

A Robust Wheel Interface With A Novel Adaptive Controller For Computer/robot-Assisted Motivating Rehabilitation

Andrew R. Theriault
Marquette University

Recommended Citation

Theriault, Andrew R., "A Robust Wheel Interface With A Novel Adaptive Controller For Computer/robot-Assisted Motivating Rehabilitation" (2013). *Master's Theses (2009 -)*. Paper 206.
http://epublications.marquette.edu/theses_open/206

A ROBUST WHEEL INTERFACE WITH A NOVEL ADAPTIVE CONTROLLER FOR
COMPUTER/ROBOT-ASSISTED MOTIVATING REHABILITATION

by

Andrew R. Theriault, B. S.

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

May 2013

ABSTRACT
A ROBUST WHEEL INTERFACE WITH A NOVEL ADAPTIVE CONTROLLER FOR
COMPUTER/ROBOT-ASSISTED MOTIVATING REHABILITATION

Andrew R. Theriault, B. S.

Marquette University, May 2013

TheraDrive is a low-cost robotic system for post-stroke upper extremity rehabilitation. This system uses off-the-shelf computer gaming wheels with force feedback to help reduce motor impairment and improve function in the arms of stroke survivors. Preliminary results show that the TheraDrive system lacks a robust mechanical linkage that can withstand the forces exerted by patients, lacks a patient-specific adaptive controller to deliver personalized therapy, and is not capable of delivering effective therapy to severely low-functioning patients.

A new low-cost, high-force haptic robot with a single degree of freedom has been developed to address these concerns. The resulting TheraDrive consists of an actuated hand crank with a compliant transmission. Actuation is provided by a brushed DC motor, geared to output up to 50 lbf (223 N) at the end effector. To enable safe human-machine interaction, a special compliant element was developed to function also as a failsafe torque limiter. A load cell is used to determine the human-machine interaction forces for use by the robot's impedance controller. The impedance controller renders a virtual spring that attracts or repels the end effector from a moving target that the human must track during therapy exercises. As exercises are performed, an adaptive controller monitors patient performance and adjusts the spring stiffness to ensure that exercises are difficult but doable, which is important for maintaining patient motivation. Experiments with a computer model of a human and robot show the adaptive controller's ability to maintain difficulty of exercises after a period of initial calibration.

Seven human subjects (3 normal, 4 stroke-impaired) were used to test this system alongside the original TheraDrive system in order to compare both systems. Data showed that the new system produced a larger change in normalized trajectory tracking error when assistance/resistance was added to exercises when compared to the original TheraDrive. Data also showed that adaptive control led subject performance to be closer to a desired level. Motivation surveys showed no significant difference in subject motivation between the two systems. When asked to choose a preferred system, stroke subjects unanimously chose the new robot.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Overview	1
1.2	Background on Stroke	1
1.3	Stroke Rehabilitation Robots	2
1.4	Low-Cost Systems	4
1.5	TheraDrive Background	5
1.6	Concurrent Research	7
1.7	Problem Definition	7
1.8	Specific Aims	9
CHAPTER 2	DESIGN OF SYSTEM	10
2.1	Design Goals	10
2.2	Mechanical Design	10
2.2.1	Motor and Gearing	10
2.2.2	Compliant Actuators	12
2.2.3	Compliant Torque Limiter	13
2.2.4	Load Cell	17
2.3	Electrical Design	20
2.3.1	Servo Amplifier	20
2.3.2	Bridge Amplifier	20
2.3.3	Encoder	21
2.4	Software Design	21
2.4.1	Force Feedback	21
2.4.2	Adaptive Control Strategies	22
2.4.3	Control Design	23
2.4.4	Task Software	25
2.5	Discussion and Conclusion	26
CHAPTER 3	SIMULATION EXPERIMENTS	28
3.1	Introduction	28
3.2	Simulation Design	28
3.3	Experiment Design and Hypotheses	30
3.4	Simulation Results	31

3.5	Discussion and Conclusion	35
CHAPTER 4	HUMAN SUBJECT EXPERIMENTS	37
4.1	Introduction	37
4.2	Data Collection Procedure	37
4.3	Old versus New TheraDrive Systems	39
4.3.1	Procedure	39
4.3.2	Results	40
4.3.3	Analysis and Discussion	42
4.4	Nonadaptive versus Adaptive Control	43
4.4.1	Procedure	43
4.4.2	Results	48
4.4.3	Discussion and Conclusion	50
4.5	Comparison of Real and Simulated Subjects	51
CHAPTER 5	CONCLUSION AND FUTURE DIRECTIONS	53
5.1	General Conclusion	53
5.2	Contributions	53
5.3	Future Directions	54
	BIBLIOGRAPHY	55
	APPENDIX A EXPERIMENTAL PROCEDURE	57
A.1	Procedure	57
A.2	Motivation Survey	59
A.3	Post-Activity Survey	61
A.4	Final Survey	62
	APPENDIX B RAW EXPERIMENTAL DATA	63
	APPENDIX C BILL OF MATERIALS	66
	APPENDIX D MECHANICAL DRAWINGS	67
D.1	Drawings	67
D.2	Assembly	83
	APPENDIX E SOURCE CODE	84
E.1	Simulation Model	84
E.2	Real-Time Robot Controller	86
E.3	Adaptive Controller	87
E.4	Subject GUI	89

E.5	Data Analysis	92
E.6	Miscellaneous Scripts	94
APPENDIX F	LOAD CELL	95
F.1	Amplifier Schematic	95
F.2	Calibration Data	96

CHAPTER 1 INTRODUCTION

1.1 Overview

This project encompasses the design and testing of a single-degree-of-freedom haptic robot for post-stroke arm rehabilitation. The robot is designed to overcome many of the weaknesses of its predecessor, the TheraDrive system. TheraDrive uses Logitech steering wheel joysticks as a haptic interface for rehabilitation. It showed successes in a pilot study; however, the joysticks cannot withstand the rigors of use.

Specific aims of this project are to:

1. Develop a device that functions as a drop-in replacement for the Logitech wheel and is able to exert larger forces on the patient.
2. Extend the functionality of the TheraDrive system to accommodate low-functioning patients.
3. Carry out a preliminary study to assess the performance and usability of the new device.

1.2 Background on Stroke

With the increasing portion of elderly people in the population, stroke is becoming increasingly prevalent as a cause of death or disability [1]. The number of elderly in the U.S. is expected to nearly double by the middle of the century if current trends continue. Additionally, modern medicine improves the odds of survival from stroke, and future improvements in care and treatment will likely increase the odds further. These factors combined mean that there is a rapidly increasing number of stroke survivors who will be left to deal with the aftermath of their strokes.

Stroke is an interruption of blood flow to the brain, which rapidly causes nerve cell damage or death. It is usually brought about by an embolism or by a blood vessel rupture [1]. The brain damage from a stroke frequently causes loss of motor coordination or hemiparesis as well as some cognitive impairments. Loss of motor function is the most debilitating effect of stroke. It greatly impairs a stroke survivor's ability to perform activities of daily living. Activities such as self-feeding and dressing become difficult, and gait is often impaired as well. Cognitive impairments resulting from stroke can include speech impairment, loss of sensation in affected limbs, and inability to process sensory data.



Figure 1.1: MIT-MANUS robot with patient, from [2].

After stroke, patients must undergo physical therapy to regain lost motor function. Traditionally, this therapy has been performed manually by a physical therapist, but robotic systems, with their ability to measure and exert motion accurately, are becoming more common.

1.3 Stroke Rehabilitation Robots

MIT-Manus was the first arm rehabilitation robot (circa 1988) and is now commercially available from Interactive Motion Technologies [2]. In its original form, it was an end-effector robot composed of a planar five-bar linkage with two degrees of freedom (Figure 1.1). Being a parallel linkage, it has less position error than a serial linkage, but it has more complex dynamics. Manus uses direct-drive motors to actuate this linkage, and these motors must be very large to exert sufficient joint torques. More recently, additions have been made to the system, including a vertical motion component [3] and a set of actuators for hand/wrist rehabilitation.

Over the past two decades, MIT-Manus has been used successfully to treat over 300 stroke subjects, as reported in various studies [2]. A pilot study to determine the effectiveness of the vertical motion addition was carried out on ten subjects [3]. The study followed subjects through twelve weeks of therapy split into two six-week segments, with each week containing three one-hour training sessions for a total of 36 training sessions. For the first six-week segment, subjects performed planar therapy exercises using the original configuration of MIT-Manus. Therapy sessions consisted of point-to-point movement tasks where the patient started with the arm in a home position and reached to a given point, returning to home after each repetition. After the first six weeks, patients' motor performance was evaluated and recorded, and then patients began the

second six-week segment of training. The second six weeks of training point-to-point functional reaching tasks and vertical elevation tasks. At the end of training, patients’ motor performance was evaluated again. Compared to baseline performance before therapy, patients showed significant improvement after the first six weeks of therapy. Improvement continued to a lesser degree through the second six weeks, but it was not statistically significant.

ARMin is an arm rehabilitation robot developed at ETH Zurich with six degrees of freedom—four of which are actuated—for the elbow and shoulder joints [4, 5]. Its design is a hybrid between an exoskeletal robot and an end-effector robot, with an end-effector type linkage actuating the shoulder and an exoskeleton actuating the elbow and forearm. This design allows the shoulder joint to move naturally instead of constraining it to the axes of an exoskeletal joint. It is advantageous over a system like MIT-Manus because it allows each joint of the arm to be controlled independently rather than controlling only the end of the arm. However, this comes with the disadvantage of increased cost and complexity. The robot is fixed to a vertical rail on a wall behind the patient and to the patient’s elbow and forearm. By moving up and down the rail, the robot lifts the upper arm to rotate the shoulder about the frontal axis. Other joint rotations are accomplished by motors aligned with the respective joint axes.

Driver’s SEAT, developed by M. Johnson et al., introduces a driving simulator interface to rehabilitation robots [6]. It uses a force-feedback steering wheel with a force sensor for each hand. Having separate sensors for each hand allows the robot to respond differently to force input from each hand. In this case, the robot resists forces applied by a stroke patient’s unimpaired arm, forcing the patient to use the impaired arm to assist in the completion of steering tasks. This is done to counteract the phenomenon of learned non-use, where stroke survivors stop using the impaired arm, favoring the unimpaired arm instead. A study using eight stroke subjects was performed to evaluate the ability of Driver’s SEAT to force use of the impaired arm. Subjects performed steering tasks with the system with and without resistance to the unimpaired arm. Force and EMG data were collected during the exercises. When the unimpaired arm was resisted, subjects showed an increase in EMG activity and force generated in the impaired arm, compared to the condition where both arms were treated equally. The increased muscle activation and force generated in the impaired arm indicated that the impaired arm was making a larger contribution towards moving the wheel.



Figure 1.2: Photograph of Palanca slider device on a tabletop, from [7].

1.4 Low-Cost Systems

A major disadvantage of systems like MIT-MANUS and ARMin is the high cost. It can cost tens or hundreds of thousands of dollars to implement these systems. For this reason, there is a trend towards creating less expensive systems that are able to achieve similar treatment outcomes.

Palanca was an early system built to this end [7]. Although not a true robot due to the lack of an actuator, it still fits within the paradigm of rehabilitation robotics because it is a human-machine interface that automates the rehabilitation process. The goal of the Palanca project was to bring the benefits of post-stroke robotic therapy to developing countries at a cost their citizens could afford. The system, shown in Figure 1.2, consists of a slider on a 1 m track, interfaced with a computer for use as a game controller. A row of infrared sensors determine the position of the slider and transmit it to the computer via a parallel port connection. By moving the slider with their hands, patients control the paddle in a Pong-like game.

A pilot study was done with Palanca to assess its efficacy as a rehabilitation tool. A total of four subjects participated, using the system 30 minutes per day, five days per week, for a total of 13 sessions each. Subjects were evaluated pre- and post-therapy using the Wolf Motor Function Test (WMFT). The WMFT evaluates motor function by measuring the time required to complete movement tasks, such as picking up an object. After treatment, subjects could complete WMFT

tasks, on average, three times as quickly as they had done before treatment. This data suggests that even low-cost systems can be effective rehabilitation tools.

The TheraJoy system and UniTherapy software were developed at Marquette University by L. Johnson, Feng, et al. to investigate the use of force feedback gaming joysticks as therapy robots in an under-supervised environment [8, 9]. Joysticks were used to perform positioning and tracking exercises with the UniTherapy software. Force feedback was used to provide assistive or resistive forces to the patient during the execution of movements. These joysticks could be used unmodified, or they could be modified to enlarge the workspace or change the workspace plane as seen in Figure 1.3. Specific muscle groups could be targeted with this system by changing the type and location of the joystick.

UniTherapy provides the computer interface for TheraJoy [9]. Its development was inspired by JavaTherapy, a web-based interface for rehabilitation using joysticks [10]. It is used by both the therapist and the patient to design and execute exercises. The UniTherapy patient interface is shown in Figure 1.3. The patient is presented with a cursor (dot) that follows the movement of the joystick, a target (box) that the patient must reach, and a path that the target follows for tracking exercises. UniTherapy allows at-home therapy by having the option of connecting to another computer running UniTherapy over the Internet. This allows a therapist to connect remotely to the patient’s home computer to monitor exercise. When running a local session of UniTherapy, the patient and therapist simply share a dual-monitor computer.

The TheraJoy pilot study was performed to assess the viability of TheraJoy and UniTherapy as therapy tools [8]. Sixteen high- and low-functioning stroke subjects and eight unimpaired subjects, participated in a mock therapy session. Subject performance on exercises was calculated within UniTherapy using root-mean-square tracking error, percentage of time on target, and mean time to target. EMG data was collected from muscle groups of interest on subjects’ arms. Performance metrics were found to be sensitive to the ability level of the subject, and muscle group activation was found to be sensitive to the location of the joystick. Also, subjects were generally receptive towards the system with its familiar joystick interface. These findings suggested that TheraJoy was a potentially useful therapy system.

1.5 TheraDrive Background

TheraDrive is a system currently in testing at the Clement J. Zablocki VA Medical Center [11]. It builds on the successes of the Driver’s SEAT and TheraJoy, using off-the-shelf force

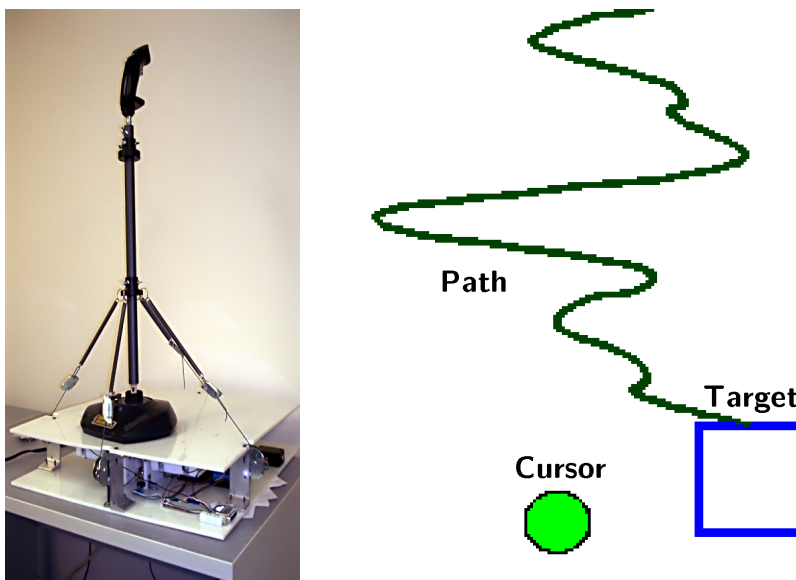


Figure 1.3: TheraJoy modified joystick (left) and UniTherapy patient task interface (right), from [8].

feedback steering wheels (Logitech Wingman) as therapy robots with a single degree of freedom. This steering wheel is mounted to an adjustable frame, allowing the plane of motion to be moved between the axial and sagittal planes on either side of the subject or the axial and frontal planes in front of the subject. A crank handle, which the patient grips, is mounted to the wheel. In the center of the frame is a movable chair on rails to ease the process of seating an impaired patient, and in front of the frame is a task display for visual feedback. The entire system is built from readily available parts, and it can be disassembled and folded for added portability. Ultimately, the goal of this project is to create a low-cost portable physical therapy system for stroke survivors that can be used at home or in a clinic. Other existing rehabilitation robots are not able to fill this niche because they are too bulky or too expensive.

Therapy with the system involves positioning and tracking exercises (rote therapy) as well as driving games (fun therapy). For rote therapy, the patient is presented with a point-to-point positioning task or a trajectory-following task by the UniTherapy software program, originally developed for the TheraJoy project. The wheel is used to move a cursor to a specified point or to guide the cursor along a moving path. The therapist can select from a large array of point layouts and trajectory shapes to adjust the task difficulty to the patient's ability level. Assistive or resistive forces can be generated by the wheel's force feedback motor, and the magnitude of these forces is set by the therapist before each exercise. This assistance/resistance usually takes the form of a spring force that attracts/repels the patient from the target position, but it can be changed to

emulate a mass, a damper, or random perturbations. Game therapy simply involves having the patient play a computer game using the wheel as a controller while the wheel motion is recorded. Typically a driving game such as Need for Speed or TrackMania is played, as these are well-suited to the wheel interface and designed to support force-feedback controllers.

A pilot study to assess the TheraDrive system was done in 2009 [11]. Ten subjects underwent 24 one-hour training sessions over the course of six to eight weeks. Subjects were divided into a group that did only rote therapy and a group that did only game therapy. Every four sessions, the subjects were evaluated using metrics including the Fugl-Meyer scale and the Ashworth test for spasticity. Both groups showed improvements in motor function and decreases in spasticity, but the sample size was too small to determine if the results were statistically significant. Additionally, the game-based therapies enhanced the motivation and engagement of patients and caused a slight increase in functionality gain over simple tracking and positioning exercises. However, one low-functioning patient showed little to no gain from therapy because the wheel could not exert sufficient assistive force to allow him to perform the tasks. This highlights a major weakness of the TheraDrive system—the wheel cannot exert sufficient assistive force to provide effective therapy to low-functioning patients.

1.6 Concurrent Research

Concurrent to this project, a mechanically similar system was developed at EFPL (Swiss Federal Institute of Technology, Lausanne) as a prototyping platform for exoskeletal elbow joint haptic controllers [12]. This system consists of a crank arm coupled to a motor through a torsional spring. An encoder on each side of the spring allows the measurement of both end effector position and interaction force. The patient holds the handle of the crank such that the axes of rotation of the elbow and crank are nearly coincident, isolating the elbow joint as the joint to be actuated. Experiments done using this system were used to investigate the effects of error in model parameters of a model-based impedance controller on the natural frequency of a haptic system.

1.7 Problem Definition

The TheraDrive system has shown success in preliminary human subject trials, but these trials have also revealed the limitations of the system. Stroke patients using TheraDrive showed

improvements in motor function after a typical course of therapy and found the training regimen to be interesting and engaging [11].

One stroke survivor with inability to use his hand for grasping and very little arm movement participated in the pilot study. An external forearm and upper arm support sling were employed so that he was able to perform exercises. He needed more assistance to complete tasks than the system could provide. Thus, there is a need to improve the wheel interface to provide better support for low-functioning stroke survivors who have difficulty with hand opening/closing and difficulty supporting the forearm against gravity.

The game controllers used in the TheraDrive system do not perform as desired. The maximum force-feedback moment produced by the wheel is not sufficiently large that patients cannot overpower it. This means that the wheel cannot be used to build strength in patients or to emulate a rigid constraint. The motor driving the wheel is a small brushed DC motor, which is underpowered for this task. Another major concern about the wheel is its ability to withstand off-axis forces and moments. This is an important issue because most patients using the wheel are uncoordinated, so they often exert lifting or bending forces on it. Since the wheel uses plastic-on-plastic bushings instead of roller bearings, its already-short wear life is shortened even further by increased loading on these surfaces. These sliding surfaces produce a significant amount of wear debris, which falls onto the gears and embeds itself between the gear teeth, as seen in Figure 1.4, causing damage to the gears. Backlash is also a concern with the wheel, since it uses molded plastic gears to couple the force-feedback motor. These gears have considerable backlash and compliance, both of which increase as the gear wears. The backlash of the gearing creates a dead zone where the force feedback does not influence the wheel's motion. Essentially, the wheel is a toy, and it lacks many features that would make it ideal for use as a therapeutic tool. Several Logitech wheels wore out over the course of the pilot study as a result of the large forces in the normal and tangential directions that users placed on the wheel. There is a need to improve the wheel interface and improve its ability to be effective throughout the therapy.

Patient-specific adaptive control is another important feature that the TheraDrive system currently lacks. Patients experience differing levels of impairment at different points in their range of motion, and a controller should be developed to account for this fact in order to deliver personalized therapy. The current controller for the wheel is only able to simulate a linear spring with constant stiffness. This means that during an exercise, patients can have great difficulty moving the wheel at some points along its travel but move the wheel with relative ease at other

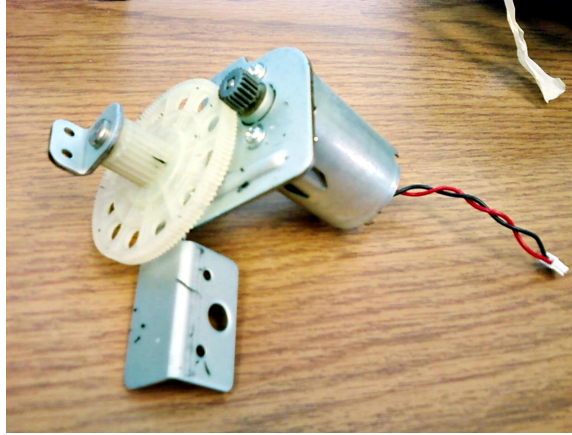


Figure 1.4: Photograph of failed gears from Logitech Wingman showing black wear debris between the teeth of the small plastic gear and on the mounting bracket.

points. Thus, a controller must be designed to adapt to each patient's individual form of impairment, based upon range of motion, torque, and speed.

1.8 Specific Aims

The specific aims of this project are to:

1. Develop a device that functions as a drop-in replacement for the Logitech wheel and is able to exert larger forces on the patient than the Logitech wheel is able to generate.
2. Extend the functionality of the TheraDrive system to accommodate low-functioning patients.
3. Carry out a preliminary study to assess the performance and usability of the new device.

CHAPTER 2 DESIGN OF SYSTEM

2.1 Design Goals

The problem definition in the previous chapter gives rise to the key design criteria:

1. The new robot crank system must be integrated into the existing Theradrive system and be able to mount to the adjustable frame in front and on the sides, allowing exercises to be performed in different planes.
2. The robot must be low-cost to maintain the affordable theme (less than \$3000 USD).
3. The robot must support torques on the crank larger than 220 in-lb (25 N-m).
4. The controller must be patient-specific and adapt forces at the crank to accommodate a variety of patients with strokes, especially low-functioning patients with motor weakness and poor coordination.
5. The robot must be backdrivable and safe (torque limited).
6. The forces on the crank arm must be measured and recorded.

2.2 Mechanical Design

2.2.1 Motor and Gearing

An actuator that can output mechanical power of the same order of magnitude as the human arm was constructed to fulfill the first specific aim and provide patients with large supportive forces. Due to budget and weight constraints, the motor selected is an aftermarket treadmill motor from Turdan Industry Co. It is a 2-pole brushed DC motor rated at 2.9 HP (2175 W) at 70% duty with a maximum speed of 8000 rpm and with windings rated for 130 V. Motor constants of this motor are $k_e = 0.155 \frac{\text{V}}{\text{rad/s}}$ and $k_T = 1.37 \frac{\text{in}\cdot\text{lb}}{\text{A}}$ ($0.155 \frac{\text{N}\cdot\text{m}}{\text{A}}$). The weight of this motor is 11 lb (5 kg), making it the lightest motor available that fit the design requirements. Although the motor is only rated for rotation in one direction, its windings and commutator are symmetrical, so it achieves the same performance turning backwards as it does turning forwards. The motor is overpowered for this application, but through the servo amplifier's voltage/current limiters and a mechanical torque limiter, the maximum mechanical power delivered to the patient

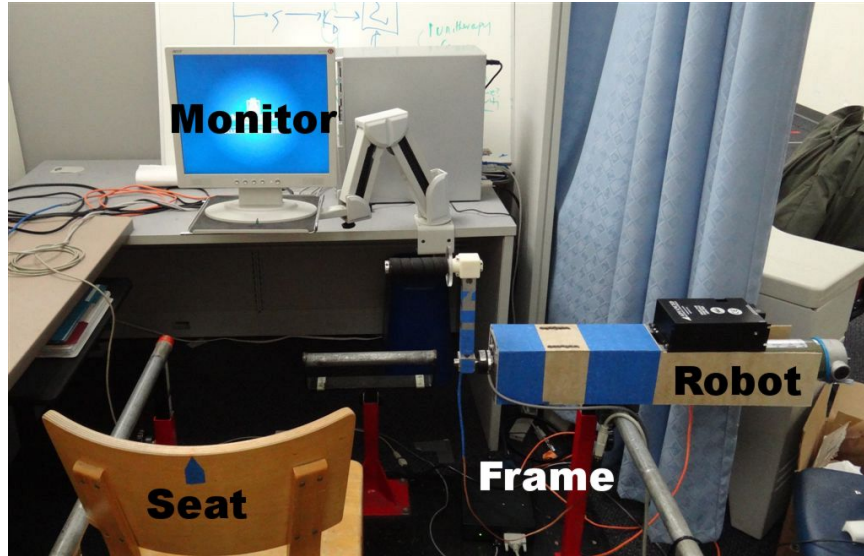


Figure 2.1: Photograph of new robotic system with mounting frame and patient seat.

is 1.27 HP (400 in-lb at 200 rpm, or 950 W). A maximum output torque of 400 in-lb (45 N-m) translates to a linear force at the end effector of 50 lbf (222 N), a value on the same order of magnitude as the linear output force of the human arm at the hand. Motor cooling is not necessary because the motor is being run well below its maximum rated power.

