Lakehead University

Knowledge Commons,http://knowledgecommons.lakeheadu.ca

Electronic Theses and Dissertations | Electronic Theses and Dissertations from 2009

2015-06-15

Balance control of a five-DOF robot leg

Abela, Joshua L.

http://knowledgecommons.lakeheadu.ca/handle/2453/629

Downloaded from Lakehead University, KnowledgeCommons

Balance Control of a Five-DOF Robot Leg

by
Joshua L. Abela

A Thesis
Presented to Lakehead University
in Partial Fulfillment of the Requirement for the Degree of
Master of Science
in
Control Engineering

Thunder Bay, Ontario, Canada

05/08/2015

Abstract

A biped robot, also known as a humanoid robot, is built to resemble the shape and perform the actions of the human body. While functioning, a biped robot interacts with surrounding human environments. Currently, various robots have been developed to resemble many parts of the human body, such as the head or torso. This material focuses on the development of one robotic leg. Research in humanoid robots will expand knowledge of the human body, while producing greater understanding of the precise motions of the human gait. The field of research in biped robots is very interesting, and creating something similar to that of the human body is a challenging task. The concept of walking robots is motivating and interesting enough, to perform research in the field.

There are two different robot designs, one for the simulation based purposes and the other for real-time data collection. The simulations will be used to help understand the formulas that were developed and researched, in order to control a biped robot. These methods include the Denavit-Hartenberg parameters, Newton-Euler Recursion, Trajectory Generation, Center of Mass and Zero Moment Point.

The second robot design, which provides real-time data collection, will be done on a single five degree of freedom legged robot. This robot leg is equipped with a motor and encoder at each joint that will be used to move and track its position. The foot has four force moment sensors on the bottom of the foot that will be used to help balance the robot leg in the upright position. Since its only a single legged robot, balance is its primary objective.

Biographical Summary

Joshua L. Abela was born on March 23, 1989 in Windsor Ontario. He grew up in the southern Canadian city with a close and very supportive family. Along with his parents, Larry and Colleen, and sister, Felicia, Joshua was blessed to be surrounded by his four endlessly supportive grandparents, George and Josephine Abela, and Alan and Margaret Holmes. Upon graduation of high school, he pursued his interest in electronics at Saint Clair College. The course was a 3 year program that awarded him with an Advanced Diploma in Electronic Engineering- Industrial Automation. After his graduation from College, Joshua applied to the university transition program at Lakehead University with his two close friends. After being granted acceptance, Joshua moved to Thunder Bay, to continue his educational career at Lakehead University. After two and a half years his undergraduate degree was obtained and the pursuit for a graduate degree started. After applying and being accepted, Joshua began his research in biped robots under the supervision Dr.Liu. Upon completion of the MSc, Joshua's third major educational accomplishment, he will pursue employment in the research and development field.

Acknowledgments

I would like to thank everyone that has helped me, in one way or another, throughout my time as a graduate student.

I would also like to thank Dr. Xiaoping Liu for all the hours of help and guidance during my project. I would also like to thank Kailash Bhatia for helping me in the machining of parts and for letting me use the machine shop.

Lastly I would like to thank Janelle Martineau for waiting patiently for me to finish my graduate program and for assisting me in the editing stage of my thesis.

Joshua Abela

Contents

List of Figures

List of Tables

List of Abbreviations

ADC – Analog to Digital.
ASIMO – Advanced Step in Innovative Mobility.
CMOS – Complementary Metal-Oxide Semiconductor.
COM – Center of Mass.
COP – Center of Pressure.
CPU – Central Processing Unit.
DC – Direct Current.
DOF – Degree of Freedom.
DSP – Double Support Phase.
GPIO – General Purpose Input Output.
GRF – Ground Reaction Force.
IEEE – Institute of Electrical and Electronics Engineers.
IO – Inpout Output.
LIPM – Linear Inverted Pendulum Model.
MCU – Multicore Microcontroller Unit.
PC – Personal Computer.
PD – Proportional Derivative.
PID – Proportional Integral Derivative.
PWM – Pulse Width Modulation.
RAM – Random Access Memory.
SP – Support Polygon.
SSP – Single Support Phase.
TTL – Transistor to Transistor Logic.
USB – Universal Serial Bus.
ZMP – Zero Moment Point.

Chapter 1

Literature Review

1.1 Motivation

Since the beginning of time man has always looked for an easier way to perform or carry out a task. Robotics is just one of the many ideas that were created as a result of this motivation. Research in robotics is still prevalent in todays age, possibly more now than ever before, even with the amazing strides that have been made in the field.

There are many different purposes for the development of robots and humanoids. They can be used to perform tasks that may be dangerous for humans or to work alongside with humans.

One workplace where robotics have been utilized for many years is the automotive industry. Robots can perform jobs better than humans without the need to take breaks and lower error. This may lead to the loss of certain jobs, but will create jobs for people who invent and create more efficient robots.

1.2 Trajectory Generation

There are many different types of robots that are created and developed each and everyday. The biped robot is a very highly researched area with new developments happening all the time. One of the first recorded publications for the Institute of Electrical and Electronics Engineers (IEEE) in the field of walking robots was in 1977 [14]. This short paper discussed an approach that applies the principles of Lagrangian dynamics to derive the equations of motion of locomotion gaits, multivariable feedback to design locomotion controls and state-variable techniques to analyze locomotion dynamics. One of the most simplified biped models was used in this paper.

The robot had no knees and no feet and it was restricted to the sagittal plane. One of the goals of this short paper was to design a walking gait that would be similar to a human walking speed and stride length.

Many papers have been written and discussed for the generation of trajectory for biped walking robots. Designing the robots gait is a very crucial step in the field of research of biped robots. The trajectory generation must be designed so that all the joints of the bipeds legs move in a human like walking fashion while it stays balanced and under control. Increased motion and movement in the bipeds leg will occur, with an increased number of joints. This all can be accompanied by the efficiency and energy consumption of the walking pattern that is designed. Energy consumption can be simulated before the build process so you will know how much energy will be used and if any changes need to be made to the design [25].

Many papers discuss experiments that have designed the robots gait through generating trajectories for each joint. In paper [6], by Feng and Sun, a simple trajectory generation is used for the design of the walking gait. Their design only needs three trajectory control parameters, which are the length of a step, max height of swing foot and the stepping cycle. This paper presents control in both the lateral and sagittal plane. The trajectory of the stance leg is generated by the "Three Dimensional Linear Inverted Pendulum Mode" [6] and the swing leg by a combination of sinusoid curves.

In [34], the paper aims to analyze the ground reaction force in terms of smooth transition. The landing impact under the parametric change, for the frontal gait trajectory, is realized to be the inverted pendulum model, with the parametric change in the proportional gain of the PID controller. The paper focuses on the ground reaction force (GRF) and parametric effect of controller gain, and the frontal plane trajectory on GRF, in addition to the gait trajectory planning. A simulation is presented and the comparison of the parametric change is discussed.

Paper [33], presents a humanoid walking pattern generated by the ground reaction force (GRF). By investigation ground reaction force can be obtained from the walking gait. A novel method can be designed from these forces, to design humanoid zero moment point (ZMP) trajectories and to mimic the human ZMP trajectories. A robot model is presented for the alternating support leg, to smoothly avoid the contact impact. The proposed methods are verified by dynamic simulation, and a walking experiment on a real humanoid robot.

In paper [9], a real-time walking pattern generation for humanoid robots using the linear inverted pendulum model (LIPM) is discussed. The LIPM hs two drawbacks, such as instability and non-minimum phase property. This paper proposes a new real-time approach to resolve these issues, by combining a feedback and feedforward controller. The feedback controller will employ pole placement to shift the poles of the LIPM, and improve system stability. The feed-forward controller, will use advanced pole-zero cancellation to reduce the non-minimum phase property. The efficiency is verified by walking pattern planning examples, and experiments with a humanoid robot.

The control of a biped humanoid is challenging task due to the hard-to-stabilize dynamics. Paper [5], proposes a reference generation algorithm, based on the LIPM and a moving support foot ZMP references. The Fourier series approximation simplifies the solution, and generates a smooth ZMP reference.

The development of a real-time motion generation method, that controls the center of gravity (COG) by indirect manipulation of the ZMP is proposed in [27]. The real-time response will provide humanoid robots with high-mobility. The algorithm is presented in four parts; referential ZMP planning, ZMP manipulation, COG velocity decomposition to joint angles and local control of joint angles. The paper proved its effectiveness by computer simulation.

1.3 Trajectory Tracking

Once the trajectory is generated, the next step would be to track, it to ensure the robot is following it correctly and that it doesn't get interrupted. There are a variety of controllers that can be used, such as proportional derivative (PD) [22], proportional derivative integral (PID) [23], fuzzy and computed torque[24] just to name a few. The controller will be in charge of directly controlling the joint angles and using the error in a closed loop system.

Paper [28], presents a new methodology to solve the inverse dynamics of the humanoid robot. The objective was to calculate the joint's torque in order to have an optimal robot gait. The inverse dynamics is separated into two parts the upper and lower joints. The distribution is based on the reaction force calculation for each supported leg. With this process the optimal robot gait is possible [28].

In paper [17], a design of a six-axis force sensor that measures the ground reaction force of

human or humanoid robot is shown. The concept of the design uses parallel support mechanisms, that allow large torques and forces when the foot hits the ground. The mechanism realizes high impact tolerance for desired components of force. The design is light and thin, so it can fit between the sole and ankle joints.

Another design for force sensors is presented in [2]. The design is used to measure the plantar distributed normal forces and friction, during human walking. The results lead to an improved ZMP calculation that is easily programmable. The new method has many advantages including; light and novel structure, low coupling, easy for programming, less affection on normal walking, suited for long-time and diverse terrain testing.

Paper [15], presents a method for measuring the ZMP throughout the whole walking phase. The system uses a biped robot, that has a ZMP measurement system, using two universal force moment sensors. The accuracy of the ZMP measurement system is experimentally confirmed and the ZMP throughout the whole walking phase is measured.

1.4 System Stability and Balance Control

The robot must maintain system stability and balance control, without these the robot could sway or fall off course. Different types of sensors and control schemes can be implemented to prevent balance and stability failure from occurring. One type of sensor is the *F*orce Moment Sensor that is placed on the bottom of the bipeds feet.

One type of force moment sensors is the force sensing resistor [4]. Force sensing resistors are polymer thick film devices which show a decrease in their terminal resistance whenever there is a increasing force applied to their surface. These force sensors can be used to determine where the center of pressure is located on the foot.

Once the disturbance is located, there are a couple of different ways to tackle this problem. The first is to move the leg joints in such a way that it will move the center of pressure of the robot toward the center of the foot for better balance. The second method is more dynamic in which you could accelerate forward to compensate for the unbalanced leg.

In paper [31], by Wei, discuss a criterion for balance recovery through analysis of initial phase plane based on Capture Points, and give step-out strategy for push recovery. This paper will be using the inverted pendulum mode including the angular moment. The step length and step

amount are a crucial role in the balance recovery. Step-one-out strategy is presented for balance recovery planning thats based on the analytical method.

Paper [20] discuss the foces acting on the center of pressure (COP) and ZMP. In the paper the COP is defined as the ground-feet contact forces and the ZMP as the gravity plus inertia forces. The ZMP and COP is proven to be related and control aspects are examined. A virtual COP-ZMP is defined, allowing the robot to traverse on uneven terrain.

Paper [8], treats locomotion dynamics relative to planar motion, under the assumption that, the leg masses can be ignored compared to the body mass. The biped will possess one rotational degree of freedom (DOF), and two translational DOF, leading to a sixth-order nonlinear differential equations. The nonlinear equations will be linearized and feedback control laws are then derived to produce a desirable, stable, forward motion.

1.5 Control System

In [30], the paper presents a control system thats energy-efficient. An energy-efficient support vector machine (EE-SVM) learning control system, considers the energy cost of each sample from the bipeds dynamics. The energy cost of the biped walking are calculated, and then the samples are weighted, with the inverse of the energy costs. Then the EE-SVM objective function will follow the principle that, the sample with the lowest energy consumption will be treated as the most important. This will highly increase the energy efficiency of the biped walking.

To have stable control of humanoid robots, ground contact forces should be controlled for compensating the dynamic disturbances by unactuated body movement. The stability in the sense of ZMP, guarantees secure contact during control, a necessary condition for stable motion. Paper [18], proposes a method to control the GRF at the ZMP by, modifying the systems reference acceleration.

A method for the design of a dynamic biped machine is presented in [7]. A minimum energy criteria is assumed. There is a conflict between the minimum energy design and the stability, as shown in the paper. This will then lead to the discussion of a more stable, dynamic biped machine. A new set of control laws are derived for the locomotion of the system. This whole paper focuses on the control of the system without any external sensors. This type of application would be perfect for prosthetics and orthotics.

To reduce the complex walking dynamics of a biped, simple linear differential equations were derived. Paper [12], uses a particular class of trajectories from an ideal biped model, where the COG of the biped moves horizontally. The term "potential energy conserving orbit" is introduced to describe this class of trajectories. Control laws were then formulated for walk initiation, walk continuation, and walk termination.

Paper [1], proposes a indirect ZMP controller for biped robot systems. The ZMP information consists of the position and acceleration of the COG, then the ZMP can be indirectly controlled by the COG. The indirect controller is proposed to derive the desired motion of COG from the reference ZMP and the COG error.

1.6 Biped Robots Today

There is a wide range of different robots all over the world today. They can be seen anywhere from in the home to the workplace. People are now accustom to working along side robots and depending on them for everyday things.

Building robots to resemble the human structure and that are modeled around their actions will provide a better understanding of humans as a whole. If human structure can be mimicked in robotics, those parts can then be used to replace human's parts which are too injured to heal. This is where robotic prosthetic limbs can be researched and developed. Some examples of prosthetic limbs are: powered legs, ankle-foot orthosis and forearm prothesis.

Robots are being researched and developed to assist and perform tasks including helping the sick and elderly or even dangerous jobs. Ordinary jobs like receptionist or automotive worker are also great areas to implement robotic counterparts. As long as they are equipped with the right tools, they can perform tasks from operating equipment to driving vehicles just like humans.

Robotics can also be seen today in the form of entertainment. One example is at Universal Studious, a robot named Ursula is built to resemble a female that can sing, dance, play music and speak. Several Disney attractions employ animated attractions for humans to be entertained and amused by.

The increase in worldwide robotic research led to the 2013 DARPA robotics challenge, where the top 16 humanoid robots competed for a 2 million US dollars cash prize. The wining team, SCHAFT, was able to develop a humanoid robot that could drive a vehicle, travel over uneven

terrain, climb a ladder, clear debris, close and open a door, cut a whole in a wall and operate a valve and hose.

Figure 1.1: Winner of the 2013 DARPA Robotics Challenge, SCHAFT

1.6.1 ASIMO

One popular humanoid robot is ASIMO (Advanced Step in Innovative Mobility) by Honda. This was the first humanoid robot to walk on its own. ASIMO was developed to be a multi-functional mobile assistant when it was first introduced back in October 2000. Today you can see ASIMO at many different electronic shows. Disneyland even has a fifteen minute show attributed to it.

Development of humanoid robotics began in the 1980s at Honda. The goal of the company was to create a robot that could adapt to human situations but also improve the way of life. The first line of biped robots was the E0 series between 1986 and 1993. Then the development of the P series was introduced in 1993 and continued until 1997. The P series was the first self-regulating, humanoid robot with wireless communications.

The research in the P and E series lead to the development of ASIMO. Unlike P and E series, ASIMO was the first to incorporate predicted movement control, for better joint flexibility and a more smoother and human-like walking motion. ASIMO can perform tasks like running up to 9km/hour and recognize objects, postures, gestures, its surrounding environment, sounds and faces. All of these features allow ASIMO to interact with humans, while communicating occurs in both English and Japanese.

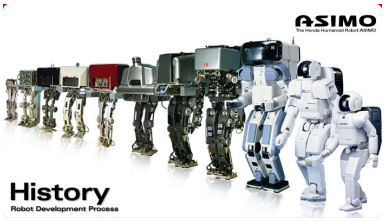

Figure 1.2: ASIMO

1.6.2 Google

After purchasing 7 new diverse robotics companies in one year, Google has begun to monopolize the industry, leading many to believe they will be the next great robotic giant. Google's research in robotics varies from bipeds to four legged robots and self driving cars. With research in such diverse areas, possibilities of use are endless.

One example, a four legged robot called BigDog, is a quadrupedal robot designed to carry luggage for soldiers in places where terrain is too difficult for vehicles. As seen in figure 1.3.

Figure 1.3: BigDog quadruped robot

1.6.3 Rethink Robotics

Rethink Robotics is a company that develops robotic machines that are used to assist humans in the workplace. The flagship product of the company is Baxter, the worlds first low

cost industrial humanoid robot. Baxter increases worker productivity by doing tasks that never have been automated before. This will improve the manufactures ability to compete with other low-cost labor and reducing the need to send work offshore. This is one example of how robots and humans can work together. You can see a picture of Baxter below in figure 1.4.

Figure 1.4: Baxter, the world first low-cost industrial humanoid robot

1.7 Goal of Thesis

The goal of this thesis is to perform the balance control of a five DOF robot leg. The use of information obtained from class work and reading notable literature was used in the design and building of a single leg robot with balance control. The thesis shows the simulations used to find the COM, ZMP, Newton Euler Recursion and path trajectory. All important things to consider when trying to understand the theories behind biped locomotion. An experiment is performed on a five DOF robot leg. Several different tests were conducted using three different controllers, P, PD and PID. Feedback from the force moment sensors mounted on the feet and the joint encoders will be used to achieve this balance.

1.8 Organization of Thesis

The thesis is organized into 8 chapters

Chapter 1 introduces the topics that are associated with biped walking robots and the notable

robots that have been designed.

Chapter 2 gives the reader brief theoretical background about terms and theories that will be used throughout this thesis.

Chapter 3 discusses the robots kinematics and dynamics, regarding the shape and how the modeling of the robot is determined. Determining the DH parameters for the structure of the single leg structure.

Chapter 4 provides the details about the mechanical and electrical design on Prototype Two.

Chapter 5 explains the control system for the robot. This includes the control for the balance of the leg.

Chapter 6 shows the simulations that were done in Simulink and an explanation of their results.

Chapter 7 examines the experimental results from the balancing robot prototype.

Chapter 8 summarizes the thesis and future work that can be done to improve the robot.

Chapter 2

Theoretical Background of a Biped Robot

2.1 Joint Kinematics

The links that compose a mechanism, in this case a robot manipulator, are assumed to be perfectly rigid bodies having a surface that's geometrically perfect in both position and shape. These rigid bodies are then connected together at joints that are in ideal contact with each other without any clearance between them. These two surfaces geometric shape will then determine the freedom of motion between these two links, or the joint kinematics.

A kinematic joint is a connection between two bodies that constrain their relative motion. Two bodies that are in contact with each other create a simple kinematic joint. There are two different types of kinematic joints, lower pair joints and higher pair joints.

2.1.1 Lower Pair Joints

Lower pair joints are more favorable because wear will be spread over the whole surface and lubrication is trapped in the small clearance space. There are 6 possible forms of lower pair joints: revolute, prismatic, helical, cylindrical, spherical and planar joints. For this paper only revolute joints will be used and demonstrated.

2.1.2 Revolute Joints

Just as the name implies, a revolute joint is two links that are connected together and moves in revolution. One of the links will stay stationary as the other one can move, so the angle between them will be changing, as seen from figure 2.1. Theta (θ) is the changing variable that represents the degree in which the link has changed. Thus a revolute joint has only one degree of freedom (DOF). Equation (2.1) represents the joint rotation matrix of a revolute joint [21].

$$ {}^{j}_{i}R = \begin{pmatrix} cos\theta_i & -sin\theta_i & 0 \\ sin\theta_i & cos\theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.1} $$

Where ${}^{j}_{i}R$ is the transformation from the Plucker coordinate system i to Plucker coordinate system j for spacial velocities.

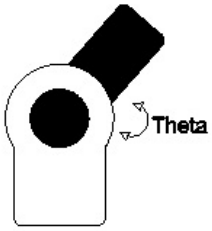

Figure 2.1: Revolute joint

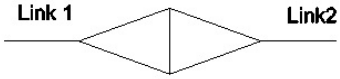

Figure 2.2: Revolute symbol

2.2 Description of an Orientation

As you can see from figure 2.3 the body frame $\{B\}$ is rigidly attached to the object. Frame $\{B\}$ can be described as the orientation of frame $\{B\}$ with respect to $\{A\}$.

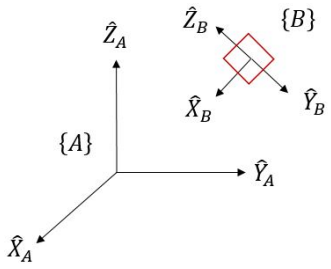

Figure 2.3: Body frame B with fixed frame A

The orientation of $\{B\}$ with respect to $\{A\}$ is described by the three vectors:

$$ {}^{A}_{B}\hat{X} = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} = \begin{bmatrix} \hat{X}_B \hat{X}_A \\ \hat{X}_B \hat{Y}_A \\ \hat{X}_B \hat{Z}_A \end{bmatrix} \tag{2.2} $$

$$ {}^{A}_{B}\hat{Y} = \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix} = \begin{bmatrix} \hat{Y}_B \hat{X}_A \\ \hat{Y}_B \hat{Y}_A \\ \hat{Y}_B \hat{Z}_A \end{bmatrix} \tag{2.3} $$

$$ {}^{A}_{B}\hat{Z} = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} = \begin{bmatrix} \hat{Z}_B \hat{X}_A \\ \hat{Z}_B \hat{Y}_A \\ \hat{Z}_B \hat{Z}_A \end{bmatrix} \tag{2.4} $$

The components of each vector are the projection of each vector on the unit direction of the frame $\{A\}$. The rotation matrix of frame $\{B\}$ with respect to frame $\{A\}$ is:

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}_{B}\hat{X} & {}^{A}_{B}\hat{Y} & {}^{A}_{B}\hat{Z} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \hat{X}_B \hat{X}_A & \hat{Y}_B \hat{X}_A & \hat{Z}_B \hat{X}_A \\ \hat{X}_B \hat{Y}_A & \hat{Y}_B \hat{Y}_A & \hat{Z}_B \hat{Y}_A \\ \hat{X}_B \hat{Z}_A & \hat{Y}_B \hat{Z}_A & \hat{Z}_B \hat{Z}_A \end{bmatrix} \tag{2.5} $$

Equation (2.5) is the projection of frame $\{B\}$ onto $\{A\}$ in each of the X in (2.2), Y in (2.3) and Z in (2.4) axis. The rotation matrix contains nine elements, while only three parameters are required to define the orientation of a body in space.

${}^{A}_{B}R$ is the rotation matrix that transforms a vector expressed in coordinate frame $\{B\}$ to a vector expressed in coordinate frame $\{A\}$. It provides a representation of the orientation of

frame $\{B\}$ relative to frame $\{A\}$ and thus, can be a representation of rotation from frame $\{B\}$ to frame $\{A\}$.

2.3 Rotation Matrices

The following three figures are basic rotations by an angle θ about the X, Y and Z axis, in three dimensions, using the right hand rule.

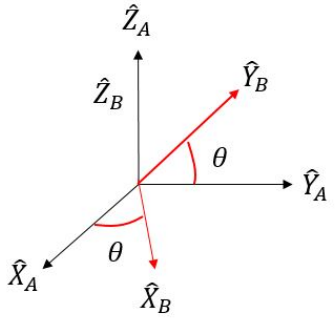

Figure 2.4: Rotation about Z-axis

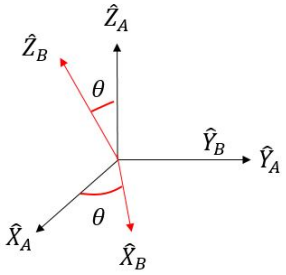

Figure 2.5: Rotation about Y-axis

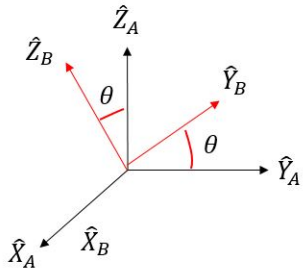

Figure 2.6: Rotation about X-axis

The following equations (2.6), (2.7) and (2.8) are the resulting rotation matrices for the three basic rotations.

Rotated about Z-axis:

$$ {}^{A}_{B}R_Z = \begin{vmatrix} cos\theta & sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{2.6} $$

Rotated about Y-axis:

$$ {}^{A}_{B}R_Y = \begin{vmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{vmatrix} \tag{2.7} $$

Rotated about X-axis:

$$ {}^{A}_{B}R_X = \begin{vmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{vmatrix} \tag{2.8} $$

2.4 Center of Mass

Every system at every instant in time, has a unique location in space that is the average position of the system's mass. This place is called the center of mass, which is referred to as COM. As described in physics, COM is the distribution of mass in space and the unique point where the weighted relative position, of the distributed mass, sums up to zero. The distribution of the mass is balanced around this center point.

The COM is a useful reference point that involve calculations in mechanics with mass distributed in space, such as rigid body dynamics. The frame of the center of mass is an inertial frame in which the COM of the system is at rest with respect to the origin of the coordinate system.

The COM of a system at the position $\vec{r}_{cm}$ is defined by

$$\begin{aligned} \vec{r}_{cm} &= \frac{\sum \vec{r}_i m_i}{m_{tot}} \quad \text{for discrete systems} \\ &= \frac{\int \vec{r} dm}{m_{tot}} \quad \text{for continuous systems} \end{aligned} \tag{2.9}$$

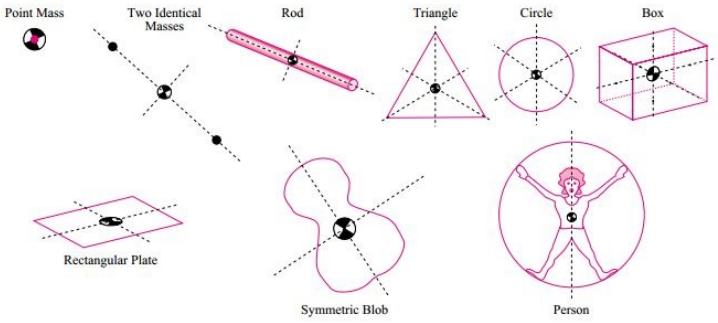

Figure 2.7: Different shapes and objects with varying COM

As you can see from the above figure 2.7, all types of shapes and objects have a varying COM. The example that is closest to a biped robot would be the person. When designing a humanoid robot, the COM should be around the same spot as a human being.

2.5 Denavit-Hartenberg Theory

One commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, or D-H, convention. The D-H convention employs each homogeneous transformation (A_i) represented as a product of four basic transformations. Consider frame $\{i-1\}$, $\{x_{i-1}, y_{i-1}, z_{i-1}\}$ and frame $\{i\}$, $\{x_i, y_i, z_i\}$. If x_i is perpendicular to z_{i-1} and x_i intersects z_{i-1} then there exists unique parameters θ_i, a_i, d_i and α_i such that the homogeneous transformation from $\{i-1\}$ to $\{i\}$ is described by $A = R_Z(\theta_i)T_Z(d_i)T_X(a_i)R_X(\alpha_i)$. As seen in figure 2.8.

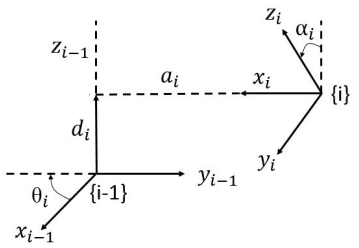

Figure 2.8: D-H from frame $\{i\}$ to frame $\{i-1\}$

If x_i is perpendicular to z_{i-1} and x_i intersects z_{i-1} and D-H convention is satisfied then

$$
\begin{aligned}
A_i &= R_Z(\theta_i)T_Z(d_i)T_X(a_i)R_X(\alpha_i) \\
&= \begin{vmatrix} cos\theta_i & -sin\theta_i & 0 & 0 \\ sin\theta_i & cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha_i & -sin\alpha_i & 0 \\ 0 & sin\alpha_i & cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
&= \begin{vmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix}
\end{aligned}
\tag{2.10}
$$

A_i is the homogeneous transformation from frame $\{i-1\}$ to frame i. The four parameters θ_i, a_i, d_i and α_i are associated with link i and joint i. The four parameters are generally given the following names: angle, length, twist and offset respectively. For a revolute joint θ_i is the variable and a_i, d_i and α_i are the constants. For a prismatic joint d_i is the variable and a_i, θ_i

and α_i are the constants.

link	a_i	d_i	α_i	θ_i	
1	a_1	d_1	α_1	θ_1	$\leftarrow A_1$
2	a_2	d_2	α_2	θ_2	$\leftarrow A_2$
3	a_3	d_3	α_3	θ_3	$\leftarrow A_3$

Table 2.1: D-H table

The following nine steps are used to draw and fill out the D-H table as seen from Table 2.1.
Step 1: Locate and label the joint axes z_0, ..., z_{n-1}

Step 2: Establish the base frame. Set the origin on the z_0 axis. The x_0 and y_0 axes are chosen conveniently to form a right-hand frame. For $i = 1$, ... $n - 1$, perform steps 3 to 5.

Step 3: Locate the origin o_i where the common normal to z_i and z_{i-1} intersects z_i. If z_i intersects z_{i-1} locate o_i at this intersection. If z_i and z_{i-1} are parallel, locate o_i at joint i.

Step 4: Establish x_i along the common normal between z_{i-1} and z_i through o_i or in the direction normal to the $z_{i-1} - z_1$ plane if z_{i-1} and z_i intersect.

Step 5: Establish y_i to complete a right-hand frame.

Step 6: Establish the end effector frame $o_n x_n y_n z_n$.

Step 7: Create a table of link parameters like table 2.1

a_i= distance along x_i from o_i to the intersection of the x_i and z_{i-1} axes.

d_i= distance along z_{i-1} from o_{i-1} to the intersection of the x_i and $z_i - 1$ axes. d_i is a variable if joint i is prismatic.

α_i= the angle between z_{i-1} and z_i measured about x_i.

θ_i= the angle between x_{i-1} and x_i measured about z_{i-1}. θ_i is variable if joint i is revolute.

Step 8: Form the homogeneous transformation matrices A_i by substituting the above parameters into equation (2.10)

Step 9: Form ${}^{i-1}_{n}T = A_i...A_n$. This then gives the position and orientation of the tool frame expressed in base coordinates.

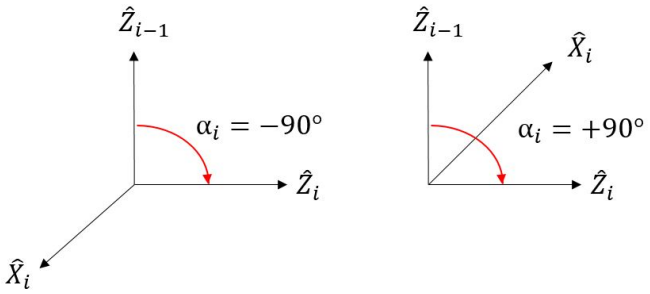

Figure 2.9: If Z_{i-1} and Z_i intersect

2.6 Newton-Euler Formulation

The Newton-Euler formulation is done by treating each link separately, and writing down the equations describing its linear motion and angular motion. Each link is coupled to other links so each equation will contain coupling forces and torques that appear in the neighboring equations. By doing a so-called forward-backward recursion, we are able to determine all of these coupling terms and eventually arrive at a description of the manipulator as a whole.

There are two different aspects between which we must distinguish. First, interest in the closed-form equations that describe the time evolution of the generalized coordinates. Second, knowing what generalized forces and torques need to be applied in order to realize a particular time evolution of the generalized coordinates. In the second aspect we only want to know what time dependent function τ (torque) produces a particular trajectory q.

The three facts of the Newtonian mechanics are as follows:

1. Every action has an equal and opposite reaction. Thus if body one applies a force f and

torque τ to body two, then body two applies a force of -f and a torque of $-\tau$ to body one.

2. The rate of change of the linear momentum equals the total force applied to the body.

3. The rate of change of the angular momentum equals the total torque applied to the body.

Now we can derive the Newton-Euler formulation of the equations of motion for an n-link manipulator. Choose frames 0,...,n, where frame 0 is an inertial frame, and frame i is rigidly attached to link i. The first set of vectors pertain to the velocities and accelerations of various parts of a manipulator.

$$\vec{a}_{c,i} = \text{the acceleration of the center of mass of link } i. \tag{2.11}$$

$$\vec{a}_{e,i} = \text{the acceleration of the end of link } i. \tag{2.12}$$

$$\vec{\omega}_i = \text{the angular velocity of frame } i \text{ with respect to frame 0.} \tag{2.13}$$

$$\vec{\alpha}_i = \text{the angular acceleration of frame } i \text{ with repect to frame 0.} \tag{2.14}$$

The next several vectors pertain to forces and torques:

$$\vec{g}_i = \text{the acceleration due to gravity (expressed in frame } i). \tag{2.15}$$

$$\vec{f}_i = \text{the force exherted by link } i\text{-1 on link } i. \tag{2.16}$$

$$\vec{\tau}_i = \text{the torque exherted by link } i\text{-1 on link } i. \tag{2.17}$$

$$\prescript{i+1}{i}{R} = \text{the rotation matrix from frame } i{+}1 \text{ to frame } i. \tag{2.18}$$

The final set of vectors pertain to the physical features of the manipulator. Note that each of

the following vectors is constant as a function of q. So no matter what the configuration of the manipulator is, these will stay the same.

$$m_i = \text{the mass of link } i. \tag{2.19}$$

$$I_i = \text{the inertia matrix of link } i. \tag{2.20}$$

$$\vec{r}_{i,ci} = \text{the vector from joint } i \text{ to the center of mass of link } i. \tag{2.21}$$

$$\vec{r}_{i+1,ci} = \text{the vector from joint } i{+}1 \text{ to the center of mass of link } i. \tag{2.22}$$

$$\vec{r}_{i,i+1} = \text{the vector from joint } i \text{ to joint } i{+}1. \tag{2.23}$$

Now consider the free body diagram from figure 2.10. This shows the link i together with all the forces and torques acting on it. f_i is the force applied by link $i-1$ to link i. Then from the law of action and reaction, link $i+1$ applies a force of $-f_{i+1}$ to link i but this vector needs to be expressed in frame i. So it gets multiplied by the rotation matrix ${}^{i+1}_{i}R$. A similar explanation is formed with the implementation of torques. The force $m_i g_i$ is the gravitational force.

