2016

Cascade Control of a Hydraulic Prosthetic Knee

Xin Hui
Cleveland State University

Cascade Control of a Hydraulic Prosthetic Knee

XIN HUI

Bachelor of Science in Mechanical Engineering

Shenyang Ligong University, Shenyang, China

July, 2010

submitted in partial fulfillment of requirements for the degree

MASTERS OF SCIENCE IN MECHANICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

May, 2016

We hereby approve this thesis for

Xin Hui

Candidate for the Master of Science in Mechanical Engineering
degree for the Department of Mechanical Engineering
and the CLEVELAND STATE UNIVERSITY
College of Graduate Studies

Thesis Chairperson, Hanz Richter, Ph.D.

Department & Date

Daniel Simon, Ph.D.

Department & Date

Antonie van den Bogert, Ph.D.

Department & Date
Student's Date of Defense: March 29, 2016

ACKNOWLEDGMENTS

This research was partially supported by the State of Ohio, Department of Development and Third Frontier Commission, which provided funding through the Cleveland Clinic Foundation in support of the project Rapid Rehabilitation and Return to Function for Amputee Soldiers. This work was also partially supported by the National Science Foundation under grant number 0826124.

I would like to express my sincere appreciation to Dr. Hanz Richter, my advisor, for all of his help and guidance throughout this project. I truly appreciate all of the valuable time that he has spent on this project, this work would not have been possible without him. I would also like to express my gratitude to Dr. Daniel Simon and Dr. Antonie van den Bogert for letting me be a part of the research group and serving as my committee members.

I would like to thank my fiancee, Xiaoxu Li, for all her encouragement, patience and love, I am truly grateful. I would also like acknowledge my good friend Andrew Hess for the writing correction. Finally, I would like to thank my parents who always give me comfort and support. I appreciate all you have given, without you, I never would have made it through.

Cascade Control of a Hydraulic Prosthetic Knee

XIN HUI

ABSTRACT

A leg prosthesis test robot with hydraulic knee actuator is modeled and tested with closed loop control simulation. A cascade control architecture is designed for the system, the outer loop is controlled by a robust passivity-based controller (RPBC) and the inner loop is controlled by an optimization method. The control algorithm provides knee angle tracking with an RMS error of 0.07 degrees. The research contributes to the field of prosthetics by showing that it is possible to find effective closed loop control algorithm for a newly proposed hydraulic knee prosthesis. The simulations demonstrate the efficiency of RPBC's ability to control complex, nonlinear and multivariable system with plant variability and parameter uncertainty. Dynamic equations for the hydraulic knee actuator are derived from bond graph, an optimization method is used to solve the inversion problem. Low-pass filters are implemented to eliminate signal chatter. Necessary modifications of knee actuator parameters are discussed and recommended to achieve better tracking performance.

TABLE OF CONTENTS

ABSTRACT v

LIST OF FIGURES viii

LIST OF TABLES xi

I Introduction 1

1.1 Transfemoral Prostheses 2

1.2 Prosthesis Testing 2

1.3 Robot Control 4

1.4 Summary of Contributions 5

1.5 Scope of Thesis 6

II Robust Passivity-Based Control Design 8

2.1 Introduction 8

2.2 Robust Passivity-Based Control Theory 9

2.3 RPBC For 2-Link Planar Manipulator 13

2.4 RPBC for CSU Robot 21

III Hydraulic Knee Actuator 36

3.1 Hydraulic Knee Actuator 36

3.2 Bond Graph of Hydraulic Knee Actuator 39

3.3 Linear Cylinder Equations 42

3.4 Valve Positions by Approximate Inversion 44

IV Cascade Control of Prosthesis Test Robot 49

4.1 Cascade Control Architecture 49

4.2 Cascade Control Simulation with Original Model 51

4.3 Cascade Control Simulation with Modified Model 52

4.3.1 Simulation with Modified Model 52

4.3.2 Simulation with Low-Pass Filter 63

4.4 Conclusion 64

V Conclusions and Recommendations 72

5.1 Conclusions 72

5.2 Recommendations for Future Work 73

BIBLIOGRAPHY 74

A MATLAB PROGRAMS 77

LIST OF FIGURES

1.1 Machine Schematic 4

1.2 Prosthesis Testing Robot 5

2.1 2-Link Planar Manipulator 14

2.2 Diagram of Two-Link Planar Manipulator under RPBC in Simulink 28

2.3 Desired and Actual Value of q1 vs. Time 29

2.4 Desired and Actual Value of q2 vs. Time 29

2.5 Robot Coordinate Frame 30

2.6 RPBC of CSU Robot in Matlab Simulink 31

2.7 Actual and Desired Value of Hip Displacement vs. Time 32

2.8 Actual and Desired Value of Thigh Angle vs. Time 32

2.9 Actual and Desired Value of Knee Angle vs. Time 33

2.10 Ground Reaction Force vs. Time 33

2.11 Control Signal u1 vs. Time 34

2.12 Control Signal u2 vs. Time 34

2.13 Control Signal u3 vs. Time 35

3.1 Hydraulic Schematic 37

3.2 Geometry of Hydraulic Actuator 39

3.3 Bond Graph of Hydraulic Knee Actuator 40

3.4 Sign Convention Interpretation 41

3.5 Matlab Simulink of Valve Controls Optimization 46

3.6 The Actual VS Reference Knee Moment 47

3.7 High Pressure Valve Control Signal VS Time 47

3.8 Low Pressure Valve Control Signal VS Time 48

4.1 Cascade Control Architecture 50

4.2 Diagram of The Cascade Control Model in Simulink 54

4.3 Ground Reaction Force Profile 55

4.4 Hip Displacement Tracking vs. Time with Original Model 55

4.5 Thigh Angle Tracking vs. Time with Original Model 56

4.6 Knee Angle Tracking vs. Time with Original Model 56

4.7 Control Signal u_1 vs. Time with Original Model 57

4.8 Control Signal u_2 vs. Time with Original Model 57

4.9 High Pressure Valve Opening vs. Time with Original Model 58

4.10 Low Pressure Valve Opening vs. Time with Original Model 58

4.11 Hip Displacement Tracking vs. Time in Modified Model 59

4.12 Thigh Angle Tracking vs. Time in Modified Model 59

4.13 Knee Angle Tracking vs. Time in Modified Model 60

4.14 Ground Reaction Force vs. Time in Modified Model 60

4.15 Control Signal u_1 vs. Time in Modified Model 61

4.16 Control Signal u_2 vs. Time in Modified Model 61

4.17 Low Pressure Valve Opening vs. Time in Modified Model 62

4.18 Low Pressure Valve Opening vs. Time in Modified Model 62

4.19 Diagram of The Cascade Control Model with Filters in Simulink 65

4.20 Hip Displacement Tracking vs. Time in Modified Model with Filter 66

4.21 Thigh Angle Tracking vs. Time in Modified Model with Filter 66

4.22 Knee Angle Tracking vs. Time in Modified Model with Filter 67

4.23 Ground Reaction Force vs. Time in Modified Model with Filter 67

4.24 Control Signal u_1 vs. Time in Modified Model with Filter 68

4.25 Control Signal u_2 vs. Time in Modified Model with Filter 68

4.26 Low Pressure Valve Opening vs. Time in Modified Model with Filter 69

4.27 Low Pressure Valve Opening vs. Time in Modified Model with Filter 69

4.28 High Pressure Valve Signal FFT Magnitude vs. Frequency with Filter 70

4.29 High Pressure Valve Signal FFT Magnitude vs. Frequency without Filter 70

4.30 Low Pressure Valve Signal FFT Magnitude vs. Frequency with Filter 71

4.31 Low Pressure Valve Signal FFT Magnitude vs. Frequency without Filter 71

LIST OF TABLES

2.1 Parameters of 2-link Planar Manipulator 21

2.2 Parameters of CSU Robot 26

3.1 Parameters of Hydraulic Knee Actuator 38

4.1 Parameters of Cascade Control Simulation with Original Model 51

4.2 Parameters of Cascade Control Simulation with Modified Model 53

4.3 Comparison of RMS 63

4.4 Chattering of Valve Signals 64

CHAPTER I

Introduction

Prosthesis test robots are becoming popular in testing novel transfemoral prostheses due to their high repeatability, safety and pervasive sensing ability. Robots are complex nonlinear and multivariable systems, as a consequence robust passivity based control is an excellent candidate to be utilize as a control algorithm for these systems. In this project, a transfemoral prosthesis with hydraulic knee actuator is attached to a testing robot. A cascade control architecture is used where the outer loop can be controlled by robust passivity-based controller and the inner loop can be controlled by other means, including optimization methods. The thesis establishes practical procedures for designing the robust-passivity based controller and inner-loop controller and applies them to this prosthesis test robot simulation model.

In this chapter, the literature about transfemoral prostheses, prosthesis testing and robot control is reviewed and the scope of the thesis is presented at the end.

1.1 Transfemoral Prostheses

For a transfemoral amputee, approximately 30-50 percent more energy is used during walking in comparison to an able-bodied person. This is closely associated with the complexities in movement of the knee [1]. As the literature shows, modern transfemoral prosthetic mechanisms such as active [2], passive [3], semi-active [4], ankle-knee [5], hydraulic [6] and electromechanical[7] are employed to improve walking performance. The use of microprocessor has resulted in significant advances in this field. Some examples are Otto Bock's C-Leg [8] and Ossur's Rheo knee [9], microprocessors are used to interpret and analyze signals from knee angle sensors and moment sensors to determine the type of motion being employed by the amputee and at the same time, to generate the signals to control the resistance of the prosthetic knee. However, even the most modern and technically advanced transfemoral leg prostheses still cannot fully restore normal gait and can not save much in terms of metabolic cost [10]. Therefore, many research efforts are aimed at developing better prostheses.

A hydraulic prosthetic knee concept [11] was proposed by a Cleveland Clinic research group which allows energy storage during periods of negative work. This rotary hydraulic actuator consists of a cylinder, a spring-loaded accumulator and two valves that can be used to control knee motions and regulate energy storage and return. In this thesis, the prosthetic knee is implemented as a part of the prosthesis test robot and the main focus is the optimization of the two valves' control signals.

1.2 Prosthesis Testing

With the rapid development of novel prosthetic knees, prosthesis testing becomes a necessary requirement to verify new concepts prior to their application to patients. Human gait trial is one of the most common testing approaches, however, there could be systematic errors and inaccuracies with this method. For example, the use of safety harnesses affects useful data collection, and human gait trial tests are not highly repeatable which is extremely important for data analysis. In comparison, robotic testing of prosthesis can eliminate those disadvantages, while at the same time bring additional features. Robots own a characteristic of high test repeatability and can run continuously for a long time which is necessary for real-time optimization of prosthetic control algorithms. Furthermore, safety clearances associated with human subject testing can be eliminated using a robotic test system. Simultaneously, sensors can be pervasively attached to the robot, so that forces and moments are captured directly together with the motions of each joint which are available for further calculations and evaluations.

This kind of prosthesis testing device has been investigated by the Fraunhofer Institute [12], Cleveland Clinic [13] and Cleveland State University. In this paper we employ the CSU robot [14] in combination with the hydraulic transfemoral prosthesis for testing the capability of this novel prosthesis during human gait cycle. The CSU robot has two degrees of freedom, namely hip displacement and thigh swing. It can imitate the motions of a human hip during walking and running. Normal gait data collected from able bodied persons by the Cleveland Clinic gait lab (Cleveland Clinic, Cleveland, Ohio) [15] is used as a profile for robot control. For simulating not only the swing phase but also the stance phase, a treadmill is used as a walking surface. Since we attach the hydromechanical knee on it, the system is equivalent to a 3-d.o.f. robot with a prismatic-revolute-revolute (PRR) configuration. Vertical motion of the carriage is produced by a ballscrew with a direct-drive brushless DC servomotor, thigh rotation is generated by an inchworm-gear reducer driven by a direct-drive brushless DC servomotor, while the knee rotation is achieved by a hydraulic actuator. Fig.1.1 shows the machine schematic and its components, the overall testing robot is shown in the photograph of Fig.1.2.

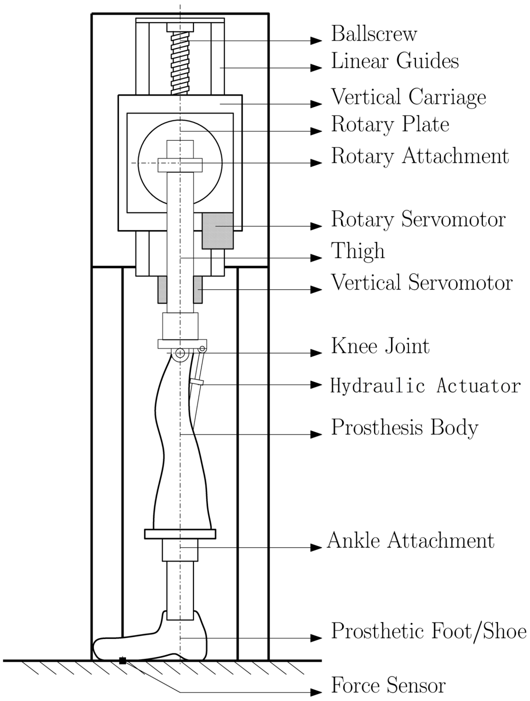

Figure 1.1: Machine Schematic

1.3 Robot Control

A robot controller determines the joint inputs required at each instant in time to make the robot tracks a commanded motion. This could be a sequence of end-effector positions and orientations or a continuous path [16]. Various control methodologies are available for robot control, including proportional-integral-derivative (PID) control [17], passivity-based control [18], robust control [19], adaptive control [20], etc. The particular control method used may cause significantly different performance. Therefore, each method has its own range of applications. For example, PID control, the most common type of control algorithm in industry only works for independent-joint setpoint regulation problems.

In the early days, a robot control system was considered as single input/single output linear system with each axis (joint) controlled independently [21].Coupling effects from the motion of other links were regarded as disturbances. However, most robots are complex nonlinear and mul-

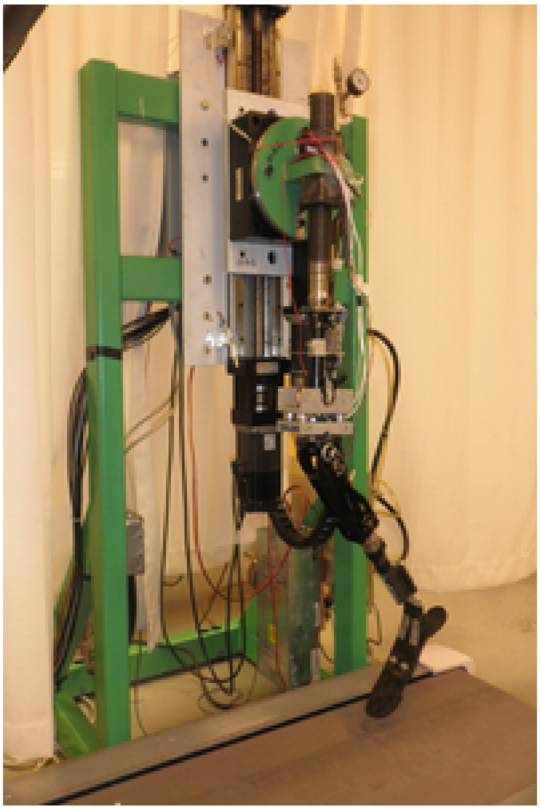

Figure 1.2: Prosthesis Testing Robot

tivariable systems and uncertainty exists in the parameters defining manipulator dynamics. Robust control and adaptive control are regarded as effective ways to deal with parametric uncertainty. By exploiting the skew-symmetry property of the robot inertia and Coriolis matrices, passivity-based control is implemented widely in robot control research. In this thesis, robust passivity-based control is employed for the prosthesis test robot.

1.4 Summary of Contributions

This project carries out modeling, control design and simulation tasks for a hydraulic knee actuator in an organized way. The prosthetic knee is integrated into a test robot, and this is reflected in the mathematical model of the overall system. A bond graph model of the hydraulic knee has been developed, along with corresponding Simulink code for integration with the test robot simulation.

Open-loop control of the hydraulic knee (and similar design [11]) has been considered before with optimization methods. To the extent of the author's knowledge, this is the first known working feedback controller for this type of hydraulic knee actuator.

The controller proposed in this thesis is of the *cascade* type: the prosthesis is regarded as an additional robotic link with direct torque control. A control moment demand signal is generated by a robust passivity-based controller, which is suitable for general robotic systems. An online optimizer then finds the combination of valve positions that minimizes the difference between demanded moment and actual moment. Computed valve positions are then applied to the hydraulic knee actuator.

The control algorithm was evaluated for its ability to track a knee angle reference signal, and the amount of valve chattering was quantified. The best results provide knee angle tracking with an RMS error of 0.07 degrees. To achieve these results, some changes to the design parameters had to be made, as detailed in Chapter 4.

This research contributes to the field of prosthetics by showing that it is possible to find effective closed loop control signals for hydraulic knee prostheses of the type considered. As discussed in Chapter 5, several limitations remain, including difficulties with real-time implementation of the online optimizer and parameter changes introduced to the original design.

1.5 Scope of Thesis

This thesis is composed of five chapters. Chapter 2 describes the application of robust passivity-based control (RPBC) in the CSU robot, where the theory of RPBC is reviewed, along with an example of RPBC using in 2-link planar manipulator, a dynamic model for the CSU robot and the simulation results. Chapter 3 presents the operating principle of a hydraulic knee actuator and its mathematical model developed in bond graph form. The inversion problem and a possible optimization concept are also discussed. In Chapter 4, the cascade control architecture is introduced and tested. Note that RPBC is used independently by assuming a perfect knee actuator is attached, then an inversion of the moment equations is attempted, to generate a set of valve inputs that result in the knee moment demanded by RPBC. It shows all simulation results including the tracking performance of each joint and the control signals. It also describes how the the best tunings are found. Chapter 5 presents conclusions and recommendations for future work.

CHAPTER II

Robust Passivity-Based Control Design

2.1 Introduction

The dynamic equations of robot manipulators constitute complex, nonlinear and multivariable systems. One of the first methods of controlling these systems was inverse dynamics which is also known as a special case of the method of feedback linearization. It relies on cancellation of nonlinearities in the system dynamics[16]. However, plant variability and uncertainty are obstacles to an exact dynamic inversion. Inverse dynamics control therefore has limited practical validity. To overcome this difficulty, motion control techniques based on the passivity property of the Euler-Lagrange equations are considered. Especially for the robust and adaptive control problems, the passivity-based approach shows significant advantages over the inverse dynamic method. Therefore, robust passivity-based control (RBPC) gained attention as a powerful nonlinear control law that can guarantee stability and tracking of arbitrary trajectories efficiently despite uncertainties in plant model parameters.

RBPC theory is outlined at the beginning of this chapter, followed by the implementations of

RBPC in a 2-link planar manipulator and the CCF Robot, where direct torque control at the knee is assumed.

2.2 Robust Passivity-Based Control Theory

Considering an n-link rigid robotic manipulator, application of the Euler-Lagrange equations leads to:

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=\tau \tag{2.1}$$

where $q(t)\in R^n$ denotes the generalized coordinates; $\tau(t)\in R^n$ represents the joint vector inputs (forces or torques); $D(q)\in R^{n\times n}$ is the inertia matrix; $C(q,\dot{q})\dot{q}\in R^n$ is the vector of Coriolis and centrifugal torques, and $g(q)\in R^n$ is the vector of gravity torques. The dynamic equations relate torque to position, velocity and acceleration.

This Euler-Lagrange equation can be derived by modeling the manipulator in joint space where the kinetic energy is given by $K(q,\dot{q})=\frac{1}{2}\dot{q}^TD(q)\dot{q}$,and the potential energy $P(q)$ is independent of $\dot{q}$. Thus

$$L=K-P=\frac{1}{2}\sum_{i,j}d_{i,j}(q)\dot{q}_i\dot{q}_j-P(q) \tag{2.2}$$

The standard robot dynamic equation has several important structural properties that can be used to develop robust or adaptive nonlinear control algorithms.

Property 1: The matrix $N(q,\dot{q})=\dot{D}(q)-2C(q,\dot{q})$ is skew-symmetric. Note that, $D(q)$ is the inertia matrix for an n-link robot and $C(q,\dot{q})$ is defined in terms of the elements of $D(q)$ according to Eq. (2.29).

Property 2: The amount of energy spent by the system is lower-bounded by a constant $-\beta$ $(\beta\geq 0)$ which is the so-called passivity property:

$$\int_0^T \dot{q}^T(\varsigma)\tau(\varsigma)d\varsigma\geq-\beta,\forall T>0 \tag{2.3}$$

Property 3: The dynamics of (2.1) is linearly parameterizable:

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=Y(q,\dot{q},\ddot{q})\Theta=\tau \tag{2.4}$$

where $Y(q,\dot{q},\ddot{q})$, an $n\times l$ matrix function, is called the regressor and Θ is an l-dimensional parameter vector.

Property 4: The inertia matrix $D(q)$ is symmetric positive definite and bounded as

$$\lambda_1(q)I_{n\times n}\leq D(q)\leq\lambda_n(q)I_{n\times n} \tag{2.5}$$

where $0<\lambda_1(q)\leq\ldots\leq\lambda_n(q)$ denote the n eigenvalues of $D(q)$ for any fixed coordinate q. The functions λ_1 and λ_n can be chosen as positive constants while the the robot contains only revolute joints.

Based on the properties of skew symmetry and linearity in the parameters, the robust passivity based control (RBPC) is developed as follows. Consider a control input of the form

$$u=\hat{M}(q)a+\hat{C}v+\hat{g}-Kr \tag{2.6}$$

where the quantities v , a ,and r are given as

$$v=\dot{q}^d-\Lambda\tilde{q}$$

$$a=\dot{v}=\ddot{q}^d-\Lambda\dot{\tilde{q}}$$

$$r=\dot{q}-v=\dot{\tilde{q}}+\Lambda\tilde{q}$$

where K and Λ are diagonal matrices of positive gains. Note that the notation $\hat{(.)}$ denotes the nominal value of $(.)$ and $\tilde{(.)}=(.)-\hat{(.)}$ represents the error of the system parameters.

By applying the linear parameterization property to the robot dynamics, the control is expressed as

$$u = Y(q, \dot{q}, a, v)\hat{\Theta} - Kr \tag{2.7}$$

Combining the Eq. (2.6) with Eq. (2.1), we obtain

$$M(q)\dot{r} + C(q, \dot{q})r + Kr = Y(\hat{\Theta} - \Theta) \tag{2.8}$$

Now choose $\hat{\Theta} = \Theta_0 + \delta\Theta$, where Θ_0 is the best estimated parameter vector, and $\delta\Theta$ is an additional control term. Then system (2.8) becomes

$$M(q)\dot{r} + C(q, \dot{q})r + Kr = Y(\tilde{\Theta} + \delta\Theta) \tag{2.9}$$

where $\tilde{\Theta} = \Theta_0 - \Theta$ is the parametric uncertainty in the system. Suppose the uncertainty is bounded in norm by a nonnegative constant ρ such that

$$\left\|\tilde{\Theta}\right\| = \|\Theta - \Theta_0\| \leq \rho \tag{2.10}$$

then we can design the additional control term $\delta\Theta$ according to

$$\delta\Theta = \begin{cases} -\rho\frac{Y^T r}{\|Y^T r\|}; & \text{if} \quad Y^T r \neq 0 \\ 0 & \text{if} \quad Y^T r = 0 \end{cases} \tag{2.11}$$

To prove the stability and uniform ultimate boundedness of the tracking errors, the following Lyapunov function is considered.

$$V = \frac{1}{2}r^T M(q)r + \tilde{q}^T \Lambda K \tilde{q} \tag{2.12}$$

Calculating $\dot{V}$ yields

$$\dot{V} = r^T M\dot{r} + \frac{1}{2}r^T \dot{M}r + 2\tilde{q}^T \Lambda K\dot{\tilde{q}} \tag{2.13}$$

Substitute Eq. (2.9) into Eq. (2.13)

$$\dot{V} = r^T M(M^{-1}Y(\tilde{\Theta}+\delta\Theta) - M^{-1}Cr - M^{-1}kr) + \frac{1}{2}r^T \dot{M}r + 2\tilde{q}^T \Lambda K\dot{\tilde{q}}$$

$$\dot{V} = -r^T kr - r^T Y(\tilde{\Theta}+\delta\Theta) + \frac{1}{2}r^T(\dot{M}-2C)r + 2\tilde{q}^T \Lambda K\dot{\tilde{q}} \tag{2.14}$$

the $\frac{1}{2}r^T(\dot{M}-2C)r$ term can be eliminated due to the skew-symmetry property, then substitute the definition of r into the equation, we derive

$$\dot{V} = -e^T Qe + r^T Y(\tilde{\Theta}+\delta\Theta) \tag{2.15}$$

where

$$Q = \begin{bmatrix} \Lambda^T K\Lambda & 0 \\ 0 & K \end{bmatrix}$$

and

$$e = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} q - q^d \\ \dot{q} - \dot{q}^d \end{bmatrix}$$

the term $-e^T Qe$ is negative definite, according to the Eq. (2.11), if $Y^T r = 0$, then

$$\dot{V} = -e^T Qe < 0; \tag{2.16}$$

if $Y^T r \neq 0$, then

$$\dot{V} = -e^T Qe + r^T Y(\tilde{\Theta} - \rho\frac{Y^T r}{\left\|Y^T r\right\|}) = -e^T Qe + r^T Y\tilde{\Theta} - \rho\left\|Y^T r\right\| \tag{2.17}$$

However, because of $\left|r^T Y\tilde{\Theta}\right| \leq \left\|r^T Y\right\|\left\|\tilde{\Theta}\right\|$, we can get $\left|r^T Y\tilde{\Theta}\right| \leq \left\|r^T Y\right\|\rho$. If $r^T Y\tilde{\Theta} \geq 0$, then $\left|r^T Y\tilde{\Theta}\right| = r^T Y\tilde{\Theta} \leq \left\|r^T Y\right\|\rho$, so $r^T Y\tilde{\Theta} - \rho\left\|Y^T r\right\| \leq 0$ and $\dot{V} < 0$. On the other hand, if $r^T Y\tilde{\Theta} < 0$, then $r^T Y\tilde{\Theta} - \rho\left\|Y^T r\right\| < 0$ and $\dot{V} < 0$.

