Institut für Roboterforschung

technische universität dortmund

Motion Plan and Control of Humanoid Walking Robots

Bei der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität Dortmund
vorgelegte

Dissertation

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

von

M. Sc. Jing Liu

Dortmund, 2015

Hauptreferent: Prof. Dr.-Ing. Uwe Schwiegelshohn
Korreferent: Prof. Dr. phil. André Seyfarth

Prüfungskommission:

Hauptreferent:	Prof. Dr.-Ing. Uwe Schwiegelshohn
Korreferent:	Prof. Dr. phil. André Seyfarth
Prüfungsvorsitzender:	Prof. Dr.-Ing. Christian Rehtanz
Mitprüfer:	apl. Prof. Dr. rer. nat. Frank Hoffmann

Tag der mündlichen Prüfung: 15. Dezember 2015

Dissertationsort: Technische Universität Dortmund, Dortmund

Abstract

Walking for most humans and animals is an easy task due to the inherent robustness and the natural dynamics of the walking mechanism. Walking, however, for humanoid robots is not that easy because of its nonlinearity, high dimensionality, and natural instability.

Effective use of humanoid robots in unstructured environments for human beings requires that they have autonomous and reliable locomotion capability. Locomotion for humanoid robots can take many forms. This thesis covers motion plan and control of humanoid walking robots.

Five consecutive stages are addressed from the perspective of stable dynamic walking. Firstly, a natural and efficient walking pattern is designed on the basis of the insight gained from human walking. The walking pattern involves the configuration with stretched knees. Secondly, in view of the modeling error, as well as the environment uncertainty such as the unevenness and inclination of the surface, a posture controller which online controls the orientation of the upper body of the robot is developed from the viewpoint of stability. Walking stability of the robot is improved through the control scheme. In the third stage, a gait controller which online controls the swing-leg is developed. Fast and precise trajectory tracking of the swing-leg is achieved which enables humanoid robots to quickly swing the swing-leg so that fast walking can be realized. In the fourth stage, a controller with the purpose of decreasing the landing force and improving the walking stability by means of force control is developed. In the final stage, strategies for restoring equilibrium in the presence of external disturbances to maintain upright standing posture are analyzed. Simple models are introduced to exploit boundaries that determine the strategies used for preventing a fall.

Simulation and experiments were performed with the humanoid robot NAO developed at Aldebaran Robotics in France. Using the walking system, NAO achieved dynamic stable walking and reached a maximum walking speed of 0.24 m/s in experiments. Besides, with the presented walking pattern NAO could walk stably with almost stretched knees at a lower speed, e.g. 0.10 m/s, so that the corresponding energy consumption was lower since the robot did not bend the knees all the time, which means that the robot is able to walk for a longer time without battery charging and has less heating problem.

Thesis Supervisor: Uwe Schwiegelshohn
Title: Professor, Department of Electrical Engineering and Information Technology, Technische Universität Dortmund

Acknowledgements

Over the past years, I have had the fortunate opportunity to pursue an investigation of a topic that has interested me since I was a college student: control of a biped humanoid robot. This exciting area of research is interdisciplinary combining a number of different fields such as robotics, bipedal locomotion, control theory, and biomechanics.

First, I would like to thank my supervisor, Uwe Schwiegelshohn, for giving me the opportunity to engage in such an interesting and challenging research, for giving me the freedom to pursue my own ideas, and for providing guidance and advice throughout my work at the Robotics Research Institute. I feel grateful for the understanding, support, and respect Uwe has given me.

I appreciate the members at the Robotics Research Institute who provided crucial help over the years. I am especially thankful to Oliver Urbann, who has wealth of experience with the framework of the humanoid robot NAO, for his invaluable help and suggestions. Discussion with him was inspiring and productive. I would like to thank Stefan Tasse who gave me suggestions on carrying out the experiments and making comparison. I thank Matthias Hofmann and Ingmar Schwarz for their help on my work. I would like to express my gratitude to Jörg Platte, Kamol Limtanyakul, Matthias Hebbel and Walter Nistico who gave me much help on my early-stage research. I am deeply grateful for having had the chance to work with these very talented and highly motivated people. Thanks to Peter Resch and René Schubert for technical support. I would also like to thank Sabine Winterhoff for her help on my work and life.

I am very grateful to the members in the Graduate School of Production Engineering and Logistics in TU Dortmund University, Meni Syrou, Gundula Plaep, Veronika Christina Siedlaczek and Lina Schmitz, for their enthusiastic help and support.

I would like to thank my Chinese friends for their suggestions on my research and help in my daily life, and for all the fond memories.

I thank my family for constant support and encouragement. My daughter makes my life full of joy and happiness. I thank my husband for his consideration, understanding, and great support on my research. I am very grateful to my parents-in-law due to their help and support. I give my appreciation to my parents for their raise and cultivation. My father always encourages me to keep a peaceful state of mind in the face of difficulties. My mother told me to treat science and knowledge scrupulously the first day I went to school. Her words have had a great influence throughout my lifetime.

This thesis is dedicated to my beloved mother.

Contents

List of Figures

List of Tables

List of Abbreviations

AMPM	Angular Momentum inducing inverted Pendulum Model
BoS	Base of Support
CAN	Controller Area Network
CMP	Centroidal Moment Pivot
CoG	Center of Gravity
CoM	Center of Mass
CoP	Center of Pressure
CPG	Central Pattern Generator
DoF	Degree of Freedom
DSP	Double-Support Phase
FCS	Foot Coordinate System
FRI	Foot-Rotation Indicator
FSR	Force Sensitive Resistor
GCoM	Ground projection of Center of Mass
GRF	Ground Reaction Force
HAT	Head, Arm and Trunk
IMU	Inertial Measurement Unit
LIPM	Linear Inverted Pendulum Mode
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
ODE$^{(1)}$	Open Dynamics Engine
ODE$^{(2)}$	Ordinary Differential Equation

PD	Proportional-Derivative
QP	Quadratic Programming
RCS	Robot Coordinate System
ROM	Range of Motion
SSP	Single-Support Phase
ZMP	Zero-Moment Point
ZRAM	Zero Rate of change of Angular Momentum

1 Introduction

A humanoid robot, generally having a torso, a head, two arms, and two legs, is a robot with its body shape built to resemble that of the human body. During the past three decades research and development in robotics have expanded from traditional industrial robot manipulators to autonomous, wheeled or legged robots. Bipedal humanoid robots essentially differ from other kinds of robots such as industrial ones in that their movements are human-like using biped gait. One of the key issues for humanoid robots is autonomous and reliable bipedal locomotion. Types of bipedal locomotion include walking, running, or hopping. Among these, bipedal walking has been studied extensively for better understanding and rehabilitating human locomotion, as well as replicating human locomotion in machines. Except for passive dynamic walking [76] explored by McGeer which utilizes the robot natural dynamic, e.g. link lengths, link masses, joint damping, walking surface slope, foot shape, to maintain stability, bipedal walking is generally mentioned as powered bipedal walking. The most common issues related to powered biped walking are calculating theoretically stable walking patterns either online or offline, and designing the appropriate online controllers in order to achieve stable walking.

This thesis focuses on planning human-inspired bipedal walking motion, and controlling posture and gait of a biped humanoid robot. The goal of this thesis is to realize stable dynamic walking of the biped humanoid robot by means of both position control and force control approaches.

1.1 Motivation

Research on humanoid robots is one of the topics that catch comprehensive attention. Humanoid robots are being developed nowadays to further understand human locomotion, to coexist and collaborate with humans or subsitute humans to perform tasks, to assist in the welfare or medical field, and to provide entertainment.

Being as a Research Tool

Building and controlling humanoid walking robots can help us understand how humans walk. The walking process of humans was first studied as far back as the Second World War in the effort of developing artificial legs for disabled soldiers. Despite continuous research on the subject since then, we have not yet fully understood the underlying basic principles of human walking due to the high complexity of human musculo-skeletal system [71]. The symbiosis between controlling robots and understanding humans can be achieved by using a robot to test theories on how humans walk, to further understand the biomechanics of both robots

and humans, to discover limitations to walking in both humans and robots, and to suggest testable hypotheses on how humans control walking [97].

Performing Human Tasks

Biped humanoid robots are able to well locomote on various terrains such as uneven surfaces or stairs. So they have better adaptability to the environment where humans live. They are capable of sharing the same space with humans and require no changes to the existing infrastructure. Since humanoid robots have the capability to use tools and operate equipments designed for human beings, they could theoretically perform any task a human being can. Humanoid robots could be used for missions too dangerous or difficult for human beings like disaster relief, bomb disarming, planetary exploration.

Besides, because of the anthropomorphic appearance and movement, humanoid robots enhance the chances of acceptance by human beings and simplify human-machine communication. Therefore, they have great potential in the service area, for example, providing assistance and companion to the sick, the handicapped and the elderly.

Assisting in the Welfare or Medical Field

The aging population is growing dramatically in America and Japan in recent years, and the trends are similar in other countries in developed regions of the world. A significant percentage of the current aging population suffer from lower-limb disabilities [28, 85]. Biped humanoid robots can be used as a tool to establish a human walking model in order to facilitate the development of welfare and rehabilitation instruments such as walking machines for assistance or training, and estimate the effectiveness of such machines.

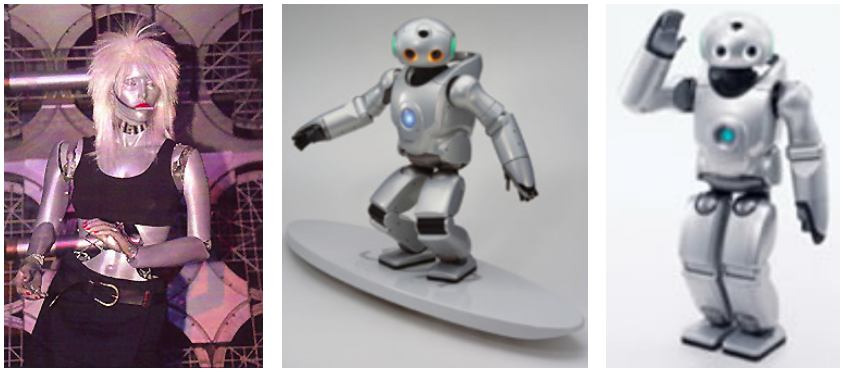
In addition, one of the aims of humanoid research is to build better orthosis and prosthesis for human beings. Robotic bipedal systems can be simulated as a means to understand cause-effect relationships between neuromuscular control mechanisms and the resulting movement in human models, and to optimize orthotic and prosthetic designs in order to enhance their performance and interaction with the user.

Providing Entertainment

Humanoid robots are becoming increasingly popular in the entertainment industry, as shown in Figure 1.1. Ursula[1], an entertainment robot developed by Florida Robotics to amuse crowds at Universal Studios, is a remote-controlled full-size female robot that can walk, dance, play music and speak to her audiences. Sony has developed small but remarkable robots called SDR-4X[2] and SDR-4X II[2] that can dance and sing for the entertainment purpose.

Besides the anticipated demand for humanoid robots, recent technology advances such as autonomous task-level planning and control systems, the development of sophisticated non-linear control algorithms and the development of a new class of actuators, also make the investigation of control of humanoid machines possible.

[1] Copyright©: Florida Robotics.
[2] Copyright©: Sony Corporation.



(a) Ursula (b) SDR-4X (c) SDR-4X II

Figure 1.1: Humanoid robots for the purpose of entertainment. Sources: (a) Ursula. http://www.floridarobotics.com/ursula.htm; (b) and (c) Product & Technology Milestones - Robot. http://www.sony.net/SonyInfo/CorporateInfo/History/sonyhistory-j.html.

1.2 Problem Statement

Biped humanoid walking, when viewed as a general dynamical system, is difficult.

First, a humanoid is a high-dimensional, nonlinear and hybrid system. A humanoid robot has multiple articulated joints, and therefore, has a large number of degrees of freedom (DoFs). For example, the humanoid robot used in this thesis has 25 DoFs. Apart from this, dynamics of a humanoid are highly nonlinear and tightly coupled. Furthermore, a humanoid is a hybrid system. The dynamic equations are not continuous since the dynamics change as the humanoid robot interchanges between the single support and double support, whereas physical walking variables such as the joint trajectory, the position and velocity of the center of mass, are continuous with respect to time. As a result, it is challenging to implement accurate control on a humanoid robot.

Second, bipedal humanoid robots are regarded as one of the two types of legged robots that are most prone to instabilities[3][16]. A humanoid robot is inherently unstable since the center of mass is high compared with the limited size of the base of support, and is therefore very sensitive to external disturbances. So the overall system is likely to tip over about one of the foot edges, especially in the single support phase, small external disturbances are sometimes sufficient to make a robot fall.

Third, a humanoid robot realizes walking via joints and links, while a human being walks by means of muscles, bones and joints. The joint of a humanoid robot is composed of motors and gears and has mechanical limitations. The hip and ankle joints of the humanoid robot used in this thesis can deliver a maximum continuous torque of 16.1 mNm. It is insufficient to recover the robot from perturbations by using hip or/and ankle joints. By contrast, the hip torque a human being utilizes to recover from perturbations while standing or walking is about 40 Nm to 50 Nm [75, 125]. Therefore, the dynamic performance of a humanoid robot is limited.

In a word, it is challenging to control a humanoid robot to achieve stable dynamic walking especially in the presence of external disturbances, such as upper body motions, unevenness

[3]The other type is monopods.

of the surface, external forces, etc.

1.3 Control Types

There have been many approaches to the problem of controlling humanoid robots. Two of the most common approaches can generally be described as position control and force control, resulting in distinct design of humanoid robots and separate style of control. Advantages and disadvantages of these two approaches are analyzed below.

1.3.1 Position Control

Benjamin Stephens in his PhD thesis [106] pointed out that robots with position control, such as the one depicted in Figure 1.2, are generally constructed using electric motors with high gear ratios. This allows for smaller, cheaper motors to operate efficiently while providing enough torque. Controllers for these systems generally exploit this feature by creating walking patterns that can be tracked very accurately. This effectively simplifies walking control to footstep planning, for example.

The high reflected inertia and friction created by the gearbox creates a high impedance which can be useful for rejecting small disturbances. However, the performance of force control which is useful for handling larger disturbances is poor due to the high joint impedance and low control bandwidth; besides, the interaction between the robot and the environment is not sufficiently considered in the position control approach.

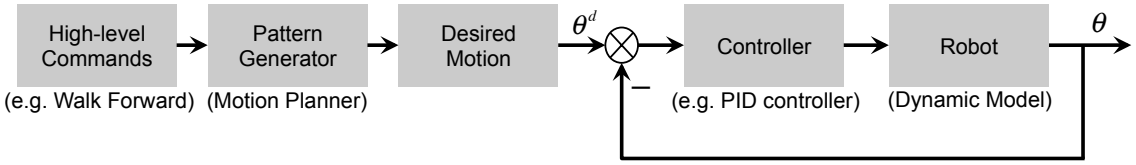


Figure 1.2: Position control.

1.3.2 Force Control

Force control was originally studied for the precise control of industrial robot manipulators. The concept of force control is that the end-effector of a robot manipulator is expected to have a suitable compliant behavior in the workspace (e.g. Cartesian space) in the situation where the environment continuously exerts a dynamic or kinematic constraint on the manipulator's motion. As opposed to position-controlled robots which have high stiffness, robots with force control have lower stiffness, and are compliant and robust to force disturbances.

The dynamic robot-environment interaction of bipedal walking is realized through the robot foot. Thus the biped locomotion, if viewed the foot as the end-effector, is essentially equivalent to the robot manipulators in a sense of force control. For control purposes the entire dynamics of the humanoid robot and the environment should be treated as a whole. Methodologies aiming at pure joint trajectory control or pure position control are not appropriate for such a system. Rather, control methods governing not only foot motion but contact forces between the foot and the ground should be used for precisely controlling this type of system even when there are uncertainties and variations.

Two approaches have been suggested for assuring compliant motion [55]. The first approach is with the aim of controlling the force/torque and position/orientation in a non-conflicting way. Hybrid position/force control proposed by Railbert and Craig [11, 98] decomposes the task space into two subspaces, position-controlled subspace and force-controlled subspace (as illustrated in Figure 1.3). A selection matrix S is used to determine which degrees of freedom are to be position-controlled, and which are to be force-controlled. Afterwards, Khatib proposed hybrid position/force control in Cartesian space [57] by using the appropriate transformation and describing the manipulator dynamics in Cartesian space. Hybrid position/force control is a straightforward approach to the problem of controlling the force and position simultaneously.

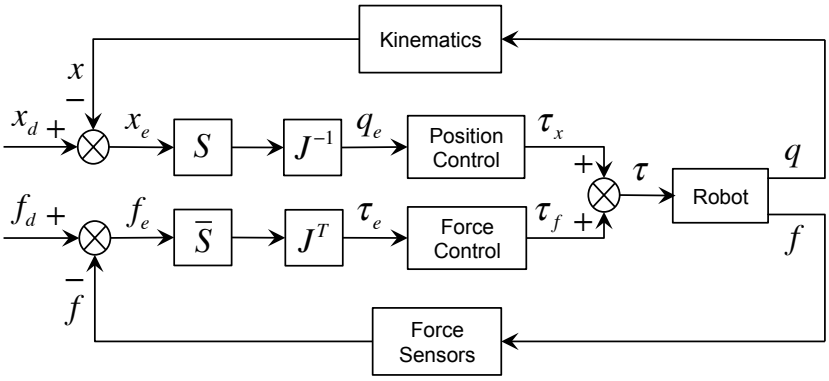


Figure 1.3: Hybrid position/force control, as proposed in [11]. J is the Jacobian matrix, S, $\overline{S}$ denote respectively the selection matrices for position and force controlled degrees of freedom, x_d is the desired motion trajectory, x is the task-space position, f_d is the desired force trajectory and f denotes the contact force.

The second approach is aimed at developing a relationship between the interaction force and end-effector position. Impedance control put forward by Hogan [29] regulates the end-effector position and its dynamic relationship with the force, rather than controlling these variables separately. Therefore, impedance control provides a unified approach for both free motion and constrained motion. A more detailed review of impedance control can be found in [72].

Force control approach is in general implemented on force/torque-controllable robots. The disadvantage of force control, however, is that designing such a controller can be difficult given the complex dynamics of the robot.

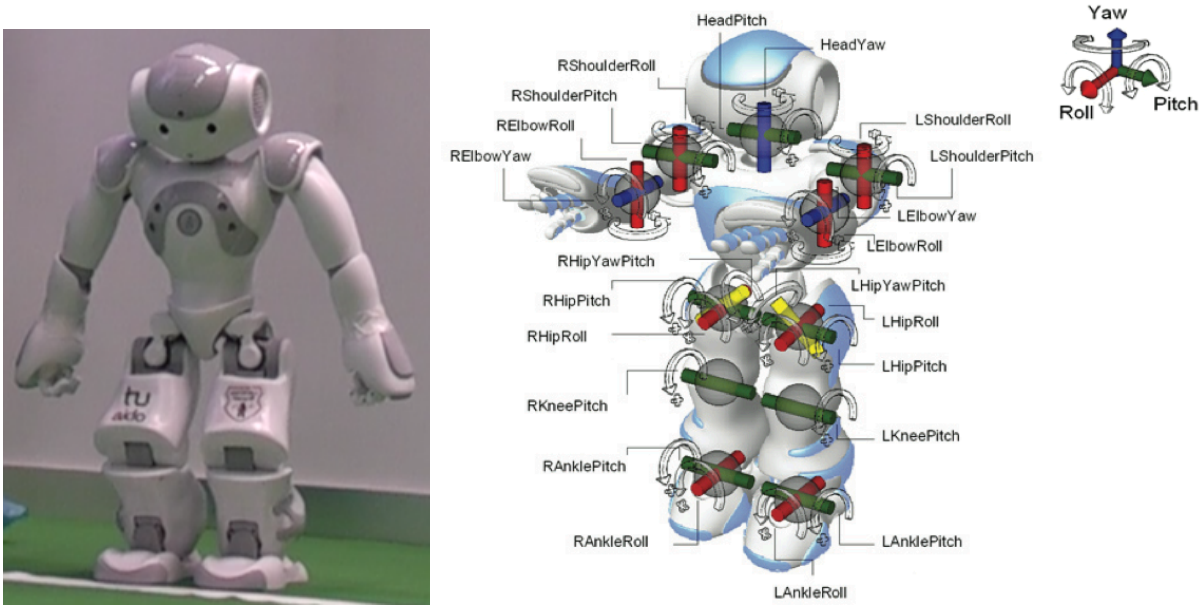
This thesis combines the advantages of both position control and force control. The robot can precisely track the reference walking pattern when the leg is in swing phase. It can also appear compliance while interacting with the ground.

1.4 Humanoid Robot Platform

NAO (Figure 1.4(a)), an integrated, programmable, medium-sized humanoid robot developed by Aldebaran Robotics[4] in France, has been used in this thesis as the research platform to

[4]Company Website for the Robot NAO. http://www.aldebaran.com, 2014.

study motion planning and develop control algorithms for humanoid walking robots.



(a) (b)

Figure 1.4: Humanoid robot NAO: (a) photograph (b) joint structure [2].

1.4.1 Specifications

The version of NAO we worked on is H25 v4.0. It has five kinematic chains (head, two arms and two legs). Its height and weight are 57.3 cm and 5.2 kg, respectively. It has in all 25 DoFs, five for each leg, five for each arm, one for each hand, one for the pelvis, and two for the head. Each joint is actuated by a brushed DC coreless motor with a reduction gear. Encoders on all servos record the actual values of all joints in real time. More basic technical data of NAO is listed in Appendix A.

1.4.2 Ranges of Motion

Figure 1.4(b) illustrates the joint structure of NAO. Table 1.1 shows the ranges of motion (ROMs) of the joints of NAO and a standard human being. It is worth noting that NAO does not have a waist joint. Indeed, the waist joint plays an important role in planning motion and improving stability of a humanoid robot. The two-DoF waist joint enables HRP-2 to get up itself [50]. HRP-2 is also able to crawl on hands and knees using the waist joint. The moment generated about the yaw axis can be compensated by using waist motion of a humanoid robot [127], therefore, the walking stability is improved. In addition, Huang *et al.* utilized the waist joint of a biped robot to enlarge the stability margin [32]. The extra DoFs in the upper body can also make the walking gait smoother [51]. Furthermore, the waist joint allows to increase step length and provides more mobility for lateral motions [122], as well as extends the working space of arms.

Table 1.1: ROMs of joints of NAO and a standard human being.

Joint			Human Being [1, 18]	NAO
Head		Pitch	-55° to $(70^\circ \sim 90^\circ)$	-38.5° to 29.5°
		Roll	-45° to 45°	no existence
		Yaw	-70° to 70°	-119.5° to 119.5°
Right Arm	Shoulder	Pitch	-90° to 140°	-119.5° to 119.5°
		Roll	-180° to 50°	-76° to 18°
		Yaw	-45° to 130°	no existence
	Elbow	Roll	0° to 140°	2° to 88.5°
		Yaw	-90° to 90°	-119.5° to 119.5°
	Wrist	Pitch	-60° to 60°	no existence
		Yaw	$(-30^\circ \sim -50^\circ)$ to 20°	-104.5° to 104.5°
Right Hand			open and close	open and close
Waist		Pitch	-30° to 75°	no existence
		Roll	-35° to 35°	no existence
		Yaw	-30° to 30°	no existence
Right Leg	Hip	Pitch	-120° to 30°	-101.54° to 27.82°
		Roll	$(-45^\circ \sim -50^\circ)$ to $(20^\circ \sim 30^\circ)$	-42.30° to 23.76°
		Yaw	-40° to 45°	no existence
	Knee	Pitch	0° to 150°	-5.90° to 121.47°
	Ankle	Pitch	-20° to 50°	-67.97° to 53.40°
		Roll	no existence	-45.03° to 22.27°
		Yaw	-20° to 30°	no existence

[1] The convention of the moving direction of the revolute joints is in accordance with that in NAO system.
[2] NAO has two HipYawPitch joints that are controlled by one motor.

1.5 Thesis Contributions

The contributions of this thesis are:

1. The design of a natural, human-like walking pattern on a level surface.

2. The development, in simulation and in experiments, of postural control algorithms which realize stable dynamic walking of a humanoid robot on a level surface.

3. The realization, in simulation, of adaptive control in precise trajectory tracking and dynamic parameter estimate of the swing-leg of the robot.

4. The proposal of a force control approach to decrease the landing impact force.

5. The analysis of the strategies for restoring balance in the presence of external disturbances.

The outcome of this research shows the significant effect of adding upward and downward motion of the upper body of the robot in the sagittal plane on reducing the energy consumption when the robot walks at a lower speed. It verifies the effectiveness of the posture controller in maintaining a constant upper body posture against environment uncertainty and thus achieving stable dynamic walking of the robot. It also demonstrates the advantage of adaptive control over the proportional-derivative (PD) control in accurate trajectory tracking. This research paves the way toward a better understanding of robot walking and balance.

1.6 Thesis Outline

This thesis proceeds as follows:

Chapter 2 gives background for this thesis, consisting of providing the stability criteria which are used to judge whether the walking system is stable or not, and surveying the related works with regard to bipedal humanoid robots.

Chapter 3 proposes a motion generation method for humanoid robots motivated by biomechanical studies on human walking. The proposed walking pattern involves a three-dimensional motion of the upper body of the robot which is more natural and human-like.

Chapter 4 proposes two control methods to manipulate the upper body orientation of humanoid robots aiming at stable dynamic walking. A torso pitch/roll controller utilizes internal joints to stabilize the robot based on sensory feedback. A yaw moment controller counteracts yaw moment about the support foot using arm movement.

Chapter 5 presents the application of adaptive control to precise trajectory tracking and dynamic parameter estimate of the swing-leg of a humanoid robot. Simulation based on both Matlab/SimMechanics and Open Dynamics Engine (ODE$^{(1)}$)[5] is implemented.

Chapter 6 proposes a control method based on the force control approach in order for decreasing the landing force and improving the stability of the walking system.

Chapter 7 analyzes strategies for restoring equilibrium and maintaining upright standing posture in the presence of external disturbances. Simple models are introduced to exploit boundaries that determine the strategy used for preventing a fall.

Chapter 8 concludes the thesis and puts forward suggestions for future research.

Appendix A gives the basic technical data of the humanoid robot used in this thesis.

Appendix B gives the forward kinematics of the five chains of the robot. The forward kinematics, which is used to online calculate the walking distance of the robot in Chapter 4, is also given.

Appendix C derives the dynamic equations of the three-link leg which are used for the adaptive control method in Chapter 5.

Appendix D gives the Jacobian matrix for the force control in Chapter 6.

[5] http://www.ode.org/.

2 Background

In this chapter we review the existing stability criteria for system stability analysis, the simplified models that can generally describe humanoid dynamics, as well as the previous related works on humanoid robots.

Stability criteria, reviewed in Section 2.1, are notions used to ensure the dynamic stability of a walking robot. Simplified models, reviewed in Section 2.2, simplify humanoid dynamics and facilitate motion design and control of a humanoid robot. There are many biped humanoid robots, some of which will be reviewed in Section 2.3. The existence of these robots and some of their control techniques have motivated much of the work and the approach followed in this thesis.

2.1 Stability Criteria

The concept of *support pattern* (*support polygon*), first put forward by HILDEBRAND [25], is a convenient way to show each step of a gait cycle. The support polygon is defined as the area on the ground which creates a bounding box around all of the contacts between the feet and the ground.

Locomotion researchers distinguish between gaits that are statically stable and gaits that are dynamically stable.

A biped robot gait is said to be statically stable and a human posture is said to be balanced if the ground projection of its center of mass (GCoM) falls within the support polygon [16]. Early biped walking of robots involved static walking [52, 53, 115]. The step time was over 10 seconds per step and the balance control strategy was based on control of the center of gravity (CoG). The disadvantage of static walking is that the motion is slow. For example, one walking cycle of the biped robot WAP-3 on level ground required about 1 minute and 30 seconds [52].

Researchers thus began to focus on dynamic walking of biped robots. It is fast walking and the step time is less than 1 second per step. However, if the inertial forces generated from the acceleration of the robot body are not suitably controlled, a biped robot easily falls down [59].

Several notions are introduced in order to control inertial forces and verify system dynamic stability. They are the Zero-Moment Point (ZMP) [123], Foot-Rotation Indicator (FRI) point [16], Centroidal Moment Pivot (CMP) [95] or Zero Rate of change of Angular Momentum (ZRAM) point [17], Valid Stable Region [34], respectively.

2.1.1 Zero-Moment Point

Vukobratović *et al.* in [123] give an interpretation of ZMP concept. ZMP is defined as that point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes. ZMP notion considers the locomotion mechanism in the single support phase, with the whole foot being on the ground.

In [123], to facilitate the analysis, the part of the mechanism above the ankle of the support foot (point A) can be neglected and its influence can be replaced by the force $\boldsymbol{F_A}$ and moment $\boldsymbol{M_A}$ (refer to Figure 2.1). The weight of the foot itself acts at its gravity center (point G). The foot also experiences the ground reaction force at point P. In general, the total ground reaction consists of the force $\boldsymbol{R}$ (R_x, R_y, R_z) and moment $\boldsymbol{M}$ (M_x, M_y, M_z). The necessary and sufficient condition in dynamic equilibrium is that for the point P on the sole, $M_x = 0$ and $M_y = 0$. Since both components relevant to the realization of dynamic balance are equal to zero, the point P is Zero-Moment Point. The static equilibrium equations for the supporting foot are as follows:

$$\begin{cases} \boldsymbol{R} + \boldsymbol{F}_A + m_s\boldsymbol{g} = 0 \\ \overrightarrow{OP} \times \boldsymbol{R} + \overrightarrow{OG} \times m_s\boldsymbol{g} + \boldsymbol{M}_A + M_Z + \overrightarrow{OA} \times \boldsymbol{F}_A = 0 \end{cases} \tag{2.1}$$

where $\overrightarrow{OP}$, $\overrightarrow{OG}$ and $\overrightarrow{OA}$ are radius vectors from the origin of the coordinate system O_{xyz} to the point P, point G, and ankle joint point A, respectively, while the foot mass is m_s and $\boldsymbol{g}$ is the gravity acceleration.

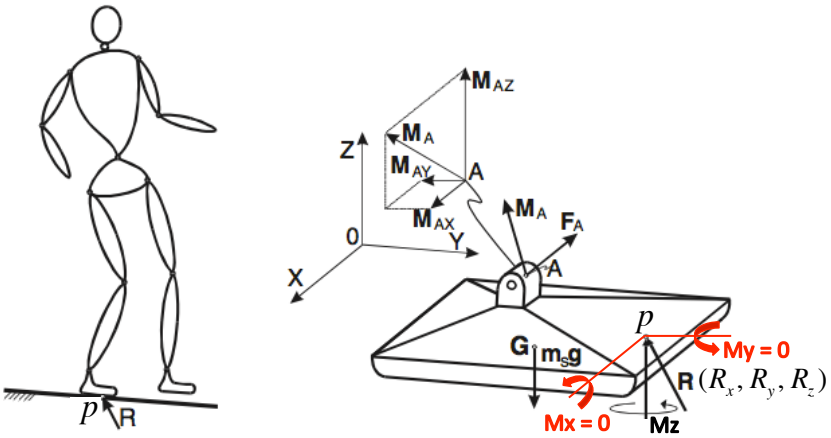


Figure 2.1: Zero-Moment Point notion from [123].

If the robot is treated with detailed information of body segment dynamics, the interpretation can be expressed as

$$\left(\sum_{i=1}^{n} (\boldsymbol{p}_i - \boldsymbol{p}_{zmp}) \times m_i\boldsymbol{a}_i + \sum_{i=1}^{n} \frac{d(\boldsymbol{I}_i\boldsymbol{\omega}_i)}{dt} \right)_{horizontal} = \left(\sum_{i=1}^{n} (\boldsymbol{p}_i - \boldsymbol{p}_{zmp}) \times m_i\boldsymbol{g} \right)_{horizontal} \tag{2.2}$$

where

$\boldsymbol{p}_i = [x_i, y_i, z_i]^T$	the center of mass (CoM) position of the i^{th} segment
$\boldsymbol{p}_{zmp} = [x_{zmp}, y_{zmp}, z_{zmp}]^T$	position of the ZMP
$\boldsymbol{a}_i = [\ddot{x}_i, \ddot{y}_i, \ddot{z}_i]^T$	acceleration of the CoM of the i^{th} segment
$\boldsymbol{I}_i$	inertial tensor of the i^{th} segment about the segment's CoM
$\boldsymbol{\omega}_i$	angular velocity of the i^{th} segment
$\boldsymbol{g}$	gravity acceleration
n	total number of body segments

From Eq.(2.2), the position of the point $p(x_{zmp}, y_{zmp}, z_{zmp})$ can be computed analytically by computing the cross product, based on the state and acceleration of the robot's articulated links and joints. Note that in the conventional definition of ZMP, $z_{zmp} = 0$.

$$\begin{aligned} x_{zmp} &= \frac{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)x_i - \sum_{i=1}^{n} m_i\ddot{x}_i z_i - \left[\sum_{i=1}^{n} \frac{d(I_i\omega_i)}{dt}\right]_Y}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \\ y_{zmp} &= \frac{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)y_i - \sum_{i=1}^{n} m_i\ddot{y}_i z_i + \left[\sum_{i=1}^{n} \frac{d(I_i\omega_i)}{dt}\right]_X}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \end{aligned} \tag{2.3}$$

If the robot is treated as a point mass concentrated on the CoM, Eq.(2.2) becomes

$$\Big((\boldsymbol{p}_G - \boldsymbol{p}_{zmp}) \times M\boldsymbol{a}_G + \dot{\boldsymbol{L}}_G\Big)_{horizontal} = \Big((\boldsymbol{p}_G - \boldsymbol{p}_{zmp}) \times M\boldsymbol{g}\Big)_{horizontal} \tag{2.4}$$

where

$\boldsymbol{p}_G = [x_G, y_G, z_G]^T$	trajectory of the CoG
M	total weight of humanoid robot
$\boldsymbol{a}_G = [\ddot{x}_G, \ddot{y}_G, \ddot{z}_G]^T$	acceleration of the CoG
$\boldsymbol{L}_G = [L_{G_x}, L_{G_y}, L_{G_z}]^T$	angular momentum of the robot about the CoG
$\boldsymbol{g} = [0, 0, -g]^T$	gravity acceleration

Computing the cross product in Eq. (2.4), the ZMP can be expressed as a function of the CoG position, the CoG acceleration and net moment about the CoG:

$$\begin{aligned} x_{zmp} &= \frac{M[x_G(\ddot{z}_G + g) - (z_G - z_{zmp})\ddot{x}_G] - \dot{L}_{G_y}}{M(\ddot{z}_G + g)} \\ y_{zmp} &= \frac{M[y_G(\ddot{z}_G + g) - (z_G - z_{zmp})\ddot{y}_G] + \dot{L}_{G_x}}{M(\ddot{z}_G + g)} \end{aligned} \tag{2.5}$$

Another method to obtain the position of the point p is with the aid of force sensors. For the application of dynamically balanced gait, the center of pressure (CoP) and the ZMP are equal. The CoP is the point where the resultant of the normal forces acts. Due to the nature of the force sensors only the normal forces are measured, no information about tangential forces are available. Let f_{ni} be the measured normal force at the force sensor i, p_i be the position of sensor i in the robot coordinate system, $R_n = \sum f_{ni}$ be the force resultant at the

CoP, and n be the number of force sensors in the feet. Then the position of the CoP P_{cop} can be calculated as:

$$P_{cop} = \frac{1}{R_n} \sum_{i=1}^{n} f_{ni} * p_i$$

The ZMP stability criterion is that if the computed acting point of the ground reaction force (GRF) p is within the support polygon, p is ZMP, the system is in dynamic equilibrium. Therefore, in order to ensure dynamic equilibrium, the computed point p must be within the support polygon. Else, if the computed point p is outside the support polygon, p is called a fictitious ZMP. In this case the GRF acting point is on the support polygon border. The unbalanced moment arouses the mechanism rotation about the edge of the support polygon, as a result the system is likely to tip over. The intensity of the unbalanced moment depends on the distance from the support polygon edge to the fictitious ZMP (FZMP).

The first practical application of ZMP was in the realization of the complete dynamic walking of the biped walking robot WL-10RD at Waseda University in Japan in 1984 [123], TAKANISHI *et al.* used ZMP to discern the stability of each state of the dynamic walking of the robot [115]. Since then, a number of successful biped humanoid robots have applied the ZMP concept to their walking pattern generation and control.

However, ZMP notion has limitations. First, the key prerequisite with this notion is that the support foot rests fully on the floor. Actually the foot of a robot is not generally in perfect contact with the ground. Moreover, humans do not use ZMP stability criterion. We allow our feet to roll, for example, toe-off and heel-strike. Second, the ZMP notion cannot describe robots with point feet, for example, robots walking on stilts. Third, ZMP is used to determine the dynamic stability of a legged robot in the case that the robot walks on a horizontal plane. But a robot may walk on stairs or a rough terrain, or manipulate an object using two hands as well as two legs. Therefore, considering the contact points between the robot and the environment in three-dimensional space, the virtual horizontal plane was constructed [113] and the enhanced ZMP was proposed [41]. Besides, in order to incorporate hand interaction forces, the conventional definition of ZMP was augmented [20]. Fourth, once the ZMP is at the very edge of the foot envelope, additional rotational dynamics of the foot, such as different rates of rotational acceleration, are no longer discernible using the ZMP [95]. The FRI point solves the problem by quantifying the rotation of the stance foot during the single support phase.

2.1.2 Foot-Rotation Indicator Point

The FRI point is a point on the foot/ground surface, within or outside the support polygon, where the net ground reaction force would have to act to keep the foot stationary. FRI point is a useful quantity for determining whether the foot will roll [28].

The equation for rotational dynamic equilibrium is obtained by noting that the sum of the external moments on the robot is equal to the sum of the rates of change of angular momentum of the individual segments about the same point. As shown in Figure 2.2, taking moments at the origin O, and treating the stance foot as the focus of attention, the dynamics of the rest of the robot can be represented by Eq.(2.6).

$$\boldsymbol{M} + \boldsymbol{OP} \times \boldsymbol{R} + \boldsymbol{OG}_1 \times m_1 \boldsymbol{g} - \boldsymbol{\tau}_1 - \boldsymbol{OO}_1 \times \boldsymbol{R}_1 = \dot{\boldsymbol{H}}_{G1} + \boldsymbol{OG}_1 \times m_1 \boldsymbol{a}_1 \tag{2.6}$$

where

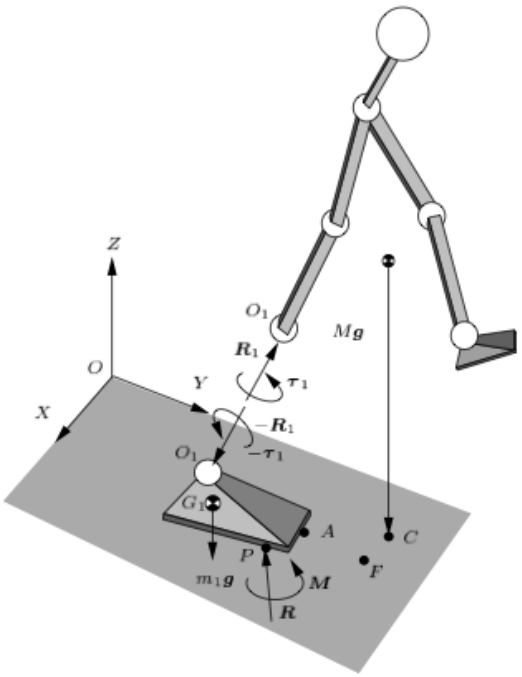


Figure 2.2: Foot Rotation Indicator point from [16].

$\boldsymbol{M}$	ground reaction torques
$\boldsymbol{R}$	GRF
m_1	the mass of the stance foot
G_1	the CoM location of the stance foot
$-\boldsymbol{R}_1$	ankle force exerted by the rest of the robot
$-\boldsymbol{\tau}_1$	ankle torque exerted by the rest of the robot
P	CoP
$\boldsymbol{a}_1$	the CoM linear acceleration of the stance foot
$\dot{\boldsymbol{H}}_{G1}$	the torque about the stance foot

The equations for static equilibrium of the foot are obtained by setting the dynamic terms in Eq.(2.6) to zero.

$$\boldsymbol{M} + \boldsymbol{OP} \times \boldsymbol{R} + \boldsymbol{OG}_1 \times m_1\boldsymbol{g} - \boldsymbol{\tau}_1 - \boldsymbol{OO}_1 \times \boldsymbol{R}_1 = \boldsymbol{0} \tag{2.7}$$

We can always find a point F, when taking the point F as the reference point to compute the moments, and considering only the tangential (XY) vector components of Eq.(2.7), Eq.(2.8) is satisfied.

$$\left(\boldsymbol{\tau}_1 + \boldsymbol{FO}_1 \times \boldsymbol{R}_1 - \boldsymbol{FG}_1 \times m_1\boldsymbol{g}\right)_t = \boldsymbol{0} \tag{2.8}$$

The subscript t implies the tangential components. Then the FRI point F is defined as the point on the foot/ground contact surface, within or outside the convex hull of the foot-support area, at which the resultant moment of the force/torque impressed on the foot is normal to the surface [16].

When viewing the robot minus the foot as a system, the dynamics of this system at point

F can be presented. If only the tangential components are considered, we have

$$\left(\boldsymbol{\tau}_1 + \boldsymbol{FO}_1 \times \boldsymbol{R}_1 + \sum_{i=2}^{n} \boldsymbol{FG}_i \times m_i \boldsymbol{g}\right)_t = \left(\sum_{i=2}^{n} \dot{\boldsymbol{H}}_{Gi} + \sum_{i=2}^{n} \boldsymbol{FG}_i \times m_i \boldsymbol{a}_i\right)_t \tag{2.9}$$

Using Eq.(2.8) and Eq.(2.9), we get

$$\left(\boldsymbol{FG}_1 \times m_1 \boldsymbol{g} + \sum_{i=2}^{n} \boldsymbol{FG}_i \times m_i (\boldsymbol{g} - \boldsymbol{a}_i)\right)_t = \left(\sum_{i=2}^{n} \dot{\boldsymbol{H}}_{Gi}\right)_t$$

Explicit expressions for the coordinates of F, $\boldsymbol{OF}(OF_x,\ OF_y,\ OF_z = 0)$ can be then obtained. To ensure no foot rotation, the FRI point must remain within the support polygon. If the calculated FRI point is outside the support polygon, there will be an uncompensated moment which causes the foot to rotate.

Though FRI point concept is applicable only during the single support phase of a biped, it can additionally perform the role of measuring the instability of a biped (see Eq.(2.6), where $\boldsymbol{H}_{G1}$ contains information on angular acceleration of the stance foot).

Hofmann applied the FRI point to the balance control of bipedal walking in his PhD thesis [28]. According to the torque balance equation of FRI, the torque in the system was divided into three parts, the orbital torque, the spin torque and the ankle torque.

2.1.3 Centroidal Moment Pivot

CMP notion is based on a fundamental law of nature that the angular momentum of a system about its CoM is conserved in the absence of external torques or the external torques sum up to zero. This is expressed as Eq.(2.10). Conversely, a non-constant angular momentum due to external forces/torques gives rise to a non-zero torque about CoM and consequently implies the tendency of the robot to tip forward, as formulated by Eq.(2.11).

$$\begin{aligned} \boldsymbol{H}_G &= \boldsymbol{k} \\ \dot{\boldsymbol{H}}_G &= 0 \end{aligned} \tag{2.10}$$

$$\boldsymbol{\tau}_{CoM} = \dot{\boldsymbol{H}}_G \tag{2.11}$$

where

$\boldsymbol{H}_G$	the angular momentum about the CoM
$\boldsymbol{k}$	a constant vector

The rotational stability criterion of CMP point analyzes the moment equation of the robot (refer to Figure 2.3) with the foot assumed to be on planar support surface. The torque about the CoM can be expressed as:

$$\boldsymbol{\tau}_{CoM} = \boldsymbol{GP} \times \boldsymbol{R} \tag{2.12}$$

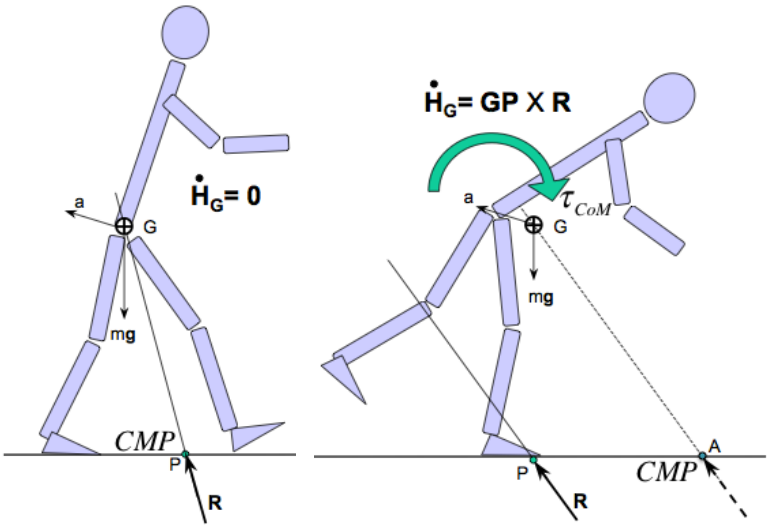
where

$\boldsymbol{\tau}_{CoM}$	the torque about the CoM
G	the position of CoG of the robot
$\boldsymbol{R}$	resultant GRF
P	CoP where $\boldsymbol{R}$ is acting

Combining Eq.(2.11) and Eq.(2.12), we have

$$\boldsymbol{\tau}_{CoM} = \dot{\boldsymbol{H}}_G = \boldsymbol{GP} \times \boldsymbol{R}$$

$\dot{\boldsymbol{H}}_G = 0$ implies $\boldsymbol{GP}$ is parallel to $\boldsymbol{R}$. This also means the GRF passes through the CoM and consequently generates a zero moment and the robot is rotationally stable (see Figure 2.3(a)). In this case P is the CMP. $\dot{\boldsymbol{H}}_G \neq 0$ indicates that GRF does not pass through the CoM thus generating a net clockwise or counterclockwise moment around the CoM (In the situation shown in Figure 2.3(b), the moment is clockwise.). In this situation the robot is rotationally unstable and has the tendency of tipping forward. However, we can find a point A in order for the GRF vector to pass through the CoM, and point A is named the CMP.



(a) (b)

Figure 2.3: The essence of stability analysis based on $\dot{\boldsymbol{H}}_G$ from [17].

The CMP is defined as the point where a line parallel to the GRF, passing through the CoM, intersects with the external contact surface [95]. The CMP is not necessarily inside the support polygon, where the ZMP would have to be. CMP is valid during both the single and double support phases of walking. We get insight from the CMP of developing the robot controller by attempting to make $\dot{\boldsymbol{H}}_G = 0$ or $\boldsymbol{H}_G = \textbf{constant}$.

Goswami and Kallem proposed the same point called the Zero Rate of change of Angular Momentum (ZRAM) point. In [17] they proposed three control strategies in order for $\dot{\boldsymbol{H}}_G = 0$ to recapture balance, including enlarging the size of support polygon, changing the position of the CoG, and changing the direction of GRF by centroidal acceleration.

By expressing the dynamic stability in terms of linear and angular momentum, Kajita *et al.* applied the Resolved Momentum Control method [44] to planning motion of a humanoid robot. The generated kick motion and walking motion were evaluated on the humanoid robot HRP-2.

Measures of biped robot stability that are manifested as a point on the ground surface are ZMP, FRI point and CMP (ZRAM point). Popovic *et al.* pointed out that when the stance

foot is at rest during single support, and when there is zero moment about the CoM, the ZMP, FRI, and CMP coincide [95].

2.1.4 Valid Stable Region

Valid stable region is defined as the area of the ZMP with a stability margin larger than the change of the ZMP position due to environmental disturbances. The valid stable region Ω is denoted by the following equation:

$$\Omega = \{(x_{zmp}, y_{zmp}) | d_s(x_{zmp}) \geq d_f(x_{zmp}), d_s(y_{zmp}) \geq d_f(y_{zmp})\}$$

where $d_s(x_{zmp}), d_s(y_{zmp})$ denote minimum distances from the ZMP to the boundary of the stable region; $d_f(x_{zmp}), d_f(y_{zmp})$ denote the change of the ZMP position due to disturbances in the x-axis and y-axis, respectively, and d_{zmp} indicates the stability margin with respect to the current position of the ZMP (see Figure 2.4).

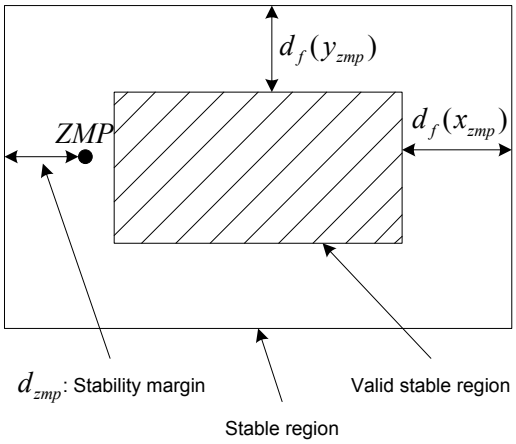


Figure 2.4: Valid stable region.

$d_f(x_{zmp})$ and $d_f(y_{zmp})$ can be calculated analytically [34]. When the ZMP is inside the valid stable region, the system will be stable even if affected by disturbances. In this case, the system can execute tasks even without control for stability. When the ZMP is inside the stable region but outside the valid stable region, the system will probably become unstable in the presence of disturbances. It is, therefore, necessary in this case to control the system for stability and task execution.

In [32], Huang *et al.* proposed a method of designing the hip trajectory of a biped robot with the purpose of maximizing the stability margin. The concept of valid stable region was adopted in [34] for motion planning of a manipulator.

Among the above notions, ZMP concept has gained extensive recognition and at present is the main method in robotics to analyze, predict, and control postural balance in biped robots. This thesis is based on the ZMP concept.

2.2 Simplified Models

Before jumping into bipedal walking algorithms, let's start by examining simplified models of bipedal walking. The models are enlightening so that we can draw intuition from them. We start with a simple inverted pendulum and progress to more detailed models.

2.2.1 Inverted Pendulum Model

Pratt demonstrated the similarity between human walking and an inverted pendulum [97]. So understanding the dynamics of an inverted pendulum will help us understand the dynamics of walking. Figure 2.5(a) shows an inverted pendulum of length ℓ with a point mass, m, at its end. g is the standard gravity on the surface of the Earth, and θ is the angular displacement measured from the equilibrium position. Then we have an analogical model of bipedal walking, that mass corresponds to the body mass of a bipedal robot and the pendulum is equivalent to its leg.

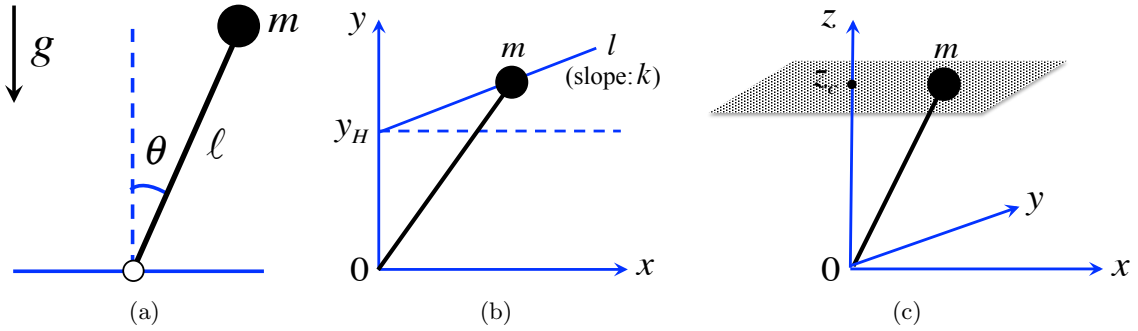


Figure 2.5: Inverted pendulum models. (a) inverted pendulum model (b) LIPM (c) 3D-LIPM

The torque due to gravity is

$$\boldsymbol{\tau} = mg\ell \sin\theta \tag{2.13}$$

The change in angular momentum about the pivot is

$$\boldsymbol{\tau} = I\ddot{\theta} = m\ell^2\ddot{\theta} \tag{2.14}$$

where I is the moment of inertia for the point mass.

We get the equations of motion of the pendulum:

$$\ddot{\theta} = \frac{g}{\ell}\sin\theta \tag{2.15}$$

The insight we can draw from the inverted pendulum model is that, suppose the mass is traveling from left to right, if the mass is on the left of the pivot point, it will slow down as it rises, converting kinetic energy into gravitational potential energy; if it is on the right of the pivot point it will speed up, converting gravitational potential energy back into kinetic energy [97]. This model can be used to explain the reason why in human walking human body must slow down and then speed up during each step. Accordingly, each step is divided into speed up phase, constant velocity phase and slow down phase while designing the walking pattern of the robot in Chapter 3.

2.2.2 Linear Inverted Pendulum Mode

In 1991 Kajita *et al.* proposed the concept of the Linear Inverted Pendulum Mode (LIPM) for the design and control of dynamic walking motion of a biped robot [48]. As shown in Figure 2.5(b), it is a planar inverted pendulum model with two constraints. One constraint keeps the CoG of the body to move on a straight line which has the slope of k and intersects with y-axis at y_H. The other keeps the rotation rate of the body constant.

The motion equation for the model is

$$m(y\ddot{x} - x\ddot{y}) + I\ddot{\theta} = \tau + mgx \tag{2.16}$$

where τ is the ankle torque of the support leg.

The two constraints are

$$\begin{aligned} y &= kx + y_H \\ \dot{\theta} &= \omega_c \qquad (\omega_c \text{ is constant.}) \end{aligned} \tag{2.17}$$

Differentiating Eq. (2.17), the constraints in terms of acceleration are:

$$\begin{aligned} \ddot{y} &= k\ddot{x} \\ \ddot{\theta} &= 0 \end{aligned} \tag{2.18}$$

Substituting Eq. (2.17) and Eq. (2.18) into Eq. (2.16), a simple linear dynamics can be expressed in Eq.(2.19).

$$\ddot{x} = \frac{g}{y_H}x + \frac{1}{my_H}\tau \tag{2.19}$$

The insight we can draw from the LIPM is as follows. The differential equation of motion of an inverted pendulum is linear in the Cartesian coordinate frame, and it does not depend on neither the structure of a leg nor the constraint parameters (except y_H). These make the design of biped locomotion simplified.

Kajita *et al.* then extended LIPM to 3D Linear Inverted Pendulum Mode (3D-LIPM) by applying a constraint control to an inverted pendulum such that the mass moves along an arbitrary defined plane [43], as illustrated in Figure 2.5(c). Especially, if the constraint plane is horizontal, the dynamic equations of motion are given by

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}\tau_y \tag{2.20}$$

$$\ddot{y} = \frac{g}{z_c}y - \frac{1}{mz_c}\tau_x \tag{2.21}$$

where τ_x, τ_y are the torques around x-axis and y-axis, respectively.

The dynamic equations of motion become linear and decoupled so that the motion of the robot can be obtained analytically.

2.2.3 Angular Momentum Inducing Inverted Pendulum Model

Komura *et al.* proposed an Angular Momentum inducing inverted Pendulum Model (AMPM) to counteract the increased angular momentum due to external perturbations [63].

In the AMPM, the ZMP is allowed to move over the ground, and its position must be linearly dependent on that of the CoM. The horizontal component of the ground force vector

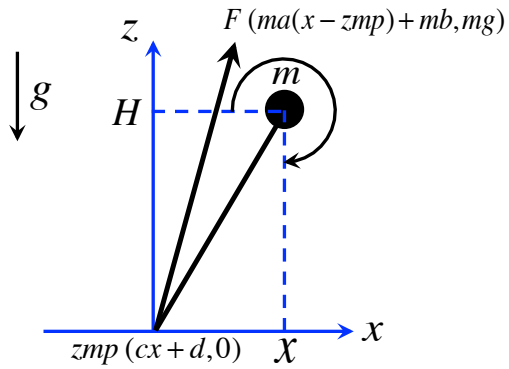


Figure 2.6: Angular momentum inducing inverted pendulum model.

is allowed to change, by an amount which must be linearly dependent on the CoM. As shown in Figure 2.6, the position of the CoM is (x, H), the position of the ZMP is $(cx + d, 0)$, and the vector of the ground force is parallel to the vector $(a(x - zmp) + b, g)$ where a, b, c, d are constants, m is the mass of the system and g is the acceleration of gravity. Then the increased angular momentum between time t_1 and t_2 can be quantificationally obtained:

$$\omega_{t_1,t_2} = \left[\frac{m(1-c)(aH-g)}{\sqrt{a-ac}} \left(-C_1 e^{-(\sqrt{a-ac})t} + C_2 e^{(\sqrt{a-ac})t} \right) + mgt\left(\frac{b}{a}\right) \right]\Bigg|_{t_1}^{t_2} + \omega_1 \qquad (2.22)$$

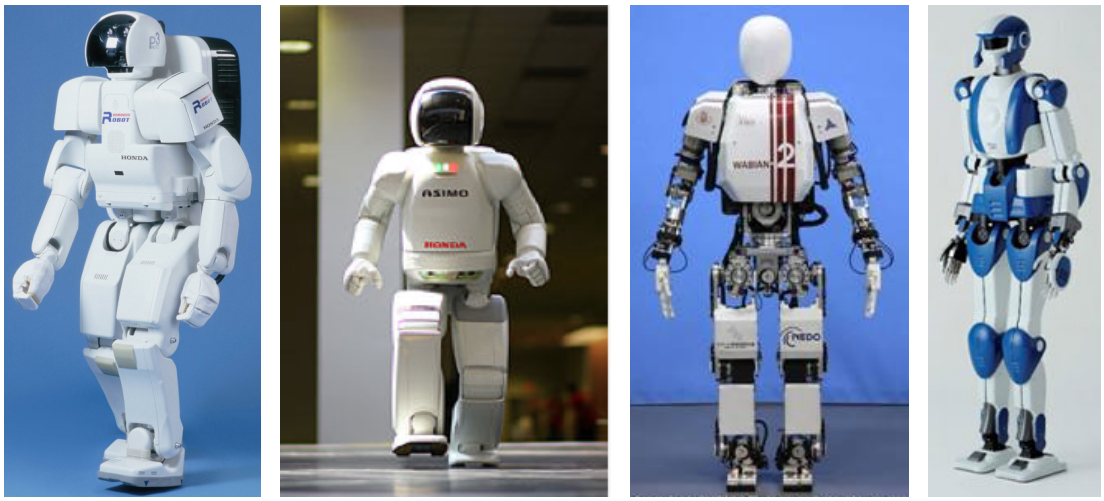
where ω_1 is the angular momentum at $t = t_1$, and C_1, C_2 are constant values which can be determined by initial conditions at $t = 0$. More detailed calculation can be found in [63].

In [63] the increased angular momentum was offset to zero by adjusting the direction of the GRF vector so that the GRF pass through the CoM again. This is realized by accelerating the CoM during the double support phase. The idea is in accordance with the concept of CMP or ZRAM in which $\boldsymbol{H}_G$ should be constant. The method is similar to the hip strategy used by human beings to maintain balance against perturbations while keeping upright standing posture. AMPM is considered as an extension of Kajita and Tani's LIPM, because besides the linear momentum the former model incorporates the centroidal angular momentum to control the robot.

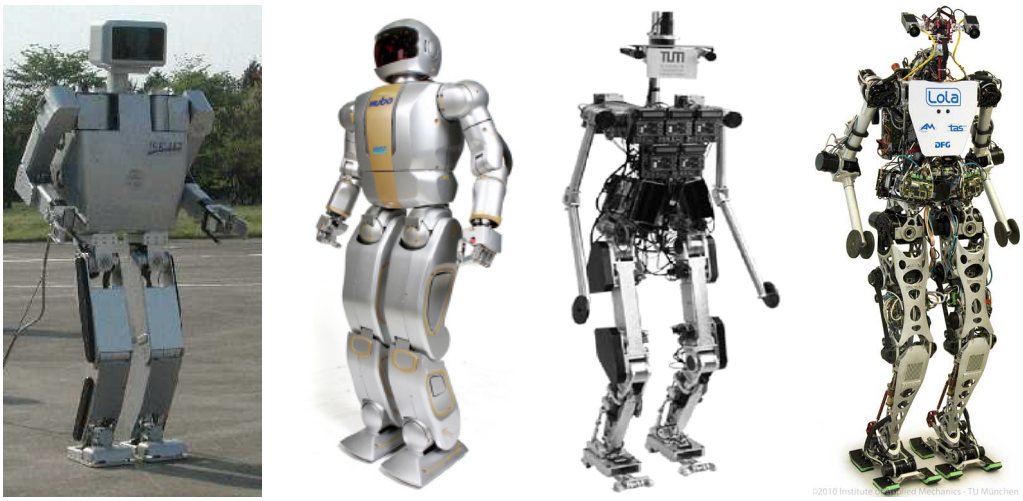
2.3 Related Works

Many research groups have developed their own humanoid robot platforms in order to realize robots that can coexist with humans and perform a variety of tasks. For example, as shown in Figure 2.7, a HONDA research group has developed the humanoid robots P2, P3, and ASIMO. Waseda University has developed the WABIAN series. The Japanese National Institute of Advanced Industrial Science and Technology (AIST) and Kawada Industries, Inc. have developed HRP series. The University of Tokyo has built the H6 and H7. The Korea Advanced Institute of Science and Technology (KAIST) has developed KHR and HUBO. The Technical University of Munich has developed Johnnie and Lola. These are the most advanced and well-known biped humanoid robots in the world. Actually, there is a large and growing body of ongoing research on humanoid and legged robots. Due to the very large number of publications in this field, only the work considered to be the most relevant for this

thesis is summarized briefly in the following. In particular, only work on fully actuated, three dimensional humanoid robots is considered.



(a) P3 (b) ASIMO (c) WABIAN-2R (d) HRP-4



(e) H7 (f) HUBO2 (g) JOHNNIE from [73] (h) LOLA

Figure 2.7: Well-known biped humanoid robots. Sources: (a) http://asimo.honda.com/ASIMO_DCTM/News/images/highres/21_P3.jpg; (b) http://asimo.honda.com/ASIMO_DCTM/News/images/highres/ASIMOatFIRST3.jpg; (c) http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/index.htm; (d) http://www.aist.go.jp/aist_j/press_release/pr2010/pr20100915/pr20100915.html; (e) http://www.jsk.t.u-tokyo.ac.jp/research/h6/H7_05.jpg; (f) http://hubolab.kaist.ac.kr/p_hubo2p; (h) http://www.amm.mw.tum.de/index.php?id=93.

2.3.1 Honda Motor

Honda in Japan has exhibited bipedal walking robots P1, P2, P3[1] from 1993 to 1997 [30]. P1, with a height of 1915 mm and a weight of 175 kg, was mainly used for fundamental research on bipedal walking mechanism.

Honda developed its own first humanoid walking robot P2 in 1996 [27]. P2 is a self-contained humanoid robot. It is 1820 mm tall and weighs 210 kg. The whole body has a total of 30 DoFs. Possessing two arms and two legs, P2 has more humanoid characteristics than P1. The essence of the posture control is similar to the CMP concept introduced in Section 2.1. P2 could walk at the speed of 3 km/h. However, it is too tall and heavy to work in practice. Besides, the power expenditure while walking was high. In December 1997, the company brought forth the humanoid robot P3, which is similar to P2 but much smaller and lighter, with a height of 1600 mm and a weight of 130 kg. The performance and walking ability were improved as well. P3 has realized dynamic walking on a plane and stairway [27].

ASIMO[1], showed up in 2000, is 1200 mm in height, 52 kg in weight and has in total 26 DoFs. Walking speed can range from 0 to 1.6 km/h [100]. ASIMO can walk and turn fluidly, go up and down stairs, walk on a slope, adjust walking speed through altering walking frequency and step, operate switches and knobs, and even dance. ASIMO takes advantage of the 'i-WALK' technology [30], which embeds a control system that can predict the future motion online, thus making locomotion much smoother and more natural. The second generation of ASIMO is 1.3 m tall and weights 54 kg with 34 DoFs. The posture control algorithm and highly responsive hardware improved the walking speed to 2.7 km/h and enabled ASIMO to run at 6 km/h. A three-mass model was used as an approximate dynamics model to generate the walking pattern [117]. The all-new ASIMO being developed has the same height as the second generation, but decreases 6 kg in weight. The robot has in all 57 DoFs. Running speed can achieve 9km/h.

The great success of HONDA humanoid robots triggered the world's research on humanoid robots. Since the second prototype HONDA humanoid robot P2 was revealed in 1996, many biped humanoid robots have been developed in the world.

2.3.2 Waseda University

Researchers at Waseda University Japan have been studying on bipedal walking robots since 1969. In 1969, late Prof. Kato initiated modern research on biped walking robots at the Waseda University by constructing an anthropomorphic pneumatically activated pedipulator WAP-1 [52]. In 1972, WL-5 accomplished automatic biped walking by computer control for the first time in the world [115]. In 1973, WABOT-1, which is known as the world's first full-scale anthropomorphic robot, was developed by Kato *et al.* [53]. Up to present, researchers at the Waseda University have built a series of bipedal walking robots and bipedal humanoid robots.

Research Achievements on WL series

Since the year 1969, WL series bipedal walking robots have gone through the experience from statically stable walking, quasi-dynamic walking to complete dynamically stable walking. During the period of development, four main achievements have been accomplished.

[1] Copyright©: Honda.

1. The development of the control method which takes advantage of three-DoF trunk motion to compensate for the three-axis moment given an arbitrary planned ZMP. Before walking, desired lower limb and ZMP trajectories were preset. The balancing motion of the trunk was computed by iteratively solving the approximate equations of motion using the desired lower limb and ZMP trajectories. During walking, these precalculated trajectories were played back with the program control. With this method, dynamic biped walking was achieved. It was reported that the three-axis trunk motion improved the stability of the robot and allowed WL-12 RV to walk 50% faster than with two-axis trunk motion [127] .

2. Dynamic walking was realized under an unknown external force [116]. Online control was added to compute the trunk motion which was altered in real time from the preset trajectory to maintain the trajectories of preset stance leg and ZMP. The landing points of the swing leg was changed to return the trunk motion to the preset trajectory within three steps. A learning method was implemented and the compensative trunk motion was optimized [68].

3. The compliant foot mechanism was developed to absorb impact and contact forces between the foot and the ground using elastic pads [128]. Virtual compliance control with variant compliance parameters for the sake of reducing the virtual stiffness of the swing foot was also studied to decrease the impact forces [110].

4. Parallel mechanisms WL-15, WL-16 and WL-16R [110] were developed for the application in the medical and welfare field.

WABIAN series

Based on the studies on biped walking robots, the research on humanoid robots WABIAN series began in 1996 as the second stage of robot studies.

WABIAN [126] is a 35-DoF humanoid bipedal walking robot with an anthropomorphic body. The walking control method implemented on biped walking robots was further extended by using arm trajectories. Trajectories of the arm and leg joints as well as ZMP were predetermined. Compensatory motions of the trunk are then computed not only by the desired lower limb motion and ZMP trajectories but also by the desired upper limb motion. WABIAN can walk dynamically, dance in place, and carry load using its arms.

An impedance control method for WABIAN-RIII was created to absorb the impact/contact forces generated between the landing foot and the ground, which can adjust impedance like the relaxed and hardened motion of muscles of a human [70]. An online locomotion pattern generation was developed for a biped humanoid robot having a trunk, which is based on visual and auditory sensors [69].

WABIAN-2/LL was developed as the lower limb mechanism of the bipedal humanoid robot WABIAN-2. By predetermining the knee joint pattern, the singularity problem using inverse kinematics to creat knee-stretched walking was solved [86]. Compared with conventional walking with constant waist height, knee-stretched walking realizes lower energy consumption in knee actuators. Dynamic walking was achieved at the walking speed of 0.96 s/step with 0.35 m step length on a flat floor.

