Controlling Locomotion of a Robotic Leg

Sponsors: Simon Xing (sixing@calpoly.edu) and Charlie Refvem (crefvem@calpoly.edu)

Team Members:

Tai Mitchell

tmitch08@calpoly.edu

Patrick Ward

paward@calpoly.edu

Ben Spin

bspin@calpoly.edu

Connor Bush

Cobush@calpoly.edu

Mechanical Engineering Department

California Polytechnic State University, San Luis Obispo

June 2021

Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Abstract

Dr. Xing and Professor Refvem are working to research and develop a quadruped robot that is capable of basic movements including walking, running, and jumping. As senior project group F-11, we are joining a team of engineers to assist in the development of the quadruped. Our team was tasked with creating a mathematical model, designing a control method, and implementing that control method for the quadruped's legs in Simulink. This will allow both current and future students to understand the response of the system and provide a building point for future researchers to create working quadrupedal robots. This report documents our research and cumulative work to reach our goals. The report highlights our final design for the controller loop, our implementation process for each controller component, and our design verification tests to justify our work.

Table of Contents

Contents

1.0 Introduction

This section of the report introduces the origin of this project and the overall design objectives. At the end of this introduction, the reader should be familiar with the individuals behind this project's proposal, and the team that will lead its design.

1.1 The Customer

Dr. Xing and Professor Charlie both work at California Polytechnic State University. They are interested in combining their disciplines of mathematical modeling, and electrical mechanical interactions respectively, to form the Cal Poly Legged Robots team. The Legged Robots team strives to develop a full-size quadrupedal robot to be deployed for search and rescue, as well as for last mile delivery. Future students interested in mechatronics and robotics will continue the development of this robotic leg from the control system developed by this team.

1.2 The Problem

The goal of this project is to develop a way to control the locomotion of a single robotic leg. The potential applications for this leg are numerous, but the initial concept is for search and rescue missions or last mile delivery. The goal for the leg is to be capable of passing a vertical jump test in which the robot will be able to jump to 40% of its own hip height repeatedly. As this project will not be completed this year, it is imperative that the process of this project is well documented so it can be easily continued by future students.

1.3 The Team

Our senior project group is composed of four members, Tai Mitchell, Ben Spin, Patrick Ward and Connor Bush. Each member is pursuing an undergraduate degree in mechanical engineering and on track to graduate in Spring 2021. Tai Mitchell was raised on the island of Kauai and applies a positive mind set and strong work ethic to maintain productivity in team environments. Ben Spin grew up in Grass Valley and is interested in how mechatronics and automation will affect the changing landscape of the automotive industry. Patrick Ward is from Modesto, California and is interested in robotics, automation, and electronic mechanical interactions. Connor Bush is an outdoors man and a jack-of-all-trades that is pursuing a concentration in mechatronics and a minor in physics.

1.4 This Report

This report serves as a Final Design Review for our senior project. It outlines the final product of our controller and our design direction. This report highlights the background information for this project in Section 2.0, detailing the research conducted and customer considerations. Section 3.0 covers the overall objective of the project how it will satisfy our sponsors' and futures students' needs. The initial modeling procedure is in Section 4.0, showing how we first iterated through our designs. The second half of the report details our final controller design in Section 5.0, our process for creating, assembling, and implementing all design components in Section 6.0, and how we tested and verified our design in Section 7.0. In addition, this report covers how this project was managed in Section 8.0, and it includes our conclusions and recommendations for the future of this project. Attachments are included for all documents referenced in the paper as well as references to any academia cited.

2.0 Background

The Background section discusses the foundational knowledge required to understand project. This section provides information and references for every solution method discussed in this report. The background section includes detail discussions of all our sources and the critical findings from our research into industry and current quadruped robots.

2.1 History

Quadruped robots were first conceptualized in the 1960’s, but it was not until the 1970’s that the first comprehensive quadruped was developed [1]. It quickly became obvious that designing the control system was the largest challenge in designing a stable quadruped. Following this realization, universities and companies have worked to develop control systems that account for the dynamic motion of a quadruped [2]. This included an in-depth study of natural quadrupeds like dogs, horses, etc. to help determine the motion and response that the system needs [3]. This section will investigate the team's findings from the primary sponsor meetings, alternate solutions, patents, and technical research.

2.2 Customer Wants and Needs

Weekly meetings are conducted with sponsors Dr. Xing and Professor Refvem. From these meetings, the goals for the final product, due by the end of this year, were determined. A single robotic leg has been modeled using computer aided design. This CAD model will be provided for generating a 3D animation in Simscape, which will be used as a testing tool for our analytical models. The direction of this project is to integrate a control system into the current leg model with constraints like inertia, dampening, and disturbances. Dr. Xing and Professor Refvem talked to us about potential differences in Senior Project procedure for our project. We determined that ideation for this project would occur in selecting a control method. We extended our focus to one other key group, the future students that will take over our work to complete the physical design of the quadrupedal robot. These students require complete and thorough documentation of our work which will allow them to continue research on the quadrupedal robot with minimal confusion.

As a final product, this design would be marketed as a search-and-rescue or last-mile-delivery robot. Based on the robots desired ability to be controlled remotely and transverse a wide range of terrains, this product would excel in expanding a search perimeter in the organizers need extra assistance with finding missing individuals.

2.3 Research

Our research included searches into other quadruped robots, control systems, animal gaits, and motor function. Combining these fields of research, our team to develop a more thorough understanding of how quadrupeds work and how to apply that to our own project. Because quadruped robots have been developed over the past 50 years, there is an extensive selection of scholarly articles and references available to us.

Table 2.1 is a list of existing products. Each product has a description in the paragraphs following the table. All these designs can provide great inspiration and will help lead us through the process of creating our own control system.

Table 2.1 Existing Quadrupedal Robots

Sources	Existing Designs
Boston Dynamics	Big Dog
NYU	SOLO
MIT	Cheetah 3
Stanford	The Pupper
Zhejiang University	Small Prototype

Boston Dynamics designed a working robotic search and rescue robot. The walking gait was modeled after a horse's gait. The robot is capable of traversing stairs, hills, and remains standing after being pushed from the side. This project was conducted for a MATLAB competition to display the modeling capabilities of Simulink [4]. Boston Dynamics provides a formal report and documentation of programming code that details the progress made from initial kinematic analysis to integration of a control system. Boston Dynamics serves as both a guide for the designing of a quadrupedal robot and as an example of the potential of this project.

NYU presents a relatively low-cost quadruped that can perform standard functions such as running, walking jumping and navigating. A specific topic of interest is their use if torque-controlled motors to mimic behavior of animals [5]. It was discussed between the senior project team and sponsors that manipulation of torque-controlled motors would be used to dampen the impact felt by the robotic leg after performing a jump. This source discusses how variable stiffness was tested by modeling the controller with different parameters. NYU's study will help in fine tuning the control system integrated into our robot.

One well known quadruped is the mini cheetah, which is recognized for being the first quadrupedal robot to do a backflip. The robot weighs 20lb, uses gear boxes from motor to legs to maximize torque input, and has three motors on each leg. One exciting feature of this design is that the robot uses reactive balance and locomotion controllers [6]. The complexity of the controller design is beyond the scope of this project but will serve as a guide in developing controllers for the legs in the future.

Stanford university developed a small bot called Pupper that can hop, trot, and run. The basis of the design is that the robot should be low cost and of simple design. Stanford's robot features a RPi4 computer that receives commands from a PS4 controller. The robot has 3 servo motors per leg and its motion can be simulated in PyBullet before physical testing [7][8]. Stanford's team provides software installation, calibration, and robot operation information that will serve as a helpful example in this team's own design process.

Zhejiang University produced the final existing product and quadrupedal robot that we investigated during preliminary design. This source provides documentation on the design of a small robot capable of running and climbing stairs. The final product weighs 20lbs with a controller operating at 2kHz [9]. To consider a wide range of design conditions, it is important to study existing designs such as this one to produce an effective and well documented design.

Table 2.2 lists technical literature that that was sourced to help this team with the initial mathematical modeling of the robot leg system and the design of the control system. Three of the sources are textbook references, two are papers, and one is an article.

Table 2.2 Technical Literature for Robotic Leg Control Systems

Technical Literature	Authors
Vector mechanics for engineers	Beer, F. P., Johnston, E. R., Mazurek, D. F., Cornwell, P. J., & Self, B. P.
Control Systems Engineering	Nise, N. S.
Control systems engineering: a practical approach	Owen, F.
Physical Modeling in MATLB	Allen, B. D.
Quadrupedal Robot Locomotion: A Biologically Inspired Approach and Its Hardware Implementation	Espinal, A., Rostro-Gonzalez
Cam Drive Step Mechanism of a Quadruped Robot	Sun, Q., Wang, C

To accurately predict the motion of the leg, equations of motion need to be determined for each linkage of the leg. Forward kinematics will be used to place the leg in the desired moving position, and inverse kinematics will return the leg to prepare the leg for its next movement directive [10]. The textbook *Vector mechanics for engineers* will be an essential source in the first portion of this project.

Norman Nise's *Control Systems Engineering* will be an invaluable source throughout this entire senior project. The textbook includes sections like modeling in the time domain, time response, and reduction of multiple subsystems [11]. Most of the team is taking ME-422 Mechanical Controls alongside senior project and will employ this textbook to learn the skills necessary to design a working control system.

The final textbook source is *Controls systems engineering: a practical approach*. Whereas the other controls system textbook is a more general overview of controls method. This source provided by Frank Owen takes a more practical approach to the implementation of control systems into larger systems [12]. Beyond the theory, this textbook will be able to help a lot with the debugging and troubleshooting that this team will have to undergo in the process of fine tuning the controller to mimic the physical model.

The first paper source that was selected is Allen Downey's Physical Modeling in MATLAB. MathWorks provides a directory on physical modeling and is focused on teaching how to program, model, and simulate in MATLAB [13]. This project requires the conversion our equations of motion into a compatible form for MATLAB script. Modeling of the robotic leg will need to be produced with in Simscape while the control system will be produced in Simulink. All these steps will require an in-depth understanding of MATLAB and the text *Physical Modeling in MATLAB* will provide immense support [14]. In addition, we will need to be able to import the CAD model provided to us into MATLAB or Simscape [15].

Another paper source found was a text of Quadrupedal Robot Locomotion: A Biologically Inspired Approach. This source directly addresses the key considerations of quadrupedal robot design. The paper documents the design methodology with a solid mathematical background, the biological validation, numerical simulations, and hardware implementation [3].

Finally, we sourced an article on the Cam Drive Step Mechanism of a Quadruped Robot. The article is thorough in its investigation of the mechanical theories of the quadruped robot walking mechanism and selection of required control parameters. The article also describes the kinematic principles of the quadruped robot walking mechanism and the construction of a three-dimensional model that can be prototyped and tested [1].

Table 2.3 Additional Sources to Develop Mathematical Model

Additional Sources	Authors
“Back Matter.” Power System Operations	Baringo, Luis, and Antonio J. Conejo
Differential Equations and Linear Algebra	Goode, S. and Annin, S.
"A Lagrangian network for kinematic control of redundant robot manipulators"	Jun Wang, Qingni Hu and Danchi Jiang
Minor Change, Major Gains: The Effect of Orientation Formulation on Solving Time for Multi-Body Trajectory Optimization	Knemeyer, A., Shield, S., Patel, A.,
Introduction to Robotics: Mechanics and Control.	Craig, John J

The text *Power Systems Operations* details how to use the newton Raphson method combined with the Jacobian matrix to solve systems of non-linear equations [16]. It gives easy to follow instructions to follow for varying scenarios, including a system of non-linear equations containing multiple unknowns like our own design. Following these steps will allow us to complete our backward kinematic analysis and begin forming transfer functions for our controller. In addition, a portion of the section touches on solving optimization problems which may be helpful for our own system once we move farther into our controller designing stages.

The purpose of *Differential Equations and Linear Algebra* is to serve as a reference for the development of our mathematical model [17]. The textbook includes chapters that cover first-order differential equations, matrices, and systems of linear equations. One of the most useful sections can be found in chapter 9 and 10. Chapter 9, systems of differential equations, explores the use of Jacobian matrices which serve as the primary solution method for determining the inverse kinematics of our system. Chapter 10, the Lagrange method, serves as an introduction into using the Lagrangian method to develop the correct number of solutions to match our degrees of freedom. This source will be central in our investigation of possible solution methods for our mathematical model.

To control our robotic leg, we will be utilizing the Lagrangian method. *A Lagrangian network for kinematic control of redundant robot manipulators* outlines a technique used for controlling high redundancy operations with real time solutions to inverse kinematic problems [18]. The text also includes example Simulink block diagrams that will be helpful references when outlining our plan for our own model. When compared with other potential models for robotic leg control, the real-time approach as outlined in this paper has much higher precision of returned position and velocity. This resource offers a detailed approach to an advanced control system that, even if not directly applied, gives wider range and context to the Lagrangian method.

Minor Change, Major Gains discusses the advantages and disadvantages of using relative and absolute coordinates to define systems of connected links. Using an absolute coordinate system requires more physical constraints and generates a larger number of coordinate variables needed to define the systems joints. A relative coordinate system generates fewer coordinate variables but generates many complex Coriolis terms since each link defined relative to the previous links. These Coriolis terms are shown to significantly impact the time it takes to solve the equations of motion and could potentially have a negative impact on linked systems like ours. The kinematics and kinetics of quadrupeds are directly linked to the Jacobian matrix that describes the position of each joint. Reducing the complexity of the Jacobian greatly increases the rate at which the solver can determine the solution to the equations of motion. This can be done in two ways; reducing the number of links in the system and reducing the dependency of coordinate variables on each other [19]. This paper argues that as the number of links in the system increases, absolute coordinate systems require roughly 50% less calculations than relative coordinate systems. Though the paper pushes for absolute coordinate system it continuously presses the underlying question: Will increasing the complexity of the equations justify the decrease in solution time?

Introduction to Robotics: Mechanics and Control has many applicable sections. The book details manipulator kinematics, inverse manipulator kinematics, Jacobians: velocities and static forces, linear control of manipulators, nonlinear control of manipulators, and force control of manipulators [20]. This book contains the information to guide us through completing the forward and backward kinematics as well as our mathematical model. In addition, its discussions on control systems will give us tools to choose the best controller for our system. The text contains examples and descriptions of the necessary methods to accomplish all our analysis and design.

The last section of our research regarded finding current patents that we can use for our project. A list of these patents can be found in Table 2.4. Each one applies to an important mechanism to consider in our project.

Table 2.4 Patents for Robotic Leg Control Systems

Patents	Patent #
Servo System and Driving Device for Controlling Servo System	US Patent No. 8659254
Method and Apparatus for Dynamic Control of Robot	US Patent No. 4834200
Closed Loop Servo Motor Response	US Patent No. 9018544 B2
Servo Motor Controller and Self-Monitoring Function of Stiffness	US Patent No. 9429936 B2

2.4 Kinematics: Forward and Inverse

Forward and inverse kinematics are two key components in modeling the physical motion of our robotic leg. Forward kinematics refers to the set of equations defined by a model's geometry used to compute position based on joint parameters. Forward kinematics are used in the final step of our control loop to track position after we have selected a motion path. Inverse kinematics is the process of determining what joint angles are necessary to produce a certain position. Inverse kinematics is a valuable tool that serves as the first component in our control loop to determine the necessary angles of the hip and knee based on the input footpath.

2.5 Newton Raphson Method

The Newton Raphson method is an approximation method that finds roots of real valued functions. This method calculates changes in the value of variables to approach a known solution. It begins using a guessed value. Because the final value is known, the Newton Raphson method calculates the difference between the guessed value and the final value. The guessed value is then updated using the error from the previous iteration and used as the guessed value in the subsequent iteration. By updating the guess value with the error from the previous iteration, the method quickly converges to a precise approximation. As the value approaches the desired final input, the error goes to zero and the method has effectively approximated the solution.

2.6 Lagrangian Mechanics

Lagrangian Mechanics is a branch of Classical Field Theory that enables the description of physical systems in terms of the degrees of freedom. Lagrangian Mechanics can only be applied to systems with finite degrees of freedom. The result of using Lagrangian Mechanics with match the results of using Newtons Second Law assuming the system is defined using the same degrees of freedom. There is a quantum analog to Lagrangian Mechanics, Quantum Field Theory, but this will be unnecessary for this application. Lagrangian Mechanics utilizes partial derivatives to completely define the equations of motion for a system. The simplest application of Lagrangian Mechanics is the Euler-Lagrange equation, where the system is not acted on by any external forces. However, because our project involves not only external forces, but also a position constraint on the leg we will need a more developed Lagrangian equation. The external force and position constraint are accounted for by Rayleigh's Dissipation Factor and Lagrangian Multiplier, respectively. The application of Lagrangian Mechanics and the required components is discussed further in Section 4.3.5.

3.0 Objectives

Included in this section are the primary design considerations and requirements of this project. Our sponsors main objectives, along with reach goals, are included in this section. This section also includes the expectations we have set for ourselves in terms of quality of work and documentation.

3.1 Problem Statement

Robotics researchers in conjunction with professors Xing and Revfem, need to develop a control system for the locomotion of a single robotic leg. The potential applications for this leg are numerous, but the initial concept is for search and rescue missions or last mile delivery. This leg must be capable of passing a vertical hopping test in which the robot will be able to jump to 40% of its own hip height repeatedly. As this project will not be completed this year, it is imperative that the process of this design is well documented such that it can be easily continued by future students.

Figure 3.1 depicts boundary diagram of the problem that we are taking on. The dotted lines demonstrate the area where we can create solutions and make contributions. Once all the mathematical modeling and control programming are implemented, we hope to test with a physical model depending on Cal Poly regulations.

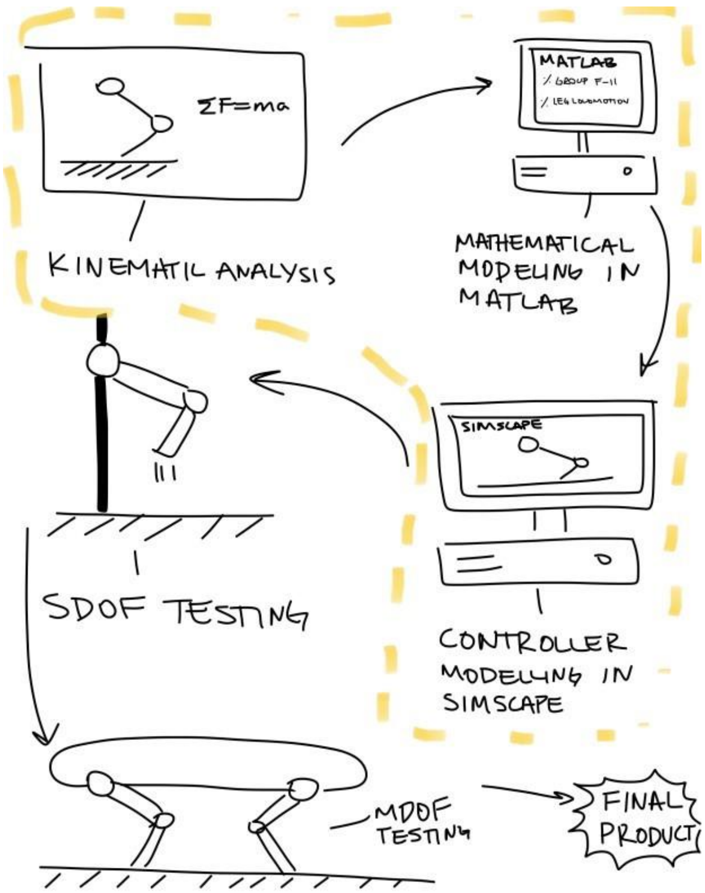


Figure 3.1 Boundary Diagram for Problem Definition

3.2 Scope

Professor Refvem and Xing are working in conjunction with other senior project teams and Cal Poly Legged Robots Team to develop a quadrupedal robot. Currently the robot's development is in its early stages. One current senior project team is working on delivering a physical model of the leg. They will provide a CAD model for us to use in our work towards a working control model and program.

3.3 Needs and Wants

After discussing with Dr. Xing and Professor Refvem we were able to create a list of needs and wants tailored to our sponsors. In addition, we have incorporated other students that will interface with our product when it is complete. Keeping these individuals in mind will help us to create a product serving as a launch point for further projects. The last group we considered was the end customer of the fully developed quadrupedal robot. This consideration is abstract during our part of the project but is worth keeping in mind as we continue. The needs and wants of these customers are listed below in Table 3.1.

Table 3.1 Customer List of Needs and Wants

Needs	Wants
Dr. Xing and Professor Refvem	
Forward Kinematics	Robust Model (reject disturbances)
Backward Kinematics	Used for Search and Rescue
Mathematical Model	Used for Last Mile Delivery
Control Method	Jump 40-20 Percent of Hip Height
Simscape Implementation of Control Method and the Code Required	Move at Speeds Comparable to an Agile Animal
Leg Hop Up and Down Repeatably	
Future Students	
Well Documented Code	Meeting Time to Walk Through Problem
Instructions on Using our Final Product	Tutorial video of the Work we Performed over Senior Project
Direction on Where to Take the Project	
End Users	
Ability to Remotely Traverse Various Terrains	Easy to Operate User Interface

The QFD process is designed to streamline and focus the design process. The House of Quality (See Attachment A) considers the customer, requirements, specifications, competition, and the end goals. By combining these considerations into a single table, designers can identify specific areas of the design that will create a better design for the customer. The QFD completed for this project highlighted several traits that all successful quadruped projects had in common, traverse capability, a modifiable controller and thoughtful documentation. Our QFD is shown in Attachment A.

Table 3.2 lists the engineering specifications for this project that will be used to gauge out progress throughout the project. The specifics of each specification and the method of measuring each specification are listed in the description column below.

Table 3.2 Engineering Specifications

Engineering Specification	Description
Verified Hand Calculations	Kinematic calculations and equations of motion. This will be verified by our sponsors.
Documented Hand Calculations	Well commented work that any user can read, understand, and then apply. Present to intermediate dynamics class with instructor permission.
Documented MATLAB Code	Well commented work that any user can read, understand, and then apply. Present to intermediate dynamics class with instructor permission.
MATLAB Simulation	Model of the leg using calculated values and simplified materials. Visually verify that the simulation produced realistic results.
Documented Simscape Model	Well commented work that any user can read, understand, and then apply. Present to controls class with instructor permission.
Simscape Model Animation	Model of the leg using calculated values and accurate material properties. Visually verify that the animation produced is realistic to leg operation.
Calculation Walkthrough	Video or document describing the logical process of the work and the governing equations. Poll during/after PDR presentation to verify general understanding.
Code Walkthrough	Video or document describing the various functions in the code and how to use them. Poll during/after CDR to verify general understanding.
Simscape Walkthrough	Video or document describing the functions of the simulation and how to modify parameters. Poll during CDR to verify general understanding of system.
Connected Walkthrough	Video or document of the entire process, like an executive summary Poll during/after FDR presentation to verify general understanding.
Simulation Matches Physical Model	Simulated model will be compared to a physical model once the physical model is built.

As discussed previously, the focus of this project is to create a platform from which future student can easily propel this project forward. Because the scope of this project is focused on passing on the information, the highest risk specifications will be the documentation. The creation of easy to understand and adaptable controller will require attention to detail, and forethought, while working through the material to provide accurate and detailed explanations.

Table 3.3, shown below, provides more detail on the targets for our engineering specifications. These targets are performance based and therefore will be evaluated in simulation before on the test rig. These specifications consider the wants and needs of our sponsor and the end user. Much of the specifications listed below will depend on user interaction with our design. As we will not be taking this product to its final stage and we will rely on the experience and understanding of users when engaging with our code and models, to determine success in our specifications. In the table below, the right-most three columns provide additional information on the target value. Min implies that the target value is the minimum acceptable value, while max implies that the target value is the maximum acceptable value. The Risk column has three possible entries, L,M, and H which represent low, medium, and high risk respectively. Lastly, the compliance column refers to the method which we will determine compliance. T, A, I, and S represent testing, activity, inspection, and survey, respectively

Table 3.3 Target Goals to Meet Engineering Specifications

Specification #	Specification Description	Target Value	Tolerance	Risk	Compliance
1	Verified Hand Calculations	Verified by sponsors	Min	M	T, A
2	Documented Hand Calculations	20-minute presentation to intermediate dynamics students	Max	M	T, S
3	Documented MATLAB Code	20-minute presentation to intermediate dynamics students	Max	M	T, S
4	MATLAB Simulation	Visual confirmation that simulated model is not finnicky and spazzy	Min	L	I
5	Documented Simscape Model	20-minute presentation to controls students	Max	M	T, S
6	Simscape Model Animation	Visual confirmation that leg motion and state variables are realistic	Min	L	I
7	Calculation Walkthrough	75% of class gets correct answer on poll	Min	L	T
8	Code Walkthrough	75% of class gets correct answer on poll	Min	M	T
9	Simscape Walkthrough	75% of class gets correct answer on poll	Min	M	T
10	Connected Walkthrough	75% of class gets correct answer on poll	Min	H	T
11	Simulation Matches Physical Model	Robot jumps 20% of hip height	Min	H	T, I

4.0 Initial Modeling

Since the CAD model for the quadrupedal robotic leg is already designed and the scope of this project is only concerned with the control system design, we will be following a different track in the ideation process. Our ideation will focus more on choosing and justifying the proper method for any given function of the robotic leg. With the help of our senior project coach, Dr. Schuster, we determined an alternate plan to better suit our project. Our alternate plan was to determine the equations of motion, forward kinematics, and inverse kinematics of the robotic leg. Using this information, a mathematical model will be produced and then used as a starting point in designing our control system.

4.1 Functional Decomposition

The first step in our initial modeling was to break down the general functions, sub functions, and basic functions. The block design of our functional decomposition tree serves as a checklist for future code, Simulink block diagrams, and documentation. The decomposition tree, shown below in Figure 4.1, highlights the key functions we want the leg to be able to perform as well as what our sponsor wants at the end of the project. All the functions contain sub functions and basic functions that are essential to the success of this project. This tree serves as a listing of our most important design criteria.

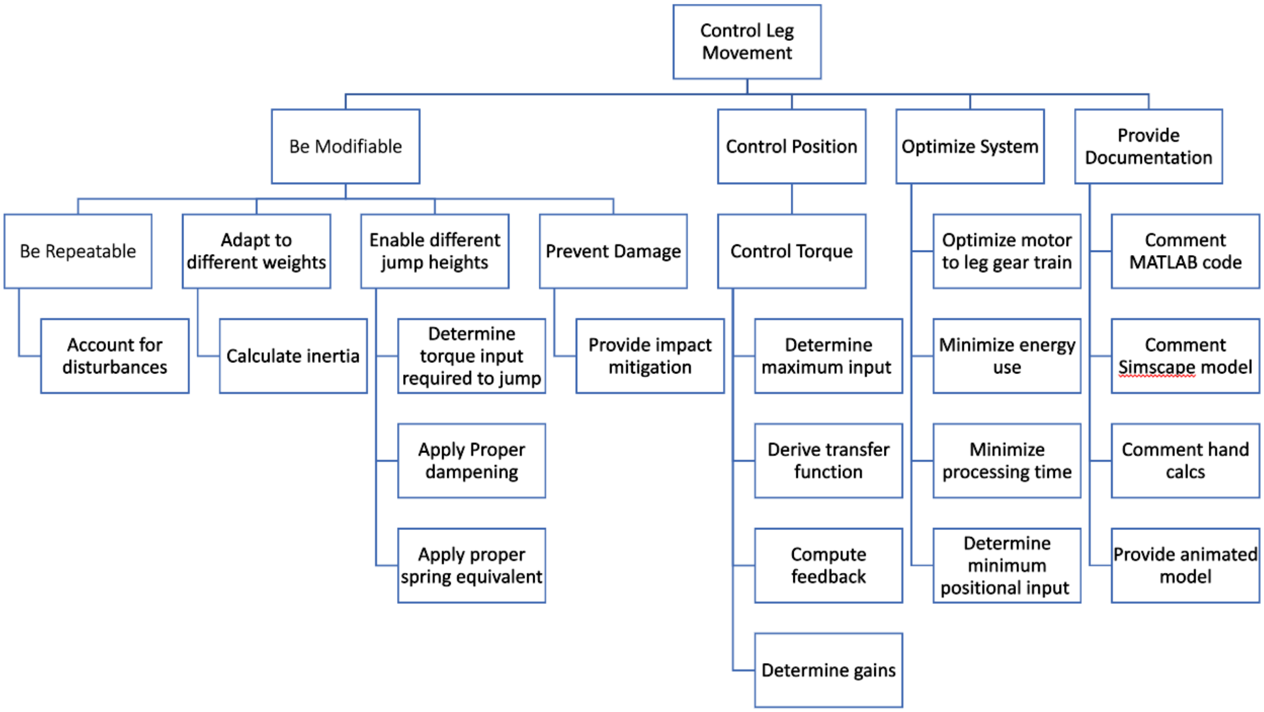


Figure 4.1 Functional Decomposition Tree

4.2 Models

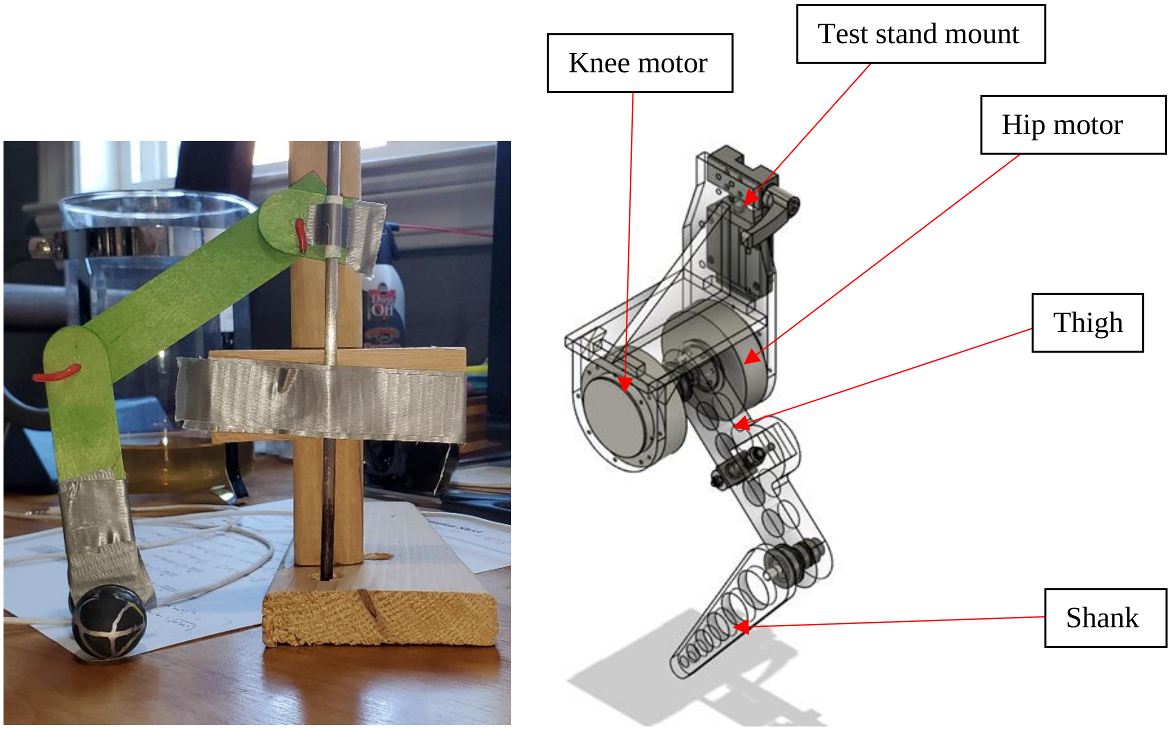


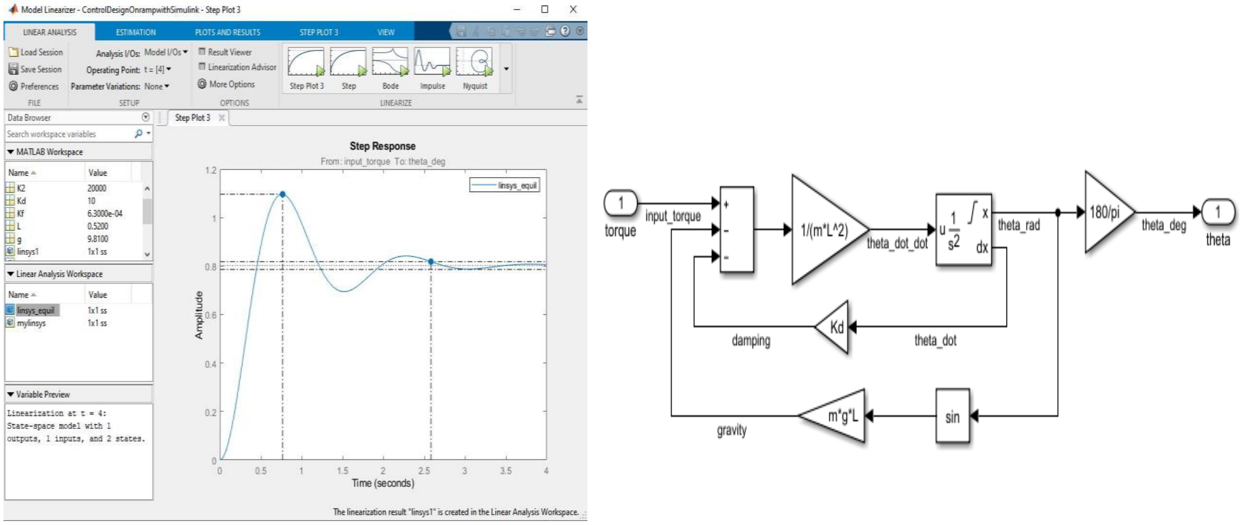
Figure 4.2 Initial Prototype vs CAD Model

Next in our modeling process, all team members produced initial prototypes that modeled the main joints and members of the robotic leg. Figure 4.2 displays the orientation of the robotic leg with relation to the test stand that it will be attached to and compares it to the actual CAD model. This model was selected as the best among the four prototypes produced. This is the only prototype shown because this project is not concerned with ideation of the robotic leg. The design of the robotic leg has already been selected and designed by a previous senior project team. These prototypes served only as visualization tools.

4.2.1 Tutorial Models

As our physical prototyping was limited, our group completed a series of Simulink tutorials that showed us the basics of designing controllers in Simulink. The two most useful tutorials are listed below and additional notes on the process are listed in Attachment B. The first two figures (4.3 and 4.4) are for the Control Design OnRamp while the last figure (4.5) is for the StateSpace tutorial.

Figure 4.3 shows the plot and block diagram of the control system produced in a tutorial for a robotic leg controller. The model features a linearized plant that can be used with a tuned PID controller to produce the desired overshoot and response time (A PID controller is a Proportional Integral Derivative controller. These controllers account for the previous behavior of the system as well as the desired behavior of the system to allow for precise control.). This model produced in Simulink is a simple model that teaches the user about control systems and feedback loops. This first section of the tutorial will serve as a basis for simplified, linear models of a robotic leg.



MATLAB Workspace

Name	Value
K2	20000
Kd	10
Kf	6.3000e-04
L	0.5200
g	9.8100
linsys1	1x1 ss

Linear Analysis Workspace

Name	Value
linsys_equil	1x1 ss
mylinsys	1x1 ss

Variable Preview

```
Linearization at t = 4:
State-space model with 1
outputs, 1 inputs, and 2 states.
```

The linearization result "linsys1" is created in the Linear Analysis Workspace.

Figure 4.3 Robotic Leg Plant Linearization

The plot shown in Figure 4.4 is an example of how to tune PID control systems to specified parameters. This model is used to produce a robotic leg with a unity feedback loop. The dotted line is the response without control and the solid line is response after controller implementation. The PID controller adjusts to an error input that comes from the output of the plant. This is a basic example of how to design controllers and will be helpful in our consideration of how to model our final controller for the robotic leg. Refer to Attachment B for a continuation of this tutorial.

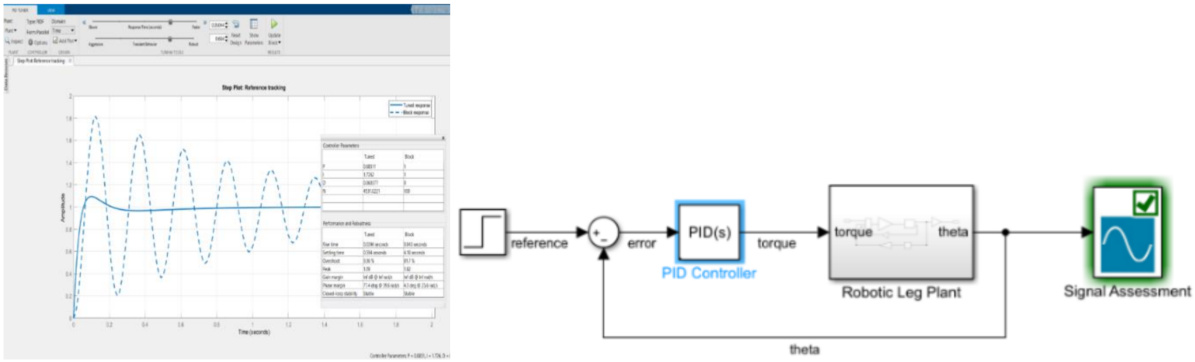
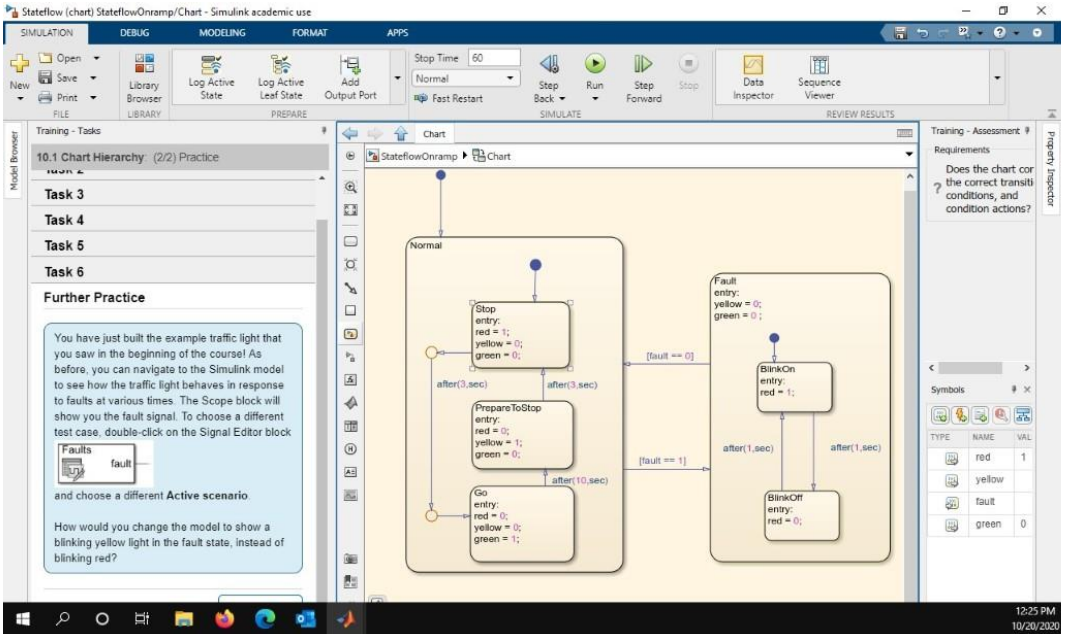


Figure 4.4 Control System PID

Figure 4.5 displays the model that was produced after completing the StateSpace tutorial. The StateSpace tutorial is used to model controller logic and works as a bridge between MATLAB and the electronic hardware. States represent the different conditions that the system can be in. The system transitions between states when specific conditions are met. In this example, the state changes from Normal to Fault when the variable “fault” is 1 or 0. Once the forward and inverse kinematics are solved, they can be used within states and implemented into the control system. We will refer to this tutorial when we begin interfacing our controller with the robotic leg hardware.



10.1 Chart Hierarchy: (2/2) Practice

Task 3

Task 4

Task 5

Task 6

Further Practice

You have just built the example traffic light that you saw in the beginning of the course! As before, you can navigate to the Simulink model to see how the traffic light behaves in response to faults at various times. The Scope block will show you the fault signal. To choose a different test case, double-click on the Signal Editor block

Faults fault

and choose a different **Active scenario**.

How would you change the model to show a blinking yellow light in the fault state, instead of blinking red?

Does the chart cor the correct transiti conditions, and condition actions?

TYPE	NAME	VAL
	red	1
	yellow	
	fault	
	green	0

Figure 4.5 State Space

4.2.2 Physical Model

Figures 4.6 and 4.7 illustrate the two scenarios that we are analyzing. Point O represents the hip, point A represents the knee, and point B represents the foot. The length between O and A is the thigh, the length between A and B is the shank, and the length between O and B is our imaginary vector that closes our vector loop. Since our primary objective is to maintain a steady hopping motion, there is an airborne regime and a grounded regime. The airborne regime has only dimensional restraints meaning that it is only limited by its dimensions. For example, in this airborne regime the foot can only extend from the hip as far as the combined length of the thigh and the shank. The grounded regime has the dimensional constraint and one additional constraint between the foot and the hip. This constraint is due to the fact that the leg is mounted on a test stand and our assumption of the no slip condition at the foot. These two physical constraints force our model to maintain a constant horizontal distance between the foot and the hip when the leg is in contact with the ground. We are considering the possibility of maintaining these constraints throughout our airborne regime to improve jump reliability. These constraints are included per the request of our sponsors for realistic modeling. The leg will have material attached to the foot which justifies the no-slip condition and constant horizontal distance of the foot from the hip.

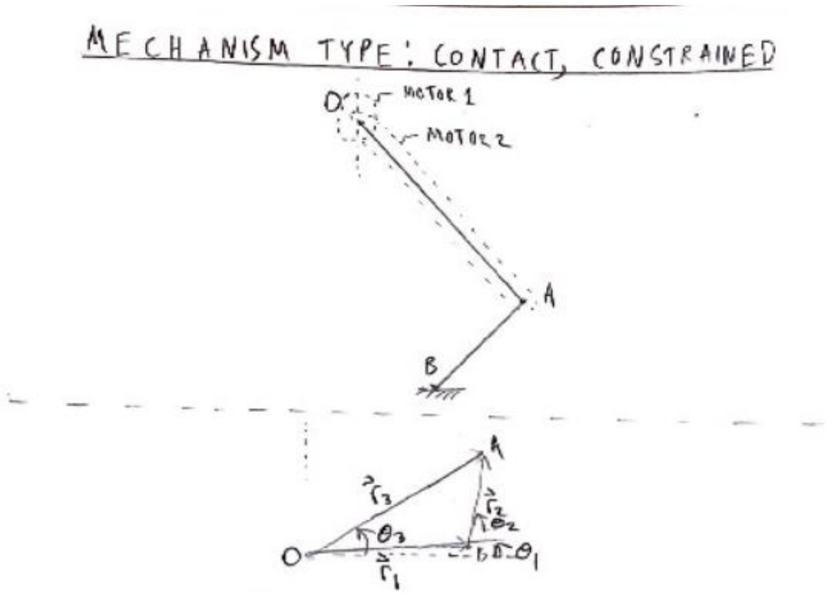


Figure 4.6 Robotic Leg in Contact with Ground – The bottom vector loop is a reorientation of the leg so that it is easiest to solve using trigonometric relations.

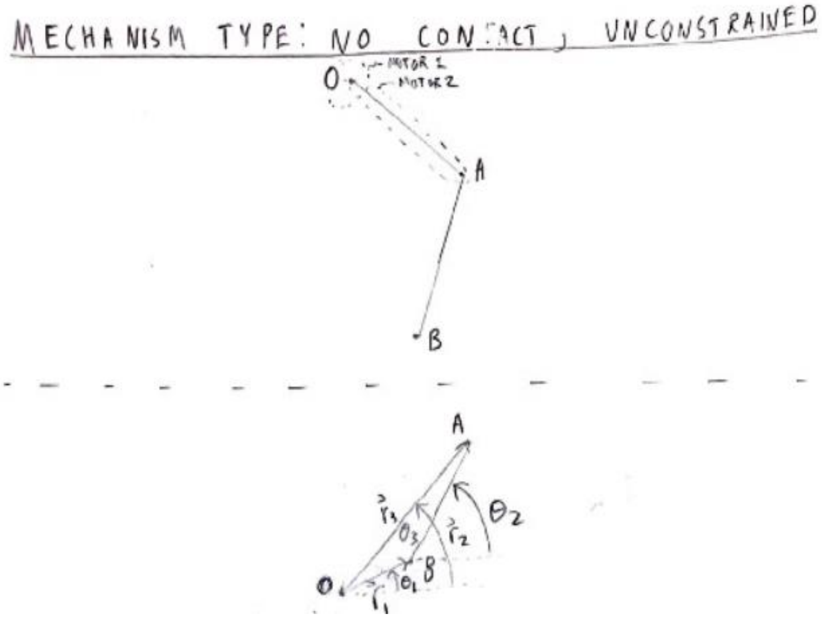


Figure 4.7 Robotic Leg with No Ground Contact – The bottom vector loop is a reorientation of the leg so that it is easiest to solve using trigonometric relations.

To control the jumping motion of the leg we will position the location of the foot using our controller. Each foot location has two unique physical solutions that are mirror images of each other, as seen in Figure 4.8. Because we will be effecting incremental changes in the leg position from our controller, we will almost instantly converge to the nearest solution. Since the program always converges to the closest solution, making small incremental changes will allow for us to retrieve a correct solution each time without jumping across the axis of reflection.

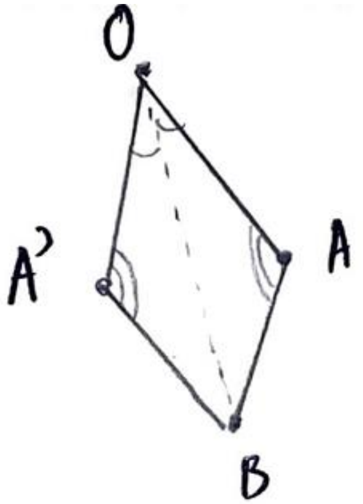


Figure 4.8 Sketch of two possible leg configurations for one specified foot location

4.3 Model Analysis

Before we design our controller, we first need to fully analyze our model to determine the potential pieces of our control loop. Our analysis includes both the forward and inverse kinematics. Both forward and inverse kinematics are explained in the following sub-sections. We are currently developing our kinetic models for the system. Our current position with the kinetic analysis is detailed below as well.

Before defining our methods for forward and inverse kinematics we will briefly define the two concepts. Forward kinematics uses the equations of motions for our robotic leg to compute the end effector location (in our case the foot) from some specified joint parameters. Inverse kinematics conversely are using a specified end effector location to determine some joint parameters. This is necessary for creating a path for our robotic foot to follow.

4.3.1 Forward Kinematics

To determine our forward kinematics (refer to background section 2.4), we utilized a vector loop method to define our system seen at the bottom of both Figure 4.6 and Figure 4.7. To determine the positional relationship between the hip, the knee, and the foot, we used trigonometric relations. We differentiated the positional equations once and twice with respect to time to get the velocity equations and acceleration equations, respectively. To numerically solve these systems of equations we found initial conditions for position, velocity, and acceleration by assuming known values for two angles and their respective time derivatives. We analyzed the system in two scenarios as discussed earlier: airborne and grounded. The difference between the two scenarios is that while the airborne scenario is constrained only by its physical dimensions, the grounded scenario is also constrained by its initial horizontal distance between the hip and the foot. This small change adds one more relational equation to the positional system, velocity system, and acceleration system. Once all these equations were determined along with the initial conditions, we were able to numerically solve the system. To numerically solve the kinematics in MATLAB we used Simulink. One of our block diagrams shown in Figure 4.9, consisting of input blocks, a function block, and four integrator blocks. The input is some assumed theta 3 angular acceleration which is integrated once to get angular velocity over time and twice to get angular position over time. All theta 3 information is input to the function block. Inside the function block the kinematic equations are organized into matrices and inverse matrix math is used to solve the system of equations. The output from the function block is the acceleration vector between the hip and the foot. This acceleration vector is integrated twice to get both the velocity and position information for this vector. From this information we can generate animations of the leg over time.

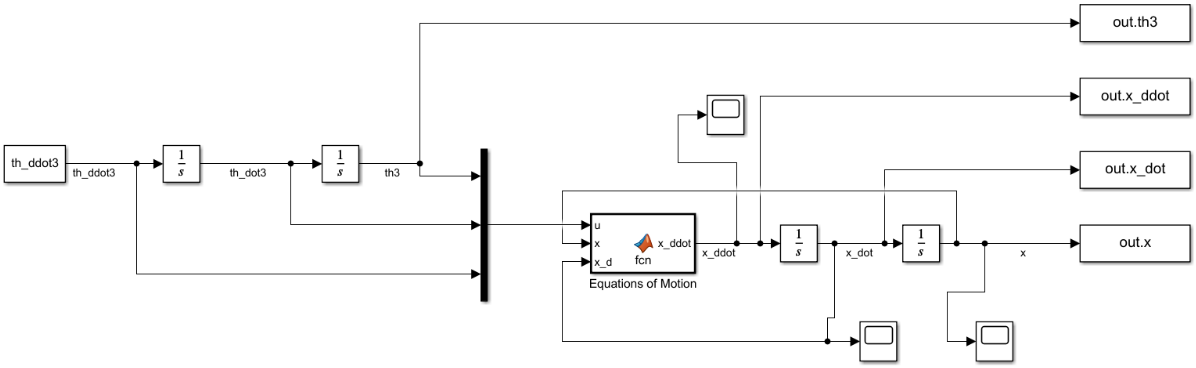


Figure 4.9 Simulink Block Diagram for Forward Kinematics – The input is angular acceleration of Θ_3 and output is position, velocity, and acceleration of the vector connecting the hip and the foot.

Attachment C includes the full code used to solve the system and generate animations of our system. Attachment D includes the hand calculations for setting up our system equations in MATLAB. We were able to confirm that our forward kinematic equations were correct using visual confirmation through animation. The animations were generated by setting initial conditions in the MATLAB workspace, passing the values into the Simulink model, and then passing the results from Simulink back to the workspace. Snapshots of these animations can be found in Figure 4.9 and Figure 4.10. The green line traces the motion of the foot and the red line traces the motion of the knee in Figure 4.9. The green line traces the hip, and the red line traces the knee in Figure 4.10. The motion of these animations is not necessarily a motion we would have our leg perform during testing but demonstrates the expected response from the given initial conditions.

