Spring 2013

Admittance control of the intelligent assist robot manipulator for people with duchenne muscular dystrophy

Madeline Corrigan
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

ADMITTANCE CONTROL OF THE INTELLIGENT ASSIST ROBOT MANIPULATOR FOR PEOPLE WITH DUCHENNE MUSCULAR DYSTROPHY

by
Madeline Corrigan

Duchenne muscular dystrophy (DMD), a neuromuscular disease with a prevalence of 1 in 3500 male births, results in characteristic muscle weakness which is progressive with age and leads to loss of independence. And, in this population, maintaining optimal quality of life depends on the preservation of self-sufficiency. Despite the loss of function, non-ambulant people with DMD retain some muscle strength, just not sufficient strength to overcome the force of gravity. There are a number of upper-limb passive and active orthotic devices that attempt to augment the loss of upper limb function in people with DMD by taking advantage of this residual muscle strength by providing anti-gravity assistance. The majority of these devices, as well as currently available robotic manipulators, are considerably limited in the functionality that they provide, rendering them obtrusive and unaccommodating, resulting in lack of use by this population.

This thesis presents the design of a novel upper limb assistive robotic device. This design involves the use of admittance control as the interface for the intelligent Assist Robot Manipulator (iARM). A thorough qualitative and quantitative analysis of the prototype is performed, the results of which are presented. The quantitative analysis focuses on the ideal delay that is required of human-machine interfaces to ensure comfort and passivity. Additionally, potential contributors to the delay of the iARM are investigated.

ADMITTANCE CONTROL OF THE INTELLIGENT ASSIST ROBOT MANIPULATOR FOR PEOPLE WITH DUCHENNE MUSCULAR DYSTROPHY

by
Madeline Corrigan

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Biomedical Engineering

Department of Biomedical Engineering

May 2013

APPROVAL PAGE

ADMITTANCE CONTROL OF THE INTELLIGENT ASSIST ROBOT MANIPULATOR FOR PEOPLE WITH DUCHENNE MUSCULAR DYSTROPHY

Madeline Corrigan

Dr. Richard Foulds, Thesis Advisor
Associate Professor of Biomedical Engineering, NJIT

Dr. Sergei Adamovich, Committee Member
Associate Professor of Biomedical Engineering, NJIT

Dr. Max Roman, Committee Member
Assistant Research Professor of Biomedical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Madeline Corrigan

Degree: Master of Science

Date: May 2013

Undergraduate and Graduate Education:

- Master of Science in Biomedical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2013

- Bachelor of Science in Biomedical Engineering,
  University of Minnesota Institute of Technology, Minneapolis, MN, 2010

Major: Biomedical Engineering

Presentations and Publication:

For my MDA family.

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my research advisor, Dr. Richard Foulds, who provided me with invaluable insight and support that was instrumental to this project and to my graduate educational experience at New Jersey Institute of Technology. I would also like to extend a special thank you to Dr. Sergei Adamovich and Dr. Max Roman for participating in my thesis committee. I would also like to thank John Hoinowski for sharing his expertise.

Thank you to my fellow graduate students, Kiran Karunakaran and Kevin Abbruzzese, for their support and help.

I would like to acknowledge my family. Their unconditional love and encouragement have been absolutely vital to my success.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Motivation

Duchenne Muscular Dystrophy (DMD) is a muscle wasting disease whose progression robs a person's ability to live their life independently. It imposes the demands of costly and intrusive assistive support and personal care for the most modest of daily tasks. Self-sufficiency is the foundation to the preservation of self-esteem and quality of life for all persons. For the population suffering from DMD, upper limb function is key to obtaining and preserving these vital human elements. Current upper limb powered and passive orthotic devices and robotic manipulators are significantly limited in their ability to suitably restore functionality and improve quality of life for those with DMD.

1.2 Specific Aims

This thesis will review current devices that aim to augment upper limb functionality in people with DMD as well as assess their advantages and evaluate their limitations. Subsequently, a design for a novel upper limb assistive robotic device that attempts to overcome these limitations will be presented. A prototype of the device will be evaluated qualitatively and quantitatively for its effectiveness in accomplishing specific design requirements. The quantitative analysis will focus on the delay of the device in regard to the ideal delay required of human-machine interfaces.

CHAPTER 2

BACKGROUND

2.1 Duchenne Muscular Dystrophy (DMD)

2.1.1 Overview of DMD

Muscular Dystrophy is a group of devastating disorders that greatly limit a person's ability to use their muscles. Duchenne Muscular Dystrophy (DMD), the most common type of Muscular Dystrophy, is an X-linked recessive neuromuscular disorder with an incidence of 1 in 3,500 male births. It is characterized by the lack of the protein dystrophin, which provides structural stability to the dystrophin-associated protein complex in the cell membrane of muscle cells [1,2]. The absence of dystrophin in the muscle cell membrane results in the five mechanisms of DMD pathophysiology, which are: the mechanical weakening of the sarcolemma (the cell membrane of a muscle cell), inappropriate Ca2+ influx (which is involved in skeletal muscle contraction), aberrant cell signaling, increased oxidative stress, and recurrent muscle ischemia (restricted blood supply) [3]. These mechanisms directly result in the progressive weakening of skeletal, respiratory, and cardiac muscles causing decreased independence and shortened life expectancy [2].

2.1.2 Progression of DMD

The onset of muscle weakness in children with DMD typically occurs before 5 years of age, the time at which diagnosis typically occurs. This is soon followed by gait difficulty and eventually loss of ambulation which occurs on average by the age of 15 years. At this time, patients are fully dependent on a wheelchair. Chronic respiratory failure

develops in the advanced stages of the disease, and death most often occurs around the age of 20 [4, 1].

The administration of mechanical ventilation is commonplace when patients begin to exhibit respiratory failure. Its application is responsible for increasing the median survival rate for people with DMD to more than 25 years with some studies reporting an increase of up to 30 years of age. Other treatments, such as steroids for muscle and cardiac function, slow the progression and increase the life expectancy of people with DMD. This increase in life expectancy results in a significant portion of the population of people with DMD living with a strong dependency on personal and technical care and support. Despite the advancements in steroid treatments and assisted mechanical ventilation, there remains no cure for DMD [4, 1, 2].

Upper limb functional assessment studies show progressive upper limb weakness in people with DMD. The onset of upper limb weakness occurs during ambulation and gradually increases with time in a proximal to distal gradient. Typically, the decline in muscle strength is greatest at younger ages, with loss per year decreasing with age. There has been limited functional assessment research on older individuals with DMD that are non-ambulant and a small number of studies with modest sample sizes that investigate and aim to quantify the progression of muscle strength and functional ability in people with DMD [5, 6].

Steffensen et al. performed a study that aimed to assess the physical capacity in non-ambulatory people with DMD that included 19 individuals. The study used the EK scale and MRC% as assessment tools. The EK scale, a valid measure of functional ability in people with DMD which provides a close correlation to muscle strength, is an

assessment tool that includes 10 categories scored from 0 to 3. The EK sum is a numerical score computed from the sum of the scores of all 10 categories, with 0 representing the highest degree of independence and 30 representing the lowest degree of independence. The MRC%, a measure of maximal muscle strength, is the percentage of normal maximal obtainable muscle contraction. It was computed in this study from the assessment of forty muscle groups around the neck, shoulders, elbows, wrists, fingers, trunk, hips, knees and ankles. Figure 2.1 shows the data from the results of the study. Functional ability significantly decreased with age, which is reflected by an increase in EK sum with age. Similarly, muscle strength, reflected by MRC%, significantly deteriorated with age. Linear regression analysis determined the mean annual change in EK sum score to be 1.3 and the mean annual change in MRC% to be -2.0. Furthermore, this study found that the median age for loss of ambulation in the subjects was 10 years of age, with a range of 8 to 12 years of age [6].

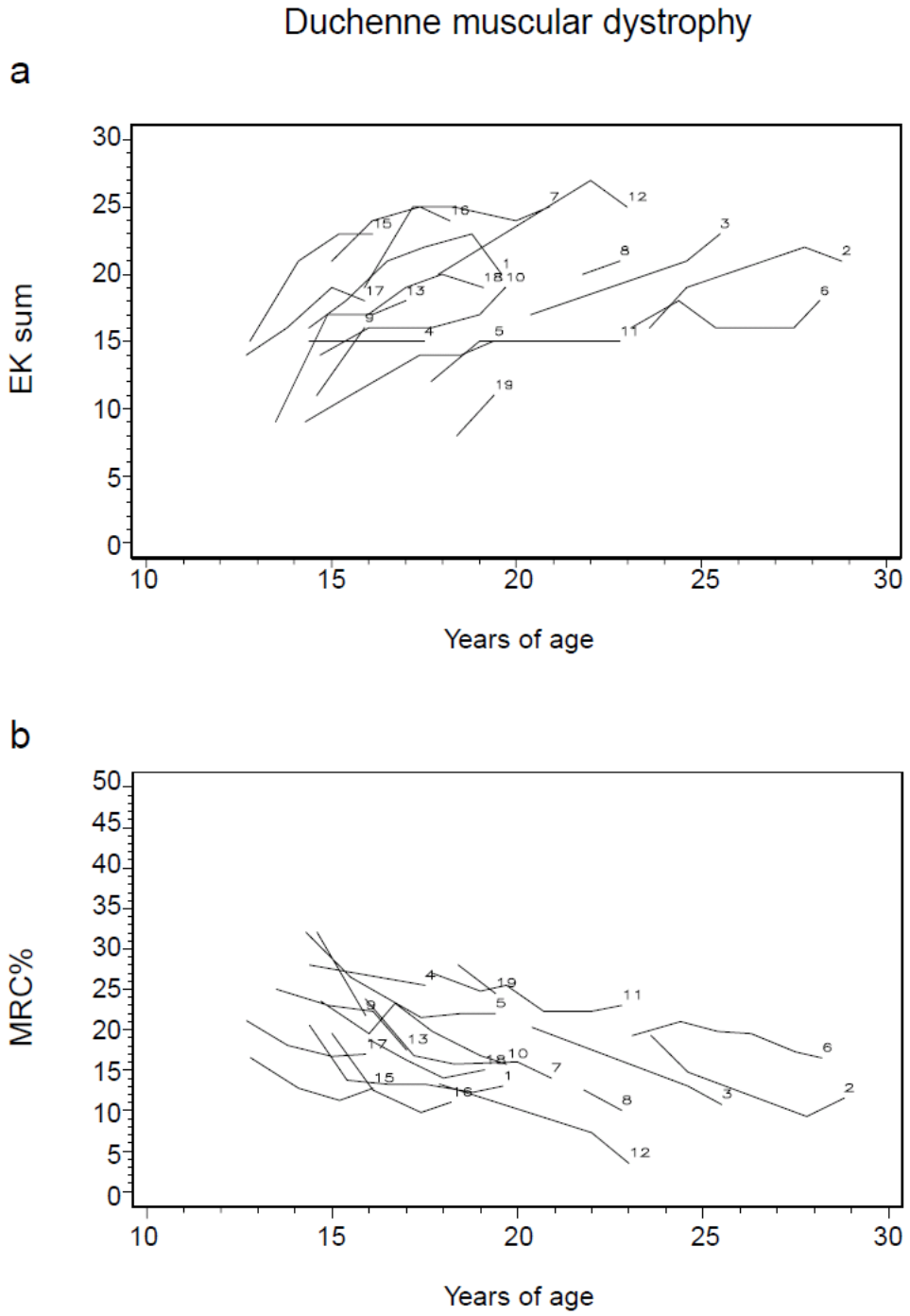

Figure 2.1 Data from individual assessments of variables (a) EK sum and (b) MRC% in individuals with DMD (n=19) in relation to their ages. Participants presented by case number. Participants tested only once marked with asterisk (participant 14).
Source: [6]

Kohler et al. investigated the change in physical disability with age in 29 patients with DMD. The subject's degree of physical disability was scored from 0 to 80, with a higher score reflecting greater physical disability based on the subjects' ability or inability to perform activities of daily living and their dependence on technical aids and personal care aids. The total physical disability score was the sum of scores rated from 0 to 10 in eight separate aspects of daily living: mobility without technical aids, mobility with technical aids, transfers, static body control, changes of body position, dressing, feeding, and breathing. The data was collected during at least two assessment sessions at yearly intervals. Figure 2.2 shows the results of the assessment for each of the 29 people with DMD. The results clearly demonstrate the progressive limitation in activities of daily living and the dependence on others and technical aids [4].

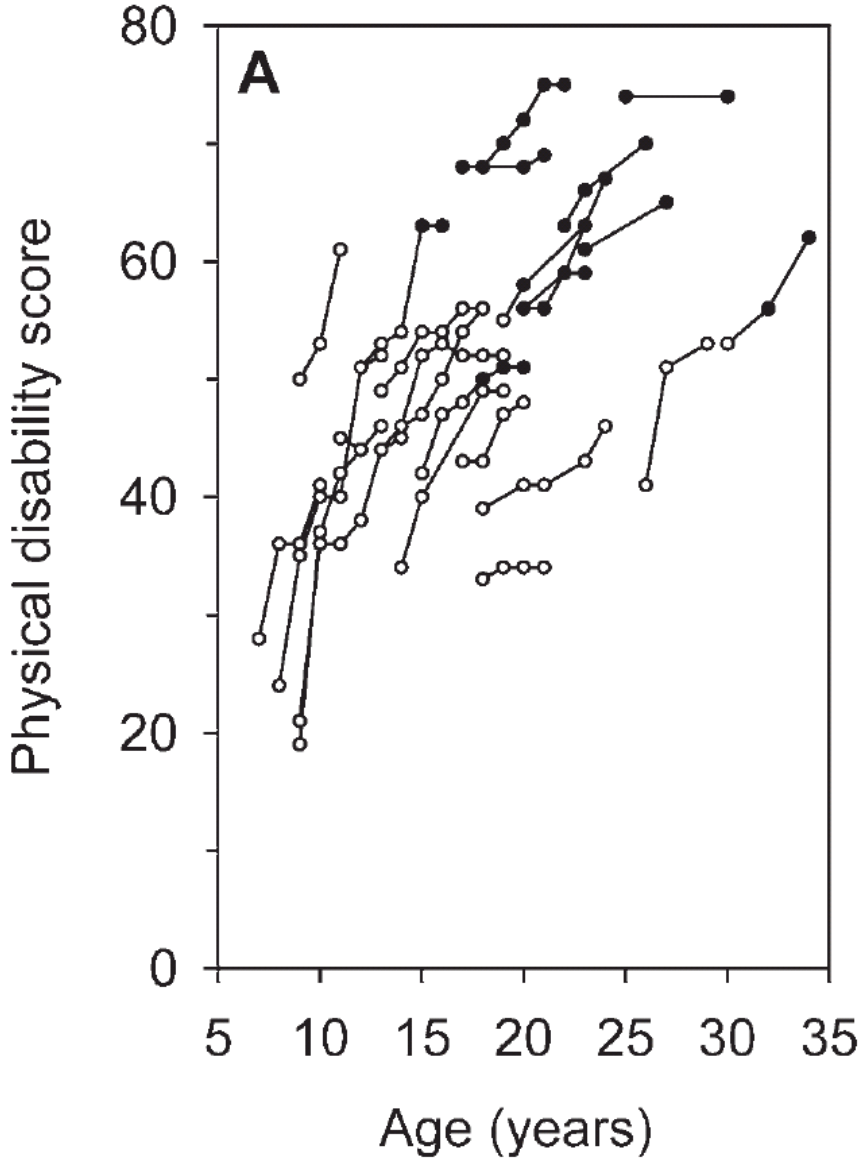

Figure 2.2 Disability scores of 29 patients, with at least two observations obtained at yearly intervals before the main data acquisition, clearly illustrate the progressive limitation in activities of daily living, and the dependence on physical and technical support. Lines connect data in individual patients without noninvasive ventilation (NIPPV; open circles) and with NIPPV (closed circles).
Source: [4]

A study by Jung et al. aimed to specifically look at the progression of upper extremity functional loss in people with DMD. The study involved the evaluation of upper extremity function using the Brooke scale. The Brooke scale consists of ratings between 1 and 6, with a 1 reflecting full upper extremity functionality and a 6 representing no useful function of the upper extremities. Table 2.1 shows the grading for the Brooke scale. The functional assessment was performed on 90 subjects with DMD in up to three separate sessions. Figure 2.3 shows a plot of the Brooke scale scores from each assessment plotted versus age. As demonstrated by the results of this study, the Brooke scale score was shown to increase linearly with age, ($p<0.001$). This demonstrates a linear decrease in upper extremity function in people with DMD. However, it is important to note that some studies have stated that muscle strength increases in the early stages of DMD and decreases as DMD progresses. Furthermore, some qualify the progressive loss as non-linear, worsening sharply in proportion to age in only some stages of life [1].

Table 2.1 Grading for the Brooke Scale

Grade	Functional description
	Brooke scale for upper extremity
1	Starting with arms at the sides, the patient can abduct the arms in a full circle until they touch above the head
2	Can raise arms above head only by flexing the elbow (shortening the circumference of the movement) or using accessory muscles
3	Cannot raise hands above head, but can raise an 8-oz glass of water to the mouth
4	Can raise hands to the mouth, but cannot raise an 8-oz glass of water to the mouth
5	Cannot raise hands to the mouth, but can use hands to hold a pen or pick up pennies from the table
6	Cannot raise hands to the mouth and has no useful function of hands

Source: [1]

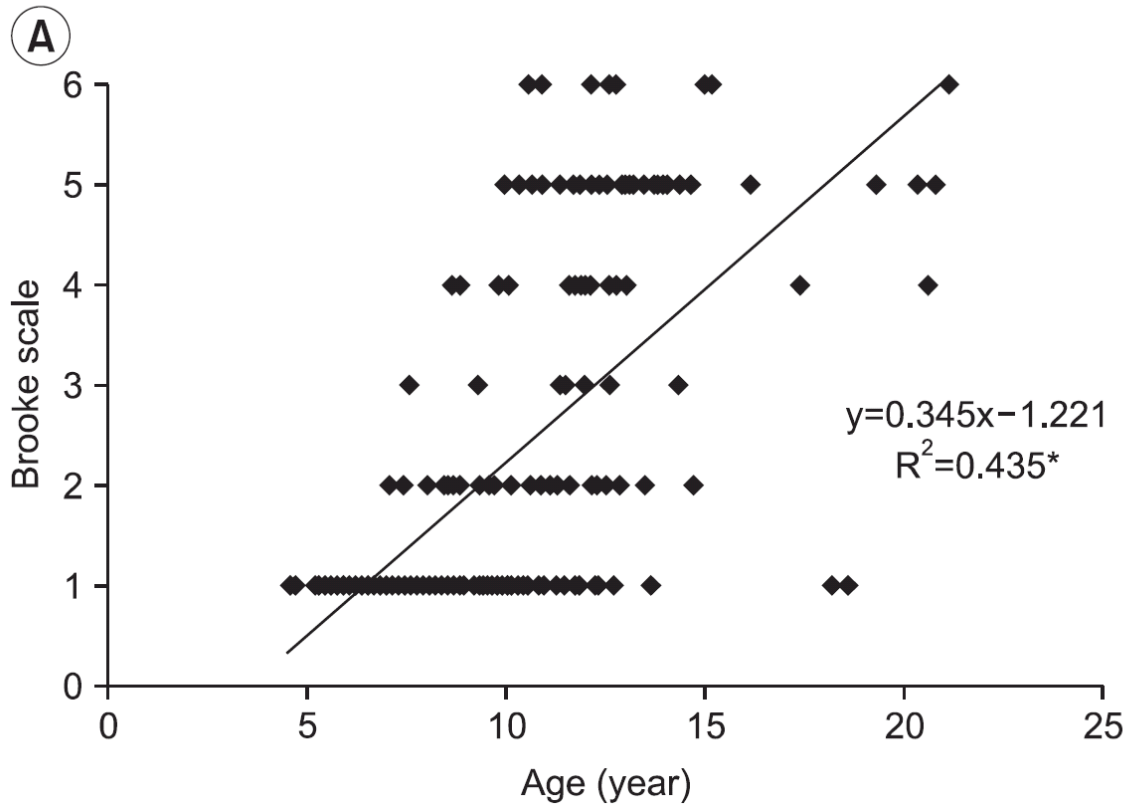

Figure 2.3 The Brooke scale of Duchenne muscular dystrophy patients with age.
Source: [1]

Bartels et al. investigated the distal motor function of the upper limbs in adults with DMD. The study involved the use of the Motor Function Measure (MFM) a Medical Research Council scale for manual muscle testing that provides a standardized assessment of motor capacity in a person with a neuromuscular disorder. The MFM score, which has been proven to be a valid measure of motor function for people with DMD, was used to assess the strength of 8 muscle groups: flexion and abduction of the shoulder, flexion and extension of the elbow, flexion and extension of the wrist, adduction of the thumb, and flexion of the index finger. The MFM score, which is given as a percentage of maximum possible score, is higher when there is less impairment in

muscle strength and lower when there is more impairment in muscle strength. The assessment was performed on 70 subjects with DMD ages 20-43. Figure 2.4 shows the MFM D3 upper limb score, reflecting upper limb motor function, plotted against age. The results of the study show that all of the subjects had severely impaired motor function as well as a clear decrease in distal motor function of the upper limb with age. Furthermore, none of the subjects were able to lift their entire arm against gravity. However, 37% of subjects could still lift their hand against gravity and 7% of subjects could still raise their lower arm against gravity. Figure 2.5 shows the individual item scores of the MFM D3 upper limb assessment scores. This study emphasized the importance of maintaining upper limb muscle strength and range of motion in adults with DMD because of their correlation with upper limb function. The study further notes that upper limb function deserves attention in rehabilitation and research [2].

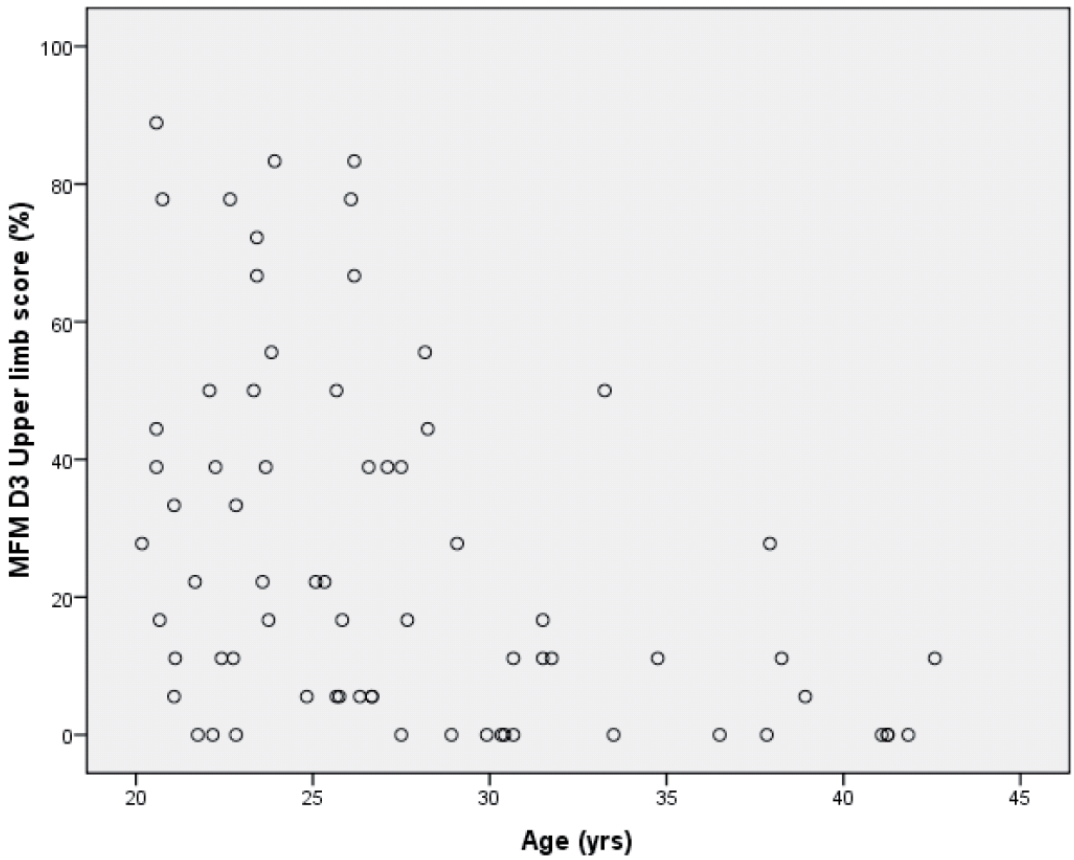

Figure 2.4 Age and distal motor function of the upper limb (MFM D3 upper limb score) as a percentage of the maximal score.
Source: [2]

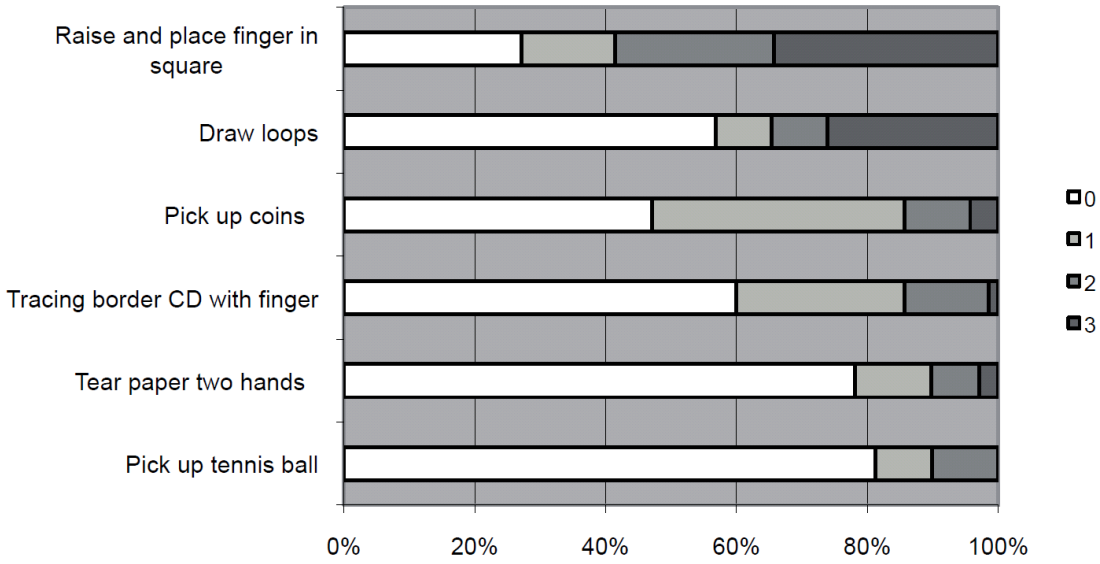

Figure 2.5 Item scores MFM D3 upper limb score. Percentage of participants shown. 0: cannot initiate task; 1: partially performs task; 2: performs task incompletely or completely but imperfectly; 3: performs task fully and normally.
Source: [2]

2.1.3 Antigravity Strength

The progressive loss of upper extremity function is accelerated by the decreased antigravity strength experienced in people with DMD. When the progressive loss of muscle strength involves the inability to overcome the force of gravity, there is an immediate and significant decrease in the active range of motion despite any remaining muscle strength in the upper extremities. As the muscles begin to lose their ability to overcome the force of gravity in a proximal to distal fashion, movements are limited to the hand and wrist and eventually limited to only the fingers. Again, this loss in movement or active range of motion is limited despite the fact that there is remaining

muscle strength in the upper extremities. This remaining muscle strength is simply not sufficient to overcome gravity [3, 5].