So we conclude that the tracking error is uniformly ultimately bounded under the control $\delta\Theta$ from Eq. (2.11). However, the control $\delta\Theta$ is discontinuous on the subspace of $Y^T r = 0$ which leads to chattering problems in practice where the control switches rapidly between the control value in Eq. (2.11). To eliminate chattering, a continuous control can be designed according to

$$\delta\Theta = \begin{cases} -\rho\frac{Y^T r}{\left\|Y^T r\right\|}; & \text{if} \quad \left\|Y^T r\right\| > \varepsilon \\ -\frac{\rho}{\varepsilon} Y^T r & \text{if} \quad \left\|Y^T r\right\| \leq \varepsilon \end{cases} \tag{2.18}$$

where the constant ε is deadzone which can be chosen as large as necessary to eliminate chattering.

In order to demonstrate the effectiveness of RPBC, we apply it to a trajectory tracking control of a 2-link planar manipulator in the next section.

2.3 RPBC For 2-Link Planar Manipulator

Consider the two-link planar arm with two revolute joints shown in Fig.2.1. We establish the base frame $o_0x_0y_0z_0$ as shown. The joint axes z_0, z_1, z_2 are pointing out of the page. To fix the notation, we set $i = 1, 2$, q_i represents the joint angle, which is also known as a generalized coordinate; l_i represents the length of link i; l_{ci} represents the distance from the center of mass of link i to the previous joint; m_i denotes the mass of link i; and I_i denotes the moment of inertia about an axis through the center of mass of link i parallel to the z_i-axis.

The transformation matrices by following DH(Denavit-Hartenberg) convention [16] are de-

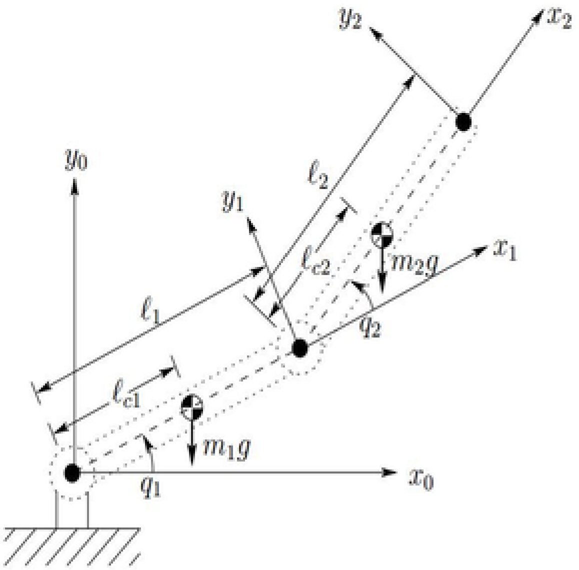

Figure 2.1: 2-Link Planar Manipulator

rived as

$$A_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1c_1 \\ s_1 & c_1 & 0 & l_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^0 = A_1^0A_2^1 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1c_1 + l_2c_{12} \\ s_{12} & c_{12} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $s_1 = \sin(q_1)$, $s_2 = \sin(q_2)$, $c_1 = \cos(q_1)$, $c_2 = \cos(q_2)$, $s_{12} = \sin(q_1 + q_2)$ and $c_{12} = \cos(q_1 + q_2)$.

Notice that the first three rows of the first three columns of A_1^0 and A_2^0 represent the rotation matrices R_1^0 and R_2^0. Moreover, the first three entries of the last column of A_1^0 and A_2^0 describe the origin o_1 and o_2 in the base frame.

Since both two joints are revolute, the Jacobian matrix to the centers of mass are 6×2 matrices with the form

$$J(q) = \begin{bmatrix} z_0 \times (o_c - o_0) & z_1 \times (o_c - o_1) \\ z_0 & z_1 \end{bmatrix} \tag{2.19}$$

The various quantities above can be found as

$$o_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, o_1 = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix}, z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The Jacobian matrix is derived after the required calculations

$$J_1 = \begin{bmatrix} -l_{c1} s_1 & 0 \\ l_{c1} c_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} -l_{c2}s_{12} - l_1 s_1 & -l_{c2}s_{12} \\ l_{c2}c_{12} + l_1 c_1 & l_{c2}c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$v_{c1} = J_{v_{c1}}\dot{q} \tag{2.20}$$

where,

$$J_{v_{c1}} = \begin{bmatrix} -l_{c1}s_1 & 0 \\ l_{c1}c_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.21}$$

Similarly,

$$v_{c2} = J_{v_{c2}}\dot{q} \tag{2.22}$$

where

$$J_{v_{c2}} = \begin{bmatrix} -l_{c2}s_{12} - l_1 s_1 & -l_{c2}s_{12} \\ l_{c2}c_{12} + l_1 c_1 & l_{c2}c_{12} \\ 0 & 0 \end{bmatrix} \tag{2.23}$$

So the linear velocity term of the kinetic energy is

$$\frac{1}{2}m_1 v_{c1}^T v_{c1} + \frac{1}{2}m_2 v_{c2}^T v_{c2} = \frac{1}{2}\dot{q}^T (m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}})\dot{q} \tag{2.24}$$

For angular velocity terms.

$$\omega_1 = \dot{q}_1 k, \qquad \omega_2 = (\dot{q}_1 + \dot{q}_2)k \tag{2.25}$$

Since w_i is parallel to the z-axes of each joint coordinate frame, the rotational kinetic energy is $I_i w_i^2$. So the overall rotational kinetic energy in terms of the generalized coordinates is

$$\frac{1}{2}\dot{q}^T \{ I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \}\dot{q} \tag{2.26}$$

Add the two matrices in Eq. (2.24) and Eq. (2.26) to obtain the inertia matrix

$$D(q) = m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} + \begin{bmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \tag{2.27}$$

Applying the standard trigonometric identities to compute the elements of $D(q)$

$$\begin{aligned} d_{11} &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2 \\ d_{12} &= d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 \\ d_{22} &= m_2 l_{c2}^2 + I_2 \end{aligned} \tag{2.28}$$

Then computing the Chrostoffel symbols by using

$$c_{ijk} := \frac{1}{2}\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\} \tag{2.29}$$

This leads to

$$
\begin{aligned}
c_{111} &= \frac{1}{2}\frac{\partial d_{11}}{\partial q_1} = 0 \\
c_{121} = c_{211} &= \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = -m_2 l_1 l_{c2} \sin q_2 \\
c_{221} &= \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = -m_2 l_1 l_{c2} \sin q_2 \\
c_{112} &= \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = m_2 l_1 l_{c2} \sin q_2 \\
c_{122} = c_{212} &= \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = 0 \\
c_{222} &= \frac{1}{2}\frac{\partial d_{22}}{\partial q_2} = 0
\end{aligned}
$$

The potential energy for each link is obtained by multipling its mass, gravitational acceleration and the height of its center of mass. Thus

$$
\begin{aligned}
P_1 &= m_1 g l_{c1} \sin q_1 \\
P_2 &= m_2 g (l_1 \sin q_1 + l_{c2} \sin(q_1 + q_2))
\end{aligned}
$$

So the total potential energy is

$$P = P_1 + P_2 = (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2) \tag{2.30}$$

Hence,

$$g_1 = \frac{\partial P}{\partial q_1} = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2) \tag{2.31}$$

$$g_2 = \frac{\partial P}{\partial q_1} = m_2 l_{c2} g \cos(q_1 + q_2) \tag{2.32}$$

Eventually, the Euler-Lagrange equations of the system are obtained as

$$\begin{aligned} & d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{121}\dot{q}_1\dot{q}_2 + c_{211}\dot{q}_2\dot{q}_1 + c_{221}\dot{q}_2{}^2 + g_1 = \tau_1 \\ & d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{112}\dot{q}_1{}^2 + g_2 = \tau_2 \end{aligned} \tag{2.33}$$

Since

$$C_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i = \sum_{i=1}^{n} \frac{1}{2}\left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} + \frac{\partial d_{ij}}{\partial q_k}\right)\dot{q}_i \tag{2.34}$$

In this case, the matrix $C(q,\dot{q})$ is calculated as

$$C = \begin{bmatrix} e\dot{q}_2 & e\dot{q}_1 + e\dot{q}_2 \\ -e\dot{q}_1 & 0 \end{bmatrix} \tag{2.35}$$

where $e = -m_2 l_1 l_{c2} \sin q_2$.

To convert the Euler-Lagrange equations into the form shown as Eq. (2.4) which is represented by regressor and vector parameters, the parameter vector is set up as

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \\ \Theta_5 \end{bmatrix} = \begin{bmatrix} m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + I_1 + I_2 \\ m_2 l_1 l_{c2} \\ m_2 l_{c2}^2 + I_2 \\ m_1 l_{c1} + m_2 l_1 \\ m_2 l_{c2} \end{bmatrix} \tag{2.36}$$

then the inertia and gravitation matrix elements can be written as

$$d_{11} = \Theta_1 + 2\Theta_2 \cos(q_2) \tag{2.37}$$

$$d_{12} = d_{21} = \Theta_3 + \Theta_2 \cos(q_2) \tag{2.38}$$

$$d_{22} = \Theta_3 \tag{2.39}$$

$$g_1 = \Theta_4 gcos(q_1) + \Theta_5 g \cos(q_1 + q_2) \tag{2.40}$$

$$g_2 = \Theta_5 g \cos(q_1 + q_2) \tag{2.41}$$

Substituting these into Eq. (2.33), yields

$$Y(q,\dot{q},\ddot{q}) = \begin{bmatrix} \ddot{q}_1 & \cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) - \sin(q_2)(\dot{q}_1{}^2 + 2\dot{q}_1\dot{q}_2) & \ddot{q}_2 & g\cos(q_1) & g\cos(q_1+q_2) \\ 0 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1{}^2 & \ddot{q}_1 + \ddot{q}_2 & 0 & g\cos(q_1+q_2) \end{bmatrix} \tag{2.42}$$

Since the regressor and parameters are developed, RPBC theory can be implemented to design a robust controller to satisfy performance specifications. As discussed in the previous section, if the parameter uncertainty can be bounded such that

$$\left\|\tilde{\Theta}\right\| = \|\Theta - \Theta_0\| \leq \rho$$

where ρ is an non-negative constant, our RPBC controller will work effectively. While, ρ is able to be computed from setting an uncertainty level for parameters. This is executed by the Matlab command shown in appendix A.

In order to run the 2-link robot with the robust passivity based controller, the parameters of this manipulator are chosen as shown in Table 2.1. In additional, the uncertainty level is determined as 0.3 in this simulation, which means the value of parameters are selected arbitrarily in the range of 30 percent fluctuation from the nominal value, the deadzone of the controller is chosen as 1. The trajectory reference for these two joints are sine waves, the amplitude, frequency, and phase angles are 1, 1, $\pi/2$, and 1, 1, 0 respectively. The controller is adjusted to give a better performance by tuning the controller gains L and K through trial and error. The system is simulated for 20 seconds and the Matlab Simulink diagram and simulation results are shown in Fig. (2.2) through Fig. (2.4).

Parameters	**Values**	**Units**
m_1	1	kg
m_2	0.75	kg
l_1	1	m
l_2	0.8	m
l_{c1}	0.5	m
l_{c2}	0.7	m
I_{1x}	0.001	kgm^2
I_{1y}	0.002	kgm^2
I_{1z}	0.02	kgm^2
I_{2x}	0.001	kgm^2
I_{2y}	0.001	kgm^2
I_{2z}	0.01	kgm^2

Table 2.1: Parameters of 2-link Planar Manipulator

The simulation results shows that the actual outputs perfectly tracked the desired values of q_1 and q_2. Therefore, a robust passivity based controller is very suitable to this two-link planar manipulator system.

2.4 RPBC for CSU Robot

The CSU Robot is a 3-link rigid robot with a prismatic-revolute-revolute (PRR) configuration. It is developed for testing prosthetic legs through producing the same swing stance trajectories as human gait including hip vertical displacement and thigh swing. As shown in Fig. (2.5).

The frame assignments are set up following the standard Denavit-Hartenberg convention. The prismatic joint corresponds to the hip vertical displacement, with coordinate q_1. The rotary joint attached to the hip block and knee joint represent thigh swing and knee angle respectively, with coordinates q_2 and q_3. A rigid foot is attached to the ankle with an angle of 90 degrees, in additional, a load cell is mounted at the bottom of the foot which is located at $[l_{cx} \quad -l_{cy} \quad 0]^T$ in frame 3 coordinate to measure the vertical ground reaction force. It is also considered to be the only point on the foot that contacts the ground which plays important role in force feedback controls. In this section, assuming that an actively-controlled knee is attached to the robot, it becomes a fully-actuated 3-link planar robot. In this case, a robot dynamic model in joint coordinates can be written as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + J_e^T F_e + g(q) = F_a \tag{2.43}$$

where $q = [q_1 \; q_2 \; q_3]^T$ is the vector of joint displacements, $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is Centripetal and Coriolis matrix, J_e is the kinematic Jacobian of the point which has external force. $g(q)$ is the gravity vector and F_a is a vector of combined actuator inputs, where their inertial and frictional effects are considered.

Similarly, the inertia matrix, Centripetal and Coriolis matrix and gravity vector can be found following the same approach as used in the development of 2-link manipulator system in Section 2.3. Hence, $D(q)$ is

$$\begin{aligned}
&D(1,1) = m_1 + m_2 + m_3\\
&D(1,2) = D(2,1) = (c_3\cos(q_2+q_3) + l_2\cos(q_2)) + m_2(c_2\cos(q_2) + l_2\cos(q_2))\\
&D(1,3) = D(3,1) = c_3 m_3 \cos(q_2+q_3)\\
&D(2,2) = I_{2z} + I_{3z} + c_2^2 m_2 + c_3^2 m_3 + l_2^2(m_2+m_3) + 2c_2 l_2 m_2 + 2c_3 l_2 m_3 \cos(q_3)\\
&D(2,3) = D(3,2) = m_3 c_3^2 + l_2 m_3 \cos(q_3) c_3 + I_{3z}\\
&D(3,3) = m_3 c_3^2 + I_{3z}
\end{aligned}$$

$C(q,\dot{q})$ is

$$C(1,1) = 0$$

$$C(1,2) = -\dot{q_2}(l_2m_3 + m_2(c_2 + l_2))\sin(q_2) - c_3m_3(\dot{q_2} + \dot{q_3})\sin(q_2 + q_3)$$

$$C(1,3) = -c_3m_3\sin(q_2 + q_3)(\dot{q_2} + \dot{q_3})$$

$$C(2,1) = 0$$

$$C(2,2) = -c_3l_2m_3\dot{q_3}\sin(q_3)$$

$$C(2,3) = -c_3l_2m_3\sin(q_3)(\dot{q_2} + \dot{q_3})$$

$$C(3,1) = 0$$

$$C(3,2) = c_3l_2m_3\dot{q_2}\sin(q_3)$$

$$C(3,3) = 0$$

$g(q)$ is

$$g(q_1) = -g(m_1 + m_2 + m_3)$$

$$g(q_2) = -c_3gm_3\cos(q_2 + q_3) - g(m_2(c_2 + l_2) + l_2m_3)\cos(q_2)$$

$$g(q_3) = -c_3gm_3\cos(q_2 + q_3)$$

Since the location vector of load cell is known in frame 3, its world frame location can be readily computed using the transformation matrices as

$$Z_{LC} = q_1 - l_{cy}\cos(q_2 + q_3) + (c_3 + l_{cx})\sin(q_2 + q_3) + l_2\sin(q_2) \tag{2.44}$$

The treadmill belt deflection can be calculated by finding the difference between this coordinate and treadmill standoff height to estimate the vertical ground reaction force, F_{GV}. The velocity

Jacobian at the load cell location is derived as

$$J_e(1,1) = 0$$

$$J_e(1,2) = -(c_3 + l_{cx})\sin(q_2 + q_3) + l_{cy}\cos(q_2 + q_3) - l_2\sin(q_2)$$

$$J_e(1,3) = -(c_3 + l_{cx})\sin(q_2 + q_3) + l_{cy}\cos(q_2 + q_3)$$

$$J_e(2,1) = J_e(2,2) = J_e(2,3) = 0$$

$$J_e(3,1) = 1$$

$$J_e(3,2) = (c_3 + l_{cx})\cos(q_2 + q_3) + l_{cy}\sin(q_2 + q_3) + l_2\cos(q_2)$$

$$J_e(3,3) = (c_3 + l_{cx})\cos(q_2 + q_3) + l_{cy}\sin(q_2 + q_3)$$

The horizontal component of foot velocity V_f can be obtained from the velocity Jacobian above, then horizontal friction force can be calculated as follow

$$F_{GH} = -\mu F_{GV} sign(V_f + V_b) \tag{2.45}$$

where μ is the coefficient set as 0.15, and V_b is the treadmill belt speed set as 1.47 m/s.

Thus,

$$F_e = [F_{GH}\ 0\ -F_{GV}]^T \tag{2.46}$$

To convert the Euler-Lagrange equations into the form represented by regressor and vector parameters, the parameter vector is set up as

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \\ \Theta_5 \\ \Theta_6 \\ \Theta_7 \\ \Theta_8 \\ \Theta_9 \\ \Theta_10 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 + m_3 \\ m_3 l_2 + m_2 l_2 + m_2 c_2 \\ c_3 m_3 \\ I_{2z} + I_{3z} + J_m r^2 + c_2^2 m_2 + c_3^2 m_3 + l_2^2 m_2 + l_2^2 m_3 + 2 c_2 l_2 m_2 \\ l_2 m_3 c_3 \\ m_3 c_3^2 + I_{3z} \\ b_1 \\ f \\ m_0 \\ b_2 \end{bmatrix} \tag{2.47}$$

where m_1 is the linearly-moving mass of link 1; m_2 is the total mass of thigh and its relevant blocks; m_3 is the mass below knee; m_0 is is an equivalent inertial mass for rotating components associated with link 1; l_2 is the nominal thigh length; l_3 is the overall length of link 3, from knee joint to the bottom of shoe; c_2 is the distance from the center of mass of link 2 to o_2; c_3 is the distance from knee joint to the center of mass of link 3 including shoe; I_{2z} is the overall inertia of link 2; I_{3z} is the overall inertia of link 3 including shoe; J_m is the inertia of the rotary motor; r is the gear reduction ratio in the rotary actuator; f is the linear damping ratio in link 1; b_1 is the rotary actuator damping ratio and b_2 is the damping ratio at knee joint. The values of these parameters from[14] are listing in Table 2.2.

The first two joints of the robot are driven by servo DC motors with amplifier gains of $k_1 = 375N/V$ and $k_2 = 15Nm/V$, whereas, the knee joint in this case is assumed to be ideally driven by torque directly for convenience.

Most robotic systems have parametric uncertainty problems, the CSU robot is not an excep-

Parameters	**Values**	**Units**
m_0	317.5	kg
m_1	43.28	kg
m_2	8.57	kg
m_3	2.33	kg
l_2	0.425	m
l_3	0.527	m
c_2	-0.339	m
c_3	0.32	m
I_{2z}	0.435	kg-m^2
I_{3z}	0.062	kg-m^2
J_m	0.000182	kg-m^2
r	80	-
b_1	9.75	N-m-s
b_2	1	N-m-s
f	83.33	N-s/m

Table 2.2: Parameters of CSU Robot

tion. To overcome this difficulty, a robust passivity based controller is considered due to its robust characteristic which is good at maintain performance in terms of stability, tracking errors, or other specifications despite parametric uncertainty, external disturbances or unmodeled dynamics present in the system. In this section, a robust passivity based controller is implemented to the 3-link robot system for trajectory tracking of these three joints. A Matlab Simulink diagram of the system is shown in Fig.2.6.

In the simulation, the standoff height is defined as $0.935m$, treadmill belt stiffness is 37000 N/m, and the uncertainty level of parameters is selected as 0.1. The motion profiles come from healthy human gait data [11] have been differentiated offline to generate the required feed forward term. Saturation blocks are applied in the simulation to satisfy the limitations of servo amplifier. The trajectory tracking results and control signals are shown in Fig.2.7 through Fig.2.13.

The simulation results indicate that the robust passivity based controller is able to drive the hip displacement, thigh angle and knee angle very close to the desired motion trajectories although 10 percent of parametric uncertainty exists in the model. Meanwhile, the simulated ground reaction force is located in an reasonable range. However, the chattering problem of control signals is obvious. Observing that chattering in voltage signals is not too problematic, but that for hydraulic valve opening signal can not be that noisy. The solution of solving the chattering problem will be discussed later in Chapter 5.