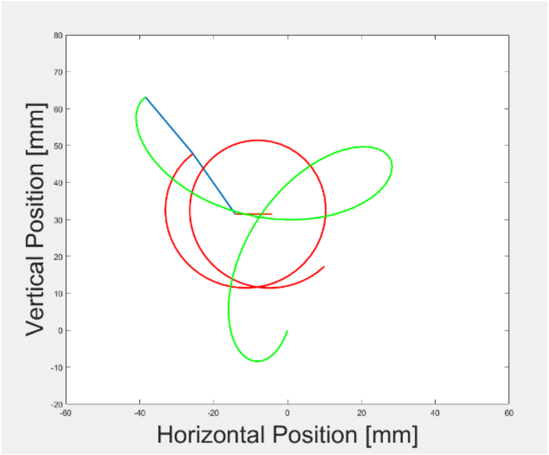


Figure 4.10 Airborne Kinematic Scenario Animation Snapshot

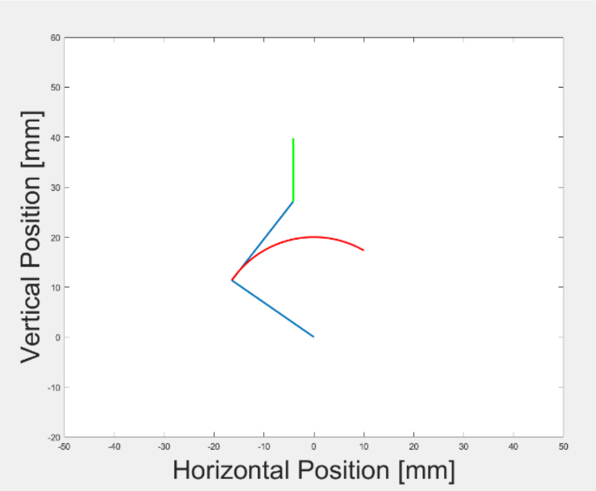


Figure 4.11 Grounded Kinematic Scenario Animation Snapshot

4.3.2 Inverse Kinematics

Inverse Kinematics will be used in our open loop controller. Our controller receives a desired position input, and inverse kinematics outputs the corresponding angles required to reach that position. Our Inverse Kinematics function file will achieve this by forming a Jacobian matrix with our existing equations of motions, and then applying the Newton Raphson method using this Jacobian matrix. The Newton Raphson method is an iterative technique where a delta guess is calculated on each pass. These delta guesses are in reference to the necessary angles to achieve desired leg geometry. A loop containing the Newton Raphson is ran as the MATLAB Function.

With the solved angles, our controller and motor block will then apply the torques our motors need to apply to reach the solved angles, see section 5.1.2.3 and 5.1.3.1 for more details. At this point our inverse kinematics code takes two inputs of length r_1 and angle θ_1, returning angles θ_2 and θ_3. These r_1 and θ_1 values define a vector representing the desired foot position of our model, shown in Figure 4.7 as vector OB. Our MATLAB code then runs an iterative solver to converge on the values of θ_2 and θ_3 which represent the angle of the thigh and the shank off a horizontal axis for the input foot position r_1 and θ_1.

We turned this code into a function file that constantly accepts new inputs and continuously moves the foot. We implemented a time out feature that ensures the code will not solve for angles θ_2 and θ_3 indefinitely if no solutions exist. Based on the performance of our solver, we set the timeout limit to 8 iteration. Therefore, the solver will break if more than 8 iterations are required to solve the required angles. Another consideration with this modeling is that there are always two possible solutions for θ_2 and θ_3, and we do not want our leg quickly cranking in the opposite direction. To prevent this, we will add code that ensures that the angles cannot change too quickly. Attachment E contains the hand calculations used to find our Jacobian matrix. Attachment F contains the code used to run the iterative Newton Raphson solver for the inverse kinematics.

4.3.2.1 Alternative Methods

There are many methods to modeling that we did not pursue. The Newton Raphson method is described in the background section 2.4. and this section will cover a few other methods for solving inverse kinematics: Cyclic Coordinate Descent (CCD), Forward and Backward Reaching Inverse Kinematics (FABRIK) and Follow the Leader (FTL).

CCD is an iterative optimization method. The position of each joint is optimized to get the end-effector as close to the desired position as possible. This process is completed in a loop until a solution is found or the solver reaches an iteration limit. This method is commonly used to solve more complicated inverse kinematics problems with large numbers of links. CCD rapidly approaches the desired position of the end effector by determining the location in each links range of motion that will place the end effector nearest to the desired point [24]. An example of how this method is implemented is shown below in Figure 4.12. Because this method is designed to minimize the number of iterations required to reach the solution, it often results in jerky motion and erratic discontinuities. This result is not acceptable for our application.

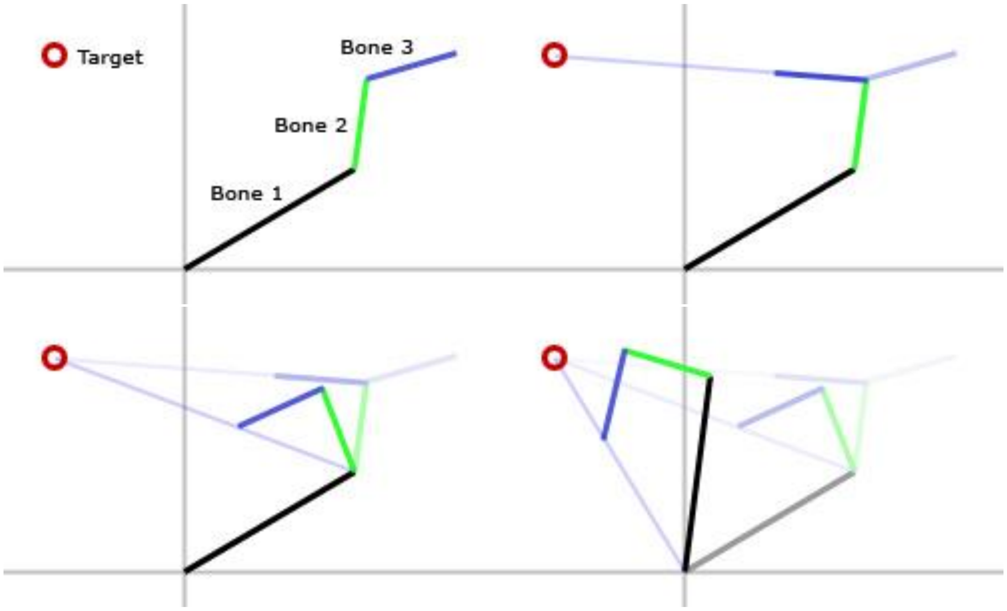


Figure 4.12 CCD Iteration Example [24]

FABRIK iterates through each joint and determines the solution position to reach a desired input end effector position. Like CCD, FABRIK starts with by moving the end effector close to the desired position and then iterating down to the base. Unlike CCD, FABRIK then iterates back up from the root to the end effector to complete one iteration loop. Each link has a set length that is used to determine the position of the links after each node is moved. This iteration loop is shown below in Figure 4.13. For example, in Figure 4.13b, the intersection of the line p'$_2$ – p'$_3$ and segment d$_2$ determines the new position of node p$_2$, p'$_2$ [25]. This method operates at a low computational cost and converges quickly. This was a strong contender for our inverse kinematics method, but our sponsors recommendation of the Newton Raphson method resulted in FABRIK as our second choice.

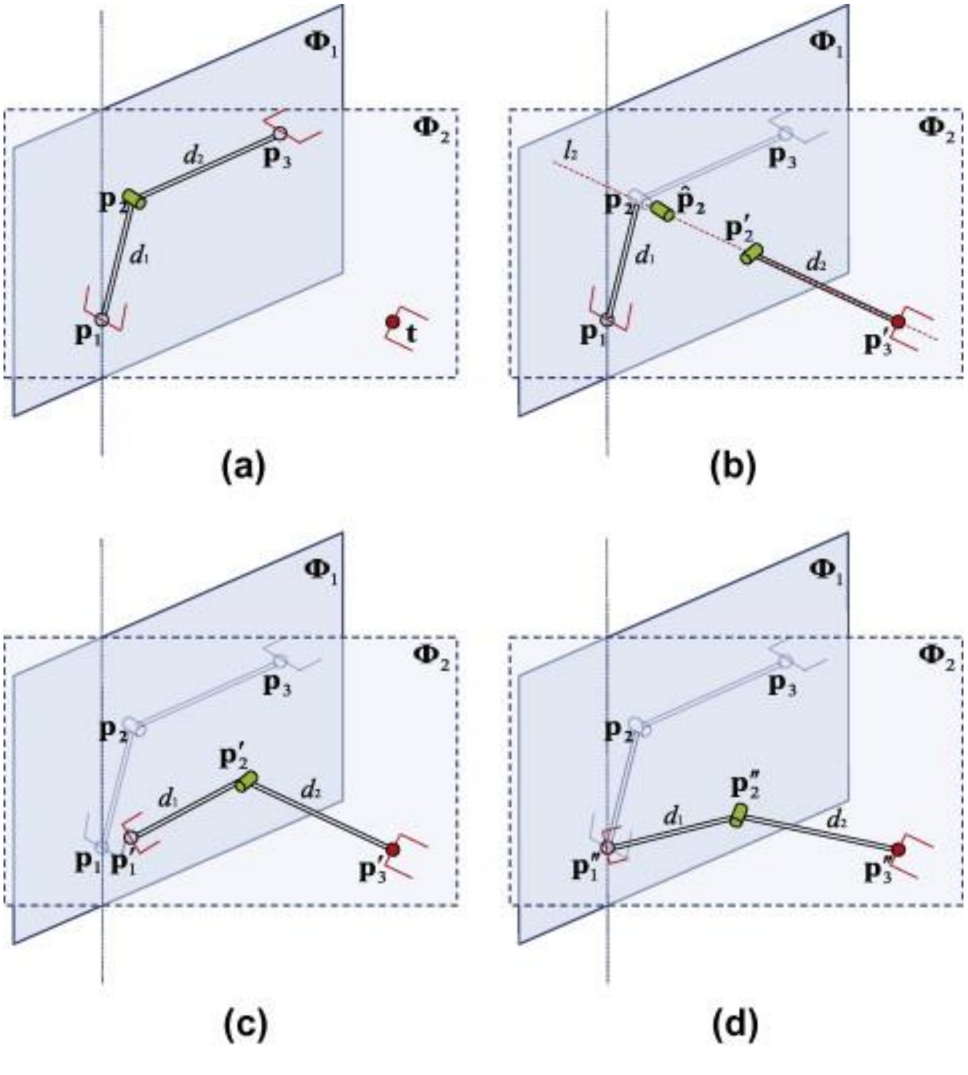
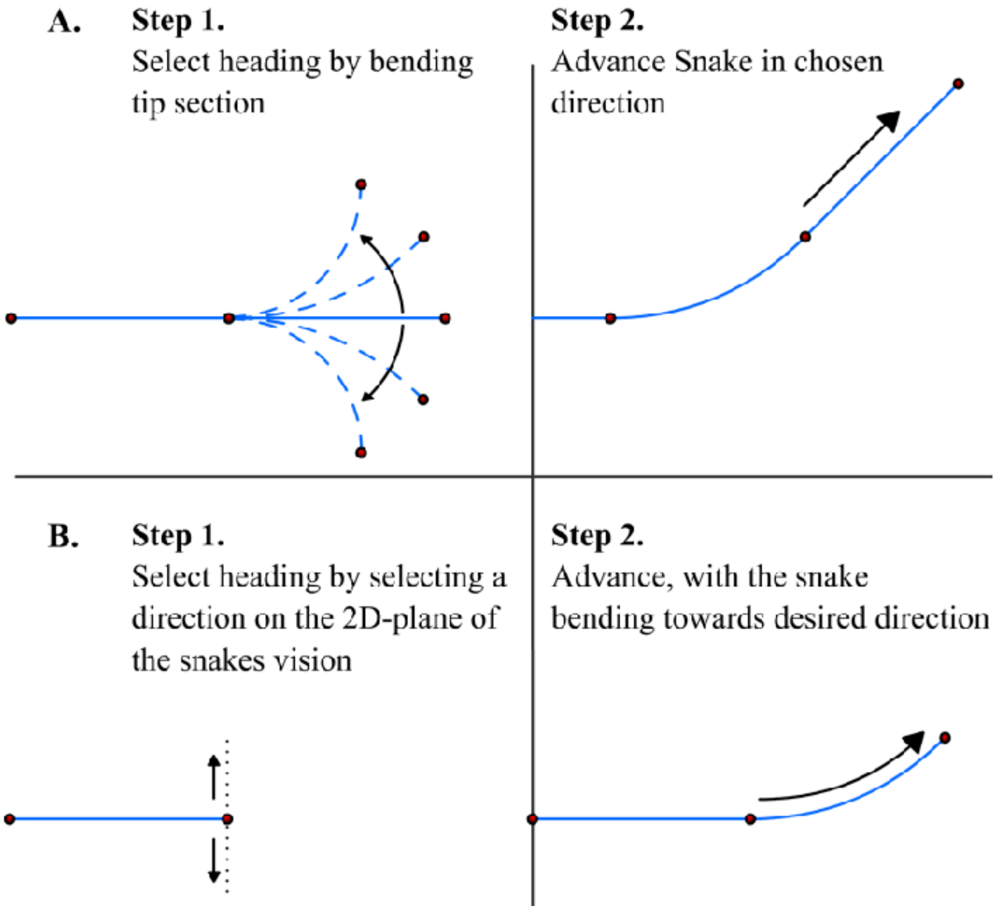


Figure 4.13 FABRIK Iteration Example [25]

FTL is designed for rope simulation and is often applied to manipulate kinematic chains. This method uses the end effector link to drive the positions of the other links in the system. It is comparable to pulling a rope. The rope follows the path that it is pulled along. This method cuts the rope into the sections where each section represents a link [26]. Because only the first link is moving and all the other links simply follow the leader, each link is considered as one degree of freedom. This greatly reduces the number of degrees of freedom for robots with many links. For example, a robot with 10 links that each have 3 degrees of freedom would have 30 degrees of freedom using conventional inverse kinematic methods. Using FTL, this system would only have 10 degrees of freedom. This is the main advantage of FTL. However, FTL does not iterate to its solution. Rather the leader chooses a direction depending on the immediate surroundings, shown in Figure 4.14. This results in a system that does not have precise control over each link and requires a much more complicated controller logic.

Figure 4.14 FTL Iteration Example [26]


A.
Step 1.
Select heading by bending tip section
Step 2.
Advance Snake in chosen direction
B.
Step 1.
Select heading by selecting a direction on the 2D-plane of the snakes vision
Step 2.
Advance, with the snake bending towards desired direction

4.3.3 Newton Raphson Code

The code for running the Inverse kinematics is, at this point, still a simple solver that requires changes in the code to change the initial conditions. As discussed in 4.2.2, eventually we will use forward kinematics to determine the inputs instead of manually coding the input values. However, the method will be largely the same. The first step in the Newton Raphson method is creating a Jacobian matrix using our positional equations of motion shown below in Figure 4.15.

$$J = \begin{bmatrix} \dfrac{\partial F_1}{\partial x_1} & \cdots & \dfrac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial F_m}{\partial x_1} & \cdots & \dfrac{\partial F_m}{\partial x_n} \end{bmatrix}. \qquad J(x,y) = \begin{bmatrix} -r_2 c_2 & c_3 r_3 \\ s_2 r_2 & -s_3 r_3 \end{bmatrix}$$

Figure 4.15 Jacobian Matrix Example – In this figure, the generic Jacobian is displayed on the left and the Jacobian for our system is displayed on the right.

The Jacobian is created from a vector function that is dependent on more than one variable. As shown on the left, each component of the vector function is partially differentiated with respect to each variable. For our system, we have a vector function for the two theta values we are interested in finding, Θ_2 and Θ_3. These angles represent the hip angle and shank angle, respectively. The length of the hip and shank are denoted as r_2 and r_3 respectively. C and S represent cosine and sine, respectively.

A Jacobian matrix is the partial derivative of a set of equations with multiple variables. As our model moves in a 2D space, it is represented by the 2X2 matrix shown above. A more detailed derivation of our Jacobian matrix can be found in Attachment E. Once this Jacobian matrix has been defined, we run an iterative process to solve for our two unknowns, θ_2 and θ_3. The solver stops running when the error is smaller than some set value, ε. This result of the solving process is shown below in Figure 4.16, where the solution is determined when the error is less than ε. The solver converges upon an answer for the two-unknown shown in the plot. The entirety of this code can be found in Attachment F.

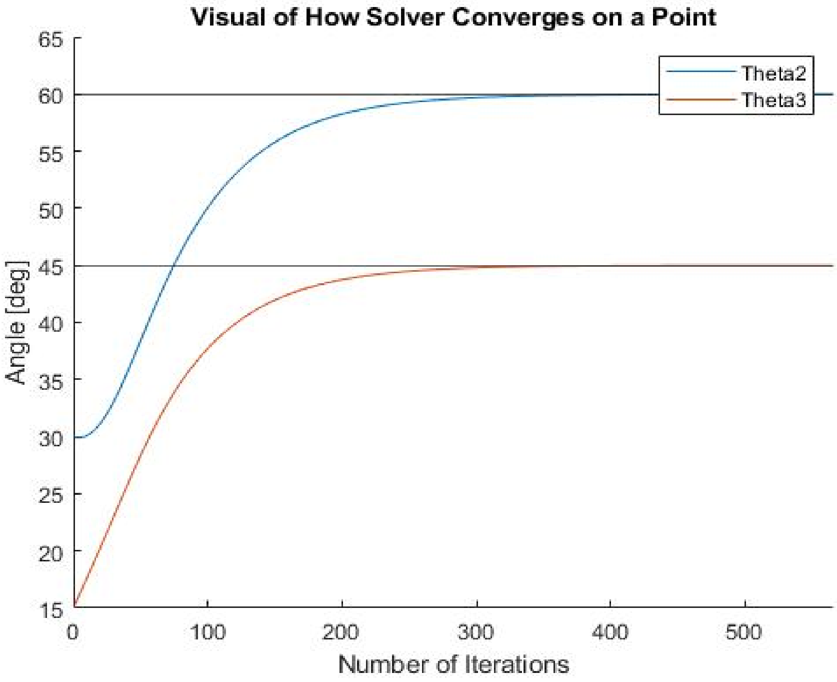


Figure 4.16 Convergence Plot

4.3.4 Newtonian Kinetics

The kinetic analysis includes inertial properties and forces. To solve our kinetic model, we utilized free body diagrams and kinetic diagrams to derive the governing equations of the system. The primary type of equation we used was the sum of moments an example taken from our hand calcs in Attachment D is shown below as Eq 1.

$$Eq\ 1{:}\ \sum M_O = M_O - m_D g(c_3 r_3 - 0.5 c_2 r_2)$$

Where M_O is the moment about the hip, m_D is the mass of the shank, g is gravity, c_3 and c_2 are cosine of the shank and thigh angles respectively, and r_3 and r_2 are the thigh and shank length respectively. This is the most useful approach to a system focused on rotational motion. During this analysis it is necessary to relate the mass accelerations to the angular accelerations, this is shown below in Eq 2.

$$Eq\ 2{:}\ \ddot{x}_D = \ddot{x}_O - \ddot{\theta}_3 s_3 r_3 - c_3 r_3 \dot{\theta}_3^2 + 0.5\ddot{\theta}_2 r_2 + 0.5 c_2 r_2 \dot{\theta}_2^2$$

Where $\ddot{x}_D$ is the horizontal acceleration of the shank mass, $\ddot{x}_O$ is the horizontal acceleration of the hip mass, θ_3 and θ_2 are the thigh and shank angles respectively. The matrix equations necessary to complete the kinetics is shown below in the derivation from Eq 3 to Eq 4.

$$Eq\ 3{:}\ f = M\ddot{\chi}$$

$$f = M(A\ddot{x} + a)$$

$$f = MA\ddot{x} + Ma$$

$$Eq\ 4{:}\ \ddot{x} = (MA)^{-1}(f - Ma)$$

Where f is 5x1 matrix of kinematic diagram equations, M is a 4x5 matrix made up of the free body diagram equations, and $\ddot{\chi}$ is made up of $(A\ddot{x} + a)$. To summarize the calculation process, we used FBDs and KDs to determine moment equations, used rigid body acceleration equations to relate mass accelerations to angular accelerations, and then we put it all in matrix form with our kinematic equations. The complete calculations can be seen in Attachment D. This allowed us to use Simulink and numerically solve the system. The Simulink program utilizes a function block that takes in all the constants of our system like mass or member lengths. This function block then completes the matrix math shown above. Currently we have had success with our airborne scenario using these methods. We have been able to create an animation that visually supports the effectiveness of our equations, completed in the code shown in Attachment G. A snapshot of the animation is shown below in Figure 4.18. The blue line traces the motion of the hip, the red line traces the motion of the knee, and the green line traces the motion of the foot. We can see that given our initial conditions the leg moves as expected by displaying inertial behavior and displaying the effects of gravity. However, with our grounded scenario we have found some issues. The animation proves that somewhere within our analysis we made a mistake. This can be seen in Figure 4.19 where a snapshot of the animation is displayed. The green line traces the motion of the hip and the red line traces the motion of the knee. From this animation we can see that the constraint holding the horizontal distance between the foot and the knee immediately begins to break. To mitigate this, we are continuing to search for errors and are discussing the process with Dr. Xing and Professor Refvem. For the full MATLAB code used to generate the solution to the kinetics see Attachment G. For the full hand calculations for setting up the solution to the kinetics see Attachment D.

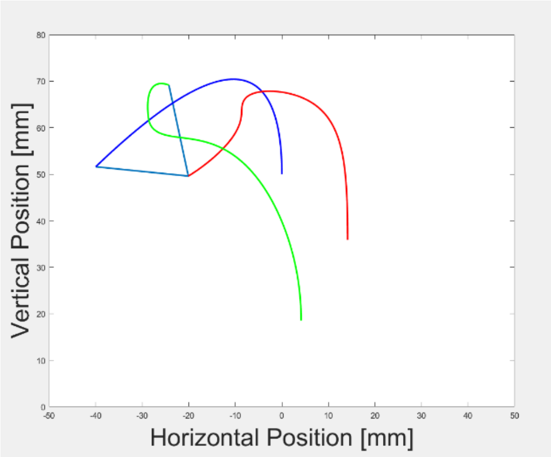


Figure 4.18 Airborne Kinetic Scenario Animation Snapshot

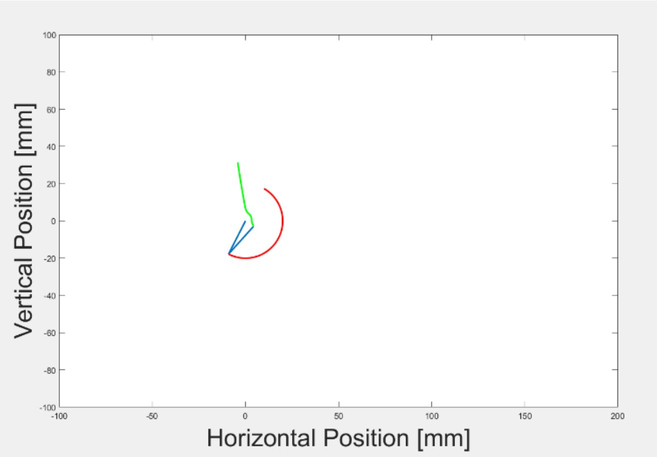


Figure 4.19 Grounded Kinetic Scenario Animation Snapshot

4.3.5 Lagrangian Mechanics

The work for this section is in Attachment H and is still being developed. We used Lagrangian Mechanics to verify the Kinetic model of our system. Because the results should be the same for both methods, we compared the resulting animations of the two methods to verify our mathematical models. The model used for both methods is shown below in Figure 4.20. We restricted the legs motion to vertical translation and rotation. The system is modeled as two-point masses with one at the hip and one at the knee, m1 and m2, respectively. This system is modeled so that the foot position is constant, due to the no-slip condition, and is represented by X in Figure 4.20. The test stand is annotated in red and is represented by the vertical black line. Defining the system in term of its degrees of freedom is the first step in applying Lagrangian Mechanics. We defined our system in terms of Θ_1 and Θ_2 , since these variables represent the degrees of freedom of the system for the grounded model. The position of the knee, hip, and foot are defined in terms of Θ_1 and Θ_2 , where Θ_1 and Θ_2 represent the thigh angle and shank angle, respectively.

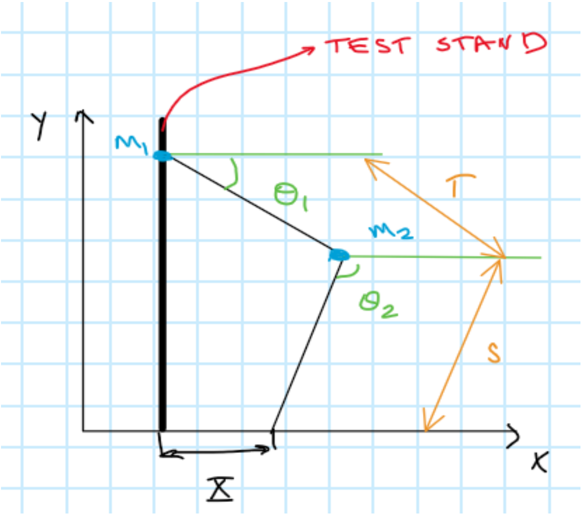


Figure 4.20 – System Model for the Robotic Leg

After we defined the system model, we constructed the Lagrangian. This is the second step in applying Lagrangian Mechanics. The Lagrangian itself is simple. It is represented by the difference in kinetic energy and potential energy. The Lagrangian is an expression in terms of energy that can easily be related to forces and accelerations by differentiation. Equation 5 shows the Lagrangian's general formulation and Equation 6 shows the Lagrangian for our model.

$$Eq\ 5{:}\quad L = T - U$$

$$Eq\ 6{:}\ L = m_1 TScos(\theta_1)\cos(\theta_2)\,\dot{\theta_1}\dot{\theta_2} + \frac{1}{2}T^2\dot{\theta_1}^2(m_1\cos^2(\theta_1) + m_2\sin^2(\theta_1)) + \frac{1}{2}(m_1 + m_2)S^2\cos^2(\theta_2)\,\dot{\theta_2}^2 - m_1 gTsin(\theta_1) - (m_1 + m_2)gSsin(\theta_2)$$

With the Lagrangian constructed, we now needed to define the applied torque and position constraint to the system. As discussed in the background section 2.6, the torque is included using Rayleigh's Dissipation Equation and the position constraint is using Lagrangian Multipliers. The definitions of each equation are shown in equation 7 and equation 8, respectively. In the following equations, *f* represents the dissipative force, β represents the constraint equation, *q* represents a particular degree of freedom, and $\dot{q}$ represents the time derivative of the same degree of freedom.

$$Eq\ 7{:}\ Rayleigh's\ Dissipation\ General\ Form = \frac{\partial f}{\partial \dot{q}}$$

$$Eq\ 8{:}\ Lagrangian\ Multiplier\ General\ Form = \ \lambda\frac{d\beta}{d\dot{q}}\frac{d\beta}{dq}$$

We then needed to apply these definitions to our system. The torque is simply replaced into Rayleigh's Dissipation Equation and is shown in equation 9. However, the position constraint must be derived from the system model before it is included in the Lagrangian multiplier. Equation 10 shows the derivation of our position constraint and equation 11 shows the Lagrangian multiplier for our system. In the following equations, τ represents the torque applied at a particular degree of freedom and X represents the constant foot position. The time derivative of X is set equal to β because the positional constraint is holonomic, meaning the constraint equation must equal zero. This is true for the velocity of the foot.

$$Eq\ 9\text{:}\ \frac{\partial f}{\partial \dot{q}} = \tau$$

$$Eq\ 10\text{:}\ X = Tcos(t\theta_1) + Scos(\theta_2) = constant$$

$$\dot{X} = -Tsin(\theta_1)\dot{\theta}_1 - S\sin(\theta_2)\dot{\theta}_2 = 0$$

$$Tsin(\theta_1)\dot{\theta}_1 + S\sin(\theta_2)\dot{\theta}_2 = 0 = \beta$$

$$\beta = Tsin(\theta_1)\dot{\theta}_1 + S\sin(\theta_2)\dot{\theta}_2$$

$$Eq\ 11\text{:}\ \lambda\frac{d\beta}{d\dot{q}}\frac{d\beta}{dq} = \lambda\frac{d(\,Tsin(\theta_1)\dot{\theta}_1 + S\sin(\theta_2)\dot{\theta}_2)}{d\dot{q}}\frac{d(\,Tsin(\theta_1)\dot{\theta}_1 + S\sin(\theta_2)\dot{\theta}_2)}{dq}$$

With a defined Lagrangian, Rayleigh’s Dissipation function, and Lagrange Multiplier, Lagrangian Mechanics is now used to determine the equations of motion that define the system. We first determined the equations of motion for an applied torque without the foot constraint, show in equation 12. We solved Then we included the foot constraints to the system, using equation 13. We are still currently working to derive the constrained model.

$$Eq\ 12\text{:}\ \frac{\partial L}{\partial q} - \frac{dL}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = \frac{\partial f}{\partial \dot{q}}$$

$$Eq\ 13\text{:}\ \frac{\partial L}{\partial q} - \frac{dL}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = \frac{\partial f}{\partial \dot{q}} + \lambda\frac{d\beta}{d\dot{q}}\frac{d\beta}{dq}$$

4.3.6 Summary of Control

We will in this section we will break down our current code down to its most basic function for sake of understanding.

The Newtonian model of our leg establish the equations of motion that govern our entire system. Within MATLAB we code these equations so that they except input values of angles and return the resultant geometry. Lagrangian Mechanics fulfill the same purpose and are used as an additional check to the Newtonian method. Both sets of equations will be backed up and tested virtually by using animations. These animations are a representation of how the physical leg will react and move when given certain inputs. If our equations are free of error, we should be able to observe recognizable leg movement.

The Newton Raphson section of our controller involves inverse kinematics. This performs the opposite function of the Lagrangian and Newtonian systems. While Lagrangian and Newtonian are fed angle values and compute leg geometry, Newton Raphson takes leg geometry and computes the necessary angles to achieve this. This is done iteratively by guessing and redefining set parameters until values fall within an acceptable range. This step is necessary for leg pathing so that we may convert a series of desired foot positions into angles for our motors. This code is verified through the observation and plotting of angle convergence.

4.3.7 Outstanding Challenges, Concerns, Unknowns

Our code continues to be a work in progress and there are still several high-level issues to address. Though our model for inverse kinematics is running smoothly for single input positions, we need to adapt it to except multiple inputs, as well as constrain how quickly our data points can change. The challenge will be to link the forward and inverse kinematic solvers and account for the possibility of two physical solutions as pictured in Figure 4.8. Within our forward kinematics, we have also encountered one bug that is affecting the quality of our produced animation. There is an ongoing effort to locate and fix that bug, and we have consulted with our sponsors on how to fix the animation.

Once we get both sets of code running smoothly, the next step will be implementing a control system and connecting our MATLAB models to the physical leg. As none of us have much experience with this, there is some concern with the adaptability of our code. This will be an ongoing process with our sponsors, as well as some possible consultation with professors at Cal Poly who have controls lab experience. It is possible that we will have to remake and adapt our code to fit this purpose.

There are some minimal dangers associated with running motors in a lab setting. It is possible that our code could overload motors. To prevent over-current issues, we will include fuses. To prevent over-voltage issues, we will add voltage regulators. There is also some potential for finger pinching when running the motors and leg on the stand. To prevent this, we will keep hands away from moving parts. The precautions for our testing will be like those taken for the Motomatic lab in ME422. With proper forethought, this will be a low-risk project. For more information regarding how we mitigate these risks, refer to Attachment I which discusses design risk assessments.

4.3.8 Modeling Decision Making

There was relatively little mobility in the direction that we could take for our design. All physical modeling was done by a previous senior project group and we rely on their judgement for the ideation process there. The model for forward kinematics was the one presented by our sponsors and was their requested style. The boundary conditions for our leg were also provided to us and defined by our sponsors. Within Inverse kinematics we used the Newton Raphson method, this for the reasons outlined in 4.3.2 Inverse Kinematics. There were additional possibilities to increase the precision seen in the Newton Raphson method with FABRIK. However, after creating our model, we determined that we were sufficiently precise. This is a dynamic project, and the possibility remains that we will have to go back and rethink some of the methods that we have used to this point. Our goal is to keep the connection between different aspects of our design separate, so that these changes will not break the whole system.

5.0 Final Design

Our final design for our controller and our control loop are discussed below. Design description and functionality of each subsystem and component are discussed. The section details how each portion functions, explains what it is, and why we think it will work. FMEA updates covers all our additions to FMEA. Cost Analysis details the few costs our senior project team is incurring. Remaining Concerns addresses the major problems that lay ahead which we have not yet solved. Our design has changed from the previous chapters in that we are no longer considering airborne motion of the leg. We will restrict our testing and modeling to only the condition foot in contact with ground.

5.1 Design Description, Functionality, Justification

At this stage of the design process, we have completed the individual components needed to make up the control loop for the robotic leg. We are planning to implement a PID controller tuned with either CRONE or RlTool. These tools will also allow us to linearize our plant, allowing for effective use of the PID. Our control loop is made up of an input height for hip position followed by inverse kinematics that convert the desired positional height into angle values. The angle values are subtracted by the feedback angles that were found from the previous iteration in the summing junction. The adjusted angle output from the summing junction is sent into the PID controller which outputs the necessary voltage to actuate our leg. The encoders in the motors output the robot angles that are used in the feedback loop. These encoder angles are also the current angle of the thigh and shank with which we can calculate positional data using the forward kinematics. Figure 5.1 shows the flow of the control loop as discussed above.

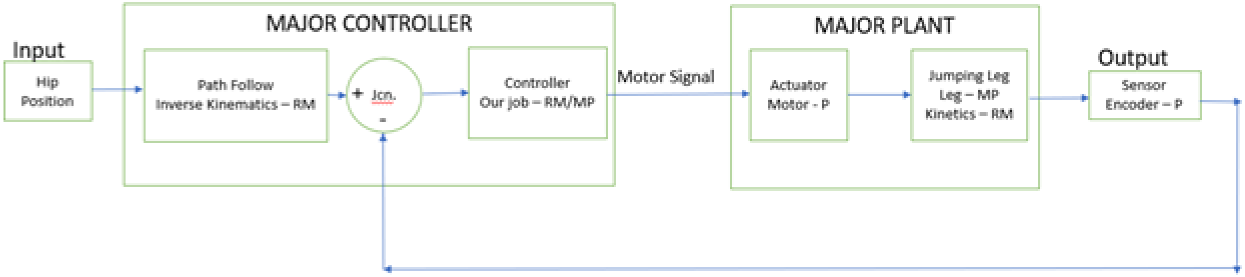


Figure 5.1 Final design for robot leg control loop

As our design stands, we can effectively simulate and tune the control of our leg when the foot is in contact with the ground.

5.1.1 Input Hip Position

The input portion of our leg will determine the motion and the path that the leg follows. For the leg to move as we desire a path of the hip height will be predetermined and calculated. These values will be fed into our control loop through the Input Hip Position block.

5.1.2 Major Controller

This section is the most important part of our design. This subsystem uses Inverse kinematics, a feedback, and a PID controller to modulate our leg. For the real-world application this is the primary system that is applicable. It takes in the input, changes it to the right data type and compares the data type to the current position of the leg producing a new adjusted data. This new data is sent through the controller to accurate produce the necessary voltage to move the motors accordingly.

5.1.2.1 Inverse Kinematics

This portion of the control loop takes in desired hip height values and converts them into thigh and shank angles. These angles are the angles necessary for the leg to position the hip at the correct location. The Inverse Kinematics utilizes the Jacobian Matrix and the Newton Ra Waiting to type the rest of this to

Inverse kinematics allow for precise pathing and positional control. As discussed in 4.3.2 kinematics takes some desired foot position and returns the angles needed to achieve this position. By positioning the inverse kinematics in the beginning of our control loop we are directly converting desired foot coordinates into the unit we will be using through much of the control loop, motor angles.

In order to ensure that our code was working properly initially we fed an arbitrary set of known angles and corresponding foot position into the code and confirmed that our code returned the correct angles. This is outlined in 4.3.2 and involved plotting each iteration converging upon a point. Since then, the code has been dramatically improved lowering the number of necessary iterations to 4-5 from 600+. We also found that it was necessary to convert the code that we had into a compact function that could called from Simulink. Once the code was converted to a function file, we then had to make sure that it functioned the same as the original code. We confirmed this the same way that we did the original, simply plotting the convergence and insuring proper output angles for a known set.

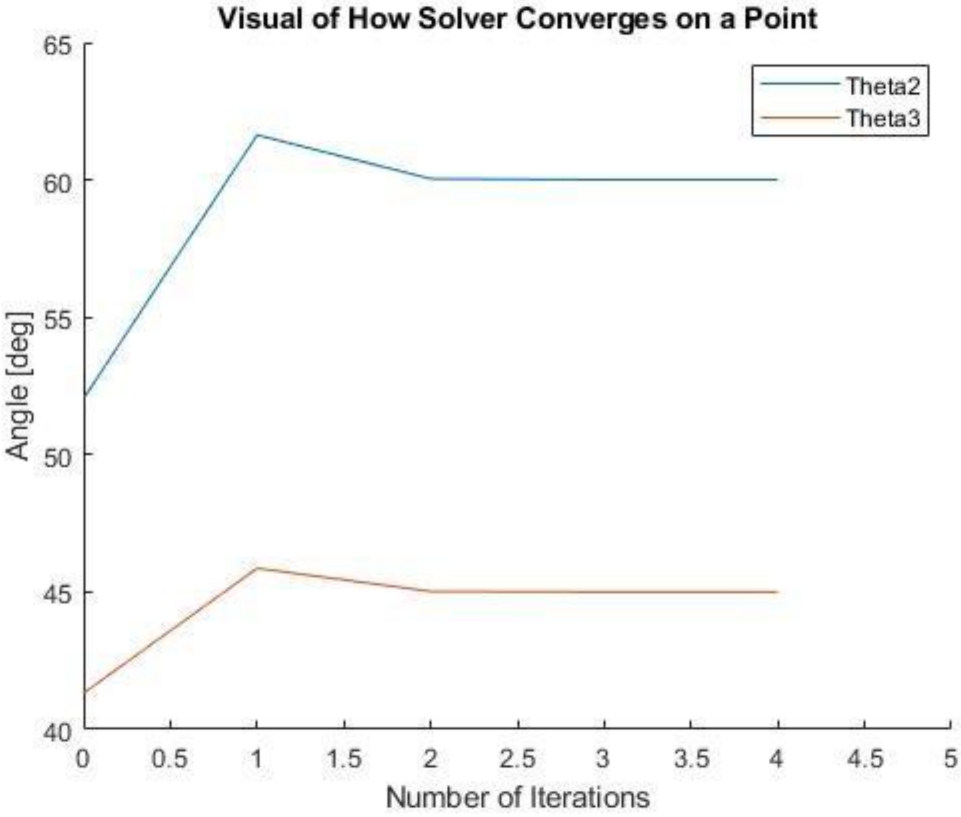


Figure 5.2

5.1.2.2 Summing Junction and Feedback Loop

The summing junction is an essential piece of the control loop. In the case of our loop, we are using a negative feedback controller. This is how we are able to compare our data with desired results to determine how hard our controller should work to correct our signal. The feedback loop functions as an error signal, and the summing junction allows us to process this error signal.

5.1.2.3 PID Controller

The controller in our control loop has yet to be implemented and will be the final piece in our full control loop. We have two possible options that we are exploring. The first is a CRONE controller which is a fractional order controller available for select research groups. This is a robust controller with proven results that would be a solid asset to our project and team. The issue lies with poor communication with CRONE company and difficulties securing the software. To further our attempt to get access to CRONE software we have asked our sponsors to reach out to CRONE in the hope that they will be more responsive to them.

The second option that we plan to pursue unless progress is made soon with the CRONE controller is to design a PID controller and use rlTool to tune our response. This would also require that we linearize our system. The PID options uses tools and methods that our team is practiced and comfortable with and seems a solid route if applicable to the final loop design.

Controllers are used to compare a desired response with a given input. The controller provides some sort of correction to keep the response within a desired range. We can confirm a function controller by observing simulated motor responses with a scope in Simulink. This will also be necessary for the tuning process.

5.1.3 Major Plant

This section is primarily for testing our controller and tuning it to get the best results. This subsystem is made up of an Actuator block and a Leg and Kinetics block. The Actuator block will receive the voltage from the controller and model the behavior of the real-world motors that we have selected. The Leg and Kinetics block will receive the applied torque from the Actuator block and simulate the leg's reaction and motion to that torque outputting the shank and thigh angular data.

5.1.3.1 Actuator/Motor

This component is merely a gain block in our control loop. The reason why this block has been so simplified is the fact that the motors we are using in the real world come with a built-in controller minimizing our modeling work. The gain for this block will be determined by the motor spec sheet. We are currently working on sourcing the funding for an additional motor, and once one is produced, we should know the gain values that we need.

5.1.3.2 Leg and Kinetics

This component constitutes the mass properties and the motion behavior due to applied torques to the leg. This component is modeling the real-world version of the leg in MATLAB. Our design utilizes the MATLAB add-on, Simscape Multibody, to import a CAD model of the robotic leg that a previous senior design project team designed. The add-on takes the mass properties and the physical dimensions from the CAD file and applies the torque to the body to simulate the robotic leg motion. From this simulation we can get the thigh and shank angular data. We are verifying this component by completing Lagrangian and Newtonian solutions to the kinetics as well. These solutions work the same as the Multibody modeling but will have to be slightly more simplified models utilizing lumped masses. Since they are simplified, they will be perfect for checking our work but are not sufficient for basing our controller tuning on.

5.1.4 Sensor Output

The sensors are a real-world component. They represent the motor encoders and will return the angle data for the leg. However, in our mathematical model controller the output of our control loop is the angle data for the leg. This is then directly used as feedback in the control system to calculate the error signal and continue controlling the leg.

5.2 Changes since CDR

While the general layout of our control loop has remained consistent there have been several changes since we presented our CDR. The functionalities of these changes are outlined concretely in Section 6 but will be identified in this section. As we started with a very basic controller structure, most changes were in adding individual components.

Our input section has been adjusted to house two simple function blocks. One is to convert our input path into polar coordinates, and the other is our inverse kinematics. We reframed our understanding of where inverse kinematics should lie from the controller section to the input section. This is because inverse kinematics is essentially still a conversion of our input path and not functional for the control portion of the model. Inverse kinematics is the final processing step before the controller section where we take desired foot position and convert to θ values for our hip and knee.

Our controller section has also been refined. We include feedback control and feedback linearization and remove the summing junction rather than a PID controller. Error is still considered and refined but a summing junction was not necessary in our final design. Feedback control and feedback linearization function to linearize our plant and control its outputs. We compared the same sets of data but with a new approach as outlined in Section 6.

Our major plant has been updated to use a Lagrangian model of our plant. We kept our motors as a simple gain that will be adjusted before implementation on a physical model. As this controller has not been implemented on hardware, we have also removed the sensors from our control loop. This makes our plant outputs our assumed read values from the encoder. The sensors are still an important piece of the control loop and will need to be reinstated when future groups transition to a physical model.

6.0 Implementation

The culmination of our project was in creating a controller model to simulate hip height control in MATLAB using Simulink. Our controller features various components that will continue to be tuned throughout the overarching research project to meet jump height requirements. The components of our controller include inverse kinematics, feedback linearization, a motor gain, a plant model, and a feedback loop discussed previously in the final design section. This section will discuss the assembly of these components.

The first component of the controller is the hip position input shown on the far left of Figure 6.1. A user will input the desired hip position in cartesian coordinates. These coordinates are then passed through the Cart2Sphere function. This function converts the cartesian coordinates into polar coordinates. The two outputs are muxed together into the desired hip position vector and passed into the Inverse Kinematics function. The Inverse Kinematics outputs the corresponding angles the joints of the leg need to produce to achieve a designated hip height. To calculate these angles, we used the system properties, current leg position, and desired leg position. The system properties are the physical parameters of the system such as thigh length and mass of the hip motor. The derivative, and second derivative of the output angles are taken from the inverse kinematics output to determine the angular velocity and acceleration needed from our motors. All the corresponding angle data is muxed together and passed to the feedback linearization block as the desired values of θ, $\dot{\theta}$, and $\ddot{\theta}$. These operations are pictured below in Figure 6.1.

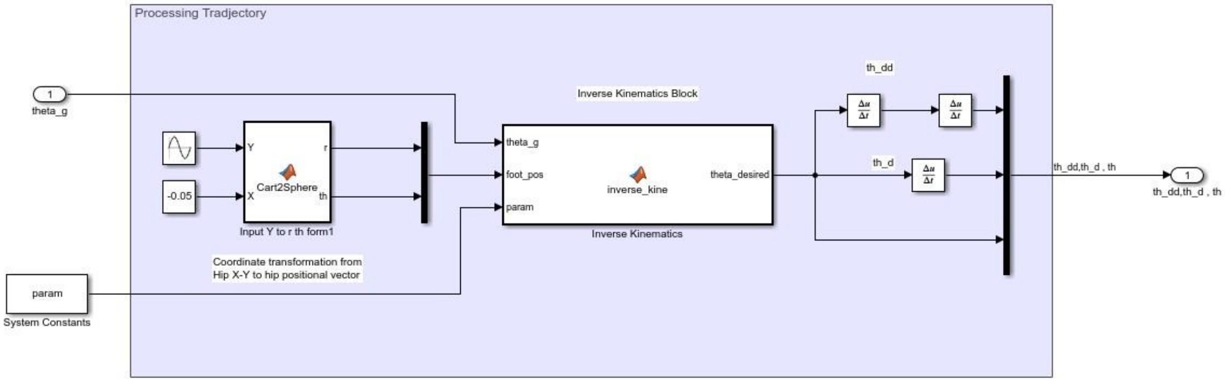


Figure 6.1 Input Section

After the signal is output by the input section, the feedback linearization section determines the torque required to move the leg into the desired position. Feedback control is used to process this data and limit error. Using the desired values of theta from the input section, current motor position, and system parameters, the feedback control function determines the angular velocity required to move the leg as a function of state. In this case, state is the difference between the desired and actual angular position and velocity. The feedback controller outputs the required angular velocity to properly control the leg and then passes this data to the feedback linearization function. This function uses the output from the feedback controller, current motor position, and system parameters to determine the required torque the motors must produce. All of these steps are shown Figure 6.2.

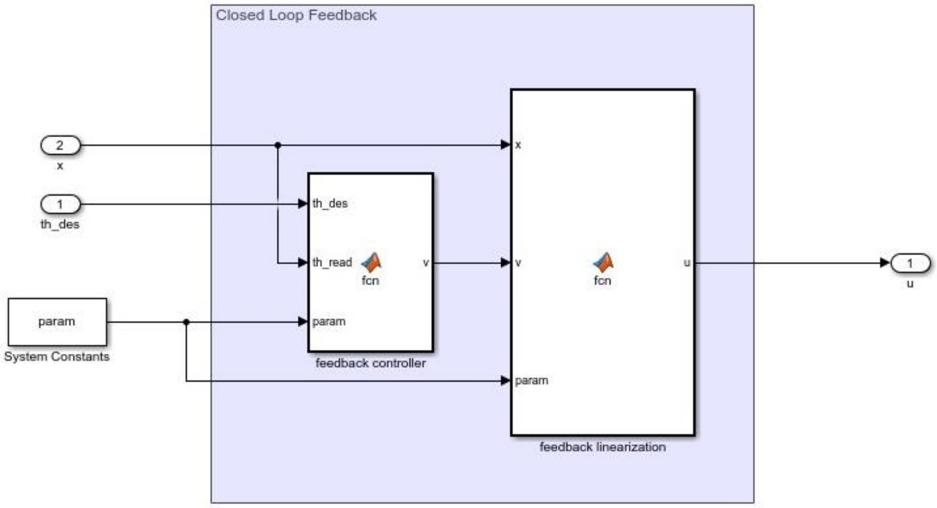


Figure 6.2 Feedback Linearization Section

After the required torque is output by the feedback linearization function, the plant model can simulate the system. We modeled our motor gains as simple gains because they contain built in controllers that have been finely tuned to output the exact torque requested. This assumption is sufficient for our simulated model and will be addressed in the conclusion and recommendations section. The motor gain is passed into the Langrangian model which represents our physical system. We designed our controller so that the model could be swapped out with another Mathematical model if the user desires. The only requirement is that the model accepts torque inputs and outputs the angular position and velocity of the leg. These output values are fed back into in the input section and feedback linearization block to control the motor.

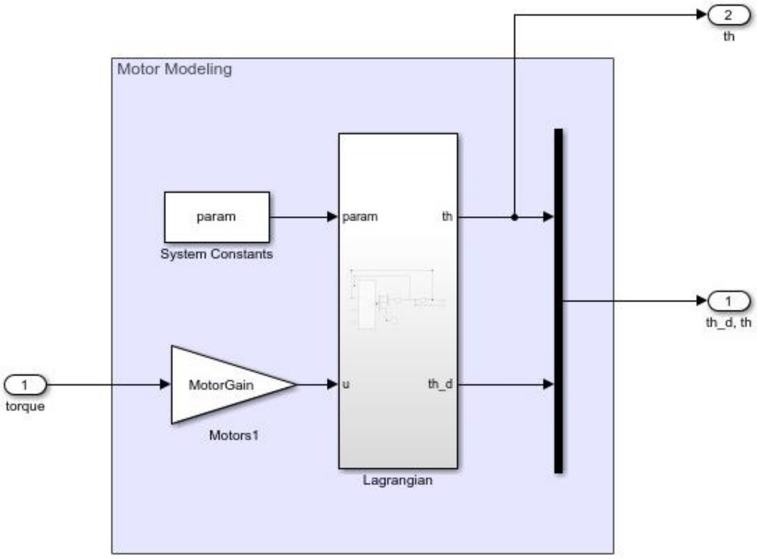


Figure 6.3 Plant Model Section

We combined the above-mentioned sections to create our controller, shown below in Figure 6.4. The inputs and outputs of each section are labeled to demonstrate where each data set is employed. We included the “to workspace block” as well as the “Encoder” Output block to show the fully functioning controller. This model completed an iteration in 2.63 milliseconds on average. This performance will be discussed further in the Design Verification Section.

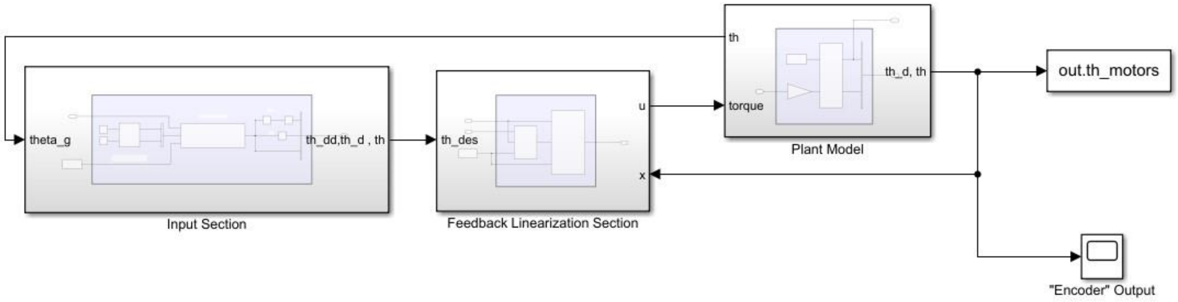


Figure 6.4 Assembled Controller

As discussed above, the controller is designed so that the user can input any model into the controller if it accepts torque as an input and outputs the resultant angular data. Previous iterations on the Simscape multibody model had created a 2D virtual model of the leg using revolve joints and geometry blocks. The final iteration uses a prismatic joint to place the final constraints on the robotic leg. The prismatic joint prevents the hip from moving in the horizontal direction and only allows movement in the Y direction. The prismatic joint sees the rotational movement from the revolve joint and converts this movement to translational. Joints blocks serve as the main measurement tool for this model. From the joint blocks, we can track angular position and velocity. Additionally, any inputted torque is applied directly to the joints, which enabled us to use it in our controller.