These observations are demonstrated through the use of aquatic therapy for people with DMD. As a human body is immersed in water, water is displaced which results in the force of buoyancy. This force of buoyancy is opposite the force of gravity and therefore decreases joint loading forces on the joints that are immersed in water. It is a widely accepted qualitative assessment that people with DMD have an increased degree of freedom of movement when they are provided with antigravity assistance in the form of buoyancy. Furthermore, multiple studies have reported that the buoyancy of water enables independent initiation of movements for people with DMD that are less likely when the person is on land [7, 8, 9, 10].

2.1.4 Disuse Atrophy and Contractures

Physical therapy for people with DMD typically involves exercise therapy. Exercise therapy is used to preserve muscle function by preventing disuse atrophy, which is the secondary deterioration of muscle strength that results from a person's actual performance despite a greater potential capacity. The physiological advantages of preventing disuse atrophy through the performance of submaximal exercise can be characterized by the potential positive effects of exercise on the five mechanisms of DMD pathophysiology: providing a membrane protective effect, improving Ca2+ handling, activating compensatory signaling pathways, increasing antioxidant capacity, and causing angiogenesis (the growth of new blood vessels from pre-existing vessels) [11, 12, 13, 14].

Another secondary complication resulting from the disuse of muscles is the development of contractures. Contractures are the loss of joint motion due to tightening of muscle, tendons, and ligaments. The force generated by a muscle is a factor of the length at which the muscle contracts. Therefore, if a muscle is held in a shortened position, as is the case with contractures, it will be further weakened. In the presence of primary muscle weakness, any further secondary loss of muscle strength should be avoided at all costs. Typical physical therapy interventions that are administered to people with DMD in order to prevent contractures include active, active-assisted, or passive stretching. [11, 14, 15].

2.2 Orthotic and Robotic Devices for People with DMD

2.2.1 Quality of Life and Self Sufficiency

As the disease progresses, individuals with DMD lose the ability to independently perform simple tasks such as grooming, scratching, drinking, eating, and picking up or placing objects [16]. It has been shown that there is a direct correlation between preserved independence and self-sufficiency and optimal quality of life [17]. Orthotic and robotic devices for the upper extremities have the common goal of providing independent manipulation for people with DMD by preserving self-sufficiency, increasing independence, allowing privacy, and therefore maintaining an optimal sense of well-being [18]. Moreover, individuals with DMD have difficulty or the inability to perform simple job related tasks. According to a study by Schuyler et al., a slight increase in manipulation ability and strength, which can potentially be provided by an

upper extremity orthosis or robotic manipulator, more than doubles the number of jobs that a person can be eligible for [19].

Despite the efforts to develop upper extremity orthotic and robotic devices for people with DMD to augment or replace the loss of functional abilities, only a small number have been commercialized and many prototypes have suffered difficulties such as high cost, high power consumption, heavy weight, low user acceptance, poor reliability, unacceptable appearance, poor function and complex user interface [16]. In addition, none of the devices that have become commercially available have been successful in the marketplace or widely distributed [18].

Current state of the art upper extremity assistive devices for people with DMD fall into one of three categories: passive orthoses, active orthoses, and robotic manipulators. The following sections discuss the current state of the upper extremity assistive device market for people with DMD, which includes commercially available orthoses and robotic manipulators as well as devices in the research and development stage.

2.2.2 Passive Orthoses

The Balanced Forearm Orthosis (BFO) shown in Figure 2.6, also known as the mobile arm support, is a passive device, meaning that it is powered by the body. Developed in 1965, the BFO allows movement in the horizontal plane for people with weak musculature, such as individuals with DMD. Enhanced versions of the BFO, including the commercially available Jaeco/Rancho MultiLink MAS developed by The Rancho Los Amigos Rehabilitation Engineering Program, compensate for the weight of the arm allowing vertical movements by canceling the nonlinear effects of gravity. Gravity

compensation allows subjects with DMD, who could not otherwise raise their arm against gravity, to perform a number of activities of daily living that are typically difficult or impossible to perform independently. These non-powered devices are preferred to powered devices in many cases because they use residual strength and natural control that is still present. However, the application of mobile arm supports has decreased in recent years due in part to the fact that they require specialists to fit and tune the devices in order to meet the individual needs and characteristics of the user. The complexity of adjusting these devices is demonstrated by the fact that the device will become unbalanced if the arm is misplaced as much as 5 mm. Furthermore, many currently available mobile arm supports have limited range of motion, non-perfect balancing, and comfort problems. Mobile arm supports require some muscle force in order to accelerate and decelerate the device as well as to overcome friction and balancing errors. BFO devices are severely lacking in their functionality in that they require the user to have the full functionality and grasping strength of their hand and wrist and do not consider a changing load that results from picking up objects [20, 21, 22, 23, 24].

Figure 2.6 Balanced Forearm Orthosis (BFO).
Source: [20]

There have been passive orthotic designs that aim to overcome the limitation of mobile arm supports that do not provide gravity compensation when the user picks up an object. The Anthropomobile Robot Arm (Armon), pictured in Figure 2.7, is a mobile arm support that allows the user to initiate electronic adjustments to the gravity balancing force whenever they pick up an object. Similarly, the Wilmington robotic exoskeleton (WREX) is a wheelchair mountable, passive arm orthosis that is gravity-balanced with rubber bands. Though it was not originally designed to adjust the gravity compensation to a varying load, Daniel et al. added an electronic component that allowed for dynamic gravity compensation. These devices have been shown to successfully increase the upper extremity range of motion of people with DMD, however; they require the user to adjust the gravity balancing force with switches or buttons whenever they pick up an object. This is cumbersome, non-intuitive, and requires the input of additional energy. In

addition, these devices retain the limitations of other mobile arm supports in that they require the user to have full hand and wrist strength and range of motion as well as proper fitting of the device in order to achieve a significant level of functionality [22,23, 25, 26].

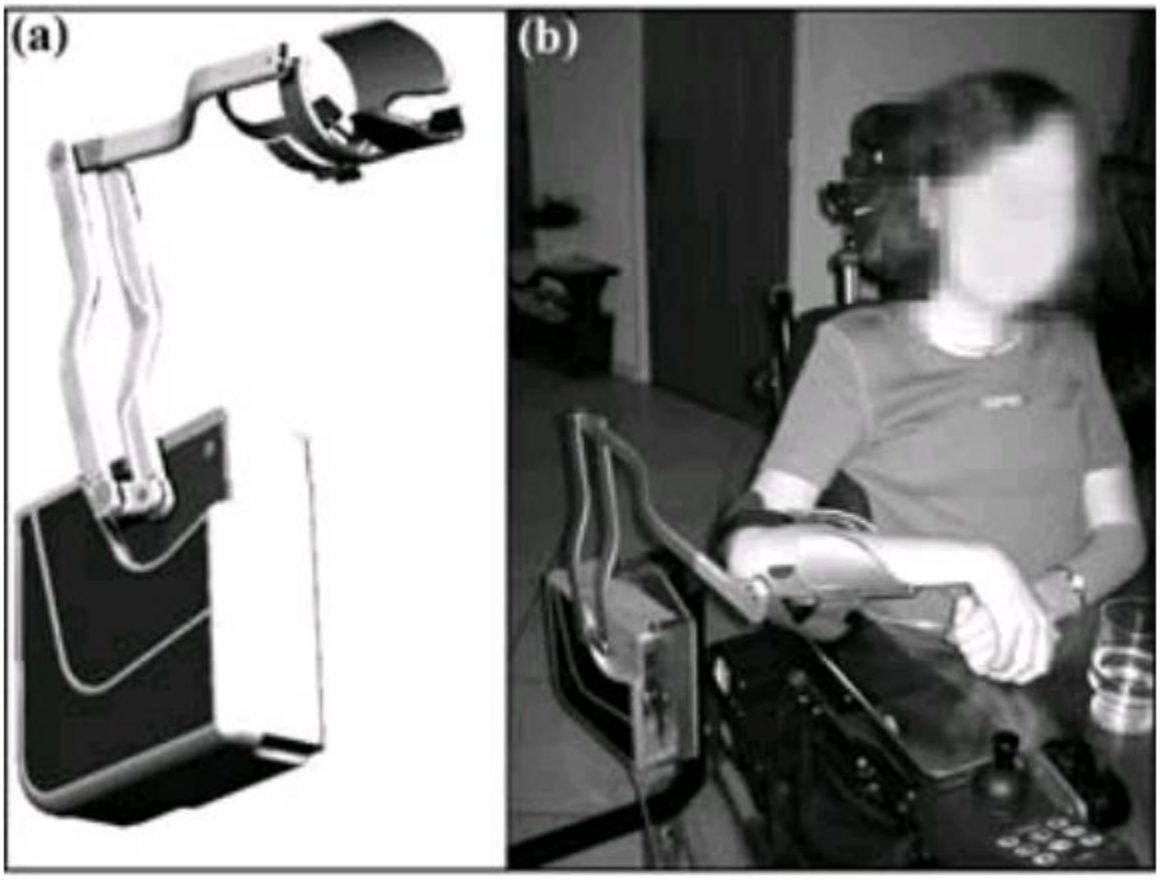

Figure 2.7 Armon Arm Support Prototype: (a) computer-aided drawing showing interface, parallelogram linkage, and box containing spring mechanism and (b) device in evaluation test.
Source: [23]

2.2.3 Active Orthoses

Active orthoses, also known as dynamic arm supports, are exoskeletons that support and direct the arm through the use of control inputs in order to perform tasks of daily living. These devices, such as the ARMin pictured in Figure 2.8, a six degree of freedom

exoskeleton, have the potential to enhance or augment muscle capacity and allow increased joint range of motion for people with DMD. Additionally, if an orthosis is made available in the early stages of the disease, it has been suggested that the user will maintain a larger range of motion over the course of progression because of the prevention of contractures. Unfortunately, as is the case with passive orthoses, these devices require the user to have full use of their hand, wrist, and fingers for the device to be functional in the independent performance of activities of daily living. Further limitations of exoskeletons include their large size, non-portability, substantial weight, power consumption and the fact that functionality is generally overshadowed by the burden on the user. Existing dynamic arm supports require more power for a full day of use than can be provided by a portable power source. An additional significant problem in designing adequate powered upper-limb orthoses is to manufacture them economically. To date, most powered orthoses are controlled by switches or joysticks controlled by the contralateral hand, head, or tongue. These control schemes are not intuitive and therefore require operating modes and training. Joysticks and switches are not well suited for people with decreased muscle strength because they require the user to grasp the device for a long period of time which results in fatigue [20, 27, 16, 28, 29, 30].

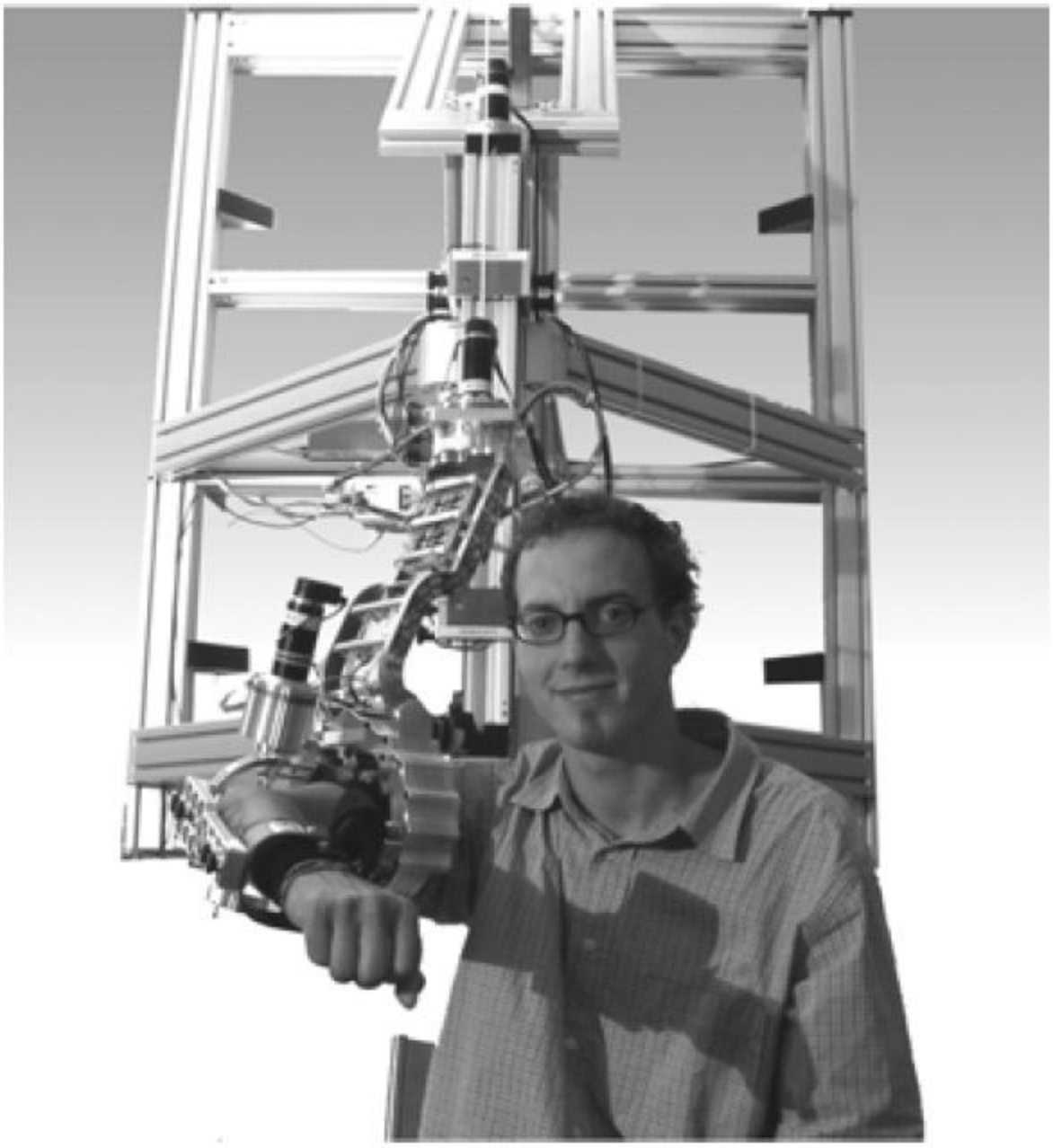

Figure 2.8 ARMin with a healthy subject.
Source: [29]

In an attempt to overcome the limitation of power consumption, Tsagarakis et al. designed a seven degree of freedom upper arm exoskeleton. The prototype, pictured in Figure 2.9, used a new kind of pneumatic Muscle Actuators (pMA) as the power source, which allow for simplicity, lightness, and decreased power consumption, but also the

need for compressed air. With the use of this prototype, motion is limited to less than the typical arm range of motion. This motion limitation is a direct result of the size of the pneumatic Muscle Actuators [31]. Similarly, Agrawal et al. attempted to overcome the power consumption limitation with the design of a cable driven upper arm exoskeleton. Though the prototype consumed less energy than the typical active orthosis, the design did not provide full functional capability because of the limited number of actuators as well as the unilateral properties of the cables which transmit power only in tension. The result was an exoskeleton workspace that was smaller than the potential reachable workspace [30].

Figure 2.9 Exoskeleton prototype using pneumatic Muscle Actuators (pMA) attached to a person's arm.
Source: [31]

Romilly et al. aimed to overcome the most significant limitation of most active orthoses which is the cost associated with manufacturing, rendering them unattainable for the everyday user with DMD. In order to manufacture a device with reasonable expense, the complexity of the device must be minimized. Romilly et al. argued that design compromises were necessary in order to meet this requirement. In order to minimize the complexity, they investigated potential options for decreasing the degrees of freedom in an exoskeleton design. Ultimately, they found that any potential option for decreasing the degrees of freedom below six restricted the number of tasks that can be performed by the user, resulting in a clear loss of functionality [27].

In an attempt to overcome the user interface limitation of most upper arm powered orthoses, Ramanatha et al. investigated the use of hand gestures to control an exoskeleton. Using a touchless interface with IR-sensors adapted to the motor capacity of the user, the device recognized six hand positions to control four degrees of freedom in an exoskeleton. However, the mapping of hand positions to control four degrees of freedom, though it prevents the fatigue associated with the use of joysticks and switches, is a similarly non-intuitive control mechanism. Further, the study concluded that the detection of hand movement is affected by factors such as illumination, light sources, skin reflectance, and characteristic hand movement [16].

2.2.4 Robotic Manipulators

Robotic manipulators perform tasks that require reaching and grasping to assist people with decreased arm strength, such as individuals with DMD, in order to allow independence in activities of daily living as well as potential facilitation of employment. The Assistive Robotic Manipulator (iARM), commercially available from Exact

Dyanmics, B.V., of The Netherlands, is a wheelchair mountable, six degree of freedom assistive robot with a gripper that allows people with impaired arm function to perform a range of activities such as eating, drinking, brushing teeth, scratching, and even delicate tasks such as handling a DVD or USB stick, painting, and putting in earrings. The iARM, pictured in Figure 2.10, uses interfaces such as a keypad, joystick, or single switch [32, 33]. The Teachmover manipulator from Microbot Inc., Mountain View, CA, is a wheelchair mountable assistive robot that has been commercially available for over 22 years. Similar to the iARM, it is powered with a 12-V wheelchair battery. The interface to the Teachmover manipulator is a multibutton touchpad. However, the interface can be changed, as was the case in a study by Shramowiat et al., in which the interface was altered to use two toggle switches [34].

Similar studies performed by Bach et al. integrated both touch sensitive and toggle switch user interfaces for two six degree of freedom robots with grippers: the Cobra RS2 manipulator by Cobra, Darmstadt, West Germany and the Microbot 453-H manipulator by Movemaster, Mountain View, CA [35]. All of these robotic manipulators have resulted in an improved initiative and sense of independence in activities of daily living and decreased dependence on personal care attendants. In addition, the cost of a robotic manipulator can be offset by the decrease in attendant care costs [34, 35]. Even so, the progressive muscle weakness that is present in people with DMD impairs hand and wrist function that are essential for using button, joystick, and switch interfaces [17].

Fortunately, as demonstrated by the aforementioned studies, the majority of robotic manipulators allow for the integration of user-specific interfaces that can overcome this limitation with the integration of interfaces such as chin or head-position

control, sip-and-puff switches, and voice control [34]. However, all of these control schemes are non-intuitive and therefore require operating modes and training [16]. The adequacy of an assistive device is likely to be compared to the time it takes a personal care attendant to perform the same task, and it has been found that the majority of currently implemented interfaces require a significant amount of time to complete a task and frequently overshoot the target [18,34, 35, 16]. Even more, the development of user-specific interfaces to accommodate functional variations requires the time-consuming and costly work of a professional [34].

Figure 2.10 The intelligent Assist Robot Manipulator (iARM).
Source: [33]

In a study that aimed to overcome the limitation of non-intuitive, user-specific interfaces, Didi et al. developed a new control scheme for the Manus robot, called the Assistive Control System (ACS). The control scheme was designed to prevent the

requirement of repetitive actions to complete a task associated with mapping typical user input modalities to six-degree of freedom movements. In order to do so, the ACS used preprogrammed, commonly used gestures. Unexpectedly, it was found that users of this control system felt as though they were observers of an automated task instead of in control of the task. This was considered to be a major disadvantage because users specifically expressed their need to feel active and in control during the task execution [36].

CHAPTER 3

DESIGN

Diminished independence and quality of life due to progressive muscle weakness and decreased upper limb functionality underscores the need for successful design of an upper extremity assistive technology that restores arm function and allows for increased independence in the performance of activities of daily living. It must overcome the limitations of previous devices, be economically viable, user friendly and widely accessible to the DMD community. The design and implementation of such a device would allow the user to gain an important degree of independence, involvement and control otherwise missing from their lives. The result would greatly enhance quality of life and decrease dependency on specialized care.

3.1 Requirements

A successful upper extremity assistive device will provide upper extremity functionality by using the valuable mechanical and user interface aspects of current passive orthoses, active orthoses, and robotic manipulators while overcoming their limitations. The list of the following design requirements is summarized in Table 3.1.

A successful design will build upon the gravity-balancing features of passive and active orthoses which take into account that inadequate arm function in individuals with DMD is a result of decreased strength in anti-gravity muscles despite residual muscle strength. In addition, the use of the residual strength and natural control that is still present in these individuals has the potential to prevent the development of contractures

and allow for submaximal activity which has a potential positive effect on the five mechanisms of DMD pathophysiology. Furthermore, a successful design needs to incorporate the major objective of robotic manipulators, which is the ability to grasp and lift objects without the requirement of hand and wrist strength, a current limitation in the majority of passive and active orthoses.

A successful design will also need to overcome the major limitation of passive and active orthoses by eliminating the complex requirements of properly fitting, tuning, and placing the user's arm in the device. Further design requirements include overcoming the limitation of passive orthoses in being able to balance increased weight against gravity associated with picking up an object and to be able to do this without the requirement of additional user input. The device will need to achieve these limitations while being portable, without being too large in size or weight and without consuming too much power. The device must use an intuitive control scheme without the use of switches, joysticks, buttons, or other interface mechanisms that result in fatigue and all of which are associated with the inadequacy of mapping user input into complex movements, significantly increasing the time it takes to complete a task. This must be achieved without the use of preprogrammed movements in order to ensure that the user feels active and in control during task execution. All of this must be achieved without the compromise of joint range of motion, workspace, or degrees of freedom, a feat which is yet to be achieved by commercially available and prototype designs. And lastly, these improvements and modifications must be accomplished without requiring unreasonable expense or complexity.

