2016-08-01

Position and Stiffness Control of Inflatable Robotic Links Using Rotary Pneumatic Actuation

Charles Mansel Best
Brigham Young University

Position and Stiffness Control of Inflatable Robotic Links Using Rotary Pneumatic Actuation

Charles Mansel Best

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Marc D. Killpack, Chair
Randal W. Beard
Mark B. Colton

Department of Mechanical Engineering

Brigham Young University

July 2016

ABSTRACT

Position and Stiffness Control of Inflatable Robotic Links Using Rotary Pneumatic Actuation

Charles Mansel Best
Department of Mechanical Engineering, BYU
Master of Science

Inflatable robots with pneumatic actuation are naturally lightweight and compliant. Both of these characteristics make a robot of this type better suited for human environments where unintentional impacts will occur. The dynamics of an inflatable robot are complex and dynamic models that explicitly allow variable stiffness control have not been well developed. In this thesis, a dynamic model was developed for an antagonistic, pneumatically actuated joint with inflatable links.

The antagonistic nature of the joint allows for the control of two states, primarily joint position and stiffness. First a model was developed to describe the position states. The model was used with model predictive control (MPC) and linear quadratic control (LQR) to control a single degree of freedom platform to within 3° of a desired angle. Control was extended to multiple degrees of freedom for a pick and place task where the pick was successful ten out of ten times and the place was successful eight out of ten times.

Based on a torque model for the joint which accounts for pressure states that was developed in collaboration with other members of the Robotics and Dynamics Lab at Brigham Young University, the model was extended to account for the joint stiffness. The model accounting for position, stiffness, and pressure states was fit to data collected from the actual joint and stiffness estimation was validated by stiffness measurements.

Using the stiffness model, sliding mode control (SMC) and MPC methods were used to control both stiffness and position simultaneously. Using SMC, the joint stiffness was controlled to within 3 Nm/rad of a desired trajectory at steady state and the position was controlled to within 2° of a desired position trajectory at steady state. Using MPC,the joint stiffness was controlled to within 1 Nm/rad of a desired trajectory at steady state and the position was controlled to within 2° of a desired position trajectory at steady state. Stiffness control was extended to multiple degrees of freedom using MPC where each joint was treated as independent and uncoupled. Controlling stiffness reduced the end effecter deflection by 50% from an applied load when high stiffness (50 Nm/rad) was used rather than low stiffness (35 Nm/rad).

This thesis gives a state space dynamic model for an inflatable, pneumatically actuated joint and shows that the model can be used for accurate and repeatable position and stiffness control with stiffness having a significant effect.

Keywords: inflatable, soft robot, optimal control, contact, pneumatic actuation, model predictive control

ACKNOWLEDGMENTS

I would like to acknowledge and give thanks to my wife, Misti for the support and understanding she has given throughout my time as a student. She has helped significantly by listening to my ideas and challenges and giving good feedback.

I would also like to thank Dr. Marc Killpack as my adviser for the help and support he has given me. Early on, I realized that his ideas and suggestions led to positive results in my research. Dr. Killpack prioritizes his students needs regardless if the need is school related or not. He has spent many hours reviewing my writings, often during his personal time, to ensure that I meet my deadlines.

I would also like to acknowledge my colleagues in the lab, Eric1, Joshua, Morgan, Phil, Levi, Dan, Jon, Eric2, and Vallan, whom with many I have shared publications and have taken classes together. Many of them I have relied on for understanding of difficult subjects and more importantly, they are my friends.

I would like to acknowledge that this work was supported by an Early Career Faculty grant from NASAs Space Technology Research Grants Program. I would also like to acknowledge those at Pneubotics who built the robots used in this thesis and thank them for their help and support when repairs were needed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1.   INTRODUCTION

There has been much interest in making robots more effective in interacting with humans or human environments. Robots currently have limited uses in homes, hospitals, schools, or anywhere people might be. One reason why robots are not common in these places is because robots can be dangerous to people or property when there is an unintentional impact. Clever algorithms and hardware modifications have been developed that reduce impact forces and avoid unintentional contact. However, these methods often rely on high fidelity models with very specific applications. What is wanted is a robotic platform that can freely bump, scrap, or nudge its surroundings without damaging itself or its environment.

An inflatable robot with pneumatic actuation would be an ideal candidate for a more versatile robot that is capable of human environments. The natural compliance and low inertia of the system are ideal characteristics for a human environment friendly robot. These natural characteristics come at the cost of an increase in complexity for control and sensing. This work shows different control methods for an inflatable robot with pneumatic actuation, specifically with the goal to control both position and actuator stiffness. Reasons that simultaneous position control and actuator stiffness control are desirable include that stiffness can be used to dampen out unwanted dynamics, store energy, increase or decrease end effector stiffness, and to minimize contact forces all while performing tasks that require accurate position control. Specific examples are discussed in Section 1.2.

1.1 Contributions

This work shows that both position and stiffness of inflatable structures with pneumatic actuation can be controlled repeatedly and accurately. The following is a list of major contributions in this work.

- Successful position control within 3° of the commanded joint angle using multiple control methods for a single degree of freedom platform and a multiple degree of freedom platform. (Chapter 2)

- Development of a minimal state space dynamic model that can be used to predict future position, stiffness, and pressure states of an antagonistic, pneumatically actuated joint. (Chapter 3)

- Development of sliding mode control laws to track desired position and stiffness trajectories. (Chapter 4.1)

- Development of a model predictive control method to track desired position and stiffness trajectories. (Chapter 4.2)

- Extended the model predictive control method to multiple degrees of freedom and showed end effecter displacement decreased by 50% when stiffness is controlled at 50 Nm/rad rather than 35 Nm/rad showing that active stiffness control significantly effects the system behavior. (Chapter 4.4)

This work presents an improved model and controller for an actuation design utilizing inflatable fabric-based joints for simultaneous position and stiffness control. For a more detailed description of this platform see Section 2.1. This system applies a torque to the joint through expanding chambers pressing upon an soft structure rather than a rigid armature. For this system, the joint and link structure come from inflatable polymer bladders within non-rigid fabrics, resulting in an entirely compliant platform. The model proposed in this work expands on the models developed in [1], [2], and [3] to account for and control joint stiffness. A fabric-based platform of this nature is intrinsically lightweight, compact, and compliant due to the physical properties of the air used for both actuation and structure.

1.2 Variable Stiffness in Current Robotic Platforms

Stiffness is a measure of how resistant to deflection a structure is and compliance is a measure of how much deflection will occur from an external force. By definition stiffness and compliance are inverses of each other but when applied mathematically in models, they give the same result. For example if c is compliance, k is stiffness and x is a variable affected by stiffness then $kx = \frac{1}{c}x$ so the terms stiffness and compliance are used interchangeably throughout this work depending on context.

High compliance or low stiffness allow a system to deviate from its equilibrium point (or where the system will naturally settle) when there are applied external forces. Position controlled systems will continue to apply more force until the desired position is reached which can be damaging to a person or workspace which is in the way. Controlling the stiffness of a system gives the controller a secondary goal which allows for deviation. The amount of deviation will increase with lower stiffness and will decrease with higher stiffness. Systems where the stiffness can be changed through control or actuation are said to have variable stiffness or in other related research, variable stiffness is referred to as variable impedance [4].

Significant research, both in terms of the amount and impact, has been completed in the area of adding compliance to common robotic platforms. A system such as the DLR lightweight robot [5,6] can simulate compliance with an impedance model by utilizing high-rate feedback and low-level torque control [7]. Simulating compliance is mainly limited by control bandwidth and torque capabilities while it can also introduce instabilities in the base controller.

Hardware designs seeking to avoid these limitations have introduced elastic elements between the actuator and load. The most well known method is the use of series elastic actuators (SEA) as described in [8]. SEAs are limited in that the compliance of the actuator is passive and the mechanical passive compliance cannot be changed. Any change in compliance must be done in low-level torque control which is bandwidth and force limited due to the spring.

Designs for actuators that have variable compliance can be seen in [9], [10], and [11]. These designs typically use variable spring lengths, non-linear springs, or pneumatic actuators to affect stiffness. Optimal methods to vary compliance in [12] were shown to maximize impact forces from hammering and in [13] unintentional contact forces with a human were minimized with controlled stiffness.

Variable stiffness actuators can be put into four categories as defined by [9]. They are equilibrium-controlled stiffness or series elastic actuators (SEA), antagonistic-controlled stiffness, structure-controlled stiffness, and mechanically controlled stiffness. SEAs use a spring or other compliant mechanism put in series with the actuator to provide compliance which cannot be physically altered and can only change stiffness through active impedance control. The other three have some method of physically changing the compliant behaviour of the actuator. All four actuation types can be seen in Figures 1.1, 1.2, 1.3, and 1.4. The actuator can be designed to apply a force, displacement, or both.

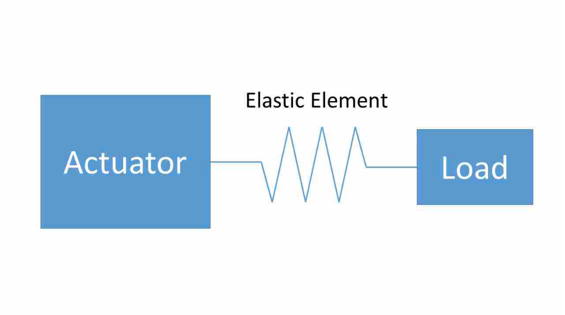

Figure 1.1: Equilibrium-controlled stiffness (SEA)

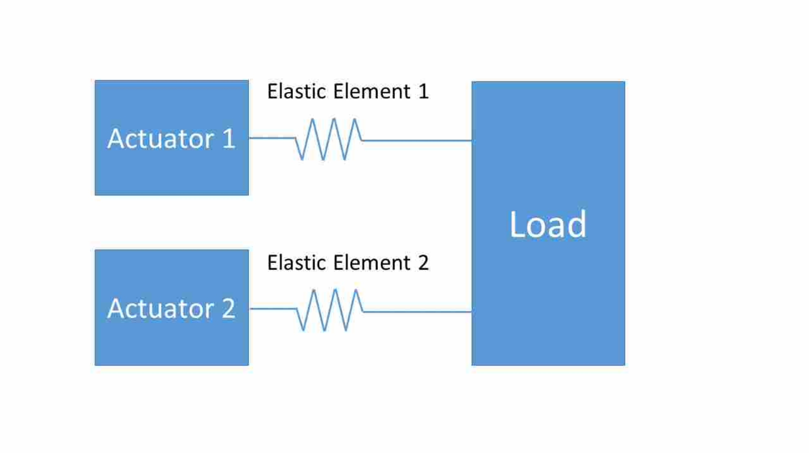

Figure 1.2: Antagonistic-controlled stiffness

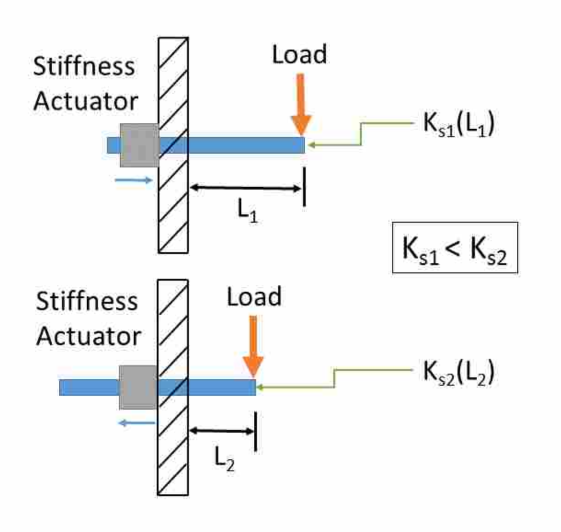

Figure 1.3: This is a structure-controlled stiffness example and by varying the length of the spring L, the overall stiffness changes.

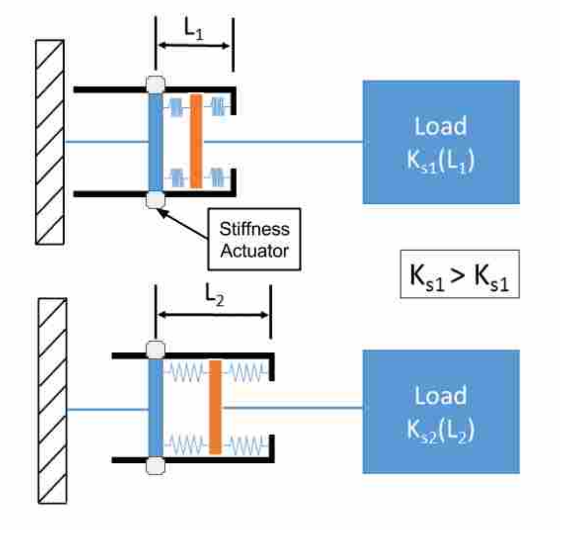

Figure 1.4: This is a mechanically controlled stiffness example and pre-stressing the springs results in higher stiffness. In either configuration, the full length of the spring is used.

The variable compliance actuation method most relevant to the platform used in this work is the antagonistic-controlled stiffness. The two actuators (see Figure 2.1) push independently on the load but the resultant position and stiffness depends on the combined effort of both actuators. For rigid, traditional servo-pneumatic actuators, work in [14] characterized models which made different temperature assumptions. Constant temperature for servo-pneumatic actuators was shown to be a good dynamic approximation at low pressures. In [15] constant temperature assumptions were used to control force and stiffness for a linear pneumatic actuator. Then in [16], two linear pneumatic actuator sets where used for antagonistic impedance control with an actuator in line cable design. What separates the pneumatic bladder design from an actuator in line cable design is that stiffness comes from a compressive force rather than a tensile force. In this thesis, similar control to what was done in [16], was implemented with a single actuator set rather than two, using a rotary actuator instead of linear actuators. One goal of this thesis is to use a modified linear model and optimal control methods to control the rotary joint with an inflatable structure instead of a rigid link and pin joint.

Similar to the actuation type used in this thesis, work has been completed on rotary elastic chamber actuators in [17] and [18] and [19]. In these works two antagonistic bellows impart torque on an armature rotating about a rigid rotary joint. A model for the system is developed and joint position control is achieved for a multiple degree of freedom arm. Stiffness is even described in the model developed in [17] but is not incorporated into control. The platforms used have rigid links and pin joints which is fundamentally different from the inflatable structures used in this thesis. The model development from these works and this thesis start from similar ideas but quickly diverge because of differences in platform structure and the goal of this work to have stiffness control.

A different actuation approach proposed in [20] and [21] utilized McKibben Artificial Muscles, which emulate human muscles. These artificial muscles expand and contract from pressurized fluids. The fluidic properties give these actuators a natural compliance and the construction of these actuators is more similar to the fabric actuators used in this work than the rigid cylinder servo-pneumatic actuators in [15] and [16]. These type of actuators provide a linear force and are used most often in line with a cable for rotational actuation.

1.3 Soft Robot Design

The unique advantages of soft, inflatable robots over rigid robots for specific applications are what motivate this research. Applications include health care, living assistance, space exploration, search and rescue, orthotics, and prosthetics [22–24]. As shown in [22], a variety of soft grippers, actuators, and structures have been designed and built with the intent of moving robotics into environments that are not currently feasible. Not all of the designs use gas as a working fluid or have inflatable components, but all are seeking an alternative approach to a rigid structural design.

Studies have been done to evaluate and quantify the dangers of unintentional impacts from rigid robots. In [13] the Head Injury Criterion (HIC) is identified as a limit for serious head injury upon robot impact, where inertia of the robot is a major driving contributor. This indicates with all operating parameters constant, a reduction in robot inertia, such as for the fabric-based soft robot joint in this thesis, directly reduces the HIC rating. Also in other research, [25] and [26], it has been shown that contact forces from inflatable links can be controlled with cable driven actuators. The ability to control forces can significantly reduce injuries or damage from unintentional impact.

One reason why robots with rigid joints and actuation have been used for so long is that low level system dynamics and sensing methods are well developed. With a soft robotic system, the dynamics are not straightforward and for each design new methods must be developed or learned through data to describe how the system behaves when actuated. An example of this is in [27], [28], and [29] where it was found that planning is viable for elastomer actuators using dynamic and constant curvature kinematic models. Designs for rotary, fabric-based, pneumatic actuated joints were proposed in [30] and the actuators used in this thesis are based on these designs. One purpose of this research is to develop a model and more general method of control for this type of system. The lack of literature on the control of inflatable structures where there is a wide range of applications suggests a viable new area for research.

1.4 Control Methods

In this work, model predictive control (MPC) is the preferred control method, as will be discussed in Chapter 2, and 4. MPC is compared to linear quadratic control (LQR) for position control and then to sliding mode control (SMC) for simultaneous position and stiffness control. One reason for this is to help identify weaknesses of the system model that the controller type might compensate. Using multiple control methods also allows for the performance of each controller to be compared which helps determine what type of controller is best for the system. MPC is not a new control method but is relatively new in its utility in the field of robotics. LQR and

SMC are standard control methods that are common and widely accepted in robotics. Comparing the performance of the different controllers validates the use of MPC and shows when a standard approach would be more appropriate. More detail about these control methods and how they are implemented will be given in the following chapters.

MPC is an optimal control method originating from the process industry in chemical production and oil refinement [31] and more recently in robotics [32–36] and UAV research [37–39]. Recent advances in computing power and optimization techniques such as those proposed in [40, 41] have made this possible. MPC allows for the minimization of customizable cost function across a discrete finite time horizon given specific constraints. MPC not only allows for model-based control but also for the consideration of real-world constraints such as joint limits, pneumatic valve actuator limitations, and hardware pressure limits.

For position control as seen in Section 2.2, the model is linear. A common control method for linear systems is LQR which is similar to MPC. MPC and LQR methods are the same in some cases such as in [42] where constraints were added to the LQR controller with a finite horizon making it MPC. It was seen in [43] that LQR methods are viable options for full body humanoid robots. LQR methods have also been used for humanoid tasks such as balancing [44] and walking [45].

When stiffness is added to the system model in Chapter 4, the stiffness terms are nonlinear. Nonlinear MPC methods are available [46] but solve rates are often several seconds. To increase the control rate to 30 Hz for simultaneous position and stiffness control, the dynamics are linearized and then linear MPC is applied. The performance of MPC is then compared to SMC which is a more conventional nonlinear control method. SMC applies a discontinuous control signal to a system to force the dynamics to slide along a subset of the full dynamics of the system. One advantage of using SMC is that system parameters do not have to be explicitly known in some cases. Instead bounds on these parameters are used to form the control law. This is benificial for soft robots with complicated dynamics where parameters are not always constant and can be configuration dependent. Methods for using SMC in robotics have been well developed for decades [47–49]. The method used in this work for the development of a sliding mode control law comes from the methods described in [50] and [51].

CHAPTER 2. MODEL DEVELOPMENT AND CONTROL WITHOUT STIFFNESS

In this Chapter, first a description of the platforms used for this thesis is given. Then a two state (angle and angular velocity) model and control method is developed for an antagonistic, pneumatic actuated, rotary joint. Model predictive control (MPC) and Linear Quadratic Control (LQR) were applied for position control for a single degree of freedom and multiple degrees of freedom within 3° of a commanded angle. For multiple degrees of freedom, a pick and place task was performed where the pick is successful ten out of ten times and the place was successful eight out of ten times. Next, how the model was extended from two states to four states to include the pressures in each actuator in [1] is explained. MPC was applied and the authors of [1] show the average rise time was reduced from 1.3 seconds to .6 seconds, the average settling time was reduced from 3.0 seconds to 1.4 seconds, and the average overshoot was reduced from 24.6% to 2.6%. The cost function for MPC was modified to allow for a pressure target in each actuator and stiffness was implicitly controlled. Finally, the work done in [3] for the development of a torque model using the two pressure states is explained. In [3], MPC using the torque model was applied to a single degree of freedom and when compared to the performance of the two state controller, average rise time increased slightly by 3% but average settling time was reduced by 15.1% and the average overshoot was reduced from 24.4% to 2.6%. Also in [3], when control is extended to a multiple degree of freedom arm, end effector position was repeatable within 3 cm and for a pick and place task the pick was successful ten out of ten times and the place was successful nine out of ten times.

The author of this thesis was the primary researcher in [2] and the results are presented in this thesis. The work in [1, 3] was collaborative, and the author of this thesis's main role in [1] was model development and in [3] was fitting the torque model to data and extending to multiple degrees of freedom. The background from these works is necessary to motivate the work done in Chapters 3 and 4.

2.1 Platform Description

The platforms used for this research are a single degree of freedom joint called a grub (Figure 2.1) and a fourteen degree of freedom humanoid robot called King Louie (Figure 2.2). Both were developed and built by Pneubotics, an affiliate of Otherlab. Besides internal electronics such as IMU and pressure sensors, these platforms are entirely made of ballistic nylon fabric with internal bladders to prevent air leakage. The structure of these robots comes from an inflatable bladder which is pressurized to between 1-2 PSI gage. At each joint, there are antagonistic actuators which can be filled to pressures between 0-25 PSI gage. For this thesis, pressures between 0-17 PSI gage are used because of pinching effects in the main chamber at higher pressures. These actuators are similar to the designs mentioned in [30].

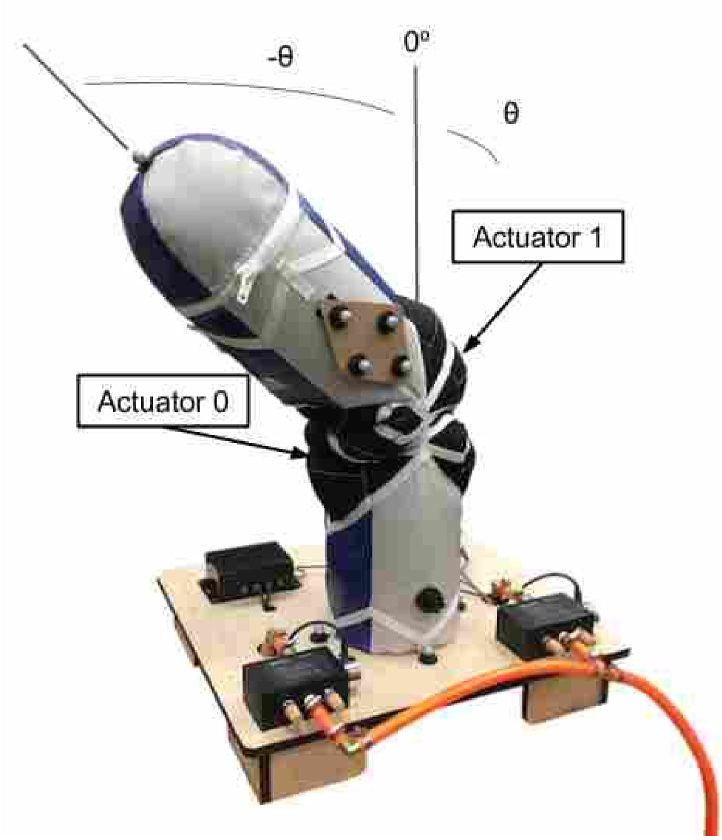

Figure 2.1: This is a single degree of freedom platform called a grub.

Air flow is controlled from the pressure source and vented to atmosphere through Enfield LS-v25 five port spool valves. For the purpose of the paper, only one output port of the Enfield

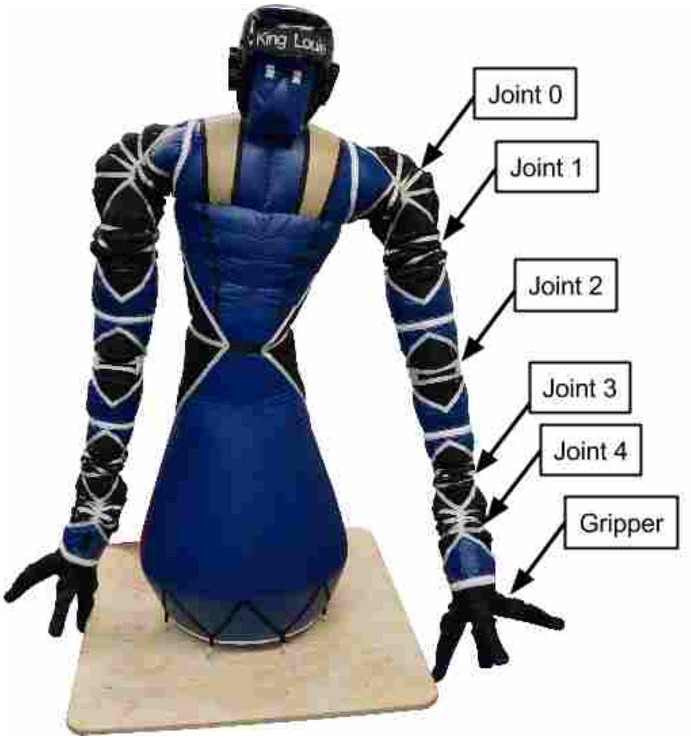

Figure 2.2: Inflatable humanoid robot named King Louie

valves is used, making these valves act as three port spool proportional flow valves. As seen in Figure 2.3, each actuator or bladder has an individual valve for control, while sharing the same pressure source.

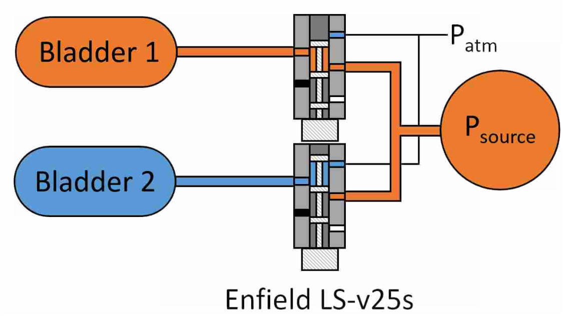

Figure 2.3: Representative figure of valve and bladder configuration (used with permission from [1])

Communication between the sensors and controller is achieved using ROS (Robot Operating System) operating in nonreal time on an Ubuntu workstation as seen in Figure 2.4. Valves are controlled from a low level ROS node with a high thread priority effectively making the process soft real time. Sensors and driver modules are run in real time from embedded hardware. Sensor values for the IMU and pressure are read and published at approximately 1 kHz. The IMUs used are InvenSense MPU-9150, containing 3-axis accelerometer, gyro, and magnetometer sensors and the pressure sensors are Honeywell HSCDRNN100PGSA3 with a 0 to 100 PSI range. The motion capture system uses eight Kestrel digital cameras which operate up to 300 Hz and the data is processed by MotionAnalysis's Cortex software. Joint angles are measured on King Louie by using a motion capture system and the joint angle for the grub is measured with an inertial measurement unit IMU located on the link. King Louie has IMU sensors on each link and motion capture data was purposefully used in a way such that it can be directly replaced by noisier IMU data in future work. Pressure for each actuator is controlled by an underlying PID controller also operating at approximately 1 kHz. Desired pressures are published over ROS at the rate the controller is operating and the valves are actuated to achieve commanded pressure values.

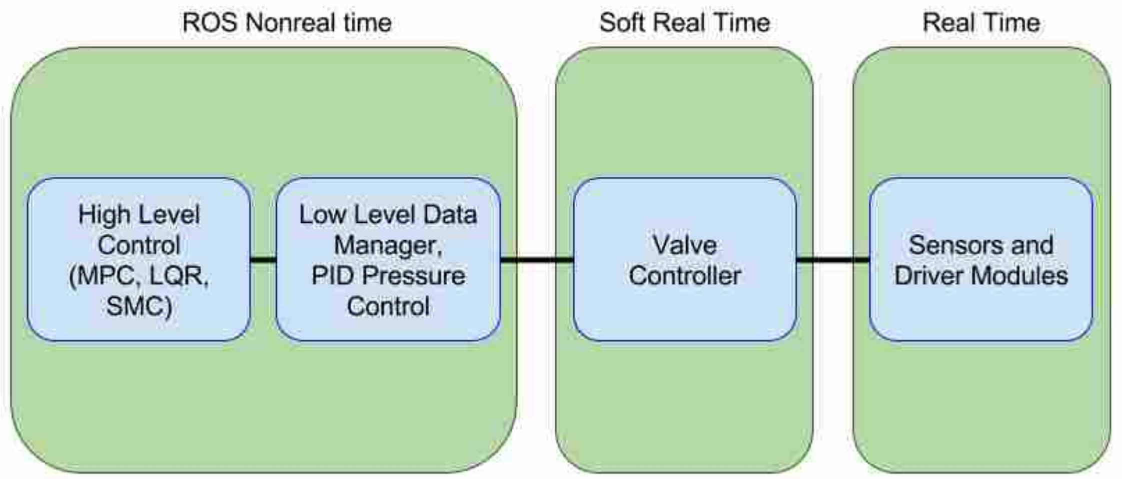
ROS Nonreal time
Soft Real Time
Real Time
High Level Control (MPC, LQR, SMC)
Low Level Data Manager, PID Pressure Control
Valve Controller
Sensors and Driver Modules

Figure 2.4: System chart showing nonreal time, soft real time, and real time operating system sections

2.2 Two State Controller

For single degree of freedom control development, both MPC and LQR were used where both controllers used the same dynamic model. The form of these two controllers is described in the next sections. A block diagram for the system can be seen in Figure 2.5. The controller sends inputs in the form of θ_d which is the desired angle based on an impedance model for torque. These angles are converted into pressure commands $\mathbf{P}_{des}$ where $\mathbf{P}_{des} = [P_{0,des}, P_{1,des}]^T$. A PID pressure controller then adjusts actuator pressures to track the pressure commands for each actuator by manipulating electrical currents sent to spool proportional flow valves which adjusts air flow. A Kalman filter is used to estimate joint angle and angular velocity from inertial measurement unit signals. The formulation and results for this controller were published in [2].

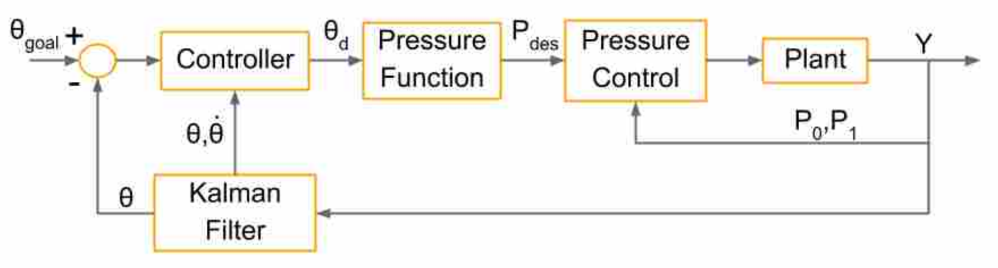

Figure 2.5: Control system block diagram for complete system

2.2.1 Model Development

Initially the grub was modeled as an inverted pendulum with states $\dot{\theta}$ and θ which are angular velocity and angle respectively defined as annotated in Figure 2.1. The differential equation that was used to describe the motion of a single link as an inverted pendulum is

$$I\ddot{\theta} + mg\frac{L}{2}sin(\theta) = \tau_a, \tag{2.1}$$

where I is the moment of inertia of the link about its joint, m is the mass of the link, g is the gravity constant, and L is the length of the link. The applied torque for the impedance control model is

$$\tau_a = k_s(\theta_d - \theta) - k_d\dot{\theta}, \tag{2.2}$$

where k_d is a damping coefficient, k_s is the stiffness coefficient and θ_d is the system input. This equation for torque is a standard impedance model for a rotational actuator. An impedance model acts as if a virtual spring with stiffness k_s and damper with a viscous friction k_d is placed between the link and a virtual link at an angle θ_d. As the virtual link is moved or as θ_d changes, a virtual torque will be applied to the actual link causing it to move with dynamics related to k_d and k_s. If the virtual torque is applied as a control input to the joint through a rotary actuator, the actual system dynamics will behave like the virtual system.

The input θ_d is assumed to be the steady state response of the impedance model but the actual inputs to the system are pressures in the actuators. For a conventional robotic actuator, θ_d would be determined by a controller and then used by Equation 2.2 to calculate a torque which would then be sent to a motor controller to apply the necessary torque. The challenge for pneumatic actuation is relating actuator pressures to torques. Instead of controlling the system in terms of torque, a direct relationship between θ_d and actuator pressures was found. If θ_d is the steady state response of the impedance model then θ_d can be related to actuator pressures by using the steady state pressure response.

Pressure Dynamics and Steady State Angle Response

The first goal was to identify how the pressure dynamics behaved and how pressures in the actuators relate to the joint angle. Figure 2.6 shows several step responses from the pressure controller to desired pressure commands. The shape of the pressure curve shows a similar response to a first order pressure model except there is odd behavior near 16 PSI. It is possible that this is where flow shifts from choked flow to unchoked flow. The sharp transition from filling to steady state just before reaching the goal suggests higher order pressure dynamics but draining still appears to be first order. The response is expected to be faster for higher source pressures.

The steady state joint angle response for a step input of pressures in both chambers on a single joint shows hysteresis and depends on the initial pressure and angle states. The angle response to the pressure dynamics shown in Figure 2.6 can be seen in Figure 2.7. As P_0 increases in Figure 2.6 the joint angle increases as seen in Figure 2.7. When P_0 decreases, the actuators relax and the joint moves back towards its original position. In Figure 2.7 the angle response is constant when there is a steady state pressure response in Figure 2.6.

