CONTROL OF ROBOTIC JOINTS USING PRINCIPLES FROM THE EQUILIBRIUM POINT HYPOTHESIS OF ANIMAL MOTOR CONTROL

A Thesis
Presented to
The Academic Faculty

by

Shane A. Migliore

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
June 2004

CONTROL OF ROBOTIC JOINTS USING PRINCIPLES FROM THE EQUILIBRIUM POINT HYPOTHESIS OF ANIMAL MOTOR CONTROL

Approved by:

Dr. Stephen DeWeerth, Advisor

Dr. Robert Butera

Dr. Lena Ting

Date Approved: 3 June 2004

ACKNOWLEDGEMENTS

This work would not have been possible without the help of several people:

- To my advisor Stephen DeWeerth and my review committee who have provided guidance and feedback along the way.
- To the members of the NeuroLab for all of their support...specifically, the intuitive Edgar Brown; the competitive Mike Reid; the entertaining Jimmy 'Danger' Ross; the diligent and bubbly Kyla Ross; the mechanically inclined Jevin Scrivens; the wondergrad Shawn O'Connor; the king of subtlety Richard Blum; the not-so-subtle William Gerken; my fellow fieldmarshalls Kate Williams and Kartik Sundar; the sarcastic and witty Michael Q. Sorensen; the ever-patient Carrie Williams; and the contemplative Samir Das.
- To my family for believing in me and supporting me throughout the years.
- To my financé Bobbi-Jo for your love and encouragement at every step along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Biological systems are able to perform complex movements with high energy-efficiency and, in general, can adapt to environmental changes more elegantly than traditionally engineered mechanical systems. The Equilibrium Point Hypothesis describes animal motor control as trajectories of equilibrium joint angle and joint stiffness. Traditional approaches to robot design are unable to implement this control scheme because they lack joint actuation methods that can control mechanical stiffness, and, in general, they are unable to take advantage of energy introduced into the system by the environment. In this paper, we describe the development and implementation of an FPGA-controlled, servo-actuated robotic joint that incorporates series-elastic actuation with specially developed nonlinear springs. We show that the joint's equilibrium angle and stiffness are independently controllable and that their independence is not lost in the presence of external joint torques. This approach to joint control emulates the behavior of antagonistic muscles, and thus produces a mechanical system that demonstrates biological similarity both in its observable output and in its method of control.

CHAPTER I

INTRODUCTION

1.1 Traditional Approaches to Robot Control

Terrestrial locomotion is a complex behavior that requires animals to maneuver through non-uniform environments and to handle unpredictable changes in their surroundings. Legged locomotion offers unique benefits over other approaches (*e.g.*, wheeled vehicles) because it increases an robot's ability to avoid obstacles and to manage various types of discontinuities (*e.g.*, span gaps, climb stairs) [21]. Similarly, segmented limbs give animals the ability to reach, push, and otherwise manipulate their environment. For decades, researchers have built robots with actuated joints in an effort to create machines that can perform complex tasks efficiently. However, the control of movement in non-ideal environments has proven to be complex problem.

A few researchers have used biped robots and a variety of control strategies to study walking [2], [7], [13], [18]. One of the most common is the trajectory-tracking technique in which robots move by following pre-planned leg position or force trajectories [2]. A limitation of this method is that it is completely feed-forward and is thus sensitive to perturbations. A more advanced version of this approach is the model-based technique in which algorithms are used to determine ideal placements of legs to maintain stability. This technique requires the robot either to have an *a priori* knowledge of its environment or to have sensors that can quickly and sufficiently detect all relevant details of the environment that are needed by the algorithm. The algorithm then compares current limb positions with these environmental constraints to determine the appropriate movements of each leg [7]. Similar algorithms can also be used to make predictions about upcoming moves and begin preparing for the move ahead of time. For example, a robot may begin to lean to one side when a turn to that side is approaching. A drawback to the model-based technique is the need to have direct, precise control over joint angles or forces to produce movements

that correspond to the algorithm's output. Imprecision can result in bad foot placement and become a source of instability.

In the late 1980's Raibert demonstrated that complex control algorithms were not necessary to control running in a three-dimensional, single-legged, hopping robot [18]. The robot was controlled using a few simple physics-based rules that computed the appropriate actuation of its leg based on the the results of the last step. Another significant contribution made by Pratt and Raibert made was the introduction of series-elastic actuators [16], [11]. This term refers to the intentional inclusion of an elastic element between the output of a geared-down motor and the load. Its purpose is to reduce the non-linear effect of load jitter or shock as well as to significantly reduce the output inertia of the motor. With a lower output inertia, the load is not held as rigidly and the natural dynamics of the actuated device become important.

Another approach to robot locomotion control—inspired by a walking toy—involves using the passive mechanics of a robot as the sole source of control [17], [13]. Because the robot has no actuators, it requires a small downhill slope to provide energy. With careful design, stable locomotion can be produced in either two or three dimensions [22]. This control scheme suffers from sensitivity to perturbations, initial conditions, and the angle of the downhill slope. (Recent computational models have demonstrated that small actuators can be used on otherwise passive robots strictly to compensate for changes in slope and thus maintain the needed level of input energy [15].)

Robots that closely emulate the mechanics and control of biological systems can provide a research tool for testing theories of animal neural control architectures. However, it is critical that the method by which robots actuate their joints, and thus control their movements, be modeled as close as possible to biology.

1.2 The Equilibrium Point Hypothesis

While a good deal is known about the output of the neuromuscular system during locomotion—such as motoneuron and muscle activity—relatively little is know about how animals control and coordinate muscular contractions to produce desired motions. The λ-version of the

Equilibrium Point Hypothesis (EPH) of motor control seeks to explain joint movements at a macroscopic level [5],[6]. Because it is unconcerned with low-level details such as muscle fiber composition or activity of individual neurons [8], it is ideal for use in a robotic system that can at best emulate macroscopic views of motor activity.

The primary aspect of this hypothesis is that there is a single control variable, λ, that can be set by the central nervous system independent of the current or past kinematic state of the animal. Although no evidence has directly linked λ to a biologically measurable quantity, this parameter is generally considered to correspond to the muscle length at which autogenic feedback begins activating its motoneuron pool [6]. This method of control does not allow the central nervous system to determine the level of α-motoneuron activity. For slow muscle stretches, activity is instead determined by the relationship between λ and muscle length, where muscle length is set by the interaction between the joint actuated by the muscle and externally applied torques [6].

Considering a single degree of freedom (DOF) joint actuated by a single muscle, the level of muscle activity can be represented as follows:

$$A = \begin{cases} L - \lambda & \text{for } L \geq \lambda \\ 0 & \text{for } \theta < \lambda \end{cases} \tag{1}$$

where A is the level of static muscle activation and L is the muscle length [1]. Note that activity ceases when the muscle is shorter than λ because the stretch reflex is silenced.

The EPH can be expanded to represent the contributions of the stretch-rate sensitivity of group Ia afferents by modifying the definition of λ to include velocity as follows:

$$\lambda^* = \lambda - \mu\omega \tag{2}$$

where μ is a second central control parameter and ω is the angular velocity of the actuated joint [20]. For the purposes of this research, we assume that the velocity component is negligible. To justify this assumption, we note that μ is commonly approximated to be small (0.58) [12], [1] and that, in this research, data was collected during steady-state conditions. Therefore, for this research, Equation 2 becomes:

$$\lambda^* = \lambda \tag{3}$$

By controlling the λ values for antagonistic muscles, the movement of joints can be influenced. Feldman and Latash [6], [12] have defined two central command signals to describe the interaction of antagonistic muscles—R (reciprocal) and C (coactivating)—as follows:

$$C = \frac{\lambda_{FLEX} + \lambda_{EXT}}{2} \tag{4}$$

$$R = \frac{\lambda_{FLEX} - \lambda_{EXT}}{2} \tag{5}$$

where λ_{FLEX} is the λ of the flexor muscle and λ_{EXT} is the λ of the extensor muscle. If a pair of antagonistic muscles had identical properties (*i.e.*, identically matched motoneurons and muscle composition, symmetric attachment points, etc.) and no external load is applied to the joint, then the time trajectory of R is proportional to the trajectory of the joint's equilibrium angle, while the trajectory of C is proportional to the trajectory of the joint's stiffness [5]. Joint stiffness is defined as its resistance to angular changes:

$$S = \frac{dT}{d\theta} \tag{6}$$

where T is the net torque applied to the joint by the muscle pair.

Because stiffness defines the level of compliance with which a joint reacts to external forces, it has direct implications on how individual joints and whole limbs are controlled in motor tasks such as balance maintenance or locomotion. The ability to control a joint's stiffness independent of its equilibrium angle is crucial. This is evidenced in experiments by Nichols and Houk [14] that demonstrated that joint stiffness, rather than muscle length, is the mechanical property regulated by stretch reflex. Since stiffness regulation is important, it is not possible to accurately emulate the dynamics of a biological joint without some method to modify the joint's stiffness as external conditions or desired performance change.

In this research, it was necessary to correlate λ with muscular forces. Isometric analysis of cat soleus muscle by Joyce [9] has demonstrated that the relationship between muscle tension and motoneuron stimulus rate can be approximated as a quadratic function for stimulus levels less than 10 Hz. (Above 10 Hz, the the relationship saturates and a maximal tension level is reached.) Similarly, Contreras-Vidal [3] has demonstrated that the active

component of muscle contraction can be modeled as a quadratic function:

$$F = \rho(L - \lambda)^2 \tag{7}$$

where ρ is a muscle-specific parameter and L is the muscle length. Joyce's data correlates well with this model, provided that motoneuron firing rate is inversely proportional to λ. With this assumption, an increase in motoneuron firing rate produces a quadratic increase in muscle force. By defining the force-length relationship of each muscle, the equilibrium point and stiffness of a joint can be controlled by appropriately setting the muscles' λ values.

1.2.1 Effects of External Torque on Joint Control

Equations 4 and 5 demonstrate that when no external torque is applied to a joint, unique expressions can be derived for the joint's equilibrium angle (proportional to R) and stiffness (proportional to C). However, with the addition of external torques, changes in R and C lose their uniqueness (*i.e.*, changes in C affect joint equilibrium angle and changes in R affect joint stiffness) [20]. As an example, Figure 1 shows a joint that has been subjected to an external torque. The initial torque-angle curves for the agonist and antagonist muscles are shown as the red and blue solid traces, respectively. The sum of these curves yields the applied joint torque (solid green trace), which is the torque produced solely by the muscles. When a 2.0 mNm external torque is applied to the joint, an equilibrium point, EP_1, is established at the intersection of the applied joint torque curve and the external torque line (black dashed trace). Next, the level of co-contraction is increased and the new agonist, antagonist and applied joint torque curves are shown as red, blue, and green dashed traces, respectively. Note that the applied joint torque curve has simply rotated about the origin, resulting in a new intersection point, EP_2, with the external torque line. Thus, a change in co-contraction level, C, has caused a change in the equilibrium joint angle. Similarly, in the presence of external torques, changes in R yield changes in both equilibrium angle and in joint stiffness. (This effect is not shown.)

While the loss of R and C uniqueness complicates the computation of a joint's equilibrium angle and stiffness, deterministic relationships are still attainable as described in Section 2.2.

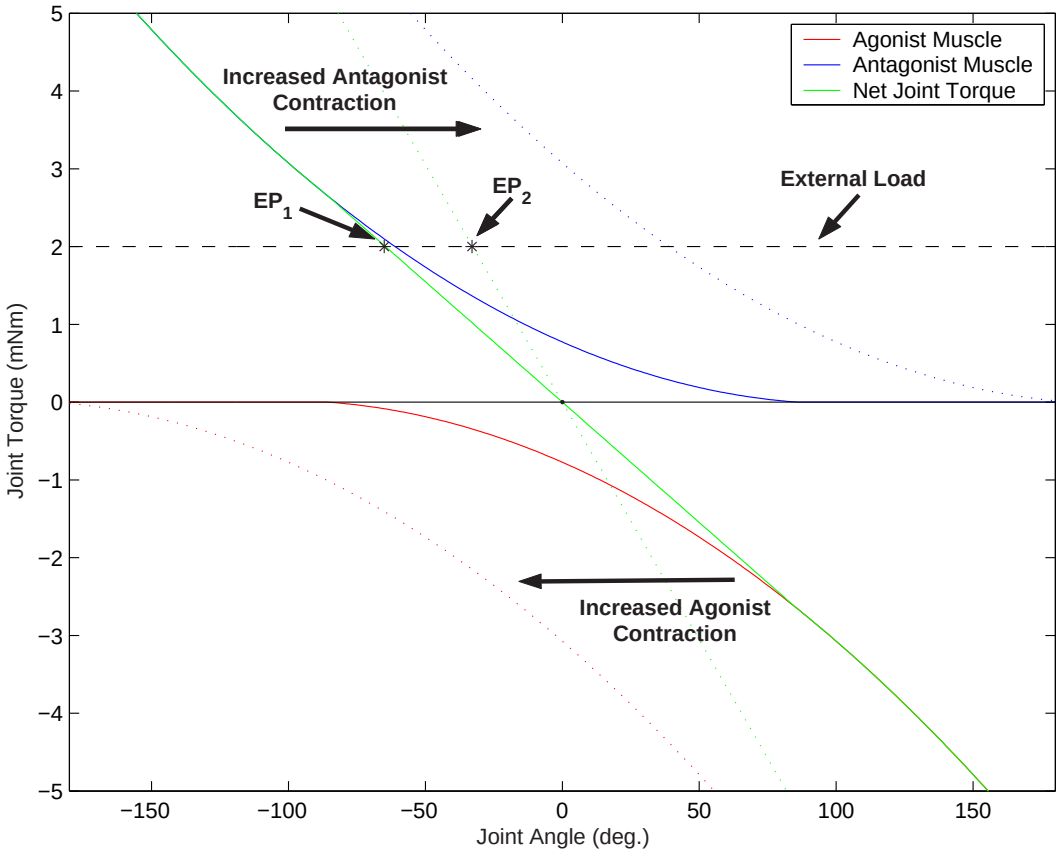

Figure 1: The effect of increasing co-contraction level on the equilibrium angle of a joint subjected to an external torque. Note that as co-contraction increases (the solid blue and red traces become dashed), the net joint torque curve becomes more steep (the solid green trace becomes the dashed green trace). To maintain a net zero torque, the equilibrium joint angle must shift from EP_1 to EP_2. Thus, a change in C has caused a change in the equilibrium joint angle.

CHAPTER II

SYSTEM MECHANICS

This chapter addresses the details of the theory and implementation of the mechanical elements of the robotic system. Relevant mechanical quantities are defined and then given context through their relationship to the Equilibrium Point Hypothesis. The design and development of a specialized nonlinear spring and the robotic joint are also discussed.

2.1 Overview

To understand the control technique that will be used, an understanding of the mechanics of joint control is necessary. Figure 2 shows a schematic drawing of the mechanics of the joint that is used in this research. Note that two servos are attached to the joint through a series connection of cables and springs. Note also that both servo pulleys have the same radius (R_J), and that this radius is larger that of the joint pulley (R_S), so some gearing up is present. There are three important angles in this system—α, β, and θ—that correspond to the agonist servo angle, the antagonist servo angle, and the joint angle, respectively. As a convention, we have chosen that clockwise rotations represent positive angular displacements for α and θ and represent negative a angular displacement for β.

The two fundamental mechanical quantities used in this research are joint equilibrium angle and joint stiffness. A joint's equilibrium angle is defined as the angle at which it would come to rest in the absence of external forces. Joint stiffness is defined as the resistance to angular displacements of the joint from its equilibrium angle, or the derivative of applied torque with respect to angular displacement (Equation 6). Mechanically, joint stiffness is created by applying equal and opposite torques to the joint via the two servos so that any angular displacement produced by an external torque causes an increased loading of one spring and a decreased loading of the other.

