Palm-sized Humanoid Robot

CHUNG, Wing Kwong

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Automation and Computer-Aided Engineering

September 2008

Thesis/Assessment Committee

Professor Hui, Kinchuen (Chair)

Professor Xu, Yangsheng (Thesis Supervisor)

Professor Wang, Changling Charlie (Committee Member)

Professor Zhang Jianwei (External Examiner)

Abstract

Recently, the development of humanoid robots with a certain level of intelligence becomes a popular research topic. The size of entertainment based humanoid robots are usually over 30cm and 1kg which are quite large and heavy. To reduce the size, extra constraints are introduced in hardware selection, mechanical design, motion planning, and even human-robot interaction, etc. In this thesis, we present a new entertainment based humanoid robot, PHR, which is palm-sized with 20 degree of freedom. To increase its portability, it is designed to be foldable.

In our research, we focus on the detail investigations in mechanical design, gait synthesis, gait planning, and balance control. Being an entertainment based humanoid robot, the interaction between human and PHR is also considered. In the thesis, we firstly describe the design of our palm-sized humanoid robot. Its actuation is based on a mini-sized servo motor. A special joint configuration for the folding operation is then introduced.

Moreover, we solve the the 3-dimensional inverse kinematics of PHR which is a key technique for gait synthesis, planning, and balance control. By determining the mapping from Cartesian space to robot joint space, the gait pattern of PHR is generated using the concept of gait frames. For gait planning, the approach of Bezier curve interpolation in gait frames is utilized. By adjusting the control points in between the foot positions given by two consecutive gait frames, the gait trajectory can then be manipulated. For the balance control, a geometric method along with the application of 3-D inverse kinematics is introduced.

For the human-robot interaction, a hand gestures recognition algorithm based on Haar wavelet representation is presented. It performs real-time recognition with high accuracy.

In the algorithm, we standardize the hand orientation by using the concept of principal axis (axis of elongation) which gives a benefit in highly reducing the database size. We then develop a codeword approach to facilitate the searching process. A fast and efficient measurement metric is finally proposed for the gestures recognition.

Finally, the experimental results of those algorithms are presented and evaluated. The thesis then sums up the contributions achieved and proposes some future works.

論文摘要

近年來，智能化人形機器人的發展成爲了機械人研究中的一個重要項目。然而，傳統娛樂用機器人的身高以及體重都分別超過 30cm 和 1kg，這較大而且重的軀體對實際應用造成了很多不便。如果把機器人的外型和體重減小，便會導致一些研發上的困難，例如硬件採用、機械設計、運動規劃甚至人機交互等等。這篇論文致力於建立一個手掌大小、具有 20 個自由度而且可變形的人形機器人。

在這個項目，我們主要集中在機械設計、步履編制、步履規劃、平衡系統以及人機交互五個方面的研究。在這篇論文，我們首先介紹機器人的設計，包括它的動力以及變形系統的關節設置。

緊接著，我們解開了機器人的三維反向關節運動，並且利用反向關節運動的結果，加上步履畫面的概念，完成了機器人的步履編制。在步履規劃方面，我們在步履畫面的基礎上運用了貝茲曲線的理論，通過貝茲曲線遞補演算法和控制點的設置實現了機器人的步履規劃。另外，我們結合了三維反向關節運動和幾何方法，實踐了機器人的的平衡系統。

在人機交互方面，論文提出以哈爾小波概念爲基礎，建立一個非穿戴式的手勢識別系統，對使用者的手勢作出實時且準確的分辨。我們利用物體的主軸，對手勢的方向作出自動的調整，從而大大減少資料庫的內容。爲了提高尋找資料的效率，我們對手勢進行編碼。此外我們提出一個新的尺度對手勢進行辨識。

最後，作者利用實驗對以上算法作出驗證。論文結尾總結了作者的工作並給出了對未來工作的展望。

Acknowledgments

I would like to express my deepest and sincerest gratitude to many special people who assisted and encouraged me to complete this thesis.

First, a special acknowledgement should be shown to my supervisor Prof. Xu Yangsheng, for his friendship, constant supervision, advices and great patience on my research throughout the years. He provided me a lot of opportunities of training, learning and exploring new things in the Advanced Robotics Lab. With his guidance, I have learnt many efficient research skills and methodologies. I am deeply impressed by his kindly care in developing my personal skills. I really feel proud to be one of his students which is truly an invaluable experience. In addition, I would extend my sincere thanks to Prof. Hui Kin-Chuen and Prof. Wang Changling Charlie from our department, and Prof. Zhang Jianwei from the University of Hamburg to serve as the members of my research committee with their professional comments and judgments.

I would like to take this chance to thank Xinyu Wu for his help and support in my research. My gratitude also goes to Mr. Hang Tong, Bufu Huang, Xi Shi, and Thomas Lau for their technical supports on my work. I am in debt to Wai Kit Mok for his help in building the prototype of plam-size humanoid robot. My heartfelt thanks are also due to the joyful experiences with my friends in the Advanced Robotics Laboratory, including Dr. Ka Keung Lee, Ms. Meng Chen, Mr. Huihuan Qian, Mr. Weizhong Ye, MR Zhi Zhong and so on. I will never forget the experiences and friendships that we built throughout the years.

Last but not least, with my love and respect, I thank the love, support and encouragement from my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Humanoid robot is one of the intelligent robots in the world. It is designed to work and cooperate with human beings. It is also designed to be entertaining. For this kind of robots, they are usually expected to play with us whenever we want. For instance, we could play with our robots when we are waiting for flights in the airport. The KHR-series humanoid robots developed by Kondo are typical examples for entertainment based robots. They are constructed by servo motors with a similar joint configuration as humans. Another example is Robonova developed by Hitec. In fact, the height of those humanoid robots is usually above 30 cm with weights over 1kg which are obviously not portable.

In this thesis, we seek to develop a palm-sized humanoid robot (PHR) which is much smaller and lighter than traditional humanoid robots. Because of the size, extra difficulties are introduced in hardware selection, mechanical design, motion planning, and human-robot interaction, etc. We focus our research on the detail investigation in the following challenge problems: 1) to design the mechanical parts of PHR with the capability to fold into a box structure; 2) to generate gait patterns of PHR on a flat ground; 3) to plan the gait in order to achieve some specific effects while walking; 4) to develop a balance system which is feasible in correcting the tilting of PHR; and 5) to develop a human-robot interaction system based on hand gestures recognition.

During the mechanical design of PHR, we propose a special joint configuration for PHR to perform box transformation. Also, it is actuated by mini-sized servo motors which gives a benefit in highly reducing the size.

For its gait motion, we first introduce the robot constraints in finding the mapping between the Cartesian coordinate and the joint coordinate (Forward and Inverse Kinematics). We then describe a Bezier curve interpolation approach for gait planning. To keep the robot in balance, a geometric approach balance algorithm is implemented to amend the tilting of PHR.

For the interaction between humans and PHR, we focused on the use of hand gestures as a communication link. It is a computer vision based approach without using any sensory gloves. A glove-free hand recognition system can increase the flexibility of interaction since there is no needs for a user to bring a glove together with the robot. Although there are many other robot interaction methodologies like voice and RF remote, we propose to use hand gestures since it is not easily affected by the surrounding noises. For instance, public areas like airport is usually noisy which makes commands recognition through voice becoming not feasible. In fact, the light condition in those places is always even and suitable for many vision based systems.

By this project, we hope to develop a new generation of humanoid robot. This robot has the potential to be a toy of humans because of different reasons. Firstly, it is portable because of the size. In places such as train, airports, and ships, a user can play with it while boring. It can then be automatically folded and easily put back to a bag. Secondly, 3-D inverse kinematics of PHR has been determined, so we can plan different motions such as dancing, kicking and turning of PHR directly by considering the motion paths in Cartesian space. The stability of motions is also maintained by applying the proposed balance algorithm. Thirdly, the proposed hand gestures recognition system acts as a direct and easy communication link between PHR and users. It can also be further implemented to translate sign languages since the hand gestures chosen in this project are from some traditional sign languages.

1.2 Related Work

In the last section, the background and motivation of this thesis are presented. Since our works focus on the development of a humanoid robot. Related literatures are therefore studied in this section.

1.2.1 History of Humanoid Robots

The development of humanoid robots has been started for more than 30 years. The first full-scale anthropomorphic robot, WABOT-1 was built by the Waseda University in 1973 [30]. It consists of 3 major systems. They are i) limb control ; 2) vision system; and 3) conversation system. It is able to communicate with people in Japanese and measure the position of objects using external sensors. It can also walk, grip and transport objects.

Twelve years later, the second version, Wabot-2 is a robot musician and brings out the idea in developing a "personal robot". It can communicate with a person, read a musical score, play tunes on an electronic organ. It is also able to accompany a person by listening to the person singing. The humanoid robots, Wabot-1 and Wabot-2 are shown in Fig. 1.1.

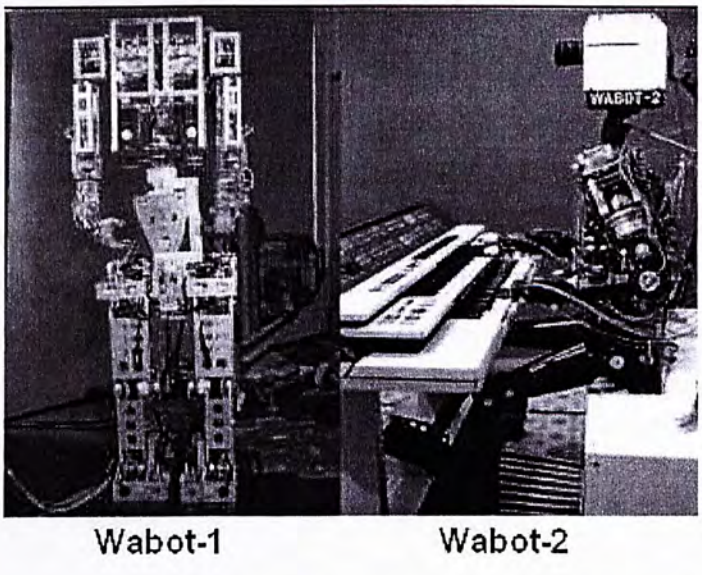


Figure 1.1: The humanoid robots, Wabot-1 and Wabot-2

Subsequently, Honda introduced their "Human" robot, P2 in Tokyo during 1996. It is 180 cm tall and weighs 210 kg which can operate completely independently for 15 minutes. It is the first humanoid robot that performs stable walking with its legs. A year later, a new version, P3 is developed by Honda which is 160 cm tall and weighs 130 kg. Another smaller humanoid robot, ASIMO was introduced by Honda in 2000. It is 120 cm tall and weighs only 43 kg. Fig. 1.2 shows the robots, P1, P2, P3 and ASIMO.

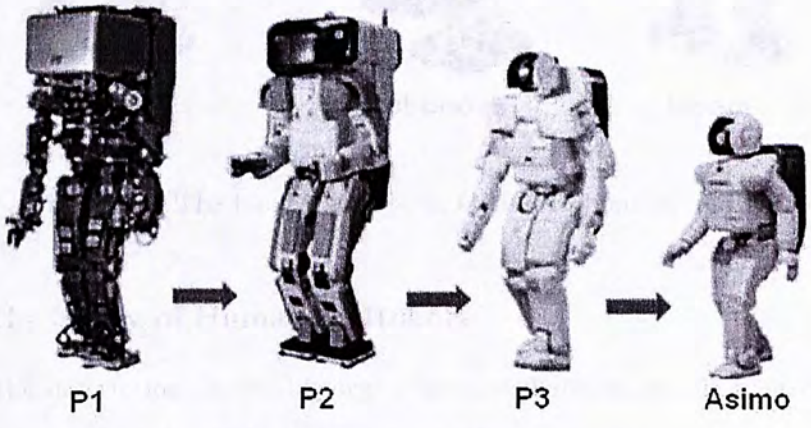


Figure 1.2: The humanoid robots, P1, P2, P3, and ASIMO

After the breakthrough made by Honda, the development of humanoid robots becomes more popular in the world. Different companies such as Sony, Hitec and Hondo, etc started their robotic groups for the development of humanoid robots. Many humanoid robots such as QIRO (Quest for cuRIOsity), Robonova, and Isobot are then introduced during the last fews years. Fig. 1.3 shows the robots, QRIO, Robonova, and Isobot. With the rapid development of Micro-Electro-Mechanical Systems (MEMS) technologies, they are expected to be smaller and lighter in the future.

Figure 1.3: The humanoid robots, QRIO, Robonova, and Isobot

1.2.2 The Study of Humanoid Robots

After the description on the history of humanoid robots, we first introduce the objectives of researchers in studying humanoid robots. By this, we then conclude some fundamental points which represent the directions in developing our palm-sized humanoid robot, PHR.

First of all, the study of humanoid robots is to perform human tasks. Their role therefore becomes a personal assistance. They are designed not only to help humans in doing some dirty and dangerous tasks, but also to take care sick and elderly people. The humanoid robot, HRP-2 is a typical example as a robotic platform for the Humanoid Robotics Project headed by the Manufacturing Science and Technology Center [32]. Its height and weight are 1.54 m and 58 kg respectively. It possesses 30 degree of freedom (DOF) and a vision system. Through the embedded software system, it can perform some kitchen helping tasks [47].

In addition, the study of humanoid robots enhances the understanding of human body structures. In this aspect, a humanoid robot can be regarded as an experimental platform for researchers to simulate different human motions. Through the simulation, they can model the relationship between motions and the structures of a human body. One of the

most popular researches for humanoid robots is gait synthesis [28]. It is to generate a natural gait for humanoid robots based on the analysis of human walking patterns [29].

Moreover, humanoid robots can be developed to provide entertainments such as QRIO. It is about 0.6 m tall and weighted 7.3 kg. It can perform stable dynamic walking, running, throwing a ball, ribbon dance and even fan dance [34]. It also has embedded voice and face recognition systems making it capable of remembering people [21].

From the above studies, it can be concluded that the development of a humanoid robot requires three basic elements. They are a) mechanical design, b) motion planning and control, and c) human-robot interaction. For the mechanical design, it aims to construct a joint configuration that is similar to that of humans. Motion planning allows the robot to perform a task. Balance algorithms are needed to maintain its stability and reduce the risk of falling down during motions. The study in human-robot interactions provides a natural and effective way for the communication between humans and robots. In this thesis, the development of PHR will therefore focus on the detail investigations in those areas.

1.3 Thesis Overview

Chapter 2 introduces the mechanical design of PHR. It first describes the proposed joint configuration of PHR. The transformation motion of PHR is then followed. The hardware and software platforms are finally reported.

Chapter 3 presents the kinematics of PHR. Forward kinematics of PHR is first described. Inverse kinematics of PHR is then followed. The main difference between our method and traditional approaches is that we are considering the 3-D case based on geometrical analysis. Aforementioned, the motion of a robot is usually 3-D and it is necessary to determine the mapping between the Cartesian space and the robot joint space. It also enhances the flexibility of motion planning in PHR. In this project, a geometric approach is introduced to solve the kinematics of the PHR. Instead of using other approaches such as jacobian matrix, geometric approach can give a clearly understanding about the structure of PHR. In other words, the mapping solution is computed through the direct analysis of

the physical structure (joints relationship of PHR).

Chapter 4 discusses the concepts utilized in gait planning for PHR. It is a gait frames based approach together with the use of curve interpolation. Five major motion frames are first defined as the key frames in a complete gait cycle. After that, the inter-gait frames are determined using Bezier-curve interpolation which provides the flexibility in gait planning by control points manipulation. The corresponding joint angles in each of the frames are finally computed using the inverse kinematics.

Aforementioned, a humanoid robot is usually suffered from problem in maintaining its balance, Chapter 5 describes a balance algorithm used in PHR. It is a geometric approach with the use of 3-D inverse kinematics to compute the tilting-compensated action. Given a tilting on PHR, the corresponding shifting in lateral or longitudinal direction is first computed using geometric analysis. After that, the extent in tilting is then used to update the Cartesian coordinates of corresponding joints. Tilting-compensated action is finally determined using the inverse kinematics.

In Chapter 6, we propose a human-robot interaction system through hand gestures. It consists of three novelties. The hand orientation in 2-D is first standardized by utilizing the concept of axis of elongation. It highly reduces the database size since only a few images are needed to form the database. We also develop a codeword approach to facilitate the searching process. For the gestures recognition, a fast and efficient measurement metric is also proposed .

In Chapter 7, we first give a conclusion on this project. After that, we suggest some meaningful future developments on PHR.

Chapter 2

Architecture

2.1 Introduction

It is mentioned that the objective of this project is to develop a palm-sized humanoid robot, PHR which consists three unique features when compared with existing robots. They are 1) mini-sized; 2) light; and 3) foldable. In the following parts of this section, we first intend to present the mechanical design of PHR. Then we will describe the hardware platform utilized in PHR. After that, we will present the software platform.

2.2 Mechanical Design

The left part in Fig. 2.1 shows the proposed humanoid robot, PHR which possesses 20 degree of freedom (DOF). Its height and weight are 19 cm and 550 g respectively, which are almost half size of many existing humanoid robots. The right part in Fig. 2.1 gives a comparison of size between PHR and other humanoid robots, KHR-1 and Robonova. Their joint configurations are shown in Fig. 2.2.

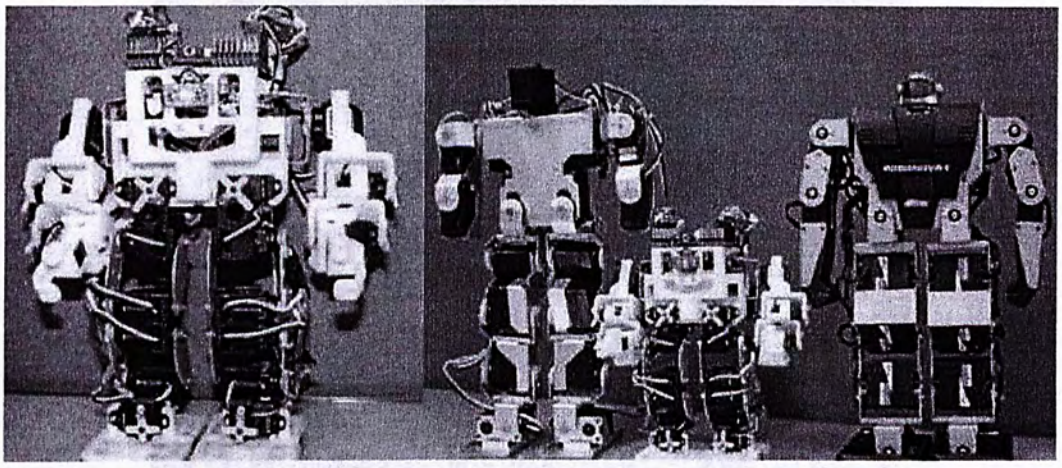


Figure 2.1: The proposed palm-sized humanoid robot, PHR

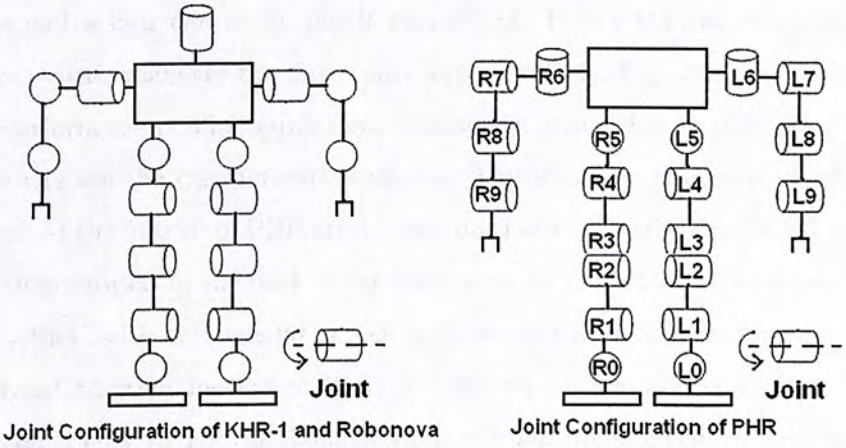


Figure 2.2: The comparison of joint configurations with PHR

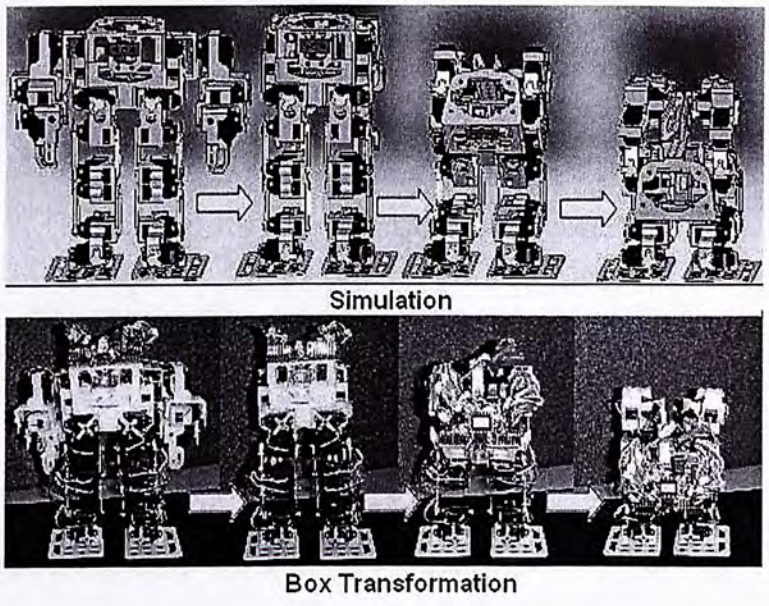


Figure 2.3: Procedures of box transformation for PHR

In addition to the reduction in size, we design and develop a mechanism of box folding to further improve the robot portability. Through this, it is capable to transform into a box shape and a user can easily put it into a bag. It will transform itself back to the human-like status whenever the user wants to play with it. Fig. 2.3 shows the procedures for the transformation. The upper figures show the simulation results using SolidWorks and the others are the transformation results. The whole process consists of three main procedures. At the first step, PHR straightens both arms as shown in the left most figures. They are then rotated to the back of the body with an angle 90^o. After that, PHR bends its body forward with an angle 90^o which is shown as the right most figures.

The transformation motion of PHR is achieved by two special designs in its joint configuration (Fig. 2.4). For the shoulder joints (L6 and R6 in red color), they are vertically oriented which allows the arms to be flipped backward. Another design is the double actuation at each of the knee joints (L2, L3, R2, and R3 in blue color) which gives two major benefits in PHR. Since the motion range of a traditional servo motor is about 140 degrees, double actuation provides a larger range of motion and allows the robot body

bending forward to a larger extent during the transformation. Also, better supporting power can be provided.

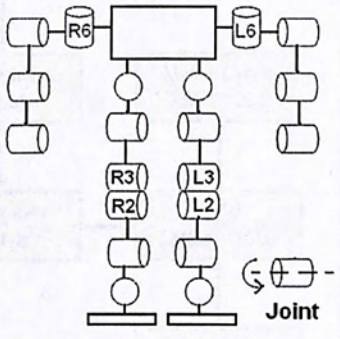


Figure 2.4: The special designs in the joint configuration of PHR

2.3 Hardware Platform

In addition to the mechanical design of PHR, we introduce the hardware platform for PHR and is shown in Fig. 2.5.

