國立臺灣大學工學院機械工程學研究所

碩士論文

Department of Mechanical Engineering

College of Engineering

National Taiwan University

Master Thesis

# 雙足機器人之控制與自然步態

# Control and Natural Walk of a Biped Robot

紀孟谷

Meng-Ku Chi

指導教授：黃漢邦 博士

Advisor: Han-Pang Huang, Ph.D.

中華民國 100 年 6 月

June 2011

# 誌謝

在台大兩年的研究生活中，經歷了很多開心歡笑和痛苦挫折，也因為度過了這兩年時光，更加磨練了我對團體生活還有獨立研究的看法還和心智。

在很多需要感謝的人之中，首先我要先感謝三位不辭辛勞從外校來幫我做口試審查的教授：劉正良教授、顏鴻森教授、蔡得民教授；在兩年的時光中，一直指導我的指導教授：黃漢邦教授，謝謝您這兩年來的指導和愛護，雖然我始終在面對你時還是有著一份害怕的心態，不過逐漸已經成為尊敬。

在實驗室的同學裡，感謝像哥哥也像老師的嚴舉樓學長，這兩年來受你照顧太多，不知道要如何感謝您才得體，也許祝福您畢業和身體健康是我不管在哪都能祝福你的事；也感謝柏霆，子豪，聖諺，毓文，登湖，和相元學長的照顧，沒有你們，實驗室會少一分色彩；助理潘潘，美足，馨儀在行政和生活上幫助我；同屆的政修，晞帆，柏緯，宜霖，謝謝您們這兩年來的打氣和努力，也恭喜您們畢業了；學弟妹們，泓逸！你是最棒的；啓舜，繼續努力加油；聖翔，你的責任心和努力我都有看到；瑋軒，你也要好好加油；王璟萍，祝妳早日畢業，秀婷和衍文軟體組靠你們了；朋友方面最要感謝的是台中某隻貓，你的鼓勵和打氣是我求學過程中最寶貴的；最後要謝謝這兩年依舊一直陪我的女友，妳總是陪在我身邊，也期待你早日畢業。

紀孟谷/2011/07 於機器人實驗室

# 中文摘要

本論文主要是研究人類的走路姿態並且應用在本實驗室開發的人型機器人。研究主要是建立可應用的零力矩點的軌跡、重心在垂直方向的軌跡、以及動量補償器。

一般應用於人型機器人的零力矩點軌跡集中在腳底板中央，並且在換支撐腳時軌跡瞬間移動到此支撐腳，因此容易造成速度和加速度不連續；所以我們提出改善此零力矩點軌跡。藉由改善過後的軌跡可以讓人型機器人行走時更加穩定。利用 Preview Control 生成的重心軌跡可以自由地調整重心在垂直方向的位置，使人型機器人不必總是彎著膝蓋走路，減少能量的消耗。人型機器人的自然步態規劃綜合了上述的控制器，再對逆運動學的演算法改善。本論文提出兩個方法來達到腳趾彎曲和腳跟著地，再將兩個方法做模擬和比較。

在物理環境模擬方面我們利用 ADAMS 進行，而所有的控制程式皆在 MATLAB 上撰寫，兩者的連接是用 MATLAB 上的 Simulink。實作方面，我們也自行開發人型機器人，搭配新設計的腳底板結構。所有機構皆由 SolidWorks 和 CATIA 來設計並作應力分析。

**關鍵字**: 零力矩點，變化的重心高度，Preview Control，自然步態，逆運動學，腳趾彎曲，腳跟著地

# Abstract

This thesis examines the walking motion of human beings and applies its findings to a humanoid robot developed by our laboratory. Our goal is to construct the usable Zero Moment Point (ZMP) trajectory, a Center of Gravity (COG) trajectory in the vertical direction, and momentum compensation.

Conventional ZMP trajectories applied to humanoid robots are usually located at the center of each foot pad, shifting instantaneously to the new supporting leg as support changes from one foot to another. Velocity and acceleration become unsmooth. We used Preview Control to generate a COG trajectory with the ability to arbitrarily adjust position in the vertical direction. Observations of human walking motion enabled us to plan COG trajectory with continuous smooth change of velocity and acceleration. The robot now has no need to keep its knee joints constantly bent, and therefore consumes less power. Its natural walk is a result of integrating the adjusting ZMP and COG trajectory controls by using the modified inverse kinematics algorithm. This thesis proposes two methods to derive the toe-off and heel-contact motions necessary for a natural walk. The proposed algorithms are justified through simulation and experiments.

Our simulation physical environment was constructed on MSC ADAMS, and all controlling functions were built in MathWorks MATLAB. The two software environments were connected by Simulink in MATLAB. We also developed a humanoid robot with new foot pads to generate a natural walk. All mechanisms were designed in Dassault Systems SolidWorks, and stress analysis performed using Dassult Systems CATIA.

**Keywords**: ZMP, Varying COG Height, Preview Control, Natural Walk, Inverse Kinematics, Toe-off Motion, Heel-contact Motion

# Contents

# List of Tables

# List of Figures

# Nomenclature

**Notations**

| | |
|---|---|
| **Coordinate System** | |
| $\{R\}$ | coordinate system of robot |
| $\{W\}$ | coordinate system of world |
| **Inverse Kinematics** | |
| $h$ | threshold value |
| $x$ | robot position |
| $m_i$ | mass of link i |
| $H(\theta)$ | performance criterion |
| $M$ | mass |
| $J_e$ | Jacobian matrix for each joint |
| $J_{COG}$ | COG Jacobian |
| $J^+$ | pseudo inverse |
| $W$ | diagonal weighting matrix |
| $\alpha$ | positive damping factor |
| $\theta$ | joint angle of robot |
| **ZMP Analysis , Varying COG, Momentum Compensation** | |
| $x_{ZMP}$ | ZMP position in x-direction |
| $y_{ZMP}$ | ZMP position in y-direction |
| $x_{toe}$ $x_{heel}$ | positions of toe and heel in x-direction |
| $y_{toe}$ $y_{heel}$ | positions of toe and heel in y-direction |
| $COG_Z$ | COG Height |
| $J_{mom}$ | momentum Jacobian |
| **Natural walk** | |
| $L$ | Foot pad length |
| $J_o$ | known Jacobian matrix from known trajectory |
| $z(t)$ | position of heel in z-direction at time t |
| $\dot{\theta}_{Rtoe}$ $\dot{\theta}_{Ltoe}$ | angular velocity of toe joint |
| $\theta_{toe,t}$ | angle of toe at time t |

# Chapter 1 Introduction

## 1.1 Motivation

A humanoid robot, similar in appearance to a human being, can walk with two legs and perform tasks with two arms. They are designed in this way so that they can serve humans in their daily living environment. Such kind of humanoid robots are useful in aging society [14]. Robots are also useful for performing tasks in conditions and environments that are hostile to humans. For example, the 311 earthquake off Japan made the Fukushima Daichi nuclear power station a very hostile radioactive environment in which human beings could work for very short periods. The USA dispatched a robot, PACKBOT [74], to observe the center of the nuclear plant. The National Aeronautics and Space Administration (NASA) also uses robots to search unknown planets.


(a)


(b)

Figure 1-1 (a) Service humanoid robot, ASIMO [84] (b) PACKBOT [85]: exploring robot.

Many people ask why robots are designed in the shape of humans. There is need

for machines that can perform specific tasks, but their repertoire is very small and their tasks must be known before design starts. A humanoid robot is a general-purpose machine, adaptable to a wide range of tasks that may not even have been thought of when the robot was designed.

Many robots nowadays are humanoid in neither appearance nor motion. They may be wheeled, quadruped, hexapod, or snake-like, and their high mobility allows them to perform tasks well. Even so, humans still feel a need, possibly stimulated by movies and novels, to humanoid robots. Movies like *"The Terminator"* and the *"Star Wars"* series motivate dramatic development of humanoid robots, which spring up like bamboo shoots after a spring rain. Ever more impressive and human-like robots appear, like those in *"Transformer"* and *"Avatar"*. They may be fictitious, but they persuade us to believe that they could be developed in real life.

Even given this belief, it is hard to design a humanoid robot with its physical functions exactly equivalent to those of a human. For example, a robot's walking mechanism must be different from a human's, because its stability and human-like motion cannot be governed in the absence of the human learning algorithm (generated in the cerebrum) and the human sense of equilibrium (generated in the cerebellum). For robots to walk in a convincing manner, therefore, it is critical to control the high nonlinearity for 30-40 degrees of freedom (DOFs) as well as the inherent instability

(inverted pendulum).

## 1.2 Humanoid Robots

To appear like human, robots must walk on two legs; i.e. they must have biped locomotion. Several control methods and algorithms have been developed in this field. When analyzing biped locomotion, we separate it into four categories- Planning, Stabilization Control, Gait Generation, and Trajectory Generation, as shown in Figure 1-2. This thesis covers the areas shown in blue.

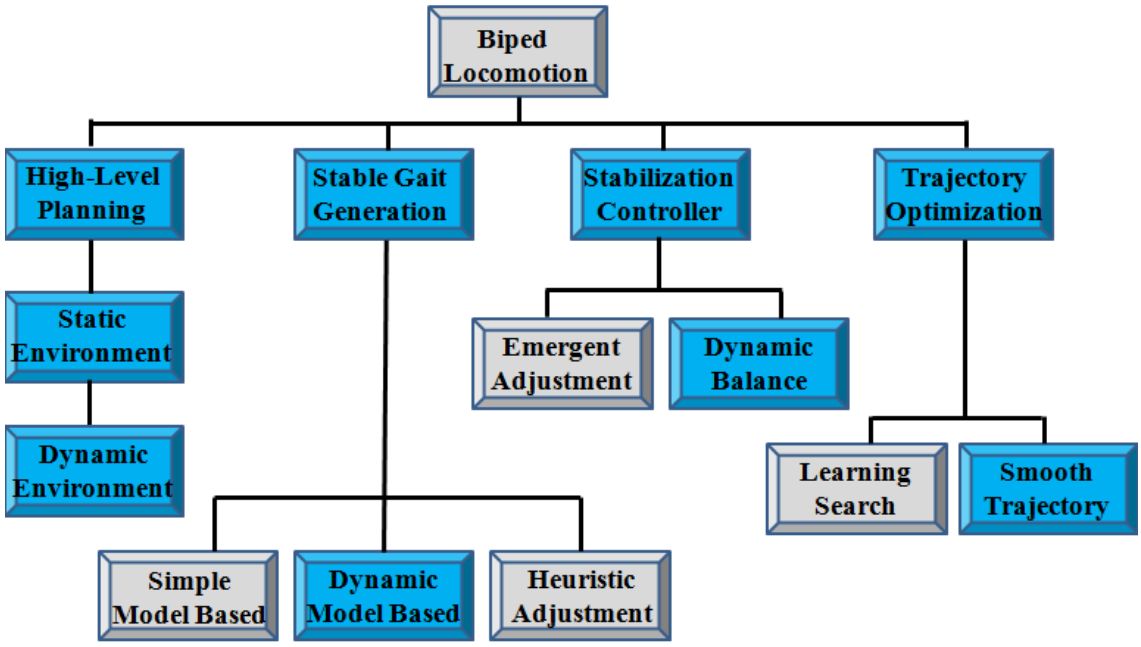



Figure 1-2 Classification of researches for biped locomotion.

High-level planning is mainly concerned with collecting and fixing the pattern of walking. That enables us to plan the path and goal appropriately, so we can take the planned information into the environment. The planning result combines several low-level planning motions, including walking in straight line, in curves, up stairs, with

different stride lengths, and at different speeds [32]. The information from high-level planning enables the accomplishment of stable gait generation, thus ensuring that the robot walks with asymptotic stability. The stabilization controller uses several algorithms such as the ankle stabilizer and ZMP limiter. Trajectories such as COG and ZMP for each joint are optimized or fixed, thus enabling the robot to walk more smoothly and stably—more humanly.

### 1.2.1 Development of Humanoid Robots

In 2010, HONDA corporation celebrated birthday of 10 years old humanoid robot "ASIMO" [12]. ASIMO is one of the successful humanoid robots. Its height is 130 cm and weight is 54 kg, and it has 34 DOFs. This remarkable robot can do lots of human motions including walk, run, waving hands, nod, etc. ASIMO really impresses on most of researchers in robotic territory. HONDA humanoid robot triggered the world's research on humanoid robots. Therefore, new humanoid robots has been fabricated, such as HRP-2, HRP-3,HRP-4, HUBO [54], WABIAN, and so on.

The National Institute of Advanced Industrial Science and Technology (AIST) develops the humanoid robots of HRP series. Different from the ASIMO, HRP series adopt the feminine style, and the appearances are similar to the human beings. HRP-2 [24, 26] was a remarkable humanoid robot whose height is 154 cm and weight is 58 kg. This robot not only has the cooling system for longer motion but also utilize FEM

(finite element method) to reduce mechanical resonance. HRP-2 is a prototype for subsequent HRP series robots. HRP-3 [27] was improved some problems of HRP-2, and it had been designed to work outdoors. The latest HRP series was HRP-4C [28] which was designed in 2009. This robot has a fashionable body shape, and its human-like face does impress robotic researchers.

Korea Advanced Institute of Science and Technology (KAIST) developed a humanoid robot, HUBOⅡ [54]. Its appearance is similar to predecessor, HUBO (KHR3), and it has 40 DOFs for various motion. HUBOⅡ only weights 45 kg which is lighter than his brother HUBO (55kg), and capable of walking two times faster (1.4km/h). It also can run up to 3.3 km/h.

WABIAN-2 [49], was designed by Waseda University, and it can walk much like the human beings. Since WABIAN-2 has toe joints and heel joints, it can walk with toe-off and heel-contact. Moreover, WABIAN-2 has additional joint in waist, and make its knee joints more straight during walking period. This robot adopted the genetic algorithm to generate the joint trajectory, and the walking motions are similar to the human beings.

Boston Dynamics designed an anthropomorphic robot, PENTMAN [82], for testing chemical protection clothing. It was used by the US Army. This robot had to be supported mechanically and had a limited repertoire of motion. The predecessor, BIG

DOG, was also fabricated for delivering goods and materials during warfare. Although PENTMAN's robust balance is not as good as BIG DOG, it has a walking speed of up to 7.2 km/h (4.4 mph).

Unlike most of the biped robots, DLR developed a biped robot, DLR-Biped [51], which walked with torque joints. This modular structure, DLR-KUKA Lightweight-Robot (LWR), can keep the development time and cost low. It has 6 degrees-of-freedom each legs, and has a walking speed of up to 0.15m/s.

LOLA is an anthropomorphic autonomous humanoid robot [40] which was developed by Technical University of Munich. This robot has 25 DOFs for performing various tasks. In order to generate high-speed walking, further design goals can be improved, such as high center of mass and low moments of inertia of the leg links. By investment casting, this robot largely reduces the weight, but it still has the humanoid appearance (height :180cm). Although LOLA is fabricated with aluminum, the strength of mechanism is very solid.

### 1.2.2 Stable Walking for Humanoid Robots

Stability is critical for a humanoid robot to walk. Locomotion is governed by a planning process to determine stride length, positions of end-effectors, and stability in motion. Most current researchers believe that a combination of high-level dynamic algorithms and appropriate mechanisms will help achieve stability.

The use of a simple model is advantageous as it allows rapid calculation and real-time implementation. Kajita et al. adopted the Linear Inverted Pendulum Model [21, 22, 23] and Minakata et al. proposed the Virtual Inverted Pendulum Method [42]. Both methods can simplify models for easy calculation, however, such simplification may allow modeling errors to render biped robots unstable in certain circumstances. It is therefore necessary to consider the limitations of these models.

Some researchers, lacking the needed stability algorithm, generated locomotion through trial and error. Lee et al. [36] adopted an evolution algorithm (EA) to determine walking parameters, while Kuffner et al. [33] proposed a heuristics search to generate a sequence of footstep locations. Although they are very time-consuming, the contribution made by these heuristics search methods for biped robots is still significant. By constructing a huge database and developing feasible experiences for robots, researchers have enabled robots to perform more actions of different types.

### 1.2.3 Human-Like Walking

In recent years, an ever increasing number of humanoid robots can walk stably, prompting the belief that, just because they are labeled humanoid, they can walk or run just like human beings. They, however, walk with bent knees because of the limitations imposed by modeling or hardware [12, 22, 25, 48]. Sano et al. [59] used the inverted pendulum's angular momentum to generate ankle torque. This was a start to recognize

that human feet are not permanently parallel with the ground during locomotion. Wikipedia [88] stated that human gait consists of (1) forefoot strike, (2) midfoot strike, and (3) heel strike. This definition separates the foot into toe, midfoot, and heel, each with its own functions Some researchers have therefore begun to study the influence of toe and heel [66, 71], and, unlike earlier researchers, have introduced the viewpoint of using the straight knee for part of the robotic stride [38, 41, 50]. Of the four postures used in human locomotion, only one (midfoot strike) occurs in conventional humanoid robots, whereas forefoot strike (toe-off motion), heel strike (heel-contact motion), and straight knee are all missing.

There are many advantages to implementing natural walking. Stride length, limited in conventional humanoid robots because of their constantly parallel-to-the-ground feet, can be lengthened by adding toe and heel functions. By straightening the knee for part of the stride cycle, power consumption can be reduced.

The disadvantages of adding these functions are potential loss of stability resulting from the small contact area when heel or toe is the only part of the foot on the ground. This presents a great challenge, in particular maintaining the stability of a walking robot.

## 1.3 Thesis Organization

This thesis is organized into seven chapters. They are shown in Figure 1-3.

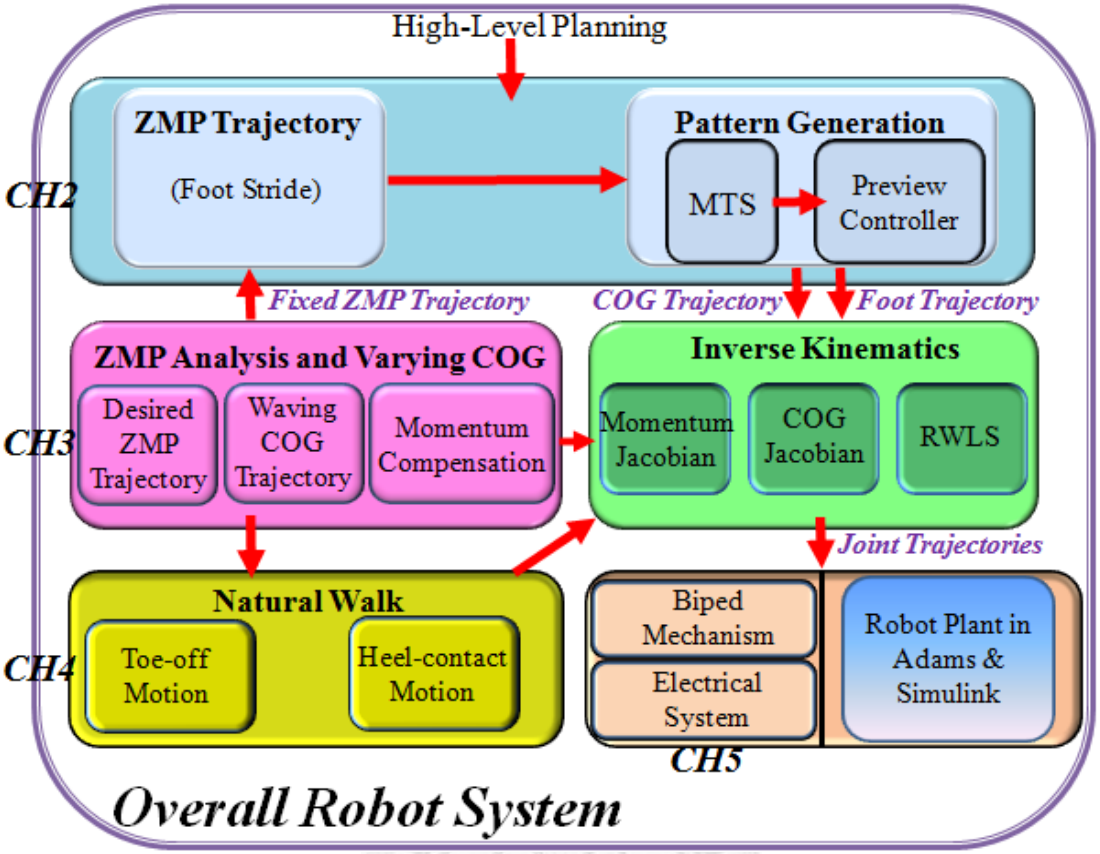



Figure 1-3 Thesis structure.

In this chapter, Chapter 1, we have briefly summarized the development of humanoid robots and introduced the challenges to make them walk naturally, as humans do.

Chapter 2 provides the background knowledge needed to overcome the challenge of biped locomotion. It will include the Zero Moment Point (ZMP) method, the Inverse Kinematics (IK) solver, and the preview controller to vary the Center of Gravity (COG).

Chapters 3 and 4 form the kernel of this thesis, describing our results and the simulations we used to prove our method. Chapter 3 explains the need to plan the desired ZMP trajectory for humanoid robots, so they can walk, as humans do, with

varying COG height and momentum compensation. Chapter 4 proposes two methods for natural walking. After comparing the results, we adopted the “Toe Motion without Extra DOFs” method to generate toe-off and heel-contact motions.

Our laboratory designed a humanoid robot, Voyager, to implement and test the proposed methods. Its feet consists of specially designed toe and heel joints to enable natural locomotion. The new foot mechanisms were designed with stress analysis. All actuators are communicated through a Controller Area Network (CAN) bus. “Voyager’s” hardware is discussed in Chapter 5.

Chapter 6 details the experiment and implementation results, while Chapter 7 provides conclusions and some suggestions for future research.

## 1.4 Contributions

A. Since the conventional ZMP trajectory caused discontinuous acceleration or jerk, we modified it to enable us to plan and implement a more natural walk.

B. We also considered the effect of COG on walking. We found that varying the COG height led to a reduction of redundant power consumption. On the other hand, the practice of momentum compensation made our humanoid robot walk much similar to humans.

C. We proposed two methods to realize the toe-off motion. After comparing their pros and cons, we selected the better of the two to perform the heel-contact. Since the

robot end-effector trajectory is generated by MTS, the dropping value of the swinging leg can be anticipated before contact occur, allowing us to propose a compensatory way to realize the heel-contact motion. Once the toe-off and heel-contact motions had been combined, natural locomotion could be constructed.

D. We designed a new foot pad with toe and heel joints. In order to ensure its strength, we performed stress analysis in CATIA. The stress results give us real hope that we will be able to reduce the weight of the mechanism.

E. To prove our theory, we took the joint trajectories into our humanoid robot, Voyager. The humanoid robot is also loaded with the new foot pads. With the stable robot system, our robot really can realize the toe-off and heel-contact motions.

# Chapter 2 Dynamic Walking Generation

## 2.1 Introduction

This chapter introduces some fundamental concepts of biped humanoid robots. Walking issues of humanoid robots are different from human walking. However, the human walking motion can help us to construct the biped robot walking system

Previously, biped robot's locomotion is static walking which the projection of center of gravity (COG) on the ground should be always located inside the foot support area. However, the dynamic walking can be implemented with less motion constraints and let the COG move smoothly. Although the walking speed is increasing, the issue of stability rises.

Among many researches of dynamic walking, the stability is a critical point. To satisfy the stability, Zero-Moment Point (ZMP) is perhaps the most important method for controlling the biped robot walking motion. The basic concept of ZMP method is let ZMP of the biped robot stay inside the support polygon. Then we control the whole motors of the biped robot and change the posture which directly influences the locomotion of the COG and end-effectors. By using the car-table model, we can connect the ZMP and the COG dynamic relationship. Based on ZMP and COG relationship, we

plan the COG trajectory via pattern generator using preview control [19].

Kinematics is a method that links the Cartesian space with the joint space, since the end-effectors trajectories are planned in Cartesian space. By solving the inverse kinematics, we can map desired trajectory of COG and footstep to the joint space. Due to some drawbacks in inverse kinematics, we adopted robust damped least square method (RWLS) [79]. The structure of dynamic walking motion generation is shown in Figure 2-1.

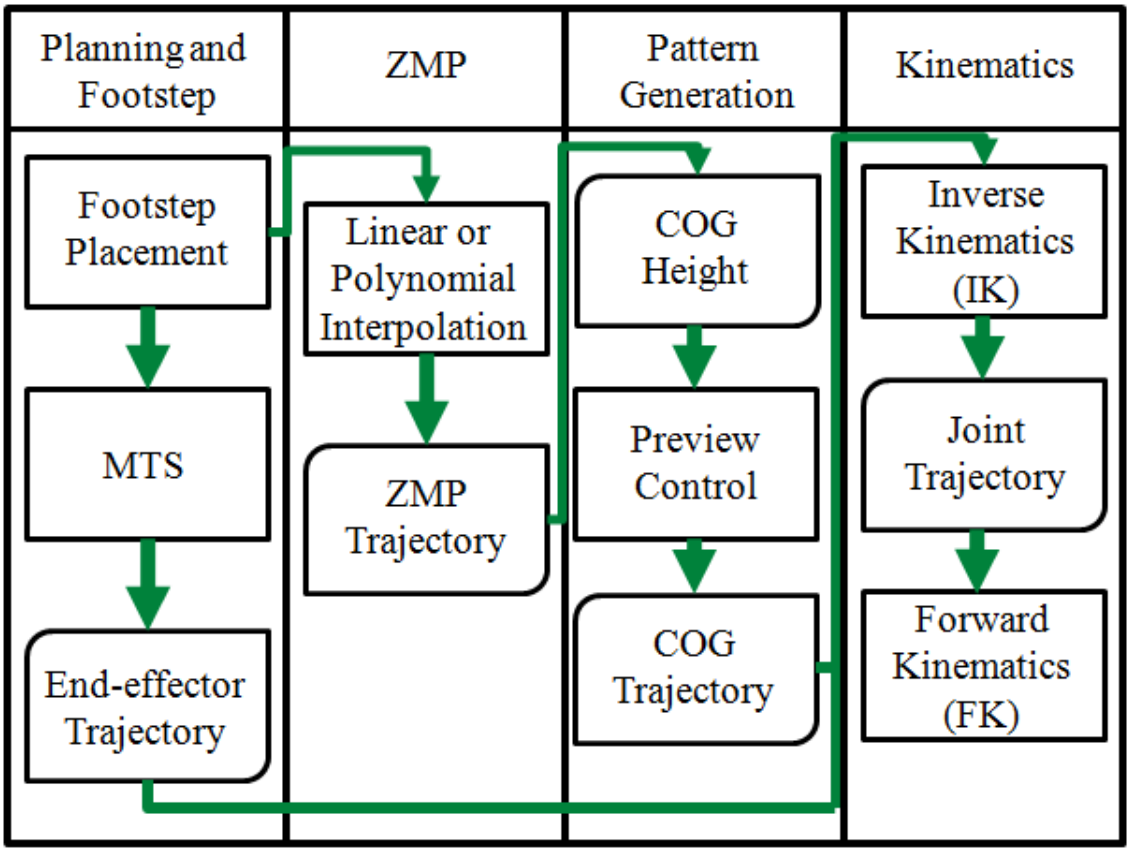



Figure 2-1 Dynamic locomotion generation.

## 2.2 Forward Kinematics

Our robot's movable joints are based on joint space. However, the COG trajectory

and end-effector trajectory are established in the task space. Therefore, forward kinematics system and inverse kinematics system must be utilized.

Before we apply the method, we should know how to describe the structure of the robot. Hence, we must construct the parameters of robotic system, which involves the orientation of each joint, the length and angle of each link, and the limited movable range of each joint. Based on Devanit-Hartenberg (DH) [68, 72] representation, we converted four independent parameters into a mathematical formula. By applying this formula, we can find a transformation matrix to establish the structure of robot.

For biped robot, we need to describe the relations between each joint and each end-effector in the given coordinate system $\{R\}$ on the robot. We make the landing leg as the coordinate origin. By moving the robot's coordinate system $\{R\}$, we determine the robot's pose in the world.

During walking period, the biped robot changes the coordinate system $\{R\}$ all the time. The link between the robot coordinate system $\{R\}$ and the world, or so-called task space and noted as $\{W\}$, is quite important. Thus, we assume that the origin of $\{R\}$ always changes with $\{W\}$. The relationship between $\{R\}$ and $\{W\}$ is shown in Figure 2-2.

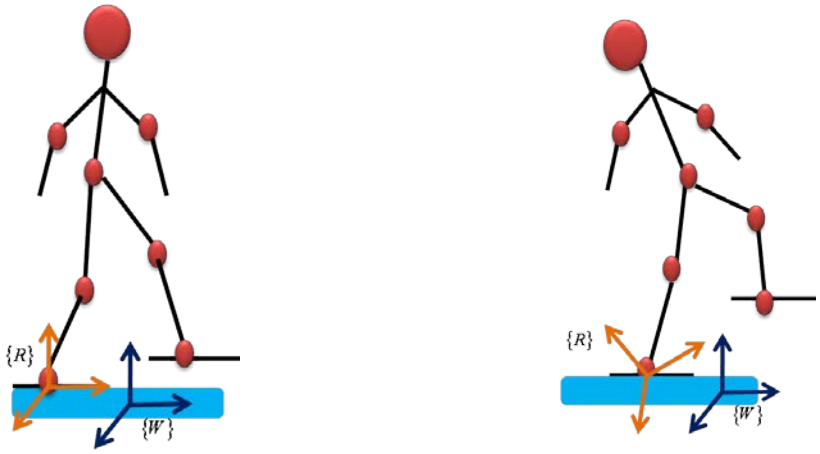



Figure 2-2 The relationship between $\{R\}$ and $\{W\}$.