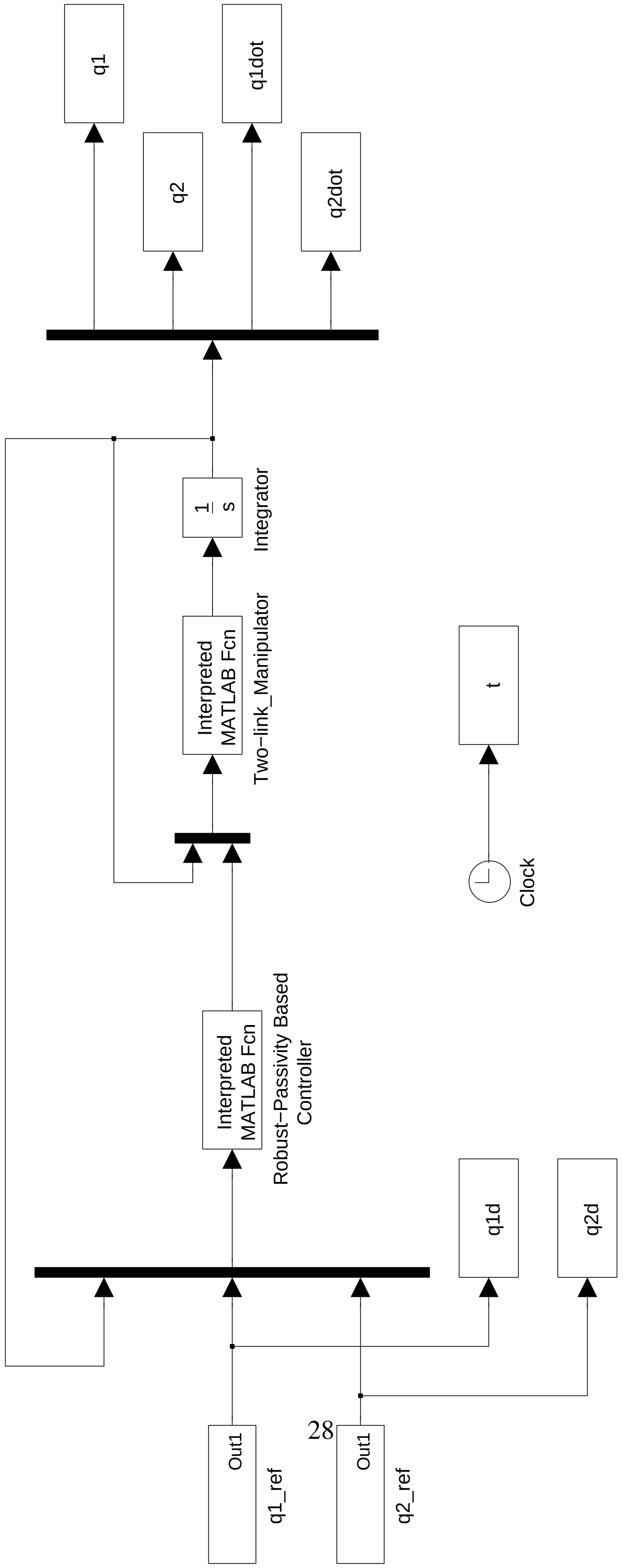

Figure 2.2: Diagram of Two-Link Planar Manipulator under RPBC in Simulink

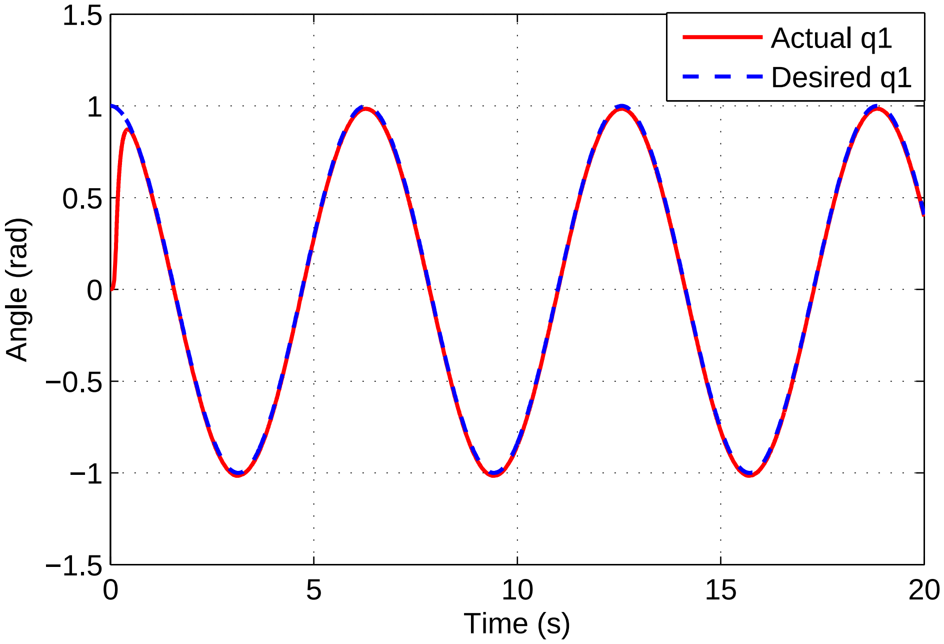

Figure 2.3: Desired and Actual Value of q1 vs. Time

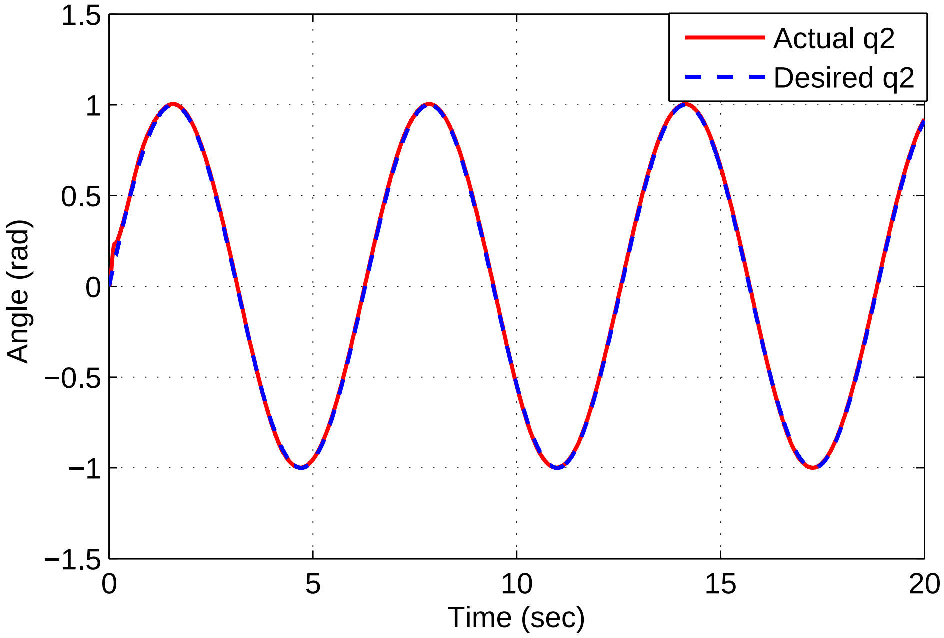

Figure 2.4: Desired and Actual Value of q2 vs. Time

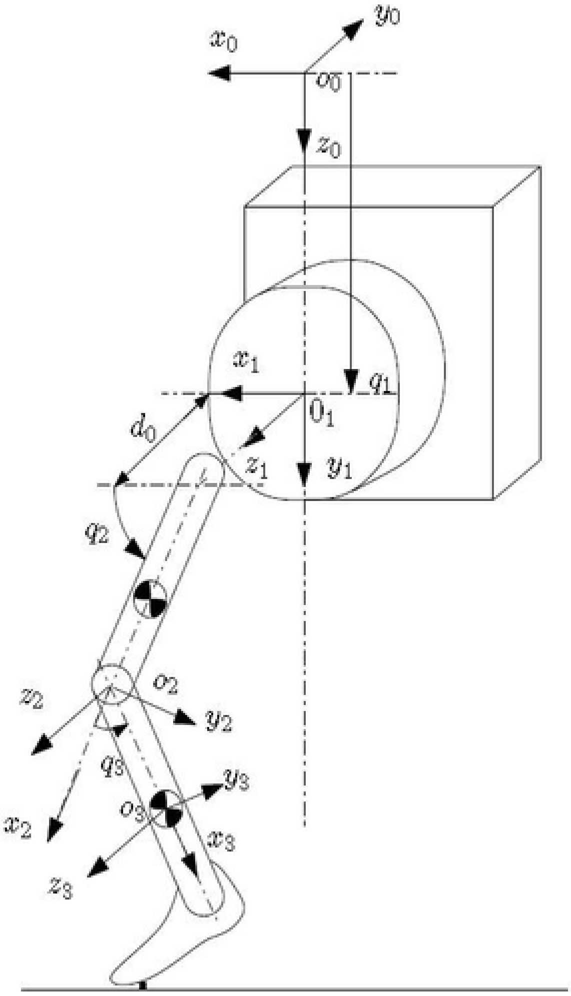

Figure 2.5: Robot Coordinate Frame

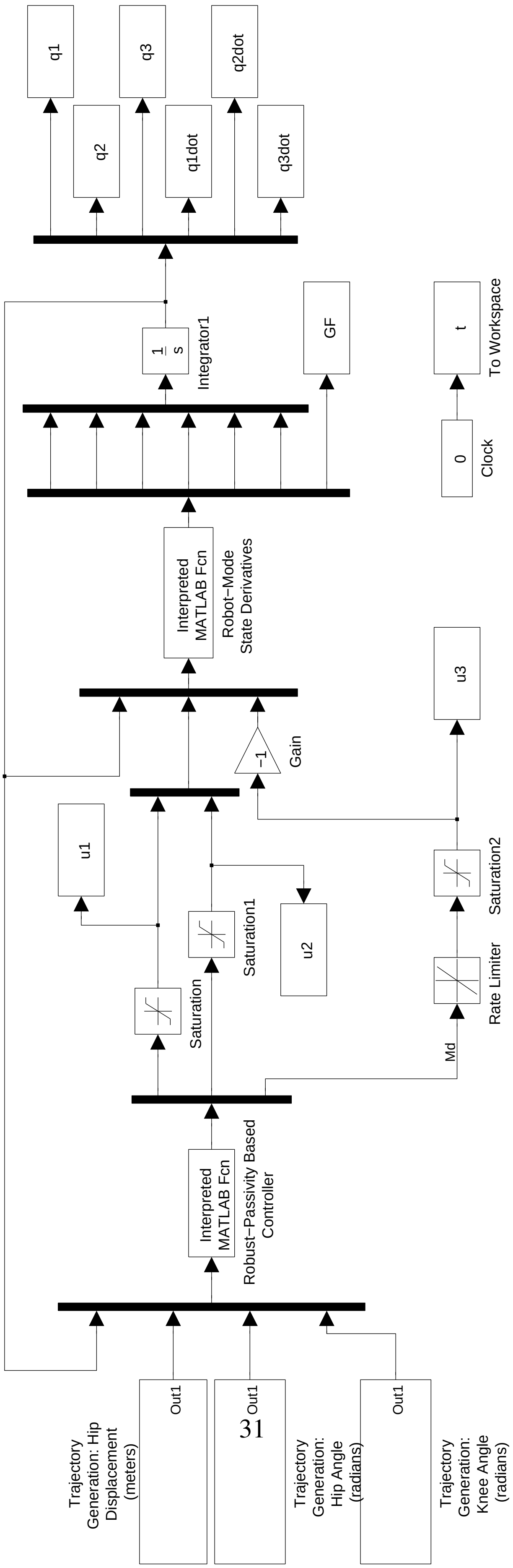
Trajectory Generation: Hip Displacement (meters)
Out1
Trajectory Generation: Hip Angle (radians)
Out1
Trajectory Generation: Knee Angle (radians)
Out1
Interpreted MATLAB Fcn
Robust−Passivity Based Controller
Saturation
Saturation1
Md
Rate Limiter
Saturation2
u1
u2
u3
−1
Gain
Interpreted MATLAB Fcn
Robot−Mode State Derivatives
1/s
Integrator1
GF
0
Clock
t
To Workspace
q1
q2
q3
q1dot
q2dot
q3dot

Figure 2.6: RPBC of CSU Robot in Matlab Simulink

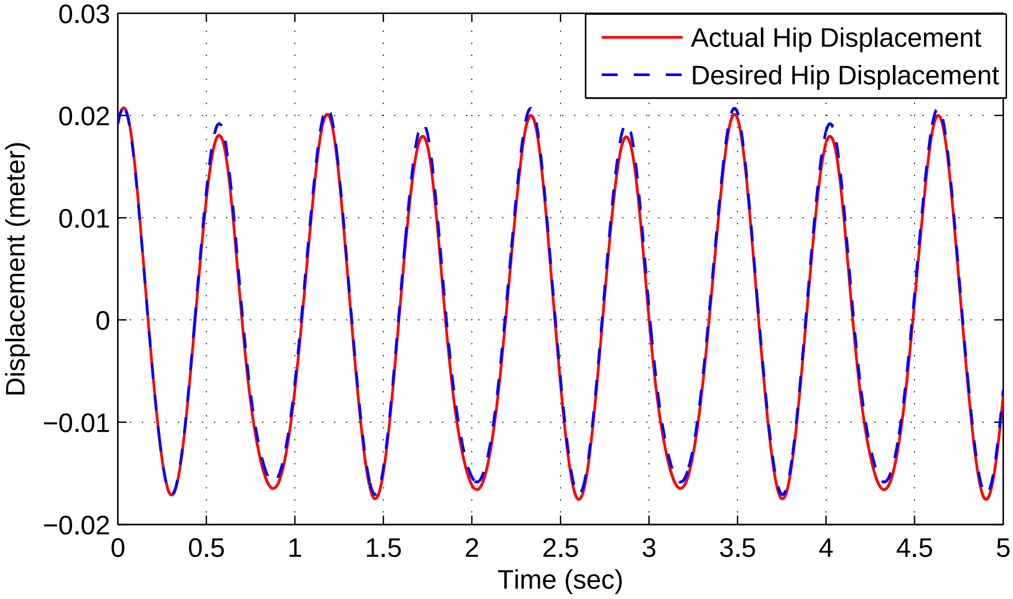

Figure 2.7: Actual and Desired Value of Hip Displacement vs. Time

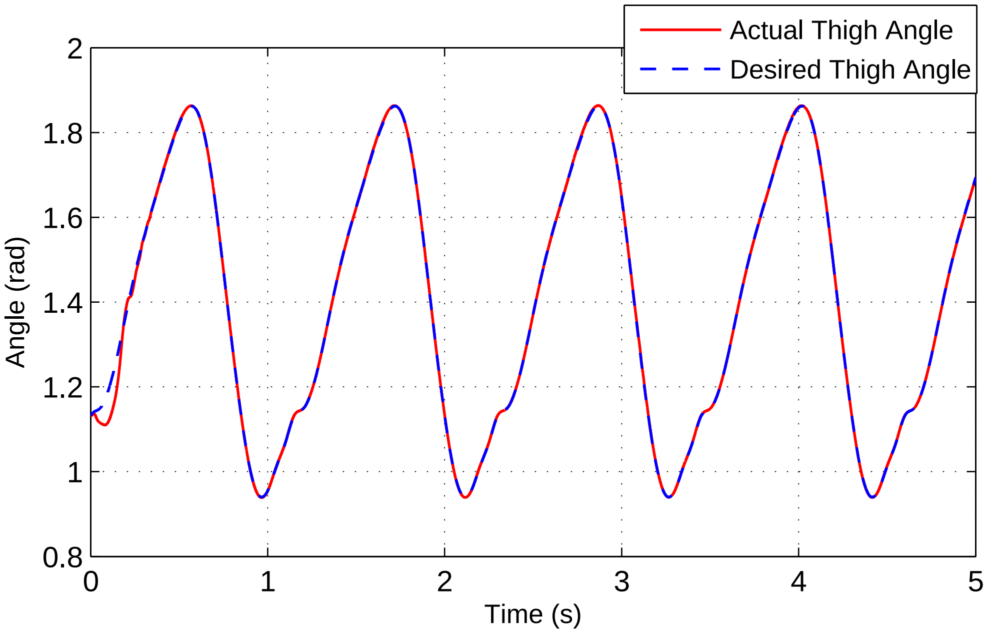

Figure 2.8: Actual and Desired Value of Thigh Angle vs. Time

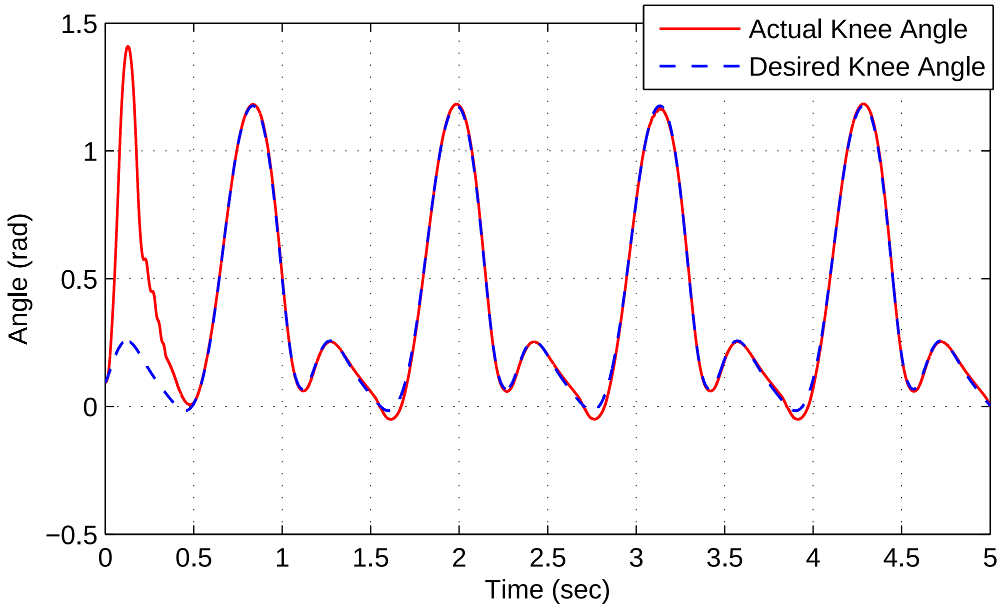

Figure 2.9: Actual and Desired Value of Knee Angle vs. Time

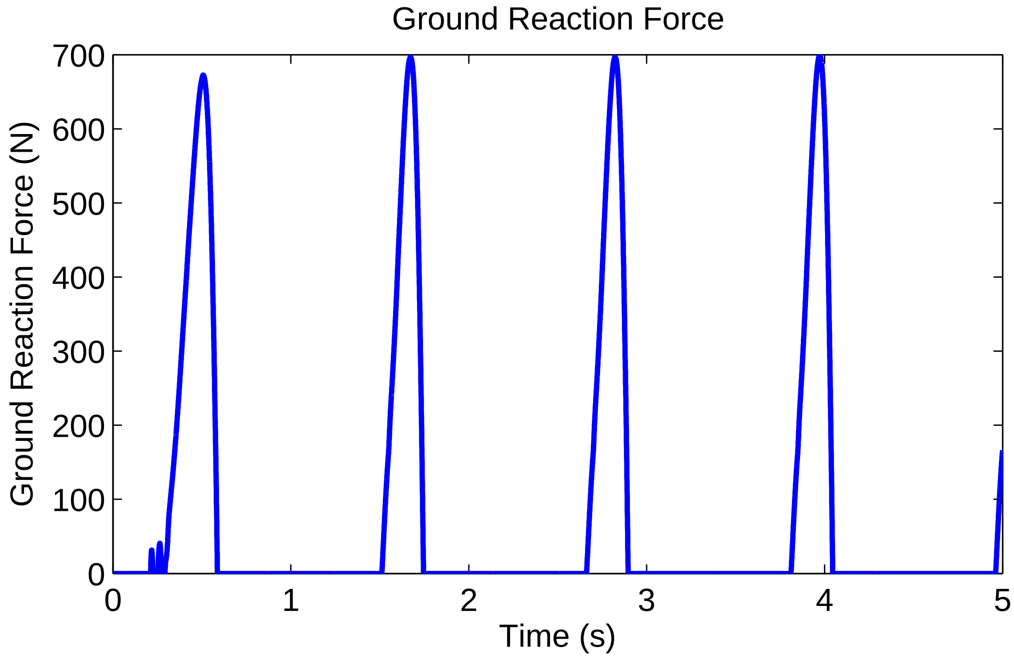

Figure 2.10: Ground Reaction Force vs. Time

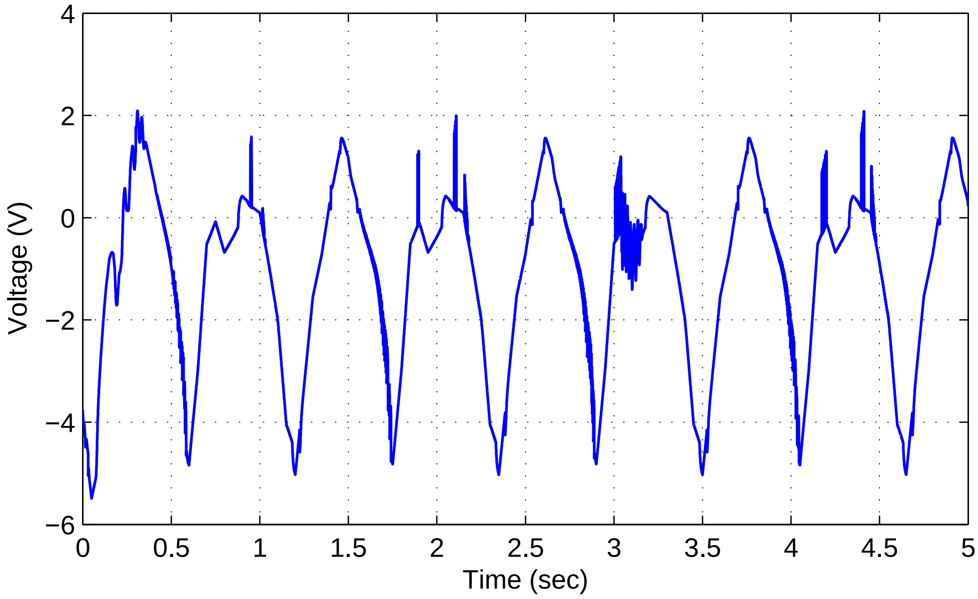

Figure 2.11: Control Signal u1 vs. Time

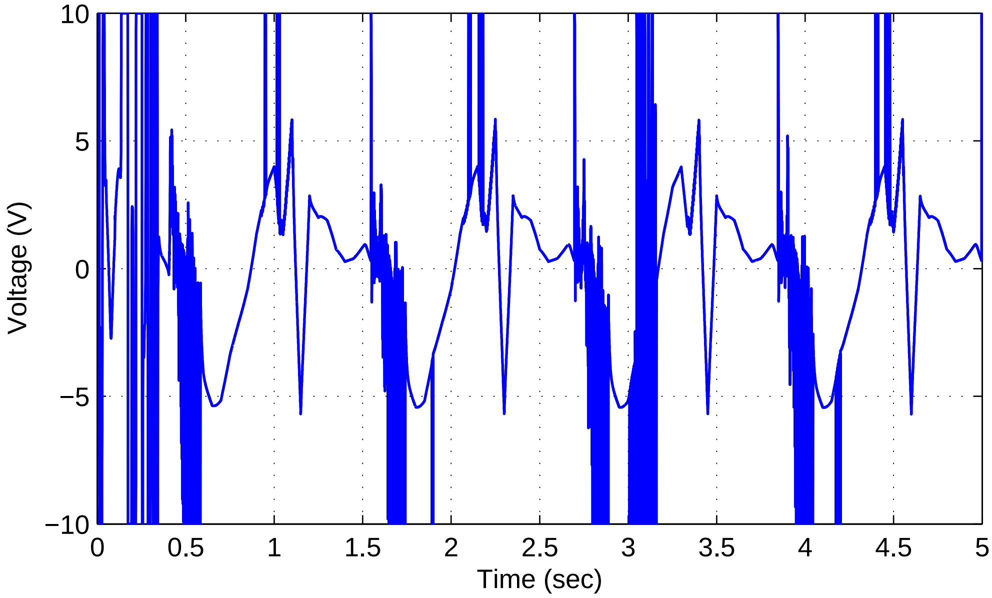

Figure 2.12: Control Signal u2 vs. Time

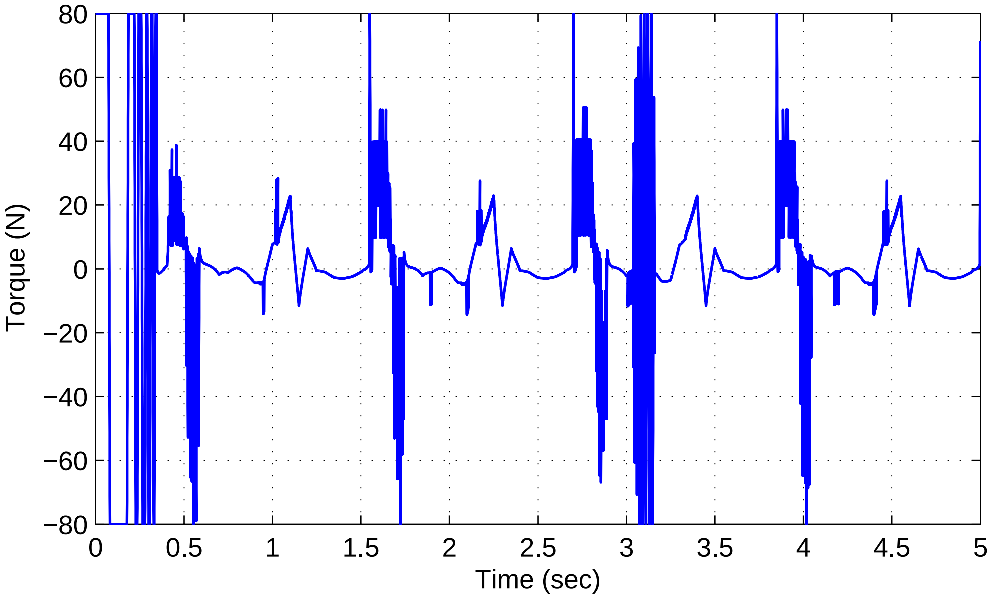

Figure 2.13: Control Signal u3 vs. Time

CHAPTER III

Hydraulic Knee Actuator

3.1 Hydraulic Knee Actuator

Controlled damper mechanisms are widely used in advanced prosthetic knees, including transfemoral devices. A damper is regarded as a brake to limit knee flexion during certain phase of walking. Such devices do not provide power to assist the user in performing motion tasks. Even slow walking involves phases where power is required. This difference relative to able-bodied joint function causes undesirable compensatory behaviors and results in a much higher energy cost for transfemoral amputees[22]. To help overcome this problem, a hydraulic knee actuator was designed at the Cleveland Clinic which can store energy from walking and release it for certain phases of walking requiring positive power. The schematic of this hydraulic knee actuator is shown in Fig. (3.1) and Table. (3.1) shows the definition of the variables.