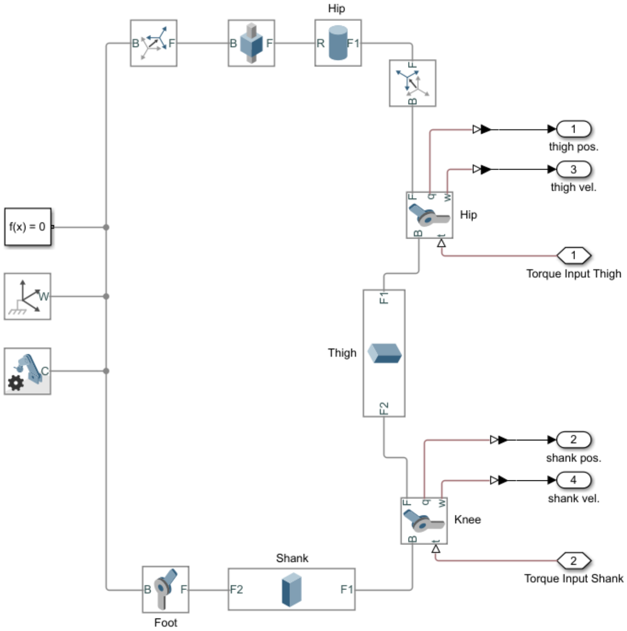
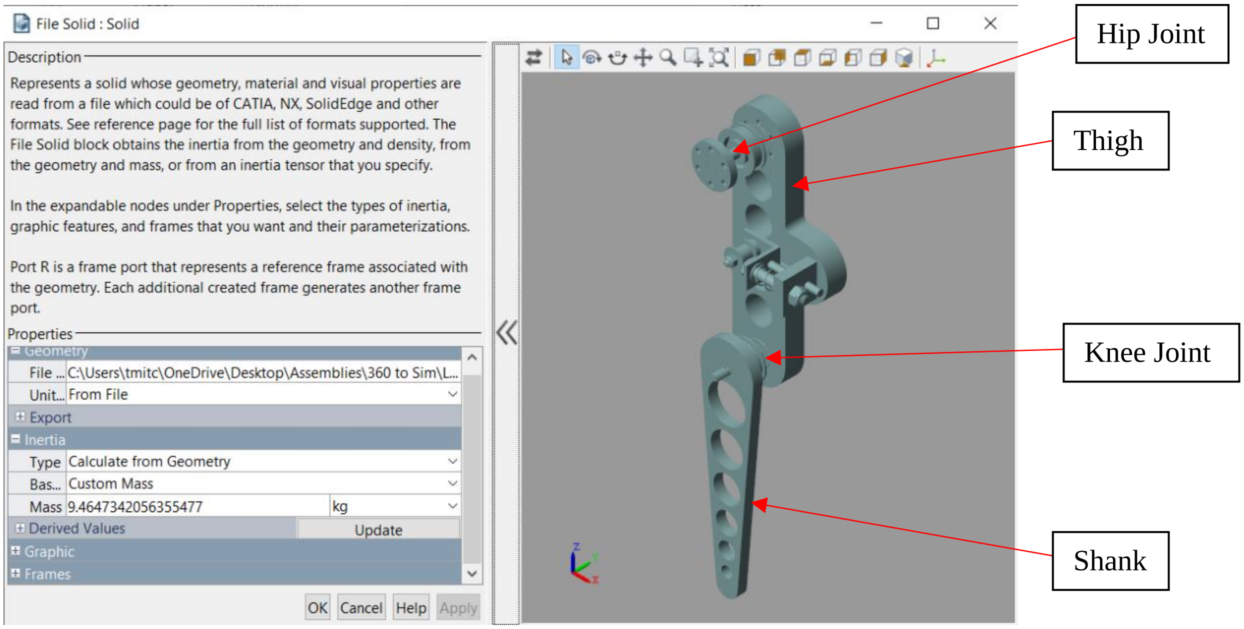


Figure 6.5 Multibody Logic Diagram

The use of step files allows us to easy adapt our Multibody model if the design of the robotic leg remains somewhat similar. If modifications need to be made to the CAD files, the part only needs to be redownloaded as a step file and updated in the Simulink file as seen in Figure 6.6. Additionally, if the assembly is made in SolidWorks and not Fusion 360, then a Simulink Multibody Link Plug-in can be used to import a SolidWorks .xml file into Multibody. The Multibody Plug-in can be downloaded from the MathWorks site and installed using the Help Center instruction page “Install the Multibody Link Plug-in". The advantage of using this Plug-in is that an .xml file will capture all the assembly's dynamics and eliminate the need to connect each solid part with joints.


File Solid : Solid

Description

Represents a solid whose geometry, material and visual properties are read from a file which could be of CATIA, NX, SolidEdge and other formats. See reference page for the full list of formats supported. The File Solid block obtains the inertia from the geometry and density, from the geometry and mass, or from an inertia tensor that you specify.

In the expandable nodes under Properties, select the types of inertia, graphic features, and frames that you want and their parameterizations.

Port R is a frame port that represents a reference frame associated with the geometry. Each additional created frame generates another frame port.

Properties

Geometry
- File ... C:\Users\tmitc\OneDrive\Desktop\Assemblies\360 to Sim\L...
- Unit... From File

Export

Inertia
- Type: Calculate from Geometry
- Bas... Custom Mass
- Mass: 9.4647342056355477 kg

Derived Values — Update

Graphic

Frames

Figure 6.6 Inserting CAD files into Simulink Multibody

The feedback loop is shown connecting to the Negative Feedback junction in Figure 6.1. The output from our model is passed into this junction to calculate the error signal. However, our model is dependent on sine and cosine functions, making it non-linear. Because of this, feedback linearization is also to be included in our controller so that the PID controller can work properly. This will allow the controller to filter out the non-linear terms in the equations of motion while still accurately controlling the motors. This will be an ongoing project for the research group and is not included in our controller yet. It will be placed in between the plant output and the Negative Feedback junction when completed. For details on the process of developing the Feedback Linearization equations, see Section 5.1.

We have run into some troubles regarding the continuation of our Manufacturing process. The first and most pressing concern is that the model made by the previous senior project group is not up to the standards of the sponsors. This leads to a few meaningful problems. The first is that we need accurate and final conditions for the leg model to properly design a controller for the quadrupedal robot. To combat this, we are generalizing as many aspects as possible so that when a final leg design is decided upon, updating our control loop should be a quick process.

7.0 Design Verification

This section will cover what specifications we will test as we continue developing our controller. Our specifications are realistic modeling, position control, jumping leg design, optimized code, documentation, and readable code. We discuss each specification, respectively. See Attachment L for Design Verification Plan (DVP).

7.1 Test Procedures

Test procedures are the primary tool to prove to our sponsors that the documentation and manufacturing of this project yield a successful controller for the robotic leg. Each test procedure will be specified along with an overview and details of the data collection methods.

7.1.1 Model Verification

The purpose of this test was to verify that our three virtual models of the leg accurately reflected the real-world conditions. Because we used three different methods to model the SDOF leg, we used this test to validate the performance of each model. The three models we tested were the Newtonian, Lagrangian, and Multibody models that evaluated our virtual plant. The model used for the Lagrangian and Newtonian method were the same, with a lumped mass at the hip and massless linkages. Because of this the results from these two methods were expected to be identical. The Multi-Body model was expected to have a similar result, but because it included precalculated mass and inertial properties we did not expect it to be identical.

The only equipment needed to run this test were a PC and the MATLAB software. The PC needed to be capable of running complex simulations for at least 10 seconds. Older computers with less available RAM may not have been able to run this test. MATLAB 2020 needed to be installed on the computer with the Simscape Multibody package included. There were no hazards to the individual that needed to be considered, and therefore no personal protective equipment was purchased.

The procedure for this test included first opening MATLAB and downloading the necessary .mlx and .slx files that included the Newtonian, Lagrangian, and Multibody models. The next step was to run all three models with the same initial conditions. The initial torque and angle inputs of the hip and knee needed to remain constant for each model. Next the output of each model needed to be used to generate plots and animations from the data. These figures were the primary tools for confirming that the models behaved as expected. The final step was to repeat this process for all predetermined initial conditions. This final step was designed to confirm data convergence for a range of different inputs.

To pass the criteria for this test, the Newtonian and Lagrangian needed to be within 0.1 degrees from each other at the hip and knee while the multi-body model should have been within 2 degrees of the other two models. This test failed if any of the models encountered singularities that prevented the simulation from running the full amount of time, or if the response of the models did not fall within the specified range.

7.1.2 Documentation

The purpose of documentation testing was to ensure that our work was presented in a manner that the upcoming senior project team can follow. The software that we designed needed to provide documented code, user manuals, and other tools to assist the next team in fully assembling the robotic leg. This test was designed to present our documentation to our sponsors, peers, and the additional members of the robotic locomotion team. Team members should have been able to easily comprehend and quickly locate the information presented in the User Manual and use it to operate the different tools we designed. Based on our documentation, other members should have been able to identify and understand the modeling methods this team used to design the different controller components, such as the plant and the linearized controller.

The only equipment needed to run this test were the User Manual and access to the Google Sheet survey. The User Manual and the Google Sheet survey can both be accessed through the F-11 Robot Leg Motion One Drive account, or by contacting members of the Cal Poly Legged Robot club.

The procedure for this test first included presenting all documentation to the intended audience. The audience read through all the presented information. The target audience was tested with questions that confirmed comprehension on each major tool and model. The Google Sheets survey collected information on whether the documentation is effective.

To pass the criteria of this test, the documentation must have scored an average of 80% or more as judged by the Google Sheet. Sponsor feedback was also necessary to determine quality of the User Manual. The test failed if the target audience deemed the document too brief or too convoluted. We considered any score below 80% to indicate that future students will not be able to reproduce our results.

7.1.3 Optimized Code

The purpose of this test was to verify that our controller code was optimized. This test procedure was designed to verify that the three models, Newtonian, Lagrangian, and Multi-Body, all yield results in an acceptable amount of time. In addition, we tested the run time for the complete Control Loop Containing the Lagrangian. Computer computational speed was crucial to the success of this project. For the leg to function properly, we needed to be able to issue controlled commands in a reasonable amount of time. This test determined if the controller we modeled can run in an acceptable amount of time given the state diagram of the system. The controller had to operate at least 3 times as fast as was required by the state diagram. The state diagram was defined by the research group as 300 milliseconds, therefore the controller code must complete all operations within 100 milliseconds.

The only equipment needed to run this test were a PC and the MATLAB software. This test required the same equipment as the model verification test. Refer to section 7.1.1 to review the necessary equipment. There were no hazards to the individual that needed to be considered, and therefore no personal protective equipment was purchased.

The procedure for this test first included running MATLAB 2020 and preparing to operate both Simulink and Simscape Multibody. Next, we opened the .mlx file code for the control loop. The code for the controller had a built-in timer to determine the amount of time it took to run the code. Next, we input the desired initial torque and position conditions. These initial conditions remained constant for each test to ensure equivalent results. After running the code for both the Newtonian and Lagrangian, we compared time required to run the test. Finally, we started the Multibody simulation and recorded the run time provided by Simscape.

For a program to pass, it had to run under 100 milliseconds as specified in the test description. If the code took too long to run, we would need to optimize the code using some of the methods below. Otherwise, we consulted with Charlie Refvem for additional optimization techniques: minimize lines, minimize loops, and initializing arrays before using them.

7.1.4 Virtual Position Control

The purpose of this test was to virtually test the position trajectory control feature of our controller. The goal was to simulate the movement of the hip of the SDOF leg along an input path. We performed this test using the control loop we created in MATLAB as well as the Lagrangian and MultiBody models. Connecting the desired model to our controller, we employed the feedback linearization technique to control the location of the hip. Data on the angle of the thigh and shank were collected as the model moved along the input path. The passing criteria for the test was to control the hip to within ±1 degree of the desired path. The failure criteria for the test were if the controller encounters a singularity or if response time is greater than 0.1s.

The procedure for this test requires the user to define a path for the hip to follow. For our initial testing, we assigned a straight path that moved the hip upward along the test stand. This path is passed to the controller and the results are simulated. Because we knew the values for the assigned hip path, we compared the data for known locations along the trajectory. Code in the controller produced plots of the error between the desired angle and the actual output angle for visual representation of controller accuracy. This test was initially run with the Lagrangian model five times along the same path to verify that the results were consistent. After the Lagrangian model passed, the MultiBody model was tested five times as well. Our Lagrangian model passed testing and was used in our Control Loop. While our MultiBody model passed with regards to comparing against other models, we had trouble implementing it correctly into our controller. We believe that with some refinement of the model however this is still possible.

7.1.5 Twin Motor Control

The purpose of this test was to employ the controller designed by our team using the fully assembled SDOF leg on its test stand. The goal of the test was to show that the controller could move the hip to a desired height. This test was not completed by Senior Project Team F-11 due to delays in motor shipping. This test required two AK80-6 motors to test the fully assembled SDOF leg on its test stand. Additionally, issues with the test stand constructed by a previous senior project team delayed our timeline for testing. Because the motors were not received and the test stand was not functional, we were unable to perform this test. However, the procedure is valid and can be used in the future once both motors are in house and the test stand is assembled to our Sponsors satisfaction.

7.2 Results

The following section includes the results of the tests conducted above. The results demonstrate that we have met the specifications set for the project and verified our design. Results include pictures, collected data, uncertainty analysis, etc..

7.2.1 Model Verification

Through the process of completing our model verification testing we recorded raw angular position data for the Newtonian model, the Lagrangian model our team created, the Lagrangian model created by the Cal Poly Legged Robotics Club, and the MultiBody model. This data was collected under 5 different system conditions. The raw data for this test can be found in Attachment M. Plots of the data versus time can be seen in Figures 7.1-7.10.

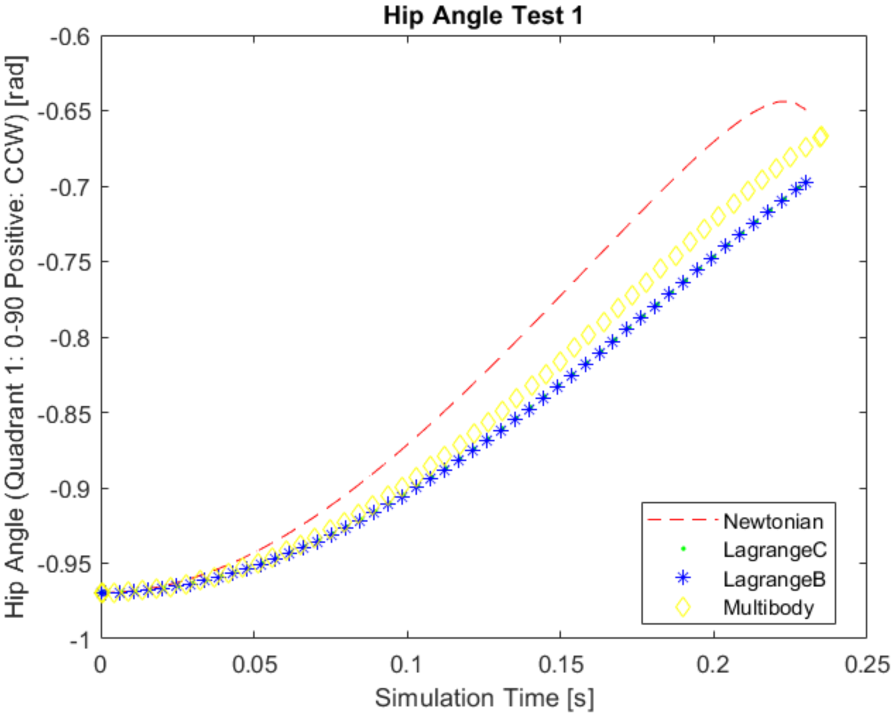


Figure 7.1 Hip angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, and 0 torque at both the knee and hip.

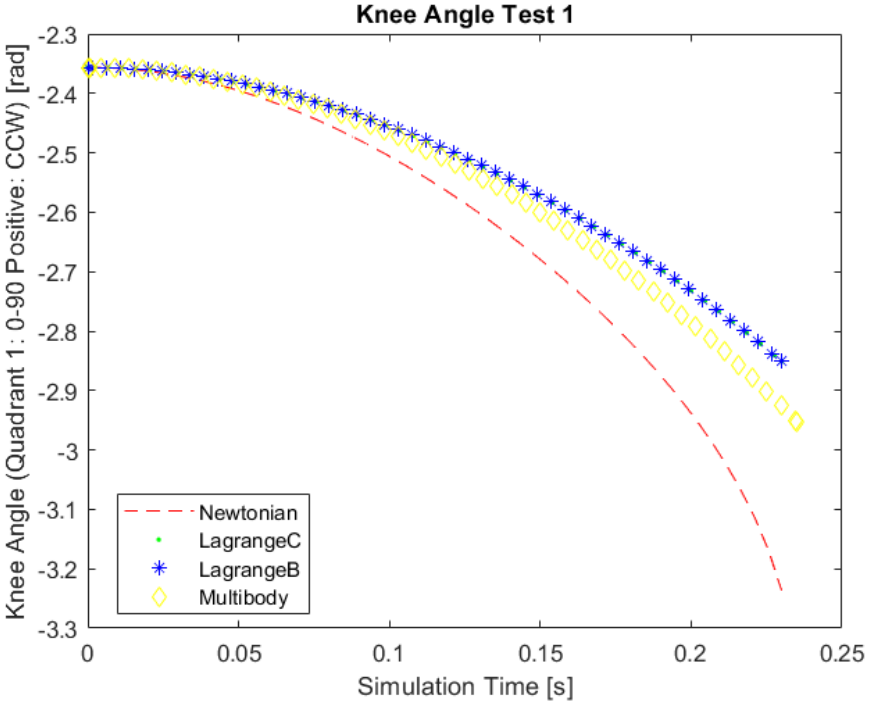


Figure 7.2 Knee angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, and 0 torque at both the knee and hip.

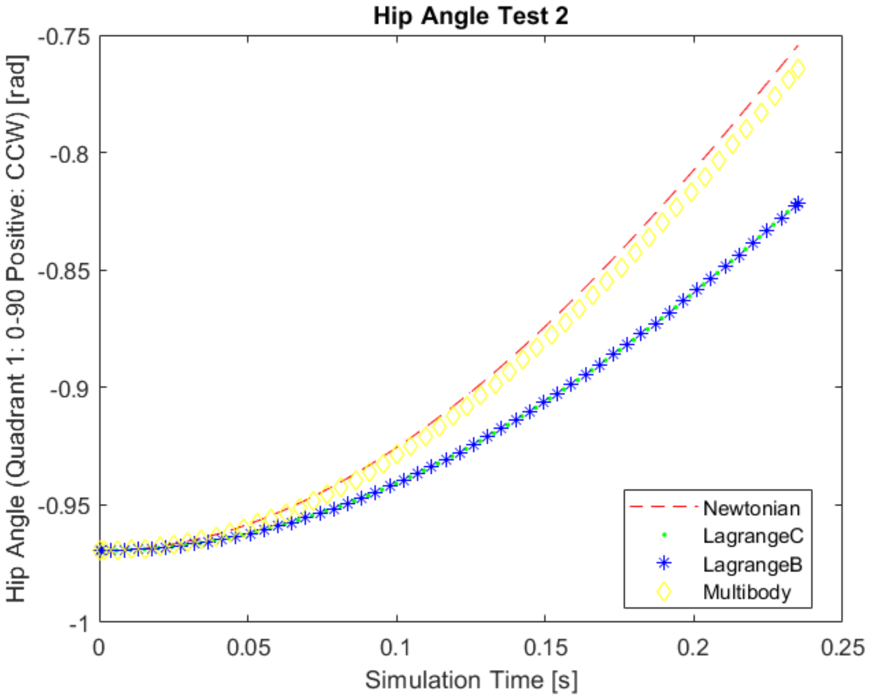


Figure 7.3 Hip angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 20 torque at the hip, and 0 torque at the knee.

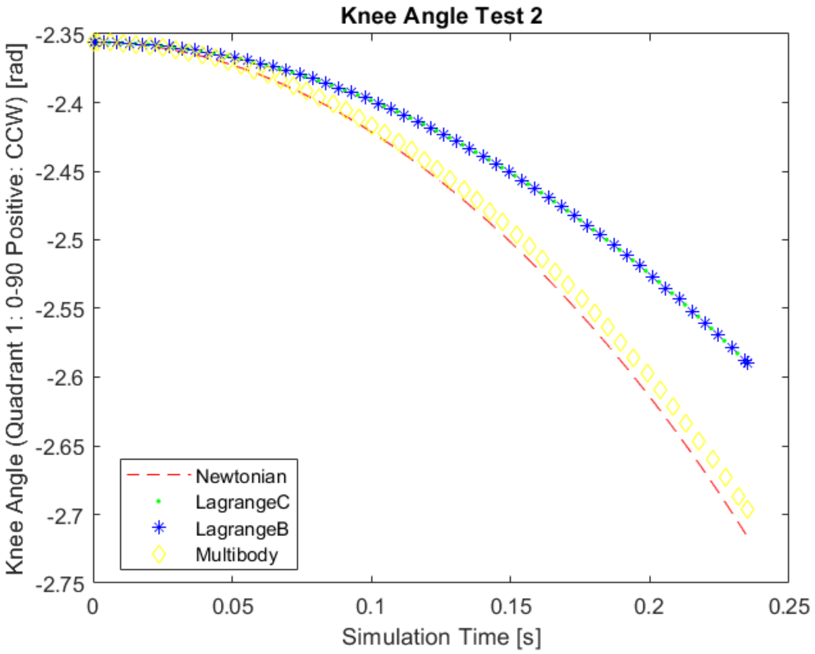


Figure 7.4 Knee angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 20 torque at the hip, and 0 torque at the knee.

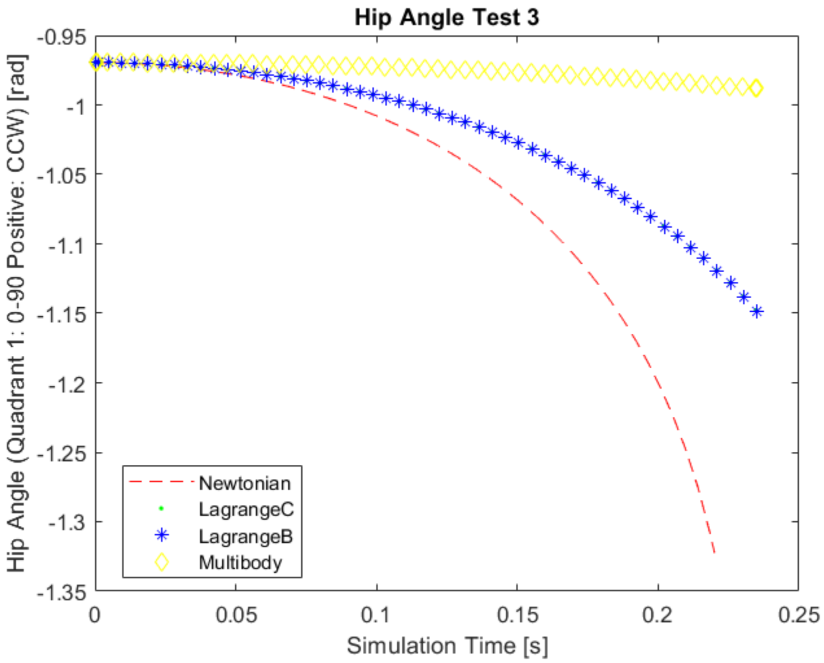


Figure 7.5 Hip angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 45 torque at the hip, and 0 torque at the knee.

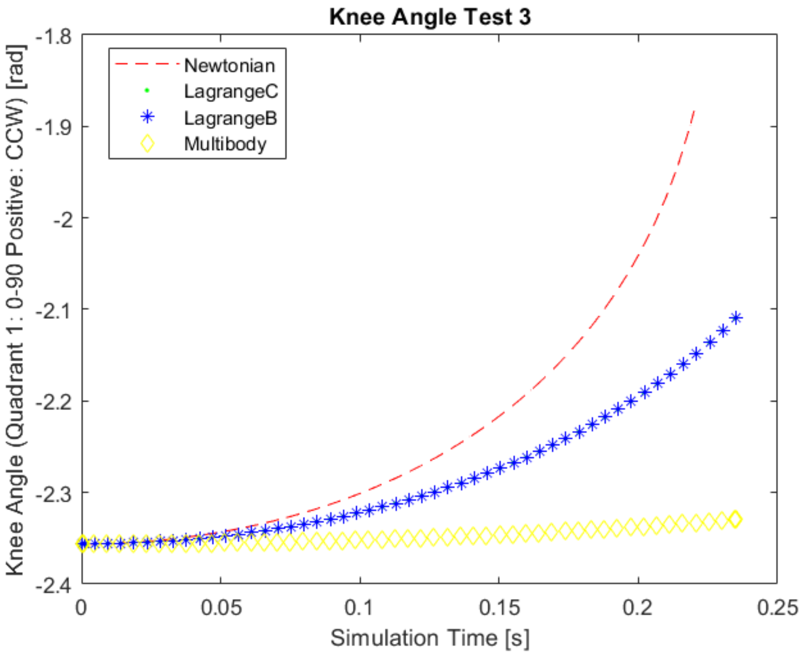


Figure 7.6 Knee angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 45 torque at the hip, and 0 torque at the knee.

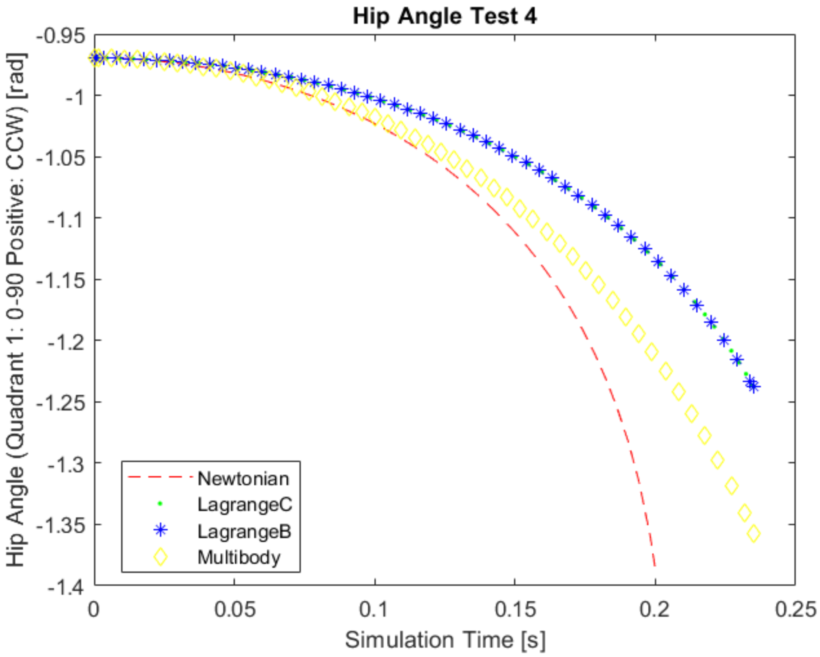


Figure 7.7 Hip angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 20 torque at the hip, and -20 torque at the knee.

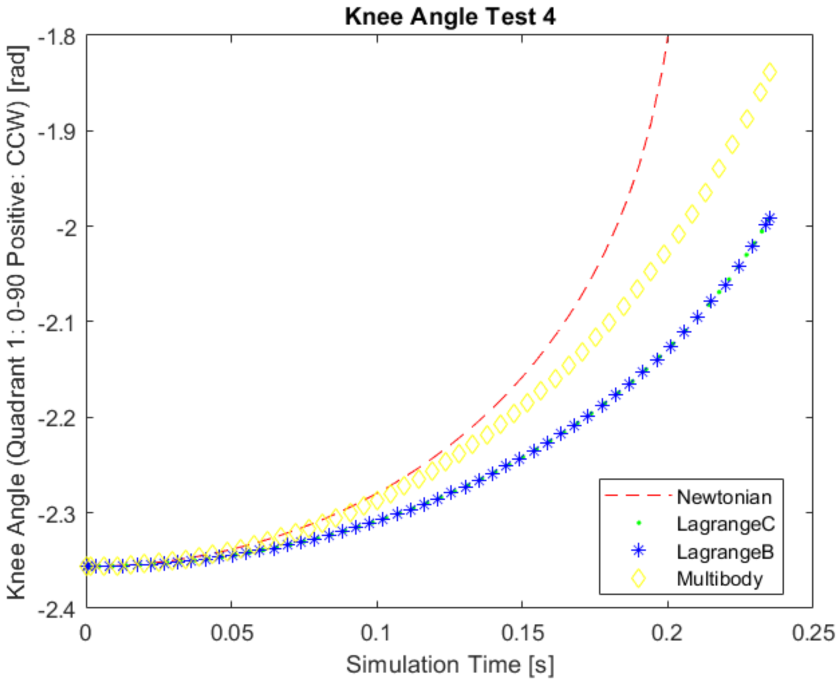


Figure 7.8 Knee angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -55.55 degree hip angle, -135 degree shank angle, 20 torque at the hip, and -20 torque at the knee.

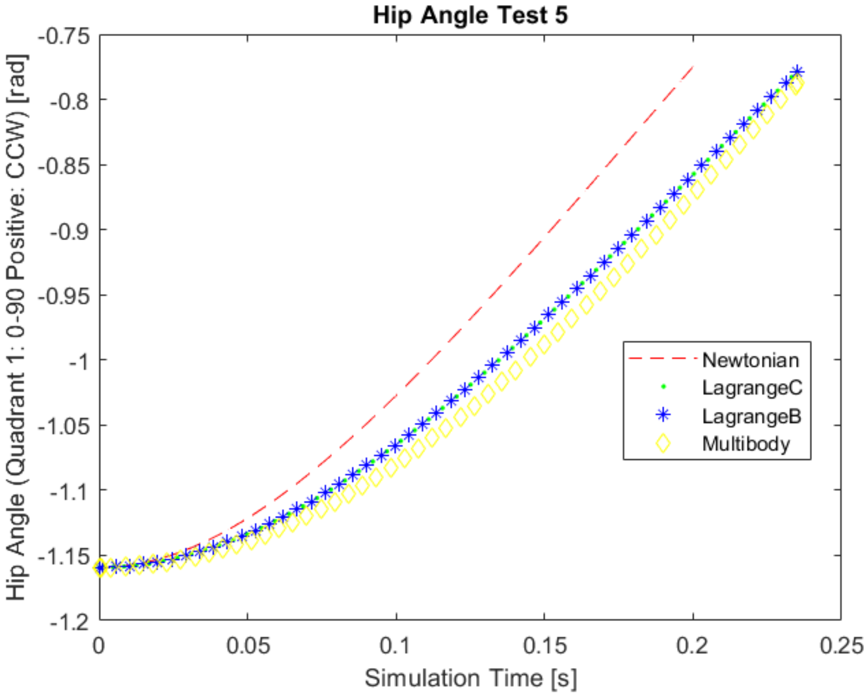


Figure 7.9 Hip angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -66.42 degree hip angle, -120 degree shank angle, and 0 torque at both the knee and hip.

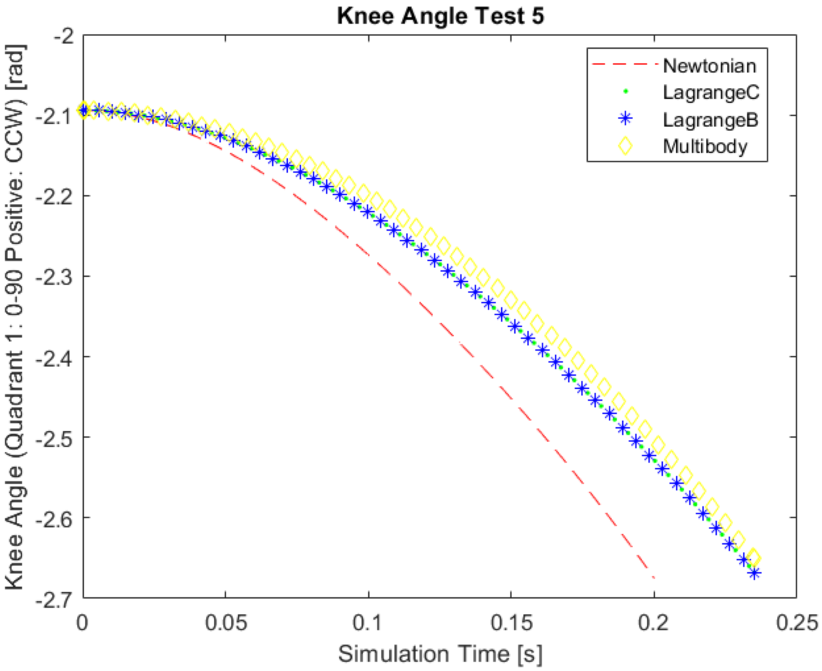
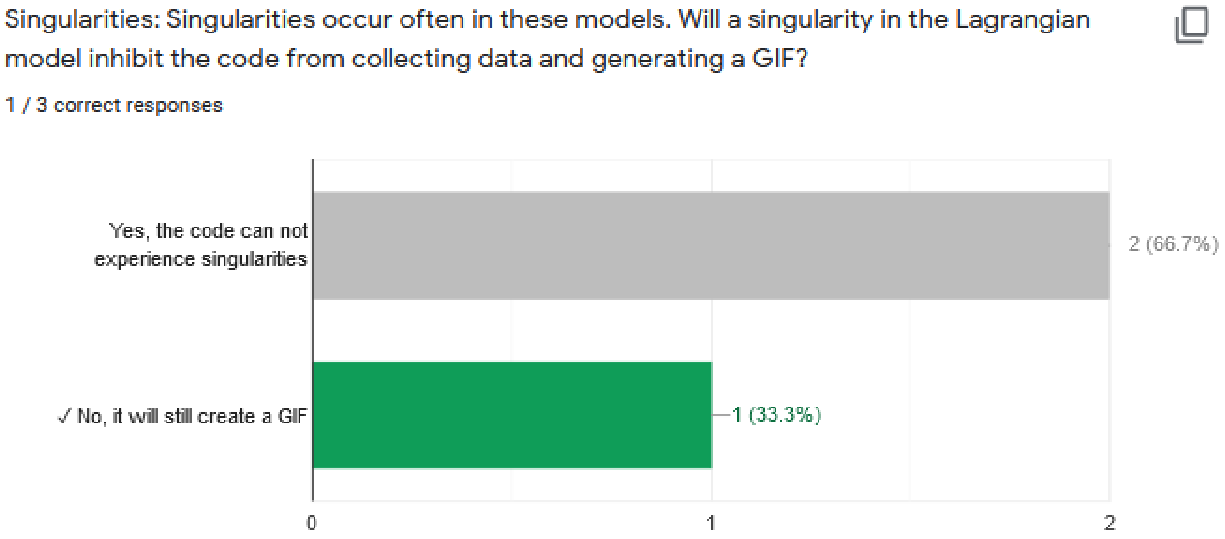


Figure 7.10 Knee angle of the Newtonian, LagrangianC (model completed by Team F11), LagrangeB (model completed by Cal Poly Legged Robotics), and Multibody model as a function of time. The initial conditions of this simulation: -66.42 degree hip angle, -120 degree shank angle, and 0 torque at both the knee and hip.

Note that the Newtonian model diverged in each test. This result displayed itself in the animation of the leg in freefall, in which the leg accelerated downward roughly with roughly double the magnitude of gravity. This error is still under examination by our team and has not been rectified. However, the trajectory of the leg is accurate relative to the other models, as shown in the Figures above.

7.2.2 Documentation

To test our documentation, we created a survey that required our fellow students to study portions of the controller and then demonstrate comprehension of its content. We deemed that an 80% on the survey demonstrated successful and complete documentation. After polling our classmates, we received strong results which indicated that our documentation was sufficient for users to operate and modify the code. Though we only received 3 submissions and each scored above an 80%, the responses still drew attention to some details of our project documentation. The incorrectly answered questions are shown below in Figures 7.11 – 7.13 and the actions taken to increase user understanding of the material is addressed. The correct answer is shown in green for each question.



Singularities: Singularities occur often in these models. Will a singularity in the Lagrangian model inhibit the code from collecting data and generating a GIF?

1 / 3 correct responses

Yes, the code can not experience singularities
2 (66.7%)
✓ No, it will still create a GIF
1 (33.3%)
0
1
2

Figure 7.11 – Question: Will a singularity in the Lagrangian model inhibit the code form collecting data and generating a GIF?

This question addressed the issue of whether the MATLAB script would generate a GIF regardless of the presence of a singularity. As noted above, the code will not generate a GIF even if the system encounters a singularity. This is because the singularity simply breaks the Simulinik solver, the MATLAB code will not continue past the simulation. However, this question is a bit confusing because the GIF-generating section of code can still be run after the singularity occurs, it just must be done manually. We included documentation specifying this feature of the code as it is often useful to see when and how the singularity appears in the system by animating the results in a GIF.

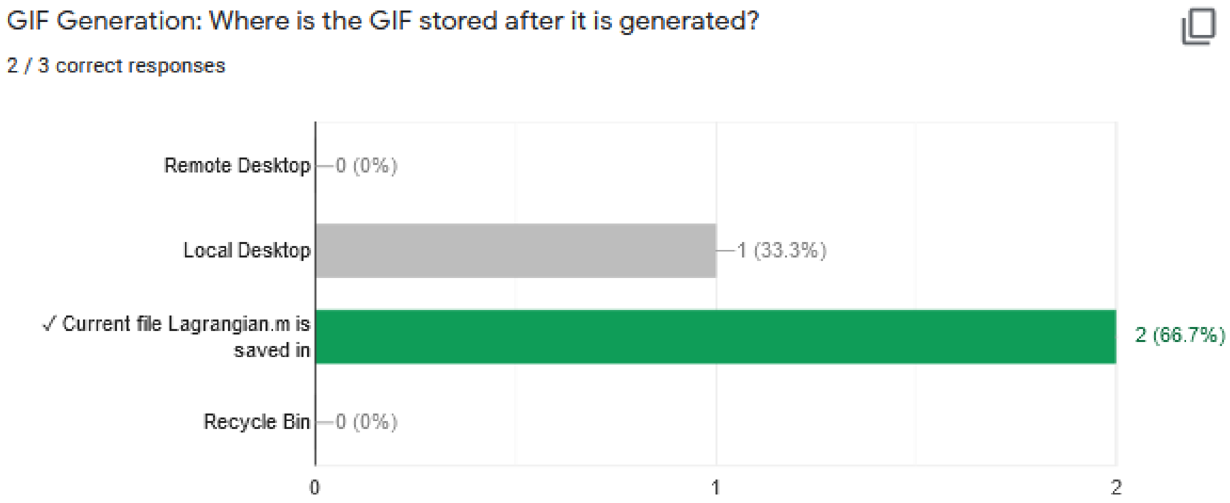


Figure 7.12 – Question: Where is the GIF stored after it is generated?

This question addressed the issue of how to access the GIF that the Lagrangian code generates after it is run. As noted above, the GIF is stored in the same file as the file "Lagrangian.m" is currently stored in. This is a default MATLAB function but can be frustrating if the user does not know where to look. To fix this documentation issue, we included documentation in the introduction to compliment the documentation that was located near the GIF-generating code.

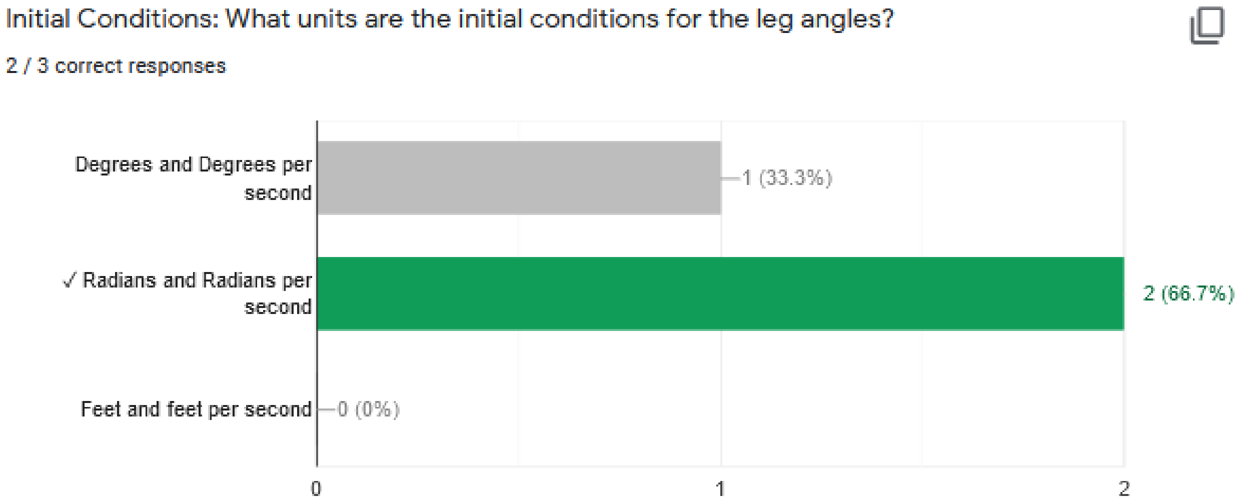


Figure 7.13 – Question: What units are the initial conditions for the leg angles?

This question addressed the units used in our code. The response to this question was a bit confusing, since the units for the initial conditions are listed next to the code defining them. Regardless, to clarify the units used in the system, we created a list of units in the introduction to avoid further confusion. Confusing degrees and radian in MATLAB can be a critical error that carries through an entire project, so we want to encourage proper unit documentation and usage throughout our program.

Using the results from our survey, we were able to identify some major and minor documentation issues throughout our code. This test allowed us to sample our documentation before handing off the final package to our sponsors. We learned to be more careful with units and to ensure that documentation of important values and features is noted more than once in the program.

7.2.3 Optimized Code

During our code optimization testing we recorded multiple full simulation run times for each program tested. These times were divided by the number of iterations giving us the time for one control cycle or one simulation cycle to occur. This exposes how quickly our programs can receive information and give an output. Below in Tables 7.1-7.4 the run time data is displayed.

Table 7.1 Run Time Data Multibody Model

Multibody	
Full Sim Run Time	One Iteration
(sec)	(ms)
19.036305	339.9340179
4.280495	76.43741071
1.946986	34.76760714
4.430197	79.11066071
Average:	132.5624241

Table 7.2 Run Time Data Newtonian Model

Newtonian	
Full Sim Run Time	One Iteration
(sec)	(ms)
1.349837	24.99698148
0.452596	8.381407407
1.234216	22.85585185
0.663042	12.27855556
Average:	17.12819907

Table 7.3 Run Time Data Lagrangian Model

Lagrangian	
Full Sim Run Time	One Iteration
(sec)	(ms)
1.208687	0.021583696
0.312674	0.005583464
0.339507	0.006062625
0.295182	0.005271107
Average:	0.009625223

Table 7.4 Run Time Data Control Loop with Lagrangian

Control Loop with Lagrangian	
Full Sim Run Time	One Iteration
(sec)	(ms)
26.085546	7.692582129
1.009561	0.297717782
1.677924	1.676247752
0.862658	0.861796204
Average:	2.632085967

After recording and analyzing the data, all programs except the Multibody model pass the test criteria of running faster than 100 [ms]. The Multibody may be the one program that cannot be modified to decrease run time. Almost all of the code comes prepackaged by Simscape. The overall program is then pieced together by connecting the different building blocks provided by Simscape. One possibility for decreasing the run time is to have access to a better computer. Despite the Multibody model failing the test, the project is almost unaffected. This is because the primary run time of concern is the controller's run time. The combined run time average of the controller and the Lagrangian model was 2.63 [ms] which is far below the 100 [ms] requirement. Since this time accounts for the Lagrangian run time as well it is clear that the controller runs even faster than the recorded 2.63 [ms]. This verifies that our code optimization is sufficient for proceeding with the project.

7.2.4 Virtual Position Control

The virtual position control test verified the performance of the controller by comparing the output values to the input values. The output values represent the current position of the motor as determined by our mathematical model. The input values represent a path input from the user for the leg to follow. As discussed previously, the hip is constrained to vertical motion and the foot is fixed horizontally. Each of the testing iterations produced 4 plots. A plot is shown for the input and output at both the hip and the knee as well as the error at both the hip and the knee. Table 7.5 displays the Figures that were associated with each test for easy navigation of the results. All of the raw data collected during testing can be found in Attachment N.

Table 7.5 – Testing Specifications

Test #	Hip Input Type	Figures	Initial conditions			
			θ_1 [deg]	θ_2 [deg]	$\dot{\theta_1}$ [deg/s]	$\dot{\theta_2}$ [deg/s]
1	Sinusoidal Hip Motion	7.14 – 7.17	-30	-135	0	0
2	Constant Hip Height	7.18 – 7.21	-30	-135	0	0
3	Ramp Hip Input	7.22 – 7.25	-30	-135	0	0
4	Sinusoidal Hip Motion	7.26 – 7.29	-60	-125	0	0
5	Constant Hip Height	7.30 – 7.33	-60	-125	-5	6.6076

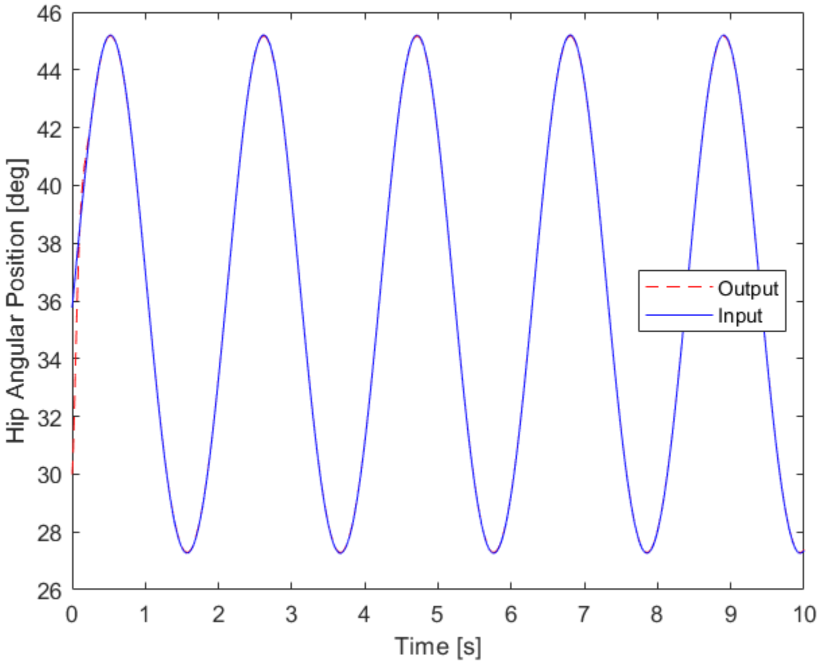


Figure 7.14 – Graph of the input and output hip angular position vs. time for Test 1. This test employed Sinusoidal Hip Motion with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the hip to the desired angular position in less than one second.

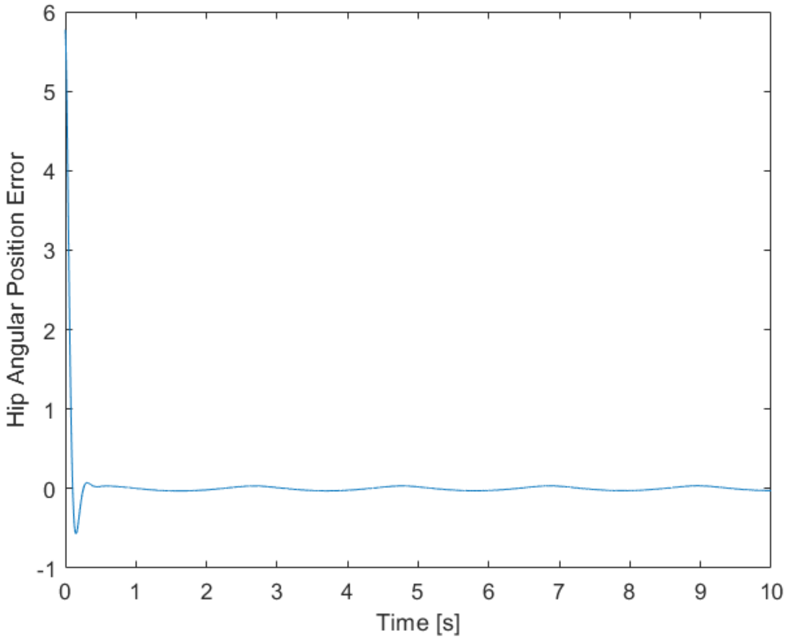


Figure 7.15 – Graph of the error signal for the hip angular position for Test 1. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired hip angle.

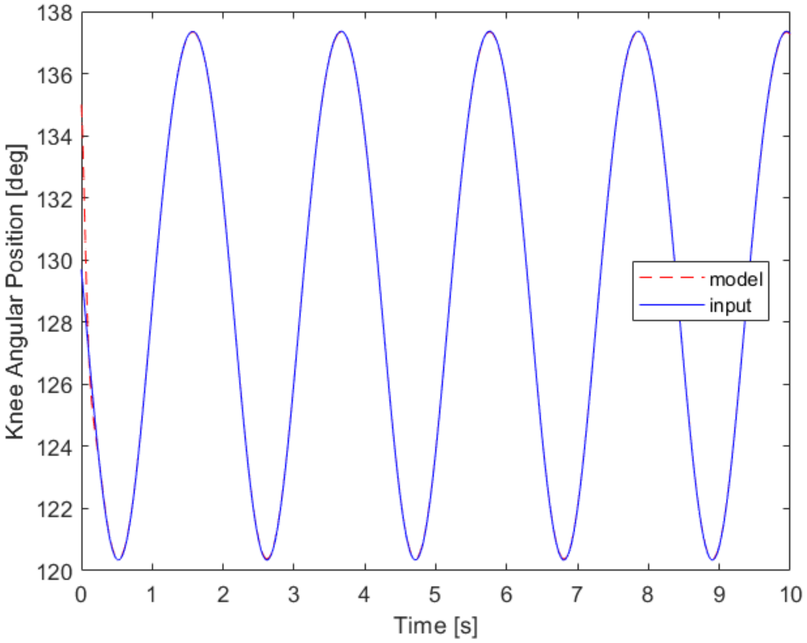


Figure 7.16 – Graph of the input and output knee angular position vs. time for Test 1. This test employed Sinusoidal Hip Motion with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the knee to the desired angular position in less than one second.

