QUANTITATIVE ANALYSIS OF TRAJECTORY TRACKING CHARACTERISTICS FOR A FUNCTIONAL HUMAN WRIST UNDER RESISTIVE LOADS

by

Jeffrey Tylyn Naylor

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

The University of Utah

May 2015

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Jeffrey Tylyn Naylor**.
Has been approved by the following supervisory committee members:

Sanford G. Meek	, Chair	**1/7/2015**. Date Approved
Andrew S. Merryweather	, Member	**1/7/2015**. Date Approved
V. John Mathews	, Member	**1/7/2015**. Date Approved

and by **Tim A. Ameel**, Chair/ Dean of

the Department/College/School of **Mechanical Engineering**.

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Functional neuromuscular stimulation (FNS) is electrical stimulation for muscle control. This ability has brought about a new advent in the field of prosthetics called neuroprosthetics. Neuroprosthetics consists of a wide field of devices that stimulate muscles or nerve tissue to either control part of the human body or to give it feedback. Strokes and spinal cord injuries cause a neural disconnect between the brain and the body. Recent research with FNS is exploring methods of bypassing this disconnect and allowing the affected person to control their body with just a thought. This same technology is also being used in robotic limbs that are controlled by thought and are capable of giving the wearer feedback about their environment.

Researchers use control algorithms to convert brain signals into motion. With the development and testing of these control algorithms the question has arisen of how to determine when a controller is good enough. How should the neuroprosthetic perform? A standard is needed with which neuroprosthetic control can be measured.

The purpose of this research is to measure standard engineering control metrics from functional human wrists in order to develop a standard for future neuroprosthetic design. Three different types of functions were presented to healthy human subjects for trajectory tracking exercises. These functions included step functions to measure transient responses, ramp functions to measure steady state responses, and periodic functions, which are most typical of normal activities of daily living. Varying loads were applied to the participants' wrists, and wrist position was measured. The data from these three experiments were used to measure standard engineering control metrics. From these control metrics statistical regression models were developed to provide a quantitative view of healthy human wrist control with a load applied to it.

This work is dedicated to my family.

My wife, Rachel, who has loved and supported me through many years of school even when the end seemed out of sight. She has truly been my inspiration academically, and I have achieved far more with her by my side than I ever would have alone.

My parents, Gary and Lyn, for teaching me that by working hard anything is possible. They gave me a love of learning and the power to plow through when things are hard.

I love you all.

TABLE OF CONTENTS

Chapters

Appendices

LIST OF FIGURES

Figure

ABBREVIATIONS

Spinal Cord Injury (SCI)
Functional Neuromuscular Stimulation (FNS)
Longitudinal Intrafascicular Electrode (LIFE)
Transverse Intrafascicular Multielectrode (TIME)
Utah Slanted Electrode Array (USEA)
Flat Interface Nerve Electrode (FINE)
Activities of Daily Living (ADLs)
Degree of Freedom (DOF)
Electromyogram (EMG)
Maximal Volitional Force (MVF)
Range of Motion (ROM)
Graphic User Interface (GUI)
Part Number (PN)
Data Acquisition (DAQ)
Electromotive Force (EMF)
Pulse Width Modulated (PWM)
National Instruments (NI)
Proportional Integral Derivative (PID)
Statistical Package for the Social Sciences (SPSS)
Analysis of Variance (ANOVA)
Multivariate Analysis of Variance (MANOVA)
Quantile-Quantile (QQ)

ACKNOWLEDGEMENTS

I would like to thank Seth Paul who worked with me to develop the system used in these experiments. We spent many late nights and long weeks learning new things and fixing many problems. I would also like to thank Nate Rohbock, the statistics wizard, who helped me understand and develop the statistical models developed in this research. Finally I would like to thank my committee, thank you all for your knowledge and time.

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Motivation

Spinal cord injuries and strokes are both neuromuscular injuries that leave the affected body parts intact but disconnected from motor control. Each year in the United States approximately 800,000 people suffer from strokes [1], and 12,000 people suffer spinal cord injuries [2]. Both of these injuries can cause moderate to severe paralysis and even death. For survivors, rehabilitation focuses on gaining back as much motor function as is possible and coping with the remaining deficiencies. With the advances being made in the fields of robotic and neuroprosthetics, scientists are working toward a future where no one will have to live with a loss of function because of these injuries.

A stroke is caused by the loss of blood flow to a portion of the brain [3]. Eighty-seven percent of strokes are ischemic [1]: caused by the blockage of blood vessels in the brain. The remainders are hemorrhagic: caused by the rupturing of blood vessels in the brain. In either case the lack of oxygen causes the brain cells to die, disconnecting the neuromuscular pathways. Since strokes are “the leading cause of serious long-term disability in the United States” [1, p. e137] it is clear that many people who have had a stroke live the rest of their lives suffering with loss of normal function. For these people, rehabilitation methods rely on neuroplasticity or the ability of the brain to redirect neural pathways. Thus "the most important element in any rehabilitation program is carefully directed, well-focused, repetitive practice…" [4].

A spinal cord injury (SCI) is defined as "damage to any part of the spinal cord or nerves at the end of the spinal canal…" [5, p. 1]. Similar to a stroke, a spinal cord injury disconnects the neural pathways that are used to control the body. Depending on the severity of the injury some or all of the person's function and feeling can be lost below the injury site. Treatment is primarily geared toward coping with loss of function [6].

Since after both a stroke and SCI the physiology of the extremities are intact, scientists are working on stimulating the muscles of the affected limb in order to increase muscle strength, improve passive range of motion, and aid in voluntary motor control [7]. These techniques are currently being used around the world as an integrated part of poststroke and post-spinal-cord injury rehabilitation. Studies show that muscle strength and range of motion improves during the use of electrical stimulation, but that these benefits are lost within a few weeks of terminating the exercise regimen [8]. For patients with a severe loss of functionality this may not be of any benefit. For this reason scientists have been working on using electrical stimulation to bypass the damaged neuromuscular pathway with the idea of controlling the patient's muscles indefinitely.

1.1.2 Functional Neuromuscular Stimulation

Functional neuromuscular stimulation (FNS) is electrical stimulation to control muscles. FNS is fundamental to the field of neuroprosthetics, which consists of devices that "restore or support parts of the neuromuscular or neurosensory systems by stimulating muscle or neural tissue electrically" [9, p. 3]. This field includes devices such as cochlear implants, deep brain stimulators, and devices used to control a person's existing limb or a robotic limb used for amputees. Although there are many challenges still to overcome [10], exciting advances are being accomplished in FNS, which expands the potential of neuroprosthetics.

Development of FNS is most apparent in the electrodes used to deliver current to the muscle fibers. Surface electrodes are the least invasive and most clinically proven means of stimulation. However they activate muscles indiscriminately and require larger amounts of current. This prevents both selective muscle control and control for long durations due to muscle fatigue [11]. Invasive electrodes include those implanted into the muscle tissue and those implanted into the peripheral neural tissue. These implanted electrodes are more selective in the muscles they stimulate and require less current, which causes less muscle fatigue [12].

Peripheral nerve stimulators require the least amount of current to enervate the muscle and are the most selective as they are nearest the individual motor neurons. Electrodes connected to the exterior of the nerve are often referred to as cuff electrodes and include many different designs. Further reduction of necessary stimulation current is accomplished by piercing the nerve fascicles. Examples of electrodes implanted interfascicularly include the longitudinal intrafascicular electrode (LIFE) [15], the transverse intrafascicular multielectrode (TIME) [16], and the Utah Slanted Electrode Array (USEA) [17].

1.2 Literature Review

1.2.1 Feline Joint Control

Recent work at the University of Utah by Frankel et al. experimented with new applications of FNS to control both the force and joint motion of an anesthetized feline. In this study [18], the USEA was used to achieve fatigue resistant muscular stimulation of the feline's sciatic nerve. Three studies were performed with FNS controllers, and in them control of both the force and joint motion of the feline's hind ankle was achieved. At the end of these studies a preliminary effort was made to compare the results of the controllers with those of a functional human's abilities, setting the ground work for this study.

Potential real world applications of FNS such as gait, stance, and even holding a glass of water are examples of resistive loads that require maintaining force control for relatively long durations of time. Fatigue resistant control was achieved by implementing both intrafascicular and asynchronous stimulation. It has been shown that implanting the electrode intrafascicularly reduces muscle fatigue [12] by reducing the current required to stimulate the muscle to 10–100 μA [12] from the approximately 100 mA [19] used in surface stimulation. It has also recently been shown that asynchronous multielectrode stimulation in this area further reduces fatigue [20] while still achieving desired muscular stimulation.

In the first two of the studies a means of controlling force trajectories of a feline's ankle in real-time was developed. This constituted "the first successful use of a multiple input single output closed-loop control strategy for asynchronous intrafascicular muscle stimulation to produce precise, time-varying, fatigue-resistant isometric muscle force" [18, p. 53]. In the third study, the controller developed in the second study was expanded to control the feline's joint position against a resistive load [18]. The ability to track desired time-varying signals that this study shows demonstrates potential for controlling human joints in activities such as walking, standing, holding objects, and more complex tasks.

At the end of these studies the need to evaluate the quality of the results obtained from FNS control was recognized. A comparison was proposed between the obtained results and those of functional humans performing similar tasks. A preliminary study was performed in which four human subjects pushed on the same cat foot plate used for the feline study. A force was applied to the foot plate, and the subjects tracked a moving signal on a visible oscilloscope display. The results of the previously mentioned controllers were then compared to these human subjects, showing potential for this as a method of comparison.

Although this preliminary comparison of the controller's results to functional humans proved to be a good method of measuring the quality of the controller, there are a number of improvements that should be done to provide concrete metrics for future controller design. First, the human comparison was performed on the same equipment as the feline study. The equipment was not designed for comparable human movements or loads. Also this comparison never fully provides metrics that can be used for future design; it simply lays the ground work for the method to compare results. The question remains, what are the natural control characteristics of a functional human wrist?

1.2.2 Functional Wrist Characterization

In the medical field, functionality of an upper extremity is measured by scoring its ability to perform activities of daily living (ADLs) [21], [22]. While this is effectively used to evaluate the effect of rehabilitation methods [23], [24] and patient progress [25], it is far from a comprehensive analysis of the human arm. The wrist is a complex joint that involves the coordination of many bones and muscles. However, despite the wrist's complexity it has been shown that it can be approximated by a two degree of freedom (DOF) joint [26], typically with the radial-ulnar deviation axis slightly distal from the flexion-extension axis

[27].

Passive and active range of motion have been studied during ADLs [28], [29] to evaluate disabilities [30], validate joint measurement techniques [31], and to aid in ergonomics design [32]. Other characteristics of the human wrist such as joint stiffness [33], [34], damping [35], and inertia [36] have been measured to understand how the human wrist performs with disturbances as well as to aid in the design of rehab techniques and devices [37]. Many of the studies of wrist characteristics fall into the category of either studies of wrist kinematics [38]–[42] or of wrist kinetics [42], [43].

The study of wrists that have been affected by injury and disease is also a large field of research. Studies have been performed to see how well the wrist performs after neuromuscular injuries [44] and under different normal conditions [45], [46]. Research has also studied how well ADLs can be performed with wrist impairments [47] and what compensation is needed from other joints [48] during these ADLs.

A deeper understanding of the joint is obtained by measuring muscle activation during wrist movement [49], [50] and its response to loads [51]. This is done using Electromyogram (EMG) [52] to look at the electrical signals being sent to stimulate the muscle. As understanding of the electrical control of wrist motion has increased, electrical stimulation has been used to reproduce these control signals to the wrist [53]–[57], thus moving the medical field from being simply observers of these characteristics to being controllers of them.

1.2.3 Research Outline

Extensive studies of wrist characteristics have been performed including those of Strick and Hoffman [41], [50], [58]–[64] and more recently Charles [42]. While these studies have looked at a large variety of the joint's characteristics they do not look at the wrist from an engineering control aspect. If one's arm function is lost either by amputation or neuromuscular injury and that function can be replaced by robotics or by stimulation of one's existing arm, how should that arm perform? What rise time, peak time, overshoot, settling time, and steady state error are good for a normal functioning wrist?

The purpose of this study was to obtain these engineering control metrics from functional human subjects. This was done by designing a test mechanism similar in concept to that used in the feline study but designed ergonomically for human movements and comparable human loads. The metrics can then be used to determine the quality of controllers being designed for future neuroprothetics.

1.3 References

[1] S. Go et al., "Heart disease and stroke statistics—2013 update: a report from the American Heart Association," *Circulation*, vol. 127, no. 1, 2013.

[2] The National Spinal Cord Injury Statistical Center. (2013, February) *Spinal cord injury facts and figures at a glance* [Online] Available: https://www.nscisc.uab.edu/PublicDocuments/fact_figures_docs/Facts%202013.pdf

[3] A Mayo Foundation for Medical Education and Research. (2014, February 14) *Stroke* [Online] Available: http://www.mayoclinic.org/diseases-conditions/stroke/basics/definition/con-20042884

[4] National Institute of Neurological Disorders and Stroke. (2013, December 16) *Post-Stroke Rehabilitation Fact Sheet* [Online] Available: http://www.ninds.nih.gov/disorders/stroke/poststrokerehab.htm

[5] A Mayo Foundation for Medical Education and Research (2011, October 22) *Spinal Cord Injury* [Online] Available: http://www.mayoclinic.org/diseases-conditions/spinal-cord-injury/basics/definition/CON-20023837

[6] C. Donnelly et al., "Client-centred assessment and the identification of meaningful treatment goals for individuals with a spinal cord injury," *Spinal*

Cord, vol. 42, no. 5, pp. 302–307, 2004.

[7] M. Glanz et al., "Functional electrostimulation in poststroke rehabilitation: a meta-analysis of the randomized controlled trials," *Arch. Phys. Med. Rehabil.,* vol. 77, no. 6, pp. 549–553, 1996.

[8] A. D. Pandyan et al., "Effects of electrical stimulation on flexion contractures in the hemiplegic wrist," *Clin. Rehabil.,* vol. 11, no. 2, pp. 123–130, 1997.

[9] K. W. Horch et al., *Neuroprosthetics: theory and practice*. Toh Tuck Link, Singapore: World Scientific, 2004, ch. 1, p. 3.

[10] M. A. Lebedev and M. A. Nicolelis, "Brain–machine interfaces: past, present and future." *TRENDS Neurosci.,* vol. 29, no. 9, pp. 536–546, 2006.

[11] D. J. Tyler and D. M. Durand, "Functionally selective peripheral nerve stimulation with a flat interface nerve electrode," *Neural Syst. IEEE Trans. Biomed. Eng.,* vol. 10, no. 4, pp. 294–303, 2002.

[12] G. G. Naples et al., "A spiral nerve cuff electrode for peripheral nerve stimulation," *IEEE Trans. Biomed. Eng.,* vol. 35, no. 11, pp. 905–916, 1988.

[13] S. Mraz. (2014, July 25) *Technology Adds the Sense of Touch to Prosthetic Hands* [Online]. Available: http://machinedesign.com/medical/technology-adds-sense-touch-prosthetic-hands

[14] C. Hassler et al., "Polymers for neural implants," *J. Polym. Sci., Part B: Polym. Phys.,* vol. 49, no. 1, pp. 18–33, 2011.

[15] K Yoshida et al., "Development of a thin-film longitudinal intrafascicular electrode," *Proc. Fifth Ann. Conf. IFESS*, pp. 279–284, 2000.

[16] W. Jensen, "Transverse, Intrafascicular Multichannel Electrode system for induction of sensation and treatment of phantom limb pain in amputees (TIME)," Aalborg Univ., Denmark, Tech. Rep. FP7 - ICT-2007.3.6 Micro/Nanosystems priority, 2008.

[17] E. Jarvik, "The Bionics Man," *Continuum,* Summer 2014.

[18] M. A. Frankel, "Control methods of asynchronous intrafascicular multi-electrode stimulation for neuromuscular prostheses," Ph.D dissertation, Univ. Utah, Salt Lake City, UT, 2013.

[19] M. R. Popovic et al., "Surface-stimulation technology for grasping and walking neuroprostheses." *IEEE Engin. Med. Bio. Mag.,* vol. 20, no. 1, pp. 82–93, 2001.

[20] D. McDonnall et al., "Interleaved, multisite electrical stimulation of cat sciatic nerve produces fatigue-resistant, ripple-free motor responses*," IEEE Trans. Neural Sys. Rehab. Eng.,* vol. 12, no. 2, pp. 208–215, Jun. 2004.

[21] R. C. Lyle, "A performance test for assessment of upper limb function in physical rehabilitation treatment and research," *Int. J.Rehabil. Res.,* vol. 4, no. 4, pp. 479–491, 1981.

[22] A. R. Fugl-Meyer et al., "The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance," *Scand. J. Rehabil. Med.,* vol. 7, no. 1, pp. 13–31, 1974.

[23] A. D. Pandyan, M. H. Granat, and D. J. Stott, "Effects of electrical stimulation on flexion contractures in hemiplegic wrist," *Clin. Rehabil.,* vol. 11, no. 2, pp. 123–130, 1997.

[24] J. Powell et al., "Electrical stimulation of wrist extensors in poststroke hemiplegia," *Stroke,* vol. 30, no. 7, pp. 1384–1389, 1999.

[25] G. Broeks et al., "The long-term outcome of arm function after stroke: results of a follow-up study," *Dis. Rehabil.,* vol. 21, no. 8, pp. 357–364, 1999.

[26] J. G. Andrews and Y. Youm, "A biomechanical investigation of wrist kinematics," *J. Biomech.,* vol. 12, no. 1, pp. 83–93, 1997.

[27] R. B. Brumbaugh et al., "An in-vivo study of normal wrist kinematics," *J. Biomech. Eng.,* vol. 104, no. 3, pp. 176–181, 1982

[28] J. Ryu et al., "Functional ranges of motion of the wrist joint," *J. Hand Surg.*, vol. 16, no. 3, pp. 409–419, 1991.

[29] A. K. Palmer et al. "Functional wrist motion: a biomechanical study," *J. Hand Surg.,* vol.10, no. 1, pp. 39–46, 1985.

[30] R. H. Brumfield and J. A. Champoux, "A biomechanical study of normal functional wrist motion," *Clin. Orthop. Relat. Res.*, vol. 187, pp. 23–25, 1984.

[31] M. M. Horger, "The reliability of goniometric measurements of active and passive wrist motions," *Am. J. Occup. Ther.,* vol. 44, no. 4, pp. 342–348, 1990.

[32] *Anthropometry Biomech.,* Man-System Integration Standards, NASA-STD-3000, 1995.

[33] D. Formica et al., “The passive stiffness of the wrist and forearm,” *J. Neurophysio.,* vol. 108, no. 4. pp. 1158, 2012.

[34] T. Sinkjaer and R. Hayashi, “Regulation of wrist stiffness by the stretch reflex,” *J. Biomech.,* vol. 22, no. 11, pp. 1133–1140, 1989.

[35] T. E. Milner and C. Cloutier, “Damping of the wrist joint during voluntary movement,” *Exper. Brain Res.,* vol. 122, no. 3, pp. 309–317, 1998.

[36] S. K. Charles and N Hogan, “Dynamics of wrist rotations,” *J. Biomech.,* vol. 44, no. 4, pp. 614–621, 2011.

[37] D. J. Williams et al., “A robot for wrist rehabilitation,” *Engineering in Medicine and Biology Society* in *23rd Ann. Int. Conf. IEEE*, vol. 2, pp. 1336–1339, 2001.

[38] Y. Youm and A. Flatt, “Kinematics of Human Motion,” *Clin. Orthop. Relat. Res.,* vol. 149, pp. 21–32, 1980.

[39] Y. Youm et al., “Kinematics of the wrist. 1. An experimental study of radial-ulnar deviation and flexion-extension,” *J. Bone Jt. Surg.,* vol. 60, no. 4, pp. 423–431, 1978.

[40] G. R. Sennwald et al., “Kinematics of the wrist and its ligaments,” *J. Hand Surg.,* vol. 18, no. 5, pp. 805–814, 1993.

[41] D. S. Hoffman and P. L. Strick, “Step-tracking movement of the wrist in humans. 1. Kinematic analysis,” *J. Neurosci.,* vol. 6, no. 11, pp. 3309–3318, 1986.

[42] S. K. Charles, “It’s all in the wrist: a quantitative characterization of human wrist control,” Ph.D. dissertation, Dept. Mech. Eng., MIT, Cambridge, MA, 2008.

