. H25 - links — aldebaran 2.1.4.13 documentation. [http://doc.aldebaran.com/2-1/family/nao\\_h25/links\\_h25.html](http://doc.aldebaran.com/2-1/family/nao_h25/links_h25.html), 2018. Accessed: 2018-09-30.
- [139] SoftBank Robotics. H25 - maas — aldebaran 2.1.4.13 documentation. [http://doc.aldebaran.com/2-1/family/nao\\_h25/masses\\_h25.html](http://doc.aldebaran.com/2-1/family/nao_h25/masses_h25.html), 2018. Accessed: 2018-09-30.
- [140] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, 2006.
- [141] Mike Stilman, Christopher G Atkeson, James J Kuffner, and Garth Zeglin. Dynamic programming in reduced dimensional spaces: Dynamic planning for robust biped locomotion. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2399–2404. IEEE, 2005.
- [142] Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*, volume 2, pages 1404–1409, 2002.
- [143] Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1404–1409. IEEE, 2002.
- [144] Zhenglong Sun and Nico Roos. An energy efficient gait for a nao robot. In *Proceedings of the Benelux Conference on Artificial Intelligence*, pages 323–329. Delft University of Technology, 2013.
- [145] Zhenglong Sun and Nico Roos. An energy efficient dynamic gait for a nao robot. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 267–272, 2014.
- [146] Gentaro Taga. A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics*, 73(2):97–111, 1995.

- [147] Gentaro Taga. A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, 78(1):9–17, 1998.
- [148] Gentaro Taga, Yoko Yamaguchi, and Hiroshi Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3):147–159, 1991.
- [149] SAIDA Takao, Yasuyoshi Yokokohji, and Tsuneo Yoshikawa. Fsw (feasible solution of wrench) for multi-legged robots. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 3, pages 3815–3820. IEEE, 2003.
- [150] Hiroki Takeuchi. Development of mel horse. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, volume 3, pages 3165–3171. IEEE, 2001.
- [151] Takashi Takuma and Koh Hosoda. Controlling the walking period of a pneumatic muscle walker. *The International Journal of Robotics Research*, 25(9):861–866, 2006.
- [152] Russ Tedrake, Teresa Weirui Zhang, Ming Fai Fong, and H. Sebastian Seung. Actuating a simple 3D passive dynamic walker. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 5, pages 4656–4661. IEEE, 2004.
- [153] Russ Tedrake, Teresa Weirui Zhang, and H Sebastian Seung. Learning to walk in 20 minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, volume 95585, pages 1412–1939. Yale University, 2005.
- [154] D. Tlalolini, C. Chevallereau, and Y. Aoustin. Optimal reference motions with rotation of the feet for a biped. In *Proceedings of the 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1027–1036. American Society of Mechanical Engineers, 2008.
- [155] Richard Quint van der Linde. Design, analysis, and control of a low power joint for walking robots, by phasic activation of mckibben muscles. *IEEE Transactions on Robotics and Automation*, 15(4):599–604, 1999.
- [156] Theodore Von Karman and G. Gabrielli. What price speed? specific power required for propulsion of vehicles. *Mechanical Engineering*, 72:775–781, 1950.
- [157] Miomir Vukobratović and Branislav Borovac. Zero Moment Point – thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.
- [158] L.C. Wang and Chih Cheng Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.

- [159] Wei Wang, Robert N.K. Loh, and Edward Y. Gu. Passive compliance versus active compliance in robot-based automated assembly systems. *Industrial Robot: An International Journal*, 25(1):48–57, 1998.
- [160] Chris Welman. *Inverse kinematics and geometric constraints for articulated figure manipulation*. PhD thesis, Simon Fraser University, 1993.
- [161] Eric R. Westervelt, Jessie W. Grizzle, and Daniel E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 2003.
- [162] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [163] David R. Wright. *Finite state machines*. Carolina State University, 2005.
- [164] Xuesu Xiao and Whittaker William L. Energy considerations for wheeled mobile robots operating on a single battery discharge. Technical Report CMU-RI-TR-14-16, Robotics Institute, Carnegie Mellon University, 2014.
- [165] Shishir Nadubettu Yadukumar, Murali Pasupuleti, and Aaron D. Ames. Human-inspired underactuated bipedal robotic walking with AMBER on flat-ground, up-slope and uneven terrain. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2478–2483. IEEE, 2012.
- [166] Shishir Nadubettu Yadukumar, Murali Pasupuleti, and Aaron D. Ames. From formal methods to algorithmic implementation of human inspired control on bipedal robots. In *Algorithmic Foundations of Robotics X*, pages 511–526. Springer, 2013.
- [167] Jin’ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue, and Atsuo Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, volume 1, pages 368–374. IEEE, 1999.
- [168] Chenbo Yin, Albert Albers, Jens Ottnad, and Pascal Haussler. Stability maintenance of a humanoid robot under disturbance with fictitious zero-moment point (FZMP). In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3149–3156. IEEE, 2005.
- [169] KangKang Yin, Kevin Loken, and Michiel van de Panne. Simbicon: Simple biped locomotion control. In *ACM Transactions on Graphics*, volume 26, pages 105–115. ACM, 2007.
- [170] Kan Yoneda and Shigeo Hirose. Tumble stability criterion of integrated locomotion and manipulation. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 870–876. IEEE, 1996.
- [171] Fu Zhongtao, Wenyu Yang, and Zhen Yang. Solution of inverse kinematics for 6r robot manipulators with offset wrist based on geometric algebra. *Journal of Mechanisms and Robotics*, 5:310081–310087, 08 2013.