It has been shown that stiffness of biological joints is both linear and elastic about a set

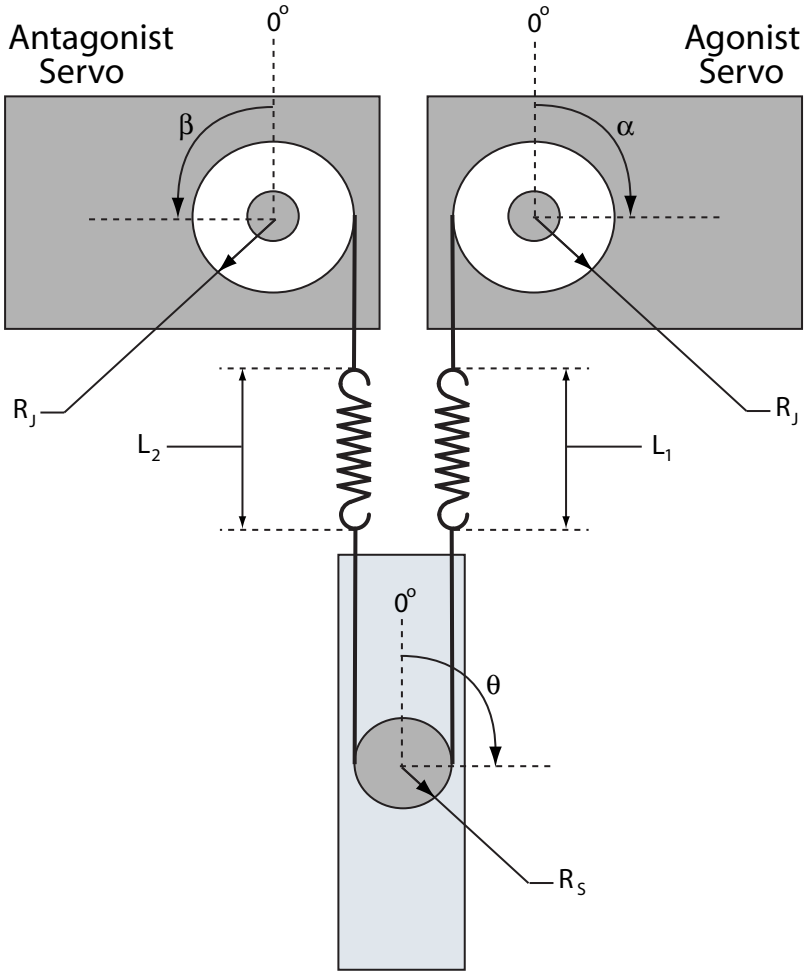

Figure 2: The mechanics of a joint actuated with antagonistic actuators using series-elastic cables.

operating point [10]. Section 2.2 shows that this behavior can be produced through series-elastic actuation of a mechanical joint with either linear or quadratic springs. Although linear springs are easier to work with and are readily available, they do not allow the joint stiffness to be varied because, under any amount of co-contraction, the movement of a joint causes equal and opposite length changes in the two springs and hence equal and opposite force changes. Because the sum of the spring forces is always constant, the resulting stiffness is always proportional to the sum of the two spring constants. The desired behavior is that as co-contraction increases, stiffness increases. Quadratic springs produce the same performance as linear springs for a set operating point, but also allow the joint stiffness to be varied as needed. (The details of this discussion are presented in Section 2.2.)

2.2 Application of the Equilibrium Point Hypothesis to Joint Control

The control of servos requires that they receive a desired angle signal as input at all times; it is this angle that they are designed to reach using any available torque. A primary contribution of this research is the design and implementation of an algorithm that allows a user to control a joint by specifying its equilibrium angle and stiffness rather than by specifying individual servo angles. For the following discussion, we will use the assumption that the system is calibrated such that when the joint angle and both servo angles are set to 0^o, the cables are idle (*i.e.*, the cables are taut but are conveying no tension). This configuration is shown in Figure 2. As mentioned in Section 2.1, the use of linear springs in a series-elastically actuated joint prevents the application of variable stiffness. The achievable stiffness with linear springs is:

$$S_{Linear} = \frac{-\pi R_J^2}{180}(k_1 + k_2) \qquad \text{for } \theta \in \{-\alpha \leq \theta \leq \beta\} \tag{8}$$

where k_1 and k_2 are the two linear spring constants. Because all parameters in this equation are constants (*i.e.*, there is no dependence on servo angles), the stiffness will always be constant, provided both springs are always in tension. Tension is guaranteed when $-\alpha \leq \theta \leq \beta$.

Figure 3 shows the effect of co-contraction on stiffness when linear springs are used. The

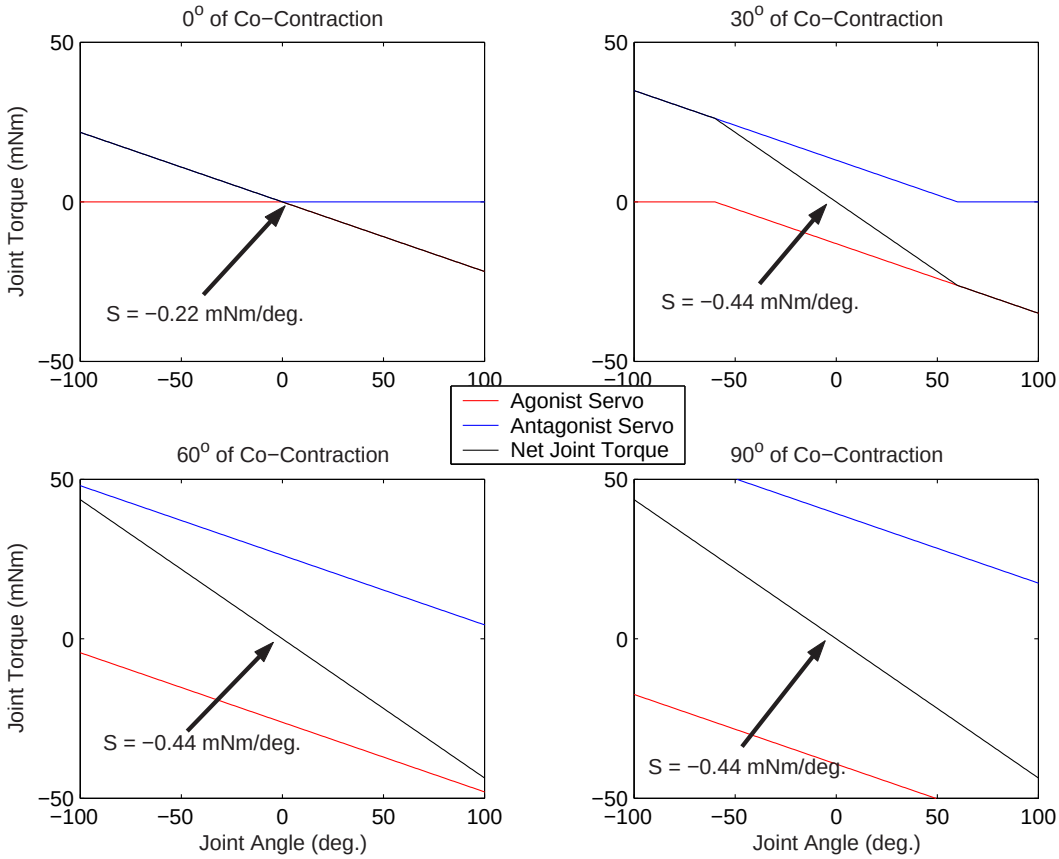

Figure 3: The effect of co-contraction on net joint torque using linear springs. Note that in all cases of non-zero co-contraction, joint stiffness (the slope of the black trace) is constant near the joint's equilibrium angle (0^o).

black trace in each panel represents the applied joint torque, which is the torque applied to the joint by the servos. Note that as co-contraction level increases from zero, the slope of the operating region (*i.e.*, the region in which both cables are in tension) is always -0.44 $\frac{\text{mNm}}{\text{deg}}$.

To produce variable joint stiffness, we have chosen to use nonlinear springs as part of the actuation system. In an effort to maintain similarity with Equation 7, we have chosen to implement springs with quadratic force-length characteristics such that:

$$F = a(L - L_0)^2 + b(L - L_0) + c \tag{9}$$

where L is the spring's length, L_0 is the spring's resting (no-load) length and a, b, and c are spring constants. Using these springs and the mechanical configuration from Figure 2, the equation for applied joint torque, T, is:

$$T = -\frac{\pi R_J}{180^2}(180b + a\pi R_S(\alpha + \beta))(R_S(\alpha - \beta) + 2R_J\theta) \qquad \text{for } \theta \in \{-\alpha \leq \theta \leq \beta\} \tag{10}$$

Furthermore, the equation for joint stiffness ($\frac{dT}{d\theta}$) becomes:

$$S = -\frac{2a\pi^2 R_J^2 R_S}{180^2}(\alpha + \beta) - \frac{b\pi R_J^2}{90} \qquad \text{for } \theta \in \{-\alpha \leq \theta \leq \beta\} \tag{11}$$

In this case, stiffness is indeed dependent on the servo angles—α and β. In fact, the stiffness equation is linearly dependent on the co-contraction, C, with a constant offset proportional to b, the linear term of the quadratic springs. Figure 4 shows a plot of the applied joint torques that result when quadratic springs are used by co-contracting servos. It can be seen that in the operating region, a linear slope is present in the applied joint torque. Furthermore, this slope increases in magnitude as co-contraction increases.

Figure 5 is a plot of stiffness versus co-contraction for quadratic springs (blue data points), and for linear springs (red data points). Note that when using linear springs, the stiffness is constant for non-zero levels of co-contraction; in the case of quadratic springs, stiffness is linearly related to the level of co-contraction.

While the quadratic spring of the form shown in Equation 9 appears to yield the desired joint controllability, there is one significant limitation. Equation 11 demonstrates that the stiffness the joint is proportional to the co-contraction and has a constant offset proportional to b, the linear term in the quadratic force-length relationship. Because of this offset, the

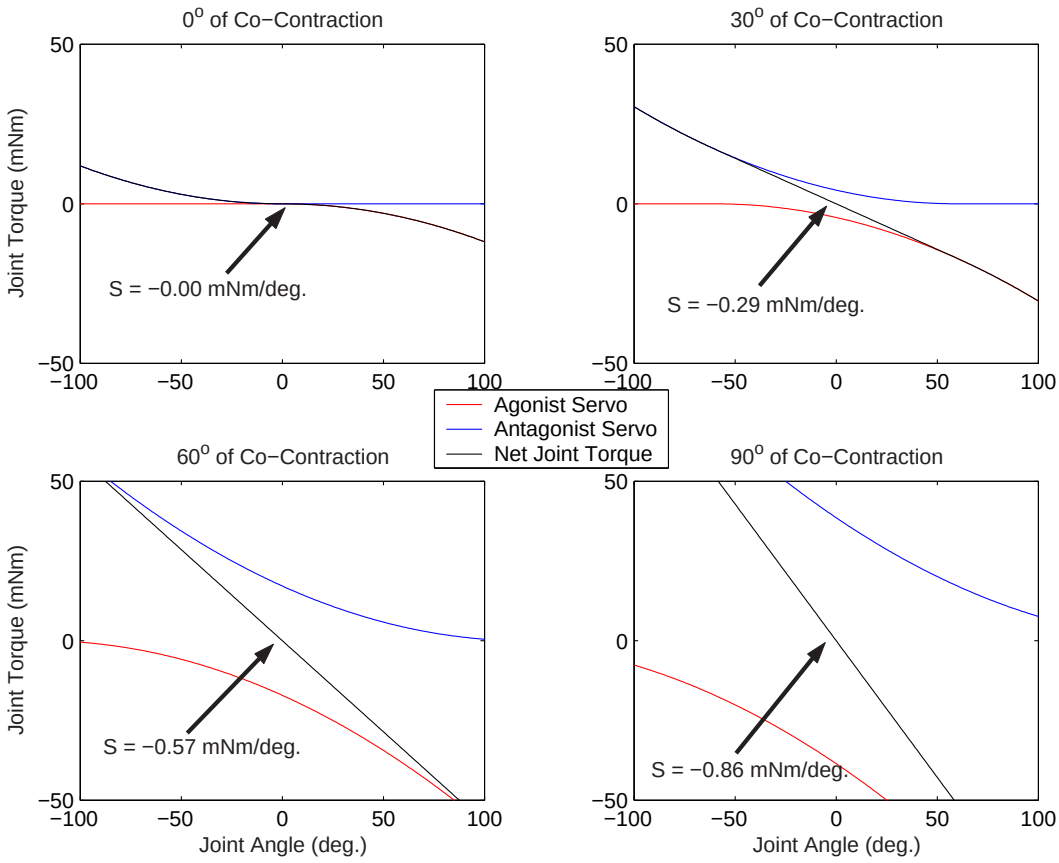

Figure 4: The effect of co-contraction on net joint torque using quadratic springs.

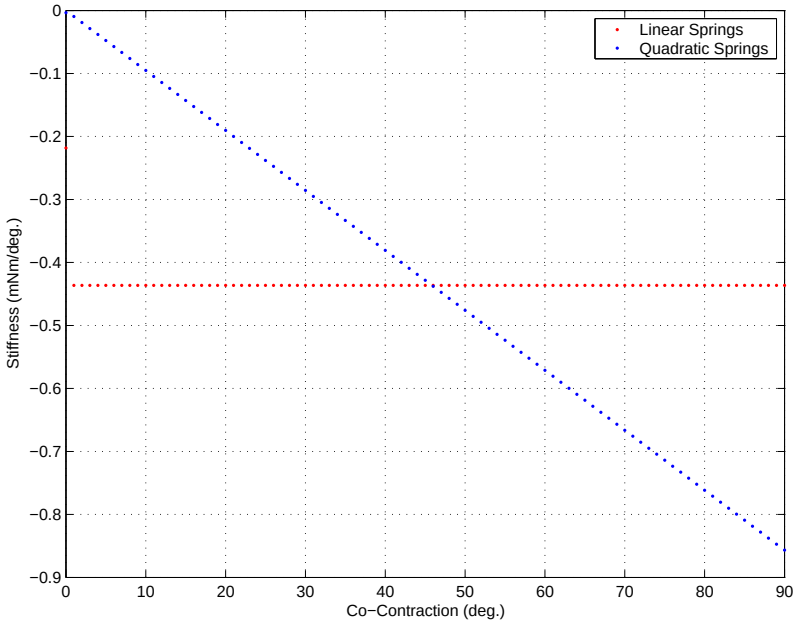

Figure 5: Comparison of the joint stiffness values resulting from various levels of co-contraction when using linear and quadratic springs.

joint stiffness can not be reduce in magnitude to a value less than $\frac{b\pi R_J^2}{90}$. It is therefore desirable for the spring to take the form:

$$F = a(L - L_0)^2 + c \tag{12}$$

The benefits of springs of this form are described in more detail in by English [4]. Section 2.3 describes the steps taken to implement springs of this form. Note that in the remainder of this section, calculations will include the linear spring term because non-idealities in the manufacturing process tend to yield non-zero values of b.

The equilibrium joint angle of the system can be found by computing the joint angle at which the applied joint torque is equal and opposite that of an externally applied torque. Setting the applied torque (from Equation 10) equal to the externally applied torque, which will be referred to by the variable *Load*, yields the following equation for θ_{EQ}, the joint's equilibrium angle:

$$\theta_{EQ} = -\frac{180^2 Load + \pi R_J R_S(\alpha - \beta)[180b + a\pi R_S(\alpha + \beta)]}{2\pi R_J^2[180b + a\pi R_S(\alpha + \beta)]} \tag{13}$$

Note that the equation for joint stiffness (Equation 11) is independent of external torques because it is assumed that the externally applied torque does not vary with θ, and therefore, $\frac{dT}{d\theta}$ does not include any term containing *Load*.

We can solve Equations 11 and 13 for α and β so that θ_{EQ} and S are controllable by simply setting appropriate servo angles. The equations for the servo angles are:

$$\alpha = -\frac{90(b\pi R_J^2 + 90S)}{a\pi^2 R_S R_J^2} - \frac{R_J}{R_S}\left(\theta_{EQ} - \frac{Load}{S}\right) \qquad \text{for } S \in \{S < 0\} \tag{14}$$

$$\beta = -\frac{90(b\pi R_J^2 + 90S)}{a\pi^2 R_S R_J^2} + \frac{R_J}{R_S}\left(\theta_{EQ} - \frac{Load}{S}\right) \qquad \text{for } S \in \{S < 0\} \tag{15}$$

To summarize, by using nonlinear (quadratic) springs, the equilibrium angle and stiffness of the mechanical joint can be linearly and independently controlled, provided that the two actuation cables for each joint always remain in tension (*i.e.*, $-\alpha \leq \theta \leq \beta$).

