The Thesis Committee for Andrew J. Zelenak
Certifies that this is the approved version of the following thesis:

A Compliant Control Law for Industrial, Dual-Arm Manipulators

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: _____
Sheldon Landsberger

Co-Supervisor: _____
Mitch Pryor

A Compliant Control Law for Industrial, Dual-Arm Manipulators

by

Andrew J. Zelenak, B.S.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May, 2013

Acknowledgements

I would like to thank the following people. Without them, this thesis would not have been possible.

- My family, including Steve and Amy Zelenak, Georgie Zelenak, and Catherine Knight, for their love.
- Dr. Mitch Pryor, for providing a tremendous amount of good advice and freedom to experiment.
- Dr. Sheldon Landsberger, for his good advice and constant support.
- My fellow students at The University of Texas at Austin.
- Many good teachers at the Colorado School of Mines, The University of Texas at Austin, and Poudre High School. Learning is easier when you have a good instructor.
- Many friends in Colorado and Texas who made it fun along the way.

Abstract

A Compliant Control Law for Industrial, Dual-Arm Manipulators

Andrew J. Zelenak, M.S.E.

The University of Texas at Austin, 2013

Supervisor: Sheldon Landsberger

Co-supervisor: Mitch Pryor

Many of the first robots ever built, decades even before the first industrial robots, were humanoids. It seems like researchers have always sought to imitate the human form with their robots, and with good reason. Humans are incredibly flexible; they can perform a huge variety of tasks, from locomotion over rough terrain, to delicate assembly, to heavy lifting. A human's second arm allows him to lift twice as much weight. His workspace is approximately doubled, and he can perform a broader variety of tasks as items are passed back and forth between hands. We sought to impart some of that same functionality to a strong, rigid, dual-arm robot. Specifically, we developed a control law that allows two robot arms to lift and manipulate an object in cooperation.

As opposed to the prior art, our control law is tailored for industrial robots. These robots do not usually allow torque control and their control frequency is generally 60 Hz. Through the use of fuzzy logic, the control law is quite robust at 60 Hz control rates. Its simple structure reduces the computational cost of the algorithm by approximately 75% over Jacobian-based methods. Stability is proven and the controller parameters can be

adjusted to handle perturbances of arbitrary magnitude. Since the robots behave as an admittance, torque control is not required. Several experiments were conducted to benchmark and validate the performance of this control law. The controller is able to maintain a clamp force within ± 4N despite a wide variation in trajectory and control frequency. This fine level of force control makes the controller suitable for delicate tasks.

The conclusion suggests several extensions that would make this control law more useful. For example, adaptive control would improve the performance. A position feedback controller should be cascaded so that the robot arms' tracking accuracy is improved. Many tasks (such as co-robotics) require external compliance, and we show how external compliance could easily be incorporated.

Table of Contents

List of Tables

List of Figures

CHAPTER 1 : INTRODUCTION

From the beginning, the paragon of robotics research has been the human being. da Vinci was sketching mechanical humanoids in the 15th century. The Westinghouse Electric Corporation was exhibiting humanoids such as Elektro at several World's Fairs in the 1920-30's [Reis, 2013], around the same time that Isaac Asimov was imagining a future full of humanoid robots. All of these developments, it is worth noting, came prior to the first industrial robot.

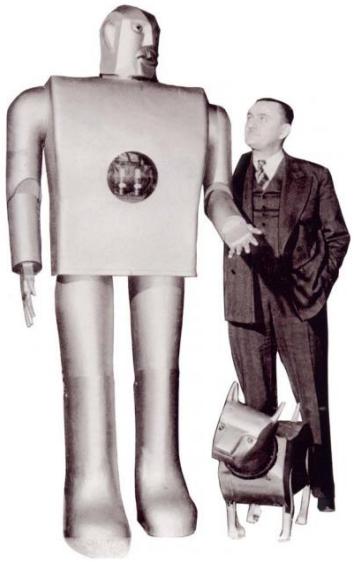

Figure 1-1. Humanoid Robot Elektro at the World's Fair

Until recently, researchers could only envy the ease with which humans recognize objects, interact with their environment, and ambulate across rough terrain. There was a very clear divide between robotic performance and human performance. In the early 21st century, however, developments in the fields of software, sensing, hardware, and control theory bring robots closer to human performance than ever before.

There are many examples of their advancing capabilities. Boston Dynamics has demonstrated a biped (PETMAN) that navigates rough terrain with surprising agility. Vision sensors and algorithms allow objects to be recognized with 90% accuracy [O'Neil, 2013], and they are grasped with 80-100% success rates [Zelenak, 2013]. The Defense Advanced Research Projects Agency (DARPA) seeks to develop adaptable rescue robots that can use human tools and drive human vehicles ["Tactical Technology Office," 2013]. Six of the seven most well-funded competitors in a new DARPA competition are using dual-arm, anthropomorphic robots ["DARPA Robotics Challenge Track A", 2013]. Dual arm robots will become more common. This thesis seeks a control law that makes them more practical and more useful across a broad range of tasks.

Figure 1-2. DARPA Challenge Anthropomorph from Virginia Tech ("DARPA Robotics Challenge Track A", 2013]

1.1 INDUSTRIAL SIGNIFICANCE OF DUAL-ARM MANIPULATION

The benefits of a dual-arm manipulator are significant. For example, they double the payload and extend the work envelope. Stable manipulation of bulky objects and two-hand assembly tasks become possible. There may be some cost savings if two smaller, inexpensive arms can be used in place of one larger manipulator.

Given these advantages, why is cooperative manipulation by strong industrial robots such a rarity? Dual-arm manipulators can crush or drop objects due to small positional inaccuracies that arise from sensor error, calibration error, or mechanical tolerances. In an unstructured environment, these errors are unavoidable [Braun, 2004]. This thesis presents a control law which allows cooperative industrial dual-arm manipulators to be employed in unstructured environments with greater confidence.

1.2 THE CO-ROBOTICS CHALLENGE

There is a trend towards more human-robot interaction. The trend spans consumer robotics, manufacturing, and space exploration and is known as co-robotics. *Popular Mechanics* selected co-robotics as one of its top ten emerging tech trends for 2013:

> *"Old-school industrial robots work best alone—try to help an assembly-line welding bot and you'll probably get welded. But the next generation of robots will work closely with humans, augmenting our capabilities and compensating for our weaknesses. That's why the National Robotics Initiative is pouring up to $50 million a year into co-robotics. The initiative is backed by agencies ranging from NASA (robots to help astronauts and to explore terrain where humans can't go) and the National Institutes of Health (robot surgery for everyone and home care for the elderly) to the Department of Agriculture (robots that can deworm animals and sense fruit ripeness). A key first step to robot—human interaction: full-size*

humanoids such as UPenn and Virginia Tech's SAFFiR (above), which will help fight fires." ["The 10 Tech Terms to Know," 2013]

There is an ongoing push to develop International Organization for Standardization (ISO) requirements for human-robot interaction [Harper, 2010]. The standard writers note that this new generation of co-robots will need a new level of flexibility and safety. "Flexibility" is an ambiguous word and at least two meanings apply in this case. First, the new co-robots must be adaptable to the world around them, since they will be operating in unstructured environments. Secondly, the robots must be physically flexible as opposed to rigid.

In robotics parlance, physical flexibility is known as "compliance." Compliance is important because it can protect humans in case of unexpected contact. Compliance also alleviates the large force errors that arise from small positional inaccuracies in stiff systems. Imagine a car crashing into a brick wall. If the car's frame were completely rigid, a tremendous amount of force would be transmitted to the driver. However, if the frame is designed to crumple, it absorbs some of the impact energy and the driver may remain unharmed. The same concept applies to compliant robots.

1.3 COMPLIANCE, ADMITTANCE, AND IMPEDANCE

Hogan, [1984] pioneered the study of compliant robotics. Prior to Hogan's seminal paper, a robot was generally controlled in either the force domain or the position domain. Hogan emphasized that *both* force and position must be controlled, in the general case, when a robot interacts with the environment. He tied force and position together by implementing a virtual spring law. Strictly speaking, the interaction law does

not have to be a spring law; damping terms and other terms can also be included. In this thesis, we use fuzzy logic.

There are two ways to approach the spring law. If the robot gets pushed off its trajectory, then it could be programmed to push back with a certain force. This is properly termed "Impedance Control" and is the much more common method. Alternatively, the robot could be programmed to sense applied forces. When a force is detected, it will deviate from its path. This is properly termed "Admittance Control".

To be clear, admittance and impedance are subsets of compliance as illustrated in Figure 1-3. Compliance can be implemented in hardware or software. This thesis takes the software approach. It is more flexible and it is applicable to off-the-shelf manipulators.

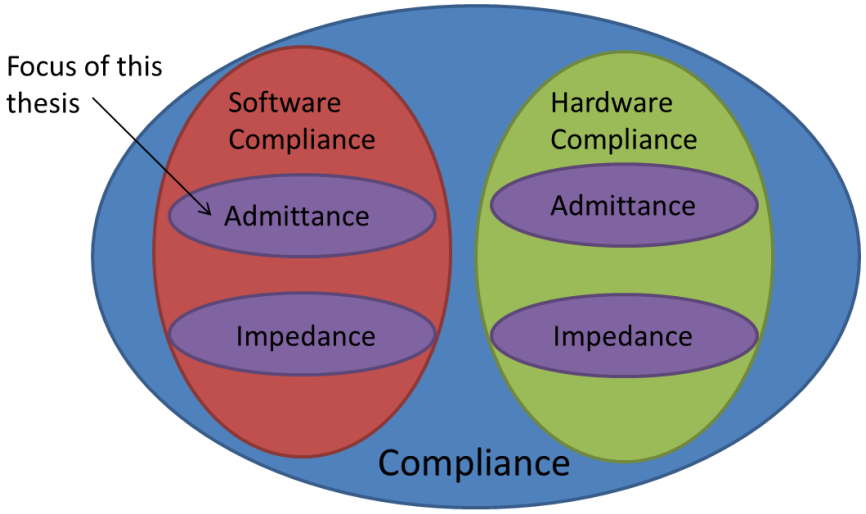

Figure 1-3. Compliance Venn Diagram

Compliant control creates the potential for safe operation and a more intuitive system reaction to changes in the environment, but tuning the controller – especially for a broad range of possible interactions – remains and open issue.

1.4 STATE OF THE ART COMPLIANT ROBOTS

This section takes a look at three broad categories of compliant robot. There are compliant actuators, software-based compliance, and soft-body compliance.

- *Actuator Compliance*

One group of compliant robot uses specialized actuators to modulate its interaction forces with the environment. Essentially, the actuators are comprised of a spring in series with a motor. This type of arrangement is called a "series elastic actuator." The spring reduces the rigidity of the arm; it also absorbs some impact force if the robot has an unexpected collision. But, a series elastic actuator can reduce the degree of accuracy required by some tasks. For example, if a rigid robot needs to press a button, it must have a very accurate world model and very accurate position control. A series-elastic-driven robot can achieve success with much less accuracy. Hence, the designer can use cheaper, less accurate components, from gearboxes to vision systems.

An example of a series-elastic-driven robot is Baxter from Rethink Robotics, as shown in Figure 1-4 ["How Rethink Robotics Built Its New Baxter Robot Worker," 2012]. This is a startup company that aims to replace humans on assembly lines. Its designers leverage the low-cost advantage of series elastic actuators to sell the robot for just $22,000. Of course, there are disadvantages to the series elastic actuator. The compliance will reduce its positional accuracy. This type of robot also tends to have a lower payload than a comparable rigid robot. Finally, it would be difficult to modify Baxter's hardware if the work environment changes. A press-fit task might require more stiffness than a pick-and-place task, and Baxter cannot perform the press-fit.

Figure 1-4. Baxter, the low-cost manufacturing robot

- *Software compliance*

ABB, one of the world's largest robot manufacturers, has recognized the value of compliance. ABB is a huge Swiss conglomerate that specializes in industrial automation. Currently, it is the #150th most valuable company in the world ["Global 2000 Leading Companies," 2012]. In 2006, ABB introduced its SoftMove software ["Robot Compliance in One Cartesian Direction," 2013]. This program imparts a "virtual compliance" to rigid industrial robots in one direction only. The spring rate and damping can be adjusted and there is an option for gravity compensation. ABB cites injection molding as one application where one-dimensional compliance is useful. In this case, the robot must have high radial stiffness and accuracy as it removes a part from the machine, but free-floating motion in the axial direction is important.

The ability to dynamically adjust compliance parameters on the fly is the primary advantage of software compliance. A drawback to the technique is its dependence on computational power. To get reliable performance from the robot, its software must

operate in real time. For delicate tasks, a high control frequency is important. Thus, software compliance will likely become more common as computer speed continues to increase in coming years.

- *Soft-body Robots*

Researchers have been using highly customized, flexible, compliant robotic bodies to achieve specialized tasks. An example is the "squid robot" from Harvard University ["Squid Robot," 2011]. It is actuated with pneumatics, and its soft body allows it to move in a wavelike motion. A nice feature of this robot is its ability to squeeze through tight spaces, as shown in Figure 1-5.

Figure 1-5. Squid Robot

1.5 IDEAL DUAL-ARM MANIPULATOR

This section presents twelve characteristics of an ideal dual-arm industrial controller, followed by an explanation for each:

- *Since the robot's interactions with the environment are nonlinear, the controller should be nonlinear.*
- *There should be some method of compliance, such as admittance control, for regulating dynamic interactions with the environment.*
- *There should be a method for limiting internal forces during cooperative manipulation.*
- *The controller should be robust.*
- *To be compatible with industrial robots, the control law should be written as an admittance rather than an impedance.*
- *The controller should be model-free.*
- *There should be a method of distributing the forces between arms.*
- *It is desirable (and mandatory in our case) to formulate the control law in velocity space.*
- *The arms should operate indistinguishably, i.e. both arms should follow the same control law.*
- *Computational cost of the algorithm should be minimized.*
- *Global asymptotic stability of the control law should be proven.*
- *The control law should generalize to redundant manipulators.*

Since the robot's interactions with the environment are nonlinear, the controller should be nonlinear.

The design and analysis of a control system is easier if it is linear. Then, tools such as root-locus plots and pole placement can be used in the design of the controller. Unfortunately, this is not the case here. A linear system f(x) must satisfy two conditions:

$$f(x+y) = f(x) + f(y) \qquad \text{(Principle of Additivity)}$$

$$f(\alpha x) = \alpha f(x) \qquad \text{(Principle of Homogeneity)}$$

In tightly-controlled circumstances it is possible to come up with an accurate compliance model of the environment. But this model will almost certainly be nonlinear. See, for example, (Diolaiti, 2005]. Diolaiti explains why linear models and the coefficient of restitution approach that are commonly used to model impact force are not very accurate. Then he develops a nonlinear model that correctly predicts contact force versus penetration depth for a silicon gel and for a stiff plastic. This process required a great deal of experimental data collection and detailed knowledge of the material properties. For example, the simplifications that apply to a viscous material are not applicable for a stiff material. It seems unlikely that an operator would care to spend so much time developing a contact model for each material that a robot might encounter.

While we may not have a good model of the manipulator and/or the environment, we know enough to conclude that the manipulation problem is nonlinear. There are numerous uncertainties that may contribute, e.g. variable control frequency, variable inertia and stiffness of the object, sensor uncertainty, and thermal expansion, to name a few. Clearly some of these variables will have some nonlinear effects such as hysteresis. Therefore we will need a nonlinear controller to tackle the problem. The controller should perform well across a wide range of materials of uncertain properties.

There should be some method of compliance, such as admittance control, for regulating dynamic interactions with the environment.

Hogan, [1984] recognized the nonlinearity of manipulation, and he explains the need to control both position and force. By doing so, the large force errors that arise from small positional accuracies can be controlled. The robots are then able to operate in an unstructured environment.

There should be a method for limiting internal forces during cooperative manipulation.

If two arms are rigidly manipulating a quasi-rigid object, there is the potential to develop large internal forces. The forces arise from small position inaccuracies [Braun, 2004]. The positional inaccuracies might be caused by controller delays, poor calibration, sensor error, mechanical tolerances, etc. Bonitz, [1996], Caccavale, [2008] and Schneider, [1992] have developed controllers that control the internal force.

The controller should be robust.

A "robust" controller is able to function properly despite uncertainty. This is extremely important because there will always be some uncertainty in the real world. A few of the many factors that may contribute to uncertainty are:

- Misalignment of robot mounting positions
- Computational delays
- Finite sensor resolution
- Finite sensor sampling period
- Thermal expansion
- Drivetrain backlash

To be compatible with industrial robots, the control law should be written as an admittance rather than an impedance.

The majority of robots do not allow torque control of the revolute joints [Ferretti, 2000], [Wimbock, 2012]. Thus, it is difficult to specify the end effector force. For this reason, admittance control is preferred if the robot does not allow torque control.

The controller should be model-free.

When we say "model-free", we mean that the controller should not rely on a model of the robots' dynamics or a model of the object's dynamics. Often a robot manufacturer does not know or make the inertial data for its robots available publicly. Even if that data is available, it will not be accurate if the robot is modified (e.g. with cable routing or end effector modifications). Dynamic end-effectors such as grippers may change position as the robot operates, thus modifying the robot's inertial properties. Finally, machining tolerances, variable states of lubrication, power fluctuations, etc., will introduce even more uncertainty. For these reasons and more, the control law should not require a dynamic model of the manipulator.

Likewise, an object's inertial properties are rarely known beforehand, so it is ideal if the control law can accurately track a trajectory without such knowledge [Braun, 2004]. Bonitz, [1996] presents a control law that does not require knowledge of dynamic properties, but it does not allow dynamic interactions with the environment.

There should be a method of distributing the forces between arms.

Bonitz, [1996] and Wimbock, [2012] point out that load sharing is a primary reason to use a dual-arm robot. In order to minimize the size of the manipulators, there should be a method to evenly split the load or, at least, cap the fraction that will be carried by one arm. Kosuge, [1997] presents an excellent, easy-to-implement method for load sharing, but it is predicated on knowing the object's properties beforehand.

It is desirable (and mandatory in our case) to formulate the control law in velocity space.

Most compliant controllers to date have been implemented in the position domain. See e.g. [Caccavale, 2008; Hogan, 1984]. However, as Duchaine, [2007] found, there are reasons to program a robot in the velocity domain. Consider, for example, that encoders have finite resolution. For high-frequency control of a robot, the motion over each time step may be so small that it does not even register on the encoders.

Duchaine did an experiment that found, surprisingly, that position-based control felt "greatly overdamped" to humans, whereas velocity-based control of a robot felt more similar to its programmed virtual mass. Test subjects were able to guide a robot that was programmed in velocity space through a maze both more quickly and more accurately than a position-based robot with the same virtual mass and damping parameters. Additionally, Cusumano, [2012] found that humans use speed control rather than position control when they walk on a treadmill. The motion control software at our lab only allows high-frequency velocity control, so position control is not an option.

Neither Duchaine nor Cusumano tried to explain *why* velocity control was the preferred method, but we can make a few educated guesses. These explanations are best made graphically. Consider Figure 1-6. This digital controller is trying to track the function $y=x^2$. Clearly velocity control provides a much smoother and more accurate output than position control.