As the figure shows, the hydraulic actuator is controlled by two valves. The high pressure valve (HPV) controls flow to an accumulator with spring where energy can be recovered and the low pressure valve (LPV) controls flow that bypasses the accumulator. The system preforms the

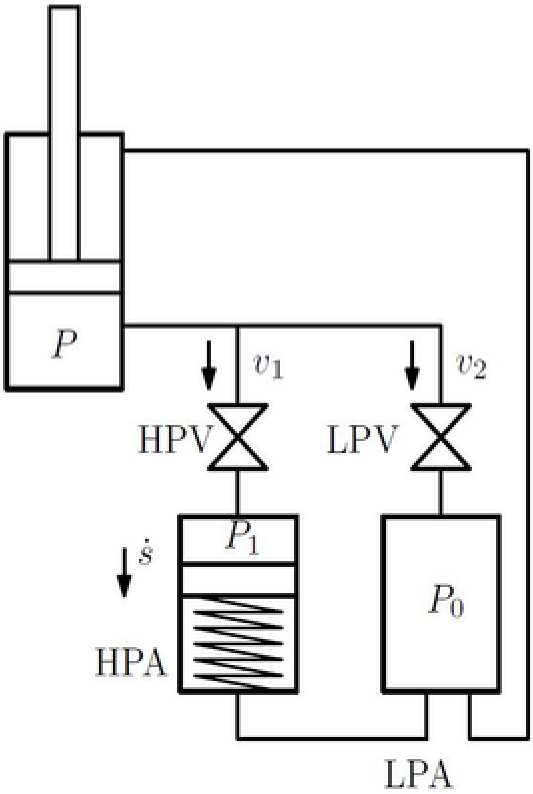

Figure 3.1: Hydraulic Schematic

same as a controlled damper device when HPV keeps closed and LPV is used for control. Complex knee functions can be achieved by controlling the two valve openings, including transitions between sitting and standing, stairs climbing and other phases of gait.

The hydraulic cylinder is attached to the prosthesis as shown in Fig. (3.2). The static force and moment relationship can be derived as

$$M = F_{cyl}\cos\gamma h \tag{3.1}$$

where

$$\cos\gamma = \frac{H\cos\alpha}{L_{cyl}} \tag{3.2}$$

Parameters	**Definitions**
P	Pressure in linear cylinder
P_1	Pressure in spring loaded reservoir
P_0	Pressure in constant pressure reservoir
s	Fluid stored in spring loaded reservoir
k	High pressure accumulator spring elasticity
u_1	Normalized control of high pressure valve normalized [0-1]
u_2	Normalized control of high pressure valve normalized [0-1]
v_1	Flow rate through high pressure valve
v_2	Flow rate through low pressure valve

Table 3.1: Parameters of Hydraulic Knee Actuator

$$L_{cyl} = \sqrt{(H\cos\alpha)^2 + (h - H\sin\alpha)^2} = \sqrt{H^2 + h^2 - 2Hh\sin\alpha} \tag{3.3}$$

so

$$M = F_{cyl}\frac{h\cos\alpha}{\sqrt{1 + (\frac{h}{H})^2 - \frac{2h\sin\alpha}{H}}} \tag{3.4}$$

now let

$$\Gamma = \frac{h\cos\alpha}{\sqrt{1 + (\frac{h}{H})^2 - \frac{2h\sin\alpha}{H}}} \tag{3.5}$$

then

$$M = \Gamma F_{cyl} \tag{3.6}$$

also

$$P = \frac{F_{cyl}}{A} = \frac{M}{\Gamma A} = \frac{1}{\Gamma A}M \tag{3.7}$$

here A is the linear cylinder piston area. Define

$$G = \frac{1}{\Gamma A} \tag{3.8}$$

then

$$P = GM \tag{3.9}$$

also volume rate in cylinder

$$v = -A\dot{L_{cyl}} = A\Gamma\dot{\alpha} = (\frac{1}{G})\dot{\alpha} \tag{3.10}$$

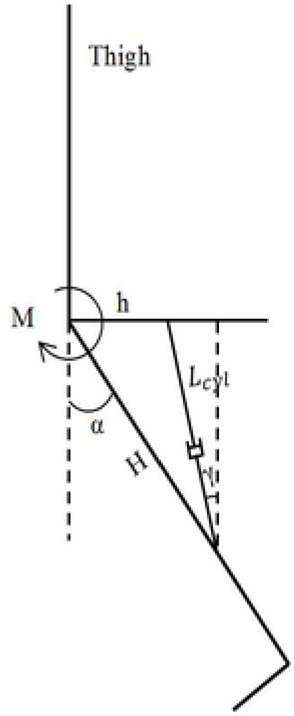

Figure 3.2: Geometry of Hydraulic Actuator

3.2 Bond Graph of Hydraulic Knee Actuator

The concept of bond graph was introduced in 1959 by Henry Paynter of the Massachussetts Institute of Technology. A dynamic system can be represented as a bond graph which describes the same information as dynamic equations. The fundamental ideal of a bond graph is that power is transported by a combination of effort and flow between connected components. In this section, a bond graph is derived with specified sign convention to represent the dynamics of CCF hydraulic knee actuator, also the relevant equations are obtained from the bond graph.

From the schematic of the CCF hydraulic knee actuator, a bond graph can be achieved as

Fig. (3.3), where *SE* (source effort) represents the torque of knee joint, *MTF* (modulated transformer) describes the power transformation from knee torque and angular velocity to hydraulic cylinder pressure and fluid flow, MR_1 (modulated resistor) and MR_2 represents the resistance of *HPV* and *LPV*, and *C* stands for the compliance of spring in the high pressure accumulator.

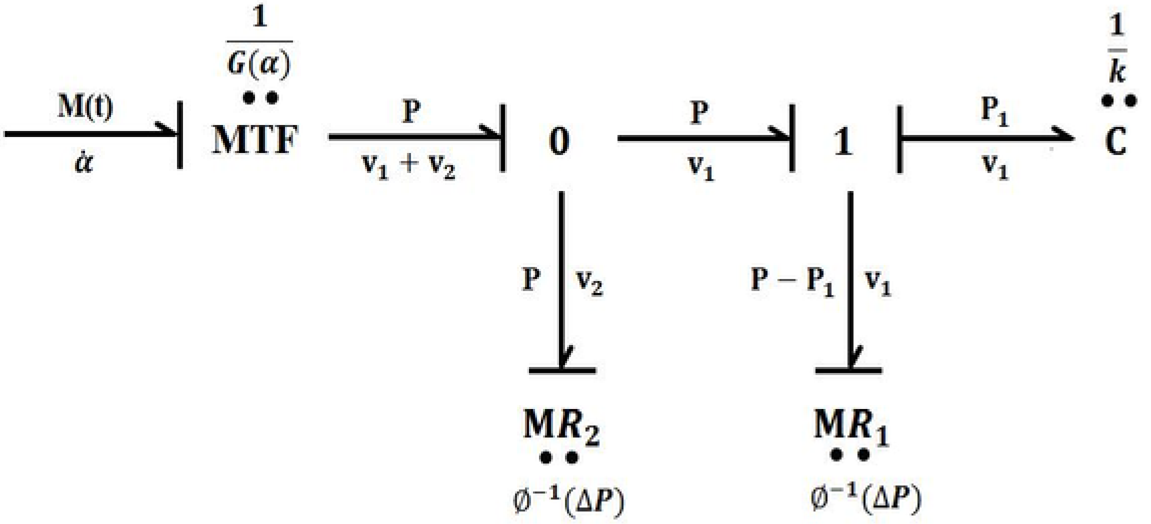

Figure 3.3: Bond Graph of Hydraulic Knee Actuator

This bond graph is developed based on effort causality, the sign convention is defined as $\dot{s} = v_1$ (flow is positive when going to the HPA and LPA). In addition, pressures in this bond graph are referenced to P_0(suppose $P_0 = 0$). The sign conventions of energy storage and return analysis is shown in Fig. (3.4).

The relationship of the pressure drop P_d and flow rate v through the control valves can be represented as equations

$$\Delta P_1 = P - P_1 = \frac{|v_1|\, v_1}{{C_{1max}}^2 {u_1}^2} + B_1 v_1 \tag{3.11}$$

$$\Delta P_2 = P - P_0 = \frac{|v_2|\, v_2}{{C_{2max}}^2 {u_2}^2} + B_2 v_2 \tag{3.12}$$

where C_{max} is the valve's lowest resistance to flow, B is the coefficient of viscous drag, and $u(t)$ is

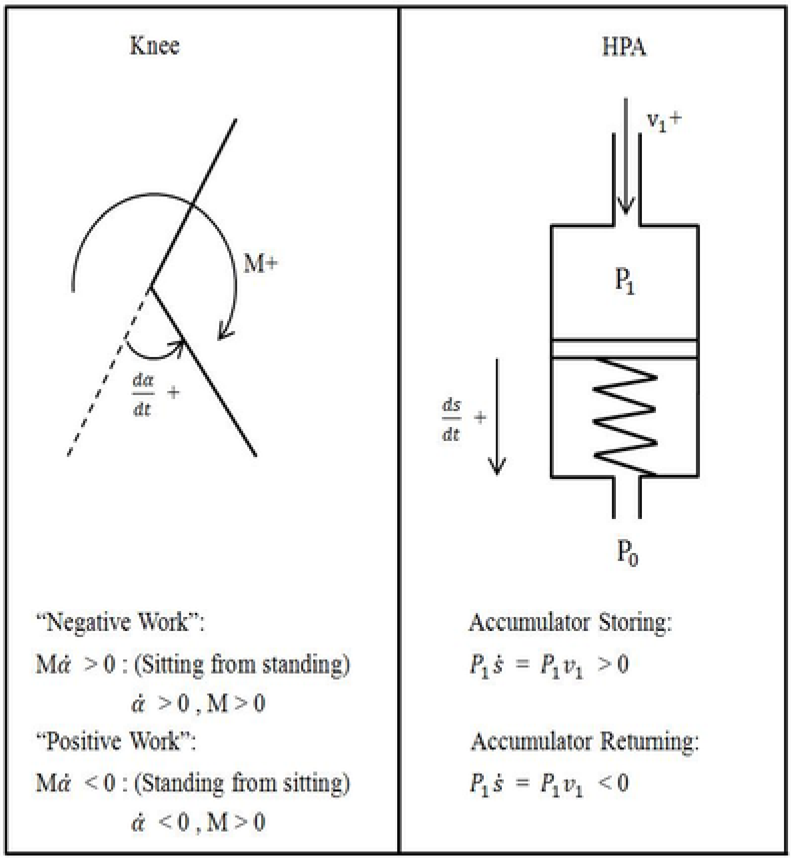
Knee
M+
$\frac{d\alpha}{dt}$ +
"Negative Work":
$M\dot{\alpha} > 0$: (Sitting from standing)
$\dot{\alpha} > 0$, M > 0
"Positive Work":
$M\dot{\alpha} < 0$: (Standing from sitting)
$\dot{\alpha} < 0$, M > 0
HPA
v_1+
P_1
$\frac{ds}{dt}$ +
P_0
Accumulator Storing:
$P_1\dot{s} = P_1 v_1 > 0$
Accumulator Returning:
$P_1\dot{s} = P_1 v_1 < 0$

Figure 3.4: Sign Convention Interpretation

the control signal between zero(closed) and one(fully open). Other equations corresponding to the bond graph are

$$M = \frac{1}{G(\alpha)} P \tag{3.13}$$

$$\dot{\alpha} = G(\alpha)(v_1 + v_2) \tag{3.14}$$

$$P_1 = ks + P_0 \tag{3.15}$$

$$\dot{s} = v_1 \tag{3.16}$$

3.3 Linear Cylinder Equations

From the equations above, the knee moment M provided by hydraulic cylinder can be found from knee angle α, knee angle velocity $\dot{\alpha}$, fluid volume s, HPV control signal u_1 and LPV control signal u_2. Different cases of combinations of v_1 and v_2 are considered to calculate M.

Case 1: When HPV is closed and LPV is open which means $v_1 = 0$, $v_2 \neq 0$.

$$\begin{aligned}
&v_2 = \dot{\alpha}/G(\alpha)\\
&P = \frac{\left|\dot{\alpha}/G(\alpha)\right|\dot{\alpha}/G(\alpha)}{{C_{2max}}^2{u_2}^2} + B_2\dot{\alpha}/G(\alpha) + P_0\\
&M = P/G(\alpha) = (\frac{\left|\dot{\alpha}/G(\alpha)\right|\dot{\alpha}/G(\alpha)}{{C_{2max}}^2{u_2}^2} + B_2\dot{\alpha}/G(\alpha) + P_0)/G(\alpha)
\end{aligned}$$

Case 2: When HPV is open and LPV is closed which means $v_1 \neq 0$, $v_2 = 0$.

$$\begin{aligned}
&v_1 = \dot{\alpha}/G(\alpha)\\
&P = \frac{\left|\dot{\alpha}/G(\alpha)\right|\dot{\alpha}/G(\alpha)}{{C_{1max}}^2{u_1}^2} + B_1\dot{\alpha}/G(\alpha) - ks - P_0\\
&M = P/G(\alpha) = (\frac{\left|\dot{\alpha}/G(\alpha)\right|\dot{\alpha}/G(\alpha)}{{C_{1max}}^2{u_1}^2} + B_1\dot{\alpha}/G(\alpha) - ks - P_0)/G(\alpha)
\end{aligned}$$

Case 3: When HPV and LPV are both closed which means $v_1 = 0$, $v_2 = 0$.

$$\dot{\alpha} = 0$$

Case 4: When HPV and LPV are both open.

$$
\begin{aligned}
& v_2 = \dot{\alpha}/G(\alpha) - v_1 \\
& \Delta P_1 = P - P_1 = \frac{|v_1|\, v_1}{{C_{1max}}^2 {u_1}^2} + B_1 v_1 \\
& \Delta P_2 = P - P_0 = \frac{\left|\dot{\alpha}/G(\alpha) - v_1\right| (\dot{\alpha}/G(\alpha) - v_1)}{{C_{2max}}^2 {u_2}^2} + B_2(\dot{\alpha}/G(\alpha) - v_1) \\
& \Delta P_1 - \Delta P_2 = -P_1 + P_0 = -ks
\end{aligned}
$$

So

$$
\frac{|v_1|\, v_1}{{C_{1max}}^2 {u_1}^2} + B_1 v_1 - \frac{\left|\dot{\alpha}/G(\alpha) - v_1\right| (\dot{\alpha}/G(\alpha) - v_1)}{{C_{2max}}^2 {u_2}^2} - B_2(\dot{\alpha}/G(\alpha) - v_1) + ks = 0 \tag{3.17}
$$

if $v_1 > 0$ and $v_2 > 0$, it can be simplified as

$$
(\frac{1}{{C_{1max}}^2 {u_1}^2} - \frac{1}{{C_{2max}}^2 {u_2}^2}){v_1}^2 + (B_1 + B_2 + \frac{2\dot{\alpha}/G(\alpha)}{{C_{2max}}^2 {u_2}^2})v_1 - \frac{\dot{\alpha}/G(\alpha)^2}{{C_{2max}}^2 {u_2}^2} - B_2\dot{\alpha}/G(\alpha) + ks = 0 \tag{3.18}
$$

Similarly, other quadratic equations could be found as

for $v_1 > 0$ and $v_2 < 0$,

$$
(\frac{1}{{C_{1max}}^2 {u_1}^2} + \frac{1}{{C_{2max}}^2 {u_2}^2}){v_1}^2 + (B_1 + B_2 - \frac{2\dot{\alpha}/G(\alpha)}{{C_{2max}}^2 {u_2}^2})v_1 + \frac{\dot{\alpha}/G(\alpha)^2}{{C_{2max}}^2 {u_2}^2} - B_2\dot{\alpha}/G(\alpha) + ks = 0 \tag{3.19}
$$

for $v_1 < 0$ and $v_2 > 0$,

$$
(-\frac{1}{{C_{1max}}^2 {u_1}^2} - \frac{1}{{C_{2max}}^2 {u_2}^2}){v_1}^2 + (B_1 + B_2 + \frac{2\dot{\alpha}/G(\alpha)}{{C_{2max}}^2 {u_2}^2})v_1 - \frac{\dot{\alpha}/G(\alpha)^2}{{C_{2max}}^2 {u_2}^2} - B_2\dot{\alpha}/G(\alpha) + ks = 0 \tag{3.20}
$$

for $v_1 < 0$ and $v_2 < 0$,

$$(-\frac{1}{{C_{1max}}^2{u_1}^2}+\frac{1}{{C_{2max}}^2{u_2}^2}){v_1}^2+(B_1+B_2-\frac{2\dot{\alpha}/G(\alpha)}{{C_{2max}}^2{u_2}^2})v_1+\frac{\dot{\alpha}/G(\alpha)^2}{{C_{2max}}^2{u_2}^2}-B_2\dot{\alpha}/G(\alpha)+ks=0 \tag{3.21}$$

Appropriate solutions of v_1 can be obtained from solving these quadratic equations. Once v_1 is known, v_2 and M will be easily calculated from Eq. (3.14) and Eq. (3.13). A Matlab function named "SolveLinearCylinderEquations" was developed by[4] and has the same function to solve knee moment from knee angle, angular velocity, fluid volumes and valve signals. This function is used in the simulation studies of Chapter 4.

3.4 Valve Positions by Approximate Inversion

At any instant in time, the moment produced by the knee actuator is a static function of the HPA volume s, the knee angle q_3, the knee angular velocity $\dot{q}_3$ and the valve positions u_1 and u_2. However, the RBPC produces a demanded knee moment at each instant in time so that knee angle can be tracked along with hip displacement and thigh angle. This leads to a model inversion problem where knee moment M is known along with q_3 and $\dot{q}_3$. Also the HPA volume given by variable s can be obtained by integration of the flow through the HPV. The objective is to find a combination of valve positions u_1 and u_2 for each instant in time to make the actuator provide the appropriate amount of knee moment.

From Eq. (3.11-3.16), the relationship between u_1, u_2 and desired knee torque is a complex nonlinear function of knee angle, knee velocity, and high pressure accumulator volume. In general, it cannot be guaranteed that solutions exist for u_1 and u_2 that result in the requested M. In addition, when solutions exist, they may be non-unique. An optimization problem needs to be solved for the valve control. In this project, a generic numeric optimization routine `fmincon` (find minimum of constrained nonlinear multivariable function) is used. It attempts to find a constrained minimum of a scalar function of several variables starting at an initial estimate. The idea is that optimized valve controls could be generated by minimizing the difference between desired knee moment and actual knee moment produced by the hydraulic system. Since the data of demanded knee moment and histories for knee angle and knee velocity are available from the research of normal human gait, the simulation for testing the optimization algorithm (fmincon) can be set up. A Matlab Simulink diagram of open-loop simulation is shown as Fig. (3.5).

The "Optimizer" block is a function that calculates the best valve controls from referenced knee moment, knee angle, knee velocity, high pressure reservoir volume and initial guesses, then the optimized u_1 and u_2 are delivered to the hydraulic knee actuator model (SolveLinearCylinderEquations) along with the referenced knee angle, knee velocity, high pressure reservoir volume. Note that the high pressure reservoir volume s is a variable which is equal to the integral of v_1. The rmscore reflects the difference between the moment generated by knee model and the reference. The simulated knee moment result is highly depend on the initial guesses in the optimizer, the initial condition of high pressure reservoir volume impacts the result obviously as well. A good set of initial conditions for s, u_1, and u_2 was determined by trial-and-error as [7 , 0.9, 0.07]. The comparison of actual knee moment and moment profile is shown as Fig. (3.6) and the valve control signals are shown as Fig. (3.7) and Fig. (3.8)

The simulation results indicates that a good set of valve positions can be obtained by minimization of the difference between actual knee moment and moment reference using fmincon in open-loop. The actual knee moment tracks the reference well in certain segments. Based on the success of the open-loop control, further implementation of fmincon in our closed-loop RBPC is expected.