[43] M. A. Lemay and P. E. Crago, “A dynamic model for simulating movements of the elbow, forearm, and wrist,” *J. Biomech.*, vol. 29, no. 10, pp. 1319–1330, 1996.

[44] F. Pisano et al., “Quantitative measures of spasticity in post-stroke patients,” *Clin. Neurophysiol.*, vol. 111, no. 6, pp. 1015–1022, 2000.

[45] F. J. Valero-Cuevas and C. F. Small, “Load dependence in carpal kinematics during wrist flexion in vivo,” *Clin. Biomech.*, vol. 12, no. 3, pp. 154–159, 1997.

[46] C. A. Lucidi and S. L. Lehman, “Adaptation to fatigue of long duration in human wrist movements,” *J. Appl. Physiol.,* vol. 73, no. 6, pp. 2596–2603, 1992.

[47] A. M. Jette, L. G. Branch, and J. Berlin, “Musculoskeletal impairments and physical disablement among the aged,” *J. Gerontol.,* vol. 45, no. 6, pp. M203–M208, 1990.

[48] B. D. Adams et al., “Impact of impaired motion on hand and upper-extremity performance,” *J. Hand Surg.*, vol. 28, no. 6, pp. 898–903, 2003.

[49] R. W. Angel, “Electromyography during voluntary movement: two-burst pattern,” *Electroencephalogr. Clin. Neurophysiol.,* vol. 36, pp. 493–498, 1974.

[50] D. S. Hoffman and P. L. Strick, “Step-tracking movements of the wrist in humans. II. EMG analysis,” *J. Neurosci.,* vol. 10, no. 1, pp. 142–152, 1990.

[51] S. J. De Serres and T. E. Milner, “Wrist muscle activation patterns and stiffness associated with stable and unstable mechanical loads,” *Exp. Brain Res.,* vol. 86, no. 2, pp. 451–458, 1991.

[52] B. E. Mustard and R. G. Lee, “Relationship between EMG patterns and kinematic properties for flexion movements at the human wrist,” *Exp. Brain Res.,* vol. 66, no. 2, pp. 247–256, 1987.

[53] N. Matsushita et al., “Electromyogram analysis and electrical stimulation control of paralysed wrist and hand,” *J. Electromyogr. Kinesiol.,* Vol. 5, no. 2, pp. 117–128, 1995.

[54] P. H. Peckham et al., “An advanced neuroprosthesis for restoration of hand and upper arm control using implantable controller,” *J. Hand Surg.,* vol. 27, no. 2, pp. 265–276, 2002.

[55] M. W. Keith et al., “Functional neuromuscular stimulation neuroprosthesis for the tetraplegic hand,” *Clin. Orthop. Relat. Res.,* vol. 233, pp. 25–33, 1988.

[56] H. Ring and N. Rosenthal, “Controlled study of neuroprosthetic functional electrical stimulation in sub-acute post-stroke rehabilitation,” *J. Rehab. Med.,* vol. 37, no. 1, pp. 32–36, 2005.

[57] J. Allen and G. F. Inbar, “FNS control schemes for the upper limb,” *IEEE Trans. Biomed. Eng.,* vol. BME-33, no. 9, 1986.

[58] D. S. Hoffman and P. L. Strick, “Step-tracking movements of the wrist. III.

Influence of changes in load on patterns of muscle-activity," *J. Neurosci.*, vol. 13, pp. 5212–5227, 1993.

[59] D. S. Hoffman and P. L. Strick, "Step-tracking movements of the wrist. IV. Muscle activity associated with movements in different directions," *J. Neurophysiol.*, vol. 81, pp. 319–333, 1999.

[60] K. D. Pfann et al., "Common principles underlying the control of rapid, single degree-of-freedom movements at different joints," *Exp. Brain Res.*, vol. 118, pp. 35–51, 1998.

[61] D. S. Hoffman and P. L. Strick, "Activity of wrist muscles during step-tracking movements in different directions," *Brain Res.*, vol. 367, 1986.

[62] S. Kakei et al., "Sensorimotor transformations in cortical motor areas," *Neurosci. Res.*, vol. 46, pp. 1–10, 2003.

[63] S. Kakei et al., "Direction of action is represented in the ventral premotor cortex," *Nat. Neurosci.*, vol. 4, pp. 1020–1025, 2001.

[64] S. Kakei et al., "Muscle and movement representations in the primary motor cortex," *Science*, vol. 285, pp. 2136–2139, 1999.

CHAPTER 2

EXPERIMENT DESIGN

2.1 Engineering Control Metrics

Engineering control metrics are standard measures of how well a system accomplishes a task for which it is being controlled. In this study these metrics were applied to the analysis of the human wrist. The metrics were obtained by measuring the angular position of the wrist while it tracked standard signals with a load applied to it. The three signals of interest were steps, ramps, and periodic motion.

Tracking of a step function is widely used in the field of robotics for system identification and for tuning a system's transient response [1]. A feedback system is given a desired instantaneous change in position or speed and its ability to track that desired signal is measured. Based on these measurements the system can be adjusted using software to have a more desirable tracking performance. This same task was applied to a human wrist and is analogous to a sudden change in wrist angle. Throwing an object, fine pointing, and fine lifting tasks are examples of this typical human motion. Step tracking motion of the human wrist has been thoroughly studied, and the wrist's performance characteristics for step tracking tasks are well known [2]–[5]. However, engineering control metrics have not been studied for these tasks. For a step tracking exercise, these metrics, as shown in Figure 2.1, include rise time (T_r), settling time (T_s), peak time (T_p), delay time (T_d), and percent overshoot (%OS). These metrics were analyzed for step functions of different amplitudes because muscle activation is different for small and large amplitude movement [2].

A ramp function is used in the field of robotics to characterize a system's steady state tracking response to a desired function. This provides new information about the systems that is not captured using a step response. Tracking of a ramp signal was performed by the human wrist and is analogous to lifting or lowering tasks where a controlled constant velocity is required. The metrics of interest in the ramp tracking exercise, as shown in Figure 2.2, are time shift (T_s), amplitude error (E_a), and variance of velocity (V_v). Amplitude error is used in lieu of steady state error because of the adaptive nature of the human control system. Variance of velocity is not a typical metric however this was determined to be a good way of quantifying the smoothness of the human's ramp tracking.

Periodic motion such as sinusoids and combinations of sinusoids are more complex functions that are time dependent. Tracking these signals is analogous to any repetitive motion task such as the motion used while brushing ones teeth. Quite a few activities of daily living show sinusoidal type motion [6]–[8]. It has been shown that humans are able to track predictable motion at frequencies near 3 Hz [9] and unpredictable motion at frequencies up to 0.5 Hz [10]. With these criteria in mind both predictable and unpredictable periodic motion were used in these experiments. The metrics of interest for predictable periodic motion, shown in Figure 2.3, were peak to peak response amplitude (A_{pp}), time shift (T_s), amplitude error (E_a), and percent variance in velocity ($\%V_V$). The metrics of interest for unpredictable periodic motion were the same as the predictable periodic motion, except the peak to peak response amplitude (A_{pp}) was not measured. Percent variance of velocity was used for the periodic responses because the desired signal has a quantifiable variance in velocity, which was not the case for the ramp signals.

2.2 Experiment Procedures

Study participants were given an explanation of the intent of the study and the requirements for participation. As a requirement for participation the participants were to have no impairment of the upper extremity or history of neuromuscular disorders. The study risks, safety features, and procedures were explained, and participants were given the opportunity to ask questions until they verbally acknowledged an understanding of the study. All participants read and signed a consent form in accordance with the University of Utah investigational review board.

Biometrics values of each participant were recorded, and a numeric value was assigned to each participant so that these values could be correlated with their experiment results. The biometrics values included sex, age, dominant hand, hand measurements, wrist range of motion and the maximal volitional force (MVF) of their wrist in flexion and extension. The equipment was then adjusted to allow the subject to sit up straight with their elbow resting on a support at a 90° angle and their forearm in the pronated position. The device handle was adjusted so that the axis of rotation of the wrist and the device were approximately the same.

Range of motion (ROM) was measured with the participant's arm strapped to the arm rest to minimize forearm rotation. Gravity compensation was applied to eliminate the effect of the handle's weight, and a feedback controller was applied to the system to compensate for the friction and inertia of the motor and gearbox. The participant was then instructed to grasp the handle and move their wrist in flexion and extension through their full ROM. The minimum and maximum angles were recorded with zero degrees being the starting horizontal position of the handle.

An isometric measurement of MVF was taken using position feedback control. The participant grasped the handle with their forearm strapped in place and was instructed to rotate the handle as forcefully as possible in both flexion and extension. Strapping down the participant's arm prevented them from using the shoulder muscles to push or pull on the handle. Control schemes for the ROM test and the MVF test a shown in Appendix D as ROM_model and Weight_model.

After measuring all of the biometric data, the participants were given time to practice on the equipment. The practice included a torque proportional to the MVF in flexion being applied to their wrist. The participants sat approximately $3/4$ of a meter (30 in.) away from a computer screen displaying a red dot 4.5 mm (0.177 in.) in diameter. The red dot was controlled by the participant and was moved in the vertical direction by rotating the wrist device handle in the vertical direction. Rotating the handle 45° corresponded to the red dot moving 10 cm on the screen. The amplitude of movement on the screen was scaled so that ±54° was the full range of the screen. A gray line was visible throughout the experiment and represented the desired signal to be tracked. The screen always displayed 5 seconds of time, 4.5 s in the future and 0.5 s in the past. This allowed the participants to anticipate the movements they would be making. As time progressed, the horizontal axis and the gray "desired signal" line shifted from right to left across the screen, while the red dot remained in the same horizontal location on the screen. The participant's screen can be seen in Figure 2.4.

Once the subject felt comfortable with the system and the graphic user interface (GUI), the experiment procedures began. The participants were instructed to track the gray line by maintaining the red dot directly on top of it as accurately as possible. A count down of 3 seconds was displayed on the left side of the screen signaling the beginning of the trial. During the 3 seconds prior to the experiment beginning, a torque was applied to the handle and its magnitude slowly increased until 0.5 seconds before the first movement in the desired signal, when it would reach and then maintain the desired torque throughout the rest of the trial. The desired torque was controlled using a feedback loop. Each trial ran for 13 seconds, not including the 3 second count down to initiation. Between each trial the subjects were give as much time as they needed before beginning the next trial.

The experiment consisted of a series of nine signals, which were repeated eight times with different forces applied to the wrist. There were three signals with step functions, three with ramp functions, and three with periodic functions. The step functions were steps of amplitude 15°, 30°, and 45°. These amplitudes were chosen because most ADLs are done within the range of 10° of flexion and 35° of extension [11]. The ramp functions had slopes of ±8 deg/s, ±16 deg/s, and ±24 deg/s. While testing the ramp tracking ability of a feline controlled using FNS it was shown that its tracking response approached a step response when the slope of the ramp exceeded 16 deg/s [12]. To test if this would be similar in functional humans the previously mentioned slopes were used. The three periodic signals consisted of predictable and unpredictable motion of different amplitudes. The predictable signals were $18 * \sin(t)$ and $31.5 * \sin(2t)$, which corresponded to movements of amplitude ±18° and ±31.5° of handle motion, respectively. The unpredictable periodic signal was a combination of sine waves having the equation $18 * \sin(1.1t) + 4.5 * \sin(2t) + 11.25 * \sin(3t)$. These functions are shown in Figure 2.5.

The forces applied during each series of signals were proportional to the MVF of the participant. A positive load rotated the handle clockwise and was proportional to the participant's MVF in flexion, and a negative load rotated the handle counterclockwise and was proportional to the participant's MVF in extension. Four different loading cases were applied to the motor, and each was repeated so that two complete sets of data were gathered for each participant. Table 2.1 shows the MVF proportional loads that were applied for each of the eight groups of trials.

Each participant performed a total of 72 trials consisting of different applied load and different magnitudes of steps, ramps, and periodic motion. Each trail was designed to capture data for both concentric and eccentric contractions. This combined with loads being applied in different directions gives data for four different muscle contractions as shown in Figure 2.6.

2.3 References

[1] *Modern control systems,* 8th ed, Addison Wesley Longman Inc., Menlo Park, CA, 1998.

[2] D. S. Hoffman and P. L. Strick, "Step-tracking movement of the wrist in humans. 1. Kinematic analysis," *J. Neurosci.,* vol. 6, no. 11, pp. 3309–3318, 1986.

[3] D. S. Hoffman and P. L. Strick, "Step-tracking movements of the wrist in humans. II. EMG analysis," *J. Neurosci.,* vol. 10, no. 1, pp. 142–152, 1990.

[4] D. S. Hoffman and P. L. Strick, "Step-tracking movements of the wrist. III. Influence of changes in load on patterns of muscle-activity," *J. Neurosci.*, vol. 13, pp. 5212–5227, 1993.

[5] D. S. Hoffman and P. L. Strick, "Step-tracking movements of the wrist. IV. Muscle activity associated with movements in different directions," *J. Neurophysiol.*, vol. 81, pp. 319–333, 1999.

[6] A. de los Reyes-Guzmán et al., "Kinematic analysis of the daily activity of drinking from a glass in a population with cervical spinal cord injury," *J. Neuroeng. Rehabil.,* vol. 7, no. 1, pp. 41–53, 2010.

[7] A. Murgia et al., “Marker placement to describe the wrist movements during activities of daily living in cyclical tasks,” *Clin. Biomech.,* vol. 19, no. 3, pp. 247–254, 2004.

[8] C. J. Van Andel et al., “Complete 3D kinematics of upper extremity functional tasks,” *Gait & Posture*, vol. 27, no. 1, pp. 120–127, 2008.

[9] M. Noble et al., “The frequency response of skilled subjects in a pursuit tracking task,” *J. Exp. Psychol.,* vol. 49, no. 4, pp. 249–256, 1955.

[10] H. P. Birmingham and F. V. Taylor, “A human engineering approach to the design of man-operated continuous control systems,” U.S. Navy, Wash. D.C., Interim Rep., 1954.

[11] R. H. Brumfield and J. A. Champoux, “A biomechanical study of normal functional wrist motion,” *Clin. Orthop. Relat. Res.*, vol. 187, pp. 23–25, 1984.

[12] M. A. Frankel, “Control methods of asynchronous intrafascicular multi-electrode stimulation for neuromuscular prostheses,” Ph.D dissertation, Univ. Utah, Salt Lake City, UT, 2013.

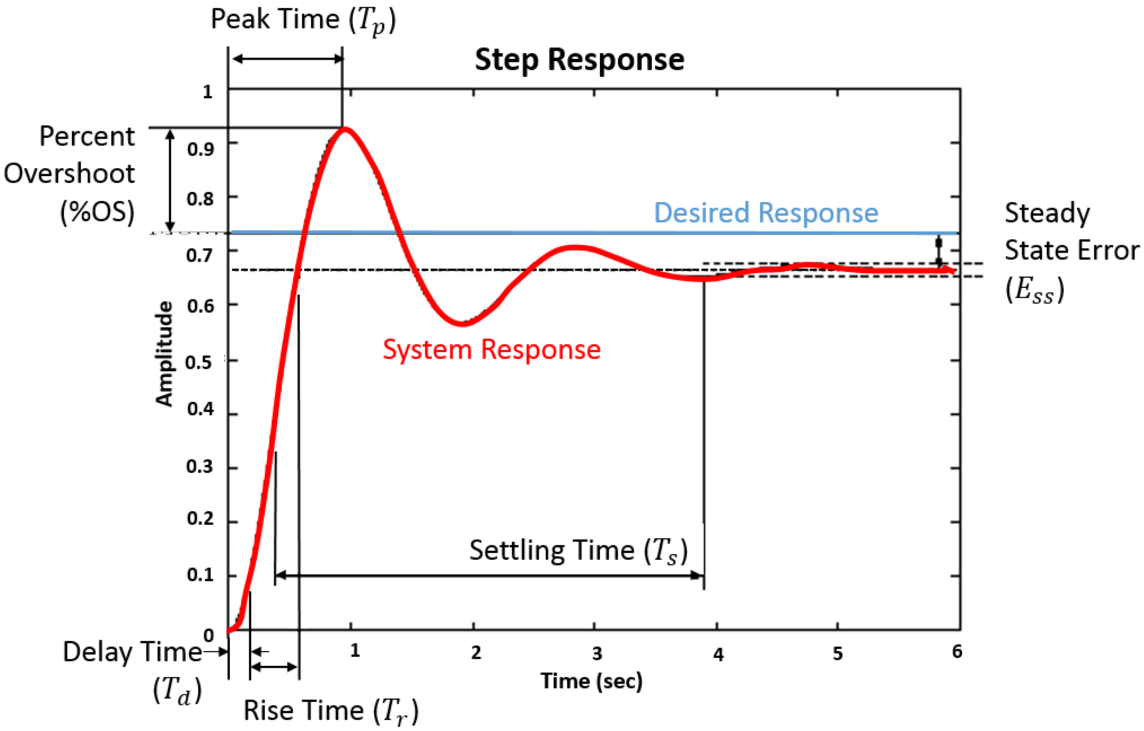


Figure 2.1: Step Response Metrics. Rise time is a measurement of the time it takes the system to travel from 10% of the final desired value to 90% of the final desired value. Settling time is a measurement of the amount of time it takes the system response to enter and not leave a ±2% band around the final desired value from the point it reaches 50% of the final value. Peak time is the amount of time it takes the response to reach its greatest value starting from the time the desired step occurred. Delay time is the time it takes the system to reach 10% of the final value from the time the desired step occurred. Delay time is not commonly used but was used here to show the response time of a human as well as the predictive nature of the human response. Percent overshoot is a measure of how far the system passes the desired value and is normalized to the amplitude of the step. Steady state error (E_{SS}) is typically measured but was not used in this project because there was no significant error. This was due to the highly adaptive nature of the human system.

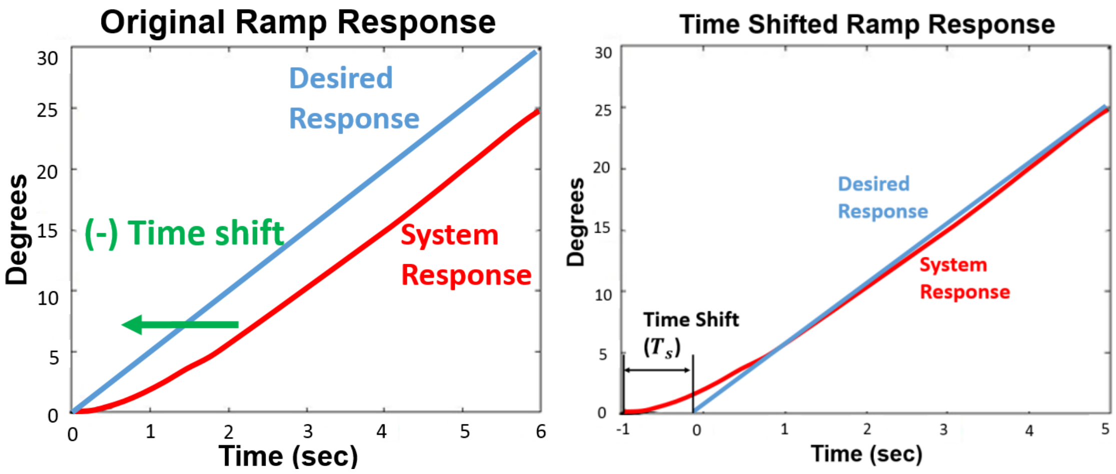


Figure 2.2: Ramp Response Metrics. The graph on the left shows a ramp function with a time delayed response. The graph on the right shows the same response that has then been shifted in time until the amplitude error was at its minimum value. Time shift was then quantified as the amount of time shift necessary to achieve the minimum value of amplitude error. Shifting the system response in the negative direction is a negative time shift, and the opposite is a positive time shift. Amplitude error was measured by summing the squared value of the error between the desired signal and the system response at each data point, then taking the square root of the total. Variance of velocity was measured as the summation of the square difference from the mean velocity at each data point during the desired position change, all divided by the number of data points during the same change. All of these metrics were quantified for each half of the trail separately.

