

# Explorations in Low-Cost Compliant Robotics

by

Adam Kumpf

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

© Massachusetts Institute of Technology 2007. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
February 2, 2007

Certified by .....  
Rodney Brooks  
Panasonic Professor of Robotics, EECS  
Thesis Supervisor

Accepted by .....  
Arthur C. Smith  
Chairman, Department Committee on Graduate Students



# Explorations in Low-Cost Compliant Robotics

by

Adam Kumpf

Submitted to the Department of Electrical Engineering and Computer Science  
on February 2, 2007, in partial fulfillment of the  
requirements for the degree of  
Master of Engineering in Electrical Engineering and Computer Science

## Abstract

This thesis presents the findings of exploratory research in low-cost compliant robotics. The most heavily leveraged trade-off is that of mechanical precision for computational power, with the hope that the price of future computation will continue to fall exponentially while the expected price of precision mechanical parts will remain relatively constant. The most novel contribution of this research is the Torsionally Compliant Elastomer Joint (TCEJ) which allows for compliance and sensing in a very small package while using extremely inexpensive components. Computational modeling of hysteresis, signal compression, and backlash are also explored to compensate for the non-idealities often found in cheap mechanical parts. Three proof-of-concept systems are described along with a set of experiments used to test their capabilities. Finally, future work is proposed that will likely shape the next generation of low-cost compliant robotics.

Thesis Supervisor: Rodney Brooks

Title: Panasonic Professor of Robotics, EECS



## Acknowledgments

I would like to give special thanks to Rod Brooks and Una-May O'Reilly for their vision, encouragement, and financial support throughout this exploration in low-cost compliant robotics. Without Rod's keen ability to observe future trade-offs and robotic systems that can take advantage of them, this research would not have been possible.

I would also like to thank Aaron Edsinger for his inspiration and support of creative ideas for robotic systems. Aaron was always willing to discuss wild ideas and point me towards helpful references when I needed them. His solid technical foundation and eye for artistic and creative expression are a constant reminder that engineering is not an excuse to disregard the importance of human interaction and emotion in design.

I also have greatly appreciated the unwavering support of my family. They have always believed in me and continue to encourage me to follow my passions in life, whatever they may be. Without them I would not have the same level of creativity, confidence, and drive that have grown to describe who I am.

Lastly, but certainly with no less regard, I would like to thank my good friends Omar Bashir and Kate Hollenbach for helping me to keep moving forward throughout the many ups and downs that accompany any exploratory research project. Omar's sense of humor and deep technical conversations have shaped both my conceptual engineering models and my outlook on life. Kate, a friend in the deepest sense of the word, has been there for many creative conversations over coffee, asking the hard questions but always encouraging me to find the answers.



# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Background Information . . . . .	16
1.1.1	Compliant Robotics . . . . .	16
1.1.2	Series Elastic Actuation . . . . .	17
1.1.3	Compliant Mechanical Design . . . . .	18
1.1.4	Robotic Control Architectures . . . . .	20
1.2	Overview . . . . .	21
<b>2</b>	<b>Future Trade-offs</b>	<b>23</b>
2.1	Computational Power vs. Mechanical Precision . . . . .	23
2.2	Large-Scale vs. Small-Scale Manufacturing . . . . .	24
2.3	Human Form vs. Mechanical Feasibility . . . . .	25
2.3.1	Range of Motion . . . . .	25
2.3.2	Friction . . . . .	25
2.3.3	Actuation Methods . . . . .	26
2.3.4	Control Architecture . . . . .	26
<b>3</b>	<b>Exploring the Problem Space</b>	<b>27</b>
3.1	Rapid Iterative Design . . . . .	27
3.2	Inexpensive Series Compliance . . . . .	30
3.2.1	Rotationally Compliant Joints . . . . .	30
3.2.2	Structural Compliance . . . . .	32
3.2.3	Inherent Drive-System Compliance . . . . .	35

3.3	Modeling of Non-Ideal Components . . . . .	37
3.3.1	Hysteresis . . . . .	37
3.3.2	Signal Compression . . . . .	41
3.3.3	Backlash . . . . .	43
3.3.4	High-Level Models . . . . .	46
3.4	Control Architecture Considerations . . . . .	47
3.4.1	Data Buses . . . . .	47
3.4.2	Prioritized Control . . . . .	47
3.4.3	Weighted Impedance Control . . . . .	49
3.4.4	Adaptability . . . . .	50
3.4.5	Topologies . . . . .	51
<b>4</b>	<b>Building Proof-of-Concept Systems</b>	<b>55</b>
4.1	1 DOF TCEJ Arm . . . . .	55
4.1.1	Design Overview . . . . .	55
4.1.2	Architecture . . . . .	56
4.1.3	Mechanical Hardware . . . . .	57
4.1.4	Electrical Hardware . . . . .	58
4.1.5	Software . . . . .	59
4.2	2 DOF Inexpensive TCEJ Arm . . . . .	60
4.2.1	Design Overview . . . . .	60
4.2.2	Architecture . . . . .	61
4.2.3	Mechanical Hardware . . . . .	62
4.2.4	Electrical Hardware . . . . .	64
4.2.5	Software . . . . .	66
4.3	4 DOF Compliant Drive-System Arm . . . . .	68
4.3.1	Design Overview . . . . .	68
4.3.2	Architecture . . . . .	69
4.3.3	Mechanical Hardware . . . . .	69
4.3.4	Electrical Hardware . . . . .	71

4.3.5	Software . . . . .	71
<b>5</b>	<b>Experiments</b>	<b>73</b>
5.1	Characteristics of TCEJ Hall-Effect Torque Sensing . . . . .	74
5.1.1	Description . . . . .	74
5.1.2	Results . . . . .	75
5.1.3	Implications . . . . .	76
5.2	Response of 2-DOF TCEJ Arm with FSR Torque Sensing . . . . .	76
5.2.1	Description . . . . .	76
5.2.2	Results . . . . .	77
5.2.3	Implications . . . . .	78
5.3	Compliant Painting Task . . . . .	79
5.3.1	Description . . . . .	79
5.3.2	Results . . . . .	81
5.3.3	Implications . . . . .	83
<b>6</b>	<b>Conclusions and Future Work</b>	<b>85</b>
6.1	Conclusions . . . . .	85
6.2	Future Work . . . . .	86
6.2.1	Variable Stiffness Compliance . . . . .	86
6.2.2	High-Level Modeling of Non-Ideal Components . . . . .	87
6.2.3	Environmental Perception . . . . .	87
6.2.4	Engineering for Production . . . . .	88
6.2.5	Data and Command Granularity Optimization . . . . .	88
6.2.6	The Need for a Soft Exterior . . . . .	88
<b>A</b>	<b>Price Data of Mechanical Parts and Computation</b>	<b>91</b>
<b>B</b>	<b>Hysteresis Modeling Code</b>	<b>93</b>
B.1	MATLAB: <i>hysteresis_sim.m</i> . . . . .	93
B.2	MATLAB: <i>fTorque.m</i> . . . . .	95
B.3	MATLAB: <i>fPlasticPWL.m</i> . . . . .	96



# List of Figures

1-1	Block diagram of a Series Elastic Actuator. . . . .	17
2-1	Price trends of precise mechanical parts and computation over time. . . . .	24
3-1	Block diagram of the rapid iterative design process. . . . .	28
3-2	Prototypes of various hubs. . . . .	29
3-3	Torsionally Compliant Elastomer Joint (TCEJ) . . . . .	30
3-4	TCEJ with integrated sensing . . . . .	31
3-5	Various elastomers used in prototypes of TCEJs. . . . .	32
3-6	Lasercut ABS flexure . . . . .	33
3-7	Stacked flexures and aluminum prototypes . . . . .	33
3-8	Lasercut linearly compliant mechanism using parallel structures . . . . .	34
3-9	Inherent drive-system compliance . . . . .	36
3-10	Hub with built-in tensioning mechanisms. . . . .	36
3-11	Graphical representation of hysteresis . . . . .	38
3-12	Block diagram of the hysteresis modeling system. . . . .	38
3-13	Mappings between elastic deformation, torque, and plastic deformation. . . . .	39
3-14	Hysteresis in a Torsionally Compliant Elastomer Joint (TCEJ) . . . . .	40
3-15	Cubic and piecewise linear signal compression . . . . .	41
3-16	Diagram of mechanical backlash. . . . .	43
3-17	Mechanical backlash compensation techniques. . . . .	44
3-18	Simple feed-forward backlash model. . . . .	45
3-19	Pseudocode for a simple prioritized control routine. . . . .	48
3-20	Example of a weighted impedance controller. . . . .	50

3-21	All-in-one control architecture. . . . .	51
3-22	Hierarchical control architecture. . . . .	52
4-1	Proof-of-Concept 1 – Isometric and side views. . . . .	56
4-2	Proof-of-Concept 1 – All-in-one control architecture. . . . .	57
4-3	Proof-of-Concept 1 – TCEJ isometric view . . . . .	58
4-4	Proof-of-Concept 2 – Isometric and top views. . . . .	60
4-5	Proof-of-Concept 2 – Hierarchical control architecture. . . . .	61
4-6	Proof-of-Concept 2 – TCEJ mounted on a window lift motor. . . . .	63
4-7	Proof-of-Concept 2 – Aluminum and lasercut ABS Joint. . . . .	63
4-8	Proof-of-Concept 2 – PCB: DSP daughter board. . . . .	64
4-9	Proof-of-Concept 2 – PCB: PWM H-Bridge mounted on motor. . . . .	65
4-10	Proof-of-Concept 3 – Isometric view. . . . .	68
4-11	Proof-of-Concept 3 – Clamps used to attach carbon-fiber to aluminum. . . . .	70
5-1	TCEJ testing device. . . . .	74
5-2	TCEJ characteristic data showing hysteresis . . . . .	75
5-3	Signal compression applied to a TCEJ with integrated FSRs. . . . .	77
5-4	Incremental position step response of a TCEJ with FSRs. . . . .	78
5-5	Physical layout used for the painting task. . . . .	80
5-6	Painting task montage. . . . .	82
5-7	Closeup of the painted object. . . . .	83

# List of Tables

3.1	Approximate characteristics of standardized high-speed data buses. . .	47
A.1	Price trend data of precise mechanical parts and computation over time.	91
A.2	Description of precise mechanical components. . . . .	91



# Chapter 1

## Introduction

Similar to how Personal Computers help humans display and compute data in spreadsheets of financial information, robots can also likely aid humans with the easy cases of simple tasks. Workers today are not being replaced by PCs, but rather stand to gain significant productivity advantages by offloading work that was once tedious, formulaic, or repetitive to their computational counterparts.

The transition from the mainframe computer to the PC took a rather large shift in perspective. Expensive machines occupying many rooms were eventually replaced by general purpose machines with simple user interfaces costing orders of magnitude less than mainframes and small enough in size to fit on the corner of one's desk.[14] We are currently in the mainframe era of robotics with large, expensive, specialized systems that rarely interact with humans. The hope is that we can now start to envision what the PC of robotics will be like in the future. In particular we would like to know how we can move towards general purpose robots that anyone can use to perform the easy cases of simple everyday tasks.

## 1.1 Background Information

### 1.1.1 Compliant Robotics

Most industrial robotic systems are designed to be very precise, fast, and powerful; a seemingly good set of characteristics to carry out a wide range of tasks. But there are two significant problems that arise from this standard way of approaching robotics. First of all, the speed and power of position controlled industrial robots disallow safe robot-human interaction requiring that special care be taken to disable the robot if a human enters its workspace to reduce the chances of severe injury or death. Secondly, the extremely stiff controller<sup>1</sup> necessary for high speed and precision does not allow for smooth force control. This lack of smooth force control is unfortunate since many common tasks such as “drilling, reaming, routing, counterboring, grinding, bending, chipping, and fettling” require it.[8]

There are, however, a few reasons that most robots today are not compliant. Controllability is certainly more difficult with soft mechanical systems, often imposing upper limits on bandwidth that are much lower than that of stiff mechanical systems. Another reason that compliant robots are rare today is the dramatic increase in complexity necessary to explicitly add predictable<sup>2</sup> and easily sensible compliance. Both controllability and complexity lead to the third, and perhaps most significant reason most robots are currently not compliant; higher development and production costs. This exploration is focused on addressing controllability, complexity, and cost with the hope that it may be a step towards future highly capable and human-friendly compliant robotic systems.

---

<sup>1</sup>The term *stiff controller* is used here to denote that perturbations from the desired position of the actuated degrees of freedom produce significantly high forces. This typically results from a large proportional and/or integral term in the controller.

<sup>2</sup>Predictability is important in compliant robotic design to allow for a mapping between the sensed deformation of the compliant material and the resultant forces or other higher order behaviors (hysteresis, wear, temperature variations, dynamics, etc.) associated with the element.

### 1.1.2 Series Elastic Actuation

Series Elastic Actuators (SEA) offer passive compliance and actuation together to low-pass filter shock loads and in general provide a softer interface to the environment.[15] By measuring the deflection of the elastic component, the controller can drive the actuator in such a way as to mimic a large range of stiffnesses<sup>3</sup>. This ability to dynamically change the perceived impedance of an actuator is extremely useful when performing tasks that vary in certainty, speed, and fragility. Another very nice property of SEAs is that the elastic component's deformation is a measure of the force or torque on a degree of freedom and can be used to perform force or torque control. With both position and force/torque information, a much richer set of control possibilities is achievable than would be with position information alone.[16]

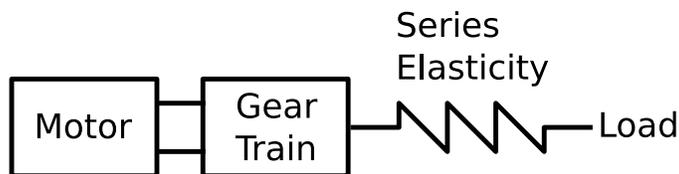


Figure 1-1: Block diagram of a Series Elastic Actuator[15].

Most SEAs are composed of a motor, a gear train, and a series elastic element which drives the load (See Figure 1-1). The motor and gearing are standard electrical and mechanical parts that would otherwise be used to directly control some part of a robot. The elastic element most commonly used is a metal compression spring since the relationship between deformation and force is linear. The spring's deformation is then measured via a potentiometer or hall-effect sensor and fed into the controller to allow for variable stiffness and force control. Since the elastic element is in series with the drive-system, position measurements are made after the compliance (at or near the load) to obtain the most accurate readings possible.

A handful of robots are beginning to exploit the advantages of SEAs and although they are all quite expensive, their ability to dynamically interact with un-

---

<sup>3</sup>There are, however, real-world limitations on the range of stiffnesses an actuator can achieve dependent most strongly on its inherent properties (inertia, back EMF, etc.)[13]

structured environments is beginning to show just how important compliance can be. Domo, a humanoid robot designed and built by Aaron Edsinger and Jeff Webber at MIT/CSAIL, uses 26 Series Elastic Actuators to interact with humans in a safe and well controlled manner. Domo is capable of aiding people in simple tasks without predefined scripts because of the extensive use of force sensing measured from its many SEAs. Instead, Domo makes gestures to signal users that it is ready to perform certain tasks and awaits force, tactile, and visual feedback from the user and the world.[7] SEAs play a crucial role in Domo’s ability to safely interact with humans and dynamically adapt to the unstructured environment in which it is operating.

While simply adding a series elastic element does not seem like it would significantly change the complexity or cost of design and fabrication, it most certainly does. In particular, constraining an elastic element to only be elastic in one degree of freedom (a spring, for example, can be compressed, bent, and twisted if not constrained) is rather difficult and often requires components with very low friction and stiction to produce accurate measurements. Ball screws, precision linear bearings, and strain gauges are often needed, and the cost and required accuracy of placement of such elements can be quite significant. A large number of roboticists understand the numerous advantages that come from softer compliant actuation methods but do not use SEAs because of the increased complexity and cost. Finding a way to make compliant robotics simpler and less expensive is an important area of research and serves as one of the main focuses of this exploration.

### **1.1.3 Compliant Mechanical Design**

Compliance can also be used passively as part of a mechanical structure to allow for a softer interface with the environment and, if well designed, even aid in the ability to perform certain tasks. The placement and stiffness of each compliant region should be carefully chosen to inherently give it the desired mechanical impedances to perform a set of tasks. For example, a 2 degree of freedom planar manipulator can be optimized for grasping a wide range of objects by choosing the best stiffness and rest angle for each joint.[5] If the mechanical structure is wisely chosen for the

tasks being performed, the likelihood of successfully completing a task, often with less controllability overhead by *letting the structure do the thinking*, is much greater. The combination of greater success and easier control suggests that adding passive mechanical compliance where it is most beneficial is an important consideration for all compliant robotic systems.

Mechanical compliance can also take on another form that is not structural; a soft outer covering. While adding a soft exterior to a fast moving and stiff robot is clearly not sufficient to make it safe or gentle<sup>4</sup>, it does lessen the initial impact with objects. When used with other compliant actuation or structural elements, a compliant skin may still be very useful to soften interaction with highly inertial components or to add grip to surfaces (e.g., fingertips). Soft coverings are usually considered after the structural parts of the robot have been fully designed, but it is likely that designs that take into account the need for a soft skin early in the robot's conception will leverage fabrication and joining techniques that directly incorporate compliant materials.

One very interesting way to add compliance to structures is Shape Deposition Manufacturing (SDM) in which objects are cast and machined in regions allowing for different stiffnesses and mechanical properties to be built into one single component.[6] SDM scales nicely to large batches and can produce compliant joints of all sizes without an abundance of mechanical fasteners and parts. Since SDM involves a casting process, structural parts and sensing elements can be placed in the mold at various stages to incorporate metals, ceramics, magnets, embedded force and position sensing, drive-systems, and actuators.

It is worth noting that compliance and force sensing do not necessarily need to be collocated (i.e., within the same component or location), although in most cases it is convenient to measure the deflection of the compliant material to obtain the forces or torques present. The separation of compliance and force sensing has some potentially useful implications. For example, an extremely non-linear, hysteretic, difficult-to-measure, or inexpensive component could be used for compliance while

---

<sup>4</sup>Robots are usually characterized by the Human Injury Criterion to determine their safety for human interaction.

still maintaining force sensing accuracy and resolution via a force sensor located near the end-effector<sup>5</sup>.

#### 1.1.4 Robotic Control Architectures

A robotic control architecture specifies the computational building blocks of a system and the associated interfaces between them and the environment. Sensors serve as the system's inputs and actuators take system outputs and cause a change in the world. The simplest architecture is that of a single computational block that drives an actuator based on readings from a sensor. All system state is stored in one place and can be used for modeling, dynamics, and control. This simple all-in-one architecture works well for a small number of sensors and actuators, but quickly becomes inferior to more structured and scalable approaches when using many inputs and outputs.

Simply distributing the computational burden among many blocks helps to alleviate the problem of scalability, but without a clear assignment of the blocks in charge of higher-level processes, control becomes very difficult. Many high-level modeling and perception routines require information about the state of the entire robot to choose which task should be performed next or how to dynamically change commands to adjust to a perceived situation. A hierarchical approach allows for distributed computation, variable time and task granularity at each level, and allows higher-level blocks to accumulate important state about lower levels to perform dynamic modeling and perception.