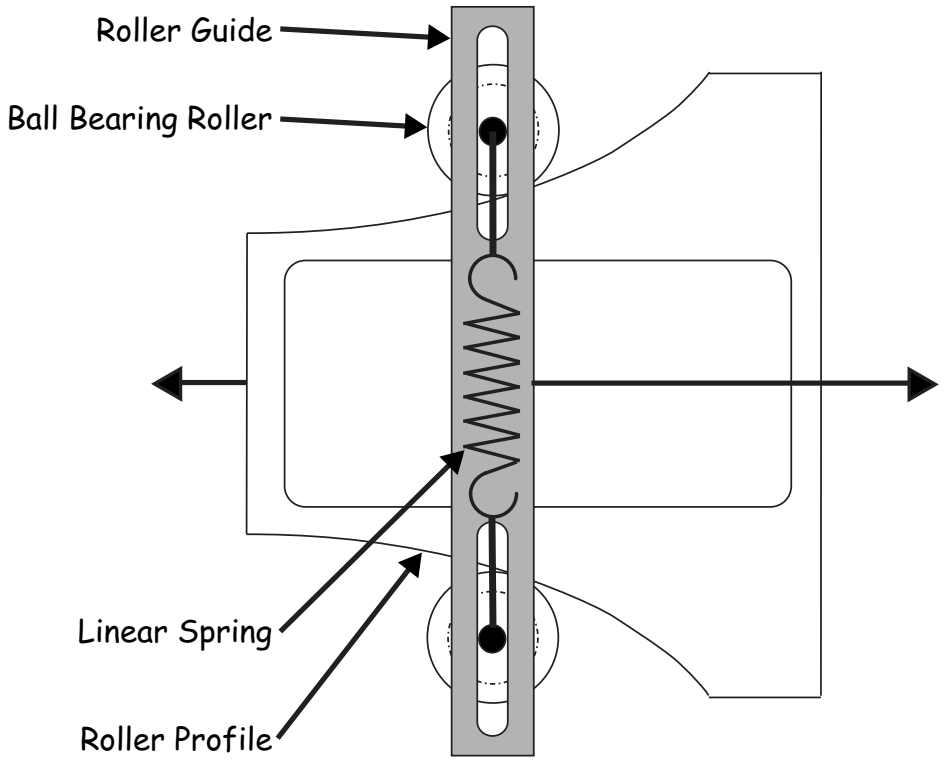

Figure 6: Quadratic spring device implemented using elastically coupled bearings that roll on a nonlinear profile.

2.3 Nonlinear Springs

The implementation of a device that could provide the desired quadratic force-length relationship proved to be one of the most challenging portions of this research. Several alternate approaches were considered (Section 2.3.3). The following sections describe the analysis and implementation of the final design, a device that uses linear springs to produce an overall quadratic force-length relationship. The procedure used for this design was general enough that it can be used to implement any continuous force-length relationship.

2.3.1 Mathematical Analysis

A schematic drawing of the nonlinear spring's design is shown in Figure 6. The primary concept for this device is that when a stretch of length l is applied to it (that is, the roller guide is moved relative to the profile a distance l), the springs (one is visible in the foreground of Figure 6 and one is behind the device) are stretched as a nonlinear function of l such that the output force of the device is a quadratic function of l.

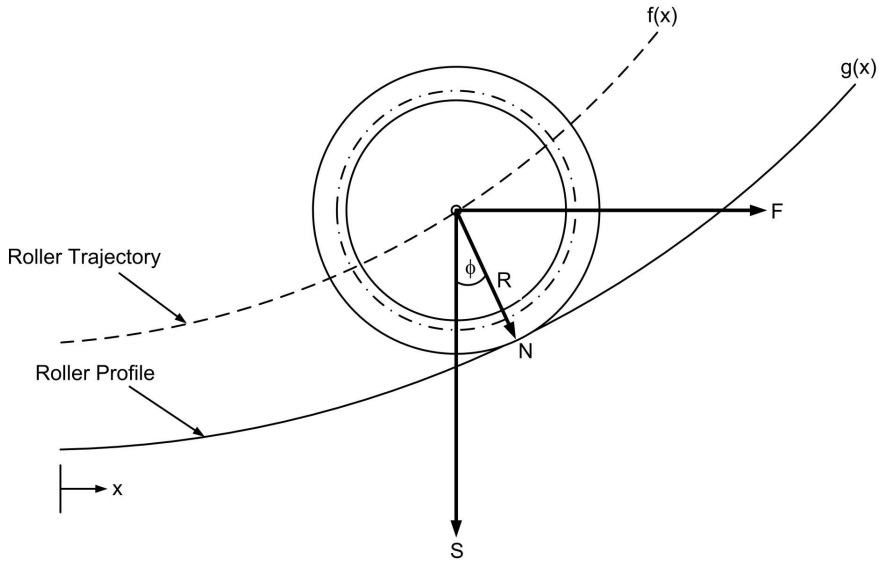

Figure 7: Variables used in analysis of spring device.

To begin analysis, a few variables need to be defined. Figure 7 shows an enlarged view of one of the ball bearing rollers used in the device. F represents half of the total output force of the device (a second force F is created by the other bearing), S represents the combined spring force applied to the rollers, N is the normal force of the bearing, ϕ is the angle the normal force makes with the vertical axis, and R is the radius of the roller. We define the independent variable, x, to have an origin at the left-most extreme of the profile, such that for all calculations $x \in \{x \geq 0\}$. $g(x)$ represents the surface of the roller profile on which the roller moves, and $f(x)$ represents the trajectory of the center of the roller. Using trigonometry, we can show the following relationships exist:

$$F = N \sin \phi \tag{16}$$

$$S = N \cos \phi \tag{17}$$

$$\frac{F}{S} = \tan \phi = \frac{dg(y)}{dy}\Big|_{y=x+R\sin\phi} \tag{18}$$

$$f(x) = g(y)\Big|_{y=x+R\sin\phi} + R\cos\phi \tag{19}$$

By definition, the force F acting on each roller is one half of the force exerted by the entire device. Thus F is one half of the overall desired force-length relationship of the

device:

$$F = \frac{ax^2 + c}{2} \tag{20}$$

The spring force, S, is defined as the sum of the forces exerted by both springs:

$$S = 4k[f(x) - L_0] + PL \tag{21}$$

where k is the linear spring constant, L_0 is the resting length of the spring, and the spring pre-load, PL, is the amount of force that must be applied to the spring for the coils to begin separating from each other. The factor of four in this equation results from two springs each being stretched on both ends (*i.e.*, both rollers are moving along trajectories of $f(x)$ causing the springs to actually be stretched to a length $2f(x)$ at any given x.

The overall objective of the following derivation is to determine $g(x)$, the function of the roller profile's surface. To begin, we note that:

$$\frac{F}{S} = \tan(\phi) = \frac{dg(x + R\sin(\phi))}{dx} \tag{22}$$

Substituting in expressions for F and S,

$$\frac{\frac{ax^2+c}{2}}{PL + 4k[f(x) - L_0]} = \frac{dg(x + R\sin(\phi))}{dx} \tag{23}$$

$$\frac{\frac{ax^2+c}{2}}{PL + 4k[g(x + R\sin(\phi)) + R\cos(\phi) - L_0]} = \frac{dg(x + R\sin(\phi))}{dx} \tag{24}$$

In the following series of equations, the independent variable y is used, which is a simple trigonometric transformation of the independent variable x, such that $y = x + R\sin(\phi)$.

$$\frac{\frac{a(y-R\sin(\phi))^2+c}{2}}{PL + 4k[g(y) + R\cos(\phi) - L_0]} = \frac{dg(y)}{dy} \tag{25}$$

Using Equation 18, we can derive two useful identities:

$$\sin(\phi) = \frac{\frac{dg(y)}{dy}}{\sqrt{1 + \left(\frac{dg(y)}{dy}\right)^2}} \tag{26}$$

$$\cos(\phi) = \frac{1}{\sqrt{1 + \left(\frac{dg(y)}{dy}\right)^2}} \tag{27}$$

Substituting these identities into Equation 25 and rearranging we obtain the final equation for $g(y)$:

$$\left(\frac{dg(y)}{dy}\right)^2\left(\frac{PL}{4k}\right)+\frac{dg(y)}{dy}\left[g(y)+R\left(\frac{1}{\sqrt{1+\left(\frac{dg(y)}{dy}\right)^2}}\right)-NL\right]- \frac{a}{8k}\left[y-R\left(\frac{\frac{dg(y)}{dy}}{\sqrt{1+\left(\frac{dg(y)}{dy}\right)^2}}\right)\right]^2-\frac{c}{8k}=0 \quad (28)$$

Because we were not able to find an analytical solution to this complicated differential equation, we opted to compute a numerical solution using Mathematica 4.1 (Wolfram Research, Inc., Champaign, IL). The blue trace in Figure 8 shows the numerical solution for $g(y)$. Also shown for comparison is $f(x)$, the desired trajectory of the ball bearing roller, as the red trace. To provide a solution to Equation 28, we choose linear springs parameters based on commercially available springs and we choose the device force-length parameters based on the desired performance of the joint. Table 1 shows all of the parameters used in the computation of the numerical solution.

Equation 28 is a closed-form equation that defines the trajectory, $g(y)$, of the roller profile's surface. It was derived by defining the desired force-length relationship to be that of Equation 20. However, this design approach generalizes such that F can be any continuous function. Therefore, the design of this nonlinear spring device can be modified to produce any arbitrary joint stiffness function (provided that the function is continuous). Generally, the effect of co-contraction on the joint stiffness function is proportional to the derivative of the individual springs' force-length relationship. That is, co-contraction produces no change in stiffness when linear springs are used; co-contraction produces a linear change in stiffness when quadratic springs are used. Thus, if we need joint stiffness to be some continuous function of co-contraction, we simply need to create roller profiles whose surface trajectories are proportional to the integral of that function.

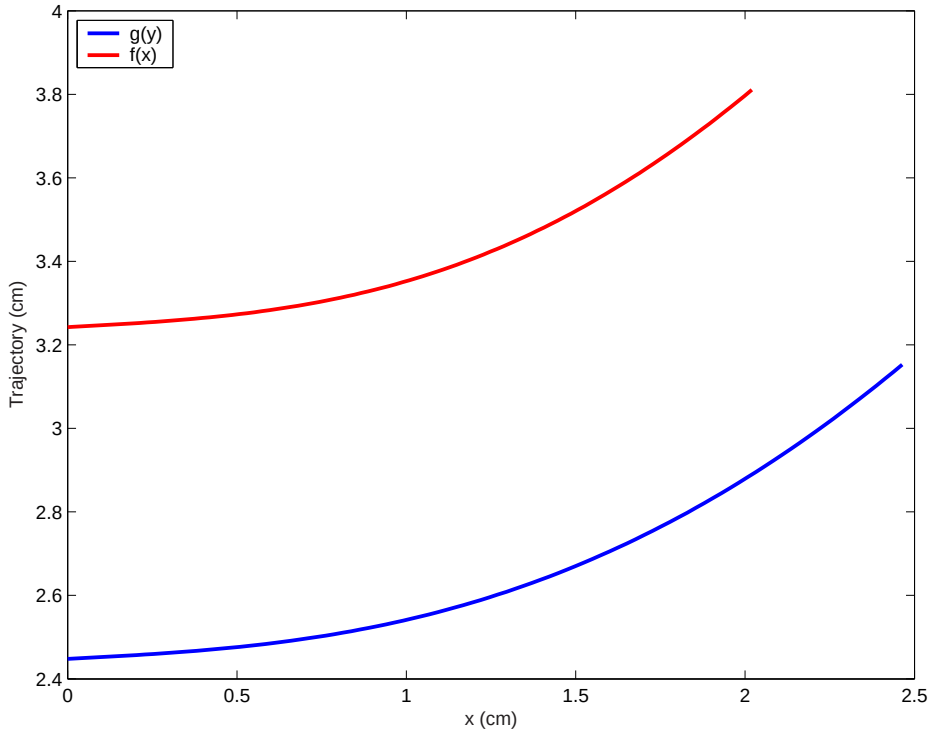

Figure 8: Numerical solution of Equation 28 for $g(y)$. Also show is $f(x)$, the desired trajectory of the ball bearing roller.

Table 1: Parameters used in producing the numerical solution to the differential equation (Equation 28) describing the curvature of the roller profile.

Parameter	**Value**
k	5.50 $\frac{\text{N}}{\text{cm}}$
NL	3.81 cm
PL	2.62 N
R	0.79 cm
a	8.27 $\frac{\text{N}}{\text{cm}^2}$
c	2.22 N

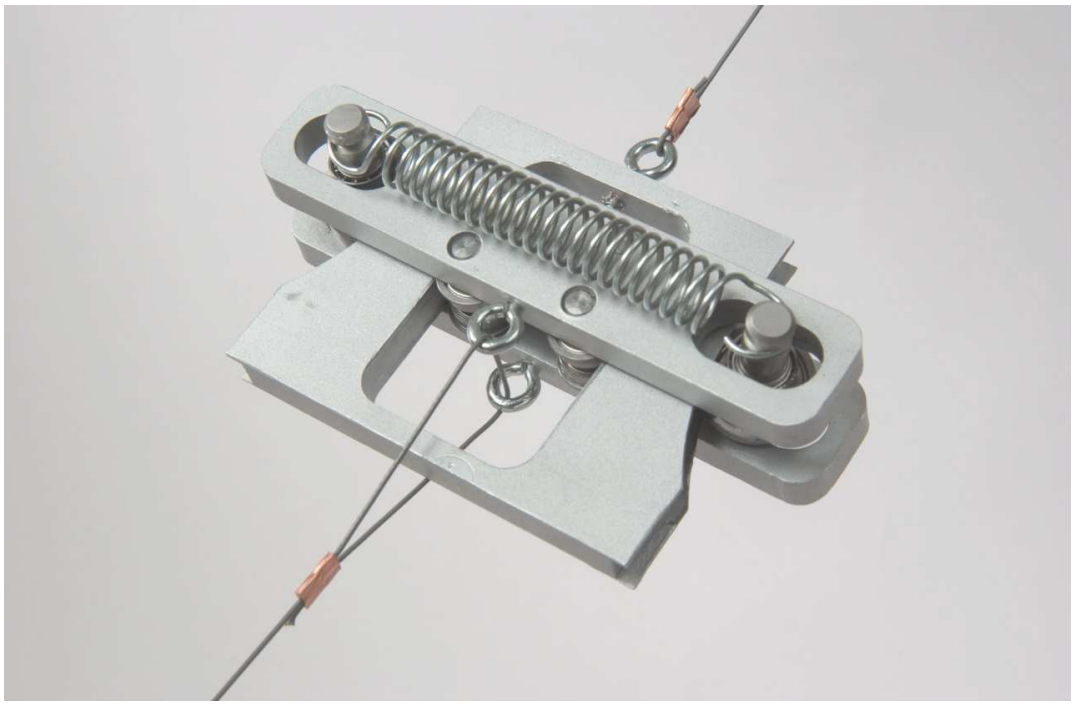

Figure 9: Implementation of the quadratic spring device.

2.3.2 Mechanical Implementation

Using this solution, we designed the spring device using AutoCad 2002 (AutoDesk, Inc., San Rafael, CA) and manufactured it using a CNC vertical milling machine. The resulting device (shown in Figure 9) is made primarily with aluminum to reduce the weight of the device, while providing a higher resistance to deformation than a non-metal such as teflon. Four press-fit dowel pins held the two roller guides to each other and prevented them from rotating relative to the profile. Two additional dowel pins were used to hold the rollers and to provide attachment points for the two linear springs. Screw eyes on the roller profile and on each of the roller guides provided attachment points for the cables used to stretch the device.

Figure 10 shows the results of characterizing the force-length relationship of two of these quadratic springs. Both springs demonstrate strong quadratic behavior ($R^2 = 0.9997$ and $R^2 = 0.9998$). However, some aspects of their performance are not ideal. The ideal force-length relationship is:

$$F = (8.27\frac{\mathrm{N}}{\mathrm{cm}^2})x^2 + 2.22\mathrm{N} \tag{29}$$

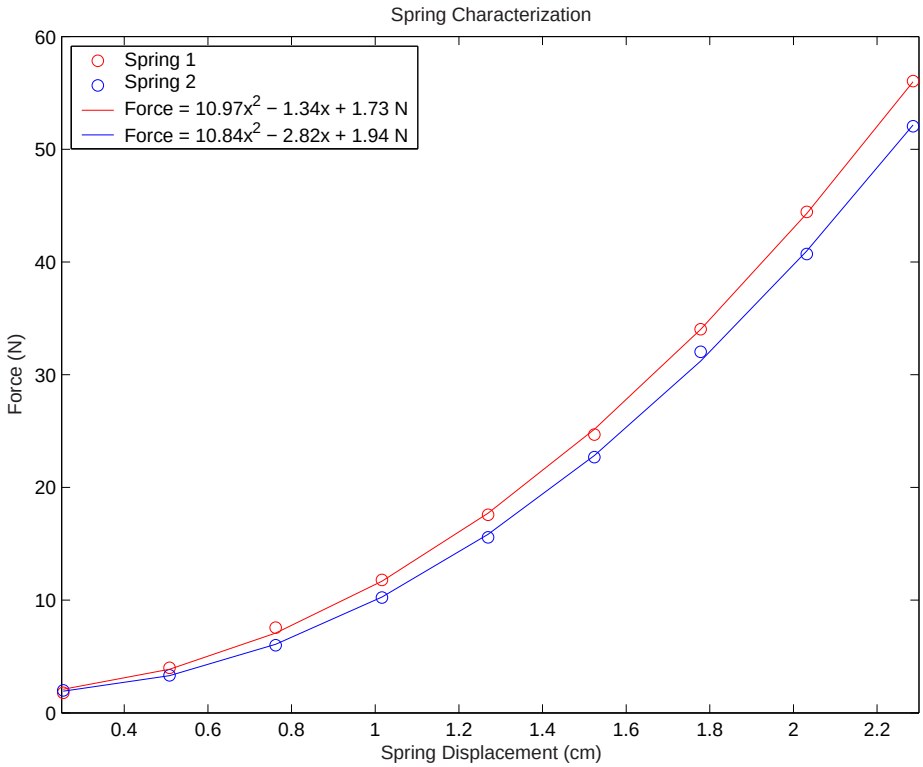

Figure 10: Force-length relationship of the quadratic spring device.

but the quadratic curve fits to the springs' data points yield the following best-fit equations:

$$\text{Spring 1:} \qquad F = (10.97\frac{\text{N}}{\text{cm}^2})x^2 - (1.34\frac{\text{N}}{\text{cm}})x + 1.73\text{N} \tag{30}$$

$$\text{Spring 2:} \qquad F = (10.84\frac{\text{N}}{\text{cm}^2})x^2 - (2.82\frac{\text{N}}{\text{cm}})x + 1.94\text{N} \tag{31}$$

Therefore, the quadratic terms of these equations have a 31.1% and 32.5% error, non-zero linear elastic terms have been introduced, and the constant offset terms have a 12.0% and 22.0% error. There are two major causes for these non-ideal characteristics—imperfect machining and load deformations of the roller profile.