- [172] Changjiu Zhou and Kanniah Jagannathan. Adaptive network based fuzzy control of a dynamic biped walking robot. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, pages 109–116. IEEE, 1996.
- [173] Changjiu Zhou and Qingchun Meng. Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy Sets and Systems*, 134(1):169–187, 2003.
- [174] Chi Zhu and Atsuo Kawamura. Walking principle analysis for biped robot with ZMP concept, friction constraint, and inverted pendulum model. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 364–369. IEEE, 2003.

z





## SUMMARY

The term **robot** comes from the Slavic word *rabota*, which means labour or work. It was first introduced into English vocabulary by the Czech playwright Karel Capek in his 1920 science fiction play *Rossumovi Univerzální Roboti* (*Rossum's Universal Robots*). Since then the term has been applied to a great variety of mechanical devices, such as industrial manipulators, autonomous land rovers, underwater vehicles, etc. The machine that is autonomous or semi-autonomous, usually in the control of the computer, can be called a robot, which presents many challenging and interesting research problems. To solve the problems, Robotics becomes a rapid-developing field of cross-disciplinary technology, which includes mechanical engineering, electronics engineering, computer science, and others. These technologies are exploited to develop robots that can replicate human actions and replace human labor. Robots can be used in any situation and for any purpose, but today many are used in manufacturing processes, and in dangerous environments such as military operations area, nuclear power plant, skyscraper, etc.

To date, robots perform great feats of manipulation, especially in the industrial area. They can move heavy objects, and repeat assemble work that usually requires high precision. Commercially available robotic toys and vacuum cleaner inhabit our living spaces, drones soar in the sky and unmanned underwater vehicles have dived into the deep water. These successes appear to omen an explosion of robotic application in our daily lives. But without advances in robotic motion, many promising robotic applications will not be possible.

A common problem faced by the robotic researchers is the movement of the robot. Depending on the environment in which they operate, mobile robots can be divided into several categories such as: flying robot, amphibious robots and land robots. The latter can be divided according to the way they move. There are wheeled robots and legged robots. Each of the robot designs has its merits and drawbacks. Compared to the wheeled robot, which are able to move in a high efficient way on even surfaces, legged robot are capable of stepping across the obstacles on uneven terrains. The modern mobile robots have a large amount of motor actions to move the whole body from one point to another. Bipedal walking (the self-propelled movement of the robots with two legs) could be considered the most representative behavior. The most common approach to control the walking behavior and balance of a robot with two legs is called Zero Moment Point (ZMP) [157]. This approach computes the point where the whole foot needs to be placed in order to have no moment in the horizontal direction. In other words, ZMP is the point where the sum of the vertical inertial and gravity force equals zero. Most state of the art platform like ASIMO [133], HRP-4 [81] and HRP-3 [80] make use of the ZMP concept. One of drawbacks of ZMP arises from the need to have the whole foot in contact with a flat surface, which can be problematic on uneven terrains. Moreover, this method requires a model of the robot and a known environment. Finally, the ZMP can only be controlled indirectly and control methods have to be found indirectly influence

the ZMP so that it stays within the stable region.