WABIAN-2 [85] is a full scale humanoid robot by mounting a new upper body mechanism which has a 2 DoFs trunk and a 2 DoFs waist on WABIAN-2/LL. WABIAN-2 has 41 DoFs with the height of 1.53 m and the weight of 64.5 kg. It was designed as a human motion

simulator and was used in the medical field to simulate the elderly and handicapped to push walk-assist machine. There are two characteristics in mechanical design. One is that both trunk and waist have 2 DoFs. The joint configuration of the trunk and waist makes the robot enable to imitate the human trunk motion. The other is that the arms can hold robot's weight while it leans against on a walk-assist machine.

Up to now, the latest version of the humanoid robot in the Waseda University is WABIAN-2R[2]. It is 1480 mm in height, and 63.8 kg in weight. Each ankle is mounted a 6-axis force/-torque sensor. In order to mimic human movements, the robot has 41 DoFs and joint ranges of motion were designed in reference to a human being. Knee-stretched walking has been realized since the 2 DoFs waist provides a redundant DoF (waist rolling motion) which solves the singularity problem in inverse kinematics. WABIAN-2R is capable of adaptive walking depending on the ground condition. The robot prefers either knee-stretched fast walking on even floor or knee-bend walking on uneven terrain. The improvement on foot mechanism was the emphasis of the robot mechanical design. First, knee-stretched walking with heel-contact and toe-off motion was accomplished indoors and outdoors by adding a passive toe joint on the foot. Second, an arch-like structure observed in human foot was added on the foot [23]. The arch structure absorbed the generated impact when the sole fully contacted the ground. Third, the foot mechanism was improved from 4-point contact mode to 3-point contact mode which makes the robot more adaptive to the unstructured environment. WABIAN-2R with 3-point contact mode was able to walk with larger stability margin and less foot slipping problem. Besides, WABIAN-2R was used as a human motion simulator and emulated a disabled person's gait [64]. Turning motion based on human motion by utilizing the slip between the feet and the ground was realized by PD controlling the roll axis angle of the ankle joint [24]. By applying stabilization algorithms to the single support phase and double support phase, the robot succeeded walking on the soft ground [22]. Jumping motion on the moon gravity was studied in the simulation level [90].

In summary, the humanoid robots at Waseda University are human-like, not only in the appearance and mechanical structure, but in walking behaviors and capabilities. From WL series to WABIAN series, we can see that the development of humanoid walking robots at Waseda University is oriented towards the goal of coexisting and cooperating with humans.

2.3.3 AIST and Kawada Industries

AIST has developed a number of sophisticated humanoid robots together with Kawada Industries. Development started with the "humanoid robotics project" (1998 to 2002) funded by the Japanese Ministry of Economy, Trade and Industry (METI).

HRP-2P, the prototype humanoid robotics platform for HRP-2, is a 30 DoFs humanoid robot that is 1.54 m tall and weights 58 kg. It is reported to be the first human-size humanoid robot that can lie down to the floor and get up from the floor using static and dynamic motions [50]. The stability of the dynamic motion is ensured by tracking the desired ZMP trajectory via controlling the hip pitch joint. In addition, a walking control and pattern generation method called preview control realized the dynamic walking of HRP-2P by solving an infinite horizon linear quadratic regulator (LQR) problem [43]. Using waist motion to compensate the moment around the yaw axis of the robot, 2.5 km/h walk was achieved by HRP-2.

[2]Copyright©: Takanishi Laboratory.

HRP-3 [51] has a larger size than HRP-2P with the height of 1.6 m and the weight of 68 kg. The number of driven joints increases to 42. The mechanical configuration of the waist and hip joints of HRP-3 inherits from HRP-2. The difference lies in that HRP-3 has a 3-fingered hand which improves the manipulation capability of the robot. The distributed control system of the HRP-3 is based on Controller Area Network (CAN), while HRP-3P is based on real-time Ethernet. In addition, HRP-3 was designed for outdoors. The robot was capable of walking on an unexpected slippery floor with low friction by adopting feedback control with a slip observer.

HRP-4C is a life-size humanoid robot which was developed for the entertainment purpose by AIST. Its dimensions are designed referring to a database of Japanese women of 20-year-old [114].

The newest robot HRP-4[3] [54] is developed for the purpose of coexisting with humans and doing safer collaborative works with humans. The concept of mechanical design includes an increase in the degree of freedom of both arms which is essential to handle objects while achieving a smaller and lighter-weight body in comparison to conventional HRP series. The robot can hold an object with a 5-fingered hand. The real-time motion control system is developed by AIST.

2.3.4 University of Tokyo

H5, H6 and H7 humanoid robots were developed at the University of Tokyo.

H5 is a child size humanoid robot developed as a research platform of dynamic bipedal locomotion. Dynamic balanced trajectory generation has been studied on this platform.

H6 [84] is 1.37 m in height and 55 kg in weight, and has a total of 35 DoFs. It has been developed for an integration of vision, tactile, motion planning and dynamic bipedal locomotion. Aircraft technologies which lead to a strong and light structure were applied to the body frame. The desired walking pattern of H6 is generated online by constructing typical stepping patterns offline in advance and then mixturing and connecting these patterns online [83].

H7[4] [42] was improved over H6 by attaching six-axis force sensors, hand pressure sensors and toe joints to the robot. An online walking pattern generation method [82] makes the dynamic walking follow the desired ZMP trajectory by modifying horizontal torso trajectory via iterative calculation when the initial trajectory is given. H6 and H7 can walk up and down 25 cm high steps and can also recognize preentered human faces by a computer vision system.

2.3.5 KAIST

KAIST developed several humanoid robots. KHR-2 [58, 59], developed in 2003 was made to resemble a child-sized human with the weight of 56 kg and the height of 1.2 m. It has total 41 DoFs. The soles are flat without toe joints. The robot is equipped with six-axis force/torque sensors on the feet and wrists and an inertial measurement unit (IMU) in the upper body. The walking pattern was desiged offline based on the inverted pendulum model. A cosine function was used for planning the lateral trajectory of the pelvis. The trajectory in the forward direction was generated by mixing the cosine and linear function. Trajectories of

[3] Copyright©: National Institute of Advanced Industrial Science and Technology (AIST) and KAWADA INDUSTRIES, INC.

[4] Copyright©: JSK Laboratory, University of Tokyo.

both feet were depicted by a cycloid function. Several online controllers based on the sensory feedback were used for stabilizing the posture of the robot according to the walking stages. With this system, KHR-2 realized dynamic walking not only on flat floor with the speed of 1 km/h, but on uneven and inclined floor with local or global inclination of $\pm 2^{\circ}$.

KHR-3 HUBO [91] was upgraded directly from KHR-2, and therefore, has the same mechanical configuration as KHR-2. The walking pattern of HUBO was online generated by using a cycloid function for the ankle trajectory and a 3^{rd} order polynomial interpolation for the pelvis trajectory. The control methods remained similar to KHR-2. The walking speed is up to 1.25 km/h. Using offline generated trajectories and online controllers, the robot can run at a speed of 3.24 km/h [9].

Albert HUBO [89] is an android type humanoid robot by combining the HUBO platform and a head which is capable of expressing facial expressions.

The latest version of the HUBO series is HUBO2[5] [35]. This robot was developed in 2009. It is 1.25 m in height, 45 kg in weight and has in all 40 DoFs. The main goal of the HUBO2 design is to achieve the lightest human-size humanoid robot in the world. HUBO2 can walk at a speed of 1.5 km/h and run at a maximum speed of 3.6 km/h. Moreover, the new walking algorithm permitted stretched leg walking of the robot, making HUBO2 different from the previous robots in the series.

2.3.6 Technical University Munich

The researchers in the Institute of Applied Mechanics, Technische Universität München developed humanoid robots Johnnie and its successor Lola.

The biped robot Johnnie[6] [73] with vision system is 1.8 m in height and 40 kg in weight. The robot has 12 DoFs in the lower body and 5 DoFs in the upper body. The shoulder joints compensate for the overall moment of momentum. Six-axis force/torque sensors in the feet and an IMU in the upper body are used for stability control. The horizontal walking pattern of the CoG was designed based on a lumped mass model. The disadvantage of the model is that the robot walks with unnatural bending knees. The CoG trajectory is planned using a fifth-order polynomial in the sagittal plane. Torque control with disturbance observers regulates the torque of the stance foot as well as that of the remaining 15 joints in order to track the reference trajectories. Since the robot with the torque control method could not achieve high speed walking in experiments, a different control method focusing on the overall system stability was applied. Balance is achieved by controlling the orientation of the upper body via regulating the torques at the ankle joints. Besides, impedance control is used to reduce the impact when the swing foot hits the ground. With this control method the robot was able to walk stably with a speed of 2.4 km/h and step length of 55 cm.

The humanoid robot Lola[6] [8] was constructed in 2006 for the purpose of fast, human-like and autonomous walking motion. The height is the same as Johnnie, but 15 kg heavier in weight. The mechanical and kinematic structure of Lola was improved over Johnnie from the biological point of view. In comparison to Johnnie, Lola has eight more DoFs - one DoF active toe joint in each foot, one DoF in each elbow, three DoFs camera head and one more DoF about roll axis in the pelvis. Additional degrees of freedom allow for more flexible and natural gait patterns. A three-point-mass model is used for real-time planning. The walking pattern is generated online by connecting the smooth and stable CoM trajectory for the next steps to

[5] Copyright©: HuboLab-KAIST.
[6] Copyright©: TUM Technische Universität München.

the current trajectory. Hybrid position/force control method in the task space is used with an inner joint position control loop, and an outer contact force control loop to provide inertial stabilization. The maximum forward walking speed of Lola was 3.34 km/h and Lola was able to walk sideways at a speed of 0.7 km/h [7]. With a computer vision system, Lola has the capability to autonomously navigate among obstacles.

2.4 Chapter Summary

This chapter provides insight for dynamic control of a biped humanoid robot.

Understanding the stability criteria presented in this chapter not only help us to understand the algorithms used in future chapters, but also the control strategies that have been discovered that humans employ. For example, humans seem to use a variety of control strategies in order to maintain balance in the presence of sudden horizontal perturbations while standing [31, 75, 77, 125]. These are referred to as the "ankle strategy", defined as torque about the ankle joint only, and the "hip strategy" defined as using hip flexors or extensors to generate shear forces at the feet that act to decelerate the CoM, and the "stepping strategy" defined as stepping out in order to reconfigure the support polygon.

These strategies are quite understandable in view of the stability criteria presented above. For smaller perturbations, the ankle strategy moves the location of the CoP and the balance is regained. When the perturbations are larger or when the support surface is narrow, the hip strategy lunges the body inertia in order to make the CoM accelerate in the opposite direction so that the GRF passes through the CoM again which indicates $\dot{H}_G = 0$. A combination of pure ankle and hip strategy is also possible. The last resort-strategy is elicited when the CoM of the body passes outside the limits of the base of support, stepping strategy keeps the CoP inside the support polygon by enlarging the size of the support polygon.

In Chapter 4, "ankle strategy" and "hip strategy" that utilize ankle and hip joints are employed to control the body posture of the walking robot based on the orientation and angular velocity of the upper body. The location of the ZMP on the foot is indirectly regulated.

3 Walking Pattern Generation

This chapter presents the motion generation method for biped humanoid robots. The basic objective is to generate a natural, human-like and efficient walking motion. Motivated by biomechanical studies on human walking, we model the walking pattern with continuous and differentiable mathematical functions. For stable walking of the robot, we design a pattern generator based on the ZMP criterion. The presented walking pattern involves a three-dimensional motion of the upper body instead of restricting the upper body motion to a flat surface (Figure 3.1). The advantages of the three-dimensional upper body motion lie in that (i) a natural and human-like walking pattern is achieved; (ii) since the robot does not always bend its legs, energy consumption to support the body weight will be small at knee joints. The comparison of energy consumption between the conventional walking pattern and the presented walking pattern will be made in Chapter 4. Moreover, the presented method needs lower calculation costs compared with the method based on the precise knowledge of robot dynamics.

Figure 3.2 illustrates the scheme of the walking system which is composed of an offline motion generator and an online control system. The offline motion generator calculates the reference walking pattern and the real-time control system aims at realizing stable, efficient and fast walking based on sensory feedback. The real-time control system that refers to position control and force control will be presented in Chapter 4, Chapter 5 and Chapter 6, respectively.

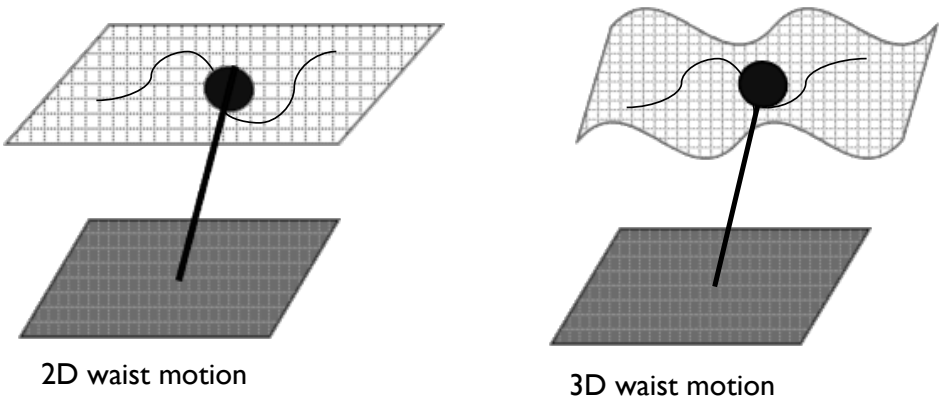


Figure 3.1: Two-dimensional waist motion vs. three-dimensional waist motion.

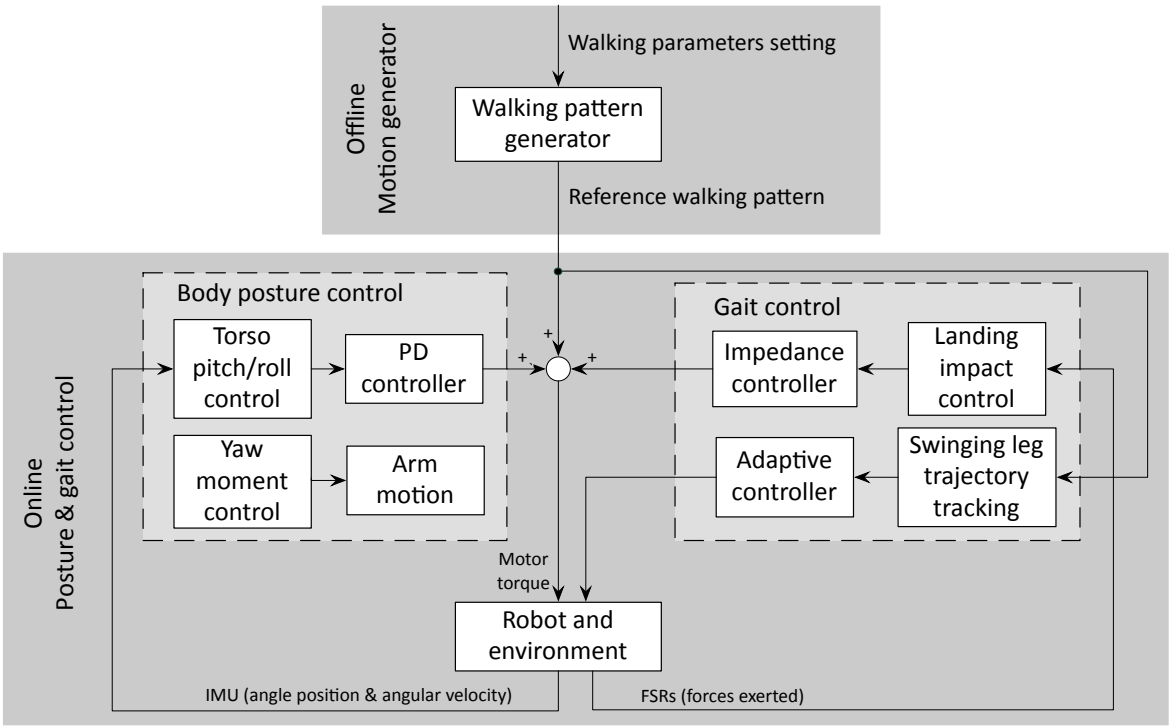


Figure 3.2: Overview of the walking control system.

3.1 Introduction

In the biped robotics research field, a *gait pattern* is a set of trajectories for the desired ZMP, the feet and the upper body. Planning the walking pattern is one of the most important parts in realizing the stable motion of a biped robot since a well-designed walking pattern can guarantee the motion stability.

3.1.1 Background and Related Works

Many different schemes for generating gait pattern of walking bipeds have been implemented during the past 30 years. The approaches can be summarized as follows.

Precise Dynamics vs. Approximated Model

From the viewpoint of control and walking pattern generation, research dedicated to biped walking pattern generation can be classified into two categories.

The first approach models the robot as a rigid multibody system which requires precise knowledge of robot dynamics, e.g. mass, location of center of mass and inertia of each link, to design walking patterns [32, 44, 116, 126, 127]. Therefore, it mainly relies on the accuracy of the models. Let us call this the ZMP-based approach since they often use the ZMP for pattern generation and walking control. The approach can derive a precise walking pattern that satisfies the desired ZMP trajectory, but it is hard to generate the walking pattern in real-time due to the large calculation burden. Further, if the mathematical model is different from the real robot, the performance is diminished.

On the contrary, the second approach uses limited knowledge of dynamics, e.g. location of total center of mass, total angular momentum, etc. The walking pattern is designed based on the limited information of a simple model [10, 43, 45, 58, 117]. We can call this the model-based approach, since it frequently uses a model. Since the controller knows little about the system structure, this approach greatly relies on feedback control. During walking, many kinds of online controllers are activated to compensate the walking motion through the use of various sensory feedback data. This approach can easily generate the walking pattern online. However, it depends strongly on the sensory feedback, hence the walking ability is confined to the performance of sensors and requires considerable experimental hand tuning.

Bending-knee Walking vs. Stretched-knee Walking

If we took the whole dynamic parameters of a humanoid robot into consideration to generate the proper walking pattern, more accurate results could be obtained. However, full dynamics need complicated and tedious computations and require a lot of time. For reducing computational difficulty, simplified humanoid robot models are used. Many researchers have employed the 3D-LIPM as the humanoid robot model.

By constraining the CoM motion on a horizontal plane, the dynamic equations of motion of the 3D-LIPM becomes linear and decoupled. This facilitates the walking motion design. However, as a result of the constant height of the CoM, robots employing this approach walk with unnatural bending knees. Consequently, the walking is less efficient since the energy consumption to support the body weight is higher in the knee joints during walking. In addition, it is hard to realize fast walking because the load is concentrated in the knee actuator. The advantage of bending-knee walking is that by lowering the CoM and thus decreasing the landing impact the stability can be easily achieved.

The other reason of walking with bending knees is that the extended knee configuration is in fact a singularity of inverse kinematics. Under conventional control, a humanoid robot cannot handle the singularity because an extremely large joint velocity would be generated which is beyond the bearable range of the mechanical joints [87]. Hence, these humanoid robots have always to bend the knees to avoid such a singularity.

Many studies which have achieved great success in realizing dynamic stable walking [10, 39, 43, 45, 58, 73, 91] involved walking patterns with bending knees.

Indeed, the knee of the support leg is not bent very much during the support phases in human walk [102]; stretched-knee motion is also produced when the heel contacts on the ground [87]. The advantages in walking with stretched knees and lifted waist are (1) Human-like natural walking motion is achieved; (2) Energy efficiency is improved since required torque and energy consumption to support the body weight become small at knee joints [66]. Hence the ability to walk with stretched knees is an important quality that a humanoid robot should possess in order for it to mimic human motion [88].

A few research studies have accomplished bipedal walking with stretched knees. Ogura *et al.* proposed a redundant mechanism with predetermined knee joint trajectories as initial walking parameters [86, 87]. Morisawa *et al.* proposed a pattern generation method with the CoG motion constrained on a parametric surface [79]. The feature of this method is that the walking pattern is divided into a time part and a spatial part. Kurazume *et al.* proposed a method of generating straight legged walking pattern by controlling the height of the CoG trajectory according to the state of the ZMP controller [66]. Sekiguchi *et al.* have so far proposed the Singularity Consistent approach which can handle singularities with stable con-

trol. In [101], they presented a walking pattern generation method through the neighborhood of the singularity by using ankle control based on the spherical inverted pendulum model. Kim *et al.* considered the knee joint fully stretched while designing the hip trajectory in the sagittal plane [60]. Handharu *et al.* solved the singularity problem in gait pattern generation by adding an extra joint in the heel in order to obtain another degree of freedom during knee stretch motion [19]. Park *et al.* also proposed a method to generate natural locomotion trajectory without bending knees [92].

Online vs. Offline

Trajectory generation method can be divided into two major schemes: online trajectory generation and offline trajectory generation.

The online trajectory generation method generates the walking pattern in real time, feeding back the present state of the system in accordance with the pre-provided goal of the motion. In this method, specific trajectories, precision, and repeatability are not important factors and the motion can be different in each step [5]. Planning and control are unified in this method [113].

Over the past decade, studies on real-time planning of a humanoid robot's gait have been published. In addition to "i-WALK" by Honda, Kajita *et al.* generated the real-time walking pattern for a biped robot with telescopic legs based on the 3D-LIPM [46]. In [43], they proposed a CoM trajectory generation method using the preview controller which involved the future ZMP reference trajectory to minimize the objective function. Lim *et al.* considered smoothly connecting the newly generated patterns to the current ones [69]. Nishiwaki *et al.* realized online generation of the desired walking pattern for the humanoid robot H6 by constructing typical stepping patterns offline in advance and then mixturing and connecting these patterns online [83]. For the humanoid robot H7, they proposed a method in which the horizontal position of the upper body trajectory is modified online to follow the desired ZMP trajectory. The walking pattern was updated by means of connecting the newly calculated trajectories to the current ones [82]. Sugihara *et al.* generated the ZMP and CoG trajectories online that track the desired trajectories by solving quadratic programming (QP) problem [112]. In [113], they realized online motion generation by controlling the CoG via indirect manipulation of the ZMP. Dimitrov *et al.* proposed a scheme for online walking pattern generation by solving a receding horizon LQR problem, i.e. QP problem [14]. Harada *et al.* proposed real-time and quasi real-time trajectory connection methods [21]. The smooth connection was realized by setting the ZMP parameters of the new trajectories as unknown constants in order for the continuity of the velocity of the CoG. Planning a humanoid robot's gait in real-time allows humanoid robots to better adapt to the environment.

Offline trajectory generation method generally generates a set of dynamic stable trajectories in advance and then tracks these pre-computed reference trajectories. It divides motion generation of humanoids into two subproblems, planning and control.

The offline trajectory generation method has been adopted mostly using the ZMP stablity criterion. Takanishi *et al.* in 1972 realized the complete dynamic walking of the biped walking robot WL-10RD on a flat floor [115]. In the single support phase, the control method was program control using preset walking pattern; in the double support phase, dynamic walking was achieved by sequence control using the predefined torque and mechanical impedance of both ankles. Kagami *et al.* proposed a walking pattern generation method by discretizing the ZMP equation as a trinomial expression. The torso trajectory was generated by iterative

calculation until the ZMP error was within the preset threshold [40]. Nagasaka *et al.* proposed the optimal gradient method to robot's gait planning [80]. Besides, the central pattern generator (CPG) [38, 62] and energy consumption optimization have also been proposed to plan the walking pattern of humanoid robots.

The motion plan in this thesis is offline, based on a point mass model, aims at natural and human-like walking patterns of humanoid robots.

3.1.2 Clues from Human Walking

Locomotion is the act of moving from place to place by means of one's own mechanisms or power. Locomotion in human beings is the result of the action of the body levers propelling the body. *Walking* is accomplished by the alternating action of the two lower extremities. It is an example of translatory motion of the body as a whole brought about by rotary motion of some of its parts [18].

Nature has solved efficiently the problem of biped locomotion during the long evolution of human specie. Therefore, the analysis of basic anthropomorphic walking may be a source of information to tackle the question of biped humanoid walking. Human walking has been widely studied and described in the literature. We do not recall all the issues of human walking, but describe the main walking principles that are the base of our method of planning the walking pattern in this thesis. Characteristics in human walking are as follows.

Characteristic A

Human walking is a periodic process. During this process, four different situations arise in sequence which can be depicted by a state machine [12] in Figure 3.3: (1) the statically stable double-support phase (DSP) in which the mechanism is supported on both feet simultaneously. The DSP starts as soon as the swinging leg meets the ground and ends when the support leg leaves the ground. In human walking cycle, the period of this phase is considered to be 10-20 % of the whole cycle [99]; (2) The pre-swing phase in which the heel of the rear foot is lifting from the floor but the toes of this foot are still on the floor; (3) Statically unstable single-support phase (SSP) in which only one foot of the mechanism is in contact with the ground while the other is being transferred from the back to front positions; and (4) the post-swing phase in which the toe of the front foot is declining towards the floor and the heel of this foot is contacting the floor.

Characteristic B [99]

In human walking, human body must slow down and then speed up again during each step because the support provided by the legs does not remain directly under the body at all times. Human walking has been partitioned into three distinct stages, as shown in Figure 3.4(a): 1) Development stage (from rest to some velocity); 2) Rhythmic stage (some constant average velocity); and 3) Decay stage (coming back to rest).

Characteristic C [99]

In normal walking, the CoM describes a smooth sinusoidal curve when projected on the plane of progression. The summits of these oscillations appear at about the middle of the stance

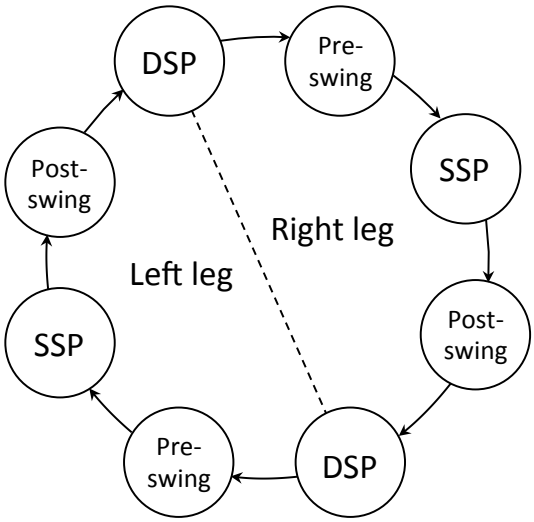


Figure 3.3: A state machine depicting periodic process of human walking.

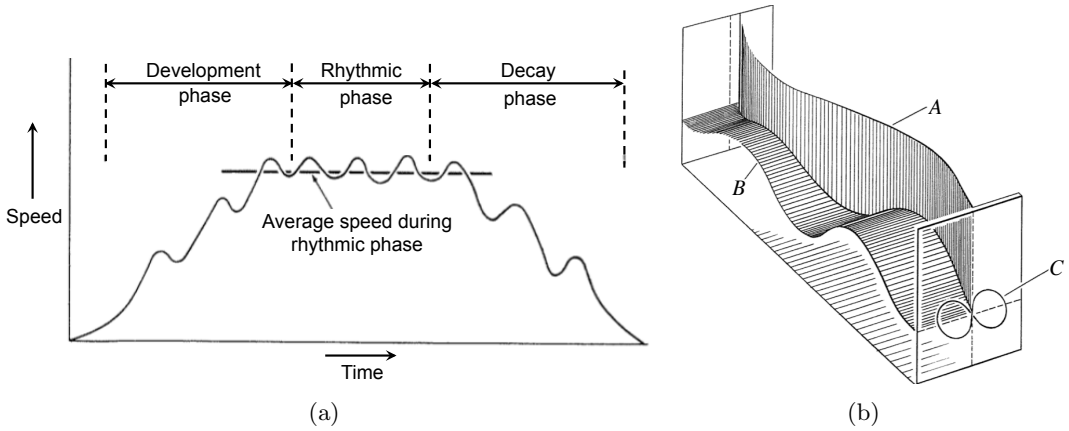


Figure 3.4: Movement of the CoM during human walking from [99]. (a) Three stages of the CoM during human walking. (b) Displacements of the CoM in three planes of space during single stride. A: Lateral displacement; B: Vertical displacement; C: Combined displacements of A and B.

phase of the supporting limb. The opposite limb is at this time in the middle of its swing phase. The CoM falls to its lowest level during the middle of double weight-bearing, when both feet are in contact with the ground. The CoM of the body moves also laterally in the horizontal plane. In this plane it describes a sinusoidal curve as well, but at one-half the frequency of the vertical displacement (Figure 3.4(b)).

3.2 Robot Model and Assumptions

In this thesis, we model the robot as one point mass concentrated on the position of the robot waist. Therefore, the walking related dynamics of the humanoid robot can be simplified as the equations of motion of the point mass, as shown in Figure 3.5.

The trajectories of the robot are defined in terms of cartesian coordinates. The global

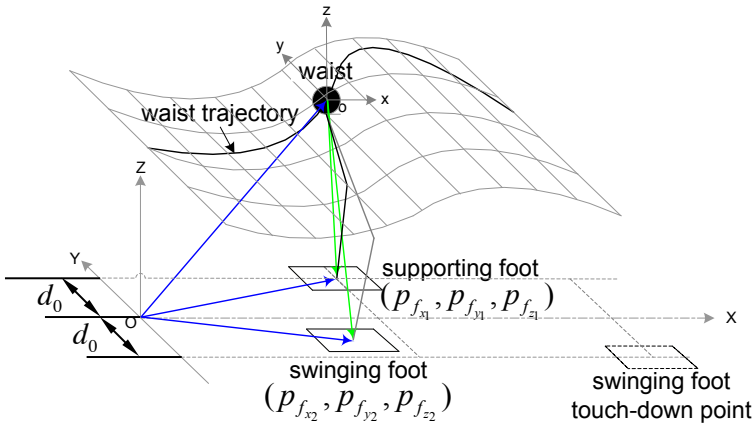
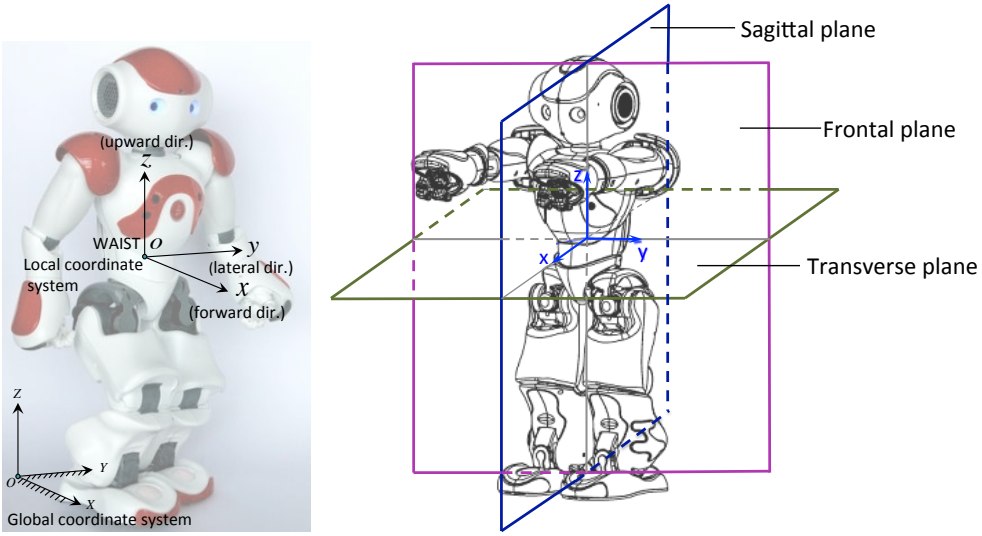


Figure 3.5: Dynamic walking model.

coordinate system OXYZ (see Figure 3.6(a)) fixed on the ground indicates the absolute positions of the waist and the two feet. The Z-axis is vertical, and the X-axis stands for the walking direction of the robot. The X-axis and Y-axis form a plane which is parallel to that of the floor. The local coordinate frame oxyz is attached on the waist, so it is used to indicate the relative positions of the two feet with respect to the waist. Figure 3.6(b) demonstrates anatomical reference planes of NAO. XY plane and XZ plane are called the transverse plane and the sagittal plane, respectively. The lateral movements of the waist lie in the transverse plane and the waist moves up and down within the sagittal plane.



(a) (b)

Figure 3.6: (a) Coordinate systems and (b) anatomical reference planes of NAO[1].

For simplicity, the following assumptions are made.

[1] Copyright©: Aldebaran Robotics.

(1) The floor for walking is rigid and cannot be moved by any forces and moments.

(2) The robot walks straight forward, so that the distance of two feet to their midline (symbolized by d_0) is a constant.

(3) Humanoid walking is a cyclic process with body movements repeated over and over. So we only need to describe the walking pattern in the course of one walking cycle, namely one stride.

(4) The contact region between the foot and the floor is a set of points.

Based on the assumption (3), supposing that the period necessary for one walking stride (see Figure 3.11) is T_{stride}, the time for the k^{th} stride walking is from kT_{stride} to $(k+1)T_{stride}$, $k = 0, 1, 2, 3, \ldots$. To simplify the analysis, we define that the k^{th} walking stride begins with the beginning of a SSP, when the right foot starts leaving the ground at $t = kT_{stride}$ while the left foot is stationary on the ground. The right foot swings from the rear to the front, and touches down onto the ground when a DSP is formed. Then the left foot lifts up and moves forward while the right foot is in contact with the ground. The k^{th} walking stride ends at $t = (k+1)T_{stride}$ with the end of the DSP when the right foot starts lifting up again.

3.3 Waist Reference Trajectory Generation

To generate a smooth trajectory, it is essential that the functions used for designing the walking pattern be differentiable and their second derivative be continuous. For this purpose, polynomial and cosine functions are used here. $\boldsymbol{p}_w(t) = [p_{w_x}(t), p_{w_y}(t), p_{w_z}(t)]$ is used to describe the desired waist trajectory, where $p_{w_x}(t)$, $p_{w_y}(t)$ and $p_{w_z}(t)$ stand for the waist motion in the forward direction, in the transverse plane and in the sagittal plane, respectively.

3.3.1 Trajectory in the Forward Direction

We describe each step in the rhythmic phase of human walking using a mathematical model (Figure 3.7). The model contains three phases of the waist motion in one step: speed-up phase, constant velocity phase and slow-down phase. The maximum velocity of the waist during one step is V_{max} and the minimum velocity is V_{min}. The forward walking speed for one step $V_{w_{x\_r}}(t)$ can be formulated by Eq.(3.1).

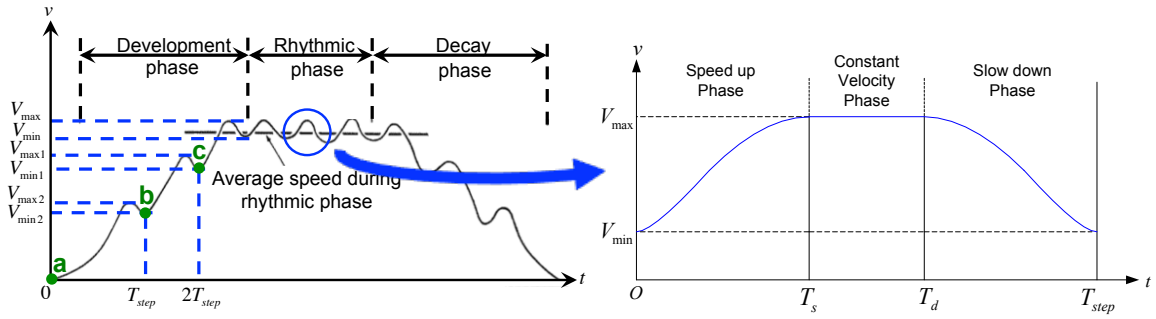


Figure 3.7: Mathematical model of one step mimicking the rhythmic phase in human walking.

$$V_{w_{x\_r}}(t) = \begin{cases} 0.5(V_{max} - V_{min})(1 - \cos\frac{\pi t}{T_s}) + V_{min} & (0 \le t < T_s) \\ V_{max} & (T_s \le t < T_d) \\ 0.5(V_{max} - V_{min})(1 - \cos\frac{\pi(t - T_d - T_s)}{T_s}) + V_{min} & (T_d \le t < T_{step}) \end{cases} \tag{3.1}$$

where T_s and T_d are the start time and ending time of the DSP respectively; T_{step} is the time for one walking step, $T_{step} = 0.5T_{stride} = T_s + T_d$. If we use double support ratio τ_{dsp} to denote the portion of the double-support phase in a walking cycle, then $T_s = 0.5T_{step}(1 - \tau_{dsp})$ and $T_d = T_{step}\tau_{dsp} + T_s$. t is the system time, and the subscript "w_{x_r}" indicates a certain physical quantity of the waist in the rhythmic phase along the X-axis (in the forward direction).

If we set $A_m = \pi(V_{max} - V_{min})/2T_s$, then A_m is the maximum acceleration. The corresponding acceleration in the process of one walking step $a_{w_{x\_r}}(t)$ is

$$a_{w_{x\_r}}(t) = \begin{cases} A_m \sin\frac{\pi t}{T_s} & (0 \le t < T_s) \\ 0 & (T_s \le t < T_d) \\ A_m \sin\frac{\pi(t - T_d - T_s)}{T_s} & (T_d \le t < T_{step}) \end{cases} \tag{3.2}$$

Based on Eq.(3.1), the absolute position of the waist $p_{w_{x\_r}}(t)$ in one walking step in the rhythmic phase can be obtained through integration,

$$p_{w_{x\_r}}(t) = \int_0^{T_{step}} V_{w_{x\_r}}(t)dt \tag{3.3}$$

Then we have,

$$p_{w_{x\_r}}(t) = \begin{cases} 0.5(V_{min} + V_{max})t - \frac{(V_{max} - V_{min})T_s}{2\pi}\sin\frac{\pi t}{T_s} & (0 \le t < T_s) \\ 0.5(V_{min} + V_{max})T_s + V_{max}(t - T_s) & (T_s \le t < T_d) \\ 0.5(V_{max} - V_{min})(T_d - T_s) + 0.5(V_{min} + V_{max})t & \\ -\frac{(V_{max} - V_{min})T_s}{2\pi}\sin\frac{\pi(t - T_d - T_s)}{T_s} & (T_d \le t < T_{step}) \end{cases} \tag{3.4}$$

We use a 4^{th} order polynomial function to describe the waist trajectory of the robot in order to imitate the development phase and decay phase in human walking:

$$V_{w_{x\_d}}(t) = \sum_{i=0}^{4} a_i t^i \tag{3.5}$$

where the subscript "w_{x_d}" indicates a certain physical quantity of the waist in the development phase along the X-axis (in the forward direction); $a_i (i = 0, 1, 2, 3, 4)$ are polynomial coefficients which are determined by boundary conditions. For example, for the first step of

the development phase of the robot the boundary conditions can be written as

$$\begin{cases} V_{w_{x\_d}}|_{t=0} = 0; \\ V_{w_{x\_d}}|_{t=T_{step}} = V_{min2}; \\ \dot{V}_{w_{x\_d}}|_{t=0} = 0; \text{[2]} \\ \dot{V}_{w_{x\_d}}|_{t=T_{step}} = 0; \text{[2]} \\ \ddot{V}_{w_{x\_d}}|_{t=T_{step}} = \dfrac{A'_m \pi}{T_s}. \text{[3]} \end{cases} \tag{3.6}$$

where $A'_m = \dfrac{\pi(V_{max2} - V_{min2})}{2T_s}$, V_{min2} and V_{max2} are the velocities shown in Figure 3.7.

Hence the waist position $p_{w_{x\_d}}(t)$ in the development phase and decay phase can be obtained through integration.

Figure 3.8 shows the waist velocity of the generated walking pattern in the forward direction.

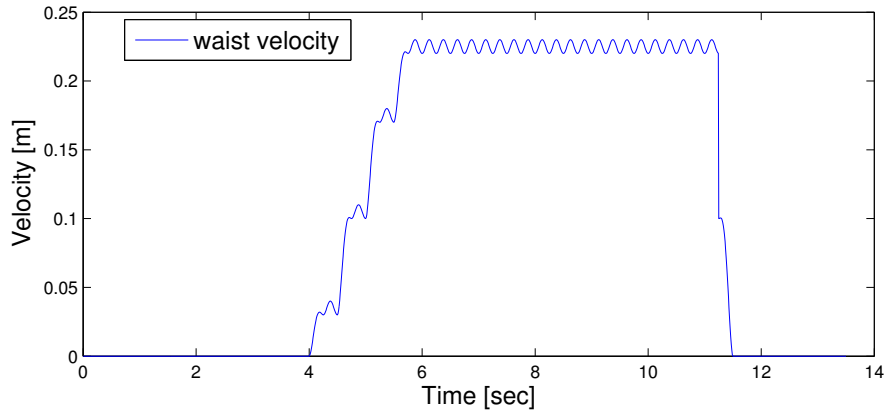


Figure 3.8: Waist velocity of the generated walking pattern in the forward direction. The horizontal axis indicates the time of the whole walking process. Before the robot walks, the posture of the robot will first change from squatting to standing. To make sure that the robot starts walking from being stationary, the robot will keep the standing posture for 4 seconds. So the data was recorded from the end of the fourth second.

3.3.2 Lateral Trajectory in the Transverse Plane

As illustrated in Figure 3.9, compared with the other walking steps, the first walking step (OABC) of the robot is a special process in step length and lateral swing amplitude since

[2] It is desirable that in the development phase the gradients of the tangent lines of the function at time $t = 0$, $t = T_{step}$, $t = 2T_{step}$ (the points like a, b and c in Figure 3.7) are zero.

[3] To generate a trajectory that smoothly connects the development phase and rhythmic phase, it is essential that the second derivative of the functions be continuous. Differentiating Eq. (3.2) in the rhythmic phase, we get $\dot{a}_{w_{x\_r}}|_{t=0} = \dfrac{A_m \pi}{T_s}$. Therefore, in the development phase $\ddot{V}_{w_{x\_d}}|_{t=T_{step}}$ must be $\dfrac{A_m \pi}{T_s}$. Here we use A'_m, it depends on how many steps the robot uses in the development phase to reach the rhythmic phase.

the robot accelerates from being stationary to some speed in this walking step. Similarly, the robot decelerates to stop walking in the last walking step. The lateral swing motion of the waist is essential to move the ZMP to each sole during walking. In order to guarantee the ZMP at the sole area of the support foot (here, right foot), the robot must lean to the right side at the beginning of walking from point O to point A.

We use a 3^{rd} order polynomial function to describe the waist trajectory for the first step in the transverse plane.

$$p_{w_y}(t) = \sum_{i=0}^{3} a_i t^i \qquad (0 \leq t < T_{step}) \tag{3.7}$$

where $a_i (i = 0, 1, 2, 3)$ are polynomial coefficients which are determined by the boundary conditions as follows:

$$\begin{cases} p_{w_y}|_{t=0} = 0; \\ p_{w_y}|_{t=T_{step}} = A_l; \\ \dot{p}_{w_y}|_{t=T_{step}} = 0; \\ \ddot{p}_{w_y}|_{t=T_{step}} = -\omega^2 A_l.^{4} \end{cases} \tag{3.8}$$

with ω being the frequency of the function that describes the lateral walking trajectory of the waist for the succeeding steps, and A_l being the lateral swing amplitude.

From $p_{w_y}|_{t=0} = 0$ we obtain $a_0 = 0$. The boundary conditions in Eq. (3.8) can be expressed by a linear system equation

$$Ax = b$$

with a solution $x = A^{-1}b$, where

$$A = \begin{pmatrix} {T_{step}}^3 & {T_{step}}^2 & T_{step} \\ 3{T_{step}}^2 & 2T_{step} & 1 \\ 6T_{step} & 2 & 0 \end{pmatrix}, \quad x = \begin{pmatrix} a_3 \\ a_2 \\ a_1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} A_l \\ 0 \\ -\omega^2 A_l \end{pmatrix}.$$

Figure 3.10 illustrates the lateral trajectory of the waist for the first step in the transverse plane.

The lateral path of the waist for the succeeding walking steps can be described by the following cosine function.

$$p_{w_y}(t) = A_l \cos \omega(t - t_0) \qquad (t \geq T_{step},\ t_0 = T_{step}) \tag{3.9}$$

3.3.3 Trajectory in the Sagittal Plane

The trajectory for the upward and downward waist motion in the sagittal plane is generated using a cosine function with its frequency being twice as that in the lateral plane (see Characteristic C). In the speed-up phase, the robot descends gradually from the summit z_{max} until the DSP starts. In the constant velocity phase, the robot maintains the lowest level z_{min}. In

[4] It is also possible to describe the waist trajectory of the first step in the transverse plane by using a 4^{th} order polynomial function. Besides the boundary conditions in Eq. (3.8), another boundary condition $\dot{p}_{w_y}|_{t=0} = c$ (c is a constant) is needed. Note that in order to make the robot lean to the right side at the beginning of walking from point O to point A, c should be a negative value. However, the 4^{th} order polynomial function has been verified neither through simulation nor through experiments on the real robot.

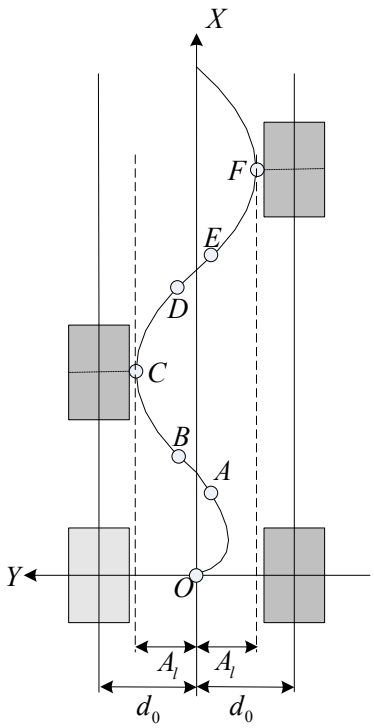


Figure 3.9: Lateral waist path.

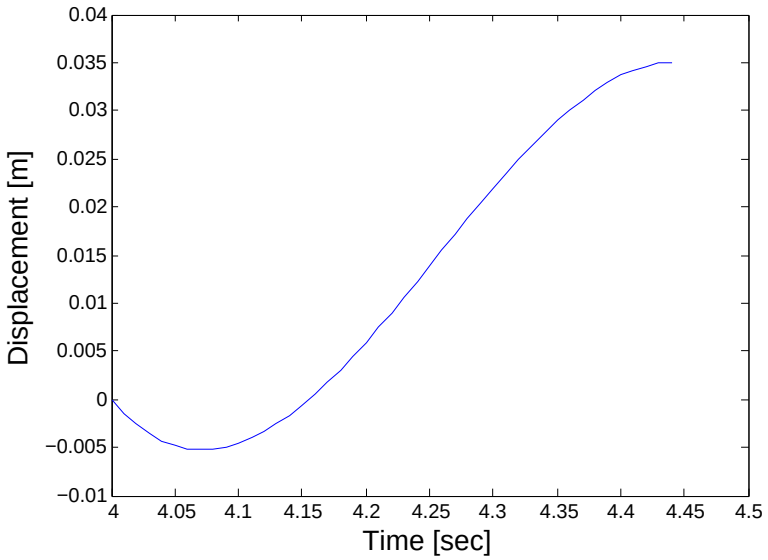


Figure 3.10: Lateral waist trajectory for the first step in the transverse plane. Parameters for plotting: $A_l = 0.035$ m, $T_{stride} = 0.88$ s.

the slow-down phase, the robot rises to the peak height z_{max} again (refer to Figure 3.11). The upward and downward motion of the waist in the sagittal plane can be formulated by

Eq.(3.10).

$$p_{w_z}(t) = \begin{cases} 0.5(z_{max} - z_{min})(1 - \cos\frac{\pi(t - T_s)}{T_s}) + z_{min} & (0 \leq t < T_s) \\ z_{min} & (T_s \leq t < T_d) \\ 0.5(z_{max} - z_{min})(1 - \cos\frac{\pi(t - T_d)}{T_s}) + z_{min} & (T_d \leq t < T_{step}) \end{cases} \tag{3.10}$$

$A_s = 0.5(z_{max} - z_{min})$ is defined as sagittal oscillation amplitude.

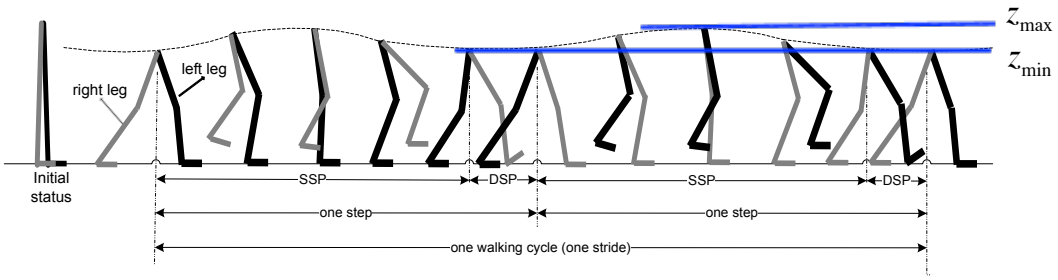


Figure 3.11: Waist motion in the sagittal plane.

3.4 Foot Reference Trajectory Generation

We use a cycloid function to describe the trajectory of the swinging foot because the cycloid function describes a path of a certain point on the circumference of a circle during circling, which is similar to the human ankle circling the tiptoe [58]. $\boldsymbol{p}_f(t) = [p_{f_x}(t), p_{f_y}(t), p_{f_z}(t)]$ represents the reference foot trajectory in the global coordinate system.

Take the left foot as an example, the trajectory of the foot is

$$\begin{cases} p_{f_x}(t) = \alpha[\psi(t) - \sin(\psi(t))] \\ p_{f_y}(t) = const. \\ p_{f_z}(t) = \delta \cdot \alpha[1 - \cos(\psi(t))] \end{cases} \tag{3.11}$$

where $\psi(t) = 2\pi t / T_{ssd}$ ($T_{ssd} = 2T_s$ is the single support duration in one step), $0 \leq \psi(t) \leq 2\pi$. δ is a variable that keeps the maximum elevation of the foot (H_{foot}) fixed, here $\delta = H_{foot}/2\alpha$. α is a synchronization parameter which will be discussed in the following section.

The right foot trajectory is the same as the left foot trajectory except for a T_{step} delay. Figure 3.12 shows the trajectories of both feet in the sagittal plane.

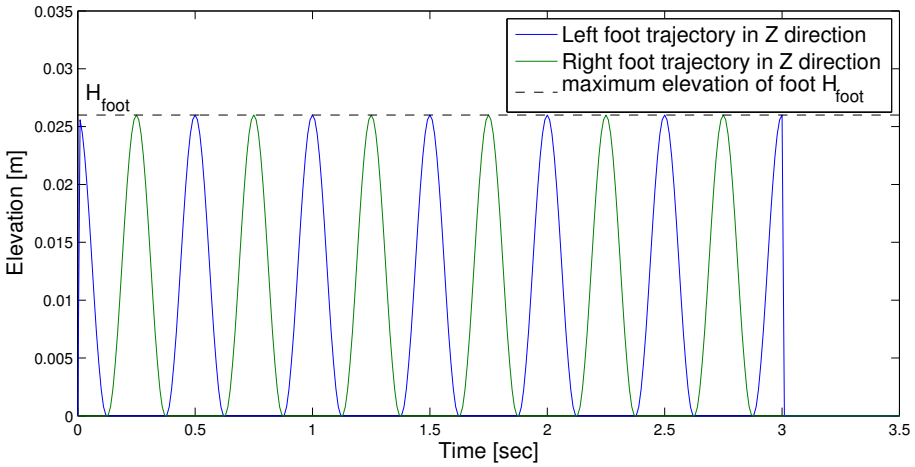


Figure 3.12: Foot trajectories using cycloid function.

3.5 Simultaneous Planning of Waist and Foot Motion

The fundamental idea for synchronizing the motion of the waist and the two feet is that

$$p_{w_x}(t) = p_{f_{x1}}(t) = p_{f_{x2}}(t) \tag{3.12}$$

holds at all $t = jT_{step}, j = 0, 1, 2, 3, \ldots$, where $p_{f_{x1}}(t)$ and $p_{f_{x2}}(t)$ are the foot positions in the forward direction.

Eq.(3.12) can be written as

$$p_{w_x}|_{(t=jT_{step})} = p_{f_{x1}}|_{(t=jT_{step})} = p_{f_{x2}}|_{(t=jT_{step})} \tag{3.13}$$

The synchronization parameter α can be calculated. For example, in the rhythmic phase Eq. 3.13 can be rewritten as

$$j(V_{max}T_d + V_{min}T_s) = j\pi\alpha, \quad j = 1, 2, 3, \ldots \tag{3.14}$$

Then we obtain

$$\alpha = \frac{V_{max}T_d + V_{min}T_s}{\pi} \tag{3.15}$$

3.6 ZMP Trajectory

ZMP is used to evaluate the stability of the generated walking pattern. We use Eq.(2.5) in Section 2.1 to calculate the position of the ZMP. We assume that the rate of the change in angular momentum around the CoG during walking is negligible since it is small compared with the influence of the motion of the CoG. Then in Eq.(2.5), $\dot{L}_{w_x} = 0$ and $\dot{L}_{w_y} = 0$. Hence the position of ZMP $\boldsymbol{p}_{zmp}(x_{zmp}, y_{zmp}, z_{zmp})$ in the global coordinate frame can be expressed as

$$\begin{cases} x_{zmp} = p_{w_x} - \dfrac{(p_{w_z} - z_{zmp})}{\ddot{p}_{w_z} + g}\ddot{p}_{w_x} \\ \\ y_{zmp} = p_{w_y} - \dfrac{(p_{w_z} - z_{zmp})}{\ddot{p}_{w_z} + g}\ddot{p}_{w_y} \end{cases} \tag{3.16}$$

Here, we assume that the robot walks on an ideal horizontal surface, then $z_{zmp} = 0$. Note that since the waist motion is not constrained on a two-dimensional surface but follows a three-dimensional curved surface, $\ddot{p}_{w_z}$ is not equal to zero.

Up to now, we have the trajectories of the waist and the two feet in the global coordinate system. The relative trajectories of both feet with respect to the waist $\tilde{\boldsymbol{p}}_f(t)$ are derived following the relationships:

$$\tilde{\boldsymbol{p}}_f(t) = \boldsymbol{p}_f(t) - \boldsymbol{p}_w(t)$$

Then the reference trajectory of each joint is calculated by solving the inverse kinematics[5] at an interval of 10 ms (100 Hz). The reference joint positions are controlled by local PID joint controllers. Figure 3.13 shows the outline of the walking pattern generation process, where $\tilde{p}_{left}$ and $\tilde{p}_{right}$ denote the relative positions of the left foot and the right foot with respect to the waist, respectively.

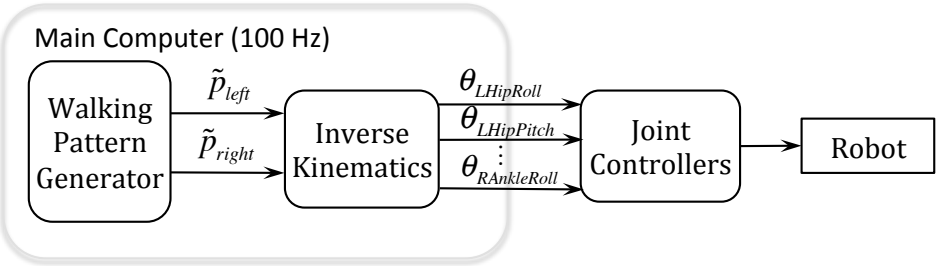


Figure 3.13: The process of generating the walking pattern.

Figure 3.14 shows the three-dimensional trajectory of the waist following the curved surface. Compared with the walking patterns based on 3D-LIPM, the waist has upward and downward motion in the vertical direction. Trajectories of the waist and both feet in the forward direction are plotted in Figure 3.15. Figure 3.16 illustrates the waist and ZMP trajectories in the transverse and sagittal planes. The waist trajectory in the sagittal plane has twice the frequency of that in the transverse plane, which is similar to the mechanism of human walking. Figure 3.17 shows the pitch angle trajectories of the knee joints of both legs. We can see that the minimum pitch angles are close to zero. Compared with the conventional walking patterns which need the humanoid robots to bend the knees all the time, the walking pattern proposed in this thesis is more natural. The stick diagrams[6] of the presented walking pattern for both legs are shown in Figure 3.18.

3.7 Software Simulation

3.7.1 Software Overview

When working with robots, the usage of simulation is often of significant importance. On the one hand, executing robot programs in a simulator offers the possibility of directly debugging

[5]For the details of the inverse kinematics of the robot NAO, refer to [56].

[6]The stick diagram is used to illustrate the trajectories of the hip, the knee and the ankle of the robot, as well as to depict the configurations of the leg at a certain time interval throughout the process of walking.

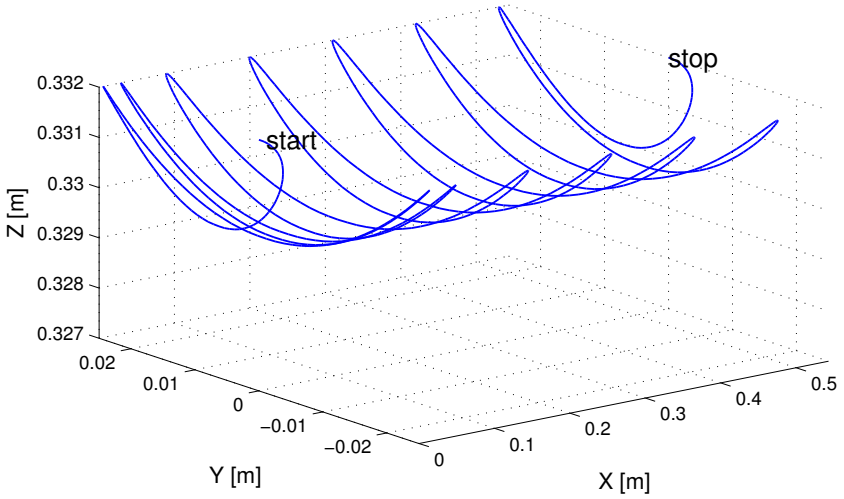


Figure 3.14: Waist trajectory following three-dimensional curved surface. Walking parameters for plotting: $A_l = 0.025$m, $V_{min2} = 0.01$ m/s, $V_{max2} = 0.02$ m/s, $V_{min1} = 0.03$ m/s, $V_{max1} = 0.04$ m/s, $V_{min} = 0.09$ m/s, $V_{max} = 0.10$ m/s, $z_{min} = 0.330$ m, $z_{max} = 0.332$ m, $H_{foot} = 0.026$ m and $T_{stride} = 1.16$ s. In the development phase of the walk, the forward velocity of the waist of the robot is gradually increased (refer to Fig. 3.7 (left)). The ascending velocities in the development phase are represented by V_{min2}, V_{max2}, V_{min1}, V_{max1}, respectively. In the steady phase of the walk, the velocities are denoted by V_{min} and V_{max}.

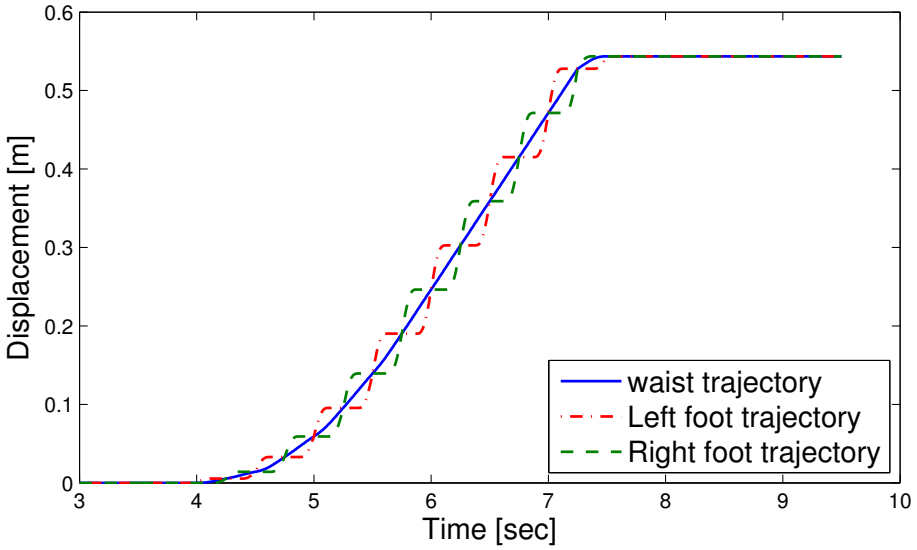


Figure 3.15: Waist and foot trajectories in the forward direction.

and testing the programs. On the other hand, it supports the process of software development by providing an replacement for robots. For example, developing and verifying various control algorithms using a software simulator before using a real robot could avoid damages to the real robot.

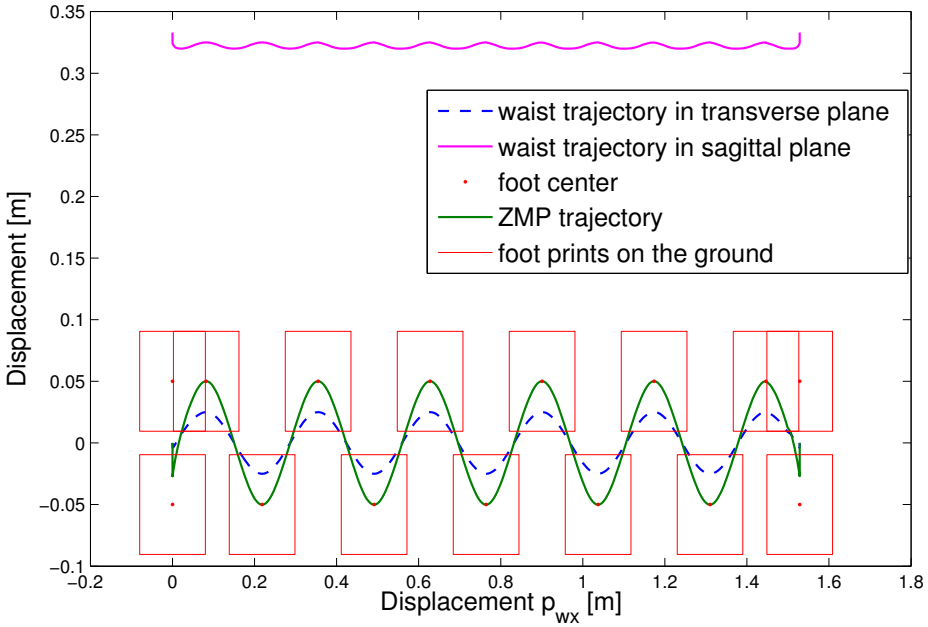


Figure 3.16: Waist motion and ZMP trajectory in the transverse and sagittal planes. Parameters for plotting: $V_{min} = 0.23$ m/s, $V_{max} = 0.24$ m/s, $A_l = 0.025$ m, $T_{stride} = 1.16$ s.

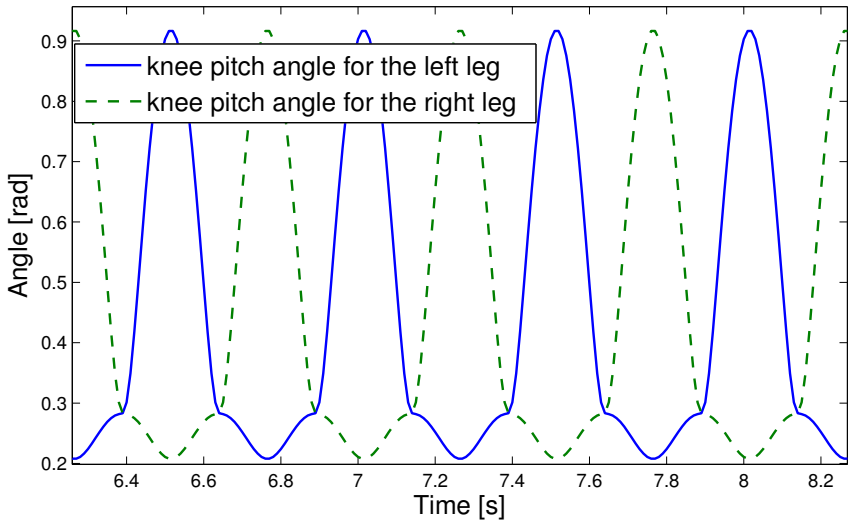


Figure 3.17: Knee pitch angles for the left and right legs.

To examine the feasibility of the presented walking pattern and verify the validity of the control algorithms in this thesis, the simulation software SimRobot was used (see Figure 3.19). SimRobot [67] is a kinematic robotics simulator developed in Bremen, Germany which utilizes the ODE[(1)] to approximate solid state physics. It is an open source which allows to adapt the code for respective software developments. In addition, SimRobot provides a convenient

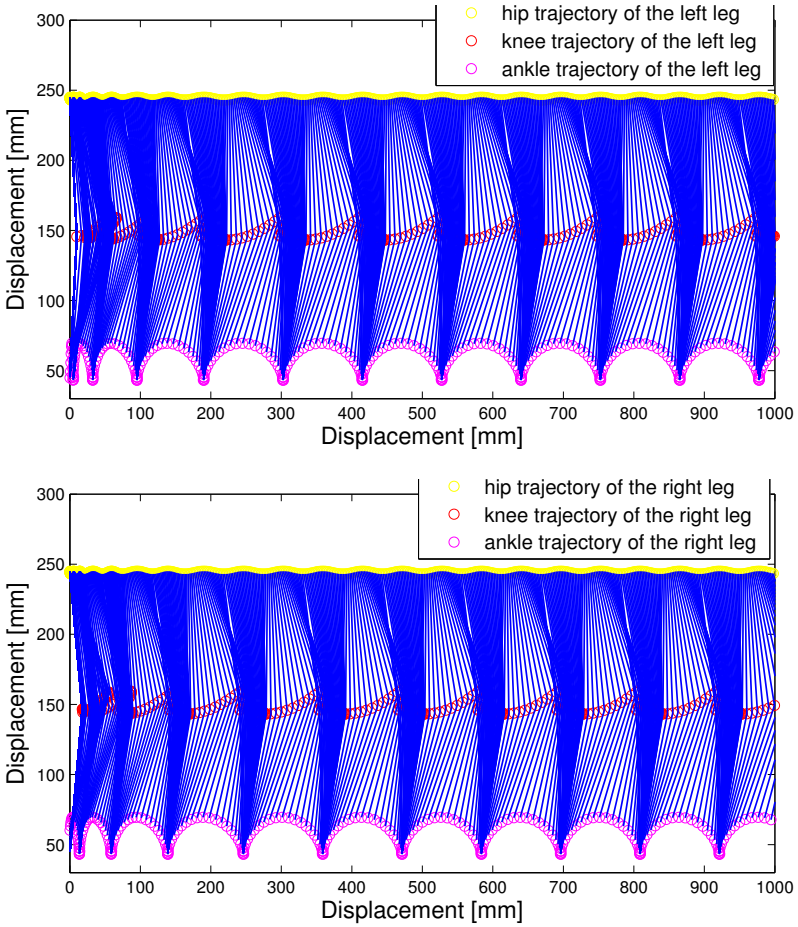


Figure 3.18: Stick diagrams of the walking pattern. (top) Left leg (bottom) Right leg.

way for software development because it can directly connect to the robot and implement online debugging. By using update steps of up to 1 kHz for the physics engine, the simulated walking closely matches the gait of the real robot [118].

3.7.2 Simulation Results

Figure 3.20 shows the snapshots of the forward walking in the simulator. In simulation, the robot increased the velocity from a stationary state at the beginning of the walk. During steady walking, the minimum velocity in one walking step was V_{min}= 0.22 m/s and the maximum velocity was V_{max}= 0.23 m/s. After steady walking with a stride of 0.1059 m, the robot reduced the velocity and stopped again. Total simulation time was 22.5 seconds. Several computer simulations demonstrated the stable walking of the robot in the simulator and indicated the feasibility of applying the presented walking pattern to the real robot.

