Controlling Chaos in a Sagittal Plane Biped Model Using the Ott-Grebogi-Yorke Method

Chun-tsung Feng

Submitted in fulfillment of the academic requirements for the degree of Master of Science in Engineering, Faculty of Engineering, University of KwaZulu-Natal

September 2012

Supervisor: Professor Glen Bright

Declaration

I Chun-tsung Feng declare that

(i.) The research reported in this dissertation/thesis, except where otherwise indicated, is my original work.

(ii.) This dissertation/thesis has not been submitted for any degree or examination at any other university.

(iii.) This dissertation/thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

(iv.) This dissertation/thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a) their words have been re-written but the general information attributed to them has been referenced;

b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

(v.) Where I have reproduced a publication of which I am an author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.

(vi.) This dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis and in the References sections.

Signed: Date:

Supervisor

_____ _____

Professor Glen Bright Date

Acknowledgements

I would to express gratitude to my supervisors and co-supervisor. First, I would like to thank Professor M.H. Ghaffari Saadat for proposing this interesting research topic and his guidance during my first year of research. I am also grateful to Mr. R. Bodger for his assistant with mechanical designs. To Professor L. W. Roberts, I greatly appreciate his support for my research work. I would like to express special thanks to Professor G. Bright for his assistant in helping me finishing my degree. Last but not least I would like to thank the University of KwaZulu-Natal for funding the research.

Abstract

Controlling a system with chaotic nature provides the ability to control and maintain orbits of different periods which extends the functionality of the system to be flexible. A system with diverse dynamical behaviours can be achieved. Trajectory flows of chaotic systems can be periodically stabilised using only small perturbations from the controlled parameter. The method of chaos control is the Ott-Grebogi-Yorke method. In non-chaotic systems large system parameters changes are required for performance changes.

A sagittal plane biped model which is capable of exhibiting periodic and chaotic locomotion was researched and investigated. The locomotion was either periodic or chaotic depending on the design parameters. Nonlinear dynamic tools such as the Bifurcation Diagram, Lyapunov Exponent and Poincaré Map were used to differentiate parameters which generated periodic motion apart from chaotic ones. Numerical analytical tools such as the Closed Return and Linearization of the Poincaré Map were used to detect unstable periodic orbit in chaotic attractors.

Chaos control of the model was achieved in simulations. The system dynamic is of the non-smooth continuous type. Differing from other investigated chaotic systems, the biped model has varying phase space dimensions which can range from 3 to 6 dimensions depending on the phase of walking.

The design of the biped was such that its features were anthropomorphic with respect to locomotion. The model, consisting of only the lower body (hip to feet), was capable of walking passively or actively and was manufactured with optimal anthropometric parameters based on ground clearance (to avoid foot scuffing) and basin of attraction simulations. During experimentation, the biped successfully walked down an inclined ramp with minimal aid. Real time data acquisitions were performed to capture the results, and the experimental data of the walking trajectories were analysed and verified against simulations. It was verified that the constructed biped exhibits the same walking trend as the derived theoretical model.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	Definition
Roman	
A	Jacobian matrix of the linearised mapping
B	Centrifugal matrix
	Shift vector
b_1	Length from knee to m_s (y projection)
b_2	Length from hip to m_t (y projection)
c	y-intercept
C	Jacobian matrix of the linearisation
c_1	Length from knee to m_s (x projection)
c_2	Length from hip to m_t (x projection)
D	Vector of constants
d_0	Initial distance between two nearby points on the Poincaré Section
d_n	Distance between two nearby points on the Poincaré Section after n^{th} iterations
D_n	Distance between two successive recurrence points on the Poincaré Section
D_t	Pointwise dimension about the target point
e_i	i^{th} stable unit eigenvector
f	Poincaré function
f_j	j^{th} contravariant basis vectors
g	Gravitational acceleration : g = 9.81 m.s^{-2}
G	Gravitational matrix
H	Inertia matrix
	Total height of the body
$\vec{H}$	Distance vector from the hip to the n^{th} swing leg mass
	Distance vector from the hip to the n^{th} non-colliding leg mass
H_n	Displacement heights of the n^{th} mass
H_t	Total height of the body
I	Identity matrix
i	Integer
I_h	Hip moment of inertia
I_s	Shank moment of inertia
I_t	Thigh moment of inertia
k	Phase space dimension
L	Lagrangian
l_k	Length from the knee to the foot radius centre
l_{ms}	Length from the knee to m_s
l_{mt}	Length from the hip to m_t
l_r	Length from hip to foot radius centre
l_s	Shank length
l_t	Thigh length
m	Slope gradient
m_h	Hip mass
m_s	Shank mass
m_t	Thigh mass
N	Number of distributed points

$\vec{O}$	Distance vector from the stance foot contact to the n^{th} point mass
	Distance vector from the colliding foot contact to the n^{th} point mass
p	Period of the orbit
p_f	Control parameter for the fixed point
p_n	Control parameter of the n^{th} state condition vector
q	Generalised coordinates vector
Q^-	Pre-collision transition matrix
$\dot{q}$	Velocity time derivative of the generalised coordinate vector
$\ddot{q}$	Acceleration time derivative of the generalised coordinate vector
Q^+	Post-collision transition matrix
q_0	Initial conditions vector
q_1	Angle between normal of the ground and stance leg
q_2	Angle between normal of the ground and swing thigh
q_3	Angle between normal of the ground and swing shank
Q_i	Generalised torques
q_i	Generalised coordinate
r	Positive growth factor
r_h	Length from foot contact to m_h
r_n	Absolute distance between the state and fixed point
r_s	Radius of roll-over foot arc
s	Number of stable eigenvalues
S	Output sensitivity
S_1	Constant value of 0 or 1
S_2	Constant value of 0 or 1
T	Kinetic energy
t	Independent time variable
U	Potential energy
u	Number of unstable eigenvalues
u_1	Stance foot joint torque
u_2	Hip joint torque
u_3	Swing knee joint torque
u_4	Stance knee joint torque
$\vec{V}$	Velocity vector of the n^{th} mass
V_{offset}	Offset voltage
V_{output}	Output voltage
w_f	Foot radius centre shift length
x_0	Distance from the reference origin to the stance foot contact point
x_n	x-coordinate of n^{th} mass
	Projections of the masses on the x-axis
	Ratio of existing population to maximum population of year n
$\dot{x}_n$	Velocity time derivative of x_n
y_n	y-coordinate of n^{th} mass
	Projections of the masses on the y-axis
$\dot{y}_n$	Velocity time derivative of y_n
Z_f	Fixed point vector
Z_n	n^{th} state condition vector

Greek

γ	Slope angle
ε_r	Tolerance value of r_n

Δp_{max}	Maximum allowable perturbation difference
ε_0	Initial distance vector between two nearby points on the Poincaré Section
ε_d	Tolerance value of D_n
ε_k	Angle between the line connecting knee to foot radius centre and shank.
ε_{rf}	Constant fraction
ε_t	Angle between the line connecting m_h to foot radius centre and thigh
	Linear size of the neighbourhood
θ	Inclination angle
θ_{ms}	Angle between the line connecting knee to m_s and shank
θ_{mt}	Angle between the line connecting m_h to m_t and thigh
λ	Lyapunov exponent
λ_i	i^{th} eigenvalue
τ	Generalised torque vector
τ_0	Average transport time from starting point to target

Superscript

-	Pre-collision
+	Post-collision
h	Heel strike
k	Knee strike
m	Number of data pairs
n	Integer number
s	Number of stable eigenvalues
T	Matrix transpose
u	Number of unstable eigenvalues

Subscript

h	Hip
i	Integer number
j	Integer number
k	Phase space dimension
l	Integer number
n	Mass number (1: hip, 2: stance thigh, 3: stance shank, 4: swing thigh; 5: swing shank)
	Integer number
s	Shank
	Number of stable eigenvalues
st	Stance leg
sw	Swing leg
t	Thigh
u	Number of unstable eigenvalues
x	x-axis
y	y-axis

List of Definitions

Anthropometry: the study of human body measurements in terms of size and proportion

Anthropomorphic: attributes of human characteristics/ behaviour to non-humans

Perturbation: agitation, small change in a system

List of Terminologies

Attractor: this is a point, curve, area or space where nearby trajectories converge into

Configuration space: the physical reachable space of a system

Basin of attraction: the subset of phase space where all flow starting within the subset remains within over time

Bifurcation: refers to the splitting of system behaviour into two regions when the control parameter is varied

Bifurcation diagram: a plot of the long-term states of a system as a function of the bifurcation parameter

Bifurcation parameter: the control parameter of a system that dictates the bifurcation process

Chaotic system: a system that is deterministic and sensitive to initial conditions

Deterministic: a deterministic system is a system that involves no randomness, the output of the systems is predictable and always the same for identical input conditions

Ergodicity: trajectories of an ergodic system will eventually visit all parts of its phase space after a sufficiently long time

Fixed Point: a point that returns to its starting position

Four phases: this refers to the unlocked knee swing phase, the knee strike phase, the locked knee swing phase, and the heel strike phase

Ground Clearance: the minimum distance between the base of the swing foot and ground of a walking step

Logistic Map: a discrete equation proposed which approximates the biological population growth rate of species

Lyapunov exponent: this is a measure of the divergence or convergence of initially nearby trajectories

Period-doubling: bifurcation of period one orbit into period two orbit

Phase diagram: a plot showing the trajectories of all the generalised coordinates of a system

Phase space: the space of attainable states of a system

Poincaré Map/ Poincaré Function: a discrete recurrence map/ function that models intersection between orbits of a system and a section

Poincaré Section: a section used to intersect the orbits of a system where the intersection points are studied

Repellor: this is a point, curve, area or space where nearby trajectories diverge from

Step: a walking footstep resulting from a leg swing

Stride: two consecutive walking footsteps resulting from two alternating leg swings

List of Abbreviations and Acronyms

2D	Two Dimensional
3D	Three Dimensional
BOA	Basin of Attraction
COM	Centre of Mass
DC	Direct Current
DOF	Degrees of Freedom
GC	Ground Clearance
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
ODE	Ordinary Differential Equation
OGY	Ott-Grebogi-Yorke
Op-amp	Operational Amplifier
PWM	Pulse-Width Modulation
UPO	Unstable Periodic Orbit
USB	Universal Serial Bus

CHAPTER 1 INTRODUCTION

1.0 Introduction

Investigations of the dynamics and stability of walking robots have been conducted by many researchers for two reasons. First is its ability to adapt to terrains of various conditions. Legged robots are capable of traversing irregular terrains and still maintain stability through foot placement selection. This ability of choosing the step location not only allows a legged robot to select the path of travel, it also allows the robot to regain its stability from disturbances through foot placement response. The above capabilities of walking robots are some of the reasons why it is favoured over wheeled robots when it comes to traversing rough landscapes. Second is that walking robots, especially bipedal machines, have human-like locomotion. It takes little energy for humans to walk due to efficient locomotion trajectories and unique segment proportionality. In order to emulate the gait characteristics of humans, certain humanoid robots are designed with anthropometric parameters.

One of the most significant problems that walking robots have is high walking energy cost. Most robots have to carry large battery packs just to perform walking. In order to improve the energy efficiency of walking robots, a class of walking machines which works on passive dynamics has been researched. Tad McGeer built the first totally unactuated passive walking robot [1] which is capable of continuous walking after a start push. It only requires energy gain from going down a gentle slope to maintain walking.

1.1 System Exhibiting Chaos

It has also been shown that passive walking models are capable of exhibiting chaotic locomotion [2], [3]. Chaotic locomotion is an asymmetric gait with no repeating trajectory patterns due to its underlying chaotic system. A chaotic system is a nonlinear deterministic system that is very sensitive to changes. Any slight variation in initial conditions will produce results that divert over time (also known as the butterfly effect). The divergence behaviour is not due to the effect of random influences, as the output states of chaotic systems are fully predictable. Typically, orbits of a chaotic system wanders about its phase space without ever crossing itself thus generating a non-recurring (chaotic) cycle.

Ott, Grebogi and Yorke have developed a chaos controlling method called the Ott-Grebogi-Yorke method (commonly known as the OGY Method) [4] which direct wandering orbits of chaotic system into periodic cycles through small parameter perturbations. The OGY method is energy efficient as it requires only small actuation control forces. It is possible to guide chaotic orbits onto different periodic cycles of choice [5]. Each selected periodic cycle constitutes specific performance behaviour. The functionality of chaotic systems is flexible due to the possibility of orbit selection through chaos control. This would not be possible in non-chaotic systems where performance changes typically require large system parameter alterations.

1.2 Motivations for Research

The research topic of this dissertation is based on the ideas of efficient passive dynamic walking and diverse performance possibilities through chaos control. The research findings of experts inspired the author to conduct an enquiry which combines the benefits of both ideas.

1.3 Research Question

This research investigated nonlinear dynamics of a biped model utilising chaos and aimed to determine whether a passive dynamic based biped model exhibiting chaos is capable of efficiency and flexibility.

So far there are no real applications with bipeds utilising chaos in this field of research which is still developing. A sagittal plane biped model which consists only of a lower body was researched. The upper body was excluded for simplification purposes as chaos control becomes very complicated for models with many degrees of freedom.

1.4 Research Objectives

The research objectives were:

- To research and investigate literature from experts related to the topic of this dissertation.
- To research and develop a sagittal plane biped dynamic model to be used for simulations and the generation of stable biped parameter ranges.
- To apply the OGY method of Chaos Control to the kneed developed biped model and to simulate these processes. To the author's knowledge, no cases of chaos control have been applied to kneed walking models.
- To design, manufacture, and assemble a sagittal plane biped for the purpose of experimentation and testing.
- To test and obtain the performance results of the biped and to verify its functionality.

1.5 Methodology Approach

The research was carried out using approaches in the order listed below.

- Gathering and comprehending the principles of passive walking robots and chaos theory which involves research and investigation of existing biped models and chaos control cases.
- Selecting a kinematic structure which represents the biped model. The model structure needed to be capable of exhibiting passive and chaotic locomotion. A large number of joint degrees of freedom had to be avoided to prevent unnecessary modelling complications.
- Formulating and deriving modelling equations for the generation of walking simulations. The algorithms needed to include and model typical events of locomotion.
- Applying the OGY method to the developed biped model to direct chaotic orbits onto the targeted periodic motion. In order to analyse the dynamics of chaos, numerical and nonlinear tools were used.
- Designing the biped which required the integration of disciplines from various fields namely mechanical, electrical, electronic, and control systems. The mechanical part of the biped involved the selection of material and machine mechanisms, and the design of structural components. The

electrical and electronic parts constitute actuators, sensors, control boards, and power supplies. The design of a control system to execute the walking routine was required. A data acquisition system capable of capturing and storing the experimental results was used.

- Testing the developed biped with regard to its hardware and software functionality and testing the overall performance of the biped.

1.6 Dissertation Overview

The outlines of each chapter are summarised below:

Chapter 1 - Introduction

Brief introductions of the background, motivations, research question, and objectives for the topic of this research are discussed.

Chapter 2 - Literature Review

This chapter presents literature review of researches conducted by experts related to the topic of the dissertation. Aspects performed by the author and not covered by other researchers, are stated.

Chapter 3 - Modelling Dynamics

Detailed derivation procedures and modelling equations of the sagittal biped model are given in this chapter as well as the parameter optimisation simulations which were conducted for the determination of stable biped dimension ranges.

Chapter 4 - Chaos Control

The general understanding of chaos is outlined here. The derivation and application procedures of the chaos control methodology are presented and successful chaos control examples are demonstrated.

Chapter 5 - Biped Design

The design specifications of the biped are listed and explained. Categories of the design process are outlined namely, electrical and electronic hardware selection, mechanical structure design, controller hardware selection, and software programming.

Chapter 6 - Testing and Results

This chapter presents the testing results of the manufactured biped. The performances and functionality of the biped are evaluated based on the comparison of experimental data and simulation results.

Chapter 7 - Conclusion and Future Research

The concluding chapter remarks on important findings, and contributions regarding the topic of the research are summarised. Outcomes of the research are evaluated against its specified objectives. Recommendations for future research conclude the chapter.

1.7 Chapter Summary

This chapter introduced the concept of this dissertation's topic. The motivation for the research of the topic was explained. The research question which was answered by the results and findings of this research, was stated. The objectives of the research were outlined. The methodology approaches used to

carry out the research were described. An overview of the chapters which highlight the significance of the dissertation, were presented.

CHAPTER 2 LITERATURE REVIEW

2.0 Introduction

Walking robots is a popular research topic and researchers have constructed robots similar in form to humans. It has been shown that chaotic gait is the central feature of the human locomotion [6]. Incorporating chaotic characteristics as part of the dynamic system of bipeds have not been achieved by many engineers. As for the control of chaos in bipeds, even less research had been conducted on it.

This chapter reviews research performed by experts related to the topic of this dissertation. The presence of chaos in different fields is discussed. An example of how chaos control is used to achieve efficient steering of a spacecraft is presented. Research conducted on passive walking models and chaos was also investigated. These results form the basis of the dissertation. Passive walkers are studied due to the underlying dynamics of chaos. Chaos control can only be implemented if a robot was controllable. This means that actuation of the passive walking model is required. Research that covered both biped and chaos models were used as the starting point. The advantages of incorporating anthropometric proportionality into model designs are stated. The sequence of dynamic modelling and the requirements of biped design are based on the general concepts of walking bipeds. Aspects not covered by other researchers are discussed in this chapter's conclusion.

2.1 Chaos Related Researches

Chaos is found in different fields of studies. One example is the presence of chaos in nature such as the weather. Chaos is also found in simple nonlinear equations such as the Logistic Map [7]. Other well-known examples of chaos are Chua's Circuit in electrical circuits [8] and the Lorenz model: a model of convecting fluids [9].

Since the publication of *Controlling Chaos* by Ott, Grebogi and Yorke [4], on the method of chaos control, many researchers have utilised chaos controlling theory to stabilise chaotic behaviours [5], [10]. Ott *et al*. have shown that it is possible to stabilise trajectories within a chaotic system through small parameter perturbations.

All chaotic systems consist of orbits of different periods. A system with diverse dynamic behaviours can then be achieved through control of the orbits. High order chaos control has been achieved of up to 90 orders [5]. The functionality of the system is extended to be flexible and dynamic control of chaotic systems only requires small perturbations. Unlike non-chaotic systems, large system parameter changes are required for performance changes.

An example that illustrates the benefits of having a chaotic system is the control of a spacecraft and the manner in which it is sent across the solar system using a limited amount of fuel [11]. The efficient spacecraft steering was possible due to the presence of chaos in a planar three-body problem which allows large dynamic behaviour changes to be accomplished by small perturbations [12]. Efficient control

cannot be accomplished in a non-chaotic system as small perturbations only produce minor dynamic changes.

Chaos is generally present in multidimensional systems that are underactuated. Fully active systems have their dynamics set according to the algorithm used to govern trajectories. Finding a chaos system with all its degrees of freedom (DOF) actuated is unlikely. Chaos can also be induced in systems through external disturbances. This situation is undesirable in this research as the required disturbance may not be achievable. Due to the reasons above, passive walking robots are studied as chaos can often be found in such systems.

2.2 Walking Bipeds and Chaos

Walking robots were researched with focus on the following categories: passive walking robots, actuated robots based on passive walking models and biped walking models exhibiting chaos.

2.2.1 Passive Walking Robots

Passive walking bipeds were first investigated by McGeer in 1990 [1]. Passive bipeds are constructed such that when given an initial push, they will walk down a gentle slope. When walking down a slope, potential energy is converted into kinetic energy. Bipeds lose some of their energy due to collision in knee and heel strikes. Continuous walking is possible if the overall velocity of the biped remains constant. Based on previous research, McGeer has extended his passive walking model to one which has knees [13]. The research shows that even a model with passive hip and knees is able to walk down a gentle slope. Ikemata *et al.* have also constructed a passive walker with knees which walked 4010 steps on a tilted treadmill [14]. Figure 2-1 (a) shows the photo of the walker used in their experiments. The design of this walker was similar to McGeer's kneed walking model. The design employed inner and outer paired legs which constrained the walkers to the sagittal plane.

Collins *et al.* have built the first three dimensional passive biped which consists of passive hips and knees, and arms as shown in figure 2-1 (b) [15]. This robot is capable of walking down a ramp with the right initial push. It was designed to manoeuvre in all three planes of motion and is therefore a three-dimensional model [15].

Another three-dimensional passive walker was developed by Tedrake *et al.* [16]. This robot was designed with two straight legs joined by a passive hip as shown in figure 2-1 (c). Its feet have curvatures both in the sagittal and frontal planes. Due to the feet curvatures, the robot oscillates laterally during its forward pitch when walking down a slope. These lateral oscillations generate ground clearance for the swinging foot while walking. The step of the robot is confined to short lengths since large steps will lead to large yaw rotations which will cause it to fall due to being imbalance.

Figure 2-1: a) The Nagoya Biped [14], b) The Cornell Biped [17], c) The MIT Biped [16]

2.2.2 Actuated Robots based on Passive Walking Models

The passive walking model has been extended to walk on flat ground through the addition of actuators [18], [19]. Actuation is used as a substitute for the potential energy gained while walking down a slope and also to control and enhance performance. The structure of the actuated robot is still based on passive walking models for energy efficiency.

Wisse and van Frankenhuyzen have designed a robot with semi-actuated hips and knees which aims to minimise energy usage by incorporating the passive walking model in its design. Figure 2-2 (a) [18] shows a photo of the design. The results show that it has the same energy efficiency as a human body when walking on flat ground [18].

Wisse *et al.* developed a robot with full human features. It has a head, an upper body with arms and a lower body with kneed legs. Figure 2-2 (b) [19] shows a photo of the robot which walks with hip actuation and synchronised arm movements. The knee and ankle bend and rotate respectively during walking to avoid foot scuffing. The robot walks on flat ground with a gait that is almost periodic [19].

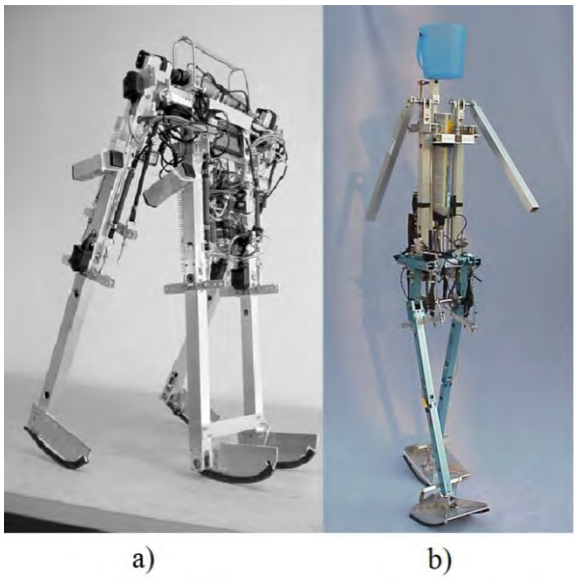

Figure 2-2: a) The Delft Biped Mike, b) The Delft Biped Denise

2.2.3 Biped Walking Models exhibiting Chaos

The simplest walking model or the compass model was probably the first used to analyse chaotic locomotion [2], [20]. The model consists of two links with hip and feet point masses (as shown in figure 2-3) which illustrates the model walking down a gentle slope. It begins with a single support phase just after the heel-strike followed by a swing phase, and ends off with a heel strike in the double support phase. All motions occur in the sagittal plane as it is assumed that no frontal motions take place. The model presented here has 2 DOF and illustrates the walking motions which involve leg swings and heel strikes.

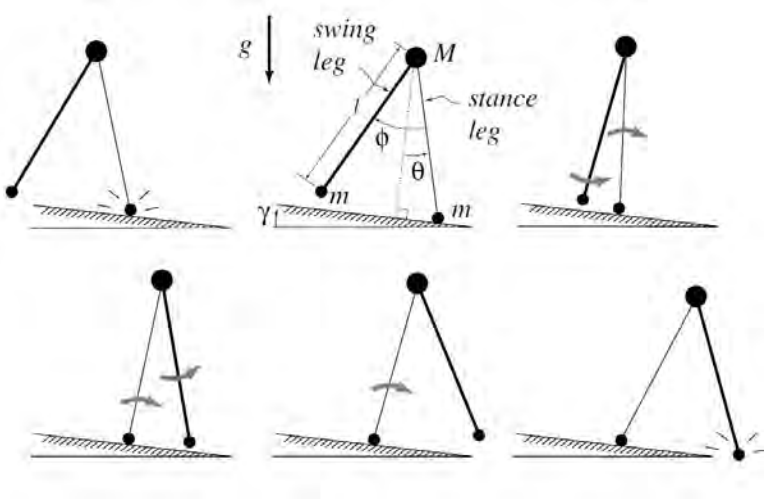

Figure 2-3: Passive walking steps of the simplest walking model [2]

Garcia *et al.* has demonstrated the effects of slope angle variation on the walking cycle of the simplest walking model [2]. The research showed that as the slope angle is increased, period doubling[1] occurs. This means it takes two or more consecutive steps of different periods before the model returns to its starting configuration. Period doubling increases until chaotic gait occurs. In chaotic gait, no two walking steps reflect the same conditions. The simplest walking model is said to have deterministic nonlinear dynamics or a chaotic system.

The stability and progression of the gait behaviours of the simplest walking model was investigated by Goswami *et al.* [21]. This is a more extensive research than Garcia *et al.*'s research [2] where nonlinear dynamic assessing tools Phase diagram[1] and Poincaré Map[1] are used by Goswami *et al.* It focuses on how a change of system parameters affects the passive gait of the biped. The gait of the biped model varied from symmetrical to asymmetrical and eventually became chaotic as the system parameters were varied.

Schwab and Wisse have investigated the stability of the periodic cycles of the simplest walking model and quantified the stability of the periodic cycles by finding the eigenvalues of Poincaré Function's Jacobian[1] and its basin of attraction[1] size [20]. It was proven that the basin of attraction quantifier is more

[1] Refer to Chapter 5

suitable for measuring stability. Wisse *et al.* have gone a step further to determine the bifurcation diagram[1] of the simplest walking model with an upper body [22].

Kurz *et al.* have experimentally illustrated that a robot can exhibit chaos behaviour due to its underlying mechanical system. The robot used in experimentation was based on the simplest walking model. It was confirmed that chaos behaviour is possible in a passive walking robot by calculating the largest Lyapunov exponent[1] and using a surrogate analysis method [3].

Kurz and Stergiou presented control of the simplest walking model using hip actuation. Their simulations illustrate that through hip actuation, periodic gait is achievable in a chaotic dynamic system [23].

Garcia *et al.* have investigated the chaotic locomotion of a kneed biped and illustrated simulations of slope angle versus walking periods through a bifurcation diagram. The diagram indicates whether the gait is periodic or chaotic as a function of slope angle. Unlike other research this case study used a kneed model [24].

2.3 Stability Measures

A robot is defined stable if it does not fall [25]. Stability is measured over a sequence of walking steps [26]. It is not a measure of local stability which is stability at an instantaneous moment of time. Common stability quantifying tools such as the Zero Moment Point [27] and Centre of Pressure are not used. These tools are used to measure the robustness of legged systems for specific conditions.

In this dissertation the locomotion studied is either periodic or chaotic. Locomotion may only be categorised into the above two categories after a sequence of successful walking steps. In periodic cycles a few converging steps are normally taken before trajectories come close to its final periodic state. Theoretically it takes an infinite number of steps to reach a periodic state. Chaotic cycles usually require more than 30 steps before they can be categorised as chaotic. Neither periodic nor chaotic gait can cause a model to fall over without external disturbances. If it does, then the gait cannot be categorised according to the above definitions. This eliminates the study of any unstable gait which leads to the study of stable gaits only. All stability measures presented in this dissertation will then be a measure of long term stability robustness.

2.4 Anthropometric Proportionality

Bipeds designed with anthropometric mass distribution are shown to have improved walking abilities [28]. Figure 2-4 [29] shows the proportionality of body segment lengths and masses of humans which are incorporated into the biped design. Table 2-1 [30] indicates the percentage mass of the body segments. A model designed with anthropometric parameters will also lead to the locomotion being more anthropomorphic.

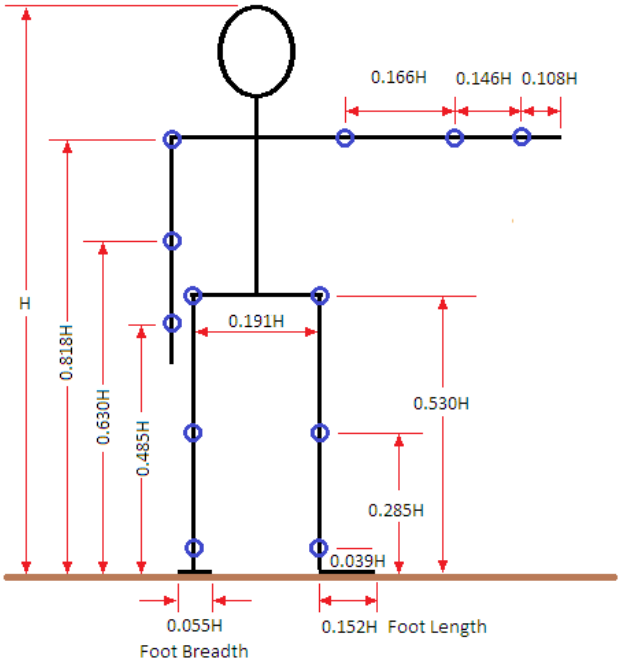

Figure 2-4: Proportionality of body segments lengths

The symbol *H* represents the total height of the body.

Table 2-1: Percentage weight of body segments

Body Segment	Percentage of Total Body Weight [%]
Lower Body	46.4
Pelvis	14.2
Mass per Leg	16.1
Thigh	10
Shank	4.65
Foot	1.45

2.4.1 Roll-over Feet

Human feet are shaped such that the centre of pressure shifts forward along the foot during a forward step. Hansen *et al.* have researched and developed feet of different roll-over shapes which imitate the functions of the human foot [31]. As the feet cannot be designed to bend like a human's, curved feet are used to replicate this shift in centre of pressure behaviour. Roll-over feet in biped designs are more robust to disturbances and therefore provide greater stability [24]. Wisse and van Frankenhuyzen have also shown that increasing the foot radius improves a biped's robustness against disturbances. This improvement is however dependant on the length of the foot [18]. A long foot with a large radius may be very robust but has a higher chance of experiencing foot scuffing [24]. In every leg length, there is a finite range of foot length for which robustness and foot clearance is optimal [13].

2.5 Concepts of Walking Bipeds

2.5.1 Classification of Biped Walkers

Three classifications of walkers are defined by Sardain *et al.* [32]. The first is a static walker with a centre of mass (COM) constantly inside its support polygon circumscribed by the ground contacting supports. Normally the support polygon is the contact area of the feet stance with ground. The second is a dynamic walker which has its centre of pressure within the support polygon. For the above two classifications, stability is measured at instantaneous moments. The third is a purely dynamic walker where stability is measured over a period of time. This research focuses on the latter type.

2.5.2 Walking Cycle

A complete walking cycle (a stride) consists of two walking steps. A step is executed when each leg takes a turn in being the stance or swing leg. Each step can be split into four phases. A step begins at the instance of double support where both feet instantaneously touch the ground. Thereafter the biped goes into a swing phase where each of the legs becomes either a stance or a swing leg. During the swing phase the stance knee remains locked and the swing knee bends. Both legs continue to rotate forward until the swing leg straightens and locks in the knee strike phase. After knee strike, the model behaves like an inverted double pendulum and goes into the locked knee swing phase. The step ends with the swing leg colliding with the ground which is the heel strike phase. In summary the four phases in a step are: the unlocked knee swing phase, the knee strike phase, the locked knee swing phase, and the heel strike phase. Figure 2-5 illustrates the four phases in a single step.

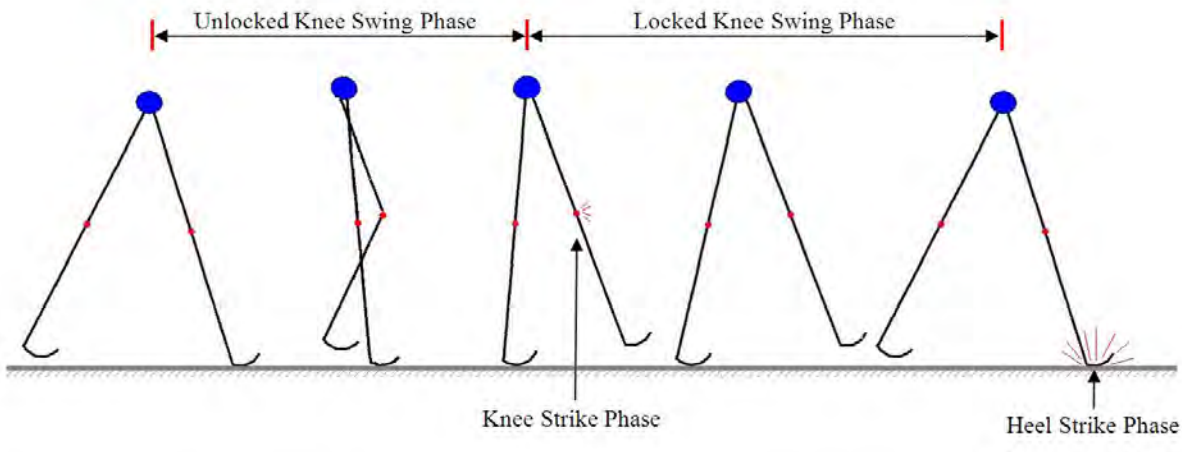

Figure 2-5: Four phases of a walking step

2.5.3 Types of Walking Failure Modes and its Counter Measures

Possible types of walking failure modes are investigated and its counter measures are presented here. This is to prevent or minimise occurrence of failure. Some common failure modes are discussed below.

2.5.3.1 Heel and Knee Strike

Impacts occur during feet collisions with the ground and during the locking of the knee. These impacts cause a disturbance to the biped system often causing it to become unstable. The study by Sardain, *et al.* showed that limbs with anthropometric distribution of masses minimise heel-strike impact [32]. In order to minimise impacts from knee and heel strike, shock absorbing materials are used in the regions of collision. Materials such as hard rubber, which behaves in a partly inelastic manner, is used as it absorbs impact energy. This corresponds with the assumption of collision being totally inelastic in modelling.

2.5.3.2 Knee Collapse

There are two types of knee collapse. First is the hyperextension of the knee. One of the easiest ways to prevent hyperextension is to install mechanical knee caps with rubber padding [14] which is collision absorbent. The second type of knee collapse is bending of the knee when it is supposed to be locked or straight. Knee collapse may occur to the stance knee during walking or the swing knee after knee strike (locked knee phase). Stance knee collapses can be prevented by the addition of roll-over feet. Due to the geometry after the installation of curved feet, there will always be a torque that opposes knee bend hence keeping the knee locked [33].

Swing knee collapse is due to the elastic behaviour of the mechanical knee cap where the shank tends to bounce back. For the prevention of swing knee collapses, researchers have installed knee latches such as magnetic latches [34], suction cups [34] or mechanical locks. The first two types of latches will still collapse under large impacts while the last tends to stay locked in the leg swing phase. The knee latches listed above require active control and only unlock from strike feedbacks. Swing knee collapse can be minimised if the collision energy is absorbed in the knee padding thus reducing bounce back of the shank.

2.5.3.3 Falling Forward and Falling Backward

Two types of falling occur in the sagittal planes, namely forward and backward falling. Falling forward is due to a swing leg's incapacity to place its foot forward fast enough before the heel strike. Falling backward is the result of a stance leg not pitching forward enough. Falling backward normally happens as a result of the hip being behind the stance support. To prevent this Wisse *et al.* offered the following:

"*You will never fall forward if you put your swing leg fast enough in front of your stance leg. In order to prevent falling backward the next step, the swing leg shouldn't be too far in front*." [35]

As suggested by Wisse *et al.*, a backward fall cannot be corrected in the same step [35]. It can be prevented but not overcome. This type of walking failure cannot be prevented through mechanical design. It is prevented through motion control such as control of the step length.

2.5.3.4 Frontal and Yaw Unbalance

Lateral fall and yaw rotation occur in all bipeds. Lateral imbalance occurs when one leg is lifted off the ground. This is when the biped's centre of pressure lies outside its support polygon. Lateral falls also result from not placing the swing foot down fast enough. During biped leg swings, momentum is generated about the yaw axis. This causes the biped to rotate about its foot contact and fall over.

To overcome the above problems, researchers have designed sagittal plane bipeds with paired legs [1], [14]. The designs have coupled legs consisting of outer and inner pairs; four legs joined to a common hip can be seen from the front view. The legs overlap each other into two legs when seen from the side or sagittal plane view.

2.6 Chapter Summary

Passive walking models are efficient models as they only require the potential energy of gentle slopes for input during walking [24]. Passive walkers are also shown to be capable of exhibiting chaotic locomotion [24] due to their underlying chaotic systems. It has been determined that there are orbits of different periods in chaotic systems [5]. Guiding chaotic orbits towards periodic orbits is achievable using the OGY Method [4]. Each controllable orbit within the chaotic system corresponds to a specific system performance. If a controllable robot is designed based on a passive walking model, it will be efficient and flexible in terms of performance.

The features of this dissertation's biped model are based on several considerations. Modelling of the biped was performed only in the sagittal plane. Chaos analysis of a three dimensional model requires more degrees of freedom than a plane model. Dynamic modelling becomes more complex as the number of DOF is increased. An overly complex model was avoided because of this complexity. By analysing the model in the sagittal plane, the paired leg design could be used. This solves the two major causes of biped locomotion failure, namely lateral and yaw imbalance.

It was decided that a lower body model with actuated hip, passive kneed legs and roll-over feet would be used. Hip actuation was selected as it is the one joint that can drive the whole model. It also serves as the control parameter for chaos control. The ankles are rigid for simplicity of modelling. Roll-over feet are used rather than pointed feet (as in the simplest walking model) because of the reasons discussed in the anthropometric parameter section.

There are similarities and differences between the author's research and that of others. The common similarities are that the biped design is based on a passive walking model, research into walking models was conducted in the sagittal plane, and the link and joint features are similar to some researchers' models [13], [14], [18]. Actuation of the hip as the method to control the biped is the same as Wisse and van Frankenhuyzen [18]. One major difference is that chaos control has been applied to a kneed walking model; to date it has only been applied to the simplest walking model robots. To the author's knowledge, no cases of chaos control have been applied to kneed walking models. Another difference is that some anthropometric proportionality was incorporated in the design which other researched models lack. This is due to the advantages described by Sardain *et al.* [28] and Hansen *et al* [31].

CHAPTER 3 MODELLING DYNAMICS

3.0 Introduction

This chapter presents the modelling and simulations of the sagittal biped's dynamics and stability. Lagrange formalism was used in the derivation of the modelling equations of the four walking phases. The simulation procedures of the modelling equations are outlined. The conditions and assumptions used in the simulation are declared. The simulation results of a typical walking step were plotted and analysed. From the plots of the phase curves, typical patterns of trajectory flow were identified and explained. Ground clearance and basin of attraction size simulations (terminology explained in later sections) were used as stability quantifiers. The relationship between the two stability quantifiers and parameters were determined. Based on the trends of stability relations and anthropomorphic proportionality, a design parameter range was proposed.

3.1 Sagittal Plane Model

The model parameters of the sagittal biped were first defined followed was the mathematical modelling of its walking dynamics.

3.1.1 Model Parameters

Model parameters of the biped are as defined in figure 3-1, figure 3-2 and figure 3-3. The sagittal plane model is of a triple linked inverted pendulum type. It has 3 and 6 DOF in configuration and phase space respectively. It was modelled on an inclined slope with an inertial reference frame as shown in figure 3-1.

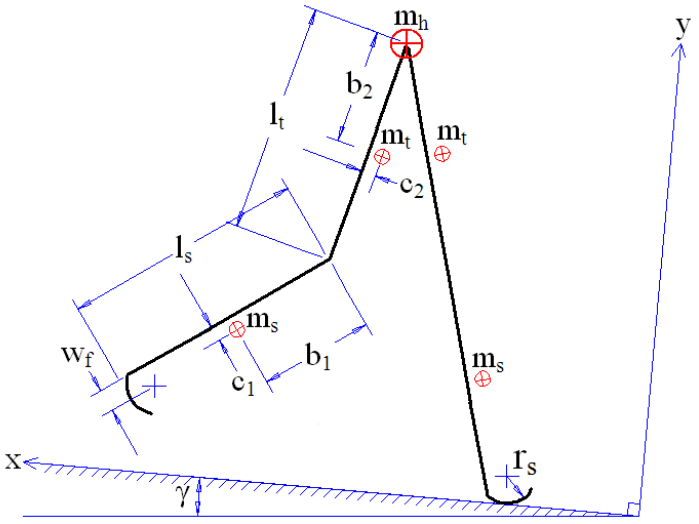

Figure 3-1: Sagittal plane model and its inertial reference frame

The independent parameters of the model are shown in figure 3-1. The lower body of the biped was modelled using five segments. The segments consist of the hip, two thighs and two shanks. The segments were modelled as point masses (red crossed circles) and moments of inertias about COM (not shown in the figure). The mass symbols are m_h, m_t, and m_s. The moment of inertia symbols are I_h, I_t and I_s. The

subscripts *h*, *t* and *s* denotes hip, thigh and shank respectively. The position of the thigh and shank COMs are constrained by parameters b_2, c_2, and b_1, c_1 respectively. The leg segments lengths l_t and l_s are the thigh and shank lengths respectively. The radius of the roll-over foot is r_s and its centre is forward shifted with an offset of w_f. The angle of the slope is γ.

Figure 3-2 illustrates the angles used to describe the configuration of the model. The angles q_1, q_2 and q_3 and its respective velocities make up the dynamic variables (generalised coordinates) of the system. All the angles are measured from the normal of the ground to the respective limb with clockwise being positive. The symbols u_1, u_2, u_3 and u_4 are joint torques acting on the stance foot, hip, swing knee and stance knee of the biped respectively. Clockwise torques have positive signs.

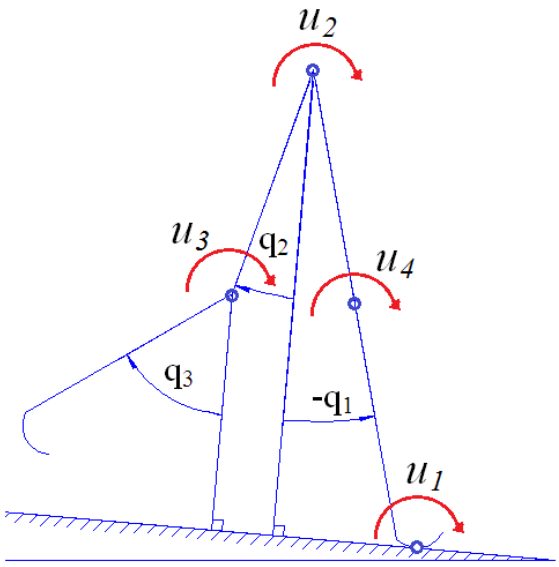

Figure 3-2: Generalised coordinates and external torques of the kneed model

Figure 3-3 shows the dependant parameters which were derived from the parameters of figure 3-1. The symbols l_r, l_{mt} and l_{ms} represent lengths from the hip to foot radius centre, and to m_t, and from knee to m_s respectively. The symbol l_k is the length from the knee to the foot radius centre (not shown in the diagram). The symbol θ_{mt} is the angle between the line connecting m_h to m_t and thigh, and θ_{ms} is the angle between the line connecting the knee to m_s and the shank. The symbol ε_t is the angle between the line connecting m_h to the foot arc centre and thigh, and ε_k is the angle between the line connecting the knee to the foot arc centre and shank.

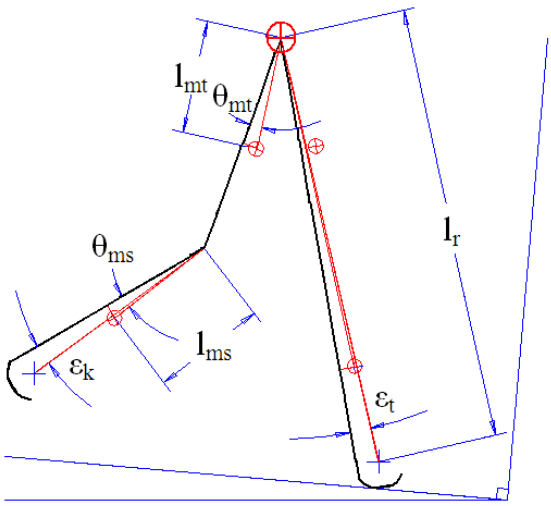

Figure 3-3: Derived biped parameters

3.1.2 Modelling of Walking Phases

3.1.2.1 Unlocked Knee Dynamics

The unlocked knee swing phase is defined as the beginning of the walking step. This swing phase corresponds to the dynamics of a three linked inverted pendulum. The modelling equations of the unlocked knee dynamics was derived using Lagrange formalism which is represented by equation 3.1:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{3.1}$$

Lagrange's equation above requires the computation of the Lagrangian L and the generalised torques Q_i in terms of the generalised coordinates q_i. The subscript i denotes the n^{th} generalised coordinate. For passive walking, the generalised torques are zero. The Lagrangian equation 3.2 is the difference between the kinetic (T) and potential energy (U) of the model.

$$L = T - U \tag{3.2}$$

The kinetic and potential energy of the model were calculated using equation 3.3 and equation 3.4.

$$\begin{aligned} T = {} & \frac{1}{2}m_h\vec{V}_1^{\,2} + \frac{1}{2}m_t(\vec{V}_2^{\,2} + \vec{V}_4^{\,2}) + \frac{1}{2}m_s(\vec{V}_3^{\,2} + \vec{V}_5^{\,2}) \\ & + \frac{1}{2}I_h(S_1\dot{q}_1^2 + S_2\dot{q}_2^2) + \frac{1}{2}(I_t + I_s)\dot{q}_1^2 + \frac{1}{2}I_t\dot{q}_2^2 + \frac{1}{2}I_s\dot{q}_3^2 \end{aligned} \tag{3.3}$$

$$U = m_h g H_1 + m_t g(H_2 + H_4) + m_s g(H_3 + H_5) \tag{3.4}$$

where the symbols$\overrightarrow{V_1}$, $\overrightarrow{V_2}$, $\overrightarrow{V_3}$, $\overrightarrow{V_4}$ and $\overrightarrow{V_5}$ are the mass velocity vectors.

Symbols H_1, H_2, H_3, H_4 and H_5 are the height displacements of the masses. The symbol g is the gravitational acceleration taken to be 9.81 m/s^2. The symbols S_1 and S_2 are weighing factors which take on values of 0 or 1. Due to design constraints, the hip will be mechanically coupled to one of the legs of the biped. Since the legs take turns being either the stance or the swing leg, the hip then takes on the

corresponding angular velocities ($\dot{q}_1$ or $\dot{q}_2$). The rotational kinetic energy of the hip then becomes $\frac{1}{2} I_h \dot{q}_1^2$ during stance phase (S_1=1, S_2=0) and $\frac{1}{2} I_h \dot{q}_2^2$ during swing phase (S_1=0, S_2=1).

The velocity vectors and mass displacement heights are shown in equation 3.5 and equation 3.6 respectively.

$$\overrightarrow{V_n} = \dot{x}_n^2 + \dot{y}_n^2 \tag{3.5}$$

$$H_n = x_n \sin\gamma + y_n \cos\gamma \tag{3.6}$$

where the symbol *n* represents the number of the n^{th} mass.

The number 1 was assigned to the hip, 2 for stance thigh, 3 for stance shank, 4 for swing thigh and 5 for swing shank. The symbols $\dot{x}_n$ and $\dot{y}_n$ are the velocity time derivatives of x_n and y_n. The position coordinates x_n and y_n are projections of the masses on the x- and y-axis respectively. Equations for the coordinates of mass positions are as follows:

Hip mass coordinate:

$$x_1 = x_0 + r_s(-q_1 + \varepsilon_t) + l_r \sin(-q_1 + \varepsilon_t) \tag{3.7}$$

$$y_1 = l_r \cos(-q_1 + \varepsilon_t) + r_s \tag{3.8}$$

where the symbol x_0 denotes the distance from the reference origin to the stance foot contact point.

Stance thigh mass coordinate:

$$x_2 = x_1 - l_{mt} \sin(-q_1 + \theta_{mt}) \tag{3.9}$$

$$y_2 = y_1 - l_{mt} \cos(-q_1 + \theta_{mt}) \tag{3.10}$$

Stance shank mass coordinate:

$$x_3 = x_1 - l_t \sin(-q_1) - l_{ms} \sin(-q_1 + \theta_{ms}) \tag{3.11}$$

$$y_3 = y_1 - l_t \cos(-q_1) - l_{ms} \cos(-q_1 + \theta_{ms}) \tag{3.12}$$

Swing thigh mass coordinate:

$$x_4 = x_1 + l_{mt} \sin(q_2 - \theta_{mt}) \tag{3.13}$$

$$y_4 = y_1 - l_{mt} \cos(q_2 - \theta_{mt}) \tag{3.14}$$

Swing shank mass coordinate:

$$x_5 = x_1 + l_t \sin(q_2) + l_{ms} \sin(q_3 - \theta_{ms}) \tag{3.15}$$

$$y_5 = y_1 - l_t \cos(q_2) - l_{ms} \cos(q_3 - \theta_{ms}) \tag{3.16}$$

Equation 3.1 is a nonlinear second order Ordinary Differential Equation (ODE) and was determined as follows: firstly by the substitution of equation 3.7 to equation 3.16, into equation 3.5 and equation 3.6.

The velocity vectors and mass displacement heights were then determined. The substitution of equation 3.5 and equation 3.6 into equation 3.3 and equation 3.4 were performed to obtain the kinetic and potential energy respectively. From equation 3.3 and equation 3.4, the Lagrangian was obtained using equation 3.2. By substituting the Lagrangian into equation 3.1 and taking the derivatives of L with respect to q_1, q_2 and q_3 and its velocities, three equations are obtained. The three equations were then rearranged into the following standard form of Lagrange's equation:

$$H(q)\ddot{q} + B(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.17}$$

where H is inertia matrix, B the centrifugal matrix and G the gravitational matrix.

The symbols q, $\dot{q}$ and $\ddot{q}$ are angular, velocity and acceleration vectors respectively. The symbol τ is the generalised torque vector of the system. For passive walking, τ is zero. For unlocked knee dynamics, the details of matrices H, B, G, τ, q, $\dot{q}$ and $\ddot{q}$ are found in section A-1 of Appendix A.

3.1.2.2 Knee Strike Transition

Knee strike occurs at the instance where the knee of the swing leg straightens. Figure 3-4 [36] illustrates the knee strike process.

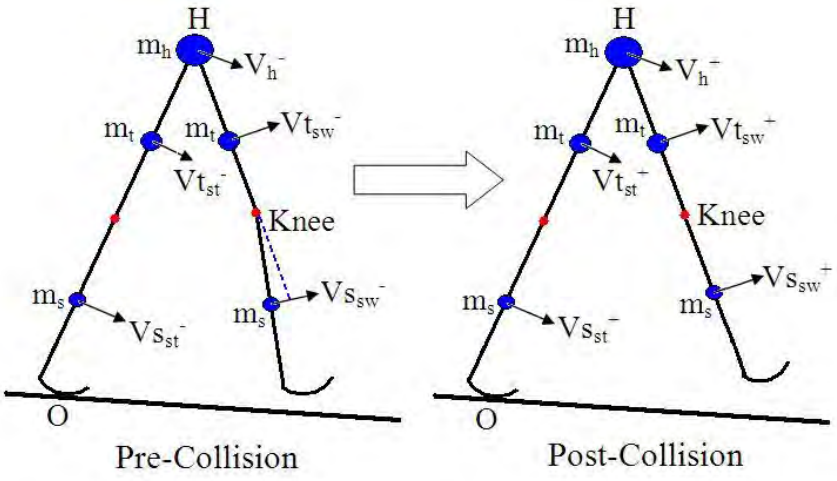

Figure 3-4: Pre- and post-collision of knee strike

Angles of the model remain the same after the knee strike transition as shown in equation 3.18. The transition of the angular velocities of the model was calculated using equation 3.19.

$$\begin{bmatrix} q_1^+ \\ q_2^+ \\ q_3^+ \end{bmatrix} = \begin{bmatrix} q_1^- \\ q_2^- \\ q_3^- \end{bmatrix} \tag{3.18}$$

$$Q^+(q_1, q_2, \dot{q}_1, \dot{q}_2)\begin{bmatrix} \dot{q}_1^+ \\ \dot{q}_2^+ \end{bmatrix} = Q^-(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)\begin{bmatrix} \dot{q}_1^- \\ \dot{q}_2^- \\ \dot{q}_3^- \end{bmatrix} \tag{3.19}$$

The symbols '+' and '-' denote post-collision and pre-collision respectively. The symbols Q^+ and Q^- are derived transition matrices based on the assumption of conservation of angular momentum during knee

strike. The conservation of momentum principle for knee and heel strikes has been applied by Chen [36]. Angular momentum is conserved for the entire system about the stance foot and the swing leg about the hip. The conservation equations are as follows:

Conservation of angular momentum about the stance foot:

$$\begin{aligned} & m_h\overrightarrow{Oh^-} \times \overrightarrow{V_h^-} + m_s\overrightarrow{Os_{st}^-} \times \overrightarrow{Vs_{st}^-} + m_s\overrightarrow{Os_{sw}^-} \times \overrightarrow{Vs_{sw}^-} + m_t\overrightarrow{Ot_{st}^-} \times \overrightarrow{Vt_{st}^-} \\ & \quad + m_t\overrightarrow{Ot_{sw}^-} \times \overrightarrow{Vt_{sw}^-} \\ = & \; m_h\overrightarrow{Oh^+} \times \overrightarrow{Vh^+} + m_s\overrightarrow{Os_{st}^+} \times \overrightarrow{Vs_{st}^+} + m_s\overrightarrow{Os_{sw}^+} \times \overrightarrow{Vs_{sw}^+} + m_t\overrightarrow{Ot_{st}^+} \times \overrightarrow{Vt_{st}^+} \\ & \quad + m_t\overrightarrow{Ot_{sw}^+} \times \overrightarrow{Vt_{sw}^+} \end{aligned} \tag{3.20}$$

Conservation of angular momentum about the hip:

$$\begin{aligned} m_s\overrightarrow{Hs_{sw}^-} \times \overrightarrow{Vs_{sw}^-} + m_t\overrightarrow{Ht_{sw}^-} \times \overrightarrow{Vt_{sw}^-} \\ = m_s\overrightarrow{Hs_{sw}^+} \times \overrightarrow{Vs_{sw}^+} + m_t\overrightarrow{Ht_{sw}^+} \times \overrightarrow{Vt_{sw}^+} \end{aligned} \tag{3.21}$$

where the terms with $\vec{O}$ denotes distance vectors from the stance foot contact to the point masses.

The terms with $\vec{H}$ are distance vectors from the hip to the swing leg masses. Terms with $\vec{V}$ denote velocity vectors of the masses. Figure 3-4 was used in the derivation of the above vectors. Equation 3.20 and equation 3.21 were simplified and rearranged into the form of equation 3.19. Details of the vectors in equation 3.20 and equation 3.21 and the transition matrices Q^+ and Q^- are expressed in section A-2 of Appendix A.

3.1.2.3 Locked knee dynamics

After the knee strike, the biped model walks with two straight legs. This swing phase corresponds to the dynamics of an inverted double pendulum. After knee lock $q_3 = q_2$ and $\dot{q}_3{}^+ = \dot{q}_2{}^+$, which reduces the number of generalised coordinate to four. Equation 3.17 was obtained for locked knee dynamics using equation 3.1 to equation 3.16, and the details of matrices H, B, G, τ, q, $\dot{q}$ and $\ddot{q}$ are found in section A-3 of Appendix A.

3.1.2.4 Heel Strike Transition

Heel strike occurs when the swing foot makes contact with the ground. Heel strike was assumed to take place only after knee strike. Figure 3-5 [36] illustrates the heel strike process.

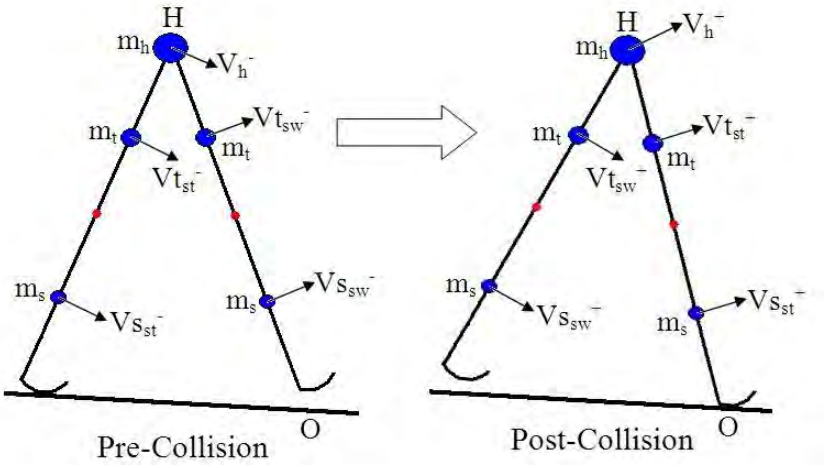

Figure 3-5: Pre-and post-collision configuration of heel strike

The transition of the angles and angular velocities are as shown in equation 3.22 and equation 3.23 respectively.

$$\begin{bmatrix} q_1^+ \\ q_2^+ \\ q_3^+ \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_1^- \\ q_2^- \end{bmatrix} \tag{3.22}$$

$$Q^+(q_1, q_2, \dot{q}_1, \dot{q}_2) \begin{bmatrix} \dot{q}_1^+ \\ \dot{q}_2^+ \end{bmatrix} = Q^-(q_1, q_2, \dot{q}_1, \dot{q}_2) \begin{bmatrix} \dot{q}_1^- \\ \dot{q}_2^- \end{bmatrix} \tag{3.23}$$

The transition matrices Q^+ and Q^- of the above were derived based on the assumption of conservation of angular momentum during heel strike. Angular momentum is conserved for the entire system about the colliding foot and the non-colliding leg about the hip. The conservation equations are as follows:

Conservation of angular momentum about the colliding foot

$$\begin{aligned} & m_h\overrightarrow{Oh^-} \times \overrightarrow{V_h^-} + m_s\overrightarrow{Os_{sw}^-} \times \overrightarrow{Vs_{sw}^-} + m_s\overrightarrow{Os_{st}^-} \times \overrightarrow{Vs_{st}^-} + m_t\overrightarrow{Ot_{sw}^-} \times \overrightarrow{Vt_{sw}^-} \\ & \quad + m_t\overrightarrow{Ot_{st}^-} \times \overrightarrow{Vt_{st}^-} \\ = {} & m_h\overrightarrow{Oh^+} \times \overrightarrow{Vh^+} + m_s\overrightarrow{Os_{st}^+} \times \overrightarrow{Vs_{st}^+} + m_s\overrightarrow{Os_{sw}^+} \times \overrightarrow{Vs_{sw}^+} + m_t\overrightarrow{Ot_{st}^+} \times \overrightarrow{Vt_{st}^+} \\ & \quad + m_t\overrightarrow{Ot_{sw}^+} \times \overrightarrow{Vt_{sw}^+} \end{aligned} \tag{3.24}$$

Conservation of angular momentum about hip

$$m_s\overrightarrow{Hs_{st}^-} \times \overrightarrow{Vs_{st}^-} + m_t\overrightarrow{Ht_{st}^-} \times \overrightarrow{Vt_{st}^-} = m_s\overrightarrow{Hs_{sw}^+} \times \overrightarrow{Vs_{sw}^+} + m_t\overrightarrow{Ht_{sw}^-} \times \overrightarrow{Vt_{sw}^+} \tag{3.25}$$

where the terms with $\vec{O}$ denotes distance vectors from the colliding foot contact to the point masses.