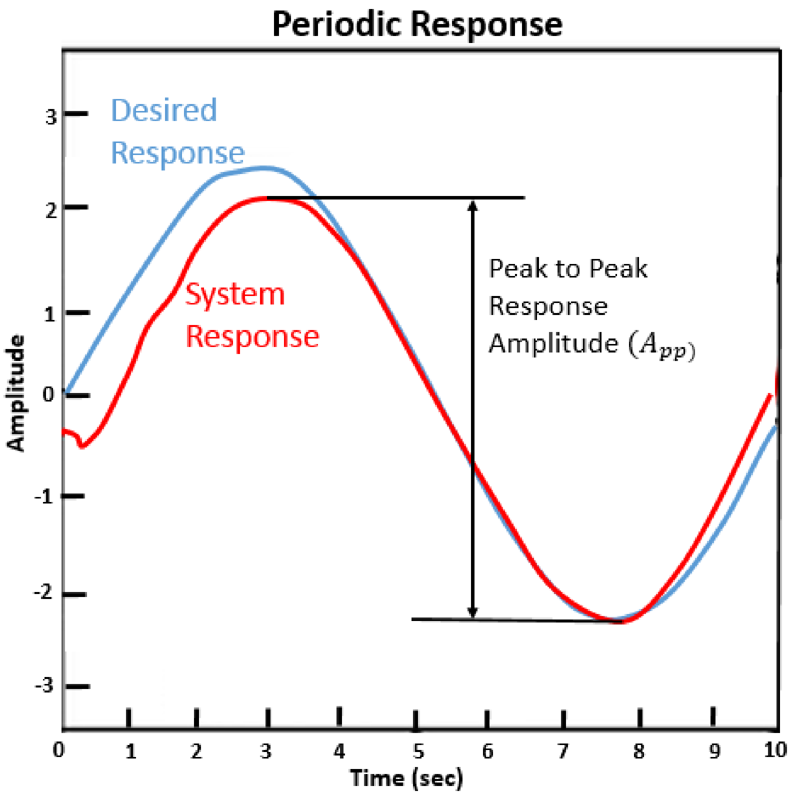


Figure 2.3: Periodic Response Metrics. Peak to peak response amplitude was measured using only the values recorded during the first period of the sine wave. The participant's minimum and maximum values were recorded, and the magnitude was compared to the magnitude of the desired peak to peak response to get the metric as a percent. This was done to compare the different trials even though they used different amplitudes. Similar to the ramp metrics the sine response was shifted in time until the amplitude error was at a minimum. Time shift was then quantified as the amount of time shift necessary to achieve the minimum value of amplitude error. Amplitude error was measured by summing the squared value of the error between the desired signal and the system response at each data point, then taking the square root of the total. The values before the signal initiation were ignored, and for programing purposes the last 0.75 second were also excluded. Variance of velocity was measured as the summation of the square difference from the mean at each data point all divided by the number of data points. Traditionally phase delay (ϕ_d) would be used as a metric; however, the human participants were highly adaptive and there was no significate phase delay.

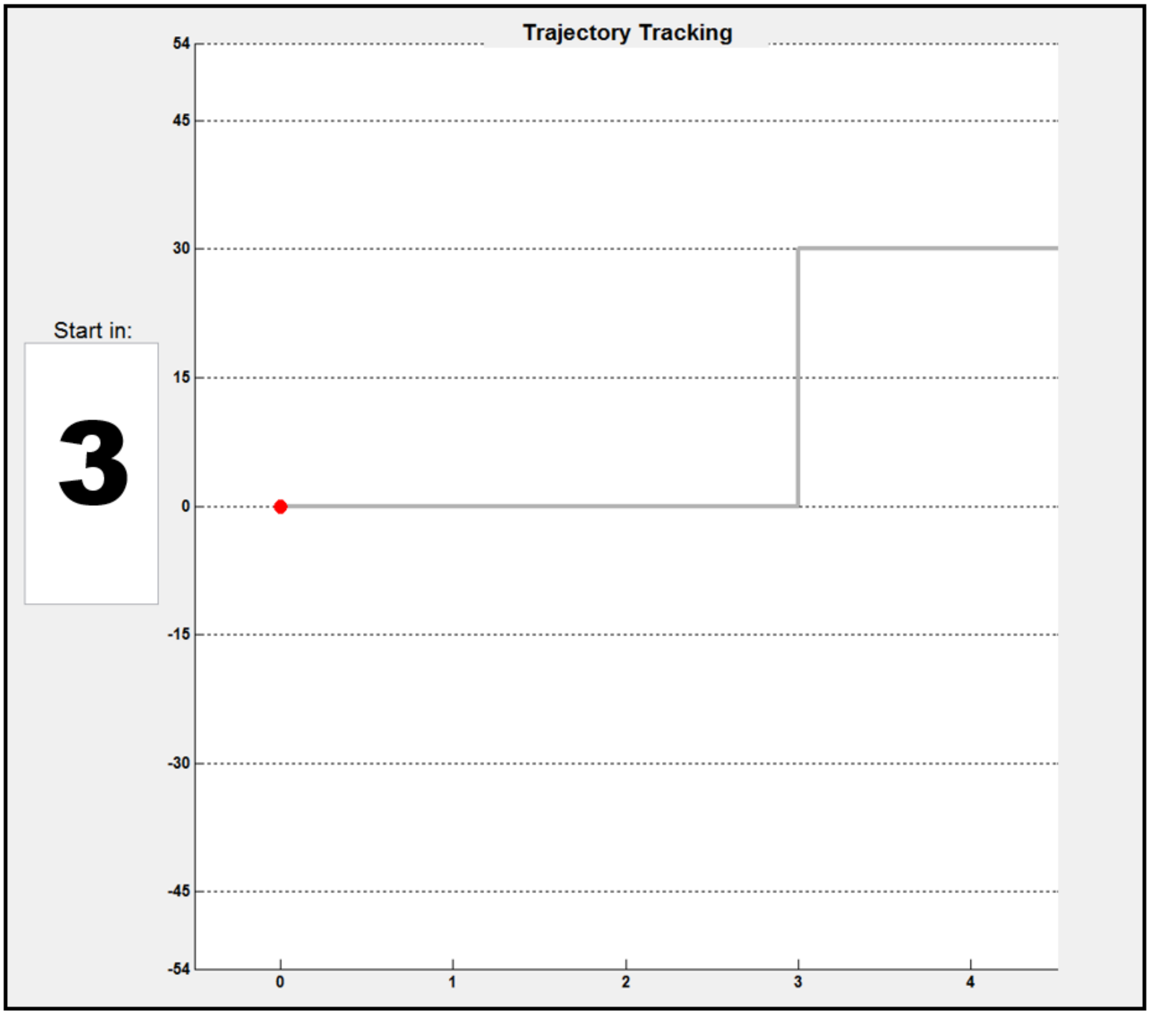


Figure 2.4: Participant's GUI. The graphic user interface (GUI) as seen by the study participant. The box on the left provided the participant a countdown to the trial initialization. The red dot's vertical position was controlled by rotating the wrist device handle, and the red dot's horizontal position on the screen remained fixed. The units of the horizontal axis were seconds, and the units of the vertical axis were degrees, which corresponded to the rotation of the device handle.

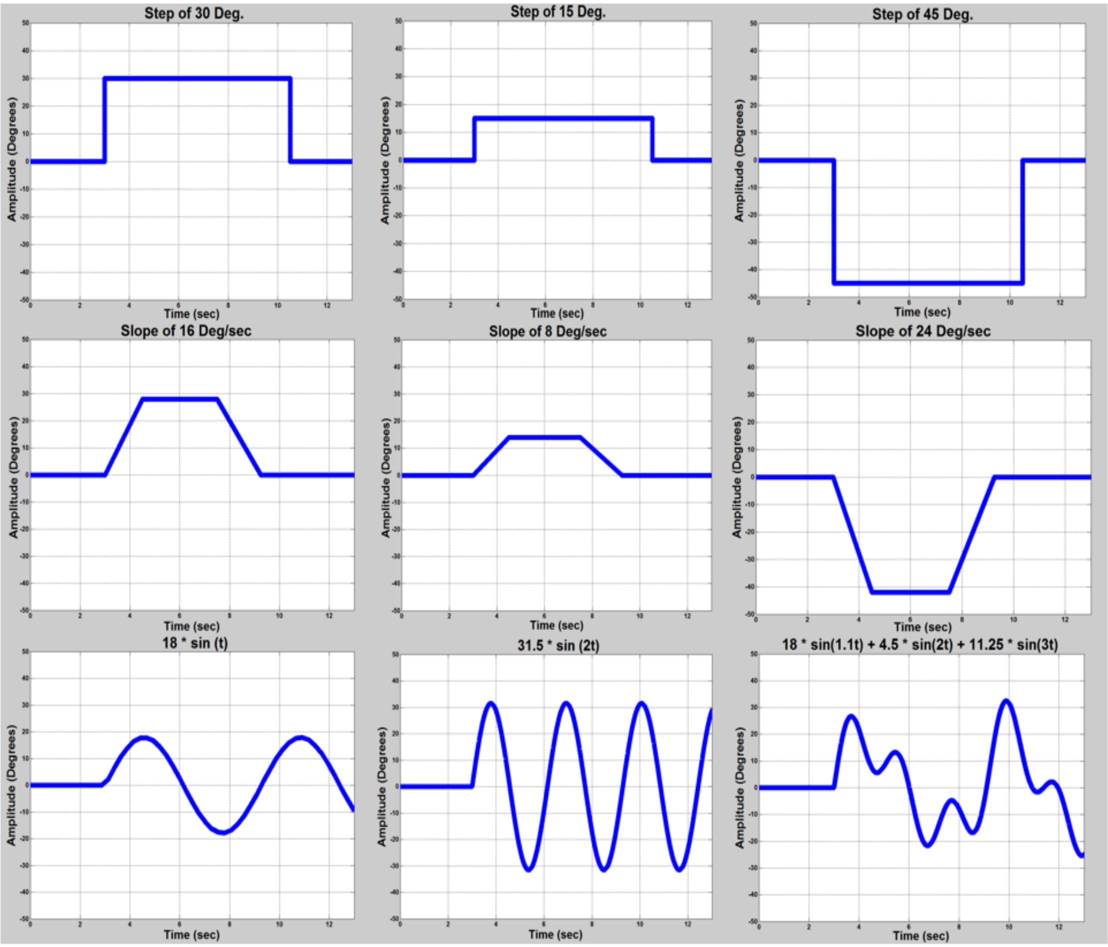


Figure 2.5: Experiment Signals. The nine signals tracked during the experiment. This set of nine signals was repeated for each of the different loads show in Table 2.1. The trials were performed in the order shown starting at the top left and ending at the bottom right.

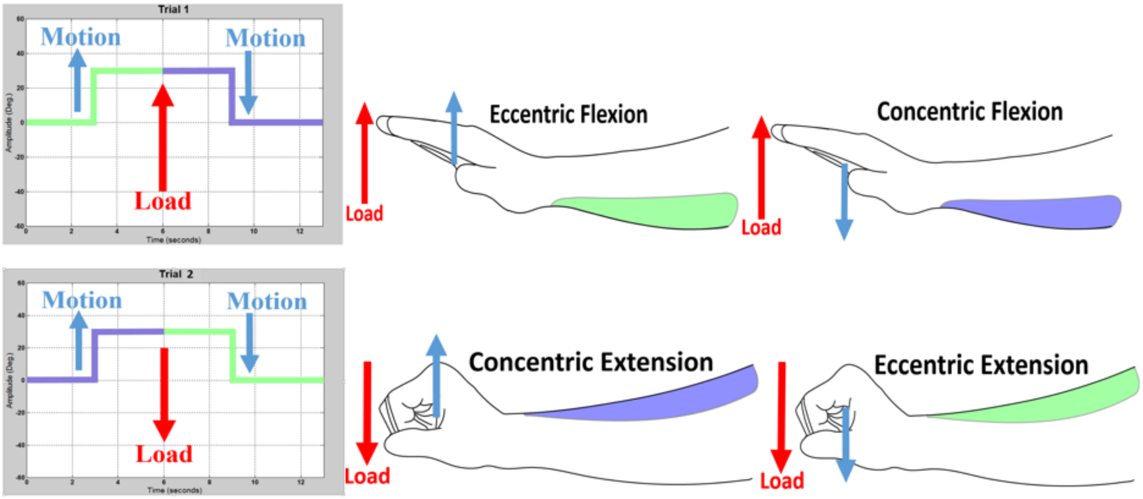


Figure 2.6: Muscle Groups. The graphs on the left show a step function that is being tracked with two different loads applied in different directions. The muscle contractions used during the tracking exercise are shown by the pictures of arms to the right of each graph. Changing the direction of the load changed which muscle groups were activated, while changing the direction of motion changed the type of muscle contraction used.

Table 2.1: Trial Groups by Weight. The order of the trial groups by load. The same force is applied for all nine signals within a group. The force is shown proportional to the maximal volitional force of the subject. F (flexion), E (extension). A positive load lifted the hand up, and a negative load pulled the hand down.

Group	1	2	3	4	5	6	7	8
Force	**0.25** F	**0.125** F	**-0.125** E	**-0.25** E	**0.25** F	**0.125** F	**-0.125** E	**-0.25** E

CHAPTER 3

EQUIPMENT DESIGN

3.1 Design Specifications

The equipment used during this research was designed with the goal of capturing the output control characteristics of a normal human wrist. To do this, normal human capabilities were taken into account while designing the mechanical, electrical, and software systems. The mechanical and electrical systems needed to be capable of matching normal human strength, and the mechanical and software systems needed to be fast enough to match human speeds. Safety was a critical concern while designing these systems because since they matched human capabilities, they were also capable of causing injury to the participants.

The major human capabilities taken into consideration were torque, speed, and range of motion. Maximum human wrist torque for average people has been shown to be around 20 Nm (177 in lb) [1], [2]. Most often this is measured with the wrist at 0° flexion and extension, the elbow at 90° of flexion and the shoulder with 0° of rotation [3], [4]. Fast wrist speed has been suggested to be around 180 deg/s [5]. The maximum range of motion for flexion and extension are 98° and 78°, respectively [6]. This was measured with the hand open, which is greater than the range of motion with the hand closed, which was used in this experiment. Compiling these metrics provided the performance requirements for the equipment.

3.2 Equipment Specifications

A Maxon RE 40 DC motor (part number (PN) 148867) was used and provided a stall torque of 2.42 Nm (21.4 in lb) and a no load speed of 7580 RPM (45480 deg/s). The motor was coupled with a Maxon planetary gearhead GP 42 c (PN 203131) to achieve both the required speed and torque. The gear ratio of approximately 230:1 increased the stall torque to about 556 Nm, but it was consequently reduced to 355 Nm since the gearbox had a maximum efficiency of 64%. While the torque was affected by the gearbox efficiency the speed was not, and the no load speed was simply reduced by the gear ratio to about 198 deg/s. The torque, speed curve is shown in Figure 3.1. This combination of motor and gearbox was selected because it had the capability of achieving a torque of 20 Nm at a speed of 180 deg/s. This is far above what a human is capable of since the reported limit of 20 Nm of torque was an isometric measurement.

A position sensor was used to collect angular position in degrees and to calculate the angular velocity of the participant's wrist, which was used for safety reasons that are described in the safety section. The sensor was a Maxon encoder HEDL 5540 (PN 110512). The encoder is connected to the back of the motor and reads 500 counts per rotation of the motor. This translates through the gear train to 115,000 counts per turn of the output shaft. This is a large resolution which gives high precision position measurement. A Texas Instruments line receiver chip (PN MC3486N) was used to convert the output signal into a more perfect square wave. This improves the accuracy of reading the rising and falling edges of the signal.

A Futek TFF400 torque sensor (PN FSH02596) was used to measure the torque being applied to the participant's wrist. The load capacity was 24 Nm (200 in lb) with a safe overload of 36 Nm (300 in lb). The output of this sensor is a voltage that was amplified by a Futek amplifier model CSG110. The ±10 VDC signal was routed to the data acquisition (DAQ) card and used in a feedback control loop to maintain a constant torque throughout the experiment. Equation 1 shows the torque that was being measured by the torque sensor. Equation 2 is the torque being applied at the point contacting the human participant. This is different from the torque being sensed because of gravity acting on the handle and the inertia of the torque sensor and handle.

$$\tau_{sensor} = N\left(\tau_m - b\dot{\phi} + Csign\left(\dot{\phi}\right) - J\ddot{\phi}\right) \quad \textbf{(1)}$$

$$\tau_{arm} = \tau_{sensor} - \frac{1}{N}\left(I\ddot{\phi} + |r_{01}|^2 m\ddot{\phi}\right) + mgr_{01}\sin(\phi/N) \quad \textbf{(2)}$$

Custom mounts were machined out of aluminum to hold the motor and the output handle bearing. The torque sensor acted as a coupler between the motor shaft and the output shaft by means of custom fabricated mounting plates. The mechanical components were all mounted to a quarter inch aluminum base plate. The base plate allowed the system to be moved or rearranged then clamped back down to any flat surface.

Current control was selected to provide the torque feedback control because (as seen in Equation 3) it is directly proportional to torque by the torque constant. This also eliminates the issue of compensating for back electromotive force (EMF) as is needed with voltage control.

$$\tau_m = k_t I_a \quad \textbf{(3)}$$

Combining and rearranging (1), (2), and (3) provides the complete system equation of the current needed to achieve a desired output torque. This equation is shown as (4). Because the gearbox inertia, viscous friction, and columbic friction are not provided by Maxon, some of the variables in (4) are unknown. System identification techniques are typically used to solve for the unknown variables in Table 3.1, which would in turn allow for the system to be controlled completely. This was not necessary in this case since the control loop was based solely on the torque sensor's readings. Because the sensor was positioned on the output shaft of the motor and the gearbox, near the participant's hand (see Figure 3.2), the control loop compensates for the unknown characteristics of the motor. Taking the torque sensor placement into consideration, (2) becomes the governing control equation of the system. Gravity compensation and inertia compensation eliminate all other variables from the equation so that the torque being measured at the sensor is proportional to that being experienced at the handle.

$$I_a = \frac{t_{arm}}{Nk_t} + \frac{1}{k_t}\left(b\dot{\phi} - Csign(\dot{\phi}) + J\ddot{\phi}\right) + \frac{1}{k_t N^2}(I\ddot{\phi} + |r_{01}|^2 m\ddot{\phi}) + \frac{mgr_{01}}{Nk_t}\sin(\phi/N) \quad \textbf{(4)}$$

The electrical system was comprised of the power supply, DAQ card, servo drive, and the computer, as shown in Figure 3.3. To provide the power for the system a TDK-Lambda LS200-24 power supply was connected to the wall outlet and transformed the 120 V, 20 A wall current to 24 V, 8.4 A. This provided sufficient voltage for the motor but more current than the motor could safely handle. A current saturation limit of 6 amps was established using the servo amplifier. This prevented the pulse width modulated (PWM) current being sent to the motor from ever going above 6 amps. A National Instruments (NI) 6323 DAQ card read the analog signal from the torque sensor and the digital signal from the encoder and provided an analog output signal of the ±10 V needed to command the motor. This output signal was used by an Advanced Motion Control 30A8 PWM servo driver to route the desired current to the motor from the power supply as a PWM signal with a switching frequency of 22 kHz.

MathWorks Matlab 2014a was used to create the GUI, which the participants used to complete the experiments. MathWorks Simulink was used for controller design, and the GUI controlled the inputs to the Simulink models. Event listeners were used to display the position of the participant's wrist on the GUI in real-time and collect this data for later analysis. The participant's GUI is shown in Figure 3.4. The Matlab code and complete Simulink models are found in Appendix C and D.

Real-time control was important for the stability of the controlled system and so that the participants would be viewing their input to the system at the same time they were performing the experiment. Real-time control was accomplished using a number of MathWorks toolboxes, including Windows Real-time Target, which prioritized the Simulink model; Matlab coder, which compiled the Matlab .m file into C; and Simulink coder, which compiled the control model into C. With the code running in compiled C and prioritized on the computer's processor, the code is able to provide a sample rate of about 200 Hz.

3.3 PID Tuning

Two types of feedback control loops were used for the experiment, torque and position. Proportional integral derivative (PID) control was used in all of the control loops, which are shown in Appendix D. The PID controller was augmented in many of the controllers with gravity compensation, virtual walls and a startup ramps. All of these were utilized to provide accurate torque and position control while maintaining a natural feel to the experiments that made the participants both comfortable and safe.

Tuning of the PID controller gains for torque control was performed by abutting the output handle against a solid surface and applying a desired step in torque to the system. A PID gain block was implemented in series in the feedback loop with the integral and derivative gains set to zero. The proportional value was set low at 0.01 and increased until the measured torque was within 5% of the desired torque. The integral value was then increased until the measured torque was driven to the desired torque value by the accumulating error. Finally the derivative value was increased to reduce the number of oscillations required to drive the system to the desired torque. This process is shown in Figure 3.5.