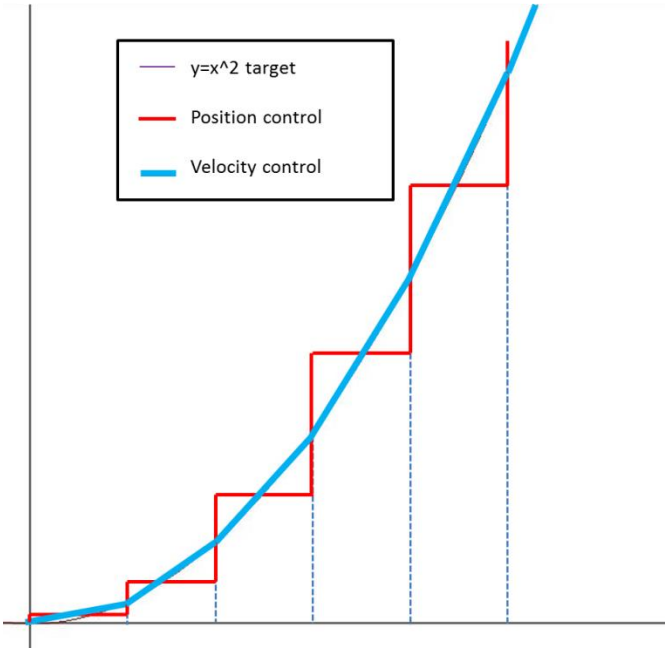

Figure 1-6. Position vs Velocity Control: Smoothness

Now consider Figure 1-7. This image shows the effect of a delayed control signal. The position control holds its position throughout the delay, so it accumulates a large amount of error. The velocity control accumulates much less error because it approximates the function more closely.

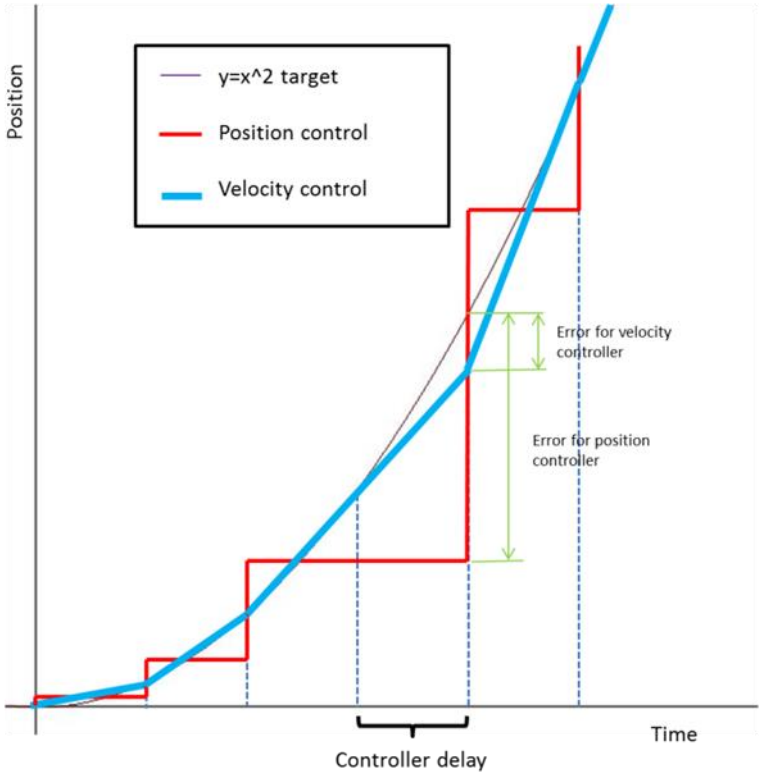

Figure 1-7. Position vs Velocity Control: Effect of Control Signal Delay

One could argue that acceleration control would be even smoother and more robust to control signal delays.

The arms should operate indistinguishably, i.e. both arms should follow the same control law.

It makes intuitive sense that a symmetric, dual-arm system should use the same control law for both arms. This simplicity will save tuning and analysis time.

Computational cost of the algorithm should be minimized.

Chapter Three covers how computational delays can exacerbate the overshoot of a manipulator. In order to minimize this problem, the code for the controller should be streamlined, and the number of operations should be minimized. Chapter Three also discusses the strategies that we use to minimize the computational delays of the algorithm.

Global asymptotic stability of the control law should be proven.

Operators should not worry about unusual operating conditions that cause rapid, increasing robot velocity. If the controller can be proven "globally asymptotically stable", then unbounded behavior is not a concern.

The control law should generalize to redundant manipulators.

Many control laws in the literature, e.g. [Bonitz, 1996; Caccavale, 2008], depend on inversion of the Jacobian to calculate a target force or position. Inversion of the Jacobian is computationally expensive and does not apply to redundant manipulators, since there are infinite solutions to the inverse kinematics problem. Thus we seek a controller that generalizes to redundant manipulators and does not rely on inversion of the Jacobian.

1.6 SPECIFIC REQUIREMENTS OF THE NUCLEAR INDUSTRY

The Nuclear Robotics Group at U.T. Austin is focused on the nuclear industry. In this industry, safety and reliability are the foremost concerns. Thus, robots are unlikely to operate at high speed. This relaxes one requirement on the controller; it is acceptable to treat the situation as quasi-static and ignore inertial forces from the object.

Also, at this point in time, whatever environment the robots operate in is likely to be highly structured. This relaxes the compliance requirement. Compliance is still needed to limit the internal stresses of the object, but this thesis does not include a method of detecting collisions and controlling external interactions. However, the development of collision-detection algorithms is an active area of study in the NRG [Schroeder, 2011].

1.7 SUMMARY OF OBJECTIVES

This thesis aims to develop a control law for a rigid, industrial, dual-arm manipulator. We seek to meet as many of the "twelve characteristics of an ideal dual-arm manipulator" as possible.

1.8 ORGANIZATION OF THE REPORT

The remainder of the report is organized as follows:

Chapter Two is a literature review. It covers previous research in the field of cooperative dual-arm control, and looks at nonlinear control techniques that may be relevant to the issue.

Chapter Three covers some of the barriers to practical implementation. Digital sampling can wreak havoc on a control law if it is not implemented correctly, and the effects of variable control frequency and network delays are discussed.

Chapter Four explains the structure of our control law in a step-by-step fashion, starting with our coordinate system convention and an introduction to fuzzy logic. It concludes with a complete description of the spatial control law.

Chapter Five analyzes the stability of the controller and its robustness.

Chapter Six covers the experimental results that validate this thesis.

Chapter Seven summarizes the contributions of this research and discusses future work.

CHAPTER 2 : LITERATURE REVIEW

A literature review is approached from two different angles. First, we look at previous cooperative dual-arm control research. How do these various control schemes stack up against the twelve characteristics of an ideal dual-arm controller that was presented in Chapter 1? Where is room for improvement, and what have been the main challenges, historically? Finally, we compare and contrast the nonlinear control strategies that may be relevant to this control problem.

2.1 PREVIOUS DUAL-ARM CONTROL LAWS

How have dual-arm robots been controlled in the past? Position control may be the most simple, intuitive type of control, but it is not suitable for cooperative manipulators. While commanding a motion from point A to point B works for unconstrained tasks such as welding, it is not sufficient when environmental interactions are involved [Hogan, 1984]. As we noted in Chapter 1, small positional inaccuracies between the arms can cause large internal forces in the objects. Force control, by itself, is also inadequate. In general, the manipulators must have some knowledge of location. Here we examine the other control methods that have been applied.

Master/slave manipulators came first because of their simplistic setup. In this configuration, one arm is position-controlled and the other arm follows in a compliant manner. As Arimoto, [1987] points out, this method does not rely on a-priori object information. Some downsides of this method are a lack of load sharing and a lack of compliance with respect to the environment.

The first attempts to reconcile force/position control were hybrid control schemes. These were popular in the late 1980's and early 1990's. Hayati, [1986] proposes a hybrid

scheme where a position control loop and a force control loop operate simultaneously. The method is predicated on precise inertia data about the robots and the object. An operator must input matrices to describe constraints on the system. Stability is proven. A nice advantage to his approach is that the control gains need no tuning; they fall out of the dynamic equations for the system.

Yoshikawa, [1993] uses a very similar architecture but he adds another feature: the ability to specify constraint forces. The output of Yoshikawa's control law is joint torques. Unfortunately, many industrial robots do not allow torque control. The most serious drawback for these hybrid schemes, in general, is their inability to handle environmental uncertainty. They depend on the operator to select the system constraints beforehand.

Bonitz, [1996][1] makes use of Hogan's impedance concept. He recognized that the manipulator forces could be divided into two categories: the motion-inducing net force and the internal force. An impedance controller is implemented to control the internal force. Bonitz treats each manipulator itself as an impedance. The upside of this approach is that a-priori knowledge of manipulator and object inertia is not required. One downside is that object weight and other dynamic forces will cause a slight variation from the target trajectory. In addition, the control law requires inversion of the Jacobian to calculate inverse kinematics. The inversion is computationally expensive and limits the method to 6-DOF or lower manipulators. Bonitz proves stability but not robustness.

This internal force-based impedance controller meets six of my twelve "characteristics of an ideal controller." Its drawbacks are the impedance formulation

[1] Bonitz is now a Principal Investigator at the Jet Propulsion Laboratory, where he has programmed the arms on the Mars rovers, among other projects [https://www-robotics.jpl.nasa.gov/people/Robert_Bonitz/].

(which is not suitable for industrial manipulators), the lack of force distribution between arms, and the controller output is not a velocity.

Caccavale, [2008] is a more recent example of a dual-arm controller that is based on Hogan's impedance concept. Caccavale uses three different feedbacks to drive the robot motion: position feedback, internal force feedback, and external force feedback. A linear Proportional-Integral-Derivative (PID) controller is used for path following on the position feedback. Impedance is implemented on the internal force feedback; it ensures that internal stresses on the object are controlled. Similarly, impedance is implemented on the external force feedback, so the manipulator can handle environmental contact. There are some drawbacks to Caccavale's controller: It requires *a-priori* inertia data for the robot and it is not suitable for industrial robots since it is an impedance. Like Bonitz, it requires inversion of the Jacobian, which is computationally expensive and limits the algorithm to 6-DOF or fewer manipulators. Finally, the linear PID control of position is not ideal and there is no method for sharing the load between arms.

An entirely different approach is the use of artificial potential fields (aka "virtual force fields") to guide the robots. With this technique, the operator sets up a potential field with an attractive maximum at the target location. Obstacles will be represented as minima. The robot will "climb the gradient" until it reaches the target location while the gradient will naturally push the robot away from obstacles. The artificial potential field method has not, to the authors' knowledge, been used on cooperative manipulators. It has, however, been used to plan manipulation by a swarm of mobile robots [Song, 2002]. Song achieves this cooperative manipulation through the superposition of many different potential fields. He sets up a visco-elastic artificial potential field that surrounds each robot; this is like a buffer zone (Figure 2-1). The decentralized controllers transition

through three modes. First, they will *approach* the object; this is achieved by setting up an attractive potential at the object location.

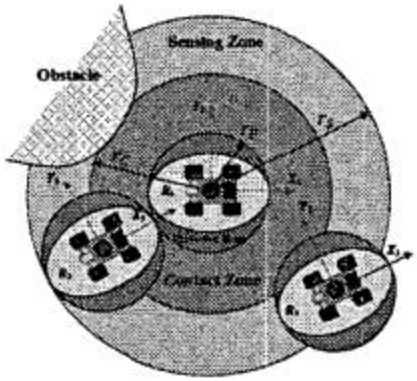

Figure 2-1. Various artificial potential fields for cooperative mobile manipulation

Next, each robot will individually go into an *organize* mode, where it seeks to distance itself from its neighbors. This process ensures that the object is trapped. When all robots have reached the *organization* mode, the controllers transition to *transportation* mode. In this stage, a transportation potential is set up at the goal, which attracts all of the robots. Figure 2-2 conveys this process.

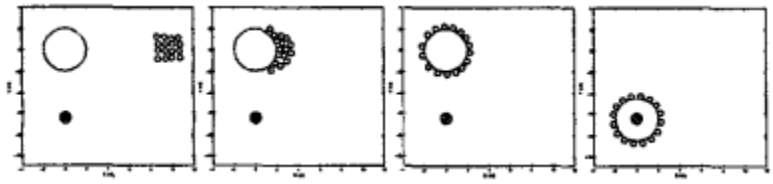

Figure 2-2. A swarm of robots surrounds and transports the ball

Some downsides to the method of artificial potential fields are described in Koren, [1991]. The four drawbacks he describes are:

- Becoming trapped at local minima.
- No passage between closely spaced obstacles.

- Oscillations in the presence of obstacles.
- Oscillations in narrow passages.

If artificial potential fields were applied to the present problem of dual-arm manipulation, it is unclear if force distribution and the limitation of internal forces could be achieved. Certainly the low computational cost, robustness, and model-free nature of the method are attractive.

Table 2-1 shows a summary of the six dual-arm control laws that are discussed here. Many of the control laws achieve half of the twelve ideal characteristics. The best performer is also the most unique, this being the method of artificial potential fields. It achieves seven of twelve. However, it has plenty of its own drawbacks as discussed on the previous page. Our new control law, as presented in the following chapters, achieves eleven of the twelve. The only characteristic that it misses is "*There should be a method of distributing the forces between arms.*"

Table 2-1. Comparison of Controllers. ✓ indicates the criterion is met.

	Arimoto, 1987- Master/slave	Hayati, 1986- Hybrid control	Yoshikawa, 1993- Hybrid control	Song, 2002- Artificial Potential Fields	Bonitz, 1996- Impedance	Caccavale, 2008-Impedance
Nonlinear	✓	✓	✓	✓	✓	✓
Compliant				✓		✓
Internal forces are limited		✓	✓	?	✓	✓
Controller is robust	✓	?	?	✓	?	?
Formulated as an admittance						
Knowledge of object's inertia is unnecessary				✓	✓	
Force is distributed between arms		✓				
Formulated in velocity space	✓					
Indistinguishable operation		✓	✓	✓	✓	✓
Low computational cost	✓			✓		
Proven global stability	✓	✓	✓	✓	✓	✓
Generalizes to redundant manipulators	✓	✓	✓			

2.2 NONLINEAR CONTROL STRATEGIES

Nonlinear control schemes can be separated into two flavors: there are methods that adapt linear techniques and there are Lyapunov based methods. Before choosing our nonlinear controller, we looked at one Lyapunov method and four linear adaptations.

Sliding Mode

The idea behind sliding mode control is to pick an equation in the state space that naturally moves the system towards equilibrium. Once the system reaches the sliding surface (a.k.a. "sliding mode"), it will naturally decay towards equilibrium (just like a marble rolling along a crack). A discontinuous, bang-bang controller is used to keep the system on the sliding mode, despite disturbances. For example, in Figure 2-3, the "surface" $\dot{x}_1 = -x_1$ is chosen. The system ends up at a stable equilibrium point at (0,0). In general, there is no guarantee that a suitable surface can be found. If a suitable surface does exist, each state and its derivative must have an opposite sign. For the example of Figure 2-3, notice that $\dot{x}_1$ is positive while x_1 is negative. The opposite sign ensures that the system will slide towards equilibrium at the origin.

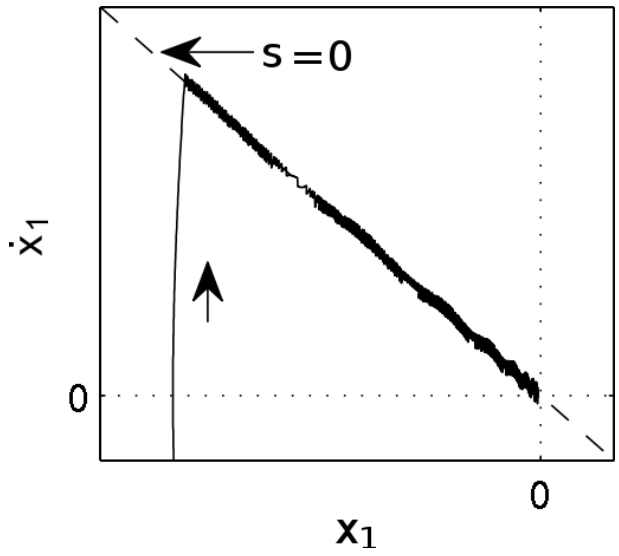

Figure 2-3. Example Sliding Mode Controller

Mathematically, consider a system with x as the state vector and candidate sliding surface σ(x)=0. If $\sigma^T\dot{\sigma} < 0$ in the neighborhood of the surface, then the signs will be opposite and the system must slide toward the origin.

Robustness is the main advantage of sliding mode control. The bang-bang controller that holds the system on the surface is not precise, and therefore it is not very sensitive to disturbances. On the other hand, the discontinuous bang-bang controller causes chatter and it requires infinite gain. Both of these attributes are undesirable on real hardware systems. Finally, a model of the system dynamics is needed if sliding mode control will be used.

Feedback Linearization

Feedback linearization is an interesting technique where transformations are applied to the input, converting the system to a linear set of equations. From there, standard linear methods like root-locus plots and pole placement can be used for design

of the controller. Some limitations on this transform are that it be invertible (so the true input value can be recalculated) and that it be smooth (so that differentiability in the original and the new coordinate system are preserved).

The transformation is achieved by successively differentiating the system equation until the input variable appears. These derivatives are used as a new coordinate system; they are linear with respect to the input variable, and arbitrary pole placement can be used for design of the controller. Unfortunately, for our dual-arm control problem, feedback linearization is not an option since we don't have a good dynamic model of the system.

Gain Scheduling

If an undergraduate student, fresh from linear controls class, were asked to design a nonlinear controller, he would probably formulate something similar to gain scheduling. Gain scheduling assumes that the control surface varies slowly. At any point, a first order Taylor Series Expansion gives a good, local estimate of the gradient of the nonlinear problem. So, the control surface is divided into a grid and approximated by linear functions at each mesh point. Standard linear controllers can be used to create a nice controller for each region. Leith, [2000] describes it as a "decomposition of nonlinear design into linear sub-problems." The trick lies in picking the regions. Often, the equilibrium points are picked as mesh points and the control problem is linearized around each equilibrium point. If the system resides in-between mesh points, the output is interpolated. Note this approach requires an equation that describes the system dynamics.

Fuzzy Logic

Alternatively, operator experience can be applied to create a family of linear controllers. When operator experience is utilized, this technique is sometimes known as "fuzzy logic." Wang, [1993] points out five advantages of fuzzy logic control:

- It does not require a dynamic model of the system.
- It is often described in terms of human logic and if-then rules. This makes it intuitive.
- The "software cost" is low because of the intuitive nature and simple implementation.
- Fuzzy controllers are nonlinear "universal approximators," i.e. a fuzzy controller is capable of replicating any function. More complicated function approximations require more if-then rules.

Fuzzy control has a few drawbacks. Since it is not based on system dynamics, it is not optimal. Also, there is no general proof of fuzzy control stability [Wang, 1993].

Despite these shortcomings, fuzzy logic is often used in industrial applications, such as mobile robots. See e.g. [Ishikawa, 1991; Juang, 2009]. It has also found industrial application in subways [Yasunobu, 1985], trains [Oshima, 1988], and chemical processes [Horiuchi, 1999]. The "model-free" nature of fuzzy logic may be its most appealing characteristic.

2.3 LITERATURE REVIEW SUMMARY

This chapter started with an examination of the cooperative dual-arm control literature, from the first master-slave systems up to compliant robots with three feedback loops. A table is used to compare the pros and cons of each controller. Clearly, there has

been some progress over three decades of research, but none of these controllers is ideal for industrial cooperative manipulation. The best controller [Bonitz, 1996] is lacking because it does not distribute force between the arms, and it cannot be applied to the majority of industrial robots (because it requires torque control).