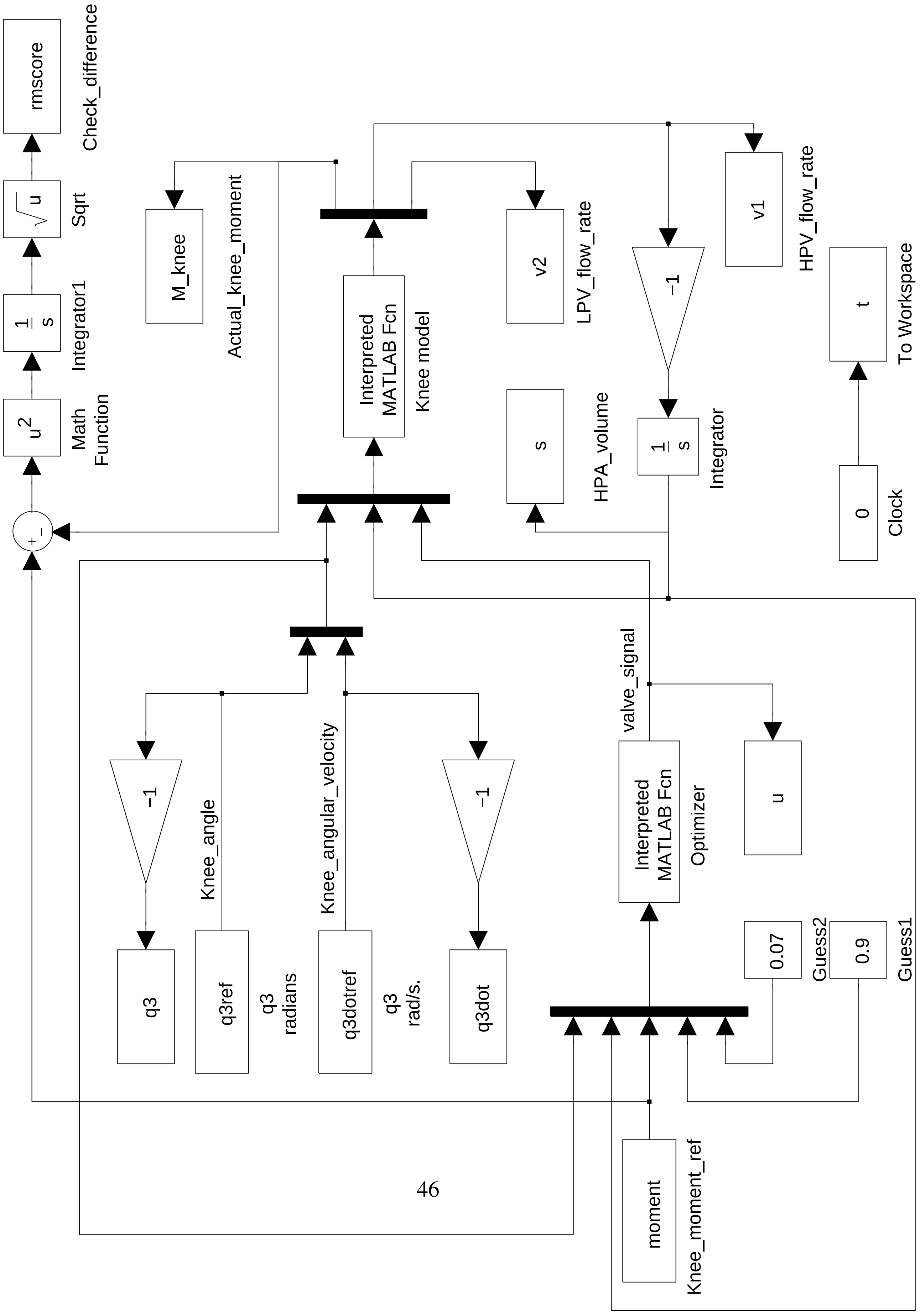

Figure 3.5: Matlab Simulink of Valve Controls Optimization

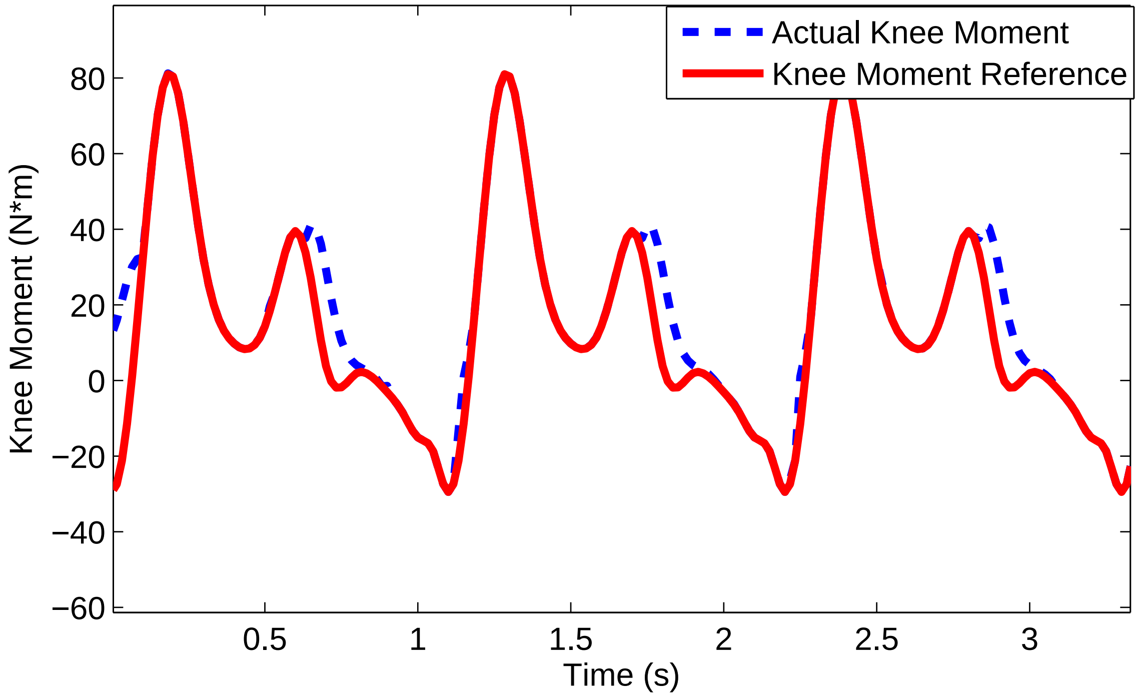

Figure 3.6: The Actual VS Reference Knee Moment

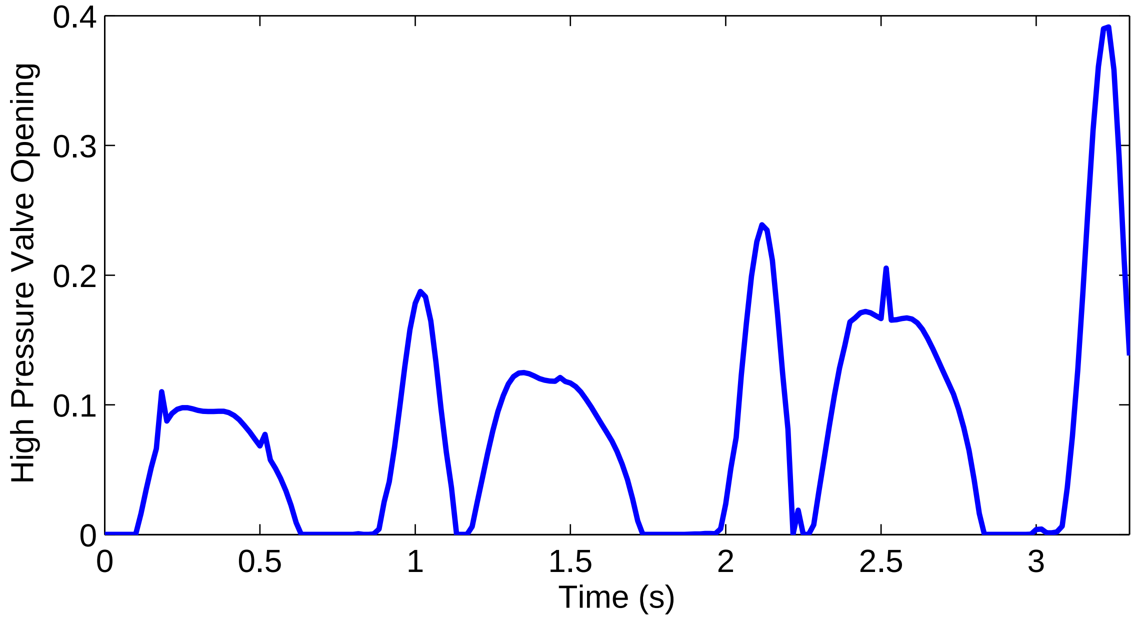

Figure 3.7: High Pressure Valve Control Signal VS Time

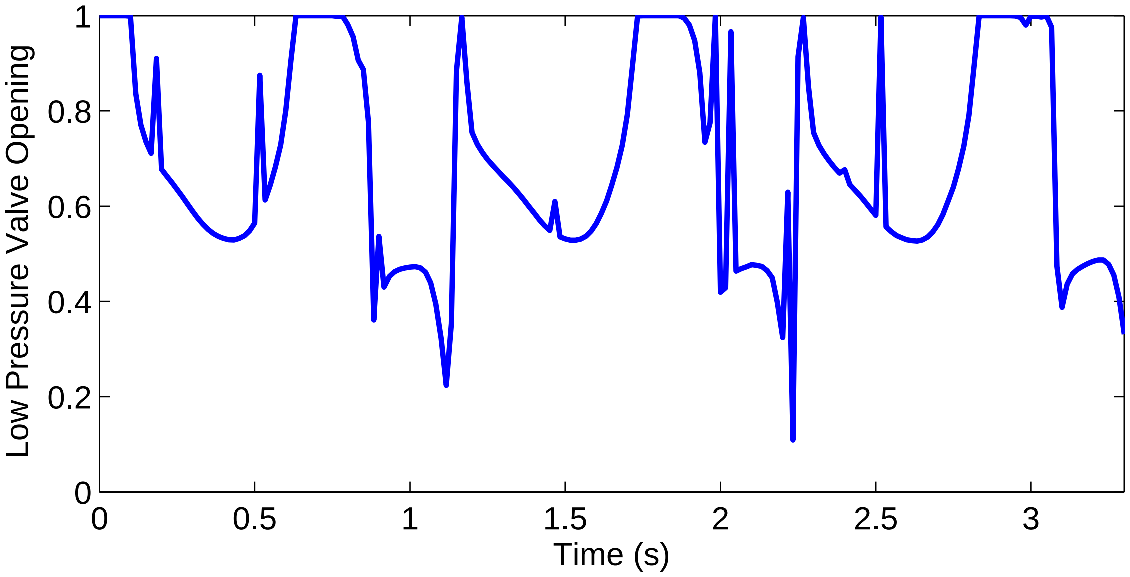

Figure 3.8: Low Pressure Valve Control Signal VS Time

CHAPTER IV

Cascade Control of Prosthesis Test Robot

4.1 Cascade Control Architecture

In the previous chapters, the implementation of Robust Passivity Based Control (RPBC) in the CSU robot system is successful, besides, the optimization problem in the CCF hydraulic knee actuator is solved properly by using `fmincon` , a general constrained minimization function available with Matlab's Optimization Toolbox. A combination of these two applications which represents a control system of a prosthesis test 3-link robot with a hydraulic knee actuator is intriguing. Therefore, a cascade control system is designed for the study. For outer loop, a robust passivity-based controller is applied to achieve tracking of reference hip displacement, thigh swing trajectories and knee angle profile obtained from able-bodied gait studies. Because of hip force and thigh moment are driven by brushless DC motors, tracking of hip displacement and thigh swing can be obtained directly by controlling the servo amplifier output voltage. However, knee moment is produced by a hydraulic actuator which includes high-pressure and low-pressure valves. An online optimizer is included in the inner loop to find the valve positions resulting in a minimal difference between demanded and actual knee moments. From Chapter 3, the moment produced by the knee actuator is a function of the form:

$$M = M(q_3, \dot{q_3}, s, u_1, u_2) \tag{4.1}$$

In a given instant of time, variables q_3, $\dot{q_3}$ and s are assumed available from sensor readings or computable from sensor readings. They are regarded as constant parameters each time the optimization problem is solved. At a given time t, the problem is formulated as:

$$\min(M(t) - M_{demand}(t))^2 \tag{4.2}$$

subject to: $0 \leq u_1 \leq 1$ and $0 \leq u_2 \leq 1$

The cascade control architecture is shown in Fig. (4.1)

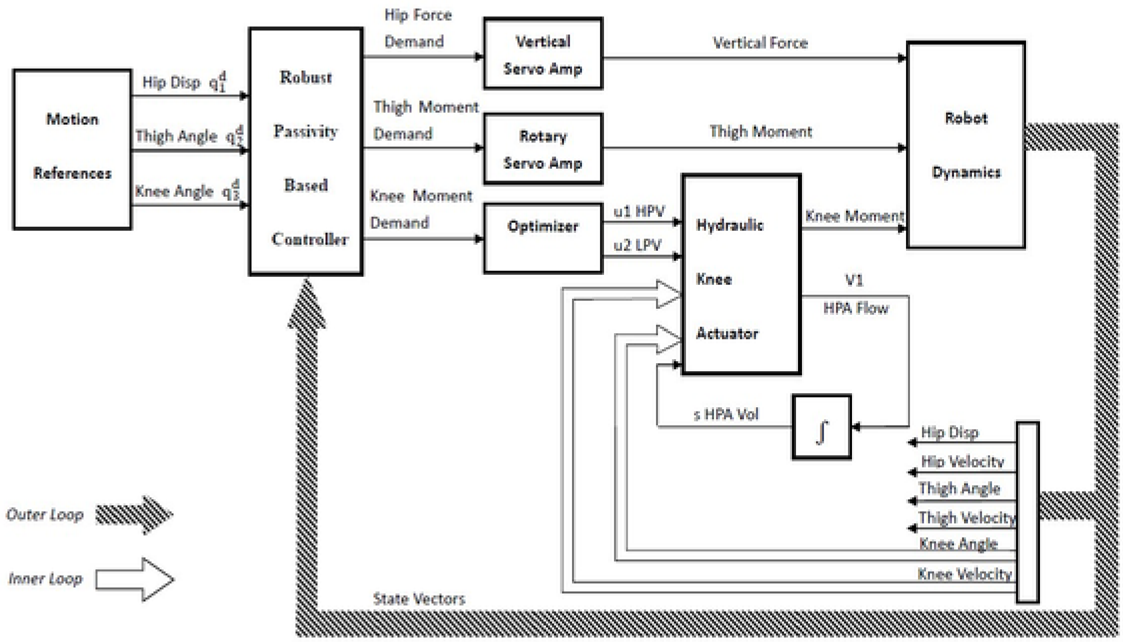

Figure 4.1: Cascade Control Architecture

4.2 Cascade Control Simulation with Original Model

A cascade control Matlab simulation is developed based on the simulation of RPBC of the CSU robot. Instead of applying moment directly to the knee joint in the robot model, the hydraulic knee actuator model is embedded in the robot model, and the control signal is generated by the optimizer. On the other side, the optimizer is now used for closed loop, other than employing the knee angle and knee velocity data from prepared reference in open loop which are collected from the real-time knee action. The Matlab Simulink diagram of the control system is shown in Fig. (4.2)

Because the mass and size of the hydraulic knee actuator are small in comparison with those of the robot, the physical parameters of the entire system are selected as in Table 2.2. Other parameters in this simulation are listed in Table 4.1. Note that the hydraulic actuator parameters are obtained from reference [11], while gains K and L, and initial guesses for the valve positions, u_{10} and u_{20} are tuned by trial-and-error. The initial accumulator volume was also increased to $7cm^3$ to obtain the results reported below.

Parameters	**Values**	**Descriptions**
K	diag[800 800 800]	RPBC gain
L	diag[100 100 400]	RPBC gain
B_1	0.127494 Mpa s cm^{-3}	Viscous drag in series with the high pressure valve
B_2	0.127494 Mpa s cm^{-3}	Viscous drag in series with the low pressure valve
C_{1max}	17.9634 cm^3 s^{-1} MPa$^{-0.5}$	The lowest resistance to flow of the high pressure valve
C_{2max}	17.9634 cm^3 s^{-1} MPa$^{-0.5}$	The lowest resistance to flow of the low pressure valve
k	3.66 Mpa cm^{-3}	High pressure accumulator spring elasticity
u_1	0-1	Normalized control of the high pressure valve
u_2	0-1	Normalized control of the low pressure valve
u_{10}	0.07	Initial opening of the high pressure valve
u_{20}	0.9	Initial opening of the low pressure valve
s_0	-7 cm^3	Initial condition of fluid stored in spring loaded reservoir

Table 4.1: Parameters of Cascade Control Simulation with Original Model

It can be seen from Fig. (4.4) and Fig. (4.5) that the actual hip displacement and thigh angle trajectories perfectly tracked the desired trajectory of real human motion. The hip joint control signal (Fig. (4.7)) and thigh joint control signal (Fig. (4.8)) are smooth and are in reasonable voltage ranges. However the knee angle tracking(Fig. (4.6)) is inferior, the actual knee angle varies from 0.3 rads to 0.7 rads which is insufficient for normal walking. As a result of the incorrect knee motion, the ground reaction force can reach only 300 N where a desired peak force should be about 1200 N.

Fig. (4.3) shows the stages of the gait cycle. The swing phase is the period from 0.7 seconds to 1.1 seconds where the ground reaction force is zero; the stance phase is the period from 1.1 second to 1.85 second where the ground reaction force varies from 0 N to 1200 N. Combining with Fig. (4.6), Fig. (4.9) and Fig. (4.10), it can be found that to improve the knee angle tracking performance, the system needs more energy which can be achieved by increasing the high pressure reservoir volume and adjusting the spring elasticity at the same time. Meanwhile, adjustment of the valve damping ratio and stiffness is necessary, because the actual knee angle oscillation level is much smaller than expected.

4.3 Cascade Control Simulation with Modified Model

4.3.1 Simulation with Modified Model

In previous section, the reason for unsatisfactory simulation result of knee action has been analyzed. To enhance knee angle tracking performance, many parameter combinations were tried in simulations. A good set of parameters is listed in Table 4.2 where values with $*$ have been modified. B_1 is decreased from 0.127 to 0.001, C_{1max} and C_{2max} are adjusted a little from 17.96 to 20 and 25, the spring elasticity is increased from 3.66 to 5, meanwhile the initial condition of fluid stored in spring loaded reservoir is enlarged ten times to $70cm^3$.

Parameters	**Values**	**Descriptions**
K	diag[800 800 800]	RPBC gain
L	diag[100 100 400]	RPBC gain
B_1	0.127494 Mpa s cm^{-3}	Viscous drag in series with the high pressure valve
B_2*	0.001 Mpa s cm^{-3}	Viscous drag in series with the low pressure valve
C_{1max}*	20 cm^3 s^{-1} MPa$^{-0.5}$	The lowest resistance to flow of the high pressure valve
C_{2max}*	25 cm^3 s^{-1} MPa$^{-0.5}$	The lowest resistance to flow of the low pressure valve
k*	5 Mpa cm^{-3}	High pressure accumulator spring elasticity
u_1	0-1	Normalized control of the high pressure valve
u_2	0-1	Normalized control of the low pressure valve
u_{10}	0.07	Initial opening of the high pressure valve
u_{20}	0.9	Initial opening of the low pressure valve
s_0*	-70 cm^3	Initial condition of fluid stored in spring loaded reservoir

Table 4.2: Parameters of Cascade Control Simulation with Modified Model
* modified parameters

It can be seen from Fig. (4.11) to Fig. (4.18) that the hip displacement and thigh angle tracking are still accurate; The knee angle has improved and the ground reaction force reaches about 1200 N which is reasonable comparing with real-human data. During the simulation, the high pressure hydraulic valve is closed for most of the time, conversely the low pressure hydraulic valve is in motion frequently, which indicates that the hydraulic knee system mainly performs as a controlled damper device. The fast switching of the valves between open and closed positions (chattering) is problematic, as real valves cannot be expected to respond so rapidly.