To couple the motor to the end effector, a planetary gearbox from Anaheim Automation with a 40:1 gear reduction ratio is used. This reduces the maximum end effector linear velocity to 14 ft/s (4.3 m/s), about triple the speed of a normal reaching movement. A planetary gearbox was selected over a harmonic drive because the planetary gearbox is more easily backdriven, allowing for smoother haptic interaction. The motor's output shaft was not compatible with the input of the gearbox (large-diameter threaded output shaft versus input for small-diameter plain shaft), so a flexible spider coupling was added between the two. The flexible coupling also reduces shock loads on the gearbox created from sudden reversal of the motor, which lengthens the expected life of the gears.

The motor and gearbox are mounted to a heavy aluminum chassis, providing sufficient rigidity while still minimizing weight. This chassis mounts to the same frame as the original TheraDrive using two heavy clamps. The robot and therapy environment are pictured in Figure 2.1.

2.2.2 Compliant Actuators

Collisions between a stiff robot and a human can cause injury to the human and damage to the robot. In order to allow safe human-machine interaction, robot actuators must have some amount of compliance. Compliant actuation is a sharp departure from industrial robotics, where robots and actuators are made with near-infinite stiffness. Van Ham et al. describe several strategies of producing a compliant actuator [13]. The simplest suggested method is a series elastic actuator (SEA), consisting of a stiff actuator in series with a compliant element such as a spring. The performance of this actuator configuration is limited because the stiffness cannot be adapted to different operating conditions without increasing the control law complexity. With variable stiffness, the compliant element of the SEA is no longer passive because the actuator stiffness is partially generated by the controller.

Variable-stiffness actuators overcome this limitation by introducing a second degree of freedom that controls the stiffness of the joint. There are three main ways to design a variable-stiffness actuator while still allowing the compliance of the actuator to be generated passively: agonist-antagonist actuators, structurally-controlled stiffness, and mechanically-controlled stiffness [13]. Agonist-antagonist actuators are a combination of two SEAs with nonlinear springs. These function much like the biceps and triceps in the human arm, with the actuators moving in the same direction to position the joint and in opposite directions to stiffen/loosen the joint. Nonlinear springs are necessary for these actuators because the stiffness of a linear spring does not change with its preload. The most notable form of these actuators is the pneumatic artificial muscle, which consists of rubber bladders that are inflated to cause a reduction in length [13].

Structurally-controlled stiffness allows adjustment of the stiffness of an actuator by altering the structure of the compliant element. An example of this is the Jack Spring actuator. This is a linear actuator that adjusts its stiffness by adding or removing active coils on a helical spring using a screw, as illustrated in Figure 2.2. It functions like a leadscrew in series with a variable spring.

With mechanically-controlled stiffness, the stiffness of an actuator is adjusted by moving the compliant element. The VS Joint developed by the DLR in Germany is an example of this actuator style [14]. It consists of rollers pressed against a cam by a set of springs (Figure 2.3). The position and stiffness of this joint are controlled independently by a drive motor and a smaller motor that adjusts the preload in the spring. When the joint is deflected, the rollers are pushed up by the cam, compressing the spring. The spring generates a restoring force that pushes the rollers

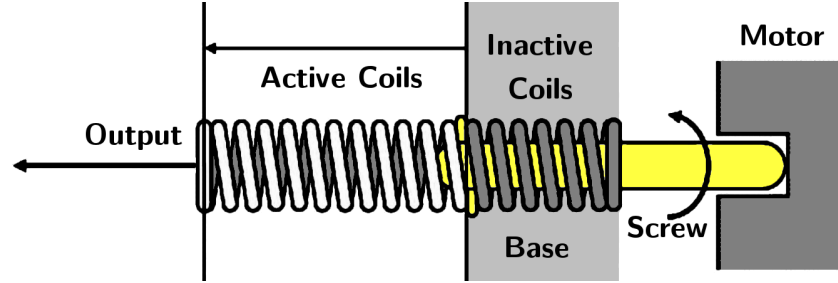


Figure 2.2: Illustration of the Jack Spring, from [13]. Rotating the screw will add or remove active coils from the spring.

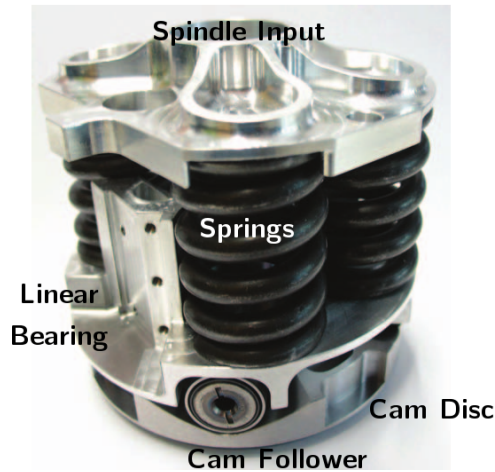


Figure 2.3: Photograph of VS Joint, from [14].

back towards the equilibrium position. Changing the shape of the cam allows for different spring types to be emulated, with progressive, degressive, or linear stiffness.

MACCEPA is another actuator with mechanically-controlled stiffness [15]. It is a powered elbow orthosis with an extension spring connected between the forearm link and a movable arm (Figure 2.4). The spring generates a restoring torque towards the movable arm when the forearm is deflected, and the joint stiffness can be changed by adjusting the pretension in the spring.

2.2.3 Compliant Torque Limiter

In order for the robot to apply large forces to the patient safely, a torque limiter must be added to the robot's transmission. Additionally, the transmission must be compliant to protect the patient from impact loads during a sudden reversal of the motor or during an accidental collision between patient and robot [13]. Both of these requirements are met with a single transmission

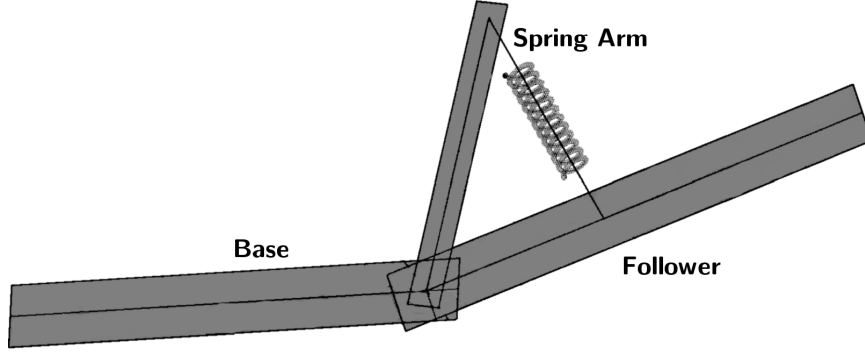


Figure 2.4: Sketch of MACCEPA elbow joint, from [15].

element in the robot. This compliant torque limiter is essentially a cross between a drill clutch and the VS Joint in [14]. It consists of a crown cam, splined to the input shaft, and a cam follower keyed to the output shaft, see Figure 2.5. The cam is held against the follower by a spring, but it is free to slide along the spline of the input shaft. Thrust bearings are placed on both sides of the limiter to take thrust loads off of the bearings in the gearbox and output shaft. Deflecting the output shaft causes the cam to compress the spring, generating a restoring torque. This creates a torsional spring from a linear spring, and the stiffness profile of the torsional spring can be altered by changing the shape of the cam surface. In an overload condition, the cam leaves the followers, disengaging the output shaft.

For an angular deflection of θ , the restoring torque $T(\theta)$ is defined by the relationship

$$T(\theta) = \left(\frac{dz(\theta)}{d\theta} \right) (r_c) (F_{ko} + kz(\theta)), \quad (2.1)$$

where $z(\theta)$ defines the shape of the cam surface, r_c is the cam radius, k is the linear spring stiffness, and F_{ko} is the preload in the spring. The preload in the spring is adjusted manually using a telescoping shaft collar to compress the spring. The torque limit is determined in part by the depth and slope of the cam profile and can be adjusted by varying the preload in the spring. The cam shape chosen for this robot is parabolic. With a parabolic cam path, increasing the preload in the spring increases the torsional stiffness. When the spring preload is low, this provides patients who cannot handle large interaction forces a softer robot to reduce discomfort. When the spring preload

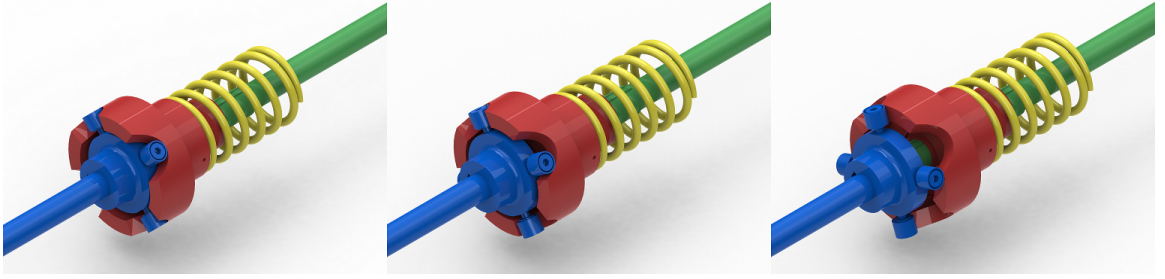


Figure 2.5: Torque limiter in equilibrium, loaded, and overloaded conditions (left to right).

is high, the robot is able to exert larger forces at the end effector without excessive output shaft deflection. The equation for restoring torque with a parabolic cam path of $z(\theta) = a\theta^2$ becomes

$$T(\theta) = 2a\theta r_c F_{ko} + ka^2 r_c \theta^3, \quad (2.2)$$

and the torsional stiffness becomes

$$\kappa(\theta) = \frac{dT}{d\theta} = cF_{ko} + 3ka^2 r_c \theta^2. \quad (2.3)$$

The cam motion is assumed to be quasi-static, as the spring force is much greater than the inertia of the moving parts. This assumption holds true for the real system except when the cam leaves the overloaded condition, moving from the flat surface to the steep parabolic surface. This is acceptable because exercises with the robot are aborted if the torque limiter is overloaded.

To size the cam to a chosen spring, five parameters must be known: the minimum torque limit, T_{min} ; the maximum torque limit, T_{max} ; the spring stiffness, k ; the spring force at maximum compression, F_{max} ; and the desired cam radius, r_c . The cam profile is parabolic, defined as $z(\theta) = a\theta^2$, and the profile depth is z_{max} . The maximum angular deflection before the follower leaves the cam then becomes

$$\theta_{max} = \sqrt{\frac{z_{max}}{a}}, \quad (2.4)$$

and the cam slope at this point is

$$\frac{dz}{d\theta_{max}} = 2\sqrt{az_{max}}. \quad (2.5)$$

The minimum torque limit is realized when there is zero preload in the spring, giving the equation

$$T_{min} = (2\sqrt{az_{max}})(r_c)(kz_{max}). \quad (2.6)$$

The maximum torque limit is realized when the spring is at full compression when the follower leaves the cam, giving the equation

$$T_{max} = (2\sqrt{az_{max}})(r_c)(F_{max} - kz_{max}). \quad (2.7)$$

This creates a system of two equations which can be solved for a and z_{max} to determine the cam shape. A final check must be done to verify that the determined cam shape will fit the allotted space without overlapping itself or creating thin features that would fail under load. If this check is failed, a different spring must be chosen.

The specific spring chosen for the torque limiter has a stiffness of 82 lb/in (144 N/cm) and a load of 137 lb (609 N) at full compression. Minimum and maximum torque limits were chosen as $T_{min} = 80$ in-lb (9 N-m) and $T_{max} = 400$ in-lb (45 N-m), respectively, and the cam radius was chosen to be 1.5 in (3.8 cm). With a crank arm radius of 8 in (20 cm), these torque limits correspond to 10 and 50 lb (44 and 222 N) of tangential force at the end effector, respectively. Solving the above equations yields a cam profile of $z(\theta) = 5.81\theta^2$ in (14.8 θ^2 cm) with a depth of $z_{max} = 0.418$ in (10.6 mm). Since the center of the cam follower must trace this profile, the physical cam surface is offset normal to the cam profile by the radius of the follower, 0.25 in (6.4 mm) in this case. The cam profile is repeated four times around the cam circumference in order to quadruple the mechanical strength of the torque limiter by using four cam followers and to balance the distribution of the spring force.

The minimum and maximum torque limits were tested by hanging barbells from the end effector with the crank arm below the horizontal position. The motor shaft was turned manually until either the crank arm rose past the horizontal position or the torque limiter disengaged. If the torque limiter did not disengage, the torque limit was not exceeded. With the torque limiter set to its minimum limit, the crank was able to lift a weight of 10 lb (44 N) but not a weight of 15 lb (66 N). With the torque limiter at its maximum limit, the crank was able to support a weight of 45 lb (200 N) but not a weight of 50 lb (222 N). This showed good agreement with the theoretical values of 10 lb (44 N) at the minimum limit and 50 lb (222 N) at the maximum limit.

Table 2.1: List of cam design parameters

Parameter	k	F_{max}	r_c	T_{min}	T_{max}	a	z_{max}
IPS value	$82 \frac{\text{lb}}{\text{in}}$	137 lb	1.5 in	80 in-lb	400 in-lb	$5.81 \frac{\text{in}}{\text{rad}^2}$	0.418 in
SI value	$144 \frac{\text{N}}{\text{cm}}$	609 N	3.8 cm	9 N-m	45 N-m	$14.8 \frac{\text{cm}}{\text{rad}^2}$	1.06 cm

2.2.4 Load Cell

To measure interaction forces between human and robot, a load cell was built into the crank handle. These interaction forces must be known in order to close the impedance control loop and for data collection during experiments/exercises. The compliant element of the drivetrain could not be used to measure torque in this case because its stiffness is variable and nonlinear, so a load cell is required. The load cell is composed of a cantilever beam with eight strain gauges mounted, fixed to the crank arm. The strain gauges used are Vishay Micro-Measurements 062LW series gauges. The beam has a circular cross-section and a conical taper, narrowing towards the free end, to maximize strain in areas under low load. This, in turn, increases the sensitivity of the load cell by maximizing the strain experienced by each strain gauge. The handle grip is fitted around the load cell and rotates freely on bearings. Fixing the load cell to the arm rather than the grip allows forces to be measured in the radial-tangential coordinate system of the crank arm and simplifies the conversion of interaction forces to joint-space torque. Strain gauges are mounted around the circumference of the beam at 90-degree intervals at two points along its length, as shown in Figure 2.6. Diametrically opposed gauges are wired in half-bridge configuration and measure bending strains at two points due to moments about the radial and tangential axes, for a total of four measured strains. The half-bridge configuration provides temperature compensation because opposing gauges will experience the same amount of thermal strain, creating no change in the ratio of the two gauges' resistances. (Temperature compensation could not be accomplished by matching the thermal expansion of the gauges to that of the load cell bulk material because strain gauges thermally matched to SAE Type 304 Stainless Steel, the load cell bulk material, were not readily available.) Temperature compensation is important because it cannot be assumed that the strain gauges will remain at a constant temperature due to their proximity to the patient's hand, a source of heat.

The four strains measured by the bridges are used to calculate bending stresses, bending moments, and shear and moment reactions at the fixed end of the beam. The interaction force is converted to joint-space torque by multiplying the tangential shear reaction force by the crank arm

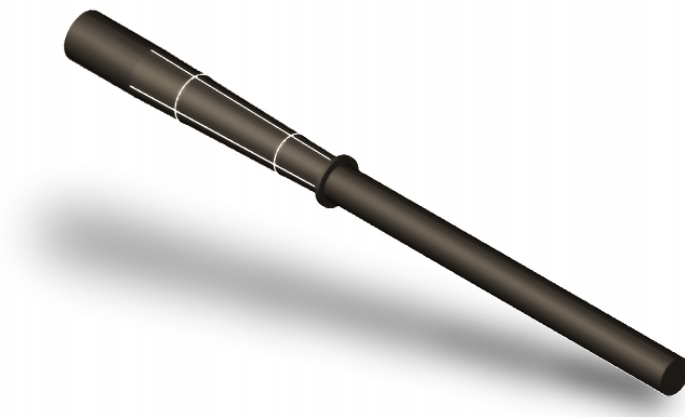


Figure 2.6: Rendering of load cell beam showing strain gauge mount points at intersections of white lines. Fixed end is at left and free end is at right.

radius, and this is used as the feedback to the impedance controller. The other three reactions are used only for clinical data collection. Calculation of the shear and moment reactions at the fixed end of the beam can be illustrated with a moment diagram (bending moment plotted as a function of position along the beam, as in Figure 2.7). Measured strains at the tangential gauge bridges are converted to bending moments using Hooke's Law and bending stress equations. A straight line is drawn through these two points, and the slope of the line is equal to the shear reaction, and the value of the line at the fixed end is the moment reaction. This procedure is repeated for the radial gauges.

The load cell was calibrated by clamping the fixed end to a workbench and hanging various weights from various points along the length of the beam and then recording the voltage outputs of the strain gauge bridge amplifier. Weights ranged from 2 to 12 lb (0.8 to 5.6 kg), in combinations of 2-lb weights and were hung 1 to 8 in (25 to 200 mm) from the fixed end of the beam. For each load condition, the amplifier output was recorded for the gauges proximal and distal to the fixed end of the beam, in the plane of loading. A total of 58 data points including a zero point were collected. This method allowed the strain gauge bridges measuring tangential loads to be isolated from the strain gauges measuring radial loads and vice versa. From the value of the weight and its distance from the fixed end of the beam, the shear and moment reactions could be calculated. Least-squares planar regressions were performed to find the shear and moment reactions, each as a function of the two amplifier outputs after offset nulling. The correlation

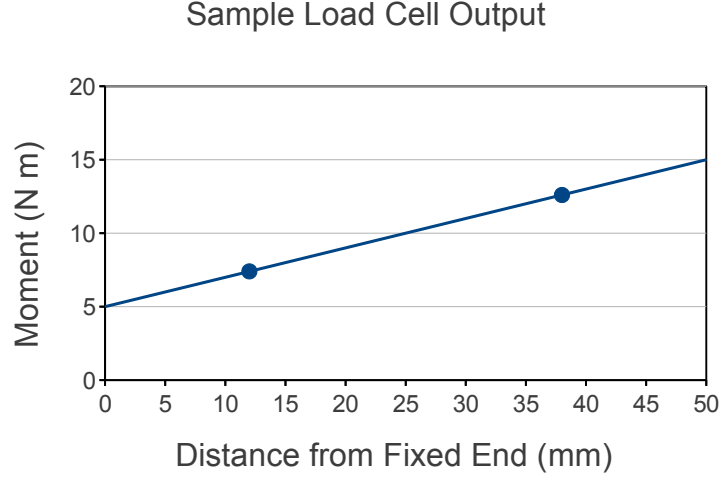


Figure 2.7: Graphical example of moment calculation from strain gauge measurements.

squared was greater than 0.99, indicating a very good fit of the plane to the data. In matrix form, the reactions for the tangential direction are as follows.

$$\begin{pmatrix} V \\ M \end{pmatrix} = \begin{bmatrix} 21.19 & -65.81 \\ 7.099 & -96.52 \end{bmatrix} \begin{pmatrix} E_p \\ E_d \end{pmatrix} \begin{matrix} \text{lbf} \\ \text{in} \cdot \text{lb} \end{matrix} = \begin{bmatrix} 94.26 & -292.7 \\ 0.8021 & -10.91 \end{bmatrix} \begin{pmatrix} E_p \\ E_d \end{pmatrix} \begin{matrix} \text{N} \\ \text{N} \cdot \text{m} \end{matrix}, \quad (2.8)$$

where V and M are the shear and moment reactions in lb and in-lb, respectively, and E_p and E_d are the voltage outputs of the amplifier from the bridges proximal and distal to the fixed end of the beam. Force and moment data for loads exceeding 12 lb (5.6 kg) must be extrapolated from this data. Equipment to measure weights larger than 2.5 lb (1.1 kg) within 0.5% was not available for this test, so the barbells used to characterize the torque limiter could not be used for load cell calibration because their tolerance was not given. This was deemed acceptable because strain gauges are highly linear force sensors when mounted to linear elastic materials, and the beam was designed not to undergo plastic deformation until the moment applied to the beam exceeded 600 in-lb (100 lb or 445 N applied to the tip).

2.3 Electrical Design

2.3.1 Servo Amplifier

Because the motor in the robot will rotate in both directions and will encounter conditions where it is backdriven, a four-quadrant servo drive is necessary. When a motor is backdriven, it behaves as an electric generator, converting mechanical power into electrical power. Four-quadrant servo drives are able to store or dissipate the electrical power generated when a motor is backdriven and can drive the motor in either direction. Driving the motor is a plug-in analog pulse-width modulated (PWM) servo amplifier from Advanced Motion Control (model 30A20AC) interfaced with a DAQ system. This amplifier does not need an external power supply and instead runs on rectified mains voltage (40-190 VDC), saving space, weight, and money. Built-in voltage and current limiters allow the amplifier to limit the maximum velocity and torque of the motor. All necessary power protection components are contained within the amplifier, including a capacitor bank and a shunt regulator. The capacitor bank stores energy and serves as a buffer, allowing the motor to consume more current than the power supply outputs for brief periods of time without overloading the supply. This is useful in applications where the motor must accelerate as quickly as possible. Shunt regulation is necessary to dissipate excess power generated by the motor when it is backdriven. Some of the energy generated can be stored in the capacitor bank, but when the capacitor bank voltage exceeds the value set in the shunt regulator, a resistor is shunted across the capacitors to dissipate the excess energy.

2.3.2 Bridge Amplifier

The strain gauge bridge amplifier is designed around the Burr-Brown INA125 instrumentation amplifier with precision ($\pm 0.5\%$) voltage reference, a chip designed to power strain gauges and amplify strain gauge signals. Four strain gauge half-bridges in the load cell produce four voltage signals proportional to strain along specific axes at specific points. The half-bridges are balanced using a 100 k Ω , 20-turn trimmer potentiometer in parallel. Bridges are completed using a buffered precision voltage divider, shared between all bridges. These four strain signals are used to calculate bending stresses, then bending moments, and finally shear and moment reactions at the fixed end of the beam. The interaction force is converted to joint-space torque by multiplying the tangential shear reaction force by the crank arm radius, and this is used as the feedback to the

impedance controller. Radio-frequency (RF) noise created by the PWM servo amplifier is a significant source of error in the load cell signal. To combat this, the strain gauges are connected to the strain gauge amplifier through shielded twisted-pair leads, and the output of each half-bridge is filtered with a RF choke and shunt capacitor. The corner frequency of the filter is 81 kHz with a damping factor $\zeta = 7.1$. (The large damping factor was due to the limited availability of suitable inductors.) Additionally, a ferrite ring was added to the motor leads to reduce the emitted RF noise. The voltage gain of the amplifier is 250 V/V, and the output range is ± 5 V.

2.3.3 Encoder

A 1000-line hollow-shaft rotary optical encoder from Red Lion (model ZPJ1000A) was chosen as the position sensor for the haptic robot. The hollow shaft allows the encoder to be mounted directly to the output shaft of the robot without requiring any belts or couplers. With x4 decoding (four counts per quadrature cycle), the resolution of measured end effector position is 80 counts per inch (32 per cm). The maximum switching frequency of this encoder is 200 kHz, 60 times the switching frequency with the output shaft rotating at its maximum speed of 200 rpm. The index of the encoder is aligned with the center of the robot workspace, where the crank arm is vertical, corresponding to the starting position of every exercise trajectory. Encoder cables are double-shielded to block RF noise emitted from the servo, and encoder signals are filtered digitally by the encoder DAQ board (Measurement Computing QUAD-04).

2.4 Software Design

2.4.1 Force Feedback

A force feedback device is able to follow the motion of the user and exert reaction forces. The wheel used in the TheraDrive system is an example of a force feedback device with a single degree of freedom. A geared DC motor is used as a torque generator to actuate the wheel and provide force feedback, and a single-turn potentiometer is used as a position sensor.

Closed-loop control of a force feedback system can be implemented using an impedance approach or an admittance approach. Impedance control takes a velocity input from the user and outputs a force at the end effector; conversely, admittance control takes a force input and outputs a velocity in response. Each of these strategies has its weaknesses—impedance control has a

minimum impedance equal to the open-loop impedance of the linkage, and admittance control is more susceptible to non-ideal dynamics [16]. TheraDrive uses impedance control because impedance control makes it easier to work with force fields, since both the force field and the impedance controller describe force as a function of position and its derivatives.

With either control strategy, the output of the device should have a high dynamic range and a high natural frequency relative to the input to keep the dynamics of the system ideal. Additionally, the system must be kept marginally stable to give it zero effective inertia, enhancing its transparency. However, since the system operates under marginally stable conditions, it is susceptible to quantization noise, sampling delays, friction, sensor/motor saturation, modeling errors, rendered stiffness, and other contributors to instability. In human-machine interaction, safety is of utmost concern, and an unstable controller poses a risk of injury to the human from large, unexpected robot movements. Stability of a haptic system can be analyzed using methods such as the Routh-Hurwitz criteria (for linear systems only), and instabilities can be mitigated by increasing sampling rates and adding damping to the controller. Because of these limitations, a force feedback device has a maximum renderable stiffness, i. e. it cannot render a perfectly rigid constraint such as a wall. This maximum stiffness is determined by the threshold change in end effector force over one sample period above which the controller becomes unstable [16].

2.4.2 Adaptive Control Strategies

Adaptive control of rehabilitation robots generally takes an assist-as-needed approach. “Assist-as-needed” means that the robot should give as little assistance as possible to get the patient as close to a desired trajectory as possible. This can be considered to be an optimization problem, with an objective function

$$J = \frac{\lambda_1}{2}e^2 + \frac{\lambda_2}{2}F^2, \quad (2.9)$$

where the robot must minimize a weighted sum of patient error e and assistive force F [17].