Secondly, several nonlinear control techniques are covered. There were three methods that adapt linear techniques to solve nonlinear problems: gain scheduling, feedback linearization, and fuzzy logic. We also investigated sliding mode control, which is based on Lyapunov methods. All of these techniques require a mathematical model of the system except for fuzzy logic, which relies on operator experience.

CHAPTER 3 : BARRIERS TO PRACTICAL IMPLEMENTATION

Digital signals are essential to the field of robotics. Much of the sensing and control research that occurs today is dependent on fast digital computations. However, it is important for engineers and scientists to realize that a digital signal does not represent the world with 100% accuracy. Digital sampling of a physical quantity, such as force or position, leads to discontinuities. This chapter explains a filtering technique that can smooth out the discontinuities.

Although processors are getting faster all the time and computers have more processing cores, computational delays can still affect a control system. It is not unusual for our control frequency at the Nuclear Robotics Group to vary by a few Hertz for the exact same piece of code! That is why it is also important for roboticists to understand how network and processing delays can affect their system.

3.1 DIGITAL FILTERS AND DIGITAL DERIVATIVES

When an analog waveform is represented digitally, some error is inevitably introduced due to quantization and a finite sampling time (Figure 3-1). If the waveform represents position, then the digital signal indicates instantaneous changes in position. The process of differentiation exacerbates the digital errors. For example, the velocity derivative of a digital position signal is much noisier than the original signal (Figure 3-2).

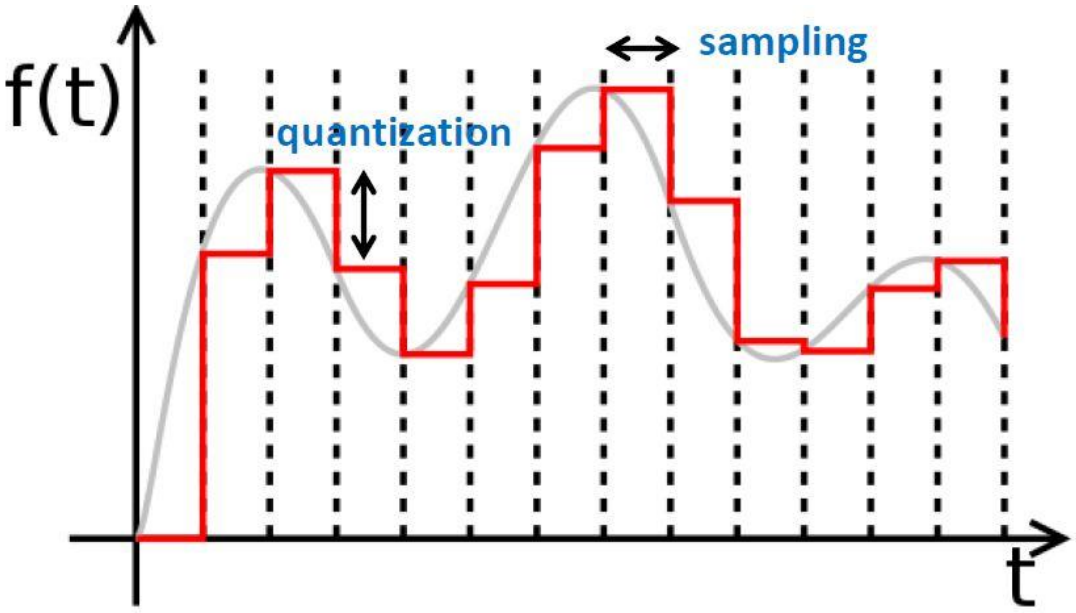

Figure 3-1. Factors that cause errors in digital signals [Paine, 2012]

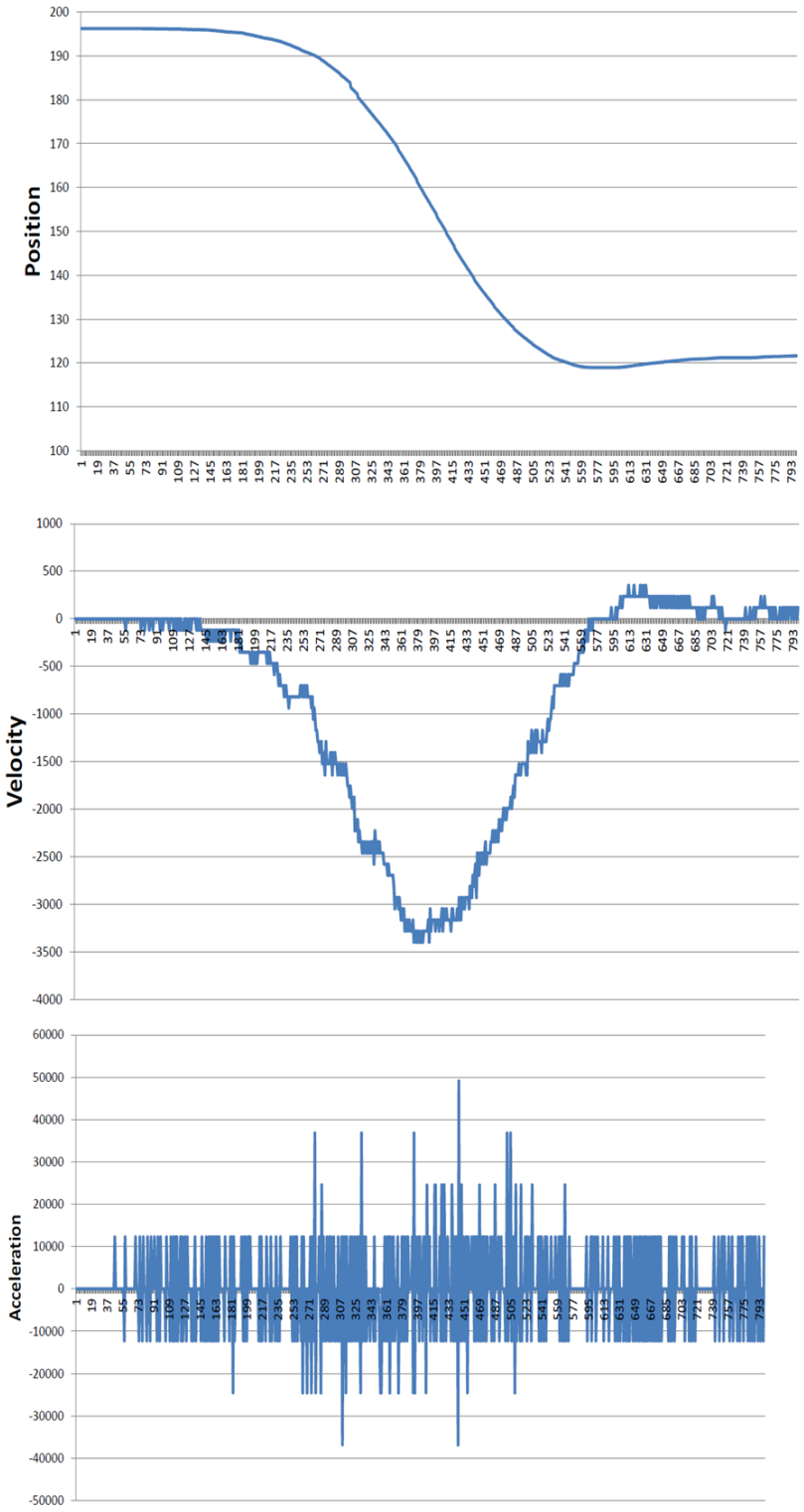

Figure 3-2. Noise is introduced with successive digital differentiations [Paine, 2012]

A common method of differentiating a signal is to transform it into the Laplace domain, a.k.a. the s-domain. The Laplace domain has the handy property that differentiation is accomplished simply by multiplying by s. The transfer functions for low-pass filters in the s-domain are well-known [Paarmann, 2001]. Once the differentiation and the filtering have been applied in the s-domain, the Tustin transform is used to get back into the digital domain, a.k.a. the discrete time domain.

Here is an example of the process as velocity is calculated from position:

1. First transform the two waveforms to the Laplace domain

$$x(t) \overset{L}{\leftrightarrow} x(s)$$
$$v(t) \overset{L}{\leftrightarrow} v(s)$$

2. Multiply by s to take the derivative of position

$$v(s) = s * x(s)$$

3. Multiply by the transfer function of a second-order low-pass filter in the s-domain

$$v(s) = \frac{1}{\frac{s}{\omega_c}^2 + \sqrt{2} * \left(\frac{s}{\omega_c}\right) + 1} * s * x(s)$$

ω_c is the desired cutoff frequency of the low pass filter (LPF), in rad/s. The resulting transfer function is:

$$H(s) = \frac{v(s)}{x(s)} = \frac{s}{\frac{s}{\omega_c}^2 + \sqrt{2} * \left(\frac{s}{\omega_c}\right) + 1}$$

4. Transform from the *s*-domain into the discrete domain with the Tustin transform by substituting for s.

$$s \leftrightarrow \frac{1 - z^{-1}}{T_s} * \frac{z}{1 + z^{-1}}$$

This results in an equation of the form:

$$H(z) = \frac{v(z)}{x(z)} = \frac{az^2 + bz + c}{z^2 + dz + e}$$

5. Cross-multiply

$$v(z) * (z^2 + dz + e) = x(z) * (az^2 + bz + c)$$

6. Divide by z^2 and rearrange

$$v(z) = x(z) * (a + bz^{-1} + cz^{-2}) - v(z) * (dz^{-1} + ez^{-2})$$

Remember that, in the discrete time notation, x(z) is the current sample of the position signal. x(z)* z^{-1} is the previous sample and x(z)* z^{-2} is the two-previous sample. At last the equation is in a form where we can get some intuitive meaning. The current velocity is a function of the current position, previous positions, and previous velocities. It is similar to a weighted average, where the weights are determined by the low-pass filter.

In practice, MATLAB software provides methods to calculate the coefficients *a* through *d*. The Appendix contains the annotated MATLAB code; it is only eight lines. The naive alternative to this Laplace-Tustin approach is a finite difference approximation, e.g.

$$v[n] = \frac{x[n] - x[n-1]}{T_s}$$

Figure 3-3 shows that the Low Pass Filter-Laplace-Tustin method produces a much smoother signal than the first order finite difference approximation. Another advantage is the designer can specify a cutoff frequency.

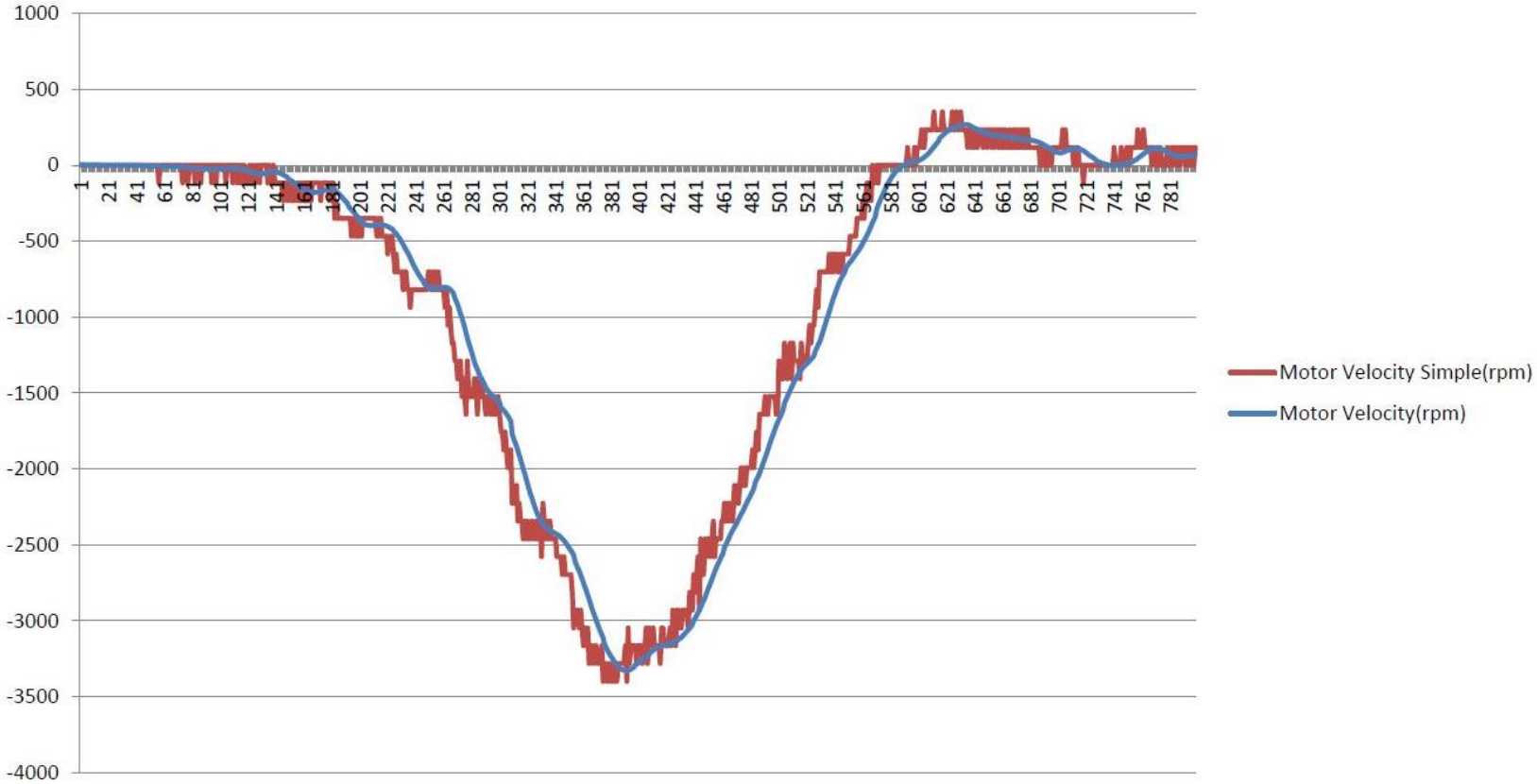

Figure 3-3. Discrete LPF is smoother than first order finite difference approximation [Paine, 2012]

The dual-arm controller that we present in this paper uses digital sampling of force data from an ATI force/torque sensor. Without some filtering, there would be a lot of chatter, so we apply this LPF-Laplace-Tustin technique to the force data. The cutoff frequency is set at 326 Hz. That may sound like an arbitrary number, but it is one of the built-in options for the force/torque sensor, and it approximately matches the control rate of our system.

3.2 NETWORK AND PROCESSING DELAYS

Recently, processing delays and network delays have been studied for their effects on robotic systems. A system which is perfectly tuned under ideal conditions will overshoot significantly if its control signal is delayed, as seen in Figure 3-4.

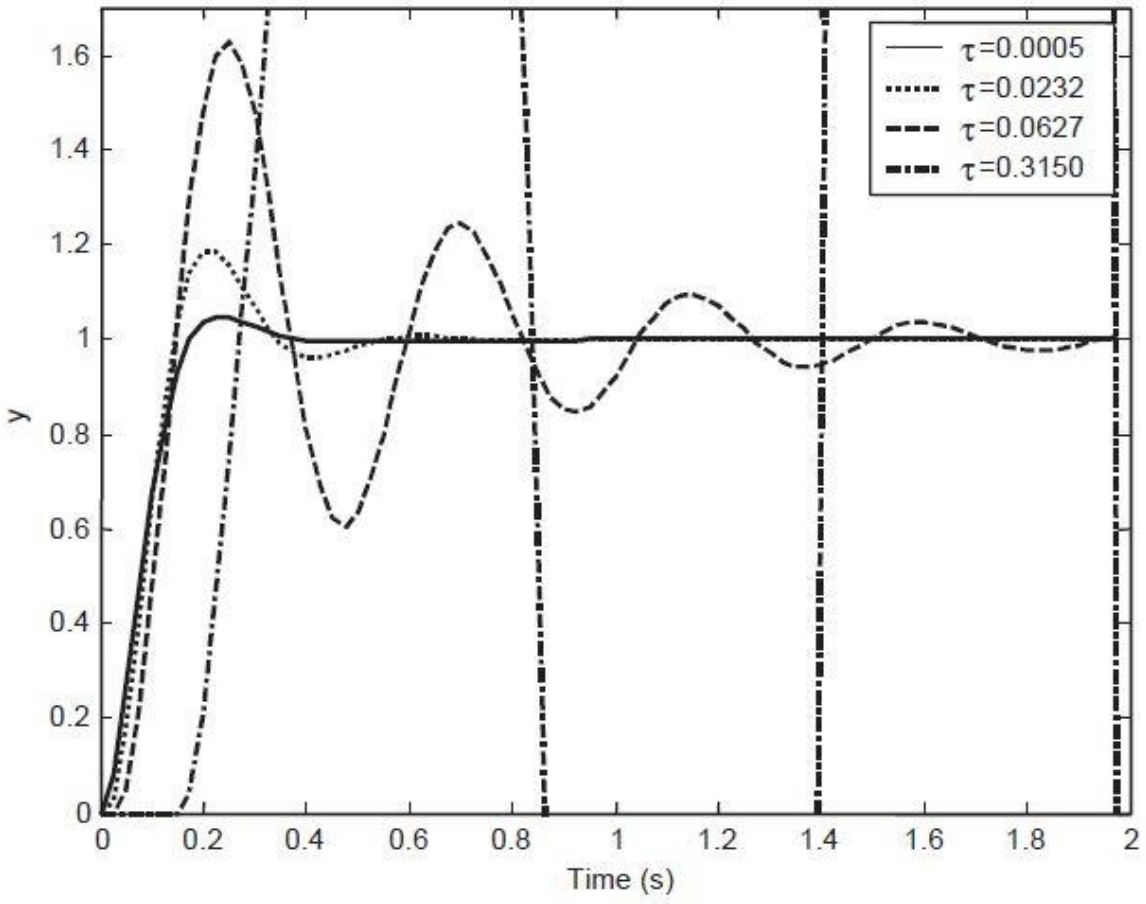

Figure 3-4. System dynamics degradation due to increasing network delay [Tipsuwan, 2003]

Tipsuwan, [2003] presents a great summary of methods that have been used to deal with control signal delay. These methods run the gamut from Perturbation Theory to Optimal Stochastic Control. The selected method should be simple to implement and should "piggyback" on current control systems, if it is to gain widespread use.

Tipsuwan mentions two techniques that could be readily applied to industry-standard PID controllers. The first is known as Fuzzy Logic Modulation. The PID gains are multiplied by a coefficient, β. β can take on two different values between zero and

one; for larger errors, β is larger. This adjustment reduces rise time yet maintains stability as errors decrease and the system approaches its target. As can be seen in Figure 3-5, the technique is quite effective [Almutairi, 2001].