The terms with $\vec{H}$ are distance vectors from the hip to the non-colliding leg masses. Terms with $\vec{V}$ denote velocity vectors of the masses. Figure 3-5 was used in deriving the above vectors. Equation 3.24 and equation 3.25 were simplified and rearranged into the form of equation 3.23. Details of the vectors in equation 3.24 and equation 3.25, and the transition matrices Q^+ and Q^- are expressed in section A-4 of Appendix A.

3.1.2.5 Joint Torques

Actuation of the hip joint was selected as the control parameter that controls chaos and executes active walks. The actuator will provide energy to the system through torque. Besides the calculation of torque magnitudes required for active control of the hip joint, the magnitude of knee lock must also be known. Torque is required to keep the knee joints locked in position. Knee lock of the stance knee is required throughout locomotion and the locking of the swing knee is required after the knee strike.

The magnitude of the hip joint torque required, depends on the type of walk and control. For passive walking (down slope walking) no torque is required as the generalised torques are zero. For active walking, torque values need to be calculated using inverse dynamics. This requires the motions of the biped to be known. From known trajectory flows, the joint torques required to produce such locomotion are determined. In dynamic modelling, joints can alter between being active or passive. Since not all the joints are actuated, especially in the stance foot joint, it will be impossible to execute all the joint torque requirements. The joint in the stance foot is an imaginary joint as the foot is not physically pivoted to the ground. One way to achieve active walking is through trial and error. This is to guess a value for the hip torque leaving other joints passive. The stability of the locomotion is then determined using forward dynamics. If the system is stable, the hip torque is valid and vice versa. As for walking control, the hip torque is calculated from algorithms which are presented in Chapter 4. Walking control is only applicable to stable locomotion as it is used to alter performances, not to maintain stability.

To calculate the locking torques at knee joints, the biped was modelled as a 4-link inverted pendulum. Figure 3-2 shows the 4 links consisting of the stance shank, stance thigh, swing thigh and swing knee. The reason for using a 4 linked model was to model the stance knee joint using an additional generalised angle. The generalised angle (q_4) is the angle between the stance shank and the normal of the ground. In the unlocked and locked knee dynamics, generalised angle (q_4) was not required as part of the modelling. The two links of the stance leg are always aligned due to the locking of its knee. The locked knee effect results in generalised angle (q_4) to equal to generalised angle (q_1) and hence the stance knee joint was not included as part of the modelling. By using the dynamic equations of the 4 link model, the torque values of all joints can be determined using inverse dynamics. The biped trajectory was first simulated using the 3 and 2 linked model. By substituting the trajectories into the 4 link model equations, all joint torques were determined using the following equations:

$$\begin{aligned} u_1 &= \tau_1 + \tau_2 + \tau_3 + \tau_4 \\ u_4 &= \tau_2 + \tau_3 + \tau_4 \\ u_2 &= \tau_3 + \tau_4 \\ u_3 &= \tau_4 \end{aligned} \tag{3.26}$$

where τ_1, τ_2, τ_3 and τ_4 are the left hand sides of equation 3.17 for a 4 linked model with $q = [q_1 \quad q_4 \quad q_2 \quad q_3]^T$.

3.2 Walking Simulations

The path of locomotion and the dynamics of the biped model were obtained through simulations of its walking algorithms. To perform trajectory simulations, model conditions and assumptions were first defined followed by the simulation of the walking phases.

3.2.1 Simulation Conditions and Assumptions

The conditions and assumptions used in the simulations are as follows: Point contact of the foot with the ground was implied as the feet are curved. No slip condition was assumed. All the pitch motions were assumed not to be bounded by feet curvature lengths. Heel strike collisions were assumed to be inelastic and double support phase was assumed to be instantaneous. Knee strike was assumed to occur before heel strike with the knee constrained not to hyperextend. Knee locks were assumed to occur instantaneously after knee strike. Knee collapse, which includes both stance knee and swing knee collapse, was assumed not to occur.

3.2.2 Walking Step Simulation Procedures

The simulations were performed using Matlab®. To simulate the trajectories of locomotion, forward dynamics principles were used. See Appendix B for Matlab codes of biped motion generation. Matlab's nonlinear equation solver ode45 was used to solve the dynamic equations. Ode45 is the solver that uses 4^{th} and 5^{th} order Runge Kutta formulae to solve non-stiff ODEs. The absolute and relative error tolerance used in computing ODEs is 10^{-10}. Absolute error measures the absolute difference between the actual and measured values. Relative error is the ratio of absolute error divided by the actual value. Small tolerances were used for accuracy as chaotic systems are sensitive to deviations. Simulation speed does not change significantly when using such small tolerance. All simulations and calculations were conducted by coding Matlab's M-File. For solving the dynamics of the biped model, the generalised coordinates q are dependant variables with time t as the independent variable.

3.2.2.1 Unlocked Knee Swing Phase

The simulation of one single walking step down a slope with angle γ, starts from the unlocked knee swing phase. Since Matlab ODE solvers can only solve first order equations, equation 3.17 (an autonomous second order equation) must be reduced to first order. Instead of using the dependant variable vector as $q = [q_1, q_2, q_3]^T$, the following dependant variables were used:

$$q = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T \tag{3.27}$$

The time derivative of equation 3.27 is then:

$$\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \ddot{q}_1, \ddot{q}_2, \ddot{q}_3]^T \tag{3.28}$$

By rearranging equation 3.17, the following first order ODE was obtained:

$$\dot{q} = Mq + N \tag{3.29}$$

with

$$M = \begin{bmatrix} 0 & I \\ 0 & -[H]^{-1}[B] \end{bmatrix} \tag{3.30}$$

$$N = \begin{bmatrix} 0 \\ -[H]^{-1}[G] + [H]^{-1}[\tau] \end{bmatrix} \tag{3.31}$$

where the symbol *I* is a 3 by 3 identity matrix.

The matrices *H, B, G* and τ are the unlocked knee dynamics matrices. Equation 3.29 was solved with initial conditions $q_0 = [q_{0_1}, q_{0_2}, q_{0_3}, \dot{q_{0_1}}, \dot{q_{0_2}}, \dot{q_{0_3}}]^T$ until condition 3.32 was met. This is when knee strike occurs.

$$q_2 = q_3 \tag{3.32}$$

3.2.2.2 Knee Strike Phase

Once condition 3.32 is met, knee strike transition occurs. This is the transition of the last state of the unlocked knee swing phase (dependant variable vector 3.27). The pre-collision dependant variable $q^- = [q_1^-, q_2^-, q_3^-, \dot{q}_1^-, \dot{q}_2^-, \dot{q}_3^-]^T$ transforms into a post-collision dependant variable $q^+ = [q_1^+, q_2^+, \dot{q}_1^+, \dot{q}_2^+]^T$ using equation 3.33.

$$q^+ = W(q_1^-, q_2^-, q_3^-)q^- \tag{3.33}$$

with

$$W(q_1^-, q_2^-, q_3^-) = \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & 0 \\ & 0 & & [(Q^+)^{-1}Q^-] \end{bmatrix} \tag{3.34}$$

where the symbols Q^+ and Q^- are the knee strike transition matrices.

The matrix $[(Q^+)^{-1}Q^-]$ is a 2 by 3 matrix where the superscript '-1' denotes matrix inverse. Note that the $q_2 = q_3$ and $\dot{q}_2 = \dot{q}_3$ conditions hold until the end of the locked knee swing phase.

3.2.2.3 Locked Knee Swing Phase

After knee strike, the dependant variables are reduced in dimension due to the knee lock condition 3.32. The dependant variable and its derivative then become as equation 3.35 and equation 3.36 respectively.

$$q = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T \tag{3.35}$$

$$\dot{q} = [\dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2]^T \tag{3.36}$$

The above dependant variables in equation 3.35 and equation 3.36, and matrices *H, B, G* and τ from the locked knee dynamics were substituted into equation 3.29. Equation 3.29 was solved until condition 3.37 was met. Condition 3.37 is the relationship between q_1 and q_2 during the instant of the double support phase.

$$q_1 = -q_2 + 2\varepsilon t \tag{3.37}$$

3.2.2.4 Heel Strike Phase

Once condition 3.37 is met, heel strike transition occurs. This is the transition of the last state of the locked knee swing phase (dependant variable q in equation 3.35). The pre-collision dependant variable $q^- = [q_1^-, q_2^-, \dot{q}_1^-, \dot{q}_2^-]^T$ transforms into a post-collision dependant variable $q^+ = [q_1^+, q_2^+, q_3^+, \dot{q}_1^+, \dot{q}_2^+, \dot{q}_3^+]^T$ using the following equation:

$$q^+ = W(q_1^-, q_2^-)q^- \tag{3.38}$$

with

$$W(q_1^-, q_2^-) = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & 0 & \\ 1 & 0 & & \\ & 0 & [(Q^+)^{-1}Q^-] & \\ & & QQ & \end{bmatrix} \tag{3.39}$$

where the symbols Q^+ and Q^- are the heel strike transition matrices.

The matrix $[(Q^+)^{-1}Q^-]$ is a 2x2 matrix where the superscript '-1' denotes matrix inverse. The symbol QQ is the second row vector of the matrix $[(Q^+)^{-1}Q^-]$. For the simulation of the next step, the end condition of the heel strike was used as the new initial condition for the unlocked knee swing phase.

3.2.3 Simulation Results of a Passive Walking Step

The trends of a typical passive walking step produced through down slope walking are presented here. Typical aspects of walking that is present in all gaits are described. The simulation result presented here provides a better understanding of biped gait in terms of trajectory flow and dynamic behaviours. All the figures are generated from the data of the same step.

Figure 3-6 shows the different stages of a walking step. Frame shots of walking are overlapped to show the progression of the leg swings relative to one fixed spot. The stance leg pitches forward with its knee locked throughout. The contact point of the stance foot moves forward as the foot pitches on its arc. The swing leg which was initially straight then bends in the knee while swinging forward. The knee bend creates a clearance height between the swing feet and the ground. The heel strike impact is larger if the ground clearance value drops from a high to a low value shortly before heel strike.

The speed of a walking step for the biped model is determined using the equation 3.40:

$$Step\ Speed = Step\ Length / Step\ Time \tag{3.40}$$

Step time is the time that the biped takes to complete one step and step length is the distance travelled of the step. Step length is calculated using the equation 3.41:

$$Step\ Length = x_{swing\ foot\ contact} - x_{swing\ foot\ contact} \tag{3.41}$$

With

$$x_{stance\ foot\ contact} = x_1 - l_t \sin(-q_1) - l_k \sin(-q_1 + \varepsilon_k) \tag{3.42}$$

$$x_{swing\ foot\ contact} = x_1 + l_t \sin(q_2) + l_k \sin(q_2 - \varepsilon_k) \tag{3.43}$$

Coordinates $x_{stance\ foot\ contact}$ and $x_{swing\ foot\ contact}$ are calculated using the starting and ending step positions respectively. The average walking speed is simply the distance travelled divided by the total time (speed = distance/time). The step length and time of the walking step in figure 3-6 are 0.3241 m and 0.6768 s respectively. This corresponds to the step speed of 0.4788 m/s.

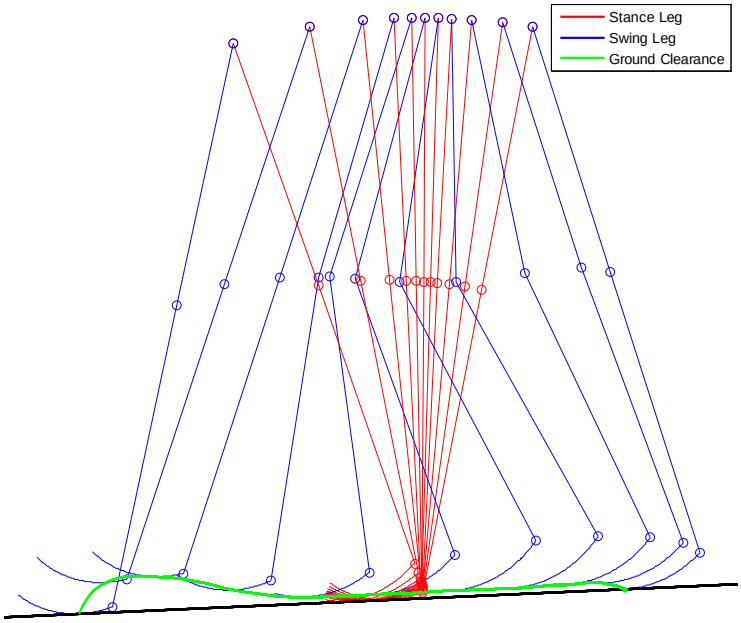

Figure 3-6: Overlapped sequence frames of a step

Figure 3-7 shows the plot of the generalised coordinates as a function of time. The figure is divided into two subplots with figure 3-7 (a) and (b) being the angular and angular velocity flows respectively. The stance leg angle q_1 increases throughout the step. Initially the swing thigh and shank rotate in opposite directions and join after knee strike. The knee and heel strikes are abbreviated as 'KS' and 'HS' respectively. In figure 3-7 (a), the swing leg angle after the knee strike decreases, then increases until heel strike. This is the swing leg retraction which helps to stabilise bipedal gaits [37]. The knee strike velocity transitions of the swing leg are instantaneous as shown in figure 3-7 (b).

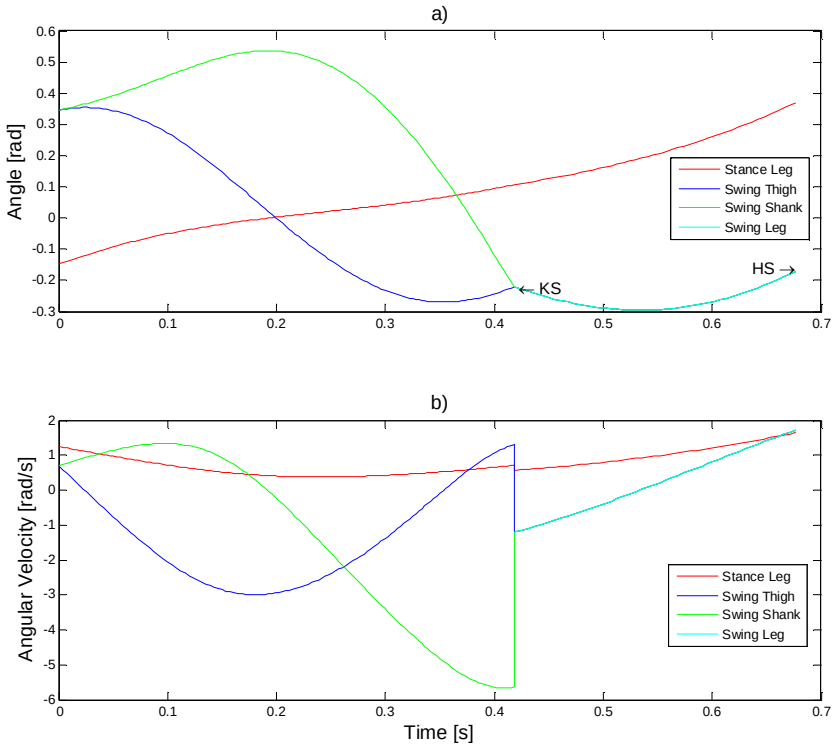

Figure 3-7: Plot of generalised coordinates versus time- a) Angular generalised coordinates, b) Angular velocity generalised coordinates

The phase space of the model ranges from 3 to 6 DOF. It is difficult to visualise phase plots with more than three variables in three dimensional or greater space. The generalised coordinates are plotted in two figures. Figure 3-8 shows the phase curve of the stance leg which is the plot of q_1 versus $\dot{q}_1$ of a step. Both knee and heel strike collisions can be identified by locating the sharp changes of the phase curve. It can be seen from the figure that the step is not a periodic one as the starting and ending points of the phase curve do not join. For periodic cycles, the absolute distance between the start and the end points of the phase curve is zero. If the distance is large, walking is most likely to be unstable and fall usually follows soon. Figure 3-9 shows the phase curves of the swing leg which is the plot of the swing thigh and shank's angle versus its own angular velocity. At the start of the step, the phase curve of the thigh and shank separates apart. Both join after knee strike, indicated by the sharp changes of the curves. Similar to the stance leg phase plot, the distance between start and end points indicates periodicity and approximates stability.

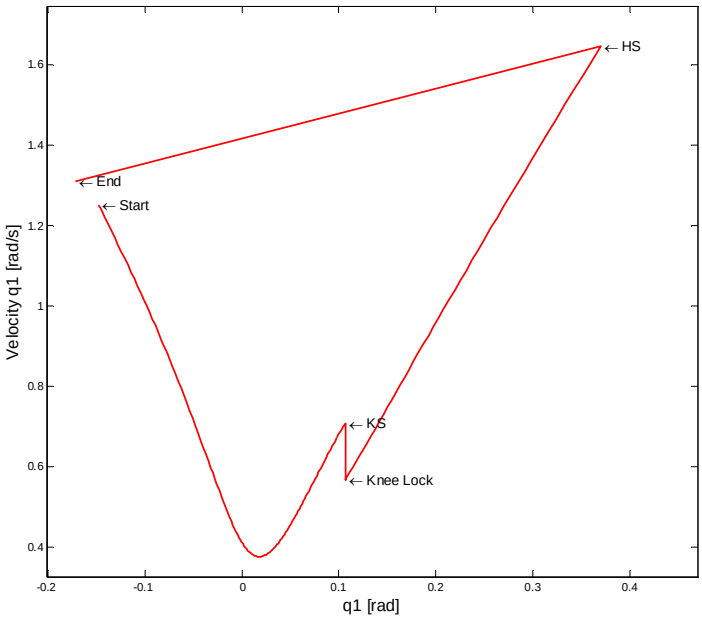

Figure 3-8: Phase curve of the stance leg

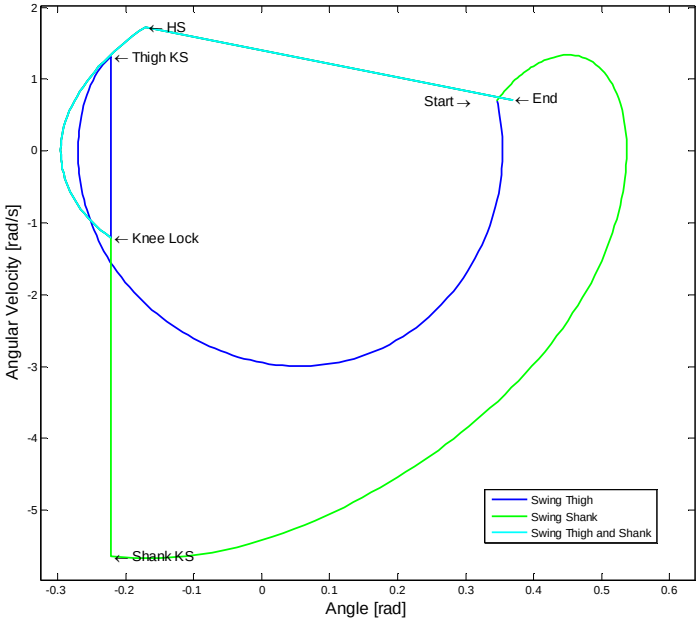

Figure 3-9: Phase curves of the swing leg

Figure 3-10 is the plot of ground clearance (GC) as a function of time. Ground clearance is the minimum distance between the base of the swing foot and the ground.

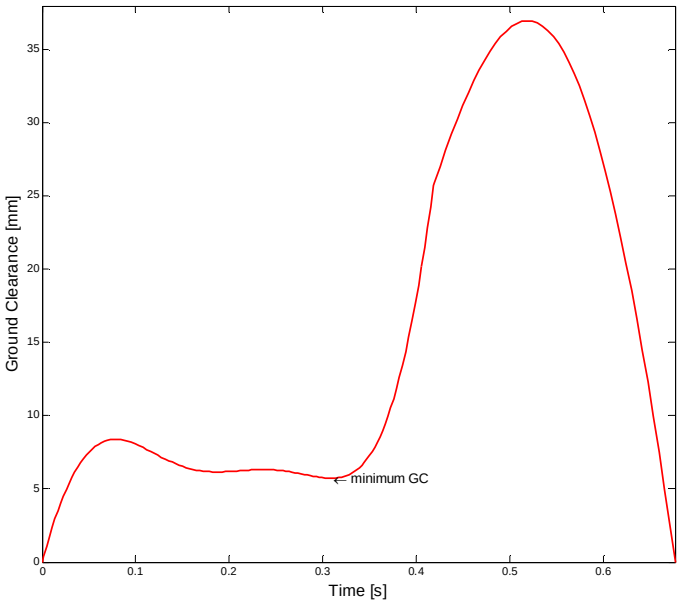

Figure 3-10: Ground clearance as a function of time

It is calculated using the following equations:

Height from ground to hip joint:

$$h_{hip} = l_t\, cos(q_1) + l_k cos(-q_1 + \varepsilon_k) + r_s \tag{3.44}$$

Height from hip to swing knee joint:

$$h_{hip\ to\ knee} = l_t\, cos(q_2) \tag{3.45}$$

Height from swing knee joint to swing foot base:

$$h_{knee\ to\ foot\ base} = l_k\, cos(q_2 - \varepsilon_k) + r_s \tag{3.46}$$

$$GC = h_{hip} - h_{hip\ to\ knee} - h_{knee\ to\ foot\ base} \tag{3.47}$$

The calculation of ground clearance is used to determine the occurrence of foot scuffing (GC<0). Foot scuffing is when the swing foot collides with the ground before the knee strike occurs. Swing knee collapse normally occurs as a result of foot scuffing which causes the biped to fall forward, around the time when the stance leg is perpendicular to the ground. This is usually the time associated with the occurrence of the smallest local minimum of the GC plot. Ground clearance also measures the sensitivity of the biped to uneven surfaces. A step with large GC means that it can avoid taller bumps of uneven surfaces without foot scuffing occurring. The GC range of figure 3-10 is between 6 mm and 7 mm which are much lower than the desired value for experiments (GC > 15mm). Large GC is associated with large knee flexions. Knee flexion is maximised when the angular acceleration of the shank swing is maximised. The acceleration rate is dependent on model parameters.

Figure 3-11 shows subplots of the joint torques against time. All the subplots peak at knee strike which is about 0.415 s after start. The peak torques shoot above the plot scales for a very short moment of time.

The peak values are much higher than the torques of the remaining times (at least 10 times higher). Such high peaks only occur theoretically as knee strike is modelled to lock instantaneously. In reality joint torque does not peak as high and is also only momentary (both knee and heel strikes). Only long term torque effects were considered in the biped designing process. Figure 3-11 (a) shows the stance foot joint torque. This is an imaginary joint which exerts almost zero torques. The knee locking torque for stance knee is shown in figure 3-11 (b). The sign of this torque is negative throughout (except for knee strike) which means that the knee joint experiences hyperextension torques. Hyperextension torque is caused by the shift in COM as a result of the geometry of roll-over feet which alters the direction of moments that act on the knee joint. The hyperextension torque is beneficial as it keeps the stance knee locked. Mechanical knee caps were used to prevent a hyper-extended knee. Figure 3-11 (c) shows that the hip joint torque is zero for passive walking. Figure 3-11 (d) shows the joint torque required of the swing knee. Before knee strike, zero torque is required. After knee strike, locking torque is required as the swing shank is constrained to move with its thigh. The knee lock strength requirement is based on the maximum of the swing joint torque (after knee strike).

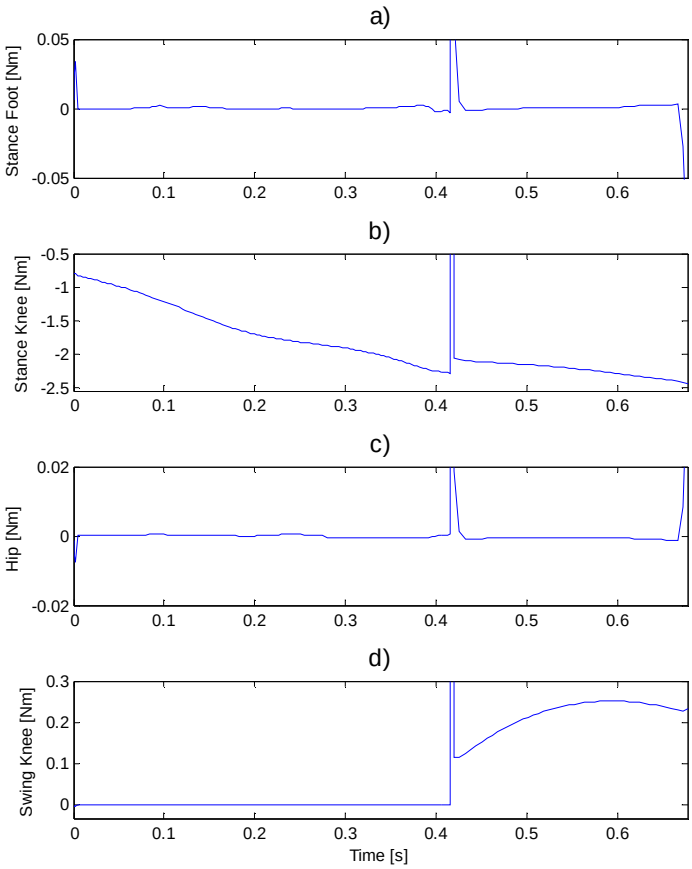

Figure 3-11: Joint torques versus time- a) Stance foot joint, b) Stance knee joint, c) Hip joint, d) Swing knee joint

3.3 Parameter Optimization

The parameters of the biped model were optimised using the following two stability quantifiers: ground clearance and basin of attraction. Range sweep of the model parameters were conducted against the

stability quantifiers. The simulated results show the relationship between model parameter and its stability.

3.3.1 Ground Clearance and Basin of Attraction

Ground clearance and basin of attraction (BOA) calculations were used to determine the range of parameters for which walking is stable. BOA is defined as the subset of phase space where all flow starting within the subset remains over time. All trajectories within the BOA are drawn toward its attractor/s. An attractor may be a point, a curve, or a manifold where nearby trajectories converge over time.

The relationship between model parameters and stability robustness were determined through simulations. Ground clearance determines the stability of walking with regard to the condition of the traversing plane. A small GC value means that the system is sensitive to ground conditions such as small protruding edges. BOA measures stability in terms of the size of initial conditions space that produces stable walking. Larger BOAs compensate for larger deviation between desired and actual initial conditions. With large BOA, the initial conditions may be slightly off but will still generate stable locomotion. Schwab and Wisse have shown that quantifying a system's stability using BOA is probably a better approach than calculating the stability of the BOA's fixed point (definition in Chapter 4) [20] which have shown that there is no obvious relationship between the stability of BOA and its fixed point.

3.3.2 Simulation Conditions and Assumptions

Simulations of GC and BOA used the same model parameters but with different conditions. The model parameters were chosen such that the trends produced from simulations are amplified. Parameters used in simulations are listed in table 3-1. Model parameters were varied in simulations in order to determine the relationship between the varied parameters and stability quantifiers (GC and BOA). Two of the model's parameters were varied in each simulation. Variable parameters were accordingly paired to determine the possible relationship between the two. Six simulations have been conducted for each stability quantifier using six paired variable parameters. The paired variables are as shown in table 3-2.

Table 3-1: Simulation Parameters

Masses	[kg]
m_h	10
m_t	7
m_s	1
Moments of Inertia	[kg .mm^2]
I_h	3500
I_t	70000
I_s	30000
Lengths	[mm]
l_t	500
l_s	500
b_2	250
c_2	0
b_1	250
c_1	15
w_f	65
r_s	130
Slope Angle	[rad]
γ	0.07

Table 3-2: Variable Parameter Pairs of Simulations

Simulation No.	Variable Parameters	Range		Units
		Minimum	Maximum	
1	m_h	1	10	[kg]
	m_t	1	10	
2	m_t	1	9	[kg]
	m_s	0.1	3	
3	l_t	200	500	[mm]
	l_s	200	500	
4	c_1	-50	65	[mm]
	c_2	-50	50	
5	b_1	100	400	[mm]
	b_2	100	400	
6	r_s	0	400	[mm]
	w_f	0	200	

The moments of inertia of body segments were assumed to be proportional to the segment's mass. The relations with units of [kg. mm^2] were used in all simulations as shown in equation 3.48.

$$I_h = 3500 m_h / 10$$
$$I_t = 70\,000 m_t / 7 \qquad (3.48)$$
$$I_s = 30\,000 m_s$$

3.3.2.1 Ground Clearance Simulation Conditions and Assumptions

Ground clearance was determined for the time span of the unlocked knee swing phase. Simulation of GC after knee strike was ignored because foot scuffing after knee strike is simply heel strike. Equation 3.47 was used in the calculation of the GC. The smallest local minimum of GC was recorded for every parameter variation. All negative GC was recorded as zero. The initial condition used for all GC simulation is shown in equation 3.49.

$$\begin{aligned} q_1 &= -0.350 \text{ rad} \\ q_2 &= -q_1 + 2\varepsilon_t \\ q_3 &= q_2 \\ \dot{q}_1 &= 1.600 \text{ rad/s} \\ \dot{q}_2 &= 0.960 \text{ rad/s} \\ \dot{q}_3 &= \dot{q}_2 \end{aligned} \tag{3.49}$$

Results from the simulations are dependent on initial conditions. A change in initial conditions, such as from a small step to a large step, will produce results that have the same trend pattern but different magnitudes of ground clearance height.

3.3.2.2 Basin of Attraction Simulation Conditions and Assumptions

The BOA simulations categorise initial condition points as stable or unstable. In simulations, the initial conditions were categorised as stable if they do not fall within 50 steps. Note that all BOA simulations in this dissertation do not take foot scuffing into account in the results. The ranges of initial conditions (dependant variable q in equation 3.27) are shown in equation 3.50.

$$\begin{aligned} q_1 &\in [-0.32, -0.15] \text{ rad} \\ q_2 &= -q_1 + 2\varepsilon_t \\ q_3 &= q_2 \\ \dot{q_1} &= \in [0.8, 0.16] \text{ rad/s} \\ \dot{q_2} &= 0.6\dot{q_1} \\ \dot{q_3} &= \dot{q_2} \end{aligned} \tag{3.50}$$

Only two of the initial conditions are variables (q_1 and $\dot{q_1}$) where the others are dependants. The phase space of BOA in simulations is a 2D plane with axes of parameters q_1 and $\dot{q_1}$. The q_1 and $\dot{q_1}$ axes have 9 and 18 interval divisions respectively. This divides the BOA phase space into a grid with 162 cells. The simulated condition variables are located at the centre of each cell. The BOA is then approximated as the area of cells that produces stable walking. The BOA stability quantifier utilise the Cell-to-cell Mapping Method [38] which measures the number of stable cells expressed as a percentage of the total number of cells.

3.3.3 Simulation Results

Simulations were conducted on the range sweep of model parameters against the two stability quantifiers. The trend pattern or relationship between the model parameters and the quantifiers are presented.

3.3.3.1 Effects of Hip and Thigh Mass Variations

Figure 3-12 shows the simulation plots where both the hip and thigh masses are varied. Figure 3-12 (a) shows that GC is directly proportional to m_t and inversely proportional to m_h. Maximum GC is obtained when both m_t and m_h are respectively maximised and minimised. The changes in GC with 1 kg change in m_t or m_h mass are approximately 11 mm and 3 mm in height, respectively. The effect of varying m_h results in much smaller changes on GC when compared to m_t changes. Figure 3-12 (b) shows the direct proportionality of m_t and inverse proportionality of m_h to BOA size. Both plots in figure 3-12, show that stable walking is possible with low m_h but not a low m_t. For m_t less than ±1.2 kg, no stable walking can be produced (figure 3-12 (b)). Large m_t mass produces larger swing thigh acceleration which causes large knee flexions hence larger GC.

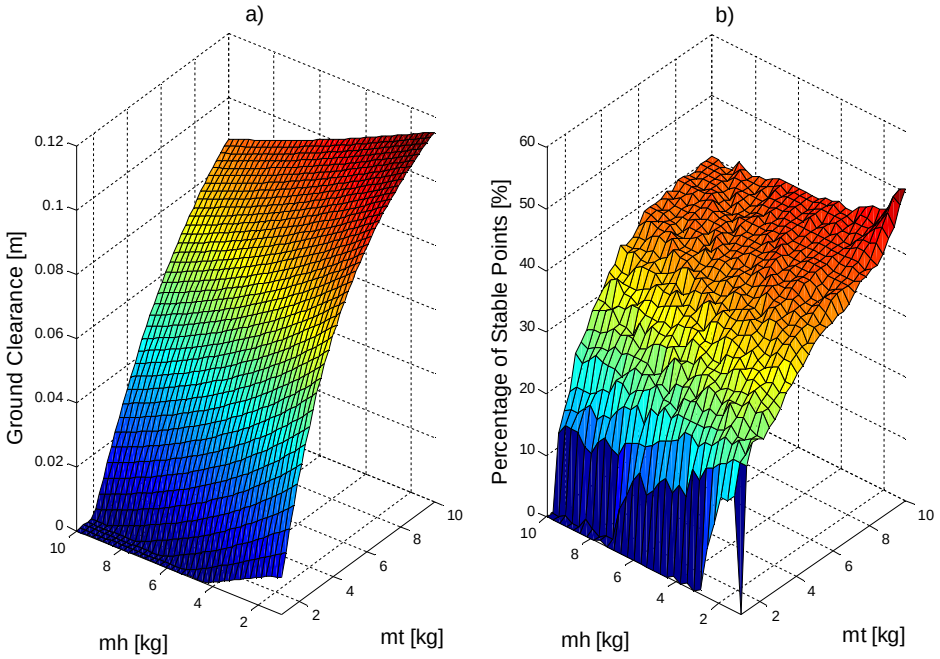

Figure 3-12: Effects of hip and thigh mass variations - a) GC plot, b) BOA plot

3.3.3.2 Effects of Thigh and Shank Mass Variations

Figure 3-13 shows are the simulations where both thigh and shank masses are varied. Figure 3-13 (a) shows that GC is directly proportional to m_t and inversely proportional to m_s. Maximum GC is obtained when both m_t and m_s are respectively maximised and minimised. Smaller m_s will result in slower acceleration of the swing shank. Slower shank accelerations cause larger knee flexions (greater GC) due to the difference in acceleration with the swing thigh. Figure 3-13 (b) shows that stable walking is possible when the ratio of m_s to m_t $\left({}^{m_s}/_{m_t}\right)$ is as follows: the lowest and highest ${}^{m_s}/_{m_t}$ ratios are 0.10 and 0.86 respectively, with an average ratio of 0.67 (stable walks). For low m_t, the ratio of ${}^{m_s}/_{m_t}$ is low and vice versa. Larger BOA is possible if m_t is increased or m_s is decreased.

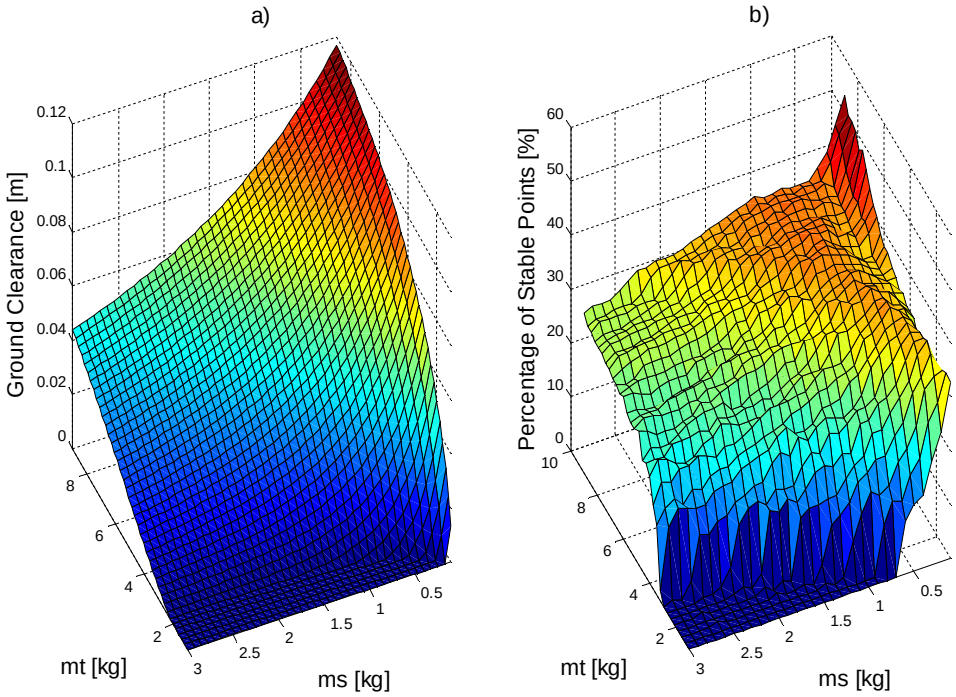

Figure 3-13: Effects of thigh and shank mass variations - a) GC plot, b) BOA plot

3.3.3.3 Effects of Thigh and Shank Length Variations

Figure 3-14 shows the simulations where both the thigh and shank lengths are varied. Figure 3-14 (a) shows that GC is directly proportional to l_t and l_s up to lengths of ±0.46 m for both. When both parameters exceed ±0.46 m, GC starts to decrease. The above does not take the sudden rise of GC spike ridge into account. The ridge with high GC values results from parameters that do not generate local GC minimum during the unlocked knee swing phase. High GC peaks are initial-condition-dependant and may not appear in other conditions. Figure 3-14 (b) shows that BOA size is directly proportional to l_t and l_s lengths. The relationship between BOA size and increase in l_t or l_s length forms a saturation curve. The saturation value is about 40 % of stable to simulated values.

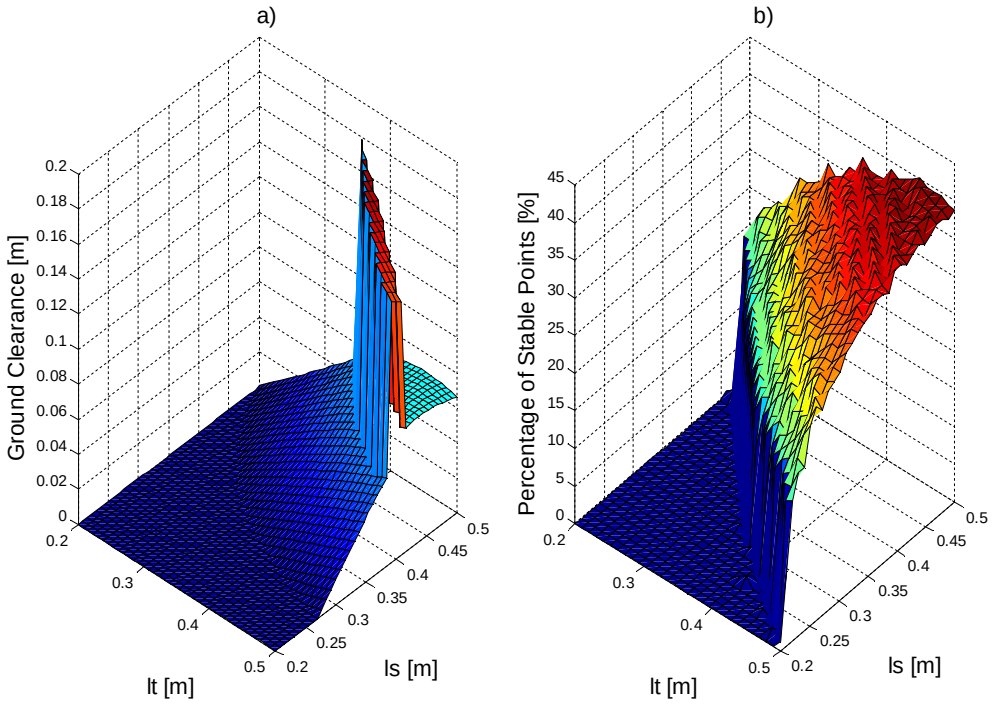

Figure 3-14: Effects of thigh and shank length variations - a) GC plot, b) BOA plot

Figure 3-14 (b) shows that no stable initial conditions exist if the following condition is met:

$$l_s + l_t < 0.7 \tag{3.51}$$

where the units are in meters.

For parameters that meet condition 3.51, stable walking is not possible for all initial conditions due to backward falls. Backward falls result from the total COM of the biped not being able to pivot over the foot contact point due to lack of forward angular momentum. Equation 3.52 is used to calculate the angular momentum of the hip.

$$Angular\ momentum\ of\ hip = r_h \times m_h \overrightarrow{V_1} \tag{3.52}$$

where the symbol $\overrightarrow{V_1}$ denotes the linear velocity of the hip.

The symbol r_h is the length from foot contact to m_h. Length r_h is directly proportional to the lengths of l_t and l_s. Condition 3.51 is the cause of backward falls which results in BOA size being zero.

3.3.3.4 Effects of Location of Centre of Mass b_2 and b_1 Variations

Figure 3-15 shows the simulations where both the b_2 and b_1 location lengths are varied. Figure 3-15 (a) shows that GC forms a saddle relation with b_2 and b_1. GC is the highest at b_2 = 0.285 m and b_1 = 0.1 m. This corresponds to m_t situated at about midpoint of the thigh and m_s being close to the knee. Knee flexion increases when m_s gets closer to the knee joint. The GC spike at b_2 = 0.4 m and b_1 = 0.1 m is due to dependence on initial condition. Figure 3-15 (b) shows that the BOA size is largest (40 % of simulated points) when b_2 and b_1 are about ±0.2 m. From the trends of figure 3-15, it is optimal to have b_2 and b_1 in

the range of 40 % to 60 % of its segment length. This is to optimise both GC and BOA size without complicating the design process such as bringing the m_s COM close to the knee joint.

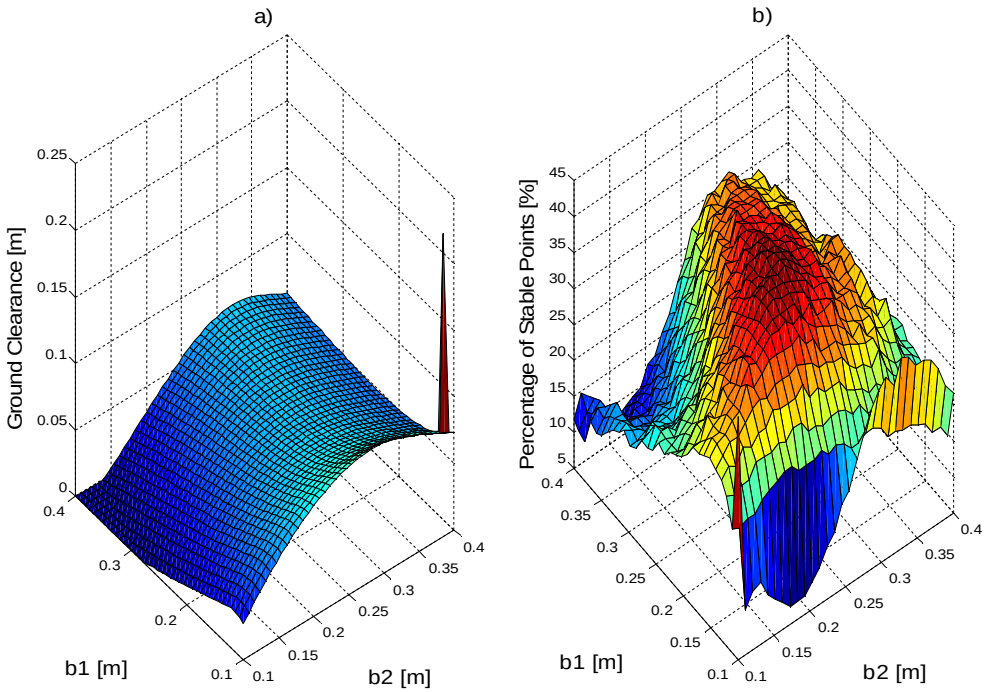

Figure 3-15: Effects of b_2 and b_1 location variation - a) GC plot, b) BOA plot

3.3.3.5 Effects of Location of Mass Centres c_2 and c_1 Variations

Figure 3-16 shows the simulations where both the c_2 and c_1 location lengths are varied. Figure 3-16 (a) shows that GC is directly proportional to c_1 and inversely proportional to c_2. The largest GC is obtained when c_1 is small and c_2 is large. As in figure 3-14 (a), the ridge peak is due to the lack of local minimum of GC and its initial-condition-dependency. Positive c_2 values cause a rapid initial swing of the swing thigh which lasts for about a quarter of the time of the unlocked knee swing phase. The swing thigh slows down which leads to a smaller knee flexion and a shorter step. Negative c_2 values have the opposite effect of being: slow at beginning, faster at end, and having a larger knee flexion and a longer step. The effect of varying c_1 on GC is much smaller when compared to varying c_2. Figure 3-16 (b) shows that BOA size is optimised when c_2 is close to zero and c_1 is at its maximum. From the results of figure 3-16, having the mass centred at its geometric centre (c_2 and c_1 =0) will be design friendly without over-compensating for the GC and BOA size.

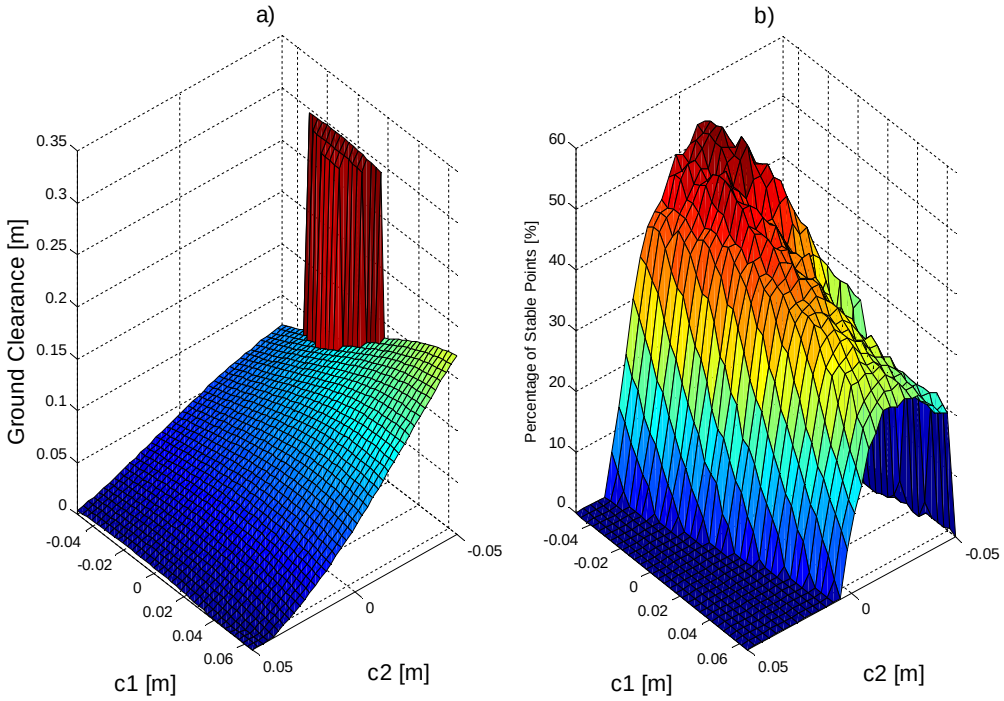

Figure 3-16: Effects of c_2 vs. c_1 location - a) GC plot, b) BOA plot

3.3.3.6 Effects of Feet Parameters w_f and r_s Variations

Figure 3-17 shows the simulations where the radius of the roll-over feet r_s and its centre offset w_f are varied. Figure 3-17 (a) shows that GC is highest when feet r_s = 0 m and w_f = 0.113 m (excluding the GC spike). The GC crest is situated according to the following equation:

$$w_f = -0.26 r_s + 0.1231 \tag{3.53}$$

where the units of r_s and w_f are in meters.

The parameters that satisfy equation 3.53 have GC > 0.02 m. The GC crest positioning alters when initial condition and other model parameters are changed. GC is also high (GC > 0.05 m) in the region formed by $r_s \in [0; 0.144]\ m$ and $w_f \in [0.051; 0.159]\ m$. GC spikes can also be found in figure 3-17 (a) due to the reason discussed previously. GC decreases until zero when r_s increases. Figure 3-17 (b) shows that as r_s increases, BOA size increase which agrees with the research findings of Wisse and van Frankenhuyzen [18]. Wisse and van Frankenhuyzen showed that increases in foot radius increases robustness (BOA size), but also increases the chance of foot scuffing. An increase in foot radius causes both the foot length and the chance of toe stubbing to increase. Similarly, smaller r_s results in larger GC but reduced BOA size. Both plots in figure 3-17, show that a horizontal shift in foot radius centre w_f improves stability. The optimal w_f offset is about 0.1 m for the simulated model. According to Franken, the w_f offset affects the acceleration of the stance leg as it changes the distance between foot contact and the total COM [39]. Figure 3-18 [39] illustrates the effects of w_f offset. For large w_f offset (w_f >>0), deceleration of the stance leg occurs due to gravitational force. Likewise if w_f = 0, acceleration of the stance leg occurs. Faster stance leg acceleration results in shorter steps as heel strike of the swing leg

occurs quicker. Slower stance leg acceleration increases step length but may also cause backward falls if the w_f offset is too large. Figure 3-17 shows that w_f is bounded to a fixed range for stable walks.

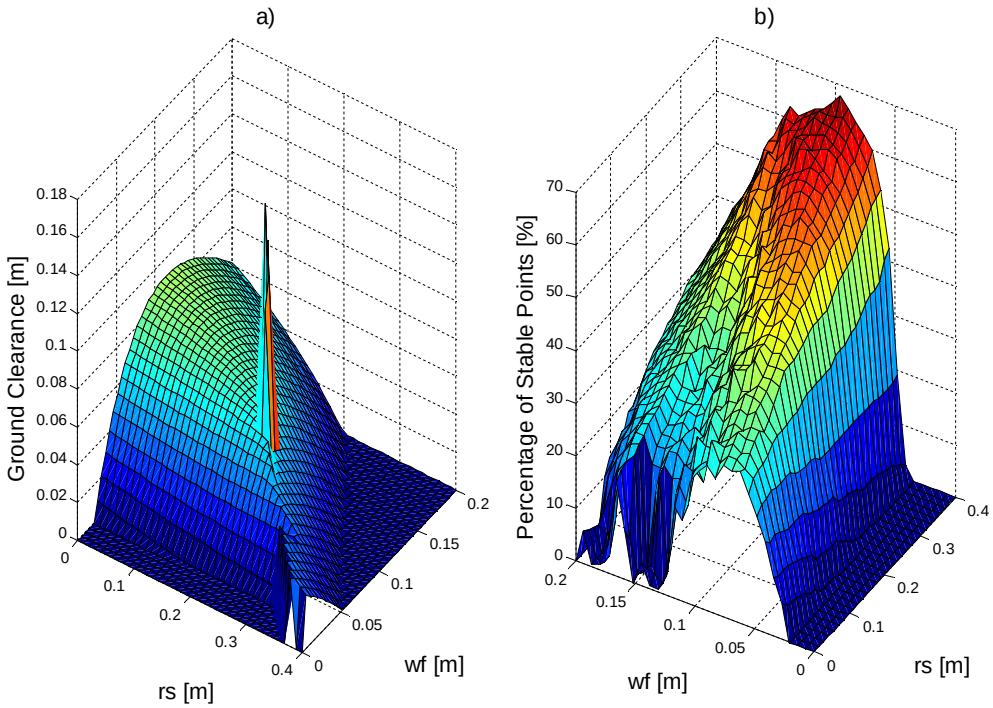

Figure 3-17: Effects of w_f and r_s variation - a) GC plot, b) BOA plot

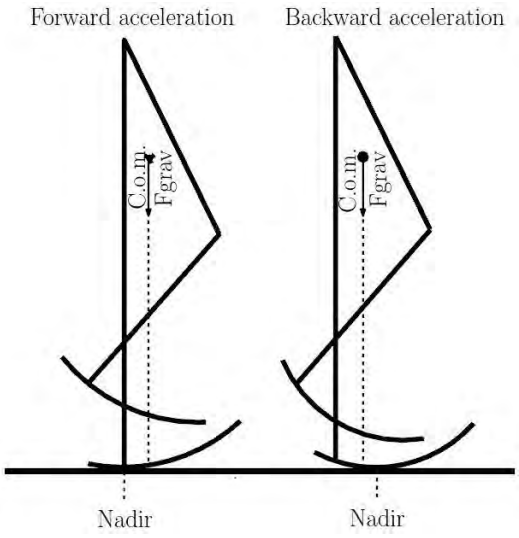

Figure 3-18: Effects w_f offset on acceleration [39]

3.4 Design Parameter Ranges

By integrating the results of GC and BOA simulations with human proportionality, a stable model with parameters of anthropometric proximity was achieved. In section 3.3, the simulations were performed on a robust model which generated a relationship between the parameters, and GC and BOA size. The simulation trends can only be used as a guide as they are dependent on parameters and initial conditions.

Actual biped parameters will not be exactly the same as the simulated model due to discrepancies. Design requirements such as the magnitude of moments of inertia, are difficult to achieve in the design. It is possible to change the dynamics of the design through small changes. Parameters that can be easily altered (manufacture-wise) will be selected as the stability tuners of the model. Typical parameters that alter stability greatly through small changes are l_t, l_s, r_s and w_f.

The design model was decided to be 0.6 m tall (hip height) as this height is the most suitable. It is large enough for operational handling while running experiments and small enough for large expensive parts not to be required. Although the simulated model is 1 m in hip height, the trend roughly applies to both models. Anthropometric parameters corresponding to the above model are listed in table 3-3. The range of stable design ranges are also shown in the table. The total mass of the design model ranges from 7.2 kg to 12.2 kg (lower body mass). When converting the design model to a full body model, the model's body mass index (BMI) ranges from 12.11 $kg.m^{-2}$ to 20.52 $kg.m^{-2}$. The anthropometric model has a total lower body mass of 9.74 kg and a BMI of 16.39 $kg.m^{-2}$. Typical human adult BMIs are in the range of 20 $kg.m^{-2}$ to 25 $kg.m^{-2}$ [40]. The design model has a smaller BMI as it has less degrees of freedom than humans. Also, larger actuation force will be required for larger mass which leads to a larger actuator being required.

The selection of the design parameter range of table 3-3 was based on the results of previous simulations. The average of the design parameter range in the table is equal to the anthropometric model parameters. Large tolerance was selected for the hip mass (33 % of anthropometric m_h) because a parameter change in the hip does not alter the stability significantly according to figure 3-12. The hip is also where the actuator is located so mass flexibility is needed. The thigh design mass has a tolerance of approximately 20 % and a minimum mass of 1.8 kg. Simulation results of figure 3.12 and 3.13 shows that stability is improved with larger m_t but this requires more actuation torque for active control. Instead, m_t was bounded such that it optimised both stability and actuation power. The shank design mass has the opposite effect to thigh mass. It should be at a minimum without over-reducing the structure of the mechanical components. The body segments' moments of inertia range has not been stipulated due to the complexity of including this as part of the design requirement.

The segment lengths of the design are dependent on each other. Condition 3.51 regarding stability applies only to the simulated model but a similar condition will govern the design model. In general, the difference between the lengths of l_t and l_s should be less than 50 % of the longer segment. The proportionality between the leg lengths and masses should also be similar to the simulated model. The selection of l_t and l_s lengths will be very close to the anthropometric model as this selection does not influence stability and provides good anthropometric appeal. The COM locations b_2 and b_1 was proposed to be at midway of the segments which agrees with the results of figure3-15 and COM is also normally situated at the mid-point of uniform sections. Similarly COM locations c_2 and c_1 were selected to lie within the segments. Feet parameters r_s and w_f of the design model differs from the roll-over foot models proposed by other researchers. The feet design parameters are based on the trend of figure 3-17. Stable feet parameters are roughly located around $r_s = 0.2\times(l_t + l_s)$ and $w_f = r_s/2$. This is $r_s = 0.106\ H_t$ and

w_f = 0.053 H_t where the symbol H_t denotes total body height in meters. McGeer proposed 0.16 H_t for r_s [1] and Hansen and Childress proposed 0.008 H_t for w_f [41].