During the automated machining of the roller profiles performed by the CNC mill, it was noted that the cutter feedrate specified was too high, resulting in the aluminum reaching excessive temperatures and being exposed to excessive forces during the cutting. This caused the metal to slightly deform during cutting (*e.g.*, expand and bow), which affected the depth of each cut. Additionally, analysis of the roller profiles after machining revealed that the high feedrate prevented some of the metal chips from being cleared properly by

the cutter. This caused some chips to be impacted into the components, resulting in an increase in the roller profiles' width and the introduction of small bumps along the path of the ball bearing rollers. As the profiles' width increases, the linear springs become more stretched, and therefore the force-length relationship of the nonlinear spring increases.

The other main source of error is the deformation (compression) of the roller profile when it is loaded. Because the profile has a large cutout in its center, only two small strips of aluminum with a 43.5 mm^2 cross-sectional area are available to oppose the compressive load produced by the two linear springs. At maximal deflection, the linear springs apply 93 N of force primarily to the longer aluminum strip, creating a 2.145 MPa stress. The resultant displacement is 0.88μm. Although this is a small deflection, we believe it contributes to the linear term in the springs' force-length relationship because a previous iteration of this design, which used a Teflon roller profile, had a calculated deflection of 122.6μm and produced a linear term of -5.07 $\frac{\text{N}}{\text{cm}}$.

The effect of having these imperfections is limited. The quadratic term of the force-length relationship, a, was arbitrarily chosen to be a reasonable value. While more than 30% error exists in the two springs, the relative values are well matched (that is, they differ by only 1.1%) and their values are within a reasonable range. The presence of a non-zero linear term, b, results in a small, constant offset in the joint stiffness (demonstrated as the second term in Equation 11 and explained in Section 2.2). As with the quadratic term, absolute error in the constant offset term, c, is unimportant provided there is minimal relative difference in the values for the two springs. In this case, the relative difference is only 9.1% so the two terms approximately cancel each other out.

2.3.3 Other Approaches

Before choosing the final design of the nonlinear spring device, other implementation options were considered. We started by contacting spring manufacturers to discuss the feasibility of simply ordering coil springs with a custom force-length characteristic. All eight of the manufacturers we contacted either stated that they were not able to design springs to fit our requirements or, in one case, attempted to design the spring and ended up producing

springs that were approximately linear.

When it became clear that we were not able to rely on designs provided by outside manufacturers, we sought a method of modeling the dynamics of conical springs (*i.e.*, springs with coils of variable diameter). An attempt was made to find a closed-form equation for a spring in which the coil diameter followed a specific function (for example, a quadratic force-length relationship). However, we could not find sufficient information in the literature to properly model all of the forces in effect. Our next attempt was to design a conical spring using a piece-wise approach. Twenty segments per coil were used and spring parameters such as pitch, spring length and wire diameter were found by trial-and-error. In this iterative process, we varied the radius associated with each coil segment using a cubic function to produce the desired force-length relationship. Once the coil radii were defined, an increasing load was applied to the spring and deflection of each segment was examined. Two cases had to be considered for each segment's deflection. Because of the conical nature of the spring, some coil segments could simply be compressed within the next larger coil without contact. (Note that conical springs are typically designed in this manner so that a fully compressed spring is flat.) In the other case, the compressed coil segment could be compressed such that in contacts the next larger coil. This scenario occurred when the radius of one segment differed by less than the wire diameter from the radius of the segment 360^0 away. The coil profile for the best design that we could produce is shown in Figure 11 and the accompanying force-length relationship of the design is shown in Figure 12.

The design fits the ideal force-length curve well when the applied load is more than 3.11 N. The problem with this design was one of manufacturability. We were unable to find a spring manufacturer capable of producing a spring that met our coil radius specifications. Once we reached this point, we decided that our spring design could not rely on custom spring manufacturers, and we began work on the design described above that uses readily-available linear springs to produce the desired force-length relationship.

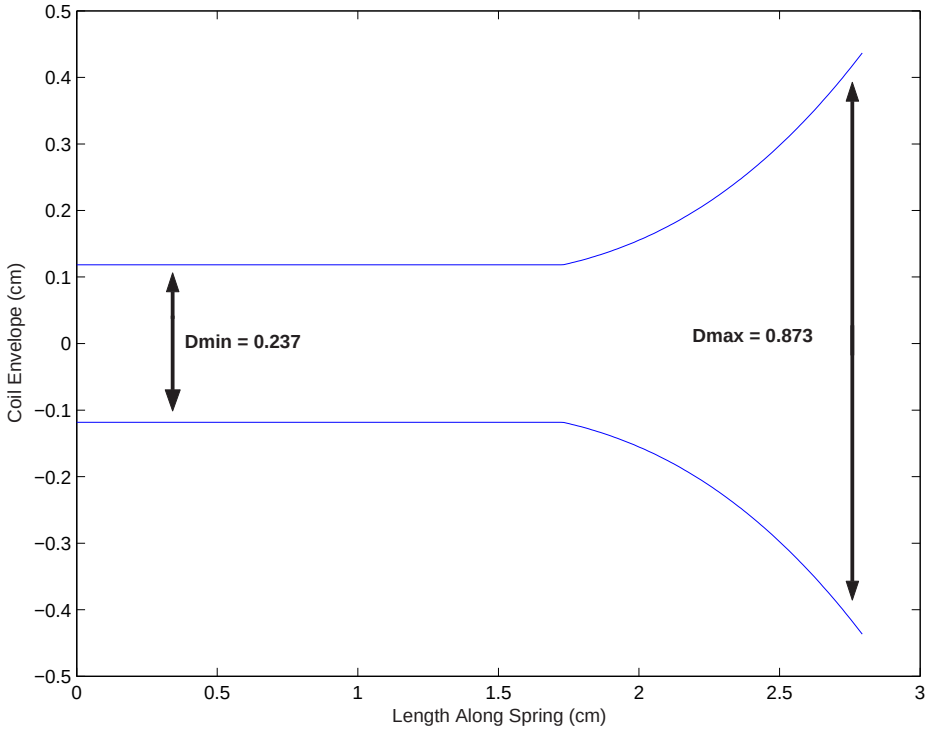

Figure 11: Coil envelope for the conical spring designed using a piece-wise approach.

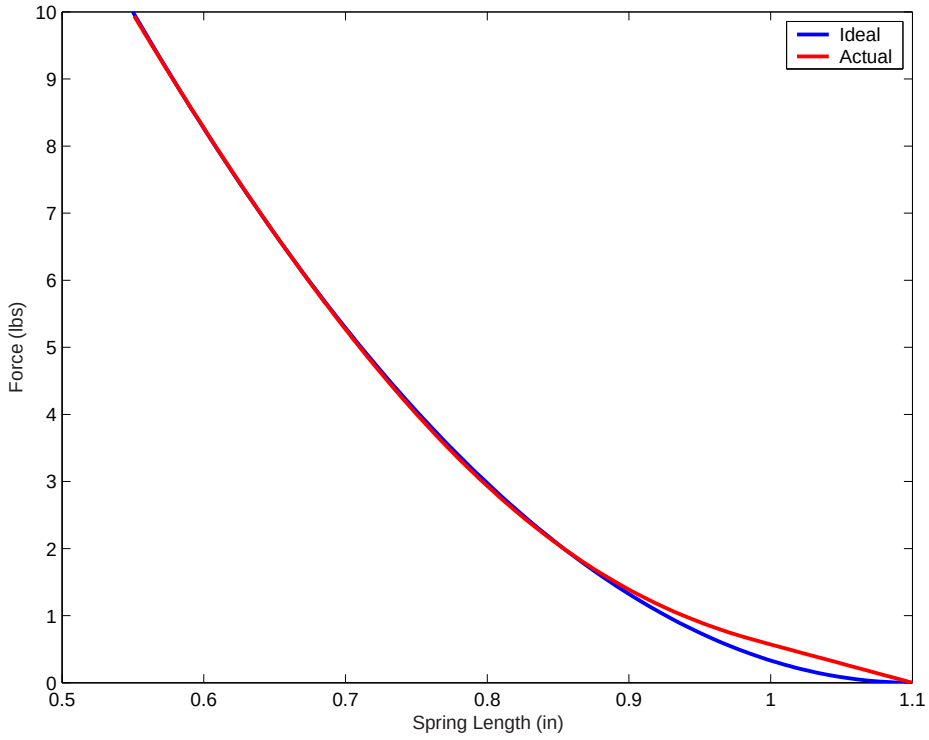

Figure 12: Ideal and simulated force-length relationship for the conical spring design.

2.4 Robotic Joint

This section describes the manufacturing of a single DOF robotic device that emulates a single, antagonistically controlled, rotational joint. The mechanical schematic for the device is shown in Figure 2. Our implementation of the actual device followed the schematic closely. A polycarbonate and stainless steel structure was built to hold in place the shaft of the rotating aluminum bar (the joint) and two servos capable of producing 1.27 Nm of stall torque (Model HS-5945MG, Hitec, Inc., Poway, CA). The servo horn of each servo was attached to the joint's shaft via the nonlinear springs described in Section 2.3. Connections to the springs were made using 1.14 mm diameter steel cable to prevent any significant contribution to the elasticity of the nonlinear spring. Joint angle was measured using an optical angle encoder (Model E2, US Digital Corp., Vancouver, WA). The completed device is shown in Figure 13.

2.5 Conclusions

We have designed and constructed a servo-controlled mechanical system that emulates the equilibrium point and stiffness control of a rotational joint using antagonistic control of movement. Series elastic actuation was implemented using nonlinear springs. It was shown that the use of springs with a quadratic force-length relationship provide an advantage over the use of linear springs because they provide a means of varying joint stiffness. Equations were derived that specify the angles required by the two servos to independently control joint equilibrium angle and stiffness.

A detailed process was developed for designing spring-like devices that can produce any smooth force-length function. These devices produce the desired force-length function by stretching linear springs in a nonlinear manner. Thus, by controlling the linear spring stretch function versus device lengthening, we can directly specify any smooth function.

Figure 13: The single degree of freedom device with antagonistic, series-elastic actuation.

CHAPTER III

SYSTEM CONTROL

This chapter focuses on the systems that were used to generate control signals for the robot. An overview of the electrical hardware is given, followed by explanations of how the control model presented in Chapter 2 was implemented. The chapter concludes with a description of the testing setup that was used to collect data.

3.1 Overview

The major goal of this research was to design and produce a system that could receive joint equilibrium angle and joint stiffness trajectories and could use this information to appropriately control a robot's servos using pulse-width modulated (PWM) signals. Because servos must be controlled with digital signals, a digital control architecture was chosen.

3.2 Field Programmable Gate Arrays

We choose to use a field-programmable gate array (FPGA) as the controller of the robotic system. An FPGA is an integrated circuit that contains huge arrays of digital gates that can be programmed to create a wide variety of digital circuits. When designing a circuit for the FPGA to implement, the user commonly only considers higher-level components such as a signal inverter rather than two interconnected transistors.

In the same manner, an FPGA's gates can be arranged to form a fully functional microprocessor, if desired. However, a major advantage of FPGA's is that they are not reliant on a processor to synchronize events. Rather, completely-parallel circuits can be designed whose computational speed is not reliant on the number of individual sub-circuits.

FPGA's are quickly reprogrammable using software tools such as MATLAB's Simulink (The Mathworks, Inc., Natick, MA) and do not require the user to have a mastery of low-level digital circuit design. FPGAs are currently available with more than 10M gates and

can be run at over clock speeds of more than 67MHz. Thus, these devices were able to provide more than enough size and speed to perform the necessary calculations for this project.

In selecting which type of digital system to use as a controller, we considered physical device size, computational speed, degree of parallelism, and design flexibility. A number of possible solutions were ruled out using these criteria:

- Computers - physically large, degree of parallelism limited by number of processors.
- DSP processors - lack parallelism.
- ASICs (application-specific integrated circuits) - one design takes several weeks to be fabricated and tested and, once manufactured, these devices cannot be reconfigured.

Because an FPGA is an integrated circuit, a support board is required to provide power, clock signals, all input/output, etc. We choose to use the Xtreme DSP Development Kit (Nallatech Ltd., Orlando, FL) for this purpose. This kit features one XC2V2000-4FG676 FPGA (Xilinx, Inc., San Jose, CA), two 14-bit ADC channels, two 14-bit DAC channels, two digital I/O bits, and 0.5MB of ZBT-SSRAM. A photograph of the kit is shown in Figure 14.

3.3 Equilibrium Point Control Model

As mentioned in Section 3.2, a major feature of FPGA's is their ability to be quickly reprogrammed. Typically, models are designed using VHDL code, which is a hardware description language that specifies which digital components should be instantiated on the FPGA and how the components should be interconnected. Xilinx offers an additional method of programming their devices. Rather than write code to specify the design, the user can create schematics in MATLAB's Simulink using a Xilinx toolbox named System Generator in which each block in the drawing corresponds to a user-level digital component on the FPGA. A simple example of a schematic that outputs the scaled version of the sum of two analog input signals is shown in Figure 15.

Figure 14: The Xtreme DSP Development Kit, an FPGA-based controller used for the system.

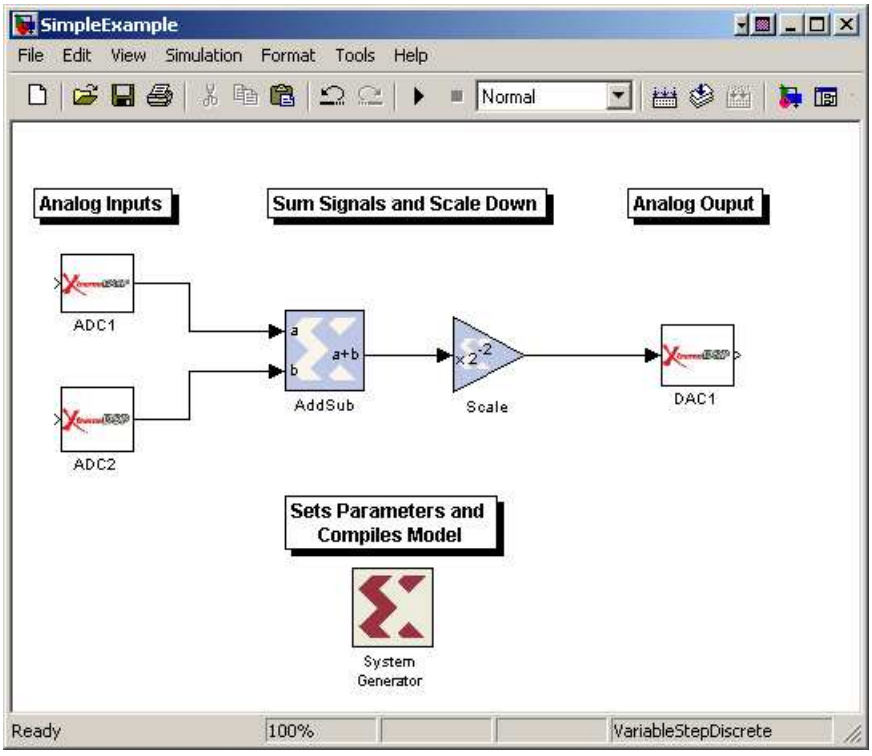

Figure 15: A simple Simulink model that outputs a scaled version of the sum of two input signals. Using the System Generator tool, the model can be compiled and implemented on a Xilinx FPGA.