Solving this optimization yields the desired assist-as-needed control law. However, solving optimization requires *a priori* knowledge of the relationship between assistance and error. For this reason, this strategy only works with model-based controllers.

Adaptive control can also be used to alter the difficulty of an exercise based on subject performance. The Wrist-Robot uses this adaptive strategy alongside a nonadaptive assist-as-needed controller [18]. Subjects use this robot to track oscillating trajectories of the three rotational degrees of freedom of the wrist. The assistive controller renders a quadratic torsional spring that attracts subjects to the desired wrist orientation. The stiffness of the spring is specified manually by the therapist. After each period of oscillation, the adaptive controller increments the oscillation frequency by a constant amount and simultaneously decrements the frequency by an amount proportional to the mean absolute tracking error over the previous period of oscillation. This allows the difficulty of the exercise to increase gradually until the appropriate difficulty level for the subject is reached.

2.4.3 Control Design

The robot is controlled using a dedicated computer running Mathworks Simulink xPC Target OS for real-time control and data acquisition. This PC is fitted with two DAQ boards to read sensors and communicate with the servo amp: an encoder board from Measurement Computing (PCI-QUAD04) and a multipurpose board from National Instruments (NI PCI-6251). A host computer running Matlab interfaces with the target computer through an ethernet connection to upload executable binaries and display the patient interface.

Control of the robot is achieved through three loops, all running at a sample rate of 1000 Hz to allow for smooth controller response. The control architecture is illustrated in Figure 2.8. The innermost control loop is the impedance proportional-integral-derivative (PID) controller. This controller was designed and tuned in simulation using Matlab's Control System Toolbox. The topology of the impedance controller is PID with feedforward, taking a force command and force feedback at the input and producing a motor current signal at the output. The feedforward path consists of a constant multiplied by the signum of the end effector velocity in order to cancel the effects of Coulomb friction. After tuning in simulation, the PID controller gains found were used as a starting point for manual tuning of the impedance controller on the real robot. The PID controller gains used, in force output per unit force error, are $k_P = 0.592$, $k_I = 5.03 \frac{1}{s}$, and $k_D = 0.00700s$. The feedforward term is ± 4.22 lb (18.8 N). Stability of the controller was analyzed using root loci. The z-domain poles of this controller lie within the unit circle, so the controller is stable, and the impedance loop as a whole is stable because the mechanical plant (mass-spring-mass) is inherently also stable.

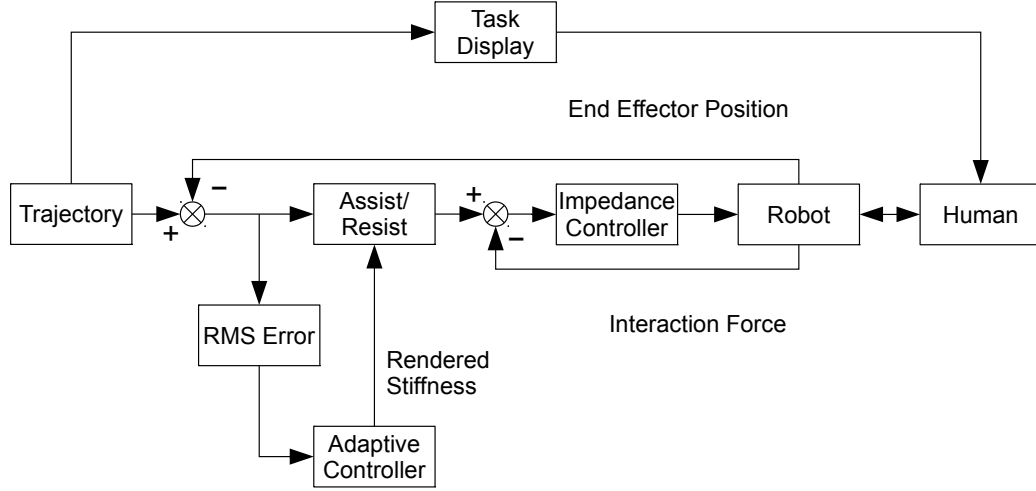


Figure 2.8: Flowchart of robot control.

Next is the assistive/resistive controller, which renders a force field in the robot's workspace. The controller divides the workspace into 17 regions, spaced $\frac{1}{16}$ of a revolution apart. Each region has a value assigned to it to define the assistive/resistive stiffness of the robot at that location, with negative stiffnesses being assistive and positive stiffnesses being resistive. It is necessary to implement position-dependent stiffness because stroke patients usually have inconsistent abilities in the range of motion of their impaired arm. The stiffnesses between regions are interpolated to produce smooth transitions between regions of different stiffnesses. Assistive stiffness is rendered as a linear spring pulling the patient towards the target end effector position with a small damper added to reduce overshoot. Resistive stiffness is rendered as a linear spring that repels the patient from the target position, creating an unstable system that the patient must stabilize. For patients requiring assistance, the stiffnesses are either zero or negative so the patient only receives assistance. The opposite is true for patients requiring resistance.

The outermost control loop is the adaptive controller. It uses a strategy similar to that of the Wrist Robot [18]. This controller evaluates patient performance in real-time and continuously adapts the stiffness of the robot to the patient's ability. Since patient performance depends in part upon the amount of assistive/resistive stiffness, altering the stiffness will cause a change in patient performance. Patient performance is quantified as the root-mean-square (RMS) trajectory tracking error over the past three seconds, a metric chosen because it is also used to quantify subject

performance in the original TheraDrive. This error is compared to a desired error of 0.25 rad (2 in or 51 mm of arc length), a value corresponding to the width of the target that the patient tracks. A nonzero tracking error is desired because this maintains difficulty of exercises. If the desired tracking error were zero, the robot would always provide maximum assistance to every patient, and if the desired tracking error were too large, the robot would always provide maximum resistance to patients. Stiffness values in a quadrant centered around the end effector are adjusted proportionally to the difference between actual and desired tracking error, essentially establishing a proportional control loop around the gains of the assistive/resistive controller. Over time, the adaptive controller shapes the stiffness profile to suit the patient's ability level, ensuring that exercises will be difficult but doable and that the patient will not be assisted more than is necessary. "Difficult but doable" means that patients should be presented with a challenge sufficient to maintain motivation and a moderate degree of exertion, but the challenge should not be so great as to cause patients to become frustrated or to fail to complete exercises.

For testing purposes, the adaptive controller has several operating modes that disable some or all of its features, allowing for specific variables to be isolated in experiments. Zero impedance mode disables all assistance and resistance and makes the end effector behave as if it were rotating freely. Static assist/resist modes disable the adaptive controller and provide a fixed stiffness. These first two modes emulate the behavior of the original TheraDrive system with the wheel rotating freely or with spring assist/resist, respectively, allowing for a direct comparison of the two systems in testing. Plain adaptive mode provides adaptive stiffness but disables position-dependent stiffness. Position-dependent adaptive mode enables all features of the adaptive controller.

2.4.4 Task Software

The task software on the host computer is made to emulate the appearance of UniTherapy to the patient, allowing it to satisfy the first specific aim of creating a drop-in replacement. The interface presents the patient with an arrow cursor that follows the position of the end effector, seen in Figure 2.9. The cursor is situated at the bottom of the screen, and the desired trajectory scrolls down the screen towards the cursor, presenting a blue box at the current target location. The cursor changes from red to green while it is within the target box. The framerate of this task display is approximately 33 Hz, the same as that of UniTherapy.

The operator interface uses Matlab scripts to automate random trajectory generation and data uploading to the robot. Most functions can be performed with a single command at the

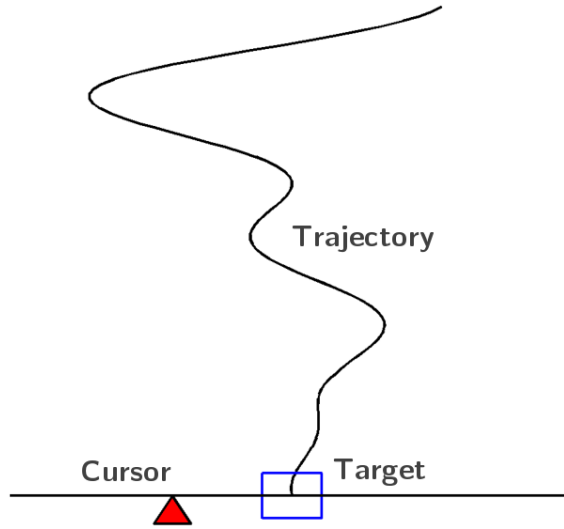


Figure 2.9: Haptic robot GUI.

Matlab command line. Data is automatically streamed via Universal Datagram Protocol (UDP) from the robot to the host computer during exercises and saved to the Matlab workspace. The streamed data packets contain a timestamp, commanded and actual position, commanded and actual haptic interaction force, off-axis forces, and controller gains. After an exercise is complete, the data can be saved to a file for later analysis.

2.5 Discussion and Conclusion

The hardware developed for the haptic robot satisfies the first specific aim of being a drop-in replacement for the Logitech wheel by mounting to the same frame and interfacing physically with human subjects in the same manner. The second specific aim is fulfilled by the robot's high force output, which enables the robot to provide assistance to subjects who are too weak or too uncoordinated to use the Logitech wheel. By emulating the main features of the UniTherapy software, the haptic robot's software presents subjects with an interface that has the same look and feel of UniTherapy, fulfilling the first specific aim. The adaptive control software provides more flexibility than UniTherapy in the way assistance/resistance is applied to the human subject, which allows therapy to be personalized to each subject; this fulfills the second specific aim of extending accommodations to low-functioning subjects.

The biggest shortcoming of the haptic robot is the performance of its motor. The treadmill motor is designed to spin in one direction at a constant speed, not to provide smooth motion at

near-stall speeds. Because of this, the motor has a low number of windings on the armature, which leads to noticeable cogging effects at slow speeds. The low number of windings also causes the torque constant of the motor to be position-dependent, leading to inconsistent performance of the impedance controller. A motor designed specifically as a servo would offer vastly superior performance, but budget and weight constraints did not permit such a motor to be used.

Because the robot lacks a dead-man switch on the crank handle, it has no means to sense whether a user is in contact with the end-effector. The handle grip spins freely, so a switch cannot be installed in it without risk of damage to the wiring of the switch. This creates a safety concern because the impedance controller is unstable if there is nothing in contact with the end effector, causing a dangerous situation. Due to this concern, subjects are made to wear a grasp-assist glove that prevents them from releasing the handle, and subjects perform exercises under close supervision. However, these safety measures are not an ideal solution to this problem because the glove does not fit all people. It is recommended that the handle be modified to accept a dead-man switch or similar device in order to enhance the current safety measures.

CHAPTER 3 SIMULATION EXPERIMENTS

3.1 Introduction

Due to the high force output of the haptic robot, experimental controllers could not be tested on the real hardware without a significant risk of operator injury or collateral damage. This created a need for a safe test platform for controllers. To this end, a simulation model was created based upon the characteristics of the real system, and care was taken to ensure the model had good fidelity to the real hardware. It models all of the major mechanical components of the robot as well as all of the robot controllers discussed in Section 2.4.3. A simulated human arm from another project was added to model the interaction between human and robot. This model was used for the initial tuning of the impedance PID controller and for the first experiments with the adaptive controller, discussed in the following.

3.2 Simulation Design

A model of the robot and human patient was written in Mathworks Matlab, Simulink, and SimMechanics to aid in controller design and testing without risk of injury. The crank and human arm were modeled as a four-bar linkage, with the motor coupled to the crank through a nonlinear spring. Parameters for the model robot, such as link inertia and motor torque constant, were found through characterization of the motor and through analysis of 3D part drawings in SolidWorks. Equations of motion were created and solved numerically within SimMechanics based upon the mechanical model. Using SimMechanics, all nonlinearities in the system, such as encoder quantization and Coulomb friction, could be modeled easily, and a more accurate model of the system than a linearized analytic model was produced. The human arm model was ported from another computational model created by Formica et al. for use with the MIT-Manus robot [19]. This model simulates the dynamics and control of the human arm with or without stroke impairment. It is a planar model consisting of a forearm/wrist segment, an elbow joint, an upper arm segment, and a fixed shoulder joint. The forearm/wrist segment is 35 cm long with a mass of 1.54 kg, and the upper arm is 25 cm long with a mass of 1.96 kg, corresponding to metrics of a human of average height and weight. The simulation model is shown graphically in Figure 3.1. A

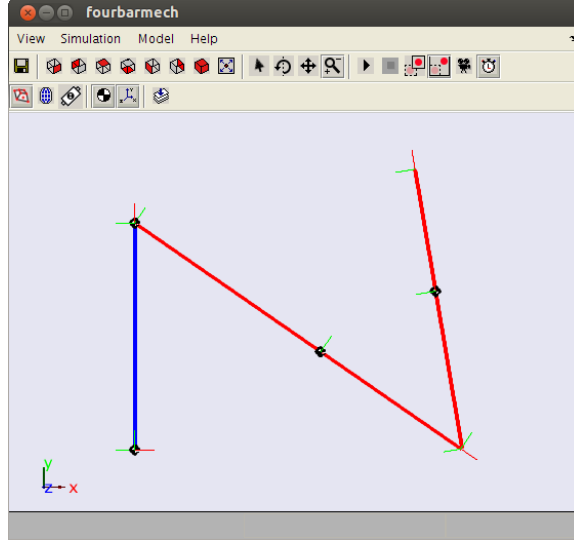


Figure 3.1: Graphical representation of simulation model showing crank arm (left), human forearm (center), and human upper arm (right).

trajectory planner and joint-space proportional-derivative (PD) controller are used to model human control of the arm. The PD controller is of the form

$$\begin{pmatrix} T_e \\ T_s \end{pmatrix} = \begin{bmatrix} P_{ee} & P_{se} \\ P_{es} & P_{ss} \end{bmatrix} \begin{pmatrix} u_e \\ u_s \end{pmatrix} + \begin{bmatrix} D_{ee} & D_{se} \\ D_{es} & D_{ss} \end{bmatrix} \begin{pmatrix} \dot{u}_e \\ \dot{u}_s \end{pmatrix} \quad (3.1)$$

with cross-coupling terms in each gain matrix being nonzero, where u is the position error vector in rad in joint space and T is the joint torque output vector in N-m. The PD controller gains are based on the stiffness and viscoelasticity of the average human arm and are defined as

$$P = \begin{bmatrix} 8.67 & 2.83 \\ 2.51 & 10.8 \end{bmatrix} \frac{\text{N} \cdot \text{m}}{\text{rad}}; \quad D = \begin{bmatrix} 0.76 & 0.18 \\ 0.18 & 0.63 \end{bmatrix} \frac{\text{N} \cdot \text{m}}{\text{rad/s}}. \quad (3.2)$$

The MIT-Manus arm model also contains a trajectory planner that simulates human trajectory planning in point-to-point movements, but this part of the model was not used because exercises consist of trajectory tracking rather than point-to-point movements, and trajectories are already defined in trajectory tracking tasks. Stroke impairment is simulated in the Manus model by perturbing the desired arm trajectory with a triangle function, creating a new trajectory that deviates from the ideal trajectory to simulate the disjointed piecewise trajectories taken by a

stroke-impaired arm. For example, if the ideal trajectory were defined as $\theta(t)$, the perturbed trajectory would be defined as

$$\theta_p(t) = \theta(t) + a (\theta(t) \bmod b - b/2), \quad (3.3)$$

where a and b define the amplitude and period of deviation, setting the severity of the simulated impairment. The arm then attempts to follow this perturbed trajectory, while the robot tries to follow the unperturbed trajectory. The trajectories used in exercises are a sum of four sine waves of different frequency and phase that create a pseudorandom smooth path.

3.3 Experiment Design and Hypotheses

To verify that the adaptive controller was able to adapt properly to patients with varying levels of impairment, simulations of the system were run with the MIT-MANUS human arm model [20]. The first simulation was run with the unimpaired arm model to verify that the adaptive controller was stable, i. e., the resistive stiffness would settle to a steady-state value after a period of time. An exercise was run for ten minutes of simulation time, in position-dependent adaptive mode starting with assistive/resistive gains of zero, and the controller was allowed to adapt to the arm's performance. The average assistive/resistive gain as a function of time was recorded for the duration of the simulation. This procedure was then repeated for a simulated stroke patient to verify that the adaptive controller would be stable when adapting to a stroke-impaired subject. It was expected that the controller gain would show a first-order decay towards a steady-state value, with the controller providing resistance to the unimpaired and assistance to the impaired arm model. A first-order response from the adaptive controller was expected because it is a simple proportional controller with a first-order transfer function.

To verify that the position-dependent aspect of the adaptive controller functioned properly, the stroke arm model was used [20]. The perturbation function used to simulate impairment was set such that the maximum perturbation would occur at ± 1 rad, where an angular position of 0 rad corresponds to the crank pointing straight up. A simulated exercise was run for ten minutes of simulation time, starting with an assistive/resistive gain of zero, and the controller was allowed to adapt to the arm's performance. Assistive/resistive gain as a function of end effector position

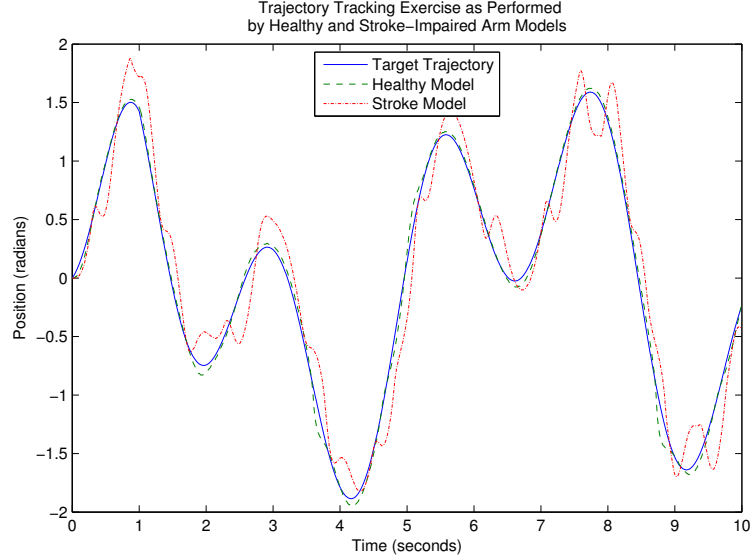


Figure 3.2: Healthy and stroke-impaired simulation models tracking a trajectory.

was recorded after the simulation was completed. The plot of gain versus position was expected to show peaks in assistive gain at ± 1 rad, where the tracking error of the arm was highest.

3.4 Simulation Results

The baseline performance of the healthy and stroke patient models in a sine tracking exercise is displayed in Figure 3.2. Both models tracked the trajectory with the robot controller running in zero-impedance mode. The healthy model tracks the trajectory with little error, but the stroke model has large errors at the points where the spatial perturbation is at a maximum.

Figure 3.3 shows the adaptive controller responding to the performance of a simulated healthy patient. A sine tracking exercise was run for ten minutes of simulation time, and the resistive stiffness, averaged over the workspace, was plotted versus time. The controller response is approximately first-order with a time constant of 90 seconds, entering a limit cycle at steady state. Steady-state oscillations have the same period of oscillation as the sinusoidal trajectory.

The performance of a simulated healthy patient before and after controller adaptation is shown in Figure 3.4. The baseline performance of the patient with zero resistance had RMS tracking error well below the target level of 0.25 rad, so the adaptive controller added resistance to the exercise until the target error was achieved. With the resistive gain adapted, the patient was

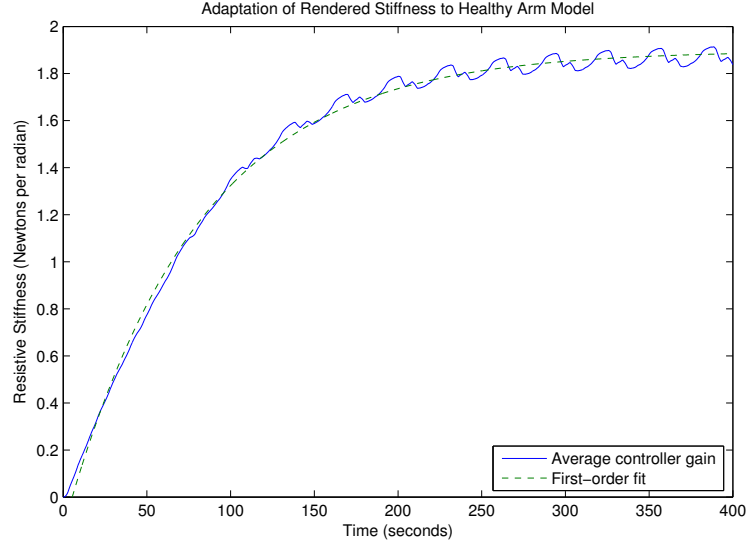


Figure 3.3: Rendered stiffness being adapted to healthy patient performance (simulated).

still able to track the reference trajectory, but with an increase in error from 0.058 rad at baseline to 0.230 rad with resistance. This increase in tracking error is apparent in Figure 3.4.

The response of the adaptive controller to a simulated stroke patient is shown in Figure 3.5. The assistive gain is plotted over time during a ten-minute sine tracking exercise. The controller response shows first-order decay with a 90 second time constant and oscillations at steady state with the same period as the tracked trajectory (12.6 seconds). RMS tracking error decreases from 0.307 rad over the first 30 seconds to 0.265 rad over the last 30 seconds.

Figures 3.3 and 3.5 show the adaptive controller modifying the stiffness of the robot to suit the ability levels of a simulated healthy subject and simulated stroke subject. With zero resistance, the healthy subject tracks the trajectory very closely, but after the adaptive controller adds resistance, the RMS tracking error increases to the desired level of 0.25 rad. Likewise, the tracking error of the stroke subject decreased until it reached the desired level as the controller gains were adapted. This demonstrates the ability of the adaptive controller in simulation to adjust the stiffness of the robot to achieve a desired level of performance from a subject.

Oscillations in controller gains, plotted in Figures 3.3 and 3.5, are caused by the shape of the trajectories the patients are tracking. This is because there are slight variations of difficulty in the motions that compose the trajectory. For example, tracking fast movements is more difficult than tracking slow movements, so changes in the trajectory velocity will perturb the gains of the adaptive controller.

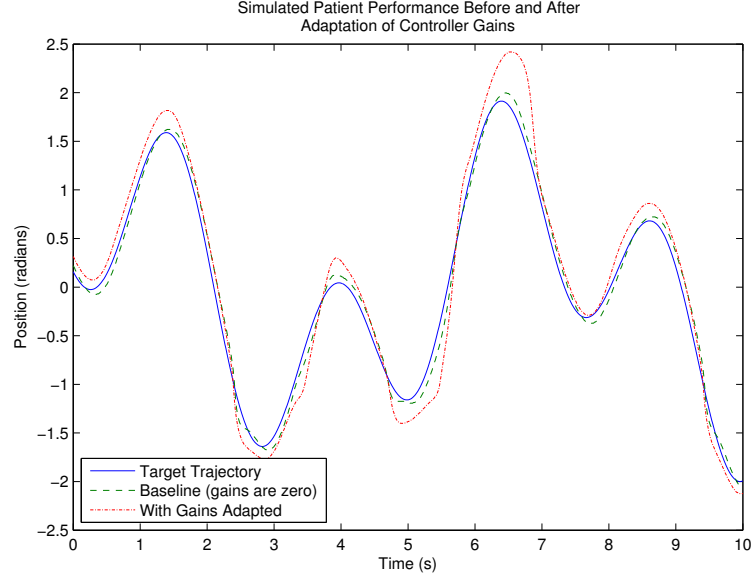


Figure 3.4: Simulated healthy patient performance before and after gains are adapted to provide sufficient resistance.

For comparison, the adaptive controller response of the Wrist-Robot is shown in Figure 3.6. The frequency of wrist oscillation—analogue to assistive/resistive gain with the haptic robot—is plotted over time for a series of eleven exercises with a human subject. A first-order response similar to that of the simulation model can be seen, but there is no data on steady-state behavior.

Performance of the position-dependent adaptive controller with a simulated stroke subject is displayed in Figure 3.7. The triangle perturbation function used to simulate impairment was defined to have peaks at ± 1 rad, causing maximum perturbation of the trajectory at the aforementioned peaks. A sine tracking exercise was run until the adaptive controller reached steady state, and the resulting gains were plotted. This graph shows two peaks where the assistive gain is larger (more negative): one large peak at 1 rad and a smaller, wider peak at -1.3 rad. These peaks correspond roughly to the locations where perturbation of the trajectory to simulate stroke was largest (± 1 rad).

Position-dependent assistive gain is important for therapy exercises with stroke patients, as they have varying levels of ability throughout the robot's workspace. Figure 3.7 shows the assistive gain of the simulated robot plotted against end effector position. This graph shows peaks in the gain that correspond roughly with the locations of peak trajectory perturbation in the stroke subject model. The peaks are of different magnitudes due to the differing mechanical advantage of the arm at these points. At -1.3 rad, the arm is extended and has little mechanical

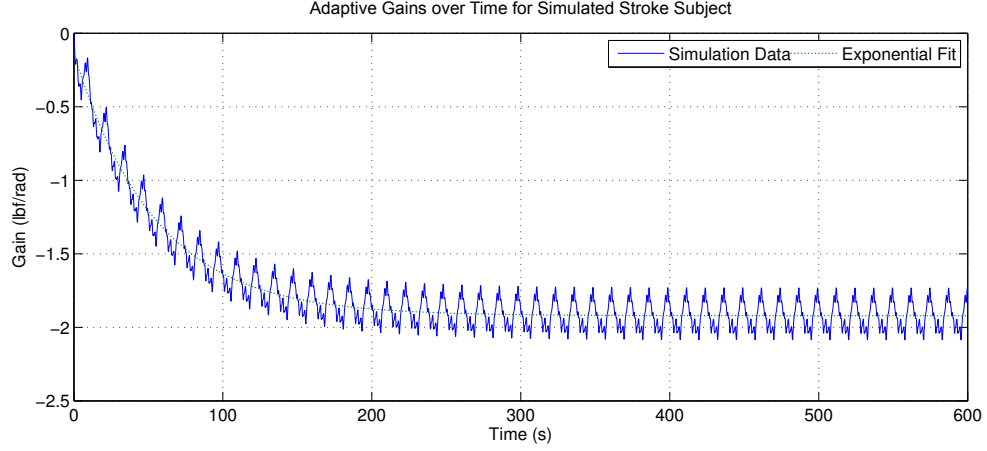


Figure 3.5: Rendered assistive stiffness adapting to simulated stroke model over time. More negative values indicate more assistive gain.