Table 3.1 Design Requirements

Design Requirements
Uses residual strength
Intuitive control paradigm
Ability to grasp and lift objects without the requirement of grasping strength
Gravity balancing capability
Gravity balancing capability in the presence of increased weight, without the use of additional user input
No requirement for user-specific interface alterations to account for functional variations in hand and wrist function
No major fitting, tuning, or adjustments required
Portable
Power consumption that allows for a portable power source
Light and compact
Reasonable expense and complexity
No compromise of joint range of motion, workspace, or degrees of freedom

3.2 Design Overview

The design presented combines the best features of passive and active orthoses and robotic manipulators, while overcoming the limitations of all three, by adding admittance control as the control paradigm to a preexisting robotic manipulator. Admittance control, though not routinely used in conventional robots, is the mapping of force to motion (compared to its inverse, impedance control, which is the mapping of motion to force). With the implementation of admittance control, the user exerts a force on a device and the device reacts with proper displacement via position or velocity control. Admittance control allows freedom in mechanical design because backlash and tip inertia can be eliminated. The inner loop of the admittance control model cancels the actual mass and friction of the device and any force or mass encountered by the end effector of the robotic manipulator is intrinsically registered and similarly cancelled [37]. This causes the user to feel as though they are pushing an object of almost negligible mass. Furthermore,

admittance control allows for the implementation of a force equal and opposite to the force of gravity which will provide antigravity support that is independent of any change in mass associated with objects in the robot's gripper. These features of admittance control allow for the user to employ any residual strength to control the trajectory of the robot end effector without the requirement of strength sufficient to overcome gravity and the mass and friction of the robot. Even more, admittance control is a truly intuitive control paradigm in that the user is moving their arm in space together with the robot end effector.

The design involves the reengineering of the user interface for the iARM robot, pictured in Figure 2.10. The iARM robot is a six degree of freedom robot designed specifically for people with limited muscular capacity, such as individuals with DMD. The iARM has a reach radius of 90cm, conveyed in Figure 3.1, which allows for sufficiently large workspace. The implementation of admittance control with this robotic manipulator does not require compromise of the joint range of motion, workspace, or degrees of freedom of the robot. The iARM gripper opens to a maximum of 9cm and has a maximum lift capacity of 1.5kg, which can accommodate most objects involved in daily living activities and has rubber fingertips which allow for extra grip while holding an object. The iARM, with a weight of 9kg, can be easily installed on most power wheelchairs, is sufficiently compact, and can be powered by the wheelchair's electrical supply, using less than 24W. The use of a preexisting robotic manipulator, which is commercially available for $21,000 to $26,000, reduces the cost and complexity of the end device by eliminating the need to engineer a robotic manipulator in addition to the user interface. Finally, the implementation of admittance control with a robotic

manipulator does not involve the tuning, fitting or adjustments required of orthoses. Because this design uses the residual strength of the entire upper extremity instead of just the wrist, hand, and fingers, it does not require user-specific alterations to account for functional variations that are required with other, non-intuitive interfaces.

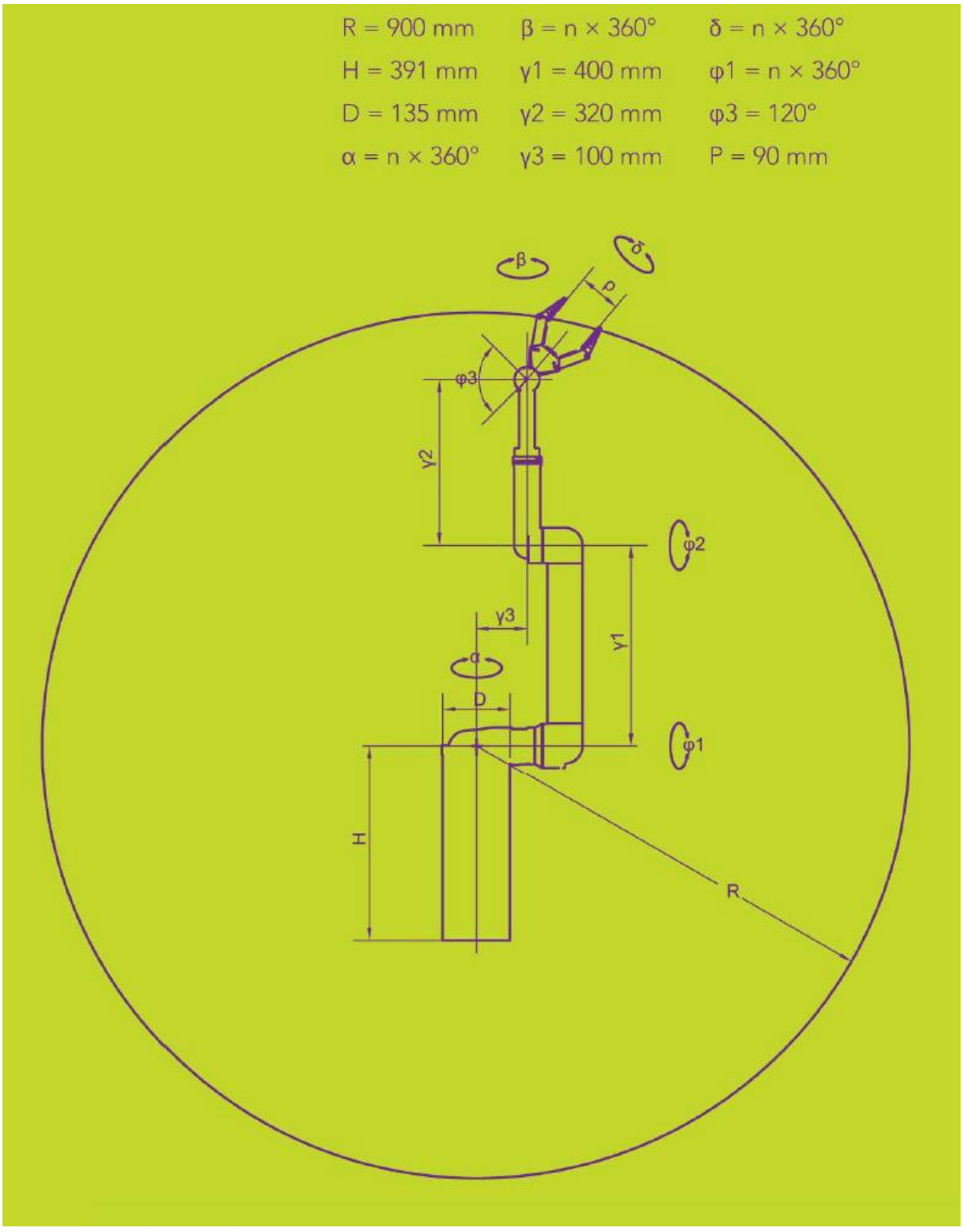

Figure 3.1 iARM joint ranges of motion, gripper opening, and workspace specifications.
Source: [33]

3.3 Implementation of Admittance Control

The following describes the implementation of admittance control as the interface for the iARM for three degrees of freedom (x, y, and z) in order to provide a proof of concept prototype.

3.3.1 Hardware

In order to implement admittance control the force of the user must be collected. This was done using the ATI Industrial Automation Mini45 Force/Torque Sensor pictured in Figure 3.2. This force/torque sensor reacts to applied forces and torques using Newton's third law: to every action there is an equal and opposite reaction. The sensor has the capability of detecting applied force in six degrees of freedom as is displayed in Figure 3.3. The Mini45 force sensor's rated sensing ranges, resolution, and counts value is available in Table 3.3. For this initial prototype the force detection in three degrees of freedom (x, y, and z) will be utilized. The force sensor outputs force data through the RS-232 serial port in resolved decimal integer unit in ASCII format. The value of one unit force can then be converted externally to the required units using known conversion rates for calculation purposes. The data are collected at a rate of 9,600 Baud, using ASCII mode and 3 axis output resulting in a throughput rate of 32Hz. With a sampling rate of 600Hz, the maximum transmission delay of the force sensor data is 6ms [38].

Figure 3.2 ATI Industrial Automation Mini45 Force Transducer.
Source: [38]

Table 3.3 ATI Industrial Automation Mini45 Specifications

Rated Sensing Ranges	
Fx	+/- 290 N
Fy	+/- 290N
Fz	+/- 580N
Resolution	
Fx	1/4N
Fy	1/4N
Fz	1/4N
Counts Value	
Fx	64/N
Fy	64/N
Fz	64/N

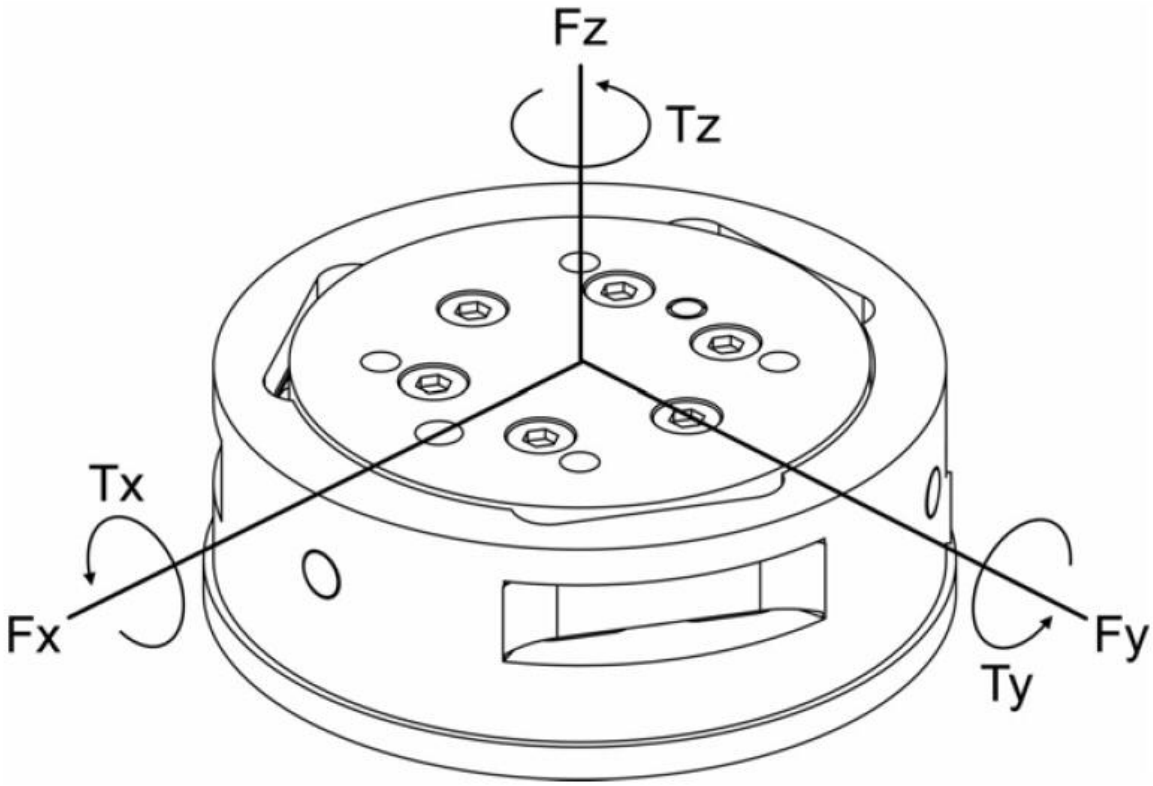

Figure 3.3 ATI Industrial Automation six degree of freedom force/torque sensor.
Source: [38]

In order to mount the force sensor to the iARM gripper and transmit the user's force to the force sensor, two aluminum plates and a handle were designed according to Figures A1, A2, and A3 in Appendix A. The two plates used to mount the force sensor were machined out of aluminum and allowed the sensor to be attached to the iARM using a double-sided adhesive. The final hardware design is pictured in figure 3.4.

Figure 3.4 Final hardware design.

3.3.2 Software

The iARM is inherently an impedance control robot, meaning that the input to the iARM is position and velocity commands. As a result, in implementing the desired control scheme there will be an inner and outer control loop, pictured in figure 3.5a. The inner control loop consists of the PID controller in the iARM firmware by Exact Dyanmics. The input to the inner control loop is position or velocity commands. The PID controller

uses inverse kinematics to output the joint angles of the robot necessary to achieve the desired position or velocity.

The outer loop consists of the MATLAB code and Simulink model developed as a part of this thesis. The MATLAB code, viewable in Appendix B, sets up the communication with the ATI Industrial Automation Force/Torque sensor. Next, the code biases the force sensor to ensure accurate force reading. At this point, if the user is holding onto the handle, the baseline sensor reading will include the force of the user's arm in the negative z direction, allowing for gravity compensation. Once the sensor is biased, the data collection begins. Any force transmitted by the subject to the handle of the force sensor will be converted into the appropriate motion using the Simulink model in Figure 3.5b. The input to the model is the force collected from the force sensor. To calculate acceleration, the force is then divided by a modeled mass. This mass, which the user will feel when driving the robot, will prevent infinite calculations of acceleration while being nearly negligible in value in order for the user to experience ease of movement while providing minimal force. The acceleration is then integrated in order to provide velocity. This velocity, multiplied by a damping constant, is fed back to the input of the model and subtracted from the input force. The damping constant ensures stability and was determined empirically to be 13 N-s/m. The Simulink model is run for 0.05 seconds for each force sample which allows for a pseudo-real-time simulation, and calculates the x, y, and z velocities separately. The velocity outputs of the Simulink model is then assigned to a MATLAB variable, which is the velocity value used to command the iARM, allowing the motors and not the user's muscles to move the user's arm.

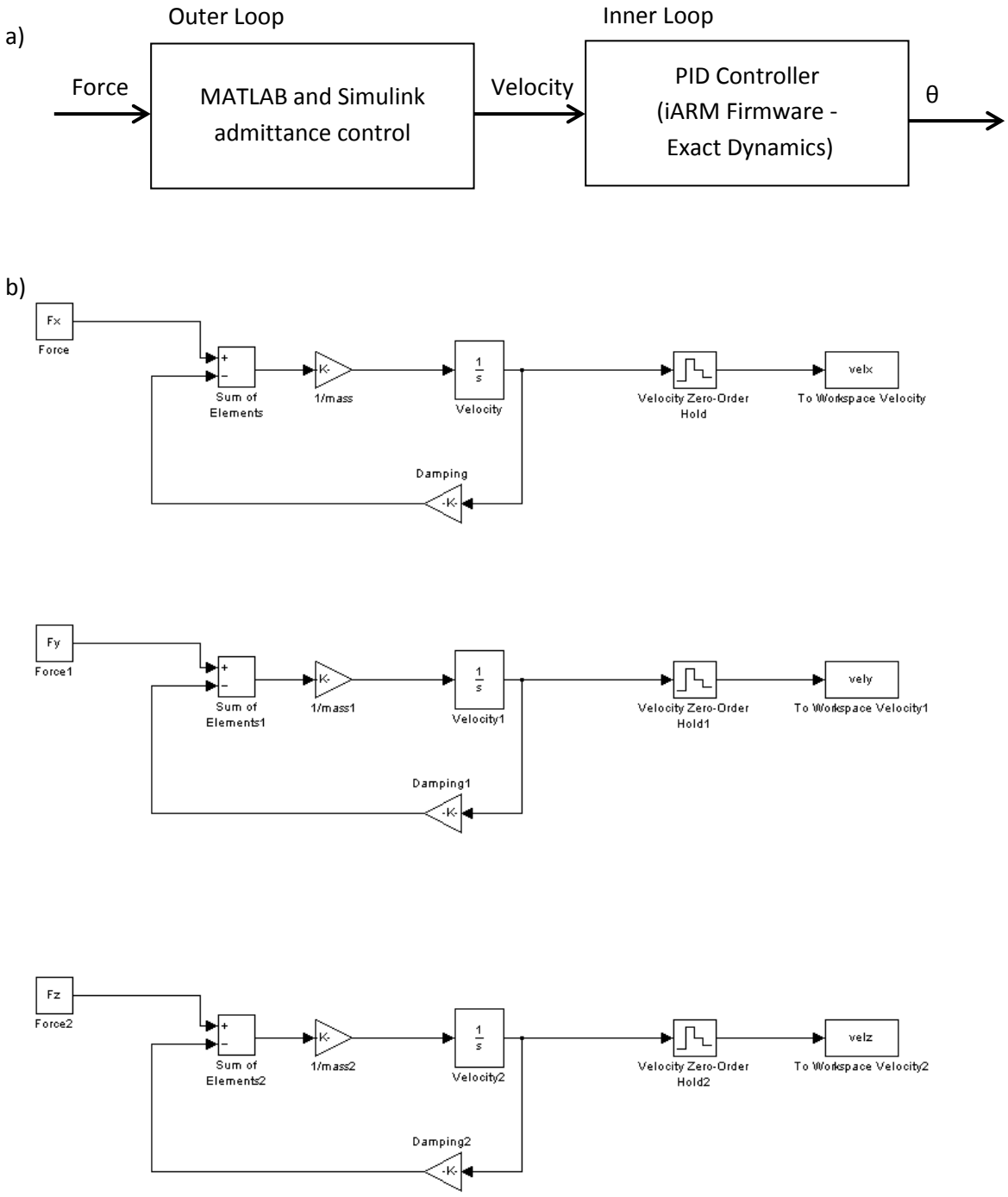

Figure 3.5 a) Inner and outer control loop and b) admittance control Simulink model for three Cartesian directions.

3.4 Delay

According to Shimoga [1992], the bandwidth with which the human finger can comfortably apply force and motion commands is 5 to 10 Hz, and 12Hz is the bandwidth beyond which the human finger cannot correct for positional disturbances [39]. It follows that for a human-machine interface to be comfortable there must be a minimum bandwidth of 10Hz, or maximum delay of 100ms. This criterion must be met in order to assure optimal and stable interaction between the human operator and the hardware [37]. In the iARM admittance control prototype, there are six potential contributors to the delay that is incurred: (1) the iARM motors, (2) the iARM firmware, (3) the MATLAB admittance control code, (4) the force sensor and (5) the communication between the computer and the iARM.

CHAPTER 4

METHODS

4.1 Quantification of Admittance Control Implementation

The implementation of admittance control as the control paradigm for the iArm was quantified through the collection of data from the iArm, force sensor and Simulink model and subsequent calculations for three distinct trajectories. The applied force was collected from the force sensor and plotted versus time. The output of the Simulink model (the Cartesian velocity that is sent as a command to the robot) was also collected and plotted versus time. The actual position of the iArm end effector was collected using the TrakSTAR and plotted versus time. And, position data was derived in order to calculate the actual end effector velocity versus time. This data was collected, calculated and plotted for the x, y, and z axis for three trajectories: a circular movement in the x,y plane, a circular movement in the x,z plane and a circular movement in the y,z plane.

4.2 Delay

As was previously stated, there are six potential contributors to the delay that is incurred with the implementation of admittance control using the iARM: (1) the iARM motors, (2) the iARM firmware, (3) the MATLAB admittance control code, (4) the force sensor and (5) the communication between the computer and the iARM. The force sensor is known to have a maximum transmission delay of 6ms. Furthermore, the iARM motors have a mechanical time constant, or time for the motor to reach 62.3% of its maximum rated speed, of 8.2ms. This means that there is room for about 85% of the optimum maximum

delay, or 85ms, to come from the MATLAB admittance control code, the communication between the computer and the iARM and the iARM firmware. The iARM firmware can be further broken into two categories: (1) the inverse kinematics and (2) any additional delay resulting from the iARM firmware. The following sections will explain the methods associated with quantifying the delay associated with the communication between the computer and the iARM, the MATLAB admittance control code and the inverse kinematics in the iARM firmware.

4.2.1 Delay in Communication Between PC and iARM

The delay in response time of the iArm was calculated for the use of the CAN-Bus, serial communication using firmware version 1.2 and serial communication using firmware version 1.3, in order to quantify the delay of the iArm as well as to determine if there was a difference in delay time when comparing the three communication modalities. In order to quantify the time of onset of movement of the iArm, the position of the end effector was collected using the TrakSTAR. The position data was then derived to give the velocity of the end effector. A threshold value was set at 2% of the maximum velocity and the onset of movement was determined based on the time corresponding to the first velocity value that was greater than the threshold [40]. The time at which the command was sent to the robot to initiate movement was determined using MATLAB's "tic" and "toc" commands. The delay was then calculated as the difference between the time at which the command was sent and the time of movement onset.

The procedure to calculate the delay in response time of the iArm was repeated ten times for each Cartesian direction. Data were collected for each of the three communication modalities: CAN-Bus, serial communication using firmware version 1.2

and serial communication using firmware version 1.3, to allow for comparison. The mean and standard deviation of the delay values for the ten trials was calculated for each axis in order to quantify the average delay of the iArm.

In order to determine whether there is a significant difference in delay between the three communication modalities, a statistical test needed to be performed on the sets of delay data. Before performing a statistical test to determine whether two samples are statistically the same or statistically different, it needed to be determined whether the data is normally distributed, thus aiding in the choice of a statistical test. A Kolmogorov-Smirnov test was used to determine whether the data has a standard normal distribution. The MATLAB function "kstest" was used to implement the Kolmogorov-Smirnov test. The function returns an H value of 1, (with a corresponding p value less than 0.05) if the test rejects the null hypothesis that the data has a standard normal distribution and an H value of 0, (with a corresponding p value greater than 0.05) is returned if the test accepts the null hypothesis and it is concluded that the data has a standard normal distribution. The test was performed with a 5% significance level. The test was run for all nine sets of delay data with a sample size of 10.

The Mann-Whitney U test, also referred to as the Wilcoxon rank-sum test, is a non-parametric test of the null hypothesis that two populations are the same. The Mann-Whitney U test was used to statistically compare the delay when using CAN-Bus, serial with firmware 1.2 and serial with firmware 1.3 communication modalities. The test was implemented using the MATLAB function "ranksum". The function returns an H value of 1 (with a corresponding p value less than 0.05) if the test rejects the null hypothesis that the two data sets are independent samples with equal medians and an H value of 0

(with a corresponding p value greater than 0.05) is returned if the test accepts the null hypothesis meaning that the two data sets being compared have statistically equal medians. The test was performed with a 5% significance level. The statistical comparison was performed separately for each axis.

4.2.2 Inverse Kinematics Delay

The next step was to determine whether the inverse kinematics caluclations, which are performed in the robot's internal software, contribute to the delay. In order to assess whether the inverse kinematic calculations cause a delay in the iArm movement, the iArm was commanded with a known Cartesian velocity that included x, y, and z velocities. The robot was then commanded to stop. The time that the start and stop commands were given were recorded using MATLAB's "tic" and "toc" commands. The iArm has an input that allows the user to collect the joint angles of all six joints at their current configuration. This was used in order to collect the joint angles of joint 1 though joint 6 at their initial position before the velocity command was sent and again at its final position after the stop command was sent. The change in joint angles divided by the time of trajectory gave the equivalent joint velocity command. As a result, equivalent joint velocity and Cartesian velocity commands were known to achieve the same trajectory. This allowed for the collection of delay data when the robot was commanded with Cartesian velocities and subsequent collection of data when the robot was commanded with joint velocities for an equivalent trajectory. Delay data was collected 10 times, using the same methodologies as described before to determine the delay in iArm response time, for the iArm being commanded with joint velocities and again for the iArm being commanded with Cartesian velocities. To allow for a more accurate

calculation of average error, the delay data was calculated using the derived x velocity, the derived y velocity, and the derived Z velocity separately.

In order to determine whether there is a significant difference in delay between commanding the robot with Cartesian velocity and commanding the robot with joint velocities, a statistical test needed to be performed on the two sets of data. A Kolmogorov-Smirnov test was again used, with the help of MATLAB's "kstest" function, to determine if the two sets of data are normally distributed, in order to assist in the choice of a statistical test. The test was performed with a 5% significance level. The Mann-Whitney U test was implemented, through the use of the MATLAB "ranksum" test, to statistically compare the delay when commanding the iArm with Cartesian versus joint velocities. The test was performed with a 5% significance level.

4.2.3 MATLAB Admittance Control Delay

The next step was to determine the increase in delay when the admittance control code is used. Because the implementation of admittance control requires more lines of MATLAB code and the use of a Simulink model, it was assumed that there would be an increase in the delay of the iArm. This is due to the increase in time to run the code, which includes reading the force sensor, running the Simulink model to determine the command velocity from the force sensor information, converting units, and finally sending a command to the robot.

In order to determine the increase in delay when the admittance control code is used, the delay of the iArm using admittance control must be quantified. Because admittance control involves the input of force into the system (and not simply a position or velocity command) the time onset of force as well as the time onset of movement of

the iArm end effector must be known. The difference will give the delay time of the iARM. The force sensor was mounted to the robot and the data was collected while the user conducted force in a single axial direction. This force was derived to give the change in force with respect to time allowing for a more accurate detection of the onset in force [40]. The onset time of force was determined to be 2% of the maximum value. The position of the iArm end effector was collected using the trakSTAR in the same single axial direction that the force was applied. These data was derived to give the velocity of the end effector versus time. Again, the onset in movement was determined to be the time at which the velocity value was greater than 2% of the maximum velocity value. This method was repeated ten times for each of the three axial directions x, y, and z. Also, with the implementation of admittance control, the total time for the MATLAB code and Simulink model to run was recorded using MATLAB's "tic" and "toc" command. The time it took to run the code for a single sample was recorded for an entire trial. These values were then averaged to give the time it takes to run the code for a single sample.