It was necessary for the comfort and safety of the participants to begin each experiment with the participant holding the output handle with zero torque being applied then increase the torque slowly over three seconds to the desired torque. However, this had the adverse effect of causing a large amount of error to build up over the first few seconds of the experiment. This accumulated error was multiplied by the integral value and drove the control more quickly to the desired value, causing the system to overshoot. Once the system overshot the desired value, all of the accumulated error needed to be "unwound" for the integral gain to take its desired effect. To correct this issue and increase the PID response, anti-windup was used to zero the accumulated error every time the threshold of the desired torque was crossed.

The same tuning methods were utilized for the position controller with the only difference being that the handle was allowed to freely move to the desired angle, which was then read by the encoder. The feedback controllers were tuned so that the system could accurately and forcefully maintain a desired torque or position while being safe and intuitive for the participants. The PID gains for each controller are found in Appendix D along with the controller schematics.

3.4 Safety Considerations

Safety was a major concern for this system as it was capable of potentially causing injury. With this in mind, the system was designed with safety features in every subsystem: mechanical, electrical, and software. The mechanical safety was comprised of an adjustable hard stop shown in Figure 3.6. This stop was adjusted to physically stop the output handle from exceeding the ROM of the participants. A soft Velcro strap held the participants arm in place during the experiment, but if the participant felt discomfort, this strap could easily be torn free to release their arm.

Electrical safety included a push button that stopped current to the motor. This was held in the researchers hand during the experiments. This was done because the researcher had the best understanding of the experiment and when it was not performing as accepted. The participants were instructed to let the researcher know if they ever felt discomfort.

Software safety included saturation blocks in the Simulink model. This ensured that if any unexpected values were reached by the controller the values would be clipped to remain within a safe limit for the participants. Virtual springs were created to keep the participant within a certain ROM during the experiments. Once the participant passed the predetermined angles of -60° and 45° an opposing torque was applied to the handle proportional to their displacement beyond the threshold. This pushed the participant back within a safe range for the experiment. Finally, while participants interacted with the system a researcher that was knowledgeable of the system was always present and watching the proceedings.

3.5 References

[1] S. L. Delp et al., “Maximum isometric moments generated by wrist muscles in flexion-extension and radial-ulnar deviation,” *J. Biomech.,* vol. 29, no. 10, pp. 1371–1375, 1996.

[2] P. Hudak et al., “Reliability of isometric wrist extension torque using the LIDO WorkSET for late follow-up of postoperative wrist patients,” *J. Hand Ther.,* vol. 10, no. 4, pp. 290–296, 1997.

[3] A. E. Bialocerkowski and K. A. Grimmer, “Measurement of isometric wrist muscle strength-a systematic review of starting position and test protocol,” *Clin. Rehabil.,* vol. 17, no. 7, pp. 693–702, 2003.

[4] M. Jung and M. S. Hallbeck, “The effect of wrist position, angular velocity and exertion direction on simultaneous maximal grip force and wrist torque under the isokinetic conditions,” *Int. J. Ind. Ergonomics,* vol. 29, no. 3, pp. 133–143, 2002.

[5] P. Fong and G. Y. F. Ng, “Effect of wrist positioning on isokinetic performance and repeatability of measurement for wrist flexors and extensors,” *Physiother. Theory Pract.,* vol. 16, no. 3, pp. 169–176, 2000.

[6] *Anthropometry and Biomechanics,* Man-System Integration Standards, NASA-STD-3000, 1995.

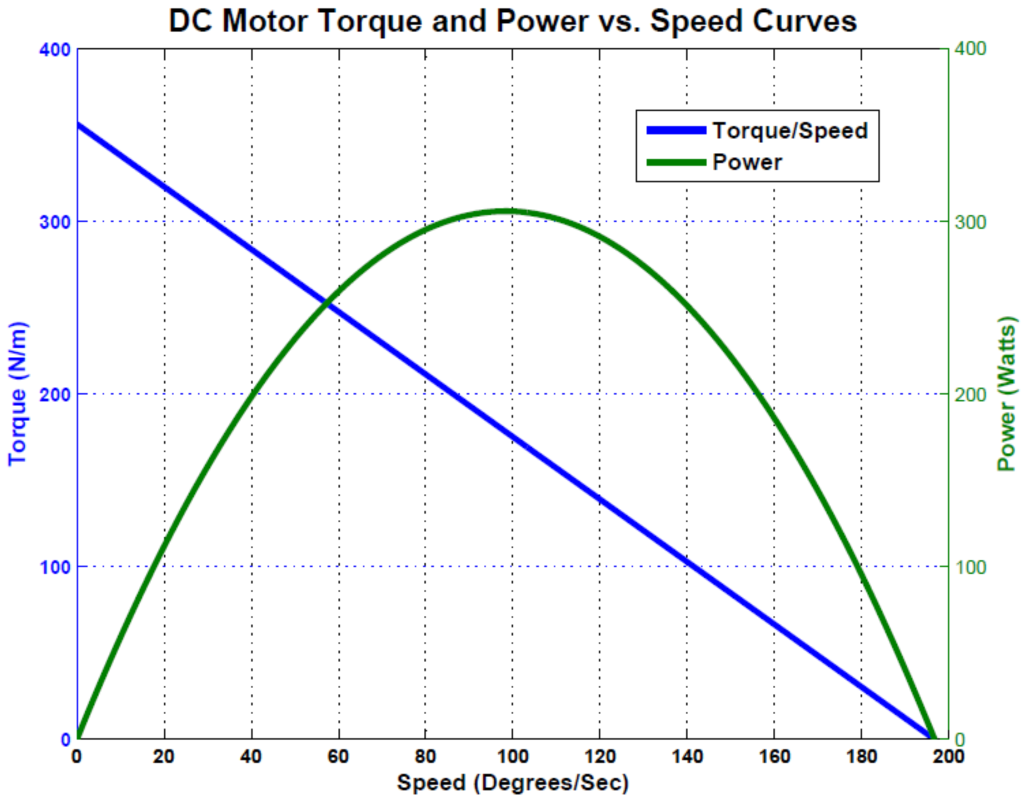


Figure 3.1: Torque Speed Curve. The torgue/speed and power/speed curves for the Maxon motor with gearbox. Stall torque was calculated with the 64% efficiency of the gearbox.

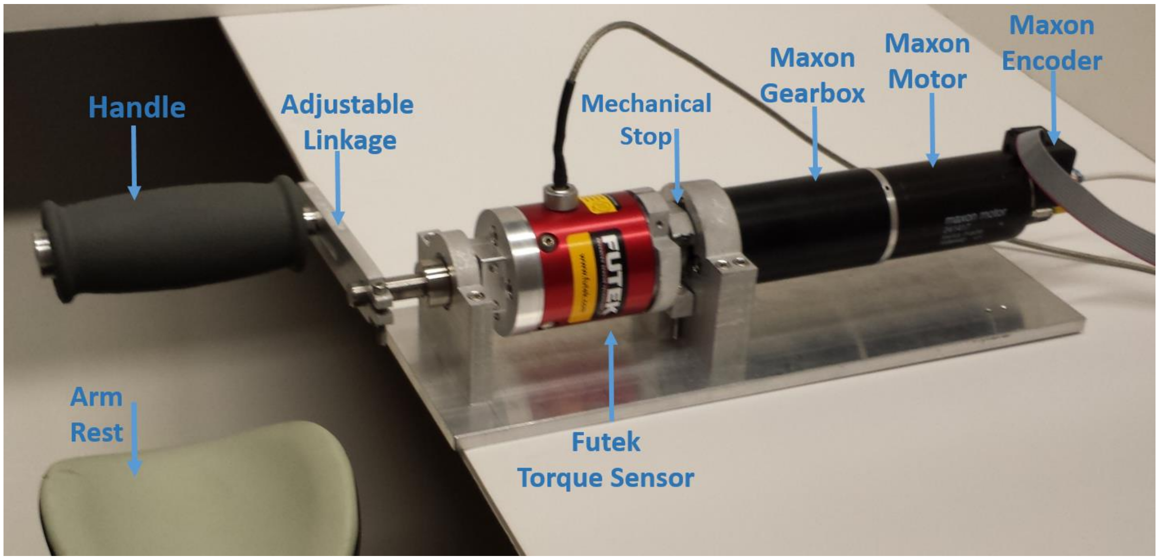


Figure 3.2: The Mechanical System. The mechanical equipment including the arm rest, handle, adjustable linkage, Futek torque sensor, mechanical stop, Maxon gearbox, Maxon motor, and Maxon encoder. The custom aluminum baseplate and mounts also shown but not labeled. Not shown are the clamps that secure the device to the table, the cover or the adjustable base of the arm rest.

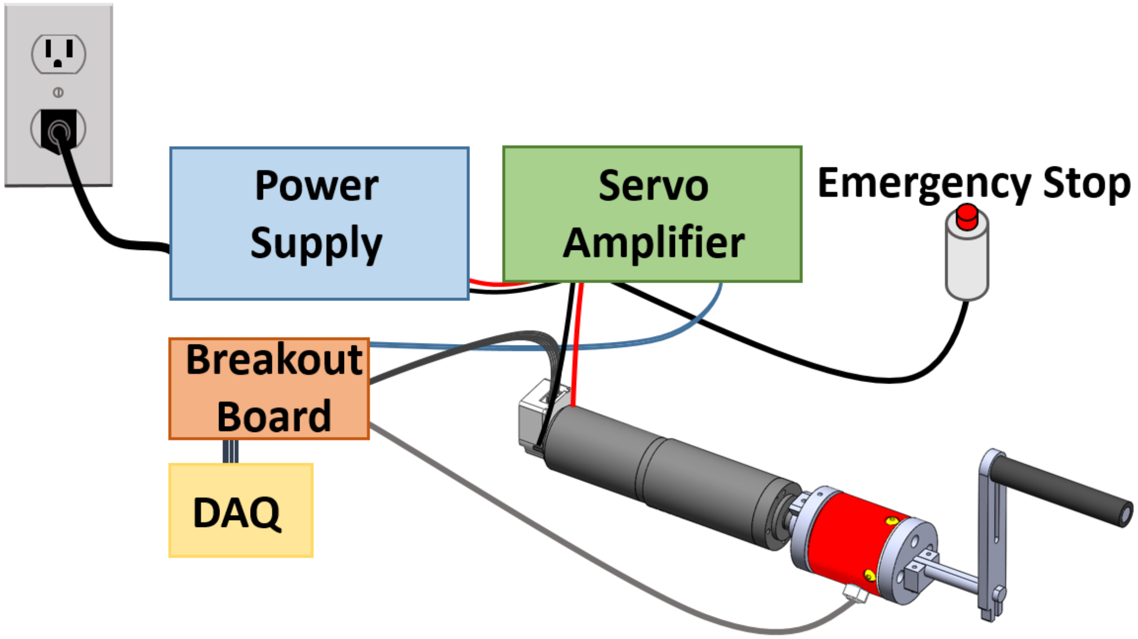
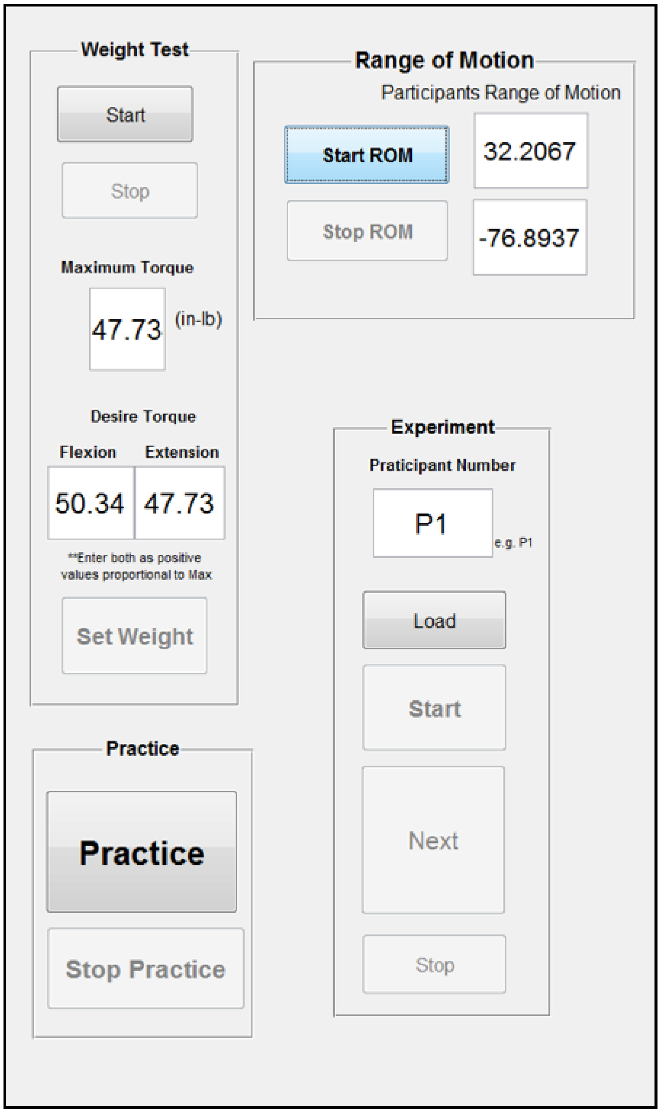


Figure 3.3: Electrical System Schematic. This is a simplification of the actual electrical system. Power from the wall outlet is converted to power usable by the system by the power supply. The power is regulated using the servo amplifier, which is controlled by the DAQ in the computer through the breakout board. Regulated PWM power is sent to the motor and the sensors send signals back to the DAQ through the breakout board. An emergency stop button cut all power to the motor. The mechanical system was created using SolidWorks.



Participants Range of Motion: 32.2067 / -76.8937

Maximum Torque: 47.73 (in-lb)

Desire Torque

Flexion	Extension
50.34	47.73

**Enter both as positive values proportional to Max

Praticipant Number: P1 e.g. P1

Figure 3.4: The Evaluator's GUI. This GUI was only seen by the researcher. The GUI was used to initiate the separate controllers that performed the tasks of measuring ROM, measuring MVF, starting a practice simulation, and running the experiments. To ensure the evaluator followed the same procedure for each participant the buttons for each step of the experiment became enabled upon completion of the previous step.

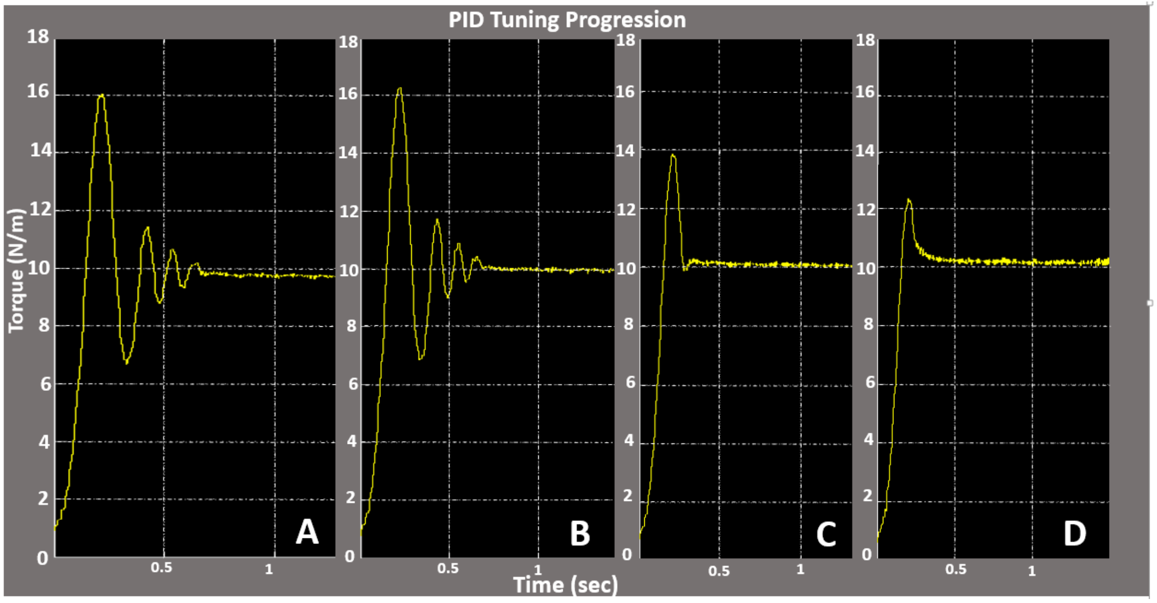


Figure 3.5: PID Tuning. These graphs show the progression of the feedback controller as the PID gains were changed. These graphs do not show each change to the PID control, only a few significant points. In each of these graphs the desired input was 10 Nm. Graph A show the point at which the controller got within 5% using only a proportional gain (P = 5). Graph B show the implementation of the integral gain and the controller reaching the desired value of 10 Nm (P = 5, I = 3). Graph C shows how the implementation of the derivative gain reduces oscillations (P = 5, I = 3, D = 0.073). Graph D show the final controller (P = 5, I = 3 D = 0.12). Further increase of the derivative gain caused chatter in the mechanical system.

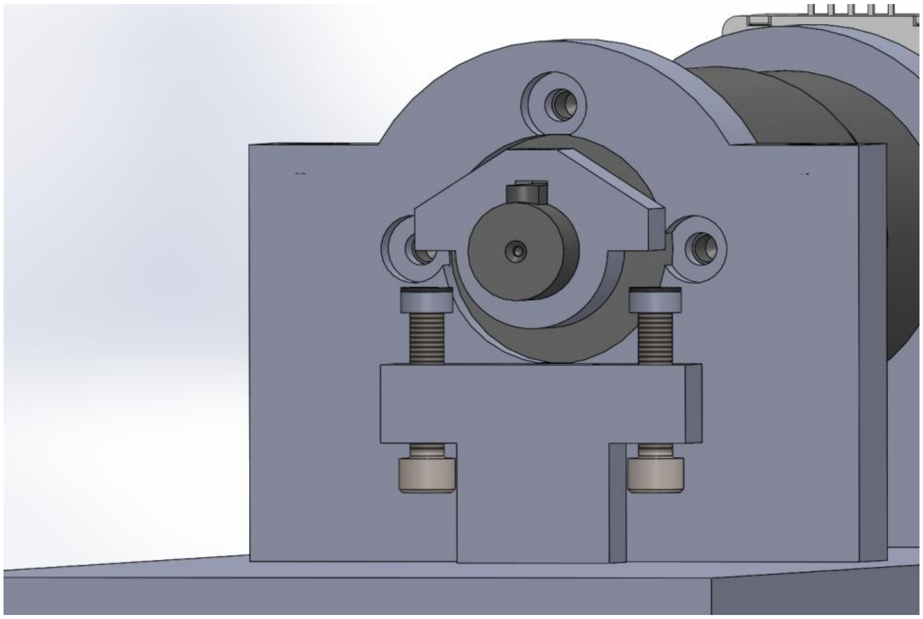


Figure 3.6: Mechanical Stop. A SolidWorks model of the hard mechanical stop. This metal stop was connected directly to the motor output shaft, which was in turn coupled to the torque sensor and the participant's handle. The design was meant to be utilized only in the event that there was a controller malfunction and that the virtual walls did not function properly.

Table 3.1: Equation Variables. This is a list of variables used in the equations below. The variable description, units, and in some cases the values are listed in the column on the right. Variables with no value present are either varying throughout the experiment or are not needed and unknown.