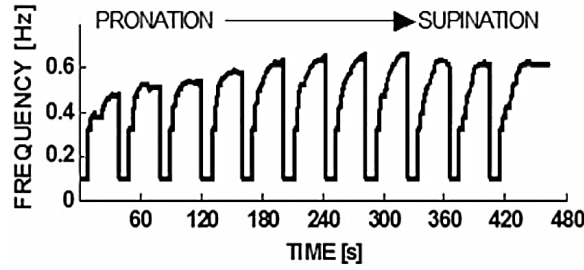


Figure 3.6: Wrist-Robot adaptive controller response through 11 exercises, from [18].

advantage over the robot, and less assistance is required to correct tracking errors. However, at 1 rad, the arm is close to the body, giving the patient a large mechanical advantage, so more assistance is required to correct the trajectory. Mechanical advantage is also the reason the peak at -1.3 rad is not at the expected position of -1 rad. The mechanical advantage of the patient is lower at the angle of -1 rad than at -1.3 rad because around -1 rad, there is a point where the tangent to the path of the elbow intersects with the axis of the crank, creating a singularity in the kinematics. At this point, the shoulder joint has zero mechanical advantage over the crank because it cannot exert any moment on it, putting the arm's overall mechanical advantage at a minimum. The arm has more mechanical advantage at -1.3 rad, so more assistance is required at this point to correct the arm's trajectory than is required at -1 rad.

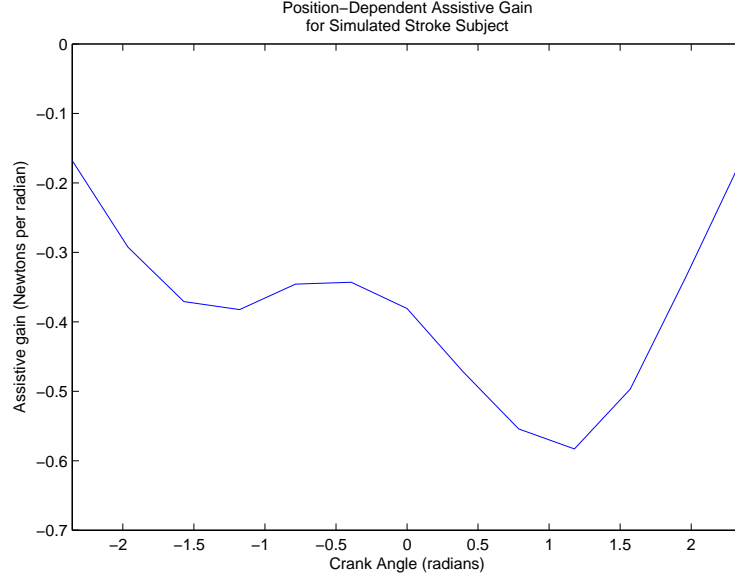


Figure 3.7: Position-dependent gains adapted to simulated stroke patient performance.

3.5 Discussion and Conclusion

The adaptive controller simulation experiments were successful in that the data recorded supports both hypotheses. The plots of gain over time in Figures 3.3 and 3.5 both show a general first-order trend, neglecting the oscillations, and Figure 3.7 shows the position-dependent adaptive controller assigning the highest gains to regions with the highest tracking errors. The controller also shows sensitivity to impairment level, as seen by the difference in steady-state gain between the healthy arm in Figure 3.3 and the stroke-impaired arm in Figure 3.5.

The first-order trend of controller gain is similar to the results seen in the Wrist-Robot experiment [18]. Subjects tracked oscillating trajectories using the robot, and the adaptive controller increased the oscillation frequency proportional to the subject's performance. Because of this relationship between performance and frequency, oscillation frequency for the Wrist-Robot is an analog to rendered spring stiffness in the haptic TheraDrive robot, as both these parameters modulate exercise difficulty. Wrist oscillation frequency with the Wrist-Robot showed the same first-order response characteristic in each of the eleven exercises in Figure 3.6 as was seen in the adaptive controller simulation.

The Wrist-Robot frequency plot shows little to no oscillation. This highlights a weakness in the simulation's adaptive controller. The assistive/reisistive gain oscillates at steady state. These oscillations are stable, as they do not grow exponentially as the exercises progress, and they are a

phenomenon caused by the shape of the exercise trajectory, as the two have the same period (12.6 seconds in Figure 3.5). Methods of removing these oscillations include increasing the length of the window used to calculate RMS tracking error and decreasing the adaptation gain. However, both these methods slow the rate of adaptation and make the controller less responsive to sudden spikes in tracking error that occur when, for example, a stroke subject becomes stuck in one position. Ultimately, the oscillations are inseparably linked to the trajectory, and eliminating them would cause an unacceptable loss of controller performance. However, the perturbation is small enough that it is imperceptible to patients, as the exercises themselves present variations of force of much greater magnitude and speed than those caused by small variations of stiffness.

Position-dependent gains developed as hypothesized. The largest assistive gains were seen in the positions where the simulated subject’s tracking error was at a maximum, which in turn correspond loosely to the points where the trajectory perturbation to simulate stroke was largest. This shows that the position-dependent adaptive controller fits the “assist as needed” paradigm; it is able to provide extra assistance only where it is necessary to maintain subject performance.

Although it was modified to work in the haptic robot simulations, the MIT-Manus arm model still behaved as it did in simulations discussed by Formica et al. [19]. The tracking error of the arm model showed sensitivity both to the level of assistance provided and to the level of impairment modeled.

CHAPTER 4 HUMAN SUBJECT EXPERIMENTS

4.1 Introduction

The final specific aim of this project was to carry out a pilot study to assess the viability of the new haptic robot as a therapy tool and to validate how well the first two aims were met. For these reasons, a human subject pilot study was performed.

The haptic robot improves upon the original TheraDrive system by increasing the torque output and adding an adaptive controller. It was hypothesized that these improvements would make the haptic robot more able to support low-functioning subjects, and these improvements would be apparent through improvements in quantitative measures of subject performance, such as RMS trajectory tracking error.

4.2 Data Collection Procedure

To compare the old and new TheraDrive systems, human subjects were used to test both systems. A total of six human subjects were used in the testing of the two TheraDrive robots, two normal subjects and four with stroke. A seventh subject participated but was dropped from the study for non-compliance. The Upper Extremity Fugl-Meyer test was used to quantify the degree of impairment in the stroke subjects [21]. This test rates the level of functionality remaining in the impaired arm on a scale of 0 to 66, with a score of 0 corresponding to a total loss of function and a score of 66 corresponding to no significant loss of function. Low-functioning subjects were defined as subjects with a score of 30 or lower; medium-functioning subjects had scores ranging from 31 to 50; and high-functioning subjects had scores of 51 or greater. Unimpaired subjects were not given the Fugl-Meyer test because these subjects had a level of function beyond the maximum level measurable by the test. Subjects and their Fugl-Meyer scores are listed in Table 4.1.

During experiment sessions, subjects performed mock therapy exercises consisting of sinusoidal trajectory tracking using each robotic system. Trajectories consisted of a sum of four sinusoids, centered about the vertical crank position, of amplitude 0.5 rad and frequencies of 0.5, 1.0, 1.5, and 2.0 rad/s. The relative phases of the sinusoids were randomized for each exercise to reduce learning effects while still preserving the range of motion and frequency content. Each exercise consisted of tracking the given trajectory for 90 seconds. Four exercises were performed for each operating mode of each robot, for a total of 24 exercises per subject. The first robot presented

Table 4.1: Listing of subjects with Fugl-Meyer scores

Subject Number	Type	UE-FM Score	Function Level
2248-01	Normal	N/A	Full
2248-03	Stroke	43	Medium
2248-04	Stroke	31	Medium
2248-05	Stroke	29	Low
2248-06	Stroke	55	High
2248-07	Normal	N/A	Full

to the subject was chosen randomly. The operating modes of the robots used in experiments were as follows:

1. Old TheraDrive, no assistance/resistance
2. Old TheraDrive, spring assist/resist
3. New TheraDrive, zero impedance (equivalent to old TheraDrive, no assistance/resistance)
4. New TheraDrive, static assist/resist (equivalent to old TheraDrive, spring assist/resist)
5. New TheraDrive, plain adaptive mode
6. New TheraDrive, position-dependent adaptive mode

Following exercises with each robot, subjects were asked to complete a series of surveys to express their opinions of each robot. A motivation survey presented subjects with a list of 25 statements such as “I believe doing this activity could be beneficial to me,” and “I think I am good at this activity,” and asked them to rate their agreement with these statements on a scale of 1 (strongly disagree) to 7 (strongly agree). Each of these statements was assigned to a certain category of motivation, and responses to the questions added points to the respective categories. A post-activity survey asked subjects about their level of exertion and overall opinion of each robot and included space for subjects to leave general comments. After completion of exercises with both robots, subjects were given a final survey in which they indicated a preference of robot along with the strength of this preference and the reason for this preference. See Appendix A for the complete listing of survey content and analysis procedures.

4.3 Old versus New TheraDrive Systems

4.3.1 Procedure

To compare the old and new systems in terms of their ability to assist/resist subjects, the root-mean-square trajectory tracking error was computed for each exercise. Root-mean-square (RMS) tracking error between the measured and commanded trajectories is calculated as follows:

$$\Delta x_{RMS} = \sqrt{\frac{1}{T} \sum_{i=1}^{T/\Delta t} (x_{meas}(i) - x_{com}(i))^2 \Delta t}, \quad (4.1)$$

where Δt is the sample period and T is the duration of the exercise. The tracking error without assistance/resistance was defined as the baseline performance for each robot, and data from each subject was normalized by the baseline performance for the appropriate robot. Normalization was performed so that the relative change in performance after the addition of assistance/resistance could be compared for each robot. It was hypothesized that the change in performance after assistance/resistance was added would be greater for the new robot versus the old robot. However, it is also hypothesized that this change in performance will only be significant for low-functioning subjects who require more assistance than the Logitech wheel can provide. A p-value of less than 0.05 was considered significant for all statistical tests performed.

To compare subject performance changes with assistance/resistance by different robots, data needed to be normalized by baseline performance with the robot used to cancel any other factors that would influence subject performance, such as robot dynamics. Data from the old and new TheraDrive robots was normalized with respect to baseline performance with each robot by the following equation, where x is the parameter of interest and \bar{x}_{base} is the baseline value of the parameter.

$$x_{norm} = \pm \frac{x_{raw} - \bar{x}_{base}}{\bar{x}_{base}} \quad (4.2)$$

The sign of the normalized value is reversed for subjects requiring assistance, allowing the absolute value of the change in performance to be compared across all subjects in one group. Data points were categorized by subject, robot mode, and trial number; and an ANOVA test was performed with the null hypothesis that measures of performance do not vary with respect to any of the three

categories. In the event that the null hypothesis of the ANOVA test was rejected for any category, a t-test would be performed comparing the means within the category, with the null hypothesis that the means are equal.

Additionally, subjects were asked to fill out surveys regarding their opinions of both robotic systems. These surveys covered topics such as motivation, comfort, perceived safety, and degree of effort. It was hypothesized that subjects would prefer using the new system over the old system.

4.3.2 Results

Results of the ANOVA test confirm the null hypotheses that *normalized* data does not vary significantly by subject ($p = 0.28$) and that data does not vary significantly across different trials of the same exercise ($p = 0.62$). The first of these two results indicates that normalization was effective in removing the influence of overall subject performance, and the second result indicates that the experiment is repeatable. The ANOVA test rejects the null hypothesis that data does not vary with respect to robot mode ($p < 0.0001$), suggesting that performance in exercises with assistance/resistance could vary between the two robots. Using a one-tailed Student's t-test, the changes in subject performance with the addition of assistance/resistance for each robot were compared, with a null hypothesis that the means are equal. The null hypothesis was rejected with a p-value of less than 0.0001, meaning that the addition of assistance/resistance with the new haptic robot causes a larger change in subject performance than the addition of assistance/resistance with the old system for all subjects in general. A tabular description of the t-test performed can be found in Table 4.2, and aggregated raw data is shown in Figures 4.1 and 4.2. Complete listings of raw data appear in Appendix B.

Data from stroke subjects' motivation surveys is graphed in Figure 4.3. Motivation data collected from normal subjects was collected but not analyzed because it was deemed irrelevant in the context of motivating rehabilitation for stroke patients. No significant difference in motivation scores between the two robots was observed, as each score for the haptic robot is within less than one standard deviation of the corresponding score for TheraDrive. Subjects performed the same tasks with each robot and viewed very similar displays, so most of the motivating features were identical across the two systems, with the exception of exercise difficulty, and scores from the motivation surveys reflected this. Additionally, the set of data was too small to produce significant results in any but the most extreme cases. When explicitly asked their preference of robot in the final post-activity survey, all stroke subjects chose the new robot over the original TheraDrive

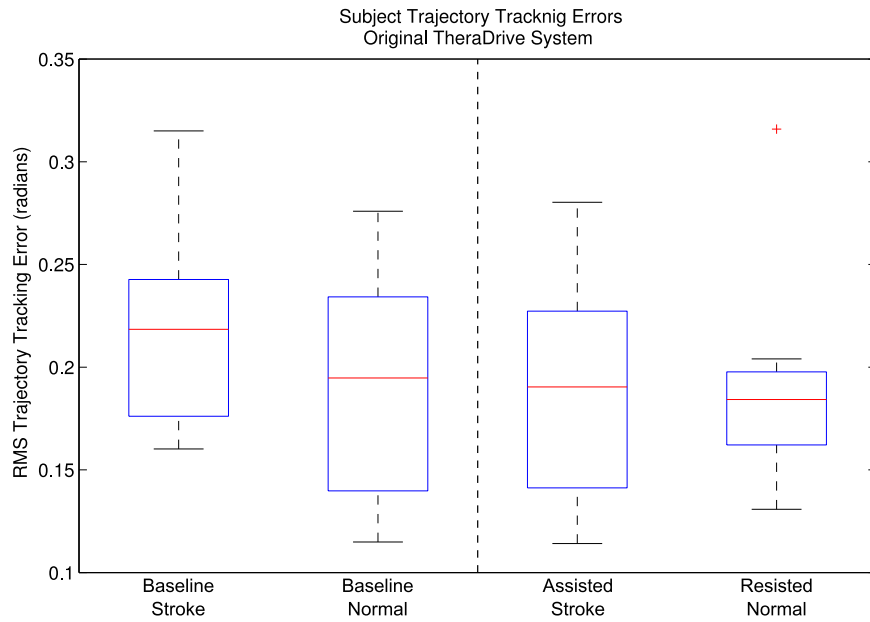


Figure 4.1: Quartile boxplots of raw RMS tracking error of subjects with the old TheraDrive system. Outliers are shown as crosses.

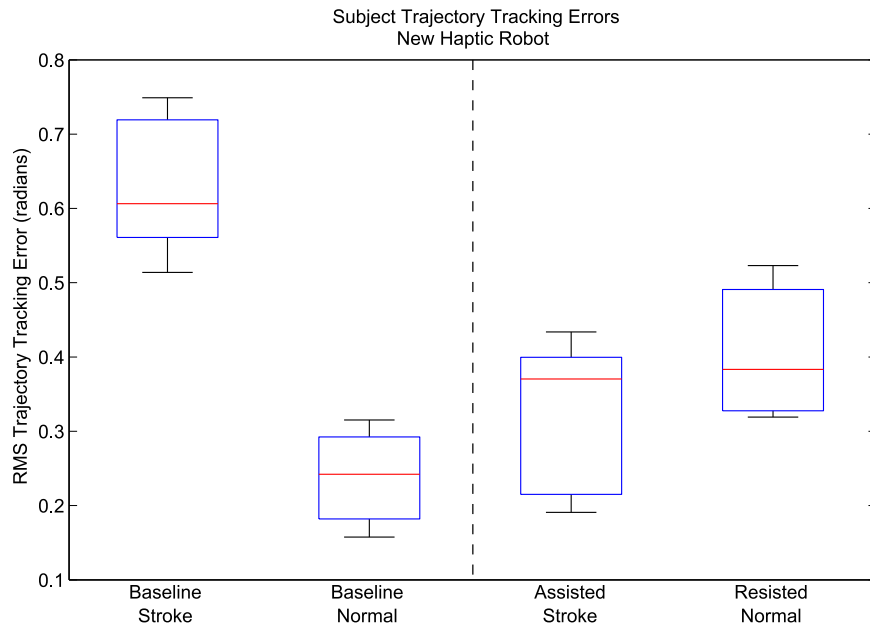


Figure 4.2: Quartile boxplots of raw RMS tracking error of subjects with the new haptic robot. Outliers are shown as crosses.

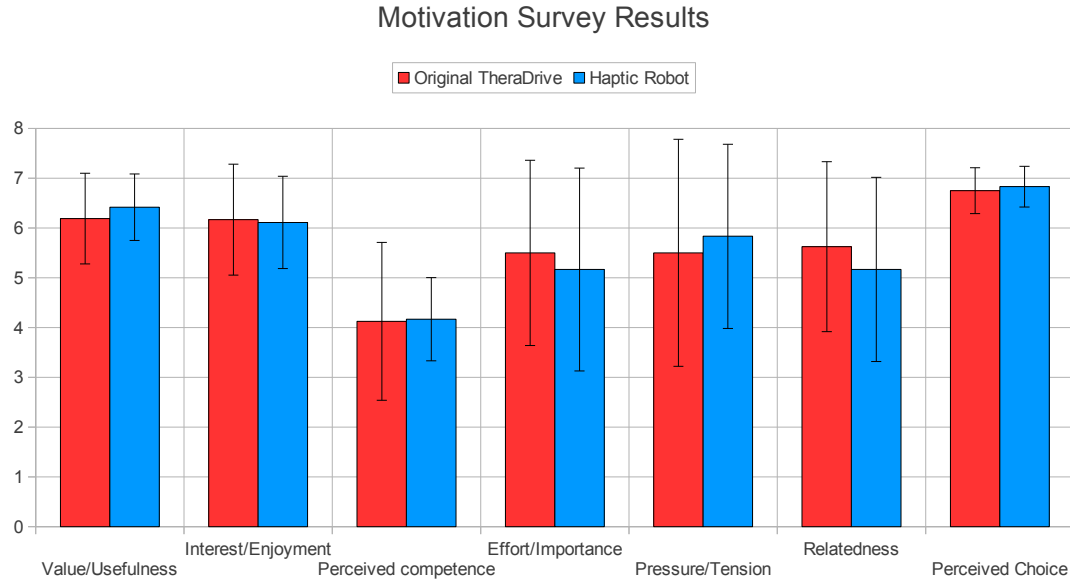


Figure 4.3: Motivation survey results for stroke subjects, plotted by category for each robot. Motivation is rated on a scale of 1 to 7. Standard deviation is shown by error bars.

Table 4.2: List of t-test parameters and result

Comparison	Condition 1	Condition 2	Hypothesis	p-value
Delta RMS error when assist/resist added	Logitech wheel	Haptic robot	$2 > 1$	0.0000

system, despite the lack of difference in motivation scores between the two systems. Subjects commented that they liked the increased assistance and increased range of motion of the new robot. One subject liked the fact that the robot had a similar feel to an arm bike used in physical therapy. Healthy subjects did not receive the new system as well as stroke subjects. Because of the high resistive force output, healthy subjects had difficulty controlling the robot, especially when changing the direction of motion. These high forces also raised concerns about safety and caused more fatigue than was desired.

4.3.3 Analysis and Discussion

The absolute normalized change in RMS tracking error after the addition of assistance/resistance was used as the primary measure to compare the efficacy of the assistive/resistive force provided by the TheraDrive joystick and the haptic robot. Although the

RMS trajectory tracking error is calculated by the same method for each robot, the tracking error must still be normalized in order to compare data across the two systems because subject's baseline performances with each system were not necessarily equal. The sign reversal on the performance change for stroke subjects gave the magnitude of the change in performance after the addition of assistance/resistance. The t-test then compared these normalized magnitudes and found the assistance/resistance of the haptic robot to produce a significantly larger change in performance than that of the joystick. This result was expected, as the haptic robot is able to exert 30 times the torque of the joystick, so it would follow that the haptic robot can provide more assistance or resistance. These results show that the mechanical linkage of the haptic robot satisfies the first two specific aims of the study because it is able to provide more support to subjects—enough for low-functioning subjects to be able to complete exercises.

In the baseline zero-impedance mode, the haptic robot is much more difficult for stroke-impaired subjects to move than the Logitech wheel. This can be seen when comparing the non-normalized RMS tracking error, shown in the leftmost boxes in Figures 4.1 and 4.2. The likely cause of this increase in difficulty is the larger workspace of the haptic robot, which has a crank arm radius approximately double that of the Logitech wheel. All of the stroke subjects had a reduced range of arm motion, which caused them difficulty in moving the haptic robot's crank through its full range of motion. This difference in baseline difficulty between the two robots may have influenced the results of the statistical comparison.

4.4 Nonadaptive versus Adaptive Control

4.4.1 Procedure

To compare adaptive versus nonadaptive controllers, data from the exercises performed by subjects with the new haptic TheraDrive system was examined. RMS trajectory tracking error in nonadaptive mode was taken as a baseline for each subject. Compared to this baseline reading, it was hypothesized that the tracking error in plain adaptive mode would be closer to the adaptive controller's desired tracking error of 0.25 rad (the width of the target box), and the tracking error in position-dependent adaptive mode would be even closer to this desired value.

To evaluate the performance of the position-dependent portion of the adaptive controller specifically, RMS tracking error as a function of end effector position was evaluated. RMS error over position was calculated by dividing the robot workspace into the 17 regions used by the

position-dependent controller and calculating the RMS tracking error only for data points within each region. It was hypothesized that the plot of the error over position would be flatter for position-dependent adaptive control than for plain adaptive control because the position-dependent controller is designed to make the RMS tracking error at any point in the workspace equal the target value, whereas the plain adaptive controller is designed to make the average error over the workspace equal a target value. Flatness was evaluated by taking the difference between the maximum and minimum error over position, excluding regions outside the limits of the exercise trajectory. Additionally, correlation r was calculated between the baseline RMS tracking error as a function of position and the position-dependent adaptive controller gains at the same positions. It was hypothesized that there would be a strong negative correlation between tracking error and controller gain (positive values of gain are resistive) because the adaptive controller is programmed to add more assistance as tracking error increases.

Force data collected in trials with the haptic robot was analyzed to investigate the effect of adaptive control on the off-axis forces exerted on the robot by subjects. Off-axis forces were the radial forces exerted on the crank by patients, and these forces produced no motion in the crank. Because off-axis forces were inherently lower when subjects were exerting less force, off-axis forces were calculated as the ratio of the magnitude of off-axis interaction force to the magnitude of total interaction force, shown below.

$$F'_r = \frac{|\vec{F} \cdot \hat{r}|}{\|\vec{F}\|} = \frac{|F_r|}{\sqrt{F_r^2 + F_t^2}}, \quad (4.3)$$

where \vec{F} is the interaction force vector with radial and tangential components F_r and F_t , and \hat{r} is the unit direction vector of the crank radius. For each exercise, the ratio of off-axis force was calculated for every data point and then averaged over the duration of the exercise, creating a single ratio that described the amount of off-axis force exerted by the subject during the exercise. These ratios were then normalized by each subject's baseline performance in the static assist/resist mode. Zero-impedance mode was not used as a baseline because with the tangential force held at zero by the robot controller, the only forces that could be exerted on the robot were off-axis forces, making F'_r always equal to 1.