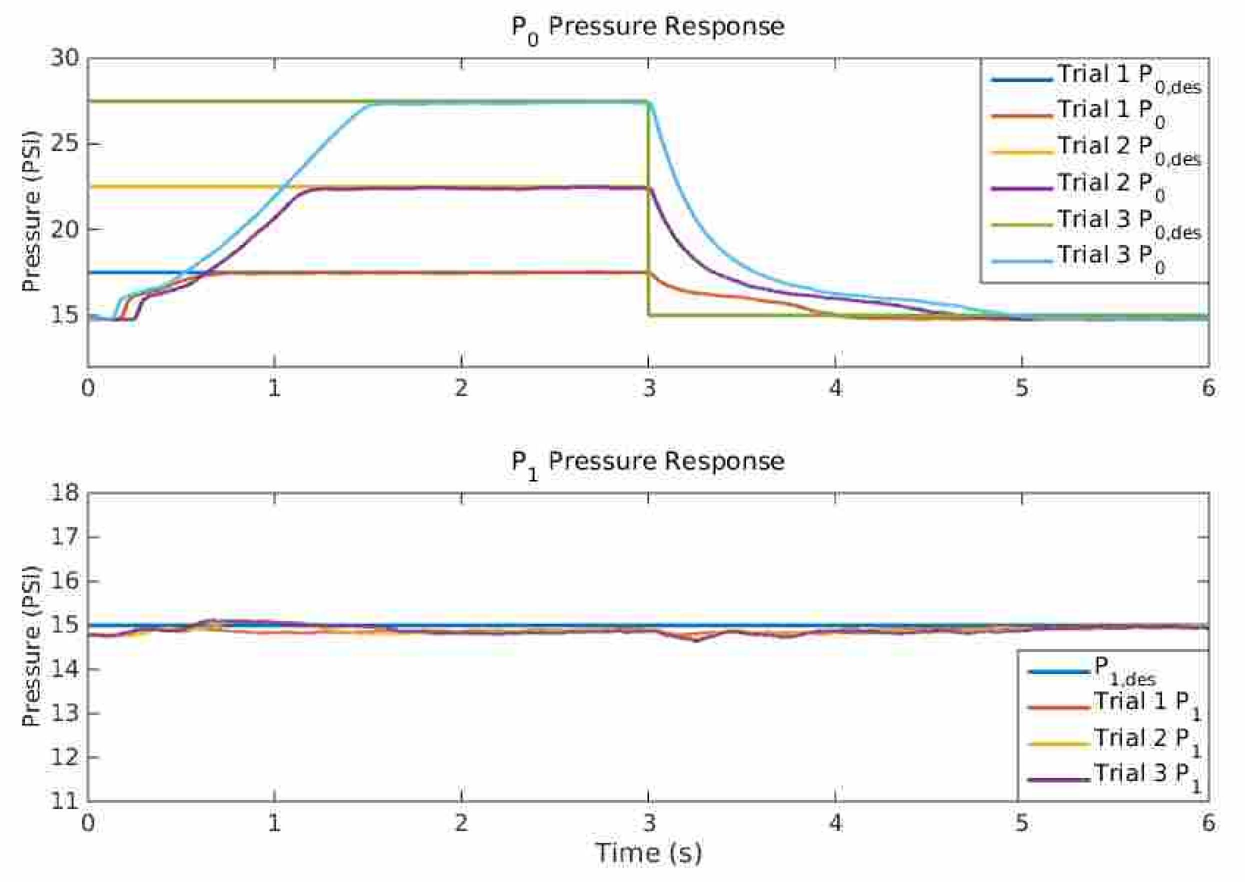

Figure 2.6: Plot showing several responses from desired pressures using the pressure controller

A sample mapping of the steady state response for angle dependency on commanded pressures can be seen in Figure 2.8. This mapping was produced using the same initial conditions which included atmospheric pressure in both actuators for each of the vertices on the P_0 and P_1 grid. The shape of the map distorts due to hysteresis when there are different initial conditions such as switching directly from one pressure set to the next without first draining but the trend stays the same. The map shown in Figure 2.8 contains hysteresis affected by how the pressure states were iterated. The steady state response can vary from the average response by as much as 80° from what is seen in Figure 2.8 due to hysteresis. Hysteresis will be dealt with when considering a single line along the pressure mapping as will be explained later. For this mapping, an infinite number of pressure combinations map to a single position output which requires a constraint on the inputs for control.

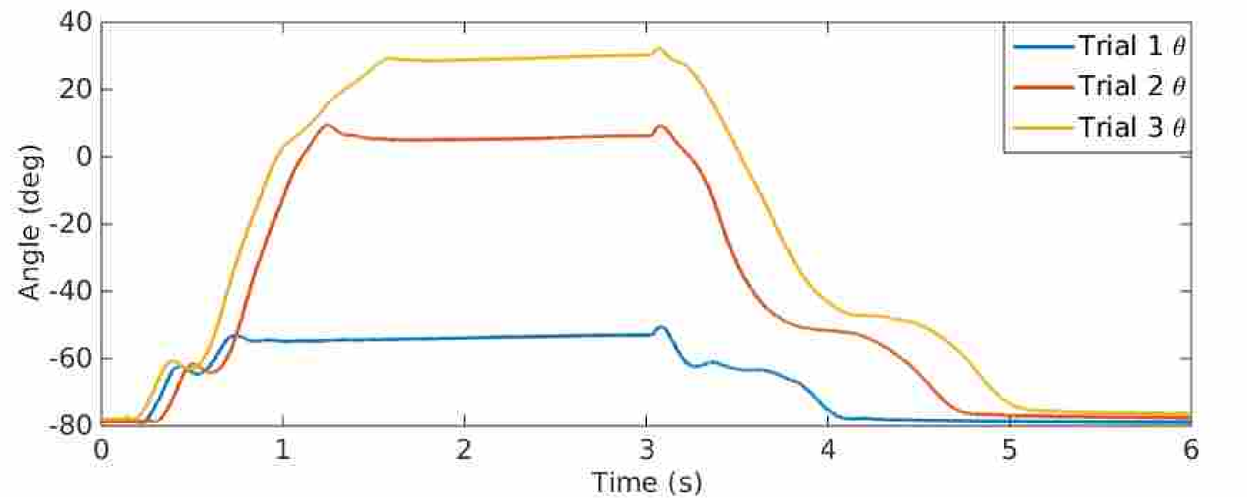

Figure 2.7: Plot showing the angle response to the pressure commands shown in Figure 2.6

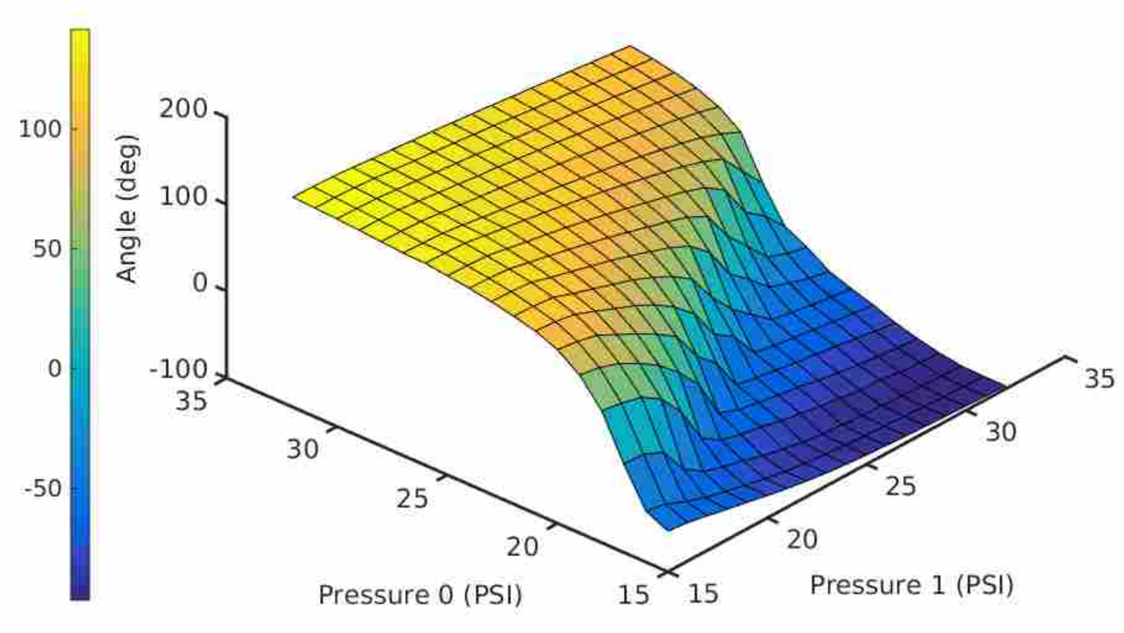

Figure 2.8: This is a sample plot of resultant steady state angle given specific initial conditions and two input pressures. Pressure 0 corresponds to the pressure in actuator 0 and pressure 1 corresponds to the pressure in actuator 1.

To initially reduce the system to a single input single output system, the inputs were constrained from two dimensions to one dimension. This in effect is a slice or a line on the pressure to angle mapping in Figure 2.8. A single pressure profile P was used to achieve a joint angle θ.

If P_0 is the pressure in one actuation chamber in absolute PSI and P_1 is the pressure in the other chamber also in absolute PSI, letting

$$P_0 = 14 + P \tag{2.3}$$

and

$$P_1 = 33 - P, \tag{2.4}$$

with the constraint

$$1 \leq P \leq 18, \tag{2.5}$$

the two pressure inputs P_0 and P_1 are combined into a single pressure input P which reduces the controller from multi-input to single input. In Equation 2.3, the numbers 14 was determined based on atmospheric pressure, and in 2.4 the number 33 was determined based on a maximum absolute pressure where pinching effects in link structure were minimal. The bounds on P in Equation 2.5 were determined to keep P_0 and P_1 within 15 and 32 PSI. In Section 2.3 and Chapter 4, it is shown how to use the full pressure set to account for stiffness and position making a multi-input multi-output (MIMO) system.

Figure 2.9 shows the average steady state angle output of a single joint for different P values collected with different initial conditions where hysteresis can be seen. The difference between how the data over the average curve and under the average curve was collected was significant. The reason why data was found over or under the average line is because of the direction the pressures were incremented while data was collected. The data over the line was collected when P started at 1 and incremented towards 18. For the data below the average line, P started at 18 and decremented towards 1. The difference between the data sets on the same side of the average curve is dependent on the initial conditions which was whether or not the actuators were allowed to drain between iterations.

The average of the data in Figure 2.9 can be used to map a desired angle such as θ_d to desired pressures which will then be sent to the pressure controller. The average is used because the data was collected in a way that would maximize hysteresis in one direction or the other. The

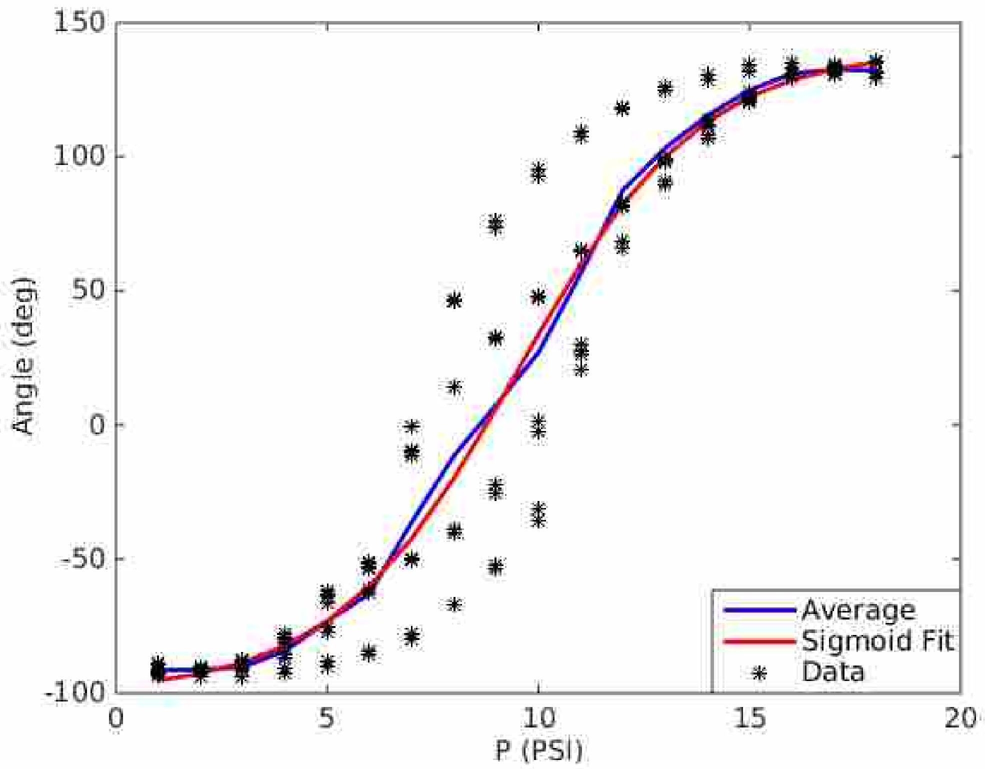

Figure 2.9: This is a sample pressure profile for a single variable *P*.

average then mitigates the effects from hysteresis. The input-output relationship in Figure 2.9 closely resembles a sigmoid function. The equation

$$\theta_d = \frac{s_2}{1+e^{(-s_1(P-s_4))}} + s_3 \tag{2.6}$$

is the general form of the sigmoid function used to map pressure inputs to input angles. The coefficient s_1 determines the slope of the curve and was fit to match the slope of the average curve in Figure 2.9. The coefficients s_2, s_3, and s_4 are determined by joint limits. The red line in 2.9 shows the sigmoid function fit to the average data line shown in blue. For position control in this chapter, θ designates the joint angle and θ_d represents the system input. The system input θ_d is a function of pressures that maps the pressure states to an angle which is necessary for the units in the simplified impedance model to be consistent.

Equation 2.6 is needed to map the system input directly to value for P to be used in Equations 2.3 and 2.4. In Equation 2.6, P can either be solved for analytically or by using an optimization. An optimization was used to avoid the potential case where there is a natural logarithm of a negative number. The algorithm used is sequential least squares programming which is useful because the bounds on P can be set. Although this method simplifies the control input problem, the constraint on commanded pressure in the two joint chambers also limits system capabilities. The ability to use the full pressure range for each bladder could result in faster system response, better dynamic behavior, and the ability to have variable stiffness at each joint.

State Space Model

For joint position control, several rigid body assumptions were made to simplify the model for the soft robot system such as there being no lateral or torsional deflection along the links. Deviations from this approximate model will degrade overall system performance, but they are treated as disturbances on the system. Future work will include aspects within the model that capture dynamics unique to an inflatable robot. However, the results of LQR and MPC for the system presented here show that a rigid body model is sufficient for basic control of the system.

Substituting in the torque term and solving for $\ddot{\theta}$ in Equation 2.1 gives

$$\ddot{\theta} = \frac{-mg\frac{L}{2}sin(\theta)}{I} + \frac{k_s(\theta_d - \theta)}{I} - \frac{k_d\dot{\theta}}{I}. \tag{2.7}$$

The damping and stiffness terms were estimated directly by applying a step response in desired pressure to the real system and fitting the model Equation 2.7 in terms of $\frac{k_d}{I}$, $\frac{k_s}{I}$, and m. The link mass m was found to be three orders of magnitude smaller than what the system identification found for k_s and k_d so the mass terms were assumed to be negligible in the model. This reduces Equation 2.7 to

$$\ddot{\theta} = \frac{k_s(\theta_d - \theta)}{I} - \frac{k_d\dot{\theta}}{I} \tag{2.8}$$

which is linear. The model can be represented in state space form as

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} + \mathbf{B}\theta_d \tag{2.9}$$

$$\mathbf{Y} = \mathbf{C}\begin{bmatrix}\dot{\theta}\\ \theta\end{bmatrix}, \tag{2.10}$$

where $\mathbf{Y}$ is the measured states and

$$\mathbf{A} = \begin{bmatrix}\frac{-k_d}{I} & \frac{-k_s}{I}\\ 1 & 0\end{bmatrix} \tag{2.11}$$

$$\mathbf{B} = \begin{bmatrix}\frac{k_s}{I}\\ 0\end{bmatrix} \tag{2.12}$$

$$\mathbf{C} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}. \tag{2.13}$$

Adding a more significant link-side load to a single joint (or a multi-joint robot), would require re-examining this approximation.

With Δt as the time step, the state space form was transformed from the continuous time-domain to the discrete domain using the matrix exponential method found in [52], where

$$\mathbf{A}_d = e^{\mathbf{A}\Delta t} \tag{2.14}$$

and

$$\mathbf{B}_d = (\mathbf{A}_d - \mathbf{I})\,\mathbf{A}^{-1}\mathbf{B} \tag{2.15}$$

so that the complete discrete time state space equation is

$$\begin{bmatrix}\dot{\theta}[t+1]\\ \theta[t+1]\end{bmatrix} = \mathbf{A}_d\begin{bmatrix}\dot{\theta}[t]\\ \theta[t]\end{bmatrix} + \mathbf{B}_d\theta_d[t], \tag{2.16}$$

where t is the discrete step index.

A single IMU is placed on the link of the grub to estimate the joint angle. With the grub oriented upwards, the joint angle can be estimated from the measurements of two perpendicular accelerometers measuring the gravity vector. This is can be calculated using trigonometry as follows:

$$\theta = atan\left(\frac{a_y}{a_x}\right) \tag{2.17}$$

where a_y is the accelerometer axis measurement parallel to the gravity vector and a_x is the accelerometer axis measurement perpendicular to the gravity vector and tangent to the joint rotational degree of freedom while the grub is oriented at 0°. This method for estimating joint angles is accurate at steady state but during actuation there are more accelerations acting on the link than gravity. A Kalman filter using both accelerometer and gyro data was used during actuation to produce a smoothed state estimate.

2.2.2 MPC

The model predictive controller uses the discrete model Equation (2.16) to predict future states across the horizon T. The system is simulated over the horizon where the inputs θ_d are picked in such a way that the cost function is minimized. The cost function minimized across the horizon is

$$\begin{aligned} f(\theta,\dot{\theta},\theta_d) &= \sum_{t=1}^{T}\left(\|\theta\,[t]-\theta_{goal}\|_{Q_1}^2+\|\theta_d\,[t]\|_{R_1}^2+\|\Delta\theta_d[t]\|_{R_2}^2\right) \\ &\quad +\|\theta[T]-\theta_{goal}\|_{Q_2}^2+\|\dot{\theta}[T]\|_{Q_3}^2, \end{aligned} \tag{2.18}$$

where θ_{goal} is the angle that the controller is trying to drive the system towards and Q_i, $i=1..3$ and R_i, $i=1,2$ are cost weights. The cost function minimization is subject to:

$$\begin{bmatrix}\dot{\theta}[t+1]\\ \theta[t+1]\end{bmatrix} = \mathbf{A}_d\begin{bmatrix}\dot{\theta}[t]\\ \theta[t]\end{bmatrix} + \mathbf{B}_d\theta_d[t] \qquad t=1..T \tag{2.19}$$

$$\theta_{min} \le \theta \le \theta_{max} \qquad t=1..T \tag{2.20}$$

$$\Delta\theta_d[t] = \theta_d[t]-\theta_d[t-1] \qquad t=1..T \tag{2.21}$$

$$|\Delta\theta_d| \leq \Delta\theta_{d,max} \qquad t = 1..T. \tag{2.22}$$

The limit $\Delta\theta_{d,max}$ was determined by the slope of the system response to a step input. From saturating the valve command and recording the angle data the slope can be calculated by fitting a line to two points along the linear region of the angle data. This limit is necessary because for the two state controller, pressure dynamics were not part of the system model. For Equation 2.21, when $t = 0$, $\theta_d[0]$ is the control input from the previous time step.

This optimization can be solved at rates up to 200 Hz with a horizon of $T = 65$ time steps using C code generated by the online convex optimization code generation tool CVXGEN [53]. CVXGEN has limits on the Karush-Kuhn-Tucker conditions for which it can provide optimized code. The horizon length was picked so that Karush-Kuhn-Tucker conditions were the most CVXGEN could handle which was about 7000. The first input from the optimized input trajectory is applied as control to the actual system and the optimization is then reformulated and runs again at the next time step with updated states from system measurements. With a fixed horizon of 65 time steps, fast control rates can be traded for a longer time horizon allowing the controller to look at future states further out making the control rate a tunable parameter. In Equation (2.18), Q_1 weights the error over the horizon, Q_2 weights the final error, Q_3 weights the the final velocity, R_1 weights the inputs over the horizon, R_2 weights the change in input over the horizon and all were tuned manually for acceptable performance. The weights for final error and final velocity add a large cost at the final step in the horizon if the system has not reached the goal. Controller performance is discussed in Section 2.2.4.

2.2.3 LQR Control

For position control using LQR, the discrete state space model from Equation (2.16) was used. A new discrete state $v[t]$, defined as

$$v[t+1] = v[t] + \left(\theta_{goal}[t+1] - \theta[t+1]\right) \tag{2.23}$$

was added to the system where θ_{goal} is the commanded joint angle where the controller is to drive the system. This was added as an discrete form of an integrator, using the method described in [52], to the system because there was significant steady state error using LQR. The new discrete state space model then becomes

$$\begin{bmatrix}\dot{\theta}[t+1]\\ \theta[t+1]\\ v[t+1]\end{bmatrix} = \mathbf{A}_i \begin{bmatrix}\dot{\theta}[t]\\ \theta[t]\\ v[t]\end{bmatrix} + \mathbf{B}_i\theta_d[t] \tag{2.24}$$

where

$$\mathbf{C}_d = \begin{bmatrix}0 & 1\end{bmatrix} \tag{2.25}$$

$$\mathbf{A}_i = \begin{bmatrix}\mathbf{A}_d & \mathbf{0}\\ -\mathbf{C}_d\mathbf{A}_d & 1\end{bmatrix} \tag{2.26}$$

$$\mathbf{B}_i = \begin{bmatrix}\mathbf{B}_d\\ -\mathbf{C}_d\mathbf{B}_d\end{bmatrix}. \tag{2.27}$$

In preliminary testing, it was seen that without a limit on the input θ_d, the controller would command large changes in inputs that would instantly saturate the pressure controller in both directions because the pressure dynamics are not accounted for in the model. In order to limit the system input, θ_d was made a state as defined by

$$\theta_d[t+1] = \theta_d[t] + \Delta\theta_d[t], \tag{2.28}$$

where $\Delta\theta_d[t]$ was the new system input. The discrete state space model for the LQR controller then becomes

$$\mathbf{X}[t+1] = \mathbf{A}_{lqr}\mathbf{X}[t] + \mathbf{B}_{lqr}\Delta\theta_d[t], \tag{2.29}$$

where

$$\mathbf{X} = \begin{bmatrix} \dot{\theta}[t] & \theta[t] & v[t] & \theta_d[t] \end{bmatrix}^T \tag{2.30}$$

$$\mathbf{A}_{lqr} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.31}$$

$$\mathbf{B}_{lqr} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T . \tag{2.32}$$

The control input $\Delta\theta_d[t]$ was found at each step using the LQR formulation

$$\Delta\theta_d[t] = -\mathbf{KX}, \tag{2.33}$$

where $\mathbf{K}$ is found by using the solution to the discrete algebraic Riccati equation using $\mathbf{A}_{lqr}$, $\mathbf{B}_{lqr}$, $\mathbf{Q}$, and R where $\mathbf{Q}$ is diagonal positive semi-definite and contains the weights on the states and R is the weight on the input. Both $\mathbf{Q}$ and R were tuned manually so the joint could achieve the commanded angles.

2.2.4 Results and Discussion for Single Joint

The MPC response, as well as the LQR responses with and without an integrator can be seen in Figure 2.10 for a single joint. The MPC response shows less overshoot and better tracking of the goal than the LQR response which for some inputs relies on the integrator for accurate tracking. However the LQR response is faster and there is less oscillation around the goal. The LQR response shows that the system model can be used for acceptable performance. Two advantages that MPC has over LQR is the ability to add constraints, and add different terms to the cost function without major changes to the system model. For example, to limit the system input, a simple constraint was added to the MPC optimization where LQR required a new state and cost to be added to the system model.

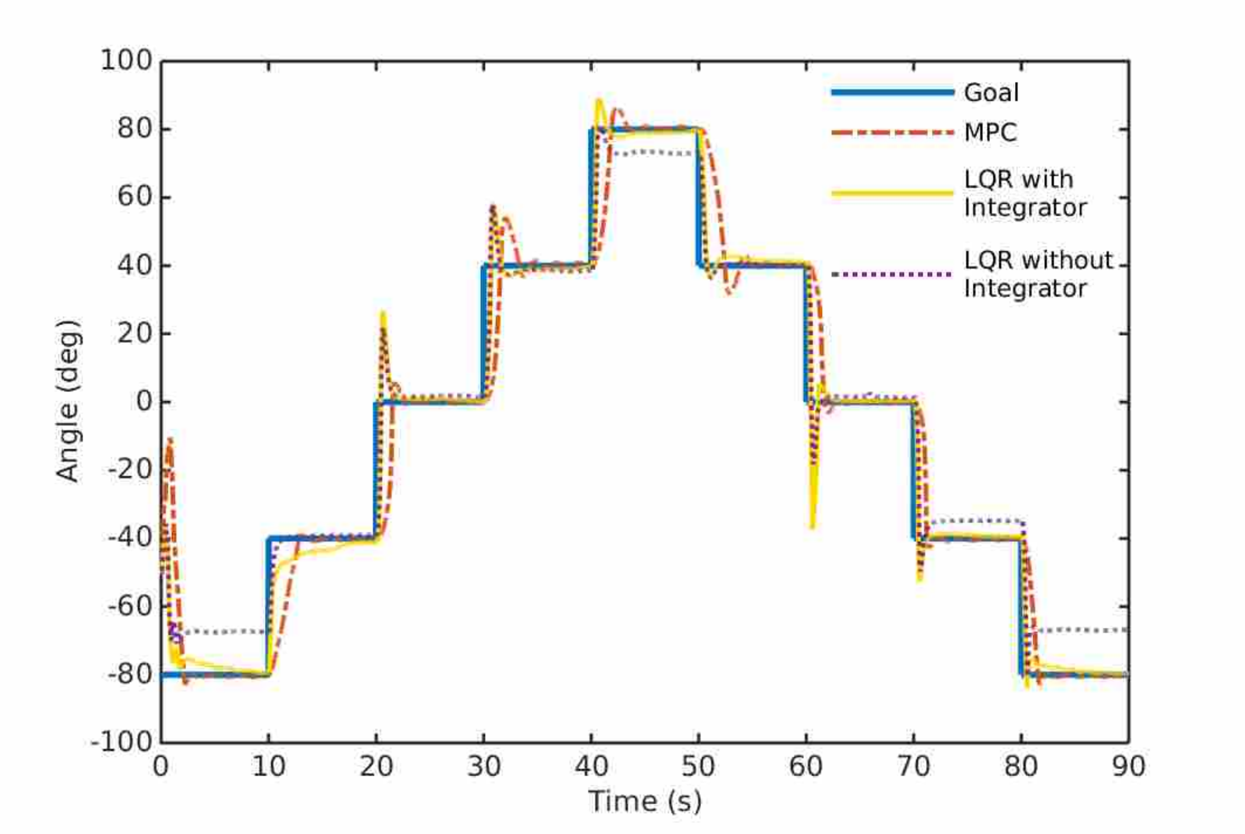

Figure 2.10: Single joint control response using MPC and LQR

2.2.5 King Louie

As shown previously MPC and LQR are both acceptable preliminary position control methods for a single degree of freedom joint. Although, the performance between the two was fairly comparable, there are pressure saturation limits and pressure dynamics that were not included in the two state control formulation. This along with other constraints (such as joint limits) that are currently included in the MPC model lead us to expect that MPC will be a better long term solution than LQR. However both MPC and LQR methods developed for a single joint were used to develop and implement MPC and LQR controllers for King Louie (a 14-DOF inflatable robot).

Model

The same rigid body model that was used for the grub was used for each joint on King Louie. Each joint was treated independently from the rest, even though inertia and gravity have a much greater effect than they did on a single joint. However, ignoring these effects in the model and treating them as disturbances does not prevent successful control of the arm. This is in part due to the low total weight of the arm which is estimated to be between 10 and 15 pounds. When compared to the actuator pressures which for this work have a maximum value of 17 PSI gage, the reaction forces between the opposing bladders are expected to be orders of magnitude greater than the forces due to mass and gravity so the system inputs will have a dominating effect on the system response. Even for the shoulder joints 0 and 1 where inertia and gravity effects are the greatest, control performance was affected by these assumptions but was still good.

Sensing

King Louie's arm kinematics can be approximated with Denavit Hartenberg parameters using an assumption of rigid links and compliant joints. These parameters are shown in Table 2.1 with values for DH_a and DH_d in meters, and DH_θ and DH_α in radians.

Table 2.1: DH parameters for King Louie

	Base-0	0-1	1-2	2-3	3-4	4-End
DH_θ	$\frac{\pi}{2}$	$\theta_1 + \frac{\pi}{2}$	θ_2	θ_3	θ_4	$\theta_5 + \pi$
DH_d	0	-0.05	0	0	0	0
DH_a	0	0.18	0.32	0.28	0.14	0.28
DH_α	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	0

For King Louie, motion capture sensors were used for joint angle estimates. The motion capture system uses quaternions for link orientations relative to the base frame. Calculating how the link quaternions were represented relative to each other allowed for joint angle estimation. With **q** signifying a quaternion, the relative quaternion between quaternions represented in a base frame is

$$\Delta\mathbf{q}_{(0-1)} = \mathbf{q}_1\mathbf{q}_0^{-1}, \tag{2.34}$$

as described in [54]. In other words, Equation 2.34 calculates the quaternion representation of $\mathbf{q}_1$ relative to $\mathbf{q}_0$. Using relative quaternions between joint quaternions and assuming rigid joints and links, the quaternion describing the rotation between two links was decomposed into the relevant Euler angles which give reasonable joint angle estimates. The motion capture reflective markers used to measure the joint angles can be seen in Figures 2.11, and 2.12.

Control

Similar to control for the grub, LQR with an integrator and MPC for controlling five degrees of freedom simultaneously were compared. Only five degrees of freedom were controlled because the gripper was treated as being either opened or closed and the hip joints were filled to maximum pressure to increase stability of the torso.

For both MPC and LQR, system parameters and weightings were tuned manually at each joint for acceptable performance. Each joint is controlled by a separate process and solves for MPC at 200 Hz for each joint. Because the MPC cost function can be minimized at 200 Hz, both the MPC and LQR controllers were operated at 200 Hz. A higher level controller passes desired joint angles to each joint separately. For this work, only step commands were passed to the controllers. One benefit to this type of controller developed for a single joint is that MPC can run at faster rates with a longer time horizon. A single controller that accounts for all the joints and more of the system dynamics would have slower solve rates and necessarily shorter horizons because of computational limitations, severely limiting the low level controller bandwidth. However, accounting for the low level control response in a higher level controller for future work, where high bandwidth is not as important, may be beneficial for overall control.

Results and Discussion for King Louie

MPC and LQR were successfully implemented for joint angle control for a multiple joint inflatable robot system. The system response to a step input sent to each joint can be seen in Figure 2.13. In Figures 2.11 and 2.12, King Louie's arm can be seen at the commanded angles. This orientation was chosen to show that King Louie can grab and manipulate objects in front of itself.

MPC and LQR with an integrator tracked the commanded joint angles with similar trends as was seen in Figure 2.10. MPC is slower to achieve the commanded angle but tracks well for all

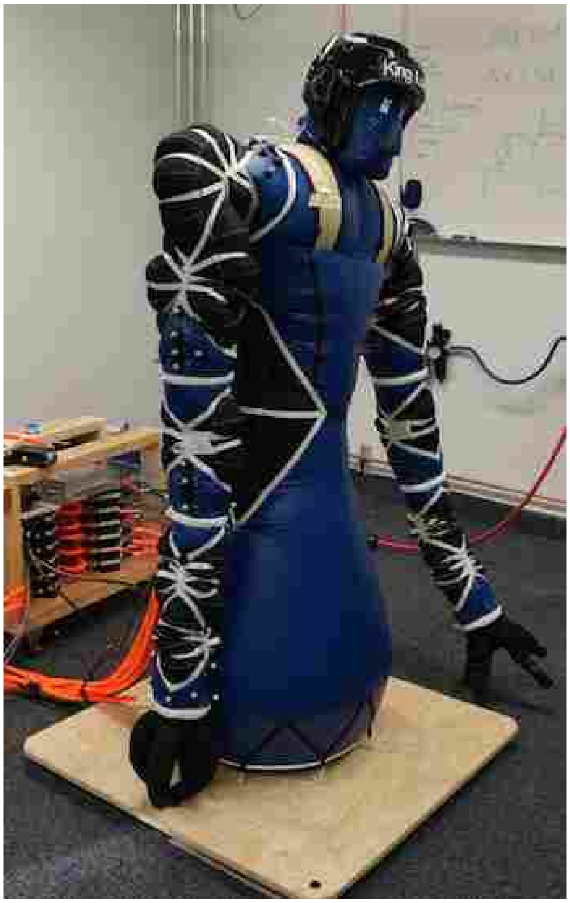

Figure 2.11: King Louie arm down

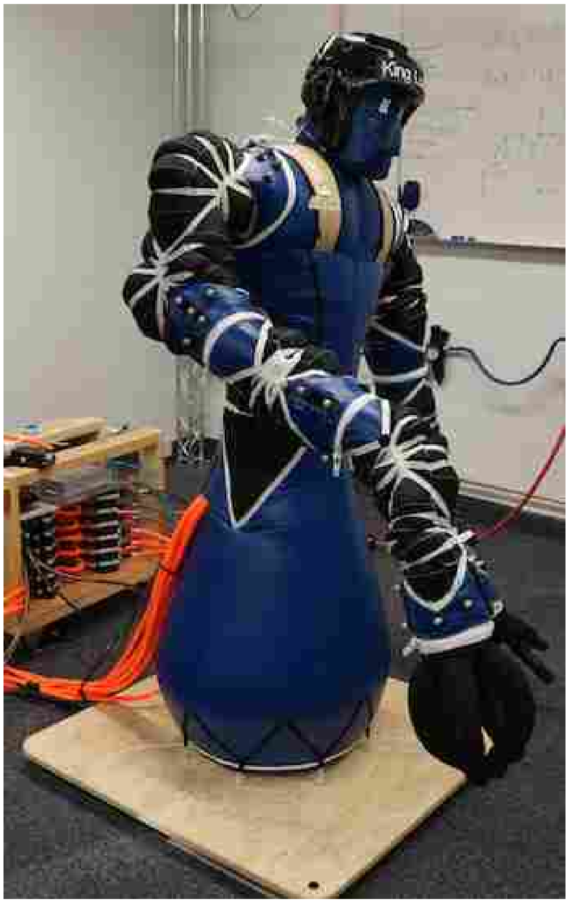

Figure 2.12: King Louie arm up

the joints with some oscillation. LQR has the faster response but does not track as well especially for Joint 2. This joint was fabricated differently than the other joints so that motion is restricted in one direction.