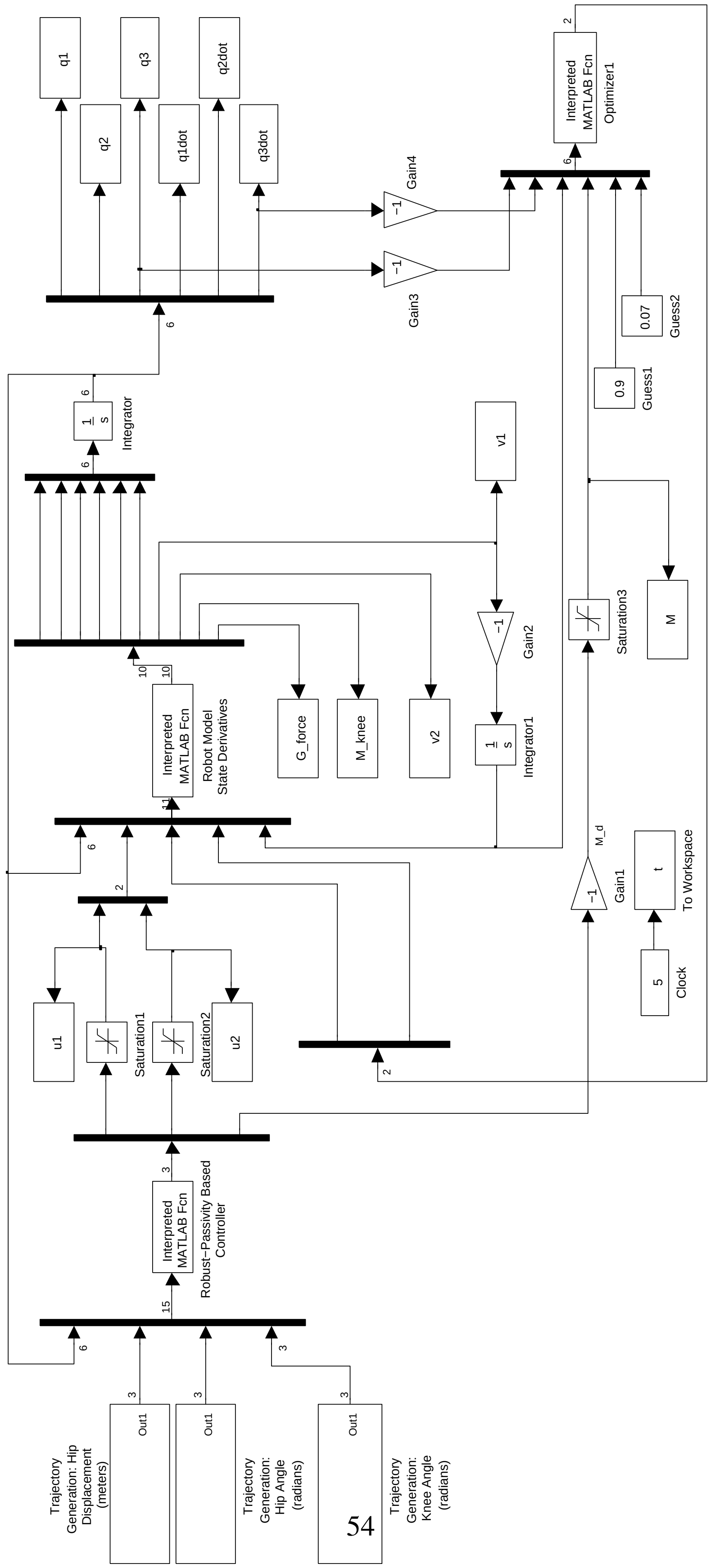

Figure 4.2: Diagram of The Cascade Control Model in Simulink

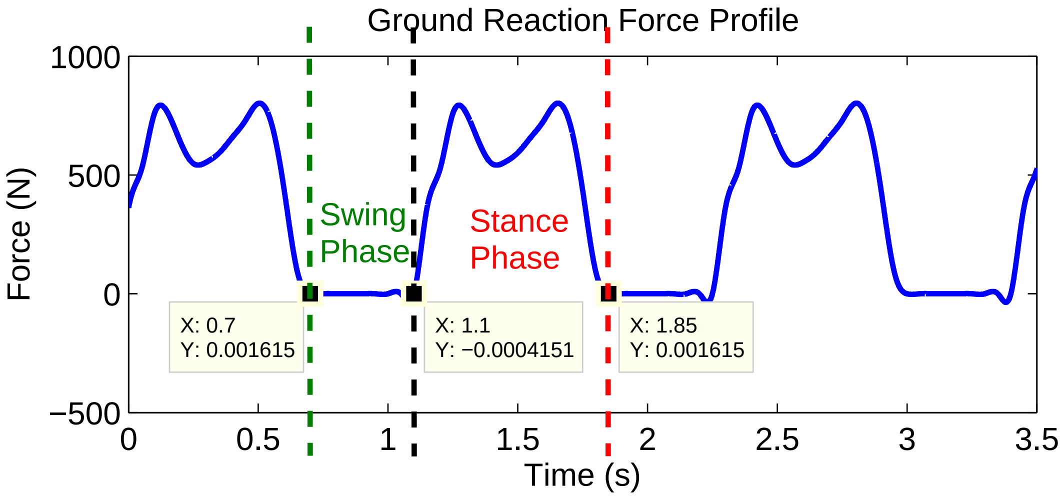

Figure 4.3: Ground Reaction Force Profile

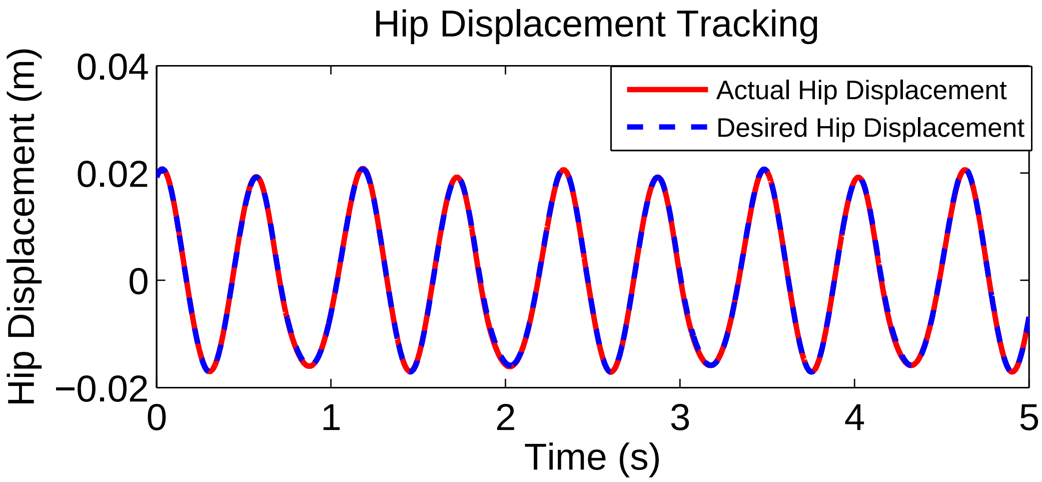

Figure 4.4: Hip Displacement Tracking vs. Time with Original Model

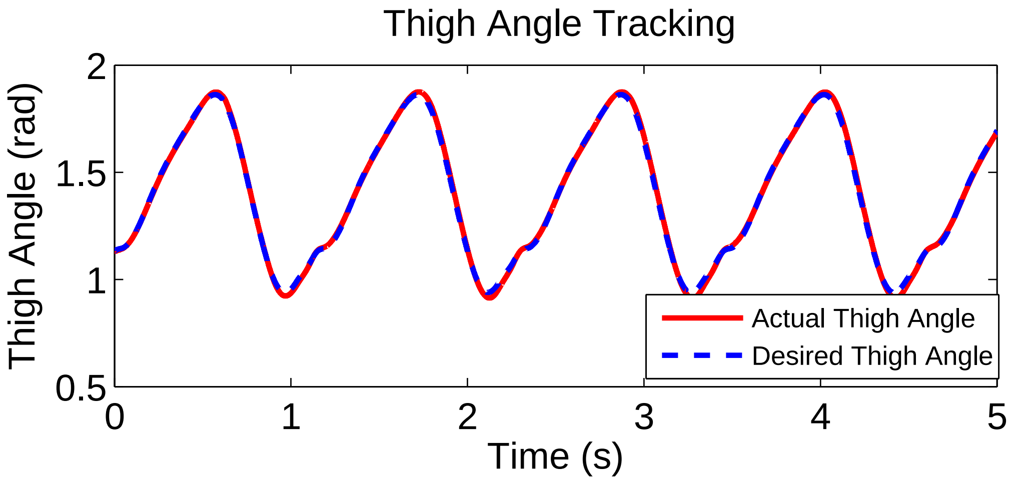

Figure 4.5: Thigh Angle Tracking vs. Time with Original Model

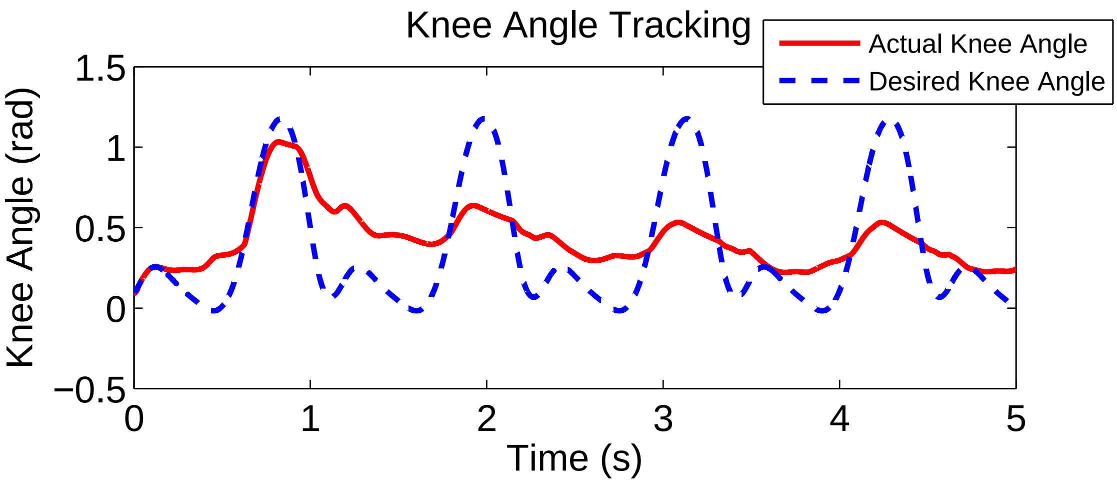

Figure 4.6: Knee Angle Tracking vs. Time with Original Model

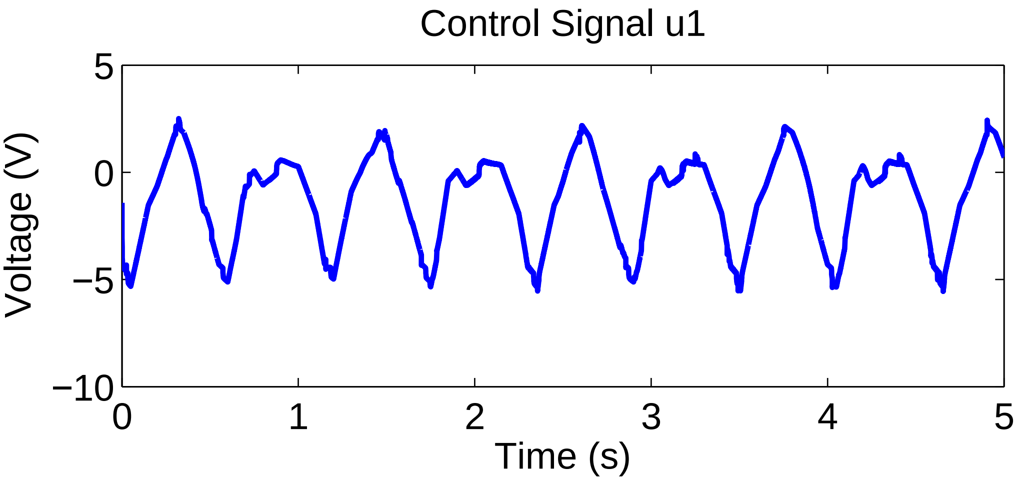

Figure 4.7: Control Signal u_1 vs. Time with Original Model

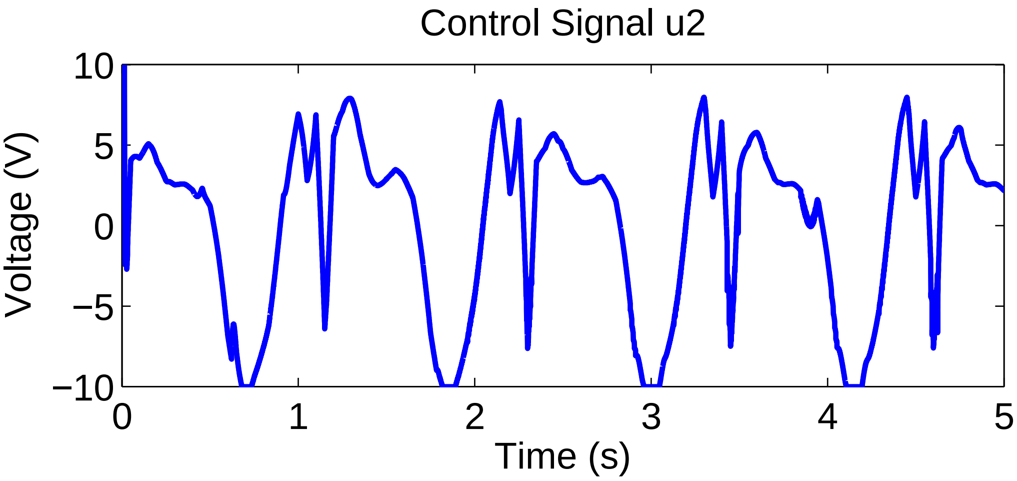

Figure 4.8: Control Signal u_2 vs. Time with Original Model

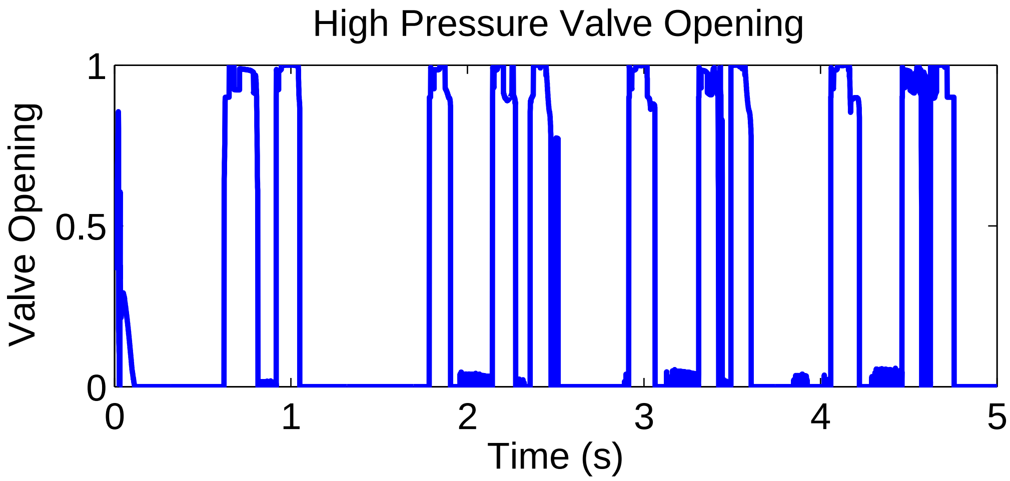

Figure 4.9: High Pressure Valve Opening vs. Time with Original Model

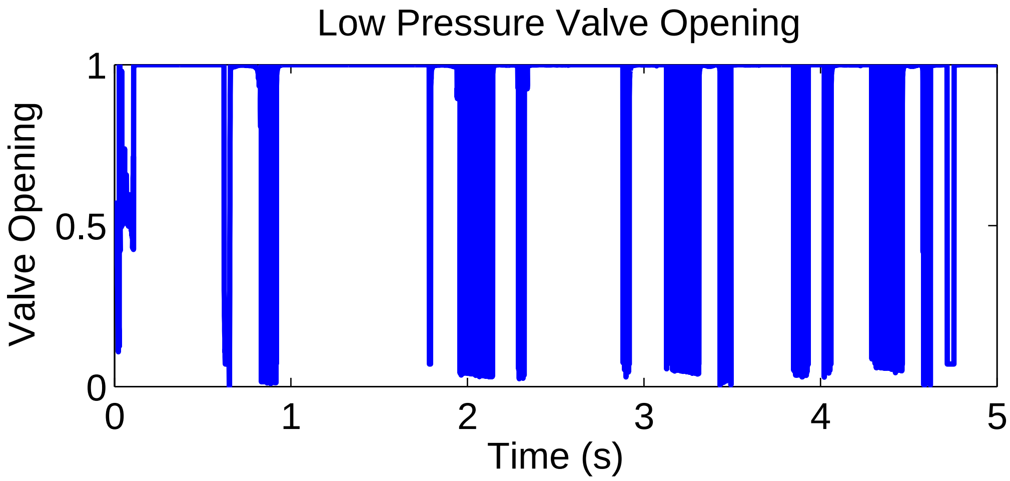

Figure 4.10: Low Pressure Valve Opening vs. Time with Original Model

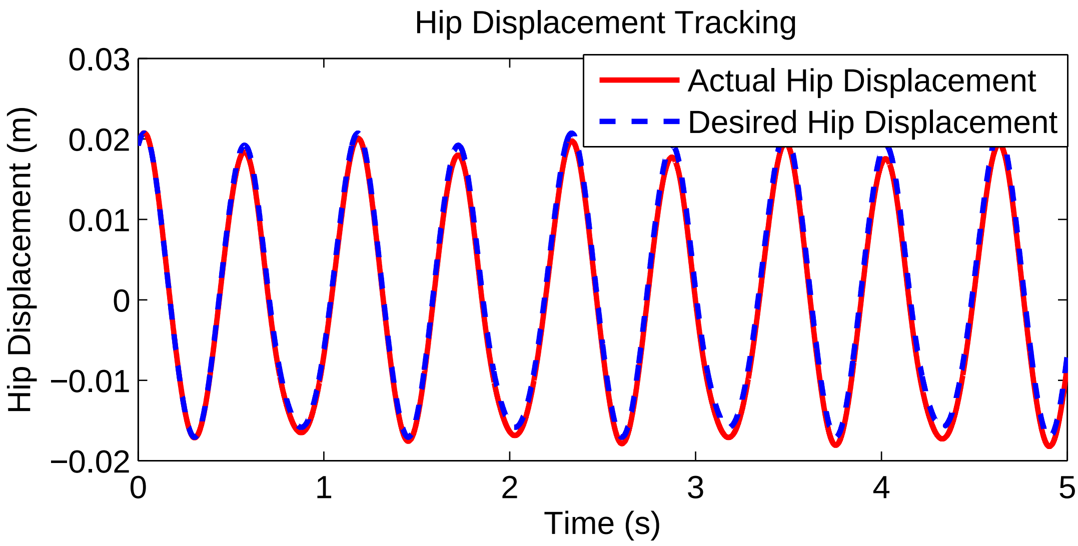

Figure 4.11: Hip Displacement Tracking vs. Time in Modified Model

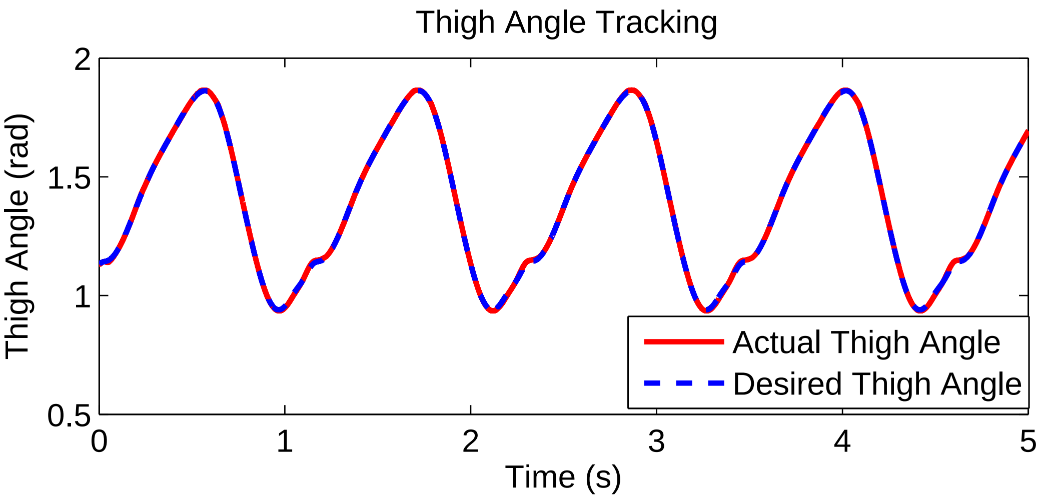

Figure 4.12: Thigh Angle Tracking vs. Time in Modified Model

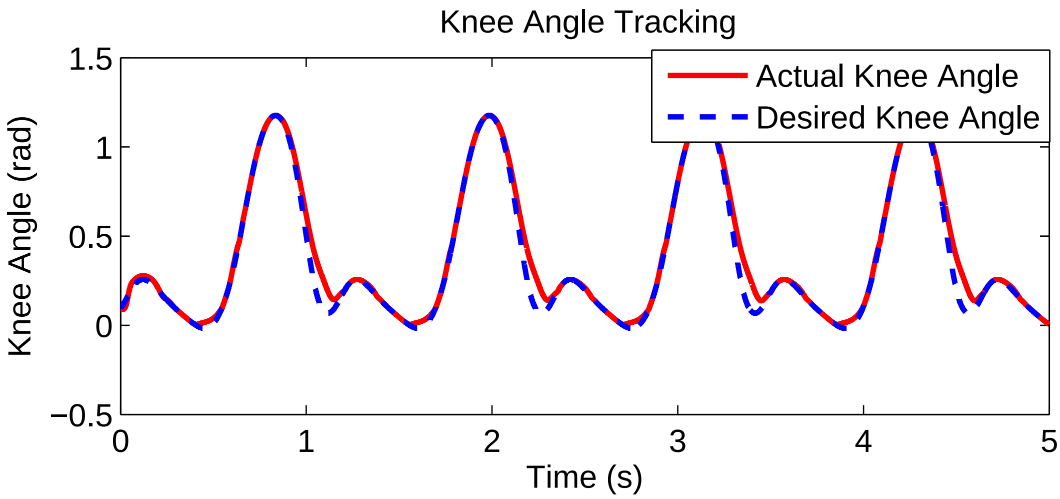

Figure 4.13: Knee Angle Tracking vs. Time in Modified Model

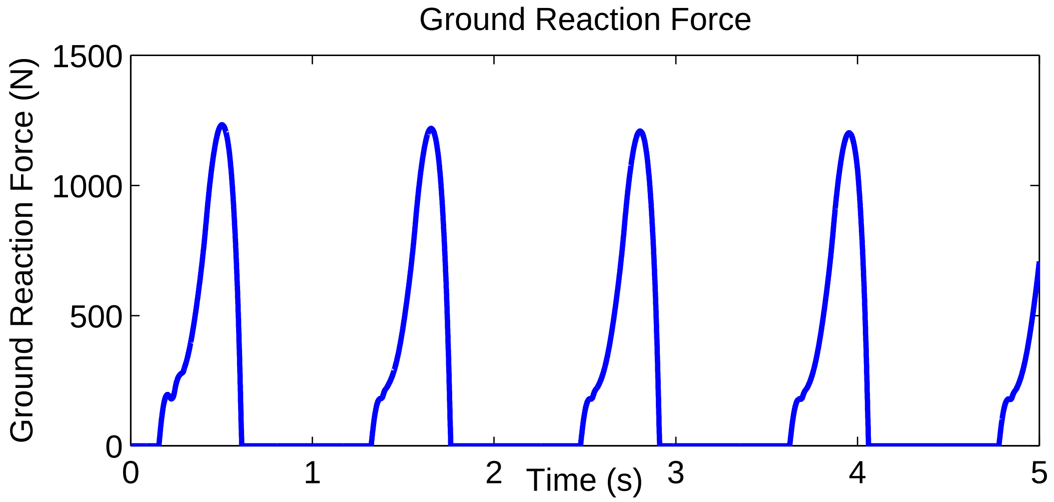

Figure 4.14: Ground Reaction Force vs. Time in Modified Model

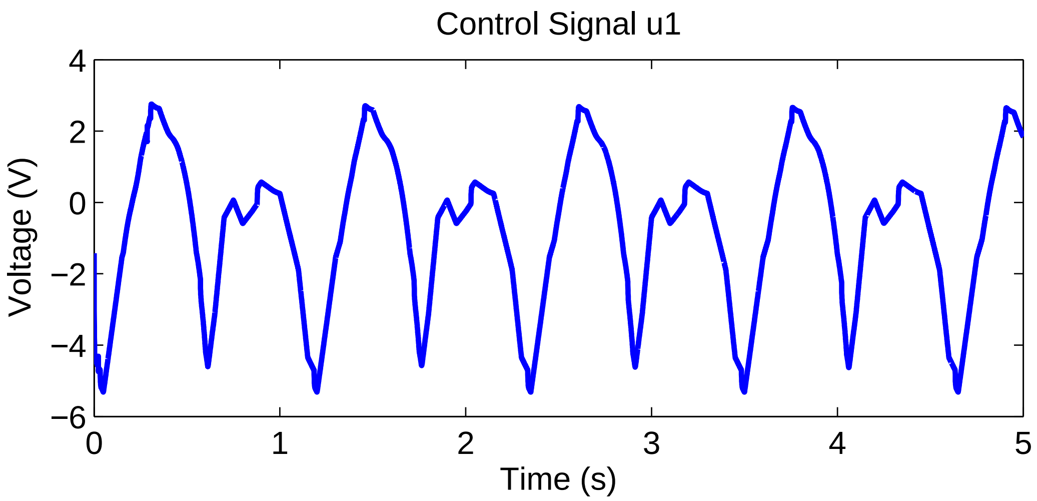

Figure 4.15: Control Signal u_1 vs. Time in Modified Model

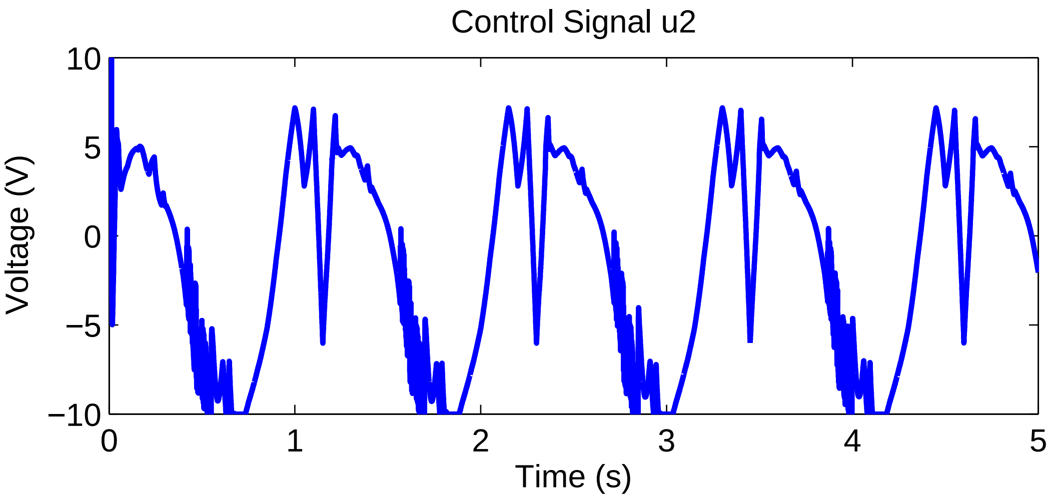

Figure 4.16: Control Signal u_2 vs. Time in Modified Model

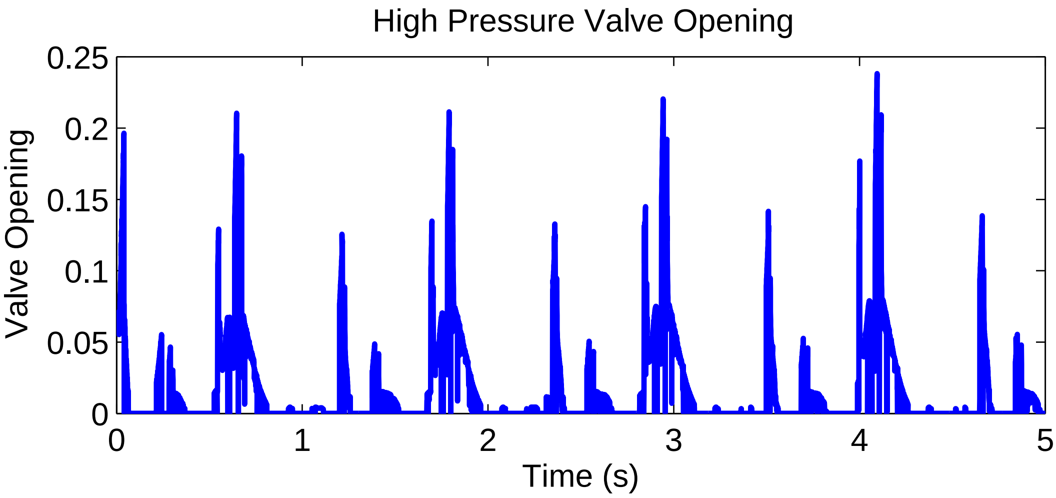

Figure 4.17: Low Pressure Valve Opening vs. Time in Modified Model

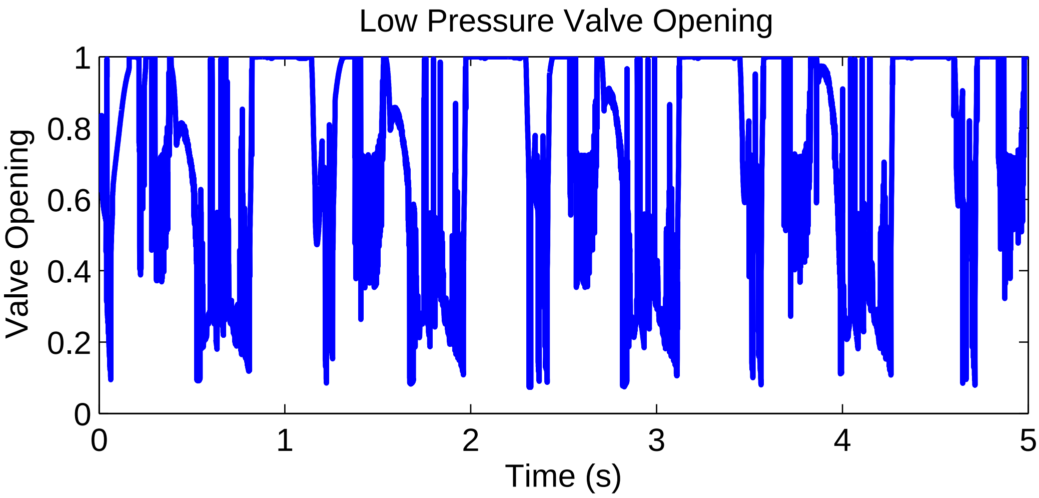

Figure 4.18: Low Pressure Valve Opening vs. Time in Modified Model

4.3.2 Simulation with Low-Pass Filter

To solve this signal chattering issue, a Simulink diagram with low pass filters is built as Fig. (4.19). The filters are added in between of optimizer and robot model as transfer functions in form of $\frac{1}{\tau s+1}$, where τ is the filter time constant. In this project, the constants are selected as 0.002 and 0.008 for high pressure valve and low pressure valve respectively. The corresponding simulation results are listed in Fig. (4.20) to Fig. (4.27). It can be seen that knee angle tracking performance is slightly poorer when the low-pass filters are added. However, chattering in the low-pressure valve is significantly smaller. The high-pressure valve also exhibits a reduction in chattering.

In order to quantify the tradeoff between chattering and tracking error, an RMS approach is used for knee angle tracking and a spectral energy method is used for valve chattering. The RMS scores are computed by

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2} \tag{4.3}$$

where N is the number of samples, δ stands for the tracking error of knee angle in rads. The RMS scores of simulations in 5 seconds are listed in Table 4.3.

	Knee Angle Tracking with Filters	**Knee Angle Tracking without Filters**
RMS Score	3.08e-3 rads	1.25e-3 rads

Table 4.3: Comparison of RMS

Meanwhile, the chattering of valve signals are calculated by applying Fast Fourier Transform (FFT) method. The FFT magnitude of the valve signals are shown in Fig. (4.28) to Fig. (4.31). To obtain a measure of chattering, the FFT magnitudes are integrated (summed) over the frequency range of the computation. This measure is related to the amount of energy contained in the signal, which is a good and practical indication of chattering. The results are shown in Table 4.4. It can be seen that chattering is reduced by approximately 50 percent, which is results in much quieter and smoother valve actions. Thus, it is advisable to include the filters, albeit at the cost of losing a little of knee angle tracking.

	Chattering with Filters	**Chattering without Filters**
High Pressure Valve Signal	3.86e+3	6.0e+3
Low Pressure Valve Signal	2.72e+4	5.58e+4

Table 4.4: Chattering of Valve Signals

4.4 Conclusion

In this chapter, the simulation of cascade control with RPBC and optimizer has been built and tested. There are physical limitations in the original hydraulic system that limit the achievable performance of the control system. Therefore, a modified hydraulic knee model has been developed by adjusting parameter values. Beside, a set of low-pass filters have been considered to eliminate the chattering problem in hydraulic valves.

In general, according to the simulation results, the cascade control with RPBC and optimizer is adequate to solve the control problem of the three link prosthesis test robot with modified hydraulic knee actuator. In the outer loop, the robust characteristic of RPBC can overcome the parametric uncertainty effectively, at the same time it can insure the tracking properly. In the inner loop, the optimizer can solve the highly nonlinear quadratic equations to control the opening of the two hydraulic valves to obtain the desired moment from the outer loop controller.