## 2.3 Inverse Kinematics

For describing the position, we usually use the task space. In contrast to forward kinematics, inverse kinematics (IK) is the computation from the motion of the end-effector back to joint angles. In other words, inverse kinematics converts the task space into joint space. The end-effector motions are decomposed to a set of positions which are computed by IK-solver. In order to fit the setting we select the dynamic simulation software ADAMS. Since motions of the end-effector are divided into s set of points, the distance of adjacent points should be close enough. If the distance is not close enough, the IK-solver might spend more computation time and the robot's motions in simulation might not be smooth. In order to overcome this problem, some interpolation algorithms may be employed, such as polynomial fitting, cubic spline fitting, conventional tension splines (CTS), Lagrange interpolations, Non-uniform rational basis spline (NURBS), and modified tension splines (MTS) [15]. In our

research, we apply the MTS developed by our laboratory for the trajectories interpolation of the end-effector.

The IK-solver for computing Jacobian matrix is introduced in Section 2.3.1. To solve the singularity configuration problem in IK, the singularity configuration. Therefore, we performed singularity avoidance method will be described in Section 2.3.2 to avoid full stretch of the links.

## 2.3.1 COG Jacobian and End-Effector Jacobian

It is required to compute the Jacobian matrix in IK. The end-effector's position and orientation $x \in R^m$ are described in task space, and the angles of joints $\theta \in R^n$ are described in joint space. The relationship between $\theta$ and $x$ is given in Eq. (2-1). If we differentiate Eq. (2-1) with respect to time, the velocity Jacobian becomes Eq. (2-2).

$$x = f(\theta) \tag{2-1}$$

$$\dot{x} = J_e(\theta)\dot{\theta} \tag{2-2}$$

where $J_e(\theta) \in R^{mn}$ is the Jacobian matrix. The suffix $e$ presents the contribution of each joint to the end-effector. When a desired end-effector velocity is given ,we can find the corresponding joint angle velocity.

Only utilizing the end-effector Jacobian is not enough for biped robot. The COG Jacobian must be taken into account. If we do not care about the angle velocity contribution of each joint to the change rate of COG, the robot might fall. In order to

match the stable condition, we still can add the influence of COG trajectory into Jacobian, namely, the Jacobian matrix of COG, $J_{COG}(\theta)$. The detailed introduction of COG Jacobian matrix is described in [75].

Before combining the COG Jacobain into the IK-solver, we need to realize that there is difference between the swing limbs and the fixed limbs at COG Jacobian matrix. The limbs include arms, legs, and head. The COG Jacobian matrix of swing limbs, $J_{limb \to COG}(\theta)$, can be expressed as (2-3).

$$J_{limb \to COG}(\theta_{limb,j}) = \frac{M_{limb,j}}{M} \dot{\theta}_j \overrightarrow{Z_j} \times \overrightarrow{r_{j \to COG}} \tag{2-3}$$

where $M$ is the mass of the robot, $M_{limb,j}$ indicates a part of mass which will be moved when the joint $j$ rotates. The direction vector of the rotating axis of joint $j$ is $\overrightarrow{Z_j}$, and the $\overrightarrow{r_{j \to COG}}$ is a vector of rotating joint $j$ to COG. Note that all the vectors are computing with respect to world space.

On the other hand, the Jacobian matrix of fixed limb, $J_{fixed \to COG}(\theta)$ is different from the swing limb. When a joint rotates on the fixed limb, the upper parts of the rotating joints will rotate. And the lower parts do not move. Therefore, the angular velocity direction of this fixed limb joint is opposite to swing limb's joint. The sign of Jacobian matrix of the fixed leg rather than the swing leg is negative. Here the equation of fixed limb Jacobian matrix is expressed as

$$J_{fixed\to COG}(\theta_{fix,j}) = -\frac{M_{fixed,j}}{M}\dot{\theta}_j \overrightarrow{Z_j} \times \overrightarrow{r_{j\to COG}} \tag{2-4}$$

Integrating the Jacobian matix of the fixed limbs, swing limbs, and COG Jacobian matix, we can derive the ideal Jacobian matrix to solve the IK problem. By performing this matrix, we can obtain the relationship between the velocity of the end-effector and the angular velocities of all joints as

$$\begin{bmatrix} \dot{x}_{fixed} \\ \dot{x}_{swing} \\ \dot{x}_{LArm} \\ \dot{x}_{RArm} \\ \dot{x}_{Head} \\ \dot{x}_{COG} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_{fixed} \\ \dot{\theta}_{swing} \\ \dot{\theta}_{LArm} \\ \dot{\theta}_{RArm} \\ \dot{\theta}_{Head} \end{bmatrix} \tag{2-5}$$

$J_e(\theta)$

$$J = \begin{bmatrix} J_{fixed} & 0 & 0 & 0 & 0 \\ J_{f\to swing} & J_{swing} & 0 & 0 & 0 \\ J_{f\to LArm} & 0 & J_{LArm} & 0 & 0 \\ J_{f\to RArm} & 0 & 0 & J_{RArm} & 0 \\ J_{Head} & 0 & 0 & 0 & J_{Head} \\ J_{fixed\to COG} & J_{swing\to COG} & J_{LArm\to COG} & J_{RArm\to COG} & J_{Head\to COG} \end{bmatrix}$$

where the suffix *fixed* expresses the fixed leg or landing leg. Note that the limbs mean the head, arms, and leg here. Hence, the $J_{f\to limb}$ represents the coupling contribution of fixed leg to other limbs. According to this configuration, the upper body which involves the arms and the head will affect the COG. However, the $J_{LArm\to COG}$, $J_{RArm\to COG}$, and $J_{Head\to COG}$ are equal to zero. Namely, it can work well without the upper body influence. Hence, it will save much time and make the computation easier.

We already have the relationship between the velocities of the end-effectors and

the angular velocities of all joints. If we give the known the velocity of the end-effector, we can obtain the angular velocity of joints by inversing $J(\theta)$. Nevertheless, the inversing Jacobian matrix $J^{-1}(\theta)$ may not exist when the system is a redundant system. Therefore, we can use pseudo inverse of $J(\theta)$ to obtain the inverse as

$$J^{+} = J^{T}(JJ^{T})^{-1} \tag{2-6}$$

Although $J^{+}$ ensures the inverse when the matrix $J(\theta)$ is not a square matrix, the high angle velocities of joints may occur. In this singularity condition, we adopt the Robust Damped Least-Square method (RDLS) and Weighted Least-Square method (WLS). In other words, we want to constrain the robot motions to avoid the robot hitting itself. By adopting these methods, we can solve the problem from computing inverse kinematics.

### 2.3.2 RDLS: Singularity Avoidance

In fact, if there is a redundant joint being added to the system, it still exists singular points in some situations [8]. This issue can be solved by minimizing a performance index Eq. (2-7).

$$P = \left\| \dot{x} - J\dot{\theta} \right\|^{2} + \alpha \left\| \dot{\theta} \right\|^{2} \tag{2-7}$$

Here we focus on the residual error and angle velocity. The symbol $\alpha$ is a positively damping factor, and $P$ is a cost function. Therefore, the formula of DLS becomes Eq. (2-8)..

$$J^{+} = J^{T}(JJ^{T} + \alpha I_{m})^{-1} \tag{2-8}$$

where $I_m$ is an identity matrix (the dimension is m by m). According to this formula, the value of damping factor will be very small at nonsingular configuration. On the contrary, at singular configuration the value of $\alpha$ becomes larger. Thus, the timing of switching $\alpha$ is quite important. Nakamura et al. [39] proposed an automatically adjusting method as

$$\alpha \begin{cases} \alpha_0(1 - h_1 / h_1^s) & \text{if } h_1 < h_1^s \\ 0 & \text{otherwise} \end{cases} \tag{2-9}$$

where

$$h_1(\theta) = \sqrt{\det(JJ^T)} \tag{2-10}$$

the $h_1^s$ indicates a threshold value of the boundary of the neighborhood of singular points, and $\alpha_0$ is a value of the damping factor when the system's solution are located at singular points. From these two equations, the damping factor $\alpha$ will become large enough to avoid singularity as $h_1$ is smaller than the threshold value $h_1^s$. The damping factor is equal to zero at nonsingular configuration.

In particular, it needs to be mentioned that when $h_1$ is larger than $h_1^s$, the value of $\alpha$ cannot be zero since it will generate some discontinuous condition in some practical applications. Hence, the value of $\alpha$ need to be chosen as almost zero but not equal to zero.

The problem of singularity has been solved. However, some poses does not reach the singularity points ,but the robot maybe knock itself. We adopt a method to solve the problem in next section.

### 2.3.3 WLS: Joint Limitation Avoidance

For a redundant system, there are infinite solutions of joint angles when we give a position and orientation of the end-effector. The system is usually utilized at biped or humanoid robot. And the general formula can be expressed as:

$$\dot{\theta} = J^{+}\dot{x} + (I - J^{+}J)\varphi \tag{2-11}$$

where $\varphi \in R^{n}$ is a vector which we can arbitrarily design. The term of $(I - J^{+}J)$ is a projection operator which will projects $\varphi$ into the null space of $J$. Note that the $(I - J^{+}J)\varphi$ is a homogeneous solution and $J^{+}\dot{x}$ is a particular solution. We substitute Eq. (2-8) into Eq. (2-11) to obtain.

$$\dot{\theta} = J^{T}(JJ^{T} + \alpha I_{m})^{-1}\dot{x} + (I - J^{T}(JJ^{T} + \alpha I_{m})^{-1}J)\varphi \tag{2-12}$$

For Gradient Projection Method (GPM) the term of $\varphi$ will be substituted by $\nabla H(\theta)$, and we have

$$\dot{\theta} = J^{+}\dot{x} + \kappa(I - J^{+}J)\nabla H(\theta) \tag{2-13}$$

where $\kappa$ is a real scalar number, and $H(\theta)$ is a performance criterion. The $\kappa$ is chosen as a positive value when $H(\theta)$ is maximized, and a negative number when $H(\theta)$ is minimized. In this formula of joint limit avoidance, the performance criterion

can be defined as Eq. (2-14):

$$H(\theta) = \sum_{i=1}^{n} \frac{1}{4} \frac{(\theta_{i,\max} - \theta_{i,\min})^2}{(\theta_{i,\max} - \theta_i)(\theta_i - \theta_{i,\min})} \tag{2-14}$$

According to this definition, $H(\theta)$ is a weighting value which can automatically change. The weighting will be larger when the joints are close to their limits.

After deciding the performance criterion, we need to know how to decide the value of the $\kappa$. Nevertheless, it is difficult to determine the value of $\kappa$ since the oscillation occurs at large $\kappa$ value and the convergent rate will be quite slow at a small value of $\kappa$. Furthermore, most often a $\kappa$ value may suit with a configuration, but fail for another configuration. Therefore, many researches have been performed about the automatic adjustment of the scalar number $\kappa$. In these researches, the methods of choosing a suitable $\kappa$ are applied case by case. Finally, some researches started to focus on the method of weighted least-square (WLS).

The scheme of a weighted least-square (WLS) for avoiding the joint limits was proposed by Dubey et al. [3]. The issue of $\kappa$ value is settled in this method. The solution of WLS can be expressed as:

$$\dot{\theta} = J_W^+ \dot{x} + (W^{-1/2} J_W^+ - J^+)\dot{x} \tag{2-15}$$

where

$$J_W = JW^{-1/2} \tag{2-16}$$

the $W$ is a diagonal weighting matrix ($n \times n$) shown in Eq. (2-17) and the *i*th element is the diagonal part. The equation of $W$ is defined as Eq. (2-18):

$$W = \begin{bmatrix} w_1 & 0 & \cdots & \cdots & 0 \\ 0 & w_2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & w_n \end{bmatrix} \tag{2-17}$$

$$w_i = 1 + \left| \frac{\partial H(\theta)}{\partial \theta} \right| \tag{2-18}$$

It should be mentioned that $\partial H(\theta) / \partial \theta_i$, is equal to zero if the *i*th joint is the middle of its working range, and moves to infinity at either limit. If a joint moves close to its limit, the value of the weighting factor gets high, and it causes reduction in its motion. In other words, since the weighting factor is closed to infinity, the corresponding joint will virtually stop and joint limit avoidance is guaranteed.

## 2.4 Pattern Generation of Dynamic Walking

We can roughly classify the walking pattern generation into two categories: with and without the ZMP stability criterion. The definition of ZMP given in [70] by Vokobratovic et al. is as follows:

*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.*

Pattern generation controllers with ZMP were widely applied to robot's stability [13], [27, 26, 54], including active and passive mechanisms.

ZMP is a key for biped robot, and it plays different roles in several researches. Some researchers utilized pattern generation with ZMP as an inverse problem. Takanishi et al. [64] solved this with Fourier transformation. The main idea is what kind of motion can get the desired trajectory. Therefore, the Fast Fourier Transformation (FFT) is applied to the ZMP reference, and the ZMP equations can be solved in the frequency domain. By adopting the inverse FFT, the ZMP equation can be derived as COG trajectory in time domain. Nishiwaki et al. [17] used this result and computed the ZMP in discrete time. If the reference data of size N are given, They demonstrated that the ZMP equations can be discretized as a trinomial equation and efficiently solved by an algorithm of O(N).

For other researches, some researchers use the ZMP as the only constraint to judge if the biped robot is falling over in a series of learning period to obtain the proper motion. Therefore, so long as the ZMP is not located out of the supporting area, the robot will not fall.

Above methods were proposed as batch processes. In order to generate the corresponding COG trajectory, they utilized a ZMP reference of a certain period. If they want to generate a proper walking pattern for a long period, they need to calculate the entire trajectory offline or must connect a pieces of the trajectories which are computed from the ZMP reference divided into short parts. For generating online biped walking

control, Nishiwaki et al. [48] updated the motion pattern contributions to maintain balance in a long time. Next section, we will introduce the cart-table robot model with the pattern generator using preview control which was proposed by Shuuji Kajita et al. [27] to solve this problem of pattern generation.

### 2.4.1 Cart-Table Model with ZMP

ZMP is a criterion of walking stability and has been widely used in the control and gait planning of biped robots. ZMP is the point where $Mx=0$ and $My=0$. The $Mx$ and $My$ express moments generated by reaction force. The suffices $x$ and $y$ represent the x-axis (sagittal direction) and y-axis (lateral direction), respectively. Vokobratovic who originally introduced the ZMP concept pointed that the COP (center of pressure) and ZMP are the same point under the gait in dynamical balance. The COP is caused by the ground reaction force. It is a point when weight of robot pressures into the foot. It is a point when the weight of the robot pressures into the foot. The schematic diagram is shown in Figure 2-3.

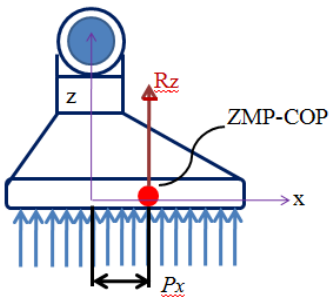



Figure 2-3 Relationship between COP and ZMP.

According to the definition, the ZMP equation can be expressed as:

$$\sum_{i=1}^{N}\left\{m_i(p_{Gi}-ZMP)\times(\ddot{p}_{Gi}+g)+I_i\dot{\omega}_i+\omega_i\times I_i\omega_i\right\}=n_Z \tag{2-19}$$

$$n_Z\times g=0 \tag{2-20}$$

where $m_i$ is the mass of link $i$ ; $N$ is number of all links, and $P_{Gi} \equiv [x_{Gi} \quad y_{Gi} \quad z_{Gi}]^T$ is the center of gravity of the link $i$; $ZMP \equiv [ZMP_x \quad ZMP_y \quad ZMP_z]^T$ is the position of ZMP; $g \equiv [0 \quad 0 \quad -g]^T$ is the acceleration of the gravity; $\omega_i$ is angular velocity of link $i$; and $n \equiv [n_{Zx} \quad n_{Zy} \quad n_{Zz}]^T$ is the moment around ZMP.

To easily implement the model, the dynamic model of the biped robot which uses ZMP can be simplified. A novel method is to derive its 3D Linear Inverted Pendulum Mode (3D-LIPM) [22], as shown in Figure 2-4 Eq. (2-19) can be simplified while the constant COG height $z_c$ is given by:

$$M\ddot{x}=\frac{Mg}{z_c}(x-x_{ZMP}) \tag{2-21}$$

$$M\ddot{y}=\frac{Mg}{z_c}(y-y_{ZMP}) \tag{2-22}$$

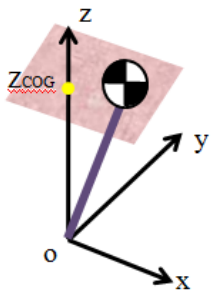



Figure 2-4 3D Linear inverted pendulum model [22].

where $M$ is the mass of the biped robot. It should be mentioned that the simplifying equation ignores the inertia terms. However, the inertia terms will generate another effect with respect to the COM. The simplified equation might lead to some error

especially in the case of high angular velocity motion. To avoid this condition, one researching group used another model, which is basic 3D linear inverted pendulum model with a fly wheel [56]. It modeled the robot as two separate inverted pendulum models which included the fixed leg and the swing leg [55]. For our robot, we use the momentum compensation to reduce the error. According to the 3D linear inverted pendulum model, we rewrite Eqns. (2-21) and (2-22). The ZMP terms are moved to the left hand side and used as the output. The ZMP equations are shown in the Eqns. (2-23) and (2-24).

$$x_{ZMP} = x - \frac{z_c}{g}\ddot{x} \tag{2-23}$$

$$y_{ZMP} = y - \frac{z_c}{g}\ddot{y} \tag{2-24}$$

These can be illustrated using a cart-table model [19] ,as shown in Figure 2-5. Eqns. (2-23) and (2-24) show the relation between the ZMP position and the COG position. The cart is heavy enough mass, and the cart is supported by a table. If the cart stays on the edge of the narrow table, the table is hard to hold the cart for a while. However, we can adopt proper acceleration $\ddot{x}$, and the cart will not fall. By using this model, the moving cart which is like our biped robot can be properly controlled.

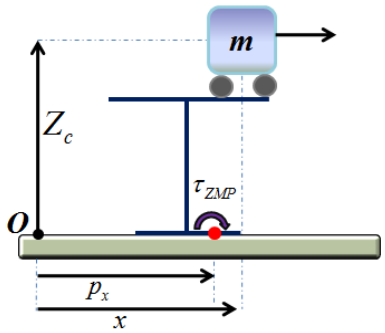



Figure 2-5 A cart-table model [19].

## 2.4.2 Cart-Table Model with Preview Control

Before we introduce the control for biped robot, we must construct the dynamic model of the cart-table. In this model, the ZMP velocity is the system input.

$$\dot{x}_{ZMP} = v \tag{2-25}$$

Then we can combine the results of Eq. (2-23) with Eq. (2-24). The system dynamic equation is given as

$$\frac{d}{dt}\begin{bmatrix} x_{COG} \\ \dot{x}_{COG} \\ x_{ZMP} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{Z_c} & 0 & \frac{-g}{Z_c} \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_{COG} \\ \dot{x}_{COG} \\ x_{ZMP} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \tag{2-26}$$

$$x_{ZMP} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_{COG} \\ \dot{x}_{COG} \\ x_{ZMP} \end{bmatrix}$$

We can use a flow chart to construct the walking pattern generator as a ZMP tracking control system, as shown in Figure 2-6. The input is the reference ZMP trajectory, and the output is the desired COG trajectory. Through the dynamic ZMP equation, the actual output ZMP might not equal to the reference ZMP trajectory. We use a feedback control

to suppress the error so that actual ZMP trajectory will close to the reference.

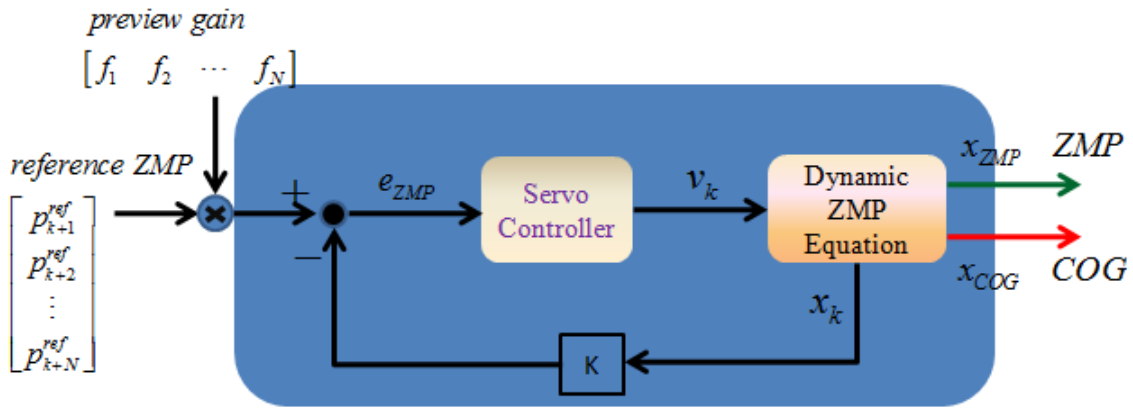



Figure 2-6 Pattern generation as ZMP tracking control.

In Figure 2-6, the servo controller is based on linear quadratic (LQ) optimal control [11, 29, 62, 67], which is a discrete mode of preview control. Thus, we can rewrite Eq. (2-26) in discrete time with sampling period T as

$$x_{k+1} = Ax_k + Bu_k$$
$$p_k = Cu_k \tag{2-27}$$

where

$$x_k \equiv \begin{bmatrix} x_{COG}(k\Delta T) \\ \dot{x}_{COG}(k\Delta T) \\ x_{ZMP}(k\Delta T) \end{bmatrix}, u_k \equiv v(k\Delta T), p_k \equiv x_{ZMP}(k\Delta T) \tag{2-28}$$

$$A = \begin{bmatrix} 1 & T & 0 \\ \frac{g}{Z_c} & 0 & -\frac{g}{Z_c} \\ 0 & 0 & 0 \end{bmatrix}, B \equiv \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, C \equiv [0 \quad 0 \quad 1] \tag{2-29}$$

Since we use the optimal control system, a performance index is defined as

$$J = \sum_{i}^{\infty} \left\{ Q(p_i^{ref} - p_i)^2 + Ru_i^2 \right\} \tag{2-30}$$

where $Q$ is a weighting value of symmetric positive semi-definite matrix, $R$ is a symmetric positive definite matrix, and $p_i^{ref}$ and $p_i$ are reference ZMP point and actual ZMP point in discrete time, respectively. In order to generate a much stable walking pattern generation, we choose the $Q$ matrix larger than $R$ for tracking performance. In other words, we care about the ZMP tracking error rather than the effect of input. The optimal controller can be obtained as

$$u_k = -Kx_k + \begin{bmatrix} f_1 & f_2 & \cdots & f_N \end{bmatrix} \begin{bmatrix} p_{k+1}^{ref} \\ p_{k+2}^{ref} \\ \vdots \\ p_{k+N}^{ref} \end{bmatrix} \tag{2-31}$$

where $K$ and $f_i$ are expressed as:

$$K \equiv (R + B^T PB)^{-1} B^T PA$$

$$f_i \equiv K \equiv (R + B^T PB)^{-1} B^T (A - BK)^{T \times (i-1)} C^T Q$$

Using the Riccati equation to find the solution $P$ as

$$P = A^T PA + C^T QC - A^T PB(R + B^T PB)^{-1} B^T PA$$

By using the preview control, we can find the desired COG trajectory, and the COG will not be constant. The changing height of COG will be described in next chapter. Furthermore, for the robustness of the preview controller, we can give a desired ZMP trajectory to make the robot walk more smoothly. The details will be described in

Chapter 3.

## 2.5 Summary

After setting the IK-solver with RWLS and using preview control to generate the pattern, the simulation structure with ADAMS and MATLAB can perform dynamic locomotion in an offline way , as shown in Figure 2-7.

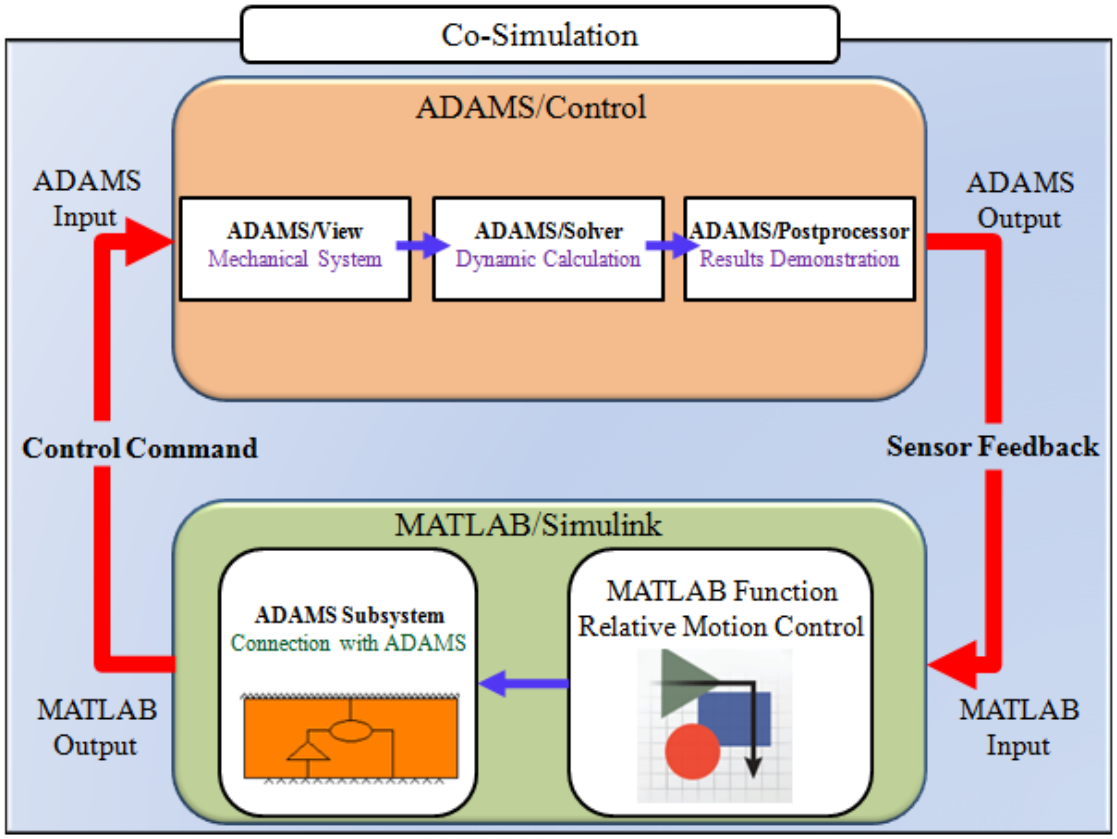



Figure 2-7 Co-simulation overall structure [79]

In simulation, the biped robot can walk at a speed of 0.8 second per step (limited by the current configuration). To maintain ZMP tracking performance, the preview control needs future information about the next two steps (at least), but even with this provision, the result is still valuable. In fact, the variation of COG height during

locomotion is based on this preview control structure. Without using the simplified ZMP model, the COG height will be varied with time

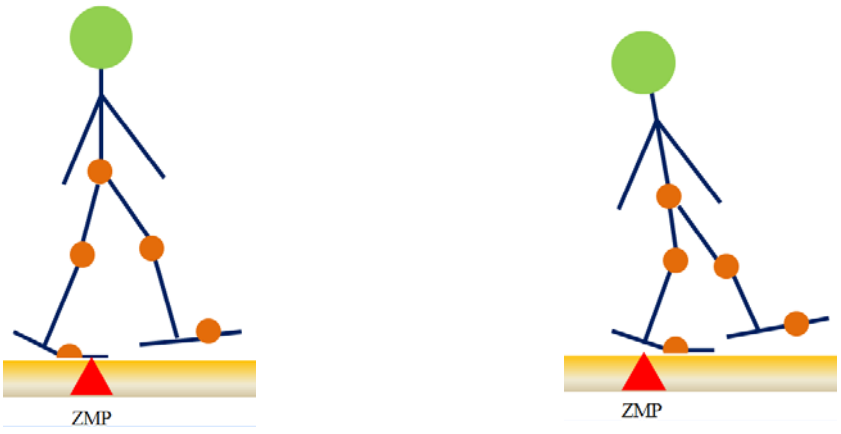
.

# Chapter 3 ZMP Analysis and Varying COG

## 3.1 Introduction

This chapter introduces the influences of ZMP location, COG distribution, and the conservation of momentum on a biped robot's locomotion. We shall discuss each in turn.

To achieve our target of a human-like walking posture, we need to change the ZMP location from the one used by conventional flat-footed robots. The ZMP method, introduced in Section 3.2, shows us that the ZMP trajectory must always locate inside the support polygon. Note also that arbitrary assignment of the speed of change of ZMP is impossible because of the difficulty of limiting ZMP in the narrow areas of the toes and heels. We must therefore choose a ZMP trajectory that will not lead to the robot's falling. Figure 3-1 compares the effects of correct and incorrect ZMP positioning.




(a) (b)

Figure 3-1 (a) A robot with effective ZMP (b) A robot with ineffective ZMP.

Shimmyo and Ohnishi describe a nested preview control that uses a virtual plane for biped pattern generation, including COG up-down motion [63]. Other researchers, noting that the ZMP trajectory directly affects the walking motion of biped robots, have made different proposals [60, 61, 78, 81], most attempting to deliver smoother ZMP and COG trajectories.

The variation of COG is the second influence discussed in this chapter. Some researchers [58] have discovered that COG height varies with time during the human walking cycle, which is as shown in Figure 3-2. Earlier biped robot control has usually adopted a constant COG height to avoid the singular condition. This results in a squatting posture for part of the cycle—a posture not adopted by humans and one which consumes a great deal of power when used by robots. The second part of this chapter covers the generation of a suitable up-down motion of COG.