The slope of the sigmoid function in Equation 2.6, could be manipulated through s_1 which was important for stability. For the grub, the coefficients where chosen to match the profile in Figure 2.9, but for King Louie the coefficients were tuned for each individual joint. A steeper slope was necessary for the joints at the wrist which is likely because these joints are much smaller than the shoulder joints, and are more sensitive to changes in pressure. A steeper slope forced the inputs towards the center of the pressure range for these joints. A more shallow slope was necessary for the shoulder joints where the full pressure range was needed to compensate for disturbances caused by the mass of the arm.

Using MPC, joint space commands were given in a sequence to perform a task. The task involved picking up a board from a chair and placing it in a box behind King Louie (See video at https://youtu.be/o044-KW921I). The board weighs 1.38 pounds (.63 kg), is 20.5 inches (52 cm) long , 3.5 inches (9 cm) wide, and 1.5 inches (4 cm) thick. The task was completed eight out of ten times successfully. For one of the failed attempts, the board bounced off the top of the box and fell out of the box. For the other failed attempt, some of the IR reflective markers were not identified by the motion capture system which caused the joint estimation to fail after already successfully retrieving the object but before placing it in the box.

2.3 Four State MPC

After showing that joint position could be controlled using a simple angle to pressure conversion in [2], collaborative work was done to include pressures as states. This was accomplished using MPC and a block diagram for the system can be seen in Figure 2.14. For a full explanation of the model formulation please see [1]. The author of this thesis was not the primary author of [1] but was significantly involved. However the results from [1] are necessary background for Chapters 3 and 4.

As seen in Figure 2.14, θ_{goal} and a pressure target P_T for both actuators ($P_T = [P_{T,0}, P_{T,1}]$) are used by MPC to determine desired pressures for a low level pressure controller. The values for P_T are determined by a desired stiffness pressure P_S and the current joint angle. MPC determines the desired pressures by minimizing the equation

$$\begin{aligned} f(\dot{\theta}, \theta, P_0, P_1, \theta_{goal}) &= minimize \sum_{t=1}^{T} \Big(||\theta_{goal} - \theta[t]||_Q^2 + ||\dot{\theta}[t]||_R^2 \\ &\quad + ||P_0[t] - P_{T,0}||_W^2 + ||P_1[t] - P_{T,1}||_W^2 \Big). \end{aligned} \tag{2.35}$$

A small cost is given for pressures that vary from P_T. This allows MPC to pick a desired pressure combination that will achieve the desired angle but it will constrain the solution to a small section of the angle mapping seen in Figure 2.8. The desired pressures are maintained by a low level PID pressure controller operating at 1000 Hz which is faster than the MPC outer loop with a maximum

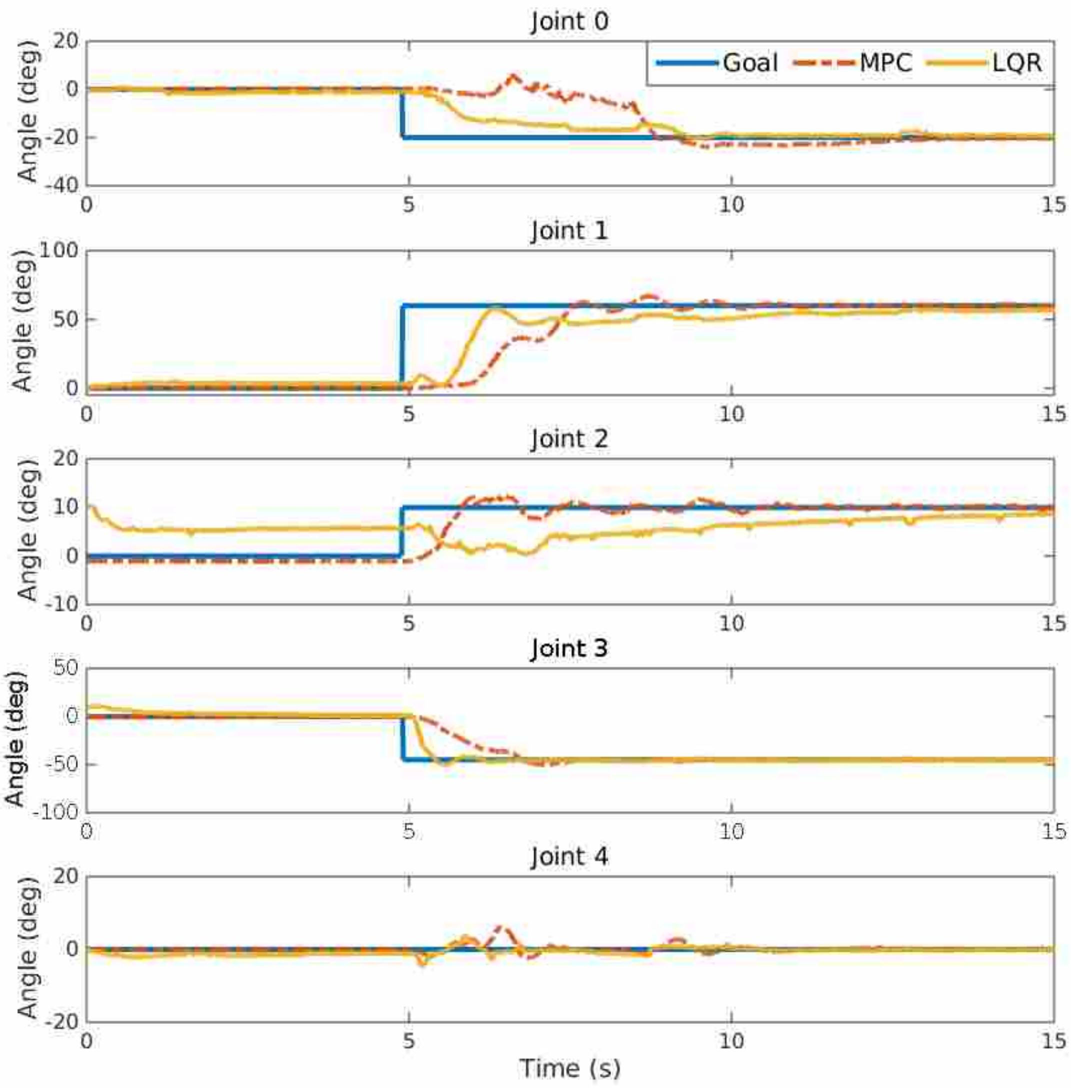

Figure 2.13: King Louie step response

rate of 200 Hz. The pressure controller and joint estimation is the same as described in Section 2.2.

The four state controller performance where P_0 and P_1 are introduced as states, compared to the two state controller performance with only $\dot{\theta}$ and θ as states can be seen in Figure 2.15. There was significant reduction in rise time, settling time, and overshoot using the four state controller.

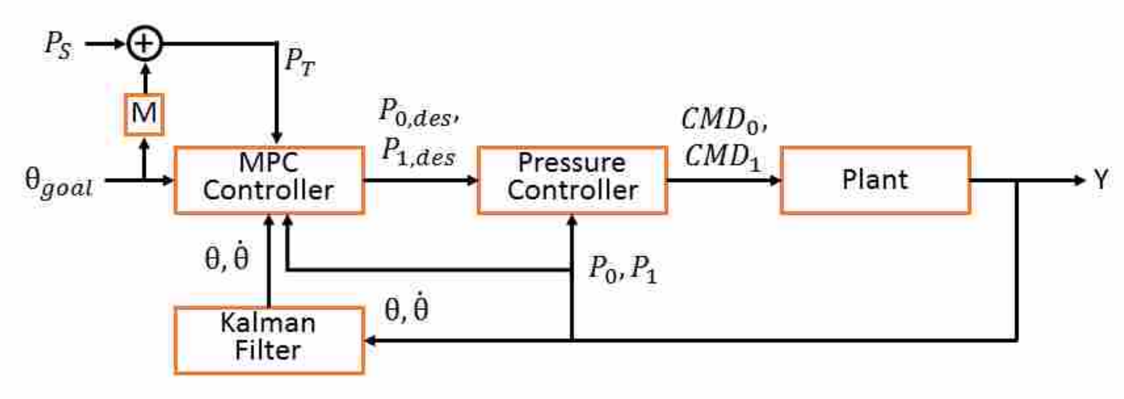

Figure 2.14: MPC control diagram for joint and stiffness control

A summary of the improved performance can be seen in Table 2.2. Using the pressures within the model significantly improved the controller performance.

Table 2.2: Performance comparison between two state MPC and new four state MPC

	Avg. Rise Time	Avg. Settling Time	Avg. % Overshoot
2 State MPC	1.319 sec	2.985 sec	24.408%
4 State MPC	0.556 sec	1.357 sec	2.561%
Improvement	137.24%	119.89%	853.21%

Another significant result from this work is that position can be maintained while the pressures are changed within the actuators using P_T. Figure 2.16 shows that as the pressure target changes for each actuator, the joint angle does not deviate from the desired angle. The actuator pressures do not track the target pressures accurately because angle tracking is the primary objective, but the trends are the same.

2.4 Torque Model

The torque model developed in [3] was used to more effectively control joint position. The equation used for this model was

$$I\ddot{\theta} - k_d\dot{\theta} = k_s\theta + a_0P_0 + a_1P_1 = \tau. \tag{2.36}$$

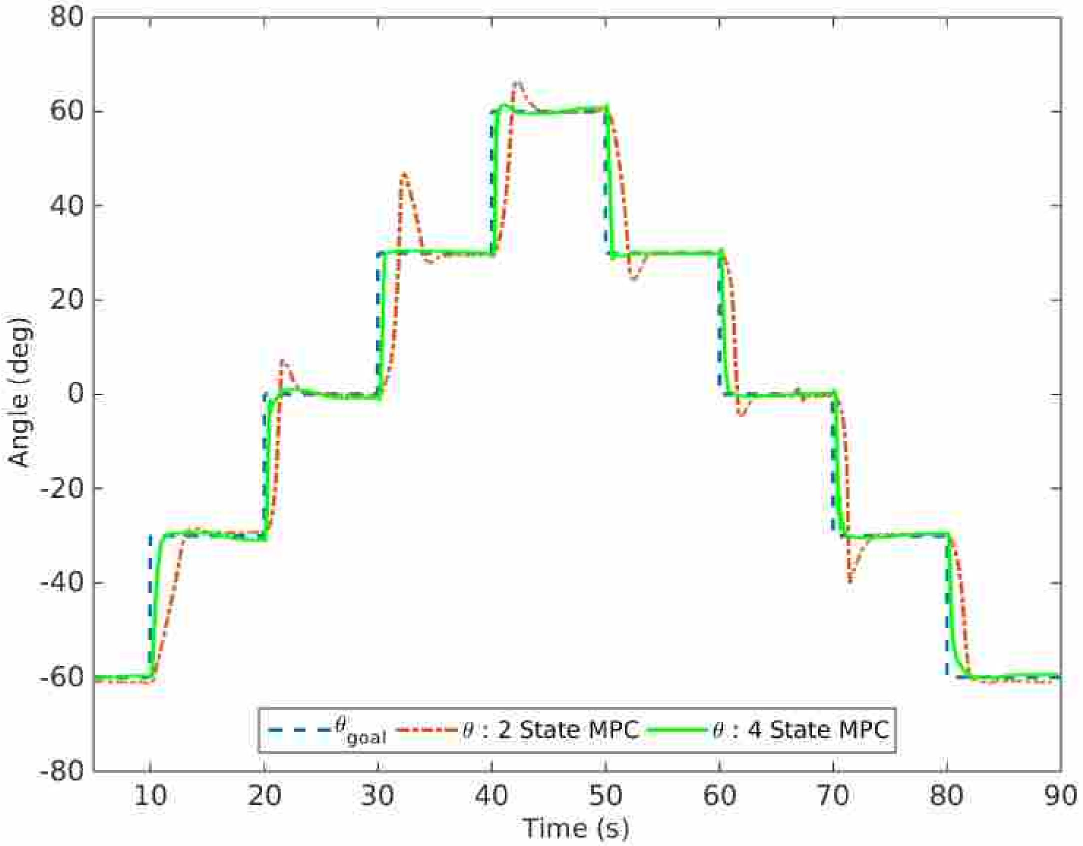

Figure 2.15: Comparison between 2 state MPC control and 4 state MPC from a series of step inputs

Reducing this equations gives

$$\ddot{\theta} = \frac{k_d}{I}\dot{\theta} + \frac{k_s}{I}\theta + \frac{a_0}{I}P_0 + \frac{a_1}{I}P_1. \tag{2.37}$$

The torque coefficients were measured by constraining the grub as seen in Figure 2.17 and measuring the output torque at different angles for an input pressure. The data collected from the test can be seen in Figure 2.17 and was reported in [3]. The linear trend of the torque data in Figure 2.18 supports the linear model derived in Equation 2.36. The torque coefficients a_0 and a_1 were calculated from the slope of a linear fit to the data.

Using the developed torque model, MPC was applied to the system assuming a first order pressure model for the closed-loop PID pressure controller dynamics. Figure 2.19 shows the block diagram for the system which is similar to the control scheme used in [1].

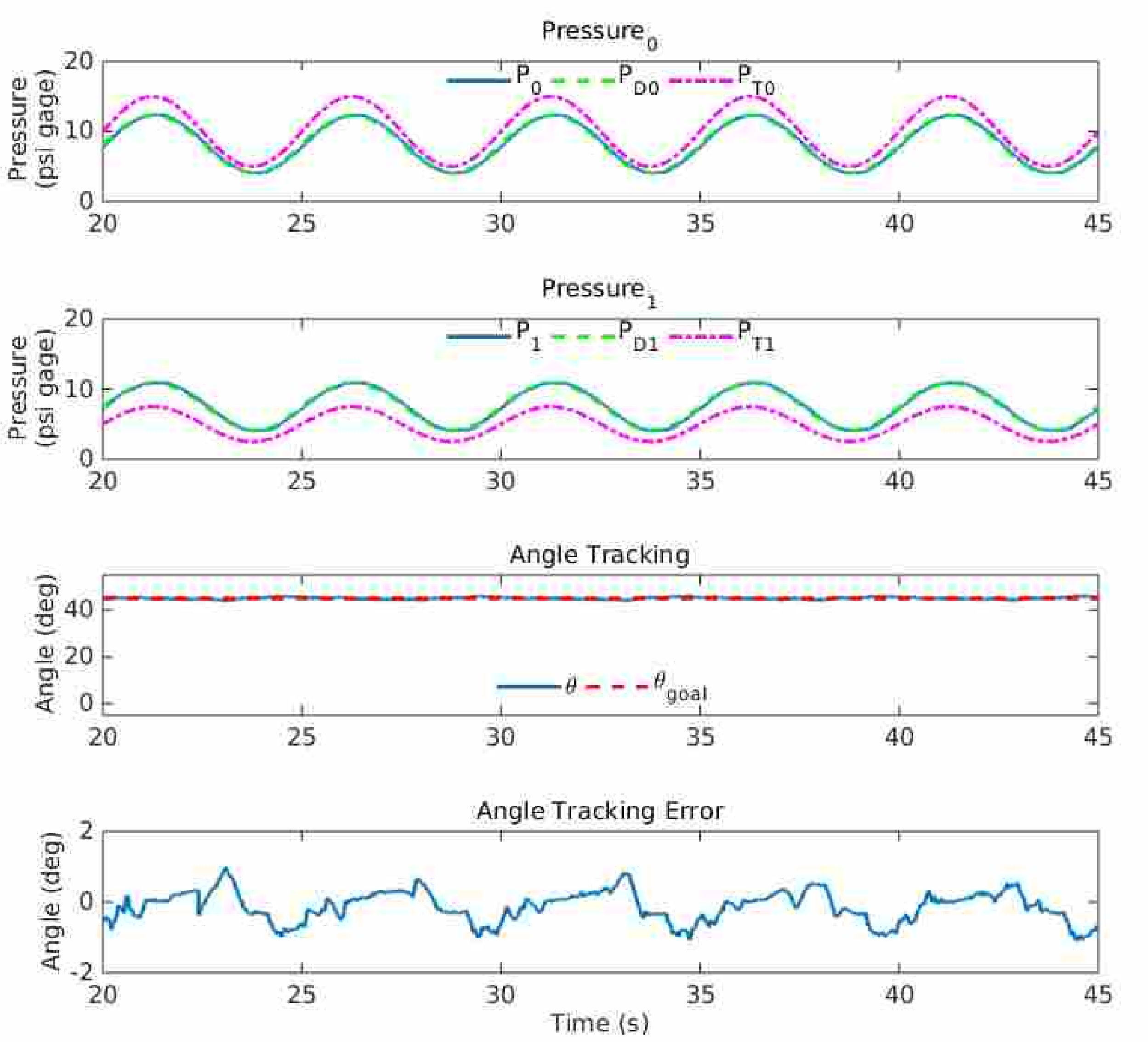

Figure 2.16: Results for Grub holding a constant angle with a sinusoidal pressure command which is related to stiffness

MPC determines desired pressures for good tracking of a desired angle signal. The desired pressures are maintained by a low level PID pressure controller operating at 1000 Hz. The pressure controller sends current commands to valves which controls air flow to each of the actuators. A Kalman filter for the grub then estimates the joint angle and angular velocity from IMU sensor data.

Controller performance can be seen in Figure 2.20 along with the performance of the two state controller. From the figure, it is clear that the torque model has significantly less overshoot

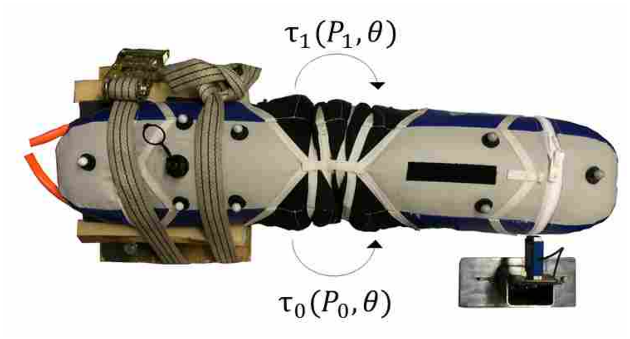

Figure 2.17: Testing setup with force sensor mounted at 0°

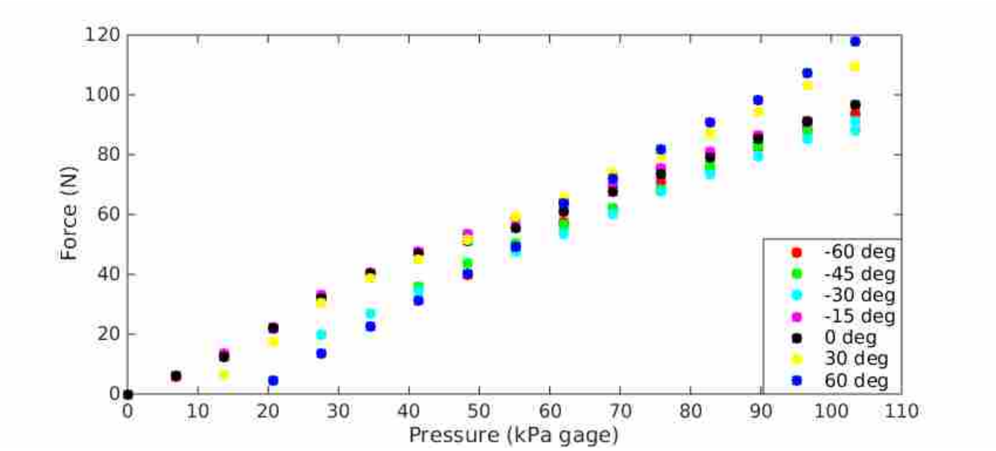

Figure 2.18: Actuator force at different pressures and angles

than the two state controller and faster initial rise times. A summary of controller improvement from the two state controller from Section 2.2 can be seen in Table 2.3.

Table 2.3: Performance comparison two state MPC and the four state MPC with the torque model

	Avg. Rise Time	Avg. Settling Time	Avg. % Overshoot
2 State MPC	1.319 sec	2.985 sec	24.408%
4 State Torque-based MPC	1.360 sec	2.592 sec	2.587%
Improvement	-3.02%	15.13%	843.35%

Using the torque model, MPC was then applied to King Louie with the same method described in Section 2.2.5 and the results can be seen in Figure 2.21. Although position control is

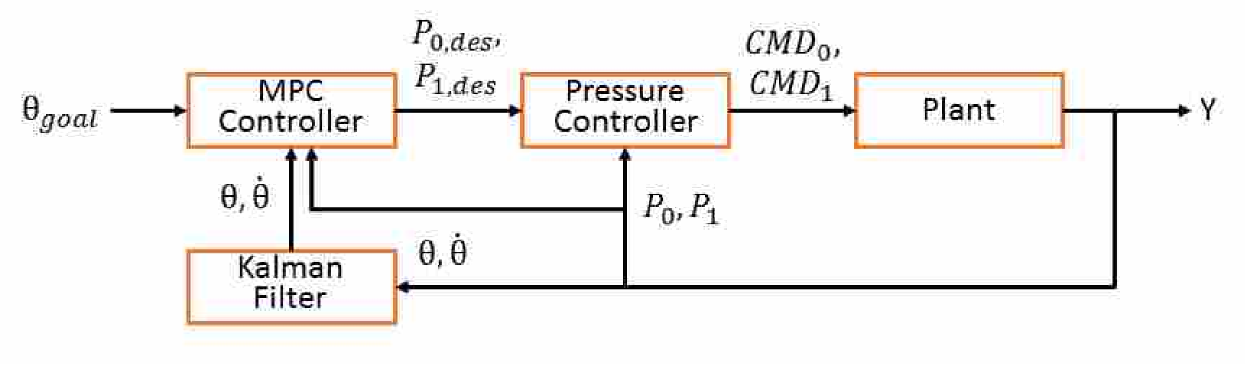

Figure 2.19: MPC control diagram for joint control

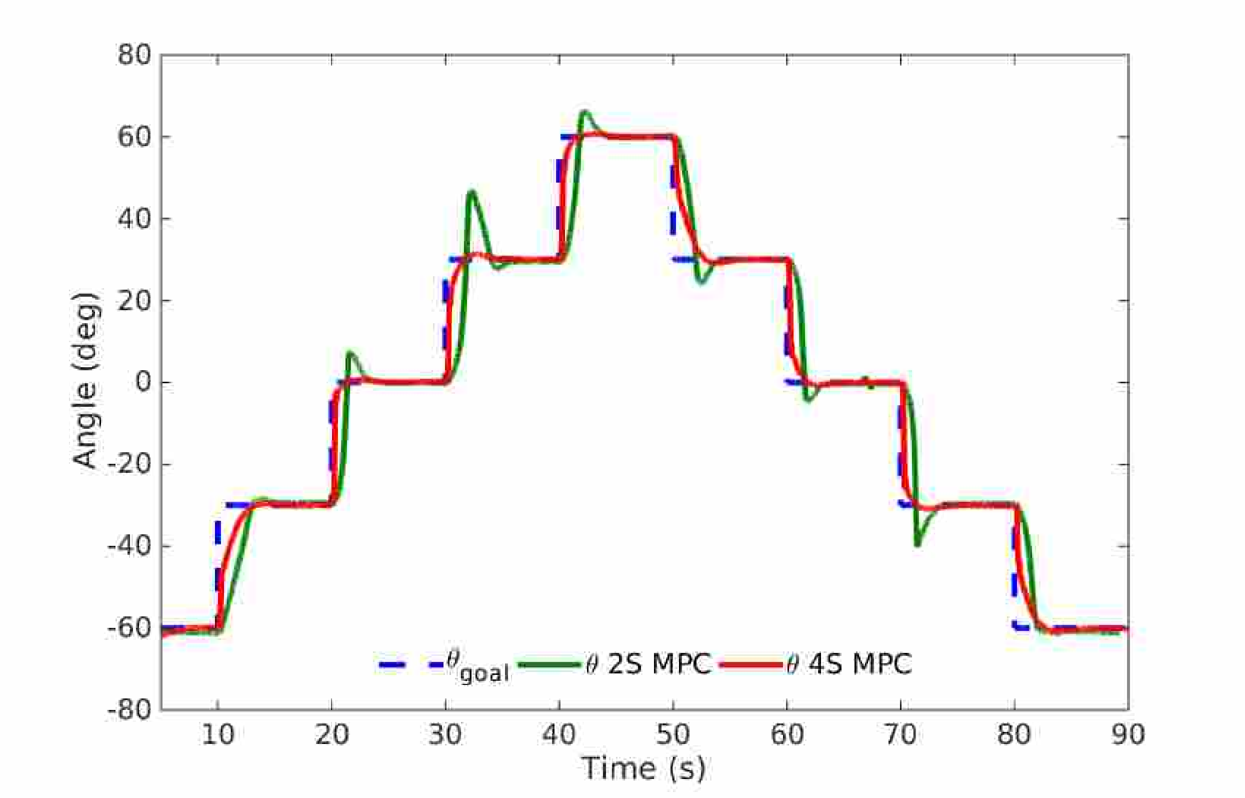

Figure 2.20: Controller comparison results for a single joint

not perfect, the four state controller does much better at tracking the desired angle and reducing oscillations than the two state controller, especially for joints 0 and 1 at the shoulder.

Joint space control was used on King Louie in [3] to show that the end effecter can repeatably reach to the same position within 3 cm. A video of these reaching trials is available at `https://youtu.be/4T-FN581RkA`. Also, like in Section 2.2.5, a pick and place task was com-

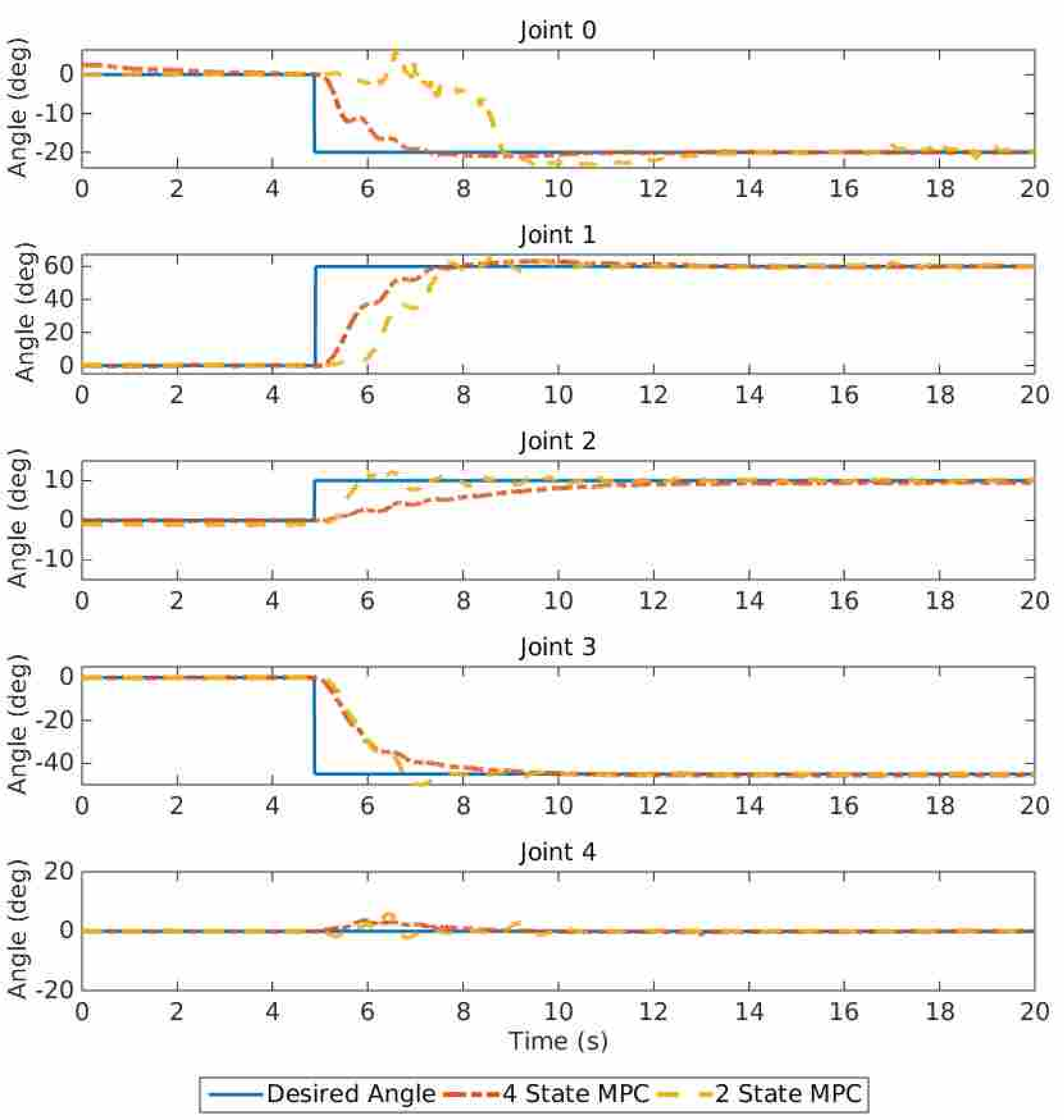

Figure 2.21: This is a step response comparison between the four state MPC response from Section 2.4 and the two state MPC response from Section 2.2.5 for King Louie's right arm.

pleted to pick up a ball and drop it in a bucket. The pick was successful ten out of ten times and the place task was successful nine out of ten times. A video of these trials is available at `https://youtu.be/p30jKn7_pV4`.

CHAPTER 3. STIFFNESS MODEL DEVELOPMENT

One goal of this research is to be able to control joint stiffness. In order to explicitly control stiffness, a model must be developed to quantify and relate stiffness to other states. In this chapter, first an angle to volume relationship is developed that describes how actuator volumes change with the joint angle of the grub. Next, equations describing the stiffness of the joint and the time derivative of the joint stiffness are formulated. These equations are used to model both stiffness and position for the joint. Lastly, the model is fit to data collected from the grub. Test results are shown that validate the stiffness model by comparing the actual joint stiffness to the estimated joint stiffness at specific pressures in the actuation chambers.

3.1 Assumptions and Basic Principles

In order to develop a model for joint stiffness, several assumptions must be made about the dynamics of the system. The first assumption made is that since air is the working fluid in the joint the ideal gas law can be used to model fluid properties. Also since the actuation pressures and mass flows are relatively small, constant temperature can also be assumed. Constant temperature for this type of actuator was shown to be a valid assumption in [14]. Results in [14] showed that models with constant temperature had almost identical results as higher fidelity models that accounted for changes in temperature. Using these assumptions, the actuator pressure can be described by the equation

$$P_i = \frac{m_i R D}{V_i}, \qquad i = 0, 1 \tag{3.1}$$

where m_i is the mass of the air in the i^{th} actuator, R is the ideal gas constant for air ($287\,Jkg^{-1}K^{-1}$), D is the temperature of the air (assumed room temperature $294\,K$), and V_i is the volume of the actuator.

An important relationship for the model development was how the joint angle relates to actuator volume. The antagonistic nature of the joint makes it so as the angle increases, one actuator expands in volume and the other compresses. Because the joint is made from fabric and is manually fabricated, it is impractical to measure the actuator bladder volume geometrically. However, by using another chamber with a known volume ($.0028\,m^3$ or $170\,in^3$), the volume of the actuator bladder can be calculated. This was done by filling the rigid black chamber shown in Figures 3.1 and 3.2 to a known pressure and releasing the air into the empty actuator bladder. The ideal gas law can be used to calculate the total mass of air from the filled rigid tank and then again using the mass of air from the first calculation to calculate the final total volume. The actuator volume is then simply the rigid tank volume subtracted from the total volume. The test setup can be seen in Figures 3.1 and 3.2 and the procedure diagram can be seen in Figure 3.3.

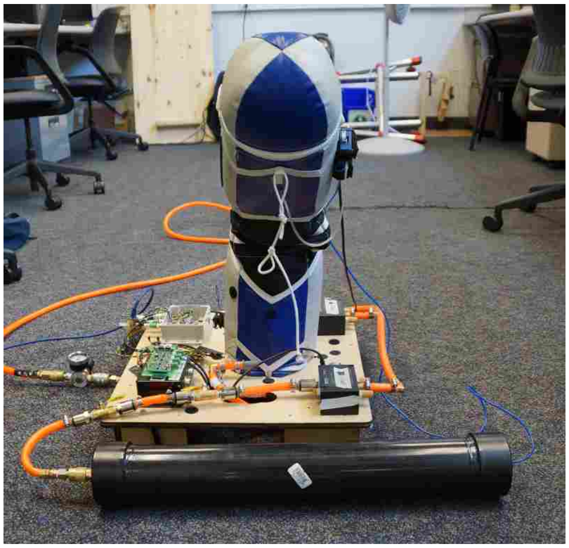

Figure 3.1: The rigid tank is connected to the valve by the orange hose (left) and is filled to a known pressure. The rigid tank pressure is measured by a pressure sensor connected by the blue tube (right). The back actuator (unseen) is filled to a known pressure and maintained.

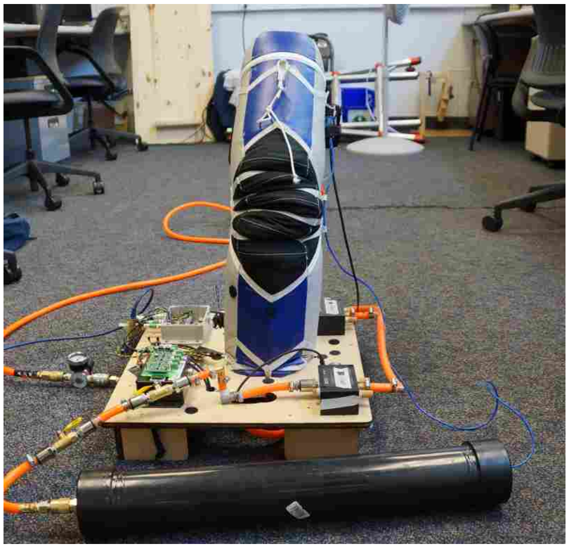

Figure 3.2: The rigid tank is allowed to empty into the front actuator (seen) which pushes the link to the opposite side.