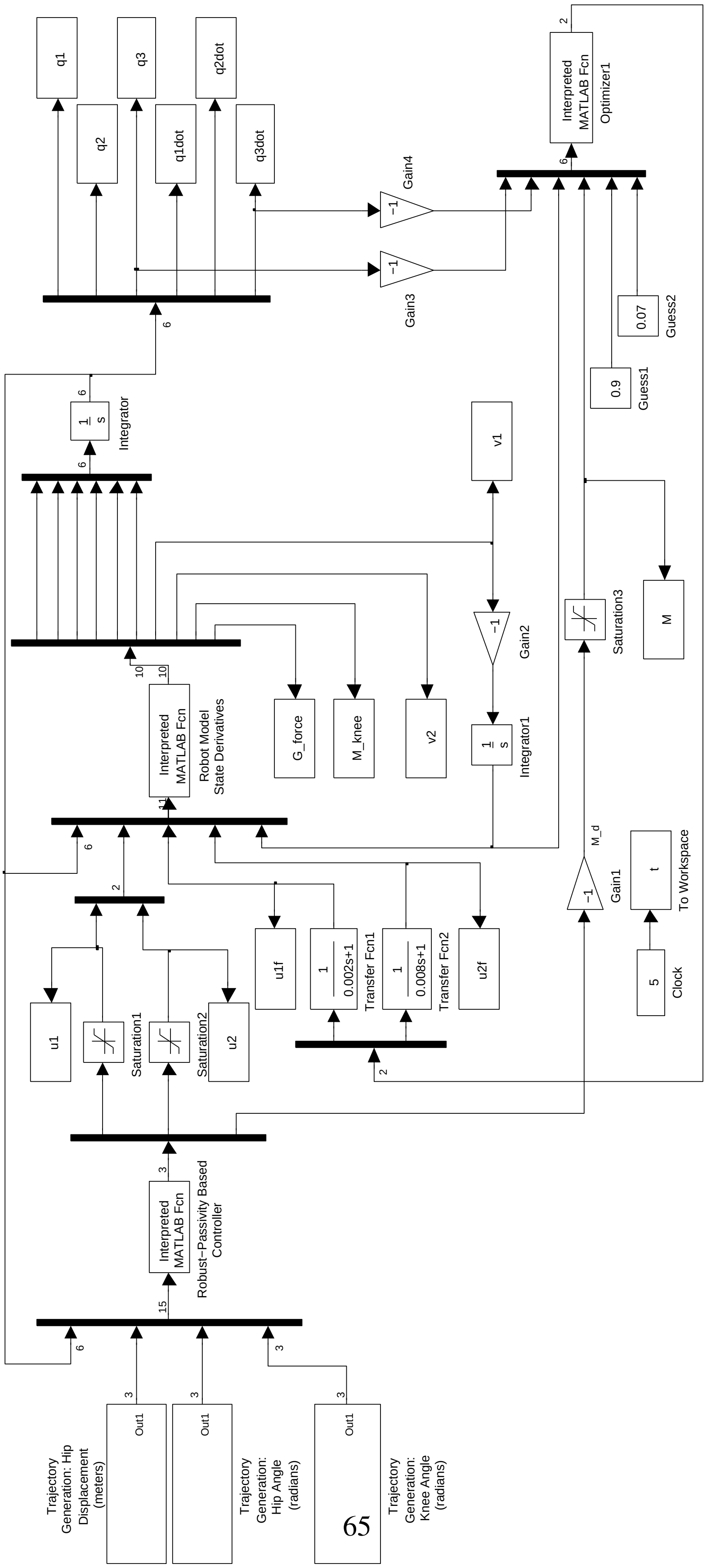

Figure 4.19: Diagram of The Cascade Control Model with Filters in Simulink

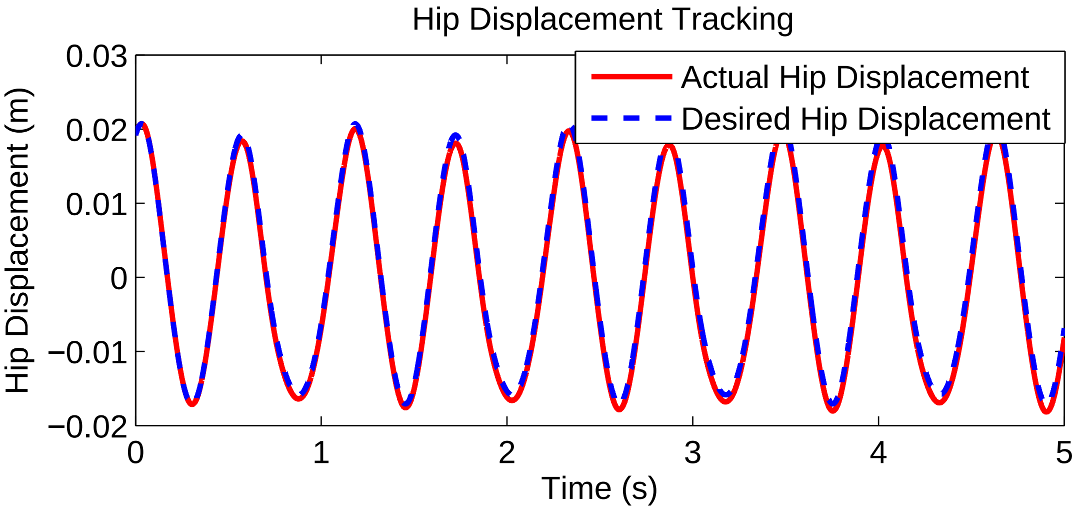

Figure 4.20: Hip Displacement Tracking vs. Time in Modified Model with Filter

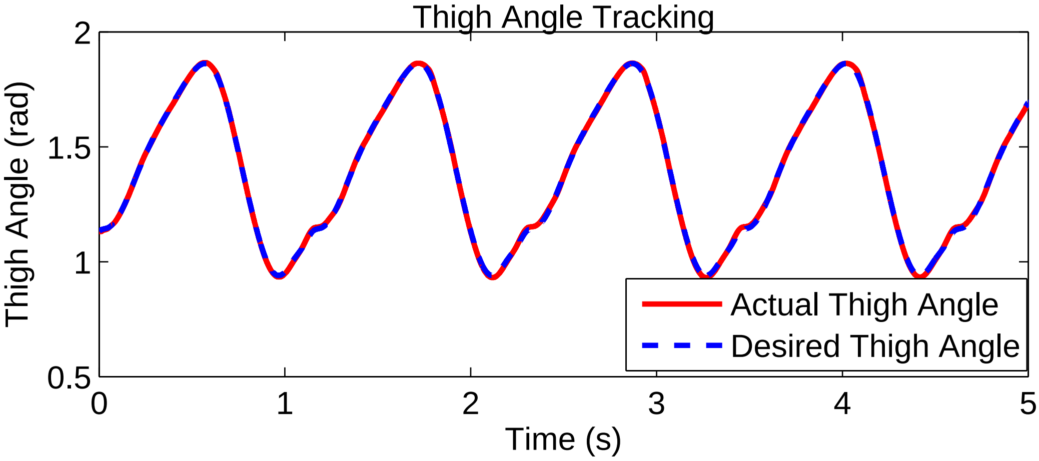

Figure 4.21: Thigh Angle Tracking vs. Time in Modified Model with Filter

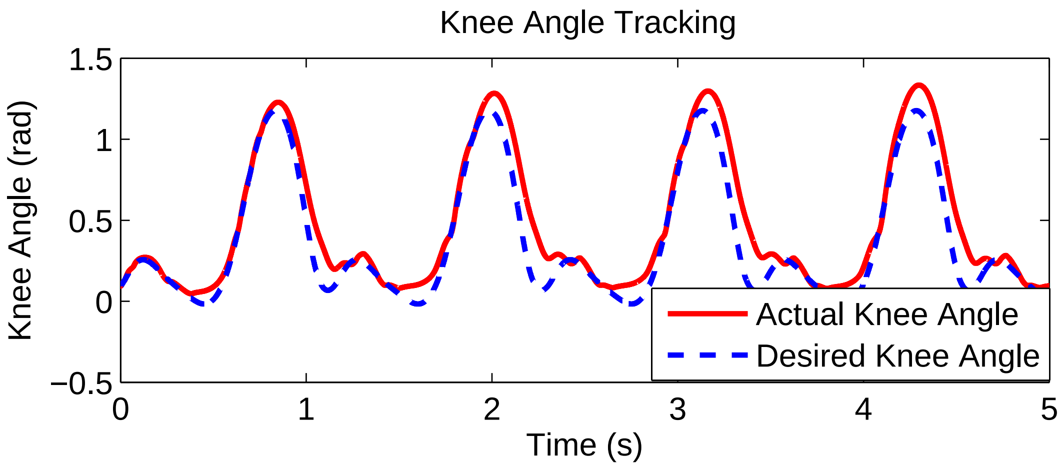

Figure 4.22: Knee Angle Tracking vs. Time in Modified Model with Filter

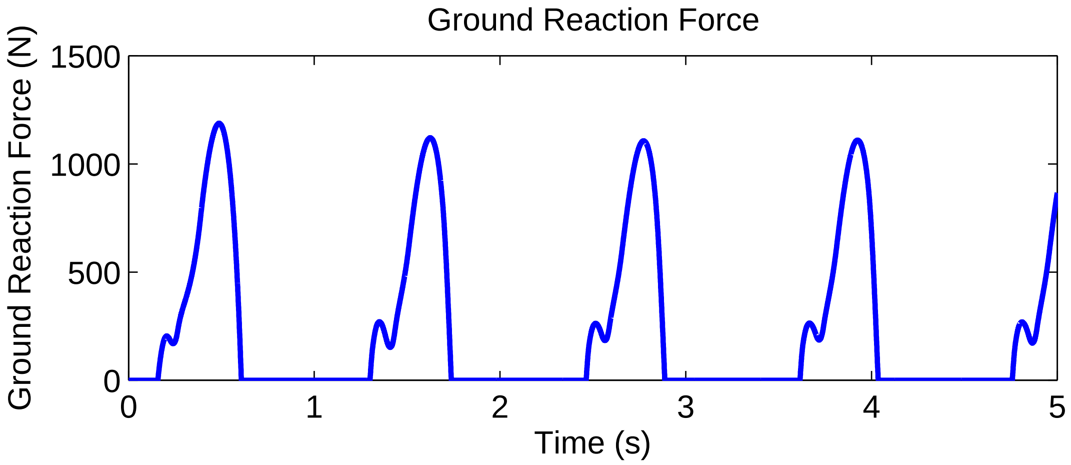

Figure 4.23: Ground Reaction Force vs. Time in Modified Model with Filter

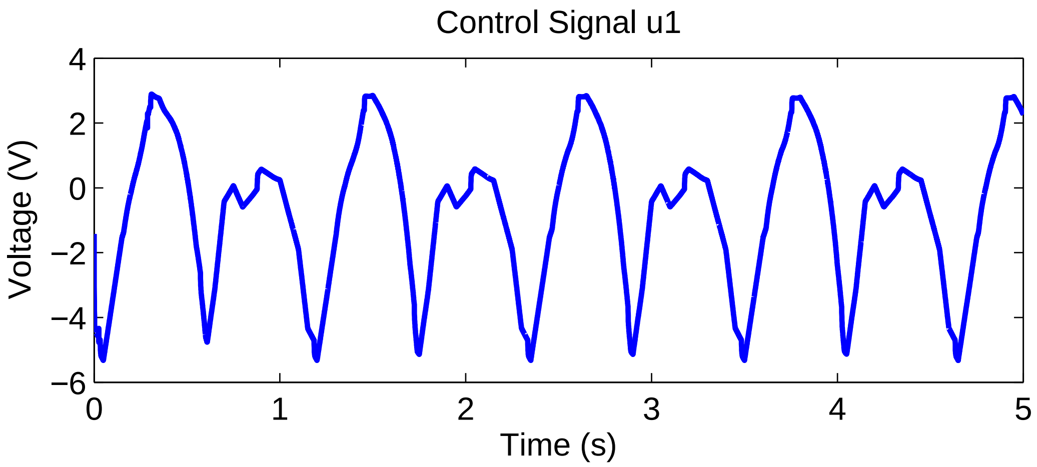

Figure 4.24: Control Signal u_1 vs. Time in Modified Model with Filter

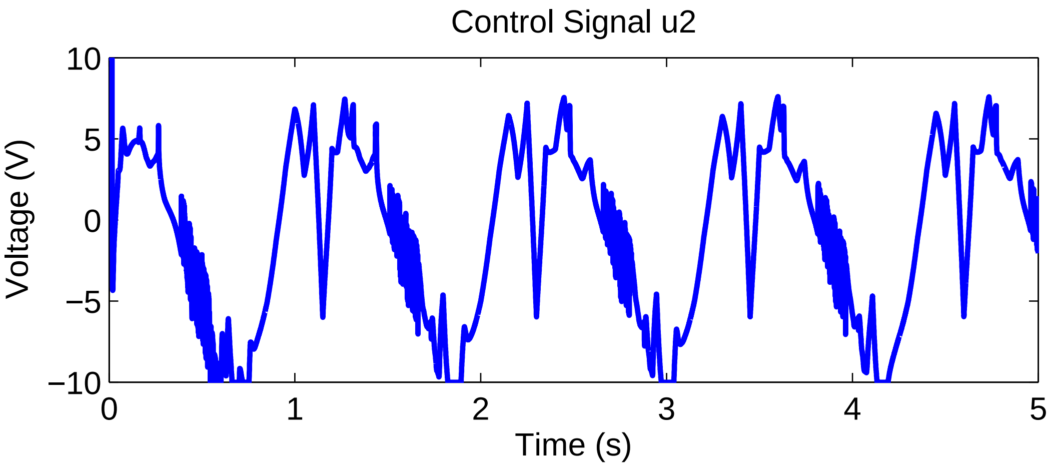

Figure 4.25: Control Signal u_2 vs. Time in Modified Model with Filter

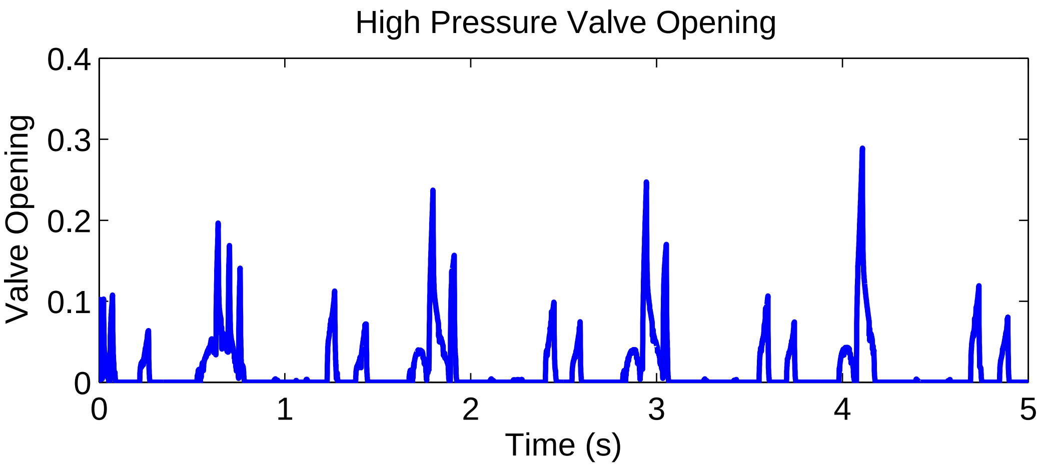

Figure 4.26: Low Pressure Valve Opening vs. Time in Modified Model with Filter

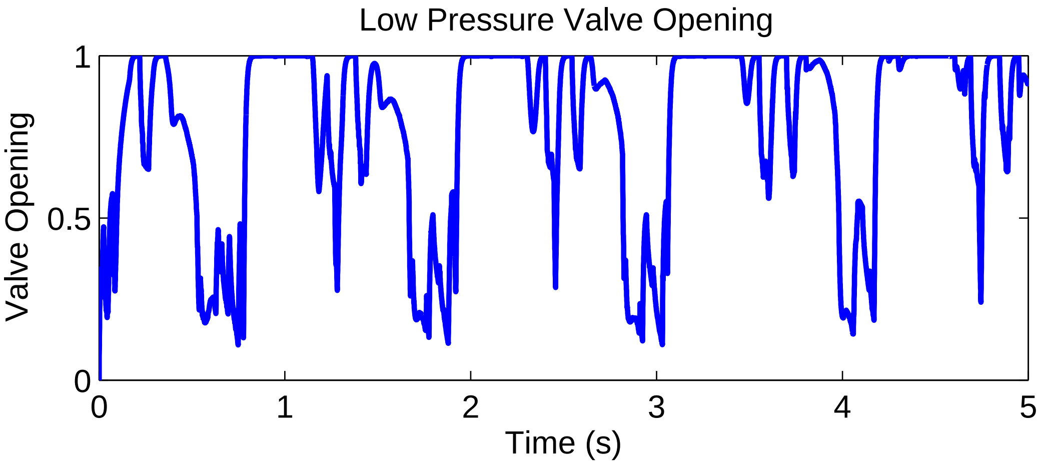

Figure 4.27: Low Pressure Valve Opening vs. Time in Modified Model with Filter

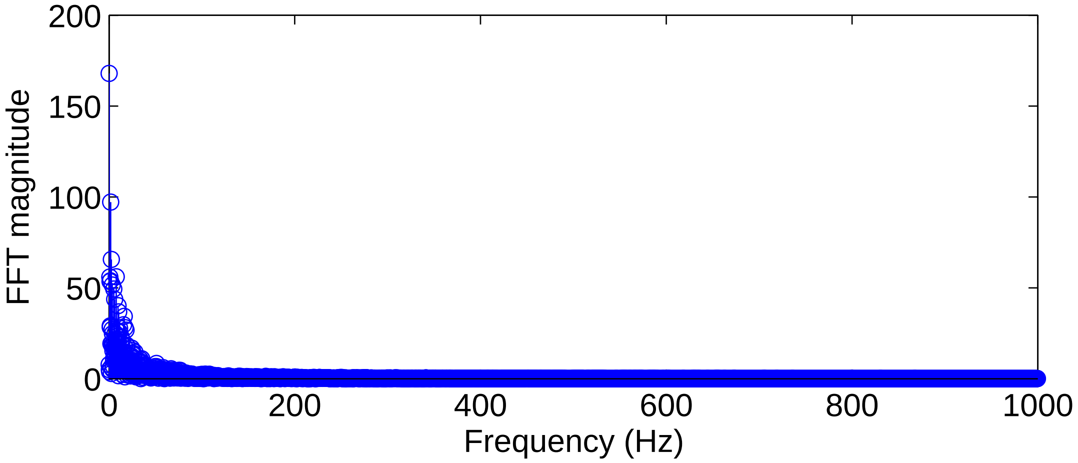

Figure 4.28: High Pressure Valve Signal FFT Magnitude vs. Frequency with Filter

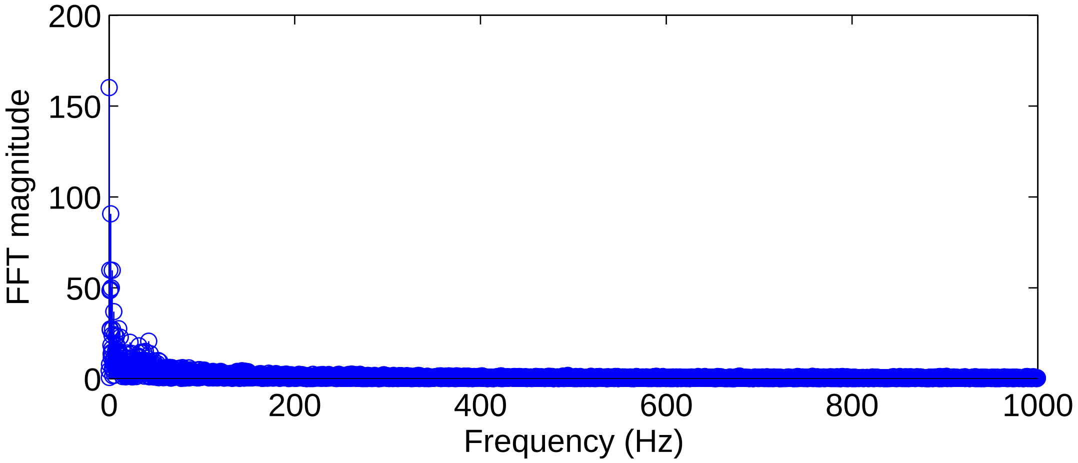

Figure 4.29: High Pressure Valve Signal FFT Magnitude vs. Frequency without Filter

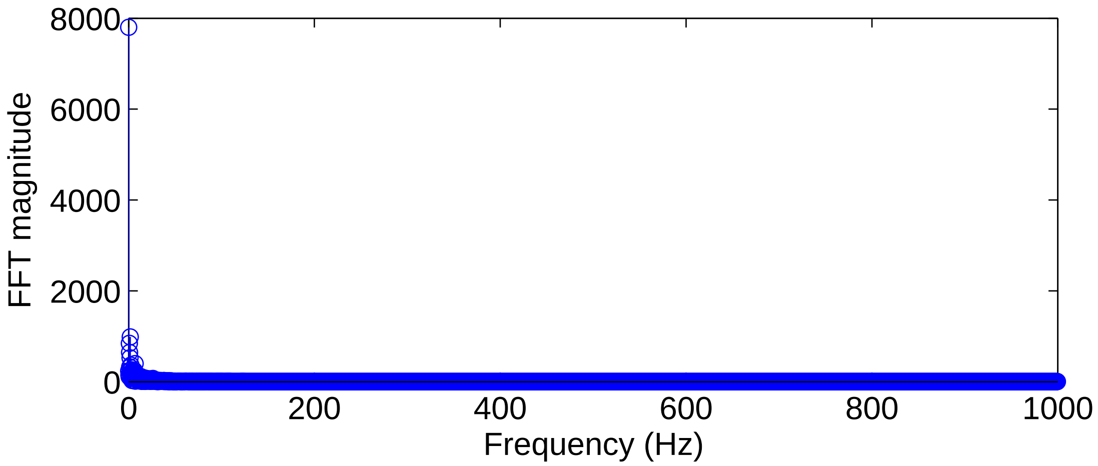

Figure 4.30: Low Pressure Valve Signal FFT Magnitude vs. Frequency with Filter

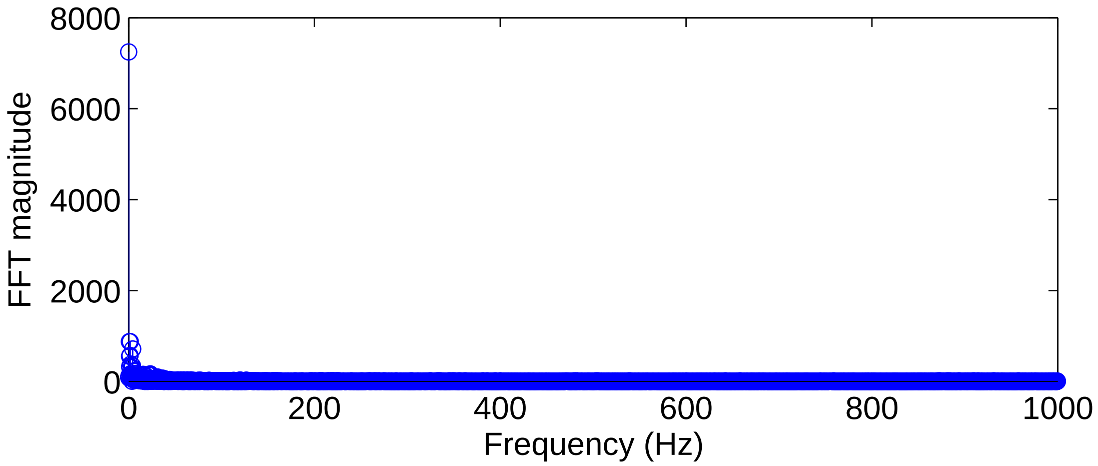

Figure 4.31: Low Pressure Valve Signal FFT Magnitude vs. Frequency without Filter

CHAPTER V

Conclusions and Recommendations

5.1 Conclusions

In this project, a hydraulic knee actuator attached leg prosthesis test robot is modeled and controlled successfully in simulation of normal walk by utilizing the cascade control architecture that the outer loop is controlled by robust passivity-based controller and the inner loop is controlled by optimization method. This process was completed in three steps. First, the RPBC was introduced and validated in direct control of a 3-link prosthesis test robot. Secondly, a novel hydraulic knee actuator was modeled in bond-graph where the dynamic equations of the linear cylinder were developed. An optimization algorithm was then optimized and validated in open-loop control of the hydraulic knee actuator. Lastly, a cascade control architecture was introduced and evaluated in simulations.

This project proves the feasibility of feedback control in prosthesis test robot with hydraulic knee actuator, the cascade control methodology solves the problem effectively. It provides support for the possibility of using a hydraulic actuator to drive a prosthetic knee joint while harvesting energy. The parametric uncertainty issue, which is the intractable element in a robot control system has been resolved successfully by the RPBC. And the valve signal chattering problem has been reduced to an acceptable level by low-pass filters.

The simulation results can be used as important reference for real-time test, however there are some potential challenges need to be overcome. First of all, the volume of high pressure reservoir is chosen as 70 cm^3 which is ten times larger than the original design, it may be a burden for patients wearing this system. Secondly, the switching frequencies of the hydraulic valves in the simulation are relatively high, which may cause troubles to the hydraulic hardware. It may also increase the cost of control, results in poor quality of tracking performance and reduction of the valves lifetime. Finally, the viscous drag of low pressure valve was too small in the simulation which is hardly realistic, therefore finding a corresponding hardware can be difficult.