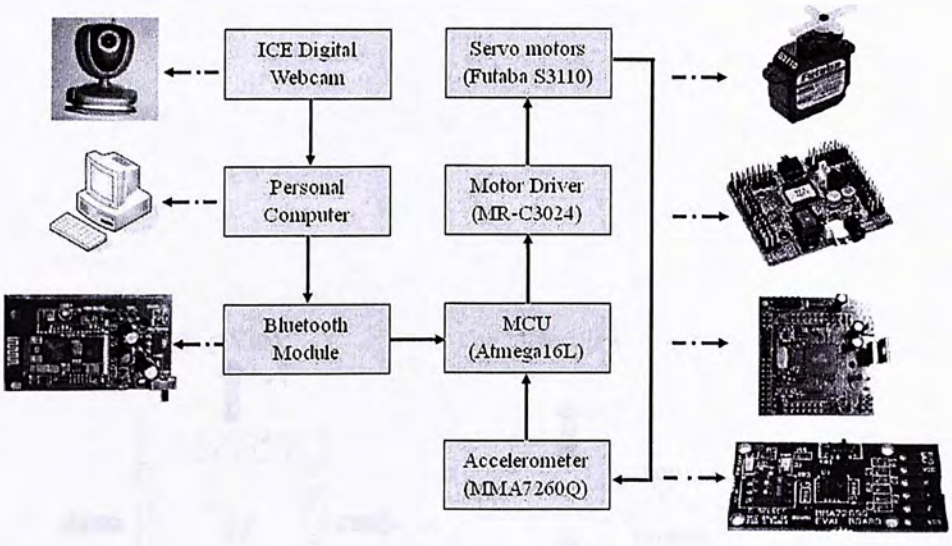
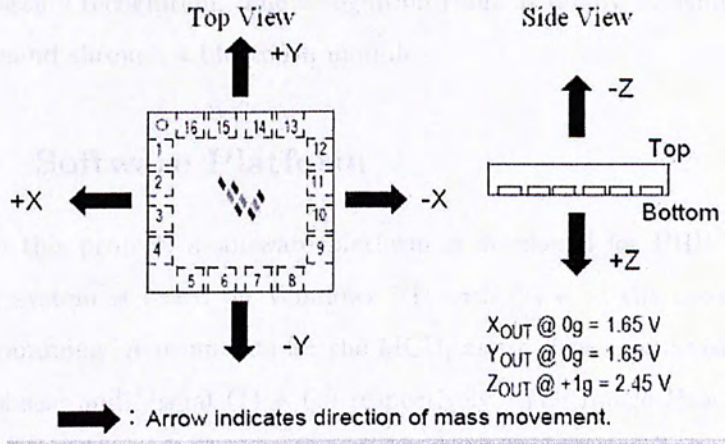


Figure 2.5: The hardware platform developed in PHR

For the actuation of PHR, servo motors, Futaba S3110 are utilized and each of which weighs only 7.7g. It provides a torque of 22 oz-in with a speed of 0.10 sec/60 deg under a working voltage of 4.8V.

For the sensory system of PHR, we first assume that the linear accelerations of the PHR along the x, y, and z axis are very small while moving. A 3-axis micro-machined accelerometer, MMA7260Q, is then attached on the body of PHR to detect the angle of inclination. It is mounted on the PHR body with the circuit board parallel to the ground. The specification of MMA7260Q under this configuration is shown in Fig.2.6 [35]-[31]. The sensitivity is set equal to 1.5g and the input voltage is 3V. The initial voltages of the three axis are 1.65V, 1.65V, and 2.45V respectively (Fig.2.6). With a tilting angle about the x, or y axis (lateral or longitudinal tilting) varied from -90^o to $+90^o$, an output voltage ranges from 0.85V to 2.45V respectively.



Device	Acceleration (g)	Sensitivity (mV/g)	Sensing Axis	Frequency (Hz)	VDD Supply Voltage (Typical) (V)	Response Time (ms)
MMA7260Q*	1.5	800	XYZ	350/150	3.3	1
	2	600	XYZ	350/150	3.3	1
	4	300	XYZ	350/150	3.3	1
	6	200	XYZ	350/150	3.3	1

*This device has selectable sensitivity (1.5g, 2g, 4g and 6g)

Figure 2.6: The specification of of MMA7260Q under the predefined configuration

For the driving system of PHR, the motor driver used is MR-C3024 which drives PHR by providing a total of 24 channels of pulse width modulation (PWM). Atmega16L is used as a microcontroller unit (MCU) which gathers data from the accelerometer and performs signal digitization. The digitized signal is then sent to the balance control algorithm.

For the vision system, an image of a hand gesture is first captured by using a ICE digital webcam. It is then transmitted to a Pentium IV 2.4-GHz personal computer (PC) for gesture recognition. The recognition result is finally transmitted to PHR as a motion command through a bluetooth module.

2.4 Software Platform

In this project, a software platform is developed for PHR (Table. 2.1). The pperating system is based on Windows XP with C++ as the programming language. The programming environments for the MCU, motor driver, and vision system are WINAVR, Robobasic and Visual C++ 6.0 respectively. The Image Processing Library utilized is OpenCV.

Table 2.1: The configurations of the software platform in PHR.

Operating System	Window XP
Programming Language	C++
Programming Environment (MCU)	WINAVR
Programming Environment (motor driver)	Robobasic
Programming Environment (vision system)	Microsoft Visual C++6.0
Image Processing Library	OpenCV

Chapter 3

Kinematics

3.1 Introduction

In our research, we study the forward and inverse kinematics of PHR which are commonly applied in gait synthesis, gait planning, and balance control [7]-[52]. In this Chapter, we first intend to present the forward kinematics of PHR. After that, we describe the inverse kinematics. They are basically two opposite mapping between the Cartesian space and joint space. It is done by analyzing the joint structures through geometric method. Once the mappings are determined, motion planning can finally be done.

3.2 Forward Kinematics

For the forward kinematics, it aims to display the stimulated result of different robot motions. Fig. 3.1 shows the joint configuration of PHR and the robot coordinate frame. For simplicity, we make two basic constraints.

- The joints L_2 and R_2 are assumed to be coincident with joints L_3 and R_3 respectively.
- The x, y, and z-axis of the robot frame are set parallel to the corresponding axis in the world frame.

The second constraint can be achieved by applying transformation matrixes to the input coordinates with respect to the world frame (Fig. 3.2). This transformation matrix consists of a translation of PHR to the origin of the world frame. A rotation of PHR about the z-axis of the world frame is then followed with the angle of rotation calculated from its hip coordinates.

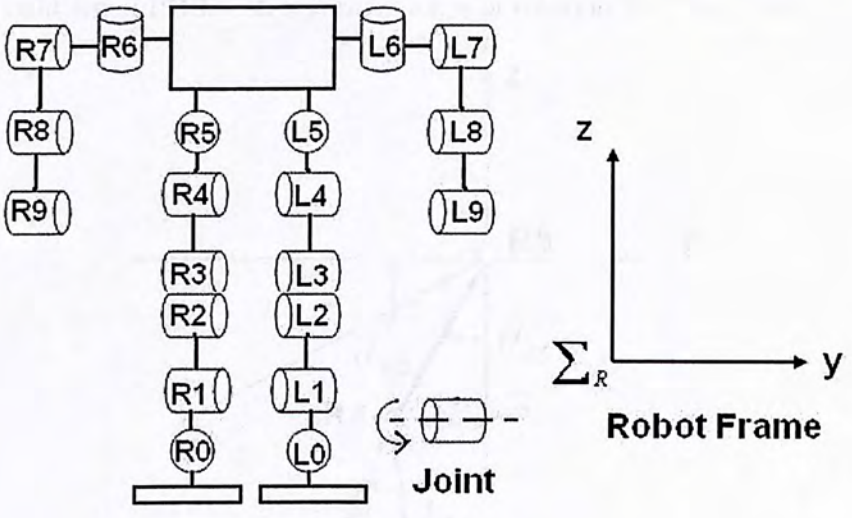


Figure 3.1: The joint configuration PHR

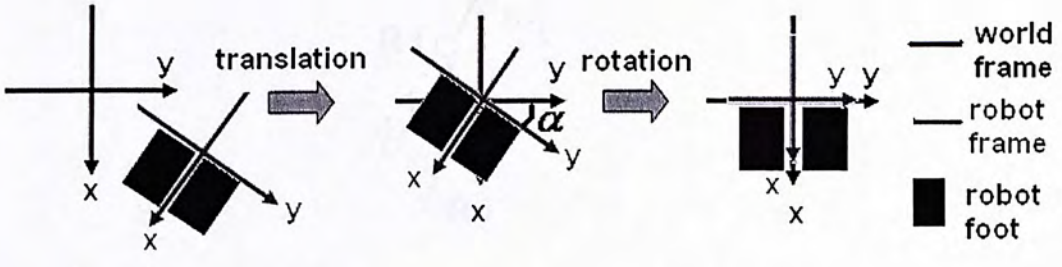


Figure 3.2: The concept of transformations applied to achieve the basic constraint

The inputs are the angles (θ_{Li} and θ_{Ri}) of joints (for $i = 1$ to $i = 8$) in PHR while the output are the corresponding Cartesian coordinates with respect to the world frame. The joint angles of L0, L9, R0, and R9 are ignored in computing the forward kinematics

as they are the end-effectors.

3.2.1 Lower Limb

We first focus on the forward kinematics of the right lower limb. It is to find the mapping from joint space (joint with index $i = 1$ to $i = 5$) to Cartesian space. Fig. 3.3 shows the right leg of PHR with a positive angle of rotation ($0 \leq \theta_{R5} \leq 90$).

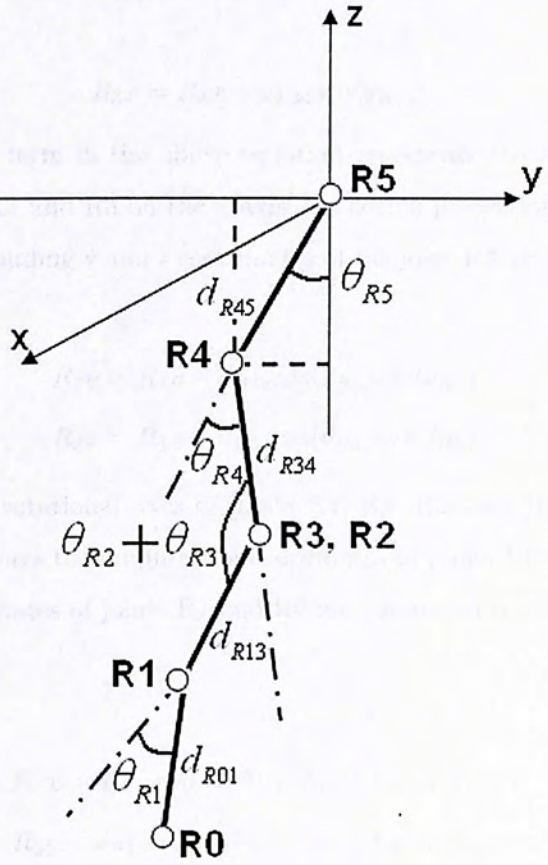


Figure 3.3: The right leg of PHR with a positive angle of rotation θ_{R5}

For the right leg of PHR, let the coordinate of joint R_5 with respect to the world frame be (R_5x, R_5y, R_5z). Based on the second constraint, the link joining the joints R5 and R4 lies on the y-z plane. The coordinate of joint R4 is then given by

$$R_4x = R_5x \tag{3.1a}$$

$$R_4y = R_5y - d_{R45}sin(\theta_{R5}) \tag{3.1b}$$

$$R_4z = R_5z - d_{R45}cos(\theta_{R5}) \tag{3.1c}$$

Aforementioned, the link joining the joints R5 and R4 lies on the y-z plane, the x-coordinate of the joint R3 is then given by

$$R_3x = R_4x + d_{R34}sin(\theta_{R4}) \tag{3.2}$$

, where the second term in the above equation represents the sine projection of link connecting the joints R4 and R3 on the x axis. Its cosine projection therefore lies on the y-z plane. The corresponding y and z coordinates of the joint R3 are determined as follows

$$R_3y = R_4y - d_{R34}cos(\theta_{R4})sin(\theta_{R5}) \tag{3.3a}$$

$$R_3z = R_4z - d_{R34}cos(\theta_{R4})cos(\theta_{R5}) \tag{3.3b}$$

From Fig. 3.1, the rotational axes of joints R1, R2, R3, and R4 are parallel to each other. Therefore, the ways to compute the coordinates of joints R0 and R1 are similar to that of R3. The coordinates of joints R1 and R0 are computed by the following equations respectively.

$$R_1x = R_3x + d_{R13}sin(-180 + \theta_{R2} + \theta_{R3} + \theta_{R4}) \tag{3.4a}$$

$$R_1y = R_3y - d_{R13}cos(-180 + \theta_{R2} + \theta_{R3} + \theta_{R4})sin(\theta_{R5}) \tag{3.4b}$$

$$R_1z = R_3z - d_{R13}cos(-180 + \theta_{R2} + \theta_{R3} + \theta_{R4})cos(\theta_{R5}) \tag{3.4c}$$

$$R_0x = R_1x + d_{R01}sin(-180 + \theta_{R1} + \theta_{R2} + \theta_{R3} + \theta_{R4}) \tag{3.5a}$$

$$R_0y = R_1y - d_{R01}cos(-180 + \theta_{R1} + \theta_{R2} + \theta_{R3} + \theta_{R4})sin(\theta_{R5}) \tag{3.5b}$$

$$R_0z = R_1z - d_{R01}cos(-180 + \theta_{R1} + \theta_{R2} + \theta_{R3} + \theta_{R4})cos(\theta_{R5}) \tag{3.5c}$$

3.2.2 Upper Limb

In this section, we introduce the forward kinematics of the right upper limb. It is to find the mapping from joint space (joint with index $i = 6$ to $i = 8$) to Cartesian space. Fig. 3.4 shows the right arm of PHR with a positive angle of rotation ($0 \leq \theta_{R6} \leq 90$).

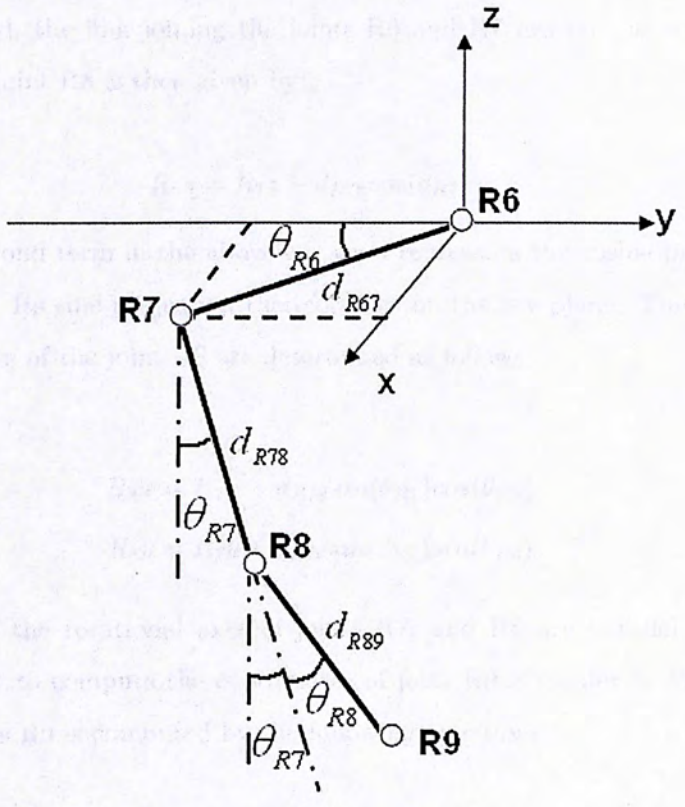


Figure 3.4: The right arm of PHR with a positive angle of rotation θ_{R6}

For the right arm of PHR, let the coordinate of joint R_6 with respect to the world frame be (R_6x, R_6y, R_6z). Based on the second constraint, the link joining the joints R6 and R7 lies on the x-y plane. The coordinate of joint R7 is then given by

$$R_7x = R_6x + d_{R67}sin(\theta_{R6}) \tag{3.6a}$$

$$R_7y = R_6y - d_{R67}cos(\theta_{R6}) \tag{3.6b}$$

$$R_7z = R_6z \tag{3.6c}$$

Aforementioned, the link joining the joints R6 and R7 lies on the x-y plane, the z-coordinate of the joint R8 is then given by

$$R_8z = R_7z - d_{R78}cos(\theta_{R7}) \tag{3.7}$$

, where the second term in the above equation represents the cosine projection of the link on the z axis. Its sine projection therefore lies on the x-y plane. The corresponding x and y coordinates of the joint R8 are determined as follows

$$R_8x = R_7x + d_{R78}sin(\theta_{R7})cos(\theta_{R6}) \tag{3.8a}$$

$$R_8y = R_7y + d_{R78}sin(\theta_{R7})sin(\theta_{R6}) \tag{3.8b}$$

From Fig. 3.1, the rotational axes of joints R7, and R8 are parallel to each other. Therefore, the way to compute the coordinates of joint R9 is similar to that of R8. The coordinate of joints R9 is computed by the following equations.

$$R_9x = R_8x + d_{R89}sin(\theta_{R7} + \theta_{R8})cos(\theta_{R6}) \tag{3.9a}$$

$$R_9y = R_8y + d_{R89}sin(\theta_{R7} + \theta_{R8})sin(\theta_{R6}) \tag{3.9b}$$

$$R_9z = R_8z - d_{R89}cos(\theta_{R7} + \theta_{R8}) \tag{3.9c}$$

The forward kinematics of the left body of PHR is expressed in a similar fashion and is listed in Appendix A.

3.3 Inverse Kinematics

In this section, the inverse kinematics of PHR is presented. We first set the following four basic constraints.

- The joints L_2 and R_2 are assumed to be coincident with joints L_3 and R_3 respectively.
- The x, y, and z-axis of the robot frame are set parallel to the corresponding axis in the world frame.
- $\theta_{L1} = \theta_{L2}$ and $\theta_{R1} = \theta_{R2}$
- $\theta_{L3} + \theta_{L4} = 90^o$ and $\theta_{R3} + \theta_{R4} = 90^o$

The first two constraints are same as those in computing the forward kinematics. Since 3-D inverse kinematics is used for gait generation, the last two constraints are set to ensure the upright of the robot body. The inputs are the Cartesian coordinates of joints L0, L5, L9, R0, R5, and R9 respect to the world frame while the output are the joint angles θ_{Li} and θ_{Ri} (for $i = 1$ to $i = 8$) in PHR. The joint angles of L0, L9, R0, and R9 are ignored in computing the inverse kinematics as they are the end-effectors.

The following section is divided into two parts. Inverse kinematics of the lower and upper limb are presented respectively.

3.3.1 Lower Limb

In this section, the inverse kinematics of the right leg of PHR is described. Fig. 3.5 shows the front view of the right leg.

From the figure, the Y-Z distance d_{RYZ} between the joints R0 and R5 and the tilt angle θ_{RYZ} on the y-z plane can be formulated as

$$d_{RYZ} = \sqrt{(R_0y - R_5y)^2 + (R_0z - R_5z)^2} \tag{3.10a}$$

$$\theta_{RYZ} = \arcsin(\frac{R_0y - R_5y}{d_{RYZ}}) \tag{3.10b}$$

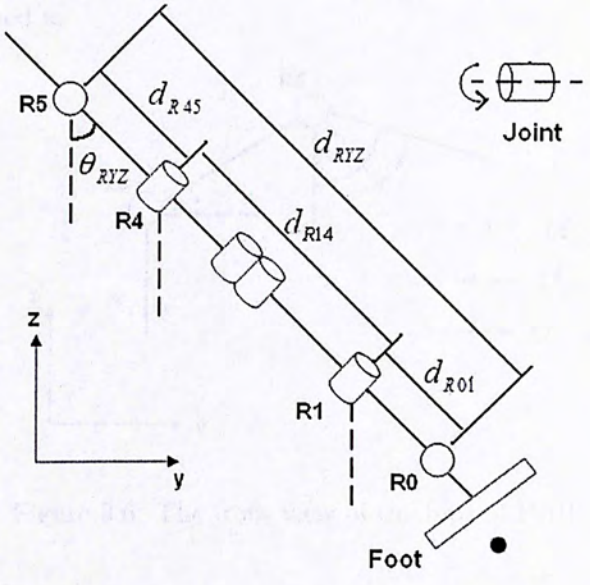


Figure 3.5: The front view of the right leg of PHR

Based on the second constraint, the line connecting the joints R4 and R5 is parallel to the y-z plane. The coordinate of joint R4 can be calculated as

$$R_4x = R_5x \tag{3.11a}$$

$$R_4y = R_5y + d_{R45}\sin\theta_{RYZ} \tag{3.11b}$$

$$R_4z = R_5z - d_{R45}\cos\theta_{RYZ} \tag{3.11c}$$

Similarly, the line connecting the joints R0 and R1 is parallel to the y-z plane, coordinate of R1 can be calculated as

$$R_1x = R_0x \tag{3.12a}$$

$$R_1y = R_0y - d_{R01}\sin\theta_{RYZ} \tag{3.12b}$$

$$R_1z = R_0z + d_{R01}\cos\theta_{RYZ} \tag{3.12c}$$

From Fig. 3.6, the angle of inclination α_{YZ} between the left and right hips (L5 and

R5) can be determined as

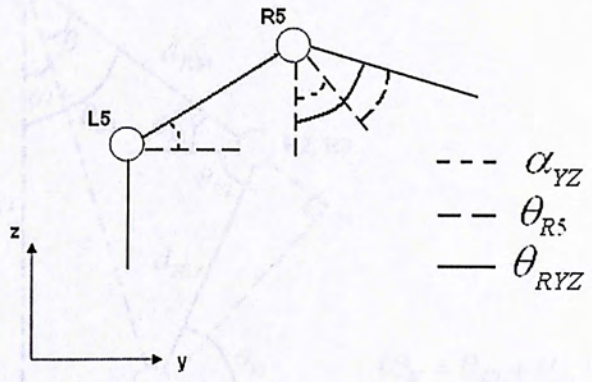


Figure 3.6: The front view of the hips of PHR

$$\alpha_{YZ} = \arctan(\frac{L_5 z - R_5 z}{L_5 y - R_5 y}) \tag{3.13}$$

The joint angle R5 is then given by

$$\theta_{R5} = \theta_{RYZ} - \alpha_{YZ} \tag{3.14}$$

From Fig. 3.7, the joint angles of R1, R2, R3, and R4 are given as follows

$$d_{R14} = d_{RYZ} - d_{R45} - d_{R01} \tag{3.15}$$

$$d_{R14X} = R_4 x - R_1 x \tag{3.16}$$

$$\theta_K = \arccos(\frac{d_{R12}^2 + d_{R34}^2 - d_{R14X}^2 - d_{R14}^2}{2 d_{R12} d_{R34}}) \tag{3.17}$$

$$\beta = \arctan(\frac{d_{R12} \sin\theta_K}{d_{R34} - d_{R12} \cos\theta_K}) \tag{3.18}$$

$$\omega = \arctan(\frac{d_{R14X}}{d_{R14}}) \tag{3.19}$$

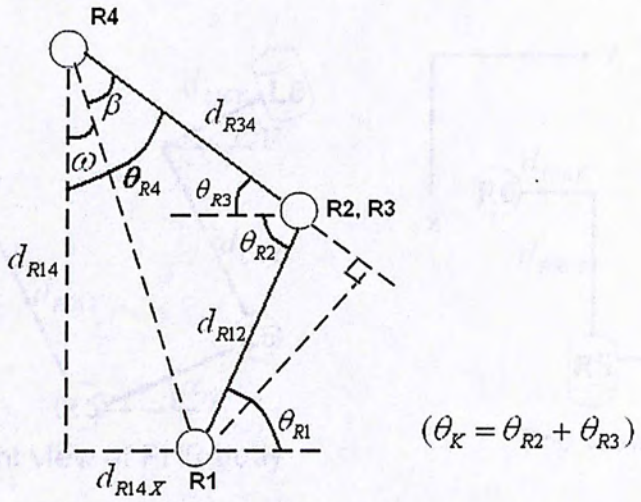


Figure 3.7: The side view of the right leg of PHR

$$\theta_{R4} = \omega + \beta \tag{3.20a}$$

$$\theta_{R3} = 90 - \theta_{R4} \tag{3.20b}$$

$$\theta_{R2} = \theta_K - \theta_{R3} \tag{3.20c}$$

$$\theta_{R1} = \theta_{R2} \tag{3.20d}$$

3.3.2 Upper Limb

In this section, the inverse kinematics of the right arm of PHR is presented. The first step is to find the Cartesian coordinate of the shoulder joints L6 and R6. Fig. 3.8 shows the front and top view of the robot body.