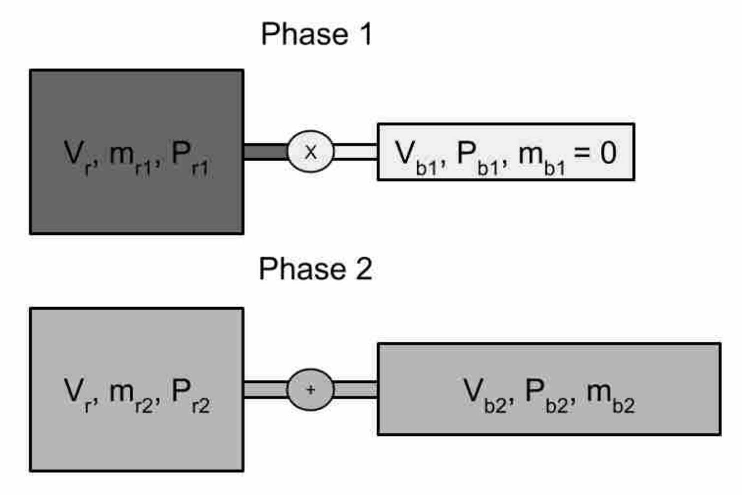
Phase 1
V_r, m_{r1}, P_{r1}
x
V_{b1}, P_{b1}, $m_{b1} = 0$
Phase 2
V_r, m_{r2}, P_{r2}
+
V_{b2}, P_{b2}, m_{b2}

Figure 3.3: This represents how the volume in the bladder changes when the pressurized air in the rigid chamber is released. For the rigid tank, V_r is the volume, m_r is the mass of the air, and P_r is the pressure. For the bladder, V_b is the volume, m_b is the mass of air, and P_b is the pressure.

For Phase 1 in Figure 3.3, the mass of the air can be calculated by

$$m_{r1} = \frac{P_{r1}V_{r1}}{(RD)}. \tag{3.2}$$

Since the actuator is made from fabric, there will still be pockets of air and the hose cannot be completely emptied. The initial volume and mass in the actuator for Phase 1 will not be exactly zero but is assumed to be zero because the actuator is compressed by the opposite actuator. For Phase 2, it is known that $P_{r2} = P_{b2} = P_2$ and $m_{r2} + m_{b2} = m_{r1}$ because it was assumed that no air was lost so the total volume of the system must be

$$V_{sys} = \frac{m_{r1}RD}{P_2}. \tag{3.3}$$

Since $V_{sys} = V_r + V_{b2}$, the volume of the actuator bladder is

$$V_{b2} = V_{sys} - V_r. \tag{3.4}$$

Each actuator bladder volume corresponds to a certain angle. Different initial pressures in the rigid tank resulted in different settling angles for the joint. To get the full range of angles, the opposite actuator was maintained at different pressures. Figure 3.4 shows a linear relationship (albeit noisy) between joint angle and volume of the actuator

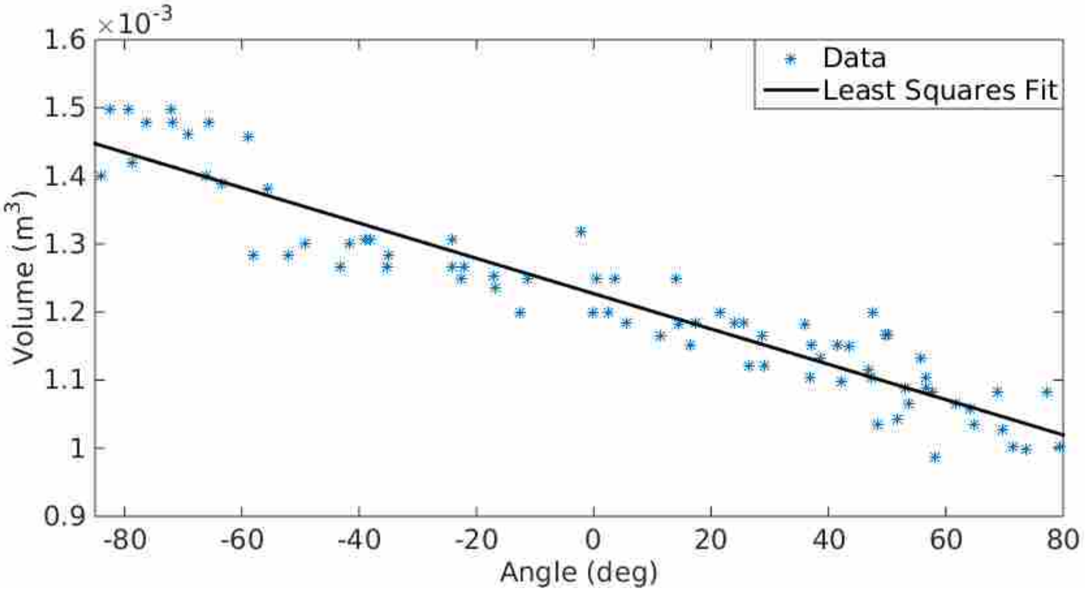

Figure 3.4: Linear fit from joint angle to actuator volume

and the linear expression for this relationship is

$$V_i = M_i\theta + b. \qquad i = 0,1 \tag{3.5}$$

Substituting this into Equation 3.1 gives

$$P_i = \frac{m_i RD}{M_i\theta + b}, \qquad i = 0,1 \tag{3.6}$$

for pressure in the i^{th} actuator bladder. The values for M_i and b can be found in Table 3.1 in Section 3.3.

The variability in the data shown in Figure 3.4 is caused by many factors. The largest variability is expected to come from the physical characteristics of the joint. Each time the joint rotates, different folds within the actuators fill and shift making the actuator volume different for each test. Because there is not a rigid separation between each of the actuators, the folds can significantly affect actuator interaction by how they push on each other. Complicating the problem even further, the actuators push on the central column which is an inflated chamber that can also fold and shift as the joint rotates. Future designs that allow for the actuator to fill and empty more uniformly without the central column are expected to reduce the volume measurement variability. Other factors that could affect the variability include noise on the pressure and angle signals, possible leakage around the pressure sensors, valves, or fittings, assuming ideal gas conditions with constant temperature.

3.2 Stiffness Equation Derivation

Using the torque model that was developed in [3] a model for joint stiffness can be developed. The output stiffness for a rotary joint with compliant actuation can be defined as

$$k = -\frac{\partial \tau}{\partial \theta}, \tag{3.7}$$

where

$$\tau = a_0 P_0 + a_1 P_1. \tag{3.8}$$

Using a negative in the equation keeps the stiffness term k positive and is the convention for this thesis. The reaction torque from the joint is dependent on how the applied joint torque (τ_a) changes with angle. The equation for stiffness becomes

$$k = -\frac{\partial}{\partial \theta}\left(a_0 P_0 + a_1 P_1\right), \tag{3.9}$$

which when using the pressure expression from Equation 3.6 and taking the partial derivative with respect to θ becomes

$$k = \frac{a_0 M_0 m_0 R D}{(M_0 \theta + b)^2} + \frac{a_1 M_1 m_1 R D}{(M_1 \theta + b)^2}. \tag{3.10}$$

This stiffness equation can be reduced by substituting back in pressures P_0, and P_1 using Equation 3.6 to get

$$k = \frac{a_0 M_0 P_0}{M_0 \theta + b} + \frac{a_1 M_1 P_1}{M_1 \theta + b}. \tag{3.11}$$

Letting

$$\alpha_i = \frac{M_i}{M_i\theta + b}, \qquad i = 0,1 \tag{3.12}$$

where *i* corresponds to the actuator index, and *k* becomes

$$k = a_0\alpha_0 P_0 + a_1\alpha_1 P_1. \tag{3.13}$$

By taking the derivative of Equation 3.13 with respect to time gives

$$\dot{k} = a_0\alpha_0\dot{P}_0 + a_1\alpha_1\dot{P}_1 - \dot{\theta}\left(a_0\alpha_0^2 P_0 + a_1\alpha_1^2 P_1\right). \tag{3.14}$$

However, by assuming a first order model on the PID pressure controller dynamics or

$$\dot{P}_i = c_0 P_i + c_1 P_{i,des}, \qquad i = 0,1 \tag{3.15}$$

which represents the PID pressure controller dynamics and using Equation 3.13 gives

$$\dot{k} = c_0 k + a_0\alpha_0 c_1 P_{0,des} + a_1\alpha_1 c_1 P_{1,des} - \dot{\theta}\left(a_0\alpha_0^2 P_0 + a_1\alpha_1^2 P_1\right). \tag{3.16}$$

A model was developed using equations describing the gas dynamics but the performance of model based control was poor. More on the development of the equations describing the gas dynamics can be seen in Appendix A.

Using the torque model Equation 2.37, the full set of state space equations for a model of angular velocity, angle, stiffness, and pressures as states becomes

$$\begin{aligned}
\ddot{\theta} &= \frac{k_d}{I}\dot{\theta} + \frac{k_s}{I}\theta + \frac{a_0}{I}P_0 + \frac{a_1}{I}P_1 && (3.17)\\
\dot{\theta} &= \dot{\theta} && (3.18)\\
\dot{k} &= c_0 k + a_0\alpha_0 c_1 P_{0,des} + a_1\alpha_1 c_1 P_{1,des} - \dot{\theta}\left(a_0\alpha_0^2 P_0 + a_1\alpha_1^2 P_1\right) && (3.19)\\
\dot{P}_0 &= c_0 P_0 + c_1 P_{0,des} && (3.20)\\
\dot{P}_1 &= c_0 P_1 + c_1 P_{1,des}. && (3.21)
\end{aligned}$$

3.3 Model Validation

The set of Equations 3.17 through 3.21 is expected to be a simplified set of dynamics of the actual inflatable joint. As such, it must be shown to represent a majority of the joint dynamics. Model parameters were fit to data collected from the grub using least squares methods. Table 3.1 contains the parameters used within the state space model. The sign for the stiffness and damping coefficients is based on the convention set by the torque model in Equation 2.37 where these terms are positive.

Table 3.1: Stiffness model parameters and units

	Parameter	Units
k_d	-10.85	Nms/rad
k_s	-51.77	Nm/rad
a_0	.0011	Nm/Pa
a_1	$-.0011$	Nm/Pa
I	.0174	kg m^2
c_0	-5.301	s^{-1}
c_1	5.301	s^{-1}
M_0	$1.4857e-04$	m^3/rad
M_1	$-1.4857e-04$	m^3/rad
b	$1.222e-03$	m^3

3.3.1 Model Simulation Compared to Actual Data

Figure 3.5 shows how the fitted model compares to data collected from the grub. For the simulation, only the initial conditions are known to the model and the same desired pressures that were applied to the grub are applied to the model. Most of the discrepancy between the model and measured data from the grub most likely come from the first order pressure model. There are gas dynamics and actuator interaction dynamics that are not included in this model.

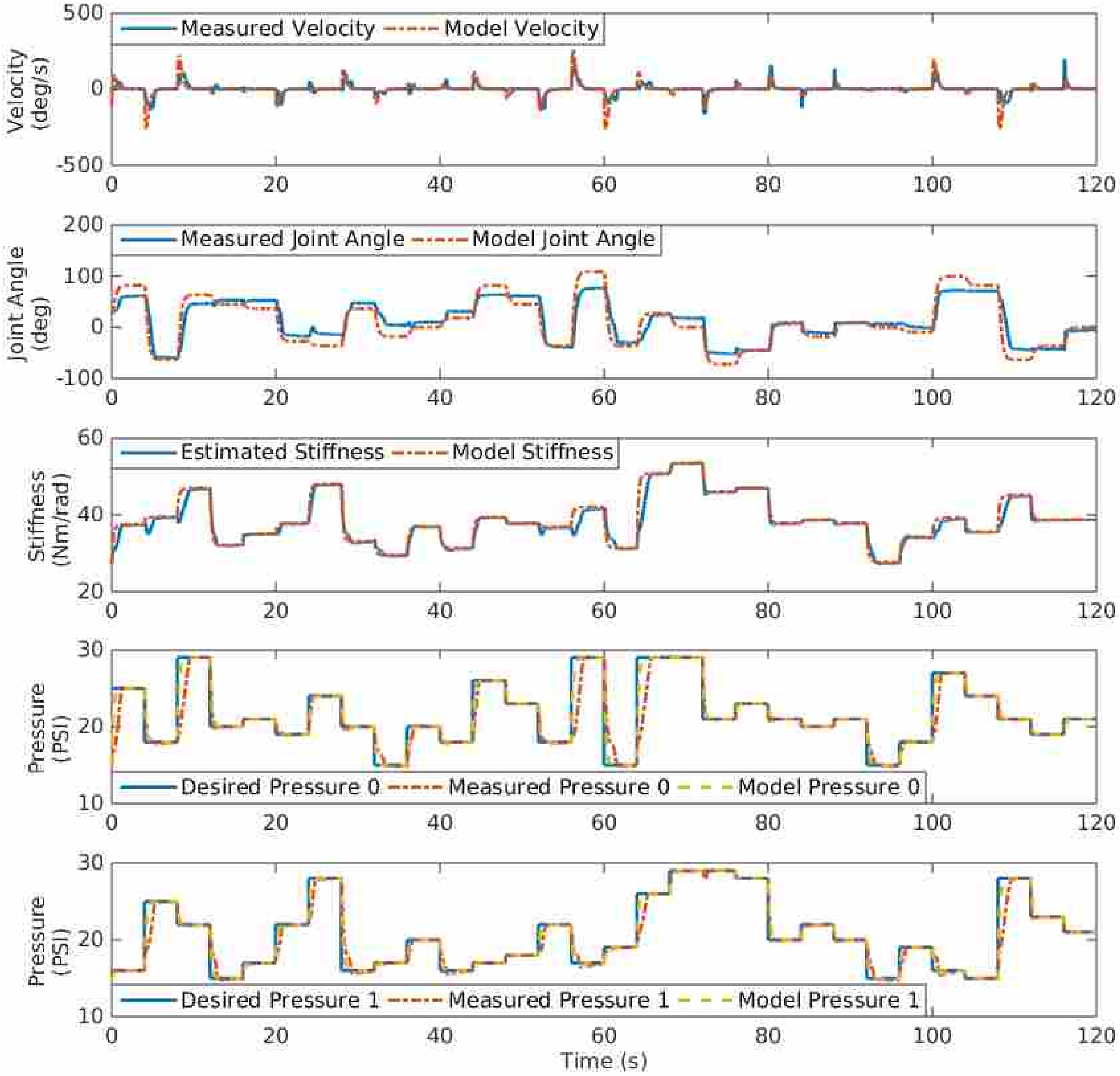

Figure 3.5: Model simulation compared to actual data collected from the grub

Work was done to include more of the gas dynamics such as mapping valve commands to mass flow rates and using mass flow rates as system inputs. This was an effective model for control in simulation where model discontinuities such as choked and unchoked flow could be easily handled but actual control implementation failed. The two main reasons why it failed using a model that included more of the gas dynamics were because of failed convergence for MPC and because the valves used were better suited for PID pressure control rather than providing reliable mass flow because there is not any feedback on the spool position. The valve command signal is noisy and the effective valve area which is necessary for reliable mass flow estimates is not consistent for the same valve current commands at different times. Future work to model gas dynamic affects can improve the accuracy of the system model but as seen in Chapter 4, the current model is sufficient for control of both joint position and stiffness.

3.3.2 Stiffness Validation

The joint stiffness for the inflatable joint cannot be measured directly so it is estimated using Equation 3.13. In order to validate that the estimated stiffness follows the same trends as the actual joint stiffness, the joint was controlled to a stiffness and a torque was applied. The actual stiffness could then be calculated by dividing the applied torque by the measured deflection. Two methods were used to measure the joint stiffness. Both methods were accomplished at two angles where the joint mounting was oriented and constrained so that weights could be hung from the link at a known distance from the joint center which was .3 m. The set-up can be seen in Figures 3.6 and 3.7.

For the first method, two different weights were applied five times and the stiffness was calculated for each test. This method was designed to show the repeatabilty of the stiffness measurement. All calculated values for stiffness were then averaged together. The results for the first method can be seen in Table 3.2. The second column shows the two applied torques used for the desired stiffness specified in the third column. Two different torques were applied to show that the average measured stiffness was not configuration dependent. At both the 0° and 45° configurations, the total stiffness or measured stiffness matches the estimated stiffness for all but low stiffnesses.

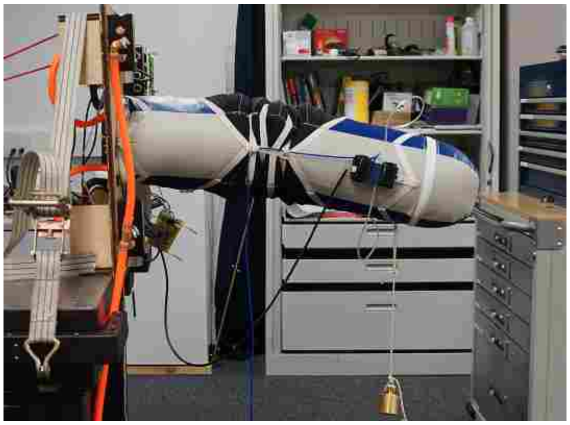

Figure 3.6: Experimental set-up for stiffness validation at aproximatly 0°

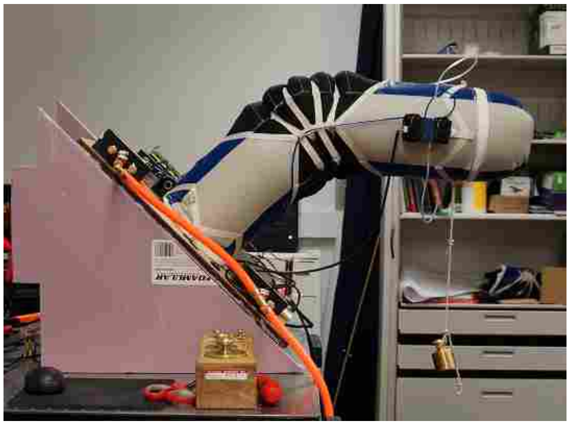

Figure 3.7: Experimental set-up for stiffness validation at aproximatly 45°

Table 3.2: Stiffness data at 0° and 45°

Angle (deg)	Torque* (Nm)	k Des (Nm/rad)	k (Nm/rad) Average for Applied Torque	Total k (Nm/rad) Average
0	4.67	50	52.41	49.76
	6.00		47.11	
0	4.00	45	46.37	46.16
	5.34		45.94	
0	3.34	40	42.17	39.33
	4.67		36.48	
0	2.67	35	34.12	30.22
	4.00		26.32	
45	4.67	50	43.11	44.14
	6.00		45.17	
45	4.00	45	41.04	41.28
	5.34		41.51	
45	3.34	40	35.94	36.51
	4.67		37.08	
45	2.67	35	32.49	31.62
	4.0		30.74	

* The applied torque corresponds to the force applied times the distance to the application point which was .3 meters.

For the second method, weights were applied from 0.5 lbs to 6 lbs in 0.5 lb increments and the deflection was recorded. This test was designed to show that the stiffness measurement is consistent for different applied torques. A least squares linear fit was applied to the data and the slope of the fit was the measured stiffness. The results of this method can be seen in Figures 3.8 and 3.9.

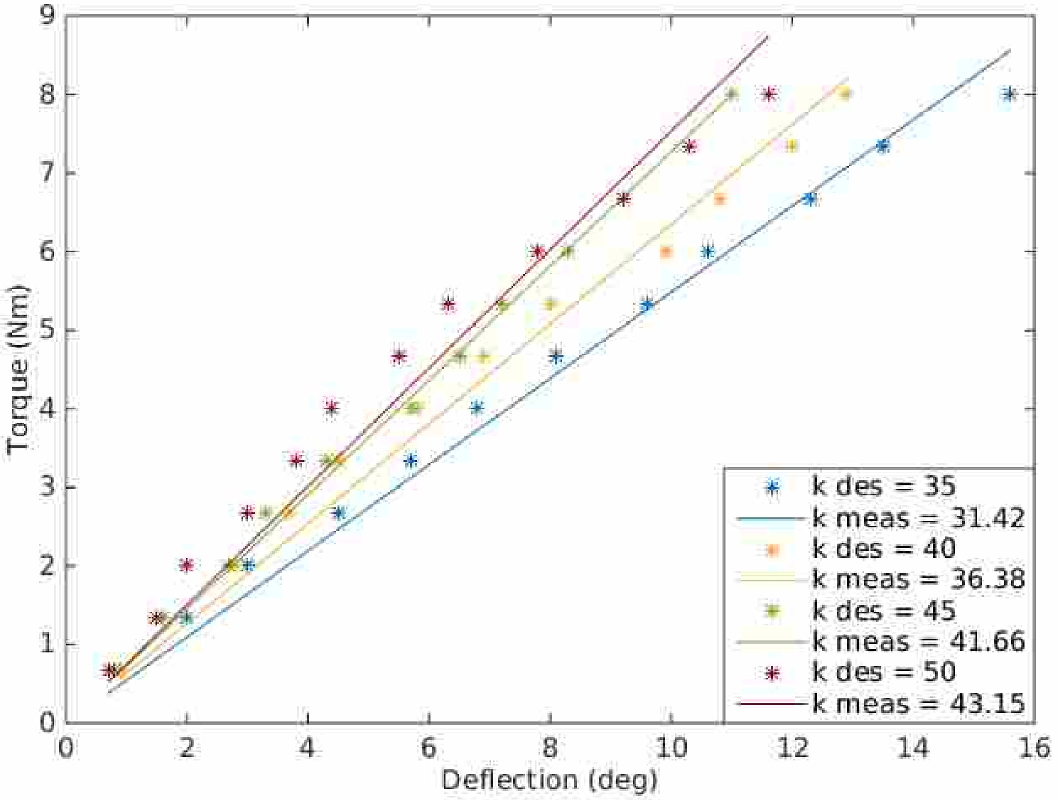

Figure 3.8: Stiffness validation at 0°

The stiffness measurement in Figure 3.9 matches closely the result found in Table 3.2. However there is a discrepancy between the stiffness measurements for the two methods at 0°. The reason for this is a hardware limitation. A crease in the actuator that formed between tests caused a pinch point which decreases the joint stiffness as deflections increase. For small deflections, the effects of the pinch point are minimal and the stiffness is a linear trend. For deflections that result in a reaction torque less than 3.336 Nm, the joint stiffness appears to be linear and can be seen in Figure 3.10. The stiffness measurement from this region matches the measurements obtained from the first method where the pinch point was not a problem. This problem will be mitigated with

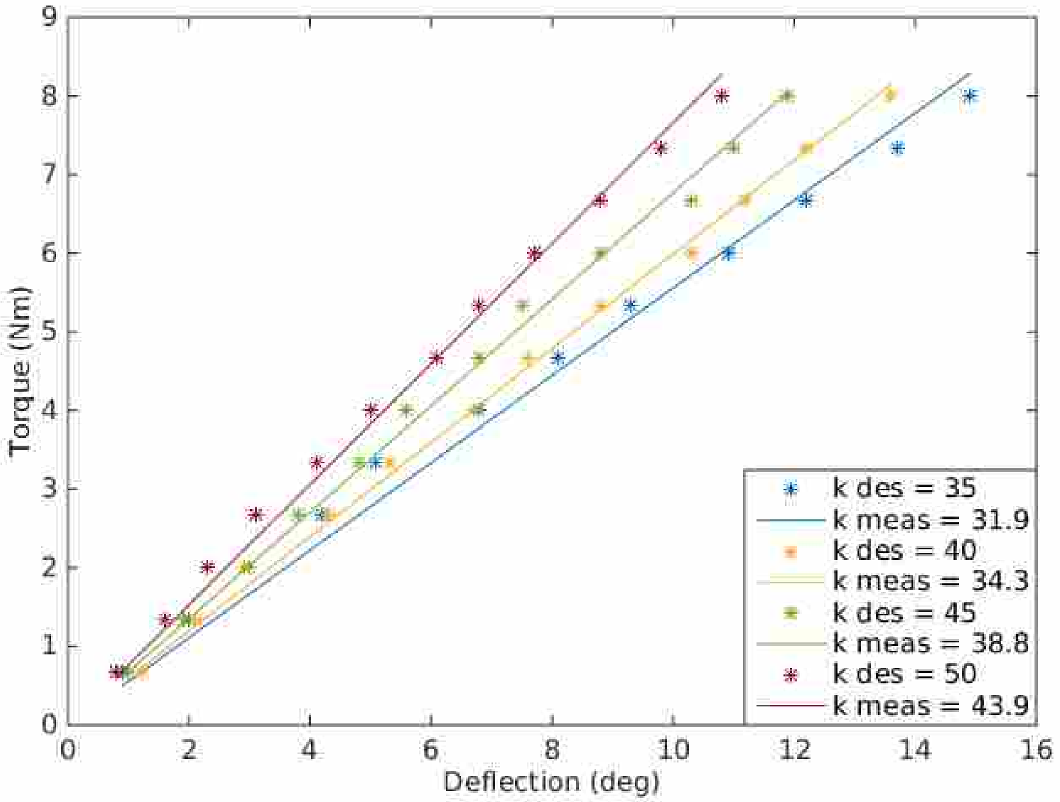

Figure 3.9: Stiffness validation at 45°

future designs where uniform filling within the actuator chamber will prevent pinch points from forming.

Because the measured stiffness matches the estimated stiffness, it is reasonable to claim that when the joint is controlled to a desired stiffness, that the stiffness estimation is a sufficient representation of the actual joint stiffness. This is important for control because stiffness cannot be measured in-line. Much work was done to tune parameters for the joint so that the estimated stiffness matched the measured stiffness. For some joints the stiffness cannot be measured as easily making it difficult to tune parameters. However, even though the joint estimation may not be accurate, if the estimated stiffness increases, the actual joint stiffness will increase and if it decreases, the actual joint stiffness will decrease.

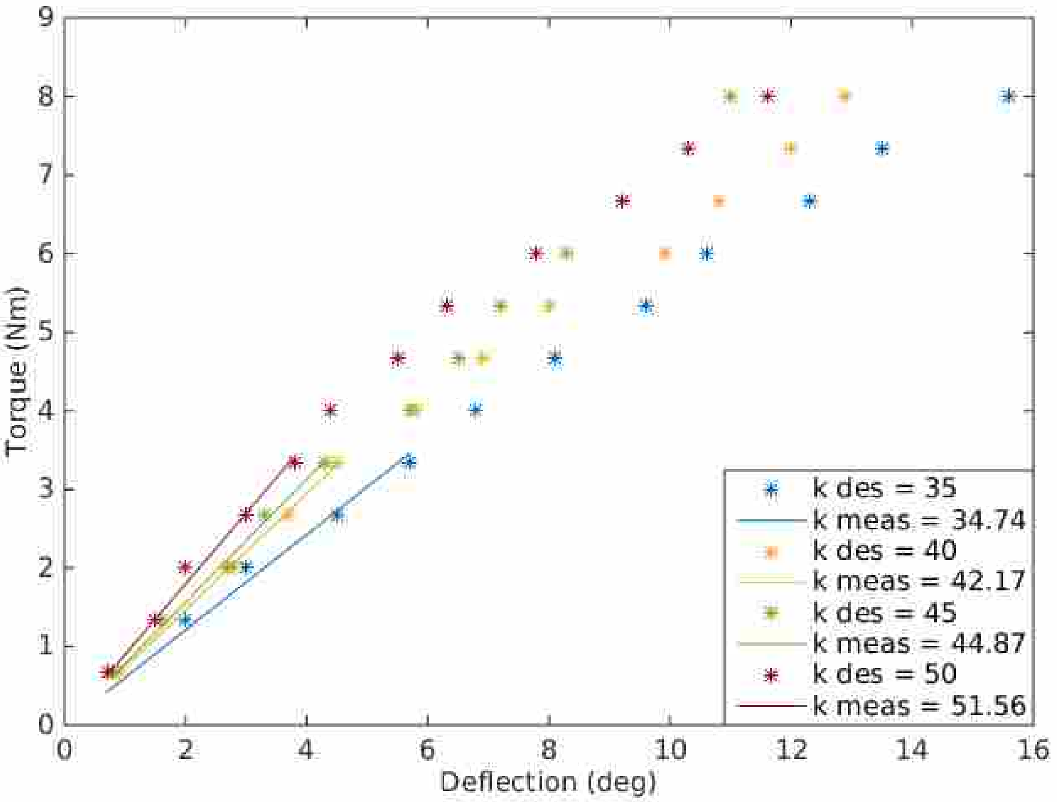

Figure 3.10: Linear region stiffness validation at 0°

CHAPTER 4. SIMULTANEOUS POSITION AND STIFFNESS CONTROL

This chapter looks at different control methods for controlling both position and stiffness simultaneously. The stiffness equations in the model developed in Chapter 3 are nonlinear so a sliding mode control law is formulated in Section 4.1. The sliding mode control (SMC) law was implemented both in simulation and on the grub by dividing the system into an angle controller and a torque and stiffness controller. For implementation on the grub, position was controlled within 2° of the desired angle, and stiffness was controlled within 3 Nm/rad at steady state. In Section 4.2, linear model predictive control (MPC) is formulated and was then implemented in simulation and on the grub. Position was controlled within 2° of the desired angle and stiffness was controlled within 1 Nm/rad of the desired stiffness at steady state. Both of the numerical results for SMC and MPC exclude the step responses where the desired stiffness and position form an unreachable pair based on joint angle and pressure limits and become competing objectives. In Section 4.3 SMC and MPC results are discussed and based on the performance, MPC is applied to multiple degrees of freedom in Section 4.4. Applying stiffness control to King Louie's right arm, it was shown that the deflection of the end effector was reduced by 50% when controlled near maximum stiffness rather than minimum stiffness which validates that stiffness is actually being controlled and has a significant effect on the system.

4.1 Sliding Mode Control

Sliding mode control (SMC) is a nonlinear control method where a discontinuous control signal is applied to a system to force the dynamics to slide along a subset of possible states. As described in [50], SMC works by first constraining a state trajectory to a manifold or surface and then driving the states to zero. The sliding surface is an arbitrary function of the states and may include system parameters. The control law is formed by looking at the stability of the surface in terms of Lyapunov and by picking the system input to be of the form

$$\mathbf{u} = -\beta(\mathbf{x})sgn(\mathbf{s}), \tag{4.1}$$

where $\mathbf{u}$ is the system inputs, $\beta(\mathbf{x})$ is a function of the system states and $sgn(\mathbf{s})$ is the signum function as a function of the sliding surface. The signum function is discontinuous and takes the form $sgn(\mathbf{s}) = \frac{\mathbf{s}}{||\mathbf{s}||}$ If a control law in the form of Equation 4.1 can be found that shows Lyaponuv stability, then sliding mode control has been established.

4.1.1 Controller Development

There are different procedures for developing a SMC control law. This work looks at the method described in [50] and [51]. Applying the method in [50] to the entire model (Equations 3.17 through 3.21) led to an uncontrollable mode. However, by dividing the model into two systems in series, an effective SMC control law was developed.

Method 1

This section shows how the standard method used [50] to establish a sliding mode control law led to an uncontrollable mode. Transforming the system model (Equations 3.17 through 3.21) into error states $\mathbf{x} = [\dot{\theta} - \dot{\theta}_{goal}, \theta - \theta_{goal}, k - k_{goal}, P_0, P_1]^T$ with inputs $\mathbf{u} = [P_{0,des}, P_{1,des}]^T$ gives the following:

$$\begin{aligned}
\dot{x}_1 &= \frac{k_d}{I}\left(x_1 - \dot{\theta}_{goal}\right) + \frac{k_s}{I}\left(x_2 - \theta_{goal}\right) + \frac{a_0}{I}x_4 + \frac{a_1}{I}x_5 && (4.2)\\
\dot{x}_2 &= x_1 && (4.3)\\
\dot{x}_3 &= c_0\left(x_3 + k_{goal}\right) + c_1 a_0 \alpha_0 u_0 + c_1 a_1 \alpha_1 u_1 - \left(x_1 + \dot{\theta}_{goal}\right)\left(a_0 \alpha_0^2 u_0 + a_1 \alpha_1^2 u_1\right) - \dot{k}_{goal} && (4.4)\\
\dot{x}_4 &= c_0 x_4 + c_1 u_0 && (4.5)\\
\dot{x}_5 &= c_0 x_5 + c_1 u_1. && (4.6)
\end{aligned}$$

This set of equations fits the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\left[\mathbf{G}(\mathbf{x})\mathbf{u} + \delta(\mathbf{t}, \mathbf{x}, \mathbf{u})\right], \tag{4.7}$$

where

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ a_0\alpha_0 & a_1\alpha_1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} c_1 & 0 \\ 0 & c_1 \end{bmatrix}. \tag{4.9}$$

The term $\mathbf{f}(\mathbf{x})$ contains the nonlinear, unforced dynamics and $\delta(\mathbf{t}, \mathbf{x}, \mathbf{u})$ contains the model uncertainties. A diffeomorphism of the system or a continuously differentiable map with a continuously differentiable inverse as defined by [50], is formed so that

$$\frac{\partial \mathbf{T}}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x}) = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}, \tag{4.10}$$

where if $\mathbf{B}(\mathbf{x})^{n \times m}$, then $\mathbf{I}$ is an m by m matrix. The most obvious choice for $\mathbf{T}$ is

$$\begin{aligned} T_1 &= x_1 \\ T_2 &= x_2 \\ T_3 &= x_3 - a_0\alpha_0 x_4 - a_1\alpha_1 x_5 \\ T_4 &= x_4 \\ T_5 &= x_5. \end{aligned} \tag{4.11}$$

As described in [50], the change of variables

$$\begin{bmatrix} \eta \\ \cdots \\ \xi \end{bmatrix} = \mathbf{T}(\mathbf{x}) = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \cdots \\ T_4 \\ T_5 \end{bmatrix}, \ \eta \in R^{n-m}, \xi \in R^m, \tag{4.12}$$

transforms the system into the form

$$\dot{\eta} = \mathbf{f}_a(\eta, \xi), \quad \dot{\xi} = \mathbf{f}_b(\eta, \xi) + \mathbf{G}(\mathbf{x})\mathbf{u} + \delta(\mathbf{t}, \mathbf{x}, \mathbf{u}). \tag{4.13}$$

The sliding manifold is then picked to be $s = \xi - \phi(\eta)$ where $\phi(\eta)$ is picked so that when the states are restricted to the sliding surface, the reduced-order model $\dot{\eta} = \mathbf{f}_a(\eta, \phi(\eta))$ is asymptotically stable at the origin. Differentiating η with respect to time and doing the change of variables from $\mathbf{x}$ to η and ξ gives

$$\begin{aligned} \dot{\eta}_1 &= \frac{a_0}{I}\xi_1 + \frac{a_1}{I}\xi_2 + \frac{k_d}{I}\left(\eta_1 + \dot{\theta}_{goal}\right) + \frac{k_s}{I}\left(\eta_2 + \theta_{goal}\right) - \ddot{\theta}_{goal} \\ \dot{\eta}_2 &= \eta_1 \\ \dot{\eta}_3 &= c_0\left(\eta_3 + k_{goal}\right) - \dot{k}_{goal}. \end{aligned} \tag{4.14}$$

It is impossible to pick $\phi(\eta)$ so that the reduced order model is asymptotically stable. The reason for this is found when $\dot{\eta}_3$ is expanded. First substitute in $\eta_3 = x_3 - a_0\alpha_0 x_4 - a_1\alpha_1 x_5$ into Equation 4.15 to get

$$\dot{\eta}_3 = c0\left(x_3 - a_0\alpha_0 x_4 - a_1\alpha_1 x_5 + k_{goal}\right) - \dot{k}_{goal}. \tag{4.15}$$

Then substitute in $x_3 = k - k_{goal}$, $x_4 = P_0$, and $x_5 = P_1$ to get

$$\dot{\eta}_3 = c0\left(k - k_{goal} - a_0\alpha_0 P_0 - a_1\alpha_1 P_1 + k_{goal}\right) - \dot{k}_{goal}. \tag{4.16}$$

Then substitute the stiffness expression Equation 3.13.

$$\dot{\eta}_3 = c0\left(k - k_{goal} - k + k_{goal}\right) - \dot{k}_{goal} \tag{4.17}$$

Simplifying results in

$$\dot{\eta}_3 = -\dot{k}_{goal}. \tag{4.18}$$

Since $\dot{\eta}_3$ is only dependent on $\dot{k}_{goal}$, the derivative is arbitrary and asymptotic stability is not possible because no choice of $\phi(\eta)$ will guarantee that $\dot{\eta}_3$ will go to zero. This shows that the diffeomorphism $\mathbf{T}(\mathbf{x})$ chosen leads to an uncontrollable mode. This does not imply that there does not exist a diffeomorphism that does lead to asymptotic stability of the subsystem, but that the obvious transformation does not work. It will be shown in the next section that when successive loop closure is used, finding a transformation is not necessary and that control can be achieved.