5.2 Recommendations for Future Work

Although the cascade control of RPBC and optimizer was tested in the prosthesis test robot and acceptable simulation result was achieved, the parameters used in the control system still need to be optimized to find the best motion tracking. Besides, the inner loop hydraulic valve control patterns are generated by a nonlinear programming solver which is not sufficient for the system in real-time, a more advanced and sophisticated optimization algorithm is imperative. In order to better simulate the normal human walking posture, an ankle joint may be added to the robot model. Also, different kinds of human gaits can be tested on the robot to validate the performance of the knee actuator in extended scenarios, for instance, running, a sit-stand-sit cycle or even climbing stairs. If a prototyped model could be achieved, experimental testing of the cascade control methodology may be completed for further validation of the simulations of this work.

BIBLIOGRAPHY

[1] R.L. Waters, J. Perry, D. Antonelli, and H. Hislop. Energy cost of walking of amputees: The influence of level of amputation. *J. Bone Joint Surg. Am.*, 1976.

[2] Akin O. Kapti and M. Sait Yucenur. Design and control of an active artificial knee joint. *Mechanism Machine Theory*, 41:1477–1485, 2006.

[3] K.R. Kaufman, J.A. Levine, R.H. Brey, B.K. Iverson, S.K. McCrady, D.J. Padgett, and M.J. Joyner. Gait and balance of transfemoral amputees using passive mechanical and microprocessor-controlled prosthetic knees. *Gait Posture*, 26:489–493, 2007.

[4] Timothy Wilmot. Intelligent controls for a semi-active hydraulic prosthetic knee. *Cleveland State University, Master's Thesis*, 2011.

[5] Qining Wang, Yan Huang, and Long Wang. Passive dynamic walking with flat feet and ankle compliance. *Robotica*, 28:413–425, 2010.

[6] Edward D. Lemaire, Reza Samadi, Louis Goudreau, and Jonathan Kofman. Mechanical and biomechanical analysis of a linear piston design for angular-velocity-based orthotic control. *Journal of Rehabilitation Research Development*, 50:43–52, 2013.

[7] Jan Andrysek and Gilbert Chau. An electromechanical swing-phase-controlled prosthetic knee joint for conversion of physiological energy to electrical energy: Feasibility study. *IEEE Transactions on Biomedical Engineering*, 54:2276–2283, 2007.

[8] M.S. Orendurff, A.D. Seval, G.K. Klute, M.L. McDowell, J.A. Pecoraro, and J.M. Czerniecki. Gait efficiency using the C-leg. *Journal of Rehabilitation Research Development*, 43(2):239–246, 2006.

[9] J.L. Johansson, D.M. Sherrill, P.O. Riley, P. Bonato, and H. Herr. Comparison of variable-damping and mechanically passive prosthetic knee devices. *Am. J. of Phys. Med. Rehabil.*, 84(8):563–575, 2005.

[10] T Chin, K Machida, S Sawamura, R Shiba, H Oyabu, Y Nagakura, I Takase, and A. Nakagawa. Comparison of different microprocessor controlled knee joints on the energy consumption during walking in transfemoral amputees: Intelligent knee prosthesis (IP) versus C-leg. *Prosthet. Orthot. Int.*, 30(1):73–80, 2006.

[11] A. van den Bogert, S. Samorezov, B. Davis, and W. Smith. Modeling and optimal control of an energy-storing prosthetic knee. *ASME J. Biomechanical Engineering*, 134(5):51–70, 2012.

[12] Investigating the performance of exoprostheses with robot-based 3d-testing methods based on motion analysis. http://www.ipa.fraunhofer.de/fileadmin/Bewegungssysteme/300-380eExoProsthesestests.pdf. Accessed March, 2016.

[13] Cleveland clinic Lerner research institute. Neuromusculoskeletal simulator. http://mds.clevelandclinic.org/Services/BioRobotics/Services.aspx. Accessed March, 2016.

[14] Hanz Richter, Daniel J. Simon, William A. Smith, and Sergey Samorezov. Dynamic modeling and parameter estimation of a leg prosthesis test robot. *Applied Mathematical Modelling*, 39(2):559–573, 2014.

[15] Rick Rarick, Hanz Richter, Antonie van den Bogert, Dan Simon, Holly Warner, and Taylor Barto. Optimal design of a transfemoral prosthesis with energy storage and regeneration. *In Proceedings of the American Control Conference*, pages 4108–4113, 2014.

[16] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling And Control*. John Wiley Sons, 2nd edition, 2006.

[17] K. J. Astrom and Tore Hagglund. PID controller: Theory, design, and tuning. *Instrument Society of America*, 1995.

[18] H. Berghuis and H. Nijmeijer. A passivity approach to controller-observer design for robots. *IEEE Trans. on Robotics and Automation*, 9:740–754, 1993.

[19] J. J. E. Slotine. The robust control of robot manipulators. *Int. J. Robotics Research*, 4(2):49–64, 1985.

[20] J. J. Craig. Adaptive control of mechanical manipulators. *Addison Wesley, Reading, MA*, 1988.

[21] Mark W. Spong and Masayuki Fujita. *Control in Robotics*. http://ieeecss.org/sites/ieeecss.org/files/documents/IoCT-Part1-04Robotics.pdf. Accessed March, 2016.

[22] Robert Gailey, Kerry Allen, Julie Castle, Jennifer Kucharik, and Mariah Roeder. Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *Journal of Rehabilitation Research and Development*, 45(1):15–29, 2008.

APPENDIX A

MATLAB PROGRAMS

Parameters of 3-link Prothesis Test Robot

threelinkrobotParameters.m

```
%Parameter_threelinkrobot




%Treadmill belt speed:
VH=3.3*1600/3600; %in m/s


%Mass of link 1: m1
m1=43.28-5.91*0.454; %in kg, this is the linearly-moving mass of link 1, removing green plate
and screws (subject to gravity)
m0=317.54; %in kg, this is an equivalent inertial mass for rotating components associated
with link 1 (not subject to gravity)
```

```
%Mass of link 2: m2
m2=5.89+5.91*0.454; %in kg, mass of green plate, screws, threaded rod (thigh) and
connecting hardware
%5.89 is for the rod only (ADD two 2 3/4" NUTS)
%5.91 lb is for the green plate and screws

%Mass of link 3: m3
m3=2.29; %in kg, mass below the knee, including Mauch knee, ankle/foot and shoe

%Motor 1 input constant
k1=375; %in N/volt
%Equiv. sliding friction in link 1:
f=83.33; %in N

%Motor 2 inertia and gear ratio:
Jm=1.822e-4; %in kg-m^2
r=80; %gear reduction ratio
%Rotary actuator damping:
b=9.75; %in N-m-s
%Motor 2 average input constant:
k2=15; %in N-m/volt

%CG parameters of link 1:
%d0 and c1y irrelevant to dynamic model

%Dimensional parameters of link 2:
```

```
l2=0.425; %in m, nominal thigh length

c2=-0.339; %in m, calculated in SolidWorks


%Dimensional parameters of link 3:

l3=0.527; %in m, overall length of L3, from knee joint to load cell on shoe

c3=0.32; %in m, distance from knee joint to L3 CG including shoe


%Rotary inertia of link 2:

I2z=0.105+0.33;%in kg-m^2, includes green plate, threaded rod and connecting hardware.


%Rotary inertia of link 3:

I3z=0.0618; %in kg-m^2, overall inertia of L3 with shoe, relative to cm


g=9.81; %acceleration of gravity, m/s^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%The above parameters are sufficient for the evaluation of D(q),

%C(q,\dot{q}) and g(q)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Additional parameters:

%For knee damper:

%damper constants for flexion and extension:


%Damper offset:

od=0.029; %in m, distance between knee joint and damper attachment point on knee plate

rd=0.1905; %in m, distance between attachment points of damper
```

```
%Load cell location:

lcx=0.207; %from origin of frame 3 to load cell along x3 direction CHECK

lcy=0.139; %from origin of frame 3 to load cell along negative y3 direction


TH1= m1+m2+m3;

TH2= m3*l2+m2*l2+m2*c2;

TH3= c3*m3;

TH4= I2z + I3z + Jm*r^2 + c2^2*m2 + c3^2*m3 + l2^2*m2 + l2^2*m3 + 2*c2*l2*m2;

TH5= l2*m3*c3;

TH6= m3*c3^2 + I3z;

TH7= b;

TH8= f;

TH9= m0;

TH_0=[TH1;TH2;TH3;TH4;TH5;TH6;TH7;TH8;TH9];


level=0.1;  %uncertainty level in individual parameters


m0_max=(1+level)*m0;

m1_max=(1+level)*m1;

m2_max=(1+level)*m2;

m3_max=(1+level)*m3;

l2_max=(1+level)*l2;

I2z_max=(1+level)*I2z;

I3z_max=(1+level)*I3z;

c2_max=(1+level)*c2;
```

```
c3_max=(1+level)*c3;

Jm_max=(1+level)*Jm;

b_max=(1+level)*b;

f_max=(1+level)*f;


TH1_max= m1_max+m2_max+m3_max;

TH2_max= m3_max*l2_max+m2_max*l2_max+m2_max*c2_max;

TH3_max= c3_max*m3_max;

TH4_max= I2z_max + I3z_max + Jm_max*r^2 + c2_max^2*m2_max + c3_max^2*m3_max

 + l2_max^2*m2_max + l2_max^2*m3_max + 2*c2_max*l2_max*m2_max;

TH5_max= l2_max*m3_max*c3_max;

TH6_max= m3_max*c3_max^2 + I3z_max;

TH7_max= b_max;

TH8_max= f_max;

TH9_max= m0_max;


TH_MAX=[TH1_max;TH2_max;TH3_max;TH4_max;TH5_max;TH6_max;TH7_max;TH8_max;TH9_max];

DTH=TH_MAX-TH_0;

global rho

rho=norm(DTH);


%Now generate actual perturbations for use in plant

m0_max=(1+(1-2*rand)*level)*m0;

m1_max=(1+(1-2*rand)*level)*m1;

m2_max=(1+(1-2*rand)*level)*m2;

m3_max=(1+(1-2*rand)*level)*m3;
```

```
l2_max=(1+(1-2*rand)*level)*l2;

I2z_max=(1+(1-2*rand)*level)*I2z;

I3z_max=(1+(1-2*rand)*level)*I3z;

c2_max=(1+(1-2*rand)*level)*c2;

c3_max=(1+(1-2*rand)*level)*c3;

Jm_max=(1+(1-2*rand)*level)*Jm;

b_max=(1+(1-2*rand)*level)*b;

f_max=(1+(1-2*rand)*level)*f;


TH1_max= m1_max+m2_max+m3_max;

TH2_max= m3_max*l2_max+m2_max*l2_max+m2_max*c2_max;

TH3_max= c3_max*m3_max;

TH4_max= I2z_max + I3z_max + Jm_max*r^2 + c2_max^2*m2_max + c3_max^2*m3_max

+ l2_max^2*m2_max + l2_max^2*m3_max + 2*c2_max*l2_max*m2_max;

TH5_max= l2_max*m3_max*c3_max;

TH6_max= m3_max*c3_max^2 + I3z_max;

TH7_max= b_max;

TH8_max= f_max;

TH9_max= m0_max;


global TH_CONTROL

TH_CONTROL=[TH1_max;TH2_max;TH3_max;TH4_max;TH5_max;TH6_max;TH7_max;TH8_max;TH9_max];


Z0=[0.01913;1.1317;0.0925;0.09324;0;0]; %initial condition for plant integrator
```

State Equations of 3-link Prothesis Test Robot

statederY.m

```
function out=stateder_swingY(t,z,u,u_1,u_2,s)
%2DOF Hip-simulating robot state derivatives (swing mode)

g=9.81; %acceleration of gravity, m/s^2
%Motor 2 average input constant:
k2=15; %in N-m/volt
%Motor 1 input constant
k1=375; %in N/volt
%Parse inputs
n=length(z);
z_1=z(1:n/2);
z_2=z(n/2+1:n);
q1=z_1(1);q2=z_1(2);q3=z_1(3);
q1dot=z_2(1);q2dot=z_2(2);q3dot=z_2(3);
%Nominal parameters
global TH_CONTROL
TH1=TH_CONTROL(1);
TH2=TH_CONTROL(2);
TH3=TH_CONTROL(3);
TH4=TH_CONTROL(4);
TH5=TH_CONTROL(5);
TH6=TH_CONTROL(6);
TH7=TH_CONTROL(7);
```

```
TH8=TH_CONTROL(8);
TH9=TH_CONTROL(9);

D(1,1)=TH9+TH1;
D(1,2)=TH2*cos(q2)+TH3*cos(q2+q3);
D(1,3)=TH3*cos(q2+q3);
D(2,1)=D(1,2);
D(2,2)=TH4+2*TH5*cos(q3);
D(2,3)=TH6+TH5*cos(q3);
D(3,1)=D(1,3);
D(3,2)=D(2,3);
D(3,3)=TH6;

C(1,1)=0;
C(1,2)=-q2dot*(TH3*sin(q2 + q3)+TH2*sin(q2))-TH3*q3dot*sin(q2 + q3);
C(1,3)=- TH3*q2dot*sin(q2 + q3) - TH3*q3dot*sin(q2 + q3);
C(2,1)=0;
C(2,2)=-TH5*q3dot*sin(q3);
C(2,3)=- TH5*q2dot*sin(q3) - TH5*q3dot*sin(q3);
C(3,1)=0;
C(3,2)=TH5*q2dot*sin(q3);
C(3,3)=0;

gg=[-g*TH1;-g*(TH2*cos(q2)+TH3*cos(q2+q3));-g*TH3*cos(q2+q3)];

%Equiv. sliding friction in link 1:
```

```
% f=83.33; %in N
%Coulomb friction term
N=[TH8*sign(q1dot);0;0];

%Rotary actuator damping:
% b=9.75; %in N-m-s
%Linear damping term
B=[0; TH7*q2dot; 0];

%Calculates the vertical position of the load cell in the world frame
lcx=0.207; %from origin of frame 3 to load cell along x3 direction CHECK
lcy=0.139; %from origin of frame 3 to load cell along negative y3 direction
c3=0.32; %cm location for link 3
l2=0.425;

standoff = 0.935;
LZ=(lcx+c3)*sin(q2+q3)+l2*sin(q2)-lcy*cos(q2+q3)+q1;
if (standoff-LZ) < 0
    GRF = 37000 * (standoff-LZ);
else
    GRF = 0;
end

Jnvt =[ 0, 0, 1;
lcy*cos(q2 + q3) - c3*sin(q2 + q3) - lcx*sin(q2 + q3) - l2*sin(q2), 0, c3*cos(q2 + q3)
+ lcx*cos(q2 + q3) + lcy*sin(q2 + q3) + l2*cos(q2);
lcy*cos(q2 + q3) - c3*sin(q2 + q3) - lcx*sin(q2 + q3), 0, c3*cos(q2 + q3)
+ lcx*cos(q2 + q3) + lcy*sin(q2 + q3)];

VF = [(lcy*cos(q2 + q3) - c3*sin(q2 + q3) - lcx*sin(q2 + q3) - l2*sin(q2))*q2dot
    + (lcy*cos(q2 + q3) - c3*sin(q2 + q3) - lcx*sin(q2 + q3))*q3dot];
%load cell horizontal velocity

VH=3.3*1600/3600; %Treadmill belt speed: in m/s

Fh = -0.15*GRF*sign(VH+VF);%horizontal friction force



Fe = [Fh;0;-GRF];

 stdout = 1; stderr = 2;

   M_knee = NaN;
   v_2 = NaN;

%    valve1proportionalityExpression = ’7.4293*x^4 - 11.877*x^3 + 5.8791*x^2 - 0.4288*x’;
%    valve2proportionalityExpression = ’7.4293*x^4 - 11.877*x^3 + 5.8791*x^2 - 0.4288*x’;
%
%    u_1 = valveProportionalityApproximation(valve1proportionalityExpression,u_1);
%    u_2 = valveProportionalityApproximation(valve2proportionalityExpression,u_2);

   d = 1.905; % (centimeters) Linear cylinder piston diameter.
```

```
H = 19; % (centimeters) Linear cylinder H dimension.

h = 2.8; % (centimeters) Linear cylinder h dimension.

A = pi*d^2/4; % Linear cylinder piston area.

B_1 = 0.127494; %0.127494;

B_2 = 0.001;% 0.127494;

C_1max = 20;%17.9634;

C_2max = 25;%17.9634;

k = 5; %3.66;

P_0 = 0.0;


G = calculateLinearCylinderPressureMomentRatio(A,q3,H,h);


B_1_plus_B_2 = B_1 + B_2;

phi_kp_G = -q3dot / G;

sqrt_eps = sqrt(eps);

u_1_C_1max_sq = u_1^2*C_1max^2;

u_2_C_2max_sq = u_2^2*C_2max^2;

f_1 = B_2*phi_kp_G + k*s - P_0;


if (u_1 < sqrt_eps) && (u_2 > sqrt_eps)

   v_1 = 0;

   v_2 = phi_kp_G;

   M_knee = (P_0 - B_2*v_2 - v_2*abs(v_2)/u_2_C_2max_sq)/G;

   % fprintf(stdout,’%s: Info: Executing null control case 1.\n’,mfilename);

   % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...

   %         phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
```

```
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end

if (u_1 > sqrt_eps) && (u_2 < sqrt_eps)
    v_1 = phi_kp_G;
    v_2 = 0;
    M_knee = (k*s +P_0 - B_1*v_1 - v_1*abs(v_1)/u_1_C_1max_sq )/G;
    %P_0 term above added by H Richter, 1/4/2013

    % fprintf(stdout,’%s: Info: Executing null control case 2.\n’,mfilename);
    % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end

if (u_1 < sqrt_eps) && (u_2 < sqrt_eps)
    v_1 = 0;
    v_2 = 0;
    M_knee = 0; % M_knee can be anything in this case!
    % fprintf(stdout,'%s: Info: Executing null control case 3.\n',mfilename);
    % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %         phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end


if abs(u_1*C_1max - u_2*C_2max) < sqrt_eps
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    % linear: a = 0
    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Case 1: (1,1)

    b = -B_1_plus_B_2 - 2*phi_kp_G/u_2_C_2max_sq;
    c = f_1 + phi_kp_G^2/u_2_C_2max_sq;
    v_1 = -c/b;
    if (v_1 >= 0) && (phi_kp_G - v_1 >= 0)
        [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
        % fprintf(stdout,'%s: Info: Executing linear case 1.\n',mfilename);
        % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
        %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
        return;
    end
    % Case 2: (-1,-1)


    b = -B_1_plus_B_2 + 2*phi_kp_G/u_2_C_2max_sq;
    c = f_1 - phi_kp_G^2/u_2_C_2max_sq;
    v_1 = -c/b;
    if (v_1 < 0) && (phi_kp_G - v_1 < 0)
        [v_2, M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
        % fprintf(stdout,'%s: Info: Executing linear case 2.\n',mfilename);
        % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
        %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
        return;
    end
    % Note cases (1,-1) and (-1,1) are superfluous here.
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
% quadratic
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case 1: (1,1)


a = -1/u_1_C_1max_sq + 1/u_2_C_2max_sq;
b = -B_1_plus_B_2 - 2*phi_kp_G/u_2_C_2max_sq;
c = f_1 + phi_kp_G^2/u_2_C_2max_sq;
v_1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 >= 0) && (phi_kp_G - v_1 >= 0)
   [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
   % fprintf(stdout,'%s: Info: Executing quadratic case 1 sub-case 1.\n',mfilename);
   % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
   %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
   M=[0;0;M_knee];
   u=diag([k1 k2 0])*[u; 0];
   zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
   out=[zdot ;v_1 ;v_2;M_knee;GRF];
   return;
end
v_1 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 >= 0) && (phi_kp_G - v_1 >= 0)
   [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
   % fprintf(stdout,'%s: Info: Executing quadratic case 1 sub-case 2.\n',mfilename);
   % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %         phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end
% Case 2: (1,-1)

a = -1/u_1_C_1max_sq - 1/u_2_C_2max_sq;
b = -B_1_plus_B_2 + 2*phi_kp_G/u_2_C_2max_sq;
c = f_1 - phi_kp_G^2 / u_2_C_2max_sq;
v_1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 >= 0) && (phi_kp_G - v_1 < 0)
    [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
    % fprintf(stdout,'%s: Info: Executing quadratic case 2 sub-case 1.\n',mfilename);
    % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %         phi_kp, s, M_knee, v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end
v_1 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 >= 0) && (phi_kp_G - v_1 < 0)
```

```
    [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
    %  fprintf(stdout,’%s: Info: Executing quadratic case 2 sub-case 2.\n’,mfilename);
    %  verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
end
% Case 3: (-1,1)

a = 1/u_1_C_1max_sq + 1/u_2_C_2max_sq;
b = -B_1_plus_B_2 - 2*phi_kp_G/u_2_C_2max_sq;
c = f_1 + phi_kp_G^2/u_2_C_2max_sq;
v_1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 < 0) && (phi_kp_G - v_1 >= 0)
    [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
    % fprintf(stdout,’%s: Info: Executing quadratic case 3 sub-case 1.\n’,mfilename);
    % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
    %         phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
    M=[0;0;M_knee];
    u=diag([k1 k2 0])*[u; 0];
    zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
    out=[zdot ;v_1 ;v_2;M_knee;GRF];
    return;
```

```
end
v_1 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 < 0) && (phi_kp_G - v_1 >= 0)
   [v_2, M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
   % fprintf(stdout,’%s: Info: Executing quadratic case 3 sub-case 2.\n’,mfilename);
   % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
   %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
   M=[0;0;M_knee];
   u=diag([k1 k2 0])*[u; 0];
   zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
   out=[zdot ;v_1 ;v_2;M_knee;GRF];
   return;
end
% Case 4: (-1,-1)

a = 1/u_1_C_1max_sq - 1/u_2_C_2max_sq;
b = -B_1_plus_B_2 + 2*phi_kp_G/u_2_C_2max_sq;
c = f_1 - phi_kp_G^2/u_2_C_2max_sq;
v_1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
if isreal(v_1) && (v_1 < 0) && (phi_kp_G - v_1 < 0)
   [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
   % fprintf(stdout,’%s: Info: Executing quadratic case 4 sub-case 1.\n’,mfilename);
   % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
   %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
   M=[0;0;M_knee];
   u=diag([k1 k2 0])*[u; 0];
```

```
       zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
       out=[zdot ;v_1 ;v_2;M_knee;GRF];
       return;
    end
    v_1 = (-b - sqrt(b^2 - 4*a*c)) / (2*a);
    if isreal(v_1) && (v_1 < 0) && (phi_kp_G - v_1 < 0)
       [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1);
       % fprintf(stdout,'%s: Info: Executing quadratic case 4 sub-case 2.\n',mfilename);
       % verify(B_1,B_2,C_1max,C_2max,G,k,P_0,...
       %          phi_kp,s,M_knee,v_1,v_2,u_1,u_2);
       M=[0;0;M_knee];
       u=diag([k1 k2 0])*[u; 0];
       zdot=[z_2;inv(D)*((u-B-N-M-Jnvt*Fe)-C*z_2-gg)];
       out=[zdot ;v_1 ;v_2;M_knee;GRF];
       return;
    end


    v_1 = NaN;
    % fprintf(stderr, '%s: Warning: No solution found to the rotary equations.\n',mfilename);
    % keyboard % Stop execution and give the user a chance to debug.


  return;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [v_2,M_knee] = solve_v_2_M_knee(phi_kp_G,G,v_1,u_1_C_1max_sq,k,s,B_1)
```

```
    v_2 = phi_kp_G - v_1;

    M_knee = (k*s-B_1*v_1-v_1*abs(v_1)/u_1_C_1max_sq)/G;

    return;

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function verify(B_1,B_2,C_1max,C_2max,G,k,P_0,phi_kp,s,M_knee,v_1,v_2,u_1,u_2)


    stderr = 2;


    equation_1 = (u_1*C_1max)^2*(k*s - M_knee*G - B_1*v_1) - v_1*abs(v_1);

    equation_2 = (u_2*C_2max)^2*(P_0 - M_knee*G - B_2*v_2) - v_2*abs(v_2);


    equation_3 = phi_kp - G*(v_1 + v_2);

    sqrt_eps = sqrt(eps);

    if (abs(equation_1) > sqrt_eps) || (abs(equation_2) > sqrt_eps) || ...

          ((abs(equation_3) > sqrt_eps) && (abs(v_1) > sqrt_eps) ...

          && (abs(v_2) > sqrt_eps))

        fprintf(stderr, '%s: Error: Verification of a solution to the rotary equations failed.\n'

        ,mfilename);

    end

    return;

end
```