From the figure, the Cartesian coordinate of joint L6 and R6 are determined as

$$L_6x = L_5x - d_{R56X} \tag{3.21a}$$

$$L_6y = L_5y + d_{R56Y}cos(\alpha_{YZ}) - d_{R56Z}sin(\alpha_{YZ}) \tag{3.21b}$$

$$L_6z = L_5z + d_{R56Y}sin(\alpha_{YZ}) + d_{R56Z}cos(\alpha_{YZ}) \tag{3.21c}$$

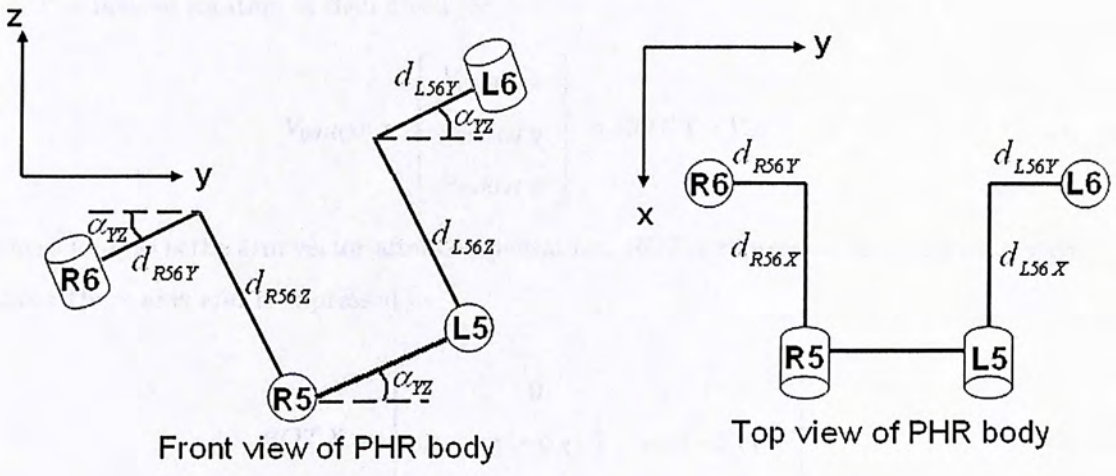


Figure 3.8: The front and top view of the robot body

$$R_6x = R_5x - d_{R56X} \tag{3.22a}$$

$$R_6y = R_5y - d_{R56Y}cos(\alpha_{YZ}) - d_{R56Z}sin(\alpha_{YZ}) \tag{3.22b}$$

$$R_6z = R_5z - d_{R56Y}sin(\alpha_{YZ}) + d_{R56Z}cos(\alpha_{YZ}) \tag{3.22c}$$

, where α_{YZ} indicates the angle of inclination between the left and right hips (L5 and R5).

In this project, we utilize the geometric approach to solve the inverse kinematics. The effect due to the hip inclination (α_{YZ}) on the Cartesian coordinates of arm joints will first be compensated as it can simplify the computation. The compensation aims to perform an inverse rotation on the arm vector V_{69} (from joint R6 to R9) about the x-axis . The angle of rotation is α_{YZ} and the arm vector is expressed as

$$V_{69} = \begin{bmatrix} V_{69}x \\ V_{69}y \\ V_{69}z \end{bmatrix} = \begin{bmatrix} R_9x \\ R_9y \\ R_9z \end{bmatrix} - \begin{bmatrix} R_6x \\ R_6y \\ R_6z \end{bmatrix} \tag{3.23}$$

The inverse rotation is then given by

$$V_{69ROT} = \begin{bmatrix} V_{69ROTx} \\ V_{69ROTy} \\ V_{69ROTz} \end{bmatrix} = ROTX * V_{69} \tag{3.24}$$

where V_{69ROT} is the arm vector after compensation. $ROTX$ represents the rotation matrix about the x-axis and is expressed as

$$ROTX = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(-\alpha_{XY}) & sin(-\alpha_{XY}) \\ 0 & -sin(-\alpha_{XY}) & cos(-\alpha_{XY}) \end{bmatrix} \tag{3.25}$$

After the inverse rotation, the next step is to find the angles for joints R_6, R_7, and R_8. Fig. 3.9 shows the configuration of the right robot arm after the compensation.

Based on the constraints and inverse rotation, the rotational axis of joint R6 is parallel to the z axis. The joint angle θ_{R6} can therefore be determined as

$$\theta_{R6} = 180 - \lambda_1 - \lambda_2 \tag{3.26}$$

, where λ_1 and λ_2 are expressed as

$$\lambda_1 = atan(\frac{V_{69}ROTX)}{V_{69}ROTY} \tag{3.27a}$$

$$d_{R69XYZ} = \sqrt{V_{69ROT}x^2 + V_{69ROT}y^2 + V_{69ROT}z^2} \tag{3.27b}$$

$$d_{R69XY} = \sqrt{V_{69ROT}x^2 + V_{69ROT}y^2} \tag{3.27c}$$

$$d_{R79XYZ} = \sqrt{d^2_{R69XYZ} - d^2_{R67}} \tag{3.27d}$$

$$h = \sqrt{d^2_{R79XYZ} - V_{69ROT}z^2} \tag{3.27e}$$

$$\lambda_2 = acos(\frac{d^2_{R69XY} + d^2_{R67} - h^2}{2d_{R69XY}d_{R67}}) \tag{3.27f}$$

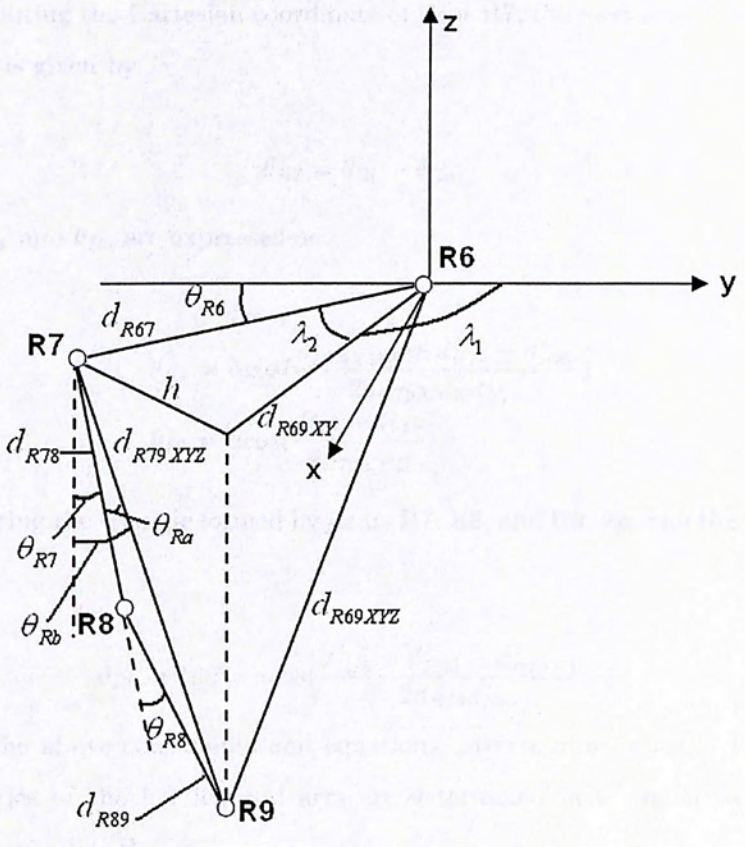


Figure 3.9: The configuration of the robot right arm after the compensation

By utilizing the joint angle θ_{R6} , the Cartesian coordinate of joint R7 is given by

$$R_7x = R_6x + d_{R67}sin(\theta_{R6}) \tag{3.28a}$$

$$R_7y = R_6y - d_{R67}cos(\theta_{R6}) \tag{3.28b}$$

$$R_7z = R_6z \tag{3.28c}$$

After computing the Cartesian coordinate of joint R7, the next step is to find the joint angle θ_{R7} and is given by

$$\theta_{R7} = \theta_{Rb} - \theta_{Ra} \tag{3.29}$$

, where θ_{Ra} and θ_{Rb} are expressed as

$$\theta_{Ra} = acos(\frac{d^2_{R79XYZ} + d^2_{R78} - d^2_{R89}}{2d_{R79XYZ}d_{R78}}) \tag{3.30a}$$

$$\theta_{Rb} = acos(\frac{R_7z - R_9z}{d_{R79XYZ}}) \tag{3.30b}$$

By considering the triangle formed by joints R7, R8, and R9, θ_{R8} can then be computed by

$$\theta_{R8} = 180 - acos(\frac{d^2_{R78} + d^2_{R89} - d^2_{R79XYZ}}{2d_{R78}d_{R89}}) \tag{3.31}$$

Based on the above constraints and equations, inverse kinematics of PHR is solved. The joints angles of the left leg and arm are determined in a similar fashion and are described in Appendix B.

Chapter 4

Gait Synthesis

4.1 Introduction

In the previous chapter, we investigate the inverse kinematics of PHR. It is a technique to compute the mapping from the Cartesian space to the joint space of PHR. In this Chapter, we suggest the use of 3-D inverse kinematics as a basic approach for gait synthesis.

For the gait synthesis of PHR, we adopt gait frames approach to generate the gait pattern for PHR. For each gait frame, we want to design the key posture for PHR during each gait step. A gait frame therefore represents the key step in a human gait pattern. This task is to define the key gait frames for a complete gait. It is done by using the gait concept presented in [54]. Once key gait frames are defined, a complete gait trajectory can be synthesized using Bezier curve interpolation. This approach can also be used to generate different gait trajectories for PHR through the manipulation of control points.

4.1.1 Difference Between Human and Robot Joints

In this thesis, PHR is expected to play and entertain humans in our life. Therefore, it is designed to walk naturally in order to present the sense of intimacy. However, the differences in joint structure between humans and biped robots make the gait synthesis

becomes complicated. Fig. 4.1 show the joint structures for humans and PHR respectively.

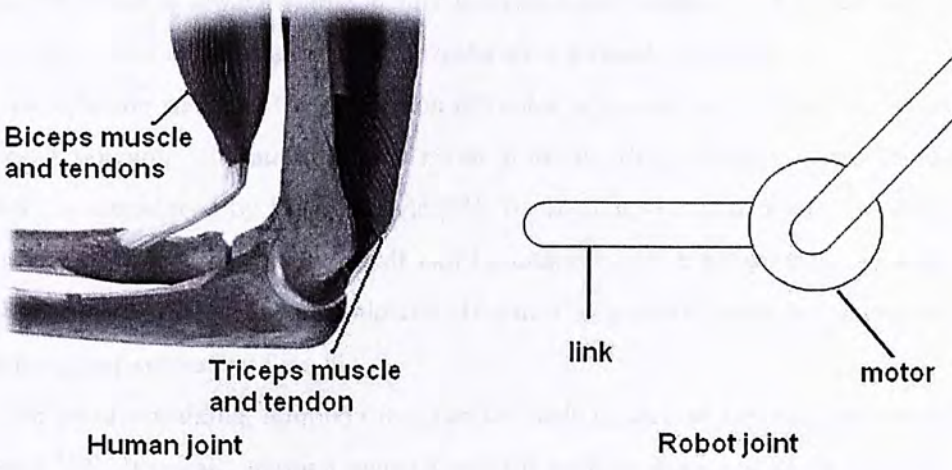


Figure 4.1: The joint structures for humans and PHR respectively

According to Ryosuke, the main difference between their joint structure is the actuation [38]. Human bones are linked at joints through the tendons, ligaments, and muscle. It is actuated by the muscle motions (contraction and relaxation) together with the functions of tendons and ligaments. For links in PHR, they are connected through motors. It is therefore actuated by the rotation of motors together with the embedded gearboxes.

Based on the difference in joint actuation, human gaits become more complex dynamic activities and are difficult to be modelled. One of the reasons is that muscle can be soft or hard. For instance, it is hard while contracting and becomes soft while relaxing. Also, the degree of freedom in human legs are much more than those of biped robots. For humans, joints can turn any direction while the motion of a motor is restricted in one direction only.

4.1.2 Difference Types of Gait for Humanoid Robots

For walking robots like humanoid robots, there are mainly three types of gait. They are 1) static gait; 2) dynamic gait; and 3) passive gait. For static walking, the control methodology is to ensure the projection of a robot's center of gravity (or center of mass)

always on the convex hull of contact points on the ground (support polygon). Through this, the robot is always stable at any position while walking. This approach is applied when the robot is walking on a flat ground with a small velocity.

In dynamic walking, the projection of center of gravity of a robot can be outside the support polygon. To maintain the robot stability, the concept of Zero Moment Point (ZMP) is introduced by Vukobratovic [51]. By keeping the ZMP inside the support polygon, the stability of a dynamic gait can be maintained. A robot with dynamic gait can therefore walk with a higher velocity. It can also perform some advanced actions like climbing and successive flips [63].

For passive walking, a biped robot can be made to walk stably with no energy input or control [33]. It usually happens when a robot is walking down a slope. During the motion, a biped robot can be regarded as two coupled pendulums. The stance leg performs like an inverted pendulum while the swing leg acts as a free pendulum attached to the stance leg at the hip. A nominal gait pattern that repeats itself is then generated if a sufficient mass is given at the hip [57]-[1].

4.2 Related Works

With the use of inverse kinematics, some researchers proposed the analysis of humans gait patterns in order to compute the gait for a humanoid robot. According to [54], the proposed algorithm generates a natural gait pattern for a biped robot based on human gait patterns. Because of this, key frames of the gait pattern for PHR are based on those defined in [54].

As aforementioned, the differences in joint structure between humans and biped robots make the gait synthesis complicated. Many researchers approach this problem through energy optimization which optimizes the gait energy of biped robots [12]-[60]. Roussel proposed an algorithm to generate unconstrained optimal gait trajectories by assuming the body mass of a biped robot is concentrated on the hip [48]. It in fact is a simplified robot dynamics which ignores the effects of centripetal forces.

Besides, Silva and Machado modeled and studied the energy efficiency of a planar

biped [10]. To maintain the smoothness of walking, they set two restrictions: the body posture maintains upright and the forward velocity of body is constant. The influence of locomotion variables such as step length, hip height, and hip ripple on the energy flow were also discussed. Instead of using a single fitness function, Capi and Yokota proposed an multi-objective evolutionary algorithm which minimizes both the energy consumption and torque change of a biped robot [11].

In addition to the above algorithms, some researchers proposed the the use of adaptive techniques such as genetic algorithm to solve the optimization problems. Tang developed a genetic algorithm based gait generation algorithm which optimizes the walking pattern of a biped robot through the minimization of energy consumption [66]. Stability of the robot is also maintained by maximizing the dynamic stable margin. Park and Choi proposed an approach to optimize the energy consumption of a biped robot while walking [41]. They applied a genetic algorithm to search the optimal gait pattern and optimal location of the mass centers of links in a biped robot .

Based on the above algorithms, gait energy and stability of a biped robot can be optimized by minimizing the energy consumption and employing learning algorithms. However, they still can not generate a natural gait pattern as they only focus on the robot's structure with little analysis on human gaits. Therefore, the study and model of human gait patterns is necessary for the synthesize of a natural gait.

In this thesis, we describe a successful example of modeling human walking patterns for the gait synthesis of PHR. The modeling process includes the analysis and identification of key phases in a human gait. From the key phases, a complete trajectory can be synthesized by curve interpolation. Once the gait profile is defined, 3-D inverse kinematics is applied to compute the corresponding joint angles. The goal is therefore to generate the profiles of joints in PHR for a complete gait through the Bezier curve interpolation of key gait phases.

4.3 Gait Frame

In this section, the concept of gait frames for PHR is presented. Each gait frame represents the key phase in a human walking pattern. Therefore, the analysis of human gait patterns is first described. Based on the patterns, we then define the gait frames for PHR.

4.3.1 Analysis of Human Gait

According to [38], a normal walking pattern of humans can be represented by Fig. 4.2.

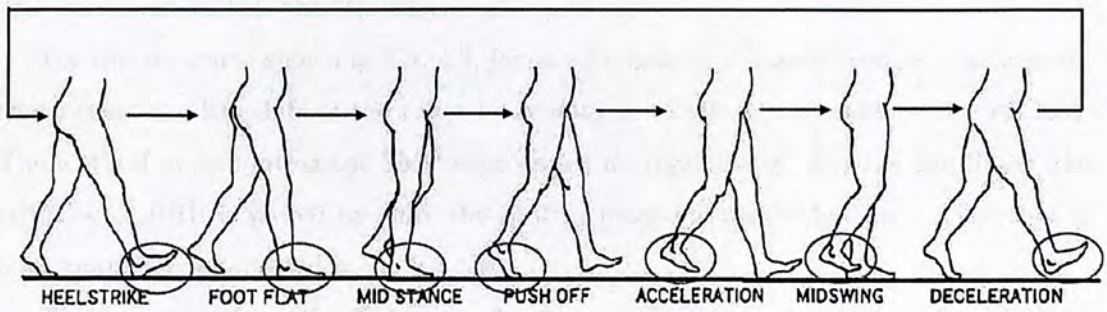


Figure 4.2: A normal walking pattern of a human

From the figure, a human walking pattern can be divided into three major support phases and each of which represents a type of body support by the legs. For the first phase, body weight of a human is supported by both legs. The others two are the phases that the body weight is supported only by the left and right leg respectively.

According to Fig. 4.2, a complete gait cycle is formed by two forward steps. A forward step actually represents two transitions. Each transition consists of three phrases and they are ordered as 1) double support phrase, 2) single support phrase, and 3) double support phrase. For a transition, the swinging leg first accelerates while leaving the ground (double support phrase). It moves up and forward until the foot reaches a certain step height (single support phrase). The swinging leg finally decelerates and strikes on the ground to end a forward step (double support phrase). A continuous human gait is therefore a repetition

of the two transitions since the end posture of the second transition is the starting posture of the first one.

4.3.2 Gait Frame for PHR

After analyzing of the human gait pattern, five keys frames for the robot gait are defined in Fig. 4.3. They consist of 3 types of phase, a) double support phase (DSP); b) left support phase (LSP); and c) right support phase (RSP). A gait cycle is formed by using three DSPs, one LSP and one RSP. In the cycle, the begin, middle and end frames are DSP, LSP and RSP are then inserted between two DSPs. In this thesis, LSP is the 2nd frame and RSP is the 4th frame.

For the 1st frame shown in Fig. 4.3, joints with index $i = 0$ and 5 rotate in a direction that causes the left shift of the robot body until the ZMP is projected on the left foot. The method in computing the ZMP is presented in Appendix C. For the 2nd frame, the right leg of PHR is picked up since the ZMP is projected on the left foot. The robot is now completely supported by its left leg.

The transition from the 2nd to the 3rd frame indicates a phase change from LSP to DSP. It means the right leg is put forward down. At the same time, joints with index $i = 0$ and 5 rotate in a direction that causes the right shift of the robot body until the ZMP is projected on the right foot.

The transition from the 3rd to the 4th frame indicates a phase change from DSP to RSP. In this transition, the left leg is picked up and the robot is now completely supported by its right leg. The left leg is then put forward down in the 5th frame. At the same time, joints with index $i = 0$ and 5 rotate in a direction that causes the left shift of the robot body until the ZMP is projected on the left foot.

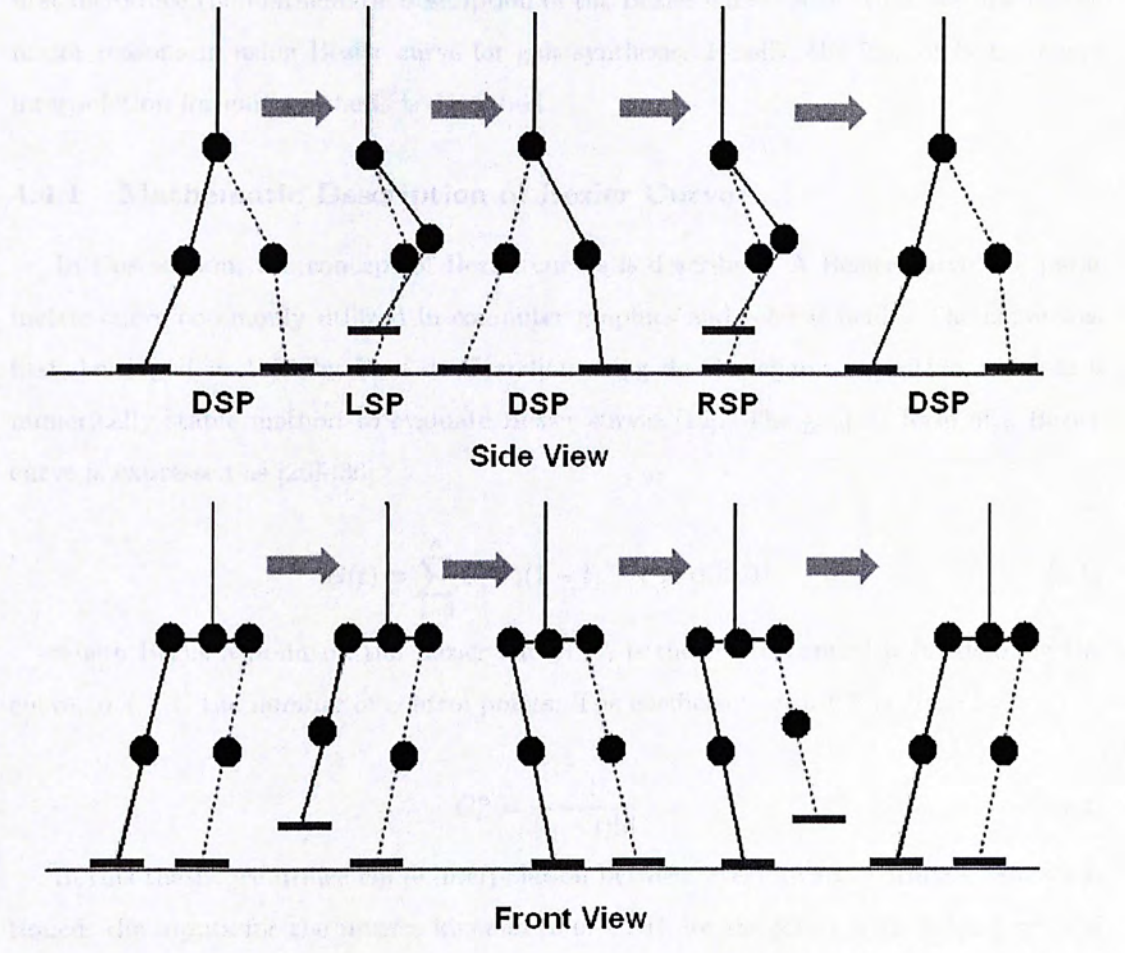


Figure 4.3: The 5 key gait frames defined for PHR

4.4 Gait Synthesis

Based on the gait frames defined in the previous section, we present the approach to generate the gait trajectory using Bezier curve interpolation. In the following sections, we first introduce the mathematic description of the Bezier curve. After that, we discuss the major reasons in using Bezier curve for gait synthesis. Finally, the idea of Bezier curve interpolation for gait synthesis is described.

4.4.1 Mathematic Description of Bezier Curve

In this section, the concept of Bezier curves is described. A Bezier curve is a parametric curve commonly utilized in computer graphics and related fields. The curve was first developed in 1959 by Paul de Casteljau using de Casteljau's algorithm which is a numerically stable method to evaluate Bezier curves [15]. The general form of a Bezier curve is expressed as [26]-[36]

$$B(t) = \sum_{i=0}^{n} C_i^n P_i (1-t)^{n-i} t^i, t \in [0,1] \tag{4.1}$$

where B(t)is a point on the Bezier curve. P_i is the $i-th$ control point defining the curve. $n+1$ is the number of control points. The coefficient term C_i^n is given by

$$C_i^n = \frac{n!}{(n-i)!i!} \tag{4.2}$$

In this thesis, we utilize curve interpolation between every two gait frames. Aformentioned, the inputs for the inverse kinematics of PHR are the joints with index i =0 and i = 5 (lower limb). It means that the coordinate of a joint (with index i=0 or 5) in the first and second frames defines the first and last control points of the Bezier curve respectively. In between the two points, we set two more control points and the total number of control points is equal to 4 (i.e. n = 3). Therefore, we apply the cubic Bezier curve in the representation of gait profile and the corresponding expression is written as

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3, t \in [0,1] \tag{4.3}$$

4.4.2 Reasons for Using Bezier Curve for Gait Synthesis

In this section, we discuss the reason for using Bezier curve for the interpolation of gait frames. From the mathematic description presented, a Bezier curve is constructed by a sequence of cubic segments, rather than linear ones. It therefore generates a smoother gait trajectory than that through a polynomial curve. Also, the computation of Bezier curve can be done by a recursion process (subdivision). It means that the arithmetic operation required to build the curve is simple.

However, the non-local control of a Bezier curve is an undesired feature. Moving a control point affects the whole curve. Although there are local controllable curves like B-splibe curve, it needs a higher computation. To compensate this, we divide a whole gait cycle into segments of Bezier curves. It is achieved by performing interpolation between two successive gait frames. By this, moving a control point changes only one segment of a gait trajectory.

4.4.3 Gait Synthesis Using Bezier Curve Interpolation

After introducing the general form of a cubic Bezier curve, the approach of Bezier curve interpolation is presented. Since we perform interpolation between every two successive frames, the total number of Bezier curve for each joint of a complete gait cycle is therefore equal to 4. The flow of curve interpolation is shown in Fig. 4.4.