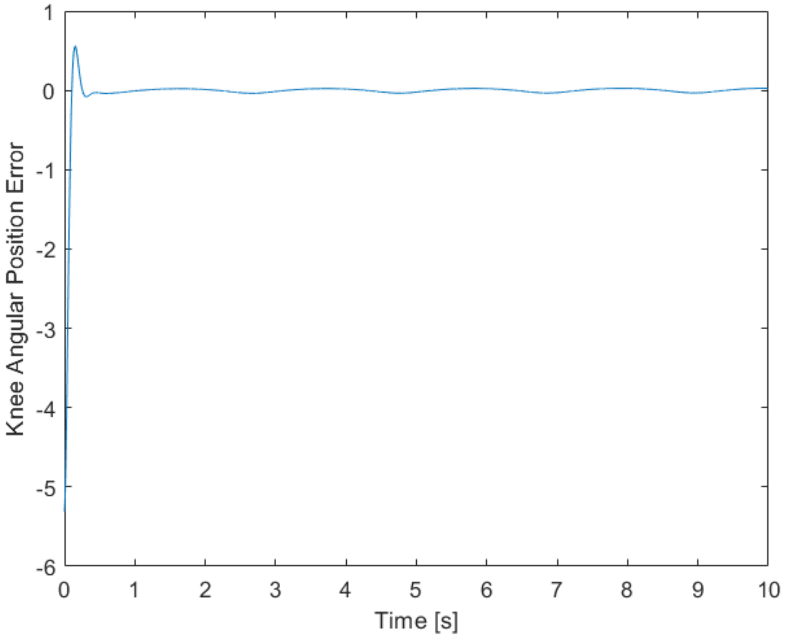


Figure 7.17 - Graph of the error signal for the knee angular position for Test 1. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired knee angle.

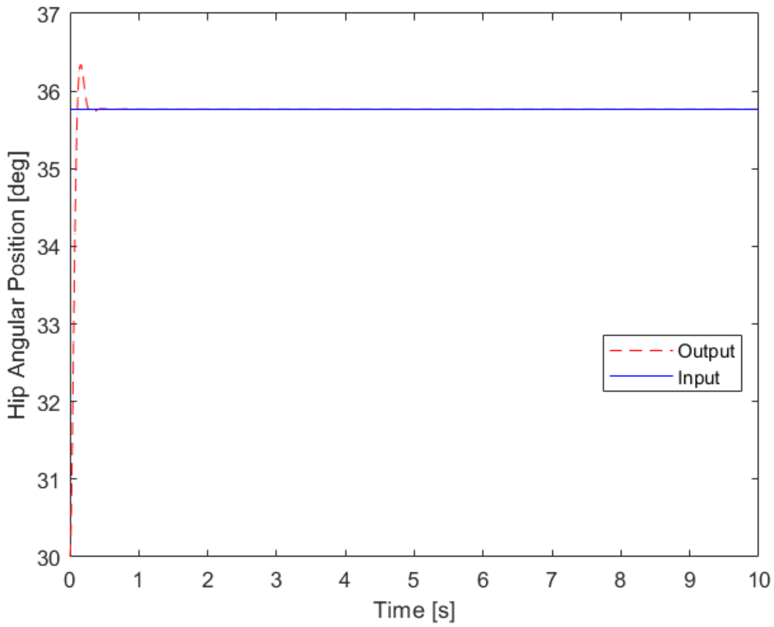


Figure 7.18 – Graph of the input and output hip angular position vs. time for Test 2. This test employed Constant Hip Height with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the hip to the desired angular position in less than one second.

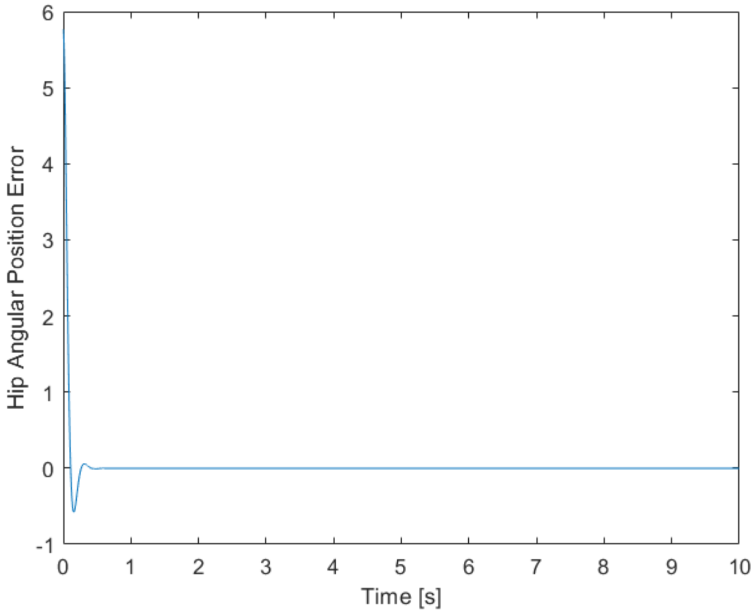


Figure 7.19 - Graph of the error signal for the hip angular position for Test 2. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired hip angle.

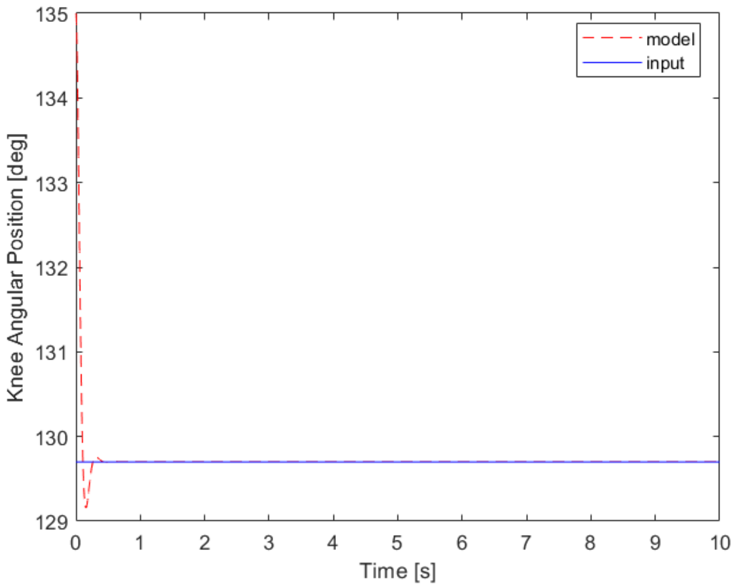


Figure 7.20 – Graph of the input and output knee angular position vs. time for Test 2. This test employed Constant Hip Height with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the knee to the desired angular position in less than one second.

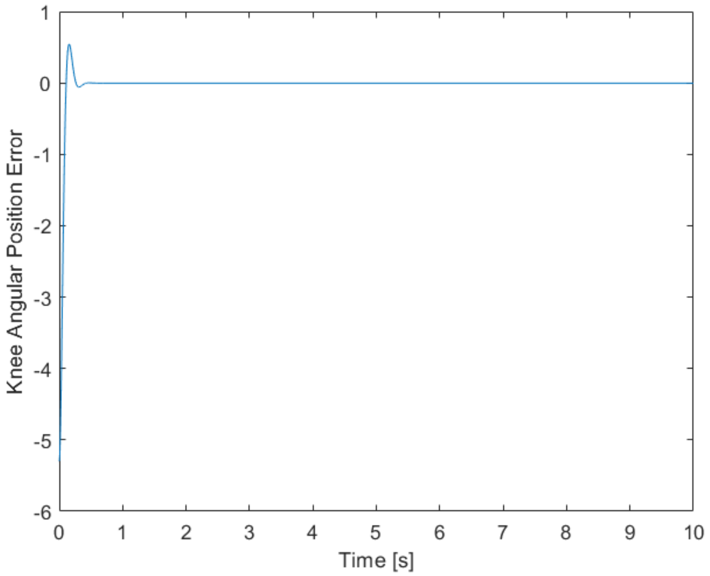


Figure 7.21 – Graph of the error signal for the knee angular position for Test 2. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired knee angle.

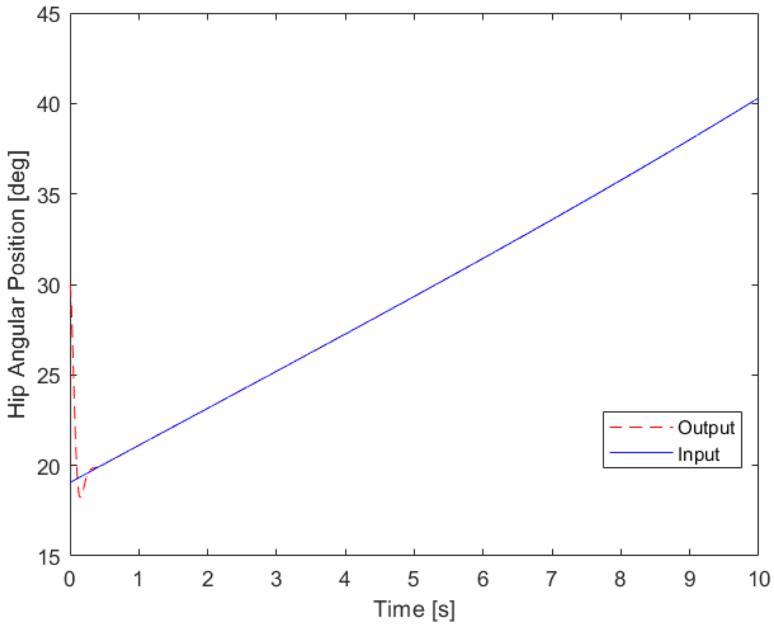


Figure 7.22 – Graph of the input and output hip angular position vs. time for Test 3. This test employed Ramp Hip Input with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the hip to the desired angular position in less than one second.

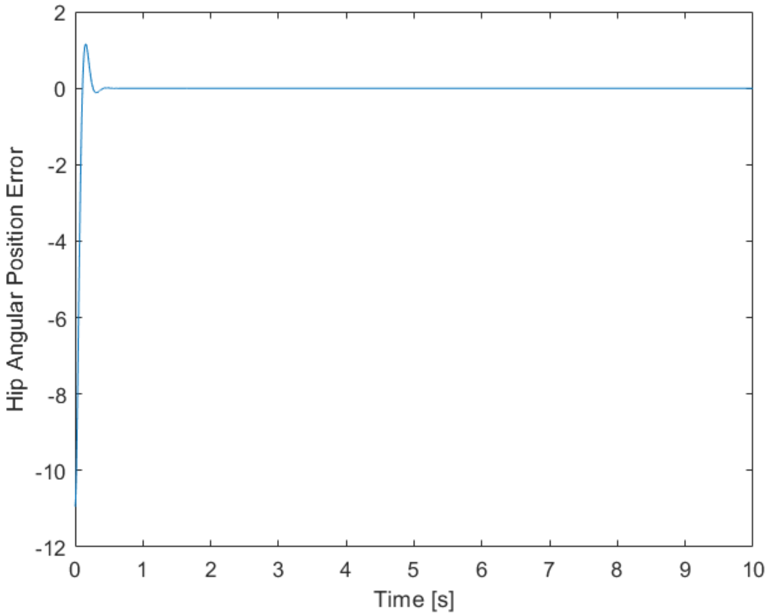


Figure 7.23 – Graph of the error signal for the hip angular position for Test 3. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired hip angle.

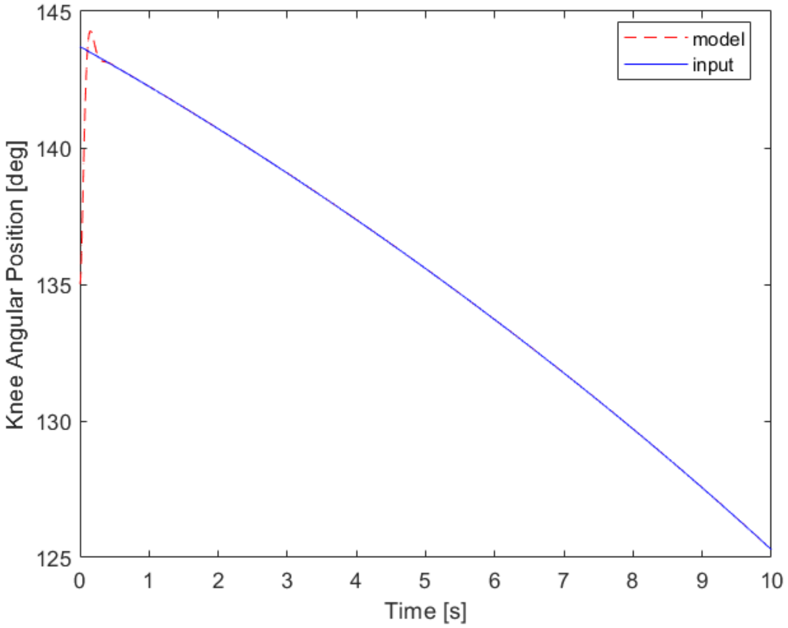


Figure 7.24 – Graph of the input and output knee angular position vs. time for Test 3. This test employed Ramp Hip Input with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the knee to the desired angular position in less than one second.

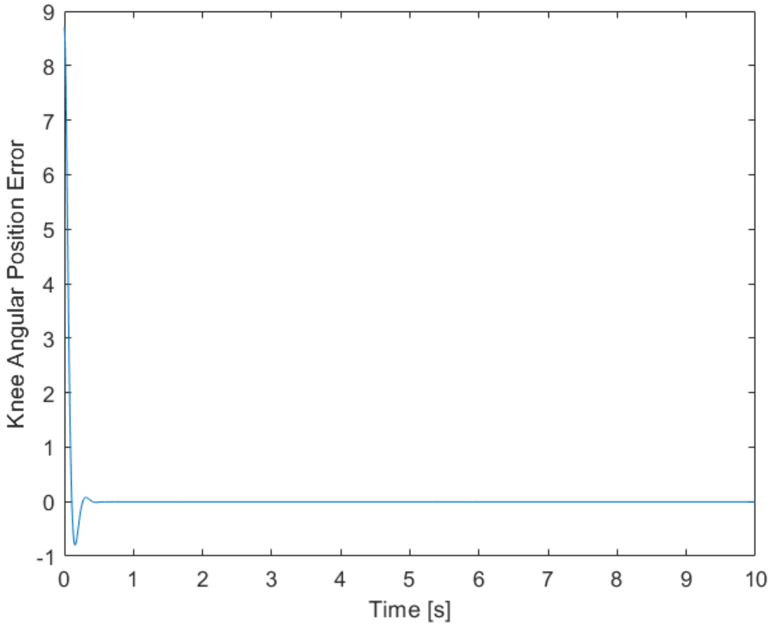


Figure 7.25 – Graph of the error signal for the knee angular position for Test 3. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired knee angle.

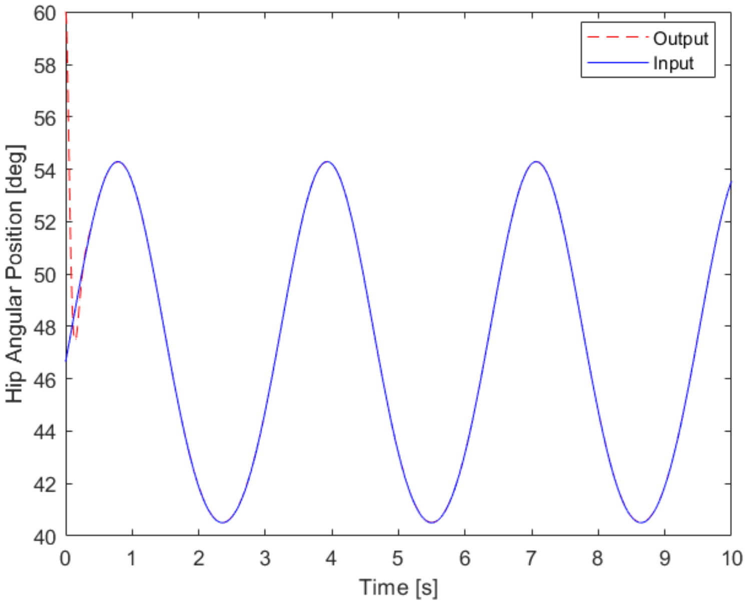


Figure 7.26 – Graph of the input and output hip angular position vs. time for Test 4. This test employed Sinusoidal Hip Motion with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the hip to the desired angular position in less than one second.

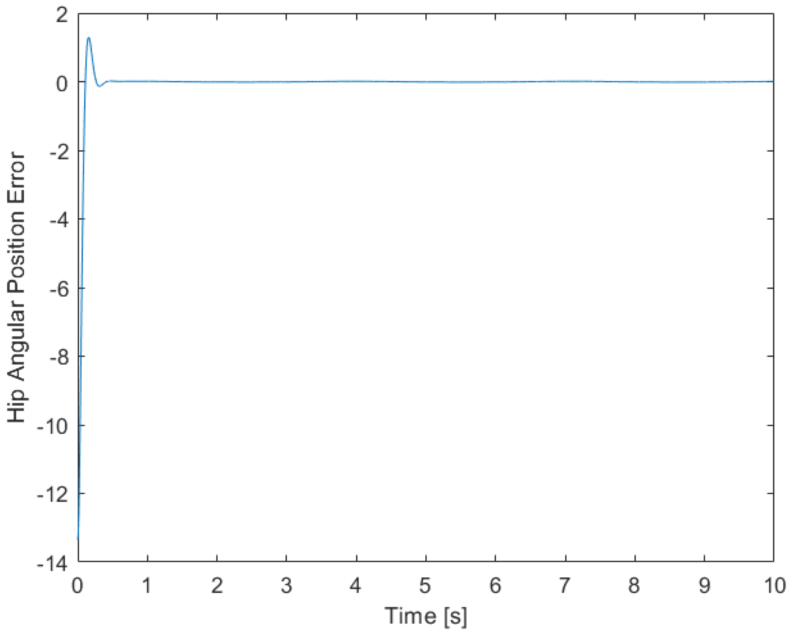


Figure 7.27 – Graph of the error signal for the hip angular position for Test 4. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired hip angle.

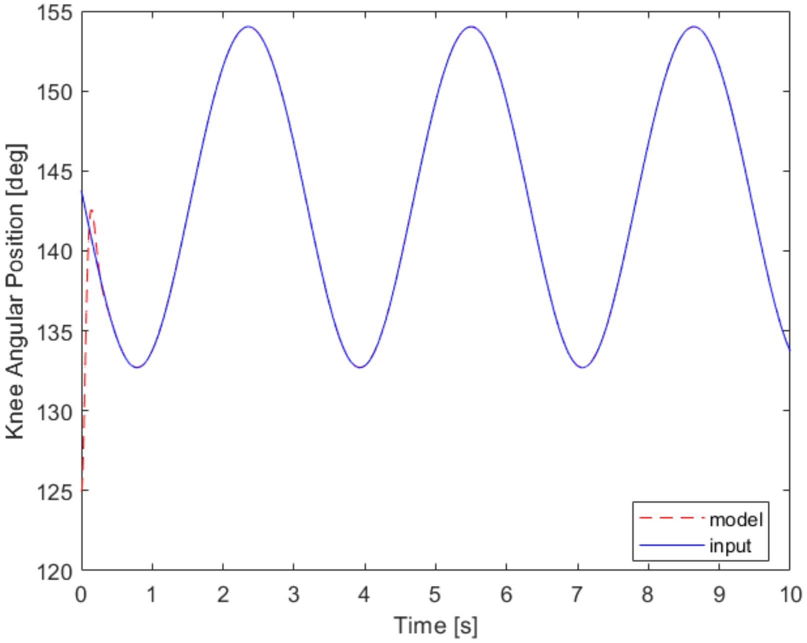


Figure 7.28 – Graph of the input and output knee angular position vs. time for Test 4. This test employed Sinusoidal Hip Motion with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the knee to the desired angular position in less than one second.

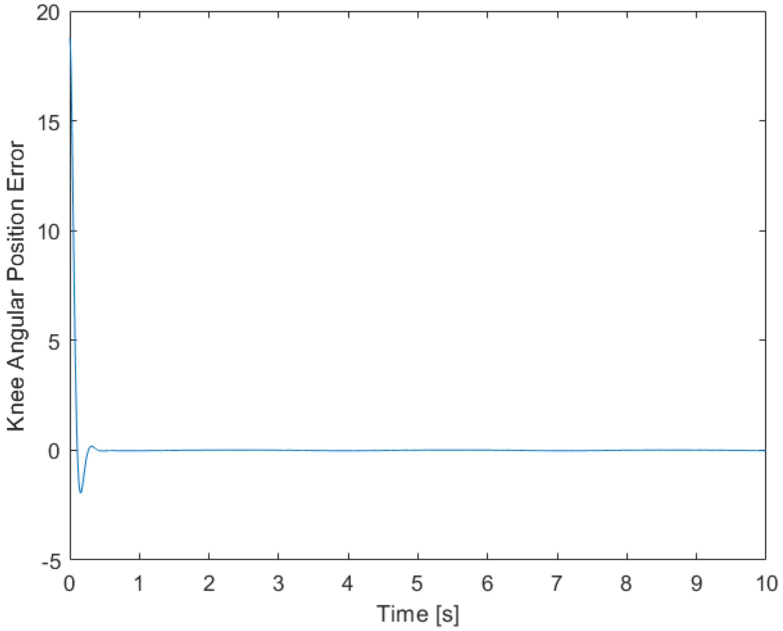


Figure 7.29 - Graph of the error signal for the knee angular position for Test 4. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired knee angle.

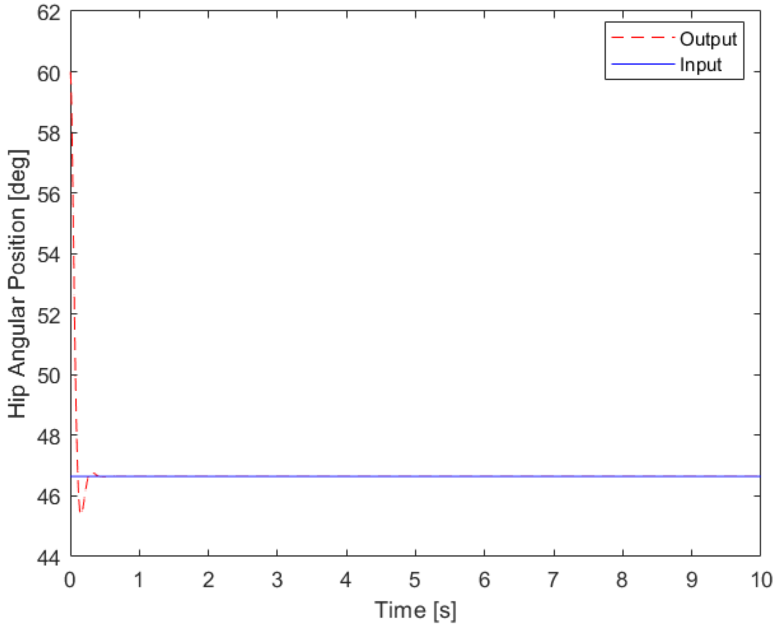


Figure 7.30 – Graph of the input and output hip angular position vs. time for Test 5. This test employed Constant Hip Height with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the hip to the desired angular position in less than one second.

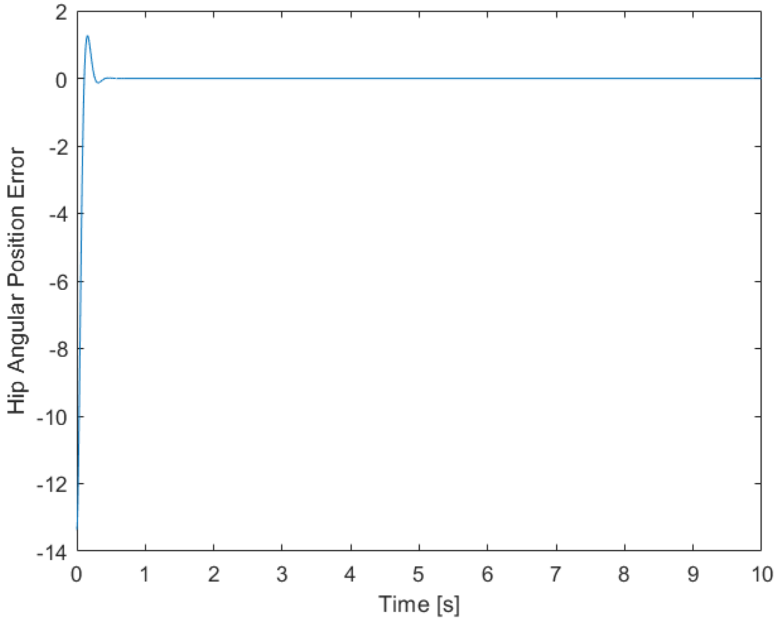


Figure 7.31 – Graph of the error signal for the hip angular position for Test 5. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired hip angle.

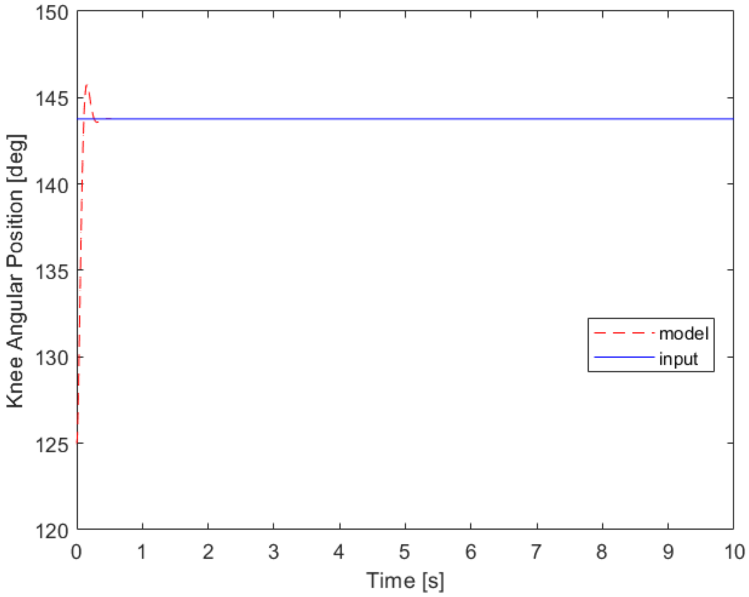


Figure 7.32 – Graph of the input and output knee angular position vs. time for Test 5. This test employed Constant Hip Height with initial conditions listed above in Table 7.6. Note that the output is initially separate from the input position, but that it rapidly converges to the input path. This indicates our controller brought the knee to the desired angular position in less than one second.

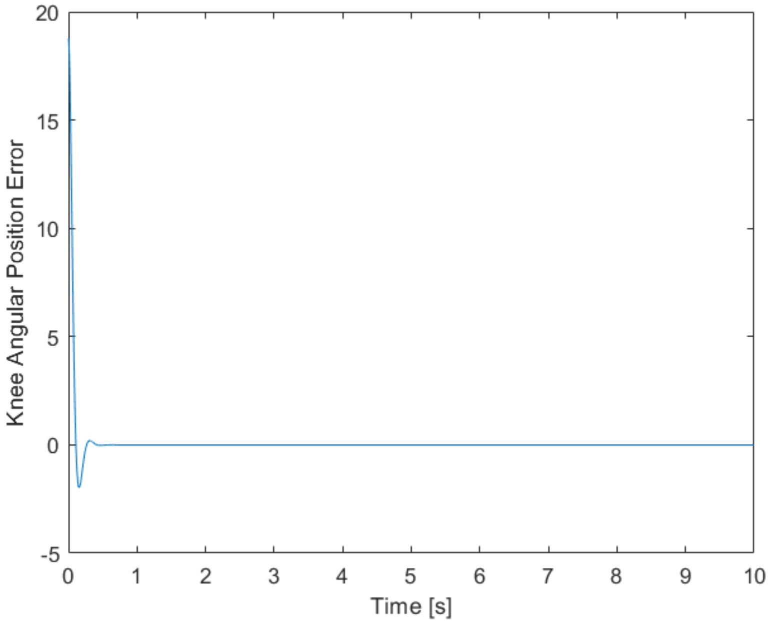


Figure 7.33 - Graph of the error signal for the knee angular position for Test 5. The positional error is in degrees and is oscillating around zero. This demonstrated that this test passed with the desired accuracy of 1 degree from the desired knee angle.

After testing the controller with various paths and initial angles, we were determined that our controller functions exactly as desired. It can control the leg precisely, within one degree of the desired position. This is very promising for teams moving forward and implementing this controller on the physical leg.

7.3 Discussion

This section will discuss concluding thoughts from the verification testing process. If any tests were listed in the DVPR and not discussed in this report, the reasoning is justified in this section. Major challenges and lessons for future group testing is also addressed.

7.3.1 Missing Tests

The only test we did not complete that is listed on our DVPR is the jumping leg test. We ambitiously planned to fully design the controller by the end of winter quarter and begin testing the physical leg in spring quarter. However, due to complications with the current leg design and test stand design, we were unable to successfully assemble a functioning test stand. Additionally, our controller was not functioning properly until the middle of Spring Quarter due to technical issues with the feedback linearization and mathematical models. Lastly, we were unable to secure the second motor required for functional physical testing due to shipping issues. All other specifications and tests were met as indicated in the sections above.

7.3.2 Challenges and Lessons

The first challenge we encountered was with mathematical modeling. Our models provided different issues along the way. The Newtonian encountered issues with the resultant accelerations during animations, the Lagrangian encountered many singularities, and the Multibody provided barriers of knowledge as none of the members had used the program before. These are just a few examples of issues we encountered with these models and we learned that modeling a simple system is rarely straightforward. The model development takes a fair amount of time and effort to complete well and can always benefit from an extra set of eyes looking over it.

The physical component handoff between senior project teams was another challenge we faced. The components were not well designed, and the functions of components was not obvious. This resulted in wasted time assembling the physical test rig and eventually resulted in damaged components. We learned that handing off physical components is incredibly difficult unless explicit and thorough documentation is included. This is part of the reason our sponsors stressed the documentation of this project. Through the process we also learned that coordinating with the sponsors is the best way to maintain quality documentation because they are the reference for students moving forward.

The last issue we encountered in building this controller was in connecting the various components and models into the controller. Because we each wrote different sections of the controller, we used different variables and nomenclature which made connecting different components difficult. Additionally, we encountered issues aligning the coordinates used in our mathematical models to the coordinate system in MATLAB. We learned that the alignment of the coordinates in MATLAB is not well documented and must be considered when importing work. Lastly, we encountered minute errors frequently throughout the process of designing our controller. Often, the errors were simply misplaced negative signs or misnamed variables. We learned to take our time when inputting the equations to ensure that we avoided as many errors as possible.

8.0 Project Management – Complete Design Process

In this section we will review and assess the design process that our team underwent through the duration of this project. We will touch on the successes and learning experience we plan to carry with us through our professional careers.

8.1 Simscape Model

The Simscape model is used to control the animation created using Multibody. This simulation is more accurate than the mathematical models because it includes distributed mass and inertia of the system while the mathematical model considers only point masses. Using this model, we will tune our controller so that it is able to control the motion of the leg accurately and precisely. This model will validate our design if we are unable to secure the funds to construct a physical prototype.

8.1.1 Simscape Model Process and Successes

It was essential that we developed models that accurately replicated real world conditions for our leg. More developed means we had more options and tools for comparison. For much of the project it was our intention to use the Simscape model in our final controller. In the end this was not implemented. This is mostly due to difficulties we had in learning how to implement the Simscape model.

It was beneficial for our whole team to learn more about Simscape modeling and Simulink at large. Simscape also helped gives a more intuitive sense of how the hardware and controller would eventually interact. However, like most of the modeling that we did for this project we ran into bugs and errors that took excessive time to iron out. The biggest lesson that we took away was the value of using tools that you are already familiar with to complete a project. Having to learn new things and implement at the same time can be an unnecessarily lengthy process.

8.2 Finalize Mathematical Model

We are required to provide forward and backward kinematics of the leg along with a mathematical model. To complete this task, we will need to finalize our hand calcs with the revisions and updates provided by our sponsors. The Lagrangian and Kinetic models will be verified using intuition and can be compared against each other to determine general accuracy of our system model. If the animate leg follows an intuitive path given the values we apply, we can determine that our model is accurate for our system. These animations will then be compared to the Simscape animation that is generated with more accurate geometries.

8.2.1 Mathematical Model Process and Successes

The mathematical modeling was a process that we began from the inception of this project. It was necessary step for every aspect of our controller. We began with Newtonian modeling and eventually added Lagrangian to have more models that we can test against. We tested both models against one another as well as against other models that were provided by our sponsors and other members of the research team. While we were able to get both models working, we opted to use the Lagrangian. This decision came down to response speed of the model and how closely it aligned with the other provided models.

Working with these two methods we were able to recognize the value of having two models to compare. While we ran into troubles with both models and resolving issues took longer than expected, insights were gained from those difficulties that helped to solve errors in other models. We also developed an appreciation for the simplicity and effectiveness of the Lagrangian method. In future projects of this sort, we will likely use it again. The main lesson to learn from our Mathematical Modeling was the importance of double-checking hand calculations with a peer. There were many instances where an ongoing problem was quickly solved by getting another pair of eyes on it.

8.3 Testing the Controller

Once we establish our control method and develop a controller, we will use the previous senior project's CAD model in a Simscape to verify the functionality of our controller. During this phase we will create ways to test our program to ensure it works perfectly. After successfully creating an effective control program that is fully implemented in Simscape and verified through testing, we will ensure that we have sufficient documentation for future students to take over the project. Following finalizing our documentation we will attempt to reach any stretch goals we set for ourselves. The final piece of our project process will be to complete the Final Design Review (FDR). After the FDR we will hand off our completed work and our senior project will be complete. See Table 5.1 and the GANNT chart in Attachment M for details on this timeline.

8.3.1 Controller Process and Successes

We were provided reference controllers to test our own model against and compare results. We were able to successfully match our sponsors level of control with a high degree of precision. Our attention then turned to updating our documenting to make the code as easy to understand as possible for future users. In the same way we were able to test and compare our mathematical model's different approaches, comparing controllers against each other helped find ways to improve both controller models: ours and our sponsors. The finalization of our control loop came later in the process than our team was expecting. This did not leave much time for hitting our stretch goals or controller refinement, but we are satisfied with the performance of our controller on the simulated model.

Our team developed insights into using feedback linearization methods to linearize our plants and learned new ways to effectively test and run a control method. We also gained some rudimentary knowledge of CAN communication between Simulink and motor hardware. We feel confident that the next group will be able to pick up where we left off and smoothly transition to a physical model. The biggest lesson that our team will take away from working with this controller is the value in incremental changes. We tried a few times to dive deep into the controller without fully mapping out and plan and this led to very slow initial progress. Starting with a road map and a bare model was key for our success in developing a functioning controller. In the future we will remember this and work on only solving the biggest issues first before trying to make everything fit together.

9.0 Conclusion and Recommendations

This Final Design Review document covers the current progress that Team F-11 has made on the design of a controller for a SDOF leg. It covers the state of our design currently and details the methods we used to model our system, verify our design, and test our controller. To date, we have completed model verification, documentation, and animation of SDOF leg controller. We ask that our sponsors review the material presented in this document to give approval on both the current design and the forward direction of the project. We coordinated with our sponsors throughout the design of the controller and completed sufficient steps to ensure a smooth handoff to our sponsors and future students. For access to any code, notes, or work please feel free to reach out to our project sponsors, Professor Xing and Charlie Refvem.

In our testing of the controller, we encountered many issues and will address them briefly here. The first major issue we encountered was singularities. There is no way to avoid singularities in this solution. Because of the nature of the leg, there are configurations that are physically impossible for the leg to reach and therefore impossible for the solver to determine. To reduce time required in testing, we recommend changing the setting in MATLAB Simulink to fixed step size. This will inhibit MATLAB from reducing step size when trying to solve the system and allow the solver to converge on the error more rapidly.

The second major issue we encountered was determining initial conditions that represent reasonable leg positions. This is due to the way that MATLAB defines the orientation of angles. To resolve this, we recommend simply fiddling with the angles until the leg is in the desired position. This is can often be fixed by simply flipping the sign of the initial conditions until the leg is in the desired position. We found success with both initial conditions as negative angles as well as both initial conditions with positive angles. It is important to note that the motion generated by the controller is not impacted by undesirable initial conditions but is inverted across the x axis of the animation. However, with the models the initial conditions will impact the result of the legs motion because it is only subject to gravity and is not controlled.

A third recommendation we have is to update the motor gains. Upon consulting with our sponsors, we determined that a simple gain would suffice in the context of our simplified model of the leg. Moving forward, we would highly recommend updating the simple motor gain to a second order motor transfer function to model the system more accurately. Because of the highly tuned nature of the motors, we do not anticipate noticeable change in results. However, we still believe this is an appropriate and realistic next step for this controller.

A fourth recommendation is to focus on testing controller code operation. Our current test includes the time required for MATLAB to solve the system to generate feedback form the “encoders” in each iteration. This is not realistic for measuring the performance speed of the controller alone. We recommend testing controller speed in MATLAB by measuring the time it takes the controller to convert feedback data from the plant model into an output torque to the plant model.

A fifth recommendation is to focus on modifying the system gains in the feedback controller. Our controller currently employs derivative blocks to determine the angular velocity and acceleration required to control our leg. We recommend testing an alternative method to determine the required angular data. Additionally, we recommend updating the feedback linearization constants by selecting different operating parameters for the controller: rise time and peak time or % Overshoot and settling time. We used % overshoot and settling time for our controller, so we recommend determining the constants using rise time and peak time first.

The last major recommendation that we have is to consult the User Reference Manual and both of our sponsors with any questions on linear algebra that drives our models and controller. The User Reference Manual is designed to be a “how to use” guide of our three mathematical models that discusses their functions in greater detail than is presented in this paper. Additionally, we are incredibly grateful to have worked with both of our sponsors as they provided significant guidance on how to derive the feedback linearization equations in our controller.

We are excited to see the continuation of this project and its eventual implementation into physical hardware. We hope to be involved in the next stages of the handoff of this material ,and in ensuring a smooth transition to the next stage. We enjoyed the opportunity to work with a talented group of our peers as well as our stellar sponsors and project coach.

References

1. Sun, Q., Wang, C., Zhao, D., Zhang, C. (2014) Cam Drive Step Mechanism of a Quadruped Robot, Volume 2014(Article ID 851680), 9 pages http://dx.doi.org/10.1155/2014/851680
2. McClamroch, N. M., Wang, D. (1988) Feedback Stabilization and Tacking of Constrained Robots, *IEEE Transactions on Automatic Control*, Volume 33(No. 5),
3. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E. I., Orneas-Rodriguez, M., Puga-Soberanes, H. J., Sotelo-Figuera, M. A., Melin, P. (2016) Quadrupedal Robot Locomotion: A Biologically Inspired Approach and Its Hardware Implementation, (Article ID 5615618), 13 pages
4. *Spot*®. Spot® | Boston Dynamics. https://www.bostondynamics.com/spot
5. Grimminger, F., Meduri, A., Khadiv, M., Viereck, J., Wuthrich, M., Naveau, M., Berenz, V., Heim, S., Widmaier, F., Flayols, T., Fiene, J., Badri-Sprowitz, A., & Righetti, L. (2020). An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research. *IEEE Robotics and Automation Letters*, *5*(2), 3650-3657. [9015985]. https://doi.org/10.1109/LRA.2020.2976639
6. Bledt, G., Powell, M. J., Katz B., Di Carlo, J., Wensing, P. M., Sangbae, K. (2018). MIT Cheetah 3: Design and Control of a Robus, Dynamic Quadruped Robot. *Research Gate* https://www.researchgate.net/publication/329759867
7. Stanford University, Pupper Description, https://stanfordstudentrobotics.org/pupper
8. Stanford University, Pupper Open Source Materials, https://pupper.readthedocs.io/en/latest/
9. Zhejiang University. (2018, December 19). *New version of "Jueying" robot dog released*. ZJU Newsroom. https://www.zju.edu.cn/english/2018/1219/c19573a937920/page.htm.
10. Beer, F. P., Johnston, E. R., Mazurek, D. F., Cornwell, P. J., & Self, B. P. (2016). *Vector mechanics for engineers* (Eleventh). McGraw-Hill.
11. Nise, N. S. (2011). *Control Systems Engineering*. John Wiley & Sons.
12. Owen, F. (2018). *Control systems engineering: a practical approach*. PolyX Publishing.
13. Mathworks, Training Modules, Open Source Materials, https://www.mathworks.com/matlabcentral/fileexchange?q=%22physical+modeling%22
14. Downey, Allen B. *Physical Modeling in MATLAB*. third ed., Green Tea Press, 2011.
15. Mathworks, Importing CAD into SimMechanics, https://www.mathworks.com/videos/importing-a-cad-suspension-assembly-into-simmechanics-89612.html
16. Baringo, Luis, and Antonio J. Conejo. "Back Matter." *Power System Operations*, by Antonio J. Conejo and Luis Baringo, Springer International Publishing, 2018, pp. 271–291.
17. Goode, Stephen W., and Scott Annin. *Differential Equations and Linear Algebra*. Pearson, 2017.
18. Jun Wang, Qingni Hu and Danchi Jiang, "A Lagrangian network for kinematic control of redundant robot manipulators," in *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1123-1132, Sept. 1999, doi: 10.1109/72.788651.
19. Knemeyer, A., Shield, S., Patel, A., Minor Change, Major Gains: The Effect of Orientation Formulation on Solving Time for Multi-Body Trajectory Optimization, *IEEE Robotics and Automation Letters*, vol. 5, no. 4, October 2020.
20. Craig, John J. *Introduction to Robotics: Mechanics and Control*. Third ed., Pearson, 2018.

21. Cho, J., Kong, K. (2020) The Analysis of Mechanical Structure of a Robotic Leg in Running for Impact Mitigation, *Applied Sciences*, Volume 10(Article ID 1365), doi:10.3390/app10041365
22. Hyde, R. A., Wendlandt, J. (Not Stated) Tool-Supported Mechatronic System Design, The MathWorks Inc.
23. Md. Atique, M. U., Md. Sarker, R. I., Md. Ahad, A. R. Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using DenavitHartenberg convention. Heliyon 4 (2018) e01053. doi: 10.1016/j.heliyon.2018. e01053
24. Onose A., Dumitrescu B., Adaptive Cyclic and Randomized Coordinate Descent for the Sparsetotal Least Squares Problem, IEEE, Vol. 015 23rd European Signal Processing Conference, doi: 10.1109/EUSIPCO.2015.7362673
25. Aristidou, A., Lasenby, J., FABRIK: A fast, iterative solver for the Inverse Kinematics problem, IEEE, Graphical Models Vol.73, pp.243-260, doi: 10.1016/j.gmod.2011.05.003
26. X. Yang, S. Song, L. Liu, T. Yan and M. Q. -. Meng, "Design and Optimization of Concentric Tube Robots Based on Surgical Tasks, Anatomical Constraints and Follow-the-Leader Deployment," in IEEE Access, vol. 7, pp. 173612-173625, 2019, doi: 10.1109/ACCESS.2019.2956830.

Appendices

Attachment A : QFD

Correlations	
Positive	+
Negative	−
No Correlation	

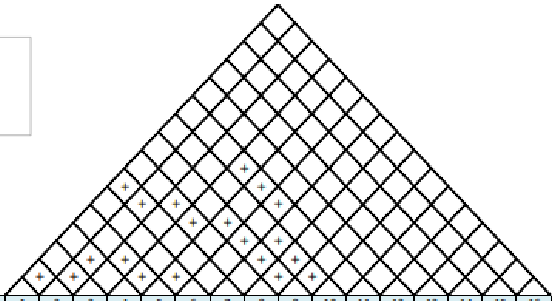
Relationships	
Strong	●
Moderate	○
Weak	▽

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

QFD House of Quality

Project: Quadruped Leg Locomotion

Revision Date: 03 OCT 2020

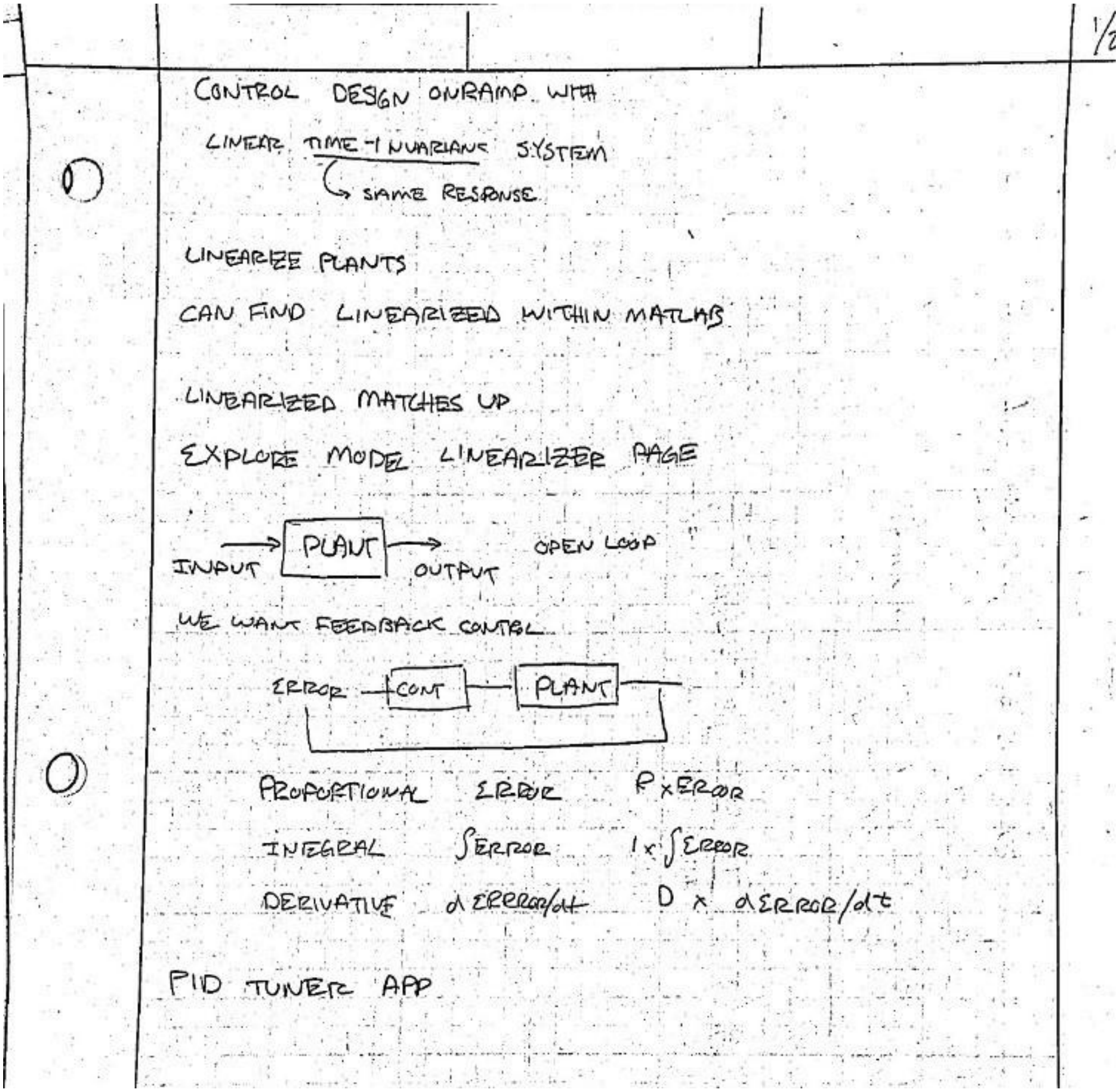


	WHO: Customers							Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
								Direction of Improvement	◇	◇	◇	◇	▲	▲	◇	◇	◇	◇	◇						NOW: Curr. Products					
Row #	Weight Chart	Relative Weight	Simon and Charlie	Future Students	End User		Maximum Relationship	HOW : Engineering Specifications (Tests) WHAT: Customer Requirements (Needs/Wants)	Verified hand calculations	Documented calculations	Documented MATLAB code	MATLAB Simulation	Documented Simscape Model	Simscape model animation	Calculation walkthrough	Code walkthrough	Simscape walkthrough	Connected walkthrough	Simulation matches physical						Boston Dynamics	NYU Solo	MIT Cheetah 3	Stanford - Pupper	Zhejiang University	Row #
1	▮	6%	8	2	5		10	Upward jump	○			●		●					●						4	4	5	3	2	1
2	▮	9%	5	8	9		10	Modifiable Controller	▽	○	●	○	●	▽		○			○						5	4	5	5	3	2
3	▮	11%	10	8	10		10	Forward and backward kinematics	●	▽	○	●		○					●						5	5	5	3	2	3
4	▮	3%	5	2	2		10	Critical force-couple vector	●		●	●	○		▽	▽									5	4	5	3	2	4
5	▮	7%	5	2	10		10	Traverse capability						●					○						5	5	5	2	1	5
6	▮	12%	10	10	10		10	User manual	○	▽			●		○	●	●	○							3	5	3	5	1	6
7	▮	12%	10	10	10		10	Documentation	●	●	●	○	●	○	●	●	●	●							5	5	4	5	1	7
8	▮	9%	10	4	10		10	Repeatability	▽	▽	▽	▽	▽	▽	▽										4	5	5	3	1	8
9	▮	9%	8	7	7		10	Ease of maintenance	▽	▽	○	▽	▽	▽	○	○	○	●							2	3	3	5	3	9
10	▮	8%	8	7	5		10	Ease of set up			○				●	●	●	●							2	3	2	5	3	10
11	▮	5%	6	4	2		10	Measuring position (ang. and rect.)	●	○	●	▽	▽	▽	○	○	○	○							4	4	5	2	2	11
12	▮	5%	6	4	2		10	Measuring velocity (ang. and rect.)	●	○	●	▽	▽	▽	○	○	○	○							4	4	5	2	2	12
13	▮	5%	6	4	2		10	Measuring acceleration (ang. and rect.)	●	○	●	▽	▽	▽	○	○	○	○							4	4	5	2	2	13
14		0%																													14
15		0%																													15
16		0%																													16

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	HOW MUCH: Target Values							5 mintue video or 5 page doc	5 mintue video or 5 page doc	5 mintue video or 5 page doc	5 mintue video or 5 page doc						
	Max Relationship	9	9	9	9	9	9	9	9	9	9	9					
	Technical Importance Rating	446.4	218.8	440.1	277.1	340	221	297.5	387.5	357.1	337.5	197.7	0	0	0	0	0
	Relative Weight	13%	6%	13%	8%	10%	6%	8%	11%	10%	10%	6%	0%	0%	0%	0%	0%
Curr. Products	Boston Dynamics	5	5	5	4	2	2	5	5	4	3	5					
	MIT Cheetah 3	5	4	3	2	2	2	5	2	3	3	5					
	Stanford - Pupper	5	5	4	2	2	2	3	5	1	3	5					
	Zhejiang University	5	1	2	2	2	1	1	1	1	1	5					
	Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Attachment B : Simulink Tutorials