Table 3-3: Anthropometric and Design Parameters

Masses	Anthropometric Parameters	Design Parameter Range	
	[kg]	Minimum [kg]	Maximum [kg]
m_h	2.98	2	4
m_t	2.1	1.8	2.6
m_s	1.28	0.8	1.5
Lengths	[mm]	Minimum [mm]	Maximum [mm]
l_t	277	250	300
l_s	323	300	325
b_2	-	$l_t/2 - 20$	$l_t/2 + 20$
c_2	-	-20	20
b_1	-	$l_s/2 - 20$	$l_s/2 + 20$
c_1	-	-20	40
w_f	9 = (0.008H_t) [41]	$r_s/2 - 20$	$r_s/2 + 20$
r_s	181 = (0.16H_t) [1]	$0.2\times(l_t + l_s) - 20$	$0.2\times(l_t + l_s) + 20$

3.5 Chapter Summary

Based on the dynamic simulations of the sagittal plane model, trajectory flows and stability of locomotion were determined. Trajectory paths as a function of time can be simulated if the initial condition is known. Basic understandings of locomotion were gathered using phase curve plots. A plot of joint torque was used to determine the locking torques for the knee joint designs.

Simulation routines were extended to form GC and BOA stability quantifiers. These two robustness quantifiers determine the relationship between parameters and stability. Simulations of GC determine the occurrence of foot scuffing and sensitivity of walking to ground unevenness. GC simulations trends are initial-condition-dependent and fail to determine stability beyond a foot step. The BOA simulations determine the phase space size of initial conditions that produce long-term stability. Foot scuffing was ignored in the BOA simulations. This shortfall was overcome by combining trend patterns with GC simulations. A BOA size measurer was known to be a better long-term stability quantifier.

By combining the trend pattern of stability simulations and anthropometric proportionality, the design parameter range was determined. Certain parameter ranges are governed by essential stability trends. Other parameters are based on anthropometric proportionality if they do not oppose the stability trend. The combined effects were aimed at generating stable anthropomorphic locomotion.

CHAPTER 4 CHAOS CONTROL

4.0 Introduction

Chaos control aims to direct wandering orbits to a periodic one. It does not stabilise the system through path corrections or instability preventions. Chaos control is only applicable to systems that are already stable and control only applies to orbits within the chaotic attractor. If a flow starts outside the chaotic attractor or leaves the attractor due to external disturbance, its stability cannot be maintained using chaos control. The advantage of chaos control is the ability to direct chaotic flows onto different orders of periodic orbits. The performance of the system is then flexible as it is dependent on the chosen periodic orbit of interest.

This chapter presents the methodology of chaos control and its application in a biped model. The background understanding of chaos is outlined. The methodology of the OGY method is explained in detail. The preparation procedures required for chaos control of the biped model are presented. Simulation control of a chaotic orbit into a limit cycle has been achieved with the aid of different chaos analysing tools presented in this chapter. The chapter's conclusion evaluates the application of chaos control in a biped model in terms of advantages and disadvantages. An unexpected outcome regarding the underlying chaotic system of the biped model was realized.

4.1 Understanding Chaos

Chaos is used to describe systems that are deterministic but also very sensitive to initial conditions. Initial conditions with a small difference can diverge over time due to the nature of the system (commonly known as the butterfly effect) [42]. The divergence behaviour is not due to randomness or external influences, as outputs of chaotic systems are predictable. Trajectories of chaotic systems do not cross over time [43] which means no two instantaneous conditions of any trajectory are the same in the chaotic phase space. Chaotic trajectories are also bounded within a closed region to form a chaotic basin of attraction. Since all chaotic flows are bounded to a closed region, and do not cross themselves, the phase space of a chaotic system must have at least three dimensions [9].

One of the most important properties of chaos is ergodicity. In an ergodic system, any trajectory will eventually visit all parts of the phase space after a sufficiently long time [44]. It is due to this property that any unstable periodic orbits, in the chaotic system, are attainable.

4.1.1 Unstable Periodic Orbits (UPO)

An UPO is defined as a periodic orbit that is unstable. It can be offset by the slightest change or disturbance. UPOs are characterised by stable and unstable manifolds which attract and repel nearby trajectories respectively. Where the manifolds cross, is the fixed point of UPO, which is also a saddle point. This means that trajectories near the saddle point are either attracted or repelled, except when on the fixed point. Orbit dynamics would remain constant if the orbit travel on a path of saddle points and there were no disturbances acting on it, but this characteristic is unlikely. In chaotic systems, stable and unstable manifolds cross each other many times [9]. These crossings result in large numbers of UPOs

being embedded in the chaotic system's basin of attraction [4]. Due to the attraction and repulsion of orbits around UPOs, orbit behaviour seems to be random.

4.1.2 Route to Chaos

A dynamic system may exhibit periodic, chaotic, or unstable behaviour depending on the control parameter. The control parameter is a system parameter that can be changed or varied. Typically, system behaviour changes from single period to multiple period then to chaos (infinite period), which then diverges.

In nonlinear dynamics, bifurcation is a term that refers to the splitting of system behaviour into two regions if the control parameter (also called the bifurcation parameter) is varied. Bifurcation diagrams reflect the long-term state of systems as a function of the bifurcation parameter [45]. Long-term states may be either periodic or chaotic. Figure 4-1 [46] shows the bifurcation diagram for the logistic map function. The logistic map function is a discrete equation proposed by May which approximates the biological population growth rate of species [47]. The relationship between consecutive years of population ratio is described using equation 4.1 [47]:

$$x_{n+1} = r x_n (1 - x_n) \tag{4.1}$$

where the symbol $x_n \in [0, 1]$, denotes the ratio of existing population to maximum population of year *n*.

The symbol *r* is the positive growth factor and is also the bifurcation parameter. For $r \in [0, 1)$, long-term population 'x_{n+1}' reaches zero. For $r \in [1, 3)$, long-term population reaches steady state (period 1). At $r=3$, bifurcation of period 1 into period 2 occurs as shown in figure 4-1. This is called period-doubling bifurcation (or simply period-doubling). For discrete functions, a period *n* state means that repetition of solutions occurs every *n* iteration. Period doubling continues to occur as *r* increases until *r*=3.57. Thereafter long-term states are no longer periodic as no repetition can be observed.

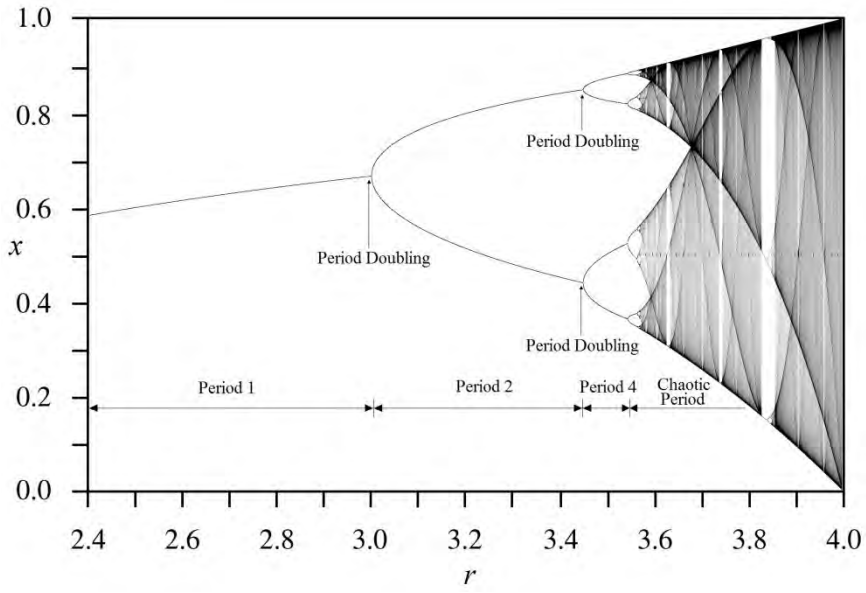

Figure 4-1: Bifurcation Diagram for the Logistic Map Function

4.1.3 Poincaré Map Function

The tool commonly used for analysing nonlinear dynamic flow is the Poincaré Map function or simply the Poincaré Map [48]. A Poincaré Map is a function that describes the dynamics between intersection points of a chosen plane section and the orbits of a continuous system and thus transforms a continuous orbit system into a recurrence map or a discrete dynamical system [61]. The advantage of using this tool is that it reduces the phase space observed by one dimension while still preserving the properties of the original system [43]. The reduced dimension advantage only holds if the plane section can be modelled as autonomous equation(s) (explicitly time-independent equations). This allows the information presented to be visualised easier through the reduced dimensionality. Figure 4-2 [5] shows how the Poincaré Map (2D) is generated, in a 3D system, by observing the intersection points.

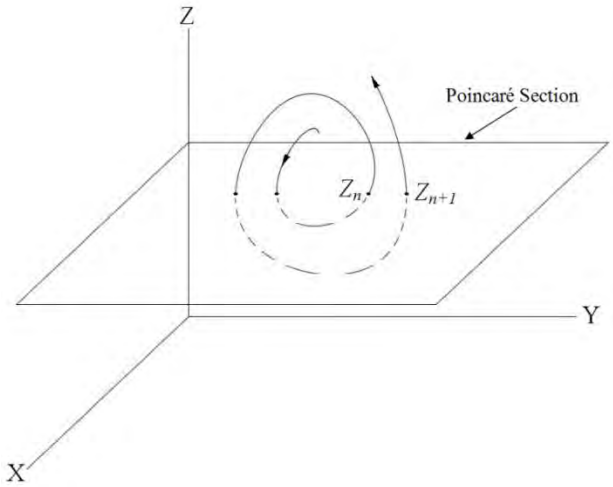

Figure 4-2: Poincaré Map formed by the intersection of the Poincaré Section and the continuous dynamical orbit - two consecutive intersection points are illustrated

The section that intersects a system flow is called the Poincaré Section. For a 3D system, the Poincaré Section (a plane) is called a Poincaré Plane. Two types of intersection directions can occur between the system trajectory and the Poincaré Section. The first is the crossing of trajectory from the one side of the section to the other (positive z-axis direction in figure 4-2) and the second is vice versa (negative z-axis direction in figure 4-2). The dynamic behaviours of intersection points can be determined through the calculation of its eigenvalues. Intersection points of periodic orbits have eigenvalues of attractors. For an intersection point that is within the BOA of an attractor, the consecutive intersection points will be withdrawn toward the fixed point(s) of the attractor. For period 1 orbits, also called limit cycles, there is only one Poincaré crossing at steady state. Similarly, a period *n* orbit will have *n* Poincaré intersections at steady state. Figure 4-3 (a) and (b) [5] illustrate the Poincaré crossings of period 1 and 2 orbits at steady states respectively. Time of travel for periodic orbits remains constant at steady state.

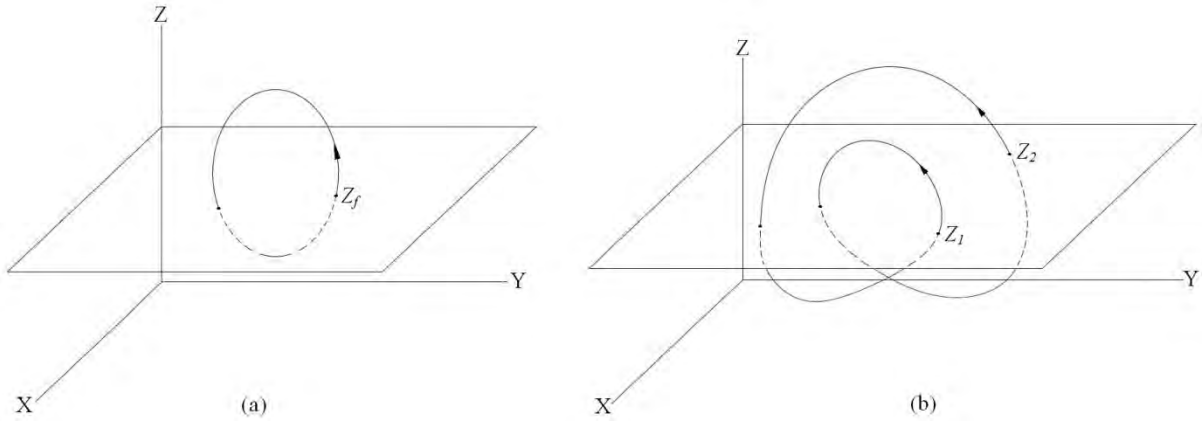

Figure 4-3: Ordering of periodic orbits based on the number of Poincaré Section intersections
a) Limit Cycle, b) Period 2 Orbit

4.2 Chaos Control Methodology [4]

Ott, Grebogi and Yorke [4] have shown that periodic orbits can be achieved by stabilising UPOs using small perturbations. Chaos control is possible due to ergodicity [43] and large number of UPOs in chaotic systems [4]. Dynamic performance of chaotic systems varies depending on the trajectory flow. Every UPO within the chaotic attractor corresponds to a steady state performance. Without control, flows within a chaotic attractor get attracted and repelled by UPO fixed points. It will take a very long time before the system flow can actually land and travel on any of the UPOs [49]. Once on the UPO, any slight disturbance will offset the system flow due to the unstable nature of the UPO. The system flow will again begin to wander within the chaotic attractor.

The chaos control process shortens the time it takes to reach the desired UPO [50]. When the system flow is within proximity of the desired UPO, a small perturbation is induced. The perturbation force will direct the system flow toward the UPO on the next returning cycle. The system flow is kept on the UPO through constant perturbation control.

The OGY method revolves around the dynamics of fixed points in a Poincaré Section. Intersection points on the Poincaré Map are expressed in terms of discrete mathematical functions. The Poincaré function maps its current state to the next recurrence point as follows [51]:

$$Z_{n+1} = f(Z_n, p_n) \tag{4.2}$$

where the symbol f is the Poincaré function, p_n the control parameter, Z_n the n^{th} state condition vector with k phase space dimensions and Z_{n+1} the consecutive recurrence point.

If the state conditions maps back to itself, then it is a fixed point (a limit cycle) as follows:

$$Z_f = f(Z_f, p_n) \tag{4.3}$$

where Z_f is the fixed point and p_f is the control parameter value corresponding to the fixed point.

The intersection between UPOs and the Poincaré section forms saddle fixed points. Note that all fixed points in a chaotic attractor are saddle points which imply that all the fixed points of chaotic systems are UPOs. A fixed point's positioning shifts with control parameter changes (as shown in figure 4-6 (b). The shift of fixed point is approximated using equation 4.4 [52]:

$$Z_f(p_{n+1}) \sim Z_f(p_n) + B(p_{n+1} - p_n) \tag{4.4}$$

where the control parameter varies from p_n to p_{n+1}.

B is the shift vector which is:

$$B = \frac{\partial}{\partial p} Z_f(p)\Big|_{p=p_n} \tag{4.5}$$

Finding a Poincaré function that describes the whole system is either complicated or impossible. For simplicity, the Poincaré function is linearised about a point which centres in the region of interest. Linearisation is a good approximate when working points are in close proximity to the linearised point. Most times the linearised point is the target point of the process; the UPO's saddle fixed point. The equation for linearising the Poincaré function about a fixed point with parameter shift is shown in equation 4.6 [52]:

$$Z_{n+1} - Z_f(p_{n+1}) \sim A(Z_n - Z_f(p_{n+1})) \tag{4.6}$$

where

$$A = \frac{\partial}{\partial Z} f(Z, p)|_{Z=Z_n} \tag{4.7}$$

Note that matrix *A* is a square matrix. The Poincaré function without the parameter shift is obtained by substituting equation (4.4) with equation (4.6). After rearranging, results equation 4.8:

$$Z_{n+1} - Z_f(p_n) = A\left(Z_n - Z_f(p_n) - B(p_{n+1} - p_n)\right) + B(p_{n+1} - p_n) \tag{4.8}$$

Matrix *A* is the Jacobian of the mapping where its eigenvalues and eigenvectors characterise the system. Equation 4.8 is the linearised form of equation 4.2 about a fixed point. Assuming Matrix *A* has *s* stable and *u* unstable, eigenvalues (refer to figure 4-4 [53]), where $s+u=k$. Let the eigenvalues λ_i of matrix *A* satisfy the following inequality:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_u| > 1 > |\lambda_{u+1}| > \cdots |\lambda_k| \tag{4.9}$$

where i =1, 2, ... k. Eigenvalues also determine the rate of convergence or divergence toward or away from the fixed point. Let e_i be the corresponding unit eigenvectors of matrix A, resulting e_i to be the stable eigenvectors in the same direction as the stable manifolds (for i=1 to u) [53].

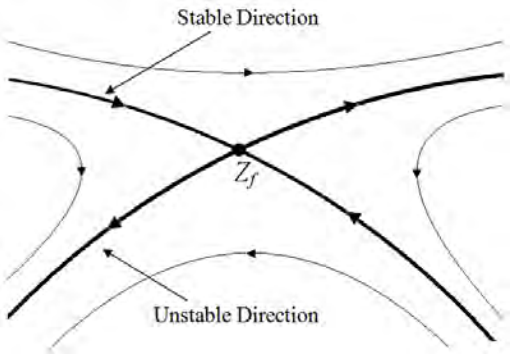

Figure 4-4: Behaviour of trajectories near a saddle point on the Poincaré Section

A state point in the neighbourhood of the fixed point will be driven by the characteristic of the manifolds as shown in figure 4-4 (thin lines). When the state point is on the manifold, it will be drawn either toward or away from the fixed point along the line of the manifold (thick lines). If the state point is at the fixed point, it will remain there under zero disturbance conditions.

Table 4-1 [9] summarises the possible types of fixed points characterised by the behaviour of the trajectories around it. Table 4-1 reflects the fixed points such as nodes or repellors (spirals or not) which are either stable or unstable. The outcome state of these fixed points can only be altered through large changes. Trajectories in systems with saddle points wander between fixed points, generates chaotic orbits. This is due to the force of attraction and repulsion on orbits. Since chaotic orbits are neither influenced by exclusively large attracting or repelling behaviour, they can be directed and guided using small perturbations.

Table 4-1: Types of fixed points categorised by their eigenvalues

Type of Fixed Point	Eigenvalues Categories	Stability
Node	$\|\lambda_i\|<1$ and Real	Stable
Repellor	$\|\lambda_i\|>1$ and Real	Unstable
Saddle	$\|\lambda_i\|<1$ with at least one $\|\lambda_i\|>1$ and allλ_i Real	Critically Stable
Spiral Node/Repellor	Real λ_i with complex conjugate λ_i pair	Stable Spiral/Unstable Spiral

Continuing from the defining of eigenvectors e_i, let unit vectors f_j satisfy the following conditions:

$$f_j \cdot e_i = 1 \text{ for } j = i \tag{4.10}$$

$$f_j \cdot e_i = 0 \text{ for } j \neq i \tag{4.11}$$

where $j = 1, 2, \ldots k$. f_j forms the contravariant basis vectors of eigenvectors e_i [53].

When perturbation is applied to the system for u cycles, the state condition Z_n then moves on to the stable manifold on the Z_{n+u} recurrence [54]. Convergence is achieved by changing the control parameter such

that the projection of the $(Z_{n+u} - Z_f)$ vector onto the stable manifold becomes zero (as shown in figure 4-5). Convergence starts when the dot products in equation 4.12 become zero [54]:

$$f_l \cdot (Z_{n+u} - Z_f) = 0 \tag{4.12}$$

where l = 1, 2, ..., u. If $f_l \cdot (Z_{n+u} - Z_f) = 0$ then $(Z_{n+u} - Z_f)$ is perpendicular to f_l.

Since f_l is also perpendicular to e_i for $i \neq l$, then $(Z_{n+u} - Z_f)$ must be parallel to e_i. $(Z_{n+u} - Z_f)$ is then parallel to all the stable direction(s) and Z_{n+u} lies on the stable manifold(s). Figure 4-5 [53] illustrates the above process for an unstable manifold.

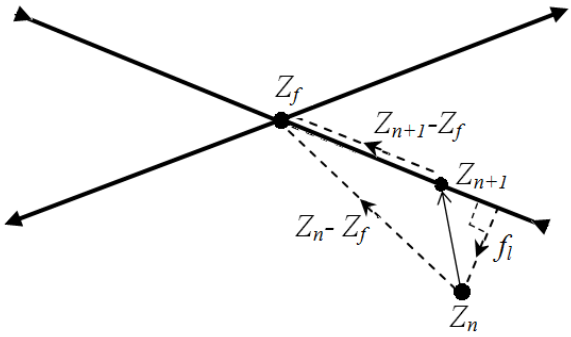

Figure 4-5: Directing the state point Z_n onto the stable manifold of the fixed point

The magnitude of the control parameter required to achieve the equality of equation 4.12 is determined in a number of steps. Equation 4.8 is substituted into equation 4.12 as shown in figure 4.13:

$$\begin{aligned}
0 &= f_1 \cdot A\left(Z_{n+u-1} - Z_f - B(p_{n+u} - p_{n+u-1})\right) + f_1 \cdot B(p_{n+u} - p_{n+u-1}) \\
0 &= f_2 \cdot A\left(Z_{n+u-1} - Z_f - B(p_{n+u} - p_{n+u-1})\right) + f_2 \cdot B(p_{n+u} - p_{n+u-1}) \\
&\vdots \\
0 &= f_u \cdot A\left(Z_{n+u-1} - Z_f - B(p_{n+u} - p_{n+u-1})\right) + f_u \cdot B(p_{n+u} - p_{n+u-1})
\end{aligned} \tag{4.13}$$

Let $\Delta p_{n+1} = (p_{n+1} - p_n)$, $\Delta p_{n+2} = (p_{n+2} - p_{n+1})$, ... $\Delta p_{n+u-1} = (p_{n+u-1} - p_{n+u-2})$ and $\Delta Z_n = (Z_n - Z_f)$. By substituting the above terms in equation 4.8, for (n+1) = (n+1) to (n+u-1), equations 4.14 are obtained:

$$\begin{aligned}
Z_{n+1} - Z_f &= A(Z_n - Z_f - B\Delta p_{n+1}) + B\Delta p_{n+1} \\
&= A(\Delta Z_n - B\Delta p_{n+1}) + B\Delta p_{n+1} \\
Z_{n+2} - Z_f &= A(Z_{n+1} - Z_f - B\Delta p_{n+2}) + B\Delta p_{n+2} \\
&= A^2(\Delta Z_n - B\Delta p_{n+1}) + AB(\Delta p_{n+1} - \Delta p_{n+2}) + B\Delta p_{n+2} \\
Z_{n+3} - Z_f &= A(Z_{n+2} - Z_f - B\Delta p_{n+3}) + B\Delta p_{n+3} \\
&= A^3(\Delta Z_n - B\Delta p_{n+1}) + A^2B(\Delta p_{n+1} - \Delta p_{n+2}) \\
&\quad + AB(\Delta p_{n+2} - \Delta p_{n+3}) + B\Delta p_{n+3} \\
&\vdots \\
Z_{n+u-1} - Z_f &= A(Z_{n+u-2} - Z_f - B\Delta p_{n+u-1}) + B\Delta p_{n+u-1} \\
Z_{n+u-1} - Z_f &= A^{u-1}(\Delta Z_n - B\Delta p_{n+1}) + A^{u-2}B(\Delta p_{n+1} - \Delta p_{n+2}) \\
&\quad + A^{u-3}B(\Delta p_{n+2} - \Delta p_{n+3}) + \cdots + AB(\Delta p_{n+u-2} - \Delta p_{n+u-1}) \\
&\quad + B\Delta p_{n+u-1}
\end{aligned} \tag{4.14}$$

The last equation of equation 4.14 is the one of interest. For simplification of the processes, matrix A is manipulated as shown in equation 4.15 to equation 4.20:

$$\begin{gathered} Ae_1 = \lambda_1 e_1 \\ Ae_2 = \lambda_2 e_2 \\ \vdots \\ Ae_k = \lambda_k e_k \end{gathered}$$

$$[Ae_1\ Ae_2 \cdots Ae_k] = [\lambda_1 e_1\ \lambda_2 e_e\ \cdots \lambda_k e_k] \tag{4.15}$$

$$A[e_1 \cdots e_k] = [e_1 \cdots e_k]\begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_k \end{bmatrix}$$

$$\therefore A = [e_1 \cdots e_k]\begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_k \end{bmatrix}[e_1 \cdots e_k]^{-1} \tag{4.16}$$

The first step of equation 4.15 is simply the definition of square matrices and its eigenvectors. Equation 4.16 is obtained from equation 4.15 by means of distribution law. The vector products between f_j and e_i vectors (equation 4.17) are taken as follows:

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix}[e_1 \quad e_2 \quad \cdots \quad e_k] = \begin{bmatrix} f_1e_1 & f_1e_2 & \cdots & f_1e_k \\ f_2e_1 & f_2e_2 & & f_2e_k \\ \vdots & & \ddots & \vdots \\ f_ke_1 & f_ke_2 & \cdots & f_ke_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{4.17}$$

$$\therefore \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix} = [e_1 \quad e_2 \quad \cdots \quad e_k]^{-1} \tag{4.18}$$

Since the vector product between f_j and e_i vectors is equal to an identity matrix, vector f_j is then an inverse (equation 4.18) of vector e_i. Equation 4.18 is substituted into equation 4.16 resulting:

$$A = [e_1 \cdots e_k]\begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_k \end{bmatrix}\begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix} \tag{4.19}$$

$$A = \lambda_1 e_1 f_1 + \lambda_2 e_2 f_2 + \cdots + \lambda_k e_k f_k \tag{4.20}$$

After multiplying out equation 4.19, equation 4.20 is obtained. The dot product between f_j and matrix A of equation 4.20 is taken as follows:

$$\begin{aligned} f_j \cdot A &= f_l \cdot (\lambda_1 e_1 f_1 + \lambda_2 e_2 f_2 + \cdots + \lambda_k e_k f_k) \\ &= \lambda_1 f_l \cdot e_1 f_1 + \lambda_2 f_l \cdot e_2 f_2 + \cdots + \lambda_k f_l \cdot e_k f_k \\ &= \lambda_1 0 f_1 + \lambda_2 0 f_2 + \cdots + \lambda_j 1 f_j + \cdots + \lambda_k 0 f_k \\ &= \lambda_j f_j \end{aligned} \tag{4.21}$$

Some terms of equation 4.21 equals zero and falls away due to equation 4.11. Similarly, the dot product between f_j and powers of matrix A of equation 4.20 results:

$$f_j \cdot A^l = \lambda_j f_j A^{l-1} = \lambda_j^2 f_j A^{l-2} = \cdots = \lambda_j^l f_j \tag{4.22}$$

Equation 4.22 and the last equation of equation 4.14 are substituted into equation 4.13. The Δp_{n+1} term of simulataneous equations 4.13 is then solved. The general equation for Δp_{n+1} is:

$$\Delta p_{n+1} = \sum_{p=1}^{u} \frac{\lambda_p^u f_p \Delta Z_n}{\left(\prod_{q=1,q\neq p}^{u}(\lambda_p - \lambda_q)\right)(\lambda_p - 1) f_p B} \tag{4.23}$$

The Δp_{n+1} equation for systems with one unstable manifold (u=1) is:

$$\Delta p_{n+1} = \frac{\lambda_1 f_1 \Delta Z_n}{(\lambda_1 - 1) f_1 B} \tag{4.24}$$

while the equation for two unstable manifolds (u=2) is:

$$\Delta p_{n+1} = \frac{\lambda_1^2 f_1 \Delta Z_n}{(\lambda_1 - \lambda_2)(\lambda_1 - 1) f_1 B} - \frac{\lambda_2^2 f_2 \Delta Z_n}{(\lambda_1 - \lambda_2)(\lambda_2 - 1) f_2 B} \tag{4.25}$$

In summary, the OGY method controls the systems through the perturbation of the control parameter and only works when the current trajectory point is in close proximity to the target. This is due to the constraint which results from linearising the Poincaré map. The control parameter in systems is then perturbed according to the following piecewise function [52]:

$$p_{n+1} = \begin{cases} p_n + \Delta p_{n+1}, & r_n < \varepsilon_r \\ p_n, & r_n > \varepsilon_r \end{cases} \tag{4.26}$$

where the symbols r_n and ε_r are distance and tolerance value respectively.

The purposes of the two symbols are explained in later sections. Note that it takes u Poincaré Map iterations before the state point falls on the stable manifold. After control for u iterations, the perturbation parameter is switched back to the original. It takes a few more iterations before the state point converges at the fixed point. The perturbation force remains constant between the Poincaré iterations. For stability maintenance in a situation with noise or disturbances, continuous control is required.

4.2.1 Summary of the OGY method

The OGY method begins with the selection of an UPO as target. The position of its saddle fixed point and its stable and unstable directions are determined. Figure 4-6 [53] describes the process. Figure 4-6 (a) shows when the state point is within proximity to the fixed point. Without control, the state point will move toward the fixed point along the stable manifold and then away from the fixed point along the unstable manifold. When the control parameter is perturbed, the fixed point changes its location as shown in figure 4-6 (b). The manifolds of the new fixed point will then drive the state point toward the direction of the original stable manifold. Once the control parameter is switched back to the nominal value, the system state point will be on the stable manifold. It converges toward the fixed point along the stable direction (as shown in figure 4-6 (c). In order to maintain the state point on the fixed point, perturbation control is applied continuously.

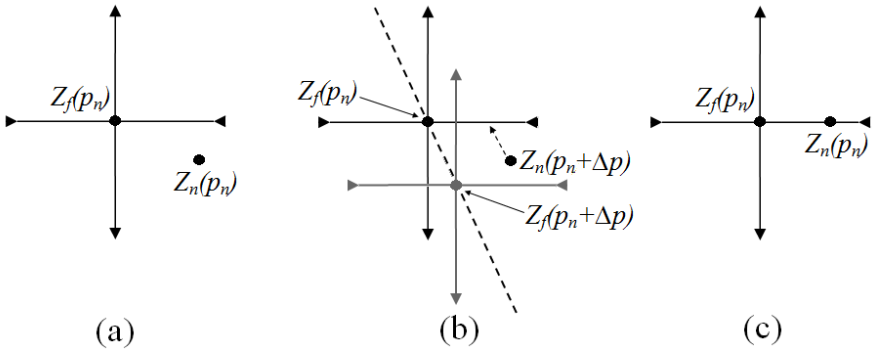

Figure 4-6: Perturbation control through the OGY method - a) State point within proximity of the fixed point of UPO, b) Fixed point and manifold shift due to control parameter perturbation, c) State point on the stable manifold

4.3 Chaos Control Procedures

Chaos control of the biped model was carried out in a number of procedural steps. The first step is to determine the biped's stability. If the biped is capable of walking a hundred consecutive steps, then it is categorised as stable. Another reason for the large number of steps is that the typical waiting time for a system flow to get close to the target point, is long. The biped may need to walk large number of steps before control conditions are met. The calculation of the Lyapunov Exponent determines whether the system is chaotic or not since chaos control can only occur in chaotic systems. All new method terminologies presented in this paragraph are explained in later sections. The OGY method involves the control of state points near UPO fixed points. The selection of a Poincaré Section is used to focus on one fixed UPO point since all the trajectory points of an UPO are fixed points. Converging on one UPO fixed point leads to convergence of the entire orbit. The closed return method was used for the detection of UPOs on Poincaré's Section as well as the approximation of UPO fixed point coordinates. The precise fixed point coordinates were determined using the linearisation of the Poincaré Map method using results from the closed return method as a starting point. The linearization of the Poincaré Map method was also used for the calculation of the linearised Poincaré Map of equation 4.8. From equation 4.8, the control parameter equation 4.23 was determined. Chaos control is applicable if all the control method conditions and linearisation limitations are met. The steps in the above procedure are summarised in the following list:

Step 1: Stability check of the system requires continuous walking of at least 100 steps.

Step 2: The system is determined chaotic or non-chaotic through the calculation of the Lyapunov Exponent.

Step 3: Selection of a suitable Poincaré Section.

Step 4: If the system is chaotic, the closed return method is used to detect and approximate period p UPO fixed point(s).

Step 5: Linearisation of the Poincaré Map method is used to determine the precise fixed point coordinate and to calculate the linearised Poincaré Map equation.

Step 6: The control parameter value is determined using the linearised Poincaré Map Equation.

Step 7: Perturbation control is applicable if the control conditions and linearisation limitations are met.

4.3.1 Lyapunov Exponent

In order to benefit from the advantages of chaos control, the system must be chaotic by nature or have induced chaos. How does one determine whether the behaviour of the system is chaotic or stochastic? Stochastic systems are non-deterministic systems with outcome states which depend on calculable dynamics and random influence factors (unknown factors). Lyapunov exponents are used to distinguish whether a system is chaotic or stochastic.

A Lyapunov exponent (λ) measures the divergence or convergence of initially nearby trajectories [9]. Note that the symbol of the Lyapunov exponent has no subscript (not to be confused with eigenvalues λ_i). By determining the sign of the Lyapunov exponents, the diverging or converging characteristics of a flow is determined. Lyapunov exponents are categorised in table 4-2 [9].

Table 4-2: Categories of the Lyapunov exponent

Lyapunov Exponent	System Behaviour
$\lambda>0$	Chaotic diverging
$\lambda=0$	Periodic, neither converging or diverging
$\lambda<0$	Stable periodic converging

For $\lambda>0$, orbits are chaotic and nearby trajectories diverge before reaching the boundary of a chaotic system. Since the chaotic subspace is bounded, trajectories cannot continue to diverge through time. For $\lambda=0$, it corresponds to periodic orbits which neither diverge nor converge; the distance between trajectories remain constant. Period-doubling occurs at $\lambda=0$. For $\lambda<0$, nearby trajectories converge to a periodic orbit. The behaviour of a system (chaotic or stochastic) is determined from Lyapunov exponents. Lyapunov exponent can be calculated using equation 4.27 to equation 4.29. The distance between two initially nearby trajectories is determined in equation 4.27 [9]:

$$d_n = \left| f^{(n)}(Z_0 + \varepsilon_0) - f^{(n)}(Z_0) \right| \tag{4.27}$$

The symbol d_n is the absolute distance between two nearby points on the Poincaré Map after n^{th} iterations. The symbol ε_0 is the initial distance vector between the two points.

It is assumed that the rate of separation has an exponential growth in equation 4.28 [9]:

$$d_n = d_0 e^{\lambda n} \tag{4.28}$$

where d_0 is the initial distance between the two points and λ the Lyapunov exponent after the n^{th} iteration.

By manipulating equation 4.28, the Lyapunov exponent is determined, resulting:

$$\lambda = \frac{1}{n} \ln \frac{d_n}{d_0} \tag{4.29}$$

The number of iterations (n) allowed is limited by the boundary of the chaotic phase space and the magnitude of d_0 [43]. The smaller the d_0, the more iteration can be conducted before reaching the

boundary limit. *n* is also influenced by the rate of separation λ itself. Figure 4-7 (a) shows the saturation of the $ln\ d_n$ due to over-iteration. The Lyapunov exponent decreases after the boundary number is reached (equation 4.29) as $ln\ d_n$ and $ln\ d_0$ remain constant while *n* continues to increase. The Lyapunov exponent can only be accurately determined when the iteration number is not exceeded.

Equation 4.29 calculates the Lyapunov exponent based on the iteration number and separation distances d_0 and d_n. A more accurate approach to calculating the Lyapunov exponent is to fully utilise all iteration points. Lyapunov exponent is determined using the slope of a straight line and the straight line relationship is obtained from equation 4.29, resulting:

$$\begin{aligned} d_n &= d_0 e^{\lambda n} \\ \therefore \ln d_n &= \ln d_0 e^{\lambda n} \\ &= \ln d_0 + \ln e^{\lambda n} \\ &= \lambda n + \ln d_0 \end{aligned} \tag{4.30}$$

A straight line has a standard equation form of $y = mx + c$. Based on equation 4.30, the symbols *y* and *x* are replaced by $\ln d_n$ and *n* respectively. The symbols $m=\lambda$ is the gradient of the slope and $c= \ln d_0$ is the *y*-intercept. The Lyapunov exponent is the gradient of the slope determined from the plot of $ln\ d_n$ versus *n* as shown in figure 4-7 (b) [9]. The unit of λ depends on d_n which has combined units of rad and rad/s. Due to this muddled combination of units in calculations, all λ values presented in this dissertation are assumed unitless but consistent. The starting condition in all λ calculations is the immediate moment after heel strike transition.

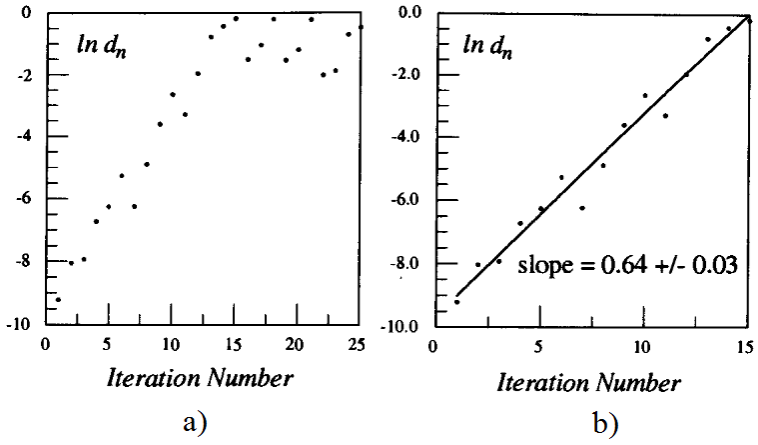

Figure 4-7: Plot of $ln\ d_n$ versus iteration number *n* - a) Iteration exceeding saturation, b) Iteration within saturation

Equation 4.29 and equation 4.30 calculate the Lyapunov exponent of a certain initial condition. The average Lyapunov exponent for a system is determined by averaging the Lyapunov exponents of different initial conditions and is determined as shown in equation 4.31 [9]:

$$\lambda_{average} = \frac{1}{N}\sum_{i=1}^{N} \lambda(z_i) \tag{4.31}$$

where the symbol *N* denotes the number of distributed points.

The term $\lambda(Z_i)$ is the initial condition-dependant Lyapunov exponent. Typically, evenly distributed initial conditions of the chaotic BOA are used for calculating the Lyapunov exponent.

Equation 4.28 assumes that the rate of separation has exponential growth. This assumption can be verified using figure 4-7 (a). If the plotted data points fall approximately on a straight line, then the assumption is valid. The reasons for the plotted points not lying exactly on a straight line can be the following:

- In theoretical simulations, round-off errors accumulate and thus the system behaviour may deviate from the actual. All experimental results are embedded with some sort of noise which can either be electronic noise or minor external influence forces. For electronic noise, the precision of the data is affected but not the system behaviour trend.
- For minor external influences, the system behaviour is influenced, and the extent depends on the frequency and magnitude of the influence. If the plotted points do not form an approximate straight line, then the determined Lyapunov exponent is meaningless.

In summary, if a linear relationship is obtained for the plot of $\ln d_n$ versus the iteration number *n* and the Lyapunov exponent is positive, then the system is chaotic. If the plot of $\ln d_n$ versus the iteration number *n* forms irregular trend patterns (neither exponential nor polynomial relations), then the system is most likely stochastic or noise-influenced.

Figure 4-8 [9] shows the long-term states of a system as a function of its control parameter. Figure 4-8 (a) shows the rise and fall of the Lyapunov exponent round about the zero line. Interpretation of the Lyapunov exponent's magnitude is categorised in table 4-2. Figure 4-8 (b) shows the bifurcation diagram of the same system. Both plots of figure 4-8 show the same categorisation for long-term states. For every zero crossing of the Lyapunov exponent, it corresponds to period doublings of the bifurcation diagram. Periodic and chaotic orbits are also similarly categorised. The difference between the two plots is illustrated in figure 4-8 (a) which shows the rate of separation and figure 4-8 (b) which shows the order of periods. Both plots can be used to determine whether a system is chaotic or not.

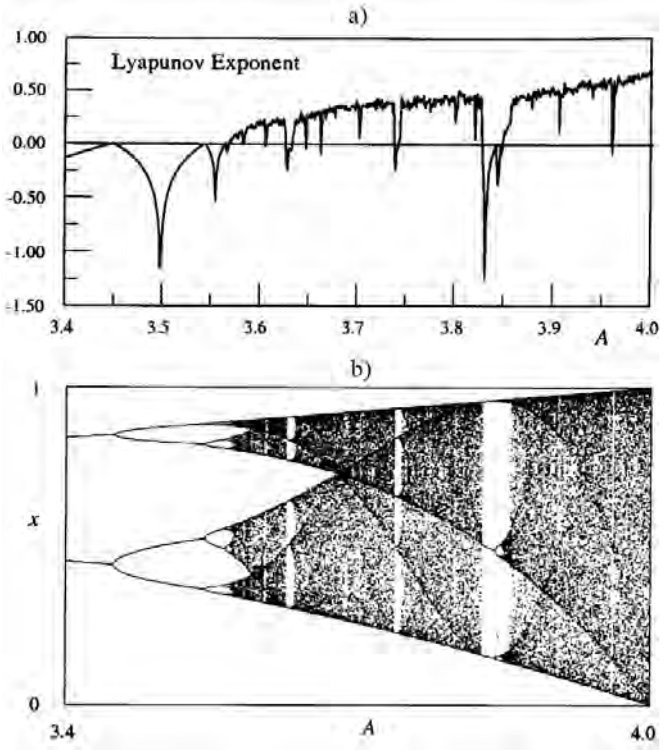

Figure 4-8: Graphic representations of system behaviour as a function of control parameter - a) Lyapunov exponent as a function of control parameter, b) Bifurcation diagram of the same system

4.3.2 Poincaré Section

Before the Poincaré Map is determined, a Poincaré Section needs to be selected. A Poincaré Section is chosen based on the time or occurrence of events. For time-based sections, the dynamics are analysed at different times separated by equal periods. Time-based sections are mostly applied to smooth continuous flows. For the sagittal model, it is more appropriate to select an event-based Poincaré Section. Since the periods of the walking steps are not constant, having sections with the same time interval spacing will lead to a Poincaré Map that shows the comparison between the different walking stages of a step. The Poincaré Section is best chosen at events such as the knee or heel strike transitions for same-phase comparisons. There is also a difference between choosing the moment before or after strike transitions as phase space dimensions differ between the two. The respective phase spaces before and after transitions are: 6 and 4 for knee strike, and 4 and 3 for heel strike. Immediate moments after transition was selected as these have the least dimensions. The dimension reducing advantage of using a Poincaré Map does not apply as Poincaré Sections cannot be modelled using autonomous equations. This means that the modelled flows have its usual phase dimensions. Two Poincaré Sections were used in one walking step for reasons that will be discussed in section 4.3.5. The procedures involved in determining the Poincaré Map is described after the 4.3.3 UPO detection section.

4.3.3 Unstable Periodic Orbits Detection

UPO can be found using different methods namely close return [43], linearisation of the Poincaré Map [55] and the efficient iterative method [56]. Close return and linearisation of the Poincaré Map were the methods used for UPO detection. Close return is a faster method and is used to approximate the UPO

state condition or coordinate. The UPO coordinate was then used in the linearisation of the Poincaré Map method as the starting condition. Linearisation of the Poincaré Map method is a more accurate approach but it takes longer to simulate.

4.3.3.1 Close Return Method

This method simply measures the distance between two successive recurrence points on the Poincaré Section as shown in equation 4.32 [43]:

$$D_n = \|Z_{n+1} - Z_n\| \tag{4.32}$$

For fixed points, $Z_{n+1} = Z_n = Z_f$ and $D_n = 0$. In simulations and practice, the following constraint is used:

$$D_n < \varepsilon_d \tag{4.33}$$

where ε_d is the tolerance value of D_n.

From a set of recurrence points, the smallest D_n of the set was determined. The UPO coordinate is then simply the Z_n coordinate used in the smallest D_n calculation. If the smallest D_n is larger than ε_d, then UPO exists in a higher period. This means that D_n is calculated using recurrence points of larger return number.

4.3.3.2 Linearisation of the Poincaré Map

This method determines the linearised Poincaré Map of a region about a point using data points [55]. The UPO of the linearised region is determined using the linearised mapping matrices. First a set of Poincaré Section recurrence points are obtained for a constant control parameter. This data set does not have to be from one initial condition. Results from different initial conditions can also be used. The coordinate set is filtered according to constraint 4.34 and constraint 4.35:

$$r_n = \|Z_n - Z_f\| \tag{4.34}$$

$$r_n < \varepsilon_r \tag{4.35}$$

where the symbol ε_r is the tolerance value of r_n.

The UPO coordinate determined from the closed return method is used as the fixed point Z_f in equation 4.34. The filtering process eliminates all Z_n intersection points not within the radius r_n boundary of the fixed point (as shown in figure 4-9 [5]). Equation 4.32 is also applied to ensure that all consecutive points satisfy condition 4.33. The acceptable points Z_n are recorded with their recurrence points Z_{n+1} in pairs. The paired data are then used to determine the linearised dynamics around the fixed point. The linearised relationship is a good approximation if the used data points are close to the fixed point, resulting that the smaller the r_n radius value, the more accurate the linearisation.

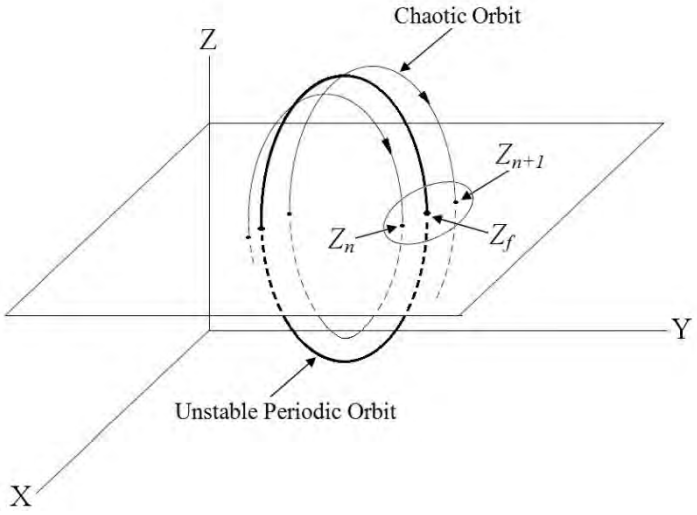

Figure 4-9: Obtaining sets of recurrence data points near the fixed point

The linearised form of Poincaré Map (equation 4.6) about a fixed point is [55]:

$$Z_{n+1} = CZ_n + D \tag{4.36}$$

where C is the Jacobian matrix of the linearisation and D the constant vector.

The linearised map's accuracy (equation 4.36) decreases, the further a points is away from the fixed point. Therefore, mapping points Z_n have to be in close proximity (within r_n radius boundary) with the fixed point in order for the algorithm to be effective. Equation 4.36 is determined from previous data pairs as indicated in equation 4.37:

$$\begin{aligned} Z_{n+1}^1 &= CZ_n^1 + D \\ Z_{n+1}^2 &= CZ_n^2 + D \\ &\vdots \\ Z_{n+1}^m &= CZ_n^m + D \end{aligned} \tag{4.37}$$

where the symbol *m* denotes the number of data pairs.

Equations 4.37 are arranged into the following matrix form:

$$[Z_{n+1}^1 \cdots Z_{n+1}^m] = [C \quad D]\begin{bmatrix} Z_n^1 & \cdots & Z_n^m \\ 1 & \cdots & 1 \end{bmatrix} \tag{4.38}$$

Let $E = [Z_{n+1}^1 \cdots Z_{n+1}^m]$ and $F = \begin{bmatrix} Z_n^1 & \cdots & Z_n^m \\ 1 & \cdots & 1 \end{bmatrix}$ then:

$$[C \quad D] = EF^T(FF^T)^{-1} \tag{4.39}$$

where the symbol T denotes transpose.

Matrix $[C \quad D]$ has row rank dimension of Z_n and column rank dimension of row rank plus one. The minimum number of data pairs used must have at least the number of elements in vector Z_n. Linearisation accuracy increases with more data pairs used.

Large numbers of data pairs are used to avoid singularities when calculating the inverses in equation 4.39. Due to the ergodic characteristic of chaotic systems, all parts of the system will be visited by any trajectory after a sufficiently long time. This ensures that there will be sufficient data pairs if the iteration number is large. The fixed point is calculated again for better accuracy. The fixed point in the linearised region of equation 4.36 is determined as:

$$Z_f \approx (I - C)^{-1} D \tag{4.40}$$

4.3.4 Approximation of the Linearisation Poincaré Function

The linearised form of the Poincaré Map (equation 4.8) about a fixed point is approximated. Matrix *A* of equation 4.8 is approximated as equation 4.41 [5]:

$$A \sim C \tag{4.41}$$

where the matrix *C* is determined from equation 4.39.

Matrix *B* of equation 4.8 is determined using two sets of data pairs. Both sets have the same parameters except for the control parameter. The control parameter changes from $p_n \rightarrow p_n + \Delta p$ where Δp is small. Matrix *B* is approximated as equation 4.42 [5]:

$$B \sim \left(Z_f(p_n + \Delta p) - Z_f(p_n) \right) / \Delta p \tag{4.42}$$

Equation 4.42 is the manipulated form of equation 4.4. The selection of perturbation difference Δp in equation 4.42 is dependent on the magnitude of control parameter p_n. Due to approximations, control algorithm is only effective for $p_n > \Delta p$. If large Δp is selected, approximation will also be inaccurate as the system dynamics is greatly altered by Δp. For effective approximation and control, Δp must be larger and in similar magnitude with p_n.

The linearised Poincaré Map of period 1 UPO was determined using equation 4.32 to equation 4.42. For period 2 Poincaré Map, state points and their second recurrence points are used. This means the term Z_{n+1} of equation 4.32 and equation 4.36 to equation 4.39 were replaced by Z_{n+2}. The terms Z_{n+1} and p_{n+1} of equation 4.8 was also replaced by Z_{n+2} and p_{n+2} respectively. In general, for a period p UPO ($p \in$ counting numbers), equation 4.32 to equation 4.42 are used for the determining of the linearised Poincaré Map with replacement equations as:

$$D_n = \left\| Z_{n+p} - Z_n \right\| \tag{4.43}$$

Equation 4.43 replaces equation 4.32 and equation 4.44 to equation 4.47 replaces equation 4.36 to equation 4.39 respectively.

$$Z_{n+p} = CZ_n + D \tag{4.44}$$

$$\begin{aligned} Z^1_{n+p} &= CZ^1_n + D \\ Z^2_{n+p} &= CZ^2_n + D \\ &\vdots \end{aligned} \tag{4.45}$$

$$Z_{n+p}^{m} = CZ_{n}^{m} + D$$

$$\left[Z_{n+p}^{1} \cdots Z_{n+p}^{m}\right] = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} Z_{n}^{1} & \cdots & Z_{n}^{m} \\ 1 & \cdots & 1 \end{bmatrix} \tag{4.46}$$

$$\begin{bmatrix} C & D \end{bmatrix} = EF^{T}(FF^{T})^{-1} \tag{4.47}$$

where matrices $E = \left[Z_{n+p}^{1} \cdots Z_{n+p}^{m}\right]$ and $F = \begin{bmatrix} Z_{n}^{1} & \cdots & Z_{n}^{m} \\ 1 & \cdots & 1 \end{bmatrix}$ are used in equation 4.47.

Equation 4.8 is then replaced by the equation:

$$Z_{n+p} - Z_{f}(p_{n}) = A\left(Z_{n} - Z_{f}(p_{n}) - B\left(p_{n+p} - p_{n}\right)\right) + B(p_{n+p} - p_{n}) \tag{4.48}$$

The linearised Poincaré Map equation above approximates mapping of points within the region of radius ε_{r}. The closer the state point is to the fixed point, the more accurate the mapping and vice versa.

4.3.5 Implementing chaos control in the sagittal biped model dynamics

Chaos control was applied after strike transitions in a walking step. Phase space dimension changes according to different stages of walking which results in a change in the number of unstable manifolds. System flow is non-smooth as a result of strike transitions. The Poincaré Mapping method does not model the dynamic effects between recurrence points; it only predicts outcome states based on inputs. The non-smooth dynamics of strike transitions that occur between mappings are not accounted for. Strike transition may at times affect the outcome states beyond the approximation of the Poincaré Map. Applying perturbation control once only in a walking step is insufficient due to unstable manifold number variation and the effects of strike transitions.

There are four distinct events (four possible Poincaré Sections) for which control can be applied. They are the immediate moments before and after knee and heel strike transitions. The respective phase spaces before and after these transitions are: 6 and 4 for knee strike, and 4 and 3 for heel strike. Events before the transitions cannot be used due to unsolvable matrices. All dependant variable vectors (equation 3.27 and equation 3.35), at moments just before strike transitions, have linearly dependent generalised coordinates which results in the matrix (EF^{T}) of equation 4.39 and equation 4.47 having linearly dependent columns. Matrices with linear dependent columns do not have inverses which results in equation 4.39 and equation 4.47 having no solutions. The dependant variable vectors, at immediate moments after strike transitions, have phase space independency which means that their matrices are solvable and that solutions exist. Chaos control was applied twice within a step (after knee and heel strikes) which means that the controlled parameter may change twice in magnitude during a step. Note that the control parameter remains constant for the duration of the locked or unlocked knee swing phase.

Due to the presence of transition events in the sagittal model, chaos control can be applied twice in one step instead of the normal once per cycle. This additional perturbation control helps to stabilise the system because the delay time between disturbance and reaction is reduced. The procedure used to control the sagittal biped model is indicated by the detection of a period *p* UPO which is conducted using closed return and linearisation of the Poincaré Map method. The selection of period *p* UPO was based on the heel strike Poincaré Map. The UPO at heel strike is simpler to determine as it has a smaller phase

dimension than UPO at knee strike. The immediate moment after heel strike has a phase dimension of 3; it changes to 6 during the unlocked knee swing phase. If heel strike occurs periodically, knee strike is also periodic. From the determined knee and heel strike UPO, the respective linearised Poincaré Map functions were determined. The approximated map functions (from equation 4.48) are:

Linearised Poincaré Map at instant after knee strike:

$$Z_{n+p}^{k} + Z_{f}^{k}(p_{n}^{k}) = A^{k}\left(Z_{n}^{k} - Z_{f}^{k}(p_{n}^{k}) - B^{k}\left(p_{n+p}^{k} - p_{n}^{k}\right)\right) + B^{k}\left(p_{n+p}^{k} - p_{n}^{k}\right) \tag{4.49}$$

Linearised Poincaré Map at instant after heel strike:

$$Z_{n+p}^{h} + Z_{f}^{h}(p_{n}^{h}) = A^{h}\left(Z_{n}^{h} - Z_{f}^{h}(p_{n}^{h}) - B^{h}\left(p_{n+p}^{h} - p_{n}^{h}\right)\right) + B^{h}\left(p_{n+p}^{h} - p_{n}^{h}\right) \tag{4.50}$$

where the superscript *k* and *h* denotes knee and heel strike respectively.

The generalised coordinates in the following state condition vectors are: $Z^{h} = [q_1, \dot{q_1}, \dot{q_2}]^{T}$ and $Z^{k} = [q_1, q_2, \dot{q_1}, \dot{q_2}]^{T}$. The perturbation or control parameter (p_n) of the biped model is the hip joint torque u_2. The objective was to control u_2 such that the system orbit crosses both the knee (Z_f^k) and heel (Z_f^h) fixed points. Perturbation control was only applied when system flow was close to the fixed points. The matrices of equation 4.49 and equation 4.50, with equation 4.23, were used to calculate the perturbation difference of equation 4.51 and equation 4.55 respectively. Control was applied at immediate moments after strike transitions.

4.3.5.1 Locked Knee Swing Phase Control

For locked knee swing phase control, control was applied according to the following conditions:

$$p_{n+1}^{k} = \begin{cases} p_{n}^{k} + \Delta p_{n+1}^{k}, & r_{n}^{k} < \varepsilon_{rf}^{k}\varepsilon_{r}^{k} \\ p_{n}^{k}, & r_{n}^{k} > \varepsilon_{rf}^{k}\varepsilon_{r}^{k} \end{cases} \tag{4.51}$$

where the symbol $\varepsilon_{rf}^{k} \in (0, 1]$ is a constant fraction.

Even though the linearised Poincaré Map region covers a radius of ε_r^k, successful control only occurs when r_n^k is a fraction of ε_r^k. The accuracy of control is also dependent on the magnitude of the perturbation force. If the perturbation is too large, the linearisation approximation becomes inaccurate. In addition, the following perturbation magnitude condition also applies:

$$\begin{gathered} \text{If } \Delta p_{n+1}^{k} > \Delta p_{max}^{k}, \\ \Delta p_{n+1}^{k} = 0 \end{gathered} \tag{4.52}$$

where the symbol Δp_{max}^{k} denotes the maximum perturbation difference for locked knee swing phase control.

The control parameter u_2 for locked knee swing phase is as follows:

$$u_2 = \Delta p_{n+1}^{k} + p_{n}^{k} \tag{4.53}$$

Equation 4.53 with u_1=0 is substituted into equation A.13 to compute the locked knee swing phase dynamics. Equation A.13 becomes:

$$\tau = \begin{bmatrix}\tau_1 \\ \tau_2\end{bmatrix} = \begin{bmatrix}-(\Delta p_{n+1}^k + p_n^k) \\ \Delta p_{n+1}^k + p_n^k\end{bmatrix} \tag{4.54}$$

4.3.5.2 Unlocked Knee Swing Phase Control

For unlocked knee swing phase control, control is applied according to the following conditions:

$$p_{n+1}^h = \begin{cases} p_n^h + \Delta p_{n+1}^h, & r_n^h < \varepsilon_{rf}^h \varepsilon_r^h \\ p_n^h, & r_n^h > \varepsilon_{rf}^h \varepsilon_r^h \end{cases} \tag{4.55}$$

where the symbol $\varepsilon_{rf}^h \in (0, 1]$ is a constant fraction.

Constant ε_{rf}^h serves the same purpose as ε_{rf}^k. Similar to condition 4.52, the following perturbation magnitude condition also applies:

$$\begin{gathered}\text{If } \Delta p_{n+1}^h < \Delta p_{max}^h, \\ \Delta p_{n+1}^k = 0\end{gathered} \tag{4.56}$$

where the symbol Δp_{max}^h denotes the maximum perturbation difference for unlocked knee swing phase control.