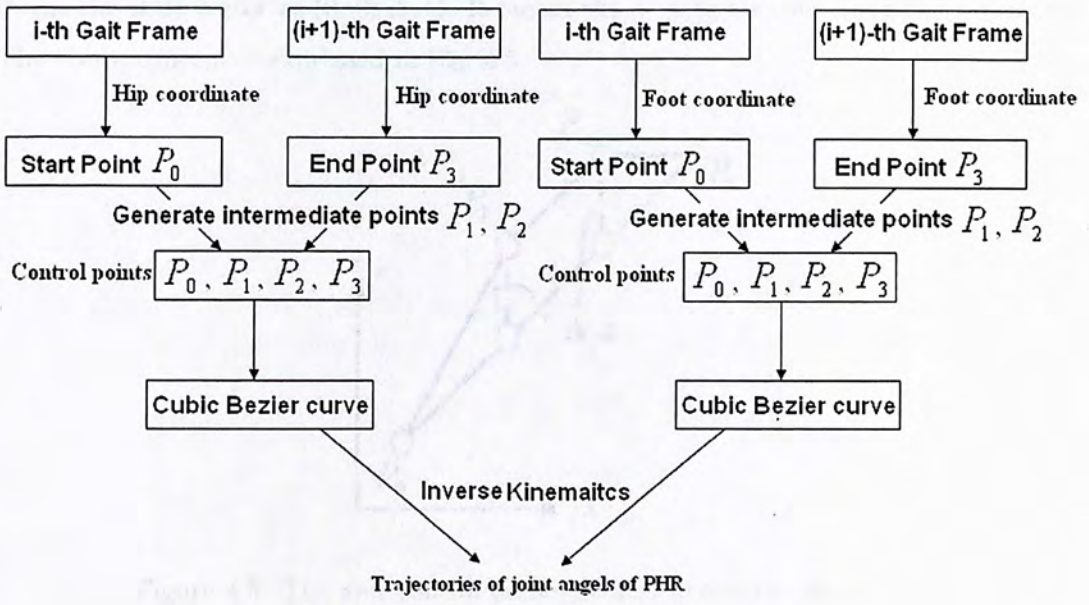


Figure 4.4: The flow of Bezier curve interpolation between two successive gait frames

For each of the joints (i=0 and i=5) at each pair of successive frames, its coordinations are first set as the start and end control points of a cubic Bezier curve (P_0 and P_3). In between the two control points, two intermediate control points are defined as

$$P_1 = \begin{bmatrix} P_1x \\ P_1y \\ P_1z \end{bmatrix} = \begin{bmatrix} P_0x \\ P_0y \\ P_0z \end{bmatrix} + \frac{1}{3}\begin{bmatrix} P_3x - P_0x \\ P_3y - P_0y \\ P_3z - P_0z \end{bmatrix} + \begin{bmatrix} \Delta_1x \\ \Delta_1y \\ \Delta_1z \end{bmatrix} \tag{4.4}$$

$$P_2 = \begin{bmatrix} P_2x \\ P_2y \\ P_2z \end{bmatrix} = \begin{bmatrix} P_0x \\ P_0y \\ P_0z \end{bmatrix} + \frac{2}{3}\begin{bmatrix} P_3x - P_0x \\ P_3y - P_0y \\ P_3z - P_0z \end{bmatrix} + \begin{bmatrix} \Delta_2x \\ \Delta_2y \\ \Delta_2z \end{bmatrix} \tag{4.5}$$

where (P_kx, P_ky, P_kz) is the Cartesian coordinate of the k-th control point (k=0, 1, 2, and 3). (Δ_jx, Δ_jy, Δ_jz) is a shift vector which represents the deviation of the j-th control point (j=1, 2) away from the line segment joining P_0 and P_3. In this thesis, we

define the shift vector as (0, 0, $\Delta_j z$). It means the deviation exists along the z-axis only. The above concept is expressed in Fig. 4.5.

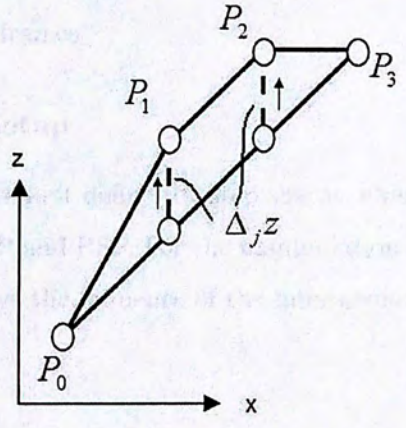


Figure 4.5: The four control points defined in a cubic Bezier curve

After defining the four control points, we can compute those intermediate points on the curve by using the equation listed in the previous section. Those points actually represent the joints' trajectories (in Cartesian space) between two successive gait frames. For each pair of successive frames, we get 2 Bezier curves per leg. One is for the hip joint (joint index i=5) and the other is for the foot joint (joint index i=0). The resolution of the curves is defined by the parameter t which is set equal to 60 in this thesis. For those 60 point pairs (hip and foot joints), we compute the corresponding joint angles of PHR using the 3-D inverse kinematics. Based on the four pairs of successive gait frames, a complete gait cycle can be synthesized by repeating the above processes.

4.5 Experiments

In our experiments, we show the gait pattern synthesized through the concept of Bezier curve interpolation on gait frames.

4.5.1 Experimental Setup

For the gait synthesis, we first define the step size as 40mm. The foot is picked 10mm above the ground in the LSP and RSP. For the manipulation of gait trajectory, we choose two different $\Delta_j z$. It displays the influence of the intermediate control points (j=1, 2) on the path.

4.5.2 Results

By using the above algorithms, we generate the gait pattern for PHR. We use Matlab to peform the gait synthesis and simulation. Fig. 4.6 and Fig. 4.7 shows the gait tarjectories of the first gait step with $\Delta_j z = 5$ and $\Delta_j z = 10$ respectively. The front and side views of the five gait frames are then described in Fig. 4.8.

The joint angles of the left and right legs in the gait frames are plotted in Fig. 4.9 and Fig. 4.10 respectively. The corresponding numerical values are listed in Table. 4.1.

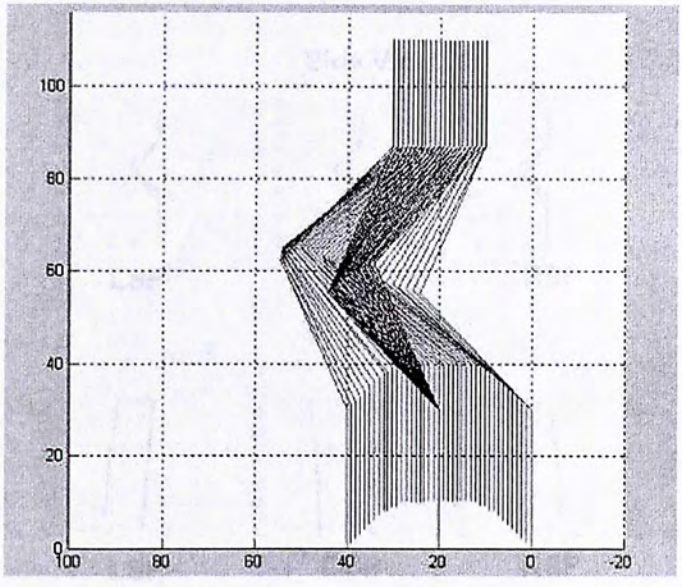


Figure 4.6: The gait tarjectories of the first gait step of PHR with $\Delta_j z = 5$ ((j=1, 2))

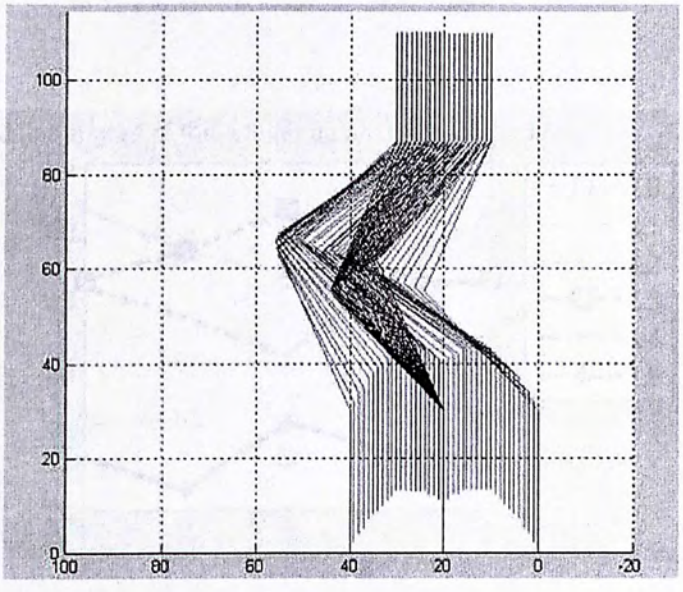


Figure 4.7: The gait tarjectories of the first gait step of PHR with $\Delta_j z = 10$ ((j=1, 2))

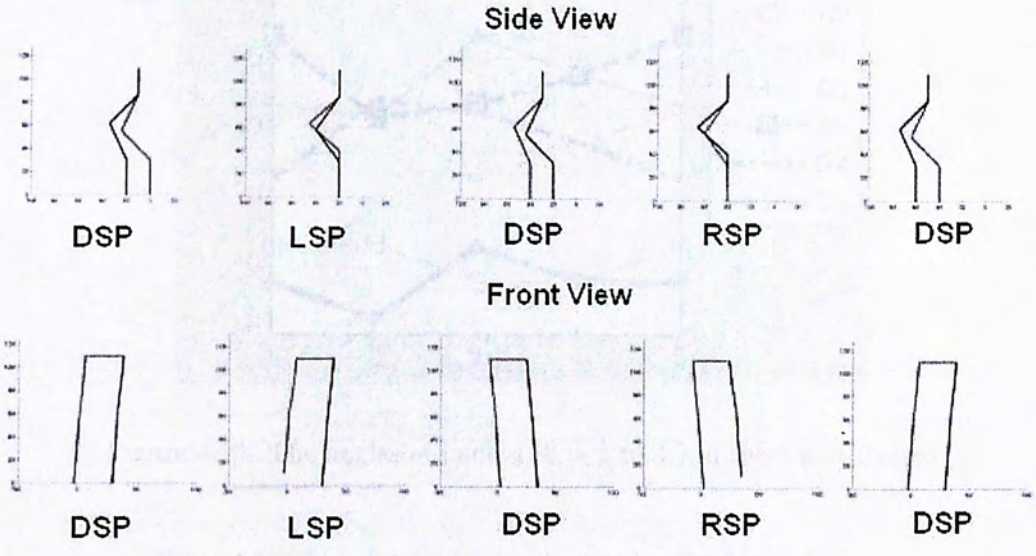


Figure 4.8: The front and side views of the five gait frames

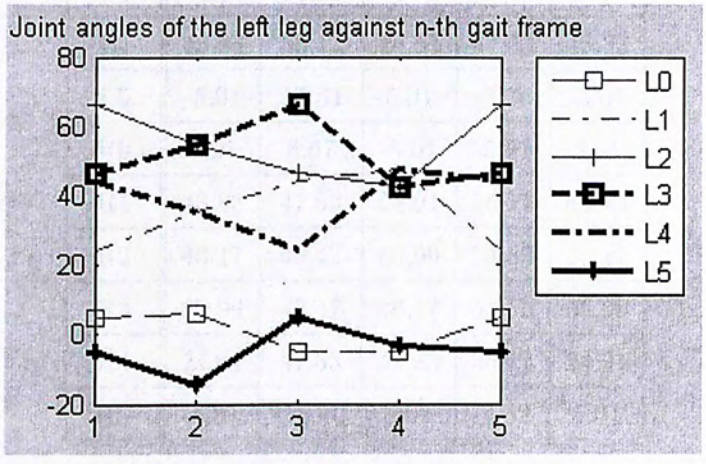


Figure 4.9: The angles of joints (L_i = 1 to 4) in the 5 gait frames

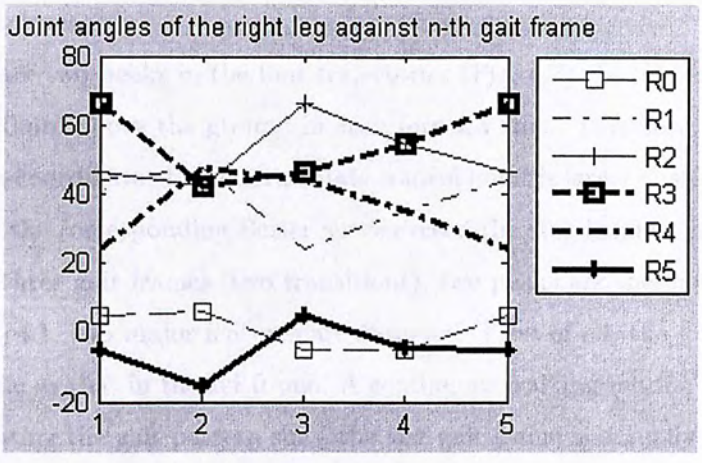


Figure 4.10: The angles of joints (R_i = 1 to 4) in the 5 gait frames

Table 4.1: The joint angles (in degree) for the 5 gait frames.

Joint	DSP	LSP	DSP	RSP	DSP
L0	5.01	6.07	-5.01	-5.01	5.01
L1	24.01	35.17	43.83	47.53	24.01
L2	65.99	54.83	46.17	42.47	65.99
L3	46.17	54.83	65.99	42.47	46.17
L4	43.83	35.17	24.01	47.53	43.83
L5	-5.01	14.34	5.01	-3.26	-5.01
R0	5.01	6.07	-5.01	-5.01	5.01
R1	43.83	47.53	24.01	35.17	43.83
R2	46.17	42.47	65.99	54.83	46.17
R3	65.99	42.47	46.17	54.83	65.99
R4	24.01	47.53	43.83	35.17	24.01
R5	-5.01	-15.40	5.01	-4.32	5.01

4.6 Discussion

From Fig. 4.6 and Fig. 4.7, the difference in gait trajectories of the right leg indicates the effect of z-deviation $\Delta_1 z$ and $\Delta_2 z$ of the intermediate control points between two

successive gait frames.

Also, there are two peaks in the foot trajectories (Fig. 4.7). In this experiment, PHR picks the foot 10mm above the ground in each forward step. Therefore, a peak actually shows that the z-coordinate of an intermediate control point is larger than 10mm and some of the points on the corresponding Bezier curve exceed the step height. Since a single gait step represents three gait frames (two transitions), two peaks are therefore resulted.

From Table. 4.1, two major features are observed. First of all, the joint angles in the 5th frame is same as that in the 1st frame. A continuous walking motion can therefore be formed by repeating the gait pattern since the last gait frame is actually equal to the 1st frame. Also, the pattern for the joints ($i = 1$ to 4) on one of the legs is reversed to that of another leg (Fig. 4.9 and Fig. 4.10). It is because those 4 joints on both legs lie on the same plane. The leg motions in picking up and putting down are the same with a reverse order.

4.7 Conclusion and Future Work

In this chapter, the algorithm for gait synthesis of PHR is presented. We utilize the concept of gait frames in defining a gait cycle. For a gait cycle, the gait trajectory is generated using Bezier curve interpolation. For the Bezier curve interpolation between successive gait frames, gait trajectory can be manipulated by changing the intermediate control points. Through this, gait of PHR can be modified for some specific purposes like obstacle avoidance.

After the curve interpolation, the corresponding joints angles are determined by using the 3-D inverse kinematics. The proposed algorithm improves the flexibility in motion planning when compared with 2-D approach. The whole algorithm can obviously be applied to other humanoid robots with a similar joint configuration.

The gait pattern synthesized in this thesis focuses on the flat ground motion. Also, the criteria of walking speed is not considered in this thesis. In the future, motion planning in a complex scene such as walking on an inclined or rough ground will first be employed. The ways to improve the walking speed will also be studied.

Chapter 5

Balance Algorithm for PHR

5.1 Introduction

As aforementioned, a humanoid robot is inherently suffered from instability and problems in balance. To maintain the motion stability and reduce the risk of falling, balance control scheme should therefore be implemented.

In this chapter, we adopt a balance control scheme for PHR which aims to amend the abnormal tilting of PHR using geometric approach. The control methodology first analyzes the effect of tilting on PHR. After that, it figures out the ways to compensate such an influence. It is done by analyzing the shift in robot body under a certain extent of tilting. Once the shift is computed, a compensated action can be synthesized using the 3-D inverse kinematics. The balance algorithm can also be applied in other humanoid robots if the corresponding inverse kinematics is known.

5.2 Related Works

In reality, motions of humanoid robots are usually suffered from instability due to the unevenness of a ground. Balance and stability are two major criteria for the development of

humanoid robots. According to the control system in the Honda humanoid robot, the first development stage of control system is based on the concept of using impact absorption mechanism [45], [59]. The design aims to make the foot soft by using some pieces of rubber bush and sponge for shock absorbance (Fig. 5.1). Although a humanoid robot with the above mechanism can walk on a ground with unevenness surface, the restoring force (given by the sponge) is limited [22]. Therefore, a balance control system is necessary for humanoid robots.

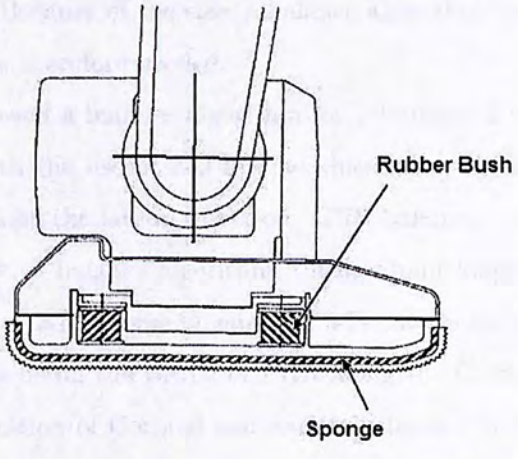


Figure 5.1: The foot structure developed for honda humanoid robot

To measure the robot stability, the concept of Zero Moment Point (ZMP) is proposed by Vukobratovic and has been discussed in the previous chapter. Based on the measurement metric, many control methodologies are proposed. Lim developed a control algorithm for human-like walking of a biped humanoid robot in 2000 [39]. It aims to maintain the stability of a biped robot under dynamic walking. The control methodology is to compensate the moments generated by the limbs' motions. Some force feedback based control algorithms are also presented which require the monitoring of different sensory information in realtime [44]-[49].

In addition to the analysis of forces and moments, some researchers implement balance control algorithms through different adaptive techniques such as neural network and fuzzy

logic. Katic presented a survey and tried to discuss the basic principles and concepts of neural network control techniques in humanoid robots [8]. It shows different examples to demonstrate the pros and cons in applying those learning techniques.

Moreover, Su proposed a fuzzy rule based balance controller for humanoid robots in 2007 [62]. It first analyzes the force (feet) and acceleration (body) of a robot. A fuzzy controller is then utilized to adjust the hip or ankle motor according to the tilting. However, due to the size and hardware constraints, it is not practical to apply such algorithms in a mini-humanoid robot. Because of the size, a balance algorithm based on limited channels of senory information is therefore needed.

In 2005, Park proposed a balance algorithm for a humanoid robot, GHR [46]. It is a geometric approach with the use of 2-D inverse kinematics. A surfboard is first used to create a tilting angle along the lateral direction. GHR balances itself by shortening one of its legs. To be an effective balance algorithm, tilting about longitudinal axis should also be considered. Instead of 2-D inverse kinematics, 3-D one should therefore be considered.

In this thesis, we focus on the tilting of PHR along the Coronal and Sagittal planes. Fig. 5.2 shows the definition of Coronal and Sagittal planes [16]-[23]. Because of the size and limitations in hardware platform, we propose to develop a balance scheme for PHR using geometric method. It reqiures only two channels of tilting information. It works together with the 3-D inverse kinematics of PHR to compute the tilting compensated motions. Based on geometric approach, the computation is uncomplicated and fast. Another benefit is that the algorithm can obviously be applied in other biped robots once we know their inverse kinematics.

5.3 Balance Algorithm

In this section, the concept of the proposed balance control scheme is presented. Any tilting on a robot can actually be regarded as inclinations along the Coronal and Sagittal planes. Based on the small hardware platform on PHR, we develop a balance control scheme which is uncomplicated and requires only two channels of sensory information. It is a geometric method with the use of 3-D inverse kinematics to compute the compensated

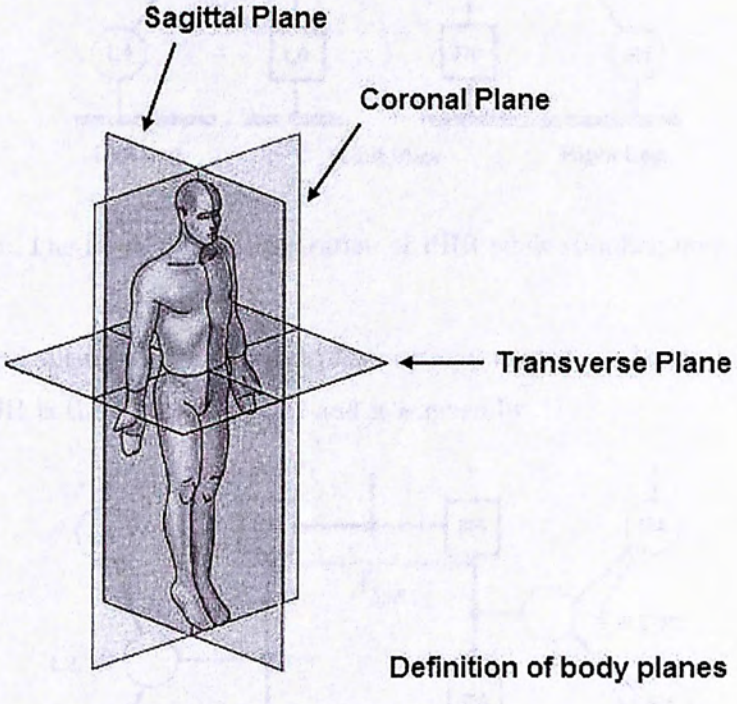


Figure 5.2: The definition of Coronal and Sagittal planes

actions in correcting the tilting.

Fig. 5.3 shows the lower limb configuration of PHR while standing on a flat ground.

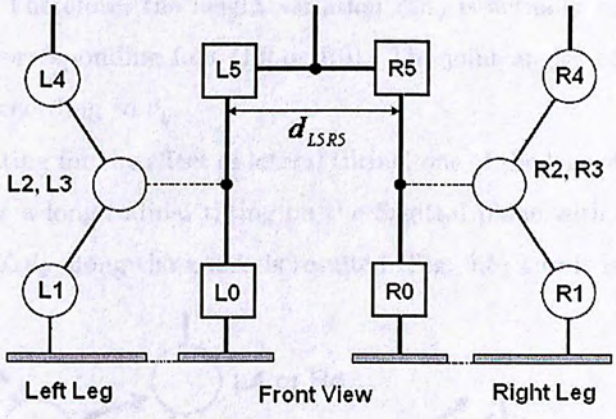


Figure 5.3: The lower limb configuration of PHR while standing on a flat ground

For a lateral tilting on the Coronal plane with an angle θ_y, a height variation $\triangle h_y$ of the feet of PHR is the result (Fig. 5.4) and it is given by

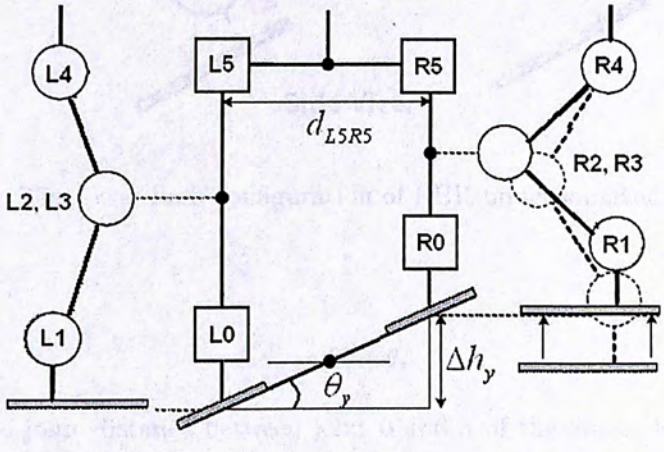


Figure 5.4: The lower limb configuration of PHR under lateral tilting

$$\triangle h_y = d_{L5R5} \tan \theta_y \tag{5.1}$$

Based on the height variation, one of the leg will be bent which shortens the length between its foot and hip. By this corrective action, the tilting of PHR on the Coronal plane is adjusted. Therefore, the height variation $\triangle h_y$ is actually used to adjust the z-coordinate of the corresponding foot (L0 or R0). The joint angles of both foots ($i = 0$) are also modified according to θ_y.