Cascade Model and Successive Loop Closure

Representing the system as a cascade and using successive loop closure allows for the development of a sliding mode control law without the need for a transformation. Figure 4.1 is the block diagram of the two controller system. The SMC Angle Control block in the diagram determines a torque in order to achieve the desired angle. The desired torque (τ_{des}) and desired stiffness (k_{goal}) is fed into the inner loop SMC Torque and Stiffness Control where desired pressures ($P_{0,des}$ and $P_{1,des}$) are determined to achieve the desired torque and stiffness. To determine valve commands (CMD), the desired pressures are then fed into a low level PID pressure controller that is running at 1000 Hz. The outer loop with the sliding mode control runs at 100 Hz. The same Kalman filter used in Section 2.2.1 is used to estimate angular velocity ($\dot{\theta}$) and angle (θ) states. These are used with pressure data (P_0 and P_1) in Equation 3.13 to estimate the stiffness state (k) for the time step.

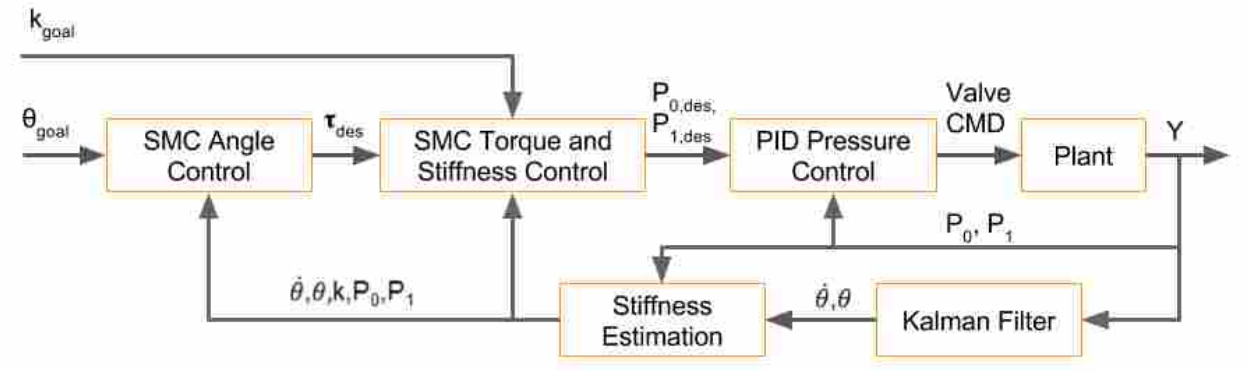

Figure 4.1: SMC flow chart

Position Through Torque Control

The first SMC designated as SMC Angle Control in Figure 4.1, determines the necessary torque to track a desired angle signal. The output from this controller is a desired torque that the second SMC, designated as SMC Torque and Stiffness Control in Figure 4.1, will track. The second SMC determines pressures necessary to track both a desired torque and a desired stiffness signal. Without a second controller, the desired torque output from the first SMC is useless since pressure is controlled and not torque directly.

To derive the SMC Angle Control, equations similar to state Equations 3.17 and 3.18 from the system model are used. The difference is that torque is considered instead of pressure states. This is essentially a controller for a simple rotary joint with torques applied at the joint or τ_{des} taking the form

$$I\ddot{\theta} - k_d\dot{\theta} - k_s\theta = \tau_{des}. \tag{4.19}$$

The goal is to find a control law for τ_{des} to stabilize the joint. The torques applied at the joint are the controlled torque (τ_{smc}) and a returning spring torque ($-k_s\theta$) from the physical structure of the joint. To compensate for steady state error an integrator ($-K_i \int \left(\theta - \theta_{goal}\right)$) can be added so that

$$\tau_{des} = \tau_{smc} - k_s\theta_{goal} - K_i \int \left(\theta - \theta_{goal}\right). \tag{4.20}$$

Combining Equations 4.19 and 4.20 gives

$$\ddot{\theta} = \frac{k_d}{I}\dot{\theta} + \frac{k_s}{I}\left(\theta - \theta_{goal}\right) - \frac{K_i}{I}\int\left(\theta - \theta_{goal}\right). \tag{4.21}$$

A change of variables to error states where $\mathbf{x} = \left[\dot{\theta}, \theta - \theta_{goal}, \int\left(\theta - \theta_{goal}\right)\right]^T$ gives

$$\begin{aligned} \dot{x}_1 &= \frac{k_d}{I}x_1 + \frac{k_s}{I}x_2 - \frac{K_i}{I}x_3 + \frac{\tau_{smc}}{I} && (4.22)\\ \dot{x}_2 &= x_1 - \dot{\theta}_{goal} && (4.23)\\ \dot{x}_3 &= x_2 && (4.24) \end{aligned}$$

as a set of state equations. Using these equations the sliding surface was picked to be

$$s = x_1 + K_e x_2 + \frac{K_i}{I}x_3. \tag{4.25}$$

There is not a direct method for picking a surface but this surface follows the form of the sum of the states multiplied by coefficients. Taking the time derivative of Equation 4.25 and substituting in from Equations 4.22 through 4.22 gives

$$\dot{s} = \frac{k_d}{I}x_1 + \frac{k_s}{I}x_2 - \frac{K_i}{I}x_3 + \frac{\tau_{smc}}{I} + K_e\left(x_1 - \dot{\theta}_{goal}\right) + \frac{K_i}{I}x_2, \tag{4.26}$$

where K_e and K_i are SMC tunable gains. The control input τ_{smc} is then picked to be

$$\tau_{smc} = -I\left(K_e x_1 + \frac{K_i}{I}\left(x_2 - x_3\right) + \nu\right) \tag{4.27}$$

to stabilize the system where ν contains the model uncertainties. If this is substituted into Equation 4.26, it becomes

$$\dot{s} = -\nu + \frac{k_d}{I}x_1 + \frac{k_s}{I}x_2 - K_e\dot{\theta}_{goal}. \tag{4.28}$$

The goal is to have $\dot{s}$ go to zero by sliding along the manifold described in Equation 4.25. This is done by picking ν to be

$$\nu = \beta(\mathbf{x}) \frac{s}{||s||}. \tag{4.29}$$

The term $\beta(\mathbf{x})$ is determined by finding a bound on $|\nu|$ which is done by starting with the ν that makes $\dot{s} = 0$. The bound on $|\nu|$ can be found from the following:

$$\begin{aligned}
\nu &= -K_e\dot{\theta}_{goal} + \frac{k_s}{I}x_2 + \frac{k_d}{I}x_1 && (4.30)\\
|\nu| &= |-K_e\dot{\theta}_{goal} + \frac{k_s}{I}x_2 + \frac{k_d}{I}x_1| && (4.31)\\
|\nu| &\leq K_e|\dot{\theta}_{goal}| + \frac{k_{s,max}}{I}|x_2| + \frac{k_{d,max}}{I}|x_1|. && (4.32)
\end{aligned}$$

Because it was shown that $|\nu|$ is bounded, the sliding mode control law can be formed by

$$\tau_{smc} = -I\left(K_e x_1 + \frac{K_i}{I}(x_2 - x_3) + \beta(\mathbf{x})\, sat\left(\frac{s}{\mu}\right)\right), \tag{4.33}$$

where

$$\beta(\mathbf{x}) = K_e|\dot{\theta}_{goal}| + \frac{k_{d,max}}{I}|x_1| + \frac{k_{s,max}}{I}|x_2| + \beta_0. \qquad k_{s,max}, k_{d,max}, \beta_0, \mu \geq 0 \tag{4.34}$$

and $sat\left(\frac{s}{\mu}\right)$ is the saturation function which is an approximation of $\frac{s}{||s||}$ from Equation 4.29. The terms μ and β_0 are tunable controller parameters. Now that there is a control law for the control torque (τ_{smc}), the desired torque for the joint can be calculated at each time step using Equation 4.20.

Torque and Stiffness Control

The goal of the second SMC is to track the desired torque (τ_{des}) from the first controller and a desired stiffness (k_{goal}) from a reference signal. The torque at the joint is defined as

$$\tau = a_0 P_0 + a_1 P_1. \tag{4.35}$$

Taking the time derivative of Equation 4.35 and using Equations 3.19 through 3.21 gives the following model for torque and stiffness:

$$\begin{aligned}
\dot{\tau} &= c_0\tau + a_0c_1P_{0,des} + a_1c_1P_{1,des} && (4.36)\\
\dot{k} &= c_0k + a_0\alpha_0c_1P_{0,des} + a_1\alpha_1c_1P_{1,des} - \dot{\theta}\left(a_0\alpha_0^2P_0 + a_1\alpha_1^2P_1\right) && (4.37)\\
\dot{P}_0 &= c_0P_0 + c_1P_{0,des} && (4.38)\\
\dot{P}_1 &= c_0P_1 + c_1P_{1,des}. && (4.39)
\end{aligned}$$

Equations 4.36 and 4.37 form a two-input two-output model so a sliding mode control approach can be used to control both joint torque and stiffness. This method is the same approach used in [15] for force and stiffness control of a linear pneumatic actuator, but the control law in this work is for a rotary actuator rather than a linear actuator.

The sliding surface is first defined as the error between the joint torque (τ) and the desired torque (τ_{des}) defined as

$$S_1 = \tau - \tau_{des}, \qquad (4.40)$$

and the error between the joint stiffness (k) and the desired joint stiffness (k_{goal}) defined as

$$S_2 = k - k_{goal}. \qquad (4.41)$$

As described in [51] the MIMO sliding mode control law is given as

$$\mathbf{u} = \mathbf{B}^{-1}\left(\dot{\mathbf{x}}_r - \mathbf{f} - \mathbf{K}sat\left(\frac{\mathbf{S}}{\mu}\right)\right), \qquad (4.42)$$

where $\mathbf{u}$ is the system inputs, $\mathbf{B}$ contains input dynamics, $\dot{\mathbf{x}}_r$ is the derivative of the input trajectories which can be calculated using numerical methods, $\mathbf{f}$ contains the nonlinear state dynamics, $\mathbf{K}$ is the controller gains and $sat\left(\frac{\mathbf{S}}{\mu}\right)$ is the saturation function of the manifold signals where the saturation function is defined as

$$sat(y) \quad = \quad \begin{cases} y & if\ ||y|| \leq 1 \\ \frac{y}{||y||} & if\ ||y|| > 1. \end{cases} \tag{4.43}$$

Putting Equations 4.36 and 4.37 into the form of Equation 4.42 gives the following control law:

$$\begin{bmatrix} P_{0,des} \\ P_{1,des} \end{bmatrix} = \begin{bmatrix} c_1 a_0 & c_1 a_1 \\ c_1 a_0 \alpha_0 & c_1 a_1 \alpha_1 \end{bmatrix}^{-1} \left[\begin{bmatrix} \dot{\tau}_{des} \\ \dot{k}_{goal} \end{bmatrix} - \begin{bmatrix} c_0 \tau \\ c_0 k - \dot{\theta}\left(a_0 \alpha_0^2 P_0 + a_1 \alpha_1^2 P_1\right) \end{bmatrix} \right. \tag{4.44}$$
$$\left. - \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} \begin{bmatrix} sat\left(\frac{S_1}{\mu}\right) \\ sat\left(\frac{S_2}{\mu}\right) \end{bmatrix} \right].$$

4.1.2 Simulation Results

Both sliding mode controllers were shown to be successful in simulation separately and then together for overall control of both angle and stiffness. The SMC Angle Control for position was able to track steps in desired position and the SMC Torque and Stiffness Control was able to track steps in both desired torque and stiffness. SMC for the system was simulated in MATLAB. Forward simulations are based on an explicit Runge-Kutta method based on the Dormand-Prince pair [55] by using MATLAB's ode45 function.

SMC Angle Control Simulation

The first simulation was SMC Angle Control where the output τ_{des} was used in a plant simulated using only Equation 4.19. The controller gains for angle control are in Table 4.1 and they were tuned manually for good performance. The controller response to a series of step commands can be seen in Figure 4.2 and how the sliding surface evolves can be seen in Figure 4.3. The joint angle response behaves exactly how it was expected. The desired torque spikes and pulls back just before reaching the desired angle. The plot of the sliding surface s in Figure 4.3 goes to zero after each step which shows stability of the system.

Table 4.1: Controller gains for independent angle controller

	Gain
K_e	10.0
K_i	0.1
β_0	1.0
μ_1	10.0
$k_{d,max}$	15.0
$k_{s,max}$	60.0

SMC Torque and Stiffness Control Simulation

For simulation of SMC Torque and Stiffness control, arbitrary step commands in desired torques (τ_{des}) and desired stiffnesses (k_{goal}) were used as reference trajectories and the plant was simulated using Equations 4.36 through 4.39. The controller gains for torque and stiffness control are in Table 4.2 and they were tuned manually for good performance with a higher priority on torque tracking. Good tracking of the desired torque signal is necessary for good angle tracking. The controller response to a series of step commands can be seen in Figure 4.4 and how the sliding surface evolves can be seen in Figure 4.5. The spikes in desired pressure are expected but because of the first order pressure model, these spikes have little effect on the pressure dynamics. There is good tracking of both the desired torque signal and the desired stiffness signal. For high torques and high stiffnesses such as the region from 50 seconds to 70 seconds, the tuning preference for torque tracking can be seen. The model used for torque, Equation 3.8, allows for the high torques seen in Figure 4.4. The actual torque applied by the joint is expected to be less than the estimated value. In Figure 4.5 both S values trend to zero which shows that sliding mode control is stabilizing the system.

Table 4.2: Controller gains for independent torque and stiffness controller

	Gain
K_1	30.0
K_2	4.0
μ_2	10.0

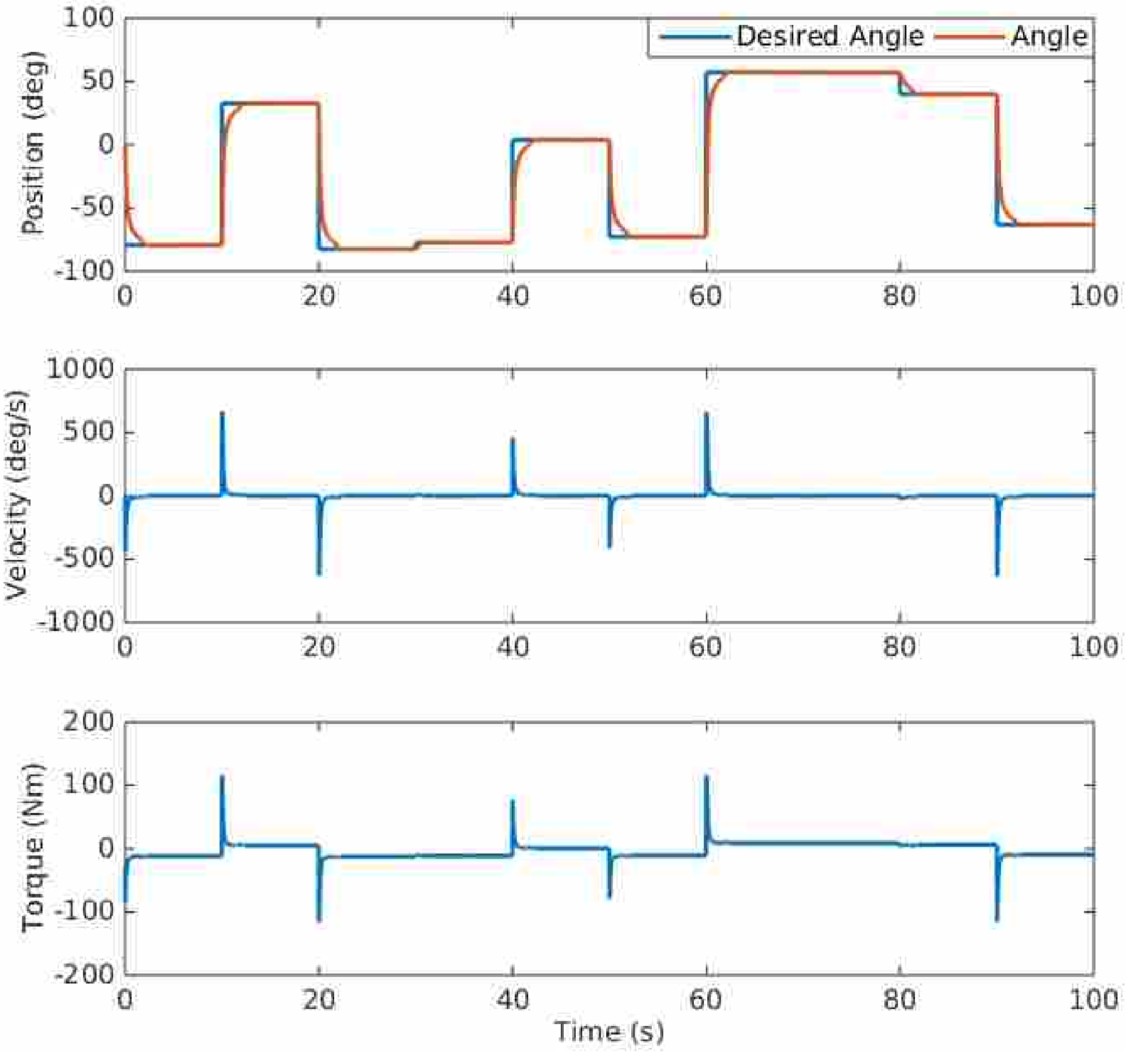

Figure 4.2: Simulated independent angle control response

Full System SMC Simulation

For full state simulation arbitrary angle and stiffness goals as step commands are used and the plant is simulated using Equations 3.17 through 3.21. The controller gains for angle and stiffness control are in Table 4.3. The controller gains in this table are the same as the gains in Tables 4.1 and 4.2. This shows that the dynamics between the two controllers do not change much. The controller response to a series of step commands can be seen in Figure 4.6 and how the sliding surface evolves can be seen in Figure 4.7. There is good tracking for both position and stiffness

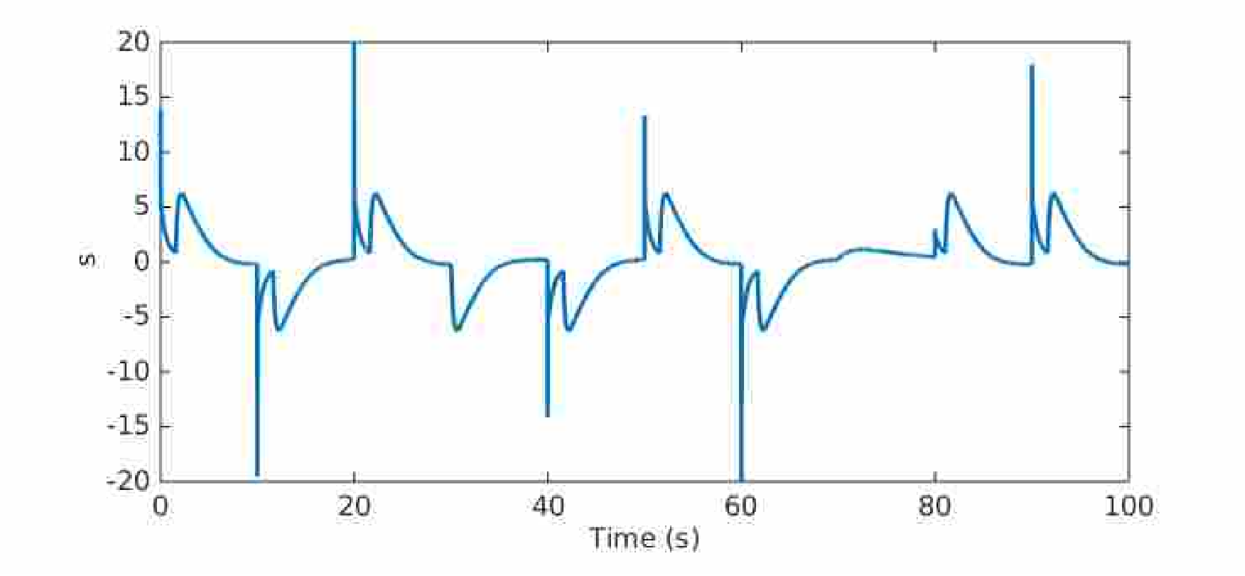

Figure 4.3: Sliding surface for simulatied independent angle control

which validates the use of the successive loop closure control system. There are again spikes that occur in desired torque signal which gets passed onto the desired pressure signal. In simulation, these spikes have little effect on the system states. In Figure 4.7, again all three surfaces are driven to zero showing stability of the system.

Table 4.3: Controller gains for sliding mode control in simulation

	Gain
K_e	10.0
K_i	0.1
β_0	1.0
μ_1	10.0
$k_{d,max}$	15.0
$k_{s,max}$	60.0
K_1	30.0
K_2	4.0
μ_2	10.0

Simulation results show that the successive loop closure sliding mode control scheme can work to control both joint angle and stiffness. In Figure 4.7 the three sliding surfaces trend to zero which is one of the goals for sliding mode control. The pressure states and desired pressures are

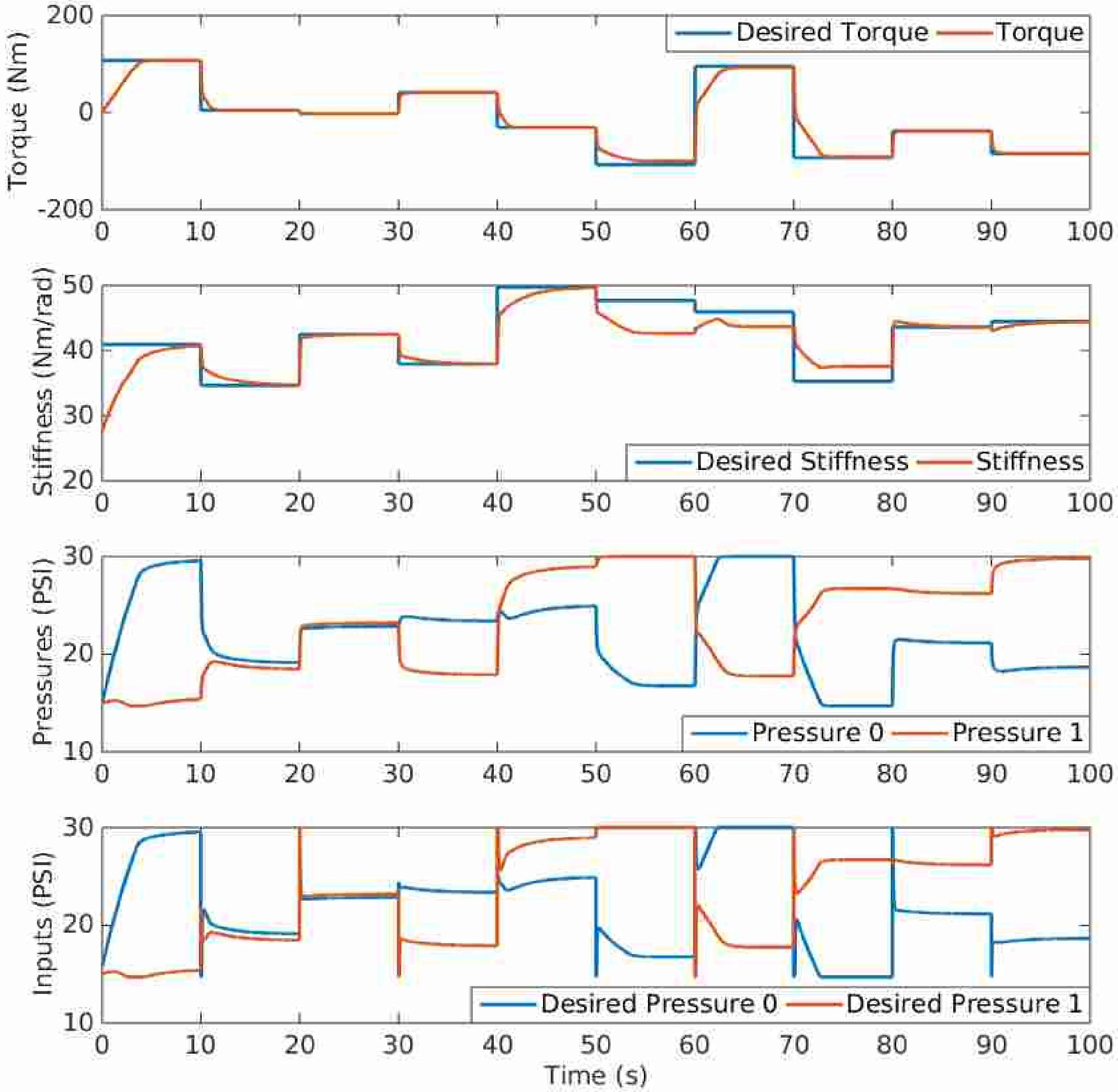

Figure 4.4: Simulated independent torque and stiffness control response

within the range of achievable pressures for the grub. Actual control on the grub is not expected to perform as well as simulation because the simulation results use the developed model for the plant dynamics. In order to prevent integrator wind up, the integrator was set to zero for any absolute error greater than 5° so that the integrator starts at zero for each new step command. This also prevents windup before the integrator initially moves toward the goal from a step in the reference signal.

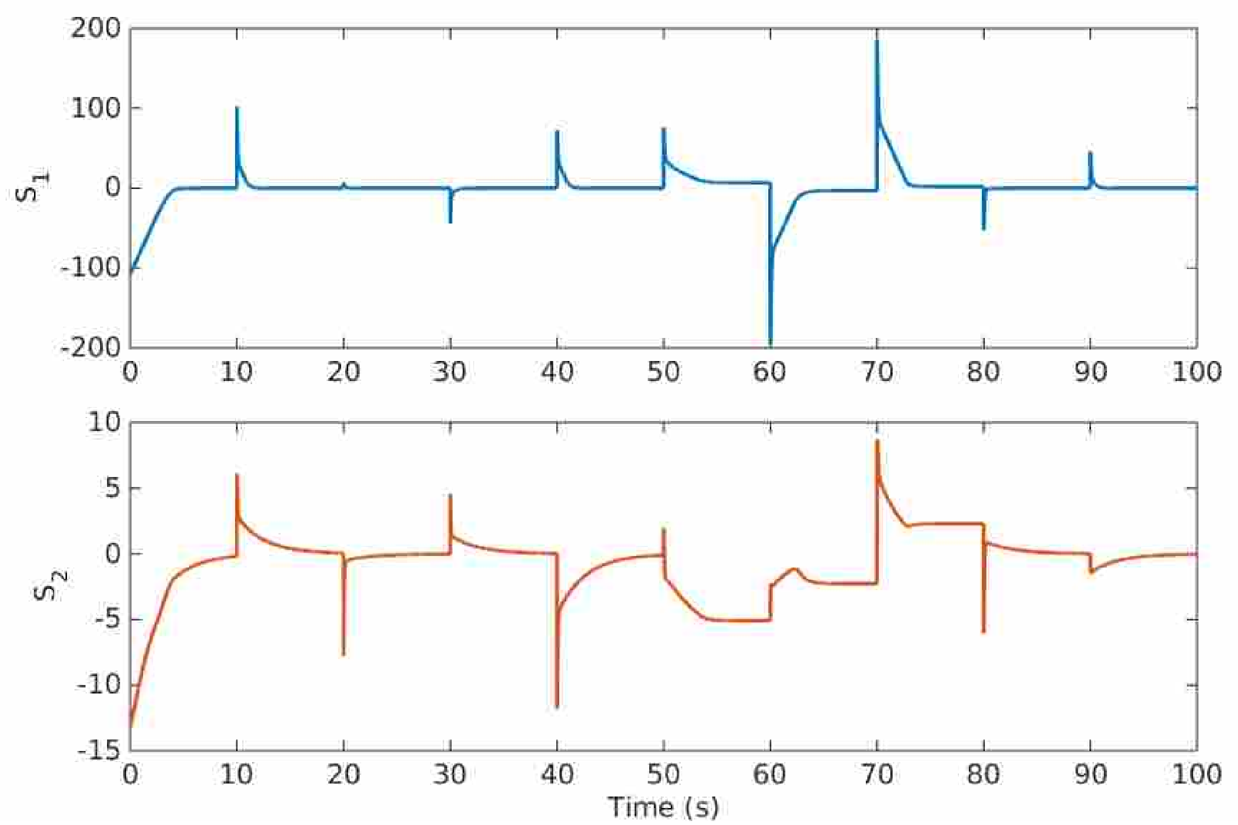

Figure 4.5: Sliding surfaces for simulated independent torque (S_1) and stiffness (S_2) control

4.1.3 Grub Implementation

Since it was shown in simulation that SMC worked for the system model, SMC was applied to the one degree of freedom grub platform seen in Figure 2.1. The controller gains for angle and stiffness control used on the grub are in Table 4.4 and they were tuned manually with position control being weighted more important. The grub response to a series of step commands can be seen in Figure 4.8 and how the sliding surface evolves can be seen in Figure 4.9. The preference towards position tracking can be easily seen in Figure 4.8 especially from 60 seconds to 75 seconds. During this time, the angle is achieved but actuator pressures are saturated in a way that prevents stiffness from increasing without changing the joint angle. The steady state error for position is within 2° of the desired angle and the stiffness was controlled to within 3 Nm/rad of the desired stiffness signal at steady state except for the region between 60 seconds and 75 seconds.

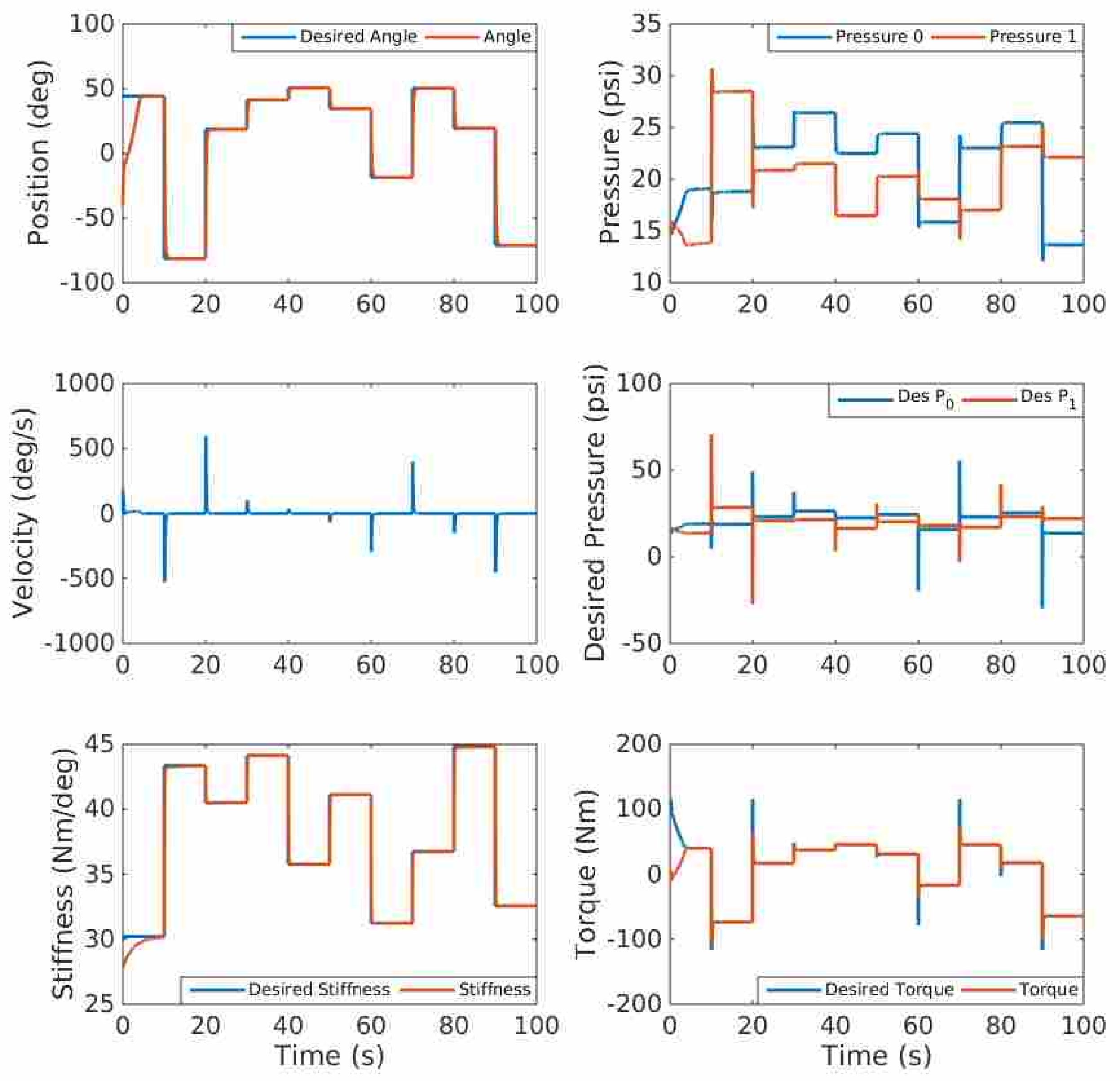

Figure 4.6: Simulated angle and stiffness control response

The differences between the simulated SMC as seen in Figure 4.6 and actual SMC on the grub as seen in Figure 4.8 are quite apparent. In simulation, both angle and stiffness are tracked quickly and accurately. Even the desired torque is tracked quite well.