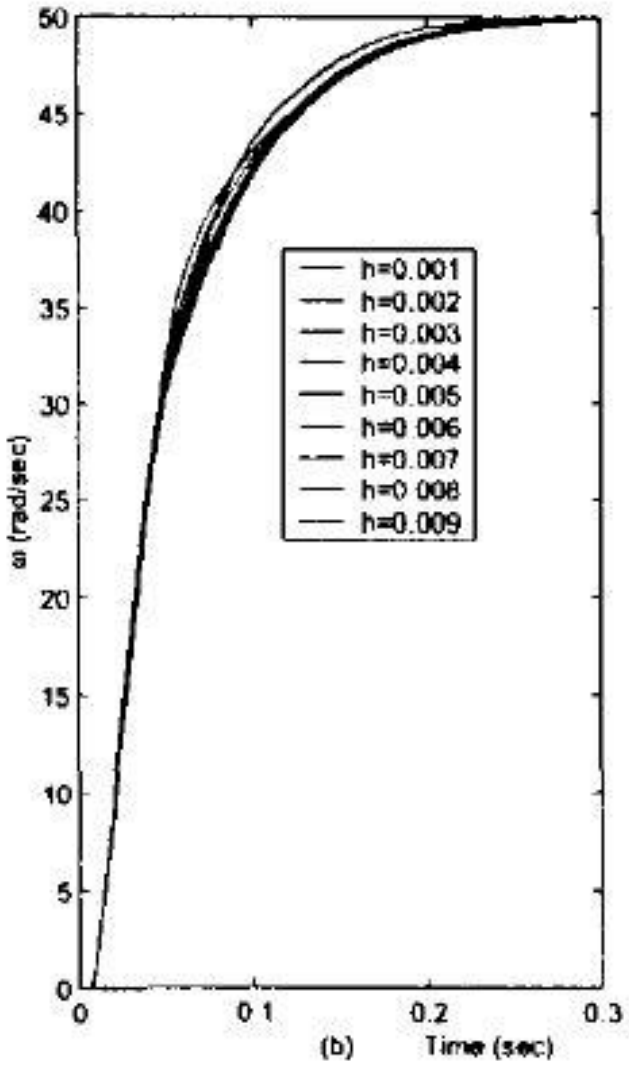

Figure 3-5. Fuzzy logic modulation creates a stable system despite varying network delays [Almutairi, 2001]

A disadvantage of Fuzzy Logic Modulation is the need to compute or tune many values: α1, β1, α2, and β2. Computing suitable β's, in particular, requires a heavy amount

of testing and optimization based on cost functions. Such extensive tuning is a barrier to the widespread use of Fuzzy Logic Modulation. Another barrier is that the delay must be known.

The other method for adjusting gains on-the-fly is known as End-User Control Adaptation. Essentially, a large databse of optimal gains is computed, beforehand, for a large variety of delays. As the system operates, the gains are chosen from this databse to match the operating conditions. This technique also performs quite well, yet heavy testing is required beforehand and it also depends on knowledge of the control signal delay. The cost surface representing the database of optimal gains can be seen in Figure 3-6.

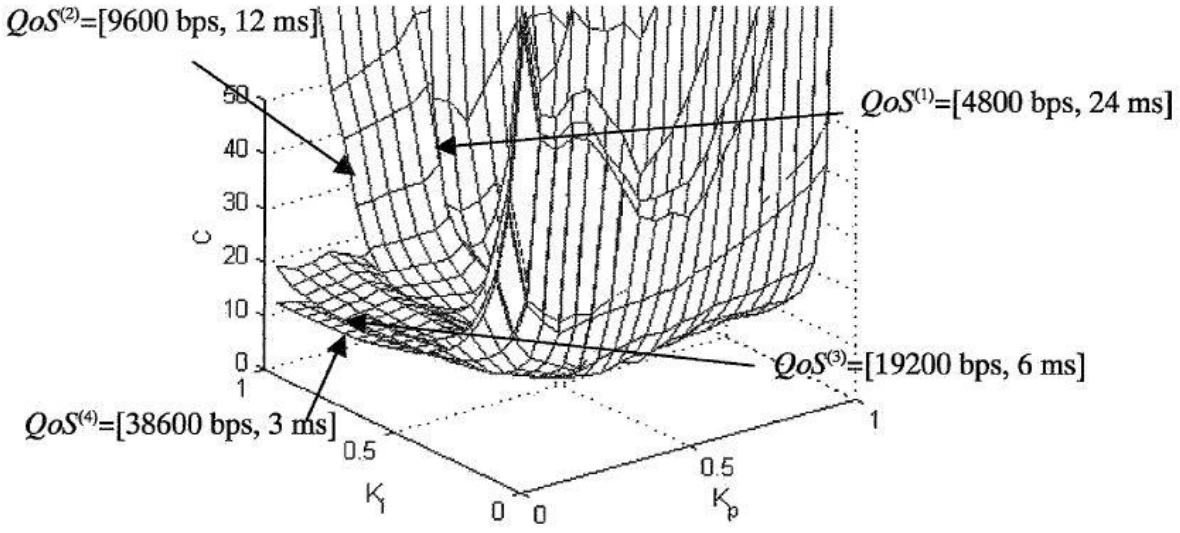

Figure 3-6. Precomputed cost surface of control gains given network parameters [Tipsuwan, 2003]

3.3 OUR STRATEGY TO REDUCE PROCESSING DELAYS

We adapted an active rather than a reactive approach to handle processing delays. In other words, rather than reacting to the effects of delays, we sought to minimize delays in the first place. Specifically, we aimed for lean, minimal code. Function arguments were passed by reference rather than by value to minimize the execution time. We eliminate some unnecessary code by using one of the built-in capabilities of the force-torque sensors; their electronics are capable of performing coordinate transformations on the force measurements. Finally, we set the thread priority[2] of the controller to a very high value so it assumes priority over other applications.

Our choice of a fuzzy logic controller also reduces the processing delay of the code relative to other control laws. We will see in Chapter Four how the fuzzy logic controller works. Suffice it to say, for now, that the calculation of a target velocity in each dimension requires 50 floating point operations. We calculate for four dimensions since two of the moments are related to equal-and-opposite translations. Adding in a few other, necessary additions brings the total computational cost of the fuzzy control law to 244 floating point operations. This metric of number of floating point operations is often used when comparing computer algorithms, since the actual runtime will vary from one computer to another.

Now we will compare the 244-operation cost of our fuzzy logic to the control law of Bonitz, [1996] for a 6-DOF robot:

Equation 3-1. Bonitz's Control Law

$$\tau_i = D_i\{J_i^{-1}(M_i^{-1}[M_i\ddot{x}_{id} + B_i\delta\dot{x}_i + K_i\delta x_i - \delta\tilde{f}_{Ii}] - \dot{J}_i\dot{q}_i)\} + E_i + J_i^T\tilde{f}_i$$

[2] A "thread" is a program that can execute independently. Computers can share processor resources to run multiple threads simultaneously.

The matrices in Bonitz's law are [6x6] and there are two [6x6] inversions. A naïve cost estimate of matrix multiplication assumes that the operation cost is $O(n^3)$. The cost for matrix inversion is also $O(n^3)$. This is the number of operations it would take to perform the multiplication or Gaussian elimination by hand. So, each [6x6] inversion or multiplication costs ~6^3=216 operations. A [6x6]x[6x1] multiplication will take 66 operations; we assume the cost of a [6x6] transpose is equal to the number of elements (36); the cost of [6x1]+[6x1] is 6. Working through Bonitz's control law from inside to outside, summing up the computational costs, there are 1032 operations. This is over four times as many operations as the fuzzy logic controller. Note, the two matrix inversions are the largest contributor, by far, to the high computation cost of Bonitz's controller, accounting for about 41% of its computational cost.

There are more efficient ways to multiply or invert matrices than $O(n^3)$. The current record is $O(n^{2.3727})$ [Williams, 2011]. However, these super-efficient methods are numerically unstable and difficult to program. If Williams' simplifications were applied, they would bring the computational cost of Bonitz's control law down to 740 operations— which is closer but still over three times as many as the 244-operation cost of the fuzzy logic controller.

The net effect of these considerations is a control rate of 250 Hz. The maximum control rate that we could hope to achieve with our Agile Planet controllers is 1 kHz. There is room for further improvement, but the typical industrial controller operates at 60 Hz. So we achieve a control frequency that is roughly four times better than a typical industrial robot.

3.4 CHAPTER SUMMARY

This chapter has presented two challenges that face every modern controls engineer. First we looked at digital sampling error and how it can be reduced by filtering and application of the Tustin Transform. Then we considered the difficulties that are caused by processing delays, and outlined our approach to reduce these delays as much as possible. The experimental results below include some results regarding processing delays.

CHAPTER 4 : FORMULATION OF THE CONTROL LAW

This chapter describes our dual-arm control law in a step-by-step fashion. First, it explains the hardware that we had to work with and the coordinate system that was applied. The chapter then introduces fuzzy logic control and gives an example in one dimension. Finally, the extension to a full spatial implementation is described.

4.1 DESCRIPTION OF THE ROBOTIC SYSTEM

U.T. Austin's Industrial Reconfigurable Anthropomorphic Dual-arm (IRAD) robot was used to demonstrate the proposed control law (Figure 4-1). IRAD comprises two Motoman SIA5D manipulators which are controlled by Agile Planet AX controllers. These robots are stiff, accurate industrial manipulators. Their size and payload are similar to a human arm, but their motions are more repeatable (±0.06 mm) ["SIA5D Datasheet," 2012]. The AX controllers are special because they can provide up to a 1 kHz control signal to the servos ["AX-I3 Solution," 2013]. This is significantly faster than a typical 60 Hz industrial robot controller. Agile Planet also provides the motion control software for these controllers.

Motion control programs for the AX controllers are executed on a Windows CE operating system. Windows CE, which typically runs on cell phones and smaller devices, is actually embedded in the Windows 7 operating system of a PC. This architecture was developed by Kuka Robotics ["CEWIN – The Real-Time Extension for Windows XP," 2013] and the embedded operating system is sold under the trade name "CeWin". It provides a familiar programming environment and interface along with the real-time control that is important for motion planning.

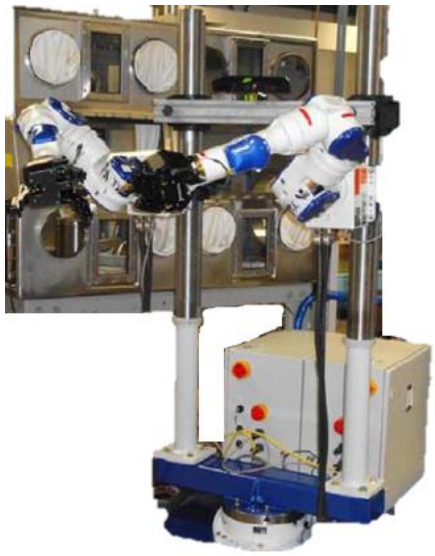

Figure 4-1. IRAD robot in front of a glovebox for nuclear material handling

An ATI Gamma SI-65-5 force and torque sensor ("F/T sensor") is mounted on the wrist of each arm, as shown in Figure 4-2. These sensors take force and moment readings for all three Cartesian directions at one kilohertz with 0.025N/0.7N*mm resolution ("F/T Sensor: Gamma," 2013]. ATI's software interface provides the ability to cancel the bias readings before every experiment, and the ability to transform the force data with on-board electronics. Both of these capabilities reduce the development time and the computational cost for the rest of the software.

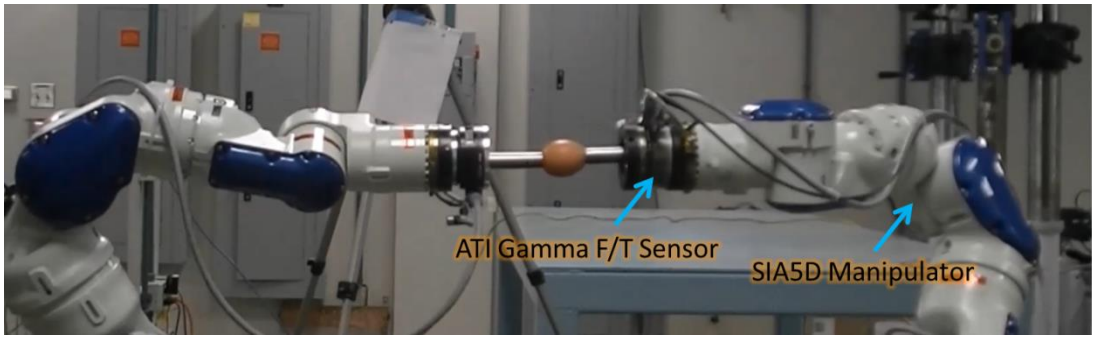

Figure 4-2. Experimental Setup

4.2 COORDINATE SYSTEM CONVENTION

For this dual-arm task, there are four coordinate systems to be assigned: one for each force/torque sensor and one for each robot. Setting up these axes was actually one of the most difficult tasks in the whole project. Two methods were considered. The first is easier for engineers to understand but it introduces a bit more programming complexity. The second coordinate system is simpler to program but less intuitive. We will describe both systems here, starting with the system that was implemented.

An object-fixed reference frame was chosen where the X-axis is directed along the normal between Arm 1 and Arm 2 (Figure 4-3). We define a common toolpoint at the midpoint. Both robots are programmed to move in this tool-fixed coordinate system. The adoption of this common toolpoint makes the problem easier to visualize and motion planning is easier because we can send identical commands to both arms. A scalar "L" is defined as half the distance between the end effectors. We call this the "lever arm" and it is used to determine the rotational velocities.

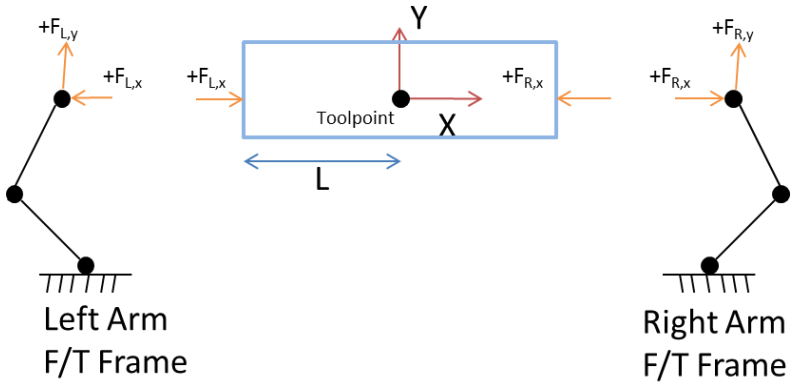

Figure 4-3. Coordinate system convention

Each force/torque sensor has a unique reference frame. The X- and Y-axes of the two force/torque sensors are anti-parallel, while the Z-axis is parallel. It is incongruous that the Z-axis is different, and it requires one additional heuristic in the control algorithm to handle this special case. In the C++ programming language, we implement the extra heuristic with one *if...then* statement. The next section gives a more specific explanation.

Unfortunately, it is impossible to have all three axes be anti-parallel when two "right hand" coordinate systems are used. A benefit of this setup is that a tensile force registers as a positive X-force on both force/torque sensors. That conforms to the convention from mechanics of materials, which is familiar to many engineers. It is more intuitive to talk about axial forces in this manner than it would be with the second coordinate system.

The second method (as shown in Figure 4-4) was not used, but it is discussed because it has some advantages. Note that all three sets of axes are aligned, so the internal and net wrenches in all three dimensions can be determined using the same logic. The downside is the loss of the ability to distinguish an axial force as tensile or compressive based on its sign. An operator would have to determine what sensor the force data is coming from (left or right) then compare to Figure 4-4 to determine if it is tensile or compressive. It is our opinion that the first coordinate system is more intuitive for the end-user.

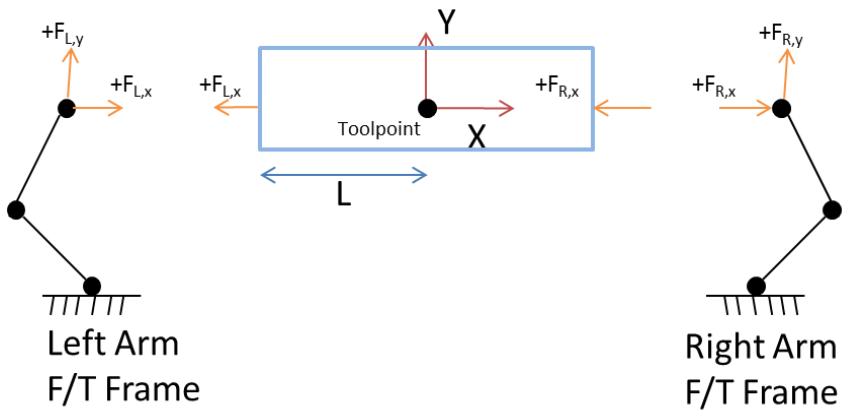

Figure 4-4. Alternate coordinate system

4.3 CALCULATION OF INTERNAL FORCE

Generally, an internal force does not contribute to the motion of the object. For our case, with two arms potentially applying force to the grasped object, the internal force in each dimension can be calculated as *the extent to which the force from the left arm is equal and opposite to the force from the right arm*. Sign conventions for the internal force must be made clear. The logic that we used in calculating the internal force is as follows:

X Force

An internal X-force is defined as positive if it puts the object in tension or negative if it puts the object in compression. With that convention in mind, here is the logic for calculating internal X-force:

- If (F_L>0 and F_R>0), F_{INT}= minimum(F_L,F_R)
    - Both forces are tensile.
- Else If (F_L<0 and F_R<0), F_{INT}= -minimum(|F_L|,|F_R|)

- Both forces are compressive.

- Else, $F_{INT}=0$
  - At least one of the forces is zero and/or there is a net force.

X Moment

An internal X-moment is defined as positive if it creates a counterclockwise moment about the toolpoint X-axis. Here is the logic for calculating internal X-moment:

- If ($M_L>0$ and $M_R<0$), F_{INT}= minimum($|F_L|$,$|F_R|$)
  - Positive moment about toolpoint X-axis.
- If ($M_L<0$ and $M_R>0$), F_{INT}= -minimum($|F_L|$,$|F_R|$)
  - Negative moment about toolpoint X-axis.
- Else, $F_{INT}=0$
  - At least one of the forces is zero and/or there is a net moment.

Y Force and Z Moment

An internal Y-force is defined as positive if it creates a counterclockwise moment about the toolpoint Z-axis. For calculating the internal Y-force, the same logic applies as for the X-force (since all three X-axes are parallel, as are the Y-axes).

Z Force and Y Moment

An internal Z-force is defined as positive if creates a counterclockwise moment about the toolpoint Y-axis. The Z-forces need their own logic since, unlike the other axes, the Z-axes are antiparallel:

- Else If ($F_L>0$ and $F_R<0$), F_{INT}= minimum($|F_L|$,$|F_R|$)
  - Internal counterclockwise moment about Y.

- Else If (F_L>0 and F_L*F_R<0), F_{INT}= -minimum($|F_L|$,$|F_R|$)
    - Internal clockwise moment about Y.
- Else, F_{INT}=0
    - At least one of the forces is zero and/or there is a net moment.

4.4 ASSUMPTIONS

Our control law depends on the assumptions of a rigid grasp and a rigid object. If this were not the case, then the lever arm would change as forces are exerted; this change in distance would cause error when the rotational velocities are calculated. If the target trajectory does not involve rotations about the Y or Z axes, this assumption can be dropped, because the other axes do not depend on the lever arm.

These rigidity assumptions are not necessary under the following conditions:

- The trajectory does not include any rotations.
- The object's elasticity is well-known, so that its change in length can be predicted.
- The grasps are point grasps, i.e. the rotation of the grippers relative to the object is insignificant.

4.5 ONE DIMENSIONAL FUZZY LOGIC

In Chapter Two we explained some of the features of fuzzy logic. Namely, that it is model-free, and it relies on operator experience to produce a smooth control signal. Here we will describe our fuzzy logic controller for the axial force, because that is the easiest case to visualize. Refer to Figure 4-3 for the coordinate system conventions.

The first step was to write a set of rules, based on the experience of the author, that adjusts the robot's velocity in order to control the compressive force on the object. There were a total of seven such rules in the "fuzzy set", as shown in Table 4-1. Why seven? There is no hard-and-fast reason. Five rules might have been sufficient and it would have reduced the computation time; nine almost certainly would have been excessive. Seven rules will give a more well-defined and likely more robust output than five rules. An odd number was used because we assumed the rule set would be generally symmetric. There are three rules for tension, three rules for compression, and one rule that applies when the axial force is ideal.