After compensating for the effect of lateral tilting, one of the legs of PHR is now longer than the other. For a longitudinal tilting on the Sagittal plane with an angle θ_x, a shift of the robot body, $\triangle d_x$ along the x-axis is resulted (Fig. 5.5) and it is determined as

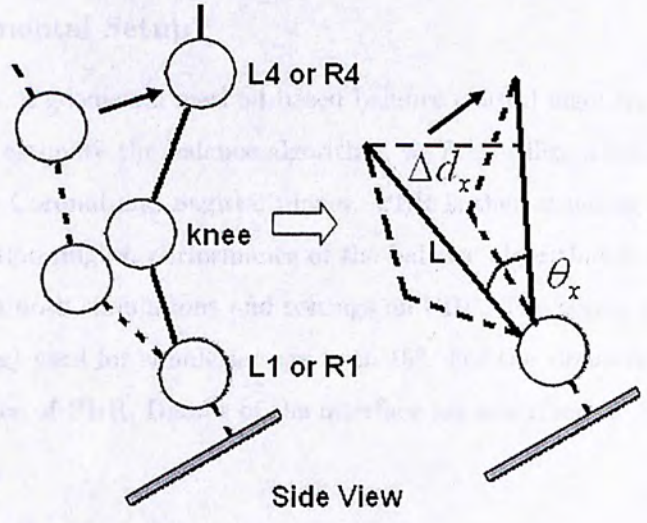


Figure 5.5: The lower limb configuration of PHR under longitudinal tilting

$$\triangle d_x = L \sin \theta_x \tag{5.2}$$

where L is the joint distance between joint 0 and 5 of the longer leg. Based on the value of body shift, a reverse shift at the hips' coordinate is performed. By this corrective action, the tilting of the robot on the Sagittal plane is adjusted. Therefore, the shift $\triangle d_x$ is actually used to adjust the x-coordinates of the hip joints (L5 and R5). Their z-coordinates are also updated to be the joint distance between joint 0 and 5 of the longer leg. The joint angles of L1 and R1 are also modified according to θ_x.

Based on the above balance algorithm, the tilting of PHR can be detected and corrected. If any lateral or longitudinal tilting on PHR is found, the corresponding equations are applied. Corrective actions are then computed using the 3-D inverse kinematics.

5.4 Experiments

In our experiments, we evaluate the proposed balance control scheme by showing the influences of tilting and the corresponding compensated-actions of PHR.

5.4.1 Experimental Setup

In this project, a geometric method based balance control algorithm is developed for PHR. In order to estimate the balance algorithm, we first utilize a board which provides tilting along both Coronal and Sagittal planes. PHR is then standing on the board. By giving the inclination angles, performance of the balance algorithm is estimated. In this thesis, we perform both simulations and testings on PHR. The tilting angles (lateral and longitudinal tilting) used for simulation are both 15^o. For the simulations, they are done in the user interface of PHR. Details of the interface are described in Appendix D.

5.4.2 Results

By using the balance algotirhm, we generate some stimulated and testing results for PHR. Fig 5.6 shows the effect of a lateral tilting on PHR. PHR then balances itself by shortening one of the legs and the corresponding improvements are shown in Fig 5.7. Fig 5.8 shows the effect of a longitudinal tilting on PHR. PHR then balances itself by shifting the body backward and the corresponding results are shown in Fig 5.9.

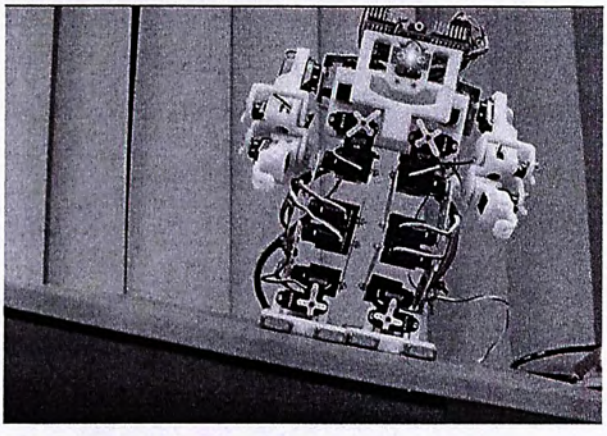


Figure 5.6: The effect of a lateral tilting on PHR

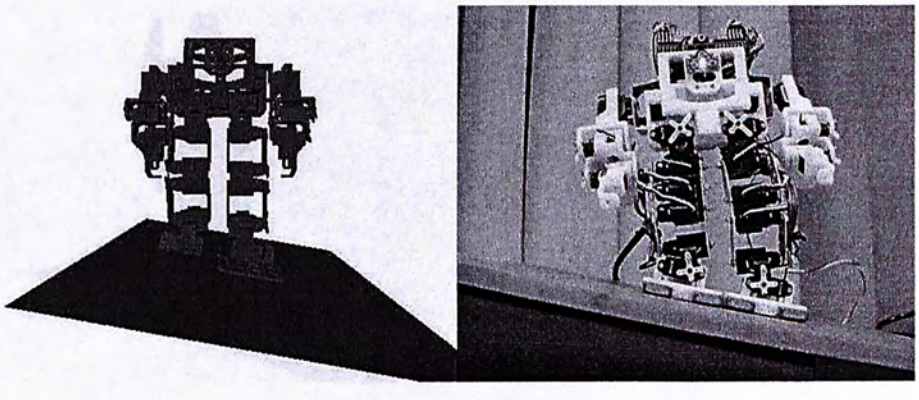


Figure 5.7: The simulated and testing results of PHR under tilting along the Coronal plane

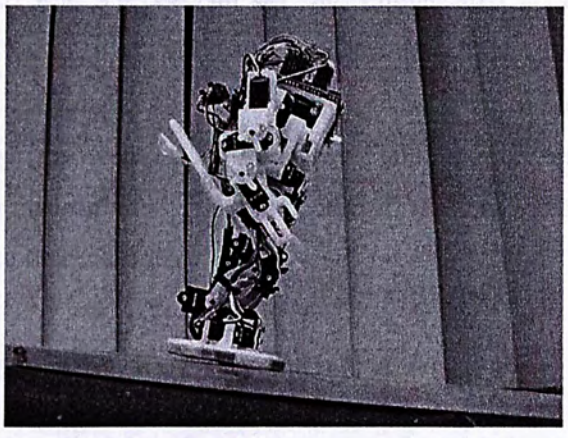


Figure 5.8: The effect of a longitudinal tilting on PHR

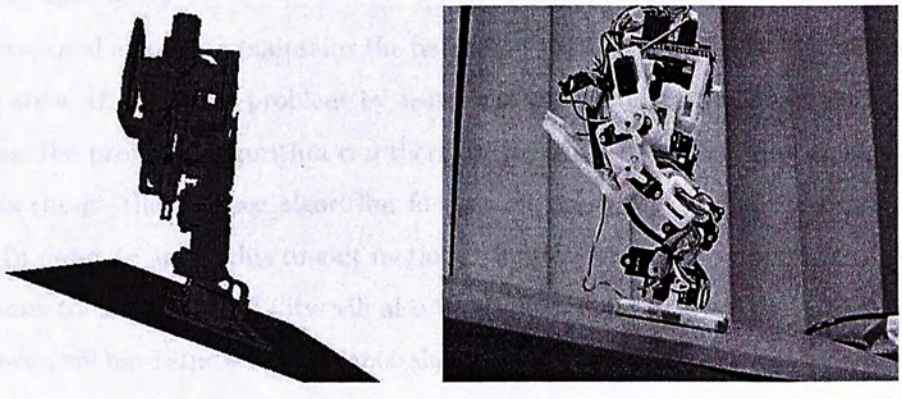


Figure 5.9: The simulated and testing results of PHR under tilting along the Sagittal plane

5.5 Discussion

From Fig. 5.7 and Fig. 5.9, we can see the compensated actions generated by the balance algorithm. The body of PHR keeps upright in both experiments and it therefore trys to maintain the ZMP inside the contact polygon between the foots and the ground.

In addition, the left part of both figures shows a simulation result while the right part represents the testing performance. It can be observed that the testing results give similar performances to that of simulations.

5.6 Conclusion and Future Work

In this chapter, the algorithm for balance control of PHR is presented. We apply the geometric method together with the use of 3-D inverse kinematics to implement the balance algorithm. For the algorithm, we first utilize the accelerometer to measure the tilting angles of PHR along the Coronal and Sagittal planes. From the sensory information, we then analyze the influences of both lateral and longitudinal tilting on PHR by using the geometric method. We finally compute the compensated action for PHR through the 3-D inverse kinematics.

The proposed algorithm maintains the balance of PHR by keeping the upright posture. Since we solve the balance problem by using the geometric approach and 3-D inverse kinematics, the proposed algorithm can therefore be applied to other humanoid robots.

In this thesis, the balance algorithm focuses on the tilting of PHR under a certain posture. In order to apply this to gait motions, the relationship of walking velocities and accelerations to the robot stability will also be studied.

Moreover, we have studied the balance algorithm of PHR with both legs for supporting. Since the gait pattern of PHR includes the single support phrase, the balance algorithm shall therefore consider this case.

Chapter 6

Human-robot Interaction System through Hand Gestures

6.1 Introduction

After discussing the kinematics, gait and balance control of PHR, we present the human-robot interaction. It is actually a recognition system which allows the communication with PHR through hand gestures. In this chapter, we first give an overview on some related works. Then, we describe our approach for database establishment. This task is to generate and store the features of hand gestures. It is done by using the concept of Haar wavelet representation. Once the database is built, recognition of hand gestures is performed. It is a new measurement metric which utilizes the concept of L^2 norm with a penalty factor.

6.2 Related Works

Traditionally, sign language is commonly used as a communication language for auditory handicapped people. In some cases, it is used to assist voice communication. There is

also a trend towards using hand gestures as a perspective way of communication between humans and robots. Therefore, many hand gesture recognition researches are proposed. In this research area, the recognition of hand gestures can be classified into glove based techniques and computer vision based techniques.

For glove based techniques, they mainly utilize sensory gloves to measure the angles and spatial positions of hands and fingers. According to [40], a wearable electronic technology that recognizes and translates sign language into spoken English is proposed by Allen. Brashear has also proposed a lab-based sign language recognition system. It consists of multiple sensors for disambiguation of noisy data which can improve the recognition accuracy.[14].

Moreover, Hollar demonstrated an approach in hand gesture recognition using a sensing glove with 6 embedded accelerometers [55]. It recognizes 28 static hand gestures and the computation time is 1 characters/second. However, the proposed algorithm is not suitable for realtime application. Lamar and Bhuiyant have also developed another recognition system by using colored gloves and neural networks algorithm [50]. The success rate ranges from 70% to 93%. Although the proposed systems can recognize hand gestures, the wearing of a sensory glove is not convenient for daily application.

For computer vision based techniques, one or a set of cameras are utilized to capture hand images for recognition. According to [42], a method of hand gestures recognition is proposed by Shin. It is based on computer vision techniques without restricting backgrounds or using any markers. The proposed method first separates the region of hand motion from complex background images by measuring the entropy from adjacent frame images. Hand gestures are then recognized by the approach of improved centroidal profile. However, mis-recognitions can be caused by hand gestures with similar spatial features. Therefore the number of hand gestures that can be recognized by the proposed algorithm is limited. To be an effective hand gestures recognition system, it should be glove-free, fast, accurate, and has a small database.

In 1995, Jacobs has presented a fast image Querying algorithm [3]. It gives benefits in fast recognition with low database size. For a 128x128 image query on a database of

20,000 images, the recognition time is under 0.5 second. In fact, the success rate is highly influenced by the object orientation in an image. To apply the algorithm in our hand gestures recognition system, improvements should be made in hand extraction, features selection, database searching, and recognition accuracy.

6.3 Flow of Hand Gesture Recognition

In this section, the flow of the hand gestures recognition algorithm is presented. Fig. 6.1 shows the outline of the proposed algorithm. A hand image with a resolution of 160x120 is first captured using a ICE digital webcam (Fig. 6.2). A hand is then extracted from the image using a skin color approach. The extracted hand is rotated which standardizes the hand's orientation by applying the concept of principal axis (axis of elongation). Before extracting the hand features, the rotated image is resized into a square image with a resolution of 64x64.

During the feature extraction process, the resized image is first transformed to YIQ color space. After the conversion, the resized image is transformed using 2-D normalized standard Haar wavelet decomposition. Key features of a hand sign are then selected by using truncation and quantization. Those selected features are compressed and stored in a database. During the recognition process, a measurement metric is utilized to measure the similarity between the features of a test image and those in the database. We evaluate our algorithm by using some randomly selected hand gestures from a traditional sign language (Fig. 6.3).

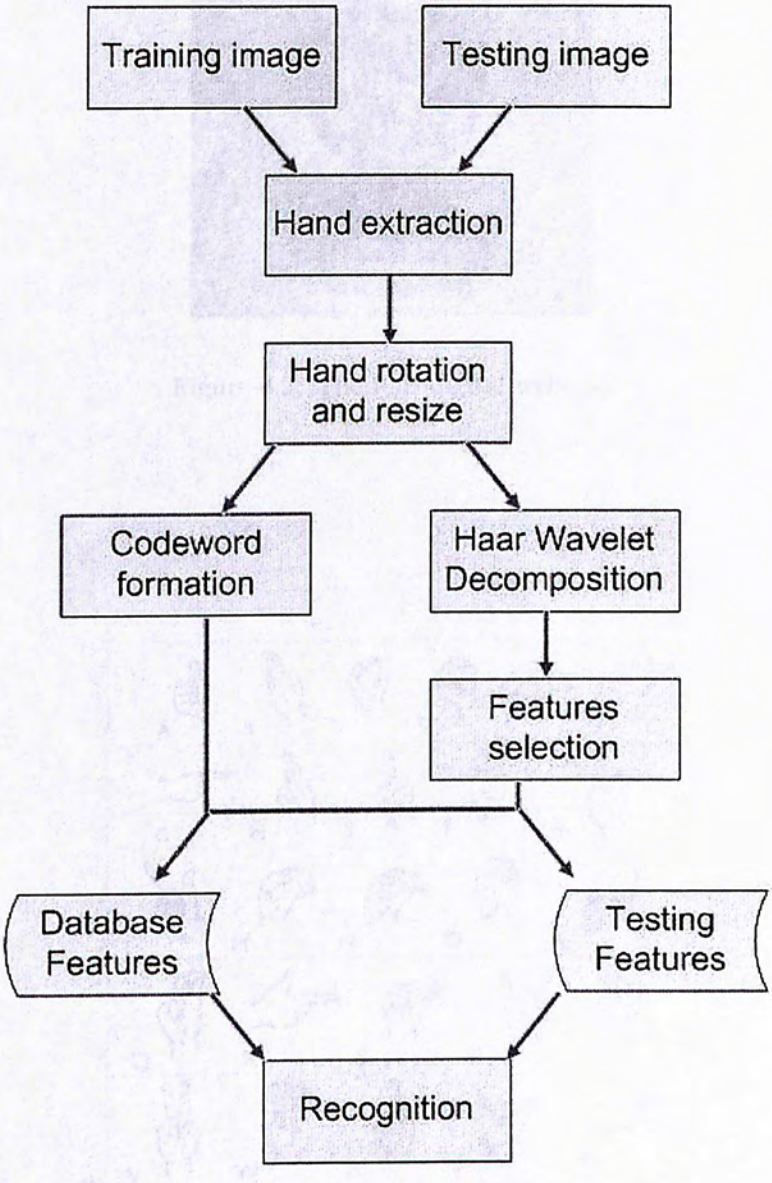


Figure 6.1: The flow of the proposed hand gesture recognition algorithm

Figure 6.2: The ICE digital webcam

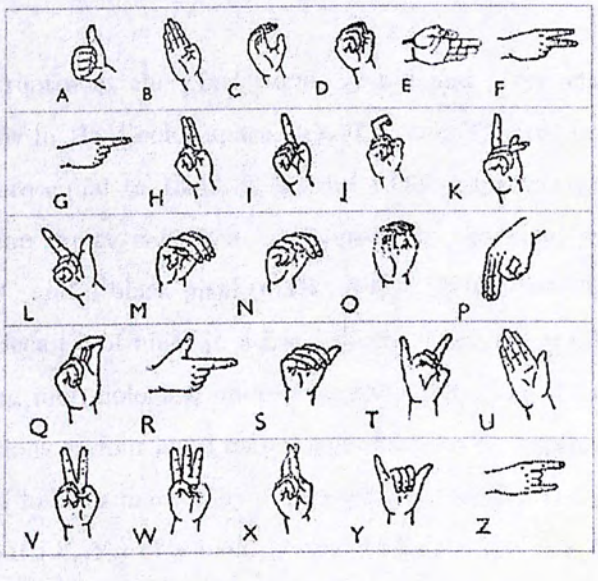


Figure 6.3: The hand gestures in a traditional sign language

6.4 Database Establishment

In this section, we focus on the discussion in constructing the database. It consists of three tasks. They are a) Hand Detection and Preprocessing, b) Extraction of Features, and c) Storage of Features. Details of those tasks are presented in the following subsections.

6.4.1 Hand Detection and Preprocessing

During the hand extraction process, an image with a resolution of 160×120 is first captured. The hand is then extracted using the skin color approach [13]. We assume that the front arm of a user is covered by clothes. A pixel is defined as a skin pixel if it satisfies the following conditions

- 0.5(R-G) > T_E
- $\frac{R}{G} < T_R$
- Hue < T_H

where R and G represent the pixel values in red and green channels respectively. Hue is the component in HSV color space. T_E, T_R, and T_H are thresholds determined experimentally and are equal to 13.42, 1.76, and 23.89 respectively. By the skin color approach, a hand map image can then be defined. In the hand map image, a white pixel (pixel value = 1) and a black pixel (pixel value = 0) indicate the skin and non-skin pixels respectively. Because of noise in a hand image, holes are resulted which are then minimized by utilizing morphological operations [20] - [53]. Fig. 6.4 shows the effects of morphological operations on four hand map images and the corresponding extracted hand regions. An extracted hand is marked by a blue rectangle with a red point indicating the centroid. The centroid (X_c,Y_c) of a hand is used to locate the axis of elongation and is computed by

$$X_c = \frac{1}{A}\sum_{i=1}^{64}\sum_{j=1}^{64} i \times H[i,j] \tag{6.1}$$

$$Y_c = \frac{1}{A}\sum_{i=1}^{64}\sum_{j=1}^{64} j \times H[i,j] \tag{6.2}$$

, where $[i, j]$ is the pixel coordinate at the hand map image with pixel value $H[i,j]$. A is the hand area which is given by

$$A = \sum_{i=1}^{64}\sum_{j=1}^{64} H[i,j] \tag{6.3}$$

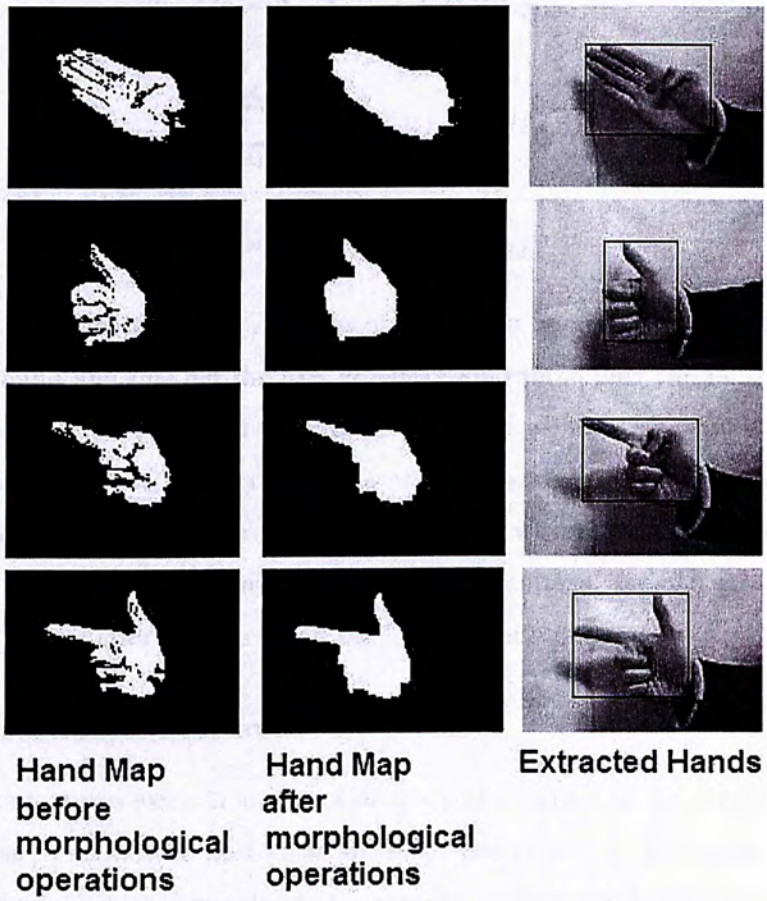


Figure 6.4: The effects of morphological operations on four of the hand map images

After the extraction, hand rotation is performed. Rather than storing a huge number of images in the database, only a few images are needed once the hands' orientations are

fixed. To achieve this, the idea of principal axis is introduced [27]. For a hand map image, angle of the gesture's principal axis is given by

$$\theta = 0.5 \times \arctan \frac{b}{a - c} \tag{6.4}$$

, where θ indicates the orientation of the principal axis which passes through the centroid of a hand. The variables a, b, and c are given by

$$a = \sum_{i=1}^{64} \sum_{j=1}^{64} (i - X_c)^2 H[i, j] \tag{6.5}$$

$$b = \sum_{i=1}^{64} \sum_{j=1}^{64} (i - X_c)(j - Y_c) H[i, j] \tag{6.6}$$

$$c = \sum_{i=1}^{64} \sum_{j=1}^{64} (j - Y_c)^2 H[i, j] \tag{6.7}$$

For a particular hand gesture, its axis of elongation relative to the hand is fixed. By rotating the hand and making the axis of elongation overlapping the image's horizontal axis, the hand orientation in different images with the same hand gesture is then fixed. The database size can be highly reduced by this approach. To facilitate the recognition, the rotated hand image is then resized into an image with 64x64 pixels. Fig. 6.5 shows the principal axis of a hand gesture with different orientations. Fig. 6.6 shows some of the resized hand images after standardizing the hand orientations.

6.4.2 Extraction of Features

During the features extraction, the color space of a resized image is first transformed. Previous research shows the best color space for the proposed recognition algorithm is YIQ. The formulae which approximate the conversion from the RGB color space to YIQ space are given by

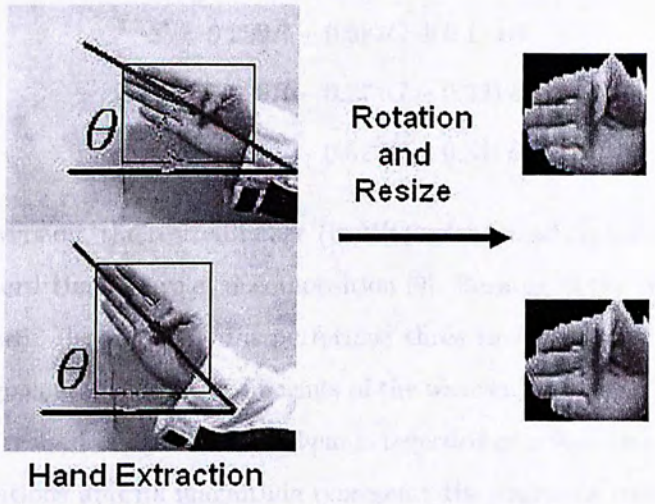


Figure 6.5: The idea of standardizing the orientation of a hand gesture

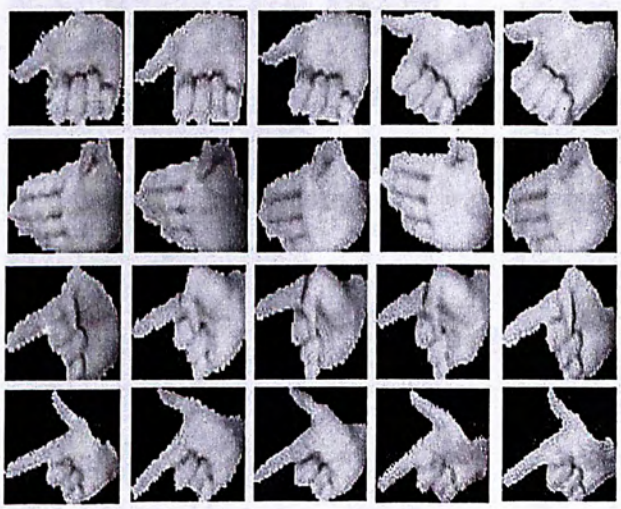


Figure 6.6: The resized and rotated hand images

$$Y = 0.299R + 0.587G + 0.114B \tag{6.8a}$$

$$I = 0.596R - 0.274G - 0.321B \tag{6.8b}$$

$$Q = 0.211R - 0.523G + 0.311B \tag{6.8c}$$

After the conversion, the resized image (in YIQ color model) is transformed using 2-D normalized standard Haar Wavelet decomposition [9]. Because of the three color channels in the resized image, decomposition is performed three times (channel by channel). The transformed image contains detail coefficients of the wavelet basic functions with the same dimension as the resized image. Each of them is regarded as a feature of the hand gesture at different resolutions and its magnitude represents the degree of contribution. Fig. 6.7 shows the luminance channel of some transformed images. Different colors in the images represent the magnitudes of the detail coefficients.