Actual control on the grub is not as good as simulation which is to be expected. There is significant overshoot in the angle tracking and there is poor tracking of the stiffness signal. The poor stiffness tracking is the cost for the accuracy in tracking the torque trajectory which is necessary for accurate angle tracking. The controller was tuned in this way because for many

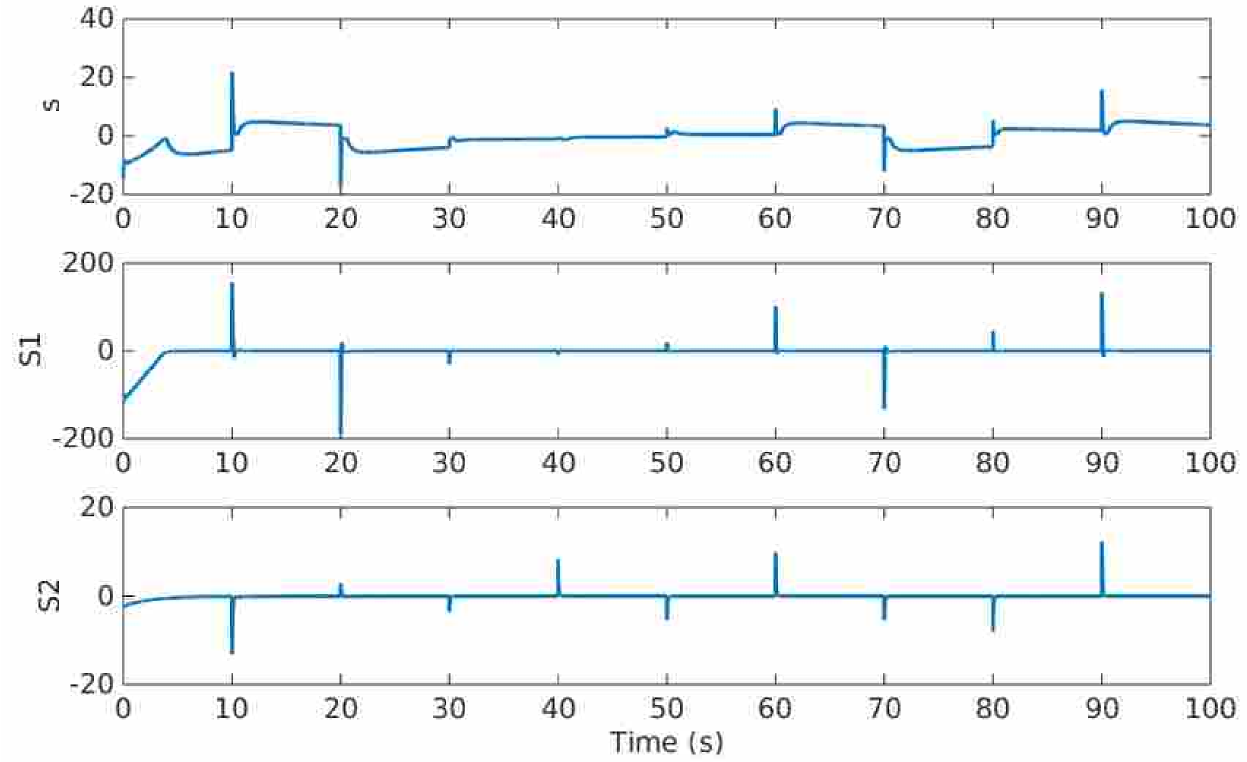

Figure 4.7: This is the sliding surfaces for simulation angle and stiffness control. s corresponds to Equation 4.25, S_1 corresponds to Equation 4.40 and S_2 corresponds to Equation 4.41.

Table 4.4: Controller gains for sliding mode control on the grub

	Gain
K_e	10.0
K_i	0.5
β_0	1.0
μ_1	200.0
$k_{d,max}$	15.0
$k_{s,max}$	60.0
K_1	30.0
K_2	8.0
μ_2	10.0

applications, position accuracy is very important to complete a task. Even though the stiffness tracking on the grub does not have as fast of a response time as the stiffness tracking in simulation, given enough time the desired stiffness is achieved for most step commands after the desired angle has been achieved.

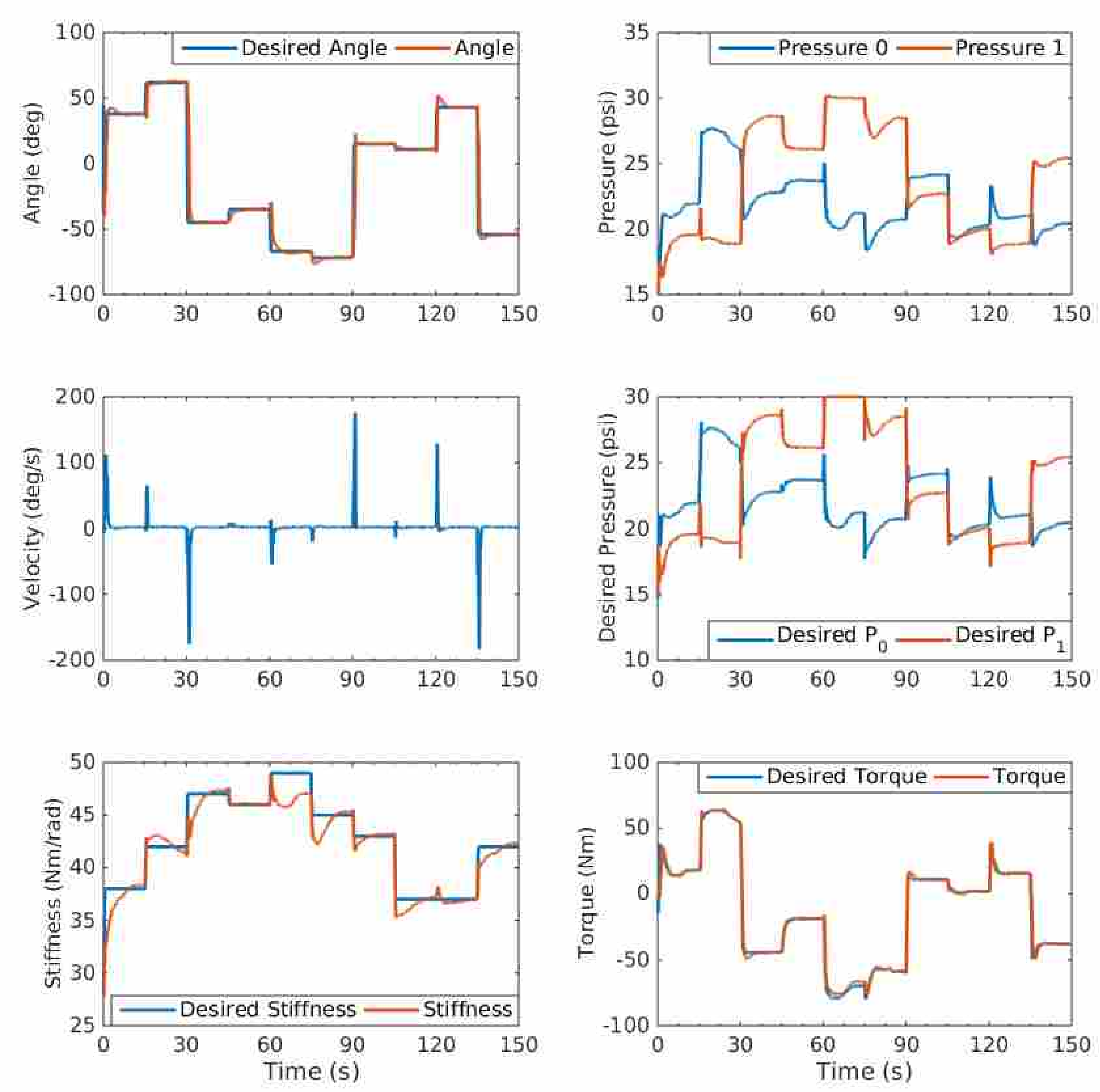

Figure 4.8: SMC response on the grub

The grub is more sensitive to changes in the commanded pressure than the model predicts when the valves first open or close. There is a sudden change of pressure in the actuators probably due to fluid capacitance within the lines. As seen from simulation in Figure 4.6, large control spikes up to 100 Nm in magnitude per time step occur when there is a change in the desired angle trajectory. In order to prevent these spikes for SMC on the grub, a slew rate was implemented on

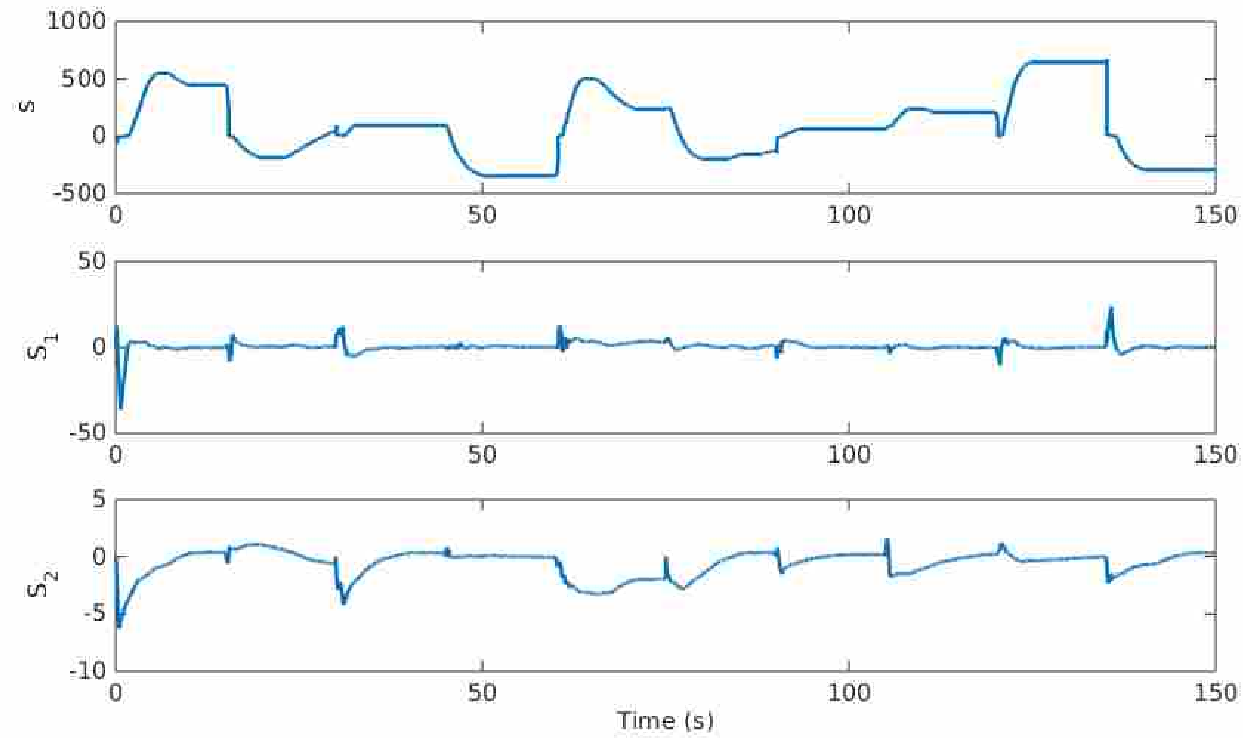

Figure 4.9: Sliding surfaces from SMC on the grub

the output of SMC to prevent the desired torque from changing too quickly. The slew rate was 100 Nm/rad or since the controller was running at 100 Hz, the slew rate was 1 Nm/rad.

To prevent overshoot from integrator wind up, just like in simulation, the integral was zeroed for error greater than 25° to restart the integrator for new large step commands and so that the integrator does not start to build until the position is close to the goal. The larger integrator region than simulation was necessary for actual control because of hysteresis in how the grub responds to different torques at different angles.

The controller performance can be greatly improved with a better model. Although there are robust features of SMC, this type of control is still susceptible to bad performance when there are errors in the model. For example, the model used does not include specific gas dynamics or how the actuators interact with each other. The performance of SMC on the grub shows that the model used captures a majority of the actual grub dynamics. A model that includes more of the gas and fabric dynamics would allow for the development of a better sliding mode control law, but would greatly increase the complexity of the controller and would require more complicated system identification for each new joint or actuator. The reason for this is because the gas and fabric dynamics are discontinuous and the controller would need to adapt to each dynamic mode. Choked flow and unchoked flow, filling or draining, pinched actuators or open actuators are all different modes that would require different control laws. Instead of capturing all of these details, a minimal set of dynamics where both joint angle and stiffness can be controlled was developed.

4.2 Model Predictive Control

The main idea of model predictive control is that system inputs are determined by minimizing a cost function over a finite time horizon subject to the dynamics of the system expressed as an equality constraint. This optimization occurs for every time step and only the first resultant control input is applied. This allows the controller to benefit from both closed-loop feedback control and optimal control. The ability to look forward at future states and using an updated trajectory from the current states at each time is what makes this control method good at overcoming disturbances and discrepancies in the model. The main limitations to MPC are the ability to find a solution to the optimization problem and solving the optimization at fast enough rates for control.

4.2.1 Controller Development

The controller development for MPC is much simpler than the controller development for SMC. The model predictive controller takes in the estimated states and uses a linearized and discretized version of the system model to determine commands for desired pressures at 30 Hz to track a desired angle and stiffness trajectory. The control rate was limited to how fast the optimizer could find a solution which on average was between 40 and 50 Hz. The desired pressures are determined by minimizing a cost function that accounts for the system states and inputs across a horizon. The desired pressures are fed into a PID pressure controller running at 1000 Hz to control the actuator pressures. A flow chart for this process can be seen in Figure 4.10.

4.2.2 Linearization and Discretization

Although there are solvers that are capable of minimizing a cost function for nonlinear dynamics, linearized dynamics have been chosen for faster solve times and for simplicity. System

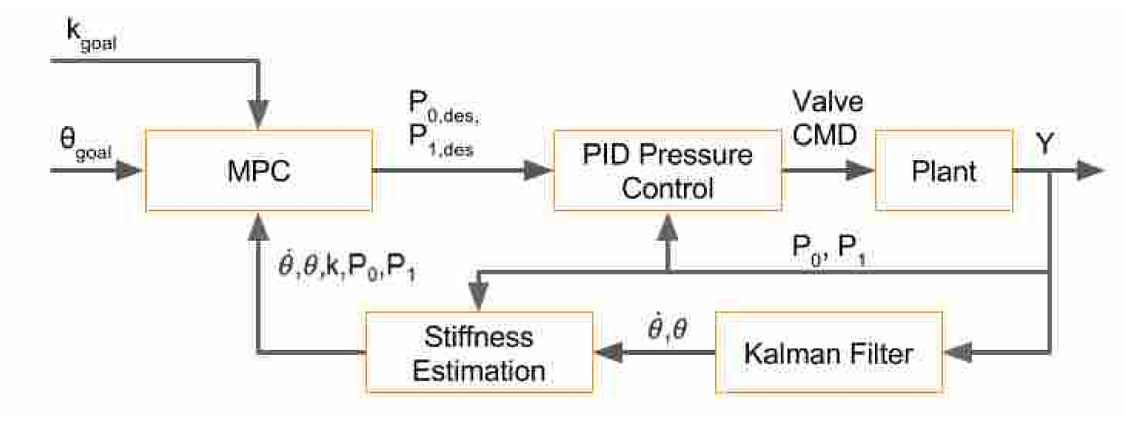

Figure 4.10: MPC flow chart

states are defined as $\mathbf{x} = \left[\dot{\theta}, \theta, k, P_0, P_1\right]^T$ and system inputs are defined as $\mathbf{u} = \left[P_{0,des}, P_{1,des}\right]$. Writing the dynamics in the form $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{u})$, the system unforced dynamics are

$$\mathbf{F} = \begin{bmatrix} \frac{k_d}{I}\dot{\theta} + \frac{k_s}{I}\theta + \frac{a_0}{I}P_0 + \frac{a_1}{I}P_1 \\ \dot{\theta} \\ c_0 k - \dot{\theta}\left(a_0\alpha_0^2 P_0 + a_1\alpha_1^2 P_1\right) \\ c_0 P_0 \\ c_0 P_1 \end{bmatrix}, \tag{4.45}$$

and the forced dynamics are

$$\mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ a_0\alpha_0 c_1 P_{0,des} + a_1\alpha_1 c_1 P_{1,des} \\ c_1 P_{0,des} \\ c_1 P_{1,des} \end{bmatrix}. \tag{4.46}$$

The linearized unforced response becomes

$$\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}}\bigg|_{\mathbf{x}_{init}, \mathbf{u}_{init}}, \tag{4.47}$$

and the linearized forced response becomes

$$\mathbf{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{u}}\bigg|_{\mathbf{x}_{init},\mathbf{u}_{init}}, \tag{4.48}$$

where $\mathbf{x}_{init}$ and $\mathbf{u}_{init}$ are the states and inputs that the system is linearized about. For MPC, $\mathbf{x}_{init}$ and $\mathbf{u}_{init}$ are the states and inputs from the previous time step. The linearized dynamics then becomes

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u}, \tag{4.49}$$

with derivatives $\Delta\dot{\mathbf{x}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_{init}$, states $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{init}$, and inputs $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_{init}$.

Discretizing the system allows using the system model to predict future states at the next time step with given inputs. One method to change to a discrete system is the matrix exponential which is formulated as

$$\mathbf{A}_d = e^{(\mathbf{A}\Delta t)}, \tag{4.50}$$

where Δt is the time step between discrete states. As long as $\mathbf{A}$ is invertible, the discrete form of the forced dynamics becomes

$$\mathbf{B}_d = \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I})\mathbf{B}, \tag{4.51}$$

where $\mathbf{I}$ is the identity matrix with the same dimensions as $\mathbf{A}$. The linearized discrete dynamics for the system model then becomes

$$\Delta\mathbf{x}[t+1] = \mathbf{A}_d\Delta\mathbf{x}[t] + \mathbf{B}_d\Delta\mathbf{u}[t], \tag{4.52}$$

where t is the discrete step index. The actual states and inputs of the system can then be calculated by adding the change in linearized state to the initial condition or

$$\begin{aligned} \mathbf{x}[t] &= \mathbf{x}_{init} + \Delta\mathbf{x}[t] \qquad (4.53) \\ \mathbf{u}[t] &= \mathbf{u}_{init} + \Delta\mathbf{u}[t]. \qquad (4.54) \end{aligned}$$

4.2.3 MPC Simulation

MPC for the system was simulated in MATLAB as proof of concept that MPC can control both stiffness and angle and to find a set of controller weights to use for actual implementation. Forward simulations are based on an explicit Runge-Kutta method based on the Dormand-Prince pair [55]. The control law for the system is the minimization of a cost function that is dependent on a desired trajectory, system states, and inputs. The cost function is minimized using MATLAB's fmincon function which is an interior-point algorithm [56–58] where the input bounds are atmospheric pressure 14.7 PSI (101.4 kPa) absolute and the source pressure which is regulated to 30 PSI (206.8 kPa) absolute. The results of the simulation can be seen in Section 4.2.3.

Cost Function

Using the system states $\mathbf{x}$ and inputs $\Delta\mathbf{u}$ from Equation 4.49, the cost function becomes

$$\begin{aligned}\min_{\Delta\mathbf{u}} J(\mathbf{x}) &= \sum_{t=1}^{T}\Big(||\dot{\theta}[t]||^2_{Q_{vel}} + ||\theta[t]-\theta_{goal}||^2_{Q_{pos}} + ||k[t]-k_{goal}||^2_{Q_{stf}} \\ &\quad + ||\Delta P_{0,des}[t]||^2_{R_0} + ||\Delta P_{1,des}[t]||^2_{R_1} + H(\Delta\mathbf{u}[t])\Big),\end{aligned} \tag{4.55}$$

where θ_{goal} is the desired angle, k_{goal} is the desired stiffness, Q_{vel} is the velocity weighting, Q_{pos} is the position error weighting, Q_{stf} is the stiffness error weighting, R_0 is the weighting on the change in $P_{0,des}$, and R_1 is the weighting on the change in $P_{1,des}$. To prevent ocillations, a cost is added for large changes in the inputs. This cost is defined by

$$H(\Delta\mathbf{u}[t]) = \sum_{i=0}^{1} h\left(\Delta P_{i,des}[t]\right), \tag{4.56}$$

and

$$
\begin{aligned}
\textbf{if}: &\Delta P_{i,des}[t] > h_r \\
&h\left(\Delta P_{i,des}[t]\right) \;=\; \left(\Delta P_{i,des}[t] - h_r\right)^2 \\
\textbf{else if}: &\Delta P_{i,des}[t] < -h_r \\
&h\left(\Delta P_{i,des}[t]\right) \;=\; \left(\Delta P_{i,des}[t] + h_r\right)^2. \\
\textbf{else}: & \\
&h\left(\Delta P_{i,des}[t]\right) \;=\; 0
\end{aligned}
\tag{4.57}
$$

The parameter h_r is a tunable value that determines how big $\Delta P_{0,des}$ or $\Delta P_{1,des}$ can be before cost starts to be accrued. Before $\Delta P_{0,des}$ or $\Delta P_{1,des}$ get too big, no cost is added from this term in the cost function. The cost function Equation 4.55 is subject to:

$$\Delta\mathbf{x}[t+1] = \mathbf{A}_d \Delta\mathbf{x}[t] + \mathbf{B}_d \Delta\mathbf{u}[t] \quad t = 1..T \tag{4.58}$$

$$\mathbf{x}[t] = \mathbf{x}_{init} + \Delta\mathbf{x}[t] \quad t = 1..T \tag{4.59}$$

$$\theta_{min} \leq \theta[t] \leq \theta_{max} \quad t = 1..T \tag{4.60}$$

$$P_{min} \leq P_i[t] \leq P_{max}, \quad t = 1..T \tag{4.61}$$

where θ_{min} and θ_{max} are joint limits.

Results

The weights used in the cost function are in Table 4.5 and the simulation was run at 30 Hz with a horizon of five steps. The control rate and horizon length for simulation were determined by the possible control rate and horizon length of actual implementation on the grub. Actual implementation is limited by how fast the solver can find a solution where this is not a constraint in simulation. Ten random desired angle and stiffness step commands with a duration of 10 seconds were sent to the controller. The results of the simulation can be seen in Figure 4.11.

Table 4.5: Controller gains for model predictive control for grub simulation

Q_{pos}	50.0
Q_{stf}	.2
Q_{vel}	2.0
R_0	0.05
R_1	0.05
h_r	$2.298e3$
T	5
rate	30 Hz
θ_{min}	-120°
θ_{max}	120°

In simulation both angle and stiffness are controlled quite well for random desired angles and stiffnesses. From 20 seconds to 30 seconds in Figure 4.11 neither the desired angle or stiffness is achieved because of system limits. Large angles with either high or low stiffness is hard for the system to achieve. This can be seen in the fast oscillations in the command signal for Desired P_1 in this duration. A similar system limit effect is seen in from 40 to 50 seconds. Higher weights for either stiffness or angle error will give better tracking at the expense of poorer tracking of the other. Increasing the pressure limits would allow for better position and stiffness tracking but the current limits are set based on the current hardware design.

There was significant oscillations in the control signal immediately following the step input. The reason for this is most likely the linearized dynamics which could be rectified by using a nonlinear solver. These oscillations were not present in simulation when using APMonitor [46] as a nonlinear solver but the solve rate was between 1 and 5 Hz. APMonitor is optimization software that uses linear, nonlinear, and mixed integer programming solvers specifically built for problems like nonlinear MPC. The solver used in MATLAB for the linearized dynamics has a comparable solve rate between 1 and 6 Hz. However, as will be shown later, the solve rate for actual control on the grub using a Python solver with a different algorithm was between 40 and 50 Hz. APMonitor will have the same solve rate whether used in MATLAB or Python so using it as a nonlinear solver at this time is not feasible. APMonitor did however outperform a Python nonlinear solver which took 3 to 5 seconds to find a solution per time step.

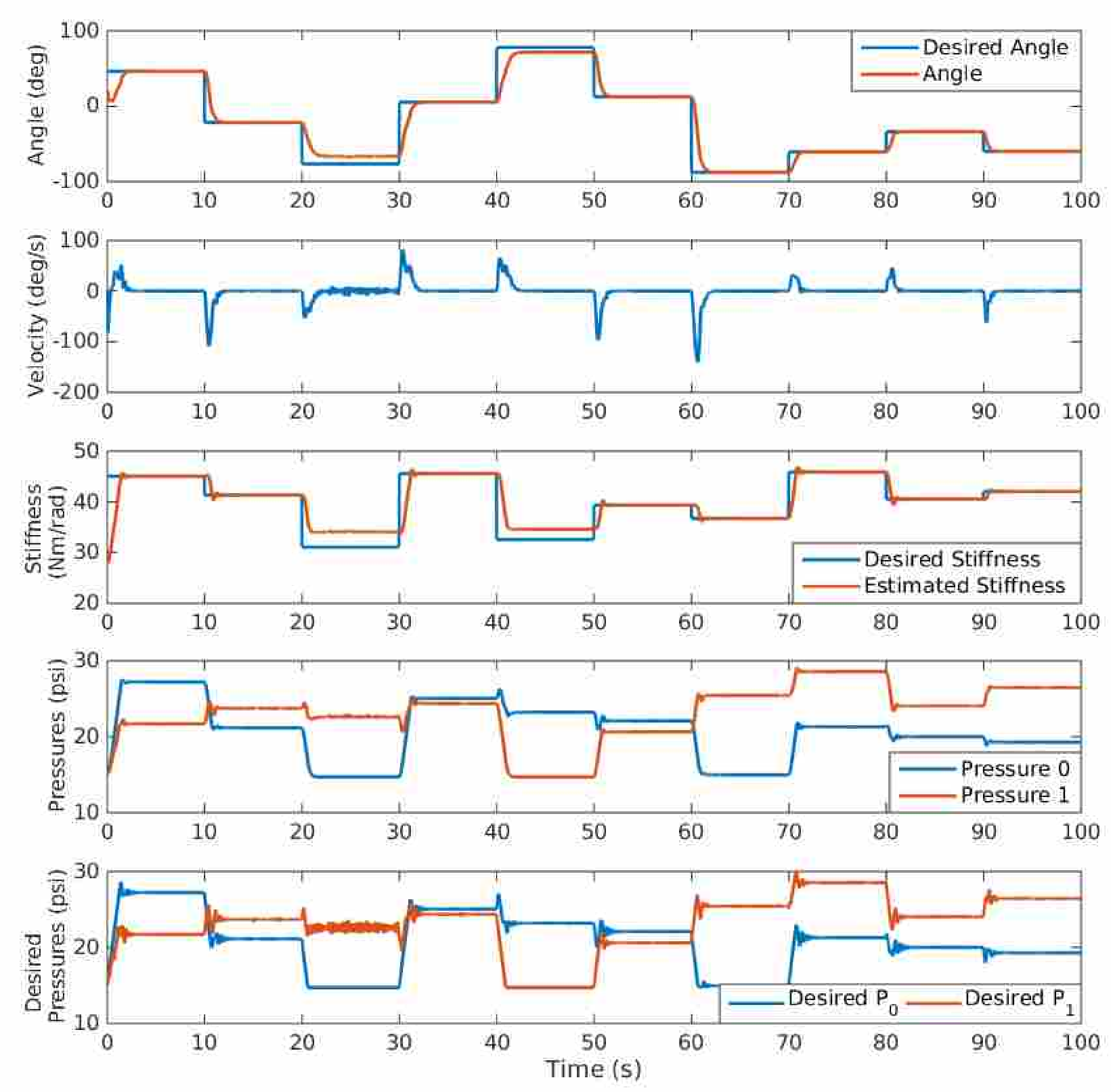

Figure 4.11: MPC simulation of one degree of freedom grub

4.2.4 Grub Implementation

MPC was implemented on the grub using a similar method as the simulation. The control law for implementation is also the minimization of a cost function that is dependent on a desired trajectory, system states, and inputs. The cost function was minimized using a sequential least squares programming algorithm (SLSQP) [59]. The bounds on the inputs were the same as in simulation which were atmospheric pressure 14.7 psi (101.4 kPa) and the source pressure which is regulated to 30 psi (206.8 kPa).

Cost Function

Using the system states **x** and inputs Δ**u** from Equation 4.49, the cost function becomes

$$\begin{aligned}\min_{\Delta\mathbf{u}} J(\mathbf{x}) \quad = \quad & \sum_{t=1}^{T}\Big(||\dot{\theta}[t]||^2_{Q_{vel}} + ||\theta[t]-\theta_{goal}||^2_{Q_{pos}} + ||k[t]-k_{goal}||^2_{Q_{stf}} \\ & + ||\Delta P_{0,des}[t]||^2_{R_0} + ||\Delta P_{1,des}[t]||^2_{R_1}\Big),\end{aligned} \tag{4.62}$$

where θ_{goal} is the desired angle, k_{goal} is the desired stiffness, Q_{vel} is the velocity weighting, Q_{pos} is the position error weighting, Q_{stf} is the stiffness error weighting, R_0 is the weighting on the change in $P_{0,des}$, and R_1 is the weighting on the change in $P_{1,des}$. The cost for large inputs ($H(\Delta\mathbf{u})$) that was include in Equation 4.55 for simulation was not needed for implementation on the grub. This is most likely because of unmodeled dynamics such as friction or the first order pressure model approximation which dampen out oscillation that occur in simulation. The cost function Equation 4.62 is subject to:

$$\Delta\mathbf{x}[t+1] = \mathbf{A}_d\Delta\mathbf{x}[t] + \mathbf{B}_d\Delta\mathbf{u}[t] \quad t = 1..T \tag{4.63}$$

$$\mathbf{x}[t] = \mathbf{x}_{init} + \Delta\mathbf{x}[t] \quad t = 1..T \tag{4.64}$$

$$\theta_{min} \leq \theta \leq \theta_{max} \quad t = 1..T \tag{4.65}$$

$$P_{min} \leq P_i[t] \leq P_{max}, \quad t = 1..T \tag{4.66}$$

where θ_{min} and θ_{max} are joint limits.

Results

The weights used in the cost function are in Table 4.6 and the simulation was run at 30 Hz with a horizon of five steps. The weights used for the grub are similar to the weights used in simulation. The weights for position and stiffness are about half of what they were in simulation but this effectively doubles the weighting on velocity. The weights on the system inputs decreased from .05 to .001 suggesting that the inputs have a stronger effect in the system model than they do on the actual grub.

The same desired angle and stiffness commands that where used in Figure 4.8 for SMC were used with MPC for comparison between SMC and MPC. The results of the implementation on the grub can be seen in Figure 4.12.

Table 4.6: Controller gains for model predictive control on the grub

Q_{pos}	28.75
Q_{stf}	.1
Q_{vel}	2.0
R_0	0.001
R_1	0.001
T	5
rate	30 Hz
θ_{min}	$-120°$
θ_{max}	$120°$

Both angle and stiffness are controlled on the grub using MPC. Just like in simulation there is a region where a certain desired angle and stiffness combination is not achievable. From 60 seconds to 75 seconds the dynamics don't allow both a large angle and large stiffness. The controller seems to be limited by source pressure of 30 psi (206.8 kPa) during this period because P_1 is saturated. The controller could lower $P_{0,des}$ to achieve the desired angle but this would lower the joint stiffness. Changing the weights in the cost function to favor angle or stiffness would give better tracking at the expense of the other. It is interesting to note that as seen in Figure 4.12, the solve time during this period increases.

Another interesting occurrence is the angle response near 30 seconds and 90 seconds just before reaching the desired angle. This occurs when the forward simulation predicts that the joint angle will cross over the desired angle trajectory. As expected, MPC then reduces desired pressures to reduce error accrued by overshoot. Unfortunately, MPC reduces the pressure too much. A higher order pressure model and a torque model that accounts for actuator interactions would better predict this behavior. Small step sizes in the desired angle trajectory do not exhibit this behavior. However, even with a model that does not account for all of the grub dynamics, position and stiffness can be controlled.

4.3 MPC and SMC Discussion

The grub response for both MPC and SMC can be seen in Figure 4.13. Both control methods show similar performance for angle and stiffness tracking and the desired pressure control signals from both controllers follow similar trends. Between MPC and SMC with the current model, MPC is a better method for control of the grub because MPC is not as affected by model discrepancies and MPC solve rates are fast enough for good control.

Angle tracking is a higher priority for the tuning of the SMC so it does better than MPC but worse for stiffness tracking. It was found that SMC did better when either angle or stiffness was weighted higher. When weighted equally, tracking was poor for both states because of considerable oscillations. MPC was much easier to tune for equal tracking priority.

SMC seemed to rely heavily on the integrator state for angle tracking with large steps where hysteresis in the grub dynamics differed from how the model predicted the system would behave. MPC did not have this problem but was limited by saturation pressures.

The main strength of SMC is that nonlinear dynamics are accounted for and control rates can be relatively fast compared to MPC. The control laws developed in this chapter are easy to implement in code and solve quickly. Control performance however depends greatly on the accuracy of the original model. A higher fidelity model would improve SMC performance and could possibly eliminate the need for the integrator or even the split controller.

