The University of Western Australia
Department of Electrical and Electronic Engineering

Honours Thesis

Bipedal Dynamic Walking in Robotics

Elliot Nicholls

October 26, 1998

in loving memory of Papa

Abstract

It is easier for bipedal robots to exist in a human oriented environment than for other types of robots. Furthermore, dynamic walking is more efficient than static walking. For a biped robot achieve dynamic balance while walking, a dynamic gait must be developed. Two different approaches to gait generation are presented—an intuitive approach using software for gait animation, and a periodic approach that provides a scalable gait with parameters for controlling step length, step height and step period.

A biped robot also requires a control system to ensure the stability of the robot while walking. Various simple control techniques were tried—proportional control and proportional integral control systems were implemented to modify trunk motion in order to compensate for lower limb movement.

A seven link biped robot with human proportions was designed and constructed, to apply the developed gaits and control systems. Several experiments were conducted to examine the stability of the developed gaits. The ability of the control systems to stabilise and balance the robot while stationary and in motion was trialled.

It was found that control using the intuitive approach to gait generation is problematic, due to difficulty scaling the gait. With this gait the robot successfully took three steps. The periodic function approach to gait generation is scalable, but slow to implement directly. Simple control systems allowed the robot to balance sufficiently while standing in an upright stationary position. However, these control systems were insufficient to stabilise the robot while walking.

Reasons for success or failure of the gait generation methods and control systems are discussed and analysed. Future research methods are proposed to improve upon the theories developed in this dissertation.

Acknowledgements

Firstly, I would like to thank Thomas Bräunl for allowing me to perform research in the area bipedal robotics, and for supervising my work. I would like to thank the Departmental General Workshop and Richard Meager in particular for help with structural design and for skilled construction of the biped robot used in experiments. The staff of the Departmental Electronic Workshop were also very helpful, and I thank them also.

In addition, I would like to thank Joon Ng for many discussions on ideas and concepts. Special thanks to Voon-Li Chung for proofing this dissertation many times, and to Daniel Harvey and Chris deSilva for their constructive comments and suggestions.

I would especially like to thank my Mother, Father, and family, for all their support, love and understanding during this year. Special thanks to Leonie, who believed in me even when I didn't. Without them the difficulties would have been much greater, and this thesis would not have been written.

Thanks also to Peter Gifford and to my homegroup, whose support and prayers throughout the year have kept me going. I would also like to thank my friends for all their support over the past year. Thanks to the CIIPS people, with whom I had many late nights (and early mornings) while writing and experimenting.

Finally, I would like to give thanks to The Author, who has sustained me, and who has told the real story.

Contents

Abstract i

Acknowledgements iii

1 Introduction 1

1.1 Why study legged locomotion? 2

1.2 Dynamic and static walking 3

1.3 Nomenclature 4

1.4 Research objectives 5

1.5 Overview 5

2 Literature review 7

3 Design 13

3.1 General design 13

3.2 Design process 14

3.2.1 Balance 14

3.2.2 Forces, moments and torque 15

3.2.3 Proportions and mass distribution 16

3.2.4 Strength 17

3.3 Final design 18

3.4 Construction 18

3.4.1 Required design modifications—center of mass 18

4 Microcontroller architecture and sensor hardware 21

4.1 Controller architecture 21

4.1.1 Hardware description 21

4.1.2 Software development process 23

4.2 Sensors 24

4.2.1 Acceleration sensors 24

4.2.2 Foot sensors . . . 26
4.3 Testing and calibration . . . 26
4.3.1 Servos . . . 27
4.3.2 Acceleration sensors . . . 27

5 Gait development 29
5.1 Intuitive approach . . . 30
5.2 Periodic function approach . . . 31
5.2.1 Foot trajectory . . . 31
5.2.2 Inverse kinematics . . . 33
5.2.3 Application of periodic function approach . . . 35
5.2.4 Gait parameters . . . 36

6 Gait Generator software 39
6.1 Program description . . . 39
6.2 Program requirements . . . 40
6.3 Program evaluation . . . 41
6.4 System limitations . . . 41

7 Motion control 43
7.1 Control problems . . . 44
7.1.1 Trunk motion in the lateral plane . . . 44
7.1.2 Body attitude . . . 44
7.2 Control method . . . 45
7.2.1 Proportional control . . . 45
7.2.2 Integral control . . . 46
7.2.3 Proportional and integral control . . . 47
7.2.4 Implementing the PI controller . . . 47
7.3 Defining the error signal . . . 48
7.3.1 Feedback . . . 49

8 Results 51
8.1 Walking experiments . . . 51
8.1.1 The intuitive approach to gait generation . . . 51
8.1.2 The periodic function approach to gait generation . . . 55
8.1.3 Stationary walking on the ground plane . . . 56
8.2 Stationary balancing in response to an external force . . . 57
8.2.1 No control . . . 61
8.2.2 Proportional control . . . 63

8.2.3 Proportional integral control 63

9 Conclusion 67

10 Future work 69

10.1 Design 69

10.2 Gait generation 71

10.3 Motion control 71

List of Figures

1.1 A photograph of the highly successful Honda P-2 and P-3 robots 2
1.2 Diagram illustrating static and dynamic walking. 3
1.3 Biped walking robot link and point mass model 4

2.1 A photograph of the Shadow biped robot 11
2.2 A photograph of the WABIAN robot 11

3.1 Link structure of the designed robot, with axis indicating roll, pitch and yaw. . . 14
3.2 A photograph and specifications of the Futaba S9402 servo. 15
3.3 Figure showing the relationship between torque Γ and force $\mathbf{F}$. 16
3.4 The final design for the robot. 19
3.5 A photograph of the constructed robot. 20

4.1 Photograph of some of the hardware used in the biped walking robot. 22
4.2 Plots showing the acceleration data obtained from the ADXL05 accelerometer . 25

5.1 An illustration of the gait cycle. 30
5.2 Graphs of joint anngles for the intuitive approach for gait generation 32
5.3 Diagram showing the biped robot in single support phase 33
5.4 Plot of joint angles solved using inverse kinematics. 38

6.1 A screenshot of the Gait Generator interface. 40

7.1 The output from a proportional controller for a given error input signal 46
7.2 The output from an integral controller for a given error input signal 47
7.3 The output from a proportional integral (PI) controller for step error input signal. 48
7.4 The output from a proportional integral (PI) controller for a piecewise linear error input signal. 48

8.1 Results of the intuitive approach to walking, implemented without control. . . . 53
8.2 Results of the intuitive approach to walking, implemented with proportional control. 54
8.3 Results of the intuitive approach to walking, implemented without control. . . . 56

8.4 Results of stationary walking on the ground plane, implemented without control at an update rate of 13 Hz (78 ms period). 58
8.5 Results of stationary walking on the ground plane, implemented without control at an update rate of 11.7 Hz (85 ms period). 59
8.6 Results of stationary walking on the ground plane, implemented with proportional control. 60
8.7 Results of stationary balancing in response to an external force, without control. 62
8.8 Results of stationary balancing in response to an external force, with proportional control. 64
8.9 Results of stationary balancing in response to an external force, with proportional integral control. 65

10.1 Alternative design to the current biped robot. 70
10.2 Using an artificial neural network (ANN) for control of the dynamic walking gait. 72

CHAPTER 1

Introduction

In recent years there has been much interest stimulated in dynamic walking in bipedal robotics and legged locomotion in general. Part of the reason for this interest is the need for robots which can operate in human oriented environments. Humans present a very elegant model of locomotion to emulate.

Bipedal robots will operate in a human environment with much greater efficiency than any other type of robot yet devised. It is hoped that eventually bipedal robots can be used to complete tasks which are too difficult or dangerous for humans. This includes applications such as working in extreme environmental conditions (such as in fire rescue operations), with toxic gases or chemicals, with explosives (such as land mines) or as an aide to humans in similar situations. Also, a useful by-product of research into bipedal robotics will be the enhancement of prosthetic devices.

The state of research into bipedal robotics has progressed to the stage where dynamic walking gaits are being studied. Human beings usually employ a dynamic gait when walking as it is faster and more efficient than static walking [1].

Dynamic walking is characterised by a small period in the walking cycle where the center of gravity of the robot is not projected vertically onto the area of either foot [2]. This requires there to be a period of controlled instability in the gait cycle, which is difficult to accomplish unless the mechanical system has been designed bearing this in mind.

There have already been some bipedal robots which have realised this goal to varying degrees of success. Most notably are the Honda P-2 and P-3 robots, shown in Figure 1.1. The P-2 robot alone is reputed to have a development cost of $3 million dollars. This particular biped robot is human sized with a mass of over 200 kg. The robot can climb stairs and push objects, and incorporates a virtual-reality remote manipulator control which allows a remote operator to manipulate the gripper on the robot.

However, unless similar robots can be developed for only a fraction of the cost, they cannot be easily transformed for use in their intended application. Moreover, research institutions

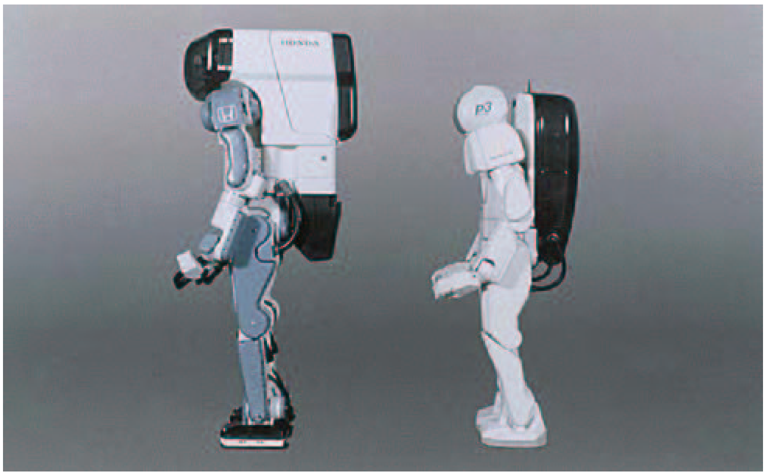


Figure 1.1: A photograph of the highly successful Honda P-2 (left) and P-3 (right) robots. (Source http://www.honda.co.jp/)

cannot participate in developing such technologies while they are so expensive.

This thesis presents a small scale low cost bipedal walking robot which can be used to investigate control and stability issues inherent in life sized robots. We examine in detail the issues surrounding the design of such a robot as well as the development of the control software.

In this chapter we examine the motives for research into dynamic walking bipedal robotics, and review some concepts and terminology which will be used in the remainder of this dissertation.

1.1 Why study legged locomotion?

Although legged machines seem to hold people in fascination and awe at their apparent complexity, legged robots have much more potential than their wheeled counterparts. Both types of robots are designed to travel in some prespecified environment, however it can be seen that legged robots are much more mobile. This is mainly because wheels are not suited to rough terrain, despite attempts to adapt such vehicles to these environments. While this statement may seem straightforward, it is helpful to consider exactly why.

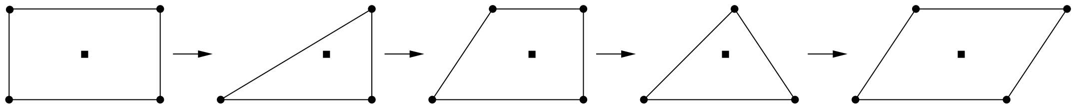
The first reason why legged robots are more mobile is that they can use isolated footholds separated by unusable terrain to optimise support and traction [3]. In other words, where wheels require a continuous, unbroken path of support, legged robots can traverse terrain which is discrete and discontinuous. It is this flexibility in changing the configuration of the support base that makes legged locomotion so adaptable and versatile.

Legs can also function as a built-in suspension system, allowing the body to travel along a different trajectory than the feet. In this way, the path of a payload may be smooth, despite the granularity and roughness of the environment. In fact, it is this decoupling and independence of the motion of the legs and the body which gives rise to superior speed and efficiency.

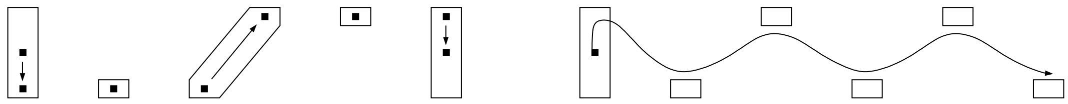
While wheeled robots are undoubtedly more efficient on smooth flat surfaces, many environments deviate substantially from this ideal. In fact, as humans we often create the environments which we inhabit in such a manner that they are largely unsuitable for wheeled robots. Therefore it is natural for legged robots to be more efficient and versatile in operating in human environments. However, despite the many excellent examples of legged locomotion we have all around us, we still are a long way from producing reliable and usable legged robots and machines. By building biped robots and trying to emulate these examples, more will be learnt about the difficulties of the problem and as a result the solution will become clearer. This does not deter the research effort in this area, which is producing increasingly useful models of biped locomotion.

1.2 Dynamic and static walking

Static balance or static walking refers to a system which stays balanced by always keeping the center of mass (COM) of the system vertically projected over the polygon of support formed by the feet [3]. While this is the case there can be no horizontal acceleration due to tipping moments caused by gravity. Therefore whenever a foot or leg is moved, the COM must not leave the area of support formed by the feet still in contact with the ground. This is illustrated by Figure 1.2.



(a) static quadruped walking (feet are denoted by •)



(b) static biped walking

(c) dynamic biped walking

Figure 1.2: Diagram illustrating static and dynamic walking, from an overhead viewpoint. In each subfigure the direction of motion is from left to right, and the position of the center of mass is indicated by ■. Polygons indicate the area of support formed by the feet. For subfigure (c) the trajectory of the center of mass is indicated.

In contrast, a dynamically or actively balanced system[1] is not constrained in such a manner.

---

[1] In this thesis, the terms active and dynamic balance refer to the same concept.

The COM may leave the support area formed by the feet for periods of time. This allows the system to experience tipping moments, which give rise to horizontal acceleration. However, such periods of time are kept brief and strictly controlled so that the system does not become unstable. Thus one may think of a dynamically balanced system as one where small amounts of controlled instability are introduced in such a way as to maintain the overall equilibrium. Tipping moments in one direction are negated by tipping moments in the opposite direction.

When we compare the two methods of balance, we see that the static method is highly restrictive and results in movement which is slow. Very rarely do animals and humans exhibit such behaviour for this reason—the velocity achievable is very low and the motion is not efficient. However, we can see that by removing the constraining nature of the rule for static balance that the mobility of the system is increased. This is due to the increased flexibility of the movement of the legs and placement of the feet. The accelerating tipping moments can be used to achieve higher speeds, move all legs at once or to utilise footholds which are far apart.

Therefore it can be seen that in order for a bipedal robot to gain efficiency and speed, it will require dynamic balance. Much of this dissertation will be concerned with analysing the forces on the system which result from the COM being outside the base of support of the robot. In many cases the base of support is only one foot, which gives rise to substantial accelerations possible in the horizontal plane.

1.3 Nomenclature

In this dissertation, various terms will be used to describe the frame of reference when describing biped robots. A three dimensional Cartesian axis is used, positioned with the origin at the base of the robot (as indicated in Figure 1.3), with the forward direction of motion proceeding along the x-axis.

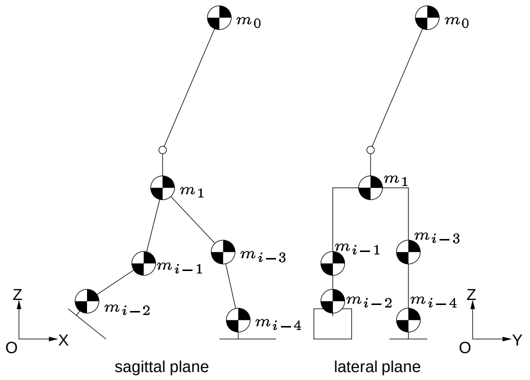


Figure 1.3: Biped walking robot link and point mass model. In determining the mass distribution while the robot is in motion, we model each link of the robot as a rigid body with a point mass attached.

The horizontal plane upon which the robot walks is referred to as the ground plane, defined by the xy-axis. Two other planes are defined—the sagittal plane, which is synonymous with the xz-plane, and the lateral or frontal plane, defined by the yz-plane. In addition to this terminology, each axis is defined as a rotational axis for referring to joints. The x-axis is known as the roll axis, the y-axis as the pitch axis and the z-axis the yaw axis.

A joint with motion restricted to rotation in the plane or translation along a line is referred to as having one degree of freedom (DOF). As an illustration, in Figure 1.3 the robot drawn has seven rotational DOF on the pitch axis (at the ankle, knees and hips), and one rotational DOF on the roll axis (at the base of the trunk).

1.4 Research objectives

The aim of this research is to design, build and implement a robot to achieve bipedal dynamic walking. While is a difficult task, especially given the severe time constraints, this has been achieved in some form. To clarify the goal, we hope to develop the robot and control system to the stage where it can walk unsupported on a flat surface indefinitely (given sufficient power supply conditions). It is hoped that the control system will be able to accept user variation of gait parameters such as step length, step period and foot lift height as a coarse basis for providing a method of controlling the speed and position of the robot.

Under ideal conditions, a model that includes gait adjustments for slight disturbances in the environment will be achieved. A real application will call for a robot to function under non-ideal conditions, and it will always be the case that the environment will vary in different locations. Therefore the robot will be required to adjust to these fluctuations in order to maintain sufficient conditions for balance.

The research can be divided into two distinct and logically separable areas – *design* and *implementation*. The design phase refers to the research involved in creating the physical robot upon which experiments will be performed. This focuses on the devices that are used in the robot as well as physical aspects such as balance, mass distribution and forces. In contrast, the implementation phase refers to the research involved in developing a control system and preparing the robot for experimentation. This involves developing gaits, determining key control parameters and developing tools to help implement the control system and analyse collected data. In this thesis we will address both of these research phases in detail.

1.5 Overview

The structure of this dissertation is divided into ten parts. Chapter 2 reviews past literature and the current state of research in bipedal robotics. In Chapter 3, the design of the robot that was used in experiments is described, and Chapter 4 details the hardware platform used, discussing the microcontroller, servos and sensors.

Gait generation is discussed in Chapter 5, and two different approaches are presented—an intuitive approach and a periodic function approach. In Chapter 6 the Gait Generator software is described, that was developed to aide the development of gaits for the robot.

Chapter 7 discusses the problem of controlling dynamic gaits, and presents two simple control methods that may help achieve this. The results of several sets of experiments that were conducted to test the gait generation approaches and control techniques are detailed and analysed in Chapter 8.

Conclusions regarding the experiments are drawn in Chapter 9, and possible future research that could be conducted is discussed in Chapter 10.

CHAPTER 2

Literature review

The first biped robot to be successfully created and use dynamic balance was developed by Kato in 1983 [4]. While this robot largely used static walking, it was termed quasi-dynamic due to a small period in the gait where the body was tipped forward to enable the robot to gain forward acceleration and thus achieve a forward velocity. This achievement has largely been cited as the defining moment where the focus of research shifted from static to dynamic walking.

Since this time, progress has been somewhat sluggish. The same research group produced the WL-10RD robot which walked once more with quasi-dynamic balance in 1985 [5]. The robot was required to return again to static balance after the dynamic transfer of support to the opposite foot. However Miura and Shimoyama [6] abandoned static balance entirely in 1984 when their stilt biped BIPER-3, which was modelled after a human walking on stilts, showed true active balance. Simple in concept, it contained only three actuators; one to change the angle separating the legs in the direction of motion, and the remaining two which lifted the legs out to the side in the lateral plane. Since the legs could not change length, the side actuators were used to swing the leg through without scuffing the foot on the walking surface. An inverted pendulum was used to plan for foot placement by accounting for the accelerating tipping moments which would be produced. This three degree-of-freedom robot was later extended to the seven degree-of-freedom BIPER-4 robot.