Equations 14 and 15 give expressions for the servo angles, α and β, that must be set to yield an appropriate equilibrium joint angle and stiffness for a given external load. Thus, the role of the controller can be reduced to receiving the three input signals (θ, S, and *Load*), performing the calculations in Equations 14 and 15, and producing PWM output signals specifying α and β. Because of the limitation of only having two ADC's on the FPGA board, *Load* was specified as a parameter for each experiment, while θ and S were analog input signals. To simplify the calculations that needed to be performed on the FPGA, Equations 14 and 15 were rearranged and constant parameters were grouped into coefficients of the resulting polynomial. This is shown in Equations 32 and 33.

$$\alpha = -\overbrace{\left(\frac{R_J}{R_S}\right)}^{\epsilon_1}\left(\theta - \frac{Load}{S}\right) - \overbrace{\left(\frac{90^2}{a\pi^2 R_S R_J^2}\right)}^{\epsilon_2} S - \overbrace{\left(\frac{90b}{a\pi R_S}\right)}^{\epsilon_3} \tag{32}$$

$$\beta = \underbrace{\left(\frac{R_J}{R_S}\right)}_{\epsilon_1}\left(\theta - \frac{Load}{S}\right) - \underbrace{\left(\frac{90^2}{a\pi^2 R_S R_J^2}\right)}_{\epsilon_2} S - \underbrace{\left(\frac{90b}{a\pi R_S}\right)}_{\epsilon_3} \tag{33}$$

It can be seen that ϵ_1, ϵ_2, and ϵ_3 are derived solely from mechanical and mathematical constants. Therefore, these three coefficients were calculated *a priori* leaving simpler expressions for the FPGA to compute. Figure 16 shows a Simulink schematic that computes the expressions for α and β using ϵ_1, ϵ_2, ϵ_3, and *Load* as constants and θ and S as analog input signals.

While this circuit performs the computations necessary to determine the appropriate servo angles, other support circuitry must also be implemented on the FPGA so that the input and output signals are formatted properly for interfacing with external devices. For example, the signals sampled by the ADCs must be scaled so that their signal levels are in an appropriate range for use in calculations. Also, because servos require PWM control signals, extra circuitry must be added to convert α and β from DC signals to PWM signals before they are output through the DACs.

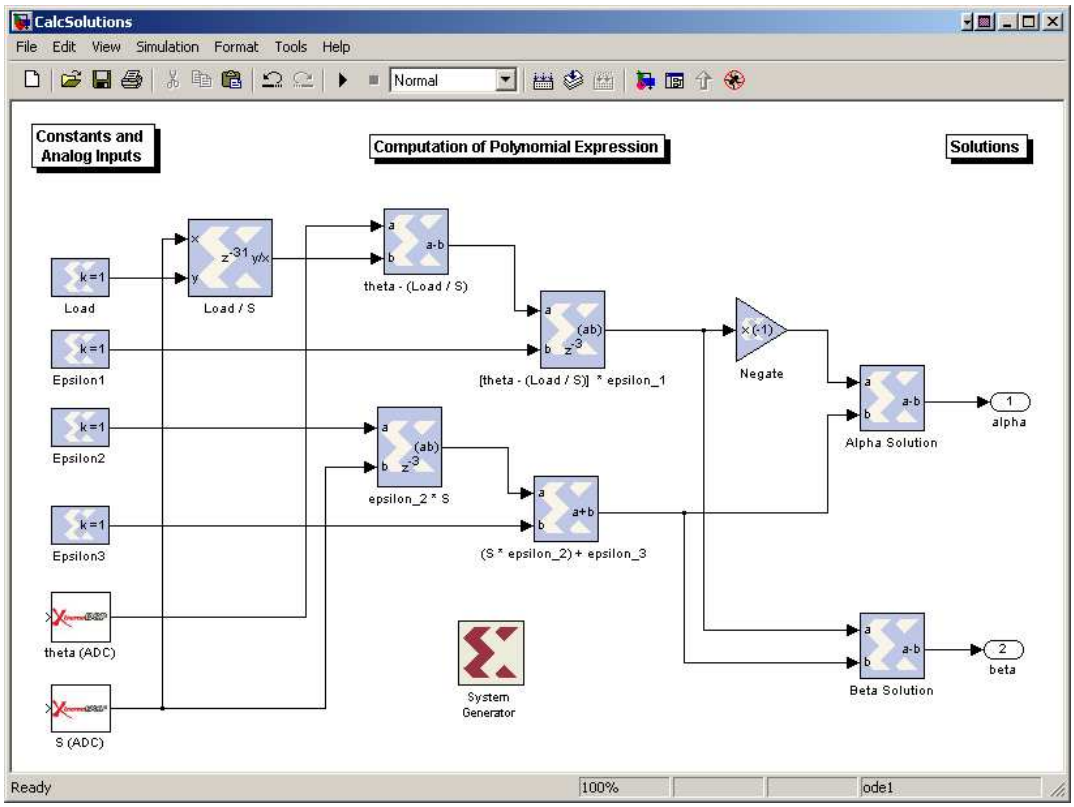

Figure 16: A Simulink model that computes the expressions for α and β.

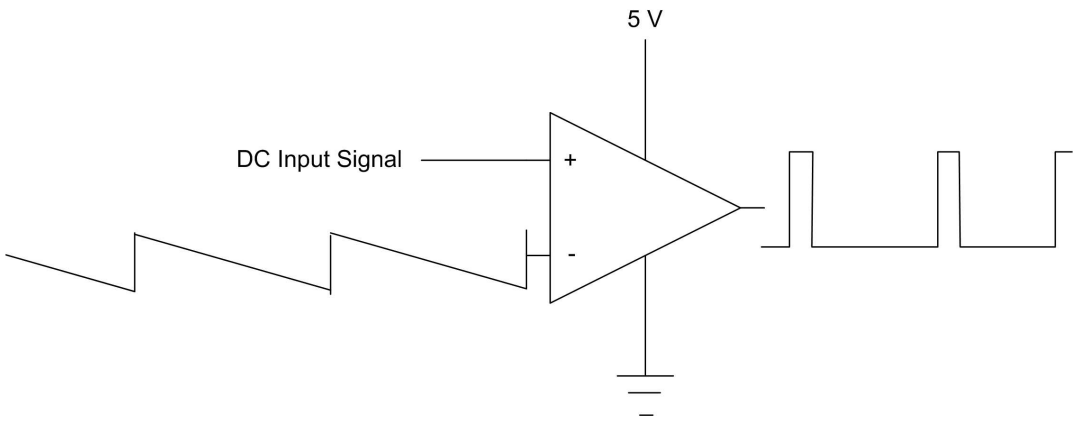

Figure 17: PWM conversion takes place by comparing the DC input signal to a repeating ramp signal. Whenever the ramp signal is lower in value, the output of the comparator goes high.

3.3.1 PWM Signal Generation

To convert DC output signals into PWM output signals a comparator circuit was designed to compare the DC signal with a repeating ramp signal. A comparator has a high output level when the non-inverting input terminal has a greater voltage applied to it than the negative input terminal. Figure 17 shows the general setup of the circuit. Note that the input signal is applied to the non-inverting input terminal. Therefore, as the input signal level increases, the width of the positive pulse produced by the comparator increases. This is shown in Figure 18.

To control the properties of the PWM signal, the ramp signal had to be carefully designed. The servos require a 5.0 V, 50 Hz PWM signal. There is a one-to-one relationship between the pulse width of the PWM signal they receive and the angles they produce. A 0.9ms pulse corresponds to -90^o, a 1.5 ms pulse corresponds to -0^o, and a 2.1 ms pulse corresponds to 90^o.

We set the clock frequency of the FPGA to 65.0MHz. To produce a PWM waveform with a frequency of 50Hz, the ramp signal's period was set to $1.30\text{x}10^6$ clock cycles. To produce the appropriate pulse width, the ramp signal was specified such that, for each period of the PWM signal, the servo input signal is greater than the ramp signal for at least 0.9ms (for a servo angle of -90^o) and no more than 2.1ms (for a servo angle of 90^o). As

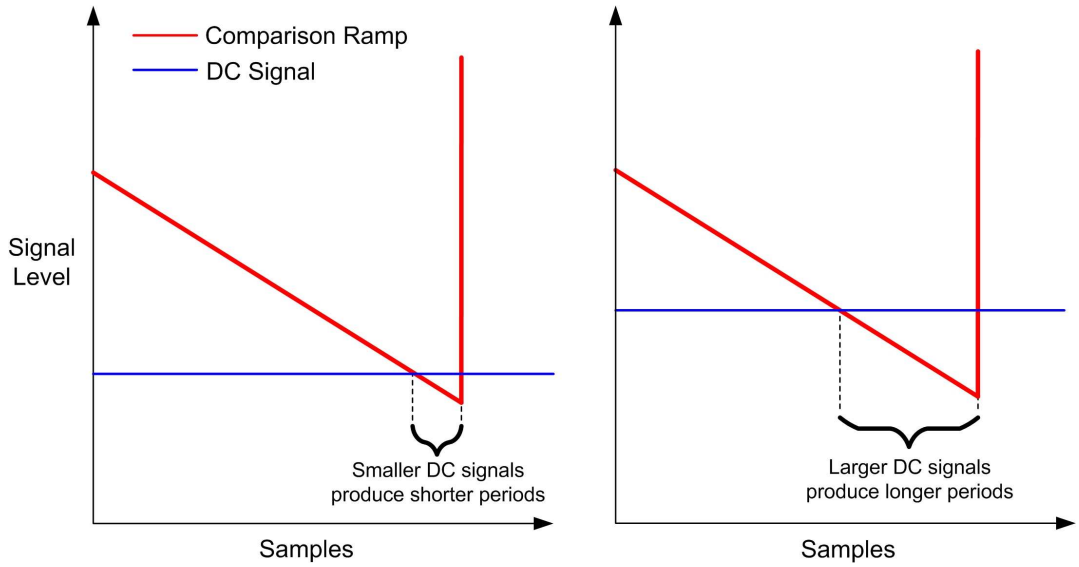

Figure 18: The effect of varying input signal value on the width of the PWM pulses.

shown in (Figure 19), this was accomplished by creating a ramp signal that varied between 1,202,500 and -97,500. To allow a comparison of the input and ramp signals, the ramp signal was then scaled by a factor of $\frac{90}{39{,}000}$ to produce a ramp signal that varied between 2775 and -225. Because the input signal was always between -90^o and 90^o, the PWM pulse width was always between 0.9ms (Point C) and 2.1ms (Point A).

In digital hardware this design was implemented using a counter, a scaling element, and two relational elements. The free-running counter began at 1,202,500 and decremented by one for each clock cycle. When the counter reached -97,500, a relational element was triggered to reset the counter back to 1,202,500. The value of the counter was fed through a scaling element that multiplied the counter value by $\frac{90}{39{,}000}$ to place it in the appropriate range. Finally, the relational operators compared the DC servo angle signals to the ramp signal and produced PWM signals for α and β. The Simulink model used for this sub-circuit is shown in Figure 20.

3.3.2 Complete Model

The final model implemented on the FPGA to control the system is shown in Figure 21. Note that calibration blocks for the two ADC's have been added and that the appearance

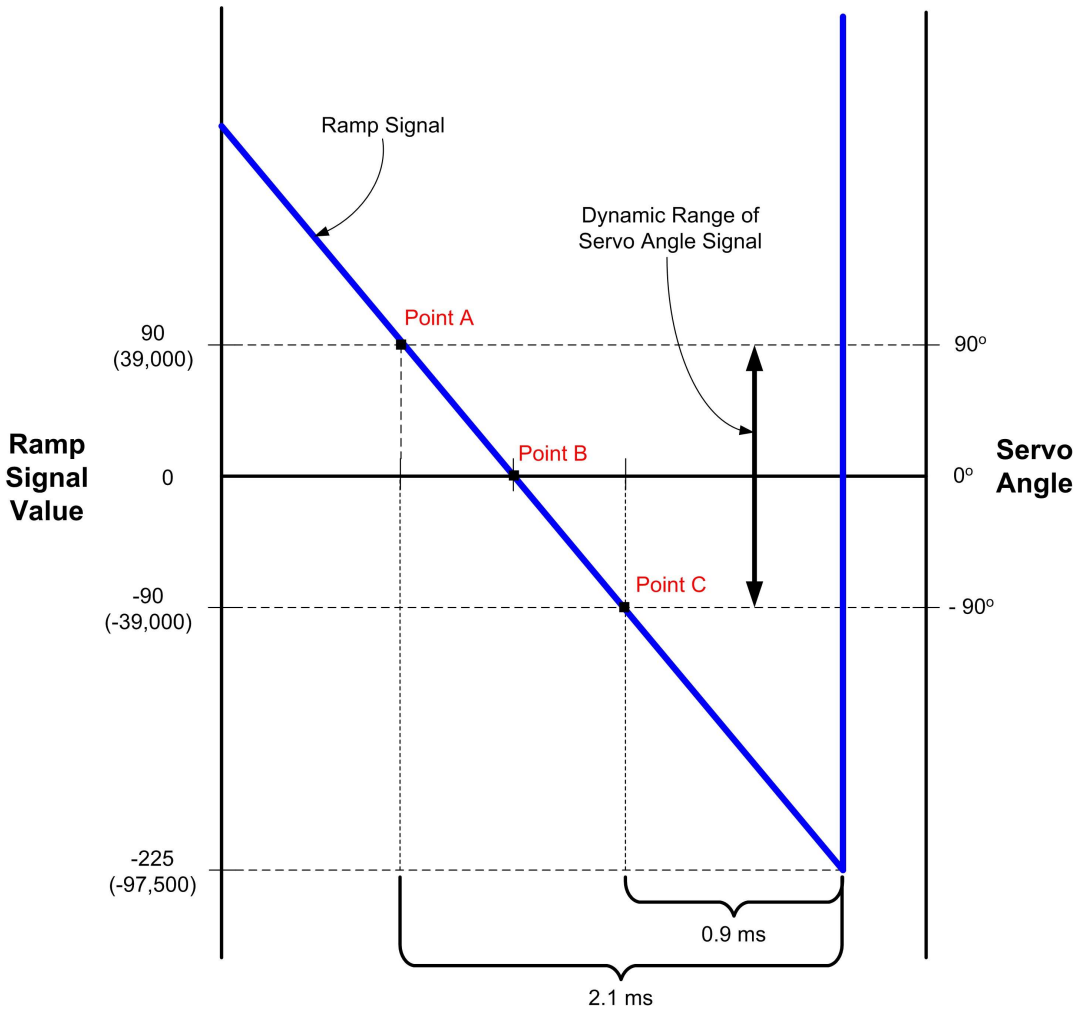

Figure 19: Expanded view of how scaling and shifting of the ramp signal was done to produce the desired PWM waveform. Note that the values for the ramp signal value are given in both scaled and unscaled (in parentheses) units.

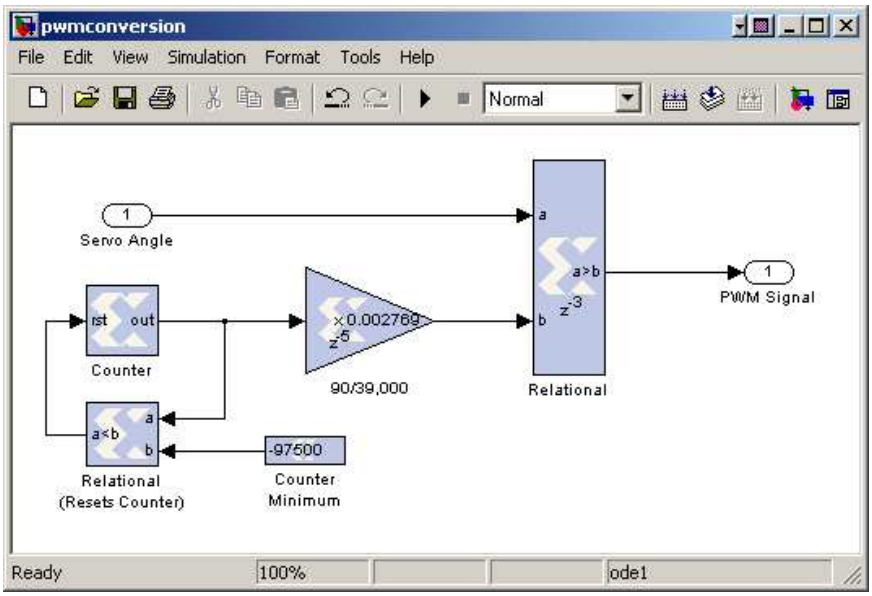

Figure 20: A Simulink model that converts DC servo angle signals into PWM signals appropriate for controlling servos.

has been simplified by grouping the many of the components shown in Figures 16 and 20 into subsystems named 'Calc Angles' and 'Ramp Signal Generator', respectively.