Table 4-1. Linguistic description of the fuzzy set for X-translation

Linguistic Rule
If there is way too much tension, move inward quickly.
If there is a moderately excessive amount of tension, move inward rather quickly.
If there is a bit too much tension, move inward slowly.
If the axial force is ideal, do not change the velocity.
If there is a bit too much compression, move outward slowly.
If there is a moderately excessive amount of compression, move outward rather quickly.
If there is way too much compression, move outward quickly.

The next step was to assign numerical values to these linguistic descriptions. In our case, the author manually assigned these values based on his experience. It is possible to adapt these rules over time or learn them from scratch, as well, and that is something

we suggest for future work. Table 4-2 shows the conversion from rhetorical to mathematical constructs.

Table 4-2. Numerical Definition of the fuzzy set for X-translation

Linguistic Rule	Numerical Definition
If there is a highly excessive amount of tension, move inward quickly.	If (axial_force_error = 12.5lbf), add 1% speed in the +X direction.
If there is a moderately excessive amount of tension, move inward rather quickly.	If (axial_force_error = 2.7lbf), add 0.6% speed in the +X direction.
If there is a bit too much tension, move inward slowly.	If (axial_force_error = 0.8lbf), add 0.1% speed in the +X direction.
If the axial force is ideal, do not change the velocity.	If (axial_force_error = 0 lbf), do not alter the speed in the X direction.
If there is a bit too much compression, move outward slowly.	If (axial_force_error = -1.0lbf), add -0.1% speed in the +X direction.
If there is a moderately excessive amount of compression, move outward rather quickly.	If (axial_force_error = -3.0lbf), add -0.6% speed in the +X direction.
If there is a highly excessive amount of compression, move outward quickly.	If (axial_force_error = -11.5lbf), add -0.1% speed in the +X direction.

For each of these seven rules, we define a Gaussian bell curve that is centered at the force as defined in Table 4-2. These curves are used to transition from a discrete model of the behavior, as in Table 4-2, to a probabilistic model. For example, the following equation defines the Gaussian curve for the "moderate amount of compression" rule:

Equation 2. General form of Gaussian curve

$$membership = e^{-\frac{1}{2}\left(\frac{-F_{error}-center\ position}{width}\right)}$$

Equation 3. Explicit expression for "moderate compression"

$$= e^{-\frac{1}{2}\left(\frac{-F_{error}-3}{1}\right)}$$

Equation 4. Simplified expression for "moderate compression"

$$= e^{-\frac{1}{2}(-F_{error}-3)}$$

This bell curve has a peak value of one, centered at F_{error}=-3lbf (Figure 4-5) with two variables that can be adjusted by the designer. “Width” controls the width of the curve: a smaller value here will make this rule active over a larger range of forces. “Center position” locates the center peak of the curve. Equation 3 shows all parameters for the bell curve explicitly. However, it requires one more operation than Equation 4. So when we programmed these fuzzy membership rules, we used the form of Equation 4.

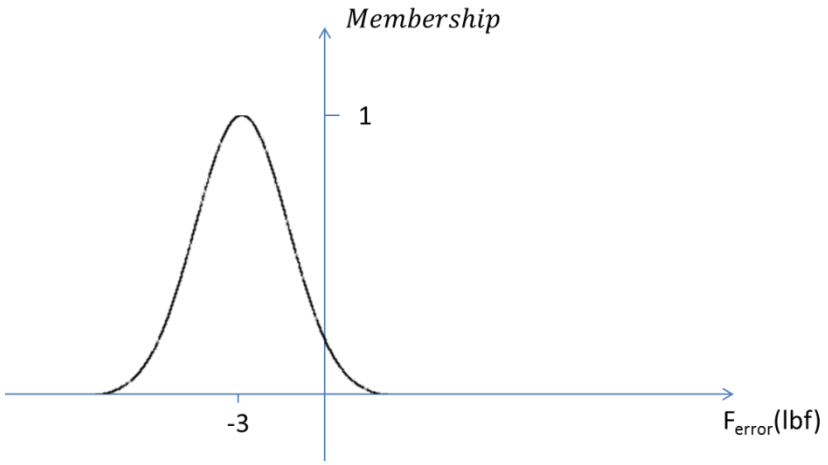

Figure 4-5. Membership rule for "moderate compression"

The complete set of fuzzy membership rules is shown in Figure 4-6. The “degree of truth” that is calculated from each of these bell curves is a measure of how much that

rule applies and it may vary from zero to one. For example, at F_{int}=1.0lbf, the three rules on the left-half plane barely apply at all; their degree of truth is approximately zero. Likewise, the rules for "moderate tension" and "excessive tension" are too far out; their degrees of truth are also approximately zero. The rule for "ideal axial force" is slightly active, having a degree of truth ~0.2. Finally, the rule for "a bit too much tension" is highly active, having a degree of truth ~0.7. To calculate the overall control effort, we take a weighted average that is based on the memberships. So, in this example, the control signal would be dominated by the "a bit too much tension" rule with a small contribution from the "ideal axial force" rule. The remaining five rules do not apply at this instant, so they have a minimal effect on the weighted average.

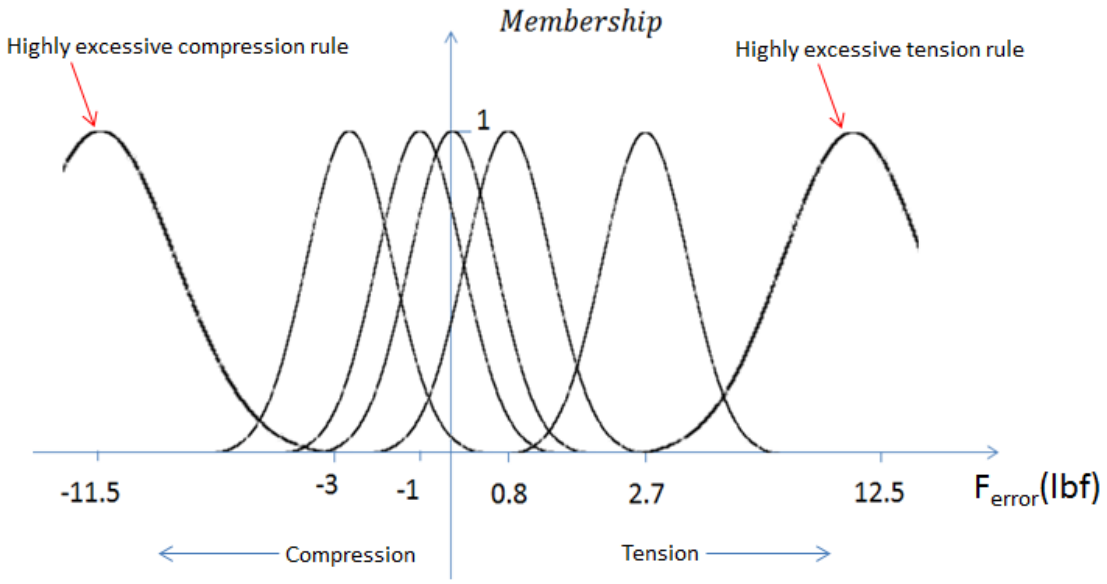

Figure 4-6. Complete set of fuzzy membership rules

This reveals one of the nicest features of fuzzy logic: the probabilistic approach converts a discrete set of rules to a smooth, continuous function. This conversion is sometimes referred to as "fuzzification". The fuzzification formula for calculating the net controller effort based on a weighted average is:

Equation 5. Weighted average for fuzzy logic

$$\Delta v_{internal} = \frac{\sum_{i=0}^{6} DoT[i] * \Delta v[i]}{\sum_{i=0}^{6} DoT[i]}$$

In Equation 5. Weighted average for fuzzy logic DoT stands for "Degree of Truth". This is the degree of truth that is calculated from the Gaussian curve, based on the current force error. Δv is the change in speed that was defined for each rule, as in Table 4-2. Figure 4- summarizes the algorithm with a block diagram.

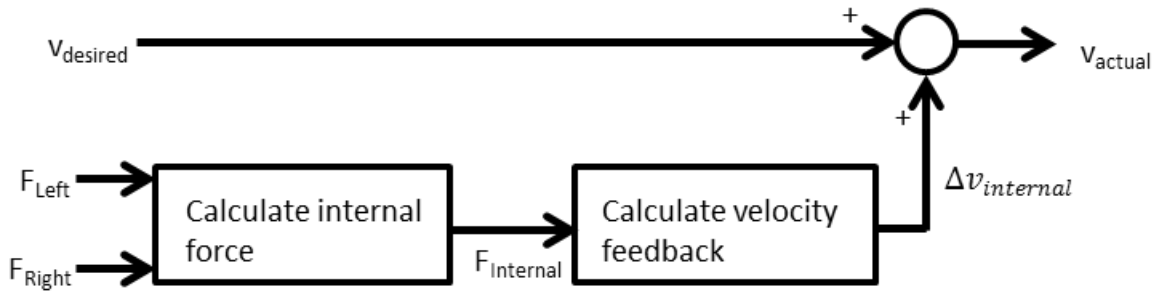

Figure 4-7. Block diagram for the internal force controller

4.6 SPATIAL FUZZY COMPLIANCE

How can this fuzzy logic be extended to the spatial case, i.e., to three translational axes and three rotational axes? We have already covered X-translation, so let us look at X-rotation now. As shown in Figure 4-8 (which is reproduced here from an earlier section), there is no lever arm for X-rotation because the X-axes for the force/torque

sensors are aligned with the X-axis of the toolpoint. This is a nice simplification; X-rotation is independent from motion of the remaining five axes. So it is straightforward to apply fuzzy logic to calculate the admittance in this direction. We keep the same fuzzy structure that was used for X-translation, merely modifying the value of some parameters.

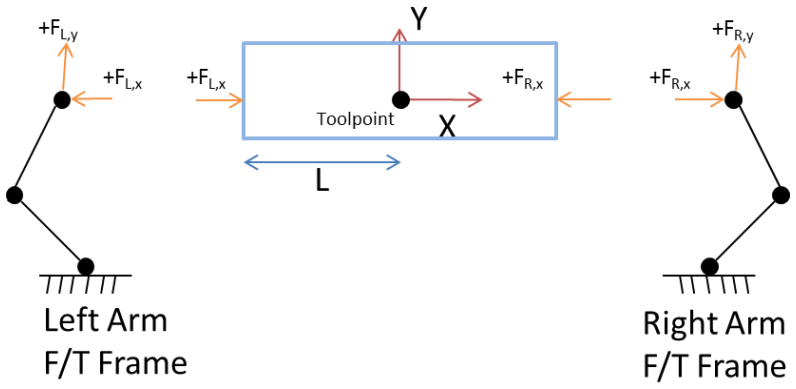

Figure 4-8. Coordinate system convention

Table 4-3. Fuzzy set for X-rotation

Linguistic Rule	Numerical Definition
If there is a highly excessive (+) moment error about the toolpoint X-axis, each robot should (+) rotate about its own X-axis quickly.	If (X_moment_error = 12.5 lbf*in), add 10% speed in the +X direction.
If there is a moderately excessive (+) moment error about the toolpoint X-axis, each robot should (+) rotate about its own X-axis rather quickly.	If (X_moment_error = 2.7 lbf*in), add 6% speed in the +X direction.
If there is a bit too much (+) moment error about the toolpoint X-axis, each robot should (+) rotate about its own X-axis quickly.	If (X_moment_error = 0.8 lbf*in), add 1% speed in the +X direction.
If the moment about the toolpoint X-axis is ideal, do not change the angular X-velocity.	If (X_moment_error = 0 lbf*in), do not change the angular X-velocity.
If there is a bit too much (-) moment error about the toolpoint X-axis, each robot should (-) rotate about its own X-axis quickly.	If (X_moment_error = -1 lbf*in), add -1% speed in the +X direction.
If there is a moderately excessive (-) moment error about the toolpoint X-axis, each robot should (-) rotate about its own X-axis rather quickly.	If (X_moment_error = -3 lbf*in), add -6% speed in the +X direction.
If there is a highly excessive (-) moment error about the toolpoint X-axis, each robot should (-) rotate about its own X-axis quickly.	If (X_moment_error = -11.5lbf*in), add -10% speed in the +X direction.

Unfortunately, there is no simplification for (Y-translation,Y-rotation) and (Z-translation, Z-rotation). Because the force/torque Y-axes are not coincident with the toolpoint Y-axis, there is a lever arm. The lever arm means that Z-rotation about the toolpoint must be accompanied by an equal-and-opposite Y-translation of the robots in order to maintain the orientation of the grippers in relation to the object. We will refer to this as "commensurate translation" and Figure 4-9 demonstrates the concept.

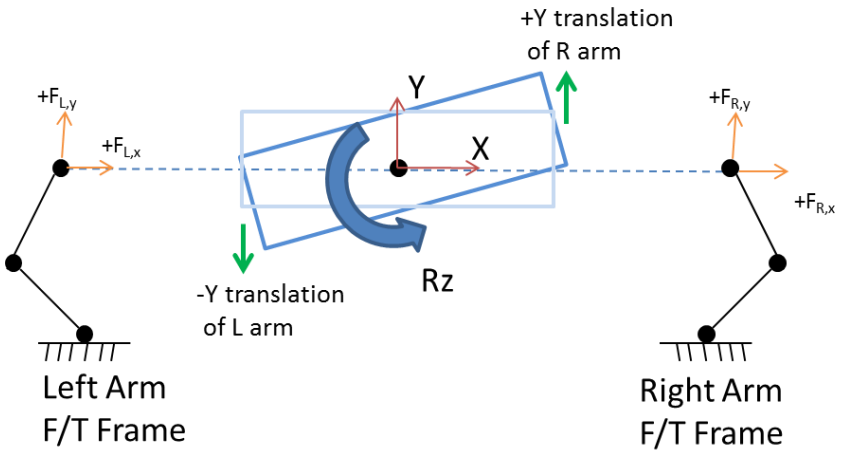

Figure 4-9. Commensurate Translation

Since rotations and translations in these dimensions are inherently linked, there is not a separate fuzzy rule set for force error and moment error. Rather, we calculate the Y- and Z- translational speed feedbacks from internal force. Commensurate rotations are then calculated with the familiar formula for linking tangential velocity to angular velocity:

$$v_{tan} = radius * \omega$$

We replace radius with the lever arm measurement, yielding:

$$v_z = \frac{LeverArm}{k} * \omega_y$$

$$v_y = \frac{LeverArm}{k} * \omega_z$$

Here, *k* is a unit conversion constant. For our system, it converts the translational velocity (in %) to a rotational velocity in rad/s. We found, experimentally, that *k* is approximately equal to 6000 when the lever arm is measured in millimeters. We can examine the effect of *Lever Arm* in the limiting case as *Lever Arm* approaches zero. Then, the object has zero thickness; the robot arms will be in contact with each other. The object may rotate at any rate without any commensurate translation. Alternatively, *Lever Arm* may approach infinity if the grasped object is very long. Then an infinite commensurate translation is required to achieve the smallest rotation.

4.7 EXTENSION TO SINGLE-ARM MOTION

We have described a rather complex control law for dual-arm manipulation. Can the same control law be applied to a single-arm manipulator? The answer is yes. For a single arm, of course, the internal force is undefined; it will always be zero. Then the velocity feedback due to internal force will be zero; the robot will track its target trajectory exactly. The final chapter recommends that external compliance be implemented as future work. This external compliance could also be extended to the single-arm case.

4.8 CHAPTER SUMMARY

This chapter has described the dual-arm robotic system on which we conduct our research. It explains a coordinate system that simplifies the dual-arm problem, and discusses the formulae for calculating internal force on the grasped object. We explain

how a discrete set of rules, derived from the experience of a human operator, can be "fuzzified" and extended to smoothly control the internal force on the object.

Fuzzifying the internal force and moment for the X-axis is straight forward. However, rotation and translation for the Y- and Z- axes are inherently linked, so we supplement the fuzzy set with the equation of tangential velocity for these axes. Thus, we use four sets of fuzzy rules and two tangential velocity equations to calculate the feedback velocity in six dimensions. From the standpoint of computational cost, it is nice to have fewer fuzzy sets because they are the most computationally expensive part of the algorithm.

CHAPTER 5 : STABILITY ANALYSIS AND ROBUSTNESS

An important consideration for any controller is stability analysis. Will it maintain the desired internal force, despite small perturbations? Typically this analysis is done in state space. Presently, the only state of interest is the internal force error, F_{error}. The controller output is an equal-and-opposite velocity to each arm, $\dot{x}_{int}$. The golden standard for stability is to prove "global asymptotic stability." That is, no matter what velocity and internal force the robot starts at, does the controller bring it to the desired internal force?

Stability analysis for fuzzy controllers is difficult, because they are not one function. Rather, they are a weighted average of several functions. One way around this dilemma is to find a simpler function that bounds the fuzzy controller outputs [Choi, 2000]. If we can find a stability condition for this simple controller, then we draw some conclusions about the fuzzy controller as well. We start this approach by defining a bang-bang controller that bounds the fuzzy output (see Figure 5-1). If the internal force is negative, the velocity command is a positive constant η. For a positive internal force, the command is negative η:

$$u(t) = \dot{x} = \left\{ \begin{matrix} -\eta, & F_{error} > 0 \\ \eta, & F_{error} < 0 \\ 0, & F_{error} = 0 \end{matrix} \right\} = -\eta * \text{sign}(F_{error})$$

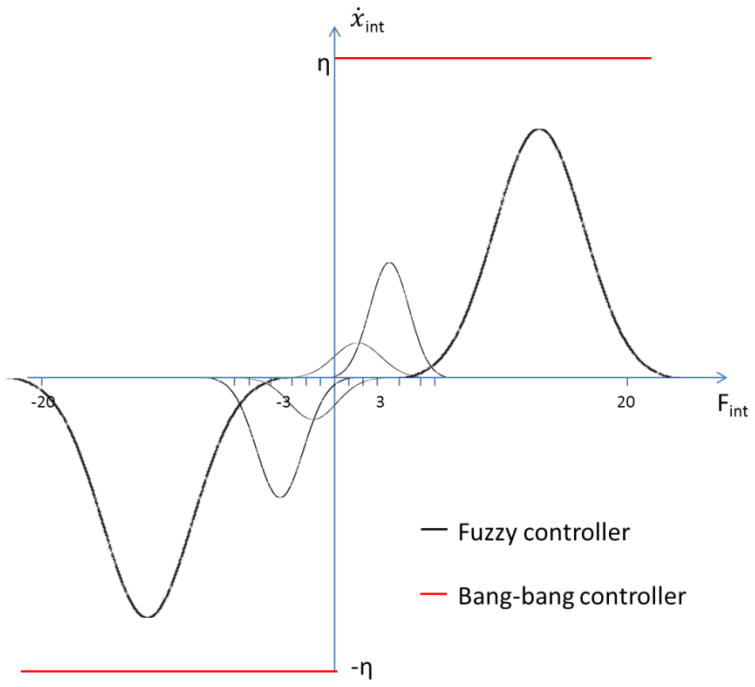

Figure 5-1. Bang-bang controller bounds fuzzy controller

In order to plot a phase portrait of the system's behavior in state space, we need a dynamic model of the system. Let us start with a simple model where the clamped object is symmetric, linear, and elastic, i.e. it can be modeled as an ideal spring. Because of symmetry, the problem can be regarded as symmetric with a fixed midpoint (Figure 5-2). The dynamic equation for this system is $F_{error} = k\, x_{error}$. We need this equation because it gives a way to relate the robot velocity to the force in the object. Namely, $\dot{F}_{error} = k\, \dot{x}_{int}$.