3.8 Walking Experiments

Software simulation has indicated the feasibility of applying the presented walking pattern to the real robot. To further test the dynamic stability of the walking pattern fulfilled on the real

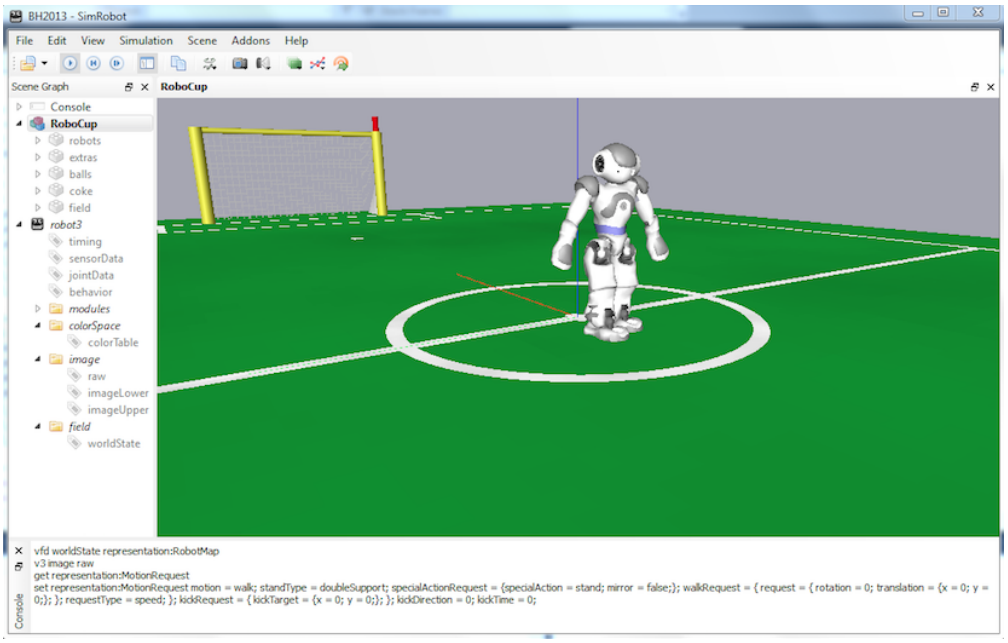


Figure 3.19: Walking simulation in SimRobot.

robot, we carried out walking experiments on the humanoid robot NAO. The robot walked on a flat and noninclined floor but with local unevenness. In the experiments, the robot walked in all 61 steps in 15 seconds.

We have taken a large number of experiments with different sets of walking parameters (T_{stride}, τ_{dsp}, A_l, z_{min}, z_{max}, H_{foot}, V_{min} and V_{max}). For each experiment we adjusted these parameters within the reasonable ranges. The experiments showed that with the presented walking pattern, the robot could walk; however, the walks with various sets of walking parameters were all unstable. The problem was that the upper body of the robot swayed back and forth during walking, and the amplitude of the oscillation became larger as the robot walked. After a few steps the robot fell down. Figure 3.21 illustrates the inclination of the upper body of the robot in the course of a walk which is the output of the walking pattern generator without any feedback control. It is found that the inclinations of the upper body about the Y-axis (pitch angles) are several times larger than 0.4 rad and the maximum pitch angle of the upper body is more than 1.0 rad. It is sufficient to make the robot fall down and thus the walking was unstable. Therefore, in view of the environment uncertainty such as the unevenness and inclination of the surface that may cause the walking unstable, a posture control method to compensate for the body inclination of the humanoid robot aiming at stable dynamic walking is necessary, and will be presented in Chapter 4.

3.9 Discussion

3.9.1 CoM Calculation

The CoM corresponds to the weighted average location of all the mass in the body. From a physics point of view, the body (even if oddly-shaped) could be represented by a point mass located at the CoM. Knowledge of the CoM is important to postural and stability control of

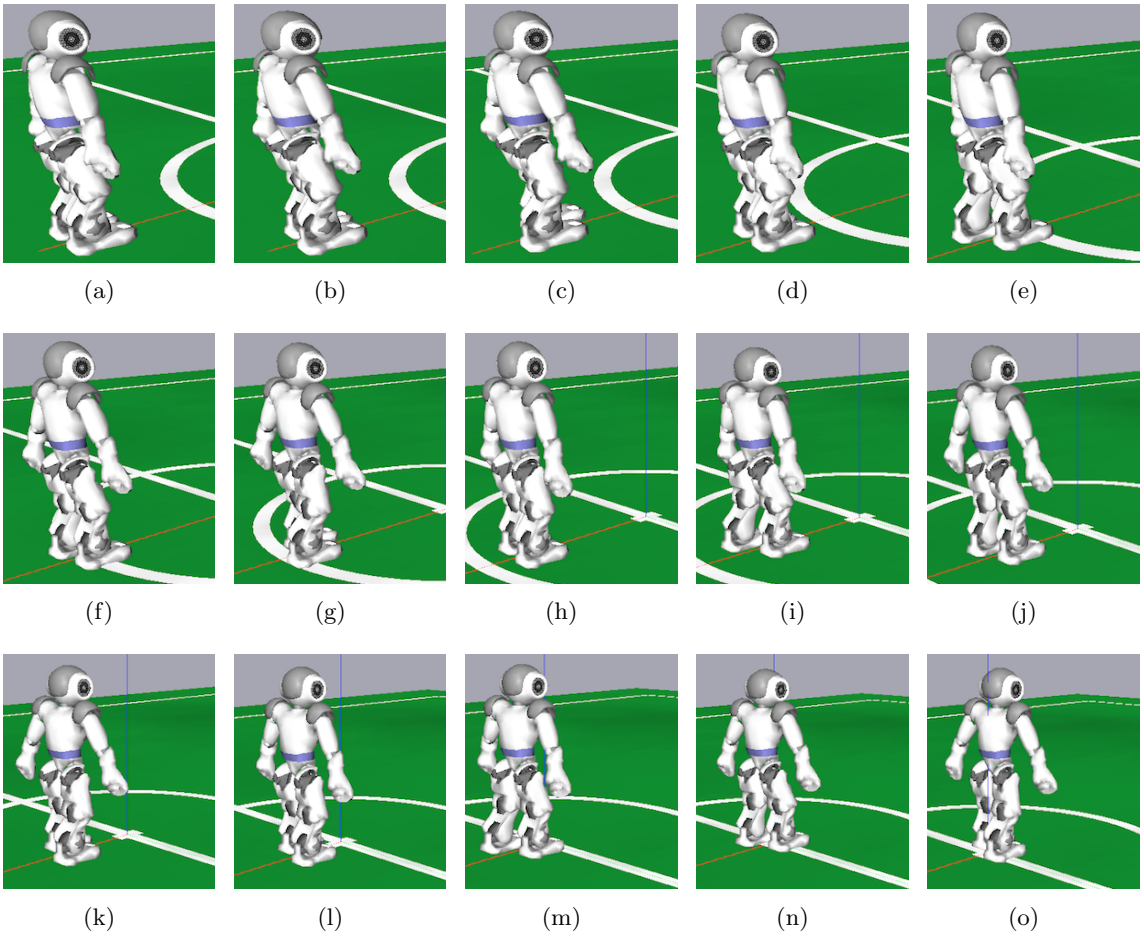


Figure 3.20: Snapshots of forward walking in SimRobot simulator.

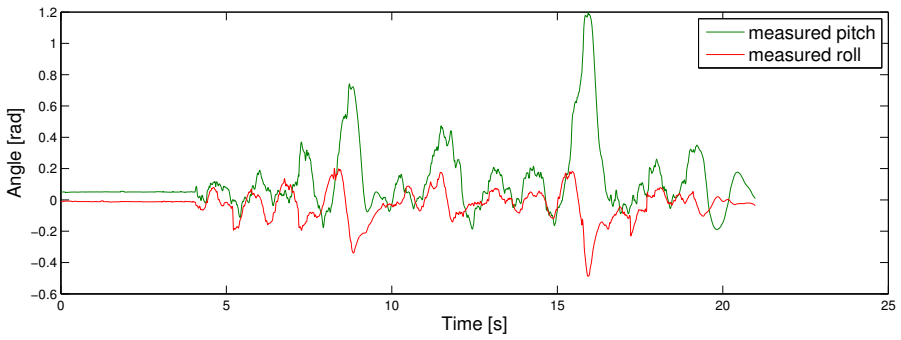


Figure 3.21: Body inclination without feedback control.

humanoid robots.

The robot consists of a group of connected parts, e.g. joints and the corresponding links. Each part has its own known mass and its own CoM (local CoM) at a known static position. During walking, the position of the CoM changes as the joint configuration of the robot changes. For any given configuration of the robot, forward kinematics can be used to locate

the exact position of the CoM of each part relative to the reference point by setting the position of the local CoM to be the end-effector.

Then the position of the CoM of the whole body can be calculated by Eq.(3.17).

$$p_{CoM} = \frac{\sum_{i=1}^{n} p_i m_i}{\sum_{i=1}^{n} m_i} \tag{3.17}$$

where n is the number of parts; p_i is the location of the CoM of each part relative to the reference point, as for NAO, the reference point is the origin of the local coordinate frame, namely the waist; m_i is the mass of each part, and p_{CoM} is the position of the CoM of the whole body.

3.9.2 Equivalence of the Waist and CoM in Designing the Walking Pattern

In this thesis, we investigated the walking pattern of a humanoid robot that involves the three-dimensional motion of the waist. The presented walking pattern generation approach takes the waist of the robot as the reference point. In most cases, motion planning of a humanoid robot is based on the CoM or CoG of the robot [10, 43, 45, 66, 73, 79]. As a matter of fact, the waist and the CoM of a robot do not overlap during walking since the robot waist is a fixed point (refer to Figure 3.6) whereas the CoM of the robot changes as the joint configuration of the robot changes. We found that designing the walking pattern of a humanoid robot can base either on the waist or on the CoM.

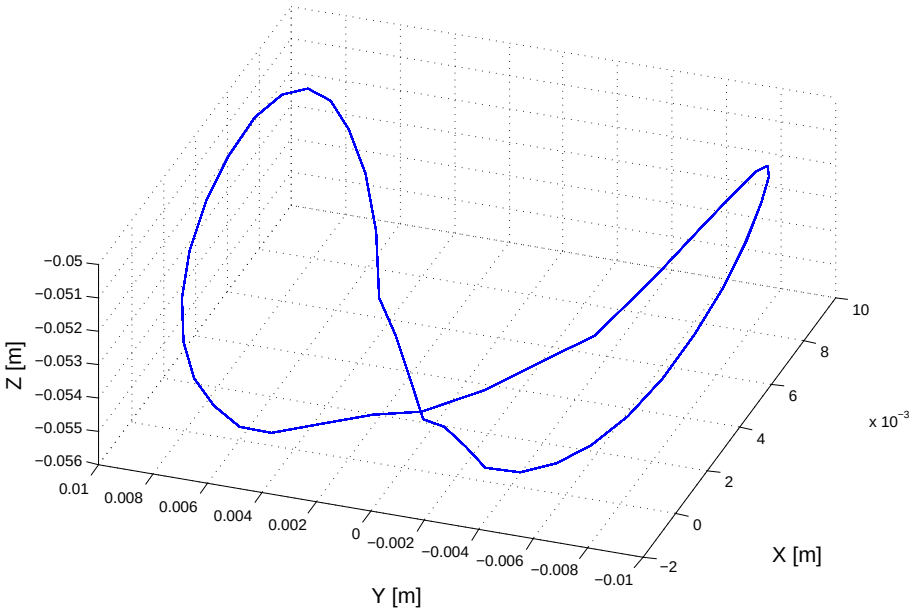


Figure 3.22: Spatial position of the CoM in the local coordinate frame. Parameters are T_{stride} = 0.88 s, A_l = 0.04 m, z_{min} = 0.325 m, z_{max} = 0.328 m, V_{min} = 0.24 m/s and V_{max} = 0.25 m/s.

Figure 3.22 plots the spatial trace of the actual CoM in the local coordinate system during walking. In order to facilitate the analysis of influence of the CoM position on the generated walking pattern, the spatial trace is decomposed into trajectories in x, y and z directions, as demonstrated in Figure 3.23. We can see that similar to the periodic motion of the waist,

the CoM movement is also periodic, and the position of the CoM during walking deviates from the waist. The CoM ranges from -2 mm to 8 mm in the x direction, and therefore has a slight effect on the forward motion of the generated walking pattern. The CoM oscillates laterally within ± 1.0 cm, and in the sagittal plane -5.1 cm to -5.6 cm. Therefore, though the waist in most of the time does not overlap with the CoM of the robot, generating a walking pattern based on the motion of the waist is equivalent to that based on the motion of the CoM providing the walking parameters such as A_l, z_{min} and z_{max} are appropriately determined.

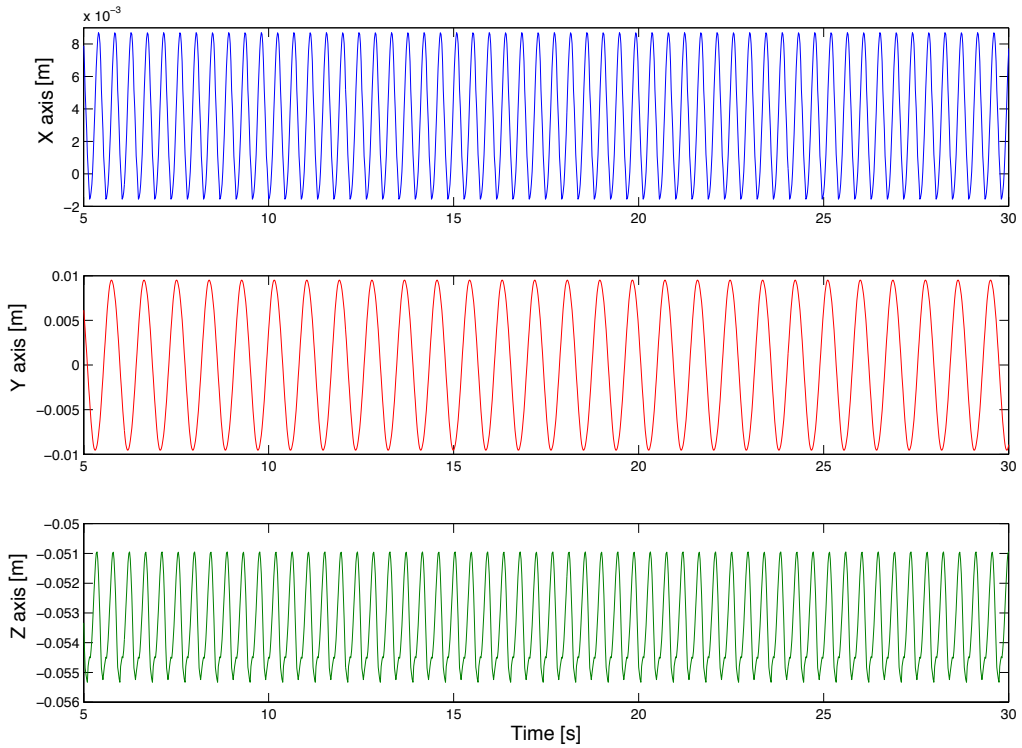


Figure 3.23: CoM position in x, y and z directions.

3.10 Chapter Summary

In this chapter, a walking pattern generation method based on the studies on human walking with the aim of achieving natural and efficient biped walking is presented. The walking pattern of a bipedal humanoid robot is modeled with continuous and differentiable mathematical functions. Then a pattern generator satisfying the ZMP criterion is designed. Compared with the walking pattern that restricts the motion of the upper body to a horizontal plane, the presented walking pattern involves three-dimensional upper body motions and leg configurations with almost stretched knees. The advantages of stretched-knee walking are that (i) a natural and human-like walking pattern is achieved, and (ii) since the robot does not always bend its legs, energy consumption to support the body weight is small at knee joints. The comparison of energy consumption between the conventional walking pattern and the presented walking pattern will be made in Chapter 4. Moreover, the presented method needs lower calculation costs compared with the method based on precise knowledge of robot dynamics. The software

simulation demonstrated the stable walking of the robot in the simulator and indicated the feasibility of applying the presented walking pattern to the real robot.

4 Posture Control

4.1 Introduction

Although the designed walking pattern ensures the dynamic stability of a humanoid robot, the robot may tip over during walking as the actual ZMP trajectory is different from the reference ZMP trajectory due to the modeling error, the environment uncertainty such as the unevenness of the surface, as well as disturbances. A loss of stability might result in a fall which may bring about serious consequence for the robot. Therefore, stability control is of great importance for the overall performance of the humanoid robot.

This chapter presents body posture control methods aiming at stable dynamic walking. The body posture control consists of two parts. One is online torso pitch/roll control which is based on sensory feedback and modifies the reference trajectories in real time in order to stabilize the robot. The other is yaw moment control which compensates the yaw moment around the support foot during walking by using arm movement. The control methods presented in this chapter have been applied to NAO both in simulation and in experiments.

The scheme of the overall walking control system is illustrated in Figure 3.2. The control system consists of a posture controller for the upper body, a position controller for the swinging leg, and a force controller for the landing leg. They are enumerated as follows:

- Body posture control
- Swing-leg trajectory tracking
- Landing impact control

Swing-leg trajectory tracking and landing impact control will be presented in Chapter 5 and Chapter 6, respectively. The online control system is generally applicable across the spectrum of bipedal humanoid robots.

4.2 Background and Related Works

From biological point of view, *posture* describes the orientation of any body segment relative to the gravitational vector. It is an angular measure from the vertical [125]. An anthropomorphic system is said to have *postural stability* if it possesses some regulating system which does not allow the ZMP to run away from the stable region under external perturbations [124].

4.2.1 Clues from Human Walking

The law of conservation of angular momentum states that the angular momentum is a conserved physical quantity for isolated systems when no external moments act on the body's CoM. In the case of legged locomotion, there is no a priori reason for the law to hold since the body interacts with the environment. However, biomechanical studies on human walking have revealed that the angular momentum is highly regulated by the central nervous system throughout a movement cycle [94] and remains small during steady-state human walking. The central nervous system manages to keep the large inertial load of head, arms and trunk erect within $\pm 1.5°$ [125].

4.2.2 Related Works

Many research studies have addressed the control strategy of body posture stabilization of humanoid robots. Löffler *et al.* controlled the orientation and rotational velocity of the upper body of a biped robot by regulating contact torques at the robot feet [73]. Huang *et al.* controlled the body posture of a humanoid robot constant by using reflex action of hip joints [33], whereas Sugahara *et al.* implemented a different strategy on a parallel leg mechanism by rotating the whole body around the foot via resolving inverse kinematics of the leg [110]. Kim *et al.* managed an upright robot posture by PI controlling the ankle joints of a robot that walks on uneven/inclined floors [59]. Stephens *et al.* achieved posture control by applying a virtual torque to the torso of a humanoid robot. The virtual torque is realized by full body joint torques calculated by means of dynamic balance force control method [108]. It is also possible to accelerate the upper body horizontally or to vary the step length to keep an upright posture [73]. In addition, some researchers stabilized walking robots by considering the angular momentum [44, 63, 78].

Being inspired by human walking as well as from the viewpoint of stability, it is desirable that the body posture is constant especially when the humanoid robot has no waist joint. The body posture controller in this thesis is composed of a torso pitch/roll controller and a yaw moment controller, which online controls the actual body posture of the robot upright (desired body posture) all the time.

4.3 Torso Pitch/Roll Control

4.3.1 Posture Detection

The real-time information on the upper body posture, i.e. the actual body orientation and angular velocity, is provided by the IMU which is a combination of two gyrometers and one accelerometer located in the torso. Table 4.1 shows the details of the IMU.

Table 4.1: IMU information of NAO.

IMU	Axis	Number	Precision	Position
Accelerometer	3-axis	1	1% (acceleration: ∼2g)	robot torso
Gyrometer	1-axis	2	5% (angular speed: ∼500°/s)	robot torso

When the robot stands still, the angular velocity of the torso measured by the gyrometers is around $-18°/s$ whereas the actual value should be around $0°/s$. So the data from the gyrometers needs to be calibrated. The calibration algorithm is as follows:

$$C_x := (1-\alpha)\cdot C_x + \alpha \cdot \dot{\theta}_m \tag{4.1}$$
$$C_y := (1-\alpha)\cdot C_y + \alpha \cdot \dot{\phi}_m \tag{4.2}$$
$$\dot{\theta}_c = \dot{\theta}_m - C_x \tag{4.3}$$
$$\dot{\phi}_c = \dot{\phi}_m - C_y \tag{4.4}$$

where (C_x, C_y) represent calibration angular velocities and they are updated in real time during the walk of the robot. ($\dot{\theta}_m$, $\dot{\phi}_m$) are raw measurement data obtained from gyrometers. α is a weight value. It determines to what extend the updated output samples (C_x, C_y) are dependent on the input measurement samples ($\dot{\theta}_m$, $\dot{\phi}_m$) and the preceding outputs. Here $\alpha = 0.001$, which means the preceding outputs contribute more to the updated calibration angular velocities while the measurements are not trustable and thus contribute less. ($\dot{\theta}_c$, $\dot{\phi}_c$) denote the calibrated angular velocities about X- and Y-axis.

4.3.2 Posture Estimation

Accurate estimation of the upper body state (e.g. orientation and angular velocity) is important for posture control. The estimates of unknown variables provided by the method of Kalman filter [49] tend to be more precise than those based on a single measurement alone since a Kalman filter operates recursively on streams of noisy input data in order for producing a statistically optimal estimate of the underlying system state.

State Estimator

The dynamics of the upper body can be written as a discrete linear system in state space form

$$x_{k+1} = Ax_k + Bu_k \tag{4.5}$$

where $x_k = [\theta_{pitch} \quad \phi_{roll} \quad \dot{\theta}_{pitch} \quad \dot{\phi}_{roll}]^T$ is the state vector at time step k, indicating the orientation and angular velocity of the robot body about the Y- and X-axis, respectively. Since there is no control superimposed on the upper body, the control input $u_k = 0$ at time step k. The matrix A in Eq.(4.5) is given by

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

where T is the time step.

A linear measurement model is also assumed to have the form,

$$y_k = Cx_k \tag{4.7}$$

where y_k is a vector of measurements. The structure of C depends on the type of sensor modalities and state description.

In the case of body posture estimate of the robot NAO,

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

So we have $y_k = x_k$.

It is possible to form a state estimator and implement a Kalman filter by combining this measurement model with the dynamics given by Eq.(4.5) and Eq.(4.6) and assuming noisy process and measurement models for the system

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w \\ y_k = Cx_k + v \end{cases}$$

where w and v are vectors representing the standard process and measurement noise variables respectively which are assumed to be Gaussian processes with covariances given by Q and R,

$$w \sim N(0, Q)$$
$$v \sim N(0, R)$$

Kalman Filter Algorithm

Kalman filter is often conceptualized as two distinct phases: predict and update. The predict phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current time step, it does not include observation information from the current time step. In the update phase, the current a priori prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the a posteriori state estimate. The five core regression formulae of Kalman filter algorithm are as follows:

Predict

Predicted (a priori) state estimate $\hat{x}_{k|k-1} = A_k\hat{x}_{k-1|k-1} + B_k u_{k-1}$

Predicted (a priori) estimate covariance $P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$

Update

Optimal Kalman gain $K_k = P_{k|k-1}C_k^T(C_k P_{k|k-1} C_k^T + R_k)^{-1}$

Updated (a posteriori) state estimate $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1})$

Updated (a posteriori) estimate covariance $P_{k|k} = (I - K_k C_k)P_{k|k-1}$

As presented above in the last section, A and C are constant matrices and the control input $u_k = 0$. We assume that Q and R are also constant matrices, so the above formulae can be simplified as

1. $\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1}$
2. $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$
3. $K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}$
4. $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1})$
5. $P_{k|k} = (I - K_k C)P_{k|k-1}$

The starting state of the upper body is initialized $\hat{x}_{0|0} = [0 \quad 0 \quad 0 \quad 0]^T$ and

$$P_{0|0} = \begin{bmatrix} L & 0 & 0 & 0 \\ 0 & L & 0 & 0 \\ 0 & 0 & L & 0 \\ 0 & 0 & 0 & L \end{bmatrix}$$

where L is a number. Before the robot walks, the posture of the robot will first change from squatting to standing. The initial standing posture of the robot will be slightly different each time due to the difference in zero positions of the joints. Therefore, L should be a suitably large number. Here L is determined to be $L = 1000$.

Covariance matrices Q and R

The covariance matrices Q and R need to be known for implementing a Kalman filter. However, it is difficult to obtain the exact covariance matrices. Yet covariance matrices can be estimated, and to what extend the estimation of the covariance matrices match the real system affects the quality of the state estimation of a Kalman filter. Therefore, getting good estimates of the noise covariance matrices Q and R is important in practical implementation of the Kalman filter.

To determine the estimates of the covariance matrices Q and R, and achieve better performance of the Kalman filter, walking experiments with the same set of walking parameters were carried out on the real robot NAO. Table 4.2 shows the walking pattern parameters. The state estimator took the data from the IMU as inputs. The performance of various state estimators in the presence of different covariance matrices Q and R is compared, as shown in Figure 4.1 and Figure 4.2. When the matrix R is fixed, increasing the values of the terms in the matrix Q will decrease the estimate errors of the upper body orientation. On the contrary, the estimate errors of the upper body orientation will reduce as the values of the terms in R are lowered when Q is fixed. Eventually, the noise covariance matrices Q and R are determined as $Q = \text{diag}[10^{-1} \quad 10^{-1} \quad 10^{-5} \quad 10^{-5}]$ and $R = \text{diag}[10^{-4} \quad 10^{-4} \quad 10^{-3} \quad 10^{-3}]$ for estimating the orientation and angular velocity of the upper body of the robot.

Table 4.2: Parameters used in the walking experiments for determination of the covariance matrices Q and R.

Symbol	Description	Value
T_{stride}	Walking period	0.50 s
τ_{dsp}	Double support ratio	0.01
H_{foot}	Maximum elevation of foot	0.032 m
z_{min}	Lowest level of the waist	0.321 m
z_{max}	Peak height of the waist	0.323 m
A_l	Lateral swing amplitude of the waist	0.005 m
$\bar{v}$	Average walking speed	0.169 m/s
L_{step}	Average step length	0.055 m

4.3.3 Pitch/Roll Controller

Two strategies are described here:

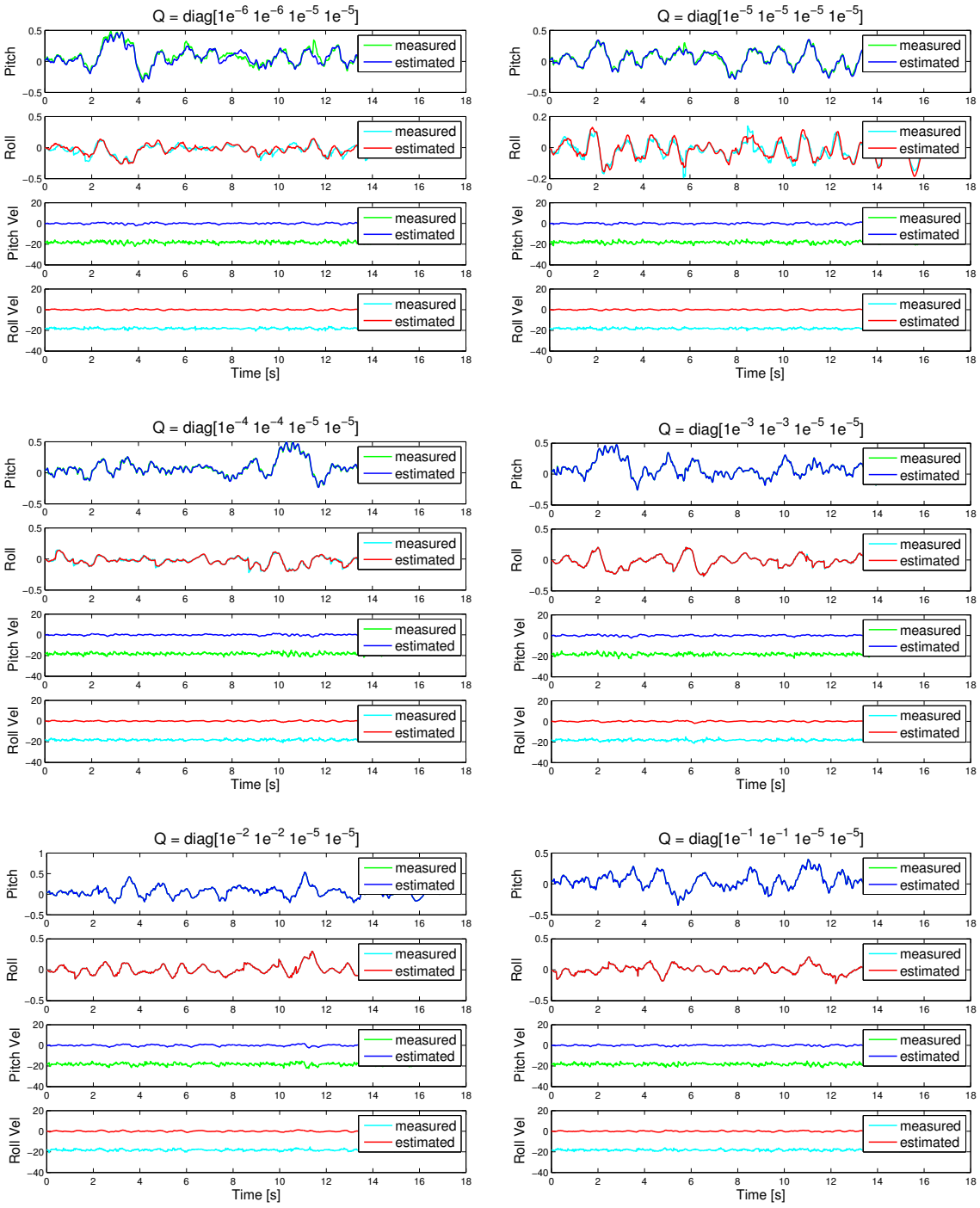


Figure 4.1: Various state estimator performances in the presence of different covariance matrices Q when the covariance matrix R is fixed, $R = \text{diag}[10^{-3} \quad 10^{-3} \quad 10^{-3} \quad 10^{-3}]$.

- Hip Strategy: hip joints are used to control the body posture.
- Ankle Strategy: ankle joints are utilized to control the body posture.

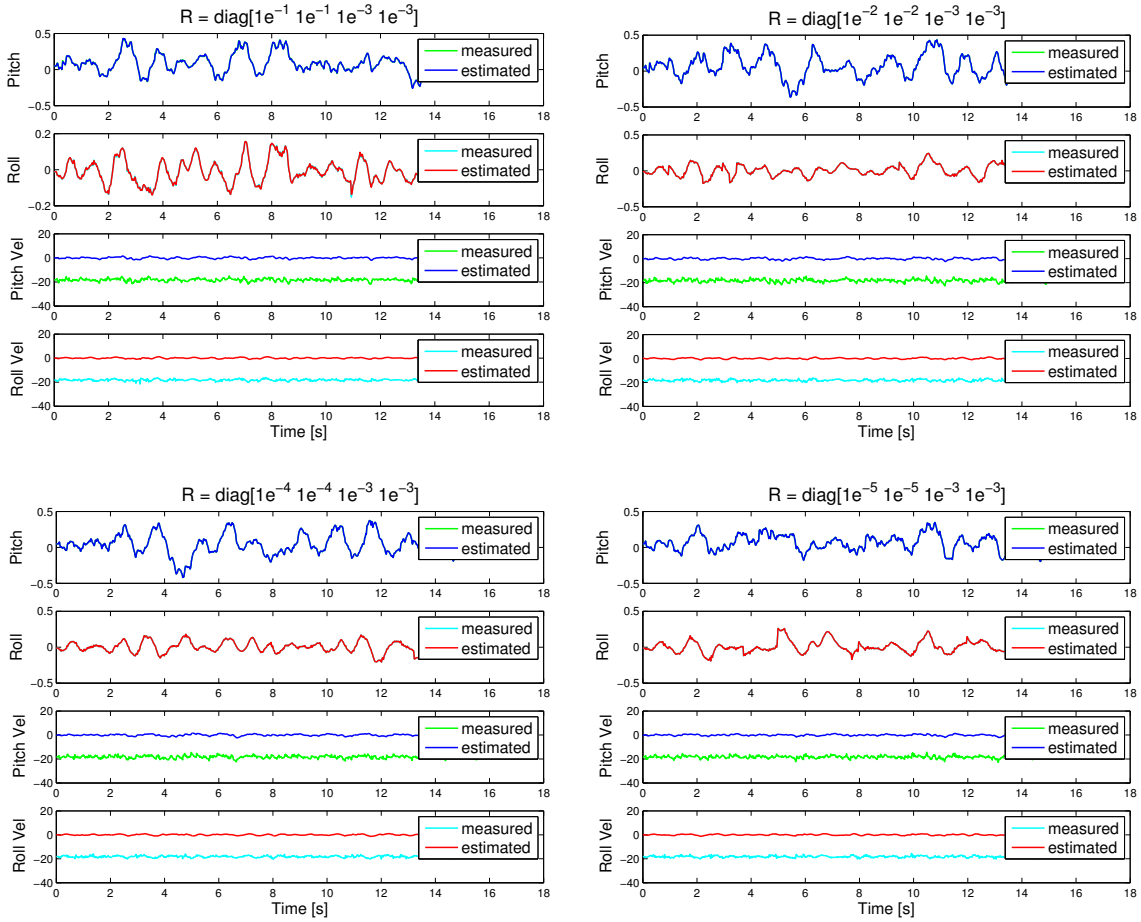


Figure 4.2: Various state estimator performances in the presence of different covariance matrices R when the covariance matrix Q is fixed, $Q = \text{diag}[10^{-1} \quad 10^{-1} \quad 10^{-5} \quad 10^{-5}]$.

4.3.3.1 Hip Strategy

We first consider utilizing hip joints to control the body posture of the robot when the walking floor is flat (see Figure 4.3(a)). The other reason is that the hip joints are near the body, therefore the most effective way to keep the desired body posture is to control the hip joints.

We introduce a body posture controller to correct the inclination of the torso of the robot. The correct angles ($\Delta\theta^{hip}$, $\Delta\phi^{hip}$) are expressed in a PD (Proportional-Derivative) manner and then added to the reference trajectories of the hip joints.

$$\Delta\theta^{hip} = K_{p1}(\theta^{torso}_{des} - \theta^{torso}_{m}) + K_{d1}(\dot{\theta}^{torso}_{des} - \dot{\theta}^{torso}_{m}) \tag{4.8}$$

$$\Delta\phi^{hip} = K_{p2}(\phi^{torso}_{des} - \phi^{torso}_{m}) + K_{d2}(\dot{\phi}^{torso}_{des} - \dot{\phi}^{torso}_{m}) \tag{4.9}$$

$$\vartheta^{hip}_{mod} = \vartheta^{hip}_{ref} + \Delta\theta^{hip} \tag{4.10}$$

$$\varphi^{hip}_{mod} = \varphi^{hip}_{ref} + \Delta\phi^{hip} \tag{4.11}$$

where K_p and K_d are the proportional and derivative gains of PD controller; (θ^{torso}_{des}, ϕ^{torso}_{des}) are the desired torso inclination about the Y- and X- axis, ($\dot{\theta}^{torso}_{des}$, $\dot{\phi}^{torso}_{des}$) stand for the desired

angular velocity of the torso about the Y- and X- axis; (θ_m^{torso}, ϕ_m^{torso}) and ($\dot{\theta}_m^{torso}$, $\dot{\phi}_m^{torso}$) represent the measured orientation and angular velocity of the torso, and they are estimated by a Kalman filter; (ϑ_{ref}^{hip}, φ_{ref}^{hip}) denote the reference trajectories of the hip/ankle joints about the Y-axis and X-axis; (ϑ_{mod}^{hip}, φ_{mod}^{hip}) are the modified angles sent to the robot servomotor. This control method is performed on both left and right hip joints during the walking process.

4.3.3.2 Ankle Strategy

Next let us consider the inclination of the torso is corrected by using ankle joints (refer to Figure 4.3(b)). The correct angles ($\Delta\theta^{ankle}$, $\Delta\phi^{ankle}$) are expressed in a PD manner and then added to the reference trajectories of the ankle joints. The meaning of the symbols here is the same as that used in the hip strategy.

$$\Delta\theta^{ankle} = K_{p1}(\theta_{des}^{torso} - \theta_m^{torso}) + K_{d1}(\dot{\theta}_{des}^{torso} - \dot{\theta}_m^{torso}) \tag{4.12}$$

$$\Delta\phi^{ankle} = K_{p2}(\phi_{des}^{torso} - \phi_m^{torso}) + K_{d2}(\dot{\phi}_{des}^{torso} - \dot{\phi}_m^{torso}) \tag{4.13}$$

$$\vartheta_{mod}^{ankle} = \vartheta_{ref}^{ankle} + \Delta\theta^{ankle} \tag{4.14}$$

$$\varphi_{mod}^{ankle} = \varphi_{ref}^{ankle} + \Delta\phi^{ankle} \tag{4.15}$$

Then (θ_{mod}^{ankle}, ϕ_{mod}^{ankle}) are the modified ankle angles and sent to the robot servomotor. Ankle strategy is also executed on both ankle joints during walking.

4.3.4 Experimental Results

Dynamic walking experiments were performed to verify the presented walking pattern generation method and body posture control method using the humanoid robot NAO.

4.3.4.1 Experimental System Description

Figure 4.4 shows the leg configuration of NAO. The proposed walking system is implemented as shown in Figure 4.5. Reference joint trajectories are calculated by inverse kinematics given the relative positions of both feet with respect to the waist (indicated by $\tilde{p}_{left}$ and $\tilde{p}_{right}$) which are planned by the walking pattern generator. During walking, the legs of the robot are controlled by local PID controllers to follow the reference trajectory. The online torso pitch/roll controller modifies the reference trajectories in real time based on the sensor feedback in order to keep the upper body posture of the robot constant and thus to stabilize the robot. A Kalman filter is implemented to accurately estimate the state of the upper body because the measured data from IMU contains noises.

4.3.4.2 Dynamic Bipedal Walking

The parameters for the standard forward walking are given in Table 4.3. The lateral swing amplitude of the waist is preset to a small value, e.g. 1 mm, since the interchange of the SSP and DSP arouses the natural oscillation of the upper body of the robot in the frontal plane. Experiments also showed that with a shorter walking period the walking performance of the robot went worse as the robot moved laterally with a bigger amplitude because the inclination of the upper body about the X-axis was larger. As a result, the walking was less stable or even unstable. Note that in normal human walking the lateral swing amplitude tends

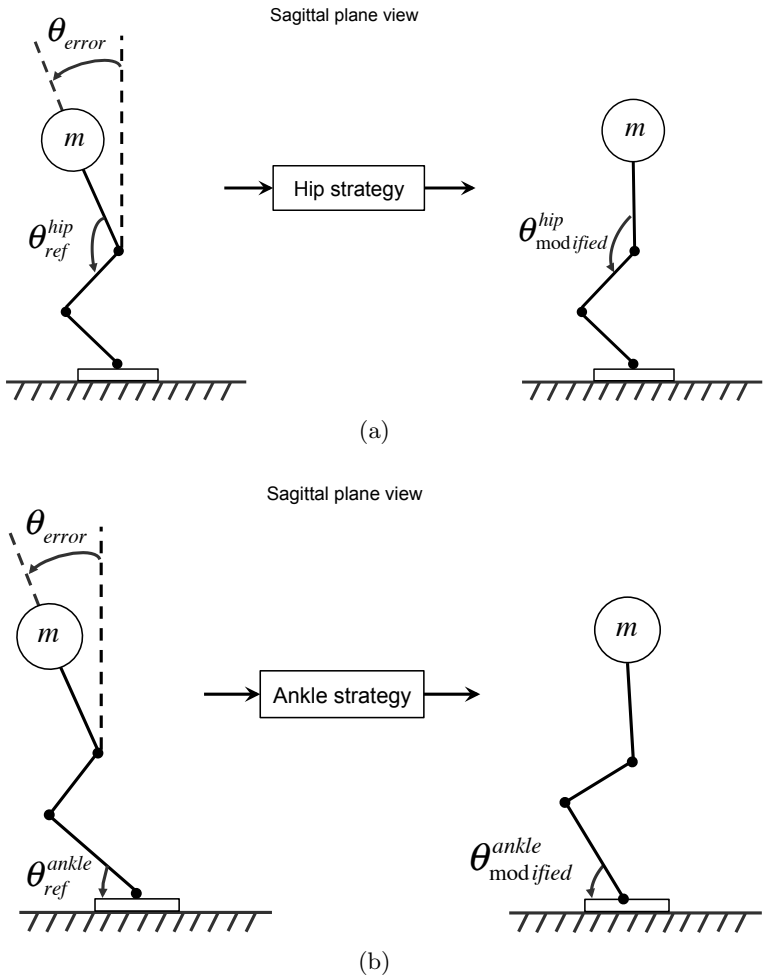


Figure 4.3: Analysis of (a) hip strategy (b) ankle strategy in sagittal plane view.

to decrease as the walking speed increases [99]. Therefore, when the robot walks faster, this value is reduced to 0.5 mm. The sagittal oscillation amplitude is 2 mm. While standing still, the distance between the centers of two feet is 100 mm. So d_0 is preset to 50 mm during walking. The walking period is set to 0.5 s in order to overcome the landing impact force. We set the double support ratio in a walking cycle to be 1% because the humanoid robot NAO has neither toe joints nor heel joints. The step height during walking is 26 mm.

Hip strategy and ankle strategy were verified, respectively, through experiments. In all experiments, the robot walked 61 steps in 15 seconds on a flat and noninclined floor with local unevenness. Several sets of experiments were taken on the same robot, but under different walking speed, step length and waist height.

1) Walking Experiments Without Posture Control: Figure 3.21 illustrates the inclination of the upper body of the robot in the course of a walk which is the output of the walking pattern generator without considering torso pitch/roll control. The maximum value of the pitch angles about the Y-axis is more than 1.0 rad. It is sufficient to make the robot fall down, so the walking was unstable.

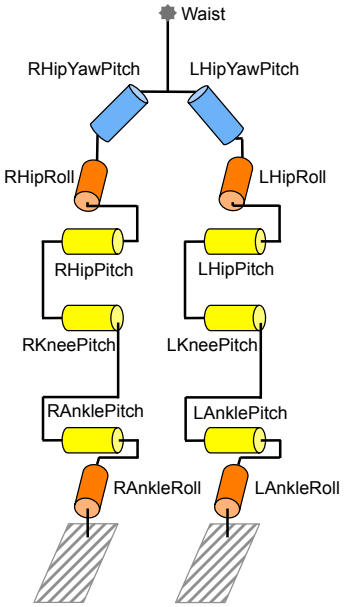


Figure 4.4: Leg configuration of the humanoid robot NAO.

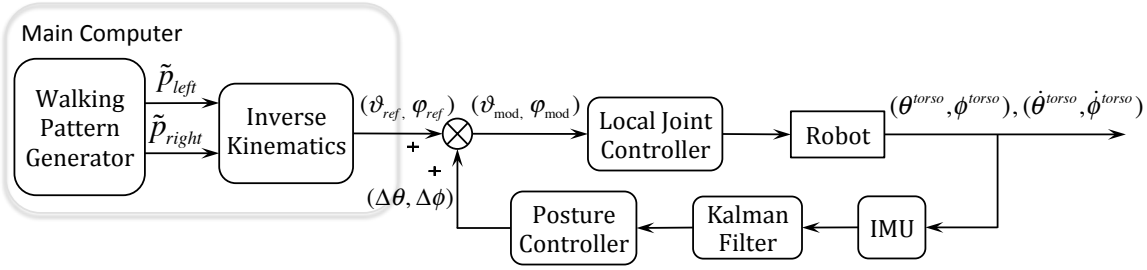


Figure 4.5: Torso pitch/roll control of NAO.

2) Walking Experiments With Hip Strategy: In the second experiment, the body posture of the robot was adjusted in real time by using hip joints. Three sets of experiments were carried out.

First, the validity of the control method was proved as the robot took a stable walking[1] at a lower speed. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.327$ m, $z_{max} = 0.329$ m. The robot walked 1.2912 m in 15 seconds[2]. Therefore, the average walking speed was $\bar{v} = 0.0861$ m/s with the average step length of $L_{step} = 0.0212$ m. The gains used for the torso pitch/roll controller were $K_p = 0.35$, $K_d = -0.05$[3]. Figure 4.6 demonstrates the walking performances. Figure 4.6(a) shows the reference and modified angles of the hip joints of both legs. The measured inclination of the upper body of the robot during walking is plotted in

[1]Literature [17] gives an understanding of stability, that a locomotion mode is understood to be stable if it is sustainable without a fall, and if it allows a safe return to a statically stable configuration.

[2]The walking distance is calculated by implementing the forward kinematics between the two legs in real time. The forward kinematics of the robot is given in Appendix B.

[3]For both hip strategy and ankle strategy, the used proportional gain K_p and derivative gain K_d of the PD controller were hand-tuned via a large number of experiments and proved to be the best values.

Table 4.3: Parameters of the forward walking pattern.

Symbol	Description	Value
A_l	Lateral swing amplitude of waist	1 (mm)
A_s	Sagittal oscillation amplitude of waist	2 (mm)
d_0	Foot collision avoidance distance	50 (mm)
T_{stride}	Walking period (stride time)	0.5 (s)
T_{step}	Step time	0.25 (s)
τ_{dsp}	Double support ratio	0.01 (1%)
T_{ssp}	Single-support time	$T_{step} \times (1 - \tau_{dsp})$
T_{dsp}	Double-support time	$T_{step} \times \tau_{dsp}$
H_{foot}	Maximum elevation of foot	26 (mm)

Figure 4.6(b). Figure 4.6(c) illustrates the reference and actual velocities of the waist in the forward direction from the start to the end of the walk. Measured ZMP trajectories in the global coordinate system are shown in Figure 4.6(d).

Then the hip strategy was verified at a medium walking speed. In the experiment displayed by Figure 4.7, the walking parameters were $A_l = 0.001$ m, $z_{min} = 0.323$ m, $z_{max} = 0.325$ m. The walking distance of the robot was 2.8794 m. So the average walking speed was $\bar{v} = 0.1920$ m/s and the average step length was $L_{step} = 0.0472$ m. The parameters used for the torso pitch/roll were $K_p = 0.40$ and $K_d = -0.05$, respectively.

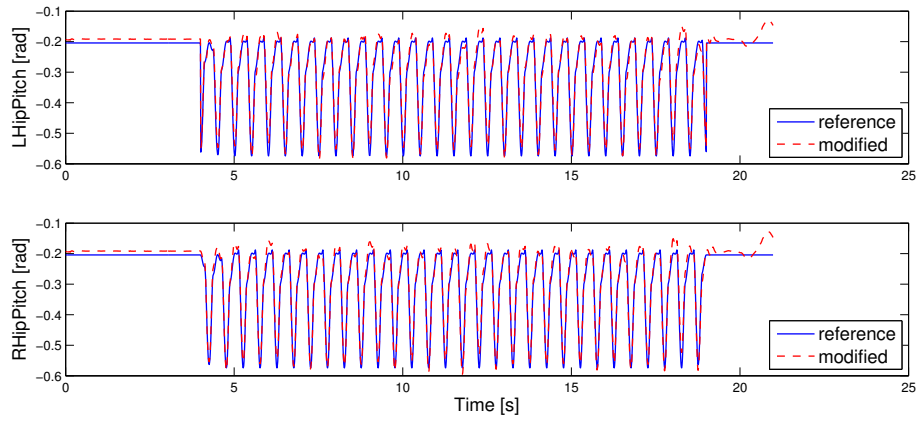
Figure 4.8 demonstrates walking performances of a stable walking when the robot walked at a higher speed. The walking parameters were $A_l = 0.0005$ m, $z_{min} = 0.313$ m, $z_{max} = 0.315$ m. The robot travelled 3.5469 m in 15 seconds. The average walking speed was $\bar{v} = 0.2365$ m/s and the average step length was $L_{step} = 0.0581$ m. The proportional and derivative gains K_p and K_d were 0.4 and -0.05, respectively. Due to the robot inertia, the stop of the walking causes a summit in the inclination about the Y-axis at the end of the walking, as shown in Figure 4.8(b). In Figure 4.8(c), the velocity of the robot is gradually increased step by step from being stationary at the beginning of walking since the sudden start of walking with a high speed can make the robot unstable. This process mimicks the development phase in human walking.

Before the robot walks, the posture of the robot will first change from squatting to standing. To make sure that the robot starts walking from being stationary, the robot will keep the standing posture for 4 seconds. So the data was recorded from the end of the fourth second. As the walking speed increased, the waist height was decreased to ensure the dynamic stability of the robot.

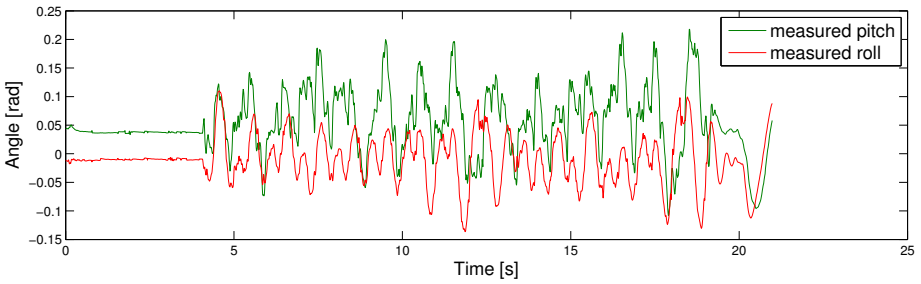
Experiments have indicated that as the proportional gain K_p rises, the upper body of the robot will lean backwards. Leaning slightly backwards, to some extend, is beneficial to higher speed walking. However, the upper body of the robot appears to oscillate when K_p is greater than 0.05, and consequently the walking becomes unstable. Besides, the derivative gain is a negative value. When it reduces to or less than -0.06 the robot body will shake.

The experiments have shown that the dynamic stable walking can be achieved through the proposed walking system with the hip strategy. Figure 4.9 shows the snapshots of the walking experiment using hip strategy with the robot NAO.

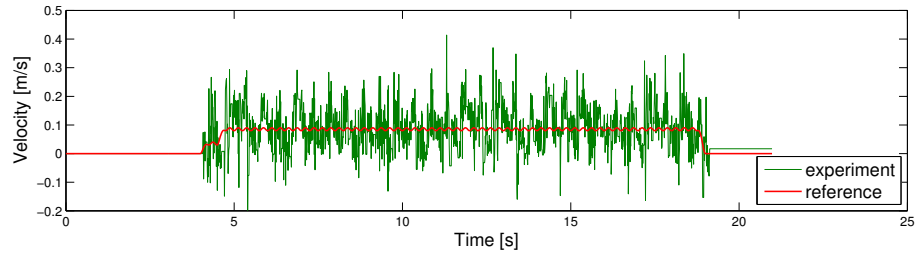
3) Walking Experiments With Ankle Strategy: In the third experiment, the ankle strategy



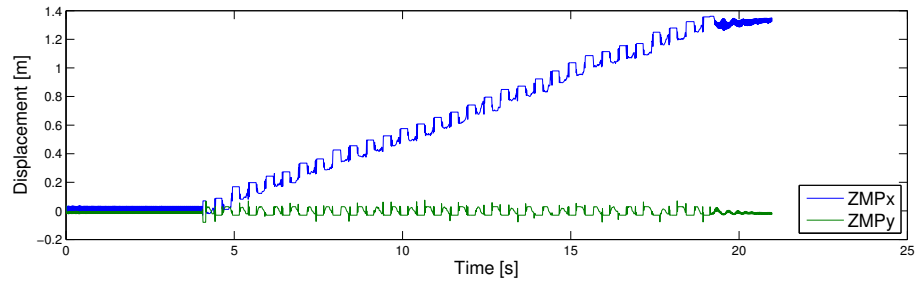
(a) reference and modified angles of hip joints.



(b) measured upper body inclination.

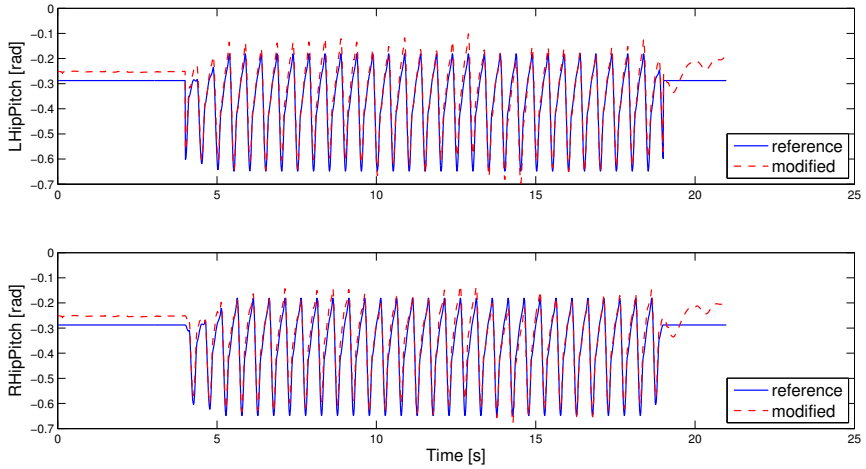


(c) Waist velocity.

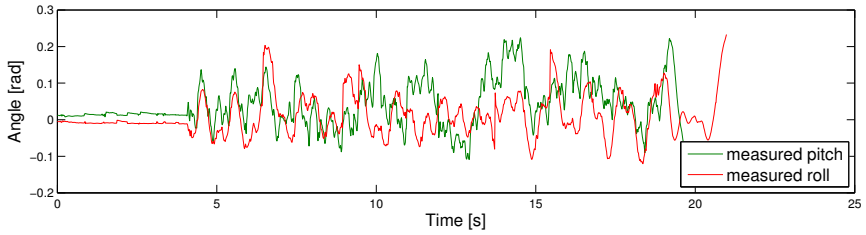


(d) ZMP trajectories.

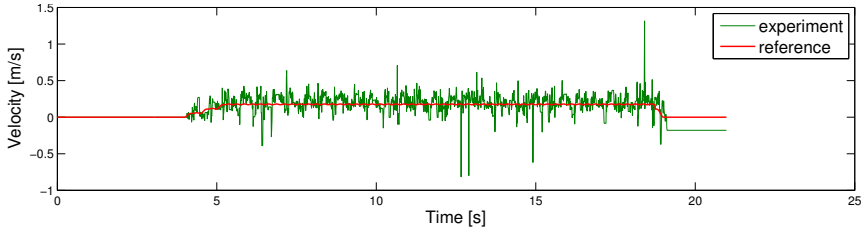
Figure 4.6: Walking performance using the hip strategy at a lower speed. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.327$ m, $z_{max} = 0.329$ m. The robot travelled 1.2912 m in 15 seconds. The parameters for the PD controller were $K_p = 0.35$, $K_d = -0.05$.



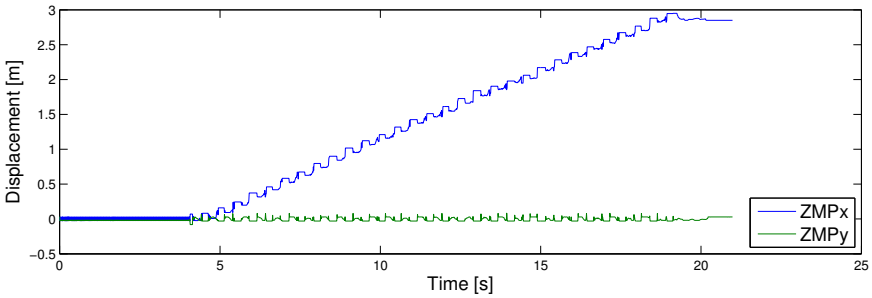
(a) reference and modified angles of hip joints.



(b) measured upper body inclination.

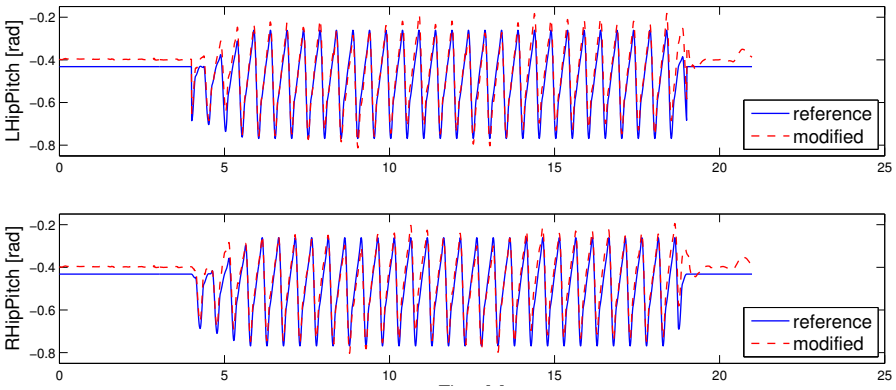


(c) Waist velocity.

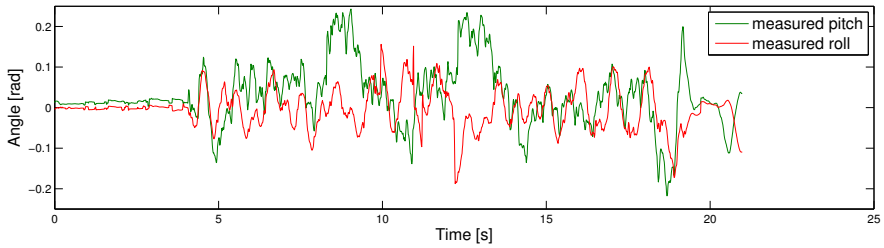


(d) ZMP trajectories.

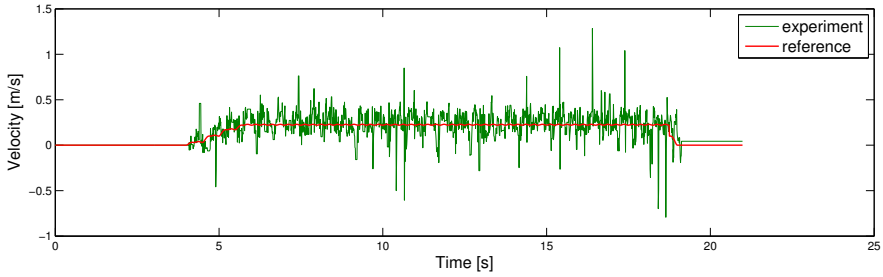
Figure 4.7: Walking performance using the hip strategy at a medium speed. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.323$ m, $z_{max} = 0.325$ m. The walking distance was 2.8794 m in 15 seconds. The parameters for the PD controller were $K_p = 0.40$, $K_d = -0.05$.



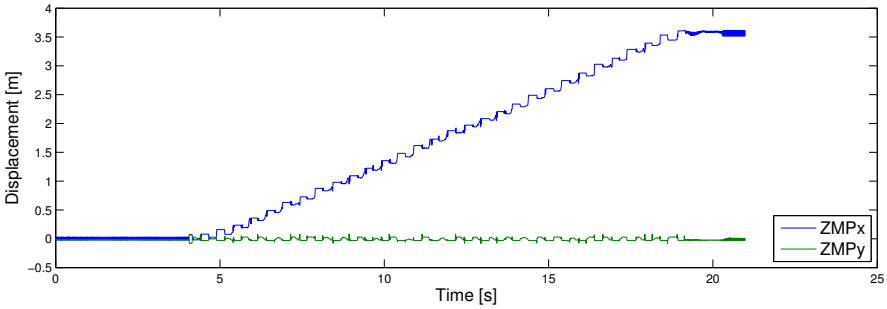
(a) reference and modified angle of hip joints.



(b) measured upper body inclination.

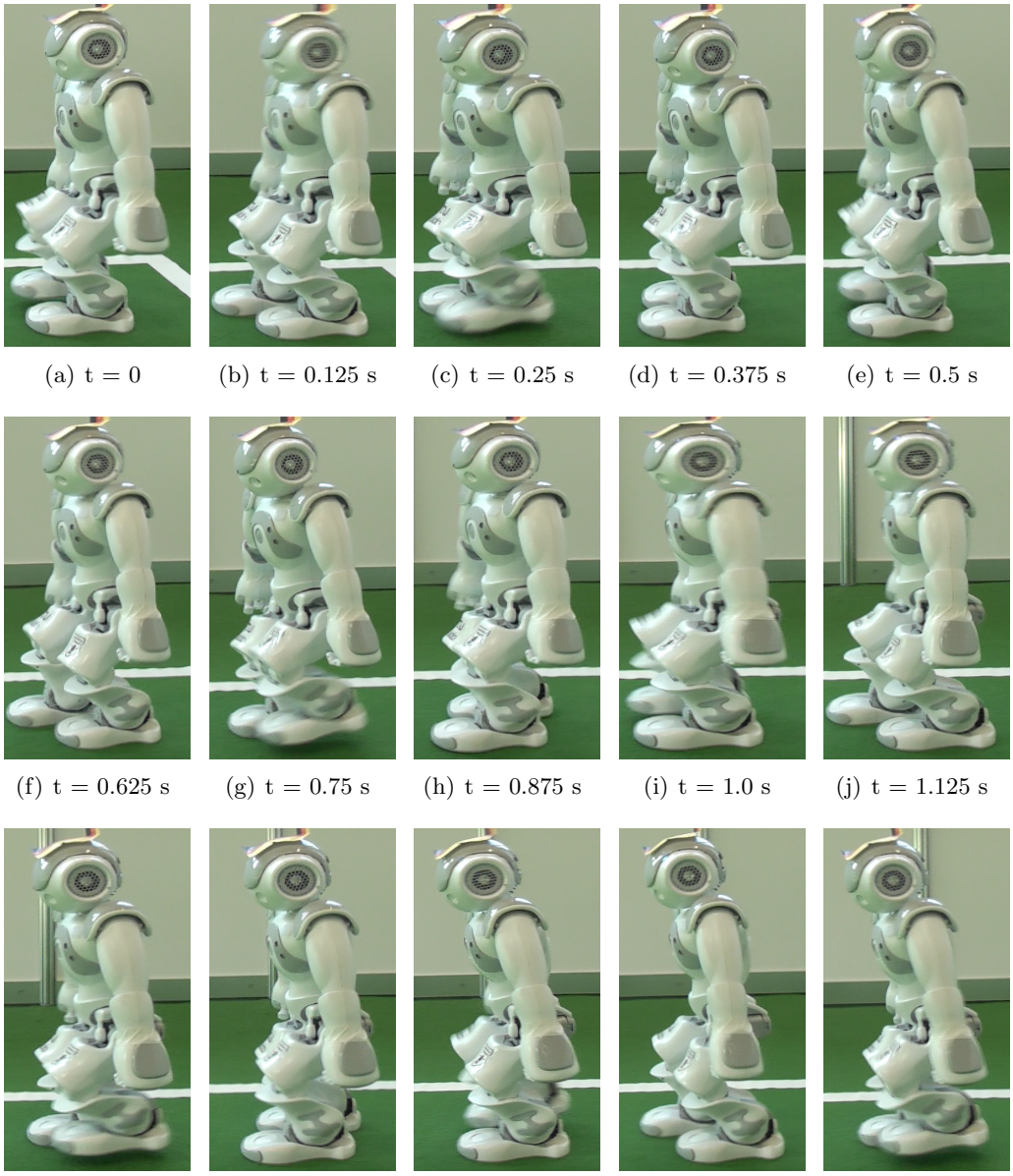


(c) Waist velocity.



(d) ZMP trajectories.

Figure 4.8: Walking performance using the hip strategy at a higher speed. The walking parameters were $A_l = 0.0005$ m, $z_{min} = 0.313$ m, $z_{max} = 0.315$ m. The robot walked 3.5469 m in 15 seconds. The parameters for the PD controller were $K_p = 0.40$, $K_d = -0.05$.



(a) t = 0 (b) t = 0.125 s (c) t = 0.25 s (d) t = 0.375 s (e) t = 0.5 s

(f) t = 0.625 s (g) t = 0.75 s (h) t = 0.875 s (i) t = 1.0 s (j) t = 1.125 s

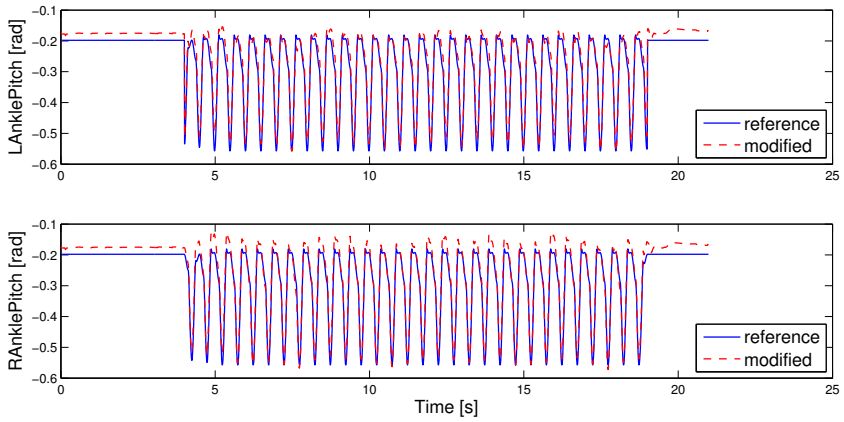
(k) t = 1.25 s (l) t = 1.375 s (m) t = 1.5 s (n) t = 1.625 s (o) t = 1.75 s

Figure 4.9: Snapshots of forward walking using the hip strategy.

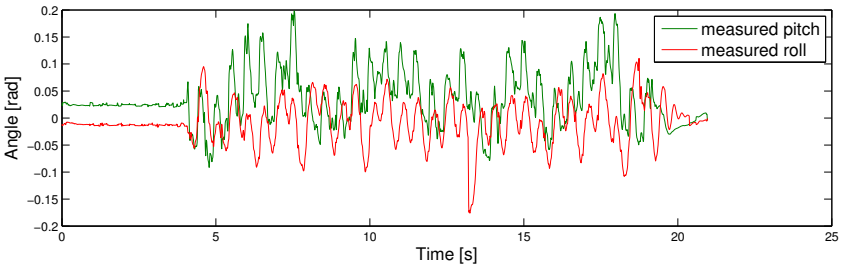
was employed to keep the body posture of the robot upright in real time. Three sets of experiments were taken.

Similarly, we first let the robot walk at a low speed and experimental results are demonstrated in Figure 4.10. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.327$ m, $z_{max} = 0.329$ m. The robot walked 1.1654 m in 15 seconds. The average walking speed was $\bar{v} = 0.0777$ m/s and the average step length was $L_{step} = 0.0191$ m. The parameters of the torso pitch/roll controller were $K_p = 0.30$, $K_d = -0.05$.

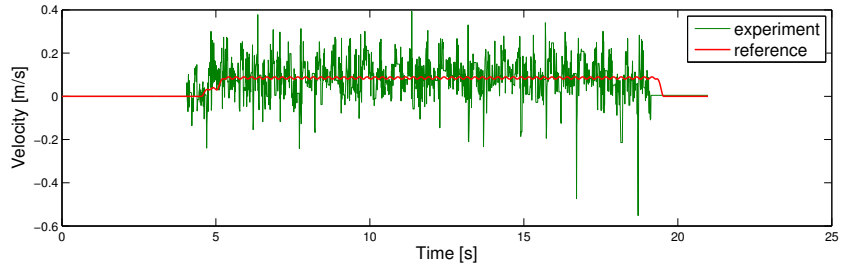
Then the robot accomplished stable walking at a medium speed, as shown in Figure 4.11. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.323$ m, $z_{max} = 0.325$ m. The walking



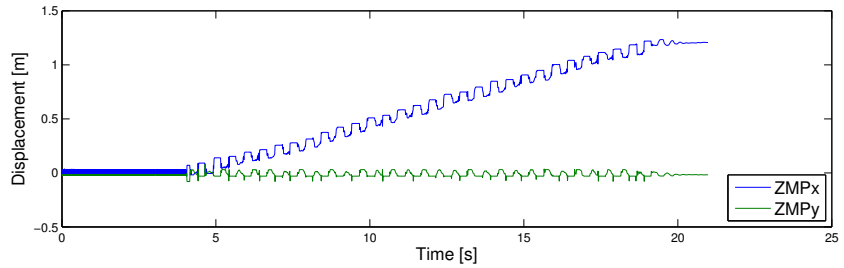
(a) reference and modified angle of ankle joints.