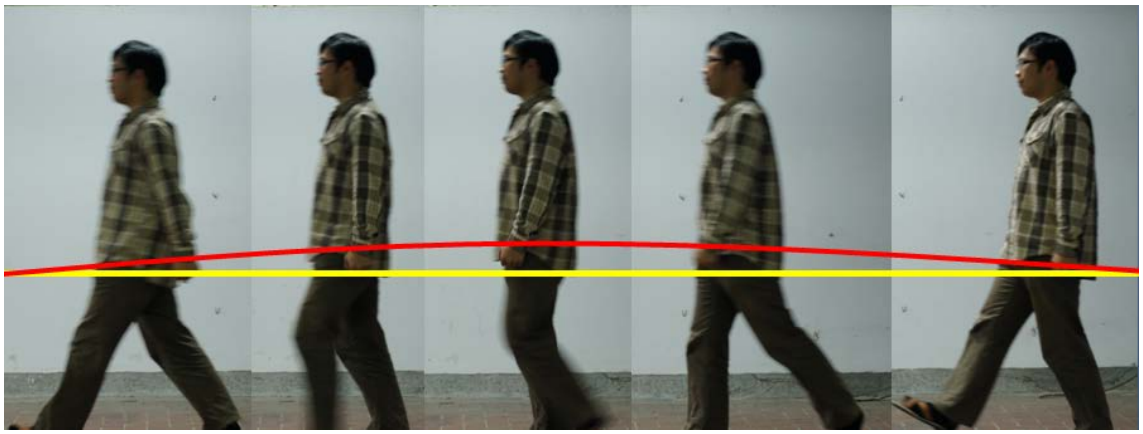


Figure 3-2 The shift of COG in human walking. The red line is actual, the yellow line is reference.

Kobayashi et al. proposed the automatic navigation of COG in the sagittal and

lateral planes [30]. Ahn et al. developed a walking algorithm for biped robots in the motion-embedded COG Jacobian framework, and the COG trajectory could wave in the vertical plane [35]. More COG trajectory of researches can refer to [2, 39, 58].

The third influence to be added to our robot's solver is momentum compensation, or more specifically, Resolved Momentum Control. Robots become unstable when executing slightly exaggerated motions that are easily achieved by human beings. The momentum effect is shown in Figure 3-3. One of the differences between biped robots and human beings in performing these tasks is the ability to maintain stability in terms of response to the expected motions—an ability not possessed by biped robots because of deficient dynamic constraint and modeling error. Unexpected z-rotational slip is likely if the constraint of the angular momentum of the z-axis is not obtained by solving inverse kinematics.

Figure 3-3 The movement comparison between human and biped robot [86].

Many researchers have pointed out the importance of momentum for robots. Umetani and Yoshida [69] proposed a resolved motion-rate control combined with

momentum conservation. From [53] described a manipulator mounted on a non-initial base (e.g., ships on the sea), and showed how non-inertial effects can be taken into account in planning motion. There were some balancing cases, with momentum as the issue. For example, Kagami et al. [16] proposed a balance control that manipulated the COG with nonlinear programming optimization.

By combining the varying COG' height, desired ZMP trajectory, and Resolved Momentum Control, we can produce a robot walking motion that is more human than earlier, flat-footed models, and we have conducted simulations to demonstrate the effect.

## 3.2 Adjusting ZMP Trajectory

To prevent a biped robot falling over, the ZMP must be inside the support polygon or support area. We must therefore find out where the support area. It is crucial to find out where is the support area. Most researchers located ZMP at the center of the flat foot during single support, since its support area is much larger. The robot rarely falls down. During the double-support phase, however, the ZMP moves very fast, as shown in Figure 3-4. Once the ZMP trajectory is assigned, the related COG can be derived by solving the ZMP equation. The high speed of the ZMP trajectory in the double-support phase means that the acceleration of COG may need to be large enough (and well enough controlled in the upper body) to track it. This leads to increased possibility of

instability.

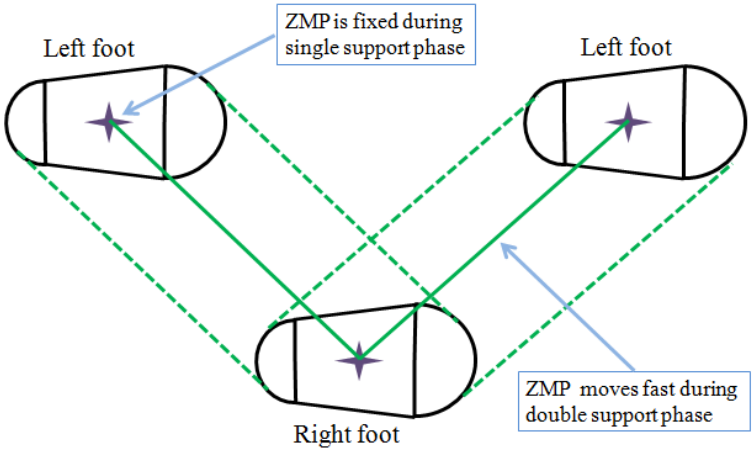



Figure 3-4 Common ideal ZMP trajectory.

### 3.2.1 ZMP Trajectory for Single and Double Support

Before we decide the ZMP trajectory for a biped robot, we can consider the human ZMP trajectory, as shown in Figure 3-5. Note that the ZMP is not fixed at the midfoot but change its position with time. We can divide the walking cycle into three phases—heel-contact motion, midfoot contact, and toe-off motion, as shown in Figure 3-6.

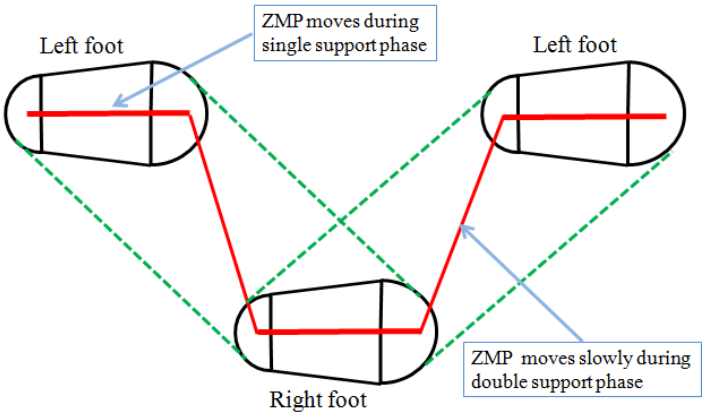



Figure 3-5 Human ZMP trajectory during walking.

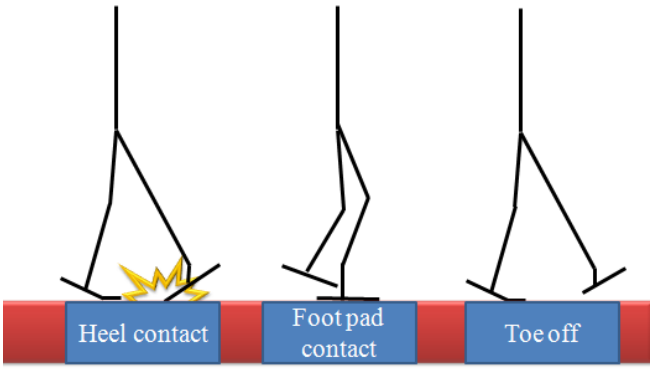


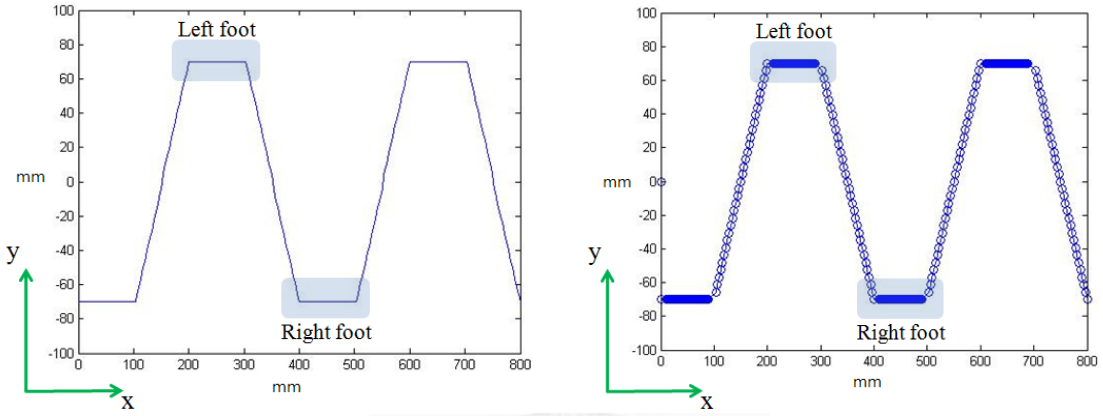

Figure 3-6 Three phases of human walking.

We need to know the locations of the support areas in the three phases of human foot motion shown in Figure 3-6, during both single and double support. During single support, the full body weight is on one foot, but is shared by both feet during double support. The area used during single support is therefore smaller than that used during double support, and if it becomes too small to control, instability is likely. Chapter 4 provides a more detailed description of the three-phase support distribution model.

### 3.2.2 ZMP Trajectory for Walking Period

The reference ZMP trajectory needs to be decided. We can design one based on the human ZMP trajectory with its ZMP motion from heel to toe, as shown in Figure 3-7. Once adopted, the trajectory can be mapped against time in discrete steps through the cycle. However, simple linear interpolation could result in points of low stability that can be overcome only by introducing high acceleration values, resulting in a jerky motion. A smoother trajectory can be computed using a third-order polynomial function. Here we must particularly note that straight-line walking is in the x-direction only, so

we need interpolate the ZMP trajectory only along the x-direction during single support. During double support, however, we need to consider the y-direction as well. Integration gives the ZMP trajectories shown in Eq. (3-1), Eq. (3-2), and Figure 3-8.




(a) (b)

Figure 3-7 (a) ZMP trajectory in continuous time, (b) ZMP trajectory in discrete time.

ZMP in x-direction

$0 \le t \le t_1$ during single support , $t_1 \le t \le t_2$ during double support

$$x_{ZMP} = at^3 + bt^2 + ct + d$$

$$\text{I.C. } x_{ZMP} = x_{heel}, \quad v = 0 \text{ at } t = 0 \qquad (3\text{-}1)$$
$$x_{ZMP} = x_{toe}, \quad v = 0 \text{ at } t = t_1 \quad , 0 \le t \le t_1$$

$$x_{ZMP} = x_{toe}, \quad v = 0 \text{ at } t = t_1$$
$$x_{ZMP} = x_{next\_heel}, \quad v = 0 \text{ at } t = t_2 \quad , t_1 \le t \le t_2$$

ZMP in y-direction

$t_1 \le t \le t_2$ Double support

$$y_{ZMP} = et^3 + ft^2 + gt + h \qquad (3\text{-}2)$$

$$\text{I.C. } y_{ZMP} = y_{toe}, \quad v = 0 \text{ at } t = t_1$$
$$y_{ZMP} = y_{next\_heel}, \quad v = 0 \text{ at } t = t_2 \quad , t_1 \le t \le t_2$$

where *a* to *h* are coefficients, and $x_{heel}$, $x_{toe}$, and $y_{toe}$ are the positions of heels and toes on their respective axes. The suffix *next_heel* means the position of the next foot to land,

and $v$ expresses the velocity.

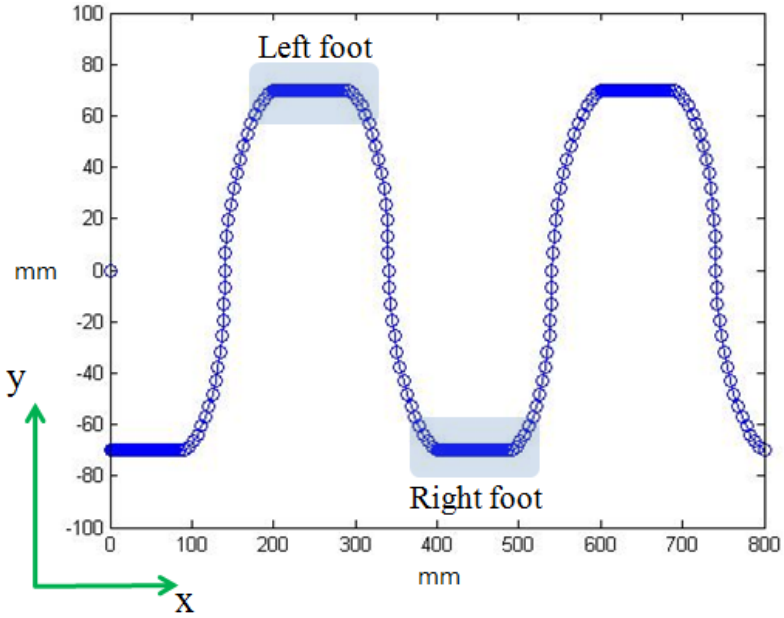



Figure 3-8 ZMP trajectory with smoothly interpolating points.

### 3.2.3 Simulations and Results

Figure 3-9 shows the robot motion achieved by taking the changed reference ZMP into the IK-solver. The original ZMP retained the COG at the center of the foot throughout the cycle, resulting in rapid upper body motion on transition to double support.

The new reference ZMP allows the robot's COG to move during the transition, so upper body movement is smooth and slow, and stability will be shown to have been improved, both in the simulation and in the practical experiment. Figure 3-9 also shows that velocity and acceleration have finite values throughout.

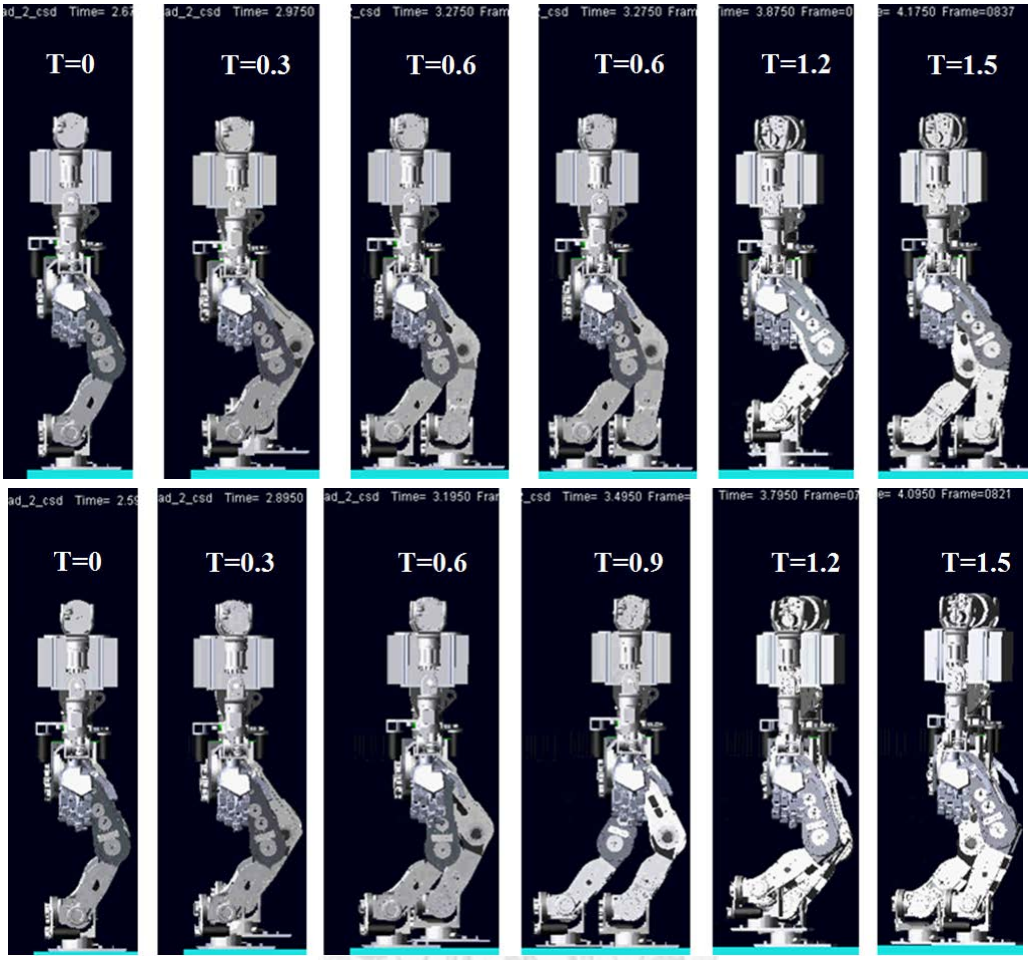



Figure 3-9 Walking motion of the humanoid robot using the common (upper) and smooth (lower) ZMP trajectories.

# 3.3 Varying Height of COG

Human walking motion involves the vertical movement of COG throughout the cycle. This capability will be useful for biped robots, and must be taken into account, as human-like motion will give robots an expanded repertoire of human-like behaviors.

## 3.3.1 Dynamic ZMP Equation with Varying COG

Section 2.4.1, proposed the dynamic ZMP equation, but Eq. (2-23) and (2-24) are simplified to apply to the inverted pendulum model. Eq. (3-3) and Eq. (3-4). show the governing equations of the inverted pendulum model used in most ZMP-COG pattern

generation algorithms.

$$X_Z = X_C - \ddot{X}_C \left( \frac{Z_C - Z_Z}{g + \ddot{Z}_C} \right) \tag{3-3}$$

$$Y_Z = Y_C - \ddot{Y}_C \left( \frac{Z_C - Z_Z}{g + \ddot{Z}_C} \right) \tag{3-4}$$

where *X*, *Y*, and *Z* denote the position in x, y, and z directions. The subscript *C* denotes the COG, and the subscript *Z* denotes the ZMP. Using the simplified equation to model our robot will lead to modeling error because of the non-constant COG height. However, it is difficult to define state $\ddot{Z}$ if the non-simplified Eq. (3-3) and Eq. (3-4), are used instead. For this reason, we use preview control to generate the reference COG trajectory It turns out the COG changes slowly, the modeling error decreases. The process is sketched in Figure 3-10. By using this method, we can assign the COG trajectory arbitrarily. The next section introduces the human-like COG trajectory.

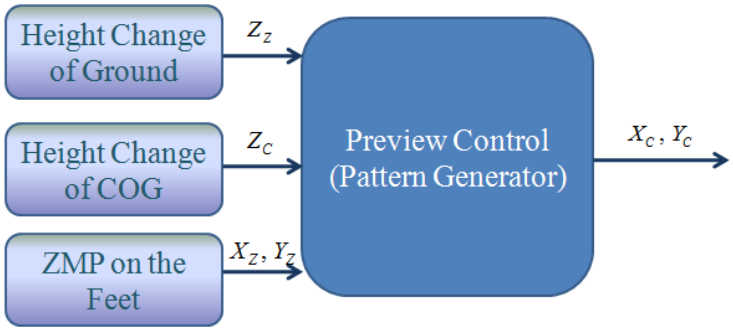



Figure 3-10 Input-output relationship of the controller.

## 3.3.2 Human-Like Trajectory of COG

The robot's COG can be computed from the individual COGs of all its components. As most of these are not regular solids, we can use SolidWorks software to find the

robot's COG, as shown in Figure 3-11. Once we have established the COG height, we can start to assign the COG trajectory in the z direction.

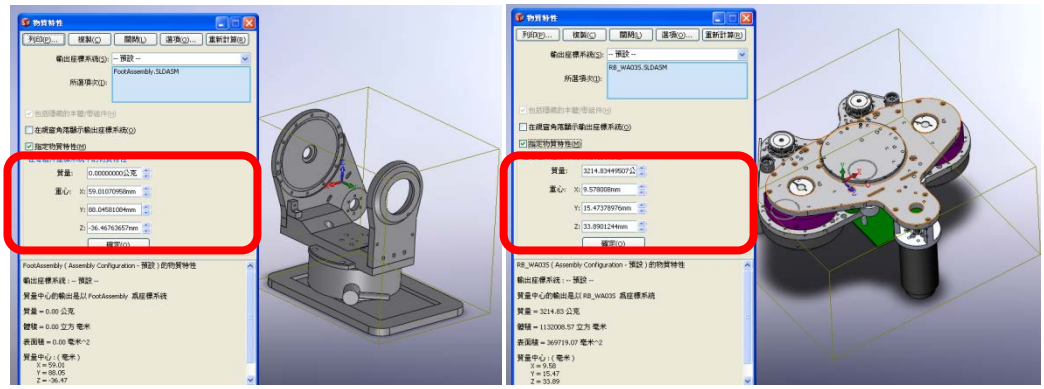


Figure 3-11 Using SolidWorks to calculate the COG of all components.

About the motion of the COG, the advantages of waving COG are:

1. The knee joints of humanoid robot do not always bend during walking period.
2. We do not estimate the COG height to avoid the singular condition.

We need to analyze the human COG trajectory so that we can adopt it for our robot, avoiding the bent-knee walking of the robot and obviating the necessity of estimating COG height to avoid the singular condition. We used OpenSim software to observe human COG trajectories, as shown in Figure 3-12.

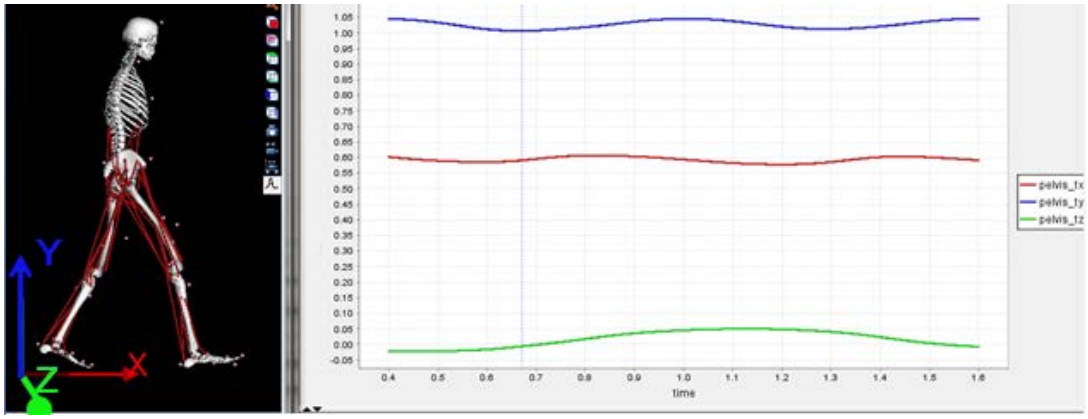



Figure 3-12 The human COG trajectory, visualized with OpenSim.

The blue line, which is similar to a sine or cosine wave, denotes the COG trajectory in the z-direction. The trough appears during double support, and the curve peaks at the center of the single-support period. Both these characteristics are used as design parameters.

Analysis using the quartic order polynomial function to yield the up-down motion as shown in Figure 3-13, and Eq. (3-5), and Eq. (3-6).

Up Motion

$$COG_Z = at^4 + bt^3 + ct^2 + dt + e$$

$$\text{I.C. } COG_Z = COG_{low},\ v = 0 \quad \text{at } t = 0 \tag{3-5}$$

$$COG_Z = COG_{high},\ v = 0 \quad \text{at } t = t_1$$

$$COG_Z = \frac{COG_{low} + COG_{high}}{2} \quad \text{at } t = \frac{t_1 - 0}{2} \quad \text{up motion} \quad 0 \le t \le t_1$$

Down Motion

$$COG_Z = at^4 + bt^3 + ct^2 + dt + e$$

$$\text{I.C. } COG_Z = COG_{high},\ v = 0 \quad \text{at } t = t_1 \tag{3-6}$$

$$COG_Z = COG_{low},\ v = 0 \quad \text{at } t = t_2$$

$$COG_Z = \frac{COG_{high} + COG_{low}}{2} \quad \text{at } t = \frac{t_1 + t_2}{2} \quad \text{down motion} \quad t_1 \le t \le t_2$$

$COG_Z$ is the COG trajectory in the z-direction, and the suffixes $high$ and $low$ denote the highest and lowest points of the trajectory. The walking period of one step is from $t = 0$ to $t = t_2$, and the highest point of the COG trajectory appears at $t = t_1$.

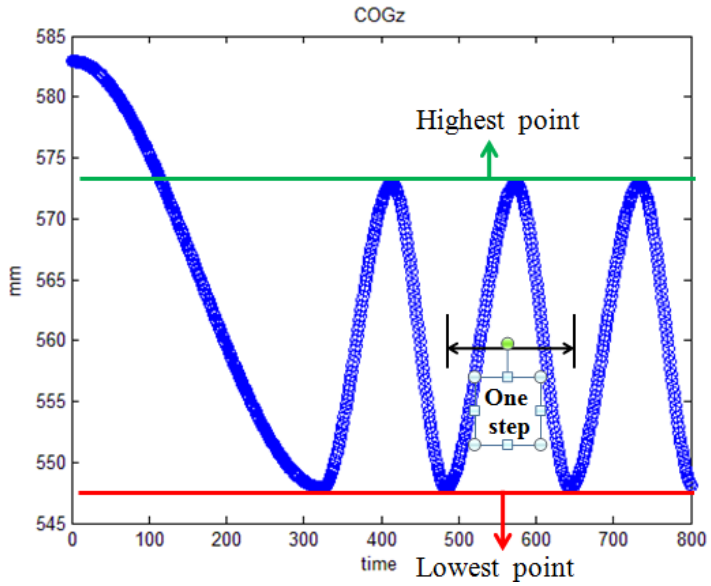



Figure 3-13 The trajectory of the reference COG in the z-direction.

To keep the rate of change of COG small, accelerations and jerks need to be continuous and finite, and this provision is satisfied by using the quartic orders polynomial function. This reference trajectory gives the humanoid robot the same locomotion as a human's, as the simulation illustrated in Figure 3-14 shows.

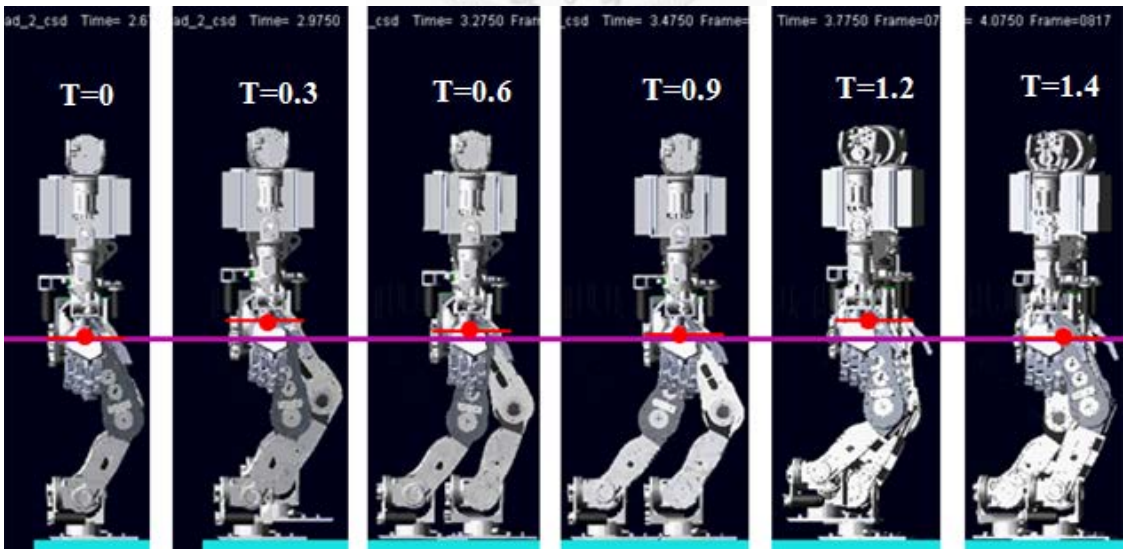



Figure 3-14 Simulation of changing COG height for a humanoid robot.

Note that the knee straightens for part of the step cycle, just like a human knee does.

Vertical COG movement makes the robot's locomotion much more human-like than the flat-footed motion employed by earlier humanoid robots.

One of the reasons humans have adopted this COG trajectory is that it is more efficient—it conserves energy. To prove that vertical COG movement saves robot power consumption, we have measured knee-joint angles for motion with and without COG variation. The results are shown in Figure 3-15 (knee angles) and Figure 3-16 (torque). Note that the ADAMS software imposes some limitations when computing the simulation, so some uncharacteristic peaks appear in the results.

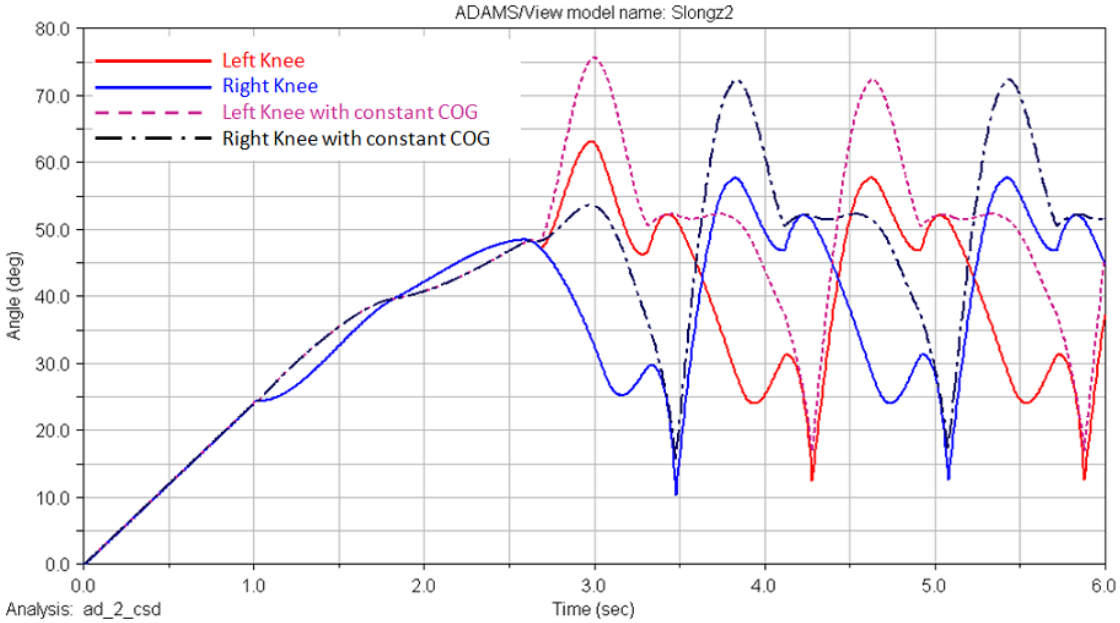



Figure 3-15 The angles of the knee joints with and without variant COG.