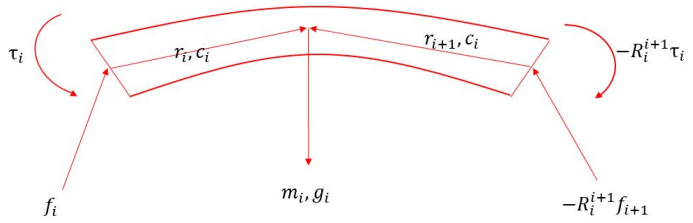

Figure 2.10: Moments and forces on link i

The heart of the Newton-Euler formulation consists of finding the vectors f_1, ..., f_n and τ_1, ..., $\tau_n$corresponding to a given set of vectors q, $\dot{q}$, $\ddot{q}$. So we will be finding the forces and toques in the manipulator that correspond to a given set of generalized coordinates and their first two derivatives.

$$f_i = {}_i^{i+1}Rf_{i+1} + m_i a_{c,i} - m_i g_i \tag{2.24}$$

$$\tau_i = {}_i^{i+1}R\tau_{i+1} - f_i \times r_{i,ci} + ({}_i^{i+1}Rf_{i+1}) \times r_{i+1,ci} + \alpha_i + \omega_i \times (I_i\omega_i) \tag{2.25}$$

So you can either use equations (2.24) and (2.25) corresponding to a particular trajectory $q(.)$, or obtain the closed-form dynamical equations. The idea is as follows: Given q, $\dot{q}$, $\ddot{q}$, suppose we are able to determine all the velocities and accelerations of the various parts of the manipulator, that includes $a_{c,i}$, ω_i and α_i. Then we can solve (2.24) and (2.25) recursively to find all the forces and torques. First, set $f_{n+1} = 0$ and $\tau_{n+1} = 0$. This will express that there is no $n+1$ link and we can solve (2.24). By successively substituting in $i = n, n-1, ..., 1$ we will find all the forces. Similarly we can solve for (2.25). By successively substituting $i = n-1, ..., 1$ we find all the torques. Note that the above iterations is in the direction of decreasing i.

The solution is complete once we find an easily computed relation between q, $\dot{q}$, $\ddot{q}$ and $a_{c,i}$, ω_i and α. This can be done through recursive computation with increasing i. This will be shown in 2.6.1

2.6.1 Forward Recursion (Kinematics)

Forward recursion is the process of determining the acceleration of each link as well as the acceleration of the end of each link. The following will explain the equations to use and the steps to implement these equations.

Angular velocity of frame i with respect to frame 0:

$$\omega_i = [{}_i^{i-1}R]^T\omega_{i-1} + \hat{b}_i\dot{q}_i \tag{2.26}$$

This equation (2.26) expresses the fact that the angular velocity of frame i equals that of

frame $i-1$ plus the added rotation from joint i.

Where

$$\hat{b}_i = \ {}^{1}_{0}R]^T \hat{Z}_{i-1} \tag{2.27}$$

is the axis of rotation of joint i in frame i.

Angular acceleration of frame i with respect to frame 0:

$$\alpha_i = \ {}^{i-1}_{i}R]^T \alpha_{i-1} + \hat{b}_i \ddot{q}_i + \omega_i \times \hat{b}_i \dot{q}_i \tag{2.28}$$

α_i is the derivative of the angular velocity in frame i.

Acceleration of the end of link i:

$$a_{e,i} = \ {}^{i-1}_{i}R]^T a_{e,i-1} + \dot{\omega}_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1}) \tag{2.29}$$

Acceleration of the center of mass of link i:

$$a_{c,i} = \ {}^{i-1}_{i}R]^T a_{e,i-1} + \dot{\omega}_i \times r_{i,ci} + \omega_i \times (\omega_i \times r_{i,ci}) \tag{2.30}$$

Follow the following steps to solve the forward recursion problem.

Start with the initial conditions $\omega_0 = 0$, $\alpha_0 = 0$, $a_{c,0}$ and $a_{e,0}$. Then solve equations (2.26), (2.28), (2.29) and (2.30) to find ω_i, α_i, $a_{e,i}$ and $a_{c,i}$ for i increasing from 1 to n.

2.6.2 Backward Recursion (Dynamics)

Backward recursion is the process of determining the torque and force of each link. The following will explain the equations to use and the steps to implement these equations.

Refer to equations (2.24) and (2.25).

Follow the following steps to solve the forward recursion problem.

Start with the initial conditions $f_{n+1} = 0$ and $\tau_{n+1} = 0$. Then solve equations (2.24) and (2.25) to compute f_i and τ_i for decreasing i from n to 1.

2.7 The Gait Cycles

The gait cycle or otherwise known as the walking cycle is broken down into two sections, single support and double support phases. The gait cycle extends from heel strike to heel strike

of one leg and includes the stance and swing phases of both legs.

In the basic gait cycle, movements are divided into times as to when the foot is on the ground and when its not. The stance phase, when the foot is on the ground, makes up most of the gait cycle. When the foot is off the ground its called the swing phase.

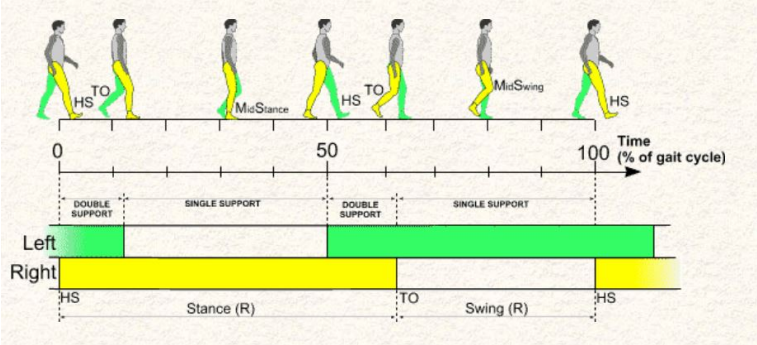

Figure 2.11: Gait cycle

The stance phase of a gait can be described as the initial point of contact between the foot and the surface, while the entire foot makes contact and until just the toes are leaving the surface.

The gait cycle can be broken down into three points:

- Contact period, when the heel strikes the ground.
- The mid-stance period, when the foot is flat on the ground.
- The propulsive period when the heel is lifted off the ground.

2.7.1 Single Support Phase (SSP)

Single support phase is just as it sounds, the biped will be supported with just one leg. The support surface is what's used to keep the biped upright and balanced. The center of mass must be within this support phase. So if there is just one foot on the ground then the support phase is restricted to the size of the foot.

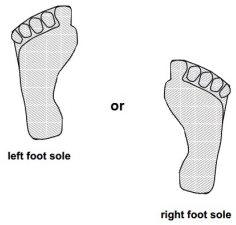

Figure 2.12: Single support

As you can see from figure 2.12 the support is restricted to the size of a human foot. During the constructing of a biped, the support area or foot can designed any which way that will satisfy the user. The bigger the better, since it will create a larger support for the center of mass to try and stay in.

2.7.2 Double Support Phase (DSP)

The double support phase is a little different than the single support phase, because it is not just the area under the two feet. If you look at figure 2.13 you can see it is both feet plus the area between them. So it creates a larger area for support and the center of mass to fall in.

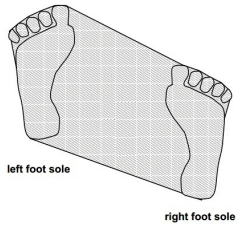

Figure 2.13: Double support phase

The easiest part of controlling a biped is having it stand straight up since the support region is much larger than in the SSP. While walking the DSP only occurs for a limited time during this time if the speed of the walk increases, the DSP time will decreases, eventually turning into running.

2.8 Zero Moment Point (ZMP)

The term ZMP was first coined by *Vukobratović* and *Stepanenko* in 1972 [29]. They said, *"In figure 2.14 an example of force distribution across the foot is given. As the load has the same sign all over the surface, it can be reduced to the resultant force R(f_p), the point of attack f which*

will be in the boundaries of the foot. Let the point on the surface of the foot, where the resultant R passed, be denoted as the zero-moment point"[29].

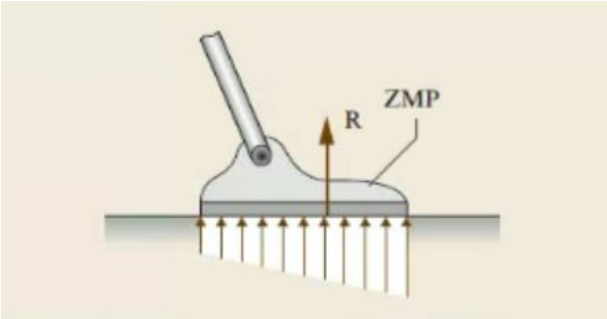

Figure 2.14: Original definition of the ZMP

To clarify the above statement consider a rigid foot with a flat sole that has complete contact with the surface, as shown in figure 2.15. The influence of the biped is replaced by the force, $\vec{F}_a$ and the moment, $\vec{M}_a$, acting on the point A. The gravitational acceleration, $\vec{g}$, is acting in the negative z direction. To keep the whole biped balanced, the reaction force, $\vec{F}_p = (F_{px}, F_{py}.F_{pz})$ and moment, $\vec{M}_p = (M_{px}, M_{py}, M_{pz})$ are acting on point, P. The horizontal reaction force (F_{px}, F_{py}) will be the friction force that is compensating for the horizontal components of force $\vec{F}_a$. The vertical component of the reaction moment, M_{pz}, is balancing the vertical component of $\vec{M}_a$ and the moment induced by the force $\vec{F}_a$. Assuming that there will be no slip the static friction will be represented by F_{px}, F_{py} and M_{pz}. To compensate for the horizontal components of $\vec{M}_a$ (M_{ax}, M_{ay}), the point P will have to be shifted in such a way that F_{pz} is fully compensating for them. This will then imply that the horizontal components of $\vec{M}_p$ are reduced to zero.
Hence

$$M_{px} = M_{py} = 0 \tag{2.31}$$

The above equality led to the natural choice for the term ZMP. If the support polygon isn't large enough to include point P, then the force F_p will act on the foot's edge and the uncompensated part of M_a and F_a will then result in a rotation about the edge. This could lead to the biped falling over if it's not within the support polygon. Now, the static equilibrium equations can be stated with the forces and moments around the origin.

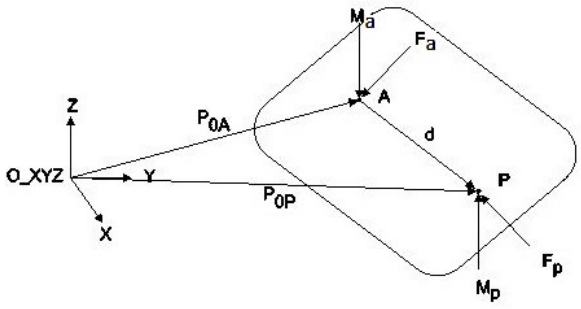

Figure 2.15: Forces and moments acting on a rigid foot with a flat sole; fully supported with the foot.

Sum of forces:

$$\vec{F_p} + \vec{F_a} = 0 \tag{2.32}$$

Sum of moments:

$$\vec{p}_{OP} \times \vec{F_p} + \vec{M_a} + M_{pz} + \vec{p_{OA}} \times \vec{F_a} = 0 \tag{2.33}$$

The terms p_{OA} and p_{OP} are the vectors from the base frame origin O_{XYZ} to the points A and P. If you place the base-frame-origin on the XY-plane then the sum of moments will result in:

$$(\vec{p}_{OP} \times \vec{F_p})_{XY} + (M_a)_{XY} + (\vec{p_{OA}} \times \vec{F_a})_{XY} = 0 \tag{2.34}$$

With equation (2.34) the position of the zmp can be computed but it doesn't answer the question of whether or not the biped is in dynamic equilibrium. The biped will be in dynamic equilibrium if point P (ZMP) is within the support polygon. This will then lead to the criterion that, to achieve a stable gait the ZMP should be within the support polygon at every single instant.

2.8.1 General ZMP

How the general ZMP works:

1. First, sensors in the feet of the standing robot report the weight distribution to the processor.

2. The algorithm will then use these reported weights and the positions of these sensor to calculate the torque acting upon the robots feet.

3. The gyroscopes and accelerometers send data about the upper body to the processor.
4. The algorithm then combines this data to determine the placement of the next step.
5. The processor actuates the motors in the hips, knees and feet to move up and allow the biped to move forward.
6. As the lifted foot comes in contact with the ground the sensors report the weight and gyroscopes report the balance again.
7. The biped will adjust the motors until it reaches a ZMP.

2.8.2 ZMP Derivation

The following will be just one of the many of different ways to derive the ZMP equations using forward kinematics. To calculate the point, P, there are several assumptions that need to be made:

1. The biped robot consists of n rigid links.
2. All kinematic information, such as position of COM, link orientation, velocities, etc... are all known and calculated by forward kinematics.
3. The floor is rigid and motionless.
4. The feet cannot slide over the floor's surface.
5. All joints are actively actuated.

The first thing to do is calculate the total mass m_{tot}:

$$m_{tot} = \sum_{i=1}^{n} m_i \tag{2.35}$$

You must also calculate the distance from the base-frame-origin to its COM, as seen in the figure 2.16.

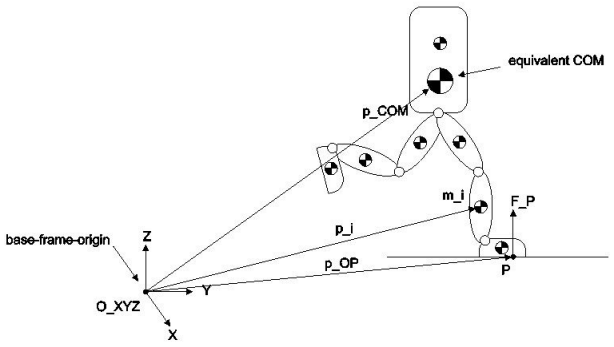

Figure 2.16: Schematic biped model with point P.

With this, we can then find the total linear momentum, $\vec{P}$, and total angular momentum, $\vec{H}$, with respect to the base-frame-origin.

$$\vec{P} = \sum_{i=1}^{n} m_i \dot{\vec{p}}_i \tag{2.36}$$

$$\vec{H} = \sum_{i=1}^{n} (\vec{p}_i \times m_i \dot{\vec{p}}_i + \vec{I}_i \vec{\omega}_i) \tag{2.37}$$

As seen from equation (2.37), $\vec{I}_i$ and $\vec{\omega}_i$ are respectively the inertia tensor and the angular velocities of the i-th link with respect to the base-frame-origin.

The equation for $\vec{I}_i$ is as follows:

$$\vec{I}_i = R_i I_i R_i^T \tag{2.38}$$

R_i is the rotation matrix of the i-th link with respect to the base-frame-origin. I_i is the inertia matrix of the i-th link with respect to the link-frame-origin attached to their links. The time derivative of $\vec{P}$ and $\vec{H}$ is the rate of change of the linear and angular momentums.

$$\dot{\vec{P}} = \sum_{i=1}^{n} m_i \ddot{\vec{p}}_i \tag{2.39}$$

$$\dot{\vec{H}} = \sum_{i=1}^{n} (\dot{\vec{p}}_i \times (m_i \dot{\vec{p}}_i) + \vec{p}_i \times (m_i \ddot{\vec{p}}_i) + \vec{I}_i \dot{\vec{\omega}}_i + \vec{\omega}_i \times (\vec{I}_i \vec{\omega}_i)) \tag{2.40}$$

Note that $\dot{\vec{p}}_i \times (m_i \dot{\vec{p}}_i) = 0$ because $\dot{\vec{p}}_i$ and $m_i \dot{\vec{p}}_i$ are parallel. With this information the following will hold:

$$\vec{F}_p = -\vec{F}_a = \dot{\vec{P}} - m_{tot}\vec{g} \tag{2.41}$$

$$\vec{M}_o = \dot{\vec{H}} - \vec{p} \times m_{tot}\vec{g} \tag{2.42}$$

$\vec{F}_p$ and $\vec{M}_o$ are the external force and moment that describes how the floor is reacting to the biped with respect to the base frame origin. $\vec{F}_a$ is the force that the biped is acting upon the floor. The external force $\vec{F}_p$ is acting on point P, so the moment will be:

$$\vec{M}_O = \vec{p}_{OP} \times \vec{F}_p + \vec{M}_p \tag{2.43}$$

$\vec{M}_p$ will be the moment at P and $\vec{p}_{OP}$ is the vector from the base-frame-origin to point P. Since $\vec{M}_P$ is on the point P, being the ZMP, it is $\vec{M}_P = [0\,0\,M_Z]^T$. Now we can let (2.42) equal (2.43) and solve for M_P.

$$\vec{M}_p = \dot{\vec{H}} - \vec{p}_{COM} \times m_{tot}\vec{g} + (\dot{\vec{P}} - m_{tot}\vec{g}) \times \vec{p}_{OP} \tag{2.44}$$

Then from this, we can find the distance from the base-frame-origin to the location of the zmp. $\vec{p}_{ZMP} = \vec{p}_{OP} = [x_{ZMP}, y_{ZMP}, z_{ZMP}]^T$

$$x_{ZMP} = \frac{m_{tot} g_z P_{COMx} + z_{ZMP} \dot{P}_x - \dot{H}_y}{m_{tot} g_z + \dot{P}_z} \tag{2.45}$$

$$y_{ZMP} = \frac{m_{tot} g_z P_{COMy} + z_{ZMP} \dot{P}_y + \dot{H}_x}{m_{tot} g_z + \dot{P}_z} \tag{2.46}$$

It should be noted that z_{ZMP} represents the height of the floor, so when the XY-plane is placed on the floor z_{ZMP} will be zero. Where g_z is the z component of the gravity vector. Also $\dot{H}_y$ and $\dot{H}_x$ are the x and y components of $\dot{\vec{H}}$. The same can be said for P_{COMy} and P_{COMx}.

Looking at the equations from Huang [10] and assuming that $Z_{ZMP} = 0$, the following equations can derive the position of the ZMP:

$$x_{ZMP} = \frac{\sum_{i=1}^{n} (m_i (p_{ix} (\ddot{P}_{iz} + g_z) - p_{iz} (\ddot{P}_{ix} + g_x)) - I_{iy} \omega_{iy}}{\sum_{i=1}^{n} m_i (\ddot{P}_{iz} + g_z)} \tag{2.47}$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} (m_i (p_{iy} (\ddot{P}_{iz} + g_z) - p_{iz} (\ddot{P}_{iy} + g_y)) - I_{ix} \omega_{ix}}{\sum_{i=1}^{n} m_i (\ddot{P}_{iz} + g_z)} \tag{2.48}$$

The only difference between equations (2.45), (2.46) and (2.47), (2.48) is that in Huang's equations have the rate of linear and angular momentums expanded into components.

Chapter 3

Robot Kinematics and Dynamics

3.1 Kinematics

Robot kinematics is the study of the movement of multi-degree of freedom kinematic chains that form a structure of robotic systems. The links of the robot are modeled as rigid bodies and the joints are assumed to provide pure rotational translations [26]. The kinematics of a robot manipulator involves the DH parameters, as discussed in section 2.5 will help determine the forward and reverse kinematics. Stated more differently, the forward kinematics problem gives you the position and the orientation of the end effector, if the values of the joint variables are given. The joint variables are the angle to which each joint is positioned and the lengths of each link.

The following is the D-H parameters for the right foot standing flat on the floor. In order to identify which variable corresponds to which joint, the four are indicated in both the chart and on the figure.

link	a_i	d_i	α_i	θ_i	
0	a_1	0	90	90	$\leftarrow A_0$
1	a_2	0	-90	$\theta_1 + 90$	$\leftarrow A_1$
2	a_3	0	0	θ_2	$\leftarrow A_2$
3	a_4	0	0	θ_3	$\leftarrow A_3$
4	a_5	0	90	θ_4	$\leftarrow A_4$
5	0	0	-90	$\theta_5 + 90$	$\leftarrow A_5$

Table 3.1: D-H table for simulation model of the right foot

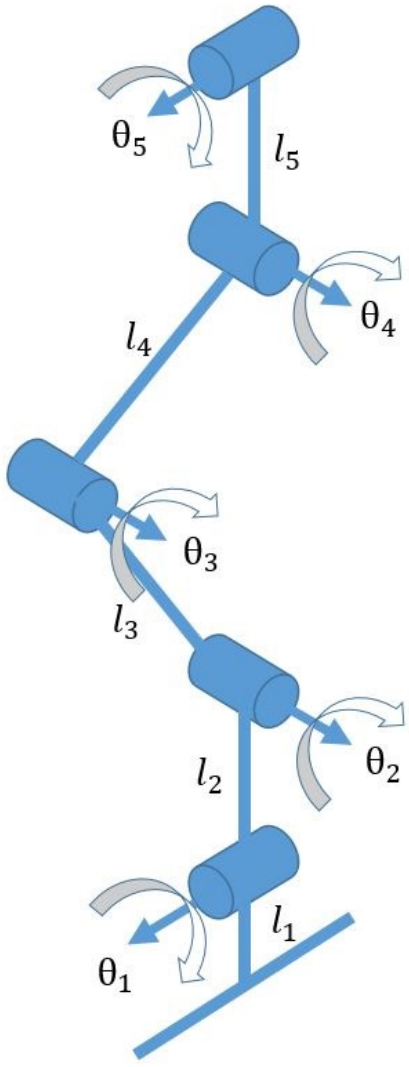

Figure 3.1: Model used to determine the DH parameters of the right foot.

The next step is to determine the transformation matrices, A_1 through to A_5. The results are shown below.

$$A_1 = \begin{vmatrix} cos(\theta_1 + \frac{\pi}{2}) & 0 & -sin(\theta_1 + \frac{\pi}{2}) & a_2 cos(\theta_1 + \frac{\pi}{2}) \\ sin(\theta_1 + \frac{\pi}{2}) & 0 & cos(\theta_1 + \frac{\pi}{2}) & a_2 sin(\theta_1 + \frac{\pi}{2}) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.1}$$

$$A_2 = \begin{vmatrix} cos(\theta_2) & -sin(\theta_2) & 0 & a_3 cos(\theta_2) \\ sin(\theta_2) & cos(\theta_2) & 0 & a_3 sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.2}$$

$$A_3 = \begin{vmatrix} cos(\theta_3) & -sin(\theta_3) & 0 & a_4 cos(\theta_3) \\ sin(\theta_3) & cos(\theta_3) & 0 & a_4 sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.3}$$

$$A_4 = \begin{vmatrix} cos(\theta_4) & 0 & sin(\theta_4) & a_5 cos(\theta_4) \\ sin(\theta_4) & 0 & -cos(\theta_4) & a_5 sin(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.4}$$

$$A_5 = \begin{vmatrix} cos(\theta_5 + \frac{\pi}{2}) & 0 & -sin(\theta_5 + \frac{\pi}{2}) & 0 \\ sin(\theta_5 + \frac{\pi}{2}) & 0 & cos(\theta_5 + \frac{\pi}{2}) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.5}$$

The purpose of the transformation matrices is used to determine the position and orientation of each link. So A_1 will be the position and orientation of the first link and its the same procedure for the rest of the links.

3.1.1 Forward Kinematics

The forward kinematics will give you the location of the end effector, given all the joint angles. This involves multiplying the transformation matrices, A_1 through A_5, that were found using the DH parameters. This will yield the transformation matrix and describe where the end effector is with respect to the base frame. So the whole transformation will be from the foot all the way up to the hip.

$$^{0}_{n}T = \ A_1 A_2 ... A_n \tag{3.6}$$

$$^{0}_{2}T = \ A_1 A_2 \tag{3.7}$$

$$^{0}_{3}T = \ ^{0}_{2}T A_3 = A_1 A_2 A_3 \tag{3.8}$$

$$^{0}_{4}T = \ ^{0}_{3}T A_4 = A_1 A_2 A_3 A_4 \tag{3.9}$$

$$^{0}_{5}T = \ ^{0}_{4}T A_5 = A_1 A_2 A_3 A_4 A_5 \tag{3.10}$$

For a generic situation you would multiply out 0 to n like in equation 3.6.

When you expand equation (3.10), you will end up with a resultant 4 by 4 matrix. The top left 3 by 3 matrix is the rotation matrix from the base frame to the top hip joint. The 3 by 1 column vector on the right is the displacement of the hip compared to the foot (base). Its the same for the other three equations (3.7), (3.8) and (3.9). By expanding these, you find the rotation matrices and displacements of each specific joint.

By using the link lengths that are shown in table 3.2, the forward kinematics can be determined.

3.2 Dynamics

Dynamic equations of motion provide relationships between actuation and contact forces acting on robot mechanisms, and the motion trajectories and accelerations that result. Dynamics are an important part for mechanical design, control and execution of simulations. There are two types of dynamics that are going to be used, forward and inverse. Both fall under the

Link	Length (m)
l_1	0.033
l_2	0.064
l_3	0.17
l_4	0.16
l_5	0.071

Table 3.2: Robot link lengths for simulation

Newton-Euler formulation discussed in section 2.6. Inverse dynamic algorithms are used to perform calculations on rigid body models of robot mechanisms. Inverse dynamics determine the joint forces and torques from a specific robot trajectory. The position, velocity and acceleration are all used. Forward dynamics is when the joint forces and torques are specified or known and the joint accelerations need to be determined.

There have been several different methods to address the problem of inverse dynamics, but the two most popular are the Lagrangian and the Newton Euler formulation (section 2.6).

For the simulations shown later on, Newton Euler was used. Referring to section 2.6 will give an insight into the equations that were used to calculate the dynamic results.

3.3 Trajectory Generation

The walking of the robot can only occur with a two legged system. Since the experimental robot is only one leg, the simulation model will have a walking pattern generation within it. The walking pattern is designed around the lengths of the links, distance between the feet, and the stride length and velocity. The paper [6],"A Simple Trajectory Generation Method for Biped Walking" by Feng and Sun was used to help in this design . The paper discusses how the walking pattern for a five degree of freedom robot was designed. The design for the robot is similiar to the one used for simulations, although some parameters had to be changed.

The walking pattern is separated into two phases. There is the standing phase and the swinging or walking phase. Both of these phases will have the same length periods and include all five joints.

3.3.1 Lateral Motion

The standing phase is the phase the leg starts in. Its designed to help balance the system when the opposing leg is in the swinging phase. There are two different coordinate planes the standing phase operates in. The first plane is the lateral motion, which is the motion from side to side. This will affect Joints 1 and 5, the hip and ankle joint. In figure 3.2 you can see Joints 1 and 5 move to the side..

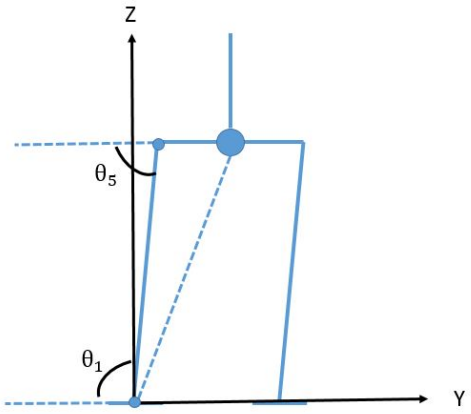

Figure 3.2: Lateral plane of the robot.

The following equations are used to generate the standing phase and how to calculate the joint angles 1 and 5.

$$y(t) = \frac{\lambda}{2}e^{-\sqrt{k}t} + \frac{\lambda}{2}e^{\sqrt{k}t} = \lambda cosh(\sqrt{k}t) \tag{3.11}$$

$$\lambda = \frac{L_s}{cosh(\frac{T}{2}\sqrt{k})} \tag{3.12}$$

$$k = \frac{g}{z_c} \tag{3.13}$$

$$t \in [0, \frac{T}{2}] \tag{3.14}$$

$$\theta_1 = 0.5\pi + \arctan(\frac{y - L_s}{z_c}) \tag{3.15}$$

$$\theta_5 = \pi - \theta_1 \tag{3.16}$$

$y(t)$: hip trajectory

g: gravity

z_c: height of mass center in lateral plane

T: half period

L_s: distance between two feet's center when standing still

3.3.2 Sagittal Motion

The sagittal motion, the second plane the robot will be actauted in, involves the front to back motion. This will include the front to back motion. Seen in figure 3.3, Joints 2, 3 and 4 will be affected. Joint 2 is part of the ankle, joint 3 is the knee and joint 4 is part of the hip.

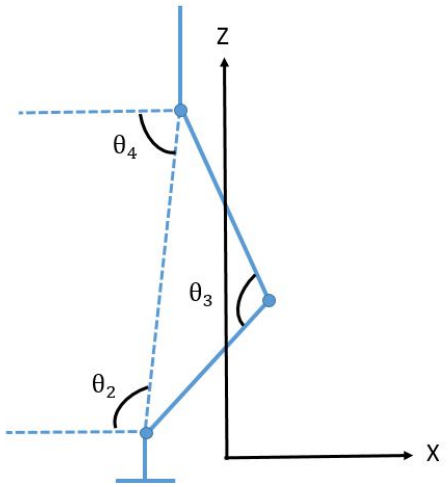

Figure 3.3: Sagittal plane of the robot.

The following equations are used to determine the joint angles and generate the swing trajectories.

$$x(t) = \frac{\gamma}{2}e^{\sqrt{k}t} - \frac{\gamma}{2}e^{-\sqrt{k}t} = \gamma \sinh(\sqrt{k}t) \tag{3.17}$$

$$\gamma = \frac{D_s}{2\sinh(\frac{T}{2}\sqrt{k})} \tag{3.18}$$

$$k = \frac{g}{z_c} \tag{3.19}$$

$$t \in [0, \frac{T}{2}] \tag{3.20}$$

$$\theta_2 = \alpha + \gamma_2 \tag{3.21}$$

$$\theta_3 = \arccos(\frac{l_{shin}^2 + L_{thigh}^2 - l_o^2}{2l_{shin}l_{thigh}}) \tag{3.22}$$

$$\theta_4 = \beta + \gamma_1 \tag{3.23}$$

$$l_o = \sqrt{(x - x_a)^2 + (z_c - z_a)^2} \tag{3.24}$$

$$\alpha = \arccos(\frac{l_{shin}^2 + L_o^2 - l_{thigh}^2}{2l_{shin}l_o}) \tag{3.25}$$

$$\beta = \pi - \theta_3 - \alpha \tag{3.26}$$

$$\gamma_2 = \frac{\pi}{2} + \arctan(\frac{x - x_a}{z_c - z_a}) \tag{3.27}$$

$$\gamma_1 = \pi - \gamma_2 \tag{3.28}$$

$$z_a(t) = \frac{H_s}{2}(1 + \sin(\frac{2\pi}{T}t + \frac{\pi}{2})) \tag{3.29}$$

$$x_a(t) = D_s(\sin(\frac{\pi}{T}t)) \tag{3.30}$$

$$t \in [0, \frac{T}{2}] \tag{3.31}$$

$x(t)$: swing ankle position

D_s: half step length

T: half period

z_c: height of mass center in sagittal plane

l_{shin}: length of shin

l_{thigh}: length of thigh

H_s: max height ofswing foot

Chapter 4

Mechanical and Electrical Design of the Robot Leg

4.1 Mechanical

The mechanical design is very important, since it can affect the way your whole control system works. You want it to be light weight and shaped in a way that the center of mass isn't too close to the ground, but at the same time isn't too high up. This chapter will briefly discuss the construction of the first prototype, then proceed to go onto the second.

The first prototype was designed around the idea of using scrap from old cars, using the motors from automatic car door windows. These motors were discovered after the complete build of the leg proved to be too heavy and large for the electronic system to handle. Figure 4.1 shows a picture of these motors. The bottom joint of the ankle, used for the lateral motion, drew too much current to operate, due to the heavy weight that was attached to it. This was just one of the problems with Prototype One.

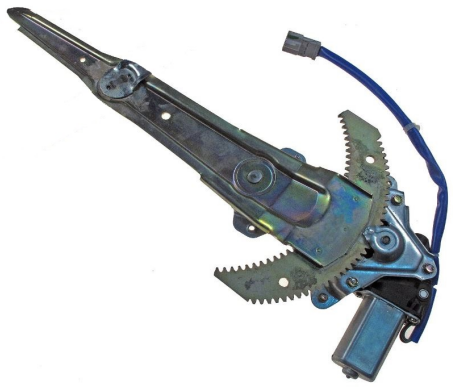

Figure 4.1: Motors from automatic car windows.