(b) measured upper body inclination.

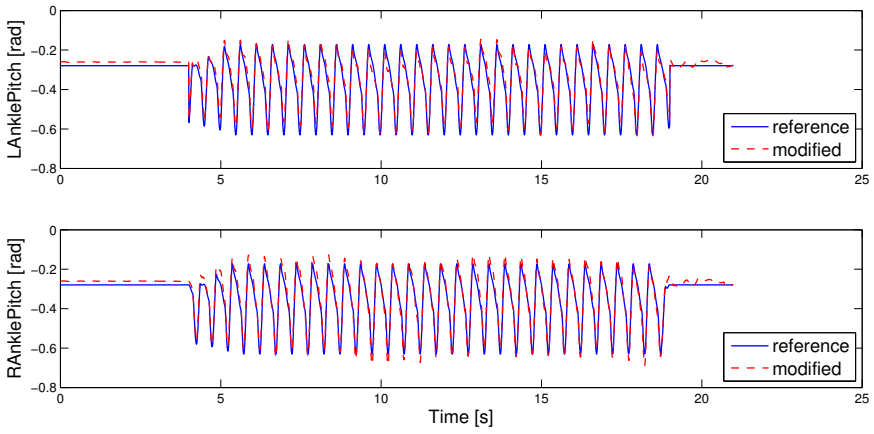


(c) Waist velocity.

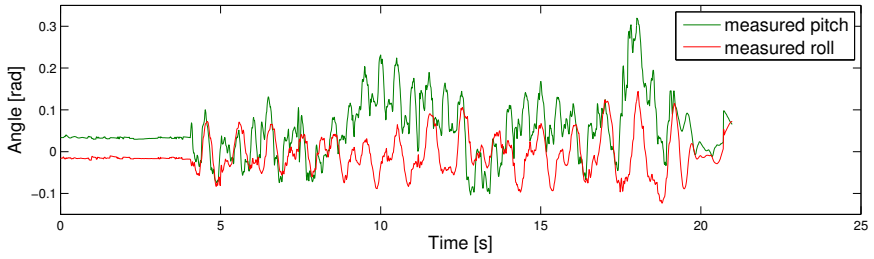


(d) ZMP trajectories.

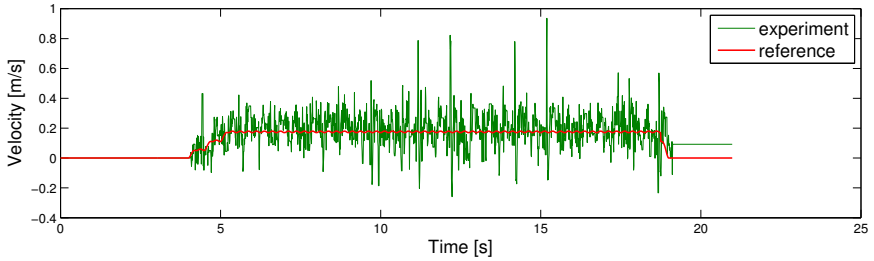
Figure 4.10: Walking performance using the ankle strategy at a lower speed. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.327$ m, $z_{max} = 0.329$ m. The robot walked 1.1654 m in 15 seconds. The parameters for the PD controller were $K_p = 0.30$, $K_d = -0.05$.



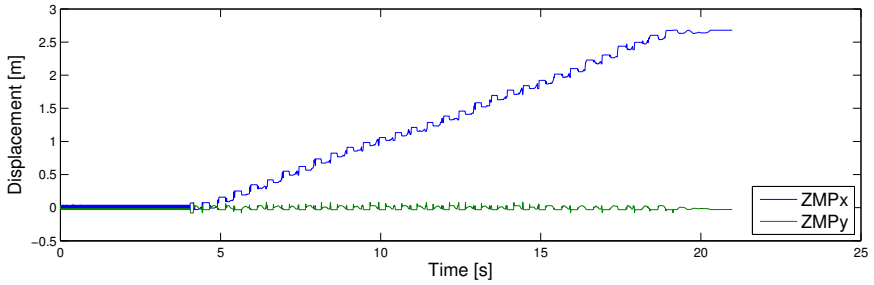
(a) reference and modified angle of ankle joints.



(b) measured upper body inclination.



(c) Waist velocity.



(d) ZMP trajectories.

Figure 4.11: Walking performance using the ankle strategy at a medium speed. The walking parameters were $A_l = 0.001$ m, $z_{min} = 0.323$ m, $z_{max} = 0.325$ m. The robot travelled 2.6119 m in 15 seconds. The parameters for the PD controller were $K_p = 0.28$, $K_d = -0.06$.

distance was 2.6119 m in 15 seconds. Therefore, the average walking speed was $\bar{v} = 0.1741$ m/s with the average step length $L_{step} = 0.0428$ m. The parameters of the pitch/roll controller were $K_p = 0.28$, $K_d = -0.06$.

The validity of the ankle strategy was also confirmed through walking experiments at a higher speed. The walking parameters were $A_l = 0.0005$ m, $z_{min} = 0.313$ m, $z_{max} = 0.315$ m. Figure 4.12 demonstrates the walking performances of the stable walking using the ankle strategy. Figure 4.12(a) shows the reference and modified angles of the ankle joints. The measured inclination of the upper body during walking is plotted in Figure 4.12(b). Due to the robot inertia, a summit in the inclination about the Y-axis is caused by the stop of the walking at the end. Figure 4.12(c) illustrates the reference and actual velocities of the waist in the forward direction from the start to the end of the walk, and the ZMP trajectories in the global coordinate system are shown in Figure 4.12(d). In the experiment, the robot travelled 3.5035 m in 15 seconds. The average walking speed was $\bar{v} = 0.2336$ m/s and the average step length was $L_{step} = 0.0574$ m. The proportional and derivative gains K_p and K_d used for the torso pitch/roll controller were 0.28 and -0.06, respectively.

Since the ankle joints are farther away from the torso of the robot than the hip joints, the small adjustment in the ankle joints will arouse larger correction in the attitude of the upper body. The value of proportional gain used in the ankle strategy is therefore smaller than that used in the hip strategy.

Experiments have shown the validity of controlling the upper body posture by means of the ankle strategy, although the ankle joints are farther away from the upper body than the hip joints and are normally considered not effective in controlling the body posture.

In conclusion, the presented walking pattern generation method and body posture control method using hip strategy and ankle strategy are valid and effective in stabilizing the robot.

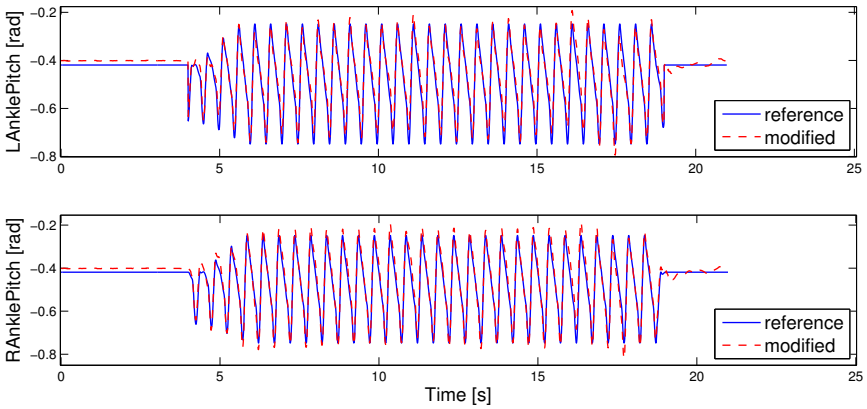
4.3.4.3 Energy Consumption Comparison

Chapter 3 presents a walking pattern generation method aiming at realizing human-like stretched-knee walking of a humanoid robot. The ability to walk with stretched knees is imperative for a humanoid robot to rehabilitate human motion. Besides, walking with stretched knees is also the requirement for an efficient bipedal gait. To elucidate the gait efficiency of the presented walking pattern in this thesis, we investigated the energy consumption of the actuators in the knee joint during the walking process of a robot that fulfills the presented walking pattern and conventional walking pattern respectively.

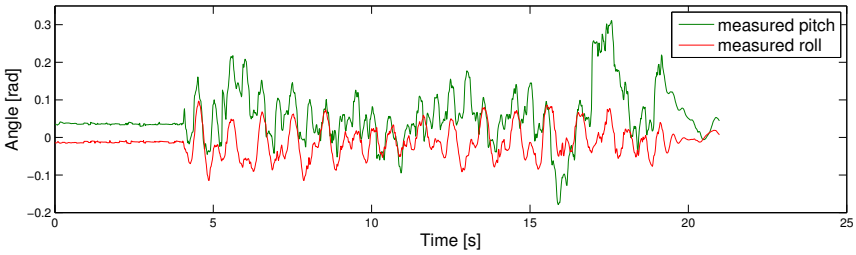
Experiments

To investigate the energy consumption of the knee joint actuators between the conventional bending-knee walking pattern [119] with the upper body motion constrained on a horizontal plane and the presented walking pattern in this thesis, experiments were carried out by using NAO with current sensors. The experiments were carried out on the same robot at four various walking speeds, approximately 0.05 m/s, 0.10 m/s, 0.15 m/s and 0.20 m/s, respectively. For each speed, the robot walked 20 seconds and the experiment was repeated 30 times. Then average walking speed and energy consumption were calculated.

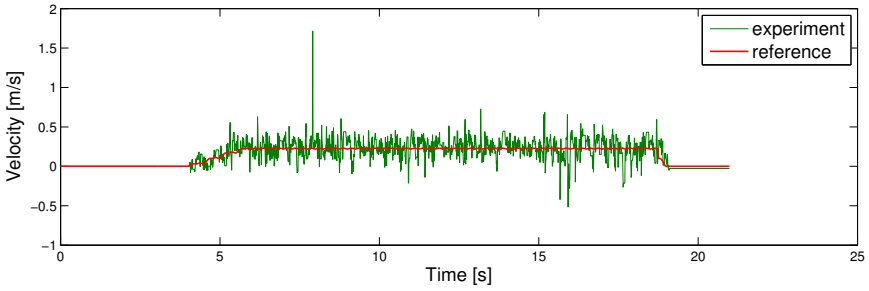
All the walks in the experiments were stable. The walking parameters of the presented and conventional walking patterns are shown in Table 4.4 and Table 4.5, respectively.



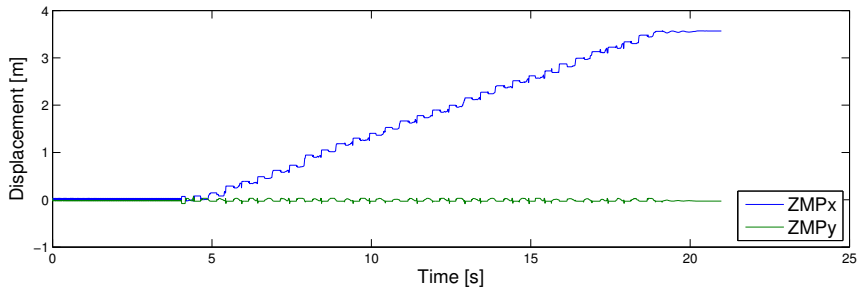
(a) reference and modified angle of ankle joints.



(b) measured upper body inclination.



(c) Waist velocity.



(d) ZMP trajectories.

Figure 4.12: Walking performance using the ankle strategy at a higher speed. The walking parameters were $A_l = 0.0005$ m, $z_{min} = 0.313$ m, $z_{max} = 0.315$ m. The robot walked 3.5035 m in 15 seconds. The parameters for the PD controller were $K_p = 0.28$, $K_d = -0.06$.

Table 4.4: Walking experiment parameters of the presented walking pattern.

Speed (cm/s)[4]	5.21±0.23	9.91±0.69	16.39±0.53	20.04±0.58
Duration (s)	20	20	20	20
Times	30	30	30	30
Step height (cm)	2.0	2.0	2.4	2.4
Waist height z_{min} (m)	0.331	0.331	0.326	0.321
Waist height z_{max} (m)	0.332	0.332	0.327	0.322

Table 4.5: Walking experiment parameters of the conventional walking pattern.

Speed (cm/s)[4]	5.17±0.17	10.29±0.53	15.28±0.72	20.20±0.95
Duration (s)	20	20	20	20
Times	30	30	30	30
Step height (cm)	2.0	2.0	2.0	2.0
Waist height z (m)	0.318	0.318	0.318	0.318

Calculation of the Average Energy Consumption

W_i is used to represent the electrical work in the circuit in the process of the i^{th} walk, then W_i can be expressed as

$$W_i = \sum_{j=1}^{2000}(U_j I_j \Delta t_j) = U(\sum_{j=1}^{2000} I_j)\Delta t \tag{4.16}$$

with U denoting the voltage in the armature circuit, I being the current in the armature circuit, and Δt denoting the sampling interval. In the calculation, the nominal voltage $U = 21.6\ V$; the current data is provided by the in-built current sensors; the sampling interval is $\Delta t = 10$ ms. Since the time for each walk is 20 seconds, there are in all 2000 samples.

Then the mean value of the energy consumption after n times of walking is

$$\overline{W} = \frac{W_1 + W_2 + \cdots + W_n}{n} \tag{4.17}$$

with $n = 30$ in the experiment.

Experimental Results

Figure 4.13 shows an example of the measured current of the knee joint actuators from the start to the end of a walk. Energy consumption of the two kinds of walking patterns is illustrated in Figure 4.14 and Figure 4.15. In both figures, the minimum, the maximum and the average values of the energy consumption of the left and right knees under four different

[4]To show data distribution, we use 'mean ± standard deviation' to express the value of each speed. Standard deviation formula: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^2}$ with σ denoting standard deviation, x_i denoting each value of dataset, $\bar{x}$ being the arithmetic mean of the data, and N the total number of data points.

walking speeds are indicated. In the calculation, the nominal voltage in the armature circuit is 21.6 V.

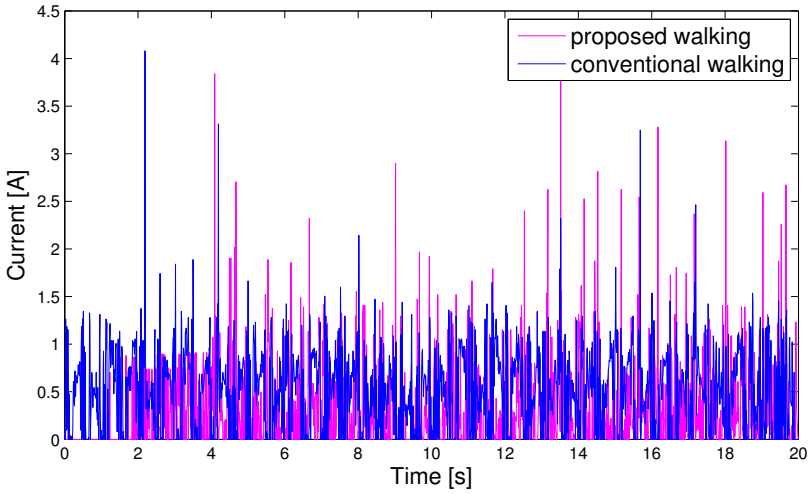


Figure 4.13: Measurement of knee joint current while the robot walks at 0.10 m/s.

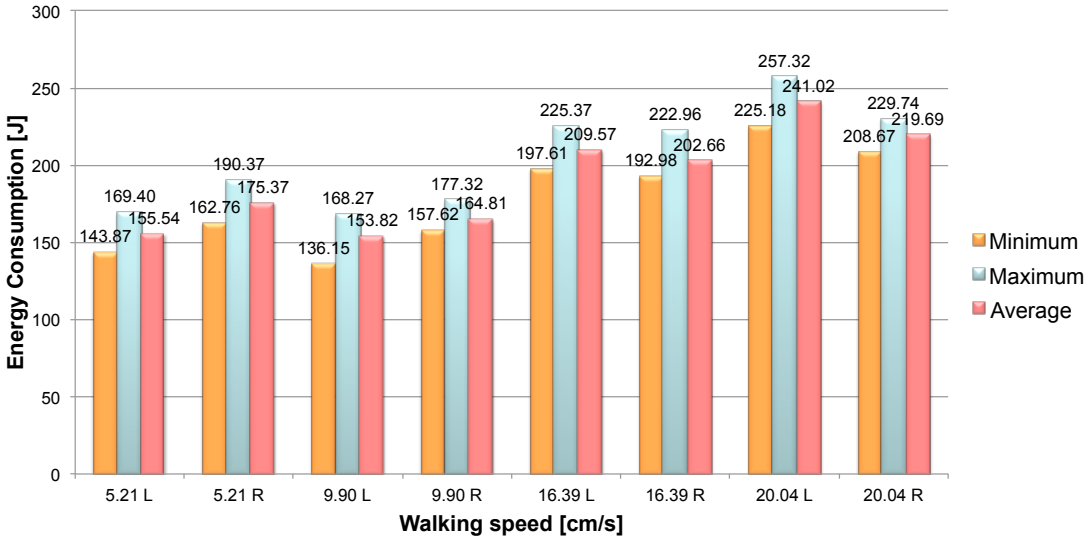


Figure 4.14: Energy consumption of knee joints with the presented walking pattern at different walking speeds.

Analysis

It is found that when the robot walks at a lower speed, e.g. 0.05 m/s and 0.10 m/s, the presented walking pattern outperforms the conventional walking pattern saving 15% and 20% energy, respectively. However, when the walking speed is higher, e.g. 0.15 m/s and 0.20 m/s, the knee actuators with the conventional walking pattern consumed less energy.

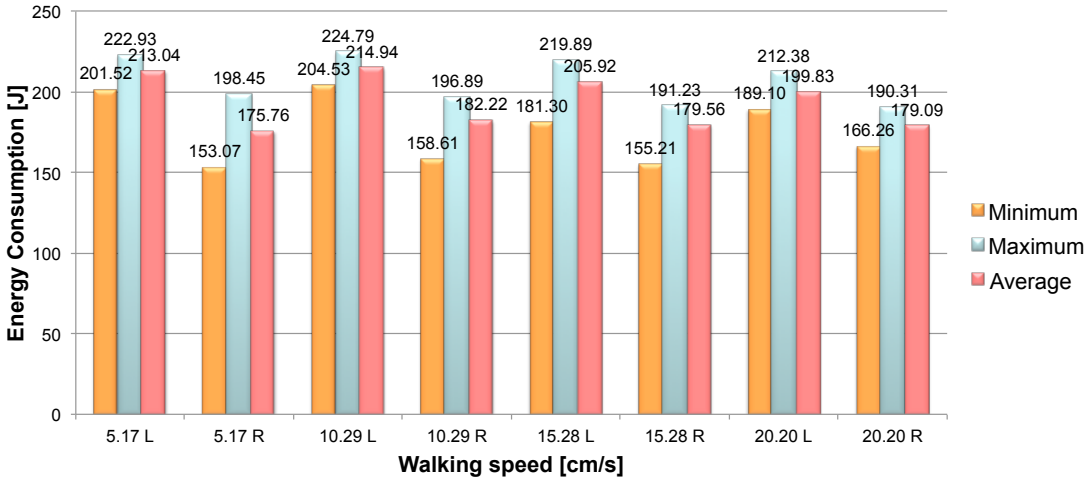


Figure 4.15: Energy consumption of knee joints with the conventional walking pattern at different walking speeds.

Instead of restricting the upper body motion to a flat surface, the presented walking pattern involves an upward and downward upper body motion. The vertical motion of the upper body results in an inertial force in the vertical direction which poses a challenge to the walking stability of the robot. Especially at a higher walking speed, the robot fulfilling the presented walking pattern needs to lower its waist height in order to ensure the dynamic stability. Moreover, to avoid walking backwards the robot has to increase the elevation of the step height. As a result, the robot walks with more bent knees, and more energy is consumed at the knee joints.

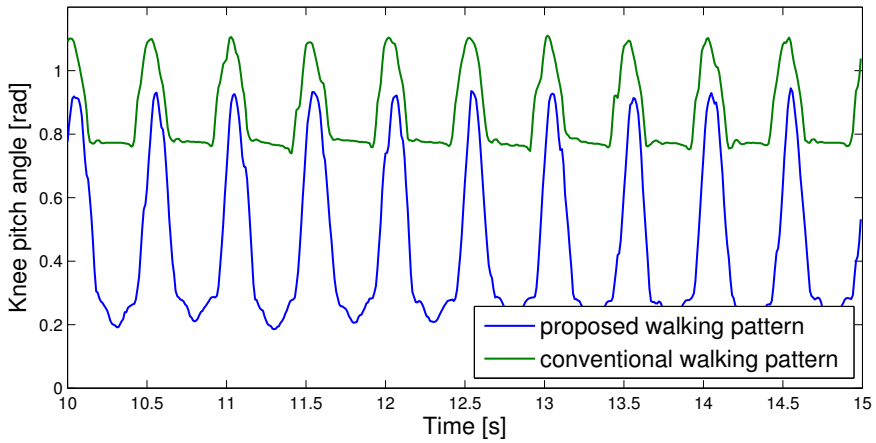
When the robot walks at a lower speed, the presented walking pattern clearly outperforms the conventional walking pattern in energy consumption since the robot can walk stably at a higher waist height with almost stretched knees. Figure 4.16 shows the comparison of the knee joint motion between the conventional walking pattern and the presented walking pattern at the walking speed of 0.05 m/s and 0.10 m/s, respectively. With the presented walking pattern, the pitch angles of the knee joint of the leg are close to zero when this leg is in support phase and the other leg swings forward. Hence, the walking of the robot that fulfills the presented walking pattern includes configurations with almost stretched knees.

Regarding the conventional walking pattern, the waist height and the step height are fixed at different walking speeds, so the energy consumption of the knee joint actuators basically remains the same.

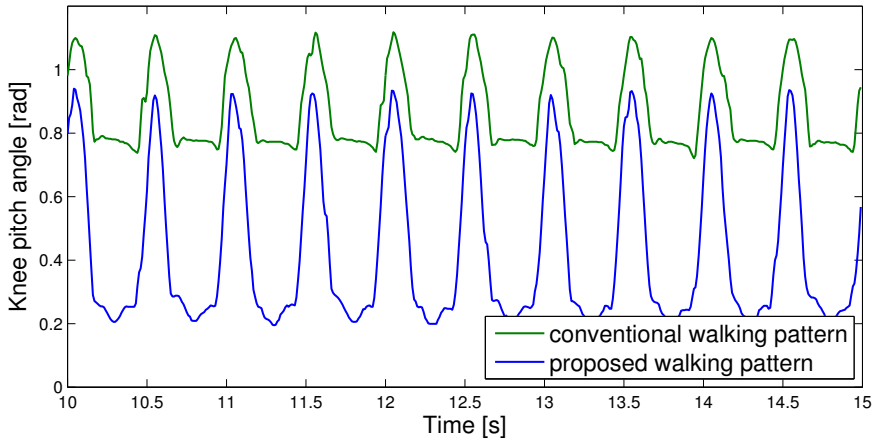
To conclude, the presented walking pattern is more efficient at a lower walking speed, which means the robot fulfilling the proposed walking pattern is able to walk for a longer time without battery charging and has less heating problem.

4.4 Yaw Moment Control

During the walking of the robot, a moment around the yaw-axis of the support foot is generated by the acceleration and deceleration of the swing-leg. Normally this moment is balanced by the frictional moment between the floor and the support foot. However, as the walking



(a) $v = 0.05\text{m/s}$



(b) $v = 0.10\text{m/s}$

Figure 4.16: Knee joint motion at different walking speeds.

speed increases the moment becomes larger. If it is beyond the frictional moment that the floor can provide, the support foot will rotate and slip, then the robot might lose its balance and fall down. Therefore, it is essential to compensate and decrease the moment especially for the fast walking of the robot.

4.4.1 Clues from Human Walking

Human beings compensate not only for the pitch and roll-axis moment but also for the yaw-axis moment by swinging both arms and rotating the waist to maintain the total stability of walking [127]. Perry *et al.* pointed out that a reciprocal arm motion provides a purposeful counterforce to minimize the rotatory displacement of the body by the locomotor mechanics of the legs [93].

4.4.2 Related Works

Humanoid robot LOLA used arm motion to avoid foot slipping. Ulbrich *et al.* argued that introducing elbow joints is advantageous for fast walking, since they permit translational arm swing which is more effective than a purely rotational motion [122]. Biped walking robot WL-12RV takes advantage of three-DoF waist motion to compensate for the yaw-axis moment as the robot walked faster [127]. Hirabayashi *et al.* proposed to compensate the yaw moment by rotating the waist joint of a biped robot [26]. Nagasaka *et al.* proposed optimal gradient method for yaw moment compensation of the humanoid robot H5 [80].

4.4.3 Arm Motion

The yaw moment in this thesis is compensated by arm motion since the humanoid robot NAO does not have waist joint.

First, we analyze the moment around the support leg as the robot walks (refer to Figure 4.17). We discuss the moment around the point O, which is the center of the support foot sole. The robot is regarded as a 4-mass point model consisting of an upper body, a swinging leg and two arms. The masses of the upper body and the swing-leg are m_b and m_s, respectively, and the masses of the left and right arms are m_{al} and m_{ar}, respectively.

The distance between the CoM of the upper body and the support leg is d, and between the two legs is $2d_0$, which is a constant (refer to Figure 3.9). The distance between the two arms is symbolized by $2d_1$. We set the body acceleration $\ddot{x}_b(t)$ in the opposite direction of the swing-leg acceleration $\ddot{x}_s(t)$ and assume that both arms of the robot swing at the same acceleration $\ddot{x}_a(t)$.

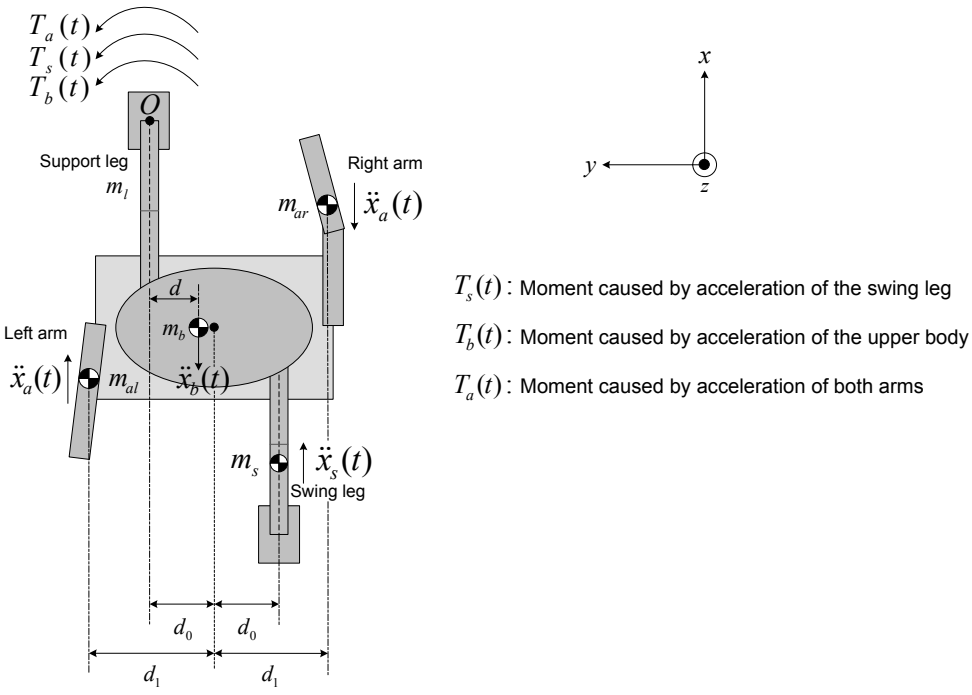


Figure 4.17: Moments around the support leg during walking.

The moment $T_s(t)$ around the point O caused by the swing-leg acceleration $\ddot{x}_s(t)$ is

$$T_s(t) = m_s \ddot{x}_s(t)(2d_0)$$

The moment $T_b(t)$ around the point O caused by the acceleration of the upper body $\ddot{x}_b(t)$ is

$$T_b(t) = -m_b \ddot{x}_b(t)d$$

The moment $T_a(t)$ around the point O caused by the acceleration of the both arms $\ddot{x}_a(t)$ is

$$T_a(t) = -m_{ar} \ddot{x}_a(t)(d_0 + d_1) - m_{al} \ddot{x}_a(t)(d_1 - d_0)$$

Then the amount of the moment $T(t)$ around the point O is calculated by Eq.(4.18).

$$\begin{aligned} T(t) &= T_s(t) + T_b(t) + T_a(t) \\ &= 2m_s d_0 \ddot{x}_s(t) - m_b \ddot{x}_b(t)d - m_{ar} \ddot{x}_a(t)(d_0 + d_1) - m_{al} \ddot{x}_a(t)(d_1 - d_0) \end{aligned} \tag{4.18}$$

It is desirable to keep the moment $T(t)$ zero during the walking. By setting $T(t) = 0$ in Eq.(4.18), $\ddot{x}_a(t)$ is calculated as follows

$$\ddot{x}_a(t) = \frac{2m_s d_0 \ddot{x}_s(t) - m_b \ddot{x}_b(t)d}{m_{al}(d_1 - d_0) + m_{ar}(d_0 + d_1)} \tag{4.19}$$

This is the desired acceleration of arms for compensating the yaw moment. The upper body acceleration is negligible because $\ddot{x}_s(t) \gg \ddot{x}_b(t)$. By integrating Eq.(4.19) twice, and with the initial conditions $x_a(0) = x_s(0) = 0$, $\dot{x}_a(0) = \dot{x}_s(0) = 0$, we obtain the position of arms:

$$x_a(t) = \frac{2m_s d_0}{m_{al}(d_1 - d_0) + m_{ar}(d_0 + d_1)} x_s(t) \tag{4.20}$$

Eq.(4.20) shows that the robot is expected to swing the arms based on the trajectories of the swing-leg in the single support phase. In this thesis, the arm motion used for compensating the yaw moment is realized by the pitch joint in the shoulder. Since the position of the CoM of both arms can be calculated in real time, the shoulder pitch angle can be expressed in terms of arcsin function.

$$\theta_{pitch}^{shoulder}(t) = \arcsin \frac{x_a(t)}{d_a} \tag{4.21}$$

where d_a is the distance between the CoM of the arm and the shoulder joint. $\theta_{pitch}^{shoulder}$ is sent to the servomotor as reference trajectories of the shoulder joints of the robot.

4.4.4 Simulation and Experiments

Figure 4.18 shows the compensatory shoulder pitch angles at three different walking speeds both in simulation and in experiments. When the robot walks faster, arm motion with larger swing amplitude is needed. Seen from the results, the shoulder pitch angles for yaw moment compensation are bigger in experiments than in simulation. The reason is that, under the same walking parameters the walking speed is higher in experiments, which leads to larger step length and thus larger shoulder angles according to Eq.(4.20).

Currently, the walking period of the robot is 0.5 s in order to avoid oscillation and re-bounding of the landing foot brought about by the big landing impact force. As a result, the

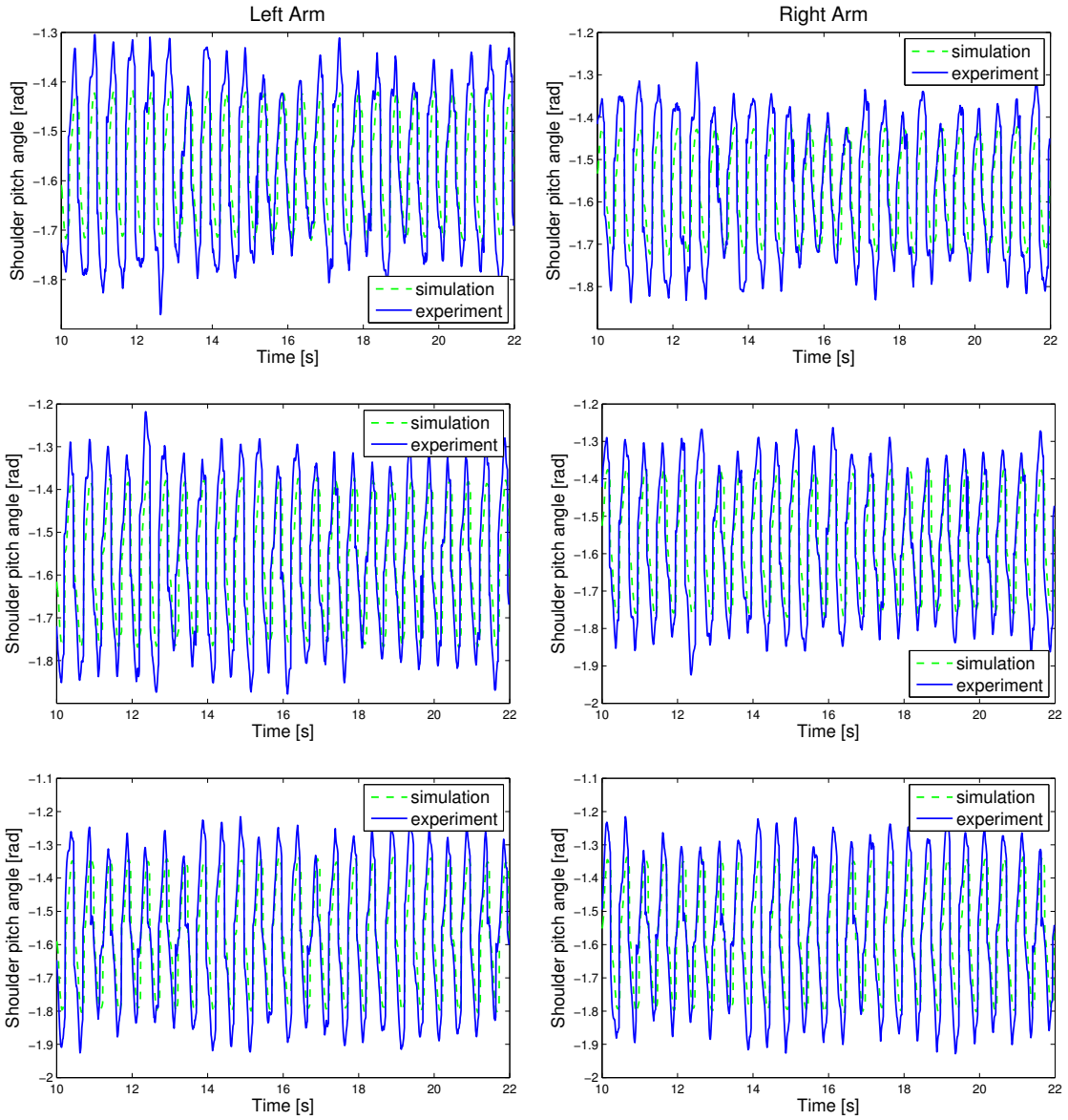


Figure 4.18: Comparison of compensation angles in simulation and experiments. The robot walks 81 steps in 20 seconds. $T_{stride} = 0.50$ s, $A_l = 0.001$ m. The average walking speed $\bar{v}$ is (top) 0.1095 m/s (middle) 0.1571 m/s (bottom) 0.1940 m/s, respectively. The average step length $\bar{L}$ is (top) 0.0270 m (middle) 0.0388 m (bottom) 0.0479 m, respectively.

robot has to swing its arms very rapidly. The upper body of the robot appears to vibrate due to the fast movement of the arms. The walking is consequently unstable. Extending the walking period can make the arm movement slower. The problem, however, caused by the landing impact force will arise again. Chapter 6 focuses on decreasing the landing impact force by using force control approach.

4.5 Chapter Summary

In this chapter, a posture control method to compensate for the upper body inclination of a humanoid robot with the purpose of stable dynamic walking is presented. The online posture controller is based on sensory feedback and modifies the reference trajectories in real time in order to stabilize the robot. The presented control algorithm does not require complex dynamic equations and parameter adjustment. Experiments have shown the validity of controlling the upper body posture by means of both hip strategy and ankle strategy. NAO achieved stable dynamic walking and reached a maximum walking speed of 0.24 m/s in experiments.

Experiments have also demonstrated the efficiency of the walking pattern presented in this thesis which involves three-dimensional upper body motion. Compared with the conventional walking pattern, the knee joints of the presented walking pattern consume less energy when the robot walks at a lower speed since the robot can walk stably with almost stretched knees.

In order to maintain the total walking stability, this chapter also presents a control method to compensate the yaw moment around the support foot by using arm motion. Experiments verified that larger swing amplitude of the arms is needed when the robot walks faster. However, adding the arm motion made the walking unstable. One possible solution is to prolong the walking period to slow the arm movement. But this would bring about oscillation and rebounding of the landing foot caused by the big landing impact force, which would still result in ustable walking. A force control scheme aiming at reducing the landing impact force will be presented in Chapter 6.

5 Swing-leg Trajectory Tracking

5.1 Introduction

From this chapter onward, the control methods that regulate the motion of the lower extremities of the robot are presented. In the course of walking, there are two distinct modes for the lower extremities, *free space motion* and *constrained motion*. When the lower extremity is in swing phase, its motion is free since it has no interaction with the environment. Position control with properly tuned gains should be used for free space motion. While the lower extremity is in stance phase, its motion is constrained because of the interaction with the environment. Especially, a transient phenomenon of landing impact forces occurs during transition from the free space motion to constrained motion. Force control approach should be used to control the impact forces and to track forces once stable contact with the ground has been established.

In this chapter, adaptive control is applied to controlling the free space motion of the robot leg in order to achieve fast and precise tracking of the reference walking pattern. The robot can quickly swing the swing-leg so that fast walking can be achieved. The adaptive control is based on the dynamic model of the swing-leg and compensates the modeling inaccuracies. The model is updated and the performance of the adaptive controller is improved as adaptation goes on. The control method presented in this chapter has been verified in simulation based on both Matlab/SimMechanics and the ODE$^{(1)}$.

5.2 Equations of Motion

In Appendix C, we derive the dynamic equations of motion for the swing-leg. These equations can be written as

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \vec{g}(q) = \boldsymbol{\tau}$$

where $q \in \mathbb{R}^n$ is a vector of joint positions, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the generalized joint torque vector, $H(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix and it is symmetric, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ represents the centrifugal and Coriolis terms, and $\vec{g}(q) \in \mathbb{R}^n$ is the gravitational vector, and n is the number of the joints.

5.3 Adaptive Control

Adaptive control is one of the two major and complementary approaches to dealing with model uncertainty[1]. We use the adaptive control approach put forward by Slotine and Li [103].

[1]The other approach is robust control.

The design of an adaptive controller usually involves the following three steps:

- choose a control law containing variable parameters for the actuator torques
- derive an adaptation law for those unknown parameters of the model such that the system output $q(t)$ closely tracks the desired trajectory $q_d(t)$
- analyze the convergence properties of the resulting control system

Figure 5.1 shows the control diagram using the adaptive controller. The model under the adaptive control is updated online, based on the measured performance.

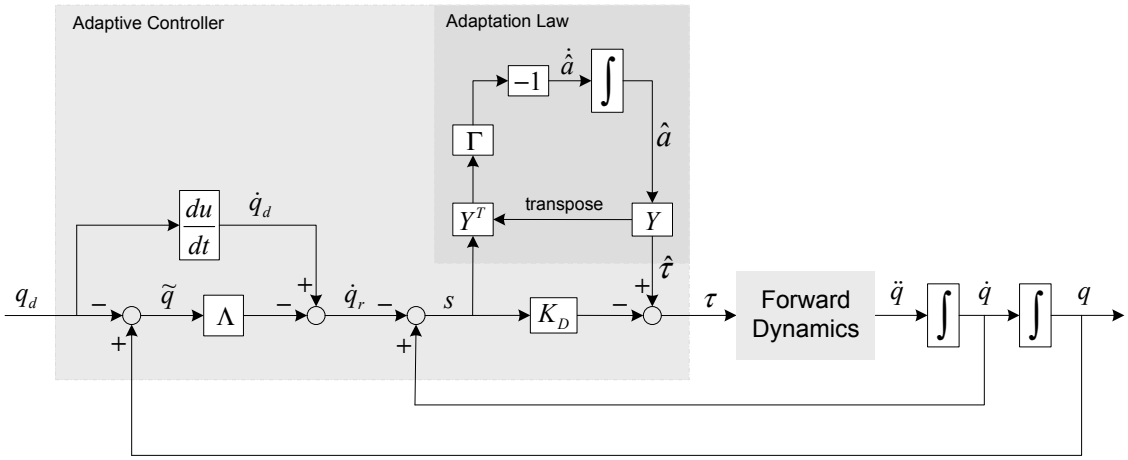


Figure 5.1: Control diagram of adaptive control.

Control Law

Taking the control law to be

$$\tau = \hat{\tau} - K_D s \tag{5.1}$$

which includes a feedforward term $\hat{\tau}$ in addition to a feedback PD term $K_D s$, where

$$\hat{\tau} = \hat{H}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{g}(q) \tag{5.2}$$

Let us define $\tilde{q} = q - q_d$ as the tracking error, and $\dot{q}_r = \dot{q}_d - \Lambda\tilde{q}$ as the reference velocity vector which is formed by shifting the desired velocities $\dot{q}_d$ according to the position error $\tilde{q}$. Λ is a symmetric positive definite matrix. Let us define the vector $s = \dot{\tilde{q}} + \Lambda\tilde{q}$, then from the definitions of $\tilde{q}$ and $\dot{q}_r$ we have

$$s = \dot{q} - \dot{q}_r \tag{5.3}$$

Here s can be interpreted as a velocity error term, as shown in Figure 5.1.

Assuming bounded initial conditions, showing the boundedness of s also shows the boundedness of $\tilde{q}$ and $\dot{\tilde{q}}$, and therefore of q and $\dot{q}$; similarly, if s tends to 0 as t tends to infinity, so do the vectors $\tilde{q}$ and $\dot{\tilde{q}}$.

We use the reparametrization technique to linearize Eq.(5.2). There exists a known matrix $Y = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ such that

$$H(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + g(q) = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)a \tag{5.4}$$

where a is the constant vector of unknown parameters describing the system's equivalent mass and inertia properties. Matrix Y is derived in Appendix C given the definition of the vector a.

Then

$$\hat{\tau} = Y\hat{a} \tag{5.5}$$

where $\hat{a}$ is the estimate of a.

Adaptation Law

The adaptation law

$$\dot{\hat{a}} = -\Gamma Y^T s \tag{5.6}$$

is chosen to update the parameter estimates $\hat{a}$ according to the correlation integrals. Γ is a symmetric positive definite matrix of adaptation weights.

Global Stability Validation

To prove convergence of both error and the dynamic parameters, we use the following Lyapunov function candidate:

$$V(t) = \frac{1}{2}\left[s^T H s + \tilde{a}^T \Gamma^{-1}\tilde{a}\right] \tag{5.7}$$

where $\tilde{a} = \hat{a} - a$, defined as the parameter estimation error.

Differentiating Eq.(5.7),

$$\dot{V}(t) = s^T H\dot{s} + \frac{1}{2}s^T\dot{H}s + \dot{\hat{a}}^T\Gamma^{-1}\tilde{a} \tag{5.8}$$

Note on Eq. (5.8)

If a vector $x = x(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^T$ *and a symmetric matrix* $A = A(t) = (A_{(ij)}(t))_{(n\times n)}$ *are differentiable, then the first-order derivative of the quadratic form* $x^T Ax$ *with respect to the variable* t *is*

$$\frac{\mathrm{d}}{\mathrm{d}t}(x^T Ax) = x^T \dot{A}x + 2x^T A\dot{x}.$$

Proof:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(x^T Ax) &= \big[\frac{\mathrm{d}}{\mathrm{d}t}(x^T)\big]Ax + x^T\frac{\mathrm{d}}{\mathrm{d}t}(Ax)\\ &= \big[\frac{\mathrm{d}}{\mathrm{d}t}(x^T)\big]Ax + x^T\dot{A}x + x^TA\dot{x}\end{aligned}$$

Also

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}(x^T)Ax\right]^T = x^TA^T\dot{x} = x^TA\dot{x}$$

Since $\big[\frac{\mathrm{d}}{\mathrm{d}t}(x^T)\big]Ax$ *is a scalar, it is equal to its transpose. So*

$$\big[\frac{\mathrm{d}}{\mathrm{d}t}(x^T)\big]Ax = x^TA\dot{x}.$$

Therefore, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}(x^T Ax) = x^T \dot{A}x + 2x^T A\dot{x}.$$

Combining Eq.(5.3) results in

$$\dot{V}(t) = s^T(H\ddot{q} - H\ddot{q_r}) + \frac{1}{2}s^T\dot{H}s + \dot{\hat{a}}^T\Gamma^{-1}\tilde{a} \tag{5.9}$$

Substituting $H\ddot{q}$ from the system dynamics $H\ddot{q} = \tau - C\dot{q} - g = \tau - C(s + \dot{q_r}) - g$ yields

$$\begin{aligned}\dot{V}(t) &= s^T(\tau - H\ddot{q_r} - C\dot{q_r} - g) - s^TCs + \frac{1}{2}s^T\dot{H}s + \dot{\hat{a}}^T\Gamma^{-1}\tilde{a}\\ &= s^T(\tau - H\ddot{q_r} - C\dot{q_r} - g) + \dot{\hat{a}}^T\Gamma^{-1}\tilde{a}\end{aligned} \tag{5.10}$$

Since components of C are derived using Eq.(C.14), the matrix $(\dot{H} - 2C)$ is skew-symmetric[2]. Therefore, for a given vector s, $s^T(\frac{1}{2}\dot{H} - C)s = 0$.

[2] In mathematics, particularly in linear algebra, a skew-symmetric matrix is a square matrix A whose transpose is also its negative; that is, it satisfies the condition $-A = A^T$.

Note on Eq. (5.10)

The quadratic function associated with a skew-symmetric matrix is always zero.

Proof: let M be a $n \times n$ skew-symmetric matrix and x an arbitrary $n \times 1$ vector. Then the definition of a skew-symmetric matrix implies that

$$x^T M x = -x^T M^T x$$

Since $x^T M^T x$ is a scalar, the right-hand side of the above equation can be replaced by its transpose. Therefore,

$$x^T M x = -x^T M x$$

This shows that

$$\forall x, x^T M x = 0$$

Substituting Eq.(5.1) results in

$$\dot{V}(t) = s^T(\hat{\tau} - K_D s - H\ddot{q}_r - C\dot{q}_r - g) + \dot{\hat{a}}^T \Gamma^{-1} \tilde{a} \tag{5.11}$$

Substituting Eq.(5.4) and Eq.(5.5) yields

$$\begin{aligned} \dot{V}(t) &= s^T(Y\hat{a} - Ya) - s^T K_D s + \dot{\hat{a}}^T \Gamma^{-1} \tilde{a} \\ &= s^T Y \tilde{a} - s^T K_D s + \dot{\hat{a}}^T \Gamma^{-1} \tilde{a} \end{aligned} \tag{5.12}$$

Using Eq.(5.6), we get

$$\dot{V}(t) = -s^T K_D s \leq 0 \tag{5.13}$$

Therefore, both global stability of the system and convergence of the tracking error are guaranteed by the above adaptive controller.

5.4 Simulation Results

As shown in Figure 5.1, the adaptive controller produces the applied joint actuators torques. Given the specified torques, the resulting joint accelerations are to be determined, that is $\tau \Longrightarrow \ddot{q}$. This is known as the forward dynamics problem. Forward dynamics is required for robot simulation [15]. Then by integrating these accelerations, the simulated plant state, i.e. the velocities and positions of joints, is updated as functions of time.

5.4.1 Simulation Based on Matlab/SimMechanics

Matlab/SimMechanics[3] provides a multibody simulation environment for 3D mechanical systems including robots. The multibody system is modeled using blocks representing bodies, joints, constraints, and force elements, and then SimMechanics formulates and solves the equations of motion for the complete mechanical system [120]. SimMechanics also enables us to use blocks from the Simulink environment. In this way, it is possible to design a mechanical model in SimMechanics and, simultaneously, incorporate a control part of Simulink into the unified environment.

[3]The Mathworks, Inc., Natick, MA, USA.

SimMechanics software provides an intuitive, less time-consuming and less effort-requiring way of solving the forward dynamics of the simulated mechanical system. In the forward dynamics mode, a SimMechanics simulation uses the Simulink suite of ordinary differential equation (ODE$^{(2)}$) solvers to solve Newton's equations and obtain the resulting motions [121].

To implement the model-based adaptive contol and compare the tracking performance of adaptive control with PD control, we use the simulation and analysis tools provided by SimMechanics to simulate the forward dynamics of the swing-leg of the robot. We start with the simple planar model of the swing-leg to test ideas, then we extend the planar model to a three-dimensional model. In simulation, we take the masses, the positions of the CoM and the inertia tensors of the leg segments of the real robot as the model parameters. The adaptive/PD control and forward dynamics were implemented in Matlab/Simulink R2012b with the solver ode15s.

5.4.1.1 Planar Model Simulation

Step Response

The robot leg, initially at rest at $[q_1 \quad q_2 \quad q_3] = [0 \quad 0 \quad 0]$, is step-commanded to a desired configuration, for example, $[q_{d1} \quad q_{d2} \quad q_{d3}] = [-60^\circ \quad 45^\circ \quad -30^\circ]$. The actual parameters of the leg model[4] used in the simulation (left leg, refer to Figure C.1) are

$m_1 = 0.3897\ kg$	$m_2 = 0.2914\ kg$	$m_3 = 0.1618\ kg$
$I_1 = 0.00159107278\ kg \cdot m^2$	$I_2 = 0.00112865226\ kg \cdot m^2$	$I_3 = 0.00064434250\ kg \cdot m^2$
$l_1 = 0.1\ m$	$l_2 = 0.1029\ m$	$l_3 = 0.04519\ m$
$b_1 = 0.05373\ m$	$b_2 = 0.04936\ m$	$b_3 = 0.03239\ m$

leading to

$$\begin{aligned}\boldsymbol{a} &= [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8 \quad a_9]^T \\ &= [0.00725 \quad 0.0031 \quad 0.00355 \quad 0.000539 \quad 0.000814 \quad 0.000524 \quad 0.650 \quad 0.3044 \quad 0.0514]^T.\end{aligned}$$[5]

The corresponding tracking results, tracking errors[6] and control torques using adaptive control are plotted in Figure 5.2, starting without any a priori information ($\hat{a}(0) = 0$), with $\Gamma = \text{diag}[0.5 \quad 0.1 \quad 0.1 \quad 0.3 \quad 0.08 \quad 0.04 \quad 9 \quad 15 \quad 6]$, $\Lambda = 200\text{I}$ and $K_D = 100\text{I}$. The parameters of the adaptive controller were chosen empirically[7].

[4] The parameters of the leg model used in the simulation, such as the mass, the position of the CoM and the inertia, are derived from the technical data of the real robot provided by Aldebaran Robotics.

[5] The definition of the vector $\boldsymbol{a}$ for the planar model can be found in Appendix C.

[6] Tracking error is a measure of how closely the system response q follows the reference input q_d. In this chapter, tracking error is denoted as $\tilde{q}$ and defined as $\tilde{q} = q - q_d$.

[7] In this chapter, to determine the parameters of an adaptive controller, such as Γ, Λ and K_D, a large number of simulations were implemented. The preferred parameters were proved to be the best in view of the performance of the controller. Once the parameters of the controller were determined, the simulation results, such as the tracking results, tracking errors, control torques and dynamic parameter estimates can be obtained theoretically via only one simulation given the mechanical system, the initial conditions of the system and the desired input signal. Since SimMechanics is a numerical solver, the same numerical solutions (simulation results) will be obtained at each instant of time when the mechanical system, the contoller and the input signals are specified.

The top graph of Figure 5.3 shows adaptation of the first 6 parameters which represent the dynamic inertial terms, within 2.5 seconds. The bottom graph of Figure 5.3 shows adaptation of the gravitational terms within 500 seconds. The parameters converge resulting in the following values

$$\begin{aligned} a = \quad & [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8 \quad a_9]^T \\ = \quad & [-0.0260 \quad -0.0185 \quad 0.0105 \quad 0.0078 \quad -0.0021 \quad -0.0046 \quad 0.6552 \quad 0.3391 \quad 0.0944]^T. \end{aligned}$$

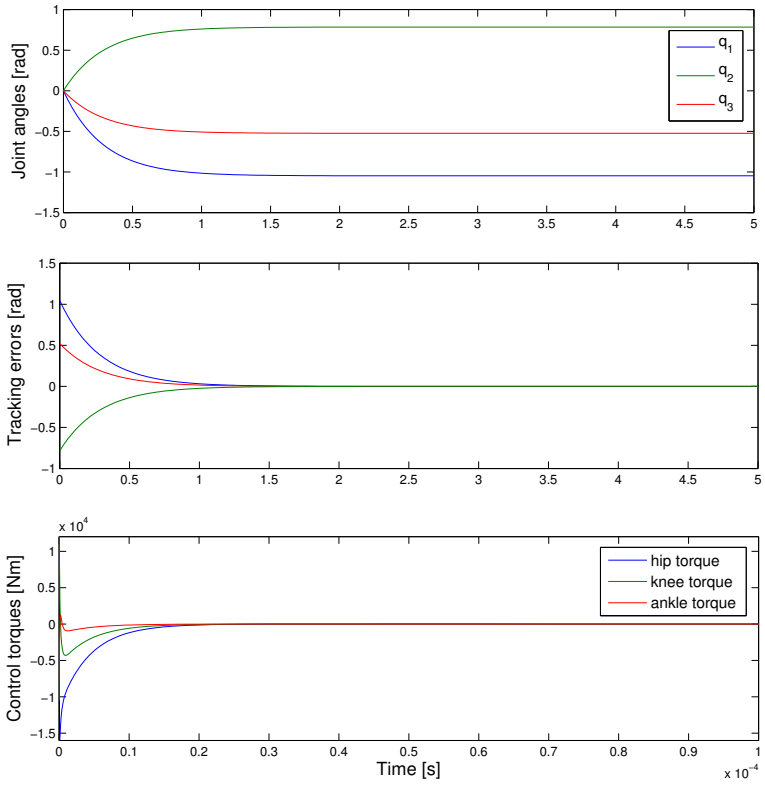


Figure 5.2: Tracking performance (top), tracking errors (middle) and control torques (bottom) with adaptive control implemented on the planar model using step function as input.

For comparison, we also use a PD controller to control the swing-leg for the same desired joint position. The control law used for the PD controller is

$$\tau = -\overline{K}_P\tilde{q} - \overline{K}_D\dot{q} \tag{5.14}$$

where $\overline{K}_P$ and $\overline{K}_D$ are constant diagonal positive matrices, and they are chosen to be equivalent to the feedback PD term $-K_D s$ in the adaptive controller. According to the definition of the vector s ($s := \dot{\tilde{q}} + \Lambda\tilde{q}$),

$$-K_D s = -K_D(\dot{\tilde{q}} + \Lambda\tilde{q}) = -\Lambda K_D\tilde{q} - K_D\dot{\tilde{q}} \tag{5.15}$$

For the adaptive controller, the used parameters are $\Lambda = 200\mathrm{I}$ and $K_D = 100\mathrm{I}$. Therefore, the equivalent gains for the PD controller are $\overline{K}_D = K_D = 100I$ and $\overline{K}_P = \Lambda K_D = 200K_D$. Figure 5.4 shows the tracking errors and control torques using the PD controller.

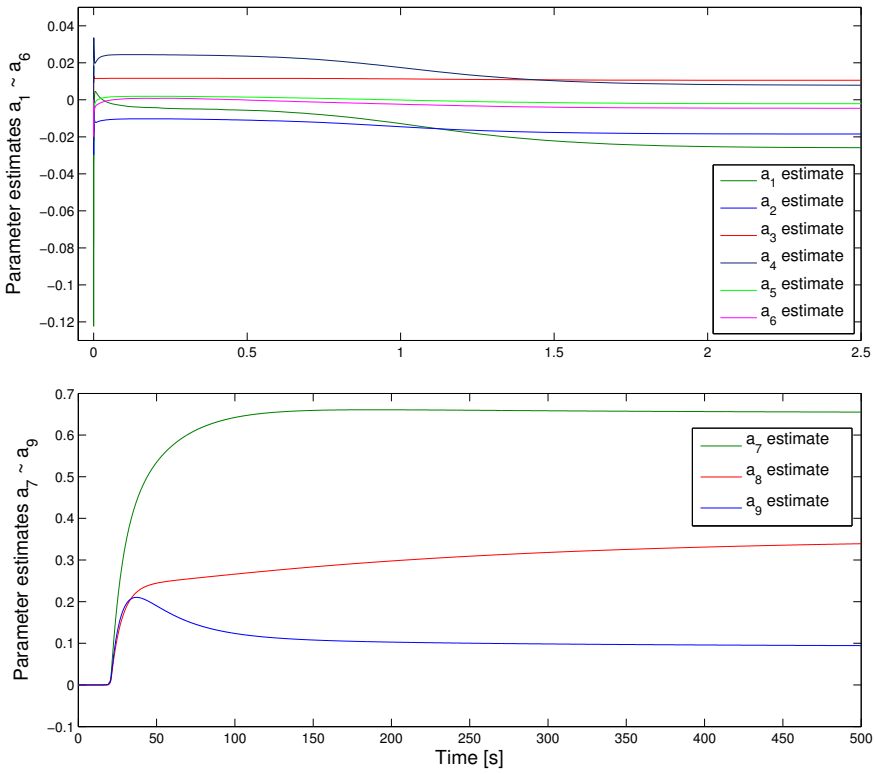


Figure 5.3: Inertial parameter estimates (top) and gravitational parameter estimates (bottom) of the planar model under step response with adaptive control.

Trajectory Tracking

We now consider the case when the swing-leg is actually required to follow a desired trajectory, rather than merely reach a desired position. To compare the real-time tracking performances of adaptive control and PD control, we used three signals, each consisting of a large amplitude, low frequency, sine wave and a low amplitude, high frequency, sine wave, as desired trajectories of the hip, knee and ankle joints.

$$
\begin{aligned}
q_{d1} &= 0.8\sin(2\pi(0.05)t) + 0.25\sin(2\pi(1.0)t) \\
q_{d2} &= q_{d1} + 0.4\sin(2\pi(0.03)t) + 0.15\sin(2\pi(1.5)t) \\
q_{d3} &= q_{d2} + 0.6\sin(2\pi(0.04)t) + 0.1\sin(2\pi(2)t)
\end{aligned}
$$

The parameters used for adaptive control are $\Gamma = \text{diag}[0.001 \quad 0.002 \quad 0.002 \quad 0.001 \quad 0.001 \quad 0.001 \quad 2 \quad 0.8 \quad 0.06]$, $\Lambda = 500\text{I}$ and $K_D = 100\text{I}$. For PD control, the equivalent parameters are $K_D = 100I$ and $K_P = \Lambda K_D = 500K_D$. All the parameters here were chosen empirically.

Figure 5.5 shows the tracking results and the tracking errors using adaptive control and PD control, respectively. We see that tracking is poor when the dynamic parameters are all at zero. When adaptation turns on, the tracking becomes better and good tracking is achieved. Therefore, the adaptive control handles the dynamic demands of trajectory tracking more effectively than the PD control. The top graph of 5.6 shows adaptation of the first 6

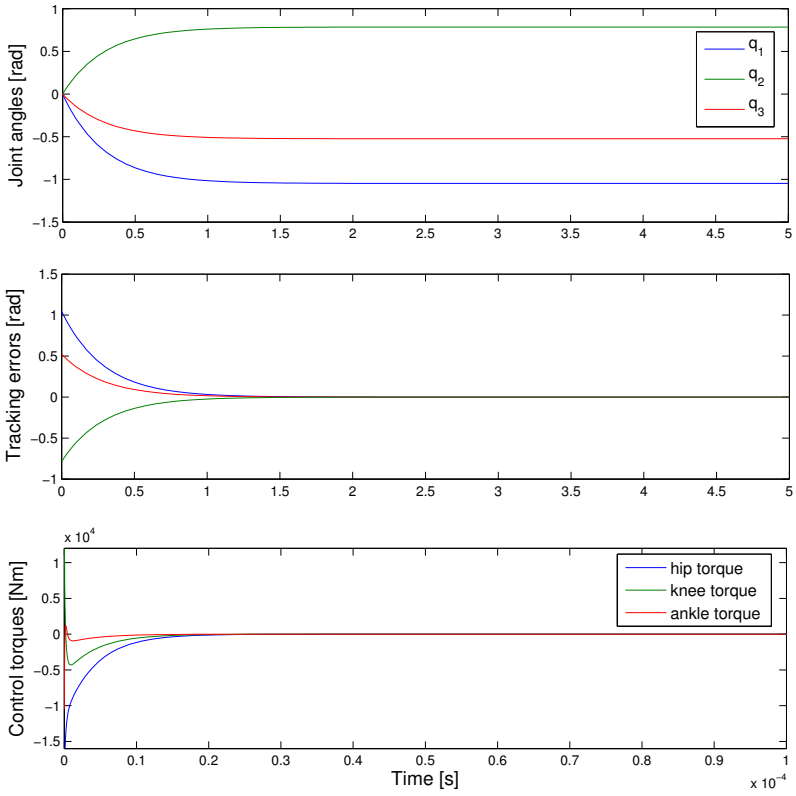


Figure 5.4: Tracking performance (top), tracking errors (middle) and control torques (bottom) with PD control on the planar model using step function as input.

parameters which represent the dynamic inertial terms, within 20 seconds. The bottom graph of Figure 5.6 shows adaptation of the gravitational terms within 300 seconds.

5.4.1.2 3D Model Simulation

Then we extend the 2D planar model of the swing-leg to the 3D model by adding two joints, the hip roll joint and ankle roll joint to the frontal plane (refer to Figure C.2). The adaptive control approach to fast and precise tracking of the reference walking pattern of the swing-leg is further verified through simulations on the 3D model. Figure 5.7 shows the simulation diagram to solve the forward dynamics of the 3D model of the swing-leg using SimMechanics in Matlab/Simulink.

Step Response

The robot leg, initially at rest at $[q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5] = [0 \quad 0 \quad 0 \quad 0 \quad 0]$, is step-commanded to a desired configuration, for example, $[q_{d1} \quad q_{d2} \quad q_{d3} \quad q_{d4} \quad q_{d5}] = [-18° \quad -30° \quad +30° \quad -45° \quad -20°]$. The actual parameters used in the simulation, such as the masses, the positions of the CoM and the inertia tensors of the leg segments (left leg), are derived from the technical

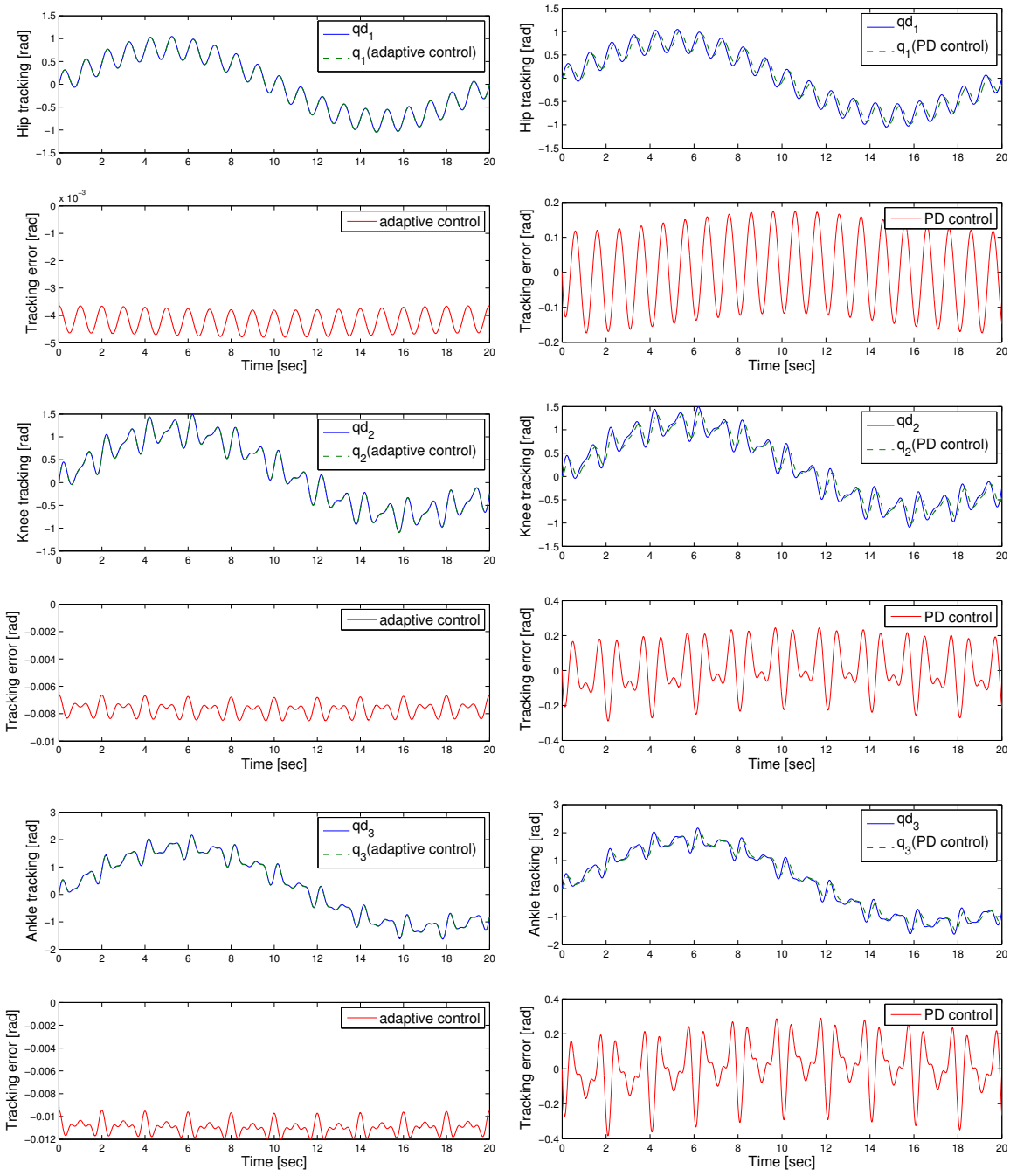


Figure 5.5: Tracking performance comparison using adaptive control (left column) and PD control (right column) on the planar model taking dynamic sine waves as input signal.

data of the real robot provided by Aldebaran Robotics, which are listed as follows:

$m_1 = 0.3897\ kg$ $m_2 = 0.2914\ kg$ $m_3 = 0.1618\ kg$

$I_{xx_1} = 0.00163671962\ kg \cdot m^2$ $I_{xx_2} = 0.00118207967\ kg \cdot m^2$ $I_{xx_3} = 0.00026944182\ kg \cdot m^2$

$I_{yy_1} = 0.00159107278\ kg \cdot m^2$ $I_{yy_2} = 0.00112865226\ kg \cdot m^2$ $I_{yy_3} = 0.00064434250\ kg \cdot m^2$

$l_1 = 0.1\ m$ $l_2 = 0.1029\ m$ $l_3 = 0.04519\ m$

$b_1 = 0.0538\ m$ $b_2 = 0.0496\ m$ $b_3 = 0.0413\ m$

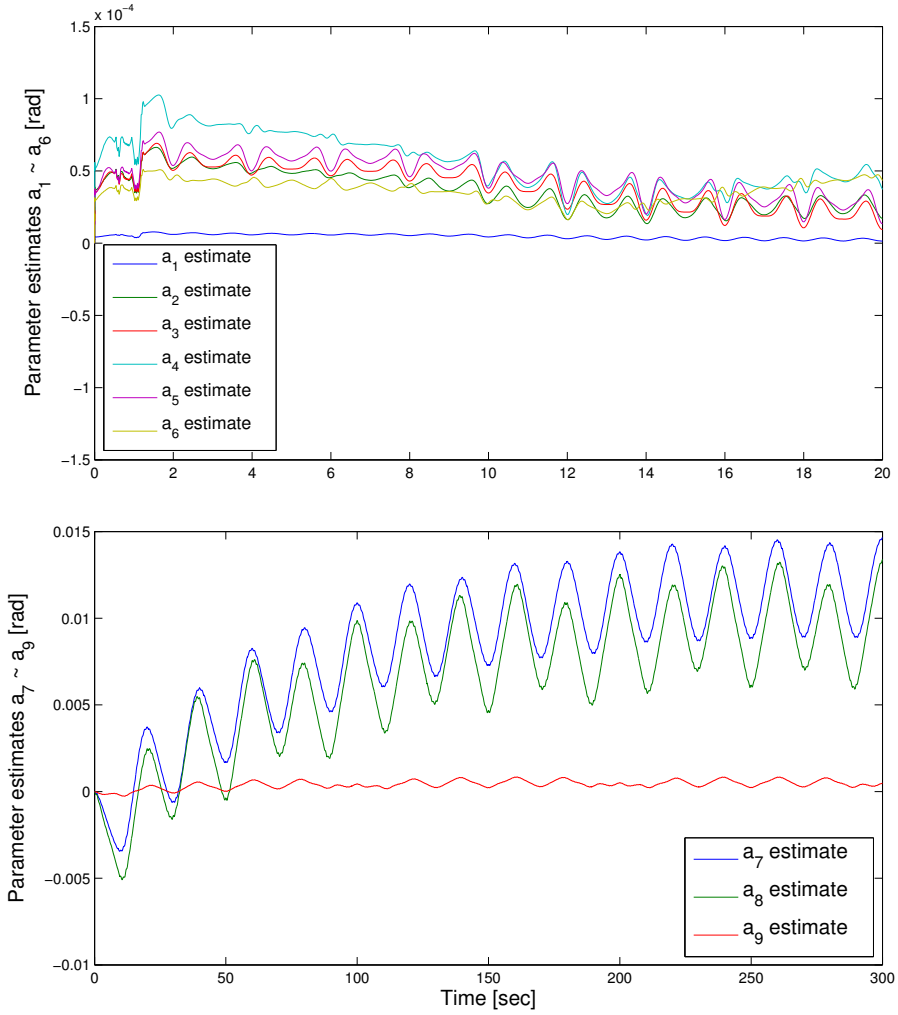


Figure 5.6: Dynamic estimates of inertial parameters (top) and gravitational parameters (bottom) of the planar model for trajectory tracking under adaptive control.

leading to

$$\begin{aligned}\boldsymbol{a} = \; & [a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_{15} \quad a_{16} \quad a_{17}]^T \\ = \; & [0.0011 \quad 0.000717 \quad 0.0029 \quad 0.000276 \quad 0.0016 \quad 0.0017 \quad 0.000668 \quad 0.000688 \quad 0.0031 \\ & 0.0028 \quad 0.0002694 \quad 0.0016 \quad 0.0011 \quad 0.000644 \quad 0.65 \quad 0.305 \quad 0.0656]^T.\end{aligned}$$ [8]

The corresponding tracking performance, tracking errors and control torques using adaptive control are plotted in Figure 5.8, starting without any a priori information ($\hat{a}(0) = 0$), with $\Gamma = \text{diag}[0.05 \quad 0.01 \quad 0.01 \quad 0.03 \quad 0.08 \quad 0.04 \quad 0.05 \quad 0.01 \quad 0.01 \quad 0.03 \quad 0.08 \quad 0.04 \quad 0.01 \quad 0.03 \quad 9 \quad 15 \quad 6]$, $\Lambda = 700\text{I}$ and $K_D = 100\text{I}$. The parameters here were chosen empirically.