1. Control Design OnRamp



1/2

CONTROL DESIGN ONRAMP WITH

LINEAR TIME-INVARIANC SYSTEM

→ SAME RESPONSE

LINEARIZE PLANTS

CAN FIND LINEARIZED WITHIN MATLAB

LINEARIZED MATCHES UP

EXPLORE MODEL LINEARIZER PAGE

INPUT → PLANT → OUTPUT

OPEN LOOP

WE WANT FEEDBACK CONTRL

ERROR — CONT — PLANT —

PROPORTIONAL ERROR P×ERROR

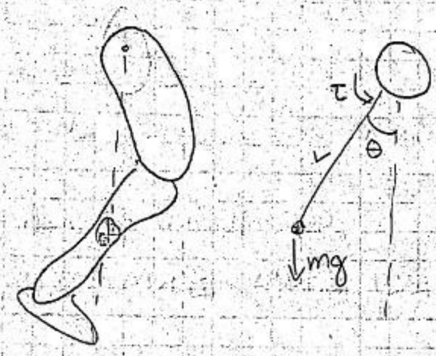
INTEGRAL ∫ERROR I×∫ERROR

DERIVATIVE dERROR/dt D × dERROR/dt

PID TUNER APP

1. Control Design OnRamp

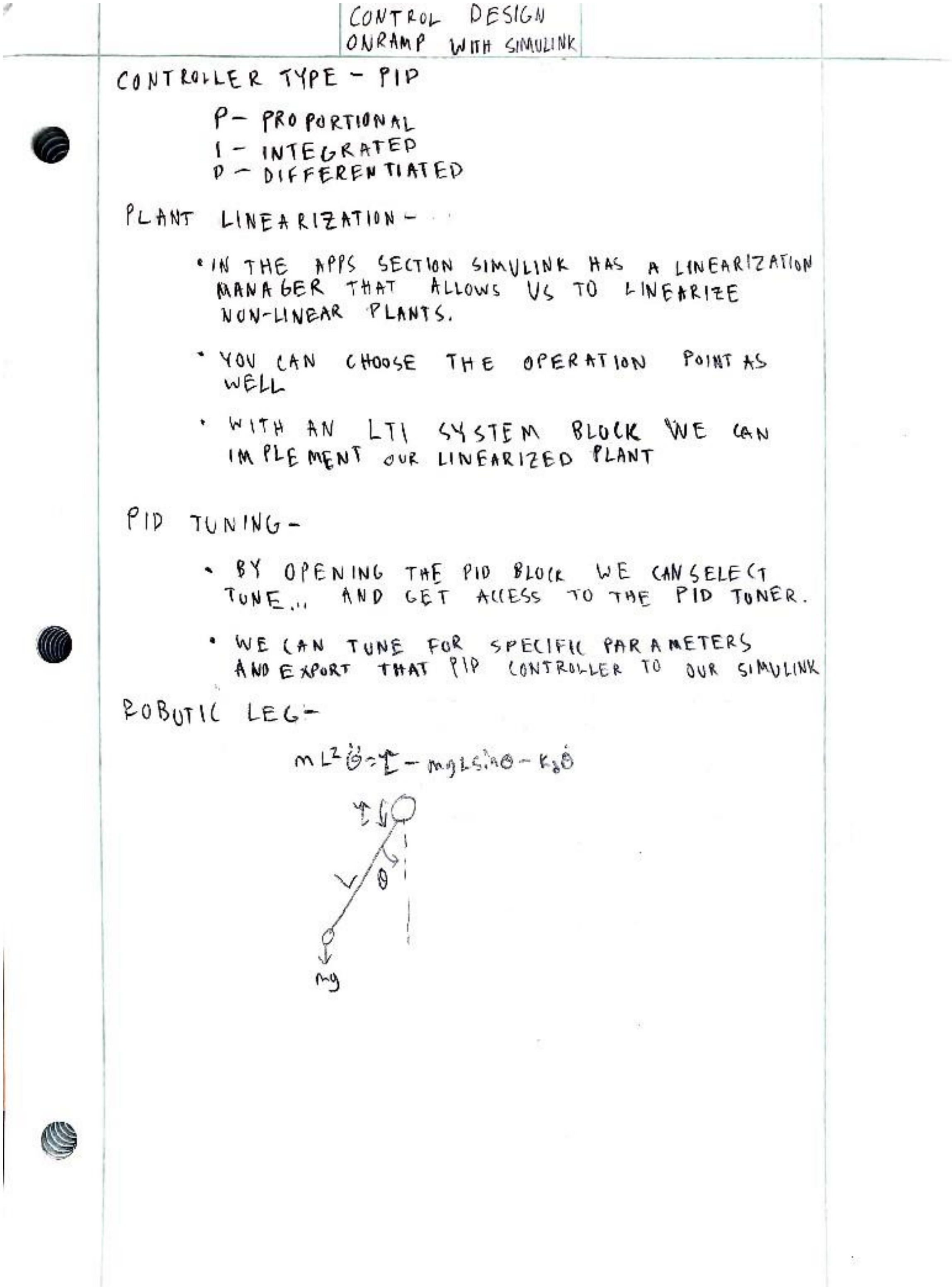
ROBOTIC LEG CONTROL



$$mL^2\ddot{\theta} = \tau - mgL\sin\theta - K_d\dot{\theta}$$

WE WANT LESS THAN .8 SETTLING TIME , AND <10% OVERSHOOT

2. Control Design PID



CONTROL DESIGN ONRAMP WITH SIMULINK

CONTROLLER TYPE - PID

P - PROPORTIONAL
I - INTEGRATED
D - DIFFERENTIATED

PLANT LINEARIZATION -

- IN THE APPS SECTION SIMULINK HAS A LINEARIZATION MANAGER THAT ALLOWS US TO LINEARIZE NON-LINEAR PLANTS.
- YOU CAN CHOOSE THE OPERATION POINT AS WELL
- WITH AN LTI SYSTEM BLOCK WE CAN IMPLEMENT OUR LINEARIZED PLANT

PID TUNING -

- BY OPENING THE PID BLOCK WE CAN SELECT TUNE... AND GET ACCESS TO THE PID TUNER.
- WE CAN TUNE FOR SPECIFIC PARAMETERS AND EXPORT THAT PID CONTROLLER TO OUR SIMULINK

ROBOTIC LEG -

$$mL^2\ddot{\theta} = \tau - mgL\sin\theta - k_d\dot{\theta}$$

Attachment C : Forward Kinematic Code

This section of code establishes all the variables and constants for the system

```
 % global allows for the same variables to be used throughout all function
 % files containing global "variable name"`

 global g % Gravity
 g = 9.81;

 % Shank info
 global r2 % Shank Length
 r2 = 20; % Selected by designer

 % Changes made here only affect kinetic simulation
 global mD % Shank mass
 mD = 1; % Selected by designer
 global ID % Shank moment of inertia
 ID = 1; % Selected by designer
 global MA % Moment applied to the knee at the top of the shank
 MA = -60; % Selected by designer

 % Changes made here only affect kinematic simulation
 global th2 % Angle of shank measured from right horizontal in the counter
clockwise direction
 th2 = deg2rad(60); % Selected by designer
 global th_dot2 % Angular velocity of shank in the counter clockwise direction
 th_dot2 = 5*((2*pi)/60); % Selected by designer
 global th_ddot2 % Angular acceleration of shank in the counter clockwise
direction
 th_ddot2 = -10*((2*pi)/3600); % Selected by designer

 % Thigh info
 global r3 % Thigh Length
 r3 = 20; % Selected by designer

 % Changes made here only affect kinetic simulation
 global mO
 mO = 2;
 global IO % Thigh moment of inertia
 IO = 1; % Selected by designer
 global MO % Moment applied to the hip at the top of the thigh
 MO = -40; % Selected by designer
 global X_ddotO % Initial acceleration of the robot body in the X-direction
 X_ddotO = -5; % Selected by designer
```

```
global Y_ddotO % Initial acceleration of the robot body in the Y-direction
Y_ddotO = -9.8; % Selected by designer

% Changes made here only affect kinetic simulation
global th3 % Thigh angle measeured from right horizontal in the counter clockwise direction
th3 = deg2rad(-45); % Selected by designer
global th_dot3 % Thigh angular velocity in the counter clockwise direction
th_dot3 = -10*((2*pi)/60); % Selected by designer
global th_ddot3 % Thigh angular acceleration in the counter clockwise direction
th_ddot3 = 10*((2*pi)/3600); % Selected by designer
```

This section is for determining the initial position conditions for the mechanism based on the choices for r2, r3, th2, and th3. fminsearch minimizes min1.mlx so that the vector loop is complete with three sections (r1, r2, and r3). It gives the length of r1 and the angle of th1.

```
% guess whatever to give the model somewhere to start
IC_guess = [24
            160];
IC_actual1 = fminsearch(@min1, IC_guess); % Gives the real Initial position conditions based on minimazing min1
```

Kinematics of the Airborne scenario:

FreeMech returns the initial linear velocity, rotational velocity, linear acceleration, and rotational acceleration of r1. This is based upon the initial position conditions derived from above.

```
% gives the initial velocity and acceleration conditions of the the system
[x_dot,x_ddot] = FreeMech(IC_actual1);
% Motion relative to the ground
a_o = [0
       0];

v_io = [-1
        0];

th1 = deg2rad(IC_actual1(2));
s1 = sin(th1);
c1 = cos(th1);
r_io = [-IC_actual1(1)*c1
        -IC_actual1(1)*s1];
```

The simulation is run, and the data is reorganized below.

```
% simulation data is stored in variable out
out = sim("UnconstrainedNoContact.slx");
```

```
 % Positional info for simulation
 s2 = sin(out.th2);
 c2 = cos(out.th2);
 s3 = sin(out.th3);
 c3 = cos(out.th3);

 % Animation for verification
 hf5 = figure(5);
 hf5.Visible = 'on';
 hf5.WindowState = 'maximized';
 for i = 1:length(out.tout)
            % O A        B
     y_pos = [out.r_o(i,2) out.r_o(i,2)+s3(i)*r3 out.r_o(i,2)+s3(i)*r3-
s2(i)*r2];
     x_pos = [out.r_o(i,1) out.r_o(i,1)+c3(i)*r3 out.r_o(i,1)+c3(i)*r3-
c2(i)*r2];
     plot(x_pos, y_pos);
     hold on
     plot(out.r_o(1:i,1),out.r_o(1:i,2));
     plot(out.r_o(1:i,1)+c3(1:i)*r3,out.r_o(1:i,2)+s3(1:i)*r3,"r");
     plot(out.r_o(1:i,1)+c3(1:i)*r3-c2(1:i)*r2,out.r_o(1:i,2)+s3(1:i)*r3-
s2(1:i)*r2,"g");
     xlabel('Horizontal Position [mm]');
     ylabel('Vertical Position [mm]');
     hold off
     axis('equal');
     xlim([-100 100]);
     ylim([-100 100]);
     drawnow;
     frame = getframe(hf5); % this is for saving the gif
     im{i} = frame2im(frame); % this is for saving the gif
 end
```

This creates a gif for sharing the animation

```
 filename = 'Mech1.gif';
 for i = 1:length(out.tout)
     [A, map] = rgb2ind(im{i},256);
     if i == 1
         imwrite(A, map, filename, 'gif', 'LoopCount', Inf, 'DelayTime', 0.1);
     else
         imwrite(A, map, filename, 'gif', 'WriteMode','append',
'DelayTime',0.1);
     end
 end
```

Kinematics of the Grounded scenario:

```
% The initial velocity and accerleration conditions for the constrained and
contact kinematics simulation
[x_dot,x_ddot] = ConsMech(IC_actual1);
% Converts the initial positional conditions for the unconstrained no
% contact kinematic simulation to the initial positional conditions of the
% constrained and contact kinematic simulation.
IC_actual3 = IC_actual1;
IC_actual3(3,1) = th2;

th1 = deg2rad(IC_actual3(2));
s1 = sin(th1);
c1 = cos(th1);
r_io = [-IC_actual3(1)*c1
        -IC_actual3(1)*s1];
```

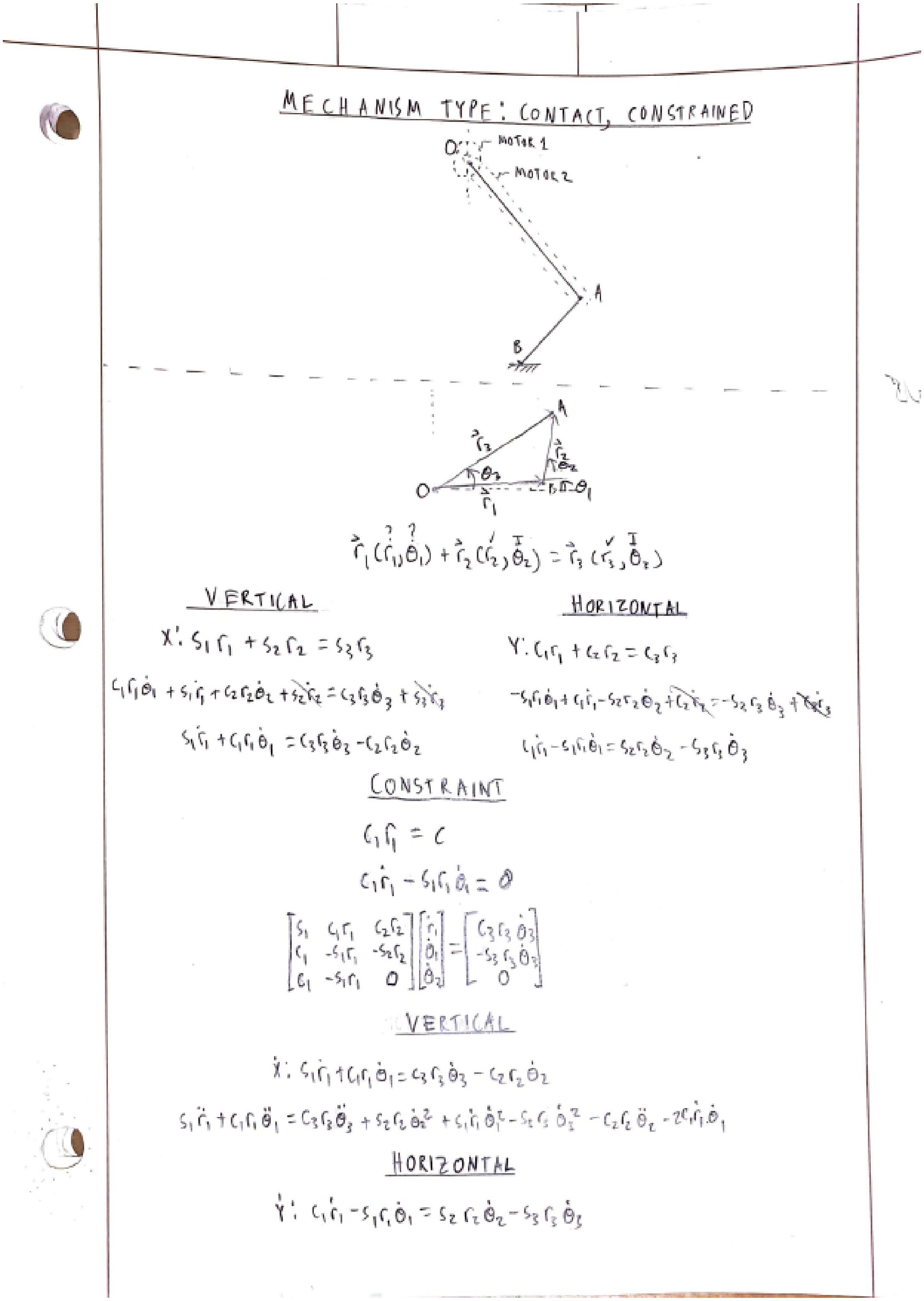
The simulation is run, and the data is reorganized below.

```
% Stores the simulation data
out = sim("ConstrainedContact.slx");

% Positional info for the simulation
s1 = sin(out.x(:,2));
c1 = cos(out.x(:,2));
s2 = sin(out.x(:,3));
c2 = cos(out.x(:,3));
s3 = sin(out.th3);
c3 = cos(out.th3);


% Animation for verification
hf6 = figure(6);
hf6.Visible = 'on';
hf6.WindowState = 'maximized';

for i = 1:length(out.tout)
           % B A        O
    y_pos = [0 s2(i)*r2 s2(i)*r2-s3(i)*r3];
    x_pos = [0 c2(i)*r2 c2(i)*r2-c3(i)*r3];
    plot(x_pos, y_pos);
    hold on
    plot(c2(1:i)*r2,s2(1:i)*r2,"r");
    plot(c2(1:i)*r2-c3(1:i)*r3,s2(1:i)*r2-s3(1:i)*r3,"g");
    xlabel('Horizontal Position [mm]');
```

```
     ylabel('Vertical Position [mm]');
     hold off
     axis('equal');
     xlim([-100 200]);
     ylim([-100 100]);
     drawnow;
     frame = getframe(hf6); % this is for saving the gif
     im{i} = frame2im(frame); % this is for saving the gif
 end
```

```
 % Produces a gif of the animation
 filename = 'Mech3.gif';
 for i = 1:length(out.tout)
     [A, map] = rgb2ind(im{i},256);
     if i == 1
         imwrite(A, map, filename, 'gif', 'LoopCount', Inf, 'DelayTime', 0.1);
     else
         imwrite(A, map, filename, 'gif', 'WriteMode','append',
'DelayTime',0.1);
     end
 end
```

Attachment D : Forward Kinematics and Kinetics Hand Calcs



MECHANISM TYPE: CONTACT, CONSTRAINED

O
MOTOR 1
MOTOR 2
A
B
A
$\vec{r}_3$
$\vec{r}_2$
θ_2
θ_3
O
$\vec{r}_1$
B
$\pi-\theta_1$

$$\vec{r}_1(\overset{?}{\dot{r}_1},\overset{?}{\dot{\theta}_1}) + \vec{r}_2(\overset{/}{\dot{r}_2},\overset{I}{\dot{\theta}_2}) = \vec{r}_3(\overset{\checkmark}{\dot{r}_3},\overset{I}{\dot{\theta}_3})$$

VERTICAL

$$X': s_1r_1 + s_2r_2 = s_3r_3$$

$$c_1r_1\dot{\theta}_1 + s_1\dot{r}_1 + c_2r_2\dot{\theta}_2 + \cancel{s_2\dot{r}_2} = c_3r_3\dot{\theta}_3 + \cancel{s_3\dot{r}_3}$$

$$s_1\dot{r}_1 + c_1r_1\dot{\theta}_1 = c_3r_3\dot{\theta}_3 - c_2r_2\dot{\theta}_2$$

HORIZONTAL

$$Y': c_1r_1 + c_2r_2 = c_3r_3$$

$$-s_1r_1\dot{\theta}_1 + c_1\dot{r}_1 - s_2r_2\dot{\theta}_2 + \cancel{c_2\dot{r}_2} = -s_2r_3\dot{\theta}_3 + \cancel{c_3\dot{r}_3}$$

$$c_1\dot{r}_1 - s_1r_1\dot{\theta}_1 = s_2r_2\dot{\theta}_2 - s_3r_3\dot{\theta}_3$$

CONSTRAINT

$$c_1r_1 = c$$

$$c_1\dot{r}_1 - s_1r_1\dot{\theta}_1 = 0$$

$$\begin{bmatrix} s_1 & c_1r_1 & c_2r_2 \\ c_1 & -s_1r_1 & -s_2r_2 \\ c_1 & -s_1r_1 & 0 \end{bmatrix}\begin{bmatrix} \dot{r}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} c_3r_3\dot{\theta}_3 \\ -s_3r_3\dot{\theta}_3 \\ 0 \end{bmatrix}$$

VERTICAL

$$\dot{X}: s_1\dot{r}_1 + c_1r_1\dot{\theta}_1 = c_3r_3\dot{\theta}_3 - c_2r_2\dot{\theta}_2$$

$$s_1\ddot{r}_1 + c_1r_1\ddot{\theta}_1 = c_3r_3\ddot{\theta}_3 + s_2r_2\dot{\theta}_2^2 + s_1r_1\dot{\theta}_1^2 - s_3r_3\dot{\theta}_3^2 - c_2r_2\ddot{\theta}_2 - 2c_1\dot{r}_1\dot{\theta}_1$$

HORIZONTAL

$$\dot{Y}: c_1\dot{r}_1 - s_1r_1\dot{\theta}_1 = s_2r_2\dot{\theta}_2 - s_3r_3\dot{\theta}_3$$

$$c_1\ddot{r}_1 - s_1 r_1 \ddot{\theta}_1 = s_2 r_2 \ddot{\theta}_2 + c_2 r_2 \dot{\theta}_2^2 + 2 s_1 \dot{r}_1 \dot{\theta}_1 + c_1 r_1 \dot{\theta}_1^2 - s_3 r_3 \ddot{\theta}_3 - c_3 r_3 \dot{\theta}_3^2$$

CONSTRAINT

$$c_1 \dot{r}_1 - s_1 r_1 \dot{\theta}_1 = 0$$

$$-s_1 \dot{r}_1 \dot{\theta}_1 + c_1 \ddot{r}_1 - s_1 r_1 \ddot{\theta}_1 - c_1 r_1 \dot{\theta}_1^2 - s_1 \dot{r}_1 \dot{\theta}_1 = 0$$

$$c_1 \ddot{r}_1 - s_1 r_1 \ddot{\theta}_1 = 2 s_1 \dot{r}_1 \dot{\theta}_1 + c_1 r_1 \dot{\theta}_1^2$$

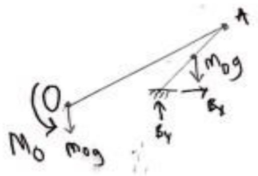
$$\begin{bmatrix} s_1 & c_1 r_1 & c_2 r_2 \\ c_1 & -s_1 r_1 & -s_2 r_2 \\ c_1 & -s_1 r_1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} c_3 r_3 \ddot{\theta}_3 + s_2 r_2 \dot{\theta}_2^2 + s_1 r_1 \dot{\theta}_1^2 - s_3 r_3 \dot{\theta}_3^2 - 2 c_1 \dot{r}_1 \dot{\theta}_1 \\ c_2 r_2 \dot{\theta}_2^2 + 2 s_1 \dot{r}_1 \dot{\theta}_1 + c_1 r_1 \dot{\theta}_1^2 - s_2 r_3 \ddot{\theta}_3 - c_3 r_3 \dot{\theta}_3^2 \\ c_1 r_1 \dot{\theta}_1^2 + 2 s_1 \dot{r}_1 \dot{\theta}_1 \end{bmatrix}$$

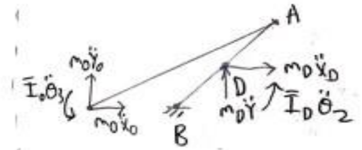
MECHANISM TYPE: CONTACT, CONSTRAINED

KINETICS

FBD

KD



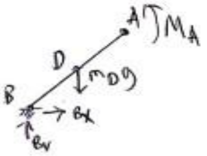


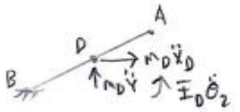
$$\Sigma M_B = M_O - m_Og(\tfrac{1}{2}c_2r_2) - m_Og(c_2r_2 - c_3r_3)$$

$$\Sigma M_B = \bar{I}_D\ddot{\theta}_2 + \bar{I}_O\ddot{\theta}_3 - m_D\ddot{x}_D(\tfrac{1}{2}s_2r_1) + m_D\ddot{y}_D(\tfrac{1}{2}c_2r_2) - m_O\ddot{x}_O(s_2r_2 - s_3r_3) + m_O\ddot{y}_O(c_2r_2 - c_3r_3)$$

FBD

KD





$$\Sigma M_B = M_A - m_Dg(\tfrac{1}{2}c_2r_2)$$

$$\Sigma M_B = \bar{I}_D\ddot{\theta}_2 - m_D\ddot{x}_D(\tfrac{1}{2}s_2r_2) + m_D\ddot{y}_D(\tfrac{1}{2}c_2r_2)$$

KINEMATICS

$$\ddot{x}_D = -\tfrac{1}{2}\ddot{\theta}_2 s_2 r_2 - \tfrac{1}{2}c_2 r_2 \dot{\theta}_2^2$$

$$\ddot{y}_D = \tfrac{1}{2}\ddot{\theta}_2 c_2 r_2 - \tfrac{1}{2}s_2 r_2 \dot{\theta}_2^2$$

$$\ddot{x}_O = \ddot{\theta}_1 s_1 r_1 + c_1 r_1 \dot{\theta}_1^2 - c_1 \ddot{r}_1 + 2 s_1 \dot{r}_1 \dot{\theta}_1 = 0$$

$$\ddot{y}_O = -\ddot{\theta}_1 c_1 r_1 + s_1 r_1 \dot{\theta}_1^2 - s_1 \ddot{r}_1 - 2 c_1 \dot{r}_1 \dot{\theta}_1$$

$$\underset{\underline{\ddot{\chi}}}{\begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{x}_D \\ \ddot{y}_D \\ \ddot{x}_0 \\ \ddot{y}_0 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{2}s_2r_2 & 0 \\ 0 & 0 & \frac{1}{2}c_2r_2 & 0 \\ -c_1 & s_1r_1 & 0 & 0 \\ -s_1 & -c_1r_1 & 0 & 0 \end{bmatrix}} \underset{\underline{\ddot{x}}}{\begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}} + \underset{a}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{2}c_2r_2\dot{\theta}_2^2 \\ -\frac{1}{2}s_2r_2\dot{\theta}_2^2 \\ c_1r_1\dot{\theta}_1^2 + 2s_1\dot{r}_1\dot{\theta}_1 \\ s_1r_1\dot{\theta}_1^2 - 2c_1\dot{r}_1\dot{\theta}_1 \end{bmatrix}}$$

$$f_2 = M_2\ddot{\chi}_2$$

$$f_2 = M_2(A_2\ddot{x} + a_2)$$

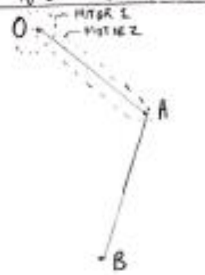
$$f_2 = M_2A_2\ddot{x} + M_2a_2$$

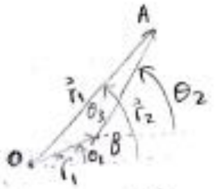
$$\ddot{x} = (M_2A_2)^{-1}(f_2 - M_2a_2)$$

$$\begin{bmatrix} s_2r_2\dot{\theta}_2^2 + s_1\dot{r}_1\dot{\theta}_1^2 - s_3r_3\dot{\theta}_3^2 - 2c_1\dot{r}_1\dot{\theta}_1 \\ c_2r_2\dot{\theta}_2^2 + 2s_1\dot{r}_1\dot{\theta}_1 + c_1r_1\dot{\theta}_1^2 - c_3r_3\dot{\theta}_3^2 \\ c_1r_1\dot{\theta}_1^2 + 2s_1\dot{r}_1\dot{\theta}_1 \\ M_0 - m_Dg(\frac{1}{2}c_2r_2) - m_0g(c_2r_2 - c_3r_3) \\ M_A - m_Dg(\frac{1}{2}c_2r_2) \end{bmatrix} = f_2$$

$$M_2 = \begin{bmatrix} s_1 & c_1r_1 & c_2r_2 & -c_3r_3 & 0 & 0 & 0 & 0 \\ c_1 & -s_1r_1 & -s_2r_2 & s_3r_3 & 0 & 0 & 0 & 0 \\ c_1 & -s_1r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{I}_D & \bar{I}_0 & -m_D(\frac{1}{2}s_2r_2) & m_D(\frac{1}{2}c_2r_2) & -m_0(s_2r_2 - s_3r_3) & m_0(c_2r_2 - c_3r_3) \\ 0 & 0 & \bar{I}_D & 0 & -m_D(\frac{1}{2}s_2r_2) & m_D(\frac{1}{2}c_2r_2) & 0 & 0 \end{bmatrix}$$

MECHANISM TYPE: NO CONTACT, UNCONSTRAINED





$$\vec r_1(\overset{?}{\dot r_1}, \overset{?}{\dot\theta_1}) + \vec r_2(\overset{\checkmark}{\dot r_2}, \overset{\checkmark}{\dot\theta_2}) = \vec r_3(\overset{\checkmark}{\dot r_3}, \overset{\checkmark}{\dot\theta_3})$$

VERTICAL:

$Y: s_1 r_1 + s_2 r_2 = s_2 r_3$

$c_1 r_1 \dot\theta_1 + s_1 \dot r_1 + c_2 r_2 \dot\theta_2 + \cancel{s_2 \dot r_2} = c_3 r_3 \dot\theta_3 + \cancel{s_3 \dot r_3}$

$c_1 r_1 \dot\theta_1 + s_1 \dot r_1 = c_3 r_3 \dot\theta_3 - c_2 r_2 \dot\theta_2$

HORIZONTAL:

$X: c_1 r_1 + c_2 r_2 = c_3 r_3$

$-s_1 r_1 \dot\theta_1 + c_1 \dot r_1 - s_2 r_2 \dot\theta_2 + \cancel{c_2} \dot r_2 = -s_3 r_3 \dot\theta_3 + \cancel{c_3} \dot r_3$

$c_1 \dot r_1 - s_1 r_1 \dot\theta_1 = s_2 r_2 \dot\theta_2 - s_3 r_3 \dot\theta_3$

$$\begin{bmatrix} s_1 & c_1 r_1 \\ c_1 & -s_1 r_1 \end{bmatrix} \begin{bmatrix} \dot r_1 \\ \dot\theta_1 \end{bmatrix} = \begin{bmatrix} c_3 r_3 \dot\theta_3 - c_2 r_2 \dot\theta_2 \\ s_2 r_2 \dot\theta_2 - s_3 r_3 \dot\theta_3 \end{bmatrix}$$

VERTICAL:

$\dot Y\ c_1 r_1 \dot\theta_1 + s_1 \dot r_1 = c_3 r_3 \dot\theta_3 - c_2 r_2 \dot\theta_2$

$c_1 \dot r_1 \dot\theta_1 - s_1 \dot r_1 \dot\theta_1^2 + \underline{c_1 r_1 \ddot\theta_1} + \underline{s_1 \ddot r_1} + c_1 \dot r_1 \dot\theta_1 = -s_3 r_3 \dot\theta_3^2 + c_3 r_3 \ddot\theta_3 + s_2 r_2 \dot\theta_2^2 - c_2 r_2 \ddot\theta_2$

$c_1 r_1 \ddot\theta_1 + s_1 \ddot r_1 = c_3 r_3 \ddot\theta_3 + s_2 r_2 \dot\theta_2^2 + s_1 \dot r_1 \dot\theta_1^2 - s_3 r_3 \dot\theta_3^2 - c_2 r_2 \ddot\theta_2 - 2c_1 \dot r_1 \dot\theta_1$

HORIZONTAL:

$\dot X: c_1 \dot r_1 - s_1 r_1 \dot\theta_1 = s_2 r_2 \dot\theta_2 - s_3 r_3 \dot\theta_3$

$\underline{c_1 \ddot r_1} - s_1 \dot r_1 \dot\theta_1 - s_1 \dot r_1 \dot\theta_1 - c_1 r_1 \dot\theta_1^2 - \underline{s_1 r_1 \ddot\theta_1} = s_2 r_2 \ddot\theta_2 + c_2 r_2 \dot\theta_2^2 - s_3 r_3 \ddot\theta_3 - c_3 r_3 \dot\theta_3^2$

$c_1 \ddot r_1 - s_1 r_1 \ddot\theta_1 = s_2 r_2 \ddot\theta_2 + c_2 r_2 \dot\theta_2^2 + 2s_1 \dot r_1 \dot\theta_1 + c_1 r_1 \dot\theta_1^2 - s_3 r_3 \ddot\theta_3 - c_3 r_3 \dot\theta_3^2$

$$\begin{bmatrix} s_1 & c_1 r_1 \\ c_1 & -s_1 r_1 \end{bmatrix}\begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \end{bmatrix} = \begin{bmatrix} c_3 r_3 \ddot{\theta}_3 + s_2 r_2 \dot{\theta}_2^2 + s_1 \dot{r}_1 \dot{\theta}_1^2 - s_3 r_3 \dot{\theta}_3^2 - c_2 r_2 \ddot{\theta}_2 - 2 c_1 \dot{r}_1 \dot{\theta}_1 \\ s_2 r_2 \ddot{\theta}_2 + c_2 r_2 \dot{\theta}_2^2 + 2 s_1 \dot{r}_1 \dot{\theta}_1 + c_1 r_1 \dot{\theta}_1^2 - s_3 r_3 \ddot{\theta}_3 - c_3 r_3 \dot{\theta}_3^2 \end{bmatrix}$$

MOTION AND POSITIONS OF O, A, & B

O: POSITION – $\vec{r}_0$ X: r_{0x} Y: r_{0y}

VELOCITY – $\vec{v}_0$ $\dot{X}$: v_{0x} $\dot{Y}$: v_{0y}

ACCELERATION – $\vec{a}_0$ $\ddot{X}$: a_{0x} $\ddot{Y}$: a_{0y}

A: POSITION – $\vec{r}_0 + \vec{r}_3$ X: $r_{0x} + s_3 r_3$ Y: $r_{0y} + c_3 r_3$

VELOCITY – $\vec{v}_0 + \dot{\theta}_3 \times \vec{r}_3$

ACCELERATION – $\vec{a}_0 + \ddot{\theta}_3 \times \vec{r}_3 + \dot{\theta}_3 \times (\dot{\theta}_3 \times \vec{r}_3)$

B: POSITION – $\vec{r}_0 + \vec{r}_1$ X: $r_{0x} + s_1 r_1$ Y: $r_{0y} + s_1 r_1$

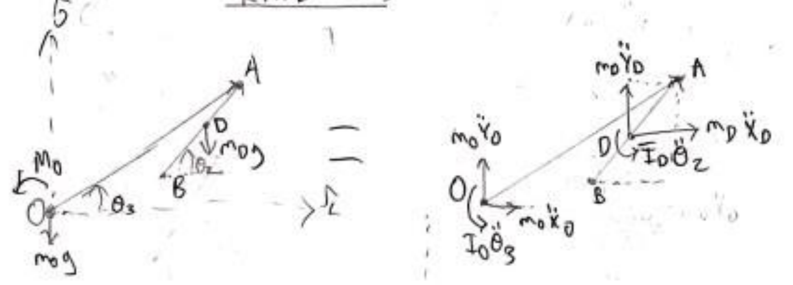
VELOCITY – $\vec{v}_0 + \dot{\theta}_1 \times \vec{r}_1 + \dot{\vec{r}}_1$

ACCELERATION – $\vec{a}_0 + \ddot{\theta}_1 \times \vec{r}_1 + \dot{\theta}_1 \times (\dot{\theta}_1 \times \vec{r}_1) + 2\dot{\theta}_1 \times \dot{\vec{r}}_1 + \ddot{\vec{r}}_1$

MECHANISM TYPE: NO CONTACT, UNCONSTRAINED

KINETICS

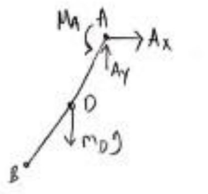
FBD   KD



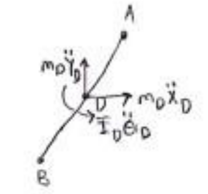
$$\Sigma M_O = M_o - m_D g\,(C_3 r_3 - \tfrac{1}{2} C_2 r_2)$$

$$\Sigma M_O = \bar{I}_O \ddot{\theta}_3 + \bar{I}_D \ddot{\theta}_2 + m_D \ddot{Y}_D (C_3 r_3 - \tfrac{1}{2} C_2 r_2) - m_D \ddot{X}_D (S_3 r_3 - \tfrac{1}{2} S_2 r_2)$$

FBD



KD



$$\Sigma M_A = M_A + m_D g\,(\tfrac{1}{2} C_2 r_2)$$

$$\Sigma M_A = \bar{I}_D \ddot{\theta}_2 + m_D \ddot{X}_D (\tfrac{1}{2} S_2 r_2) - m_D \ddot{Y}_D (\tfrac{1}{2} C_2 r_2)$$

KINEMATICS:

$$\ddot{X}_A = \ddot{X}_O - \ddot{\theta}_3 S_3 r_3 - C_3 r_3 \dot{\theta}_3^2$$

$$\ddot{Y}_A = \ddot{Y}_O + \ddot{\theta}_3 C_3 r_3 - S_3 r_3 \dot{\theta}_3^2$$

$$\ddot{X}_D = \ddot{X}_A + \tfrac{1}{2} \ddot{\theta}_2 S_2 r_2 + \tfrac{1}{2} C_2 r_2 \dot{\theta}_2^2$$

$$\ddot{Y}_D = \ddot{Y}_A - \tfrac{1}{2} \ddot{\theta}_2 C_2 r_2 + \tfrac{1}{2} S_2 r_2 \dot{\theta}_2^2$$

$$\ddot{X}_D = \ddot{X}_O - \ddot{\theta}_3 S_3 r_3 - C_3 r_3 \dot{\theta}_3^2 + \tfrac{1}{2} \ddot{\theta}_2 S_2 r_2 + \tfrac{1}{2} C_2 r_2 \dot{\theta}_2^2$$

$$\ddot{Y}_D = \ddot{Y}_O + \ddot{\theta}_3 C_3 r_3 - S_3 r_3 \dot{\theta}_3^2 - \tfrac{1}{2} \ddot{\theta}_2 C_2 r_2 + \tfrac{1}{2} S_2 r_2 \dot{\theta}_2^2$$

$$\underset{\underline{\ddot{X}}}{\begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{X}_D \\ \ddot{Y}_D \end{bmatrix}} = \underset{A}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2}s_2r_2 & -s_3r_3 \\ 0 & 0 & -\frac{1}{2}c_2r_2 & c_3r_3 \end{bmatrix}} \underset{\underline{\ddot{x}}}{\begin{bmatrix} \ddot{r}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}} + \underset{a}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \ddot{X}_D - c_3r_3\dot{\theta}_3^2 + \frac{1}{2}c_2r_2\dot{\theta}_2^2 \\ \ddot{Y}_D - s_3r_3\dot{\theta}_3^2 + \frac{1}{2}s_2r_2\dot{\theta}_2^2 \end{bmatrix}}$$

$$f_1 = M_1\ddot{X}_1$$

$$f_1 = M_1(A_1\ddot{x} + a_1)$$

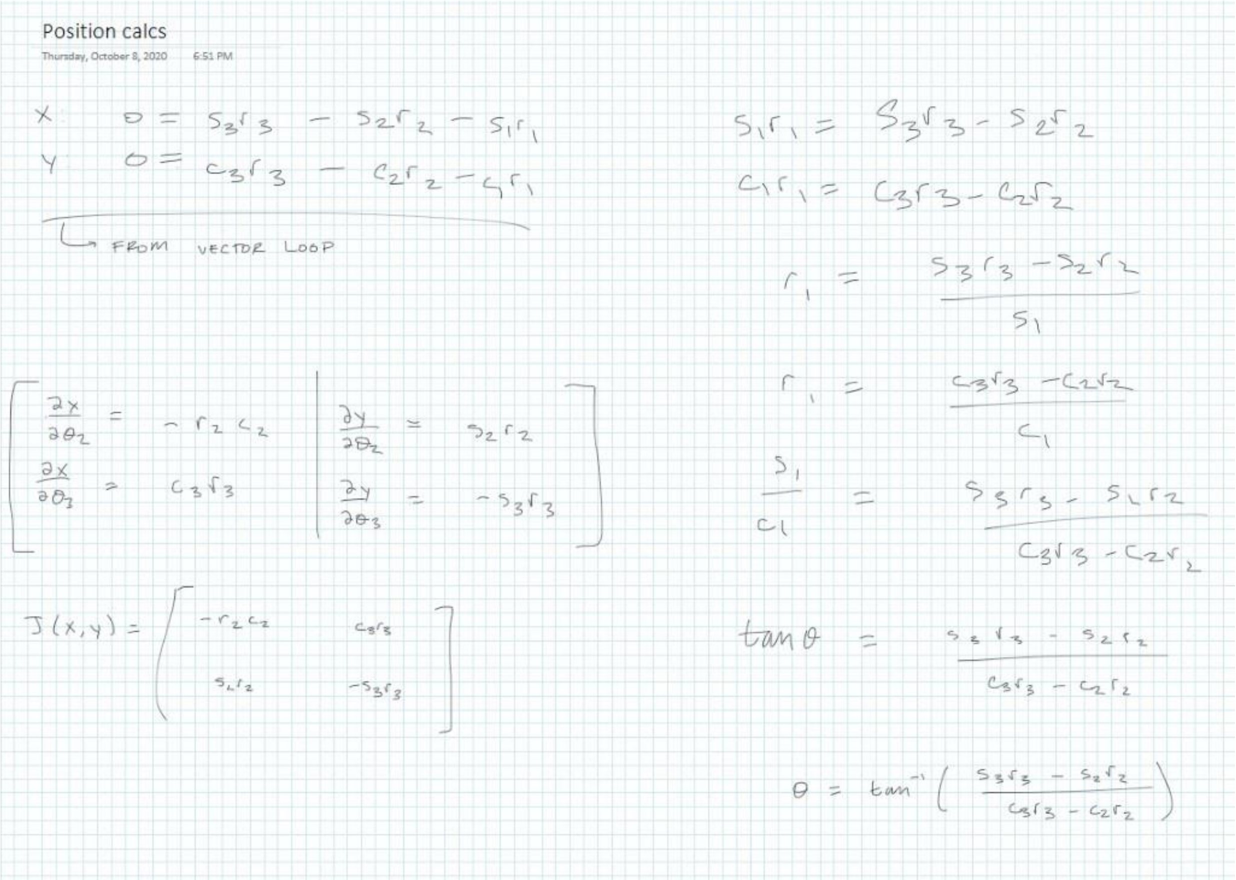
$$f_1 = M_1A_1\ddot{x} + M_1a_1$$

$$\ddot{x} = (M_1A_1)^{-1}(f_1 - M_1a_1)$$

$$\begin{bmatrix} s_2r_2\dot{\theta}_2^2 + s_1\dot{r}_1\dot{\theta}_1^2 - s_3r_3\dot{\theta}_3^2 \quad -2c_1\dot{r}_1\dot{\theta}_1 \\ c_2r_2\dot{\theta}_2^2 + 2s_1\dot{r}_1\dot{\theta}_1 + c_1r_1\dot{\theta}_1^2 \quad -c_3r_3\dot{\theta}_3^2 \\ M_D - m_Dg(c_3r_3 - 1/2\,c_2r_2) \\ M_A + m_Dg(1/2\,c_2r_2) \end{bmatrix} = f_1$$

$$\begin{bmatrix} s_1 & c_1r_1 & c_2r_2 & -c_3r_3 & 0 & 0 \\ c_1 & -s_1r_1 & -s_2r_2 & s_3r_3 & 0 & 0 \\ 0 & 0 & I_D & J_0 & -m_D(s_2r_3 - 1/2\,s_2r_2) & m_D(c_3r_3 - 1/2\,c_2r_2) \\ 0 & 0 & I_D & \theta & m_D(1/2\,s_2r_2) & -m_D(1/2\,c_2r_2) \end{bmatrix} = M_1$$

Attachment E: Inverse Kinematics Hand Calc



Position calcs

Thursday, October 8, 2020 6:51 PM

$X: \quad 0 = s_3 r_3 - s_2 r_2 - s_1 r_1$

$Y: \quad 0 = c_3 r_3 - c_2 r_2 - c_1 r_1$

↳ FROM VECTOR LOOP

$$\left[\begin{array}{ccc|ccc} \frac{\partial x}{\partial \theta_2} & = & -r_2 c_2 & \frac{\partial y}{\partial \theta_2} & = & s_2 r_2 \\ \frac{\partial x}{\partial \theta_3} & = & c_3 r_3 & \frac{\partial y}{\partial \theta_3} & = & -s_3 r_3 \end{array}\right]$$

$$J(x,y) = \begin{bmatrix} -r_2 c_2 & c_3 r_3 \\ s_2 r_2 & -s_3 r_3 \end{bmatrix}$$

$s_1 r_1 = s_3 r_3 - s_2 r_2$

$c_1 r_1 = c_3 r_3 - c_2 r_2$

$$r_1 = \frac{s_3 r_3 - s_2 r_2}{s_1}$$

$$r_1 = \frac{c_3 r_3 - c_2 r_2}{c_1}$$

$$\frac{s_1}{c_1} = \frac{s_3 r_3 - s_2 r_2}{c_3 r_3 - c_2 r_2}$$

$$\tan\theta = \frac{s_3 r_3 - s_2 r_2}{c_3 r_3 - c_2 r_2}$$

$$\theta = \tan^{-1}\left(\frac{s_3 r_3 - s_2 r_2}{c_3 r_3 - c_2 r_2}\right)$$

Attachment F: Inverse Kinematics Code

Inverse Kinematics for In Contact

```
% Knowns
th_1 = 30.9481; % Theta One
s1 = sind(th_1); % Sin theta 1
c1 = cosd(th_1); % Cos theta 1
r1 = 1.0660; % Hip to foot length
r2 = 1; % Shank length
r3 = 2; % Thigh length
th_2c = 60;
th_3c = 45;
% Initial Guesses

th_2 = 30; % Theta 2
th_3 = 15; % Theta 3


% Our reasonable tolerance for position
epsilon = 5e-5;

% Setting a base epsilon to insure that code is read
epsilon1 = 1;
epsilon2 = 1;

th2_pl = [];
th3_pl = [];
```

Looping to converge on answer

```
while (epsilon1>= epsilon || epsilon2 >= epsilon)

    % Defining Sin and Cos for our angles
    s2 = sind(th_2);
    s3 = sind(th_3);
    c2 = cosd(th_2);
    c3 = cosd(th_3);

    %Equations of Motion based off our angles
    x = s1*r1 + s2*r2 - s3*r3;
    y = c1*r1 + c2*r2 - c3*r3;

    %Partial derivatives
    dx_dth2 =c2*r2;
    dx_dth3 =-c3*r3;
    dy_dth2 =-s2*r2;
    dy_dth3 =s3*r3;

    %Defining Jacobian
    Jac = [dx_dth2 , dx_dth3 ; dy_dth2 , dy_dth3];

    %Setting old thetas and storing as "n"
    th_2n = th_2;
    th_3n = th_3;
    %Reguessing
    xn_yn = [th_2;th_3] - ((Jac)\[x;y]) ;
    th_2 = xn_yn(1);
    th_3 = xn_yn(2);

    %Subtracting to refine our guesses
    epsilon1 = abs(th_2 - th_2n);
    epsilon2 = abs(th_3 - th_3n);

    th2_pl = [th2_pl , th_2];
    th3_pl = [th3_pl , th_3];

end
```

Plotting the Values

```
figure (1);
hold on;
plot (1:length(th2_pl), th2_pl);
plot (1:length(th3_pl), th3_pl);
yline (th_2c);
yline (th_3c);
xlim ([0 , length(th2_pl)])
hold off;
xlabel('Number of Iterations');
ylabel('Angle [deg]');
legend ('Theta2' , 'Theta3');
title ('Visual of How Solver Converges on a Point');
```

Attachment G : Kinetics Code

This section of code establishes all the variables and constants for the system

```
 % global allows for the same variables to be used throughout all function
 % files containing global "variable name"`

 global g % Gravity
 g = 9.81;

 % Shank info
 global r2 % Shank Length
 r2 = 20; % Selected by designer

 % Changes made here only affect kinetic simulation
 global mD % Shank mass
 mD = 1; % Selected by designer
 global ID % Shank moment of inertia
 ID = 1; % Selected by designer
 global MA % Moment applied to the knee at the top of the shank
 MA = -60; % Selected by designer

 % Changes made here only affect kinematic simulation
 global th2 % Angle of shank measured from right horizontal in the counter
clockwise direction
 th2 = deg2rad(60); % Selected by designer
 global th_dot2 % Angular velocity of shank in the counter clockwise direction
 th_dot2 = 5*((2*pi)/60); % Selected by designer
 global th_ddot2 % Angular acceleration of shank in the counter clockwise
direction
 th_ddot2 = -10*((2*pi)/3600); % Selected by designer

 % Thigh info
 global r3 % Thigh Length
 r3 = 20; % Selected by designer

 % Changes made here only affect kinetic simulation
 global mO
 mO = 2;
 global IO % Thigh moment of inertia
 IO = 1; % Selected by designer
 global MO % Moment applied to the hip at the top of the thigh
 MO = -40; % Selected by designer
 global X_ddotO % Initial acceleration of the robot body in the X-direction
 X_ddotO = -5; % Selected by designer
 global Y_ddot0 % Initial acceleration of the robot body in the Y-direction
 Y_ddot0 = -9.8; % Selected by designer

 % Changes made here only affect kinetic simulation
 global th3 % Thigh angle measeured from right horizontal in the counter
clockwise direction
 th3 = deg2rad(-45); % Selected by designer
 global th_dot3 % Thigh angular velocity in the counter clockwise direction
 th_dot3 = -10*((2*pi)/60); % Selected by designer
 global th_ddot3 % Thigh angular acceleration in the counter clockwise direction
 th_ddot3 = 10*((2*pi)/3600); % Selected by designer
```

This section is for determining the initial position conditions for the mechanism based on the choices for r2, r3, th2, and th3. fminsearch minimizes min1.mlx so that the vector loop is complete with three sections (r1, r2, and r3). It gives the length of r1 and the angle of th1.

```
 % guess whatever to give the model somewhere to start
 IC_guess = [24
             160];
 IC_actual1 = fminsearch(@min1, IC_guess); % Gives the real Initial position
conditions based on minimazing min1
```

Airborne Kinetics

```
 % Changes the Initial conditions used for the kinematics to initial
 % conditions usable by the kinetics simulation
 IC_actual2 = IC_actual1;
 IC_actual2(3,1) = th2;
 IC_actual2(4,1) = th3;

 % Contains the initial conditions for the velocities of the system. They
 % are all set to zero assuming that the leg starts at rest.
 CHI_dot = zeros(4,1);

 % simulation data is stored in variable out
 out = sim("UnconstrainedNoContactKinetics.slx");

 % Positional info for simulation
 s2 = sin(out.x(:,3));
 c2 = cos(out.x(:,3));
 s3 = sin(out.x(:,4));
 c3 = cos(out.x(:,4));

 % Animation for verification
 hf7 = figure(7);
 hf7.Visible = 'on';
```

```
hf7.WindowState = 'maximized';

for i = 1:length(out.tout)
           % O A        B
    y_pos = [out.YO(i) out.YO(i)+s3(i)*r3 out.YO(i)+s3(i)*r3-s2(i)*r2];
    x_pos = [out.XO(i) out.XO(i)+c3(i)*r3 out.XO(i)+c3(i)*r3-c2(i)*r2];
    plot(x_pos, y_pos,'LineWidth', 2);
    hold on
    plot(out.XO(1:i),out.YO(1:i),'b','LineWidth', 2);
    plot(out.XO(1:i)+c3(1:i)*r3,out.YO(1:i)+s3(1:i)*r3,"r",'LineWidth', 2);
    plot(out.XO(1:i)+c3(1:i)*r3-c2(1:i)*r2,out.YO(1:i)+s3(1:i)*r3-
s2(1:i)*r2,"g",'LineWidth', 2);
    xlabel('Horizontal Position [mm]','FontSize', 30);
    ylabel('Vertical Position [mm]','FontSize', 30);
    hold off
    axis('equal');
    xlim([-50 50]);
    ylim([0 80]);
    drawnow;
    frame = getframe(hf7); % this is for saving the gif
    im{i} = frame2im(frame); % this is for saving the gif
end
```

```
% Produces a gif of the animation
filename = 'Mech2.gif';
for i = 1:length(out.tout)
    [A, map] = rgb2ind(im{i},256);
    if i == 1
        imwrite(A, map, filename, 'gif', 'LoopCount', Inf, 'DelayTime', 0.1);
    else
        imwrite(A, map, filename, 'gif', 'WriteMode','append',
'DelayTime',0.1);
    end
end
```