Many hierarchical control architectures have been developed for robotic platforms, including a very well researched telerobot control system created by NASA, NASREM.[3] While the details of NASREM are outside the scope of this research, it is worth noting two of the key elements that make it both extremely robust and easily expandable. The first element is the scaling of urgency based upon depth within the hierarchy. This means that small perturbations in an actuator's position (low-level) should be handled as quickly as possible, while altering an actuators trajectory (high-

---

<sup>5</sup>It is typically best to place position and force sensing elements as close to the output as possible to reduce inaccuracies caused by friction, stiction, slippage, and noise

level) can occur at a substantially slower rate. Secondly, NASREM divides each goal into sub-goals recursively and plans ahead based on its ability to complete lower-level goals. For example, if a fingertip cannot move to its desired position within a specified time window, a higher-level controller will re-plan future sub-goals accordingly.

It is important to note that the way in which blocks in a control architecture are connected together is an area of research in and of itself. The number of data buses from which to choose is quite large with slight differences between each one to enhance it in some way. Some buses offer higher speeds, less wires for connectivity, less latency, easier scalability to a large number of nodes, and error detection. But these enhancements do not come without a price, making for a delicate trade-off that must be carefully considered to select an appropriate data bus for the constraints of a given control architecture.

## 1.2 Overview

This exploratory research leverages the trade-off between mechanical precision and computational power. Low-cost compliant robotic parts are created along with a set of modeling tools to compensate for their non-ideal characteristics. Three robotic arms are also designed and constructed to test the compliant elements, computational models, control architectures, and mechanical structures. Chapter 2 discusses potential trade-offs that are likely to affect the way future robot systems are constructed. The design approach, various low-cost compliant elements, computational models, and architecture considerations are then presented in Chapter 3. Each of the three robotic arms are then described in detail and tested in Chapter 4 and Chapter 5, respectively. Finally, conclusions and future work are presented in Chapter 6.



# Chapter 2

## Future Trade-offs

By considering trade-offs that are likely to influence the future of robotics, explorations can be focused on developing methods, technologies, and implementations that push the field of robotics forward. This section describes some of those trade-offs along with their likely effects on the ways we perceive future robotic systems.

### 2.1 Computational Power vs. Mechanical Precision

Recent history has shown that the price of precise mechanical parts is roughly constant over time while the price of computation<sup>1</sup> is falling at an exponential rate[4] (see Figure 2-1). This suggests that a wise choice may be to replace mechanically precise parts with lower-cost equivalents by compensating for them with computation. In particular, as long as the non-linearities associated with friction, stiction, and imprecise manufacturing can be sensed, it is likely that a well designed controller can compensate for the non-ideal behavior of lower-cost mechanical parts.

---

<sup>1</sup>The price of computation is approximated here by the cost of 100 Million Instructions Per Second (100 MIPS).

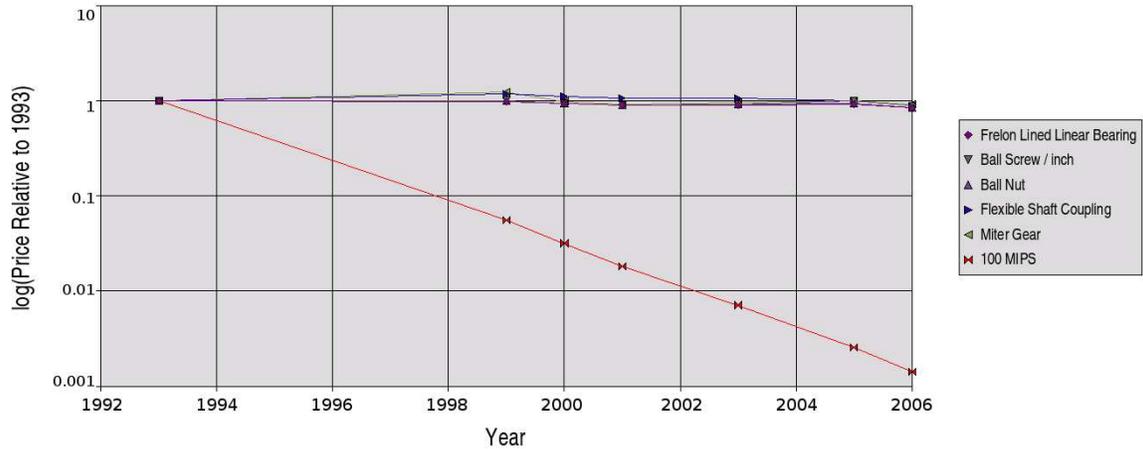


Figure 2-1: The price trends of various precise mechanical parts and computation over time. Prices have been adjusted for inflation by using the average US Consumer Price Index of common goods (fuel, energy, bread, etc.) relative to 1993. Numerical data can be found in Appendix A.

## 2.2 Large-Scale vs. Small-Scale Manufacturing

Currently robots are used mostly for large-scale manufacturing. But similar to the way mainframe computers became more abundant and useful when they evolved to Personal Computers, it is worth considering how robots may also become more prevalent and helpful when used with small-scale manufacturing.

Imagine a small pastry shop of only 4 employees and the simple cases of easy tasks they could offload to an easy-to-use small-scale manufacturing robot. Humans would almost certainly still be performing the more difficult tasks of measuring ingredients and carefully shaping or folding pastries, but a robot could easily assist in the placement and removal of baking sheets from the oven, potentially freeing up an employee to work a cash register or begin making the next set of pastries. Safety, cost, and ease-of-use issues have kept this from becoming a reality, but with the possibility of low-cost compliant robotics, it may not be too far away.

## 2.3 Human Form vs. Mechanical Feasibility

Many robots are designed to mimic the form and function of humans performing tasks. It is likely, however, that more efficient and sensible mechanical solutions exist that take advantage of the ability to decompose the task into parts that can be handled by different robotic systems. For example a dexterous hand need not be at the end of an arm, but rather, could be placed within a table where motors and sensors can be used without the common weight constraints imposed upon them.

### 2.3.1 Range of Motion

Mammalian joints often have a limited range of motion, whereas a simple electric motor can rotate without limitation. While this may seem quite obvious, it is an important distinction that allows robots to excel at configurations and tasks where an extremely wide range of motion is necessary. For example, if a human holds a screw driver in his/her hand and wishes to turn it two full rotations, releasing and re-grasping the screwdriver, perhaps several times, will be required. However, if the screwdriver was being used by a robot with an unlimited rotational range of motion it could complete the task without the need to release and regrasp the screwdriver.

### 2.3.2 Friction

The lining of human joints contains synovial fluid providing for a very low coefficient of friction of less than 0.01, while man made substances specifically designed to reduce friction between parts, like Teflon, have a coefficient of friction of about 0.04. This means that even the most carefully designed robots often still contain a significant amount of friction, either at the joint, within gearing, or somewhere in the means for power distribution (cabling, hydraulics, pneumatics). Choosing designs and actuation methods that allow for robust modeling and compensation of friction between parts is likely to be a very important consideration.

### 2.3.3 Actuation Methods

There are many different actuator technologies[11] to choose from when deciding how a robotic system will be driven. Muscles are the actuators used by humans, but nothing quite like muscle has been constructed that can easily integrate with the electronic nature of robotic control systems.<sup>2</sup> The most common forms of electrical to mechanical actuation are electromagnetic motors and solenoids. Shape Memory Alloy (SMA), a metal wire that constricts when it is heated<sup>3</sup>, seems like a good candidate for mimicking mammalian muscles, but its dependence on temperature and limited relaxation rate due to necessary cooling pose severe limitations.

### 2.3.4 Control Architecture

Contrary to the signals used for electrical circuitry that travel at the speed of light, signals found in biological organisms move at a much slower rate. In humans, the “wiring” (i.e., neurons) consists of a potassium/sodium ion channel that uses more or less of those ions to cause electrical potential differences (voltages). While the electrical currents themselves propagate at speeds similar to that of electrical circuits, the extra time necessary to push more or less ions into the channel is relatively slow, and made even slower by the necessary accumulation of calcium ions in the neuromuscular junction to produce muscle contraction. This limits the closed loop bandwidth for visually servoed muscle control from the brain significantly to  $\approx 7Hz$ , and from pattern based muscle control occurring in the spine to  $\approx 35Hz$  [12][17]. The substantially higher closed loop control bandwidths possible in electrical robotic control architectures can likely provide much faster and more precise movement than can be achieved by humans.

---

<sup>2</sup>Power density is not the limiting factor when comparing mammalian muscle to high-end robotic actuator technologies. Instead, the main differences lie in the form, inherent non-linearities, control structure, and power-distribution methods.

<sup>3</sup>Heating is often created in SMA by running a current through the wire. Power dissipation is  $P = I^2R$ , so doubling the current through the wire causes a 4x increase in joule heating.

# Chapter 3

## Exploring the Problem Space

This chapter explores potential solutions to the need for low-cost compliant robotic parts as well as ways to compensate for their non-idealities. The problem space is quite large, so this chapter focuses on the most promising ideas developed during the course of this exploration to serve as a springboard for future research.

### 3.1 Rapid Iterative Design

To conceive, design, test, and learn about new ideas as fast as possible, a modified version of the rapid iterative design method was used. Simple iterative design methods typically consist of only three main steps (e.g., *Observe*  $\mapsto$  *Brainstorm*  $\mapsto$  *Prototype* is the method used by IDEO, a world-renowned product design company) and leave much of the implementation and important details up to the designer. The modifications I made to the process make the inputs and outputs of each step explicit and also include additional steps to allow for easier and more structured implementations.

Each iteration of the iterative design process (see Figure 3-1) starts with active observation, interactively taking in the people, places, and things around at the time. By actively manipulating, feeling, conceptualizing, and experiencing that which is observed, one can begin to find problems with their surroundings and also take note of solutions to similar or related problems. Once a set of problems, or problem space, has been identified it can be used as the centerpiece for brainstorming numerous

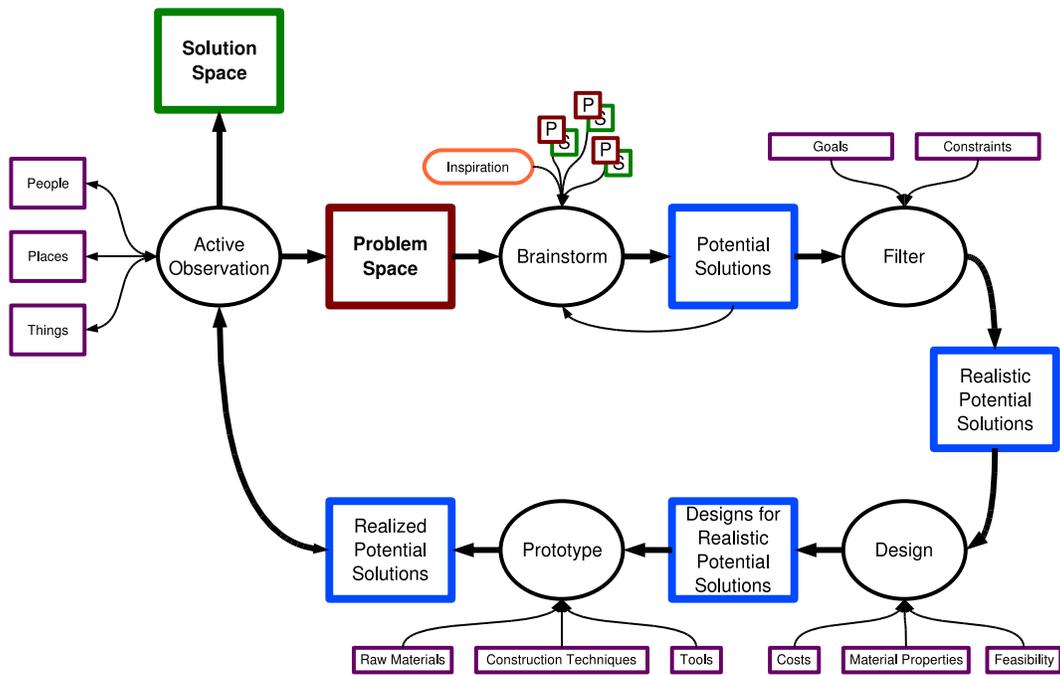


Figure 3-1: Block diagram of the rapid iterative design process.

useful and potentially exotic ideas. Some ideas follow naturally from the observed problems while others may require the study of various *problem*  $\rightarrow$  *solution* mappings or inspiration from seemingly unrelated experiences, thoughts, observations, and fields of study. Ideas generated during the brainstorming process should not be critically analyzed at the time since doing so is likely to keep unconventional or wild ideas from being considered, which if not useful themselves, often can at least spark new ways of thinking.

After many potential solutions have been created from the brainstorming process, the next step is to filter them to find the subset of ideas which can be realistically pursued. A set of constraints and goals are taken into account at this step as well since they shape what constitutes a *realistic* idea that has the potential for success. Designs are then created from the set of realistic potential solutions considering the feasibility of production methods, material properties, and costs. The set of designs for the potential solutions take on a wide range of forms (from simple napkin sketches to

fully specified 3D CAD models) dependent on the materials on hand, the complexity of the idea, and the time allotted.



Figure 3-2: A handful of prototypes created to explore various drive hubs with built-in tensioning mechanisms. All of the hubs shown here were designed, fabricated, and tested within a few days' time.

Prototyping, perhaps one of the most exciting parts of the rapid iterative design process, takes in a set of designs, raw materials, construction techniques, and tools to produce realizations of the ideas being considered (see Figure 3-2). In most cases production can be very time consuming and costly, so prototyping uses a slightly different set of tools to make quick realizations of ideas without the expensive processes or time consuming nature of standard fabrication. An incomplete, but fairly comprehensive, set of tools used for prototyping includes: paper-prototyping supplies (paper, glue, pens, transparency sheets, scissors, etc.), machining tools (various hand tools, mill, lathe, bandsaw, press, etc.), foam-cutter, lasercutter, 3D printer, water-jet cutter, small-batch polymer casting supplies, sheet metal cutters and shapers, and computer/software packages for modeling, testing, and visualization.

Once the ideas have been realized, they do not immediately become solutions. The iterative design process starts over back at the active observation stage at which point the prototypes can be manipulated, felt, conceptualized, and experienced along with the other observations of people, places, and things to hopefully add both new problems and solutions to their corresponding spaces. There is no end to the cycle, but time and cost constraints are likely to put the cycle on hold once a set of solutions

(or lack of satisfactory solutions in some cases) has been developed.

## 3.2 Inexpensive Series Compliance

### 3.2.1 Rotationally Compliant Joints

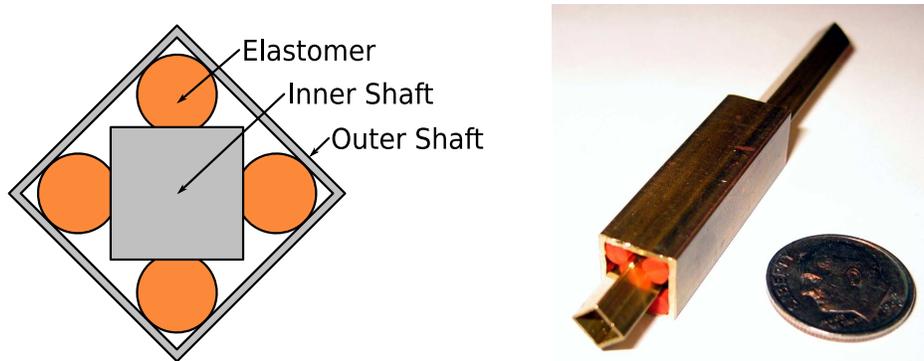


Figure 3-3: Torsionally Compliant Elastomer Joint (TCEJ) – **Left**: Top view of joint. **Right**: Isometric view of joint prototype.

Many designs and configurations for a self-contained rotationally compliant joint were considered, but one seemed to rise above the rest due to its simplicity, low-cost, and the ability to control its compliant properties. The Torsionally Compliant Elastomer Joint (TCEJ), shown in Figure 3-3, consists of an inner shaft coupled to a hollow outer shaft via an elastic element. By using a regular polygon with a small number of sides (in this case  $n = 4$ ; a square), the elastomer is highly constrained and unable to slide along either the inner or outer shaft walls. When a torque is applied to either shaft, the elastomer will be compressed slightly, transferring the torque, but also allowing for an angular displacement proportional to the load. This torque proportional to angular displacement behaves much like a torsional spring and provides a compliant interface between the input and output shafts.

In search of background information on similar torsionally compliant configurations I came across Dexter Axle's *Torflex*<sup>TM</sup>. Originally patented[1] at least as far back as 1963<sup>1</sup>, the *Torflex* is a passive damping axle for trailer suspension systems.

<sup>1</sup>After contacting Dexter Axle, there was a suggestion that the original patent may date back to

The configuration is extremely similar to the TCEJ, but the outer shaft is welded to the frame of a vehicle and the inner shaft is attached to one of the vehicle's tires. As the vehicle encounters bumps in the road, the elastomer passively deforms offering for a smoother ride. The mechanical configuration of the TCEJ and the *Torflex* are almost identical, but two key differences should be noted – The TCEJ is both actively driven and includes integrated torque sensing.

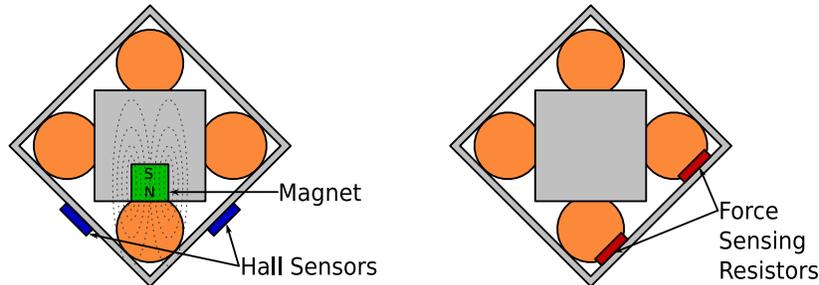


Figure 3-4: TCEJ with integrated sensing – **Left**: A differential pair of hall effect sensors are used to measure angular difference between inner and outer shafts. **Right**: Force Sensing Resistors (FSR) differentially measure the orthogonal force between the elastomer and the outer shaft wall.

Two different methods were used to achieve torque sensing within the TCEJ. The first sensing method was that of an embedded magnet and a differential pair of hall-effect sensors (see Figure 3-4:**Left**). The angular displacement between the inner and outer shafts was measured by the relative magnetic field strength observed at the two hall sensors. The mapping between angular displacement and torque is not direct with highly hysteretic elastomers, but this can be compensated for as described later and shown in Figure 3-14. The second method of sensing torque was to use a differential pair of embedded Force Sensing Resistors (FSR) placed between the outer shaft wall and the elastomer (see Figure 3-4:**Right**). FSRs are often used with skepticism in robotics since they only accurately measure forces orthogonal to their surface. However, in this particular configuration, only measuring orthogonal forces actually makes the sensing more robust since it will not be greatly affected by shear forces or strains that may result from pre-loading or hysteresis. The FSR as early as 1956 filed in Germany under the inventor's name Niehart, but I was not able to locate it in *esp@cenet*, the public European patent database.

sensing configuration also has the benefits of easily scaling to larger joints and not being strongly affected by noisy magnetic fields.



Figure 3-5: Various elastomers used in prototypes of TCEJs. **From left to right:** Buna-N, Silicone, Neoprene, multi-strand Latex (bungee).