[8] The definition of the vector $\boldsymbol{a}$ for the 3D model of the swing-leg can be found in Appendix C.

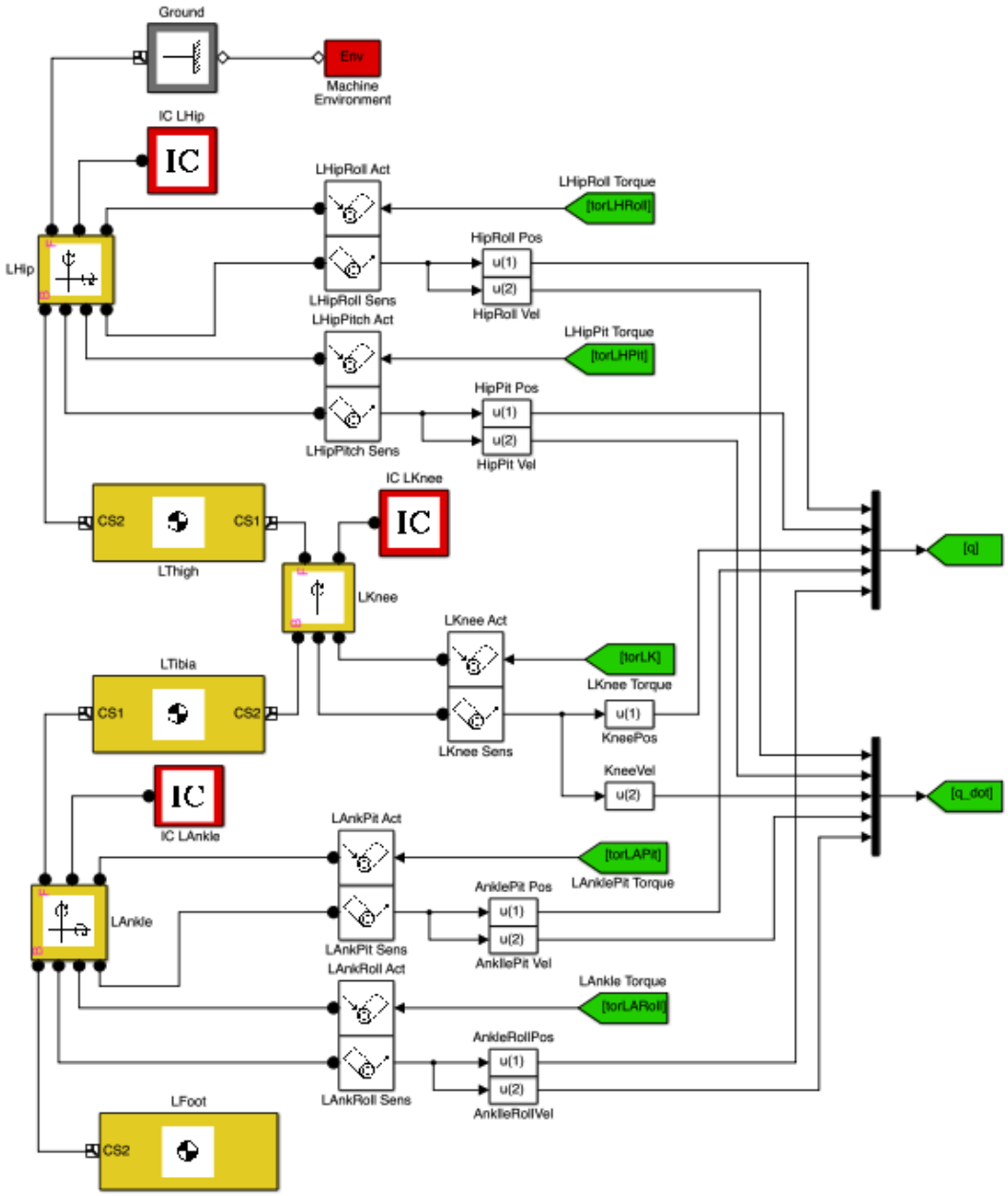


Figure 5.7: Simulation diagram to solve the forward dynamics of the 3D model of the swing-leg using SimMechanics in Matlab/Simulink .

The top graph of Figure 5.9 shows adaptation of the first 14 parameters, which represent the dynamic inertial terms, within 1 seconds. The bottom graph of Figure 5.9 shows adaptation of the gravitational terms within 500 seconds. The parameters converge resulting in the following

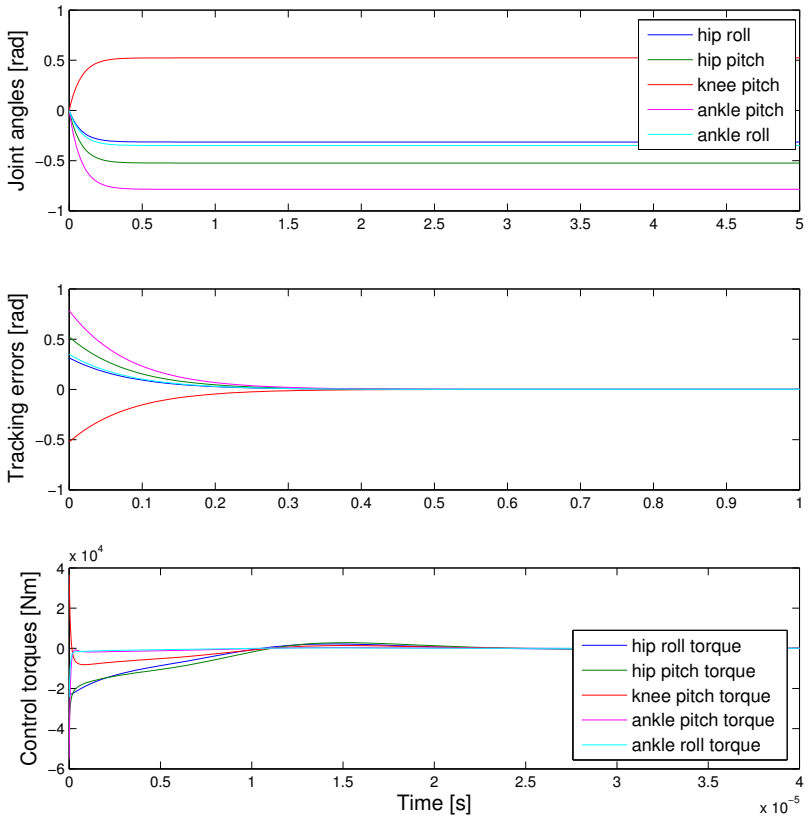


Figure 5.8: Tracking performance, tracking errors and control torques using adaptive control on the 3D swing-leg model, taking step function as input.

values

$$\begin{aligned} a = \; & [a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_{15} \quad a_{16} \quad a_{17}]^T \\ = \; & [0.0026 \quad 0.0121 \quad 0.0005 \quad 0.0026 \quad 0.0042 \quad 0.0484 \quad -0.0243 \quad 0.0015 \quad -0.0037 \\ & -0.0268 \quad -0.0006 \quad 0.0394 \quad 0.0210 \quad 0.0045 \quad 0.6927 \quad 0.1369 \quad 0.0929]^T. \end{aligned}$$

Trajectory Tracking

The swing-leg is now required to follow a desired trajectory. The signals we used as the desired trajectories of the hip, knee and ankle joints were extracted from the offline walking pattern generated by the method presented in Chapter 3.

The parameters used for adaptive control are $\Gamma = \text{diag}[0.02 \quad 0.08 \quad 0.05 \quad 0.01 \quad 0.02 \quad 0.06 \quad 0.01 \quad 0.05 \quad 0.07 \quad 0.03 \quad 0.01 \quad 0.04 \quad 0.05 \quad 0.05 \quad 9 \quad 15 \quad 6]$, $\Lambda = 500\text{I}$ and $K_D = 100\text{I}$. For PD control, the equivalent parameters are $K_D = 100I$ and $K_P = \Lambda K_D = 500K_D$. All the parameters here were chosen empirically.

The tracking results and tracking errors using adaptive control and PD control are plotted in Figure 5.10. We see that the tracking becomes significantly better with adaptive control in comparison to that with PD control. Simulation results show that, the adaptive control

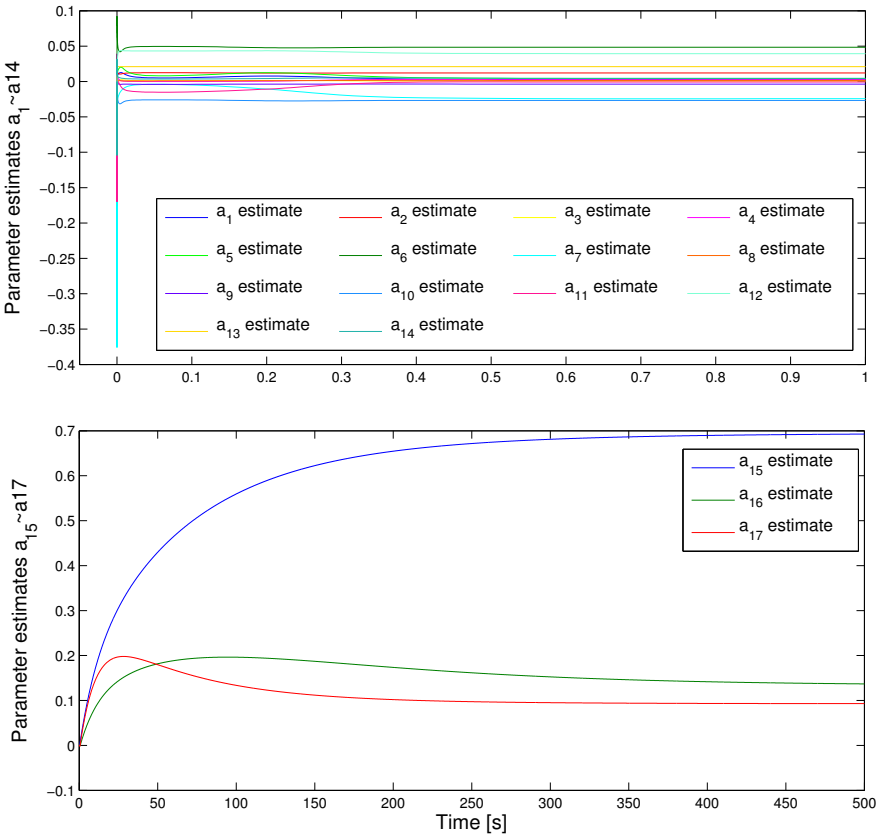


Figure 5.9: Inertial parameter estimates (top) and gravitational parameter estimates (bottom) under step response implementing adaptive control on the 3D swing-leg model.

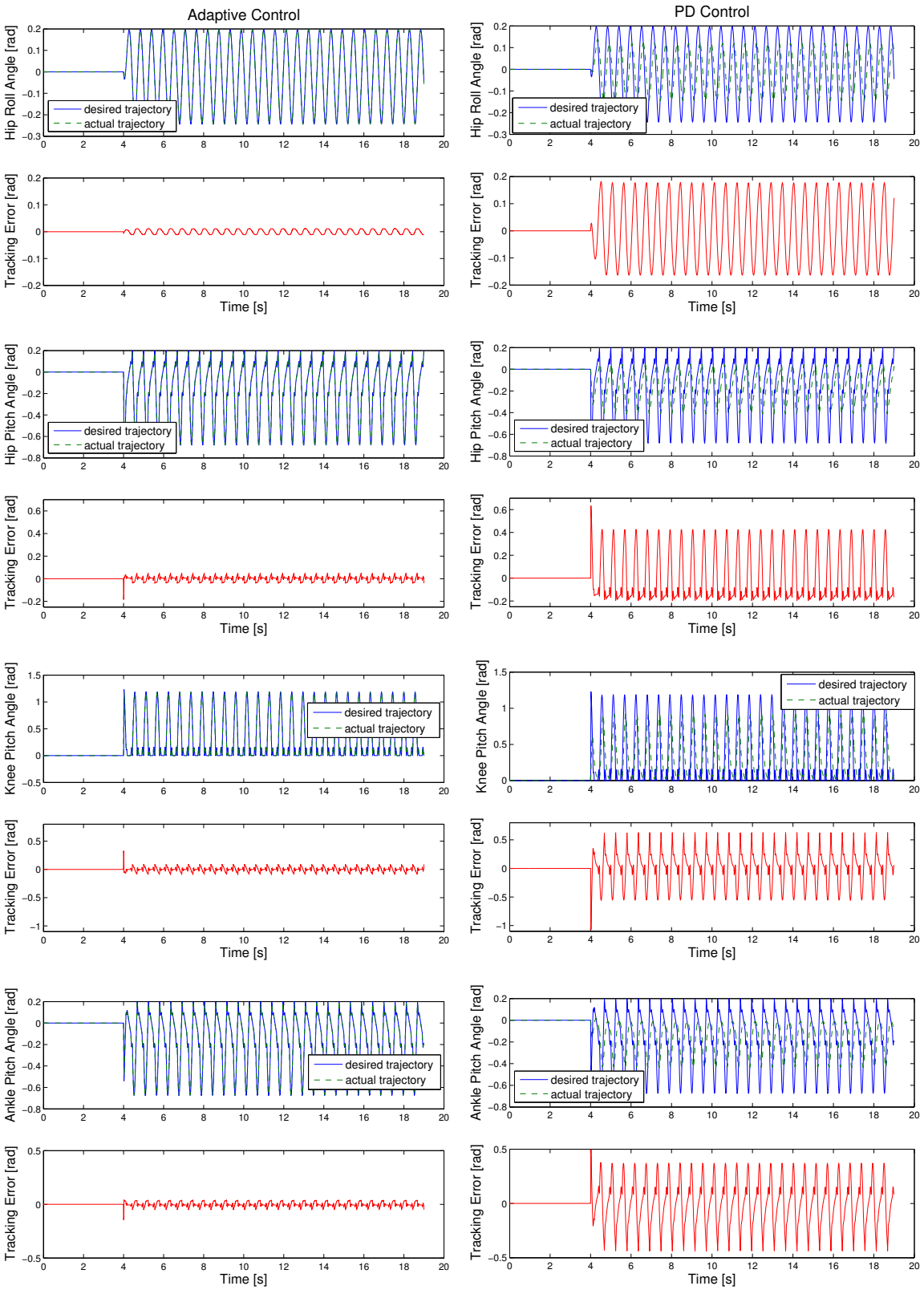
outperforms PD control in handling the dynamic demands of trajectory tracking. Figure 5.11 shows the results of parameter estimates with adaptive control.

5.4.2 Simulation Based on the ODE$^{(1)}$

To further verify the advantage of the adaptive control in dynamic trajectory tracking over the PD control, software simulation was carried out in a robot simulator. The simulator is programmed by C/C++ and is built using the ODE$^{(1)}$ physics engine. Table 5.1 lists the parameters of the robot model in the simulator. Figure 5.12 shows the snapshots of the forward walking in the simulator. The walking parameters used for generating the reference walking pattern are listed in Table 5.2.

Adaptive control and PD control were implemented in the ODE$^{(1)}$ simulator. Figure 5.13 and Figure 5.14 show the zoomed-in views of the trajectory tracking performance for both legs under multiple sets of parameters of the adaptive controller and PD controller. In the adaptive control, Γ = diag[0.002 0.008 0.005 0.001 0.002 0.006 0.001 0.005 0.007 0.003 0.001 0.004 0.005 0.005 0.09 0.015 0.06].

Seen from the simulation results, there is no clear evidence that the adaptive control outperforms the PD control in dynamic trajectory tracking using the 3D model of the swing-


Adaptive Control
PD Control
Hip Roll Angle [rad]
Tracking Error [rad]
Hip Pitch Angle [rad]
Knee Pitch Angle [rad]
Ankle Pitch Angle [rad]
desired trajectory
actual trajectory
Time [s]

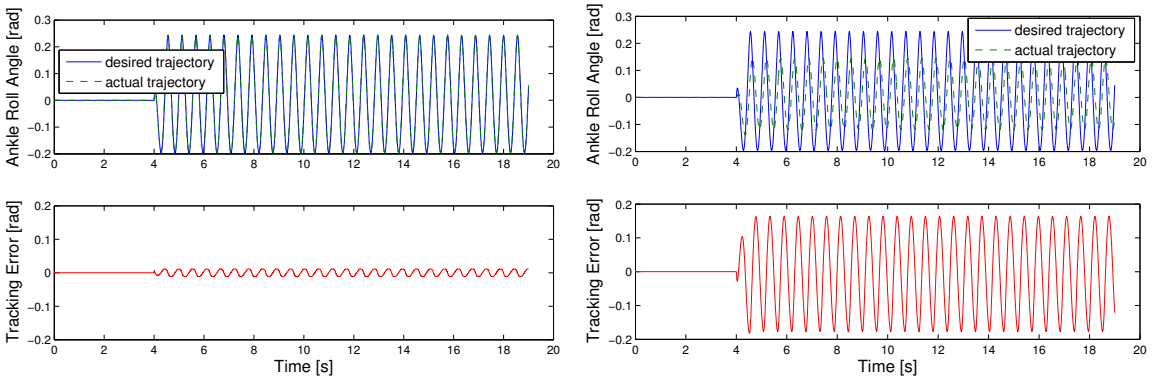


Figure 5.10: Tracking performance comparison by implementing adaptive control (left) and PD control (right) on the 3D swing-leg model taking reference walking pattern as input.

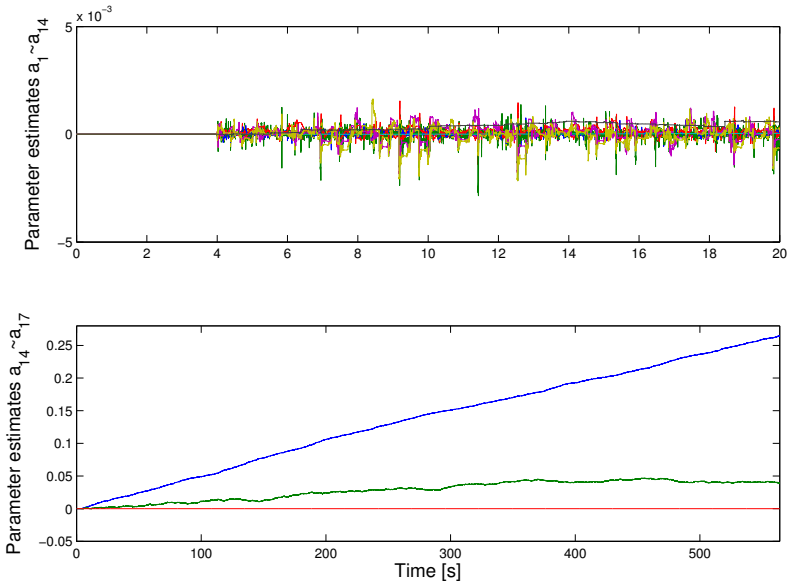


Figure 5.11: Dynamic estimates of inertial parameters (top) and gravitational parameters (bottom) for the 3D swing-leg model under adaptive control using reference walking pattern as input.

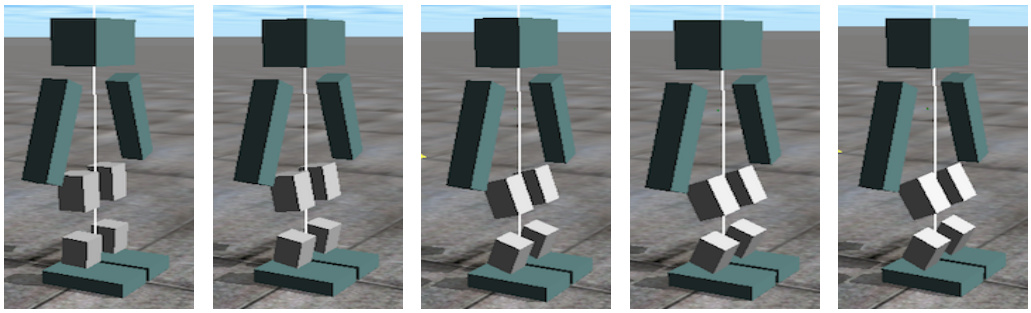
leg in the ODE$^{(1)}$ simulator. The reason is that the robot in the simulator is indeed not a torque-control robot; only very small control torques can be as input, or else the robot will collapse. On the other hand, the adaptive controller and PD controller need to output large torques to achieve precise trajectory tracking. Therefore, the limitation on the magnitude of the control torque with the ODE$^{(1)}$ simulator reduces the accuracy of the position control.

Table 5.1: Parameters of the robot model in the ODE$^{(1)}$ simulator.

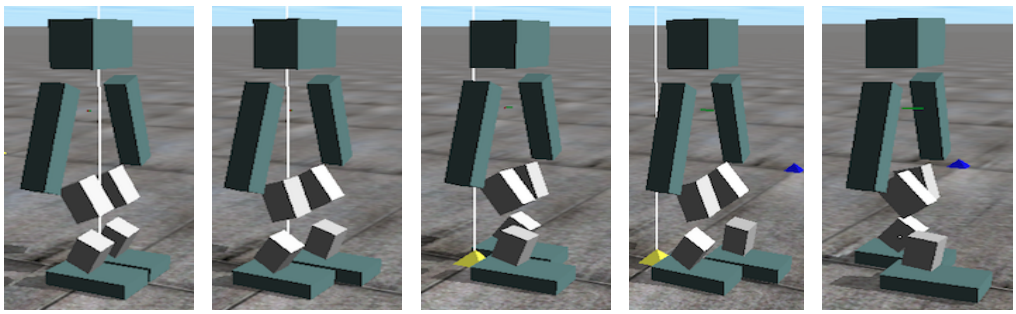
Links	Lengths (m)	Masses (kg)
Thigh	0.1	0.533
Tibia	0.1	0.423
Foot	0.046	0.158
Upper Body		2.118

Table 5.2: Walking parameters for the ODE$^{(1)}$ simulator.

Walking Parameters	Symbol	Value
Lateral swing amplitude of waist	A_l	0.03 (m)
Sagittal oscillation amplitude of waist	A_s	0.001 (m)
Foot collision avoidance distance	d_0	0.05 (m)
Walking period (stride time)	T_{stride}	0.80 (s)
Double-support ratio	τ_{dsp}	0.1 (10%)
Maximum elevation of foot	H_{foot}	0.024 (m)
Average Walking Speed	$\bar{v}$	0.277 (m/s)
Average Step Length	L_{step}	0.112 (m)
Total Simulation Time	T_{total}	32 (s)



(a) t=0s (b) t=0.5s (c) t=0.95s (d) t=1.3s (e) t=1.7s



(f) t=2.0s (g) t=2.4s (h) t=2.75s (i) t=3.15s (j) t=3.5s

Figure 5.12: Snapshots of forward walking in the ODE$^{(1)}$ simulator.

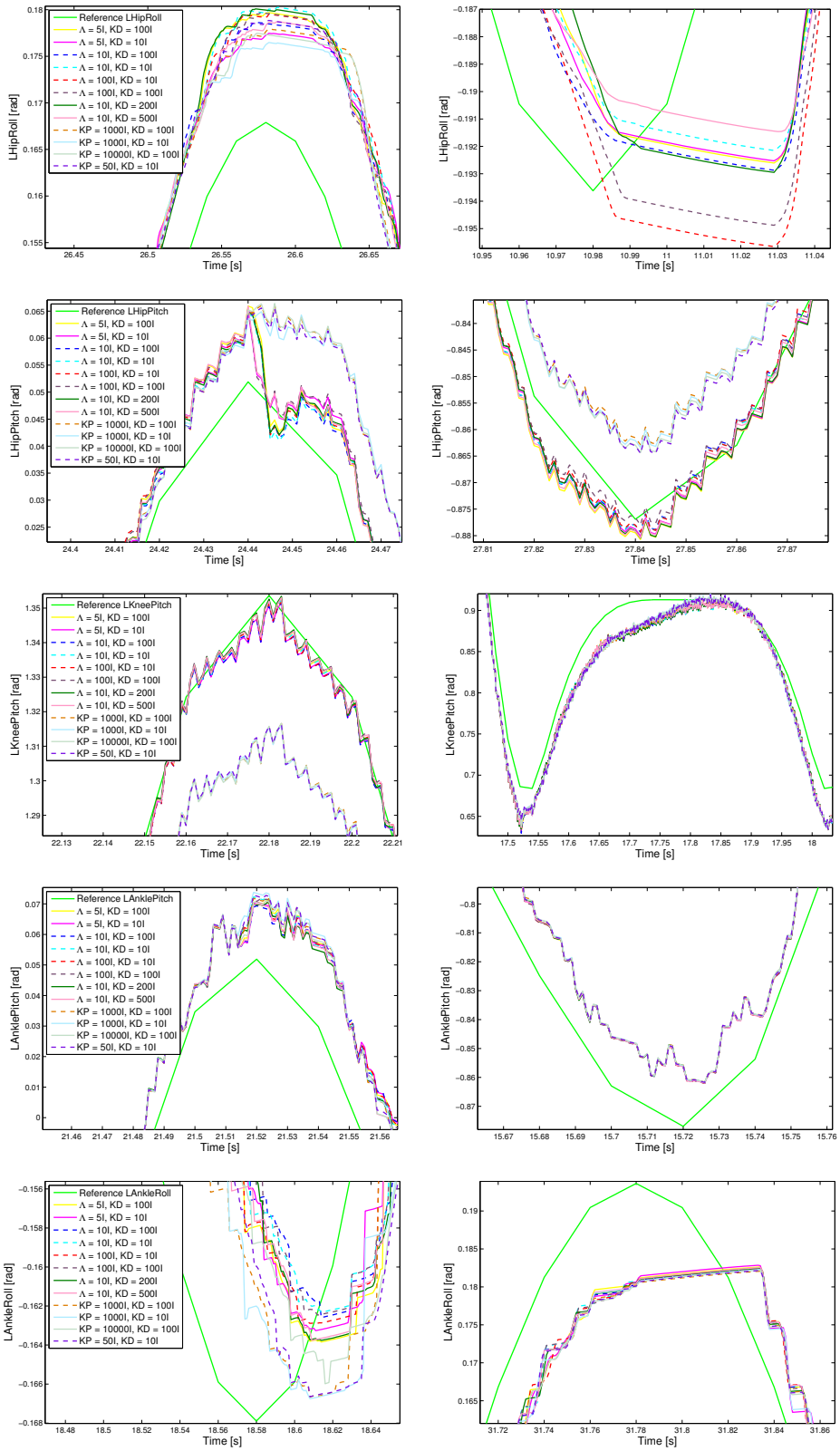


Figure 5.13: Zoomed-in view of the trajectory tracking of the left leg for multiple sets of parameters of the adaptive controller and PD controller.

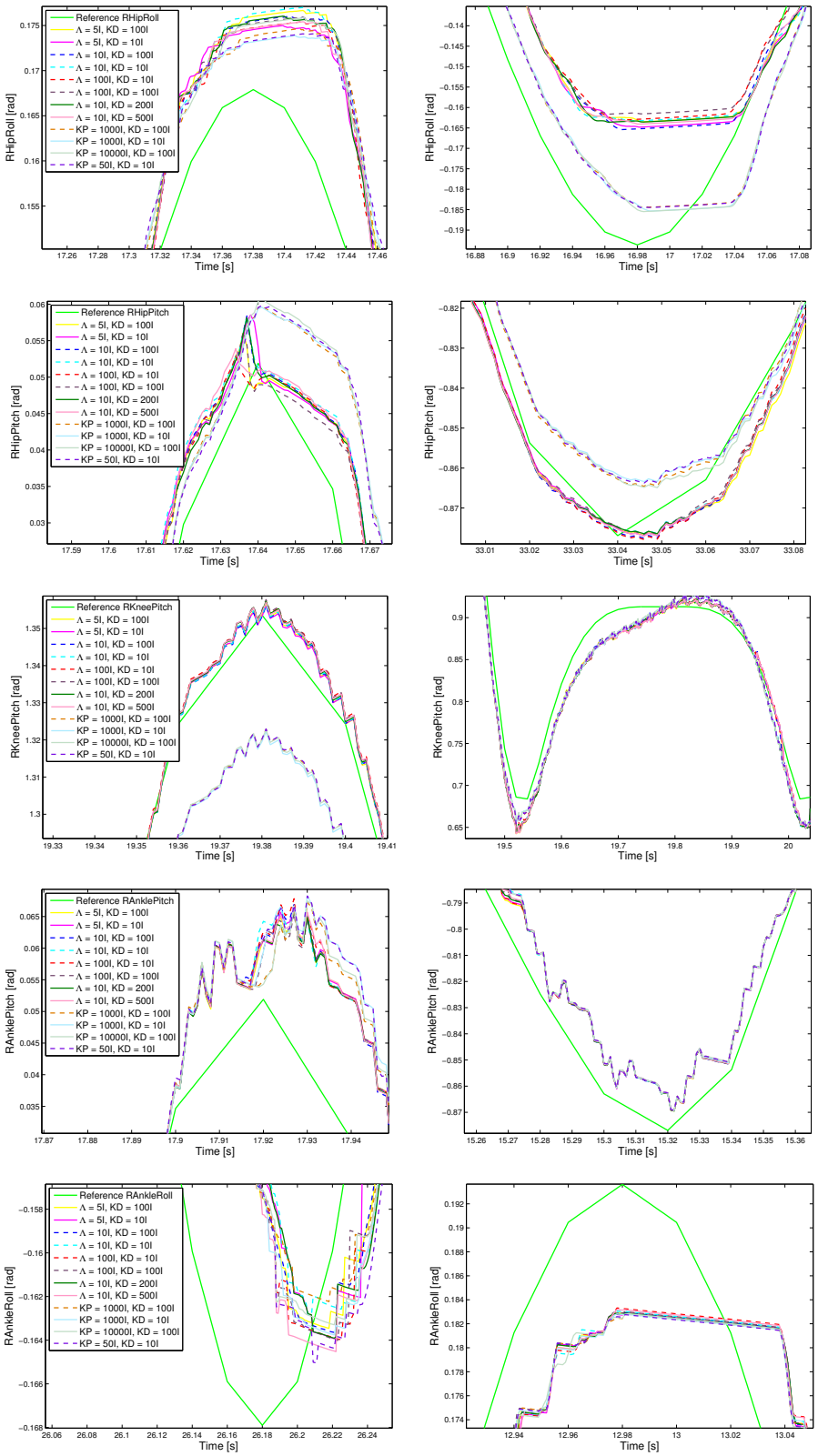


Figure 5.14: Zoomed-in view of the trajectory tracking of the right leg for multiple sets of parameters of the adaptive controller and PD controller.

5.5 Chapter Summary

In this chapter, the adaptive control method applied to the swing-leg of the humanoid robot for fast and precise trajectory tracking and dynamic parameter estimate is presented. Fast walking of the robot can be realized. First, the planar model of the swing-leg is establised. Then by adding two extra joints in the frontal plane, 3D model of the swing-leg is set up. Simulation based on both Matlab/SimMechanics and ODE$^{(1)}$ physics engine was carried out.

The effectiveness of the adaptive control in handling the dynamic demands of trajectory tracking is compared with the PD control. The advantage of the adaptive control in precise dynamic trajectory tracking over the PD control is well confirmed on the planar model and 3D model of the swing-leg by using the simulator based on Matlab/SimMechanics. However, the superiority is not obvious when the adaptive control and PD control are applied to the 3D model of the swing-leg in the ODE$^{(1)}$ simulator. The reason is that the robot in the simulator is not a torque-control robot in a real sense, only very small control torques can be input, or else the robot will collapse; the adaptive control needs nevertheless large torques to achieve precise trajectory tracking. Therefore, the limitation on the magnitude of the control torque with the ODE$^{(1)}$ simulator reduces the accuracy of the position control.

6 Landing Impact Control

6.1 Introduction

Switching from free space motion to constrained motion will experience transient impact force caused by transfer of body weight in an extremely short time. Human beings reduce the full intensity of the impact by shock-absorbing reactions of the muscles at the ankle, knee and hip [93]. The heel contact plays an essential role in reducing the force of limb loading.

As the humanoid NAO has no heels, the soles are always parallel to the ground during walking. As a result, the whole sole of the foot will abruptly land on the ground at the beginning of the double support phase. This will bring about a large impact force which may cause oscillation and rebounding of the landing foot, and consequently the walking of the humanoid robot might become unstable. A control method to decrease the landing force and increase the stability of the walking system is therefore required. Figure 6.1 illustrates the force distribution of the robot NAO in the process of a walk. In this thesis, impedance control is applied to moderating the landing impact force.

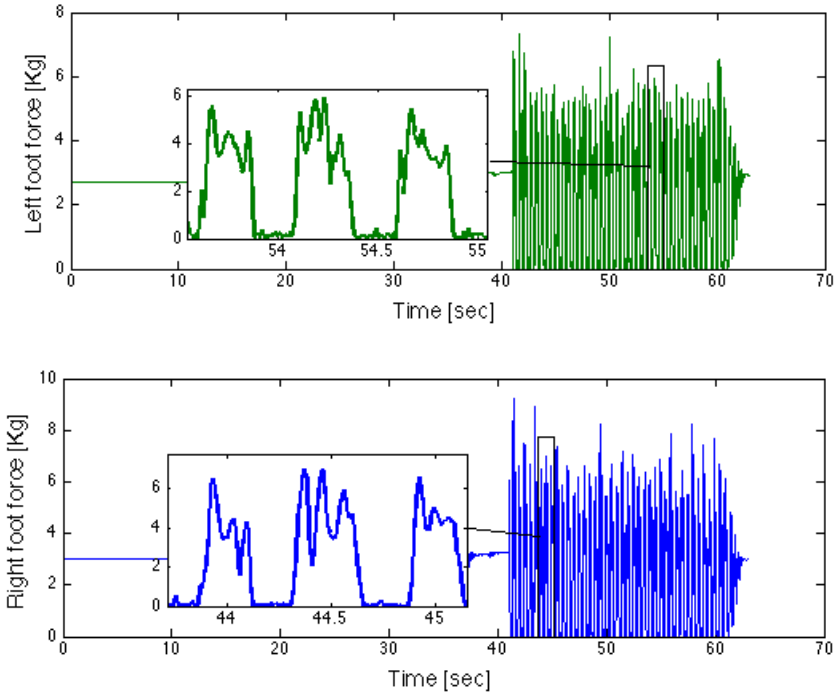


Figure 6.1: Force distribution during walking of the robot NAO.

6.2 Related Works

Yamaguchi used passive shock absorbing mechanism with shock absorbing material against the impacts [128]. Kim *et al.* modeled the robot's foot and the ground as an one-port network system and alleviated the landing force by modifying the desired height of the landing foot according to the calculated energy of the network [61]. The method does not need knowledge of robot dynamics. Kajita *et al.* controlled the impact force by lifting the landing foot at a certain speed during the landing phase [47]. Besides, some biped walking robots and biped humanoids have applied impedance control to reducing the landing impact force. Lim *et al.* employed impedance control with large damping during the landing phase of a humanoid robot [70]. Sugahara *et al.* reduced the landing impact of a biped locomotor by using variable compliance parameters which depend on the walking phase [110]. Löffler also used an impedance-control-based method to absorb the landing impact [73].

6.3 Force Detection

The contact force can be sensed typically by force sensors that are mounted on the sole of the foot. As shown in Figure 6.2, each foot of NAO is equipped with four Force Sensitive Resistors (FSRs) on the sole which measure a resistance change according to the pressure applied. The working range of each FSR is from 0 N to 25 N. Normally, a filter, e.g. low-pass filter, is used to get reliable force value because the measured value from FSR sensors have high-frequency noises.

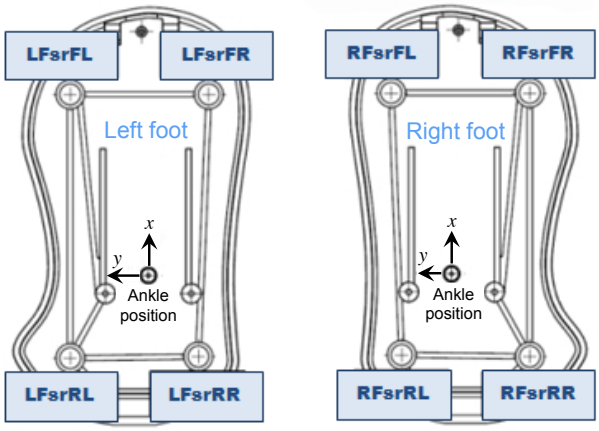


Figure 6.2: FSR locations on the foot.

6.4 Model of Dynamics

There are many methods for generating the dynamic equations of a mechanical system. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation or analysis.

In this section, we give a global characterization of the dynamics of a rigid body subject to external forces and torques. First, the well-known form of the Newton–Euler equations based on body linear momentum and angular momentum laws is used to describe the dynamics of a

rigid body. Then Lagrangian equations, which rely on the energy properties of a mechanical system are used to describe the dynamics of the system.

Newton-Euler Equations (When the Robot is Viewed as a Rigid Body)

Computation of the motion of a rigid body consists of solving a set of ODEs$^{(2)}$. The most common way to describe the motion of a rigid body is by means of the Newton-Euler equations, which define the time derivatives of the linear momentum, $\boldsymbol{P}$, and angular momentum, $\boldsymbol{L}$, as a function of external force, $\boldsymbol{F} = [f_x \quad f_y \quad f_z]^T$, and torque (principal moment), $\boldsymbol{T} = [T_1 \quad T_2 \quad T_3]^T$:

$$\begin{aligned} \boldsymbol{F}(t) &= \dot{\boldsymbol{P}}(t) = m\ddot{\boldsymbol{r}}(t) \\ \boldsymbol{T}(t) &= \dot{\boldsymbol{L}}(t) = \mathcal{I} \cdot \dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}(t) \times (\mathcal{I} \cdot \boldsymbol{\omega}(t)) \end{aligned} \tag{6.1}$$

where m is the mass of the body, $\ddot{\boldsymbol{r}} = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]^T$ represents the linear acceleration of the CoM, $\boldsymbol{\omega} = [\omega_1 \quad \omega_2 \quad \omega_3]^T$ denotes the angular velocity about the principal axes, and $\mathcal{I}$ is the inertia tensor of the body taking the principal axes as the body axes and it has the diagonal form

$$\mathcal{I} = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

The scalar form of a body dynamics equations can be written as

$$\begin{cases} m\ddot{x} = f_x \\ m\ddot{y} = f_y \\ m\ddot{z} = f_z \end{cases} \tag{6.2}$$

and

$$\begin{cases} I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 = T_1 \\ I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 = T_2 \\ I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 = T_3 \end{cases} \tag{6.3}$$

Lagrange's Equations (When Viewed as an Open-chain Robot)

We rely on a Lagrangian derivation of the dynamics. This technique has the advantage of requiring only the kinetic and potential energies of the system to be computed. In Appendix C, we derive the dynamic equations of motion of a system without constraints (the swing-leg).

The dynamic equations of motion for the constrained system using Lagrange's equations can be written as

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + J(\theta)^T f \tag{6.4}$$

with $\theta \in \mathbb{R}^n$ being the joint angles for the open-chain robot, $H(\theta) \in \mathbb{R}^{n \times n}$ denoting the symmetric inertia matrix, $C(\theta, \dot{\theta})\dot{\theta}$ representing the Coriolis and centrifugal force terms, $g(\theta) \in \mathbb{R}^n$ being the gravity term, $\tau \in \mathbb{R}^n$ denoting the applied joint torques, and f being the constraint forces and moments. The effect of the external force and moment is given by $J(\theta)^T f$, $J(\theta)$ represents the Jacobian matrix of the robot. The calculation of Jacobian matrix for force control is presented in Appendix D.

Full Nonlinear Dynamic Equations of Motion

Equations of motion can be formalized in a number of different coordinate systems. m independent coordinates are necessary to describe the motion of a system having m DoFs. Any set of m independent coordinates is called generalized coordinates (e.g. q_1, q_2, . . . q_m).

Combining Eq. (6.1) and Eq. (6.4), we get the full nonlinear dynamic equations of motion for the constrained system

$$M(q)\ddot{q} + N(q,\dot{q}) = S\tau + J^T(q)\boldsymbol{F} \tag{6.5}$$

where $q \in \mathbb{R}^{n+6}$ is the set of generalized coordinates, and it is the configuration vector that includes n joint angles and six absolute position and orientation of the robot. $M(q) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the inertia matrix, $N(q,\dot{q}) \in \mathbb{R}^{n+6}$ is a vector of nonlinear gravitational, centrifugal and Coriolis terms. $S \in \mathbb{R}^{(n+6)\times n}$ maps the n joint torques τ to the appropriate rows of Eq.(6.5); generally, these equations are written such that $S = [0, I]^T$. $\boldsymbol{F}$ represents the generalized forces including the external forces and moments.

Equations of Motion in the Cartesian Space

Taking Eq.(6.4), we have

$$H(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + \boldsymbol{g}(\theta) = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{6.6}$$

where $\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ is the joint input torque vector and $\boldsymbol{\tau}_e \in \mathbb{R}^{n\times 1}$ denotes the generalized vector of joint torques exerted by the environment on the end-effector.

According to the definition of Jacobian matrix in robotics, the relationship between translational and rotational velocities of the end-effector in Cartesian space and joint velocities in the joint space can be described with a linear equation:

$$\dot{x} = J(\theta)\dot{\theta} \tag{6.7}$$

Then

$$\dot{\theta} = J(\theta)^{-1}\dot{x} \tag{6.8}$$

Differentiating Eq. (6.7) results in

$$\ddot{x} = \dot{J}(\theta)\dot{\theta} + J(\theta)\ddot{\theta} \tag{6.9}$$

Then we get

$$\ddot{\theta} = J(\theta)^{-1}\ddot{x} - J(\theta)^{-1}\dot{J}(\theta)\dot{\theta} \tag{6.10}$$

Taking Eq. (6.8) and Eq. (6.10) into Eq.(6.6), it is possible to derive the dynamics equations of motion in the Cartesian space:

$$H_x(x)\ddot{x} + C_x(x,\dot{x})\dot{x} + \boldsymbol{g}_x(x) + h(x,\dot{x}) = \boldsymbol{f} + \boldsymbol{f}_e \tag{6.11}$$

with

$$\begin{aligned} H_x(x) &= J(\theta)^{-T}H(\theta)J(\theta)^{-1} \\ C_x(x,\dot{x}) &= J(\theta)^{-T}C(\theta,\dot{\theta})J(\theta)^{-1} \\ \boldsymbol{g}_x(x) &= J(\theta)^{-T}\boldsymbol{g}(\theta) \\ h(x,\dot{x}) &= -H_x(x)\dot{J}(\theta)J(\theta)^{-1}\dot{x} \end{aligned} \tag{6.12}$$

where $\boldsymbol{f}$ is a vector of applied control force, $\boldsymbol{f} = J(\theta)^{-T}\boldsymbol{\tau}$; $\boldsymbol{f}_e = J(\theta)^{-T}\boldsymbol{\tau}_e$ represents the generalized vector of forces exerted by the ground on the foot.

6.5 Impedance Control

Let a second-order impedance model be given by

$$f_e = M(\ddot{x} - \ddot{x}_d) + B(\dot{x} - \dot{x}_d) + K(x - x_d) \tag{6.13}$$

where x_d, $\dot{x}_d$ and $\ddot{x}_d$ are the desired position, velocity and acceleration of the end-effector, respectively. M, B and K are diagonal positive definite matrices representing the virtual inertia, damping and stiffness of the system.

The reference acceleration of the end-effector is then given by

$$\ddot{x} = \ddot{x}_d - M^{-1}\Big[B(\dot{x} - \dot{x}_d) + K(x - x_d) - f_e\Big] \tag{6.14}$$

Taking Eq.(6.14) into Eq.(6.11) yields

$$\begin{aligned}\tau = J(\theta)^T H_x(x)\Big[\ddot{x}_d - M^{-1}\Big[B(\dot{x} - \dot{x}_d) + K(x - x_d) - f_e\Big]\Big] \\ + J(\theta)^T C_x(x, \dot{x})\dot{x} + J(\theta)^T \boldsymbol{g}_x(x) + J(\theta)^T h(x, \dot{x}) - J(\theta)^T f_e\end{aligned} \tag{6.15}$$

Taking Eq. (6.12) into Eq. (6.15), the impedance control law is obtained,

$$\begin{aligned}\tau = J(\theta)^T H_x(x)\Big[\ddot{x}_d - M^{-1}\Big[B(\dot{x} - \dot{x}_d) + K(x - x_d) - f_e\Big]\Big] \\ + C(\theta, \dot{\theta})\dot{\theta} + \boldsymbol{g}(\theta) - H(\theta)J(\theta)^{-1}\dot{J}(\theta)\dot{\theta} - J(\theta)^T f_e\end{aligned} \tag{6.16}$$

The terms that compensate for velocity dependent forces and gravity $(C(\theta, \dot{\theta})\dot{\theta} + \boldsymbol{g}(\theta) - H(\theta)J(\theta)^{-1}\dot{J}(\theta)\dot{\theta})$ can be considered as feedforward terms.

The foot of the robot is desired to achieve stable landing on the ground without bounding and oscillatory behavior, so $\ddot{x}_d = 0$. Eq.(6.16) can be simplified by assuming that $H_x(x) = M$. Moreover, it is possible to consider the impedance control law in the quasi-static condition which means that the velocity $\dot{x} \approx 0$, and thus $\dot{\theta} \approx 0$. The approximation is reasonable since force control tasks are generally executed at a low speed. Then Eq.(6.16) can be further simplified by neglecting the terms containing $\dot{\theta}$[1]. Hence, the following simplified impedance control law is obtained:

$$\tau = J(\theta)^T\Big(B(\dot{x}_d - \dot{x}) + K(x_d - x)\Big) + g(\theta) \tag{6.17}$$

This control scheme is equivalent to a PD control law with gravity compensation. It indicates that the force between the robot and the environment is controlled via regulating the position (x) and velocity $(\dot{x})$ of the robot foot in the Cartesian space. When in contact with a stiff environment, the position of the environment can be set as the origin $x = 0$; also,

[1]The velocity $\dot{x}$ in the Cartesian space in the control law cannot be neglected since $\dot{x}$ is proportional to the sum of the individual joint angular velocities, as well as to the square of the sum of the individual joint angular velocities. Namely, $\dot{x} \propto \Big[\sum_{i=1}^{n}\dot{\theta}_i + (\sum_{i=1}^{n}\dot{\theta}_i)^2\Big]$, n is the number of the joints of the leg. The related derivation can be found in Appendix C.

the desired velocity is usually zero $\dot{x}_d = 0$. The equilibrium state and the global asymptotic stability of the system with the control law formulated in Eq.(6.17) are discussed in [6].

Once the stable landing is achieved, the robot should be able to keep walking according to the originally planned walking pattern. Kim *et al.* recovered the CoG motion to follow the original trajectory by means of PD control [61]. Lim *et al.* adopted polynomial interpolation in the subsequent stance phase in order to recover the preset walking pattern which was changed by the impedance control in the landing phase [70].

6.6 Chapter Summary

This chapter provided a control method with the aim of decreasing the landing impact force by means of compliant force control approach. The control algorithm is given in the quasi-static condition. A specific representation of the model's dynamic components will permit a better understanding of the interaction between the robot and the environment.

7 Balance Maintenance

Balance maintenance plays an important role with regard to safe and successful coexistence of humanoid robots with humans. This chapter presents balance recovery strategies in the presence of external perturbations in order for the robot to maintain an upright posture. Simple dynamic models are described that can generally represent humanoid balance. The dynamics are analyzed to determine recovery strategies and predict their stability.

The strategies for restoring balance of a humanoid robot presented in this chapter are as follows:

- Ankle strategy
- Hip strategy
- Combined strategy

7.1 Introduction

From biological point of view, *balance* is a generic term describing the dynamics of body posture to prevent falling. It is related to the inertial forces acting on the body and the inertial characteristics of body segments [125].

7.1.1 Clues from Human Locomotion

Biomechanical studies on human balance [31, 75, 77, 125] have revealed that humans use a variety of control strategies in order to maintain balance in the presence of sudden horizontal perturbations. The most appropriate strategy is selected according to the size of the postural challenge as well as the current feedback information provided by the sensory systems of the human body. Figure 7.1 illustrates four basic strategies that human beings apply to compensate for external perturbations. For smaller perturbations, torque about the ankle joints is used to restore equilibrium by moving the body CoM forward and backward, and the posture of the rest body remains unchanged. This is termed "ankle strategy". When the perturbations are larger or when the support surface is narrow, the use of the hip flexors or extensors is involved to generate shear forces at the feet that act to decelerate the CoM, this is so called "hip strategy". A combination of pure ankle and hip strategy is also possible which is so called "combined strategy". The last resort-strategy is elicited when the CoM of the body passes outside the limits of base of support (BoS), in this case human beings will step out in order to reconfigure the BoS, which is called "stepping strategy".

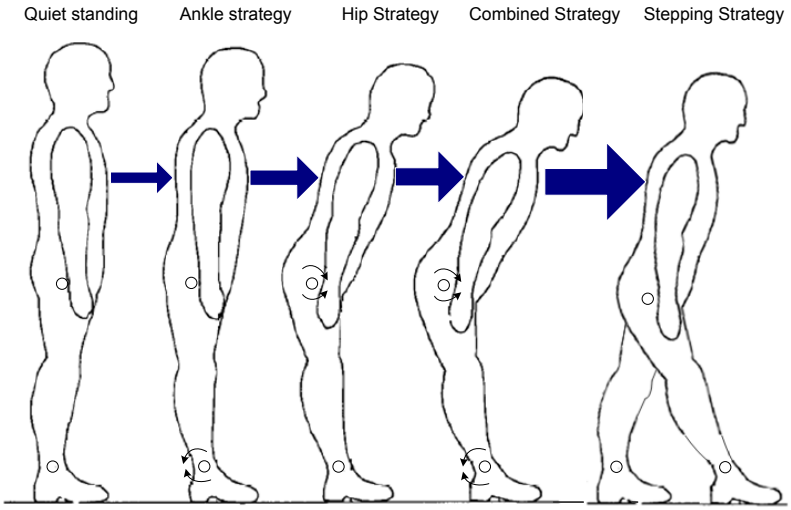


Figure 7.1: Four basic strategies against external perturbations from [125].

7.1.2 Related Works

Many researchers have studied biped balance control. Unification of models and strategies was used for analysis of humanoid balance [105, 111]. Kuo used a Linear Quadratic Gaussian (LQG) controller to model human central nervous system behavior of selecting ankle and/or hip strategy [65]. Taking linear and angular momentum into account, Macchietto *et al.* proposed an optimization method for balance recovery which controlled and tracked the trajectories of the CoM and CoP simultaneously [74]. Hofmann in his PhD thesis studied standing balance with regard to the spin angular momentum [28]. Azevedo *et al.* realized balance control for a seven-actuated-joint planar biped in simulation by solving quadratic optimization problem with balance objective and constraints [4]. Stephens mimicked the ankle and hip strategies by regulating the CoP position and meanwhile keeping the robot posture upright using an integral controller [107]. Furthermore, ankle and hip strategies were realized on real humanoid robots by meas of either a unified dynamic model [37] or seperate dynamic models [81]. Pratt *et al.* added a flywheel to the LIPM for analyzing hip strategy and defining stable footstep locations of a biped system, known as Capture Region [96]. Stepping strategy was achieved on humanoid robots by stabilizing the GCoM within the support polygon, and this was realized through planning the trajectories of the CoM and the swing foot solving the QP problem and then tracking the trajectories [109], or through directly controlling the CoM using force control, meanwhile online planning and tracking the swing foot trajectory [36].

7.2 State Estimation

A good estimate of the CoM position and velocity is important for choosing strategies. As we have discussed in Chapter 4, if the state estimator has the following form

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w \\ y_k = Cx_k + v \end{cases}$$

where w and v are vectors representing the standard process and measurement noise variables respectively which feature Gaussian processes with covariance given by Q and R,

$$w \sim N(0, Q)$$
$$v \sim N(0, R)$$

then it is possible to implement Kalman filter and obtain the estimated state of the CoM.

We choose 3D-LIPM which is based on the CoM dynamics by assuming the height of the CoM is constant and the angular momentum around the CoM is zero.

$$\begin{cases} \ddot{x} = \omega^2(x - x_{zmp}) \\ \ddot{y} = \omega^2(y - y_{zmp}) \end{cases} \tag{7.1}$$

where $\omega = \sqrt{\frac{g}{z_0}}$, z_0 is the constant height of the CoM, (x, y) is the position of the CoM, and (x_{zmp}, y_{zmp}) is the position of the ZMP.

Let $x_k = [x \quad \dot{x} \quad x_{zmp} \quad y \quad \dot{y} \quad y_{zmp}]^T$ be the state vector and $u_k = [\dot{x}_{zmp} \quad \dot{y}_{zmp}]^T$ be the input at time step k. Eq.(7.1) can be written as a discrete linear system in state space form,

$$x_{k+1} = Ax_k + Bu_k \tag{7.2}$$

The matrices in Eq.(7.2) are given by

$$A = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ \omega^2 T & 1 & -\omega^2 T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & \omega^2 T & 1 & -\omega^2 T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

and

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & T \end{bmatrix} \tag{7.4}$$

where T is the time step.

Assuming a linear measurement model

$$y_k = Cx_k$$

where y_k is a vector of measurements. The structure of C depends on the type of sensor modalities and state description.

For example, if

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ m\omega^2 & 0 & -m\omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & m\omega^2 & 0 & -m\omega^2 \end{bmatrix}$$

then $y_k = [x \quad x_{zmp} \quad f_x \quad y \quad y_{zmp} \quad f_y]^T$, where $f = [f_x \quad f_y]$ is the total horizontal force on the CoM.

Therefore, Kalman filter as shown in Figure 7.2 can be implemented and the state of the CoM can be estimated.

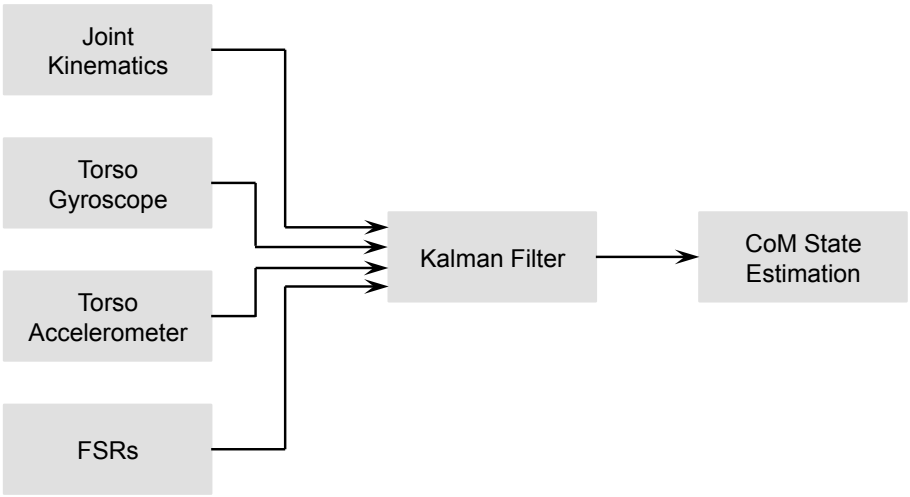


Figure 7.2: CoM state estimation using Kalman filter.

7.3 Ankle Strategy

Ankle strategy is a reaction to small perturbations using torque exerted by the stance ankle to generate a force on the CoM. Since the GRF passes through the CoM, the moment about the CoM is zero. From the robotics point of view, the ankle strategy turns the body into an inverted pendulum, balanced upright using ankle torque [3].

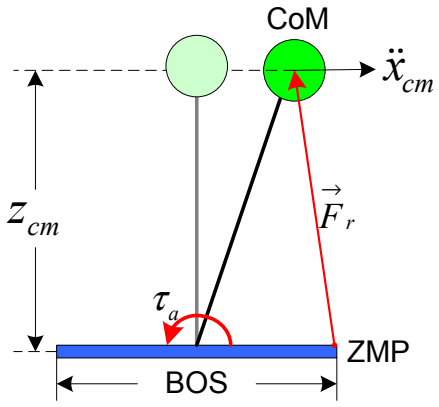


Figure 7.3: Schematic diagram of ankle strategy.

7.3.1 Open Loop

Applying the torque equilibrium condition to the model shown in Figure 7.3, we have

$$\tau_a + m\ddot{x}_{cm}z_{cm} - m\ddot{z}_{cm}x_{cm} = mgx_{cm} \tag{7.5}$$

where τ_a is the torque executed on the ankle joint, x_{cm} and z_{cm} are the horizontal and vertical CoM positions, m is the mass of the CoM. If the angle between the stance link and the vertical axis is small, or if movements are relatively slow, then in a small neighborhood area the vertical CoM position can be considered a constant value, namely $z_{cm}= k$ (k is a constant). Hofmann also argues that controlling horizontal movement of the CoM is more important for maintaining balance [28]. Then

$$\ddot{z}_{cm} = 0 \tag{7.6}$$

Eq.(7.5) becomes

$$\tau_a + m\ddot{x}_{cm}z_{cm} = mgx_{cm} \tag{7.7}$$

In humanoids, the location of the ZMP is roughly proportional to the magnitude of the torque at the ankle. It can be formulated as [107]:

$$x_{zmp} = \frac{\tau_a}{F_{normal}} \tag{7.8}$$

with F_{normal} denoting the normal force which cancels the weight and downward acceleration of the CoM, so $F_{normal} = m(\ddot{z}_{cm} + g)$. Therefore,

$$\tau_a = m(\ddot{z}_{cm} + g)x_{zmp} \tag{7.9}$$

Taking Eq.(7.6) into it,

$$\tau_a = mgx_{zmp} \tag{7.10}$$

Substituting Eq.(7.10) into Eq.(7.7) and rearranging, we get

$$\ddot{x}_{cm} - \frac{g}{z_{cm}}x_{cm} + \frac{gx_{zmp}}{z_{cm}} = 0 \tag{7.11}$$

Let $\omega^2 = \frac{g}{z_{cm}}$, $f = -\frac{gx_{zmp}}{z_{cm}}$, Eq.(7.11) becomes $\ddot{x}_{cm} - \omega^2 x_{cm} = f$ with the initial conditions $x(0) = x_0$ and $\dot{x}(0) = \dot{x}_0$. The solution of the equation is

$$x_{cm}(t) = (\frac{f}{\omega^2} + x_0)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{f}{\omega^2} \tag{7.12}$$

Since $\sinh x = \frac{e^x - e^{-x}}{2}$ and $\cosh x = \frac{e^x + e^{-x}}{2}$, if the exerted ankle torque can make the system recover from perturbations, $x_{cm}(t)$ definitely has a convergent value. Therefore, the sufficient and necessary condition is

$$\frac{f}{\omega^2} + x_0 = -\frac{\dot{x}_0}{\omega}$$

Namely

$$x_0 + \frac{\dot{x}_0}{\omega} = x_{zmp} \tag{7.13}$$

The convergent trajectory of $x_{cm}(t)$ is

$$x_{cm}(t) = (\frac{f}{\omega^2} + x_0)e^{-\omega t} - \frac{f}{\omega^2} \tag{7.14}$$

$$\lim_{n\to\infty} x_{cm}(t) = -\frac{f}{\omega^2}$$

The first order derivative of Eq.(7.12) is

$$\frac{\dot{x}_{cm}(t)}{\omega} = (\frac{f}{\omega^2} + x_0)\sinh(\omega t) + \frac{\dot{x}_0}{\omega}\cosh(\omega t) \tag{7.15}$$

Using Eq.(7.12) and Eq.(7.15),

$$x_{cm}(t) + \frac{\dot{x}_{cm}(t)}{\omega} = -\frac{f}{\omega^2}$$

Note that from Eq.(7.13), the boundary situations emerge when the ankle torque is saturated with the ZMP at the edge of the BoS, that is $x_{zmp} = x_{zmp\_max}$ or $x_{zmp} = x_{zmp\_min}$, where x_{zmp_max} and x_{zmp_min} are the front and back borders of the BoS.

If the initial state of the CoM of the humanoid robot is inside the area $x_{zmp\_min} \leq x_0 + \frac{\dot{x}_0}{\omega} \leq x_{zmp\_max}$, the system can potentially recover from the perturbations using ankle strategy. The convergent value of $x_{cm}(t)$ is formulated by Eq.(7.14).

Otherwise, if $x_0 + \frac{\dot{x}_0}{\omega} > x_{zmp\_max}$ or $x_0 + \frac{\dot{x}_0}{\omega} < x_{zmp\_min}$, the ankle torque alone cannot restore balance and either a different balance strategy is needed or a step should be initiated to prevent falling. Under the circumstances, the values of $x_{cm}(t)$ are divergent.

Algorithm 1 shows calculation of potential recovery trajectories of the CoM given different initial conditions using ankle strategy. Figure 7.4 demonstrates the calculated trajectories. Two bold green lines indicate the critical boundaries of using ankle strategy for restoring equilibrium in the presence of perturbations. The region between these two lines is the potentially stable region. All trajectories that start within the region are potentially stable and all that start outside the region will fall.

7.3.2 Closed Loop

Now we consider that the CoM is PD controlled. The control law is

$$\begin{aligned} \ddot{x}_{cm} &= k_p(x_{set} - x_{cm}) + k_d(\dot{x}_{set} - \dot{x}_{cm}) \\ &= k_p(x_{set} - x_{cm}) - k_d\dot{x}_{cm} \end{aligned} \tag{7.16}$$

where k_p and k_d are the proportional gain and derivative gain, respectively.

Rearranging Eq.(7.16), we get

$$\ddot{x}_{cm} + k_d\dot{x}_{cm} + k_p x_{cm} = k_p x_{set}$$

Solutions of the second-order non-homogeneous linear differential equation with constant coefficients depend on the values of k_p and k_d.

Algorithm 1 Calculate $x_{cm}(t)$ using ankle strategy given initial conditions

Let $f_{max} = -gx_{zmp\_max}/z_{cm}$ and $f_{min} = -gx_{zmp\_min}/z_{cm}$
Initial Conditions: x_0, $\dot{x}_0$, $f = -\omega\dot{x}_0 - \omega^2 x_0$
if $f \geq f_{max}$ and $f \leq f_{min}$ **then**
    $x_{cm}(t) = (\frac{f}{\omega^2} + x_0)e^{-\omega t} - \frac{f}{\omega^2}$
end if
if $f \geq f_{min}$ **then**
    $f \Leftarrow f_{min}$
    $x_{cm}(t) = (\frac{f_{min}}{\omega^2} + x_0)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{f_{min}}{\omega^2}$
    $\lim_{t\to\infty}\left[x_{cm}(t) - \frac{\dot{x}_{cm}(t)}{\omega}\right] = -\frac{f_{min}}{\omega^2}$
end if
if $f \leq f_{max}$ **then**
    $f \Leftarrow f_{max}$
    $x_{cm}(t) = (\frac{f_{max}}{\omega^2} + x_0)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{f_{max}}{\omega^2}$
    $\lim_{t\to\infty}\left[x_{cm}(t) - \frac{\dot{x}_{cm}(t)}{\omega}\right] = -\frac{f_{max}}{\omega^2}$
end if

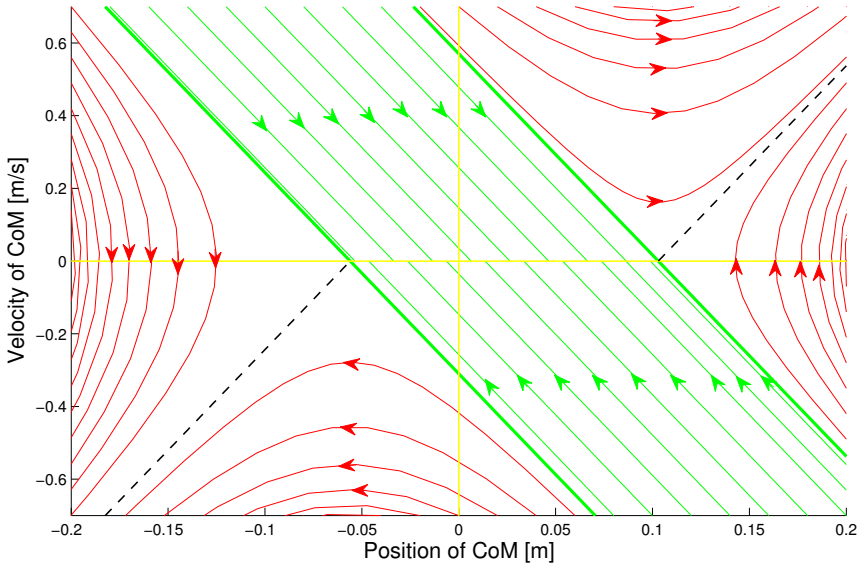


Figure 7.4: Potentially stable region (green) and unstable region (red) given by different initial conditions. Parameters are g =9.81 m/s^2, z_{cm} = 0.32 m, x_{zmp_max} = 0.103 m, x_{zmp_min} = −0.056 m.