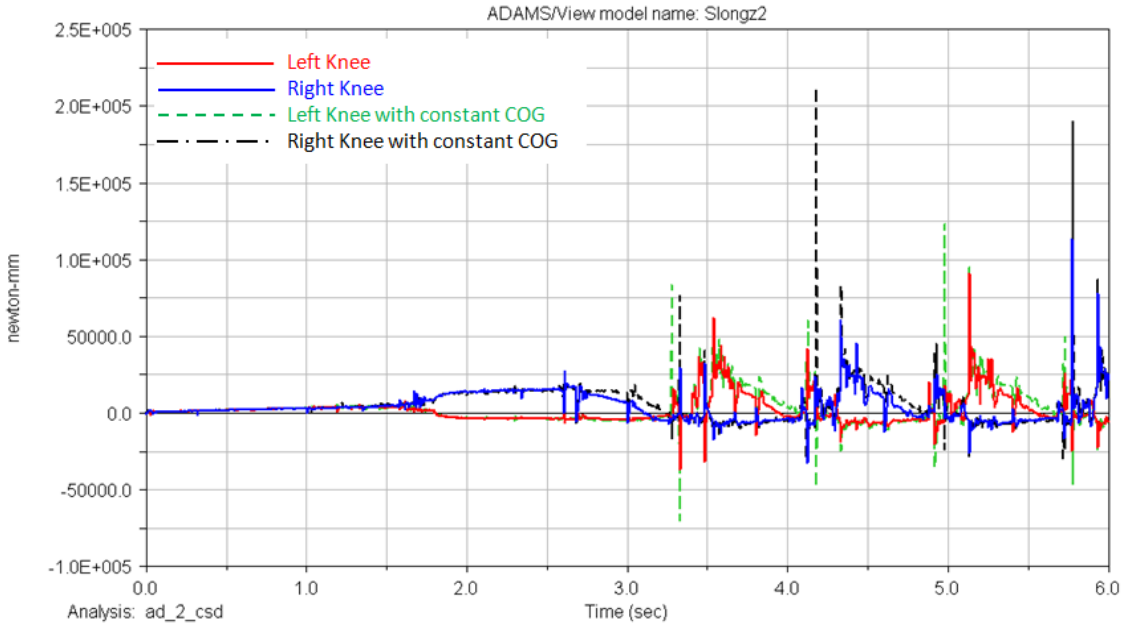



Figure 3-16 The power consumptions of the knee joints with and without variant COG.

# 3.4 Momentum Compensation

Our biped robot can walk in the way we designed it to do, once preview control has constructed the pattern generation. There are still some limitations, however, as humans, unlike our robot at this stage, can maintain stability in terms of response to the expected motion. The robot becomes unstable when walking in a straight line, as it slips on its standing leg. This is because of modeling error and deficient dynamic constraints. The simplified equation used to construct the dynamic model takes no account of the fact that the moment of inertia on spinning linkages, giving rise to tracking errors for both COG and ZMP. This problem can be overcome by generating waist and arm rotation to compensate for the momentum generated by the moving linkages. In this section, we combine the momentum compensation term into the IK-solver, and carry out some simulations to prove the value of doing so. The main application is described in Section

4.6.

### 3.4.1 Momentum Derivation

Before we introduce the Resolved Momentum Control into dynamic motion, we will show the scheme of momentum derivation, as shown in Figure 3-17. It must be derived the affected momenta of inertia on the fixed leg. The ideal physical quantity that must be derived is the affected momenta of inertia on the fixed leg of all rotational axes, in which multiple joint angular velocities is equal to the angular momentum on the fixed foot generated by the axes. In Figure 3-17, we can know the sequence of deriving the equation, and the choice of the coordinate is critical.

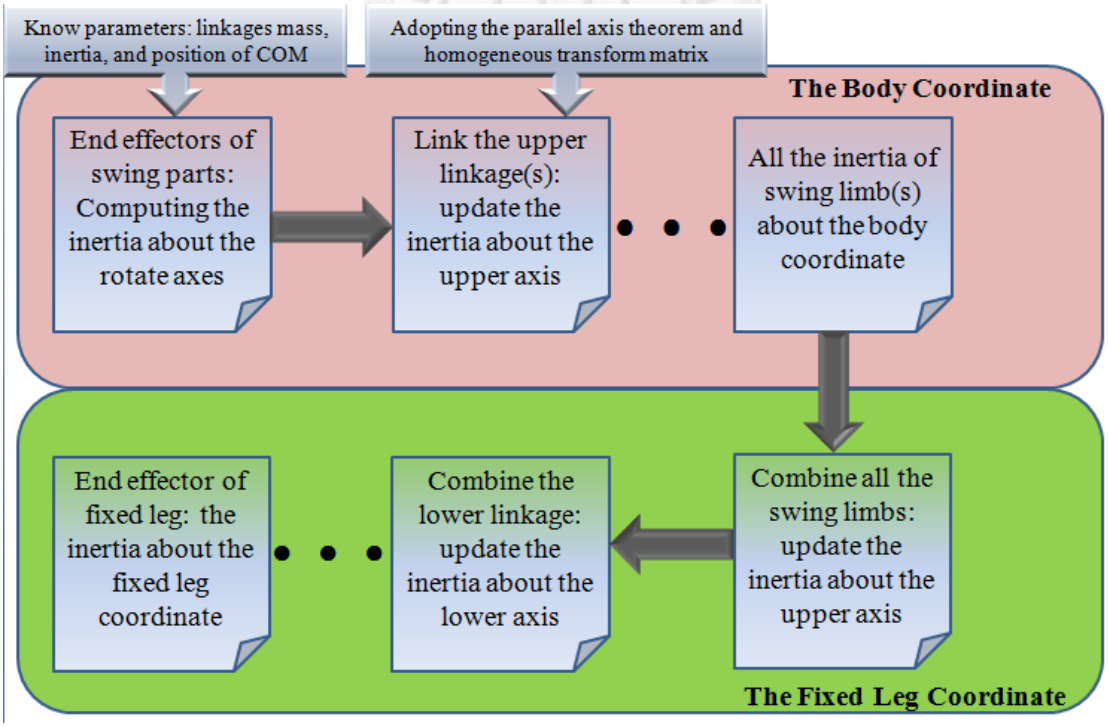



Figure 3-17 Scheme for derivation of inertia.

The main coordinate we adopt is the coordinate $\Sigma_B$, and the $v_B$ and $w_B$ are the translation velocity and the angular velocity which represented by Kajita et al. [20]. In

this coordinate, the origin is a free-flying base link (pelvis). The roughly diagram is shown in Figure 3-18. Taking the linear momentum and the angular momentum into account, the represented equation is given in Eq. (3-7).

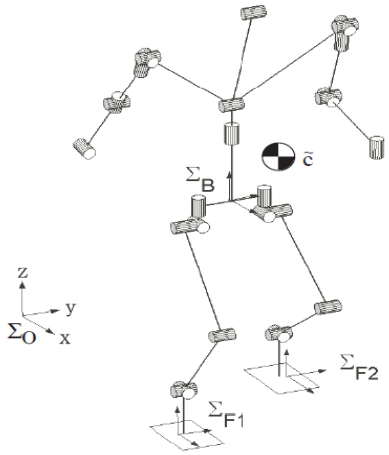



Figure 3-18 Model of humanoid robot [20].

$$\begin{bmatrix} P \\ L \end{bmatrix} = \begin{bmatrix} \tilde{m}E & \tilde{m}\hat{r}_{B\to\tilde{c}} & M_{\dot{\theta}} \\ 0 & \tilde{I} & H_{\dot{\theta}} \end{bmatrix} \begin{bmatrix} v_B \\ \omega_B \\ \dot{\theta} \end{bmatrix} \tag{3-7}$$

where $\tilde{m}$ indicates the total mass of the robot, $E$ is the $3\times3$ identity matrix which is distinct from the inertial matrix $\tilde{I}$, the $\tilde{I}$ is the $3\times3$ inertial matrix with respect to the COG and $\tilde{r}_{B\to\tilde{c}}$ is $3\times1$ vector from the base to the center of gravity (COG) of robot. The final terms $M_{\dot{\theta}}$ and $H_{\dot{\theta}}$ are the $3\times n$ inertial matrices that indicate how the joint speed affects the linear and angular momentums, respectively. The ^ is an operator changing a $3\times1$ vector into a skew symmetric matrix $(3\times3)$. $\dot{\theta}$ is a $n\times1$ column vector which are velocities of all joints as its elements and $n$ is the total number of joints. This equation is derived with respect to coordinate $\Sigma_B$. In addition,

all the positions, velocities, and angular velocities are represented in Cartesian space $\Sigma_0$ which is located on the ground.

## 3.4.2 Momentum of the Fixed Leg

In practic we need to calculate the momentum of the robot about the fixed leg. Therefore, we need to shift the coordinate $\Sigma_B$ to the fixed leg coordinate $\Sigma_{FL}$ and $\Sigma_{FR}$ (depending on left or right foot is landing), whose origin locates on the fixed leg. We rewrite $M_{\dot{\theta}}$ and $H_{\dot{\theta}}$ to Eq. (3-8) by taking momentum on the fixed leg coordinate instead of the $\Sigma_B$.

$$\begin{aligned} M_{global} &= \begin{bmatrix} M_{fixed} & M_{swing} \end{bmatrix} \\ H_{global} &= \begin{bmatrix} H_{fixed} & H_{swing} \end{bmatrix} \end{aligned} \tag{3-8}$$

where the suffix *fixed* represents the fixed leg, and the *swing* includes the swing leg and other free parts. Here $M_{global}$ and $H_{global}$ are represented in fixed leg coordinate. Then we separate the contribution of momentum from COG to masses of linkages of the robot with respect to the corresponding rotational axes to rewrite Eq. (3-7) as

$$\begin{bmatrix} P \\ L \end{bmatrix} = \begin{bmatrix} M_{global} \\ H_{global} \end{bmatrix} \dot{\theta} \tag{3-9}$$

$$\begin{aligned} M_{global} &\equiv \begin{bmatrix} m_1 & m_2 & \cdots & m_n \end{bmatrix} \\ H_{global} &\equiv \begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix} \end{aligned} \tag{3-10}$$

where $m_j$ and $h_j$ are 3×1 vectors, which are the columns of the inertia matrices corresponding to the joint. The $m_j$ and $h_j$ are represented below:

$$\begin{cases} m_j \equiv P_j / \theta_j = a_j \times \left(\tilde{c}_j - r_j\right)\tilde{m}_j \\ h_j \equiv L_j / \theta_j = \tilde{c}_j \times m_j + \tilde{I}_j a_j \end{cases} \quad \text{for joints of swing limbs} \qquad (3\text{-}11)$$

$$\begin{cases} \hat{m}_j \equiv P_j / \theta_j = -a_j \times \left(\hat{c}_j - r_j\right)\hat{m}_j \\ h_j \equiv L_j / \theta_j = \hat{c}_j \times \hat{m}_j + \hat{I}_j(-a_j) \end{cases} \quad \text{for joints of fixed leg} \qquad (3\text{-}12)$$

where $r_j$ and $a_j$ are the position vector and the unit vector of the rotational axis of the joint $j$, respectively. For swing limbs' joints, $\tilde{m}$, $\tilde{c}$, and $\tilde{I}$ are the mass, location of COG, and inertia tensor of affected linkages driven by the joint $j$, respectively. The affected linkages mean the moving linkages which are moved by joint $j$. On the other hand, for fixed leg joints, $\hat{m}$, $\hat{c}$, and $\hat{I}$ indicate the same variable as swing limbs. However, the affected linkages are different from swing limbs. When joint $j$ rotates, the affected linkages are the parts above this joins. Hence, from Eqns. (3-11) and (3-12), the swing limbs' moving direction are just the opposite to fixed legs'. For any individual linkage, we can measure the inertia tensor about its COG, its weight, and the location of COG by utilizing the CAD software.

In order to derive $\tilde{m}_j$, $\tilde{c}_j$, and $\tilde{I}_j$ of the upper linkages that affect plural linkages whose $\tilde{m}_{j-1}$, $\tilde{c}_{j-1}$, and $\tilde{I}_{j-1}$ of lower linkages are known, the method we utilized is recursive calculation.

$$\tilde{m}_{j-1} = \tilde{m}_j + m'_{j-1} \qquad (3\text{-}13)$$

$$\tilde{c}_{j-1} = \left(\tilde{m}_j \tilde{c}_j + m'_{j-1} c_{j-1}\right) / \left(\tilde{m}_j + m'_{j-1}\right) \qquad (3\text{-}14)$$

$$\tilde{I}_{j-1} = \tilde{I}_j + m'_j D\left(\tilde{c}_j - \tilde{c}_{j-1}\right) + R_{j-1} I_{j-1} R_{j-1}^T + m'_{j-1} D\left(c_{j-1} - \tilde{c}_{j-1}\right) \qquad (3\text{-}15)$$

$$D(r) \equiv \left(r^T r\right) I - r r^T$$

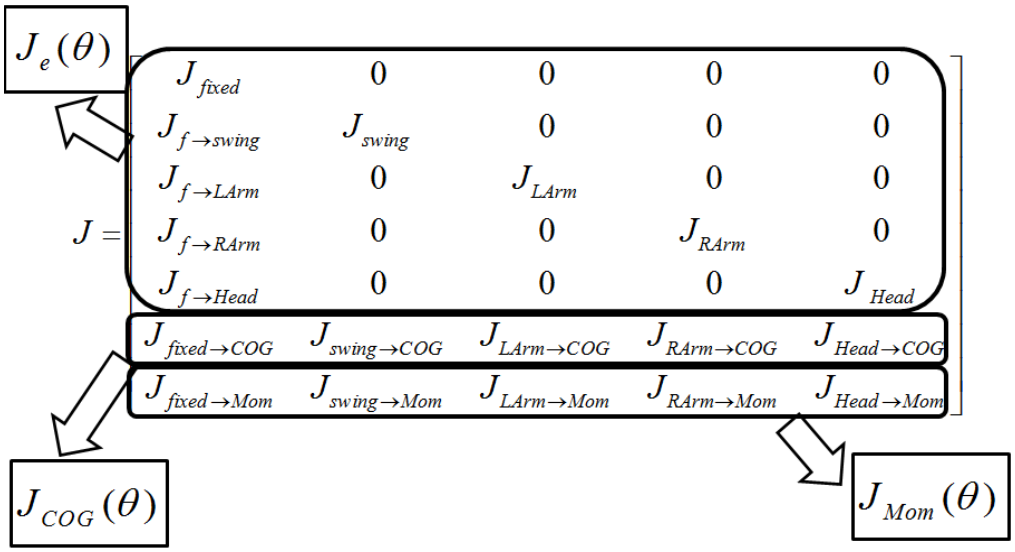
where $R_{j-1}$ is the $3\times3$ rotation matrix of the $j-1$ th joint. The order of recursive calculation is computing from the tips of swing limbs to the fixed leg as the ground rod. As a result, we can derive the momentum about the fixed leg as described in Eq. (3-9).

### 3.4.3 Adding Momentum Jacobian to Inverse Kinematics

To achieve the desired locomotion of robot with momentum compensation, we have set some constraints to our IK Jacobian matrix. In Section 3.4.2, we have derived the momentum on the fixed leg. In IK solver, we change the Jacobian matrix to satisfy corresponding joints to generate the desired motion. Here we utilize the same work which is like the end-effector or COG Jacobians. The general equation we adopt is $\dot{x} = J\dot{\theta}$, and the momentum Jacobian should match this equation, too. Since the momentum equation is $Momenta = J_{mom}\dot{\theta}$, the $J_{mom}$ is equivalent to $\begin{bmatrix} M_{global} \\ H_{global} \end{bmatrix}$ in Eq. (3-9). According to the end-effectors and COG Jacobian matrices in Section 2.3.1, we can rewrite the inverse kinematics equation (Eq. (2-5)) to a new equation with the constraint of angular momentum, as shown in Eq. (3-16). By using this description, we can find the new joint trajectories after computing with RWLS. With the desired

momentum we set, the motion of the robot will offset the momentum of z-direction about the fixed leg. Since we want the momentum in z-direction to be equal to zero when walking straight, the momentum generated by the robot limbs should be compensated. After adding the momentum term to the Jacobian matrix, the errors which are generated by modeling error and errors of estimated inertia can be decreased. The next section will show some simulation about practical applications.

$$\begin{bmatrix} \dot{x}_{fixed} \\ \dot{x}_{swing} \\ \dot{x}_{LArm} \\ \dot{x}_{RArm} \\ \dot{x}_{Head} \\ \dot{x}_{COG} \\ L \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_{fixed} \\ \dot{\theta}_{swing} \\ \dot{\theta}_{LArm} \\ \dot{\theta}_{RArm} \\ \dot{\theta}_{Head} \end{bmatrix} \tag{3-16}$$




## 3.4.4 Simulation with Momentum Compensation

After adding the momentum terms to the IK-solver, we can enable the robot to make a movement to compensate for any unnecessary momentum that it generates. We

choose to use the waist joint for this purpose, as shown in Figure 3-19. This motion allows the robot to take some dozens of steps in a straight line without deviating. Note also that compensatory waist motion allows all joints to operate within their limits of motion, and confers an even more human-like motion to the robot. The result of the simulation is shown in Figure 3-20. Additionally, the trajectories of the waist also avoid the joint limits, as show in Figure 3-19.

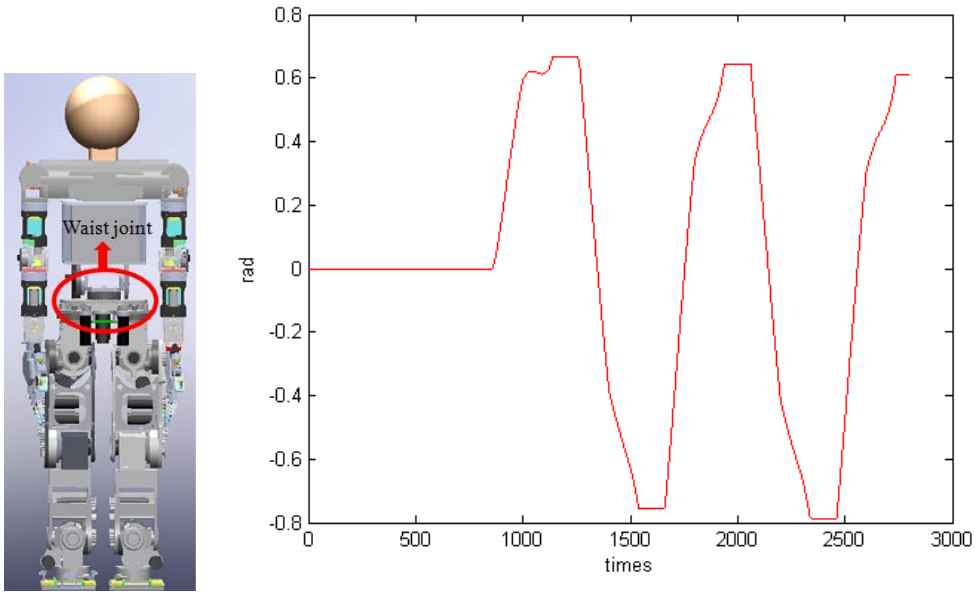



Figure 3-19 The left picture is waist joint of the robot, and the right figure is the trajectory of waist joint of momentum compensation.

In Figure 3-19, the angle of the waist is constant for part of the cycle, as we have allowed the robot to pause the momentum function during the double-support phase. In this phase, both legs are standing legs, so no obvious momenta are generated The momentum compensation is able to make the robot walk more straight and more like human being.

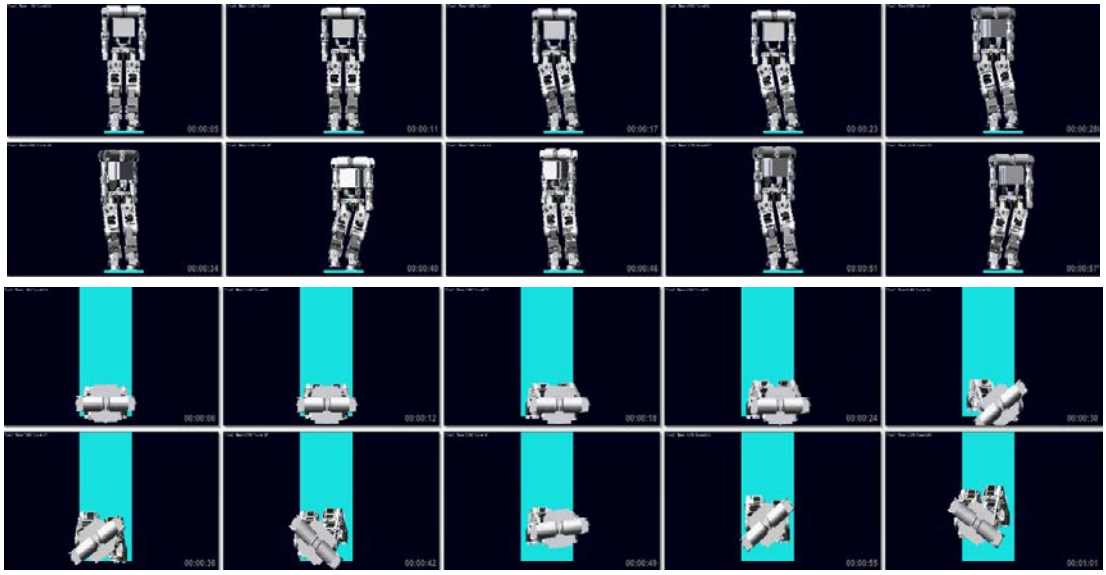


Figure 3-20 Simulations of walking straight with momentum compensation. The sampling rate is 0.06sec/picture.

## 3.5 Summary

This chapter has introduced three methods by which a robot's walking motion can approximate more closely to that of a human being. They are adjusting the ZMP trajectory, varying the COG height, and compensating for momentum compensation.

The conventional ZMP trajectory allows discontinuities in velocity and acceleration during parts of the cycle, and is also unable to incorporate toe-off and heel-contact motions. Our planning has allowed us to modify the trajectory, removing the discontinuities and providing the more natural walking motion that we discuss in the following chapter.

We have designed a new COG trajectory that reproduces the natural vertical COG movement seen in humans, thus allowing straightened knees during part of the robot's stride and a consequent reduction in power consumption.

The third influence on natural walking is momentum compensation. We have implemented this feature, which reduces the redundant momenta caused by walking movements, by enabling the upper body to rotate as necessary during the stride cycle.

# Chapter 4 Dynamic Locomotion with Natural Walk

## 4.1 Introduction

Humanoid robots are now able to perform a variety of stable walking tasks, although the walking motions of most are different from those of humans, which are illustrated in Figure 4-1 . Their flat feet allow them to walk at different speeds [6, 7, 25, 57], but their stride lengths are restricted by a lack of articulated toes and heels.

Figure 4-1 Motion of human walking. [83]

Recent studies have aimed at giving humanoid robots a “natural” or “human-like” walk. To do this, it is necessary to discover how robots can use toes and heels successfully, how to enable the knees to straighten as necessary, and how to lengthen stride as required without having to lengthen the legs. Several methods have been proposed to solve these problems.

Konno et al. adopted passive toes fitted with torsion springs to achieve a natural

walk [31]. HRP2, the humanoid robot proposed by Sellaouti et al., [61] walks with passive toe joints, and its ZMP trajectory successfully lengthened the step length, thus directly increasing the walking speed. Humanoid robots with toe joint designed for natural walking are common [71]. Sato et al. [60] proposed trajectory planning involving active toe and heel joints. Handharu et al. [9] used known hip and foot trajectories to look for a time-based solution using inverse kinematics. Harada et al. [10] used motion capture to generate human joint trajectories for humanoid robots. In those researches, they divided the walking cycle into four phases for which they a planned a corresponding ZMP trajectory.

WABIAN-2R [50] , achieved a more human-like straight knee walking pattern [87]. It used a knee trajectory generated by genetic algorithm without singularity. It did, however, need two extra degrees of freedom at the pelvis to solve the inverse kinematics. Li [38] took a different approach, using preview control and a virtual spring-damper to generate a COM trajectory that also straightened the knee.

Most studies have employed toe joints to achieve a natural walk, and Figure 4-2 shows the walking motions with and without toe joints.

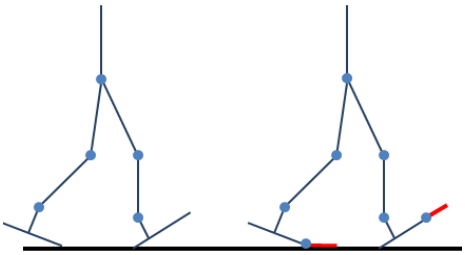


Figure 4-2 Foot rotation at the end of the single support phase.

This figure shows how hard it is to maintain the stability of the flat-footed robot if we activate a foot rotation at the end of the single-support phase. A large foot gives better stability than a small one, so for a robot with toe articulation it might be logical to increase the area of the toe pad. This solution, however, would make the robot foot quite different from a human one.

Controlling a robot with active toe joints is easier than controlling one with passive toe joints, but there are some disadvantages. Active toes need motors and gears that take up space needed for the operation of other mechanisms. They also consume power. These two points are critical to the decision to use active or passive toes.

This chapter begins with an explanation of the phases of walking motion before introducing two methods to generate the motion of toe joints. We also consider heel-contact motion because our desired ZMP trajectory gives the robot both toe-off motion and heel-contact motion. This increases stride length and avoids singularity in all joints.

## 4.2 Landing States and Desired ZMP Trajectory

To achieve natural walk motion, we need to analyze the landing states during walking. Figure 4-3. shows the landing states of the robot feet without toes and heels. As there is neither heel nor toe joint rotation, the ZMP trajectory is static under the center of the standing leg.

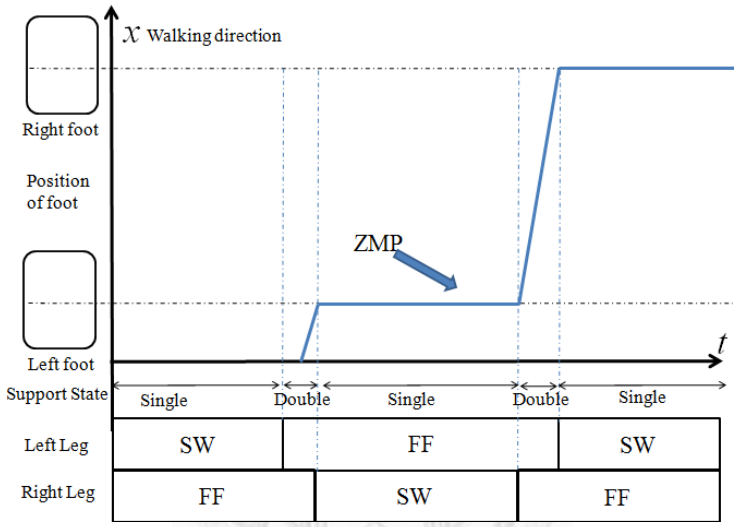



Figure 4-3 ZMP trajectory of humanoid robot with flat foot.

In Figure 4-3, SW and FF represent the swinging foot and standing (fixed) foot respectively. Since our robot will add the toe and heel motions to achieve a natural walk, the ZMP trajectory must be modified. Section 3.2 explained that the ZMP trajectory must be located in the support polygon, not only during the toe-off phase but also during the heel-contact phase. The modified ZMP trajectory, talking account of heel and toe action, must therefore match the support phases (toe, heel, and standing foot phases), as shown in Figure 4-4.

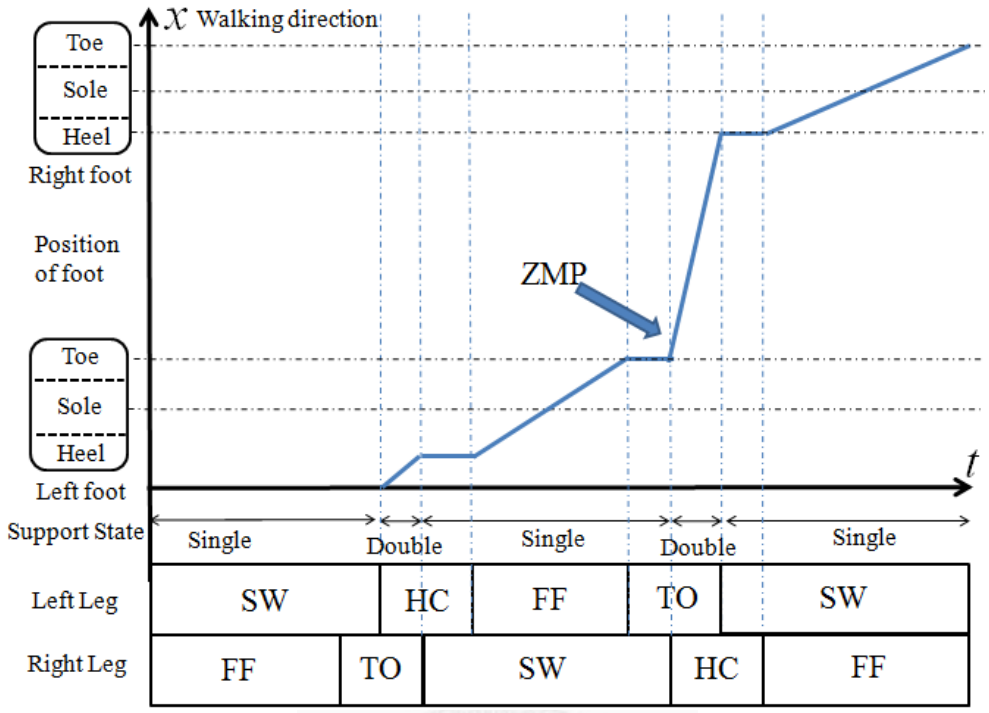



Figure 4-4 ZMP trajectory of humanoid robot with human-like foot.

All the ZMP trajectories are linked by a fourth order polynomial equation, as described in Section 3.2.