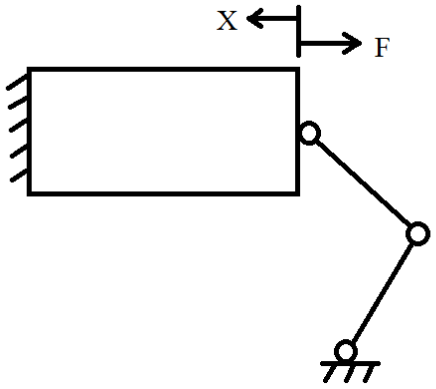

Figure 5-2. Symmetric spring model

A plot of several trajectories in state space makes it clear that the bang-bang controller is globally, asymptotically stable. No matter where it starts, the system is pushed towards equilibrium at the vertical axis, $F_{error}=0$ (Figure 5-3).

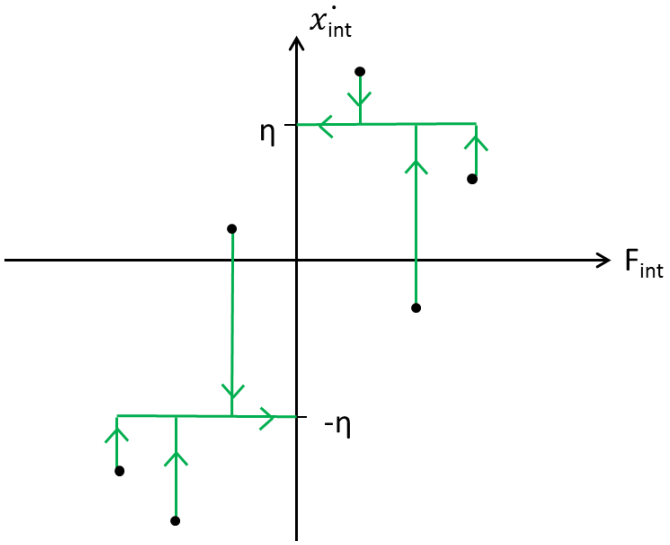

Figure 5-3. Phase portrait of robot admittance

5.1 LYAPUNOV STABILITY AND ROBUSTNESS PROOF

Once again, the control equation was:

$$\dot{x} = \delta - \eta * \text{sign}(F_{error})$$

Here we have added another term, δ, to account for any uncertainty or disturbances in the system. For example, δ might account for the net effect of poor sensor resolution, an inaccurate lever arm measurement, and computational delays. Now we will use Lyapunov Stability Theory to prove that this controller is stable despite these disturbances and uncertainty.

Define a candidate Lyapunov function, based on the distance away from the desired force:

$$V = \frac{1}{2} F_{error}^{\ 2}$$

For a stable system, the requirement is:

$$\dot{V} < 0 \ \ \forall \, F_{error} \in \boldsymbol{R}$$

$$\dot{V} = \mathrm{F}_{error}\dot{\mathrm{F}}_{error} = \mathrm{F}_{error}\left(\delta - \eta * \mathrm{sign}(\mathrm{F}_{error})\right) < 0$$
$$\mathrm{F}_{error}\delta - \eta\mathrm{F}_{error}\frac{\mathrm{F}_{error}}{|F\mathrm{F}_{error}|} = \mathrm{F}_{error}\delta - \eta|\mathrm{F}_{error}| < 0$$
$$\eta|\mathrm{F}_{error}| > \mathrm{F}_{error}\delta$$
$$\eta > \delta * sign(\mathrm{F}_{error})$$

Equation 6. Requirement for a Stable Bang-Bang Controller

$$\eta > |\delta_{max}|$$

Equation 6 is the requirement for a stable, bang-bang controller. If the controller effort, η, is greater than the largest expected disturbance, then the system will be stable. Since we have demonstrated stability even in the case of unknown disturbances, this controller is "robust".

5.2 FUZZY CONTROL

We have just seen the stability proof for the simple bang-bang controller. How will it vary when fuzzy logic is applied? The difference is that fuzzy logic will slow the system down as it approaches zero internal force. So the system still moves towards equilibrium, but it will take longer and be smoother than the bang-bang controller. The overlaid blue trajectories in Figure 5-4 are an example when there is no perturbance δ.

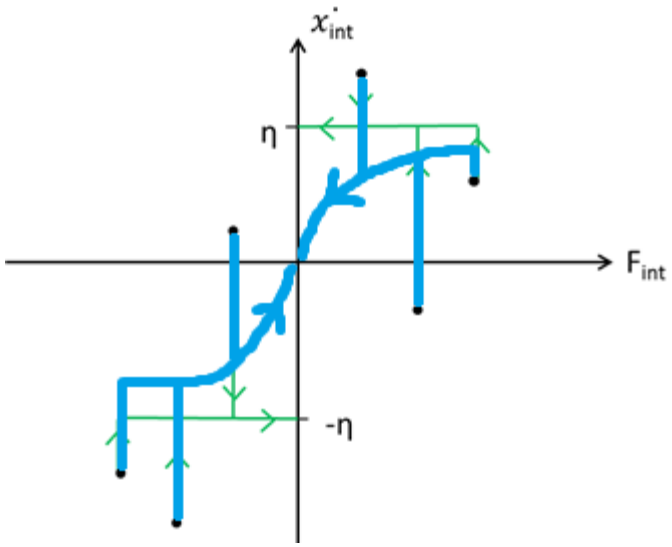

Figure 5-4. Fuzzy logic phase portrait

The same condition for global, asymptotic stability holds: if the controller effort is greater than δ_{max}, it will be asymptotically stable. If not, there will be some steady state error. Figure 5-5 demonstrates the steady-state error concept.

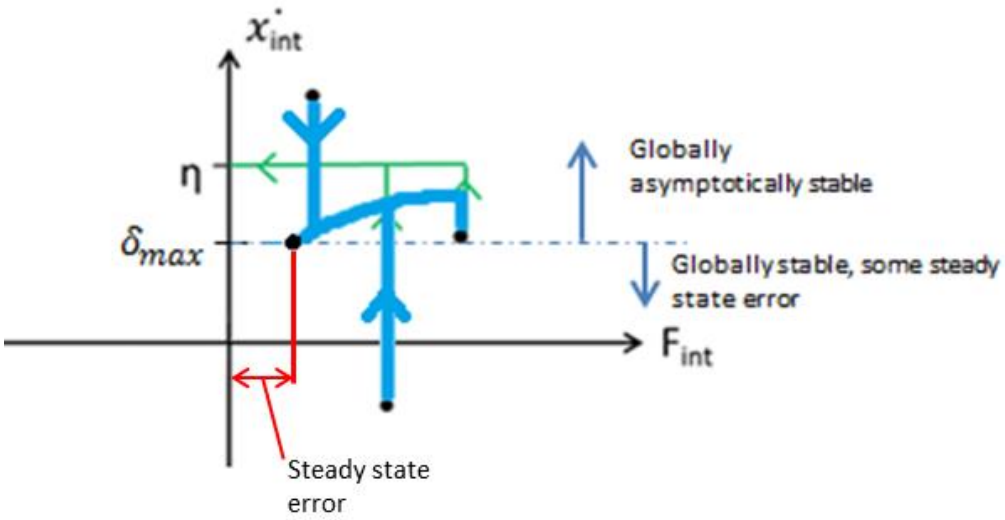

Figure 5-5. Phase portrait with perturbance

CHAPTER 6 : EXPERIMENTAL RESULTS

6.1 BASELINE SETUP

Data for every experiment was collected for the same baseline task: a urethane rod is clamped axially and moved through a specified trajectory. Specifically, the polyurethane rod is of 80A durometer hardness, 48mm in diameter, and 288mm in length. This object is hard and quasi-rigid, but it does provide a little "give" under impact; it was chosen because it would not damage the robots in the event of a fault.

The robot arms slowly approached and clamped this urethane rod axially with a target compressive force of 13.3 N. Then the arms moved the object through a spatial trajectory as data was collected. This spatial motion profile incorporates all six axes. Each axis follows a sinusoidal trajectory at a slightly different frequency. These slightly different frequencies cause a complex spatial trajectory with a very long period. It is designed to move the robot arms through a wide range of position and orientation, covering most of the reachable workspace and providing a thorough test of the algorithm. Unless otherwise specified, the specific velocity commands for each axis were:

Equation 7. Baseline trajectory

- Tx= 12.5cos(0.1* π*time) mm/s
- Ty= 5cos(0.11* π*time) mm/s
- Tz= 5cos(0.1* π*time) mm/s
- Rx= 50cos(0.1* π*time) rad/s
- Ry= 12.5cos(0.1* π*time) rad/s
- Rz= 12.5cos(0.1* π*time) rad/s

The amplitudes were chosen to ensure the object stays within the reachable workspace of both robot arms. The frequencies were then selected to provide a range of accelerations and velocities similar to the sponsor's target applications: precision manufacturing tasks, likely inside a glovebox. The target internal force for all dimensions is zero, except in the axial direction; that target force is a compressive 13.3 N.

6.2 List of potential experiments

First, we listed the ideal set of experiments that could help us quantify and understand the performance of this controller. For example, how does the control frequency impact our ability to manage internal forces in a held object? Another example is to examine the relative import of the joint configuration and inverse kinematics resolution relative to the end-effector tracking resolution. In other words, does the location of the experiment in the workspace impact the effectiveness of the control strategy?

Given additional time and resources, it may be possible to empirically evaluate the robustness of the controller for a broad (even general) task space. Given the expense and advanced capabilities of the controller used to perform these experiments, it is also possible to experimentally restrict its performance. Doing so helps us understand the minimal requirements for a controller and thus reduce cost or better understand the robustness of a high performance controller. Table 6-1 covers this range of experiments and the reasons why they would expand our knowledge of the system.

Table 6-1. Proposed categories of experiments

Experiment	**Purpose**
Plot of force and position tracking in six dimensions	• Benchmarking against other systems • Suitability for industrial tasks • Validate that the interactions between control heuristics for different axes do not cause instability • Validate the control law for a variety of trajectories
Impact of control parameters at the initial and final task instances	• In many cases, the controllers fail due to transience or highly nonlinear conditions that exist only at the beginning or end of a given task
Evaluate the force tracking vs control rate vs maximum acceleration	• Test of robustness • Pre-estimation of minimum control rate for a given task • Ensure that for a given trajectory, a given control rate does not allow the internal force to exceed task requirements • Validate analytical models that determine the necessary control frequency for a given force and/or motion task
Plot of minimum control rate vs. maximum acceleration for a given criterion	• Test of robustness • Pre-estimation of minimum control rate for a given task
Evaluate force and position tracking vs error in lever arm measurement	• Test of robustness • Develop guidelines for the measurement accuracy of the "lever arm" parameter
Plots of force/position tracking as fuzzy laws are adapted by a machine learning algorithm	• Attempt to improve performance • Comparison to human-tuned control laws
Plots of force/position tracking as object is changed	• Applicable to rigid and non-rigid objects?
Examine if the proposed control laws will work on different hardware	• Test of robustness • Gain insight into compliance and backlash of the SIA5D and other systems
Compare the proposed controller performance against relevant state of the art or other common controllers used for benchmarking	• Such experiments are necessary and yet difficult to complete as many common benchmark controllers require the user to spend an excessive amount of time "tuning" the system. Thus, the results of these experiments would need to include some quantification of effort related to results
Evaluate force/position tracking with position control and compare to the velocity control results	• Validate our theories regarding the smoothness and robustness of velocity control as compared to position control

Due to practical considerations, we trimmed this list down to a subset of the four most important experiments:

- Ability to accurately track a desired internal force during an experimentally repeatable task
- Impact of control rate and initial speed on the transient force at the beginning of motion
- Task performance vs. control frequency
- Control rate vs. task failure

6.3 FORCE TRACKING EXPERIMENT

Figure 6-1 shows plots of force in the X axial direction and the moment about Y. The other forces were not included on this plot because they were extremely close to zero. This plot shows that our control law maintains the axial compressive force within four Newtons of the target force. Put into perspective, it takes about six times this force to crush an egg. The reader may notice a transient in the compressive force at the beginning of the plot; the next experiment explores this phenomenon.

The plot of Y-moment variance is much more than the X-force. It appears our controller has more trouble maintaining this moment; it maintains the Y moment within 17 N*cm of its target value. To put that in perspective, it takes about seven times this much torque to unscrew the lid on a glass bottle [“Torque Guide,” 2013].

Recall, the forces in the other four dimensions were not plotted because they were very nearly zero. We can conclude that the controller does an excellent job controlling the moment in five dimensions and a satisfactory job in the 6^{th} direction: the Y moment. We speculate that difficulty with the Y moment arose from an inaccurate measurement of

the distance from robot endface to common toolpoint. An accurate measurement here is crucial because it links the equal-and-opposite translation of the arms to their rotations about the Y axis. Also, the lever arm tends to amplify any error. It is easier to maintain a force than a moment. The final chapter includes some recommendations for future work that may help improve this measurement.

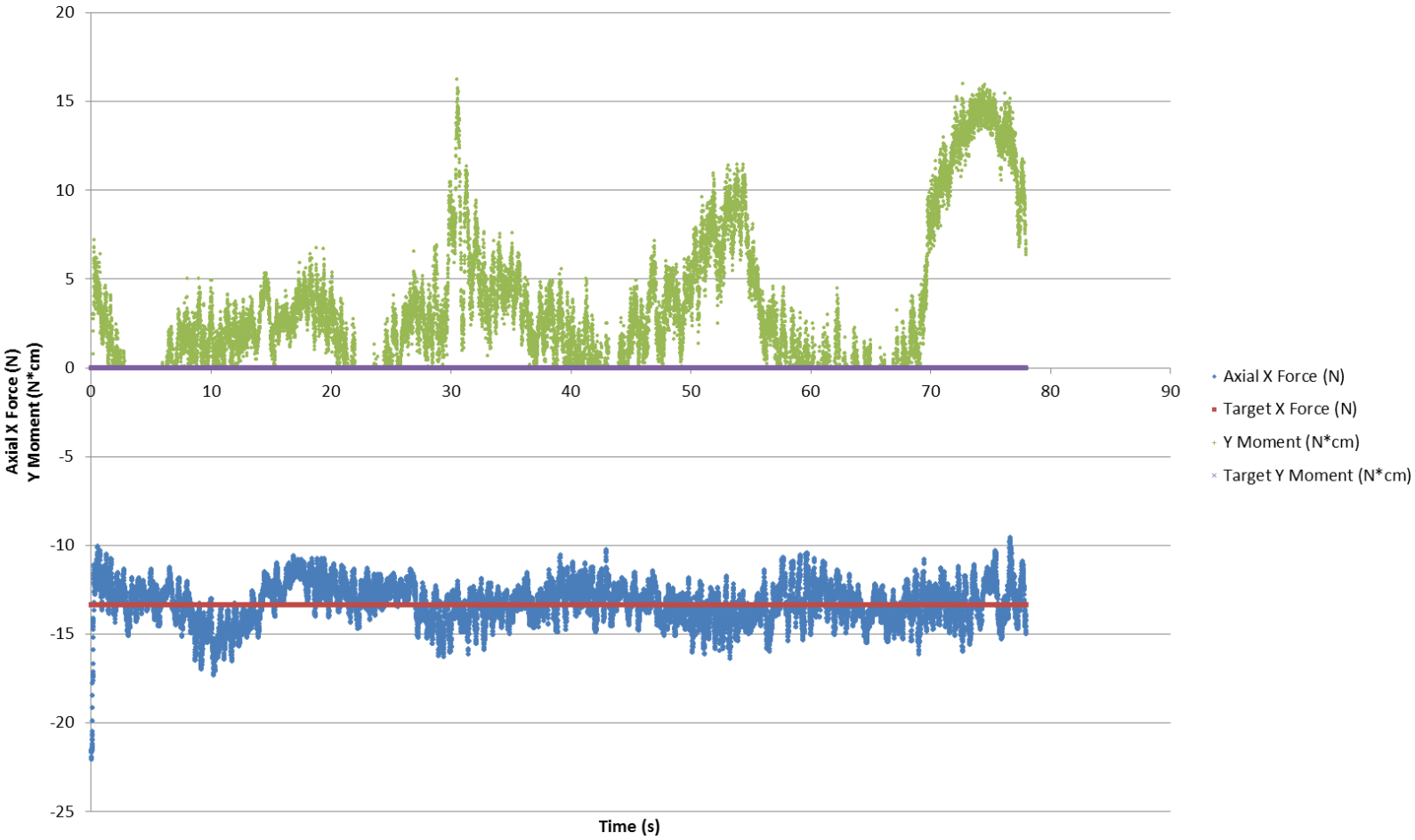

Figure 6-1. Accuracy of Force Tracking, 250 Hz Control Rate

6.4 INITIAL TRANSIENT EXPERIMENT

We pointed out a transient in the compressive force at the beginning of motion (Figure 6-1). There are two potential explanations for this transient:

- When the arms first approach to clamp the object, they overshoot the target force and create some excessive compression.
- The function cosine was used for our trajectory calculation. Of course, cosine(t) has a maximum at t=0, so the arms "jumped" from zero speed to full speed at the beginning of the experiment.

To discover which two explanations (or both) was the culprit, we conducted an experiment with three trials. The first trial is the control; this is the baseline case with the regular 0.5% approach speed and a cosine trajectory. The second trial used the same cosine trajectory but the approach speed was reduced by a factor of five. The final trial used the same approach speed as the control but we changed to a sine trajectory. Since sine(t=0)=0, this trajectory starts off slowly.

Figure 6-2 shows the results of the experiment. Clearly both factors (gentle start vs. impulse start and approach speed) contributed to the transient at the beginning of motion. However, the approach speed was a more significant factor. When the approach speed was reduced by a factor of five, the force overshoot at the beginning of the experiment was completely eliminated. In fact, it causes a small undershoot, since the target clamp force was 13.3 N. Thus, to reduce or eliminate initial transience issues leading to excessive internal forces, the objects must be clamped slowly. This result conforms with our intuition.