Next, we focus on the stability of the system by analyzing the controllability and convergency of the system. Assuming that $\ddot{z}_{cm} = 0$, the ZMP equation is as follows:

$$x_{zmp} = x_{cm} - \frac{\ddot{x}_{cm} z_{cm}}{g} \tag{7.17}$$

Taking Eq.(7.16) into Eq.(7.17)

$$\begin{aligned} x_{zmp} &= x_{cm} - [k_p(x_{set} - x_{cm}) - k_d\dot{x}_{cm}]\frac{z_{cm}}{g} \\ &= (1 + \frac{k_p z_{cm}}{g})x_{cm} + \frac{k_d z_{cm}}{g}\dot{x}_{cm} - \frac{k_p z_{cm}}{g}x_{set} \end{aligned} \tag{7.18}$$

The ZMP constraint is

$$x_{zmp\_min} \leq x_{zmp} \leq x_{zmp\_max} \tag{7.19}$$

Taking Eq.(7.18) into Eq.(7.19) and rearrange,

$$x_{zmp\_min} + \frac{k_p z_{cm}}{g}x_{set} \leq (1 + \frac{k_p z_{cm}}{g})x_{cm} + \frac{k_d z_{cm}}{g}\dot{x}_{cm} \leq x_{zmp\_max} + \frac{k_p z_{cm}}{g}x_{set} \tag{7.20}$$

Let $k_1 = \frac{k_d z_{cm}}{g}$, $k_2 = 1 + \frac{k_p z_{cm}}{g}$
Then Eq.(7.20) becomes

$$x_{zmp\_min} + (k_2 - 1)x_{set} \leq k_1\dot{x}_{cm} + k_2 x_{cm} \leq x_{zmp\_max} + (k_2 - 1)x_{set}$$

Now we have four straight lines representing the critical boundaries of the open-loop control (l_1, l_2) and close-loop control (a, b) of the CoM using ankle strategy for balance recovery. The intersection points among the lines can be calculated (see Figure 7.5).

$$\begin{aligned} l_1 &: \dot{x}_{cm} = -\omega x_{cm} + \omega x_{zmp\_max} \\ l_2 &: \dot{x}_{cm} = -\omega x_{cm} + \omega x_{zmp\_min} \\ a &: \dot{x}_{cm} = -\frac{k_2}{k_1}x_{cm} + \frac{x_{zmp\_max} + (k_2 - 1)x_{set}}{k_1} \\ b &: \dot{x}_{cm} = -\frac{k_2}{k_1}x_{cm} + \frac{x_{zmp\_min} + (k_2 - 1)x_{set}}{k_1} \end{aligned}$$

As illustrated in Figure 7.6, if the state variables $(x_{cm}, \dot{x}_{cm})$ are inside the boundary region between a and b, $(x_{cm}, \dot{x}_{cm})$ are potentially controllable, and how they are convergent to the set point depends on the proportional gain k_p and derivative gain k_d because the values of k_p and k_d decide the root distribution of the characteristic equation; or else, the state variables are uncontrollable. When the CoM state $(x_{cm}, \dot{x}_{cm})$ is in the region surrounded by the lines l_1, l_2 and $(x_{cm}, \dot{x}_{cm})$ trajectories which pass the point A and point D, $(x_{cm}, \dot{x}_{cm})$ either stably converges to the reference or rests in the potentially stable region. Figure 7.7 displays the ZMP trajectories and corresponding required ankle torques by PD controlling the CoM given different initial conditions using ankle strategy during balance recovery. The former two are potentially stable cases with convergent trajectories, and the latter two are unstable because the ZMP trajectories and ankle torques appear a trend of divergence.

7.4 Hip Strategy

Pure hip strategy involves the application of hip torque to generate horizontal ground forces which keep the CoM over the foot. The ankle in pure hip strategy is unactuated. Note that in this case the GRF doesn't pass through the CoM any more, so there is a moment about

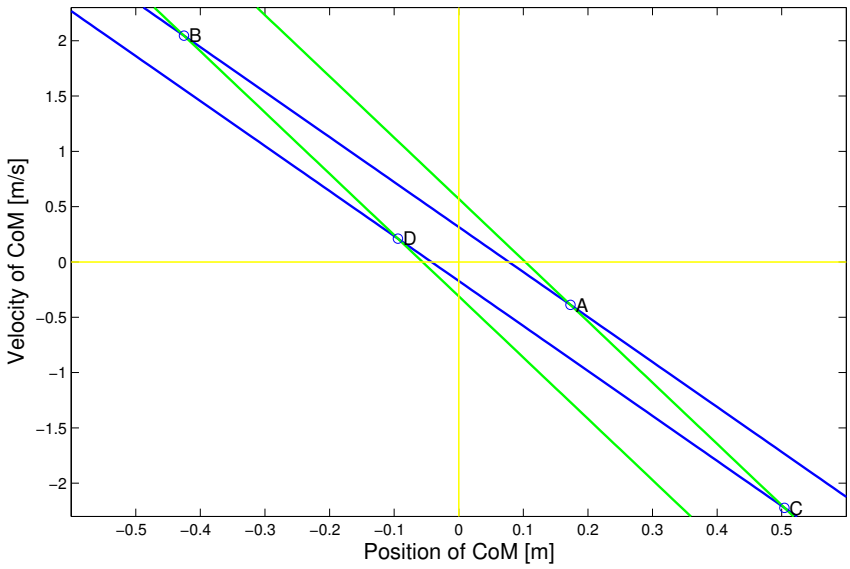


Figure 7.5: Four straight lines a, b, l_1, l_2. The green lines are l_1 and l_2, the blue lines are a and b. A, B, C and D are the four intersection points between l_1, l_2 and a, b. Parameters are g =9.81 m/s^2, z_{cm} = 0.32 m, x_{zmp_max} = 0.103 m, x_{zmp_min} = −0.056 m, k_p = 10 and k_d = 10.

the CoM. A useful quantity for representing this situation is the CMP (refer to Section 2.1). Using hip strategy, the angular stability will be temporarily sacrificed in order to gain greater horizontal force on the CoM. Figure 7.8 exhibits the model of hip strategy.

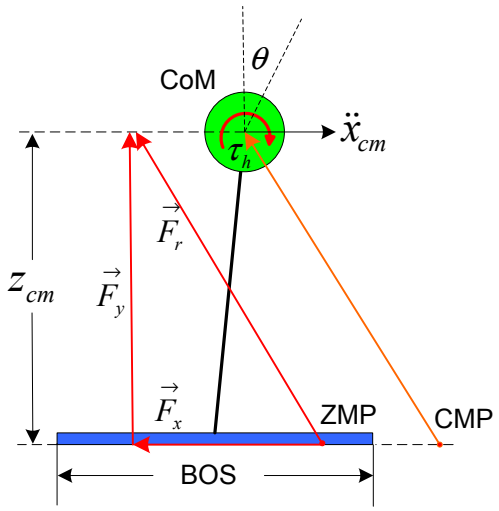
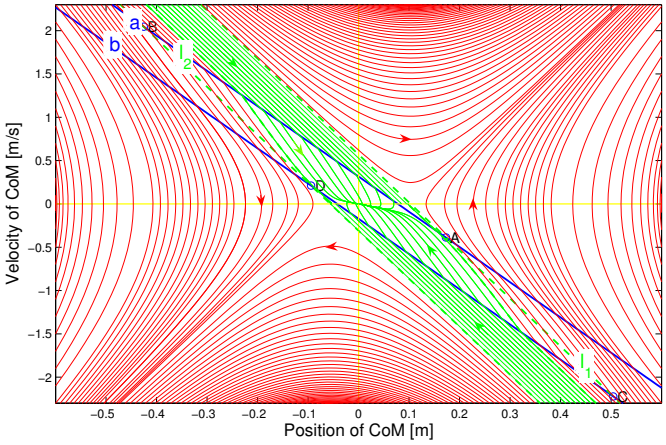


Figure 7.8: Schematic diagram of hip strategy.

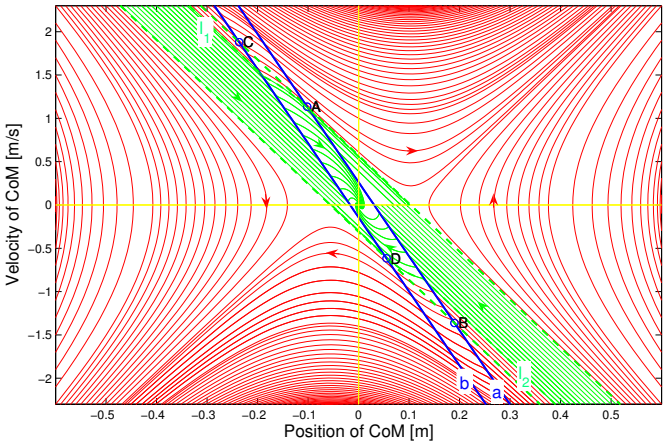
Applying the torque equilibrium condition into this model, we have

$$\tau_h + m\ddot{x}_{cm}z_{cm} - m\ddot{z}_{cm}x_{cm} = mgx_{cm} \tag{7.21}$$

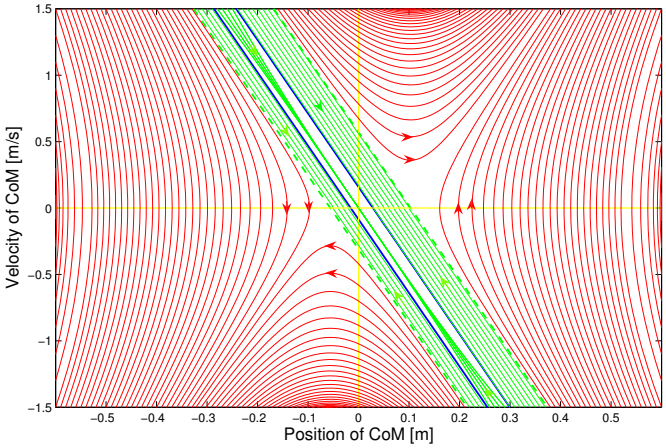
where τ_h is the torque that the hip joint provides, $\tau_h = I\ddot{\theta}$. I is the rotational inertia of the



(a) k_p=10, k_d=10



(b) k_p=72, k_d=12



(c) k_p=80.08, k_d=20

Figure 7.6: Feedback trajectories created by PD controller using different proportional and derivative gains. In (c), no intersection points since lines are parallel.

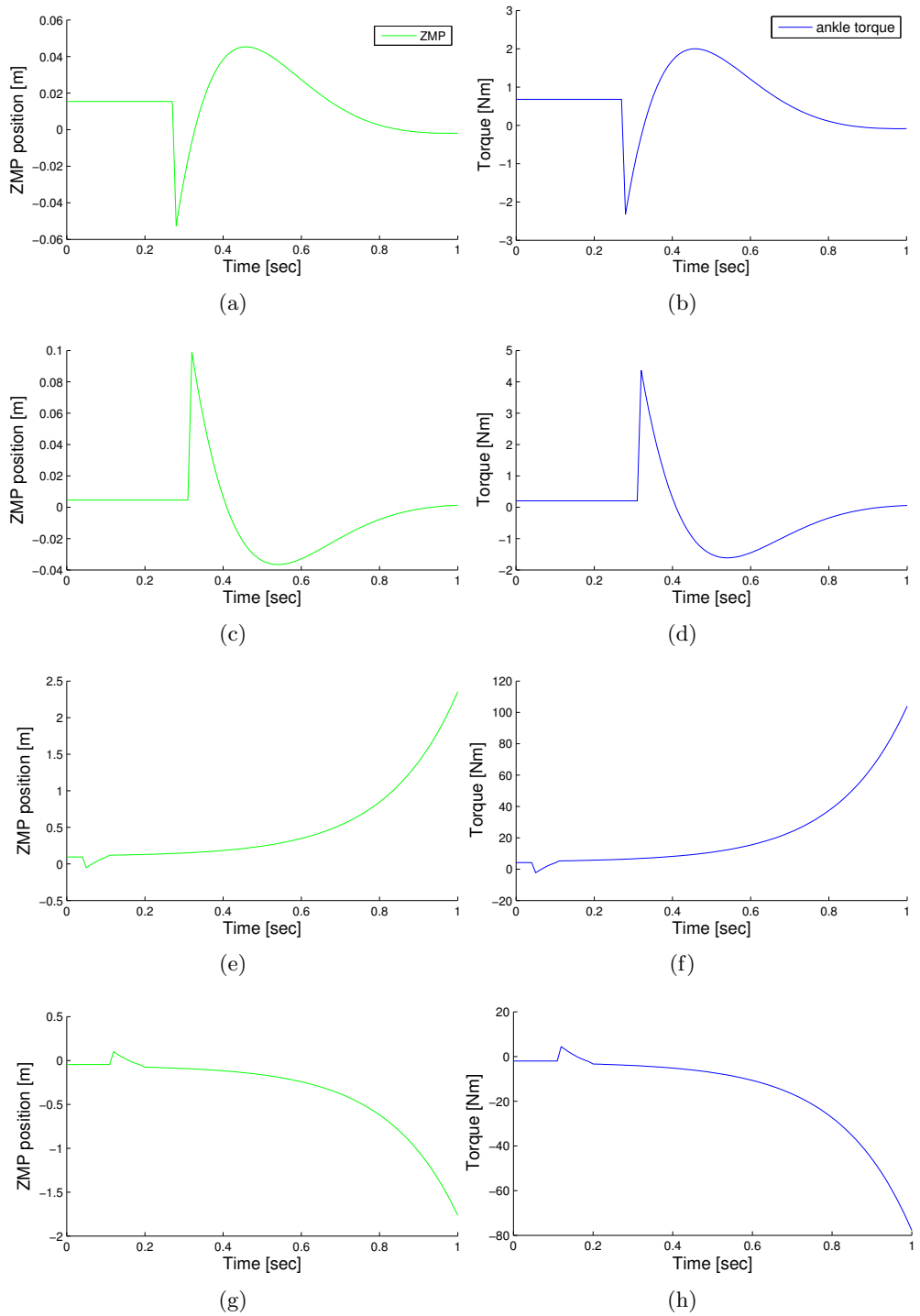


Figure 7.7: ZMP trajectories (green) and required ankle torques (blue) during balance recovery by PD controlling the CoM. $k_p = 72$ and $k_d = 12$. The initial conditions are (a, b) $(x_0, \dot{x}_0) = (-0.4 \text{ m}, 2.3 \text{ m/s})$; (c, d) $(x_0, \dot{x}_0) = (0.42 \text{ m}, -2.3 \text{ m/s})$; (e, f) $(x_0, \dot{x}_0) = (-0.32 \text{ m}, 2.3 \text{ m/s})$ and (g, h) $(x_0, \dot{x}_0) = (0.37 \text{ m}, -2.3 \text{ m/s})$, respectively.

head, arm and trunk (HAT) . $I = m_{HAT} \cdot d^2$, where d is the distance between the CoM of HAT and the hip joint. m is the total mass of the robot. Here, we ignore the vertical movement of CoM, thus $\ddot{z}_{cm}$=0.

Rearranging Eq.(7.21), we get

$$\ddot{x}_{cm} - \frac{g}{z_{cm}} x_{cm} = -\frac{1}{m z_{cm}} \tau_h \tag{7.22}$$

Let $\omega^2 = \frac{g}{z_{cm}}$ and $f = -\frac{1}{m z_{cm}} \tau_h$, then $\ddot{x}_{cm} - \omega^2 x_{cm} = f$ with the initial conditions $x(0) = x_0$ and $\dot{x}(0) = \dot{x}_0$. The solution of Eq.(7.22) is

$$x_{cm}(t) = (\frac{f}{\omega^2} + x_0)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{f}{\omega^2} \tag{7.23}$$

Suppose the robot is moving at $\dot{x}_0$, the upper body is rotating at $\dot{\theta}_0$ and has an angle of θ_0 with respect to vertical. We wish to find a hip torque profile that will bring the robot to rest over its foot with no forward velocity or angular velocity. We use the torque profile [96] proposed by Pratt *et al.*, which provides the most influence on velocity by accelerating the HAT as hard as possible in one direction and then decelerates it, bringing it to a stop at the maximum angle θ_{max}.

$$\tau_h(t) = \tau_n u(t) - 2\tau_n u(t - t_1) + \tau_n u(t - t_2) \tag{7.24}$$

where τ_n is the nominal torque that the hip joint of the robot provides. $u(t - T)$ is the unit step function that satisfies

$$u(t - T) = \begin{cases} 0 & \text{if } t < T \\ 1 & \text{if } t \geq T \end{cases}$$

t_1 is the time at which the HAT stops accelerating and starts decelerating and t_2 is the time at which the HAT comes to a stop.

The controller using the above torque profile is known in control theory as a bang–bang controller, which is a feedback controller that switches abruptly between two states. They are often used to control a plant that accepts a binary input.

Substituting Eq.(7.24) into Eq.(7.23) and rearranging, the trajectory of the CoM is obtained.

$$\begin{aligned} x_{cm}(t) &= x_0\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{\tau_n}{mg}[(\cosh(\omega t) - 1)u(t) - \\ &\quad 2(\cosh(\omega(t - t_1)) - 1)u(t - t_1) + (\cosh(\omega(t - t_2)) - 1)u(t - t_2)] \\ \dot{x}_{cm}(t) &= \omega x_0\sinh(\omega t) + \dot{x}_0\cosh(\omega t) - \frac{\tau_n}{mg}[\omega\sinh(\omega t)u(t) + (\cosh(\omega t) - 1)\delta(t) - \\ &\quad 2\omega\sinh(\omega(t - t_1))u(t - t_1) - 2(\cosh(\omega(t - t_1)) - 1)\delta(t - t_1) + \\ &\quad \omega\sinh(\omega(t - t_2))u(t - t_2) + (\cosh(\omega(t - t_2)) - 1)\delta(t - t_2)] \end{aligned}$$

If the exerted hip strategy can make the system recover from perturbations, $x_{cm}(t)$ converges as $t \to \infty$. The necessary and sufficient condition is

$$x_0 + \frac{\dot{x}_0}{\omega} - \frac{\tau_n}{mg}(1 - 2e^{-\omega t_1} + e^{-\omega t_2}) = 0 \tag{7.25}$$

Since $\tau_h = I\ddot{\theta}$, given the torque profile in Eq.(7.24), the HAT angular velocity and position will be

$$\dot{\theta}(t) = \dot{\theta}_0 + \frac{\tau_n}{I}(t \cdot u(t) - 2(t - t_1) \cdot u(t - t_1) + (t - t_2) \cdot u(t - t_2)) \tag{7.26}$$

$$\theta(t) = \theta_0 + \dot{\theta}_0 t + \frac{\tau_n}{I}(\frac{1}{2}t^2 \cdot u(t) - (t - t_1)^2 \cdot u(t - t_1) + \frac{1}{2}(t - t_2)^2 \cdot u(t - t_2)) \tag{7.27}$$

At time $t = t_2$ we want $\dot{\theta}(t_2) = 0$, solving Eq.(7.26) we get

$$0 = \dot{\theta}_0 + \frac{\tau_n}{I}(t_2 - 2(t_2 - t_1)) = \dot{\theta}_0 + \frac{\tau_n}{I}(2t_1 - t_2)$$

So

$$t_2 = 2t_1 + \frac{I}{\tau_n}\dot{\theta}_0 \tag{7.28}$$

In particular, if $\dot{\theta}_0 = 0$, then $t_2 = 2t_1$.

Substituting Eq.(7.28) into Eq.(7.25) and rearranging, we get

$$t_1 = -\frac{1}{\omega}\ln\left[e^{\frac{\omega I}{\tau_n}\dot{\theta}_0} \pm e^{\frac{\omega I}{2\tau_n}\dot{\theta}_0}\sqrt{\frac{mg}{\tau_n}(x_0 + \frac{\dot{x}_0}{\omega}) + e^{\frac{\omega I}{\tau_n}\dot{\theta}_0} - 1}\right]$$

Note that here the sign '$\pm$', the negative value '$-$' should be chosen in order for $t_1 \geq 0$, namely

$$t_1 = -\frac{1}{\omega}\ln\left[e^{\frac{\omega I}{\tau_n}\dot{\theta}_0} - e^{\frac{\omega I}{2\tau_n}\dot{\theta}_0}\sqrt{\frac{mg}{\tau_n}(x_0 + \frac{\dot{x}_0}{\omega}) + e^{\frac{\omega I}{\tau_n}\dot{\theta}_0} - 1}\right]$$

with the constraint for the square root that

$$\frac{1}{\tau_n}(x_0 + \frac{\dot{x}_0}{\omega}) \geq \frac{1}{mg}(1 - e^{\frac{\omega I}{\tau_n}\dot{\theta}_0})$$

t_2 can be solved by Eq.(7.28). Having the torque τ_n and the time t_1, t_2, the trajectories of $x_{cm}(t)$ and $\dot{x}_{cm}(t)$ can be obtained.

Now we consider the critical boundaries for applying the pure hip strategy. Assuming that the hip pitch joints of the robot reach the ROM limit at $t = t_2$, from Eq.(7.27) we get

$$\theta(t_2) = \theta_{max} = \theta_0 + \dot{\theta}_0 t_2 + \frac{\tau_n}{I}(\frac{1}{2}t_2^2 - (t_2 - t_1)^2) \tag{7.29}$$

Substituting Eq.(7.28) into Eq.(7.29) and rearranging, we get a quadratic equation of variable t_1

$$\frac{\tau_n}{I}t_1^2 + 2\dot{\theta}_0 t_1 + (\frac{I}{2\tau_n}\dot{\theta}_0^2 + \theta_0 - \theta_{max}) = 0$$

with the solution $t_1 = -\frac{I}{\tau_n}\dot{\theta}_0 \pm \frac{I}{\tau_n}\sqrt{\frac{\dot{\theta}_0}{2} - \frac{\tau_n}{I}(\theta_0 - \theta_{max})}$

Because $t_1 > 0$, the constraint for the square root is $\frac{1}{2}\dot{\theta}_0^2 - \frac{\tau_n}{I}(\theta_0 - \theta_{max}) \geq \dot{\theta}_0^2$, namely $\frac{1}{2}\dot{\theta}_0^2 + \frac{\tau_n}{I}(\theta_0 - \theta_{max}) \leq 0$. If $(\theta_0, \dot{\theta}_0)$ satisfy the inequility,

$$t_1 = -\frac{I}{\tau_n}\dot{\theta}_0 + \frac{I}{\tau_n}\sqrt{\frac{\dot{\theta}_0^2}{2} - \frac{\tau_n}{I}(\theta_0 - \theta_{max})}$$

$$t_2 = -\frac{I}{\tau_n}\dot{\theta}_0 + \frac{2I}{\tau_n}\sqrt{\frac{\dot{\theta}_0^2}{2} - \frac{\tau_n}{I}(\theta_0 - \theta_{max})}$$

We use τ_{n_f} and τ_{n_b}, θ_{max_f} and θ_{max_b}, t_f and t_b to respectively denote the maximum torques the hip joints of the robot can apply, the ROM limits of hip pitch joints, and the acceleration and deceleration time in the forward and backward directions. From Eq.(7.25), the critical boundaries for applying the pure hip strategy to maintain the balance are

$$x_0 + \frac{\dot{x}_0}{\omega} - \frac{\tau_{n\_f}}{mg}(1 - 2e^{-\omega t_{1\_f}} + e^{-\omega t_{2\_f}}) = 0$$

$$x_0 + \frac{\dot{x}_0}{\omega} - \frac{\tau_{n\_b}}{mg}(1 - 2e^{-\omega t_{1\_b}} + e^{-\omega t_{2\_b}}) = 0$$

Therefore, if the initial state of the robot $(x_0, \dot{x}_0)$ satisfies the inequality (7.30), then the robot can potentially recover from the perturbation by executing the pure hip strategy. Or else, the robot cannot restore equilibrium using pure hip strategy only. Other strategies should be taken under the circumstance, such as combined ankle and hip strategy or stepping strategy. Calculation of the trajectories of the CoM given different initial conditions using pure hip strategy is shown in Algorithm 2. Figure 7.9 exhibits the calculated results.

$$\frac{\tau_{n\_b}}{mg}(1 - 2e^{-\omega t_{1\_b}} + e^{-\omega t_{2\_b}}) \leq x_0 + \frac{\dot{x}_0}{\omega} \leq \frac{\tau_{n\_f}}{mg}(1 - 2e^{-\omega t_{1\_f}} + e^{-\omega t_{2\_f}}) \tag{7.30}$$

where

$$t_{1\_f/b} = -\frac{I}{\tau_{n\_f/b}}\dot{\theta}_0 + \frac{I}{\tau_{n\_f/b}}\sqrt{\frac{\dot{\theta}_0^2}{2} - \frac{\tau_{n\_f/b}}{I}(\theta_0 - \theta_{max\_f/b})}$$

$$t_{2\_f/b} = -\frac{I}{\tau_{n\_f/b}}\dot{\theta}_0 + \frac{2I}{\tau_{n\_f/b}}\sqrt{\frac{\dot{\theta}_0^2}{2} - \frac{\tau_{n\_f/b}}{I}(\theta_0 - \theta_{max\_f/b})}$$

7.5 Combined Strategy

Combined strategy applies both ankle strategy and hip strategy to safeguard equilibrium of the system against external perturbations.

Applying the torque equilibrium condition into the model shown in Figure 7.10, we have

$$\tau_h + \tau_a + m\ddot{x}_{cm}z_{cm} - m\ddot{z}_{cm}x_{cm} = mgx_{cm} \tag{7.31}$$

where τ_h and τ_a are the torques that the hip joint and the ankle joint provide respectively. $\tau_a \cong mgx_{zmp}$ and $\tau_h = I\ddot{\theta}$. I is the rotational inertia of the HAT. $I = m_{HAT} \cdot d^2$, where d is the distance between the CoM of HAT and the hip joint. m is the mass of the robot. Here, we ignore the vertical movement of CoM, thus $\ddot{z}_{cm} = 0$. Then Eq.(7.31) becomes

$$\tau_h + \tau_a + m\ddot{x}_{cm}z_{cm} = mgx_{cm} \tag{7.32}$$

Algorithm 2 Calculate $x_{cm}(t)$ using pure hip strategy given initial conditions

Given τ_{n_f}, τ_{n_b}
Initial Conditions: x_0, $\dot{x}_0$, $\theta_0 = \dot{\theta}_0 = 0$,
if $x_0 + \frac{\dot{x}_0}{\omega} \leq \frac{\tau_{n\_f}}{mg}(1 - 2e^{-\omega t_{1\_f}} + e^{-\omega t_{2\_f}})$ and $x_0 + \frac{\dot{x}_0}{\omega} \geq 0$ **then**
  $t_{1\_f} = -\frac{1}{\omega}\ln\left[1 - \sqrt{\frac{mg}{\tau_{n\_f}}}(x_0 + \frac{\dot{x}_0}{\omega})\right]$, $t_{2\_f} = 2t_{1\_f}$
  $x_{cm}(t) = f(x_0, \dot{x}_0, t_{1\_f}, t_{2\_f}, \tau_{n\_f})$
else if $x_0 + \frac{\dot{x}_0}{\omega} \geq \frac{\tau_{n\_b}}{mg}(1 - 2e^{-\omega t_{1\_b}} + e^{-\omega t_{2\_b}})$ and $x_0 + \frac{\dot{x}_0}{\omega} \leq 0$ **then**
  $t_{1\_b} = -\frac{1}{\omega}\ln\left[1 - \sqrt{\frac{mg}{\tau_{n\_b}}}(x_0 + \frac{\dot{x}_0}{\omega})\right]$, $t_{2\_b} = 2t_{1\_b}$
  $x_{cm}(t) = f(x_0, \dot{x}_0, t_{1\_b}, t_{2\_b}, \tau_{n\_b})$
else if $x_0 + \frac{\dot{x}_0}{\omega} > \frac{\tau_{n\_f}}{mg}(1 - 2e^{-\omega t_{1\_f}} + e^{-\omega t_{2\_f}})$ **then**
  $t_1 \Leftarrow t_{1\_f}$, $t_2 \Leftarrow t_{2\_f}$
  $x_{cm}(t) = f(x_0, \dot{x}_0, t_1, t_2, \tau_{n\_f})$
else if $x_0 + \frac{\dot{x}_0}{\omega} < \frac{\tau_{n\_b}}{mg}(1 - 2e^{-\omega t_{1\_b}} + e^{-\omega t_{2\_b}})$ **then**
  $t_1 \Leftarrow t_{1\_b}$, $t_2 \Leftarrow t_{2\_b}$
  $x_{cm}(t) = f(x_0, \dot{x}_0, t_1, t_2, \tau_{n\_b})$
end if

Rearranging Eq.(7.32), we get

$$\begin{aligned} \ddot{x}_{cm} - \frac{g}{z_{cm}}x_{cm} &= -\frac{1}{mz_{cm}}(\tau_h + \tau_a) \\ &= -\frac{1}{mz_{cm}}\tau_h - \frac{g}{z_{cm}}x_{zmp} \end{aligned} \tag{7.33}$$

Let $\omega^2 = \frac{g}{z_{cm}}$, and $f = -\frac{1}{mz_{cm}}\tau_h - \frac{g}{z_{cm}}x_{zmp}$, then Eq.(7.33) becomes $\ddot{x}_{cm} - \omega^2 x_{cm} = f$ with the initial conditions $x(0) = x_0$ and $\dot{x}(0) = \dot{x}_0$. The solution of the equation is

$$x_{cm}(t) = (\frac{f}{\omega^2} + x_0)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) - \frac{f}{\omega^2} \tag{7.34}$$

7.5.1 Bang-Bang Control

We apply the torque profile adopted in the pure hip strategy

$$\tau_h(t) = \tau_h u(t) - 2\tau_h u(t - t_1) + \tau_h u(t - t_2) \tag{7.35}$$

where τ_h is the torque that the hip joint can apply. t_1 is the time when the HAT stops accelerating and starts decelerating and t_2 is the time when the HAT comes to a stop.

Substituting Eq.(7.35) into Eq.(7.34), we get the solution for the trajectory of CoM

$$\begin{aligned} x_{cm}(t) &= (x_0 - x_{zmp})\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) + x_{zmp} - \frac{\tau_h}{mg}[(\cosh(\omega t) - 1)u(t) - \\ &\quad 2(\cosh(\omega(t - t_1)) - 1)u(t - t_1) + (\cosh(\omega(t - t_2)) - 1)u(t - t_2)] \end{aligned}$$

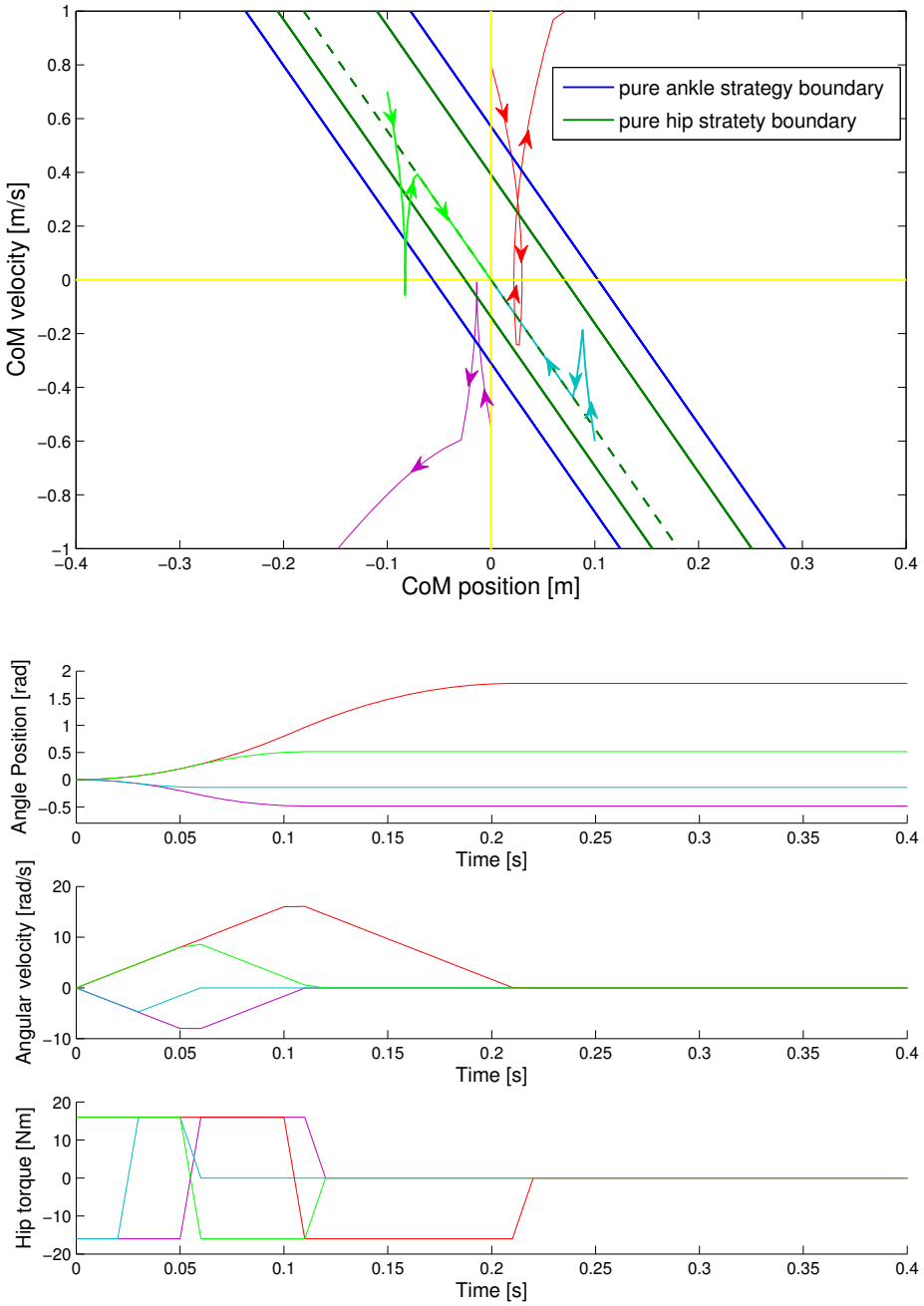


Figure 7.9: Trajectories using angular momentum of the hip joints to safeguard balance. (Top) CoM trajectories. The trajectories which start inside the pure hip strategy boundaries $(x_0, \dot{x}_0) =$ (-0.1 m, 0.7 m/s) and $(x_0, \dot{x}_0) =$ (0.1 m, -0.6 m/s) are potentially stable; On the contrary, the ones starting outside the area $(x_0, \dot{x}_0) =$ (0 m, -0.55 m/s) and $(x_0, \dot{x}_0) =$ (0 m, 0.8 m/s) cannot restore equilibrium. The figure is plotted using parameters of $\theta_0 = 0$ rad, $\dot{\theta}_0 = 0$ rad/s, $I =$ 0.1 kgm^2, $\tau_{n\_f} = 16$ Nm, $\tau_{n\_b} = -16$ Nm, $\theta_{max\_f} = 1.772308$ rad and $\theta_{max\_b} = -0.485624$ rad; (Bottom) corresponding hip joint angle, angular velocity and hip torque.

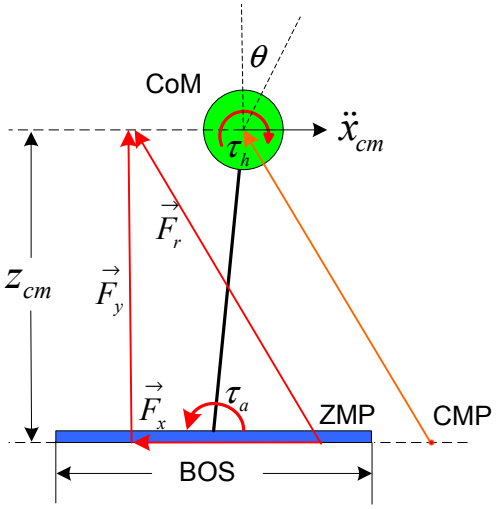


Figure 7.10: Schematic diagram of combined strategy.

The necessary and sufficient condition that $x_{cm}(t)$ converges as $t \to \infty$ is,

$$(x_0 - x_{zmp}) + \frac{\dot{x}_0}{\omega} - \frac{\tau_h}{mg}(1 - 2e^{-\omega t_1} + e^{-\omega t_2}) = 0 \tag{7.36}$$

Using Eq.(7.26), the angular velocity of the hip pitch joint can be calculated. For simplicity, suppose $\dot{\theta}_0 = 0$, then $t_2 = 2t_1$ from Eq.(7.28). Let $t_1 = T$, Eq.(7.36) becomes

$$(x_0 - x_{zmp}) + \frac{\dot{x}_0}{\omega} - \frac{\tau_h}{mg}(1 - e^{-\omega T})^2 = 0$$

The time T can be analytical solved.

$$T = -\frac{1}{\omega} \ln \left[1 - \sqrt{\frac{mg}{\tau_h}(x_0 + \frac{\dot{x}_0}{\omega} - x_{zmp})} \right]$$

The critical boundaries for applying the combined hip strategy to maintain the balance are

$$(x_0 - x_{zmp\_max}) + \frac{\dot{x}_0}{\omega} - \frac{\tau_{h\_f}}{mg}(1 - e^{-\omega t_{1\_f}})^2 = 0$$

$$(x_0 - x_{zmp\_min}) + \frac{\dot{x}_0}{\omega} - \frac{\tau_{h\_b}}{mg}(1 - e^{-\omega t_{1\_b}})^2 = 0$$

$$t_{1\_f/b} = \sqrt{\frac{I}{\tau_{max\_f/b}}(\theta_{max\_f/b} - \theta_0)}$$

$$t_{2\_f/b} = 2t_{1\_f/b} = \sqrt{\frac{4I}{\tau_{max\_f/b}}(\theta_{max\_f/b} - \theta_0)}$$

Therefore, if the initial state of the robot satisfies the inequality (7.37), the system can potentially stabilize by executing the combined strategy and recover from the perturbations.

Or else, the system will fall down or it has to take a step to avoid falling down. Figure 7.11 demonstrates the calculated results.

$$x_{zmp\_min} + \frac{\tau_{h\_b}}{mg}(1 - e^{-\omega t_{1\_b}})^2 \leq x_0 + \frac{\dot{x}_0}{\omega} \leq x_{zmp\_max} + \frac{\tau_{h\_f}}{mg}(1 - e^{-\omega t_{1\_f}})^2 \tag{7.37}$$

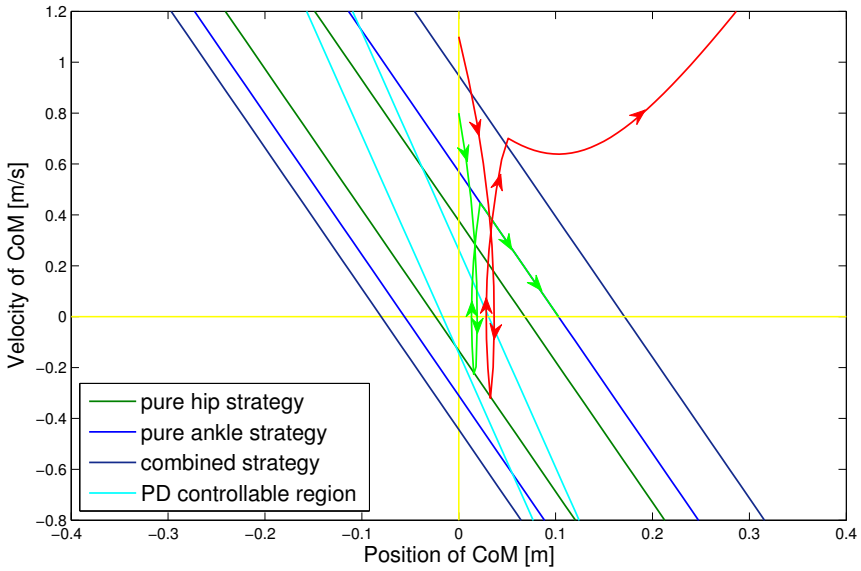


Figure 7.11: Trajectories of CoM using combined strategy to safeguard balance. The trajecotries start with initial conditions $(x_0, \dot{x}_0)$ = (0 m, 0.8 m/s) (green) and $(x_0, \dot{x}_0)$ = (0 m, 1.1 m/s) (red), respectively.

7.5.2 LQR Control

We apply the LQR optimal control to drive the system back to the origin after an initial impulse disturbance gives a non-zero velocity of the CoM.

The equations of motion can be transformed into a first-order system with system dynamics

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{7.38}$$

with $x \in \mathbb{R}^n$ being the state vector, $u \in \mathbb{R}^m$ denoting the input of the system, and y denoting the output of the system.

Let $x = [x_{cm} \quad \theta \quad \dot{x}_{cm} \quad \dot{\theta}]^T$. Combining Eq.(7.33), Eq.(7.38) can be written as

$$\begin{pmatrix} \dot{x}_{cm} \\ \dot{\theta} \\ \ddot{x}_{cm} \\ \ddot{\theta} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x_{cm} \\ \theta \\ \dot{x}_{cm} \\ \dot{\theta} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\omega^2 & -(mz_{cm})^{-1} \\ 0 & I^{-1} \end{bmatrix} \begin{pmatrix} x_{zmp} \\ \tau_h \end{pmatrix}$$

$$y = \begin{pmatrix} x_{cm} \\ \theta \\ \dot{x}_{cm} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{cm} \\ \theta \\ \dot{x}_{cm} \\ \dot{\theta} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x_{zmp} \\ \tau_h \end{pmatrix}$$

The quadratic cost function for the infinite horizon is

$$J = \frac{1}{2} \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \tag{7.39}$$

where Q is an $n \times n$ symmetric positive semidefinite matrix and R is an $m \times m$ symmetric positive definite matrix. It has been shown in classical optimal control theory that for a continuous time system, to minimize the above quadratic cost function, the LQR optimal control has the feedback form

$$u(t) = -K(t)x(t)$$

where $K(t)$ is a properly dimensioned matrix, given as $K(t) = R^{-1}B^T S$, and S is the solution of the algebraic Riccati equation which is given as:

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

We implemented the optimal control in Matlab/Simulink R2012b. Using syntax [K, S, e] = LQR(A, B, Q, R) in Matlab, the state-feedback gain K and the solution of the algebraic Riccati equation S can be obtained for continuous-time models with dynamics $\dot{x} = Ax + Bu$. Figure 7.12 demonstrates the simulation model in MATLAB/Simulink.

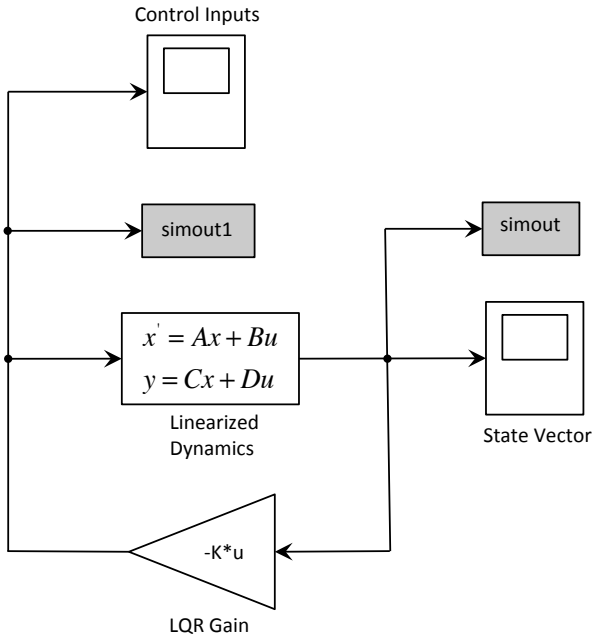


Figure 7.12: LQR optimal control diagram in Matlab/simulink.

Figure 7.13 and Figure 7.14 show simulation results of the optimal control, where the manually adjusted parameters are

$$Q = diag[2000 \quad 500 \quad 1 \quad 10];$$
$$R = diag[4500000 \quad 100].$$

Q and R were determined empirically.

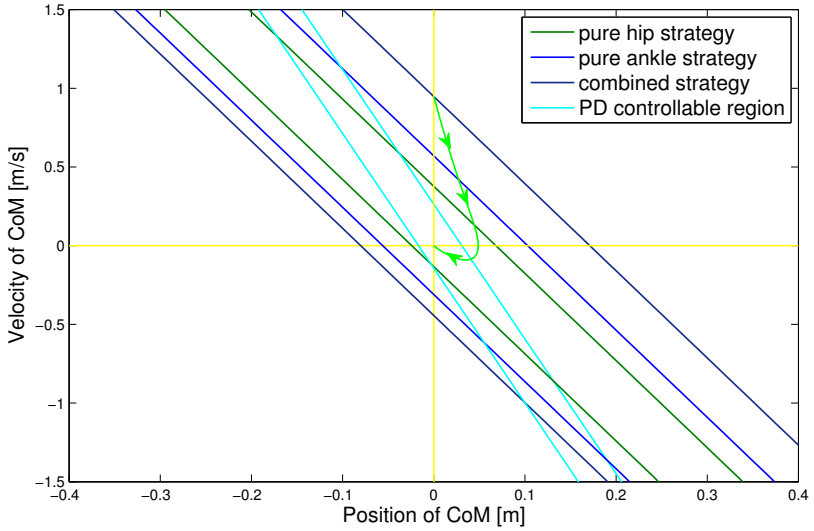


Figure 7.13: CoM trajectory using LQR optimal control. The initial condition is $(x_0, \theta_0, \dot{x}_0, \dot{\theta}_0)$ = (0 m, 0 rad, 0.9475 m/s, 0 rad/s).

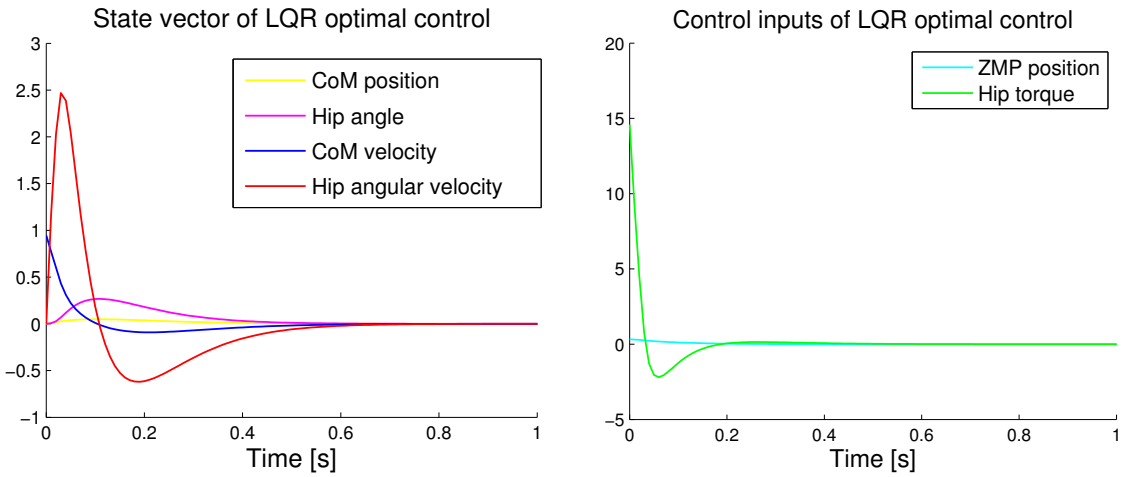


Figure 7.14: State vector and control inputs of LQR optimal control.

7.6 Chapter Summary

This chapter theoretically analyzes several simple models for rescue strategies applied to restoring balance while original equilibrium of the system is disturbed by unexpected external disturbances. Using these simple models, we developed analytic boundaries that are functions of the CoM and CoP that can be measured or calculated easily for both robots and humans. The boundaries predict whether or not a fall is inevitable. We explore three strategies for recovery: 1) using ankle torques (CoP balancing), 2) using hip joints (CMP balancing), and 3) using both ankle torques and hip joints. These models can be used in robot controllers or in analysis of human balance and locomotion.

8 Conclusions

Bipedal humanoid robots have great potential in the service area. Because of their anthropomorphic appearance and motion, they enhance the chances of acceptance by human beings and simplify human-machine communication. In addition, bipedal humanoid robots are able to well locomote on various terrains such as uneven surfaces or stairs. So they have better adaptability to the environment where human beings live and require no changes to the existing infrastructure. One of the most basic requirements for such robots is autonomous, reliable and fast biped locomotion. The preceding chapters of this thesis have presented a framework for realizing stable and efficient locomotion of humanoid robots both in simulation and in experiments. A brief summary of key ideas and contributions is given in the following section. The final section outlines suggestions for future research on biped locomotion.

8.1 Summary

Motivated by biomechanical studies on human locomotion, this thesis covers motion generation and control of biped walking robots.

In the first part, a new method for designing a natural and efficient walking pattern of humanoid robots was presented. We use a simplification of the walking related dynamics of the humanoid robot by applying the equations of motion to a point mass concentrated on the position of the robot waist. Inspired by the walking patterns occurring in human beings, we model the walking pattern of a bipedal humanoid robot by continuous and differentiable mathematical functions. The walking pattern satisfies the ZMP stability criterion. The proposed walking pattern involves upward and downward motion of the upper body of the robot. In comparison to the walking pattern that restricts the upper body motion to a flat plane, the proposed walking pattern involves three-dimensional upper body motion which is more natural and human-like. Besides, the walking pattern generation method needs lower calculation cost compared with that based on precise knowledge of robot dynamics.

The second part of this thesis presents a real-time walking control system aiming at fast, stable and autonomous biped walking. Several controllers for stabilizing biped walking were developed, including a torso pitch/roll controller, a yaw moment controller, a swing-leg trajectory tracking controller and a landing impact controller. Based on sensory feedback, the torso pitch/roll controller utilizes hip/ankle joints and modifies the reference joint trajectories in real time in order to keep an upright posture of the upper body of the robot during walking. The yaw moment controller uses arm motion to compensate for the yaw moment generated on the support foot caused by the acceleration and deceleration of the swing-leg. The swing-leg trajectory tracking controller realizes fast and precise tracking of the reference walking pattern of the swing-leg using adaptive control approach in order to achieve fast walking of

the robot. The landing impact controller employs compliant force control approach with the purpose of decreasing the impact force.

Using this system, NAO achieved dynamic stable walking and reached a maximum walking speed of 0.24 m/s in experiments. In addition, NAO could walk with almost stretched knees while the walking speed was lower, e.g. 0.10 m/s, so that the efficient walking was realized since the knee joints consumed less energy compared with the conventional walking pattern.

The last part presents the balance recovery strategies in the presence of external perturbations in order for the robot to maintain an upright standing posture. Using simple models that biomechanists and roboticists used previously to explain humanoid balance and control, human-like rescue strategies are theoretically analyzed. The analysis paves the way toward building controllers that ensure a humanoid can withstand disturbances.

8.2 Future Work

From the experience gained during this thesis, suggestions for future research on the humanoid robot NAO are put forward.

Increasing Walking Speed

Using the walking system presented in this thesis, NAO reached a maximum walking speed of 0.24 m/s in experiments. This speed exceeds the "maximum walking speed" given by Aldebaran in an earlier specification[1]. However, the walking speed is still moderate. Increasing the walking speed, and at the same time maintaining the dynamic walking stability should be one of the objectives for future research. Improvement of the current walking pattern and control algorithms for the purpose of fast walking is suggested. For example, instead of cycloid function an optimizational foot trajectory with the aim of decreasing the landing impact force and better synchronizing with the motion of the upper body can be designed.

Landing Impact Control

Since NAO is not force-controllable, the impedance control method presented in Chapter 6 to decrease the landing impact force cannot be implemented on the robot. Currently, oscillation and rebounding of the foot brought about by the impact force are counteracted by shortening the walking period to about half second. The rapid movement of the legs results in fast arm motion for compensating the yaw moment (proposed in Chapter 4), which causes vibration of the upper body of the robot and consequently unstable walking.

Not only stable landing of the foot on the ground but also compensatory arm motion can be achieved through reducing the landing impact force. Landing impact control is, therefore, of great importance for the overall stability of the walking system. The future work is suggested to involve the realization of the landing impact control algorithm, based either on the position control approach or on the force control approach, on the real robot.

[1] In this earlier specification, the maximum walking speed of NAO was up to 0.14 m/s. Source: Release notes - 1.12/What's new 1.12/Motion/Faster Walk. http://doc.aldebaran.com/1-14/news/1.12/whatsnew.html.

Balance Recovering While Standing

Simple models for balance recovery strategies against unexpected external disturbances have been theoretically analyzed in Chapter 7. The robot controller that is inspired by human balance strategies and responsible for regulating the location of CoP to ensure that a humanoid can stand with its feet flat on the ground and withstand disturbances is recommended to be developed for the future work.

For humanoid robots, following suggestions are made.

Application in Human-living Environment

One of the most useful applications of humanoid robots is in dangerous or otherwise unreachable scenarios, for example disaster relief, bomb disarming, planetary exploration or anti-terrorism action. These tasks require robust and efficient systems. However, there are currently few examples of humanoid robots performing tasks outside of the research laboratory. Not only humanoid robot controllers but also the hardware and system integration need to be improved to allow robots capable of real world application.

Application in Being as a Research Tool

One of the applications of building and controlling humanoid robots is to help us understand how humans locomote. The design of humanoid robot hardware and control can benefit from existing biomechanical models and observations of humans. However, conversely there is little effort to use these results to test biomechanical models of humans. A greater efforts on such bidirectional cooperation will lead to accelerated development of robust human-like control for robots, and new tools and devices for evaluating and assisting humans.

Bibliography

[1] Normal Ranges of Joint Motion. URL: http://people.bath.ac.uk/masrjb/Stretch/stretching_8.html#SEC92, 2014.

[2] Aldebaran Robotics. Technical Specification for Robocup Edition Only. Available online. URL: https://www.ucursos.cl/ingenieria/2009/2/EL710/1/material_docente/previsualizar?id_material=243598.

[3] Atkeson, C. G., and Stephens, B. Multiple Balance Strategies From One Optimization Criterion. In *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 57–64.

[4] Azevedo, C., Poignet, P., and Espiau, B. Artificial Locomotion Control: from Human to Robots. *Robotics and Autonomous Systems* (2004), 203–223.

[5] Azimi, E., Ghobadi, M., Esfahani, E. T., Keshmiri, M., and Tehrani, A. F. Three-Dimensional Smooth Trajectory Planning Using Realistic Simulation. *RoboCup 2004* (2004), 381–393.

[6] Baptista, L. F., Sousa, J. M., and Sá da Costa, J. M. G. Fuzzy Predictive Algorithms Applied to Real-time Force Control. *Control Engineering Practice 9*, 4 (2001), 411–423.

[7] Buschmann, T. *Simulation and Control of Biped Walking Robots*. PhD thesis, Technische Universität München, München, 2010.

[8] Buschmann, T., Lohmeier, S., and Ulbrich, H. Humanoid Robot Lola: Design and Walking Control. *Journal of Physiology - Paris 103* (2009), 141–148.

[9] Cho, B.-K., Park, S.-S., and Oh, J.-H. Controllers for Running in the Humanoid Robot, HUBO. In *Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots* (2009), pp. 385–390.

[10] Choi, Y., Kim, D., Oh, Y., and You, B.-J. Posture/Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian with Embedded Motion. *IEEE Transactions on Robotics 23*, 6 (2007), 1285–1293.

[11] Craig, J. J., and Raibert, M. H. A Systematic Method of Hybrid Position/Force Control of a Manipulator. In *Proceedings of the IEEE Computer Society's 3rd International Computer Software and Applications Conference* (1979), pp. 446–451.

[12] Dekker, M. Zero-Moment Point Method for Stable Biped Walking. Internship report, Eindhoven University of Technology, Netherlands, 2009.

[13] Denavit, J., and Hartenberg, R. S. A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *Transactions of the ASME Journal of Applied Mechanics 23* (1955), 215–221.

[14] Dimitrov, D., Wieber, P.-B., Ferreau, H. J., and Diehl, M. On the Implementation of Model Predictive Control for On-line Walking Pattern Generation. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation* (2008), pp. 2685–2690.

[15] Featherstone, R., and Orin, D. E. Robot Dynamics: Equations and Algorithms. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* (2000), pp. 826–834.

[16] Goswami, A. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *International Journal of Robotics Research 18*, 6 (1999), 523–533.

[17] Goswami, A., and Kallem, V. Rate of Change of Angular Momentum and Balance Maintenance of Biped Robots. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation* (2004), pp. 3785–3790.

[18] Hamilton, N., Weimar, W., and Luttgens, K. *Kinesiology: Scientific Basis of Human Motion*, 12th ed. McGraw-Hill, New York, 2011.

[19] Handharu, N., Yoon, J., and Kim, G. Gait Pattern Generation with Knee Stretch Motion for Biped Robot using Toe and Heel Joints. In *Proceedings of the 2008 8th IEEE-RAS International Conference on Humanoid Robots* (2008), pp. 265–270.

[20] Harada, K., Kajita, S., Kaneko, K., and Hirukawa, H. Pushing Manipulation by Humanoid considering Two-Kinds of ZMPs. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* (2003), pp. 1627–1632.

[21] Harada, K., Kajita, S., Kaneko, K., and Hirukawa, H. An Analytical Method on Real-Time Gait Planning for a Humanoid Robot. In *Proceedings of the 2004 IEEE-RAS/RSJ International Conference on Robotics and Automation* (2004), pp. 640–655.

[22] Hashimoto, K., Kang, H.-j., Nakamura, M., Falotico, E., Lim, H.-o., Takanishi, A., Laschi, C., Dario, P., and Berthoz, A. Realization of Biped Walking on Soft Ground with Stabilization Control Based on Gait Analysis. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), pp. 2064–2069.

[23] Hashimoto, K., Takezaki, Y., Hattori, K., Kondo, H., Takashima, T., Lim, H.-o., and Takanishi, A. A Study of Function of Foot's Medial Longitudinal Arch Using Biped Humanoid Robot. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 2206–2211.

[24] Hashimoto, K., Yoshimura, Y., Kondo, H., Lim, H.-o., and Takanishi, A. Realization of Quick Turn of Biped Humanoid Robot by Using Slipping Motion with

Both Feet. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation* (2011), pp. 2041–2046.

[25] Hildebrand, M. Symmetrical Gaits of Horses. *Science 150*, 3697 (1965), 701–708.

[26] Hirabayashi, T., Ugurlu, B., Kawamura, A., and Zhu, C. Yaw Moment Compensation of Biped Fast Walking Using 3D Inverted Pendulum. In *Proceedings of the 10th IEEE International Workshop on Advanced Motion Control* (2008), pp. 296–300.

[27] Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. The Development of Honda Humanoid Robot. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation* (1998), vol. 2, pp. 1321–1326.

[28] Hofmann, A. *Robust Execution of Bipedal Walking Tasks from Biomechanical Principles*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2006.

[29] Hogan, N. Impedance Control: An Approach to Manipulation, Part I - Part III. *Journal of Dynamic Systems, Measurement and Control 107* (Mar. 1985), 1–24.

[30] Honda Worldwide. Company Website for the Robots P1, P2, P3 and ASIMO. URL: http://world.honda.com/ASIMO/history/p1_p2_p3.html, 2013.

[31] Horak, F. B., and Nashner, L. M. Central Programming of Postural Movements: Adaptation to Altered Support-Surface Configurations. *Journal of Neurophysiology 55*, 6 (June 1986), 1369–1381.

[32] Huang, Q., Kajita, S., Koyachi, N., Kaneko, K., Yokoi, K., Arai, H., Komoriya, K., and Tanie, K. A High Stability, Smooth Walking Pattern for a Biped Robot. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (1999), pp. 65–71.

[33] Huang, Q., and Nakamura, Y. Sensory Reflex Control for Humanoid Walking. *IEEE Transactions on Robotics 21*, 5 (2005), 977–984.

[34] Huang, Q., Tanie, K., and Sugano, S. Coordinated Motion Planning for a Mobile Manipulator Considering Stability and Manipulation. *International Journal of Robotics Research 19*, 8 (2000), 732–742.

[35] HuboLab. Website for the HuboLab-KAIST. URL: http://hubolab.kaist.ac.kr/p_hubo2p, 2014.

[36] Hyon, S.-H., and Cheng, G. Disturbance Rejection for Biped Humanoids. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation* (2007), pp. 2668–2675.

[37] Hyon, S.-H., Osu, R., and Otaka, Y. Integration of Multi-level Postural Balancing on Humanoid Robots. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* (2009), pp. 1549–1556.

[38] Inada, H., and Ishii, K. Behavior Generation of Bipedal Robot Using Central Pattern Generator (CPG) (1st Report: CPG Parameters Searching Method by Genetic Algorithm). In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2003), pp. 2179–2184.

[39] Ito, D., Murakami, T., and Ohnishi, K. An Approach to Generation of Smooth Walking Pattern for Biped Robot. *7th International Workshop on Advanced Motion Control* (July 2002), 98–103.

[40] Kagami, S., Kitagawa, T., Nishiwaki, K., Sugihara, T., Inaba, M., and Inoue, H. A Fast Dynamically Equilibrated Walking Trajectory Generation Method of Humanoid Robot. *Autonomous Robots 12* (Jan. 2002), 71–82.

[41] Kagami, S., Nishiwaki, K., Kitagawa, T., Sugihara, T., Inaba, M., and Inoue, H. A Fast Generation Method of a Dynamically Stable Humanoid Robot Trajectory with Enhanced ZMP Constraint. In *Proceedings of the IEEE International Conference on Humanoid Robotics (Humanoid 2000)* (2000).

[42] Kagami, S., Nishiwaki, K., Kuffner, J., Kuniyoshi, Y., Inaba, M., and Inoue, H. Online 3D Vision, Motion Planning and Biped Locomotion Control Coupling System of Humanoid Robot H7. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 2557–2562.

[43] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* (2003), pp. 1620–1626.

[44] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. Resolved Momentum Control: Humanoid Motion Planning Based on the Linear and Angular Momentum. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2003), pp. 1644–1650.

[45] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., and Hirukawa, H. Biped Walking Pattern Generation by a Simple Three-dimensional Inverted Pendulum Model. *Journal of Advanced Robotics 17*, 2 (2003), 131–147.

[46] Kajita, S., Matsumoto, O., and Saigo, M. Real-time 3D Walking Pattern Generation for a Biped Robot with Telescopic Legs. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation* (2001), pp. 2299–2306.

[47] Kajita, S., Nagasaki, T., Kaneko, K., Yokoi, K., and Tanie, K. A Running Controller of Humanoid Biped HRP-2LR. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), pp. 616–622.

[48] Kajita, S., and Tani, K. 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* (1991), pp. 1405–1411.

[49] Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME Journal of Basic Engineering*, 82 (1960), 35–45.

[50] Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Kajita, S., Yokoi, K., Hirukawa, H., Akachi, K., and Isozumi, T. The First Humanoid Robot that has the Same Size as a Human and that can Lie down and Get up. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* (2003), pp. 1633–1639.

[51] Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., and Akachi, K. Humanoid Robot HRP-3. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008), pp. 2471–2478.

[52] Kato, I., Mori, Y., and Masuda, T. Pneumatically Powered Artificial Legs Walking Automatically Under Various Circumstances. In *Proceedings of the Fourth International Symposium on External Control of Human Extremities* (1972), pp. 458–470.

[53] Kato, I., Ohteru, S., Kobayashi, H., Shirai, K., and Uchiyama, A. Information-Power Machine with Senses and Limbs (Wabot 1). In *Proceedings of the First CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators* (1973), vol. 1, pp. 11–24.

[54] Kawada Industries. Company Website for the HRP-4 Robot. URL: http://global.kawada.jp/mechatronics/hrp4.html, 2013.

[55] Kazerooni, H., Sheridan, T. B., and Houpt, P. K. Robust Compliant Motion for Manipulators, Part I: The Fundamental Concepts of Compliant Motion. *IEEE Journal of Robotics and Automation 2* (June 1986), 83–92.

[56] Kerner, S. *Einsatz propriozeptiver Sensorik in der Bewegungssteuerung humanoider Roboter*. PhD thesis, Technische Universität Dortmund, Dortmund, Germany, 2013.

[57] Khatib, O. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE Journal of Robotics and Automation 3* (Feb. 1987), 43–53.

[58] Kim, J.-Y., Park, I.-W., and Oh, J.-H. Experimental Realization of Dynamic Walking of Biped Humanoid Robot KHR-2 using ZMP Feedback and Inertial Measurement. *Journal of Advanced Robotics 20*, 6 (2006), 707–736.

[59] Kim, J.-Y., Park, I.-W., and Oh, J.-H. Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor. *Journal of Intelligent and Robotic Systems 48*, 4 (Apr. 2007), 457–484.

[60] Kim, M.-s., Park, I.-W., Kim, J.-Y., and Oh, J.-h. Stretch-Legged Walking in Sagittal Plane. In *Proceedings of the 2007 7th IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 276–281.

[61] Kim, Y.-D., Lee, B.-J., Ryu, J.-H., and Kim, J.-H. Landing Force Control for Humanoid Robot by Time-Domain Passivity Approach. *IEEE Transactions on Robotics 23*, 6 (2007), 1294–1301.

[62] Komatsu, T., and Usui, M. 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* (July 2005), vol. 2, pp. 987–992.

[63] Komura, T., Leung, H., Kudoh, S., and Kuffner, J. A Feedback Controller for Biped Humanoids that can Counteract Large Perturbations During Gait. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), pp. 1989–1995.

[64] Kondo, H., Morishima, A., Ogura, Y., Momoki, S., Shimizu, J., Lim, H.-o., and Takanishi, A. Algorithm of Pattern Generation for Mimicking Disabled Person's Gait. In *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (2008), pp. 724–729.

[65] Kuo, A. D. An Optimal Control Model for Analyzing Human Postural Balance. *IEEE Transactions on Biomedical Engineering 42*, 1 (Jan. 1995), 87–101.

[66] Kurazume, R., Tanaka, S., Yamashita, M., Hasegawa, T., and Yoneda, K. Straight Legged Walking of a Biped Robot. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005), pp. 3095–3101.

[67] Laue, T., Spiess, K., and Röfer, T. Simrobot – A General Physical Robot Simulator and its Application in RoboCup. In *RoboCup 2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence* (2006), Springer, pp. 173 –183.

[68] Li, Q., Takanishi, A., and Kato, I. Learning Control of Compensative Trunk Motion for Biped Walking Robot based on ZMP Stability Criterion. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems* (1992), pp. 597–603.

[69] Lim, H.-o., Kaneshima, Y., and Takanishi, A. Online Walking Pattern Generation for Biped Humanoid Robot with Trunk. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* (2002), pp. 3111–3116.

[70] Lim, H.-o., Setiawan, S. A., and Takanishi, A. Position-based Impedance Control of a Biped Humanoid Robot. *Advanced Robotics 18* (2004), 415–435.

[71] Lipfert, S. W., Günther, M., Renjewski, D., Grimmer, S., and Seyfarth, A. A Model-experiment Comparison of System Dynamics for Human Walking and Running. *Journal of Theoretical Biology 292* (Jan. 2012), 11–17.

[72] Lu, Z., and Goldenberg, A. A. Robust Impedance Control and Force Regulation: Theory and Experiments. *The International Journal of Robotics Research 14* (June 1995), 225–254.

[73] Löffler, K., Gienger, M., Pfeiffer, F., and Ulbrich, H. Sensors and Control Concept of a Biped Robot. *IEEE Transactions on Industrial Electronics 51*, 5 (2004), 972–980.

[74] Macchietto, A., Zordan, V., and Shelton, C. R. Momentum Control for Balance. *ACM Transactions on Graphics 28*, 3 (Aug. 2009).

[75] Maki, B. E., and McIlroy, W. E. The Role of Limb Movements in Maintaining Upright Stance: The "Change-in-Support" Strategy. *Physical Therapy 77*, 5 (May 1997), 488–507.

[76] McGeer, T. Passive Dynamic Walking. *International Journal of Robotics Research 9*, 2 (1990), 68–82.