Variable	Explanation (Value and Units)
$\boldsymbol{\tau_{sensor}}$	Torque measured by the Futek torque sensor (Nm)
$\boldsymbol{N}$	Gear ratio (230)
$\boldsymbol{\tau_m}$	Torque being applied by the motor (Nm)
$\boldsymbol{b}$	Viscous Friction Coefficient (34.58e-7 + gearbox friction Nm*s)
$\boldsymbol{\dot{\phi}}$	Velocity as measured at the motor output (rad/s)
$\boldsymbol{C}$	Coulomb Friction Coefficient (Nm)
$\boldsymbol{J}$	Motor rotational inertia and gear inertia in the gearbox (Kgm^2)
$\boldsymbol{\ddot{\phi}}$	Acceleration as measured at the motor output (rad/s^2)
$\boldsymbol{\tau_{arm}}$	Torque at the output handle (Nm)
$\boldsymbol{I}$	Inertia of the torque sensor, adjustable link and handle (0.1859 Kgm^2))
$\boldsymbol{m}$	Mass of the handle and adjustable link (300 g)
$\boldsymbol{r_{01}}$	Center of mass of the handle and adjustable link (0.0762 m)
$\boldsymbol{\phi}$	Angular position as measured at the motor (rad)
$\boldsymbol{k_t}$	Torque constant (0.0302 Nm/A)
$\boldsymbol{I_a}$	Current being applied to the motor (A)
$\boldsymbol{V_a}$	Voltage being applied to the motor (V)
$\boldsymbol{t}$	Time (sec)
$\boldsymbol{R}$	Electrical resistance of the motor (0.299 Ω)
$\boldsymbol{L}$	Inductance of the motor (0.0000823 H)
$\boldsymbol{V_{emf}}$	Back electromotive force (V)
$\boldsymbol{g}$	Acceleration due to gravity (9.81 m/s^2))

CHAPTER 4

ANALYSIS

4.1 Data Collection

4.1.1 Physical Measurements

Seventeen people participated in the experiment. All of the participants were males between the ages of 23 and 34. Physical measurements were recorded for each participant and are listed in Table 4.1. These measurements include grasp length, ROM max, ROM min, and the participant’s MVF in flexion and extension. Hand grasp was measured with the participant gripping the wrist device handle as shown in Figure 4.1. The average hand grasp measurement was 7.26 cm (2.86 in.). Measurement methods for ROM and MVF are described in Chapter 2, Section 2.2. The participant’s ROM and MVF were measured twice, once before each attempt at the experiment, and the average of the participant’s individual values are listed in Table 4.1. The combined average maximum and minimum ROM for all of the participants were 53.45° and -89.33°, respectively, with zero being the wrist in a neutral position. The average MVF in flexion and extension for all of the participants were 11.20 Nm and 8.52 Nm, respectively.

4.1.2 Postprocessing

Each participant performed the entire experiment twice with a break in the middle. Thus two complete sets of experimental data were collected for each of the 17 participants.

The raw position data were analyzed to extract the desired metrics using Matlab code that is not included in the Appendices due to its length. Once the metrics were recorded, the metrics for each participant were averaged. The processed data for each of the 17 participants contained some 236 recorded metrics: 120 for the step functions, 72 for the ramp functions, and 44 for the periodic functions.

4.1.3 Statistical Methods

A Shapiro-Wilk normality test was performed on each of the 236 data sets. The tests were performed using IBM's Statistical Package for the Social Sciences (SPSS) [1]. In Appendix B section B.1 tables show the normality significance for the step, ramp, and periodic functions, respectively. The null hypothesis that the data sets were normally distributed was rejected for a p-value less than or equal to 0.05. Based on the results of the Shapiro-Wilk tests, parametric and nonparametric tests were used to perform analysis of variance (ANOVA) tests, which compared the results of changing only one of the independent variables while all of the others were held constant. These results are found in section B.2 in Appendix B.

Likelihood ratio tests were performed using R [2] to determine the independence of the dependent variables in each of the groups of functions: steps, ramps, and periodic. The null hypothesis of independence was rejected with a p-value less than or equal to 0.05. Correlation matrices were computed using R to demonstrate the intervariable dependence of the dependent variables. Once dependence was established a Box-Cox [3] transformation was performed to meet the normality assumption required to perform a multivariate analysis of variance (MANOVA).

The Box-Cox transformation is shown in (5) [3], where c is a constant that ensures the value inside the natural log is positive and λ is varied to optimize the normality of the transformed data. To reduce the complexity of the statistical models, the Box-Cox transformation was only performed if the distribution was significantly improved by doing so. This determination was made using univariate and bivariate quantile-quantile (QQ) plots. The least normal variable was transformed first then the QQ plots were looked at again to determine the benefits of transforming more variables.

$$y = \begin{cases} \frac{(y+c)^{\lambda}-1}{\lambda} & for\ \lambda \neq 0 \\ \ln(y+c) & for\ \lambda = 0 \end{cases} \tag{5}$$

Once the transformation was completed the MANOVA was performed to find the significance of all of the major effects and all of the interactions. A Wilks Λ distribution was used to determine significance which was accepted for a p-value less than or equal to 0.05. Finally, statistical models were derived using least squares regression to determine β values for each of the major effects and for all of the interaction terms. The β values make up the coefficients of (6), which forms the basic statistical models. The equation for the transformed variables is shown in (7).

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j \tag{6}$$

$$\frac{(y+c)^{\lambda}-1}{\lambda} = y_T = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j \tag{7}$$

Equations 6 and 7 are not interchangeable; thus only (6) should be used for the nontransformed data, and only (7) should be used for the transformed data. Caution should be taken with the transformed data as the results of the statistical model require some manipulation before the data can be interpreted. Equation 8 shows the manipulated equation that can then be directly interpreted. The derivation for this equation can be found in Appendix A.

$$y = e^{\frac{\ln(\lambda\left(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p+\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\beta_{ij}x_i x_j\right)+1)}{\lambda}} - c \tag{8}$$

4.2 Results

4.2.1 Step Function Results

A likelihood ratio independence test on the entire step function data set showed that independence was rejected with a p-value << 0.01. Table 4.2 contains the correlation matrix for the step function dependent variables.

Based on the pretransformed bivariate QQ plot in Figure 4.2, peak time and percent overshoot were transformed using the Box-Cox transformation. Peak time was transformed with λ = 0.1 and $c = 0$, and percent overshoot was transformed with λ = -0.3 and $c = 3.175$. Figure 4.2 shows the results of the transformation. Although there is a significant improvement posttransformation, only a few of the data sets can truly be considered to have a normal distribution. The benefits of further variable transformation was not deemed sufficient to lose the interpretability of the final statistical model.

After the variable independence was rejected and the data were transformed to meet the requirement of the MANOVA, the MANOVA test was performed with the result that the major effects of angle, type of contraction, and muscle group were significant. Weight did not have a significant effect within the range of the experiment. The MANOVA also showed that all of the interaction terms were significant except for angle and weight. Weight was left in the statistical model only because the interaction between muscle group, type of contraction, and weight were significant. The complete result of the MANOVA test for the step function data is found in Table 4.3.

The statistical model for healthy human wrist control for step tracking exercises within the bounds of the experiment is found in Tables 4.4 and 4.5. The models are of the forms of (7) and (8) and are to be interpreted based on the following criteria. The dependent variables rise time, delay time, settling time, peak time, and percent overshoot are the result y. The inputs x_1 through x_4 are the independent variables of interest, namely angle, type of contraction, weight, and muscle group, respectively. Because there are only two possibilities for type of contraction as well as for the muscle group used these variables are input into the model as Boolean values. Eccentric contraction is one and concentric is zero. Flexion is one and extension is zero. The intercept β_0 is used to compensate for the cases of concentric contraction and extension. Angle and weight are to be input as values 15 – 45 and 0.125–0.25, respectively.

The model residuals are provided in Figure 4.3 to demonstrate the quality of the model. A good model has residuals with a “random scattering of points around the zero value on the vertical axis” [4, p. 123]. Based on this criteria this model is considered a reasonably good estimate of human wrist control metrics for step tracking exercises.

Based on the result of the ANOVA and the MANOVA, a weight change in the range of 0.125 MVF to 0.25 MVF does not have a significant effect on any of the independent variables measured in the step tracking experiments. This could potentially be because the weight range did not apply sufficient load to make a significant difference. Angle was the variable that had the single largest impact on the recorded metrics. There are many other major effects and interactions that can be examined using the model provided. For instance, settling time tended to decrease as the magnitude of the step size increased. This seemed to be an aspect of the feed forward nature of the human controller. Once the step was recognized by the participant as a small one, the participants would take more time to ensure their motion was smoother. This translated to a longer time to reach the desired final step magnitude.

Due to the amount of variables in this project it is not possible to produce a full discussion on each of the metrics. For results specific to further research needs, select the variable of interest and hold the others constant. An example of how to use the model is provided in Table 4.6 and (9). Equation 9 is based on (6) with the β values substituted in from Tables 4.4 and 4.5.

$$\begin{aligned} Rise\ Time &= 0.16772 + 0.009134(35) + 0.16795(1) - 0.22921(0.25) - 0.14535(0) - 0.00215(35 * 1) \ldots \\ &-0.00305(35 * 0.25) + 0.00084(35 * 0) - 0.27946(1 * 0.25) - 0.03950(1 * 0) + 0.85788(0.25 * 0) = 0.426\ sec \end{aligned} \tag{9}$$

4.2.2 Ramp Function Results

The likelihood ratio independence test on the ramp function data sets showed that independence was rejected with a p-value << 0.01. Table 4.7 contains the correlation matrix for the ramp function dependent variables.

Based on the pretransformed bivariate QQ plot in Figure 4.4, variance of velocity and amplitude error were transformed using the Box-Cox transformation. Variance of velocity was transformed with $\lambda = 0.1$ and $c = 0$, and amplitude error was transformed with $\lambda = 0.1$ and $c = 0$. Figure 4.4 shows the results of the transformation. Although there is not as significant of an improvement as with some of the step function variables, the final decision to transform both variance of velocity and amplitude error was based on the results of the residual plots.

After independence was rejected for the dependent variables and the data were transformed to meet the requirement of the MANOVA, the MANOVA test was performed with the result that of the major effects, only slope and muscle group were significant.

Weight was close to having a significant effect, and the type of contraction was not significant. The MANOVA also showed that out of all of the interactions only type of muscle contraction and weight were significant. Since each of the test variables had a significant effect with either a major effect or an interaction they were all included in the model. The complete result of the MANOVA test for the step function data is found in Table 4.8.

The ramp model's residuals are provided in Figure 4.5 to demonstrate the quality of the model. Based on this criteria for a good model [4] this model is considered a reasonably good estimate of human wrist control metrics for ramp tracking exercises.

Based on the result of the ANOVA and the MANOVA, the type of contraction used does not have a significant effect on humans during ramp tracking exercises within the parameters of the experiment. Weight change in the range of 0.125 MVF to 0.25 MVF does not have a significant effect based on the threshold of 0.05; however, it is close. As with the step tracking experiments this may be because the weight range used in the experiment did not apply sufficient load to make a significant difference. Once again angle was the variable that had the single largest impact on the recorded metrics. This can be seen clearly from the results of the ANOVA in Appendix B.

By using the statistical model provided, trends in the metrics can be determined. For instance as the slope increases, so does the variance in velocity. Changing the muscle group does not seem to make an impact on this trend. This would suggest that the smoothness of the participant's tracking decreases the faster they move regardless of the muscle group they use. Although this is not unexpected, this example demonstrates how the model can be used to verify these trends.

Due to the amount of variables in this project it is not possible to produce a full discussion on each of the metrics. For results specific to further research needs, select the variable of interest and hold the others constant. The β values from Tables 4.9, λ, and c all plugged into the appropriate equations providing an example of how to use the ramp model which is provided in Table 4.10, (10), and (11). Equation 10 is based on (7), and (11) is based on (8).

$$\begin{gathered} y_T = 3.06553 + 0.12570(20) + 0.10314(0) + 0.74231(0.20) - 0.01034(1) - 0.00856(20 * 0) \ldots \\ -0.02013(20 * 0.20) + 0.00957(20 * 1) - 0.19597(0 * 0.2) + 0.40282(0 * 1) - 0.22716(0.2 * 1) = 5.7831 \end{gathered} \tag{10}$$

The result of (10) is then inserted into (8).

$$Variance\ of\ Velocity = y = e^{\frac{\ln(0.1(5.7831)+1)}{0.1} - 3.175} = 92.748 \tag{11}$$

4.6 Periodic Function Results

The likelihood ratio independence test on the periodic function data sets showed that independence was rejected with a p-value << 0.01. Table 4.11 contains the correlation matrix for the periodic function dependent variables.

The high correlation of three of the four variables suggests that the evaluation of one of the variables would give a reasonable model for the other two. Because of this high correlation a higher confidence level can be placed on the ANOVA. An attempt was still made to determine the feasibility of the MANOVA. The pretransformation and the post-transformation bivariate QQ plots are shown in Figure 4.6. The variables with the worst fit were transformed first, and a determination was made to transform other variables. The transformations did not have a major impact on the distribution even with all of the variables transformed. The QQ plot on the right side of Figure 4.6 shows all of the variables transformed with the Box-Cox transformation. Percent variance of velocity was transformed with λ = -0.5 and $c = 0$, amplitude error was transformed with λ = 0.1 and $c = 0$, peak to peak response amplitude was transformed with λ = 0.8 and $c = 0$, and time shift was transformed with λ = -16.3 and $c = 2$.

The QQ plots shown in Figure 4.6 contain the data sets for the slow, fast, and unpredictable periodic function trails. The unpredictable trails were a fledgling attempt at determining the difference between the control metrics of a complicated periodic motion verses a simple repeating sine function. The unpredictable data set should not be compared directly to the other two data sets as they are far more dissimilar than originally anticipated. Because of the difference two distinct groupings of data points were anticipated on the QQ plots. As can be seen in Figure 4.6 three distinct groupings can be seen in some cases. This was not anticipated and is due to the large step in frequency between the 1Hz sine function and the 2 Hz sine function. It was not previously known what would be a good progression, but now it is obvious that this was too great.

Based on the results of the correlation matrix and the QQ plot it was determined that the MANOVA would not be an accurate analysis. The results of the periodic function ANOVA are found in Appendix B in section B.2. The results of the ANOVA show that time shift was not significantly affected by any of the independent variables. This is to be expected as time shift and delay time for all of the experiments were essentially a measure of the participant's predictive nature minus their reaction time. It is not a good measure of reaction time because the participants knew when the change would happen. The ANOVA result clearly show that the frequency of the periodic motion has a significant effect on the dependent variables. This was clear in all of the variables except time shift. Amplitude error and peak to peak response amplitude are not affected by any of the independent variables except frequency. Percent variance of velocity had mixed results without enough evidence to determine a pattern.

Overall the periodic function experiment will need to be revisited. Although many things can be learned about transient and steady state control of the human wrist by the step and ramp experiments, periodic functions are more representative of typical human tasks [5] – [7]. This being said, further studies should look more deeply into the control metrics of the human wrist during periodic tracking exercises. These studies should include a much finer change in frequency, changes in amplitude, and a greater range of load being applied to the wrist.

4.3 References

[1] IBM Corp., *Statistical Package for the Social Sciences.* New York, United States, 2011.

[2] R Development Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013.

[3] G. E. P. Box and D. R. Cox, “An analysis of transformations,” *J. R. Stat. Soc.,* Series B, pp. 211–252, 1964.

[4] R. J. Freund et al., “Problems with observations” in *Regression Analysis,* 2nd ed. Academic Press, 2006, ch. 4, p. 123.

[5] A. de los Reyes-Guzmán et al., “Kinematic analysis of the daily activity of drinking from a glass in a population with cervical spinal cord injury,” *J. Neuroeng. Rehabil.,* vol. 7, no. 1, pp. 41–53, 2010.

[6] A. Murgia et al., “Marker placement to describe the wrist movements during activities of daily living in cyclical tasks,” *Clin. Biomech.,* vol. 19, no. 3, pp. 247–254, 2004.

[7] C. J. Van Andel et al., “Complete 3D kinematics of upper extremity functional tasks,” *Gait & Posture*, vol. 27, no. 1, pp. 120–127, 2008.

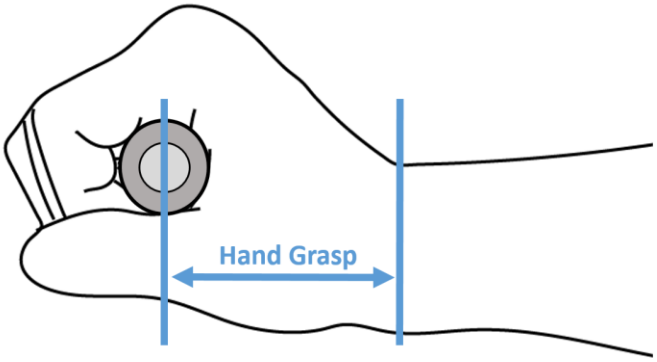


Figure 4.1: Grasp Length. Grasp length was measure with the hand closed around the device handle. The measurement was then taken from the center of the equipment handle to the radial fossa, which was used as a reference point for the rotation point of the wrist.

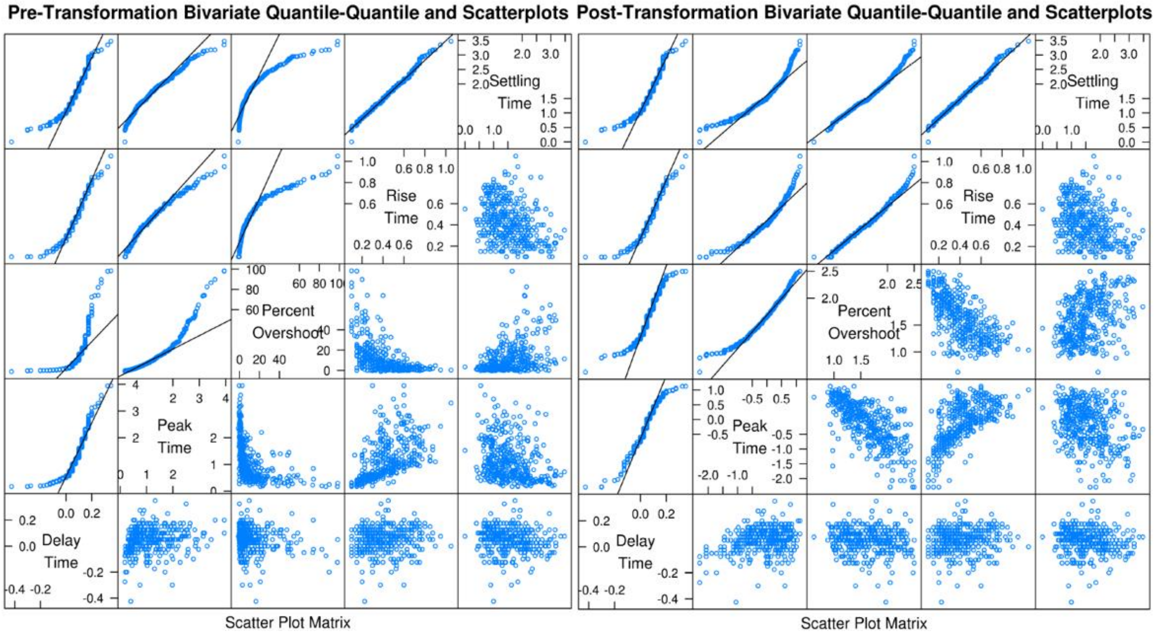


Figure 4.2: Step QQ Plots. Bivariate QQ plots of the pretransformation data and the post-transformation data sets. The upper left portion of the graph shows what normally distributed data would look like as a black line. The bottom right portion of the graph shows the distribution of the data sets. Normally distributed data look like a centered circle. These plot are to demonstrate the difference between the pretransformation data and the post-transformation data. Only percent overshoot and peak time were transformed.

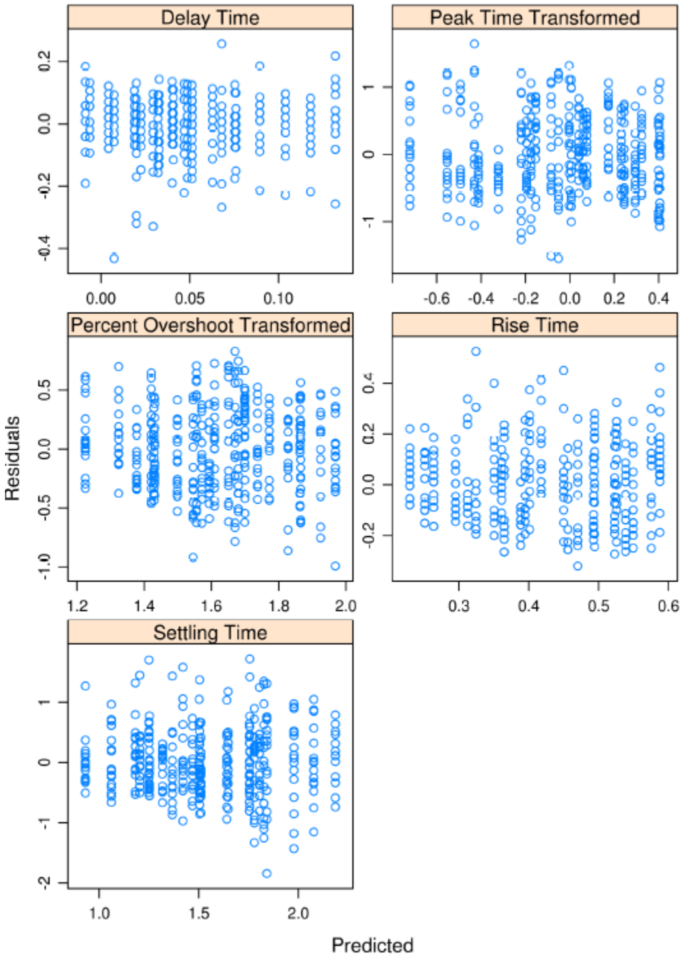


Figure 4.3: Step Model Residuals. Plot of the residuals for each of the dependent variables.