The actions of the standing foot are more important to natural walking than those of the swinging foot. As Figure 4-4 intimates, the standing foot's postures have three phases:

1. Heel-contact phase
2. Fixed foot phase
3. Toe-off phase (Toe contact phase)

The corresponding motions are shown in Figure 4-5. Looking at a single leg, we can see that its cycle starts at heel-contact and then passes through the fixed-foot and toe-off phases, then to the swinging leg and so back to heel-contact.

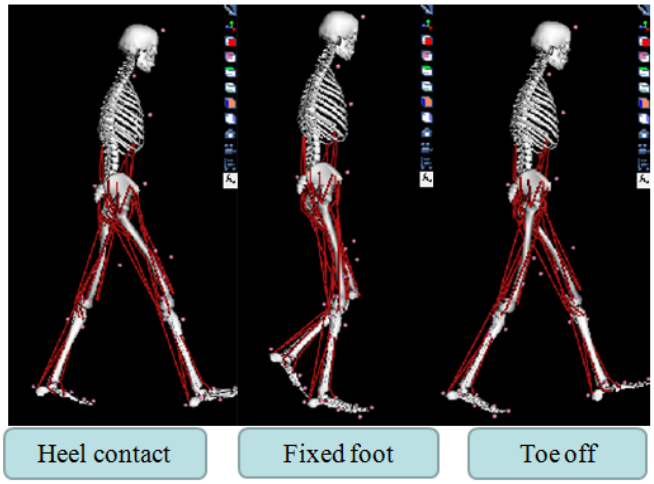



Figure 4-5 Three phases of the right foot during its support period.

We will develop two methods to achieve these phases in practice. For simplicity, our explanation applies to toe motion only, as the heel motion is very similar. Those two methods are Add Joint to Fixed Leg, and Add Toe's Contribution to Other Joints without Adding Toe or Heel Joints.

# 4.3 Adding Active Joint to Fixed Leg

The addition of a joint means modifying the IK-solver to accommodate the extra function. Since we will additionally add a joint, the IK-solver should be modified. The original IK equation appears as Eq.(2-5), and its modification as Eq. (4-1).

$$\dot{x}_{m\times 1} = \begin{bmatrix} J_{swing} \\ J_{fixed} \\ J_{COG} \end{bmatrix}_{m\times n} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_{12} \end{bmatrix}_{n\times 1} \tag{4-1}$$

where $m$ and $n$ are both 12. Both legs have 6 DOFs in the original model, and we set

the ankle as the end-effector.

Since toe motion starts at the end of the fixed-foot phase, the toe joint angle velocity needs to be solved only for the standing leg. Changing the IK-solver by setting two new modes. These two modes model the toe joint motion of the standing leg. According to kinematics, the end-effector is now the toe joint, rather than the ankle. The framework of switching from left to right standing leg modes is shown in Figure 4-6.

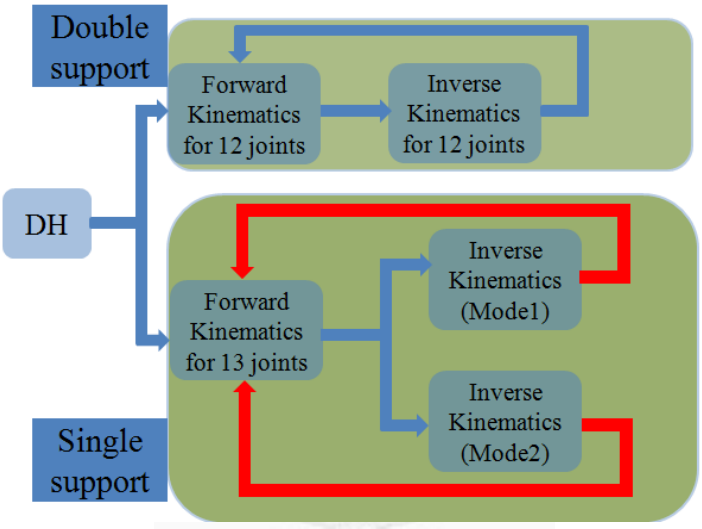



Figure 4-6 Switching the IK modes—flow chart.

The modified IK-solver is shown as Eq. (4-2). Adopting the desired ZMP trajectory, which generates the corresponding COG trajectory, will let the toe joint rotate when the ZMP is located under the toe pad. Since the ZMP trajectory is still in the support polygon, the humanoid robot will not fall. The MATLAB simulation is shown in Figure 4-7, while Figure 4-8 shows the result of taking the joint trajectory into the ADAMS simulation.

$$\dot{x}_{m\times 1} = J_{m\times n}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \vdots\\ \dot{\theta}_6\\ \dot{\theta}_{Rtoe}\\ \dot{\theta}_7\\ \dot{\theta}_8\\ \vdots\\ \dot{\theta}_{12}\end{bmatrix}_{n\times 1} \cdots\cdots \text{mode1}, \quad \dot{x}_{m\times 1} = J_{m\times n}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \vdots\\ \dot{\theta}_6\\ \dot{\theta}_7\\ \dot{\theta}_8\\ \vdots\\ \dot{\theta}_{12}\\ \dot{\theta}_{Ltoe}\end{bmatrix}_{n\times 1} \cdots\cdots \text{mode2} \qquad (4\text{-}2)$$

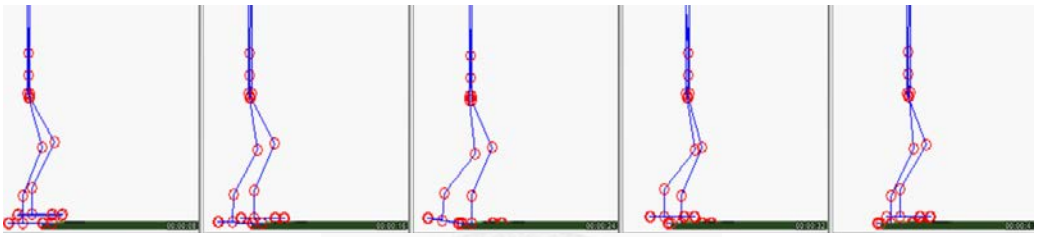


Figure 4-7 MATLAB simulation of adding toe joint into fixed leg.

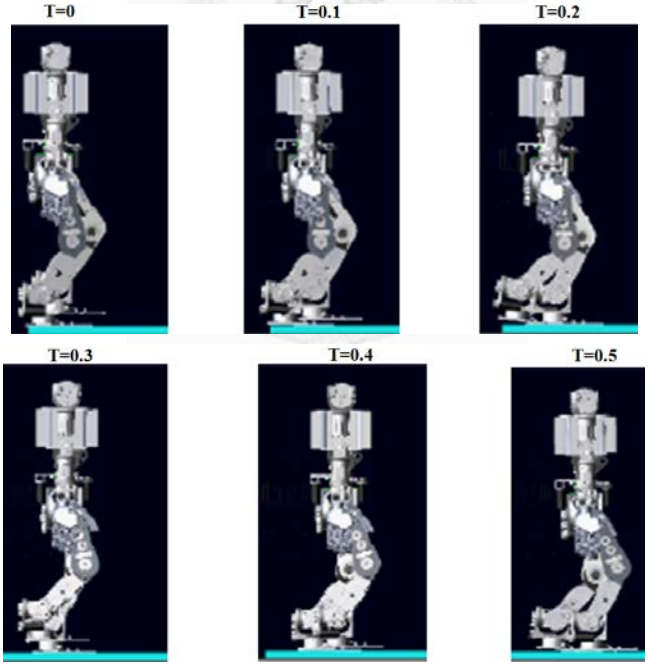



Figure 4-8 ADAMS simulation of adding toe joint into fixed leg.

In Eq. (4-2), $m=13$, and $n=12$. This method successfully produces toe motion, but some problems remain. The next section describes some other simulations and provides discussion.

This method also allows variation of the COG trajectory, as shown in Figure 4-9. Without added toe motion, our robot's stride length is 200mm when it is squatting down 30mm. With the toe motion, stride is increased half as much again (300mm) without singularity. Figure 4-10 shows the angle trajectories of the knee joints. It is difficult to reduce them to a value close to zero because the angular velocities of joints are solved by IK, which has a redundant joint in the single-support phase, making IK solution hard to control during this phase.

To list the drawbacks of this method:

1. The need for IK to solve for a redundant joint causes unexpected motion during the single-support phase. Inappropriate planning of the end-effector and COG trajectories can lead to toe rotation at the wrong phase.

2. The toe joint is not solved for the swinging leg phase, so we must provide it with a recovery trajectory.

3. Although we give the desired end-effector and COG trajectories, it only generates few values of toe rotation.

The first and third issues both concern the IK-solver, so it is difficult to overcome them. There is, however, another way to achieve toe motion.

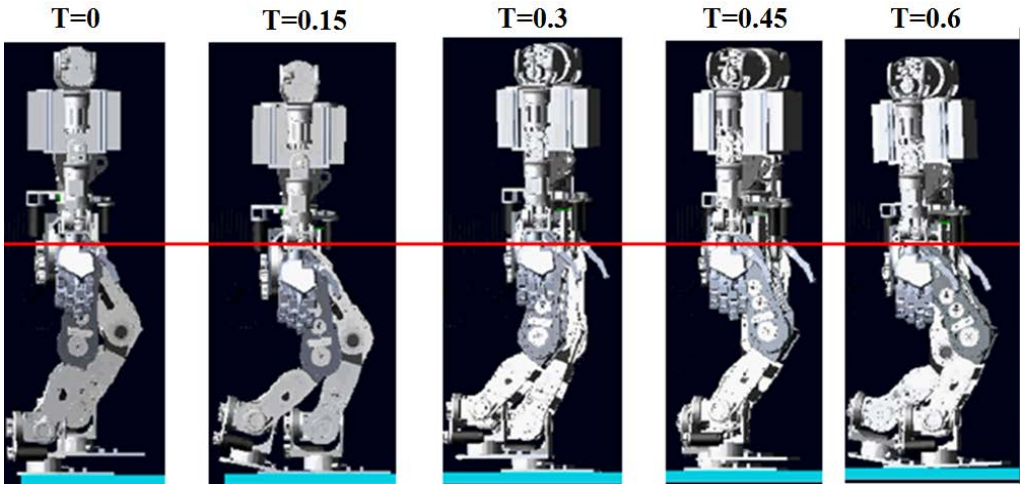



Figure 4-9 Simulation of adding toe joint to fixed leg with varying COG height.

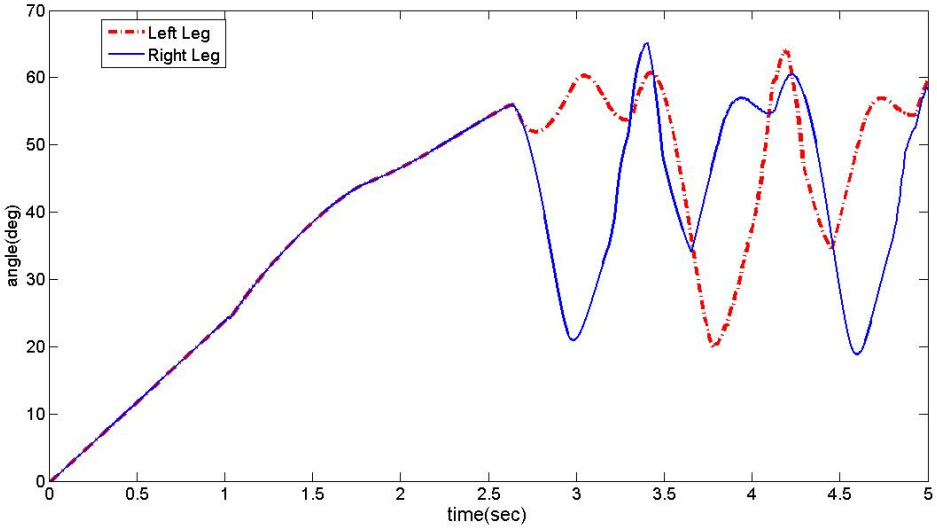



Figure 4-10 Angles of knees obtained by adding toe joint to fixed leg.

There are two limitations to consider if this method is to be used:

Firstly, the simulation always updates the position of the robot joints and linkages during the walking cycle, so we need to recompute the direction in which the robot is currently facing. If there were no toe joints, the foot itself gives the correct direction. In our robot, this vector is used to update the whole-body position, but it takes that direction from the toe pad, not the foot, as shown in Figure 4-11.

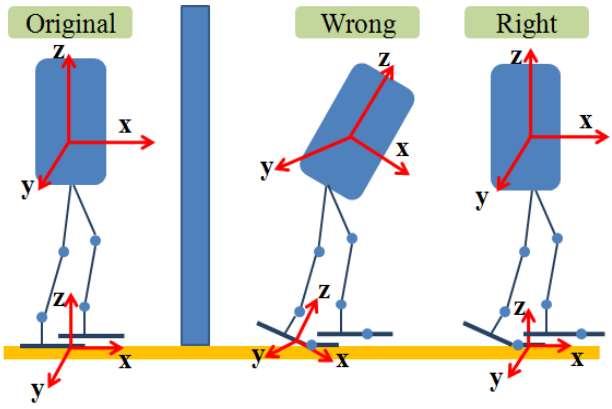



Figure 4-11 Directions of upper body and walk.

Secondly, the toe joints may rotate in the wrong direction. If this happens to the toe on the standing leg, the foot pad will hit the ground and destabilize the robot. We can avoid this possibility by modifying the Jacobian matrix. We rewrite the contribution of the corresponding toe joint as zero, and make other joints solve the problem, as shown in Eq. (4-3).

$$\dot{x}_{m\times 1} = \begin{bmatrix} J_{swing} \\ J_{fixed} \\ J_{COG} \end{bmatrix}_{m\times n} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_{12} \\ \dot{\theta}_{toe} \end{bmatrix}_{n\times 1} \tag{4-3}$$

# 4.4 Toe Motion Without Extra DOFs

If passive, sprung toe joints are to be used instead of active ones, we need to know how to control them. The passive toes with joints and spring has been designed in order to generate the toe motion. It is quite different with active toe joints in controlling since we have to make projection of COG of whole body in z-direction locate in the toe pad during toe-off phase.

Given a known toe trajectory, we can compute the contribution of the toe joint from the Jacobian matrix, and then take this known contribution to the end-effector or COG velocities. The end-effector velocities, which are also known, are subtracted from the toe's contribution, as shown in Eq. (4-4).

$$\dot{x}_{m\times 1} = J_{m\times n}\dot{\theta}_{n\times 1} \tag{4-4}$$

where $m = 12$, and $n = 14$

$$J_o = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_6} & \frac{\partial f_1}{\partial \theta_{Rtoe}} & \frac{\partial f_1}{\partial \theta_7} & \cdots & \frac{\partial f_1}{\partial \theta_{12}} & \frac{\partial f_1}{\partial \theta_{Ltoe}} \\ \frac{\partial f_2}{\partial \theta_1} & \cdots & \frac{\partial f_2}{\partial \theta_6} & \frac{\partial f_2}{\partial \theta_{Rtoe}} & \frac{\partial f_2}{\partial \theta_7} & \cdots & \frac{\partial f_2}{\partial \theta_{12}} & \frac{\partial f_2}{\partial \theta_{Ltoe}} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_{12}}{\partial \theta_1} & \cdots & \frac{\partial f_{12}}{\partial \theta_6} & \frac{\partial f_{12}}{\partial \theta_{Rtoe}} & \frac{\partial f_{12}}{\partial \theta_7} & \cdots & \frac{\partial f_{12}}{\partial \theta_{12}} & \frac{\partial f_{12}}{\partial \theta_{Ltoe}} \end{bmatrix}_{12\times 14}, \quad \theta = \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_6 \\ \dot{\theta}_{Rtoe} \\ \dot{\theta}_7 \\ \vdots \\ \dot{\theta}_{12} \\ \dot{\theta}_{Ltoe} \end{bmatrix}_{14\times 1}$$

(4-5)

where $f_m = f\left(\theta_1, \cdots \theta_{12}, \theta_{Rtoe}, \theta_{Ltoe}\right)$. Thus, $J_o$ is a known $12\times 14$ matrix, and $\dot{\theta}_{Rtoe}$ and $\dot{\theta}_{Ltoe}$ are also known parameters. Then we move all the known parameters to the left hand side of Eq. (4-4), as show in Eq. (4-6).

$$\dot{x}_{12\times1} - \begin{bmatrix} \frac{f_1}{\partial\theta_{Rtoe}} \times \dot{\theta}_{Rtoe} + \frac{f_1}{\partial\theta_{Ltoe}} \times \dot{\theta}_{Ltoe} \\ \vdots \\ \frac{f_{12}}{\partial\theta_{Rtoe}} \times \dot{\theta}_{Rtoe} + \frac{f_{12}}{\partial\theta_{Ltoe}} \times \dot{\theta}_{Ltoe} \end{bmatrix}_{12\times1} = J\dot{\theta}$$

$$J_{12\times12} = \begin{bmatrix} \frac{f_1}{\partial\theta_1} & \cdots & \frac{f_1}{\partial\theta_{12}} \\ \vdots & \ddots & \vdots \\ \frac{f_{12}}{\partial\theta_1} & \cdots & \frac{f_{12}}{\partial\theta_{12}} \end{bmatrix}, \quad \dot{\theta}_{12\times1} = \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_{12} \end{bmatrix} \tag{4-6}$$

where $J$ is a $12\times12$ matrix. The modified equation will change the end-effector trajectory, $\dot{x}$, but still make the humanoid robot perform the correct motion. The end-effector is also the toe joint, and has the desired ZMP trajectory. By adopting this method, the toe joint will not rotate in the wrong direction. The results and simulations are discussed in the next section.

## 4.4.1 Simulations of Toe Motion Without Extra DOFs

The toe trajectory can be designed with a third-order polynomial function, as shown in Eq. (4-7). The trajectory itself, shown in Figure 4-12, avoids any discontinuities, accelerations, and jerks that will destabilize the robot. This toe joint trajectory will work for active toe joints, but this simulation uses passive toe joints—we control the other joints to make up for the lack of toe control. Applying our new trajectory to ADAMS simulation yields the results shown in Figure 4-13 and Figure 4-14.

$0 \le t \le t_1$ toe off phase , $t_1 \le t \le t_2$ during toe recovering phase

$$\theta_{toe,t} = at^3 + bt^2 + ct + d$$

I.C. $\theta_{toe,o} = 0, \quad v = 0$ at $t = 0$

$$\theta_{toe,t_1} = \theta_{\max}, \quad v = 0 \text{ at } t = t_1 \quad , 0 \le t \le t_1$$

(4-7)

$$\theta_{toe,t_1} = \theta_{\max}, \quad v = 0 \text{ at } t = t_1$$

$$\theta_{toe,t_2} = 0, \quad v = 0 \text{ at } t = t_2 \quad , t_1 \le t \le t_2$$

$$\dot{\theta}_{toe,t} = \theta_{toe,t} - \theta_{toe,t}$$

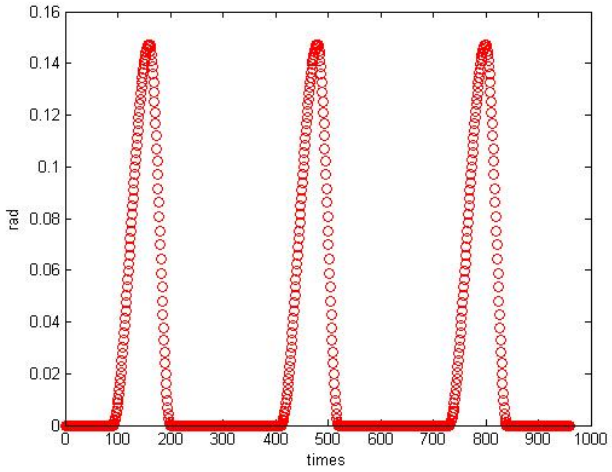



Figure 4-12 Reference trajectory of the toe joint.

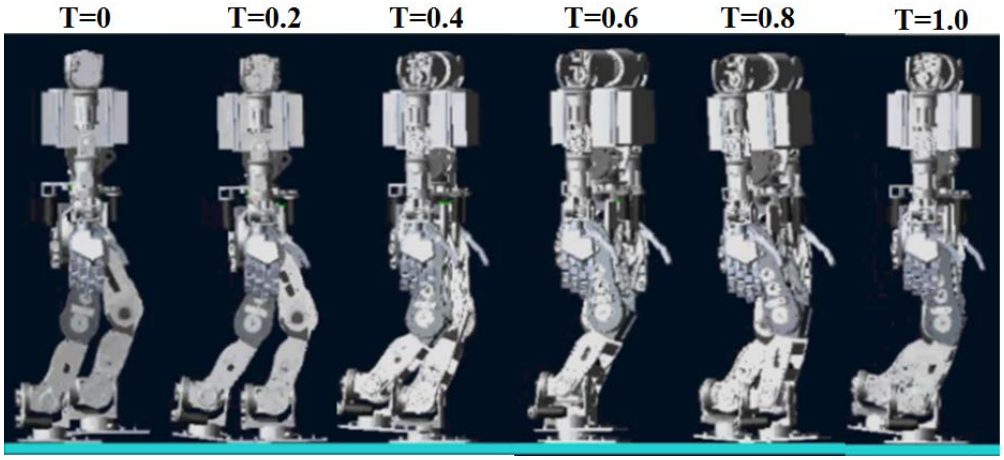



Figure 4-13 Simulation of toe motion without extra DOFs in ADAMS.

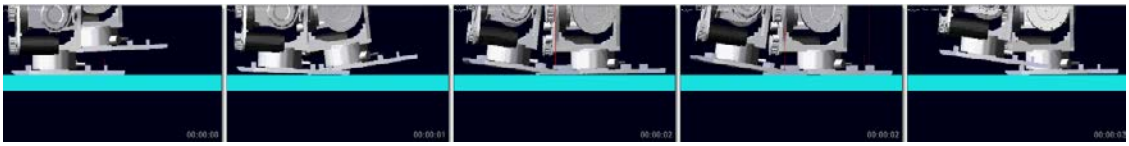


Figure 4-14 Close-up of the toe-off motion.

### 4.4.2 Results and Comparisons

This method allows arbitrary assignment, within their limiting ranges, of the rotating angles of the toe. Given the toe joint angles, we can use geometry to compute the stride length, and this works out at 500mm, much larger than that of the robot without toe joints.

Other humanoid robots squat throughout the walking cycle, but this method, once COG is allowed to vary, makes the knees nearly straight, as shown in Figure 4-15. Power consumption is reduced, and the walking motion is similar to a human's.

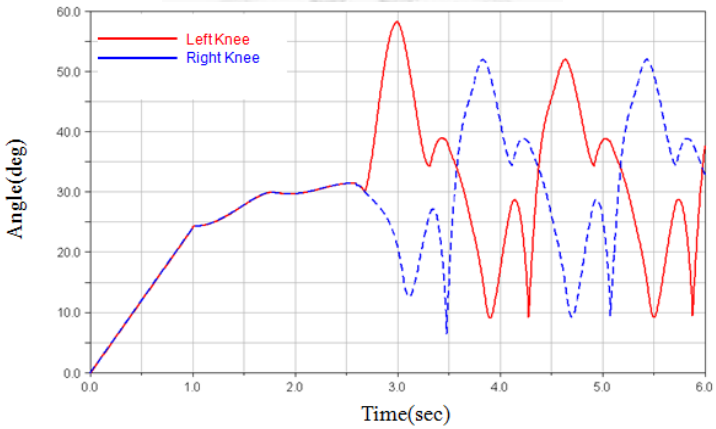



Figure 4-15 Angles of knees by a method without extra DOFs

A crucial part of this study is the comparison of the two methods outlined above—Add Joint to Fixed Leg, and Add Toe's Contribution to Other Joints Without Extra DOFs. Their advantages and disadvantages are shown in Table 4-1. As the second method omits the redundant joint for the IK-solver, unexpected motion is avoided. Passive toe joints do not consume power. The first method increases step length, but the second increases it even more. We have therefore decided to take our research forward

with the second method, to achieve our next goal, heel-contact motion.

Table 4-1 Comparison of methods 1 and 2

| | Maximum step length | Minimum knee joint angle | Unanticipated motion | Easily reach the singularity |
|---|---|---|---|---|
| Method 1 | 300mm | 20° | yes | no |
| Method 2 | 500mm | 9° | no | yes |

# 4.5 Motion of Heel Contact

Heel-contact, or heel strike, is as natural to human walking as the toe-off motion described above. It is particularly obvious when a human is taking large strides. Figure 4-16 illustrates contact area during walking cycle.

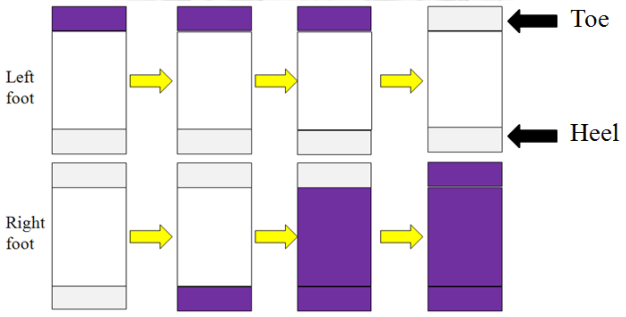



Figure 4-16 The contact area of humanoid robots during walk.

If the human foot and shin were fixed at a right angle, the only way to increase stride length would be to squat, but a flexible ankle and a straight knee permit the heel to strike before the sole of the foot, allowing a longer stride.

There is a second reason for using heel-contact motion: it lengthens the double-support period. During long strides, the COG of the whole body needs to move forward rapidly, leaving the support polygon and endangering stability. Heel-contact

puts the swinging leg down earlier, starting the double-support phase as soon as possible and minimizing the period of instability. The support area is defined by the toe pad of the standing leg and the heel pad of the swinging leg, as shown in Figure 4-17.

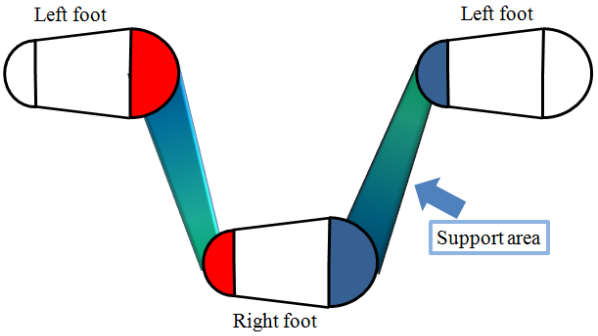



Figure 4-17 Support area during heel-contact phase.

There are two distinct actions in heel-contact: Angling the foot just before heel-contact, and changing that angle just as the heel-contact phase ends. The method employed to angle the foot is similar to toe-off motion, so we have added the joint without adding any control. The direction of foot angle adopted during the toe-off phase is different from the one adopted during the heel-contact phase, but there is no need to add the contribution of heels to solve the IK equation. We can use the motion of toe-off in the opposite rotating direction, as shown in Figure 4-18. We need to guarantee the position of the foot pad in the air, as the wrong angle will let the foot pad hit the ground.

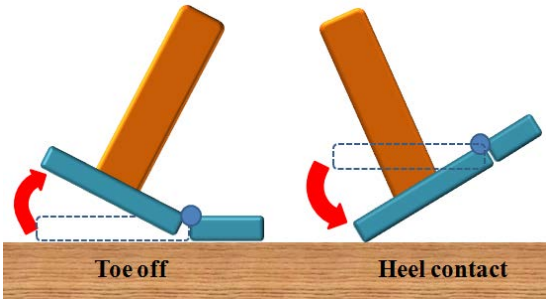



Figure 4-18 Rotating directions of toe-off and heel-contact motions.

The most difficult problem of heel-contact motion is the recovery of the bent foot pad. Our end-effector trajectory for the swinging leg is linked by MTS [15] , and is planned before the end-effector moves. That enables us to find the descending linear velocity of the foot pad in the z-direction, which we can convert to angular velocity for the toe joint. The foot pad will go down and recover to match the ground at the same time. The heel joints are passive, but need not be accounted for in the forward kinematics, as knowing the relative positions of the heel and footpad is enough. When the heel pad contacts the ground, double support phase starts. As long as the COG trajectory allows it, the heel joints can bear part of the robot's weight from the humanoid robot.

The compensating equation is shown in Eq. (4-8) and Figure 4-19. The angular velocity of the toe joint needs to match that of the falling foot pad, or the impact force generated on contact will induce instability. The torsion springs in the passive heel joints absorb that force, and for them to do so efficiently they must have the correct elastic coefficient; they will then play the same role as human heels. The simulation of

the heel-contact motions is shown in Figure 4-20, and a close-up of the heel is shown in Figure 4-21.

$$
\begin{aligned}
&0 \le t \le T \\
&z(t-1) > ground \quad z(t) < ground \quad z(T) > ground \\
&Z_1 = z(t) - ground \\
&Z_{i+1} = z(t+i) - z(t+i-1) \\
&\theta_i = \sin^{-1}(\frac{Z_i}{L}) \qquad i = 1, 2, \cdots
\end{aligned}
\tag{4-8}
$$

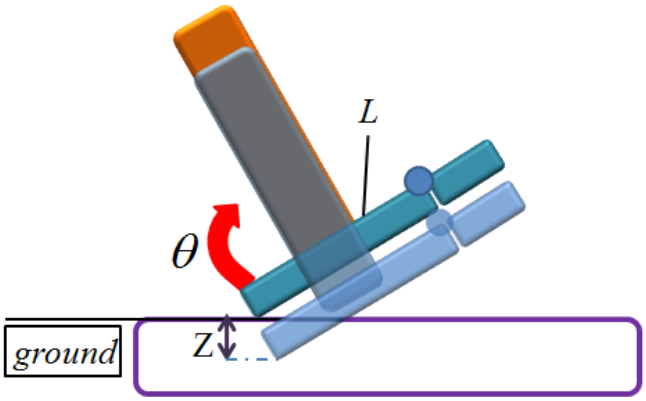



Figure 4-19 Recovering heel motion.