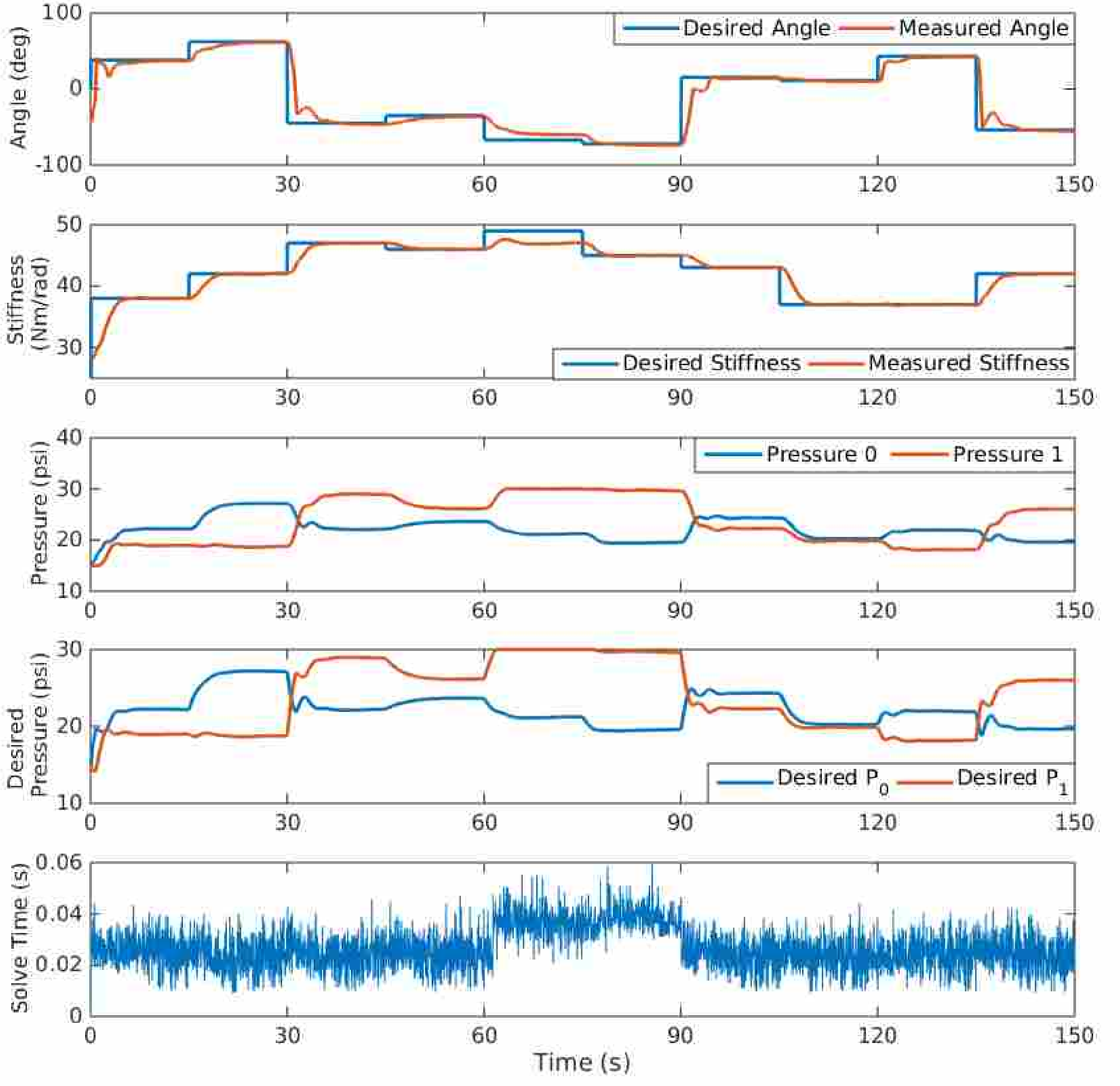

Figure 4.12: MPC response on the grub

While SMC was able to control at a rate of 100 Hz consistently, MPC is limited by how fast the solver can find solutions. The control rate for MPC was set to 30 Hz which is achievable, but in regions where the dynamics are complicated, the solve time increases significantly. This did not seem to affect control in a significant way for this test but shows the main limitation of MPC. A higher fidelity model with more states and state interactions could make MPC infeasible or intractable with the current hardware and solver algorithms because solve times would be too slow for good control performance.

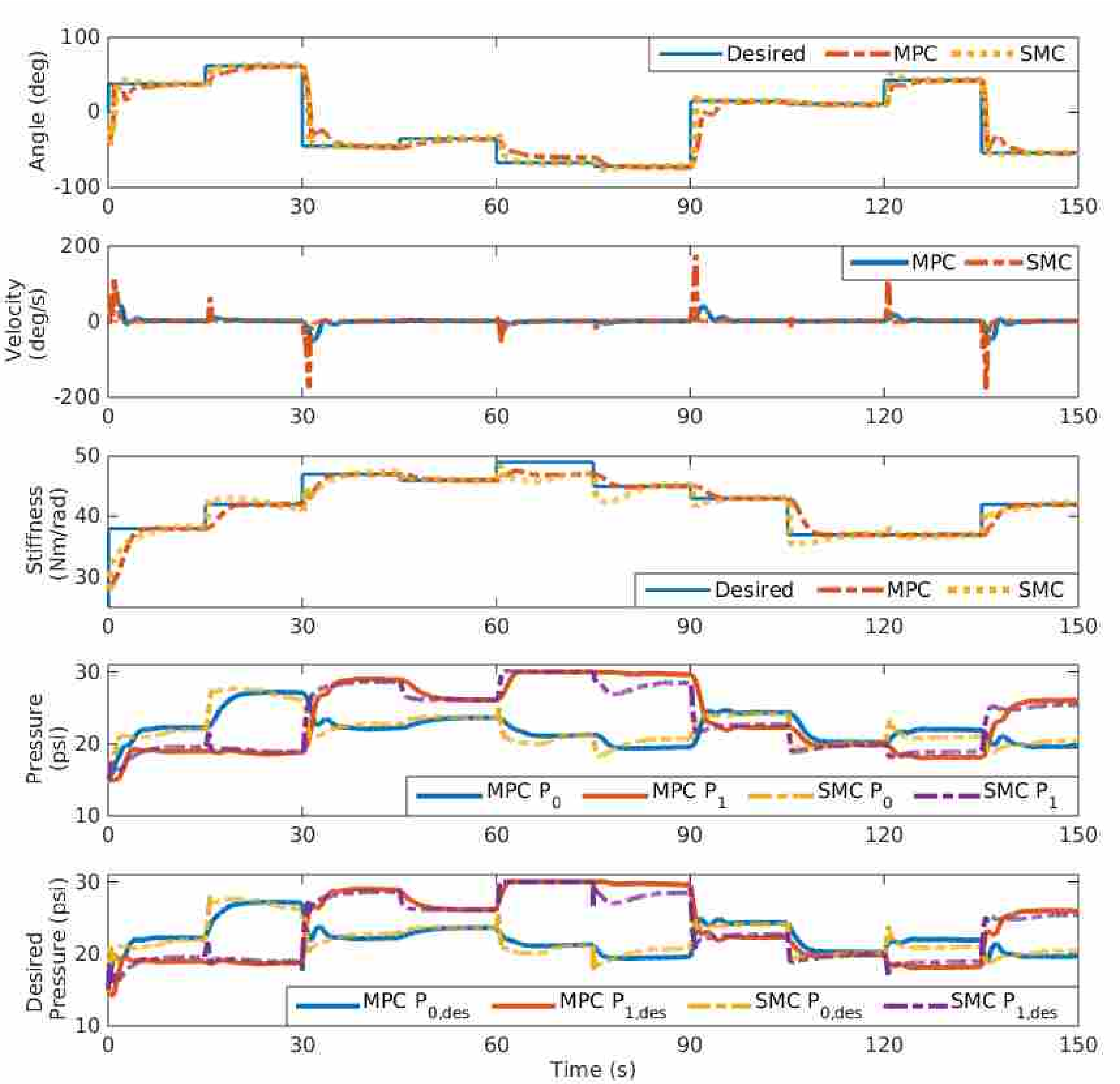

Figure 4.13: MPC and SMC response on the actual grub

4.4 MPC for Multiple Degrees of Freedom

Control for a single degree of freedom shows proof of concept but applications are limited. Extending control to multiple degrees of freedom is necessary to show the value of the ability to control stiffness and position. This section applies stiffness and position control to the right arm of King Louie seen in Figure 4.14, and shows that the end effector stiffness changes when stiffness in the joints is changed. Deflection was reduced by approximately 50% when the arm was commanded at 50 Nm/rad stiffness rather than 35 Nm/rad stiffness.

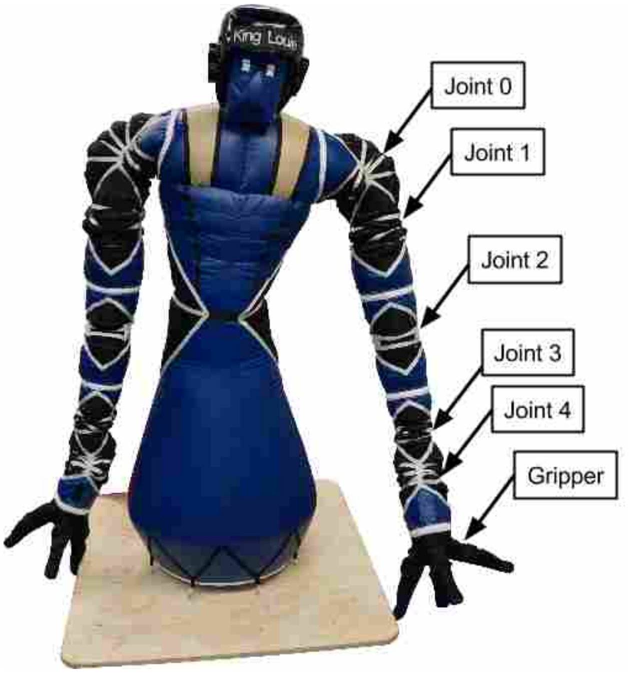

Figure 4.14: Inflatable humanoid robot named King Louie

4.4.1 Implementation

MPC for both position and stiffness was applied using the same controller as the grub on each individual joint on King Louie's right arm. This was done in Section 2.2.5 for two state MPC and Section 2.4 for four state MPC that accounts for torque with good results. The cost function used for each joint was

$$\begin{aligned}\min_{\Delta\mathbf{u}} J(\mathbf{x}) &= \sum_{t=1}^{T}\Big(||\dot{\theta}[t]||^2_{Q_{vel}} + ||\theta[t]-\theta_{goal}||^2_{Q_{pos}} + ||k[t]-k_{goal}||^2_{Q_{stf}} \\ &\quad + ||\Delta P_{0,des}[t]||^2_{R_0} + ||\Delta P_{1,des}[t]||^2_{R_1} + ||\dot{k}[t]||^2_{Q_{dstf}}\Big),\end{aligned} \tag{4.67}$$

where θ_{goal} is the desired angle, k_{goal} is the desired stiffness, Q_{vel} is the velocity weighting, Q_{pos} is the position error weighting, Q_{stf} is the stiffness error weighting, R_0 is the weighting on the change in $P_{0,des}$, R_1 is the weighting on the change in $P_{1,des}$ and $\dot{k}[t]$ was calculated using Equation 3.19. The cost function Equation 4.67 is subject to:

$$\Delta\mathbf{x}[t+1] = \mathbf{A}_d\Delta\mathbf{x}[t] + \mathbf{B}_d\Delta\mathbf{u}[t] \quad t = 1..T \tag{4.68}$$

$$\mathbf{x}[t] = \mathbf{x}_0 + \Delta\mathbf{x}[t] \quad t = 1..T \tag{4.69}$$

$$\begin{aligned}\dot{k}[t] &= c_0 k[t] + a_0\alpha_0[t]c_1P_{0,des}[t] + a_1\alpha_1[t]c_1P_{1,des}[t] \\ &\quad -\dot{\theta}[t]\left(a_0\alpha_0^2[t]P_0[t] + a_1\alpha_1^2[t]P_1[t]\right) \quad t = 1..T\end{aligned} \tag{4.70}$$

$$\theta_{min} \le \theta \le \theta_{max} \quad t = 1..T \tag{4.71}$$

$$P_{min} \le P_i[t] \le P_{max}, \quad t = 1..T \tag{4.72}$$

where θ_{min} and θ_{max} are joint limits. The cost on $\dot{k}[t]$ with the weight Q_{dstf} was added to reduce oscillations about the desired stiffness trajectory.

When applied to each joint on King Louie, the controller weights and the first order pressure model coefficients were tuned manually for each joint for best performance and the results can be seen in Table 4.7. The process for picking weights and parameters for each joint would be improved with system identification.

Table 4.7: MPC weights and parameters used for each joint on King Louie

	Joint 0	Joint 1	Joint 2	Joint 3	Joint 4
Q_{vel}	2.0	1.0	2.0	2.0	2.0
Q_{pos}	100.0	100.0	28.75	28.75	28.75
Q_{stf}	2.0	2.0	1.0	1.0	1.0
Q_{dstf}	0.05	0.05	0.05	0.05	0.05
R_0	0.002	0.002	0.0005	0.0005	0.0005
R_1	0.002	0.002	0.0005	0.0005	0.0005
θ_{max}	$120°$	$120°$	$120°$	$120°$	$120°$
θ_{min}	$-120°$	$-120°$	$-120°$	$-120°$	$-120°$
c_0	-2.65	-2.65	-5.301	-5.301	-5.301
c_1	2.65	2.65	5.301	5.301	5.301

Using the same SLSQP method described in Section 4.2.4 to minimize the cost function, Equation 4.67, MPC was implemented at 30 Hz to track a step command in position for each joint while maintaining a constant stiffness. The results can be seen in Figure 4.15. Joint 2 or the elbow is not included in Figure 4.15 because it was constructed differently than the other joints. The elbow was sewn so that the applied torque is restricted in one direction. The current model does not account for this and position tracking is poor for this joint.

Looking at Figure 4.15, stiffness is tracked well for all four joints for the duration of the test. Position tracking is good for Joints 1, 3, and 4 but Joint 0 shows poor tracking. There is 8° overshoot after the step at 15 seconds and the joint moves 23° in the wrong direction after the step at 30 seconds. The reason for this is most likely because of the independent and decoupled joint controllers. The orientation of Joint 0 with the shoulder makes it so this joint works directly against gravity and because it is the first joint, the inertia effects from the other links are most apparent. The desired pressure signal shows that the controller is applying a pressure combination that should produce a torque that will drive the joint angle to the desired angle. However that torque is not significant enough to overcome the gravity and inertia forces of the other links. Once the other links get close to their desired angles, the inertial forces decrease enough that Joint 0 moves toward the desired angle. The pressure controller shows good tracking of the desired pressure signal from MPC. The solve times for the optimization average just over .02 seconds for all four joints which show that the controller can run effectively at 30 Hz.

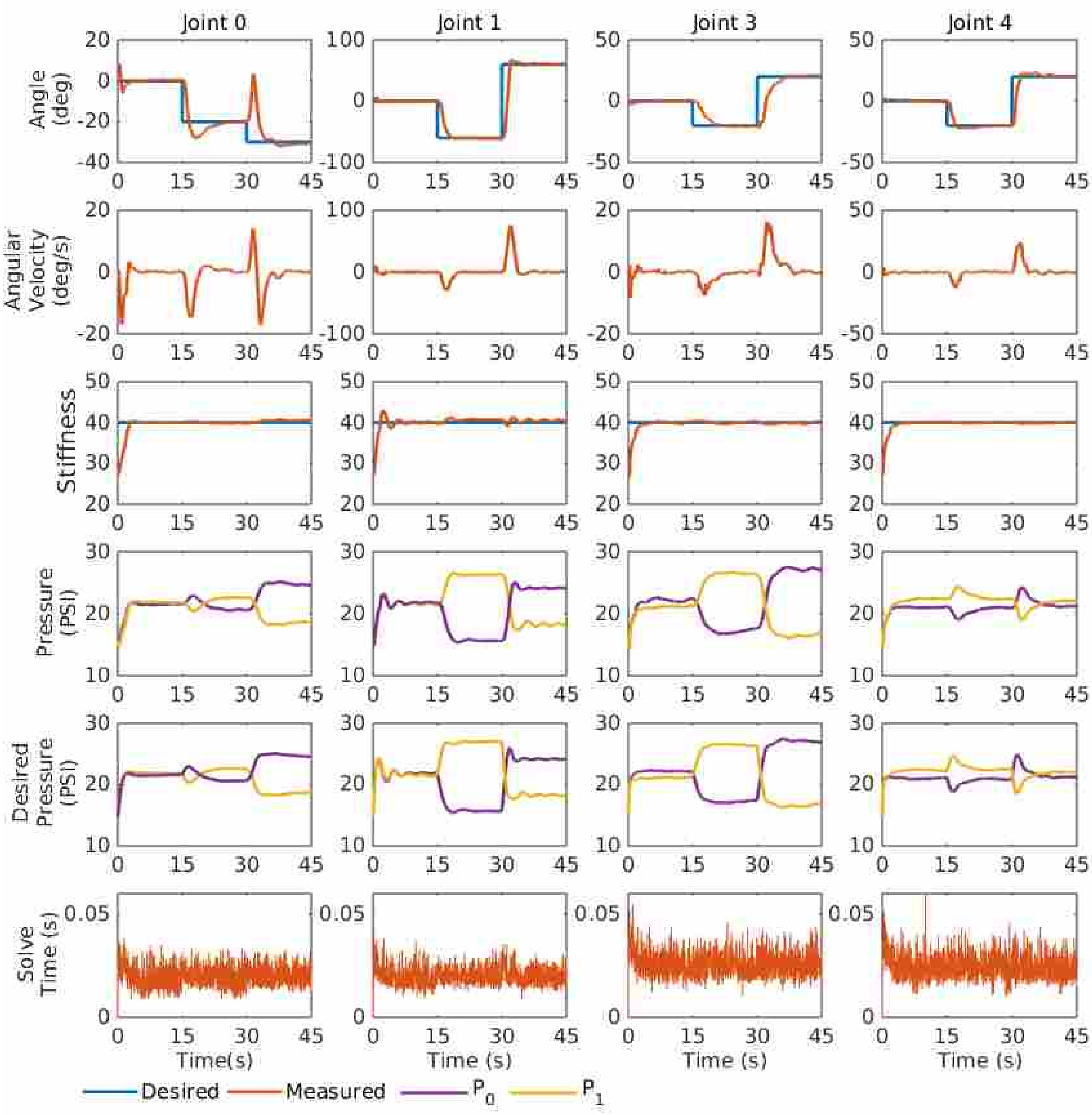

Figure 4.15: Angle step response of King Louie's right arm while maintaining a constant stiffness of 40 Nm/rad

Even though stiffness is controlled to a desired value for each joint, the actual stiffness of the joint is not expected to be the estimated value. Extensive testing was done on the grub as shown in Section 3.3.2 to validate that the controlled stiffness was the actual stiffness of the joint. For good estimation of joint stiffness as seen in Equation 3.13, accurate torque coefficients (a_0 and a_1) and a good angle to volume relationship (M_i and b) must be determined. Because of the construction of King Louie's arm, these parameters are difficult to determine for each joint. Instead, the grub parameters are used and tests were performed to show that overall arm stiffness can be increased or decreased. Future work where actuator designs allow for these coefficients to be determined analytically from geometry would greatly simplify this problem.

4.4.2 Stiffness Test

The stiffness of King Louie's arm was tested by commanding a joint angle for each joint and a single stiffness to be applied to all the joints. Once all the joints achieve the stiffness and joint angles, a 4 lb weight was applied to the end effector. Figure 4.16 shows the arm in the initial position and Figure 4.17 shows the arm at maximum deflection for a commanded stiffness of 35 Nm/rad. The desired joint angles for this configuration are $\theta_0 = -30^\circ$, $\theta_1 = 60^\circ$, $\theta_2 = 0^\circ$, $\theta_3 = 20^\circ$, and $\theta_4 = 20^\circ$.

The maximum deflection is measured by first recording the XYZ coordinates of the end effector before the load is applied and then tracking the end effector coordinates using motion capture data during the joint controller response to the load. The deflection vector can then be calculated by subtracting the initial coordinate vector from the end effector coordinates while under load. The deflection magnitude for the test is then the magnitude of the deflection vector with the biggest euclidean norm. This test was completed for stiffnesses from 35 Nm/rad to 50 Nm/rad in increments of 1 Nm/rad. For each tested stiffness, three trials were completed and the results can be seen in Figures 4.18 and 4.19.

The data shown in Figure 4.18 shows a downward trend in deflection as stiffness is increased. This means that controlling joint stiffness affects end effector stiffness in a predictable way. Figure 4.19 shows the settling error for each joint after the load is applied. Because MPC determines the control input based on weightings between stiffness error and position error, at some point there is a balancing effect. In order to achieve high stiffnesses, the joint angle is limited

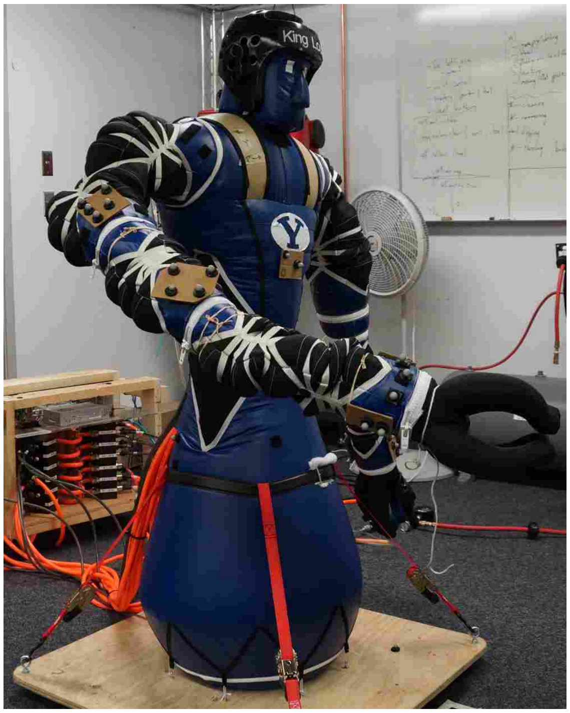

Figure 4.16: King Louie with arm up

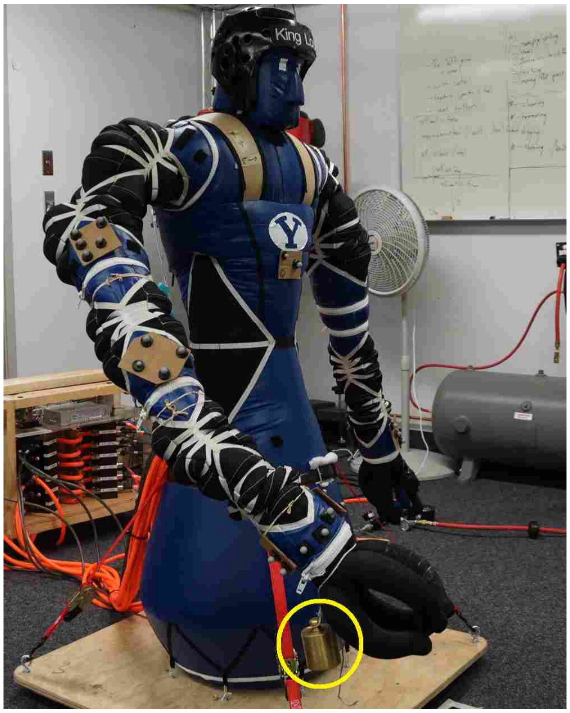

Figure 4.17: Deflection from 4 lb load (weight is circled)

and there is high steady state error. That is why position error increases for all joints for stiffnesses between 47 Nm/rad and 50 Nm/rad. Joint 3 shows significant error for these tests for all but around 40 Nm/rad. The actuators in the wrist are significantly smaller than the actuators in the grub or in King Louie's shoulder. The smaller size suggests that not enough torque can be applied to support the final links at low and high stiffnesses. These figures show that deflection and steady state error at this configuration can be minimized if the joints are maintained at about 45 Nm/rad.

Controlling at a single constant stiffness defeats the purpose of using variable stiffness actuators. A higher fidelity model that accounts for the link and joint interactions could be used to determine what stiffnesses are best for specific objectives. The model developed in this work is limited to a single degree of freedom but future work that expands the model to multiple degrees of freedom would be valuable for this type of system and could greatly improve the performance of the controller for an arm like King Louie's. The model would have to account for cross coupling in the joints and gravity effects but this is standard for any robotic arm with more than one degree of freedom.

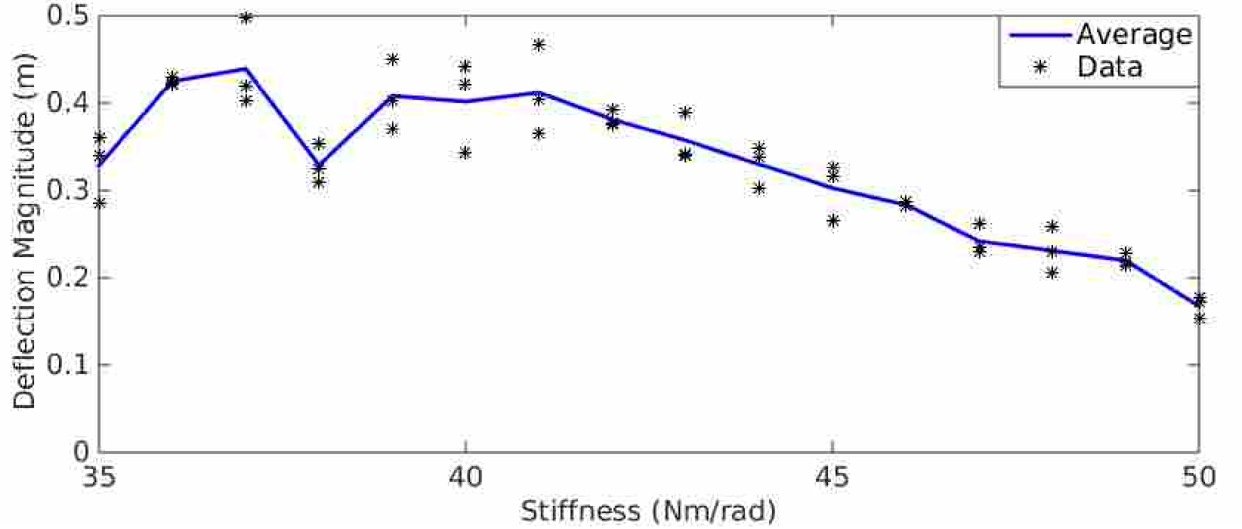

Figure 4.18: Deflection of King Louie's right arm from a 4 lb weight at different stiffnesses

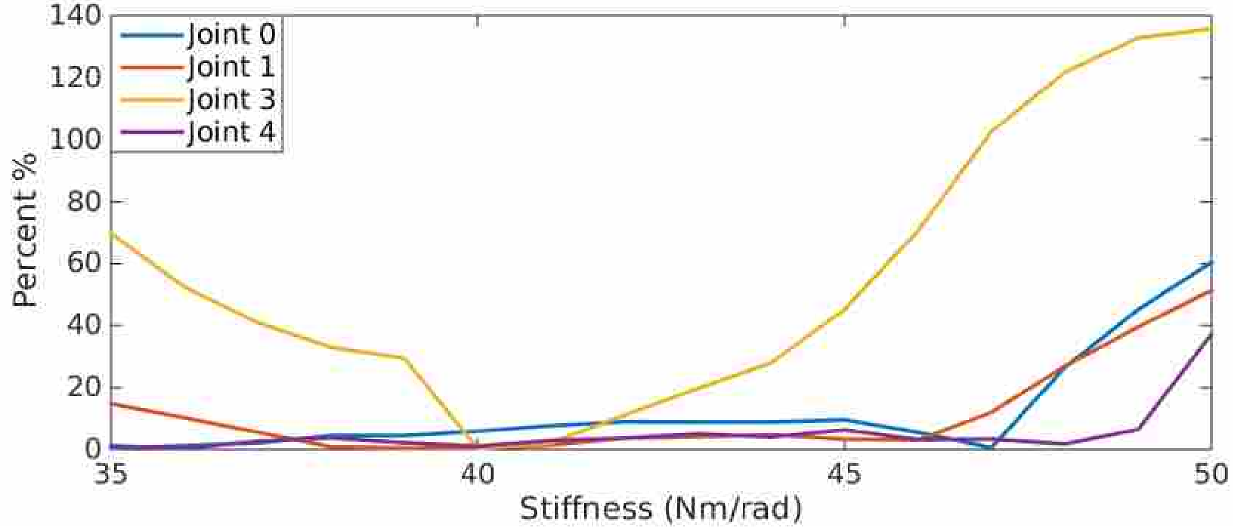

Figure 4.19: Settling error relative to commanded position after load

CHAPTER 5. CONCLUSION

This thesis gives different methods for simultaneous stiffness and position control of an inflatable robot with pneumatic actuation. The reason why the control of a robot of this type is of interest is because it is naturally lightweight and the antagonistic nature of the actuator with air as the working fluid gives the joint natural compliance. Compliance and low inertia are desirable characteristics for robots that operate around humans or in human environments. Robots of this type have not been used in the past because of the difficulty in modeling of the system and because the sensing and actuation are very different from standard techniques. However, this thesis shows that the system is controllable and has value and potential for future work.

5.1 Future Work

All dynamic modeling and control for this thesis was done for a single degree of freedom joint. This was sufficient for the control of multiple degrees of freedom but performance can be significantly improved if the dynamic model and controller took into account cross coupling in the joints and gravity affects. Also a model that accounts for deflections in the links from the inflatable structure could also improve performance.

Control could also be improved if the dynamic model also accounted for more of the gas dynamics. The gas dynamics are discontinuous in terms of choked and unchoked flow, and the behavior of the valve. This would require a controller that could account for the dramatic shift in dynamics as the system shifts from one mode to another. In this thesis, all of the gas dynamics are modeled as a first order pressure model. Simply increasing the order of the model and including cross coupling terms between the pressures in each actuator could improve performance rather than a direct model of the working fluid.

A higher fidelity model would significantly improve the performance of SMC at the same time reducing the effectiveness of MPC if the current solve rates for optimization stayed the same.

SMC implementation is not as computationally taxing as MPC which would be limited by solve rates as the model becomes more complex. Work can be done to improve the efficiency of the MPC solver for higher fidelity models. A higher fidelity model will only increase the complexity of the controller so identifying an efficient SMC structure will be important.

For this thesis, motion capture was necessary for joint angle estimation of multiple degrees of freedom. One motivation for this work is to make robots better suited for human environments and relying on motion capture does not help with this goal. Future work to develop sensing techniques that will free the robot from its dependency on motion capture will help make this platform usable in human environments.

5.2 Contributions

This thesis first showed that the position of an inflatable, pneumatically actuated joint can be controlled using a two state model in Chapter 2. LQR and MPC methods were used to control within 3° of a desired position for a single degree of freedom in Section 2.2. Control was then implemented on multiple degrees of freedom to repeatably accomplish a pick and place task in Section 2.2.5. As explained in Sections 2.3 and 2.4, the authors of [1] and [3] showed that the two state model could be extended to four states to include actuator pressures for significant improvement in control performance. This chapter concludes that inflatable robots can have accurate and repeatable position control.

Then in Chapter 3, a state space dynamic model was developed that can be used to predict future position, stiffness, and pressure states of an antagonistic, pneumatically actuated joint. The model was fit to data collected from the grub such as the angle to volume relationship, damping and stiffness terms, and a first order pressure model. Figure 3.5 shows how the model compares to actual data collected from the grub. The stiffness state was validated by comparing the estimated joint stiffness with the measured joint stiffness in Section 3.3.2. This chapter concludes that system states for angular velocity, angle, stiffness, and pressures can be used to model the dynamics of the system.

Lastly in Chapter 4, the state space dynamic model was used to control stiffness and position using both MPC and SMC methods. The SMC implementation on the grub was described in Section 4.1 and it was able to control the joint angle to within 2° of the desired angle at steady state and within 3 Nm/rad of the desired stiffness. MPC was implemented on the grub which is described in Section 4.2, and was able to control the joint angle to within 2° of the desired angle at steady state and within 1 Nm/rad of the desired stiffness. Based on the results of the two controllers, it was concluded that for the current dynamic model that MPC is the preferred control method as discussed in Section 4.3. MPC was then implemented on King Louie in Section 4.4, and it was shown that by controlling to higher stiffnesses, the deflection of the end effector decreases. Specifically, the deflection decreased by approximately 50% when controlled at the maximum stiffness around 50 Nm/rad than at 35 Nm/rad. This chapter concludes that stiffness can be controlled and that its effects are predictable and significant.