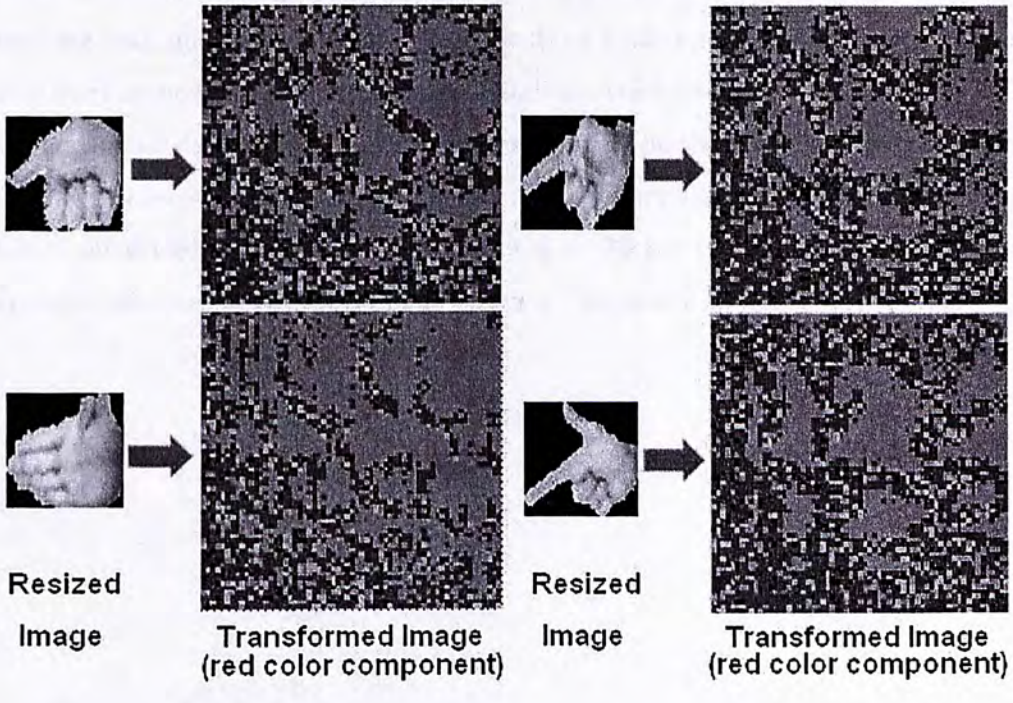


Figure 6.7: The transformed images of the luminance channel

Rather than using all the detail coefficients in the transformed image, truncation is performed which reduces the database size. During the truncation, we keep only the coefficients with largest magnitudes. The discriminatory power of the measurement metric is also improved because it lets the metric to consider only the most important features. To evenly distribute the detail coefficients in the transformed image, it is divided into 4 equal portions. For each portion in a color channel, only the first 180 coefficients with the largest magnitudes are kept. Actually, the more the detail coefficients we truncate, the smaller the database size and the faster computation we get. However, details of the hand gesture will also be lost with the truncation of coefficients. Experiments show that the optimal number of detail coefficients for recognition is 180 per portion.

After the truncation, quantization of detail coefficients is performed which gives benefits in speeding the search, reducing the storage and improving the discriminatory power of the metric. Therefore, detail coefficients are quantized to three levels (0 and ± 1) according to their magnitudes. They are then stored in a database. Fig. 6.8 shows the results after truncating and quantizing the detail coefficients of 4 hand gestures (luminance channel). Yellow lines in the images before truncation indicate the boundaries of the 4 portions. The white and black dots in the images after quantization indicate the quantized coefficients with magnitudes $+1$ and -1 respectively. After the quantization, the total number of non-zero quantized coefficients is $(180 \times 4) \times \frac{2}{3} = 480$ per color channel. Fig. 6.9 shows the quantization results of the 15 hand gestures (luminance channel).

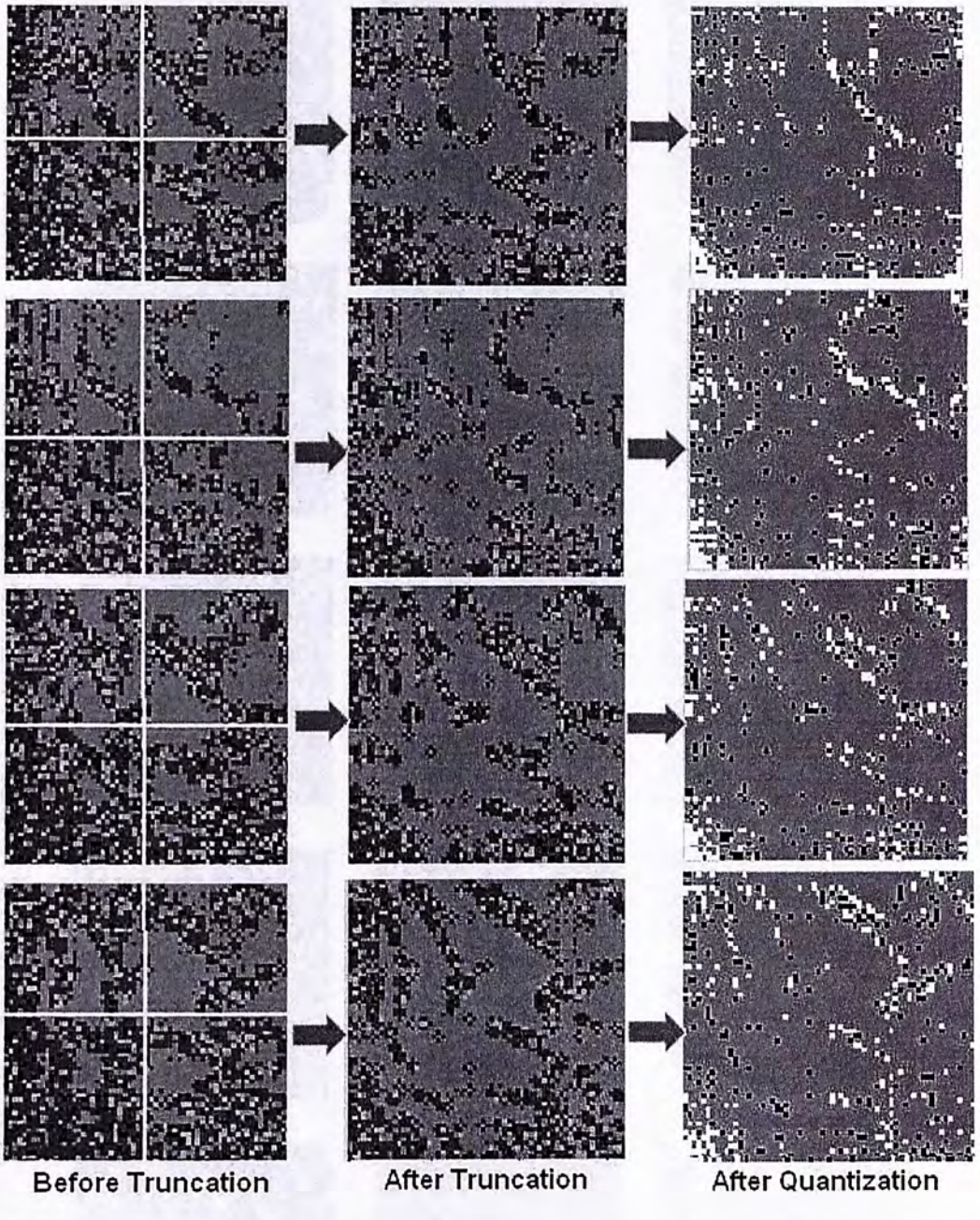


Figure 6.8: The results after truncating and quantizing the detail coefficients of 4 hand gestures

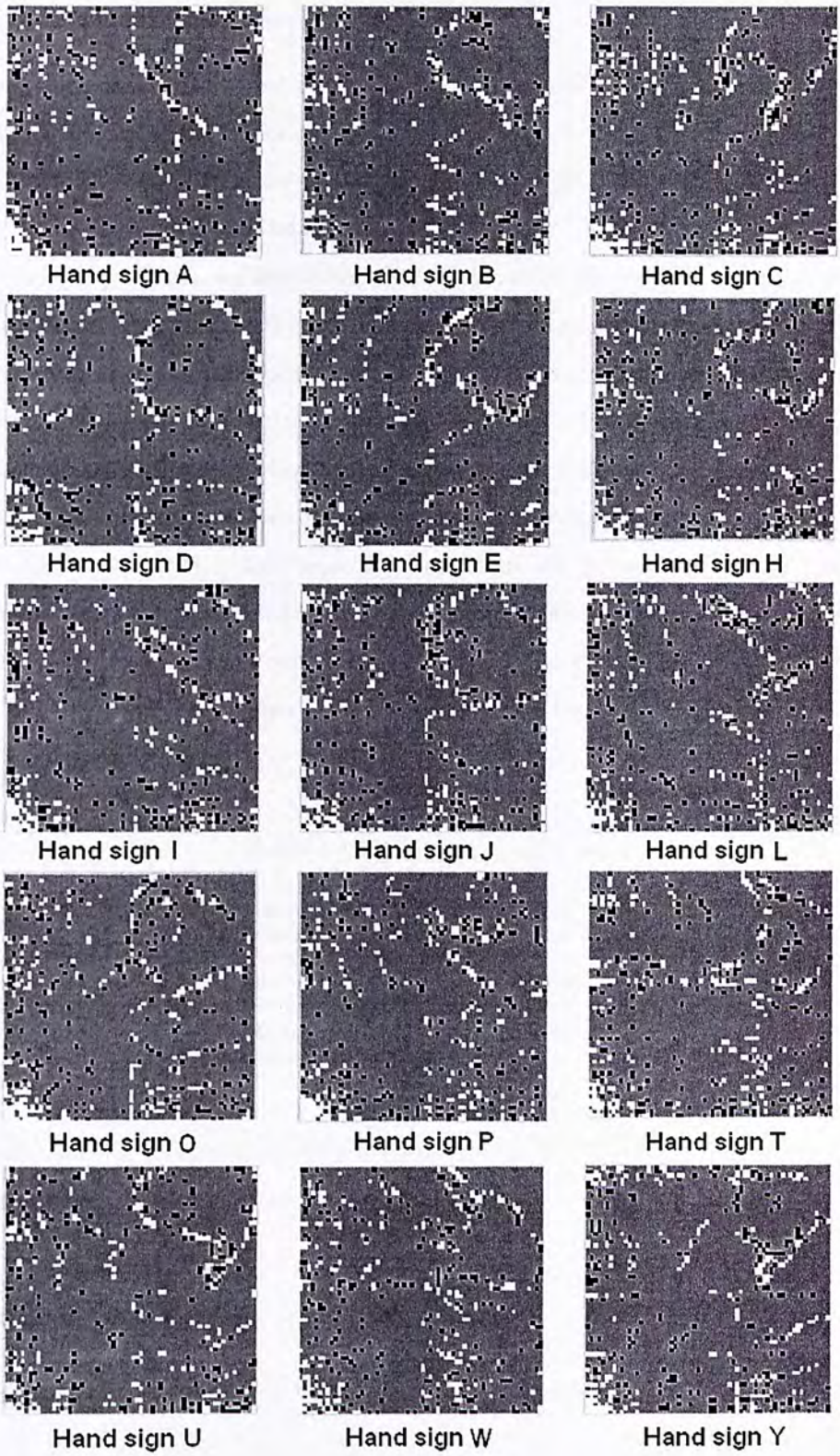


Figure 6.9: The quantization results of the 15 hand gestures(luminance channel)

6.4.3 Storage of Features

After the quantization, features of each hand sample are first stored in text files. Obviously, the size of the uncompressed text files is larger than the original hand image (in jpeg format) since a jpeg file is already compressed [19]. Therefore, we compress the hand features stored in the database.

For the compression, we utilize the free compression library, zlib. It was written by Jean-loup Gailly and Mark Adler [37] and the public version is first released in 1995. According to [25], zlib is an abstraction of the DEFLATE compression algorithm used in the gzip file compression program [18]. It provides in-memory compression and decompression functions, including integrity checks of the uncompressed data.

During the compression, two arrays of non-zero quantized values (+1 and -1) in each color channel are stored in two separate files. Each file is compressed using zlib. We compare the efficiency of our features extraction algorithm on data compression by using three reference files. Each of them stores a channel of the original pixel values in the hand image. The following table shows a comparison between the compression results of feature files and the corresponding reference files.

Table 6.1: Compression results.

File	Size before compression	Size after compression	Compression power
Features files	16.36kbs	806bs	20.78
Reference files	39.60kbs	16.90kbs	2.34

From table 6.1, there is a significant improvement on the compression power between compressed features files and reference files. The compression power is defined as [17]

$$Data\ compression\ ratio\ (Compression\ power) = \frac{Uncompressed\ size}{Compressed\ size} \tag{6.9}$$

From the above equation, a larger compression power represents a higher the com-

pression efficiency. The difference in compression power is actually due to the application of truncation and quantization on the Haar wavelet coefficients. It is because the two techniques together with the Haar wavelet decomposition create many sequences of continuous zeros in the feature files which facilitate the compression algorithm. Therefore, the database for our hand gestures recognition system is built from feature files after the zlib compression.

6.5 Hand Gesture Recognition

During the gesture recognition, a test image is first captured and transformed into the quantized wavelet coefficients using the above algorithms. To facilitate the searching process, the idea of codeword scheme is introduced. Since the orientation of a hand gesture is standardized, the distribution of hand pixels in a resized image is almost fixed. Recognition of the test image is performed based on those database images with a similar distribution. The computation power can then be lowered.

For the codeword assignment, a resized image is first divided into four equal portions in the same way as the transformed image (Fig. 6.8). The area of a hand in each of the four portions is then computed. If the hand area in a portion occupies more than half of the portion area, a code of 1 is assigned. Otherwise, 0 is assigned. A codeword which shows the distribution of hand pixels is then formed by arranging those 4 codes together. If any of the database images has a same codeword as the test image, recognition is then performed. Fig. 6.10 shows the codewords of 15 hand images.

During the recognition, we propose a measurement metric which is a weighted L^2 norm with the use of a penalty factor. It is expressed as

$$Score_{Total} = Score_A - Score_Q + Score_P \tag{6.10}$$

, where $Score_A$, $Score_Q$, and $Score_P$ are defined as

$$Score_A = \omega_0 |Q_c[0,0] - T_c[0,0]| \tag{6.11}$$

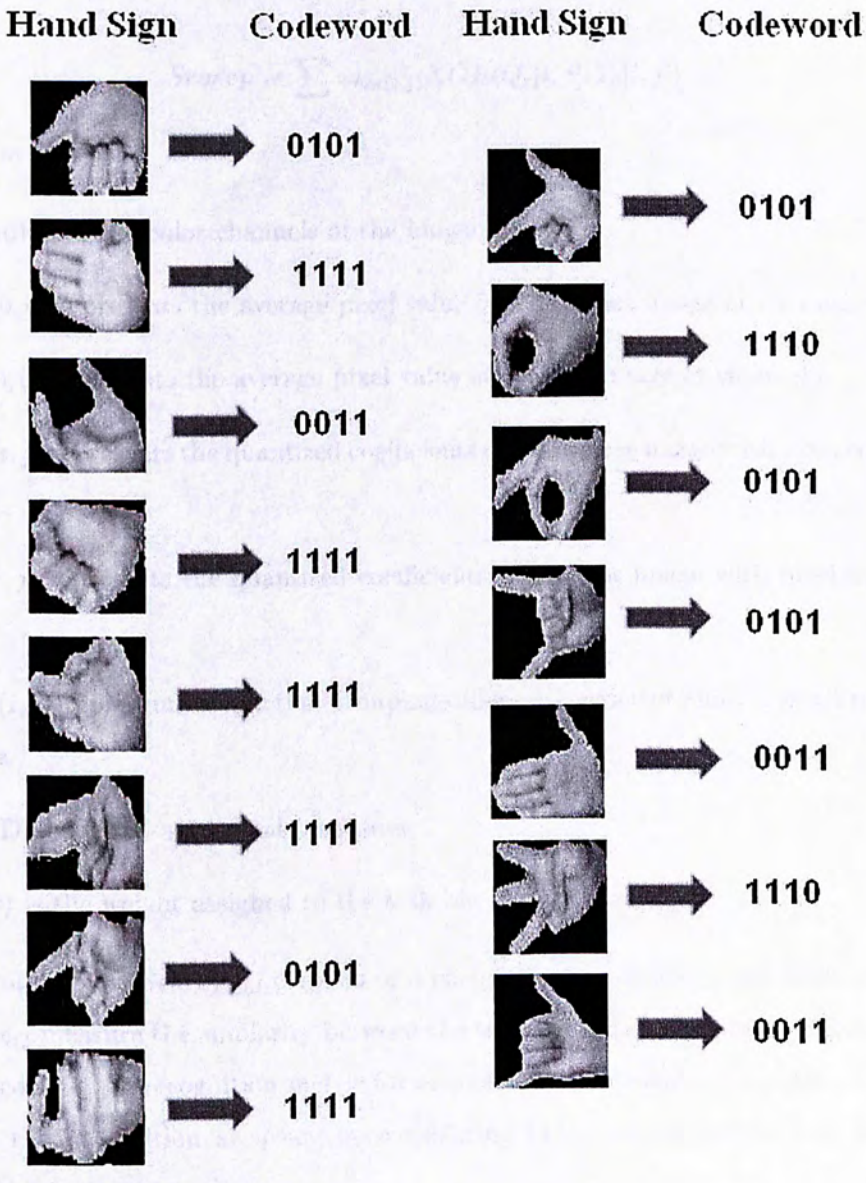


Figure 6.10: The codewords of 15 hand images

$$Score_Q = \sum_{i,j} \omega_{bin(i,j)} AND(Q_c[i,j], T_c[i,j]) \tag{6.12}$$

$$Score_P = \sum_{i,j} \omega_{bin(i,j)} XOR(Q_c[i,j], T_c[i,j]) \tag{6.13}$$

, where

- c indicates the color channels of the image
- $Q_c[0,0]$ represents the average pixel value of a database image at channel c
- $T_c[0,0]$ represents the average pixel value of the test image at channel c
- $Q_c[i,j]$ represents the quantized coefficients of a database image with pixel coordinate [i,j]
- $T_c[i,j]$ represents the quantized coefficients of the test image with pixel coordinate [i,j]
- $bin(i,j)$ represents a function grouping different coefficients into a small number of bins.
- AND and XOR are logical operators
- $\omega(k)$ is the weight assigned to the k-th bin

The total score, $Score_{Total}$ consists of 3 parts: $Score_A$, $Score_Q$, and $Score_P$. $Score_A$ and $Score_Q$ measure the similarity between the test and database images. $Score_P$ gives a penalty score to the recognition metric for mismatched coefficients. Therefore, the metric improves the recognition accuracy by considering both the similarities and differences between the two gestures.

In this algorithm, a bin function and a weight function are defined in the metric to improve the performance. For the bin function, it aims to group the quantized coefficients according to the scale of the wavelet functions to which they correspond. It gives a benefit in reducing the number of weights required for the metric and is given by

$$bin(i,j) = min\{max\{level(i), level(j)\}, 5\} \tag{6.14}$$

, where $level(i) = \lfloor \log_2(i) \rfloor$ and $k \in 0,1,...,5$.

Based on the bin function, different weights are assigned according to the bin values of quantized coefficients. The aim is to further improve the discriminatory power of the proposed metric. From the quantization results in Fig. 6.9, the probability of having a non-zero quantized coefficient decreases with the increase in the level of Haar Wavelet basis function. It can be observed through the distribution of non-zero quantized coefficients in a quantized image. Therefore, the weight function is defined by

$$\omega(k) = k \tag{6.15}$$

By applying the above recognition metric, the database image with a minimum $Score_{Total}$ indicates the matched hand gesture.

6.6 Experiments

In our experiments, we show the performance of our hand gestures recognition system.

6.6.1 Experimental Setup

Fig. 6.11 shows an interface developed for the recognition system. The left window displays the hand capturing process. A hand region is extracted and marked by a blue rectangle. A red point in the rectangle indicates the centroid of the extracted hand. The right part shows the recognition result and the corresponding score. For each of the hand gestures in our database, 40 images are employed in creating the database and 30 images are used for the evaluation.

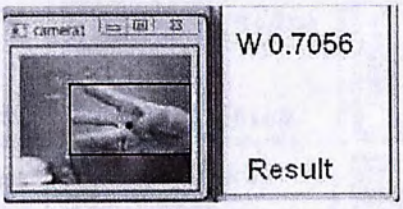


Figure 6.11: Interface developed for our hand gesture recognition system

6.6.2 Recognition Results

In this part, we first evaluate the algorithm by varying the hand orientation. A hand gesture 'U' is utilized and Fig. 6.12 shows the corresponding recognition results. It gives accurate recognitions under a significant variation in hand orientation.

Figure 6.12: The recognition results of a hand gesture with different orientations

Furthermore, we randomly select 15 hand gestures for testing. Some of the recognition results are presented in Fig. 6.13. The overall recognition results are reported in Table. 6.2. The overall success rate is 94.89% and the recognition time is about 0.4 seconds on a Pentium IV 2.4-GHz personal computer (PC).

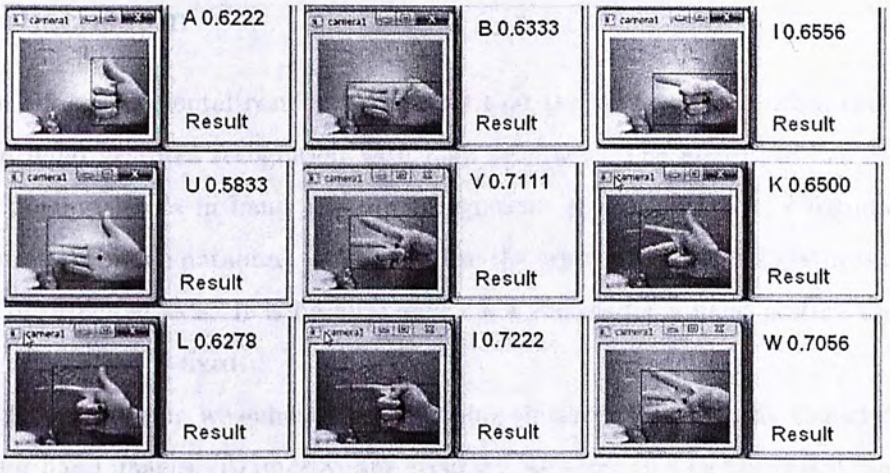


Figure 6.13: The recognition results of different hand gestures

For the hand gesture 'A', the success rate is below 90%. It is due to the variation in thumb orientation of gesture 'A' in testing images. It is observed that the area ratio between the thumb and palm is relative larger than other gestures. The thumb therefore contains a significant number of detail coefficients and its orientation gives a notable influence on the recognition accuracy.

Table 6.2: Recognition results of 15 hand gestures.

Sign	Correct	Sign	Correct	Sign	Correct
A	86.67%	H	96.67%	P	96.67%
B	90.00%	I	93.33%	T	96.67%
C	90.00%	J	90.00%	U	96.67%
D	96.67%	L	100%	W	96.67%
E	100%	O	100%	Y	93.33%

6.7 Discussion

From the experimental results, we can see that the proposed algorithm can perform real-time hand gestures recognition with high accuracy. The algorithm has considered three important issues in hand gestures recognition: size of database, recognition time, and accuracy. For the database, we standardize the orientation of hand gestures using the concept of principal axis. It is because only a few images for a hand gesture are needed once its orientation is fixed.