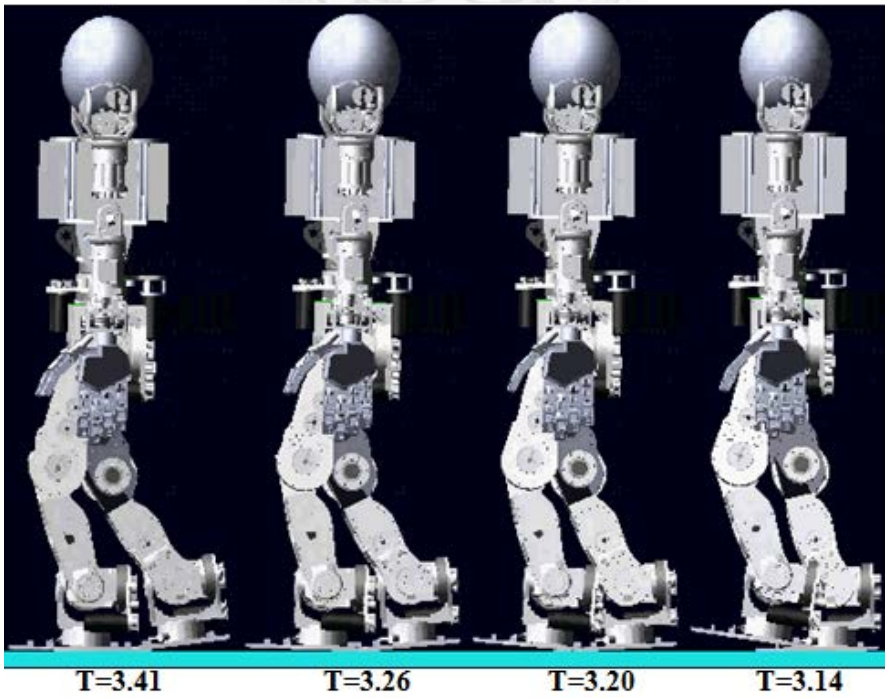



Figure 4-20 Walking motion with heel-contact and toe-off motions.

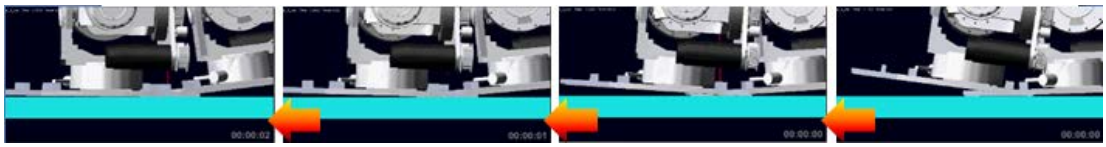


Figure 4-21 Close-up of heel-contact motion (robot walking toward the left).

$z$ is the height of the foot pad from ground, and $ground$ is the position where it will land. The parameter $L$ is the length of our robot foot, as shown in Figure 4-19.

# 4.6 Summary

This chapter introduces research into natural walking for humanoid robots, and provides the results of simulations as well as description and discussion.

We have expanded on the theme of desired ZMP trajectories, explained in Chapter 3, defining the phases of the single-support period of human walking as heel-contact, fixed-foot, and toe-off. This enables us to define the areas in contact at each phase.

Next, we compared the flat feet with human-like feet, showing the many advantages of the latter.

The third topic, methods for achieving toe-off motion, forms the key point in this chapter. We proposed two ways, Add Active Joint to Fixed Leg and Toe Motion without Extra DOFs to Solve IK. Comparison of the two methods led to our decision to adopt the second method. We also described heel-contact motion.

We can now combine the research of Chapter 3 and Chapter 4 to describe a way of achieving a natural walk with a designed ZMP trajectory and lengthening stride by adopting a variable COG height. The addition of momentum compensation will create a

walk that is not only natural, but very similar to the walk of a human being, as shown in Figure 4-22. Figure 4-23 shows the joint trajectories, and Figure 4-24 shows that the maximum step length achieved is 500mm.

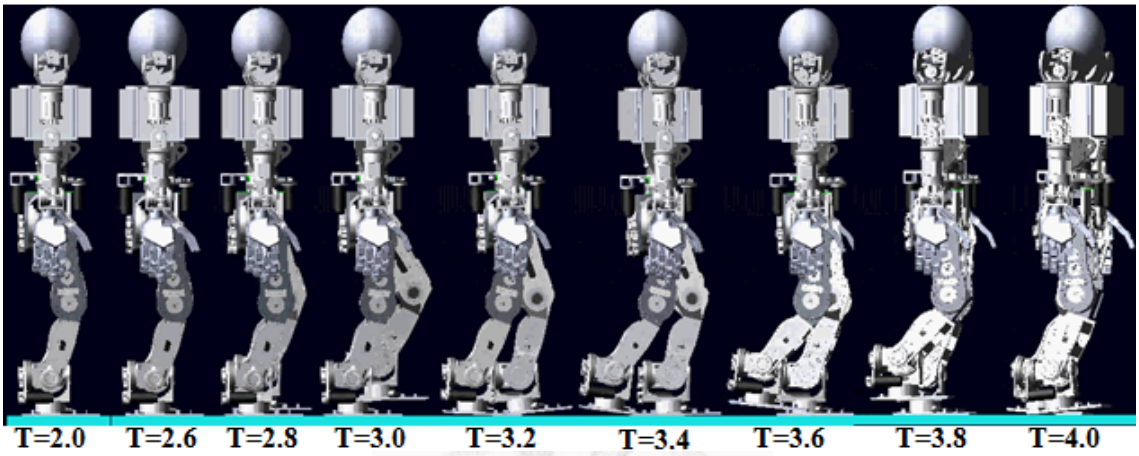



Figure 4-22 Simulation of natural walk with momentum compensation. Step length is 400mm.

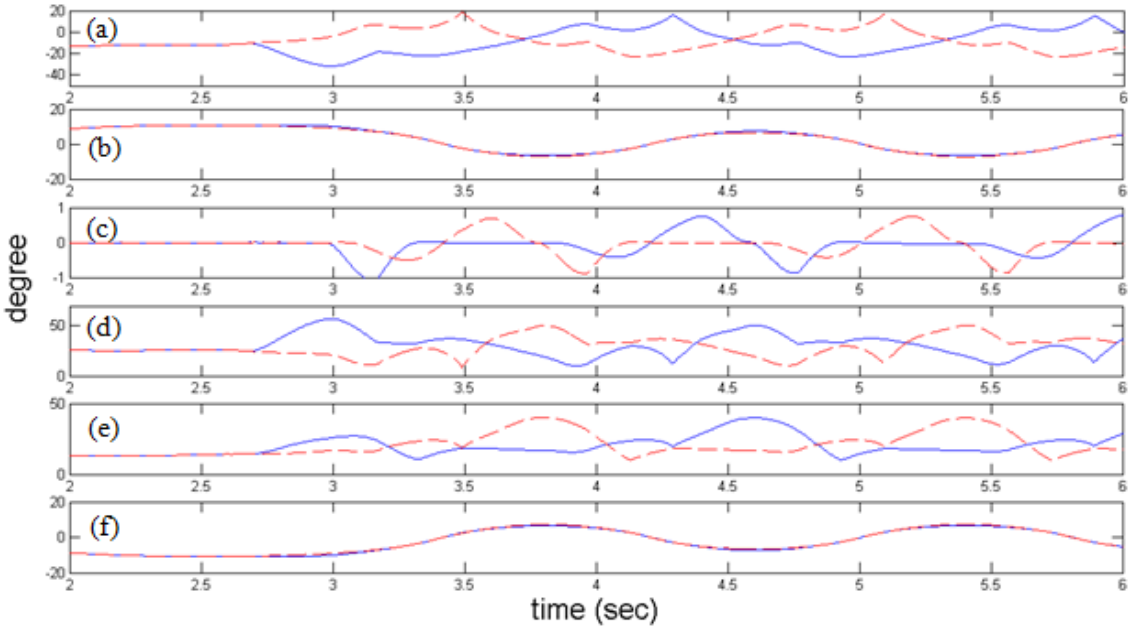



Figure 4-23 The values of joint trajectories from our humanoid robot. (a) Hip pitch, (b) Hip roll, (c) Hip yaw, (d) Knee pitch, (e) Ankle pitch, (f) Ankle roll.

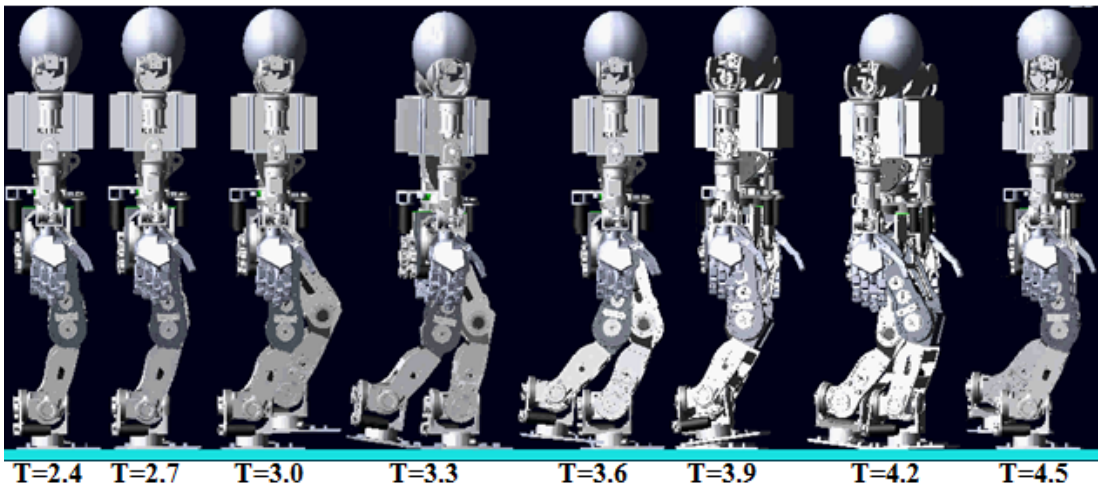



Figure 4-24 Simulation of natural walk with walking speed 625mm/s and step length 500mm.

A comparison of the walking speed of our robot with those of other humanoid robots, illustrated in Table 4-2, shows that our speed is satisfactory, if not spectacular.

One of the advantages for natural walking is less power consumption. Figure 4-25 is the result of summing the power consumed by each of the twelve joints, using ADAMS, so we can compare conventional and natural robot walking at 0.8 sec per step. The maximum step length is 250mm when the COG height goes down 20mm from the standing posture.

Table 4-2 Walking speed and leg length of different humanoid robots.

| | WABIAN-2 | HRP-2J (simulation) | ROBIAN (simulation) | LOLA | Our Robot (simulation) |
|---|---|---|---|---|---|
| Speed (km/hr) | 1.875 | 1.62 | 2.34 | 2.5 | 2.25 |
| Leg Length (mm) | 700 | 750 | 1100 | 970 | 715 |

As ADAMS is the simulating environment, the colliding setting is similar to that of

a rigid body and will cause unanticipated peaks when the foot strikes the ground. The total power consumption for the flat foot is $2.0597\times10^{4}$ W in flat foot, while the human-like foot consumes much more efficient $1.7261\times10^{4}$ W.

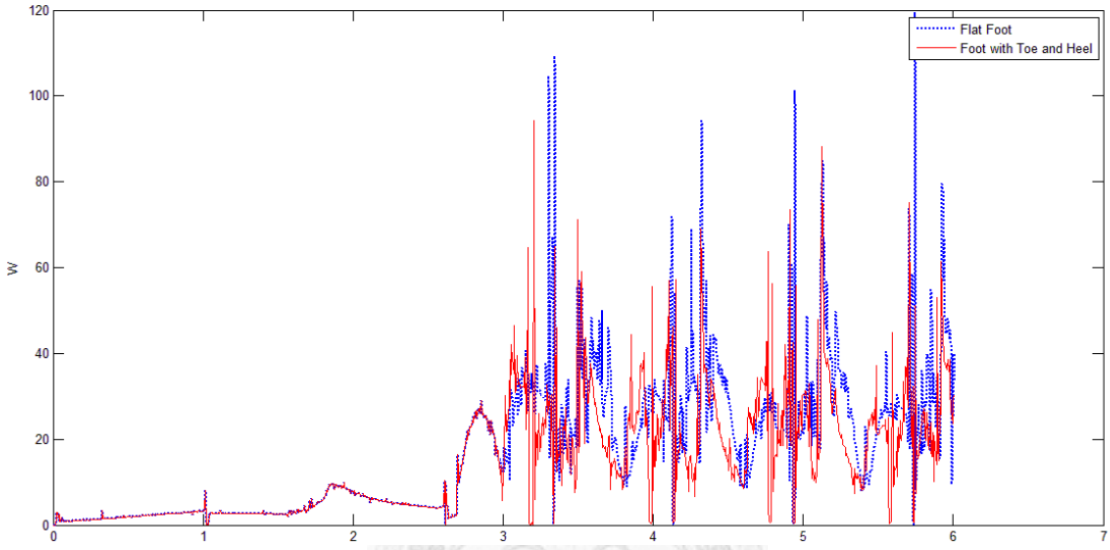



Figure 4-25 Power consumption of the flat foot and human-like foot at 250mm/step.

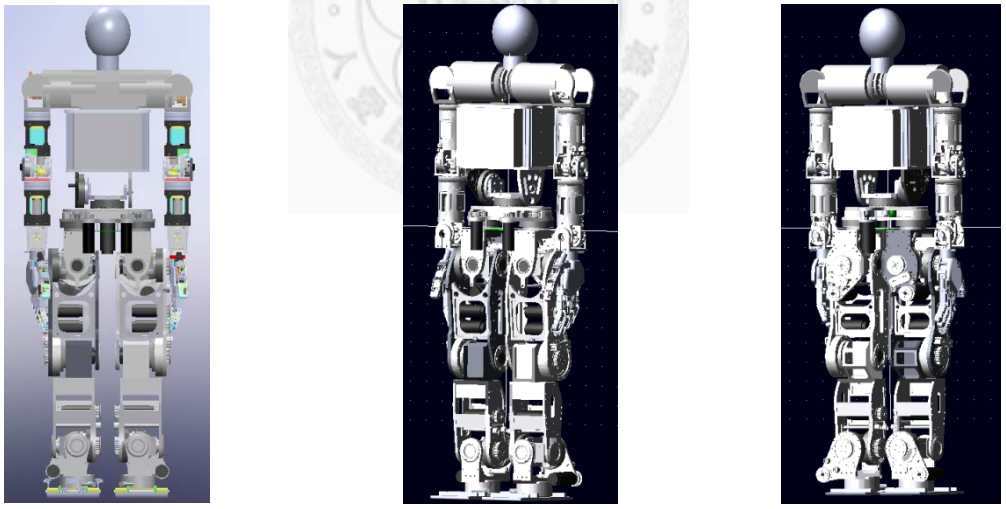
The next chapter describes the mechanism design and mechatronics system of our humanoid robots, with feet design for natural walking.

# Chapter 5 Mechanism Design and Mechatronics System

## 5.1 Introduction

In previous chapters, we have described the adjustable ZMP trajectory, variable COG, momentum compensation, and natural walk. We can incorporate these factors into simulation in which a humanoid robot performs a natural walk.

We have designed and built a humanoid robot named “Voyager” to verify our methods, and prove the practicability of the tasks simulated using the same design in ADAMS.



(a) Front view (b) Front View (a) Back View

Figure 5-1 Overview of the humanoid robot shown in SolidWorks and ADAMS.

Our robot, Voyager, is an anthropomorphic humanoid with modularized arms, simplified trunk, and five-fingered hands. This chapter presents the development of the humanoid mechanism with a trunk, foot pad mechanism, and electrical system. It

includes our arrangement of the electric hardware and the solutions needed for some hardware and firmware problem before we can conduct long-term experiments stably.

The next section introduces the mechanisms we design for Voyager. Section 5.3 covers the electrical system.

## 5.2 Mechanism Design

The original model was designed by Cheng [5], as shown in Figure 5-2. We apply all our simulations data and specify the design features we need, such as joint limitation, walking posture, squatting motion, measured forces, torque, and power consumption, so we can adapt the model to include the factors in our design. The resulting specification describes a new humanoid mechanism with the same work capabilities as the original, but with a more anthropomorphic shape, as shown in Figure 5-3.

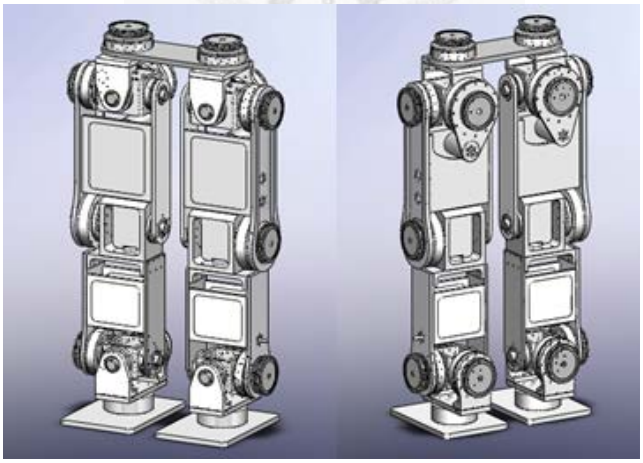


Figure 5-2 Front view and back view of the humanoid mechanical structure [5].

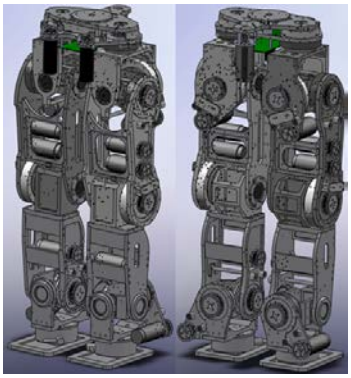


Figure 5-3 Front view and back view of the new humanoid mechanical structure.

To link the humanoid mechanism to the upper modularized arms and hands, we design a new waist and trunk mechanism. The concept and specification of the mechanism appear in the next section.

## 5.2.1 Design Specification

Before we start designing, we list the constraints we need to apply. Apart from cost, these include the shape, weight, and volume of the mechanism. For example, the boards, blocks and even the supporting structure are all fairly light in comparison with the much heavier motors and harmonic drives. The robot has to be easily assembled and disassembled, and we also have to design a robot suitable for the experiments planned for it and adaptable for future applications. To summarize our design concepts, our robot must:

i. be able to reproduce the fundamental human movements,

ii. be easily checked, fixed, assembled, and disassembled,

iii. endure an impact with a compact structure,

iv. have an anthropomorphic appearance while being roomy enough to contain and organize the electrical components,

v. have a strong transmission capable of delivering a sufficiently high speed.

After clarifying the concept, we use joint configuration similar to that used for the humanoid robot ASIMO (Figure 5-4). The hip has three joints (roll, pitch, and yaw), the knee one (pitch), and each ankle two (pitch and roll). The joints link the four coaxial mechanisms.

Having considered the design concept for humanoid robots and chosen the fundamental configuration, we modify our humanoid robot's movable range of joints as shown in Table 5-1. This allows our robot to perform the experiments we have designed for our research. Having decided the movable range, we can determine the principal specification for our robot. The detailed specification is shown in

Table 5-2, and the photos in Figure 5-5.

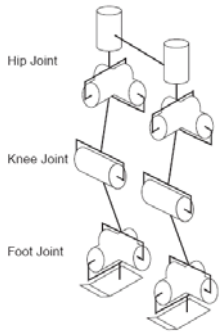



Figure 5-4 The arrangement of links and joints of ASIMO.

Table 5-1 Comparison of movable of joints

| Joints | | Human Being | HRP-2 | Hubo | Voyager |
|---|---|---|---|---|---|
| **Trunk** | Yaw | -40~-40 deg | -45~45 deg | Unknown | -60~60 deg |
| | Pitch | -30~45 deg | -5~60 deg | None | -15~40 deg |
| **Hip** | Yaw | -45~45 deg | -45~30 deg | -45~0 deg | -45~23 deg |
| | Roll | -45~20 deg | -35~20 deg | -31~28 deg | -40~15 deg |
| | Pitch | -125~15 deg | -125~42 deg | -90~90 deg | -72~22 deg |
| **Knee** | Pitch | 0~130 deg | 0~150 deg | -10~150 deg | -90~142 deg |
| **Ankle** | Roll | -20~30 deg | -20~35 deg | -23~23 deg | -30~39 deg |
| | Pitch | -20~45 deg | -75~42 deg | -90~90 deg | -40~45 deg |
| **Toe** | Pitch | -20~90 deg | 0~60 deg | None | 0~80 deg |

Table 5-2 Principal Specification of Voyager

| | | |
|---|---|---|
| **Planned Dimensions (mm)** | Height | 1400 |
| | Depth | 310 |
| | Width | 445 |
| **Planned weight inc. batteries (kg)** | | 60 |
| **Number of DOFs** | Head | 21 DOFs |
| | Arm | 2 Arms × 6 DOFs |
| | Hand | 2 Hands × 12 DOFs |
| | Waist | 2 DOFs |
| | Leg | 2 Legs × 6 DOFs |
| | Total | 71 DOFs |

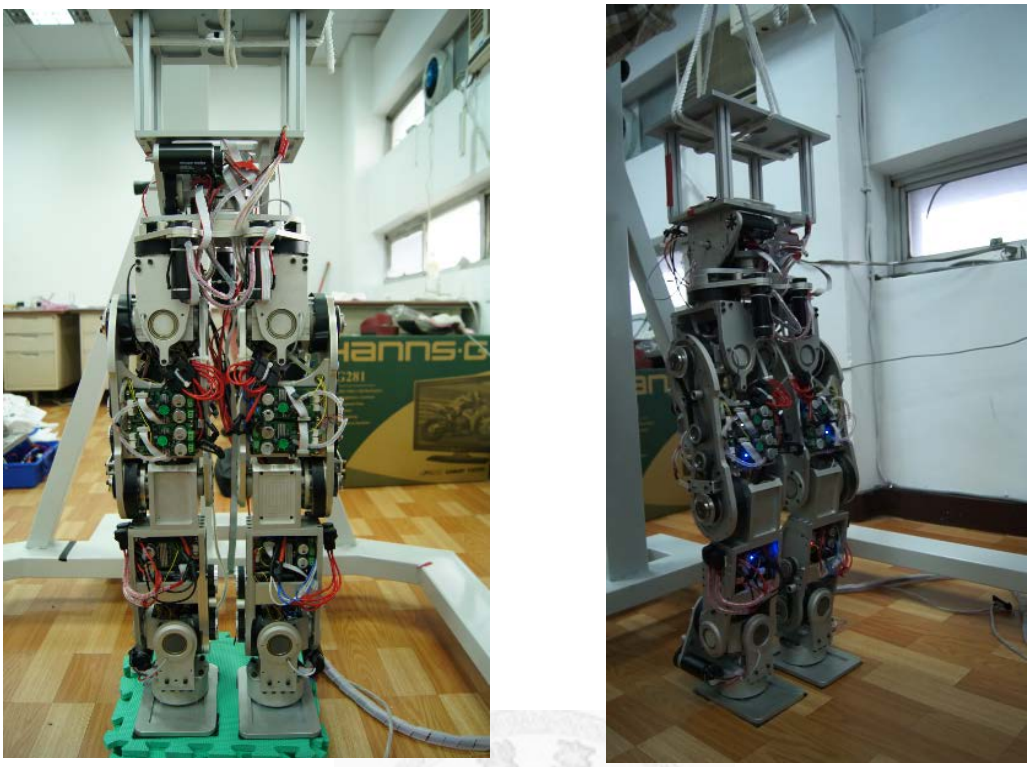


Figure 5-5 The photo of constructed humanoid robot.

### 5.2.2 Transmission and Fundamental Settings

To reduce unnecessary power consumption, we set the applied voltage of actuators at 48 volts. The actuators, listed in Table 5-3, are chosen based on the power consumption simulation results. Cheng [5], measured the required torque and power for different motions in ADAMS, as shown in Figure 5-6. The transmission, including the motors, belts, and pulleys, are chosen by our colleague Chao [4]. Details are listed in Chao [4]. We ensure that the transmission can bear the limitations of robot motion.

Table 5-3 Maximum power requirement of each actuator from [5]

| Joints | | Motion | | | Actuator spec. (48V) |
|---|---|---|---|---|---|
| | | Squatting | Walking Forward | Curving | Max. Power |
| **Hip** | Yaw | 3.4W | 5.6W | 15W | 60W |
| | Roll | 4.8W | 58.3W | 59.8W | 150W |
| | Pitch | 23.6W | 25.9W | 24.5W | 90W |
| **Knee** | Pitch | 68.3 W | 55.6W | 19.6 W | 150W |
| **Ankle** | Roll | 6.6W | 43.8W | 23.8W | 90W |
| | Pitch | 28.6W | 27.5W | 33.4W | 90W |

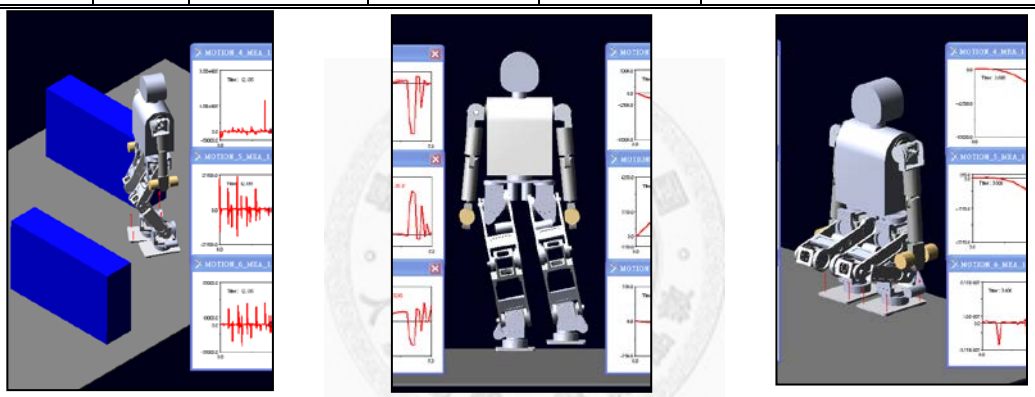


Figure 5-6 Measuring the torque and power consumption by different motions.[5]

Experience shows that the support structure makes up more than 60% of the weight of the whole robot. We structure material was chosen to provide maximum strength for minimum weight and cost. Titanium has the highest strength-to-weight ratio of any metal, but is too expensive. Aluminum alloy and steel are cheaper, but still light and strong. From aluminum alloy 6061-T6, 7075-T6, and steel 4130 (compared in Table 5-4), we selected 7075-T6 for our mechanism.

Table 5-4 Strength of aluminum alloy 6061-T6, 7075-T6, and steel alloy 4130

| Materials | Ultimate Tensile Strength | Yield Strength | Elongation |
|---|---|---|---|
| **6061-T6** | 290MPa | 241 MPa | 9-10 % |
| **7075-T6** | 510 - 538 | 434-476 MPa | 5-8 % |
| **Steel 4130** | 590–760 MPa | 480–590 MPa | 20-30% |

Since we want to manipulate the multiple DOFs robot repeatedly, automatic homing is necessary, so we set up proximity sensors for this task. The limit switch for each joint automatically signals when it closes to its limit. Figure 5-7 shows the arrangement of the proximity sensors and the limit switches. High-temperature driver failures are guarded against by strategically placed heat sinks and cooling fans, as shown in Figure 5-8.

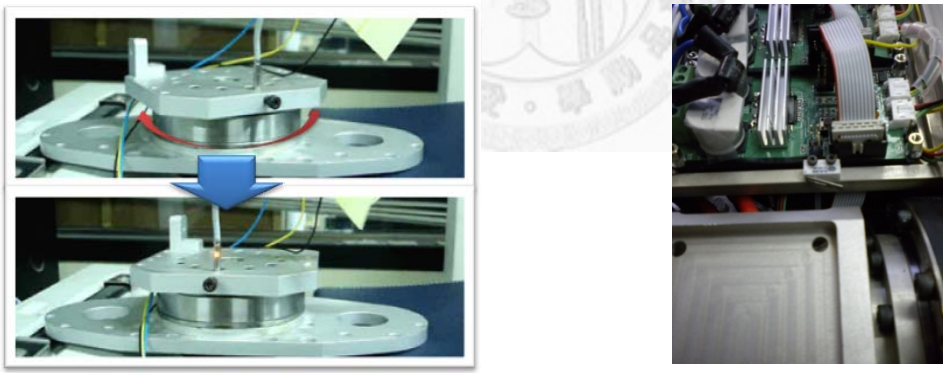


Figure 5-7 The proximity sensor and limit switch.

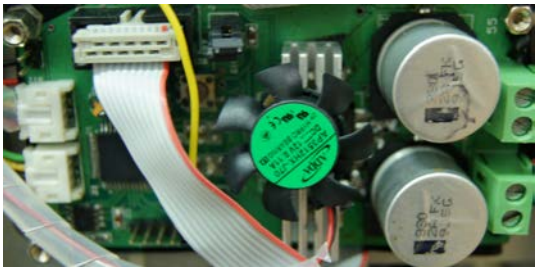


Figure 5-8 The sink and cooling fan for driver.

### 5.2.3 Modified Foot Pad

The new humanoid mechanism needs foot pads modified to facilitate natural walking, as the original model is flat-footed. More and more robots have toes and heels [1, 18, 65, 73]. Quezdou et al. compare several kinds of feet for humanoid robots [52]. They examine four different feet, including active and passive feet, and discuss the pros and the cons of each. Our foot pad has two passive joints (toe and heel). The detailed specification and mechanism are shown in Table 5-5 and Figure 5-9.

Table 5-5 Specification of the proposed foot pad mechanism

| | | |
|---|---|---|
| **Dimensions** | Height | 27(mm) |
| | Depth | 210(mm) |
| | Width | 140(mm) |
| **D.O.F.** | Pitch ×2 | 0~80 degree |

The anthropomorphic foot pad is in three parts—toe, heel, and sole. The toe and heel share the same design concept. We chamfered the leading edge to avoid instability during the toe-off and heel-contact phases. The motions of the passive joints during support and recovery are handled by one torsion spring for each toe and heel. The contact areas of toe and heel are bigger than they are in humans, to ensure that the ZMP point remains in the support polygon during the toe-off and heel-contact phases.