The control parameter u_2 for unlocked knee swing phase is:

$$u_2 = \Delta p_{n+1}^h + p_n^h \tag{4.57}$$

Equation 4.57 with u_1=0 and u_3=0 is substituted into equation A.4 to compute the unlocked knee swing phase dynamics. Equation A.4 becomes:

$$\tau = \begin{bmatrix}\tau_1 \\ \tau_2 \\ \tau_3\end{bmatrix} = \begin{bmatrix}-(\Delta p_{n+1}^h + p_n^h) \\ \Delta p_{n+1}^h + p_n^h \\ 0\end{bmatrix} \tag{4.58}$$

4.4 Chaotic and Periodic Orbits Simulations

Simulations were conducted on the passive walking of the biped model on constant angled slopes. It was determined that each slope angle has underlying dynamics which are either periodic or chaotic provided no actuations were involved. A bifurcation diagram and calculation of the Lyapunov exponent were used in the categorising periodic or chaotic dynamic process. Period 1 and 2 orbit system characteristics were analysed. Identifying the trend patterns of the periodic orbit was aimed at assisting understanding of chaotic orbits when they are periodically controlled. Chaos control of a period 1 UPO was conducted.

Model parameters used in the simulations were based on the design parameter range of table 3-3. All simulation parameters fall within the design range limit. The moments of inertia of the model segments were estimated using approximated biped geometry and masses. All simulations in this section use the

model parameter shown in table 4-3. All the simulation assumptions stated in Chapter 3 about modelling, also hold here.

It was assumed that the hip segment always rotates with an angle and angular velocity of q_1 and $\dot{q}_1$ respectively. This means that the constants *S1* and *S2* of equation 3.3 take on the values of 1 and 0 respectively. The reason for this assumption is that it produces a symmetrical or period 1 gait. If the hip segment was modelled as if it was coupled to one of the legs, the system dynamics vary between even and odd steps. This variation results in limping gaits of small and large steps. In reality coupling of the hip segment to one of the legs can be avoided if an upper body is present like in humans thus avoiding the alternating of system dynamics between walking steps. The alternating effect can also be reduced from the perspective of the mechanical design. Possible methods are to minimise the magnitude of the hip's moments of inertia or to split the hip into two segments where each segment is coupled to each leg.

Table 4-3: Simulation Model Parameters

Masses	[kg]
m_h	3
m_t	2.5
m_s	0.9
Moments of Inertia	[kg .mm^2]
I_h	1200
I_t	20000
I_s	15000
Lengths	[mm]
l_t	277
l_s	323
b_2	138.5
c_2	0
b_1	161.5
c_1	20
w_f	50
r_s	100

4.4.1 Differentiating Chaotic and Periodic Regions

A bifurcation diagram of the modelled system was generated first. Plotting of the Lyapunov exponents uses the converged states determined in the bifurcation diagram simulation as initial conditions. For accuracy, the Lyapunov exponent is determined by averaging the Lyapunov exponents of different initial conditions. Here only one initial condition is used per slope angle. The magnitude of the determined Lyapunov exponents may not be as accurate, but the sign of the values are the same for both methods. System behaviour (chaotic or periodic) can be identified from just the sign of the Lyapunov exponent when using table 4-2.

The bifurcation diagram of figure 4-10 and the Lyapunov exponent graph of figure 4-11 share the same categorisation of system behaviour as a function of the slope angle γ. Both figures show that orbits are

periodic for $\gamma \in [0.0189, 0.095]$ rad and mostly chaotic for $\gamma \in (0.095, 0.1]$ rad. For slope angles beside the abovementioned range, stable walking does not occur passively. Period-doubling first occurs at γ=0.08 rad in terms of increasing slope angle. It bifurcates into period 2 orbits for $\gamma \in [0.08, 0.09]$ rad. It continues to bifurcate into higher order periods (period 4, 8, 16 etc.) for $\gamma \in [0.09, 0.095]$ rad. The Lyapunov exponent graph in figure 4-11 confirms the abovementioned categorisation. The Lyapunov exponent graph pinpoints better results when it comes to the exact slope angles at which bifurcation occurs as illustrated at high period bifurcations. Figures 4-11(a) and (b) show that all bifurcation points occur at zero crossings of the graph. Figure 4-11 (b) shows that there are regions of periodic orbits within the chaotic region ($\gamma \in (0.095, 0.1]$ rad). This is illustrated by the negative Lyapunov exponents on the graph (figure 4-11 (b)). These periodic regions immersed within the chaotic regions are called periodic windows. One example occurs at γ =0.0974 rad. In figure 4-10 (b), it can be roughly seen that there appears to be a gap located at γ =0.0974 rad. This periodic window is confirmed by the γ =0.0974 rad negative Lyapunov exponent spike of figure 4-11 (b).

System dynamics at three slope angles were investigated. Period 1 and 2 orbits located at γ =0.06 rad and γ =0.09 rad respectively were analysed and analysis and control of chaos at γ =0.0985 rad were conducted. The three slope angles selected above do not lie close to period-doubling points or periodic windows. This ensures that the observed behaviour is not influenced by orbits of different characteristics as a result of rough slope angle interval resolution.

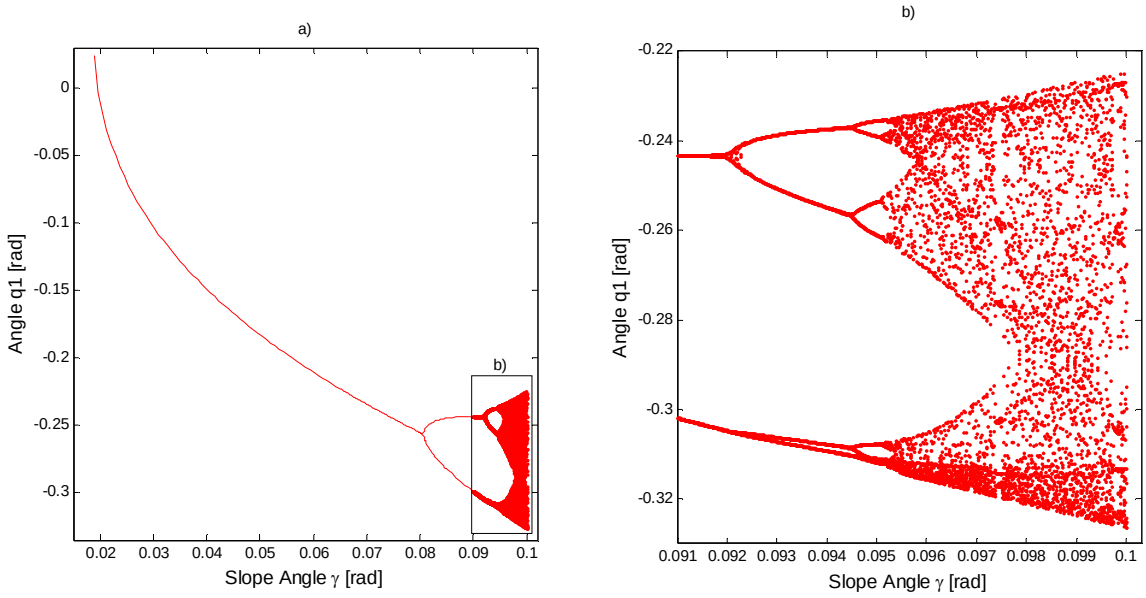

Figure 4-10: Bifurcation diagrams of the simulated biped model - a) Entire slope angle range of stable walking, b) Magnified view of chaotic regions

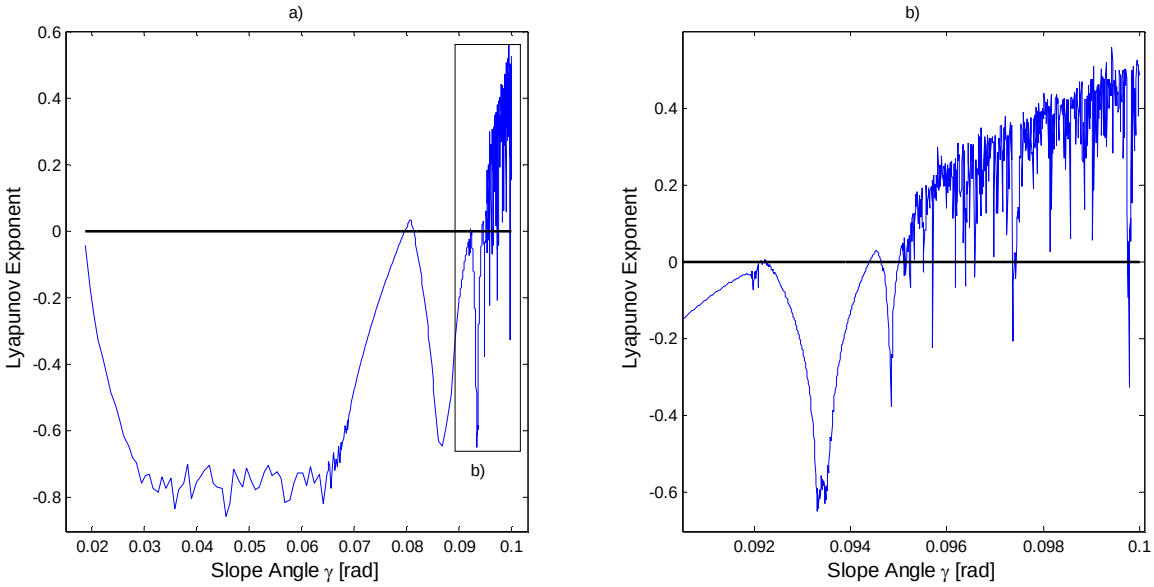

Figure 4-11: Lyapunov exponent graph of the simulated biped model - a) Entire slope angle range of stable walking c) Magnified view of chaotic regions

4.4.2 Limit Cycle

The limit cycle (period 1 orbit) of γ =0.06 rad were studied. Figure 4-12 shows the BOA on the heel strike Poincaré section of the limit cycle (shaded area). It is a section plane of the 3D initial condition phase space. The generalised coordinates q_1 and $\dot{q}_1$ were varied with $\dot{q}_2$=0.6$\dot{q}_1$. The full BOA includes the independent variation of $\dot{q}_2$, but for simplicity of presentation, the BOA region was reduced to a single plane. The entire section range of BOA is shown in figure 4-12 (a), where most initial condition combinations are unrealistic in the sense that they correspond to irregular configurations of the biped such as legs wide apart. The typical stable operation ranges are shown in figure 4-12 (b). The area of BOA in figure 4-12 (b) illustrates that stable walking is more likely to occur in typical operation ranges rather than in the range of figure 4-12 (a). Figure 4-12 (b) also shows that as the initial step length increases, the initial stance leg velocity ($\dot{q}_1$) also increases for stable walking. The magnitude of q_1 at starting position is proportional to initial step length (see figure 3-6). As stride length increases, a larger stance leg momentum is required because pitch motions are dependent on such momentum.

All trajectories with initial conditions within the BOA of the system will converge to one fixed point. The fixed points on the knee and heel Poincaré Sections are indicated in vector 4.59 and vector 4.60:

$$Z_f^k = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.1303 \quad -0.3630 \quad 0.8991 \quad -0.8045]^T \tag{4.59}$$

$$Z_f^h = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.2105, \quad 1.4589 \quad 0.4023]^T \tag{4.60}$$

where the units are rad and rad/s for q and $\dot{q}$ respectively, and this applies to the entire dissertation unless otherwise specified.

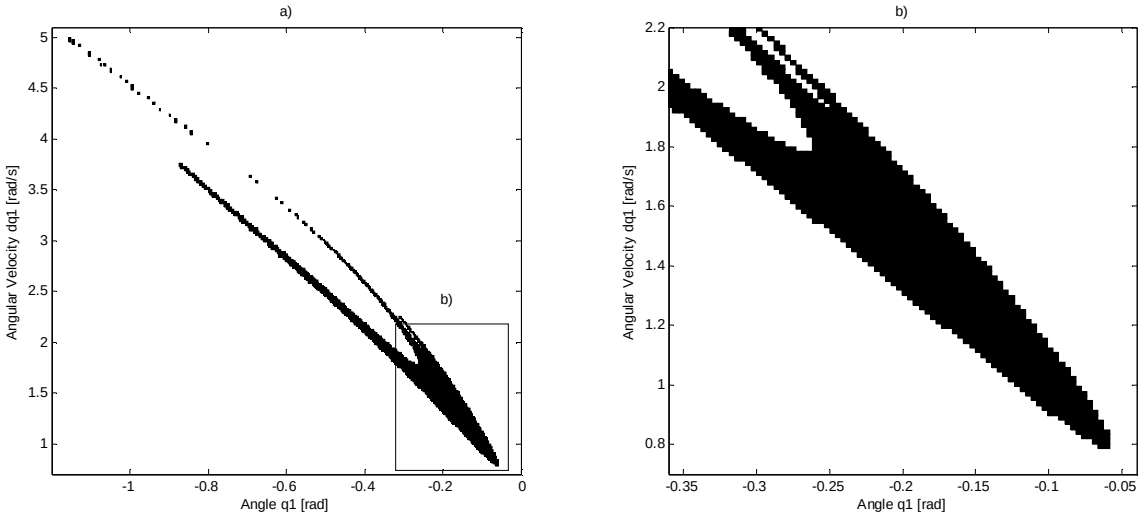

Figure 4-12: Limit cycle's basin of attraction - a) Entire BOA, b) Typical operational BOA

The above coordinates are rounded off to four decimal places. The most robust initial condition (on heel Poincaré Map) according to figure 4-12 (b) is located at $q_1 = -0.22\,rad$ and $\dot{q}_1 = 1.5\,rad/s$ (approximate centre of the BOA area). These parameters are closely located to the fixed point coordinates stated above. The eigenvalues at the fixed points are:

Z_f^k Eigenvalues: λ_1= 0.6160; λ_2=0; λ_3=0; λ_4=0

Z_f^h Eigenvalues: λ_1= 0.7001; λ_2=0; λ_3=0

All eigenvalues of fixed points are real and less than one. This implies that the fixed points are stable nodes, according to table 4-1. All state points within the BOA of the fixed point converge radially toward the fixed point. Figure 4-13 shows the plot of ($ln\ d_n$) versus iteration number which was used for the calculation of the Lyapunov exponent. The graph shows that saturation of d_n occurs from iteration number 15 onwards. The Lyapunov exponent is then the gradient of the straight line approximation of the first 15 iterated points (the gradient of the red line in figure 4-13). The Lyapunov exponent, using one initial condition, of the limit cycle, is λ=-1.0982 and corresponds to stable periodic orbit which validates the determined eigenvalues.

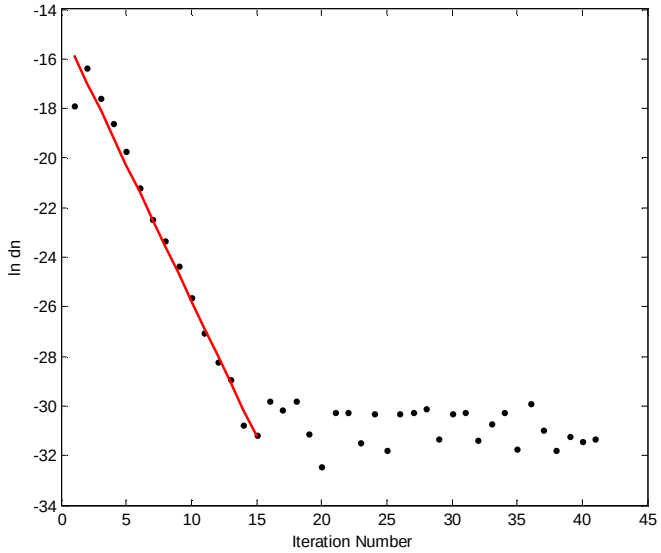

Figure 4-13: Plot of *ln* d_n versus iteration number.

Closed return plots using the knee and heel Poincaré Section are as shown in figure 4-14. The initial conditions of the plot are $[q_1, \dot{q_1}, \dot{q_2}] = [-0.23\,rad, 1.7\,rad/s,\ 1.02\,rad/s]$ (immediate moment after heel strike). The r_n distance between initial and fixed point condition is 0.6634. The closed return plots show that the distance between the consecutive recurrence points (D_n) converges after 7 iterations. The limit cycle closed return diagrams shows D_n approaching zero when system flow converges. Since the eigenvalues of knee and heel fixed points are similar, their rate of convergence is also similar according to the figure.

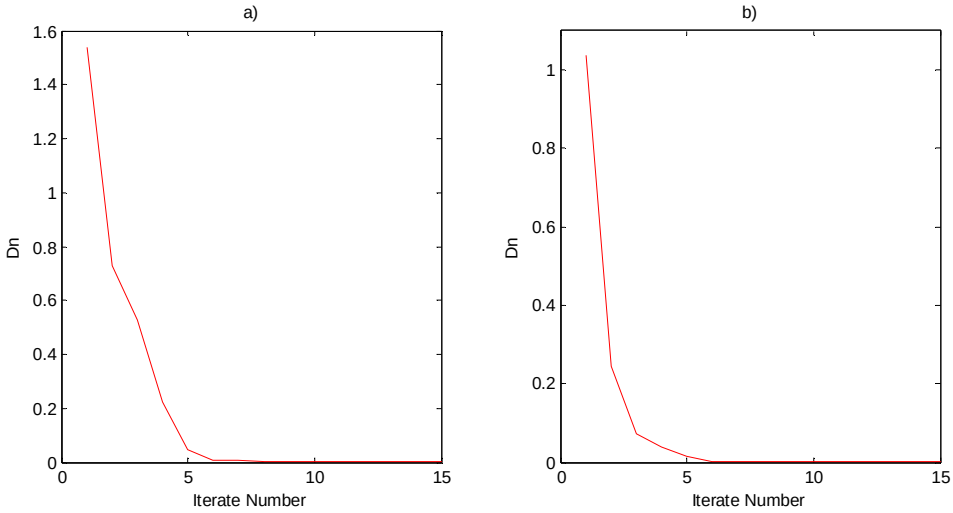

Figure 4-14: Period 1 Closed Return Plot - a) Knee plot, b) Heel plot

The phase portrait of figure 4-15 shows the progression of periodic generalised coordinates for one of the biped's leg. For limit cycles, left and right leg phase portraits are the same because each step is symmetrical. Phase portrait is the projection of a dynamical system's state trajectory onto a phase plane (a

2D plane) for easier visuals. The projection may be points or curves from which the system's flow trend can be easier observed. The plotted phase portrait illustrate an initial stance leg walking cycle (two walking steps). Walking starts at coordinates $[q_1, \dot{q}_1] = [-0.2105\, rad, 1.4589\, rad/s]$ along stance leg phase curve. It undergoes typical phases of walking as discussed in Chapter 3. At the beginning of the second step, the leg becomes swing leg and its trajectory is described using swing thigh and shank phase curves. The swing leg then combines at knee lock and the cycle repeats after swing leg heel strike. Note that for limit cycles, the starting and ending coordinates are the same point. The limit cycle has a walking step length and time of 0.369 m and 0.634 s respectively. This corresponds to a walking speed of 0.582 m/s.

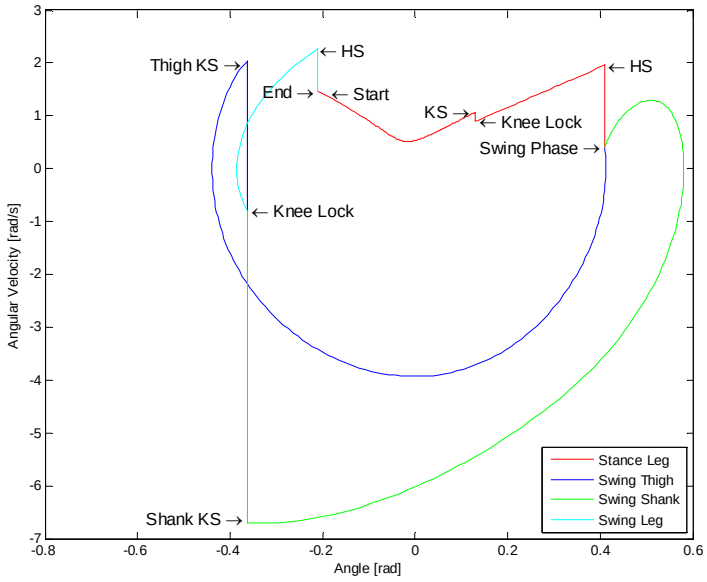

Figure 4-15: Period 1 phase portrait

It is possible to transform a periodic system to a chaotic one through actuation. Figure 4-16 shows the evolution of long-term state of walking as the hip joint torque u_2 is varied. For constant values of u_2 at γ=0.06 rad, the figure shows evolution of periodic to chaotic orbits as u_2 decreases. The variation of u_2 is capable of changing the system dynamics as shown in figure 4-16 but actuation torque is relatively high (when compared to control of chaotic system).

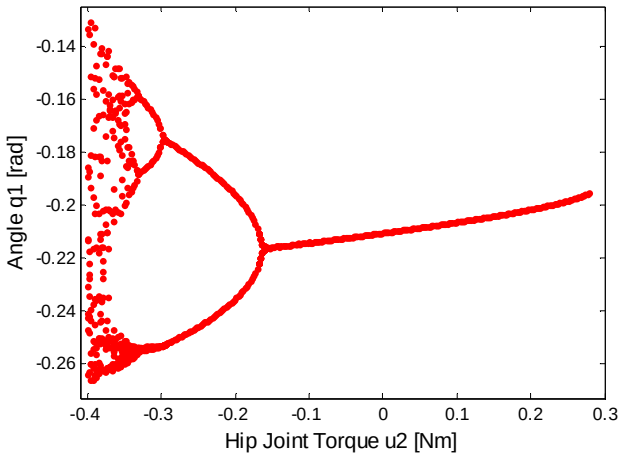

Figure 4-16: Bifurcation diagram of hip joint torque u_2 variation at γ =0.06 rad

4.4.3 Period 2 Orbit

A period 2 orbit of γ =0.09 rad were studied. Its BOA is shown in figure 4-17 (using the same simulation conditions as defined in the previous section). Figure 4-17 (a) shows the range of initial conditions for which walking is stable. The typical initial conditions during operations are shown in figure 4-17 (b). The initial stance leg velocity $\dot{q_1}$ range between steeper and gentle slopes, are similar (stable $\dot{q_1}$ of figure 4-17 (b) and figure 4-12 (b) because stable orbits of higher slope angles have larger initial inner leg angles (see figure 4-17 (b)) and larger step lengths.

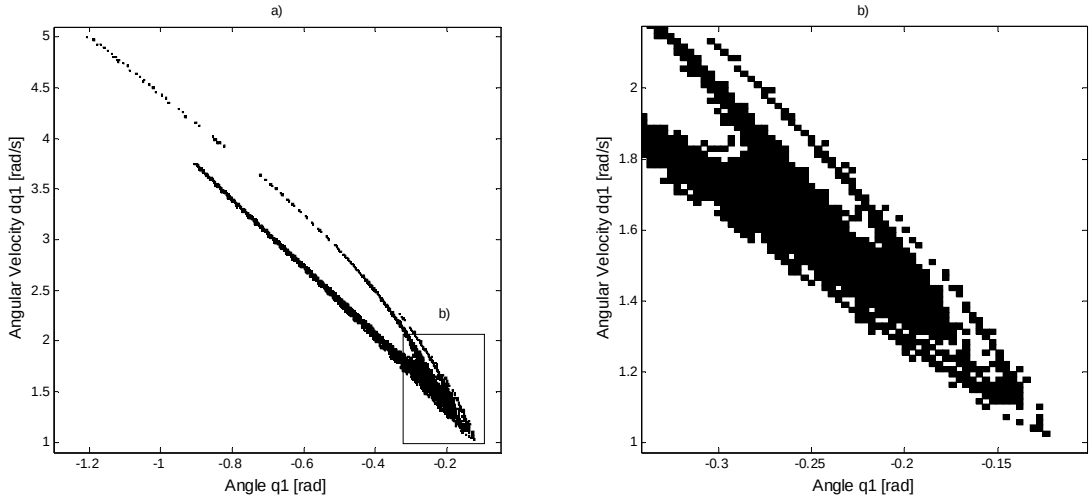

Figure 4-17: Period 2 orbit's basin of attraction - a) Entire BOA b) Typical operational BOA

All trajectories starting within the BOA of period 2 orbit converges to two fixed point pairs. The fixed point pairs and their eigenvalues are:

Fixed point pair 1:

$$Z_f^k = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.0446 \quad -0.5384 \quad 0.8570 \quad -1.3879]^T \tag{4.61}$$

$$Z_f^h = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.2432 \quad 1.5830 \quad 0.1128]^T \tag{4.62}$$

with

Z_f^k Eigenvalues: λ_1= 0.7470; λ_2=0; λ_3=0; λ_4=0

Z_f^h Eigenvalues: λ_1= 0.7205; λ_2=0; λ_3=0

Fixed point pair 2:

$$Z_f^k = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.1733 \quad -0.4088 \quad 1.2679 \quad -0.8341]^T \tag{4.63}$$

$$Z_f^h = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.2989 \quad 1.6447 \quad -0.1291]^T \tag{4.64}$$

with

Z_f^k Eigenvalues: λ_1= 0.7143; λ_2=0; λ_3=0; λ_4=0

Z_f^h Eigenvalues: λ_1= 0.7376; λ_2=0; λ_3=0

All eigenvalues of fixed points are real and less than one. Trajectories will converge to the first and second fixed point on the first and second Poincaré recurrence respectively. The Lyapunov exponent of the period 2 orbit is determined from figure 4-18. The Lyapunov exponent calculated using the first 50 iterations is λ=-0.2402 which indicates that the orbit is periodically stable.

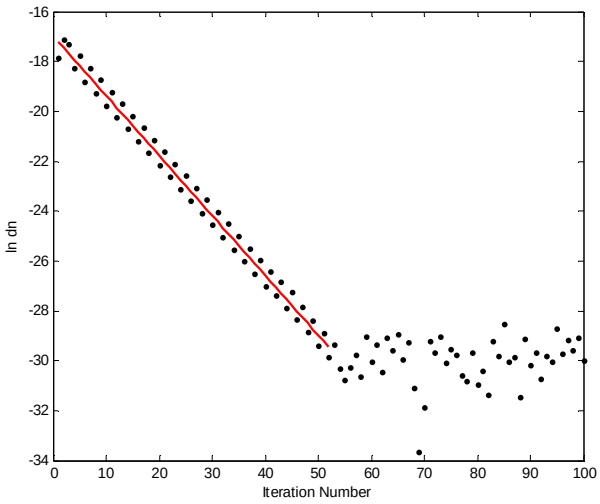

Figure 4-18: Plot of *ln* d_n versus iteration number

The knee and heel closed return plots are shown in figure 4-19. The initial conditions of the plot are $[q_1, \dot{q_1}, \dot{q_2}] = [-0.24\,rad, 1.6\,rad/s,\ 0.96\,rad/s]$ (immediate moment after heel strike). The r_n distance between initial and first fixed point condition is 0.8474. The plot on the figure indicates convergence after 15 iterations with knee and heel converged, D_n values of 0.7132 and 0.3609 respectively. Period 2 closed return diagrams have their D_n equal to a constant non-zero value at convergence. This value of D_n is the distance between the two fixed points on the Poincaré Map. For period 3 and greater periodic orbits, at flow convergence, the magnitude of D_n varies according to iteration number. For a period p orbit, at convergence, D_n varies between p constant values.

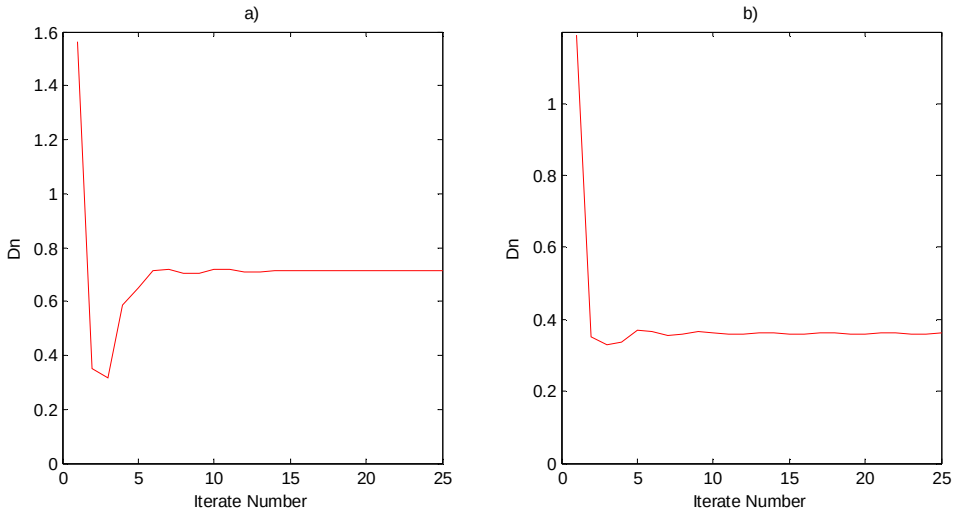

Figure 4-19: Period 2 Closed Return Plot - a) Knee plot, b) Heel plot

Figure 4-20 shows the plot of generalised coordinate as a function of time. The generalised coordinates plotted are states after convergence where the plots start with the heel condition of the first fixed point. The plot of figure 4-20 portrays the trajectories of a two step walking cycle where each leg takes a turn in being the stance or swing leg. The step length and time of the first step are 0.464 m and 0.598 s respectively (step speed of 0.776 m/s). The second step has a length and time of 0.412 m and 0.679 s respectively (step speed of 0.607 m/s). The above corresponds to a limping gait where one leg always takes smaller steps than the other. Figure 4-21 shows the phase portrait of the left and right leg of the biped model. The generalised coordinate of each leg in a cycle is plotted using figure 4-21. The right leg is the initial stance leg and left leg the corresponding swing leg.

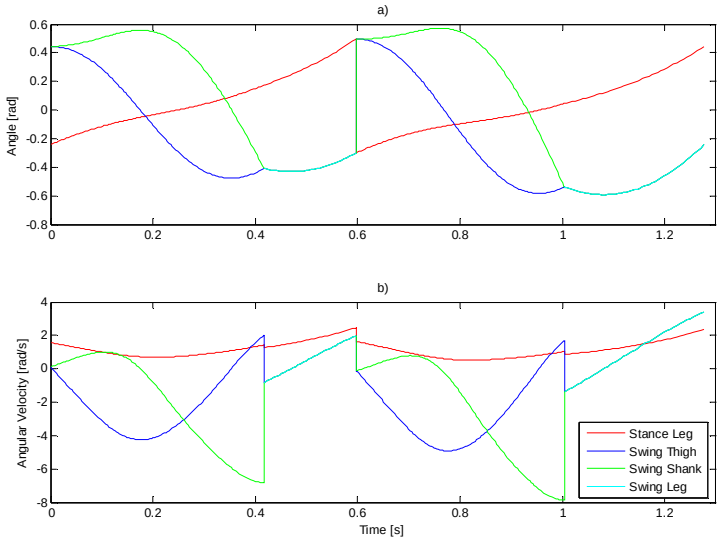

Figure 4-20: Plot of generalised coordinates versus time - a) Angular generalised coordinates, b) Angular velocity generalised coordinates

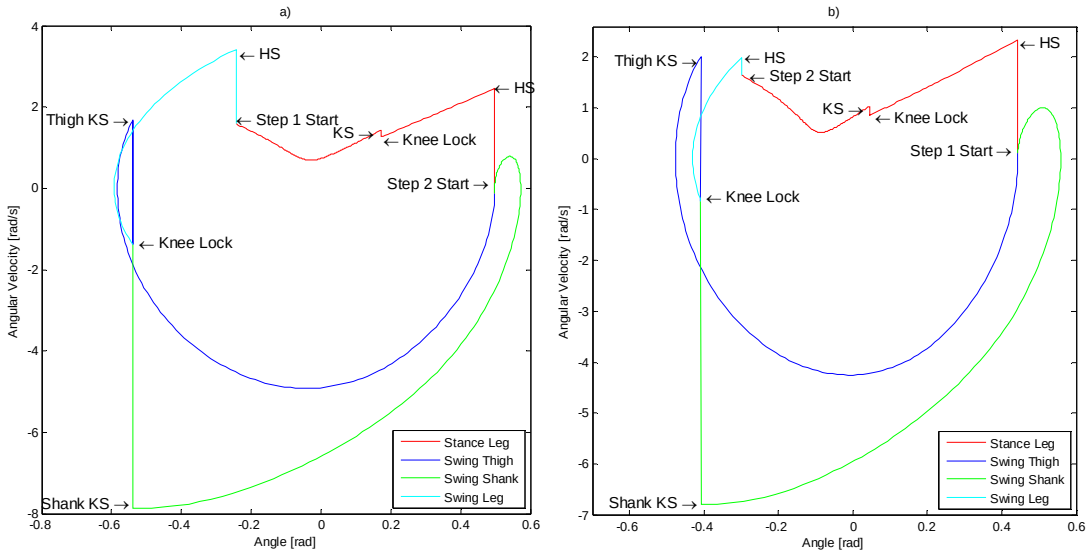

Figure 4-21: Period 2 phase portraits - a) Right leg (initial stance), b) Left leg (initial swing)

Like the limit cycle system, chaotic orbits are achievable in the period 2 system through actuation. Different system behaviours are possible depending on the value of the controlled parameter. Figure 4-22 shows that the chaotic region moves closer to the passive condition (u_2=0) when slope angle is decreased (compared with figure 4-16). Periodic orbits are dominant for u_2>0. Smaller actuation torque is required to generate chaotic walks (u_2<-0.54Nm).

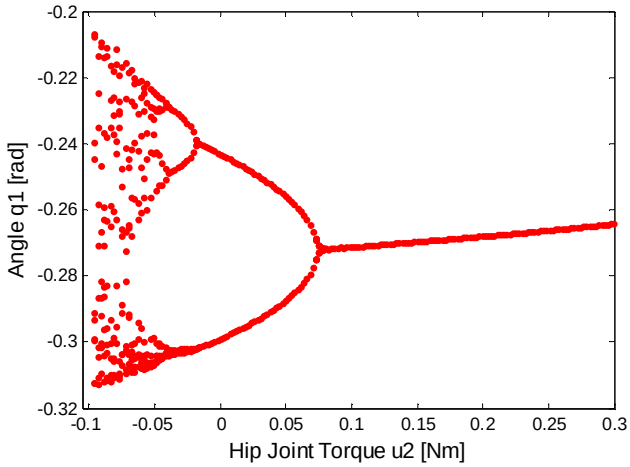

Figure 4-22: Bifurcation diagram of hip joint torque u_2 variation at γ =0.09 rad

4.4.4 Period 1 UPO Control

This section involves the control of a period 1 UPO at γ =0.0985 rad. The planed BOA for this slope angle is shown in figure 4-23 (with $\dot{q}_2$=0.6$\dot{q}_1$). Similar to the BOA figures in previous sections, the operational initial condition range is simply the shaded area in figure 4-23 (b). As slope angle increases, the area of BOA of typical operational ranges scatters. This results in initial conditions becoming sensitive to disturbances or deviations. The size of the BOA area also dictates whether the operating

conditions are robust enough for experimental walks. If the area is too small (limited q_1 and $\dot{q}_1$ range), then the chance of stable experimental walks decreases because the initial condition of the walks cannot be accurately controlled. The Lyapunov exponent determined from figure 4-24 is λ= 0.3870. It was determined using the gradient of the first 25 unsaturated points. The gradient is positive which confirms that the system is chaotic.

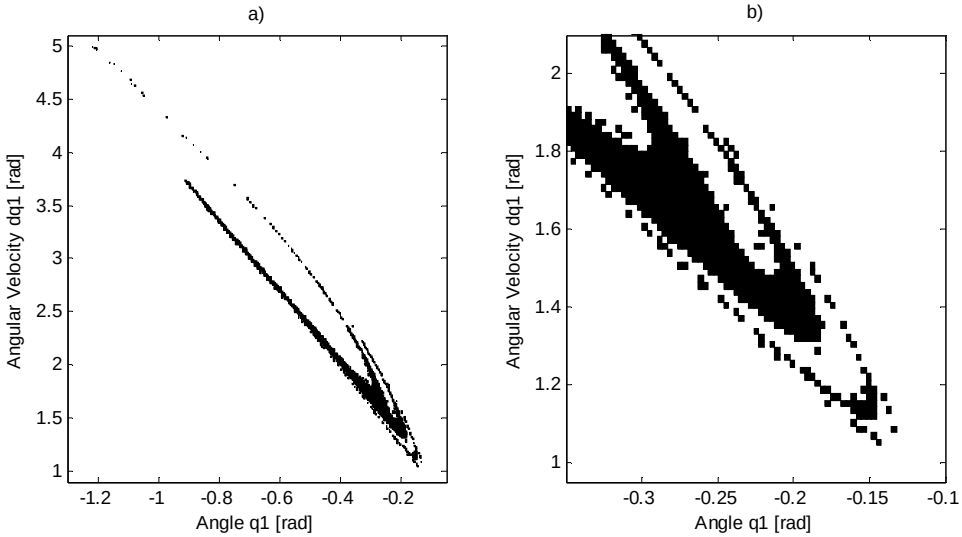

Figure 4-23: Period 1 UPO's basin of attraction - a) Entire BOA, b) Typical operational BOA

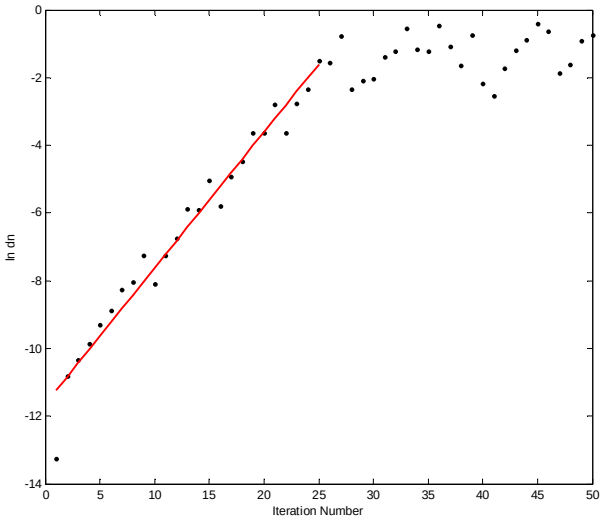

Figure 4-24: Plot of *ln* d_n versus iteration number.

Closed return distances of 200 consecutive walking steps were plotted as shown in figure 4-25. All initial conditions with D_n<0.1 were inspected. The lowest D_n pair (consecutive knee and heel D_n) was selected as the initial fixed points for linearisation of the Poincaré Map. Figures 4-25 (a) and (b) show that the lowest D_n pair is at the 135th iteration/ step. All points in figure 4-25 with D_n<0.1 are closed returns from a single fixed point. In any chaotic system of certain conditions and parameters, there is generally more than one period p UPO [9]. It is however not the case here as only one period 1 UPO could be found for γ =0.0985 rad. The fixed points found, using linearisation of the Poincaré Map method, are:

$$Z_f^k = [0.0807 \quad -0.5199 \quad 1.0230 \quad -1.2620]^T \tag{4.65}$$

$$Z_f^h = [-0.2891 \quad 1.6320 \quad -0.1708]^T \tag{4.66}$$

with corresponding control parameter:

$$p_n^k = p_n^h = 0 \tag{4.67}$$

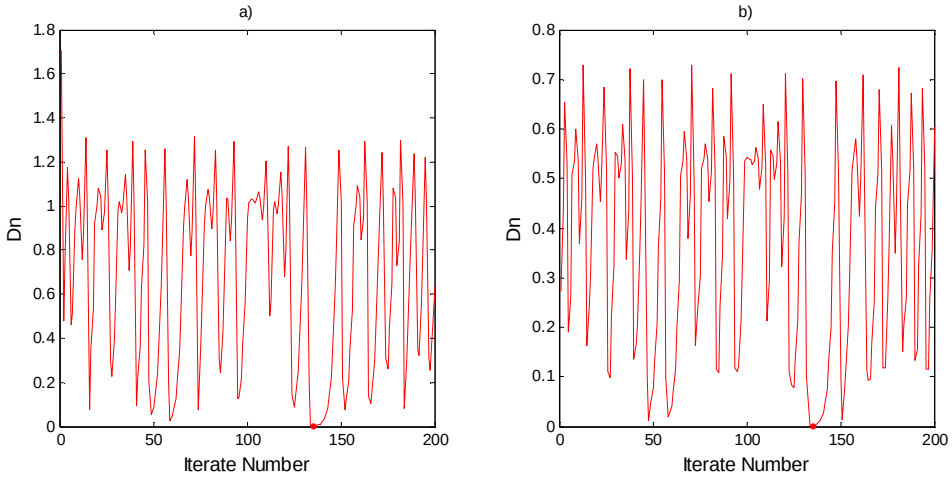

Figure 4-25: UPO detection Closed Return Plot - a) Knee plot, b) Heel plot

The period 1 UPO has a walking step length and time of 0.459 m and 0.653 s respectively (walking speed of 0.703 m/s). For the calculation of the linearised Poincaré Map (equation 4.48), data from 300 consecutive walking steps were used. The period 1 UPO linearised Poincaré Map was determined using equation 4.32 to equation 4.42. The tolerance values used are as follows:

The tolerance value of knee and heel r_n: ε_r^k =0.2; ε_r^h = 0.1

The tolerance value of knee and heel D_n: ε_d^{knee} = 0.2; ε_d^{heel} = 0.1

Figure 4-26 shows the plots of heel and knee r_n as a function of the iteration number. All points below the ε_r line (red dashed line) are within the ε_r limit. All points that satisfy both ε_r and ε_d limits are used to determine the linearised Poincaré Map matrices. The use of smaller ε_r or ε_d limits results in a more accurate but smaller linearised space mapping. Smaller tolerance limits also result in a smaller acceptable data set. If the iteration number is limited, such as the number of walking steps in the experiments, the mapping accuracy will also be limited due to insufficient data sets.

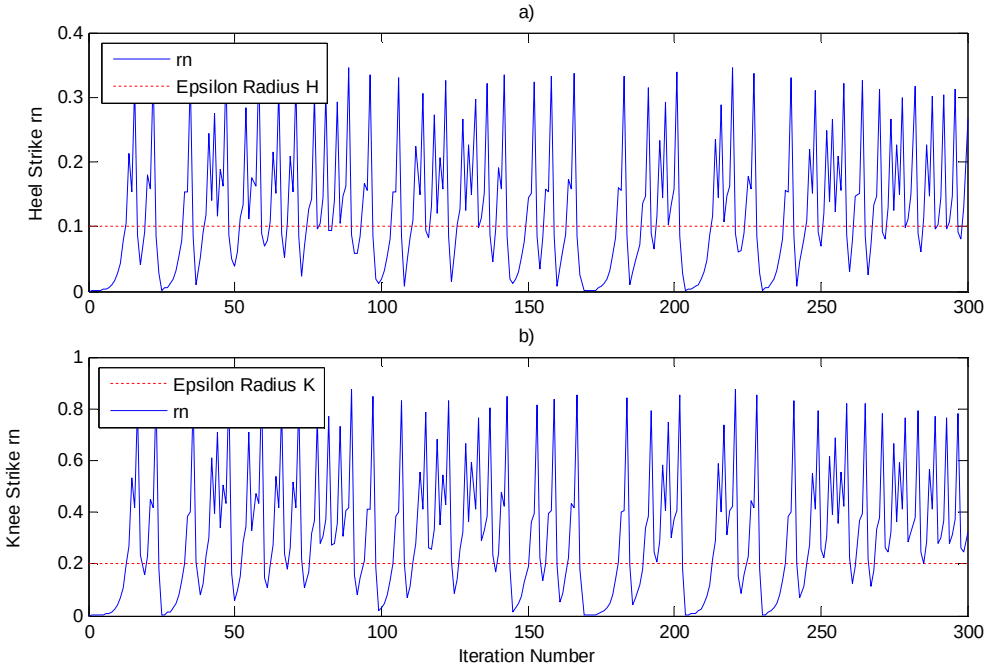

Figure 4-26: Plot of r_n as a function of iteration number. a) Heel r_n plot b) Knee r_n plot

The selection of control parameter change Δp in equation 4.42 for the linearisation of the Poincaré Map process is dependent on the underlying dynamics of the system. Figure 4-27 shows the bifurcation diagram of the u_2 variation on the long-term dynamic of the system. For γ =0.0985 rad, if u_2 is in the range of [0.025, 0.3] Nm, then orbits are periodic. If orbits are periodic, no recurrence point near the linearised fixed point can be found which results in a no shift vector B (equation 4.5). If u_2<- 0.0157 Nm, walking becomes unstable. Due to the above, the selected control parameter change Δp must be in the range of [-0.0157, 0.25) Nm for the period 1 UPO control.

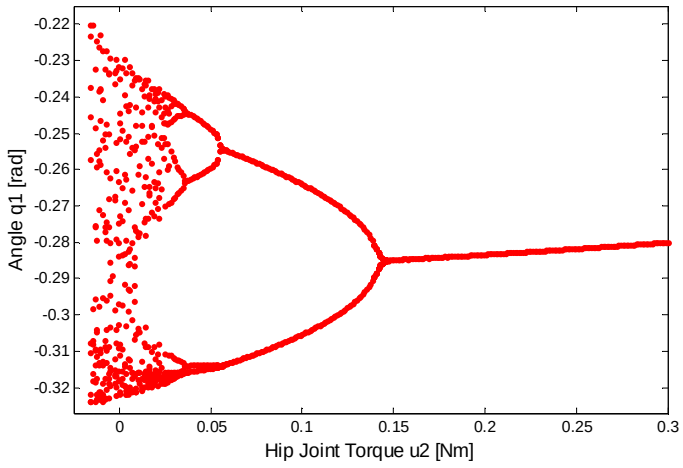

Figure 4-27: Bifurcation diagram of hip joint torque u_2 variation at γ =0985 rad

For the period 1 UPO, the matrices of the knee and heel linearised Poincaré Maps of equation 4.49 and equation 4.50 respectively are as follows:

Knee linearised Poincaré Map matrices:

$$A^k = \begin{bmatrix} -31.4881 & 8.4255 & 5.2538 & 1.0807 \\ -26.5639 & 7.1739 & 4.3177 & 0.9403 \\ -109.9170 & 29.3161 & 18.5407 & 3.7247 \\ -180.5760 & 48.1321 & 31.3741 & 5.8548 \end{bmatrix} \quad (4.68)$$

$$B^k = [0.1070 \quad 0.3720 \quad 0.2439 \quad 0.1750]^T \quad (4.69)$$

Heel linearised Poincaré Map matrices:

$$A^h = \begin{bmatrix} -2.1509 & -0.6530 & 0.0311 \\ 0.9707 & 0.2548 & -0.0074 \\ -13.484 & -4.4472 & 0.2985 \end{bmatrix} \quad (4.70)$$

$$B^h = [0.0323 \quad 0.0842 \quad 0.5073]^T \quad (4.71)$$

The computation of the perturbation difference Δp_{n+1} (equation 4.23) depends on the number of unstable manifolds at the fixed point. The eigenvalues of the fixed points, arranged in the order of inequality 4.9, are:

Z_f^k Eigenvalues: λ_1= 1.6760; λ_2= -1.6386; λ_3= 0.0767; λ_4= -0.0328

Z_f^h Eigenvalues: λ_1= -1.6528; λ_2= 0.0830; λ_3= -0.0278

There are two unstable manifolds at Z_f^k and one unstable manifold at Z_f^h. The perturbation difference equations of the fixed points becomes:

$$\Delta p_{n+1}^k = \frac{{\lambda_1^k}^2 f_1^k \left(Z_n^k - Z_f^k\right)}{(\lambda_1^k - \lambda_2^k)(\lambda_1^k - 1) f_1^k B^k} - \frac{{\lambda_2^k}^2 f_2^k \left(Z_n^k - Z_f^k\right)}{(\lambda_1^k - \lambda_2^k)(\lambda_2^k - 1) f_2^k B^k} \quad (4.25)$$

where

$$\lambda_1^k = 1.6760$$

$$\lambda_2^k = -1.6386$$

$$f_1^k = [-80.8295 \quad 21.3354 \quad 15.5580 \quad 2.2361]$$

$$f_2^k = [-48.4328 \quad 13.0523 \quad 6.6158 \quad 2.0589]$$

and

$$\Delta p_{n+1}^h = \frac{\lambda_1^h f_1^h \left(Z_n^h - Z_f^h\right)}{(\lambda_1^h - 1) f_1^h B^h} \quad (4.24)$$

where

$$\lambda_1^h = -1.6528$$

$$f_1^h = [7.7268 \quad 2.3789 \quad -0.1142]$$

The above two perturbation difference equations were applied in conjunction with equation 4.51 to equation 4.58 in the control of period 1 UPO. The constants in equation 4.51 and equation 4.55 are $\varepsilon_{rf}^k = 0.5$ and $\varepsilon_{rf}^h = 0.5$ respectively. The limits of perturbation difference in equation 4.52 and equation 4.56 are $\Delta p_{max}^k = \Delta p_{max}^h = 0.04\,Nm$. Limitations due to the effects of u_2 variation on system dynamics, as shown in figure 4.27, are not a concern in the chaos control process. Since the perturbation torque changes between iterations, except after convergence, long-term dynamic effects which require constant u_2, will not influence the intended chaotic behaviour.

Control of the period 1 UPO began with the following unlocked knee swing phase initial condition:

$$[q_1, \dot{q_1}, \dot{q_2}] = [-0.28\,rad, 1.6 rad/s, -0.17\,rad/s] \tag{4.72}$$

Chaos control was turned on in the first 20 steps, turned off in the next 30 steps, and turned back on in the last 30 steps. Figure 4-28 shows the plot of the distance (r_n) between the state point and the fixed point for knee and heel Poincaré Sections. The red and blue dash lines on the figure are the control conditions for condition 4.51 and condition 4.55 respectively. Chaos control algorithm was executed if one of the knee or heel r_n values was below its control condition line. From the figure it can be seen that at iteration one or step one, both control conditions are satisfied which results in unlocked or locked knee swing phase perturbation control (depends on which one comes first).

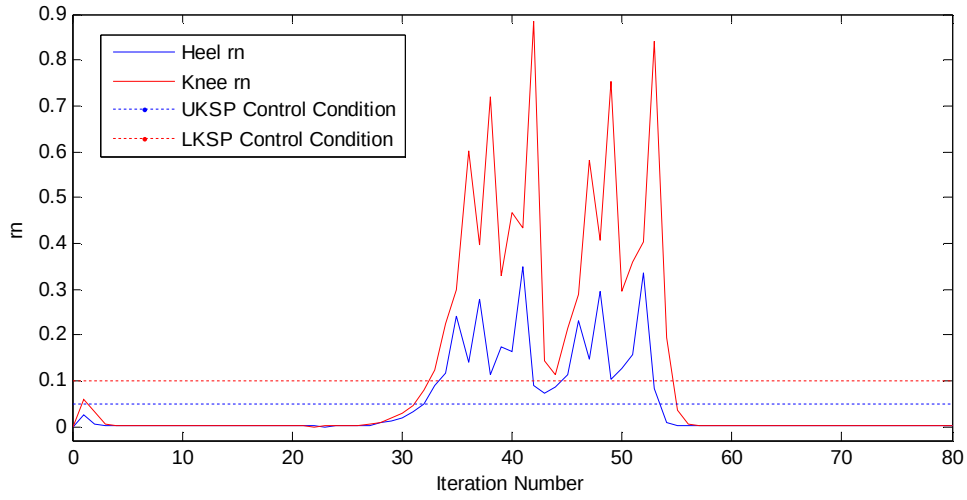

Figure 4-28: Plot of r_n distance versus iteration number with control

Figure 4-29 shows the plot of the perturbation torque u_2 as a function of the iteration number. The control parameter remains constant for the duration of a walking phase. Since chaos control can be applied twice in a step, the control parameter may change its value between walking phases. Figure 4-29 shows that after the control condition was met in step one, perturbation was applied in step two. After step two, the perturbation torque decreases significantly because the state point is now on the stable manifold. The state then converges toward the fixed point along the stable manifold in the next two steps.

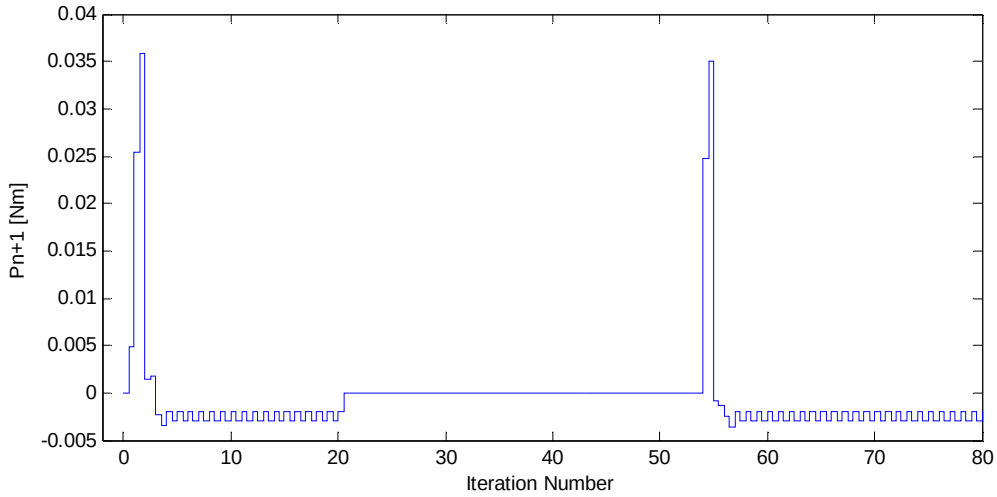

Figure 4-29: Plot of perturbation torque as a function of iteration number

The converged state points are:

$$Z^{k}_{converged\ point} = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.0802, -0.5208, 1.0217, -1.2633]^T \quad (4.73)$$

$$Z^{h}_{converged\ point} = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.2892, 1.6318, -0.1722]^T \quad (4.74)$$

The distance between the fixed points and the converged points are: $r_n^k = 0.00208$ and $r_n^h = 0.00144$. At convergence (after 4 iterations) the control parameter oscillates around -0.0025 Nm. Since UPOs are unstable by nature, it will require control to maintain their periodicity. The value of the calculated fixed point may differ from the actual point due to linearisation and approximation. Due to this difference, the control parameter at convergence is close but not exactly equal to its nominal value of $p_n^k = p_n^h = 0$.

When control was turned off at the 20th iteration, the state moves away from the fixed point and wanders chaotically as indicated in figure 4-28. Control was turned back on after the 50th iteration. At iteration number 54, the control condition was met again. The chaotic orbit converges onto the previous UPO fixed point once more. If no control is applied to the system, the chaotic orbit will wander around the UPO fixed point as shown in figure 4-30. It can be observed that the frequency of a chaotic orbit coming close to a period 1 UPO fixed point, is low. This of course depends on the radius of the 'neighbourhood' of the fixed point.

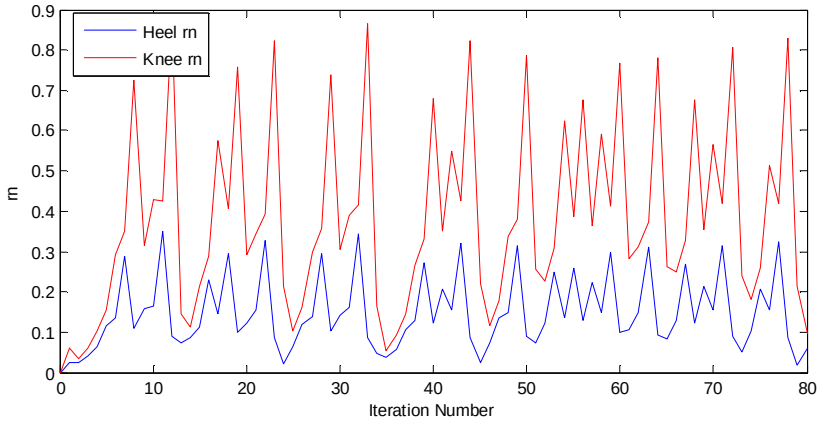

Figure 4-30: Plot of r_n distance versus iteration number without control

4.4.5 Higher Period UPOs

The fixed points of a period 2 UPO for γ =0.0985 rad was determined and control of it was attempted. The fixed point pairs and its eigenvalues:

Fixed point pair 1:

$$Z_f^k = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.0182 \quad -0.5844 \quad 0.8320 \quad -1.5805]^T \tag{4.75}$$

$$Z_f^h = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.2442 \quad 1.6100 \quad 0.0769]^T \tag{4.76}$$

For:

Z_f^k Eigenvalues: λ_1= 3.0291; λ_2= -2.2492; λ_3= 0.0597; λ_4= -0.0216

Z_f^h Eigenvalues: λ_1= 3.1986; λ_2=-1.9513; λ_3= 0.0619

Fixed point pair 2:

$$Z_f^k = [q_1, q_2, \dot{q_1}, \dot{q_2}]^T = [0.1670 \quad -0.4372 \quad 1.3065 \quad -0.9282]^T \tag{4.77}$$

$$Z_f^h = [q_1, \dot{q_1}, \dot{q_2}]^T = [-0.3300 \quad 1.6963 \quad -0.3201]^T \tag{4.78}$$

For:

Z_f^k Eigenvalues: λ_1= 5.4908; λ_2= -2.1833; λ_3= -0.2269; λ_4= 0.0230

Z_f^h Eigenvalues: λ_1= 1.0214; λ_2= -0.1360; λ_3= 0.0548

Control of the period 2 UPO was not achieved as the trajectory fails to converge to the fixed points. Three of the four fixed points above has two unstable manifolds. Fixed points with two unstable manifolds

require two iterations before the state point can converge on to the stable manifold. Since the UPO has periods of two, control correction of the same fixed point can only occur for once every two iterations. The linearised Poincaré Maps also does not take strike transition effects into account and chaotic orbits are non-smooth. It may be due to the above combined effects that result in non-converging of chaotic orbits. Control of period 2 UPO using different slope angles was also not achieved.