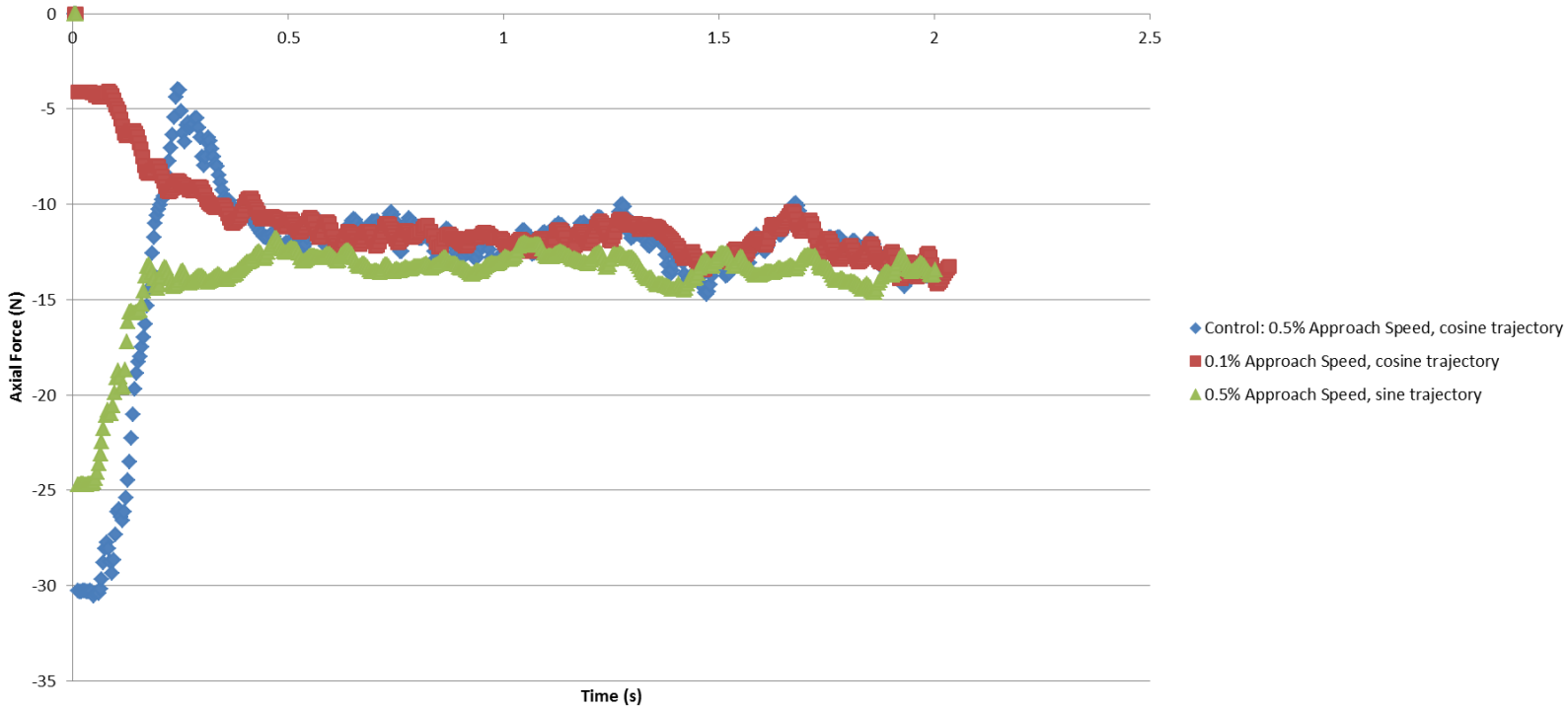

Figure 6-2. Effect of approach speed on initial force error

6.5 POSITION TRACKING EXPERIMENT

Now we clamp the same object and move over the same trajectory, collecting position data rather than force data. Position data is collected from the hardware using the software interface method *GetCurrentHandPosition()* for the left arm. Thus, the position is calculated from forward kinematics from the left robot's servo data. According to the SIA5 Motoman datasheet ["SIA5D Datasheet," 2012], this position should be repeatable to ±0.06mm. We chose to collect data only for the axial X-direction because it has the longest range of motion, but we could have done the same for any other axis. The plot of actual position versus target position (Figure 6-3) has a couple salient features. First, the object quickly acquires some position error. As the previous experiment shows, this initial transient is largely due to overshoot while clamping at the beginning of the trajectory; as the robots correct for that force error, about 70mm of positional error accumulates. Because there is no position feedback, that error is never corrected. At this point in time, we rely on "dead reckoning." The final chapter of this report recommends some future work that would correct this issue.

The other interesting feature is the lag that develops between target position and actual position. We speculate that this lag starts due to the tiny yet finite compliance in the object and the actuators. Backlash in the drivetrain may be the main driver of this phenomenon; as the robots reverse direction, their gear teeth are unloaded and they must rotate a finite distance before engaging again. This is another reason why some position feedback needs to be incorporated.

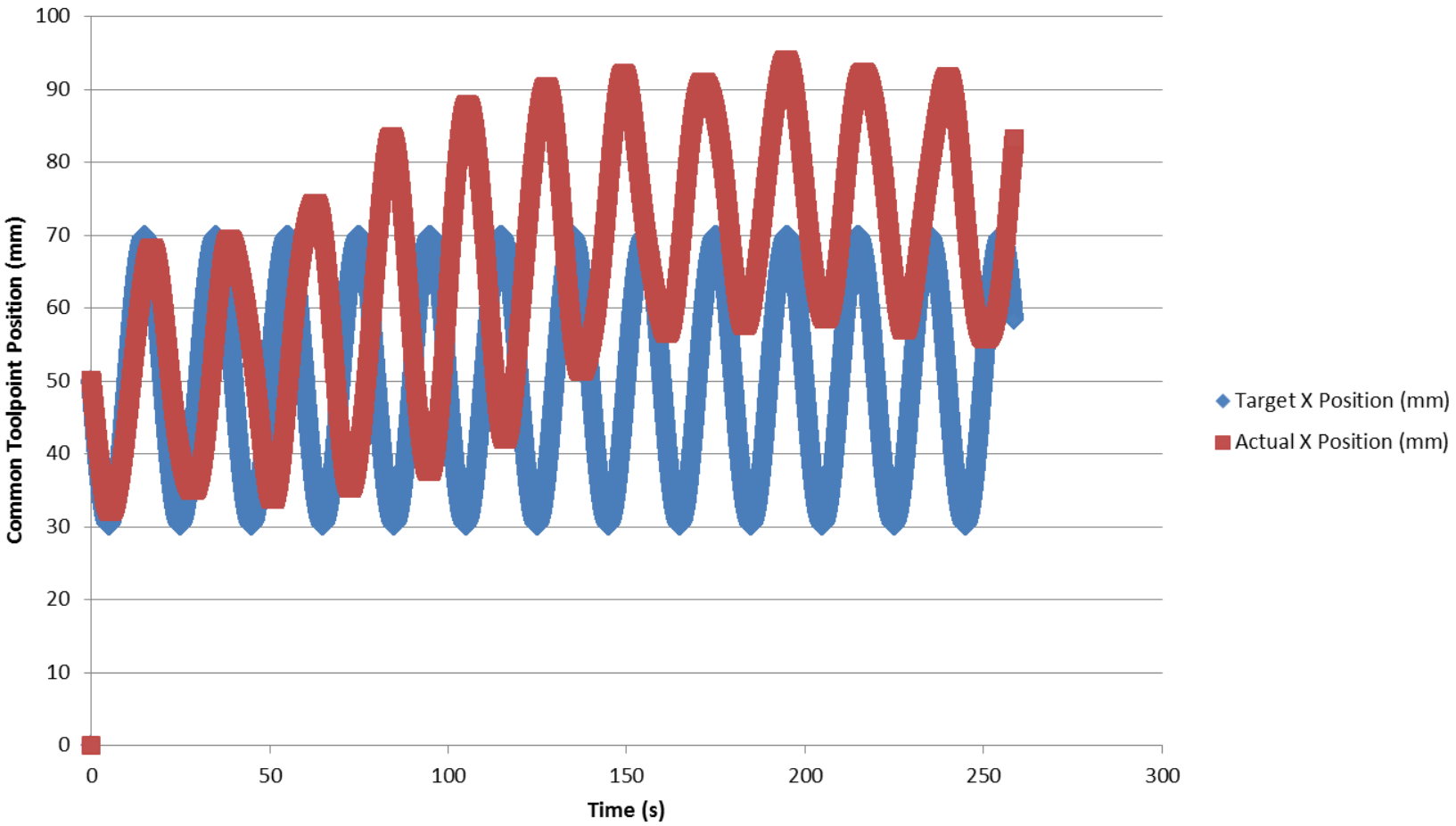

Figure 6-3. Toolpoint Tracking Accuracy

6.6 VARIABLE CONTROL RATE EXPERIMENT

This experiment involved clamping the same object and following the identical trajectory while varying the control frequency. This tests the robustness of the controller and opens the door to future experiments to quantify minimal controller requirements (see Recommendations for Future Work in the final chapter). In order to vary the control rate, the *Sleep(int timeinmilliseconds)* function from the Windows C++ library was inserted in the main loop. This allowed us to vary the control rate from 18 Hz down to 250 Hz. A typical industrial robot operates at 60 Hz.

The results as shown in Figure 6-4 were both interesting and surprising. A control rate as low as 33 Hz seemed to maintain the compressive force just as well as the 250 Hz rate for the task described in Equation 7 above. At 18 Hz, we observed a failure where the robot arms briefly lost contact with the object and it slipped a few millimeters (this occurred at t=1s on Figure 6-4). We finished collecting data for that trial, but did not try any lower control rates.

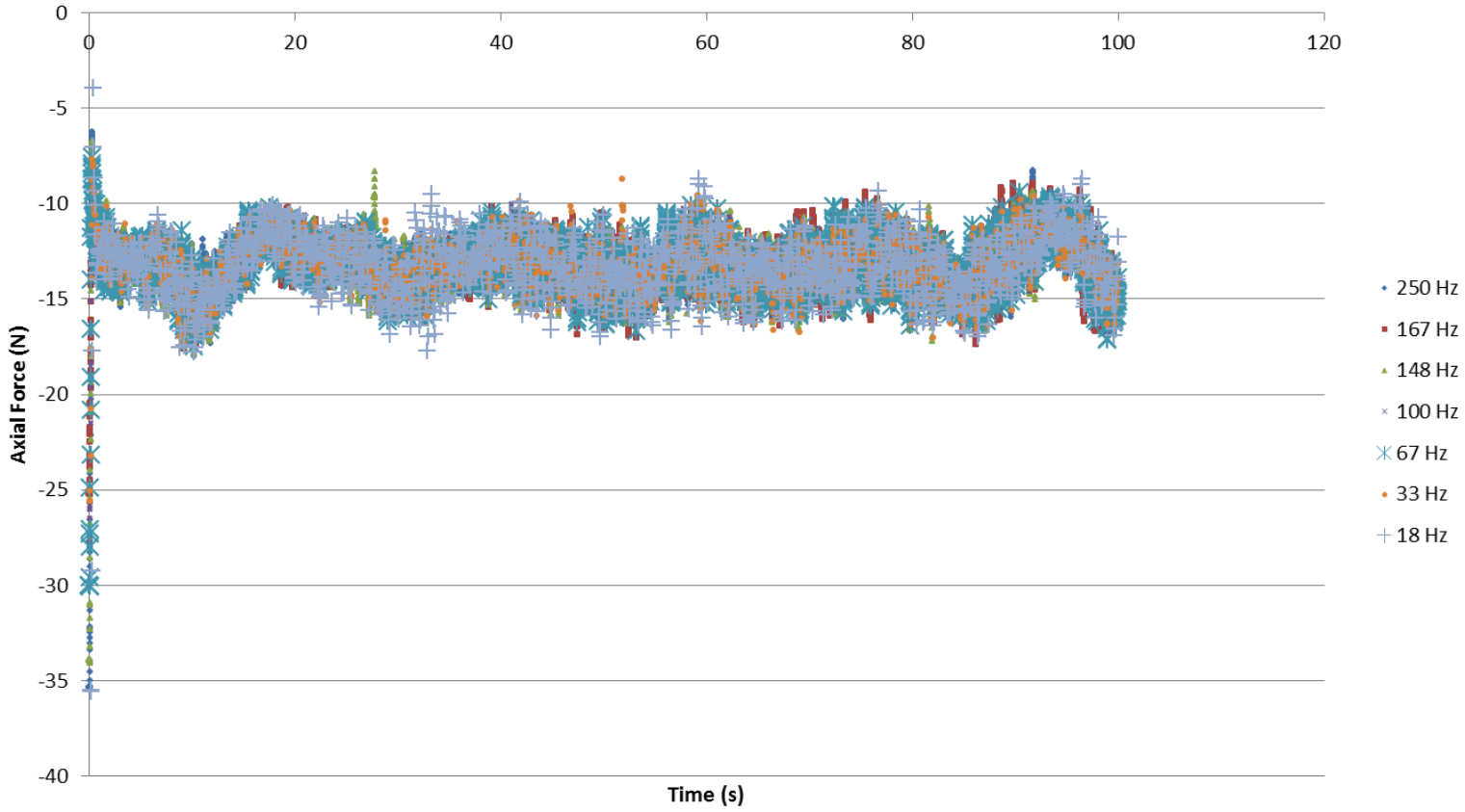

Figure 6-4. Effect of Variable Control Rate

To quantify our statement, we compared the variance for each control rate (Figure 6-5). There is a very weak correlation between control rate and clamp force variance. The vast majority of the performance degradation occurs at the 18 Hz rate. Above this control rate, the performance was quite robust to variations in the control rate. In fact, if the 18 Hz control rate is disregarded, the trend is actually slightly reversed. There was no performance gain from running the experiment at 250 Hz rather than 33 Hz.

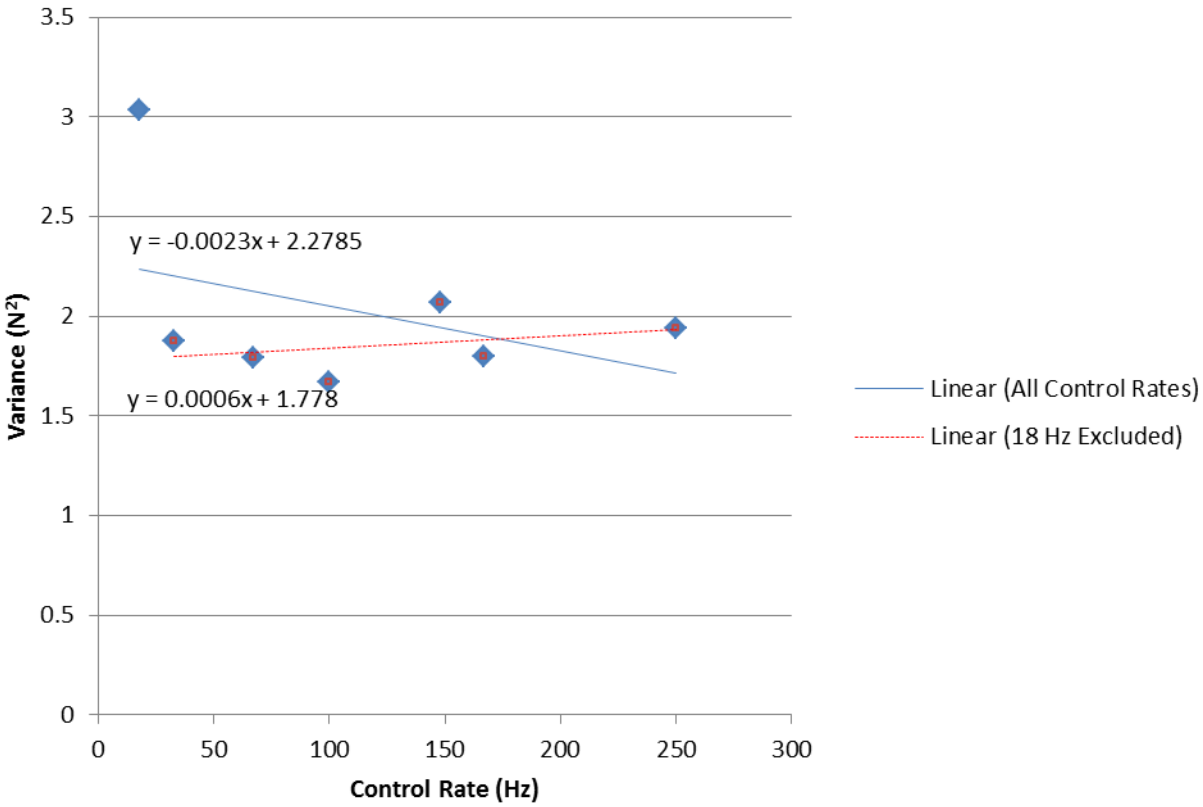

Figure 6-5. Effect of Control Rate on Variance of Clamp Force

This is an encouraging result! It means for this task, with the Agile Planet AX controller, the algorithm was robust enough to handle a wide variety of control rates, and industrial 60 Hz controllers should be sufficient to maintain a clamp force within ±5 N at

this speed. The next experiment looks at the effect of control rate when the trajectory's speed is varied. The graph also raises some questions: If we tweaked the gains, would the system be capable of even better performance at 250 Hz? And why, exactly, is it so robust with regards to variable control rate?

We can speculate about the answer to the second question; much of the robustness certainly arises from our use of velocity control. To explain why, we'll refer again to Figure 6-56, which first appeared in Chapter One. It shows how velocity control is more robust against controller delays because it approximates the slope of the function. Position control will maintain its position if there is a control signal delay, which leads to a much larger position error.

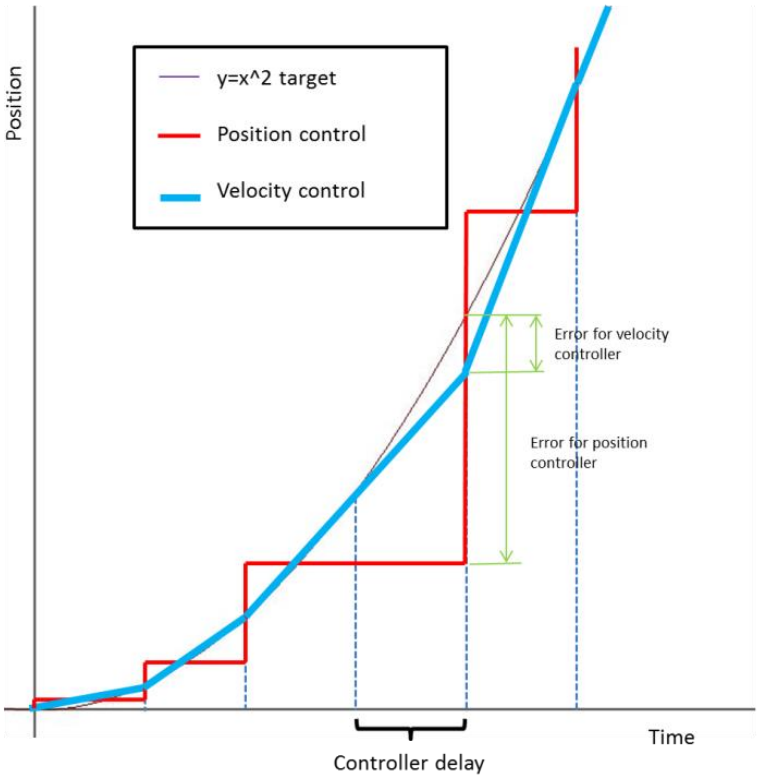

Figure 6-5. Robustness of Velocity Control

How much of the robustness is due to the fuzzy logic controller? Would another type of controller also be so robust, given the advantages of velocity control that were mentioned above? These questions remain for future work.

6.7 CONTROL RATE AT FAILURE EXPERIMENT

We are interested in predicting the necessary control rate for a given manipulation task. To make the prediction, we designed an experiment where the maximum acceleration of the trajectory along the x-axis was set to four different values as shown in Table 6-2. For each trajectory, we began motion at full speed and ran the robot for 10s. If it was successful, we lowered the control rate in one-Hz intervals until reaching the minimum acceptable control rate. There were two parts to our definition of acceptable:

- The robots maintained contact with the object (i.e. the compressive force remained greater than zero).
- The compressive force did not exceed 44.48N (ten pounds).

As we increased the velocity, the frequency was increased proportionally. This was done in order to keep the motion within the workspace of the robots.