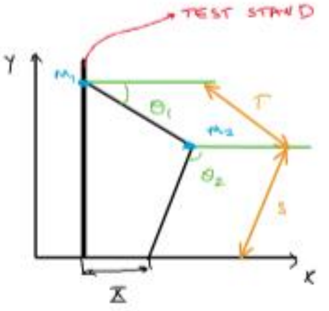
Grounded Kinetics

```
IC_actual2 = IC_actual1;
IC_actual2(3,1) = th2;
IC_actual2(4,1) = th3;

CHI_dot = zeros(4,1);

out = sim("ConstrainedContactKinetics.slx");
```

```
s1 = sin(out.x(:,2));
c1 = cos(out.x(:,2));
s2 = sin(out.x(:,3));
c2 = cos(out.x(:,3));
s3 = sin(out.x(:,4));
c3 = cos(out.x(:,4));

%figure;
%dbg = c1.*out.x(:,1);
%plot(out.tout, dbg);

% Animation for verification
hf8 = figure(8);
hf8.Visible = 'on';
hf8.WindowState = 'maximized';

for i = 1:length(out.tout)
           % B A        O
    y_pos = [0 s2(i)*r2 s2(i)*r2-s3(i)*r3];
    x_pos = [0 c2(i)*r2 c2(i)*r2-c3(i)*r3];
    plot(x_pos, y_pos,'LineWidth', 2);
    hold on
    plot(c2(1:i)*r2,s2(1:i)*r2,"r",'LineWidth', 2);
    plot(c2(1:i)*r2-c3(1:i)*r3,s2(1:i)*r2-s3(1:i)*r3,"g",'LineWidth', 2);
    xlabel('Horizontal Position [mm]','FontSize', 30);
    ylabel('Vertical Position [mm]','FontSize', 30);
    hold off
    axis('equal');
    xlim([-100 200]);
    ylim([-100 100]);
    drawnow;
    frame = getframe(hf8); % this is for saving the gif
    im{i} = frame2im(frame); % this is for saving the gif
end
```

Attachment H: Lagrangian

SCHEMATIC:



$$x_1 = \emptyset \qquad x_2 = T\cos\theta_1$$

$$y_1 = T\sin\theta_1 + S\sin\theta_2 \qquad y_2 = S\sin\theta_2$$

$$\dot{x}_1 = \emptyset \qquad \dot{x}_2 = -T\sin\theta_1\dot{\theta}_1$$

$$\dot{y}_1 = T\cos\theta_1\dot{\theta}_1 + S\cos\theta_2\dot{\theta}_2 \qquad \dot{y}_2 = S\cos\theta_2\dot{\theta}_2$$

WITH: m_1 = THIGH MOTOR
m_2 = SHANK MOTOR

T = TOTAL LENGTH OF THIGH
S = TOTAL LENGTH OF SHANK

TEST STAND IS SHOWN IN RED & RESTRICTS MOTION OF m_1 TO ONLY ALLOW MOTION IN Y-DIRECTION

$$KE = KE_1 + KE_2$$

$$KE_1 = \tfrac{1}{2}m_1\dot{x}_1^2 + \tfrac{1}{2}m_1\dot{y}_1^2$$

$$KE_1 = \tfrac{1}{2}m_1\left(T^2\cos^2\theta_1\dot{\theta}_1^2 + TS\cos\theta_1\cos\theta_2\dot{\theta}_1\dot{\theta}_2 + S^2\cos^2\theta_2\dot{\theta}_2^2\right)$$

$$KE_2 = \tfrac{1}{2}m_2\left(T^2\sin^2\theta_1\dot{\theta}_1^2 + S^2\cos^2\theta_2\dot{\theta}_2^2\right)$$

$$KE = \tfrac{1}{2}m_1\left(T^2\cos^2\theta_1\dot{\theta}_1^2 + 2TS\cos\theta_1\cos\theta_2\dot{\theta}_1\dot{\theta}_2 + S^2\cos^2\theta_2\dot{\theta}_2^2\right) + \tfrac{1}{2}m_2\left(T^2\sin^2\theta_1\dot{\theta}_1^2 + S^2\cos^2\theta_2\dot{\theta}_2^2\right)$$

$$KE = m_1TS\cos\theta_1\cos\theta_2\dot{\theta}_1\dot{\theta}_2 + \tfrac{1}{2}m_1T^2\cos^2\theta_1\dot{\theta}_1^2 + \tfrac{1}{2}m_2T^2\sin^2\theta_1\dot{\theta}_1^2 + \tfrac{1}{2}(m_1+m_2)\left(S^2\cos^2\theta_2\dot{\theta}_2^2\right)$$

$$\boxed{KE = m_1TS\cos\theta_1\cos\theta_2\dot{\theta}_1\dot{\theta}_2 + \tfrac{1}{2}T^2\dot{\theta}_1^2\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right) + \tfrac{1}{2}(m_1+m_2)\left(S^2\cos^2\theta_2\dot{\theta}_2^2\right)}$$

$$PE = gm_1y_1 + gm_2y_2$$

$$PE = gm_1(T\sin\theta_1 + S\sin\theta_2) + gm_2(S\sin\theta_2)$$

$$\boxed{PE = gm_1T\sin\theta_1 + gS\sin\theta_2(m_1+m_2)}$$

$$L = T - V$$

$$L = KE - PE$$

$$L = m_1TS\cos\theta_1\cos\theta_2\dot{\theta}_1\dot{\theta}_2 + \tfrac{1}{2}T^2\dot{\theta}_1^2\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right) + \tfrac{1}{2}(m_1+m_2)\left(S^2\cos^2\theta_2\dot{\theta}_2^2\right) - gm_1T\sin\theta_1 - gS\sin\theta_2(m_1+m_2)$$

→ SOLVE TO FIND EQN OF MOTION FOR θ_1 & θ_2

→ GENERAL FORM

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot q}\right) = \boxed{\frac{\partial f}{\partial \dot q}} + \boxed{\lambda \frac{d\beta}{dq} \frac{d\beta}{dq}}$$

f = RAYLEIGH'S DISSIPATION

ACCOUNTS FOR NON-CONSERVATIVE FORCES] TORQUE GOES HERE

LAGRANGIAN MULTIPLIER

ACCOUNTS FOR SYSTEM CONSTRAINTS] FOOT CONSTRAINT GOES HERE

* $\dot q$ = TIME DERIVATIVE OF q

q = GENERALIZED COORD. USED IN SYSTEM (ie: θ_1 & θ_2)

** IF NO NON-CONSERVATIVE FORCES OR CONSTRAINTS, SET β & f EQUAL TO ZERO

→ FIRST, CONSIDER SYSTEM WITH TORQUE BUT NO CONSTRAINT

θ_1 :

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = \tau_1$$

$$\frac{\partial L}{\partial \theta_1} = -m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_1 \dot\theta_2 + T^2 \dot\theta_1^2 \left(-m_1 \cos\theta_1 \sin\theta_1 + m_2 \sin\theta_1 \cos\theta_1\right) + g m_1 T \cos\theta_1$$

$$\frac{\partial L}{\partial \theta_1} = -m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_1 \dot\theta_2 + T^2 \dot\theta_1^2 \cos\theta_1 \sin\theta_1 (m_2 - m_1) + g m_1 T \cos\theta_1$$

$$\frac{\partial L}{\partial \dot\theta_1} = m_1 TS \cos\theta_1 \cos\theta_2 \dot\theta_2 + T^2 \dot\theta_1 (m_1 \cos^2\theta_1 + m_2 \sin^2\theta_1)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = -m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_2 \dot\theta_1 - m_1 TS \cos\theta_1 \sin\theta_2 \dot\theta_2^2 + m_1 TS \cos\theta_1 \cos\theta_2 \ddot\theta_2 + T^2 \dot\theta_1 \left(-2m_1 \cos\theta_1 \sin\theta_1 \dot\theta_1 + 2m_2 \sin\theta_1 \cos\theta_1 \dot\theta_1\right) + T^2 \ddot\theta_1 \left(m_1 \cos^2\theta_1 + m_2 \sin^2\theta_1\right)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = -m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_2 \dot\theta_1 - m_1 TS \cos\theta_1 \sin\theta_2 \dot\theta_2^2 + m_1 TS \cos\theta_1 \cos\theta_2 \ddot\theta_2 + 2T^2 \dot\theta_1^2 \cos\theta_1 \sin\theta_1 (m_2 - m_1) + T^2 \ddot\theta_1 \left(m_1 \cos^2\theta_1 + m_2 \sin^2\theta_1\right)$$

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = -m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_1 \dot\theta_2 + T^2 \dot\theta_1^2 \cos\theta_1 \sin\theta_1 (m_2 - m_1) + g m_1 T \cos\theta_1$$

$$- \left[-m_1 TS \sin\theta_1 \cos\theta_2 \dot\theta_2 \dot\theta_1 - m_1 TS \cos\theta_1 \sin\theta_2 \dot\theta_2^2 + m_1 TS \cos\theta_1 \cos\theta_2 \ddot\theta_2 + 2T^2 \dot\theta_1^2 \cos\theta_1 \sin\theta_1 (m_2 - m_1) + T^2 \ddot\theta_1 \left(m_1 \cos^2\theta_1 + m_2 \sin^2\theta_1\right) \right]$$

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = -\cancel{m_1 TS\sin\theta_1\cos\theta_2\dot\theta_1\dot\theta_2} + \cancel{T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1)} + gm_1T\cos\theta_1 + \cancel{m_1TS\sin\theta_1\cos\theta_2\dot\theta_2\dot\theta_1} + m_1TS\cos\theta_1\sin\theta_2\dot\theta_2^2$$

$$- m_1TS\cos\theta_1\cos\theta_2\ddot\theta_2 - \overset{-T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1)}{\cancel{2T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1)}} - T^2\ddot\theta_1\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right)$$

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = + gm_1T\cos\theta_1 + m_1TS\cos\theta_1\sin\theta_2\dot\theta_2^2 - m_1TS\cos\theta_1\cos\theta_2\ddot\theta_2 - T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1) - T^2\ddot\theta_1\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right)$$

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_1}\right) = \tau_1$$

$$gm_1T\cos\theta_1 + m_1TS\cos\theta_1\sin\theta_2\dot\theta_2^2 - m_1TS\cos\theta_1\cos\theta_2\ddot\theta_2 - T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1) - T^2\ddot\theta_1\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right) = \tau_1$$

$$\boxed{- m_1TS\cos\theta_1\cos\theta_2\ddot\theta_2 - T^2\ddot\theta_1\left(m_1\cos^2\theta_1 + m_2\sin^2\theta_1\right) = \tau_1 - gm_1T\cos\theta_1 - m_1TS\cos\theta_1\sin\theta_2\dot\theta_2^2 + T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1)}$$

θ_2:

$$\frac{\partial L}{\partial \theta_2} - \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot\theta_2}\right) = \tau_2$$

$$\frac{\partial L}{\partial \theta_2} = -m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 - (m_1+m_2)\left(s^2\cos\theta_2\sin\theta_2\dot\theta_2^2\right) - gS\cos\theta_2(m_1+m_2)$$

$$\frac{\partial L}{\partial \dot\theta_2} = m_1TS\cos\theta_1\cos\theta_2\dot\theta_1 + (m_1+m_2)\left(s^2\cos^2\theta_2\dot\theta_2\right)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot\theta_2} = -m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2 - m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 + m_1TS\cos\theta_1\cos\theta_2\ddot\theta_1 + (m_1+m_2)\left(-2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2 + s^2\cos^2\theta_2\ddot\theta_2\right)$$

$$\frac{\partial L}{\partial \theta_2} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_2}\right) = -m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 - (m_1+m_2)\left(s^2\cos\theta_2\sin\theta_2\dot\theta_2^2\right) - gS\cos\theta_2(m_1+m_2)$$

$$-\left[-m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2 - m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 + m_1TS\cos\theta_1\cos\theta_2\ddot\theta_1 + (m_1+m_2)\left(-2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2 + s^2\cos^2\theta_2\ddot\theta_2\right)\right]$$

$$\frac{\partial L}{\partial \theta_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_2}\right) = -m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 - (m_1+m_2)\left(s^2\cos\theta_2\sin\theta_2\dot\theta_2^2\right) - gS\cos\theta_2(m_1+m_2) + m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2$$

$$+ m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 - m_1TS\cos\theta_1\cos\theta_2\ddot\theta_1 - (m_1+m_2)\left(-2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2 + s^2\cos^2\theta_2\ddot\theta_2\right)$$

$$\frac{\partial L}{\partial \theta_2} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta_2}\right) = \tau_2$$

$$-m_1TS\cos\theta_1\cancel{\sin}\theta_2\dot\theta_1\dot\theta_2 - (m_1+m_2)(S^2\cos\theta_2\sin\theta_2\dot\theta_2^2) - gS\cos\theta_2(m_1+m_2) + m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2 + m_1TS\cancel{\cos\theta_1}\sin\theta_2\dot\theta_1\dot\theta_2 - m_1TS\cos\theta_1\cos\theta_2\ddot\theta_1 - (m_1+m_2)\left(-2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2 + S^2\cos^2\theta_2\ddot\theta_2\right) = \tau_2$$

$$\boxed{-m_1TS\cos\theta_1\cos\theta_2\ddot\theta_1 - (m_1+m_2)S^2\cos^2\theta_2\ddot\theta_2 = \tau_2 + (m_1+m_2)(S^2\cos\theta_2\sin\theta_2\dot\theta_2^2) + gS\cos\theta_2(m_1+m_2) - m_1TS\cos\theta_1\sin\theta_2\dot\theta_1\dot\theta_2 - (m_1+m_2)(2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2)}$$

$$\begin{bmatrix} -T^2(m_1\cos^2\theta_1 + m_2\sin^2\theta_1) & -m_1TS\cos\theta_1\cos\theta_2 \\ -m_1TS\cos\theta_1\cos\theta_2 & -(m_1+m_2)S^2\cos^2\theta_2 \end{bmatrix}\begin{bmatrix}\ddot\theta_1 \\ \ddot\theta_2\end{bmatrix} = \begin{bmatrix} \tau_1 - gm_1T\cos\theta_1 - m_1TS\sin\theta_1\cos\theta_2\dot\theta_2\dot\theta_1 + T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1) \\ \tau_2 + (m_1+m_2)(S^2\cos\theta_2\sin\theta_2\dot\theta_2^2) + gS\cos\theta_2(m_1+m_2) - m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2 - (m_1+m_2)(2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2) \end{bmatrix}$$

$$\boxed{\begin{bmatrix}\ddot\theta_1 \\ \ddot\theta_2\end{bmatrix} = \begin{bmatrix} -T^2(m_1\cos^2\theta_1 + m_2\sin^2\theta_1) & -m_1TS\cos\theta_1\cos\theta_2 \\ -m_1TS\cos\theta_1\cos\theta_2 & -(m_1+m_2)S^2\cos^2\theta_2 \end{bmatrix}^{-1}\begin{bmatrix} \tau_1 - gm_1T\cos\theta_1 - m_1TS\cos\theta_1\sin\theta_2\dot\theta_2^2 + T^2\dot\theta_1^2\cos\theta_1\sin\theta_1(m_2-m_1) \\ \tau_2 + (m_1+m_2)(S^2\cos\theta_2\sin\theta_2\dot\theta_2^2) + gS\cos\theta_2(m_1+m_2) - m_1TS\sin\theta_1\cos\theta_2\dot\theta_1^2 - (m_1+m_2)(2S^2\sin\theta_2\cos\theta_2\dot\theta_2^2) \end{bmatrix}}$$

Attachment I : Design Hazard Checklist

Y	N	
×		1. Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and sheer points?
	×	2. Can any part of the design undergo high accelerations/decelerations?
	×	3. Will the system have any large moving masses or large forces?
	×	4. Will the system produce a projectile?
	×	5. Would it be possible for the system to fall under gravity creating injury?
	×	6. Will a user be exposed to overhanging weights as part of the design?
	×	7. Will the system have any sharp edges?
	×	8. Will any part of the electrical systems not be grounded?
	×	9. Will there be any large batteries or electrical voltage in the system above 40 V?
×		10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
	×	11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?
	×	12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
	×	13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
	×	14. Can the system generate high levels of noise?
	×	15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures, etc?
	×	16. Is it possible for the system to be used in an unsafe manner?
	×	17. Will there be any other potential hazards not listed above? If yes, please explain on reverse.

Description of Hazard	**Planned Corrective Action**	**Planned Date**	**Actual Date**
As part of the testing process for this design, pinch points will be introduced when running the motor and the leg	This hazard is unavoidable when dealing with rotating machinery. To avoid injury, we will keep hands away from moving parts.	Spring Quarter	
The movement of our leg is dependent on the battery of the motor. While controlling the leg, it is possible that our code could overload motors.	To prevent issues with over current we will include fuses, and for over voltage we will add voltage regulators.	Winter Quarter	

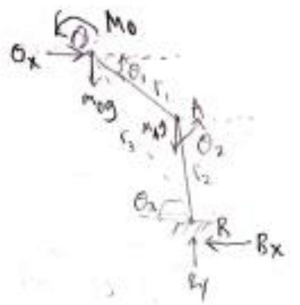
Attachment J : New Kinetics Hand Calcs

1

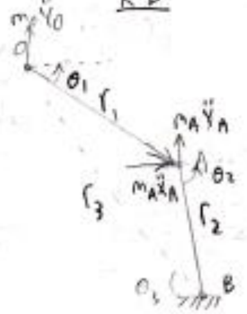
MECHANISM TYPE: CONTACT, CONSTRAINED

KINETICS

FBD



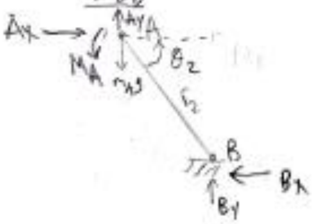
KD



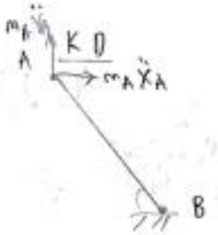
$$\Sigma M_B = M_0 + m_0g\,c_3r_3 - O_x s_3 r_3 + m_Ag\,c_2r_2$$

$$\Sigma M_B = -m_0\ddot{Y}_0 c_3r_3 - m_A\ddot{X}_A(s_2r_2) - m_A\ddot{Y}_A(c_2r_2)$$

FBD



KD



$$\Sigma M_B = M_A + m_Ag(c_2r_2) - A_x(s_2r_2) - A_y(c_2r_2)$$

$$\Sigma M_B = -m_A\ddot{X}_A(s_2r_2) - m_A\ddot{Y}_A(c_2r_2)$$

√EQ 1 $\overset{I}{M_0} + \check{m}_0\check{g}\check{c}_3\check{r}_3 - O_x\check{s}_3\check{r}_3 + \check{m}_A\check{g}\check{c}_2\check{r}_2 = -m_0\ddot{Y}_0c_3r_3 - m_A\ddot{X}_A(s_2r_2) - m_A\ddot{Y}_Ac_2r_2$

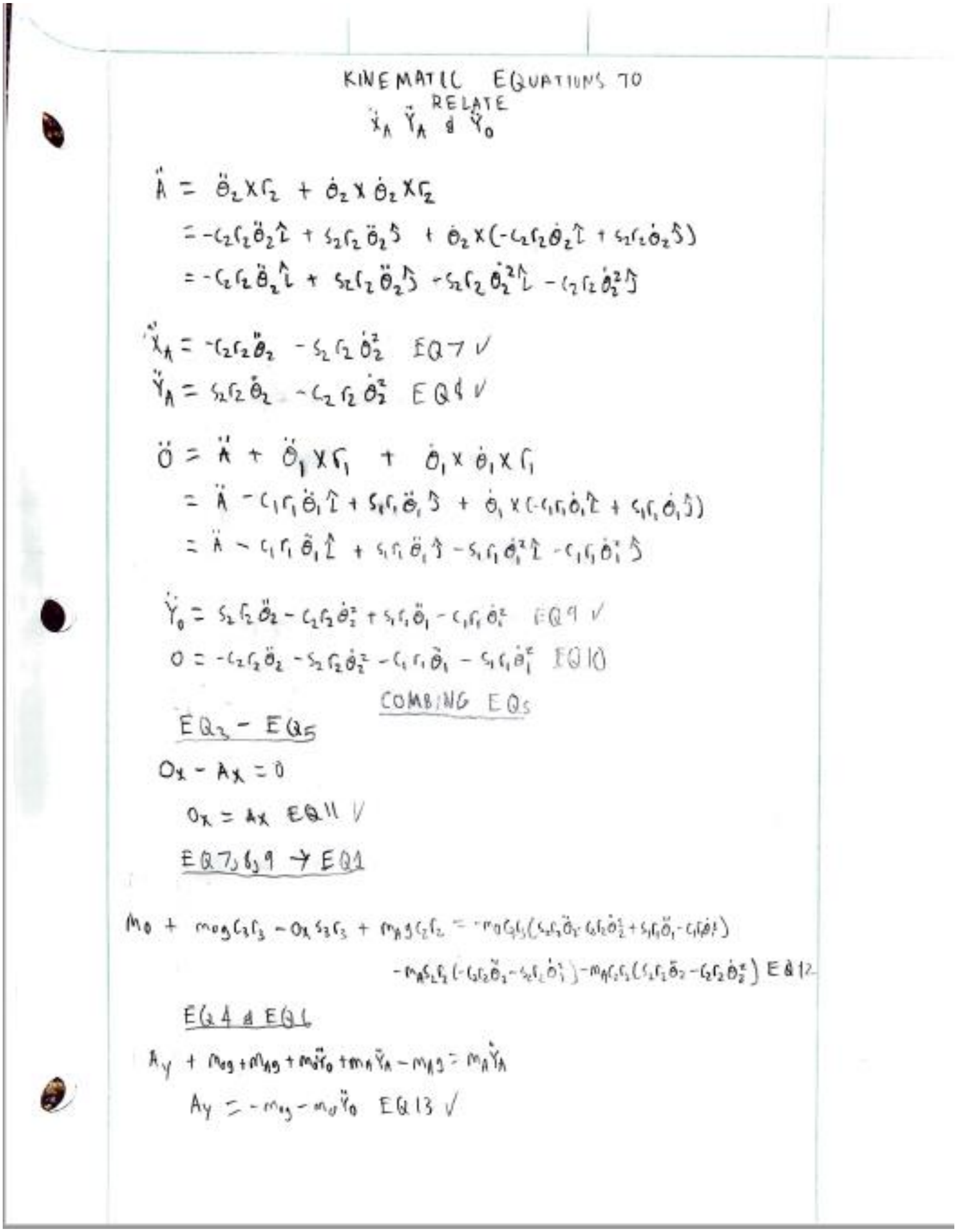
√EQ 2 $\overset{I}{M_A} + \check{m}_A\check{g}\check{c}_2\check{r}_2 - A_x\check{s}_2\check{r}_2 - A_y\check{c}_2\check{r}_2 = -\check{m}_A\ddot{X}_A\check{s}_2\check{r}_2 - \check{m}_A\ddot{Y}_A\check{c}_2\check{r}_2$

√EQ 3 $O_x - B_x = m_A\ddot{X}_A$

√EQ 4 $B_y - m_0g - m_Ag = m_0\ddot{Y}_0 + m_A\ddot{Y}_A$

√EQ 5 $A_x - B_x = m_A\ddot{X}_A$

√EQ 6 $A_y + B_y - m_Ag = m_A\ddot{Y}_A$



KINEMATIC EQUATIONS TO RELATE

$\ddot{x}_A \; \ddot{Y}_A \; \& \; \ddot{Y}_0$

$$\ddot{A} = \ddot{\theta}_2 \times r_2 + \dot{\theta}_2 \times \dot{\theta}_2 \times r_2$$

$$= -c_2 r_2 \ddot{\theta}_2 \hat{i} + s_2 r_2 \ddot{\theta}_2 \hat{j} + \dot{\theta}_2 \times (-c_2 r_2 \dot{\theta}_2 \hat{i} + s_2 r_2 \dot{\theta}_2 \hat{j})$$

$$= -c_2 r_2 \ddot{\theta}_2 \hat{i} + s_2 r_2 \ddot{\theta}_2 \hat{j} - s_2 r_2 \dot{\theta}_2^2 \hat{i} - c_2 r_2 \dot{\theta}_2^2 \hat{j}$$

$\ddot{X}_A = -c_2 r_2 \ddot{\theta}_2 - s_2 r_2 \dot{\theta}_2^2$ EQ 7 ✓

$\ddot{Y}_A = s_2 r_2 \dot{\theta}_2 - c_2 r_2 \dot{\theta}_2^2$ EQ 8 ✓

$$\ddot{O} = \ddot{A} + \ddot{\theta}_1 \times r_1 + \dot{\theta}_1 \times \dot{\theta}_1 \times r_1$$

$$= \ddot{A} - c_1 r_1 \ddot{\theta}_1 \hat{i} + s_1 r_1 \ddot{\theta}_1 \hat{j} + \dot{\theta}_1 \times (-c_1 r_1 \dot{\theta}_1 \hat{i} + s_1 r_1 \dot{\theta}_1 \hat{j})$$

$$= \ddot{A} - c_1 r_1 \ddot{\theta}_1 \hat{i} + s_1 r_1 \ddot{\theta}_1 \hat{j} - s_1 r_1 \dot{\theta}_1^2 \hat{i} - c_1 r_1 \dot{\theta}_1^2 \hat{j}$$

$\ddot{Y}_0 = s_2 r_2 \ddot{\theta}_2 - c_2 r_2 \dot{\theta}_2^2 + s_1 r_1 \ddot{\theta}_1 - c_1 r_1 \dot{\theta}_1^2$ EQ 9 ✓

$0 = -c_2 r_2 \ddot{\theta}_2 - s_2 r_2 \dot{\theta}_2^2 - c_1 r_1 \ddot{\theta}_1 - s_1 r_1 \dot{\theta}_1^2$ EQ 10

COMBING EQs

EQ 3 - EQ 5

$O_x - A_x = 0$

$O_x = A_x$ EQ 11 ✓

EQ 7, 8, 9 → EQ 1

$$M_0 + m_0 g c_3 r_3 - O_x s_3 r_3 + m_A g c_2 r_2 = -m_0 c_3 r_3 (s_2 r_2 \ddot{\theta}_2 - c_2 r_2 \dot{\theta}_2^2 + s_1 r_1 \ddot{\theta}_1 - c_1 r_1 \dot{\theta}_1^2)$$

$$- m_A s_2 r_2 (-c_2 r_2 \ddot{\theta}_2 - s_2 r_2 \dot{\theta}_2^2) - m_A c_2 r_2 (s_2 r_2 \ddot{\theta}_2 - c_2 r_2 \dot{\theta}_2^2)$$ EQ 12

EQ 4 & EQ 6

$A_y + m_0 g + m_A g + m_0 \ddot{r}_0 + m_A \ddot{Y}_A - m_A g = m_A \ddot{Y}_A$

$A_y = -m_0 g - m_0 \ddot{Y}_0$ EQ 13 ✓

EQ 11 & 13 → EQ 2

$M_A + m_A g c_2 r_2 - O_x s_2 r_2 + (m_O g + m_O \ddot{Y}_O) c_2 r_2 = -m_A \ddot{x}_A s_2 r_2 - m_A \ddot{y}_A c_2 r_2$ EQ 14

EQ 7,6,9 → EQ 14

$M_A + m_A g c_2 r_2 - O_x s_2 r_2 + m_O g c_2 r_2 + m_O c_2 r_2 (s_2 r_2 \ddot\theta_2 - c_2 r_2 \dot\theta_2^2 + s_1 r_1 \ddot\theta_1 - c_1 r_1 \dot\theta_1^2) =$

$-m_A s_2 r_2 (-c_2 r_2 \ddot\theta_2 - s_2 r_2 \dot\theta_2^2) - m_A c_2 r_2 (s_2 r_2 \ddot\theta_2 - c_2 r_2 \dot\theta_2^2)$ EQ 15

EQ 10 & 12 & 15

EQ 10':

$c_2 r_2 \ddot\theta_2 + c_1 r_1 \ddot\theta_1 = +s_2 r_2 \dot\theta_2^2 - s_1 r_1 \dot\theta_1^2$

EQ 12'

$\ddot\theta_2 (m_O c_3 r_3 s_2 r_2 - m_A s_2 r_2 c_2 r_2 + m_A c_2 r_2 s_2 r_2) + \ddot\theta_1 (m_O c_3 r_3 s_1 r_1) - O_x s_3 r_3 = m_O c_3 r_3 (c_2 r_2 \dot\theta_2^2 + c_1 r_1 \dot\theta_1^2)$

$+ m_A \dot\theta_2^2 r_2^2 (s_2^2 + c_2^2)$

$\ddot\theta_2 (m_O c_3 r_3 s_2 r_2) + \ddot\theta_1 (m_O c_3 r_3 s_1 r_1) - O_x s_3 r_3 = m_O c_3 r_3 (c_2 r_2 \dot\theta_2^2 + c_1 r_1 \dot\theta_1^2) + m_A \dot\theta_2^2 r^2$

EQ 15'.

$\ddot\theta_2 (m_O c_2 r_2 s_2 r_2 + m_O c_2 r_2 s_1 r_1 - m_A s_2 r_2 c_2 r_2 + m_A s_2 r_2 c_2 r_2) + \ddot\theta_1 (m_O c_2 r_2 s_1 r_1) - O_x s_2 r_2 =$

$m_A s_2 r_2 s_2 r_2 \dot\theta_2^2 + m_A c_2 r_2 c_2 r_2 \dot\theta_2^2 + m_O c_2 r_2 c_2 r_2 \dot\theta_2^2 + m_O c_2 r_2 c_1 r_1 \dot\theta_1^2 - m_O g c_2$

$- M_A - m_A g c_2 r_2$

$\ddot\theta_2 (m_O c_2 s_2 r_2^2 + m_O c_2 r_2 s_1 r_1) + \ddot\theta_1 (m_O c_2 r_2 s_1 r_1) - O_x s_2 r_2 = m_A \dot\theta_2^2 r_2^2 + m_O c_2^2 r_2^2 \dot\theta_2^2 + m_O c_2 r_2 c_1 r_1 \dot\theta_1^2$

$- m_O g c_2 r_2 - M_A - m_A g c_2 r_2$

COMBINE EQ15 & EQ12':

$\ddot\theta_2 (m_O c_3 r_3 s_2 r_2) + \ddot\theta_1 (m_O c_3 r_3 s_1 r_1) - \{[\ddot\theta_2 (m_O c_2 s_2 r_2^2 + m_O c_2 r_2 s_1 r_1) + \ddot\theta_1 (m_O c_2 r_2 s_1 r_1) - m_A \dot\theta_2^2 r_2^2 - m_O c_2^2 r_2^2 \dot\theta_2^2$

$- m_O c_2 r_2 c_1 r_1 \dot\theta_1^2 + m_O g c_2 r_2 + M_A + m_A g c_2 r_2] / s_2 r_2\} s_3 r_3 =$

$m_O c_3 r_3 (c_2 r_2 \dot\theta_2^2 + c_1 r_1 \dot\theta_1^2) + m_A \dot\theta_2^2 r_2^2$

$\ddot\theta_2 (m_O r_3 c_3 s_2^2 r_2^2) + \ddot\theta_1 (m_O c_3 r_3 s_1 r_1 s_2 r_2) - \ddot\theta_2 (m_O c_2 s_2 s_3 r_3 r_2^2 + m_O c_2 r_2 s_1 r_1 s_3 r_3) - \ddot\theta_1 (m_O c_2 r_2 s_1 r_1 s_3 r_3)$

$+ M_A \dot\theta_2^2 r_2^2 s_3 r_3 + m_O c_2^2 r_2^2 \dot\theta_2^2 s_3 r_3 + m_O c_2 r_2 c_1 r_1 s_3 r_3 \dot\theta_1^2 - m_O g c_2 r_2 s_3 r_3 - M_A s_3 r_3 - m_A g c_2 r_2 s_3 r_3$

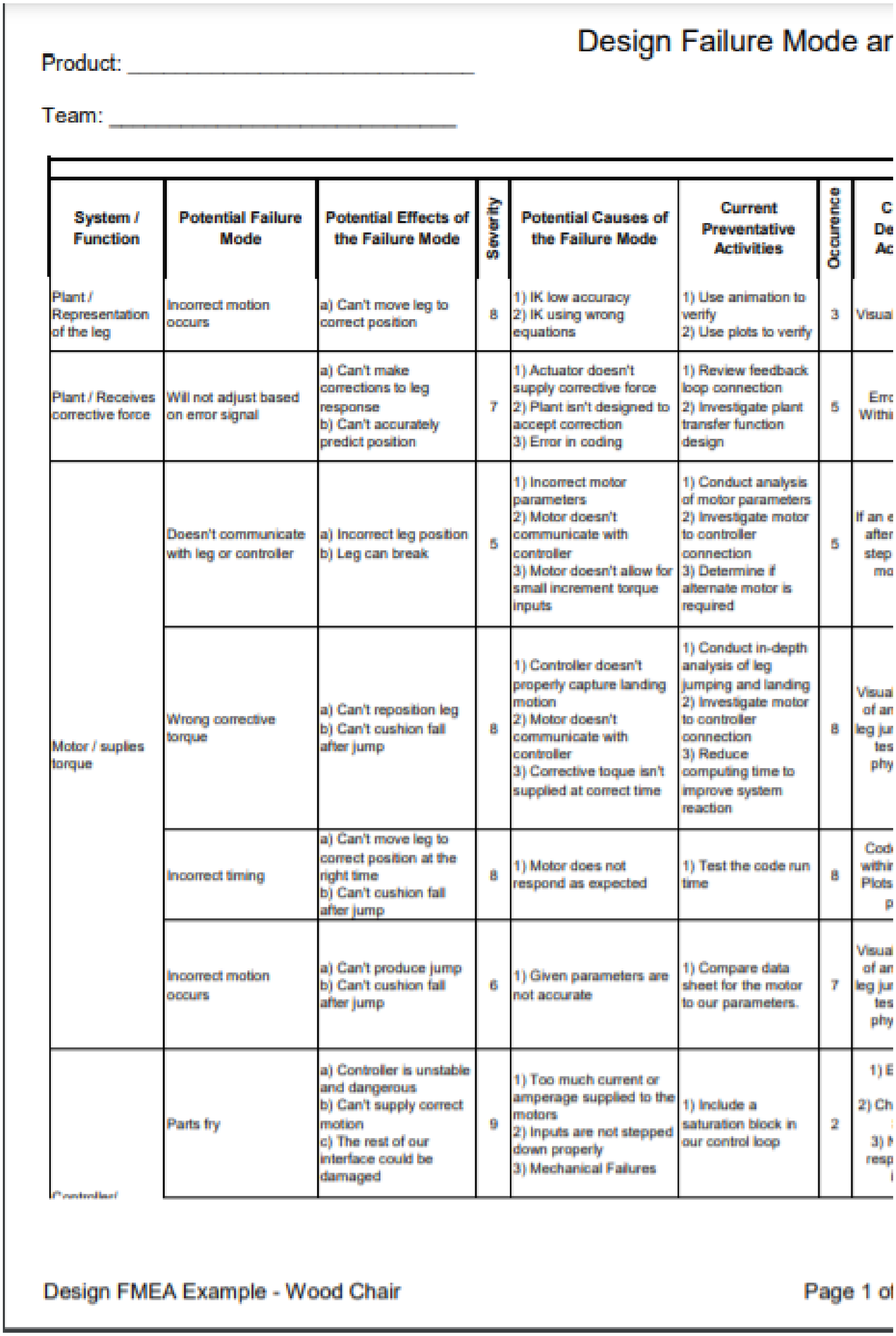
$= m_O c_3 r_3 s_2 r_2 \; (c_2 r_2 \dot\theta_2^2 + c_1 r_1 \dot\theta_1^2) + m_A \dot\theta_2^2 s_2 r_2^3$

$$\ddot{\theta}_2 (m_0 c_2 r_2 s_2^2 r_2^2 - m_0 c_2 s_2 s_5 r_3 r_2^2 - m_0 s_1 r_1 c_2 r_1 s_2 r_5) + \ddot{\theta}_1 (m_0 s_1 r_1 s_2 r_2 c_1 r_1 - m_0 s_1 r_1 c_2 r_2 s_2 r_5)$$

$$= m_0 c_2 r_2 s_2 r_2 (c_2 r_2 \dot{\theta}_2^2 + c_1 r_1 \dot{\theta}_1^2) + m_A \dot{\theta}_1^2 s_2 r_2^3 - m_A \dot{\theta}_2^2 r_2^2 s_5 r_3 - m_0 c_2^2 r_2^2 \dot{\theta}_2^2 s_2 r_5$$

$$- m_0 c_2 r_2 c_1 r_1 s_2 r_2 \dot{\theta}_1^2 + m_0 g c_2 r_2 s_2 r_5 + m_A s_2 r_5 + m_A g c_2 r_2 s_2 r_5$$

Attachment K-FMEA



Design Failure Mode ar

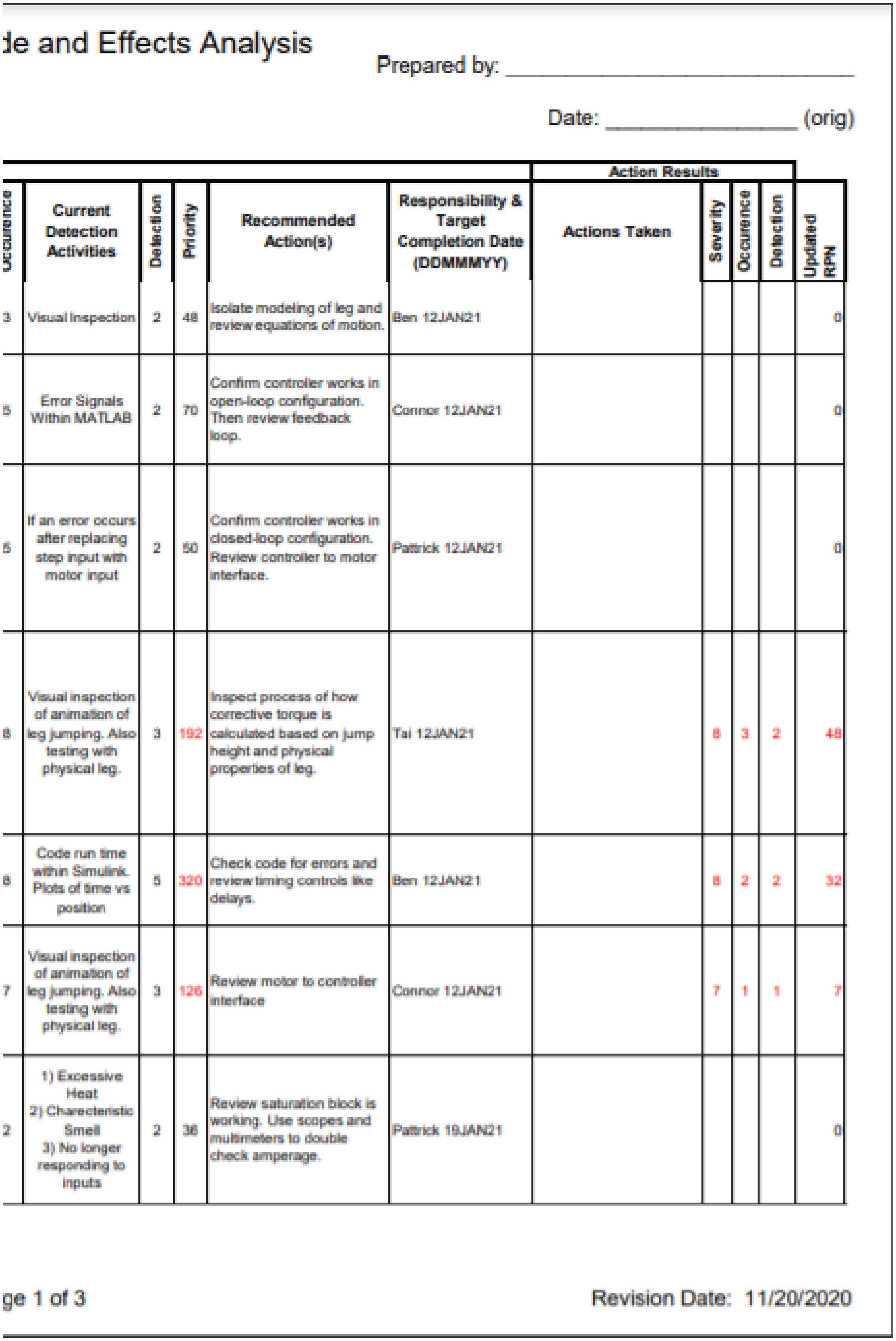
Product: ______________________________

Team: ______________________________

System / Function	Potential Failure Mode	Potential Effects of the Failure Mode	Severity	Potential Causes of the Failure Mode	Current Preventative Activities	Occurence	C De Ac
Plant / Representation of the leg	Incorrect motion occurs	a) Can't move leg to correct position	8	1) IK low accuracy 2) IK using wrong equations	1) Use animation to verify 2) Use plots to verify	3	Visual
Plant / Receives corrective force	Will not adjust based on error signal	a) Can't make corrections to leg response b) Can't accurately predict position	7	1) Actuator doesn't supply corrective force 2) Plant isn't designed to accept correction 3) Error in coding	1) Review feedback loop connection 2) Investigate plant transfer function design	5	Erro Withi
Motor / suplies torque	Doesn't communicate with leg or controller	a) Incorrect leg position b) Leg can break	5	1) Incorrect motor parameters 2) Motor doesn't communicate with controller 3) Motor doesn't allow for small increment torque inputs	1) Conduct analysis of motor parameters 2) Investigate motor to controller connection 3) Determine if alternate motor is required	5	If an e after step mo
	Wrong corrective torque	a) Can't reposition leg b) Can't cushion fall after jump	8	1) Controller doesn't properly capture landing motion 2) Motor doesn't communicate with controller 3) Corrective toque isn't supplied at correct time	1) Conduct in-depth analysis of leg jumping and landing 2) Investigate motor to controller connection 3) Reduce computing time to improve system reaction	8	Visual of an leg ju tes phy
	Incorrect timing	a) Can't move leg to correct position at the right time b) Can't cushion fall after jump	8	1) Motor does not respond as expected	1) Test the code run time	8	Cod withi Plots p
	Incorrect motion occurs	a) Can't produce jump b) Can't cushion fall after jump	6	1) Given parameters are not accurate	1) Compare data sheet for the motor to our parameters.	7	Visual of an leg ju tes phy
Controller/	Parts fry	a) Controller is unstable and dangerous b) Can't supply correct motion c) The rest of our interface could be damaged	9	1) Too much current or amperage supplied to the motors 2) Inputs are not stepped down properly 3) Mechanical Failures	1) Include a saturation block in our control loop	2	1) E 2) Ch 3) I resp

Design FMEA Example - Wood Chair

Page 1 o



de and Effects Analysis

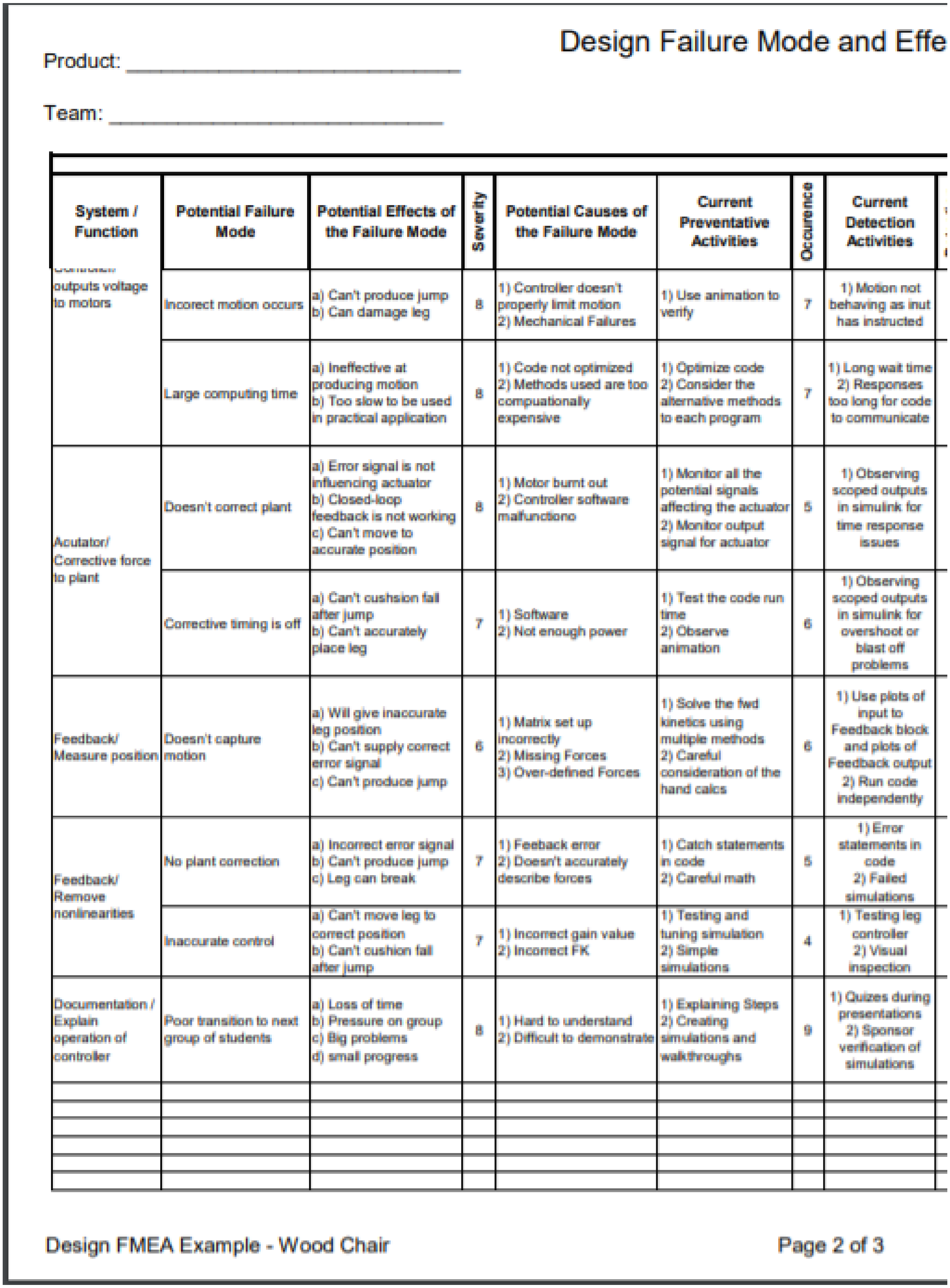
Prepared by: ______________________________

Date: ________________ (orig)

						Action Results				
ccurence	Current Detection Activities	Detection	Priority	Recommended Action(s)	Responsibility & Target Completion Date (DDMMMYY)	Actions Taken	Severity	Occurence	Detection	Updated RPN
3	Visual Inspection	2	48	Isolate modeling of leg and review equations of motion.	Ben 12JAN21					0
5	Error Signals Within MATLAB	2	70	Confirm controller works in open-loop configuration. Then review feedback loop.	Connor 12JAN21					0
5	If an error occurs after replacing step input with motor input	2	50	Confirm controller works in closed-loop configuration. Review controller to motor interface.	Pattrick 12JAN21					0
8	Visual inspection of animation of leg jumping. Also testing with physical leg.	3	192	Inspect process of how corrective torque is calculated based on jump height and physical properties of leg.	Tai 12JAN21		8	3	2	48
8	Code run time within Simulink. Plots of time vs position	5	320	Check code for errors and review timing controls like delays.	Ben 12JAN21		8	2	2	32
7	Visual inspection of animation of leg jumping. Also testing with physical leg.	3	126	Review motor to controller interface	Connor 12JAN21		7	1	1	7
2	1) Excessive Heat 2) Charecteristic Smell 3) No longer responding to inputs	2	36	Review saturation block is working. Use scopes and multimeters to double check amperage.	Pattrick 19JAN21					0

ge 1 of 3

Revision Date: 11/20/2020



Design Failure Mode and Effe

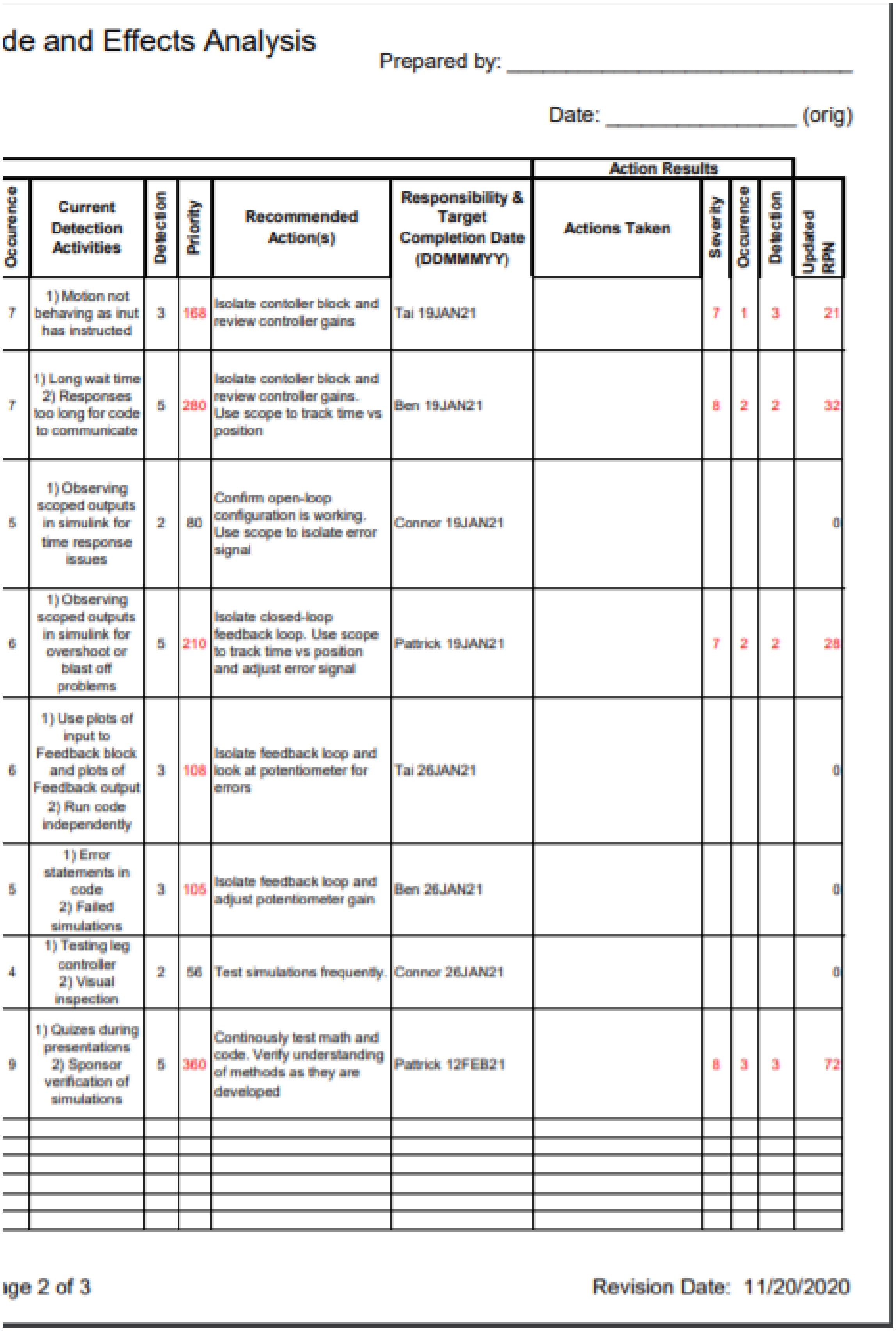
Product: ______________________________

Team: ______________________________

System / Function	Potential Failure Mode	Potential Effects of the Failure Mode	Severity	Potential Causes of the Failure Mode	Current Preventative Activities	Occurence	Current Detection Activities
Controller/ outputs voltage to motors	Incorect motion occurs	a) Can't produce jump b) Can damage leg	8	1) Controller doesn't properly limit motion 2) Mechanical Failures	1) Use animation to verify	7	1) Motion not behaving as inut has instructed
	Large computing time	a) Ineffective at producing motion b) Too slow to be used in practical application	8	1) Code not optimized 2) Methods used are too compuationally expensive	1) Optimize code 2) Consider the alternative methods to each program	7	1) Long wait time 2) Responses too long for code to communicate
Acutator/ Corrective force to plant	Doesn't correct plant	a) Error signal is not influencing actuator b) Closed-loop feedback is not working c) Can't move to accurate position	8	1) Motor burnt out 2) Controller software malfunctiono	1) Monitor all the potential signals affecting the actuator 2) Monitor output signal for actuator	5	1) Observing scoped outputs in simulink for time response issues
	Corrective timing is off	a) Can't cushsion fall after jump b) Can't accurately place leg	7	1) Software 2) Not enough power	1) Test the code run time 2) Observe animation	6	1) Observing scoped outputs in simulink for overshoot or blast off problems
Feedback/ Measure position	Doesn't capture motion	a) Will give inaccurate leg position b) Can't supply correct error signal c) Can't produce jump	6	1) Matrix set up incorrectly 2) Missing Forces 3) Over-defined Forces	1) Solve the fwd kinetics using multiple methods 2) Careful consideration of the hand calcs	6	1) Use plots of input to Feedback block and plots of Feedback output 2) Run code independently
Feedback/ Remove nonlinearities	No plant correction	a) Incorrect error signal b) Can't produce jump c) Leg can break	7	1) Feeback error 2) Doesn't accurately describe forces	1) Catch statements in code 2) Careful math	5	1) Error statements in code 2) Failed simulations
	Inaccurate control	a) Can't move leg to correct position b) Can't cushion fall after jump	7	1) Incorrect gain value 2) Incorrect FK	1) Testing and tuning simulation 2) Simple simulations	4	1) Testing leg controller 2) Visual inspection
Documentation / Explain operation of controller	Poor transition to next group of students	a) Loss of time b) Pressure on group c) Big problems d) small progress	8	1) Hard to understand 2) Difficult to demonstrate	1) Explaining Steps 2) Creating simulations and walkthroughs	9	1) Quizes during presentations 2) Sponsor verification of simulations