Another problem with Prototype One was, when it was designed, friction wasn't taken into account, meaning that when the joints move, most of the links were rubbing against each other causing unnecessary friction. Without having loose links and joints, problems can occur during the balance control.

Prototype Two is the design that is currently being used and tested on. The mistakes made in Prototype One were used to help design Prototype Two. The connecting links do not rub against each other and the whole leg is much lighter.

4.2 Mechanical Structure

4.2.1 Materials Used

The mechanical structure needs to have a lightweight, durable and rigid design. Aluminum alloy was used for the mechanical structure of the robot leg. Aluminum is easier to work with and lighter than steel, making it a more effective material. Aluminum is also cheap and easy to manipulate when designing and building by hand.

The only downfall to aluminum is that it's not as rigid compared to steel. So, to tackle this problem, the links are made a little shorter, so there isn't as much bending when the joints are being actuated. The rigidity of aluminum is 27 GPa compared to steels 79.3 GPa, so you can see just from that steel is about three times more rigid than aluminum.

The whole structure is put together with small nuts and bolts with locking washers. Using nuts and bolts allowed for changes to be made much easier than if permanently fastening the

materials together. Unfortunately, overtime, because of vibration, some of these nuts could come loose, so proper maintenance is essential.

4.2.2 Shaft Design and Connection

The shafts of all the motors and encoders are connected to the joints using set screw hubs. You can see what this looks like in figure 4.2. One is used for the motor side and the other for the encoder side. The motor is 6mm diameter and the encoder is 8mm so the holes had to be machined for the larger diameter. These two hubs are then screwed together so that the motor and the encoder will be in line with each other.

Figure 4.2: Set screw hub for motors and encoders.

If the motor and encoder are not in line with each other, it will cause some mechanical problems. You need the motor and encoder to spin at the same time and at the same angle. If one of them is off, it can cause the whole motor or encoder to move out of place. You can see from figure 4.3 what will happen if the shafts are not connected in a straight line. The red line shows, if one of them is off, the force and or toque will be applied in the Y direction, which will hinder the movement of the shafts. This will cause unnecessary friction and could halt the movement of the joint.

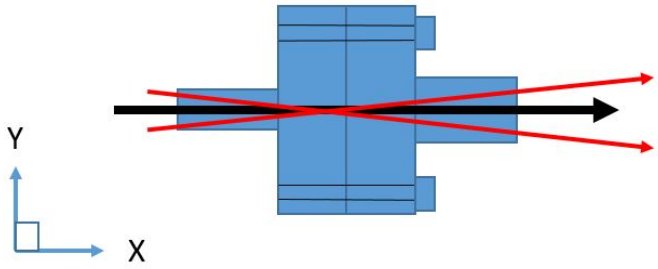

Figure 4.3: Motor and encoder hub connection.

The downfall of the set screw hub is that it can damage the shaft of the encoder or motor. If the set screw is too tight it will bend the shaft and make a little divot in it. You can use a clamping hub, like in figure 4.4, but there is a chance that the shaft could slip within the hub.

Figure 4.4: Clamping hub.

4.2.3 Foot Design

The foot design is important for the stability of the leg. As mentioned in sections 2.7.1 and 2.7.2 the support phase depends on the size of the base of the foot. The foot is roughly around 0.19 m by 0.12 m. It isn't one solid piece of aluminum, there are holes drilled out of it, so the weight is kept to a minimum. The foot is rectangular so it will be easier to determine where the center of pressure is located. Placement of the force moment sensors are located in each of the four corners.

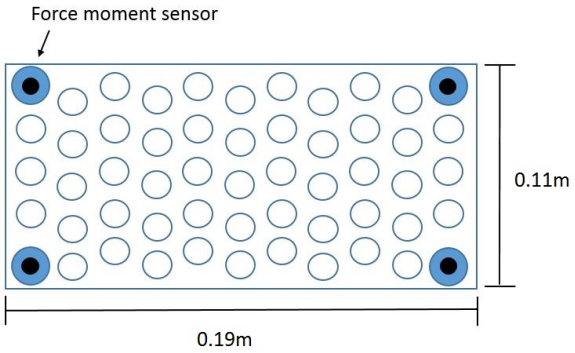

Figure 4.5: Illustration of the base of the foot.

4.3 Means of Actuation

For this specific mechanical design the actuation of the joints will be done by permanent magnet DC motors. The more powerful motors will be placed on the bottom ankle joint, for lateral motion and the knee joint, for the sagittal motion. DC motors have a high enough torque to control all the joints and links. Each joint will have its own motor attached to an encoder.

The bottom ankle joint motor has the higher gear ratio so there will be enough stiffness to control the inertia of the leg. This is important because for the small changes required by the controller the output might not be able to reach the required set point. This will cause an unwanted steady state error. The table below shows which motors were used for each joint and some of their specifications.

	Electric Brushless DC Motor	Voltage (V)	Output Power (W)	Torque (Nm)
Ankle (Lateral)	Faulhaber Motor 3257G024CR	24	83	0.539 stall
Ankle (Sagittal)	Maxon Motor 395823	24	60	0.0856 continuous
Knee	Faulhaber Motor 3257G024CR	24	83	0.539 stall
Hip (Lateral)	Maxon Motor 395823	24	60	0.0856 continuous
Hip (Sagittal)	Maxon Motor 395823	24	60	0.0856 continuous

Table 4.1: Motors used for each joint

4.3.1 Overall Design

The overall design of the structure is supposed to resemble a human. Prototype One resembled a more human like structure because of its size but it wasn't able to be used. Prototype Two has just five degrees of freedom and is very similar to that of a real human leg. One difference is in the ankle, instead of revolute joints, human joints act more like ball joints.

The most important motor, that bears the largest amount of mass is the lateral ankle joint. This joint has to move the whole leg from side to side. If there was two legs then the hip joint would be the most important, because it would have to keep the balance of the stance leg and the swinging leg.

Figure shows what the completed structure looks like. The structure has 5 motors and encoders on each joint and four force moment sensors on the bottom of the foot.

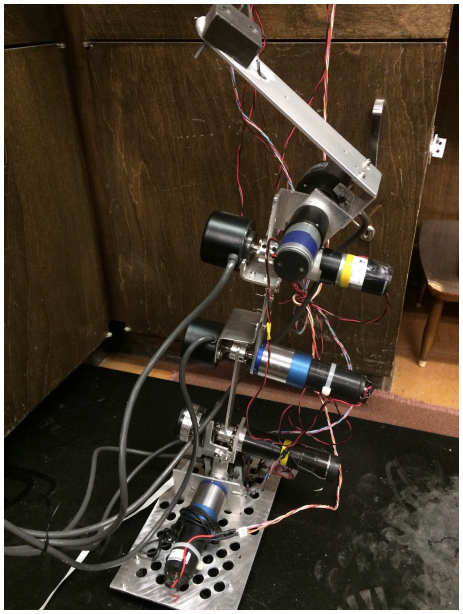

Figure 4.6: Prototype Two.

4.3.2 Construction

The construction of the leg was all done by hand, using various tools and machines including band saw, drill press and files. All the parts were assembled using bolts, nuts and washers. If another prototype was to be made, it could be designed using AutoCad or SolidWorks so the parts could be machined on a computerized numerical control (CNC) machine. If the parts were machined with a CNC it would have made it easier to construct the leg. The hardest part about constructing the leg by hand is getting all the holes to line up. The encoders were the most difficult part to assemble since the diameter of the mounting bracket was so large.

4.4 Electrical Design

The electrical design is centered and controlled by three F28M35H52C1 concerto microcontrollers. They are implemented on the C2000 development boards. There are three boards being used. One board is used for the two bottom encoder joints of the leg, three PWM signals, three direction signals and the four force moment sensors. The second board has two encoders, for the knee and thigh joints, three PWM and three direction signals. The third board has the fifth and final encoder, for the hip. These three boards will be in charge of sending and receiving all the data necessary to balance the leg. The PWM and direction signals will be sent to a separate

motor drive circuit board. There is a motor drive circuit board for the two bottom joints and another board for the top three joints. All of this will be connected to a Windows PC. This communication will involve a c++ program communicating with the development boards and Code Composer Studio software. Figure 4.7 shows a block diagram outlining the whole electrical system of the robot.

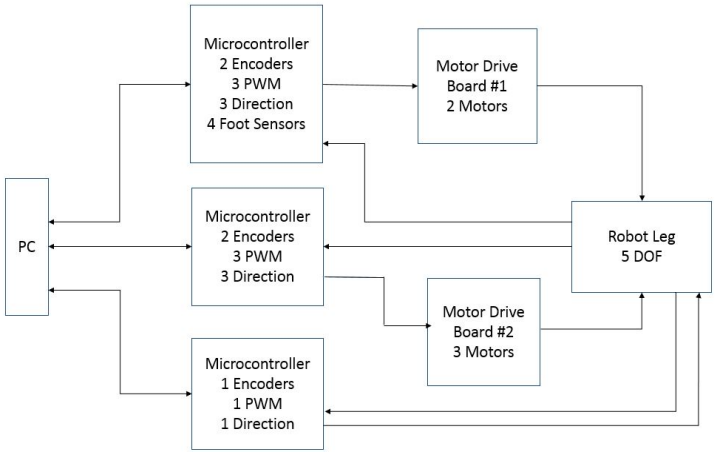

Figure 4.7: Block diagram of the systems electronic flow.

Figure 4.7 shows the flow of information and that the PC, microcontrollers, motor drivers and robot leg are all connected. The c++ program on the computer will communicate with microcontrollers to tell the robot leg what to do. The microcontrollers will tell the motor drivers when to turn on and what direction the motors needs to turn. This information all depends on what the force sensors on the foot read and where the encoders are positioned. The force sensor and encoder information gets sent back from the robot leg to the microcontrollers, which sends it to the PC. They all depend on each other to make the whole system balance.

4.4.1 Sensors (Force Moment Sensor)

The sensors are a very important part to the balance of the robot. Force moment sensors are placed on the bottom of the foot to determine where the center of pressure is located. There is a sensor in each of the four corners of the base of the foot. The sensors that were used for this particular robot are the FC22 Compression Load Cells by measurement specialties.

The FC22 comes in a ten to one hundred pound range with features including low noise,

small size, reliability and fast response time. For the first robot design, the choice of using force sensors with a one hundred pound range was selected. This was due to the great weight that the first prototype had. The second prototype was much smaller and lighter, making the one hundred pound range too big. This also caused it to inaccurately affect the sensors. After changing the range from one hundred pounds to twenty five pounds, the reading worked much better. Figure 4.8 shows what the force sensor looks like, with just three wires, power, ground and the signal.

Figure 4.8: FC2231-0000-0025-L force sensor.

4.4.2 Encoders

The most important part of feedback is the angle at which each motor or encoder is at. This is done by the use of encoders. There are several different types of encoders that can be used, including potentiometers. They can all be used to determine the angle at which all five of the joint angles are at.

4.4.3 Potentiometers

Potentiometers are a great low cost solution to help track the position of the joint angles. They are small and light weight but would produce noise with the output signal that would require filtering. The EVW-AE4001B14 is one type of potentiometer that is used for this type of application.

Figure 4.9: EVW-AE4001B14 potentiometer.

4.4.4 Incremental Rotary Encoder

Another type of joint angle feedback can be done with the use of an incremental rotary encoder. The incremental rotary encoder can be either mechanical or optical. The mechanical type is used typically as a digital potentiometers or in consumer electronics. The incremental rotary encoder is probably the most widely used type of encoder due to its low cost and high resolution. Optical encoders are used when higher speeds and greater precision is needed.

4.4.5 Absolute Rotary Encoder

The absolute rotary encoder is used for Prototype Two's legs joint angle position. The absolute rotary encoder is great for high precise resolution. There is ten bit resolution that uses gray code for the positioning, operating at twelve to twenty four volts. Gray encoding is a system of binary counting in which two adjacent codes differ, by only one bit position. You can see this in the below example.

Sector	Contact 1	Contact 2	Contact 3	Angle (Degrees)
0	off	off	off	0 to 45
1	off	off	on	45 to 90
2	off	on	on	90 to 135
3	off	on	off	135 to 180
4	on	on	off	180 to 225
5	on	on	on	225 to 270
6	on	off	on	270 to 315
7	on	off	off	315 to 360

Table 4.2: Gray code vs binary

The example's transition from one to two only involves one of the contacts to change its state, from on to off or vise versa. This way, either direction the shaft is turned, you will always get the correct value and it doesn't overlap with any other sectors. The reason a change was needed to made from binary code to gray code was that, its very unlikely that switches will change states exactly in order. Table 4.3 will help you understand. Looking at the binary code, you can see that more than one bit can change state during transition. This can cause a problem because in that exact moment the switches can appear to be in the real position or the transitional state, between the other two positions. This can cause a joint to move in the wrong direction which can lead to problems. The gray code solves this problem, by only changing one bit at a time so there is no confusion.

Dec	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Table 4.3: Gray code example

Figure 4.10: E6C3 AG5C Omron absolute encoder

4.5 Electronics

4.5.1 DSP Board

The F28M35H52C1 is the basis for controlling the robotic leg. The microcontroller is part of the concerto family in a multicore system-on-ship microcontroller unit (MCU), with independent communication and real-time control subsystems. The communication is based on the 32-bit ARM Cortex-M3 CPU and it features a wide range of communication peripherals. The communication peripheral that will be used to communicate to the PC is done through ethernet. Real-time control is run using the 32-bit C28x floating-point CPU and features high precision

control peripherals. Table 4.4 shows some of the features that this board has.

	F28M35H52C
CPU	C28x,Cortex-M3
Frequency (MHz)	150
RAM (KB)	136
Flash (KB)	1024
Timers	3x32-Bit CPU
ADC channels	16
USB	1
GPIO	64
Ethernet	1
IO supply (V)	3.3

Table 4.4: DSP board features

The sampling time for system is set to 10ms. Since there are 3 microcontrollers being used, there is a slight delay between the samples from each microcontroller, after a few hundred samples. The delay isn't big enough to disrupt the system negatively.

4.5.2 Motor Drive Circuit Board

There are two motor drive circuit boards. The first board is for the two bottom joints, the sagittal and later motion. The second board is for the top three joints, two lateral and one sagittal motion. The circuit boards use direction signals and pulse width modulation signal inputs to control the output of the corresponding motors. To drive the DC motors, LMD18200 3A H-Bridge motor driver chips are used.

The LMD18200 main design is for control in motion applications. The device is built using a multitechnology process which combines both bipolar and CMOS control circuitry with DMOS power devices. This part is perfect for driving DC and stepper motors. Some of the features include 3 amps of continues output, operates at supply voltages up to 55 volts, TTL and CMOS compatible inputs and a shortened load protection.

As for the control of the motors, the microcontroller has its own built in feedback so all that needs to be sent, is the duty cycle for the motors. There are motor control libraries in the C28x,

that are taken advantage of.

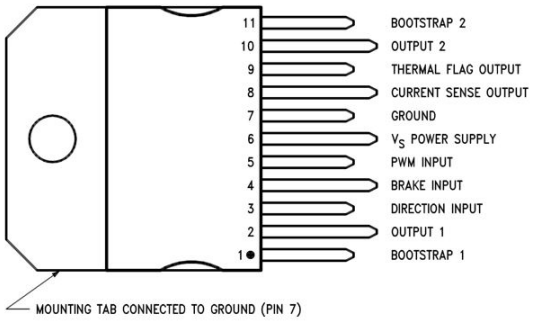

Figure 4.11: LMD18200 motor drive chip with the pin outs.

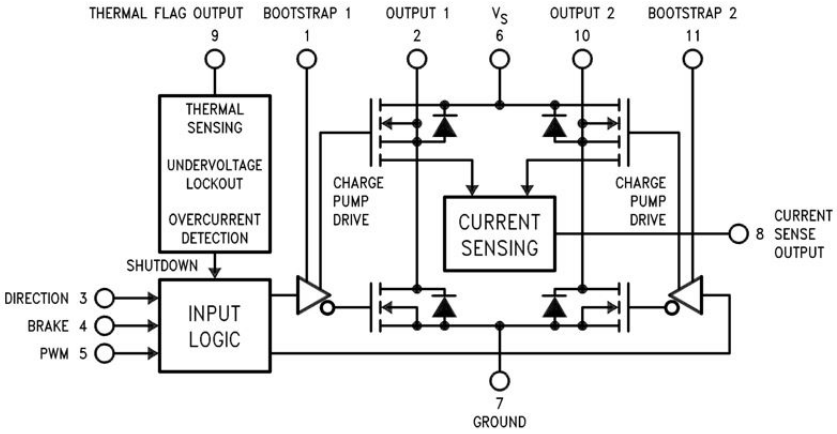

Figure 4.12: LMD18200 functional block diagram.

Chapter 5

Control System

5.1 Introduction

For a single five degree of freedom robot leg, a PID controller will work just fine. The PID control will be used to balance the leg in the upright position, when a disturbance is applied. To do this, an understanding of how a PID controller works is important and how to alter the incoming signals for noise, so you will have a clean signal to control. That will lead you to the next section, Digital Filter Design, where the process of filtering the noisy signal is done.

5.2 Digital Filter Design

There are many sources of error that can cause a control scheme to deviate from its course. One of the reasons for this is unwanted noise and interference. If the signal is too noisy it can cause the joints of the robot to follow this noise and create an unstable path. Since the force moment sensors are whats used to control the balance of the robot, the signal must be as clean as possible and be free of noise. From figure 5.1, you can see that this signal isn't smooth and will cause problems for the control system. The joints that are associated with the front to back motion will follow the oscillations and the manipulators leg will begin to shake.

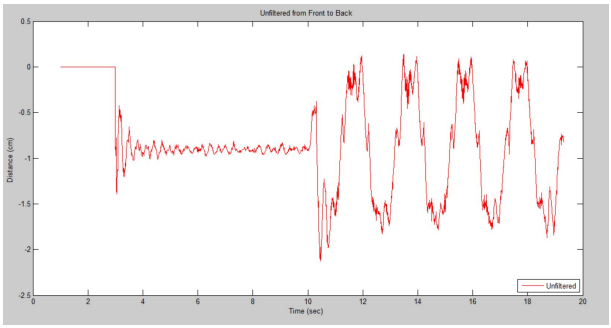

Figure 5.1: Unfiltered force moment sensor signal from front to back movement.

To get rid of this problem, a filter is introduced to the signal. The filter will be used to clean up the noise and get rid of the oscillations that will cause the shaking. The filter that was selected for this job is the Butterworth Filter.

5.2.1 Butterworth Filter

A Butterworth filter is designed to have such a cutoff frequency, that it's as flat as possible, in the passband. The Buttterworth filter can also be referred to as a maximally flat magnitude filter. The term maximally flat is because it has no ripples in the passband and rolls off towards zero in the stopband. You can see in the bode plot (figure 5.2) that the response slopes off linearly towards negative zero. A Butterworth filter has a monotonically changing function with omega, the angular frequency. Butterworth filters have a more linear phase response, which is ideal for nice smooth transitions.

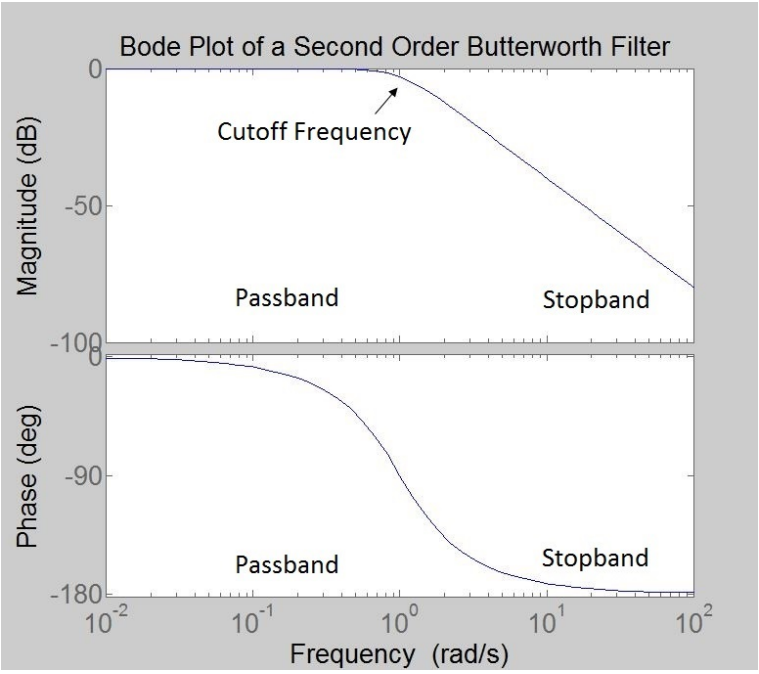

Figure 5.2: Bode plot of a Butterworth filter.

The Z-domain transfer function can be seen below, for a second order IIR filter.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \tag{5.1}$$

$$y[n] = -\sum_{k=0}^{2} a_k y[n-k] + \sum_{k=0}^{2} b_k y[n-k] \tag{5.2}$$

The cut-off frequency is designed so that the frequency is low enough to eliminate most of the high frequency noise, and at the same time, high enough so that the step response is reasonable.

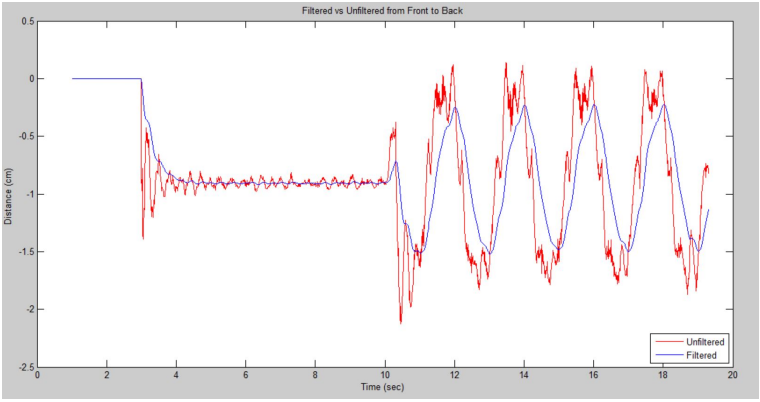

Figure 5.3: Filtered signal from the force sensors.

You can see from figure 5.3 that the Butterworth filter filtered out the unwanted noise and spike. The Butterworth filter provided a nice smooth path for the joints of the manipulator to follow. This will result in a more stable upright leg, which you can see in the below comparison of Butterworth, Chebyshev and Elliptic. You can see from figure 5.4 that the Butterworth has no ripple and rolls off more smoothly. These are characteristics that are essential to the type of signal that is needed. It should be noted that the Butterworth filter in the experiment is a 2nd order filter with a cutoff frequency of 0.5Hz and sampling period of 0.01 seconds. The parameters for the Butterworth are: $b_0 = 0.0002414$, $b_1 = 0.0004827$, $b_2 = 0.0002414$, $a_0 = 1$, $a_1 = -1.9556$ and $a_1 = 0.9565$.

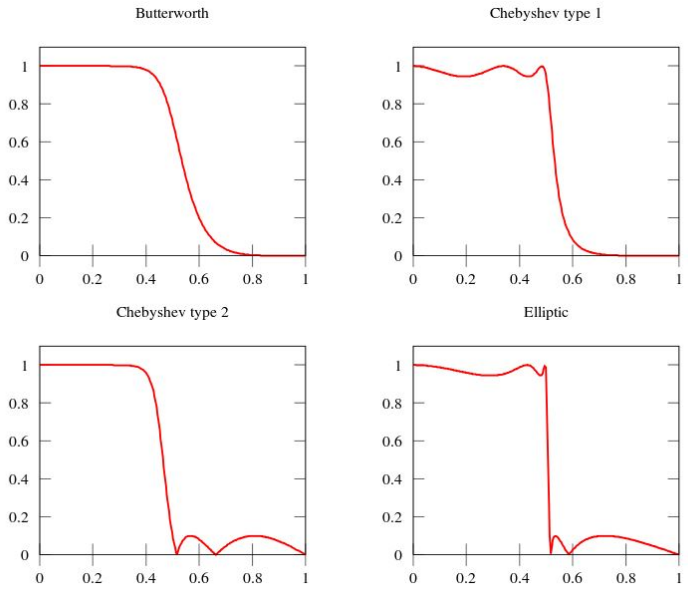

Figure 5.4: The Butterworth filter as compared to Elliptic and Chebyshev.

5.3 Flowchart and Block Diagram

One way to understand the control scheme for the balance of the leg is to observe it in a flowchart. The following flowchart is a very simplified version of the whole control process.

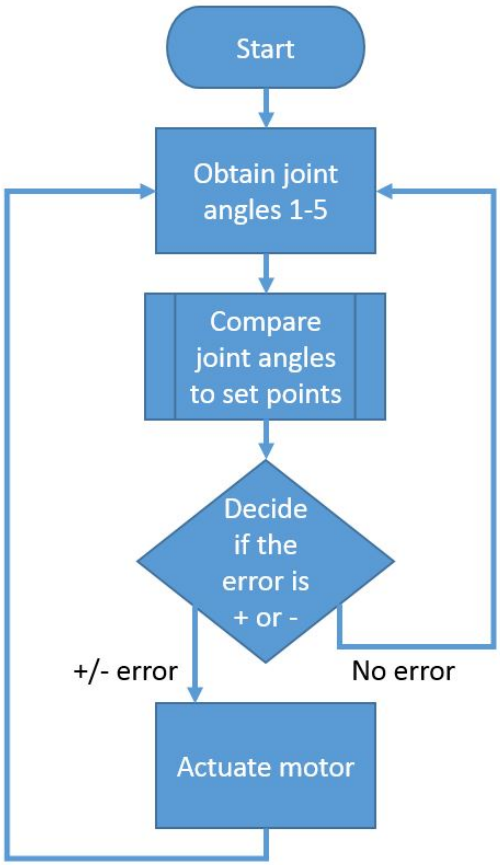

Figure 5.5: Flowchart of the proposed control system.

Looking at figure 5.5, you can see the control scheme in general is very simple. The process starts by taking the encoder readings from each joint and comparing them to the set points. If these encoder reading are not at the set points then the motors will turn, trying to make each joint reach there set point.

The next step is to understand the block diagrams and how a PID controller works. The simple block diagram can be seen in figure 5.6. The set point is r, e is the error, u is the controller output and y is the plant output. Comparing figures 5.5 and 5.6, you can see the relationship between them. y is the joint angles you obtain, e is the comparison between the set point r and

y and u is the signal that will be used to actuate the motors. This process will continue like this until the operation is stopped.

This then brings up the question of what is inside the controller box. A PID controller is used to control the balance of the leg and keep the joints at the proper set points during the experimental set up. The output of the controller, or the input to the plant, can be seen a follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_p \frac{de}{dt} \tag{5.3}$$

An explanation to equation (5.3) can be found in figure 5.6. The variable e represents the tracking error, the difference between the desired input value or set point (r) and the actual output (y). The error signal (e) is then sent to the PID controller. The controller will then compute the derivative of this error signal as seen in equation (5.3). Then the control signal (u) will be equal to the proportional gain (K_p), times the magnitude of error plus the integral gain (K_i), times the integral of the error, plus the derivative gain (K_d), times the derivative of error.

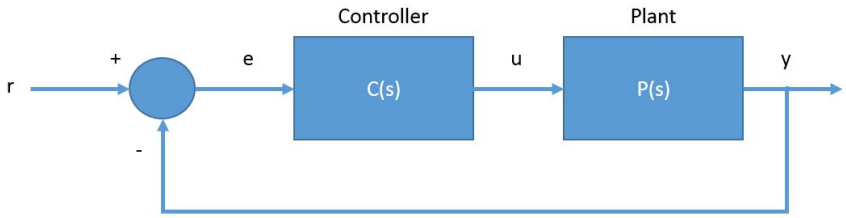

Figure 5.6: Closed loop block diagram.

This new control signal (u) will be sent to the plant and a new output (y) will be obtained. This new output will then be fed back and compared to the set point and a new reference signal (r) will be determined. The controller will then take this new error signal and compute the results just like before. This will run for as long as it is needed.

To obtain the transfer function for the PID controller you take the Laplace transform of equation (5.3).

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{5.4}$$

The proportional controller part will have an effect on reducing the rise time and reduce the steady state error, but it will never eliminate it. The derivative part of the controller can increase

the stability of the system, reducing the overshoot, and improving the transient response. The integral part will help eliminate the steady state error.

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small change	Decrease	Decrease	No change

Table 5.1: Controller parameter effects.

Since K_p, K_i and K_d are dependent on one another, changing one of them will affect the others.

5.4 PID Controller Design

All the joints are controlled with a PID controller. PID control is implemented on the two ankle joints, knee joint and the two hip joints. The two ankle joints are more important in maintaining stability, compared to the knee joint.

The controller is designed so that each individual joint will have its own gain tuned to a specific performance.

The PID controller's time domain equation is shown in equation (5.3). Equation (5.5) will show the frequency domain and in equations, (5.6) and (5.7), the discrete implementation.

$$u(s) = (K_p + \frac{K_i}{s} + K_d s)E(s) \tag{5.5}$$

$$u(n) = K_p e[n] + K_i \sum_{k=0}^{n} e[n-k] + k_d \frac{e[n]-e[n-1]}{T_s} \tag{5.6}$$

$$e(n) = r[n] - y[n] \tag{5.7}$$

T_s: The sampling period

$E(s)$: Error

$e[n]$: The discrete error

$e[n-1]$: Previous samples error

$r[n]$: The setpoint

$y[n]$: Measured angle (plant output)

$u[n]$: Discrete control input

5.5 Balance Control

To control the balance of the leg, readings from the four force moment sensors on the bottom of the foot are used. These readings will be used in the center of pressure calculation discussed in the next subsection. The COP calculation will be used in the controller to balance the leg.

5.5.1 Center of Pressure (COP) Calculation

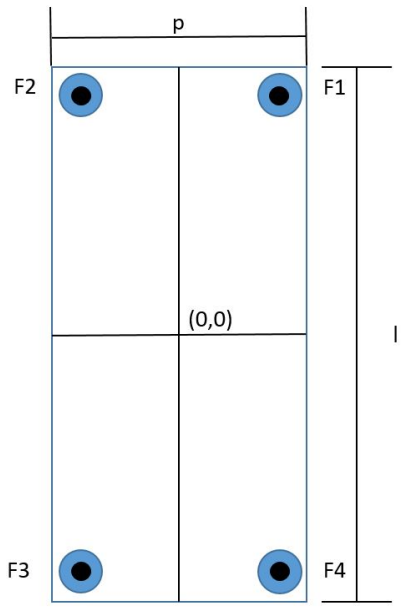

Figure 5.7: Block diagram for a PD controller.

To determine the center of pressure, the foot must be sectioned off into four coordinates. These four coordinates will hold a force sensor in each of them. The center of pressure will have two separate calculations, one for the sagittal motion (front to back) and one for lateral motion (side to side).

$$Front2back = \ [\frac{F_1+F_2-F_3-F_4}{F_1+F_2+F_3+F_4}]\frac{l}{2} \tag{5.8}$$

$$Side2side = \ [\frac{F_1+F_4-F_2-F_3}{F_1+F_2+F_3+F_4}]\frac{p}{2} \tag{5.9}$$

$Front2back$: Distance from center towards the front or back (cm)

$Side2side$: Distance from the center towards the left or right side (cm)

F_1: Force moment sensor 1

F_2: Force moment sensor 2

F_3: Force moment sensor 3

F_4: Force moment sensor 4

l: Front to back length (m)

p: Side to side length(m)

To compute the center of pressure, equation (5.8) must be implemented first. Equation (5.8) takes the four force moment sensor readings and computes the difference from the center of the foot. The value for $Front2back$ will be the distance from the center of the foot to the center of pressure in the sagittal motion. So, if the value for $Front2back$ equaled positive (m), then the center of pressure would be towards the front. If the value equaled negative (m), then the center of pressure would be towards the back of the foot. The same can be said for the $Side2side$ calculation. If it's computed to be a positive value, the pressure is leaning towards the right and if the value is negative, the pressure is leaning towards the left.

Sensor	Front2back	Side2side
F_1	Positive	Positive
F_2	Positive	Negative
F_3	Negative	Negative
F_4	Negative	Positive

Table 5.2: Table used for center of pressure calculations.

5.5.2 Center of Pressure Example Calculation

The following is an example to help in the understanding of the center of pressure calculations.

Sensor	Reading
F_1	30
F_2	20
F_3	10
F_4	10

Table 5.3: Example sensor readings.

$$Front2back = \left[\frac{F_1+F_2-F_3-F_4}{F_1+F_2+F_3+F_4}\right]\frac{l}{2} = \left[\frac{30+20-10-10}{30+20+10+10}\right]\frac{20}{2} = 0.0429m \tag{5.10}$$

$$Side2side = \left[\frac{F_1+F_4-F_2-F_3}{F_1+F_2+F_3+F_4}\right]\frac{p}{2} = \left[\frac{30+10-20-10}{30+20+10+10}\right]\frac{10}{2} = 0.0071m \tag{5.11}$$

After the calculations, you can determine where the center of pressure is. Looking at the result from equation (5.10), the center of pressure is leaning towards the front by 0.0429m. The same for equation (5.11), the center of pressure is leaning 0.0071m towards the right.