Another approach had been taken by Raibert [3], who developed a planar hopping robot. This robot used a pneumatically driven leg for the hopping motion and was attached to a tether which restricted the motion to three degrees of freedom (pitch motion, and vertical and horizontal translation) along a radial path inscribed by the tether. A state machine was used to track the current progress of the hopping cycle, triggered by the sensor feedback. The state machine was then used to modify the control algorithm used to ensure the stability of the machine. A relatively simple control system was used which modified three parameters of the hopping gait, namely forward speed, foot placement and body attitude. The success of this research motivated Raibert to extend the robot and control system to hopping in three dimensions, pioneering the area of ballistic flight in legged locomotion.

Continuing through the years, a dynamic running robot was developed by Hodgins, Koechling and Raibert [3, 7], extending the previous studies of one-legged hopping machines in two and three dimensions. This robot was constrained to two dimensions (motion in the sagittal plane), and used a similar control method as for the hopping robot in two-dimensions. This control system decoupled the three important control parameters of body height, foot placement and body attitude, controlling these three aspects of the running gait through the use of a state machine. The state machine switched states when certain key feedback events occurred, and the robot was controlled differently depending upon the current state of the system.

Much early research around this time focused on intensely analytical techniques for designing and controlling robot motion. This had the tendency to produce complex equations governing the motion of the robot, which often had no solution and had to be approximated or linearised. Sometimes this approach was successful despite such shortcomings. Kajita *et al.* [8] used this approach to control bipedal dynamic walking by restricting the movement of the center of mass (COM) in an ideal sense to the horizontal plane only. This motion was termed a “potential energy conserving orbit” and could be expressed by a simple linear differential equation, which simplified the calculations involved.

Other similar analytical approaches actually increased the complexity of the problem by introducing new links to the biped model. Takanishi *et al.* [9, 10] used the robot WL-12RIII with a control system which manipulated the zero moment point (ZMP) to achieve dynamic stability, even on uneven surfaces. This robot had seven links including a trunk or upper torso link with two degrees of freedom, thus allowing it to pitch and roll relative to the forward direction of the robot. As an extension to this work, Yamaguchi *et al.* [11] used the robot WL-12RV in a similar fashion, adding the feature of a yaw-axis movement to the trunk motion. This allowed compensation for yaw moments occurring about the foot in contact with the ground, eliminating the unwanted behaviour of the robot to turn at higher velocities. This addition allowed the robot to travel up to 50 percent faster than previous efforts had achieved.

A different analytical approach examined how mechanical design contributes to robot performance. McGeer [12] showed that a correctly designed biped walker with no actuation and no control could walk down gentle slopes. His research showed that passive-dynamic walking is possible[1]. The slope allowed the robot to regain through gravity energy lost though friction and impulse collisions. Garcia *et al.* [13] also showed this using the simplest purely mechanical model possible—a double pendulum. This work highlighted the fact that mechanical design is equally, if not more important, than the control method used. This suggests that more effort spent on ensuring a correct mechanical system design will simplify the complexity of the control system required.

---

[1] The term passive here refers to the fact that actuation is not present in this walker.

A third approach to bipedal dynamic walking has only recently emerged in the last few years. Analysis using the dynamic equations of motion can be complex, non-linear and may have no closed form solution. Artificial neural networks are well suited to this type of control problem having the advantage that they can learn and adapt the behaviour of the system to a desired state, even if this state is not clearly defined. The benefit of this approach is that complex dynamics and kinematic equations need not be known, or a greatly simplified version may be used instead. The result is that neural networks may be used in real-time to adapt the walking gait on-line, a problem which previous control methods have not been able to significantly address.

Doerschuk *et al.* [14, 15] applied an intelligent learning approach to control the legs of a simulated biped robot while in ballistic flight. They used a Cerebellar Model Articulation Controller (CMAC) neural network to impose a previously generated gait onto a simulated seven link biped robot. A more impressive use of neural networks can be seen in the research of Miller and Kun [16, 17], who used three CMAC neural networks in an attempt to produce a control system which could operate in a wider range of environments by adapting various parameters of the walking gait such as step length, step height and step period. One network was used to learn the required motions to achieve side balance in the sagittal plane, one performed forward/backward balance in the lateral plane, and the last network learned the closed chain kinematics in order to keep the feet parallel to the ground via actuated ankles. While not entirely successful, the robot did learn the required behaviour in order to start walking from a stationary position and later come to rest through variation of the parameters within a limited range.

More recent research is being performed in the United Kingdom by the Shadow Robot Group who have developed the Shadow Walker prototype (see Figure 2.1). Following an anthropomorphic design and using a wooden frame, they have constructed a biped robot using special ‘air’ muscles developed by the group. The pneumatic air muscle behaves in a similar manner to a biological muscle, contracting up to 40% of its length when actuated with a supply of air. The compliant muscle has a power to weight ratio of approximately 400:1, vastly outperforming conventional actuators. Twenty-eight air muscles (fourteen per leg) actuate the eight joints in the robot. With twelve degrees of freedom, the muscle arrangement is designed to closely mimic the human leg muscles by placing the air-muscles in corresponding human muscle positions.

Another current research project is the WABIAN Humanoid project at the University of Waseda in Japan. The aim of this research group is to develop anthropomorphic robot mechanisms using bio-mechatronic techniques. This includes research on human motion dynamics, human-like mechanism design, and mind analysis and synthesis. Based upon the previous work by Kato, Takanishi and Yamaguchi, with the WL series of robots, the project was established in 1992 to combine the fields of vision, information processing, brain modelling, mechanical design, active sensor integration (tactile, visual and sound), robot psychology, speech recognition and conversation. In an attempt to enable robots and humans to build common mental and physical spaces, the project comprises of over 50 full-time researchers.

While there has been some excellent research performed to date in bipedal dynamic walking, and significant progress has been made, it should be noted that all of the above control methods were either simulated or where they controlled a real robot, they were executed from an external control source. In most cases the power source for the robot was also external. This further limits the practicality of the developed robots—to be truly mobile they will need to overcome such limitations. We have chosen to address these issues by designing a self-contained robot with both the control and power sources on board.

Figure 2.1: A photograph of the Shadow biped robot. Note the use of pneumatic ‘air’ muscles. (Source http://www.shadow.org.uk/)

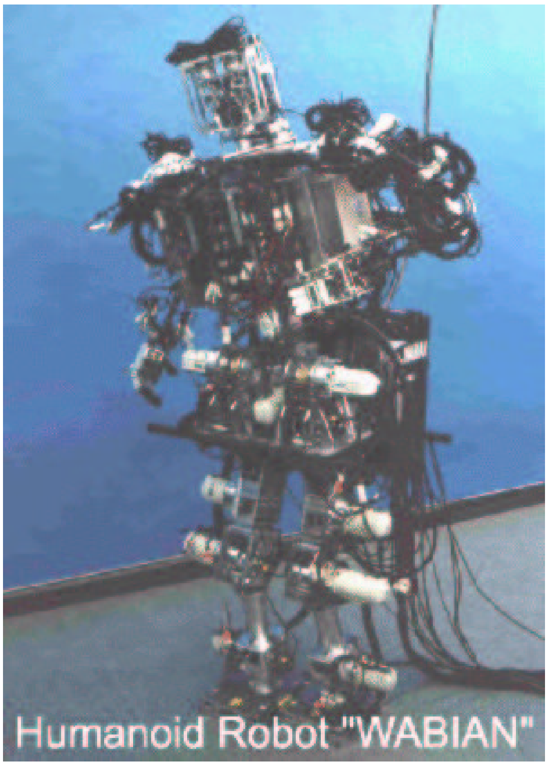


Figure 2.2: A photograph of the WABIAN robot, part of the University of Waseda Humanoid project. (Source http://www.shirai.info.waseda.ac.jp/humanoid/)

CHAPTER 3

Design

The first task in this research project was to design the robot which would be used to investigate theories of bipedal walking. However, since the funding for the project was limited, a very cost effective design needed to be developed in order to succeed. This design was also required to withstand the rigours of mechanical stress imposed upon it during experimentation. In this Chapter the design process will be described, important aspects of the mechanical design of the biped robot will be discussed and the final constructed robot will be presented.

3.1 General design

The link structure of the robot is shown in Figure 3.1. The motivation for this design can be found in the research by Takanishi *et al.* [9, 10] and Yamaguchi *et al.* [11]. The basic configuration of the robot is made up of two legs as the lower limbs, each having 4 degrees of freedom (DOF). Three of these are rotational on the pitch axis at the hip, knee and ankle. The fourth is also rotational and located at the hip on the yaw axis. This effectively allows us to model the hip as a two DOF rotational joint. The trunk also has 2 DOF in the roll and pitch axis at the pelvis. However as can be seen, one these is shared with the rotational DOF on the pitch axis at the hip.

The 2 DOF in the trunk are used for stabilising the robot during the walking gait. This occurs as the trunk is moved in such a manner during walking in order to compensate for the movement of the lower limbs. As the trunk is moved relative to the lower limbs, the position of the center of mass (COM) will change, thus allowing the stability of the system to be corrected. Therefore, the design parameters require careful consideration and planning in order to achieve dynamic stability.

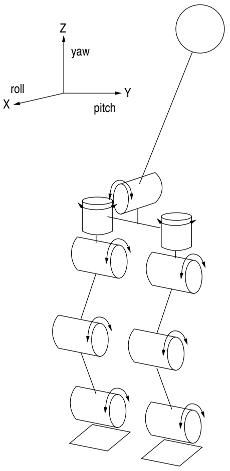


Figure 3.1: Link structure of the designed robot, with axis indicating roll, pitch and yaw.

3.2 Design process

The design process involves the creation of a specification for the building of a robot upon which the chosen model of dynamic walking will be implemented. The aim is to derive the specifications such that the chosen walking model will succeed. This is not a trivial task—there are many considerations to take account of in order to ensure that the biped robot will be stable while walking. The most important of these are balance, forces, moments, torque, proportions, mass and strength.

3.2.1 Balance

Balance is the most important consideration. The aim of the design is to ensure that the walking model will achieve dynamic balance. Control methods are used to gain dynamic balance, however the mechanics must be chosen so that the robot can respond rapidly enough to the required movements. One means of achieving this goal is to distribute the mass throughout the robot such that any movements required to balance the robot are small. These movements can therefore be made rapidly without generating large moments which would further destabilise the robot. To achieve this, the COM should be placed in a location low enough to stabilise the robot inertially, but high enough so that it can be moved only small amounts to correct for undesired behaviour.

The correct placing for the COM is in the lower trunk, similar to humans. This provides for stability and allows the trunk to be moved, shifting the COM to achieve desired accelerations to counteract existing undesired accelerations [9, 10].

A second means of achieving the goal of rapid response to required movements is to use fast actuators. Since only electric motor devices were within the budget allowed for the project, servos were chosen for this task. Futaba S9402 servos (see Figure 3.2) were used, which are commonly employed for use in radio controlled toy vehicles such as cars, helicopters and boats. These servos were available for A$150.00 and were high torque, as required for lifting large masses such as the frame, electronics and other parts which make up the robot.

Specifications:

Dimensions:	40.5 × 20 × 37.5 mm
Mass:	0.055 kg
Operating speed:	0.10 sec/60°(600°/sec)
Output torque:	8.0 kg·cm (0.784 Nm)
Other features:	coreless, high-torque, aluminium geared (hard Alumite treated) and dust/drop proof

Figure 3.2: A photograph and specifications of the Futaba S9402 servo.

3.2.2 Forces, moments and torque

Forces are an integral part of any mechanical system and must be taken into account. In the case of the bipedal robot, there not only exist forces due to the acceleration field of the earths gravity, but forces produced by the robot itself. In both cases these must be carefully controlled to ensure the robot maintains balance, and are controlled by moving the robot and altering its attitude to change the acceleration experienced.

The mechanics must be able to provide enough force under all required circumstances to ensure the correct response to the control system. To this end, the maximum force required to be exerted by the servos was calculated by writing a `C` program which used kinematics and estimated mass distributions to calculate the maximum torque required by the servos. The calculations revealed that the maximum torque would be experienced at the ankle and knee joints, and would be approximately 0.718 Nm.

The definition of torque is given as

$$\Gamma = Fl = Fr\sin(\theta) \tag{3.1}$$

where Γ is the torque or moment[1] F is the magnitude of the applied force, r is the radius of the applied force from the axis of rotation and θ is the angle of the of the applied force to the reference axis (the axis that **F** acts along—see Figure 3.3). The quantity l is the moment arm, and represents the perpendicular distance from the rotation axis to the vector **F**, or the reference

---

[1] The terms torque and moment are synonymous, and both will be used in this dissertation.

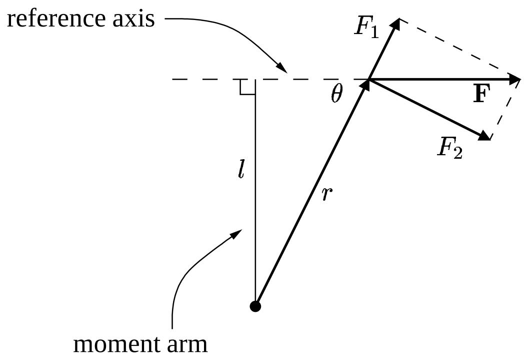


Figure 3.3: Figure showing the relationship between torque Γ and force $\mathbf{F}$.

axis. This relationship can be used to resolve all required servo torques to keep the robot links in static equilibrium as a rigid body. These resolved torques represent a reasonable estimate of the torque required from the servos, and allows a choice of servo to be made.

In order to simplify the construction of the robot as much as possible, no gearing was used at the actuated joints. That is, the output from the servos was directly coupled to the joints. This meant that all of the available torque was transferred to the movement of the links with as close to full efficiency as possible, due to the absence of external gears[2].

Since the precise construction of the robot was not known in advance, this estimation was used to select the Futaba S9402 servos[3] that were used as the joint actuators. An estimate of the mass distribution was obtained from measuring material samples and weighing of component parts.

3.2.3 Proportions and mass distribution

McGeer [12] demonstrated that proportions can be more important than control by showing that dynamic walking can be achieved without actuation or control. In particular, mechanical parameters such as length and mass distributions may have a greater effect on the human gait than previously imagined. Therefore, considering the proportions of the robot such as leg link length and mass distribution throughout the body are imperative to success.

Some components of the robot were already pre-dimensioned, such as the servos, camera, electronics and batteries. Such inflexible parts had to be incorporated, constraining the minimum dimensions for the robot. However, in order to keep the mass of the robot to a minimum, it was important to use as little material in the robot frame structure as possible. One further objective is to design the robot with proportions which will allow the robot to balance relatively

---

[2] While no gears were used at the joints, it should be noted that gears exist internally in the servos, resulting in gearing efficiency losses. However this is accounted for in the servo specifications.

[3] Due to a shortage of supply of this servo in Australia, a slightly less powerful servo was also used in the upper hip joints, which require less torque.

easily while still allowing the research into dynamic walking to be accomplished. For this reason the mass and mass distribution play a large role in determining the designed dimensions of the robot.

The mass of the robot and the distribution of mass throughout the body of the robot is related to the forces operating on and within the robot. If the mass of the robot is too large, it will not be able to respond to the control system rapidly enough or even function correctly at all, especially if the servos chosen are not powerful enough. More importantly, the mass distribution within the robot also effects the balance of the robot, since this determines the location of the COM. In particular, the absolute mass distribution will vary as the robot links move relative to each other, meaning that the position of the COM within the robot will change during the walking gait. The movement of the COM will have a significant influence on the stability of the robot, therefore considering the mass distribution is also important to achieving dynamic walking.

In order to stabilise the robot as much as possible, we place the COM as low as possible while still allowing the trunk to be useful for compensating for the movements of the lower limbs. One means of achieving this is to use leg links which are as short as possible. Since we directly couple the servos to the link joints, we are also constrained in how small we can make this dimension. Furthermore, we would like to minimise and restrict the movement of the COM in a predictable manner, so we can directly control this as a method to balance the robot. To do this we make the ratio of the combined mass of the leg links to the combined mass of the upper body as small as possible. In this manner, movements of the leg will affect the position of the COM marginally, and even then only in the sagittal plane. We can control the side balance of the robot by swaying the trunk in the lateral plane. Since these planes are perpendicular, we need not be concerned about movements of the leg affecting the side balance of the robot through shifting the COM relative to the robot.

3.2.4 Strength

The robot must be designed with strength in mind since it is anticipated that the robot may be subjected to considerable stress during operation. Therefore all parts of the design must be robust enough to withstand these forces. For this reason the frame of the robot was constructed using 3 mm aluminium channel for the links and 5 mm machined aluminium flat plate for the joint brackets, making the structure of the robot very strong[4]. The Futaba servos used were coreless for long wear, shock resistant, and had metal gears to allow safe external manipulation of the output shaft. The acceleration sensors chosen were semi-solid-state and as a result can withstand high impulse shocks[5].

---

[4] All aluminium used in the construction of the robot frame was 60-series grade aluminium.

[5] The ADXL05 acceleration sensor is rated to withstand a maximum shock acceleration of 1000 g while unpowered and 500 g while powered.

3.3 Final design

The final design is shown in Figure 3.4[6]. The light-blue sphere is a QuickCAM digital camera, that may be used for image processing. The three green rectangles are the circuit boards which make up the EyeBOT hardware platform which will be used to control the robot. The 9 grey rectangular prisms represent the servos which are responsible for mechanical movement. Finally the yellow rounded rectangular prisms are the battery power supply, which consists of 6 AA size 1 Ah high capacity NiCAD batteries, three in each foot. These devices are discussed in more detail in Chapter 4.

3.4 Construction

The final, constructed biped robot[7] can be seen in Figure 3.5. Overall the robot characteristics were as planned except for the placement of the COM. However, the balance was found to be exceptional, with the robot being able to statically balance quite stably without actuation in an upright position. This meant that the COM, although not in the expected vertical position, was found to be in the correct position with respect to the horizontal plane.

3.4.1 Required design modifications—center of mass

The distribution of mass of the constructed robot differed slightly from the estimated mass distribution due to two factors. The first reason was that the amount of aluminium required to make up the structure was underestimated. This increased the final mass of the robot above the estimated mass by approximately 0.37 kg, making the final mass of the robot 1.78 kg. The main reason for the underestimation was that heavier brackets were used around the servos to increase the strength at link joints, and to prevent unwanted forces occurring against the servo axis of rotation.

The second and more important reason for the actual mass distribution differing from the estimated distribution is that most of the heavier aluminium was added to the robot legs, while less aluminium was used than estimated in the upper torso. The result of this difference shifted the COM of the robot significantly lower than planned and expected. In fact, the COM resided significantly below the torso joint, which greatly reduced the effect that swinging the torso from side to side would have on shifting the COM in the lateral plane. Therefore, in order to increase stability in the lateral plane, raising the COM further above the torso joint was required. This was attempted by shifting the batteries from the feet to the upper torso. This had the desired

---

[6] This technical drawing was prepared with AutoCAD Release 14

[7] The author would like to express thanks to the Departmental General Workshop and in particular to Richard Meager for constructing the robot and for structural design of the robot.