Design FMEA Example - Wood Chair

Page 2 of 3



de and Effects Analysis

Prepared by: ______________________________

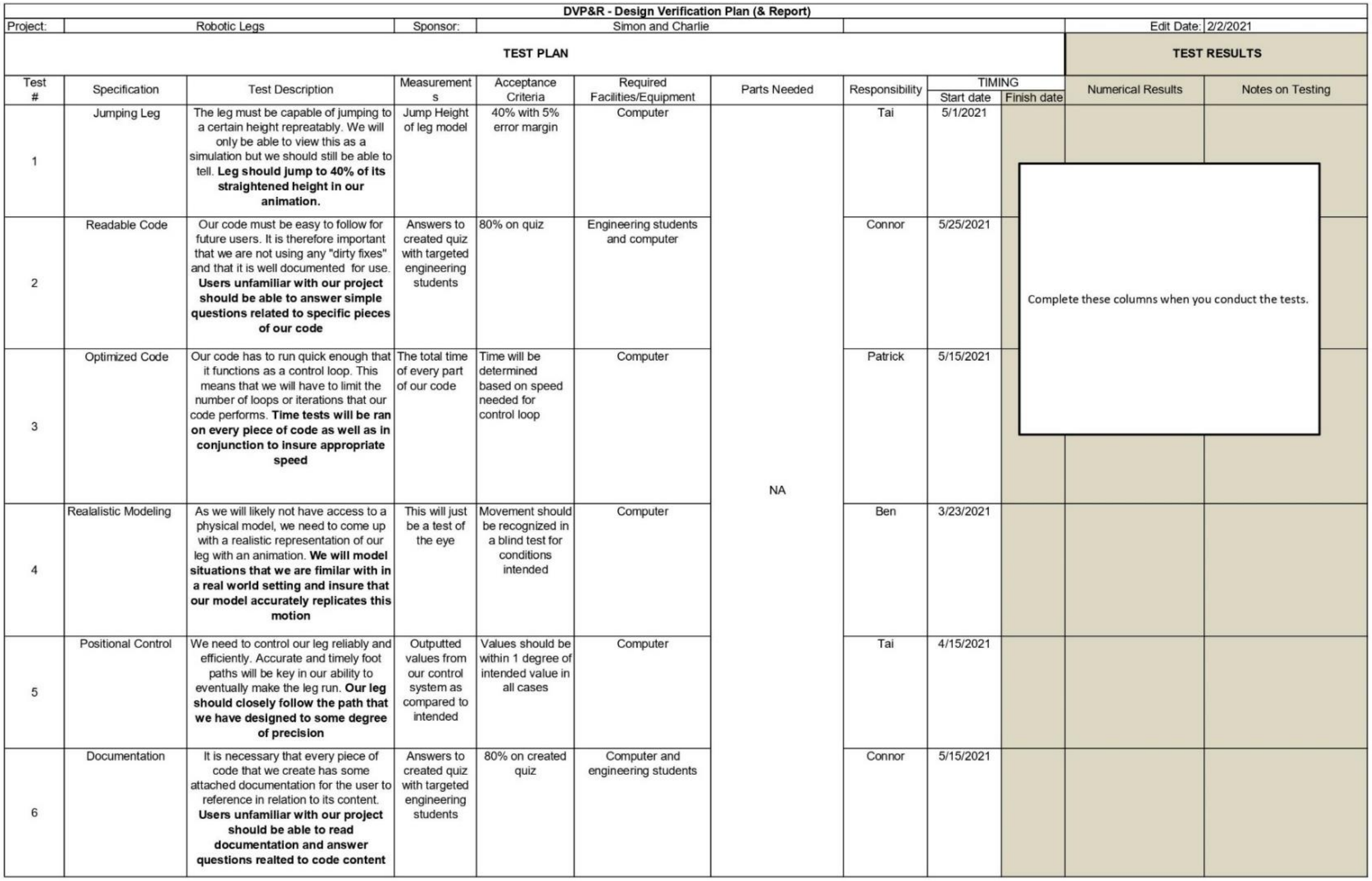
Date: ________________ (orig)

						Action Results				
Occurence	Current Detection Activities	Detection	Priority	Recommended Action(s)	Responsibility & Target Completion Date (DDMMMYY)	Actions Taken	Severity	Occurence	Detection	Updated RPN
7	1) Motion not behaving as inut has instructed	3	168	Isolate contoller block and review controller gains	Tai 19JAN21		7	1	3	21
7	1) Long wait time 2) Responses too long for code to communicate	5	280	Isolate contoller block and review controller gains. Use scope to track time vs position	Ben 19JAN21		8	2	2	32
5	1) Observing scoped outputs in simulink for time response issues	2	80	Confirm open-loop configuration is working. Use scope to isolate error signal	Connor 19JAN21					0
6	1) Observing scoped outputs in simulink for overshoot or blast off problems	5	210	Isolate closed-loop feedback loop. Use scope to track time vs position and adjust error signal	Pattrick 19JAN21		7	2	2	28
6	1) Use plots of input to Feedback block and plots of Feedback output 2) Run code independently	3	108	Isolate feedback loop and look at potentiometer for errors	Tai 26JAN21					0
5	1) Error statements in code 2) Failed simulations	3	105	Isolate feedback loop and adjust potentiometer gain	Ben 26JAN21					0
4	1) Testing leg controller 2) Visual inspection	2	56	Test simulations frequently.	Connor 26JAN21					0
9	1) Quizes during presentations 2) Sponsor verification of simulations	5	360	Continously test math and code. Verify understanding of methods as they are developed	Pattrick 12FEB21		8	3	3	72

ige 2 of 3

Revision Date: 11/20/2020

Attachment L - DVP



DVP&R - Design Verification Plan (& Report)

Project:	Robotic Legs	Sponsor:	Simon and Charlie	Edit Date:	2/2/2021

TEST PLAN | **TEST RESULTS**

Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	TIMING		Numerical Results	Notes on Testing
								Start date	Finish date		
1	Jumping Leg	The leg must be capable of jumping to a certain height repreatably. We will only be able to view this as a simulation but we should still be able to tell. **Leg should jump to 40% of its straightened height in our animation.**	Jump Height of leg model	40% with 5% error margin	Computer	NA	Tai	5/1/2021			
2	Readable Code	Our code must be easy to follow for future users. It is therefore important that we are not using any "dirty fixes" and that it is well documented for use. **Users unfamiliar with our project should be able to answer simple questions related to specific pieces of our code**	Answers to created quiz with targeted engineering students	80% on quiz	Engineering students and computer		Connor	5/25/2021			
3	Optimized Code	Our code has to run quick enough that it functions as a control loop. This means that we will have to limit the number of loops or iterations that our code performs. **Time tests will be ran on every piece of code as well as in conjunction to insure appropriate speed**	The total time of every part of our code	Time will be determined based on speed needed for control loop	Computer		Patrick	5/15/2021			
4	Realalistic Modeling	As we will likely not have access to a physical model, we need to come up with a realistic representation of our leg with an animation. **We will model situations that we are fimilar with in a real world setting and insure that our model accurately replicates this motion**	This will just be a test of the eye	Movement should be recognized in a blind test for conditions intended	Computer		Ben	3/23/2021			
5	Positional Control	We need to control our leg reliably and efficiently. Accurate and timely foot paths will be key in our ability to eventually make the leg run. **Our leg should closely follow the path that we have designed to some degree of precision**	Outputted values from our control system as compared to intended	Values should be within 1 degree of intended value in all cases	Computer		Tai	4/15/2021			
6	Documentation	It is necessary that every piece of code that we create has some attached documentation for the user to reference in relation to its content. **Users unfamiliar with our project should be able to read documentation and answer questions realted to code content**	Answers to created quiz with targeted engineering students	80% on created quiz	Computer and engineering students		Connor	5/15/2021			

Complete these columns when you conduct the tests.

Attachment M – Model Verification Data:

Test 1											
Newtonian			LagrangianC			LagrangianB			Multibody		
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.79	0.00	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.79	0.01	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.02	0.60	-0.79	0.01	-0.97	-2.36
0.02	-0.97	-2.36	0.01	-0.97	-2.36	0.02	0.60	-0.79	0.02	-0.97	-2.36
0.02	-0.96	-2.37	0.01	-0.97	-2.36	0.03	0.61	-0.79	0.02	-0.97	-2.36
0.03	-0.96	-2.37	0.02	-0.97	-2.36	0.03	0.61	-0.79	0.03	-0.96	-2.36
0.03	-0.96	-2.37	0.02	-0.97	-2.36	0.03	0.61	-0.80	0.03	-0.96	-2.37
0.04	-0.95	-2.38	0.03	-0.96	-2.36	0.04	0.61	-0.80	0.04	-0.96	-2.37
0.04	-0.95	-2.38	0.03	-0.96	-2.37	0.04	0.61	-0.81	0.04	-0.96	-2.38
0.05	-0.95	-2.39	0.04	-0.96	-2.37	0.05	0.62	-0.81	0.05	-0.95	-2.38
0.05	-0.94	-2.40	0.04	-0.96	-2.37	0.05	0.62	-0.81	0.05	-0.95	-2.39
0.06	-0.94	-2.41	0.04	-0.96	-2.38	0.06	0.62	-0.82	0.06	-0.95	-2.39
0.06	-0.93	-2.41	0.05	-0.95	-2.38	0.06	0.63	-0.83	0.06	-0.94	-2.40
0.07	-0.92	-2.42	0.05	-0.95	-2.39	0.07	0.63	-0.83	0.07	-0.94	-2.40
0.07	-0.92	-2.43	0.06	-0.95	-2.39	0.07	0.64	-0.84	0.07	-0.93	-2.41
0.08	-0.91	-2.44	0.06	-0.94	-2.40	0.08	0.64	-0.84	0.07	-0.93	-2.42
0.08	-0.90	-2.45	0.07	-0.94	-2.40	0.08	0.65	-0.85	0.08	-0.92	-2.43
0.08	-0.90	-2.47	0.07	-0.93	-2.41	0.09	0.65	-0.86	0.08	-0.92	-2.43
0.09	-0.89	-2.48	0.08	-0.93	-2.42	0.09	0.66	-0.87	0.09	-0.91	-2.44
0.09	-0.88	-2.49	0.08	-0.92	-2.42	0.10	0.66	-0.88	0.09	-0.91	-2.45
0.10	-0.87	-2.50	0.09	-0.92	-2.43	0.10	0.67	-0.89	0.10	-0.90	-2.46
0.10	-0.86	-2.52	0.09	-0.91	-2.44	0.10	0.67	-0.89	0.10	-0.89	-2.47
0.11	-0.86	-2.53	0.10	-0.91	-2.45	0.11	0.68	-0.90	0.11	-0.89	-2.48
0.11	-0.85	-2.55	0.10	-0.90	-2.46	0.11	0.69	-0.91	0.11	-0.88	-2.49
0.12	-0.84	-2.56	0.11	-0.90	-2.47	0.12	0.69	-0.92	0.12	-0.87	-2.51
0.12	-0.83	-2.58	0.11	-0.89	-2.48	0.12	0.70	-0.94	0.12	-0.86	-2.52
0.13	-0.82	-2.59	0.11	-0.88	-2.49	0.13	0.71	-0.95	0.13	-0.86	-2.53
0.13	-0.81	-2.61	0.12	-0.88	-2.50	0.13	0.71	-0.96	0.13	-0.85	-2.54
0.14	-0.80	-2.63	0.12	-0.87	-2.51	0.14	0.72	-0.97	0.14	-0.84	-2.56
0.14	-0.79	-2.64	0.13	-0.86	-2.52	0.14	0.73	-0.98	0.14	-0.83	-2.57
0.15	-0.78	-2.66	0.13	-0.86	-2.53	0.15	0.74	-0.99	0.15	-0.82	-2.58
0.15	-0.77	-2.68	0.14	-0.85	-2.54	0.15	0.74	-1.01	0.15	-0.82	-2.60
0.16	-0.76	-2.70	0.14	-0.84	-2.55	0.16	0.75	-1.02	0.15	-0.81	-2.61
0.16	-0.75	-2.72	0.15	-0.83	-2.57	0.16	0.76	-1.03	0.16	-0.80	-2.63

0.16	-0.74	-2.74	0.15	-0.83	-2.58	0.17	0.77	-1.05	0.16	-0.79	-2.65
0.17	-0.73	-2.77	0.16	-0.82	-2.59	0.17	0.77	-1.06	0.17	-0.78	-2.66
0.17	-0.72	-2.79	0.16	-0.81	-2.61	0.18	0.78	-1.08	0.17	-0.77	-2.68
0.18	-0.71	-2.81	0.17	-0.80	-2.62	0.18	0.79	-1.09	0.18	-0.76	-2.70
0.18	-0.70	-2.84	0.17	-0.80	-2.64	0.18	0.80	-1.11	0.18	-0.75	-2.72
0.19	-0.69	-2.87	0.18	-0.79	-2.65	0.19	0.81	-1.12	0.19	-0.75	-2.73
0.19	-0.68	-2.89	0.18	-0.78	-2.67	0.19	0.81	-1.14	0.19	-0.74	-2.75
0.20	-0.67	-2.92	0.19	-0.77	-2.68	0.20	0.82	-1.16	0.20	-0.73	-2.77
0.20	-0.67	-2.95	0.19	-0.76	-2.70	0.20	0.83	-1.17	0.20	-0.72	-2.79
0.21	-0.66	-2.99	0.19	-0.76	-2.71	0.21	0.84	-1.19	0.21	-0.71	-2.81
0.21	-0.65	-3.03	0.20	-0.75	-2.73	0.21	0.85	-1.21	0.21	-0.70	-2.83
0.22	-0.65	-3.07	0.20	-0.74	-2.75	0.22	0.85	-1.23	0.22	-0.70	-2.86
0.22	-0.64	-3.11	0.21	-0.73	-2.77	0.22	0.86	-1.25	0.22	-0.69	-2.88
0.23	-0.64	-3.17	0.21	-0.72	-2.78	0.23	0.87	-1.27	0.23	-0.68	-2.90
0.23	-0.65	-3.25	0.22	-0.72	-2.80	0.23	0.88	-1.29	0.23	-0.67	-2.93
0.23	-0.66	-3.29	0.22	-0.71	-2.82	0.24	0.88	-1.30	0.23	-0.67	-2.95
0.23	-0.67	-3.35	0.23	-0.70	-2.84				0.24	-0.67	-2.95
0.24	-0.68	-3.38	0.23	-0.69	-2.86						
			0.24	-0.69	-2.87						

Test 2											
Newtonian			LagrangianC			LagrangianB			Multibody		
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.79	0.01	-0.97	-2.36
0.01	-0.97	-2.36	0.01	-0.97	-2.36	0.01	0.60	-0.79	0.01	-0.97	-2.36
0.02	-0.97	-2.36	0.01	-0.97	-2.36	0.02	0.60	-0.79	0.02	-0.97	-2.36
0.02	-0.97	-2.36	0.02	-0.97	-2.36	0.02	0.60	-0.79	0.02	-0.97	-2.36
0.03	-0.97	-2.36	0.02	-0.97	-2.36	0.03	0.60	-0.79	0.03	-0.97	-2.36
0.03	-0.97	-2.36	0.02	-0.97	-2.36	0.03	0.60	-0.79	0.03	-0.97	-2.36
0.04	-0.96	-2.36	0.03	-0.97	-2.36	0.04	0.61	-0.79	0.03	-0.96	-2.36
0.04	-0.96	-2.37	0.03	-0.97	-2.36	0.04	0.61	-0.79	0.04	-0.96	-2.37
0.04	-0.96	-2.37	0.04	-0.97	-2.36	0.05	0.61	-0.79	0.04	-0.96	-2.37
0.05	-0.96	-2.37	0.04	-0.96	-2.36	0.05	0.61	-0.80	0.05	-0.96	-2.37
0.05	-0.96	-2.38	0.05	-0.96	-2.37	0.06	0.61	-0.80	0.05	-0.96	-2.37
0.06	-0.95	-2.38	0.05	-0.96	-2.37	0.06	0.61	-0.80	0.06	-0.96	-2.38
0.06	-0.95	-2.38	0.06	-0.96	-2.37	0.06	0.61	-0.80	0.06	-0.95	-2.38
0.07	-0.95	-2.39	0.06	-0.96	-2.37	0.07	0.62	-0.81	0.07	-0.95	-2.38
0.07	-0.95	-2.39	0.07	-0.96	-2.38	0.07	0.62	-0.81	0.07	-0.95	-2.39

0.08	-0.94	-2.40	0.07	-0.95	-2.38	0.08	0.62	-0.81	0.08	-0.95	-2.39
0.08	-0.94	-2.40	0.08	-0.95	-2.38	0.08	0.62	-0.82	0.08	-0.94	-2.40
0.09	-0.94	-2.41	0.08	-0.95	-2.38	0.09	0.62	-0.82	0.09	-0.94	-2.40
0.09	-0.93	-2.41	0.09	-0.95	-2.39	0.09	0.63	-0.82	0.09	-0.94	-2.41
0.10	-0.93	-2.42	0.09	-0.95	-2.39	0.10	0.63	-0.83	0.10	-0.93	-2.41
0.10	-0.92	-2.42	0.10	-0.94	-2.39	0.10	0.63	-0.83	0.10	-0.93	-2.42
0.11	-0.92	-2.43	0.10	-0.94	-2.40	0.11	0.63	-0.83	0.10	-0.92	-2.42
0.11	-0.92	-2.44	0.10	-0.94	-2.40	0.11	0.64	-0.84	0.11	-0.92	-2.43
0.12	-0.91	-2.44	0.11	-0.94	-2.41	0.12	0.64	-0.84	0.11	-0.92	-2.44
0.12	-0.91	-2.45	0.11	-0.93	-2.41	0.12	0.64	-0.85	0.12	-0.91	-2.44
0.12	-0.90	-2.46	0.12	-0.93	-2.42	0.13	0.65	-0.85	0.12	-0.91	-2.45
0.13	-0.90	-2.46	0.12	-0.93	-2.42	0.13	0.65	-0.86	0.13	-0.90	-2.46
0.13	-0.89	-2.47	0.13	-0.92	-2.43	0.14	0.65	-0.86	0.13	-0.90	-2.46
0.14	-0.89	-2.48	0.13	-0.92	-2.43	0.14	0.66	-0.87	0.14	-0.89	-2.47
0.14	-0.88	-2.49	0.14	-0.92	-2.44	0.14	0.66	-0.87	0.14	-0.89	-2.48
0.15	-0.88	-2.50	0.14	-0.91	-2.44	0.15	0.66	-0.88	0.15	-0.88	-2.49
0.15	-0.87	-2.51	0.15	-0.91	-2.45	0.15	0.67	-0.89	0.15	-0.88	-2.50
0.16	-0.86	-2.52	0.15	-0.90	-2.45	0.16	0.67	-0.89	0.16	-0.87	-2.50
0.16	-0.86	-2.53	0.16	-0.90	-2.46	0.16	0.68	-0.90	0.16	-0.87	-2.51
0.17	-0.85	-2.54	0.16	-0.90	-2.47	0.17	0.68	-0.90	0.17	-0.86	-2.52
0.17	-0.85	-2.55	0.17	-0.89	-2.47	0.17	0.68	-0.91	0.17	-0.85	-2.53
0.18	-0.84	-2.56	0.17	-0.89	-2.48	0.18	0.69	-0.92	0.18	-0.85	-2.54
0.18	-0.83	-2.57	0.18	-0.88	-2.49	0.18	0.69	-0.93	0.18	-0.84	-2.55
0.19	-0.83	-2.58	0.18	-0.88	-2.49	0.19	0.70	-0.93	0.18	-0.84	-2.56
0.19	-0.82	-2.59	0.18	-0.88	-2.50	0.19	0.70	-0.94	0.19	-0.83	-2.57
0.20	-0.81	-2.60	0.19	-0.87	-2.51	0.20	0.71	-0.95	0.19	-0.82	-2.59
0.20	-0.81	-2.61	0.19	-0.87	-2.51	0.20	0.71	-0.96	0.20	-0.82	-2.60
0.20	-0.80	-2.63	0.20	-0.86	-2.52	0.21	0.72	-0.96	0.20	-0.81	-2.61
0.21	-0.79	-2.64	0.20	-0.86	-2.53	0.21	0.72	-0.97	0.21	-0.80	-2.62
0.21	-0.79	-2.65	0.21	-0.85	-2.54	0.22	0.73	-0.98	0.21	-0.80	-2.63
0.22	-0.78	-2.67	0.21	-0.85	-2.55	0.22	0.73	-0.99	0.22	-0.79	-2.65
0.22	-0.77	-2.68	0.22	-0.84	-2.56	0.22	0.74	-1.00	0.22	-0.78	-2.66
0.23	-0.77	-2.69	0.22	-0.84	-2.56	0.23	0.74	-1.01	0.23	-0.78	-2.67
0.23	-0.76	-2.71	0.23	-0.83	-2.57	0.23	0.75	-1.02	0.23	-0.77	-2.69
0.24	-0.75	-2.72	0.23	-0.83	-2.58	0.24	0.75	-1.02	0.24	-0.76	-2.70
			0.24	-0.82	-2.59						

Test 3											
Newtonian			LagrangianC			LagrangianB			Multibody		
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36

0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.01	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.79	0.01	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.78	0.02	-0.97	-2.36
0.02	-0.97	-2.35	0.01	-0.97	-2.36	0.02	0.60	-0.78	0.02	-0.97	-2.36
0.02	-0.97	-2.35	0.01	-0.97	-2.36	0.02	0.60	-0.78	0.03	-0.97	-2.36
0.03	-0.97	-2.35	0.02	-0.97	-2.36	0.03	0.60	-0.78	0.03	-0.97	-2.36
0.03	-0.97	-2.35	0.02	-0.97	-2.35	0.03	0.60	-0.78	0.04	-0.97	-2.36
0.03	-0.97	-2.35	0.03	-0.97	-2.35	0.04	0.60	-0.78	0.04	-0.97	-2.36
0.04	-0.97	-2.35	0.03	-0.97	-2.35	0.04	0.60	-0.78	0.05	-0.97	-2.36
0.04	-0.98	-2.35	0.04	-0.97	-2.35	0.05	0.60	-0.78	0.05	-0.97	-2.36
0.05	-0.98	-2.34	0.04	-0.97	-2.35	0.05	0.60	-0.78	0.06	-0.97	-2.35
0.05	-0.98	-2.34	0.05	-0.97	-2.35	0.06	0.59	-0.77	0.06	-0.97	-2.35
0.06	-0.98	-2.34	0.05	-0.98	-2.35	0.06	0.59	-0.77	0.07	-0.97	-2.35
0.06	-0.98	-2.34	0.06	-0.98	-2.35	0.07	0.59	-0.77	0.07	-0.97	-2.35
0.06	-0.98	-2.33	0.06	-0.98	-2.34	0.07	0.59	-0.77	0.07	-0.97	-2.35
0.07	-0.99	-2.33	0.07	-0.98	-2.34	0.08	0.59	-0.77	0.08	-0.97	-2.35
0.07	-0.99	-2.33	0.07	-0.98	-2.34	0.08	0.59	-0.76	0.08	-0.97	-2.35
0.08	-0.99	-2.32	0.08	-0.98	-2.34	0.08	0.58	-0.76	0.09	-0.97	-2.35
0.08	-0.99	-2.32	0.08	-0.98	-2.33	0.09	0.58	-0.76	0.09	-0.97	-2.35
0.09	-1.00	-2.32	0.08	-0.99	-2.33	0.09	0.58	-0.76	0.10	-0.97	-2.35
0.09	-1.00	-2.31	0.09	-0.99	-2.33	0.10	0.58	-0.75	0.10	-0.97	-2.35
0.10	-1.00	-2.31	0.09	-0.99	-2.33	0.10	0.58	-0.75	0.11	-0.97	-2.35
0.10	-1.01	-2.30	0.10	-0.99	-2.32	0.11	0.57	-0.74	0.11	-0.97	-2.35
0.10	-1.01	-2.30	0.10	-1.00	-2.32	0.11	0.57	-0.74	0.12	-0.97	-2.35
0.11	-1.02	-2.29	0.11	-1.00	-2.32	0.12	0.57	-0.74	0.12	-0.97	-2.35
0.11	-1.02	-2.28	0.11	-1.00	-2.31	0.12	0.56	-0.73	0.13	-0.97	-2.35
0.12	-1.02	-2.28	0.12	-1.00	-2.31	0.13	0.56	-0.73	0.13	-0.97	-2.35
0.12	-1.03	-2.27	0.12	-1.01	-2.30	0.13	0.56	-0.72	0.14	-0.98	-2.35
0.13	-1.03	-2.26	0.13	-1.01	-2.30	0.14	0.55	-0.72	0.14	-0.98	-2.35
0.13	-1.04	-2.26	0.13	-1.01	-2.29	0.14	0.55	-0.71	0.15	-0.98	-2.35
0.14	-1.05	-2.25	0.14	-1.02	-2.29	0.15	0.55	-0.71	0.15	-0.98	-2.35
0.14	-1.05	-2.24	0.14	-1.02	-2.28	0.15	0.54	-0.70	0.15	-0.98	-2.35
0.14	-1.06	-2.23	0.15	-1.02	-2.28	0.16	0.54	-0.70	0.16	-0.98	-2.34
0.15	-1.07	-2.22	0.15	-1.03	-2.27	0.16	0.53	-0.69	0.16	-0.98	-2.34
0.15	-1.07	-2.21	0.16	-1.03	-2.27	0.16	0.53	-0.68	0.17	-0.98	-2.34
0.16	-1.08	-2.20	0.16	-1.04	-2.26	0.17	0.53	-0.68	0.17	-0.98	-2.34
0.16	-1.09	-2.19	0.16	-1.04	-2.26	0.17	0.52	-0.67	0.18	-0.98	-2.34
0.17	-1.10	-2.18	0.17	-1.05	-2.25	0.18	0.51	-0.66	0.18	-0.98	-2.34
0.17	-1.11	-2.16	0.17	-1.05	-2.24	0.18	0.51	-0.66	0.19	-0.98	-2.34
0.17	-1.12	-2.15	0.18	-1.06	-2.23	0.19	0.50	-0.65	0.19	-0.98	-2.34
0.18	-1.13	-2.13	0.18	-1.06	-2.23	0.19	0.50	-0.64	0.20	-0.98	-2.34
0.18	-1.14	-2.12	0.19	-1.07	-2.22	0.20	0.49	-0.63	0.20	-0.98	-2.34

0.19	-1.16	-2.10	0.19	-1.07	-2.21	0.20	0.48	-0.62	0.21	-0.98	-2.34
0.19	-1.17	-2.08	0.20	-1.08	-2.20	0.21	0.48	-0.61	0.21	-0.98	-2.34
0.20	-1.19	-2.06	0.20	-1.09	-2.19	0.21	0.47	-0.60	0.22	-0.98	-2.33
0.20	-1.20	-2.03	0.21	-1.09	-2.18	0.22	0.46	-0.59	0.22	-0.99	-2.33
0.21	-1.22	-2.01	0.21	-1.10	-2.17	0.22	0.45	-0.58	0.23	-0.99	-2.33
0.21	-1.25	-1.98	0.22	-1.11	-2.16	0.23	0.44	-0.56	0.23	-0.99	-2.33
0.21	-1.28	-1.94	0.22	-1.12	-2.15	0.23	0.43	-0.55	0.23	-0.99	-2.33
0.22	-1.31	-1.90	0.23	-1.13	-2.14	0.24	0.42	-0.54	0.24	-0.99	-2.33
0.22	-1.32	-1.88	0.23	-1.14	-2.12						
			0.23	-1.15	-2.11						
			0.24	-1.15	-2.11						

Test 4											
Newtonian			LagrangianC			LagrangianB			Multibody		
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.00	-0.97	-2.36	0.00	-0.97	-2.36	0.00	0.60	-0.79	0.00	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.79	0.01	-0.97	-2.36
0.01	-0.97	-2.36	0.00	-0.97	-2.36	0.01	0.60	-0.78	0.01	-0.97	-2.36
0.01	-0.97	-2.35	0.01	-0.97	-2.36	0.02	0.60	-0.78	0.02	-0.97	-2.35
0.02	-0.97	-2.35	0.01	-0.97	-2.36	0.02	0.60	-0.78	0.02	-0.97	-2.35
0.02	-0.97	-2.35	0.02	-0.97	-2.35	0.03	0.60	-0.78	0.02	-0.97	-2.35
0.03	-0.97	-2.35	0.02	-0.97	-2.35	0.03	0.60	-0.78	0.03	-0.97	-2.35
0.03	-0.97	-2.35	0.03	-0.97	-2.35	0.04	0.60	-0.78	0.03	-0.97	-2.35
0.03	-0.98	-2.35	0.03	-0.97	-2.35	0.04	0.60	-0.78	0.04	-0.98	-2.35
0.04	-0.98	-2.35	0.04	-0.97	-2.35	0.05	0.59	-0.78	0.04	-0.98	-2.34
0.04	-0.98	-2.34	0.04	-0.98	-2.35	0.05	0.59	-0.77	0.05	-0.98	-2.34
0.05	-0.98	-2.34	0.05	-0.98	-2.35	0.06	0.59	-0.77	0.05	-0.98	-2.34
0.05	-0.98	-2.34	0.05	-0.98	-2.34	0.06	0.59	-0.77	0.06	-0.99	-2.33
0.05	-0.98	-2.33	0.06	-0.98	-2.34	0.06	0.59	-0.77	0.06	-0.99	-2.33
0.06	-0.99	-2.33	0.06	-0.98	-2.34	0.07	0.59	-0.76	0.07	-0.99	-2.33
0.06	-0.99	-2.33	0.07	-0.98	-2.34	0.07	0.58	-0.76	0.07	-0.99	-2.32
0.07	-0.99	-2.32	0.07	-0.99	-2.33	0.08	0.58	-0.76	0.08	-1.00	-2.32
0.07	-0.99	-2.32	0.08	-0.99	-2.33	0.08	0.58	-0.75	0.08	-1.00	-2.31
0.07	-1.00	-2.32	0.08	-0.99	-2.33	0.09	0.58	-0.75	0.09	-1.00	-2.31
0.08	-1.00	-2.31	0.08	-0.99	-2.32	0.09	0.57	-0.74	0.09	-1.01	-2.30
0.08	-1.00	-2.31	0.09	-1.00	-2.32	0.10	0.57	-0.74	0.10	-1.01	-2.29
0.09	-1.01	-2.30	0.09	-1.00	-2.31	0.10	0.57	-0.74	0.10	-1.02	-2.29
0.09	-1.01	-2.29	0.10	-1.00	-2.31	0.11	0.56	-0.73	0.10	-1.02	-2.28
0.09	-1.02	-2.29	0.10	-1.00	-2.31	0.11	0.56	-0.73	0.11	-1.03	-2.27

0.10	-1.02	-2.28	0.11	-1.01	-2.30	0.12	0.56	-0.72	0.11	-1.03	-2.26
0.10	-1.03	-2.28	0.11	-1.01	-2.30	0.12	0.55	-0.71	0.12	-1.04	-2.26
0.11	-1.03	-2.27	0.12	-1.02	-2.29	0.13	0.55	-0.71	0.12	-1.05	-2.25
0.11	-1.04	-2.26	0.12	-1.02	-2.28	0.13	0.54	-0.70	0.13	-1.05	-2.24
0.11	-1.04	-2.25	0.13	-1.02	-2.28	0.14	0.54	-0.70	0.13	-1.06	-2.23
0.12	-1.05	-2.24	0.13	-1.03	-2.27	0.14	0.53	-0.69	0.14	-1.07	-2.22
0.12	-1.05	-2.24	0.14	-1.03	-2.26	0.14	0.53	-0.68	0.14	-1.08	-2.21
0.13	-1.06	-2.23	0.14	-1.04	-2.26	0.15	0.52	-0.67	0.15	-1.08	-2.20
0.13	-1.07	-2.22	0.15	-1.04	-2.25	0.15	0.52	-0.66	0.15	-1.09	-2.18
0.13	-1.08	-2.21	0.15	-1.05	-2.24	0.16	0.51	-0.66	0.16	-1.10	-2.17
0.14	-1.08	-2.19	0.15	-1.06	-2.23	0.16	0.50	-0.65	0.16	-1.11	-2.16
0.14	-1.09	-2.18	0.16	-1.06	-2.22	0.17	0.50	-0.64	0.17	-1.12	-2.15
0.15	-1.10	-2.17	0.16	-1.07	-2.22	0.17	0.49	-0.63	0.17	-1.13	-2.13
0.15	-1.11	-2.16	0.17	-1.08	-2.21	0.18	0.48	-0.62	0.18	-1.14	-2.12
0.15	-1.12	-2.14	0.17	-1.08	-2.20	0.18	0.47	-0.61	0.18	-1.15	-2.10
0.16	-1.13	-2.13	0.18	-1.09	-2.19	0.19	0.46	-0.59	0.18	-1.17	-2.08
0.16	-1.15	-2.11	0.18	-1.10	-2.17	0.19	0.46	-0.58	0.19	-1.18	-2.07
0.17	-1.16	-2.09	0.19	-1.11	-2.16	0.20	0.45	-0.57	0.19	-1.19	-2.05
0.17	-1.18	-2.07	0.19	-1.12	-2.15	0.20	0.44	-0.55	0.20	-1.21	-2.03
0.17	-1.19	-2.05	0.20	-1.13	-2.14	0.21	0.42	-0.54	0.20	-1.23	-2.01
0.18	-1.21	-2.03	0.20	-1.14	-2.12	0.21	0.41	-0.52	0.21	-1.24	-1.99
0.18	-1.23	-2.00	0.21	-1.15	-2.11	0.21	0.40	-0.51	0.21	-1.26	-1.96
0.19	-1.25	-1.97	0.21	-1.16	-2.10	0.22	0.39	-0.49	0.22	-1.28	-1.94
0.19	-1.28	-1.94	0.21	-1.17	-2.08	0.22	0.37	-0.47	0.22	-1.30	-1.91
0.19	-1.32	-1.89	0.22	-1.18	-2.07	0.23	0.36	-0.45	0.23	-1.32	-1.89
0.20	-1.36	-1.83	0.22	-1.19	-2.06	0.23	0.34	-0.43	0.23	-1.34	-1.86
0.20	-1.39	-1.80	0.22	-1.20	-2.04	0.24	0.33	-0.42	0.24	-1.36	-1.84
			0.23	-1.21	-2.03						
			0.23	-1.22	-2.02						
			0.23	-1.23	-2.01						
			0.23	-1.24	-1.99						
			0.24	-1.24	-1.99						

Test 5											
Newtonian			LagrangianC			LagrangianB			Multibody		
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.00	-1.16	-2.09
0.01	-1.16	-2.10	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.01	-1.16	-2.10

0.01	-1.16	-2.10	0.00	-1.16	-2.09	0.00	-1.16	-2.09	0.01	-1.16	-2.10
0.02	-1.16	-2.10	0.01	-1.16	-2.10	0.01	-1.16	-2.10	0.02	-1.16	-2.10
0.02	-1.15	-2.10	0.01	-1.16	-2.10	0.01	-1.16	-2.10	0.02	-1.15	-2.10
0.02	-1.15	-2.11	0.02	-1.16	-2.10	0.02	-1.16	-2.10	0.03	-1.15	-2.10
0.03	-1.15	-2.11	0.02	-1.15	-2.10	0.02	-1.15	-2.10	0.03	-1.15	-2.11
0.03	-1.14	-2.12	0.03	-1.15	-2.10	0.03	-1.15	-2.10	0.04	-1.15	-2.11
0.04	-1.14	-2.12	0.03	-1.15	-2.11	0.03	-1.15	-2.11	0.04	-1.15	-2.11
0.04	-1.13	-2.13	0.04	-1.15	-2.11	0.04	-1.15	-2.11	0.05	-1.14	-2.12
0.04	-1.13	-2.13	0.04	-1.14	-2.12	0.04	-1.14	-2.12	0.05	-1.14	-2.12
0.05	-1.12	-2.14	0.05	-1.14	-2.12	0.05	-1.14	-2.12	0.06	-1.13	-2.13
0.05	-1.12	-2.15	0.05	-1.13	-2.13	0.05	-1.13	-2.13	0.06	-1.13	-2.13
0.06	-1.11	-2.16	0.05	-1.13	-2.13	0.05	-1.13	-2.13	0.07	-1.12	-2.14
0.06	-1.11	-2.16	0.06	-1.12	-2.14	0.06	-1.12	-2.14	0.07	-1.12	-2.15
0.06	-1.10	-2.17	0.06	-1.12	-2.15	0.06	-1.12	-2.15	0.07	-1.11	-2.15
0.07	-1.09	-2.18	0.07	-1.11	-2.16	0.07	-1.11	-2.16	0.08	-1.11	-2.16
0.07	-1.09	-2.19	0.07	-1.11	-2.17	0.07	-1.11	-2.17	0.08	-1.10	-2.17
0.08	-1.08	-2.20	0.08	-1.10	-2.17	0.08	-1.10	-2.17	0.09	-1.10	-2.18
0.08	-1.07	-2.21	0.08	-1.09	-2.18	0.08	-1.09	-2.18	0.09	-1.09	-2.19
0.08	-1.06	-2.23	0.09	-1.09	-2.19	0.09	-1.09	-2.19	0.10	-1.08	-2.20
0.09	-1.05	-2.24	0.09	-1.08	-2.20	0.09	-1.08	-2.20	0.10	-1.08	-2.21
0.09	-1.05	-2.25	0.10	-1.07	-2.21	0.10	-1.07	-2.21	0.11	-1.07	-2.22
0.10	-1.04	-2.26	0.10	-1.06	-2.23	0.10	-1.06	-2.23	0.11	-1.06	-2.23
0.10	-1.03	-2.27	0.11	-1.05	-2.24	0.11	-1.05	-2.24	0.12	-1.05	-2.24
0.10	-1.02	-2.29	0.11	-1.05	-2.25	0.11	-1.05	-2.25	0.12	-1.04	-2.25
0.11	-1.01	-2.30	0.12	-1.04	-2.26	0.12	-1.04	-2.26	0.13	-1.03	-2.26
0.11	-1.00	-2.31	0.12	-1.03	-2.27	0.12	-1.03	-2.27	0.13	-1.03	-2.28
0.12	-0.99	-2.33	0.13	-1.02	-2.29	0.13	-1.02	-2.29	0.14	-1.02	-2.29
0.12	-0.98	-2.34	0.13	-1.01	-2.30	0.13	-1.01	-2.30	0.14	-1.01	-2.30
0.12	-0.97	-2.35	0.13	-1.00	-2.31	0.13	-1.00	-2.31	0.15	-1.00	-2.31
0.13	-0.96	-2.37	0.14	-0.99	-2.33	0.14	-0.99	-2.33	0.15	-0.99	-2.33
0.13	-0.95	-2.38	0.14	-0.98	-2.34	0.14	-0.98	-2.34	0.15	-0.98	-2.34
0.14	-0.94	-2.40	0.15	-0.97	-2.35	0.15	-0.97	-2.35	0.16	-0.97	-2.36
0.14	-0.93	-2.41	0.15	-0.96	-2.37	0.15	-0.96	-2.37	0.16	-0.96	-2.37
0.14	-0.92	-2.43	0.16	-0.95	-2.38	0.16	-0.95	-2.38	0.17	-0.95	-2.39
0.15	-0.91	-2.44	0.16	-0.94	-2.40	0.16	-0.94	-2.40	0.17	-0.94	-2.41
0.15	-0.90	-2.46	0.17	-0.93	-2.41	0.17	-0.93	-2.41	0.18	-0.93	-2.42
0.16	-0.89	-2.47	0.17	-0.92	-2.43	0.17	-0.92	-2.43	0.18	-0.91	-2.44
0.16	-0.88	-2.49	0.18	-0.91	-2.45	0.18	-0.91	-2.45	0.19	-0.90	-2.46
0.16	-0.87	-2.51	0.18	-0.90	-2.46	0.18	-0.90	-2.46	0.19	-0.89	-2.47
0.17	-0.86	-2.53	0.19	-0.89	-2.48	0.19	-0.89	-2.48	0.20	-0.88	-2.49
0.17	-0.85	-2.54	0.19	-0.88	-2.49	0.19	-0.88	-2.49	0.20	-0.87	-2.51
0.18	-0.84	-2.56	0.20	-0.87	-2.51	0.20	-0.87	-2.51	0.21	-0.86	-2.53
0.18	-0.83	-2.58	0.20	-0.86	-2.53	0.20	-0.86	-2.53	0.21	-0.85	-2.55

0.18	-0.82	-2.60	0.20	-0.85	-2.55	0.20	-0.85	-2.55	0.22	-0.84	-2.57
0.19	-0.81	-2.62	0.21	-0.84	-2.56	0.21	-0.84	-2.56	0.22	-0.82	-2.59
0.19	-0.80	-2.63	0.21	-0.83	-2.58	0.21	-0.83	-2.58	0.23	-0.81	-2.61
0.20	-0.79	-2.65	0.22	-0.81	-2.60	0.22	-0.81	-2.60	0.23	-0.80	-2.63
0.20	-0.78	-2.67	0.22	-0.80	-2.62	0.22	-0.80	-2.62	0.23	-0.79	-2.65
0.20	-0.77	-2.67	0.23	-0.79	-2.64	0.23	-0.79	-2.64	0.24	-0.79	-2.65
			0.23	-0.78	-2.66	0.23	-0.78	-2.66			
			0.24	-0.78	-2.67	0.24	-0.78	-2.67			

Test 1							
Time [s]	Hip Angle [deg]	Desired Hip Angle [deg]	Error [deg]	Time [s]	Knee Angle [deg]	Desired Knee Angle [deg]	Error [deg]
0.00	30.00	35.77	5.77	0.00	135.00	129.70	-5.30
0.00	30.02	35.79	5.77	0.00	134.98	129.67	-5.31
0.00	30.06	35.82	5.75	0.00	134.94	129.65	-5.29
0.00	30.11	35.85	5.73	0.00	134.90	129.62	-5.28
0.00	30.16	35.87	5.71	0.00	134.85	129.60	-5.26
0.01	30.22	35.90	5.68	0.01	134.81	129.57	-5.23
0.01	30.28	35.93	5.65	0.01	134.76	129.55	-5.21
0.01	30.34	35.95	5.62	0.01	134.70	129.52	-5.18
0.01	30.40	35.98	5.58	0.01	134.65	129.50	-5.15
0.01	30.46	36.00	5.54	0.01	134.59	129.47	-5.12
0.01	30.53	36.03	5.50	0.01	134.53	129.45	-5.08
0.01	30.60	36.06	5.46	0.01	134.47	129.42	-5.04
0.01	30.67	36.08	5.41	0.01	134.40	129.40	-5.01
0.01	30.75	36.11	5.37	0.01	134.34	129.37	-4.97
0.01	30.82	36.14	5.32	0.01	134.27	129.35	-4.92
0.02	30.90	36.16	5.27	0.02	134.20	129.32	-4.88
0.02	30.98	36.19	5.21	0.02	134.13	129.30	-4.83
0.02	31.06	36.22	5.16	0.02	134.06	129.27	-4.79
0.02	31.14	36.24	5.10	0.02	133.98	129.24	-4.74
0.02	31.23	36.27	5.04	0.02	133.91	129.22	-4.69
0.02	31.31	36.30	4.98	0.02	133.83	129.19	-4.64
0.02	31.40	36.32	4.92	0.02	133.75	129.17	-4.58
0.02	31.49	36.35	4.86	0.02	133.67	129.14	-4.53
0.02	31.58	36.38	4.80	0.02	133.59	129.12	-4.47
0.02	31.67	36.40	4.74	0.02	133.51	129.09	-4.42
0.03	31.76	36.43	4.67	0.03	133.42	129.07	-4.36
0.03	31.85	36.46	4.60	0.03	133.34	129.04	-4.30
0.03	31.95	36.48	4.54	0.03	133.25	129.02	-4.24
0.03	32.04	36.51	4.47	0.03	133.17	128.99	-4.18
0.03	32.14	36.54	4.40	0.03	133.08	128.96	-4.12
0.03	32.23	36.56	4.33	0.03	132.99	128.94	-4.06
0.03	32.33	36.59	4.26	0.03	132.91	128.91	-3.99
0.03	32.43	36.62	4.19	0.03	132.82	128.89	-3.93
0.03	32.53	36.64	4.12	0.03	132.73	128.86	-3.87
0.03	32.62	36.67	4.05	0.03	132.64	128.84	-3.80
0.04	32.72	36.70	3.97	0.04	132.55	128.81	-3.74
0.04	32.82	36.72	3.90	0.04	132.46	128.79	-3.67
0.04	32.92	36.75	3.83	0.04	132.36	128.76	-3.60

0.04	33.02	36.78	3.76	0.04	132.27	128.73	-3.54
0.04	33.12	36.80	3.68	0.04	132.18	128.71	-3.47
0.04	33.22	36.83	3.61	0.04	132.09	128.68	-3.41
0.04	33.32	36.86	3.54	0.04	132.00	128.66	-3.34
0.04	33.42	36.88	3.46	0.04	131.90	128.63	-3.27
0.04	33.52	36.91	3.39	0.04	131.81	128.61	-3.21
0.04	33.62	36.94	3.32	0.04	131.72	128.58	-3.14
0.05	33.72	36.96	3.24	0.05	131.63	128.56	-3.07
0.05	33.82	36.99	3.17	0.05	131.53	128.53	-3.00
0.05	33.92	37.02	3.10	0.05	131.44	128.50	-2.94
0.05	34.02	37.04	3.02	0.05	131.35	128.48	-2.87
0.05	34.12	37.07	2.95	0.05	131.26	128.45	-2.80
0.05	34.22	37.10	2.88	0.05	131.16	128.43	-2.74
0.05	34.32	37.12	2.81	0.05	131.07	128.40	-2.67
0.05	34.41	37.15	2.74	0.05	130.98	128.38	-2.60
0.05	34.51	37.18	2.67	0.05	130.89	128.35	-2.54
0.05	34.61	37.20	2.60	0.05	130.80	128.32	-2.47
0.06	34.70	37.23	2.53	0.06	130.71	128.30	-2.41
0.06	34.80	37.26	2.46	0.06	130.62	128.27	-2.34
0.06	34.90	37.28	2.39	0.06	130.53	128.25	-2.28
0.06	34.99	37.31	2.32	0.06	130.44	128.22	-2.22
0.06	35.09	37.34	2.25	0.06	130.35	128.20	-2.15
0.06	35.18	37.36	2.18	0.06	130.26	128.17	-2.09
0.06	35.27	37.39	2.12	0.06	130.17	128.14	-2.03
0.06	35.37	37.42	2.05	0.06	130.08	128.12	-1.97
0.06	35.46	37.44	1.99	0.06	130.00	128.09	-1.90
0.06	35.55	37.47	1.92	0.06	129.91	128.07	-1.84
0.07	35.64	37.50	1.86	0.07	129.83	128.04	-1.78
0.07	35.73	37.52	1.79	0.07	129.74	128.02	-1.72
0.07	35.82	37.55	1.73	0.07	129.66	127.99	-1.66
0.07	35.91	37.58	1.67	0.07	129.57	127.97	-1.61
0.07	35.99	37.60	1.61	0.07	129.49	127.94	-1.55
0.07	36.08	37.63	1.55	0.07	129.41	127.91	-1.49
0.07	36.17	37.66	1.49	0.07	129.32	127.89	-1.44
0.07	36.25	37.68	1.43	0.07	129.24	127.86	-1.38
0.07	36.34	37.71	1.37	0.07	129.16	127.84	-1.33
0.07	36.42	37.73	1.32	0.07	129.08	127.81	-1.27
0.08	36.50	37.76	1.26	0.08	129.00	127.79	-1.22
0.08	36.58	37.79	1.21	0.08	128.93	127.76	-1.17
0.08	36.66	37.81	1.15	0.08	128.85	127.73	-1.11
0.08	36.74	37.84	1.10	0.08	128.77	127.71	-1.06
0.08	36.82	37.87	1.05	0.08	128.70	127.68	-1.01
0.08	36.90	37.89	0.99	0.08	128.62	127.66	-0.96