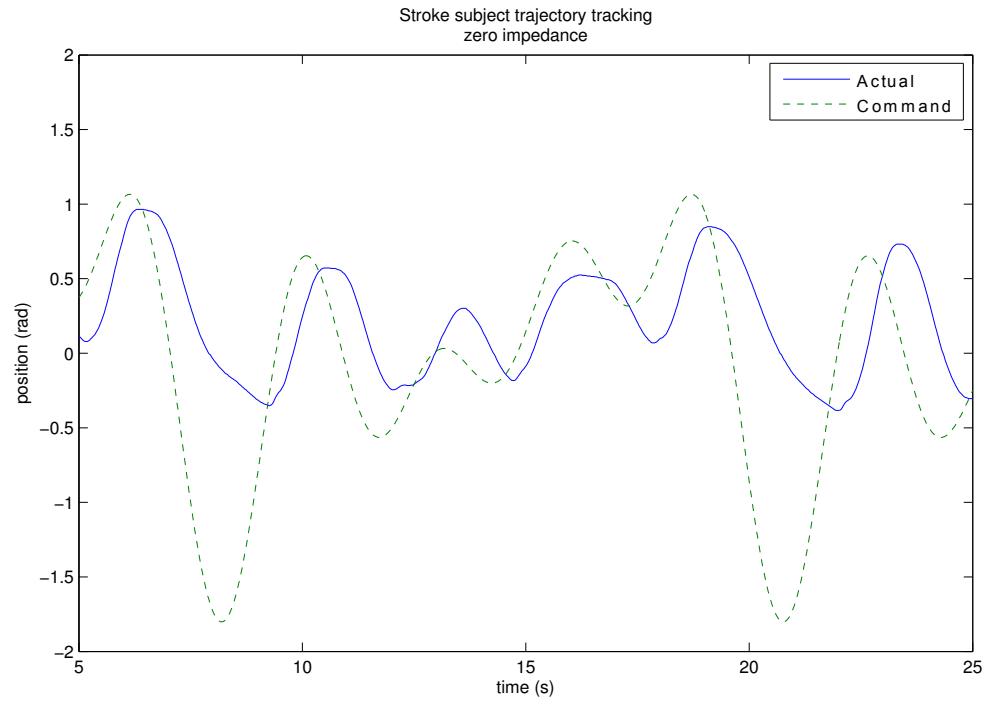


Figure 4.4: Stroke subject (2248-05) tracking trajectory in zero-impedance mode

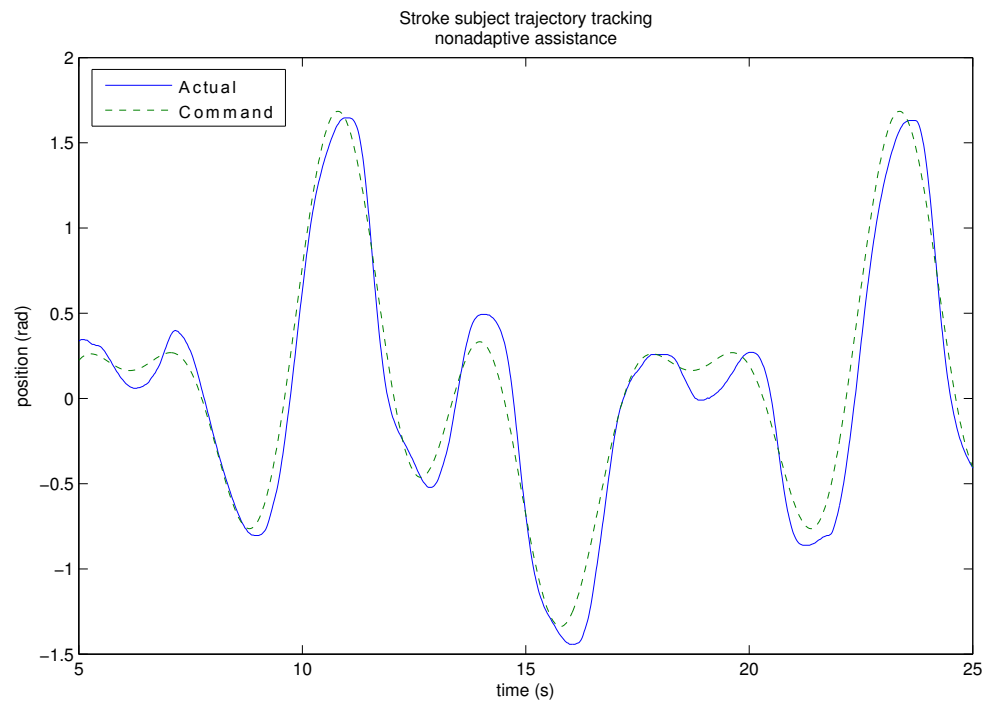


Figure 4.5: Stroke subject (2248-05) tracking trajectory in static spring assist mode

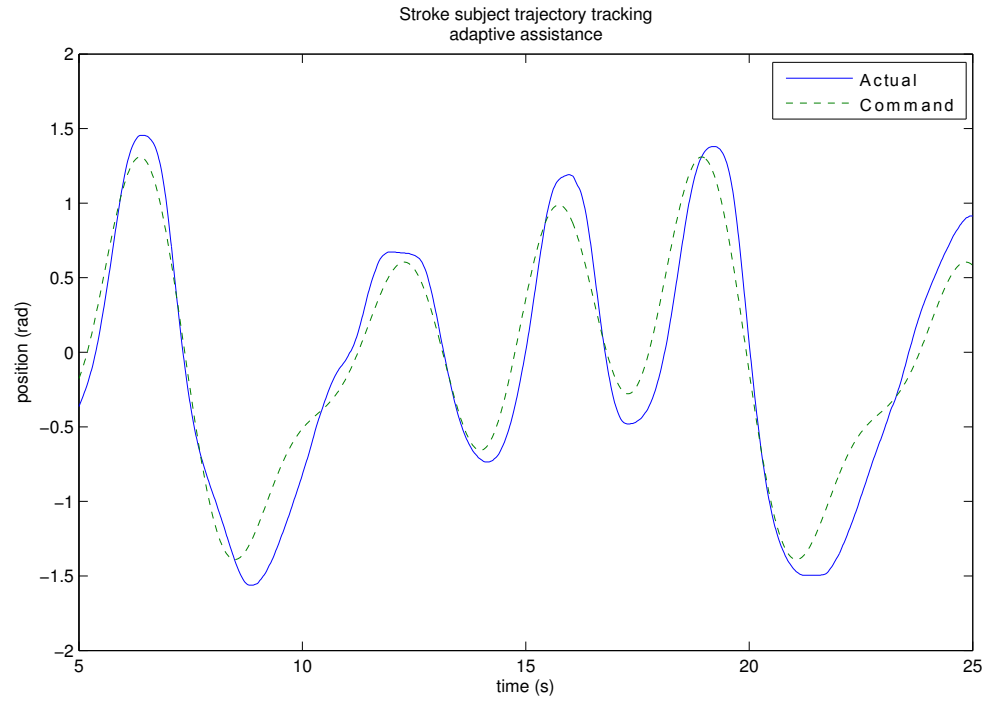


Figure 4.6: Stroke subject (2248-05) tracking trajectory in adaptive spring assist mode

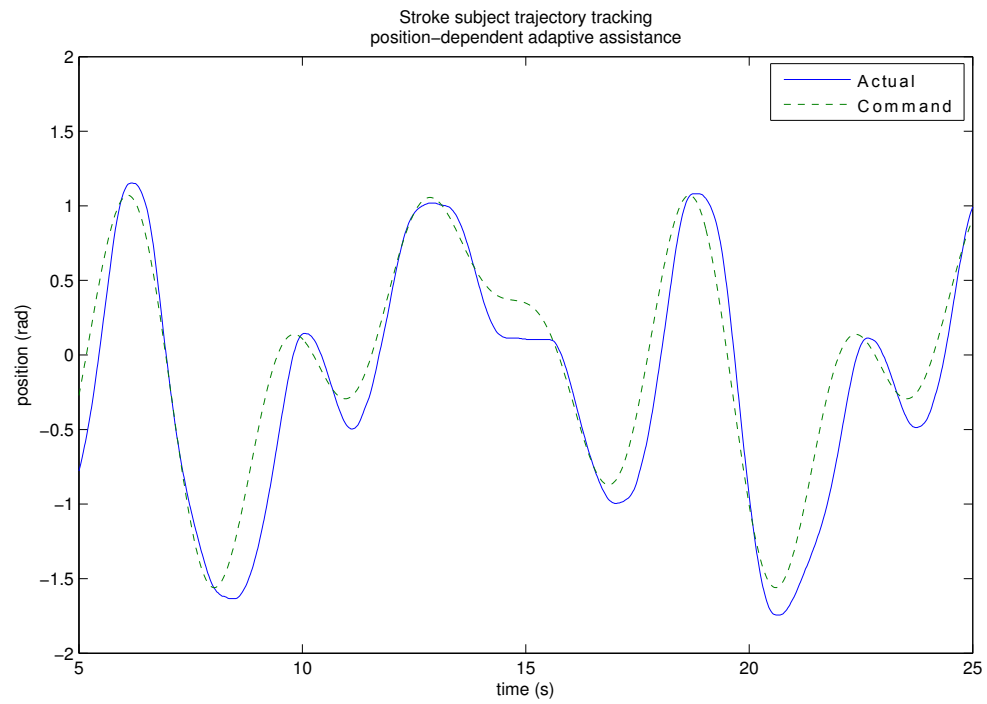


Figure 4.7: Stroke subject (2248-05) tracking trajectory in position-dependent adaptive spring assist mode

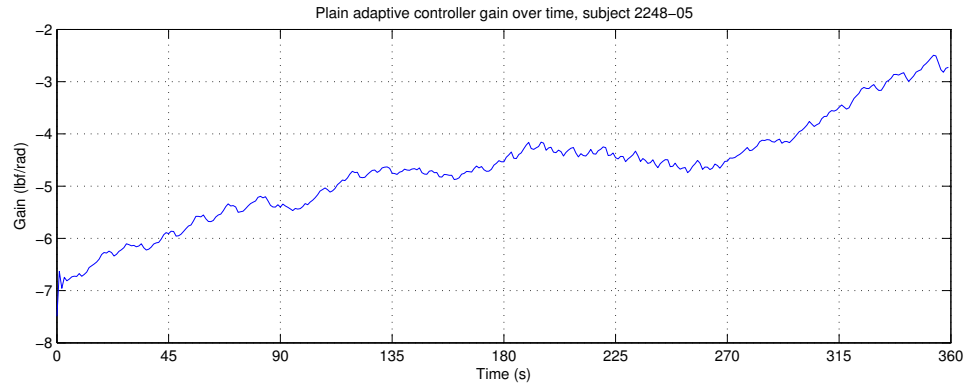


Figure 4.8: Plain adaptive controller gain plotted over time for stroke subject 2248-05

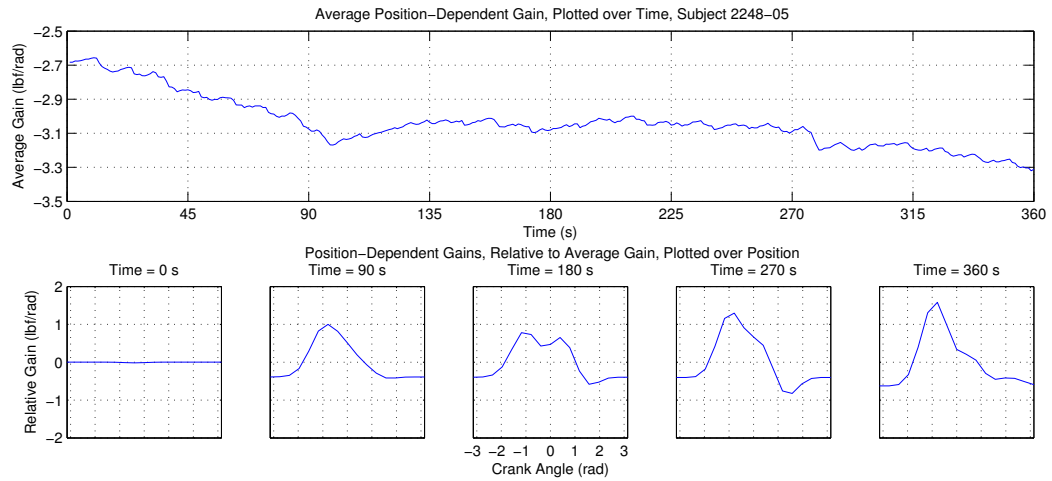


Figure 4.9: Position-dependent gains adapting over time to stroke subject 2248-05, plotted versus time (top) and position (bottom)

4.4.2 Results

Figures 4.4 through 4.7 show the lowest-functioning stroke subject (2248-05, with a Fugl-Meyer score of 29) tracking trajectories in the four operating modes of the new robot. Since the ANOVA test performed previously showed that subject performance varied between robot operating mode, t-tests could be used to compare subject performance with plain adaptive and position-dependent adaptive control to performance with nonadaptive control in exercises with the new haptic robot. A t-test showed that the RMS tracking error was significantly lower—and closer to the desired RMS error of 0.25 rad—for exercises using the plain adaptive controller versus the nonadaptive controller ($p=0.026$). However, t-tests did not show any significant difference ($p = 0.3984$, from Table 4.3) in performance when comparing the position-dependent adaptive controller to the plain adaptive controller. A t-test comparing the flatness of the RMS tracking error over position for plain and position-dependent adaptive control also did not show significant difference in the means ($p = 0.6916$). Correlation tests showed a weak negative correlation of $r = -0.5045$ between baseline RMS tracking error over position and position-dependent adaptive controller gains.

Stroke subject 2248-05 is shown tracking a trajectory with nonadaptive assistance in Figure 4.5. Through a process of trial and error, the assistive gain for this exercise was set to a value that provides the subject with enough assistance to complete the exercise but not so much assistance that the subject does not need to work to move the robot. This process is not an accurate way to find an appropriate controller gain, and it can be time-consuming, taking up to ten minutes. The plain adaptive controller is able to set appropriate gains for a subject more accurately, shown by the results of the first t-test in Table 4.3. As with the simulation controller gains, there are small oscillations present at steady state in the real gains that follow the shape of the trajectory, as seen in Figure 4.8. However, with the human subject, the overall first-order decay shape seen with the simulated subject is not apparent, and the controller does not settle to a steady-state value. This is due to two factors: the exercise trajectory was changed every 90 seconds, and the real subjects did not perform as consistently as the models.

Figure 4.9 shows the position-dependent adaptive controller adapting over time and over the robot workspace to stroke subject 2248-05. The top plot shows the average of all position-dependent gains over position plotted against time over the course of four 90-second exercises. Since each successive exercise started with the controller gains from the end of the preceding exercise, the gain data from all four exercises was concatenated seamlessly (new exercises

Table 4.3: List of t-tests performed with parameters and results of tests

Comparison	Condition 1	Condition 2	Hypothesis	p-value
RMS trajectory tracking error, haptic robot	Static stiffness	Adaptive stiffness	$2 < 1$	0.0259
RMS trajectory tracking error, haptic robot	Plain adaptive	Position-dependent	$2 < 1$	0.3984
Range of RMS error over position, haptic robot	Plain adaptive	Position-dependent	$2 < 1$	0.6916
Off-axis force, haptic robot	Static stiffness	Plain adaptive	$2 < 1$	0.2051
Off-axis force, haptic robot	Static stiffness	Position-dependent	$2 < 1$	0.0705

begin at 0, 90, 180, and 270 seconds). The bottom plots show the position-dependent gains minus the average gain plotted against position at specified points in time, allowing the shapes of the gain profiles to be compared side-by-side. Unlike the plain adaptive controller gain, the average position-dependent controller gain does settle near a steady-state value. After 90 seconds, the gain profile has also approximately reached its steady-state shape, showing a similar time constant to the average gain. Like the average gain, the position-dependent gain profile is sensitive to changes in exercise trajectory. Gains outside of ± 2 rad show no change because these regions are outside the range of positions of any trajectory that could be generated, so the subject rarely enters these regions. Profiles at 180, 270, and 360 seconds in Figure 4.9 are all at steady-state values, but each has a different shape due to the slight variations of difficulty in the randomly-generated exercise trajectories. The correlation between these gain profiles and subject RMS error over position shows a weak negative correlation of $r = -0.5045$ across all subjects, indicating that the controller does adjust gains appropriately. However, there was no significant improvement in the flatness of RMS error over position or overall RMS error from the plain adaptive controller to the position-dependent adaptive controller, with respective p-values of 0.400 and 0.617.

Off-axis forces exerted by stroke subjects in three of the four haptic robot operating modes are compared in Figure 4.11. The force ratio described in Equation 4.3 is normalized by the respective subject's baseline performance (static assist mode). The two adaptive control modes both show a wider range of off-axis force ratios versus the baseline data, and the mean baseline value falls within the ranges of values for the two adaptive modes. Plain adaptive control shows no significant decrease in off-axis force from baseline ($p = 0.2051$ from Table 4.3). This is because both methods provide assistance by the same method—a virtual linear spring that attracts the end effector to the desired position. Position-dependent adaptive control shows some decrease in

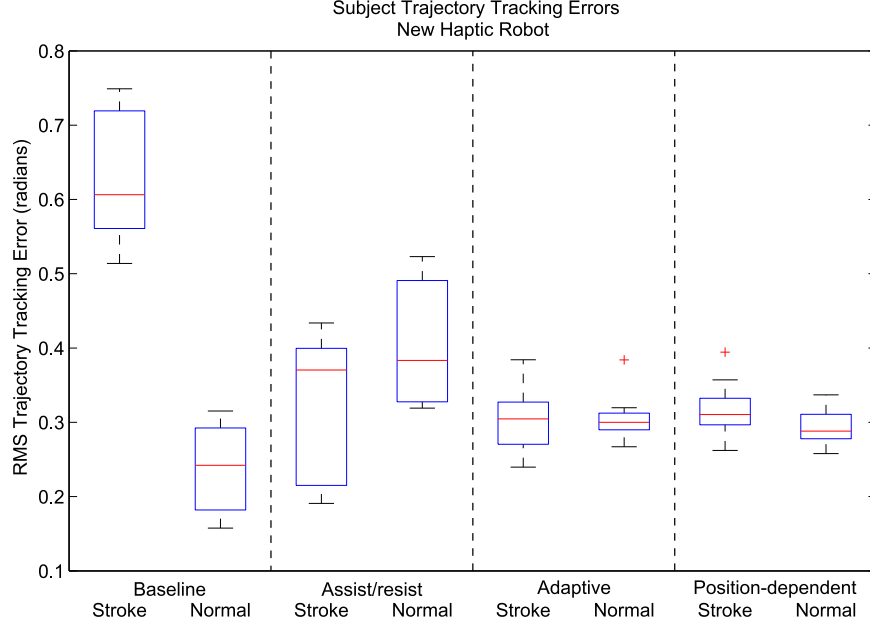


Figure 4.10: Quartile boxplot of raw RMS tracking error of subjects with the new haptic robot in all four control modes. Outliers are shown as crosses.

off-axis force, but this decrease is not statistically significant at the $p = 0.0500$ level, by a small margin ($p = 0.0705$). This could be due to the small sample size of 12 points per category and the wide range of values. The distribution of off-axis force for position-dependent mode is skewed heavily in the negative direction, as seen in the rightmost boxplot in Figure 4.11, so significant results may be produced with a larger sample size.

4.4.3 Discussion and Conclusion

The lack of significant change in RMS tracking error between plain and position-dependent adaptive control modes was not expected. The likely cause can be seen in Figure 4.10. Subject RMS tracking error with the plain adaptive controller is already at the target value of 0.25 rad, so it is not possible for the tracking error in position-dependent adaptive mode to get any closer to this value. However, controller performance can be evaluated by other metrics than subject RMS tracking error. Results of the force analysis suggest that although position-dependent adaptive control produced little change in trajectory tracking error versus plain adaptive control, position-dependent adaptive control did produce a noticeable reduction in the off-axis forces that stroke subjects exert. A reduction in off-axis forces meant that stroke subjects were exerting more productive force (force that results in motion) on the crank handle. These

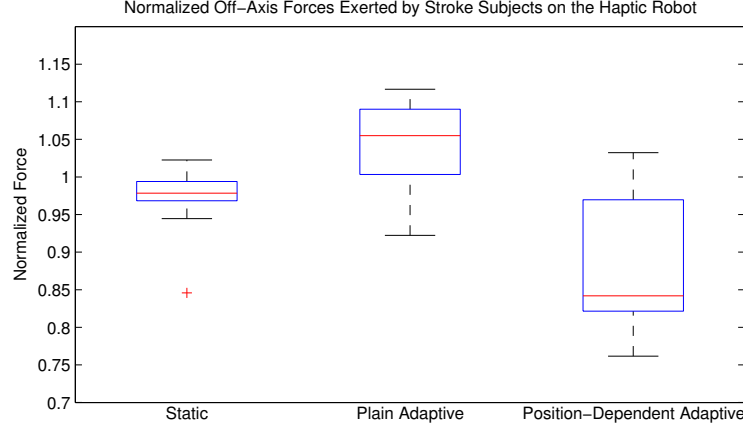


Figure 4.11: Boxplot of normalized off-axis force ratio for stroke subjects using haptic robot in three control modes. Outliers are shown as crosses.

results also suggest that the off-axis force ratio (Equation 4.3) is more sensitive to controller performance than RMS tracking error.

The weak negative correlation between position-dependent gain and position-dependent error and the lack of change in flatness of error over position from plain to position-dependent adaptive control indicate that the shape of the gain profile over space only loosely matches the error profile. There are two factors that contribute to this: trajectory boundaries and the adaptive controller's smoothing window. To prevent subjects from experiencing sudden changes in rendered stiffness, a smoothing window is applied, over space, to the adaptive controller gains. However, this smoothing window has a drawback of distorting the shape of the gain profile. Trajectory boundaries create regions where the gains of the adaptive controller cannot adapt because only the gains in regions near the end effector are changed with each iteration of the position-dependent algorithm. If the end effector is not near a certain region, then that region's gain will not change. Conversely, regions where the end effector spends a large amount of time adapt much more quickly than the boundary regions, distorting the gain profile.

4.5 Comparison of Real and Simulated Subjects

Comparing the performance of the *simulated* healthy subject in Figure 3.2 and the *real* healthy subject in Figure 4.12 reveals minor differences between the model and the real subject. The simulated subject is able to track trajectories with greater speed and accuracy than the real

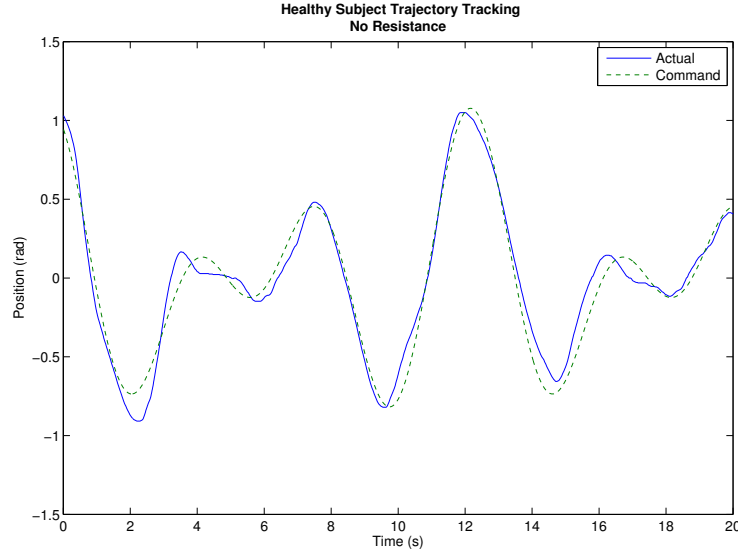


Figure 4.12: Plot of unimpaired subject (4432-01) tracking a trajectory with robot in zero-impedance mode

human. This is acceptable because the simulation allows testing to be done with a simulated subject that is stronger and faster than the most capable real subject the robot will encounter. If an adaptive controller algorithm is able to provide sufficient challenge to the simulated healthy subject, it will follow that the controller is also able to challenge any real subject.

The simulated stroke subject model in Figure 3.2 behaves nothing like the real stroke subject in Figure 4.4 when tracking trajectories in zero-impedance mode. The real stroke shows a limited range of motion and a noticeable delay of movement. Spatially, the trajectory the real stroke subject follows is very similar to the commanded trajectory when inside the subject's limited range of motion; none of the perturbations present in the simulated stroke model appear in the real subject. These differences stem from the fact that the MIT-Manus arm model was intended to simulate stroke subjects performing point-to-point movements on a tabletop. The Manus simulation model imitated the zig-zag paths that real stroke subjects took during movements between points [19]. Using this model with a crank constraint and a pre-defined trajectory did not produce behavior seen in real stroke subjects because the model was not being used for its intended purpose.

CHAPTER 5 CONCLUSION AND FUTURE DIRECTIONS

5.1 General Conclusion

Building upon research done by M. Johnson, X. Feng, L. Johnson, et al. with the Driver's SEAT, UniTherapy, TheraJoy, and TheraDrive, a new haptic robot for post-stroke rehabilitation was created [6, 9, 8, 11]. The new haptic robot has been proven to be a viable replacement for the Logitech wheel in the TheraDrive system, for stroke subjects who require assistance to complete tasks. Stroke-impaired subjects were able to perform tasks effectively with the robot, and they showed favorable opinions of it, unanimously indicating the new haptic robot to be preferred over the original TheraDrive system. Experimental data shows the adaptive controller to be able to tailor the difficulty of exercises to the ability level of subjects. Data also suggests that the adaptive controller also decreases the amount of unproductive force that subjects exert on the robot during exercises. This demonstrates that the first and second specific aims of the project have been satisfied, as the haptic robot is able to exert sufficient force to support all subjects and is able to provide custom-tailored support to subjects that is superior to that offered by the original TheraDrive system. However, this system is not recommended for use with high-functioning stroke survivors who would be working with the robot in resistive mode, as subjects found the robot difficult to control when resistance was added.

A significant source of error in the experiments performed was the small sample sizes. Only 2 simulated, 2 unimpaired, and 4 stroke subjects were used for the experiments performed. The data collected is insufficient to prove that the haptic robot is an effective therapy tool, but the successes of these preliminary experiments are enough to merit larger trials with the system. Another source of error is the lack of a procedure for determining assistive/resistive gains for TheraDrive and the haptic robot in the static spring assist/resist mode. Gains were chosen by a guess-and-check method until the investigator and the subject agreed that the exercise looked and felt correct. This method was very subjective, and it may have made some data unreliable.

5.2 Contributions

- A portable high-force haptic system was created for use as a post-stroke rehabilitation robot.
- The robot implements a novel adaptive controller that is able to monitor subject performance and adjust the assistance provided to the subject on an assist-as-needed basis in real-time.

- A detailed simulation environment was written to allow prototyping of robot controllers that could be run on simulated or real hardware without modification.

5.3 Future Directions

Safety concerns from the unimpaired subjects regarding the resistive behavior of the haptic robot need to be addressed. When resisting subjects, the robot produces large forces that reverse suddenly, making it very difficult for the subjects to control. The method of providing resistive force must be revised to prevent these reversals of force, some of which were as large as 36 lb peak-to-peak.

Since the behavior of the simulated and real stroke subjects showed little agreement, a new stroke model needs to be developed for this system. This model needs to include delayed movements and a limited range of motion like that seen in the real stroke subjects. Also needed is the ability to model subjects who always have tone in some muscles, which causes them great difficulty moving a joint in one direction but not the opposite direction; e. g., tone in the biceps makes elbow extension more difficult than flexion.