Period 2 or higher UPOs for biped walking correspond to limping gaits. Typically symmetrical gaits are often the ones of interest as performance is consistent. Unlike limit cycles, performance of higher period orbits such as step length and speed, differ according to period. Period 3 or higher UPOs have not been simulated due to undesirable biped performance reasons.

4.5 Chaos Targeting

Chaos targeting is the process of steering the system dynamic to the system flow of desire. Targeting control shortens the transport time for the system flow to get close to the target point [57]. System flow is steered into the region of the target point through a sequence of perturbation control.

Chaos control using the OGY method relies on the ergodicity of the system in order to bring the orbit flows close to the region of target point. Only once the system flow is in the vicinity of the target point, can the system be stabilised. Depending on the system, typical transport time can be exceedingly long when using the OGY method. Typically for high dimensional systems, the average transport time is beyond feasible waiting time. Transport time also depends on the initial condition of the system. For systems such as the sagittal biped model, this implies that long walking distance needs to be traversed before control can take place. This may cause difficulties in the conduction of experiments. The average transport time for a point to get close to a fixed point, due to ergodic recurrence, is a power law relation as indicated in equation 4.79 [58]:

$$\tau_0 \sim (1/\varepsilon_t)^{D_t} \tag{4.79}$$

where τ_0 is the average transport time from starting point to target, ε_t is the linear size of the neighbourhood and D_t is the pointwise dimension about the target point.

General chaos targeting procedures are presented in the Handbook of Chaos Control [52], and by Macau and Grebogi [59]. The targeting procedure describes the formulation of a short constructive orbit through forward and backward iteration processes. Targeting control was not implemented in this research due to difficulties in conducting the targeting procedures.

4.6 Chapter Summary

The OGY method for chaos control does not require prior knowledge of the system such as modelling equations. The control of a chaotic system using control parameters derived from previous trajectory data set has been experimentally conducted by Ditto *et al.* [10] and Shinbrot *et al.* [42]. None of the chaos control algorithms presented in this chapter is based on the governing equations of the biped model derived in Chapter 3. For theoretical and experimental chaos control of the biped model, the required inputs are simply the captured results of walking trajectories. All that is required of the chaos control

procedure are large numbers of data points from heel and knee strike conditions. From the gathered data, UPO point(s) are detected and the one of interest is selected. Again using the captured data set, a linearised Poincaré Map and a perturbation equation about the UPO fixed point are calculated. Chaos control is applicable to the system if the control conditions are met.

This chapter has demonstrated the control of a period 1 UPO using a stated biped model. The convergence results of the period 1 UPO at the specified slope angle was shown. Control of period 1 UPO at different slope angles is also achievable (not shown in the chapter). A period 2 UPO was located but its control was not achieved due to reasons previously discussed. Control of period 3 or higher UPOs was not simulated as these cases correspond to asymmetric gaits or limping gaits (likewise with period 2 UPO). Asymmetric gaits are undesirable as their system dynamics vary from step to step. High dimensional orbit control may be possible but it is undesirable for the biped model due to performance inconsistency.

The statement regarding the use of small perturbation in controlling chaotic systems has been substantiated. Figure 4-27 shows the change of long-term states due to hip torque variations: large torque variation is required in order to shift a chaotic orbit at u_2=0 Nm to a period 1 cycle ($|u_2|$>0.15 Nm). Figure 4-29 plots the control torque required to converge a period 1 UPO with the same conditions as above. Both the maximum perturbation torque ($|u_2|$>0.035 Nm) and converged torque ($|u_2|$>0.0025 Nm) of chaos control are much smaller than the torque used to shift chaotic orbit into periodic regions. This proves the energy efficiency of chaos control over conventional parameter variations of periodic systems.

It was intended to locate many period 1 UPOs for a certain operating condition. The system performance can then be altered through the control of chaotic orbits onto the different UPOs. From the literature review, chaotic systems are shown to have large numbers of UPOs in their chaotic attractors [4]. There are also generally more than one period p UPOs for any given condition [9] which is not the case for the simulated model. From simulations it has been determined that there is only one period 1 UPO for each slope angle. It is uncertain whether only one period 1 UPO exists with different model parameters. The system remains chaotic due to the dynamics of different orders of UPOs co-existing. Even though there is only one period 1 UPO fixed point per Poincaré Section, there are other UPO fixed points of a different order on it as well. The initial idea of directing a chaotic orbit to different period 1 UPO was proven not possible. Obtaining performance change through control of orbits onto higher dimensional orbits is not desirable for the reasons previously stated.

CHAPTER 5 BIPED DESIGN

5.0 Introduction

Design of the biped requires knowledge from various fields of research. The design process was split into the following categories: electronic hardware selection, mechanical structure design, controller hardware selection, and software programming. The design specification of the biped is governed by constraints, assumptions, and conditions stipulated in the chapter on modelling dynamics. The actuators and sensors were first selected, followed by the design of the mechanical biped segments. The design incorporates the optimal parameter range determined in Chapter 3 plus counter-measures to different modes of walking failures. Commercial control boards were selected to operate the electrical and electronic components for ease of programming. The design of the control system was not conducted due to the difficulties stated in this chapter. Simulations were conducted using hypothesised experimental conditions and the results used to determine the control system's performance requirements.

5.1 Design Requirements

The design specifications comprise research findings, parameter optimisation and the constraints discussed in previous chapters. The dimensions of the biped parameters are based on the simulated results in the modelling dynamics chapter. The design parameters must be within range according to table 3-3 and the biped must operate in accordance with certain modelling assumptions. The biped's locomotion must be restricted to sagittal plane walking and leg swing motions must be constrained from yaw or roll rotations.

Since the focus of the dynamics is based on the sagittal motions of the biped, the structure of the biped only needs to be in accordance with the model from its side view. From the sagittal plane view, the biped, in accordance with the studied model (figure 3-1), consists of two legs connected at their hip joint. Each leg consists of a thigh, a knee joint and shank. All joints (low resistance joints) must be capable of passive swinging motions as in passive modelling. Knee caps or other mechanisms must be included to prevent hyperextension of the knees. To emulate the knee strike transition effect, the knees must be impact absorbent and inelastic. The shank's ankle must be rigid. A roll-over foot design (feet with circular curvatures) must be used. The design of the feet base must also comply, as closely as possible, with a no-slip assumption and inelastic heel strike condition.

Actuation of the hip and knee joint locking mechanism was required; the hip joint actuation was selected as the control parameter in the chaos control chapter. Hip joint actuation implies that the relative angle between the thighs must be controllable. Self-locking and unlocking knees were required during the execution of the four phases. The stance knee needs to remain locked as an additional stance knee collapse prevention feature while the swing knee must lock after knee strike and the stance knee unlock after each heel strike.

Data acquisitions regarding the generalised coordinates were required for data analyses and control reasons. The generalised coordinates (q_1, q_2 and q_3) must be recordable as a function of time. The velocity

generalised coordinates are simply the time derivative of its angular coordinates. Most importantly, the immediate generalised coordinates after knee and heel strikes must be recorded with accuracy. The additional inclusion of a hip torque measurer was used for two reasons: even though the hip joint torque can be calculated from captured generalised coordinate data, it is measured for result comparison, and the torque measurement can serve as feedback for the control system. The design of a control system which controls the hip joint torque was required in accordance with the chaos control strategy.

5.2 Electronic Hardware

The electronic design is categorised into the sections: actuators and sensors. The actuator was selected first and is based on the forces or torques required to drive the system. This was followed by the selection of appropriate sensors that can measure the required data during testing. Sensing was required as part of the control process as it provides feedback to the system.

5.2.1 Actuators

5.2.1.1 Hip Actuator

Hip actuation is required for chaos control or as a source of torque generation in active walks. The biped needs to be capable of passive and active walks. During passive walks, the actuator must act as a free rotational joint with minimal retarding force. In active walks it must be capable of supplying the required torque. The actuator may be required to alternate between passive and active modes at certain times which means that the actuator will be either on or off throughout the walking cycle. The size of the actuator must not be too large as it needs to fit into the hip segment and the weight must be small enough so that the mass of the hip segment falls within the specified range. The typical types of actuators used as robotic drives are servos and direct current (DC) motors. Other types of actuators such as air muscles and hydraulic actuators are not considered. This is mainly due to weight constraints and the fact that these types of actuators are generally more costly.

Servos are motors with integrated gearbox and angular position controllers which can be controlled to precisely rotate to the desired angles. Servo motors also has advantage over other motors in terms of position control and accuracy. Its limitations are that commercial servos typically have very small torque outputs (less than 4Nm) when compared to geared DC motors. Most servos on the market either have small torque output limitations or it comes at a high cost. Powered servos are constantly in position control. If the servo shaft is offset due to an external force, its integrated controller will attempt to rotate the shaft back to its original position. If the external force is much greater than the servo torque, gearbox damage may occur. The control parameter of the biped has units of torque. Because of the servo's control angular position, the torque control methodology will need to be converted to an equivalent position control algorithm.

DC motors without high reduction gearboxes attached are back-driveable with small retarding forces. This implies that it can rotate passively or actively depending on whether it is powered or not. DC motors with the specified weight constraint typically have much smaller output torques than required. This means that a gearbox is required to step up the torque output. The disadvantage of using a motor coupled to a

gearbox is that the gearbox imposes large resistive torques due to its gears. The gear resistance is proportional to its gearbox reduction ratio. Typically DC motors are controlled such that it turns with just enough power to overcome the resistive torque problem. With control, geared DC motors can be either active or passive.

A DC motor was selected as the hip actuator. Even though servos are geared DC motors with integrated control, they are not used due to their torque limitations. The first part of the funding had to be spent prior to a deadline given. The DC motor, being the second most expensive single item, was purchased prior to the research results of the chaos control. The selection of the motor was based on the results in Chapter 3 and the design specifications of other researches similar to this dissertation.

A biped model (Dribbel), with similar features and parameters as the one presented in this dissertation, developed by the University of Twente, was investigated [60]. It however has a lower body model and flat feet. Dribbel is 1.01 m in height, weighs roughly 9.5 kgs and is capable of walking on flat ground when the hip is actuated. It requires a peak torque of 12 Nm and continuous torque of 3 Nm for walking. The geared DC motor that it uses has a maximum continuous torque capacity of 15 Nm.

An approximation of hip torque was determined using a model with parameters based on the average of table 3-3 [62]. The slope angle of the simulations was set to zero which is flat ground. Through trial and error it was determined that with a hip torque (u_2) of -2 Nm allowed, walking continues for a few more steps on flat ground before the biped fell. The magnitude of hip torque for continuous walking also depends on the starting condition.

Based on the results of the flat ground walking simulation and Dribbel design parameters, a Maxon DC motor with gearbox was selected for its properties of light weight, small size and high torque capacity. The selected model was a Maxon Motor Re40, which is a permanent magnet, brushed typed motor. A planetary Gearhead GP 42C was selected as its gearbox. The motor and gearbox have the following capacities and specifications:

Table 5-1: Specifications of Motor and Gearbox

Motor + Gearbox Specifications	Value
Power rating	150 W
Nominal Voltage	24 V
No load speed (motor)	7580 rpm
Nominal current (max. continuous current)	5.77 A
Gearbox reduction	73.5:1
No load output speed (gearbox)	102 rpm
Maximum continuous torque (gearbox)	13.4 Nm
Intermittently permissible torque output (gearbox)	22.5 Nm
Maximum efficiency	72%
Weight (Combined)	0.94 kg

The simulation results of chaos control in Chapter 4 (figure 4-29) show that for one scenario the maximum required perturbation torque is less than 0.04 Nm. Based on the perturbation magnitudes of chaos control, it is safe to estimate that the perturbation magnitude of chaos control is within ± 1Nm of the nominal hip torque value. Also from the hip joint torque variation simulation of figure 4-16, the results show that hip torque exceeding the $|u_2|$ >0.5 Nm range leads to unstable walking. Based on the above conditions, the selected actuator is over-designed in terms of chaos control requirement. In terms of active walking, the torque requirement will be verified in Chapter 6.

DC motors are controlled through the variation of its supplied voltage. For the selected DC motor, the maximum operational voltage is 24V with a maximum continuous current of 5.77A.

5.2.1.2 Knee Lock Actuator

A knee locking system is required to hold the leg straight during walking. Knee lock is used to prevent stance knee collapse which must unlock at the start of the swing phase. Knee lock of the swing leg after knee strike is required as condition 3.32. It is possible to design a passive locking system. The momentum of the swing shank prior to knee strike is sufficient to overcome typical retarding forces in locks. The problem arises in the design of the passive unlocking function. Typically the resistive force that prevents the knee unlocking is larger than the swing leg force which unlocks it. Because of this, most knee locking devices include an actuator to overcome the resistive force. Some knee locking systems used in other research designs are mentioned below.

A combination of solenoid and a permanent magnet is used for the knee lock [34]. When the knee joint is straightened during the instant of knee lock, the permanent magnet comes into effect and holds the knee through magnetism. The size and strength of the magnet depends on the mass of the shank. To release the knee, the solenoid is powered which counters the magnetism of the permanent magnet. This method is not always reliable. If the impact is large, the collision force will overcome the magnetism which leads to the shank bouncing back.

The most common type of knee lock utilises some kind mechanical locking pin or hooks. These types of locking systems are normally designed using springs. An actuator is used for the lock release function. Its disadvantage is that sometimes, when the flexion torque of the knee is large, the mechanism tightens. As the locking device is held tight, the actuator is not able to release it in time.

The above-mentioned knee locking devices were not used because of the disadvantages mentioned. The device designed in this dissertation is the mechanical locking type. When the knee straightens, the actuator rotates a mechanical stop into place. For unlocking, the mechanical stop is released. The mechanism was designed such that the actuator does not directly oppose the flexion torque of the swing leg. Actuators are only required to overcome surface frictions. The knee lock design is presented in the mechanical design section of this chapter (section 5.3.4). The actuators chosen for the locking and unlocking rotation of the mechanical stop are servos because they provide the rotating action and are considerably light weight and small. Hitec HS-322HD servos were chosen as they are locally available and fulfil the requirements. The specifications of the chosen servo are as table 5-2 [63].

The servos are controlled using the Pulse-Width Modulation (PWM) method. PWM signals are square waves resulting from switching a constant voltage value on and off and the signals are quantified in terms of their duty cycle. Duty cycles express the time that the voltage is on as a percentage of its cycle period. The angle of the servo is directly proportional to the duty cycle of the PWM. Duty cycles of 0 % and 100 % correspond to -90° and 90° of the servo position. In order to run the selected servo at its maximum speed capacity, its supply voltage is the rated maximum voltage (6 V).

Table 5-2: Specifications of Hitec HS-322HD Servo

Specifications	Value
Operating Voltage	4.8-6.0 V
Operating Speed (4.8V)	0.19s/60° at no load
Operating Speed (6.0V)	0.15s/60° at no load
Stall Torque (4.8V)	0.294 Nm
Stall Torque (6.0V)	0.363 Nm
PWM Pulse	3-5 V peak to peak
Weight	0.043 kg

5.2.2 Sensors

Sensors are used for data capture and control purposes. Optical encoders and accelerometers were used for measuring generalised coordinates. It is complex to measure the generalised coordinate angles the way they are defined. The manipulation of the measured angles is required to obtain the generalised coordinates. Micro switches were used as knee and heel strike detection sensors. A torque transducer was selected for the measurement of the hip joint torque.

5.2.2.1 Optical Encoders

Optical encoders were selected for the measurement of the knee and hip joint angles. The angle measured at the knee joint is the acute angle formed by the thigh and shank of the leg. The hip joint angle is the inner angle between the inner and outer thighs. The angular velocities of joints were calculated using the time derivative of the measured angles. The measured angles were then converted to generalised coordinates q_2, q_3, $\dot{q}_2$ and $\dot{q}_3$. Optical encoders were used because of their low operational resistance and simplicity. The optical encoder was a HEDS 5540-A06 with specifications as shown in table 5-3 [64]. The encoder is an incremental type with 500 output pulse (square waves) per revolution. The angle of rotation is determined by counting the number of pulses generated during the shaft turns. The encoder's reference angle is its power up position. This means that the measured joints have to be in the reference position for every start-up.

Table 5-3: Specifications of HEDS 5540

Specifications	Value
Count per Revolution (CPR)	500 CPR
Number of Channels	3 Channels
Supply Voltage	0 to 7V

The resolution of the optical encoders is dependent on the pulse counting algorithm of the controller. The typical output waveforms of a three channel optical encoder are shown in figure 5-1 [64]. For every revolution there are 500 square waves in channels A and B's waveforms. Note that channel B's output is half a square wave out of phase with channel A. Channel I generates on index wave per revolution. Typically channel I's waveform is used as a counting reference for channels A and B. The angle of rotation is determined from the count of rise or fall edges, or both, in the channels. For the count of rise or fall edges of one channel, a count of 500 rise edges corresponds to 500 square waves which is a rotation of 2π radians. This has a resolution of $2\pi/500$ counts= 0.0126 [rad]. For the count of rise and fall edges using one channel, there is a 1000 count in every revolution (resolution of 0.00628 [rad]). If the rise and fall edges of both channels are counted, then there will be a 2000 count per revolution (resolution of 0.00314 [rad]). To determine the direction of the rotation, channel A and B's outputs are compared. If the rotation is clockwise, channel B will lead channel A, and vice versa. Encoders were operated using the configuration of 500 counts per revolution.

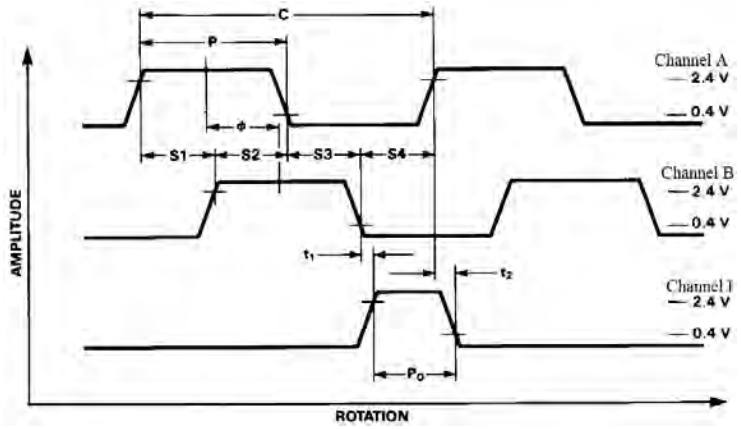

Figure 5-1: HEDS 5540 optical encoder output waveforms

5.2.2.2 Accelerometer

Accelerometers were used for the measurement of the angle between the stance leg and the horizontal ground. The measured angle was then converted to generalised coordinates q_1 and $\dot{q}_1$. Since the feet of the legs are not fixed to the ground, sensors that require no contacts such as accelerometers, gyroscopes and Inertial Measurement Unit (IMUs), are considered.

An accelerometer measures the rate of acceleration of an object. It can also be used for tilt sensing relative to the earth's gravity. Orientation is determined from the projection of acceleration due to gravity on the accelerometer axes. The disadvantage of tilt sensing is that it adds directional acceleration to gravitational acceleration which results in wrong tilt angle output. Inclination measurement using accelerometers is most accurate in a static orientation. A gyroscope measures the rate of rotation of an object. The orientation of the object is determined by integrating the measured angular velocity against time. The disadvantage of using a gyroscope is that its output values drift and accumulate errors over time. Inertial measurement unit measures the orientation and velocity of an object by combining information from both accelerometer and gyroscope. Accelerometer measurements are inaccurate in the short term

due to additional acceleration noises. Gyroscopes are accurate for a short period of time before errors accumulate. IMU utilises both accelerometer and gyroscope measurements to calibrate each other in order to generate the most accurate measurements.

Although IMU is the optimal selection, it carries a much higher cost and so accelerometers were used. The accelerometers are positioned at the foot arc centre. The linear acceleration of the stance foot arc centre is small as it travels the least distance in a walking step. The effect of directional acceleration on the accelerometer is then almost negligible. The selected accelerometer is a triple axis accelerometer breakout board using an MMA7361 integrated circuit made by Freescale Semiconductors. Its specifications are listed in table 5-4 [65].

Table 5-4: Specifications of Freescale Semiconductor MMA7361 Accelerometer

Specifications	Value
Supply Voltage	2.2 - 3.6V
Zero g Output	1.65±0.165 V
Output Sensitivity (1.5g)	800±60 mV/g
Output Sensitivity (6g)	206±15.4 mV/g

The output voltage of the accelerometer depends on its orientation. Figure 5-2 [66] illustrates the inclination sensing of the accelerometer using two of its axes (the plane spanned by the axes is normal to the ground). When using dual axes for sensing, each axis generates an output voltage. The relationship between the output voltage and the orientation of the axis is as follows [66]:

Voltage output of the x-axis:

$$V_{output\,x} = V_{offset\,x} + S_x \sin(\theta) \tag{5.1}$$

Voltage output of the y-axis:

$$V_{output\,y} = V_{offset\,y} + S_y \cos(\theta) \tag{5.2}$$

where the subscripts x and y denote the x and y-axes respectively. The symbols V_{output}, V_{offset} and S denote output voltage, offset voltage, and output sensitivity respectively. The symbol θ is the inclination angle of the x-axis relative to the ground as shown in figure 5-2. The acceleration outputs on the axes are determined from the rearrangement of equations 5.1 and 5.2 into the following forms:

Output acceleration on x-axis:

$$A_x = \sin(\theta) = \frac{V_{output} - V_{offset}}{S_x} \tag{5.3}$$

Output acceleration on y-axis:

$$A_y = \cos(\theta) = \frac{V_{output\,y} - V_{offset\,y}}{S_y} \tag{5.4}$$

The inclination angle of the accelerometer can be calculated from equation 5.3 or equation 5.4. The resolution of the accelerometer is dependent on the angle of inclination. The sensitivity of the outputs is inclination angle dependent (as shown in equation 5.1 and equation 5.2). The resolution of the x-axis lags the y-axis by a quarter of a revolution. Due to this difference in resolution of the axes, the measurements of both axes are combined. This solves the resolution-dependent problem and a more accurate reading is obtained. The algorithm used to calculate the inclination using measurements from the dual axis is as follows:

$$\tan(\theta) = \frac{A_x}{A_y} \tag{5.5}$$

Depending on the sign on the output accelerations, the angle of inclination (θ) is determined as:
If $A_x \geq 0$ and $A_y \geq 0$ then

$$\theta = \tan^{-1}(A_x/A_y) \tag{5.6}$$

If $A_x > 0$ and $A_y < 0$ then

$$\theta = \frac{\pi}{2} - \tan^{-1}(A_x/A_y) \tag{5.7}$$

If $A_x \leq 0$ and $A_y \leq 0$ then

$$\theta = \frac{\pi}{2} + \tan^{-1}(A_x/A_y) \tag{5.8}$$

If $A_x < 0$ and $A_y > 0$ then

$$\theta = -\tan^{-1}(A_x/A_y) \tag{5.9}$$

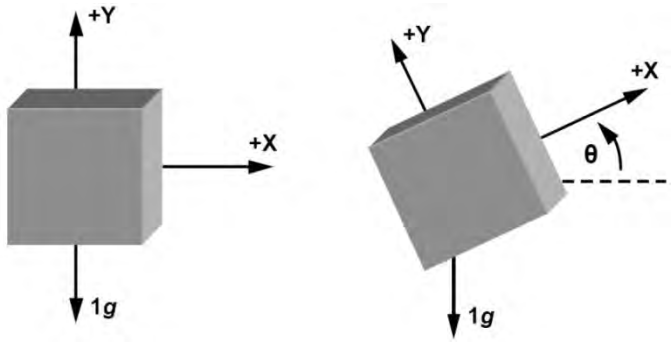

Figure 5-2: Dual axis tilt sensing

Each leg required an accelerometer for inclination measurement. The voltage output specifications of the accelerometers were verified in experiments. The specifications of the accelerometers are as shown in table 5-5. Figure 5-3 and figure 5-4 shows the experimental data plot of the accelerometer output voltage as a function of the inclination angle. Figure 5-3 (a) and figure 5-4 (a) show how the resolution of the outputs changes as the inclination angle is varied. For example, in figure 5-3 (a) the difference in output voltage between the inclination angles 80° and 90° is 0.010V, whereas the difference between inclination angles 0° and 10° is 0.131V. Output sensitivity and offset voltage were respectively determined from the gradient and y-intercepts of the graph of figure 5-3 (b) and figure 5-4 (b).

Table 5-5: Actual accelerometer specifications

Accelerometer 1		Accelerometer 2	
Specifications	Value	Specifications	Value
S_x	0.7835 V/g	S_x	0.7678 V/g
$V_{offset\ x}$	1.5465 V	$V_{offset\ x}$	1.6051 V
S_y	0.7834 V/g	S_y	0.7590 V/g
$V_{offset\ y}$	1.6495 V	$V_{offset\ y}$	1.6506 V

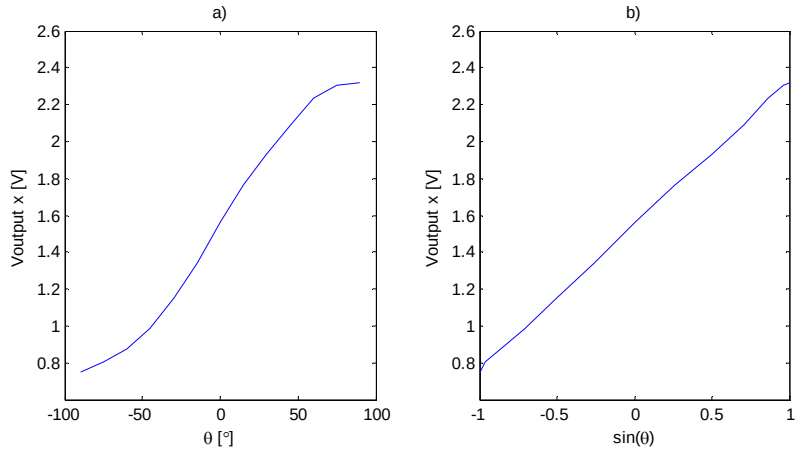

Figure 5-3: Accelerometer 1's x-axis voltage output - a) Voltage output versus θ, b) Voltage output versus $\sin(\theta)$

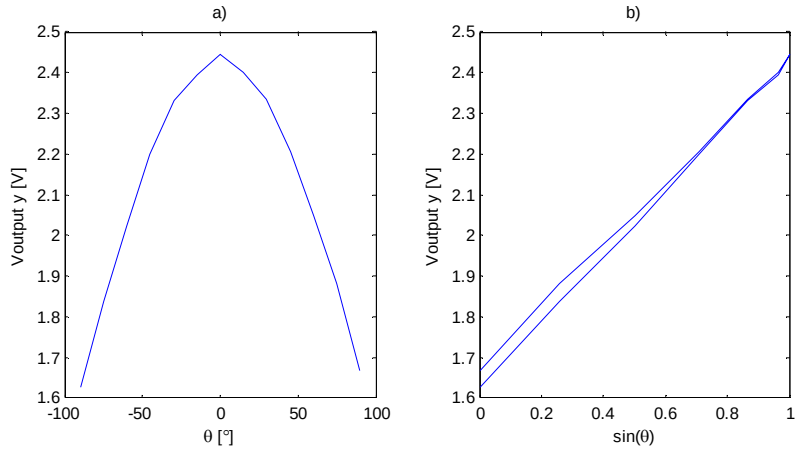

Figure 5-4: Accelerometer 1's y-axis voltage output - a) Voltage output versus θ, b) Voltage output versus $\sin(\theta)$

5.2.2.3 Micro Switches

For the detection of knee and heel strikes, micro switches were used. When the switches are pressed, the circuits are closed, and binary signals are sent to the controller. Generalised coordinates at the moment after impact strikes are identified through signals from the micro switches to the controller. The knee micro switches were also used for triggering the knee locks. The selected micro switches consist of levers with rollers.

5.2.2.4 Torque Transducer

A torque transducer was used for the measurement of the hip joint torque (u_2). A selection was made from a variety of reaction torque transducers. Reaction torque transducers measures torque of loads that have a limited range of angular rotation. The sensor itself cannot rotate continuously in one direction. Since the legs of the biped rotate back and forth (limited rotational range), reaction torque sensors were used. The criteria for the selection of the sensor were based on size, weight, cost and torque measuring range. A reaction torque transducer from Lorenz Messtechnik GmbH was selected as it satisfies all requirements and had the lowest cost. The selected transducer model was a DK-15/M312 and its specifications are shown in table 5-6 [67].

Table 5-6: Specifications of DK-15/M312 torque transducer

Specifications	Value
Sensitivity	1.00 mV/V
Measuring Range	±12 Nm
Excitation Voltage	2-12 V
Weight	0.140 kg

Torque transducers comprise of strain gauges which measures torsional strain and generate voltage outputs. The chosen model has output sensitivity of 1.00 mV/V. When the excitation voltage is *n* volts, the output voltage range is *n* millivolt, and the measured torque will be $^{12}/_{n}$ Nm per millivolt.

5.3 Mechanical Design

This section examines the mechanical design of the theoretical model. The mechanical design is based on optimal parameters determined from dynamic simulations and counter measures to typical walking failure modes. Biped designs from other researches were studied and used as part of the design options. The hip was the first designed segment, followed by the thighs and the shanks.

5.3.1 Paired Leg Design

The biped design was chosen to consist of inner and outer paired legs which are coupled to a common hip. From the side (sagittal) view the design appears to have two legs as in modelling. The paired leg design is based on the concept of McGeer's first passive walking biped [1]. Figure 2-1 (a) and figure 2-2 (a) shows photos of researched bipeds using the same design principle. Figure 5-5 shows the assembly views of the biped design using Autodesk Inventor drawing software. Figure 5-5 (a) shows the overlapping of the paired legs which makes the biped seem to have two legs from the side view. From the same view, the designed biped segments match the sagittal plane model of figure 3-1. Figure 5-5 (b) shows that paired legs consists of inner and outer couples. Figure 5-5 (c) presents an isometric view of the biped.

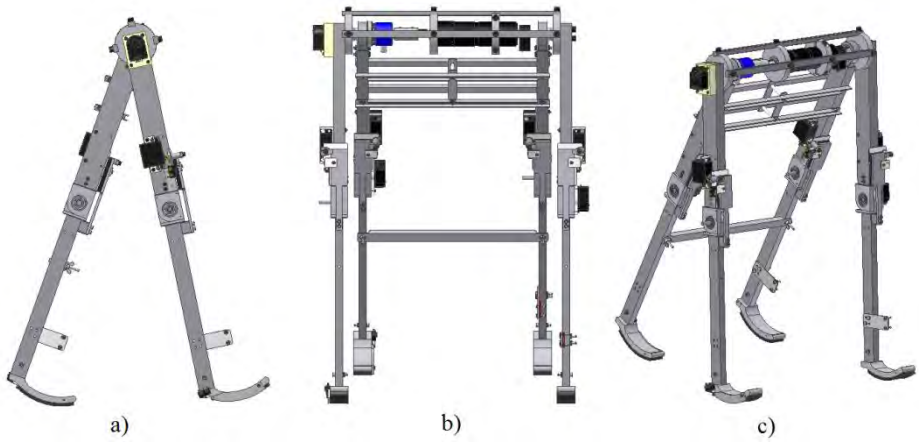

Figure 5-5: Biped assembly views - a) Side view, b) Front view, c) Isometric view

5.3.2 Hip Segment Design

The design started with the hip segment as this is the geometric centre of the biped. Figure 5-6 shows a cross section front view of the biped's hip. The inner leg pair (red components) is coupled at the thigh and the outer leg pair (blue components) is coupled to the hip segment. Due to mechanical coupling, the hip always rotates with the angular velocity of the outer thighs. In sagittal view, the inner and outer leg pairs correspond to the left and right leg of the sagittal plane biped model respectively. The hip joint actuation torque is transferred from the motor to the inner legs via the motor shaft and the torque transducer. The torque transducer measures the transmitted torque. The hip joint encoder measures the acute angle between the inner and outer legs. The hip joint encoder shaft is attached to the inner legs and its rotation is measured by the optical encoder fixed to the right outer thigh. Figure 5-7 shows the design equivalence of the biped model's hip segment. The total mass and the moment of inertia of the hip segment in modelling are calculated based on figure 5.7.

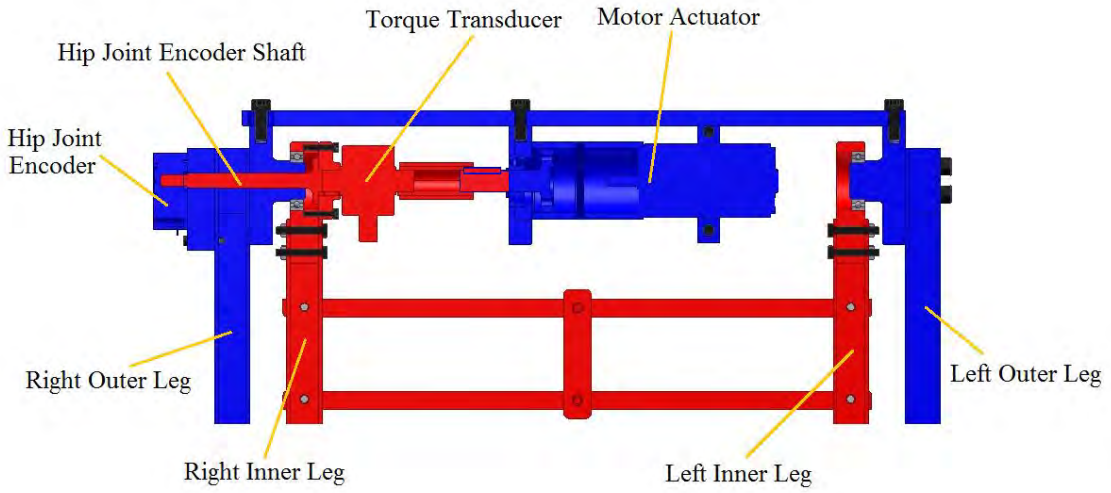

Figure 5-6: Cross section front view of the hip

Figure 5-7: Hip segment

5.3.3 Thigh Segements Design

The inner and outer thigh segments are as shown in figure 5-8. The inner and outer legs were designed such that their mass and moments of inertia are approximately equal. Figures 5-8 (a) and (b) shows the design equivalence of the biped model's left and right thigh segments respectively. Knee locking systems are present in all the thighs. Two optical encoders were used for the measurement of the knee joint angles. The knee encoders of the inner and outer legs are located at the inner right leg and outer left leg respectively.

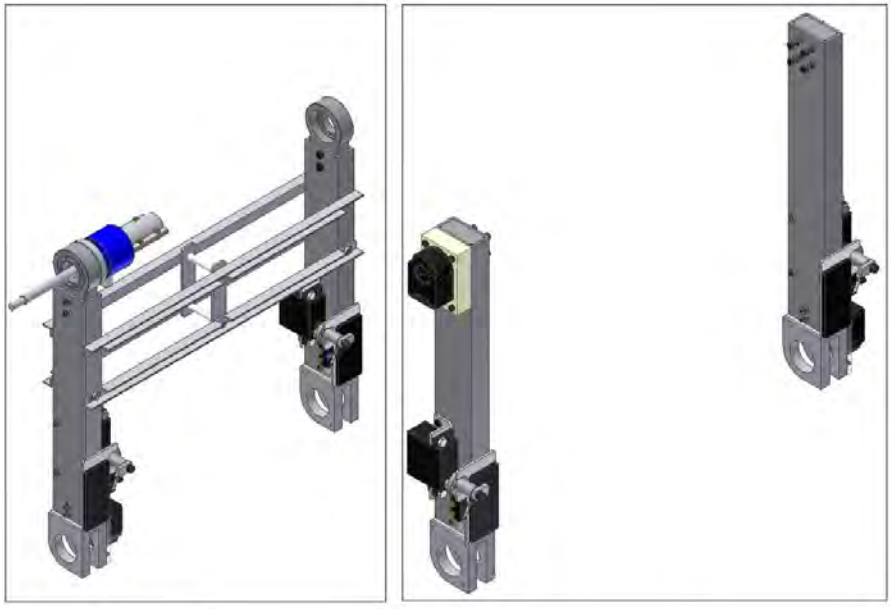

Figure 5-8: Thigh segments - a) Inner thigh, b) Outer thigh

5.3.4 Knee Segments Design

The design of the knee joint was made passive as the joint is unactuated. Figure 5-9 shows the Autodesk Inventor drawing of the outer right knee. The encoder measuring the knee joint angle is attached to one side of the thigh segment. The angular rotation of the knee joint shaft coupled to the shank was measured.

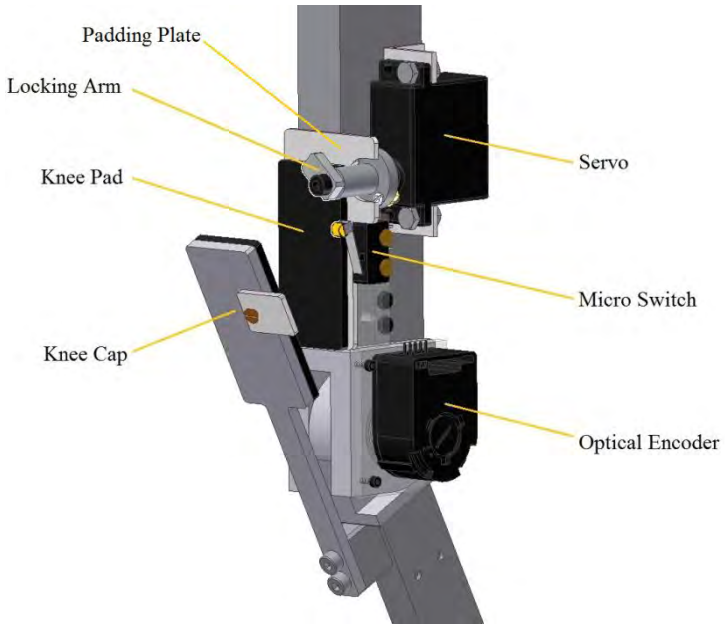

Figure 5-9: Outer right knee joint with knee lock, micro switch and encoder

The servo actuator is secured to the side of the thigh. Attached to the servo horn is the locking arm mechanism. When the servo horn rotates, the locking arm switches between locking and unlocking positions. The locking arm in figure 5-9 is in its unlocking position (diagonally positioned). When the swing leg straightens, the extended plate of the knee cap will press against the roller micro switch. The circuitry inside the micro switch is then closed which signals the servo to activate. Once the locking arm turns to its locking (horizontal) position, the top corner of the knee cap will be in position, constrained by the locking arm.

The knee cap was designed to prevent knee hyperextension and for impact absorption (knee pads). Knee collapse due to knee flexion is prevented using the knee lock system. A step in the locking arm was designed to press against the padding plate when pulled due to flexion torque (refer to figure 5-9). Since the padding plate is at the receiving end of flexion torque, it was designed to securely hold against the thigh. The magnitude of flexion torque (u_3) depends on operating conditions and is calculated using equation 3.26. Figure 3-11 (d) shows the plot of the swing knee joint torque using model parameters similar to the designed biped. The strength of the padding plate was designed based on the maximum magnitude of flexion torque in figure 3-11 (d) with an added safety factor.

The torque capacity required of the servo depends on the friction force between the knee cap and locking arm surface. Another servo selection factor was rotational speed. The servo needs to rotate and lock before the shank bounces back due to impact. The datasheet of the Hitec HS-322HD Servo, reflects a maximum rotational speed of 60° in 0.15 seconds at no load. The locking arm needs to turn 25° between

the locking and unlocking positions. The fastest reaction time of the locking system is then 0.0625 seconds. With added controller delays and the mass effects of the locking arm, the reaction time is approximated to be about 0.2 seconds.

5.3.5 Shank Segements Design

The inner and outer shank segments are as shown in figure 5-10. All shanks are identical without the additional components of the micro switch and accelerometer systems. The mass and moment of inertia of the inner and outer shank reflect less than 3 % difference. Figures 5-10 (a) and (b) shows the design equivalence of the biped model's left and right shank segments respectively. Rubber pads are attached to the bottom of the feet as impact absorbers. The pads also partially emulate the assumption of the feet being inelastic in modelling. The selection of soft or hard pads has different effects. Softer pads have better inelastic properties as the foot tends to bounce less after heel strike. However, the softer the padding, the more it deforms under pressure. When a stance foot pad deforms under load weight, the height of the entire segments drop which, in effect, lowers the grounding clearance and increases foot scuffing occurrence. Harder padding deforms less but has higher elasticity which may cause bouncing effects.

Figure 5-10: Shank segments - a) Inner shank, b) Outer shank

Figure 5-11 shows drawings of the heel strike detection design and the positioning of the accelerometer. The contact point of the heel strike foot varies depending on the orientation of the legs. The contact detection plate is screwed to the side of the foot. The base of the contact detection plate has the same radius as the foot (foot radius = 105 mm, see figure 5-11 (b)). When the foot is lifted off the ground, the contact detection plate lowers and the micro switch is released as shown in figure 5-11 (a). When heel strike occurs, the contact detection plate is forced upward which presses the micro switch.

The accelerometer was mounted so that it is located at the foot arcs' centre point. The centre point of the stance foot arc has the least linear acceleration while walking. The accelerometer is at the centre to minimise the effect of directional acceleration so that the measured acceleration will be dominated by gravity due to the inclination of the accelerometer.

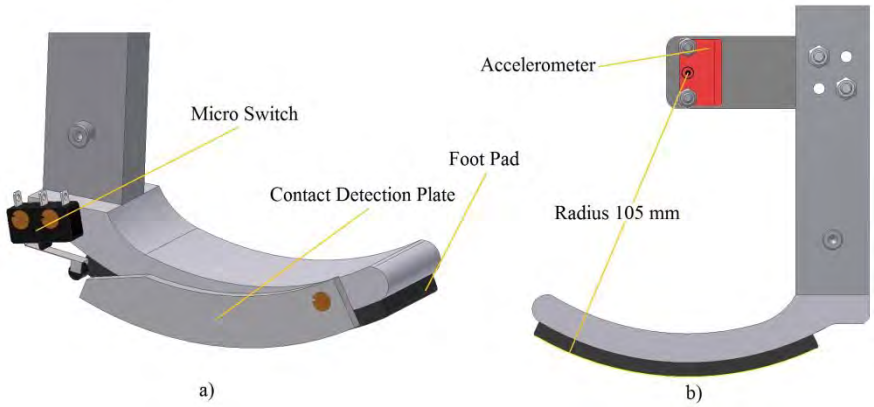

Figure 5-11: Foot attachments - a) Heel strike detection design, b) Positioning of accelerometer

5.3.6 Biped Design Parameters

All parameters of the biped design are within the parameter design range of table 3-3. Table 5-7 lists the calculated values of biped design parameters using the iProperties function of Autodesk Inventor. The inner and outer columns of the table denote inner and outer legs respectively. Parameters from the inner and outer columns are averaged and listed in the average column. In modelling, the inner and outer legs correspond to the left and right legs. For modelling simplicity, it is assumed that the left and right legs have the same parameters with values of the average column. The largest percentage difference between the average and actual parameters is 6.4 % (mass m_t).

Table 5-7: Biped Parameters

Parameters	Inner	Outer	Average
Masses		[kg]	
m_h		2.278	
m_t	1.827	2.077	1.952
m_s	1.138	1.202	1.170
Moments of Inertia		[kg .mm^2]	
I_h		1353	
I_t	18835	18936	18885
I_s	20875	20875	20875
Lengths		[mm]	
l_t		277	
l_s		330	
b_2	123	119	121
c_2	1	1	1
b_1	149	146	148
c_1	12	12	12
w_f		50	
r_s		105	

5.4 Controller Hardware

The design and selection of the controller components were conducted after the selection of the actuators and sensors. The controller is required to execute the control actions and capture motion data. It is not mounted onto the biped structure and controls the hardware on the biped through wire connections.

5.4.1 Controller Board

A controller board to control all the electronic components was required. The microcontroller board selected was an Arduino Uno. The board was selected mostly due to its user friendly coding language (based on C/C++), and online technical support. Specifications of the Arduino Uno board are shown in table 5-8 [68]. The board is compatible with and capable of handling all the selected electronic hardware.

Table 5-8: Specifications of Arduino Uno microcontroller board

Specifications	Value
Microcontroller	ATmega 328
Input Voltage	7-12V
Operating Voltage	5V
Digital Input/ Output Pins	14
Analog Input Pins	6
Clock Speed	16 MHz

Arduino Uno boards were used in order to split up the handling tasks. The built in Inter-Integrated Circuit (I^2C) communication system was used for communication between the Arduino boards. This is a two-wire interface system which communicates through the Serial Data Line (SDL) and the Serial Clock Line (SCL).

5.4.2 H-Bridge Motor Driver

A H-bridge motor driver was selected to control the DC motor. H-bridges are typically used for the control of forward and backward rotations of DC motors through the control of its supplied voltage direction. The selected motor driver is also capable of varying the magnitude of the supplied voltage through PWM control. The selected motor driver is a Pololu VNH3SP30 with specifications shown in table 5-9 [69].

Table 5-9: Specifications of Pololu VNH3SP30 motor driver

Specifications	Value
Operating Voltage	5.5-36 V
Maximum current rating	30 A
Maximum PWM frequency	10 kHz
Maximum continuous current	9 A

5.4.2.1 Snubber Design

The DC motor was required to rotate back and forth during walking. As the motor changes its rotational direction, there is a sudden change of direction in the supplied voltage which results in the occurrence of voltage or current spikes across the motor. These spikes may exceed the motor's operational limits. DC motors are designed to tolerate limited voltage or current spikes (their durability depends on the motor make and the magnitudes of the spikes). In order to protect the selected motor against continuous occurrence of power spikes, a snubber circuit is placed across the motor for protection. The designed snubber circuit uses a large capacitance capacitor (10 000 µF) to reduce voltage spike effect. A small capacitor (100 nF) is used to filter any electronic noises and current spikes are reduced using a 220 µH inductor. The schematic of the snubber circuit design is shown in figure 5-12.

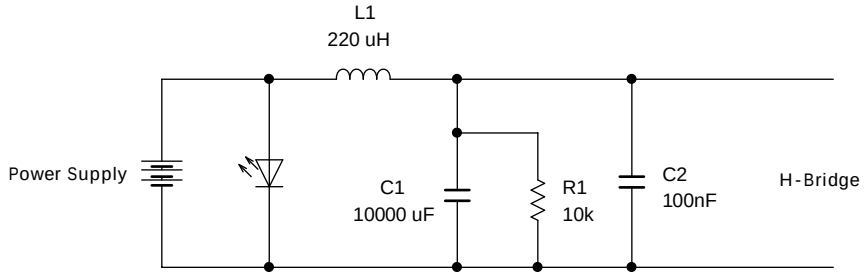

Figure 5-12: Snubber design schematic

5.4.3 Torque Transducer Voltage Amplifier

When the selected torque transducer is supplied with 12 V excitation voltage, its output voltage range is [-12 mV, 12 mV] for a respective torque range of [-12 Nm, 12 Nm]. The expected operational torque range is within [-5 Nm, 5 Nm] which corresponds to a transducer voltage range of [-5 mV, 5 mV].

When using the analog voltage read function of Arduino Uno with the reference type 'Internal', the input pin has voltage limits of [0, 1100] millivolts. The readings of analog input pins of Arduino Uno have 1024 intervals which correspond to a resolution of 1.075mV per interval. The output voltage range of the torque transducer needs to be amplified due to Arduino Uno's large analog reading resolution. A voltage amplifier circuit was designed using operational amplifiers (op-amp) for torque transducer output voltage amplification. This amplifies a voltage range of [-5 mV, 5 mV] to [100 mV, 1100 mV] and corresponds to a resolution of 0.0108 Nm per interval. The op-amps used are LF351N and the voltage amplifier schematic is shown in figure 5-13. The process of amplification is described below:

Step 1: Generates constant voltage of 6mV using the first op-amp.

Step 2: Inverts the sum of the 6mV and the output voltage of the transducer using the second op-amp. The expected output range is then [-11mV, -1mV].

Step 3: Amplifies the output voltage from the second op-amp with a gain of -100 using the third op-amp. The final expected output range is then [100mV, 1100mV].

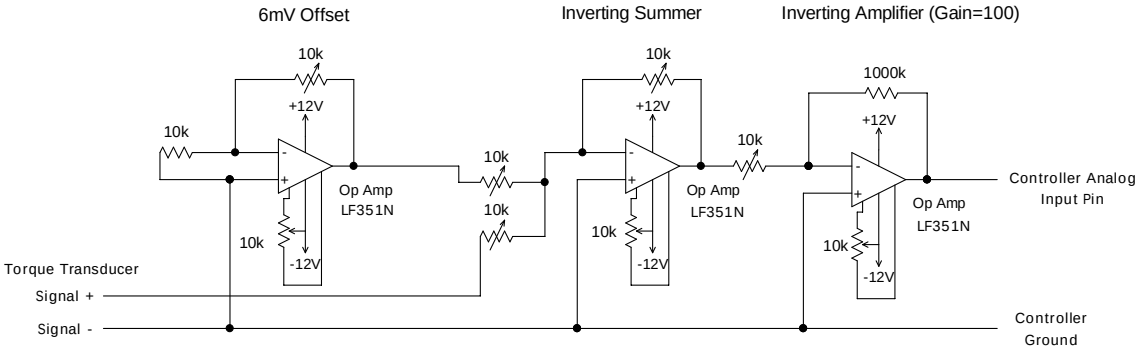

Figure 5-13: Torque transducer voltage amplifier schematic

5.4.4 Pull-Up Resistors

If the connection wire between the encoder and the controller is long, the pulse heights may be lower than expected (typically pulse heights are one volt lower than expected). Depending on the controller, some may not be able to read pulses with shorter heights as a count which results in miss counts. Pull-up resistors are connected to the signal outputs of the encoder and used to uplift encoder pulse voltage heights to the magnitude of the supplied voltage (Vcc). Figure 5-14 [64] shows the schematic of the pull-up resistors connected to the output pins of an optical encoder.

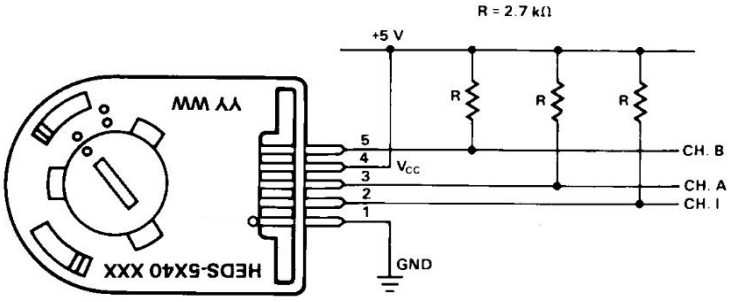

Figure 5-14: Schematic of pull-up resistors in the HEDS 5540 encoder connection circuit

5.4.5 Pull-Down Resistors

When a micro switch is pressed due to contact, the circuitry is closed, and a high voltage signal is sent to the controller's input pin. When the micro switch is released, the voltage signal is supposed to change from high (Vcc) to low (ground). The input pins of the controller may sometimes read the signals as high even if the micros switches are not pressed due to floating voltage on the pin or external interference.

Pull-down resistors are connected to the outputs of the micro switches and ensure that the input signals to the controller are pulled down to ground level when a micro switch is released. This prevents false signals being picked up in the controller. Figure 5-15 shows the schematic of the pull-down resistor used in the micro switch connection circuit.

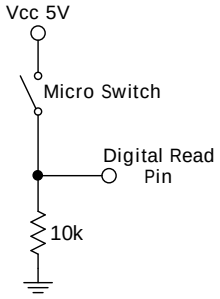

Figure 5-15: Schematic of pull-down resistor in the micro switch connection circuit

5.4.6 Power Supply Units

5.4.6.1 Power Supplies

Direct current power supply is required to power the electrical and electronic components. A Laboratory DC power supply unit which supplies 24V to the motor and 10V to the controller was used. The 10V supply to the controller was stepped down into the different voltages required. Batteries were used for the powering of the torque sensor.

5.4.6.2 Voltage Regulators

Positive voltage regulators were used for the step-down of voltages and for voltage protection. The voltage regulators used were the L78XX series with TO-220 terminals. 12V and 10V regulators were used as voltage protection for the torque transducer and the Arduino Uno boards respectively. The 10V

supplied to the controller was stepped down using a 6 V regulator, for the powering of the servos. The regulated 6 V was further stepped down to a 5 V supply source for the micro switches. The schematic for all the voltage regulator connections are shown in figure 5-16.

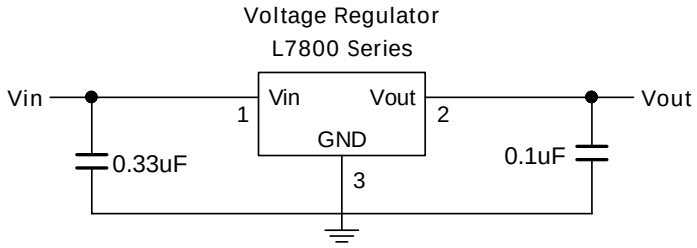

Figure 5-16: Schematic of voltage regulator circuit

5.4.6.3 Fuse Selections

Fuses from the fast acting glass tube fuse series were used as overcurrent protections. Current protection is essential as current spikes can damage electronic hardware. Fuses are used as a sacrificial device which breaks the circuit when its current limit is exceeded. Three fuses were used in the controller, namely a 6 A fuse for the protection of the motor (connected to the power supply of the H-bridge); a 2.5 A fuse for the protection of all the hardware using the 12 V power supply (the fuse rating was calculated using the sum of all the dependant hardware's operational current ratings), and a fuse (0.05 A) to the torque transducer current protection.

5.4.7 Controller Schematics

The detailed schematics of the controller are presented in Appendix C. The schematic diagrams were drawn using ProfiCAD electrical diagrams software. Figure 5-17 shows a simplified schematic of the controller together with all the electrical and electronic components used for the biped design.

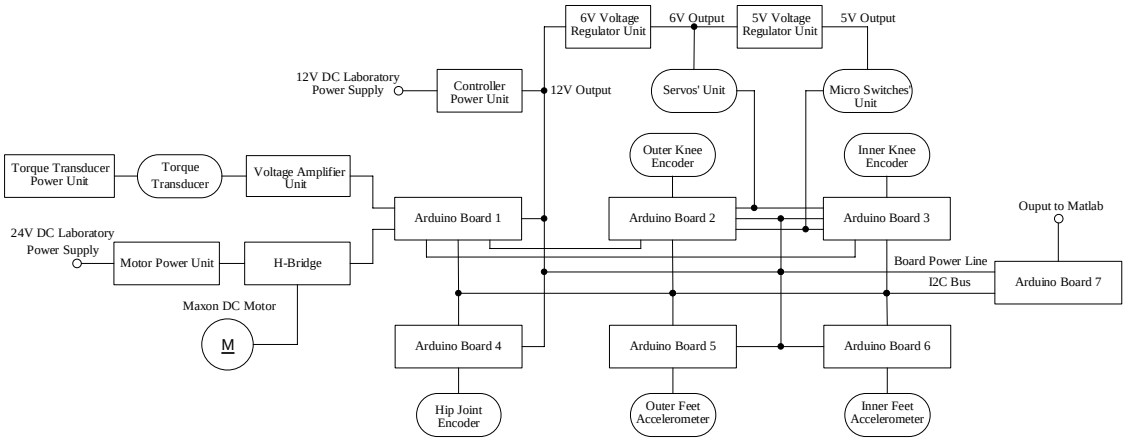

Figure 5-17: Simplified schematic of the controller

5.5 Software Programming

5.5.1 Controller Interface

Communication between the controller boards and Matlab was required. Signals from sensors are read by different Arduino boards and then sent to a master board. The data received from the slave boards are transmitted to Matlab, via a computer, for computation via the master board. All sensor data are stored on a computer through Matlab commands.

An I^2C bus was used as the interface device between the Arduino boards which have built-in Wire library communication commands. For communication between the master board and computers, Arduino boards utilise their serial ports or UARTs (universal asynchronous receiver/transmitters). Communication was set up between the serial port of the master board and the USB (universal serial bus) port of the computer with Matlab running on the computer. The baud rate used for all data receiving and transmissions was selected as 9600 bits per second.

5.5.2 Routined Control

The controller was programmed to execute routine walking (periodic cycles). The programmed routine was aimed at generating a sequence of walking steps so that the locomotion data could be captured and recorded for analysis. All the forces of the controllable actuators were programmed as constant values which also includes the torque actuation from the DC motor. The magnitude of the torque actuation required for walking was determined by a trial and error approach. A ramp with a slope angle of 4.4° (0.08 rad) and a length of 6 metres was used for experimentation. The code routines programmed on the Arduino boards are as follows:

Arduino board 1:

- Reads the voltage reading of the torques transducer (after amplification).
- Reads the heel strike voltage signals from Arduino boards 2 or 3. If a heel strike signal is detected, it turns the motor on for the set time period. The direction of the motor rotation is in accordance with the activation of the inner or outer feet's micro switch.
- Sends the torque transducer reading to Arduino board 7.

Arduino board 2:

- Constantly updates the outer knee encoder's pulse count value.
- Reads the voltage signals from the outer knee and inner feet's micro switches. If the outer knee switch is activated (knee strike occurrence), executes outer knee servos locking commands. If the inner feet switch is activated (heel strike occurrence), unlocks the outer knees and sends a voltage high signal to Arduino board 1.
- Sends readings from the outer knee encoder, inner feet switch, and outer knee switch to Arduino board 7.

Arduino board 3:

- Performs the same tasks as Arduino board 2 except with different sensors. The board 2 sensors (outer knee encoder, outer knee and inner feet micro switches) are replaced by board 3 sensors (inner knee encoder, inner knee, and outer feet micro switches) respectively.

Arduino board 4:

- Constantly updates the hip joint encoder's pulse count value.
- Sends the reading of the hip joint encoder to Arduino board 7.

Arduino board 5:

- Reads the x- and y-axes voltage values from the outer feet accelerometer.
- Sends the above readings to Arduino board 7.

Arduino board 6:

- Reads the x- and y-axes voltage values from the inner feet accelerometer.
- Sends the above readings to Arduino board 7.

Arduino board 7:

- Receives all sensor readings from boards 1 to 6.
- Transmits the received readings to Matlab in a set sequence.