An effective delay was calculated as the total delay of the iARm when using admittance control minus the average time it takes to run the MATLAB admittance control code. The average time to run through the admittance control code was subtracted from each of the samples of delay data to give the effective delay, or the delay of the iArm that is not directly related to the runtime of the MATLAB code. This effective delay was then compared to the delay data collected for the serial communication using firmware 1.3 (because this was the communication modality used when implementing admittance control). The samples were first checked for normality

using the Kolmogorov-Smirnov test. Assuming the samples were not normally distributed, the Mann-Whitney U Test was performed in order to assess if there was a statistical difference between the effective delay and the delay associated with sending the iARM a single command.

4.3 HapticMASTER

4.3.1 HapticMASTER Delay

In order to investigate the potential of using the iArm for admittance control it was important to compare it to what is considered the gold standard of admittance control, the HapticMASTER. To do so, the delay of the haptic master was calculated using the same procedure described previously to determine delay. Delay data was determined for 10 trials for each of the three x, y, and z axis. Though the haptic master has a sample rate of 2500Hz, it was currently being controlled using MATLAB. Therefore, it would be assumed that the sample rate was decreased. Determining the delay of the HapticMASTER using MATLAB to program the robot allowed for the determination of whether or not the HapticMASTER was within the ideal delay parameters of 0.10 seconds.

4.3.2 HapticMASTER as the Interface for the iARM

Finally, in order to demonstrate the fact that the use of admittance control allowed for far superior control of a reaching and grasping robot such as the iArm, MATLAB code (viewable in Appendix C) was implemented allowing a user to control the iArm according to the position of their arm when connected to the cuff of the HapticMASTER and with their arm supported against gravity. This allowed for the demonstration of the

benefits of admittance control as a control paradigm for a reaching and grasping robot when the delay of the control system is less than the ideal delay of 0.10 seconds. The position of the iArm end effector and the position of the HapticMASTER end effector were plotted versus time, for two trials, in order to provide a plot for demonstration of this implementation.

CHAPTER 5

RESULTS

5.1 Quantification of Admittance Control Implementation

Figure 5.1 shows the trajectory for the circular motion in the x,y plane.

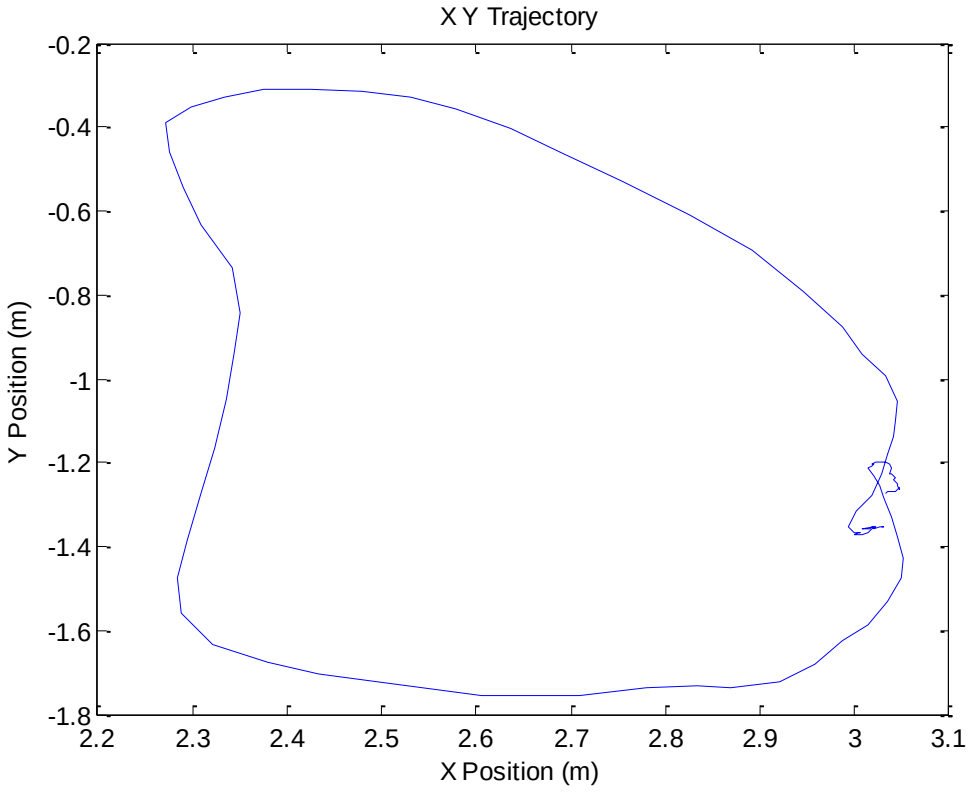

Figure 5.1 Circular Trajectory in x,y plane.

Figures 5.2, 5.3, and 5.4 show the plots of applied force, command velocity, gripper position, and derived gripper velocity versus time for the x, y, and z axes, respectively, or the circular trajectory in the x, y plane.

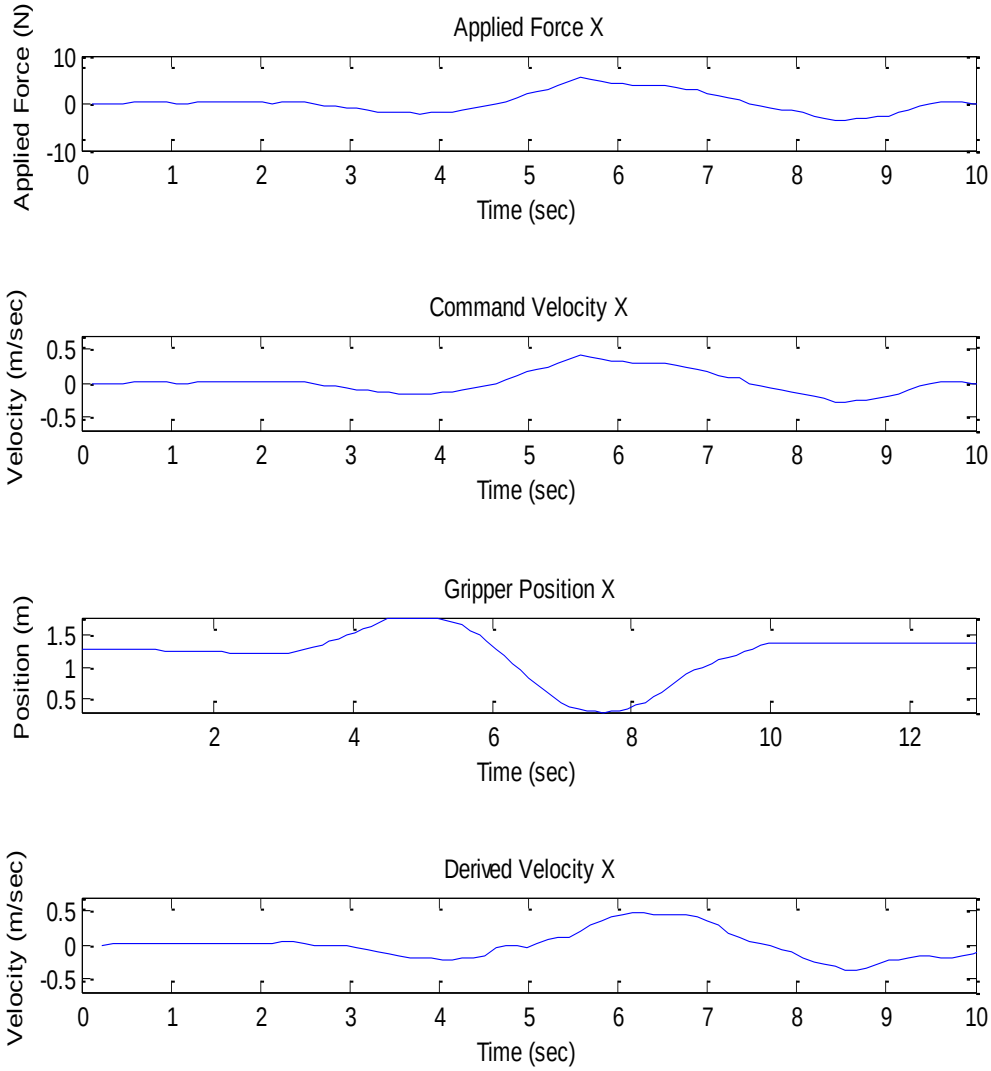

Figure 5.2 Applied force in the x direction, command velocity in the x direction, gripper position in the x direction, and derived gripper velocity in the x direction versus time for the circular trajectory in the x, y plane.

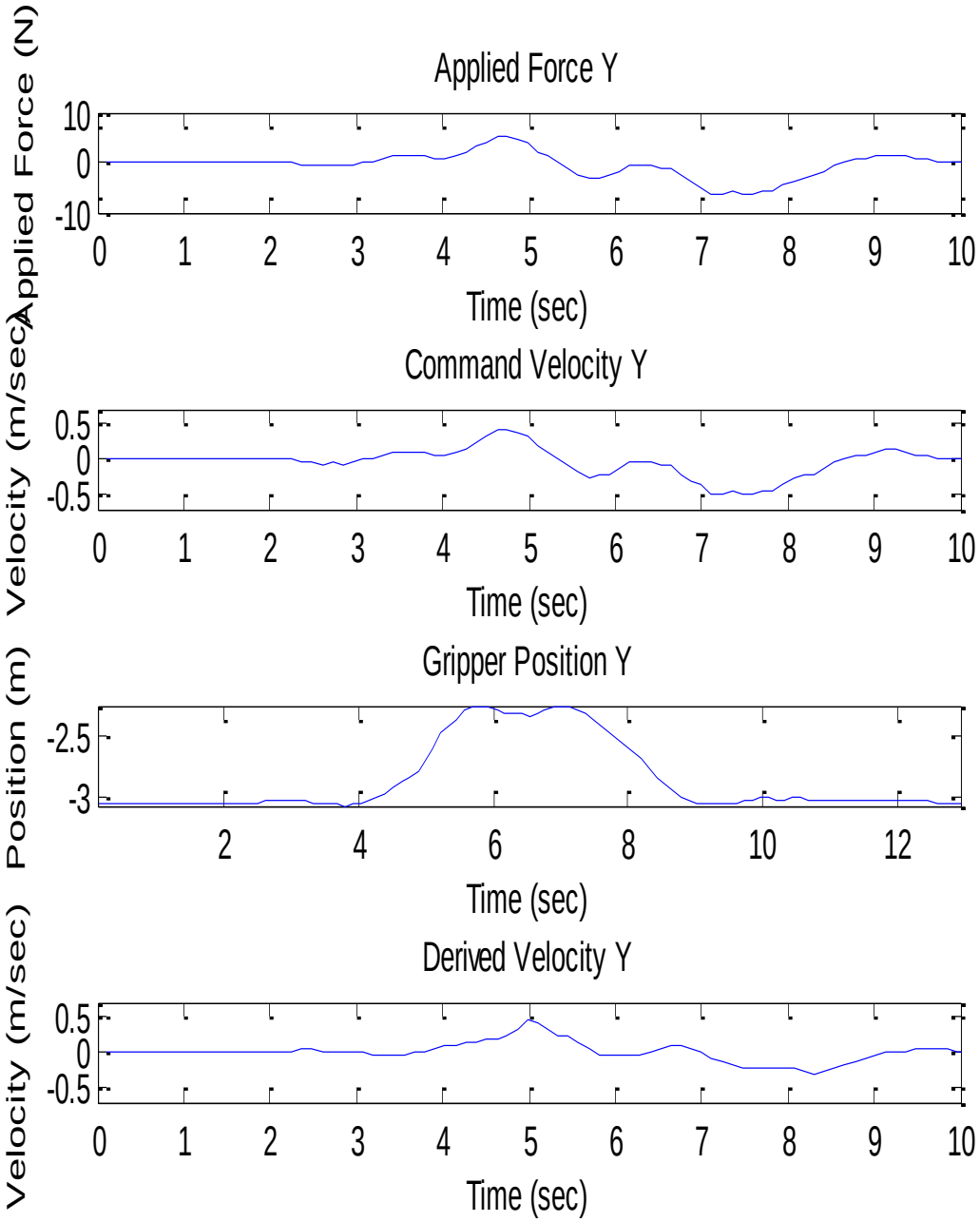

Figure 5.3 Applied force in the y direction, command velocity in the y direction, gripper position in the y direction, and derived gripper velocity in the y direction versus time for the circular trajectory in the x, y plane.

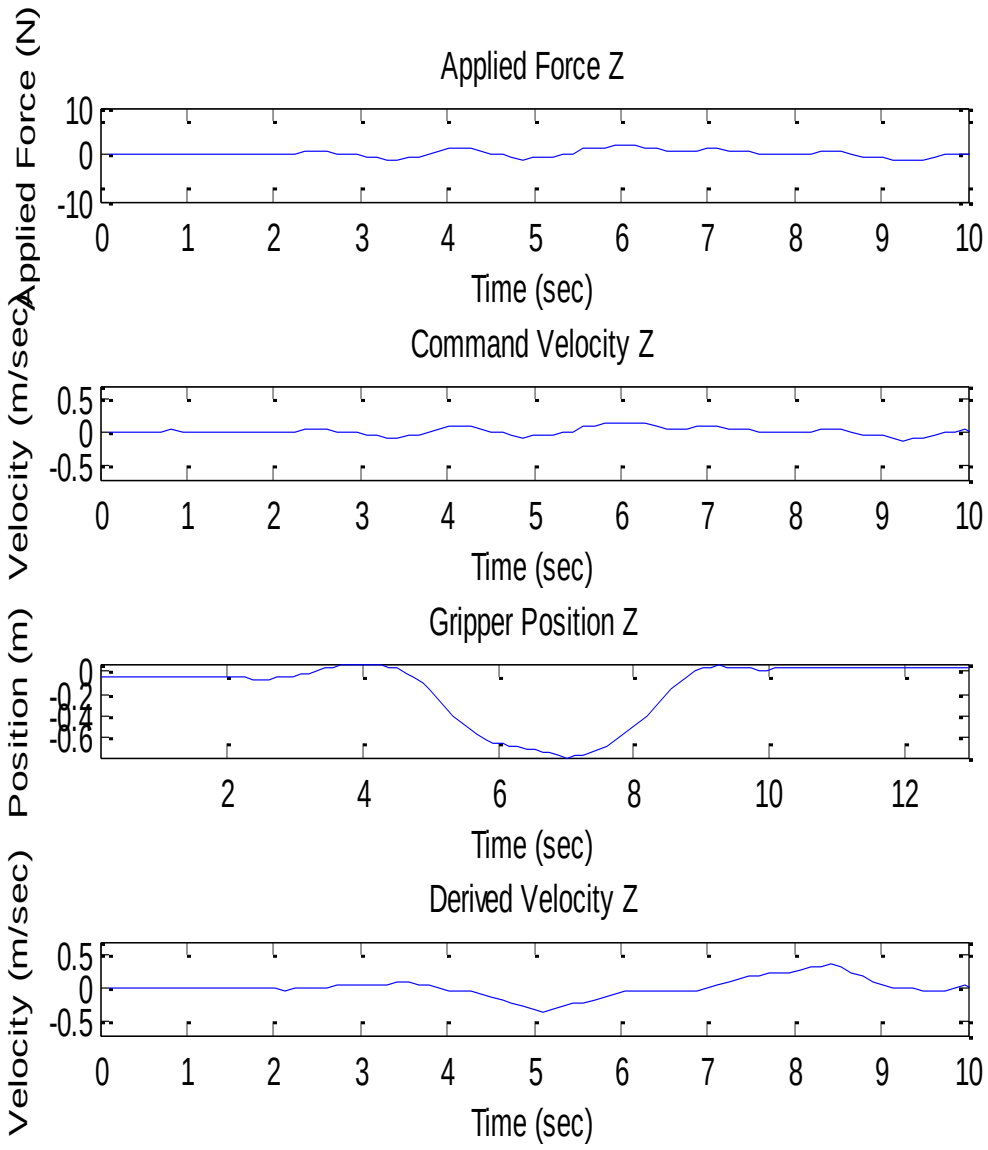

Figure 5.4 Applied force in the z direction, command velocity in the z direction, gripper position in the z direction, and derived gripper velocity in the z direction versus time for the circular trajectory in the x, y plane.

Figure 5.5 shows the trajectory for the circular motion in the x,y plane.

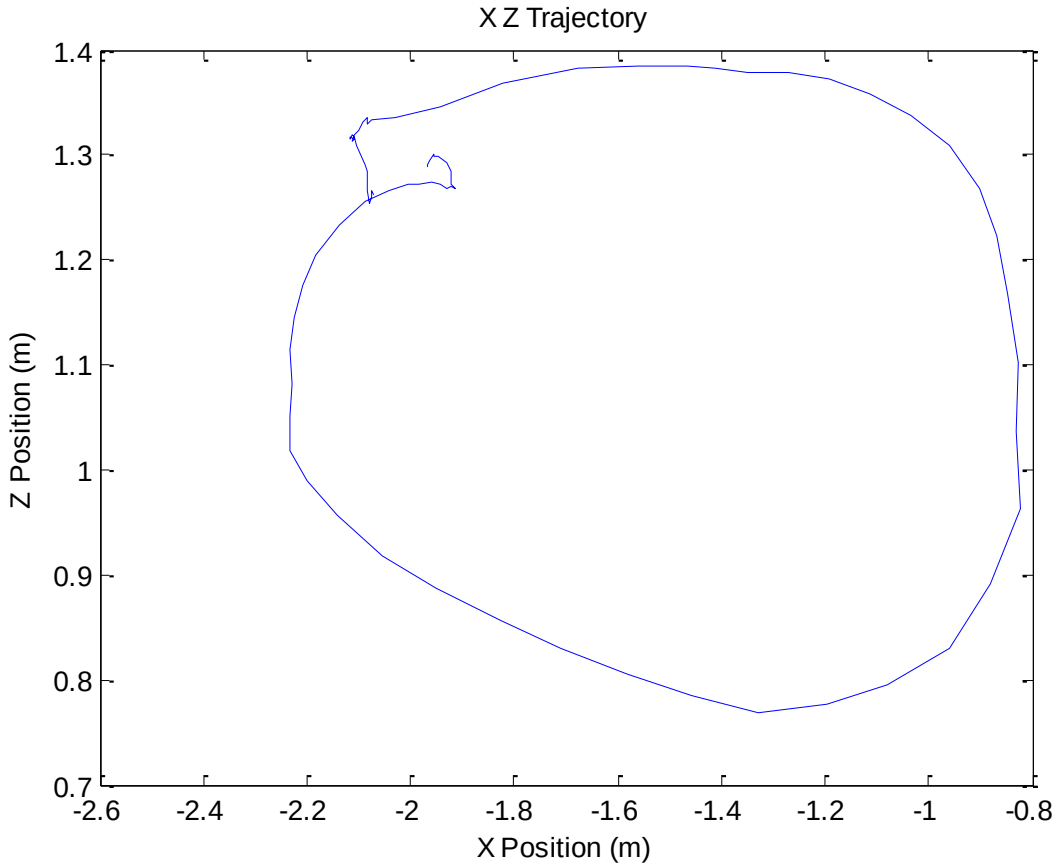

Figure 5.5 Circular Trajectory in x,z plane.

Figures 5.6, 5.7, and 5.8 show the plots of applied force, command velocity, gripper position, and derived gripper velocity versus time for the x, y, and z axes, respectively, for the circular trajectory in the x, z plane.

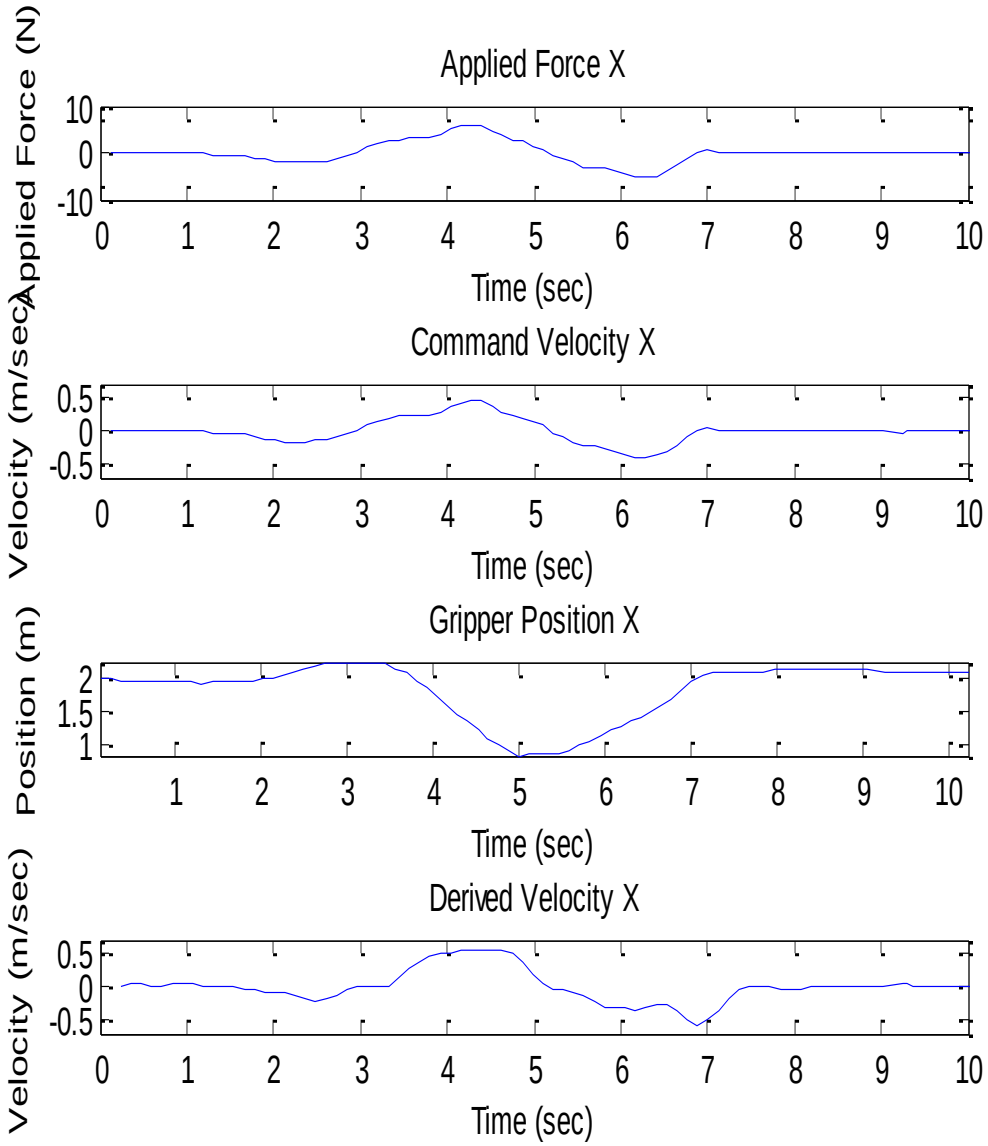

Figure 5.6 Applied force in the x direction, command velocity in the x direction, gripper position in the x direction, and derived gripper velocity in the x direction versus time for the circular trajectory in the x, z plane.