The position-dependent adaptive controller contains several parameters that affect its performance including adaptation gain, RMS error timespan, spatial smoothing window, and number of regions. These parameters were constant for all experiments at values found to work in simulations. A model of how each parameter affects subject performance and controller performance should be written. This would aid in tuning the controller to work optimally for real subjects.

Direction-dependent adaptive control would be another possible control strategy to investigate. It would function in a similar manner to the two current adaptive control strategies, but it would have a different set of gains for each direction of motion. This would be useful for the stroke subjects previously mentioned who have difficulty moving joints in one direction but not the other.

BIBLIOGRAPHY

- [1] N. F. Gordon, M. Gulanick, F. Costa, G. Fletcher, B. A. Franklin, E. J. Roth, and T. Shephard, "Physical activity and exercise recommendations for stroke survivors: An american heart association scientific statement from the council on clinical cardiology, subcommittee on exercise, cardiac rehabilitation, and prevention; the council on cardiovascular nursing; the council on nutrition, physical activity, and metabolism; and the stroke council," *Stroke*, vol. 35, pp. 1230–1240, May 2004.
- [2] M. L. Aisen, H. I. Krebs, N. Hogan, F. McDowell, and B. Volpe, "The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke," *Archives of Neurology*, vol. 54, pp. 443–446, April 1997.
- [3] H. I. Krebs, M. Ferraro, S. P. Buerger, M. J. Newbery, A. Makiyama, M. Sandmann, D. Lynch, B. T. Volpe, and N. Hogan, "Rehabilitation robotics: pilot trial of a spatial extension for mit-manus," *Journal of Neuroengineering and Rehabilitation*, vol. 1, October 2004.
- [4] T. Nef, M. Mihelj, and R. Riener, "Armin: a robot for patient-cooperative arm therapy," *Medical & biological engineering & computing*, vol. 45, pp. 887–900, August 2007.
- [5] P. Staubli, T. Nef, V. Klamroth-Marganska, and R. Riener, "Effects of intensive arm training with the rehabilitation robot armin ii in chronic stroke patients: four single-cases," *Journal of Neuroengineering and Rehabilitation*, vol. 6, p. 46, December 2009.
- [6] M. J. Johnson, H. F. M. Van der Loos, C. G. Burgar, P. Shor, and L. J. Leifer, "Experimental results using force-feedback cueing in robot-assisted stroke therapy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, pp. 335–348, September 2005.
- [7] S. R. Wood, N. Murillo, P. B. y Rita, R. S. Leder, J. T. Marks, and S. J. Page, "Motivating, game-based stroke rehabilitation: A brief report," *Topics in Stroke Rehabilitation*, vol. 10, pp. 134–140, October 2003.
- [8] M. J. Johnson, X. Feng, L. M. Johnson, and J. M. Winters, "Potential of a suite of robot/computer-assisted motivating systems for personalized, home-based, stroke rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 3, p. 29, March 2007.
- [9] X. Feng and J. M. Winters, "A pilot study evaluating use of a computer-assisted neurorehabilitation platform for upper-extremity stroke assessment," *Journal of NeuroEngineering and Rehabilitation*, May 2009.
- [10] D. J. Reinkensmeyer, C. T. Pang, J. A. Nessler, and C. C. Painter, "Web-based telerehabilitation for the upper extremity after stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 102–108, 2002.
- [11] R. Ruparel, M. J. Johnson, E. Strachota, J. McGuire, and G. Tchekanov, "Evaluation of the theradrive system for robot/computer assisted motivating rehabilitation after stroke," in *31st Annual International IEEE EMBS Conference*, pp. 811–814, April 2009.
- [12] M. D. Rinderknecht, "Design of a demo experimental setup for human augmentation," tech. rep., École Polytechnique Fédérale de Lausanne, 2011.
- [13] R. van Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber, "Compliant actuator designs: Review of actuators with passive adjustable compliance/controllable stiffness for robotic applications," *IEEE Robotics & Automation Magazine*, vol. 16, pp. 81–94, September 2009.

- [14] S. Wolf and G. Hirzinger, "A new variable stiffness design: Matching requirements of the next robot generation," in *IEEE International Conference on Robotics and Automation*, pp. 1741–1746, May 2008.
- [15] I. Vanderniepen, R. van Ham, J. Naudet, M. van Damme, B. Vanderborght, R. Versluys, and D. Lefeber, "Novel compliant actuator for safe and ergonomic rehabilitation robots design of a powered elbow orthosis," in *IEEE 10th International Conference on Rehabilitation Robotics*, pp. 790–797, June 2007.
- [16] X. Dai, Y. Zhang, and D. Wang, "Maximum virtual stiffness distribution analysis and measurement for haptic device," in *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 915–921, August 2009.
- [17] J. L. Emken, R. Benitez, and D. J. Reinkensmeyer, "Human-robot cooperative movement training: Learning a novel sensory motor transformation during walking with robotic assistance-as-needed," *Journal of Neuroengineering and Rehabilitation*, vol. 4, p. 8, March 2007.
- [18] L. Masia, M. Casadio, P. Giannoni, G. Sandini, and P. Morasso, "Performance adaptive training control strategy for recovering wrist movements in stroke patients: a preliminary, feasibility study," *Journal of Neuroengineering and Rehabilitation*, vol. 6, p. 44, December 2009.
- [19] D. Formica, L. Zollo, and E. Guglielmelli, "Torque-dependent compliance control in the joint space of an operational robotic machine for motor therapy," in *IEEE 9th International Conference on Rehabilitation Robotics*, pp. 341–344, June 2005.
- [20] A. R. Theriault, M. L. Nagurka, and M. J. Johnson, "A robust wheel interface with a novel adaptive controller for computer/robot-assisted motivating rehabilitation," in *ASME International Symposium on Flexible Automation*, June 2012.
- [21] A. R. Fugl-Meyer, L. Jääskö, I. Leyman, S. Olsson, and S. Steglind, "The post-stroke hemiplegic patient: A method for evaluation of physical performance," *Scandinavian journal of rehabilitation medicine*, vol. 7, no. 1, p. 13, 1975.

APPENDIX A EXPERIMENTAL PROCEDURE

A.1 Procedure

After arrival, subjects will review and sign the consent form with study personnel. Stroke subjects will be evaluated by a therapist to determine their level of impairment using Fugl-Meyer scores. Healthy subjects will not need to be evaluated. Subjects will then be instructed in the use of the robot and its safety features.

Subjects will perform therapy exercises in random order, decided prior to the session. It will be randomly determined which robot the patient will use first, the order of operating modes used for the new robot, and the order of therapy exercises performed in each mode. The robots and their operating modes are listed as follows:

1. TheraDrive

- No-force mode – The wheel of the TheraDrive system will be allowed to rotate freely. No assistive/resistive forces will be applied.
- Force-feedback mode – TheraDrive will provide assistive/resistive forces at a level set prior to each exercise.

2. New Robot

- Zero-impedance mode – The crank arm will feel as if it can rotate freely, though it will be actuated. No assistive/resistive forces will be applied.
- Static control mode – The crank will provide assistive/resistive forces at a level set prior to each exercise.
- Adaptive control modes – The crank will provide assistive/resistive forces. The robot controller will measure subject performance in real-time and adjust the assistive/resistive forces accordingly to maintain a certain level of difficulty.

Exercises consist of trajectory-following tasks. Trajectory-following exercises present the patient with a cursor that follows the robot end effector and a target that moves along a path. The objective of the exercise is to keep the cursor on the target for the duration of the exercise.

Subjects will perform a 15-minute series of exercises in random order with each robot in each operating mode, always starting with the no-force or zero-impedance mode. After each exercise, subjects will be asked to rate their level of exertion and level of discomfort on a scale of 1 to 10.

Subject will be asked to complete a survey after completing work with each robot. Survey will contain questions regarding ease of use, perceived utility, perceived safety, confidence during use of robot. Subjects will also be asked what they would like improved with each robot and will be asked to indicate their preferred robot and the strength of that preference.

Outline of Visit:

- Consent (15 minutes)
- Evaluation, collect Fugl-Meyer score/Ashworth (45 minutes)
- Training (15 minutes)
- Subject is randomly assigned to use TheraDrive or new robot first
- TheraDrive:
 - For each task, RMS tracking error, exertion level, and discomfort level are recorded.
 - No-force tasks (15 minutes)
 - Force-feedback tasks (15 minutes)
 - Questionnaire regarding subject's overall opinion of TheraDrive and exercises (15 minutes)
- Break (15 minutes): Surveys
- New robot:
 - For each task, detailed force and motion data, exertion level, and discomfort level are recorded.
 - Zero-impedance tasks (15 minutes)
 - Static control tasks (15 minutes)
 - Adaptive control tasks (30 minutes)
 - Questionnaire regarding subject's overall opinion of new robot and exercises (15 minutes)
- Final questionnaire regarding subject's preference of robot (15 minutes)

A.2 Motivation Survey

Subjects were asked to rate their agreement with the following statements on a scale of 1 (strongly disagree) to 7 (strongly agree). These responses are categorized by Table A.1, and responses in each category are averaged to produce a set of motivation scores for each subject. For questions with inverted scoring, a response of 7 gives a score of 1, etc.

1. I would be willing to do this again because it has value to me.
2. I enjoyed doing this activity.
3. I think I am good at this activity.
4. I put a lot of effort into this activity.
5. I did not feel nervous at all while participating.
6. I'd like a chance to interact with this system more often.
7. I did this activity because I had no choice.
8. I think doing this activity could help me to concentrate more.
9. This activity was fun to do.
10. I am satisfied with my performance at this task.
11. It was important to me to do well at this task.
12. I was very relaxed in doing this.
13. I felt distant from the system when interacting with it.
14. I believe doing this activity could be beneficial to me.
15. I would describe this activity as very interesting.
16. I was skilled at this activity.
17. I didn't put much energy into this.
18. I was anxious while working on this task.
19. I felt close to the system when interacting with it.

20. I did this activity because I wanted to.
21. I think this is an important activity.
22. This is an activity that I couldn't do very well.
23. I tried very hard to complete this activity.
24. I felt pressured while doing this.
25. I'd really prefer not to interact with this system in the future.

Table A.1: Motivation survey question categories and scoring

Question	Category	Invert?
1	Value/Usefulness	normal
2	Interest/Enjoyment	normal
3	Perceived competence	normal
4	Effort/Importance	normal
5	Pressure/Tension	normal
6	Relatedness	normal
7	Perceived Choice	inverted
8	Value/Usefulness	normal
9	Interest/Enjoyment	normal
10	Perceived competence	normal
11	Effort/Importance	normal
12	Pressure/Tension	normal
13	Relatedness	inverted
14	Value/Usefulness	normal
15	Interest/Enjoyment	normal
16	Perceived competence	normal
17	Effort/Importance	inverted
18	Pressure/Tension	inverted
19	Relatedness	normal
20	Perceived Choice	normal
21	Value/Usefulness	normal
22	Perceived competence	inverted
23	Effort/Importance	normal
24	Pressure/Tension	inverted
25	Relatedness	inverted

A.3 Post-Activity Survey

POST-ACTIVITY-SCREENING FORM

A robust wheel interface with a novel adaptive controller for computer/robot-assisted motivating rehabilitation

Subject Number: _____

Date: ____/____/____

Time: ____:____

1. How will you rate your overall experience? _____ (on the scale of 1-10).
2. How tired are you? _____ (on the scale of 1-10).
3. How hard do you think you worked? _____ (on the scale of 1-10).
4. How well could you see the information on the screen? _____ (on the scale of 1-10).
5. How well did you understand the instructions given to you? _____ (on the scale of 1-10).
6. Which part of the experiment according to you was toughest?

7. Do you have any comments about the device and software used?

8. Do you have any comments about the procedure?

9. Is there any extra information you would like to share?

Page 1 of 1

Figure A.1: Post-Activity Survey.

A.4 Final Survey

<p>FINAL ACTIVITY-SCREENING FORM</p> <p><i>A robust wheel interface with a novel adaptive controller for computer/robot-assisted motivating rehabilitation</i></p> <p>Subject Number: _____</p> <p>Date: ____/____/____</p> <p>Time: ____:____</p> <ol style="list-style-type: none"> 1. How will you rate your overall experience? _____ (on the scale of 1-10) 2. Which robot do you prefer? _____ 3. How strongly do you prefer this robot? _____ (on the scale of 1-10) 4. Why do you prefer this robot? What do you like about it? _____ _____ 5. What would you change about this robot? _____ _____ 6. Why do you not prefer the other robot? _____ _____ 7. Provide any additional comments here. _____ _____
<p>Page 1 of 1</p>

Figure A.2: Final Survey

APPENDIX B RAW EXPERIMENTAL DATA

Table B.1: Raw RMS tracking errors in radians for subjects using TheraDrive joystick

Exercise	Trial	2248-01	2248-04	2248-05	2248-06	2248-07
free	1	0.2355	0.3150	0.1649	0.1602	0.1931
free	2	0.2329	0.2862	0.2533	0.2312	0.1605
free	3	0.1964	0.2220	0.2073	0.2149	0.1149
free	4	0.2759	0.2321	0.1727	0.1795	0.1190
spring	1	0.1901	0.1602	0.1141	0.2803	0.1493
spring	2	0.1308	0.2200	0.1156	0.1989	0.3159
spring	3	0.1784	0.2202	0.1746	0.2513	0.1750
spring	4	0.1914	0.1223	0.1818	0.2343	0.2040

Table B.2: Raw RMS tracking errors in radians for subjects using Haptic Robot

Exercise	Trial	2248-01	2248-03	2248-05	2248-06	2248-07
zero-imp	1	0.1575	0.7297	0.7182	0.5437	0.3152
zero-imp	2	0.2725	0.6873	0.5646	0.5572	0.3121
zero-imp	3	0.1820	0.7203	0.6351	0.5778	0.2421
zero-imp	4		0.7490	0.5700	0.5139	
static	1	0.3361	0.4017	0.1973	0.4337	0.4927
static	2	0.3191	0.3654	0.2289	0.3973	0.5231
static	3		0.3470	0.2012	0.3758	0.4304
static	4		0.3754	0.1909	0.4202	0.4889
adaptive	1	0.2670	0.3380	0.2436	0.3483	0.3046
adaptive	2	0.2954	0.2825	0.2690	0.3842	0.3052
adaptive	3	0.2854	0.2719	0.3163	0.3046	0.2945
adaptive	4	0.3841	0.3094	0.2396		0.3196
pos-adapt	1	0.2579	0.2621	0.3945	0.3085	0.2875
pos-adapt	2	0.2751	0.2983	0.3123	0.3106	0.2888
pos-adapt	3	0.2805	0.3327	0.3104	0.2950	0.2972
pos-adapt	4	0.3369	0.2657	0.3571	0.3320	0.3244

Table B.3: Raw off-axis force ratio for stroke subjects using Haptic Robot

Exercise	Trial	2248-03	2248-05	2248-06
static	1	0.6798	0.6944	0.7365
static	2	0.7174	0.5971	0.7927
static	3	0.7220	0.6795	0.8111
static	4	0.7048	0.6187	0.8140
adaptive	1	0.6910	0.6413	0.6401
adaptive	2	0.6873	0.7083	0.6923
adaptive	3	0.6905	0.6580	0.6615
adaptive	4	0.6875	0.8081	0.6615
pos-adapt	1	0.6986	0.7032	0.6555
pos-adapt	2	0.5971	0.7230	0.6304
pos-adapt	3	0.6668	0.6766	0.6663
pos-adapt	4	0.6944	0.6864	0.6006

Table B.4: Motivation survey scores for subjects using TheraDrive joystick

Category	2248-01	2248-03	2248-04	2248-05	2248-06	2248-07
Value/Usefulness	5.00	5.00	6.75	6.75	6.25	2.75
Interest/Enjoyment	7.00	4.67	7.00	7.00	6.00	3.00
Perceived competence	5.00	2.00	5.00	5.00	4.50	5.25
Effort/Importance	5.25	4.75	7.00	6.00	4.25	4.00
Pressure/Tension	7.00	4.75	5.50	7.00	4.75	4.25
Relatedness	5.25	4.75	6.50	5.50	5.75	3.75
Perceived Choice	7.00	7.00	7.00	7.00	6.00	6.00

Table B.5: Motivation survey scores for subjects using Haptic Robot

Category	2248-01	2248-03	2248-05	2248-06	2248-07
Value/Usefulness	5.75	6.50	6.75	6.00	4.25
Interest/Enjoyment	7.00	5.33	7.00	6.00	3.67
Perceived competence	6.00	3.50	4.75	4.25	3.50
Effort/Importance	6.00	4.50	6.00	5.00	4.75
Pressure/Tension	5.25	6.25	5.50	5.75	4.50
Relatedness	5.50	4.75	4.75	6.00	4.50
Perceived Choice	7.00	7.00	7.00	6.50	6.50

Table B.6: Post-activity survey results

Category	2248-01	2248-03	2248-04	2248-05	2248-06	2248-07
Post-activity Theradrive overall	7	3	10	10	9	7
Post-activity Theradrive effort	3	1	3	8	1	2
Post-activity Theradrive fatigue	4	1	2	7	1	1
Post-activity haptic robot overall	9	5		10	8	7
Post-activity haptic robot effort	6	1		7	2	2
Post-activity haptic robot fatigue	6	1		7	8	4
Haptic robot preference	0	+10		+8	+5	-5
TheraDrive preference	0	-10		-8	-5	+5

APPENDIX C BILL OF MATERIALS

Table C.1: Bill of Materials

Part	Qty.	Source
Ruland JC16 jaw coupler hub	1	McMaster-Carr 9845T202
Ruland JD16 yellow spider	1	McMaster-Carr 9845T14
16mm to 5/8" rigid shaft coupler	1	McMaster-Carr 61005K133, modified
5/8" telescoping shaft collar	1	McMaster-Carr 9666T3
1/2" cam follower	1	McMaster-Carr 1460T11
5/8" thrust bearings	2	McMaster-Carr 5909k32, 5909k45
17mm thrust bearing	1	McMaster-Carr 5909k14, 5909k74
Nippon Bearing SSPF20 ball spline nut	1	McMaster-Carr 61145k46
Nippon Bearing SSP 20 spline shaft	1	direct from manufacturer
Anaheim Automation GBPH0602-NP-40 gearbox	1	direct from manufacturer
Red Lion ZPJ1000A encoder	1	Newark 19P7200
6203 series bearing	2	American Science and Surplus
8mm ID bearing	2	American Science and Surplus
A-M-C 30A20AC servo amp	1	direct from manufacturer
2" dia. compression spring	1	McMaster-Carr 96485K131
2" dia. spring guard	1	McMaster-Carr 1878K57
Turdan Industry treadmill motor	1	Treadmill Doctor
5/8" keyed shaft	8"	standard part
5/32" shaft	3"	standard part
4" by 5/8" aluminum bar stock	72"	standard part
3" dia. aluminum rod stock	8"	standard part
1.25" square aluminum bar stock	8"	standard part
1" by 1" by 1/8" aluminum L stock	96"	standard part
1" square by 1/8" aluminum square tube stock	12"	standard part
Burr-Brown INA125 amplifier with reference IC	1	Digikey
Burr-Brown INA2126 dual amplifier IC	2	Digikey
100k Ω multi-turn trimmer pots	4	standard part
Vishay 062LW strain gauges	8	direct from manufacturer
350 $\Omega \pm 0.02\%$ precision resistors	2	Vishay

APPENDIX D MECHANICAL DRAWINGS

D.1 Drawings

Dimensions are in inches unless otherwise specified.

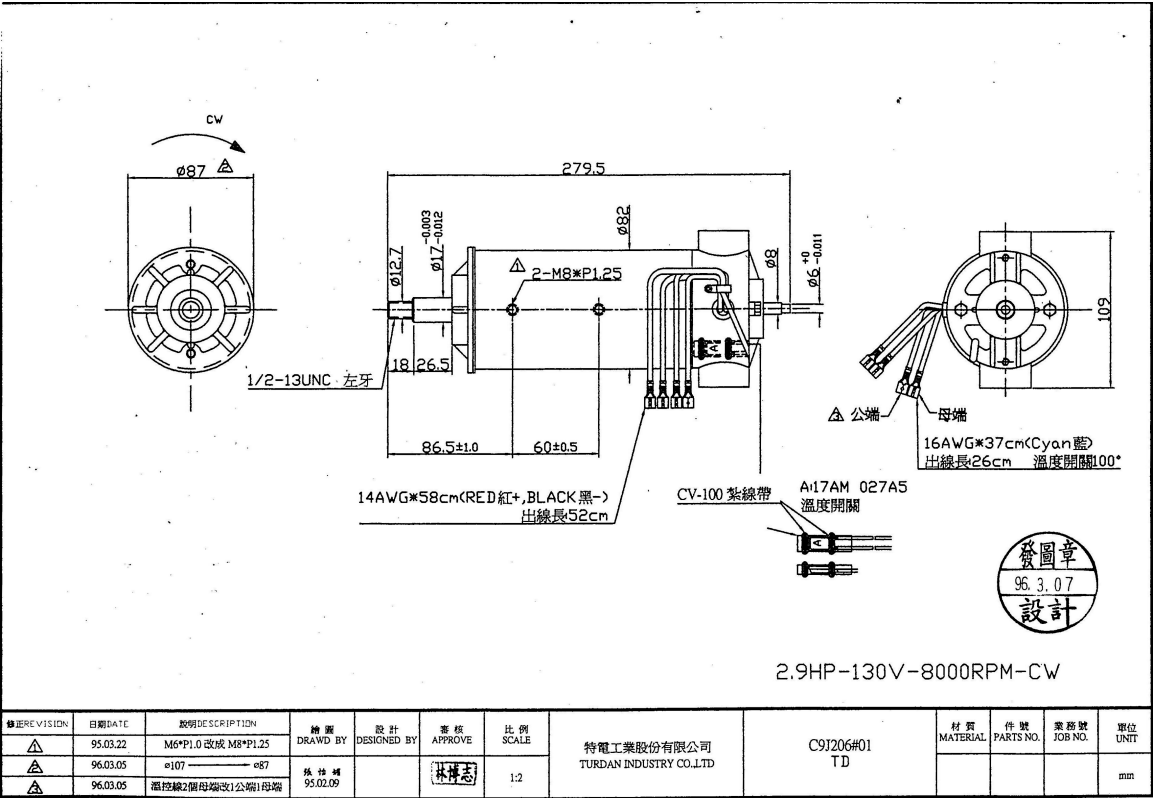


Figure D.1: Treadmill motor drawing, courtesy of Turdan Industry Co. Dimensions in mm.

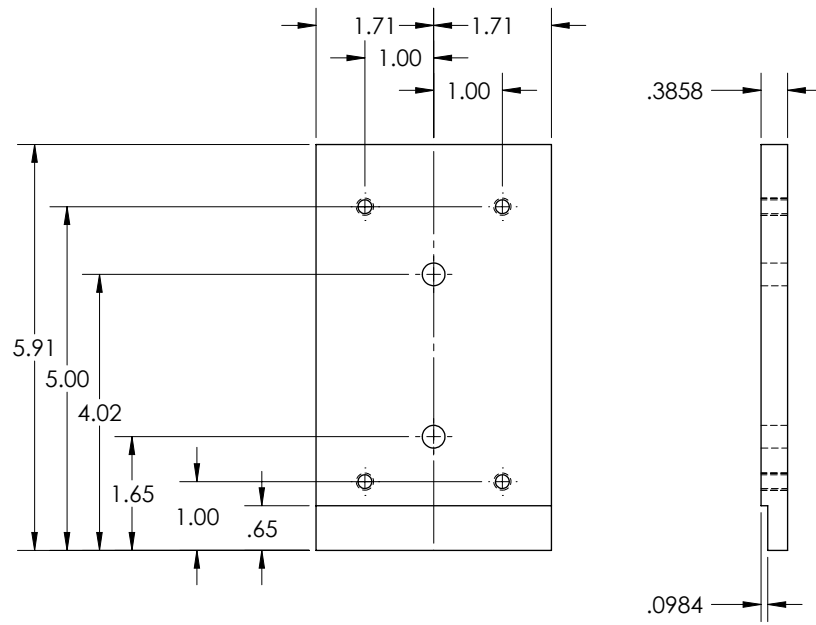


Figure D.2: Motor mounting block.