Table 6-2. Trajectory, Acceleration, and Minimum Control Rate

Trajectory (mm/s)	Maximum Acceleration (mm/s^2)	Minimum Control Rate (Hz)
$6.25\sin(0.05\pi t)$	.3125	4
$12.5\sin(0.1\pi t)$	1.25	4
$25\sin(0.2\pi t)$	5	4
$50\sin(0.4\pi t)$	20	4

Our hypothesis was that a higher acceleration and a larger maximum velocity would require a higher control rate. However, the data surprised us. As the reader can see, despite a sixty-four fold increase in acceleration and an eight-fold increase in velocity,

the minimum control frequency did not vary from four Hz. At this time, we cannot explain this phenomenon. A test with a more narrowly-defined metric of success might be necessary to distinguish between the control rates.

CHAPTER 7 : CONCLUSIONS AND FUTURE WORK

7.1 RESEARCH SUMMARY

The control law as presented in this report advances the state-of-the-art for industrial, dual-arm manipulators. Unlike the previous work that was described in Chapter Two, this control law is suitable for common industrial manipulators because it behaves as an admittance. It allows for very fine control of internal forces and moments on the object (±5 N and ±17 N*cm, respectively). We have also described how programming the controller in velocity space provides a great amount of robustness with regard to control signal delay.

7.2 OVERVIEW OF OBJECTIVES

In Chapter Two, Table 2-1 benchmarked six controllers from the literature against twelve characteristics of an ideal dual-arm controller. After our own controller was programmed and tested, we use the same twelve characteristics to compare it against the field (Table 7-1).

Table 7-1. Comparison with controllers from the literature

	Arimoto, 1987- Master/slave	Hayati, 1986- Hybrid control	Yoshikawa, 1993- Hybrid control	Song, 2002- Artificial Potential Fields	Bonitz, 1996- Impedance	Caccavale, 2008- Impedance	Zelenak, 2013- Admittance
Nonlinear	✓	✓	✓	✓	✓	✓	✓
Compliant				✓		✓	
Internal forces are limited		✓	✓	?	✓	✓	✓
Controller is robust	✓	?	?	✓	?	?	✓
Formulated as an admittance							✓
Knowledge of object's inertia is unnecessary				✓	✓		✓
Force is distributed between arms		✓					
Formulated in velocity space	✓						✓
Indistinguishable operation		✓	✓	✓	✓	✓	✓
Low computational cost	✓			✓			✓
Proven global stability	✓	✓	✓	✓	✓	✓	✓
Generalizes to redundant manipulators	✓	✓	✓				✓

Our own controller does quite well against these twelve characteristics, as it satisfies ten. Its two faults, by this metric, are that it doesn't explicitly divide the load between arms; and it has no compliance with the environment. Load division is only necessary when dealing with heavy payloads; otherwise it is strictly a bonus. In "Recommendations for future work", we explain that it would be straight-forward to add an external compliance. In fact, the NRG has already programmed an external compliance controller and it just remains to be integrated. The control law of this thesis does a few things that most of the prior art did not; namely it behaves as an admittance (so it is suitable for industrial robots), and it is formulated in velocity space (which, as we showed in Chapter Six, makes for smoother, more robust motion).

7.3 RECOMMENDATIONS FOR FUTURE WORK

Adaptive control

In this report, the gains for the fuzzy logic were based on operator intuition. We did not make any attempt to optimize the gains, but there seems to be some potential for improvement (as described in "Force tracking experiment" in Chapter Six). Some type of adaptive fuzzy logic control (or Adaptive Neuro Fuzzy Inference System, as it is called in the literature) would extract the last bit of improvement from the system. There are also adaptive schemes where the developer does not provide any input; the system could "learn" the fuzzy control law completely from scratch.

Position feedback

Chapter Six also mentions the subpar position tracking because it has no position feedback. It is possible to implement position feedback since this internal admittance controller takes a target velocity, v_{target}, as a reference. There would be no code modifications to the internal admittance controller; the position feedback would be implemented on v_{target} before it gets passed into this controller. For example, one could implement an artificial potential field type of system, where v_{target} is continuously adjusted to always point towards the target location.

External compliance

Chapter Four described the development of the fuzzy control law and how the forces on the object could be split into net force and internal force. We then described how the internal force could be used to calculate an internal compliance. But the net force was never used; future work should develop an "external compliance" that is based on this net force. This external compliance would govern the robots' interactions with its environment, as described in Hogan, [1984]. For operation in a dynamic environment, that external compliance is essential. The process of applying this external compliance would be straightforward: the net force on the object is the sum of the forces from the two arms, and a fuzzy logic controller (which would be very similar to the internal force controller as presented here) would calculate a velocity adjustment from the net force. Figure 7-1 shows how external compliance would fit into the block diagram.

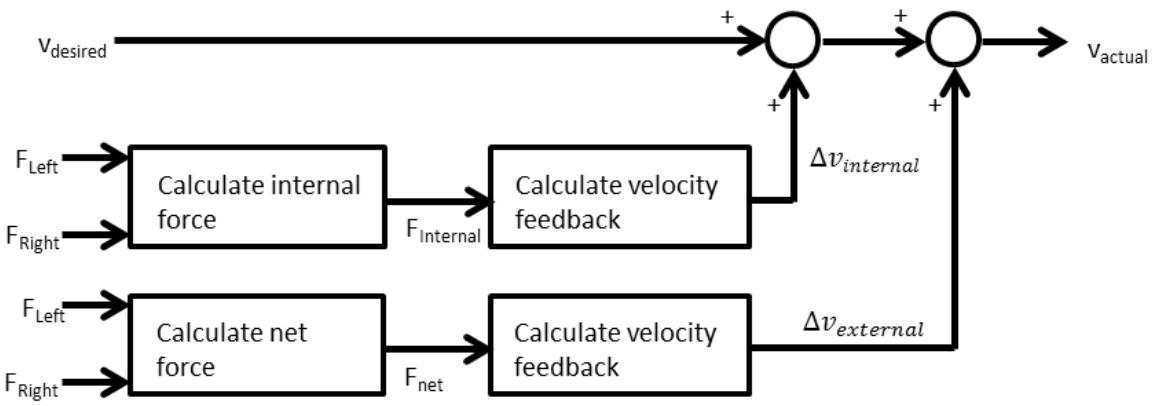

Figure 7-1. Block diagram for addition of external compliance

Change to Lever Arm measurement

In Chapter Four we described how the length of the object must be accurately measured when it is clamped between the arms. This length measurement, or "lever arm", is crucial for proper rotation of the arms. A bit of error here will eventually lead the arms to diverge. So we would like to implement:

- A highly accurate method to measure this distance, or
- A method to adjust this measurement on the fly.

A highly accurate method might consist of moving the robots together before the object is in place and using forward kinematics to calculate the distance between robots. Then, when the object is clamped, calculate the forward kinematics to determine the length of the object with a much higher degree of accuracy.

It is possible to compare the orientations of the two robot arms, in world coordinates, as the object is manipulated. The orientations should always be equal and opposite; if they are not, feedback could be used to adjust the orientation of the arms, thus avoiding this "lever arm" measurement completely.

Predict minimum control rate

An experiment in Chapter Six showed (surprisingly) that this controller is quite robust against variable control rates, and several hypotheses to explain this were proposed. Can we use our understanding of this phenomenon to predict the minimum acceptable control rate for a task beforehand? Such an endeavor would, most likely, require measurements of the object's compliance and the robot's inherent compliance, and it would depend on the speed of the task. If successful, the results would open new doors to robot control with low-power computers and relax the requirements for computationally-efficient programming.

Sensor fusion

We can draw some parallels between our controller and the human nervous system. Like our controller, humans do not have great position tracking with their motions. Like our controller, the human body does not need an extremely fast control rate (a human's reaction time is about 215 ms) ["Reaction Time Statistics," 2013]. Like our controller, the human body uses velocity control to accomplish its motion [Cusumano, 2009]. And we would like to modify the external compliance calculation to draw one more parallel with the human body: *Humans depend heavily on constant sensory input to adjust their control method*.

Imagine bending over to pick up a box. If the box is very heavy, we will adjust our stance and our effort to compensate for that extra weight. For a robot, that would be analogous to updating the inertial model of the box.

Now imagine that same box is wedged under a cabinet. If a human tries to lift it and gets stopped by the cabinet, we can say, "I can't move because the cabinet blocked that box." Whereas a robot may not consider the cabinet; when it tries to lift the box, it

will think, “Wow, this box is really heavy since it is not moving. I am going to increase the model mass to a very large number.” Eventually the robot will break something or reach saturation of its actuators.

We would like to integrate sensor data so the robot can make the distinction. Is a feature of the external environment interacting with the motion of this box?

- Yes ➔ the robot should behave as an external admittance so it doesn’t damage anything.
- No ➔ start updating the inertial model of the box, so we can have the most accurate motion possible. No external admittance needed.

Of course, even this rudimentary sensor fusion doesn’t completely match the abilities of a human. Humans are able to cope with situations where they are deprived of one sensory input or another. For example, humans can navigate across a dark room while remembering where the coffee table is. That is yet another level of sophistication.

7.4 CONCLUDING REMARKS

This report goes a long way towards making cooperative manipulation with industrial robots more practical:

- Internal forces in the object are controlled.
- The controller is robust and global stability is proven.
- Since the controller is formulated as an admittance, it is applicable to industrial robots.
- The controller is model-free.
- The computational cost of the controller is low compared to the prior art.

However, it seems that this study has created more questions than it answered. Is it the fuzzy logic or the velocity control that makes this controller so robust? Can we predict the minimum control rate for a task beforehand? How much better would the system perform if we were adapting the fuzzy control laws? The list goes on, but as Frank Herbert concluded, “The beginning of knowledge is the discovery of something we do not understand.”

Appendix

SAMPLE MATLAB CODE FOR DIGITAL SAMPLING

This is the sample MATLAB code for digital sampling, as explained in Chapter Three. It samples a digital signal and applies a Butterworth Low Pass Filter. The first step is to specify the coefficients of the Butterworth filter. MATLAB has a command to do this automatically:

```
[numerator_coeff, denom_coeff] = butter(2, cutoff_freq*2*pi, 's'); %Generate the filter coefficients for a second-order Butterworth LPF

Filter_TF = tf(numerator_coeff, denom_coeff); %Create a variable that MATLAB recognizes as a transfer function

Laplace_TF = tf( [ 1 0], 1 );      %Multiply by s to take the derivative
```

Then the Tustin transform is applied, to transform back to the digital domain:

```
Discrete_TF = c2d(Laplace_TF, sample_time, 'tustin');
```

Recover the coefficients of the discrete time transfer function:

```
[num, den] = tfdata( Discrete_TF);
num=cell2mat(num);
den = cell2mat(den);
```

Finally, get the coefficients *a* through *d*:

```
a = num(1);      b = num(2);      c= num(3);       d = -den(1);      e = -den(2);
```

References

Almutairi, Naif B., Mo-Yuen Chow, and Y. Tipsuwan. "Network-based Controlled DC Motor with Fuzzy Compensation." *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE* 3 (2001): 1844-49. Print.

Arimoto, Suguru, Fumio Miyazaki, and Sadao Kawamura. "Cooperative Motion Control of Multiple Robot Arms or Fingers." *Proceedings of the 1987 IEEE Conference on Robotics and Automation* 4: 1407-12. Print.

AX-I3 Solution. Agile Planet, n.d. Web. 17 Apr. 2013. <http://www.agileplanet.com/products/ax-i3>.

Bonitz, R. C. "Internal Force-Based Impedance Control for Cooperating Manipulators." *IEEE Transactions on Robotics and Automation* 12.1 (1996): 78-89. Print.

Braun, B. M. "A Framework for Implementing Cooperative Motion on Industrial Controllers." *IEEE Transactions on Robotics and Automotion* 20.3 (2004): 583-89. Print.

Caccavale, F., et al. "Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators." *IEEE/ASME Transactions on Mechatronics* 13.5 (2008): 576-86. Print.

CEWIN - The Real-time Extension for Windows XP Embedded - Accelerates the Pace of Progress. KUKA, n.d. Web. 17 Apr. 2013. <http://www.kuka-robotics.com/india/en/pressevents/news/NN_040317_CeWin.htm>.

Choi, Byung-Jae, Seong-Woo Kwak, and Byung Kook Kim. "Design and Stability Analysis of Single-Input Fuzzy Logic Controller." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 30.2 (2000): 303-09. Print.

DARPA Robotics Challange Track A - Teams and Designs. DARPA, n.d. Web. 12 Apr. 2013. <http://www.darpa.mil/Our_Work/TTO/Programs/DARPA_Robotics_Challenge/Track_A_Participants.aspx>.

Diolaiti, N. "Contact Impedance Estimation for Robotic Systems." *IEEE Transactions on Robotics* 21.5 (2005): 925-35. Print.

Duchaine, Vincent, and Clement M. Gosselin. "General Model of Human-Robot Cooperation using a Novel Velocity Based Variable Impedance Control." *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint* (2007): 446-51. Print.

Ferretti, Gianni, et al. "Impedance Control for Industrial Robots." *IEEE International Conference On Robotics and Automation* 4 (2000): 4027-32. Print.

F/T Sensor: Gamma. ATI Industrial Automation, n.d. Web. 17 Apr. 2013. <http://www.ati-ia.com/products/ft/ft_models.aspx?id=Gamma>.

Global 2000 Leading Companies. Forbes, Apr. 2012. Web. 15 Apr. 2013. <http://www.forbes.com/companies/abb/>.

Harper, Christopher, and Gurvinder Virk. "Towards the Development of International Safety Standards for Human Robot Interaction." *International Journal of Social Robotics* 2.3 (2010): 229-34. Print.

Hayati, Samad. "Hybrid Position/Force Control of Multi-Arm Cooperating Robots." *Proceedings of the 1986 IEEE International Conference on Robotics and Automation* 3 (1986): 82-89. Print.

Hogan, Neville. "Impedance Control: An Approach to Maniuplation: Part I-Theory." *Journal of Dynamic Systems, Measurements, and Control* 107.1 (1985): 1-7. Print.

Horiuchi, Jun-Ichi, and Kohki Hiraga. "Industrial Application of Fuzzy Control to Large-Scale Recombinnant Vitamin B2 Production." *Journal of Bioscience and Bioengineering* 87.3 (1999): 365-71. Print.

How Rethink Robotics Built Its New Baxter Robot Worker. IEEE Spectrum, Oct. 2012. Web. 15 Apr. 2013. <http://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker>.

Ishikawa, Shigeki. "A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control." *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on* (1991): 1013-18. Print.

Juang, Chia-Feng, and Chia-Hung Hsu. "Reinforcement and Optimized Fuzzy Controller for Mobile-Robot Walll-Following Control." *Industrial Electronics, IEEE Transactions on* 56.10 (2009): 3931-40. Print.

Koren, Yoram, and Johann Borenstein. "Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation." *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (1991): 1398-404. Print.

Kosuge, Kazuhiro, Tomohiro Oosumi, and Kunihiro Chiba. "Load Sharing of Decentralized-Controlled Multiple Mobile Robots Handling a Single Object." *Proceedings of the 1997 IEEE International Conference on Robotics and Automation* 4: 3373-78. Print.

Leith, Douglas J., and William E. Leithead. "Survey of Gain-Scheduling Analysis and Design." *International Journal of Control* 73.11 (2000): 1001-25. Print.

O'Neal, Brian, and Mitch Pryor. "Cylindrical Projection Histograms for 3D Object Recognition and Pose Estimation." 1 Jan. 2013. MS.

Oshima, Hiroyasu, Seiji Yasunobu, and S. I. Sekino. "Automatic Train Operation System Based on Predictive Fuzzy Control." *In Artificial Intelligence for Industrial Applications, 1988. IEEE AI'88., Proceedings of the International Workshop on* (1988): 485-89. Print.

P, Cusumano Joseph, John Joby, and Jonathan B. Dingwell. "Evidence for Goal Equivalent Control in Treadmill Walking." *Proceedings of the NACOB* (2009): n. pag. Print.

Paarmann, Larry. *Design and Analysis of Analog Filters: A Signal Processing Perspective*. Illustrated ed. N.p.: Springer, 2001. Print.

Paine, Nicholas. "Discrete Differentiation." Longhorn Robotics Group. Engineering Teaching Center, Austin. 2 May 2012. Speech.

Reaction Time Statistics. HumanBenchmark.com, n.d. Web. 17 Apr. 2013. <http://www.humanbenchmark.com/tests/reactiontime/stats.php>.

Reis, Ed. "Westinghouse Robots Were Fascinating Characters." *Engineers' Society of Western Pennsylvania*. Engineers' Society of Western Pennsylvania, n.d. Web. 12 Apr. 2013. <http://www.eswp.com/PDF/PEspring05_12-13.pdf>.

"Robot Compliance in One Cartesian Direction." *SoftMove*. ABB, n.d. Web. 15 Apr. 2013. <http://www.abb.com/product/seitp327/bd86a9974df2ce0cc12573ed00546ca1.aspx>.

Schneider, Stanley A. "Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results." *IEEE Transactions on Robotics and Automation* 8.3 (1992): 383-94. Print.

Schroeder, Kyle, Mitch Pryor, and Troy Harden. *On the Use of Joint Torque Sensors for Collision Detection in a Confined Environment*. Research rept. no. LA-UR-11-02804. N.p.: n.p., 2011. *Los Alamos National Laboratory*. Web. 28 Apr. 2013. <http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-02804>.

"SIA5D Datasheet." *Solutions in Motion*. Yaskawa Motoman Robotics, Mar. 2012. Web. 17 Apr. 2013. <http://www.motoman.com/datasheets/SIA5D.pdf>.

Song, Peng, and Vijay Kumar. "A Potential Field Based Approach to Multi-Robot Manipulation." *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conf. on* 2 (2002): 1217-22. Print.

Squid Robot: Gumby-Like Robot Squeezes Through Tight Spaces. The Christian Science Monitor, 29 Nov. 2011. Web. 15 Apr. 2013. <http://www.csmonitor.com/Science/2011/1129/Squid-robot-Gumby-like-robot-squeezes-through-tight-spaces-VIDEO>.

Tactical Technology Office. DARPA, n.d. Web. 12 Apr. 2013. <http://www.darpa.mil/Our_Work/TTO/Programs/DARPA_Robotics_Challenge.aspx>.

The 10 Tech Terms to Know in 2013. Popular Mechanics, n.d. Web. 15 Apr. 2013. <http://www.popularmechanics.com/technology/engineering/news/the-10-tech-terms-to-know-in-2013?click=main_sr#slide-3>.

Tipsuwan, Yodyium, and Mo_yuen Chow. "Control Methodologies in Networked Control Systems." *Control Engineering Practice* 11.10 (2003): 1099-111. Print.

Torque Guide. SKS Bottles, n.d. Web. 17 Apr. 2013. <http://www.sks-bottle.com/TorqueHelp.html>.

Wang, L. X. "Stable Adaptive Fuzzy Control of Nonlinear Systems." *Fuzzy Systems, IEEE Transactions on* 1.2 (1993): 146-55. Print.

Williams, V. Vassilevska. "Breaking the Coppersmith-Winograd Barrier." Nov. 2011. MS.

Wimbock, Thomas, and Christian Ott. *Towards Service Robots for Everyday Environments*. Berlin Heidelberg: Springer, 2012. Print.
Yasunobu, Seiji, and Shoji Miyamoto. "Automatic Train Operation System by Predictive Fuzzy Control." *Industrial Applications of Fuzzy Control* (1985): 1-18. Print.
Yoshikawa, Tsuneo, and Xin-Zhi Zheng. "Coordinated Dynamic Hybrid Position/Force Control for Multiple Robot Manipulators Handling One Constrained Object." *The International Journal of Robotics Research* 12.3 (1993): 219-30. Print.
Zelenak, Andy, et al. "Intelligent Grasping with the Robotic Opposable Thumb." 1 Jan. 2013. TS.