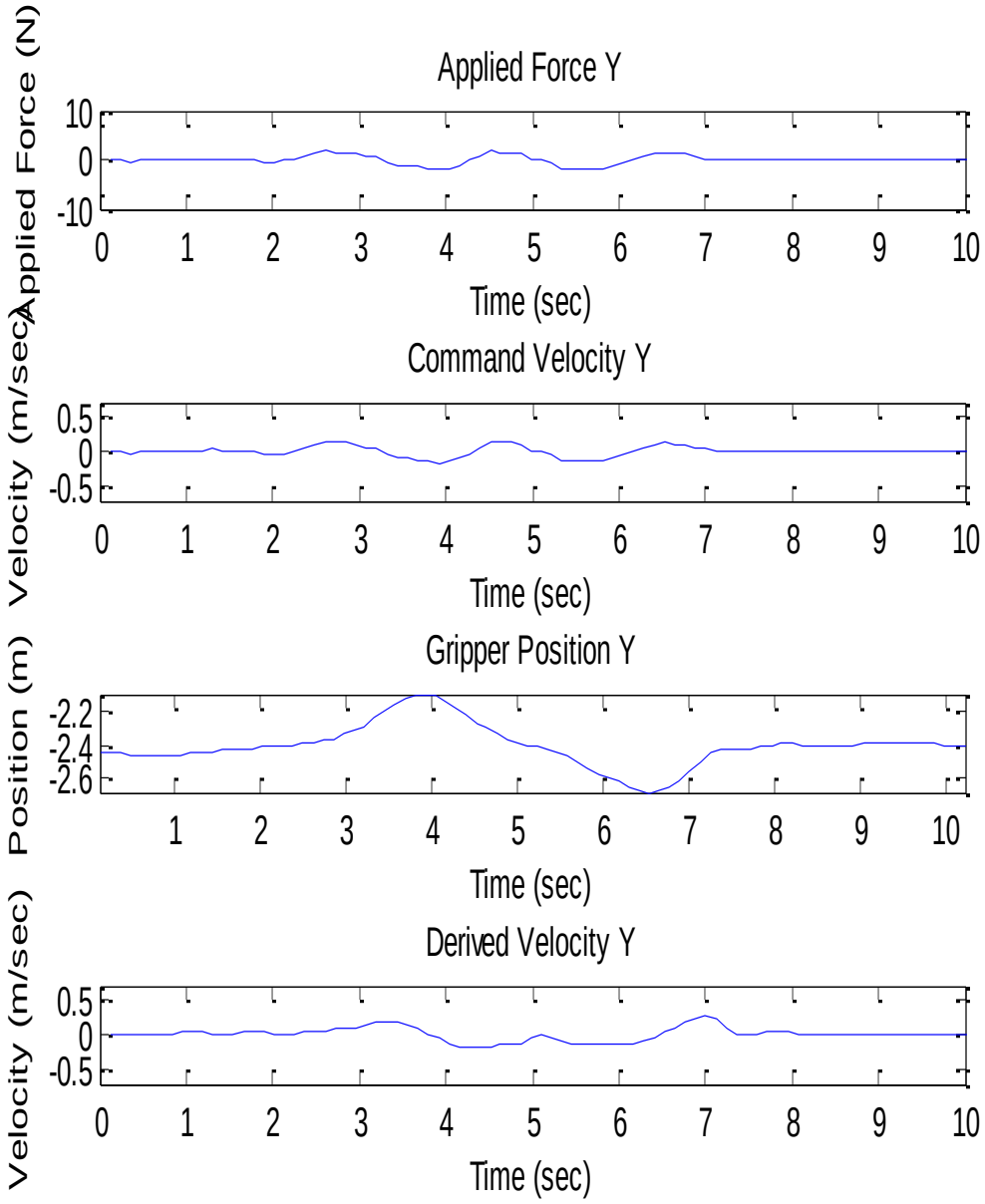

Figure 5.7 Applied force in the y direction, command velocity in the y direction, gripper position in the y direction, and derived gripper velocity in the y direction versus time for the circular trajectory in the x, z plane.

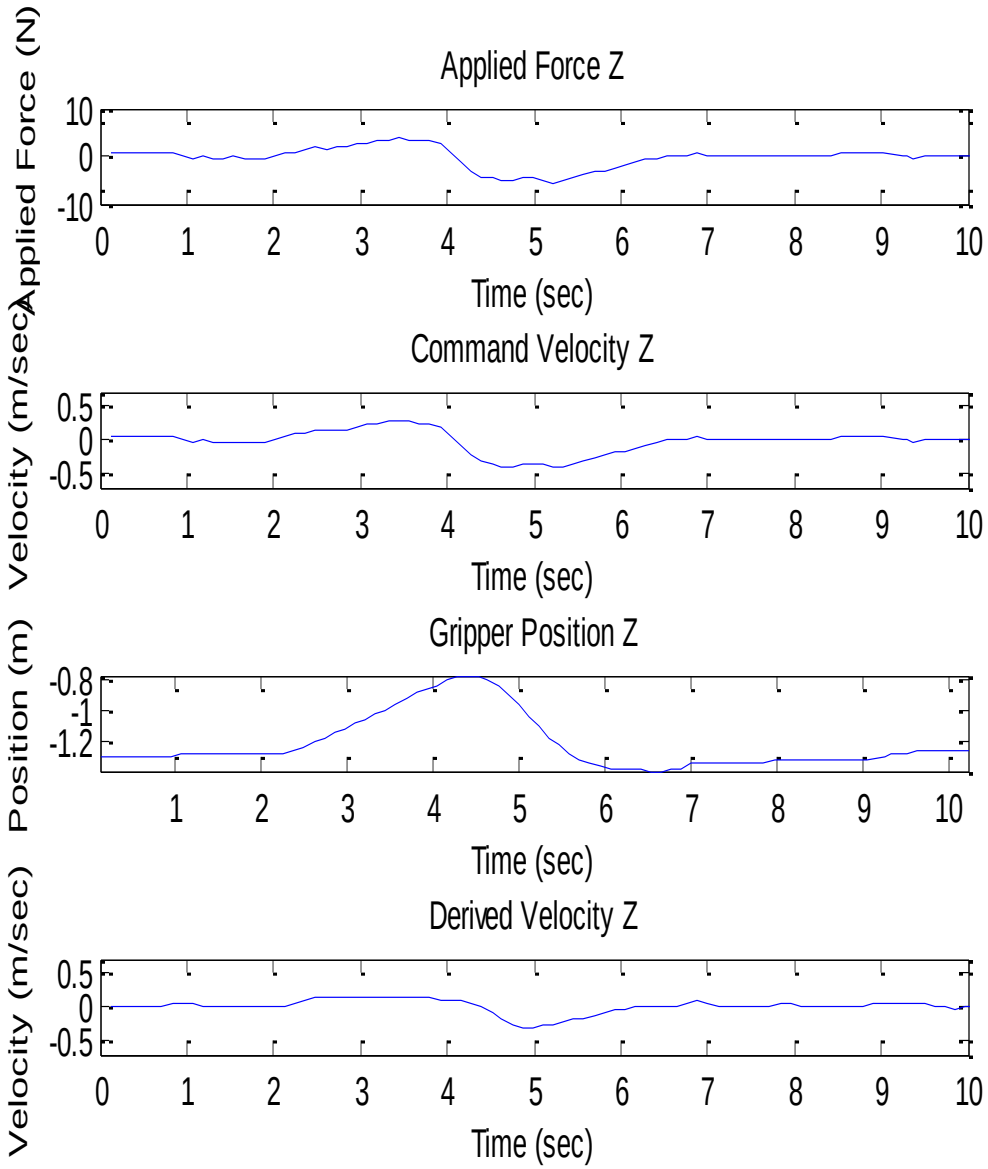

Figure 5.8 Applied force in the z direction, command velocity in the z direction, gripper position in the z direction, and derived gripper velocity in the z direction versus time for the circular trajectory in the x, z plane.

Figure 5.9 shows the trajectory for the circular motion in the y,z plane.

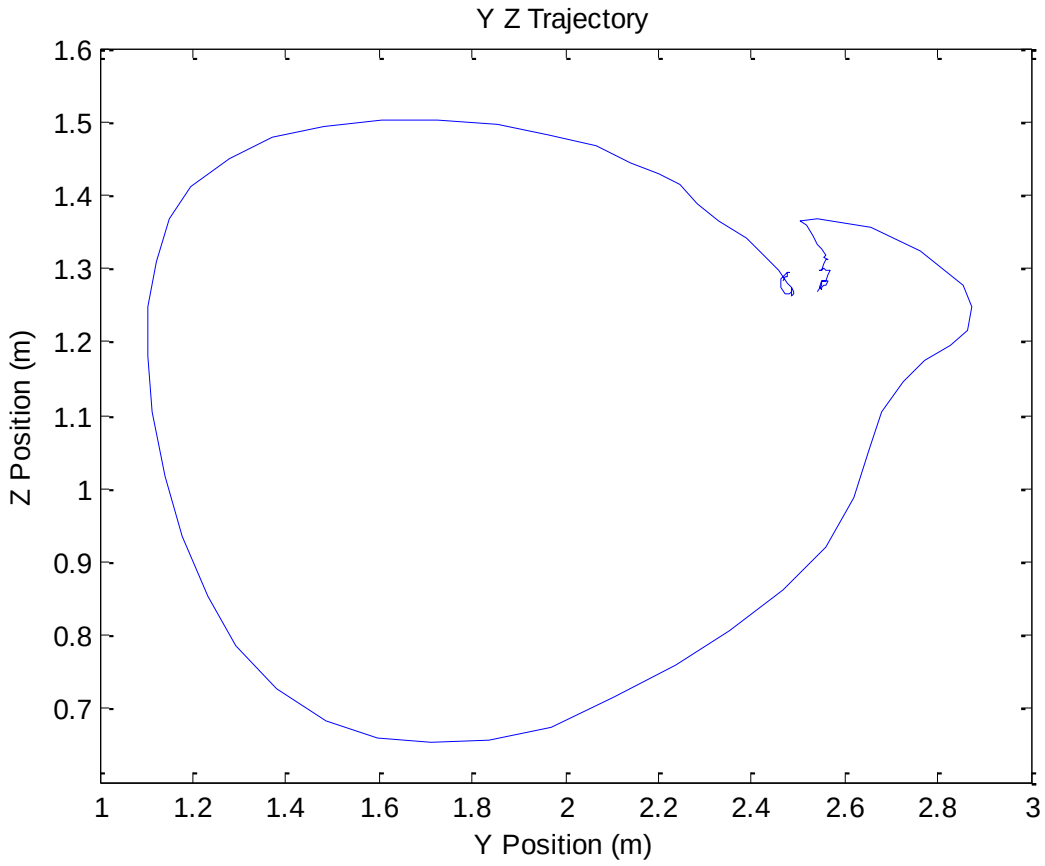

Figure 5.9 Circular Trajectory in y,z plane.

Figures 5.10, 5.11, and 5.12 show the plots of applied force, command velocity, gripper position, and derived gripper velocity versus time for the x, y, and z axes, respectively, for the circular trajectory in the y, z plane.

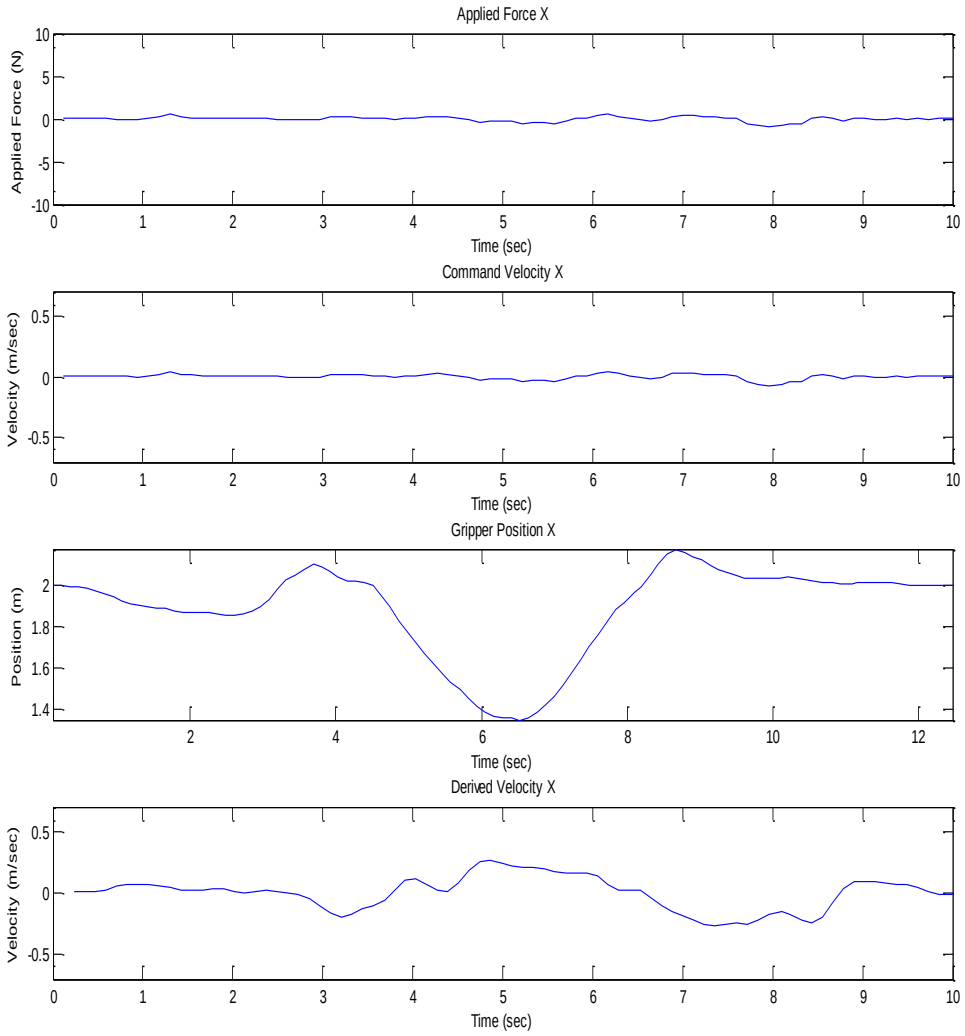

Figure 5.10 Applied force in the x direction, command velocity in the x direction, gripper position in the x direction, and derived gripper velocity in the x direction versus time for the circular trajectory in the y, z plane.

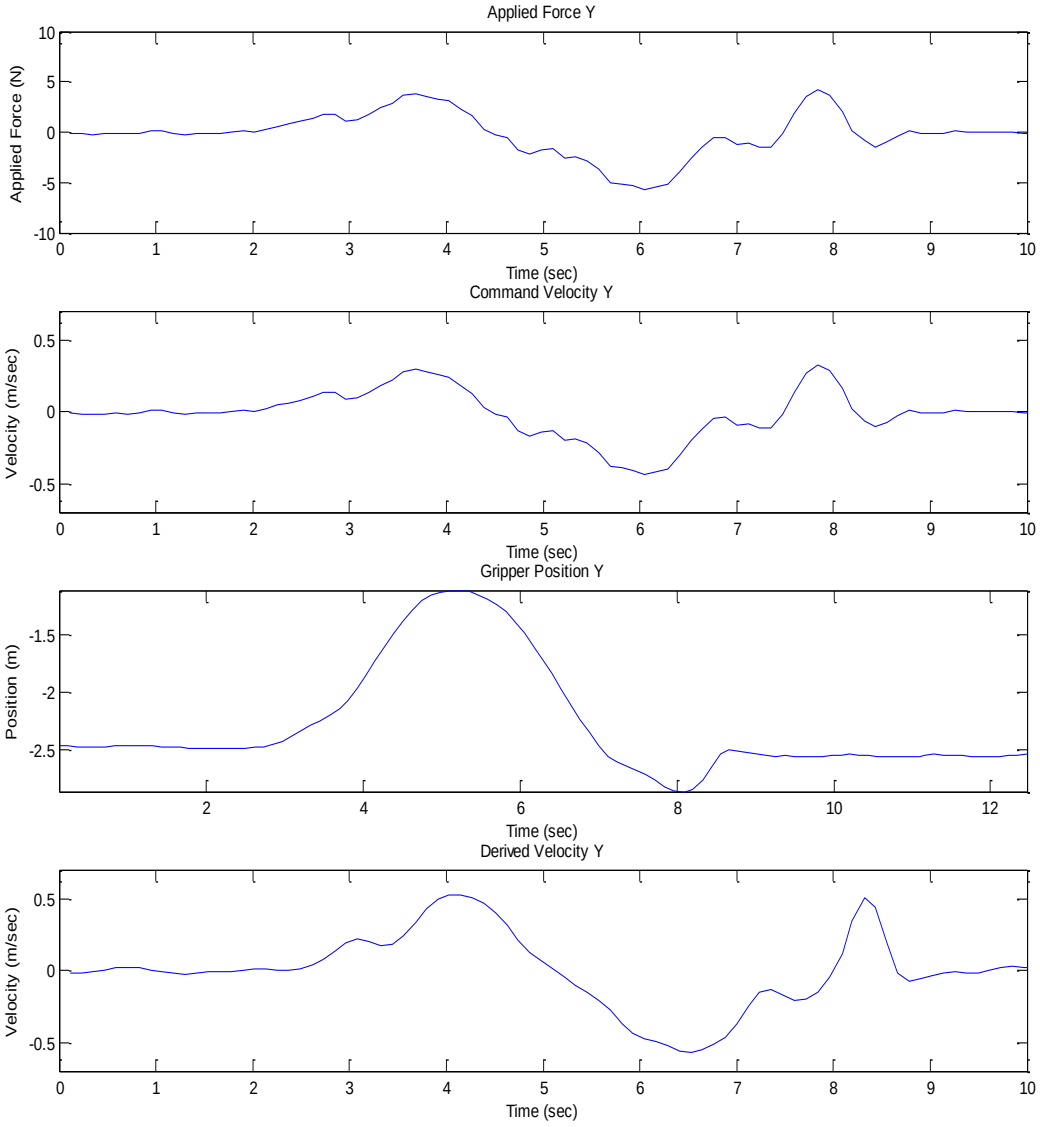

Figure 5.11 Applied force in the y direction, command velocity in the y direction, gripper position in the y direction, and derived gripper velocity in the y direction versus time for the circular trajectory in the y, z plane.

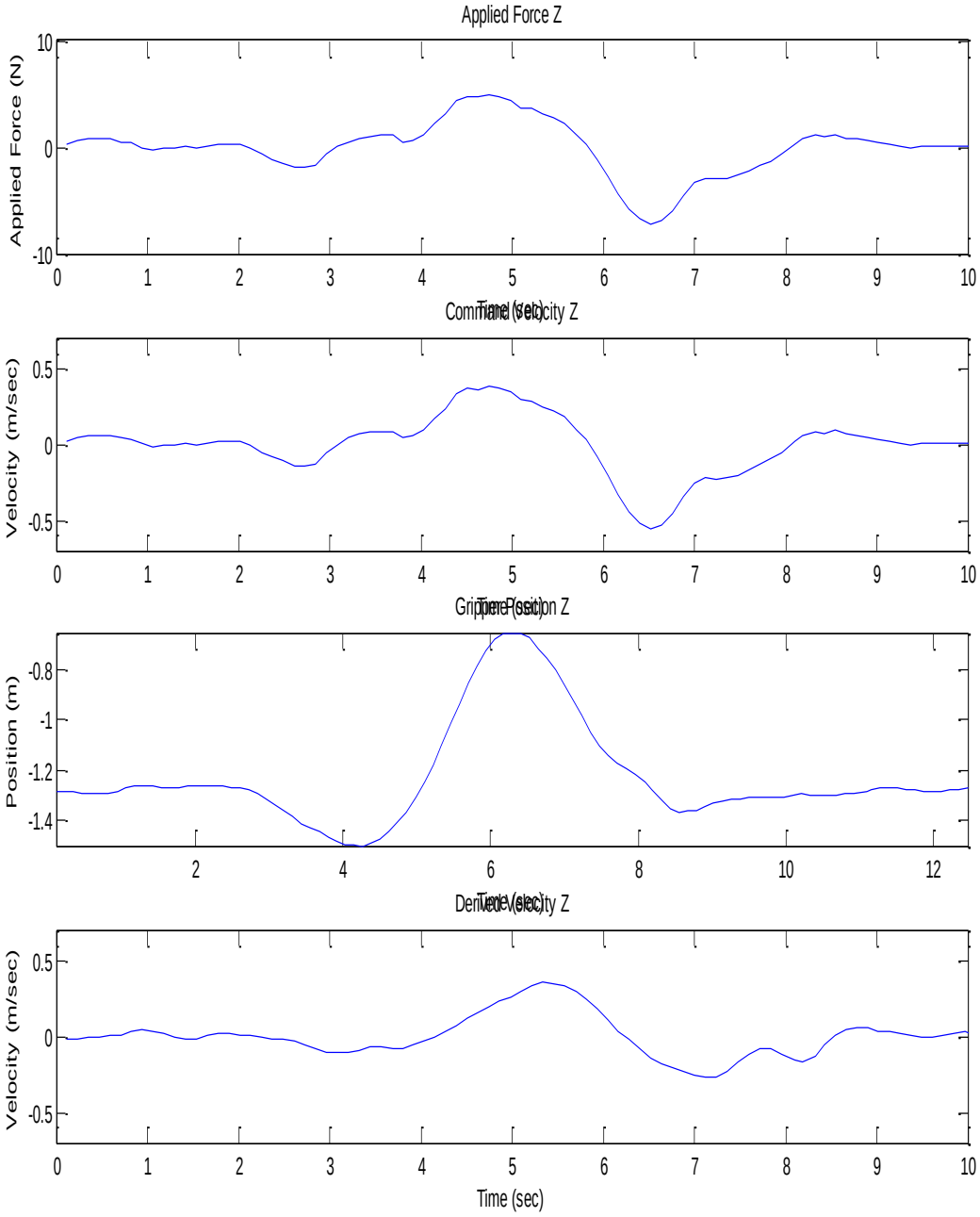

Figure 5.12 Applied force in the z direction, command velocity in the z direction, gripper position in the z direction, and derived gripper velocity in the z direction versus time for the circular trajectory in the y, z plane.

5.2 Delay

5.2.1 Delay in Communication Between the PC and the iARM

Figures 5.13, 5.14, and 5.15 are graphical representations of how the movement onset was determined for the CAN-Bus communication, serial communication with firmware 1.2, and serial communication with firmware 1.3, respectively.

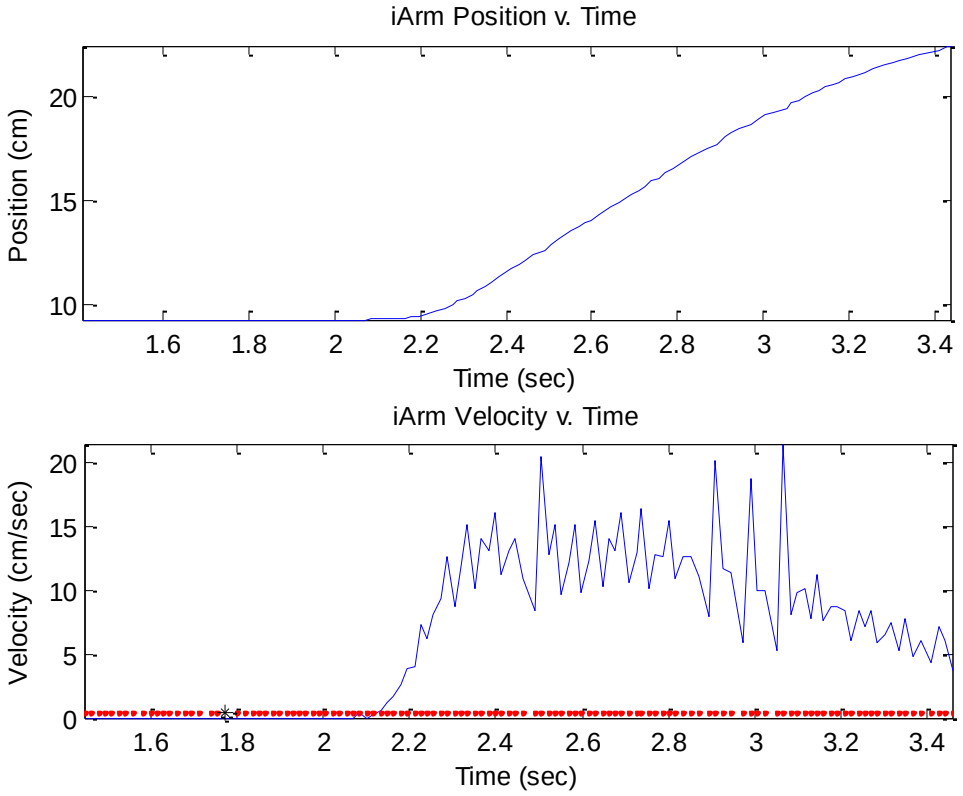

Figure 5.13 iARM position versus time (top) and iARM velocity versus time (bottom) using CAN-Bus communication.

Figure 5.13 contains two plots that demonstrate how the movement onset of the iArm end effector was determined for the CAN-bus communication. The top plot is the position of the iArm end effector versus time. The bottom plot is the change in position,

or calculated velocity versus time. The black star represents the point in time at which the command was send to the robot, which for this trial was 1.786 seconds. The red line represents the threshold, which is 2% of the maximum velocity. The corresponding time at which the velocity rises above this threshold value indicated the movement onset of the iArm end effector, which for this trial was 2.130 seconds. The delay of the iArm was then be calculated by subtracting the time at which the command was sent from the time of movement onset, which for this trial was calculated to be 0.344 seconds.