3.4 Testing Setup

To perform automated testing on the completed system, we used a dSpace controller board (dSpace, Inc, Novi, MI) to provide trajectories for joint equilibrium angle and stiffness while recording the actual joint angle. dSpace models were also designed using MATLAB's Simulink; the design used is shown in Figure 22. This model allows the user to select either a constant or a sinusoidal input signal using the switches. The signals are then formatted to be in an appropriate range for the DAC's and output to the FPGA. Additionally, the joint angle measured by the optical encoder on the robotic device is recorded. The full wiring schematic of the system, including the dSpace controller board and a simple interface circuit board is shown in Figure 23. Note that the interface circuit board is only used to organize wiring cables, route signals appropriately, and increase the capacitance of the +6V power supply. The board has no active devices and performs no calculations. Figure 24 shows the complete experimental setup including the robotic device and all testing equipment.

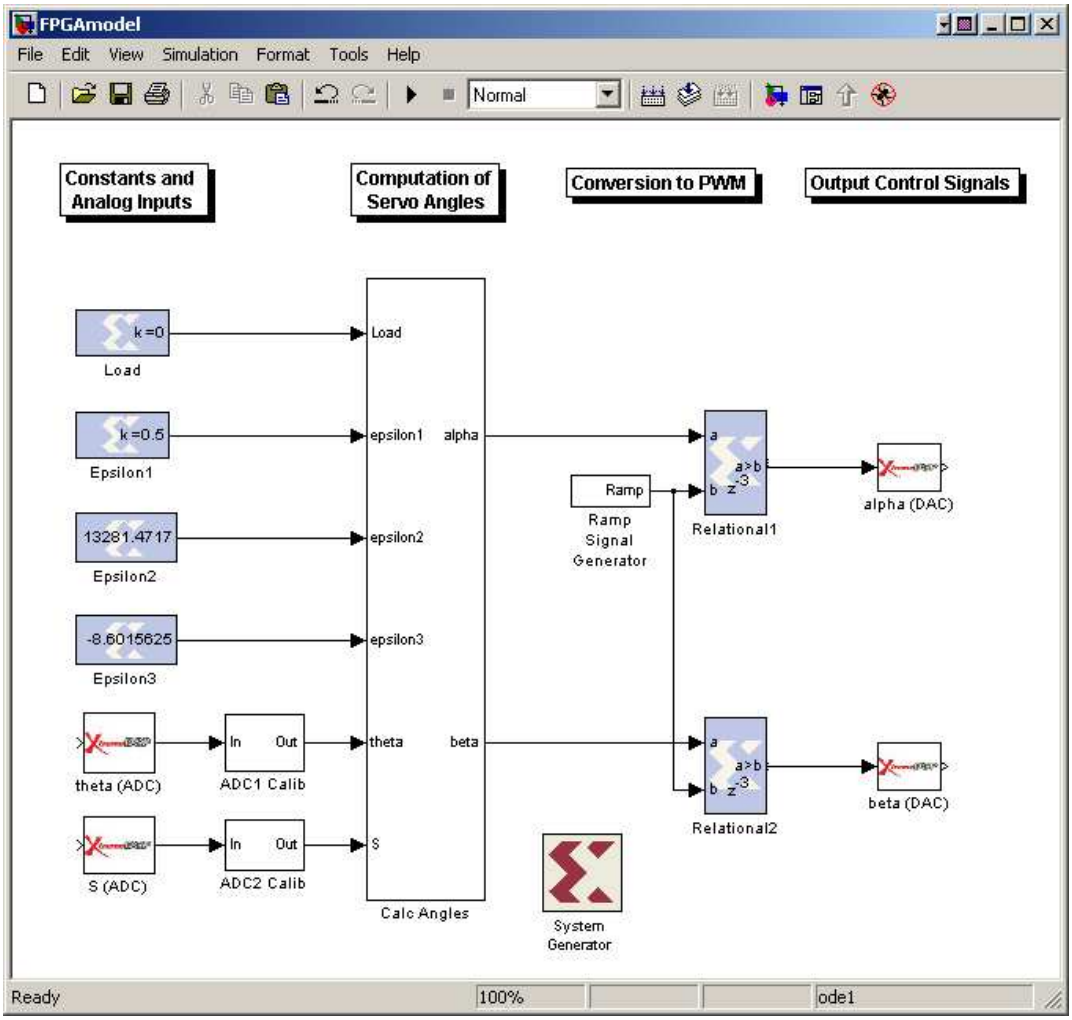

Figure 21: The Simulink model implemented on the FPGA to control the system. The 'Calc Angles' subsystem contains the model shown in Figure 16 and the 'Ramp Signal Generator' subsystem contains the model shown in Figure 20.

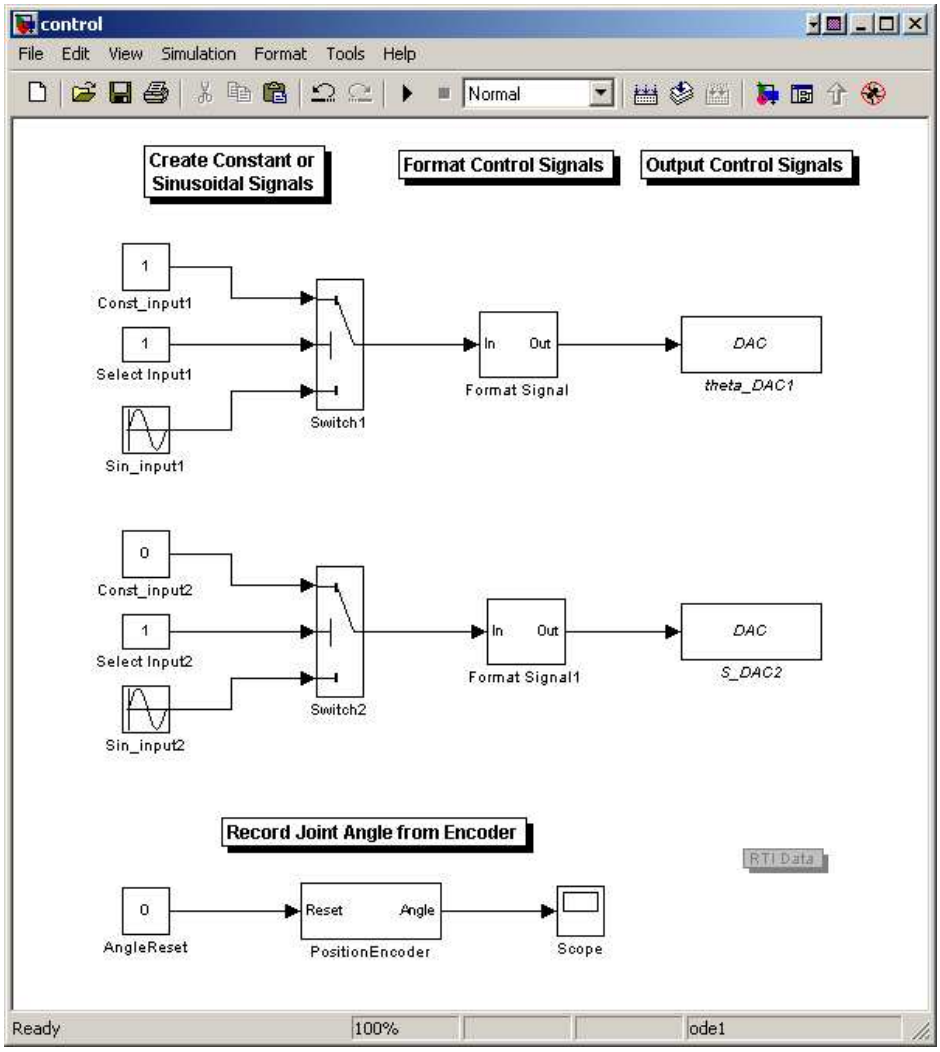

Figure 22: The dSpace model used to perform automated testing of the system.

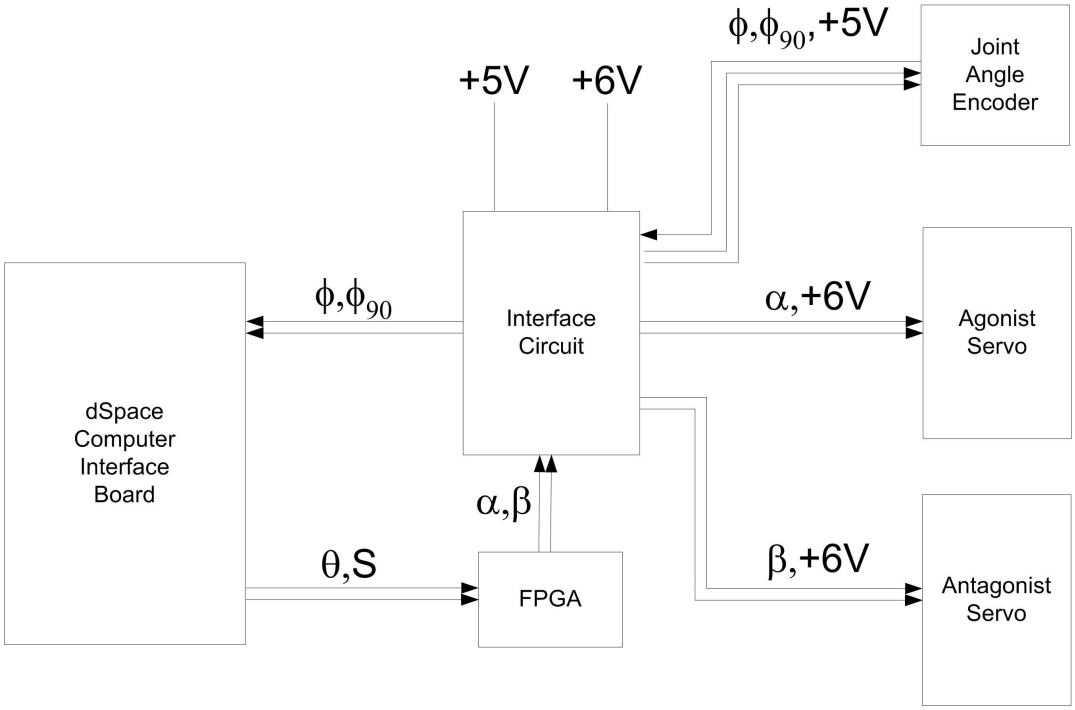

Figure 23: The electrical connectivity used for testing the final system, including the dSpace automated testing board.

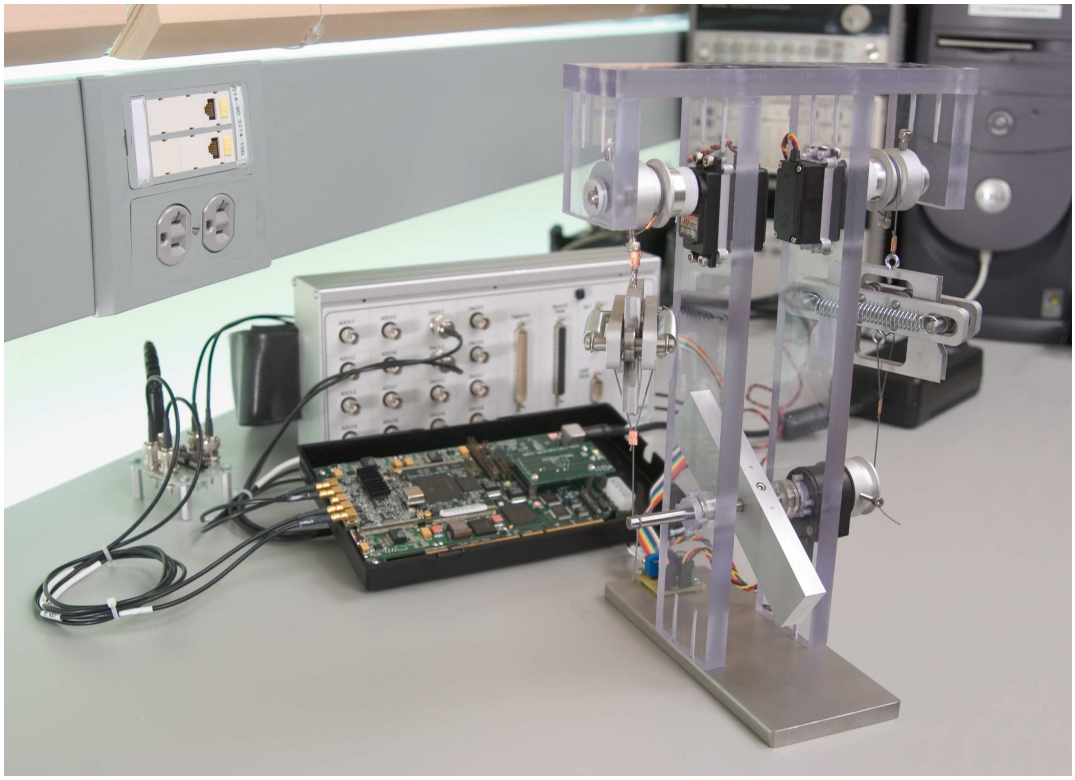

Figure 24: The complete experimental setup including the robotic device and all testing equipment.

3.5 Conclusions

We have implemented the control model for this architecture using an FPGA. This high-speed, digital device provides a means to compute the joint control equations in parallel. The model was programmed using a MATLAB toolbox named System Generator, in which high-level blocks (such as multipliers, ADC's and DAC's) were assembled as a wiring schematic. The model had two inputs—joint equilibrium angle and joint stiffness—and had two outputs—the two servo angles, α and β. Because servos must be driven with PWM signals, the output stage of the model contained DC signal to PWM signal converters. The entire architecture was tested using a dSpace computer interface board, which output joint equilibrium angle and joint stiffness signals to the FPGA and recorded the joint angle signal for analysis.

CHAPTER IV

IMPLEMENTATION OF JOINT CONTROL

This final chapter presents the results obtained from experiments with the robot. Validation of the theoretical model and its implementation in electrical and physical hardware is demonstrated using static and dynamic tests. We conclude with a brief discussion of the applications of this approach to robotic joint control.

4.1 Results and Validation of the Implementation

The primary goal of this research was to develop a robotic joint whose equilibrium angle and stiffness could be independently controlled. This was implemented by actuating a joint with a single, rotational degree of freedom with two strong servos that antagonistically pulled on the joint via nonlinear springs. The following subsections describe the series of experiments that were performed which validate the success of this approach. Throughout this discussion, the following variables are used:

- θ_c represents the commanded (desired) joint angle
- θ_m represents the measured joint angle
- S_c represents the commanded (desired) joint stiffness
- S_m represents the measured joint stiffness

4.1.1 Accuracy of Joint Actuation

The first test performed was an analysis of the accuracy with which the joint could be actuated. This was done by setting $\theta_c = 0^o$ and measuring θ_m when S_c was set to a series of values between -0.010 $\frac{\text{mNm}}{\text{deg}}$ and -0.700 $\frac{\text{mNm}}{\text{deg}}$. Ten trials were performed for each stiffness value. For each trial, the joint was randomly perturbed and allowed five seconds to settle

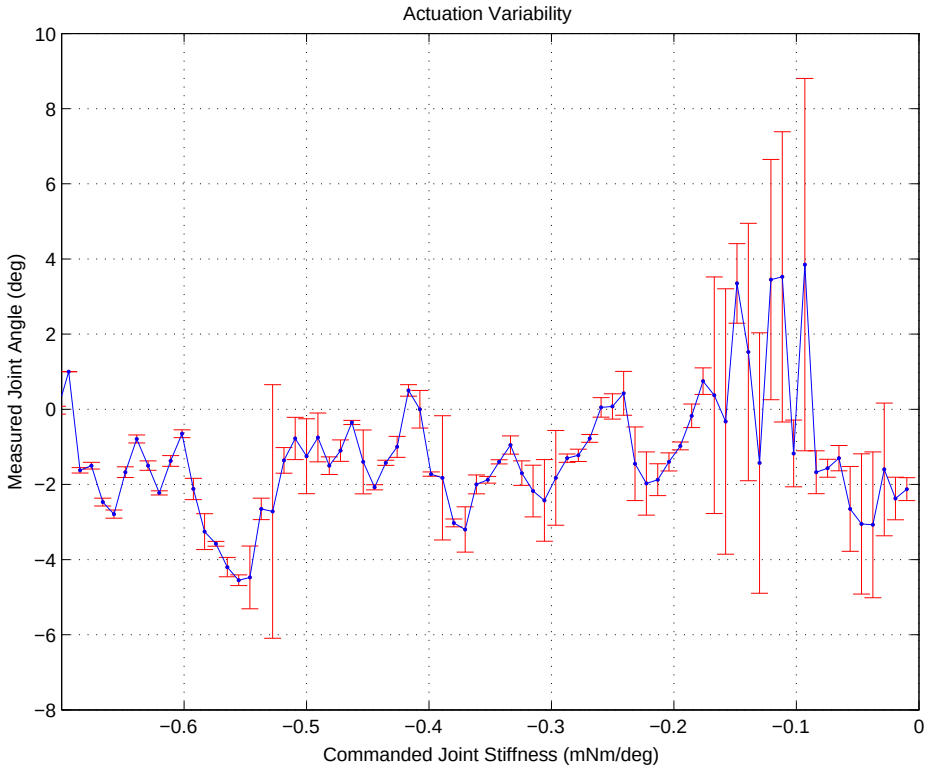

Figure 25: The accuracy with which the joint is actuated given a variety of stiffness values. Ten trials were performed for each stiffness value. For each trial, the joint was randomly perturbed and allowed five seconds to settle on its final value (settling actually only took approximately three seconds). Standard deviation for each stiffness level is shown as red error bars. It can be seen that as stiffness increases, the accuracy of the actuation also increases.

on its final value (settling actually only took approximately three seconds). The results of this experiment are shown in Figure 25.