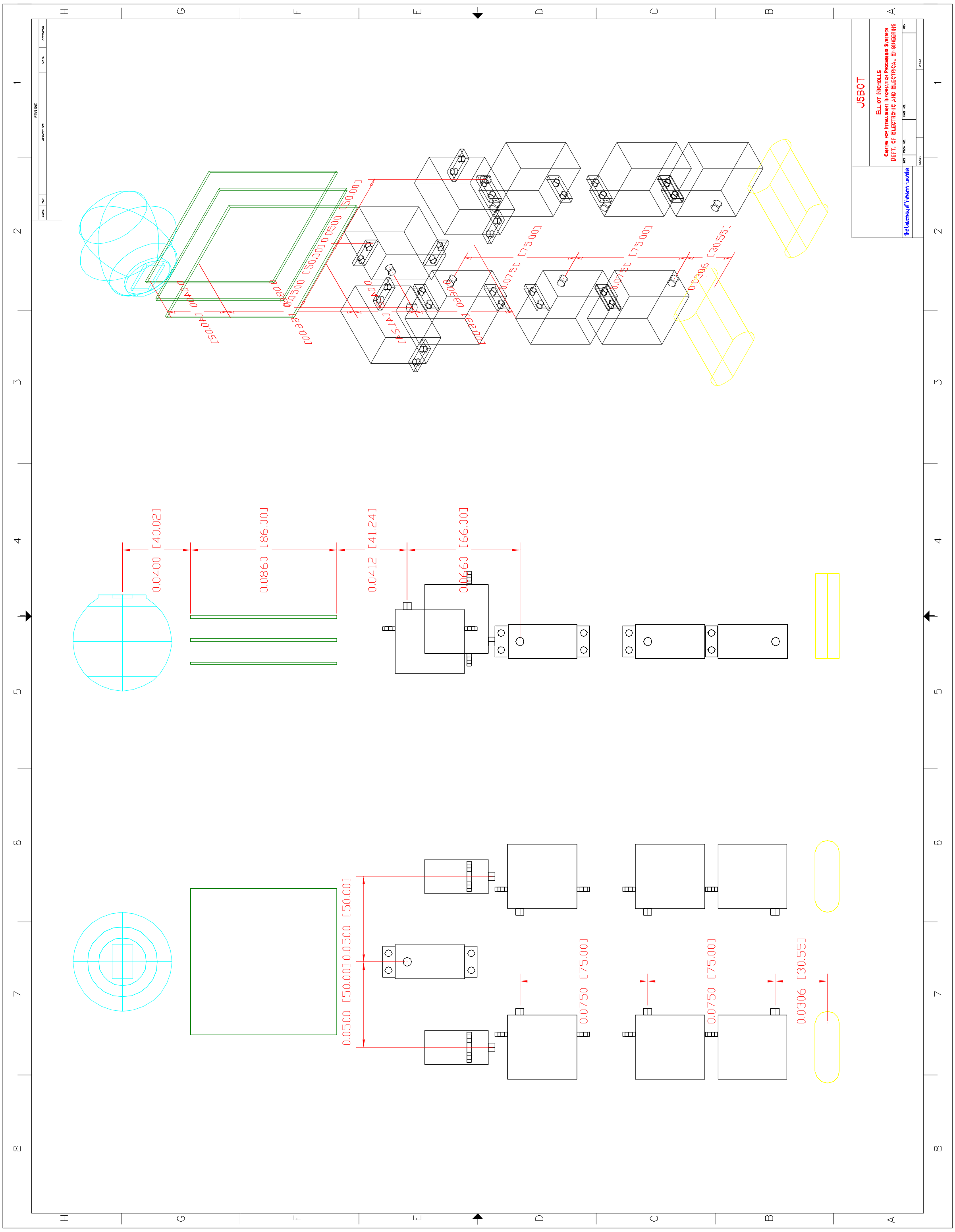


Figure 3.4: The final design for the robot. The light-blue sphere is a QuickCAM digital camera, the three green rectangles are the circuit boards of the EyeBOT hardware platform, the 9 grey rectangular prisms represent the servos which are responsible for mechanical movement and the yellow rounded rectangular prisms are the NiCAD battery power supply.

Figure 3.5: A photograph of the constructed robot. Overall the robot characteristics were as planned except for the COM. However the balance was found to be exceptional, with the robot being able to statically balance quite stably without actuation in an upright position.

effect of allowing the robot to almost achieve static balance when one foot was lifted off the ground plane and the trunk was tilted to the limit of its extent of movement in the lateral plane. Later, a further pair of batteries were fitted near the top of the trunk to allow the robot to statically balance on one foot—however, the COM still resided below the torso joint.

CHAPTER 4

Microcontroller architecture and sensor hardware

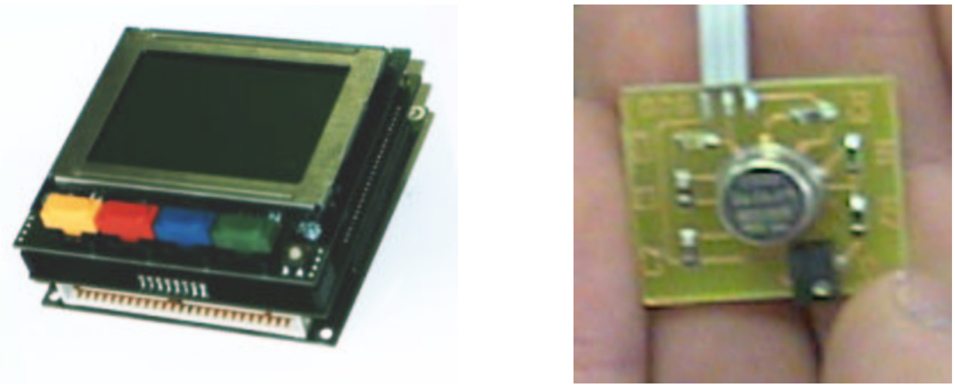
As mentioned in the introduction, the biped robot was designed with the aim that the controller and power supply would be on board, making the robot self sufficient and unreliant on outside support. In this chapter we will review the electronic inventory of the robot and examine each part in turn. First we will look at the microcontroller architecture, and the software development process. Next we will reveal the details of the sensors used and how they interface with the microcontroller. Finally we will discuss the testing and calibration which was carried out on the servos and sensors. The hardware discussed in this section can be seen in Figure 4.1.

4.1 Controller architecture

The microcontroller hardware platform is built around the Motorola 68332 32-bit microcontroller, known as the Eyebot hardware platform. The system runs at 25 MHz and has 1 Mb of RAM and 512 Kb of flash PROM, which is used to store the operating system and can also be used to store user programs. The important aspects of this platform are the system interfaces, including the timer processing unit (TPU) as well as the software development process.

4.1.1 Hardware description

The Eyebot hardware platform contains a serial port for communications with a PC, and a background debugging port which is also used to download new versions of the operating system. The platform also has multiple digital output ports and one digital input port. Additionally, 16 analogue to digital conversion ports are provided. A parallel input port allows connection of a Connectix QuickCAM CCD camera (either black and white or colour), allowing complex image processing to be performed without the need for a frame grabber. Also on-board is an LCD



(a) Eyebot hardware

(b) Acceleration sensor

(c) NiCAD rechargeable batteries

(d) Foot switches (indicated by arrows)

Figure 4.1: Photograph of some of the hardware used in the biped walking robot.

monochrome graphics display with a resolution of 64x128 pixels, as well as 4 keys. However the most important feature is the 16 digital input/output timer processing unit ports, which are ideal for directly controlling the servos. This part of the microcontroller will be explained in more detail below.

Another important feature of the Eyebot hardware platform is the hardware description table (HDT) which stores an inventory of interfaces available. By creating such a table, the user can inform the operating system which devices are currently available on the system. This helps to ensure user programs will run correctly by protecting the system. A feature of the HDT is that it is small, and need only be downloaded to inform the operating system of changes to the inventory. Previously this would have involved recompiling the entire operating system.

TPU unit and servo control

The timer processing unit (TPU) is an important feature of the Motorola 68332 microcontroller. Running at one quarter of the CPU clock rate, it is separate from the CPU core (although part of the same chip) and therefore does not create an extra load on programs which are running on the microcontroller. For this reason it is ideal for interfacing with and controlling the servos.

The servos are controlled by sending a pulse width modulated (PWM) digital signal of a constant period. The duty cycle instructs the servo to rotate to the absolute angle it is required to position at. For most servos, a 10% duty cycle will send the servo to the midpoint of the servo range. The servo has internal feedback in the form of a proportional derivative (PD) controller, which ensures that it positions at the correct angle, assuming there are no impediments to its rotation.

An assembly instruction can be called from a user program to set the duty cycle on particular TPU ports to control attached servos in this manner[1]. The function takes a byte value and converts this to a duty cycle on the TPU port by comparing it with parameters associated with the servo in the hardware description table. In this manner the servo can be accurately controlled without loading the microcontroller, leaving it free to perform higher level control programs.

4.1.2 Software development process

As mentioned the Eyebot hardware has an existing operating system known as RoBIOS, which is capable of multitasking. This provides the support for running programs, providing input and output routines and a large library of other functions. In order to develop programs to run on the Eyebot platform, there is a sequence of steps which must be observed.

All programs are written in ANSI C or assembly language, and must be compiled and linked against the RoBIOS libraries. Programs may also be linked against other standard library routines which are supported by RoBIOS such as `stdlib` and `math`. Once the program is ready for testing, it must be cross-compiled, linked, assembled and stripped. This is performed on a Linux platform, which has the necessary binaries installed to perform these functions.

Once this step is complete, the resulting HEX file is downloaded serially from the PC to the Eyebot hardware. The RoBIOS handles the incoming data file, and stores it in RAM. If the user desires the program may also be saved permanently in ROM for later access. Finally the program can then be run. It should be noted that programs can be written to upload data to the PC. Frequent use was made of this facility to upload sensor data collected during experiments for later analysis.

---

[1] The period of the PWM signal for the servos is 20ms, and the usual duty cycle variation is about ±5% to modulate the servo through its entire range. The bias value is about 10% duty cycle.

4.2 Sensors

Two different types of sensors are used in the robot. The first are acceleration sensors which are used for feedback regarding the balance of the robot. The second are switches that are used in the foot to detect whether the foot is in current contact with the ground or not.

4.2.1 Acceleration sensors

The acceleration sensors currently used are the Analog Devices ADXL05 single chip accelerometer. This is a relatively new chip which has only become available since mid-1995. This is a more sensitive version of the ADXL50 chip that is used in automobile Supplemental Restraint Systems (SRS) to activate safety airbags. The ADXL05 chip has a maximum resolution of 0.005 g and has a range of measurement from 0 g to ±50 g full scale[2]. The chip operates off a single 5 V power supply, which makes it ideal for interfacing to the Eyebot hardware.

The chip contains all the circuitry required to measure acceleration accurately, including a micro-machined sensor, oscillator, modulator, demodulator, voltage reference, signal conditioning, amplification and self-test circuitry. The range of operation is user selectable via external resistances, and the chip can measure both AC acceleration such as motion or vibrations and DC accelerations such as those due to the earth's gravity field.

Since the maximum acceleration expected to be experienced is that due to the earths gravity field, the range of operation was limited to ±1 g. This allowed for the maximum resolution within the required range. The output from the sensor is simply an analogue voltage signal, which is linear[3] and was scaled by the chosen resistors to approximately 2500 mV/g, with a zero g bias level of 2.5 V.

The output from the ADXL05 sensor was fed straight into the analogue to digital (A/D) converter. Initially two sensors were used—one to measure accelerations occurring along the y-axis (side acceleration) and one to measure acceleration along the x-axis (front/back acceleration). What is important to note is that the acceleration sensors were mounted on the pelvis of the robot, just above the hips. This is the most stable position for measuring acceleration as it captures both forward tilt and side tilt but will not be directly influenced by the movements of the leg links or the torso.

Once the signal is passed into the A/D converter, an assembly routine is used to read the A/D converter chip from the microcontroller. The routine returns a 10bit value, which represents the relative magnitude of the acceleration experienced by the sensor. This can easily be converted to a standardised value representing in g's using the following formula:

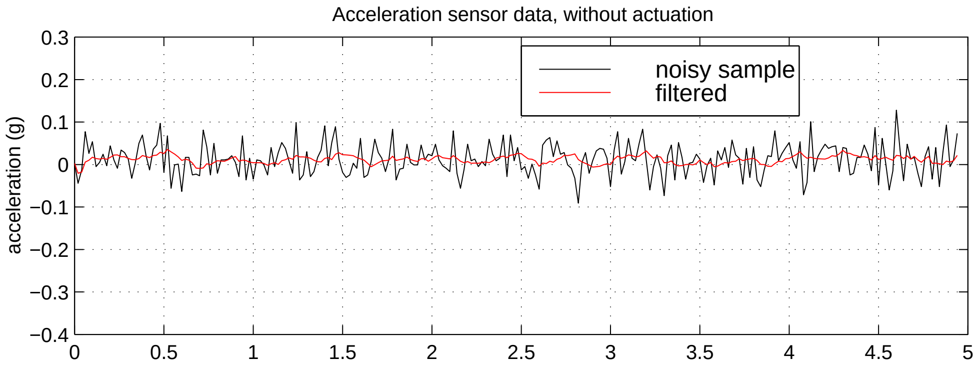
---

[2] 1 g is the acceleration exerted on an object by the earths gravity field. This is approximately 9.8 ms^{-2}.

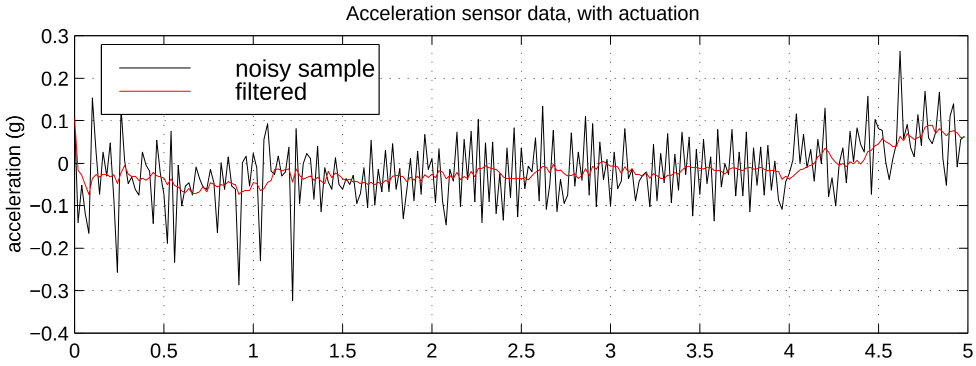
[3] The manufacturer's specification for the sensor states that the nonlinearity of the signal is 0.5%. The signal model can be approximated as linear for our purposes.

$$\text{Value in g's} = 10 \text{ bit value} \times \left(\frac{2}{1023}\right) - \left(\frac{2}{1023}\right) \tag{4.1}$$

An example of the acceleration data obtained while the robot is walking is shown in Figure 4.2.



(a) Data obtained without servo actuation



(b) Data obtained with servo actuation

Figure 4.2: Plots showing acceleration data obtained from the ADXL05 accelerometer. Plot (a) shows data obtained in the absence of servo actuation while (b) shows data obtained with servo actuation while the robot is stationary. Note that the noise increases significantly both in frequency and magnitude when the servos are actuated.

Acceleration sensor noise

Unfortunately the acceleration sensor output was found to be very noisy, as can be seen in Figure 4.2. There are two main sources for this noise. The first of these is the background noise, which is produced largely by vibrations in the robot structure, which are caused by the operation of the servos. There is nothing that can be done to reduce this noise, as the acceleration sensors are already vibration damped through attachment to the robot chassis by a damping material.

The second source of noise is from the sensor itself. The ADXL05 specifications state that the noise can be reduced and the signal to noise ratio can be improved by reducing the bandwidth of the measured signal. Two common methods of reducing the bandwidth are to use a low pass or bandpass filter. This can be done in by adding extra hardware circuitry, however it is fairly inexpensive simply to do this in software. This is easily done by taking a sliding average of some arbitrary number of previous measurements. This has the same effect as a low pass filter.

It was found that the average of the previous 10–20 measurements gave a satisfactory filtering effect. The effect of averaging different numbers of measurements can be seen in Figure 4.2. The filtered value was used to detect the acceleration being experienced by the robot.

4.2.2 Foot sensors

The foot sensors used were surface mount push-button switches, which were normally open. With an area of 2 mm by 6 mm and a depth of 1.6 mm they were easily recessed into the underside of the feet. Small printed circuit boards (PCBs) were fabricated and an area on the underside of the feet at the front and the back was milled out of the aluminium and the circuit boards were attached by small screws. Each PCB had two switches attached, so that each foot had a total of four foot switches attached, one near each corner of the foot (see Figure 4.1(d)).

A simple circuit consisting of a pull-up resistor was then used to attach each switch to one of the bits of the 8 bit digital latch input port of the Eyebot hardware. A simple assembly function allows the input latch buffer to be read, returning a single byte result, with each bit representing the binary status of one switch. A true result means that the switch is in contact with the ground.

Having sensors at the four corners of each foot allows some useful signals to be generated by combining the values of the individual switches. Combining the values of pairs of switches on either side of the foot is an indication of the status of the balance in the lateral plane. Similarly, combining the values from front or back pairs of switches will reveal if the foot is parallel to the ground in the sagittal plane. Finally, the combining the values of all switches on each foot can determine whether the foot is in contact with the ground.

4.3 Testing and calibration

When the hardware is newly installed or altered, it must be tested and calibrated in order to work accurately, predictably and reliably. To achieve this, both the servos and the acceleration sensors must be calibrated.

4.3.1 Servos

As mentioned before, the duty cycle sent to the servo dictates the absolute angle it should turn to. This process must be calibrated. Even between servos of the same brand the midpoint duty cycle bias and full range offset vary. This calibration process is performed by measuring the duty cycle required to center the servo in its range and the offset required to turn the servo to its range extents. These are then entered into the hardware description table in milliseconds, along with the period of the signal, which should always be 20 ms regardless of the servo brand. Once this is calibrated it can be used to rotate the servo in a linear fashion by altering the duty cycle.

A RoBIOS function which does this takes a single byte value, and converts this value using the hardware description table so that a value of 128 is the central position and values of 0 and 255 respectively will rotate the servo to the limit of its extent.

4.3.2 Acceleration sensors

The acceleration sensors also need calibration. The first step is to set the 0 g bias level to 2.5 V, which is at the center of the A/D converter range. This is done by placing the sensor on a level surface such that it experiences no acceleration, since it is perpendicular to the earths gravity field. A resistor is varied while the reading from the sensor is observed, until it converges to the desired value. The full range of the acceleration sensor can then be tested by holding it parallel to the earths gravity field, so that it experiences ±1 g. These values should match the expected range limits of the sensor.

CHAPTER 5

Gait development

Before a bipedal robot can walk, a gait or walking pattern must be developed for the robot to follow. There are many different ways of doing this, however the aim of gait development is to produce a gait which is dynamically stable. If the gait is dynamically stable, when the robot walks according to the gait in the absence of external disturbances, it will achieve dynamic walking. However, if the gait is not dynamically stable then the robot will fall over, since the system will be unstable.

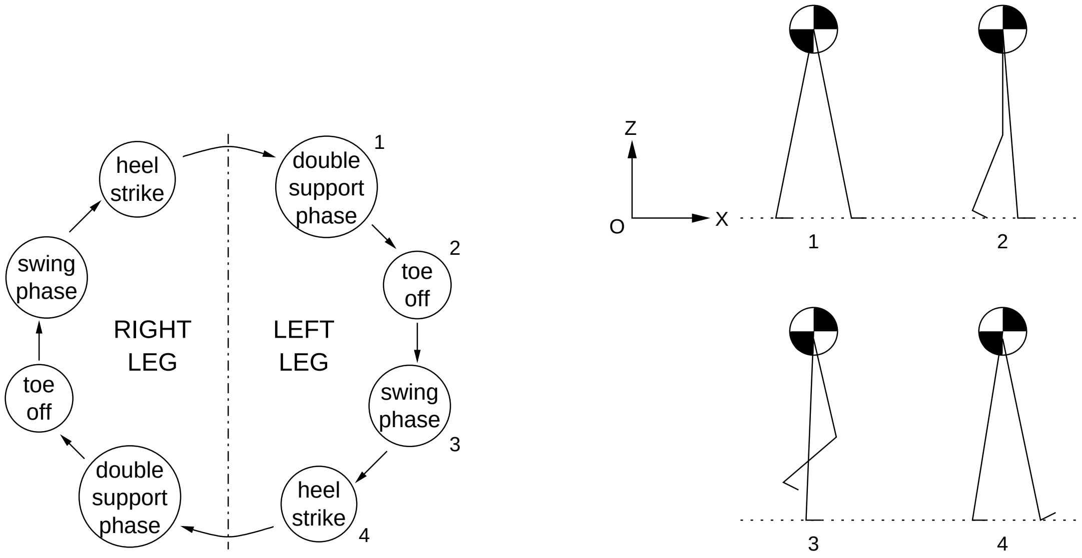
Walking is a repetetive motion, which consists of two main phases which alternate on each leg:

1. **double support phase** This phase exists when both feet are in solid contact with the ground plane. In this phase the robot is stable with a relatively large support base. The system enters this state when the front foot contacts the ground, and leaves this state when the rear foot breaks contact with the ground.