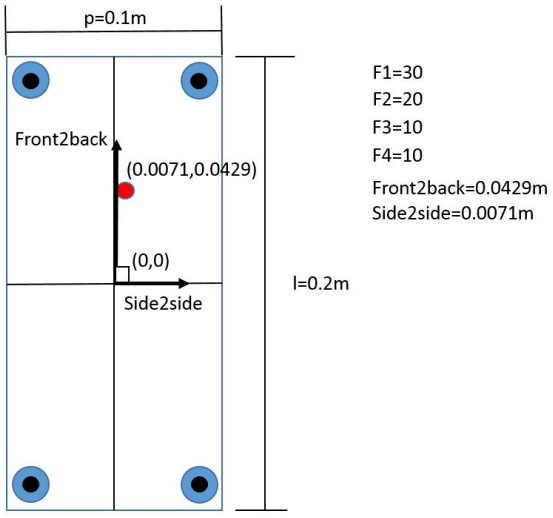

Figure 5.8: Red dot shows where the center of pressure will be.

Therefore, you can see from figure 5.8 that the center of pressure will lie near the front and slightly to the right.

5.5.3 Implementing the COP Calculation into the Controller

Once the center of pressure is determined, deciding on what to do with it is the next step. The calculations from equations (5.8) and (5.9) will serve as the new setpoints. Each motor acts in a certain direction, either from front to back (sagittal) or side to side (lateral).

Motor	Setpoint
Motor 1	$Side2side \times K_{FB}$
Motor 2	$Front2back \times K_{SS}$
Motor 3	0 or disturbance
Motor 4	0 or disturbance
Motor 5	0 or disturbance

Table 5.4: Motor/joint setpoints.

Looking at table 5.4 you can see that Motors 1 and 2 will follow the Side2side calculation or the Front2back calculation. Motors 3, 4 and 5 will either have disturbance applied to them or stay at their zero position. Since the COP is a unit of distance in m, the Side2side and Front2back calculations will be multiplied by a certain gain, $K_{FB} = -0.15$ and $K_{SS} = -0.08$. This gain will transform the units from m to degrees, so the motors will turn appropriately. The value of the setpoint will be the opposite sign depending on the Front2back and Side2side calculations.

Using the example in figure 5.8, you will want Motors 1 to turn left and Motor 2 to turn back, so the center of pressure (red dot) turns towards the center (0,0). This process is shown in the block diagram (figure 5.9) to give a more clear understanding.

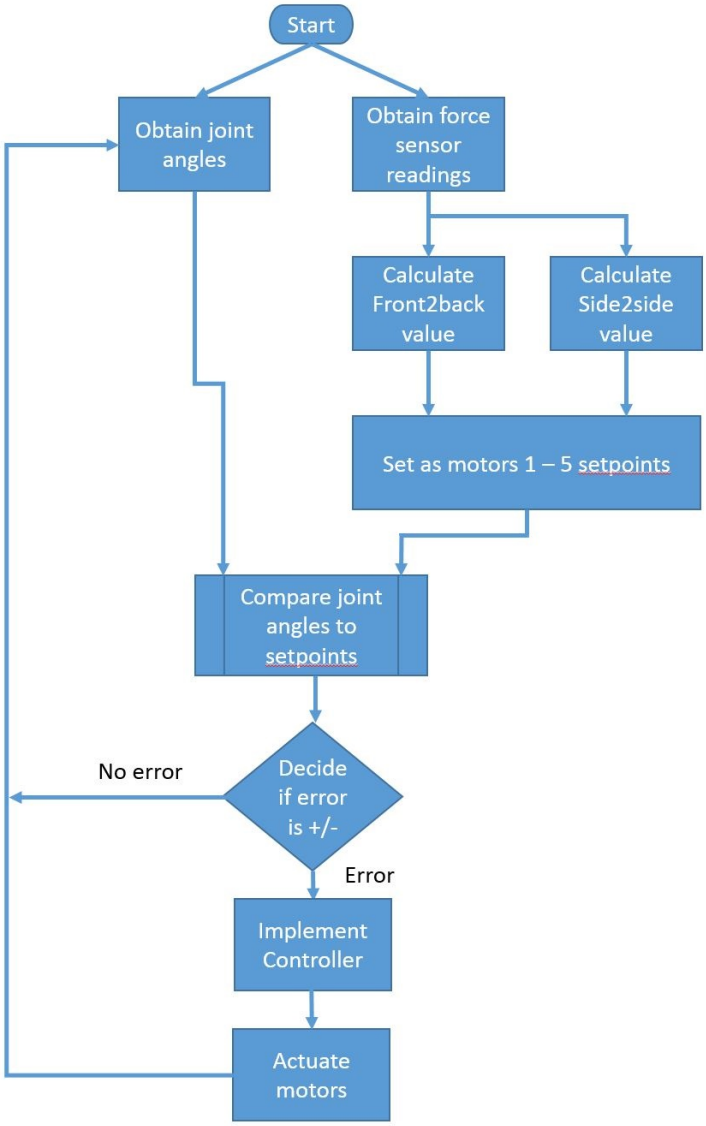

Figure 5.9: Red dot shows where the center of pressure will be.

Chapter 6

Simulations

6.1 Simulation Model

An experimental robot, created by former Lakehead University graduate student Andrew Cudoski [3], was the main influence behind the simulation model's design. The simulation model will be used to design and compute the walking trajectories, zero-moment-point and center of mass. All three computations will include the use of DH parameters and Newton Euler recursion.

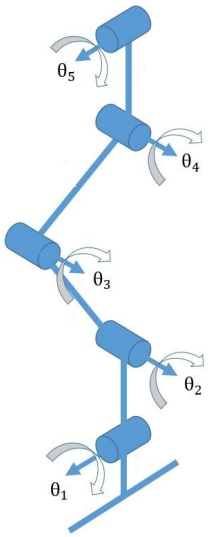

Figure 6.1: Simulation model.

Figure 6.1 is a representation of the simulation model. This particular model has five degrees of freedom with all five of the joints being fully actuated. This leg could potentially be used as

the right or left leg of a much bigger system. There is a massless upper body with a square foot as the support.

6.2 Link Parameters

The following are the link parameters that were used for the simulation model. The parameters for the experimental part couldn't be obtained because all the parts were made by hand and or obscure shapes. These parameters can be used to calculate the center of mass, zero moment point and the trajectory generation.

Link Number	Link Name	Mass (kg)	$\vec{r}_{i,ci}$(m)$\times 1 \times 10^{-2}$	$\vec{r}_{i+1,ci}$(m)$\times 1 \times 10^{-2}$	$\vec{r}_{i,i+1}$(cm)
1	foot	0.646	(-0.00882,0.02959,-0.0197)	(0.00882,-0.00699,0.0197)	(0,0.0226,0)
2	ankle	0.564	(0.06966,-0.00262,0.02779)	(-0.00436,0.00262,-0.02779)	(0.0653,0,0)
3	shin	0.392	(0.25863,0.00037,0.05991)	(-0.08863,-0.00037,-0.05991)	(0.17,0,0)
4	thigh	1.224	(0.24993,0.00002,0.02559)	(-0.08293,-0.00002,-0.02559)	(0.160,0,0)
5	hip	0.229	(0.10663,0.05991,0.00052)	(-0.03563,-0.05991,-0.00052)	(0.071,0,0)

Table 6.1: Mass and center of mass for the robot links.

Link Number	Link Name	Inertia Matrix (kgm^2)
1	foot	$\begin{bmatrix} 1830950.43 & -91559.15 & -104215.93 \\ -91559.15 & 2213772.77 & 78621.04 \\ -104215.93 & 78621.04 & 579380.56 \end{bmatrix} \times (1 \times 10^{-9})$
2	ankle	$\begin{bmatrix} 2292134.59 & 794.06 & -12903.57 \\ 794.06 & 1637642.22 & 5737.41 \\ -12903.57 & 5737.41 & 1068580.16 \end{bmatrix} \times (1 \times 10^{-9})$
3	shin	$\begin{bmatrix} 2098163.66 & 44269.45 & -323.75 \\ 44269.45 & 3000992.32 & 797.89 \\ -323.75 & 797.89 & 1190933.39 \end{bmatrix} \times (1 \times 10^{-9})$
4	thigh	$\begin{bmatrix} 3006750.60 & -72.91 & 259609.69 \\ -72.91 & 8591311.73 & 2260.29 \\ 259609.69 & 2260.29 & 6667178.05 \end{bmatrix} \times (1 \times 10^{-9})$
5	hip	$\begin{bmatrix} 2361441.77 & 52721.73 & 3485.92 \\ 52721.73 & 1163049.86 & 16081.48 \\ 3485.92 & 16081.48 & 1319604.34 \end{bmatrix} \times (1 \times 10^{-9})$

Table 6.2: Inertia matrices for each link.

6.3 Trajectory Generation Results

The trajectory generation is the path that each joint will take, during the standing and walking phase. Each joint will affect one direction at a time. Joints 1 and 5 will move in the lateral motion while Joints 2, 3 and 4 will move in the sagittal motion (Chapter 3). The walking and standing pattern can be referred to as the gait cycle. Subsection 2.7 "The Gait Cycle" will help in understanding the simulation results.

6.3.1 Standing Phase

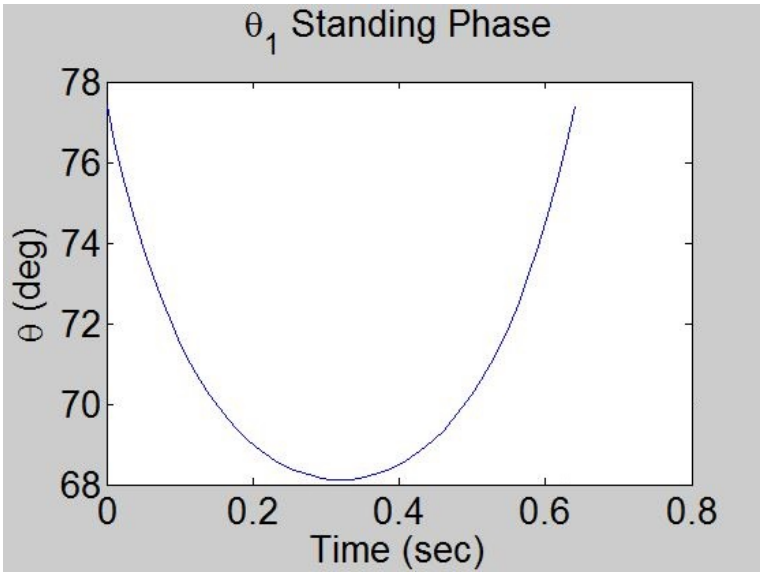

Figure 6.2: θ_1 (ankle) during standing phase

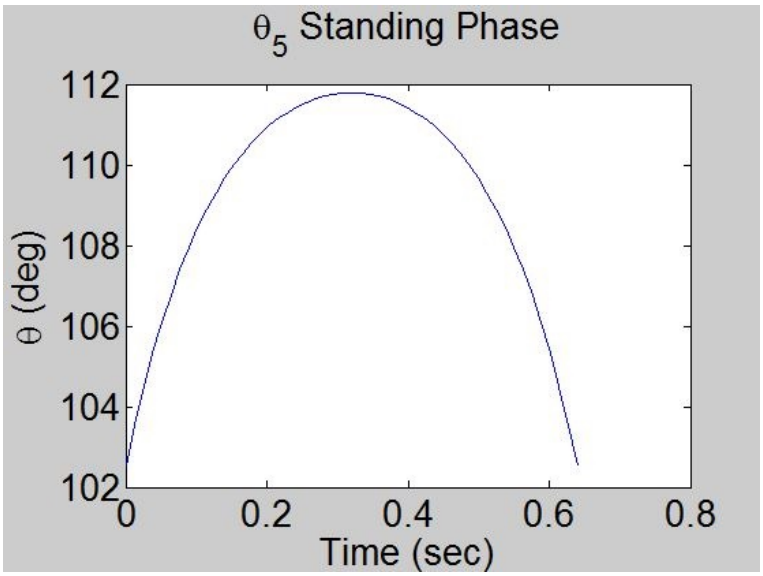

Figure 6.3: θ_5 (hip) during standing phase

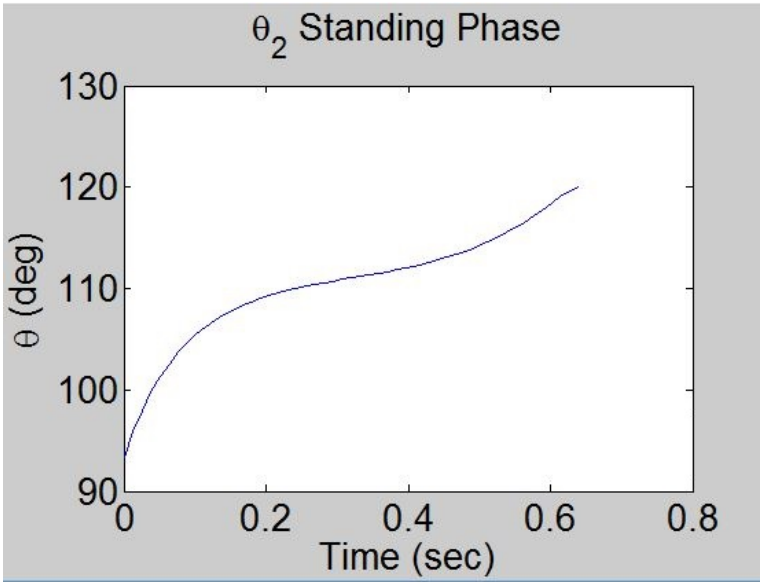

Figure 6.4: θ_2 (ankle) during standing phase

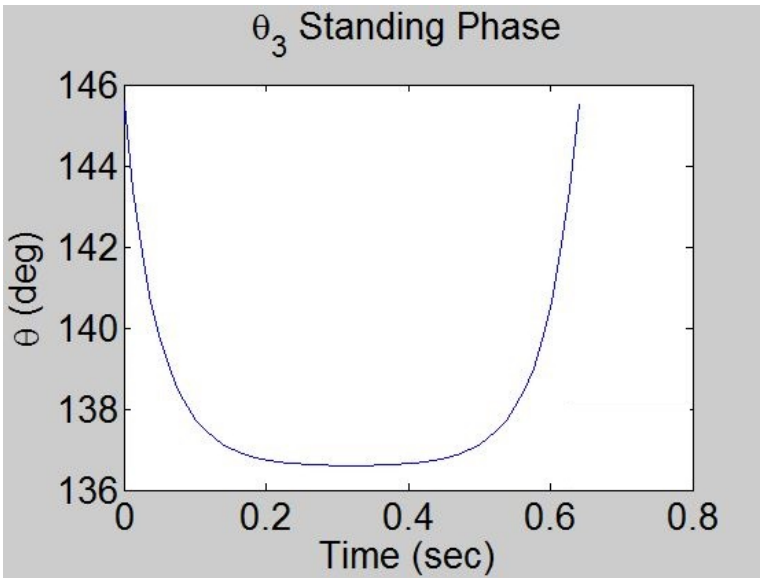

Figure 6.5: θ_3 (knee) during standing phase

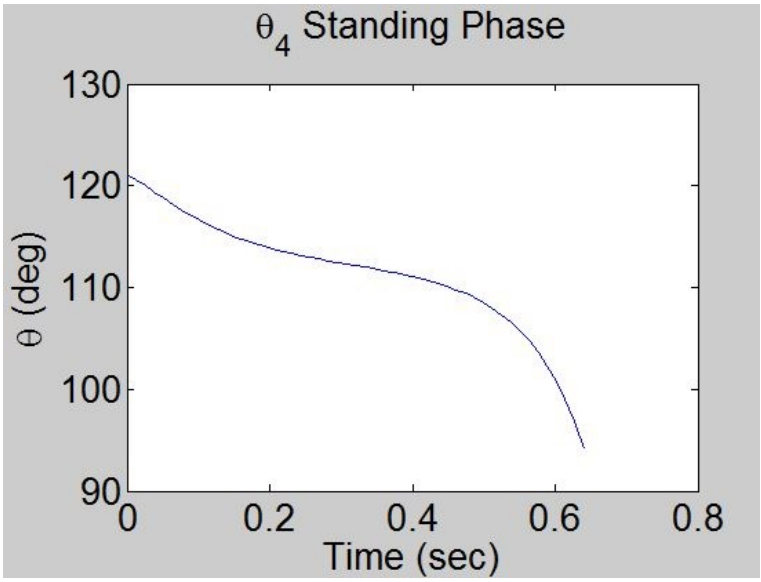

Figure 6.6: θ_4 (thigh) during standing phase

All of the figures are plotted for a single period, with a duration of 0.64 seconds. The horizontal axis represents the time domain (seconds) and the vertical axis represents the joint angle (rad).

Figures 6.2 and 6.3 correspond to Joints 1 and 5, the ankle and hip joints. These joints turn in a lateral motion from side to side. In the plots you can see there is a slight curve that these joints follow. Since the single leg is taken from a two leg system, the lateral joints are bending slightly so that the opposite leg has more ground clearance. Also, this causes the mass to be shifted to the leg that's flat on the ground in the standing phase. You can see from those two figures that the joint starts at one angle, then as the other leg starts to take the walking motion the joint will turn to compensate for the weight. At 0.32 seconds the walking foot should be at it's max height and the standing leg will be tilted as far as it needs too.

The other three joints are in the sagittal motion. Since the leg isn't walking, the joints only have to slightly bend, due to the opposing leg walking. These three joints are the ankle, knee and hip. Looking at figure 6.4, the ankle is moving forward as the angle starts at about 1.6 rad and moves to a final position of about 2.01 rad. The difference is about 0.41 rad or 23 degrees. This is just a forward motion that the ankle will take during the standing phase. Similarly, the knee (figure 6.5), will move forward but will then return to the same angle. As for the hip, seen

in figure 6.6, the joint angle will decrease as the hip bends forward.

6.3.2 Swinging Phase

As the standing phase comes to an end, the leg will then return to the swinging phase. The walking phase period is just as long as the standing phase, 0.64 seconds.

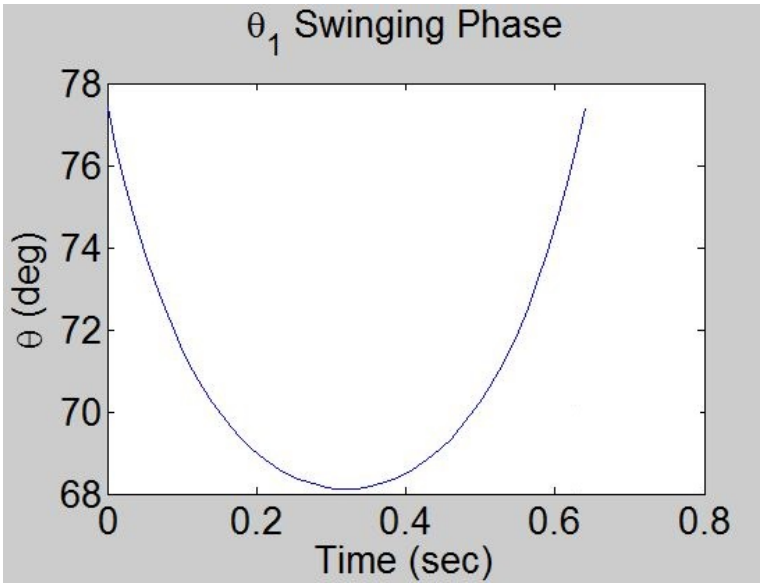

Figure 6.7: θ_1 during swinging phase

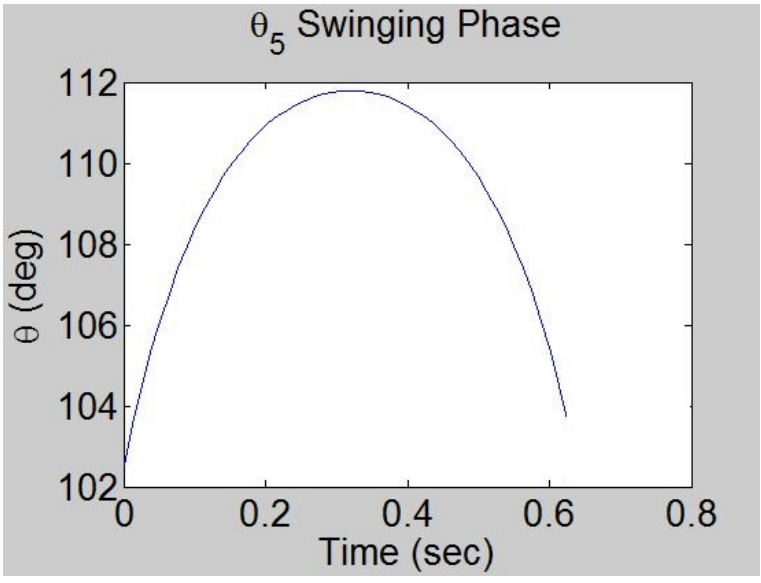

Figure 6.8: θ_5 (hip) during swinging phase.

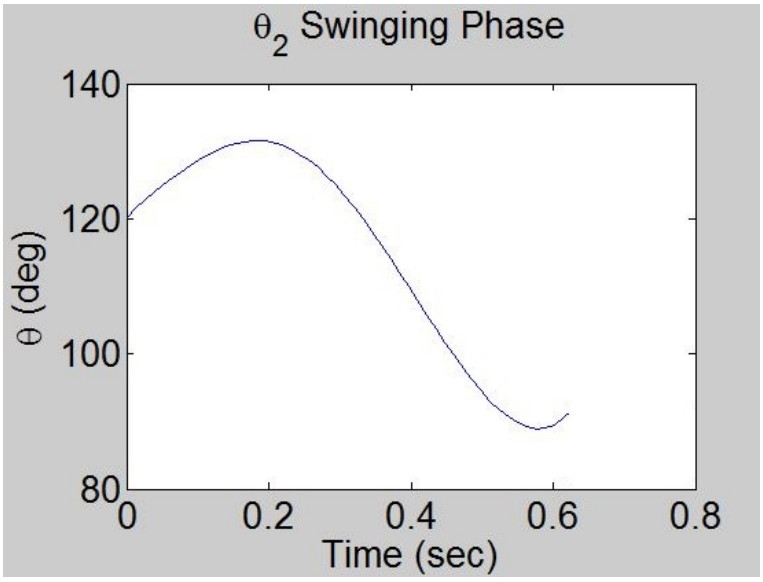

Figure 6.9: θ_2 (ankle) during swinging phase

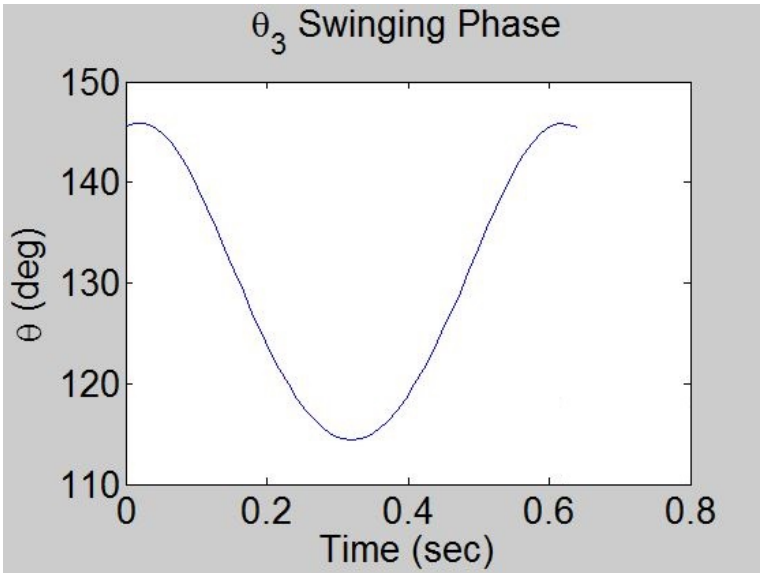

Figure 6.10: θ_3 (knee) during swinging phase

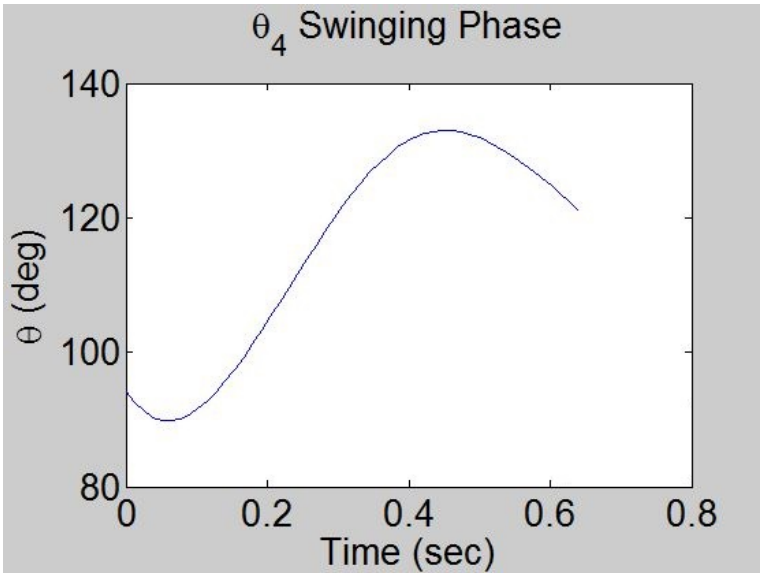

Figure 6.11: θ_4 (thigh) during swinging phase

Joints 1 and 5 are the same, when compared in the standing phase. During the swinging phase, the weight of the swinging leg will be shifted to the side of the standing leg, to help for balance control which was discussed in the standing phase results. Now, the results for the rest of the joints will be different. The first joint of discussion is the second ankle joint. The joint is in the sagittal motion, so it will be used to move the foot in the forward motion. The ankle will bend forward, as the back of the foot leaves the ground, then, as it begins to lower back down the angle will change, so the back of the foot touches down first.

The next joint is the knee (figure 6.10). The knee is in charge of moving the links above it, the thigh and the hip. The movement in the swinging phase acts similar to what was seen in the standing phase. It starts off at a certain angle and it will finish at that same angle. The knee will begin to move forward and swing the rest of the upper body with it. The knee is important because it is the joint that joins the lower part and the upper part of the leg together.

The last joint is the thigh. The thigh is used to help swing the upper body weight, producing a forward momentum. The plot shows that when the thigh is in movement, it will start in deceleration, then quickly accelerate for half the period of the walking phase, before it starts to decelerate, before the foot touches the ground again.

6.3.3 Gait Cycle Summary

- Heal touches the ground, while the hip is bent and the knee is fully extended.
- As the whole foot begins to touch the ground, the knee and hip joint angles will start to decrease.
- When the foot is fully flat on the ground, the hip will start to extend, the knee and ankle will start to bend slightly.
- In mid stance, the knee will be fully extended and support will come from one single leg. The second leg should be the maximum height off the floor.
- The leg will then prepare for the stance foot to be lifted off the floor.
- While the swing leg touches the ground, the stance leg will begin to leave the ground. The knee and hip will be bending while the ankle starts the motion for the foot to leave the floor.
- When the front of the foot leaves the ground the hip will become less extended and the knee will begin to extend, lifting the foot off the ground.
- During the early stages of the swing, the hip and knee will start to extend and the ankle will start to decrease its angle back to a flat position.
- During mid swing, the hip, knee and ankle will all start to bend, preparing itself for deceleration and for the foot to touch the ground

6.4 Newton-Euler

The Newton-Euler procedure is used to determine the forces and toques of each individual link. Looking at figure 6.12, you can see a basic idea of how the forces and torques are calculated. Starting from the trajectory generation, the DH parameters are obtained, as well as the rotation matrices. Then from the DH parameters, the forwards kinematics and forward recursion is solved, then the end and center accelerations can be obtained. As for the backward recursion, it's obtained from the rotation matrices and the joint forces, and torques can be found.

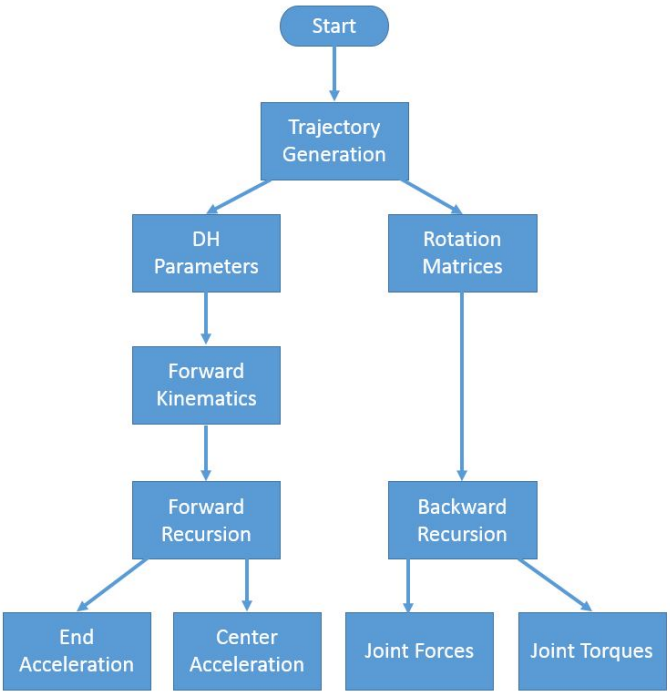

Figure 6.12: Newton Euler block diagram.

6.4.1 Standing Phase

Figures 6.13 through 6.22 show the forces and torques of each link during the standing phase. In all of the, plots there is a spike at the initial startup. The spike is due to the initial conditions. When the simulation starts, the joint angles are at 0 degrees, they need to move to their initial conditions as quickly as possible, causing the spike in the force and torques. After the spike, the values are nice and smooth. The torques are measured in $Newton\ m$ and the forces are measured in $Kgm\ /s^2$. There isn't much change during one period because this is the standing phase, so there isn't much movement in the joints.

Figures 6.23 through 6.32 show the plots for the center and end accelerations of each link. Just like the forces and torques, the accelerations have a spike at the beginning then they start to level off smoothly. There isn't much deviation because it's the standing phase. The acceleration is measured in m/s^2.