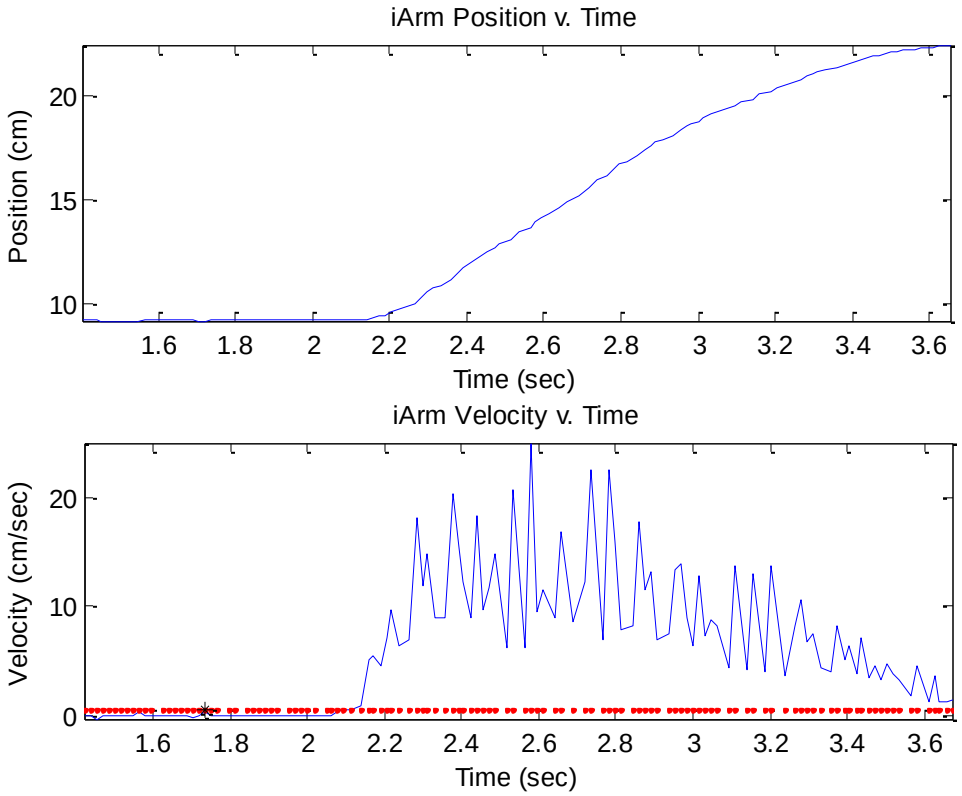

Figure 5.14 iARM position versus time (top) and iARM velocity versus time (bottom) using serial communication with firmware 1.2.

Figure 5.14 contains two plots that demonstrate the same method being used to calculate the delay of the iArm for serial communication with firmware 1.2. In this trial, the time at which the command was sent was 1.757 seconds. The movement onset was determined to be 2.101 seconds. As a result, the delay for this trial was calculated to be 0.344 seconds.

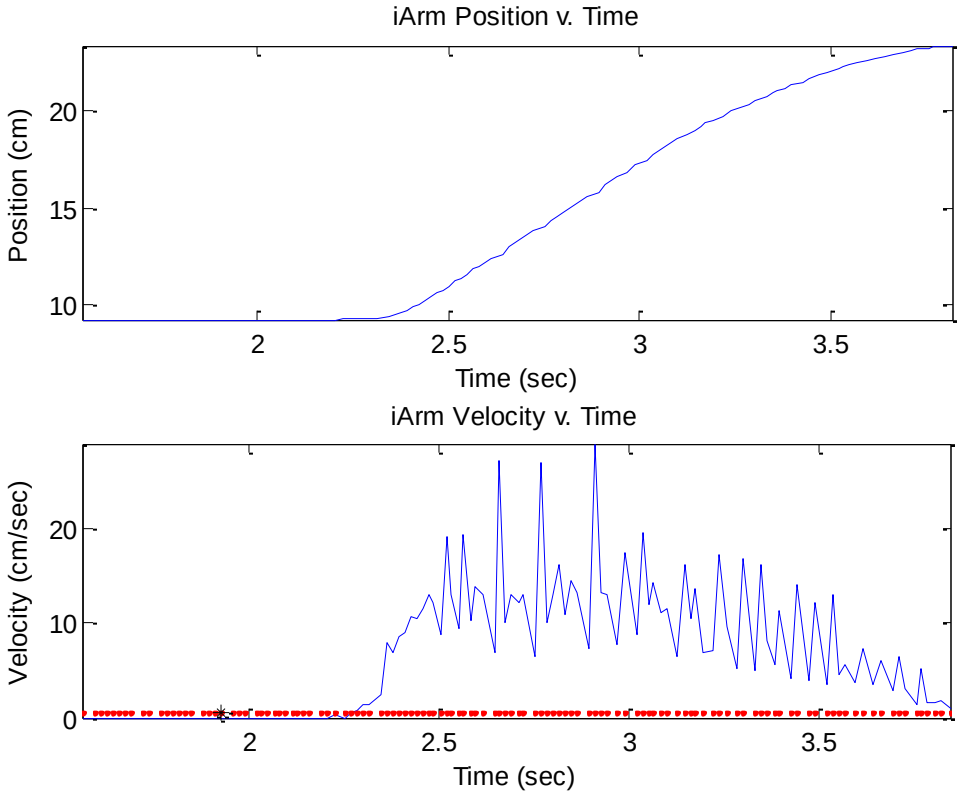

Figure 5.15 iARM position versus time (top) and iARM velocity versus time (bottom) using serial communication with firmware 1.3.

Figure 5.15 contains two plots that demonstrate the same method being used to calculate the delay of the iArm for serial communication with firmware 1.3. In this trial, the time at which the command was sent was 1.953 seconds. The movement onset was

determined to be 2.297 seconds. As a result, the delay for this trial was calculated to be 0.344 seconds.

Tables 5.1, 5.2, and 5.3 show the delay data collected for all three axes for CAN-Bus communication, serial communication using firmware 1.2, and serial communication using firmware 1.3, respectively.

Table 5.1 iARM Delay Using CAN-Bus Communication

	CAN-Bus		
Axis	**X**	**Y**	**Z**
1	0.344	0.299	0.264
2	0.327	0.294	0.313
3	0.356	0.313	0.281
4	0.327	0.314	0.280
5	0.344	0.280	0.311
6	0.328	0.298	0.294
7	0.324	0.282	0.326
8	0.345	0.312	0.326
9	0.346	0.316	0.327
10	0.329	0.311	0.295
Average	0.337	0.302	0.302
Standard Deviation	0.011	0.013	0.022

The average delay using the CAN-Bus communication was 0.337 seconds (with a standard deviation of 0.011) for the x axis, 0.302 seconds (with a standard deviation of 0.013 seconds) for the y axis, and 0.302 seconds (with a standard deviation of 0.002) for the z axis.

Table 5.2 iARM Delay Using Serial Communication with Firmware 1.2

	Firmware 1.2		
Axis	**X**	**Y**	**Z**
1	0.345	0.311	0.295
2	0.346	0.284	0.265
3	0.327	0.298	0.296
4	0.329	0.296	0.298
5	0.344	0.360	0.313
6	0.357	0.329	0.299
7	0.344	0.280	0.311
8	0.358	0.298	0.327
9	0.376	0.297	0.299
10	0.361	0.310	0.326
Average	0.349	0.306	0.303
Standard Deviation	0.015	0.024	0.018

The average delay using the serial communication with firmware version 1.2 was 0.349 seconds (with a standard deviation of 0.015) for the x axis, 0.306 seconds (with a standard deviation of 0.024 seconds) for the y axis, and 0.303 seconds (with a standard deviation of 0.018) for the z axis.

Table 5.3 iARM Delay Using Serial Communication with Firmware 1.3

	Firmware 1.3		
Axis	**X**	**Y**	**Z**
1	0.328	0.297	0.297
2	0.373	0.309	0.299
3	0.374	0.328	0.312
4	0.330	0.297	0.312
5	0.328	0.314	0.296
6	0.344	0.312	0.301
7	0.360	0.311	0.300
8	0.344	0.297	0.329
9	0.345	0.297	0.315
10	0.325	0.298	0.280
Average	0.345	0.306	0.304
Standard Deviation	0.018	0.011	0.014

The average delay using the serial communication with firmware version 1.2 was 0.345 seconds (with a standard deviation of 0.018) for the x axis, 0.306 seconds (with a standard deviation of 0.011 seconds) for the y axis, and 0.304 seconds (with a standard deviation of 0.014) for the z axis.

Table 5.4 shows the results of the Kolmogorov-Smirnov test. For each axis and communication instance, the collected data when tested for normality using the "kstest" matlab function the H value is 1 and the p value is significantly less than 0.05. This means that none of the delay data is normally distributed.

Table 5.4 Results of the Kolmogorov-Smirnov Test for Delay Data

	CAN-Bus		Serial: Firmware 1.3		Serial: Firmware 1.2	
axis	**H Value**	**p Value**	**H Value**	**p Value**	**H Value**	**p Value**
x	1	0.000	1	0.000	1	0.000
y	1	0.000	1	0.000	1	0.000
z	1	0.001	1	0.000	1	0.000

Tables 5.5, 5.6, and 5.7 show the results of the Mann-Whitney U test for axes x, y, and z, respectively.

Table 5.5 Results of the Mann-Whitney U Test for X Axis Delay Data

Mann-Whitney U Test: X axis						
	CAN-Bus		Serial Firmware 1.2		Serial Firmware 1.3	
	H Value	**p Value**	**H Value**	**p Value**	**H Value**	**P value**
CAN-Bus	-	-	0	0.068	0	0.361
Serial Firmware 1.2	-	-	-	-	0	0.494
Serial Firmware 1.3	-	-	-	-	-	-

The Mann-Whitney U test returned H values of 0 and corresponding p values of greater than 0.05 for the comparison of the delay when using the three different communication modalities when commanding a Cartesian velocity along the x axis to the iArm. According to the results of the Mann-Whitney U test, the null hypothesis that the compared data has statistically equal medians is accepted. In other words, according to the Mann-Whitney U test, there is no statistical difference in the delay of the iArm when comparing the use of Can-Bus communication, serial communication with firmware 1.2,

and serial communication with firmware 1.3 when a Cartesian velocity command along the x axis is sent to the robot.

Table 5.6 Results of the Mann-Whitney U Test for Y Axis Delay Data

Mann-Whitney U Test: Y axis						
	CAN-Bus		**Serial Firmware 1.2**		**Serial Firmware 1.3**	
	H Value	**p Value**	**H Value**	**p Value**	**H Value**	**P value**
CAN-Bus	-	-	0	0.791	0	0.970
Serial Firmware 1.2	-	-	-	-	0	0.647
Serial Firmware 1.3	-	-	-	-	-	-

The Mann-Whitney U test returned H values of 0 and corresponding p values of greater than 0.05 for the comparison of the delay when using the three different communication modalities when commanding a Cartesian velocity along the y axis to the iArm. According to the Mann-Whitney U test, there is no statistical difference in the delay of the iArm when comparing the use of Can-Bus communication, serial communication with firmware 1.2, and serial communication with firmware 1.3 when a Cartesian velocity command along the y axis is sent to the robot.

Table 5.7 Results of the Mann-Whitney U Test for Z Axis Delay Data

Mann-Whitney U Test: Z axis						
	CAN-Bus		**Serial Firmware 1.2**		**Serial Firmware 1.3**	
	H Value	**p Value**	**H Value**	**p Value**	**H Value**	**P value**
CAN-Bus	-	-	0	0.733	0	0.705
Serial Firmware 1.2	-	-	-	-	0	0.705
Serial Firmware 1.3	-	-	-	-	-	-

The Mann-Whitney U test returned H values of 0 and corresponding p values of greater than 0.05 for the comparison of the delay when using the three different communication modalities when commanding a Cartesian velocity along the z axis to the iArm. According to the Mann-Whitney U test, there is no statistical difference in the delay of the iArm when comparing the use of Can-Bus communication, serial communication with firmware 1.2, and serial communication with firmware 1.3 when a Cartesian velocity command along the z axis is sent to the robot.

It can be followed that there is no statistical different in the delay of the iArm when comparing the use of CAN-Bus communication, serial communication with firmware 1.2, and serial communication with firmware 1.3 when any Cartesian velocity command is sent to the robot.

5.2.2 Inverse Kinematics Delay

Table 5.8 shows the delay data collected from the 10 trials performed commanding the robot with a Cartesian velocity command that included velocity values in the x, y, and z direction.

Table 5.8 iARM Delay with Cartesian Velocity Command

Cartesian Command Delay				
Trial	**X**	**Y**	**Z**	**Average**
1	0.270	0.270	0.270	0.270
2	0.261	0.261	0.261	0.261
3	0.272	0.272	0.285	0.276
4	0.265	0.265	0.265	0.265
5	0.267	0.267	0.267	0.267
6	0.283	0.283	0.283	0.283
7	0.282	0.282	0.268	0.277
8	0.298	0.298	0.298	0.298
9	0.276	0.276	0.276	0.276
10	0.276	0.276	0.276	0.276
Average	0.275	0.275	0.275	0.275
Standard Deviation	0.011	0.011	0.011	0.011

According to collected data, the average delay that resulted from commanding the robot with a Cartesian velocity command was 0.275 seconds (with a standard deviation of 0.011).

Table 5.9 shows the delay data collected from the ten trials performed commanding the robot with a joint velocity command that included angular velocity values for joint 1 through joint 6.

Table 5.9 iARM Delay with Joint Velocity Command

Joint Command Delay				
Trial	**X**	**Y**	**Z**	**Average**
1	0.263	0.270	0.234	0.256
2	0.276	0.276	0.277	0.276
3	0.302	0.265	0.276	0.281
4	0.267	0.291	0.279	0.279
5	0.297	0.267	0.270	0.278
6	0.305	0.291	0.291	0.296
7	0.281	0.308	0.265	0.285
8	0.270	0.285	0.288	0.281
9	0.308	0.267	0.282	0.286
10	0.272	0.275	0.290	0.279
Average	0.284	0.280	0.275	0.280
Standard Deviation	0.017	0.014	0.017	0.010

According to collected data, the average delay that resulted from commanding the robot with a joint velocity command was 0.280 seconds (with a standard deviation of 0.010).

Table 5.10 shows the results of the Kolmogorov-Smirnov test on the two sets of data, (ten trials of delay when commanding the robot with Cartesian velocity and ten trials of delay when commanding the robot with joint velocity). The results of the Kolmogorov-Smirnov test were H values of 1 with corresponding p values of less than 0.05. This means that neither set of data is normally distributed.

Table 5.10 Results of the Kolmogorov-Smirnov Test on Cartesian Velocity Command and Joint Velocity Command Delay Data

Kolmogorov-Smirnov Test		
	H Value	**p Value**
Cartesian Velocity Command	1	0.001
Joint Velocity Command	1	0.001

Table 5.11 shows the results of the Mann-Whitney U test comparing the delay that results from commanding the iArm with Cartesian versus joint velocities. The Mann-Whitney U test returned H values of 0 and corresponding p values of greater than 0.05. According to these results, the null hypothesis that the compared data sets have a statistically equal mean is accepted. In other words, according to the Mann-Whitney U test, there is no statistical difference in the delay of the iArm when it is sent a Cartesian velocity command versus when it is sent a joint velocity command.

Table 5.11 Results of the Mann-Whitney U Test Comparing the iARM Delay with Cartesian Velocity versus Joint Velocity

Mann-Whitney U Test	
H Value	**p Value**
0	0.095

5.2.3 MATLAB Admittance Control Delay

Figures 5.16, 5.17, and 5.18 show graphical representations of the delay calculation from single trials with the implementation of admittance control from a trajectory in the x, y, and z directions, respectively.

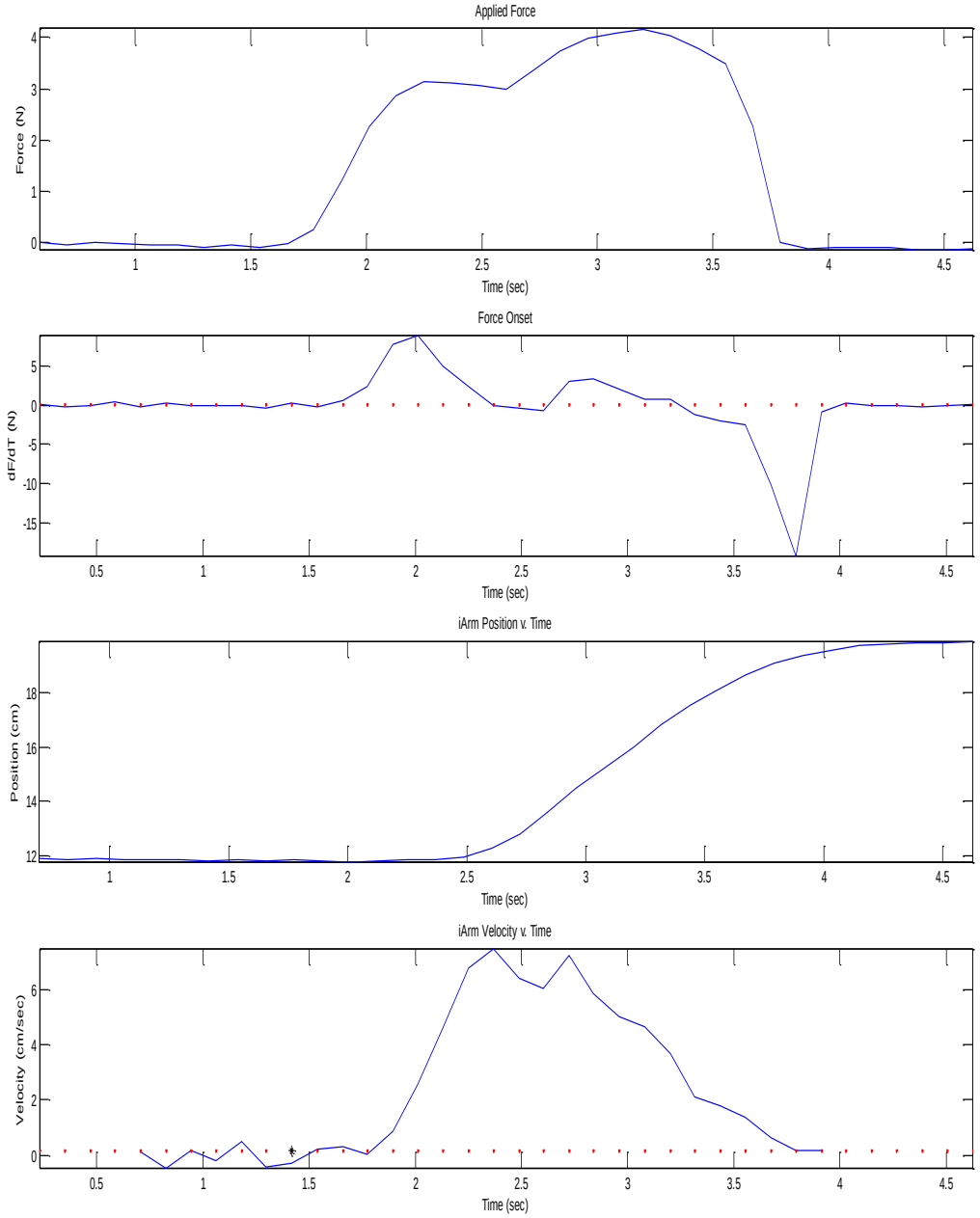

Figure 5.16 Applied force versus time (top), derived force versus time (second), iARM position versus time (third), and iARM velocity versus time in the x direction.

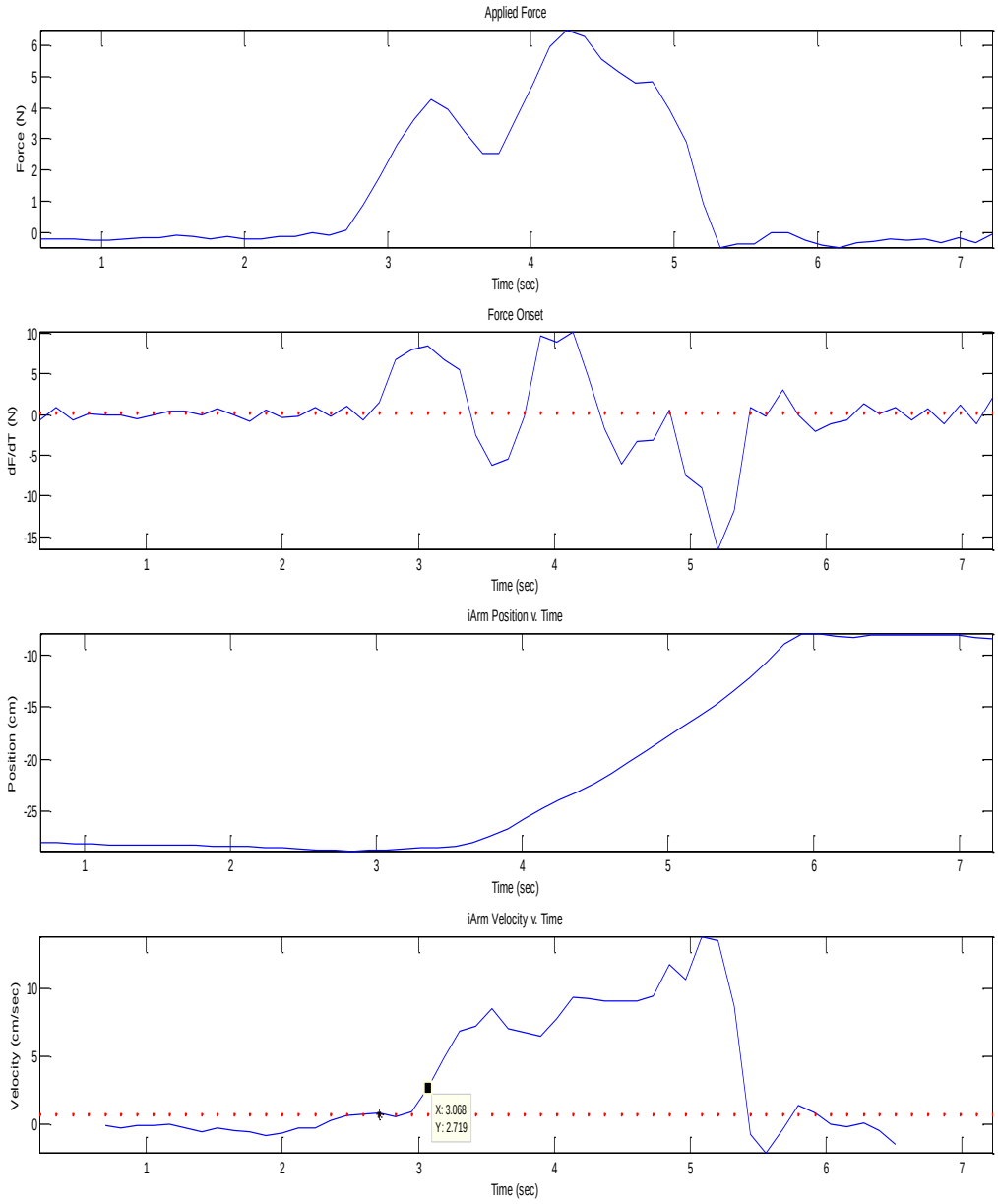

Figure 5.17 Applied force versus time (top), derived force versus time (second), iARM position versus time (third), and iARM velocity versus time in the y direction.

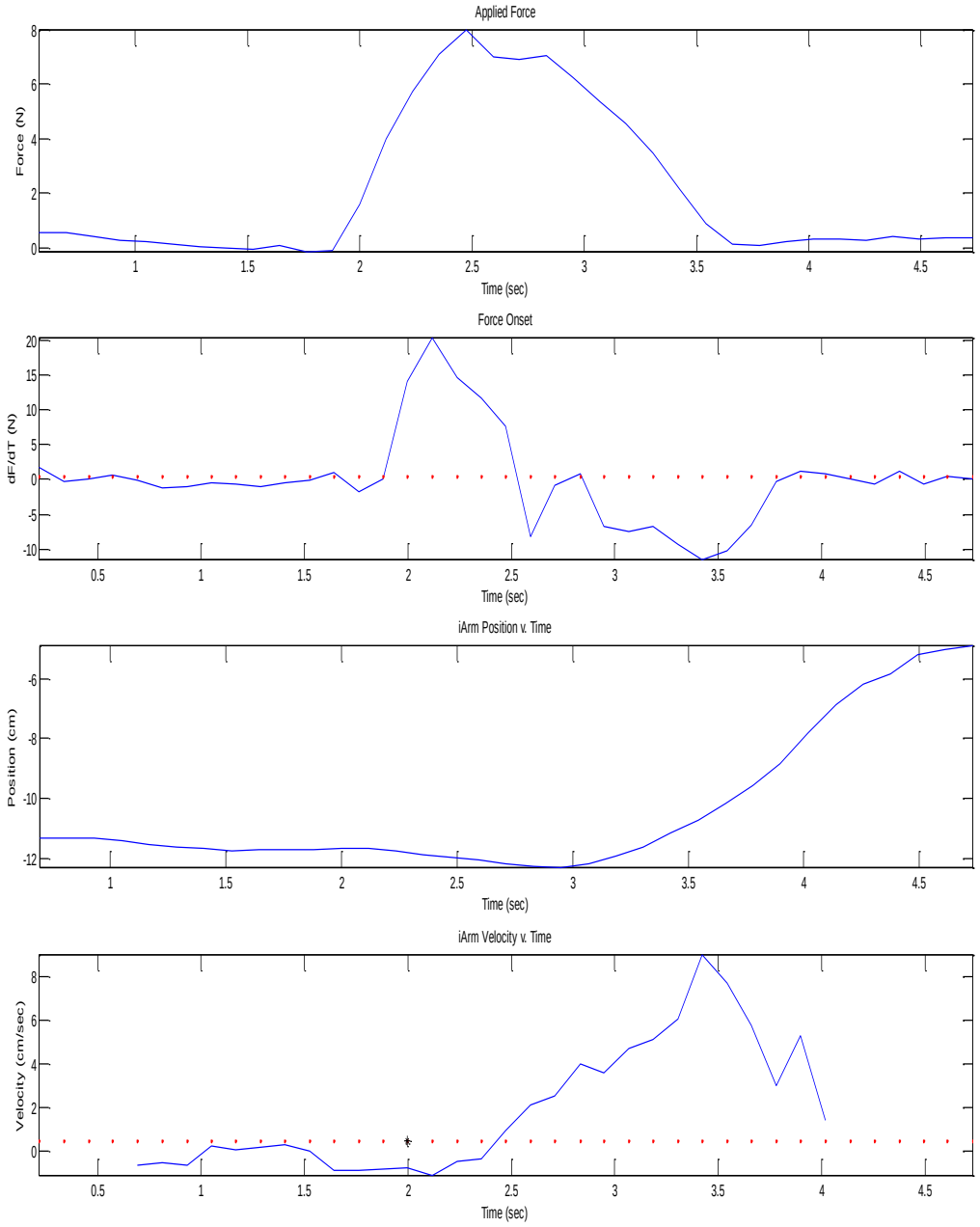

Figure 5.18 Applied force versus time (top), derived force versus time (second), iARM position versus time (third), and iARM velocity versus time in the z direction.