The compliant properties of a TCEJ can be changed drastically by scaling the joint, using different elastomers, setting the elastomer's pre-load, and choosing the difference in size between the inner and outer shafts. With all else held constant, a longer joint will be stiffer than a shorter joint because more elastomer will be deformed for a given angular displacement. There are many different elastomers from which to choose and the properties between them vary immensely. Figure 3-5 shows a few examples of elastomers that were tested. Silicone is soft and fairly temperature invariant, but lacks tensile strength and tears easily. Buna-N, a Nitrile elastomer, is chemically inert and abrasion resistant but suffers from significant hysteresis. Latex is very stretchy, cheap, and has low hysteresis but breaks down easily when exposed to most oils and UV radiation. Cast polyurethane, with many durometers and resiliencies to choose from, is likely to be a good candidate but requires high pressure during the casting process to pre-load the joint.

### 3.2.2 Structural Compliance

A large number of mechanical configurations and fabrications techniques were explored in an effort to find low-cost structural compliance. Lasercutting materials with significant complexity seemed to be one of the fastest ways to test ideas, but its limitations on material choice and non-planar cutting produced results that were observed to be only marginally satisfactory. Previous research done with Shape Deposition

Manufacturing (SDM) seems like a much more reasonable candidate for structure compliance in future explorations.



Figure 3-6: Lasercut ABS flexure – **Left**: No torque applied. **Right**: External torque applied.



Figure 3-7: **Left**: Stacked lasercut ABS flexures used for a mechanically compliant arm. **Right**: Aluminum prototypes of flexures.

Figure 3-6 shows a structurally compliant part lasercut from 0.25” ABS plastic to allow for bending along a single axis (a flexure). By varying the thickness of the veins and roundedness of all cuts, a wide range of stiffnesses can be achieved while avoiding excessive stress concentrations around sharp corners. Other materials, such as PVC, would have been interesting to test, but due to hazardous fumes and acid that result from cutting the material with a laser, it was not possible. As shown in Figure 3-7, the flexures can be stacked to increase rigidity in the non-compliant direction, while also increasing the stiffness (similar to springs in parallel) of the overall structurally compliant arm. This particular arm quickly fatigued with bending due to square cuts

and the choice of material; ABS. The right image in the figure shows another attempt at a similar structurally compliant configuration, but machined out of aluminum. Using spring steel would have allowed for more bending before plastically deforming, but aluminum served as an easy-to-machine prototype of the idea to observe some of its basic properties.

Another important class of structurally compliant components are those that deform linearly instead of rotationally. This kind of behavior is most commonly created by constraining a compliant element (like a spring) such that deformation can only occur in a single direction. This often involves a set of precise mechanical parts that provide a track or slide to only allow linear movements and can be quite expensive and complex to properly implement. If, however, the linear nature is inherent in the mechanical configuration of the compliant element, there is likely to be an opportunity for a much simpler and lower cost solution.

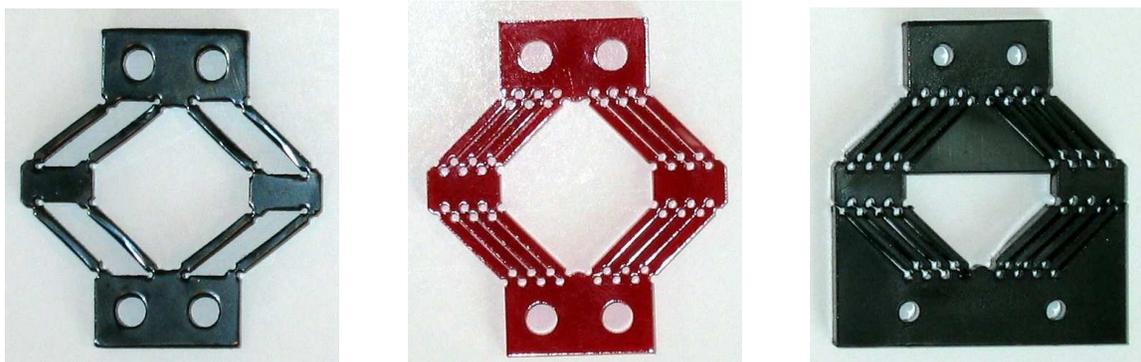


Figure 3-8: Three versions of a lasercut linearly compliant mechanism using parallel structures. **Left:** Initial version made from ABS to test strength and stiffness. **Middle:** More links added to increase stiffness and an acrylic version (shown) is tested for material comparison to ABS. **Right:** Limiting elements added to protect against overloading in either direction.

Figure 3-8 shows three prototypes of a linearly compliant mechanical structure. The left image shows the first design, lasercut from black ABS, based on a set of parallel linkages that are free to hinge at regions where the material is very narrow in width. The observed behavior of the structure was very soft and compliant, but was prone to quickly fatigue and break after applying a sizable force a small number

of times. For the second version, shown in the middle, more linkages were added to increase stiffness and lessen the likelihood of breaking. The part was then lasercut out of both ABS and red acrylic. The added parallel linkages did strengthen and stiffen the structure but could still be broken under a heavy load. The acrylic was stiffer than the ABS, but instead of fatiguing to the point of failure, it immediately shattered under sizable loads. The final version, shown on the right, has added mechanical constraints to limit the range of motion under loads great enough to damage the material. This version worked quite well and did not seem to suffer from fatigue issues (at least an order of magnitude better than previous versions). The range of motion is fairly small and could likely be sensed well with a magnet and hall-effect sensor, a potentiometer, or an embedded force sensing resistor.

### 3.2.3 Inherent Drive-System Compliance

Instead of putting compliance in the mechanical structure or the joint, a third option exists to put the compliance in the drive-system<sup>2</sup> itself. While all series elastic actuators place compliance between the actuator and the output, inherent drive-system compliance is more specifically focused on modifying the drive components of non-compliant actuation systems by choosing softer materials to give them compliant properties. The softness is then considered to be inherent in the component's composition.

Figure 3-9 shows an inherently compliant drive-system that uses nylon instead of steel cabling. Most cable driven robotic systems go to great efforts to eliminate the stretch often associated with inexpensive cables. But when compliance is desired at the output, allowing the cable to stretch proportionally to its tension is an option worth exploring. The use of a stretchy cable to mimic an inline spring has been studied and shown to work successfully on a very small scale[21] but has yet to be applied to larger systems. It is important that the cable never becomes loose (loose

---

<sup>2</sup>*Drive-system* is used here to denote the element in between the actuator and the degree of freedom being controlled. The drive-system typically consists of gears, cables, hydraulics, pneumatics, chains, belts, or other power distribution and transformation elements.



Figure 3-9: Inherent drive-system compliance is produced by exploiting the stretchy nature of amorphous nylon fishing line used as cabling.



Figure 3-10: Hub with built-in tensioning mechanisms to allow for the large displacement necessary to pre-load a stretchy cable. **Left:** Top view of hub. **Right:** Bottom view of hub.

cabling may cause misalignment or allow the cable to slide off of the hub) so a method for sufficiently pre-loading the cable is necessary. The method used in Figure 3-9 is very similar to a guitar tuning peg in which the cable is fed through a hole in the hub and then wound around a screw until the desired tension has been reached (see Figure 3-10). Standard steel cable tensioning mechanisms were not sufficient for the stretchy nylon line (fishing line) since it required a large amount of displacement to obtain the necessary amount of pre-load.

Sensing the amount of tension in the cable directly would be quite difficult since the cable moves a large distance as it is actuated. However, if the cable is prohibited from slipping on the hub, the difference between the angle of the actuator hub and

the output hub can be used to approximate the differential tension in the cable. This sensed difference can then be used as a measure of the torque on the output degree of freedom and therefore used for impedance control. Another option would be to literally attach springs to steel cabling as a way to give the otherwise stiff cabling compliance, similar to what is achieved by Obrero's robotic hand developed by Eduardo Torres-Jara.[18] Sensing could be accomplished in the same ways describe previously, but special care would need to be taken to ensure that the spring does not interfere with either hub throughout its entire range of motion.

Another seemingly good choice for an inherently compliant drive system is that of hydraulics. Most fluids can be modeled as incompressible, but if a small amount of gas leaks into the system, it will allow for some compressibility. By controlling the amount of air in a hydraulic system, its compressibility can be greatly altered, but compressibility of the fluid alone does not guarantee compliance at the output. One of the biggest drawbacks of hydraulic systems is the stiction seen at the piston (the output) that resists a change in position until a significantly large force is applied. This behavior is detrimental to using compressible hydraulic fluids as a compliant drive-system since small forces will never be observed or controllable by the actuator.

## 3.3 Modeling of Non-Ideal Components

### 3.3.1 Hysteresis

Hysteresis occurs when a stress is applied to a material, but due to energy losses, the associated material strain differs between the loading and relaxation phases of the applied stress. Said differently, pressing on a material can cause both *elastic deformation* (e.g., an ideal spring that returns to its initial rest position when the force is removed) and *plastic deformation* (e.g., pressing on a piece of clay that takes on a new shape even after the force is removed). It is this plastic deformation that shifts the rest position of the material and thus causes hysteresis. Figure 3-11 graphically shows the effect of applying a torque to a material that is substantially hysteretic.

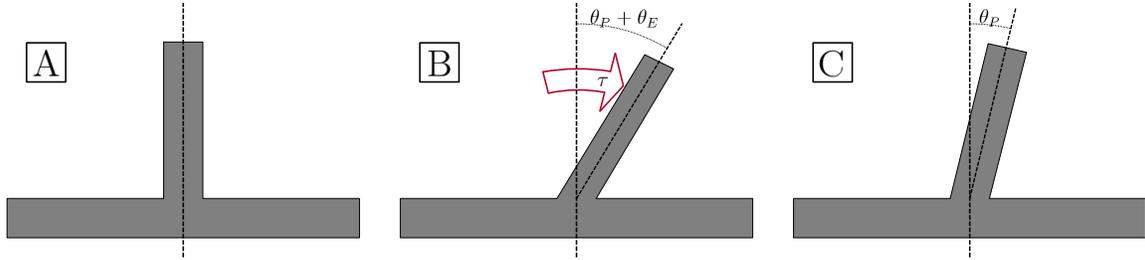


Figure 3-11: Graphical representation of hysteresis – **A**: A piece of virgin material is unstressed giving rise to no plastic or elastic deformation ( $\theta_P = 0$  and  $\theta_E = 0$  respectively). **B**: An external torque is applied to the material resulting in both plastic and elastic deformation. **C**: The external torque is removed, resulting in zero elastic deformation and some plastic deformation from the previously applied torque ( $\theta_E = 0$  and  $\theta_P > 0$ ).

Note that, by definition, elastic deformation must be zero at steady state when no external torque is applied.

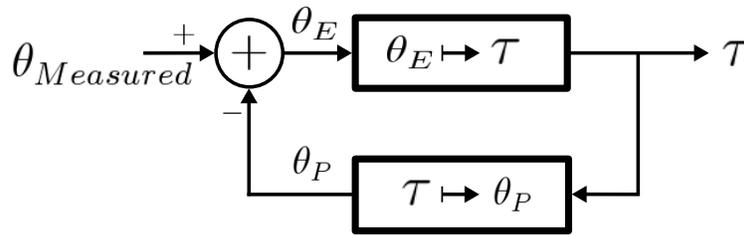


Figure 3-12: Block diagram of the hysteresis modeling system ( $\theta_{Measured} = \theta_P + \theta_E$ ).

It is often the case in compliant robotic systems that the value of the force or torque on a joint, beam, or drive-system is desired. By using Young’s Modulus<sup>3</sup>, the amount of elastic deformation (strain) can be used to calculate the force or torque (related to stress) on a material. The problem, however, is that the elastic deformation cannot be measured directly if the material is significantly hysteretic. To alleviate this problem, I designed a hysteresis modeling system that uses only the total measured deformation of the material and computationally solves for the elastic deformation. This elastic deformation can be obtained indirectly by subtracting the plastic deformation from the measured deformation, given that the plastic deformation can be produced from

<sup>3</sup>Young’s Modulus is used to relate stress to strain in a material and is physically perceived as stiffness.

its previous state and the amount of applied torque. To summarize, the torque( $\tau$ ) is a function of elastic deformation( $\theta_E$ ), the plastic deformation( $\theta_P$ ) is a state-dependent mapping of the torque( $\tau$ ), and the measured deformation( $\theta_{Measured}$ ) is the sum of both the plastic( $\theta_P$ ) and elastic( $\theta_E$ ) deformation terms. Figure 3-12 shows a block diagram of the proposed hysteresis modeling system.

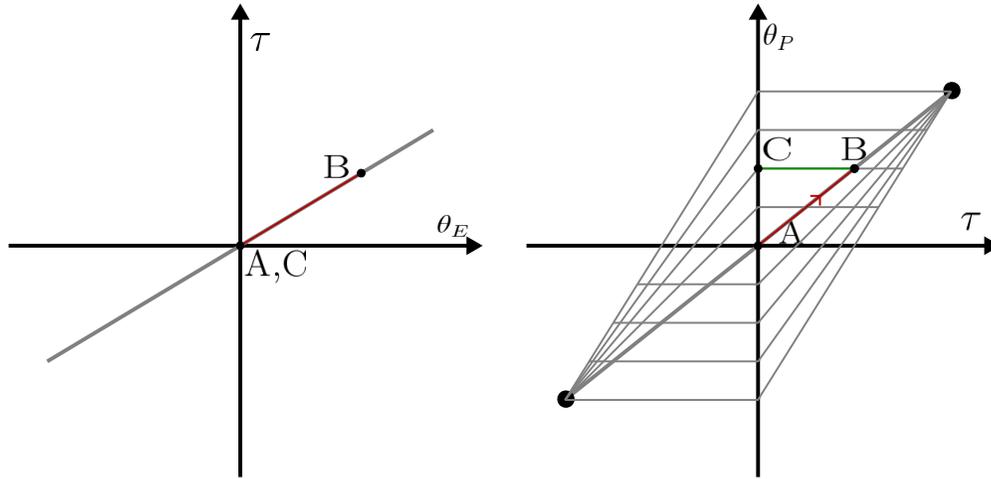


Figure 3-13: **Left:** Mapping from the amount of elastic deformation( $\theta_E$ ) to the resulting torque( $\tau$ ). **Right:** Family of mappings from the applied torque( $\tau$ ) and the previous amount of plastic deformation to the resulting amount of plastic deformation( $\theta_P$ ) caused by that torque.

For this exploratory research, a simple linear mapping was used for both of the blocks in Figure 3-12 ( $\theta_E \mapsto \tau$  and  $\tau \mapsto \theta_P$ ). These mappings are shown in Figure 3-13 and a conceptual explanation of their form follows.

The elastic deformation to torque mapping (see Figure 3-13:Left) is that of an ideal spring (Hooke's Law). The torque that the spring applies is directly proportional to the displacement from its rest position. Following the scenario depicted in Figure 3-11, the material starts out with no torque applied and thus sits at point **A**. As the material is deformed by an external torque, an opposing torque is applied by the material proportional to the amount of elastic deformation to point **B**. When the external torque is removed, it relaxes back to point **C**, which from the definition of elastic deformation must be collocated with the original rest point **A**.

The less intuitive mapping of torque to plastic deformation (see Figure 3-13:Right)

contains a family of linear segments dependent on the previous amount of plastic deformation. The rest position of the material shifts proportional to the amount of torque applied, but is heavily influenced by its previous rest position. Following the scenario from Figure 3-11, the virgin material starts with no plastic deformation at point **A**. When a torque is applied, the material plastically deforms, equivalently changing its rest position as seen at point **B**. As the torque is removed, the amount of plastic deformation remains constant, sliding horizontally to point **C**. Note that since no torque is applied in the opposite direction, the amount of plastic deformation does not change. This is an important property of the  $\tau \mapsto \theta_P$  mapping since a change in the plastic deformation on the unloading of a torque would be an elastic behavior (and thus should be included in the  $\theta_E \mapsto \tau$  mapping) instead of the plastic behavior desired. The simple model used in Figure 3-13:**Right** could be modified to include time-dependent relaxation (e.g., soft foam that slowly expands after being pressed) but was not fully implemented due to its increased complexity and additional non-linearities specific to each hysteretic material used.

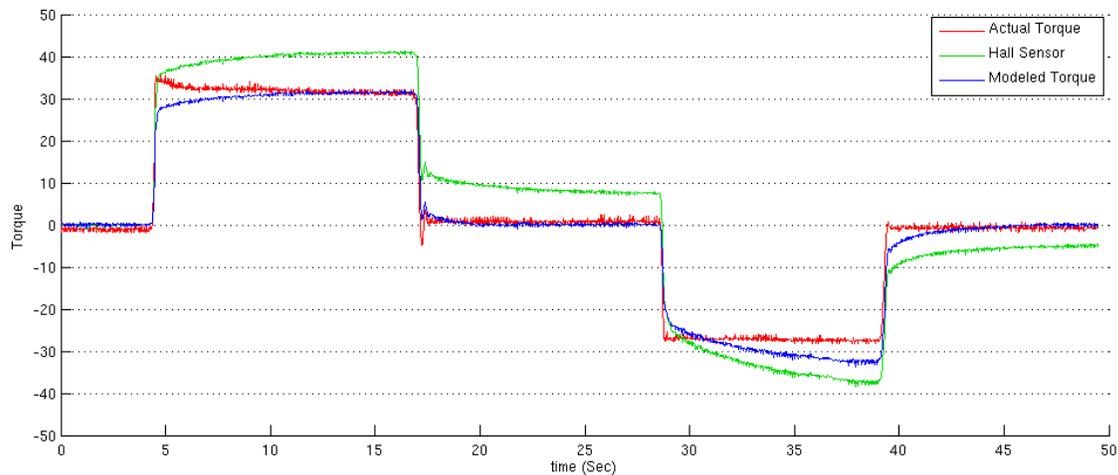


Figure 3-14: Example of hysteretic behavior observed in a Torsionally Compliant Elastomer Joint (TCEJ). Notice that the plastic deformation of the elastomer produces a non-zero sensor reading even when the actual torque on the joint is zero. The plastic deformation is modeled to offset its effect.

A system implementing a set of the two mappings ( $\theta_E \mapsto \tau$  and  $\tau \mapsto \theta_P$ ) similar to the linear model previously described was implemented and run with a set of

measured test data. The test consisted of steps in torque applied to an arm mounted on a Torsionally Compliant Elastomer Joint (TCEJ). The directly measured torque data was used only for comparison after running the sensed angular displacement data (acquired via differential hall effect sensors) through the hysteresis model. The results of using the hysteresis model, shown in Figure 3-14, suggests that even a small amount of modeling can drastically improve the behavior of a significantly hysteretic system. The elastomer’s relaxation time was not included in this simple model, but would have likely reduced the transient decay still remaining in the modeled torque as the rubber slowly relaxed after each step. The Matlab modeling code used for this example can be found in Appendix B.