For the recognition, we enhance the searching efficiency by using the idea of codeword scheme for hand images. To improve the accuracy, we propose a measurement metric that uses the weighted L^2 norm with a penalty factor to emphasis the dissimilarity between the stored and test features.

6.8 Conclusion and Future Work

In this chapter, a new approach has been proposed to recognize hand gestures. To reduce the size of database, the orientations of hand gestures are first standardized by introducing the idea of principal axis. Truncation and quantization are then utilized to facilitate the compression power and recognition.

The concept of codeword scheme is then utilized to facilitate the searching algorithm. Recognition is performed through the proposed measurement metric which is a weighted L^2 norm with the use of a penalty factor.

Experiments are performed to evaluate the proposed recognition algorithm. The overall success rate is 94.89% and the recognition time is about 0.4 seconds. The results prove that our algorithm is fast, accurate, and has a small database size.

In the future, the proposed hand gestures recognition algorithm will be further improved. First of all, an algorithm in removing the front arm of a user during hand extraction process will be implemented. Instead of dividing the resized image into 4 portions, a method to disperse the detail coefficients non-uniformly will be developed. Codewords with a higher resolution will also be utilized to further increase the searching efficiency.

Chapter 7

Conclusion

The thesis has investigated five challenge problems in developing a palm-sized humanoid robot, PHR. They are: 1) mechanical design; 2) gait synthesis; 3) gait planning; 4) balance control; and 5) human-robot interaction. In this chapter, We first summarize our research. A briefly discussion on some potential future works for our research is then followed.

7.1 Research Summary

First, a summary of the key contributions of this research is illustrated.

In Chapter 1, we firstly describe the motivation of developing PHR. After that, we provide a summary of the related works to this research topic. Lastly, we present the structure of this thesis.

In Chapter 2, we describe the design of PHR in three aspects. We firstly present the mechanical design of PHR such as the joint configuration and the function of transformation. We then describe the hardware platform of PHR including the actuation, sensory system, driving system, and vision system. The software platform is finally illustrated.

In Chapter 3, we study the kinematics of PHR including the ways in solving the forward and inverse kinematics.

In Chapter 4, we discuss the algorithm for the gait synthesis of PHR. The differences between human and robot joints are first presented. Then we describe the types of gait for humanoid robots. After that, the concept of gait frames is presented. We finally express the approach of Bezier curve interpolation for the generation of gait trajectories. Some experimental results are listed to show the generated gait pattern and the corresponding joint angles.

In Chapter 5, we describe a balance control scheme for PHR. We propose the use of a geometric method to analyze the influence of tilting along the Coronal and Sagittal planes. Based on the sensory information, we then apply the 3-D inverse kinematics to compute the tilting-compensated action.

In Chapter 6, we present the hand gestures recognition system. It is designed for the human-robot interaction. We utilize the concept of principal axis to standardize the orientations of hand gestures. With the use of truncation and quantization on Haar wavelet coefficients, size of database can be highly reduced after zlib compression. To improve the searching efficiency, we propose the idea of codeword assignment. For the recognition, we develop a measurement metric using weighted L^2 norm with a penalty term. From the experimental results, we can conclude that the proposed recognition system has three contributions, a) small database, b) fast, and c) accurate.

The contributions of this thesis can be concluded as follows:

- Design and build a foldable humanoid robot with almost the half of both size and weight of many entertainment robots.
- Design a new configuration of joints for transformation which is innovative in existing humanoid robot.
- Implement a user interface which allows a user to design and simulate different motions before testing on PHR.
- Apply the concept of 3-D forward and inverse kinematics in motion planning.
- Generate gait trajectories for PHR which are local controllable using the concept of gait frames with the Bezier curves interpolation .

- Develop a geometric method based balance algorithm for PHR which can be generalized to other humanoid robots with a similar joint configuration.
- Present a Haar wavelet based hand gestures recognition system which is fast, accurate, and has a small database.

In summary, most of these research works are important topics in promoting the development of mini-sized humanoid robots.

7.2 Future Work

The development of a mini-sized humanoid robot is a challenge topic for entertainment based robots. Until now, there are still many new research areas to be investigated. This thesis proposes a palm-sized humanoid robot with detail studies in several important aspects such as mechanical design, motion planning, and human-robot interaction. The future works about this topic are expected to mainly focus on the three parts, a) joint structure, b) intelligent systems, and c) battery.

As aforementioned, we have discussed the differences in joint structure between humanoid robots and human beings. With a different nature of actuation, it is hard for a humanoid robot to perform exactly what humans can do. In addition to joint configuration, we can imagine that a new generation of humanoid robots shall therefore be designed with a similar nature of actuation as humans.

In this thesis, we focus in the development of a humanoid robot with some basic aspects like motion planning and humanoid-robot interaction. Since PHR is designed to be entertaining, it shall therefore possess some intelligent systems such as emotion and voice systems which play the role in enhancing the human-robot interaction.

Appendix A

Forward Kinematics of PHR

In this chapter, the forward kinematics of the left body of PHR is presented. Fig. A.1 shows the joint configuration of PHR and the robot coordinate frame.

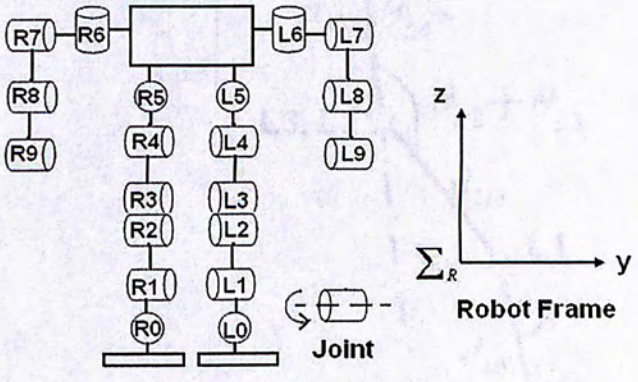


Figure A.1: The joint configuration PHR

A.1 Lower Limb

In Chapter 3, the computation of the Cartesian coordinates of joints in the right leg is presented. We now focus on the left leg of PHR. Fig. A.2 shows the left leg of PHR with a positive angle of rotation ($0 \leq \theta_{L5} \leq 90$).

For the left leg of PHR, let the coordinate of joint L_5 with respect to the world frame

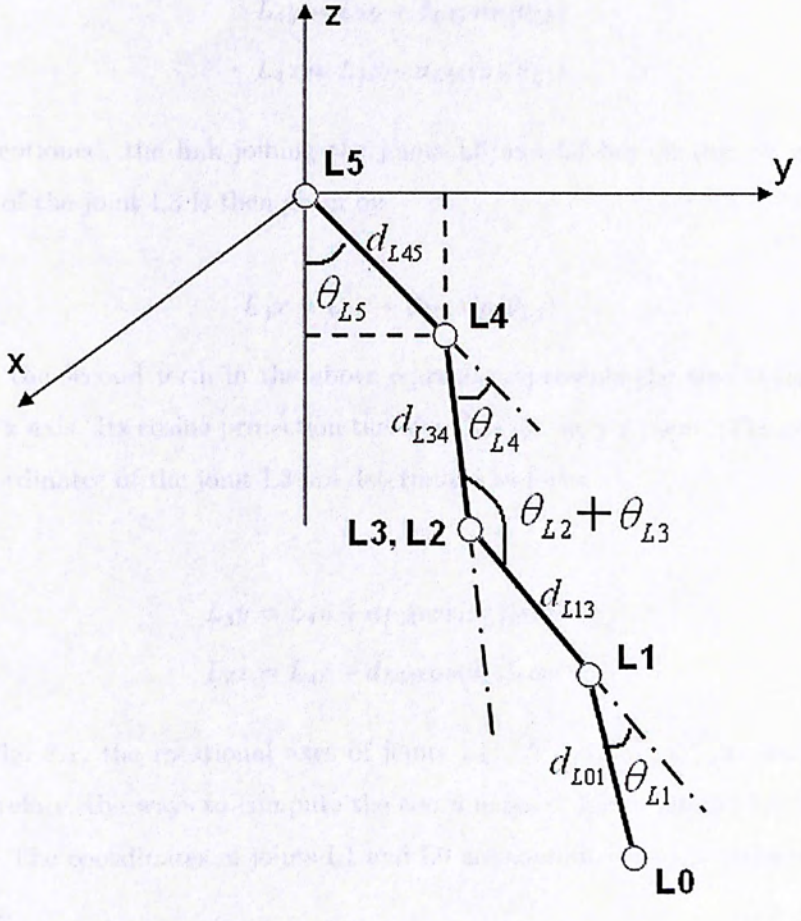


Figure A.2: The left leg of PHR with a positive angle of rotation θ_{L5}

be (L_5x, L_5y, L_5z). Based on the second constraint, the link joining the joints L5 and L4 lies on the y-z plane. The coordinate of joint L4 is then given by

$$L_4x = L_5x \tag{A.1a}$$

$$L_4y = L_5y + d_{L45}sin(\theta_{L5}) \tag{A.1b}$$

$$L_4z = L_5z - d_{L45}cos(\theta_{L5}) \tag{A.1c}$$

Aforementioned, the link joining the joints L5 and L4 lies on the y-z plane, the x-coordinate of the joint L3 is then given by

$$L_3x = L_4x + d_{L34}sin(\theta_{L4}) \tag{A.2}$$

, where the second term in the above equation represents the sine projection of the link on the x axis. Its cosine projection therefore lies on the y-z plane. The corresponding y and z coordinates of the joint L3 are determined as follows

$$L_3y = L_4y + d_{L34}cos(\theta_{L4})sin(\theta_{L5}) \tag{A.3a}$$

$$L_3z = L_4z - d_{L34}cos(\theta_{L4})cos(\theta_{L5}) \tag{A.3b}$$

From Fig. 3.1, the rotational axes of joints L1, L2, L3, and L4 are parallel to each other. Therefore, the ways to compute the coordinates of joints L0 and L1 are similar to that of L3. The coordinates of joints L1 and L0 are computed by the following equations respectively.

$$L_1x = L_3x + d_{L13}sin(-180 + \theta_{L2} + \theta_{L3} + \theta_{L4}) \tag{A.4a}$$

$$L_1y = L_3y + d_{L13}cos(-180 + \theta_{L2} + \theta_{L3} + \theta_{L4})sin(\theta_{L5}) \tag{A.4b}$$

$$L_1z = L_3z - d_{L13}cos(-180 + \theta_{L2} + \theta_{L3} + \theta_{L4})cos(\theta_{L5}) \tag{A.4c}$$

$$L_0x = L_1x + d_{L01}sin(-180 + \theta_{L1} + \theta_{L2} + \theta_{L3} + \theta_{L4}) \quad \text{(A.5a)}$$

$$L_0y = L_1y + d_{L01}cos(-180 + \theta_{L1} + \theta_{L2} + \theta_{L3} + \theta_{L4})sin(\theta_{L5}) \quad \text{(A.5b)}$$

$$L_0z = L_1z - d_{L01}cos(-180 + \theta_{L1} + \theta_{L2} + \theta_{L3} + \theta_{L4})cos(\theta_{L5}) \quad \text{(A.5c)}$$

A.2 Upper Limb

In Chapter 3, we present the computation of the Cartesian coordinates of joints in the right arm. We now focus on the left arm. Fig. A.3 shows the left arm of PHR with a positive angle of rotation ($0 \leq \theta_{L6} \leq 90$).

For the left arm of PHR, let the coordinate of joint L_6 with respect to the world frame be (L_6x, L_6y, L_6z). Based on the second constraint, the link joining the joints L6 and L7 lies on the x-y plane. The coordinate of joint L7 is then given by

$$L_7x = L_6x + d_{L67}sin(\theta_{L6}) \quad \text{(A.6a)}$$

$$L_7y = L_6y + d_{L67}cos(\theta_{L6}) \quad \text{(A.6b)}$$

$$L_7z = L_6z \quad \text{(A.6c)}$$

Aforementioned, the link joining the joints L6 and L7 lies on the x-y plane, the z-coordinate of the joint L8 is then given by

$$L_8z = L_7z - d_{L78}cos(\theta_{L7}) \quad \text{(A.7)}$$

, where the second term in the above equation represents the cosine projection of the link on the z axis. Its sine projection therefore lies on the x-y plane. The corresponding x and y coordinates of the joint L8 are determined as follows

$$L_8x = L_7x + d_{L78}sin(\theta_{L7})cos(\theta_{L6}) \quad \text{(A.8a)}$$

$$L_8y = L_7y - d_{L78}sin(\theta_{L7})sin(\theta_{L6}) \quad \text{(A.8b)}$$

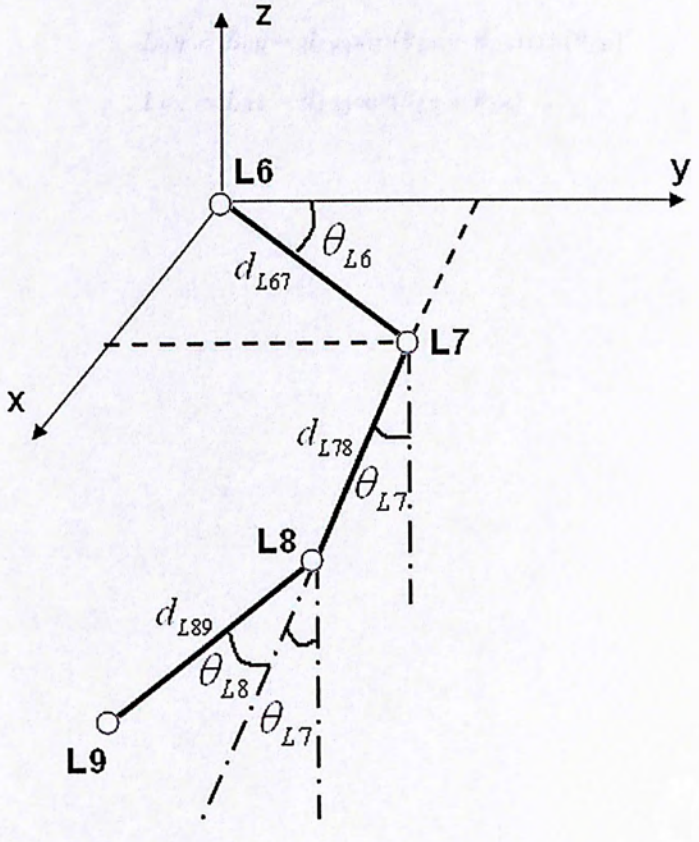


Figure A.3: The left arm of PHR with a positive angle of rotation θ_{L6}

From Fig. 3.1, the rotational axes of joints L7, and L8 are parallel to each other. Therefore, the way to compute the coordinates of joint L9 is similar to that of L8. The coordinate of joints L9 is computed by the following equations.

$$L_9x = L_8x + d_{L89}sin(\theta_{L7} + \theta_{L8})cos(\theta_{L6}) \quad \text{(A.9a)}$$

$$L_9y = L_8y - d_{L89}sin(\theta_{L7} + \theta_{L8})sin(\theta_{L6}) \quad \text{(A.9b)}$$

$$L_9z = L_8z - d_{L89}cos(\theta_{L7} + \theta_{L8}) \quad \text{(A.9c)}$$

Appendix B

Inverse Kinematics of PHR

In this chapter, the inverse kinematics of the left body of PHR is presented. Fig. B.1 shows the joint configuration of PHR and the robot coordinate frame.

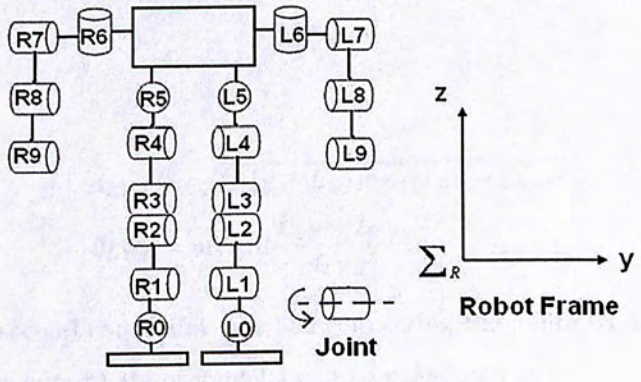


Figure B.1: The joint configuration PHR

B.1 Lower Limb

In this section, the inverse kinematics of the left leg of PHR is presented. Fig. B.2 shows the front view of the left leg.

From the figure, the Y-Z distance d_{LYZ} between the joints L0 and L5 and the tilt angle θ_{LYZ} on the y-z plane can be formulated as

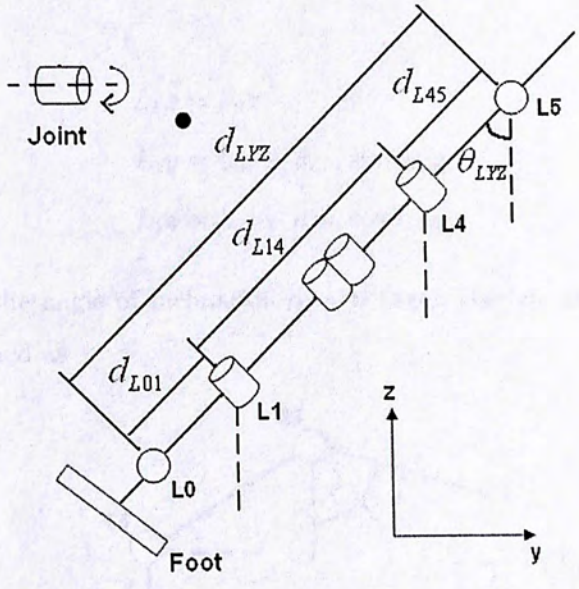


Figure B.2: The front view of the left leg of PHR

$$d_{LYZ} = \sqrt{(L_0y - L_5y)^2 + (L_0z - L_5z)^2} \tag{B.1a}$$

$$\theta_{LYZ} = \arcsin(\frac{L_0y - L_5y}{d_{LYZ}}) \tag{B.1b}$$

Based on the second constraint, the line connecting the joints L4 and L5 is parallel to the y-z plane. The coordinate of joint L4 can be calculated as

$$L_4x = L_5x \tag{B.2a}$$

$$L_4y = L_5y - d_{L45}\sin\theta_{LYZ} \tag{B.2b}$$

$$L_4z = L_5z - d_{L45}\cos\theta_{LYZ} \tag{B.2c}$$

Similarly, the line connecting the joints L0 and L1 is parallel to the y-z plane, coordinate of L1 can be calculated as

$$L_1x = L_0x \tag{B.3a}$$

$$L_1y = L_0y - d_{L01}\sin\theta_{LYZ} \tag{B.3b}$$

$$L_1z = L_0z - d_{L01}\cos\theta_{LYZ} \tag{B.3c}$$

From Fig. B.3, the angle of inclination α_{YZ} between the left and right hips (L5and R5) can be determined as

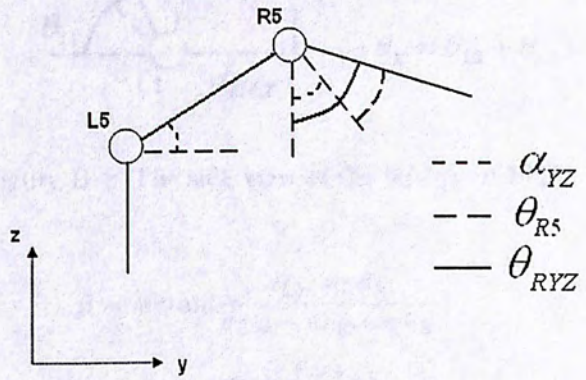


Figure B.3: The front view of the hips of PHR

$$\alpha_{YZ} = \arctan(\frac{L_5z - R_5z}{L_5y - R_5y}) \tag{B.4}$$

The joint angle L5 is then given by

$$\theta_{L5} = \theta_{LYZ} + \alpha_{YZ} \tag{B.5}$$

From Fig. B.4, the joint angles of L1, L2, L3, and L4 are given as follows

$$d_{L14} = d_{LYZ} - d_{L45} - d_{L01} \tag{B.6}$$

$$d_{L14X} = L_4x - L_1x \tag{B.7}$$

$$\theta_K = \arccos(\frac{d_{L12}^2 + d_{L34}^2 - d_{L14X}^2 - d_{L14}^2}{2d_{L12}d_{L34}}) \tag{B.8}$$

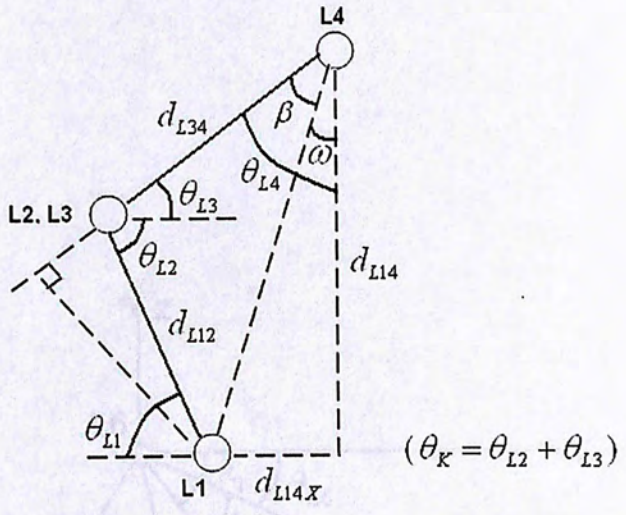


Figure B.4: The side view of the left leg of PHR

$$\beta = \arctan\left(\frac{d_{L12}\sin\theta_K}{d_{L34} - d_{L12}\cos\theta_K}\right) \tag{B.9}$$

$$\omega = \arctan\left(\frac{d_{L14X}}{d_{L14}}\right) \tag{B.10}$$

$$\theta_{L4} = \omega + \beta \tag{B.11a}$$

$$\theta_{L3} = 90 - \theta_{L4} \tag{B.11b}$$

$$\theta_{L2} = \theta_K - \theta_{L3} \tag{B.11c}$$

$$\theta_{L1} = \theta_{L2} \tag{B.11d}$$

B.2 Upper Limb

In this section, the inverse kinematics of the left arm of PHR is presented. Fig. B.5 shows the front view of the left arm after the compensation of hip inclination.

The joint angle θ_{R6} is given by

$$\theta_{L6} = \lambda_1 - \lambda_2 \tag{B.12}$$

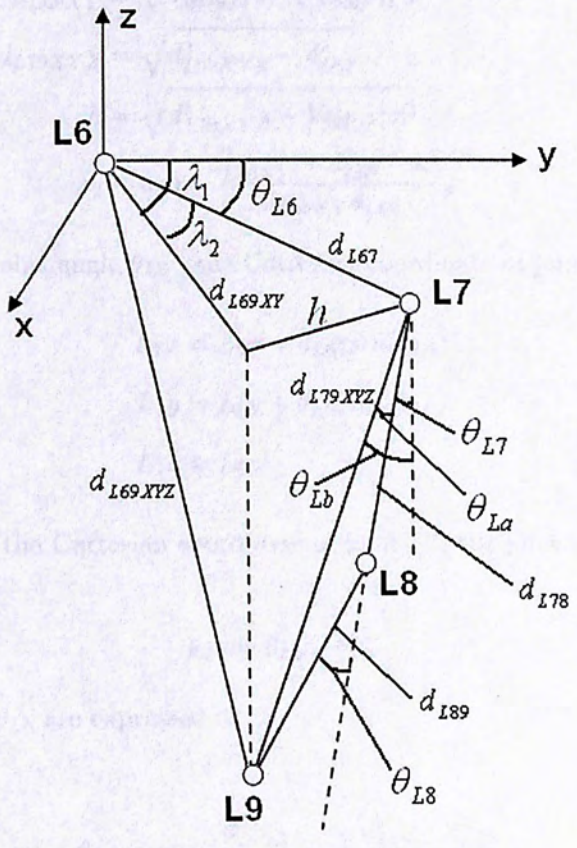


Figure B.5: The front view of the left arm of PHR

, where λ_1 and λ_2 are expressed as

$$\lambda_1 = atan(\frac{V_{69ROT}x)}{V_{69ROT}y} \tag{B.13a}$$

$$d_{L69XYZ} = \sqrt{V_{69ROT}x^2 + V_{69ROT}y^2 + V_{69ROT}z^2} \tag{B.13b}$$

$$d_{L69XY} = \sqrt{V_{69ROT}x^2 + V_{69ROT}y^2} \tag{B.13c}$$

$$d_{L79XYZ} = \sqrt{d_{L69XYZ}^2 - d_{L67}^2} \tag{B.13d}$$

$$h = \sqrt{d_{L79XYZ}^2 - V_{69ROT}z^2} \tag{B.13e}$$

$$\lambda_2 = acos(\frac{d_{L69XY}^2 + d_{L67}^2 - h^2}{2d_{L69XY}d_{L67}}) \tag{B.13f}$$

By utilizing the joint angle θ_{L6} , the Cartesian coordinate of joint L7 is given by

$$L_7x = L_6x + d_{L67}sin(\theta_{L6}) \tag{B.14a}$$

$$L_7y = L_6y + d_{L67}cos(\theta_{L6}) \tag{B.14b}$$

$$L_7z = L_6z \tag{B.14c}$$

After computing the Cartesian coordinate of joint L7, the joint angle θ_{L7} is given by

$$\theta_{L7} = \theta_{Lb} - \theta_{La} \tag{B.15}$$

, where θ_{La} and θ_{Lb} are expressed as

$$\theta_{La} = acos(\frac{d_{L79XYZ}^2 + d_{L78}^2 - d_{L89}^2}{2d_{L79XYZ}d_{L78}}) \tag{B.16a}$$

$$\theta_{Lb} = acos(\frac{L_7z - L_9z}{d_{L79XYZ}}) \tag{B.16b}$$

By considering the triangle formed by joints L7, L8, and L9, θ_{L8} can then be computed by

$$\theta_{L8} = 180 - acos(\frac{d_{L78}^2 + d_{L89}^2 - d_{L79XYZ}^2}{2d_{L78}d_{L89}}) \tag{B.17}$$

Appendix C

Zero Moment Point

In this chapter, the concept of Zero Moment Point (ZMP) is described. It was first introduced for the stability measurement of humanoid robots in 1969 by Miomir Vukobratovic at The Third All-Union Congress of Theoretical and Applied Mechanics in Moscow. It is a concept related to the dynamics and control of legged locomotion [24]. ZMP indicates the point with respect to which the dynamic reaction force at the contact between foots and a ground does not create any moment. In other words, the point where sum of the inertia force equals zero. Based on this concept, many impressive researches are achieved [66]-[56].