Another problem that motivates researcher in their pursuit of a more energy-efficient walking robot, is because a legged robot needs to carry its energy source of the limited capacity. Needless to say, a low energy consumption can directly contribute to a longer working duration. In the paper of [27], the gait of humans is assumed to be one of the most energy-efficient ways of walking. The experimental studies of human locomotion [57, 64] support the hypothesis that the choice of a gait pattern is influenced by energy considerations. In the repetitive motion of walking, the motion controller may attempt to produce walking pattern that is energy efficient because it exploits physical properties of the robot. Bearing this in mind, a possible solution for enabling an energy-efficient walk on the bipedal robot is to:

1. Characterize the biped motion in terms of a set of locomotion variables.
2. Search for the optimal locomotion variables that minimize the fitness function that concerns the energy cost.
3. Establish the correlation between these locomotion variables and the energy cost.

The two core challenges of bipedal walking, namely stability of biped walking in unknown environments and energy efficiency of the biped locomotion have led us to the following problem statement:

*How to synthesize a biped walking gait with energy efficiency and dynamical stability on a humanoid robot with high degrees of freedom on uneven terrains ?*

We identify six research questions which guide the research with regards to the problem statement. The questions address the problems of (1) dynamics model, (2) energy efficiency of locomotion in sagittal and lateral direction, (3) lateral stability, (4) controller design, (5) gait optimization on level ground, (6) gait optimization on uneven terrains.

**Research Question 1:** *Can the planar inverted pendulum model be used to create a bipedal gait with a high energy efficiency?*

**Research Question 2:** *What is the influence of the double support phase on the biped gait in the sagittal plane?*

**Research Question 3:** *What are the requirements of a double support phase with respect to lateral stability of a gait?*

**Research Question 4:** *How to design the controller for gait generation?*

**Research Question 5:** *How to optimize the energy efficiency for the walk gait on flat floor?*

**Research Question 6:** *How to optimize the stability of the walking gait on uneven terrains?*

In Chapter 1, we introduce the reader to the field of biped walking on humanoid robots. Firstly, we introduce motivation of the development of the humanoid robot. In the second section, we discuss the history and state of art in the field of the humanoid robots and the biped walking. We also present a brief overview of the core research challenges in the field of biped walking on humanoid robots. In the fourth section, we introduce and discuss the problem statement and research questions which guide the thesis. In the final section, we introduce the structure and overview of the thesis.

In Chapter 2 we provide general information about Aldebaran Nao, a humanoid robot deployed in this thesis as the experiment platform. Further, we introduce the common techniques that are applied for the purpose of kinematics analysis.

Applicable techniques for biped walk gait generation are introduced in Chapter 3. We first enumerate the different simplified models that are usually employed to describe the dynamics of biped walking. Further, we introduce the stability criteria that are applied for establishing the stability of the system. Additionally, we provide a brief survey of existing approaches for walking gait generation.

Chapter 4 answers the first three research questions. The core problem concerns the energy consumption of the walking gait modeled by a conventional (linear) inverted pendulum and the improvement in the energy efficiency. This walking gait requires stiff joints and bent legs, which result in high energy consumption. We address the research question by investigating an inverted pendulum model with telescopic legs and a leg-length policy that determines the length of the telescopic leg. With this model, we identify the leg-length policy that minimize the energy consumption of a walking gait. Moreover, we make the ankle joint under-actuated by setting its stiffness to zero to further decrease the energy consumption. Our evaluation shows that the application of our approach reduced the energy cost of knee joint in a planar walker. Additionally, we extended the model with a double support phase to improve the stability of the walking and proposed a force policy to control the lateral movement of the CoM. Our method is to manipulate the force on the swing leg by means of a force policy which is generated by regulating the knee stiffness of the swing leg. We also investigated the effect of force policy on the leg-length policy. The experiment showed that the different force policies have no significant influence on the leg-length policy, which indicates force policy is independent and does not influence the behavior of the leg-length policy.

The research questions tackled in Chapter 5 concern the implementation of the controller for gait generation and the optimization with respect to the energy efficiency of the walking gait on a level ground. To answer the question, we propose a controller that provides the appropriate joint functionality consisting of two motion generators, which are supervised by a finite state machine. The controller has 9 control parameters which can be grouped into four categories: (a) sagittal motion (b) lateral motion (c) stability (d) time control. The core problem is that given such a controller, we need a method to find the parameters that minimize the energy consumption and maximize the stability. For this purpose, we use a method of policy gradient reinforcement learning and proposed a fitness function concerning the energy cost and the stability of the walking gait. We use this method to estimate the policy's gradient in the parameter space and then follow the gradient towards an optimum where the value of the fitness function is maximal. We repeated the learning experiment 500 times, each time starting from a randomly gen-