2. **single support phase or swing phase** This phase exists where only one foot is in solid contact with the ground plane. During this phase the center of mass (COM) of the robot rotates about this contact point in the manner of an inverse pendulum, while the other leg known as the swing leg translates in preparation to come in contact with the ground for the next double support phase. The system enters this state when the swing leg foot breaks contact with the ground and leaves this state when the swing leg foot contacts the ground.

This cycle can be seen in Figure 5.1. As can be seen from the state diagram, the walking pattern alternates the single support phases between each leg, interspersed with a double support phase between each alternation.

Some different approaches to gait development are discussed below.



(a) flowchart showing the gait cycle

(b) dynamic biped walking

Figure 5.1: An illustration of the gait cycle—(a) shows some of the phases from the gait cycle, and these are correspondingly drawn in (b).

5.1 Intuitive approach

The intuitive approach to gait development is not directly based upon dynamic principles, rather it is more a method of building one step of the gait at a time. This is achieved by experimental modification of the gait by examining the performance of the gait on the robot. This approach, has no previously known data to compare the performance of the robot against, and therefore we cannot generate a perfomance measure for the gait. Without such a quantitative measure we can only observe the robot while it is enacting the gait to determine the stability of the gait. For this reason the approach is called intuitive.

This method was chosen first to develop a gait due to its simple nature. As mentioned in the introduction to this dissertation, analytical approaches are complex and often have no solution or no unique solution. An intuitive approach allows the consideration of more subjective measures such as developing the gait through examination of the human gait, which is highly efficient.

To develop a gait using the intuitive approach, we used the Gait Generator software (described in Chapter 6). Frame by frame we assemble the gait using this software tool, which allowed us to animate the assembled gait offline. The source code which the Gait Generator produced was compiled, and the resulting program loaded into the robot for the gait to be tested. While the robot was enacting the gait, an MPEG (Motion Picture Experts Group) camera was used to record the experiment. Such MPEG files can be viewed later in slow motion to analyse the gait. Post analysis sessions of video footage provided invaluable feedback which allowed the gait to be modified to become more dynamically stable.

In this way, through many trials and examinations of the behaviour of the robot, an intuitive understanding is built up which allows a reasonable dynamically stable gait to be developed. Using this method, joint angles were determined over time, and allowed the robot to take three steps before falling over without feedback and control. This shows that although the intuitive approach is not as objective and theoretically grounded in the dynamics of the robot system, it can be successful in allowing a dynamically stable gait to be rapidly developed. Of more importance is the experience and understanding of the system which was gained through this method.

An example of a gait developed using this method are presented in Figure 5.2. This shows the joint angle data over time which was developed using the Gait Generator software. As can be seen from the figure, the joint angle data consists of many piecewise linear functions. Another important feature of this gait is that it is asymmetrical. This can be seen by observing that the hip swings far more forwards than it does backwards.

5.2 Periodic function approach

One major drawback to the intuitive approach is that the gait which is developed is not easily scalable. For example, parameters which we might like to alter such as step length, step height or step period cannot easily be altered offline, and are even more difficult to alter while the robot is in motion.

To understand why we would want such parameters, we need to consider the entire walking cycle. The walking cycle begins with the robot stationary, and accelerates to the desired velocity. The cycle ends with the robot decelerating and eventually coming to rest. In order to vary the velocity, we require a method of scaling the gait under different situations. In this manner, we can develop a single set of joint angle relationships over time that completely specify the walking gait and allow the variation of the desired parameters.

Since walking is a repetitive motion which repeates over time, we use periodic functions as a basis for developing the gait. In this way, we can specify the period of the step. Further, in order to specify parameters for changing the step length and step height, we specify the trajectory of the foot over time. This allows us to solve for the required angles of the leg link joint angles.

5.2.1 Foot trajectory

As mentioned above, specifying the foot trajectory over time allows the parameters of step length and step height to be introduced to the gait. Using inverse kinematics, we can then solve for the joint angles of the leg links over time, thereby specifying the gait.

When considering the foot trajectory, we consider the period as one step. In this way we

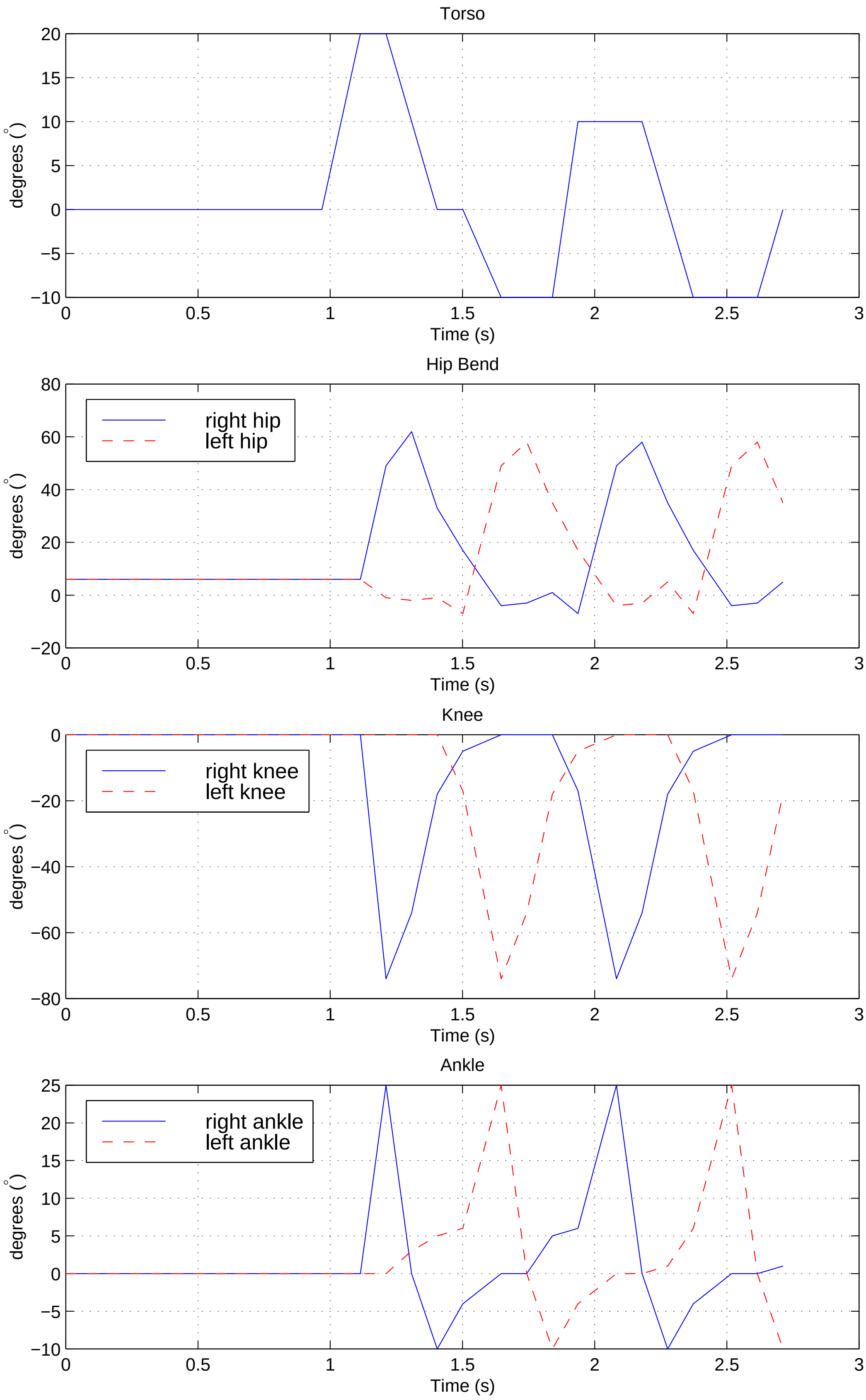


Figure 5.2: Graphs of joint angles over time for the first case of the intuitive approach to gait generation.

only need to consider the swing leg foot trajectory since the support leg foot is in contact with the floor, and therefore stationary[1]. We can specify any trajectory for the swing foot, and solve for the joint angles using inverse kinematics[2].

In specifying the foot trajectory we first make some assumptions about the gait in order to simplify the process:

1. The gait is symmetric in the sense that the trajectory of the COM of the robot swings an equal distance over the support foot. When this occurs, the COM spends the same amount of time behind the support foot as it does in front, such that the backward acceleration experienced by the COM in the first half of the single support phase will negate forward acceleration that occurs in the second half of this phase.

2. The support leg is kept fully extended and rigid during the single support phase. This simplification is made to ensure that the first assumption will hold. If the support knee is allowed to bend, then the COM will experience extra acceleration which will cause the gait to be assymetric.

5.2.2 Inverse kinematics

Given the above assumptions, we can use inverse kinamatics by calculating the joint angles of the swing leg given the position in space of the swing foot. Figure 5.3 shows the robot in single support phase, detailing the nomenclature used for calculating the joint angles.

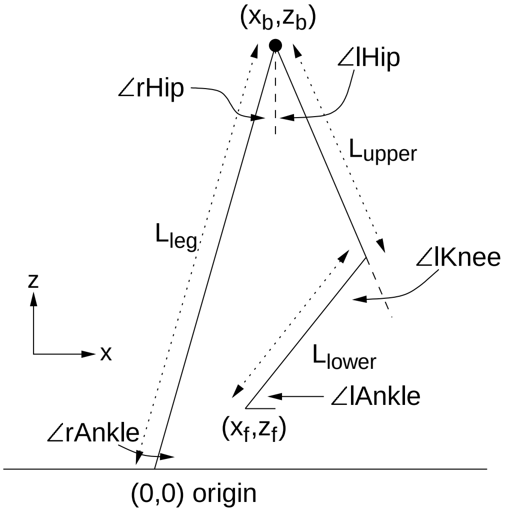


Figure 5.3: Diagram detailing the biped system in single support phase (or swing phase) for calculation of joint angles though inverse kinematics.

---

[1] We assume that there is enough friction between the support foot and the ground to enable the support foot to be stationary with respect to the ground under all normal walking conditions.

[2] Assuming a solution exists for the specified trajectory.

If we take the position of the support foot as the origin for the x-z coordinate system, we can specify the position of the swing foot (x_f, z_f) parametrically as:

$$
\begin{aligned}
x_f &= \frac{4\mathrm{L}_{\mathrm{step}} t}{T} - \mathrm{L}_{\mathrm{step}} && (5.1)\\
z_f &= \mathrm{H}_{\mathrm{step}} \sin\left(\frac{2\pi t}{T}\right) && (5.2)
\end{aligned}
$$

where $\mathrm{L}_{\mathrm{step}}$ is the length of the step[3], and $\mathrm{H}_{\mathrm{step}}$ is the maximum height of the foot during the swing phase. Since the body moves a symmetric distance about the support foot, we can also define the parametric equations governing the movement of the body (x_b, z_b) during the single support phase as follows:

$$
\begin{aligned}
x_b &= -\mathrm{L}_{\mathrm{leg}} \sin(\angle \mathrm{rHip}) && (5.3)\\
z_b &= \mathrm{L}_{\mathrm{leg}} \cos(\angle \mathrm{rHip}) && (5.4)\\
\angle \mathrm{rHip} &= \arcsin\left(\frac{\mathrm{L}_{\mathrm{step}}}{2\mathrm{L}_{\mathrm{leg}}}\right) \cos\left(\frac{2\pi t}{T}\right) && (5.5)\\
\mathrm{L}_{\mathrm{leg}} &= \mathrm{L}_{\mathrm{upper}} + \mathrm{L}_{\mathrm{lower}} && (5.6)
\end{aligned}
$$

where $\mathrm{L}_{\mathrm{leg}}$ is the length of the fully extended leg, $\mathrm{L}_{\mathrm{upper}}$ and $\mathrm{L}_{\mathrm{lower}}$ are the lengths of the upper and lower leg links respectively, and $\angle \mathrm{rHip}$ is the angle of departure of the support leg from the negative z-axis.

Next, we can solve for the swing knee ankle, using a geometric approach to inverse kinematics [18], since the system is not kinematically complex. Using the Law of Cosines, we can express the knee angle $\angle \mathrm{lKnee}$ as follows:

$$
\cos(\angle \mathrm{lKnee}) = \frac{(x_b - x_f)^2 + (z_b - z_f)^2 - \mathrm{L}_{\mathrm{upper}}^2 - \mathrm{L}_{\mathrm{lower}}^2}{2\mathrm{L}_{\mathrm{upper}}\mathrm{L}_{\mathrm{lower}}} := K \qquad (5.7)
$$

We can also express the knee angle using (5.7) as:

$$
\sin(\angle \mathrm{lKnee}) = \pm\sqrt{1 - K^2} \qquad (5.8)
$$

and therefore $\angle \mathrm{lKnee}$ can be found as:

$$
\angle \mathrm{lKnee} = \arctan(-\sqrt{1 - K^2}, K) \qquad (5.9)
$$

where $\arctan(y, x)$ is the two argument arctangent function which is defined for all $(x, y) \neq (0, 0)$ and refers to the unique angle θ such that

---

[3] We define the step length to be the distance between the feet during the double support phase.

$$\cos(\theta) = \frac{x}{(x^2+y^2)^{\frac{1}{2}}}, \quad \sin(\theta) = \frac{y}{(x^2+y^2)^{\frac{1}{2}}} \tag{5.10}$$

Similarly we can now solve for the swing hip angle $\angle\mathrm{lHip}$ using the solution for the swing knee angle to obtain

$$\angle\mathrm{lHip} \;=\; -\alpha - \beta \tag{5.11}$$

where

$$\alpha \;=\; \arctan(z_b - z_f, x_f) \tag{5.12}$$

$$\beta \;=\; \arctan\left(\mathrm{L_{lower}}\sin(\angle\mathrm{lKnee}), \mathrm{L_{upper}} + \mathrm{L_{lower}}\cos(\angle lKnee)\right) \tag{5.13}$$

The final joint angles for which we need a solution are the ankle angles. However if we constrain these such that we require both feet to always be parallel to the ground plane, then we can solve for these using angle geometry

$$\angle\mathrm{rAnkle} = -\angle\mathrm{rKnee} - \angle\mathrm{rHipB} \tag{5.14}$$

$$\angle\mathrm{lAnkle} = -\angle\mathrm{lKnee} - \angle\mathrm{lHipB} \tag{5.15}$$

The periodic solution for the joint angles produced by this method are shown in Figure 5.4. This method is very useful for controlling the three parameters of step length, height and period. In addition this method enables the specification of any reasonable foot trajectory. This can be useful for negotiating obstacles or precise foot holds. However due to the complexity of the solutions this method may be slow for implementation in a real-time system.

5.2.3 Application of periodic function approach

As with the intuitive approach to gait development, we still need to experiment to find the correct torso movements that will compensate for the movements of the lower limbs of the robot. These torso movements will ensure that the gait is dynamically stable, preventing the robot from falling over.

Since the walking gait is a repetitive periodic function, we expect that the torso movement will also be a repetitive periodic function which depends upon the parameters set for the lower limb movements. In order to find the correct relationship between the lower limb and torso movements, we need to experiment and examine the stability of the walking gait.

We can choose to implement the periodic function approach to gait development in one of two ways. In a similar fashion to the intuitive approach, we can generate sets of data in

MATLAB or C that can be compiled and downloaded into the robot. This approach has the advantage that the computations for the joint angle trajectories are performed offline, which allows a higher frame rate to be obtained for updating the joint servos. However, the disadvantage of this approach is that the parameters for the walking gait cannot be varied online, resulting in a fairly large degree of inflexibility. Therefore, despite the slower servo update rate, we choose to implement the joint angle data in real-time on the robot hardware.

While the issue of control of the walking gait is related to gait generation, it is a separate topic in its own right and will be discussed in the following chapter.

5.2.4 Gait parameters

The periodic approach to gait development provides the three parameters of step length, step height and step period. These are important parameters which can be used in a control system to stabilise the dynamic gait, as well as vary the speed and position of the robot.

Step length

The step length parameter dictates how far the robot will move with each step. It is directly related to the distance travelled by the robot. It is also related to the average velocity per step of the robot through the step period parameter.

$$\text{average velocity} = \frac{\text{step length}}{\text{step period}} \tag{5.16}$$

This parameter may also be used to correct for instability in the sagittal plane through taking larger or smaller steps to alter the position of the legs relative to the upper body. For example, if the robot is falling over forwards, a larger step can be taken to counter this effect.

Step period

The step period parameter controls how fast steps are taken. It is related to the average velocity per step through the step length parameter. This parameter can be used to correct for overbalancing in the sagittal plane in an similar fashion the step length parameter. If the robot is falling forwards then more steps can be taken quickly to change the position of the legs relative to the upper body. The parameter can also be used to correct for overbalance in the lateral plane, by ensuring that the step period is matched in both the lateral and sagittal planes. In this way the foot will contact the ground at with the right timing and to preserve stability in both directions.

Step height

The step height parameter controls the maximum height of the foot trajectory. Through adjusting this parameter, and possibly the foot trajectory, the state of the terrain can be taken into account. For example, if an obstacle exists in the path of the robot, by controlling this parameter, the robot can step over the obstacle or even onto the obstacle.