### 3.3.2 Signal Compression

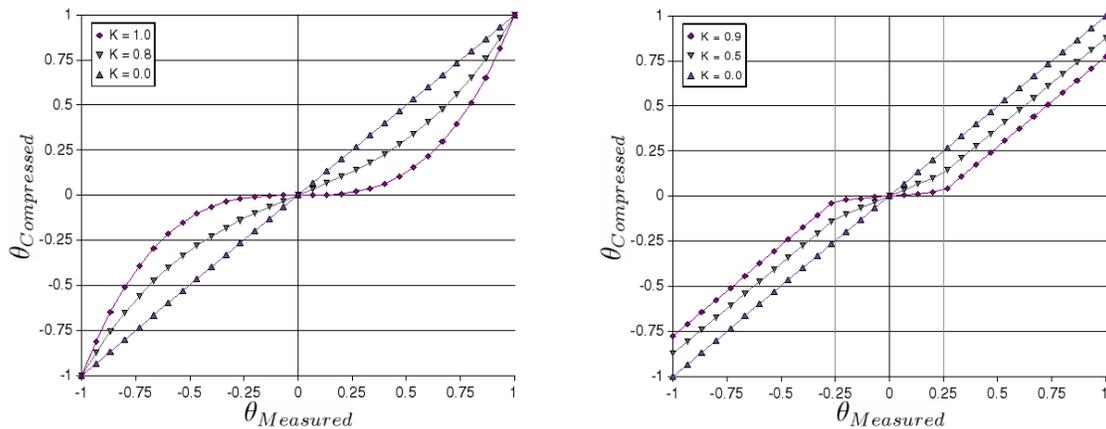


Figure 3-15: **Left:** Cubic compression.  $\theta_{Compressed} = K^3\theta_{Measured}^3 + (1 - K^3)\theta_{Measured}$  **Right:** Piecewise linear compression.  $\theta_{Measured}$  is multiplied by  $(1 - K)$  for values within a chosen compression window ( $window = 0.25$ ).

Certain sensing or control methods often give rise to an undesirable amount of increased sensitivity, particularly in low-level zero-centered signals. For example, if significant pre-loading of differential force sensors is not obtainable, there may be a jump in sensor readings around zero that does not realistically correspond to a drastic change in force. Compressing the signal around zero is one way (a different sensor configuration may be a much better way) to lessen the effect of overly sensitive

low-level sensor readings.

Two simple ways of applying compression to a measured signal were developed and are shown in Figure 3-15. The left image shows a cubic compression model with varying amounts of compression dependent on  $K$ . The mapping is purely cubic with  $K = 1.0$  and becomes more linear as  $K \rightarrow 0.0$ . The image on the right shows a piecewise linear compression model. Measured values within a given window (here the *window* was arbitrarily chosen to be 0.25) are compressed by  $K$  (multiplied by  $1 - K$ ). Values outside the window are not compressed, but do get shifted to eliminate discontinuities between the compressed and non-compressed regions. With  $K = 1.0$ , the measured values are completely compressed to 0.0 within the *window* and as  $K \rightarrow 0.0$ , there is no longer an attenuation applied to low-level signals.

Both the cubic and the piecewise linear compression models are useful, but they should be chosen carefully to match the system's observed non-linear behavior. If the measured signal's sensitivity is too high for small values and gradually decays away for larger values, a cubic model is probably a good place to start. If, on the other hand, the measured value is fairly linear but seems jumpy for small values, a piecewise linear approach is likely to produce better results. Another factor to consider when choosing a compression model is the stability of the controller since the hard *knee* of the piecewise linear model may cause abrupt transitions to the output. It is also important to note that very high compression levels ( $K \rightarrow 1.0$ ) are not recommended for either compression model as low-level signal information is likely to be set to zero and completely lost.

As an example, consider the jumpy sensor mentioned previously in this section. Using cubic compression would greatly reduce sensitivity to the jumps around zero, but would also create significant non-linearities that could pose problems in certain control implementations. If, instead, piecewise linear compression is used with an appropriate window corresponding to the size of the jump and a fairly large compression value, the discontinuity could be drastically reduced around zero while still maintaining a linear output for values outside of the compression window. If, on the other hand, the sensor is only slightly more sensitive around zero than it is for

larger values (i.e., no discontinuities or jumpiness), cubic compression may afford a better solution due to its inherent smoothness than the piecewise linear approach can supply.

### 3.3.3 Backlash

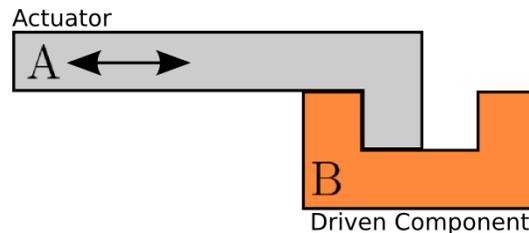


Figure 3-16: Diagram of mechanical backlash between an actuator **A** and a driven component **B**. Notice that **B** is free to move a small amount relative to **A**, regardless of the actuator's stiffness.

Backlash, the unconstrained movement or play between actuator mechanisms (Figure 3-16), produces a set of very difficult controllability problems. In particular, position and velocity control become difficult since small perturbations around the control reference will require large displacements of the actuator to remain stiff. The momentum associated with large actuator displacements is likely to cause the the output to overshoot, rapidly leading to oscillations for typical proportional and integral gains. Lowering the  $P$  and  $I$  terms in the controller does help to achieve stability, but comes at the price of significantly less stiffness and greater steady-state error. Expensive motors and gears often include anti-backlash technologies that preload the drive mechanisms to reduce or eliminate play in the output, but their high cost and necessary mechanical precision make them undesirable for low-cost compliant robotics.

Mechanical backlash reduction can take on two forms; filling in the gaps to limit unconstrained movement or externally applying an offset force to keep actuator mechanisms pushed tightly to one side (see Figure 3-17). One common way to fill in the gaps is to make adjustable drive components that can be rotated or expanded to reduce the play between gears or slides. This is the most direct way to rid the system

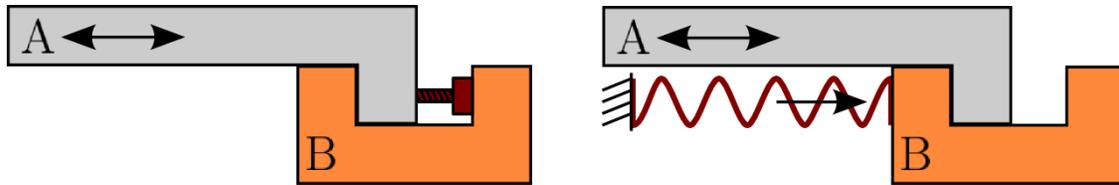


Figure 3-17: **Left:** Backlash is reduced by filling in the gaps that allow for unconstrained movement between parts. **Right:** Backlash is reduced by applying an external force that keeps parts pushed to one side.

of backlash, but it is likely to need frequent adjustments if the part is not auto-expanding (spring loaded) to compensate for part wear over time. Reducing play by filling in the gaps may also not be obtainable in actuation systems where fabricated parts disallow internal alterations. Using an external offset force to reduce backlash allows for part wear and non-constant component gaps (e.g., spacing between gears that varies slightly as they turn), but requires significantly large forces to do so. If the external force is too small, the actuator may be able to briefly push back into the gap. Using an excessively large external force may overload the motor and reduce controllability.

Computational modeling can also be used to compensate for backlash. While directly eliminating mechanical play is preferable, a good model and control scheme that incorporates backlash can often produce sufficient controllability and precision while remaining mechanically simple and low-cost. Both adaptive backlash compensation[19] and specialized PID control architectures[20] have been used and shown to be successful at computationally modeling and compensating for the nonlinearities of backlash. However, the extensive sensing, large number of carefully chosen parameters, and additional complexity needed to apply these methods make them undesirable in situations where cost and simplicity of implementation are important. A less accurate, but simpler and more easily implementable technique is to use a feed-forward approach.

As opposed to feedback systems which observe the output of a system and modify its input accordingly, feed-forward systems do not observe the output but instead predict what the output is likely to be and modify the input based on that prediction.

Feed-forward systems are not as robust or accurate as feedback systems, but their ability to improve the performance of a system without additional sensing makes them useful. In particular, backlash between gears, slides, and other mechanisms is often difficult to accurately measure and a feed-forward approach may be the only form of computational compensation that can be easily used.

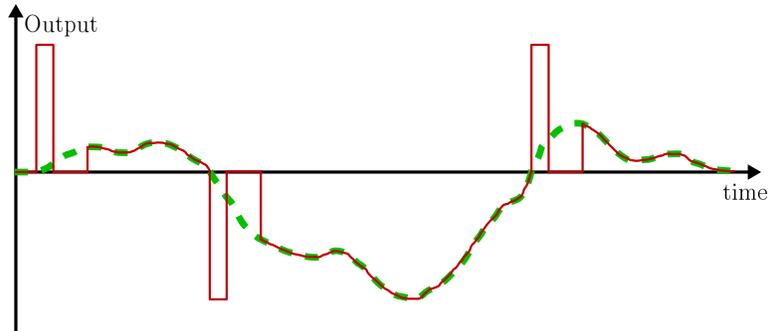


Figure 3-18: Simple feed-forward backlash model. **Dotted Green:** PID controller output signal. **Solid Red:** Actuator motor control signal. The motor is sent a fixed-width pulse and brake signal to compensate for backlash each time the PID controller output signal changes direction.

Figure 3-18 shows a simple feed-forward backlash model I developed and applied to a PID control signal. Ideally, when the controller switches direction, the actuator will move as quickly as possible through the unconstrained space, but arrive with a velocity equal to that of the driven component. Without imposing this restriction on actuator velocity, the actuator's large momentum will be transferred to the driven component upon arriving at the end of the unconstrained space and cause an abrupt acceleration most likely leading to controllability problems. The feed-forward backlash model in Figure 3-18 follows the PID control signal unless a change in direction is observed at which point a short fixed-width pulse and braking signal are sent to the actuator to quickly drive it across the unconstrained space but arrive at the other side of the driven component with negligible velocity. The pulse and brake widths can be mathematically calculated for optimal results (i.e., by using motor velocity, friction, inertia, torque, etc.), but for simple tests, values were empirically chosen by observation until operation was smooth and chirping within the drive-system was

minimized.

### 3.3.4 High-Level Models

There are many other models that can be applied to non-ideal systems to help with controllability, safety, and the reliability of operation. Regrettably, these high-level models were not explored in this research due to time constraints and the difficulties associated with testing them. None the less, they will certainly be important in the production of commercially available low-cost compliant robotic platforms and thus are briefly described here.

Three strong candidates for high-level models are part wear over time, robust fail-safe operations, and thermal modeling. Even the highest quality mechanical parts degrade over time through contact with other parts and their environment. Modeling the strains that a part undergoes over its lifetime is likely to give insight into how its performance may change. Fail-safe operations are also an important type of modeling that takes into account windows of extremes and makes sure that the system stays within those at all times. If a cable breaks and sensor readings show impossible values under normal operating conditions, a fail-safe routine can quickly act on that information to keep the robot's actions as safe as possible. Temperature also plays an important role in the way mechanical and electrical components behave. Thermal modeling of motors and compliant elements is necessary to provide robust operation over a wide range of temperatures.

In addition to the three high-level models described, an even higher level of modeling that can learn from the results of the robot's behavior and update the lower level models would be greatly beneficial. The addition of learning algorithms that can update themselves and other parameters has the potential to keep the robot functioning properly for as long as possible. High-level analysis and modeling may also be able to observe trends and predict that certain parts are likely to fail or degrade and act upon this information by shutting down the system or alerting the user.

## 3.4 Control Architecture Considerations

### 3.4.1 Data Buses

Bus	Max Bandwidth	Signals	Max Nodes	Error Checking	Complexity
<i>I</i> <sup>2</sup> <i>C</i> v2.1	3.4 Mbps	2	2 <sup>10</sup>	No	Low
SPI	10.0 Mbps	N+3	N	No	Low
CAN	1.0 Mbps	2	2 <sup>12</sup>	Yes	Med.
USB v2.0	480.0 Mbps	2	2 <sup>7</sup>	Yes	Med.
Ethernet	1000.0 Mbps	4	2 <sup>32</sup>	Yes	High
Bluetooth	3.0 Mbps	2	2 <sup>3</sup>	Yes	High

Table 3.1: Approximate characteristics of various standardized high-speed data buses.

There are many different standardized high-speed data buses to consider (see Table 3.1) when developing a robotic platform. Each bus has different strengths and weaknesses. Bluetooth, for example, allows for wireless transmission of data, but does so at the cost of high complexity and low speed. The simplest buses, like *I*<sup>2</sup>*C* and SPI, allow for better speeds but lack error checking on the hardware level, often requiring a complicated higher-level protocol to ensure that data is correct. Ethernet follows a set of robust standards and operates at very high speeds but is complex to implement and therefore is almost always used via pre-built modules and drivers. No single bus is the best choice for all robotic architectures, but rather each bus has its place depending on the scalability, complexity, datarate, and allotted development costs. An in-depth analysis of each bus is outside the scope of this exploration, but remains a critically important decision in the design and implementation of all robotic systems.

### 3.4.2 Prioritized Control

There are many different ways in which to control each actuator that is part of a robotic system including torque control, position control, and velocity control just to name a few. But with various control modes from which to choose, the way in which a mode is selected and how it transitions to and from other control modes is a very important consideration. Using a prioritized control structure (see Figure

3-19) specifies the ordering in which the control modes should be chosen dependent upon a set of constraints. For example, the first priority may be a window on the maximum allowable torques. If a measured torque exceeds the window, the controller will switch into a zero torque mode until the magnitude of the torque falls back within the window. If the first priority constraint is met, then the second priority will be checked. Suppose that the robot is capable of pushing the actuator too far and potentially hitting itself so the second priority is that of a position window that spans the allowable range of motion. If the measured angle of the arm is outside this window, then a position control mode will be used to move the arm back and keep it from damaging itself. If the second constraint is met, then a third constraint of position, torque, or velocity can also be checked. This method of priority constraint checking can continue on for many stages and is quite capable of keeping the robot arm and its surroundings safe in a wide range of situations, even if a higher-level controller is busy and unable to immediately respond.

```

if constraint_1 is not met, then use ControlMode_1;
else, if constraint_2 is not met, then use ControlMode_2;
    else, if constraint_3 is not met, then use ControlMode_3;
        ...
        ...

```

Figure 3-19: Pseudocode for a simple prioritized control routine with constraint checking.

Smoothly transitioning between control modes can be quite difficult. In particular, some constraints may conflict and oscillations are likely to result as the priority constraint checker alternates between two different control modes. Smoothly moving from one control reference to another using the same control mode (i.e. two conflicting position controllers at different priority levels) is rather straight forward and can be handled by a simple parametric weighted sum.<sup>4</sup> If the conflicting control modes, however, are of different types it is unclear what transition should be used to prevent oscillations and other undesirable controllability problems. For example, a robot arm cannot specify both position and torque since the torque is highly dependent on

---

<sup>4</sup>An example of a simple parametric weighted sum:  $Y = (1 - s)X_1 + (s)X_2$  for  $s = [0, 1]$

its location within the environment in which it is interacting.[8] The complexity of choosing PID controller gains that give desirable performance and transition smoothly between various control modes can be very difficult and remains one of the biggest drawbacks of using a prioritized control architecture.

### 3.4.3 Weighted Impedance Control

By thinking differently about the way control modes are structured and used, the problems of mode switching, conflicting constraints, and abrupt transitioning can be greatly reduced. In particular, the relationship between an actuator's velocity and force can be characterized as an impedance (i.e., a frequency dependent resistance). Similar to Ohm's Law which states that voltage is proportional to current multiplied by resistance ( $V = IR$ ), Hogan states that force is proportional to velocity multiplied by mechanical impedance ( $F = vZ$ ).[8][9][10] Conceptually, the mechanical impedance is the frequency dependent resistance to moving at a certain velocity. Moving one's arm through the air requires little force (air has a very low mechanical impedance), whereas moving one's arm through honey requires substantially more force (honey has a higher mechanical impedance than air). Trying to push a cement block along a rough table-top is harder still and can be characterized as having a high mechanical impedance. The duality between  $V = IR$  and  $F = vZ$  makes one point very clear; a robot cannot control both force and velocity since the relationship between them is dependent on the environment in which it is operating.<sup>5</sup>

The problem of mode switching that plagued the prioritized control structure can be alleviated by choosing a static control structure. One control mode can be chosen to always drive the output of the controller and other controllers can in turn drive that controller if desired (see Figure 3-20). In the case depicted in the diagram, the output is always driven from a controller in the torque mode. If position control is desired, a position gain term (in the range  $[0, 1]$ ) can be set to 1.0 and torque will be added to the torque controller's reference to push the robot towards the desired position. By

---

<sup>5</sup>This is also observed in Ohm's law. A power supply cannot set both voltage and current since the relationship between the two is dependent on the resistance of the circuit connected to it.

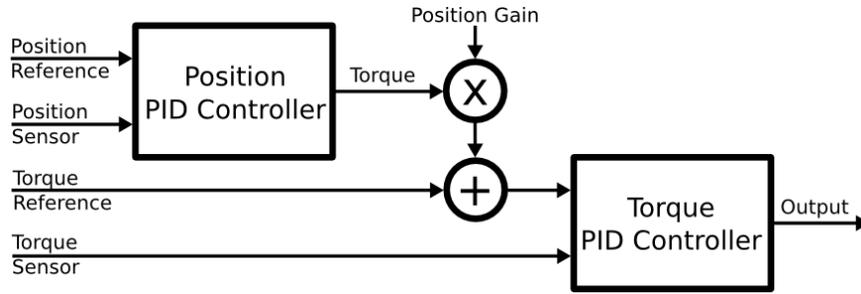


Figure 3-20: Example of a weighted impedance controller in which a static control structure has the ability to mimic multiple control modes.

smoothly changing the position gain term, the output will transition nicely between torque and position control modes. If windowing is desired, each controller block can implement such control decisions to allow for artificial walls (windowed position control) and safe operating conditions (windowed torque control).

### 3.4.4 Adaptability

Inevitably, control architectures will need to evolve as new sensors, control technologies, and modeling techniques are tested and used. While it is not the primary consideration when developing a control architecture, it is important to take into account the ease of adaptation if needed. Abstraction at each layer helps to minimize the extra work necessary to make most changes, but the overall topology, communication protocols, and hardware implementation (mechanical and electrical) are difficult to alter after a decision has been made.

Specifically, a control architecture should be chosen such that it easily scales, has the ability to be quickly updated with patches, provides an easy-to-use debugging interface, and in most cases, also allows for user-modifications to handle specialized uses and creative explorations. A full-featured set of control libraries, a well defined topological structure, and simple plug-together components are certainly a step in the right direction to allow both designers and users alike to easily adapt the robotic system. A truly user-friendly robot should also be friendly to change such that it can take on new tasks and features that may not have been considered during the initial

design of the robot.

### 3.4.5 Topologies

There are many ways that a systems-level robotic control architecture can be organized to allow for speed, robustness, low latency, centralized processing, dynamic modeling, and simplicity. The associated structures optimized for each aspect conflict with one another and therefore pose a set of trade-offs through which to navigate when choosing a topology. Higher speeds can be achieved by distributing computation among many dedicated processing blocks, but this fragmentation of data makes dynamic modeling and centralized processing difficult and time consuming. A single all-in-one processing block provides low latency and access to all sensor values for excellent dynamic modeling capabilities, but lacks processing power, does not scale well to many degrees of freedom, and is not robust in the event of failures.

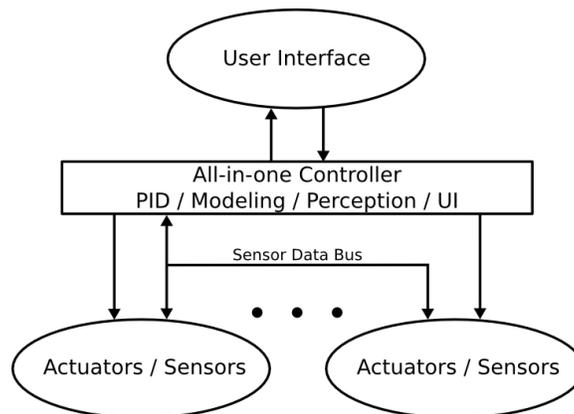


Figure 3-21: All-in-one control architecture.