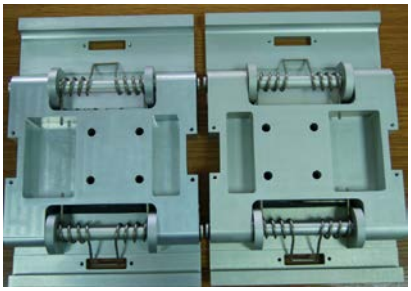


Figure 5-9 Mechanisms of foot pads with toe and heel joints

### 5.2.4 Structure Stress Analysis

Stress analysis plays an important role in our design, as we need to know how great an impact the mechanism can bear. We set the impact forces at 10 times the maximum contact forces. Stress analysis not only ensures that the mechanism will not break, but also allows us to design the lightest possible weight that satisfies the set stress conditions, as shown in Figure 5-10.

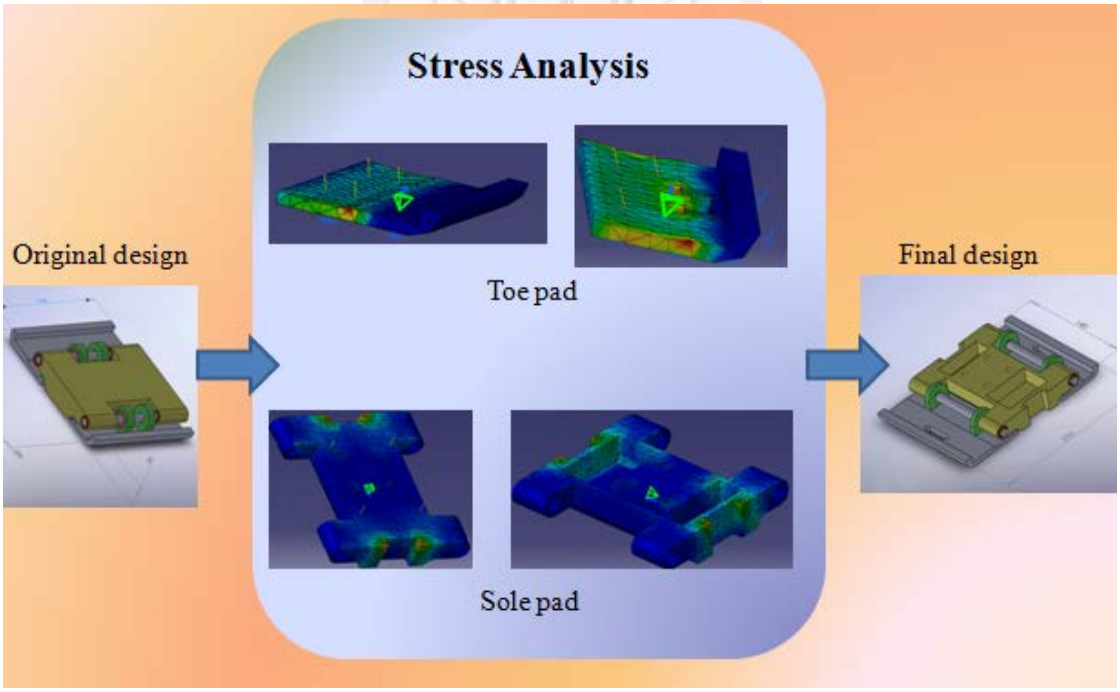



Figure 5-10 Mechanism design with stress analysis.

The mechanism is not only exposed to impact breakage, but also to shear

stress in the moving parts. According to the maximum shear stress theory, the yield strength in shear is half the normal yield strength. To overcome this, we changed our safety factor to “20.” The stress analysis is calculated in CATIA, and the results are shown in Figure 5-11, while Table 5-6 shows specifications of applied forces and torques.

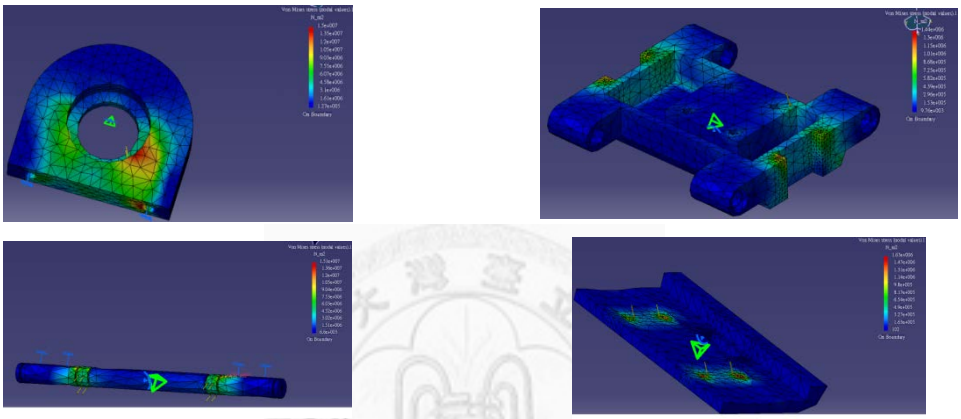


Figure 5-11 Result of the stress analysis in CATIA.

Table 5-6 Specification of applied forces and torques

| **Applied forces/torques** | $F_x$ | $F_y$ | $F_z$ | $M_x$ | $M_y$ | $M_z$ |
|---|---|---|---|---|---|---|
| **Original** | 50N | 80N | 600N | 50Nm | 50Nm | None |
| **Modified** | ±200N | ±200N | 600N | ±60Nm | ±100Nm | ±60Nm |

The whole humanoid mechanism is subjected to stress analysis so that we can achieve maximum strength for minimum weight. The flowchart below shows the overall organization of mechanical design.

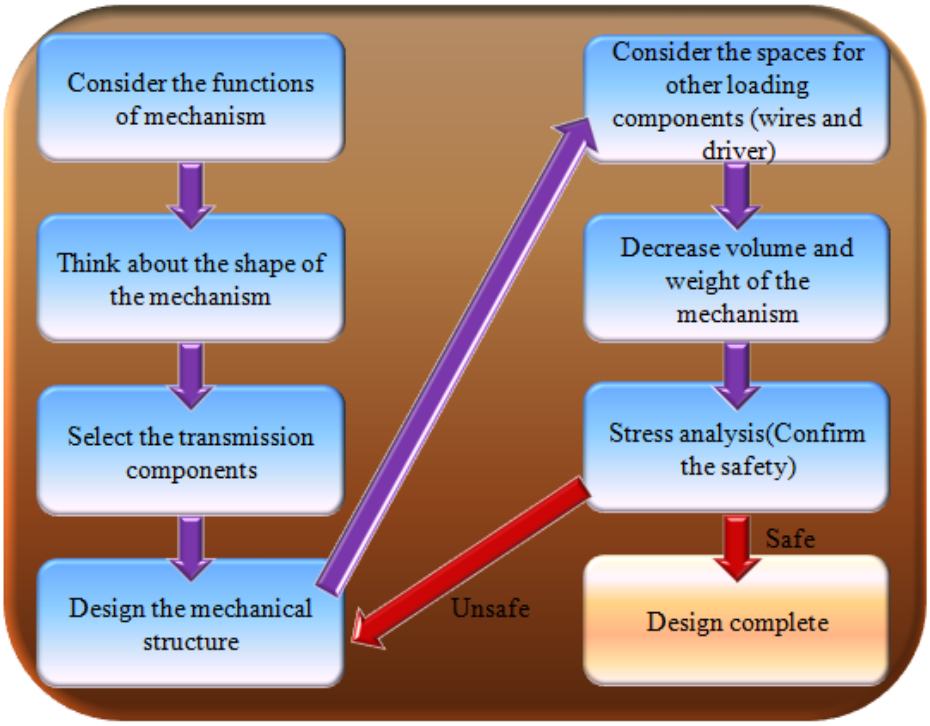



Figure 5-12 The flowchart of mechanical design

# 5.3 Mechatronic System

The mechanism has been designed, and then we will introduce the mechatronic system we utilize. For our robot, there are many feedback from sensors, motors, and other drivers. A robot system is usually composed of hardware units, computers, actuators, sensors, and controller. How to collect the data or commands from those units is quite important. If the number of the units are too many, the robot system becomes complex. For our robot, we expect that we can control more than 100 units in real-time, so the well-scheduled communication system is essential.

In order to increase the transmitting speed, some researches utilize the centralized control architecture [12, 25, 26, 34, 47]. Although centralized control system,

IEEE-1394, RS-232, and PCI bus, works with high speed, the wiring is too complicated to set up.

Our robot system adopts CAN-Bus system [37, 44, 76, 77, 80]. Although the band width of communication system of CAN may be a bit low, this system has the reliable and real-time properties. It is necessary to combine the universal serial bus (USB) when we perform and give commands from computer. Its master-slave structure needs numerous cables to connect multiple nodes. Therefore, we integrate the CAN and USB into our control system to perform commands between computers, actuators, and sensors.

In our proposed architecture, the PC links up CAN networks with USB-CAN adapter, and the networks transmit data with sampling time: 5 ms. The overall architecture is shown in Figure 5-13.

The USB-CAN adapter module (Figure 5-14) is the master unit in our CAN bus network. Beside transmitting command as a master, it mainly manages the scheduling of network communication according to our protocol for merging the USB and CAN. The micro controller of this adapter is PIC (dsPIC32MX795F512H). The detail functions we use are shown in [4].

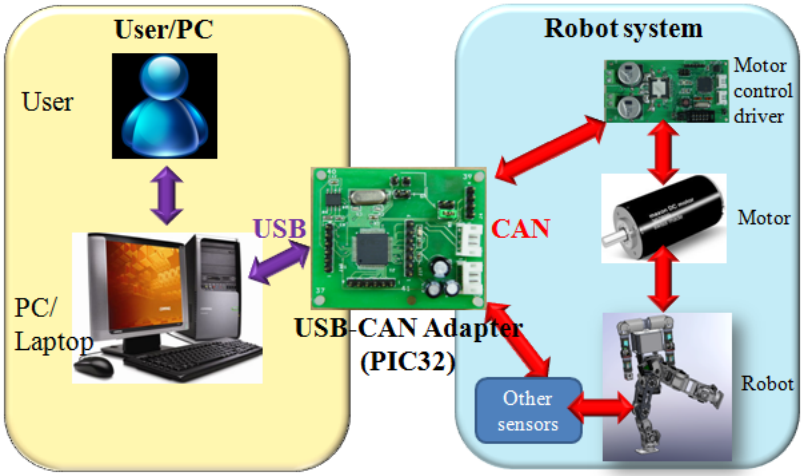



Figure 5-13 Proposed robot control system with CAN-BUS and USB.

For the joint controller, we set up 12 electrical boards (Figure 5-14) which include the microprocessor dsPIC30F4011 (PIC30) near the joints. Commands for transmission from the master node to the slave nodes will be translated into the PWM or other commands to control the actuators and sensors. In this master-slave system, transmission is bidirectional.

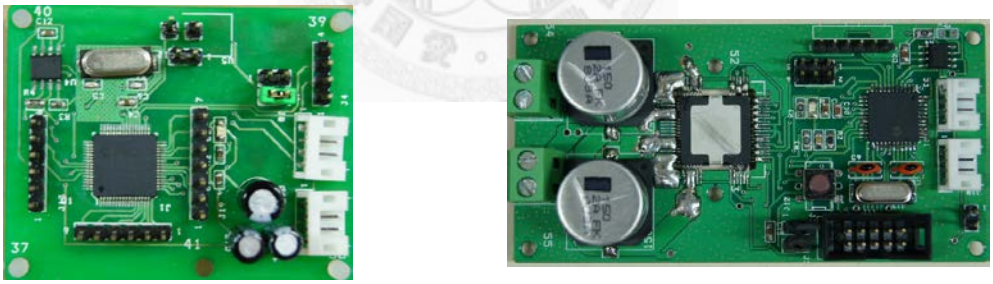


Figure 5-14 The USB-to-CAN Bus adapter module and the motor controller.

# 5.4 Summary

This chapter has described the construction of the humanoid robot mechanism with its new feet, trunk and waist, as well as the mechatronics system to control it. During the design process, we took the strength and weight into account, performing stress analysis and comparing materials. We did make some concessions on other

requirements like the movable range and mechanism weight, but our design still allows the robot to walk well.

Design and assembly are not discrete processes, and test often reveals a need for redesign. It is necessary to observe every characteristics, and ensure it is safe at every step. The new foot pad has functions other than toe-off and heel-contact motion, and will be used by colleagues for other applications.

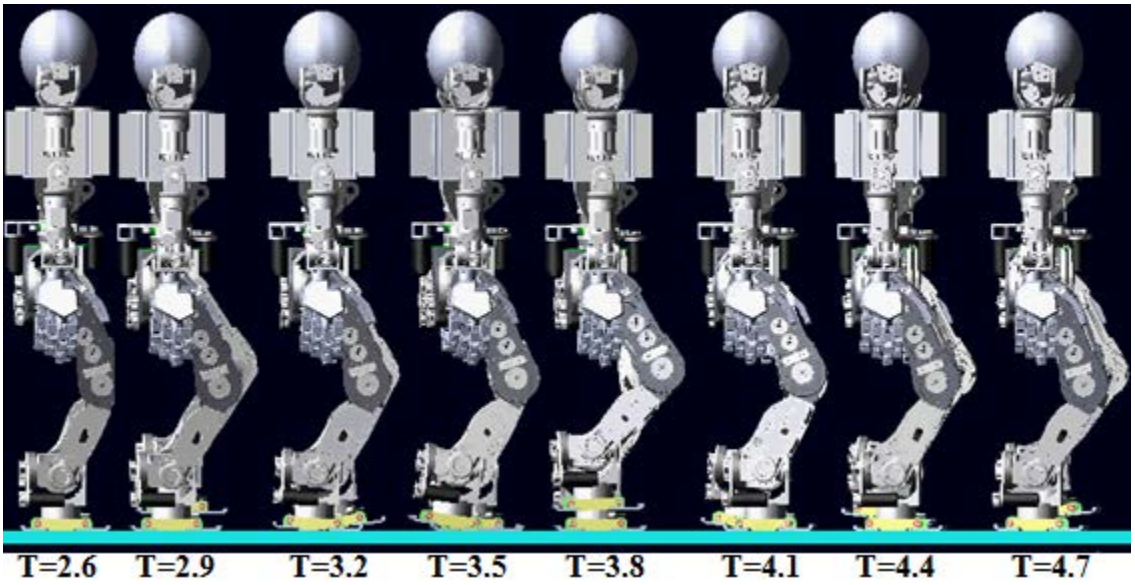



Figure 5-15 Simulation of new foot pads with toe-off and heel-contact motions.

# Chapter 6 Implementations and Experiments

## 6.1 Hardware Assembly

All mechanisms were fabricated, with due attention paid to assembly and wiring. During assembly, care was taken that moving components cannot foul other parts, and that joint friction allowed smooth movement. Figure 6-1 to Figure 6-4 show the complete humanoid robot, fully assembled. To reduce electromagnetic interference (EMI), it was necessary to separate digital wires from analog wires, and this helped us to install them in the legs in such a way that wiring problems are easy to trace—it is also easy to estimate the space available for wiring yet to be installed (Figure 6-4).

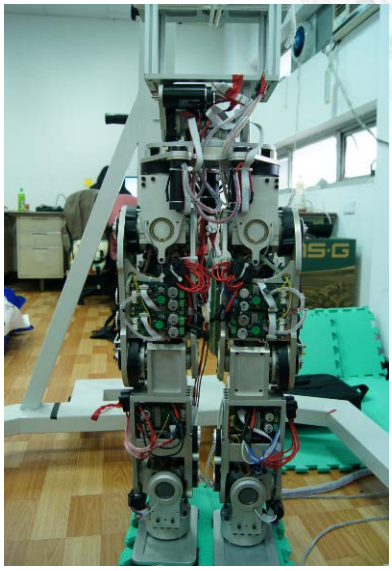


Figure 6-1 Front view of humanoid robot.

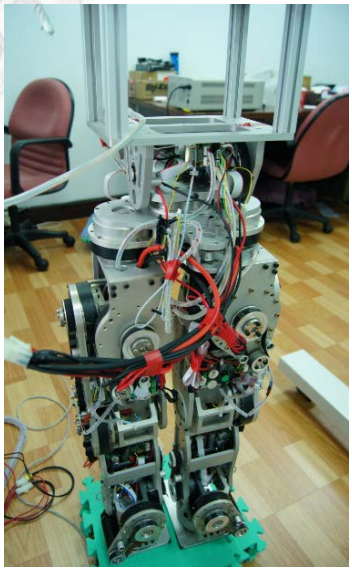


Figure 6-2 Back view of humanoid robot

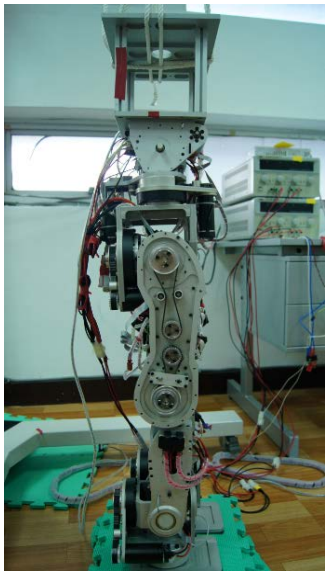


Figure 6-3 Left side view of humanoid robot

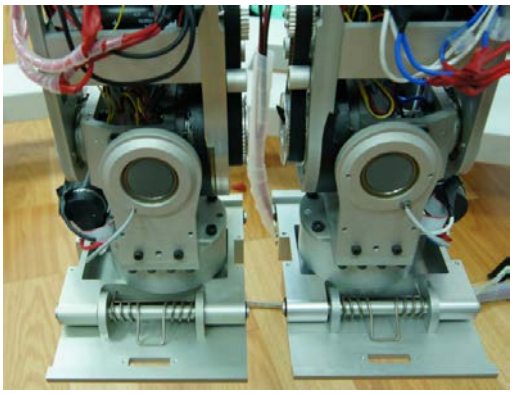


Figure 6-4 Humanoid robot with new foot pads.

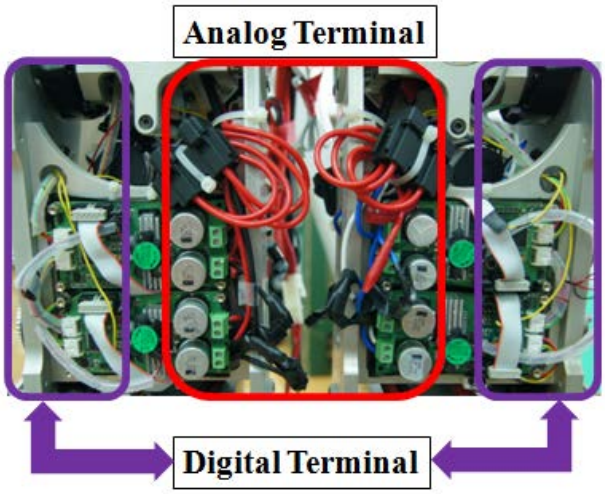



Figure 6-5 Wiring of humanoid robot

Our robot is fully assembled and wiring is complete, although the battery is still under test. The joint-resetting proximity sensors are in position and our robot can automatically return to its original posture.

# 6.2 Software and Firmware Setting

The software flowchart for hardware implementations and experiments is shown in Figure 6-6. Our program's user-friendly GUI (Graphical User Interface, Figure 6-7 )

development environment is Microsoft® Visual C++™ 8 under Microsoft® Windows™ 7.

This program provides fundamental commands, and the window shows the state of our robot, so that we can detect the success or failure of data transmission. The GUI and the firmware of the distributed system integrate those routine commands to a finite state machine, as sketched in Figure 6-8.

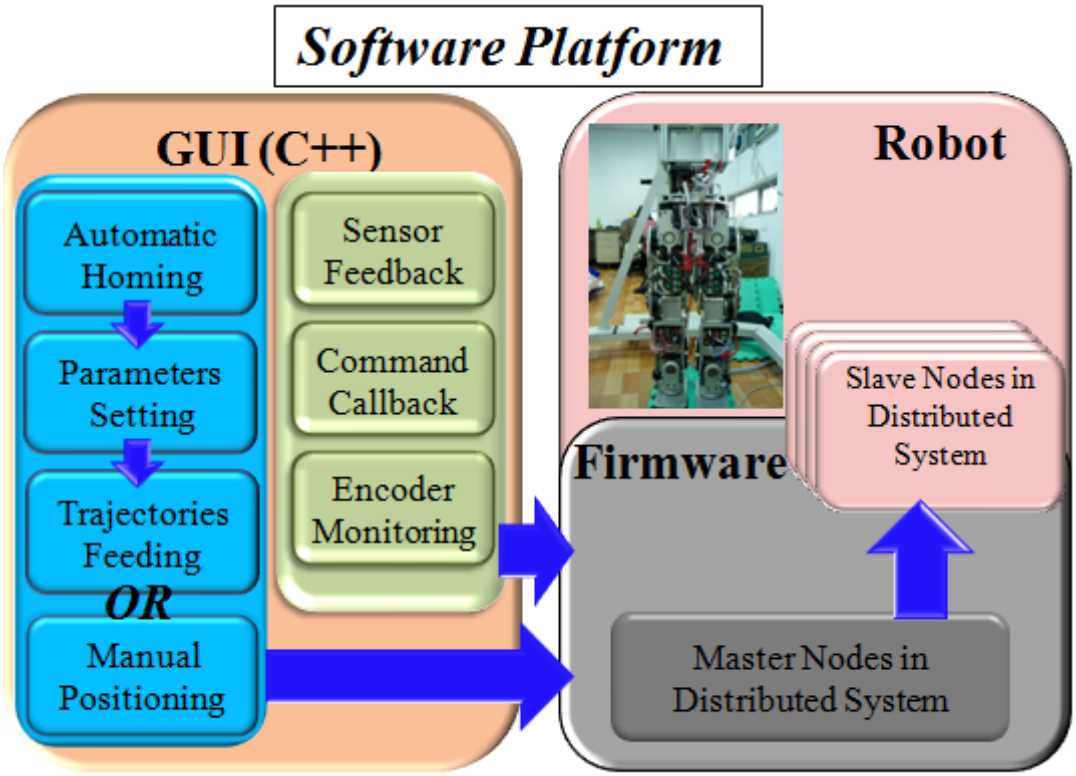



Figure 6-6 The relationship between software and hardware platforms

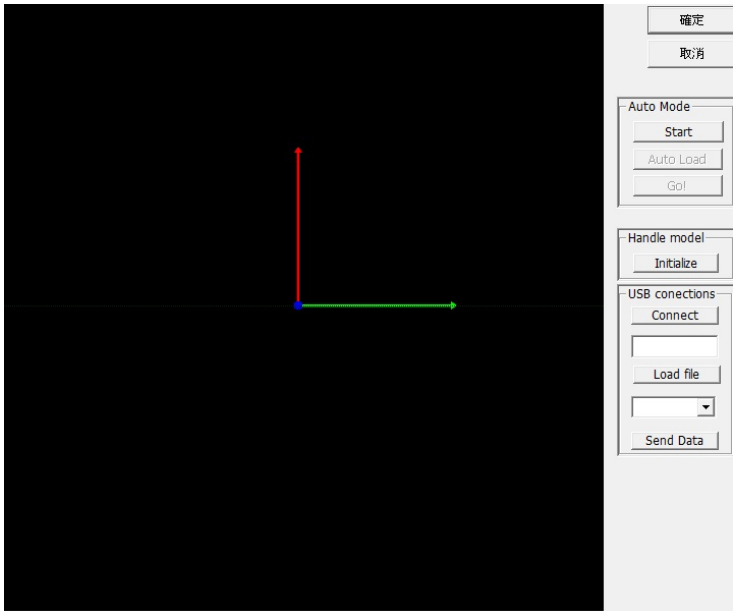



Figure 6-7 The GUI program for our robot system

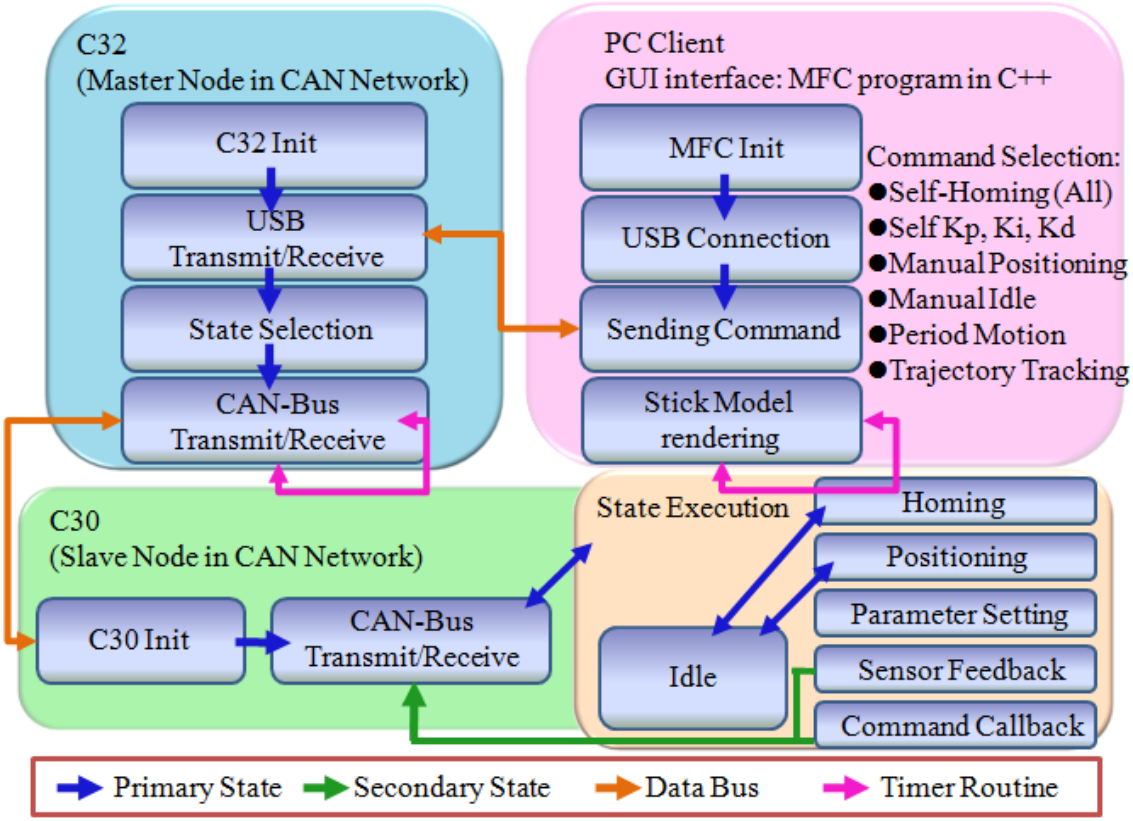



Figure 6-8 Proposed humanoid robot finite state machine for general operations

## 6.3 Experiments

In order to prove the proposed method, we took the joint trajectories into the robot system with adjusting ZMP trajectory which was introduced in Chapter 3. The result shows that the humanoid robot walks stably, as shown in Figure 6-9. The walking speed is 0.833 m/s and the step length is 1m.

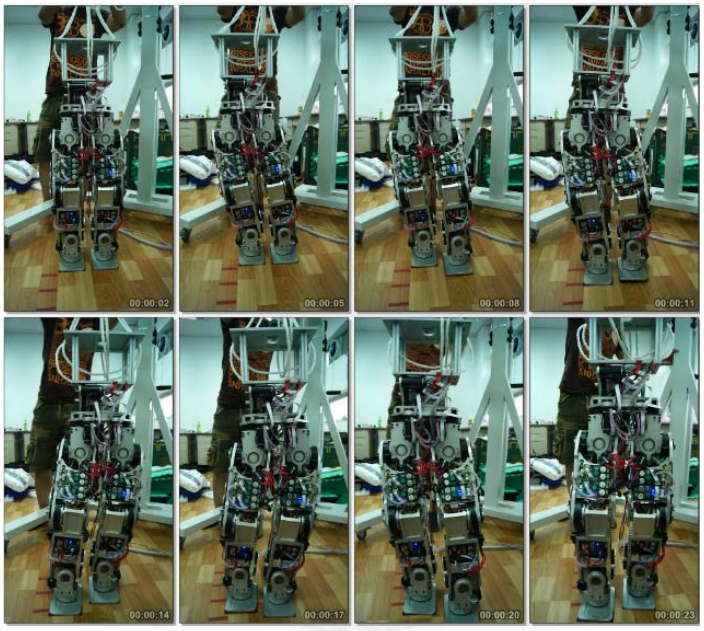



Figure 6-9 Humanoid robot, Voyager, walks with adjusting ZMP trajectory.

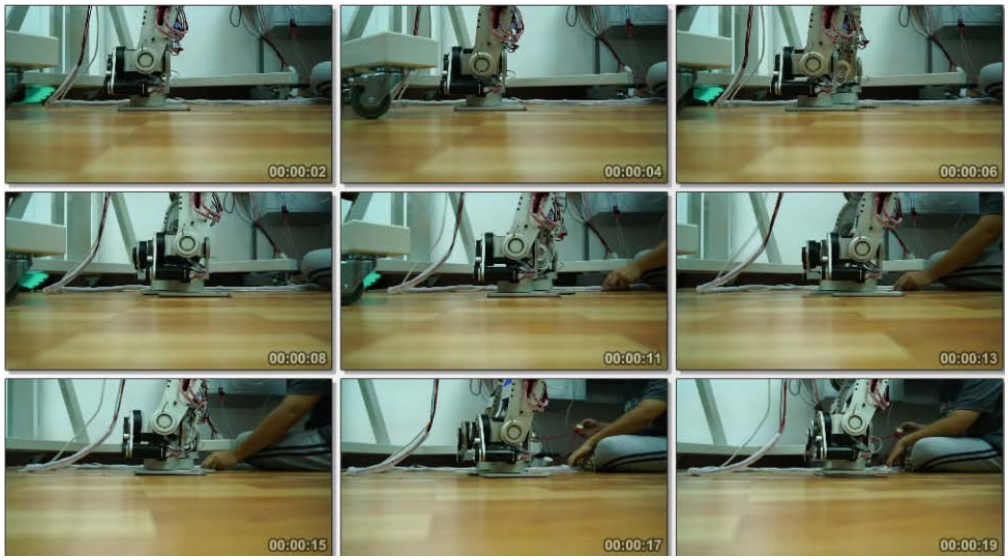



Figure 6-10 Close-up of flat foot walking.