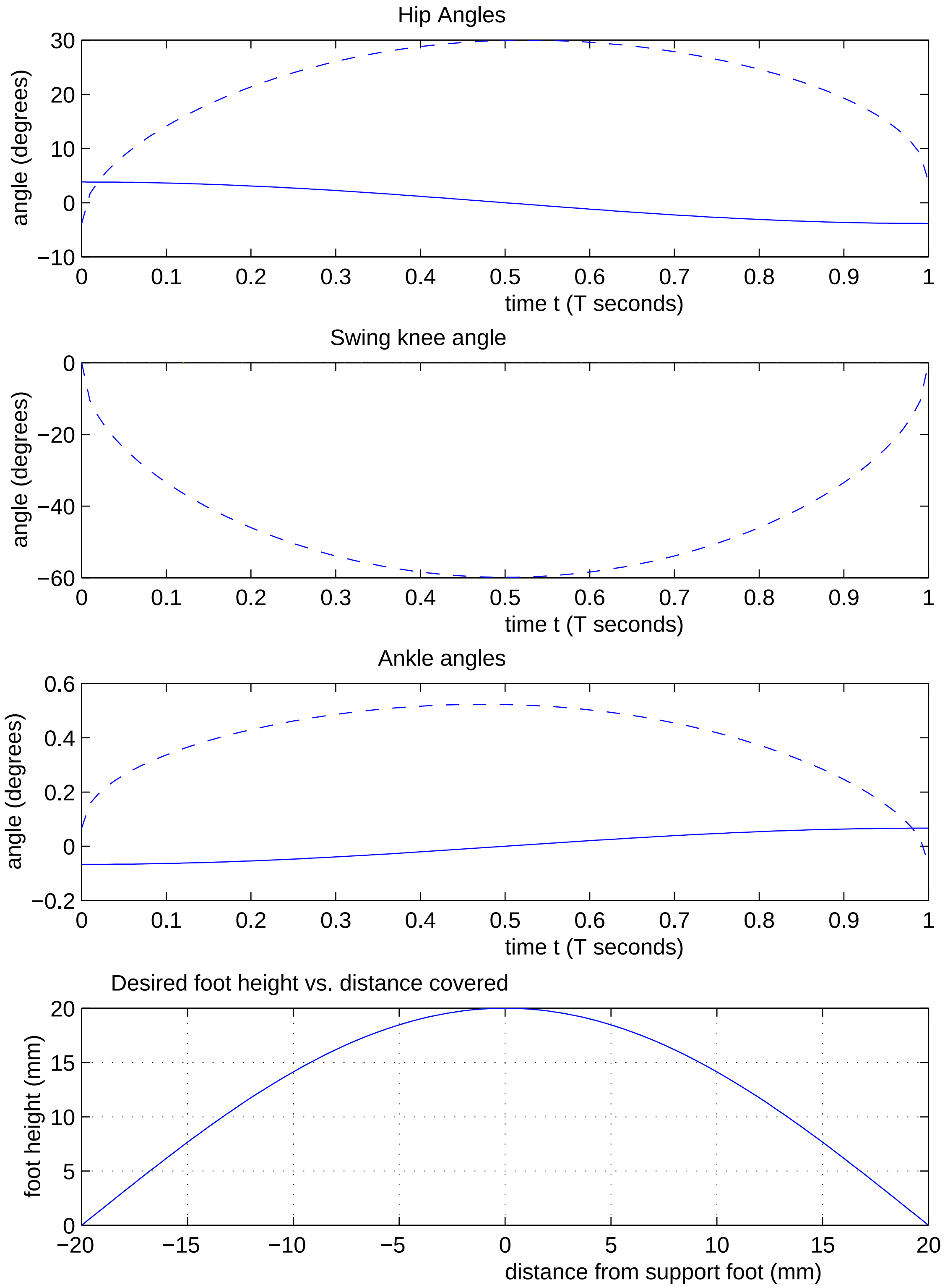


Figure 5.4: Plot of joint angles solved using inverse kinematics with a period of 1 s, a step height of 20 mm and a step length of 20 mm. The top graph shows the hip angles over time, the second shows the swing knee angle over time and the third graph details the ankle angles over time. In these first three graphs the dashed line indicate angles on the swing leg while solid lines represent angles on the support leg. The final graph shows the trajectory of the swing foot with the support foot as the origin.

CHAPTER 6

Gait Generator software

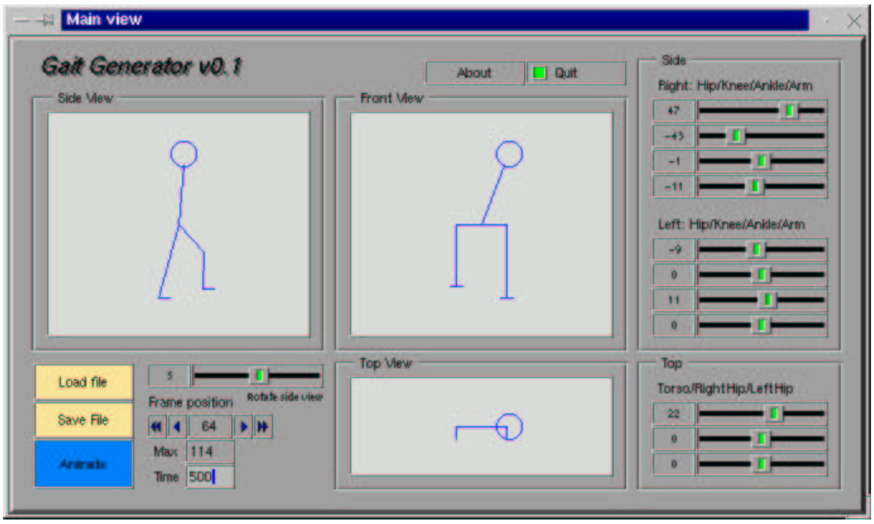
In order to study bipedal walking in robotics more effectively and efficiently, a program was developed to input, animate and simulate gaits for biped robots. Entitled the ‘Gait Generator’, it not only enabled the gait being studied to be input and processed into a form which can be tested on the biped robots, but more importantly allowed the user to visualise the gait through real-time animation. In this chapter we describe the ‘Gait Generator’ program, the underlying data requirements and capabilities of this software, and an evaluation of the usefulness and practicability of the software.

6.1 Program description

This program operates currently only on UNIX operating system platforms running the X11 window system. The program is used to enter a walking gait for the bipedal robot, which is created by entering the angles of the robot links at different times during the gait.

The user of the program has available a number of visual controls at their disposal to aid in entering the gait (see Figure 6.1). Each gait is entered frame by frame. Frames are considered as a still-shot of the angles of the robot links at a particular instant in time during the gait. At the users request, the program will animate the sequence of frames in real-time, smoothly interpolating for the time periods in between frames. This allows the user to simulate the walking gait and assess the likelihood of both the gait working correctly, and the robot walking with that particular gait.

As a visualisation tool, there are three distinct views which the user may use to assess the stability of the robot. These are the front, side and top views (see Figure 6.1). As the user inputs each frame, they will move the input sliders, which will move the corresponding links simultaneously in all three views. In this manner instant visual feedback is acquired by the user developing the gait. The user can move backwards and forwards through the sequence of frames, adding more frames and editing existing frames. In this way the sequence of move-



Frame position 64
Max 114
Time 500

Figure 6.1: A screenshot of the Gait Generator interface. Note the three different views which are present (side, front and top) which show the motion of the robot from these perspectives.

ments that will constitute the gait are built up frame by frame. Note that the difference in time between individual and sequential pairs of frames is user specifiable, thus greatly increasing the flexibility and usefulness of the software.

Once the user is satisfied that the gait is ready for testing, the gait may be saved. Three files are generated by the program in this procedure. The first is a `.gg` file, which the program may load later for the user to modify or re-simulate the gait. The second and third files are `.c` and `.h` source files for compiling with other `C` programs and downloading to the robot. In this manner the gait can easily be tested on the robot.

6.2 Program requirements

As mentioned previously, in order for the program to run it requires a UNIX operating environment running the X11 windowing system. Also required are the `xforms` X11 libraries. The data requirements of the program are simply either a gait input by the user, or a previously saved gait.

The output of the system is a gait generator format file, which saves the angles of the links at each point in time, for later reloading, editing and animation. Also output are the `C` source files which can be compiled with other source files. These source files are compiled into assembly language by a C compiler and then cross-assembled for the Motorola 68332 microcontroller which exists in the hardware of the biped robot. This compiled byte code is downloaded to the robot via a serial link cable to the UNIX machine. This cable is removed when the gait is tested on the robot.

6.3 Program evaluation

This program has been of enormous benefit to the research group in developing the first simple gaits for the biped robots. The eventual aim is to prepare several simple walking gaits which can be run on the robots. When robots execute these gaits, various sensors will be used to collect data while the robot is walking. This data can be analysed and correlated against the movements of the robot in an attempt to determine how this sensory feedback can be applied to control the robot in the future.

It is likely that as a control system is developed for the robot, this program will be used less frequently. However, in the meantime, until this stage eventuates, the Gait Generator software is an important tool in the study of bipedal walking. The tool has helped with entering, simulating and visualising walking gaits, and has saved much time in the preparation for data collection.

6.4 System limitations

While this tool is very useful, as with all software it has some limitations. Most notably the tool performs no mechanical modelling during animation and simulation of the gait. For example, the ability to simulate accurately through the use of dynamics how stable the gait will be would be useful, since this would significantly reduce the number of trials required to be performed on the actual robot.

CHAPTER 7

Motion control

If the biped robot is to successfully achieve walking with dynamic stability, the control system must ensure that the behaviour of the robot does not deviate from this stable condition. Furthermore, if the system does depart from dynamic stability, then the control system must identify this condition and bring the robot back to the dynamically stable condition. Since walking is a continuous and cyclic process, we can consider two main types of control systems—a closed-loop control system and an event driven control system. It is appropriate to consider the use of a closed-loop control system for controlling the continuous process of the walking gait. However, since walking is a cyclic or periodic process, we could also model this as a discrete process, and use an event driven control system to identify the existence of such states and modify the closed-loop control depending upon the current state of the system.

A closed-loop control system operates on a continuous process, taking the error of the system and modulating a control action that is generated to reduce the error of the system. The error e of a system is defined as the difference between the desired and actual conditions

$$e = I_{\text{desired}} - I_{\text{actual}} \tag{7.1}$$

There are two ways in which an error signal can arise. The first manner is where the actual output of the system is altered due a change of conditions. The second manner is when the desired value is changed. In either case, the response of the controller will be the same and depends only upon the magnitude and sign of the error signal.

In contrast, the control of a discrete process involves scheduling of events. For example, a control system in a production line must sequence the steps in the assembly of the final product in the correct order and possibly with the correct timing. For bipedal walking, since the walking gait is periodic and there are a finite number of states in each cycle where adjustments must be made (see Chapter 5), an event driven control system can be used. For example, there are two main states in each gait cycle—the single and double support phases, and in each phase the robot must be controlled differently, and the control system can determine current state of the robot through examining how many feet are in contact with the ground.

In this chapter we will identify some of the control problems encountered and examine several different control methods which were used to attempt to solve them.

7.1 Control problems

During the course of this research into bipedal robotics, several control problems were identified. We consider two of the most important of these here—the motion of the trunk in the lateral plane, and the body attitude throughout the gait, which determines the stability of the robot in the sagittal plane.

7.1.1 Trunk motion in the lateral plane

As described previously (see Chapter 5), planning and controlling the trunk motion is vital to maintaining dynamic balance. For balance through compensating for the movement of the lower limbs by trunk motion [9], we only need to consider motion in the lateral plane to maintain balance in this direction. The trunk motion that is desired is essentially periodic, and was previously discussed in Chapter 5.

An error signal which can be used for controlling the action of the trunk can be formulated from an acceleration sensor that is located at the hip and is orthogonal to the sagittal plane. If we can determine the desired acceleration sensor output at all times throughout the gait cycle, then we can use this to determine the corrective action required to bring the system back to stability.

We model the system as an inverted pendulum [3], allowing us to decompose the motions into the lateral and sagittal planes, and consider here movement in the lateral plane [19]. Consider that we need to decrease or increase the amplitude of the pre-planned trunk motion depending upon the error signal. Given the above, and assuming that we can determine the desired acceleration in the lateral plane, then we can implement a continuous control system.

7.1.2 Body attitude

In a similar fashion to the trunk motion discussed above, the body attitude needs to be controlled in order for the robot to maintain dynamic balance. In fact, assuming that the robot is well balanced in the lateral plane, the body attitude at any given point in time is an important determinant of the acceleration experienced by the center of mass (COM) in the sagittal plane. If this is the case, then the body attitude can be used to help prevent the robot from falling over forwards or backwards. Furthermore, in conjunction with the placement of the front foot for the single support phase, the body attitude will also determine the acceleration experienced during the single support phase and thus the resulting velocity of the COM at the end of the cycle. This means that body attitude also has a large impact on the velocity of the robot.

The error signal for controlling the body attitude can again be formulated from an acceleration sensor located at the hips and orthogonal to the lateral plane. If the desired value of the acceleration output is known at all times throughout the gait cycle, then we can use this to generate a corrective action which will maintain the robot in a stable state.

As for the case considering the trunk motion in the lateral plane, we model the system as an inverted pendulum. By considering only motion in the sagittal plane, we can again reduce the number of actions that must be taken by the controller. Consider that the body attitude needs to be increased or decreased depending upon the error signal. Given the above, and assuming that we can determine the desired acceleration in the sagittal plane for all times throughout the gait cycle, we can implement a continuous control system.

7.2 Control method

The control method used to generate a corrective response to the error signal produced by the system can be important in determining the success of the controller in keeping the system at the desired state. This research considered simple control methods, such as proportional control and integral control.

7.2.1 Proportional control

Proportional control, also known as P-control, produces a corrective action which is proportional to the size of the error signal [20]. In fact there is a linear relationship between the error signal input to the controller and the output from the controller. Generally speaking, in most systems a linear relationship exists only within a certain range of errors, known as the proportional band. The set point I_0 of a proportional control system is the controller output when the error signal is zero, and the change in controller output ΔI_{out} for a given error signal e about the set point is given by the relationship:

$$\Delta I_{\text{out}} = K_{\text{P}} e \tag{7.2}$$

where K_{P} is the proportionality constant.

For P-controllers, the output is usually given as a percentage of the maximum allowable change in output. The error signal is usually also expressed as a percentage that corresponds to the maximum allowable change in controller output. Thus a 100% error will correspond to the maximum change in controller output allowed from the set point I_0. Accordingly, we can relate the constant K_{P} to the proportional band by the relationship

$$K_{\mathrm{P}} = \frac{100}{\text{proportional band}} \tag{7.3}$$

and we can then determine the controller output as

$$I_{\text{out}} = K_{\mathrm{P}} e + I_0 \tag{7.4}$$

The output from a proportional controller with limited output response is shown in Figure 7.1.

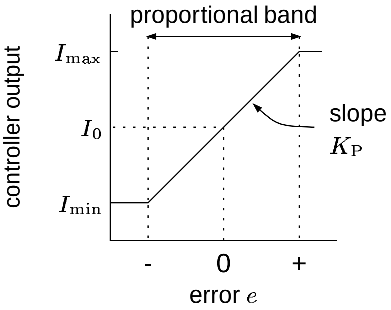


Figure 7.1: The output from a proportional controller for a given error input signal. From the graph it can be seen that in the proportional band the controller output varies linearly with the error signal.

7.2.2 Integral control

Integral control, also known as I-control, produces an output response that continues to increase as long as the error persists. The rate of change of the controller output is linearly proportional to the input error signal

$$\frac{dI_{\text{out}}}{dt} = K_{\mathrm{I}} e = \frac{1}{T_{\mathrm{I}}} e \tag{7.5}$$

where K_{I} is the proportionality constant with units of s^{-1}, and e is the error signal as previously defined. The reciprocal of this constant T_{I} is known as the integral time with units of s. If we solve the above relationship for I_{out} we obtain

$$I_{\text{out}} = K_{\mathrm{I}} \int_0^t e \, dt + I_0 \tag{7.6}$$

The output from the controller in the presence of a constant error is shown in Figure 7.2. It can be seen from the graph that the controller output continues to increase as long as the error persists.

7.2.3 Proportional and integral control

The proportional and integral controllers described above can be combined to form a proportional-integral or PI-controller. For such a controller, the output will be given by

$$I_{\text{out}} = K_{\text{P}}(e + K_{\text{I}} \int_0^t e \; dt) + I_0 \tag{7.7}$$

where I_{out} is the controller output, I_0 is the controller set point, K_{P} is the controller proportional constant, K_{I} is the controller integral constant and e is the error signal as previously defined.

The output of the PI-controller is shown in Figures 7.3 and 7.4. Effectively the output of the controller is the sum of the output of the individual controllers.

7.2.4 Implementing the PI controller

Strictly speaking, the PI controller cannot be implemented as a continuous process as the hardware is limited to sampling the current state of the system with some finite sampling rate T_{S}. Therefore, in order to implement the controller, we must discretise the relationship between the error and the controller output

$$I_{\text{out}}(t_n) = K_{\text{P}}(e(t_n) + K_{\text{I}} \int_0^{t_n} e(s) \; ds) + I_0 \tag{7.8}$$

where $t_n = n\Delta t$ is the discrete time variable. By subtracting this relationship from the same equation involving t_{n-1} we obtain

$$I_{\text{out}}(t_n) - I_{\text{out}}(t_{n-1}) = K_{\text{P}}(e(t_n) - e(t_{n-1}) + K_{\text{I}} \int_{t_{n-1}}^{t_n} e(s) \; ds) \tag{7.9}$$

Through integration by approximation using the trapezoid rule [21] (7.10)

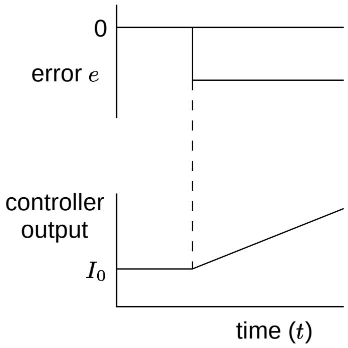


Figure 7.2: The output from an integral controller for a given error input signal. From the graph it can be seen that the controller produces an output which continues to increase for as long as the error persists.

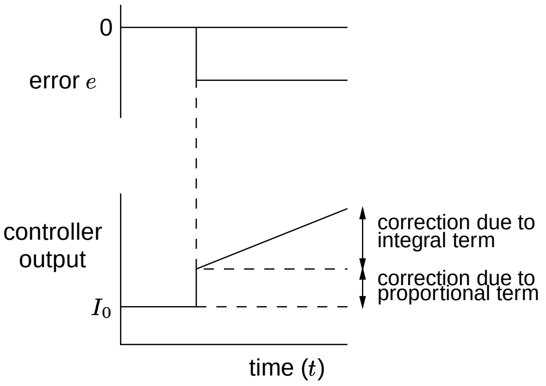


Figure 7.3: The output from a proportional integral (PI) controller for step error input signal.

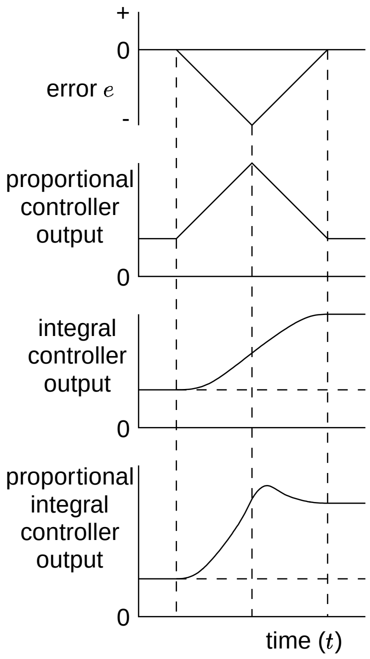


Figure 7.4: The output from a proportional integral (PI) controller for a piecewise linear error input signal.

$$\begin{aligned}\int_a^b f(t)\ dt &\approx \frac{\Delta t}{2}\left[f(t_0)+2f(t_1)+2f(t_2+\cdots+2f(t_{n-1})+f(t_n)\right]\\ &= \frac{\Delta t}{2}\sum_{i=1}^{n} f(t_n)+f(t_{n-1}) \end{aligned} \tag{7.10}$$

we can obtain a discrete variation of $I_{out}(t)$ for time $t = t_n$

$$I_{\mathrm{out}}(t_n) = I_{\mathrm{out}}(t_{n-1}) + K_{\mathrm{P}}(e(t_n) - e(t_{n-1}) + \frac{K_{\mathrm{I}}\Delta t}{2}(e(t_n) + e(t_{n-1}))) \tag{7.11}$$

where Δt is equal to the sampling rate T_{S}. This approximation is easily implemented in an iterative loop.

7.3 Defining the error signal

As mentioned previously, the ability to define an error signal depends entirely upon determining the desired state of the system. In the case of the biped walking robot, it is difficult to determine the desired feedback from the acceleration sensors as this is not previously known. This makes the problem of implementing a control system for dynamic walking difficult, as a suitable error signal cannot be accurately defined. The desired state of the system must be estimated, which can lead to significant errors when implementing the control system.

If the desired acceleration signal is poorly or even incorrectly specified, then the control system, in attempting to correct the actual acceleration to converge to the desired state, may in fact cause the system to become more unstable. For the bipedal robot, apriori information regarding the stability of the system may not be available, and it may not be known how much tolerance is permissible when estimating the desired state of the system. This is one of the major problems to overcome when attempting to implement a control system.