For a given posture of PHR, the corresponding ZMP can be computed as [64]

$$x_{ZMP} = \frac{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)x_i - \sum_{i=1}^{n} m_i\ddot{x}_i z_i - \sum_{i=1}^{n} I_{iy}\ddot{\Omega}_{iy}}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{C.1a}$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)y_i - \sum_{i=1}^{n} m_i\ddot{y}_i z_i - \sum_{i=1}^{n} I_{ix}\ddot{\Omega}_{ix}}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{C.1b}$$

where

- m_i is the mass of link i,
- I_{ix} and I_{iy} are inertial components at the center of the gravity of link i around the x and y-axis respectively,

- $\ddot{\Omega}_{ix}$ and $\ddot{\Omega}_{iy}$ are absolute angular velocity components around the x-axis and y-axis at the center of the gravity of link i respectively,
- g is the gravitational acceleration,
- $(x_{ZMP}, y_{ZMP}, 0)$ is the coordinate of the ZMP,
- (x_i, y_i, z_i) is the coordinate of the mass center of link i on an absolute Cartesian coordinate system

For the gait of PHR, the angular and linear accelerations $\ddot{\Omega}_{ix}$, $\ddot{\Omega}_{iy}$, $\ddot{x}_i$, $\ddot{y}_i$, and $\ddot{z}_i$ are small and have little effect on computing the ZMP position. They can then be neglected in the above equations. Therefore, the static stability of PHR is derived by

$$x_{CoG} = \frac{\sum_{i=1}^{n} m_i g x_i}{\sum_{i=1}^{n} m_i g} = \frac{\sum_{i=1}^{n} m_i x_i}{\sum_{i=1}^{n} m_i} \tag{C.2a}$$

$$y_{CoG} = \frac{\sum_{i=1}^{n} m_i g y_i}{\sum_{i=1}^{n} m_i g} = \frac{\sum_{i=1}^{n} m_i y_i}{\sum_{i=1}^{n} m_i} \tag{C.2b}$$

(C.2c)

Appendix D

User Interface of PHR

In this chapter, we describe the user interface for PHR. For the interface, we aim to implement some new functions in order to facilitate the use of PHR. This task is to develop a user interface for PHR which aims to provide a platform for the design and stimulation of different motions. Some basic functions are also included such as zooming, 2-D panning, and 3-D rotation. Fig D.1 shows the layout of the interface.

From the figure, we can see that the interface can be viewed as four regions. Firstly, the upper left region shows a 3-D model of PHR. It aims to display the simulated results of PHR corresponding to different joint angles (left of Fig. D.2). In order to change the viewing point and angle of the model, some functions are provided in the bottom left of the interface. They are 2-D panning, zooming and 3-D rotation of the PHR model (right of Fig. D.2).

On the right side of the PHR model, they represent functions for motions design and simulation (Fig. D.3).

For the motion design process, a user can first select a joint by clicking it in the PHR model. A red color is used to show the current selected joint. The selected joint can then be turned by using the status bar in Fig. D.3. The PHR model in the interface will display the turning effect in real-time (Fig. D.4). The design process is frame based. It means that a user needs to save each motion as a sequence of motion frames (similar to the concept of video frames in a video). A set of motion frames therefore represents a motion video

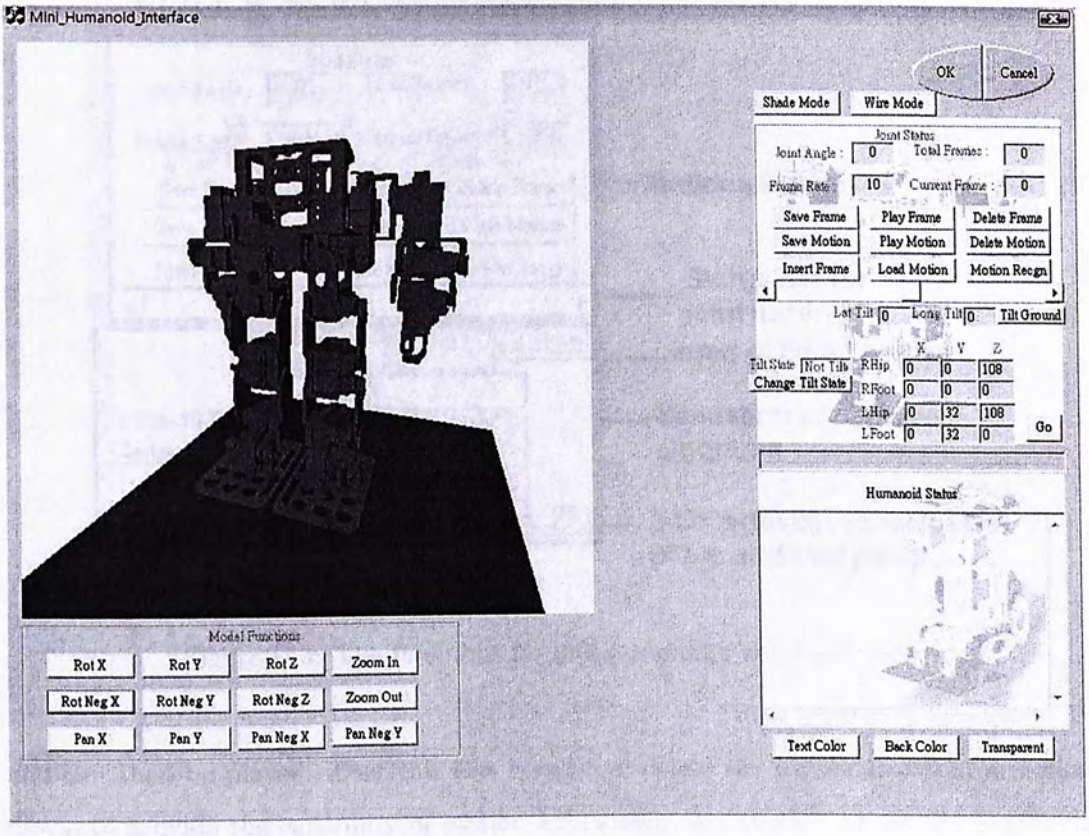


Figure D.1: The user interface of PHR

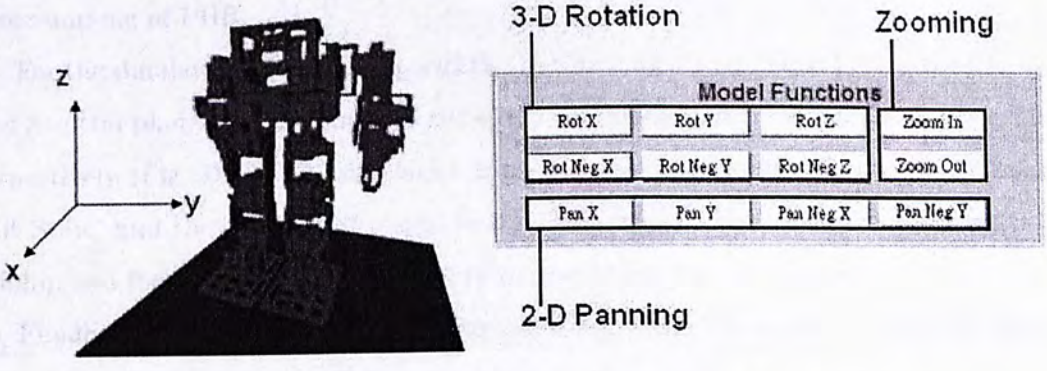


Figure D.2: The 3-D model of PHR and the viewing functions

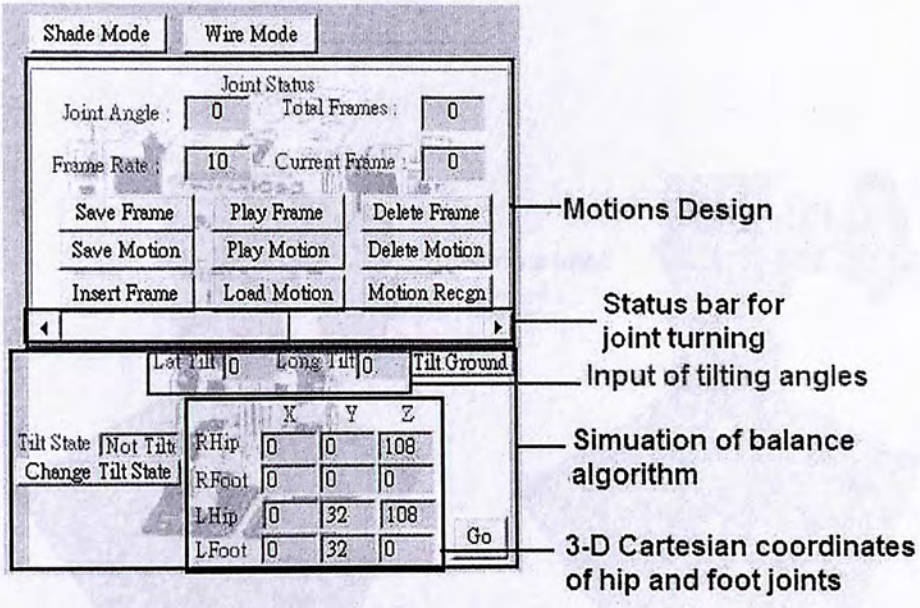


Figure D.3: The functions for motions design and simulation

and can then be played. User can also insert and delete the frames stored in a motion movie to provide the capability of motion adjustment. By continuing the movie playing and modifying process, the finalized joint angles in each motion frame can be transmitted into the controller board through serial port. This user interface can actually be treated as a virtual environment for motion design. It is an innovative and practical way for the programming of PHR.

For the simulation of balance algorithm, the tilting angles (in degree) along the Coronal and Sagittal planes are first input in the edit boxes named with 'Lat Tilt' and 'Long Tilt' respectively (Fig. D.3). The simulation is then started by selecting the button, 'Change Tilt State' and the result is displayed in the PHR model. The Cartesian coordinates of the hip and foot joints will be updated in the corresponding edit boxes.

Finally, the bottom right region of the interface shows the status of the PHR model (Fig. D.5). A text message will for example be displayed in the window once a user selects a joint in the PHR model.

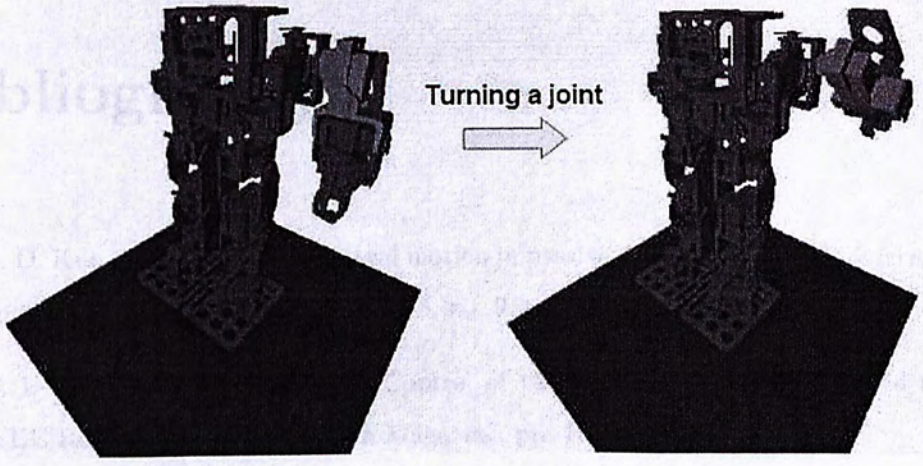


Figure D.4: The effect on the PHR model by turning a status bar

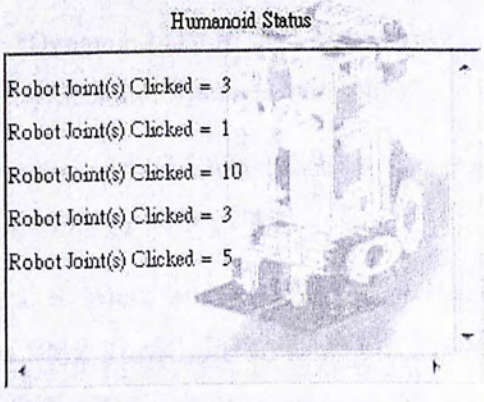


Figure D.5: The joints status of PHR

Bibliography

[1] A. D. Kuo, "Stabilization of lateral motion in passive dynamic walking", *International Journal of Robotics Research*, vol. 18, no. 9, pp. 917-930, 1999.

[2] A. L. Kun, and W. T. Miller, "Control of variable-speed gaits for a biped robot", IEEE Robotics and Automation Magazine, pp. 19-29, 1999.

[3] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast Multiresolution Image Querying", Proceedings in ACM SIGGRAPH Conference on Computer Graphics, pp. 277-286, 1995.

[4] C. Zhou, "Nero-fuzzy gait synthesis with reinforcement learning for a biped walking robot", Soft Computing, Vol. 4, pp. 238-250, 2000.

[5] C. Zhou, Q. Meng, "Dynamic balance of a biped robot using fuzzy reinforcement learning agents", Fuzzy Sets and Systems, Vol. 134, No. 1, pp.169-187, 2003.

[6] C. Zhou, "Robot learning with GA-based fuzzy reinforcement learning agents", Information Sciences, vol.145, pp.45-68, 2002.

[7] D. C. Yang, L. Liu, J. S. Wang, and K. Chen, "Gait Planning of Humanoid Robot Based on Anticipant ZMP Track", Proceedings of IWBRT'2001(2001 International Workshop on Bio-Robotics and Teleoperation), pp 168-172, 2001.

[8] D. Katic, M. Vukobratovic, "Intelligent control of humanoid robots using neural networks", 7th Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2004, pp.31-35, 2004.

[9] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin, "Wavelets for Computer Graphics: A Primer Part 1," IEEE Computer Graphics and Applications, Vol.15, No.3, pp.76-84, 1995.

[10] F. M. Silva, and J. A. T. Machado, "Energy analysis during biped walking", Proceedings in the IEEE International Conference on Robotics and Automation, Detroit, Michigan, pp.59-64, 1999.

[11] G. Capi, M. Yokota, and K. Mitobe, "A new humanoid robot gait generation based on multiobjective optimization", Proceedings in the IEEE International Conference on Advanced Intelligent Mechatronics, pp.450-454, 2005.

[12] G. Capi, and M. Yokota, "Optimal multi-criteria humanoid robot gait synthesis - an evolutionary approach", International Journal of Innovative Computing Information and Control, Vol. 2, pp.1249-1258, 2006.

[13] G. Gomez, M. Sanchez, and L. E. Sucar, "On selecting an appropriate colour space for skin detection", Proceedings of Mexican International Conference on Artificial Intelligence, Yucatan, pp. 69-78, 2002.

[14] H. Brashear, T. Starner, P. Lukowicz, H. Junker, "Using multiple sensors for mobile sign language recognition", Proceedings of IEEE International Symposium on Wearable Computers, pp. 45-52, October 2003.

[15] http://en.wikipedia.org/wiki/Bezier_curve

[16] http://en.wikipedia.org/wiki/Coronal_plane

[17] http://en.wikipedia.org/wiki/Data_compression_ratio

[18] http://en.wikipedia.org/wiki/DEFLATE

[19] http://en.wikipedia.org/wiki/JPEG

[20] http://en.wikipedia.org/wiki/Morphological_image_processing

[21] http://en.wikipedia.org/wiki/QRIO

[22] http://en.wikipedia.org/wiki/Restoring_force

[23] http://en.wikipedia.org/wiki/Transverse_plane

[24] http://en.wikipedia.org/wiki/Zero_Moment_Point

[25] http://en.wikipedia.org/wiki/Zlib

[26] http://local.wasp.uwa.edu.au/ pbourke/surfaces_curves/bezier/

[27] http://people.csail.mit.edu/bkph/courses/papers/RoVis/ch3_BI_geometric.pdf

[28] http://world.honda.com/ASIMO/history/technology.html

[29] http://world.honda.com/ASIMO/history/technology1.html

[30] http://www.androidworld.com/prod06.htm

[31] http://www.freescale.com/files/sensors/doc/fact_sheet/MMA7260QFS.pdf

[32] http://www.kawada.co.jp/global/ams/hrp_2.html

[33] http://www-personal.umich.edu/ artkuo/Passive_Walk/passive_walking.html

[34] http://www.plyojump.com/qrio.html

[35] http://www.sparkfun.com/datasheets/Accelerometers/MMA7260Q-Rev1.pdf

[36] http://www.tsplines.com/resources/class_notes/Bezier_curves.pdf

[37] http://www.zlib.net/apps.html

[38] http://www.zmp.co.jp/e-nuvo/data/e-nuvo_presentation_20051107.pdf

[39] H. O. Lim, Y. Yamamoto, and A. Takanishi, "Control to realize human-like walking of a biped humanoid robot", Systems, Man, and Cybernetics, 2000 IEEE International Conference, vol 5, pp.3271-3276, 2000.

[40] J. Allen, P. Asselin, and R. Foulds , "American Sign Language Finger Spelling Recognition System", Proceedings of Bioengineering Conference, March, 2003.

[41] J. H. Park, and M. S. Choi, "Generation of an optimal gait trajectory for biped robots using a genetic algorithm", JSME International Journal, Series C, Vol. 47, No.2, pp.715-721, 2004.

[42] J. H. Shin, J. S. Lee, S. K. Kil, D. F. Shen, J. G. Ryu, E. H. Lee, H. K. Min, and S. H. Hong, "Hand Region Extraction and Gesture Recognition using entropy analysis", Proceedings of International Journal of Computer Science and Network Security, Vol. 6 No. 2 pp. 216 222, February 2006.

[43] J. Y. Yu, "The Desingn Implementation and Plan Trajectory of a Biped Walking Robot", Department of Mechanical Engineering, Master Thesis, 2004.

[44] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K.Kaneko, K. Yokoi, and H. Hirukawa, "Real-Time Planning of Humanoid Robot's Gait for Force Controlled Manipulation", Proceedings of the First IEEE-RAS Int.International Conference on Robotics and Automation, pp. 3075-3080, New Orleans, LA, 2004.

[45] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot", Proceedings in the International Conference on Robotics and Automation, Leuven, Belgium, pp.1321-1326, 1998.

[46] K. H. Park, J. H. Jo, and J. H. Kim, "Balance Control of a Humanoid Robot Using the Inclinometer", Proceedings of 1st International Conference on Sensing Technology, pp. 72-77, 2005.

[47] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid Motion Generation System on HRP2-JSK for Daily Life Environment", Proceedings of International Conference on Mechatronics and Automation, pp.1772-1777, 2005.

[48] L. Roussel, C. Canudas-de-Wit and A. Goswami, "Generation of energy optimal complete gait cycles for biped robots", Proceedings in the IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp.2036-2041, 1998.

[49] L. Sentis, O. Khatib, "Task-Oriented Control of Humanoid Robots Through Prioritization", Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots, Santa Monica, USA, November 2004.

[50] M. Lamar and M. Bhuiyant, "Hand Alphabet Recognition Using Morphological PCA and Neural Networks", Proceedings of International Joint Conference on Neural Networks, pp. 2839-2844, Washington, USA, 1999.

[51] M. Vukobratovic and D. Juricic, "Contribution to the Synthesis of Biped Gait", Bio-Medical Engineering, *IEEE Trans. on Biomedical Engineering*, vol. BME-16, no. 1, pp. 1-6, 1969.

[52] P. F. Chen, and T. Y. Li, "Generating Humanoid Lower-Body Motions with Real-time Planning", Proceedings of 2002 Computer Graphics Workshop(CG2002) , Taiwan, June, 2002.

[53] R. C. Gonzalez, R. E. Woods, and S. L. Eddinss. Digital Image Procesing Using Matlab. Beijing: Publishing House of Electronics Industry, 2004, pp.334-377.

[54] S. Ha, Y. Han, and H. Hahn, "Adaptive Gait Pattern Generation of Biped Robot based on Humans Gait Pattern Analysis", Proceedings of International Journal of Mechanical Systems Sciene and Engineering, Vol. 1, No. 2, ISSN 1307-7473, 2008.

[55] S. Hollar, K. S. J. Pister, "Wireless Static Hand Gesture Recognition with Accelerometers - The Acceleration Sensing Glove", Technical Report, University of California, Berkley, June 2000.

[56] T. Furuta, T. Tawara, Y. Okumura, M. Shimizu, and K. Tomiyama, "Design and construction of a series of compact humanoid robots and development of biped walk control strategies", Robotics and Autonomous Systems, Vol. 37, pp.81-100, 2001.

[57] T. McGeer, "Passive dynamic walking", *International Journal of Robotics Research*, vol. 9, no. 2, pp. 62-82, 1990.

[58] T. McGeer, "Passive walking with knees", *IEEE Trans. on Robotics and Automation Conference*, pp. 1640-1645, 1990.

[59] T. Takenaka, "The control system for the Honda humanoid robot", Oxford Journals on Age and Ageing 35, pp.24-26, 2006.

[60] Y. Hasagawa, T. Arakawa, and T. Fukuda, "Trajectory generation for biped locomotion robot", Mechatronics, Vol. 10, pp.67-89, 2000.

[61] Y. Hsu, "Adaptive-Network-based Fuzzy Inference System apply to biped robot's zero-moment-point", Opto-Mechatronics Application Laboratory, Master Thesis, 1995.

[62] Y. T. Su, C. H. Kuo, C. Y. Chen, C. L. Hsu, M. F. Lu, and T. H. S. Li, "Design and Implementation of Fuzzy Auto-Balance Control for Humanoid Robot", FIRA RoboWorld Congress, USA, 2007.

[63] Y. Zheng and J. Shen, "Gait synthesis for the SD-2 biped robot to climb sloped surface", *IEEE Trans. on Robotics and Automation Conference*, vol. 6, no. 1, pp. 86-96, 1990.

[64] Z. Liu, K. Chen, and L. Liu, "A Uniform Gait Generation Method for RoboCup Humanoid Robot Based on Genetic Algorithms", Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, pp.1620-1625, June, 2006.

[65] Z. Tang, C. Zhou, Z. Sun, "Gait planning for soccer playing humanoid robots", Lecture Notes in Control and Information Sciences, Springer-Verlag, 299, pp.241-262, 2004.

[66] Z. Tang, Z. Sun, and C. Zhou, "GA Based Optimization for humanoid walking", ICGST-ARAS International Journal on Automation, Robotics and Autonomous System, Vol. 5, pp.1-10, 2006.