This plot shows that the mean θ_m obtained for each stiffness ranges between -4.55^o and 3.85^o. The standard deviation of θ_m was 2.37^o for low-magnitude stiffness values ($|S_c| < 0.176\ \frac{\text{mNm}}{\text{deg}}$), but was reduced to 1.20^o for high-magnitude stiffness values. To determine the source of the errors, we performed a high-resolution test to determine if the variability was due to the presence of high-frequency noise. The test was performed for $-0.5\frac{\text{mNm}}{\text{deg}} < S_c < -0.4\frac{\text{mNm}}{\text{deg}}$. The resulting data is shown in Figure 26, and the Fourier transform of the data is shown in Figure 27. The data demonstrates that high-frequency noise does not contribute significantly to the error. Rather, the primary source of variability is the 18 $\frac{deg}{mNm}$ frequency component, which corresponds to the large peaks in Figure 26

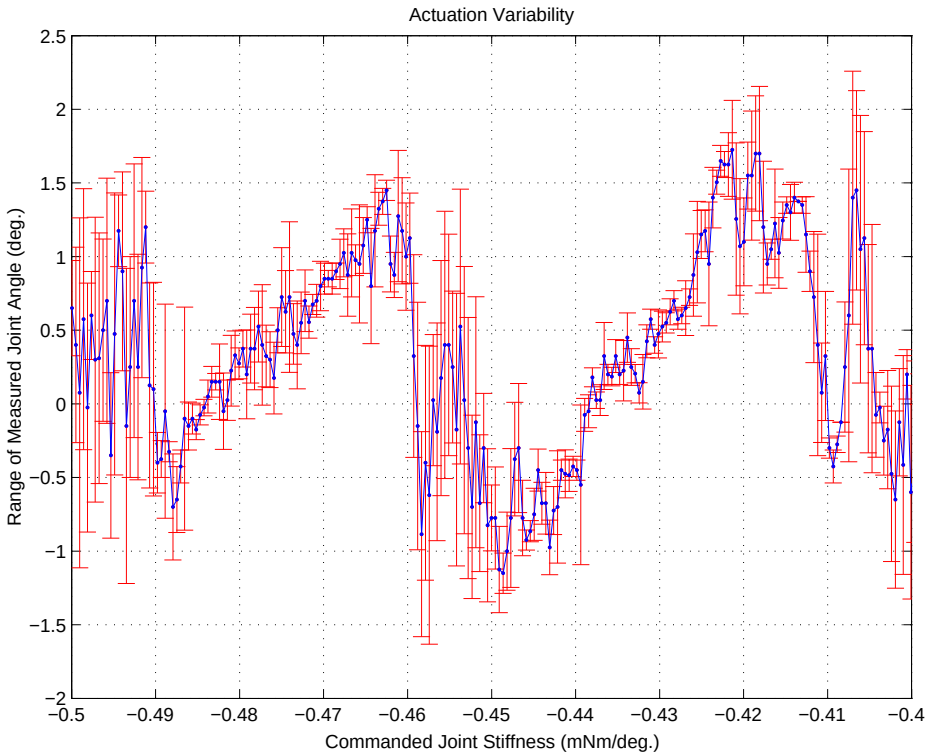

Figure 26: A high-resolution recording of measurement variability. As with the previous experiment, ten trials were performed for each stiffness value and settling was allowed to occur before recordings were taken. Standard deviation for each stiffness level is shown as red error bars.

that have a 0.055 $\frac{mNm}{deg}$ period.

The actuation variability can be explained for both the low and high magnitude stiffness regions. When stiffness is small in magnitude, the antagonistic forces used to drive the joint to θ_c are low enough that frictional forces and the bumps on the roller profile (created during its machining) can significantly hinder the joint from reaching θ_c. When stiffness increases in magnitude, the driving forces are able to overcome frictional forces, and thus reduce trial-to-trial variability. However, the joint still misses θ_c by a few degrees because the bumps create local energy minima that consistently attract the rollers to rest in certain positions. The effects of individual bumps can be seen as the large-magnitude changes in θ_m that occur throughout Figures 25 and 26. Note that a bump on one spring causes a positive deflection in θ_m, while a bump on the other spring causes a negative deflection; both positive and negative deflections can be seen in these Figures. We expect that a higher

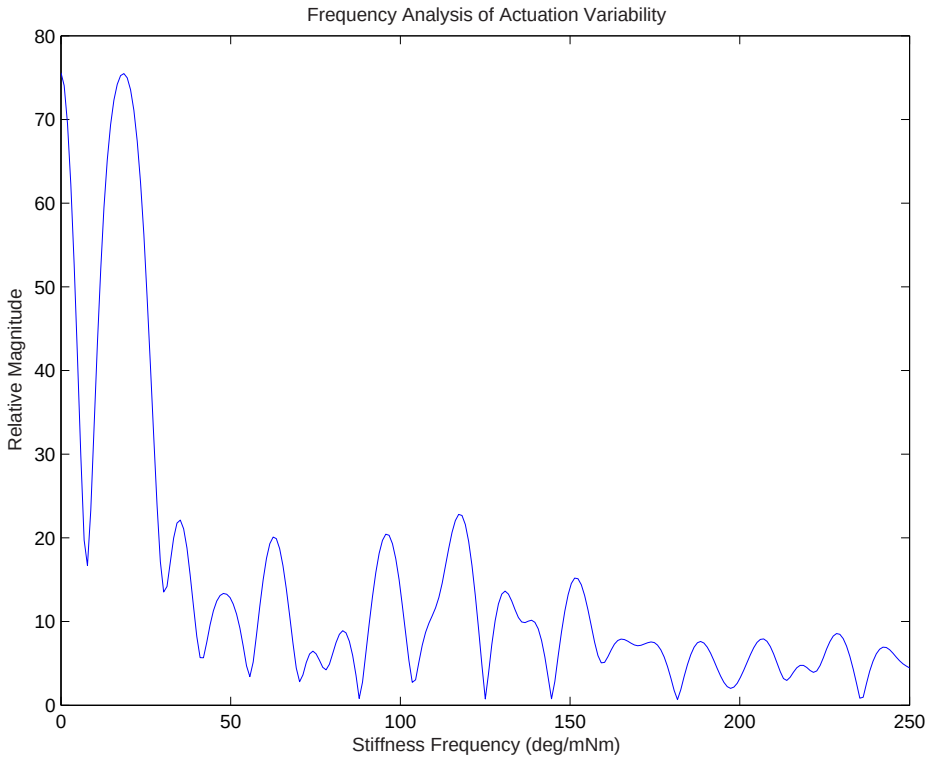

Figure 27: Frequency analysis of the data from Figure 26 showing that the primary frequency component of the error in the joint actuation is 18 $\frac{deg}{mNm}$.

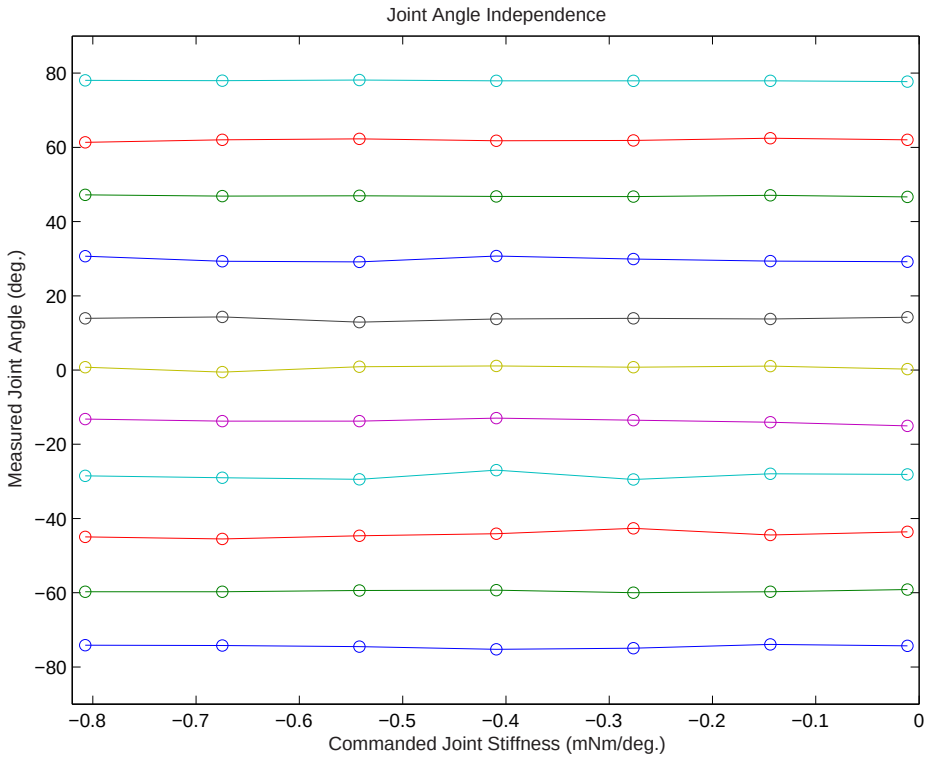

Figure 28: The independence of joint equilibrium angle over a full range of stiffnesses. In this plot, each trace represents represents one value of θ_c.

precision machining of the roller profiles would largely remove the bumps and thus reduce the variability in θ_c.

4.1.2 Independence of Joint Equilibrium Angle and Joint Stiffness

To demonstrate that the joint's equilibrium angle can be specified independent of the joint's stiffness, an experiment was performed in which θ_m was measured while θ_c and S_c were swept from -75^o through 75^o and from -0.011 $\frac{\text{mNm}}{\text{deg}}$ through -0.807 $\frac{\text{mNm}}{\text{deg}}$, respectively. The results are shown in Figure 28.

Each trace in this figure represents one value of θ_c. Ideally, each trace would be perfectly flat, demonstrating that variations in stiffness have no effect on joint angle. In the experiment, variability is seen in the same range as the variability inherent in the system (θ_m was always within 3.0^o of θ_c and the standard deviation of θ_m was 1.25^o).

The next step was to verify that the stiffness of the joint was controllable and that S_C and S_m were well correlated. Because stiffness is defined as the joint's resistance to angular

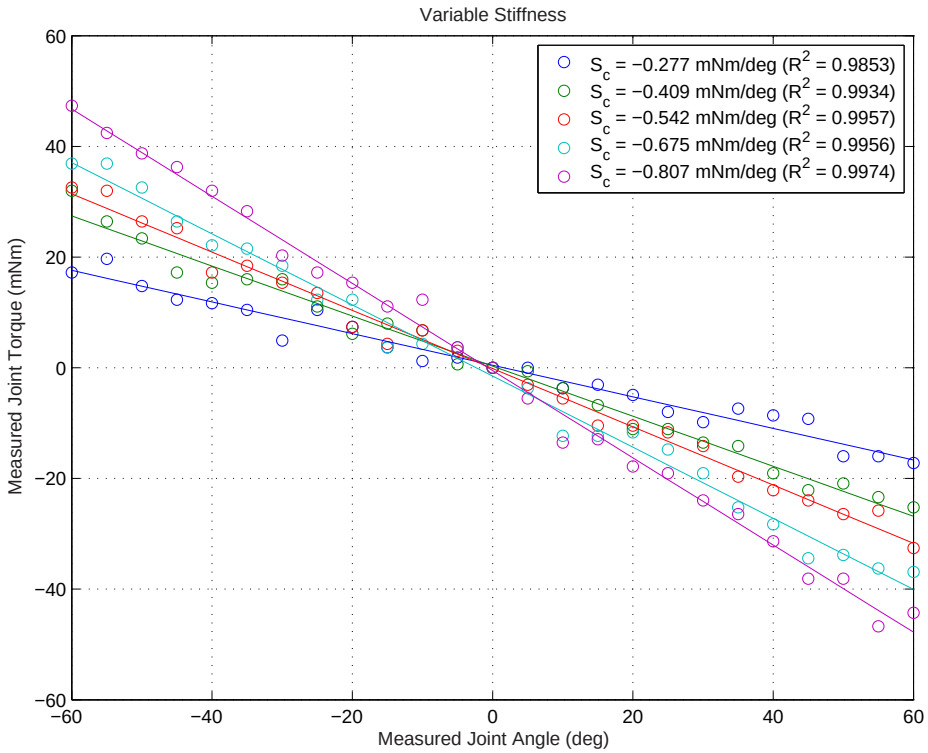

Figure 29: Variations in joint stiffness produced at a given joint equilibrium angle. External torques were applied to the joint until a set joint angle was achieved. The corresponding angle-torque pairs were then plotted as the circles. The traces represent a linear regression through each set of points. The R^2 value for each regression is shown in the legend. Note that as S_c increases, the slope of the torque-angle line (S_m) also increases in magnitude.

displacement, or $\frac{dT}{d\theta}$, we obtained values for S_m by measuring the static torque that needed to be applied to the joint to produce a series of θ_m values within $\pm 60^o$ of θ_c. Figure 29 shows the data collected for five equally spaced joint stiffnesses when $\theta_c = 0^o$.

To avoid the large effects of frictional forces and to prevent the actuating cables from going slack, the minimum value for S_c was set to -0.277 $\frac{\text{mNm}}{\text{deg}}$. It can be seen that for any of the five values of S_c, the relationship between measured joint torque and θ_m is linear. The mean correlation coefficient for the linear fits was 0.994. Moreover, as S_c increases, the slope of the torque-angle line (S_m) also increases in magnitude. To demonstrate that this behavior was independent of θ_c, this experiment was repeated for $\theta_c = \pm 45^o$. The torque-angle lines for each of these experiments are shown in Figure 30.

Figure 31 correlates S_c and S_m by plotting the slopes from Figure 30 versus S_c. Error bars for this plot were determined by performing a Monte Carlo simulation with the

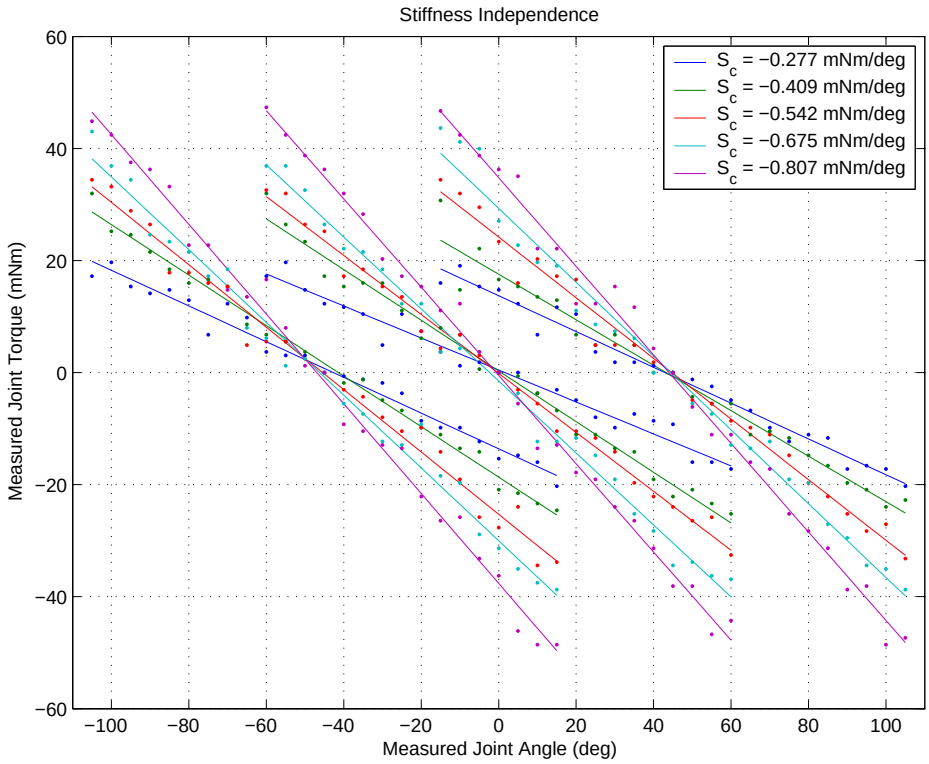

Figure 30: The independence of joint stiffness over three equilibrium angles. This plot shows the results of repeating the experiment shown in Figure 29 for $\theta_c = \pm 45^o$. Note that the torque-angle lines show minimal variability as θ_c is changed.