7.3.1 Feedback

All control systems require feedback in order to operate effectively. In this biped system, as noted in Chapter 3, the servos do not provide any feedback mechanism. However semi-solid-state acceleration sensors have been obtained which it is hoped can be used to determine the stability of the system, providing feedback for the control system.

Additionally, the four switches that have been placed underneath each foot, will provide important information about when the feet of the robot contact the ground surface. It is also hoped that this information will help determine the current state of the system and aid in controlling the walking gait.

CHAPTER 8

Results

The most important part of research and experimentation is obtaining and analysing results to verify previous explanations and theories. During the course of this research, several experiments were completed and the results were collected. The first group of experiments performed used the intuitive approach to gait generation. The second group of experiments performed applied the periodic approach to gait generation. The third group of experiments performed attempted to allow the biped robot to walk on the spot. Finally, experiments were performed in an attempt to control the robot while standing in an upright stationary position, stabilising the robot in the presence of external disturbances.

In this chapter we present the results from each of these four groups of experiments and analyse the collected data. The performance of the robot will be discussed and reasons for the success or failure of the experiments will be hypothesised.

8.1 Walking experiments

8.1.1 The intuitive approach to gait generation

The aim of the first group of experiments that were performed was concerned with examining the stability of the robot when using a gait generated via the intuitive approach method discussed in Chapter 5. While this approach was not directly based upon dynamic principles, the advantage of this method is rapid gait development though the use of visual feedback.

To test this method, two different gaits were developed using the Gait Generator software (described in Chapter 6), the first of which was presented in Figure 5.2, Chapter 5. There are several distinguishing features about this gait. Firstly, it is asymmetrical about the support leg. This can be observed from the relative magnitude of the swing of the hips about the vertical position (0°). The continuous gait is also made up of piecewise linear functions, the implementation of which in software on the Eyebot hardware is very fast, allowing a servo update rate of approximately 83.3 Hz (12 ms period)[1]. This allowed the biped walking motions to be smooth, and not jerky or erratic in appearance.

Walking without control

When the intuitive approach was attempted without control or feedback, the robot successfully took three steps. A fourth step was also attempted, however this was unsuccessful due to the system becoming too unstable, causing the robot to fall over in an unpredictable direction. The results of one such experiment can be seen in Figure 8.1, which shows the acceleration data collected in both the sagittal and lateral planes as well as the foot switch data. Also drawn on the graphs is the estimated ideal acceleration which was determined through averaging multiple trials.

Although the biped robot did take three steps, these steps were not ideal, and the robot exhibited pronounced oscillating swaying motions in the lateral plane. This can be observed in the side acceleration data. It was observed that after three steps these oscillations had accumulated to such an extent to severely undermine the stability of the system. This instability in the lateral plane caused the unpredictable behaviour of the robot after the third step.

It was considered remarkable to achieve three steps, and it was not expected that this method would be so successful. In order to improve upon this method, it was decided to implement a control system to react to the measured acceleration feedback during the walking gait.

Proportional control

In order to simplify the implementation of control, it is helpful to decompose the problem of control of the robot into the x and y directions, or into the sagittal and lateral planes. This allows us to implement independent proportional control in each plane. For feedback purposes we use two acceleration sensors, oriented along the x and y axis respectively.

However in order for the control to be successful, the desired acceleration must be found at each instance throughout the walking gait. Unfortunately, this was not known, which made the task of implementing the control difficult. In order to estimate the desired acceleration signal, multiple walking trials were conducted and the acceleration data from trials judged to be the most stable were averaged.

The results of walking with the implemented control can be seen in Figure 8.2. A distinct oscillating pattern can be observed in the data for the side acceleration. While the implemented control system did reduce the acceleration experienced in each plane, the stability of the robot did not appear to improve significantly, with oscillations again building until the biped robot became unstable and fell over.

---

[1] It should be noted that the maximum servo refresh rate is only 50 Hz. The higher calculation rate is accounted for by the Eyebot operating system routines.

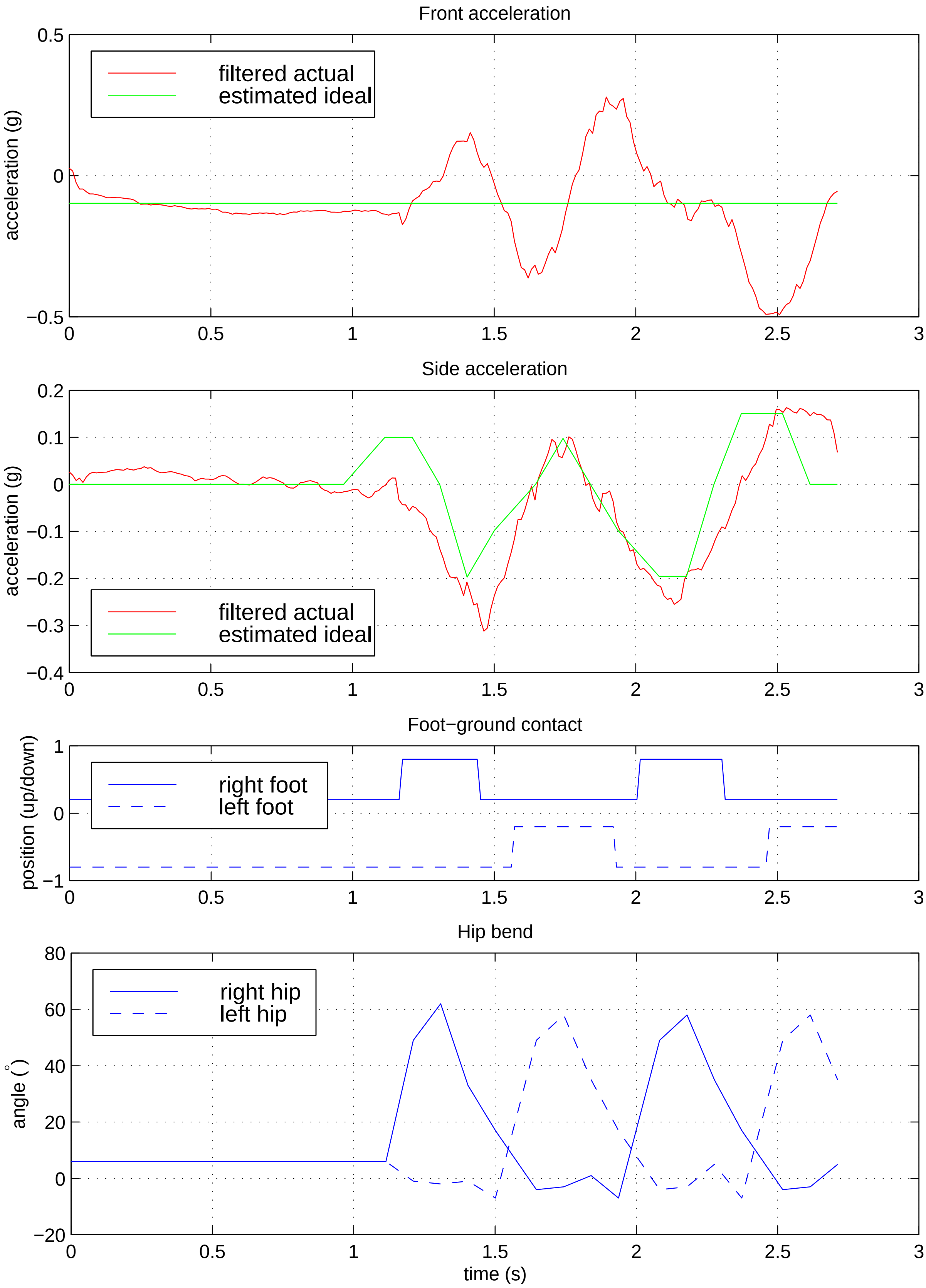


Figure 8.1: Results of the intuitive approach to walking, implemented without control.

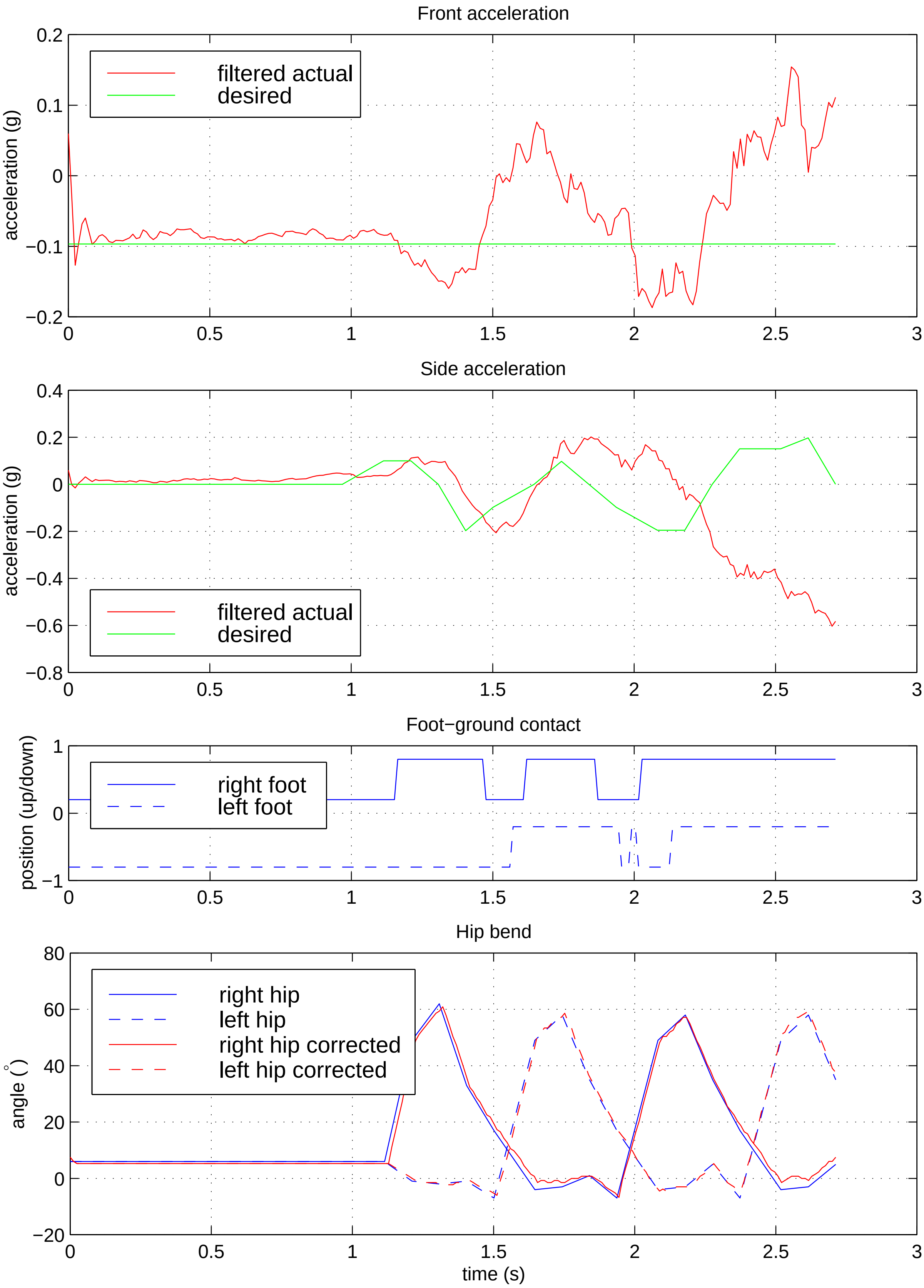


Figure 8.2: Results of the intuitive approach to walking, implemented with proportional control.

There may be two reasons for the lack of success of this approach. Firstly, the gait which has been developed may not be stable enough, causing the marginal stability of the system in the first three steps and the subsequent instability of the fourth step. Secondly, the control system may be inadequate and the desired acceleration that was determined may be incorrect. Both these issues are addressed by applying the periodic approach to gait generation which provides gait parameters which can be varied by the control system. This may provide a more stable gait and a more complex control system to overcome the difficulties exposed in the intuitive approach to gait generation.

8.1.2 The periodic function approach to gait generation

The periodic approach to gait generation (described in Chapter 5) was attempted in order to provide gait parameters which could be varied as a method of controlling the robot motion. For this method to be successful, it must be possible to modify the gait in real-time response to feedback. As a result of this requirement, the gait must be generated on-line.

As a first attempt, the gait was implemented using trigonometric functions[2]. Unfortunately, due to the computational requirements of such functions, the implementation was slow, allowing a maximum update rate of approximately 11.6 Hz (86 ms period). The slow update rate had the adverse effect of causing a jerky robot motion which was not as smooth as for the previously high update rate obtained for the intuitive approach implementation.

This caused a jerky motion which was unsatisfactory, increasing the instability of the robot. Furthermore, the reduced update rate caused a significant delay in the response of the implemented control. Therefore, in order to test the gait with a higher update rate, gait data was computed off-line for a single set of parameters. The robot movements resulting from this implementation was then smooth, with an update rate of 50 Hz (20 ms period). Unfortunately this meant that the parameters could not be varied in real-time while the robot was walking and therefore using the parameters of the periodic function approach to motion control could not be tested. However the single parameter implementation was tested and the stability is compared with the intuitive approach implementation.

No control

A walking gait using the periodic function approach was developed off-line with a step length of 20 mm, a step height of 20 mm and a step period of 0.78 s. This gait can be seen in Figure 8.3.

This gait was trialled, but due to the lack of control, only one step was successful, therefore no data is presented here. It was determined that for gait stability during walking, a control system should be implemented. However, due to the slow update rate caused by the use of

---

[2] See Chapter 5 for a derivation of such an implementation.

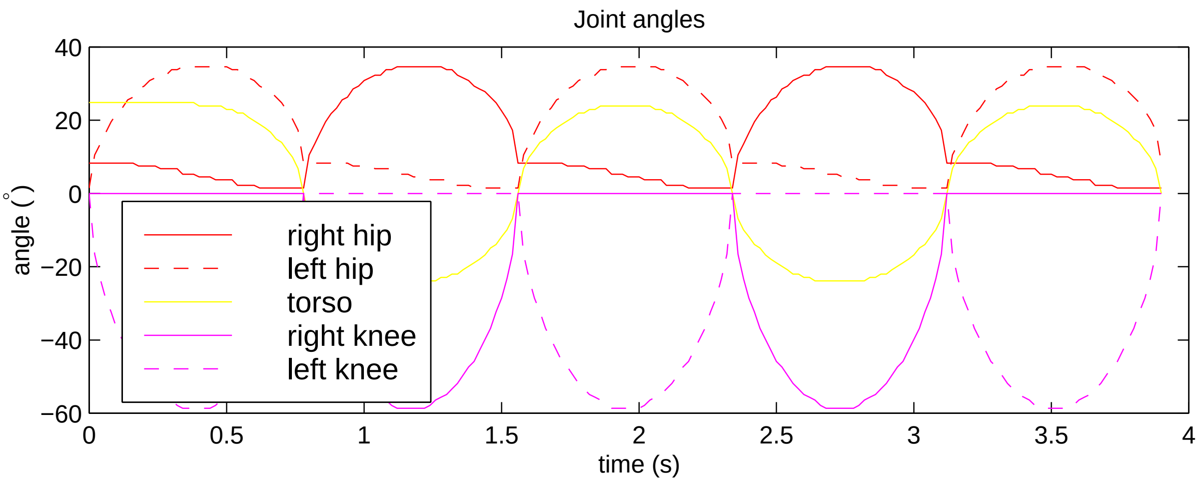


Figure 8.3: Results of the intuitive approach to walking, implemented without control.

trigonometric functions, the implementation of a control system was found to possess significant delays, that did not improve the stability of the system.

However, in order to improve the stability of the gait, it was found that the relationship between the step period and the magnitude of the torso swing should be examined. This important relationship should be established before the parameters of the gait are varied in order to ensure a stable gait.

8.1.3 Stationary walking on the ground plane

Due to the instability of both the intuitive and periodic function approaches, it was decided to examine in more detail the ability of the robot to balance in the lateral and sagittal plane while walking on the spot. The reason behind this that the overall stability of the walking gait depends upon the timing of the motions in both of these planes. If the timing is incorrect in either plane, it will affect the stability of the robot in the other plane as well as the overall stability of the walking gait. It is therefore desirable to study the relationship between the stability of the robot and the motions in these planes.

The stability of the robot in the lateral plane can be examined by enabling the robot to walk continuously in the same position on the ground plane. This has the desired effect of reducing tipping moments which are created when the robot translates a foot, which can increase the instability of the robot. Thus the problem of control is simplified, since the parameters of step height and step length are constant, and the motion is effectively restricted to the lateral plane. Through variation of the step period and the magnitude of the trunk motion in this plane, the timing relationship between these parameters can be examined.

No control

The results of experiments of stationary walking on the ground plane for different update rates can be seen in Figures 8.4 and 8.5[3]. The motion of the robot was restricted to the lateral plane in order to examine purely the stability of the robot in that plane. It was observed that system settled down to a stable state after several steps, depending upon the initial starting conditions and the chosen magnitude of the trunk motions and the step period.

Proportional control

Although the system was observed to be stable, the measured accelerations experienced by the robot in the lateral plane was still quite large (peak magnitude nearly 0.2 g). It is desirable to reduce such accelerations to increase the stability of the system. This can be achieved by implementing a proportional control system.

Proportional control was implemented in the sagittal plane, in an attempt to balance the robot in this direction while the robot is in motion. In this case, the desired acceleration in this plane is known, and should be zero. The result of implementing such a control system is shown in Figure 8.6.

Although the robot did not successfully balance while walking for more than two or three steps and required support, it can be seen from the acceleration data collected that the magnitude of the acceleration has been reduced in the lateral plane. This indicates that the stability of the robot in the sagittal plane effects the stability of the robot in the lateral plane, and is an important implication for developing a stable gait.

Due to the fact that the robot required support in order to balance in the sagittal plane while walking on the spot, it was decided to further investigate stability in this plane through experiments involving the robot balancing while stationary.

8.2 Stationary balancing in response to an external force

The results of the last three experiments motivated a final experiment to determine the stability of the robot while standing in an upright stationary position. While the robot is motionless, instability cannot be caused by motion of the limbs, but rather by the vertical projection of the center of mass (COM) leaving the support area formed by the feet. The objective of this experiment is to determine if the robot can react to a small external force in a manner which will enable the robot to return to a stable and statically balanced upright state. Simple control mechanisms of proportional and proportional integral control are used to achieve this.

---

[3] Unfortunately due to a hardware limitation, the speed of the microcontroller was reduced from 25 Mhz to 16 Mhz, resulting in a slower update rate.