Table 5.12 shows the delay data with the implementation of admittance control for the ten trials for each of the three Cartesian directions. The average delay in the movement of the iArm was 0.440 seconds (with a standard deviation of 0.057), 0.453 seconds(with a standard deviation of 0.079), and 0.452 seconds (with a standard deviation of 0.048) for the x, y, and z directions, respectively.

Table 5.12 Admittance Control Delay Data

Admittance Control Delay			
Trial	**X**	**Y**	**Z**
1.000	0.475	0.475	0.475
2.000	0.474	0.354	0.473
3.000	0.476	0.475	0.474
4.000	0.360	0.473	0.357
5.000	0.475	0.613	0.476
6.000	0.474	0.473	0.477
7.000	0.352	0.357	0.476
8.000	0.480	0.474	0.474
9.000	0.360	0.476	0.366
10.000	0.477	0.356	0.475
Average	0.440	0.453	0.452
Standard Deviation	0.057	0.079	0.048

Table 5.13 shows the average time it took to run through the admittance control code once for a single sample. The average time it took to run the code was 0.119 seconds (with a standard deviation of approximately 0).

Table 5.13 MATLAB Code Run Time Data

Time to Run MATLAB Code	
Trial	**Delay (sec)**
1	0.119
2	0.119
3	0.119
4	0.119
5	0.119
6	0.119
7	0.119
8	0.118
9	0.119
10	0.119
Average	0.119
Standard Deviation	0.000

Table 5.14 shows the effective delay data when admittance control is implemented. According to this data, the average effective delay of the iArm when admittance control is implemented is 0.321 seconds (with a standard deviation of 0.057), 0.334 seconds (with a standard deviation of 0.079), and 0.334 seconds (with a standard deviation of 0.079).

Table 5.14 Effective Admittance Control Delay Data

Effective Admittance Control Delay			
Trial	**X**	**Y**	**Z**
1	0.356	0.356	0.356
2	0.355	0.235	0.235
3	0.357	0.356	0.356
4	0.241	0.354	0.354
5	0.356	0.494	0.494
6	0.355	0.354	0.354
7	0.233	0.238	0.238
8	0.361	0.355	0.355
9	0.241	0.357	0.357
10	0.358	0.237	0.237
Average	0.321	0.334	0.334
Standard Deviation	0.057	0.079	0.079

Table 5.15 shows the results of the Kolmogorov-Smirnov test on the effective delay values. The results of the Kolmogorov-Smirnov test were H values of 1 with corresponding p values of less than 0.05. This means that these sets of data are not normally distributed.

Table 5.15 Results of the Kolmogorov-Smirnov Test on Admittance Control Delay Data

	Effective Delay	
axis	**H Value**	**p Value**
x	1	0.001
y	1	0.001
z	1	0.001

Table 5.16 shows the results of the Mann-Whitney U test comparing the two sets of data (10 trials of effective delay and 10 trials of delay when sending single command using serial communication with firmware version 1.3). The Mann-Whitney U test returned H values of 0 and corresponding p values of greater than 0.05. According to these results, the null hypothesis that the compared data sets have a statistically equal mean is accepted. In other words, according to the Mann-Whitney U test, there is no statistical difference in the delay of the iArm when sending a single command compared to the effective delay of the iArm when admittance control is implemented.

Table 5.16 Results of the Mann-Whitney U Test on Effective Delay Data

Mann-Whitney U Test		
	H Value	**p Value**
X	0	1.000
Y	0	0.139
Z	0	0.14

5.3 HapticMASTER

5.3.1 HapticMASTER Delay

Figure 5.18 contains the two plots that demonstrate how the movement onset of the HapticMASTER end effector was determined. The top plot is the position of the HapticMASTER end effector versus time. The bottom plot is the change in position, or calculated velocity versus time. In this example, the time at which the command was send to the HapticMASTER (plotted as the black star) was 1.714 seconds. The threshold value (2% of the maximum velocity), was 1.158 cm/sec (plotted as the red line). The time of movement onset was determined to be 1.820 seconds. The difference between these two time values, or the delay of the HapticMASTER in this plotted trial was 0.106 seconds.

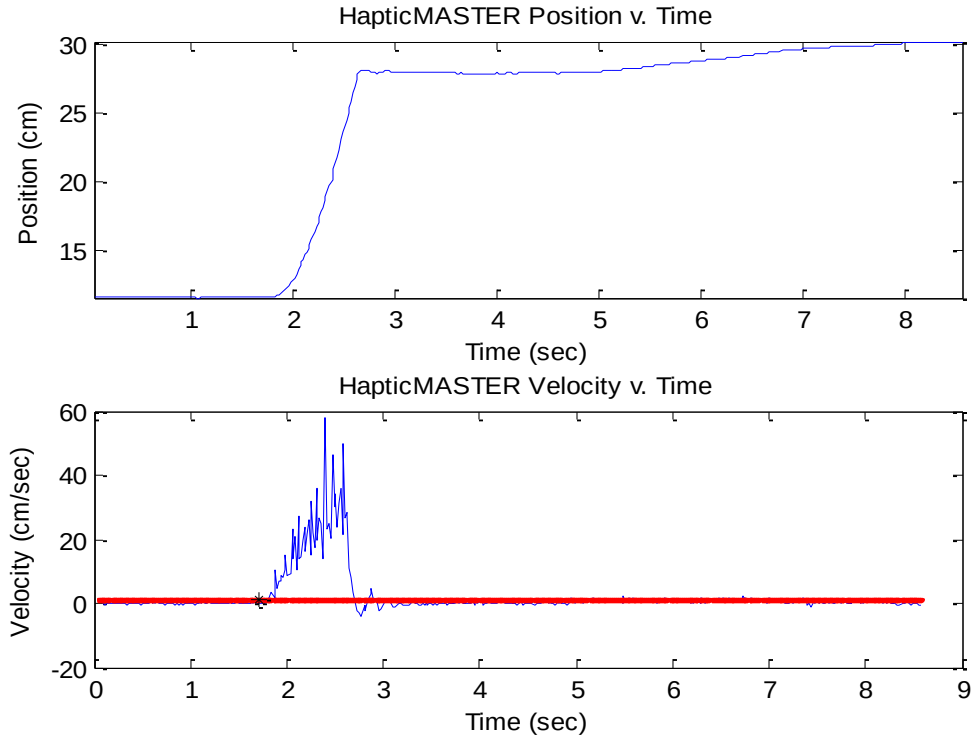

Figure 5.19 HapticMASTER position versus time (top) and HapticMASTER velocity versus time (bottom).

Table 5.17 shows the delay data collected for all ten trials for all three axes of the HapticMASTER. According to the data collected, the average delay of the HapticMASTER is 0.088 seconds (with a standard deviation of 0.019), 0.088 seconds (with a standard deviation of 0.014), and 0.077 seconds (with a standard deviation of 0.019) for the x, y, and z axis trials respectively. The average delay of the HapticMASTER is less than the ideal 0.10 second delay by 0.012 to 0.023 seconds.

Table 5.17 HapticMASTER Delay Data

HapticMASTER Delay			
Trial	**X**	**Y**	**Z**
1	0.074	0.098	0.095
2	0.076	0.091	0.077
3	0.111	0.079	0.080
4	0.092	0.107	0.045
5	0.091	0.092	0.093
6	0.075	0.076	0.059
7	0.059	0.065	0.075
8	0.118	0.093	0.061
9	0.106	0.105	0.076
10	0.076	0.077	0.107
Average	0.088	0.088	0.077
Standard Deviation	0.019	0.014	0.019

5.3.2 HapticMASTER as the Interface for the iARM

Figures 5.19 and 5.20 are plots of the iArm end effector position (plotted in red) and the HapticMASTER end effector position (plotted in blue) in the x, y, and z axis versus time. These figure provide a graphical representation of implementing the HapticMASTER as the controller of the iArm.

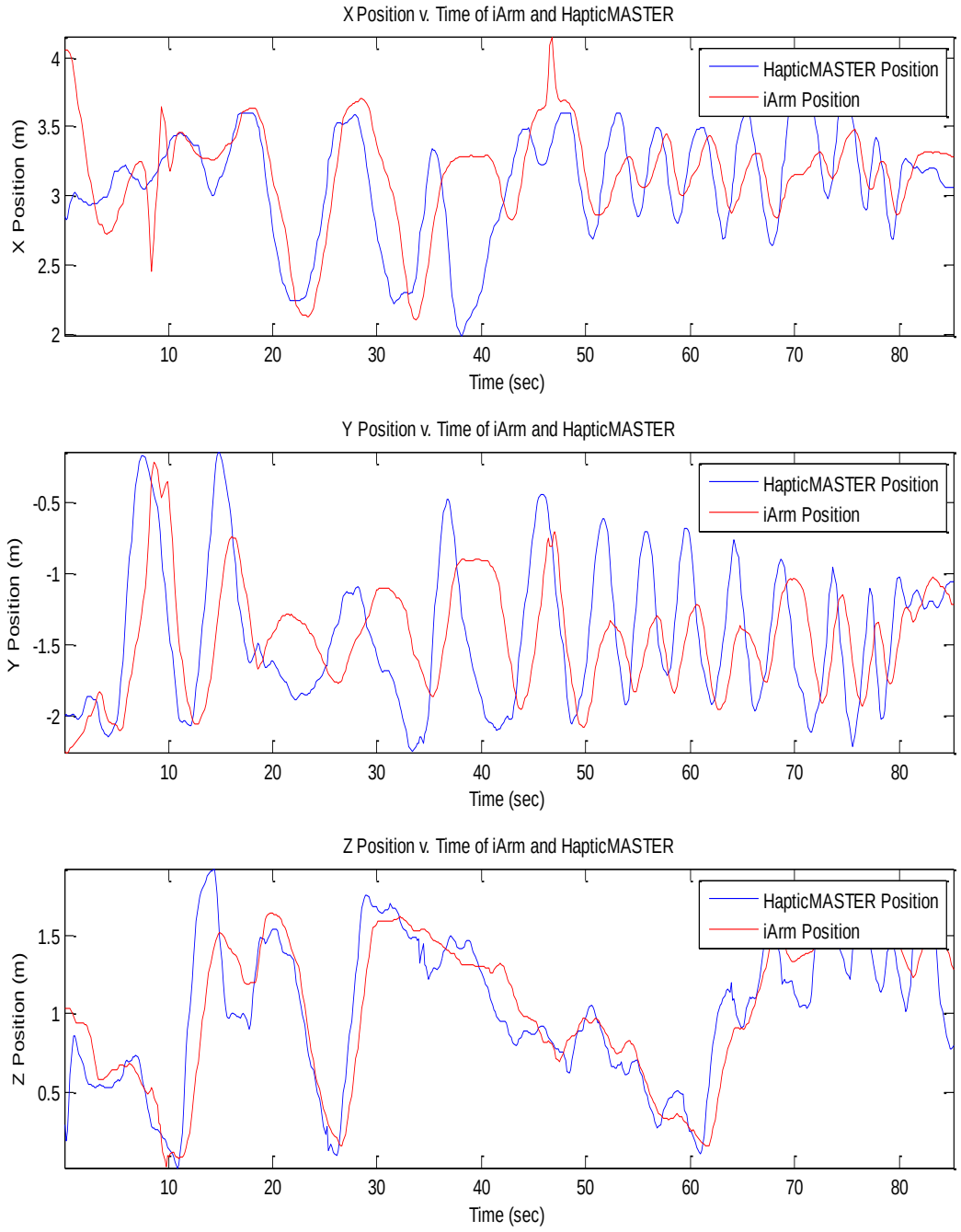

Figure 5.20 Position versus time of the iARM end effector (red) and HapticMASTER end effector (blue) versus time in the x axis (top), y axis (middle), and z axis (bottom) for trial 1.

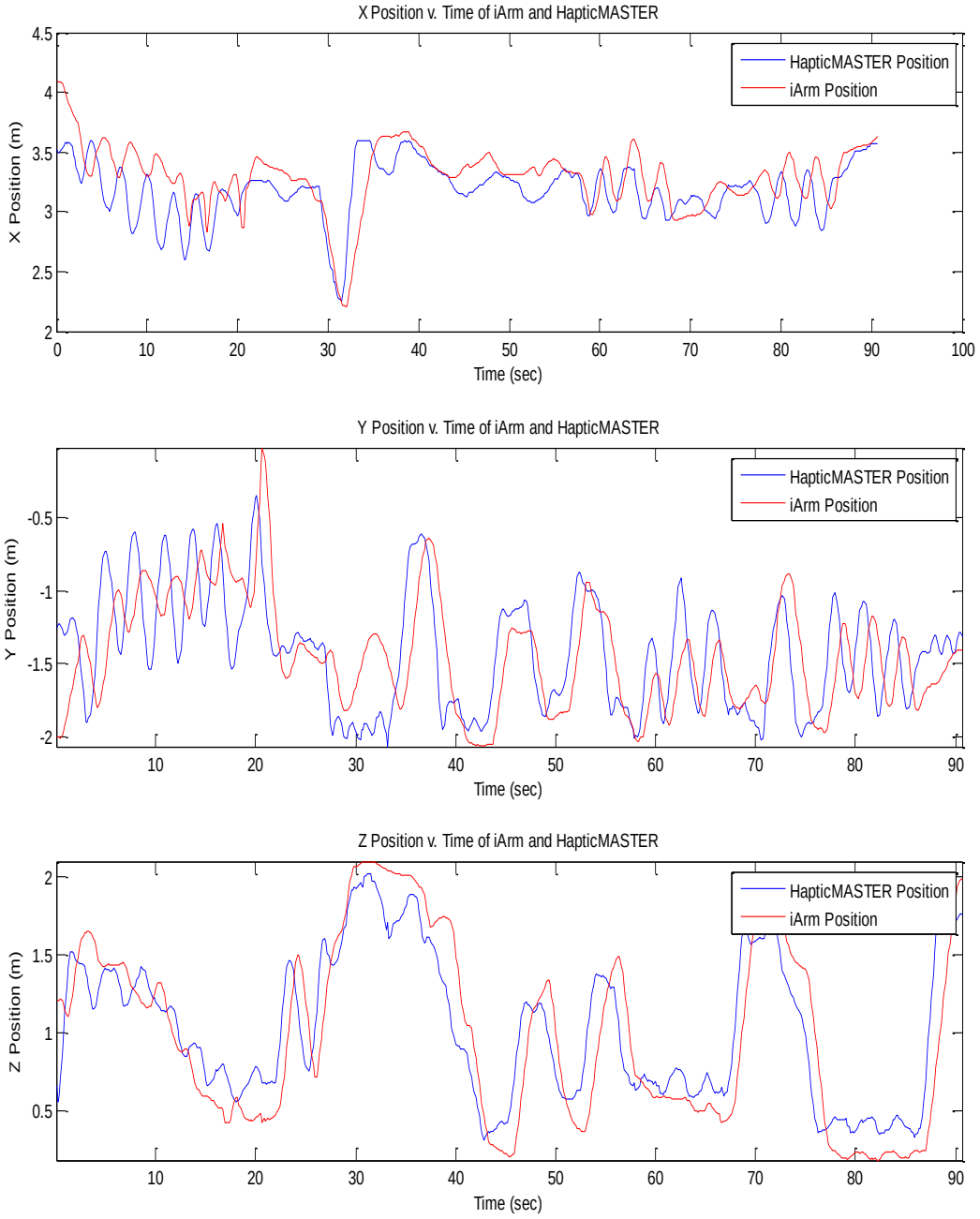

Figure 5.21 Position versus time of the iARM end effector (red) and HapticMASTER end effector (blue) versus time in the x axis (top), y axis (middle), and z axis (bottom) for trial 2.

CHAPTER 6

DISCUSSION AND FUTURE DIRECTION

The design requirements set fourth for the upper extremity assistive device for people with DMD were successfully met in the design of the prototype presented in this thesis. The proof of concept prototype demonstrates the use of an intuitive control scheme that can utilize the residual strength of individuals with DMD through the implementation of an admittance control interface, the success of which was demonstrated quantitatively. The use of admittance control, which prevents the need for user-specific interfaces, allowed for the capability of balancing the user's arm against gravity regardless of the weight of an object in the robot's gripper. Further, it prevents the need to compromise joint range of motion and workspace while eliminating the requirement of properly fitting and tuning the device. Implementing a control scheme with a preexisting robotic manipulator minimized the cost and complexity associated with the design of the device and implements a gripper eradicating the need of the user to retain muscle strength in order to pick up or interact with objects in their environment. Using the iARM allowed the device to be portable, light, compact, and require power that can be provided by a portable power source.

In order for the prototype to be comfortable and to assure optimal interaction between the human operator and the hardware the maximum delay of the device cannot exceed 100ms. The current prototype has a delay of about 4.5 times this optimal delay value. As was previously stated, the iARM motors contribute about 10ms of delay and the force sensor contributes about 6ms of delay. CAN communication is known for its

high-velocity transmission capabilities [41]. And because, according to the results, there is no statistical difference between the delay of the iARM when using CAN-Bus communication and serial communication it follows that the delay introduced as a result of the communication between the PC and the iARM is negligible. Additionally, the results showed that the inverse kinematic calculations in the iARM firmware did not contribute an assessable amount of delay. It was further demonstrated by the results that the delay introduced by the MATLAB code was about 0.12 seconds, with no additional delay being introduced aside from the runtime of the code. As a result, it is concluded that the remaining delay of approximately 0.31 seconds is a result of the iARM firmware. Furthermore, the combined delay of the MATLAB code and the iARM firmware needs to be reduced by about 35ms for the prototype to provide a successful human-machine interface.

In order to implement admittance control without the introduction of the significant delay introduced by MATLAB, future work will involve the use of MATLAB Compiler and Simulink Coder. MATLAB Compiler allows users to build standalone executables from MATLAB software that can be integrated into C or C++. Similarly, Simulink Coder generates C and C++ code from Simulink models [42, 43]. The ability to compile the admittance control MATLAB Code and Simulink model and run the code outside of the MATLAB environment reduces the application runtime. This reduction in runtime, which will directly result in a decreased delay, is realized through the comparison of computation time between MATLAB and C++. Andrews compared the computation time of MATLAB and C++ using computationally intensive pork chop plots. The results showed that C++ averages a processing speed that is over 500 times

faster than MATLAB [44]. Though the results of Andrew's study involved code that was more computationally intensive than the admittance control MATLAB Code, C++ will provide a substantial decrease in delay and allow for the potential of decreasing the total delay below 0.10 seconds.

In order to resolve the issue of the remaining contribution of delay, the iARM firmware, collaboration with Exact Dyanamics will be required. Knowing that the inverse kinematic calculations are not the cause of the firmware delay provides promise that the firmware code could be optimized without interfering with the calculations necessary to move the robot.

Future studies involving the HapticMASTER and the current device prototype will allow for further proof of concept. With the use of the HapticMASTER admittance control robot and the incorporation of gravity compensation, a study with subjects with DMD that have poor or nonexistent arm function will allow for quantifiable proof as to the success of admittance control as the ideal interface for this population. A further study, utilizing the HapticMASTER, will include subjects at different stages of functional loss of the arm, allowing for the assessment of the age and functional ranges that this technology can useful for. Additionally, future direction for the prototype will involve implementing admittance control for all six degrees of freedom of the iARM. And, assuming the delay of the iARM firmware can be decreased to be no greater than 100ms, a study involving subjects with DMD who have impaired arm function using an update prototype will allow for the investigation as to the device's successfulness as well as the assessment of future areas of improvement of the design.

APPENDIX A

MACHINE DRAWINGS

Figure A.1 to A.3 show the machine drawings for the force sensor mount and handle.

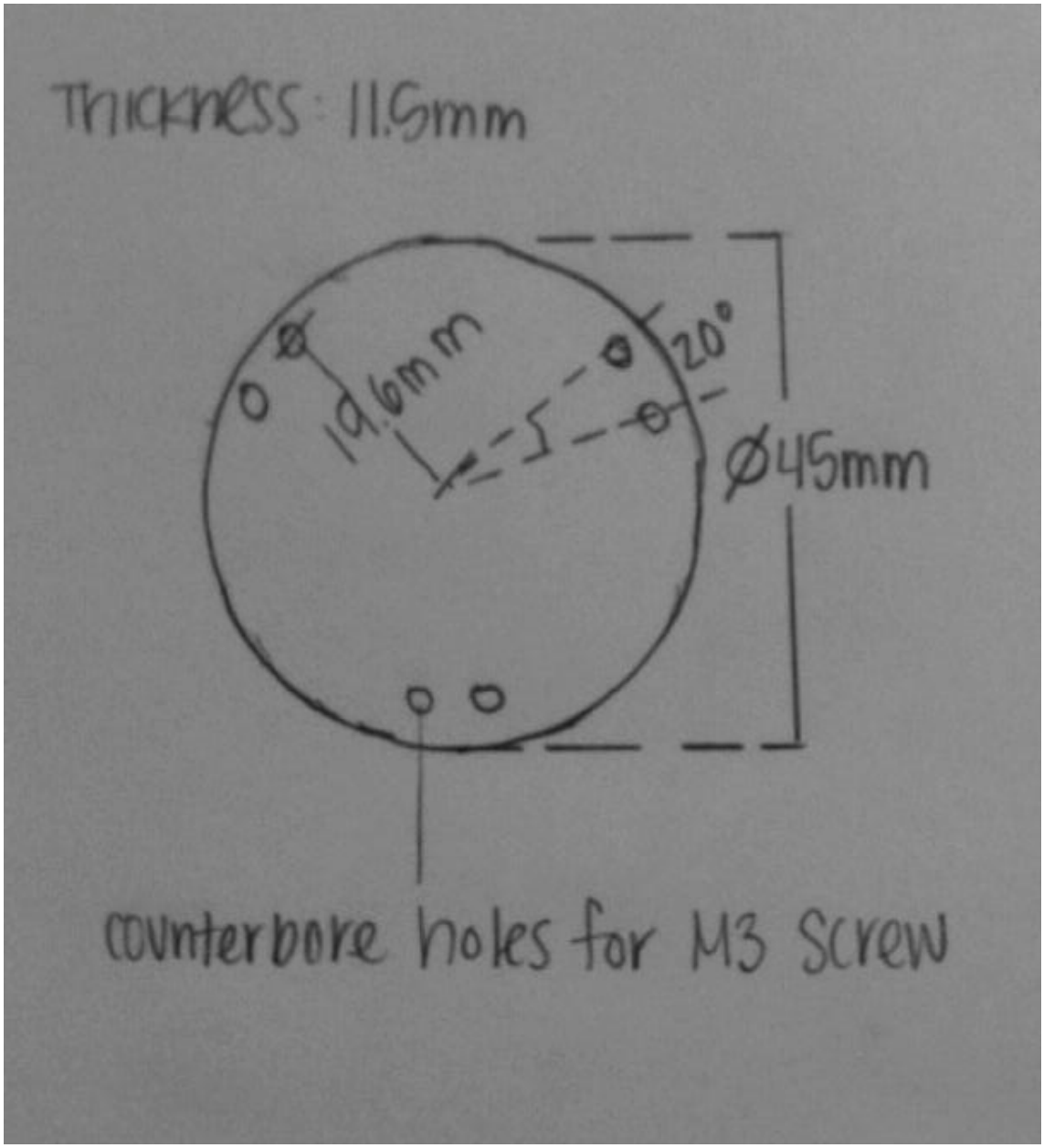

Figure A.1 Machine drawing of force sensor mounting plate - robot side.