0.08	36.98	37.92	0.94	0.08	128.55	127.63	-0.92
0.08	37.05	37.95	0.89	0.08	128.48	127.61	-0.87
0.08	37.13	37.97	0.85	0.08	128.40	127.58	-0.82
0.08	37.20	38.00	0.80	0.08	128.33	127.55	-0.78
0.09	37.28	38.03	0.75	0.09	128.26	127.53	-0.73
0.09	37.35	38.05	0.70	0.09	128.19	127.50	-0.69
0.09	37.42	38.08	0.66	0.09	128.12	127.48	-0.64
0.09	37.49	38.10	0.61	0.09	128.05	127.45	-0.60
0.09	37.56	38.13	0.57	0.09	127.99	127.43	-0.56
0.09	37.63	38.16	0.53	0.09	127.92	127.40	-0.52
0.09	37.70	38.18	0.49	0.09	127.85	127.38	-0.48
0.09	37.76	38.21	0.45	0.09	127.79	127.35	-0.44
0.09	37.83	38.24	0.41	0.09	127.73	127.32	-0.40
0.09	37.89	38.26	0.37	0.09	127.66	127.30	-0.36
0.10	37.96	38.29	0.33	0.10	127.60	127.27	-0.33
0.10	38.02	38.32	0.29	0.10	127.54	127.25	-0.29
0.10	38.08	38.34	0.26	0.10	127.48	127.22	-0.26
0.10	38.14	38.37	0.22	0.10	127.42	127.20	-0.22
0.10	38.20	38.39	0.19	0.10	127.36	127.17	-0.19

Test 2							
Time [s]	Hip Angle [deg]	Desired Hip Angle [deg]	Error [deg]	Time [s]	Knee Angle [deg]	Desired Knee Angle [deg]	Error [deg]
0.00	30.00	35.77	5.77	0.00	135.00	129.70	-5.30
0.00	30.00	35.77	5.76	0.00	135.00	129.70	-5.30
0.00	30.01	35.77	5.76	0.00	134.99	129.70	-5.29
0.00	30.02	35.77	5.75	0.00	134.99	129.70	-5.29
0.00	30.03	35.77	5.74	0.00	134.98	129.70	-5.28
0.01	30.04	35.77	5.72	0.01	134.96	129.70	-5.26
0.01	30.06	35.77	5.70	0.01	134.95	129.70	-5.25
0.01	30.08	35.77	5.68	0.01	134.93	129.70	-5.23
0.01	30.11	35.77	5.66	0.01	134.90	129.70	-5.20
0.01	30.14	35.77	5.63	0.01	134.88	129.70	-5.18
0.01	30.17	35.77	5.60	0.01	134.85	129.70	-5.15
0.01	30.20	35.77	5.57	0.01	134.82	129.70	-5.13
0.01	30.23	35.77	5.53	0.01	134.79	129.70	-5.09
0.01	30.27	35.77	5.50	0.01	134.76	129.70	-5.06
0.01	30.31	35.77	5.46	0.01	134.73	129.70	-5.03
0.02	30.35	35.77	5.41	0.02	134.69	129.70	-4.99
0.02	30.40	35.77	5.37	0.02	134.65	129.70	-4.95
0.02	30.44	35.77	5.32	0.02	134.61	129.70	-4.91
0.02	30.49	35.77	5.27	0.02	134.56	129.70	-4.86

0.02	30.54	35.77	5.22	0.02	134.52	129.70	-4.82
0.02	30.59	35.77	5.17	0.02	134.47	129.70	-4.77
0.02	30.65	35.77	5.12	0.02	134.43	129.70	-4.73
0.02	30.70	35.77	5.06	0.02	134.38	129.70	-4.68
0.02	30.76	35.77	5.01	0.02	134.33	129.70	-4.63
0.02	30.82	35.77	4.95	0.02	134.27	129.70	-4.57
0.03	30.88	35.77	4.89	0.03	134.22	129.70	-4.52
0.03	30.94	35.77	4.83	0.03	134.17	129.70	-4.47
0.03	31.00	35.77	4.77	0.03	134.11	129.70	-4.41
0.03	31.06	35.77	4.70	0.03	134.05	129.70	-4.35
0.03	31.13	35.77	4.64	0.03	134.00	129.70	-4.30
0.03	31.19	35.77	4.57	0.03	133.94	129.70	-4.24
0.03	31.26	35.77	4.51	0.03	133.88	129.70	-4.18
0.03	31.33	35.77	4.44	0.03	133.82	129.70	-4.12
0.03	31.39	35.77	4.37	0.03	133.76	129.70	-4.06
0.03	31.46	35.77	4.30	0.03	133.69	129.70	-3.99
0.04	31.53	35.77	4.23	0.04	133.63	129.70	-3.93
0.04	31.60	35.77	4.16	0.04	133.57	129.70	-3.87
0.04	31.67	35.77	4.09	0.04	133.50	129.70	-3.80
0.04	31.74	35.77	4.02	0.04	133.44	129.70	-3.74
0.04	31.82	35.77	3.95	0.04	133.37	129.70	-3.68
0.04	31.89	35.77	3.88	0.04	133.31	129.70	-3.61
0.04	31.96	35.77	3.81	0.04	133.24	129.70	-3.54
0.04	32.03	35.77	3.74	0.04	133.18	129.70	-3.48
0.04	32.10	35.77	3.66	0.04	133.11	129.70	-3.41
0.04	32.18	35.77	3.59	0.04	133.05	129.70	-3.35
0.05	32.25	35.77	3.52	0.05	132.98	129.70	-3.28
0.05	32.32	35.77	3.44	0.05	132.91	129.70	-3.21
0.05	32.39	35.77	3.37	0.05	132.85	129.70	-3.15
0.05	32.47	35.77	3.30	0.05	132.78	129.70	-3.08
0.05	32.54	35.77	3.23	0.05	132.72	129.70	-3.02
0.05	32.61	35.77	3.15	0.05	132.65	129.70	-2.95
0.05	32.68	35.77	3.08	0.05	132.58	129.70	-2.88
0.05	32.76	35.77	3.01	0.05	132.52	129.70	-2.82
0.05	32.83	35.77	2.94	0.05	132.45	129.70	-2.75
0.05	32.90	35.77	2.87	0.05	132.38	129.70	-2.69
0.06	32.97	35.77	2.80	0.06	132.32	129.70	-2.62
0.06	33.04	35.77	2.72	0.06	132.25	129.70	-2.55
0.06	33.11	35.77	2.65	0.06	132.19	129.70	-2.49
0.06	33.18	35.77	2.58	0.06	132.12	129.70	-2.42
0.06	33.25	35.77	2.51	0.06	132.06	129.70	-2.36
0.06	33.32	35.77	2.45	0.06	132.00	129.70	-2.30
0.06	33.39	35.77	2.38	0.06	131.93	129.70	-2.23

0.06	33.46	35.77	2.31	0.06	131.87	129.70	-2.17
0.06	33.52	35.77	2.24	0.06	131.81	129.70	-2.11
0.06	33.59	35.77	2.17	0.06	131.75	129.70	-2.05
0.07	33.66	35.77	2.11	0.07	131.68	129.70	-1.98
0.07	33.72	35.77	2.04	0.07	131.62	129.70	-1.92
0.07	33.79	35.77	1.98	0.07	131.56	129.70	-1.86
0.07	33.85	35.77	1.91	0.07	131.50	129.70	-1.80
0.07	33.92	35.77	1.85	0.07	131.44	129.70	-1.74
0.07	33.98	35.77	1.79	0.07	131.39	129.70	-1.69
0.07	34.04	35.77	1.72	0.07	131.33	129.70	-1.63
0.07	34.10	35.77	1.66	0.07	131.27	129.70	-1.57
0.07	34.16	35.77	1.60	0.07	131.21	129.70	-1.51
0.07	34.22	35.77	1.54	0.07	131.16	129.70	-1.46
0.08	34.28	35.77	1.48	0.08	131.10	129.70	-1.40
0.08	34.34	35.77	1.43	0.08	131.05	129.70	-1.35
0.08	34.40	35.77	1.37	0.08	131.00	129.70	-1.30
0.08	34.45	35.77	1.31	0.08	130.94	129.70	-1.24
0.08	34.51	35.77	1.26	0.08	130.89	129.70	-1.19
0.08	34.56	35.77	1.20	0.08	130.84	129.70	-1.14
0.08	34.62	35.77	1.15	0.08	130.79	129.70	-1.09
0.08	34.67	35.77	1.09	0.08	130.74	129.70	-1.04
0.08	34.72	35.77	1.04	0.08	130.69	129.70	-0.99
0.08	34.78	35.77	0.99	0.08	130.64	129.70	-0.94
0.09	34.83	35.77	0.94	0.09	130.59	129.70	-0.89
0.09	34.88	35.77	0.89	0.09	130.55	129.70	-0.85
0.09	34.92	35.77	0.84	0.09	130.50	129.70	-0.80
0.09	34.97	35.77	0.79	0.09	130.46	129.70	-0.76
0.09	35.02	35.77	0.75	0.09	130.41	129.70	-0.71
0.09	35.06	35.77	0.70	0.09	130.37	129.70	-0.67
0.09	35.11	35.77	0.66	0.09	130.33	129.70	-0.63
0.09	35.15	35.77	0.61	0.09	130.28	129.70	-0.59
0.09	35.20	35.77	0.57	0.09	130.24	129.70	-0.54
0.09	35.24	35.77	0.53	0.09	130.20	129.70	-0.50
0.10	35.28	35.77	0.49	0.10	130.16	129.70	-0.47
0.10	35.32	35.77	0.44	0.10	130.13	129.70	-0.43
0.10	35.36	35.77	0.41	0.10	130.09	129.70	-0.39
0.10	35.40	35.77	0.37	0.10	130.05	129.70	-0.35
0.10	35.44	35.77	0.33	0.10	130.02	129.70	-0.32

Test 3							
Time [s]	Hip Angle [deg]	Desired Hip Angle [deg]	Error [deg]	Time [s]	Knee Angle [deg]	Desired Knee Angle [deg]	Error [deg]

0.00	30.00	19.05	-10.95	0.00	135.00	143.71	8.71
0.00	30.00	19.06	-10.94	0.00	135.00	143.71	8.70
0.00	29.99	19.06	-10.93	0.00	135.01	143.70	8.70
0.00	29.98	19.06	-10.92	0.00	135.02	143.70	8.68
0.00	29.96	19.06	-10.90	0.00	135.04	143.70	8.67
0.01	29.93	19.07	-10.87	0.01	135.06	143.70	8.64
0.01	29.90	19.07	-10.84	0.01	135.09	143.70	8.61
0.01	29.87	19.07	-10.80	0.01	135.12	143.70	8.58
0.01	29.83	19.07	-10.75	0.01	135.15	143.70	8.54
0.01	29.78	19.07	-10.71	0.01	135.20	143.69	8.50
0.01	29.73	19.08	-10.65	0.01	135.24	143.69	8.45
0.01	29.67	19.08	-10.59	0.01	135.29	143.69	8.40
0.01	29.61	19.08	-10.53	0.01	135.34	143.69	8.35
0.01	29.54	19.08	-10.46	0.01	135.40	143.69	8.29
0.01	29.48	19.08	-10.39	0.01	135.46	143.69	8.23
0.02	29.40	19.09	-10.32	0.02	135.53	143.69	8.16
0.02	29.32	19.09	-10.24	0.02	135.59	143.68	8.09
0.02	29.24	19.09	-10.15	0.02	135.67	143.68	8.02
0.02	29.16	19.09	-10.06	0.02	135.74	143.68	7.94
0.02	29.07	19.09	-9.97	0.02	135.82	143.68	7.86
0.02	28.98	19.10	-9.88	0.02	135.90	143.68	7.78
0.02	28.88	19.10	-9.78	0.02	135.98	143.68	7.70
0.02	28.78	19.10	-9.68	0.02	136.07	143.68	7.61
0.02	28.68	19.10	-9.58	0.02	136.15	143.67	7.52
0.02	28.58	19.10	-9.47	0.02	136.24	143.67	7.43
0.03	28.47	19.11	-9.36	0.03	136.34	143.67	7.34
0.03	28.36	19.11	-9.25	0.03	136.43	143.67	7.24
0.03	28.25	19.11	-9.14	0.03	136.53	143.67	7.14
0.03	28.14	19.11	-9.02	0.03	136.62	143.67	7.04
0.03	28.02	19.11	-8.91	0.03	136.72	143.67	6.94
0.03	27.90	19.12	-8.79	0.03	136.82	143.66	6.84
0.03	27.78	19.12	-8.66	0.03	136.93	143.66	6.74
0.03	27.66	19.12	-8.54	0.03	137.03	143.66	6.63
0.03	27.54	19.12	-8.42	0.03	137.13	143.66	6.53
0.03	27.42	19.13	-8.29	0.03	137.24	143.66	6.42
0.04	27.29	19.13	-8.16	0.04	137.35	143.66	6.31
0.04	27.16	19.13	-8.03	0.04	137.45	143.66	6.20
0.04	27.03	19.13	-7.90	0.04	137.56	143.65	6.09
0.04	26.90	19.13	-7.77	0.04	137.67	143.65	5.98
0.04	26.77	19.14	-7.64	0.04	137.78	143.65	5.87
0.04	26.64	19.14	-7.51	0.04	137.89	143.65	5.76
0.04	26.51	19.14	-7.37	0.04	138.00	143.65	5.65
0.04	26.38	19.14	-7.24	0.04	138.11	143.65	5.54

0.04	26.25	19.14	-7.10	0.04	138.22	143.65	5.43
0.04	26.11	19.15	-6.97	0.04	138.33	143.65	5.32
0.05	25.98	19.15	-6.83	0.05	138.44	143.64	5.20
0.05	25.85	19.15	-6.70	0.05	138.55	143.64	5.09
0.05	25.71	19.15	-6.56	0.05	138.66	143.64	4.98
0.05	25.58	19.15	-6.42	0.05	138.77	143.64	4.87
0.05	25.44	19.16	-6.29	0.05	138.88	143.64	4.76
0.05	25.31	19.16	-6.15	0.05	138.99	143.64	4.65
0.05	25.17	19.16	-6.01	0.05	139.10	143.64	4.54
0.05	25.04	19.16	-5.88	0.05	139.21	143.63	4.43
0.05	24.91	19.16	-5.74	0.05	139.32	143.63	4.32
0.05	24.77	19.17	-5.61	0.05	139.42	143.63	4.21
0.06	24.64	19.17	-5.47	0.06	139.53	143.63	4.10
0.06	24.51	19.17	-5.34	0.06	139.64	143.63	3.99
0.06	24.38	19.17	-5.20	0.06	139.74	143.63	3.89
0.06	24.25	19.17	-5.07	0.06	139.85	143.63	3.78
0.06	24.12	19.18	-4.94	0.06	139.95	143.62	3.67
0.06	23.99	19.18	-4.81	0.06	140.05	143.62	3.57
0.06	23.86	19.18	-4.68	0.06	140.15	143.62	3.47
0.06	23.73	19.18	-4.55	0.06	140.25	143.62	3.36
0.06	23.60	19.18	-4.42	0.06	140.35	143.62	3.26
0.06	23.48	19.19	-4.29	0.06	140.45	143.62	3.16
0.07	23.35	19.19	-4.16	0.07	140.55	143.62	3.06
0.07	23.23	19.19	-4.04	0.07	140.65	143.61	2.97
0.07	23.10	19.19	-3.91	0.07	140.74	143.61	2.87
0.07	22.98	19.20	-3.79	0.07	140.84	143.61	2.77
0.07	22.86	19.20	-3.66	0.07	140.93	143.61	2.68
0.07	22.74	19.20	-3.54	0.07	141.02	143.61	2.59
0.07	22.62	19.20	-3.42	0.07	141.11	143.61	2.49
0.07	22.51	19.20	-3.30	0.07	141.20	143.61	2.40
0.07	22.39	19.21	-3.19	0.07	141.29	143.60	2.31
0.07	22.28	19.21	-3.07	0.07	141.38	143.60	2.22
0.08	22.16	19.21	-2.95	0.08	141.46	143.60	2.14
0.08	22.05	19.21	-2.84	0.08	141.55	143.60	2.05
0.08	21.94	19.21	-2.73	0.08	141.63	143.60	1.97
0.08	21.83	19.22	-2.62	0.08	141.71	143.60	1.88
0.08	21.73	19.22	-2.51	0.08	141.79	143.60	1.80
0.08	21.62	19.22	-2.40	0.08	141.87	143.59	1.72
0.08	21.52	19.22	-2.30	0.08	141.95	143.59	1.64
0.08	21.42	19.22	-2.19	0.08	142.02	143.59	1.57
0.08	21.32	19.23	-2.09	0.08	142.10	143.59	1.49
0.08	21.22	19.23	-1.99	0.08	142.17	143.59	1.42
0.09	21.12	19.23	-1.89	0.09	142.24	143.59	1.34

0.09	21.02	19.23	-1.79	0.09	142.31	143.59	1.27
0.09	20.93	19.23	-1.70	0.09	142.38	143.58	1.20
0.09	20.84	19.24	-1.60	0.09	142.45	143.58	1.13
0.09	20.75	19.24	-1.51	0.09	142.52	143.58	1.07
0.09	20.66	19.24	-1.42	0.09	142.58	143.58	1.00
0.09	20.57	19.24	-1.33	0.09	142.64	143.58	0.93
0.09	20.49	19.24	-1.24	0.09	142.71	143.58	0.87
0.09	20.40	19.25	-1.15	0.09	142.77	143.58	0.81
0.09	20.32	19.25	-1.07	0.09	142.82	143.57	0.75
0.10	20.24	19.25	-0.99	0.10	142.88	143.57	0.69
0.10	20.16	19.25	-0.91	0.10	142.94	143.57	0.63
0.10	20.08	19.25	-0.83	0.10	142.99	143.57	0.58
0.10	20.01	19.26	-0.75	0.10	143.05	143.57	0.52
0.10	19.93	19.26	-0.68	0.10	143.10	143.57	0.47

Test 4							
Time [s]	Hip Angle [deg]	Desired Hip Angle [deg]	Error [deg]	Time [s]	Knee Angle [deg]	Desired Knee Angle [deg]	Error [deg]
0.00	60.00	46.65	-13.35	0.00	125.00	143.77	18.77
0.00	60.01	46.66	-13.35	0.00	124.99	143.75	18.75
0.00	60.01	46.67	-13.34	0.00	124.98	143.72	18.74
0.00	60.02	46.69	-13.33	0.00	124.98	143.70	18.72
0.00	60.01	46.70	-13.31	0.00	124.99	143.68	18.69
0.01	59.99	46.72	-13.28	0.01	125.01	143.66	18.65
0.01	59.97	46.73	-13.24	0.01	125.04	143.64	18.60
0.01	59.94	46.74	-13.20	0.01	125.08	143.62	18.54
0.01	59.90	46.76	-13.15	0.01	125.13	143.60	18.47
0.01	59.86	46.77	-13.09	0.01	125.18	143.58	18.39
0.01	59.81	46.78	-13.03	0.01	125.25	143.56	18.31
0.01	59.75	46.80	-12.96	0.01	125.33	143.54	18.21
0.01	59.69	46.81	-12.88	0.01	125.41	143.51	18.10
0.01	59.62	46.82	-12.80	0.01	125.50	143.49	17.99
0.01	59.55	46.84	-12.71	0.01	125.60	143.47	17.87
0.02	59.47	46.85	-12.61	0.02	125.71	143.45	17.75
0.02	59.38	46.87	-12.52	0.02	125.82	143.43	17.61
0.02	59.29	46.88	-12.41	0.02	125.94	143.41	17.47
0.02	59.20	46.89	-12.30	0.02	126.06	143.39	17.32
0.02	59.10	46.91	-12.19	0.02	126.20	143.37	17.17
0.02	58.99	46.92	-12.07	0.02	126.34	143.35	17.01
0.02	58.88	46.93	-11.95	0.02	126.48	143.33	16.85
0.02	58.77	46.95	-11.82	0.02	126.63	143.30	16.67

0.02	58.66	46.96	-11.69	0.02	126.78	143.28	16.50
0.02	58.54	46.98	-11.56	0.02	126.94	143.26	16.32
0.03	58.41	46.99	-11.42	0.03	127.11	143.24	16.13
0.03	58.29	47.00	-11.28	0.03	127.28	143.22	15.94
0.03	58.16	47.02	-11.14	0.03	127.45	143.20	15.75
0.03	58.03	47.03	-10.99	0.03	127.62	143.18	15.55
0.03	57.89	47.05	-10.85	0.03	127.81	143.16	15.35
0.03	57.75	47.06	-10.70	0.03	127.99	143.14	15.15
0.03	57.62	47.07	-10.54	0.03	128.18	143.11	14.94
0.03	57.47	47.09	-10.39	0.03	128.37	143.09	14.73
0.03	57.33	47.10	-10.23	0.03	128.56	143.07	14.51
0.03	57.19	47.11	-10.07	0.03	128.75	143.05	14.30
0.04	57.04	47.13	-9.91	0.04	128.95	143.03	14.08
0.04	56.89	47.14	-9.75	0.04	129.15	143.01	13.86
0.04	56.74	47.16	-9.58	0.04	129.36	142.99	13.63
0.04	56.59	47.17	-9.42	0.04	129.56	142.97	13.41
0.04	56.44	47.18	-9.25	0.04	129.77	142.95	13.18
0.04	56.28	47.20	-9.09	0.04	129.97	142.93	12.95
0.04	56.13	47.21	-8.92	0.04	130.18	142.90	12.72
0.04	55.98	47.22	-8.75	0.04	130.39	142.88	12.49
0.04	55.82	47.24	-8.58	0.04	130.60	142.86	12.26
0.04	55.67	47.25	-8.41	0.04	130.81	142.84	12.03
0.05	55.51	47.27	-8.24	0.05	131.02	142.82	11.79
0.05	55.36	47.28	-8.08	0.05	131.24	142.80	11.56
0.05	55.20	47.29	-7.91	0.05	131.45	142.78	11.33
0.05	55.04	47.31	-7.74	0.05	131.66	142.76	11.09
0.05	54.89	47.32	-7.57	0.05	131.88	142.74	10.86
0.05	54.73	47.34	-7.40	0.05	132.09	142.71	10.62
0.05	54.58	47.35	-7.23	0.05	132.30	142.69	10.39
0.05	54.42	47.36	-7.06	0.05	132.52	142.67	10.16
0.05	54.27	47.38	-6.89	0.05	132.73	142.65	9.92
0.05	54.12	47.39	-6.73	0.05	132.94	142.63	9.69
0.06	53.97	47.41	-6.56	0.06	133.15	142.61	9.46
0.06	53.81	47.42	-6.40	0.06	133.36	142.59	9.23
0.06	53.66	47.43	-6.23	0.06	133.57	142.57	9.00
0.06	53.52	47.45	-6.07	0.06	133.78	142.55	8.77
0.06	53.37	47.46	-5.91	0.06	133.98	142.53	8.54
0.06	53.22	47.47	-5.74	0.06	134.19	142.50	8.32
0.06	53.07	47.49	-5.59	0.06	134.39	142.48	8.09
0.06	52.93	47.50	-5.43	0.06	134.59	142.46	7.87
0.06	52.79	47.52	-5.27	0.06	134.80	142.44	7.65
0.06	52.64	47.53	-5.11	0.06	134.99	142.42	7.43
0.07	52.50	47.54	-4.96	0.07	135.19	142.40	7.21

0.07	52.36	47.56	-4.81	0.07	135.39	142.38	6.99
0.07	52.23	47.57	-4.65	0.07	135.58	142.36	6.78
0.07	52.09	47.59	-4.51	0.07	135.77	142.34	6.56
0.07	51.96	47.60	-4.36	0.07	135.96	142.31	6.35
0.07	51.82	47.61	-4.21	0.07	136.15	142.29	6.14
0.07	51.69	47.63	-4.07	0.07	136.34	142.27	5.94
0.07	51.57	47.64	-3.92	0.07	136.52	142.25	5.73
0.07	51.44	47.66	-3.78	0.07	136.70	142.23	5.53
0.07	51.31	47.67	-3.64	0.07	136.88	142.21	5.33
0.08	51.19	47.68	-3.51	0.08	137.05	142.19	5.13
0.08	51.07	47.70	-3.37	0.08	137.23	142.17	4.94
0.08	50.95	47.71	-3.24	0.08	137.40	142.15	4.75
0.08	50.83	47.73	-3.11	0.08	137.57	142.13	4.56
0.08	50.72	47.74	-2.98	0.08	137.73	142.10	4.37
0.08	50.60	47.75	-2.85	0.08	137.90	142.08	4.19
0.08	50.49	47.77	-2.72	0.08	138.06	142.06	4.00
0.08	50.38	47.78	-2.60	0.08	138.22	142.04	3.83
0.08	50.27	47.79	-2.48	0.08	138.37	142.02	3.65
0.08	50.17	47.81	-2.36	0.08	138.52	142.00	3.48
0.09	50.07	47.82	-2.24	0.09	138.67	141.98	3.31
0.09	49.97	47.84	-2.13	0.09	138.82	141.96	3.14
0.09	49.87	47.85	-2.02	0.09	138.96	141.94	2.97
0.09	49.77	47.86	-1.90	0.09	139.10	141.92	2.81
0.09	49.68	47.88	-1.80	0.09	139.24	141.89	2.65
0.09	49.58	47.89	-1.69	0.09	139.38	141.87	2.50
0.09	49.49	47.91	-1.59	0.09	139.51	141.85	2.34
0.09	49.40	47.92	-1.48	0.09	139.64	141.83	2.19
0.09	49.32	47.93	-1.38	0.09	139.77	141.81	2.05
0.09	49.23	47.95	-1.29	0.09	139.89	141.79	1.90
0.10	49.15	47.96	-1.19	0.10	140.01	141.77	1.76
0.10	49.07	47.98	-1.10	0.10	140.13	141.75	1.62
0.10	48.99	47.99	-1.00	0.10	140.24	141.73	1.49
0.10	48.92	48.00	-0.91	0.10	140.35	141.71	1.35
0.10	48.85	48.02	-0.83	0.10	140.46	141.69	1.22

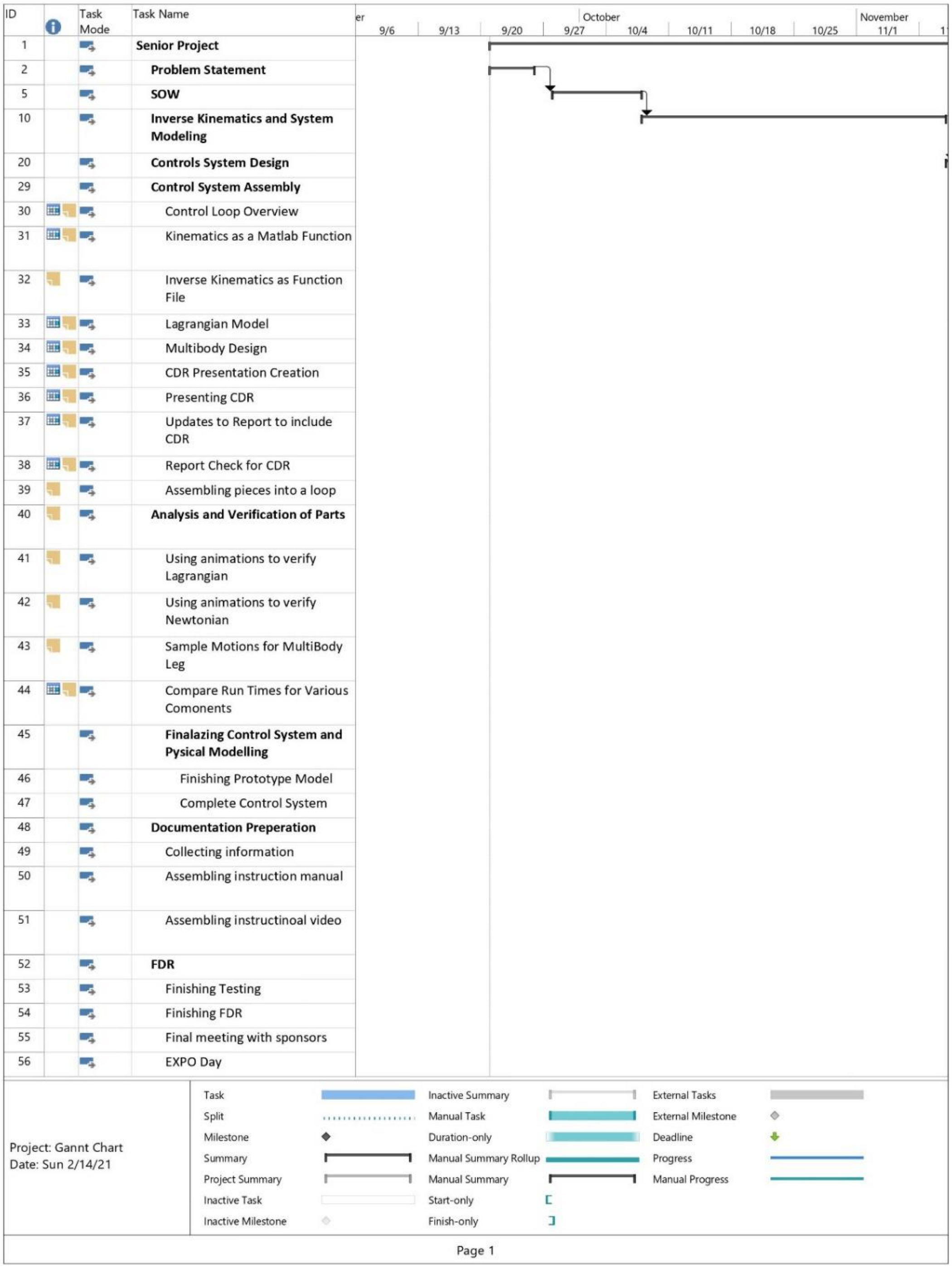
Test 5							
Time [s]	Hip Angle [deg]	Desired Hip Angle [deg]	Error [deg]	Time [s]	Knee Angle [deg]	Desired Knee Angle [deg]	Error [deg]
0.00	60.00	46.65	-13.35	0.00	125.00	143.77	18.77
0.00	60.00	46.65	-13.35	0.00	125.00	143.77	18.77
0.00	59.99	46.65	-13.35	0.00	125.01	143.77	18.76
0.00	59.98	46.65	-13.33	0.00	125.03	143.77	18.74

0.00	59.95	46.65	-13.31	0.00	125.06	143.77	18.70
0.01	59.92	46.65	-13.27	0.01	125.11	143.77	18.66
0.01	59.88	46.65	-13.23	0.01	125.16	143.77	18.61
0.01	59.83	46.65	-13.19	0.01	125.22	143.77	18.55
0.01	59.78	46.65	-13.13	0.01	125.29	143.77	18.48
0.01	59.72	46.65	-13.07	0.01	125.37	143.77	18.40
0.01	59.65	46.65	-13.01	0.01	125.46	143.77	18.31
0.01	59.58	46.65	-12.93	0.01	125.56	143.77	18.21
0.01	59.50	46.65	-12.85	0.01	125.66	143.77	18.10
0.01	59.41	46.65	-12.77	0.01	125.78	143.77	17.99
0.01	59.32	46.65	-12.68	0.01	125.90	143.77	17.87
0.02	59.23	46.65	-12.58	0.02	126.02	143.77	17.74
0.02	59.13	46.65	-12.48	0.02	126.16	143.77	17.61
0.02	59.02	46.65	-12.37	0.02	126.30	143.77	17.47
0.02	58.91	46.65	-12.26	0.02	126.45	143.77	17.32
0.02	58.79	46.65	-12.14	0.02	126.60	143.77	17.16
0.02	58.67	46.65	-12.02	0.02	126.76	143.77	17.00
0.02	58.55	46.65	-11.90	0.02	126.93	143.77	16.84
0.02	58.42	46.65	-11.77	0.02	127.10	143.77	16.67
0.02	58.29	46.65	-11.64	0.02	127.28	143.77	16.49
0.02	58.15	46.65	-11.51	0.02	127.46	143.77	16.31
0.03	58.01	46.65	-11.37	0.03	127.64	143.77	16.12
0.03	57.87	46.65	-11.22	0.03	127.83	143.77	15.93
0.03	57.73	46.65	-11.08	0.03	128.03	143.77	15.74
0.03	57.58	46.65	-10.93	0.03	128.22	143.77	15.54
0.03	57.43	46.65	-10.78	0.03	128.43	143.77	15.34
0.03	57.28	46.65	-10.63	0.03	128.63	143.77	15.14
0.03	57.12	46.65	-10.48	0.03	128.84	143.77	14.93
0.03	56.97	46.65	-10.32	0.03	129.05	143.77	14.72
0.03	56.81	46.65	-10.16	0.03	129.26	143.77	14.50
0.03	56.65	46.65	-10.00	0.03	129.48	143.77	14.29
0.04	56.48	46.65	-9.84	0.04	129.70	143.77	14.07
0.04	56.32	46.65	-9.67	0.04	129.92	143.77	13.84
0.04	56.16	46.65	-9.51	0.04	130.15	143.77	13.62
0.04	55.99	46.65	-9.34	0.04	130.37	143.77	13.40
0.04	55.82	46.65	-9.18	0.04	130.60	143.77	13.17
0.04	55.66	46.65	-9.01	0.04	130.83	143.77	12.94
0.04	55.49	46.65	-8.84	0.04	131.06	143.77	12.71
0.04	55.32	46.65	-8.67	0.04	131.29	143.77	12.48
0.04	55.15	46.65	-8.50	0.04	131.52	143.77	12.25
0.04	54.98	46.65	-8.33	0.04	131.75	143.77	12.02
0.05	54.81	46.65	-8.16	0.05	131.98	143.77	11.78
0.05	54.64	46.65	-7.99	0.05	132.22	143.77	11.55

0.05	54.47	46.65	-7.82	0.05	132.45	143.77	11.31
0.05	54.30	46.65	-7.65	0.05	132.69	143.77	11.08
0.05	54.13	46.65	-7.48	0.05	132.92	143.77	10.85
0.05	53.96	46.65	-7.32	0.05	133.15	143.77	10.61
0.05	53.79	46.65	-7.15	0.05	133.39	143.77	10.38
0.05	53.63	46.65	-6.98	0.05	133.62	143.77	10.14
0.05	53.46	46.65	-6.81	0.05	133.86	143.77	9.91
0.05	53.29	46.65	-6.64	0.05	134.09	143.77	9.68
0.06	53.13	46.65	-6.48	0.06	134.32	143.77	9.45
0.06	52.96	46.65	-6.31	0.06	134.55	143.77	9.22
0.06	52.80	46.65	-6.15	0.06	134.78	143.77	8.99
0.06	52.63	46.65	-5.99	0.06	135.01	143.77	8.76
0.06	52.47	46.65	-5.82	0.06	135.24	143.77	8.53
0.06	52.31	46.65	-5.66	0.06	135.46	143.77	8.30
0.06	52.15	46.65	-5.50	0.06	135.69	143.77	8.08
0.06	51.99	46.65	-5.35	0.06	135.91	143.77	7.85
0.06	51.84	46.65	-5.19	0.06	136.13	143.77	7.63
0.06	51.68	46.65	-5.03	0.06	136.35	143.77	7.41
0.07	51.53	46.65	-4.88	0.07	136.57	143.77	7.19
0.07	51.38	46.65	-4.73	0.07	136.79	143.77	6.98
0.07	51.23	46.65	-4.58	0.07	137.00	143.77	6.76
0.07	51.08	46.65	-4.43	0.07	137.22	143.77	6.55
0.07	50.93	46.65	-4.28	0.07	137.43	143.77	6.34
0.07	50.78	46.65	-4.14	0.07	137.64	143.77	6.13
0.07	50.64	46.65	-3.99	0.07	137.84	143.77	5.92
0.07	50.50	46.65	-3.85	0.07	138.05	143.77	5.72
0.07	50.36	46.65	-3.71	0.07	138.25	143.77	5.52
0.07	50.22	46.65	-3.57	0.07	138.45	143.77	5.32
0.08	50.08	46.65	-3.44	0.08	138.65	143.77	5.12
0.08	49.95	46.65	-3.30	0.08	138.84	143.77	4.92
0.08	49.82	46.65	-3.17	0.08	139.03	143.77	4.73
0.08	49.69	46.65	-3.04	0.08	139.22	143.77	4.54
0.08	49.56	46.65	-2.91	0.08	139.41	143.77	4.36
0.08	49.43	46.65	-2.79	0.08	139.60	143.77	4.17
0.08	49.31	46.65	-2.66	0.08	139.78	143.77	3.99
0.08	49.19	46.65	-2.54	0.08	139.96	143.77	3.81
0.08	49.07	46.65	-2.42	0.08	140.13	143.77	3.63
0.08	48.95	46.65	-2.30	0.08	140.31	143.77	3.46
0.09	48.83	46.65	-2.19	0.09	140.48	143.77	3.29
0.09	48.72	46.65	-2.07	0.09	140.65	143.77	3.12
0.09	48.61	46.65	-1.96	0.09	140.81	143.77	2.96
0.09	48.50	46.65	-1.85	0.09	140.97	143.77	2.79
0.09	48.39	46.65	-1.75	0.09	141.13	143.77	2.63

0.09	48.29	46.65	-1.64	0.09	141.29	143.77	2.48
0.09	48.18	46.65	-1.54	0.09	141.44	143.77	2.32
0.09	48.08	46.65	-1.44	0.09	141.59	143.77	2.17
0.09	47.99	46.65	-1.34	0.09	141.74	143.77	2.03
0.09	47.89	46.65	-1.24	0.09	141.89	143.77	1.88
0.10	47.79	46.65	-1.15	0.10	142.03	143.77	1.74
0.10	47.70	46.65	-1.06	0.10	142.17	143.77	1.60
0.10	47.61	46.65	-0.97	0.10	142.30	143.77	1.46
0.10	47.52	46.65	-0.88	0.10	142.44	143.77	1.33
0.10	47.44	46.65	-0.79	0.10	142.57	143.77	1.20

Attachment O : Gantt Chart

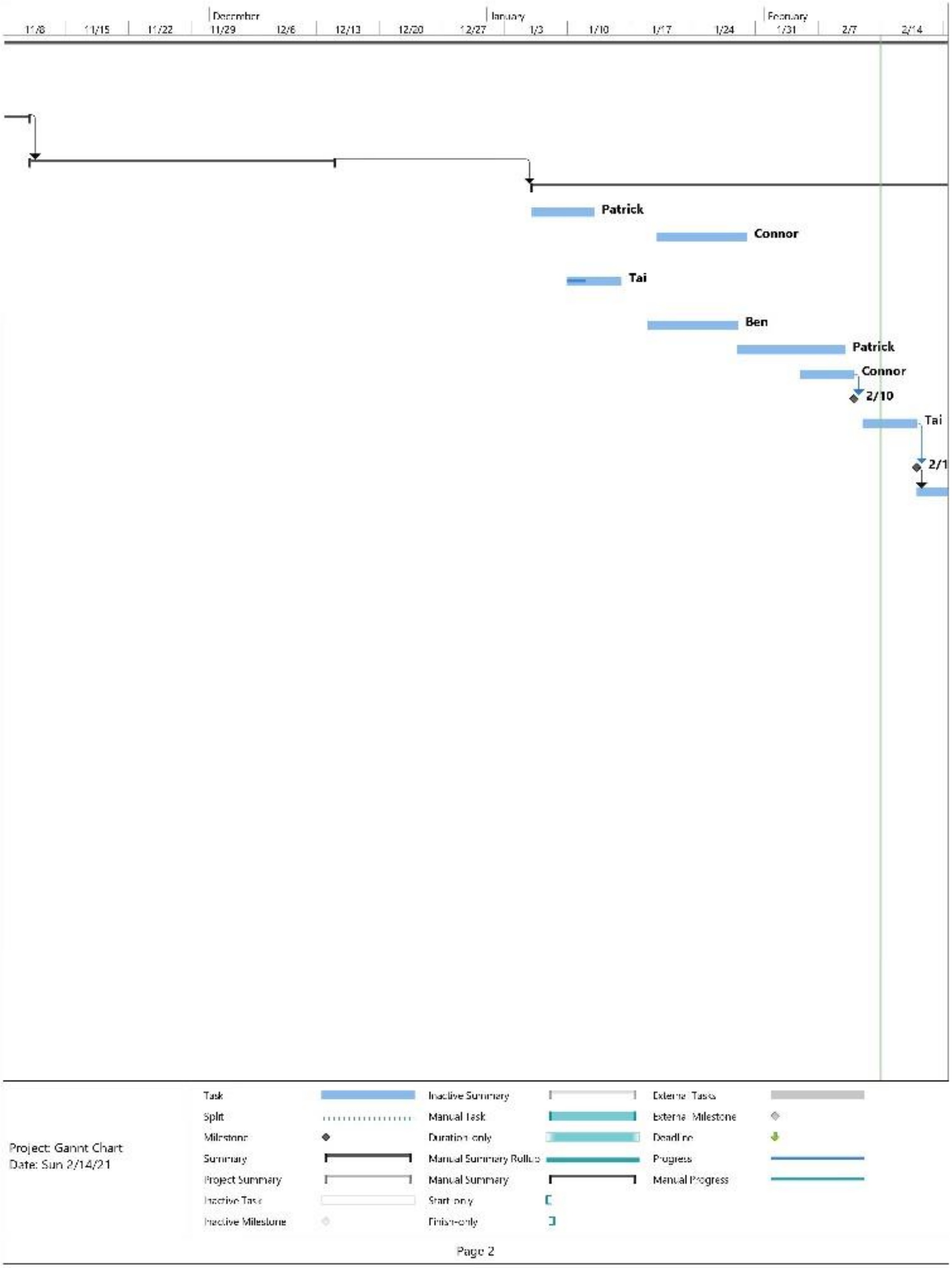


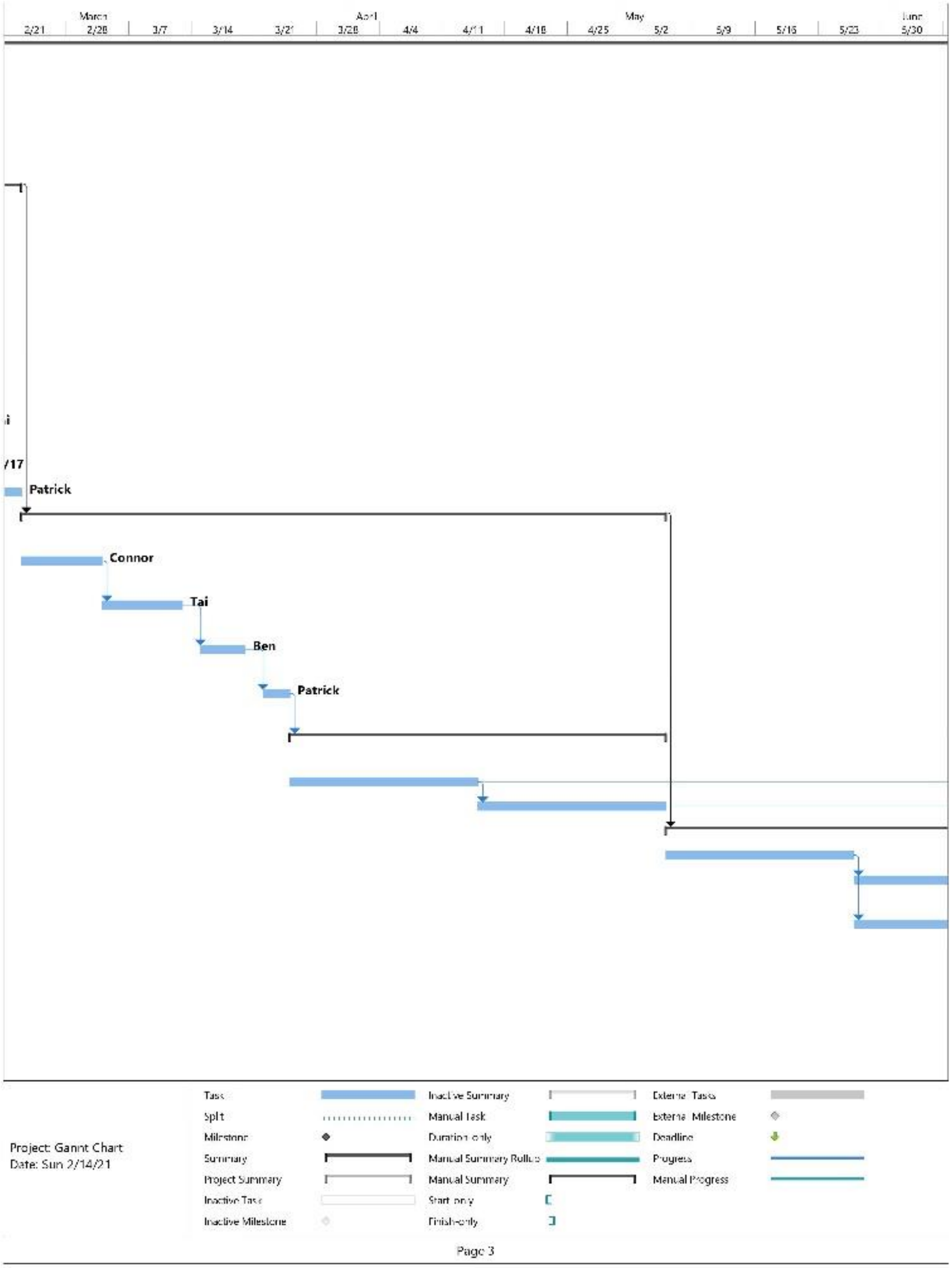
ID		Task Mode	Task Name
1			**Senior Project**
2			**Problem Statement**
5			**SOW**
10			**Inverse Kinematics and System Modeling**
20			**Controls System Design**
29			**Control System Assembly**
30			Control Loop Overview
31			Kinematics as a Matlab Function
32			Inverse Kinematics as Function File
33			Lagrangian Model
34			Multibody Design
35			CDR Presentation Creation
36			Presenting CDR
37			Updates to Report to include CDR
38			Report Check for CDR
39			Assembling pieces into a loop
40			**Analysis and Verification of Parts**
41			Using animations to verify Lagrangian
42			Using animations to verify Newtonian
43			Sample Motions for MultiBody Leg
44			Compare Run Times for Various Comonents
45			**Finalazing Control System and Pysical Modelling**
46			Finishing Prototype Model
47			Complete Control System
48			**Documentation Preperation**
49			Collecting information
50			Assembling instruction manual
51			Assembling instructinoal video
52			**FDR**
53			Finishing Testing
54			Finishing FDR
55			Final meeting with sponsors
56			EXPO Day

Project: Gannt Chart
Date: Sun 2/14/21

Task	▬	Inactive Summary	⌐¬	External Tasks	▬
Split	┈┈┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⬇
Summary	⌐¬	Manual Summary Rollup	▬	Progress	─
Project Summary	⌐¬	Manual Summary	⌐¬	Manual Progress	─
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		

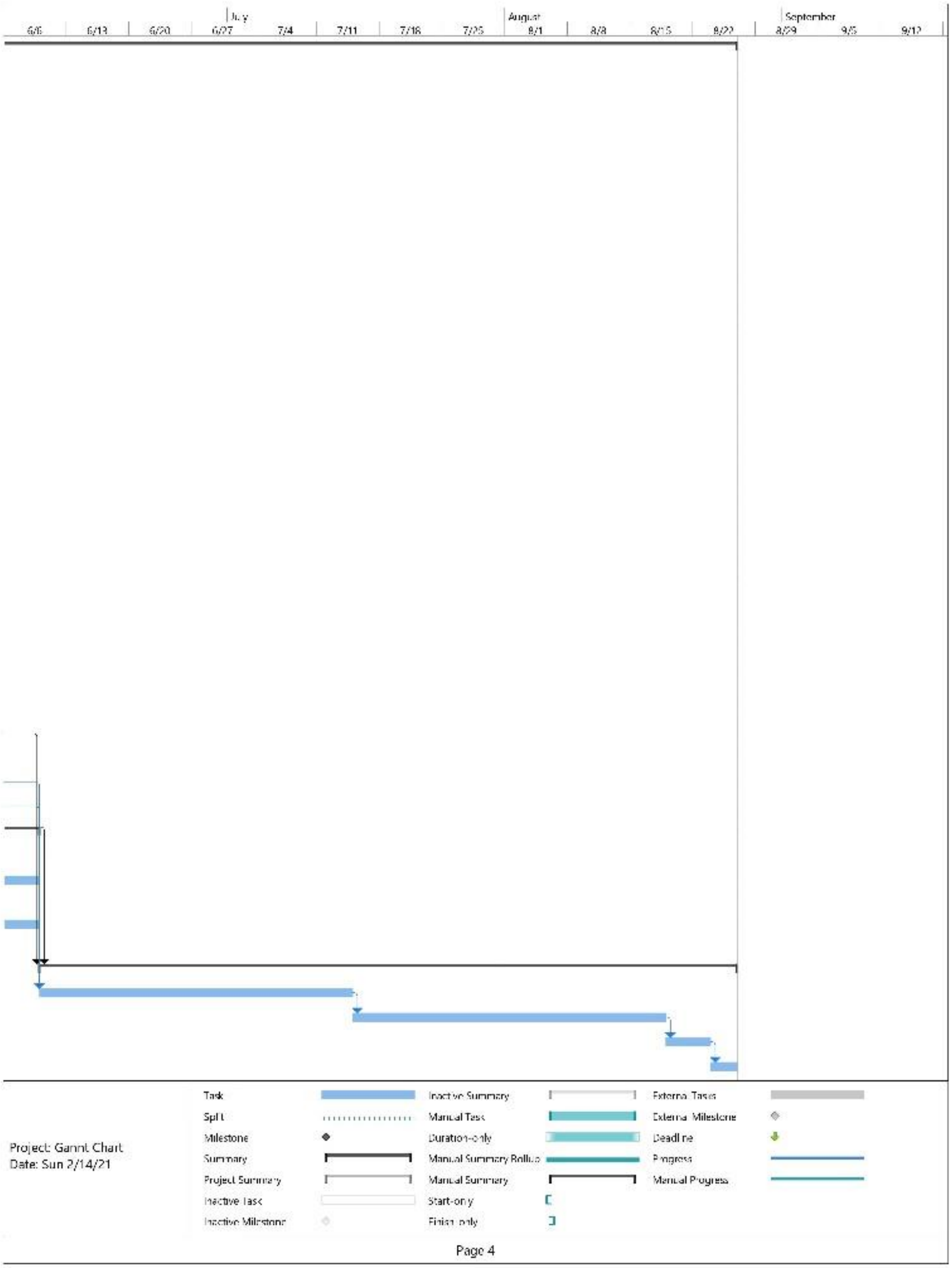
Page 1


December
January
February
11/8
11/15
11/22
11/29
12/6
12/13
12/20
12/27
1/3
1/10
1/17
1/24
1/31
2/7
2/14
Patrick
Connor
Tai
Ben
Patrick
Connor
2/10
Tai
2/1
Project: Gannt Chart
Date: Sun 2/14/21
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 2



Project: Gantt Chart
Date: Sun 2/14/21

Page 3


July
August
September
6/6
6/13
6/20
6/27
7/4
7/11
7/18
7/25
8/1
8/8
8/15
8/22
8/29
9/5
9/12
Project: Gannt Chart
Date: Sun 2/14/21
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 4