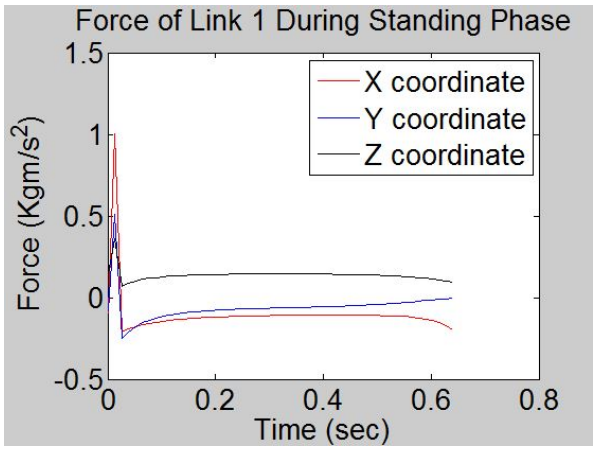

Figure 6.13: Force of link 1 during standing phase

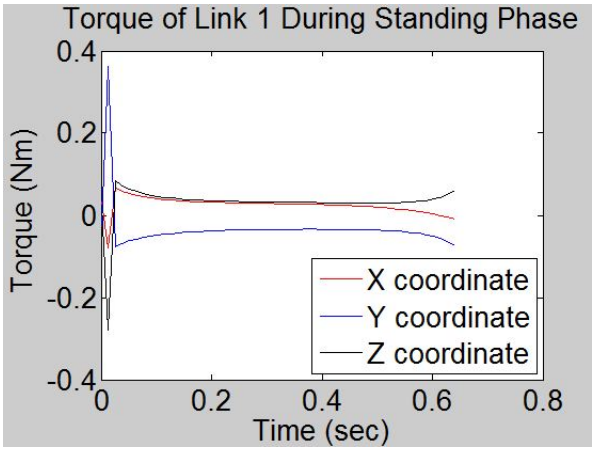

Figure 6.14: Torque of link 1 during standing phase

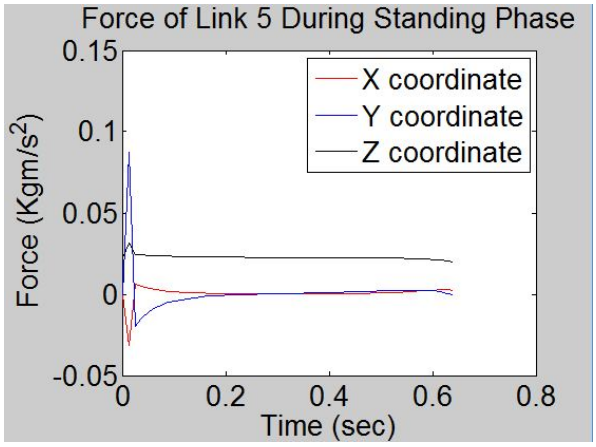

Figure 6.15: Force of link 5 during standing phase

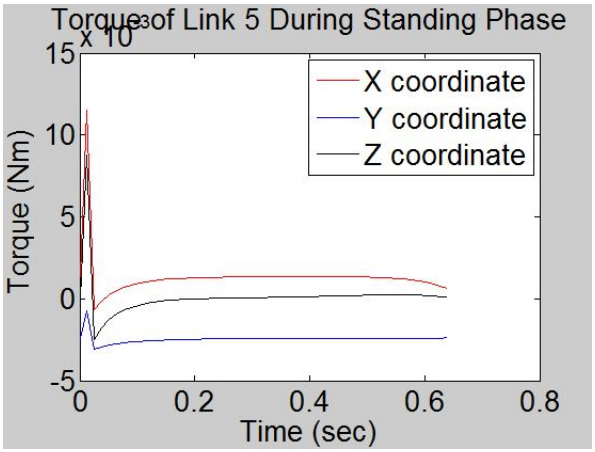

Figure 6.16: Torque of link 5 during standing phase

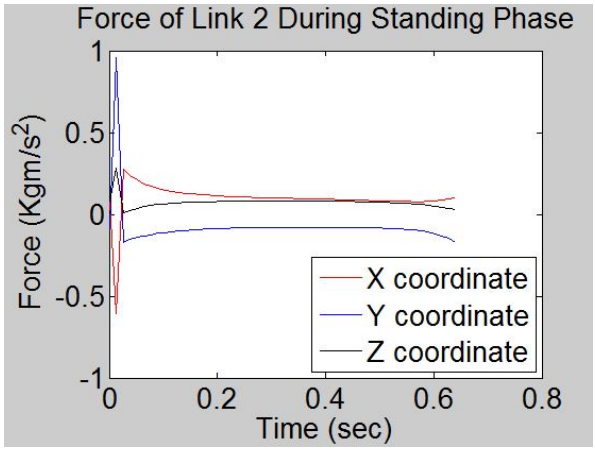

Figure 6.17: Force of link 2 during standing phase

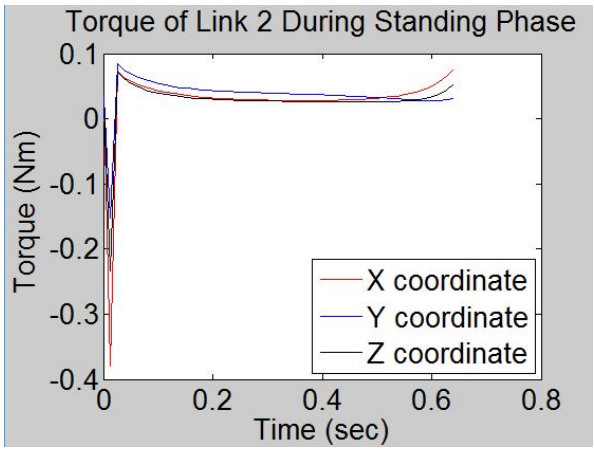

Figure 6.18: Torque of link 2 during standing phase

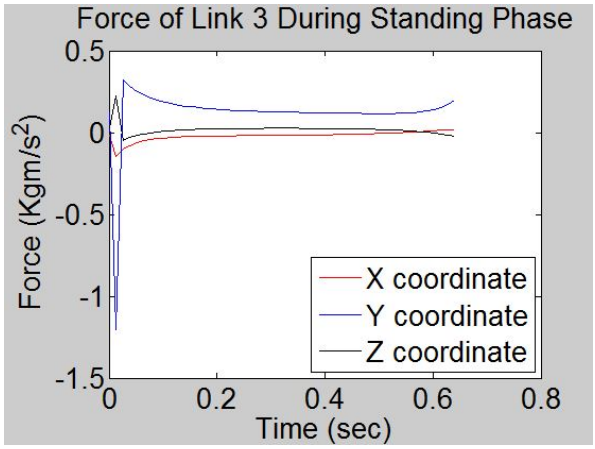

Figure 6.19: Force of link 3 during standing phase

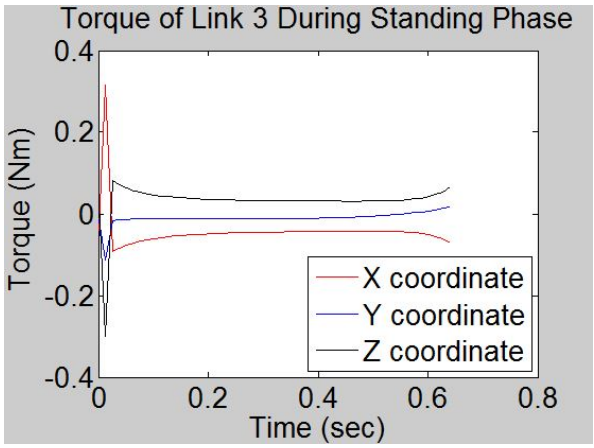

Figure 6.20: Torque of link 3 during standing phase

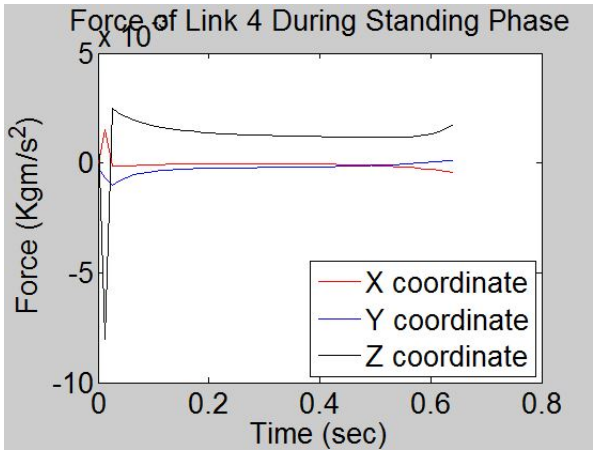

Figure 6.21: Force of link 4 during standing phase

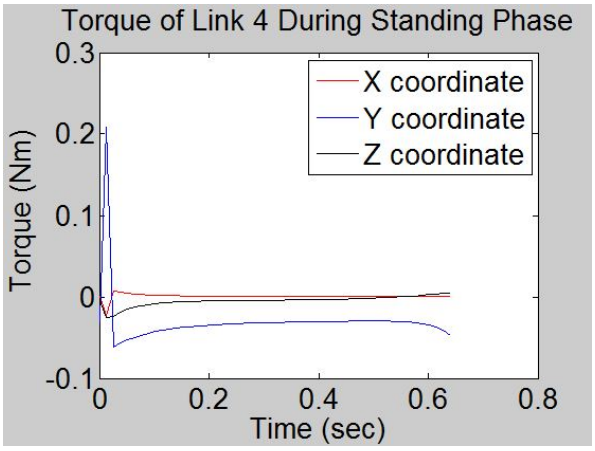

Figure 6.22: Torque of link 4 during standing phase

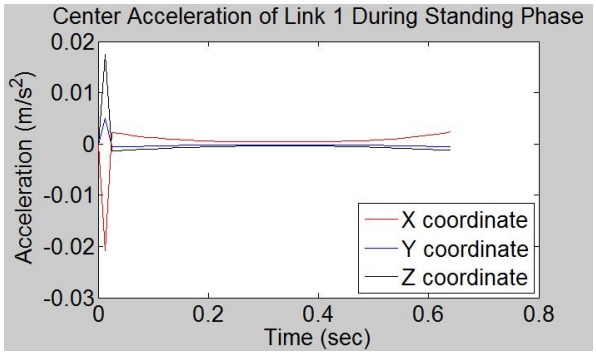

Figure 6.23: Center acceleration of link 1 during standing phase

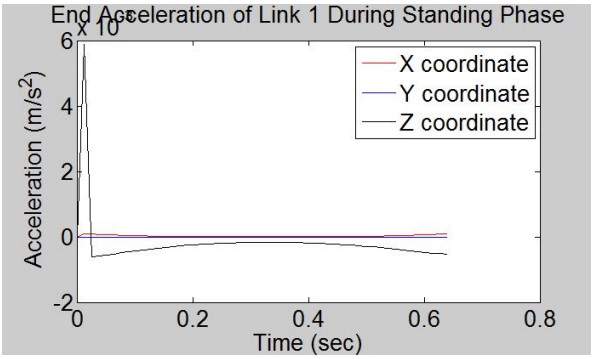

Figure 6.24: End acceleration of link 1 during standing phase

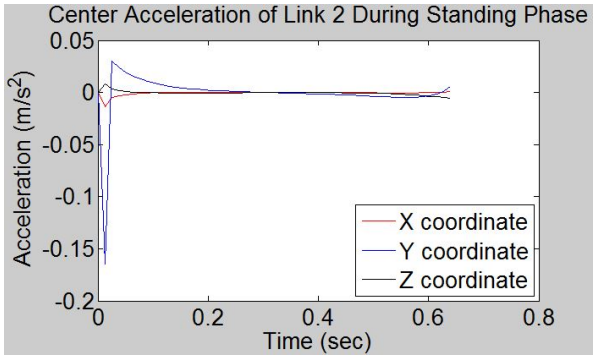

Figure 6.25: Center acceleration of link 2 during standing phase

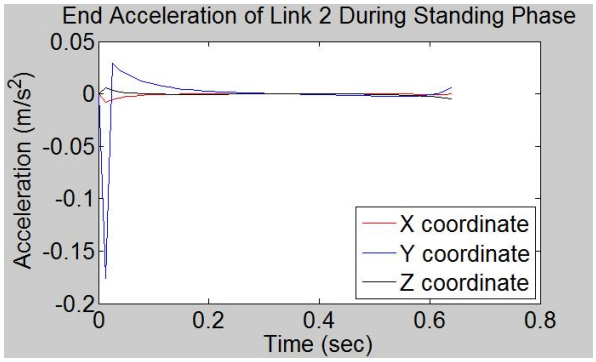

Figure 6.26: End acceleration of link 2 during standing phase

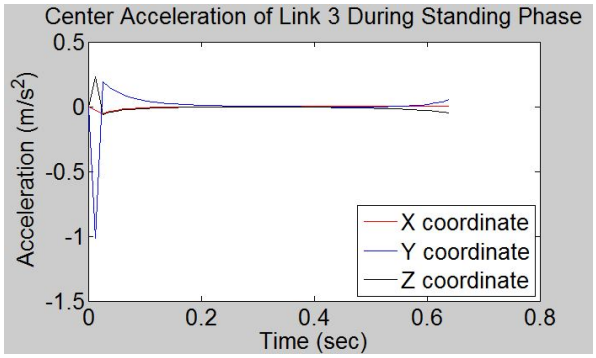

Figure 6.27: Center acceleration of link 3 during standing phase

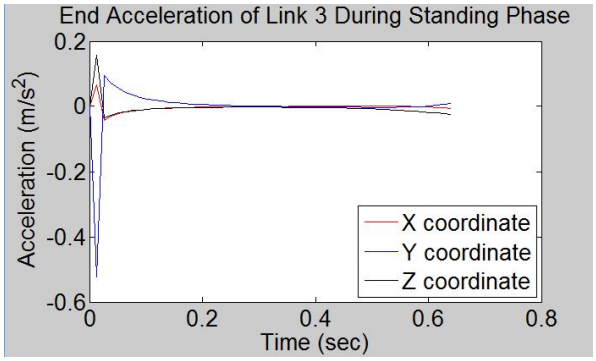

Figure 6.28: End acceleration of link 3 during standing phase

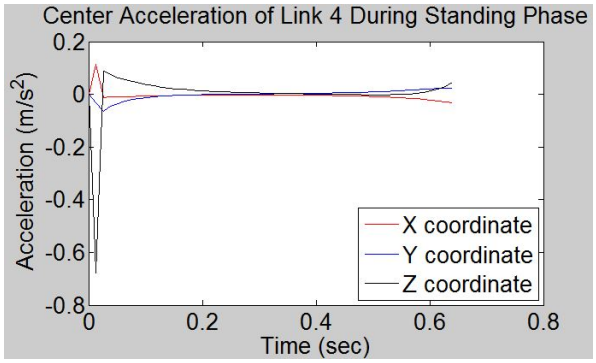

Figure 6.29: Center acceleration of link 4 during standing phase

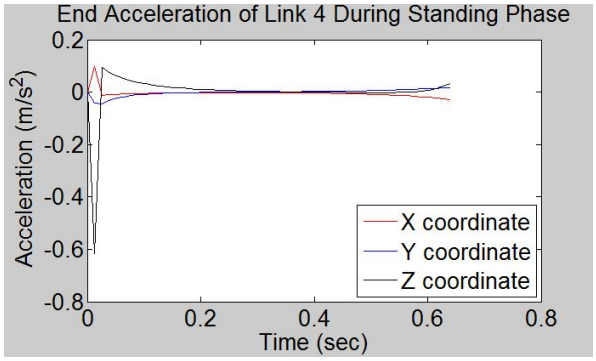

Figure 6.30: End acceleration of link 4 during standing phase

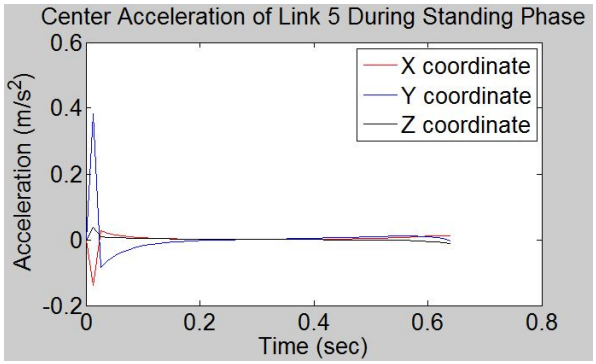

Figure 6.31: Center acceleration of link 5 during standing phase

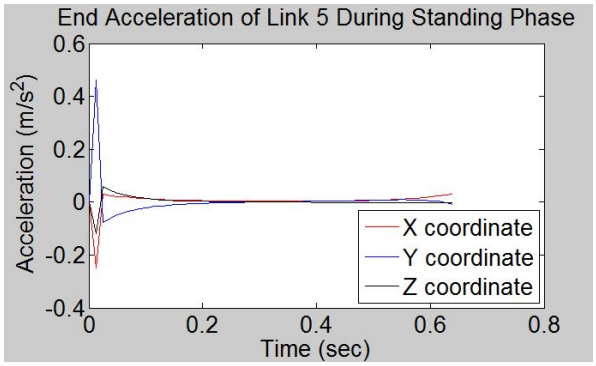

Figure 6.32: End acceleration of link 5 during standing phase

6.4.2 Swinging Phase

This next set of plots from figure 6.33 to 6.42 are the forces and torques of each link during the swinging phase. They have the same starting spike but then they follow a path that corresponds to the walking pattern generation. When each link performs a movement during the walking phase there should be either a decrease or a increase in the force and torques. Thats also applied to the center and end accelerations, seen in figures 6.43 to 6.52.

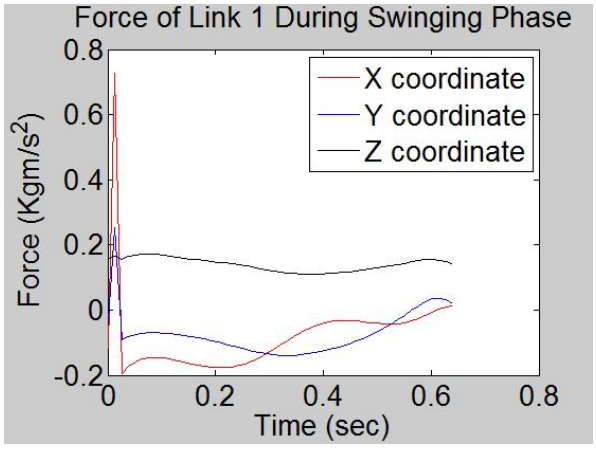

Figure 6.33: Force of link 1 during swinging phase

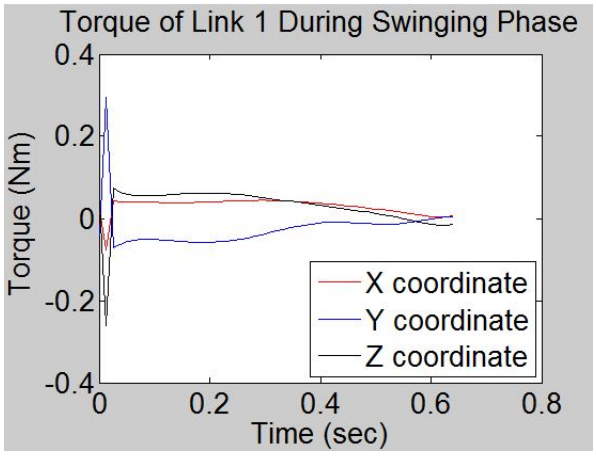

Figure 6.34: Torque of link 1 during swinging phase

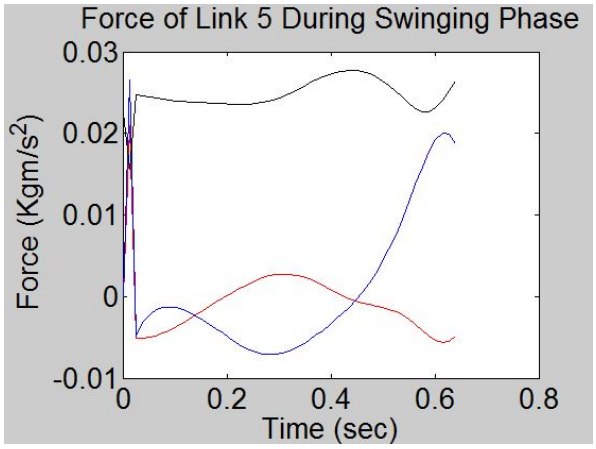

Figure 6.35: Force of link 5 during swinging phase

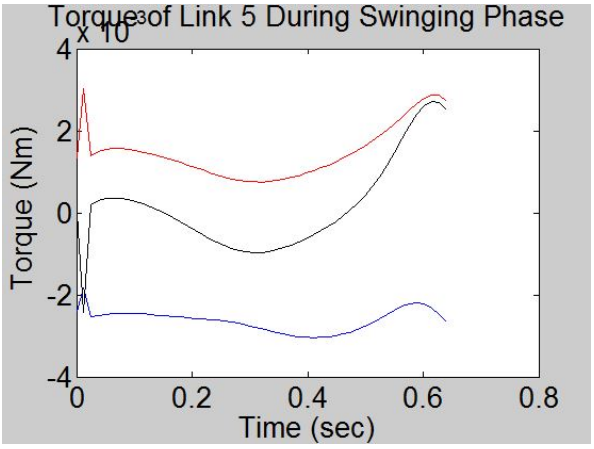

Figure 6.36: Torque of link 5 during swinging phase

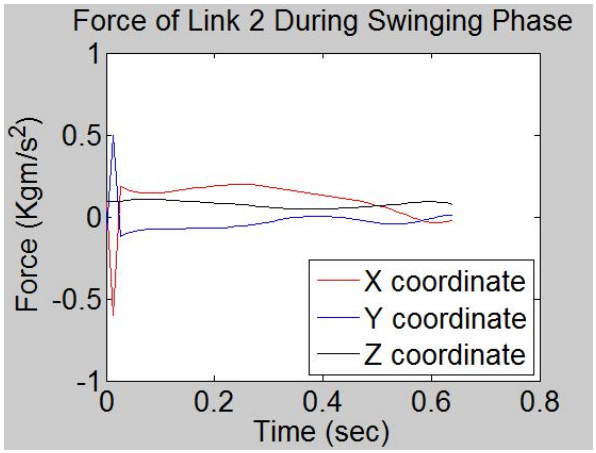

Figure 6.37: Force of link 2 during swinging phase

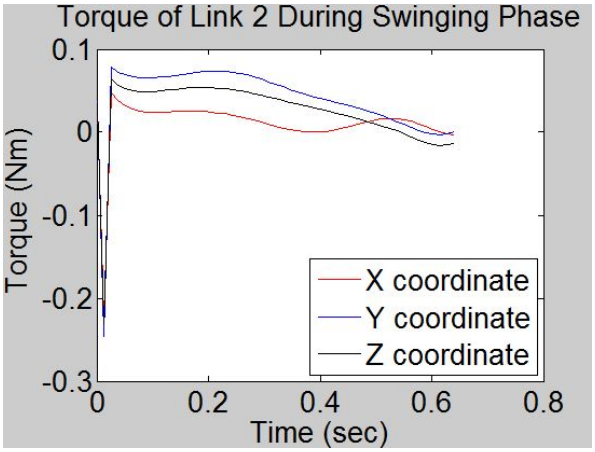

Figure 6.38: Torque of link 2 during swinging phase

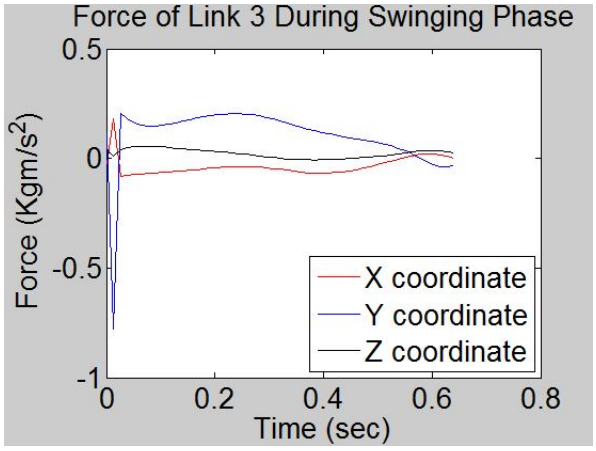

Figure 6.39: Force of link 3 during swinging phase

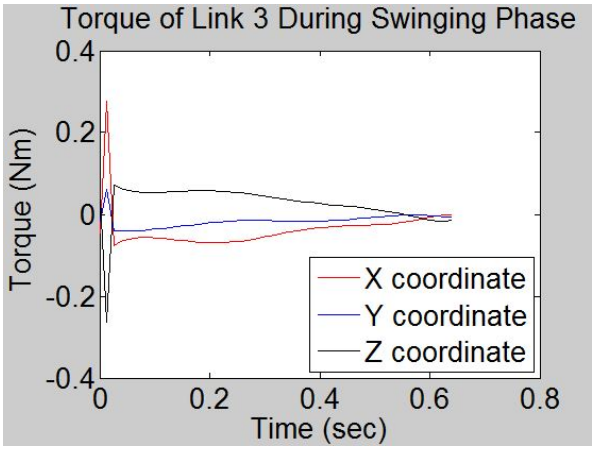

Figure 6.40: Torque of link 3 during swinging phase

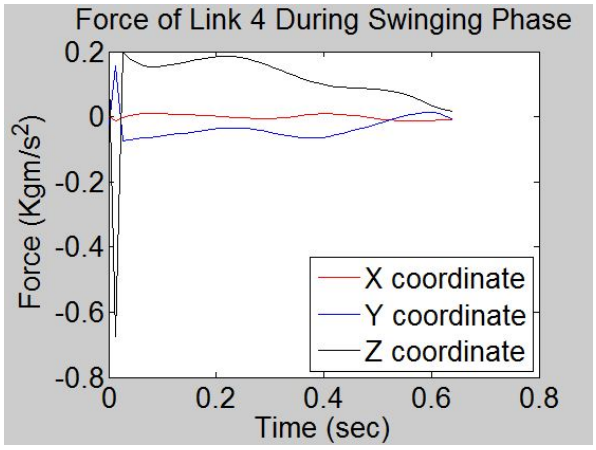

Figure 6.41: Force of link 4 during swinging phase

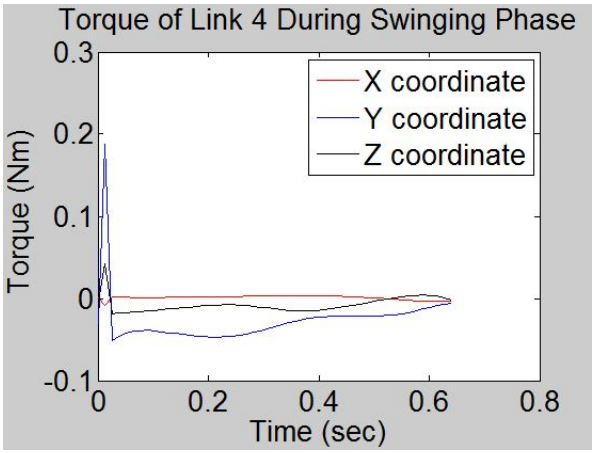

Figure 6.42: Torque of link 4 during swinging phase

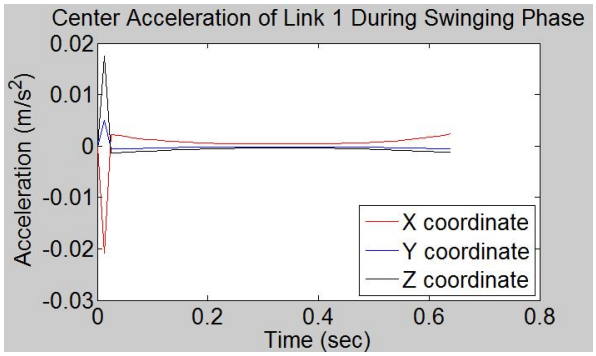

Figure 6.43: Center acceleration of link 1 during swinging phase

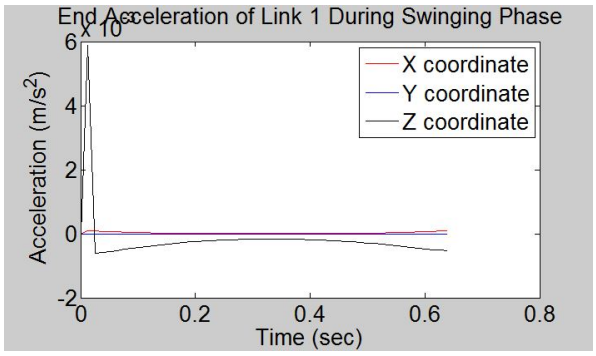

Figure 6.44: End acceleration of link 1 during swinging phase

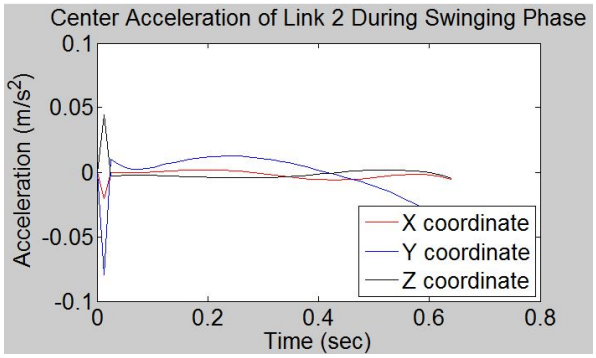

Figure 6.45: Center acceleration of link 2 during swinging phase

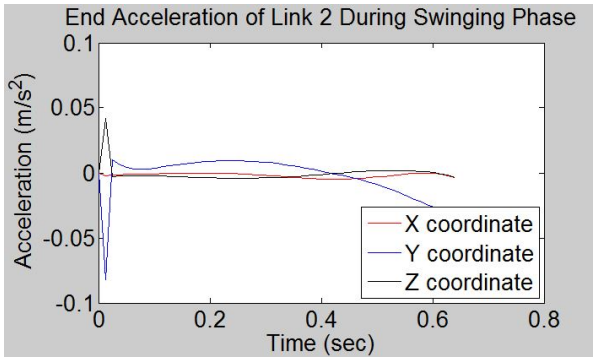

Figure 6.46: End acceleration of link 2 during swinging phase

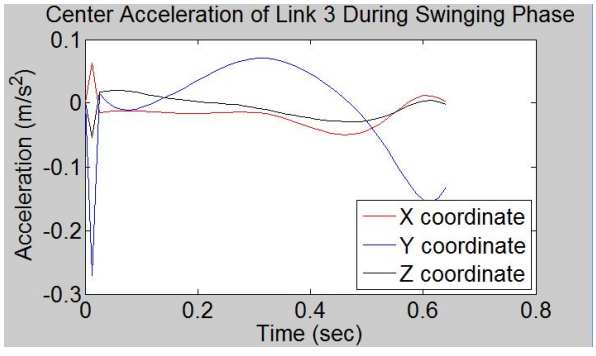

Figure 6.47: Center acceleration of link 3 during swinging phase

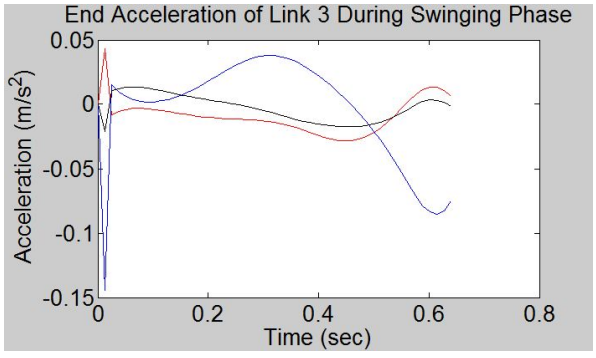

Figure 6.48: End acceleration of link 3 during swinging phase

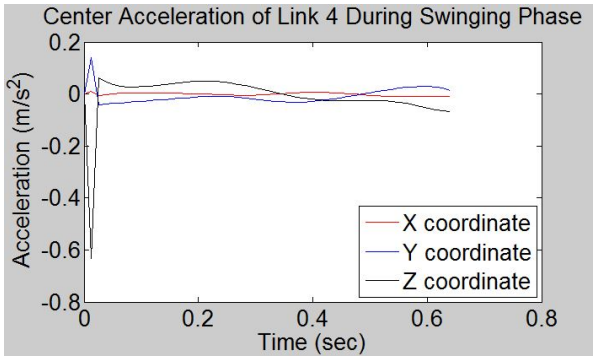

Figure 6.49: Center acceleration of link 4 during swinging phase

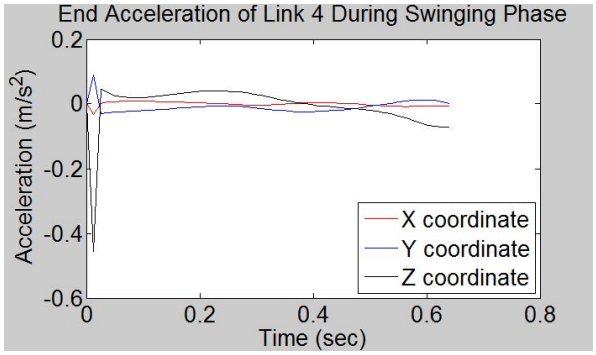

Figure 6.50: End acceleration of link 4 during swinging phase

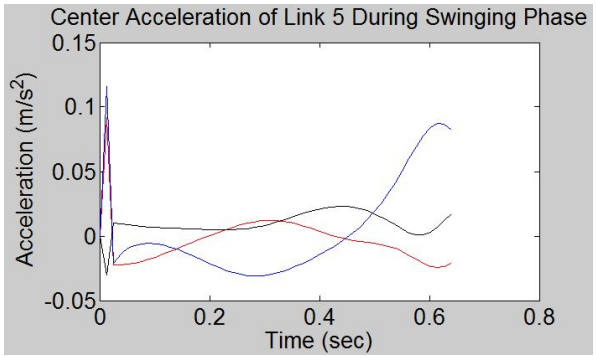

Figure 6.51: Center acceleration of link 5 during swinging phase

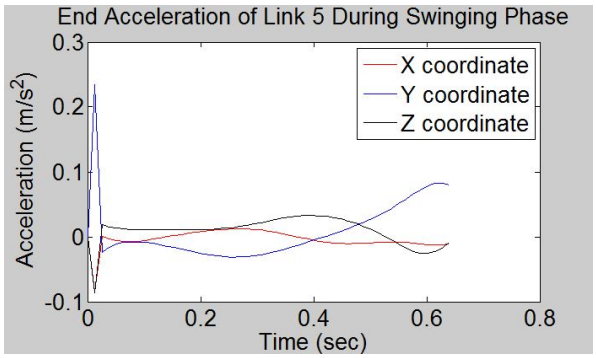

Figure 6.52: End acceleration of link 5 during swinging phase

6.5 Center of Mass

The center of mass is calculated and simulated in the process, seen in the flowchart below. Using the transformation matrices, link masses and lengths and the rotation matrices, the center of mass can be calculated.

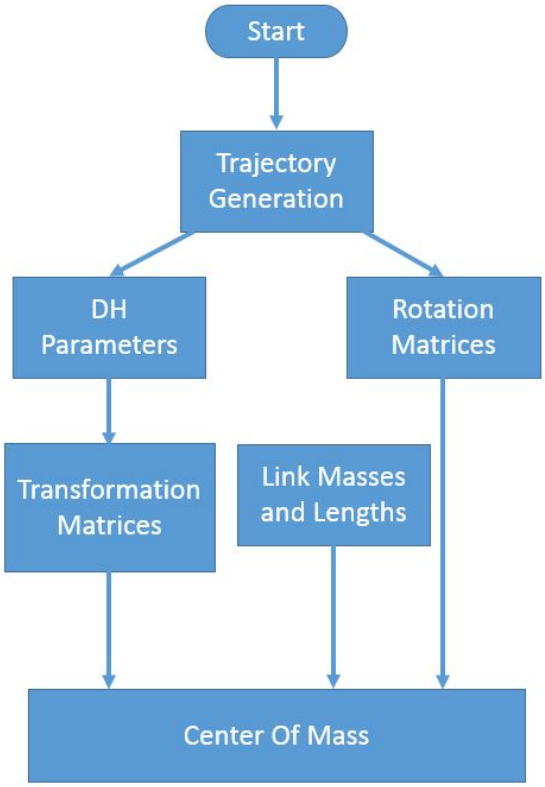

Figure 6.53: Center of mass block diagram.