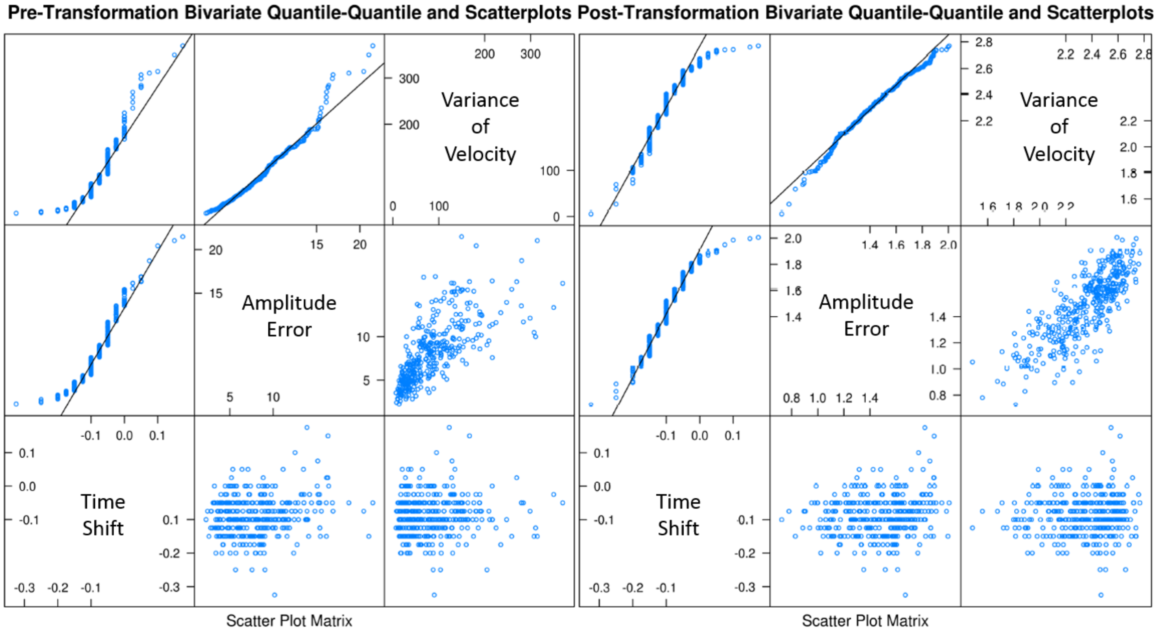


Figure 4.4: Ramp QQ Plots. Bivariate QQ plots of the pretransformation data and the post-transformation data sets. The upper left portion of the graph shows what normally distributed data would look like as a black line. The bottom right portion of the graph shows the distribution of the data sets. Normally distributed data look like a centered circle. These plot are to demonstrate the difference between the pretransformation data and the post-transformation data. Variance of velocity and amplitude error were the only variables transformed.

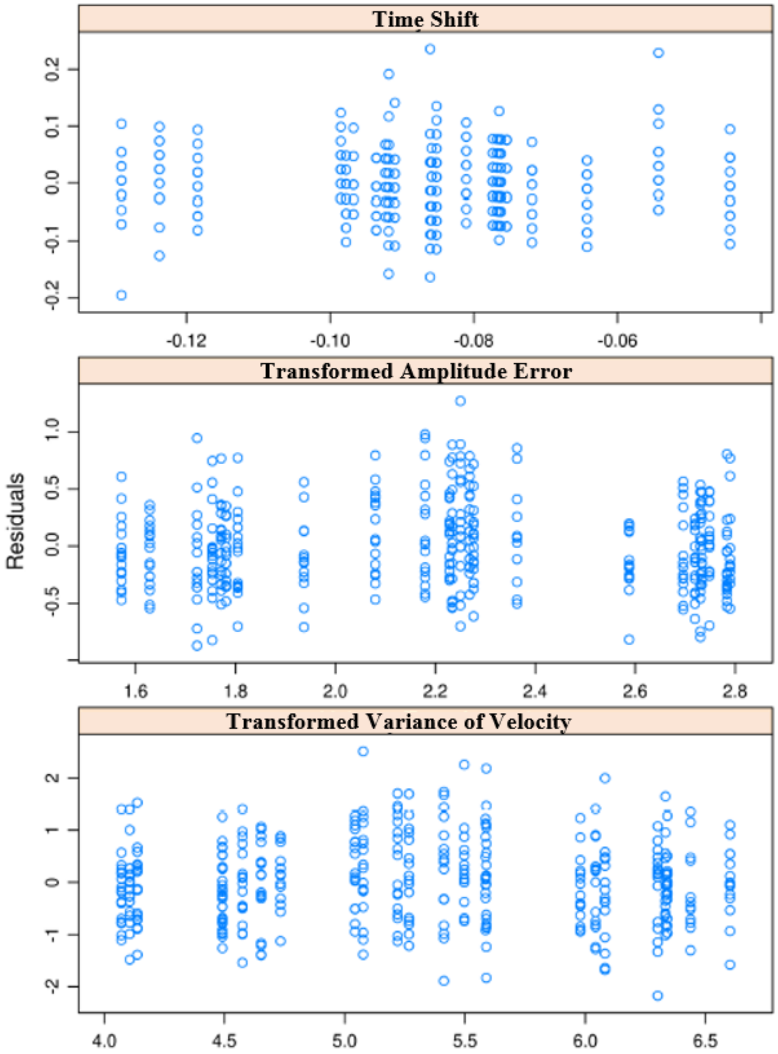


Figure 4.5: Ramp Model Residuals. Plot of the residuals for each of the dependent variables.

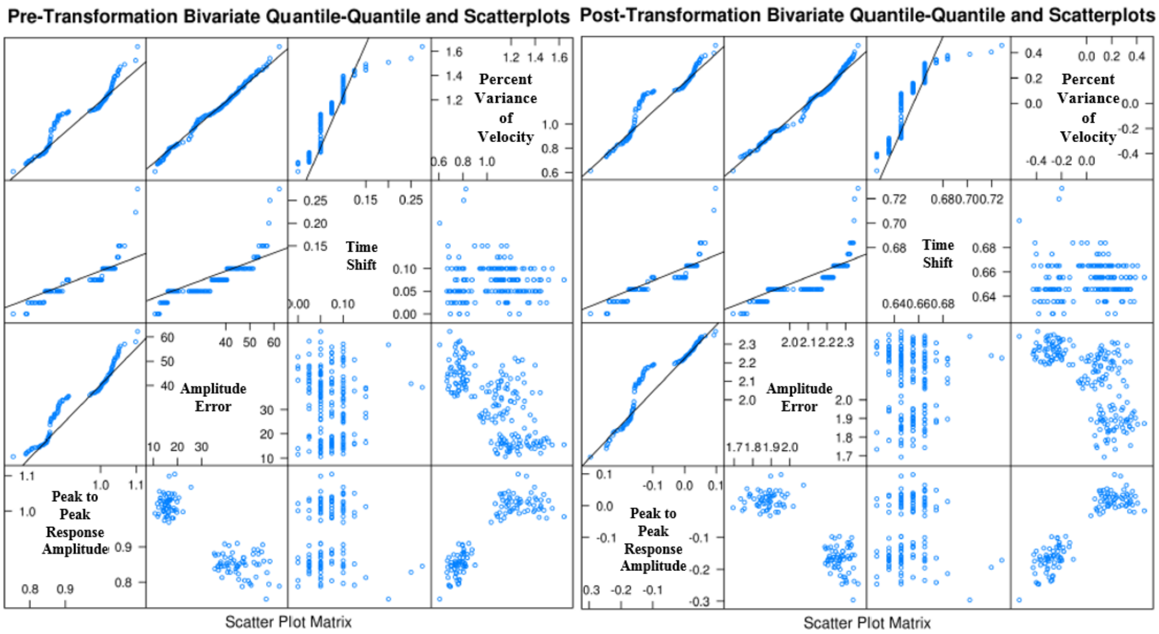


Figure 4.6: Periodic QQ Plots. Bivariate QQ plots of the pretransformation data and the post-transformation data sets. The upper left portion of the graph shows what normally distributed data would look like as a black line. The bottom right portion of the graph shows the distribution of the data sets. Normally distributed data look like a centered circle. These plot are to demonstrate the difference between the pretransformation data and the post-transformation data. All of the variables were transformed.

Table 4.1: Participant's Physical Metrics. ROM max is an average of two measurements of the angle of wrist extension while grasping the device handle. ROM min is an average of two measurements of wrist flexion while grasping the device handle. Flexion and extension are the average of two isometric measurement of wrist MVF in flexion and extension, respectively. Average values for all of the participants are shown at the bottom.

Participant	Age (Years)	Hand grasp measurement (cm)	ROM Max. (Deg.)	ROM Min. (Deg.)	Flexion (Nm)	Extension (Nm)
1	23	8	44.78	-81.61	9.40	6.73
2	27	7	64.45	-88.98	8.33	7.79
3	28	7.25	57.46	-95.83	10.89	7.26
4	29	7.25	61.34	-94.55	14.72	13.02
5	27	7.1	40.96	-93.26	11.86	8.07
6	27	7.25	52.30	-77.59	12.71	5.49
7	28	6.25	46.78	-97.00	11.88	9.92
8	28	8	52.13	-83.59	14.61	10.59
9	27	8.25	55.23	-73.44	12.26	10.56
10	27	7.5	59.26	-72.03	8.57	7.53
11	32	7.5	45.44	-90.93	13.16	9.73
12	28	6.5	42.96	-87.42	11.56	8.86
13	26	7.75	53.08	-104.54	11.24	9.55
14	34	7.25	50.29	-91.32	7.33	7.21
15	26	7	55.17	-85.26	8.36	6.98
16	25	7.5	66.73	-97.01	15.14	9.17
17	30	6	60.25	-104.29	8.44	6.44
Averages	27.76	7.26	53.45	-89.33	11.20	8.52

Table 4.2: Step Correlation Matrix. Correlation matrix for the dependent variables of the step functions. Perfect correlation is shown as 1.00.

	Rise Time	Delay Time	Settling Time	Peak Time	Percent Overshoot
Rise Time	1.00	0.11	-0.34	0.57	-0.55
Delay Time	0.11	1.00	-0.15	0.16	-0.12
Settling Time	-0.34	-0.15	1.00	-0.27	0.42
Peak Time	0.57	0.16	-0.27	1.00	-0.49
Percent Overshoot	-0.55	-0.12	0.42	-0.49	1.00

Table 4.3: Step MANOVA Results. Table of significance for all of the major effects and interactions from the MANOVA test. The first four test variables in the left column of the table are the major effects, and the last six are the interactions terms. Significance is accepted for a p-value less than 0.05.

Test Variables	p-value
Angle	<< 0.01
Type of Contraction	<< 0.01
Weight	0.4302
Muscle Group	<< 0.01
Angle: Type of Contraction	0.0185
Angle: Weight	0.8635
Angle: Muscle Group	0.5997
Type of Contraction: Weight	<< 0.01
Type of Contraction: Muscle Group	<< 0.01
Weight: Muscle Group	<< 0.01

Table 4.4: Step Statistical Model. This is the table of the β coefficients for the statistical model. None of these variables were transformed so their output results in a value for rise time, delay time, or settling time directly. Equation 6 is the equation used for all of the nontransformed data sets. The y values would be rise time, delay time, and settling time. All y value units are in seconds for this table.

	Beta	Rise Time	Delay Time	Settling Time
(Intercept)	β_0	0.16772	0.01077	2.42555
Angle	β_1	0.00913	0.00086	-0.03695
Type of Contraction	β_2	0.16795	-0.05937	0.26041
Weight	β_3	-0.22921	-0.16350	-0.83420
Muscle Group	β_4	-0.14535	0.01547	-0.01675
Angle: Type of Contraction	β_{12}	-0.00215	0.00092	0.00705
Angle: Weight	β_{13}	-0.00305	0.00359	0.04171
Angle: Muscle Group	β_{14}	0.00084	0.00033	0.00525
Type of Contraction: Weight	β_{23}	-0.27946	0.33923	-1.54843
Type of Contraction: Muscle Group	β_{24}	-0.03950	0.01659	-0.40860
Weight: Muscle Group	β_{34}	0.85788	-0.11254	1.05116

Table 4.5: Step Statistical Model Continued. A continuation of Table 4.4. This table contains all of the β coefficients for the transformed variables of peak time and percent overshoot. The major effects and the interactions are shown on the left. Peak time and percent overshoot would be the y_T values shown in (7). The units of the final results after using (8) are seconds and percent for peak time and percent overshoot, respectively.

	Beta	Transformed Peak Time	Transformed Percent Overshoot
(Intercept)	β_0	-1.11556	2.21014
Angle	β_1	0.02699	-0.01738
Type of Contraction	β_2	0.86088	-0.56781
Weight	β_3	1.05734	-0.70322
Muscle Group	β_4	-0.19325	0.07915
Angle: Type of Contraction	β_{12}	-0.01154	0.00969
Angle: Weight	β_{13}	0.00507	-0.00195
Angle: Muscle Group	β_{14}	-0.00208	-0.00073
Type of Contraction: Weight	β_{23}	-2.82752	1.91186
Type of Contraction: Muscle Group	β_{24}	0.33567	-0.11504
Weight: Muscle Group	β_{34}	0.70371	0.38354

Table 4.6: Step Model Example. Example use of the step function statistical model. The dependent variable is rise time. The desired independent variables are the variables of interest to the user of the model and are also the x values of (6) – (8).

Rise Time

Desired Independent Variables	Angle	Eccentric Contraction	Weight	Extension
x values	x_1	x_2	x_3	x_4
	35	1	0.25	0

Table 4.7: Ramp Correlation Matrix. Correlation matrix for the dependent variables of the step functions. Perfect correlation is shown as 1.00.

	Time Shift	Amplitude Error	Variance in Velocity
Time Shift	1.00	0.19	0.06
Amplitude Error	0.19	1.00	0.72
Variance in Velocity	0.06	0.72	1.00

Table 4.8: Ramp MANOVA Results. Table of significance for all of the major effects and interactions from the MANOVA test. The first four test variables in the left column of the table are the major effects, and the last six are the interactions terms. Significance is accepted for a p-value less than 0.05.

Test Variables	p-value
Slope	<< 0.01
Type of Contraction	0.2273
Weight	0.0696
Muscle Group	<< 0.01
Slope: Type of Contraction	0.7894
Slope: Weight	0.4669
Slope: Muscle Group	0.4241
Type of Contraction: Weight	<< 0.01
Type of Contraction: Muscle Group	0.0818
Weight: Muscle Group	0.0378

Table 4.9: Ramp Statistical Model. This table contains all of the β coefficients for time shift as well as the transformed variables of amplitude error and variance of velocity. The major effects and the interactions are shown on the left. Amplitude error and variance of velocity would result in an answer of the form y_T as shown in (7), and time shift would result in an answer of the form y as shown in (6). The units of the result of the time shift equation are in seconds and the results of amplitude error and variance of velocity are in degrees and degrees per second once they are transformed back to y using (8).

	Beta	Time Shift	Transformed Amplitude Error	Transformed Variance of Velocity
(Intercept)	β_0	-0.07563	1.07736	3.06553
Slope	β_1	-0.00007	0.06888	0.12570
Type of Contraction	β_2	-0.00433	-0.01399	0.10314
Weight	β_3	0.00832	0.14026	0.74231
Muscle Group	β_4	-0.01407	0.16122	-0.01034
Slope: Type of Contraction	β_{12}	-0.00067	-0.00536	-0.00856
Slope: Weight	β_{13}	-0.00050	-0.00572	-0.02013
Slope: Muscle Group	β_{14}	-0.00009	-0.00451	0.00957
Type of Contraction: Weight	β_{23}	0.03724	0.10517	-0.19597
Type of Contraction: Muscle Group	β_{24}	0.01549	0.10764	0.40282
Weight: Muscle Group	β_{34}	-0.03177	-0.06714	-0.22716

Table 4.10: Ramp Model Example. Example use of the ramp function statistical model. The dependent variable is variance of velocity. The desired independent variables are the variables of interest to the user of the model and are also the x values of (6) – (8).

Variance of Velocity

Desired Independent Variables	**Slope**	**Concentric Contraction**	**Weight**	**Flexion**
x values	x_1	x_2	x_3	x_4
	20	0	0.20	1

Table 4.11: Periodic Correlation Matrix. Correlation matrix for the dependent variables of the periodic functions. Perfect correlation is shown as 1.00.

	Time Shift	Amplitude Error	Percent Variance of Velocity	Peak to Peak Response Amplitude
Time Shift	1.00	-0.08	0.03	0.06
Amplitude Error	-0.08	1.00	-0.87	-0.91
Percent Variance in Velocity	0.03	-0.87	1.00	0.90
Peak to Peak Response Amplitude	0.06	-0.91	0.90	1.00

Table 4.12: Periodic ANOVA Results. Summary of the ANOVA results for periodic functions. For complete results see Tables B.12 – B.15 in Appendix B.

	Independent Variables		
Dependent Variables	Frequency	Weight	Muscle Group
Time Shift	Not Significant	Not Significant	Not Significant
Amplitude Error	Significant	Not Significant	Not Significant
Peak to Peak Response Amplitude	Significant	Not Significant	Not Significant
Percent Variance of Velocity	Significant	Mixed	Mixed

CHAPTER 5

CONCLUSION

5.1 Conclusions

Functional neural stimulation is quickly becoming a viable technology for the betterment of human lives. With its advent has arisen the need to directly compare designed controller results with functional human capabilities. The primary goal of this research was to provide the necessary engineering control metrics for healthy human subjects to stand as the benchmark for this comparison. By using standard engineering control metrics the provided results make it easy for neuroprosthetic developers to make direct comparisons with the results of potential control algorithms and healthy humans. This also allows for the use of classical engineering control strategies to tune the potential control algorithms to achieve the desired results.

The current goal of prosthetics is to make the user feel like they are healthy and whole. In other words, a prosthetic should act and feel just like a healthy human limb. Control algorithms govern how the prosthetic limb responds to user input and what kind of feedback is given back to the user. If the performance of a given control algorithm matches the performance of a healthy human, then part of the goal of prosthetics is accomplished.

Advances in the fields of neuroprosthetics and FMS provide hope that the neural disconnect of spinal cord injuries and strokes will be bridged by manmade electrodes.

Nerve signals will be interpreted by a computer that will determine what the prosthetic user wants to do. This research is intended to be used during the development of the interpreting algorithms and prosthetic design. Using the statistical models provided will hopefully ensure that the neuroprosthetics users function and feel like their injuries never happened.

5.2 Therapeutic Metrics

The statistical models provided in Chapter 4 may also be used as a benchmark for human wrist control in standard therapeutic methods. Current therapeutic methods rely on repetitive practice of ADLs [1]. Strength and endurance are relatively easy metrics to use for measuring progress. However, control is a much more complex metric that is difficult enough to use and that is often looked at as a byproduct of the exercises, and its progress is not measured directly. Using the same methods as were used in this research, therapists can measure the control capabilities and deficiencies of patients who have suffered a stroke. Their performance can then be compared to the provided models, and their progress can be recorded over time. Although this application of these metrics has not been explored in this work, it is one of the potential uses of quantitative models of healthy human wrist control.

5.3 Future Work

This research represents the first known attempt to develop a comprehensive model of healthy human wrist control metrics. As such there remains a lot of work to ensure that this attempt and future research in the same field are accurate and comprehensive. Validation of the provided models should be performed to ensure the participant group used is representative of males in their 20s and 30s. The number of participants in this study was limited and could be expanded in future trials to improve the distributions of the data sets and the fidelity of the models.