Note that the above tasks of the controller boards were loop executed. The captured sensor data were transmitted and converted into generalised coordinates in Matlab. The Matlab and Arduino boards' programme codes are presented in Appendix D.

5.6 Control System

A control system is required to control the actuator torque in experiments. The feedback system of the biped is shown in figure 5-18. The closed-loop diagram consists of two loops, an inner and an outer. The inner loop consists of the controller which controls actuation torque of the plant (biped system). The torque transducer is used as the feedback sensor to the inner loop. The outer loop consists of the Matlab system which controls the setpoint of the inner loop. Matlab calculates the control parameter (hip actuation torque u_2) using feedback signals from encoders and accelerometers (generalised coordinates). The chaos control algorithms are used to calculate the control parameter. The determined control parameter is the inner loop setpoint or the controller's reference value.

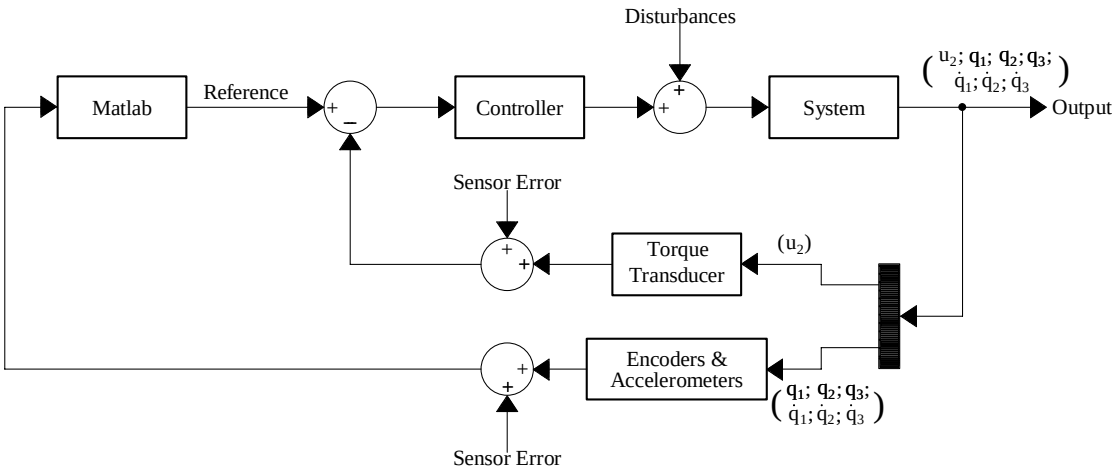

Figure 5-18: Block diagram of closed-loop feedback system of the biped

The aim of the control system programmed into the controller is to control the hip actuation torque according to the reference value. The design of a control system which is capable of controlling the biped model is complex. The biped model is a multi-body system whose transfer function varies according to configuration changes. Control systems are normally designed according to fixed transfer functions. This means that the control system of the biped needs to adapt to transfer function changes. A possible solution is to determine the transfer function range of the biped model and then design a control system/s for that range. A control system for the multi-body biped model was not designed due to the difficulties discussed above.

5.6.1 Hypothesised Experimental Chaos Control

Chaos control of the biped using hypothesised experimental conditions was simulated. Simulations of the closed-loop biped system shown in figure 5-18, using model parameters of table 4-3, were conducted. The simulated slope angle and starting condition were the same as the ones used in the Period One UPO Control section of Chapter 4. The parameter model and slope angle conditions were selected so that the results can be compared. The simulation results of Chapter 4 are seen as ideal whereas the simulation results in this section are closer to reality.

The difference in conditions between the simulation of Chapter 4 and the hypothesised experimental simulation is that the dependent variables (equation 3.27 and equation 3.35) used in the chaos control algorithms are the actual values with added percentage errors due to sensor errors. With the added sensor error, there will be a finite distance in phase space between the actual and measured dependant variables. Calculation of the perturbation force (u_2) using the chaos control algorithm, is based on the measured dependant variable. In short, depending on the magnitude of the sensor error, the calculated and the required perturbation control differs. Another difference is that the execution of the response torque (u_2) is not ideal. An ideal response for the biped is when the control parameter converges to the required magnitude instantaneously like a reference step. The actuator response is simulated as typical response behaviours which have shortcoming such as delays, rise time and settling time.

Typical errors or offsets that occur in real system control process are implemented in simulations. The feedback error from sensors is due to disturbance and sensor resolution error. Disturbance error is normally a result of unpredictable external forces. Sensor resolution error cannot be avoided as there are always limitations to the sensor's quantity of measurement. The sensor resolution error was simulated by adding pseudo-random numbers of standard uniform distribution to the sensor feedbacks. The added error is a percentage (4% or less) of the generalised coordinate's actual value. It was assumed that the sensor feedback errors do not accumulate.

A shortcoming in all control systems is the deviation between the controlled parameter and its reference value. It was assumed that the controller was capable of controlling the plant. The unit step response (step torque response) of the inner loop was assumed to behave as in figure 5-19. The step response has the following properties: Its response delay is 0.050 s due to controller delays. This response time is larger than most typical controller delay times. The rise time of the response is 0.104 s (the time from 10 % to 90 % step height). The settling time is 0.156 s (10 % error band) and has an overshoot of 0.2 %.

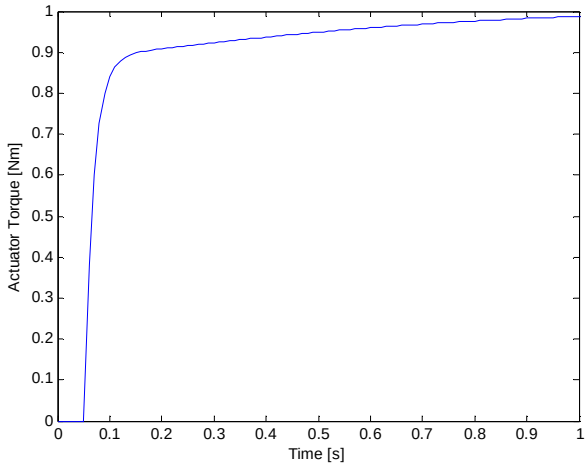

Figure 5-19: Hypothetical unit torque step response of the biped model

The results from the simulation of hypothesised experimental control are: figures 5-20 and figure 5-21 shows the r_n distance and perturbation torque plots of the controlled UPO orbit. Figure 5-20 shows that the state point of the orbit wanders in the neighbourhood of the fixed point but never converges onto it. Figure 5-21 shows constant perturbation control in the attempt to maintain the period 1 UPO cycle. Further simulations showed that a sensor resolution error of 5 % or larger results in uncontrolled chaotic orbits (in most trials). If the sensor feedback error accumulates, chaos control will not be possible.

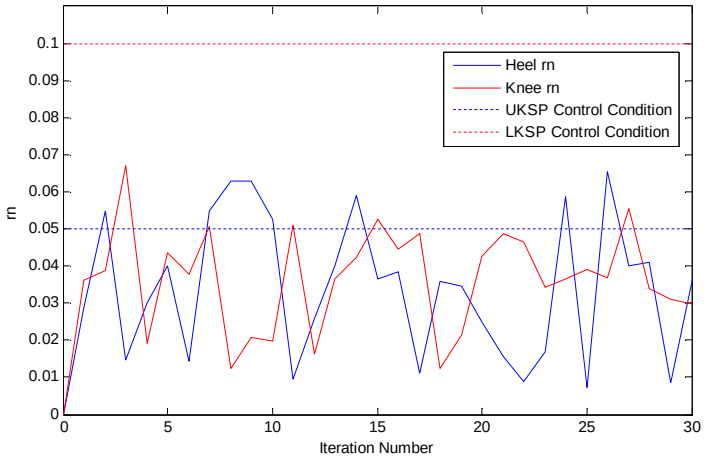

Figure 5-20: Plot of r_n distance versus iteration number as a function of iteration number using hypothesised experimental conditions

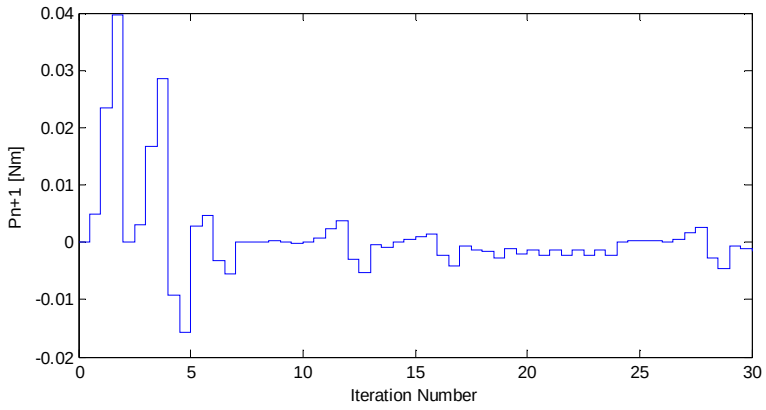

Figure 5-21: Plot of perturbation torque as a function of iteration number using hypothesised experimental conditions

The purpose of simulating chaos control of the biped using approximated experimental conditions was to determine the performance requirement of the actuator, controller, and sensors, for successful control. The magnitude of the percentage error added to the sensor feedbacks approximates the resolution required of the sensors. The response properties of the control system depend on the capacities of the actuators, system load, and control system. Although simulations were not conducted on the variation of the controlled parameter's response properties, simulations of the above can be used to determine the designed control system's effectiveness and the requirements of the actuators' performance.

5.7 Chapter Summary

A biped that meets most specification requirements was designed. All the major components of the biped were presented and discussed in this chapter. Most hardware selections were based on availability and ease of implementing control. An overview of the mechanical segments was presented without going into the details of structural design process. Controller boards developed by Arduino were selected because of its ease of coding language which is based on C++. Programming was made simple as working library codes were available. A controller was developed which is capable of controlling all actuators and also records measured signals when used in conjunction with Matlab. Matlab was used to implement the chaos control algorithms by processing received data from the controller. Due to the complexity of designing a control system for a multi-body model, chaos control of the biped could not be conducted experimentally.

CHAPTER 6 TESTING AND RESULTS

6.0 Introduction

Experimentation of a biped was conducted for the purpose of performance evaluation and modelling verifications. It is common practice to determine the difference between the design capability and actual functionality. If the constructed devices do not work according to plan, the test results normally indicate the area of deficiency. Improvements are usually easier made after visualising the execution of the designed object. It is the repeated process of testing and redesign that turns a good design into a great one.

This chapter presents the testing, results, and discussions of the manufactured biped. The system behaviour of the designed biped was first categorised against the walking slope angle which determines the range of slope angle for which testing could be conducted. The hardware components of the biped were tested for effectiveness and its shortcomings and deviations from the expected specifications are discussed. An evaluation of the controller and its software was conducted and the results from walking experimentations were compared with theoretical simulations. Similarities in the result trends are described and the differences explained. Concluding remarks are made regarding the biped and its model.

The isometric view of the assembled biped is shown in figure 6-1. More photos of the assembly from different viewpoints can be found in Appendix E.

Figure 6-1: A photo of the biped

6.1 Bifurcation Diagram and Lyapunov Exponent Graph

The bifurcation diagram (figure 6-2) and the Lyapunov exponents graph (figure 6-3) of the biped were simulated for categorisation of the system behaviour. Simulations were conducted to determine the range of slope angle for which system dynamics is periodic or chaotic. The simulated results show that stable walking is possible when the slope angle $\gamma \in [0.021, 0.093]$ rad, where orbits are chaotic within the slope range of [0.087, 0.090] rad. Typically, stable walking is not possible in experimentations for slope angles of low magnitudes ($\gamma < 0.052$ rad or 3°). As the slope angle decreases, the average ground clearance of walking also decreases which results in the biped being sensitive to disturbances and ground unevenness.

Figure 6-2 (a) shows that the designed biped does not generate passive limit cycles for all slope angles. Orbits are of period 2 for $\gamma \in [0.021, 0.072]$ rad. Reason for the non-occurrence of limit cycles is due to the difference in moment of inertias between the left and right legs (inner and outer leg pairs). The difference in moment of inertias is the result of the coupling of the hip segment to the outer leg pair. Limping, which is a gait of long and short steps, occurs due to the difference in the legs' inertia. The simulation results in previous chapters showed limit cycles because it was assumed that the hip segment rotates with the stance leg. Figure 6-2 (b) shows that at $\gamma = 0.076$ rad, the long-term walking orbit changes from a period 4 to a period 2 cycle. At $\gamma = 0.076$ rad, the period 4 cycle bifurcates into chaotic orbits which then becomes period 2 cycles. The above three changes of system behaviour occur in a very narrow range of γ thus making it appear like a sudden change.

Figure 6-3 shows the simulated Lyapunov exponent graph of the biped. Figure 6-3 (b) shows frequent occurrences of periodic windows within the chaotic region which is indicated by the oscillation of the Lyapunov exponent values between positive and negative values. Frequent changes between periodic and chaotic behaviour mean that the system is sensitive to slope angle changes. It makes chaos control difficult if the system varies between periodic and chaotic nature from small γ changes.

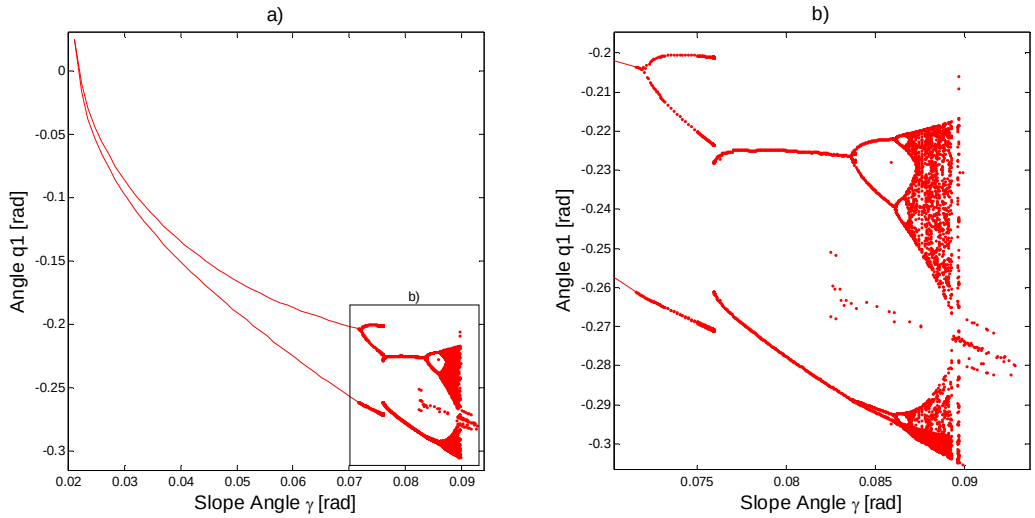

Figure 6-2: Simulated bifurcation diagrams of the biped - a) Entire slope angle range of stable walking, b) Magnified view of chaotic regions

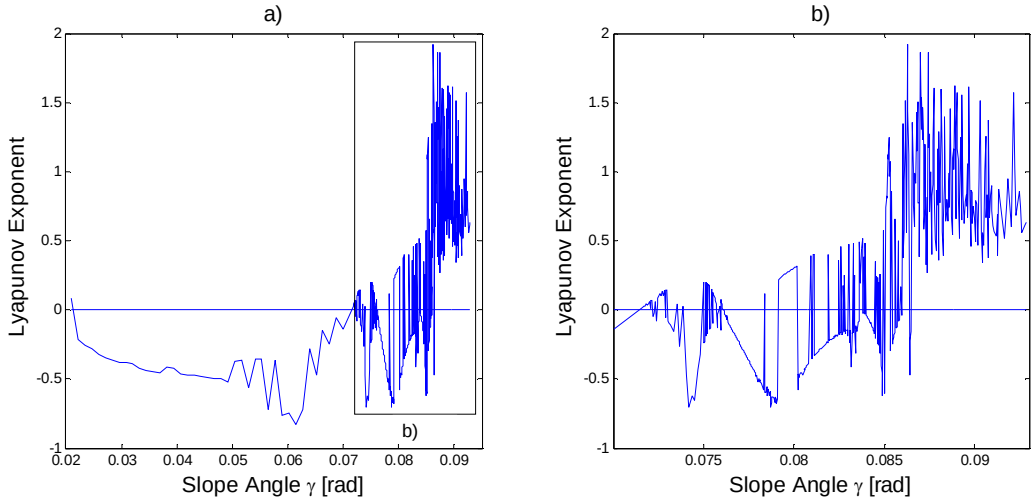

Figure 6-3: Simulated Lyapunov exponent graph of the biped - a) Entire slope angle range of stable walking, b) Magnified view of chaotic regions

6.2 Hardware Testing

6.2.1 Motor and Torque Transducer Testing

6.2.1.1 Test Set-up

Both active and passive tests were conducted on the control of the DC motor with the biped suspended in the mid-air about its hip segment. The outer leg pair of the biped was fixed in a vertical position and the inner leg pair was free to move. All initial conditions start with the leg pairs vertically suspended (hip joint angle = 0°). The knees remained locked in testing.

Data of the control process were measured using the hip joint encoder and torque measuring unit. The torque measuring unit consists of a torque transducer, a voltage amplifier and an Arduino board. The measured data were sent to Matlab where it was processed and plotted.

6.2.1.2 Control Routine Test

A control routine of the motor was conducted where the hip joint angle and torque were recorded. The control routine, similar to the experimental walking control routine, executed continuous forward and backward leg swings. Since the outer leg pair was stationary fixed, only the inner leg pair was able to rotate according to actuations. The H-bridge of the motor was supplied with 15 V. All forward and backward swings were carried out in three stages: The first stage turns the motor on with 100 % power (15 V). For forward and backward swings, the motor is turned on for 0.115 s and 0.135 s respectively. The difference in motor actuation time between the forward and backward rotations is to compensate for the difference in moments of inertia between the leg pairs. The above were attempts to generate limited cycle walking through controlled actuation. The second stage brake was executed on the motor for 0.01 s. The third stage left the motor uncontrolled for two seconds. Thereafter the direction of the rotation changed and the three stages were repeated.

Data of five leg swings, with 3 backward and 2 forward, are shown in figure 6-4. Figure 6-4 (a) shows that the forward and backward swing peak torques are 5.90 Nm and -7.79 Nm respectively. The magnitudes and signs of the measured torque values respectively indicate the time and direction that the motor was turned on. The figure also confirms that the backward torque is larger than the forward torque, due to a difference in actuation time, from the comparison of peak joint angles (see figure 6-4 (b). The smaller torque rises after the torque peaks because of the inner leg swinging back to its neutral position (hip joint angle = 0°) during the time period of the third stage. Figure 6-4 (b) shows the evolution of the hip joint angle as a result of the actuation torques. Figure 6-4 (b) shows that the inner leg pair does not return to its neutral position after every actuation swing due to the friction forces in joints and the partly non-back-drivability of the gearbox.

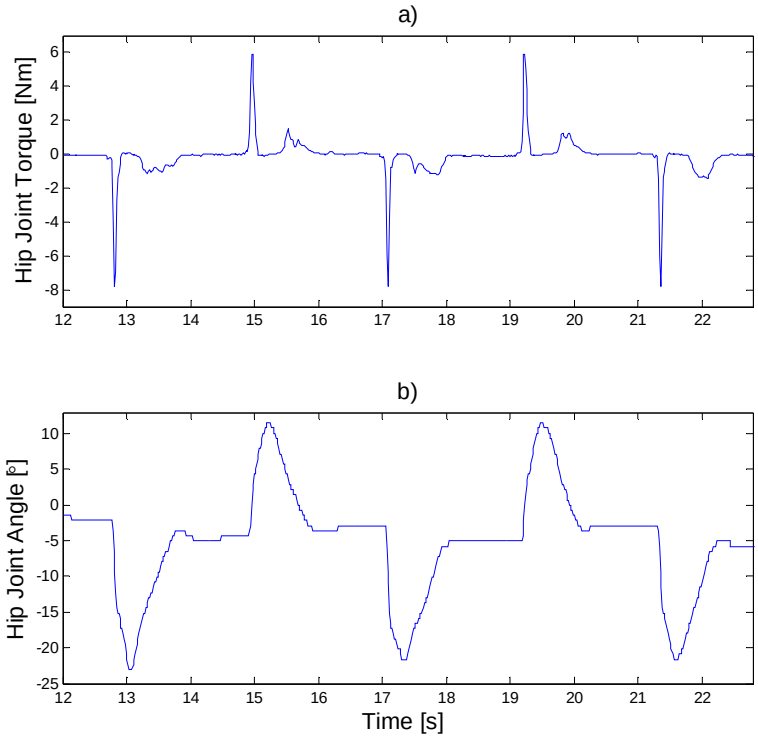

Figure 6-4: Control routine results as a function of time – a) Hip joint torque, b) Hip joint angle

6.2.1.3 Passive Leg Swing Test without Actuation

Testing was conducted on the passive swinging of the legs. The inner leg was rotated by hand to the minimum hip joint angle (-37.44°) and then released. The recorded hip joint torque and angle are shown in figure 6-5.

Figure 6-5 (b) shows that the inner legs were rotated from 0° to -37.44° between the times of 11.39 s and 12.92 s. The inner legs were held in position for some time and then released. The inner legs oscillated like a pendulum between 14.06 s and 16.58 s and settled after that. Figure 6-5 (a) shows the torque between the inner and outer leg pairs during testing. The peak torque (-3.761 Nm), which occurs at the immediate moment of release, gives an indication of the magnitude of the retarding forces acting within

the hip joint. The magnitude of the peak torque also depends on the release altitude of the inner legs. As a result of all the retarding forces acting at the hip joint, the inner legs swings were far from passive.

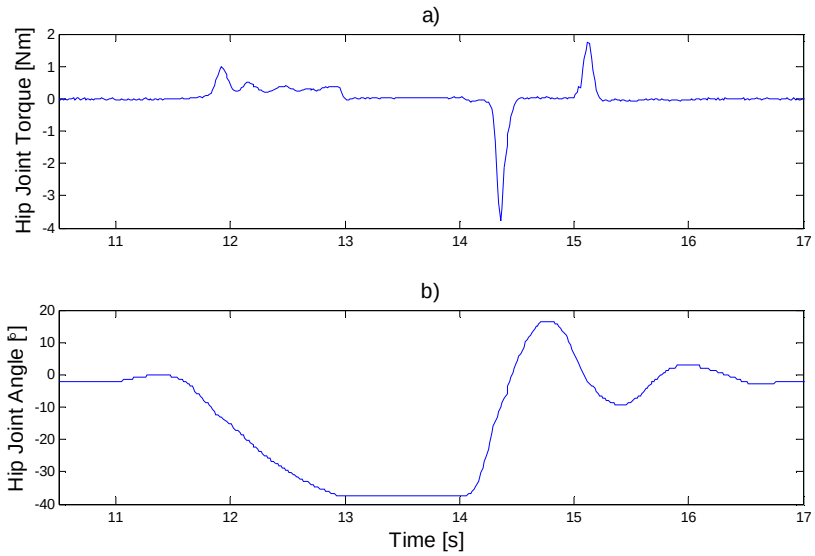

Figure 6-5: Unactuated passive leg swing as a function of time – a) Hip joint torque, b) Hip joint angle

6.2.1.4 Passive Leg Swing Test with Actuation

In order to overcome the retarding torques that occur at the hip joint, motor actuation was implemented. The retarding force of the hip joint was mostly due to the partly non-back-drivability of the gearbox (reduction ratio of 73.5:1). The motor was supplied with just enough power to overcome the frictional forces in the gearbox.

Readings from the torque transducer determines the direction and the on or off status of the motor. The sign of the torque reading decides the direction of the motor rotation. The motor was turned on only when the absolute magnitude of the torque reading was greater than 0.028 Nm. The above minimum torque limit was implied to compensate for noisy readings. Even with zero torque acting on the torque transducer, the signal reading oscillates within absolute peaks of 0.28Nm due to sensor noise. Without the minimum torque limitation, the motor vibrates vigorously, without stopping. When the motor was turned on, its supplied voltage was always 15V. No voltage supply control was implemented because performance without control was acceptable.

Figure 6-6 shows the results of leg swings with actuation. The testing procedures were the same as for the previous section. After the release of the inner legs at 21.69 s, the legs oscillated and then settled at time 27.39 s (see figure 6-6 (b)). The above oscillation time span lasted for 5.70 s (the previous section had a time span of 2.52 s). With actuation, the inner legs oscillated like a pendulum which slowly came to a stop due to frictional forces. Figure 6-6 (a) shows that the experienced hip joint torque is contained within the range of [-0.06, 0.06] Nm. The above torque peak is 50 times smaller than the peak of unactuated passive leg swings. The boundary limits of the actuated hip joint torque range are governed by the resolution and noise sensitivity of the measured torque readings.

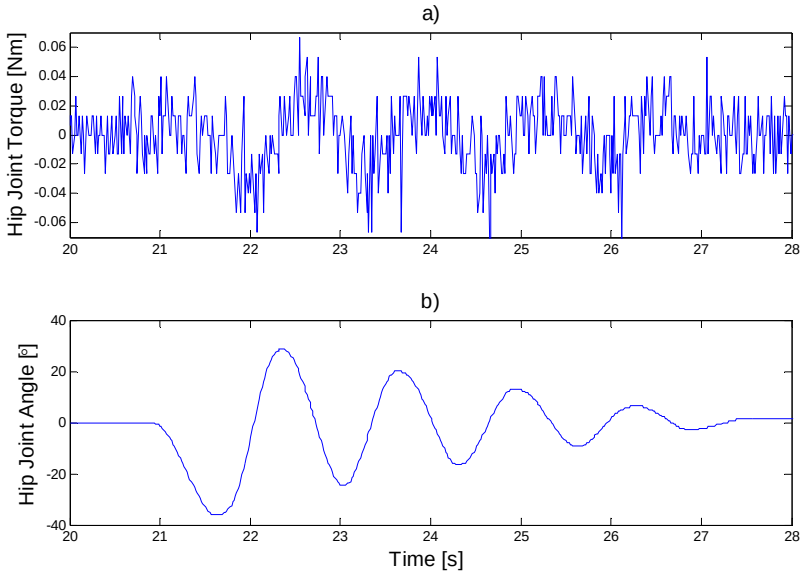

Figure 6-6: Actuated passive leg swing as a function of time – a) Hip joint torque, b) Hip joint angle

6.2.1.5 Summary of Motor and Torque Transducer Testing

The test results from the previous sections shows that the control of the motor was successful in terms of rotational direction and timing. Direction control of the motor using Arduino Uno via the H-bridge was effective. No motor failure or malfunctions were detected after repeated use. This implies that the snubber design works to protect the motor against voltage and current spikes. The torque measuring unit (torque transducer and its output voltage amplifying unit) also functioned according to design. The geared motor was made easily back-drivable through feedback signals from the torque sensor and actuation control. Back-drivability of the geared motor was required for experimental passive walking. Control and testing of the torque magnitude was not conducted due to the multi-body difficulties discussed in the previous chapter.

6.2.2 Knee Lock Mechanism Testing

The components involved in the knee lock mechanism tests were: knee and heel strike micro switches, servos, and Arduino Uno control boards. The test results show that signals from the micro switches (heel or knee detections) are captured every time without fail. The only times that strike contact was not detected was when the switches were half pressed. This occurred when the legs failed to fully straighten at knee strike.

Figure 6-7 and figure 6-8 show how the micro switches are mounted on the biped. In figure 6-7, the image shows the straightening of the knee where the knee cap presses on the roller lever of the micro switch. Figure 6-8 (a) shows the mounting of the heel strike contact detection plate and the micro switch on the side of the foot. Figure 6-8 (b) gives a close-up view of the contact detection plate pressing against the micro switch's roller lever.

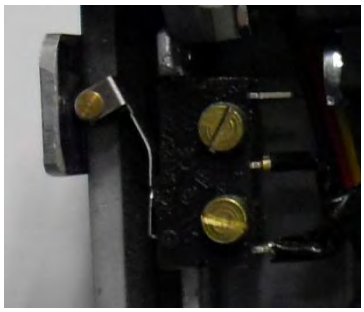

Figure 6-7: Knee strike detection micro switch

Figure 6-8: a) Heel strike detection micro switch, b) Close up view of the micro switch

Figure 6-9 shows the locking and unlocking positions of the knee lock mechanism. The locking arm undergoes 25° rotation between the locking and unlocking positions. In the experiments, the servos' speed of rotation responded to knee strikes of all the magnitudes without fail. A fast knee strike means that the servo needs to react fast before the knee cap bounces out of the locking range. During walking experimentations, the servos may sometimes not release the locks in time due to controller delays. The delays are a result of the controller multi-tasking, for example also transmitting data during the experiments.

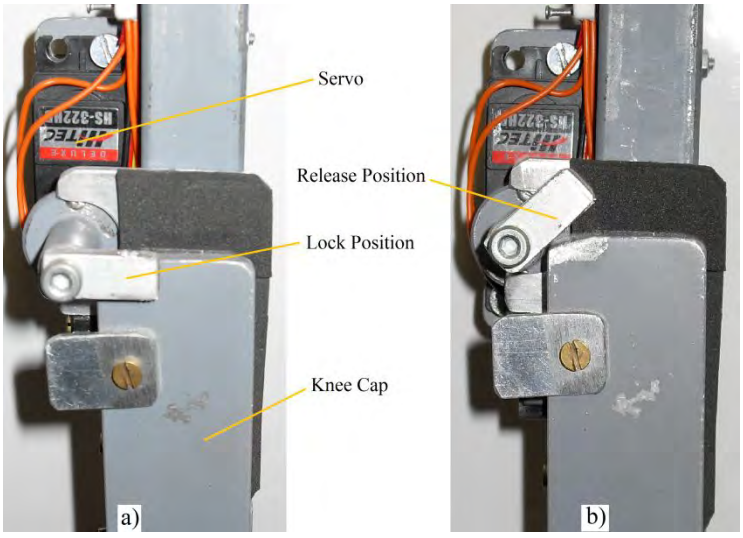

Figure 6-9: Front view of the knee lock mechanism – a) Locking arm in its locking position, b) Locking arm in its release position

6.2.3 Optical Encoder and Accelerometer Testing

The optical encoders and accelerometers were tested to check the difference between actual and measured angles. The mounting positions of the encoders and accelerometers are as shown in figure 6-10 and figure 6-11 respectively.

6.2.3.1 Sensors' Reference Lines

The reference lines of measuring for the above sensors are described below.

*Hip joint encode*r:

The hip joint encoder measures the acute angle between the inner and outer thighs. When the thighs line up, the measured angle is 0°. When the outer thighs are at the back and the inner thighs at the front, the measured angle is positive and vice versa. Due to mechanical constraints, the hip joint encoder's measuring range is [-45.36°, 45.36°] which corresponds to a pulse count range of [-63, 63] pulses. Due to the clock speed limitation of the Arduino Uno, only the rise edges of the pulses are counted for all optical encoders (resolution of 0.0126 rad).

Inner leg knee encoder:

This encoder measures the right inner knee angle. Since the thighs and shanks of the inner legs are mechanically coupled, measuring one knee angle measures both. The straightening of the knee corresponds to a knee bend angle of 0°. Since the shank is mechanically constrained not to hyperextend, the measured range is [0°, 90°] which corresponds to a pulse count range of [0, 125] pulses.

Outer leg knee encoder:

This encoder measures the left outer knee angle. The measuring conditions of this encoder are the same as the inner leg knee encoder. The measured range and pulse count range is also the same.

Inner leg accelerometer:

This accelerometer measures the angle between the left inner shank and the horizontal ground. When the shank is parallel to the normal of the ground, the measured angle is 0°. When the shank pitches backward, the measured angle is positive and vice versa. The typical measuring range of the accelerometer falls within [-45°, 45°].

Outer leg accelerometer:

This accelerometer measures the angle between left outer shank and the horizontal ground. The measuring conditions are the same as the inner leg accelerometer.

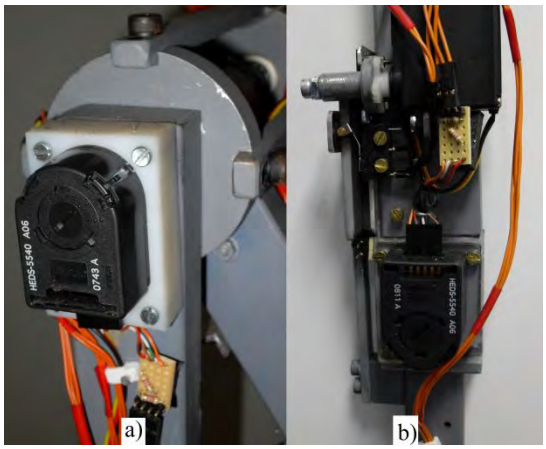

Figure 6-10: Optical encoder mounting positions – a) Hip joint encoder mounting, b) Knee joint encoder mounting

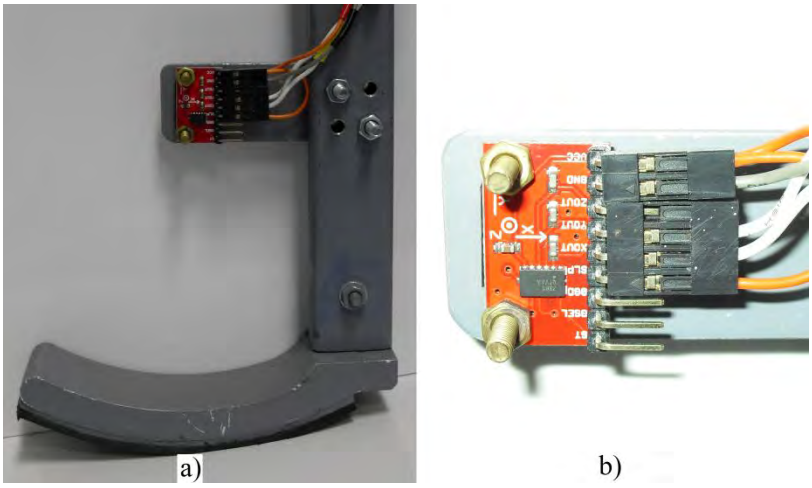

Figure 6-11: a) Accelerometer's mounting position, b) Accelerometer's measuring axes orientations

6.2.3.2 Hip Joint Encoder and Accelerometer Testing

A test was conducted to compare the measured angles using the hip joint encoder and accelerometers. The biped was suspended about its hip segment and all four knees locked for the duration of the experiment. The inner and outer leg pairs were rotated and the corresponding angles were recorded. From geometry, the relationship between the hip joint angle, and the inner and outer shank angles (angles measured using the accelerometer) are as follows:

$$Hip\ joint\ angle = Inner\ shank\ angle - Outer\ shank\ angle \qquad (6.1)$$

The inner and outer leg pairs were rotated back and forth which generated the results in figure 6-12. Figure 6-12 (a) shows the recoded hip joint angles measured using the hip joint encoder. It can be observed that the paired legs were initially lined up (hip joint angle = 0° at time = 5 s). The leg pairs then underwent processes of separation, stationary held, and reverse separation repeatedly. Figure 6-12 (b) and (c) shows the measured angles of the inner and outer shanks respectively. It can be seen that the accelerometers' measured angles spike at the first moments when the legs are held stationary. The spiked

readings are a result of the shock transmitted to the accelerometers due to impact between the inner thighs and the hip segment bars. Also to be noted is that accelerometer readings are accurate in the long-terms and less accurate in short-terms due to local errors. Figure 6-12 (d) shows a comparison of the hip joint angles from the hip joint encoder measurements and the combined results of the accelerometers (using equation 6.1). Angle measurements from the hip joint encoder produced results with better accuracy. The accelerometers' measured readings follow the same trend as the encoder's measured values without delays. However, the accelerometers' measurement deviates from the encoder, in terms of magnitude, between the range of [-5°, 8°]. This difference is due to the inaccuracy of the accelerometer specifications determined in Table 5-5.

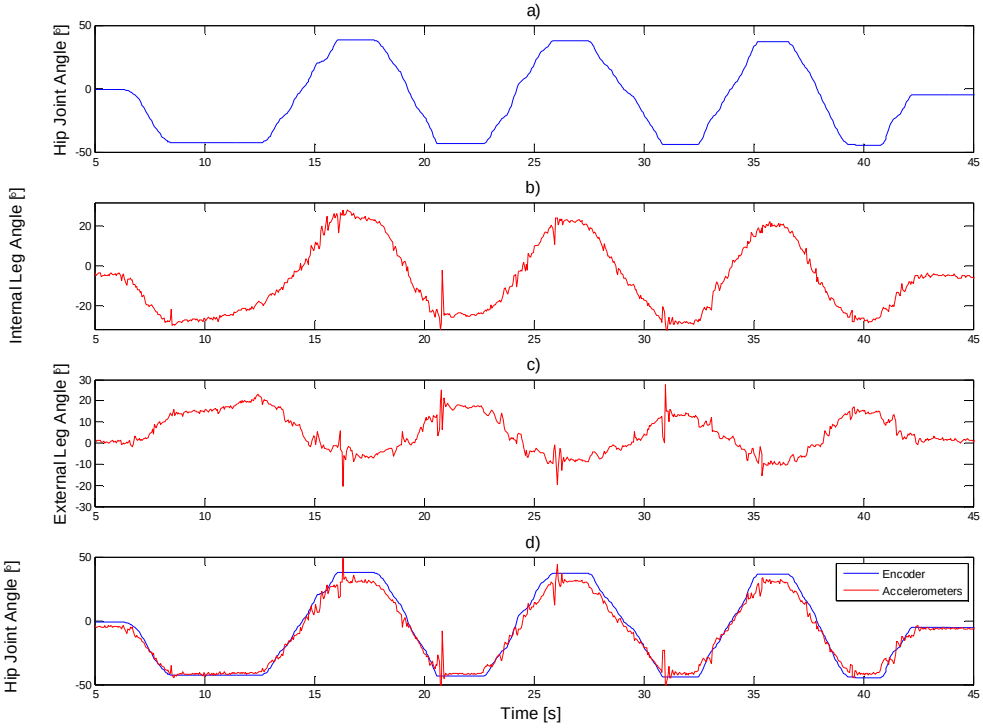

Figure 6-12: Recorded testing angles – a) Encoder measured hip joint angle, b) Inner shank angle, c) Outer shank angle, d) Comparison of hip joint angles

6.2.3.3 Summary of Optical Encoders and Accelerometers Testing

The testing results of the optical encoders and accelerometers show that both types of sensors can produce inaccurate readings depending on the situation. Typically, encoder readings have accumulative errors and accelerometer measurements contain local errors.

Tests have been conducted on the encoders to verify its precision (results not shown). The encoder shaft was rotated back and forth starting from a known position (such as the straightening of the knee). When the encoder shaft was rotated back to the original position, the pulse count should also return to the initial value. After repeated tests, the results showed that deviation from the initial pulse count increases when the encoder shaft undergoes more rotations. This implies that errors in encoder readings are accumulative. The cause for the encoder error is due to its code wheel vibrations. Pulses are generated when the code wheel within the encoder rotates. The code wheel is fixed to the encoder shaft and not the encoder. If the

encoder shaft undergoes radial play, pluses are generated even if the shaft does not rotate. The encoder shafts, as well as the joint shaft, experience radial play as a result of bearing play or mechanical lashes due to tolerance gaps. If the code wheel is free from vibrations and radial plays, pulse counts remain precise.

The results in figure 6-12 (d) show that the errors of accelerometers are local errors and do not accumulate. Errors of the accelerometer are due to inaccurate performance specifications and their over-sensitivity to shock vibrations.

The errors of encoder readings may be small but will remain unknown. The accelerometer's measurement errors can be approximated but are too large. In summary, the testing results indicated that the chosen sensors are capable of capturing the trend of the results and not their magnitudes within the required accuracy.

6.3 Software Evaluation

6.3.1 Routined Control Evaluation

The routine control of section 5.5.2 involves the programming of Arduino Uno boards for the execution of a sequence walk and data acquisition. Tasks such as sensor reading, actuator control and signal transfer are performed by different boards. All the boards carried out the tasks which were assigned to them as coded. No technical problems arose due to the programmed codes. The outcome of the control routine are discussed in section 6.4.1

6.3.2 Data Acquisition Evaluation

Measured data was transferred from one Arduino board to Matlab via a serial port connection. The selected transfer baud rate was 9600 bits per second. Faster baud rates (14400, 19200, 28800, 38400, 57600 and 115200 [bits per second]) are available but slower baud rates provide less transmission errors.

The data acquisition process takes place in real time. As the data is read, it is sent to the master Arduino board which transfers the data to Matlab. It typically takes about 1 s to start up the connection between Matlab and the master Arduino board. A result from a routine walk experiment recorded 777 sets of data in 41.885s. Each set of data contains 13 different sensor readings of a specific time moment. This means that 10101 readings were received in 41.885 s which corresponds to $\frac{10101}{41.885} = 241.16$ readings per second or 0.0041s per reading. The sampling rate of any specific sensor reading is calculated as $\frac{41.885}{777} = 0.054$ s per sample. The average sampling rate is about 0.055 s. The baud rate is considered fast for the transmission of any single reading (0.0041 s per reading). However, the baud rate is considered slow in terms of average sampling rate (0.055 s). The effectiveness of a control system depends on the time of the sampling rate. If control setpoint values are to be generated by Matlab (refer to figure 5-18), Matlab will need at least 0.055 s before it can respond. The slower the sampling rate, the longer the respond delay.

The selected data acquisition system is capable of real time control. This is to control the biped while it is in motion. This requires an effective torque controller which still needs to be investigated. Real time

control is possible due to the capability of real time data acquisition and concurrent Matlab setpoint calculations. A higher baud rate needs to be selected to achieve the above process since a faster sampling rate is required.

6.4 Walk Testing

Testing of the biped was carried out for two reasons. Firstly, to determine whether the constructed biped walks or not. Secondly, to validate the derived biped model with the actual.

6.4.1 Walking Experiments

Walking experiments of the biped were conducted on a 6 m ramp of slope angle 0.08 rad (4.4°). The programming codes described in section 5.5.2 were used for control of the biped. The biped walked on average 9 consecutive steps down the ramp before coming to a stop at the ramp end. Figure 6-13 shows the video sequence frames of two consecutive walking steps. The figure also included the start-up procedure for walking. A description of each sequence frame follows:

Frame *a*: This is the start up orientation of the biped. Initially the inner legs are forward with its feet lifted above the ramp.

Frame *b*: The inner legs are swung backward.

Frame *c*: The inner legs together with the hip are swung forward. This is to generate the forward momentum of the biped.

Frame *d*: The inner knee locks followed by inner feet heel strike. The heel strike triggers the inner foot micro switch followed by hip actuation.

Frame *e*: The outer legs swing forward due to actuation.

Frame *i*: The outer legs swing past the inner stance legs without foot scuffing.

Frame *g*: Outer knee strikes occur and the knees are locked.

Frame *h*: The outer legs continue to swing forward after knee strike until heel strike occurs. The heel strike triggers the outer foot micro switch followed by hip actuation.

Frame *i* to *l*: Frames *i* to *l* shows a walking step where the inner legs act as the swing leg. Frames *i* to *l* undergo similar routines as frames *e* to *h*. The differences are the locking of the inner knees and inner foot heel strikes. At frame *l*, the biped has completed one walking stride and the gait sequence begins from frame *d* again.

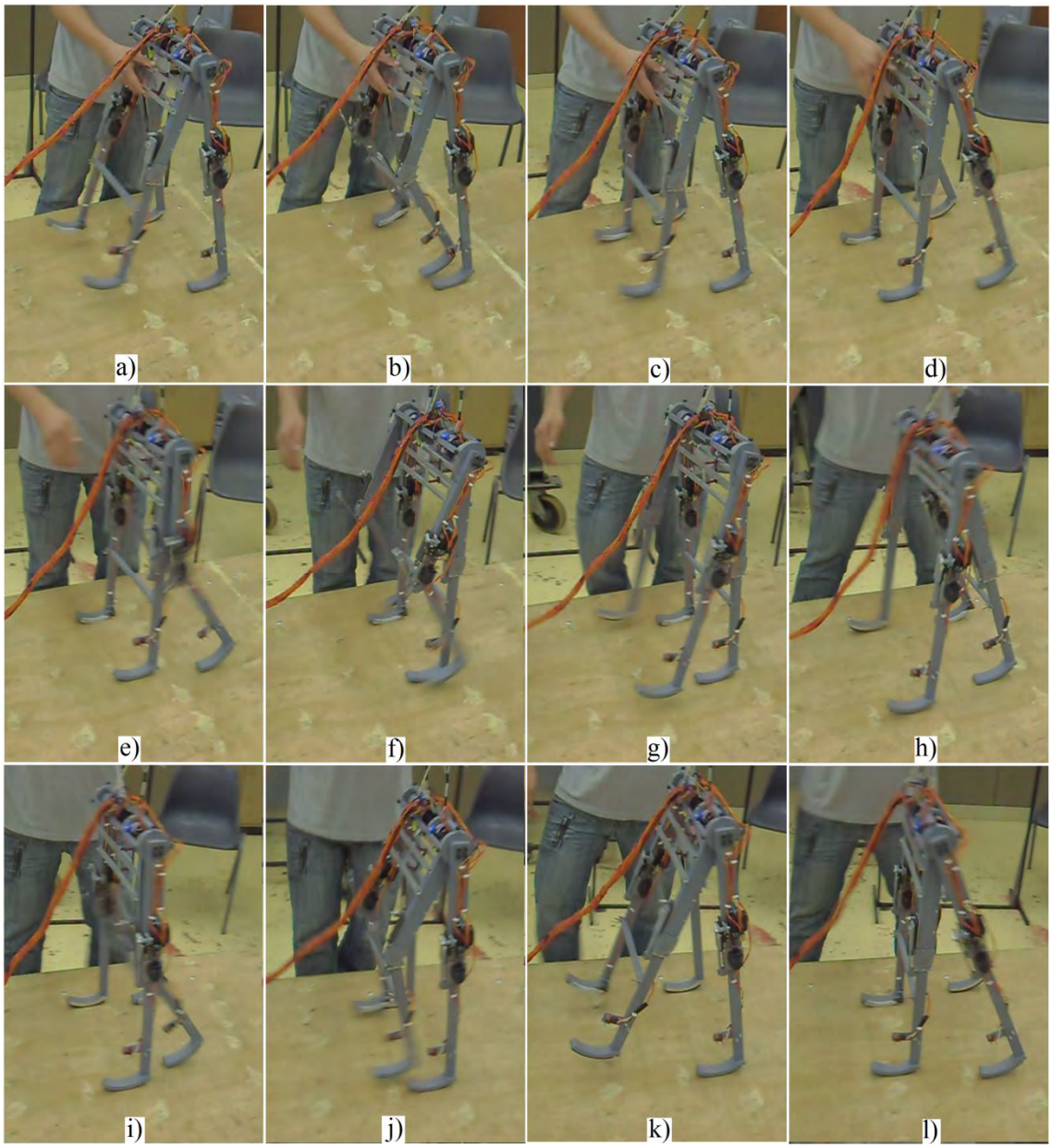

Figure 6-13: Sequence frames of two consecutive walking steps

The experimental results for 8 consecutive steps are shown in figure 6-14 and figure 6-15. Figure 6-14 shows the measured torque and angles while figure 6-15 showing the corresponding strike occurrences. The reference line for the measured angles of figure 6-14 is described in section 6.2.3.1. Figure 6-14 (a) shows the measured hip joint torque with maximum and minimum peaks of 3.81 Nm and -3.526 Nm respectively. Theoretically the biped is able to walk down the ramp without hip actuation. The constructed biped required actuation in order to increase walking ground clearances. Without actuation, the passive ground clearances were low and sensitive to disturbances which results in foot scuffing. The hip joint angle of figure 6-14 (b) oscillated between the range [-39°, 31°] which indicated that the steps of strides were not symmetric. Figure 6-14 (c) and (d) shows the times that the inner and outer knees experienced knee bends respectively. The inner and outer knees bend according to inner and outer leg swings. The shanks typically rotate between 0° and 60°. Figure 6-14 (e) and (f) shows the inner and outer shank angles respectively. It can be seen that these angles are out of typical operational range and large

reading spikes occur throughout the recording. These angles are incomprehensible due to constant transfer of shock vibrations to the accelerometers. The conversion of all the measured angles to the generalised coordinates is dependent on readings from the accelerometer. Since the accelerometers' readings are incomprehensible, no conversion could be made.

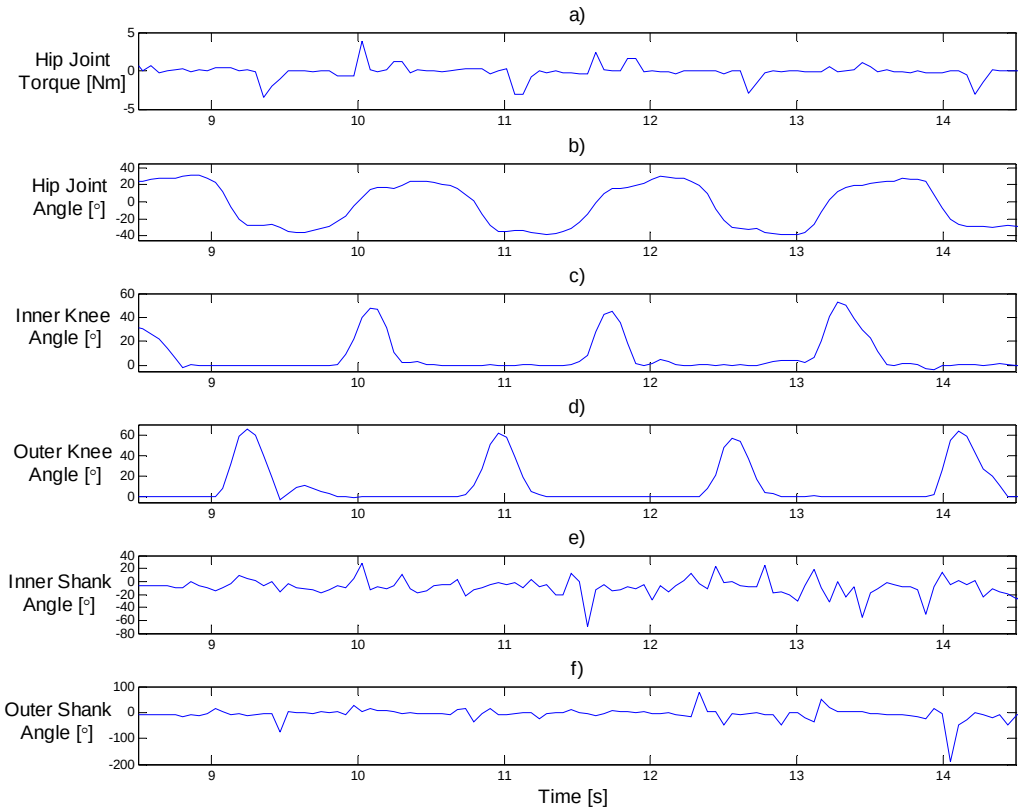

Figure 6-14: Sensor measured data as a function of time – a) Hip joint torque, b) Hip joint angle, c) Inner knee angle, d) Outer knee angle, e) Inner shank angle, f) Outer shank angle

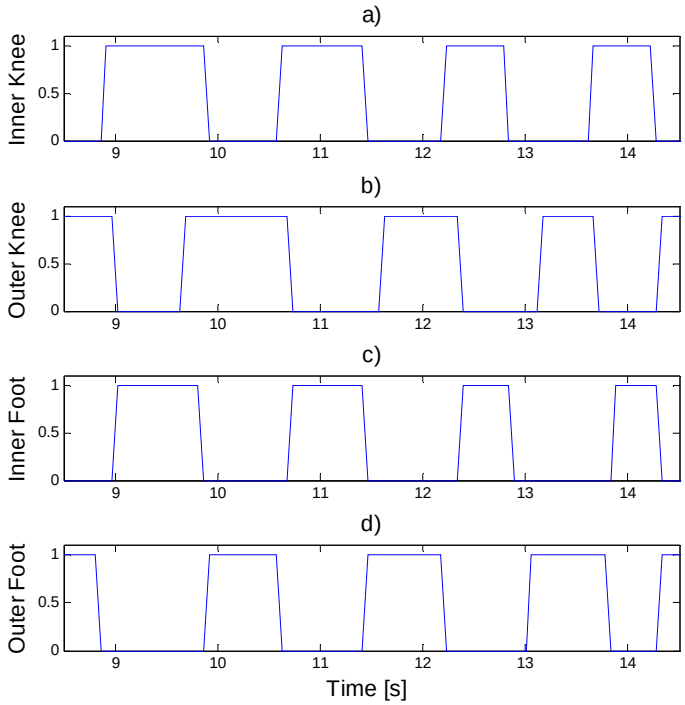

Figure 6-15: Strike detections as a function of time – a) Inner knee, b) Outer knee, c) Inner foot, d) Outer foot

Figure 6-15 shows the times when the micro switches are activated. Activation of the micro switch being pressed corresponds to an output value of 1, and 0 for switch releases. The knee switches are activated when the knees are locked and released during leg swings. The foot switches activate during the stance period and at the moment of heel strike. The activation and release of the switches synchronise with the biped trajectories of figure 6-14. The activation and release periods of the switches are not constant for two reasons: Firstly, there will be period differences between inner and outer leg swings. Secondly, the walking steps do not repeat in size or form a pattern.

6.4.2 Walking Simulations

Walking simulations were conducted using the experimental slope angle and the designed biped parameters. This was an attempt to compare the trends of experimental and theoretical locomotion. The simulated generalised coordinates were converted to the range of the sensor measurements since the reverse process was not possible. Initial conditions used for the simulation are as follows:

$$[q_1 \quad q_2 \quad q_3 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3] = [-0.269\text{rad} \quad 0.468\text{rad} \quad 0.468\text{rad} \quad 1.568\text{rad/s} \quad 0.252\text{rad/s} \quad 0.252\text{rad/s}] \tag{6.2}$$

Figure 6-16 shows the 8 consecutive step trajectories of simulation.

6.4.3 Theoretical and Experimental Walking Comparisons

For the comparison of theoretical to experimental walking trends, the walking strides from both sets of results were examined. The result in figure 6-16 has been reduced to plots of two consecutive steps as shown in figure 6-17. The numberings in figure 6-17 indicates the occurrence of events as follows:

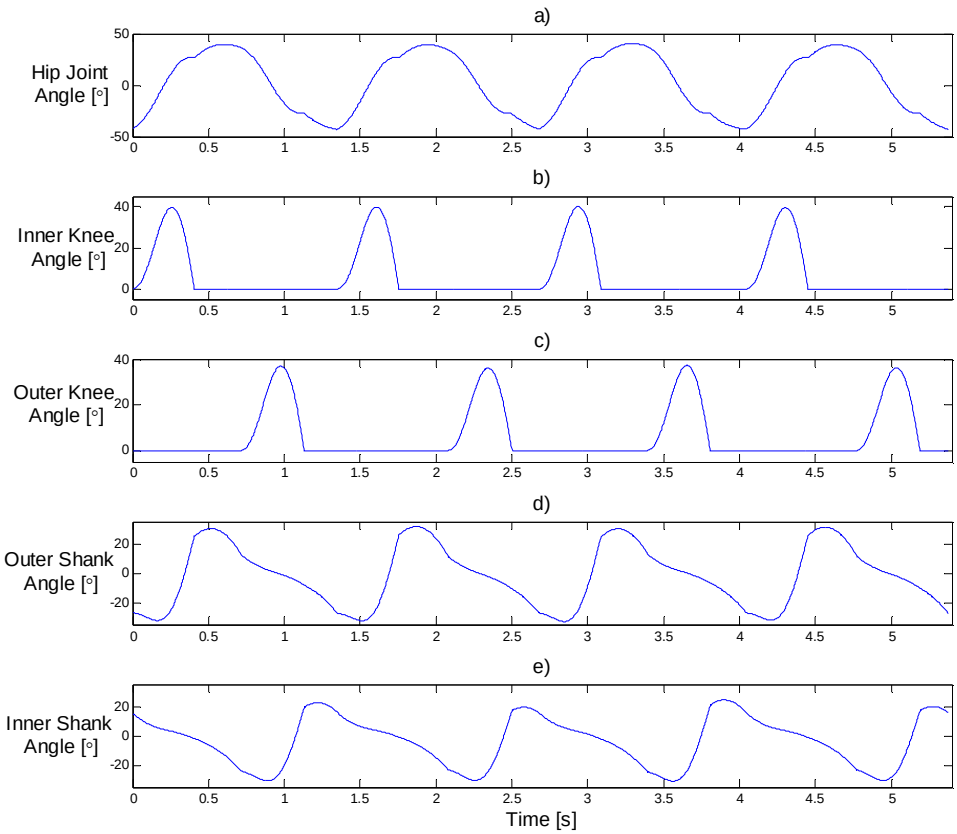

Figure 6-16: Simulated walking data as a function of time – a) Hip joint angle, b) Inner knee angle, c) Outer knee angle, d) Inner shank angle, e) Outer shank angle

Event 1: Starting condition with the inner legs as swing leg

Event 2: Inner knee strikes

Event 3: Inner foot strike

Event 4: Outer knee strike

Event 5: Outer foot strike

The hip joint angle trend in figure 6-17 (a) matches the trend in figure 6-15 (b). Both plot curves shows distinctive kinks in the plot around the region of events 2 and 4. The distinctive kinks are indications of knee strikes. Heel strike occurrences cannot be easily identified as there is no sudden change in the measured angles. The inner and outer knee bends between their respective heel strike and knee lock time intervals. Hence, the start and end times of the knee bends indicate heel and knee strikes respectively. It can be observed from the knee angle plots in figure 6-15 and figure 6-17 that the start and end times of the knee bends coincide with the leg strike events of the hip joint angle plot. No comparisons could be made for the shank angles due to incomprehensible experimental results. Nevertheless, the plotted hip and knee joint figures of both theoretical and experimental results coincide with regard to the trend patterns occurring at the starting or ending of events.