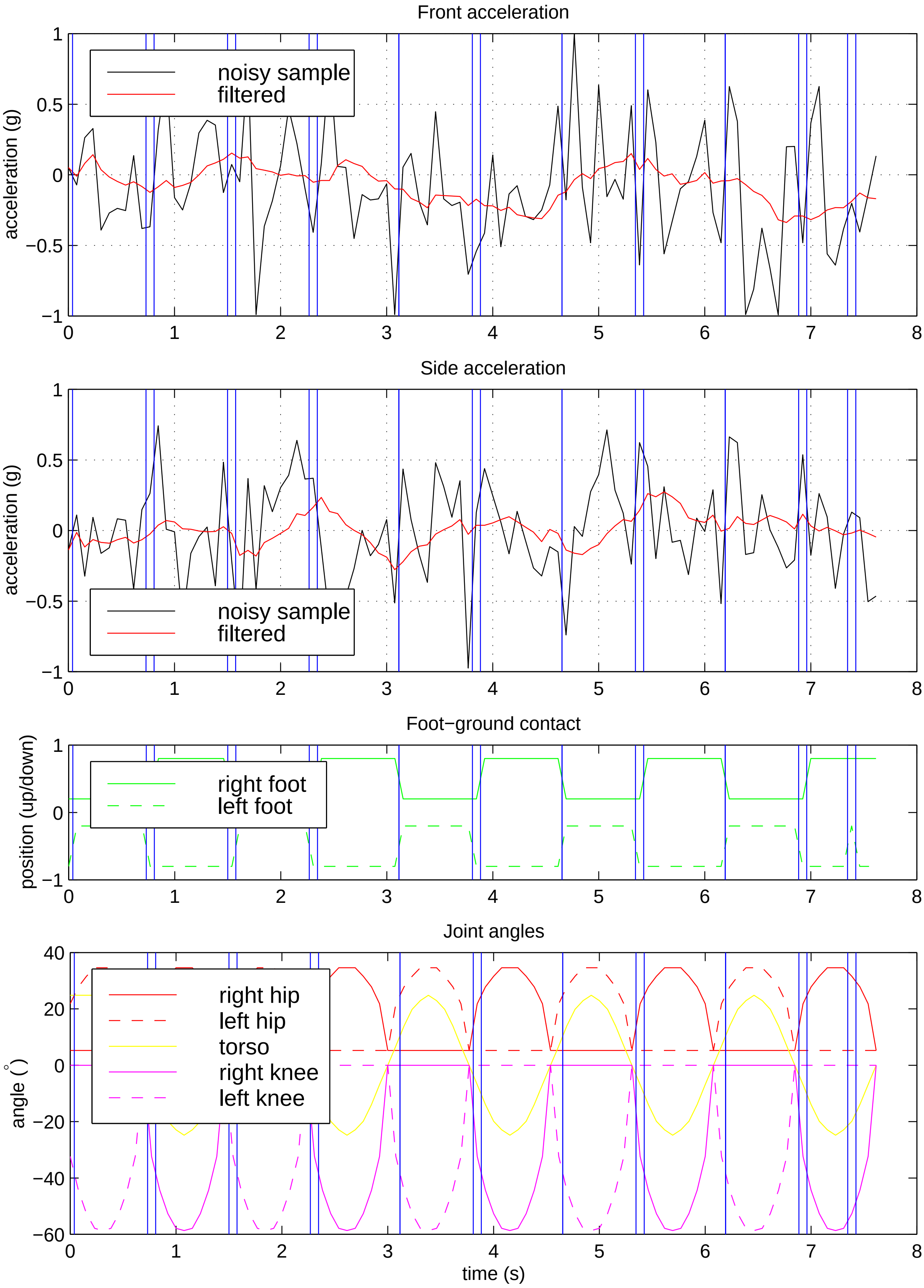


Figure 8.4: Results of stationary walking on the ground plane, implemented without control at an update rate of 13 Hz (78 ms period).

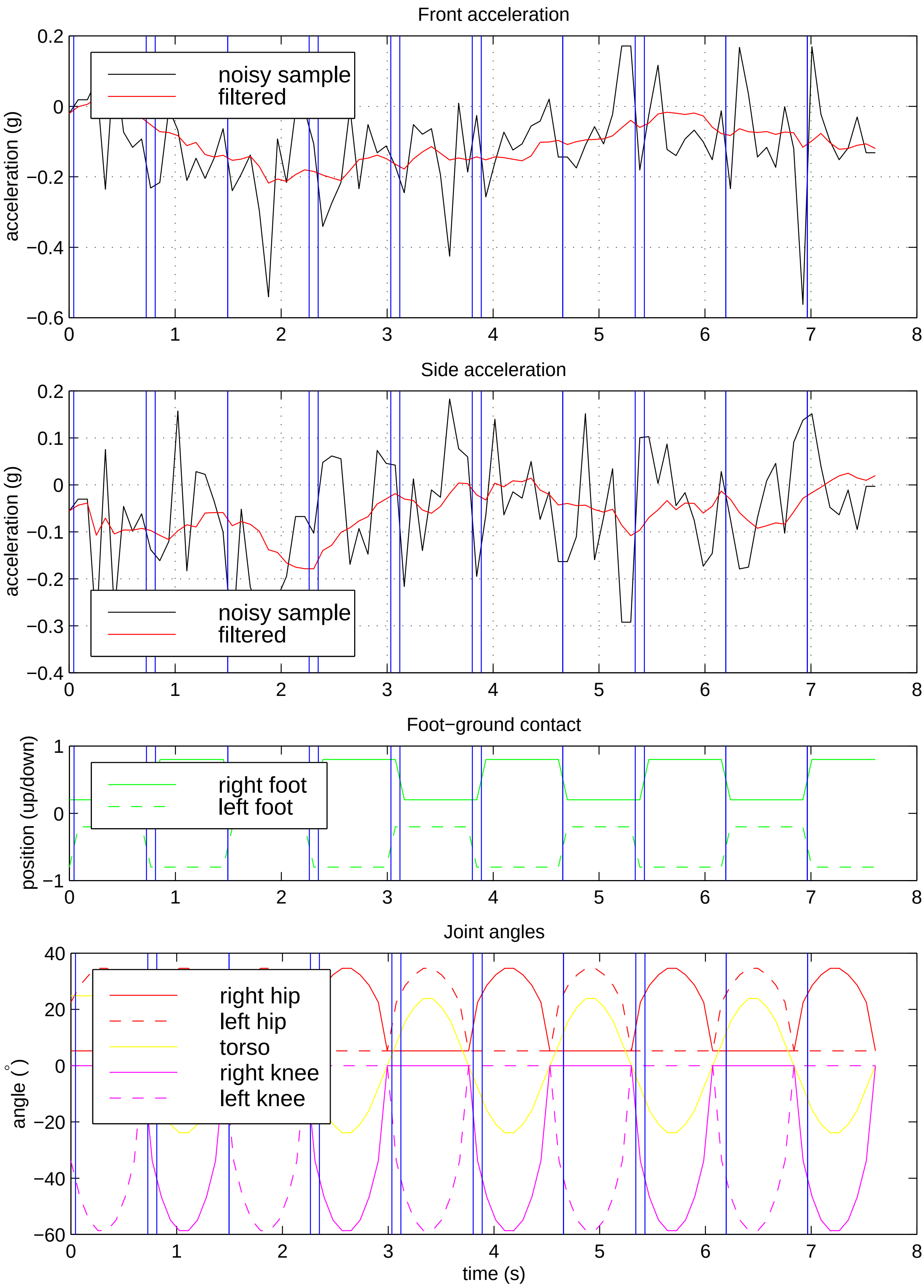


Figure 8.5: Results of stationary walking on the ground plane, implemented without control at an update rate of 11.7 Hz (85 ms period).

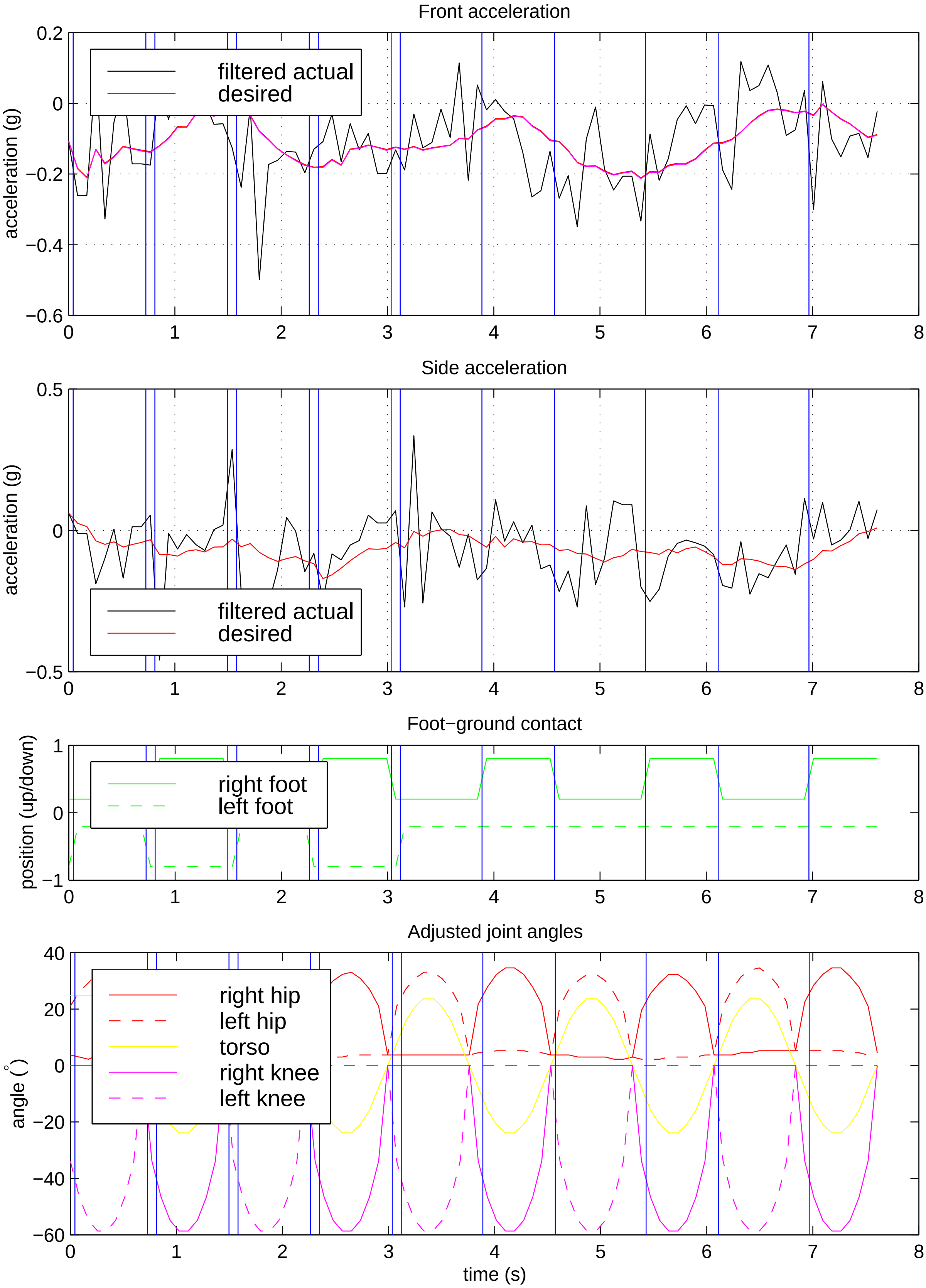


Figure 8.6: Results of stationary walking on the ground plane, implemented with proportional control.

With feedback from the acceleration sensors, a control system should be able to determine an appropriate corrective measure for the robot to take in order to maintain the stability of the system. For the case where the robot is upright and stationary, the desired acceleration of the system is known in advance—that is there should be no acceleration present in both the lateral and sagittal plane. Any deviation from this state should cause an appropriate corrective action to be taken.

For this case, we can decompose the motion of the robot into the sagittal and lateral planes, and since the reaction force of the ground plane to the robot is assumed to negate the effect of gravity, there should be no velocity experienced by the robot in the z-axis. Therefore, we can implement two separate control systems, one for each plane, each creating a trunk motion in their respective plane to stabilise the robot. The idea behind each control system is to move the COM in such a manner as to keep it within the support area formed by the feet.

8.2.1 No control

In the first experiment, the robot was commanded to an upright, stable and statically balanced position. It was found that the robot could stay balanced indefinitely in the absence of an external force. However, when a small external force was applied, the vertical projection of the COM of the robot was found to leave the area of support formed by the feet, causing the system to become unstable and causing the robot to fall over.

The results of this experiment are shown in Figure 8.7. What can be noted is that the accelerations are very small, but more importantly the robot is resting with most pressure on the front of the foot. This can be observed from the oscillating data from the heel contact. This implies that the COM is towards the front of the foot, due to a body attitude of 7°.

It was observed that due to the construction of the foot, which is similar in proportion to a human foot, the stability of the robot is poor along the negative x-axis. This is because the heel of the foot does not extend behind the ankle, and therefore it is relatively easy for the COM to leave the support area in this direction. In contrast, the stability along the positive x-axis is much better, since the toe of the foot extends quite far in front of the foot. Therefore the COM must move much further forwards in order to destabilise the robot in this direction. This problem will be called the *heel condition* and is not usually encountered in normal walking as the direction of this motion is along the positive x-axis. It is perhaps interesting to note that humans will react to backwards overbalancing by taking a step backwards, while overbalancing forwards does not usually require a forwards step to restabilise the body.

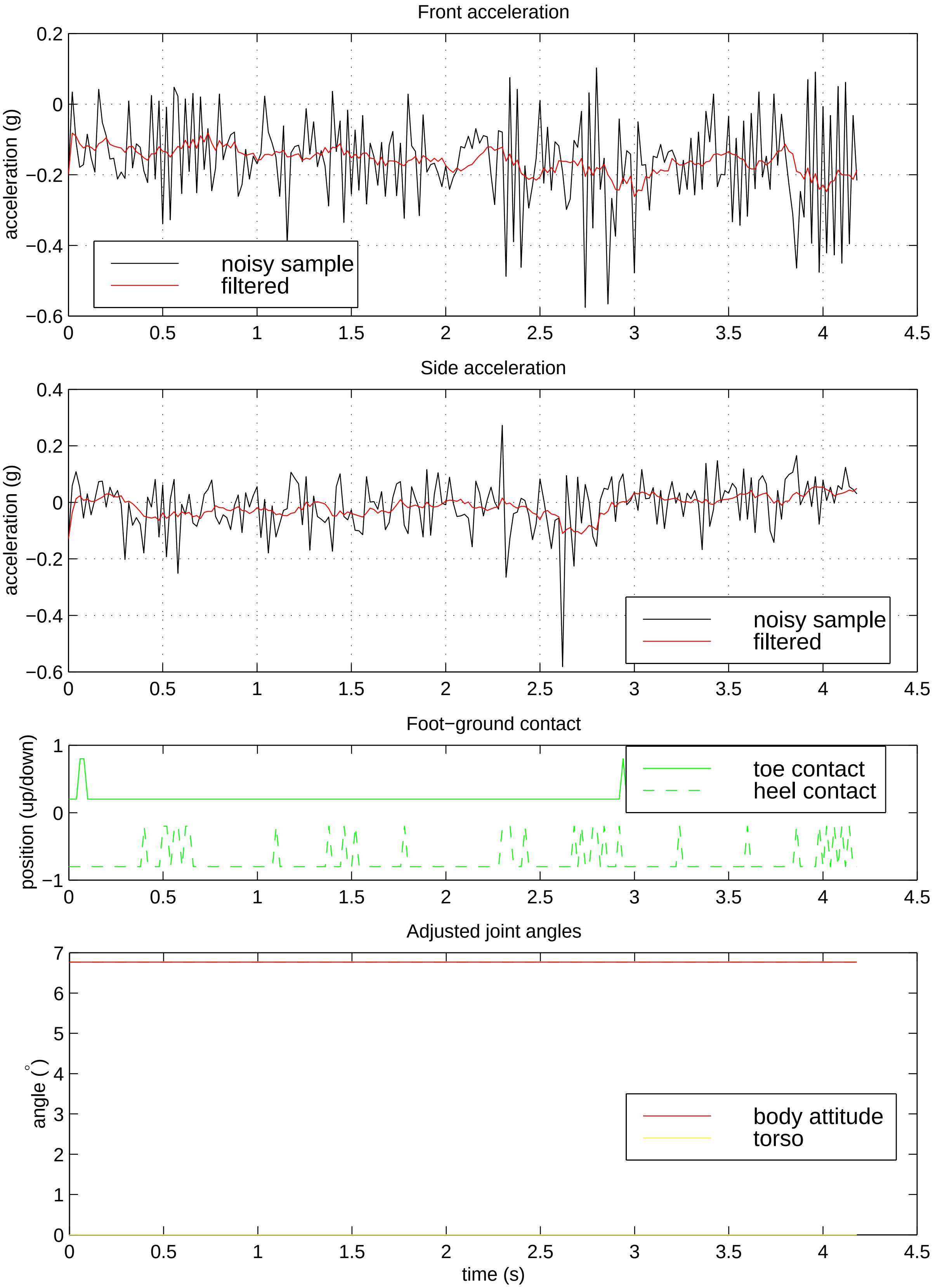


Figure 8.7: Results of stationary balancing in response to an external force, without control.

8.2.2 Proportional control

The proportional controller discussed in Chapter 7 is implemented in both the lateral and sagittal planes to compensate for the presence of an external force. The update rate for the control system is 50 Hz (20 ms period), fast enough so that any movements initiated by the controller will be smooth and not introduce further radical destabilising motions.

The results of experiments with lateral control disabled can be seen in Figure 8.8. Here we can see that sagittal stability is returned after a series of small applied forces applied along the positive x-axis. It is interesting to observe that in order to maintain balance, the heel of the feet of the robot leave the ground, causing the robot to balance only on the toes, much the same as for humans balancing in this way.

The balance of the robot was satisfactory in the direction of the positive x-axis for small forces. However as discussed above, due to the heel condition, only negligible forces could be applied along the negative x-axis without causing the robot to over balance. Occasionally the system would exhibit oscillations due to the correction made by the controller causing additional accelerations of opposite sign. If the constant of proportionality is not chosen carefully, then such oscillations in the system may build up and eventually cause the system to become unstable.

8.2.3 Proportional integral control

It was desirable to implement a proportional integral controller to improve upon the proportional controller as the addition of the integral term will increase the magnitude of the controller output as long as an error persists. This is a desired effect, as the corrective action provided by the proportional term may not be sufficient to return the system to stability.

The results of the implementation can be seen in Figure 8.9. Once again, the system is able to stabilise after the application of a series of small forces in the direction of the positive x-axis. Once again, in response to the applied force, the heels have left the ground and the robot has spent a large proportion of time balancing on the toes of the feet before returning to stability.

The addition of the integral control has improved the stability of the system, allowing the system to stabilise more rapidly and in the presence of larger applied forces. However, if the constants of proportionality and integration are not chosen carefully then the system can oscillate to eventually become unstable in the absence of any external force.