Figure 3-21 shows a simple all-in-one control architecture. A single processing block handles the reading of sensor information via a data bus, all modeling of components, a PID controller, and a set of outputs to drive an actuator(s). All user inputs and information displays are also handled by the controller. This simple architecture works very well for a small number of degrees of freedom with simple modeling and control routines but slows to a crawl with more complex component modeling, additional actuators and sensors, and time consuming user interface routines that are

often necessary to display or acquire information from the operator. The centralization of all data is very convenient for dynamic modeling, but with such limited computational power, complex dynamic modeling is rarely obtainable at reasonably fast controller speeds.

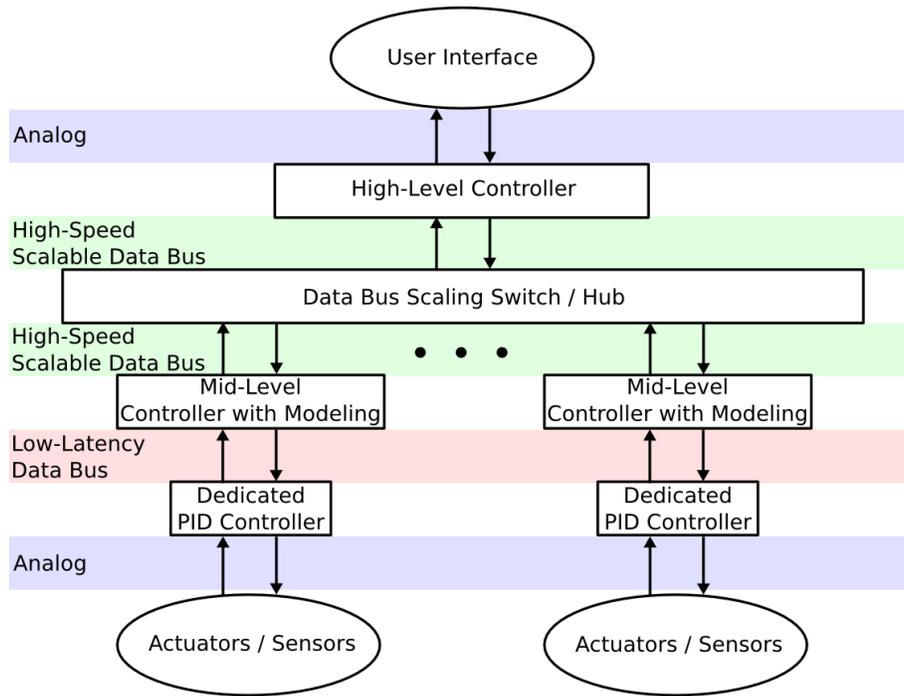


Figure 3-22: Hierarchical control architecture.

In contrast to the all-in-one control architecture, a well designed hierarchical architecture (see Figure 3-22) can solve many of the bottlenecks and problems previously mentioned but at the cost of additional complexity and latency. A dedicated PID controller handles a small number of actuators and sensors very efficiently and robustly. Limits on the maximum force allowed or range of motion that an actuator can travel can be included to reduce the likelihood of hurting itself or others if an unrealistic force or position command is given to it. Because the PID controller is a specialized computational block, it can be highly optimized for speed and robustness. Modeling that must be done quickly to be effective, like hysteresis removal and backlash compensation, should also be handled by the low-level controller, but parameters for the various models can be dynamically updated from higher-level controllers as needed.

Since the low-level controller only senses and drives a small number of actuators, it can also be placed physically close to them, allowing for a better signal to noise ratio with all electrical signals.

The mid-level controller and modeling unit provides two very important functions; running more complicated and processor intensive models specific to the degrees of freedom being controlled and bridging the gap between the low and high-level controllers via a high-speed networking interface. Without the time constraints that are normally imposed on the PID controller, the mid-level block is able to perform the majority of high-level modeling by requesting data from the low-level controller as needed and then updating its parameters, gains, and other settings. The addition of a mid-level controller also provides a level of abstraction, allowing for the subdivision of tasks which has been shown to be quite beneficial for many robotic control architectures.[3] Ethernet, a good candidate for the high-speed networking interface, is a very robust, efficient, and well standardized bus, but its bursty nature and substantial overhead in implementation requires a large number of resources to operate it. In particular, sizable buffers and state-dependent connection information must be used to ensure reliable data transfer. Allowing the low-level controller to communicate over a simple data bus is important for its fast operation but can severely limit its scalability. Having a mid-level controller allows for both optimization of the PID controller and robust high-speed connectivity among many degrees of freedom.

The high-level controller sits atop all of the other controllers as the aggregator of the system's important states and sends commands and requests to accomplish tasks. With all of the robot's important information held locally, the high-level controller can perceive the world from certain characteristics in the sensor data and act upon them accordingly. For example, if the robot's arm is blocked from moving to a certain location, the high-level controller may perceive that as an obstruction and alter the path the arm takes to reach its destination. If unexpected configurations or excessive forces are observed, the high-level controller may shutdown or reconfigure the corresponding mid-level controllers to keep interactions safe.



# Chapter 4

## Building Proof-of-Concept Systems

Making dozens of prototypes is a great way to get started with new ideas, but prototypes alone do not tell the whole story. Oftentimes unexpected interactions between parts can change the overall behavior of a system beyond what can be observed by examining the individual parts alone. Other problems may also arise such as the inability to scale an idea, random noise or crosstalk between electrical components, mechanical part wear over time, and unexpected failure modes, just to name a few. It is for these reasons that, even at an exploratory level, it is important to build working proof-of-concept systems.

### 4.1 1 DOF TCEJ Arm

#### 4.1.1 Design Overview

The first proof-of-concept system was built to test a Torsionally Compliant Elastomer Joint (TCEJ) and a corresponding hysteresis model (see Figure 4-1). Since this version of the arm was not intended to scale, an all-in-one controller architecture was chosen to reduce the implementation time. A simple arm was fabricated out of aluminum stock and assembled in a block-construction manner (right angle machining, simple mechanical fasteners, stock dimensions where possible). Torque sensing of the TCEJ was achieved via a differential pair of hall-effect sensors and an

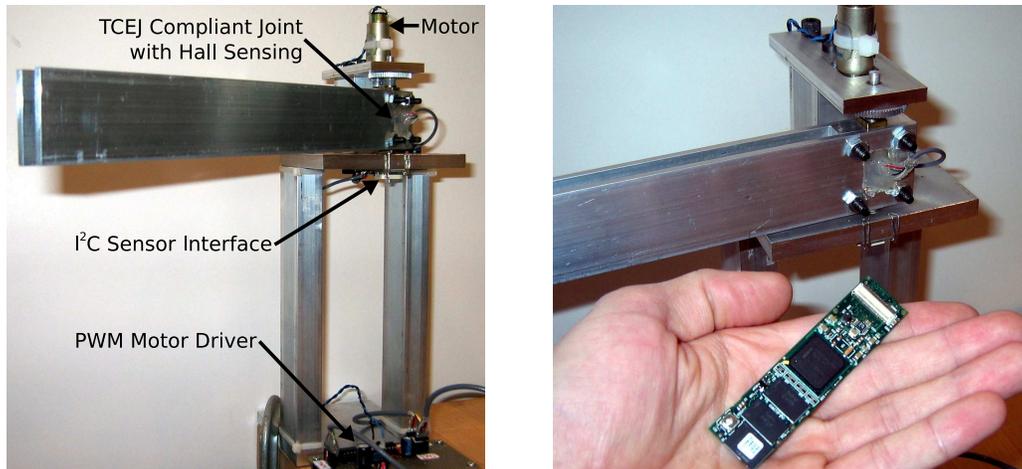


Figure 4-1: Proof-of-Concept 1 – One degree of freedom robot arm using the  $I^2C$  data bus and a TCEJ with hall-effect sensing. **Left:** The arm and its various components. **Right:** Side view of the arm and its controller, a Gumstix (32-bit, 400MHz, Linux,  $\approx$  \$100).

embedded magnet, while angular position measurements were made with a rotary potentiometer. Since the analog hall-effect signal levels were relatively small, the sensor interface board also included an op-amp with adjustable potentiometers to specify the gain and offset of the amplifier. The actuator, a precision MicroMo with anti-backlash gearhead, was used for its nice performance and size, but abandoned in future proof-of-concept systems due to its high cost. A PID controller was implemented on a Gumstix<sup>1</sup> with two control modes; torque and position. The control mode was configured in code and not dynamically changed during operation.

### 4.1.2 Architecture

The control architecture of the first proof-of-concept system (see Figure 4-2) consisted of a single level all-in-one controller; a Gumstix. This controller requested and received all sensor input data over an  $I^2C$  bus and controlled a MicroMo DC motor via a Pulse Width Modulation (PWM) output. The user interface was viewed over a terminal connection sent via Ethernet (allowing variables to be printed and modes to be changed) and data logging was possible by storing the desired information locally

<sup>1</sup>The Gumstix is a low-cost, small form-fact 32-bit Linux computer running at 400MHz.

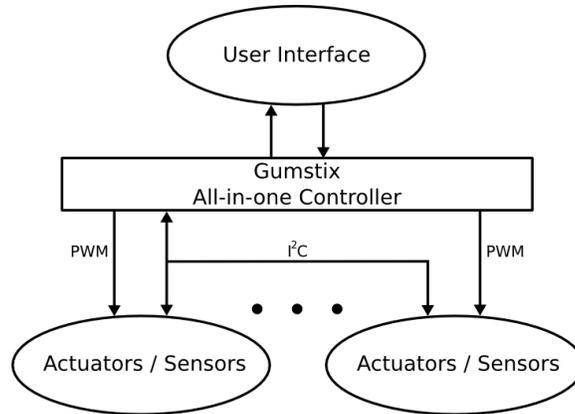


Figure 4-2: Proof-of-Concept 1 – All-in-one control architecture.

in a file. This allowed any networked computer to connect to the Gumstix, enable or disable the robot, and view its current state. All PID control and modeling of the TCEJ’s hysteresis was also handled by the Gumstix (in C code, but structured identically to the Matlab code discussed earlier in Section 3.3.1) and processed as quickly as possible. With all parts of the controller active (PID control, modeling, terminal user interface, and data logging), the control frequency was limited to  $\approx 280Hz$ . Since the control frequency of an all-in-one controller scales linearly with the number of degrees of freedom, controlling four degrees of freedom would limit the frequency to  $\approx 70Hz$ , much too slow for most robotic systems. A 400MHz processor should be capable of significantly better performance than what was observed, but limitations due to the kernel process switching time and heavy-weight communications libraries created a severe bottleneck due to the particular implementation of the controller.

### 4.1.3 Mechanical Hardware

The mechanical structure for the first system was very simplistic and designed to test a one degree of freedom (1 DOF) arm. Built entirely out of machined blocks of aluminum and steel bolts, the layout of the structure was easy to modify with various mounting plates and spacing hardware while the base could be securely clamped to a workbench.

The joint being tested, a Torsionally Compliant Elastomer Joint (TCEJ) shown

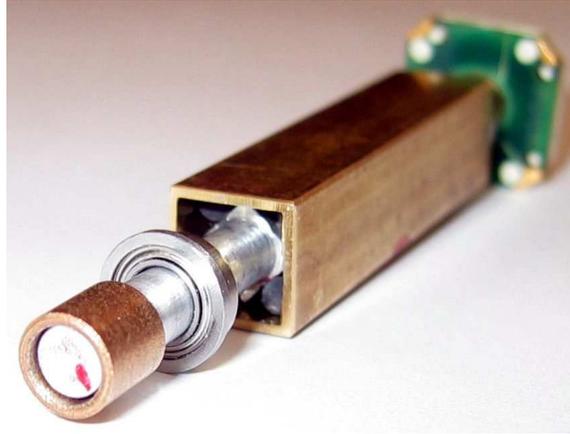


Figure 4-3: Proof-of-Concept 1 – TCEJ isometric view. **Parts from left to right:** oil-impregnated bushing, small ball bearing, TCEJ with embedded magnet, potentiometer for measuring angular displacement.

in Figure 4-3, was constructed from non-ferrous metals (brass and aluminum) and allowed for torque sensing via an embedded magnet. The square outer shaft of the TCEJ was sandwiched in the arm and held in place by four bolts. The inner shaft of the TCEJ was supported by two inexpensive oil-impregnated bushings mounted in the base and was driven by a gear coupled to a vertically mounted MicroMo DC anti-backlash gearhead motor. A pair of small ball bearings were also placed on either end of the TCEJ (between the inner and outer shafts) to restrict movement in the non-rotational directions.

#### 4.1.4 Electrical Hardware

To allow for remote sensing of multiple analog signals over a simple 2 wire interface, a multiple-input high-speed Analog to Digital Converter (ADC) was used via the  $I^2C$  data bus. A PCB was created to wire together the ADC, power regulator, sensors, differential amplifier (for torque sensing of the TCEJ), and connectors. The circuit board was then mounted directly to the bottom of the joint to obtain angular position information via a potentiometer and wired to two hall-effect sensors for torque measurements. Hot-melt glue was used as strain relief around all connectors, easily increasing the systems robustness against mechanical fatigue at the cost of

degraded aesthetics.

The power required to drive the MicroMo DC gearhead motor was substantially higher than what could be produced from the Gumstix directly and thus a motor driver circuit was necessary to isolate and buffer the PWM output signal. A simple high-current push-pull amplifier (specifically the L293) and a bit of logic was connected together to take in two signals, *PWM* and *Direction*, and drive the motor accordingly. A pair of LEDs were also included to notify the user if voltage spikes occurred in either direction. The MicroMo DC gearhead motor ( $\approx$  \$300) worked very well with the driver circuit, but was later replaced (see proof-of-concept 2 and 3) due to its extremely high cost.

#### 4.1.5 Software

Four important pieces of functionality were needed on the software level for the Gumstix to perform its role as an all-in-one controller. The first and most fundamental aspect of the controller was a feedback-driven PID control loop. The PID controller took, as input, a reference value and a sensor value, calculated the difference between the two inputs (the *error* signal), then produced an output that was the sum of the Proportional, Integral, and Derivative components of that difference. But since the TCEJ was substantially hysteretic, a second function, modeling of the sensor's value, needed to be performed before it could be used as an input to the PID controller. The hysteresis model implemented was of the form discussed earlier in Section 3.3.1 in which a simple mapping between elastic deformation, torque, and plastic deformation was used. The third important piece of functionality enabled the Gumstix's  $I^2C$  and *PWM* capabilities necessary to communicate with sensors and the motor-driver. Both of these protocols were supported in hardware by the Intel PXA255 processor (the CPU of the Gumstix), but drivers were not available and therefore were created to easily allow for their use in code. Finally, functionality was added to allow for raw, modeled, and output signal values to be printed to the console in a standardized format such that they could be saved to a file and opened for post-processing and analysis.

All of the software used on the Gumstix was written and compiled on a PC in the C programming language and then copied to the Gumstix over SSH. The ability to easily move information back and forth between the controller and a PC proved to be extremely helpful. Because the Gumstix was always connected to the network and ran a simple web-server, its status could be checked remotely at any time as well. Admittedly, Ethernet seems a bit overkill for simple data transmission to a robotic controller, but the ease of use, speed of data transfer, compatibility, and scalability made it ideal for making quick changes during the developmental process.

An experiment was constructed to more accurately characterize the TCEJ and its associated models. A more in-depth explanation and the experimental results are given in Section 5.1.

## 4.2 2 DOF Inexpensive TCEJ Arm

### 4.2.1 Design Overview

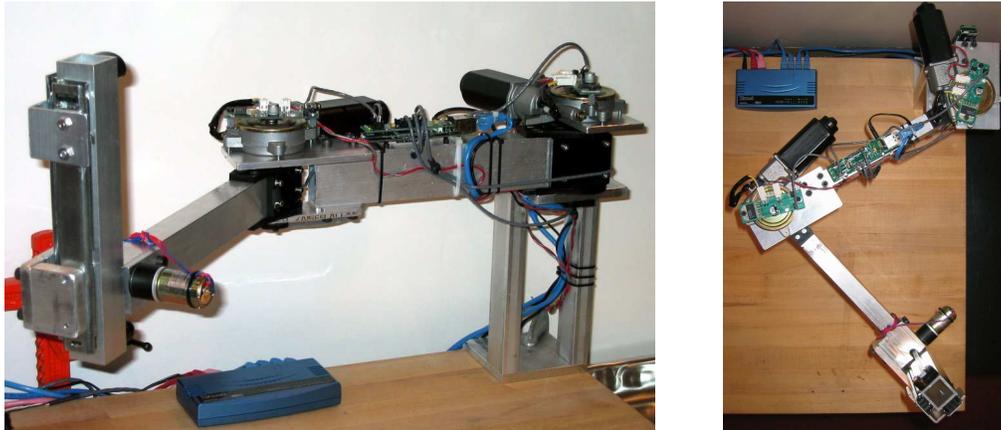


Figure 4-4: Proof-of-Concept 2 – Two TCEJs, a hierarchical control architecture with SPI/Ethernet, and an unactuated linear z-axis slide. **Left:** Isometric view. **Right:** Top view.

The second proof-of-concept (see Figure 4-4) was chosen to demonstrate various ideas for low-cost compliant robotics by using a planar two degree of freedom (2-DOF)

arm similar to the SCARA<sup>2</sup> topology. Each joint consisted of a Torsionally Compliant Elastomer Joint (TCEJ) with embedded differential force sensing resistors and was driven by a very low-cost automobile window lift motor with internal gear reduction (*ratio* =31.5:1). A hierarchical impedance control architecture was used to allow for fast low-level control while still permitting higher-level modeling, perception, and user interfaces. A third degree of freedom, the z-axis, was constructed at the end of the arm but was not electrically controlled due to time constraints. It did however serve an important role as a large mass to highlight the need for a model of the system’s dynamics in the controller to obtain greater stability.

#### 4.2.2 Architecture

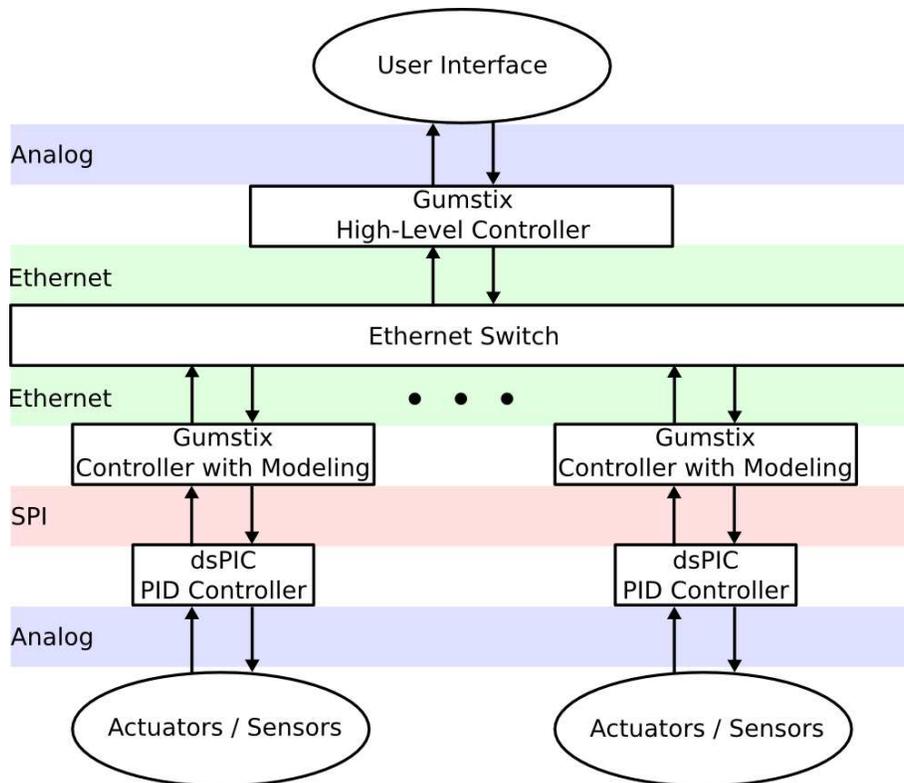


Figure 4-5: Proof-of-Concept 2 – Hierarchical control architecture.