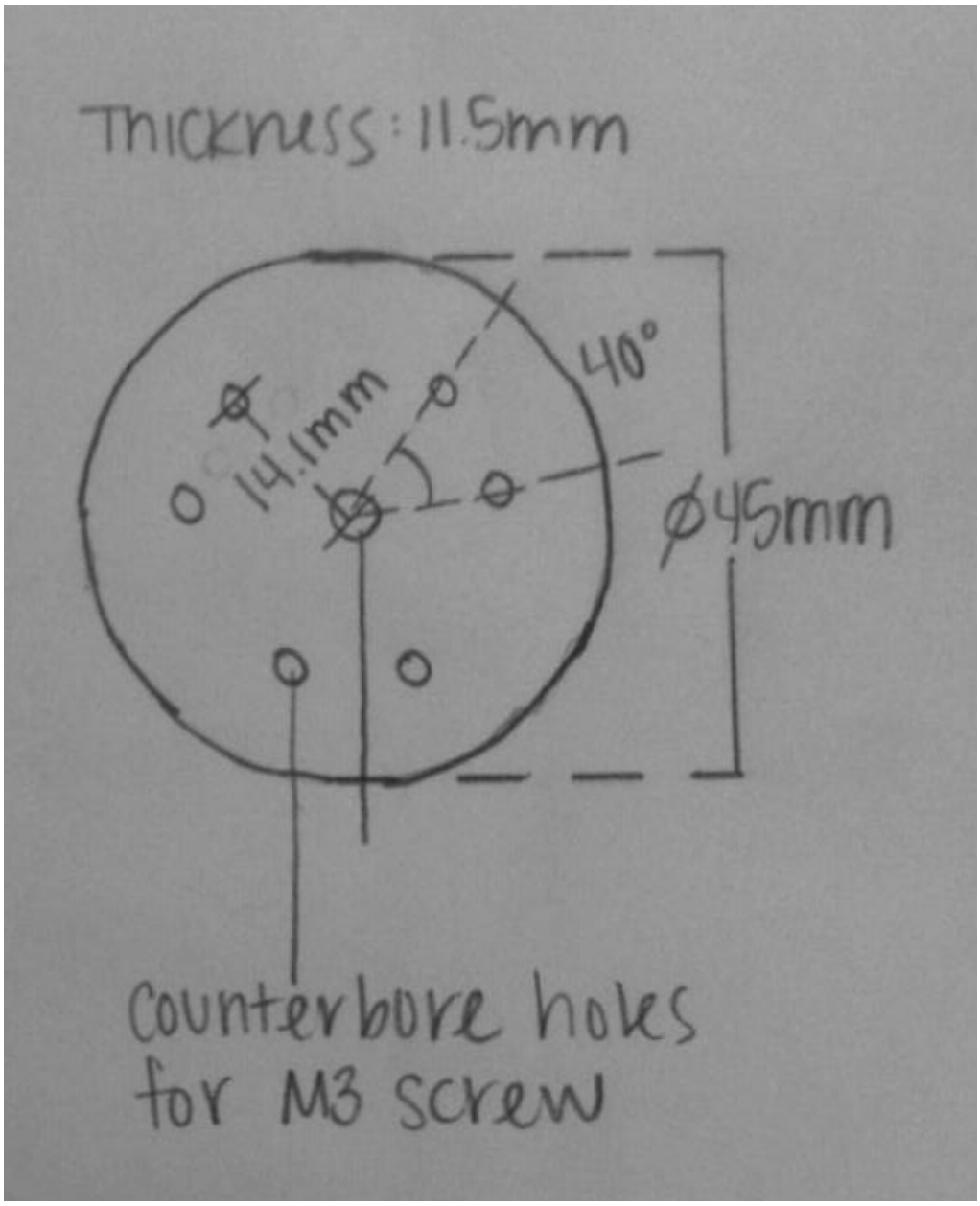

Figure A.2 Machine drawing of force sensor mounting plate – handle side.

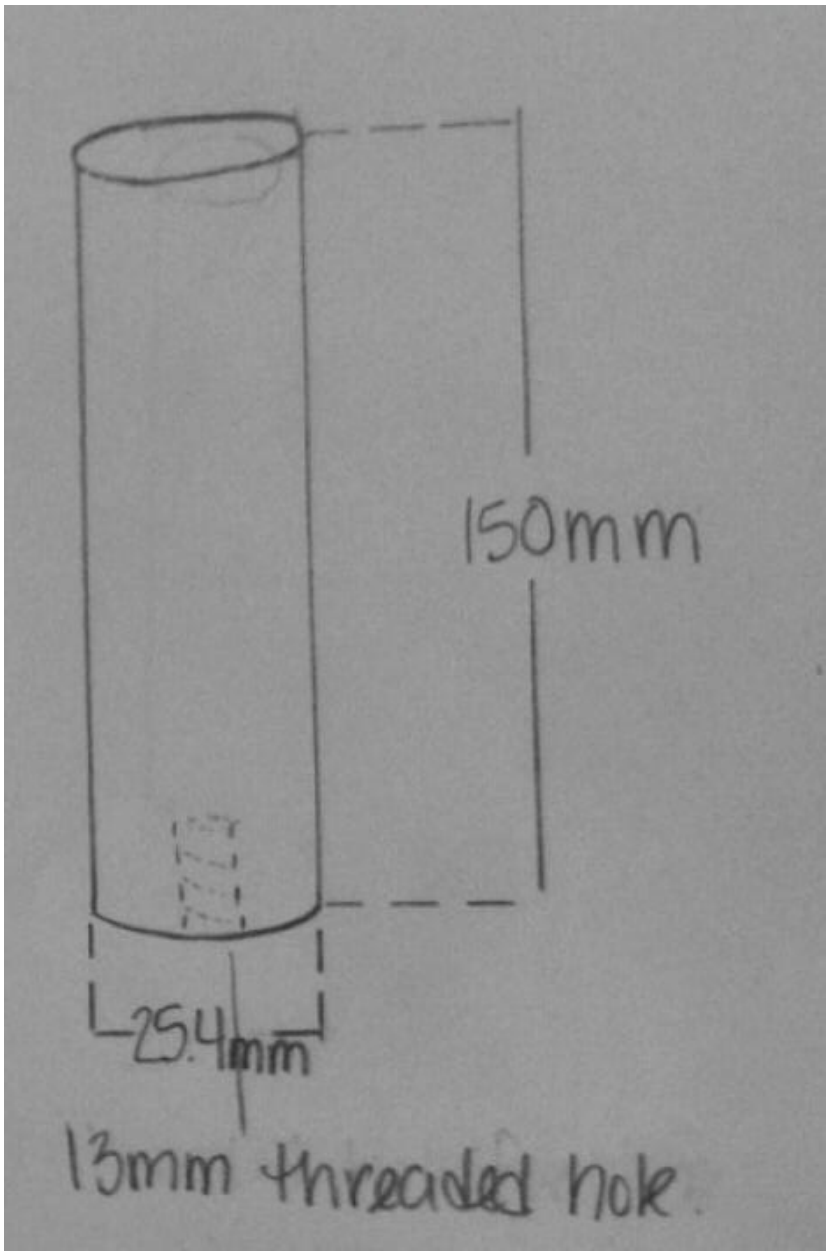

Figure A.3 Machine drawing of force sensor handle.

APPENDIX B

ADMITTANCE CONTROL MATLAB CODE

The following code is the MATLAB code used to implement admittance control.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% iArm Admittance Control
% 24 September 2012
% by Madeline Corrigan
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

%% Close FT and iArm Port
if exist ('A','var')==1
    fclose(A);
    delete(A);
    clear A
end

% if exist ('s','var')==1
%     fclose(s);
%     delete(s);
%     clear s
% end
%
% clear all
% close all
% clc
warning off all;

%% Open FT Port and Initialize
A =
serial('COM10','BaudRate',9600,'DataBits',8,'Parity','none','FlowContro
l','software','InputBufferSize',57);
fopen(A)

% Communication Setup
fprintf(A,'%s\r','CD A') % ASCII output
fprintf(A,'%s\r','CD R') % Resolved force/torque output
fprintf(A,'%s\r','SB') % Sensor bias
fprintf(A,'%s\r','CF 1') % Communication fast
fprintf(A,'%s\r','SF 100') % Frame rt

pause(.5)

% Purge Buffer
q=A.BytesAvailable;
pause(0.3)
```

```
if q>0
    junk=fread(A,q);
endpause(0.3)

% %% Initialize iARM
%
% % Serial Communication
% s =
serial('COM9','baudrate',115200,'databits',8,'parity','none','flowcontr
ol','hardware','stopbits',1);
% fopen(s)
% display('Please, turn on iARM');
% pause(35);
%
% run Unfold
% pause(5);
% display('iArm is initialized');
%
% display('Initialization complete.  Press any key to start data
collection.');
% pause

%% Ensure that Sensor is Biased
fprintf(A,'%s\r','QS'); % Start FT stream
flag = 0;
display('Please wait while sensor biases');

while flag == 0;

    for i = 1:20
        output = str2num(fscanf(A,'%s'));

        if size(output) == [1 7]
            test = output(2:4)/64; % Convert to N
            q=A.BytesAvailable;

            if q>0
                junk = fread(A,q);
            end
        end
    end

    T = mean(test,1);

    if abs(T(:,1)) < .05 && abs(T(:,2)) < .05 && abs(T(:,3)) < .05
        flag = 1;
        display('Sensor has been biased, data collection will begin');
    else
        fprintf(A,'%s\r','');
        fprintf(A,'%s\r','SB') % Sensor bias
        fprintf(A,'%s\r','QS'); % Start FT stream
    end

end
```

```
%% Start Data Collection

% Initialize Variables
Force = [];
n = 200000; % Number of samples
Vx = 0; Vy = 0; Vz = 0; % Initial velocity (for simulink)
mass = 0.2; % For simulink model
damping = 13;
forceprev = [0 0 0 0 0 0];
 i = 1;
 Fxstore = []; Fystore = []; Fzstore = [];

display('You may begin controlling the iArm');

for k = 1:10
    output = str2num(fscanf(A,'%s'));
end


tic
for j = 1:n
    q = A.BytesAvailable;
    if q == 57

    %Collect Force Sensor Data
    output = str2num(fscanf(A,'%s')); % Convert string to numeric
    if size(output) == [1 7]
    % Asssign Force Sensor Data to matrix 'force'
    Force(i,:) = output;
    force(i,1:3) = Force(i,2:4)/64; % 64 Counts/N
    force(i,4:6) = Force(i,5:7)/.3008; % 3008 Counts/Nmm

    Fx = force(i,1);
    Fy = force(i,2);
    Fz = force(i,3);

    % Run Simulink Model (Output: Velocity)
    sim('AdmittanceControl3DV.mdl') % 0.04 sec to run
    Vx = velx(2); Vy = vely(2); Vz = velz(2);

     Vxmm = -Vx*1000;
     Vymm = -Vy*1000; Vzmm = -Vz*1000;

     % Move iArm
     fprintf(s,'cv x#%d y#%d z#%d\r',[Vymm Vxmm Vzmm])

     VxStore(i) = Vxmm;
     VyStore(i) = Vymm;
     VzStore(i) = Vzmm;

     Fxstore(i) = Fx;
     Fystore(i) = Fy;
     Fzstore(i) = Fz;
```

```
     t(i) = toc;

    end
    if q > 0
    junk = fread(A,q);
    end
    i = i + 1;
    end


end

% Stop Data Stream
fprintf(A,'%s\r','')
```

APPENDIX C

IARM/HAPTIC MASTER MATLAB CODE

The following code is the MATLAB code used to implement the HapticMASTER as the controller for the iARM.

```
%Initialize iARM

% s =
serial('COM5','baudrate',115200,'databits',8,'parity','none','flowcontr
ol','hardware','stopbits',1);
% fopen(s)
% display('Please, turn on iARM');
% pause(30);
%
% display('You may now begin controlling the iARM');

%% Clean Up
%HM_cleanUp
flag = 0; % gripper closed
Fs = 11025;

fprintf(s,'%s\r','u');
pause(10)
%% Initialize Haptic Master

disp('Initializing Haptic')

HM_CleanUp


bb=Test_Haptic(4);

disp('Haptic Ready')

time  = 1000; %In miliseconds
input = 15;
reset = 0;
cs=0;
disp('Press any key to start the data collection')
pause
[position force timestamp port]=HM_dataThread(time,input,reset);
%%

position = [0,0,.25];
damping = 0;
stiffness = 40;
```

```
db = 0;
direction= [ 0 0 1];
spring=HM_create_spring(position,stiffness,damping,db,direction);

disp('Created Assistance')
pause
HM_CleanUp


for i=1:1%runs for 100 frames.  CHANGE THIS TO RUN LONGER

    HMposition(i,:)=position;
    HMforce(i,:)=force;
    HMtimestamp(i,:)=timestamp;

    x=[0 0 HMforce(i,3)]

    test_raf(-1*x) %constant force function


    pause(.01)

end



%% Define Haptic Constants

time  = 100000;
input = 15;
reset = 0;
[position force timestamp port]=HM_dataThread(time,input,reset);
%% Constants for Blocks


%%


for i= 1:1000

    HMposition(i,:)=position;

    sp=10*HMposition(i,:); %times 10

   Xmm = sp(:,1)*100
   Ymm = sp(:,2)*100
   Zmm = sp(:,3)*100
    fprintf(s,'cp x#%d y#%d z#%d\r',[Xmm Ymm Zmm]);
     pause(.1)

end
```

REFERENCES

[1] I. Y. Jung, J. H. Chae, S. K. Park, J. H. Kim, J. Y. Kim, S. J. Kim, M. S. Bang, "The Correlation Analysis of Functional Factors and Age with Duchenne Muscular Dystrophy," *Ann Rehabil Med,* vol. 36. pp. 22-32, 2012.

[2] B. Bartels, R. F. Pangelila, M. P. Bergen, N. A. M. Cobben, H. J. Stam, M. E. Roebroeck, "Upper Limb Function in Adults with Duchenne Muscular Dystrophy," *J Rehabil Med*, vol. 43. pp. 770-775, 2011.

[3] C. D. Markert, F. Ambrosio, J. A. Call, R. W. Grange, "Exercise and Duchenne Muscular Dystrophy: Toward Evidence-Based Exercise Prescription," *Muscle Nerve*, vol. 43. pp. 464-478, 2001.

[4] M. Kohler, C. F. Clarenbach, L. Boni, T. Brack, E. W. Russi, K. E. Bloch, "Quality of Life, Physical Disability, and Respiratory Impairment in Duchenne Muscular Dystrophy," *Am J Respir Crit Care Med,* vol. 172, pp. 1032-1036, 2005.

[5] E. S. Mazzone, G. Vasco, C. Palermo, F. Bianco, C. Galluccio, V. Ricotti, A. D. Castronovo, M. S. DiMAuro, M. Pane, A. Mayhew, E. Mercuri, "A critical review of functional assessment tools for upper limbs in Duchenne muscular dystrophy," *Developmental Medicine & Child Neurology*, vol. 54, pp. 879-885, 2012.

[6] B. F. Steffensen, S. Lyager, B. Werge, J. Rahbek, E. Mattsson, "Physical capacity in non-ambulatory people with Duchenne muscular dystrophy or spinal muscular atrophy: a longitudinal study," *Developmental Medicine & Child Neurology,* vol. 44, pp. 623-632, 2002.

[7] B. E. Becker, "Aquatic Therapy: Scientific Foundations and Clinical Rehabilitation Application," *Americal Academy of Physical Medicine and Rehabilitation,* vol. 1, pp. 859-872, 2009.

[8] M. Fragala-Pinkham, S. M. Haley, M. E. O'Neil, "Group aquatic aerobic exercise for children with disabilities," *Developmental Medicine & Child Neurology*, vol. 50, pp. 822-827, 2008.

[9] K. R. Wagner, N. Lechtzin, D. P. Judge, "Current Treatment of Adult Duchenne Muscular Dystrophy," *BBA – Molecular Basis of Disease*, 2011.

[10] M. Getz, Y. Hutzler, A. Vermeer, "Effects of aquatic interventions in children with neuromotor impairments: a systematic review of the literature," *Clinical Rehabilitation*, vol. 20, pp. 927-936, 2006.

[11] K. Bushby, R. Finkel, D. J. Birnkrant, L. E. Case, P. R. Clemens, L. Cripe, A. Kual, K. Kinnett, C. McDonald, S. Pandya, J. Poysky, F. Shapiro, J. Tomezsko, C. Constantin, "Diagnosis and management of Duchenne muscular dystrophy, part 2: implementation of multidisciplinary care," *The Lancet Neurology*, 2009.

[12] E. H. Cup, A. J. Pieterse, J. M. ten Broek-Pastoor, M. Munneke, B. G. van Engelen, H. T. Hendricks, G. J. van der Wilt, R. A. Oostendorp, "Exercise Therapy and Other Types of Physical Therapy for Patients With Neuromuscular Diseases: A Systematic Review," *Arch Phys Med Rehabil*, vol. 88, pp. 1452-1464, 2007.

[13] C. D. Markert, L. E. Case, G. T. Carter, P. A. Furlong, R. W. Grange, "Exercise and Duchenne Muscular Dystrophy: Where We Have Been and Where We Need to Go," *Muscle & Nerve*, pp. 746-751, 2012.

[14] M. Jansen, I. JM de Groot, N. vanAlfen, A. C. H. Geurts, "Physical training in boys with Duchenne Muscular Dystrophy: the protocol of the No Use is Disuse study," *BMC Pediatrics*, vol. 10. no. 55, 2010.

[15] M. Eagle, "Report on the Muscular Dystrophy Campaign workshop: Exercise in neuromuscular diseases," *Neuromuscular Disorders*, vol. 12, pp. 975-983, 2002.

[16] R. Ramanatha, S. P. Eberhardt, T. Rahman, W. Sample, R. Seliktar, M. Alexander, "Analysis of Arm Trajectories of Everyday Tasks for the Development of an Upper-Limb Orthosis," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 1, pp. 60-70. 2000.

[17] N. Pellegrini, B. Guillon, H. Prigent, M. Pellegrini, D. Orlikovski, J. C. Raphael, F. Lofaso, "Optimization of power wheelchair control for patients with severe Duchenne muscular dystrophy," *Neuromuscular Disorders*, vol. 14, pp. 297-300, 2004.

[18] C. A. Stanger, C. Anglin, W. S. Harwin, D. P. Romilly, "Devices for Assisting Manipulation: A Summary of User Task Priorities," *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 4, pp. 256-265, 1994.

[19] N. Pernalete, W. Yu, R. Dubey, W. Moreno, "Development of a Robotic Haptic Interface to Assist the Performance of Vocational Tasks by People with Disabilities," in *IEEE International Conference on Robotics & Automation*, Washington, DC, 2002, pp. 1269-1274.

[20] R. Rahman, W. Sample, R. Seliktar, M. Alexander, M. Scavina, “A body-powered functional upper limb orthosis,” *Journal of Rehabilitation Research and Development*, vol. 37. no. 6, 2000.

[21] F. Casolo, S. Cinquemani, M. Cocetta, “A Passive Support To Motion Capability Of Subjects Affected By Neuromuscular Diseases,” in *World Congress on Engineering*, London, UK, 2008.

[22] J. L. Herder, “Development of a Statically Balanced Arm Support: ARMON,” in *IEEE 9th International Conference on Rehabilitation Robotics*, Chicago, IL, 2005, pp. 281-286.

[23] J. L. Herder, N. Vrijlandt, T. Antonides, M. Cloosterman, P. L. Mastenbrock, “Principle and design of a mobile arm support for people with muscular weakness,” *Journal of Rehabilitation Research & Development*, vol. 43, no. 5, pp. 591-604, 2006.

[24] S. Landsberger, P. Leung, V. Vargas, J. Shaperman, J. Baumgarten, Y. L. Yasuda, E. Sumi, D. McNeal, R. Waters, “Mobile Arm Supports: History, Application, and Work in Progress”, *Top Spinal Cord Inj Rehabil*, vol. 11, no. 2, pp. 74-94, 2005.

[25] R. Daniel, T. Rahman, W. Sample, S. Agrawal, “Dynamic Simulation and Experimental Validation of an Upper Extremity Powered Orthosis,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Montreal, Canada, 2010, pp. 1-6.

[26] R. Rahman, W. Sample, R. Seliktar, M. T. Scavina, A. L. Clark, K. Moran, M. A. Alexander, “Design and Testing of a Functional Arm Orthosis in Patients with Neuromuscular Diseases,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 2, pp. 244-251, 2007.

[27] D. P. Romilly, C. Anglin, R. G. Gosine, C. Hershler, S. U. Raschke, “A Functional Task Analysis and Motion Simulation for the Develop[ment of a Powered Upper-Limb Orthosis,” *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 3, pp. 119-129, 1994.

[28] H. Martin, S. Chevallier, E. Monacelli, “Fast calibration of hand movement-based interface for arm exoskeleton control,” in *ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, 2012, pp. 25-27.

[29] N. Tobias, R. Robert, "ARMin – Design of a Novel Arm Rehabilitation Robot," in *IEEE 9th International Conference on Rehabilitation Robotics*, Chicago, IL, 2005, pp. 57-71.

[30] S. K. Agrawal, V. N. Dubey, J. J. Gangloff, E. Brackbill, V. Sangwan, "Optimization and Design of a Cable Driven Upper Arm Exoskeleton," in *ASME 2009 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE,* San Diego, CA, 2009, pp. 1-8.

[31] N. G. Tsagarakis, D.G. Caldwell, "Development and Control of a 'Soft-Actuated' Exoskeleton for Use in Physiotherapy and Training," *Autonomous Robots*, vol. 15, pp. 21-33, 2003.

[32] Y. Wakita, N. Yamanobe, K. Nagata, N. Ando, M. Clerc, "Development of User Interface with Single Switch Scanning for Robot Arm to Help Disabled People Using RT-Middleware," in *10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, 2008, pp. 1515-1520.

[33] *iARM: Intelligent Arm Robot Manipulator*, Assistive Innovations, Didam, NL. [Online]. Available: http://www.assistive-innovations.com/index.php/en/mediaitems/downloads

[34] M. Shramowiat, J. R. Bach, C. Bocobo, "Functional Enhancement of Patients with Duchenne Muscular Dystrophy with the Use of Robot-Manipulator Trainer Arms," *J Neuro Rehab*, vol. 3, pp. 129-132, 1989.

[35] J. R. Bach, A. P. Zeelenberg, C. Winter, "Wheelchair-Mounted Robot Manipulators," *American Journal of Physical Medicine & Rehabilitation,* vol. 69, no. 2, pp. 55-59, 1990.

[36] N. Didi, M. Mokhtari, A. Roby-Brami, "Preprogramed Gestures for Robotic Manipulators: An Alternative to Speed Up Task Execution Using Manus," in *International Conference on Rehabilitation Robotics*, Stanford, CA, 1999, pp. 92-98.

[37] B. Hannaford, J. H. Ryu, "Time-Domain Passivity Control of Haptic Interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, 2002.

[38] *F/T Controller Instillation and Operation Manual*, ATI Industrial Automation, Apex, NC, pp.28-72, 2004.

[39] G. C. Burdea, "Haptic Sensing and Control," in *Force and touch Feedback for Virtual Reality*, New York, NY: Wiley, 1996, ch. 2, sec. 3-5, pp. 34-39.

[40] N. Teasdale, C. Bard, M. Fleury, D. E. Young, L. Proteau, "Determining Movement Onsets From Temporal Series," *Journal of Motor Behavior*, vol. 25, no. 2, pp. 97-106, 1992.

[41] P. Ran, B. Wang, W. Wang, "The Design of Communication Convertor Based on CAN Bus", unpublished. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4608607

[42] MathWorks, *MATLAB Compiler*. [Online]. Available: http://www.mathworks.com/products/datasheets/pdf/matlab-compiler.pdf

[43] MathWorks, *Simulink Coder*. [Online]. Available: http://www.mathworks.com/products/datasheets/pdf/simulink-coder.pdf

[44] T. Andrews, "Computation Time Comparison Between Matlab and C++ Using launch Windows", unpublished. [Online]. Available: http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1080&context=aerosp

[45] R. Q. Van der Linde, P. Lammertse, E. Frederiksen, B. Ruiter, "The HapticMaster, a new high-performance haptic interface," unpublished. [Online]. Available: http://www.eurohaptics.vision.ee.ethz.ch/2002/notused/thefile.pdf

[46] D. Ragonesi, S. Agrawal, W. Sample, T. Rahman, "Quantifying Anti-Gravity Torques in the Design of a Powered Exoskeleton," in *33rd Annual International Conference of the IEEE EMBS*, Boston, MA, 2011, pp. 7458-7461.

[47] K. Uchikawa, M. Liu, K. Hanayama, T. Tsuji, T. Fujiwara, N. Chino, "Functional Status and Muscle Strength in People with Duchenne Muscular Dystrophy Living in the Community," *J Rehabil Med*, vol. 36, pp. 124-129, 2004.