[77] Mille, M.-L., Rogers, M. W., Martinez, K., Hedman, L. D., Johnson, M. E., Lord, S. R., and Fitzpatrick, R. C. Thresholds for Inducing Protective Stepping Responses to External Perturbations of Human Standing. *Journal of Neurophysiology 90*, 2 (Aug. 2003), 666–674.

[78] Mitobe, K., Capi, G., and Nasu, Y. A New Control Method for Walking Robots Based on Angular Momentum. *Mechatronics 14*, 2 (2004), 163–174.

[79] Morisawa, M., Kajita, S., Kaneko, K., Harada, K., Kanehiro, F., Fujiwara, K., and Hirukawa, H. Pattern Generation of Biped Walking Constrained on Parametric Surface. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), pp. 2405–2410.

[80] Nagasaka, K., Inoue, H., and Inaba, M. 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* (Oct. 1999), vol. 6, pp. 908–913.

[81] Nenchev, D. N., and Nishio, A. Experimental Validation of Ankle and Hip Strategies for Balance Recovery with a Biped Subjected to an Impact. In *Proceedings of the 2007 IEEE/RAJ International Conference on Intelligent Robots and Systems* (2007), pp. 4035–4039.

[82] Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. Online Generation of Humanoid Walking Motion based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), vol. 3, pp. 2684–2689.

[83] Nishiwaki, K., Sugihara, T., Kagami, S., Inaba, M., and Inoue, H. Online Mixture and Connection of Basic Motions for Humanoid Walking Control by Footprint Specification. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation* (2001), pp. 4110–4115.

[84] Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., and Inoue, H. Design and Development of Research Platform for Perception Action Integration in Humanoid Robot: H6. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2000), pp. 1559–1564.

[85] Ogura, Y., Aikawa, H., Shimomura, K., Kondo, H., Morishima, A., Lim, H.-o., and Takanishi, A. Development of a New Humanoid Robot WABIAN-2. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (2006), pp. 76–81.

[86] Ogura, Y., Kataoka, T., Shimomura, K., Lim, H.-o., and Takanishi, A. A Novel Method of Biped Walking Pattern Generation with Predetermined Knee Joint Motion. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004), pp. 2831–2836.

[87] Ogura, Y., Lim, H.-o., and Takanishi, A. Stretch Walking Pattern Generation for a Biped Humanoid Robot. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robot and Systems* (2003), pp. 352–357.

[88] Ogura, Y., Shimomura, K., Kondo, H., Morishima, A., Okubo, T., Momoki, S., Lim, H.-o., and Takanishi, A. Human-like Walking with Knee Stretched, Heel-contact and Toe-off Motion by a Humanoid Robot. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), pp. 3976–3981.

[89] Oh, J.-H., Hanson, D., Kim, W.-S., Han, Y., Kim, J.-Y., and Park, I.-W. Design of Android Type Humanoid Robot Albert HUBO. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), pp. 1428–1433.

[90] Omer, A. M. M., Lim, H.-o., and Takanishi, A. Simulation Study of a Bipedal Robot Jumping Motion Approach on Moon Gravity. In *Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics* (2010), pp. 218–222.

[91] Park, I.-W., Kim, J.-Y., Lee, J., and Oh, J.-H. Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3 (HUBO). In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (2006), pp. 1231–1236.

[92] Park, J. H., and Lee, S. Generation of Optimal Trajectory for Biped Robots with Knees Stretched. In *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics* (2009), pp. 166–171.

[93] Perry, J., and Burnfield, J. M. *Gait Analysis: Normal and Pathological Function*, 2nd ed. SLACK Incorporated, Thorofare, New Jersey, 2010.

[94] Popovic, M., Hofmann, A., and Herr, H. Angular Momentum Regulation during Human Walking: Biomechanics and Control. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2004), pp. 2405–2411.

[95] Popovic, M. B., Goswami, A., and Herr, H. Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications. *The International Journal of Robotics Research 24*, 12 (2005), 1013–1032.

[96] Pratt, J., Carff, J., Drakunov, S., and Goswami, A. Capture Point: A Step toward Humanoid Push Recovery. In *Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots* (2006), pp. 200–207.

[97] Pratt, J. E. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[98] Raibert, M. H., and Craig, J. J. Hybrid Position/Force Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control 103* (June 1981), 126–133.

[99] Rose, J., and Gamble, J. G., Eds. *Human Walking*, third ed. Lippincott Williams & Wilkins, London, 2005, pp. 1–27.

[100] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., and Fujimura, K. The Intelligent ASIMO: System Overview and Integration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 2478–2483.

[101] Sekiguchi, A., Atobe, Y., Kameta, K., Tsumaki, Y., and Nenchev, D. A Walking Pattern Generator around Singularity. In *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robots* (2006), pp. 270–275.

[102] Sekiguchi, A., Kameta, K., Tsumaki, Y., and Nenchev, D. Biped Walk Based on Vertical Pivot Motion of Linear Inverted Pendulum. In *Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (Sept. 2007), pp. 1–6.

[103] Slotine, J.-J. E., and Li, W. *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[104] Spong, M. W., and Vidyasagar, M. *Robot Dynamics and Control*, 1st ed. John Wiley & Sons, New York, 1989.

[105] Stephens, B. Humanoid Push Recovery. In *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 589–595.

[106] Stephens, B. *Push Recovery Control for Force-Controlled Humanoid Robots*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania USA, 2011.

[107] Stephens, B. J. Integral Control of Humanoid Balance. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007), pp. 4020–4027.

[108] Stephens, B. J., and Atkeson, C. G. Dynamic Balance Force Control for Compliant Humanoid Robots. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 1248–1255.

[109] Stephens, B. J., and Atkeson, C. G. Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints. In *Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots* (2010), pp. 52–59.

[110] Sugahara, Y., Mikuriya, Y., Hashimoto, K., Hosobata, T., Sunazuka, H., Kawase, M., Lim, H.-o., and Takanishi, A. Walking Control Method of Biped Locomotors on Inclined Plane. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), pp. 1977–1982.

[111] Sugihara, T. Standing Stabilizability and Stepping Maneuver in Planar Bipedalism based on the Best COM-ZMP Regulator. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* (2009), pp. 1966–1971.

[112] Sugihara, T., and Nakamura, Y. A Fast Online Gait Planning with Boundary Condition Relaxation for Humanoid Robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), pp. 305–310.

[113] Sugihara, T., Nakamura, Y., and Inoue, H. Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* (2002), pp. 1404–1409.

[114] Suleiman, W., Kanehiro, F., Miura, K., and Yoshida, E. Improving ZMP-based Control Model Using System Identification Techniques. In *Proceedings of the 9th IEEE-RAS International Workshop on Humanoid Robots* (2009), pp. 74–80.

[115] Takanishi, A., Ishida, M., Yamazaki, Y., and Kato, I. The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD. In *Proceedings of '85 International Conference on Advanced Robotics* (1985), pp. 459–466.

[116] Takanishi, A., Takeya, T., Karaki, H., and Kato, I. A Control Method for Dynamic Biped Walking under Unknown External Force. In *Proceedings of the 1990 IEEE International Workshop on Intelligent Robots and Systems* (1990), pp. 795–801.

[117] Takenaka, T., Motsumoto, T., and Yoshiike, T. Real Time Motion Generation and Control for Biped Robot - 1^{st} Report: Walking Gait Pattern Generation -. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), pp. 1084–1091.

[118] Tasse, S., Kerner, S., Urbann, O., Hofmann, M., and Schwarz, I. Nao Devils Dortmund Team Report. Robotics Research Institute, TU Dortmund University, Germany, 2012.

[119] Tasse, S., and Urbann, O. Observer Based Biped Walking Control, A Sensor Fusion Approach. *Autonomous Robots 35*, 1 (2013), 37–49.

[120] The MathWorks, Inc. SimMechanics: Model and Simulate Multibody Mechanical Systems. Available online. URL: http://www.mathworks.com/products/simmechanics/.

[121] The MathWorks, Inc. What You Can Do with SimMechanics Software. Available online. URL: http://www.mathworks.com/help/physmod/sm/mech/gs/what-you-can-do-with-simmechanics-software.html.

[122] Ulbrich, H., Buschmann, T., and Lohmeier, S. Development of the Humanoid Robot LOLA. *Applied Mechanics and Materials 5-6* (2006), 529–539.

[123] Vukobratović, M., and Borovac, B. Zero-Moment Point-Thirty Five Years of Its Life. *International Journal of Humanoid Robotics 1*, 1 (2004), 157–173.

[124] Vukobratović, M., and Stokić, D. Postural Stability of Anthropomorphic Systems. *Mathematical Biosciences 25* (1975), 217–236.

[125] Winter, D. A. Human Balance and Posture Control during Standing and Walking. *Gait and Posture 3*, 4 (Dec. 1995), 193–214.

[126] Yamaguchi, J., Soga, E., Inoue, S., and Takanishi, A. 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* (1999), pp. 368–374.

[127] Yamaguchi, J., Takanishi, A., and Kato, I. Development of a Biped Walking Robot Compensating for Three-axis Moment by Trunk Motion. In *Proceedings of the*

1993 IEEE International Conference on Intelligent Robots and Systems (1993), pp. 561–566.

[128] Yamaguchi, J., Takanishi, A., and Kato, I. Experimental Development of a Foot Mechanism with Shock Absorbing Material for Acquisition of Landing Surface Position Information and Stabilization of Dynamic Biped Walking. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation* (1995), pp. 2892–2899.

A NAO's Basic Technical Data

Table A.1 summarizes the degrees of freedom and dimensions of NAO H25 v 4.0 which is used as the research platform in the thesis. Figure A.1 shows the detailed dimensions of NAO.

Table A.1: Degrees of freedom and dimensions of NAO.

Head	Neck (pitch & yaw)	2 DoFs
Arm	Shoulder (pitch & roll)	2 DoFs × 2 = 4 DoFs
	Elbow (roll & yaw)	2 DoFs × 2 = 4 DoFs
	Wrist (yaw)	1 DoF × 2 = 2 DoFs
	Hand (open/close)	1 DoF × 2 = 2 DoFs
Torso	Pelvis (yaw/pitch)	1 DoF
Leg	Hip (pitch & roll)	2 DoFs × 2 = 4 DoFs
	Knee (pitch)	1 DoF × 2 = 2 DoFs
	Ankle (pitch & roll)	2 DoFs × 2 = 4 DoFs
Total		25 DoFs
Dimensions (mm)	Height	573
	Width (shoulder to shoulder)	275
	Depth (hand to back (with arm level to ground))	311
	NeckOffsetZ	126.50
	ShoulderOffsetY	98.00
	ElbowOffsetY	15.00
	UpperArmLength	105.00
	LowerArmLength	55.95
	HandOffsetX	57.75
	HandOffsetZ	12.31
	HipOffsetZ	85.00
	HipOffsetY	50.00
	ThighLength	100.00
	TibiaLength	102.90
	FootHeight	45.19
	Length of the sole	160
	Width of the sole	91
Weight (kg)		5.183

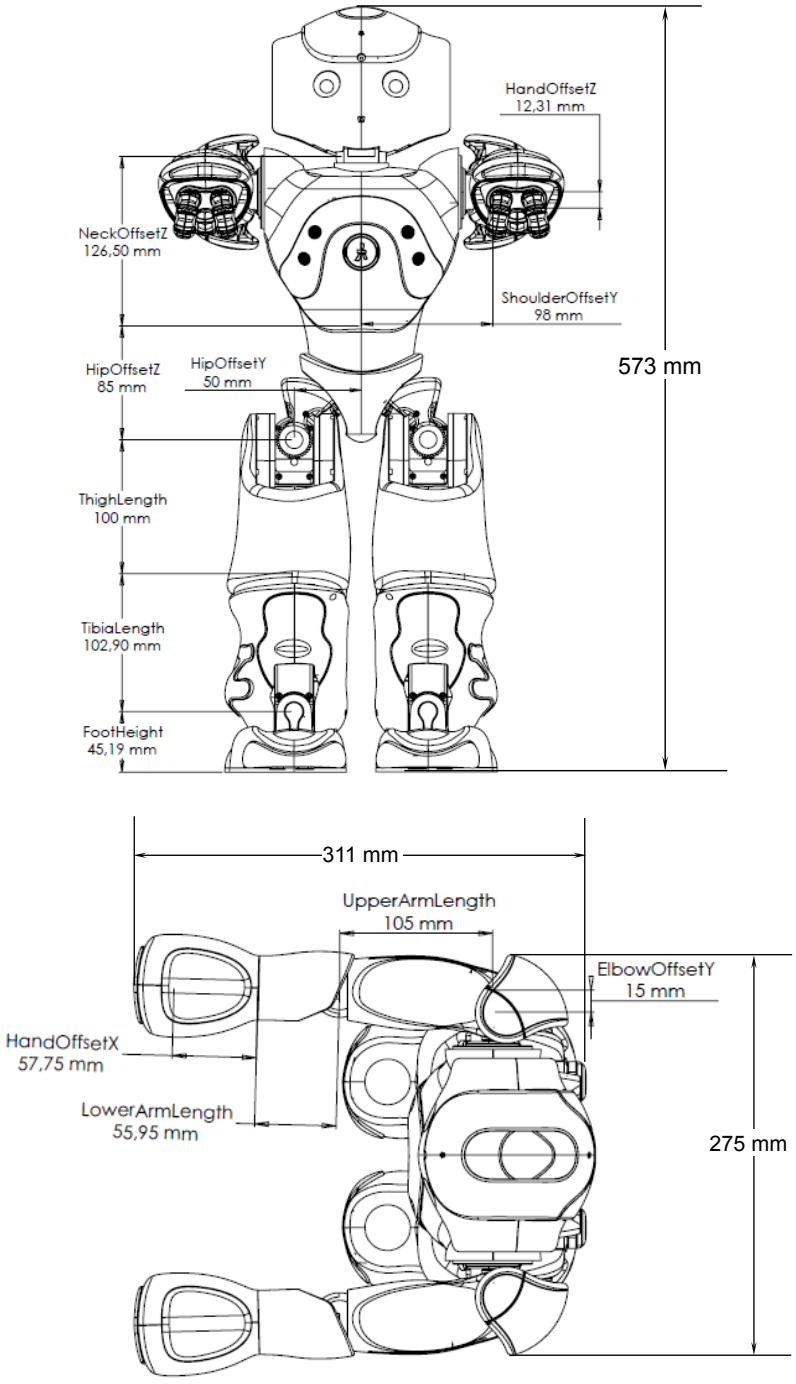


Figure A.1: Length overview of NAO[1].

[1]Sources: http://doc.aldebaran.com/2-1/family/robots/links_robot.html and http://doc.aldebaran.com/2-1/family/robots/dimensions_robot.html, Copyright©: Aldebaran Robotics.

B Forward Kinematics

For the humanoid robot NAO, the conventional base coordinate system is attached either on the robot waist or on the robot foot. We call the former the Robot Coordinate System (RCS), and the latter the Foot Coordinate System (FCS). Figure B.2 shows the conventional coordinate systems of NAO. In this appendix, we derive the position and orientation of the end-effector in the RCS and in the FCS using forward kinematics.

The joint space reveals very little information about the position and orientation of the end-effector of the kinematic chain. The *forward kinematics* defines a mapping from the joint space to the three-dimensional Cartesian space. Given a kinematic chain with m joints and a set of joint angles (θ_1, θ_2, . . . , θ_m), the forward kinematics can find the position (p_x, p_y, p_z) and the orientation (a_x, a_y, a_z) of the end-effector of the kinematic chain in the three-dimensional x-y-z space. Forward kinematics is a domain-independent problem and can be solved for any simple or complex kinematic chain yielding a closed-form, analytical solution.

The importance of solving the forward kinematics problem for NAO is threefold: apart from the ability to locate the exact position and orientation of any end-effector of the robot, it provides the means to calculate the center of mass of the robot for the current configuration, which is indispensible for balancing. It also provides a way of calculating the walking distance or walking speed of the robot in real time by accumulating the distance between two feet in the DSP. In addition, solution to the inverse kinematics problem would be intractable without solving the forward kinematics problem first.

Translation Matrix

Translation in Cartesian space is a function that translates every point by a fixed distance in a specified direction. We can describe a translation in the three-dimensional space with a 4 $\times$ 4 homogeneous transformation matrix of the following form:

$$Trans(d_x, d_y, d_z) = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where d_x, d_y, and d_z define the distance of translation along the x, y and z axis, respectively.

Rotation Matrix

In the three-dimensional Cartesian space there are three distinct rotation matrices, each one of them performing a rotation of θ about the x, y and z axis respectively, assuming a right-handed coordinate system, the rotations can be described in the three-dimensional space with

4×4 homogeneous transformation matrices,

$$Rot(x,\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; Rot(y,\theta) = \begin{pmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; Rot(z,\theta) = \begin{pmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

henceforth $c\theta = cos(\theta)$ and $s\theta = sin(\theta)$.

Denavit-Hartenberg (D-H) Parameters

Denavit and Hartenberg have proposed a way of creating a homogeneous transformation matrix that describes the position and orientation of the i^{th} coordinate frame, which is attached to the joint at one end of the link, in the $(i-1)^{th}$ coordinate frame that is fixed to the joint at the other end of the link, as a function of the joint state [13]. They concluded that this transformation matrix can be fully described using only four parameters, known as Denavit-Hartenberg (D-H) parameters: a_i, α_i, d_i, and θ_i. Before we explain these parameters we first establish the reference frame of each joint with respect to the reference frame of its previous joint (refer to Figure B.1):

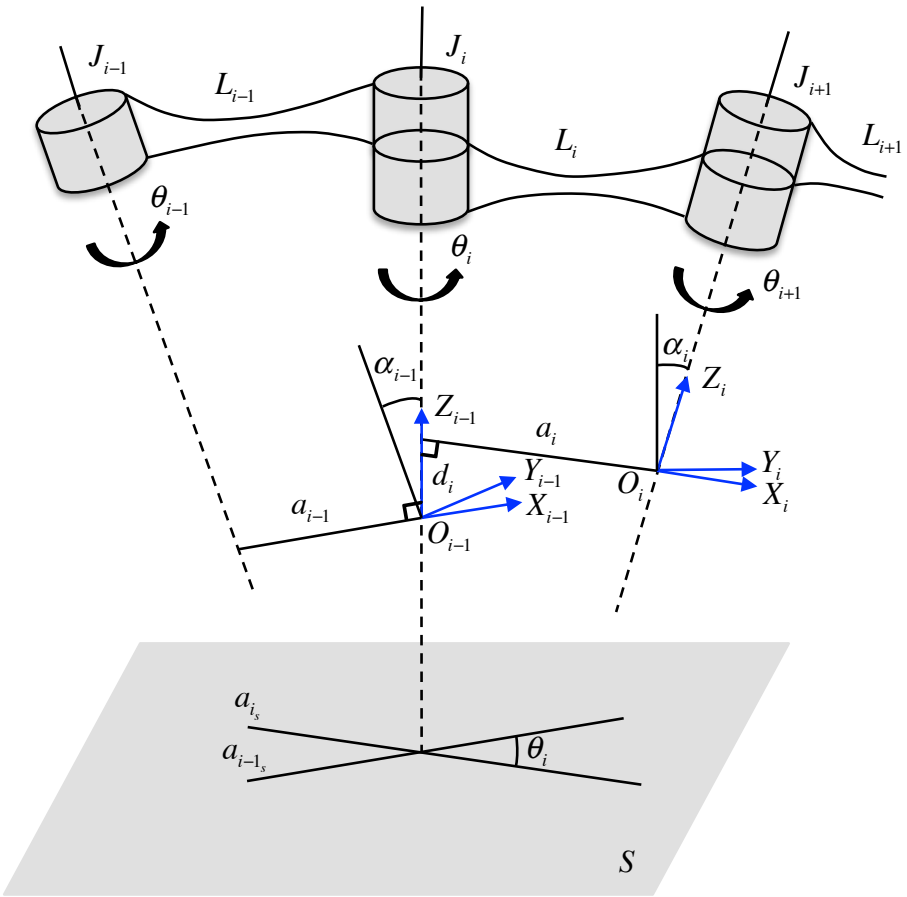


Figure B.1: D-H parameters.

- The origin is located at the intersection of the common perpendicular to axes z_{i-1} and z_i, and joint axis z_i.
- The z_i-axis is set to the direction of the joint axis.
- The x_i-axis lies along the common perpendicular to axes z_{i-1} and z_i, and is oriented from z_{i-1} to z_i.
- The y_i-axis follows from the x_i and z_i axes to form a right-handed coordinate system.

The D-H parameters are described as follows:

- a_i: the length of the common perpendicular to axes z_{i-1} and z_i. a_i length is also known as *link length*.
- α_i: the angle around x_i that makes the vector z_i codirectional with the vector z_{i-1}. The angle is also called *link twist*.
- d_i: the algebraic distance along axis z_{i-1} to the point where the common perpendicular to axis z_i is located. In the bibliography this parameter is also called *link offset*.
- θ_i: the angle around z_i that makes the common perpendicular codirectional with the vector x_{i-1}. The angle around z_i is called *joint angle*.

Now, we can move from the base reference frame of a certain joint to the transformed reference frame of this joint using the transformation matrix ${}^{i-1}T_i$, which consists of two translations and two rotations parametrized by the D-H parameters of the joint:

$$ {}^{i-1}T_i = Rot(x, \alpha_i)Trans(a_i, 0, 0)Rot(z, \theta_i)Trans(0, 0, d_i) $$

The analytical form of the resulting matrix from the above composition is the following

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

The transformation matrix ${}^{i-1}T_i$ can also be formulated as

$$ {}^{i-1}T_i = Rot(z, \theta_i)Trans(0, 0, d_i)Trans(a_i, 0, 0)Rot(x, \alpha_i) $$

Then the corresponding analytical form of the D-H matrix is expressed as follows

$$ {}^{i-1}T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.2} $$

Table B.1: D-H parameters for the head of NAO.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, 0, NeckOffsetZ)			
1	0	0	0	θ_1
2	$-\frac{\pi}{2}$	0	0	$\theta_2 - \frac{\pi}{2}$
Rotation	$Rot(x, \frac{\pi}{2})Rot(y, \frac{\pi}{2})$			

Head Forward Kinematics in the RCS

Table B.1 lists the D-H parameters for the head of NAO. In the table, θ_1 and θ_2 indicate the joint angles of HeadYaw and HeadPitch, respectively. According to (B.1), D-H matrices for the head are

$$ {}^0T_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^1T_2 = \begin{bmatrix} c(\theta_2 - \frac{\pi}{2}) & -s(\theta_2 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(\theta_2 - \frac{\pi}{2}) & -c(\theta_2 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Hence, the forward kinematics of the chain for head in the RCS can be expressed as

$$ {}^{Base}T_{End} = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot R_x(\frac{\pi}{2}) \cdot R_y(\frac{\pi}{2}) $$

Arm Forward Kinematics in the RCS

Table B.2 lists the D-H parameters for both arms. In the table, the value of the symbol 'sign' is prescribed. For the right arm, sign = 1. On the contrary, sign = −1 represents the left arm. θ_1, θ_2, θ_3, θ_4 and θ_5 indicate the joint angles of ShoulderPitch, ShoulderRoll, ElbowYaw, ElbowRoll and WristYaw respectively.

According to (B.1), D-H matrices for the arms of NAO are

$$ {}^0T_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_1 & -c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^1T_2 = \begin{bmatrix} c(\theta_2 + \frac{\pi}{2}) & -s(\theta_2 + \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s(\theta_2 + \frac{\pi}{2}) & c(\theta_2 + \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^2T_3 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & -1 & -d_{ul} \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^3T_4 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^4T_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & -d_{ll} \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Table B.2: D-H parameters for the arms of NAO.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, -sign·(ShoulderOffsetY+ElbowOffsetY), ShoulderOffsetZ)			
1	$-\frac{\pi}{2}$	0	0	θ_1
2	$\frac{\pi}{2}$	0	0	$\theta_2 + \frac{\pi}{2}$
3	$\frac{\pi}{2}$	0	UpperArmLength	θ_3
4	$-\frac{\pi}{2}$	0	0	θ_4
5	$\frac{\pi}{2}$	0	LowerArmLength	θ_5
Rotation	$Rot(y, -\frac{\pi}{2})Rot(x, -\frac{\pi}{2})$			
End-effector	${}^5T_{End}$=Trans(HandOffsetX, 0, 0)			

where d_{ul} = UpperArmLength and d_{ll} = LowerArmLength. The forward kinematics of the chains for both arms in the RCS can be expressed as

$$ {}^{Base}T_{End} = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot R_y(-\frac{\pi}{2}) \cdot R_x(-\frac{\pi}{2}) \cdot {}^5T_{End} $$

Leg Forward Kinematics in the RCS

We numbered the revolute joints of both legs in Table B.3. Table B.4 and Table B.5 list the D-H parameters of both legs in the RCS.

Table B.3: Numbering of the revolute joints for both legs (refer to Figure B.2).

Leg	Joint	Rotation	Angle
Left Leg	Hip	YawPitch	θ_1
		Roll	θ_2
		Pitch	θ_3
	Knee	Pitch	θ_4
	Ankle	Pitch	θ_5
		Roll	θ_6
Right Leg	Hip	YawPitch	θ_7
		Roll	θ_8
		Pitch	θ_9
	Knee	Pitch	θ_{10}
	Ankle	Pitch	θ_{11}
		Roll	θ_{12}

Table B.4: D-H parameters for the left leg of NAO in the RCS.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, HipOffsetY, −HipOffsetZ)			
1	$-\frac{3\pi}{4}$	0	0	$\theta_1 - \frac{\pi}{2}$
2	$-\frac{\pi}{2}$	0	0	$\theta_2 + \frac{\pi}{4}$
3	$\frac{\pi}{2}$	0	0	θ_3
4	0	−ThighLength	0	θ_4
5	0	−TibiaLength	0	θ_5
6	$-\frac{\pi}{2}$	0	0	θ_6
End-effector	${}^{6}T_{End}=Rot(z,\pi)Rot(y,-\frac{\pi}{2})$Trans(0, 0, −FootHeight)			

According to (B.1), D-H matrices for the left leg of the robot NAO in the RCS are

$$
{}^{0}T_1 = \begin{bmatrix} s\theta_1 & c\theta_1 & 0 & 0 \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^{1}T_2 = \begin{bmatrix} c(\theta_2+\frac{\pi}{4}) & -s(\theta_2+\frac{\pi}{4}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(\theta_2+\frac{\pi}{4}) & -c(\theta_2+\frac{\pi}{4}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{2}T_3 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{3}T_4 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_4 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{4}T_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & a_5 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{5}T_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

The forward kinematics of the left leg chain in the RCS can be expressed as

$$
{}^{Base}T_{End} = {}^{Base}T_0 \cdot {}^{0}T_1 \cdot {}^{1}T_2 \cdot {}^{2}T_3 \cdot {}^{3}T_4 \cdot {}^{4}T_5 \cdot {}^{5}T_6 \cdot {}^{6}T_{End}
$$

Table B.5: D-H parameters for the right leg of NAO in the RCS.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, −HipOffsetY, −HipOffsetZ)			
7	$-\frac{\pi}{4}$	0	0	$\theta_7 - \frac{\pi}{2}$
8	$-\frac{\pi}{2}$	0	0	$\theta_8 - \frac{\pi}{4}$
9	$\frac{\pi}{2}$	0	0	θ_9
10	0	−ThighLength	0	θ_{10}
11	0	−TibiaLength	0	θ_{11}
12	$-\frac{\pi}{2}$	0	0	θ_{12}
End-effector	${}^{12}T_{End}$=$Rot(z,\pi)Rot(y,-\frac{\pi}{2})$Trans(0, 0, −FootHeight)			

Similarly, the D-H matrices for the right leg of the robot NAO are

$$
{}^{0}T_7 = \begin{bmatrix} s\theta_7 & c\theta_7 & 0 & 0 \\ -\frac{\sqrt{2}}{2}c\theta_7 & \frac{\sqrt{2}}{2}s\theta_7 & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2}c\theta_7 & -\frac{\sqrt{2}}{2}s\theta_7 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^{7}T_8 = \begin{bmatrix} c(\theta_8 - \frac{\pi}{4}) & -s(\theta_8 - \frac{\pi}{4}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(\theta_8 - \frac{\pi}{4}) & -c(\theta_8 - \frac{\pi}{4}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{8}T_9 = \begin{bmatrix} c\theta_9 & -s\theta_9 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_9 & c\theta_9 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{9}T_{10} = \begin{bmatrix} c\theta_{10} & -s\theta_{10} & 0 & a_{10} \\ s\theta_{10} & c\theta_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{10}T_{11} = \begin{bmatrix} c\theta_{11} & -s\theta_{11} & 0 & a_{11} \\ s\theta_{11} & c\theta_{11} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{11}T_{12} = \begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_{12} & -c\theta_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

The forward kinematics of the right leg chain in the RCS can be expressed as

$$
{}^{Base}T_{End} = {}^{Base}T_0 \cdot {}^{0}T_7 \cdot {}^{7}T_8 \cdot {}^{8}T_9 \cdot {}^{9}T_{10} \cdot {}^{10}T_{11} \cdot {}^{11}T_{12} \cdot {}^{12}T_{End}
$$

Table B.6: D-H parameters for the left leg of NAO in the FCS.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, 0, FootHeight)$Rot(z,\pi)Rot(y,-\frac{\pi}{2})$			
1	0	0	0	θ_6
2	$\frac{\pi}{2}$	0	0	θ_5
3	0	TibiaLength	0	θ_4
4	0	ThighLength	0	θ_3
5	$-\frac{\pi}{2}$	0	0	θ_2
6	$\frac{\pi}{2}$	0	0	$\theta_1-\frac{\pi}{4}$
End-effector	${}^{6}T_{End}$=$Rot(z,\frac{\pi}{2})Rot(x,\frac{3\pi}{4})$Trans(0, −HipOffsetY, HipOffsetZ)			

Leg Forward Kinematics in the FCS

Table B.6 and Table B.7 list the D-H parameters of both legs in the FCS. D-H matrices for the left leg of the robot NAO are

$$
{}^{0}T_1=\begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0\\ s\theta_6 & c\theta_6 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^{1}T_2=\begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0\\ 0 & 0 & -1 & 0\\ s\theta_5 & c\theta_5 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^{2}T_3=\begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_5\\ s\theta_4 & c\theta_4 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},
$$

$$
{}^{3}T_4=\begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_4\\ s\theta_3 & c\theta_3 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^{4}T_5=\begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0\\ 0 & 0 & 1 & 0\\ -s\theta_2 & -c\theta_2 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},
$$

$$
{}^{5}T_6=\begin{bmatrix} c(\theta_1-\frac{\pi}{4}) & 0 & s(\theta_1-\frac{\pi}{4}) & 0\\ s(\theta_1-\frac{\pi}{4}) & 0 & -c(\theta_1-\frac{\pi}{4}) & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}.
$$

The forward kinematics for the left leg chain in the FCS can be expressed as

$$
{}^{Base}T_{End}={}^{Base}T_0\cdot{}^{0}T_1\cdot{}^{1}T_2\cdot{}^{2}T_3\cdot{}^{3}T_4\cdot{}^{4}T_5\cdot{}^{5}T_6\cdot{}^{6}T_{End}
$$

Table B.7: D-H parameters for the right leg of NAO in the FCS.

i	α_{i-1}	a_{i-1}	d_i	θ_i
Base	${}^{Base}T_0$=Trans(0, 0, FootHeight)$Rot(z,\pi)Rot(y,-\frac{\pi}{2})$			
1	0	0	0	θ_{12}
2	$\frac{\pi}{2}$	0	0	θ_{11}
3	0	TibiaLength	0	θ_{10}
4	0	ThighLength	0	θ_9
5	$-\frac{\pi}{2}$	0	0	θ_8
6	$\frac{\pi}{2}$	0	0	$\theta_7+\frac{\pi}{4}$
End-effector	${}^6T_{End}$=$Rot(z,\frac{\pi}{2})Rot(x,\frac{\pi}{4})$Trans(0, HipOffsetY, HipOffsetZ)			

D-H matrices for the right leg of NAO are

$$
{}^0T_7=\begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & 0\\ s\theta_{12} & c\theta_{12} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^7T_8=\begin{bmatrix} c\theta_{11} & -s\theta_{11} & 0 & 0\\ 0 & 0 & -1 & 0\\ s\theta_{11} & c\theta_{11} & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^8T_9=\begin{bmatrix} c\theta_{10} & -s\theta_{10} & 0 & a_{11}\\ s\theta_{10} & c\theta_{10} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},
$$

$$
{}^9T_{10}=\begin{bmatrix} c\theta_9 & -s\theta_9 & 0 & a_{10}\\ s\theta_9 & c\theta_9 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\quad
{}^{10}T_{11}=\begin{bmatrix} c\theta_8 & -s\theta_8 & 0 & 0\\ 0 & 0 & 1 & 0\\ -s\theta_8 & -c\theta_8 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},
$$

$$
{}^{11}T_{12}=\begin{bmatrix} c(\theta_7+\frac{\pi}{4}) & 0 & s(\theta_7+\frac{\pi}{4}) & 0\\ s(\theta_7+\frac{\pi}{4}) & 0 & -c(\theta_7+\frac{\pi}{4}) & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}.
$$

The forward kinematics for the right leg chain in the FCS can be expressed as

$$
{}^{Base}T_{End}={}^{Base}T_0\cdot{}^0T_7\cdot{}^7T_8\cdot{}^8T_9\cdot{}^9T_{10}\cdot{}^{10}T_{11}\cdot{}^{11}T_{12}\cdot{}^{12}T_{End}
$$

Due to the different definitions of the joint rotational direction between the robot NAO and D-H matrix, the signs of the joint data, e.g. LHipRoll joint and LAnkleRoll joint, which is obtained from the joint position sensor (magnetic rotary encoder) should be changed to the opposite when applied to forward kinematics of the leg in the RCS. Besides, since in the FCS the base frame is the foot and the end-effector is the waist, all the signs of the joint angles used for forward kinematics in the FCS should be opposite to those used in the RCS (refer to Table B.8).

Table B.8: Signs of the joint angles for the forward kinematics of the leg in the RCS and FCS.

Forward Kinematics (left leg)		
Joint Data	RCS	FCS
θ_1	θ_1	$-\theta_1$
θ_2	$-\theta_2$	θ_2
θ_3	θ_3	$-\theta_3$
θ_4	θ_4	$-\theta_4$
θ_5	θ_5	$-\theta_5$
θ_6	$-\theta_6$	θ_6

Forward Kinematics (right leg)		
Joint Data	RCS	FCS
θ_7	θ_7	$-\theta_7$
θ_8	θ_8	$-\theta_8$
θ_9	θ_9	$-\theta_9$
θ_{10}	θ_{10}	$-\theta_{10}$
θ_{11}	θ_{11}	$-\theta_{11}$
θ_{12}	θ_{12}	$-\theta_{12}$

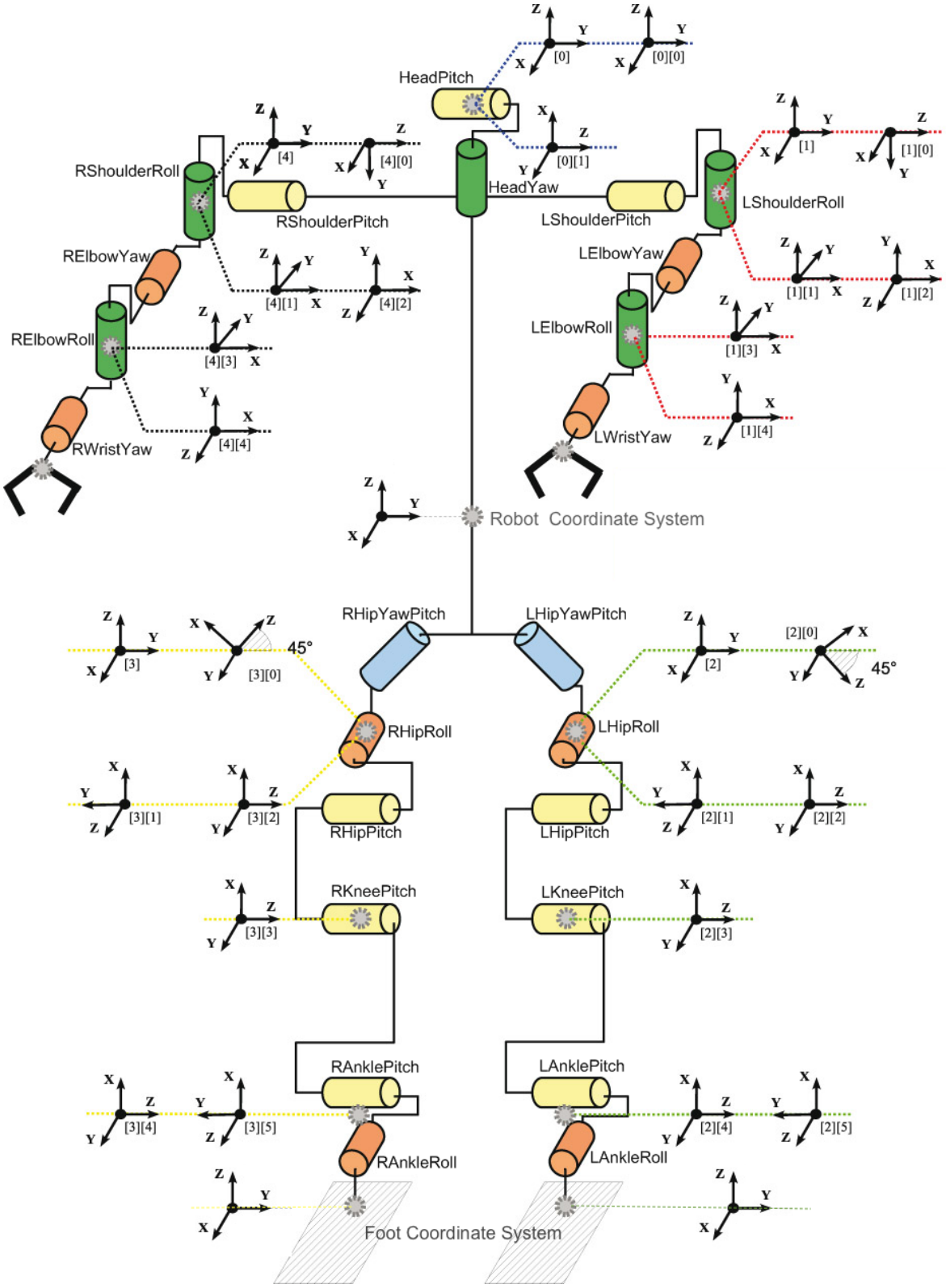


Figure B.2: Conventional coordinate systems.

C Equations of Motion

In this appendix, we derive the equations of motion of the swing-leg, which are used in Chapter 5 for trajectory tracking and parameter estimate of the swing-leg.

We start with the simple planar model to test ideas, then extend to the three-dimensional model.

Planar Model of the Swing-leg

Let us first consider a planar three-link model of the swing-leg comprising the thigh, tibia and foot. Figure C.1 shows the three-link model from the sagittal plane view. We assume that the rest of the robot is not accelerating and hence plays no role in the dynamics. l_1, l_2 and l_3 are respectively the thigh length, the tibia length and the foot height. b_1, b_2 and b_3 are the distances to the center of mass of the links. m_1, m_2, m_3, I_1, I_2 and I_3 are the mass and inertia parameters. θ_1, θ_2 and θ_3 are pitch joint angles of the hip, the knee and the ankle, respectively. θ_1 is the angle of the thigh from vertical; θ_2 is the angle of the tibia with respect to the thigh and θ_3 is the angle of the foot with respect to the thigh.

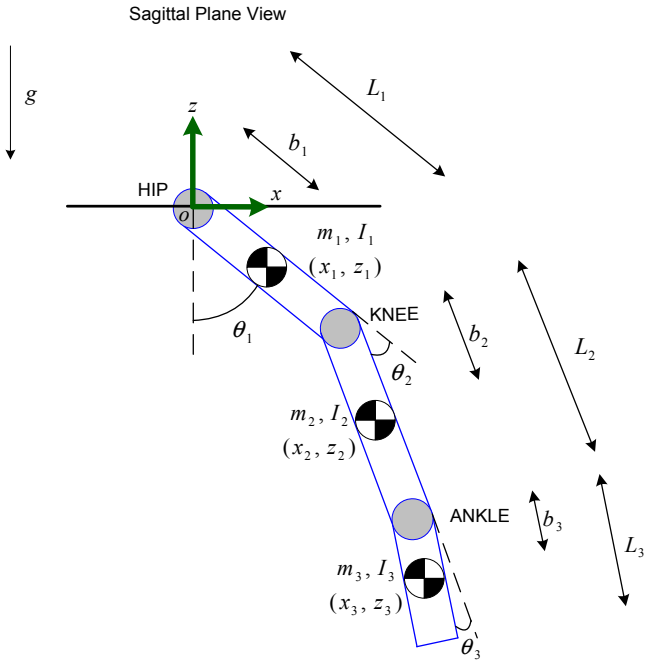


Figure C.1: Dynamic planar model of the swing-leg.

The equations of motion can be computed using Lagrange's method. The kinetic energy, K is

$$\begin{aligned} K &= K_1 + K_2 + K_3 \\ K_1 &= \frac{1}{2} m_1 {v_1}^2 + \frac{1}{2} I_1 {\omega_1}^2 \\ K_2 &= \frac{1}{2} m_2 {v_2}^2 + \frac{1}{2} I_2 {\omega_2}^2 \\ K_3 &= \frac{1}{2} m_3 {v_3}^2 + \frac{1}{2} I_3 {\omega_3}^2 \end{aligned} \tag{C.1}$$

where v_1, v_2 and v_3 are the linear velocities of the center of mass of the upper link (thigh), the lower link (tibia) and the foot link. ω_1 , ω_2 and ω_3 are the angular velocities of the links, all with respect to the fixed global coordinate frame. We have

$${v_1}^2 = (b_1 \dot{\theta}_1)^2 \tag{C.2}$$

$$\begin{aligned} x_2 &= l_1 \sin\theta_1 + b_2 \sin(\theta_1 + \theta_2) \\ z_2 &= -l_1 \cos\theta_1 - b_2 \cos(\theta_1 + \theta_2) \end{aligned} \tag{C.3}$$

$$\begin{aligned} \dot{x_2} &= l_1 \cos(\theta_1)\dot{\theta}_1 + b_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{z_2} &= l_1 \sin(\theta_1)\dot{\theta}_1 + b_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \end{aligned} \tag{C.4}$$

$${v_2}^2 = \dot{x}_2^2 + \dot{z}_2^2 = (l_1\dot{\theta}_1)^2 + (b_2(\dot{\theta}_1 + \dot{\theta}_2))^2 + 2l_1\dot{\theta}_1 b_2(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 \tag{C.5}$$

$$\begin{aligned} x_3 &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + b_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ z_3 &= -l_1 \cos\theta_1 - l_2 \cos(\theta_1 + \theta_2) - b_3 \cos(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{C.6}$$

$$\begin{aligned} \dot{x_3} &= l_1 \cos(\theta_1)\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) + b_3 \cos(\theta_1 + \theta_2 + \theta_3)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ \dot{z_3} &= l_1 \sin(\theta_1)\dot{\theta}_1 + l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) + b_3 \sin(\theta_1 + \theta_2 + \theta_3)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \end{aligned} \tag{C.7}$$

$$\begin{aligned} {v_3}^2 &= \dot{x}_3^2 + \dot{z}_3^2 \\ &= (l_1\dot{\theta}_1)^2 + (l_2(\dot{\theta}_1 + \dot{\theta}_2))^2 + (b_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3))^2 + 2l_1\dot{\theta}_1 l_2(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 \\ &\quad + 2l_1\dot{\theta}_1 b_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\cos(\theta_2 + \theta_3) + 2l_2(\dot{\theta}_1 + \dot{\theta}_2)b_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\cos(\theta_3) \end{aligned} \tag{C.8}$$

$$\begin{aligned} \omega_1^2 &= \dot{\theta}_1^2 \\ \omega_2^2 &= (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ \omega_3^2 &= (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 \end{aligned} \tag{C.9}$$

The potential energy P is

$$\begin{aligned}
P &= P_1 + P_2 + P_3 \\
P_1 &= m_1 g z_1 = -m_1 g b_1 \cos\theta_1 \\
P_2 &= m_2 g z_2 = -m_2 g (l_1 \cos\theta_1 + b_2 \cos(\theta_1 + \theta_2)) \\
P_3 &= m_3 g z_3 = -m_3 g (l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + b_3 \cos(\theta_1 + \theta_2 + \theta_3))
\end{aligned} \tag{C.10}$$

We can simplify the equations by substituting the following inertial groups:

$$\begin{aligned}
J_1 &= m_1 b_1^2 + I_1 + m_2 l_1^2 + m_3 l_1^2 \\
J_2 &= m_2 l_1 b_2 + m_3 l_1 l_2 \\
J_3 &= m_2 b_2^2 + I_2 + m_3 l_2^2 \\
J_4 &= m_3 l_2 b_3 \\
J_5 &= m_3 b_3^2 + I_3 \\
J_6 &= m_3 l_1 b_3 \\
G_1 &= (m_1 b_1 + m_2 l_1 + m_3 l_1) g \\
G_2 &= (m_2 b_2 + m_3 l_2) g \\
G_3 &= m_3 b_3 g
\end{aligned}$$

The kinetic energy and the gravitational potential energy can now be written as

$$\begin{aligned}
K = {} & (\frac{1}{2} J_1 + J_2 \cos\theta_2 + \frac{1}{2} J_3 + J_4 \cos\theta_3 + \frac{1}{2} J_5 + J_6 \cos(\theta_2 + \theta_3)) \dot{\theta}_1^2 \\
& + (\frac{1}{2} J_3 + J_4 \cos\theta_3 + \frac{1}{2} J_5) \dot{\theta}_2^2 + \frac{1}{2} J_5 \dot{\theta}_3^2 \\
& + (J_2 \cos\theta_2 + J_3 + 2 J_4 \cos\theta_3 + J_5 + J_6 \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \dot{\theta}_2 \\
& + (J_4 \cos\theta_3 + J_5) \dot{\theta}_2 \dot{\theta}_3 + (J_4 \cos\theta_3 + J_5 + J_6 \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \dot{\theta}_3
\end{aligned}$$

$$P = -G_1 \cos\theta_1 - G_2 \cos(\theta_1 + \theta_2) - G_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

The Lagrangian is $L = K - P$. Lagranges equations are

$$\frac{\mathrm{d}}{\mathrm{d}t} \left[\frac{\partial L}{\partial \dot{\theta}} \right] - \frac{\partial L}{\partial \theta} = \tau \tag{C.11}$$

Let $q = [q_1 \quad q_2 \quad q_3]^T$ be the joint angles of the robot NAO, then $q_1 = \theta_1$, $q_2 = \theta_2$ and $q_3 = \theta_3$.

Using Lagrange's method yields the equations of motion in the following form:

$$H(q)\ddot{q} + b(q, \dot{q}) + \vec{g}(q) = \tau \tag{C.12}$$

where $\tau = [\tau_1, \ \tau_2, \ \tau_3]^T$ are the applied joint torques, $H(q) \in \mathbb{R}^{3\times3}$ is the symmetric inertia matrix, $b(q, \dot{q}) \in \mathbb{R}^3$ represents the centrifugal and Coriolis terms, and $\vec{g}(q) \in \mathbb{R}^3$ is the gravity term.

The calculated terms in the matrices are as follows:

$$
\begin{aligned}
H_{11} &= J_1 + 2J_2\cos(q_2) + J_3 + 2J_4\cos(q_3) + J_5 + 2J_6\cos(q_2+q_3)\\
H_{12} &= J_2\cos(q_2) + J_3 + 2J_4\cos(q_3) + J_5 + J_6\cos(q_2+q_3)\\
H_{13} &= J_4\cos(q_3) + J_5 + J_6\cos(q_2+q_3)\\
H_{21} &= J_2\cos(q_2) + J_3 + 2J_4\cos(q_3) + J_5 + J_6\cos(q_2+q_3)\\
H_{22} &= J_3 + 2J_4\cos(q_3) + J_5\\
H_{23} &= J_4\cos(q_3) + J_5\\
H_{31} &= J_4\cos(q_3) + J_5 + J_6\cos(q_2+q_3)\\
H_{32} &= J_4\cos(q_3) + J_5\\
H_{33} &= J_5
\end{aligned}
$$

$$
\begin{aligned}
b_{11} =& -(2J_2\sin(q_2) + 2J_6\sin(q_2+q_3))\dot{q}_1\dot{q}_2 - (2J_4\sin(q_3) + 2J_6\sin(q_2+q_3))\dot{q}_2\dot{q}_3\\
&-(2J_4\sin(q_3) + 2J_6\sin(q_2+q_3))\dot{q}_1\dot{q}_3 - (J_2\sin(q_2) + J_6\sin(q_2+q_3))\dot{q}_2^2\\
&-(J_4\sin(q_3) + J_6\sin(q_2+q_3))\dot{q}_3^2\\
b_{21} =& (J_2\sin(q_2) + J_6\sin(q_2+q_3))\dot{q}_1^2 - J_4\sin(q_3)\dot{q}_3^2 - 2J_4\sin(q_3)\dot{q}_1\dot{q}_3\\
&-2J_4\sin(q_3)\dot{q}_2\dot{q}_3\\
b_{31} =& (J_4\sin(q_3) + J_6\sin(q_2+q_3))\dot{q}_1^2 + J_4\sin(q_3)\dot{q}_2^2 + 2J_4\sin(q_3)\dot{q}_1\dot{q}_2
\end{aligned}
$$

$$
\begin{aligned}
g_{11} &= G_1\sin(q_1) + G_2\sin(q_1+q_2) + G_3\sin(q_1+q_2+q_3)\\
g_{21} &= G_2\sin(q_1+q_2) + G_3\sin(q_1+q_2+q_3)\\
g_{31} &= G_3\sin(q_1+q_2+q_3)
\end{aligned}
$$

Let

$$b(q,\dot{q}) = C(q,\dot{q})\dot{q} \tag{C.13}$$

There are many choices of C that can realize the decomposition in (C.13). If a particular choice of matrix C is defined by

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{3}\frac{\partial H_{ij}}{\partial q_k}\dot{q}_k + \frac{1}{2}\sum_{k=1}^{3}(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i})\dot{q}_k \tag{C.14}$$

then the system dynamics in the sagittal plane can be rewritten as

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \vec{g}(q) = \tau \tag{C.15}$$

where

$$
\begin{aligned}
C_{11} &= (-J_2\sin(q_2) - J_6\sin(q_2+q_3))\dot{q}_2 + (-J_4\sin(q_3) - J_6\sin(q_2+q_3))\dot{q}_3 \\
C_{12} &= (-J_2\sin(q_2) - J_6\sin(q_2+q_3))\dot{q}_1 + (-J_2\sin(q_2) - J_6\sin(q_2+q_3))\dot{q}_2 \\
&\quad +(-J_4\sin(q_3) - J_6\sin(q_2+q_3))\dot{q}_3 \\
C_{13} &= (-J_4\sin(q_3) - J_6\sin(q_2+q_3))(\dot{q}_1+\dot{q}_2+\dot{q}_3) \\
C_{21} &= (J_2\sin(q_2) + J_6\sin(q_2+q_3))\dot{q}_1 - J_4\sin(q_3)\dot{q}_3 \\
C_{22} &= -J_4\sin(q_3)\dot{q}_3 \\
C_{23} &= -J_4\sin(q_3)(\dot{q}_1+\dot{q}_2+\dot{q}_3) \\
C_{31} &= (J_4\sin(q_3) + J_6\sin(q_2+q_3))\dot{q}_1 + J_4\sin(q_3)\dot{q}_2 \\
C_{32} &= J_4\sin(q_3)(\dot{q}_1+\dot{q}_2) \\
C_{33} &= 0
\end{aligned}
$$

With this particular definition of C (see Eq.(C.14)), the matrix $\dot{H}-2C$ is skew-symmetric. According to the definition of the skew-symmetric matrix, it means $\dot{H}-2C = -(\dot{H}-2C)^T$. Since H, and therefore $\dot{H}$, are symmetric matrices, the skew-symmetry of the matrix $\dot{H}-2C$ can also be written as $\dot{H} = C + C^T$.

Let us define $J_1 = a_1, J_2 = a_2, J_3 = a_3, J_4 = a_4, J_5 = a_5, J_6 = a_6, G_1 = a_7, G_2 = a_8$ and $G_3 = a_9$, then the vector of unknown parameters $a = [a_1 \quad a_2 \quad a_3 \cdots a_9]^T$.

The 3×9 matrix Y can be obtained according to Eq.(5.4).

$$
Y = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{19} \\ Y_{21} & Y_{22} & \cdots & Y_{29} \\ Y_{31} & Y_{32} & \cdots & Y_{39} \end{pmatrix}
$$

in which

$Y_{11} = \ddot{q}_{r1}$, $Y_{12} = \cos(q_2)(2\ddot{q}_{r1}+\ddot{q}_{r2}) - \sin(q_2)(\dot{q}_2\dot{q}_{r1}+\dot{q}_1\dot{q}_{r2}+\dot{q}_2\dot{q}_{r2})$, $Y_{13} = \ddot{q}_{r1}+\ddot{q}_{r2}$,
$Y_{14} = \cos(q_3)(2\ddot{q}_{r1}+2\ddot{q}_{r2}+\ddot{q}_{r3}) - \sin(q_3)\dot{q}_3(\dot{q}_{r1}+\dot{q}_{r2}) - \sin(q_3)(\dot{q}_1+\dot{q}_2+\dot{q}_3)\dot{q}_{r3}$,
$Y_{15} = \ddot{q}_{r1}+\ddot{q}_{r2}+\ddot{q}_{r3}$,
$Y_{16} = \cos(q_2+q_3)(2\ddot{q}_{r1}+\ddot{q}_{r2}+\ddot{q}_{r3}) - \sin(q_2+q_3)(\dot{q}_2+\dot{q}_3)\dot{q}_{r1} - \sin(q_2+q_3)(\dot{q}_1+\dot{q}_2+\dot{q}_3)(\dot{q}_{r2}+\dot{q}_{r3})$,
$Y_{17} = \sin(q_1)$, $Y_{18} = \sin(q_1+q_2)$, $Y_{19} = \sin(q_1+q_2+q_3)$,
$Y_{21} = 0$, $Y_{22} = \cos(q_2)\ddot{q}_{r1} + \sin(q_2)\dot{q}_1\dot{q}_{r1}$, $Y_{23} = \ddot{q}_{r1}+\ddot{q}_{r2}$,
$Y_{24} = \cos(q_3)(2\ddot{q}_{r1}+2\ddot{q}_{r2}+\ddot{q}_{r3}) - \sin(q_3)\dot{q}_3(\dot{q}_{r1}+\dot{q}_{r2}) - \sin(q_3)(\dot{q}_1+\dot{q}_2+\dot{q}_3)\dot{q}_{r3}$,
$Y_{25} = \ddot{q}_{r1}+\ddot{q}_{r2}+\ddot{q}_{r3}$, $Y_{26} = \cos(q_2+q_3)\ddot{q}_{r1} + \sin(q_2+q_3)\dot{q}_1\dot{q}_{r1}$,
$Y_{27} = 0$, $Y_{28} = \sin(q_1+q_2)$, $Y_{29} = \sin(q_1+q_2+q_3)$, $Y_{31} = Y_{32} = Y_{33} = 0$,
$Y_{34} = \cos(q_3)(\ddot{q}_{r1}+\ddot{q}_{r2}) + \sin(q_3)(\dot{q}_1+\dot{q}_2)(\dot{q}_{r1}+\dot{q}_{r2})$, $Y_{35} = \ddot{q}_{r1}+\ddot{q}_{r2}+\ddot{q}_{r3}$,
$Y_{36} = \cos(q_2+q_3)\ddot{q}_{r1} + \sin(q_2+q_3)\dot{q}_1\dot{q}_{r1}$, $Y_{37} = Y_{38} = 0$, $Y_{39} = \sin(q_1+q_2+q_3)$.

Then the adaptive control can be implemented on the planar model for trajectory tracking and parameter estimate.

3D Model of the Swing-leg

Next, we consider the three-link model of the swing-leg in three dimensions, as shown in Figure C.2. Apart from the dynamics in the sagittal plane, dynamics in the frontal plane are added. Since the knee joint cannot roll due to the mechanical design, α_1 and α_2 are used to

indicate the roll angles of the hip joint and the ankle joint, respectively. α_1 is the angle of the thigh from vertical; α_2 is the angle of the foot with respect to the thigh and tibia. I_{xx_1}, I_{xx_2}, I_{xx_3} represent the moments of inertia of the thigh, the tibia and the foot about the x-axis respectively, and I_{yy_1}, I_{yy_2}, I_{yy_3} are the moments of inertia of the thigh, the tibia and the foot about the y-axis, respectively. The meanings of the rest symbols are the same as those defined on the planar model.

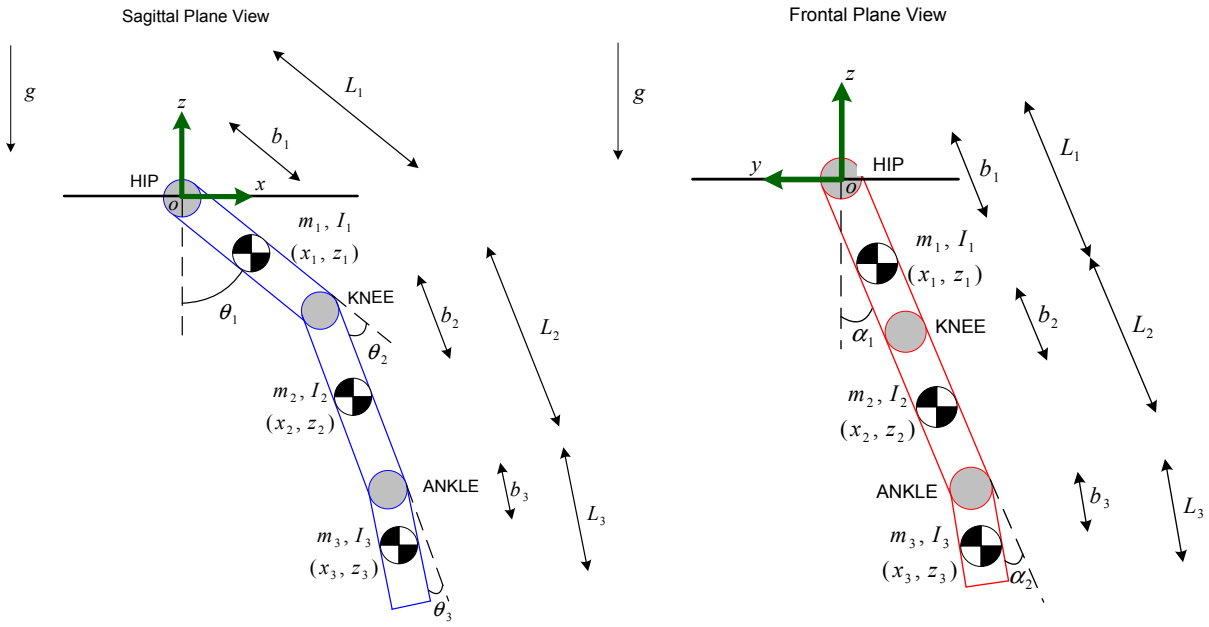


Figure C.2: Dynamic three-dimensional model of the swing-leg.

$$
\begin{aligned}
x_1 &= b_1 \cos\alpha_1 \sin\theta_1 \\
y_1 &= b_1 \sin\alpha_1 \\
z_1 &= -b_1 \cos\alpha_1 \cos\theta_1
\end{aligned}
\tag{C.16}
$$

$$
v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2 = b_1^2\left[(\dot{\alpha}_1)^2 + (\cos\alpha_1)^2(\dot{\theta}_1)^2\right] \tag{C.17}
$$

$$
\begin{aligned}
x_2 &= l_1 \cos\alpha_1 \sin\theta_1 + b_2 \sin(\theta_1+\theta_2)\cos\alpha_1 \\
y_2 &= l_1 \sin\alpha_1 + b_2 \sin\alpha_1 \\
z_2 &= -l_1 \cos\alpha_1 \cos\theta_1 - b_2 \cos(\theta_1+\theta_2)\cos\alpha_1
\end{aligned}
\tag{C.18}
$$

$$
\begin{aligned}
v_2^2 &= \dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2 \\
&= l_1^2(\dot{\alpha}_1)^2 + l_1^2(\cos\alpha_1)^2(\dot{\theta}_1)^2 + b_2^2(\dot{\theta}_1+\dot{\theta}_2)^2(\cos\alpha_1)^2 + b_2^2(\dot{\alpha}_1)^2 \\
&\quad + 2l_1b_2(\cos\alpha_1)^2(\dot{\alpha}_1)^2 + 2l_1b_2\dot{\alpha}_1\dot{\theta}_2 \sin\alpha_1 \cos\alpha_1 \sin\theta_2 \\
&\quad + 2l_1b_2(\cos\alpha_1)^2(\dot{\theta}_1)(\dot{\theta}_1+\dot{\theta}_2)\cos\theta_2 + 2l_1b_2(\sin\alpha_1)^2(\dot{\alpha}_1)^2\cos\theta_2
\end{aligned}
\tag{C.19}
$$

$$
\begin{aligned}
x_3 &= l_1 \cos\alpha_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2)\cos\alpha_1 + b_3 \cos(\alpha_1+\alpha_2)\sin(\theta_1+\theta_2+\theta_3)\\
y_3 &= (l_1+l_2)\sin\alpha_1 + b_3 \sin(\alpha_1+\alpha_2)\\
z_3 &= -l_1 \cos\alpha_1 \cos\theta_1 - l_2 \cos(\theta_1+\theta_2)\cos\alpha_1 - b_3 \cos(\alpha_1+\alpha_2)\cos(\theta_1+\theta_2+\theta_3)
\end{aligned}
\tag{C.20}
$$

$$
\begin{aligned}
v_3^2 &= \dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2\\
&= l_1^2(\dot{\alpha}_1)^2 + l_2^2(\dot{\alpha}_1)^2 + l_1^2(\cos\alpha_1)^2(\dot{\theta}_1)^2 + l_2^2(\dot{\theta}_1+\dot{\theta}_2)^2(\cos\alpha_1)^2 +\\
&\quad b_3^2\big(\cos(\alpha_1+\alpha_2)\big)^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)^2 + 2l_1l_2(\cos\alpha_1)^2(\dot{\alpha}_1)^2 + b_3^2(\dot{\alpha}_1+\dot{\alpha}_2)^2 +\\
&\quad 2l_1l_2(\sin\alpha_1\sin\theta_2\cos\alpha_1)\dot{\alpha}_1\dot{\theta}_2 + 2l_1l_2(\dot{\alpha}_1)^2(\sin\alpha_1)^2\cos\theta_2 +\\
&\quad 2l_1b_3(\sin\alpha_1)\dot{\alpha}_1\sin(\alpha_1+\alpha_2)(\dot{\alpha}_1+\dot{\alpha}_2)\cos(\theta_2+\theta_3) +\\
&\quad 2l_1b_3(\sin\alpha_1)\dot{\alpha}_1\cos(\alpha_1+\alpha_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)\sin(\theta_2+\theta_3) +\\
&\quad 2l_1l_2(\cos\alpha_1)^2\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2)\cos\theta_2 -\\
&\quad 2l_1b_3(\cos\alpha_1)\dot{\theta}_1\sin(\alpha_1+\alpha_2)(\dot{\alpha}_1+\dot{\alpha}_2)\sin(\theta_1+\theta_2) +\\
&\quad 2l_1b_3(\cos\alpha_1)\dot{\theta}_1\cos(\alpha_1+\alpha_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)\cos(\theta_2+\theta_3) -\\
&\quad 2l_2b_3(\dot{\theta}_1+\dot{\theta}_2)\cos\alpha_1\sin(\alpha_1+\alpha_2)(\dot{\alpha}_1+\dot{\alpha}_2)\sin(\theta_3) +\\
&\quad 2l_2b_3\cos\alpha_1\cos\theta_3\cos(\alpha_1+\alpha_2)(\dot{\theta}_1+\dot{\theta}_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) +\\
&\quad 2l_2b_3\sin\alpha_1\cos\theta_3\sin(\alpha_1+\alpha_2)\dot{\alpha}_1(\dot{\alpha}_1+\dot{\alpha}_2) +\\
&\quad 2l_2b_3\sin\alpha_1\sin\theta_3\cos(\alpha_1+\alpha_2)\dot{\alpha}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) +\\
&\quad 2(l_1+l_2)b_3\cos\alpha_1\cos(\alpha_1+\alpha_2)\dot{\alpha}_1(\dot{\alpha}_1+\dot{\alpha}_2)
\end{aligned}
\tag{C.21}
$$

The kinetic energy, K is

$$
\begin{aligned}
K &= K_1 + K_2 + K_3\\
K_1 &= \frac{1}{2}m_1{v_1}^2 + \frac{1}{2}I_{yy_1}\dot{\theta}_1^2 + \frac{1}{2}I_{xx_1}\dot{\alpha}_1^2\\
K_2 &= \frac{1}{2}m_2{v_2}^2 + \frac{1}{2}I_{yy_2}(\dot{\theta}_1+\dot{\theta}_2)^2 + \frac{1}{2}I_{xx_2}\dot{\alpha}_1^2\\
K_3 &= \frac{1}{2}m_3{v_3}^2 + \frac{1}{2}I_{yy_3}(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)^2 + \frac{1}{2}I_{xx_3}(\dot{\alpha}_1+\dot{\alpha}_2)^2
\end{aligned}
\tag{C.22}
$$

The potential energy P is

$$
\begin{aligned}
P &= P_1 + P_2 + P_3\\
P_1 &= -m_1 g b_1 \cos\alpha_1 \cos\theta_1\\
P_2 &= -m_2 g\big(l_1\cos\alpha_1\cos\theta_1 + b_2\cos\alpha_1\cos(\theta_1+\theta_2)\big)\\
P_3 &= -m_3 g\big(l_1\cos\alpha_1\cos\theta_1 + l_2\cos\alpha_1\cos(\theta_1+\theta_2) + b_3\cos(\alpha_1+\alpha_2)\cos(\theta_1+\theta_2+\theta_3)\big)
\end{aligned}
\tag{C.23}
$$

We can simplify the equations by substituting the following inertial groups:

$$
\begin{aligned}
&J_1 = m_1 b_1^2, && J_2 = m_2 b_2^2, && J_3 = m_2 l_1^2\\
&J_4 = m_3 b_3^2, && J_5 = m_3 l_1^2, && J_6 = m_3 l_2^2\\
&J_7 = m_3 b_3 l_1, && J_8 = m_3 b_3 l_2, && J_9 = m_2 l_1 b_2 + m_3 l_1 l_2\\
&J_{10} = I_{xx_1} + I_{xx_2}, && J_{11} = I_{xx_3}, && J_{12} = I_{yy_1}\\
&J_{13} = I_{yy_2}, && J_{14} = I_{yy_3}, && J_{15} = (m_1 b_1 + m_2 l_1 + m_3 l_1) g\\
&J_{16} = (m_2 b_2 + m_3 l_2) g, && J_{17} = m_3 b_3 g
\end{aligned}
$$

The kinetic energy and the gravitational potential energy can now be written as

$$
\begin{aligned}
K = {} & \big[\frac{1}{2}(J_{12} + J_{13} + J_{14}) + \frac{1}{2}(J_2 + J_3)(\cos\alpha_1)^2 + J_9(\cos\alpha_1)^2\cos(\theta_2) + \\
& \frac{1}{2}J_4(\cos(\alpha_1+\alpha_2))^2 + \frac{1}{2}(J_5 + J_6)(\cos\alpha_1)^2 + J_7\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_2+\theta_3) + \\
& J_8\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_3) + \frac{1}{2}J_1(\cos\alpha_1)^2\big]\dot\theta_1^2 + \\
& \big[\frac{1}{2}(J_{13} + J_{14}) + \frac{1}{2}J_4(\cos(\alpha_1+\alpha_2))^2 + \frac{1}{2}(J_2 + J_6)(\cos\alpha_1)^2 + \\
& J_8\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_3)\big]\dot\theta_2^2 + \big[\frac{1}{2}J_{14} + \frac{1}{2}J_4(\cos(\alpha_1+\alpha_2))^2\big]\dot\theta_3^2 + \\
& \big[J_{13} + J_{14} + J_4(\cos(\alpha_1+\alpha_2))^2 + J_7\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_2+\theta_3) + \\
& 2J_8\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_3) + J_9(\cos\alpha_1)^2\cos(\theta_2) + (J_2 + J_6)(\cos\alpha_1)^2\big]\dot\theta_1\dot\theta_2 + \\
& \big[J_{14} + J_4(\cos(\alpha_1+\alpha_2))^2 + \cos(\alpha_1+\alpha_2)\cos(\alpha_1)(J_7\cos(\theta_2+\theta_3) + J_8\cos(\theta_3))\big]\dot\theta_1\dot\theta_3 + \\
& \big[J_{14} + J_4(\cos(\alpha_1+\alpha_2))^2 + J_8\cos(\alpha_1+\alpha_2)\cos(\alpha_1)\cos(\theta_3)\big]\dot\theta_2\dot\theta_3 - \\
& \big[J_7\sin(\alpha_2)\sin(\theta_2+\theta_3) + J_8\sin(\alpha_2)\sin(\theta_3)\big]\dot\theta_1\dot\alpha_1 - \\
& \big[\sin(\alpha_1+\alpha_2)\cos(\alpha_1)(J_7\sin(\theta_2+\theta_3) + J_8\sin(\theta_3))\big]\dot\theta_1\dot\alpha_2 + \\
& \big[J_7\cos(\alpha_1+\alpha_2)\sin(\alpha_1)\sin(\theta_2+\theta_3) - J_8\sin(\theta_3)\sin(\alpha_2) + \\
& J_9\cos(\alpha_1)\sin(\alpha_1)\sin(\theta_2)\big]\dot\alpha_1\dot\theta_2 - \big[J_8\sin(\alpha_1+\alpha_2)\cos(\alpha_1)\sin(\theta_3)\big]\dot\theta_2\dot\alpha_2 + \\
& \big[\cos(\alpha_1+\alpha_2)\sin(\alpha_1)(J_7\sin(\theta_2+\theta_3) + J_8\sin(\theta_3))\big]\dot\alpha_1\dot\theta_3 + \\
& \big[J_4 + J_{11} + (J_7 + J_8)\cos(\alpha_1+\alpha_2)\cos(\alpha_1) + J_7\sin(\alpha_1)\sin(\alpha_1+\alpha_2)\cos(\theta_2+\theta_3) + \\
& J_8\sin(\alpha_1)\sin(\alpha_1+\alpha_2)\cos(\theta_3)\big]\dot\alpha_1\dot\alpha_2 + \\
& \big[\frac{1}{2}(J_2 + J_3 + J_4 + J_5 + J_6 + J_{10} + J_{11}) + J_9(\cos\alpha_1)^2 + J_9(\sin\alpha_1)^2\cos(\theta_2) + \\
& J_7\sin(\alpha_1+\alpha_2)\sin\alpha_1\cos(\theta_2+\theta_3) + J_8\sin(\alpha_1+\alpha_2)\sin\alpha_1\cos\theta_3 + \\
& (J_7 + J_8)\cos(\alpha_1+\alpha_2)\cos\alpha_1 + \frac{1}{2}J_1\big]\dot\alpha_1^2 + \\
& \frac{1}{2}(J_4 + J_{11})\dot\alpha_2^2
\end{aligned}
$$

$$
P = -J_{15}\cos(\alpha_1)\cos(\theta_1) - J_{16}\cos(\theta_1+\theta_2)\cos(\alpha_1) - J_{17}\cos(\alpha_1+\alpha_2)\cos(\theta_1+\theta_2+\theta_3)
$$

The Lagrangian is $L = K - P$. Let $q = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5]^T$ be the joint angles of the leg of the robot NAO, then $q_1 = \alpha_1$, $q_2 = \theta_1$, $q_3 = \theta_2$, $q_4 = \theta_3$ and $q_5 = \alpha_2$. Thus Lagranges equations can be written in the form

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \vec{g}(q) = \tau \tag{C.24}$$

According to the method in [104], the terms in the matrices H, C and g can be calculated. Appendix D introduces an algorithm for calculating the terms in the matrices H, C and g using Matlab.