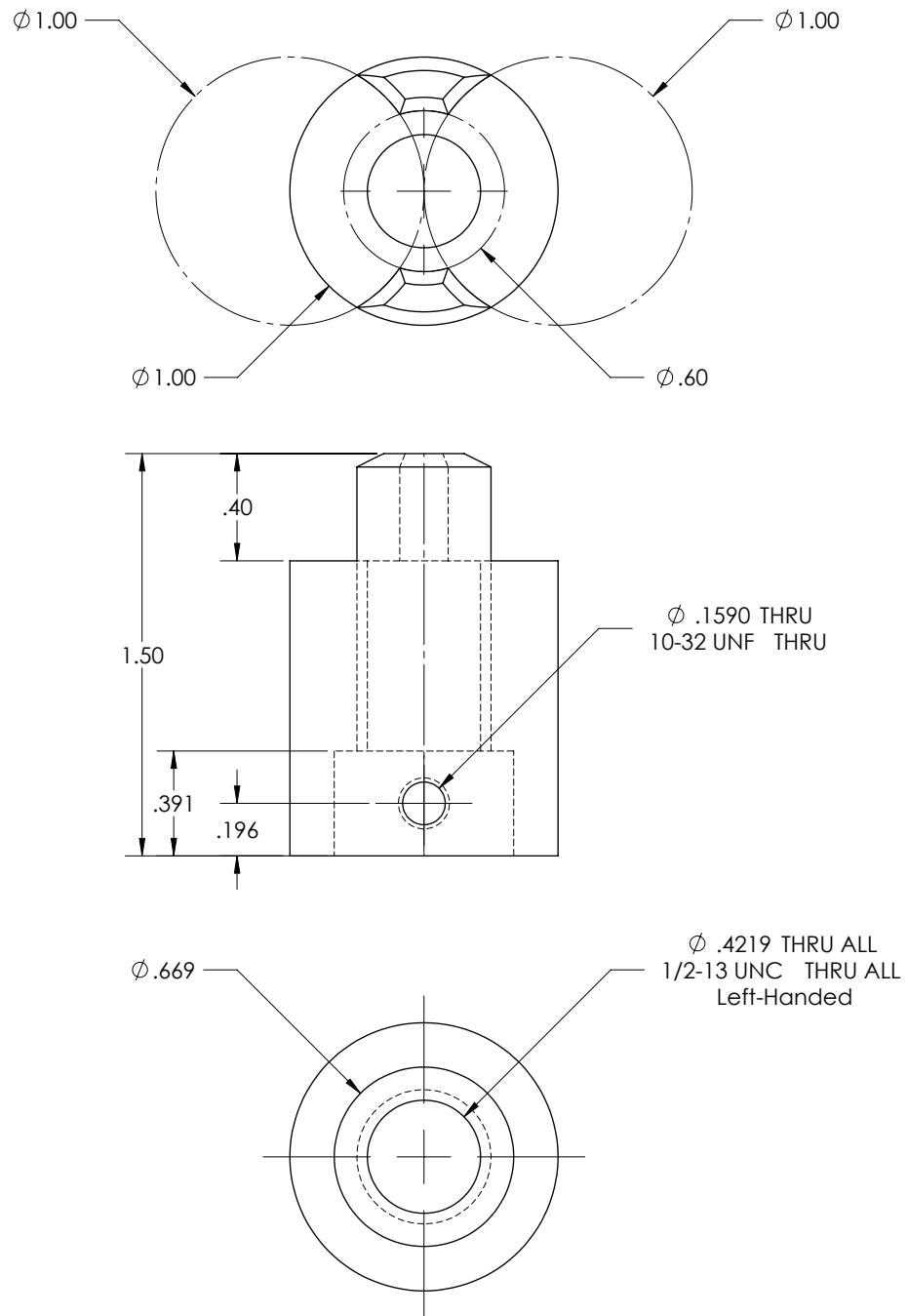


Figure D.3: Spider coupler for motor output.

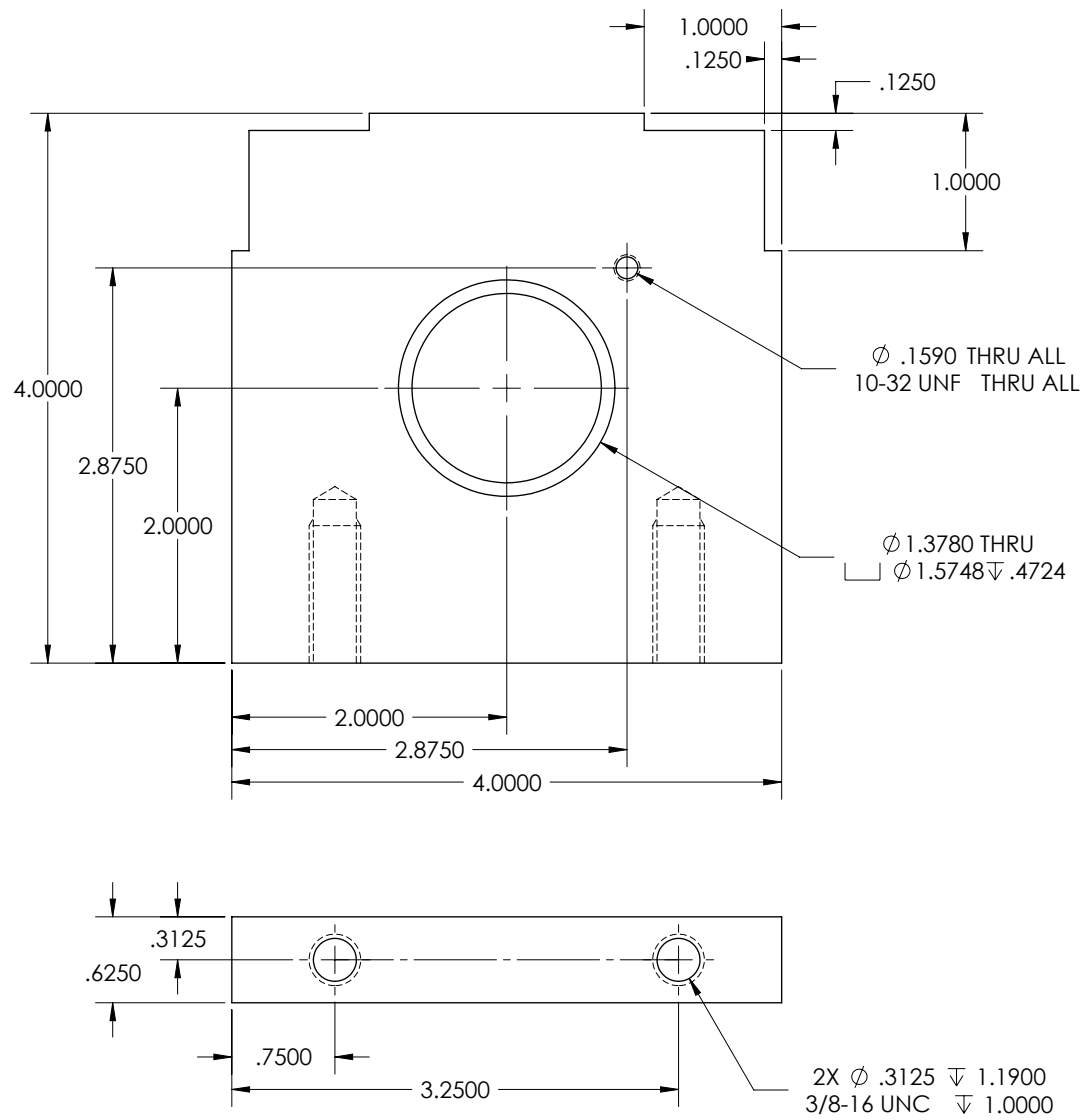


Figure D.4: Vertical mounting block 1 – holds 6203 bearing and encoder.

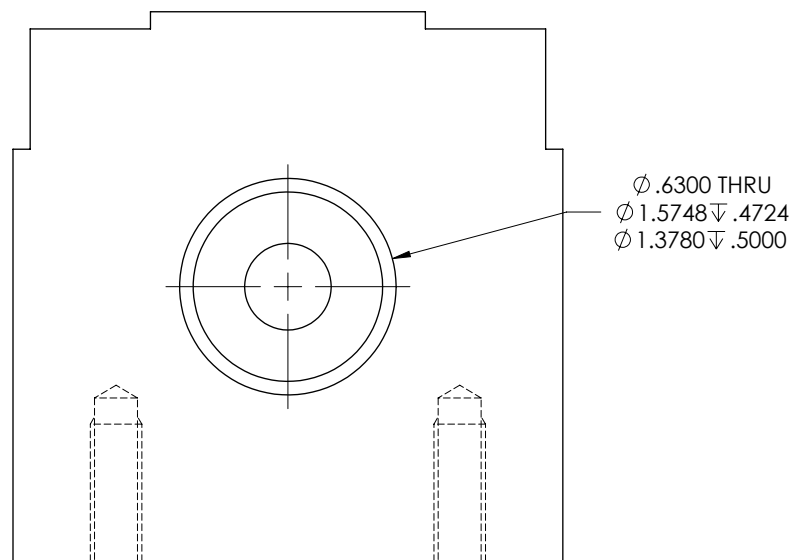


Figure D.5: Vertical mounting block 2 – holds 6203 bearing and thrust bearing.

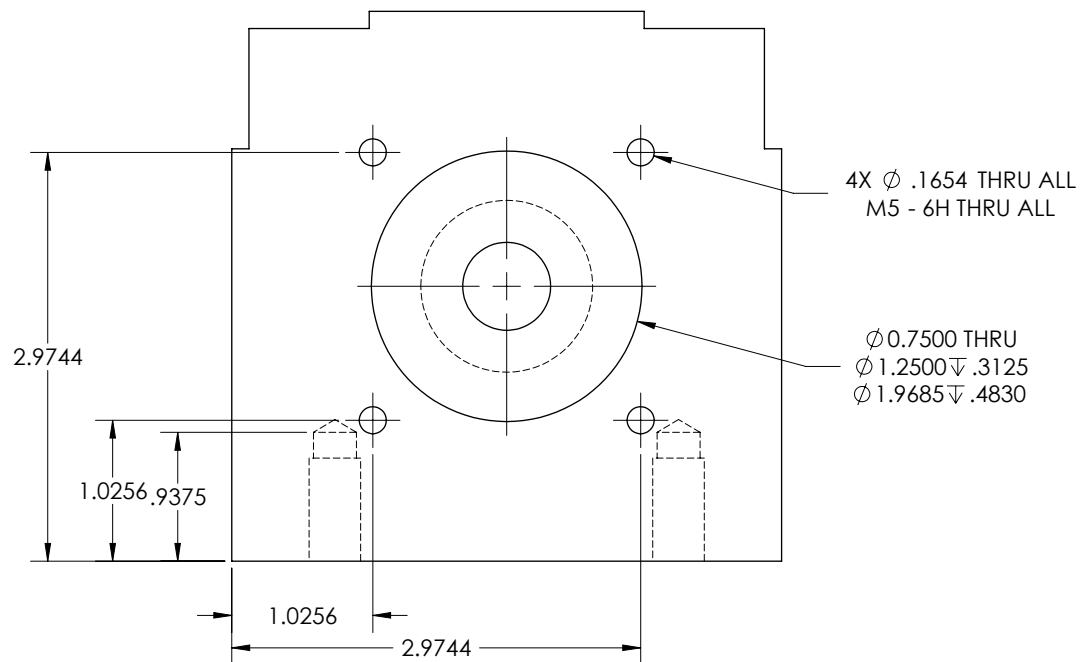


Figure D.6: Vertical mounting block 3 – holds gearbox and thrust bearing.

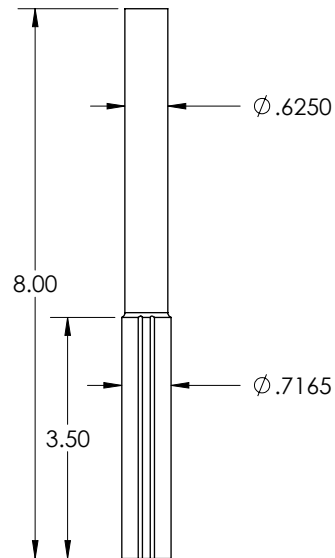


Figure D.7: Ball spline shaft.

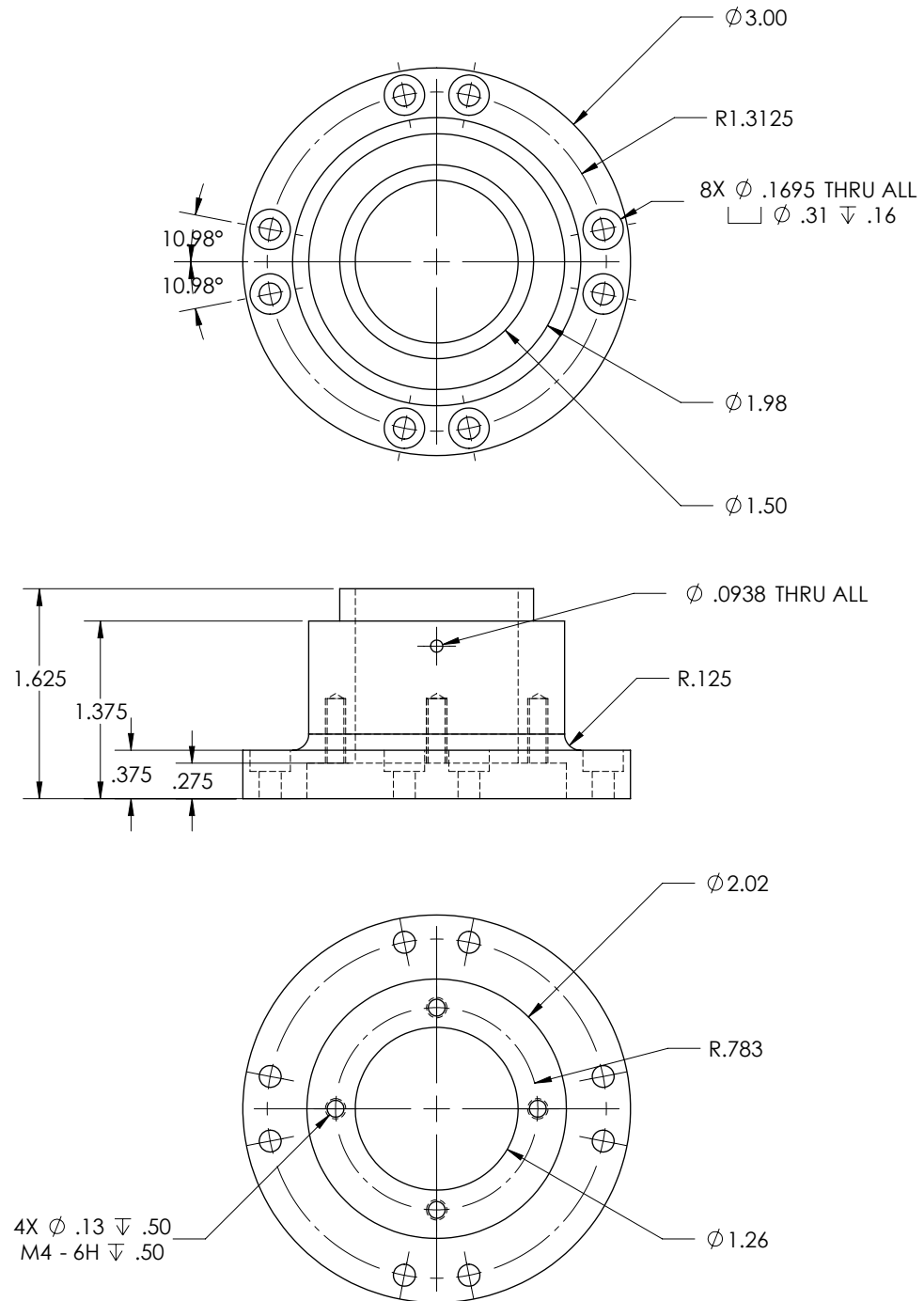


Figure D.8: Cam holder – mounts to spline nut.

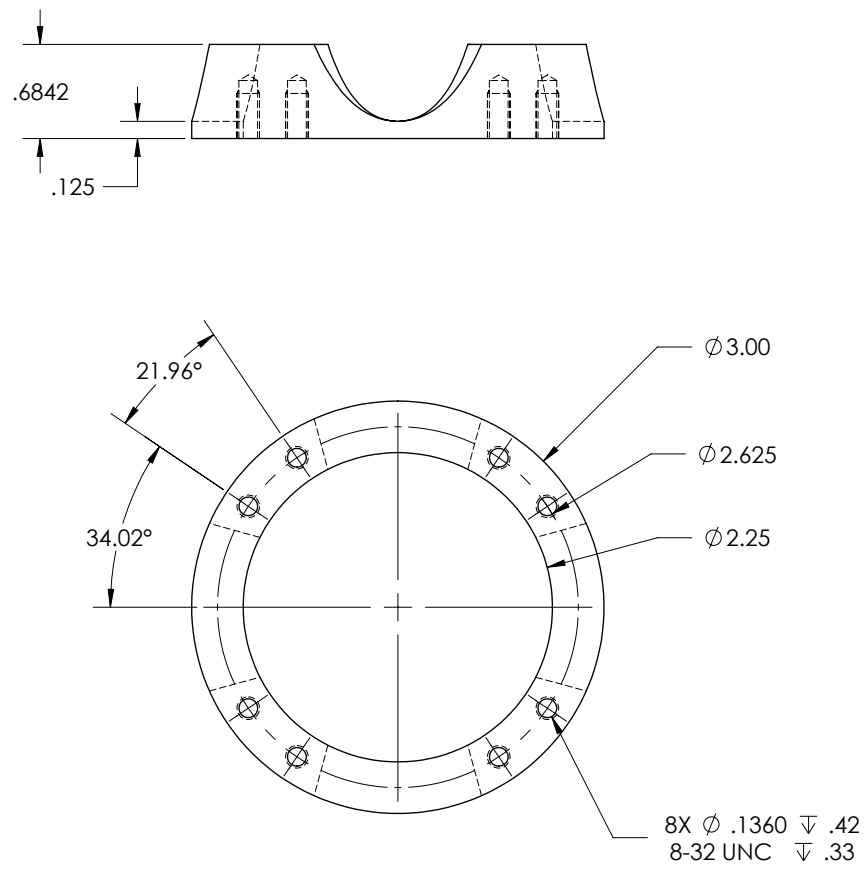


Figure D.9: Cam surface – mounts to holder.

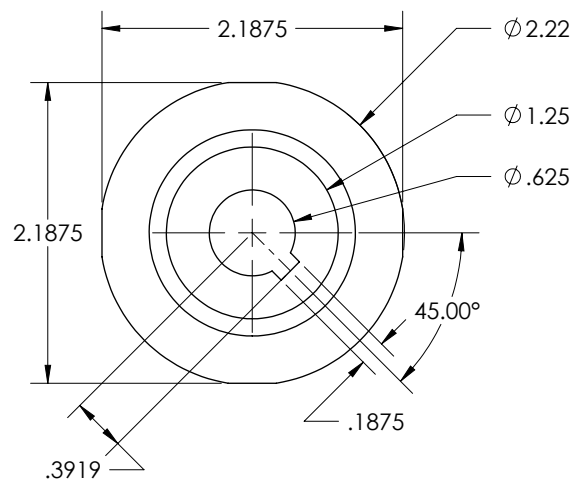
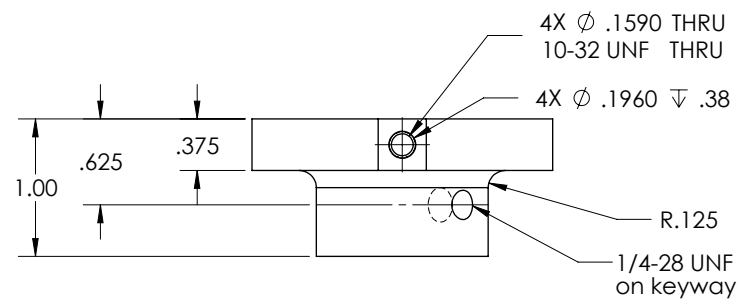


Figure D.10: Follower carrier – holds cam followers.

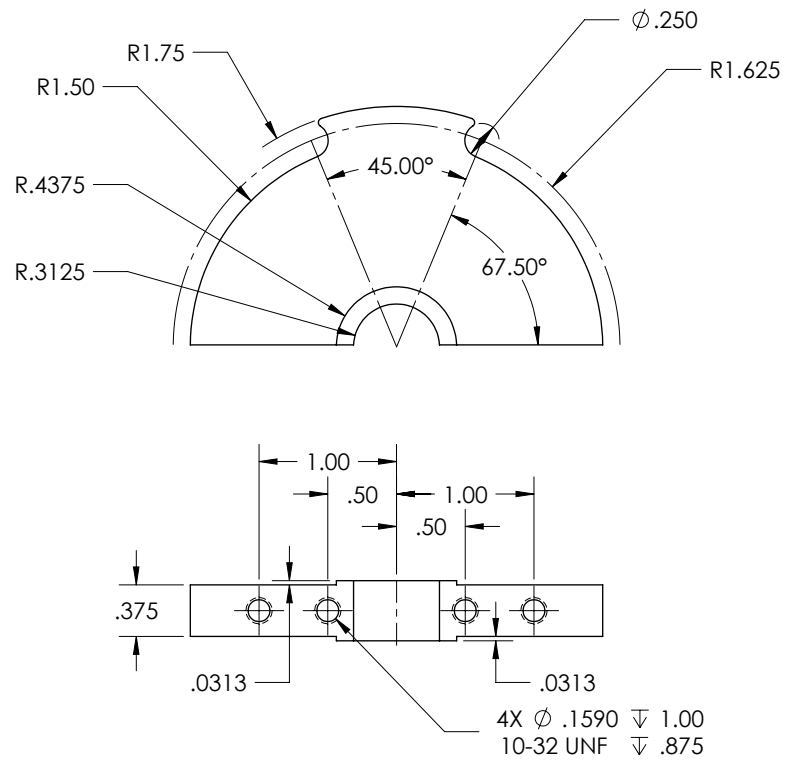


Figure D.11: End stop, top half – held between vertical blocks 1 and 2.

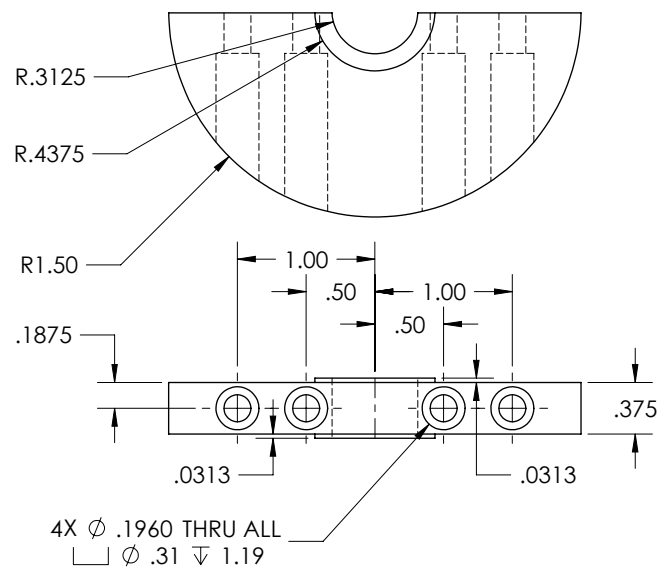


Figure D.12: End stop, bottom half – held between vertical blocks 1 and 2.

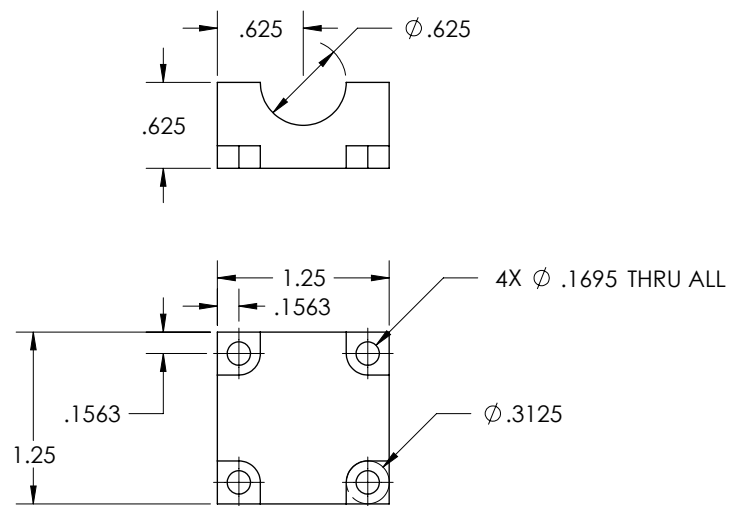


Figure D.13: Crank arm part 1 – mounts to output shaft.

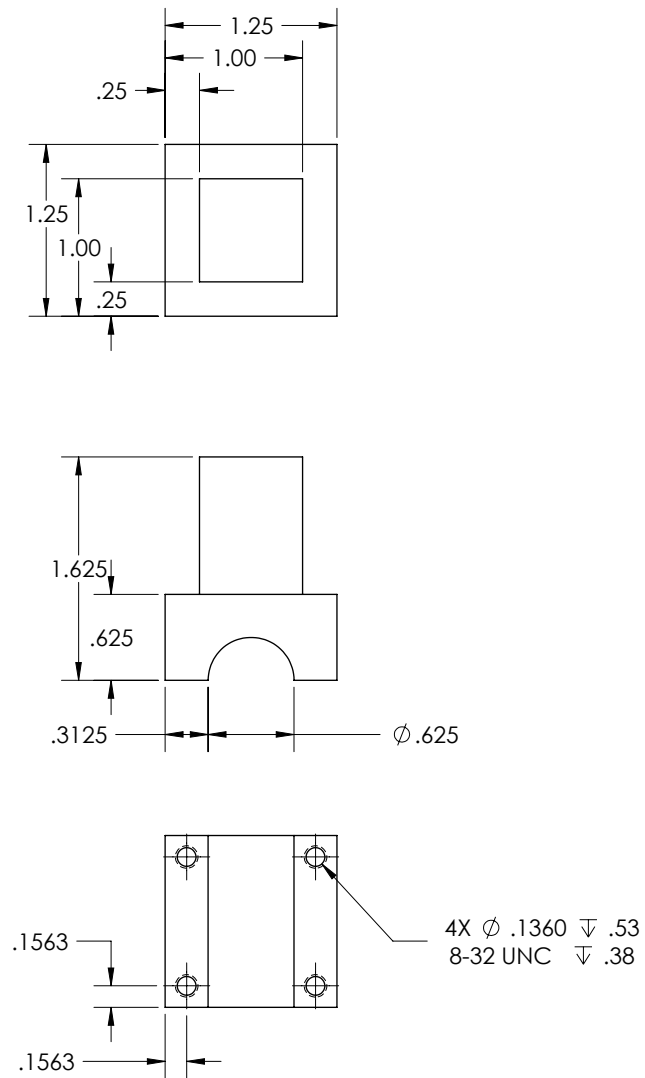


Figure D.14: Crank arm part 2 – mounts to output shaft.