The following two plots are the center of mass during the standing and walking phase. The center of mass is simply the projection of the x and y components on the ground. The x component is the horizontal axis and the y component is the vertical axis. The ideal situation would occur when the center of mass stays around the $(0, 0)$ point. This would help ensure the balance of the robot.

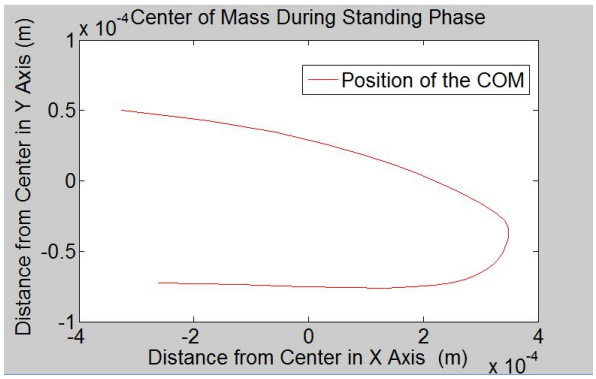

Figure 6.54: Center of mass position during the standing phase

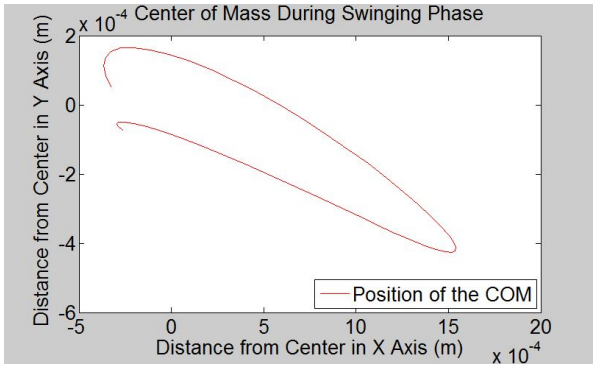

Figure 6.55: Center of mass position during the swinging phase

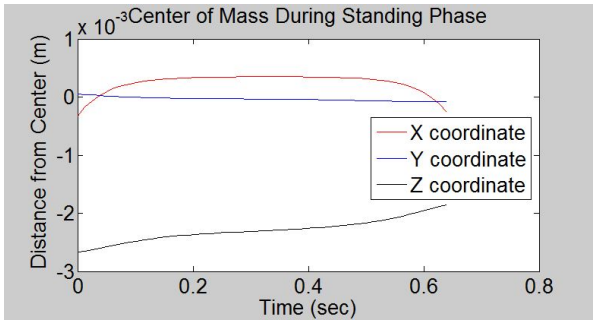

Figure 6.56: Center of mass during the standing phase

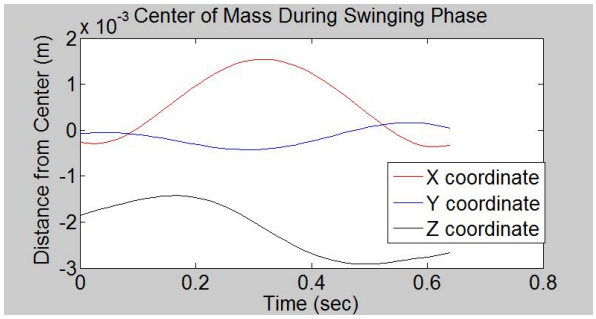

Figure 6.57: Center of mass during the swinging phase

6.6 Zero Moment Point

The zero moment point is determined by taking into account the center of mass as well as the angular velocity and acceleration and the center of link accelerations. The following plots show the position of the zero moment point. x is the horizontal axis and y is the vertical axis. The zero moment point acts similar to the center of mass. Ideally, it would stay around the center point $(0, 0)$. The x, y plot is the projection of the zero moment point onto the ground. You can see from figures 6.61 and 6.62 that the z coordinate is 0 because its flat on the ground.

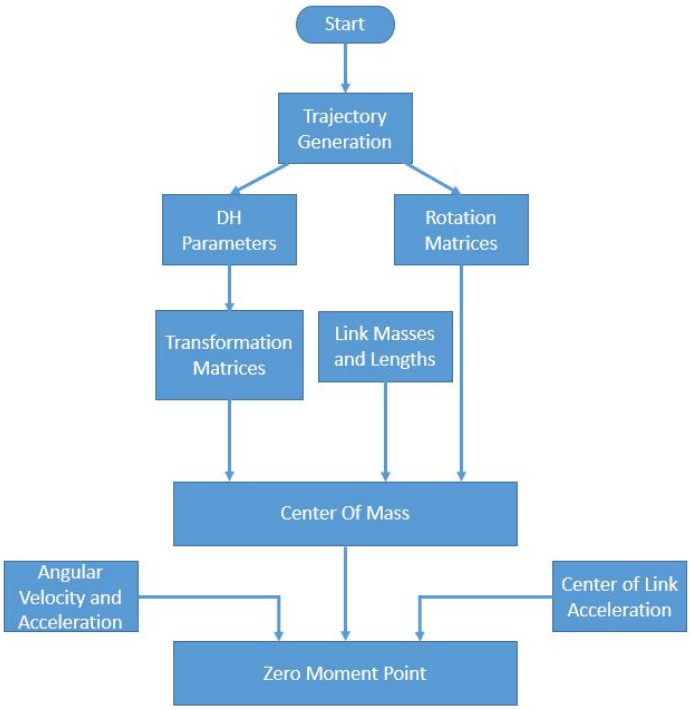

Figure 6.58: Zero moment point block diagram.

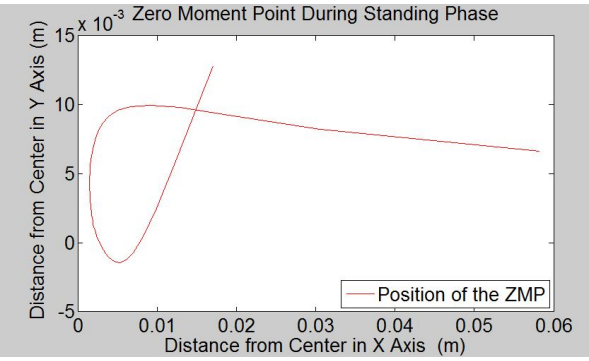

Figure 6.59: Zero moment point position during the standing phase

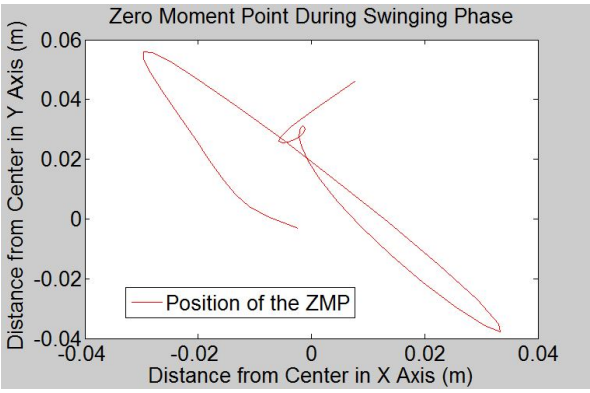

Figure 6.60: Zero moment point position during the swinging phase

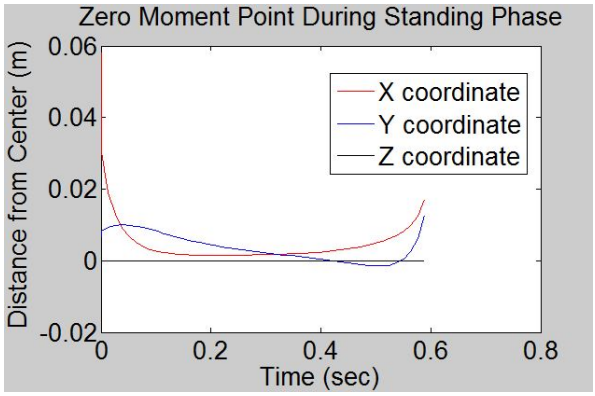

Figure 6.61: Zero moment point during the standing phase

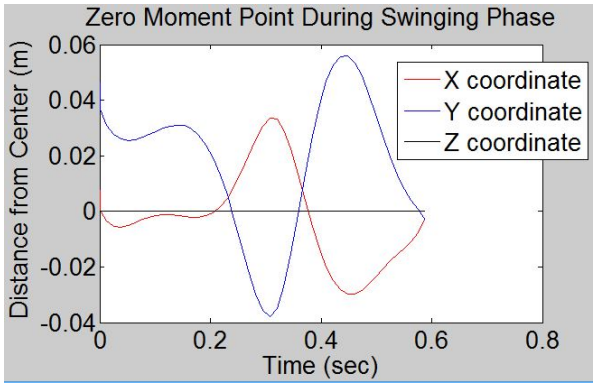

Figure 6.62: Zero moment point during the swinging phase

Chapter 7

Experimental Results

7.1 Introduction

The experimental tests were conducted on the 5 degree of freedom robot leg. Four separate tests were done: setpoint, sine wave, with and without force feedback control. The results and gains are all listed below, with the gains for the three types of controllers that were used. The parameters for the filter and sampling time used, are listed in subsection 5.2.1.

7.2 Experimental Model

The experimental model will be used to design the balance controller around the center of pressure. The structure is similar to the simulation model,with different characteristics. Prototype one had five degrees of freedom, tall, heavy and drew too much current. Prototype Two is the same as Prototype One but much smaller and lighter so current wouldn't be an issue. This can be seen in the figure below.

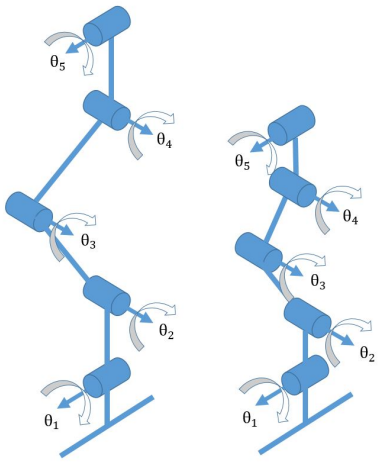

Figure 7.1: Prototype One on left and Prototype Two on right.

7.3 Setpoint Control

The first test was to set each joint one by one to a specific angle and control them using a P, PD and PID controller. All the tunning was done by trial and error. First a P controller was tunned then a PD then a PID. The use of a dynamic equation wasn't possible as inertias and masses were not known.

Motor Number	K_p	K_p,K_d	K_p,K_d,K_i
1	0.3	0.3,0.001	0.3,0.01,0.00005
2	0.5	0.5,0.00001	0.53,0.00001,0.00005
3	0.45	0.45,0.000013	0.505,0.000013,0.0000125
4	0.95	0.95,0.00001	0.95,0.00001,0.000029
5	0.35	0.35,0.0001	0.35,0.0001,0.00005

Table 7.1: Gains for setpoint control for the three controllers

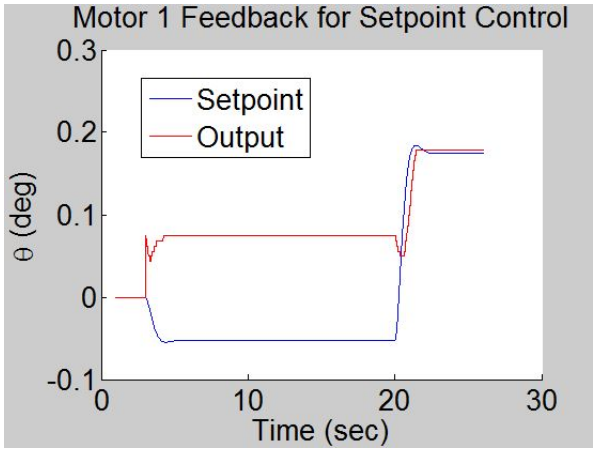

Figure 7.2: Motor 1 feedback during setpoint control

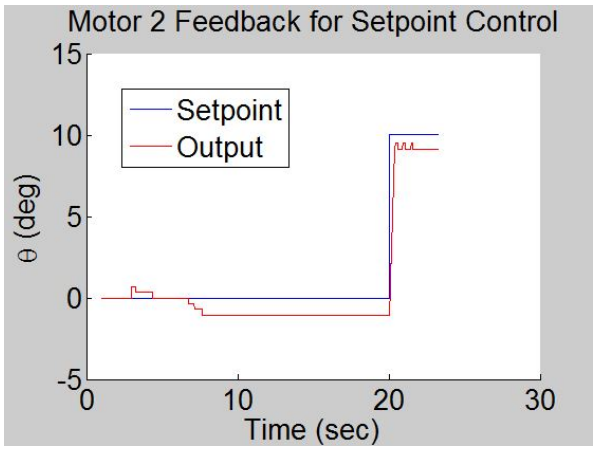

Figure 7.3: Motor 2 feedback during setpoint control

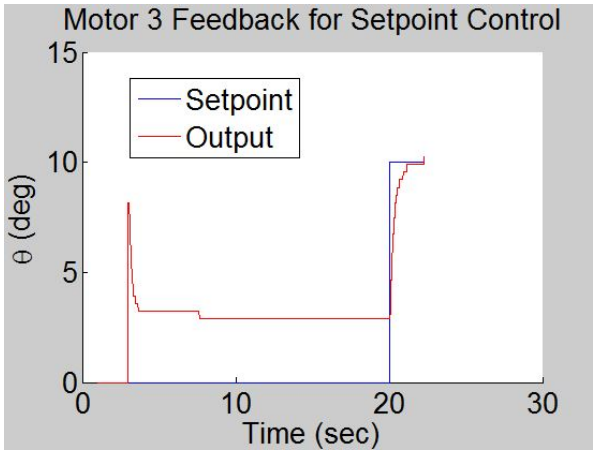

Figure 7.4: Motor 3 feedback during setpoint control

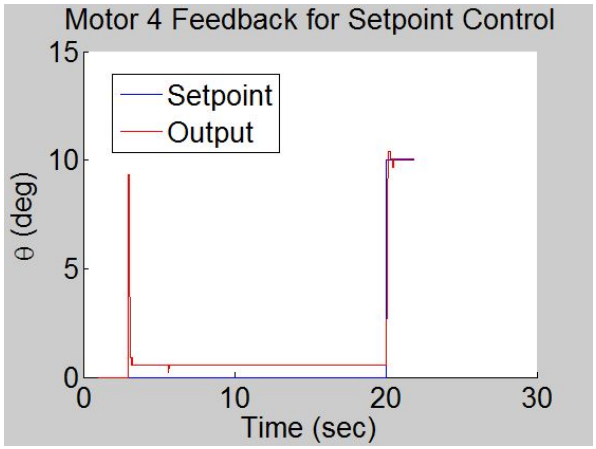

Figure 7.5: Motor 4 feedback during setpoint control

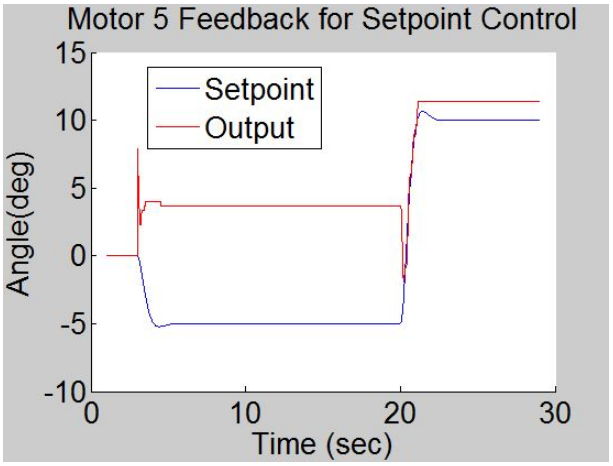

Figure 7.6: Motor 5 feedback during setpoint control

Figures 7.2 through 7.6 show the feedback for the setpoint control. Each joint was set one by one to 10 degrees and then tuned. The results shown are with a PID controller, with the gains specified in table 7.1. The first 20 seconds are used to set all the joints at 0 degrees then after that they will be controlled to the setpoint of 10 degrees. All the plots show the setpoint and the output. The horizontal axis is the time, in seconds, and the vertical axis is the joint angle, in degrees.

Upon analysis of the motors, Motors 2, 3 and 4 were found to have outputs close to the setpoint while Motors 1 and 5 were off by a couple degrees. This is due to the weight of the system and the mass attached to the hip. When the ankle motor (Motor 1, figure 7.2) is leaning to the left the mass of the remaining links and motors will be leaning to the same side, causing it to be harder for the motor to stop at the correct setpoint. The gravity and extra weight will push it farther. Its the same thing for Motor 5. There is a weight attached to the 5th link, so it simulates the weight of another leg attached to it. The gravity will push this heavy weight down causing an error in the set point. This can be corrected by adding a compensation term the controller (5.6). Then the controller becomes:

$$u(n) = K_p e[n] + K_i \sum_{k=0}^{n} e[n-k] + k_d \frac{e[n]-e[n-1]}{T_s} + Compensation \tag{7.1}$$

The compensation is shown and discussed in the next section.

7.4 Sine Wave Control

The second test was to apply a sin wave to each joint and control them using a P, PD and PID controller.

Motor Number	K_p	K_p,K_d	K_p,K_d,K_i	Compensation
1	0.744	0.744,0.00012	0.744,0.00012,0.00005	-0.12
2	0.51	0.51,0.000014	0.51,0.000014,0.000055	0
3	0.7	0.7,0.000013	0.7,0.000013,0.000024	0
4	0.95	0.95,0.00001	0.95,0.00001,0.000035	0
5	0.55	0.55,0.01	0.55,0.01,0.00007	0.15

Table 7.2: Gains for sine wave control for the three controllers.

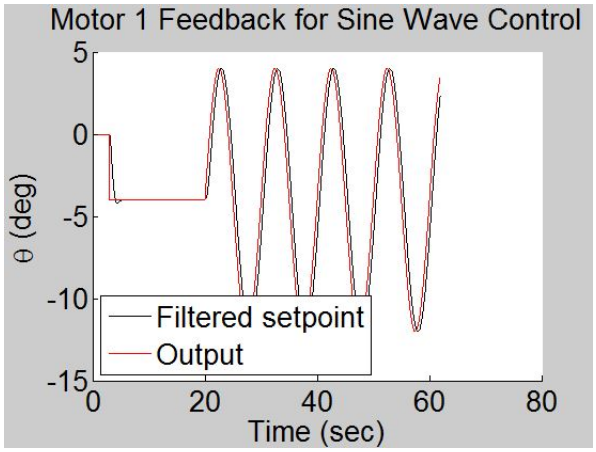

Figure 7.7: Motor 1 feedback during sine wave control

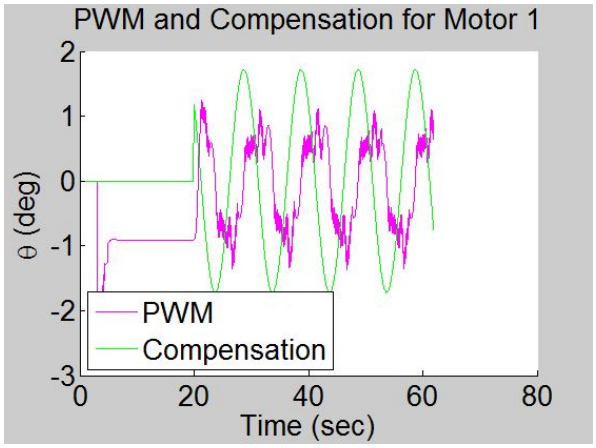

Figure 7.8: Motor 1 PWM and compensation

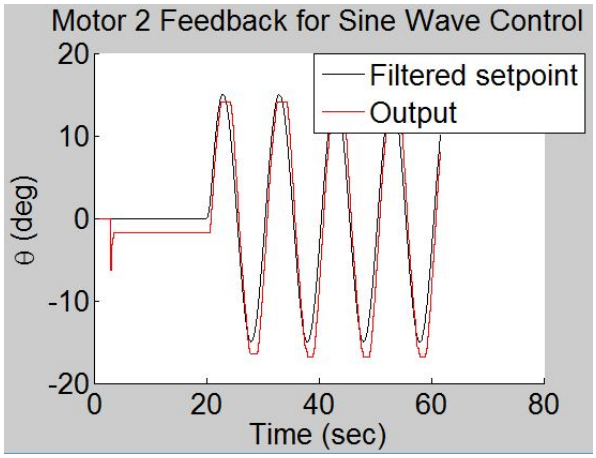

Figure 7.9: Motor 2 feedback during sine wave control

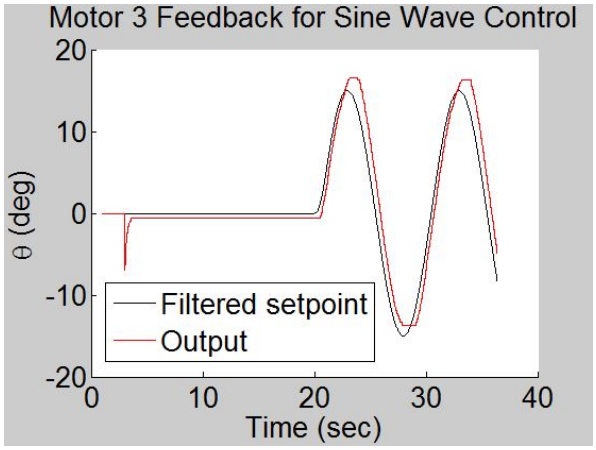

Figure 7.10: Motor 3 feedback during sine wave control

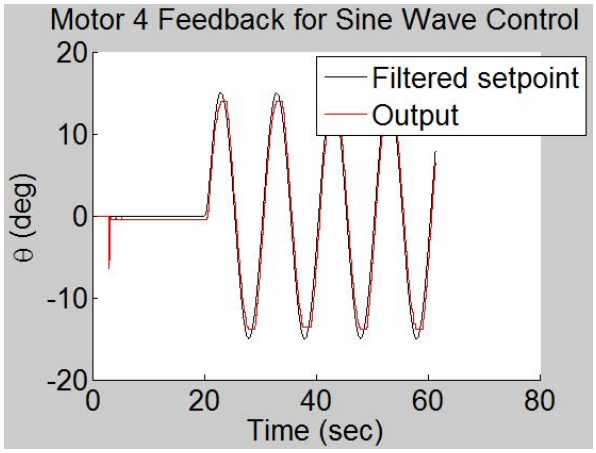

Figure 7.11: Motor 4 feedback during sine wave control

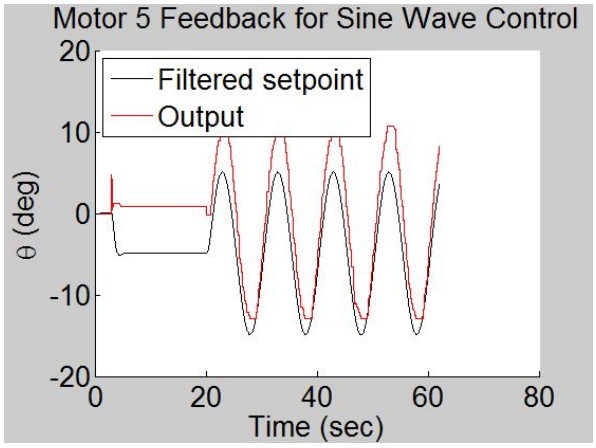

Figure 7.12: Motor 5 feedback during sine wave control

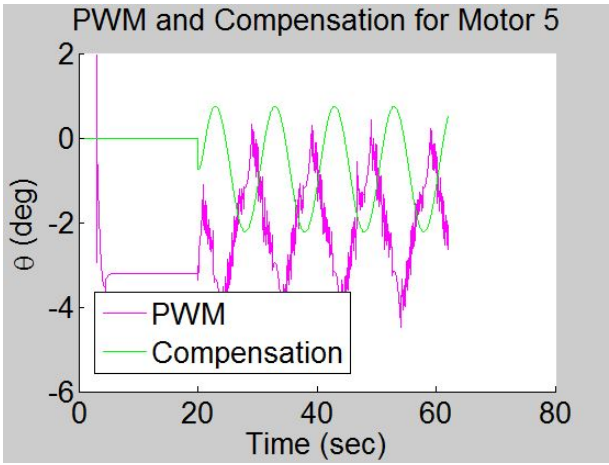

Figure 7.13: Motor 5 PWM and compensation

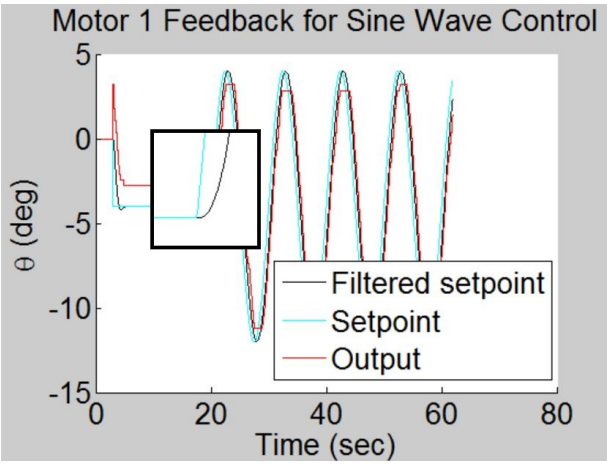

Figure 7.14: Setpoint compared to filtered setpoint

Each joint was controlled one by one, to follow a sinusoidal curve. Looking at Motors 2, 3 and 4, the plots show the setpoint, filtered setpoint and output. The filtered setpoint was introduced so it would be a gradual increase for the motor, rather than a sharp output. In figure 7.14, you can see the filtered setpoint in black and the unfiltered setpoint in cyan. The unfiltered setpoint will have the sharp edge and the filtered setpoint will have a smooth rising edge.

In figures 7.8 and 7.13 compensation and the PWM signal for the motors are shown. The PWM is from the output of the PID controller, $U(s)$. The compensation term is introduced, to compensate for the gravity that is acting on the mass of the leg and causing the error shown on the two motors. Observations of Motor 1 show that when the leg shifts to the left, gravity will

act on the mass of the system, causing the leg to go past the setpoint seen in figure 7.2. The compensation term will remove that error. The compensation will be added to the PWM so it will shift down the output to match the setpoint. The compensation is opposite compared to the PWM. This is one way of compensating for the gravity force thats shifting the leg down.

Motor 5 (figure 7.12) has a mass attached to the end of the link to demonstrate a mass of another leg. So, when the motor is at its zero position the mass will start to turn the joint and fall down due to gravity. The compensation term is applied here to help get rid of this. In the plot, overshoot occurs when the mass is turning to the opposite side, but this does not affect the center of pressure in a negative way. The mass will be lined up with the rest of the leg pushing the center of pressure to the (0,0) point. The bottom half of the graph represents the right side of the leg, if the mass causes the motor to extend beyond its setpoint the joints limit will be reached. Due to the compensation term the output is matched with the filtered setpoint.

7.5 Force Feedback Control

The third experiment was to apply the force feedback to Joints 1 and 2 and control them using a PID controller. The first joint's setpoint will use the $Side2side$ calculation while joint 2 will be use the $Front2back$ calculation from the center of pressure. This was discussed earlier in the subsection for balance control. Four separate tests were performed, one with disturbance on Motor 5, then Motor 4 , Motor 3 and then Motor 3 and 5. The results are not shown for the motors that aren't moving, since they were set to zero for the experiment.

7.5.1 Test 1

Test 1 was conducted by setting the setpoint for Motor 1 to the $Side2side$ calculation. Motor 2 was set to the $Front2back$ calculation. Motor 3 and 4 were set at their zero position. Motor 5 had the disturbance applied to it. A sine wave was used to make the mass of Motor 5 go up and down. This would simulate the mass of another leg, if it was attached. Motor 5 is in the lateral coordinate plane, so Motor 2 will be the most important motor, for compensating this moving mass.

Motor Number	K_p	K_d	K_i	Compensation Gain
1	0.5	0.02,	0.001	0.005
2	0.35	0.01	0.007	-0.1
3	0.9	0.013	0.000024	0
4	1.0	0.01	0.000035	0
5	0.55	0.01	0.0007	0

Table 7.3: Gains for force feedback control with disturbance on Motor 5

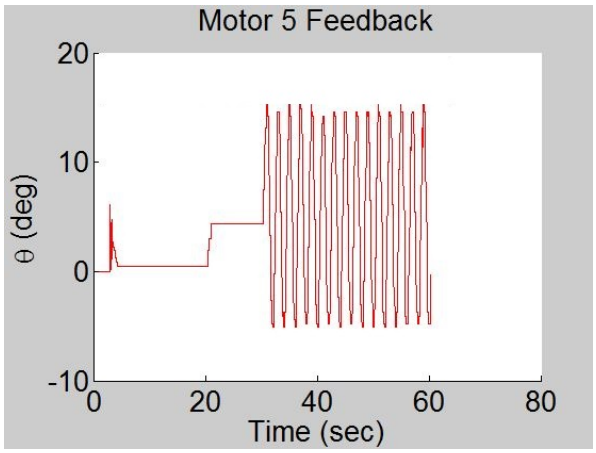

Figure 7.15: Motor 5 with sine wave disturbance

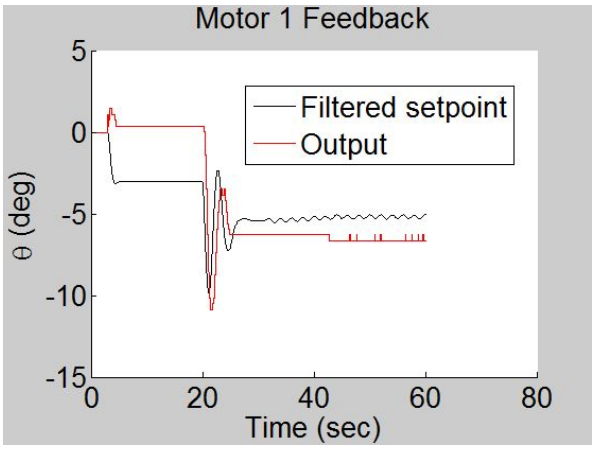

Figure 7.16: Motor 1 feedback during Motor 5 disturbance

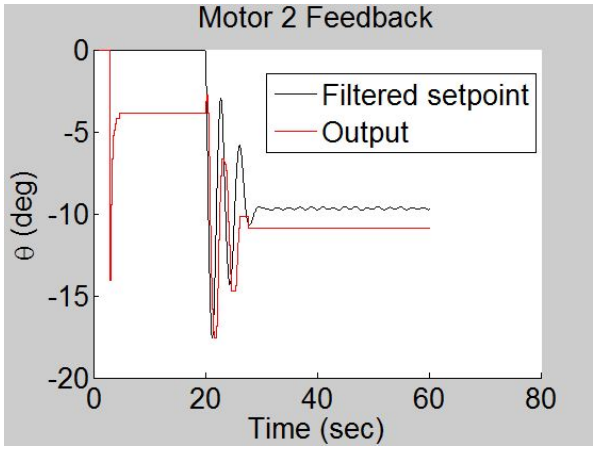

Figure 7.17: Motor 2 feedback during Motor 5 disturbance

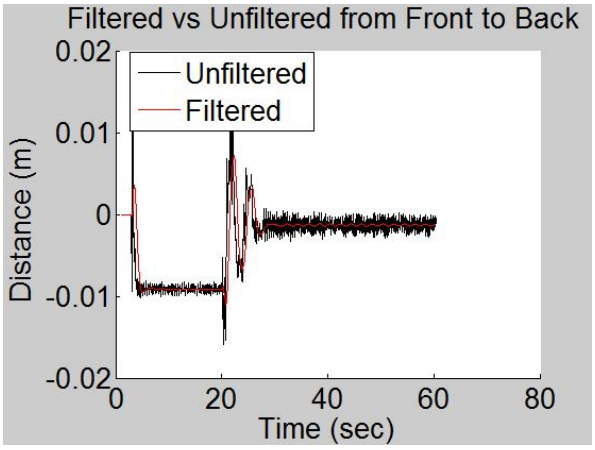

Figure 7.18: Center of pressure from front to back

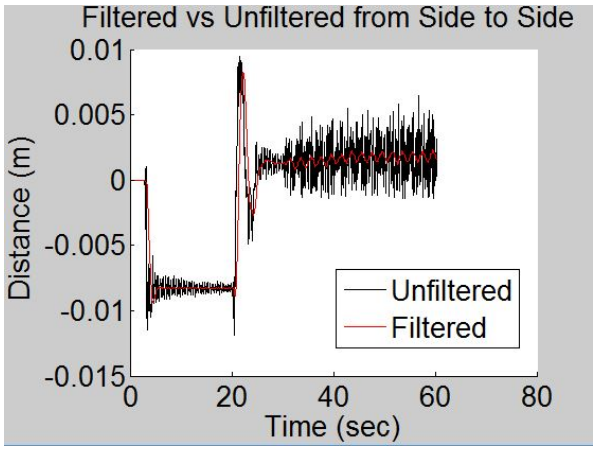

Figure 7.19: Center of pressure from side to side

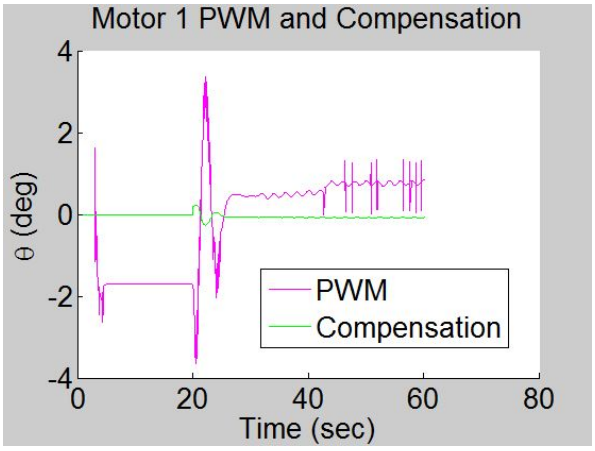

Figure 7.20: PWM and compensation for Motor 1

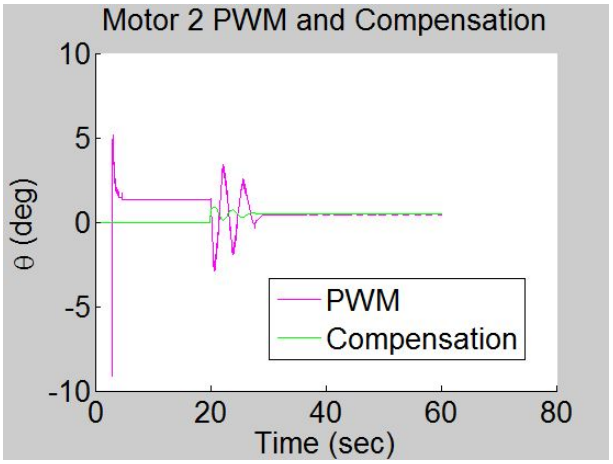

Figure 7.21: PWM and compensation for Motor 2

During the first 20 seconds of the test, all the motors will go to the specified zero position. After 20 seconds, there is 10 seconds for the leg to position itself around the center of pressure. So Motor 1 will move the leg side to side and Motor 2 will move the leg front to back. Once a total of 30 seconds have passed a sine wave is applied to Motor 5, to act as a disturbance. This motor will move the mass up and down, moving the center of pressure around. Once the COP is moved, the controller will have to move Motor 1 and 2, to correct this error.