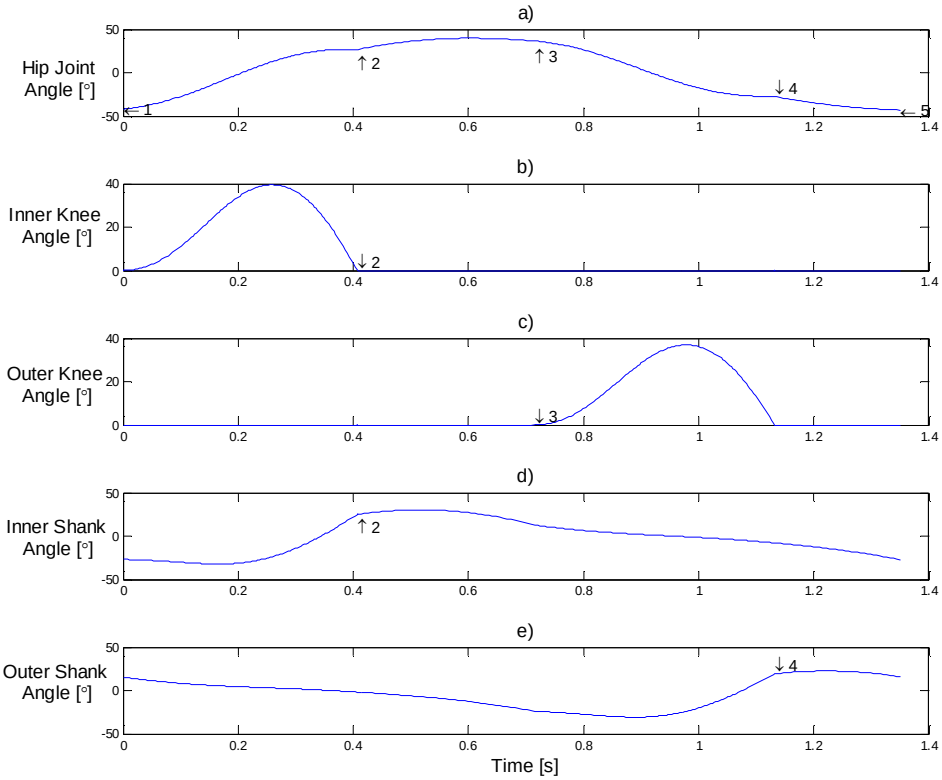

Figure 6-17: Simulated stride data as a function of time – a) Hip joint angle, b) Inner knee angle, c) Outer knee angle, d) Inner shank angle, e) Outer shank angle

6.5 Chapter Summary

Results from testing all hardware except encoders and accelerometers, showed satisfactory performances. By combining torque transducer readings and motor actuation control, the high ratio geared motor was made easily back-drivable. Actuation had to be applied in order to counteract the retarding forces of the

gearbox. The knee locking mechanisms were also effective since they lock and unlock as required. Tests showed that the optical encoders produce accurate results if reading errors do not accumulate. The accelerometers which were responsible for measuring the most significant independent generalised coordinate produced incomprehensible results. This was due to the over-sensitivity of the accelerometer to unavoidable shock vibrations. The magnitudes of the theroretical and experimental results could not be compared or verified due to the accelerometer readings.

Software evaluations show that the designed controller of Arduino boards is capable of executing the programmed commands. No problems were encountered with regard to routines not executed or programmed tasks not carried out. The data transfer interface between Matlab and the master Arduino board was robust. The average sampling rate of any sensor measured data was 0.055 s.

The biped demonstrated that it is capable of walking down a ramp with actuation. Actuation of the biped was used in order to increase the walking ground clearance. The repeatability of stable walks is largely dependent on the start-up process which establishes the initial condition of walking. If the biped is given the correct push, it has a high chance of consecutive locomotion.

Comparisons between experimental and theoretical results have been conducted. It has been verified that the constructed biped exhibits the same walking trend as the theoretical model. The above verification is crucial as it confirms the derived mathematical model to a certain extent. The magnitude of theoretical and experimental result trends could not be compared due to the errors in sensor readings discussed in this chapter.

CHAPTER 7 CONCLUSION AND FUTURE RESEARCH

7.0 Introduction

This chapter outlines the important contributions of this research. Concluding remarks and investigation findings of the dissertation chapters are summarised. Recommendations for improvements to the current designed biped based on simulation and testing results are listed. Possible future research expansion regarding the topic of chaos control in biped walking is discussed.

7.1 Research Contributions and Concluding Remarks

A lower body kneed biped was researched, designed, and fabricated to conduct experimental walking tests. It is an underactuated robot whose dynamics are controlled through hip actuations. The concepts of efficient dynamics of passive walking and flexible functionality through chaos control are the central principles of this research.

Dynamic modelling of the sagittal plane biped was performed based on equations derived using the Lagrange formalism and simulated walking motions were studied using the developed model. The two stability quantifiers namely, ground clearance and basin of attraction, were used to determine the relationship between model parameters and stability robustness. An optimised parameter range, shown in table 3-3, was obtained based on the simulated relationship plots.

Chapter 4 demonstrates the simulation control of chaos of the biped model. To the author's knowledge, no cases of chaos control have been applied to kneed walking models. The walking model of this research has more degrees of freedom than the simplest walking model, which is the most commonly used biped model for chaotic locomotion analysis. Different to most chaos control cases studied, the researched dynamic system is of the non-smooth continuous type.

Successful control of chaotic orbits onto period 1 UPOs of different slope angles has been achieved in simulations. The theory of using small perturbation forces for control of the OGY method has been verified. It was determined that for the simulated biped model, there exist only one period 1 UPO per slope angle. There are generally more than one UPO, of the same period, for any given condition [9]. The simulated biped model is shown to be one of the exceptions to the above statement. The initiative of altering system performance through the selection of period 1 UPOs was proven not possible due to the above reason. Only the performance of period 1 UPOs are desired as UPOs of higher dimensions corresponds to performance inconsistencies.

The biped was designed in accordance with specifications and constraints formulated from the results and findings in Chapter 2 to Chapter 4. All the design parameters of the biped are within the optimised range determined. A data acquisition system capable of real time data capture was developed and used during experimentations. The biped design chapter presents a novel design with regard to actuated knee locking systems. The designed system requires less actuation power as it only needs to overcome friction unlike other research designs which need to take knee flex forces into account.

Experimental testing showed that all the hardware with the exception of the encoders and accelerometers functioned effectively. Through control, and feedback signals from the torque transducer, the high reduction geared motor was made back-drivable. The designed controller executed all its programmed routines without experiencing technical failures or glitches and has demonstrated robustness. The biped successfully walked down a ramp with timed hip actuations. The comparison between the results of the experimental walks and simulation predictions indicated close similarities in trend between the two. This verifies and confirms the derived theoretical biped model to a certain extent.

Chaos control could not be implemented experimentally as the design of an effective control system for a multi-body model is complex. Even though control of the biped could not be conducted, the first step of the chaos control process which is to have the capability to measure and record the generalised coordinate readings at knee and heel strike occurrences, was taken.

7.2 Recommendation and Future Research

The captured data requires a certain level of accuracy in order for the calculation of linearised Poincaré Map equation to be effective. Since the biped generally needs to walk many steps before chaos control conditions can be met, all readings errors that are accumulative will add up over time. When the actual generalised coordinates deviates too far from the measured, the control forces calculated and applied may result in opposite effects. Due to the above causes, it is recommended that the selected sensor must have the required accuracy and must not accumulate error over time, or must self-calibrate during operation. Sensors that are sensitive in terms of acceleration such as accelerometers and inertial measurement units, are prone to shock vibrations. Use of such sensors will result in measurement spikes throughout recordings as walking robots constantly experience shocks from impact.

A control system for a multi-body model is required for the chaos control of the biped of this research. The control system must have response properties of fast rise time, short settling time, and minimal overshoot. The data transfer rate between the controller and Matlab also needs improvement as the current sampling time is considerably slow.

An extension to the OGY method of chaos control is chaos targeting. Chaos targeting shortens the transport time for the system flow to get close to the target point. Targeting is beneficial for walking experimentations as the travelled distance taken prior to reaching the control conditions are shortened. Shorter time is then required to direct the system onto a certain performance.

7.3 Research Project Summary

To answer the research question:

- It has been confirmed from simulations that a biped model based on the principles of passive dynamics, is efficient. The model only requires potential energy gained from walking down a gentle slope (minimum slope angle less than 2°) in order to sustain continuous walking.
- Using nonlinear dynamic analytical tools, the researched biped system was found to have multiple UPOs which verify the diverse functionality claim of chaotic systems. Unfortunately it was

determined that the simulated biped model, only has one period 1 UPO at every slope angle. The biped is capable of flexible performance due to the presence of multiple UPOs in its chaotic system, but only one of its orbits is a limit cycle per slope angle.

In conclusion, a biped can be both efficient and flexible in functionality but this is limited to down-slope walking, and a variety of undesired functionalities may occur.

References

1. McGeer, T. “Passive dynamic walking”, International Journal of Robotics Research, Vol. 9, No. 2, pp. 62–82, April 1990.

2. Garcia, M., Chaterjee, A., Ruina, A. and Coleman, M., “The simplest walking model: Stability, Complexity, and Scaling”, ASME Journal of Biomechanical Engineering, Vol. 120, Issue 2, pp. 281-288, 1998.

3. Kurz, M. J., Judkins, T. N., Arellano, C. and Scott-Pandorf, M., “A passive dynamic walking robot that has a deterministic nonlinear gait”, Journal of Biomechanics, Vol. 41, No. 6, pp. 1310-1316, 2008.

4. Ott, E., Grebogi, C. and Yorke, J. A., “Controlling chaos”, Physical Review Letters 64, pp. 1196-1199, 1990.

5. Shinbrot, T., Grebogi, C., Ott, E. And Yorke, J. A., “Using small perturbations to control chaos”, Nature, Vol. 363, pp. 411-417, 1993.

6. Dingwell, J. B., Cusumano, J. P., Sternad, D. and Cavanagh, P. R., “Slower speeds in patients with diabetic neuropathy lead to improved local dynamic stability of continuous overground walking’’, Journal of Biomechanics, Vol. 33, Issue 10, pp. 1269–1277, 2000.

7. Sprott, J. C., “Chaos and Time-Series Analysis”, Oxford University Press Inc., 2003.

8. Chua, L. O., Kocarev, L. J., Eckert, K. and Itoh, M., “Experimental chaos synchronization in Chua’s circuit”, International Journal of Bifurcation and Chaos, Vol. 2, No. 3, pp. 705-708, 1992.

9. Hilborn, R. C., “Chaos and Nonlinear Dynamics”, Oxford University Press Inc., 1994.

10. Ditto, W. L., Rauseo, S. N. and Spano, M. L., “Experimental control of chaos”, Physical Review Letters, Vol. 65, pp. 3211-3214, 1990.

11. Farquhar, R., Muhonen, D. and Church, L. C., “Trajectories and orbital manoeuvres for the ISEE-3/ICE comet mission”, Journal of the Astronautical Sciences, Vol. 33, No. 3, pp. 235-254, 1985.

12. Macau, E. E. N. and Grebogi, C., “Control of chaos and its relevancy to spacecraft steering”, Philosophical Transactions of the Royal Society A, Vol. 364, pp. 2463–2481, 2006.

13. McGeer T., “Passive walking with knees”, Proceedings of IEEE International Conference on Robotics and Automation, Los Alamitos, CA, pp. 1640–1645, 1990.

14. Ikemata, Y., Sano, A. and Fujimoto, H., “A physical principle of gait generation and its stabilization derived from mechanism of fixed point”, Proceedings of IEEE International Conference on Robotics and Automation, pp. 836-841, 2006.

15. Collins, S. H., Wisse, M. and Ruina, A., “A three-dimensional passive-dynamic walking robot with two legs and knees”, International Journal of Robotics Research, Vol. 20, No. 7, pp. 607–615, 2001.

16. Tedrake, R., Zhang, T.W., Fong, M.-F. and Seung H. S., “Actuating a simple 3D passive dynamic walker”, Proceedings of IEEE International Conference on Robotics and Automation, Vol.5, pp. 4656- 4661, 2004,

17. Collins, S.H., 2001. *Passive Robot* [image online] Available at: <http://www.andrew.cmu.edu/user/shc17/Passive_Robot/passive_robot_pict_thumb.jpg> [Accessed 15 February 2012].

18. Wisse, M. and van Frankenhuyzen, J., “Design and construction of mike; a 2d autonomous biped based on passive dynamic walking”, Proceeding of Conference on Adaptive Motion of Animals and Machines, Paper number WeP-I-1, Kyoto, Japan, 2003.

19. Wisse, M., Feliksdal, G., Van Frankkenhuyzen, J. and Moyer, B., “Passive-Based Walking Robot”, IEEE Robotics and Automation Magazine, Vol. 14, No. 2, pp. 52-62, 2007.

20. Schwab, A. L. and Wisse, M., “Basin of attraction of the simplest walking model”, Proceedings of ASME Design Engineering Technical Conference, DETC2001/VIB-21363, Pittsburgh, PA, 2001.

21. Goswami, A., Thuilot, B. and Espiau, B., “A study of the passive gait of a compass-like biped robot: Symmetry and chaos”, International Journal of Robotics Research, Vol. 17, No. 12, pp. 1282–301, 1998.

22. Wisse, M., Schwab, A. L. and van der Helm, F. C. T., “Passive dynamic walking model with upper body”, Robotica, Vol. 22, pp. 681–688, 2004.

23. Kurz, M. J. and Stergiou, N., “Hip actuations can be used to control bifurcations and chaos in a passive dynamic walking model”, ASME Journal of Biomechanical Engineering, Vol. 129, No. 2, pp. 216-222, 2007.

24. Garcia, M., Chatterjee, A. and Ruina, A., “Speed, efficiency, and stability of small-slope 2d passive-dynamic bipedal walking”, Proceedings of IEEE International Conference on Robotics and Automation, Piscataway, NJ, pp. 2351–2356, 1998.

25. Wisse, M., Hobbelen, D. G. E. and Schwab, A. L., “Adding the upper body to passive dynamic walking robots by means of a bisecting hip mechanism”, IEEE Transactions on Robotics, Vol. 23, No. 1, pp.112–123, 2007.

26. Hobbelen, D., Wisse, M., “Limit Cycle Walking”, Humanoid Robots: Human-like Machines, I-Tech Education and Publishing, Vienna, Chapter 14, p. 642, 2007.

27. Vukobratovic, M. and Borovac, B., “Zero moment point - thirty-five years of its life”, International Journal of Humanoid Robotics, Vol. 1, No. 1, pp. 157-173, 2004.

28. Sardain P., Rostami, M., Thomas, E. and Bessonnet, G., “Biped Robots: Correlations between technological design and dynamic behaviour”, 5th IFAC Symposium on Robot Control, pp. 417-424, Nantes, France, 1997.

29. Drillis, R. and Contini, R., “Body Segment Parameters”, New York: Office of Vocational Rehabilitation, Report No. 1166-03, New York, 1966.

30. Winter, D. A., “Biomechanics and Motor Control of Human Movement”, 2nd Edition, Wiley, New York, 1990.

31. Hansen, A. H., Childress, D. S. and Knox, E. H., “Prosthetic foot roll-over shapes with implications for alignment of trans-tibial prostheses”, Prosthetics and Orthotics International, Vol. 24, pp.205–215, 2000.

32. Sardain, P., Rostami, M. and Bessonnet, G., “An anthropomorphic biped robot: Dynamic concepts and technological design”, IEEE Transactions on Systems, Man and Cybernetics Part A, Vol. 28, No. 6, pp. 823–838, November, 1998.

33. Hansen, A., Childress, D., and Miff, S., “Roll-over characteristics of human walking on inclined surfaces”, Human Movement Science, Vol. 23, pp. 807–82, 2004.

34. Reinink, R., “Knee Locking System for a walking robot”, unpublished MSc Thesis, University of Twente, 2004.

35. Wisse, M., Schwab, A. L., van der Linde, R. Q. and van der Helm, F. C. T., “How to keep from falling forward; elementary swing leg action for passive dynamic walkers”. IEEE Transactions on Robotics, Vol. 21, No. 3, pp.393–401, 2005.

36. Hsu Chen, V. F., “Passive Dynamic Walking with Knees: A Point Foot Model”, unpublished thesis, M.A., Massachusetts Institute of Technology, 2007.

37. Wisse, M., Atkeson, C. G. and Kloimwieder, D.K., "Swing leg retraction helps biped walking stability", 5th IEEE-RAS International Conference on Humanoid Robots, pp.295-300, 2005.

38. Hsu, C. S., “Cell-to-cell mapping; a method of global analysis for nonlinear systems”, Applied Mathematical Sciences, Vol. 64, Springer-Verlag, New York, 1987.

39. Franken, M., “Ankle Actuation for Planar Bipedal Robots”, unpublished MSc Report, University of Twente, 2007.

40. Halls, S. B., 2008. *Formula for Body Mass Index*. [online] Available at: < http://www.halls.md/body-mass-index/bmirefs.htm> [Accessed 16 September 2010].

41. Hansen, A. H. and Childress, D. S., "Effects of adding weight to the torso on roll-over characteristics of walking", Journal of Rehabilitation Research and Development, Vol. 42, Issue 3, p. 381, 2005.

42. Shinbrot, T., Ditto, W., Grebogi, C., Ott, E., Spano, M. and Yorke, J. A., “Using the sensitive dependence of chaos (the "butterfly effect') to direct trajectories in an experimental chaotic system”, Physical Review Letter, Vol. 68, No. 19 , pp. 2863-2866, 1992.

43. Ott, E., “Chaos in Dynamical Systems”, Cambridge University Press, 2002.

44. Eckmann, J.-P. and Ruelle, D., “Ergodic theory of chaos and strange attractors ”, Reviews of Modern Physics, Vol. 57, No. 3, 617-656, 1985.

45. Chen, G. R., Moiola, J. L. and Wang, H. O., “Controlling Bifurcations”, Proceedings of the Indian National Science Academy, Vol. 65, Part A, No. 5, pp. 585-612, 1999.

46. Wikipedia the Free Encyclopaedia, 2005. *Logistic Map Bifurcation Diagram.* [image online] Available at: <http://en.wikipedia.org/wiki/File:LogisticMap_BifurcationDiagram.png> [Accessed 26 February 2012].

47. May, R., “Simple Mathematical Models with very Complicated Dynamics”, Nature, Vol. 261, pp. 459-67, 1976.

48. González-Miranda, J. M., “Synchronization and Control of Chaos”, Imperial College Press, 2004.

49. Grebogi, C. and Lai, Y.-C., “Controlling chaos in high dimensions,” IEEE Transaction on Circuits Systems I, Vol. 44, No. 10, pp. 971–975, 1997.

50. Vincent, T. L., “Control using chaos”, IEEE Control Systems, Vol. 17, pp. 65-76, 1997.

51. Grebogi, C. and Lai Y.-C., “Controlling chaotic dynamical systems”, Systems and Control Letters, Vol. 31, Issue 5, pp. 307-312, 1997.

52. Scholl, E. and Schuster, H. G., “Handbook of Chaos Control”, 2nd Edition, Wiley-VCH Verlag CmbH and Co. KGaA, Weinheim, 2008.

53. Ditto, W. L., Spano, M. L. and Lindner, J. F., “Techniques for the control of chaos”, Physica D: Nonlinear Phenomena, Vol. 86, Issues 1–2, pp. 198-211, 1995.

54. Ding, M. Z., Yang, W. M., In V., Ditto, W. L., Spano, M. L. and Gluckman, B., “Controlling chaos in high dimensions: theory and experiment,” Physcal Review E, Vol. 53, No. 5, pp. 4334–4344, 1996.

55. Suzuki, S., Furuta, K. and Hatakeyama, S., “Passive walking toward running,” Mathematical and Computer Modelling of Dynamical Systems, Vol. 11, No. 4, pp. 371-395, 2005.

56. Bu, S., Wang, B. and Jiang, P., “Detecting unstable periodic orbits in chaotic systems by using an efficient algorithm”, Chaos, Solitons and Fractals, Vol. 22, Issue 1, pp. 237-241, 2004.

57. Bollt, E. M., “Controlling Chaos, Targeting and Transport”, unpublished PhD thesis, University of Colorado, 1995.

58. Shinbrot, T., Ott, E., Grebogi, C. and Yorke, J. A., “Using chaos to direct orbits to targets,” Physcal Review Letters, Vol. 65, pp. 3215–3218, 1990.

59. Macau, E. E. N. and Grebogi, C., “Driving trajectories in chaotic systems”, International Journal of Bifurcation Chaos, Vol. 11, No 5, pp. 1423–1442, 2001.

60. Beekman, N., “Analysis and Development of a 2D Walking Machine”, unpublished MSc Thesis, University of Twente, 2004.

61. Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H. and Maza, D., “The control of chaos: theory and applications”, Physics Reports, Vol. 329, Issue 3, pp. 103-197, 2000.

62. Maxon Motor, n.d. *Program 2011/12 High Precision Drives and Systems*. [datasheet online] Available at: <http://pdf.directindustry.com/pdf/maxon-motor/program-2011-2012/7173-9987-_51.html> [Accessed 22 April 2012].

63. Servo City, n.d.*HS-322HD Standard Deluxe*. [datasheet online] Available at: <http://servocity.com/html/hs-322hd_standard_deluxe.html> [Accessed 12 June 2011].

64. Agilent Technologies Inc., 2002. *Quick Assembly Two and Three Channel Optical Encoders*. [datasheet online] Available at: <http://pdf1.alldatasheet.com/datasheet-pdf/view/88955/HP/HEDS-5540-A11.html> [Accessed 5April 2012].
65. Freescale Semiconductor, n.d. *±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer* MMA7361L [datasheet online] Available at: <http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf> [Accessed 13 April 2012].

66. Fisher, C.J., 2010. *Using an Accelerometer for Inclination Sensing*. [datasheet online] Available at: <http://www.analog.com/static/imported-files/application_notes/AN-1057.pdf> [Accessed 25 August 2011].

67. Lorenz Messtechnik GmbH, n.d.*Torque Sensor Static DK-15*. [datasheet online] Available at: <http://www.lorenz-messtechnik.de/english/products/static/dk-15.php> [Accessed 15 April 2012].

68. Arduino, n.d.. *Arduino Uno Board Front* [image online] Available at: <http://arduino.cc/en/uploads/Main/ArduinoUnoFront.jpg> [Accessed 14 April 2012].

69. Pololu Robotics and Electronics, n.d. *Dual VNH3SP30 Motor Driver Carrier MD03A*. [datasheet online] Available at: <http://www.pololu.com/catalog/product/707> [Accessed 9 March 2012].

Appendix A: Equations and Matrices of the Sagittal Plane Biped Model

A-1 Unlocked Knee Dynamics

The matrices in Equation 3.17 for unlocked knee dynamics are as follows:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{12} & H_{22} & H_{23} \\ H_{13} & H_{23} & H_{33} \end{bmatrix} \quad \text{A.1}$$

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & 0 & B_{23} \\ B_{31} & B_{32} & 0 \end{bmatrix} \quad \text{A.2}$$

$$G = \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix} \quad \text{A.3}$$

Where

$H_{11}=$ $I_s + I_t + I_h S_1 + l_r^2 m_h + l_{ms}^2 m_s + l_{mt}^2 m_t + 2 l_r^2 m_s + 2 l_r^2 m_t + l_t^2 m_s + m_h r_s^2 + 2 m_s r_s^2 + 2 m_t r_s^2$
$- 2 l_r l_t m_s \cos(\varepsilon_t) + 2 l_{ms} l_t m_s \cos(\theta_{ms}) - 2 l_t m_s r_s \cos(q_1) - 2 l_{ms} l_r m_s \cos(\varepsilon_t - \theta_{ms}) - 2 l_{mt} l_r m_t \cos(\varepsilon_t - \theta_{mt}) + 2 l_r m_h r_s \cos(\varepsilon_t - q_1) + 4 l_r m_s r_s \cos(\varepsilon_t - q_1) + 4 l_r m_t r_s \cos(\varepsilon_t - q_1) - 2 l_{ms} m_s r_s \cos(q_1 - \theta_{ms})$
$- 2 l_{mt} m_t r_s \cos(q_1 - \theta_{mt})$

$H_{12}=$ $- l_r l_t m_s \cos(\varepsilon_t - q_1 + q_2) - l_t m_s r_s \cos(q_2) - l_{mt} m_t r_s \cos(q_2 - \theta_{mt}) - l_{mt} l_r m_t \cos(\varepsilon_t - q_1 + q_2 - \theta_{mt})$

$H_{13}=$ $- l_{ms} m_s r_s \cos(q_3 - \theta_{ms}) - l_{ms} l_r m_s \cos(\varepsilon_t - q_1 + q_3 - \theta_{ms})$

$H_{22}=$ $m_t l_{mt}^2 + m_s l_t^2 + I_t + I_h S_2$

$H_{23}=$ $l_{ms} l_t m_s \cos(q_2 - q_3 + \theta_{ms})$

$H_{33}=$ $m_s l_{ms}^2 + I_s$

$B_{11}=$ $\dot{q}_1 \ (l_t m_s r_s \sin(q_1) + l_r m_h r_s \sin(\varepsilon_t - q_1) + 2 l_r m_s r_s \sin(\varepsilon_t - q_1) + 2 l_r m_t r_s \sin(\varepsilon_t - q_1)$
$+ l_{ms} m_s r_s \sin(q_1 - \theta_{ms}) + l_{mt} m_t r_s \sin(q_1 - \theta_{mt}))$

$B_{12}=$ $\dot{q}_2 \ (l_r l_t m_s \sin(\varepsilon_t - q_1 + q_2) + l_t m_s r_s \sin(q_2) + l_{mt} m_t r_s \sin(q_2 - \theta_{mt})$
$+ l_{mt} l_r m_t \sin(\varepsilon_t - q_1 + q_2 - \theta_{mt}))$

$B_{13}=$ $\dot{q}_3 \ (l_{ms} m_s r_s \sin(q_3 - \theta_{ms}) + l_{ms} l_r m_s \sin(\varepsilon_t - q_1 + q_3 - \theta_{ms}))$

$B_{21}=$ $-\dot{q}_1 \ (l_r l_t m_s \sin(\varepsilon_t - q_1 + q_2) + l_{mt} l_r m_t \sin(\varepsilon_t - q_1 + q_2 - \theta_{mt}))$

$B_{23}=$ $\dot{q}_3 \ l_{ms} l_t m_s \sin(q_2 - q_3 + \theta_{ms})$

$B_{31}=$ $-\dot{q}_1 \ l_{ms} l_r m_s \sin(\varepsilon_t - q_1 + q_3 - \theta_{ms})$

$B_{32}=$ $-\dot{q}_2 \ l_{ms} l_t m_s \sin(q_2 - q_3 + \theta_{ms})$

$G_{11}=$ $g (l_r m_h \sin(\varepsilon_t - \gamma - q_1) + 2 l_r m_s \sin(\varepsilon_t - \gamma - q_1) + 2 l_r m_t \sin(\varepsilon_t - \gamma - q_1) - m_h r_s \sin(\gamma) - 2 m_s r_s \sin(\gamma)$

$- 2\, m_t\, r_s \sin(\gamma) + l_{ms}\, m_s \sin(\gamma + q_1 - \theta_{ms}) + l_{mt}\, m_t \sin(\gamma + q1 - \theta_{mt}) + l_t\, m_s \sin(\gamma + q_1))$

$G_{21} = \quad g\,(l_{mt}\, m_t \sin(\gamma + q_2 - \theta_{mt}) + l_t\, m_s \sin(\gamma + q_2))$

$G_{31} = \quad g\, l_{ms}\, m_s \sin(\gamma + q_3 - \theta_{ms})$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \qquad \text{A.4}$$

Where $\tau_1 = u_1 - u_2$, $\tau_2 = u_2 - u_3$ and $\tau_3 = u_3$.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \qquad \text{A.5}$$

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \qquad \text{A.6}$$

$$\ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ q_2 \\ \ddot{q}_3 \end{bmatrix} \qquad \text{A.7}$$

A-2 Knee Strike Transition

The distance and velocity vectors of Equation 3.20 and 3.21 are as follows:

Distance vectors

$\overrightarrow{Oh^-} = \quad [l_r \sin(-q_1+\varepsilon_t),\ lr \cos(-q_1+\varepsilon_t) + r_s,\ 0]$

$\overrightarrow{Oh^+} = \quad \overrightarrow{Oh^-}$

$\overrightarrow{Os_{st}^-} = \quad [l_r \sin(-q_1+\varepsilon_t) - l_t \sin(-q_1) - l_{ms} \sin(-q_1+\theta_{ms}),\ l_r \cos(-q_1+\varepsilon_t) + r_s - l_t \cos(-q_1)$

$\quad - l_{ms} \cos(-q_1+\theta_{ms}),\ 0]$

$\overrightarrow{Os_{st}^+} = \quad \overrightarrow{Os_{st}^-}$

$\overrightarrow{Os_{sw}^-} = \quad [l_r \sin(-q_1+\varepsilon_t) + l_t \sin(q_2) + l_{ms} \sin(q_3-\theta_{ms}),\ l_r \cos(-q_1+\varepsilon_t) + r_s - l_t \cos(q_2) - l_{ms} \cos(q_3-\theta_{ms}),\ 0]$

$\overrightarrow{Os_{sw}^+} = \quad [l_r \sin(-q_1+\varepsilon_t) + l_t \sin(q_2) + l_{ms} \sin(q_2-\theta_{ms}),\ l_r \cos(-q_1+\varepsilon_t) + r_s - l_t \cos(q_2) - l_{ms} \cos(q_2-\theta_{ms}),\ 0]$

$\overrightarrow{Ot_{st}^-} = \quad [l_r \sin(-q_1+\varepsilon_t) - l_{mt} \sin(-q_1+\theta_{mt}),\ l_r \cos(-q_1+\varepsilon_t) + r_s - l_{mt} \cos(-q_1+\theta_{mt}),\ 0]$

$\overrightarrow{Ot_{st}^+} = \quad \overrightarrow{Ot_{st}^-}$

$\overrightarrow{Ot_{sw}^{-}}$= $[l_r \sin(-q_1+\varepsilon_t) + l_{mt} \sin(q_2-\theta_{mt}),\ l_r \cos(-q_1+\varepsilon_t) + r_s - l_{mt} \cos(q_2-\theta_{mt}),\ 0]$

$\overrightarrow{Ot_{sw}^{+}}$= $\overrightarrow{Ot_{sw}^{-}}$

$\overrightarrow{Hs_{sw}^{-}}$= $[l_t \sin(q_2) + l_{ms} \sin(q_3-\theta_{ms}),\ -l_t \cos(q_2) - l_{ms} \cos(q_3-\theta_{ms}),\ 0]$

$\overrightarrow{Ht_{sw}^{-}}$= $[l_{mt} \sin(q_2-\theta_{mt}),\ -l_{mt} \cos(q_2-\theta_{mt}),\ 0]$

$\overrightarrow{Hs_{sw}^{+}}$= $[l_t \sin(q_2) + l_{ms} \sin(q_2-\theta_{ms}),\ -l_t \cos(q_2) - l_{ms} \cos(q_2-\theta_{ms}),\ 0]$

$\overrightarrow{Ht_{sw}^{+}}$= $[l_{mt} \sin(q_2-\theta_{mt}),\ -l_{mt} \cos(q_2-\theta_{mt}),\ 0]$

Velocity vectors

$\overrightarrow{V_h^{-}}$= $[(-r_s-l_r \cos(q_1-\varepsilon t))\ \dot{q_1},\ -l_r \sin(q_1-\varepsilon_t)\ \dot{q_1},\ 0]$

$\overrightarrow{Vh^{+}}$= $\overrightarrow{V_h^{-}}$

$\overrightarrow{Vs_{st}^{-}}$= $[(-r_s-l_r \cos(q_1-\varepsilon_t)+l_t \cos(q_1)+l_{ms} \cos(q_1-\theta_{ms}))\ \dot{q_1},$

$(-l_r \sin(q_1-\varepsilon_t) + l_t \sin(q_1) + l_{ms} \sin(q1-\theta_{ms}))\ \dot{q_1},\ 0]$

$\overrightarrow{Vs_{st}^{+}}$= $\overrightarrow{Vs_{st}^{-}}$

$\overrightarrow{Vs_{sw}^{-}}$= $[(-r_s-l_r \cos(q_1-\varepsilon_t))\ \dot{q_1} + l_t \cos(q_2)\ \dot{q_2} + l_{ms} \cos(-q_3+\theta_{ms})\ \dot{q_3},$

$-l_r \sin(q_1-\varepsilon_t)\ \dot{q_1} + l_t \sin(q_2)\ \dot{q_2} - l_{ms} \sin(-q_3+\theta_{ms})\ \dot{q_3},\ 0]$

$\overrightarrow{Vs_{sw}^{+}}$= $[(-r_s-l_r \cos(q_1-\varepsilon_t))\ \dot{q_1} + (l_t \cos(q_2) + l_{ms} \cos(-q_2+\theta_{ms}))\ \dot{q_2},$

$-l_r \sin(q_1-\varepsilon_t)\ \dot{q_1} + (l_t \sin(q_2)\ \dot{q_2} - l_{ms} \sin(-q_2+\theta_{ms}))\ \dot{q_2},\ 0]$

$\overrightarrow{Vt_{st}^{-}}$= $[(-r_s-l_r \cos(q_1-\varepsilon_t)+l_{mt} \cos(q_1-\theta_{mt}))\ \dot{q_1},\ (-l_r \sin(q_1-\varepsilon_t) + l_{mt} \sin(q_1-\theta_{mt}))\ \dot{q_1},\ 0]$

$\overrightarrow{Vt_{st}^{+}}$= $\overrightarrow{Vt_{st}^{-}}$

$\overrightarrow{Vt_{sw}^{-}}$= $[(-r_s-l_r \cos(q_1-\varepsilon_t))\ \dot{q_1}+l_{mt} \cos(-q_2+\theta_{mt})\ \dot{q_2},\ -l_r \sin(q_1-\theta_t)\ \dot{q_1} - l_{mt} \sin(-q_2+\theta_{mt})\ \dot{q_2},\ 0]$

$\overrightarrow{Vt_{sw}^{+}}$= $\overrightarrow{Vt_{sw}^{-}}$

The transition matrices Q^+ and Q^- are 2x2 and 2x3 matrices respectively. It is expressed as follows:

$$Q^{-} = \begin{bmatrix} Q_{11}^{-} & Q_{12}^{-} & Q_{13}^{-} \\ Q_{21}^{-} & Q_{22}^{-} & Q_{23}^{-} \end{bmatrix} \quad \text{A.8}$$

$$Q^{+} = \begin{bmatrix} Q_{11}^{+} & Q_{12}^{+} \\ Q_{21}^{+} & Q_{22}^{+} \end{bmatrix} \quad \text{A.9}$$

Where

$Q_{11}^{-} =$ $- m_s\, l_t \cos(q_2)\, r_s - 2\, m_s\, l_t \cos(q_1)\, r_s - m_s\, l_{ms}\, l_r \cos(-q_3 + \theta_{ms} + q_1 - \varepsilon_t) + 2\, m_h\, l_r\, r_s \cos(q_1 - \varepsilon_t) + m_s\, l_{ms}^2$
$- 2\, m_t\, l_{mt}\, l_r \cos(\varepsilon_t - \theta_{mt}) + m_t\, l_{mt}^2 - m_s\, l_{ms}\, r_s \cos(-q_3 + \theta_{ms}) + m_s\, l_t^2 + 4\, m_s\, l_r\, r_s \cos(q_1 - \varepsilon_t)$
$- m_t\, l_{mt}\, r_s \cos(-q_2 + \theta_{mt}) - m_s\, l_t\, l_r \cos(-q_2 + q_1 - \varepsilon_t) + 4\, m_t\, l_r\, r_s \cos(q_1 - \varepsilon_t) - 2\, m_s\, l_{ms}\, l_r \cos(\varepsilon_t - \theta_{ms}) + 2$
$m_s\, l_r^2 + 2\, m_t\, l_r^2 + m_h\, l_r^2 - 2\, m_s\, l_{ms}\, r_s \cos(q_1 - \theta_{ms}) + 2\, m_t\, r_s^2 + m_h\, r_s^2 + 2\, m_s\, r_s^2 - 2\, m_s\, l_r \cos(\varepsilon_t)\, l_t$
$+ 2\, m_s\, l_t\, l_{ms} \cos(\theta_{ms}) - m_t\, l_{mt}\, l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) - 2\, m_t\, l_{mt}\, r_s \cos(q_1 - \theta_{mt})$

$Q_{12}^{-} =$ $- m_s\, l_t \cos(q_2)\, r_s + m_t\, l_{mt}^2 - m_s\, l_t\, l_r \cos(-q_2 + q_1 - \varepsilon_t) - m_t\, l_{mt}\, r_s \cos(-q_2 + \theta_{mt})$
$+ m_s\, l_{ms}\, l_t \cos(-q_3 + \theta_{ms} + q_2) - m_t\, l_{mt}\, l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) + m_s\, l_t^2$

$Q_{13}^{-} =$ $- m_s\, l_r\, l_{ms} \cos(-q_3 + \theta_{ms} + q_1 - \varepsilon_t) + m_s\, l_{ms}\, l_t \cos(-q_3 + \theta_{ms} + q_2) - m_s\, r_s\, l_{ms} \cos(-q_3 + \theta_{ms}) + m_s\, l_{ms}^2$

$Q_{21}^{-} =$ $- m_s\, l_r\, l_{ms} \cos(-q_3 + \theta_{ms} + q_1 - \varepsilon_t) - m_t\, l_{mt}\, r_s \cos(-q_2 + \theta_{mt}) - m_s\, l_t \cos(q_2)\, r_s - m_s\, r_s\, l_{ms} \cos(-q_3 + \theta_{ms})$
$- m_t\, l_{mt}\, l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) - m_s\, l_t\, l_r \cos(-q_2 + q_1 - \varepsilon_t)$

$Q_{22}^{-} =$ $m_s\, l_{ms}\, l_t \cos(-q_3 + \theta_{ms} + q_2) + m_t\, l_{mt}^2 + m_s\, l_t^2$

$Q_{23}^{-} =$ $m_s\, l_{ms}\, l_t \cos(-q_3 + \theta_{ms} + q_2) + m_s\, l_{ms}^2$

$Q_{11}^{+} =$ $4\, m_s\, l_r\, r_s \cos(q_1 - \varepsilon_t) + 2\, m_t\, r_s^2 + 2\, m_s\, r_s^2 - 2\, m_s\, l_t \cos(q_1)\, r_s - m_s\, l_r\, l_t \cos(-q_2 + q_1 - \varepsilon_t) + m_s\, l_t^2 + m_h\, l_r^2$
$- m_s\, r_s\, l_{ms} \cos(-q_2 + \theta_{ms}) - m_s\, l_r\, l_{ms} \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) - m_t\, l_r\, l_{mt} \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t)$
$- m_t\, r_s\, l_{mt} \cos(-q_2 + \theta_{mt}) + 2\, m_t\, l_r^2 + 2\, m_s\, l_r^2 - 2\, m_s\, l_r \cos(\varepsilon_t)\, l_t + m_h\, r_s^2 - 2\, m_s\, l_{ms}\, r_s \cos(q_1 - \theta_{ms})$
$+ 2\, m_h\, l_r\, r_s \cos(q_1 - \varepsilon_t) + m_t\, l_{mt}^2 - 2\, m_t\, l_{mt}\, r_s \cos(q_1 - \theta_{mt}) + 4\, m_t\, l_r\, r_s \cos(q_1 - \varepsilon_t) - 2\, m_t\, l_{mt}\, l_r \cos(\varepsilon_t - \theta_{mt})$
$+ m_s\, l_{ms}^2 - 2\, m_s\, l_r\, l_{ms} \cos(\varepsilon_t - \theta_{ms}) + 2\, m_s\, l_t\, l_{ms} \cos(\theta_{ms}) - m_s\, l_t \cos(q_2)\, r_s$

$Q_{12}^{+} =$ $m_s\, l_t^2 - m_s\, l_{ms}\, l_r \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) - m_t\, l_{mt}\, r_s \cos(-q_2 + \theta_{mt}) - m_s\, l_{ms}\, r_s \cos(-q_2 + \theta_{ms}) + m_t\, l_{mt}^2$
$- m_s\, l_t\, l_r \cos(-q_2 + q_1 - \varepsilon_t) - m_t\, l_{mt}\, l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) + m_s\, l_{ms}^2 + 2\, m_s\, l_t\, l_{ms} \cos(\theta_{ms}) - m_s\, l_t \cos(q_2)\, r_s$

$Q_{21}^{+} =$ $- m_t\, l_{mt}\, r_s \cos(-q_2 + \theta_{mt}) - m_t\, l_{mt}\, l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) - m_s\, l_{ms}\, r_s \cos(-q_2 + \theta_{ms})$
$- m_s\, l_{ms}\, l_r \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) - m_s\, l_t\, l_r \cos(-q_2 + q_1 - \varepsilon_t) - m_s\, l_t \cos(q_2)\, r_s$

$Q_{22}^{+} =$ $m_s\, l_{ms}^2 + m_t\, l_{mt}^2 + 2\, m_s\, l_t\, l_{ms} \cos(\theta_{ms}) + m_s\, l_t^2$

A-3 Locked Knee Dynamics

The matrices in Equation 3.17 for locked knee dynamics are as follows:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} \qquad \text{A.10}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & 0 \end{bmatrix} \qquad \text{A.11}$$

$$G = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} \qquad \text{A.12}$$

Where

$H_{11}=$ $I_s + I_t + I_h S_1 + l_r^2 m_h + l_{ms}^2 m_s + l_{mt}^2 m_t + 2 l_r^2 m_s + 2 l_r^2 m_t + l_t^2 m_s + m_h r_s^2 + 2 m_s r_s^2$
$+ 2 m_t r_s^2 - 2 l_r l_t m_s cos(\varepsilon_t) + 2 l_{ms} l_t m_s cos(\theta_{ms}) - 2 l_t m_s r_s cos(q_1) - 2 l_{ms} l_r m_s cos(\varepsilon_t - \theta_{ms})$
$- 2 l_{mt} l_r m_t cos(\varepsilon_t - \theta_{mt}) + 2 l_r m_h r_s cos(\varepsilon_t - q_1) + 4 l_r m_s r_s cos(\varepsilon_t - q_1) + 4 l_r m_t r_s cos(\varepsilon_t - q_1)$
$- 2 l_{ms} m_s r_s cos(q_1 - \theta_{ms}) - 2 l_{mt} m_t r_s cos(q_1 - \theta_{mt})$

$H_{12}=$ $- l_r l_t m_s cos(\varepsilon_t - q_1 + q_2) - l_t m_s r_s cos(q_2) - l_{ms} m_s r_s cos(q_2 - \theta_{ms}) - l_{mt} m_t r_s cos(q_2 - \theta_{mt})$
$- l_{ms} l_r m_s cos(\varepsilon_t - q_1 + q_2 - \theta_{ms}) - l_{mt} l_r m_t cos(\varepsilon_t - q_1 + q_2 - \theta_{mt})$

$H_{22}=$ $m_s l_{ms}^2 + 2 m_s cos(\theta_{ms}) l_{ms} l_t + m_t l_{mt}^2 + m_s l_t^2 + I_s + I_t + I_h S_2$

$B_{11}=$ $\dot{q_1}$ $(l_t m_s r_s sin(q_1) + l_r m_h r_s sin(\varepsilon_t - q_1) + 2 l_r m_s r_s sin(\varepsilon_t - q_1) + 2 l_r m_t r_s sin(\varepsilon_t - q_1)$
$+ l_{ms} m_s r_s sin(q_1 - \theta_{ms}) + l_{mt} m_t r_s sin(q_1 - \theta_{mt}))$

$B_{12}=$ $\dot{q_2}$ $(l_r l_t m_s sin(\varepsilon_t - q_1 + q_2) + l_t m_s r_s sin(q_2) + l_{ms} m_s r_s sin(q_2 - \theta_{ms}) + l_{mt} m_t r_s sin(q_2 - \theta_{mt})$
$+ l_{ms} l_r m_s sin(\varepsilon_t - q_1 + q_2 - \theta_{ms}) + l_{mt} l_r m_t sin(\varepsilon_t - q_1 + q_2 - \theta_{mt}))$

$B_{21}=$ $-\dot{q_1}$ $(l_r l_t m_s sin(\varepsilon_t - q_1 + q_2) + l_{ms} l_r m_s sin(\varepsilon_t - q_1 + q_2 - \theta_{ms}) + l_{mt} l_r m_t sin(\varepsilon_t - q_1 + q_2 - \theta_{mt}))$

$G_{11}=$ $g (l_r m_h sin(\varepsilon_t - \gamma - q_1) + 2 l_r m_s sin(\varepsilon_t - \gamma - q_1) + 2 l_r m_t sin(\varepsilon_t - \gamma - q_1) - m_h r_s sin(\gamma) - 2 m_s r_s sin(\gamma)$
$- 2 m_t r_s sin(\gamma) + l_{ms} m_s sin(\gamma + q_1 - \theta_{ms}) + l_{mt} m_t sin(\gamma + q_1 - \theta_{mt}) + l_t m_s sin(\gamma + q_1))$

$G_{21}=$ $g (l_{ms} m_s sin(\gamma + q_2 - \theta_{ms}) + l_{mt} m_t sin(\gamma + q_2 - \theta_{mt}) + l_t m_s sin(\gamma + q_2))$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \text{A.13}$$

Where $\tau_1=u_1-u_2$, $\tau_2=u_2$.

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \text{A.14}$$

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad \text{A.15}$$

$$\ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \quad \text{A.16}$$

A-4 Heel Strike Transition

The distance and velocity vectors of Equation 3.24 and 3.24 are as follows:

Distance vectors

$\overrightarrow{Oh^-}=$ $[l_r \sin(-q_2+\varepsilon_t),$ $l_r \cos(-q_2+\varepsilon_t) + r_s, 0]$

$\overrightarrow{Oh^+}=$ $[l_r sin(-q_1+\varepsilon_t),$ $l_r cos(-q_1+\varepsilon_t) + r_s, 0]$

$\overrightarrow{Os_{sw}^-}=$ $[l_r sin(-q_2+\varepsilon_t) - l_t sin(-q_2) - l_{ms} sin(-q_2+\theta_{ms}), l_r cos(-q_2+\varepsilon_t) + r_s - l_t cos(-q_2)$

$-l_{ms} \cos(-q_2+\theta_{ms}), 0]$

$\overrightarrow{Os_{st}^{+}} = [l_r \sin(-q_1+\varepsilon_t) - l_t \sin(-q_1) - l_{ms} \sin(-q_1+\theta_{ms}), l_r \cos(-q_1+\varepsilon_t) + r_s - l_t \cos(-q_1)$

$-l_{ms} \cos(-q_1+\theta_{ms}), 0]$

$\overrightarrow{Os_{st}^{-}} = [l_r \sin(-q_2+\varepsilon_t) + l_t \sin(q_1) + l_{ms} \sin(q_1-\theta_{ms}), l_r \cos(-q_2+\varepsilon_t) + r_s - l_t \cos(q_1) - l_{ms} \cos(q1-\theta_{ms}), 0]$

$\overrightarrow{Os_{sw}^{+}} = [l_r \sin(-q_1+\varepsilon_t) + l_t \sin(q_2) + l_{ms} \sin(q_2-\theta_{ms}), l_r \cos(-q_1+\varepsilon_t) + r_s - l_t \cos(q_2) - l_{ms} \cos(q2-\theta_{ms}), 0]$

$\overrightarrow{Ot_{sw}^{-}} = [l_r \sin(-q_2+\varepsilon_t) - l_{mt} \sin(-q_2+\theta_{mt}), l_r \cos(-q_2+\varepsilon_t) + r_s - l_{mt} \cos(-q_2+\theta_{mt}), 0]$

$\overrightarrow{Ot_{st}^{+}} = [l_r \sin(-q_1+\varepsilon_t) - l_{mt} \sin(-q_1+\theta_{mt}), l_r \cos(-q_1+\varepsilon_t) + r_s - l_{mt} \cos(-q_1+\theta_{mt}), 0]$

$\overrightarrow{Ot_{st}^{-}} = [l_r \sin(-q_2+\varepsilon_t) + l_{mt} \sin(q_1-\theta_{mt}), l_r \cos(-q_2+\varepsilon_t) + r_s - l_{mt} \cos(q_1-\theta_{mt}), 0]$

$\overrightarrow{Ot_{sw}^{+}} = [l_r \sin(-q_1+\varepsilon_t) + l_{mt} \sin(q_2-\theta_{mt}), l_r \cos(-q_1+\varepsilon_t) + r_s - l_{mt} \cos(q_2-\theta_{mt}), 0]$

$\overrightarrow{Ht_{st}^{-}} = [l_{mt} \sin(q_1-\theta_{mt}), -l_{mt} \cos(q_1-\theta_{mt}), 0]$

$\overrightarrow{Hs_{st}^{-}} = [l_t \sin(q_1) + l_{ms} \sin(q_1-\theta_{ms}), -l_t \cos(q_1) - l_{ms} \cos(q_1-\theta_{ms}), 0]$

$\overrightarrow{Ht_{sw}^{-}} = [l_{mt} \sin(q_2-\theta_{mt}), -l_{mt} \cos(q_2-\theta_{mt}), 0]$

$\overrightarrow{Hs_{sw}^{+}} = [l_t \sin(q_2) + l_{ms} \sin(q_2-\theta_{ms}), -l_t \cos(q_2) - l_{ms} \cos(q_2-\theta_{ms}), 0]$

Velocity vectors

$\overrightarrow{V_h^{-}} = [(-r_s - l_r \cos(q_1-\varepsilon t))\, \dot{q_1}, -l_r \sin(q_1-\varepsilon_t)\, \dot{q_1}, 0]$

$\overrightarrow{Vh^{+}} = \overrightarrow{V_h^{-}}$

$\overrightarrow{Vs_{st}^{-}} = [(-r_s - l_r \cos(q_1-\varepsilon_t) + l_t \cos(q_1) + l_{ms} \cos(q_1-\theta_{ms}))\ \dot{q_1},$

$(-l_r \sin(q_1-\varepsilon_t) + l_t \sin(q_1) + l_{ms} \sin(q1-\theta_{ms}))\, \dot{q_1}, 0]$

$\overrightarrow{Vs_{st}^{+}} = \overrightarrow{Vs_{st}^{-}}$

$\overrightarrow{Vs_{sw}^{-}} = [(-r_s - l_r \cos(q_1-\varepsilon_t))\, \dot{q_1} + l_t \cos(q_2)\, \dot{q_2} + l_{ms} \cos(-q_2+\theta_{ms})\, \dot{q_2},$

$-l_r \sin(q_1-\varepsilon_t)\, \dot{q_1} + l_t \sin(q_2)\, \dot{q_2} - l_{ms} \sin(-q_2+\theta_{ms})\, \dot{q_2}, 0]$

$\overrightarrow{Vs_{sw}^{+}} = \overrightarrow{Vs_{sw}^{-}}$

$\overrightarrow{Vt_{st}^{-}} = [(-r_s - l_r \cos(q_1-\varepsilon_t) + l_{mt} \cos(q_1-\theta_{mt}))\, \dot{q_1}, (-l_r \sin(q_1-\varepsilon_t) + l_{mt} \sin(q_1-\theta_{mt}))\, \dot{q_1}, 0]$

$\overrightarrow{Vt_{st}^{+}} = \overrightarrow{Vt_{st}^{-}}$

$\overrightarrow{Vt_{sw}^{-}} = [(-r_s - l_r \cos(q_1 - \varepsilon_t))\ \dot{q}_1 + l_{mt} \cos(-q_2 + \theta_{mt})\ \dot{q}_2,\ -l_r \sin(q_1 - \theta_t)\ \dot{q}_1 - l_{mt} \sin(-q_2 + \theta_{mt})\ \dot{q}_2,\ 0]$

$\overrightarrow{Vt_{sw}^{+}} = \overrightarrow{Vt_{sw}^{-}}$

The transition matrices Q^+ and Q^- are both 2x2 matrices. It is expressed as follows:

$$Q^- = \begin{bmatrix} Q_{11}^- & Q_{12}^- \\ Q_{21}^- & Q_{22}^- \end{bmatrix} \quad \text{A.17}$$

$$Q^+ = \begin{bmatrix} Q_{11}^+ & Q_{12}^+ \\ Q_{21}^+ & Q_{22}^+ \end{bmatrix} \quad \text{A.18}$$

Where

$Q_{11}^- =$ $-2 m_s l_t \cos(q_1) r_s - m_s l_t \cos(q_2) r_s + 2 m_s l_r r_s \cos(-q_2 + \varepsilon_t) - 2 m_s l_{ms} r_s \cos(q_1 - \theta_{ms})$
$+ m_h l_r r_s \cos(q_1 - \varepsilon_t) + m_t l_{mt}^2 + m_s l_{ms}^2 + m_h l_r^2 \cos(-q_2 + q_1) + m_h l_r r_s \cos(-q_2 + \varepsilon_t)$
$+ m_s l_t^2 + 2 m_t l_r r_s \cos(q_1 - \varepsilon_t) - m_s l_r \cos(\varepsilon_t) l_t + 2 m_s l_t l_{ms} \cos(\theta_{ms}) + 2 m_t l_r r_s \cos(-q_2 + \varepsilon_t)$
$+ 2 m_s l_r r_s \cos(q_1 - \varepsilon_t) - m_s l_t l_r \cos(-q_2 + q_1 - \varepsilon_t) - m_s l_t l_r \cos(-q_2 + q_1 + \varepsilon_t) - 2 m_t l_{mt} r_s \cos(q_1 - \theta_{mt}) + m_h r_s^2$
$+ 2 m_s r_s^2 + 2 m_t r_s^2 - m_t r_s l_{mt} \cos(-q_2 + \theta_{mt}) - m_s r_s l_{ms} \cos(-q_2 + \theta_{ms}) - m_t l_r l_{mt} \cos(\varepsilon_t - \theta_{mt})$
$- m_s l_r l_{ms} \cos(\varepsilon_t - \theta_{ms}) - m_t l_{mt} l_r \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) - m_t l_{mt} l_r \cos(q_1 + \varepsilon_t - q_2 - \theta_{mt})$
$+ 2 m_t l_r^2 \cos(-q_2 + q_1) - m_s l_{ms} l_r \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) - m_s l_{ms} l_r \cos(q_1 + \varepsilon_t - q_2 - \theta_{ms})$
$+ 2 m_s l_r^2 \cos(-q_2 + q_1)$

$Q_{12}^- =$ $- m_s l_t \cos(q_2) r_s + m_t l_{mt}^2 + m_s l_{ms}^2 + m_s l_t^2 - m_s l_r \cos(\varepsilon_t) l_t + 2 m_s l_t l_{ms} \cos(\theta_{ms})$
$- m_t r_s l_{mt} \cos(-q_2 + \theta_{mt}) - m_s r_s l_{ms} \cos(-q_2 + \theta_{ms}) - m_t l_r l_{mt} \cos(\varepsilon_t - \theta_{mt}) - m_s l_r l_{ms} \cos(\varepsilon_t - \theta_{ms})$

$Q_{21}^- =$ $- m_s l_t \cos(q_1) r_s - m_t l_{mt} r_s \cos(q_1 - \theta_{mt}) - m_s l_{ms} r_s \cos(q_1 - \theta_{ms}) + m_t l_{mt}^2 + m_s l_{ms}^2 + m_s l_t^2$
$- m_s l_r \cos(\varepsilon_t) l_t + 2 m_s l_t l_{ms} \cos(\theta_{ms}) - m_t l_r l_{mt} \cos(\varepsilon_t - \theta_{mt}) - m_s l_r l_{ms} \cos(\varepsilon_t - \theta_{ms})$

$Q_{22}^- =$ 0

$Q_{11}^+ =$ $2 m_s l_t l_{ms} \cos(\theta_{ms}) - 2 m_s l_t l_r \cos(\varepsilon_t) + m_t l_{mt}^2 - 2 m_s l_t \cos(q1) r_s + 2 m_s l_r^2 + m_s l_{ms}^2$
$- m_t r_s l_{mt} \cos(-q_2 + \theta_{mt}) - m_s l_r l_t \cos(-q_2 + q_1 - \varepsilon_t) - m_t l_r l_{mt} \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t)$
$- m_s r_s l_{ms} \cos(-q_2 + \theta_{ms}) - m_s l_r l_{ms} \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) + 2 m_t r_s^2 + 2 m_s r_s^2 + m_h r_s^2$
$- 2 m_s l_{ms} l_r \cos(\varepsilon_t - \theta_{ms}) - m_s l_t \cos(q_2) r_s + m_s l_t^2 + 4 m_t r_s l_r \cos(q_1 - \varepsilon_t) - 2 m_s l_{ms} r_s \cos(q_1 - \theta_{ms})$
$- 2 m_t l m_t r_s \cos(q_1 - \theta_{mt}) + 2 m_h r_s l_r \cos(q_1 - \varepsilon_t) + 4 m_s r_s l_r \cos(q_1 - \varepsilon_t) - 2 m_t l_{mt} l_r \cos(\varepsilon_t - \theta_{mt})$
$+ 2 m_t l_r^2 + m_h l_r^2$

$Q_{12}^+ =$ $2 m_s l_t l_{ms} \cos(\theta_{ms}) + m_t l_{mt}^2 + m_s l_{ms}^2 - m_t r_s l_{mt} \cos(-q_2 + \theta_{mt}) - m_s l_r l_t \cos(-q_2 + q_1 - \varepsilon_t)$
$- m_t l_r l_{mt} \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t) - m_s r_s l_{ms} \cos(-q_2 + \theta_{ms}) - m_s l_r l_{ms} \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms})$
$- m_s l_t \cos(q_2) r_s + m_s l_t^2$

$Q_{21}^+ =$ $- m_t r_s l_{mt} \cos(-q_2 + \theta_{mt}) - m_s l_r l_t \cos(-q_2 + q_1 - \varepsilon_t) - m_t l_r l_{mt} \cos(-q_2 + \theta_{mt} + q_1 - \varepsilon_t)$
$- m_s r_s l_{ms} \cos(-q_2 + \theta_{ms}) - m_s l_r l_{ms} \cos(q_1 - \varepsilon_t - q_2 + \theta_{ms}) - m_s l_t \cos(q_2) r_s$