erated parameter. In every experiment the parameters converge to the same optimum. From this result, we conclude that the optimum is likely the global optimum. The evaluation on a real Nao robot shows that the implementation of our approach can result in a walking gait that reduces the energy consumption by 41%, compared to the standard gait of the Nao robot provided by Aldebaran. Lastly, we investigated the stability of the proposed walking gait through its projection of the acceleration vector on the ground, since the stability criterion ZMP is not feasible in our case. Although the ground projection of the acceleration vector overshoots the support point of the new stance leg, the ground projection of the CoM does not and the proposed walking gait can still keep stable and cyclic. From this, we conclude that given the proposed controller with optimized parameters, in the absence of areas of support, our method can synthesis a walking gait that ensure the dynamical stability and energy efficiency on a level ground.

Chapter 6 addresses research question 6, which concerns the stability problem of the walking gait on uneven terrains. We assumed that each step of a robot's walk on an uneven ground can be viewed as a step on a (virtual) slope. In this way, the uneven terrains can be modeled as slopes of variable angles. We used the same method described in Chapter 5 to obtain the optimal control parameters while let the robot walk on a specified slope in both sagittal and lateral direction. The learned parameters and the corresponding information (include the slope angles represented by the height difference of two feet) from the data are used to adapt the gait controller. We presented a feedback controller based on three neural networks that adapts the gait parameters to ensure the robot's stability while walking on an uneven terrain. All of the neural networks have a simple structure with one hidden layer and use Sigmoid as the activation function. The core task is the learning, for which we used the collected data to train the neural networks using back propagation method. The first neural network has four inputs to outputs the sampled angle of the knee joint on the stance leg. The second one has three inputs and its output is the sampled knee stiffness of swing leg. The last neural network is the most simple one, which uses control parameter *Knee Bending* as the input and outputs the control parameter *Torso Roll Inclination*. With the trained networks, we implemented a controller that enables a Nao robot walk on uneven terrains. For our approach, we utilized simulator *Webots* in order to design an evaluation experiment for testing the stability of the generated walking gait. Using the parameter *perlinNOctaves* of the terrain generator in *Webots*, we can generate uneven terrains with specified roughness. Our evaluation revealed that the walking gait generated by the proposed controller is stable on the terrain with hills and high inclination. We provide a video <sup>1</sup> shows the Nao robot walking on an uneven terrain with our proposed gait controller in the *Webots*, which implies that our controller can handle the altitude differences of the foot placement and adjusts the control parameters to maintain balance. Moreover, we evaluated the energy consumption of our walking gait on uneven terrain against the Aldebaran gait on the flat ground, the result indicates that the our gait has better performance not only with respect to the stability but also with respect to the energy efficiency on uneven terrains.

Chapter 7 provides the conclusion of this thesis and discusses possibilities of future research. Taking the answers of the research questions into account, we conclude that

---

<sup>1</sup><https://youtu.be/owHiGQm8WSg>

---

our proposed method can synthesize a biped walking gait with energy efficiency and dynamical stability on a humanoid robot on level ground and uneven terrains.



# VALORIZATION

Traditionally robots are used for manufacturing tasks. Cars, for instance, are manufactured to a great extent by robots. The robots in car manufacturing repeat pre-programmed actions over and over again. More sophisticated robots are used, for instance, in container terminals to pick up containers and bring them to or from the ships. These robots do not execute the same sequence of actions over and over again, but adapt their action based on the goals set by the operators. Container terminals are a restricted environment and the autonomy of the robots is often restricted. Your self-driving car is a robot that has much more autonomy and must operate in an environment with much more uncertainty.

Robots have also appeared in our homes. We have robots that do the vacuum cleaning, mow the grass and entertain the kids. This is only the beginning. In the future, we will have robots that can do more complicated tasks, such as cleaning the toilets, washing and ironing your clothes, making order in the house, and playing football with your kids in the garden. These robots must be adapted to our living environment, which is designed for humans. It is therefore advantageous if the robots have a humanoid form. Such robots much have arms and hands to do tasks we humans are doing, and legs to maneuver in our environment.

A robot that has to maneuver in our environment must be able to walk on different surfaces, such as flat wooden or stone floors, (high) pile carpet, the grass in the garden, cobblestones street, etc. It must be able to step over obstacles such as children's toys on the floor, and it must be able to handle unexpected disturbances.