Let us define $J_1 = a_1, J_2 = a_2, J_3 = a_3, J_4 = a_4, J_5 = a_5, J_6 = a_6, J_7 = a_7, J_8 = a_8, J_9 = a_9, J_{10} = a_{10}, J_{11} = a_{11}, J_{12} = a_{12}, J_{13} = a_{13}, J_{14} = a_{14}, J_{15} = a_{15}, J_{16} = a_{16}$, and $J_{17} = a_{17}$, then $a = [a_1 \quad a_2 \quad a_3 \cdots a_{17}]^T$. The 5×17 matrix Y can be obtained which has the following form.

$$Y = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{116} & Y_{117} \\ Y_{21} & Y_{22} & \cdots & Y_{216} & Y_{217} \\ Y_{31} & Y_{32} & \cdots & Y_{316} & Y_{317} \\ Y_{41} & Y_{42} & \cdots & Y_{416} & Y_{417} \\ Y_{51} & Y_{52} & \cdots & Y_{516} & Y_{517} \end{pmatrix}$$

Having the equations of motion as well as the matrix Y, we can apply the adaptive control to the 3D model of the swing-leg for trajectory tracking and parameter estimate.

D Jacobian Matrix

In this appendix, we derive the Jacobian matrix, which is used in Chapter 6 for compliant force control.

Jacobian matrix describes a relationship between joint angle rate of change in the joint space and the translational and rotational velocities of the end-effector in Cartesian space.

$$\dot{x} = J(\theta)\dot{\theta} \tag{D.1}$$

In (D.1), $J(\theta)$ is the Jacobian matrix, $\dot{x} = [\boldsymbol{v}, \boldsymbol{\omega}]^T$ is the linear translational and rotational velocity of the end-effector, $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \cdots, \dot{\theta}_n]$ is the vector of joint angular velocity, and n is the number of the joints.

Jacobian matrix also provides a relationship between joint torques and the resultant force and torque applied by the end-effector.

$$\tau = J^T(\theta)\mathbf{F} \tag{D.2}$$

where $\boldsymbol{F} = [\boldsymbol{f}, \boldsymbol{n}]^T$ is the generalized force vector, $\boldsymbol{f}$ and $\boldsymbol{n}$ are the force and torque vectors in Cartesian space respectively, $\boldsymbol{\tau} = [\tau_1, \tau_2, \cdots, \tau_n]^T$ are the applied force or torque vector in the joint space, and n is the number of the joints.

Expressed in the Base Coordinate System

If the upper and lower halves of the Jacobian are denoted as J_v and J_ω respectively, the Jacobian can be written as

$$J^{Base} = \left[\frac{J_v}{J_\omega}\right] \tag{D.3}$$

Then (D.1) can be expressed as

$$\begin{aligned} v^{Base} &= J_v\dot{\theta} \\ \omega^{Base} &= J_\omega\dot{\theta} \end{aligned} \tag{D.4}$$

For the revolute joint, the lower half of the Jacobian J_ω is given as

$$J_\omega = [z_1^{Base}, z_2^{Base}, \cdots, z_n^{Base}] \tag{D.5}$$

where

$$z_i^{Base} = R_i^{Base}\boldsymbol{k} \tag{D.6}$$

$\boldsymbol{k}$ is the unit coordinate vector $[0, 0, 1]^T$, and R_i^{Base} is the rotational transformation matrix in ${}^{Base}T_i$.

Let ${}^{Base}x_p = [x, y, z]^T$ denote the position vector of the end-effector in the base coordinate system, then matrix J_v can be obtained by directly differentiating ${}^{Base}x_p$.

$$v = {}^{Base}\dot{x}_p = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \frac{\partial {}^{Base}x_p}{\partial \theta_1} \cdot \dot{\theta}_1 + \frac{\partial {}^{Base}x_p}{\partial \theta_2} \cdot \dot{\theta}_2 + \cdots + \frac{\partial {}^{Base}x_p}{\partial \theta_n} \cdot \dot{\theta}_n \tag{D.7}$$

$$J_v = \begin{bmatrix} \frac{\partial {}^{Base}x_p}{\partial \theta_1} & \frac{\partial {}^{Base}x_p}{\partial \theta_2} & \cdots & \frac{\partial {}^{Base}x_p}{\partial \theta_n} \end{bmatrix} \tag{D.8}$$

The other method to compute J_v is as follows:

If the upper half of the Jacobian J_v is given as $J_v = [J_{v1}, J_{v_2}, \cdots, J_{v_n}]$, then for the revolute joint i, the i^{th} column J_{v_i} is

$$J_{v_i} = z_i^{Base} \times (O_n^{Base} - O_i^{Base}) \tag{D.9}$$

where O_n^{Base} and O_i^{Base} represent the positions of the end-effector and the i^{th} joint in the base coordinate frame, respectively.

Expressed in the End-Effector Coordinate System

Jacobian matrix at the end-effector can be calculated by using the differential transform method, as shown in Figure D.1. If the Jacobian for an n-joint leg is of the form $J = [J_1, J_2, \cdots, J_n]$, the i^{th} column of the Jacobian J_i can be derived given the transformation matrix ${}^{i-1}T_n$ from each link to the end-effector. When the joint is revolute,

$$J_i = \begin{bmatrix} (p \times n)_z \\ (p \times o)_z \\ (p \times a)_z \\ n_z \\ o_z \\ a_z \end{bmatrix} \tag{D.10}$$

where $\boldsymbol{n}$, $\boldsymbol{o}$, $\boldsymbol{a}$ and $\boldsymbol{p}$ are the column vectors of the matrix ${}^{i-1}T_n$, respectively.

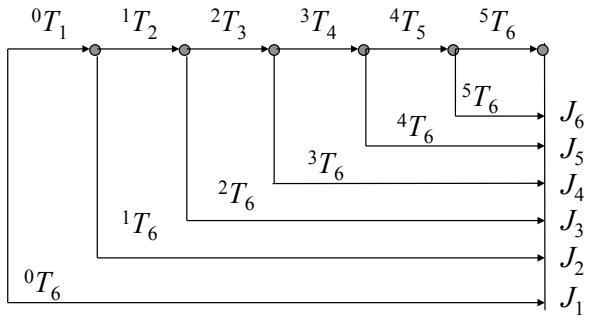


Figure D.1: Jacobian matrix calculation.

NAO's Jacobian in the RCS

We take the robot coordinate system as the base coordinate system. Let d_y = HipOffsetY, d_z = HipOffsetZ and d_{fh} = FootHeight, the Jacobian in the base coordinate system for the left leg is calculated as follows. Refer to Table B.3 for the representation of $\theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$.

$$
{}^{Base}T_1 = {}^{Base}T_0 \cdot {}^0T_1 = \overset{z_1^{Base}}{\begin{pmatrix} s\theta_1 & c\theta_1 & 0 & 0 \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & \frac{\sqrt{2}}{2} & d_y \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & -\frac{\sqrt{2}}{2} & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}
$$

$$
{}^{Base}T_2 = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 = \overset{z_2^{Base}}{\begin{pmatrix} \times & \times & c\theta_1 & 0 \\ \times & \times & -\frac{\sqrt{2}}{2}s\theta_1 & d_y \\ \times & \times & -\frac{\sqrt{2}}{2}s\theta_1 & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}
$$

$$
{}^{Base}T_3 = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 = \overset{z_3^{Base}}{\begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_2 + c\theta_2) & 0 \\ \times & \times & \frac{1}{2}((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2) & d_y \\ \times & \times & \frac{1}{2}((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2) & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}
$$

$$
{}^{Base}T_4 = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 = \overset{z_4^{Base}}{\begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_2 + c\theta_2) & \times \\ \times & \times & \frac{1}{2}((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2) & \times \\ \times & \times & \frac{1}{2}((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2) & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}}
$$

$$
{}^{Base}T_5 = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 = \overset{z_5^{Base}}{\begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_2 + c\theta_2) & \times \\ \times & \times & \frac{1}{2}((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2) & \times \\ \times & \times & \frac{1}{2}((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2) & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}}
$$

$$^{Base}T_6 = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 = \begin{pmatrix} \times & \times & \overset{z_6^{Base}}{i_{13}} & \times \\ \times & \times & i_{23} & \times \\ \times & \times & i_{33} & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\begin{aligned}
i_{13} &= c\theta_1 c(\theta_3+\theta_4+\theta_5) - c(\pi/4+\theta_2)s\theta_1 s(\theta_3+\theta_4+\theta_5) \\
i_{23} &= \frac{1}{2}((1+c\theta_1)s\theta_2 + (1-c\theta_1)c\theta_2)s(\theta_3+\theta_4+\theta_5) - \frac{\sqrt{2}}{2}s\theta_1 c(\theta_3+\theta_4+\theta_5) \\
i_{33} &= -\frac{1}{2}((1-c\theta_1)s\theta_2 + (1+c\theta_1)c\theta_2)s(\theta_3+\theta_4+\theta_5) - \frac{\sqrt{2}}{2}s\theta_1 c(\theta_3+\theta_4+\theta_5)
\end{aligned}$$

If the position of the left foot in the RCS is expressed as $^{Base}x_p^l = [x\ y\ z]$, $^{Base}x_p^l$ can be calculated by means of forward kinematics.

$$^{Base}T_{End} = {}^{Base}T_0 \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_{End} = \begin{pmatrix} \times & \times & \times & \overset{^{Base}x_p^l}{x} \\ \times & \times & \times & y \\ \times & \times & \times & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\begin{aligned}
x = & -(c\theta_1 c\theta_3 - \frac{\sqrt{2}}{2}s\theta_1 s\theta_3(c\theta_2 - s\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
& + \frac{\sqrt{2}}{2}d_{fh}s\theta_1 s\theta_6(c\theta_2 + s\theta_2) + (c\theta_1 s\theta_3 + \frac{\sqrt{2}}{2}s\theta_1 c\theta_3(c\theta_2 - s\theta_2))(d_{fh}s\theta_4 s\theta_5 c\theta_6 \\
& - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
y = & (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 \\
& - a_5 s\theta_4) + d_{fh}\frac{1}{2}s\theta_6((c\theta_1+1)c\theta_2 + (c\theta_1-1)s\theta_2) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1+1)s\theta_2 \\
& + (c\theta_1-1)c\theta_2))(d_{fh}s\theta_4 s\theta_5 c\theta_6 - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) + d_y \\
z = & (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 \\
& - a_5 s\theta_4) + \frac{1}{2}d_{fh}s\theta_6((c\theta_1-1)c\theta_2 + (c\theta_1+1)s\theta_2) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1-1)s\theta_2 \\
& + (c\theta_1+1)c\theta_2))(d_{fh}s\theta_4 s\theta_5 c\theta_6 - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) - d_z
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\partial x}{\partial \theta_1} &= (s\theta_1 c\theta_3 + \frac{\sqrt{2}}{2} c\theta_1 s\theta_3 (c\theta_2 - s\theta_2))(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
&\quad + d_{fh} \frac{\sqrt{2}}{2} c\theta_1 s\theta_6 (c\theta_2 + s\theta_2) + (-s\theta_1 s\theta_3 + \frac{\sqrt{2}}{2} c\theta_1 c\theta_3 (c\theta_2 - s\theta_2))(d_{fh} s\theta_4 s\theta_5 c\theta_6 \\
&\quad - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
\frac{\partial x}{\partial \theta_2} &= -\frac{\sqrt{2}}{2} s\theta_1 s\theta_3 (c\theta_2 + s\theta_2)(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
&\quad + d_{fh} \frac{\sqrt{2}}{2} s\theta_1 s\theta_6 (c\theta_2 - s\theta_2) - \frac{\sqrt{2}}{2} s\theta_1 c\theta_3 (c\theta_2 + s\theta_2)(d_{fh} s\theta_4 s\theta_5 c\theta_6 \\
&\quad - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
\frac{\partial x}{\partial \theta_3} &= (c\theta_1 s\theta_3 + \frac{\sqrt{2}}{2} s\theta_1 c\theta_3 (c\theta_2 - s\theta_2))(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
&\quad + (c\theta_1 c\theta_3 - \frac{\sqrt{2}}{2} s\theta_1 s\theta_3 (c\theta_2 - s\theta_2))(d_{fh} s\theta_4 s\theta_5 c\theta_6 - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
\frac{\partial x}{\partial \theta_4} &= -(c\theta_1 c\theta_3 - \frac{\sqrt{2}}{2} s\theta_1 s\theta_3 (c\theta_2 - s\theta_2))(-d_{fh} s\theta_4 s\theta_5 c\theta_6 + d_{fh} c\theta_4 c\theta_5 c\theta_6 - a_5 c\theta_4) \\
&\quad + (c\theta_1 s\theta_3 + \frac{\sqrt{2}}{2} s\theta_1 c\theta_3 (c\theta_2 - s\theta_2))(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
\frac{\partial x}{\partial \theta_5} &= -(c\theta_1 c\theta_3 - \frac{\sqrt{2}}{2} s\theta_1 s\theta_3 (c\theta_2 - s\theta_2))(d_{fh} c\theta_4 c\theta_5 c\theta_6 - d_{fh} s\theta_4 s\theta_5 c\theta_6) \\
&\quad + (c\theta_1 s\theta_3 + \frac{\sqrt{2}}{2} s\theta_1 c\theta_3 (c\theta_2 - s\theta_2))(d_{fh} s\theta_4 c\theta_5 c\theta_6 + d_{fh} c\theta_4 s\theta_5 c\theta_6) \\
\frac{\partial x}{\partial \theta_6} &= (c\theta_1 c\theta_3 - \frac{\sqrt{2}}{2} s\theta_1 s\theta_3 (s\theta_2 + c\theta_2))(d_{fh} c\theta_4 s\theta_5 s\theta_6 + d_{fh} s\theta_4 c\theta_5 s\theta_6) \\
&\quad - d_{fh} \frac{\sqrt{2}}{2} s\theta_1 c\theta_6 (c\theta_2 - s\theta_2) + (c\theta_1 s\theta_3 + \frac{\sqrt{2}}{2} s\theta_1 c\theta_3 (s\theta_2 + c\theta_2))(-d_{fh} s\theta_4 s\theta_5 s\theta_6 \\
&\quad + d_{fh} c\theta_4 c\theta_5 s\theta_6) \\
\frac{\partial y}{\partial \theta_1} &= (\frac{\sqrt{2}}{2} c\theta_1 c\theta_3 + \frac{1}{2} s\theta_1 s\theta_3 (s\theta_2 - c\theta_2))(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
&\quad - \frac{1}{2} d_{fh} s\theta_1 s\theta_6 (s\theta_2 + c\theta_2) - (\frac{\sqrt{2}}{2} c\theta_1 s\theta_3 + \frac{1}{2} s\theta_1 c\theta_3 (s\theta_2 - c\theta_2))(d_{fh} s\theta_4 s\theta_5 c\theta_6 \\
&\quad - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
\frac{\partial y}{\partial \theta_2} &= \frac{1}{2} s\theta_3 (-(c\theta_1 + 1)c\theta_2 - (c\theta_1 - 1)s\theta_2)(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) \\
&\quad + \frac{1}{2} d_{fh} s\theta_6 (-(c\theta_1 + 1)s\theta_2 + (c\theta_1 - 1)c\theta_2) - \frac{1}{2} c\theta_3 ((c\theta_1 + 1)c\theta_2 \\
&\quad + (c\theta_1 - 1)s\theta_2)(d_{fh} s\theta_4 s\theta_5 c\theta_6 - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4) \\
\frac{\partial y}{\partial \theta_3} &= (-\frac{\sqrt{2}}{2} s\theta_1 s\theta_3 + \frac{1}{2} c\theta_3 (-(c\theta_1 + 1)s\theta_2 + (c\theta_1 - 1)c\theta_2))(d_{fh} c\theta_4 s\theta_5 c\theta_6 + d_{fh} s\theta_4 c\theta_5 c\theta_6 \\
&\quad - a_5 s\theta_4) - (\frac{\sqrt{2}}{2} s\theta_1 c\theta_3 + \frac{1}{2} s\theta_3 (-(c\theta_1 + 1)s\theta_2 + (c\theta_1 - 1)c\theta_2))(d_{fh} s\theta_4 s\theta_5 c\theta_6 \\
&\quad - d_{fh} c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4)
\end{aligned}$$

$$\frac{\partial y}{\partial \theta_4} = (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(-d_{fh}s\theta_4 s\theta_5 c\theta_6 + d_{fh}c\theta_4 c\theta_5 c\theta_6 - a_5 c\theta_4) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4)$$

$$\frac{\partial y}{\partial \theta_5} = (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(d_{fh}c\theta_4 c\theta_5 c\theta_6 - d_{fh}s\theta_4 s\theta_5 c\theta_6) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(d_{fh}s\theta_4 c\theta_5 c\theta_6 + d_{fh}c\theta_4 s\theta_5 c\theta_6)$$

$$\frac{\partial y}{\partial \theta_6} = -(\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 s\theta_6 + d_{fh}s\theta_4 c\theta_5 s\theta_6) + \frac{1}{2}d_{fh}c\theta_6((c\theta_1+1)c\theta_2 + (c\theta_1-1)s\theta_2) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1+1)s\theta_2 + (c\theta_1-1)c\theta_2))(-d_{fh}s\theta_4 s\theta_5 s\theta_6 + d_{fh}c\theta_4 c\theta_5 s\theta_6)$$

$$\frac{\partial z}{\partial \theta_1} = (\frac{\sqrt{2}}{2}c\theta_1 c\theta_3 + \frac{1}{2}s\theta_1 s\theta_3(s\theta_2 - c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) - \frac{1}{2}d_{fh}s\theta_1 s\theta_6(s\theta_2 + c\theta_2) + (-\frac{\sqrt{2}}{2}c\theta_1 s\theta_3 + \frac{1}{2}s\theta_1 c\theta_3(s\theta_2 - c\theta_2))(d_{fh}s\theta_4 s\theta_5 c\theta_6 - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4)$$

$$\frac{\partial z}{\partial \theta_2} = -\frac{1}{2}s\theta_3((c\theta_1-1)c\theta_2 + (c\theta_1+1)s\theta_2)(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) + \frac{1}{2}d_{fh}s\theta_6(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2) - \frac{1}{2}c\theta_3((c\theta_1-1)c\theta_2 + (c\theta_1+1)s\theta_2)(d_{fh}s\theta_4 s\theta_5 c\theta_6 - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4)$$

$$\frac{\partial z}{\partial \theta_3} = (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4) - (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}s\theta_4 s\theta_5 c\theta_6 - d_{fh}c\theta_4 c\theta_5 c\theta_6 + a_5 c\theta_4 + a_4)$$

$$\frac{\partial z}{\partial \theta_4} = (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(-d_{fh}s\theta_4 s\theta_5 c\theta_6 + d_{fh}c\theta_4 c\theta_5 c\theta_6 - a_5 c\theta_4) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 c\theta_6 + d_{fh}s\theta_4 c\theta_5 c\theta_6 - a_5 s\theta_4)$$

$$\frac{\partial z}{\partial \theta_5} = (\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}c\theta_4 c\theta_5 c\theta_6 - d_{fh}s\theta_4 s\theta_5 c\theta_6) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}s\theta_4 c\theta_5 c\theta_6 + d_{fh}c\theta_4 s\theta_5 c\theta_6)$$

$$\frac{\partial z}{\partial \theta_6} = -(\frac{\sqrt{2}}{2}s\theta_1 c\theta_3 + \frac{1}{2}s\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(d_{fh}c\theta_4 s\theta_5 s\theta_6 + d_{fh}s\theta_4 c\theta_5 s\theta_6) + \frac{1}{2}d_{fh}c\theta_6((c\theta_1-1)c\theta_2 + (c\theta_1+1)s\theta_2) + (-\frac{\sqrt{2}}{2}s\theta_1 s\theta_3 + \frac{1}{2}c\theta_3(-(c\theta_1-1)s\theta_2 + (c\theta_1+1)c\theta_2))(-d_{fh}s\theta_4 s\theta_5 s\theta_6 + d_{fh}c\theta_4 c\theta_5 s\theta_6)$$

Combining the angular and linear Jacobians, we get the Jacobian for the left leg

$$J_{left}^{Base} = \begin{pmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} & \frac{\partial x}{\partial \theta_4} & \frac{\partial x}{\partial \theta_5} & \frac{\partial x}{\partial \theta_6} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} & \frac{\partial y}{\partial \theta_4} & \frac{\partial y}{\partial \theta_5} & \frac{\partial y}{\partial \theta_6} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} & \frac{\partial z}{\partial \theta_4} & \frac{\partial z}{\partial \theta_5} & \frac{\partial z}{\partial \theta_6} \\ z_1^{Base} & z_2^{Base} & z_3^{Base} & z_4^{Base} & z_5^{Base} & z_6^{Base} \end{pmatrix} \tag{D.11}$$

The position of the end-effector of the right leg in the base coordinate frame is expressed as ${}^{Base}x_p^r = [x, y, z]^T$, where

$$\begin{aligned}
x = & -(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}) \\
& + \frac{\sqrt{2}}{2} d_{fh} s\theta_7 s\theta_{12} (s\theta_8 - c\theta_8) + (c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (s\theta_8 + c\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} \\
& - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}) \\
y = & -(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 ((1 - c\theta_7) c\theta_8 - (c\theta_7 + 1) s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} \\
& - a_{11} s\theta_{10}) + \frac{1}{2} d_{fh} s\theta_{12} ((c\theta_7 + 1) c\theta_8 + (1 - c\theta_7) s\theta_8) + (\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 ((1 - c\theta_7) c\theta_8 \\
& - (c\theta_7 + 1) s\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}) - d_y \\
z = & (\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} s\theta_9 ((c\theta_7 + 1) c\theta_8 + (c\theta_7 - 1) s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} \\
& - a_{11} s\theta_{10}) + \frac{1}{2} d_{fh} s\theta_{12} ((1 - c\theta_7) c\theta_8 + (c\theta_7 + 1) s\theta_8) + (-\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 ((c\theta_7 - 1) s\theta_8 \\
& + (c\theta_7 + 1) c\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}) - d_z
\end{aligned}$$

The Jacobian for the right leg is as follows:

$$J_{right}^{Base} = \begin{pmatrix} \frac{\partial x}{\partial \theta_7} & \frac{\partial x}{\partial \theta_8} & \frac{\partial x}{\partial \theta_9} & \frac{\partial x}{\partial \theta_{10}} & \frac{\partial x}{\partial \theta_{11}} & \frac{\partial x}{\partial \theta_{12}} \\ \frac{\partial y}{\partial \theta_7} & \frac{\partial y}{\partial \theta_8} & \frac{\partial y}{\partial \theta_9} & \frac{\partial y}{\partial \theta_{10}} & \frac{\partial y}{\partial \theta_{11}} & \frac{\partial y}{\partial \theta_{12}} \\ \frac{\partial z}{\partial \theta_7} & \frac{\partial z}{\partial \theta_8} & \frac{\partial z}{\partial \theta_9} & \frac{\partial z}{\partial \theta_{10}} & \frac{\partial z}{\partial \theta_{11}} & \frac{\partial z}{\partial \theta_{12}} \\ z_7^{Base} & z_8^{Base} & z_9^{Base} & z_{10}^{Base} & z_{11}^{Base} & z_{12}^{Base} \end{pmatrix} \tag{D.12}$$

where

$$\frac{\partial x}{\partial \theta_7} = (s\theta_7 c\theta_9 + \frac{\sqrt{2}}{2} c\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$
$$+ \frac{\sqrt{2}}{2} d_{fh} c\theta_7 s\theta_{12} (s\theta_8 - c\theta_8) + (-s\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} c\theta_7 c\theta_9 (c\theta_8 + s\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}$$
$$- d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10})$$

$$\frac{\partial x}{\partial \theta_8} = \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 - s\theta_8)(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$
$$+ \frac{\sqrt{2}}{2} d_{fh} s\theta_7 s\theta_{12} (c\theta_8 + s\theta_8) + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (c\theta_8 - s\theta_8)(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}$$
$$- d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10})$$

$$\frac{\partial x}{\partial \theta_9} = (c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$
$$+ (c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10}$$
$$+ a_{10})$$

$$\frac{\partial x}{\partial \theta_{10}} = -(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(-d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - a_{11} c\theta_{10})$$
$$+ (c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$

$$\frac{\partial x}{\partial \theta_{11}} = -(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - d_{fh} s\theta_{10} s\theta_{11} c\theta_{12})$$
$$+ (c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (c\theta_8 + s\theta_8))(d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} s\theta_{11} c\theta_{12})$$

$$\frac{\partial x}{\partial \theta_{12}} = (c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 (c\theta_8 + s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} s\theta_{12})$$
$$+ \frac{\sqrt{2}}{2} d_{fh} s\theta_7 c\theta_{12} (s\theta_8 - c\theta_8) + (c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 (s\theta_8 + c\theta_8))(-d_{fh} s\theta_{10} s\theta_{11} s\theta_{12}$$
$$+ d_{fh} c\theta_{10} c\theta_{11} s\theta_{12})$$

$$\frac{\partial y}{\partial \theta_7} = (-\frac{\sqrt{2}}{2} c\theta_7 c\theta_9 + \frac{1}{2} s\theta_7 s\theta_9 (s\theta_8 + c\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$
$$- \frac{1}{2} d_{fh} s\theta_7 s\theta_{12} (c\theta_8 - s\theta_8) + (\frac{\sqrt{2}}{2} c\theta_7 s\theta_9 + \frac{1}{2} s\theta_7 c\theta_9 (s\theta_8 + c\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}$$
$$- d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10})$$

$$\frac{\partial y}{\partial \theta_8} = \frac{1}{2} s\theta_9 ((c\theta_7 - 1) s\theta_8 - (c\theta_7 + 1) c\theta_8)(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10})$$
$$+ \frac{1}{2} d_{fh} s\theta_{12} (-(c\theta_7 + 1) s\theta_8 + (1 - c\theta_7) c\theta_8) + \frac{1}{2} c\theta_9 ((c\theta_7 - 1) s\theta_8$$
$$- (c\theta_7 + 1) c\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10})$$

$$\frac{\partial y}{\partial \theta_9} = (\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 ((1 - c\theta_7) c\theta_8 - (c\theta_7 + 1) s\theta_8))(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12}$$
$$- a_{11} s\theta_{10}) + (\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 ((1 - c\theta_7) c\theta_8 - (c\theta_7 + 1) s\theta_8))(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}$$
$$- d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10})$$

$$\frac{\partial y}{\partial \theta_{10}} = -(\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 - \frac{1}{2}s\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(-d_{fh}s\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} - a_{11}c\theta_{10}) + (\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} - a_{11}s\theta_{10})$$

$$\frac{\partial y}{\partial \theta_{11}} = -(\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 - \frac{1}{2}s\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} - d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}) + (\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} + d_{fh}c\theta_{10}s\theta_{11}c\theta_{12})$$

$$\frac{\partial y}{\partial \theta_{12}} = (\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 - \frac{1}{2}s\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(d_{fh}c\theta_{10}s\theta_{11}s\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}s\theta_{12}) + \frac{1}{2}d_{fh}c\theta_{12}((c\theta_7+1)c\theta_8 + (1-c\theta_7)s\theta_8) + (\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8))(-d_{fh}s\theta_{10}s\theta_{11}s\theta_{12} + d_{fh}c\theta_{10}c\theta_{11}s\theta_{12})$$

$$\frac{\partial z}{\partial \theta_7} = (\frac{\sqrt{2}}{2}c\theta_7 c\theta_9 - \frac{1}{2}s\theta_7 s\theta_9(s\theta_8 + c\theta_8))(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} - a_{11}s\theta_{10}) + \frac{1}{2}d_{fh}s\theta_7 s\theta_{12}(c\theta_8 - s\theta_8) - (\frac{\sqrt{2}}{2}c\theta_7 s\theta_9 + \frac{1}{2}s\theta_7 c\theta_9(c\theta_8 + s\theta_8))(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12} - d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} + a_{11}c\theta_{10} + a_{10})$$

$$\frac{\partial z}{\partial \theta_8} = \frac{1}{2}s\theta_9((c\theta_7-1)c\theta_8 - (c\theta_7+1)s\theta_8)(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} - a_{11}s\theta_{10}) + \frac{1}{2}d_{fh}s\theta_{12}((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8) + \frac{1}{2}c\theta_9((c\theta_7-1)c\theta_8 - (c\theta_7+1)s\theta_8))(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12} - d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} + a_{11}c\theta_{10} + a_{10})$$

$$\frac{\partial z}{\partial \theta_9} = (-\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} - a_{11}s\theta_{10}) - (\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 + \frac{1}{2}s\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12} - d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} + a_{11}c\theta_{10} + a_{10})$$

$$\frac{\partial z}{\partial \theta_{10}} = (\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 + \frac{1}{2}s\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(-d_{fh}s\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} - a_{11}c\theta_{10}) + (-\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} - a_{11}s\theta_{10})$$

$$\frac{\partial z}{\partial \theta_{11}} = (\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 + \frac{1}{2}s\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}c\theta_{10}c\theta_{11}c\theta_{12} - d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}) + (-\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}s\theta_{10}c\theta_{11}c\theta_{12} + d_{fh}c\theta_{10}s\theta_{11}c\theta_{12})$$

$$\frac{\partial z}{\partial \theta_{12}} = -(\frac{\sqrt{2}}{2}s\theta_7 c\theta_9 + \frac{1}{2}s\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(d_{fh}c\theta_{10}s\theta_{11}s\theta_{12} + d_{fh}s\theta_{10}c\theta_{11}s\theta_{12}) + \frac{1}{2}d_{fh}c\theta_{12}((1-c\theta_7)c\theta_8 + (c\theta_7+1)s\theta_8) + (-\frac{\sqrt{2}}{2}s\theta_7 s\theta_9 + \frac{1}{2}c\theta_9((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8))(-d_{fh}s\theta_{10}s\theta_{11}s\theta_{12} + d_{fh}c\theta_{10}c\theta_{11}s\theta_{12})$$

$$z_7^{Base} = \begin{pmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \quad z_8^{Base} = \begin{pmatrix} c\theta_7 \\ \frac{\sqrt{2}}{2}s\theta_7 \\ \frac{\sqrt{2}}{2}s\theta_7 \end{pmatrix}$$

$$z_9^{Base} = z_{10}^{Base} = z_{11}^{Base} = \begin{pmatrix} \frac{\sqrt{2}}{2}s\theta_7(s\theta_8 - c\theta_8) \\ \frac{1}{2}((1+c\theta_7)c\theta_8 + (1-c\theta_7)s\theta_8) \\ \frac{1}{2}((1-c\theta_7)c\theta_8 + (1+c\theta_7)s\theta_8) \end{pmatrix}$$

$$z_{12}^{Base} = \begin{pmatrix} c\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) - s\theta_7 c(\theta_8 - \pi/4)s(\theta_9+\theta_{10}+\theta_{11}) \\ \frac{\sqrt{2}}{2}s\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) + \frac{1}{2}((1+c\theta_7)s\theta_8 + (c\theta_7-1)c\theta_8)s(\theta_9+\theta_{10}+\theta_{11}) \\ -\frac{\sqrt{2}}{2}s\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) + \frac{1}{2}((1-c\theta_7)s\theta_8 - (c\theta_7+1)c\theta_8)s(\theta_9+\theta_{10}+\theta_{11}) \end{pmatrix}$$

Note that if the configuration of the leg obtained from joint position sensors during walking is $\theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$, the joint angles used for D-H matrices are $\theta = [\theta_1 \quad -\theta_2 \quad \cdots \quad -\theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$ due to the different definitions of joint rotational direction between the robot NAO and D-H parameters.

NAO's Jacobian in the FCS

We now calculate the Jacobian matrix for the left leg $J(\theta) = [J_1, J_2, \cdots, J_6]$ in the foot coordinate system. According to (D.10), we have

$${}^5T_{End} = {}^5T_6\,{}^6T_{End} = \begin{bmatrix} 0 & s\theta_6 & c\theta_6 & -d_{fh}c\theta_6 \\ 1 & 0 & 0 & 0 \\ 0 & c\theta_6 & -s\theta_6 & d_{fh}s\theta_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad J_6 = \begin{bmatrix} -d_{fh}c\theta_6 \\ 0 \\ 0 \\ 0 \\ c\theta_6 \\ -s\theta_6 \end{bmatrix}$$

$${}^4T_{End} = {}^4T_5\,{}^5T_{End} = \begin{bmatrix} -s\theta_5 & c\theta_5 s\theta_6 & c\theta_5 c\theta_6 & a_5 - d_{fh}c\theta_5 c\theta_6 \\ c\theta_5 & s\theta_5 s\theta_6 & s\theta_5 c\theta_6 & -d_{fh}s\theta_5 c\theta_6 \\ 0 & c\theta_6 & -s\theta_6 & d_{fh}s\theta_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad J_5 = \begin{bmatrix} a_5 c\theta_5 - d_{fh}c\theta_6 \\ a_5 s\theta_5 s\theta_6 \\ a_5 s\theta_5 c\theta_6 \\ 0 \\ c\theta_6 \\ -s\theta_6 \end{bmatrix}$$

Likewise,

$$ {}^3T_{End} = {}^3T_4 {}^4T_{End}, \quad J_4 = \begin{bmatrix} a_4c(\theta_4+\theta_5) + a_5c\theta_5 - d_{fh}c\theta_6 \\ s\theta_6(a_5s\theta_5 + a_4s(\theta_4+\theta_5)) \\ c\theta_6(a_5s\theta_5 + a_4s(\theta_4+\theta_5)) \\ 0 \\ c\theta_6 \\ -s\theta_6 \end{bmatrix} $$

$$ {}^2T_{End} = {}^2T_3 {}^3T_{End}, \quad J_3 = \begin{bmatrix} -d_{fh}s\theta_6 s(\theta_3+\theta_4+\theta_5) \\ d_{fh}c(\theta_3+\theta_4+\theta_5) - a_5c\theta_6c(\theta_3+\theta_4) - a_4c\theta_3c\theta_6 \\ a_5s\theta_6c(\theta_3+\theta_4) + a_4c\theta_3s\theta_6 \\ c(\theta_3+\theta_4+\theta_5) \\ s\theta_6s(\theta_3+\theta_4+\theta_5) \\ c\theta_6s(\theta_3+\theta_4+\theta_5) \end{bmatrix} $$

$$ {}^1T_{End} = {}^1T_2 {}^2T_{End}, \quad J_2 = \begin{bmatrix} J_{21} \\ J_{22} \\ J_{23} \\ J_{24} \\ J_{25} \\ J_{26} \end{bmatrix} $$

where

$$ J_{21} = \frac{\sqrt{2}}{2}(c\theta_2 - s\theta_2)(a_5c\theta_5 + a_4c(\theta_4+\theta_5) - d_{fh}c\theta_6) + \frac{\sqrt{2}}{2}d_{fh}s\theta_6(c\theta_2+s\theta_2)c(\theta_3+\theta_4+\theta_5) $$

$$ J_{22} = \frac{\sqrt{2}}{2}(c\theta_2 + s\theta_2)(d_{fh}s(\theta_3+\theta_4+\theta_5) - c\theta_6(a_5s(\theta_3+\theta_4) + a_4s\theta_3)) $$
$$ + \frac{\sqrt{2}}{2}(c\theta_2 - s\theta_2)s\theta_6(a_5s\theta_5 + a_4s(\theta_4+\theta_5)) $$

$$ J_{23} = \frac{\sqrt{2}}{2}(c\theta_2 - s\theta_2)c\theta_6(a_4s(\theta_4+\theta_5) + a_5s\theta_5) + \frac{\sqrt{2}}{2}(c\theta_2+s\theta_2)s\theta_6(a_5s(\theta_3+\theta_4) + a_4s\theta_3) $$

$$ J_{24} = \frac{\sqrt{2}}{2}(c\theta_2 + s\theta_2)s(\theta_3+\theta_4+\theta_5) $$

$$ J_{25} = \frac{\sqrt{2}}{2}c\theta_6(c\theta_2 - s\theta_2) - \frac{\sqrt{2}}{2}s\theta_6(c\theta_2+s\theta_2)c(\theta_3+\theta_4+\theta_5) $$

$$ J_{26} = -\frac{\sqrt{2}}{2}s\theta_6(c\theta_2 - s\theta_2) - \frac{\sqrt{2}}{2}c\theta_6(c\theta_2+s\theta_2)c(\theta_3+\theta_4+\theta_5) $$

$$ {}^0T_{End} = {}^0T_1 {}^1T_{End}, \quad J_1 = \begin{bmatrix} J_{11} \\ J_{12} \\ J_{13} \\ J_{14} \\ J_{15} \\ J_{16} \end{bmatrix} $$

where

$$
\begin{aligned}
J_{11} &= s\theta_2 s(\theta_3+\theta_4+\theta_5)(a_5 s(\theta_3+\theta_4)+a_4 s\theta_3) - d_{fh} s\theta_2 c\theta_6 \\
&\quad + c(\theta_3+\theta_4+\theta_5)(a_5 s\theta_2 c(\theta_3+\theta_4)+a_4 s\theta_2 c\theta_3 - d_{fh} c\theta_2 s\theta_6) \\
J_{12} &= (c\theta_2 c\theta_6 - s\theta_2 s\theta_6 c(\theta_3+\theta_4+\theta_5))(a_5 s(\theta_3+\theta_4)+a_4 s\theta_3) \\
&\quad + s(\theta_3+\theta_4+\theta_5)(a_5 s\theta_2 s\theta_6 c(\theta_3+\theta_4)+a_4 s\theta_2 c\theta_3 s\theta_6 - d_{fh} c\theta_2) \\
J_{13} &= -(c\theta_2 s\theta_6 + s\theta_2 c\theta_6 c(\theta_3+\theta_4+\theta_5))(a_5 s(\theta_3+\theta_4)+a_4 s\theta_3) \\
&\quad + s\theta_2 c\theta_6 s(\theta_3+\theta_4+\theta_5)(a_5 c(\theta_3+\theta_4)+a_4 c\theta_3) \\
J_{14} &= -\frac{1}{2}((1+c\theta_1)c\theta_2 + (1-c\theta_1)s\theta_2)s(\theta_3+\theta_4+\theta_5) - \frac{\sqrt{2}}{2} s\theta_1 c(\theta_3+\theta_4+\theta_5) \\
J_{15} &= s\theta_6(\frac{1}{2}((1+c\theta_1)c\theta_2 + (1-c\theta_1)s\theta_2)c(\theta_3+\theta_4+\theta_5) - \frac{\sqrt{2}}{2} s\theta_1 s(\theta_3+\theta_4+\theta_5)) \\
&\quad - \frac{1}{2} c\theta_6((1-c\theta_1)c\theta_2 - (1+c\theta_1)s\theta_2) \\
J_{16} &= c\theta_6(\frac{1}{2}((1+c\theta_1)c\theta_2 + (1-c\theta_1)s\theta_2)c(\theta_3+\theta_4+\theta_5) - \frac{\sqrt{2}}{2} s\theta_1 s(\theta_3+\theta_4+\theta_5)) \\
&\quad + \frac{1}{2} s\theta_6((1-c\theta_1)c\theta_2 - (1+c\theta_1)s\theta_2)
\end{aligned}
$$

The Jacobian matrix for the right leg $J(\theta) = [J_1, J_2, \cdots, J_6]$ in the foot coordinate system is

$$
J_6 = \begin{bmatrix} -d_{fh} c\theta_{12} \\ 0 \\ 0 \\ 0 \\ c\theta_{12} \\ -s\theta_{12} \end{bmatrix}, J_5 = \begin{bmatrix} a_{11} c\theta_{11} - d_{fh} c\theta_{12} \\ a_{11} s\theta_{11} s\theta_{12} \\ a_{11} s\theta_{11} c\theta_{12} \\ 0 \\ c\theta_{12} \\ -s\theta_{12} \end{bmatrix}, J_4 = \begin{bmatrix} a_{10} c(\theta_{10}+\theta_{11}) + a_{11} c\theta_{11} - d_{fh} c\theta_{12} \\ s\theta_{12}(a_{11} s\theta_{11} + a_{10} s(\theta_{10}+\theta_{11})) \\ c\theta_{12}(a_{11} s\theta_{11} + a_{10} s(\theta_{10}+\theta_{11})) \\ 0 \\ c\theta_{12} \\ -s\theta_{12} \end{bmatrix}
$$

$$
J_3 = \begin{bmatrix} -d_{fh} s\theta_{12} s(\theta_9+\theta_{10}+\theta_{11}) \\ d_{fh} c(\theta_9+\theta_{10}+\theta_{11}) - a_{11} c\theta_{12} c(\theta_9+\theta_{10}) - a_{10} c\theta_9 c\theta_{12} \\ a_{11} s\theta_{12} c(\theta_9+\theta_{10}) + a_{10} c\theta_9 s\theta_{12} \\ c(\theta_9+\theta_{10}+\theta_{11}) \\ s\theta_{12} s(\theta_9+\theta_{10}+\theta_{11}) \\ c\theta_{12} s(\theta_9+\theta_{10}+\theta_{11}) \end{bmatrix}, \quad J_2 = \begin{bmatrix} J_{21} \\ J_{22} \\ J_{23} \\ J_{24} \\ J_{25} \\ J_{26} \end{bmatrix}
$$

where

$$
\begin{aligned}
J_{21} &= c(\theta_8 - \pi/4)(a_{11} c\theta_{11} + a_{10} c(\theta_{10}+\theta_{11}) - d_{fh} c\theta_{12}) + d_{fh} s(\theta_8 - \pi/4) s\theta_{12} c(\theta_9+\theta_{10}+\theta_{11}) \\
J_{22} &= d_{fh} s(\theta_8 - \pi/4) s(\theta_9+\theta_{10}+\theta_{11}) + s\theta_{12} c(\theta_8 - \pi/4)(a_{11} s\theta_{11} + a_{10} s(\theta_{10}+\theta_{11})) \\
&\quad - c\theta_{12} s(\theta_8 - \pi/4)(a_{10} s\theta_9 + a_{11} s(\theta_9+\theta_{10})) \\
J_{23} &= c(\theta_8 - \pi/4) c\theta_{12}(a_{11} s\theta_{11} + a_{10} s(\theta_{10}+\theta_{11})) + s(\theta_8 - \pi/4) s\theta_{12}(a_{10} s\theta_9 + a_{11} s(\theta_9+\theta_{10})) \\
J_{24} &= s(\theta_8 - \pi/4) s(\theta_9+\theta_{10}+\theta_{11}) \\
J_{25} &= c(\theta_8 - \pi/4) c\theta_{12} - s(\theta_8 - \pi/4) s\theta_{12} c(\theta_9+\theta_{10}+\theta_{11}) \\
J_{26} &= -s\theta_{12} c(\theta_8 - \pi/4) - c\theta_{12} s(\theta_8 - \pi/4) c(\theta_9+\theta_{10}+\theta_{11})
\end{aligned}
$$

$$J_1 = \begin{bmatrix} J_{11} \\ J_{12} \\ J_{13} \\ J_{14} \\ J_{15} \\ J_{16} \end{bmatrix}$$

where

$$\begin{aligned}
J_{11} &= s\theta_8 s(\theta_9+\theta_{10}+\theta_{11})(a_{11}s(\theta_9+\theta_{10})+a_{10}s\theta_9) - d_{fh}s\theta_8 c\theta_{12} \\
&\quad + c(\theta_9+\theta_{10}+\theta_{11})(a_{11}s\theta_8 c(\theta_9+\theta_{10})+a_{10}s\theta_8 c\theta_9 - d_{fh}c\theta_8 s\theta_{12}) \\
J_{12} &= (c\theta_8 c\theta_{12} - s\theta_8 s\theta_{12}c(\theta_9+\theta_{10}+\theta_{11}))(a_{11}s(\theta_9+\theta_{10})+a_{10}s\theta_9) \\
&\quad + s(\theta_9+\theta_{10}+\theta_{11})(a_{11}s\theta_8 s\theta_{12}c(\theta_9+\theta_{10})+a_{10}s\theta_8 c\theta_9 s\theta_{12} - d_{fh}c\theta_8) \\
J_{13} &= -(c\theta_8 s\theta_{12} + s\theta_8 c\theta_{12}c(\theta_9+\theta_{10}+\theta_{11}))(a_{11}s(\theta_9+\theta_{10})+a_{10}s\theta_9) \\
&\quad + s\theta_8 c\theta_{12}s(\theta_9+\theta_{10}+\theta_{11})(a_{11}c(\theta_9+\theta_{10})+a_{10}c\theta_9) \\
J_{14} &= \frac{1}{2}((1-c\theta_7)s\theta_8 - (1+c\theta_7)c\theta_8)s(\theta_9+\theta_{10}+\theta_{11}) - \frac{\sqrt{2}}{2}s\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) \\
J_{15} &= -s\theta_{12}(\frac{1}{2}((1-c\theta_7)s\theta_8 - (1+c\theta_7)c\theta_8)c(\theta_9+\theta_{10}+\theta_{11}) + \frac{\sqrt{2}}{2}s\theta_7 s(\theta_9+\theta_{10}+\theta_{11})) \\
&\quad + \frac{1}{2}c\theta_{12}((1-c\theta_7)c\theta_8 + (1+c\theta_7)s\theta_8) \\
J_{16} &= -c\theta_{12}(\frac{1}{2}((1-c\theta_7)s\theta_8 - (1+c\theta_7)c\theta_8)c(\theta_9+\theta_{10}+\theta_{11}) + \frac{\sqrt{2}}{2}s\theta_7 s(\theta_9+\theta_{10}+\theta_{11})) \\
&\quad - \frac{1}{2}s\theta_{12}((1+c\theta_7)s\theta_8 + (1-c\theta_7)c\theta_8)
\end{aligned}$$

θ_1 and θ_7 are the angles of the HipYawPitch joints of the left leg and right leg. They are actuated by the same motor. In both simulation and real-world walking of NAO, we set $\theta_1 = \theta_7 = 0$, so $\theta_1 = \theta_7 = 0$ while calculating the Jacobian matrices for both legs.

Full Body Humanoid Model

Figure D.2 shows the seven-link model of the robot during the SSP. Here we give the algorithm in m-code to calculate the terms in the matrices H, C, G, B and J in the full body dynamic equations of motion which has the form

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B\tau + J^T f_e \tag{D.13}$$

```
% Copyright (c) 2014 by Jing Liu. This code may be freely used for
% noncommercial ends. If use of this code in part or in whole results in
% publication, proper citation must be included in that publication.
% This code comes with no guarantees or support.
%
% A MATLAB script to generate the equations of motion for a 7-link,
% 3D, kneed, biped walker using the method of Lagrange.
%
```

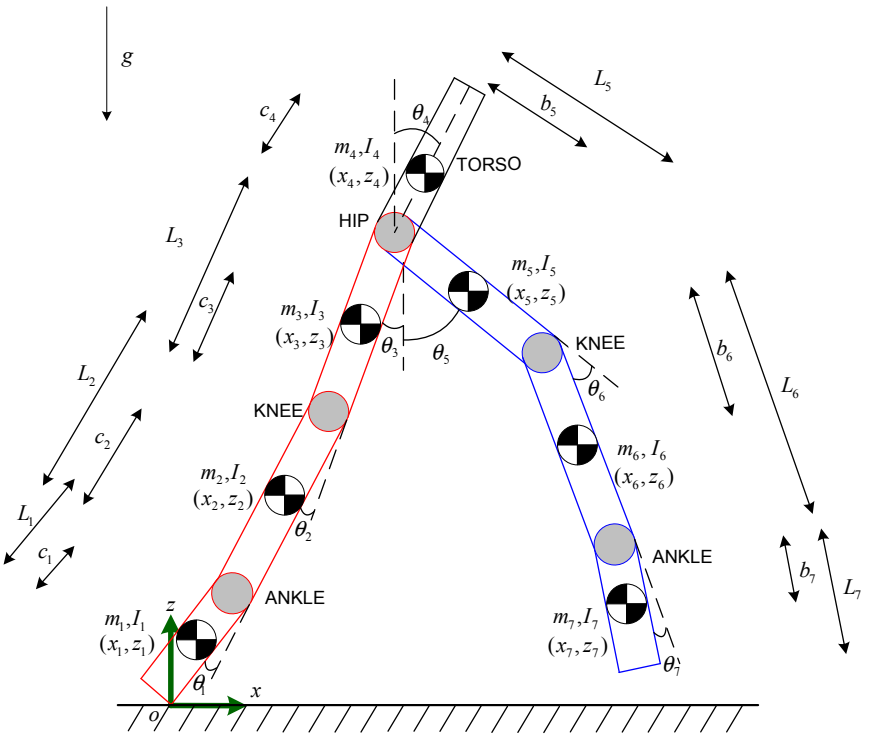
```
9  % The notation used for the equation of motions are as in Robot Dynamics
10 % and Control by Spong and Vidyasagar (1st edition, 1989), page 142,
11 % Eq. (6.3.12). The algorithm for calculating the matrices H(q), C(q, dq)
12 % and G(q) refers to Eq. (6.3.3)- Eq. (6.3.13) on pages 141-143, as well as
13 % Eric R. Westervelt's method which can be found at
14 % http://web.eecs.umich.edu/¬grizzle/westervelt_thesis/code/
15 %
16 % H(q)ddq + C(q, dq) * dq + G(q) = B * tau + E2 * f_e
17 %
18 % where f_e = [f_x; f_y; f_z], E2 = J^T, and J is the Jacobian matrix.
19 %
20 % Note the following convention
21 %
22 % 1 = stance leg, 2 = swing leg
23 %
24 % Jing Liu, Thursday June 5th 2014
25
26 clear
27
28 % ------------------------------------------------------------------
29 %
30 %  Model variables
31 %
32 % ------------------------------------------------------------------
33
34 % joint angles and velocities
35
36 syms q_torso_pitch dq_torso_pitch real
37 syms q_torso_roll dq_torso_roll real
38 syms q_hip_roll1 dq_hip_roll1 real
39 syms q_hip_pitch1 dq_hip_pitch1 real
40 syms q_knee1 dq_knee1 real
41 syms q_ankle_pitch1 dq_ankle_pitch1 real
42 syms q_ankle_roll1 dq_ankle_roll1 real
43 syms q_hip_roll2 dq_hip_roll2 real
44 syms q_hip_pitch2 dq_hip_pitch2 real
45 syms q_knee2 dq_knee2 real
46 syms q_ankle_pitch2 dq_ankle_pitch2 real
47 syms q_ankle_roll2 dq_ankle_roll2 real
48
49 q  = [q_ankle_roll1; q_ankle_pitch1; q_knee1; q_hip_pitch1; q_hip_roll1; ...
50       q_hip_roll2; q_hip_pitch2; q_knee2; q_ankle_pitch2; q_ankle_roll2; ...
51       q_torso_pitch; q_torso_roll];
52       % The angles are relative to the last link.
53 dq = [dq_ankle_roll1; dq_ankle_pitch1; dq_knee1; dq_hip_pitch1; ...
54       dq_hip_roll1; dq_hip_roll2; dq_hip_pitch2; dq_knee2; dq_ankle_pitch2;...
55       dq_ankle_roll2; dq_torso_pitch; dq_torso_roll];
56
57 % gravity
58 syms g real
59
60 % link lengths
61 syms L_torso real
62 syms L_thigh real
63 syms L_tibia real
64 syms L_foot real
65
```

```
% link masses
syms M_torso real
syms M_thigh1 M_thigh2 real
syms M_tibia1 M_tibia2 real
syms M_foot1 M_foot2 real

% link inertias
syms I_torso_xx I_torso_yy real
syms I_thigh_xx1 I_thigh_yy1 I_thigh_xx2 I_thigh_yy2 real
syms I_tibia_xx1 I_tibia_yy1 I_tibia_xx2 I_tibia_yy2 real
syms I_foot_xx1 I_foot_yy1 I_foot_xx2 I_foot_yy2 real

% center of mass offsets
syms c_torso real
syms c_thigh1 b_thigh2 real
syms c_tibia1 b_tibia2 real
syms c_foot1 b_foot2 real

% ------------------------------------------------------------------
%
%  Calculate kinetic energy
%
% ------------------------------------------------------------------

% positions of links, the reference point is the foot of the stance leg

p_CoMfoot1 = c_foot1 * [sin(q_ankle_pitch1 + q_knee1 + q_hip_pitch1) * ...
             cos(q_ankle_roll1 + q_hip_roll1); -sin(q_ankle_roll1 + ...
             q_hip_roll1); cos(q_ankle_pitch1 + q_knee1 + q_hip_pitch1) * ...
             cos(q_ankle_roll1 + q_hip_roll1)];

p_ankle1 = L_foot * [sin(q_ankle_pitch1 + q_knee1 + q_hip_pitch1) * ...
           cos(q_ankle_roll1 + q_hip_roll1); -sin(q_ankle_roll1 + ...
           q_hip_roll1); cos(q_ankle_pitch1 + q_knee1 + q_hip_pitch1) * ...
           cos(q_ankle_roll1 + q_hip_roll1)];

p_CoMtibia1 =  p_ankle1 + c_tibia1 * [sin(q_knee1 + q_hip_pitch1) * ...
               cos(q_hip_roll1); -sin(q_hip_roll1); cos(q_knee1 + ...
               q_hip_pitch1) * cos(q_hip_roll1)];

p_knee1 = p_ankle1 + L_tibia * [sin(q_knee1 + q_hip_pitch1) * ...
          cos(q_hip_roll1); -sin(q_hip_roll1); cos(q_knee1 + ...
          q_hip_pitch1) * cos(q_hip_roll1)];

p_CoMthigh1 = p_knee1 + c_thigh1 * [sin(q_hip_pitch1) * cos(q_hip_roll1); ...
              -sin(q_hip_roll1); cos(q_hip_pitch1) * cos(q_hip_roll1)];

p_hip = p_knee1 + L_thigh * [sin(q_hip_pitch1) * cos(q_hip_roll1); ...
        -sin(q_hip_roll1); cos(q_hip_pitch1) * cos(q_hip_roll1)];

p_CoMtorso = p_hip + c_torso * [sin(q_torso_pitch) * cos(q_torso_roll); ...
             sin(q_torso_roll); cos(q_torso_pitch) * cos(q_torso_roll)];

p_CoMthigh2 = p_hip + b_thigh2 * [sin(q_hip_pitch2) * cos(q_hip_roll2); ...
              -sin(q_hip_roll2); -cos(q_hip_pitch2) * cos(q_hip_roll2)];

p_knee2 = p_hip + L_thigh * [sin(q_hip_pitch2) * cos(q_hip_roll2); ...
```

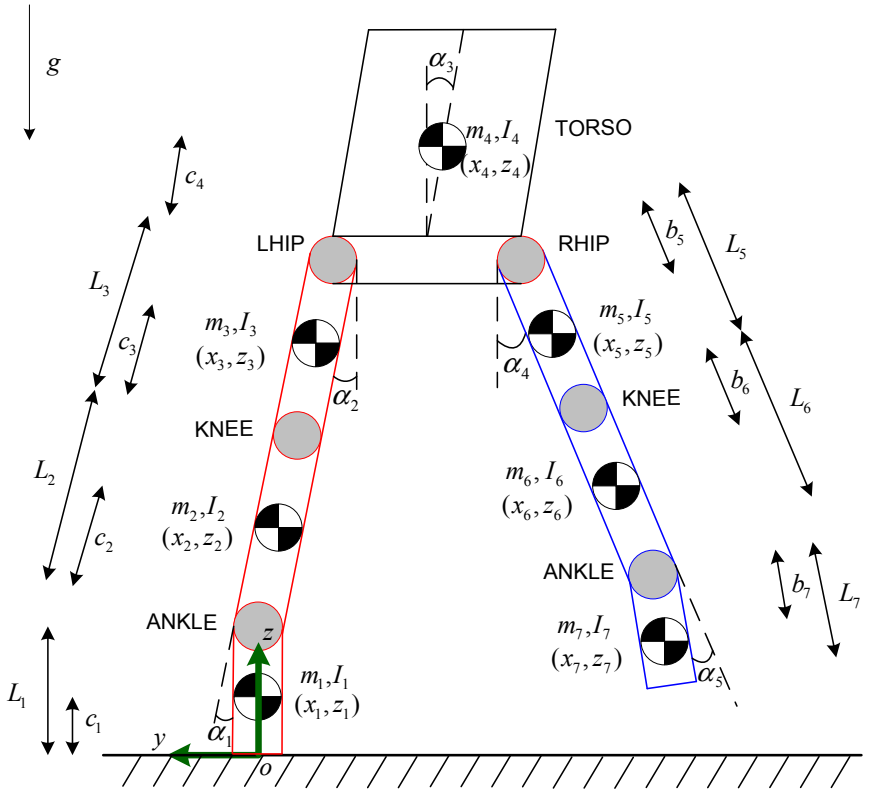
```
123           -sin(q_hip_roll2); -cos(q_hip_pitch2) * cos(q_hip_roll2)];
124
125 p_CoMtibia2 = p_knee2 + b_tibia2 * [sin(q_hip_pitch2 + q_knee2) * ...
126               cos(q_hip_roll2); -sin(q_hip_roll2); -cos(q_hip_pitch2 + ...
127               q_knee2) * cos((q_hip_roll2))];
128
129 p_ankle2 = p_knee2 + L_tibia * [sin(q_hip_pitch2 + q_knee2) * ...
130            cos(q_hip_roll2); -sin(q_hip_roll2); -cos(q_hip_pitch2 + ...
131            q_knee2) * cos((q_hip_roll2))];
132
133 p_CoMfoot2 = p_ankle2 + b_foot2 * [sin(q_hip_pitch2 + q_knee2 + ...
134              q_ankle_pitch2) * cos(q_hip_roll2 + q_ankle_roll2); ...
135              -sin(q_hip_roll2 + q_ankle_roll2); -cos(q_hip_pitch2 + ...
136              q_knee2 + q_ankle_pitch2) * cos(q_hip_roll2 + q_ankle_roll2)];
137
138 p_foot2 = p_ankle2 + L_foot * [sin(q_hip_pitch2 + q_knee2 + ...
139           q_ankle_pitch2) * cos(q_hip_roll2 + q_ankle_roll2); ...
140           -sin(q_hip_roll2 + q_ankle_roll2); -cos(q_hip_pitch2 + ...
141           q_knee2 + q_ankle_pitch2) * cos(q_hip_roll2 + q_ankle_roll2)];
142
143 % velocities of links
144
145 v_CoMfoot1 = jacobian(p_CoMfoot1, q) * dq;
146 v_CoMtibia1 = jacobian(p_CoMtibia1, q) * dq;
147 v_CoMthigh1  = jacobian(p_CoMthigh1, q) * dq;
148 v_hip = jacobian(p_hip, q) * dq;
149 v_CoMtorso = jacobian(p_CoMtorso, q) * dq;
150 v_CoMthigh2  = jacobian(p_CoMthigh2, q) * dq;
151 v_CoMtibia2 = jacobian(p_CoMtibia2, q) * dq;
152 v_CoMfoot2 = jacobian(p_CoMfoot2, q) * dq;
153
154 % kinetic energy of links, KE_ = KE_translation + KE_rotation
155
156 KE_foot1 = (1/2) * M_foot1 * v_CoMfoot1.' * v_CoMfoot1 + ...
157            (1/2) * I_foot_yy1 * (dq_ankle_pitch1 + dq_knee1 + ...
158            dq_hip_pitch1)^2 + (1/2) * I_foot_xx1 * (dq_ankle_roll1 + ...
159            dq_hip_roll1)^2;
160 KE_tibia1 = (1/2) * M_tibia1 * v_CoMtibia1.' * v_CoMtibia1 + ...
161             (1/2) * I_tibia_yy1 * (dq_knee1 + dq_hip_pitch1)^2 +...
162             (1/2) * I_tibia_xx1 * (dq_hip_roll1)^2;
163 KE_thigh1 = (1/2) * M_thigh1 * v_CoMthigh1.' * v_CoMthigh1 + ...
164             (1/2) * I_thigh_yy1 * dq_hip_pitch1^2 + ...
165             (1/2) * I_thigh_xx1 * dq_hip_roll1^2;
166 KE_torso = (1/2) * M_torso * v_CoMtorso.' * v_CoMtorso + ...
167            (1/2) * I_torso_yy * dq_torso_pitch^2 + ...
168            (1/2) * I_torso_xx * dq_torso_roll^2;
169 KE_thigh2 = (1/2) * M_thigh2 * v_CoMthigh2.' * v_CoMthigh2 + ...
170             (1/2) * I_thigh_yy2 * dq_hip_pitch2^2 + ...
171             (1/2) * I_thigh_xx2 * dq_hip_roll2^2;
172 KE_tibia2 = (1/2) * M_tibia2 * v_CoMtibia2.' * v_CoMtibia2 + ...
173             (1/2) * I_tibia_yy2 * (dq_hip_pitch2 + dq_knee2)^2 + ...
174             (1/2) * I_tibia_xx2 * (dq_hip_roll2)^2;
175 KE_foot2 = (1/2) * M_foot2 * v_CoMfoot2.' * v_CoMfoot2 + ...
176            (1/2) * I_foot_yy2 * (dq_hip_pitch2 + dq_knee2 + ...
177            dq_ankle_pitch2)^2 + (1/2) * I_foot_xx2 * (dq_hip_roll2 + ...
178            dq_ankle_roll2)^2;
179
```

```
% total kinetic energy

KE = KE_thigh1 + KE_tibia1 + KE_foot1 + KE_torso + KE_thigh2 + ...
     KE_tibia2 + KE_foot2;
KE = simple(KE);

% ------------------------------------------------------------------
%
%  Calculate potential energy
%
% ------------------------------------------------------------------

% positions of various members

PE_thigh1 = M_thigh1 * g * p_CoMthigh1(3);
PE_tibia1 = M_tibia1 * g * p_CoMtibia1(3);
PE_foot1 = M_foot1 * g * p_CoMfoot1(3);
PE_torso = M_torso * g * p_CoMtorso(3);
PE_thigh2 = M_thigh2 * g * p_CoMthigh2(3);
PE_tibia2 = M_tibia2 * g * p_CoMtibia2(3);
PE_foot2 = M_foot2 * g * p_CoMfoot2(3);

% total potential energy

PE = PE_thigh1 + PE_tibia1 + PE_foot1 + PE_torso + PE_thigh2 + ...
     PE_tibia2 + PE_foot2;
PE = simple(PE);

% ------------------------------------------------------------------
%
%  Calculate model matrices
%
% ------------------------------------------------------------------

L = KE - PE;
L = simple(L);

% gravity vector

G = jacobian(PE, q).';
G = simple(G);

% mass-inertial matrix

H = simple(jacobian(KE, dq).');
H = simple(jacobian(H, dq));

% Coriolis and centrifugal matrix

syms C real
n = max(size(q));
for k = 1 : n
  for j = 1 : n
    C(k,j) = 0 * g;
    for i = 1 : n
      C(k, j) = C(k,j) + 1/2 * (diff(H(k,j), q(i)) + ...
                diff(H(k,i), q(j)) - ...
```

```
237                 diff(H(i,j), q(k))) * dq(i);
238     end
239   end
240 end
241
242 C = simple(C);
243
244 % input matrix, Phi_0 is the actual joint angles for NAO, q_torso = 0
245
246 Phi_0 = [q_ankle_roll1;
247          q_ankle_pitch1;
248          q_knee1;
249          q_hip_pitch1 + q_torso_pitch;   % to be confirmed
250          q_hip_roll1 - q_torso_roll;     % to be confirmed
251          q_hip_roll2 + q_torso_roll;     % to be confirmed
252          q_hip_pitch2 + q_torso_pitch;   % to be confirmed
253          q_knee2;
254          q_ankle_pitch2;
255          q_ankle_roll2;
256          q_torso_pitch;
257          q_torso_roll];
258 B = jacobian(Phi_0, q);
259 B = B.' * [eye(10, 10); zeros(2, 10)];
260
261 % swing foot force input matrix
262
263 Phi_1 = p_foot2;
264 E2 = jacobian(Phi_1, q).';       % E2 = J^T
```



(a)



(b)

Figure D.2: Dynamic seven-link model of the robot during the SSP. (a) Sagittal plane view (b) Frontal plane view.