For these metrics to be utilized in future neuroprosthetic control design, further participant groups should include females and different age groups. Since the average age of a stroke victim is 70–71 years of age [2] it is critical that control metrics for that age group be gathered. Determining if there is a significant change in wrist control between age groups and gender would allow for prosthetics that truly react to the individual using it.

The wrist was chosen as an initial research point to test the viability of measuring human joint control. Further studies of other human joints will be necessary for neuroprosthetics to be a viable means of controlling the human body. Since each joint serves a different purpose and has different work spaces and different physical structures, it may be necessary to test each joint in turn.

As has been mentioned in section 4.6, the experiment for measuring the control metrics during periodic tracking exercises was poorly designed and requires further study in the future. There are a number of important variables that were not isolated or fully explored. These variables include frequency, load, amplitude, and the predictable nature of the periodic motion. Because of the similarity periodic motion shares with typical human use, these control metrics should be studied further, and a statistical model should be developed.

A few of the limiting factors encountered in the course of this study were the speed and torque the motor could provide. Although fast wrist motion has been considered to be around 180 deg./sec [3], the motor and controller were not able to maintain a constant load on the joint during the ballistic motion of the step functions. For this reason the loads used in the experiment were maintained in the range of 0.125 MVF to 0.25 MVF, which did not show as large of an effect as was expected in the experiment. Future studies should include higher loads to quantify the entire effect a resistive load has on the human control system.

5.4 References

[1] National Institute of Neurological Disorders and Stroke. (2013, December 16) *Post-Stroke Rehabilitation Fact Sheet* [Online] Available: http://www.ninds.nih.gov/disorders/stroke/poststrokerehab.htm

[2] U.S. Department of Health and Human Services. (2012, May) *Hospitalization for Stroke in U.S. Hospitals, 1989–2009* [Online] Available: http://www.cdc.gov/nchs/data/databriefs/db95.pdf

[3] P. Fong and G. Y. F. Ng, “Effect of wrist positioning on isokinetic performance and repeatability of measurement for wrist flexors and extensors,” *Physiother. Theory Prac.,* vol. 16, no. 3, pp. 169–176, 2000.

APPENDIX A

DERIVATIONS

A.1 System Equation Derivation

$$\tau_m = k_t I_a \tag{12}$$

Equation 12 shows that current is directly proportional to motor torque in SI units.

$$\tau_A = \tau_m - J\ddot{\phi} - b\dot{\phi} - Csgn(\dot{\phi}) \tag{13}$$

Equation 13 is the summation of the left hand side of Figure A.1.

$$cc \quad \tau_{arm} = \tau_B - I\ddot{\theta} - |r_{01}|^2 m\ddot{\theta} - mgr_{01}sin(\theta_{arm}) \tag{14}$$

Equation 14 is the summation of the right hand side of Figure A.1.

$$N = \frac{r_B}{r_A} \tag{15}$$

Equation 15 is the gear ratio

$$\tau_A N = \tau_B \tag{16}$$

$$\tau_{arm} = N(\tau_m - J\ddot{\phi} - b\dot{\phi} - Csgn(\dot{\phi})) - (I - |r_{01}|^2 m)\ddot{\theta} - mgr_{01}sin(\theta_{arm}) \tag{17}$$

Equation 17 is a combination of (13)–(16).

$$I_a = \frac{t_{arm}}{Nk_t} + \frac{1}{k_t}\left(b\dot{\phi} - Csign(\dot{\phi}) + J\ddot{\phi}\right) + \frac{1}{k_t N^2}(I\ddot{\phi} + |r_{01}|^2 m\ddot{\phi}) + \frac{mgr_{01}}{Nk_t}\sin(\phi/N) \tag{18}$$

Equation 18 is a combination of (17) and (12). This is the full system equation for current control. The equation has an output of torque on the participant’s arm for a given input current. This is not the control equation for the system, as the torque is not measured at the output but at the torque sensor. The torque being measured at the sensor is shown in (20).

$$\tau_{sensor} = N\big(\tau_m - b\dot{\phi} + Csign\big(\dot{\phi}\big) - J\ddot{\phi}\big) \qquad \textbf{(19)}$$

$$\tau_{arm} = \tau_{sensor} - \frac{1}{N}(I\ddot{\phi} + |r_{01}|^2 m\ddot{\phi}) + mgr_{01}\sin(\phi/N) \qquad \textbf{(20)}$$

Equation 20 shows the torque at the participants wrist compared to the torque measured at the sensor. Gravity compensation eliminates the last term of the equation, leaving only the inertia of the handle affecting the torque measurement. This was determined to be small compared to the inertia of the human hand and wrist, so it was neglected in the control schematic.

A.2 Transformed Statistical Model Derivation

Equation 21 shows the statistical model for the transformed dependent variables.

$$\frac{(y+c)^{\lambda}-1}{\lambda} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\beta_{ij}x_i x_j \qquad \textbf{(21)}$$

Equation 22 is a simplified version of that equation, where $y_T$is the outcome of the use of the β coefficient tables for the transformed variables.

$$y_T = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\beta_{ij}x_i x_j \qquad \textbf{(22)}$$

Equations (23) – (25) show the process of isolating y.

$$(y+c)^{\lambda} = \lambda y_T + 1 \qquad \textbf{(23)}$$

$$\lambda * \ln(y+c) = \ln(\lambda * y_T + 1) \qquad \textbf{(24)}$$

$$y = e^{\frac{(\ln(\lambda * y_T + 1))}{\lambda}} - c \tag{25}$$

Equation 26 is a combination of (25) and (22) and is the derivation required to isolate y.

$$y = \ e^{\frac{\ln(\lambda\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j\right) + 1)}{\lambda}} - c \tag{26}$$

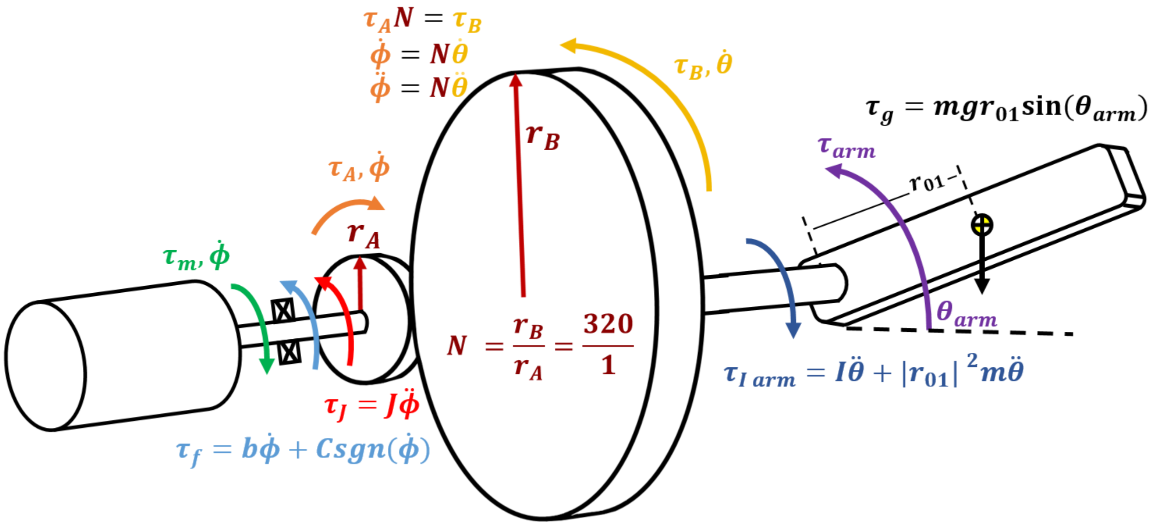

$\tau_A N = \tau_B$
$\dot{\phi} = N\dot{\theta}$
$\ddot{\phi} = N\ddot{\theta}$
$\tau_B, \dot{\theta}$
$\tau_g = mgr_{01}\sin(\theta_{arm})$
τ_{arm}
r_{01}
r_B
$\tau_A, \dot{\phi}$
r_A
$\tau_m, \dot{\phi}$
θ_{arm}
$N = \frac{r_B}{r_A} = \frac{320}{1}$
$\tau_{I\,arm} = I\ddot{\theta} + |r_{01}|^2 m\ddot{\theta}$
$\tau_J = J\ddot{\phi}$
$\tau_f = b\dot{\phi} + Csgn(\dot{\phi})$

Figure A.1: Physical System Derivation. Graphical derivation of the components of the physical system equation.

APPENDIX B

ANALYSIS TABLES AND PLOTS

B.1 Shapiro-Wilk Normality Results

Table B.1: Shapiro-Wilk normality test for the step function data. Green boxes show data that are significantly normally distributed. Red boxes show data that are not normally distributed. The threshold is 0.05.

Step Functions			Rise Time			Delay Time			Settling Time			Peak Time			Percent Over Shoot		
Muscle Group	Muscle Contraction	% MVF	15 Deg.	30 Deg.	45 Deg.	15 Deg.	30 Deg.	45 Deg.	15 Deg.	30 Deg.	45 Deg.	15 Deg.	30 Deg.	45 Deg.	15 Deg.	30 Deg.	45 Deg.
Flexion	Concentric	0.25	0.002	0.017	0.586	0.678	0.050	0.443	0.998	0.388	0.874	0.003	0.066	0.262	0.015	0.001	0.002
Flexion	Concentric	0.125	0.087	0.102	0.402	0.102	0.061	0.432	0.517	0.996	0.019	0.013	0.283	0.300	0.001	0.003	0.001
Flexion	Eccentric	0.25	0.237	0.031	0.032	0.256	0.255	0.453	0.318	0.229	0.032	0.107	0.011	0.040	0.007	0.000	0.000
Flexion	Eccentric	0.125	0.005	0.014	0.371	0.235	0.263	0.041	0.604	0.447	0.385	0.002	0.320	0.104	0.001	0.005	0.007
Extension	Concentric	0.25	0.958	0.594	0.097	0.462	0.002	0.120	0.260	0.450	0.248	0.000	0.138	0.024	0.000	0.013	0.002
Extension	Concentric	0.125	0.802	0.789	0.376	0.000	0.032	0.078	0.552	0.121	0.003	0.000	0.186	0.001	0.048	0.000	0.000
Extension	Eccentric	0.25	0.146	0.240	0.345	0.600	0.164	0.182	0.306	0.551	0.024	0.001	0.028	0.077	0.050	0.123	0.021
Extension	Eccentric	0.125	0.033	0.141	0.606	0.420	0.262	0.520	0.403	0.636	0.075	0.034	0.000	0.175	0.000	0.026	0.009

Table B.2: Shapiro-Wilk normality test for the Ramp function data. Green boxes show data that are significantly normally distributed. Red boxes show data that are not normally distributed. The threshold is 0.05.

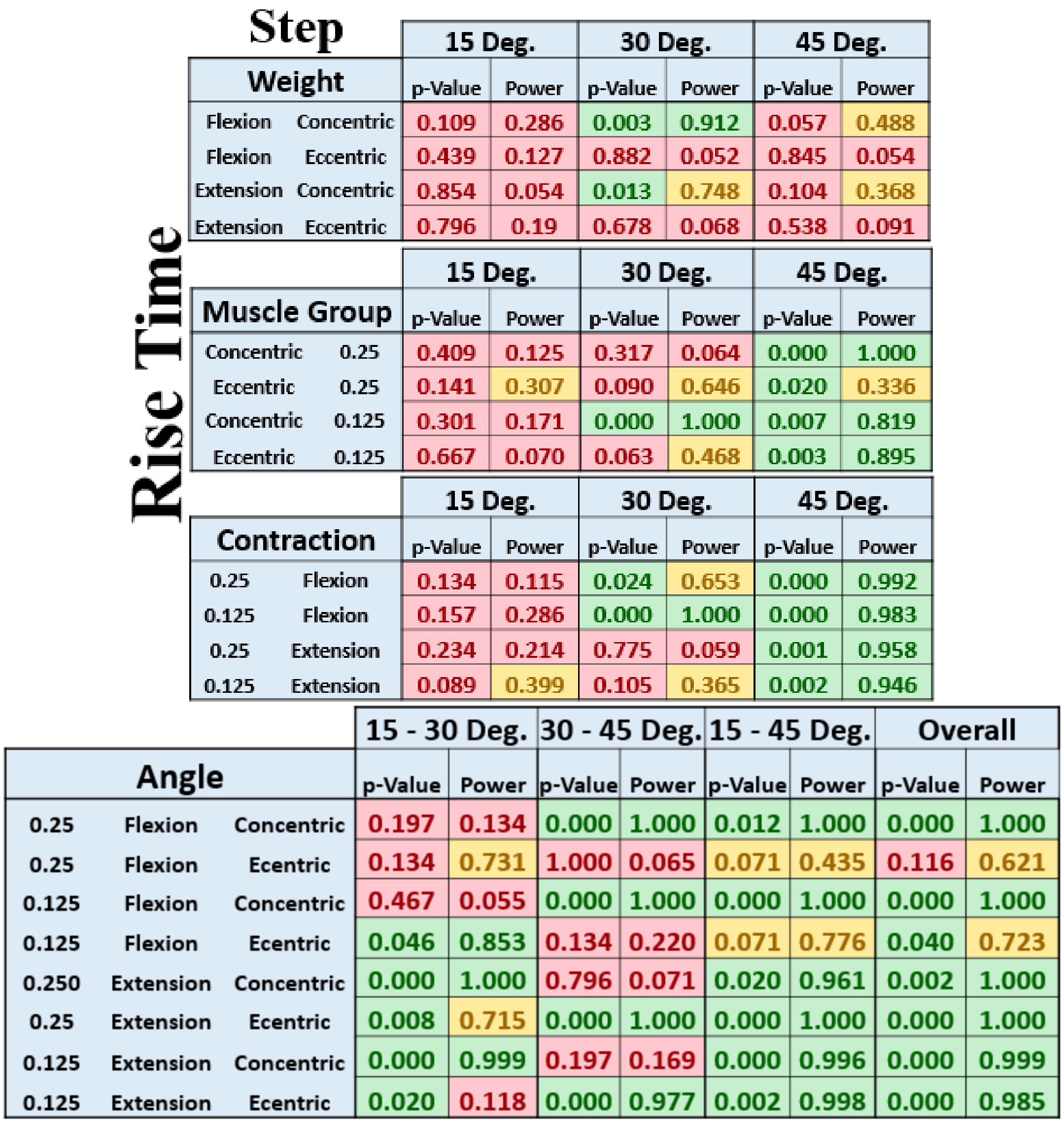
Ramp Functions			Amplitude Error			Time Shift			Variance of Velocity		
Muscle Group	Muscle Contraction	% MVF	8 Deg./s	16 Deg./s	24 Deg./s	8 Deg./s	16 Deg./s	24 Deg./s	8 Deg./s	16 Deg./s	24 Deg./s
Flexion	Concentric	0.25	0.365	0.316	0.531	0.583	0.436	0.197	0.267	0.252	0.002
Flexion	Concentric	0.125	0.035	0.304	0.001	0.207	0.057	0.051	0.009	0.129	0.772
Flexion	Eccentric	0.25	0.281	0.187	0.008	0.039	0.644	0.098	0.573	0.193	0.017
Flexion	Eccentric	0.125	0.11	0.286	0.401	0.948	0.28	0.214	0.144	0.009	0.543
Extension	Concentric	0.25	0.008	0.101	0.036	0.026	0.115	0.806	0.222	0	0.015
Extension	Concentric	0.125	0.61	0.022	0.53	0.622	0.703	0.685	0.004	0.003	0.001
Extension	Eccentric	0.25	0.226	0.085	0.026	0.743	0.034	0.243	0.015	0.15	0.114
Extension	Eccentric	0.125	0.078	0.517	0.178	0.206	0.028	0.299	0.006	0.365	0.071

Table B.3: Shapiro-Wilk normality test for the Periodic function data. Green boxes show data that are significantly normally distributed. Red boxes show data that are not normally distributed. The threshold is 0.05. Peak to Peak Response Amplitude is not measured for the Unpredictable trial.

Periodic Functions		Amplitude Error			Time Shift			% Variance of Velocity			Peak to Peak Response Amplitude	
Muscle Group	% MVF	Slow	Fast	Unpredictable	Slow	Fast	Unpredictable	Slow	Fast	Unpredictable	Slow	Fast
Flexion	0.25	0.021	0.174	0.506	0.002	0.002	0.036	0.092	0.428	0.822	0.713	0.023
Flexion	0.125	0.815	0.199	0.446	0.279	0.000	0.003	0.279	0.343	0.787	0.022	0.096
Extension	0.25	0.518	0.476	0.160	0.522	0.006	0.032	0.217	0.382	0.422	0.173	0.224
Extension	0.125	0.758	0.430	0.900	0.082	0.022	0.001	0.014	0.312	0.618	0.770	0.619

B.2 ANOVA Results

Table B.4: ANOVA tests for rise time in step functions. These analyses evaluated if the different independent variable caused a significant difference in rise time. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The angle comparison was between 15 and 30°, 30 and 45°, 15 and 45°, and an overall comparison of the data sets collected at each of the three step sizes. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Step

Rise Time

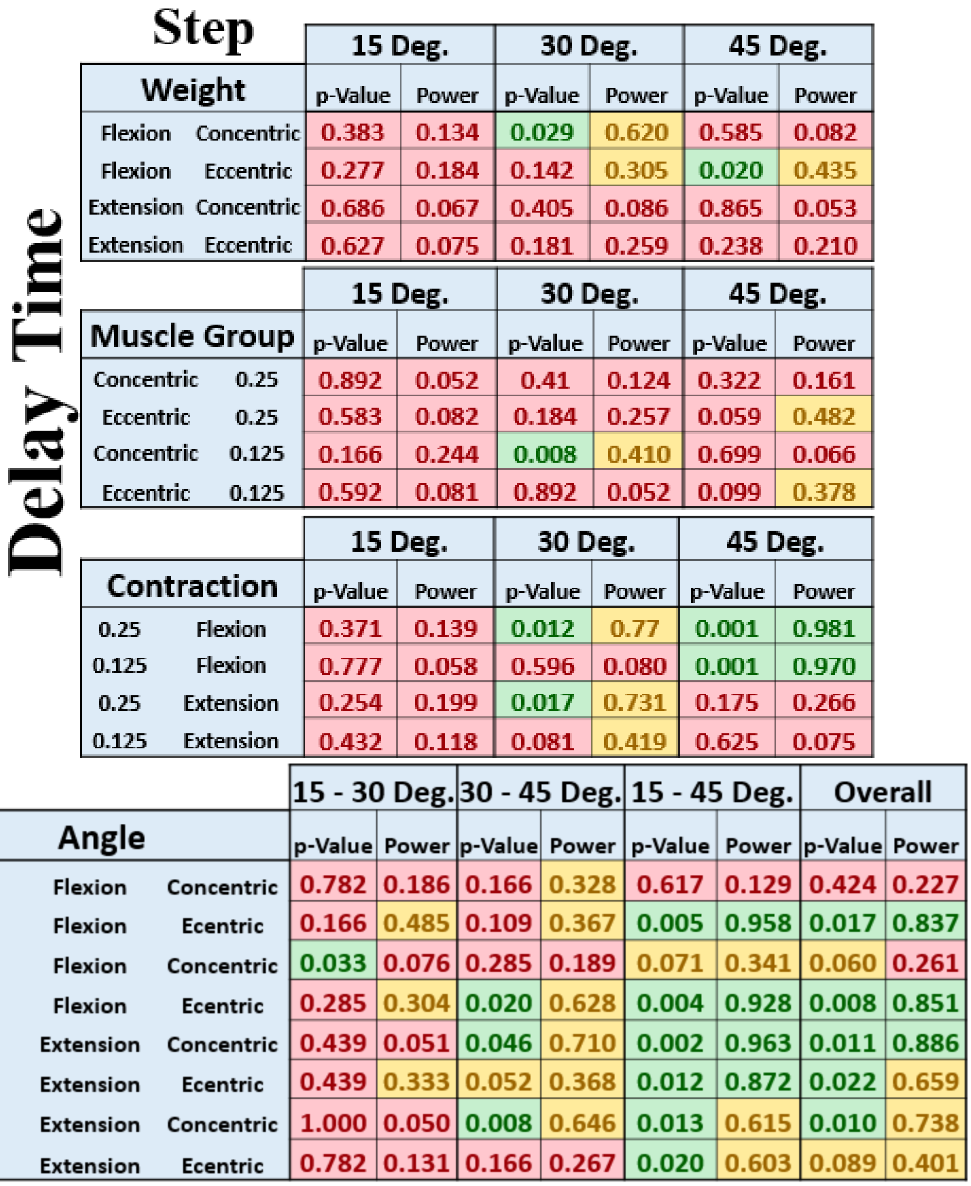
Weight		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.109	0.286	0.003	0.912	0.057	0.488
Flexion	Eccentric	0.439	0.127	0.882	0.052	0.845	0.054
Extension	Concentric	0.854	0.054	0.013	0.748	0.104	0.368
Extension	Eccentric	0.796	0.19	0.678	0.068	0.538	0.091