Besides being able to maneuver in our environment, a humanoid robot must also be energy efficient when walking in our environment. A humanoid robot has to carry its own power source and the capacity of modern batteries is still limited. Compared to the human walking gait, the walking gait of the humanoid robot is much less efficient. The difference in energy efficiency is partly caused by hardware differences such as using motors instead of muscles, but is also caused by differences in walking patterns. The former differences require developments in hardware while the latter differences can be addressed by developing intelligent controllers.

The research presented in this thesis contributes to both walking on different and possibly uneven surfaces, and to improving the efficiency of a walking gait. The results show that energy efficient walking gait of the humanoid Nao robot developed by Aldebaran can be improved by 41 % while walking on even grounds. Moreover, the energy efficient gait can also handle uneven terrains with bumps, holes and slopes. These results are not limited to the Nao robot. The proposed gait controller can be applied on any robot with a similar design, such as ASIMO developed by HONDA, the Wabian series developed by Waseda University, the HRP series built by Japanese National Institute of Advanced Industrial Science and Technology (AIST), HUBO2 from Korea Advanced Institute of Science and Technology (KAIST), Atlas developed by Boston Dynamic, the robot

H6 and H7 from University of Tokyo, Jonnie and Lola from the Technical University of Munich, and Sony's Qrio. What these robots have in common are the flat feet, which do not allow for toe-off, and full control over all degrees of freedom. Fine-tuning the gait parameters of the proposed gait controller for these specific robots using reinforcement learning, see Chapter 5.

The proposed gait controller uses under-actuated ankle joints. The walking gait should therefore also be applicable to robots with point feet, such as the RABBIT built as part of the French National Project ROBIA, MABEL developed at the University of Michigan, MARLO from Dynamic Legged Locomotion Lab at Oregon State University and HUME developed by UT Austin. For these robots the controller must be extended to handle the robot standing still since the absence of feet introduces a stability issue when the robot stops walking.

The approach developed in this thesis is applicable to the new development in humanoid robotics. The proposed approach identifies an energy efficient gait for robots where the torque on the joints is the main source of energy consumption. The energy efficiency of a robot can be improved if actuators are developed and used in which the work done and not torque is the main source of energy consumption. The approach described in Chapter 4 to identify an energy efficient gait can also be applied when better actuators come available. The same holds for developments in hardware design. Walking with feet that can toe-off is considered to be more energy efficient. The WABIAN-2R from Waseda University and the Romeo from SoftBank (Aldebaran) are robots that have feet that can toe-off. With a small adaptation, the approach described in Chapter 4 can also identify the most energy efficient gait for these robots. The gaits identified may and probably will differ from the gait identified in this thesis. However, a controller can be developed in the same way as described in Chapter 5.

The whole approach outlined in this thesis may help people with walking disabilities in two different ways. First, we may build, physically or in a simulator, robots that can model certain walking disabilities. The approach described in the thesis can be used as a tool to investigate possible walking gaits given the disability. Second, the approach described in the thesis may be used in the development of exoskeleton for paralyzed people. Creating a stable gait that naturally fits with the remaining abilities of the paralyzed person is an important issue.



# ACKNOWLEDGEMENTS

This thesis is the final result of my PhD research in the Department of Data Science and Knowledge Engineering of the Faculty of Science and Engineering in Maastricht University. I am sincerely grateful to my promotor Professor Ralf Peeters. His inspiration, guidance, and all productive discussion throughout this research greatly enlightened me. Our discussion over the structure and outline of the thesis, and his careful proof-reading regarding the early manuscripts helped me rethink the architecture of this thesis and improve the manuscript substantially. My deep gratitude and appreciation go to my co-promotor Dr. Nico Roos, who led me to this beautiful country, kindly shared with me his valuable research experience and provided me with vast opportunities to practice my presentation and writing skills. Without his patience, guidance and support, I could never have finished this thesis. He has always been a source of inspiration and humor, a true friend and a mentor to me. He offered me the considerable freedom in testifying my thoughts, tolerated all my sudden ideas and requests, and had all the wonderful moments that we have spent in discussing the research topic and hilarious life stories.

I would like to express my gratitude to the members of assessment committee who devoted their time to read my thesis and provided valuable feedback for future improvement. Prof. Gerhard Weiss, Prof. Koen Hindriks, Dr. Kenneth Meijer, Prof. Frank Thuijsman and Dr. Arnoud Visser, it is my honor to get the approval on my thesis from you.