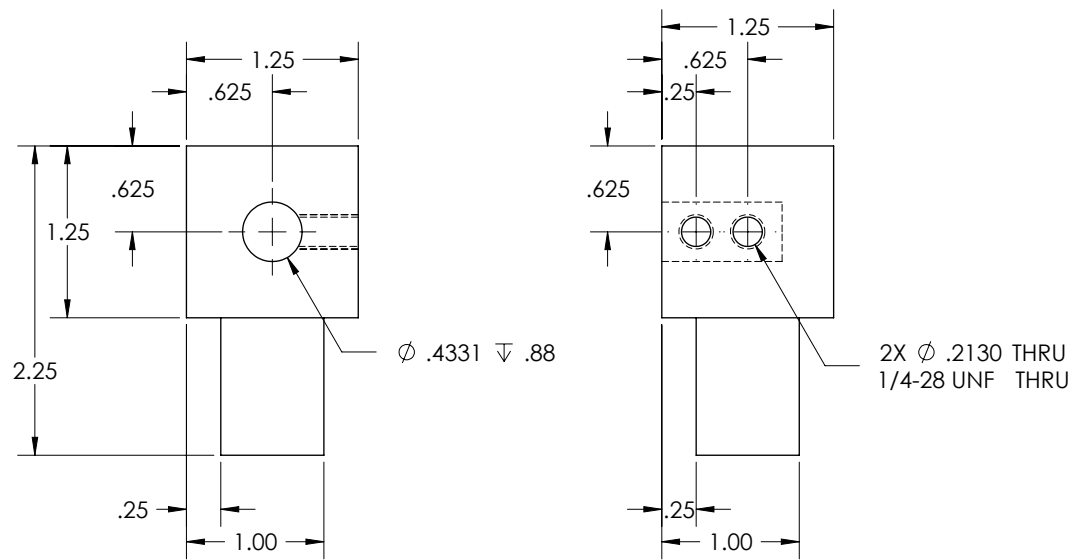


Figure D.15: Crank arm part 3 – holds load cell.

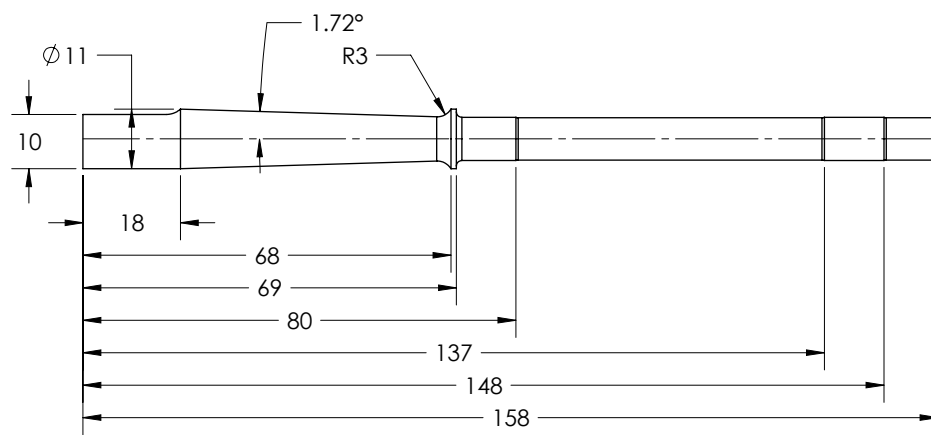


Figure D.16: Load cell beam – holds crank handle, dimensions in mm.

D.2 Assembly

The robot base plate is a 24" section of 4" by 5/8" aluminum bar. Block 1 mounts to the front end of the base, with the counterbore facing away from the front end. Block 2 mounts to the base with a 7/16" gap between it and Block 1 and with the counterbore facing towards Block 1. Output shaft contains, from front to back, Crank Arm 1 and 2, Encoder, Block 1 and bearing, End Stop, Block 2 and bearing, thrust bearing, and Follower. Block 3 mounts to the base 14" from the front end, with the gearbox pilot facing the rear. Gearbox face mounts to rear of Block 3, and thrust bearing is seated in bore on front of Block 3. Spline shaft contains, from front to back, spline nut, spring, spring cap, washer, thrust bearing, telescoping collar, and shaft coupler. Shaft coupler connects to gearbox output, pressed against thrust bearing. Small 5/16" shaft connects half of spider coupler and gearbox input. Motor Spider Coupler is installed on motor, and motor is mounted on Motor Mounting Block. Spider coupler halves are mated with spider, and Motor Mounting Block is mounted to base plate. Two 24" L-beams are attached to the corners of blocks 1, 2, and 3. Crank Arm 2 and 3 are connected together with 6.75" of 1" square tube. Load Cell is mounted in Crank Arm 3, and handle grip is installed over 8mm ID bearings on Load Cell beam.

APPENDIX E SOURCE CODE

All code is written in Matlab/Simulink unless otherwise specified.

E.1 Simulation Model

Figure E.2 shows the top-level Simulink block diagram of the simulation model. The top row of the diagram contains all the bodies and joints in the model. Stacked below each joint are the sensors and/or actuators for that joint along with the blocks that calculate the actuation force. The adaptive controller is in the bottom left corner, and the MIT-Manus arm model, shown in Figure E.1 is on the right side.

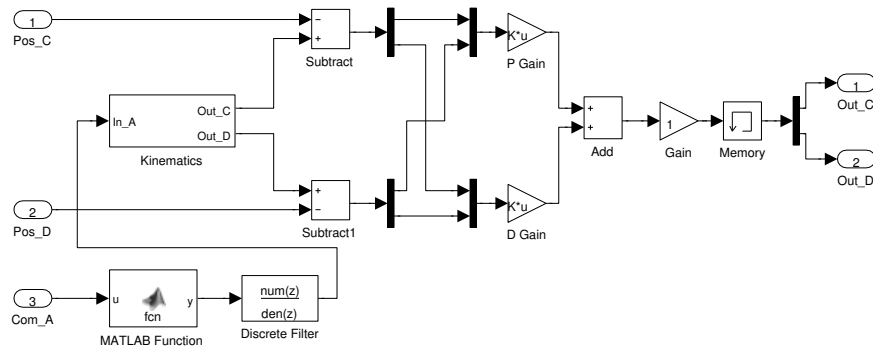


Figure E.1: Simulation model Simulink diagram, Human subsystem.

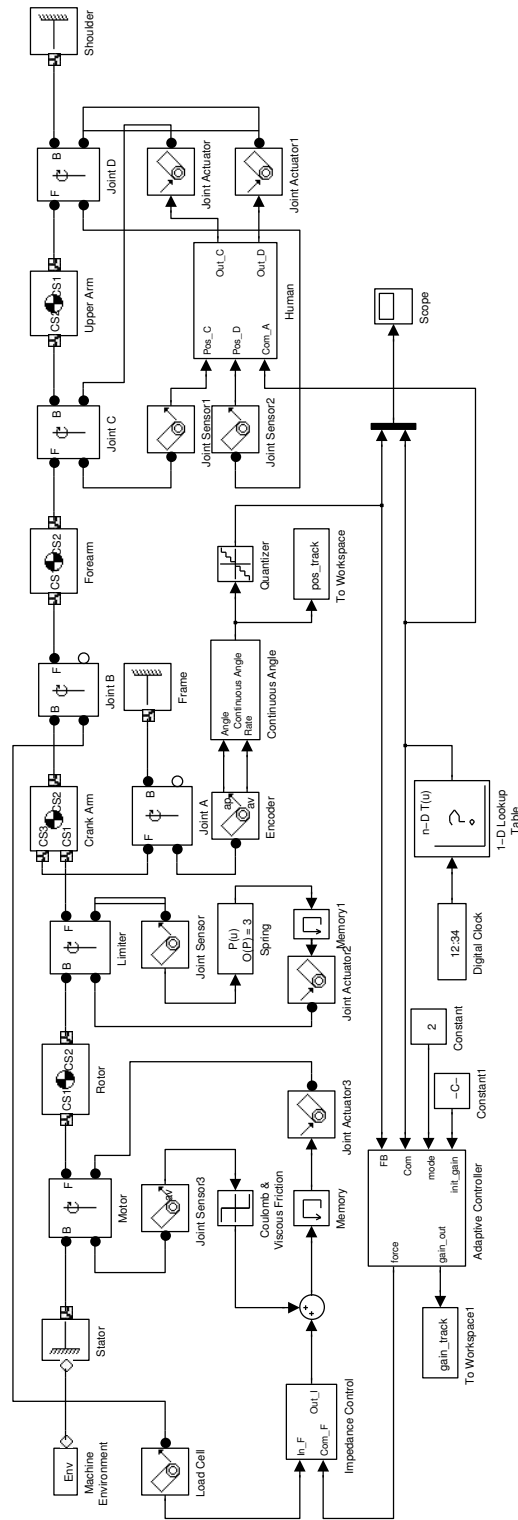


Figure E.2: Simulation model Simulink diagram, top level.

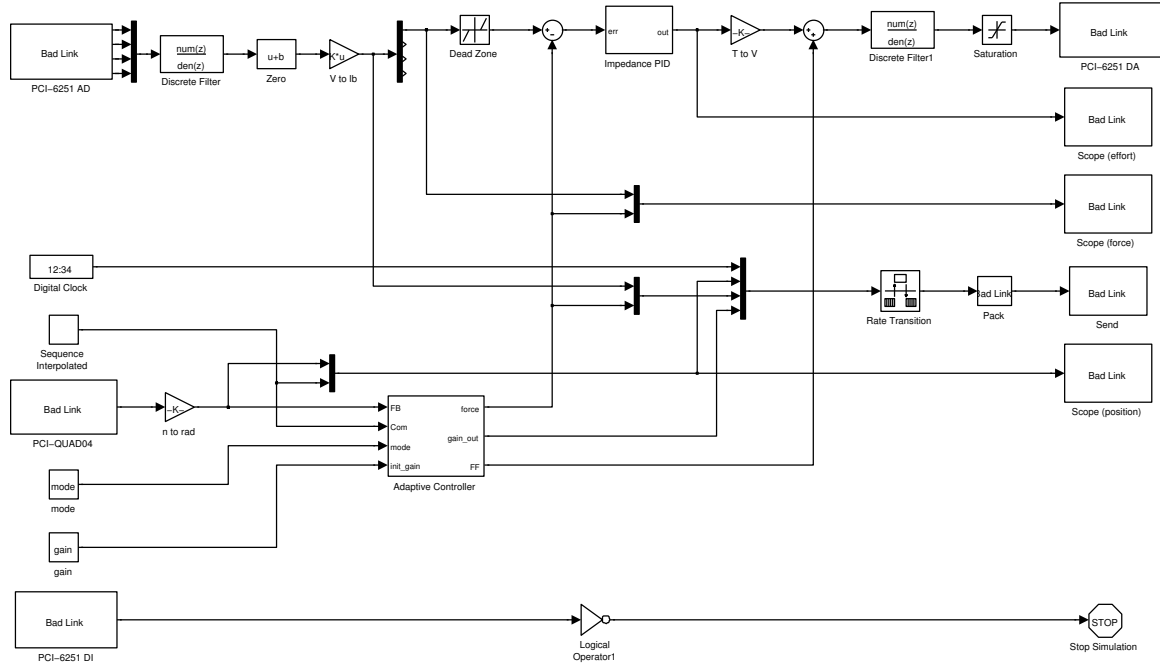


Figure E.3: Target PC Simulink diagram, top level.

E.2 Real-Time Robot Controller

The robot controller running on the xPC Target OS is shown in Figure E.3. The top left portion of the diagram is used to calculate the load cell reaction forces and to provide the feedback for the impedance controller in the top right. The adaptive controller is in the center left of the diagram, and the UDP transmission blocks are in the center right. Blocks in the bottom row abort the program if an emergency stop is pressed.

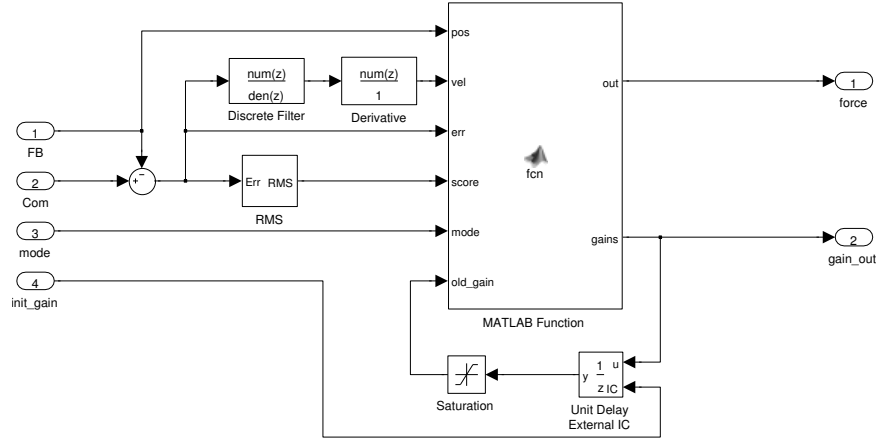


Figure E.4: Simulink diagram of adaptive controller subsystem.

E.3 Adaptive Controller

Figure E.4 shows the adaptive controller and its associated blocks, which calculate velocity, RMS error, and feedforward. This subsystem appears in both the real system and the simulation.

Adaptive controller Matlab function code is as follows:

```
function [out, gains] = fcn(pos, vel, err, score, mode, old_gain)
%#codegen

th=pos+(-2*pi:pi/8:2*pi);
win=[zeros(1,14) 0.13 0.63 1.00 0.63 0.13 zeros(1,14)]/2.52;

ctrlpts=-pi:pi/8:pi;
coeff=interp1(th,win,ctrlpts,'linear');

ov=zeros(1,17);
for i=1:17
    if old_gain(i)<=0
        ov(i) = old_gain(i)*(err-vel*0.18);
    else
        ov(i) = old_gain(i)*(pi-abs(err))*sign(err);
    end
end

out = dot(ov,coeff);

if mode==0
    gain_out=zeros(1,17);
    out=0;
elseif mode==1
    gain_out=old_gain;
elseif mode==2
    gain_out=old_gain+0.4*(0.25-score)*0.001;
```

```
elseif mode==3
    gain_out=min(old_gain+coeff*(0.25-score)*0.001, 0);
elseif mode==4
    gain_out=max(old_gain+coeff*(0.25-score)*0.001, 0);
elseif mode==5
    gain_out=old_gain+coeff*(0.25-score)*0.001;
else
    gain_out=old_gain;
end

gains=gain_out(1:17);
```

E.4 Subject GUI

Subject GUI code produces the interface seen in Figure 2.9. The source is as follows:

```
function flatdisp(block)
%MSFUNTMPL_BASIC A Template for a Level-2 MATLAB S-Function

setup(block);

%endfunction

function setup(block)

% Register number of ports
block.NumInputPorts = 2;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
block.SetPreCompInPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions = 1;
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'Real';
block.InputPort(1).DirectFeedthrough = false;

block.InputPort(2).Dimensions = -1;
block.InputPort(2).DatatypeID = 0; % double
block.InputPort(2).Complexity = 'Real';
block.InputPort(2).DirectFeedthrough = false;

% Override output port properties
block.OutputPort(1).Dimensions = 1;
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'Real';

% Register parameters
block.NumDialogPrms = 0;

% Register sample times
block.SampleTimes = [-1 0];

% Specify the block simStateCompliance.
block.SimStateCompliance = 'HasNoSimState';

block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
%block.RegBlockMethod('InitializeConditions', @InitializeConditions);
block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs); % Required
block.RegBlockMethod('Update', @Update);
%block.RegBlockMethod('Derivatives', @Derivatives);
block.RegBlockMethod('Terminate', @Terminate); % Required
```

```

%end setup

function DoPostPropSetup(block)
block.NumDworks = 4;

    block.Dwork(1).Name          = 'fighandle';
    block.Dwork(1).Dimensions    = 1;
    block.Dwork(1).DatatypeID    = 0;      % double
    block.Dwork(1).Complexity    = 'Real'; % real
    block.Dwork(1).UsedAsDiscState = true;

    block.Dwork(2).Name          = 'cursorhandle';
    block.Dwork(2).Dimensions    = 1;
    block.Dwork(2).DatatypeID    = 0;      % double
    block.Dwork(2).Complexity    = 'Real'; % real
    block.Dwork(2).UsedAsDiscState = true;

    block.Dwork(3).Name          = 'targethandle';
    block.Dwork(3).Dimensions    = 1;
    block.Dwork(3).DatatypeID    = 0;      % double
    block.Dwork(3).Complexity    = 'Real'; % real
    block.Dwork(3).UsedAsDiscState = true;

    block.Dwork(4).Name          = 'boxhandle';
    block.Dwork(4).Dimensions    = 1;
    block.Dwork(4).DatatypeID    = 0;      % double
    block.Dwork(4).Complexity    = 'Real'; % real
    block.Dwork(4).UsedAsDiscState = true;

function Start(block)

fh = figure('Position',[320 320 640 512],'Color','w');
set(fh, 'RendererMode','manual', 'Renderer','zbuffer', 'Units','normalized',...
    'ToolBar','none', 'MenuBar','none');
block.Dwork(1).Data = fh;

ah = axes('Position',[0 0 1 1], 'Visible','off',...
    'XLim',[-1,1], 'YLim',[-1,1]);

trj=-block.InputPort(2).Data';
y=1.75*(0:(length(trj)-1))/length(trj) -0.75;
th = line(trj/pi, y,'Color','k','LineWidth',2);
block.Dwork(3).Data = th;

box = line([-0.25 0.25 0.25 -0.25 -0.25]/pi,...
    [-0.25 -0.25 0.25 0.25 -0.25]/pi-0.75,'Color','b','LineWidth',2);
block.Dwork(4).Data = box;

ch = patch([-0.05 0.05 0],[-0.85 -0.85 -0.75],'y','LineWidth',2);
block.Dwork(2).Data = ch;

lh = line([0.75 -0.75],[-0.75 -0.75],'Color','k','LineWidth',2);

```



```

%endfunction

function Outputs(block)

fh=block.Dwork(1).Data;
set(0,'CurrentFigure',fh)
ch=block.Dwork(2).Data;
th=block.Dwork(3).Data;
box=block.Dwork(4).Data;

set(fh,'CurrentObject',th)
trj=-block.InputPort(2).Data';
tg=-block.InputPort(2).Data(1);
set(th, 'XData',trj/pi)
set(box, 'XData',tg/pi+[-0.25 0.25 0.25 -0.25 -0.25]/pi)

set(fh,'CurrentObject',ch)
pos=-block.InputPort(1).Data;
set(ch, 'XData',[-0.05 0.05 0]+pos/pi)

if abs(tg-pos) < 0.25
    set(ch, 'FaceColor','g')
else
    set(ch, 'FaceColor','r')
end

block.OutputPort(1).Data = fh;

%end Outputs

function Terminate(block)

fh=block.Dwork(1).Data;
delete(fh);
clear fh

%end Terminate

```

E.5 Data Analysis

This script performs the analysis of raw data from the haptic robot. Source is as follows:

```
% Haptic TheraDrive data processing script
% requires curve fitting toolbox

% Fill in missing data points
processed.time=0.01:0.01:90;
processed.signals.values=interp1(simout.signals.values(:,1),...
    simout.signals.values(:,2:25),processed.time,'pchip');

% Calculate overall RMS error
processed.rms_err_all=trapz(processed.time./90,...
    (processed.signals.values(:,1)-processed.signals.values(:,2)).^2)^0.5;

% Calculate fraction of time on target
processed.time_on_tg=trapz(processed.time./90,...
    abs(processed.signals.values(:,1)-processed.signals.values(:,2))<0.125);

% Calculate filtered position and velocity
b=1e-5*[0.089848614637233    0.359394458548934    0.539091687823401...
    0.359394458548934    0.089848614637233];
a=[1.000000000000000    -3.835825540647347    5.520819136622225...
    -3.533535219463012    0.848555999266476];
processed.position=filter(b,a,processed.signals.values(:,[1,2]),[],1);
processed.velocity=filter([-1.5 2 -0.5],0.01,processed.position,[],1);
processed.velocity(1:6,2)=zeros(6,1);
clear b a

% Calculate RMS position error as function of position
pts=-pi:pi/8:pi;
wins=eye(17);
weights=zeros(9000,17);
time=zeros(1,17);
processed.rms_err_pos=zeros(1,17);
for i=1:17
    weights(:,i)=interp1(pts,wins(:,i),processed.signals.values(:,1),'linear');
    time(i)=0.01*sum(weights(:,i));
    processed.rms_err_pos(i)=trapz(processed.time,...
        (1/time(i)).*(weights(:,i).*(processed.signals.values(:,1)-...
            processed.signals.values(:,2))).^2)^0.5;
end
clear pts wins weights time i

% Perform FFT
processed.frequency=0:100/8192:100-100/8192;
posfft=fft(processed.signals.values(:,[1,2]),8192,1);
processed.pos_fft_mag=abs(posfft);
processed.pos_fft_phase=angle(posfft);
clear posfft

% Calculate ratio of off-axis force to total force
```

```

fittedmodel=fit(processed.signals.values(:,1),processed.signals.values(:,5),...
    'a0+a1*cos(x)+b1*sin(x)+a2*cos(x*2)+b2*sin(x*2)', 'StartPoint',[0 0 0 0 0]);
grav=fittedmodel.a0+fittedmodel.a1*cos(processed.signals.values(:,1))+...
    fittedmodel.b1*sin(processed.signals.values(:,1))+...
    fittedmodel.a2*cos(processed.signals.values(:,1)*2)+...
    fittedmodel.b2*sin(processed.signals.values(:,1)*2);
f_on=processed.signals.values(:,7);
f_off=processed.signals.values(:,5)-grav;
processed.f_off_axis=mean((f_off.^2./(f_on.^2+f_off.^2)).^0.5);
processed.f_on_axis=mean((f_on.^2./(f_on.^2+f_off.^2)).^0.5);
clear fittedmodel goodness output grav f_on f_off

```

E.6 Miscellaneous Scripts

Trajectory generation

```
traj_t=0:0.01:90;
traj_x=(0.5*sin(0.5*traj_t+2*pi*rand(1))+...
        0.5*sin(1.0*traj_t+2*pi*rand(1))+...
        0.5*sin(1.5*traj_t+2*pi*rand(1))+...
        0.5*sin(2.0*traj_t+2*pi*rand(1))).*[0:0.004:1 ones(1,8750)];
```

Update adaptive controller gains for robot

```
gain=simout.signals.values(length(simout.signals.values),[9:25]);
```

Upload trajectory and operating parameters to robot

```
tg=xpc;
pid=tg.getparamid('', 'gain');
nul=tg.setparam(pid, gain);
pid=tg.getparamid('', 'mode');
nul=tg.setparam(pid, mode);
pid=tg.getparamid('', 'traj_t');
nul=tg.setparam(pid, traj_t);
pid=tg.getparamid('', 'traj_x');
nul=tg.setparam(pid, traj_x);
clear pid nul
```

Figure F.1: Schematic of strain gauge bridge amplifier.

F.2 Calibration Data

Table F.1: Raw calibration data from load cell

$P(\text{lb})$	$M(\text{in-lb})$	V_p	V_d	$P(\text{lb})$	$M(\text{in-lb})$	V_p	V_d
12.2230	59.5871	-0.7297	-1.7476	0.0000	0.0000	0.0284	0.0101
12.2230	35.1411	-0.3994	-0.7260	-0.2230	-0.1951	0.0286	0.0037
12.2230	10.6951	-0.0745	0.2891	-0.2230	-0.4181	0.0291	0.0125
8.2230	40.0871	-0.4617	-1.1601	-0.2230	-0.6411	0.0348	0.0219
8.2230	31.8641	-0.3622	-0.8319	-0.2230	-0.8641	0.0393	0.0317
8.2230	23.6411	-0.2236	-0.4640	-0.2230	-1.0871	0.0422	0.0405
8.2230	15.4181	-0.1199	-0.1296	-2.2230	-1.9451	0.0515	-0.0400
8.2230	7.1951	-0.0145	0.2035	-2.2230	-4.1681	0.0838	0.0511
6.2230	30.3371	-0.3458	-0.8870	-2.2230	-6.3911	0.1121	0.1414
6.2230	24.1141	-0.2600	-0.6396	-2.2230	-8.6141	0.1366	0.2278
6.2230	17.8911	-0.1602	-0.3566	-2.2230	-10.8371	0.1644	0.3275
6.2230	11.6681	-0.0845	-0.1073	-4.2230	-3.6951	0.0698	-0.0905
6.2230	5.4451	-0.0067	0.1451	-4.2230	-7.9181	0.1280	0.0845
4.2230	20.5871	-0.2296	-0.6102	-4.2230	-12.1411	0.1785	0.2553
4.2230	16.3641	-0.1717	-0.4409	-4.2230	-16.3641	0.2259	0.4231
4.2230	12.1411	-0.1004	-0.2507	-4.2230	-20.5871	0.2807	0.6038
4.2230	7.9181	-0.0377	-0.0798	-6.2230	-5.4451	0.0813	-0.1488
4.2230	3.6951	0.0012	0.0833	-6.2230	-11.6681	0.1674	0.1114
2.2230	10.8371	-0.1046	-0.3356	-6.2230	-17.8911	0.2473	0.3678
2.2230	8.6141	-0.0763	-0.2477	-6.2230	-24.1141	0.3263	0.6195
2.2230	6.3911	-0.0392	-0.1439	-6.2230	-30.3371	0.4181	0.8912
2.2230	4.1681	-0.0063	-0.0555	-8.2230	-7.1951	0.1009	-0.2055
2.2230	1.9451	0.0087	0.0286	-8.2230	-15.4181	0.2126	0.1309
0.2230	1.0871	0.0128	-0.0643	-8.2230	-23.6411	0.3225	0.4794
0.2230	0.8641	0.0108	-0.0578	-8.2230	-31.8641	0.4302	0.8221
0.2230	0.6411	0.0296	-0.0403	-8.2230	-40.0871	0.5438	1.1617
0.2230	0.4181	0.0314	-0.0303	-12.2230	-10.6951	0.1044	-0.3231
0.2230	0.1951	0.0203	-0.0288	-12.2230	-35.1411	0.4245	0.6808
0.0000	0.0000	-0.0051	-0.0467	-12.2230	-59.5871	0.7675	1.7211