$Q_{22}^+ =$ $m_s l_{ms}^2 + m_s lt^2 + m_t lm_t^2 + 2 m_s l_t l_{ms} \cos(\theta_{ms})$

Appendix B: Matlab Biped Motion Simulation Codes

The simulation codes for the generation of biped trajectories are presented in this appendix.

```
function sagittal_biped_model_1
%% Constants
% Masses
mh=3;   % hip
mt=2.5; % thigh
ms=0.9; % shank
% Moments of Inertia
Ih=1200/1000^2;  % hip
It=20000/1000^2; % thigh
Is=15000/1000^2; % shank
% Lengths
lt=0.277;     % thigh length
ls=0.323;     % shank length
b2= 0.277/2;  % length from hip to mt (y projection)
c2= 0 ;       % length from hip to mt (x projection)
b1=  0.323/2; % length from knee to ms (y projection)
c1= 0.020;    % length from knee to ms (x projection)
cf= 0.100;    % length from ground to foot radius center (y projection)
wf= 0.050;    % length from hip to foot radius center (x projection)
lmt=sqrt((c2)^2+(b2)^2);        % length from hip to mt
lms=sqrt((c1)^2+(b1)^2);        % length from knee to ms
lr=sqrt((lt+ls-cf)^2+(wf)^2);   % length from hip to foot centre
% Angles
themt=atan(c2/b2);          % angle between length mh to mt and leg
thems=atan(c1/b1);          % angle between length knee to ms and leg
epst=atan(wf/(lt+ls-cf));   % angle between length mh to foot radius center
%   and leg
% Radius
rs=cf;             % radius of foot curvature
% Others
g=9.81;     % gravitational constant
refine = 1; % Increases the number of output points by a factor of Refine.
sign_dq1=0; % hip moment of inertia S1
sign_dq2=1; % hip moment of inertia S2
% Torque T =[tau1; tau2;tau3]
tau1=0; % u1-u2
tau2=0; % u2-u3
tau3=0; % u3


%% Initial condition at start of step
% Slope angle
gam= 0.098542714  ;
% Independent generalised coordinates
y0_q1=-0.28;
y0_dq1=1.6;
y0_dq2=-0.17;
% Dependent coordinates
y0_q2=-y0_q1+2*epst;
y0_q3=y0_q2;
y0_dq3=y0_dq2;
y0unlockedknee =[y0_q1    y0_q2    y0_q3    y0_dq1    y0_dq2    y0_dq3];
% Simulation time span [seconds]
tunlockedknee = [0 1];
% Result vectors
q11=y0unlockedknee(1);       %q1
q22=y0unlockedknee(2);       %q2
q33=y0unlockedknee(3);       %q3
qd11=y0unlockedknee(4);      %dq1
qd22=y0unlockedknee(5);      %dq2
qd33=y0unlockedknee(6);      %dq3
t=tunlockedknee(1);          %time
tlength=length(t);
n_steps=10    ;% number of steps
```

```
for i=1:n_steps     % for 1 to number of steps
    sign_dq1=~sign_dq1; % negate S1
    sign_dq2=~sign_dq2; % negate S2


    %% Unlocked knee dynamics
    % options used in solving unlocked knee ode
    options = odeset('RelTol',1e-10,'AbsTol',1e-10,...
        'Events',@eventsunlockedknee,'Refine',refine);
    [tuk,quk,te,ye,ie] = ode45(@mn,tunlockedknee,y0unlockedknee,options);
    nt=length(tuk);
    % result output: quk=[q1 q2 q3 dq1 dq2 dq3]


    %% Knee strike transition
    Hkneestrike=Hk;
    % quk(nt,4:6).'=[dq1 dq2 dq3] at row nt
    qkneestrike=Hkneestrike*quk(nt,4:6).';
    % result output: qkneestrike=[dq1 dq2] (dq3=dq2)


    %% Locked knee dynamics
    % Input conditions yo1=[q1 q2 dq1 dq2]
    y0lockedknee=[quk(nt,1:2), qkneestrike.'];
    % Simulation time span [seconds]
    tlockedknee = [0 1];
    %options used in solving locked knee ode
    options1 = odeset('RelTol',1e-10,'AbsTol',1e-10,...
        'Events',@eventslockedknee,'Refine',refine);


    [tlk,qlk,te1,ye1,ie1] = ode45(@mn1,tlockedknee,y0lockedknee,options1);
    nt1=length(tlk);
    % result output: qlk=[q1 q2 dq1 dq2]


    %% Heel strike transition
    Hheelstrike=Hh;
    % qlk(nt1,3:4).'=[dq1 dq2] at row nt1
    qheelstrike=Hheelstrike*qlk(nt1,3:4).';
    % result output: qheelstrike=[dq1 dq2] (dq3=dq2)
    % new initial condition for next step = [q1 q2 q3 dq1 dq2 dq3]
    y0unlockedknee=[qlk(nt1,2), qlk(nt1,1), qlk(nt1,1), qheelstrike.',
qheelstrike(2).'];


    %% Store results
    t=[t; tuk+t(tlength); tlk+tuk(nt)+t(tlength)];  % time vector
    tlength=length(t);
    % generalised coordinates
    q11=[q11; quk(:,1);qlk(:,1)];       %q1
    q22=[q22; quk(:,2);qlk(:,2)];       %q2
    q33=[q33; quk(:,3);qlk(:,2)];       %q3


    qd11=[qd11; quk(:,4);qlk(:,3)];      %dq1
    qd22=[qd22; quk(:,5);qlk(:,4)];      %dq2
    qd33=[qd33; quk(:,6);qlk(:,4)];      %dq3


end
%% Figure ploting
figure(1)
subplot(2,1,1); plot(t,q11(:,1),'r-')
hold all
subplot(2,1,1); plot(t,q22(:,1),'b-')
hold all
subplot(2,1,1); plot(t,q33(:,1),'g-')
title('q1=red, q2=blue, q3=greeen')
ylabel('Angle [rad]')
subplot(2,1,2); plot(t,qd11(:,1),'r-')
hold all
subplot(2,1,2); plot(t,qd22(:,1),'b-')
hold all
subplot(2,1,2); plot(t,qd33(:,1),'g-')
title('qd1=red, qd2=blue, qd3=greeen')
ylabel('Angular velocity [rad/s]')
xlabel('Time [s]')


    %% Unlocked knee dynamics ODE matrices
```

```
    function MN=mn(tuk,quk)
        %quk=[q1 q2 q3 dq1 dq2 dq3]
        q1=quk(1);
        q2=quk(2);
        q3=quk(3);
        dq1=quk(4);
        dq2=quk(5);
        dq3=quk(6);

        H =[ Is + It + Ih*sign_dq1 + lr^2*mh + lms^2*ms + lmt^2*mt + 2*lr^2*ms +
2*lr^2*mt + lt^2*ms + mh*rs^2 + 2*ms*rs^2 + 2*mt*rs^2 - 2*lr*lt*ms*cos(epst) +
2*lms*lt*ms*cos(thems) - 2*lt*ms*rs*cos(q1) - 2*lms*lr*ms*cos(epst - thems) -
2*lmt*lr*mt*cos(epst - themt) + 2*lr*mh*rs*cos(epst - q1) + 4*lr*ms*rs*cos(epst - q1) +
4*lr*mt*rs*cos(epst - q1) - 2*lms*ms*rs*cos(q1 - thems) - 2*lmt*mt*rs*cos(q1 - themt), -
lr*lt*ms*cos(epst - q1 + q2) - lt*ms*rs*cos(q2) - lmt*mt*rs*cos(q2 - themt) -
lmt*lr*mt*cos(epst - q1 + q2 - themt), - lms*ms*rs*cos(q3 - thems) - lms*lr*ms*cos(epst
- q1 + q3 - thems);...
            - lr*lt*ms*cos(epst - q1 + q2) - lt*ms*rs*cos(q2) - lmt*mt*rs*cos(q2 -
themt) - lmt*lr*mt*cos(epst - q1 + q2 - themt),
mt*lmt^2 + ms*lt^2 + It + Ih*sign_dq2,
lms*lt*ms*cos(q2 - q3 + thems);...
            - lms*ms*rs*cos(q3 - thems) - lms*lr*ms*cos(epst - q1 + q3 - thems),
lms*lt*ms*cos(q2 - q3 + thems),
ms*lms^2 + Is];

        B =[ dq1*(lt*ms*rs*sin(q1) + lr*mh*rs*sin(epst - q1) + 2*lr*ms*rs*sin(epst - q1)
+ 2*lr*mt*rs*sin(epst - q1) + lms*ms*rs*sin(q1 - thems) + lmt*mt*rs*sin(q1 - themt)),
dq2*(lr*lt*ms*sin(epst - q1 + q2) + lt*ms*rs*sin(q2) + lmt*mt*rs*sin(q2 - themt) +
lmt*lr*mt*sin(epst - q1 + q2 - themt)), dq3*(lms*ms*rs*sin(q3 - thems) +
lms*lr*ms*sin(epst - q1 + q3 - thems));...
            -dq1*(lr*lt*ms*sin(epst - q1 + q2) + lmt*lr*mt*sin(epst - q1 + q2 - themt)),
0,                                      dq3*lms*lt*ms*sin(q2 - q3 + thems);...
            -dq1*lms*lr*ms*sin(epst - q1 + q3 - thems),
-dq2*lms*lt*ms*sin(q2 - q3 + thems),
0];

        G =[ g*(lr*mh*sin(epst - gam - q1) + 2*lr*ms*sin(epst - gam - q1) +
2*lr*mt*sin(epst - gam - q1) - mh*rs*sin(gam) - 2*ms*rs*sin(gam) - 2*mt*rs*sin(gam) +
lms*ms*sin(gam + q1 - thems) + lmt*mt*sin(gam + q1 - themt) + lt*ms*sin(gam + q1));
g*(lmt*mt*sin(gam + q2 - themt) + lt*ms*sin(gam + q2));
g*lms*ms*sin(gam + q3 - thems)];

        T=[tau1;tau2;tau3];

        M=[zeros(3) eye(3,3);zeros(3) -H\B];
        N=[zeros(3,1);-H\G+H\T];
        MN(1,1)=M(1,:)*quk+N(1);
        MN(2,1)=M(2,:)*quk+N(2);
        MN(3,1)=M(3,:)*quk+N(3);
        MN(4,1)=M(4,:)*quk+N(4);
        MN(5,1)=M(5,:)*quk+N(5);
        MN(6,1)=M(6,:)*quk+N(6);
    end
    %% Knee strike transition matrices
    function Hkneestrike=Hk
        %quk=[q1 q2 q3 dq1 dq2 dq3]
        q1=quk(nt,1);
        q2=quk(nt,2);
        q3=quk(nt,3);

        Qkneestriken =[ -ms*lt*cos(q2)*rs-2*ms*lt*cos(q1)*rs-ms*lms*lr*cos(-q3+thems+q1-
epst)+2*mh*lr*rs*cos(q1-epst)+ms*lms^2-2*mt*lmt*lr*cos(epst-themt)+mt*lmt^2-
ms*lms*rs*cos(-q3+thems)+ms*lt^2+4*ms*lr*rs*cos(q1-epst)-mt*lmt*rs*cos(-q2+themt)-
ms*lt*lr*cos(-q2+q1-epst)+4*mt*lr*rs*cos(q1-epst)-2*ms*lms*lr*cos(epst-
thems)+2*ms*lr^2+2*mt*lr^2+mh*lr^2-2*ms*lms*rs*cos(q1-
thems)+2*mt*rs^2+mh*rs^2+2*ms*rs^2-2*ms*lr*cos(epst)*lt+2*ms*lt*lms*cos(thems)-
mt*lmt*lr*cos(-q2+themt+q1-epst)-2*mt*lmt*rs*cos(q1-themt),     -
ms*lt*cos(q2)*rs+mt*lmt^2-ms*lt*lr*cos(-q2+q1-epst)-mt*lmt*rs*cos(-
q2+themt)+ms*lms*lt*cos(-q3+thems+q2)-mt*lmt*lr*cos(-q2+themt+q1-epst)+ms*lt^2,    -
ms*lr*lms*cos(-q3+thems+q1-epst)+ms*lms*lt*cos(-q3+thems+q2)-ms*rs*lms*cos(-
q3+thems)+ms*lms^2;...
            -ms*lr*lms*cos(-q3+thems+q1-epst)-mt*lmt*rs*cos(-q2+themt)-ms*lt*cos(q2)*rs-
ms*rs*lms*cos(-q3+thems)-mt*lmt*lr*cos(-q2+themt+q1-epst)-ms*lt*lr*cos(-q2+q1-epst),
```

```
ms*lms*lt*cos(-q3+thems+q2)+mt*lmt^2+ms*lt^2,              ms*lms*lt*cos(-
q3+thems+q2)+ms*lms^2];


        Qkneestrikep =[ 4*ms*lr*rs*cos(q1-epst)+2*mt*rs^2+2*ms*rs^2-2*ms*lt*cos(q1)*rs-
ms*lr*lt*cos(-q2+q1-epst)+ms*lt^2+mh*lr^2-ms*rs*lms*cos(-q2+thems)-ms*lr*lms*cos(q1-
epst-q2+thems)-mt*lr*lmt*cos(-q2+themt+q1-epst)-mt*rs*lmt*cos(-
q2+themt)+2*mt*lr^2+2*ms*lr^2-2*ms*lr*cos(epst)*lt+mh*rs^2-2*ms*lms*rs*cos(q1-
thems)+2*mh*lr*rs*cos(q1-epst)+mt*lmt^2-2*mt*lmt*rs*cos(q1-themt)+4*mt*lr*rs*cos(q1-
epst)-2*mt*lmt*lr*cos(epst-themt)+ms*lms^2-2*ms*lr*lms*cos(epst-
thems)+2*ms*lt*lms*cos(thems)-ms*lt*cos(q2)*rs,     ms*lt^2-ms*lms*lr*cos(q1-epst-
q2+thems)-mt*lmt*rs*cos(-q2+themt)-ms*lms*rs*cos(-q2+thems)+mt*lmt^2-ms*lt*lr*cos(-
q2+q1-epst)-mt*lmt*lr*cos(-q2+themt+q1-epst)+ms*lms^2+2*ms*lt*lms*cos(thems)-
ms*lt*cos(q2)*rs;
            -mt*lmt*rs*cos(-q2+themt)-mt*lmt*lr*cos(-q2+themt+q1-epst)-ms*lms*rs*cos(-
q2+thems)-ms*lms*lr*cos(q1-epst-q2+thems)-ms*lt*lr*cos(-q2+q1-epst)-ms*lt*cos(q2)*rs,
ms*lms^2+mt*lmt^2+2*ms*lt*lms*cos(thems)+ms*lt^2];


        Hkneestrike=(Qkneestrikep)\Qkneestriken;
    end
    %% Locked knee dynamics ODE matrices
    function MN1=mn1(tlk,qlk)
        %qlk=[q1 q2 dq1 dq2]
        q1=qlk(1);
        q2=qlk(2);
        dq1=qlk(3);
        dq2=qlk(4);


        H =[ Is + It + Ih*sign_dq1 + lr^2*mh + lms^2*ms + lmt^2*mt + 2*lr^2*ms +
2*lr^2*mt + lt^2*ms + mh*rs^2 + 2*ms*rs^2 + 2*mt*rs^2 - 2*lr*lt*ms*cos(epst) +
2*lms*lt*ms*cos(thems) - 2*lt*ms*rs*cos(q1) - 2*lms*lr*ms*cos(epst - thems) -
2*lmt*lr*mt*cos(epst - themt) + 2*lr*mh*rs*cos(epst - q1) + 4*lr*ms*rs*cos(epst - q1) +
4*lr*mt*rs*cos(epst - q1) - 2*lms*ms*rs*cos(q1 - thems) - 2*lmt*mt*rs*cos(q1 - themt), -
lr*lt*ms*cos(epst - q1 + q2) - lt*ms*rs*cos(q2) - lms*ms*rs*cos(q2 - thems) -
lmt*mt*rs*cos(q2 - themt) - lms*lr*ms*cos(epst - q1 + q2 - thems) - lmt*lr*mt*cos(epst -
q1 + q2 - themt);...
            - lr*lt*ms*cos(epst - q1 + q2) - lt*ms*rs*cos(q2) - lms*ms*rs*cos(q2 -
thems) - lmt*mt*rs*cos(q2 - themt) - lms*lr*ms*cos(epst - q1 + q2 - thems) -
lmt*lr*mt*cos(epst - q1 + q2 - themt),
ms*lms^2 + 2*ms*cos(thems)*lms*lt + mt*lmt^2 + ms*lt^2 + Is + It + Ih*sign_dq2];


        B =[ dq1*(lt*ms*rs*sin(q1) + lr*mh*rs*sin(epst - q1) + 2*lr*ms*rs*sin(epst - q1)
+ 2*lr*mt*rs*sin(epst - q1) + lms*ms*rs*sin(q1 - thems) + lmt*mt*rs*sin(q1 - themt)),
dq2*(lr*lt*ms*sin(epst - q1 + q2) + lt*ms*rs*sin(q2) + lms*ms*rs*sin(q2 - thems) +
lmt*mt*rs*sin(q2 - themt) + lms*lr*ms*sin(epst - q1 + q2 - thems) + lmt*lr*mt*sin(epst -
q1 + q2 - themt));...
            -dq1*(lr*lt*ms*sin(epst - q1 + q2) + lms*lr*ms*sin(epst - q1 + q2 - thems) +
lmt*lr*mt*sin(epst - q1 + q2 - themt)),
0];


        G =[ g*(lr*mh*sin(epst - gam - q1) + 2*lr*ms*sin(epst - gam - q1) +
2*lr*mt*sin(epst - gam - q1) - mh*rs*sin(gam) - 2*ms*rs*sin(gam) - 2*mt*rs*sin(gam) +
lms*ms*sin(gam + q1 - thems) + lmt*mt*sin(gam + q1 - themt) + lt*ms*sin(gam + q1));
g*(lms*ms*sin(gam + q2 - thems) + lmt*mt*sin(gam + q2 - themt) + lt*ms*sin(gam + q2))];


        T=[tau1;tau2];


        M1=[zeros(2) eye(2,2);zeros(2) -H\B];
        N1=[zeros(2,1);-H\G+H\T];
        MN1(1,1)=M1(1,:)*qlk+N1(1);
        MN1(2,1)=M1(2,:)*qlk+N1(2);
        MN1(3,1)=M1(3,:)*qlk+N1(3);
        MN1(4,1)=M1(4,:)*qlk+N1(4);
    end
    %% Heel strike transition matrices
    function Hheelstrike=Hh
        %qlk=[q1 q2 dq1 dq2]
        q1=qlk(nt1,1);
        q2=qlk(nt1,2);


        Qheelstriken =[ -2*ms*lt*cos(q1)*rs-ms*lt*cos(q2)*rs+2*ms*lr*rs*cos(-q2+epst)-
2*ms*lms*rs*cos(q1-thems)+mh*lr*rs*cos(q1-epst)+mt*lmt^2+ms*lms^2+mh*lr^2*cos(-
q2+q1)+mh*lr*rs*cos(-q2+epst)+ms*lt^2+2*mt*lr*rs*cos(q1-epst)-
ms*lr*cos(epst)*lt+2*ms*lt*lms*cos(thems)+2*mt*lr*rs*cos(-q2+epst)+2*ms*lr*rs*cos(q1-
```

```
epst)-ms*lt*lr*cos(-q2+q1-epst)-ms*lt*lr*cos(-q2+q1+epst)-2*mt*lmt*rs*cos(q1-
themt)+mh*rs^2+2*ms*rs^2+2*mt*rs^2-mt*rs*lmt*cos(-q2+themt)-ms*rs*lms*cos(-q2+thems)-
mt*lr*lmt*cos(epst-themt)-ms*lr*lms*cos(epst-thems)-mt*lmt*lr*cos(-q2+themt+q1-epst)-
mt*lmt*lr*cos(q1+epst-q2-themt)+2*mt*lr^2*cos(-q2+q1)-ms*lms*lr*cos(q1-epst-q2+thems)-
ms*lms*lr*cos(q1+epst-q2-thems)+2*ms*lr^2*cos(-q2+q1),
-ms*lt*cos(q2)*rs+mt*lmt^2+ms*lms^2+ms*lt^2-ms*lr*cos(epst)*lt+2*ms*lt*lms*cos(thems)-
mt*rs*lmt*cos(-q2+themt)-ms*rs*lms*cos(-q2+thems)-mt*lr*lmt*cos(epst-themt)-
ms*lr*lms*cos(epst-thems);...
            -ms*lt*cos(q1)*rs-mt*lmt*rs*cos(q1-themt)-ms*lms*rs*cos(q1-
thems)+mt*lmt^2+ms*lms^2+ms*lt^2-ms*lr*cos(epst)*lt+2*ms*lt*lms*cos(thems)-
mt*lr*lmt*cos(epst-themt)-ms*lr*lms*cos(epst-thems),
0];


        q1=qlk(nt1,2);
        q2=qlk(nt1,1);


        Qheelstrikep =[ 2*ms*lt*lms*cos(thems)-2*ms*lt*lr*cos(epst)+mt*lmt^2-
2*ms*lt*cos(q1)*rs+2*ms*lr^2+ms*lms^2-mt*rs*lmt*cos(-q2+themt)-ms*lr*lt*cos(-q2+q1-
epst)-mt*lr*lmt*cos(-q2+themt+q1-epst)-ms*rs*lms*cos(-q2+thems)-ms*lr*lms*cos(q1-epst-
q2+thems)+2*mt*rs^2+2*ms*rs^2+mh*rs^2-2*ms*lms*lr*cos(epst-thems)-
ms*lt*cos(q2)*rs+ms*lt^2+4*mt*rs*lr*cos(q1-epst)-2*ms*lms*rs*cos(q1-thems)-
2*mt*lmt*rs*cos(q1-themt)+2*mh*rs*lr*cos(q1-epst)+4*ms*rs*lr*cos(q1-epst)-
2*mt*lmt*lr*cos(epst-themt)+2*mt*lr^2+mh*lr^2,
2*ms*lt*lms*cos(thems)+mt*lmt^2+ms*lms^2-mt*rs*lmt*cos(-q2+themt)-ms*lr*lt*cos(-q2+q1-
epst)-mt*lr*lmt*cos(-q2+themt+q1-epst)-ms*rs*lms*cos(-q2+thems)-ms*lr*lms*cos(q1-epst-
q2+thems)-ms*lt*cos(q2)*rs+ms*lt^2;...
            -mt*rs*lmt*cos(-q2+themt)-ms*lr*lt*cos(-q2+q1-epst)-mt*lr*lmt*cos(-
q2+themt+q1-epst)-ms*rs*lms*cos(-q2+thems)-ms*lr*lms*cos(q1-epst-q2+thems)-
ms*lt*cos(q2)*rs,
ms*lms^2+ms*lt^2+mt*lmt^2+2*ms*lt*lms*cos(thems)];


        Hheelstrike=(Qheelstrikep)\Qheelstriken;
    end
    %% Termination of event for unlocked knee dynamics
    %Termination occurs when q2=q3 (swing thigh angle = swing shank anagle)
    function [value,isterminal,direction] = eventsunlockedknee(tuk,quk)
        de =quk(2)-quk(3);
        value = de;
        isterminal = 1;
        direction = 1;
    end
    %% Termination of event for locked knee dynamics
    %Termination occurs when q1=-q2 (stance leg angle = negative swing leg anagle)
    function [value,isterminal,direction] = eventslockedknee(tlk,qlk)
        dde =qlk(1)+qlk(2)-2*epst;
        value = dde;
        isterminal = 1;
        direction = 0;
    end
end
```

Appendix C: Controller Schematics

The schematics of the controller connections are presented in this appendix.

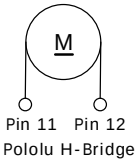

Figure C- 1: Schematic of the DC motor

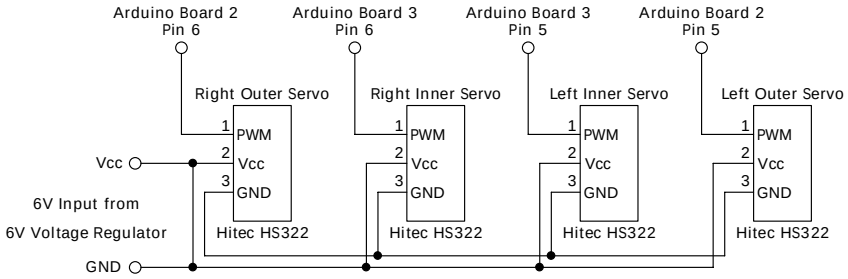

Figure C- 2: Schematic of servos' connection circuit

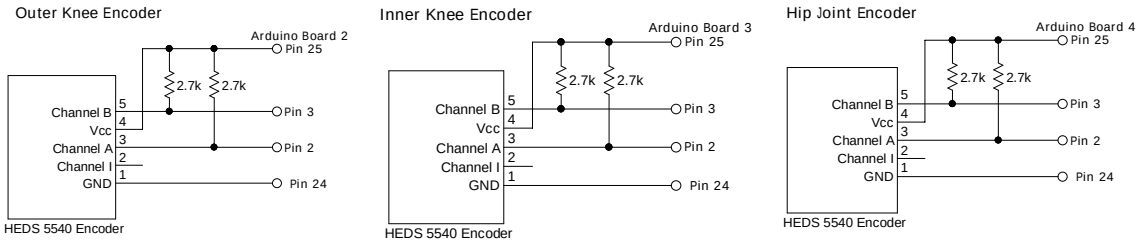

Figure C- 3: Schematics of optical encoders' connection circuit

Figure C- 4: Schematics of accelerometers' connection circuit

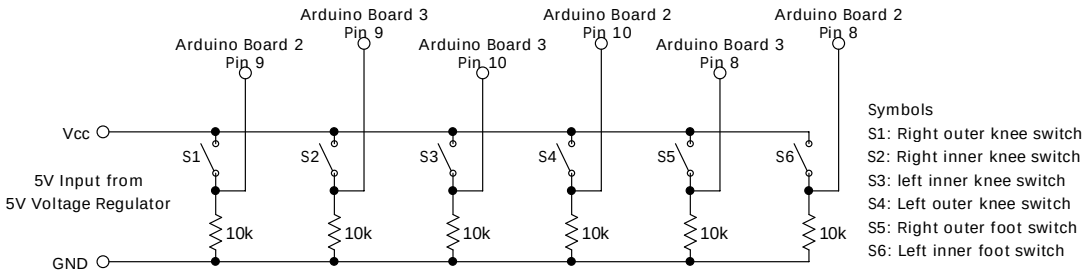

Figure C- 5: Schematic of micro switches' connection circuit

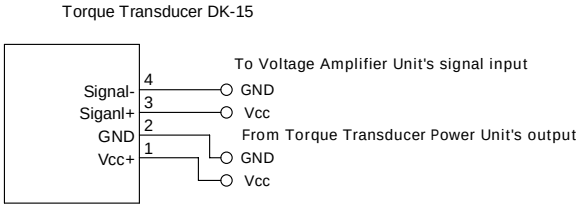

Figure C- 6: Schematic of the torque transducer DK-15

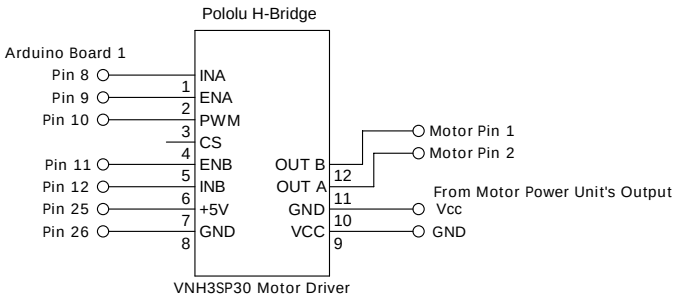

Figure C- 7: Schematics of the Pololu motor driver

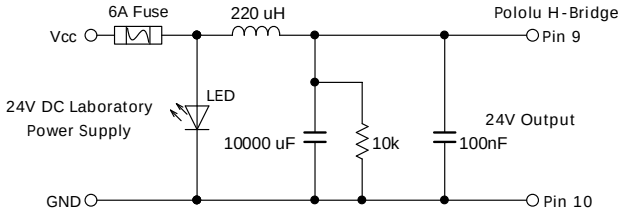

Figure C- 8: Schematic of the motor's power unit connection

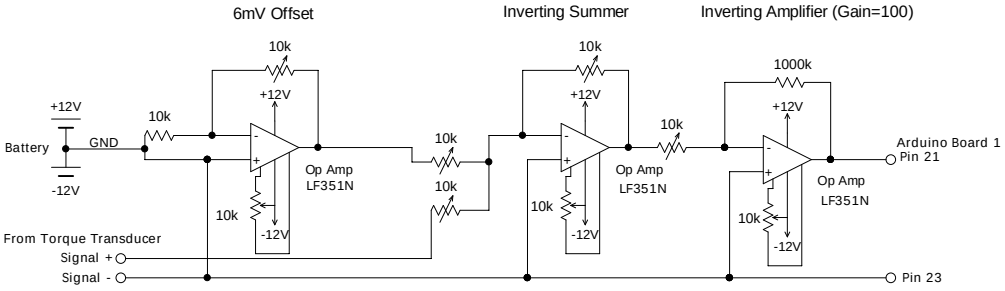

Figure C- 9: Schematic of the voltage amplifier unit

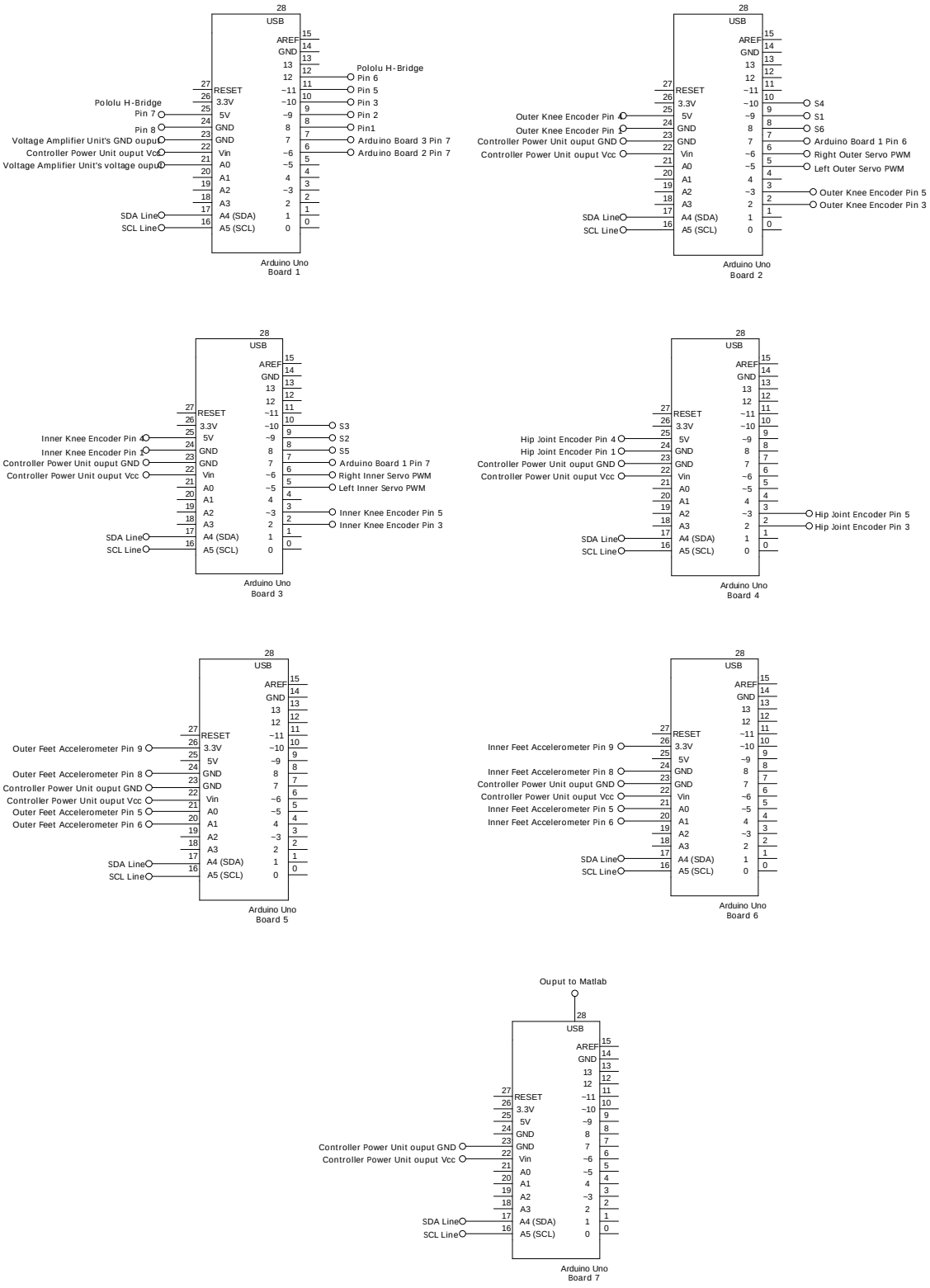

Figure C- 10: Schematics of the Arduino Uno boards

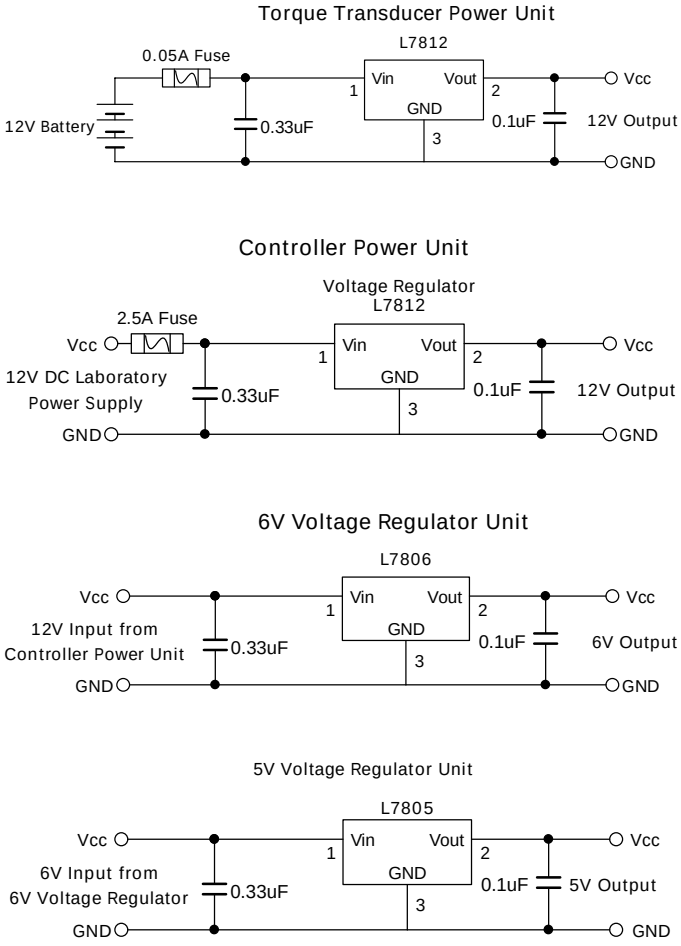

Figure C- 11: Schematics of voltage regulators

Appendix D: Software Codes

The Matlab commands used to read data from the master Arduino board, and write data in Excel format is as follows:

```
% Communications MatLab <--> Arduino
delete(instrfind({'Port'},{'COM15'}))  %clear the use of Port 'COM15'
numSec=5*60; % Recording time in seconds

time=[];
torqueAve=[];
torqueVal=[];
KEO=[];
FSI=[];
KSRO=[];
KEI=[];
FSO=[];
KSRI=[];
FEOx=[];
FEOy=[];
FEIx=[];
FEIy=[];
ME=[];

s1 = serial('COM15');   % Setup Matlab's connection to serial port 'Com15'
s1.BaudRate=9600;       % Set data transfer baud rate
fopen(s1);              % Start connection

i=0;
t0=tic;     % Start Matlab timing
while (toc(t0)<=numSec)
    i=i+1;                          % Datum number
    time(i)=toc;                    % Time [s]
    torqueAve(i)=fscanf(s1,'%d');   % Voltage number at 0Nm torque
    torqueVal(i)=fscanf(s1,'%d');   % Current torque voltage interval
    KEO(i)=fscanf(s1,'%d');         % Outer knee encoder pulse count
    FSI(i)=fscanf(s1,'%d');         % Inner foot switch condition
    KSRO(i)=fscanf(s1,'%d');        % Outer knee switch condition
    KEI(i)=fscanf(s1,'%d');         % Inner knee encoder pulse count
    FSO(i)=fscanf(s1,'%d');         % Outer foot switch condition
    KSRI(i)=fscanf(s1,'%d');        % Inner knee switch condition
    FEOx(i)=fscanf(s1,'%d');        % Outer accelerometer x-axis voltage
    FEOy(i)=fscanf(s1,'%d');        % Outer accelerometer y-axis voltage
    FEIx(i)=fscanf(s1,'%d');        % Inner accelerometer x-axis voltage
    FEIy(i)=fscanf(s1,'%d');        % Inner accelerometer y-axis voltage
    ME(i)=fscanf(s1,'%d');          % Hip joint encoder pulse count
end
fclose(s1); % Stop connection
delete(instrfind({'Port'},{'COM15'}))   % Disconnect serial Port 'COM15'

%Write data to excel
xlswrite('walk_data_1.xls', time, 'time', 'A1');
xlswrite('walk_data_1.xls', torqueAve, 'torqueAve', 'A1');
xlswrite('walk_data_1.xls', torqueVal, 'torqueVal', 'A1');
xlswrite('walk_data_1.xls', KEO, 'KEO', 'A1');
xlswrite('walk_data_1.xls', FSI, 'FSI', 'A1');
xlswrite('walk_data_1.xls', KSRO, 'KSRO', 'A1');
xlswrite('walk_data_1.xls', KEI, 'KEI', 'A1');
xlswrite('walk_data_1.xls', FSO, 'FSO', 'A1');
xlswrite('walk_data_1.xls', KSRI, 'KSRI', 'A1');
xlswrite('walk_data_1.xls', FEOx, 'FEOx', 'A1');
xlswrite('walk_data_1.xls', FEOy, 'FEOy', 'A1');
xlswrite('walk_data_1.xls', FEIx, 'FEIx', 'A1');
xlswrite('walk_data_1.xls', FEIy, 'FEIy', 'A1');
xlswrite('walk_data_1.xls', ME, 'ME', 'A1');
```

The Arduino Uno boards' coding used for the routine control of the biped is as follows:

Arduino Uno board 1 Codes:

```
#include <Wire.h>

volatile  int pwm=0; // PWM [0 to 255]
volatile  int pwm1=0; // PWM [0 to 255]
volatile int readVal;
long valSum=0;
long noOfVal=0;
volatile int torqueAve;
volatile int torqueVal;
volatile int torqueDiff;
int intHeel=0;
int extHeel=0;
int SampleTime = 1000;
int SampleTime1 = 1000;
int intLeg=1;
int extLeg=1;

void setup() {
  Wire.begin(3); // join to I2C bus with address #3
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
  pinMode(6, INPUT); // Outer foot switch
  pinMode(7, INPUT); // Inner foot switch
  pinMode(8, OUTPUT); //InA (H-bridge)
  pinMode(9, OUTPUT); //EnA (H-bridge)
  pinMode(10, OUTPUT); //PWM (H-bridge)
  pinMode(11, OUTPUT); //EnB (H-bridge)
  pinMode(12, OUTPUT); //InB (H-bridge)
  digitalWrite(9, HIGH);
  digitalWrite(11, HIGH);

  analogReference(INTERNAL); //set analog read's reference voltage to 1.1Volts
  while(noOfVal < 1000){
    readVal = analogRead(0);
    valSum = valSum + readVal;
    noOfVal = noOfVal + 1;
  }
  torqueAve=valSum/noOfVal; // interger value of torque voltage at 0Nm

  pwm = 255;
  SampleTime= 115; // [ms]
  pwm1= 5;
  SampleTime1=135; // [ms]
}

void loop() {
  int extHeel = digitalRead(6);  // Outer foot switch
  int intHeel = digitalRead(7);  // Inner foot switch

  // If outer heel strike occurs, rotate motor in forward direction
  if (extHeel>=1  && extLeg ==1) {
    digitalWrite(8, HIGH); //InA
    digitalWrite(12, LOW); //InB
    analogWrite(10,pwm); // fast initial leg swing
    delay(SampleTime);
    analogWrite(10,pwm1); // slow secondary leg swing
    delay(10);
    analogWrite(10,0); //Stop (pin no, pwm value)
    extLeg=0;
    intLeg=1;
  }

  // If inner heel strike occurs, rotate motor in reverse direction
  if (intHeel>=1  && intLeg ==1) {
    digitalWrite(8, LOW); //InA
    digitalWrite(12, HIGH); //InB
    analogWrite(10,pwm); // fast initial leg swing
    delay(SampleTime1);
    analogWrite(10,pwm1); // slow secondary leg swing
    delay(10);
    analogWrite(10,0); //Stop (pin no, pwm value)
```

```
    extLeg=1;
    intLeg=0;
  }

  valSum=0;
  noOfVal=0;
  while(noOfVal < 6){
    readVal = analogRead(0);
    valSum = valSum + readVal;
    noOfVal = noOfVal + 1;
  }
  torqueVal=valSum/noOfVal; //measure current torque value
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr1[2] = {
    torqueAve, torqueVal      };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr1, sizeof(arr1));
}
```

Arduino Uno board 2 Codes:

```
#include <Servo.h>
#include <Wire.h>

volatile int encoder0Pos = 0;
volatile boolean PastA = 0;
volatile boolean PastB = 0;
volatile int intHeel = 0; // internal heelstrike: 0= no stirke, 1=strike
volatile int extKnee = 0; // external kneestrike: 0= no stirke, 1=strike
int val7stat=0;
int val6stat=0;
Servo RightExtServo; // Right External Servo
Servo LeftExtServo; // Left External Servo

void setup() {
  Wire.begin(7); // join to I2C bus with address #7
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
  RightExtServo.attach(6);
  LeftExtServo.attach(5);
  //Turn servos to unlock position
  RightExtServo.write(65);
  LeftExtServo.write(25);
  pinMode(2, INPUT); // Encoder Channel A
  pinMode(3, INPUT); // Encoder Channel B
  pinMode(7, OUTPUT);      // Write digital state of the inner foot switch
  pinMode(8, INPUT);      // Ineer foot swirch
  pinMode(9, INPUT);      // Outer right knee switch
  pinMode(10, INPUT);      // Outer left knee switch
  PastA = (boolean)digitalRead(2);
  PastB = (boolean)digitalRead(3);
  attachInterrupt(0, doEncoderA, RISING);
  attachInterrupt(1, doEncoderB, CHANGE);
}

void doEncoderA() {
  PastB ? encoder0Pos--:  encoder0Pos++;
}

void doEncoderB() {
  PastB = !PastB;
}

void loop() {
  int val5 = digitalRead(8);   // Ineer foot switch
  int val6 = digitalRead(9);   // Outer right knee switch
  int val7 = digitalRead(10);  // Outer left knee switch

  // Innner foot heelstrike: unlock outer knees, send heelstrike signal
  if (val5 ==1) {
    RightExtServo.write(65);
```

```
    LeftExtServo.write(25);
    digitalWrite(7, HIGH);  // Write digital state of the inner foot switch
    intHeel=1; // heelstrike
    delay(15); // hold signal for some time
    intHeel=0; // end heel strike
    digitalWrite(7, LOW);  // Write digital state of the inner foot switch
    delay(600); // hold servo release for some time
    val7stat=0; // change knee status
    val6stat=0; // change knee status
  }

  // Outer left kneestike: lock outer left knee
  if (val7 == 1) {
    LeftExtServo.write(0); // lock knee
    encoder0Pos = 0; //calibrate knee encoder to 0 degree
    delay(150); // wait for some time
    int val7 = digitalRead(10);   // Outer left knee switch
    if (val7 == 1) {
      val7stat=1; // leave knee locked if switch pressed
    }
    if (val7 == 0) {
      if (val7stat == 0) {
        LeftExtServo.write(25); // open knee if switch unpressed
        val7stat=1; // change knee status
      }
    }
  }

  // Outer right kneestike: lock outer right knee, send kneestrike signal
  if (val6 == 1) {
    RightExtServo.write(90); // lock knee
    extKnee=1; // kneestrike
    delay(15); // hold signal for some time
    extKnee=0; // end knee strike
    delay(150); // wait for some time
    int val6 = digitalRead(9);   // Outer right knee switch
    if (val6 == 1) {
      val6stat=1; // leave knee locked if switch pressed
    }
    if (val6 == 0) {
      if (val6stat == 0) {
        RightExtServo.write(65); // open knee if switch unpressed
        val6stat=1; // change knee status
      }
    }
  }
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr2[3] = {
    encoder0Pos, intHeel, extKnee        };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr2, sizeof(arr2));
}
```

Arduino Uno board 3 Codes:

```
#include <Servo.h>
#include <Wire.h>

volatile int encoder0Pos = 0;
volatile boolean PastA = 0;
volatile boolean PastB = 0;
volatile int extHeel = 0; // external heelstrike: 0= no stirke, 1=strike
volatile int intKnee = 0; // internal kneestrike: 0= no stirke, 1=strike
int val7stat=0;
int val6stat=0;
Servo RightIntServo; // Right Internal Servo
Servo LeftIntServo; // Left Internal Servo

void setup() {
  Wire.begin(9); // join to I2C bus with address #9
  Wire.onRequest(requestEvent);
```

```
  Serial.begin(9600);
  RightIntServo.attach(6);
  LeftIntServo.attach(5);
  //Turn servos to unlock position
  RightIntServo.write(25);
  LeftIntServo.write(65);
  pinMode(2, INPUT); // Encoder Channel A
  pinMode(3, INPUT); // Encoder Channel B
  pinMode(7, OUTPUT);      // Write digital state of the outer foot switch
  pinMode(8, INPUT);      // Outer foot switch
  pinMode(9, INPUT);      // Inner right knee switch
  pinMode(10, INPUT);     // Inner left knee switch
  PastA = (boolean)digitalRead(2);
  PastB = (boolean)digitalRead(3);
  attachInterrupt(0, doEncoderA, RISING);
  attachInterrupt(1, doEncoderB, CHANGE);
}

void doEncoderA() {
  PastB ? encoder0Pos--:  encoder0Pos++;
}

void doEncoderB() {
  PastB = !PastB;
}

void loop() {
  int val5 = digitalRead(8);   // Outer foot switch
  int val6 = digitalRead(9);   // Inner right knee switch
  int val7 = digitalRead(10);  // Inner left knee switch

  // Outer foot heelstrike: unlock inner knees, send heelstrike signal
  if (val5 ==1) {
    RightIntServo.write(25);
    LeftIntServo.write(65);
    digitalWrite(7, HIGH);  // Write digital state of the outer foot switch
    extHeel=1; // heelstrike
    delay(15); // hold signal for some time
    extHeel=0; // end heel strike
    digitalWrite(7, LOW);  // Write digital state of the outer foot switch
    delay(600); // hold release for some time
    val7stat=0; // change knee status
    val6stat=0; // change knee status
  }

  // Inner left kneestike: lock inner left knee
  if (val7 == 1) {
    LeftIntServo.write(90); // lock knee
    delay(150); // wait for some time
    int val7 = digitalRead(10);   // Inner left knee switch
    if (val7 == 1) {
      val7stat=1; // leave knee locked if switch pressed
    }
    if (val7 == 0) {
      if (val7stat == 0) {
        LeftIntServo.write(65); // open knee if switch unpressed
        val7stat=1; // change knee status
      }
    }
  }

  // internal right kneestike: lock internal right knee
  if (val6 == 1) {
    RightIntServo.write(0); // lock knee
    encoder0Pos = 0; //calibrate knee encoder to 0 degree
    intKnee=1; // kneestrike
    delay(15); // hold release for some time
    intKnee=0; // end knee strike
    delay(150); // wait for some time
    int val6 = digitalRead(9);   // Inner right knee switch
    if (val6 == 1) {
      val6stat=1; // leave knee locked if switch pressed
    }
    if (val6 == 0) {
      if (val6stat == 0) {
        RightIntServo.write(25); // open knee if switch unpressed
        val6stat=1; // change knee status
```

```
      }
    }
  }
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr2[3] = {
    encoder0Pos, extHeel, intKnee                    };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr2, sizeof(arr2));
}
```

Arduino Uno board 4 Codes:

```
#include <Wire.h>

volatile int encoder0Pos = 0;
volatile boolean PastA = 0;
volatile boolean PastB = 0;

void setup() {
  Wire.begin(12); // join to I2C bus with address #12
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
  pinMode(2, INPUT); // Encoder Channel A
  pinMode(3, INPUT); // Encoder Channel B
  PastA = (boolean)digitalRead(2);
  PastB = (boolean)digitalRead(3);
  attachInterrupt(0, doEncoderA, RISING);
  attachInterrupt(1, doEncoderB, CHANGE);
}


void doEncoderA() {
  PastB ? encoder0Pos--:  encoder0Pos++;
}

void doEncoderB() {
  PastB = !PastB;
}

void loop() {
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr1[1] = {
    encoder0Pos    };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr1, sizeof(arr1));
}
```

Arduino Uno board 5 Codes:

```
#include <Wire.h>

volatile int sensorValuex = 0;
volatile int sensorValuey = 0;
volatile int readValx;
volatile int readValy;
long valSumx=0;
long valSumy=0;
long noOfVal=0;

void setup() {
  Wire.begin(8); // join to I2C bus with address #8
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
}

void loop() {
  valSumx=0;
  valSumy=0;
```

```
  noOfVal=0;

  while(noOfVal < 11){
    readValx = analogRead(0);
    readValy = analogRead(1);
    valSumx = valSumx + readValx;
    valSumy = valSumy + readValy;
    noOfVal = noOfVal + 1;
  }
  sensorValuex=valSumx/noOfVal;
  sensorValuey=valSumy/noOfVal;
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr1[2] = {
    sensorValuex, sensorValuey      };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr1, sizeof(arr1));
}
```

Arduino Uno board 6 Codes:

```
#include <Wire.h>

volatile int sensorValuex = 0;
volatile int sensorValuey = 0;
volatile int readValx;
volatile int readValy;
long valSumx=0;
long valSumy=0;
long noOfVal=0;

void setup() {
  Wire.begin(10); // join to I2C bus with address #10
  Wire.onRequest(requestEvent);
  Serial.begin(9600);
}

void loop() {
  valSumx=0;
  valSumy=0;
  noOfVal=0;

  while(noOfVal < 11){
    readValx = analogRead(0);
    readValy = analogRead(1);
    valSumx = valSumx + readValx;
    valSumy = valSumy + readValy;
    noOfVal = noOfVal + 1;
  }
  sensorValuex=valSumx/noOfVal;
  sensorValuey=valSumy/noOfVal;
}

void requestEvent() {
  //arr[ number of elements in array ] = {1, 2, 3, ...}
  int arr1[2] = {
    sensorValuex, sensorValuey      };
  // Wire.send((uint8_t *)&ArrayName, sizeof(ArrayName));
  Wire.send((uint8_t *)&arr1, sizeof(arr1));
}
```

Arduino Uno board 7 Codes:

```
#include <Wire.h>

void setup() {
  Wire.begin(11); // join to I2C bus with address #11
  Serial.begin(9600);
}

void loop() {
  Wire.requestFrom(3, 4); // (slave address, no. of byte number)
```

```
if(4 <= Wire.available())    // if 4 bytes were received
{
  byte c = Wire.receive(); // receive a byte as byte
  byte d = Wire.receive();
  byte e = Wire.receive();
  byte f = Wire.receive();
  int torqueAve = d<<8 | c;  //comibine two bytes into integer
  int torqueVal = f<<8 | e;
  // send the reading to Matalb
  Serial.println(torqueAve);   // Interger at 0Nm torque
  Serial.println(torqueVal);   // Current torque value

  Wire.requestFrom(7, 6);
  if(6 <= Wire.available())    // if 6 bytes were received
  {
    byte c = Wire.receive();
    byte d = Wire.receive();
    byte e = Wire.receive();
    byte f = Wire.receive();
    byte g = Wire.receive();
    byte h = Wire.receive();
    int KEO = d<<8 | c;
    int FSI = f<<8 | e;
    int KSRO = h<<8 | g;
    Serial.println(KEO);   // Outer knee encoder
    Serial.println(FSI);   // Inner foot switch
    Serial.println(KSRO);   // outer right knee switch

    Wire.requestFrom(9, 6);
    if(6 <= Wire.available())    // if 6 bytes were received
    {
      byte c = Wire.receive();
      byte d = Wire.receive();
      byte e = Wire.receive();
      byte f = Wire.receive();
      byte g = Wire.receive();
      byte h = Wire.receive();
      int KEI = d<<8 | c;
      int FSO = f<<8 | e;
      int KSRI = h<<8 | g;
      Serial.println(KEI);   //Inner knee encoder
      Serial.println(FSO);   //Outer foot switch
      Serial.println(KSRI);   //Onner right knee switch
      Wire.requestFrom(8, 4);
      if(4 <= Wire.available())    // if 4 bytes were received
      {
        byte c = Wire.receive();
        byte d = Wire.receive();
        byte e = Wire.receive();
        byte f = Wire.receive();
        int FEOx = d<<8 | c;
        int FEOy = f<<8 | e;
        Serial.println(FEOx);   // Outer foot accelerometer x-axis volatge
        Serial.println(FEOy);   // Outer foot accelerometer y-axis volatge
        Wire.requestFrom(10, 4);
        if(4 <= Wire.available())    // if 4 bytes were received
        {
          byte c = Wire.receive();
          byte d = Wire.receive();
          byte e = Wire.receive();
          byte f = Wire.receive();
          int FEIx = d<<8 | c;
          int FEIy = f<<8 | e;
          Serial.println(FEIx);   // Inner foot accelerometer x-axis volatge
          Serial.println(FEIy);   // Inner foot accelerometer x-axis volatge

          Wire.requestFrom(12, 2);
          if(2 <= Wire.available())    // if 2 bytes were received
          {
            byte c = Wire.receive();
            byte d = Wire.receive();
            int ME = d<<8 | c;
            Serial.println(ME);   // Hip joint encoder
          }// B12
        }// B10
      }// B8
    }// B9
```

```
    }// B7
  }// B3
}
```

Appendix E: Biped Assembly Views

Figure E- 1: a) Isometric view from right, b) Isometric view from left

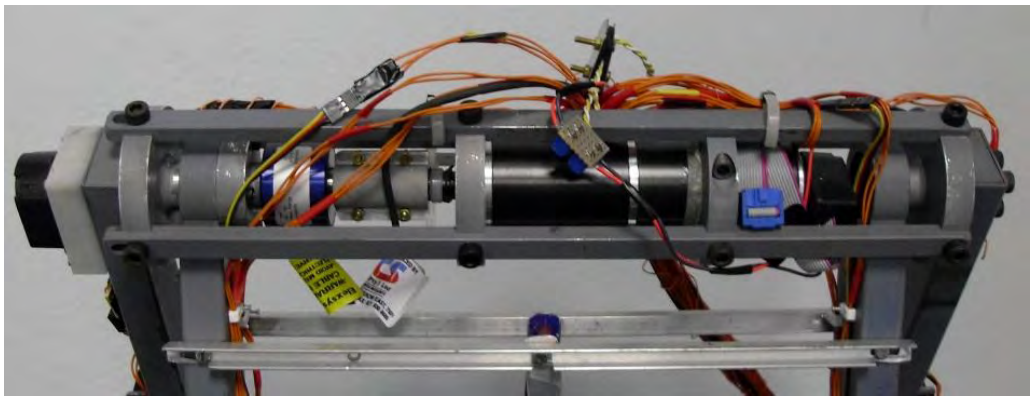

Figure E- 2: Hip segment view

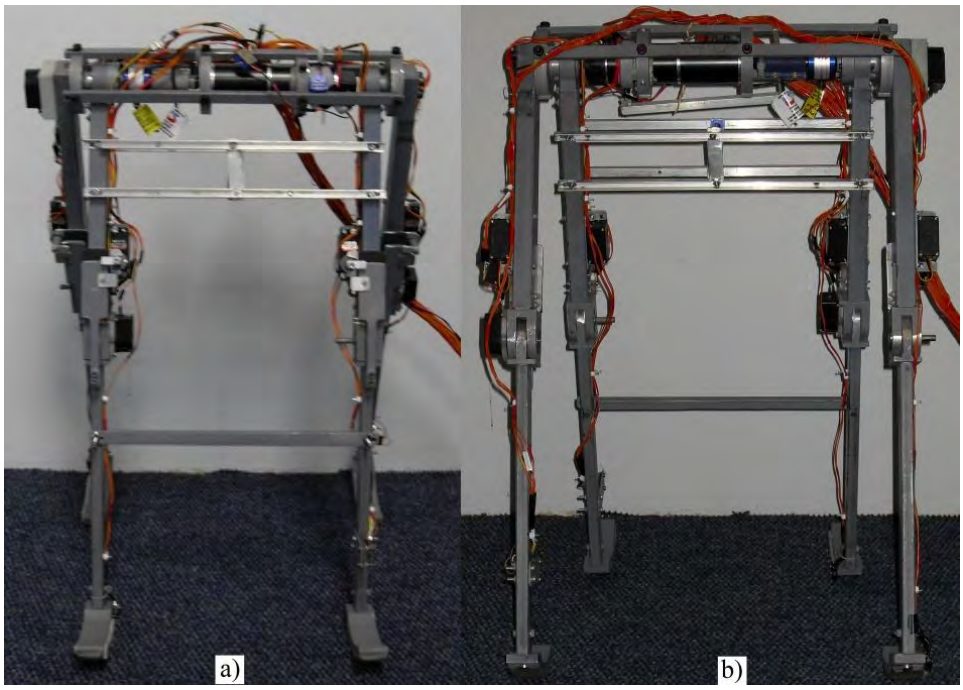

Figure E- 3: a) Front view, b) Back view

Figure E- 4: a) Left side view b) Right side view