With the smooth ZMP trajectory, our humanoid robot is able to walk stably. The contact area can be increased by humanoid robot with flat feet, and the stability is higher than humanoid robot with toe feet. The advantages of natural walk make us to overcome the stability issue. However, we cannot accurately make foot pads parallel to the ground. The point contact causes our humanoid robot to slip. Because there are two passive joints in the new foot pads, they can make foot pads become surface contact.

With new foot pads, the toe-off and heel-contact motions can be realized, as shown in Figure 6-11. In Figure 6-11, the stride length is 200mm. This walking motion is different from the one which do not adopt the toe and heel joints. To prove the practicability of toe and heel motions, the close-up can obviously observe the food pad motion, as shown in Figure 6-12.

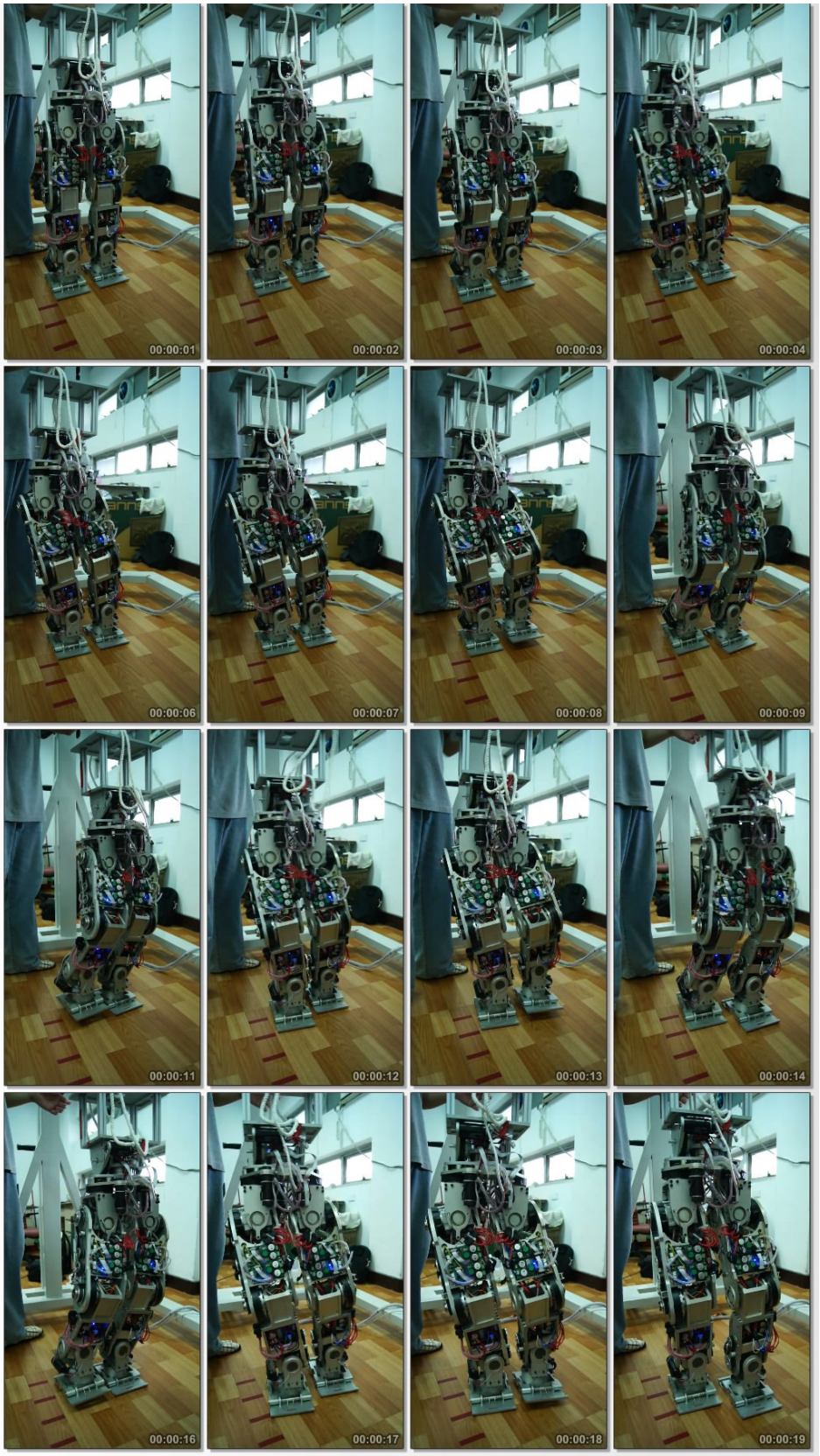



Figure 6-11 Humanoid robot, Voyager, walks with toe-off and heel-contact motions.

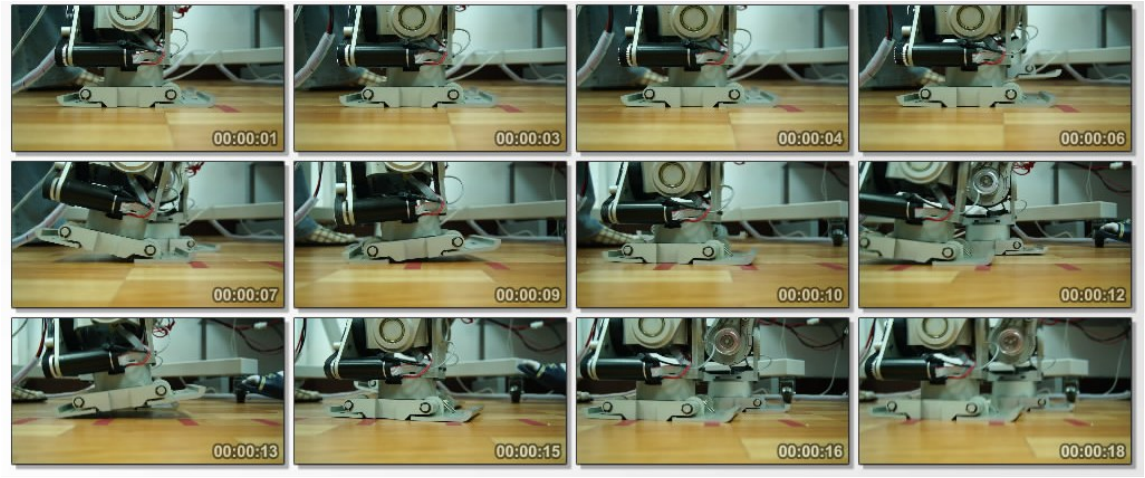



Figure 6-12 Close-up of toe-off and heel-contact motions in real humanoid robot.

The result shows our algorithms are feasible, and the new foot pads can really support our humanoid robot during walking cycle.

## 6.4 Summary

The experiments and results for natural walk are shown in this chapter. We used the CAN-BUS system to deliver the commands which are generated by our algorisms. The firmware setting helps us to control the motors, and the general operation is also proposed in our robot system.

About experiments, we realized the toe-off and the heel-contact motions in our humanoid robot. Our humanoid robot, Voyager, can walk with the new foot pads which have the passive toe and heel joints.

# Chapter 7 Conclusions and Future Works

## 7.1 Conclusions

We have constructed our humanoid robot, which can walk with toes and heels. We have also adjusted the ZMP trajectory and COG trajectory to match the natural walk. Application of momentum compensation has given the robot a better straight-line capability, and waist rotation allows it to mimic the walking motion of human beings. The main contribution of the thesis includes: (1) desired ZMP trajectory, (2) varying COG height, (3) natural walk, (4) mechanism design.

Since the conventional ZMP trajectory suits the flat foot, it will cause our humanoid robot to fall over easily as it gives rise to discontinuous velocity, acceleration, and jerks. Applying a high-order polynomial equation enabled us to optimize this trajectory. After considering various walking states, we separated the ZMP trajectory into two parts, single and double support. This method added great stability, as the simulation showed, because of the removal of irregular peaks in velocity, acceleration, and jerk. The same trajectory applies to natural walking.

By observing people's walking motion, we found that the COG trajectory constantly moves vertically, giving walkers the ability to make larger strides while

never reaching a singularity. We used the original ZMP equation to generate an available COG trajectory, and we can arbitrarily adjust the latter vertically with a preview control. Given a suitable trajectory, our robot no longer walks with constantly bent knees, enabling it to consume less power, as confirmed by ADAMS. The addition of momentum compensation to the system decreased the redundant moment, giving the robot better ability to walk in a straight line.

The main point of our study is to achieve a natural walk, and we propose two methods. In the first method, the toe joint of the supporting leg works actively. We altered the Jacobian matrix to control the 13 joints (12 leg joints and 1 toe joint on the standing leg). This method is proved to have some limitations, so we proposed the second method, “Toe Motion without Extra DOFs.” This method makes our humanoid robot straighten its knees at appropriate parts of the cycle, and reduces the power consumption. Having proposed a method of compensating the end-effector trajectory, we were able to introduce heel-contact motion, combine it with toe-off motion, and make the robot perform (in simulation) a natural walk.

To obtain practical proof of the proposed method, we designed a humanoid robot, “Voyager,” using stress analysis to assure the design. After taking power, strength, and weight into consideration, we have manufactured a robot that can perform the normal walking motion. For the mechatronic system, we adopted the CAN-Bus communicating

system, connected through USB to a personal computer so that commands can be sent to the robot. New foot pads were designed to add toes and heels, and these were also subjected to stress analysis to ensure their capability to perform a natural walk.

## 7.2 Future Works

This thesis has completed several fundamental parts on the humanoid locomotion. In the practical implementation, various issues and settlements need to be done. For example, although we optimized the ZMP trajectory for our humanoid robot, our methods are still offline method which cannot automatically generate natural walk motion online. How to measure the real ZMP is critical during natural walk. The 6 axis force/torque sensor can measure the force and torque from the foot pads. Therefore, the application of this sensor can help us to know where the ZMP locates. The robot can generate the toe-off and heel-contact motions by this information.

We hope our humanoid robot can automatically plan the walking trajectory. The online computation plays an important role for our future work. Since the computing speed in MATLAB is slower than Microsoft® Visual C++™, we must convert our programs to other platform. The natural walk also needs to plan its joint trajectories with online method.

# References


[1] S. Behnke, "Human-Like Walking using Toes Joint and Straight Stance Leg," *Proceeding of International Symposium on Adaptive Motion in Animals and Mechines*, Ilmenau, Germany, 2005.

[2] A. L. Betker, et al., "Estimation of 2-D Center of Mass Movement During Trunk Flexion-Extension Movements Using Body Accelerations," *Proceeding of IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 17, pp. 553-559, 2009.

[3] T. F. Chan and R. V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators," *Proceeding of IEEE Trans. On Robotics and Automation*, Vol. 11, No.2, pp. 286-292, April 1995.

[4] Y. W. Chao, "Mechatronic Design of a Biped Robt and Gait Coordination Control," *Master Thesis*, Department of Mechanical Engineering, National Taiwan University, 2010.

[5] T. H. Cheng, "Humanoid Robot Intelligent Walking Control," *Master Thesis*, Department of Mechanical Engineering, National Taiwan University, 2009.

[6] M. Fujita, "Digital Creatures for Future Entertainment Robotics," *Proceeding of IEEE International Conference on Robotics and Automation*, San Francisco, California, USA, pp. 801-806, April 2000.

[7] M. Gienger, K. Loffler and F. Pfeiffer, "Towards the Design of a Biped Jogging Robot," *Proceeding of IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 4140-4145, May 2001

[8] D. H. Gottlieb, "Robots and Topology," *Proceeding of IEEE International Conference on Robotics and Automation*, San Francisco, California, USA, pp. 1689-1691, April 1986.

[9] N. Handharu, J. Yoon, G. Kim, "Gait Pattern Generation with Knee Stretch Motion for Biped Robot Using Toe and Heel Joints," *Proceeding of IEEE/RAS International Conference on Humanoid Robots*, Daejeon, Korea, pp. 265-270, December, 2008.

[10] K. Harada, K. Miura, M. Morisawa, K. Kaneko, "Toward Human-Like Walking Pattern Generator," *Proceeding of IEEE/RSJ International Conference on in Intelligent Robots and Systems*, St. Louis, MO, USA, pp. 1071-1077, October 2009.

[11] M. Hayase and K. Ichikawa, "Optimal Servo system Utilizing Future Value of Desired Function," *Proceeding of Transactions of SICE*, Vol. 5, No.1, pp. 86-94,

February 1969.

[12] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The Development of Honda Humanoid Robot," *Proceeding of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, Vol. 2, pp. 1321-1326, May 1998.

[13] M. Hirose, Y. Haikawa, T. Takenaka, and, "Development of Humanoid Robot Asimo," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop2*, Vol. 13, pp. 1-6, October 2001.

[14] H. Chih-Lyang, H. C. Wu, N. W. Lu, M. L. Lin, C. H. Huang, T. C. Hsu, I C. Luo and H. J. Chou, "Realization of Humanoid Service Robotic System," *Proceeding of IEEE International Conference on Advanced Robotics and Its Social Impacts*, Taipei, Taiwan, pp. 1-6, August 2008.

[15] H. P. Huang and C. P. Liu, "A Novel Trajectory Optimization and Workspace Boundary Singularity Solution for Industrial Robots," *Proceeding of Automation the Eighth International Conference on Automation Technology Conferences*, Taichung, Taiwan, pp. 1-5, May 2005.

[16] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "Autobalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots," *Proceeding of the 4th International Workshop on Algorithmic Foundation of Robotics*, Vol. 12, pp. 105-118, January 2000.

[17] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue, "A Fast Dynamically Equilibrated Walking Trajectory Generation Method of Humanoid Robot," *Proceeding of Autonomous Robots*, Vol. 12, No.1, pp. 71-82, January 2002.

[18] S. Kajita, F. Kanehiro, M. Morisawa, S. Nakaoka and H. Hirukawa, "ZMP-Based Biped Running Enhanced by Toe Springs," *Proceeding of IEEE International Conference on Robotics and Automation*, Italy, pp. 3963-3969, April 2007.

[19] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point," *Proceeding of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Vol. 2, pp. 1620-1626, September 2003.

[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved Momentum Control: Humanoid Motion Planning Based on the Linear and Angular Momentum," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, United states, Vol. 2, pp. 1644-1650, October 2003.

[21] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A Realtime Pattern Generator for Biped Walking," *Proceeding of 2002 IEEE*

*International Conference on Robotics and Automation, May 11, 2002 - May 15, 2002*, Washington DC, United States, Vol. 1, pp. 31-37, 2002.

[22] S. Kajita, O. Matsumoto, and M. Saigo, "Real-Time 3d Walking Pattern Generation for a Biped Robot with Telescopic Legs," *Proceeding of IEEE International Conference on Robotics and Automation*, Seoul, Korea, Republic of, Vol. 3, pp. 2299-2306, May 2001.

[23] S. Kajita and K. Tani, "Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode," *Proceeding of IEEE International Conference on Robotics and Automation*, Nagoya, Jpn, Vol. 3, pp. 2885-2891, May 1995.

[24] N. Kanehira, T. U. Kawasaki, S. Ohta, T. Ismumi, T. Kawada, F. Kanehiro, S. Kajita, K. Kaneko, "Design and Experiments of Advanced Leg Module (HRP-2L) for Humanoid Robot (HRP-2) Development," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Vol.3, pp. 2455-2460, October 2002.

[25] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K Akachi, T. Kawasaki, S. Ota and T. Isozumi, "Design of Prototype Humanoid Robotics Platform for HRP," *Proceeding of IEEE/RSJ International Conference on Intellegrnt Robots and System*, Lausanne, Switzerland, Vol.3, pp. 2431-2436, October 2002.

[26] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukaka, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid Robot Hrp-2," *Proceeding of IEEE International Conference on Robotics and Automation*, New Orleans, LA, US, pp. 1083-1090, April 2004.

[27] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid Robot Hrp-3," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 2471-2478, September 2008.

[28] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. i. Nakaoka, and S. Kajita, "Cybernetic Human Hrp-4c," *Proceeding of IEEE-RAS International Conference on Humanoid Robots*, Paris, France, pp. 7-14, December 2009.

[29] T. Katayama, T. Ohki, T. Inoue, and T. Kato, "Design of an Optimal Controller for a Discrete-Time System Subject to Previewable Demand," *Proceeding of International Journal of Control*, Vol. 41, No.3, pp. 677-699, March 1985.

[30] H. Kobayashi and T. Sugihara, "Self-consistent Automatic Navigation of COM and Feet for Realtime Humanoid Robot Steering," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Mo, USA, pp. 3525-3530, October 2009.

[31] K. Konno, R. Sellaouti, F. B. Amar and F. B. Ouezdou, "Design and Development of the Biped Prototype ROBIAN," *Proceeding of IEEE International Conference on Robotics and Automation*, Lausanne, Switzerland,

pp.1384-1389, October 2002.

[32] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Online Footstep Planning for Humanoid Robots," *Proceeding of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Vol. 1, pp. 932-937, September 2003.

[33] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inaue, "Footstep Planning Among Obstacles for Biped Robots," *Proceeding of IEEE/RSJ ,International Conference Intelligent Robots and Systems*, Maui, USA. Vol.1, pp. 500-505, October 2001.

[34] Y. Kuroki, M. Fujita, T. Ishida, K. i. Nagasaka, and J. i. Yamaguchi, "A Small Biped Entertainment Robot Exploring Attractive Applications," *Proceeding of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Vol. 1, pp. 471-476, September 2003.

[35] A. Kyung-ho and O. Yonghwan, "Walking Control of a Humanoid Robot via Explicit and Stable CoM Manipulation with the Angular Momentum Resolution," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 2478-2483, October 2006.

[36] J.-Y. Lee, M.-S. Kim, and J.-J. Lee, "Multi-Objective Walking Trajectories Generation for a Biped Robot," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Vol. 4, pp. 3853-3858, September 2004.

[37] Y. Li, C. Li, and P. Chen, "Research and Design of Control System for a Tracked Sar Robot under Coal Mine," *Proceeding of IEEE International Conference on Automation and Logistics*, Shenyang, China, pp. 1957-1961, August 2009.

[38] Z. Li, N. G. Tsagarikis, D. G. Caldwell and B. Vanderborght, "Trajectory Generation of Straightened Knee Walking for Humanoid Robot iCub," *Proceeding of IEEE International Conference on Control, Automation, Robotics and Vision*, Singapore, pp. 2355-2360, March 2010.

[39] Z. Lingyan, L. Zhang, L. Wang, J. Wang, "Three-Dimensional Motion of The Pelvis During Human Walking," *Proceeding of IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Ontario, Canada, Vol. 1, pp. 335-339, July 2005.

[40] S. Lohmeier, T. Buschmann and H. Ulbrich, "Humanoid Robot LOLA," *Proceeding of IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 775-780, May 2009.

[41] K. Madani, C. Sabourin, "Multi-Level Cognitive Machine-Learning Based Concept for Human-Like “Artificaial” Walking: Application to Autonomous

Stroll of Humanoid Robots," *Neurocomputing*, Afghanistan, Vol. 74, pp. 1218-1223, October 2011.

[42] H. Minakata and Y. Hori, "Realtime Speed-Changeable Biped Walking by Controlling the Parameter of Virtual Inverted Pendulum," *Proceeding of the 20th International Conference on Industrial Electronics, Control and Instrumentation*, Bologna, Italy, Vol. 2, pp. 1009-1014, September 1994.

[43] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," *Proceeding of Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 108, No.3, pp. 163-171, September 1986.

[44] K. D. Nguyen, I. M. Chen, Z. Luo, S. H. Yeo, and H. B.-L. Duh, "A Body Sensor Network for Tracking and Monitoring of Functional Arm Motion," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, United states, pp. 3862-3867, October 2009.

[45] K. Nishiwaki and S. Kagami, "Strategies for Adjusting The ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking," *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, pp. 4230-4236, May 2010.

[46] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba and H. Inoue, "Toe Joints that Enhance Bipedal and Fullbody Motion of Humanoid Robots," *Proceeding of IEEE International Conference on Robotics and Automation*, Washington, USA, pp. 3105-3110, May 2002.

[47] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue, "Design and Development of Research Platform for Perception-Action Integration in Humanoid Robot: H6," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, Vol. 3, pp. 1559-1564, October 2000.

[48] K. Nishiwaki and S. Kagami, "Online Walking Control System for Humanoids with Short Cycle Pattern Generation," *Proceeding of International Journal of Robotics Research*, Vol. 28, No.6, pp. 729-742, June 2009.

[49] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. O. Lim and A. Takanishi, "Development of a New Humanoid Robot WABIAN-2," *Proceeding of IEEE International Conference on Robotics and Automation*, Beijing, China, pp. 76-81, October 2006.

[50] Y. Ogura, K. Shimomura, H. Kondo, A. Morishima, T. Okubo, S. Momoki, H. O. Lim and A. Takanishi, "Human-Like Walking with Knee Stretched, Heel-contact and Toe-off Motion by a Humanoid Robot," *Proceeding of IEEE International Workshop on Intellegent Robots and Systems*, pp. 3976-3981,

October 2006.

[51] C. Ott, C. Baumgartner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schaffer, M. Grebenstein and G. Hirzinger, "Development of a Biped Robot With Torque Controlled Joints," *Proceeding of IEEE International Conference on Humanoid Robots*, pp. 167-173, 2010.

[52] F. B. Ouezdou, S. Alfayad and B. Almasri, "Comparison of Several Kinds of Feet for Humanoid Robot," *Proceeding of IEEE International Conference on Humanoid Robots*, Tsukuba, Japan, pp. 123-128, Decrmber 2005.

[53] V. D. P. J. From, J. T. Gravdahl, S. Sastry, "Modeling and Motion Planning for Mechanisms on a Non-Inertial Base," *Proceeding of IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 3320-3326, May 2009.

[54] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh, "Mechanical Design of the Humanoid Robot Platform, Hubo," *Proceeding of Advanced Robotics*, Vol. 21, No.11, pp. 1305-1322, November 2007.

[55] J. H. Park and K. D. Kim, "Biped Robot Walking Using Gravity-Compensated Inverted Pendulum Mode and Computed Torque Control," *Proceeding of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, Vol. 4, pp. 3528-3533, May 1998.

[56] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *Proceeding of IEEE-RAS International Conference on Humanoid Robots*, Genoa, Italy, pp. 200-207, December 2006.

[57] Y. Sakagami, R. Watanable, C. Aoyama, S, Matsunaga, N. Higaki and K. Fujimura, "The Intelligent ASIMO: System Overview and Integration," *Proceeding of IEEE International Workshop on Intellegent Robots & Systems*, Lausannem Switzerland, pp. 2478-2483, September 2002.

[58] S. Sakka, C. Hayot and P. Lacouture, "A Generalized 3D Inverted Pendulum Model to Represent Human Normal Walking," *Proceeding of IEEE International on Humanoid Robots*, Nashville, Tennessee, USA, pp. 486-491, December 2010.

[59] A. Sano and J. Furusho, "Realization of Natural Dynamic Walking Using the Angular Momentum Information," *Proceeding of IEEE-RAS International Conference on Robotics and Automation*, Tsukuba, Japan, Vol.3, pp. 1476-1481, May 1990.

[60] T. Sato, S. Sakaino and Kouhei Ohnishi, "Trajectory Planning and Control for Biped Robot with Toe and Heel Joints," *Proceeding of IEEE International Workshop on Advanced Motion Control*, Niigata, Japan, pp. 129-136, March 2010

[61] R. Sellaouti, O. Stasse, S. Kajita, K. Yokoi and A. Kheddar, "Faster and Smoother Walking of Humanoid HRP-2 with Passive Toe Joints," *Proceeding of IEEE International Workshop on Intelligent Robots and Systems*, Beijing, China, pp. 4909-4914, October 2006.

[62] T. B. Sheridan, "Three Models of Preview Control," *Proceeding of IEEE Transactions on Human Factors in Electronics*, Vol. 7, No.2, pp. 91-102, June 1966.

[63] S. Shimmyo and K. Ohnishi, "Nested Preview Control by Utilizing Virtual Plane for Biped Walking Pattern Generation Including COG Up-Down Motion," *Proceeding of IEEE Industrial Electronics Society*, Valencia, Spain, pp. 1571-1576, November 2010.

[64] A. Takanishi, H. Lim, M. Tsuda, and I. Kato, "Realization of Dynamic Biped Walking Stabilized by Trunk Motion on a Sagittally Uneven Surface," *Proceeding of IEEE International Workshop on Intelligent Robots and Systems*, Ibaraki, Japan, pp. 323-330, July 1990.

[65] T. Takahashi and A. Kawamura, "Posture Control Using Foot Toe and Sole for Biped Robot "Ken"," *Proceeding of IEEE International Workshop on Advanced Motion Control*, Slovenia, pp. 437-442, July, 2002.

[66] D. Tlalolini, C. Chevallereau, Y. Aoustin, "Human-Like Walking: Optimal Motion of a Bipedal Robot With Toe-Rotation Motion," *Proceeding of IEEE/ASME Transactions on Mechatronics*, Vol. 16, pp. 310-320, 2011.

[67] M. Tomizuka and D. E. Rosenthal, "On the Optimal Digital State Vector Feedback Controller with Integral and Preview Actions," *Proceeding of Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 101, No.3, pp. 172-178, January 1979.

[68] L. W. Tsai, ed. "ROBOT ANALYSIS: The Mechanics of Serial and Parallel Manipulators," *Proceeding of WILEY-INTERSCIENCE*, Taiwan, pp. 31-115, 2005.

[69] Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *Proceeding of IEEE Transactions on Robotics and Automation*, Vol. 5, No.3, pp. 303-314, June 1989.

[70] M. Vukobratovic and B. Borovac, "Zero-moment point— Thirty Five Years of Its Life," *International Journal of Humanoid Robotics*, Vol. 1, No. 1, pp. 157-173, 2004.

[71] L. Wang, Z. Yu, Q. Meng and Z. Zhang, "Influence Analysis of Toe-joint on Biped Gaits," *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, Henan, China, pp. 1631-1635, June 2006.

[72] M. Xie, "Fundamentals of Robotics: Linking Perception to Action," *World Scientific, Series in Machine Perception Artificial Intelligence*, Vol. 54, 2003.

[73] K. Yamamoto, T. Sugihara and Y. Nakamura, "Toe Joint Mechanism Using Parallel Four-bar Linkage Enabling Humanlike Multiple Support at Toe Pad and Toe Tip," *Proceeding of IEEE/RAS on Humanoids Robots*, USA, pp. 410-415, November 2007.

[74] B. Yamauchi, "PackBot: A Versatile Platform for Military Robotics," *Proceeding of SPIE Unmanned Ground Vechicle Technology*, Vol. 5422, April 2004.

[75] J. L. Yan and H. P. Huang, "A Fast and Smooth Walking Pattern Generator of Biped Robot Using Jacobian Inverse Kinematics," *Proceeding of IEEE Workshop on Advanced Robotics and Its Social Impacts*, Piscataway, NJ, USA, pp. 25-30, December 2007.

[76] J. Yang, T. Zhang, J. Song, H. Sun, G. Shi, and Y. Chen, "Redundant Design of a Can Bus Testing and Communication System for Space Robot Arm," *Proceeding of International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, pp. 1894-1898, December 2008.

[77] B.-J. You, M. Hwangbo, S.-O. Lee, S.-R. Oh, Y. D. Kwon, and S. Lim, "Development of a Home Service Robot 'Issac'," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, Vol. 3, pp. 2630-2635, October 2003.

[78] C. Youngjin, D. Kim and B. J. You, "On the Walking Control for Humanoid Robot Based on the Kinematic Resolution of CoM Jacobian with Embedded Motion," *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, pp. 2655-2660, 2006.

[79] S. W. Yu, "Walking Pattern Analysis and Control of a Humanoid Robot," *Master Thesis*, Department of Mechanical Engineering, National Taiwan University, 2006.

[80] Z. Yu, Q. Huang, J. Li, Q. Shi, X. Chen, and K. Li, "Distributed Control System for a Humanoid Robot," *Proceeding of IEEE International Conference on Mechatronics and Automation*, Harbin, China, pp. 1166-1171, August 2007.

[81] T. Zhe and E. Meng Joo, "Humanoid 3D Gait Generation Based on Inverted Pendulum Model," *Proceeding of IEEE International Symposium on Intelligent Control*, pp. 339-344, October 2007.

[82] BostonDynamics, 2009, " Petman - Bigdog Gets a Big Brother", Available: <http://www.bostondynamics.com/robot_petman.html>

[83] Healthy Living Tips, March 2011, " Walking Diet", Available: <http://befitnes.blogspot.com/2011/05/walking-diet.html>

[84] IGN, March 2010, , Available: <http://movies.ign.com/articles/107/1077314p1.html>

[85] io9, January, 2008, Available: <http://io9.com/346124/will-we-hold-robots-accountable-for-war-crimes>

[86] Mail Online, November 2009, Available: <http://www.dailymail.co.uk/sport/football/article-1225377/David-Beckhams-AC-Milan-deal-relief-Los-Angeles-Galaxy-coach-Bruce-Arena.html>

[87] Waseda University, 2005, " Biped huamnoid robot : WABIAN-2R", Available: <http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/index.htm>

[88] Wikipedia, October 2007, " Gait (Human)", Available: <http://en.wikipedia.org/wiki/Gait_(human)>