I would also thank my DKE officemate, Wenzhao Qiao, Frederick Schaad, Sichi Chen, Wei Zhao, Bijan Ranjbarsahraei, Li You, Katharina Schüller, thanks for your shared stories making our office a wonderful place to work. I shared a lot of pleasures with my colleagues and friends at the Department of Data Science and Knowledge Engineering, specially to: Ronald Westra, Jet Custers, Pietro Bonizzi, Katerina Stankova, Gijs Schoenmakers, Rachel Cavill, Pieter Collins, Kurt Driessens, Jöel Karel, Rico Möckel, Jerry Spanakis, Evgueni Smirnov, Daan Bloembergen, Hendrik Baier, Nela Lekia, Daniel Claes, Haitham Bou Ammar, Michael Kaisers, Pim Nijssen, Firat Ismailoglu, Nasser Davarzani, Shuang Zhou, Chiara Sironi, Hua Zhang, Yiyong Gou, Thomas Christos and so on. I benefited greatly from our discussions which have been a highlight during my time as a PhD. I have a long list of names to thank who made my stay in Maastricht particularly fun and exciting. They are: Professor Michael Faure, Hui Wang, Jing Liu, Jun Hou, Xiahong Chen, Tianxiang He, Jie Wang, Yaojin Peng, Tian Lv, Mengxing Lv, Mengmeng Shi, Taotao Yue, Jiangqiu Ge, Ancui Liu, Yu Zhao and Xi Chen, Ai Li, Fei Sun, Ling Cao, Kuan Jiang, Yimiao Wang, Sheng Zhong, Min Lin, Caixia Liu and so on. I was so lucky to know a large group of friends, thanks to who I did not feel lonely even far from home.

I want to express my special gratitude to the China Scholarship Council (CSC) and the Chinese Embassy in the Netherlands. Thanks to the funding provided by CSC and support from the Chinese Embassy, I could have this precious opportunity to come to Europe and finish my PhD research. In addition, I want to thank the Department of Data Science and Knowledge Engineering of Maastricht University, which has provide me a

bright office, fresh coffee, enormous seminars and workshops and most importantly the chance to defend my thesis.

Last but certainly not least, I would like to thank some very important people in my life. I would like to give my special thanks to Wenqing Liao, who has always supported me and been standing by my side during the whole life-changing experience of all these years in the Netherlands. Her unconditional support is the reason that I was able to devote myself to a PhD research during these years. Without her I would not have done my PhD. My love also goes to my parent Bin Sun and Caixia Qiao for their distant love that always followed me while I was far away. I dedicate my thesis to them, for their endless love, support and care.

Maastricht, 15 February 2019

## ABOUT THE AUTHOR

Zhenglong Sun was born on June 10th, 1984 in Luoyang, China. He graduated from University of Electronic Science and Technology of China with B.Sc. in Computer Science and Engineering at August 2006 and MSc. in Information Security at August 2010.

From September 2011, he moved to Maastricht and started a Ph.D. project at the Department of Data Science and Knowledge Engineering in Maastricht University. In this project, In this project, he was responsible for designing a walking gait generator for a commercial humanoid robot, which enables it to walk on the level ground and the uneven terrains. His Ph.D. research resulted in several publication on various conferences and journals, which are included in this dissertation.





# LIST OF PUBLICATIONS

1. **Z. Sun**, and N. Roos: An energy efficient gait for a Nao robot," in BNAIC 2013: *Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands*, pp. 323-329, 2013. Delft University of Technology (TU Delft).
2. **Z. Sun**, and N. Roos: Dynamic Lateral Stability for an Energy Efficient Gait," in BNAIC 2014: *Proceedings of the 26th Benelux Conference on Artificial Intelligence, Nijmegen, the Netherlands*, 2013, pp. 234-242. Radboud University.
3. **Z. Sun**, and N. Roos: An energy efficient dynamic gait for a Nao robot," in *Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2014, pp. 267-272.
4. **Z. Sun**, and N. Roos: A Controller for Improving Lateral Stability in a Dynamically Stable Gait," in *AIAI : 12th IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2016. pp. 371-383. Springer, Cham.
5. **Z. Sun**, and N. Roos: "Dynamically stable walk control of biped humanoid on uneven and inclined terrain," *Neurocomputing*, 2018, vol. 280, pp. 111-122. Elsevier.
6. **Z. Sun**, and N. Roos: A Dynamically Stable Gait Without Support Polygons, *Autonomous Robots*, accepted with revisions, 2017.