Looking at the figures 7.16 and 7.17 you can see the output (red) follows the filtered setpoint (black). There is a steady state error around 1 degree. The important part is that the center of

pressure is around zero, so that the leg will stay balanced. The center of pressure for front to back and side to side are shown in figures 7.18 and 7.19.

One difference between the force feedback control and the sine wave control is that there is a compensation term added to Motor 2. This is because the total mass is shifted to one side, so gravity is acting upon it, so it's harder for the controller to move it from side to side.

7.5.2 Test 2

Test 2 is run the same as test 1 but this time Motor 5 is set to its zero position and Motor 4 has a disturbance at 30 seconds.

Motor Number	K_p	K_d	K_i	Compensation
1	0.5	0.02,	0.001	0.005
2	0.35	0.01	0.007	-0.1
3	0.9	0.013	0.000024	0
4	1.0	0.01	0.000035	0
5	0.3	0.01	0.0007	0

Table 7.4: Gains for force feedback control with disturbance on Motor 4.

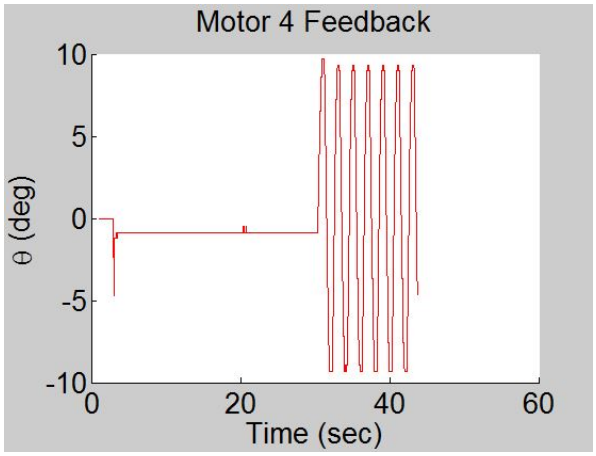

Figure 7.22: Motor 4 with sine wave disturbance.

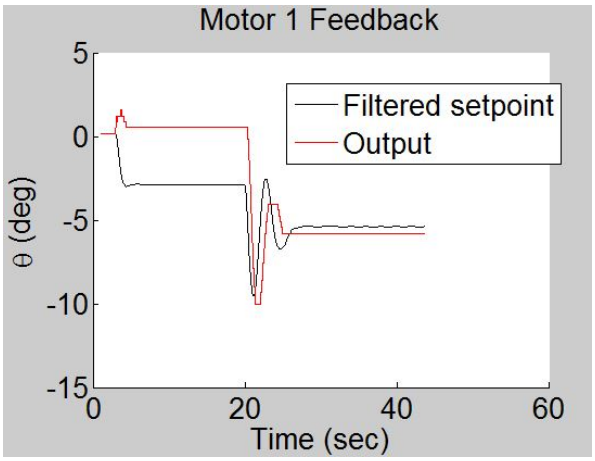

Figure 7.23: Motor 1 feedback during Motor 4 disturbance

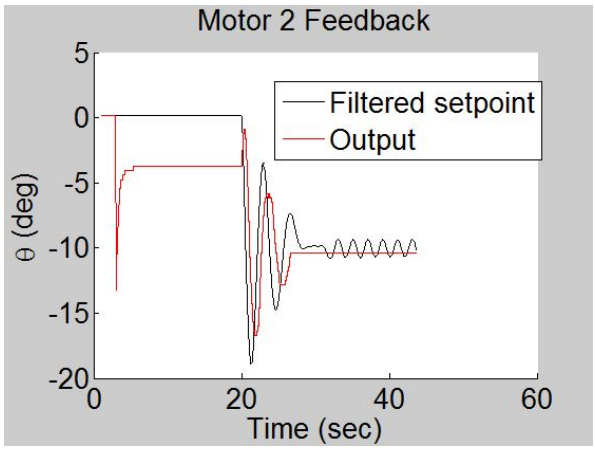

Figure 7.24: Motor 2 feedback during Motor 4 disturbance

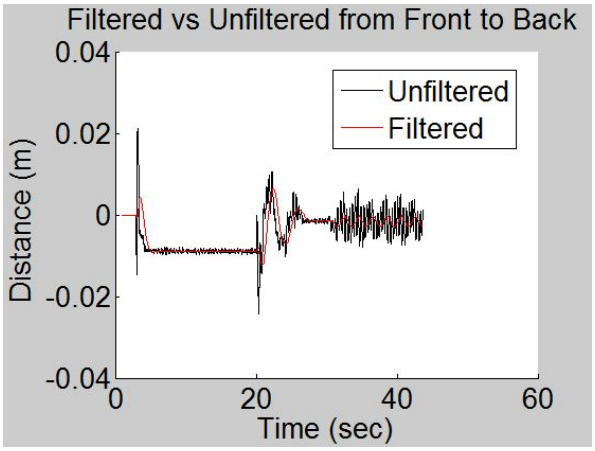

Figure 7.25: Center of pressure from front to back

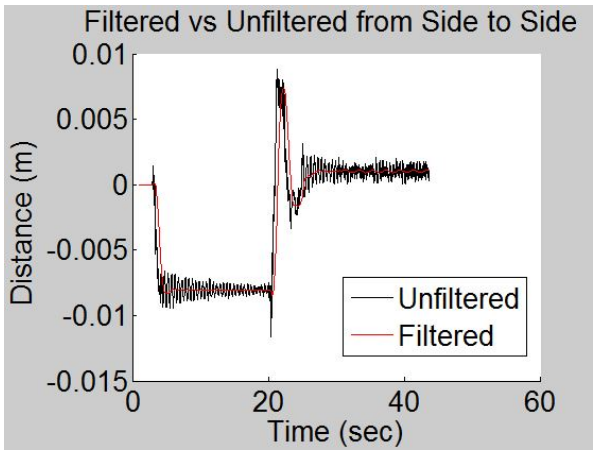

Figure 7.26: Center of pressure from side to side

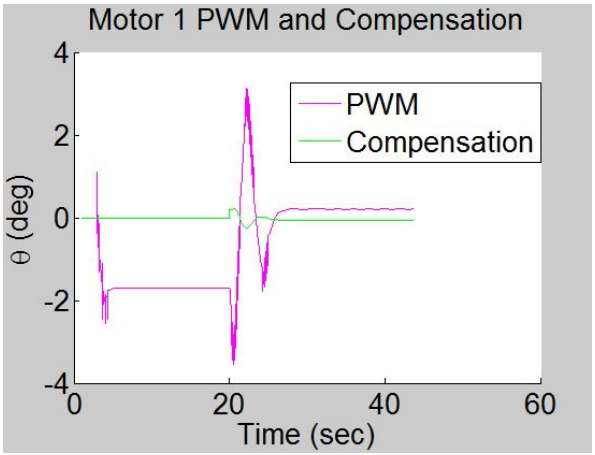

Figure 7.27: Motor 1 PWM and compensation

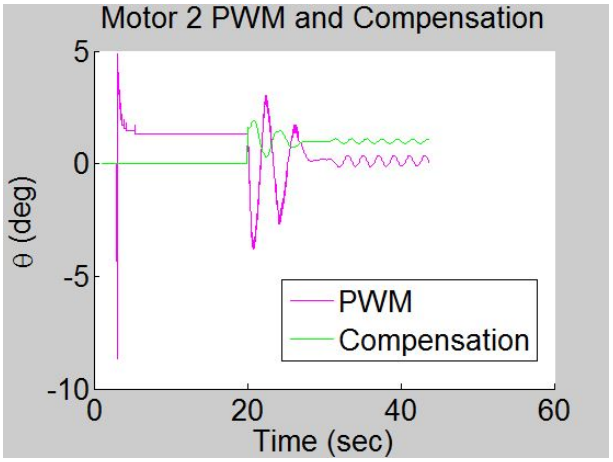

Figure 7.28: Motor 2 PWM and compensation

The results show that Motors 1 and 2 follow the correct path and help change the COP so that the leg stays balanced. Motor 1 has a steady state error of less than 1 degree as well as Motor 2. The reason Motor 2's output isn't oscillating like the setpoint is because the error isn't big enough to generate a PWM to do so.

7.5.3 Test 3

Test 3 is the same as the last two but the disturbance is on Motor 3 while Motor 4 and 5 stay at their zero position. The gains for this test stayed the same as test 2.

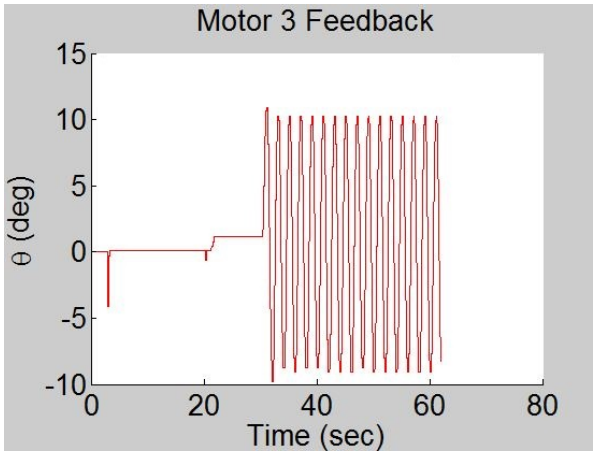

Figure 7.29: Motor 3 with sine wave disturbance

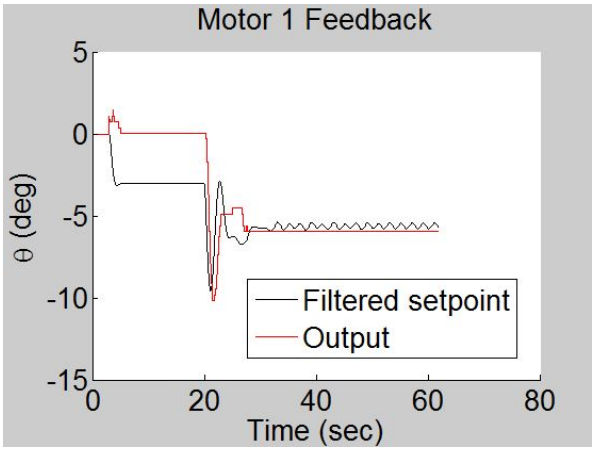

Figure 7.30: Motor 1 feedback during Motor 3 disturbance

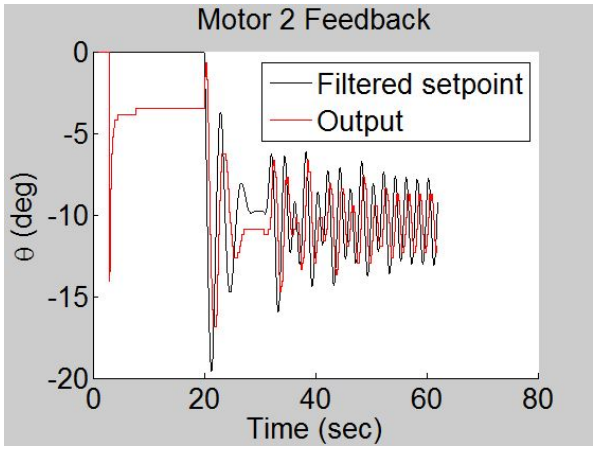

Figure 7.31: Motor 2 feedback during Motor 3 disturbance

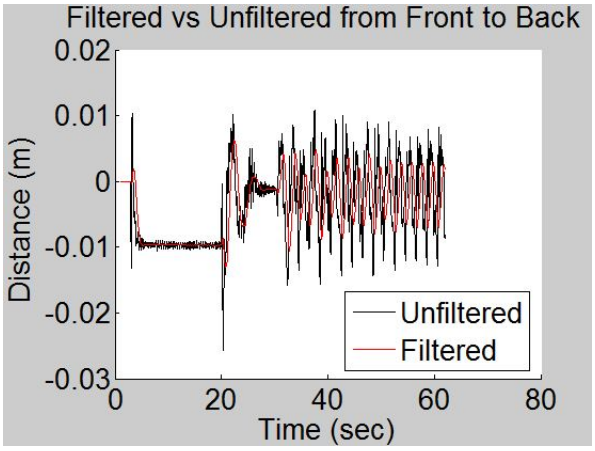

Figure 7.32: Center of pressure from front to back

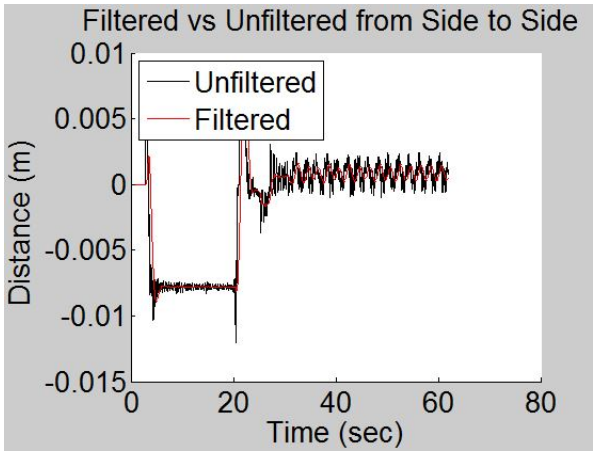

Figure 7.33: Center of pressure from side to side

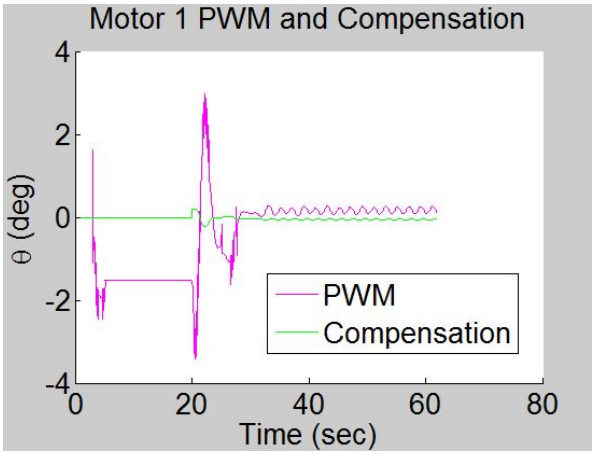

Figure 7.34: PWM and compensation for Motor 1

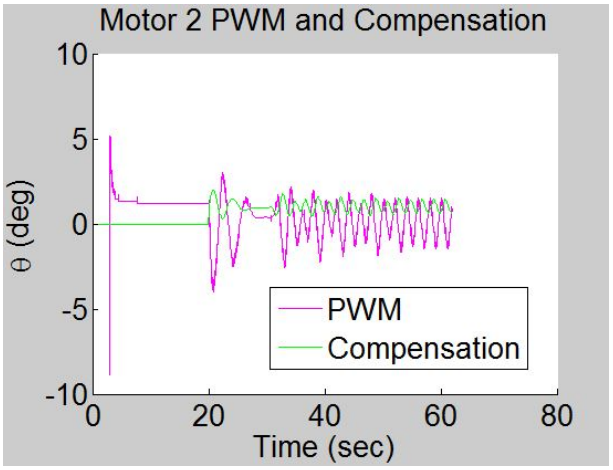

Figure 7.35: PWM and compensation for Motor 2

As seen in figure 7.29, a high frequency sine wave was applied to Motor 3 at 30 seconds. Since Motor 3 acts in the sagittal plane, Motor 2 was in charge of compensating this front to back change in COP. Following the plots for Motors 1 and 2, there is very little steady state error and the COP is controlled. The COP for front to back and side to side was controlled around the zero point, which helped the leg balance in the upright position.

7.5.4 Test 4

Lastly, test 4 is the same as previous tests but this time Motor 3 and 5 have a disturbance applied to them.

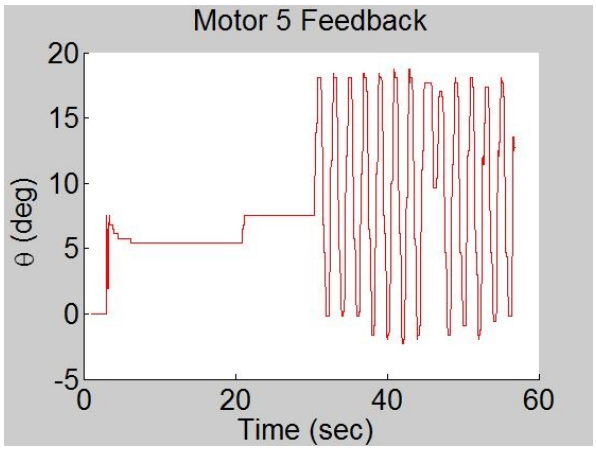

Figure 7.36: Motor 5 with sine wave disturbance

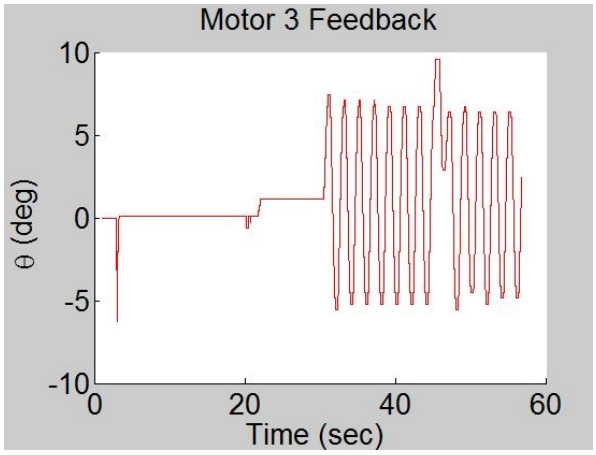

Figure 7.37: Motor 3 with sine wave disturbance

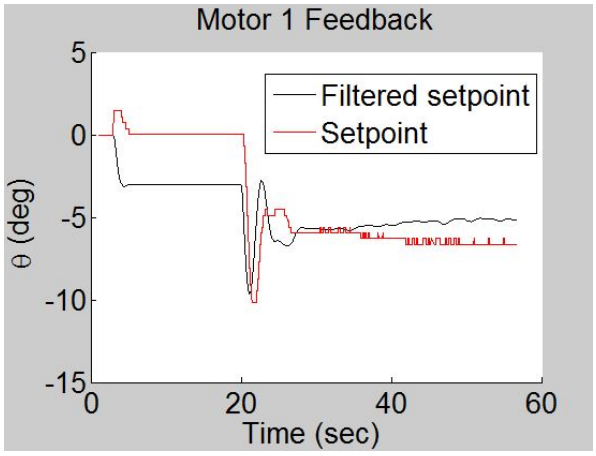

Figure 7.38: Motor 1 feedback

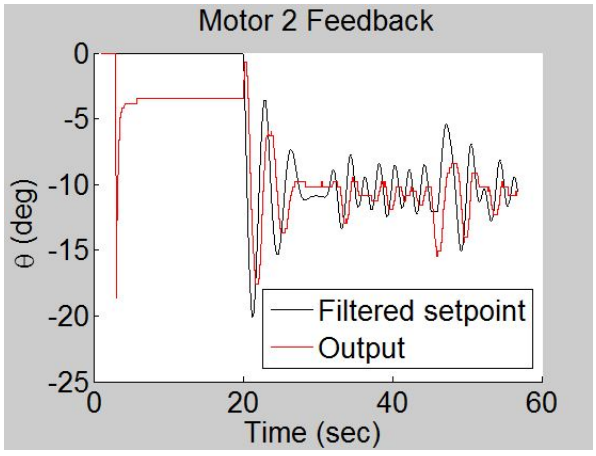

Figure 7.39: Motor 2 feedback

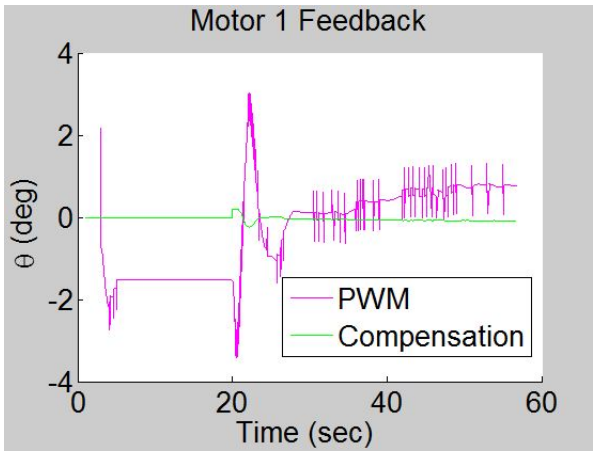

Figure 7.40: PWM and compensation for Motor 1

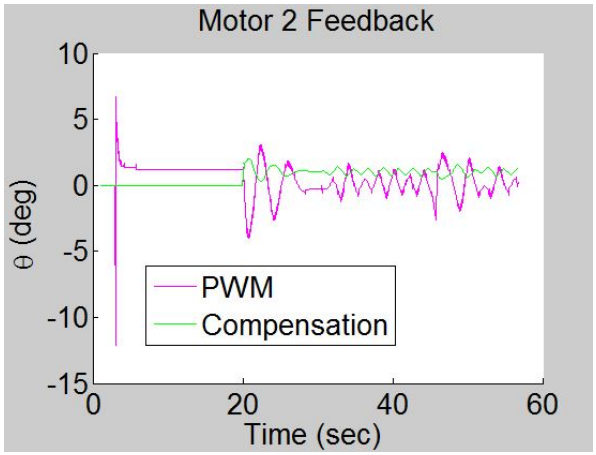

Figure 7.41: PWM and compensation for Motor 2

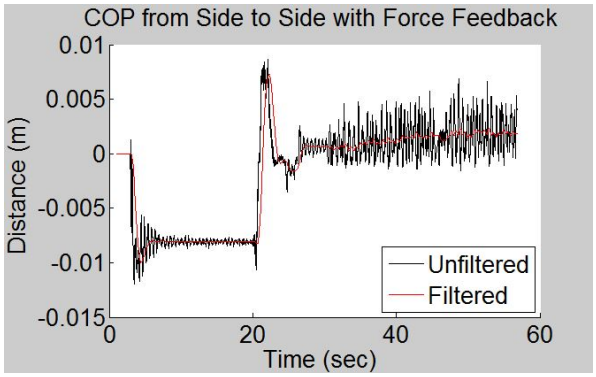

Figure 7.42: Center of pressure from side to side

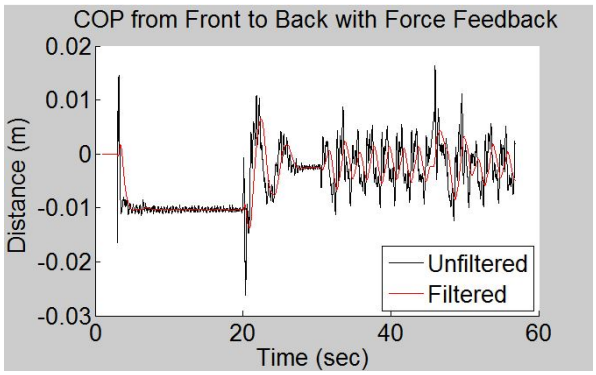

Figure 7.43: Center of pressure from front to back

Test 4 was performed to apply a disturbance in both the sagittal and lateral coordinate frame. Motor 5 and 3 both had a sine wave applied to them, seen in figures 7.37 and 7.36. Seen in Motor 1 feedback, are some spikes in the output. These spikes are from having a disturbance applied to two joints causing the leg to vibrate. The important part is that the COP is around the zero point, seen in figures 7.42 and 7.43.

7.6 No Force Feedback Control

The last experimental test performed involved measuring the COP without force feedback control. Motors 1, 2 and 4 were set to their zero position, while Motors 3 and 5 had the sine wave

disturbance. The reason for this test is to compare the COP with and without force feedback control.

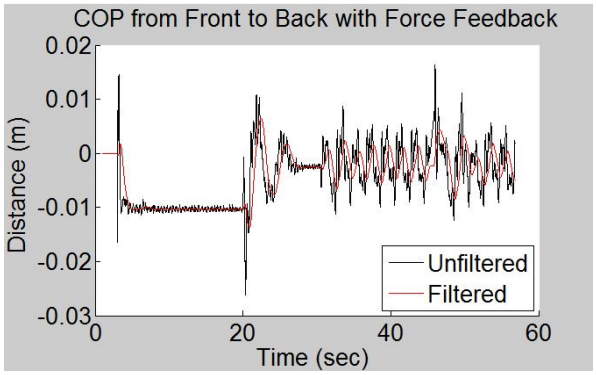

Figure 7.44: COP with force feedback

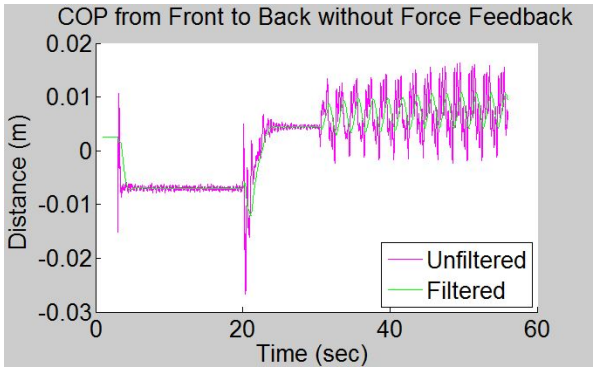

Figure 7.45: COP with no force feedback

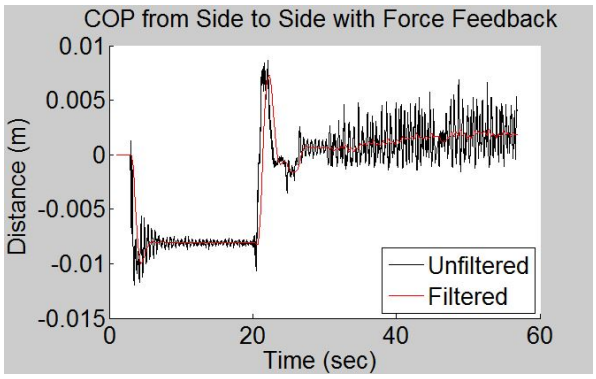

Figure 7.46: COP with force feedback

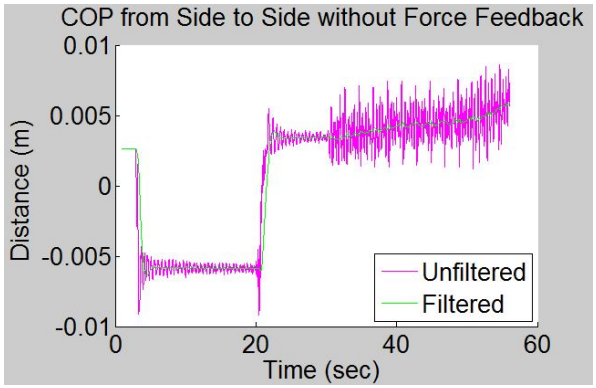

Figure 7.47: COP with no force feedback

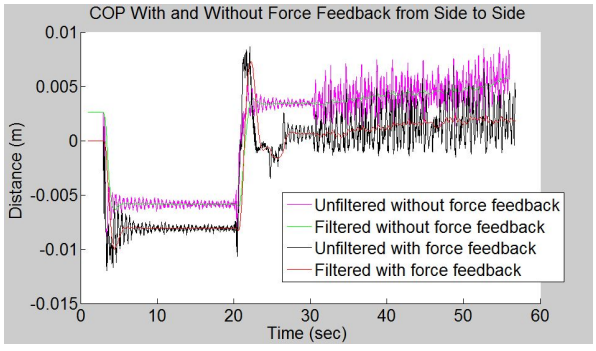

Figure 7.48: COP side to side comparison

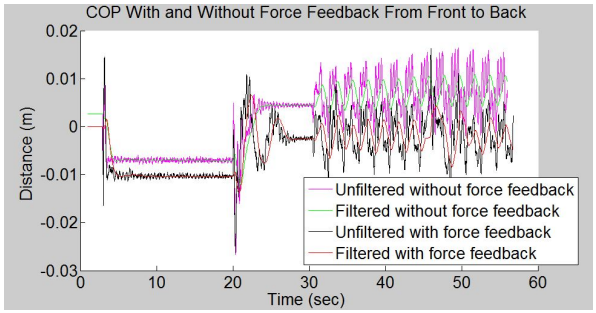

Figure 7.49: COP front to back comparison

There isn't that much change for the side to side COP because Motor 5 doesn't have that

big of an impact. The side to side motion shows, that there is a difference, between with and without force feedback. Without force feedback, the COP isn't controlled and the balance of the leg wont be in the upright position. Figures 7.48 and 7.49 show the comparison with and without force feedback on the same plot. Without the force feedback both front to back and side to side are about 0.5 cm off of the center.

Chapter 8

Thesis Summary and Future Work

8.1 Summary

The understanding of research and development of a five degree of freedom robot leg has been achieved in this thesis. From simulations to experiments, different types of theories were used and implemented in the research of robotic manipulators.

Simulations were first conducted to understand the basic theories and gain an understanding of the robotic field. Areas of interest were the DH parameters, walking generation, forward kinematics, Newton Euler recursion, center of mass and zero moment point.

The experimental robot had two designs. Prototype One was discovered to be useless for experimental purposes due to mechanical design errors and too high of a load current for the motors. Mistakes from Prototype One were learned from and Prototype Two was designed.

Prototype Two was built to run the setpoint control, sine wave control, and force feedback control. Control of the leg was achieved by using 3 microcontrollers. The controllers tested were, proportional (P), proportional derivative (PD) and proportional integral derivative (PID).

8.2 Achievements

During the course of the thesis, many achievements were made. The early stages included simulation design with results being obtained. The build of two prototypes were achieved. The first prototype, Prototype One, was finished then the second prototype, Prototype Two, was designed and built all by hand.

Many experimental tests were performed on Prototype Two. The design of the balance

controller was implemented, achieving success in the experimental testing.

8.3 Future Work

Future work includes the redesigning of the robot leg. Designing it from a CAD program lets you determine parameters like the inertia, link lengths and link masses that can be used in a dynamic equation describing the manipulator.

The next step would be to add on a second leg, so that a walking trajectory can be obtained. This will include the implementation of a new control scheme, with the support polygon being bigger. Switching to harmonic gears will also help reduce the backlash and potentially emliminate it.

Changing the controller would gain insight into which type of control technique would perform better. Some other controllers to try would be fuzzy logic, adaptive, robust and neural network. Implementing a new controller would allow for comparison, to see which type of control performed better for specific tasks. If there is a gait cycle being implemented, a more complex controller might perform better, than a PID.

Another idea, once there are two legs, is to have the walking trajectory generated on-line for an uneven surface. All of the joint angle trajectories would have to be calculated on-line, depending on what the upcoming surface behaves like. Sensors would have to be attached, to detect the surface up ahead and generate a new path.

As for the electrical design, a more simplified design could be implemented. Having all of the encoder signals sent to a single board, then multiplexed together and sent to a single microcontroller, instead of using 3 microcontrollers. This would be more efficient and eliminate the need of having two motor drive boards when a single board could be used.

Bibliography

[1] Y. Choi, B-J. You, and S.R. Oh, "On the Stability of Indirect ZMP Controller for Biped Robot Systems," in *Proc. Int. Conf. Intelligent Robots and Systems,* 2004, pp. 1966-1971.

[2] F. Chunqian, and W. Xingsong, "Force-Shoes ZMP Measuring System for Human Walker," in *Proc. Int. Conf. Mechatronics and Machine Vision in Practice,* 2008, pp. 527-532.

[3] A. Cudowski. *Design, Simulation and Control of a 12 DOF Robot.* Thunder Bay, ON: Lakehead Press, May. 2009.