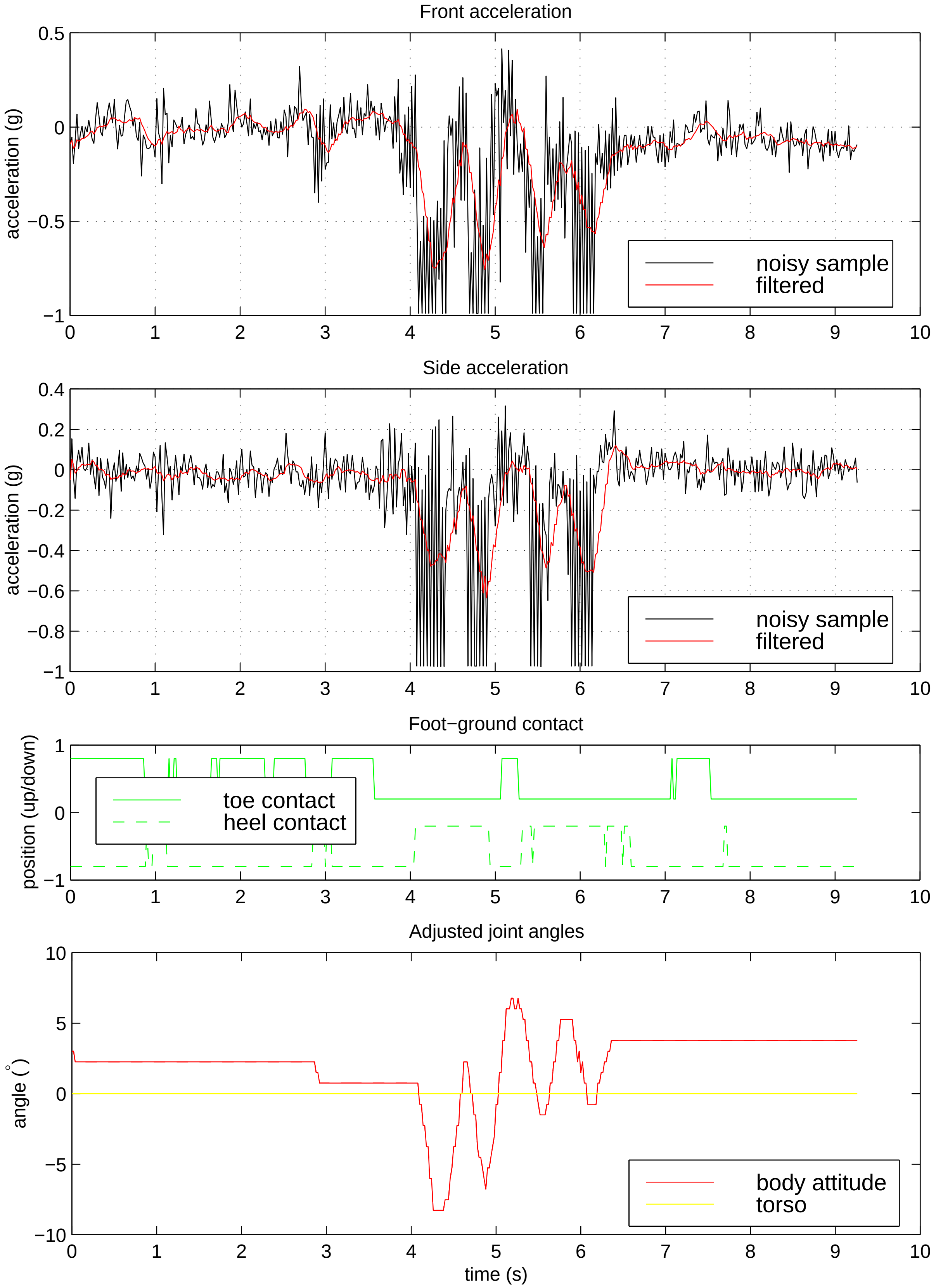


Figure 8.8: Results of stationary balancing in response to an external force, with proportional control.

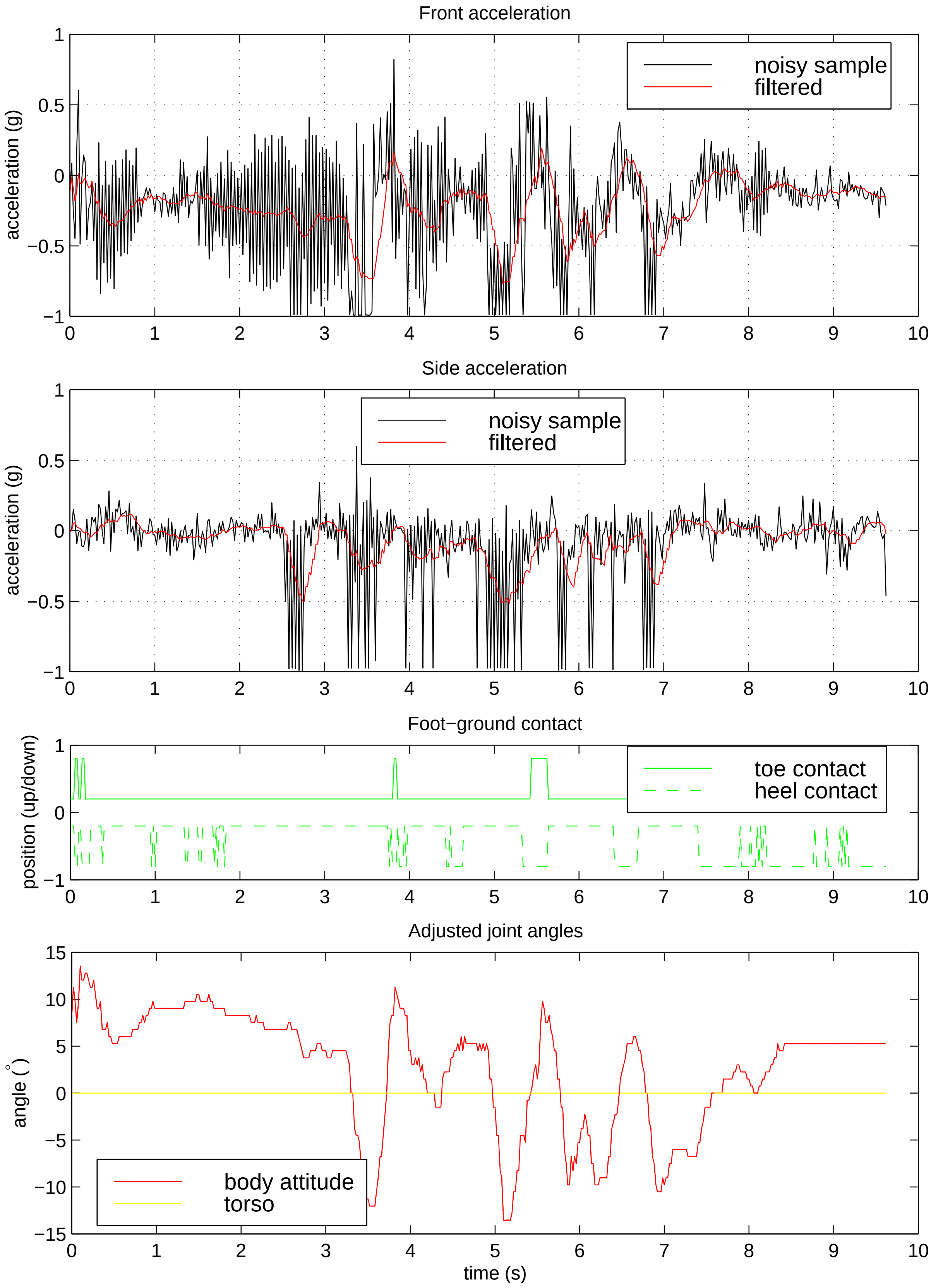


Figure 8.9: Results of stationary balancing in response to an external force, with proportional integral control.

CHAPTER 9

Conclusion

It is not trivial to implement dynamic walking in a bipedal robot, however in this dissertation the relevant issues for designing and constructing such a machine have been discussed, and the possibilities for implementing a control system to coordinate the dynamic gait have been examined.

Two methods of gait generation were investigated. The intuitive approach produced a gait which allowed the robot to walk for three steps. However, this method of gait generation from suffered slow development time of the gait, and was not scalable. In order to walk effectively, a method must be provided to alter the travel velocity, as well as to account for varying terrain.

As a second approach to gait generation, periodic functions were used to specify the trajectory of the foot. This resulted in a scalable gait where the parameters of step length, step height and step period could be varied as a means of adjusting the gait for the terrain, as well as controlling the stability of the robot. The implementation of this gait was slow due to the number of trigonometric functions involved. To overcome this problem, the equations could be approximated or lookup tables used to lower the computational complexity of the approach. However, this would adversely affect the accuracy of the algorithm.

Despite the difficulties faced in implementing the periodic approach to gait generation, this approach seems the most promising. The parameters provided by this method will be invaluable in implementing a control system which will correctly balance the robot while walking. Further research is required to determine the correct trunk motions which compensate for the movements of the lower limbs. Such research may be provided by the zero moment point (ZMP) method of gait synthesis (to be discussed in Chapter 10).

A useful tool for the development of gaits for bipedal robots has been developed. This tool has proved helpful for the input and visualisation of gait data. Mechanical modelling could be added to the tool as an improvement to gait animation and simulation.

Methods of controlling biped walking robots have been investigated. Proportional and proportional integral control systems were implemented to adapt the trunk motion to compensate for movement of the lower limbs of the robot while walking. While these control systems were not successful in stabilising the robot while walking, they were successful in balancing the robot while stationary.

A more complex control system may be required in order to stabilise the robot sufficiently. This may require an adaptive control system, such as artificial neural networks (ANN), genetic algorithms (GA) or fuzzy logic. However, it is more likely that the method of gait generation will also need improvement, in order to generate the most stable gait before control is implemented, minimising the control problem.

It is possible that the acceleration sensors may require additional filtering in order to suppress the noise they are producing in response to vibrations caused by the actuators. This noise is a significant barrier to controlling the motion of the robot in order to achieve stability while walking. It is not practical to isolate the sensors from the robot frame, which is transmitting the vibrations from the actuators to the sensors. One possible solution may be to use a low pass filter with a suitable cut-off frequency. Alternatively, servos which have less jitter may be found to overcome the problem.

The importance of gait generation has been established, as well as the significance of a control system to stabilise the robot while in motion. Both must be present for dynamic bipedal walking to succeed, and both require more research.

Research in this field is important for developing robots which can operate in normal human environments, and can adapt to disturbances and variations in the environment, enabling them to traverse over uneven terrain. In the future, with the convergence of many widely differing fields of research, this is becoming a reality.

CHAPTER 10

Future work

As the reader might well imagine, the scope for future work in bipedal robotics is immense and there are many different avenues of research which may be investigated. As mentioned in Chapter 2, there are many different approaches which can be taken to gait generation and control of the robot alone.

After some research and experimentation, the future approaches which can be undertaken and are most likely to generate increased success are more readily identified. Through attempting these new methods, it is hoped that the goal of achieving dynamic bipedal walking will be realised. This chapter will identify some areas of research which should be investigated in the future.

10.1 Design

The general design of the robot could be improved in several ways. As mentioned earlier in Chapter 3, the mass distribution of the robot is crucial as this determines the position of the center of mass (COM) of the robot, as well as the total mass. The total mass of the robot is important as this determines the maximum torque required from the servos and therefore dictates the size of the actuators which should be used. However, of more importance is the placement of the COM as this will ultimately determine the stability of the robot.

Presently, the trunk of the robot is used to stabilise and compensate for the motions of the lower limbs of the robot through alteration of the mass distribution as the trunk is moved. For this reason, the COM must be placed high enough on the trunk to allow it to be shifted sufficiently for stabilisation to occur. However, if the COM is placed too high then the robot will destabilise as the trunk is moved. This is a fine balance which must be considered in conjunction with how rapidly the trunk can be moved and the degree to which the control system is able to control the trunk motion with sufficiently fine granularity.

For the present verision of the biped robot, the COM is quite low, despite two attempts to shift the COM to a higher position (see Chapter 3). As a result, the trunk must be moved rapidly and to a relatively large degree (25°) during motion, resulting in the generation of significant moments which adversly affect stability. Several methods are proposed to solve this problem.

Firstly the robot could be redesigned to significantly reduce the mass of the legs in proportion to the mass of the upper body and trunk. This could be achieved by reducing the amount of material used in the legs and hips while adding more mass to the shoulder and head areas. Adding more mass to the shoulder and head areas of the robot was the method tried in attempting to raise the COM. This method was successful however to some degree, due to the limitation in the power of the servos it is impractical to keep adding more mass to the upper body without eliminating some from the legs and hips to ensure that the total mass of the robot does not overload the servos.

A second method of improving stabilisation of the biped robot through redesigning the robot is to increase the degrees of freedom of the robot. While this would require more actuators, enourmous benefit can be gained through increased flexibility in the range of motions which can be performed. In fact, this may allow the COM to be shifted by the same amount but with less trunk motion. One alternative design is shown in Figure 10.1.

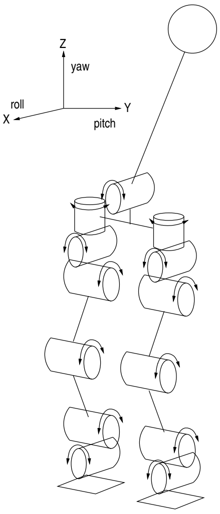


Figure 10.1: Alternative design to the current biped robot. Notice the extra degrees of freedom at the hip and ankle allow the body to sway while maintaining full foot contact with the ground plane.

In this design we can see that the extra degrees of freedom at the ankles and hips in the lateral plane allow the hips to shear in order to shift the COM in this plane. In this manner we can reduce the magnitude of the torso swing required to compensate for the various leg movements. Another advantage of this design is that the foot can maintain full contact with the ground plane while the robot sways in the lateral plane. This will increase the friction between the feet of the robot and the ground plane, reducing the effect of various forces and torques which might act on the foot causing the robot to yaw or slip.

10.2 Gait generation

As mentioned before, it is desirable to generate a gait that is dynamically stable. This is because such a gait will require minimal control in order to maintain the stability of the system. Several methods of gait generation were tried, with varying methods of success. One powerful method of generating a dynamically stable gait is synthesis through *zero moment point (ZMP)* calculation.

The ZMP is defined as the point on the ground plane where the total moment due to gravity and inertia equals zero, or upon which there is no torque acting [5]. Another synonymous definition of the ZMP is the position where the resultant of the ground reaction force of the robot penetrates the ground plane [22]. Therefore, by definition, the ZMP must always lie inside the support area formed by the feet in contact with the ground.

This method involves analysing the system by considering the equations of motion governing the robot in order to determine the where the ZMP lies. If we know the criteria for stability of the robot in terms of the ZMP, it is possible to first plan the time trajectory of the ZMP and then determine the required state of the system to meet this condition [5, 9, 10]. This method of gait generation has produced stable dynamic gaits that have enabled biped robots to walk (see Chapter 2), and should therefore be investigated.

10.3 Motion control

The control methods tried (see Chapter 7) were unsuccessful for walking, and were restricted to the use of simple P and PI controllers. It is highly likely that a more complex control method will need to be developed and used. There are many and varied control systems which can be used, and combinations of different types of control systems can be created. One branch of powerful control teqniques involves intelligent or learning control systems, such as Artificial neural networks (ANNs).

Many complex problems are intractable on classical computers running conventional algorithms or too slow to provide real-time solutions. To this end, ANNs have proved useful tools in solving difficult non-linear problems and are often used where real-time responses are required. ANNs also possess the ability to learn to adjust to presented disturbances in an appropriate manner. ANNs learn the correct responses to presented inputs through training.

In bipedal walking control, to achieve a stable gait, rapid or even predictive responses are required to the given feedback. ANNs may therefore provide a desirable alternative to control via computational algorithms.

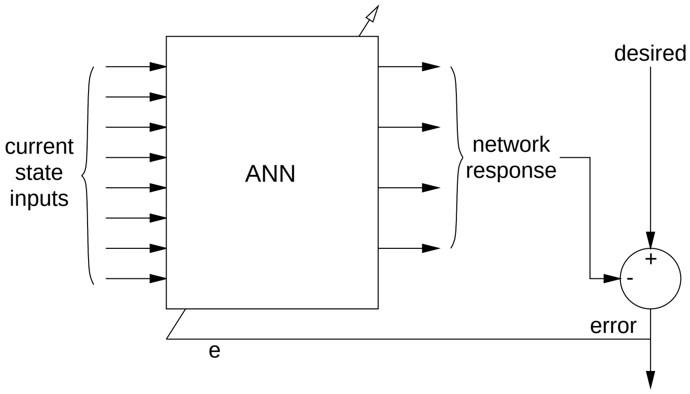


Figure 10.2: Using an artificial neural network (ANN) for control of the dynamic walking gait.

Implementing control through an ANN is shown in Figure 10.2. An error function e must be found in order to train the network. This presents a problem (discussed in Chapter 7) of determining the desired state of the system, which may not be known. Self-organising ANNs are the easiest to train due to the fact that they will learn unsupervised. Therefore an artificial neural network such as Kohonen’s Self-Organising Feature Map might be ideal. Also, the Cerebellar Model Articulation Controller (CMAC) is used extensively in robotics, and should also be investigated.

References

[1] H. Hemami and Y. F. Zheng, "Dynamics and control of motion on the ground and in the air with application to biped robots," *Journal of Robotic Systems*, vol. 1, no. 1, pp. 101–116, 1984.

[2] R. E. Goddard, F. Zheng, Yuan, and H. Hemami, "Control of the heel-off to toe-off motion of a dynamic biped gait," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 1, pp. 92–102, 1992.

[3] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA: MIT Press, 1986.

[4] T. Kato, A. Takanishi, H. Jishikawa, and I. Kato, "The realization of the quasi-dynamic walking by the biped walking machine," in *Fourth Symposium on Theory and Practice of Robots and Manipulators* (A. Morecki, G. Bianchi, and K. Kedzior, eds.), (Warsaw), pp. 341–351, Polish Scientific Publishers, 1983.

[5] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato, "The realization of dynamic walking by the biped walking robot WL-10RD," in *ICAR'85*, pp. 459–466, 1985.

[6] H. Miura and I. Shimoyama, "Dynamic walk of a biped," *International Journal of Robotics Research*, vol. 3, pp. 60–74, 1984.

[7] J. Hodgins, J. Koechling, and M. H. Raibert, "Running experiments with a planar biped," in *Robotics Research: the 3rd Int. Symp.* (O. Faugeras and G. Giralt, eds.), (Cambridge, MA), pp. 349–355, MIT Press, 1986.

[8] S. Kajita, T. Yamaura, and A. Kobayashi, "Dynamic walking control of a biped robot along a potential energy conserving orbit," *IEEE Transactions on Robotics and Automation*, pp. 431–438, Aug. 1992.

[9] A. Takanishi, Y. Egusa, M. Tochizawa, T. Takeya, and I. Kato, "Realization of dynamic biped walking stabilized with trunk motion," in *Proceedings of the Seventh CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators* (A. Morecki, G. Blanchi, and K. Jaworek, eds.), pp. 68–79, 1988.

[10] A. Takanishi, H.-o. Lim, M. Tsuda, and I. Kato, “Realisation of dynamic biped walking stabilised by trunk motion on a sagitally uneven surface,” in *Proceedings of the 1990 IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, pp. 323–330, 1990.

[11] J.-i. Yamaguchi, A. Takanishi, and I. Kato, “Development of a biped walking robot compensating for three-axis moment by trunk motion,” in *Proceedings of the 1993 IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 561–566, July 1993.

[12] T. McGeer, “Passive dynamic walking,” *International Journal of Robotics Research*, vol. 9, pp. 62–82, 1990.

[13] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, “The simplest walking model: Stability, complexity, and scaling,” t*o appear in ASME Journal of Biomechanical Engineering*, 1998.

[14] P. I. Doerschuk, V. D. Nguyen, and A. L. Li, “Neural network control of a three-link leg,” in *Proceedings of the Int. Conference on Tools with Artificial Intelligence*, pp. 278–281, 1995.

[15] P. I. Doerschuk, V. Nguyen, W. Simon, and F. Kwong, “Intelligent ballistic control of a jointed leg,” in *Proceedings of the 1996 IEEE Int. Joint Symp. on Intelligence and Systems*, pp. 117–124, 1996.

[16] W. T. Miller III, “Real-time neural network control of a biped walking robot,” *IEEE Control Systems*, pp. 41–48, Feb. 1994.

[17] A. Kun and W. T. Miller III, “Adaptive dynamic balance of a biped using neural networks,” in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 240–245, Apr 1996.

[18] W. Spong, Mark and M. Vidyasagar, *Robot dynamics and control*. New York: John Wiley & Sons, 1989.

[19] Y. Fujimoto and A. Kawamura, “Simulation of an autonomous biped walking robot including environmental force interaction,” *IEEE Robotics and Automation Magazine*, pp. 33–42, June 1998.

[20] W. Bolton, *Mechatronics—Electronic control systems in mechanical engineering*. Essex, England: Addison Wesley Longman, 1995.

[21] J. Stewart, *Calculus*. Pacific Grove, California: Brooks/Cole, 3 ed., 1995.

[22] M. Vukobratovic, *Legged Locomotion Robots*. Japan: Nikkan Kogyo Shinbunsya, 1975. Interpreted by Kato, I. and Yamashita, C.