<sup>2</sup>SCARA: Selective Compliant Assembly Robot Arm. SCARA robots typically consist of a 2-DOF planar arm with a combination vertical slide and rotary mechanism at its end to give a total of four degrees of freedom.

To allow for scalability, speed, and robustness, a hierarchical control architecture was implemented in the second proof-of-concept system (see Figure 4-5). Due to its ease of use, low-cost, and availability of interface boards to the Gumstix, Ethernet was chosen as the scalable high-speed data bus. However, Ethernet is a packet-based best-effort network with the potential for high latency and dropped packets, serving its role nicely to quickly move data between various parts of the robot, but unacceptable for directly controlling and sensing actuators. Gumstix systems were used for both the high-level controller and the mid-level controllers due to their extraordinary processing power, small form-factor, low-cost, and simple interface to Ethernet.

The data bus connecting the mid-level controller to the low-level PID controller did not need to be highly scalable or excessively robust, but it was important that the bus had very low latency, reasonable data rate, and could be easily implemented in code and hardware<sup>3</sup>. The mid-level controller was free to communicate with the high-level controller (to receive commands and report controller information) and then act on those commands by sending appropriate messages to the low-level controller. Various kinds of modeling, interpolation, and logging was also performed at the mid-level controller to linearize components, increase robustness to failure, and record large amounts of important data. The PID controller could therefore remain dedicated to precisely sensing and controlling the actuator(s) dependent upon higher-level parameters and reference values passed to it by the mid-level controller.

### 4.2.3 Mechanical Hardware

The second proof-of-concept system allowed for compliance by using TCEJs similar to those used in the first system, but with torque sensing via Force Sensing Resistors (FSR) rather than hall-effect sensors. FSRs were chosen for two reasons: (1) they scale nicely with larger joints whereas magnetic sensing becomes more prone to noise and (2) they have the potential to be less hysteretic by only sensing orthogonal

---

<sup>3</sup>The PID controller is likely to be quite limited in terms of processing power and memory compared to the Gumstix, thus requiring that the data bus be simple to implement and not require substantial data buffers or computational overhead.



Figure 4-6: Proof-of-Concept 2 – TCEJ mounted on a window lift motor.

force instead of the angular difference between the inner and outer shafts. An automobile window lift motor ( $\approx$  \$30, 10x less expensive than the MicroMo DC motor used previously) was used as the actuator for each joint and showcased the need for computational models to compensate for backlash, overheating, and dynamics. Each motor was mounted directly to its corresponding TCEJ, as shown in Figure 4-6, to eliminate the need for cables or external gearing that can quickly lead to higher costs due to additional complexity and alignment precision.

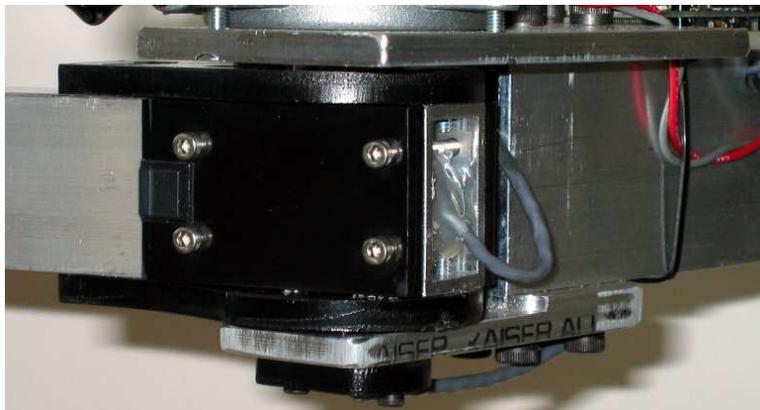


Figure 4-7: Proof-of-Concept 2 – Aluminum and lasercut ABS Joint.

To speed up the development time of the second system's hardware, the aluminum base of the first proof-of-concept system was used by simply modifying it to fit a larger

arm and motor. Similar block-construction techniques were used throughout the arm where possible to easily allow for modifications without the need for substantial machining or specialty parts. There were a few parts, however, where rounded shapes were necessary to press in bearings or reduce the likelihood of pinching wires around turning parts. By carefully choosing which parts were most important structurally and those which required rounded features, both stock aluminum tubing and ABS plastic could be used. ABS has substantially less tensile strength than aluminum, but its ability to be lasercut allows for complex two dimensional features to be realized in only a few minutes. The ABS and aluminum pieces were held together by small bolts and could be easily taken apart for modifications (see Figure 4-7).

At the end of the two degree of freedom arm, a linearly traversing degree of freedom was constructed (but was not actually wired for control due to time constraints) to enable vertical movement of the end-effector. This z-axis slide was designed with cost in mind by choosing mass-produced precision slide components intended for another market; desk drawer slides. By mounting two slides perpendicular to one another, smooth sliding action with very little play was achieved. A rack and pinion was originally chosen as the rotational-to-linear conversion mechanism, but a cable driven design may have been a better choice to eliminate the cogging between gears and allow the mass of the motor to be placed closer to the base of the arm.

#### 4.2.4 Electrical Hardware

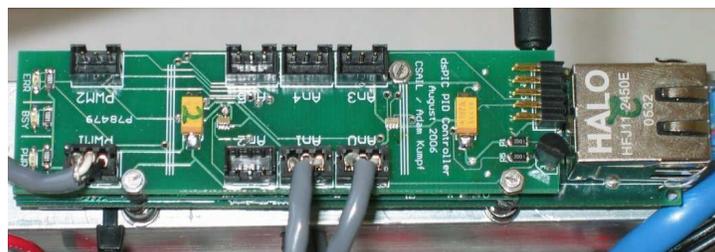


Figure 4-8: Proof-of-Concept 2 – PCB: DSP daughter board mounted on the Gumstix.

The main electrical component created for the second proof-of-concept was a

daughter board<sup>4</sup> that connected directly to the top of the Gumstix and contained a Digital Signal Processor (DSP)<sup>5</sup> to run a dedicated PID controller (see Figure 4-8). Communication between the daughter board and the Gumstix was achieved via SPI data bus pins within the connector. The dimensions of the daughter board were chosen to fit the profile of the Gumstix, leaving the overall size small and allowing for simple through-hole mounting techniques. Two large value capacitors were placed on either end of the PCB to reduce digital switching noise coming from the DSP and the Gumstix. The daughter board allowed for six analog sensor inputs and two Pulse Width Modulation (PWM) signal outputs with direction control. Ideally, programming the DSP could be done via the SPI data bus, but the additional overhead to implement this feature proved to be quite large so an external connector was included to program the DSP directly via a standard programming tool instead.

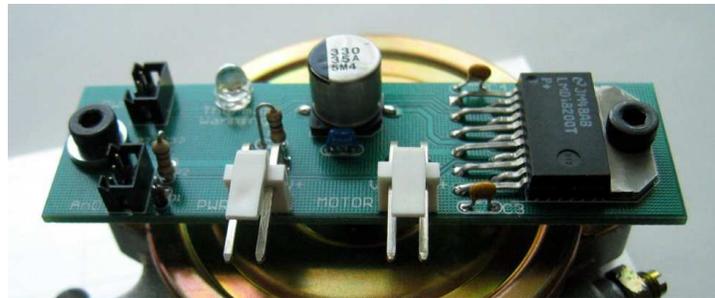


Figure 4-9: Proof-of-Concept 2 – PCB: PWM H-Bridge mounted on the window lift motor.

To drive the inexpensive automotive window lift motors, a high-current motor driver was necessary. A simple PCB, shown in Figure 4-9, was created to allow for an H-Bridge (LMD18200T) and its associated circuitry to easily interface with a PWM output, for motor speed and direction, and an analog input, for current sensing, to the daughter board. An LED was also included to alert the user if the motor driver was getting too warm and potentially nearing its shutdown temperature. The H-Bridge chip provided most of this functionality with little additional circuitry so the PCB

---

<sup>4</sup>A daughter board is a circuit board, usually with a specific set of supplementary capabilities, that directly attaches to another circuit board.

<sup>5</sup>The particular DSP used was the 16-bit, 30MHz dsPIC30F4012 from Microchip.

was quite simple with a low number of components. The size of the PCB was designed to bolt directly to the back of the window lift motor using its existing threaded holes.

### 4.2.5 Software

A substantial amount of code was needed to implement the hierarchical control architecture and was developed for both Linux (running on the Gumstix) and the DSP. Software on both platforms was built with layers of abstractions where possible and special care was taken to handle and report errors over both the Ethernet and SPI data buses.

The low-level controller code running on the DSP was designed to be extremely reconfigurable and robust while maintaining a high level of performance on its most important function; PID control. The DSP sampled all six analog inputs at 10kHz, but only performed the PID controller routine at 1.25kHz (still quite fast for robotic actuator control) to allow for 8x oversampling and averaging of the inputs. These values could then be amplified and summed together in many ways (e.g., subtract two sensors and multiply by a constant for a differential pair) to generate a signal. The signal was then run through a hysteresis and compression model to account for non-idealities of the real-world system and fed into the PID controller. One last bit of modeling was also performed after the PID controller to account for backlash when the output changed directions. An SPI routine handled all incoming and outgoing communications and allowed for modeling parameters and controller gains to be dynamically updated during operation of the actuator.

What has been thus far abstracted away as a simple “PID Controller” is actually much more complex since there are multiple control modes such as torque, position, and velocity control. Initially, a simple prioritized control structure was used in which a set of constraints would be examined and the first constraint to not be satisfied would dictate which control mode to use. However, the abrupt binary transitions between control modes quickly led to instabilities at reasonable P and I gain values. Restructuring the “PID Controller” block to instead have a static form and allow for various modes via weighted impedance control proved to be a much more sensible

structure that remained significantly more stable around regions of transition.

The mid-level controller Gumstix communicated with the DSP over the SPI data bus with appropriate drivers and a high-level protocol was created to allow for reliable data transfer. Each mid-level controller was also connected to Ethernet and ran a server to allow for commands and data to be received and requested by the high-level controller. By connecting to each mid-level server as a client, the high-level controller could store and analyze the state of the entire robot to make decisions about what actions to take. Low-resolution perception of the environment was also possible at the high-level controller by monitoring torques and positions of various parts of the robot, but could likely be improved with additional sensing of sight, sound, and touch.

While the end-to-end behavior was well defined, the interactions between three controller levels, two different processing platforms, and two drastically different data buses caused a fair amount of complexity between the ends. This can be best illustrated with a simple example. Assume that the arm had been initialized and rested at  $0^\circ$  for both degrees of freedom. The high-level controller would then like to move the first degree of freedom to a new position,  $15^\circ$ . A position command would then be marshaled, sent to the Ethernet driver, directed through an Ethernet switch, and finally arrive at the mid-level controller's Ethernet interface. The mid-level controller could then unmarshal the packet, check it for errors, handle any necessary modeling, and then rebuild the command to be sent over the SPI data bus to the low-level DSP PID controller. The DSP would wait for the data to completely arrive, check it for errors, and then update the position reference within the PID controller to move the arm to the new location. Finding a way to eliminate one or more of the steps involved in sending and receiving end-to-end commands while maintaining robustness would allow for a faster response from the high-level controller and even greater flexibility. This suggests that for a small number of degrees of freedom per mid-level controller, it may be beneficial to implement the PID controller on the Gumstix rather than the daughter board DSP.

The processing power of the Gumstix alone (400MHz, 32-bit) should be able to handle the bulk of the computational modeling and information routing without

being heavily loaded, but due to the use of pre-built modules and drivers, the actual capabilities were substantially limited. Perhaps the most significant bottleneck was the kernel process switching time which was manually configured as suggested in the Gumstix support literature, but was still not able to switch between processes faster than 10mSec. To improve this systems-level software architecture problem, running real-time linux may greatly reduce the process switching time and allow for less latency and, in turn, smoother controllability.

Experimental data for the second proof-of-concept robot arm that illustrates the limitations of its controllability and discusses some implications and potential improvements can be found in Section 5.2.

### 4.3 4 DOF Compliant Drive-System Arm



Figure 4-10: Proof-of-Concept 3 – Four degree of freedom arm with drive-system compliance, a hierarchical control architecture, and a servo-mounted paint roller. **Left:** Front angle view. **Right:** Front view.

#### 4.3.1 Design Overview

The third proof-of-concept system was chosen to demonstrate other various ideas for low-cost compliant robotics by using a four degree of freedom (4-DOF) arm similar to the SCARA topology, but designed with the task of painting in mind. To limit the inertial effects of the arm, inexpensive carbon-fiber tubes (ski poles) were used for construction and two inexpensive window lift motors were mounted at its base. A

compliant drive-system was created by using amorphous nylon (200-pound-test fishing line) for the cabling and significantly pre-loading it by using embedded winding posts in the actuator hubs. Torque sensing was achieved by measuring the angular difference between corresponding hubs (via rotary potentiometers) and angular position measurements were sensed using each joint's potentiometer alone. The same hierarchical control architecture that was used with the second proof-of-concept system was also used in this system. Two servos mounted at the end of the arm were employed to orient and lift a small paint roller chosen to perform a simple painting task. Turning the servos to the off state allowed them to be easily back-driven and served as a compliant (although, not sensed) mode.

### **4.3.2 Architecture**

The architecture for the third proof-of-concept was identical to the hierarchical version used in the second system (see Figure 4-5). While the choice of implementations, components, and buses was not optimal and could benefit from redesign, the associated developmental time and costs could not justify substantial changes in such a short exploration. Since the third systems was designed mainly as a test platform for inherent drive-system compliance, the focus was placed on modeling the robot arm's non-ideal behaviors and testing higher level perception to allow the arm to dynamically adapt to its environment while performing a task.

### **4.3.3 Mechanical Hardware**

The hardware used for the third robot arm was substantially different than the other two built before it, intentionally exposing more of the design space for low-cost compliant robotics. In particular, a cable driven approach was desired since many existing compliant robots use cables in some form to reduce inertia by moving the heavy actuators closer to the base of the arm. There were many options as to where the compliant element could be placed within a cable driven system, but making the cable itself compliant seemed elegant, simple, and easily transferable to other well

studied robotic designs.



Figure 4-11: Proof-of-Concept 3 – Clamps used to attach carbon-fiber to aluminum.

The structural support for the arm came from two important materials; low-cost carbon-fiber tubing and aluminum. Carbon-fiber is amazingly strong and lightweight, but due to its fibrous nature, cannot be easily machined to other shapes or secured with standard through-hole fasteners (bolts, screws, pins, etc.). Instead, carbon-fiber components are typically attached to other parts of a structure by clamps or epoxy. As shown in Figure 4-11, a simple clamping mechanism was employed extensively throughout the design using half of a 2-piece shaft collar, two small bolts, and a machined recess into an aluminum part. The combination of aluminum and carbon-fiber for the structure of the robot arm made for a very lightweight and strong design that could be taken apart by removing just a few bolts.

Inherent drive-system compliance was created by using amorphous nylon (200-pound-test fishing line) instead of steel cabling. As discussed in Section 3.2.3, tensioning the stretchy cable is very important and was achieved by built-in winding mechanisms housed in each actuator's hub. Well-centered and carefully machined hubs were also important to minimize false sensing of the cable's stretch throughout the robot arm's full range of motion.

Two inexpensive servos were mounted in series at the end of the arm to enable lifting and rotation of a small paint roller. To aid the servos' operations with a fairly heavy roller, a counterbalance and an extension spring were used to roughly

compensate for gravity and allow the servos to be actuated throughout their entire operating range. The paint roller, a three inch long soft-covered cylinder, was then capable of rolling flat along the surface of the workspace and also vertically along the wall of large objects. These two actions were later used to dip the roller in a tray of paint and then proceed to paint the side of an object within its workspace.

#### **4.3.4 Electrical Hardware**

The same printed circuit boards and electrical components that were used in the second proof-of-concept were also used in the third system with a small difference occurring only in the method of sensing torque. Instead of using force sensing resistors as was the case with the TCEJ, torque was measured by calculating the angular difference between the corresponding actuator and joint hubs. This could be easily sensed with rotary potentiometers and therefore did not require specialized torque sensing hardware.

#### **4.3.5 Software**

The third proof-of-concept offered two new challenges in terms of software; calculating joint torques dependent on cable stretch and performing high-level tasks to paint an object. Once the nylon cabling was sufficiently pre-loaded and bound to each hub to prevent slippage, the torque on each joint could be obtained by sensing the difference in angular displacement between the hubs over which the cable traveled and multiplying those angles by a constant of proportionality. This angular difference could then be mapped to the joint torque by taking into account the cable's elongation and stiffness parameters. In the simple case of modeling the stretchy nylon line as an ideal spring, a linear mapping could be used and was implemented in the third proof-of-concept. The sensed torque resolution was relatively poor compared to the TCEJ, and due to small mechanical imperfections in the hubs, the precision varied slightly throughout the arm's range of motion.

The second software challenge revolved around the creation and execution of high-

level tasks that were not merely forward running scripts, but rather processes which sensed the environment and adjusted to it accordingly. A library of simple operations was created to abstract away their internal complexity and allow for intuitive construction of dynamic tasks. If a sub-task failed, that information would be conveyed to the system and allow for branching to a new task if desired. Due to the abundance of the robot's global system state at the high-level controller, tasks could be flexible and sense a multitude of states to detect events such as the collision with an object or the guidance of a person holding the arm. Behavior based modeling and perception are very important features of robust robotic systems, but their implementation complexity made them difficult to fully realize on the proof-of-concept level. A fairly simple task of painting an object with the compliant arm and paint roller was constructed and tested (see Section 5.3 for more details), but a richer set of commands, sensors, and perceptive tools would likely enable the robot to be more generalizable and competent over a wider range of tasks.

# Chapter 5

## Experiments

Throughout the design and proof-of-concept phases, numerous experiments were performed to gauge the strength, stability, feasibility, and performance of components and systems. Most were simple and subjective; bending a part until it failed, judging stability over a series of inputs, checking tolerances with dial calipers, and testing models for improved behavior. Although it seems flawed to move quickly between designs without exact testing methods and data, the nature of rapid prototyping and iterative design allows for much greater exposure and synthesis when the majority of the development time is not spent on the necessary setup and acquisition associated with precision tests.

However, once an entire system was built and operational, more concrete measurements and tasks were performed for analysis and comparison. Three of those experiments were very important in this exploration and they are presented in detail throughout this chapter along with their results and implications. Each test was specifically chosen to shed light on some of the key capabilities and drawbacks of the aforementioned compliant mechanisms and computational models.