REFERENCES

[1] Gillespie, M., Best, C., and Killpack, M., 2016. "Simultaneous position and stiffness control for an inflatable soft robot." In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. vii, 2, 9, 11, 29, 32, 89

[2] Best, C. M., Wilson, J. P., and Killpack, M. D., 2015. "Control of a pneumatically actuated, fully inflatable, fabric-based, humanoid robot." In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, IEEE, pp. 1133–1140. 2, 9, 13, 29

[3] Best, C., Gillespie, M., Hyatt, P., Rupert, L., Sherrod, V., and Killpack, M. "Model predictive control for pneumatically actuated soft robots." *IEEE Robotics & Automation Magazine*. 2, 9, 31, 32, 35, 41, 89, 96

[4] Hogan, N., 1985. "Impedance control: An approach to manipulation: Part 2: Implementation." *Journal of dynamic systems, measurement, and control,* **107**(1), pp. 8–16. 3

[5] Loughlin, C., Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., and Hirzinger, G., 2007. "The DLR lightweight robot: design and control concepts for robots in human environments." *Industrial Robot: an international journal,* **34**(5), pp. 376–385. 3

[6] Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G., et al., 2010. "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing." In *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*, VDE, pp. 1–8. 3

[7] Albu-Schäffer, A., Ott, C., and Hirzinger, G., 2007. "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots." *The International Journal of Robotics Research,* **26**(1), pp. 23–39. 3

[8] Pratt, G., and Williamson, M., 1995. "Series elastic actuators." In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, Vol. 1, pp. 399–406 vol.1. 3

[9] Ham, R., Sugar, T., Vanderborght, B., Hollander, K., and Lefeber, D., 2009. "Compliant actuator designs." *Robotics Automation Magazine, IEEE,* **16**(3), September, pp. 81–94. 3

[10] Laurin-Kovitz, K. F., Colgate, J. E., and Carnes, S. D., 1991. "Design of components for programmable passive impedance." In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, pp. 1476–1481. 3

[11] Vanderborght, B., Albu-Schäffer, A., Bicchi, A., Burdet, E., Caldwell, D. G., Carloni, R., Catalano, M., Eiberger, O., Friedl, W., Ganesh, G., et al., 2013. “Variable impedance actuators: A review.” *Robotics and Autonomous Systems,* **61**(12), pp. 1601–1614. 3

[12] Garabini, M., Passaglia, A., Belo, F., Salaris, P., and Bicchi, A., 2011. “Optimality principles in variable stiffness control: The vsa hammer.” In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3770–3775. 3

[13] Bicchi, A., and Tonietti, G., 2004. “Fast and ”soft-arm” tactics [robot arm design].” *Robotics Automation Magazine, IEEE,* **11**(2), June, pp. 22–33. 3, 7

[14] Carneiro, J. F., and de Almeida, F. G., 2006. “Reduced-order thermodynamic models for servo-pneumatic actuator chambers.” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* **220**(4), pp. 301–314. 6, 37

[15] Shen, X., and Goldfarb, M., 2007. “Simultaneous force and stiffness control of a pneumatic actuator.” *Journal of Dynamic Systems, Measurement, and Control,* **129**(4), pp. 425–434. 6, 58, 99

[16] Toedtheide, A., Lilge, T., and Haddadin, S., 2016. “Antagonistic impedance control for pneumatically actuated robot joints.” In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. 6

[17] Mihajlov, M., Hübner, M., Ivlev, O., and Gräser, A., 2006. “Modeling and control of fluidic robotic joints with natural compliance.” In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, IEEE, pp. 2498–2503. 6

[18] Ivlev, O., 2009. “Soft fluidic actuators of rotary type for safe physical human-machine interaction.” *2009 IEEE International Conference on Rehabilitation Robotics, ICORR 2009,* **28359**, pp. 1–5. 6

[19] Taghia, J., Wilkening, A., and Ivlev, O., 2012. “Position control of soft-robots with rotary-type pneumatic actuators.” In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, VDE, pp. 1–6. 6

[20] Bicchi, A., Rizzini, S. L., and Tonietti, G., 2001. “Compliant design for intrinsic safety: General issues and preliminary design.” In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, Vol. 4, IEEE, pp. 1864–1869. 6

[21] Tonietti, G., and Bicchi, A., 2002. “Adaptive simultaneous position and stiffness control for a soft robot arm.” In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, Vol. 2, IEEE, pp. 1992–1997. 6

[22] Rus, D., and Tolley, M. T., 2015. “Design, fabrication and control of soft robots.” *Nature,* **521**(7553), pp. 467–475. 7

[23] Jack, A. J., 2001. “Inflatable robotics for planetary applications.” In *6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space, I-SAIRAS, Montreal, Canada*. 7

[24] Gaiser, I., Wiegand, R., Ivlev, O., Andres, a., Breitwieser, H., Schulz, S., and Bretthauer, G., 2014. "Compliant Robotics and Automation with Flexible Fluidic Actuators and Inflatable Structures." *Smart Actuation and Sensing SystemsRecent Advances and Future Challenges*, pp. 567–608. 7

[25] Sanan, S., Moidel, J., and Atkeson, C., 2009. "Robots with inflatable links." In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 4331–4336. 7

[26] Sanan, S., Ornstein, M. H., and Atkeson, C. G., 2011. "Physical human interaction for an inflatable manipulator." In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, IEEE, pp. 7401–7404. 7

[27] Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D., 2014. "Design and control of a soft and continuously deformable 2d robotic manipulation system." In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, pp. 2189–2196. 7

[28] Marchese, A. D., Katzschmann, R. K., and Rus, D., 2014. "Whole arm planning for a soft and highly compliant 2d robotic manipulator." In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, pp. 554–560. 7

[29] Marchese, A. D., Tedrake, R., and Rus, D., 2015. "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator." In *2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, New York*. 7

[30] Sanan, S., Lynn, P. S., and Griffith, S. T., 2014. "Pneumatic torsional actuators for inflatable robots." *Journal of Mechanisms and Robotics,* **6**(3), p. 031003. 7, 10

[31] Qin, S. J., and Badgwell, T. A., 2003. "A survey of industrial model predictive control technology." *Control Engineering Practice,* **11**(7), July, pp. 733–764. 8

[32] Jain, A., Killpack, M. D., Edsinger, A., and Kemp, C. C., 2013. "Reaching in clutter with whole-arm tactile sensing." *The International Journal of Robotics Research*. 8

[33] Killpack, M. D., and Kemp, C. C., 2013. "Fast reaching in clutter while regulating forces using model predictive control." In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, IEEE. 8

[34] Killpack, M. D., Kapusta, A., and Kemp, C. C., 2015. "Model predictive control for fast reaching in clutter." *Autonomous Robots*, pp. 1–24. 8

[35] Erez, T., Tassa, Y., and Todorov, E., 2012. "Infinite-horizon model predictive control for periodic tasks with contacts." *Robotics: Science and Systems VII*, p. 73. 8

[36] Rupert, L., Hyatt, P., and Killpack, M. D., 2015. "Comparing model predictive control and input shaping for improved response of low-impedance robots." In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, IEEE, pp. 256–263. 8

[37] Shim, D. H., Kim, H. J., and Sastry, S., 2003. "Decentralized nonlinear model predictive control of multiple flying robots." In *Proceedings. 42nd IEEE Conference on Decision and Control*, Vol. 4, IEEE, pp. 3621–3626. 8

[38] Leung, C., Huang, S., Kwok, N., and Dissanayake, G., 2006. “Planning under uncertainty using model predictive control for information gathering.” *Robotics and Autonomous Systems,* **54**(11), July, pp. 898–910. 8

[39] Annamalai, A. S. K., Sutton, R., Yang, C., Culverhouse, P., and Sharma, S., 2014. “Robust adaptive control of an uninhabited surface vehicle.” *Journal of Intelligent & Robotic Systems*, pp. 1–20. 8

[40] Wang, Y., and Boyd, S., 2010. “Fast model predictive control using online optimization.” *IEEE Transactions on Control Systems Technology,* **18**(2), March, pp. 267–278. 8

[41] Domahidi, A., Zgraggen, A. U., Zeilinger, M. N., Morari, M., and Jones, C. N., 2012. “Efficient interior point methods for multistage problems arising in receding horizon control.” In *Proceedings of the 51st IEEE Conference on Decision and Control*, no. EPFL-CONF-181938. 8

[42] Dimitrov, D., Wieber, P.-B., Ferreau, H. J., and Diehl, M., 2008. “On the implementation of model predictive control for on-line walking pattern generation.” In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, pp. 2685–2690. 8

[43] Mason, S., Righetti, L., and Schaal, S., 2014. “Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting.” In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, IEEE, pp. 374–379. 8

[44] Stephens, B., 2007. “Integral control of humanoid balance.” In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, IEEE, pp. 4020–4027. 8

[45] Hong, S., Oh, Y., Chang, Y.-H., and You, B.-J., 2008. “Walking pattern generation for humanoid robots with lqr and feedforward control method.” In *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, IEEE, pp. 1698–1703. 8

[46] Hedengren, J. D., Shishavan, R. A., Powell, K. M., and Edgar, T. F., 2014. “Nonlinear modeling, estimation and predictive control in apmonitor.” *Computers & Chemical Engineering,* **70**, pp. 133–148. 8, 74

[47] Zhihong, M., Paplinski, A. P., and Wu, H. R., 1994. “A robust mimo terminal sliding mode control scheme for rigid robotic manipulators.” *Automatic Control, IEEE Transactions on,* **39**(12), pp. 2464–2469. 8

[48] Young, K. D., Utkin, V. I., and Ozguner, U., 1999. “A control engineer’s guide to sliding mode control.” *IEEE transactions on control systems technology,* **7**(3), pp. 328–342. 8

[49] Utkin, V., Guldner, J., and Shi, J., 2009. *Sliding mode control in electro-mechanical systems.*, Vol. 34 CRC press. 8

[50] Khalil, H. K., 2014. *Nonlinear control.* Pearson Higher Ed. 8, 50, 51, 52, 53

[51] Slotine, J.-J. E., Li, W., et al., 1991. *Applied nonlinear control.*, Vol. 199 Prentice-Hall Englewood Cliffs, NJ. 8, 51, 58

[52] Ogata, K., 1995. *Discrete-time control systems.*, Vol. 2 Prentice Hall Englewood Cliffs, NJ. 20, 22

[53] Mattingley, J., and Boyd, S., 2012. “Cvxgen: a code generator for embedded convex optimization.” *Optimization and Engineering,* **13**(1), pp. 1–27. 22

[54] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., 2009. *Robotics: modelling, planning and control.* Springer Science & Business Media. 26

[55] Shampine, L. F., and Reichelt, M. W., 1997. “The matlab ODE suite.” *SIAM journal on scientific computing,* **18**(1), pp. 1–22. 59, 72

[56] Byrd, R. H., Hribar, M. E., and Nocedal, J., 1999. “An interior point algorithm for large-scale nonlinear programming.” *SIAM Journal on Optimization,* **9**(4), pp. 877–900. 72

[57] Waltz, R. A., Morales, J. L., Nocedal, J., and Orban, D., 2006. “An interior algorithm for nonlinear optimization that combines line search and trust region steps.” *Mathematical programming,* **107**(3), pp. 391–408. 72

[58] Byrd, R. H., Gilbert, J. C., and Nocedal, J., 2000. “A trust region method based on interior point techniques for nonlinear programming.” *Mathematical Programming,* **89**(1), pp. 149–185. 72

[59] Kraft, D., et al., 1988. *A software package for sequential quadratic programming.* DFVLR Obersfaffeuhofen, Germany. 76

APPENDIX A. STIFFNESS MODEL WITH GAS DYNAMICS

In this appendix, the torque model that was developed in [3] was used to develop a model for joint stiffness which accounts for the gas dynamics in each actuator of a rotary pneumatic actuator. This model uses the gas dynamics rather than the first order pressure model for the low level PID controller as was done in Chapter 3. The input variables for the gas dynamics model are the mass flows in each of the actuators or $\dot{m}_0$ and $\dot{m}_1$. These inputs depend on the effective valve area for the real system. The gas dynamics model was used with model predictive control (MPC) to track both a desired position and stiffness signal in simulation. A relationship between effective valve area and a commanded valve current was developed so that MPC could be applied to a single degree of freedom on the grub.

There are several reasons why the first order pressure model was used more extensively than the gas dynamics model. The main reason was the control signal produced by the gas dynamics model was erratic which did not give good performance. The first order pressure model had a smooth control signal that was capable of good tracking of both the desired angle and stiffness signals. The gas dynamic model is much more complex when implemented on the grub than the first order pressure model as will be discussed in Section A.3. The results of the exploratory work in this appendix show a viable option for position and stiffness control using the gas dynamics.

A.1 Model Derivation

The gas dynamics model uses the same dynamic equations for the position states as were used in the system model developed in Chapter 3. The derivation for the stiffness dynamic equation starts using the torque model developed in [3]. The output stiffness for a rotary joint with compliant actuation can be defined as

$$k = -\frac{\partial \tau}{\partial \theta}, \tag{A.1}$$

where

$$\tau = a_0 P_0 + a_1 P_1. \tag{A.2}$$

Using a negative in the equation keeps the stiffness term k positive and is the convention for this work. The equation for stiffness when the torque model is substituted then becomes

$$k = -\frac{\partial}{\partial \theta}\left(a_0 P_0 + a_1 P_1\right). \tag{A.3}$$

Then taking the partial derivative with respect to θ and using the ideal gas law and the angle to volume relationship $V_i = M_i\theta + b$ developed in Chapter 3, stiffness becomes

$$k = \frac{a_0 M_0 m_0 RT}{(M_0\theta + b)^2} + \frac{a_1 M_1 m_1 RT}{(M_1\theta + b)^2}. \tag{A.4}$$

Taking the derivative with respect to time then gives

$$\dot{k} = \frac{a_0 M_0 RT}{(M_0\theta + b)^2}\dot{m}_0 + \frac{a_1 M_1 RT}{(M_1\theta + b)^2}\dot{m}_1 - 2\dot{\theta}\left(\frac{a_0 M_0^2 m_0 RT}{(M_0\theta + b)^3} + \frac{a_1 M_1^2 m_1 RT}{(M_1\theta + b)^3}\right), \tag{A.5}$$

and substituting in the ideal gas law gives

$$\dot{k} = \frac{a_0 M_0 RT}{(M_0\theta + b)^2}\dot{m}_0 + \frac{a_1 M_1 RT}{(M_1\theta + b)^2}\dot{m}_1 - 2\dot{\theta}\left(\frac{a_0 M_0^2 P_0}{(M_0\theta + b)^2} + \frac{a_1 M_1^2 P_1}{(M_1\theta + b)^2}\right), \tag{A.6}$$

which defines the stiffness dynamics.

The derivative with respect to time of the angle to volume relationship is

$$\dot{V}_i = M_i\dot{\theta}. \qquad i = 0,1 \tag{A.7}$$

Now taking the derivative of the ideal gas law with respect to time gives

$$\dot{P}_i = \frac{RT}{V_i}\dot{m}_i - \frac{P_i}{V_i}\dot{V}_i. \qquad i = 0,1 \tag{A.8}$$

Substituting the angle to volume relationship then gives

$$\dot{P}_i = \frac{RT}{M_i\theta + b}\dot{m}_i - \frac{P_i}{M_i\theta + b}M_i\dot{\theta}. \qquad i = 0,1 \tag{A.9}$$

Now the system of equations used to describe the dynamics are the following:

$$\begin{aligned}
\ddot{\theta} &= \frac{k_d}{I}\dot{\theta} + \frac{k_s}{I}\theta + \frac{a_0}{I}P_0 + \frac{a_1}{I}P_1 && \text{(A.10)}\\
\dot{\theta} &= \dot{\theta} && \text{(A.11)}\\
\dot{k} &= \frac{a_0M_0RT}{(M_0\theta+b)^2}\dot{m}_0 + \frac{a_1M_1RT}{(M_1\theta+b)^2}\dot{m}_1 - 2\dot{\theta}\left(\frac{a_0M_0^2P_0}{(M_0\theta+b)^2} + \frac{a_1M_1^2P_1}{(M_1\theta+b)^2}\right) && \text{(A.12)}\\
\dot{P}_0 &= \frac{RT}{M_0\theta+b}\dot{m}_0 - \frac{P_0}{M_0\theta+b}M_0\dot{\theta} && \text{(A.13)}\\
\dot{P}_1 &= \frac{RT}{M_1\theta+b}\dot{m}_1 - \frac{P_1}{M_1\theta+b}M_1\dot{\theta} && \text{(A.14)}
\end{aligned}$$

with states $\mathbf{x} = [\dot{\theta}, \theta, k, P_0, P_1]$ and inputs $\mathbf{u} = [\dot{m}_0, \dot{m}_1]$.

A.2 Model Simulation and Control

The system of equations was used to simulate the system. Forward simulation was done using MATLAB's ODE45 function and the system inputs were determined by using a MPC algorithm to track a series of step inputs in desired stiffness and position for a single degree of freedom joint. The results can be seen in Figure A.1. The simulation results show that the system model can be used to track both a desired position and stiffness signal. Because this was preliminary work, the controller was not tuned for good performance but the results are shown for proof of concept. The oscillations around the desired angle can be reduced and the poor stiffness tracking can be improved by tuning the MPC weights and by customizing the cost function.

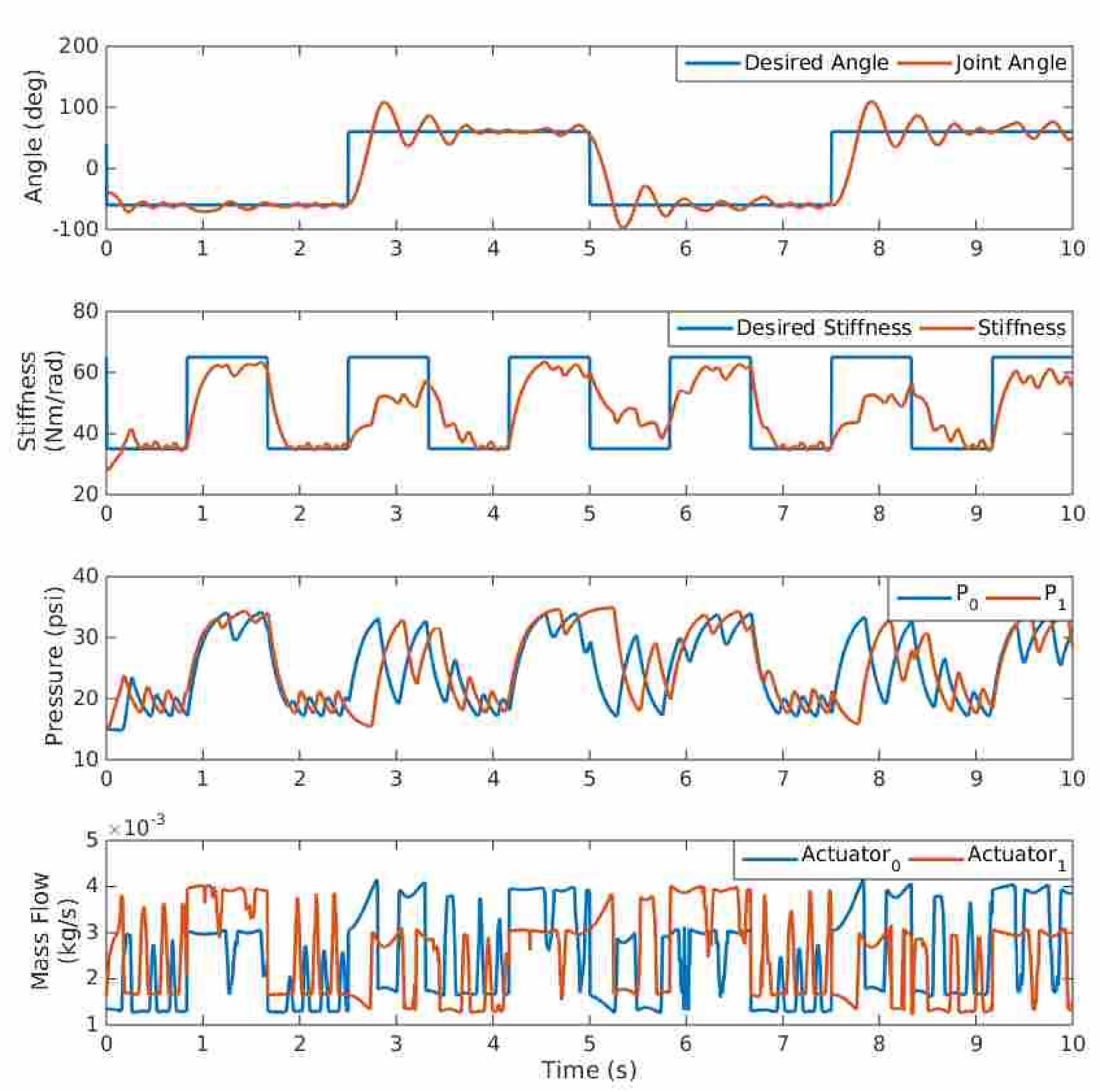

Figure A.1: Step response using MPC in simulation

A.3 Grub Implementation

The inputs to the system for the gas dynamics model developed are mass flows into each of the actuator bladders ($\dot{m}_0$ and $\dot{m}_1$). The challenge for implementation on the grub is to then find a way to map flow rates to a desired electrical current in mA used to operate the valves. The mass flow into each chamber using bidirectional spool valves was described in [15] to be a piecewise continuous function which was

$$\dot{m}(P_u,P_d,A_v,C_f) = \begin{cases} A_vC_f\sqrt{\dfrac{\gamma}{RT}\left(\dfrac{2}{\gamma+1}\right)^{(\gamma+1)/(\gamma-1)}}P_u \\ \text{if } \dfrac{P_d}{P_u} \leq C_r \text{ (choked)} \\ A_vC_f\sqrt{\dfrac{2\gamma}{RT(\gamma-1)}}\sqrt{1-\left(\dfrac{P_d}{P_u}\right)^{(\gamma-1)/\gamma}}\left(\dfrac{P_d}{P_u}\right)^{(1/\gamma)}P_u \\ \text{otherwise (unchoked)} \end{cases}, \tag{A.15}$$

where A_v is the valve area command or effective area, P_u and P_d are upstream and downstream pressures, γ is the ratio of specific heats ($\gamma = 1.4$), R comes from the universal gas constant for air, T is the air temperature which is assumed to be constant, C_f is the discharge coefficient, and C_r determines sonic and subsonic flow ($C_r = .528$).

All variables in the model for mass flow through the valve can be measured or predetermined as properties of air except the effective area and discharge coefficient. The discharge coefficient is unitless and is measured for the valve. The model described by Equation A.15 assumes an ideal nozzle, so the discharge coefficient is a fudge factor to scale the true effective area of the valve to an equivalent effective area for an ideal nozzle. Even though there are ways to determine the discharge coefficient separately from the effective area, they will be combined into a single term designated for the rest of this appendix as simply the effective area. Determining how the effective area changes with the valve electrical current commands was determined experimentally.

For the experiment, a rigid chamber was used with a known volume because this allows the mass of the air to be known based off of ideal gas and constant temperature assumptions. The valve was then used to vary the pressure in the chamber. The data collected includes the tank pressure and actual valve commands in mA. The mass of the air in the tank could then be calculated from the ideal gas law as a function of pressure. Numerically differentiating the air mass measurement gives an estimate for the mass flow into the rigid chamber. Knowing the mass flow means the only unknown variable in Equation A.15 is (A_vC_f) or the effective area. Calculating the effective area from Equation A.15 and plotting it with the valve command at each time step gives Figure A.2.

The trend of the data in Figure A.2 is that of a sigmoid function with a dead band where the valve is effectively closed. A sigmoid fit to the data takes the form

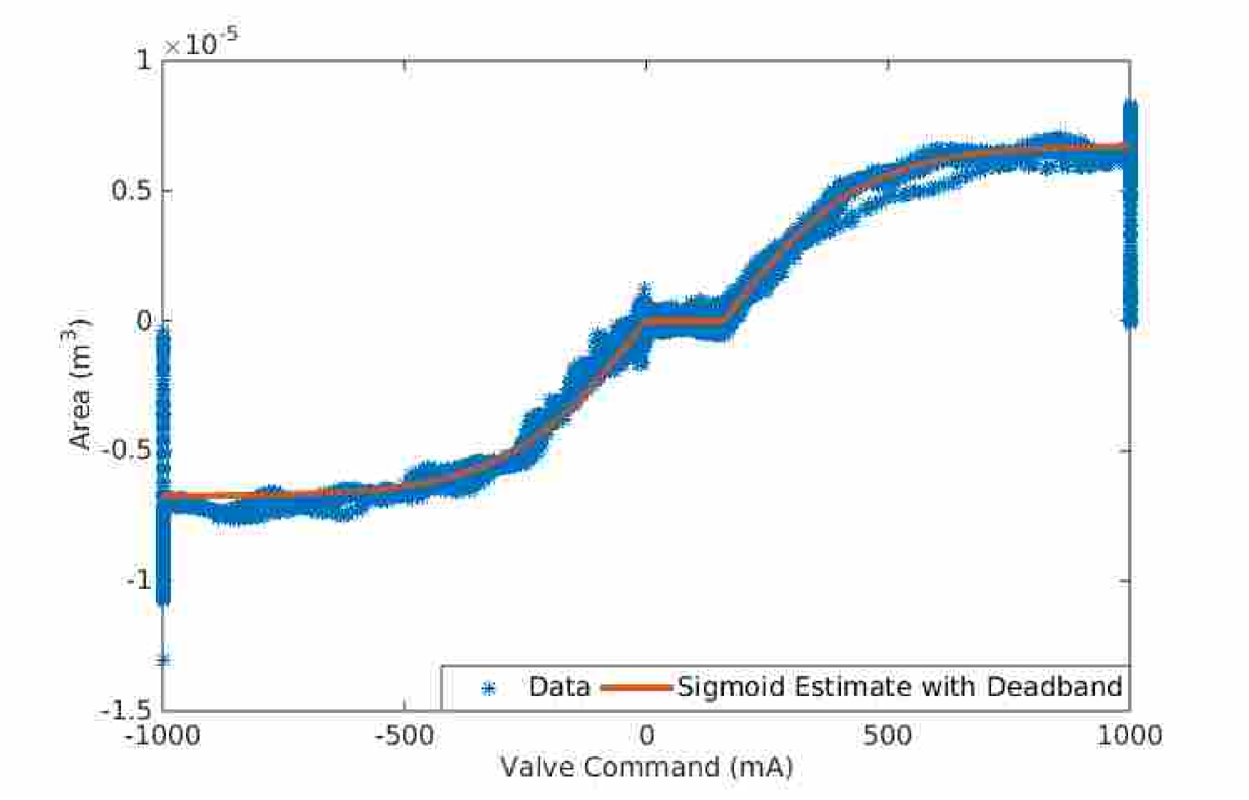

Figure A.2: The effective valve inlet area is directly related to mass flow. This graph then shows how a mass flow is converted to a valve command.

$$A_v C_f = \frac{s_2}{s_3 + e^{(s_1 * cmd)}} + s_4, \tag{A.16}$$

where $A_v C_f$ is the effective area, *cmd* is the commanded valve current and s_1, s_2, s_3, and s_4 determine the fit. Using the sigmoid fit with proper logic to handle the dead band, mass flows from the model can be mapped to a current to be applied at the valves.

Using MPC to determine the desired mass flows, Equations A.15 and A.16 can be used to determine valve electrical currents to maintain the desired mass flow effectivly controlling angle and stiffness. The results can be seen in Figure A.3 when applied to the grub using CVXGEN as a solver for the controller. The solver fails to converge until around 24 seconds. When the solver does converge, the controller saturates the commands back and forth like a bang bang controller.

By switching from the mass flow rate to a first order pressure model, where the pressures in each actuator are maintained by a PID pressure controller, the controller performance is much better. The controller performance on the grub using a first order pressure model for the gas dynamics

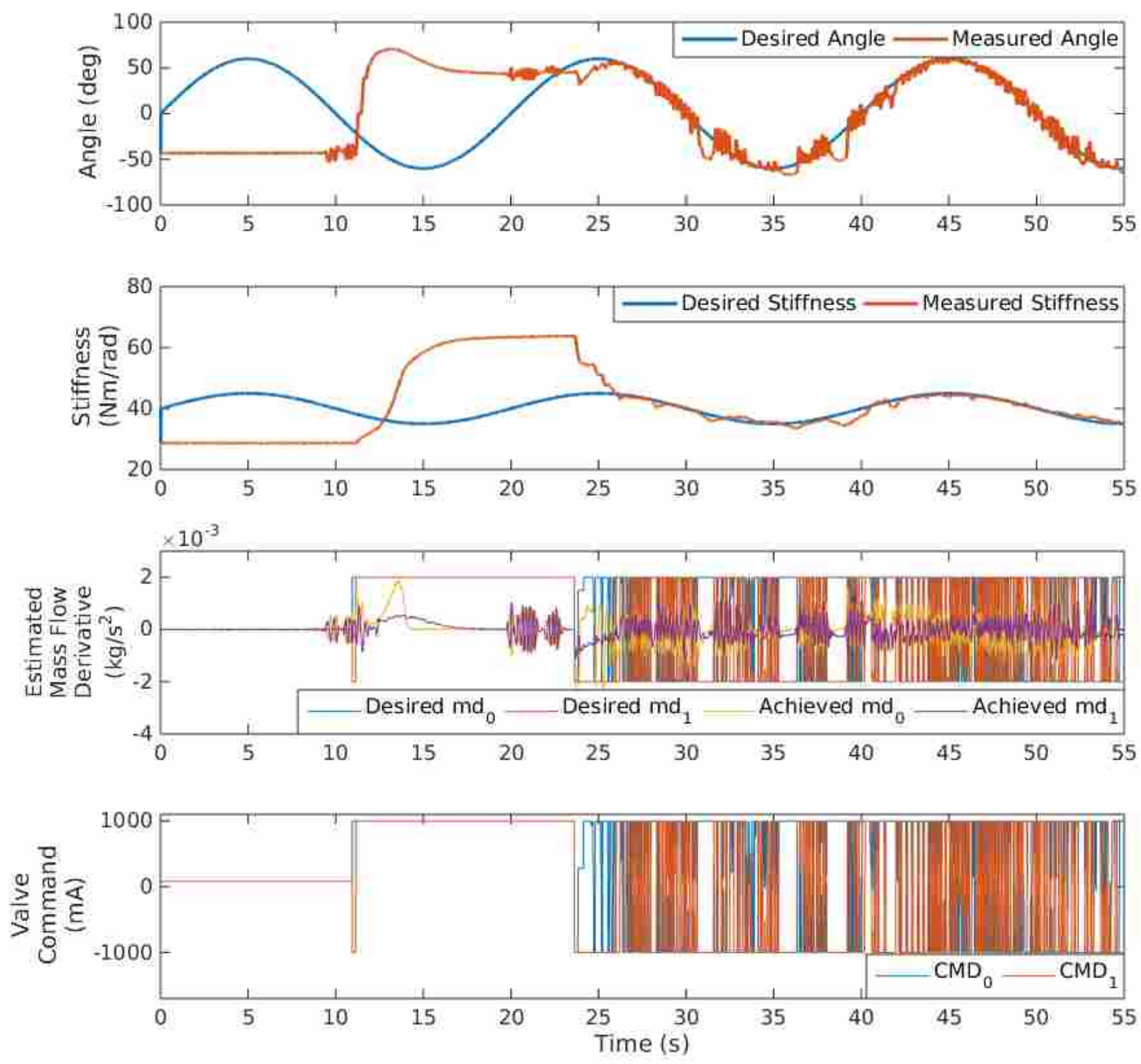

Figure A.3: MPC response on the grub using mass flow rates as the system input

can be seen in Figure A.4. The optimizer used was the sequential least squares programming option in the python package scipy using the minimize function within the optimization tool. There is much better tracking of the desired stiffness and angle signals and there is not the same saturation on the inputs as was seen in Figure A.3. Even though a first order pressure model does not describe the gas dynamics perfectly, it is accurate enough for good initial control of the grub.

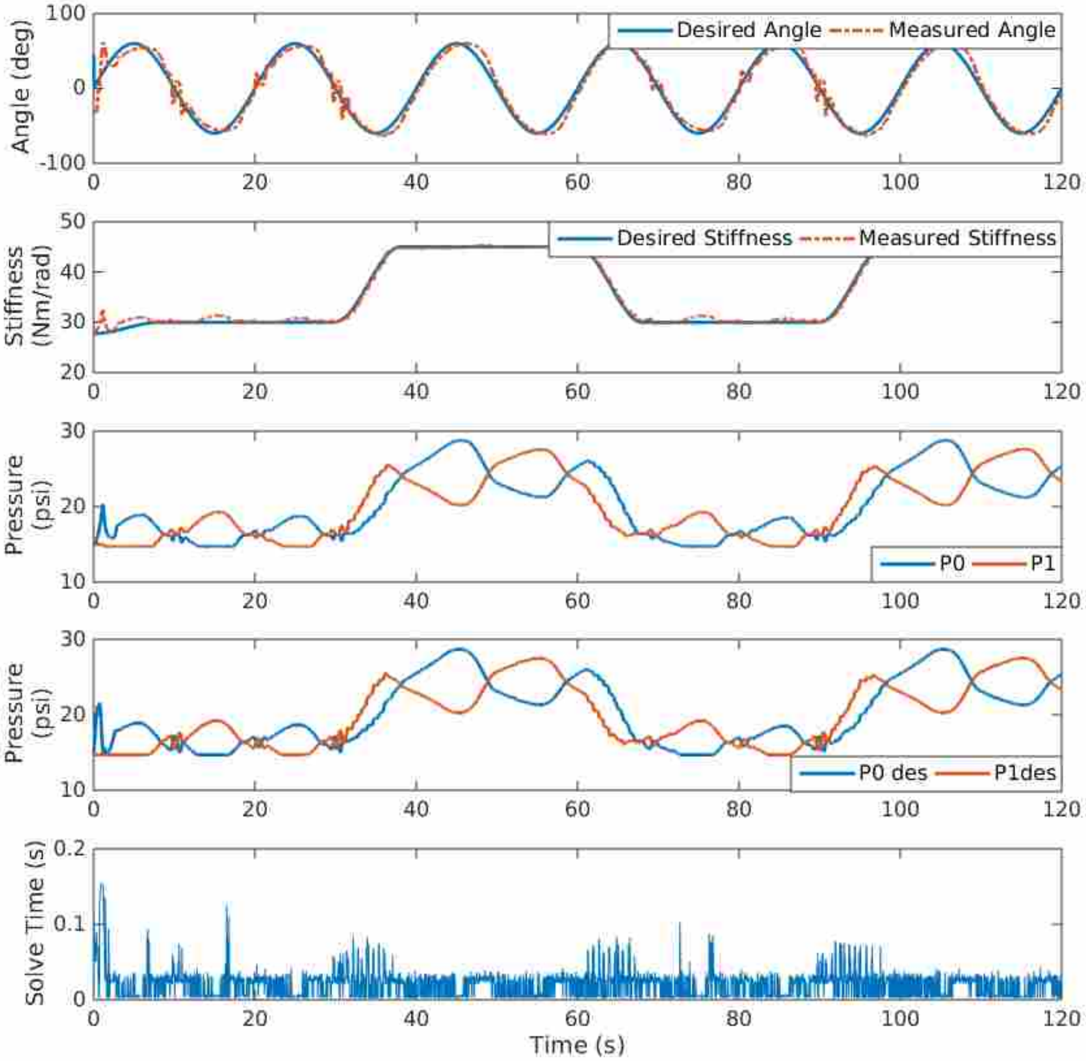

Figure A.4: MPC response on the grub using a first order pressure model for the gas dynamics