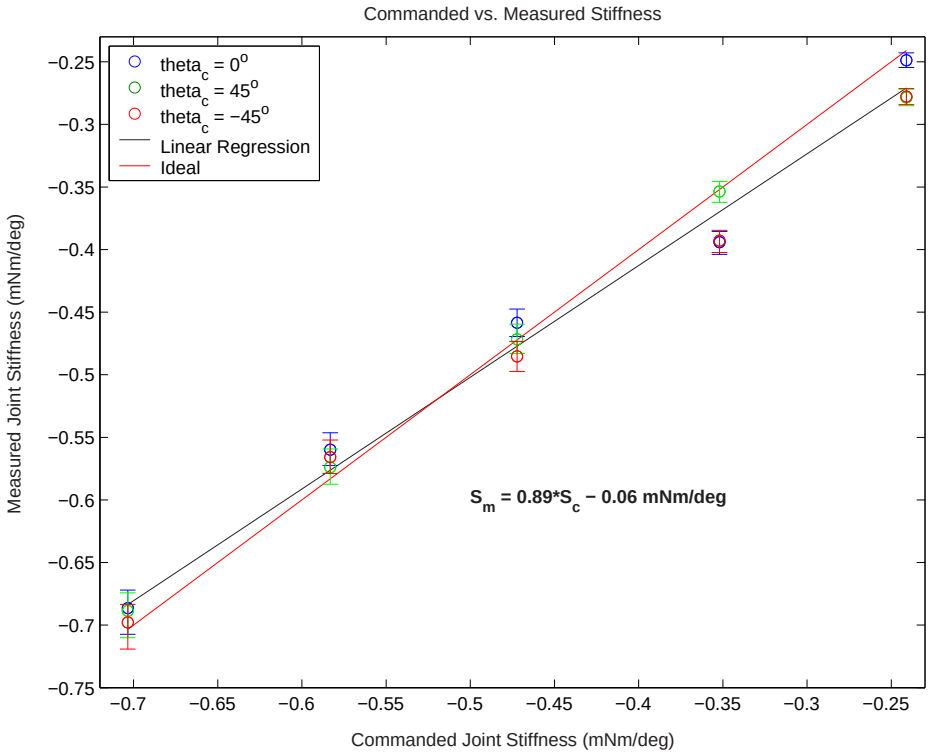

Figure 31: The relationship between commanded and measured joint stiffness showing a high correlation ($R^2 = 0.9998$) between the ideal curve and the actual stiffness values achieved. Error bars for this plot were determined by performing a Monte Carlo simulation with the error statistics taken from Section 4.1.1.

error statistics taken from Section 4.1.1. The black trace in this figure corresponds to the linear regression through these points and therefore defines the relationship between the commanded joint stiffness, S_c, and the actual measured joint stiffness, S_m. The equation for this relationship is:

$$S_m = 0.89 S_c - 0.06 \frac{\text{mNm}}{\text{deg}} \tag{34}$$

The red trace corresponds to the ideal S_c vs. S_m relationship. The similarity of the black and red traces verifies that the stiffness of the joint was controllable independent of θ_c and that S_C and S_m were well correlated. In fact, the data points fit the ideal trace with a correlation coefficient of 0.998.

The range of stiffness achievable by the joint corresponds well to the range produced in the human knee when scaled by the difference in inertia. Human knee stiffness has been shown to range from 0.700 $\frac{\text{mNm}}{\text{deg}}$ to 2.09 $\frac{\text{mNm}}{\text{deg}}$ and lower leg inertia has been estimated to be

0.437 kg m^2 [23], [19]. The inertia of the robotic limb was calculated to be 1.87x10^{-4} kg m^2, which produces a scaling factor of 2337. When the stiffness values produced by the robotic joint were scaled by this factor, the resulting stiffness values ranged between 0.467 $\frac{\text{mNm}}{\text{deg}}$ and 1.869 $\frac{\text{mNm}}{\text{deg}}$. Therefore, biologically achievable stiffness values overlap significantly with those achievable with the robotic device.

4.1.3 Compensation for Applied Loads

In the experiments performed up to this point, no loads were externally applied, and therefore the *Load* parameter of the control algorithm was set to 0.0 $\frac{\text{mNm}}{\text{deg}}$. All external torques used to characterize the joint's stiffness were assumed to be perturbations. However, in the normal use of a robotic joint, forces are applied for which compensation must be applied to maintain the desired joint equilibrium angle and stiffness. In this experiment, the effect of a non-zero external torque on θ_m is measured with and without appropriate compensation.

In the first part of the experiment, $\theta_c = 0^o$ and $Load = 0.00$ $\frac{\text{mNm}}{\text{deg}}$. An external torque equal to either -11.3 $\frac{\text{mNm}}{\text{deg}}$, -22.6 $\frac{\text{mNm}}{\text{deg}}$, or -33.9 $\frac{\text{mNm}}{\text{deg}}$ was then physically applied to the joint and θ_m was recorded for range of stiffness values. This data is displayed in Figure 32 as the bottom three traces. These traces show that when no compensation is used by the control algorithm, the joint's equilibrium angle deviates significantly from its target angle of 0^o.

The top three traces show the same experiment when compensation was applied by setting the variable *Load* equal to the value of the external torque. For large magnitude stiffnesses ($S_c < -0.28\frac{\text{mNm}}{\text{deg}}$), the compensation brings the mean joint angle to -0.66^o with a standard deviation of 0.93^o. Note that the compensated traces increase in value for $S_c > -0.28\frac{\text{mNm}}{\text{deg}}$. This occurs because one of the actuation cables had gone slack. In other words, θ had left its operating region, which was defined in Section 2.2 to be:

$$-\alpha \leq \theta \leq \beta \tag{35}$$

The gradual increase in value of the bottom three traces as S_c increases in magnitude occurs because the uncompensated external torque caused joint angle and joint stiffness signals to lose their uniqueness. A detailed explanation of this effect was given in Section

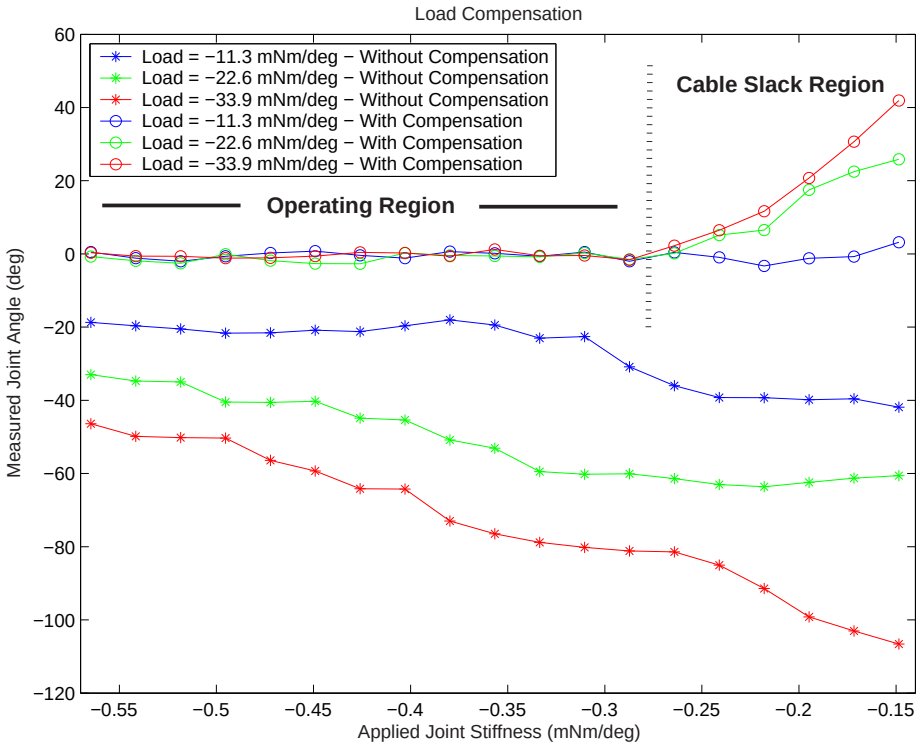

Figure 32: Compensation for an externally applied joint torque. The bottom three traces occurred when the algorithm did not expect an external load to be applied. The bottom three traces occurred when the proper compensation was used for each of the external loads.

1.2.1. To summarize, when an external torque is applied to a joint without compensation, any increase in stiffness magnitude (*i.e.*, co-contraction) causes the joint's equilibrium angle (shown as EP_1 and EP_2 in Figure 1) to move nearer its target angle. This behavior only occurs because compensation was not applied; it does not play a factor once *Load* is specified properly.

4.2 Conclusions

We have tested this architecture's static performance to validate its mechanical design. First, we showed that joint angle actuation was accurate (the standard deviation was 1.20^o), provided the stiffness magnitude was at least $0.176\frac{\text{mNm}}{\text{deg}}$. The error that was produced was primarily due to imperfections in the machining of the nonlinear spring components. Next, we demonstrated that varying stiffness does not significantly affect the equilibrium joint angle. For all eleven angles tested the deviation over the entire range of stiffnesses was less than $\pm 3.0^o$. Stiffness independence was demonstrated by measuring the deviation in joint stiffness that occurred as joint equilibrium angle was varied. The maximum stiffness deviation that occurred in this experiment was $0.041\frac{\text{mNm}}{\text{deg}}$. Stiffness actuation accuracy was then demonstrated by plotting the recorded joint stiffnesses versus the commanded joint stiffnesses. A correlation coefficient of 0.998 was achieved between the data points and the ideal linear trace. Finally, the ability of the system to compensate for externally applied torques was demonstrated by verifying that a target angle could be consistently reached despite the presence of the external torques, provided that the control algorithm received the values of the external torques as input.

4.3 Contribution of this Architecture to Robot Control

The major contribution of this work was the design, implementation, and validation of a robotic joint with independent control of equilibrium angle and stiffness. The control scheme was devised to emulate the λ-version of the Equilibrium Point Hypothesis of animal motor control in an attempt to produce a robotic system with biologically realistic joint dynamics. Control of the actuating servos was provided by an FPGA, which received desired joint

angle and stiffness as inputs and produced PWM signals needed to control the actuating servos as outputs. The system was validated by demonstrating:

- the independence of joint equilibrium angle and stiffness
- the high correlation of between commanded and measured values of joint equilibrium angle and joint stiffness
- the ability of the control scheme to compensate for applied loads

Because many existing robotic systems currently use servos to actuate their joints, this architecture may be applied without major redesigns. Each joint would need to have two dedicated servos connected to it via a series connection of non-elastic cables and quadratic springs.

While this version of the architecture proved to be successful, there exist variations that can improve the its overall performance. Because the bumps on the nonlinear springs presented complications during testing, we believe that careful re-machining of these parts may produce devices with improved performance. Furthermore, this particular implementation of quadratic springs may be replaced with any implementation that reliably produces a quadratic force-length relationship. (For example, we considered, but have not yet implemented, a design that involves cams attached to both servos and to the joint, which are connected via a series connection of cables and linear springs.) By offering flexibility in the mechanical implementation, and a straightforward control algorithm, this architecture provides the basis for a biologically inspired method of controlling robotic joints.

4.4 Future Direction

This work was the first step in applying the principles of the Equilibrium Point Hypothesis to robotic joint control. There are still limitations of the architecture in its current form that must be addressed before it can be applied to dynamic systems. For example, damping was not used in the current system because all test were performed at steady-state. However, dynamic testing requires that damping to avoid inappropriate joint oscillations. Once it is included, we can perform several experiments, including those that follow.

Pointing Recreate the trajectories for equilibrium angle and stiffness that have been empirically determined from human experiments.

Postural Stability Analyze the effect of joint stiffness on postural stability during perturbations from the environment.

Dynamic Perturbation Analyze the effect of environmental perturbations on the stability of a rhythmically moving joint, such as a leg being controlled by a neural central pattern generator.

The next step will be to apply this architecture to a multi-joint limb. This will introduce more complexity and more rich dynamics. Of primary interest will be the modeling and performance of multi-articular muscles. There will also be issues concerning the dynamics of the limb's end-point versus the dynamics of each individual joint. While these considerations will significantly complicate robot control, our solutions may also bring us closer to understanding the solutions biology has found.

REFERENCES

[1] Bellomo, A. and Inbar, G., "Examination of the λ equilibrium point hypothesis when applies to single degree of freedom movements performed with differential inertial loads," *Biol. Cybern.*, vol. 76, pp. 63–72, 1997.

[2] Caux, S. and Zapata, R., "Modeling and control of biped robot dynamics," *Robotica*, vol. 17, pp. 413–426, 1999.

[3] Contreras-Vidal, J.L., Grossberg, S. and Bullock, D., "A neural model of cerebellar learning for arm movement control: Cortico-spino-cerebellar dynamics," *Learning & Memory*, vol. 3, pp. 475–502, 1997.

[4] English, C.E. and Russell, D., "Mechanics and stiffness limitations of a variable stiffness actuator for use in prosthetic limbs," *Mechanism and Machine Theory*, vol. 34, pp. 7–25, 1999.

[5] Feldman, A.G., "Superposition of motor programs. ii. rapid flexion of forearm in man," *Neuroscience*, vol. 5, pp. 91–95, 1980.

[6] Feldman, A.G., "Once more on the equilibrium point hypothesis (λ-model) for motor control," *Journal of Motor Behavior*, vol. 18, pp. 17–54, 1986.

[7] Furuta, T., Tawara, T., Okumura, Y., Shimizu, M. and Tomiyama, K., "Design and construction of a series of compact humanoid robots and development of biped walk control strategies," *Robotics and Autonomous Systems*, pp. 81–100, 2001.

[8] Gottlieb, G. L., "Rejecting the equilibrium-point hypothesis," *Motor Control*, vol. 2, pp. 10–12, 1992.

[9] Joyce, G.C., Rack, P.M.H. and Westbury, D.R., "The mechanical properties of cat soleus muscle during controlled lengthening and shortenting movements," *J. Physiol.*, vol. 204, pp. 461–474, 1969.

[10] Kirsch, R.F., Boskov, D. and Rymer, W.Z., "Muscle stiffness during transient and continuous movments of cat muscle: Perturbation characteristics and physiological relevence," *IEEE Transactions on Biomedical Engineering*, vol. 41, no. 8, pp. 758–770, 1994.

[11] Kolacinski, R.M. and Quinn, R.D., "A novel biomimetric actuator system," *Robotics and Autonomous Systems*, pp. 1–18, 1998.

[12] Latash, M. and Gottlieb, G., "Recontruction of shifting elbow joint compliant characteristics during fast and slow movements," *Neuroscience*, vol. 43, pp. 697–712, 1991.

[13] McGeer, T., "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[14] Nichols, T.R. and Houk, J.C., "Improvement in linearity and regulation of stiffness that results from actions of stretch reflex," *J. Neurophysiol.*, vol. 39, no. 1, pp. 119–142, 1976.

[15] Ortega, R., Spong, M. and Gómez-Estern, F., "Stabilization of underactuated mechanical systems via interconnection and damping assignment," *IEEE Trans. Aut. Control*, vol. 47, no. 8, pp. 1281–1233, 2002.

[16] Pratt, G.A. and Williamson, M.M., "Series elastic actuators," *Proceedings of IROS*, 1994. Pittsburg, PA.

[17] Pratt, J.E. and Pratt, G.A., "Exploiting natural dynamics in the control of a 3d bipedal walking simulation," *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99)*, 1999. Portsmouth, UK.

[18] Raibert, M.H., *Legged Robots the Balance.* MIT Press Cambridge, MA, 1986.

[19] Tai, C. and Robinson, C.J., "Variation of human knee stiffness with angular pertubation intensity," *Proceedings of IEEE Engineering in Medicine & Biology 17th Annual Conference and 21st Can. Med. & Biol. Eng. Conference (CD-ROM, 2 pages)*, 1995. Montreal.

[20] Tresilian, J.R., "Retaining the equilibrium-point hypothesis as an abstract description of the neuromuscular system," *Motor Control*, vol. 3, pp. 67–89, 1999.

[21] Weidemann, H-J, Pfeiffer, F. and Eltze, J., "A design concept for legged robots derived from the walking stick insect," *Proceedings of the 1993, International Conference on Intelligent Robots and Systems*, pp. 545–550, 1993.

[22] Wisse, M., Schwab, A.L. and Linde, R.Q., "A 3d passive dynamic biped with yaw and roll compensation," *Robotica*, vol. 19, pp. 275–284, 2001.

[23] Zhang, L., Nuber, G., Butler, J., Bowen, M. and Rymer, W.Z., "*In vivo* human knee joint dynamic properties as functions of muscle contraction and joint position," *Journal of Biomechanics*, vol. 31, pp. 71–76, 1998.