## 5.1 Characteristics of TCEJ Hall-Effect Torque Sensing

### 5.1.1 Description

The Torsionally Compliant Elastomer Joint (TCEJ) is a very simple series compliant robotic joint with integrated torque sensing. However, due to the hysteresis of the compliant elements (e.g., buna-n, latex, silicone) and the nature of the integrated sensing, characterization of the TCEJ was an important step towards being able to use it effectively in robotic systems.

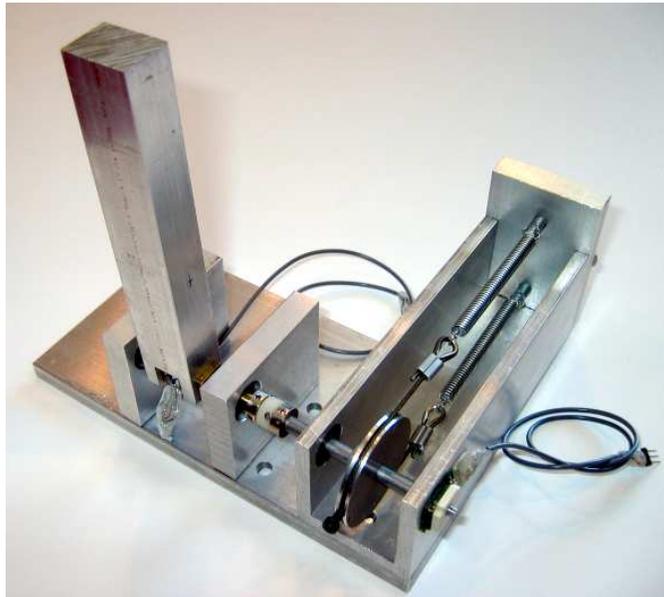


Figure 5-1: TCEJ testing device.

A testing device was designed and built to allow for characterization of the TCEJ and is shown in Figure 5-1. By attaching a pair of extension springs to a cable and wrapping the cable around a hub, a torque proportional to angular displacement was produced and measured via a rotary potentiometer. The output of the TCEJ's differential hall-effect sensors was then compared to the actual torque (proportional to angular displacement via Hooke's Law) to analyze the hysteresis and linearity of the joint. The arm, connected to the outer shaft of the TCEJ, was moved rela-

tively slowly to minimize higher-order inertial effects during the tests. The particular TCEJ configuration tested was the same as the one used in the first proof-of-concept arm. It consisted of 4 cylindrical Buna-N elastomers to achieve compliance and used integrated magnet and hall-effect torque sensing (see Figure 3-4:**Left**).

### 5.1.2 Results

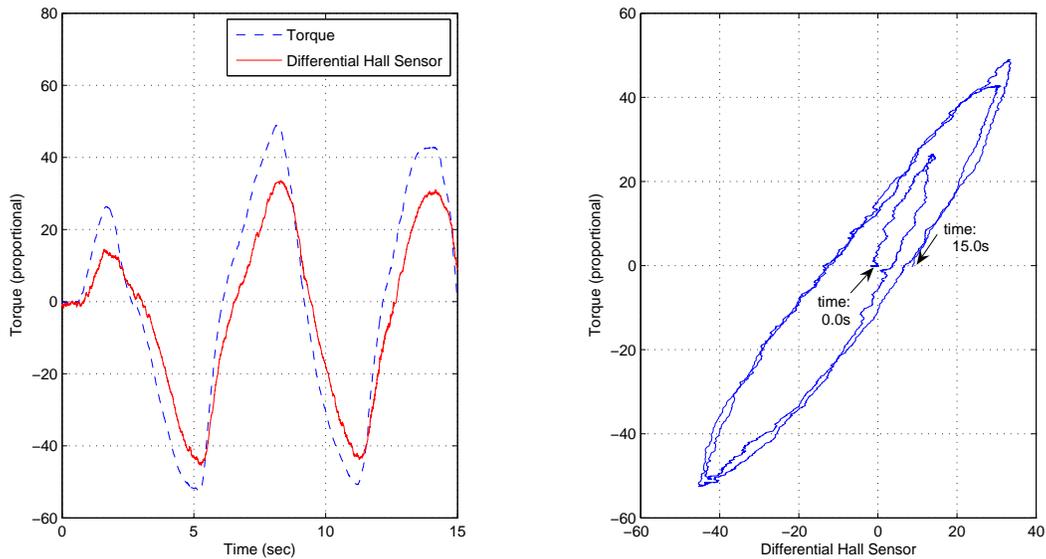


Figure 5-2: TCEJ characteristic test data. **Left**: Torque and differential hall sensor values over time. **Right**: Torque and hall sensor values plotted with respect to one another over time. Notice the material’s substantial hysteresis and how it caused the signal to no longer pass through the origin after the initial torque was applied.

Experimental results of the tests used to characterize the TCEJ are shown in Figure 5-2. The graph on the left shows two signals (the actual torque and the hall-effect sensor value) over a time of 15.0 seconds. The difference in amplitude between the two signals is not important since either can be linearly scaled, but the phase shift between the two is very significant and leads to the hysteretic behavior observed in the graph on the right. The right graph shows the two signals plotted with respect to one another over time. As the elastomer is deformed from its rest position, it no longer passes through the origin as the torque is varied due to hysteresis within the

material. It is also worth noting that the sensor output is very close to linear (i.e., the shape of the curve if the gap caused by the hysteresis was removed) over the range of torques produced during this experiment.

### 5.1.3 Implications

This experiment shows that hysteresis of the TCEJ with integrated hall-effect sensing can be quite substantial and must be accounted for to enable accurate control. In particular, the hall-effect sensor offset at zero divided by the peak torque amplitude that caused the offset is  $\approx 1/3$  and will depend strongly on a hysteresis model to compensate for the large fluctuations in plastic deformation. The first proof-of-concept system made use of this data during the configuration of its hysteresis model and proved to be quite effective, as shown previously in Figure 3-14.

Another important implication from the data collected in this experiment is that sensor values are repeatable (i.e., they re-trace the same path when periodically loaded) and reasonably linear. The fairly linear nature of the device was an unexpected result since two very non-linear elements, magnetic position sensing ( $\propto 1/r^2$ ) and the angular displacement to torque mapping ( $\propto \tan(2\theta)$ ), were integral parts of the TCEJ. While the testing device did not allow for extremely large torques to be applied to the joint, it is worth noting that the asymptotic behavior of the TCEJ's square-within-a-square design will eventually give rise to decreased sensitivity (non-linearity) when large torques are applied.

## 5.2 Response of 2-DOF TCEJ Arm with FSR Torque Sensing

### 5.2.1 Description

A different behavior was observed by the TCEJ with Force Sensing Resistors (shown in Figure 3-4:Right) than what was originally characterized by the TCEJ with hall-effect sensors. Unlike the magnet and hall sensor configuration that gave rise to a

large amount of hysteresis as the elastomer plastically deformed, FSRs measure only the force orthogonal to their surface and are therefore much less affected by plastic deformation. In fact, the dominant non-linearity of the TCEJ with FSRs was not due to hysteresis, but instead occurred as jumpiness around zero as the load shifted between the two FSRs. This increased sensitivity was mitigated by the addition of a compression model and is show in the results section below.

While torque control was smooth and reliable, position control proved to be substantially more difficult. The low-cost window lift motors used to drive the arm contained substantial backlash and imposed severe limitations on the steady state-error to reduce the risk of pushing the controller towards becoming unstable. A series of incremental position steps were commanded to the actuator after the controller had been configured (i.e., the integral term was maximized to reduce steady-state error but still remain stable) to determine if additional modeling would be necessary to achieve the resolution required for simple everyday tasks.

## 5.2.2 Results

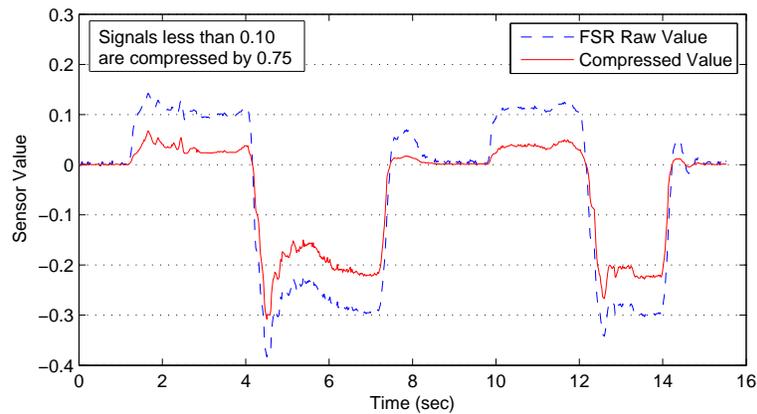


Figure 5-3: Signal compression is applied to a TCEJ with integrated FSRs to attenuate the unrealistic abrupt changes around zero. A piecewise linear compression model was used with 0.75 compression for values within the compression window of 0.1.

Figure 5-3 shows both the raw differential FSR sensor value and the compressed value over time. All sensor values that are less than the compression window of 0.1

are compressed by 0.75 (i.e., multiplied by 0.25) while values outside the window are not scaled, but instead simply shifted to avoid discontinuities between the compressed and non-compressed regions. This simple compression model greatly improved the jumpy behavior of the raw sensor value around zero to allow for more stable and accurate control.

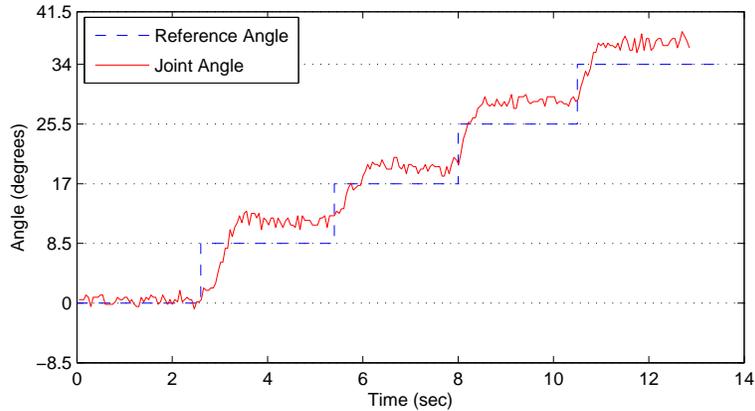


Figure 5-4: Incremental step response of a TCEJ with FSRs. Note the steady-state error of  $\approx 4^\circ$  that could not be eliminated due to backlash instabilities which resulted when using higher Integral and Proportional gains in the PID controller.

The second part of this experiment consisted of testing the obtainable accuracy of the arm’s position controller. As shown in Figure 5-4, a series of small incremental steps in position were commanded and the arm moved towards those positions accordingly. The figure clearly shows a significant amount of steady-state error that, for most robotic applications, would be too large to allow for predictable movement along a trajectory. The P and I terms of the PID controller were altered numerous times to further reduce the error, but the excessive backlash in the motor produced oscillatory behavior (audible as *chirping*) as it overcompensated for small errors until the arm eventually became unstable.

### 5.2.3 Implications

The signal compression data shows that abrupt changes can be greatly reduced with a simple piecewise linear model. However, signal compression should be approached

with caution as small values can be overly attenuated which may lead to issues regarding accuracy and controllability. If at all possible, it is probably best to physically or electrically reduce the abrupt transitions (e.g., pre-loading elastomers to a greater extent within a TCEJ) instead of relying heavily on a carefully chosen compression model.

While the resolution and accuracy of torque data has been shown to be quite reasonable when modeled correctly, the system’s ability to control position leaves much to be desired. Reducing steady-state error is very important for tasks that require placement of items or the ability to follow trajectories. Compliant robotic systems are often limited in velocity due to their low mechanical resonant frequencies, but a lack of steady-state error correction all together (i.e., it does not even occur slowly over time) is unacceptable for most applications. Since the likely cause of the instabilities associated with the higher values of gain was backlash (the play between gears within the inexpensive window lift motor), either a backlash model, a different actuator, or a mechanical backlash reduction method is probably an important consideration for future designs. This experiment demonstrated the need for a backlash model, but the model was not developed until much later and therefore could not be tested on the second proof-of-concept system within the time constraints of the exploration. The backlash model was, however, incorporated in the next experiment to aid in positioning during the painting task.

## 5.3 Compliant Painting Task

### 5.3.1 Description

To better understand the capabilities and limitations of the third proof-of-concept low-cost compliant arm, a simple high-level task was constructed to paint the side of an object. Hysteresis and backlash models were used similar to those described in the previous two sections to compensate for the non-idealities of the inexpensive window lift motor and the stretchy nylon cabling. The torque resolution was observed to be

significantly worse with the compliant cabling than what was measured with both hall-effect and FSR torque sensing in the TCEJ. This led to difficulties in measuring small torques and therefore created slightly abrupt movements for small changes in the controller's reference position.

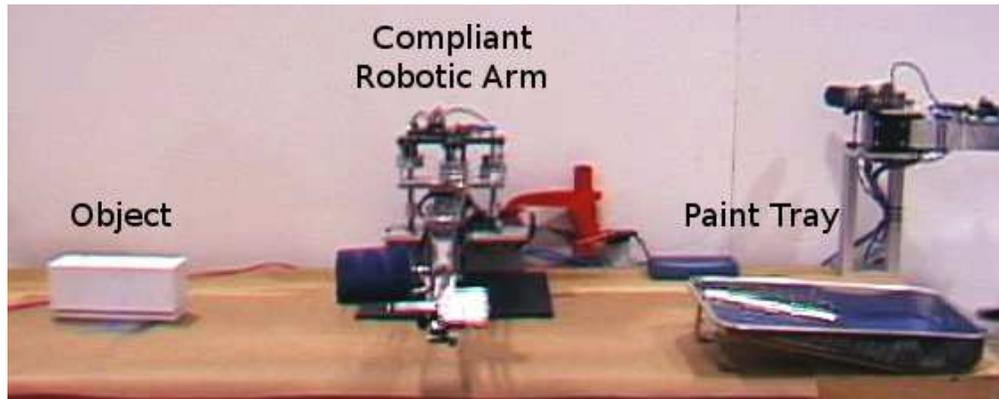


Figure 5-5: Physical layout of the workspace used during the painting task.

The high-level task of painting the side of an object (see Figure 5-5 for its physical layout) was broken down into many sub-tasks with conditionals, allowing the robot to alter its behavior depending on what it observed. The first sub-task was to orient the arm to a home location (i.e., both joints set to  $0^\circ$ ) and calibrate itself to obtain accurate torque measurements by sequentially moving the arm, averaging the sensor values, and updating the controller. The arm then moved over to a paint tray and, once there, lowered the paint roller to dip it into the paint. Then, to wipe off the excess, the paint roller servos were placed in a *pseudo-compliant* mode which allowed them to be easily back-driven as the arm pulled the roller along the incline of the paint tray. This compliant dipping and wiping sub-task was performed twice to improve the paint's coverage around the roller's perimeter. The roller was then lifted from the tray and slowly moved over to the region where the object was placed on the other side of the workbench. After lowering the roller for painting, the arm was then placed in a torque control mode and slowly ramped the output torque until either an object was detected or the range of motion suggested that no object was found. If an object was present, the arm's second joint supplied a constant torque to lightly press the

roller against the object while the first joint moved back and forth to paint it. Upon completion, the arm was placed back into a position control mode and driven to its initial home location.

### 5.3.2 Results

As shown in Figure 5-6, the painting task was successful at dipping the roller in paint, compliantly finding the object, and painting it uniformly over a wide range of object positions and rotations. In particular, the rectangular object was successfully detected and painted anywhere along a 15cm center-line with rotations ranging  $\approx \pm 20^\circ$  relative to the perpendicular intersection with the paint roller (see Figure 5-7). In cases where the object was not present, the arm successfully observed this condition and returned home appropriately. Safe human interaction was possible with the arm at all times, although external torques applied to the arm during the calibration process caused noticeable offsets in position throughout the remainder of the task.

A few other aspects of the painting task experiment are also important to note. The time to complete the task from beginning to end (calibration, dipping the roller, detecting the object, painting the object, returning home) was  $\approx 82$  seconds. This could be improved slightly by eliminating calibration and the somewhat slow detection process, but the speed of the arm's movement is somewhat limited to reduce the likelihood of paint being thrown off of the roller. There was also a small amount of jumpiness observed when the arm was making minor adjustments in the position control mode due to the fairly low resolution of the torque sensing and the lack of a precise dynamics model. Another important aspect of the task was compliant object detection. By slowly incrementing the torque, the arm applied only a small amount of force once the object was detected. This holding torque was then constantly applied during the painting process to ensure uniform roller pressure over a wide range of object angles and positions.

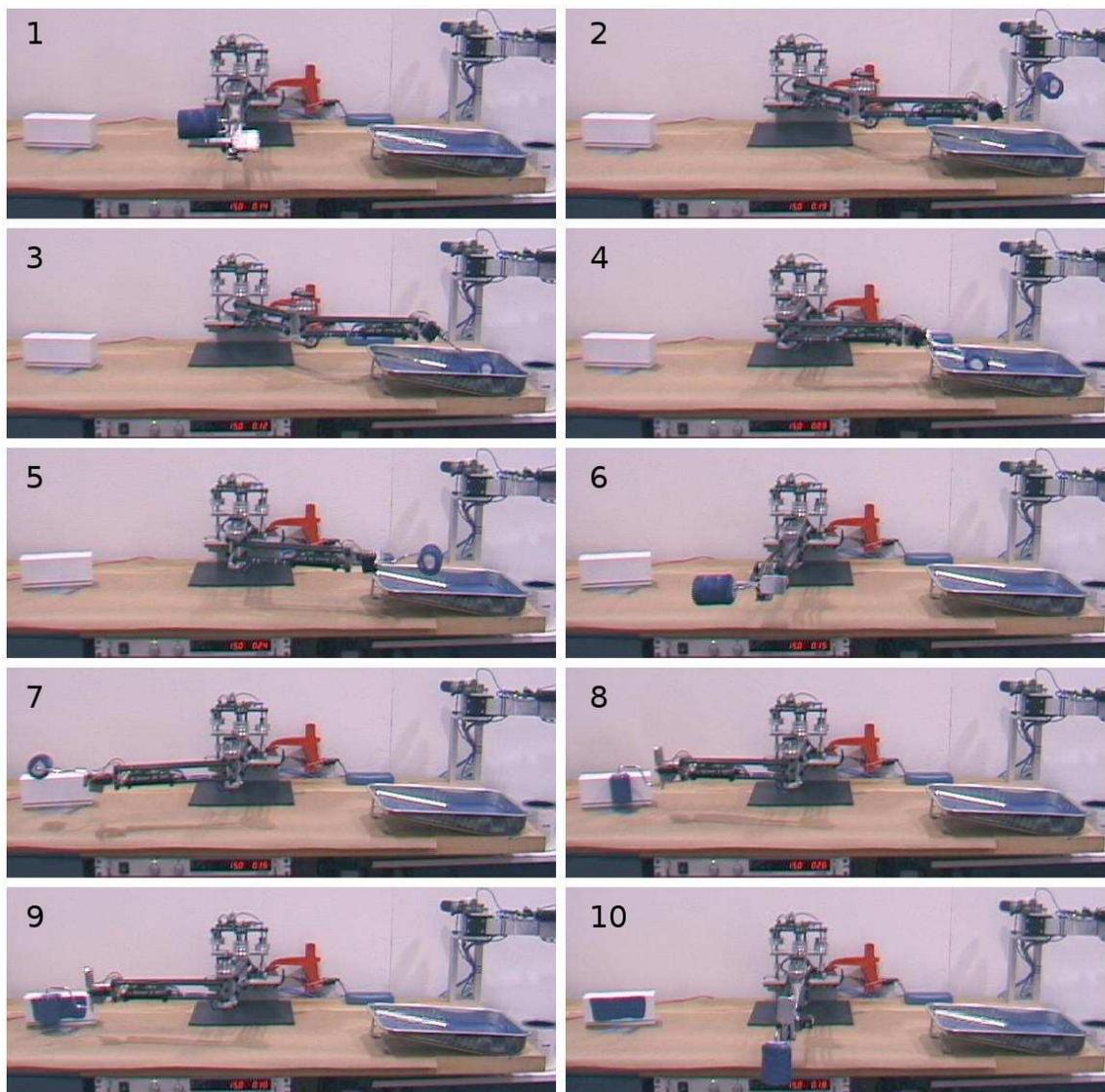


Figure 5-6: Painting task montage. **1:** Go to home position and calibrate. **2:** Move to paint tray. **3:** Dip roller in paint. **4:** Roll back to wipe excess paint off of the roller. **5:** Lift roller from tray. **6-7:** Move roller to other side of workspace. **8:** Compliantly detect object. **9:** Paint object in torque control mode. **10:** Return home.