[4] K. Erbatur, A. Okazaki, K. Obiya, T. Takahashi, and A. Kawamura, "A study on the Zero Moment Point Measurement for Biped Walking Robots," in *Proc. Int. Conf. Advanced Motion Control*, 2002, pp. 431-436.

[5] K. Erbatur and O. Kurt, "Natural ZMP Trajectories for Biped Robot Reference Generation," *Industrial Electronics.,* vol. 56, no. 3, pp. 431-438, Mar. 2009.

[6] S. Feng, and Z. Sun, "A Simple Trajectory Generation Method for Biped Walking," in *Proc. Int. Conf. Control, Automation, Robotics and Vision,* 2008, pp. 2078-2082.

[7] A. Frank, "An Approach to the Dynamic Analysis and Synthesis of Biped Locomotion Machines," *Med. and Biol. Engineering.*, vol. 8, no. 2, pp. 465-476, 1970.

[8] F. Gubina, H. Hemami, and R. B. McGhee "On the Dynamic Stability of Biped Locomotion," *Biomedical Engineering.,* vol. BME-21, no. 2, pp. 102-108, Mar. 1974.

[9] S. Hong, Y. Oh, D. Kim, and B. J. You, "Real-Time Walking Pattern Generation Method for Humanoid Robots by Combining Feedback and Feedforward Controller," *Industrial Electronics.,* vol. 61, no. 1, pp. 355-364, Jan. 2014.

[10] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning Walking Patterns for a Biped Robot," *Robotics and Automation.,* vol. 17, no. 3, pp. 280-289, 2001.

[11] S. Ito, H. Takishita, and M. Sasaki, "A Study of Biped Control using Porportional Feedback of Ground Reaction Forces," in *Proc. Int. Conf. SICE-ICASE,* 2006, pp. 2368-2371.

[12] S. Kajita, T. Yamaura, and A. Kobayashi, "Dynamic Walking Control of a Biped Robot Along a Potential Energy Conserving Orbit," *Robotics and Automation.,* vol. 8, no. 4, pp. 431-438, Aug. 1992.

[13] O. Kurt, and K. Erbatur, "Biped Robot Reference Generation with Natural ZMP Trajectories," in *Proc. Int. Conf. Advanced Motion Control,* 2008, pp. 403-410.

[14] C. Leslie Golliday, and H. Hemami, "An Approach to Analyzing Biped Locomotion Dynamics and Designing Robot Locomotion Controls," *Automatic Control.,* vol. 22, no. 6, pp. 963-972, 1977.

[15] Q. Li, A. Takanishi, and I. Kato, "A Biped Walking Robot Having A ZMP Measurement System, Using Universal Force-Moment Sensor," in *Proc. Int. Conf. Intelligent Robots and Systems,* 1991, pp. 1568-1573.

[16] A. R. Newton, and A. L. Sangiovanni-Vincentelli, "Relaxation-Based Electrical Simulation," *Electronic Devices.,* vol. 30, no. 9, pp. 1184–1207, 1983.

[17] K. Nishiwaki, Y. Murakami, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "A Six-axis Force Sensor with Parallel Support Mechanism to measure the Ground Reaction Force of Humanoid Robot," in *Proc. Int. Conf. Robotics and Automation,* 2002, pp. 2277-2282.

[18] J. Park, Y. Youm, and W-K. Chung, "Control of Ground Interaction at the Zero-Moment Point for Dynamic Control of Humanoid Robots," in *Proc. Int. Conf. Robotics and Automation,* 2005, pp. 1724-1729.

[19] J. Park, and Y. Rhee, "ZMP Trajectory Generation for Reduced Trunk Motions of Biped Robots," in *Proc. Int. Conf. Intelligent Robots and Systems,* 1998, pp. 90-95.

[20] P. Sardain, and G. Bessonnet, "Forces Acting on a Biped Robot. Center of Pressure-Zero Moment Point," *Systems, Man, and Cybernetics- Part A: Systems and Humans.,* vol.34, no. 5, pp. 630-637, Sept. 2004.

[21] B. Siciliano, and O. Khatib. *Springer Handbook of Robotics.* Secaucus, Nj: Springer-Verlag New York, Inc, pp. 19-21. 2007.

[22] B. Siciliano, and O. Khatib. *Springer Handbook of Robotics.* Secaucus, Nj: Springer-Verlag New York, Inc, pp. 139-140. 2007.

[23] B. Siciliano, and O. Khatib. *Springer Handbook of Robotics.* Secaucus, Nj: Springer-Verlag New York, Inc, pp. 139-141. 2007.

[24] B. Siciliano, and O. Khatib. *Springer Handbook of Robotics.* Secaucus, Nj: Springer-Verlag New York, Inc, pp. 144-146. 2007.

[25] H. P. Siregar, "Simulation of Power Consumption for Walking Robots," in *Proc. Int. Conf. Workshop on Robot Motion and Control,* 2005, pp. 27-32.

[26] W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control.* New York, NY: John Wiley and Sons, pp. 61-82. 2004.

[27] T. Sugihara, Y. Nakamura, and H. Inoue, "Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control," in *Proc. Int. Conf. Robotics and Automation,* 2002, pp. 1404-1409.

[28] S. de Torre, L. M. Cabas, M. Arbulu, and C. Balaguer, "Inverse Dynamics of Humanoid Robot by Balanced Mass Distribution Method," in *Proc. Int. Conf. Intelligent Robots and Systems,* 2004, pp. 834-839.

[29] M. Vukobratovic and J. Stepanenko. *On the stability of anthropomorphic systems*, Amsterdam: Elsevier Scientific Co. pp. 1-37. 1972.

[30] L. Wang, Z. Liu, C. L P Chen, Y. Zhang, S. Lee, and X. Chen, "Energy-Efficient SVM Learning Control System for Biped Walking Robots," *Neural Networks and Learning Systems.,* vol.24, no. 5, pp. 831-836, May. 2013.

[31] Y. Wei, B. Gang, and Z. Wang, "Balance Recovery for Humanoid Robot in the Presence of Unkown External Push," in *Proc. Int. Conf. Mechatronics and Automation,* 2009, pp. 1928-1933.

[32] S. Yazdekhasti, F. Sheikholeslam, and M. Ghayour, "Stability Analysis of Biped Robot with Direct Control of Zero Moment Point," in *Proc. Int. Conf. Computer and Automation Engineering,* 2010, pp. 528-532.

[33] Z. Yu, X. Chen, Q. Huang, H. Wang, S. Zhang, W. Xu, J. Li, G. Ma, W. Zhang, and N. Fan, "Humanoid Walking Pattern Generation Based on the Ground Reaction Force Features of Human Walking," in *Proc. Int. Conf. Information and Automation,* 2012, pp. 753-758.

[34] B. Yuksel, C. Zhou, and K. Leblebicioglu, "Ground Reaction Force Analysis of Biped Locomotion," in *Proc. Int. Conf. Automation and Mechatronics,* 2004, pp. 330-335.

Appendix A

Simulation Diagrams

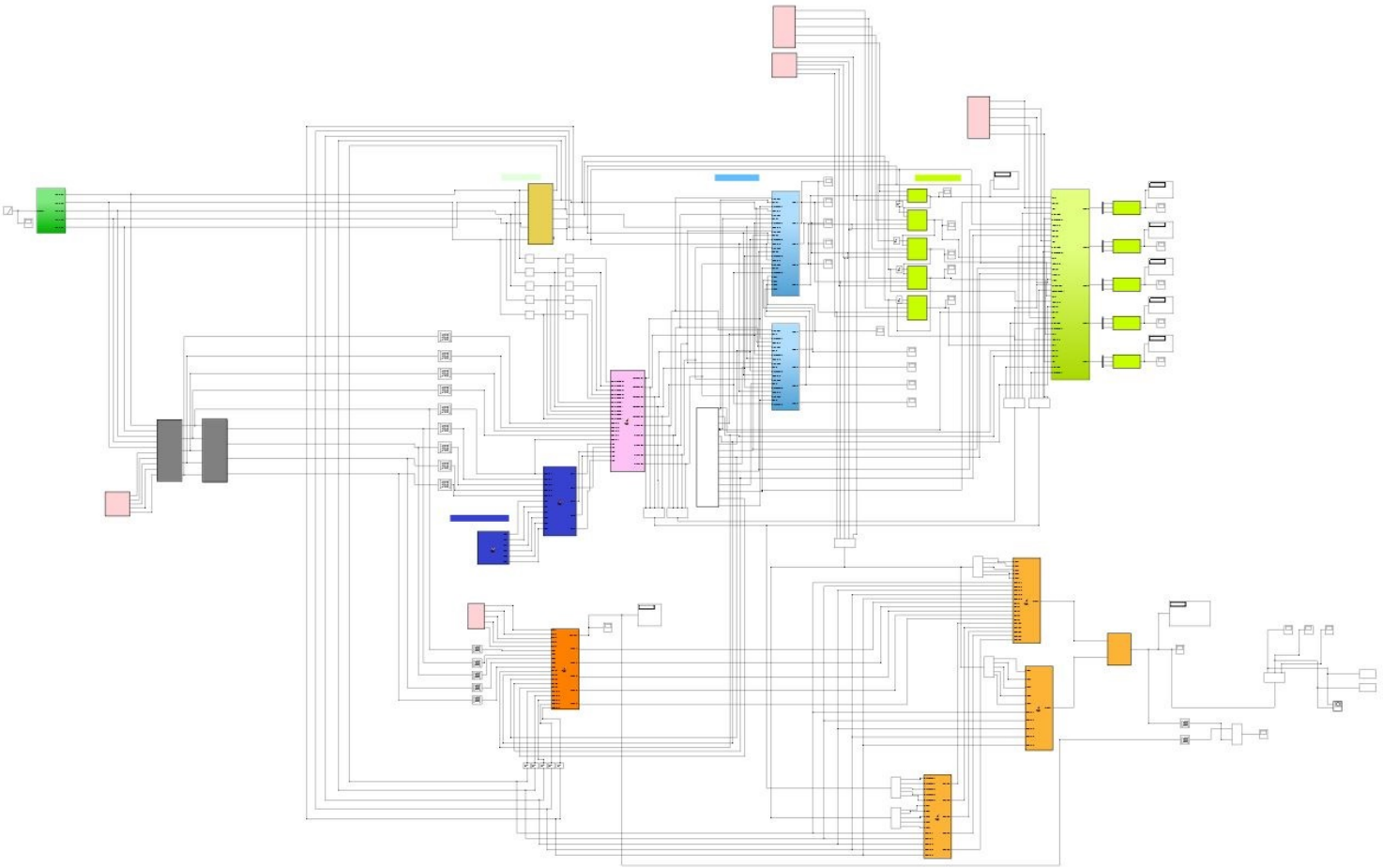

Figure A.1: Whole simulation in Simulink

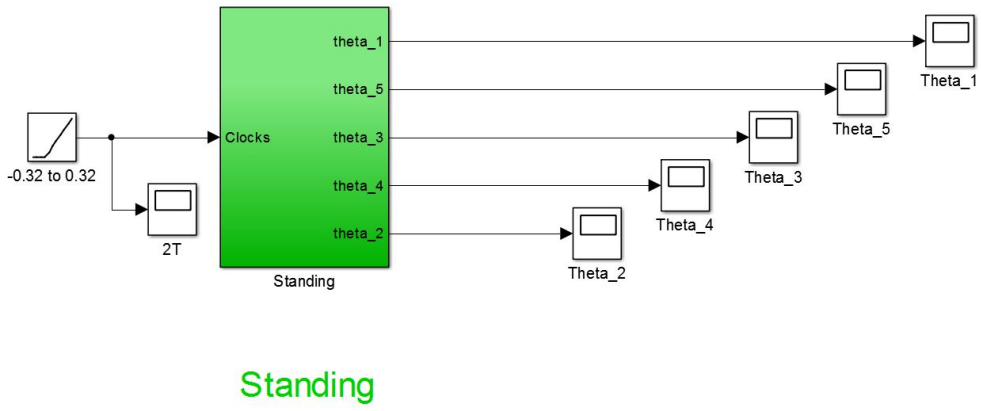

Figure A.2: Standing joint trajectory in Simulink

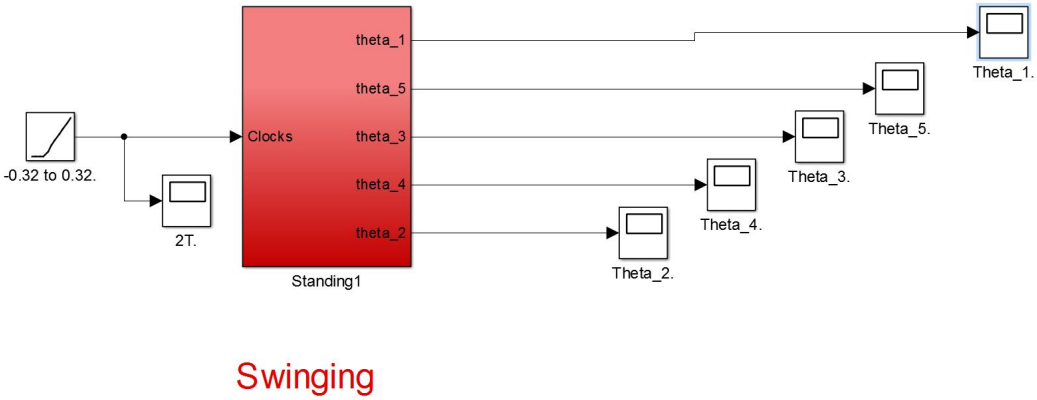

Figure A.3: Swinging joint trajectory in Simulink

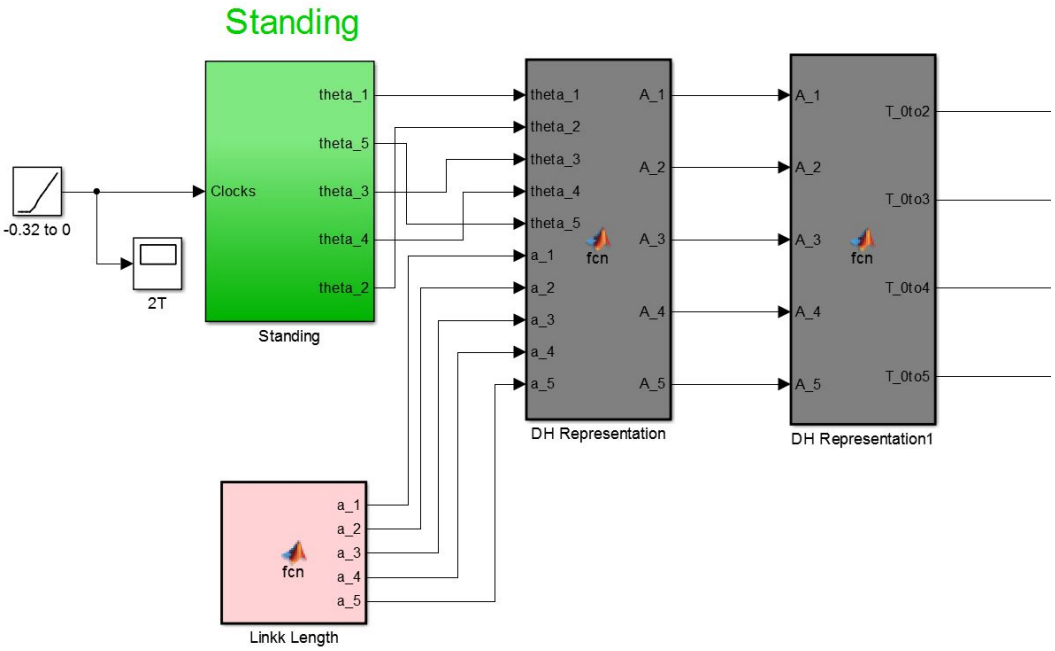

Figure A.4: DH parameters during standing phase

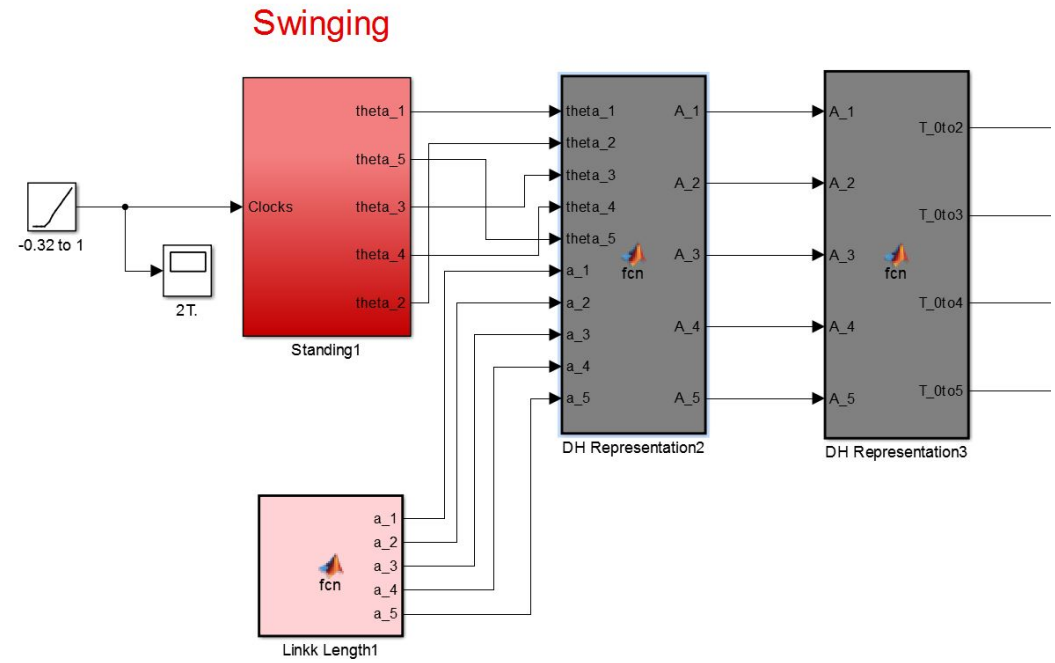

Figure A.5: DH parameters during swinging phase

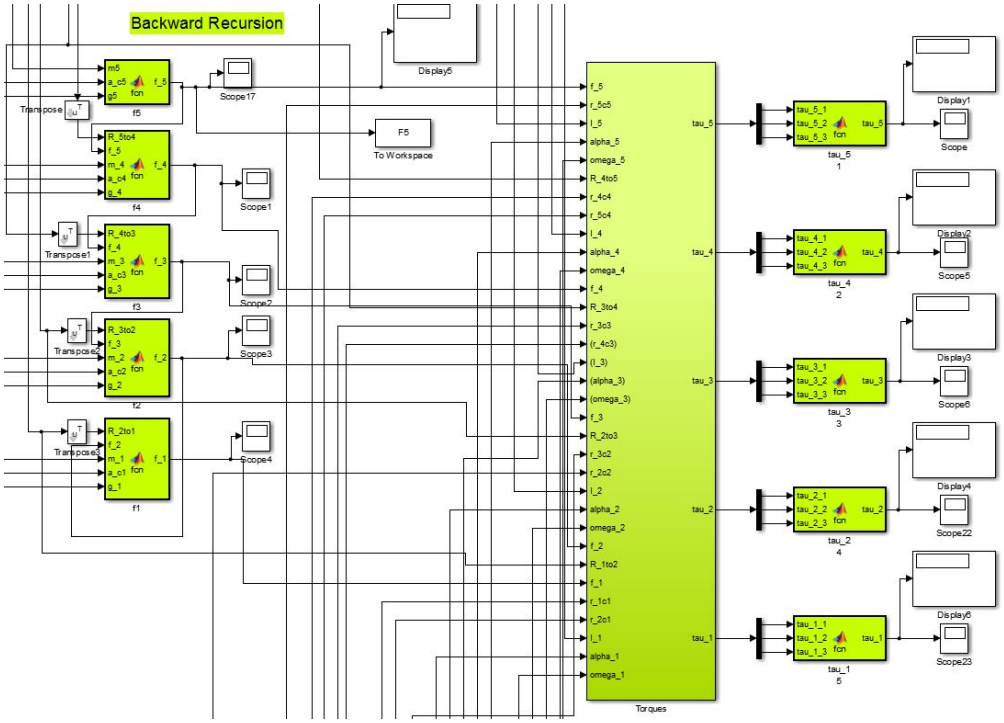

Figure A.6: Newton-Euler

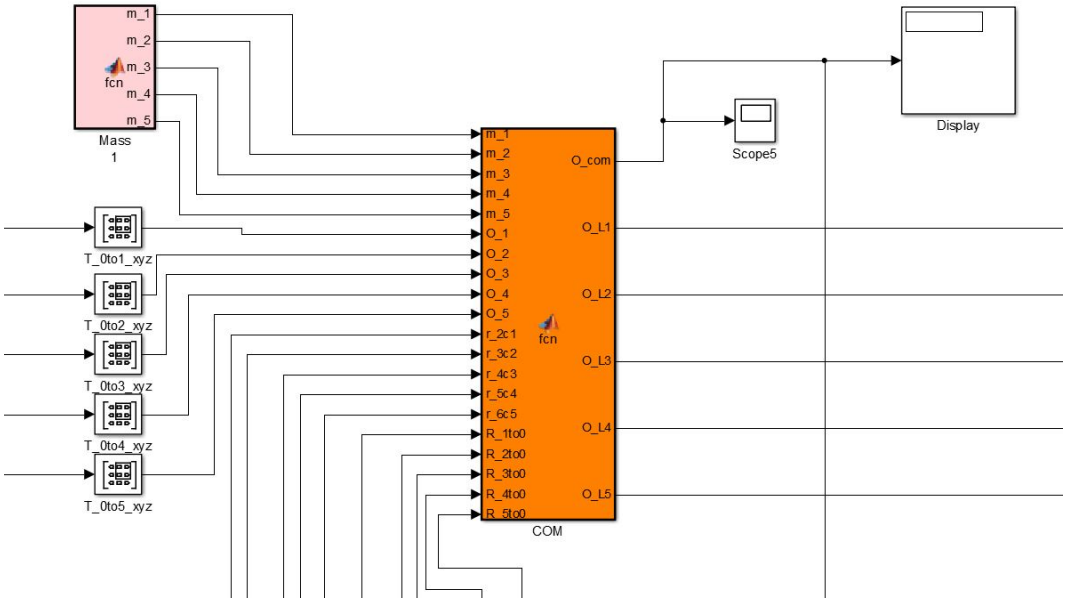

Figure A.7: COM

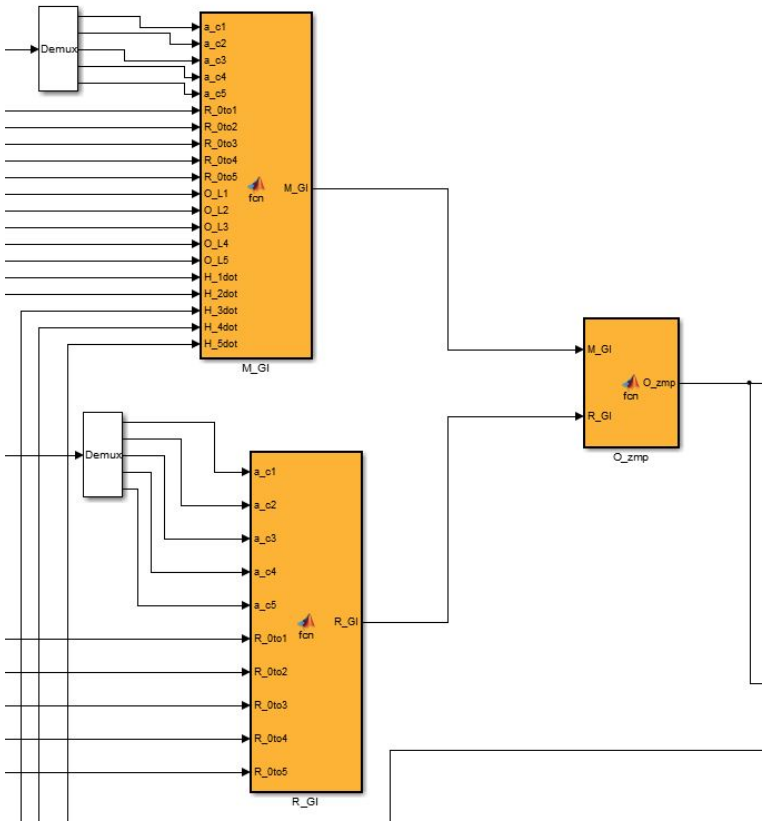

Figure A.8: ZMP

Appendix B

Experimental Setup

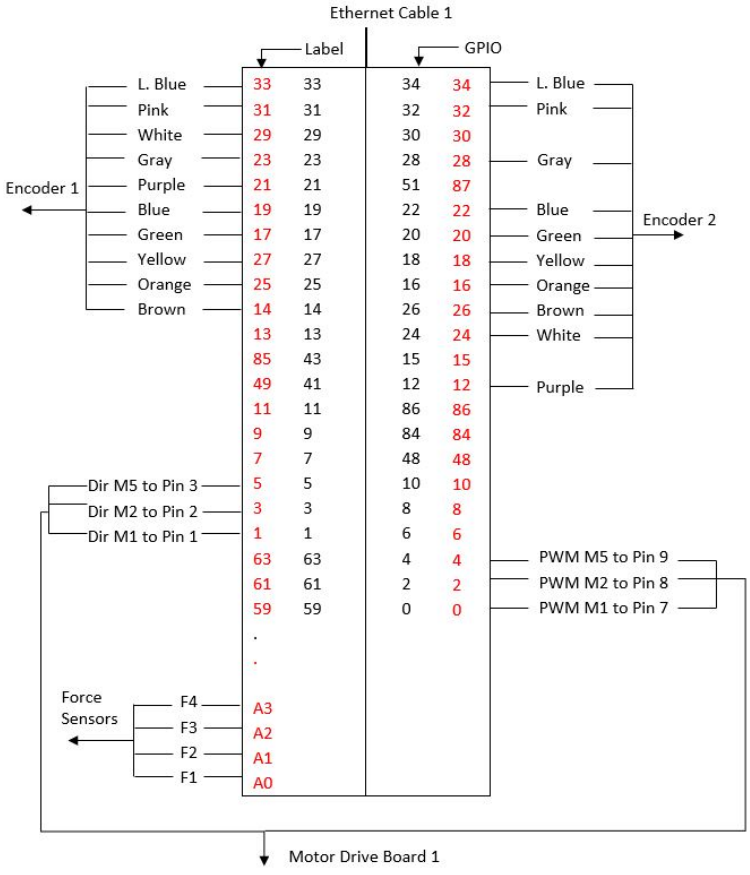

Figure B.1: Microcontroller board 1 setup

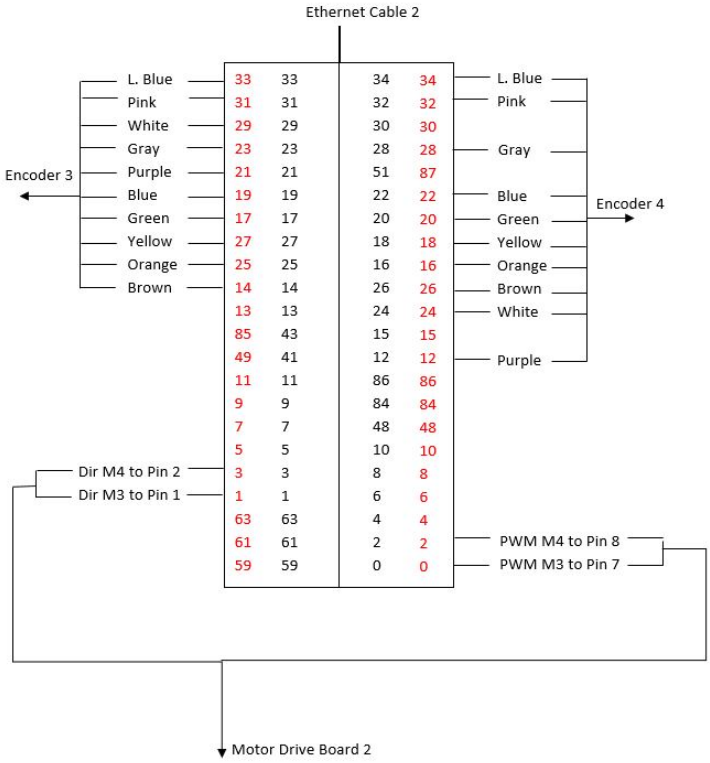

Figure B.2: Microcontroller board 2 setup

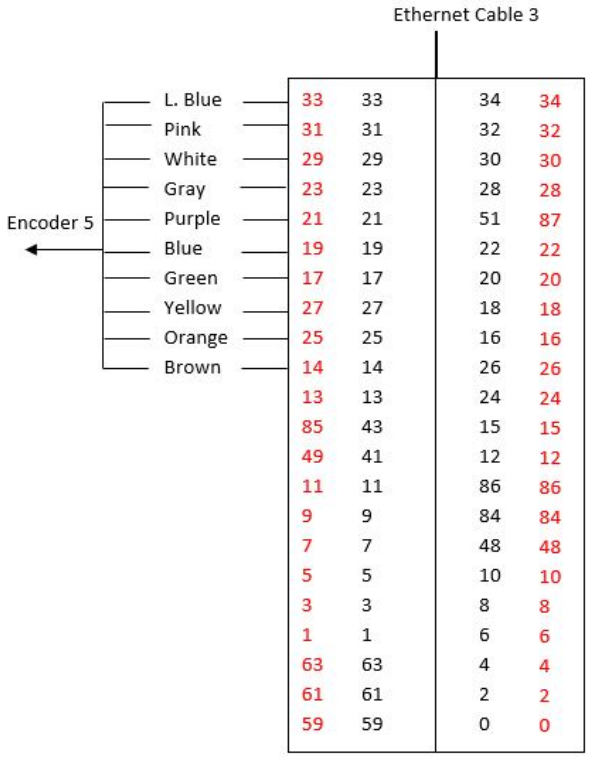

Figure B.3: Microcontroller board 3 setup

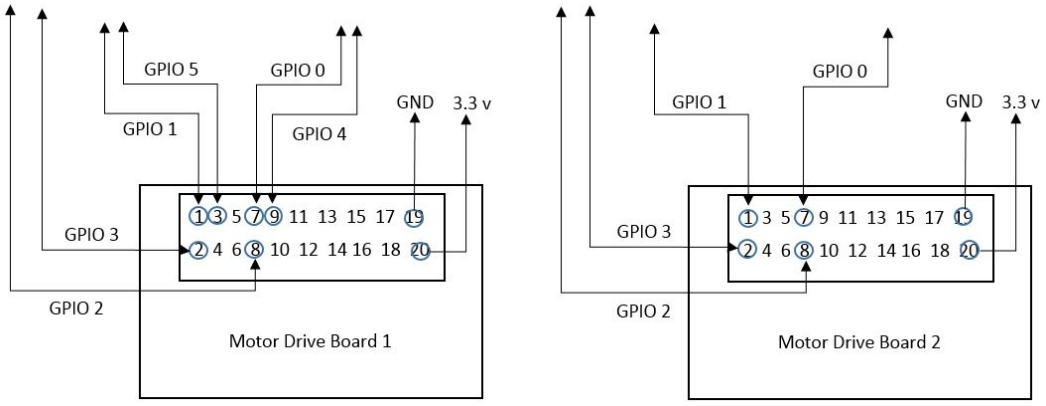

Figure B.4: Motor driver boards setup

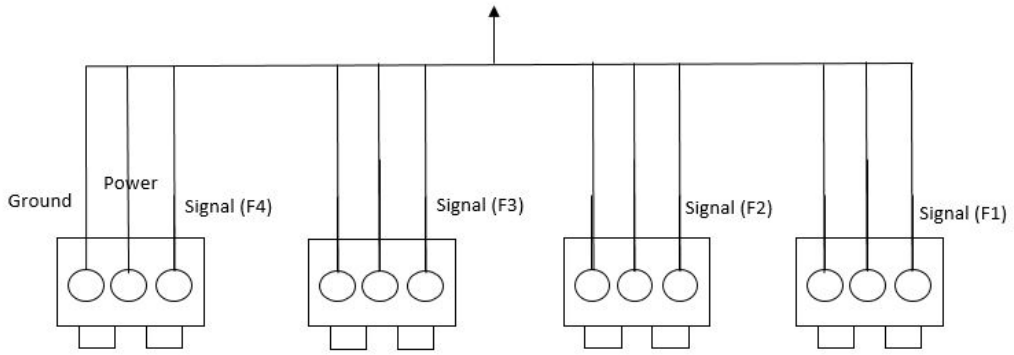

Figure B.5: Force moment sensor setup

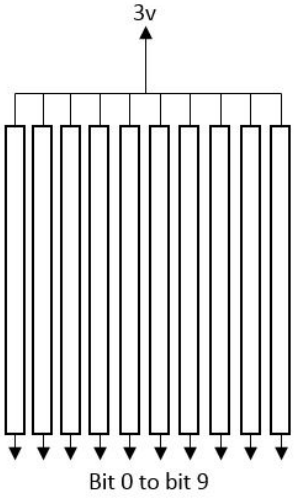

Figure B.6: Resistor bank

It's important to note that there must be a common ground between all components, and the supply voltage for the encoders is 12 to 24 volts.