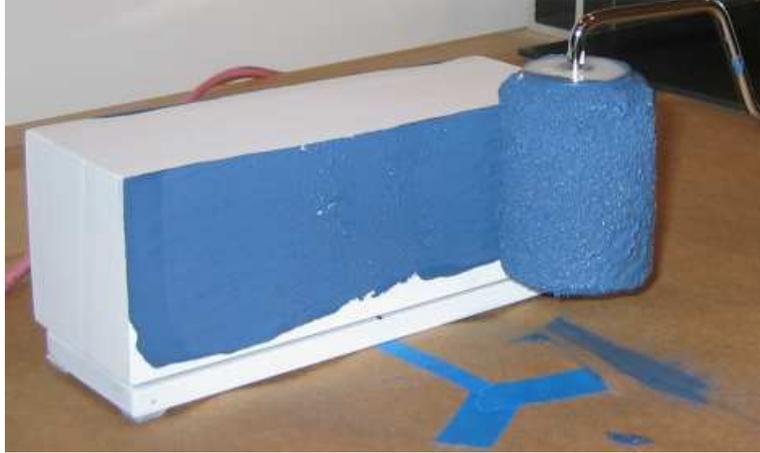


Figure 5-7: Closeup of the painted object near the end of the painting task. The blue tape was used as an indicator of the allowable rotation and position of the object's center.

### 5.3.3 Implications

While the task of moving a small roller across a known object is somewhat of a toy example, it does still illustrate that a low-cost compliant robotic system compensating for inexpensive mechanical parts with computation can indeed work. Simple hysteresis, backlash, and signal compression models were able to turn an inherently unstable and imprecise system into something substantially more controllable, safe, and adaptable.

This experiment not only highlighted the features that worked well for the painting task, but also illuminated a few aspects of the robot that were lacking in some form or another. While the use of a backlash model did increase the stability of the arm and allow for a reduction in steady-state error, the controller was still unable to handle a significantly large integral term and thus some steady-state error still existed. This error reduced the accuracy of the position controller and allowed offsets from joint torque measurements to affect the actuator's position. It was also difficult to cause the arm to appropriately respond to significant changes in the environment (e.g., forced movement of the arm, moving the paint tray, using a much smaller object) since the robot did not have a high-level model of its workspace. Certain events could be observed and altered depending on the outcome, but others lacked sufficient

information about the environment and therefore could not be changed accordingly. Visually servoing the arm for certain tasks and building a map of the physical layout of the workspace could greatly improve both the performance and the effectiveness of the robot when faced with uncertainty.

Building upon the conceptual knowledge gained from each of the experiments within this chapter, it is very likely that more complex tasks could be performed by sewing together the various modeling, construction, control, and actuation techniques presented with little or no additional cost. While only the easy task of painting the side of a rectangular block was achieved, it is certainly nothing less than a lower bound on the capabilities that can arise by exploring the trade-off between mechanical precision and computation. New mechanical configurations, more optimized system-level architectures, and the falling price of computation will all help to increase the capabilities of future inexpensive compliant robotic systems and allow for a more diverse set of compensation methods and tasks to be performed.

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

The research presented in this thesis explored new kinds of compliant robotic elements that have the potential to trade-off precision mechanical parts for the increased computational power necessary to perform accurate modeling and compensation. The price of precision mechanical parts was shown to have remained relatively constant over the last 13 years, while the price of computation has decreased exponentially over that same time. This observation led to a focus on creating low-cost compliant robotic parts that could take advantage of the ever-decreasing price of processing power.

Three realms of compliance were explored; rotational compliance, structural compliance, and drive-system compliance. The Torsionally Compliant Elastomer Joint (TCEJ) served as the centerpiece for most of the research since it offered inexpensive series compliance at the cost of much needed hysteresis and compression modeling. A hysteresis model was then developed and shown to compensate for the large offsets observed from the TCEJ's integrated sensing and problems associated with oversensitivity were reduced by the aid of a compression model. The low-cost window lift motors used to actuate the robot's joints contained significant spacing between its gears and therefore motivated the development of backlash modeling to aid in controllability as well.

To demonstrate and test the new compliant elements and their associated models, three proof-of-concept systems were built. The first two systems were designed to test the TCEJ and contrast the capabilities of two drastically different control architectures. The hierarchical architecture allowed for increased scalability and robustness, but did so at the cost of increased complexity, latency, and cost. Both TCEJ robotic arms allowed for smooth torque control and safe human interaction throughout their workspace. The third proof-of-concept system explored a different compliant design that replaced traditional steel cables with a springy nylon line and took advantage of carbon fiber where possible to greatly reduce the weight of the arm. All three systems were used for experimentation to better understand the new compliant elements, their associated models, and the overall capabilities of each system.

A handful of new ideas have been presented in this work, but there are still many important questions left to be answered. The next section provides a list of future research areas that I believe are likely to help advance the field of low-cost compliant robotics so that it may have the ability to aid our everyday lives as we transition from mainframe style robotics to the PC.

## **6.2 Future Work**

### **6.2.1 Variable Stiffness Compliance**

One important set of low-cost compliant prototypes that was not able to be researched in this exploration is that of variable stiffness compliance. The advantages that accompany the ability to dynamically change the mechanical impedance of an actuator are well documented[8] and allow for control over a very wide range of environmental admittances. Variable stiffness actuators are most commonly achieved by non-linear antagonistic drive mechanisms (e.g., muscles applying a torque on a joint in opposite directions). The non-linearity of the actuator allows for variable stiffness dependent upon the amount of pre-loading in the drive-system. The complexity of control and sensing of these actuators currently limits their feasibility and would certainly benefit

from additional exploration to reduce their cost and implementation difficulties.

### **6.2.2 High-Level Modeling of Non-Ideal Components**

While the architecture decisions made throughout this exploration were focused on using computation to aid the imprecision and non-idealities of low-cost mechanical components, substantial research still needs to be done to fully develop many of the computational models. In particular, the high-level models that either self-evolve (learning algorithms) or deal with slow, time-varying effects (like part wear and fatigue over time) have not been attempted in this initial stage of research surrounding low-cost compliant robotics. A truly robust solution will take into account all perceived and expected changes within a part to allow for reliability and consistency over the entire span of the robot's lifetime.

### **6.2.3 Environmental Perception**

Ultimately, a robot can only react in a consistently intelligent manner if it is able to accurately perceive its surrounding environment (via combined sensing of sight, sound, touch, and perhaps someday, even taste and smell) and act accordingly. For example, if a person enters the workspace and keeps the robot from performing an action it is quite different from an object being placed in the workspace that limits the robot's ability to move freely. A person preventing the robot from moving may indicate that the current action should not be performed, or perhaps that the person is choosing to guide the action. Whereas an object placed in the workspace might signal a new task to be performed or an additional sub-goal with a set of trajectories around which to navigate to avoid collision. This ability to not only detect the environment, but instead to perceive objects, people, situations, etc. is very important and still the focus of significant research within the robotics community.[7]

## 6.2.4 Engineering for Production

The work presented in this thesis is intended to be exploratory in the sense that it focuses on new ways to achieve low-cost compliant robotic systems by trading off mechanical precision for computational power. Techniques, tools, and materials used for rapid prototyping often do not directly scale to production and thus require a substantial amount of additional engineering. Production tools, materials, and methods are likely to both limit certain elements originally used in prototypes and also enable a new set of potential solutions (e.g., large scale casting, specialized tooling, complex surfacing techniques, expensive joining technologies, etc.) that may otherwise seem impractical for small batch sizes.

## 6.2.5 Data and Command Granularity Optimization

With significant latency occurring between the highest and lowest processing and control units in a hierarchical control architecture, it is very important to send commands and data at a well chosen granularity. If an entire task is given to a low-level controller, processing power may be quickly exhausted and a substantial amount of communication between low-level controllers may be necessary to verify task completion. If the command granularity is too small (e.g., sending thousands of successive position coordinates to approximate a sine wave instead of sending a single sine wave command), the latency and bursty nature of best-effort networks will likely cause jumpiness at the output since there is no guarantee that packets will arrive with the same inter-packet time intervals at which they were sent. It is for these reasons that special care should be taken to optimize communications by choosing an appropriate level of granularity for all data and commands.

## 6.2.6 The Need for a Soft Exterior

In addition to the compliance embedded within each degree of freedom of a low-cost compliant robotic system, a soft exterior coating is also very important. Even if the robot is well controlled and smoothly moves between locations, objects (including

people) in the environment may move in unexpected ways and occasionally collide with the arm. A well designed compliant robotic system will reduce the severity of these unexpected collisions by including a soft, fleshy covering over the robot wherever possible. This compliant exterior skin also provides for more environmental sensing than torque or force sensing measured at the joint alone. The robots designed and tested in this exploration did not have a soft exterior since they were used mainly as proof-of-concept systems for low-cost compliant mechanisms, but a truly compliant robotic system should exploit soft interactions at every possible level.



# Appendix A

## Price Data of Mechanical Parts and Computation

	1993	1999	2000	2001	2003	2005	2006
Linear Bearing	\$9.69	\$10.02	\$10.42	\$10.71	\$10.71	\$11.60	\$12.00
Ball Screw / inch	\$2.12	\$2.26	\$2.26	\$2.34	\$2.39	\$2.75	\$2.75
Ball Nut	\$124.09	\$131.84	\$131.84	\$136.72	\$139.46	\$153.35	\$155.65
Flexible Coupling	\$17.58	\$21.75	\$22.30	\$22.85	\$22.85	\$22.85	\$22.85
Miter Gear	\$11.57	\$15.06	\$13.02	\$13.02	\$13.40	\$15.05	\$15.76
100 MIPS[4] <sup>1</sup>	\$47.62	\$2.78	\$1.72	\$1.06	\$0.41	\$0.16	\$0.10
Relative C.P.I. <sup>2</sup>	1.000	1.055	1.138	1.222	1.219	1.305	1.476

Table A.1: Price trend data of precise mechanical parts and computation over time.[2]

Name	Description	McMaster-Carr
Linear Bearing	1/4" x 1/2" x 3/4" Frelon Lined Linear Bearing	#5986K2
Ball Screw / inch	1/2" Ball Screw (0.5" lead, 4150 Steel)	#5966K25
Ball Nut	1/2" Ball Nut (Mates with #5966K25)	#5966K15
Flexible Coupling	1/4" x 1/4" (7 degree max) precision coupling	#6208K22
Miter Gear	Steel 20 degree Miter Gear (12px15x1/2" bore)	#6529K15

Table A.2: Description of precise mechanical components used in Table A.1 and their associated McMaster-Carr part number used to obtain price data.

<sup>1</sup>MIPS pricing data was produced in 2002 and thus prices for 2003-2006 are forecast from that time. It is difficult to reproduce the exact formulation used by the report, but present day pricing seems to uphold the predicted data reasonably well.

<sup>2</sup>The Relative Consumer Price Index (C.P.I.) was calculated by taking equal contributions from 4 consumer products (Electricity, Gasoline, Bread, and Chicken) and normalizing them to 1.000 for the year 1993. Consumer pricing data was obtained from the U.S. Bureau of Labor Statistics.



# Appendix B

## Hysteresis Modeling Code

### B.1 MATLAB: *hysteresis\_sim.m*

```
function [time,ThetaM,ThetaP,ThetaE,Torque] = hysteresis_sim(time, halldiff, realTorque)
% simulation of hysteresis model.
% --> [time,ThetaM,ThetaP,ThetaE,Torque] = hysteresis_sim(time, halldiff, realTorque)
%
%
%
%          ---      ThetaE      -----
% halldiff---->(+_)----->[__fTorque()__]-----> Torque
%          (+)  ^                               |
%          |      ThetaP      -----          |
%          (-)'-----[_fPlasticPWL()_]<----'
```

```
%
%
ThetaP=0;
ThetaM=0;
ThetaE=0;
Torque=0;

pos = 0;
neg = 0;

HallDiffToDegrees=0.05;

lastThetaP=0;
PlasticDeformationData.posThetaP = 0; % start with no pre-loading
PlasticDeformationData.negThetaP = 0; % of the material.
```

```

lasttime = 0;

samples = size(time);
for iter=1:samples

    ThetaM(iter)=HallDiffToDegrees*halldiff(iter);
    ThetaE(iter)=ThetaM(iter) - lastThetaP;

    % Model Elastic Deformation ---> Torque
    Torque(iter)=fTorque(ThetaE(iter),lastThetaP);

    timestep = time(iter)-lasttime;

    % Model Torque ---> Plastic Deformation
    [ThetaP(iter),newPD]=fPlasticPWL(timestep,Torque(iter), ...
                                    PlasticDeformationData, lastThetaP);

    PlasticDeformationData = newPD;
    lastThetaP = ThetaP(iter);
    lasttime = time(iter);

    pos(iter) = PlasticDeformationData.posThetaP;
    neg(iter) = PlasticDeformationData.negThetaP;

    if(mod(iter,1000)==0)
        iter
    end
end
end

```

## B.2 MATLAB: *fTorque.m*

```
function [ Torque ] = fTorque( ThetaE, ThetaP )
% elastic deformation to torque model

a0 = 575;

% a bit more accurate for the Torsionally Compliant Elastomer Joint
% but really not much better than linear for tests.
Torque = a0*tan(2*ThetaE/(360)*2*pi);

if(Torque > 1000)
    'Torque seems excessively large!'
end
```

## B.3 MATLAB: *fPlasticPWL.m*

```
function [ newThetaP, newPD ] = fPlasticPWL( timestep, Torque, PD, lastThetaP )
% Piecewise linear plastic deformation model

% timestep is included for general plastic deformation case
% with time dependencies (material relaxation time)
% -- not used in this simple piecewise linear model

% members of PlasticData {PD} Structure...
%   created   : 'true' if it's been created (not used here)
%   posThetaP : {last ThetaP with Positive Torque}
%   negThetaP : {last ThetaP with Negative Torque}
%
posEndT = 100;           % end points that define the plastic deformation curve
posEndThetaP = 1.5;
negEndT = -posEndT;
negEndThetaP = -posEndThetaP;

% default if Torque is 0.0
newThetaP = lastThetaP;

% piecewise linear deformation model
if(Torque > 0)
    posLineSlope = (posEndThetaP-PD.negThetaP)/posEndT;
    newThetaP = max(Torque*posLineSlope+PD.negThetaP,lastThetaP);
    PD.posThetaP = newThetaP;
end
if(Torque < 0)
    negLineSlope = (negEndThetaP-PD.posThetaP)/negEndT;
    newThetaP = min(Torque*negLineSlope+PD.posThetaP,lastThetaP);
    PD.negThetaP = newThetaP;
end

newPD = PD;
```

# Bibliography

- [1] United Kingdom Patent #GB928659: Improvements in or relating to Rubber or other Elastomeric Substance Suspension means for Vehicles, June 1963.
- [2] McMaster-Carr Supply Company Catalog – New Brunswick, NJ, USA, 1993,1999,2000,2001,2003,2005,2006. Used to obtain historic pricing data of mechanical parts.
- [3] James S. Albus, Ronald Luminia, John Faila, and Albert Wavering. The NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM). *Technical Report Tech Note #1235, NASA SS-GFSC-0027, National Bureau of Standards*, 1986.
- [4] John Bourgoïn. The coming reality for soc designers. 2002. – Prices forecast by CEO of MIPS Tech. Inc.
- [5] Aaron M. Dollar and Robert D. Howe. Towards grasping in unstructured environments: Grasper compliance and configuration optimization. *Advanced Robotics*, 19(5):523–543, 2005.
- [6] Aaron M. Dollar and Robert D. Howe. A robust compliant grasper via shape deposition manufacturing. *Mechatronics, IEEE/ASME Transactions on*, 11(2):154–161, April 2006.
- [7] Aaron Edsinger and Charles C. Kemp. Manipulation in human environments. In *Proceedings of the IEEE/RSJ International Conference on Humanoid Robotics*, 2006.

- [8] Neville Hogan. Impedance control: An approach to manipulation: Part1 - Theory. *Journal of Dynamic Systems, Measurement, and Control*, pages 1–7, March 1985.
- [9] Neville Hogan. Impedance control: An approach to manipulation: Part2 - Implementation. *Journal of Dynamic Systems, Measurement, and Control*, pages 8–16, March 1985.
- [10] Neville Hogan. Impedance control: An approach to manipulation: Part3 - Applications. *Journal of Dynamic Systems, Measurement, and Control*, pages 17–24, March 1985.
- [11] John M. Hollerbach, Ian W. Hunter, and John Ballantyne. A comparative analysis of actuator technologies for robotics. *Robotics Review*, 2:299–342, 1992.
- [12] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–727, 1999.
- [13] Dale A Lawrence. Actuator limitations on achievable manipulator impedance. *IEEE*, pages 560–565, 1989.
- [14] E. K. Miller. The computer revolution. *Potentials, IEEE*, 8:27–31, May 1989.
- [15] Gill A. Pratt and Matthew M. Williamson. Series elastic actuators. *IEEE*, pages 399–406, 1995.
- [16] Gill A. Pratt, Pace Willisson, Clive Bolton, and Andreas Hofman. Late motor processing in low-impedance robots: Impedance control of series elastic actuators. In *Proceeding of the 2004 American Control Conference*, pages 3245–3251, 2004.
- [17] Stephan Salenius, Karin Portin, Matti Kajola, Riitta Salmelin, and Riitta Hari. Cortical control of human motoneuron firing during isometric contraction. *The American Physiological Society*, pages 3401–3405, 1997.

- [18] Eduardo Torres-Jara. Obrero: A platform for sensitive manipulation. *Proceedings, IEEE-RAS International Conference on Humanoid Robots*, pages 327–332, 2005.
- [19] K.T. Woo, Li-Xin Wang, F.L. Lewis, and Z.X. Li. A fuzzy system compensator for backlash. *IEEE Proceedings*, pages 181–186, May 1998.
- [20] You Wu, Kiyoshi Fujikawa, and Hirokazu. A control method of speed control drive system with backlash. *IEEE*, pages 631–636, September 1996.
- [21] Ikuo Yamano. Five-fingered robot hand using ultrasonic motors and elastic elements. In *Proceeding of the 2005 IEEE*, pages 2684–2689, 2005.