Muscle Group		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.409	0.125	0.317	0.064	0.000	1.000
Eccentric	0.25	0.141	0.307	0.090	0.646	0.020	0.336
Concentric	0.125	0.301	0.171	0.000	1.000	0.007	0.819
Eccentric	0.125	0.667	0.070	0.063	0.468	0.003	0.895

Contraction		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.134	0.115	0.024	0.653	0.000	0.992
0.125	Flexion	0.157	0.286	0.000	1.000	0.000	0.983
0.25	Extension	0.234	0.214	0.775	0.059	0.001	0.958
0.125	Extension	0.089	0.399	0.105	0.365	0.002	0.946

Angle			15 - 30 Deg.		30 - 45 Deg.		15 - 45 Deg.		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.197	0.134	0.000	1.000	0.012	1.000	0.000	1.000
0.25	Flexion	Ecentric	0.134	0.731	1.000	0.065	0.071	0.435	0.116	0.621
0.125	Flexion	Concentric	0.467	0.055	0.000	1.000	0.000	1.000	0.000	1.000
0.125	Flexion	Ecentric	0.046	0.853	0.134	0.220	0.071	0.776	0.040	0.723
0.250	Extension	Concentric	0.000	1.000	0.796	0.071	0.020	0.961	0.002	1.000
0.25	Extension	Ecentric	0.008	0.715	0.000	1.000	0.000	1.000	0.000	1.000
0.125	Extension	Concentric	0.000	0.999	0.197	0.169	0.000	0.996	0.000	0.999
0.125	Extension	Ecentric	0.020	0.118	0.000	0.977	0.002	0.998	0.000	0.985

Table B.5: ANOVA tests for delay time in step functions. These analyses evaluated if the different independent variable caused a significant difference in delay time. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The angle comparison was between 15 and 30°, 30 and 45°, 15 and 45°, and an overall comparison of the data sets collected at each of the three step sizes. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Step

Delay Time

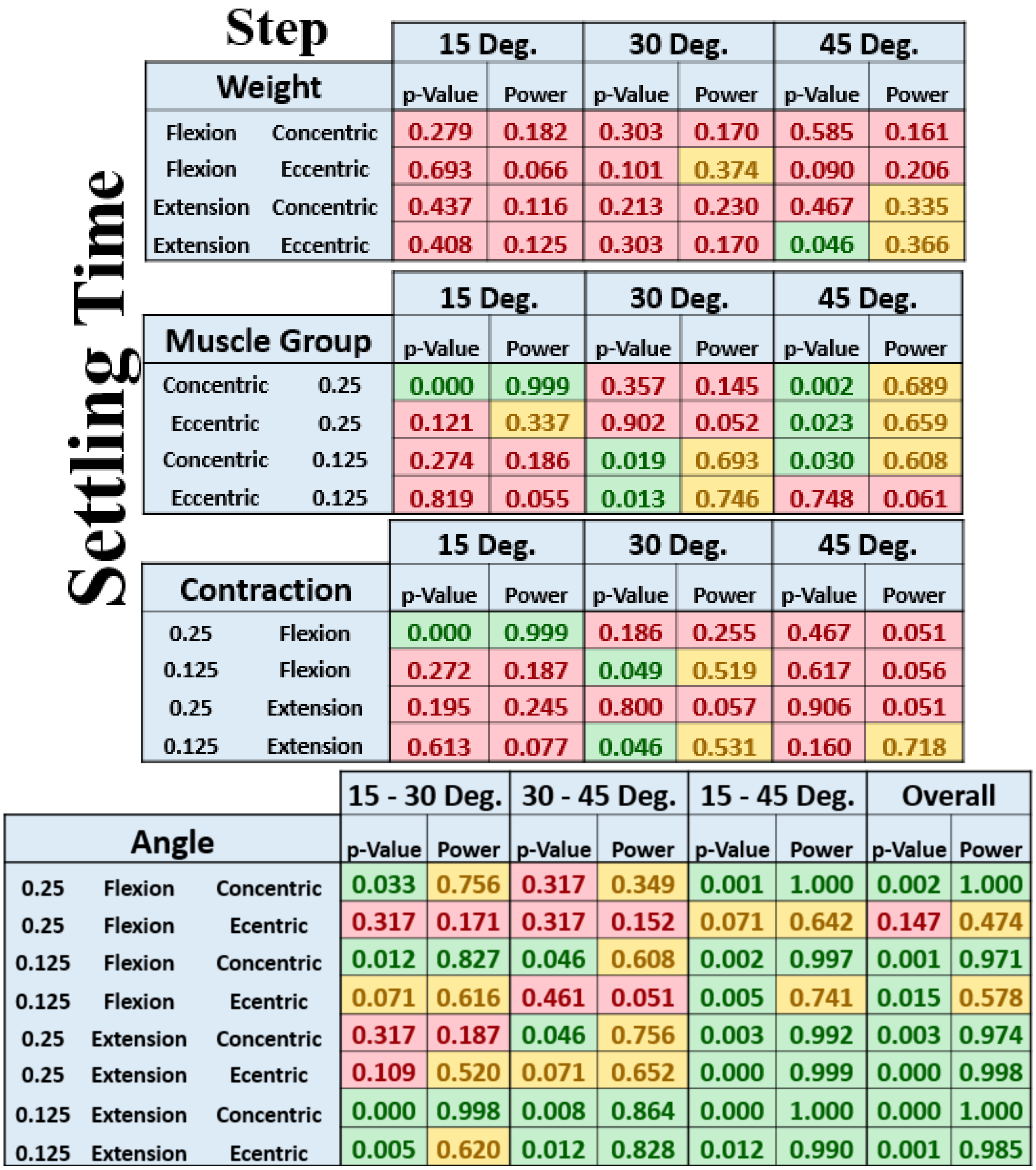
Weight		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.383	0.134	0.029	0.620	0.585	0.082
Flexion	Eccentric	0.277	0.184	0.142	0.305	0.020	0.435
Extension	Concentric	0.686	0.067	0.405	0.086	0.865	0.053
Extension	Eccentric	0.627	0.075	0.181	0.259	0.238	0.210

Muscle Group		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.892	0.052	0.41	0.124	0.322	0.161
Eccentric	0.25	0.583	0.082	0.184	0.257	0.059	0.482
Concentric	0.125	0.166	0.244	0.008	0.410	0.699	0.066
Eccentric	0.125	0.592	0.081	0.892	0.052	0.099	0.378

Contraction		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.371	0.139	0.012	0.77	0.001	0.981
0.125	Flexion	0.777	0.058	0.596	0.080	0.001	0.970
0.25	Extension	0.254	0.199	0.017	0.731	0.175	0.266
0.125	Extension	0.432	0.118	0.081	0.419	0.625	0.075

Angle			15 - 30 Deg.		30 - 45 Deg.		15 - 45 Deg.		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.782	0.186	0.166	0.328	0.617	0.129	0.424	0.227
0.25	Flexion	Ecentric	0.166	0.485	0.109	0.367	0.005	0.958	0.017	0.837
0.125	Flexion	Concentric	0.033	0.076	0.285	0.189	0.071	0.341	0.060	0.261
0.125	Flexion	Ecentric	0.285	0.304	0.020	0.628	0.004	0.928	0.008	0.851
0.25	Extension	Concentric	0.439	0.051	0.046	0.710	0.002	0.963	0.011	0.886
0.25	Extension	Ecentric	0.439	0.333	0.052	0.368	0.012	0.872	0.022	0.659
0.125	Extension	Concentric	1.000	0.050	0.008	0.646	0.013	0.615	0.010	0.738
0.125	Extension	Ecentric	0.782	0.131	0.166	0.267	0.020	0.603	0.089	0.401

Table B.6: ANOVA tests for settling time in step functions. These analyses evaluated if the different independent variable caused a significant difference in settling time. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25M VF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The angle comparison was between 15 and 30°, 30 and 45°, 15 and 45°, and an overall comparison of the data sets collected at each of the three step sizes. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Settling Time

Step

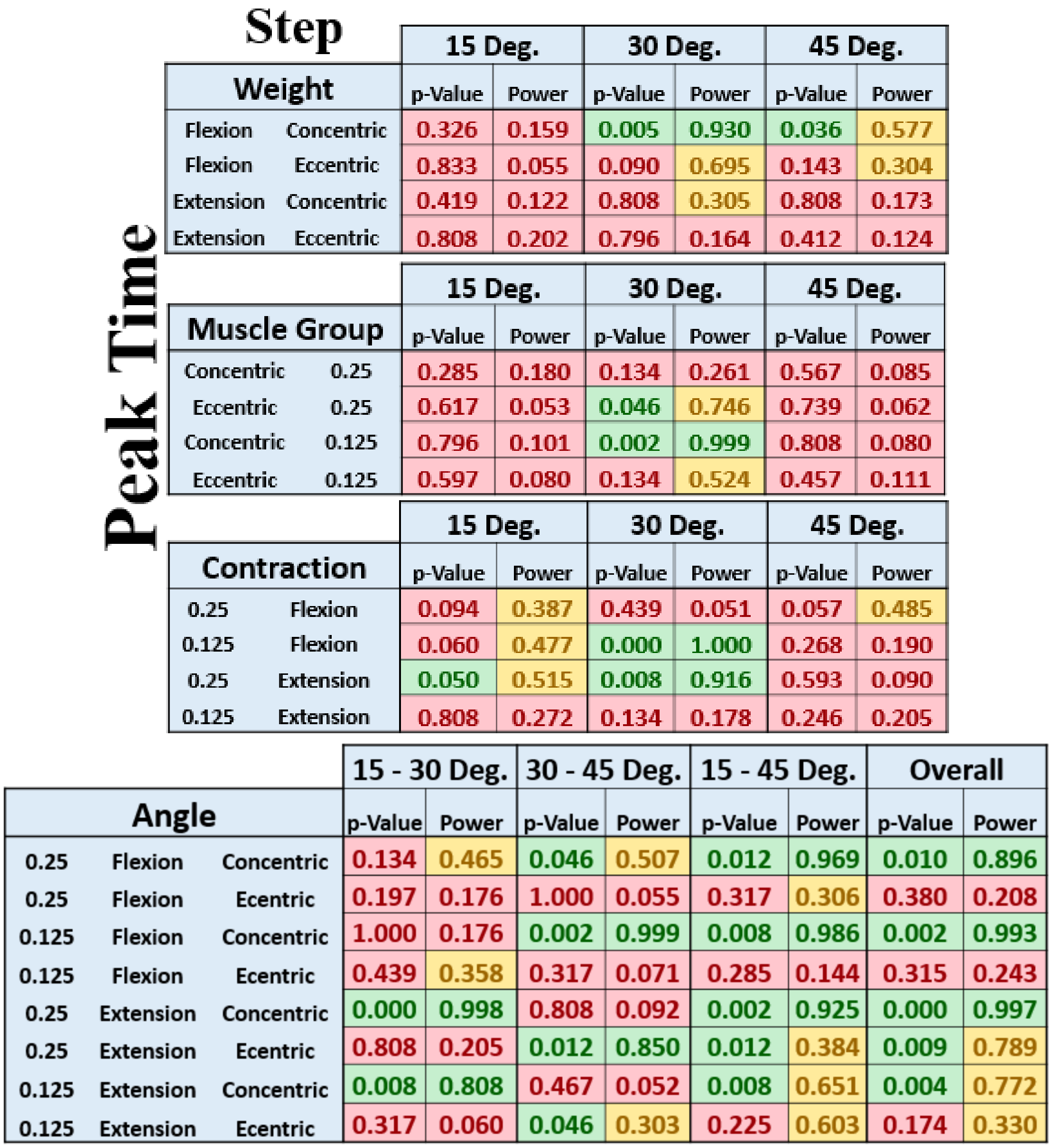
Weight		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.279	0.182	0.303	0.170	0.585	0.161
Flexion	Eccentric	0.693	0.066	0.101	0.374	0.090	0.206
Extension	Concentric	0.437	0.116	0.213	0.230	0.467	0.335
Extension	Eccentric	0.408	0.125	0.303	0.170	0.046	0.366

Muscle Group		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.000	0.999	0.357	0.145	0.002	0.689
Eccentric	0.25	0.121	0.337	0.902	0.052	0.023	0.659
Concentric	0.125	0.274	0.186	0.019	0.693	0.030	0.608
Eccentric	0.125	0.819	0.055	0.013	0.746	0.748	0.061

Contraction		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.000	0.999	0.186	0.255	0.467	0.051
0.125	Flexion	0.272	0.187	0.049	0.519	0.617	0.056
0.25	Extension	0.195	0.245	0.800	0.057	0.906	0.051
0.125	Extension	0.613	0.077	0.046	0.531	0.160	0.718

Angle			15 - 30 Deg.		30 - 45 Deg.		15 - 45 Deg.		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.033	0.756	0.317	0.349	0.001	1.000	0.002	1.000
0.25	Flexion	Ecentric	0.317	0.171	0.317	0.152	0.071	0.642	0.147	0.474
0.125	Flexion	Concentric	0.012	0.827	0.046	0.608	0.002	0.997	0.001	0.971
0.125	Flexion	Ecentric	0.071	0.616	0.461	0.051	0.005	0.741	0.015	0.578
0.25	Extension	Concentric	0.317	0.187	0.046	0.756	0.003	0.992	0.003	0.974
0.25	Extension	Ecentric	0.109	0.520	0.071	0.652	0.000	0.999	0.000	0.998
0.125	Extension	Concentric	0.000	0.998	0.008	0.864	0.000	1.000	0.000	1.000
0.125	Extension	Ecentric	0.005	0.620	0.012	0.828	0.012	0.990	0.001	0.985

Table B.7: ANOVA tests for peak time in step functions. These analyses evaluated if the different independent variable caused a significant difference in peak time. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The angle comparison was between 15 and 30°, 30 and 45°, 15 and 45°, and an overall comparison of the data sets collected at each of the three step sizes. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Step

Peak Time

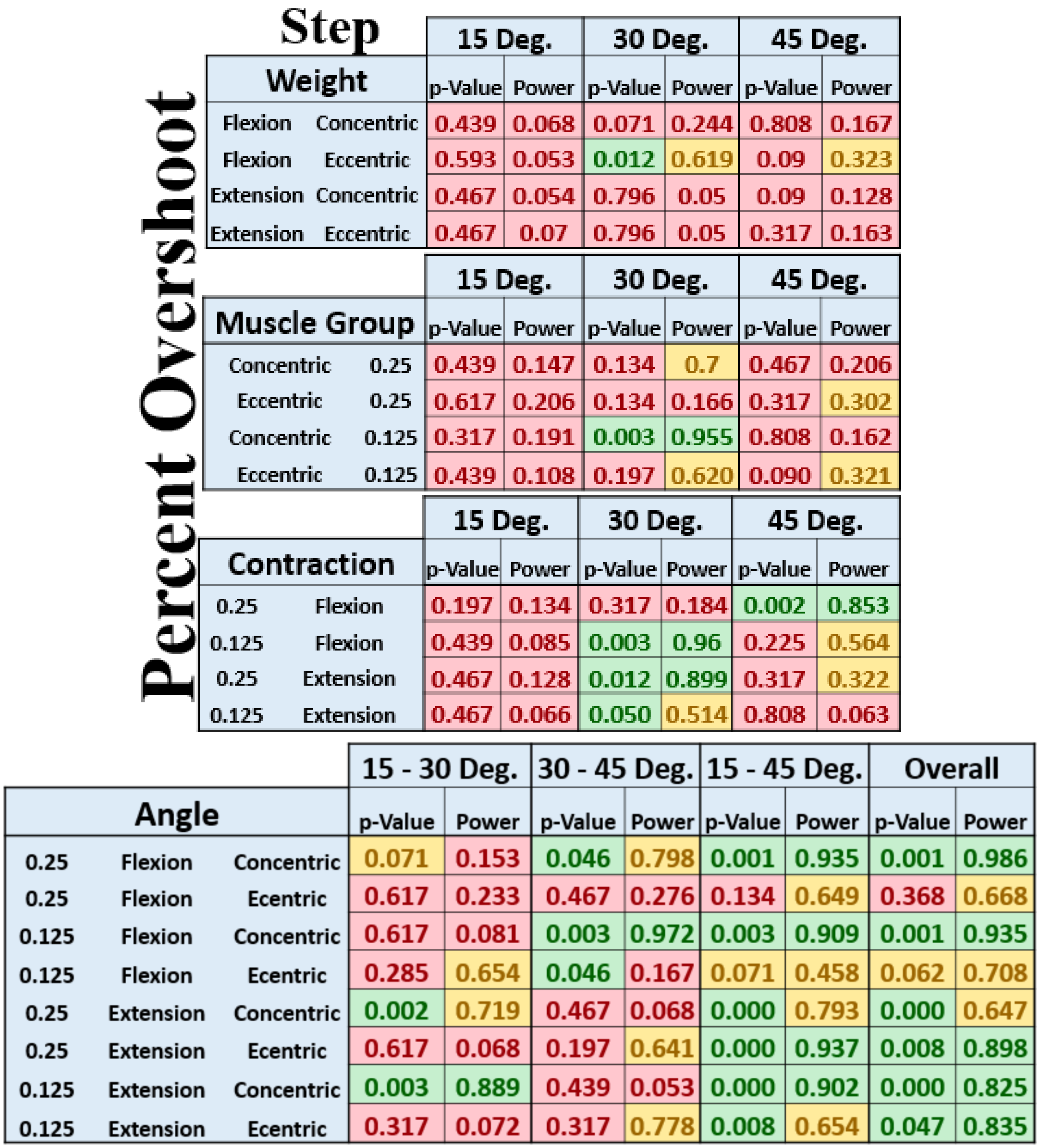
Weight		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.326	0.159	0.005	0.930	0.036	0.577
Flexion	Eccentric	0.833	0.055	0.090	0.695	0.143	0.304
Extension	Concentric	0.419	0.122	0.808	0.305	0.808	0.173
Extension	Eccentric	0.808	0.202	0.796	0.164	0.412	0.124

Muscle Group		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.285	0.180	0.134	0.261	0.567	0.085
Eccentric	0.25	0.617	0.053	0.046	0.746	0.739	0.062
Concentric	0.125	0.796	0.101	0.002	0.999	0.808	0.080
Eccentric	0.125	0.597	0.080	0.134	0.524	0.457	0.111

Contraction		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.094	0.387	0.439	0.051	0.057	0.485
0.125	Flexion	0.060	0.477	0.000	1.000	0.268	0.190
0.25	Extension	0.050	0.515	0.008	0.916	0.593	0.090
0.125	Extension	0.808	0.272	0.134	0.178	0.246	0.205

Angle			15 - 30 Deg.		30 - 45 Deg.		15 - 45 Deg.		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.134	0.465	0.046	0.507	0.012	0.969	0.010	0.896
0.25	Flexion	Ecentric	0.197	0.176	1.000	0.055	0.317	0.306	0.380	0.208
0.125	Flexion	Concentric	1.000	0.176	0.002	0.999	0.008	0.986	0.002	0.993
0.125	Flexion	Ecentric	0.439	0.358	0.317	0.071	0.285	0.144	0.315	0.243
0.25	Extension	Concentric	0.000	0.998	0.808	0.092	0.002	0.925	0.000	0.997
0.25	Extension	Ecentric	0.808	0.205	0.012	0.850	0.012	0.384	0.009	0.789
0.125	Extension	Concentric	0.008	0.808	0.467	0.052	0.008	0.651	0.004	0.772
0.125	Extension	Ecentric	0.317	0.060	0.046	0.303	0.225	0.603	0.174	0.330

Table B.8: ANOVA tests for percent overshoot in step functions. These analyses evaluated if the different independent variable caused a significant difference in percent overshoot. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The angle comparison was between 15 and 30°, 30 and 45°, 15 and 45°, and an overall comparison of the data sets collected at each of the three step sizes. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Step

Percent Overshoot

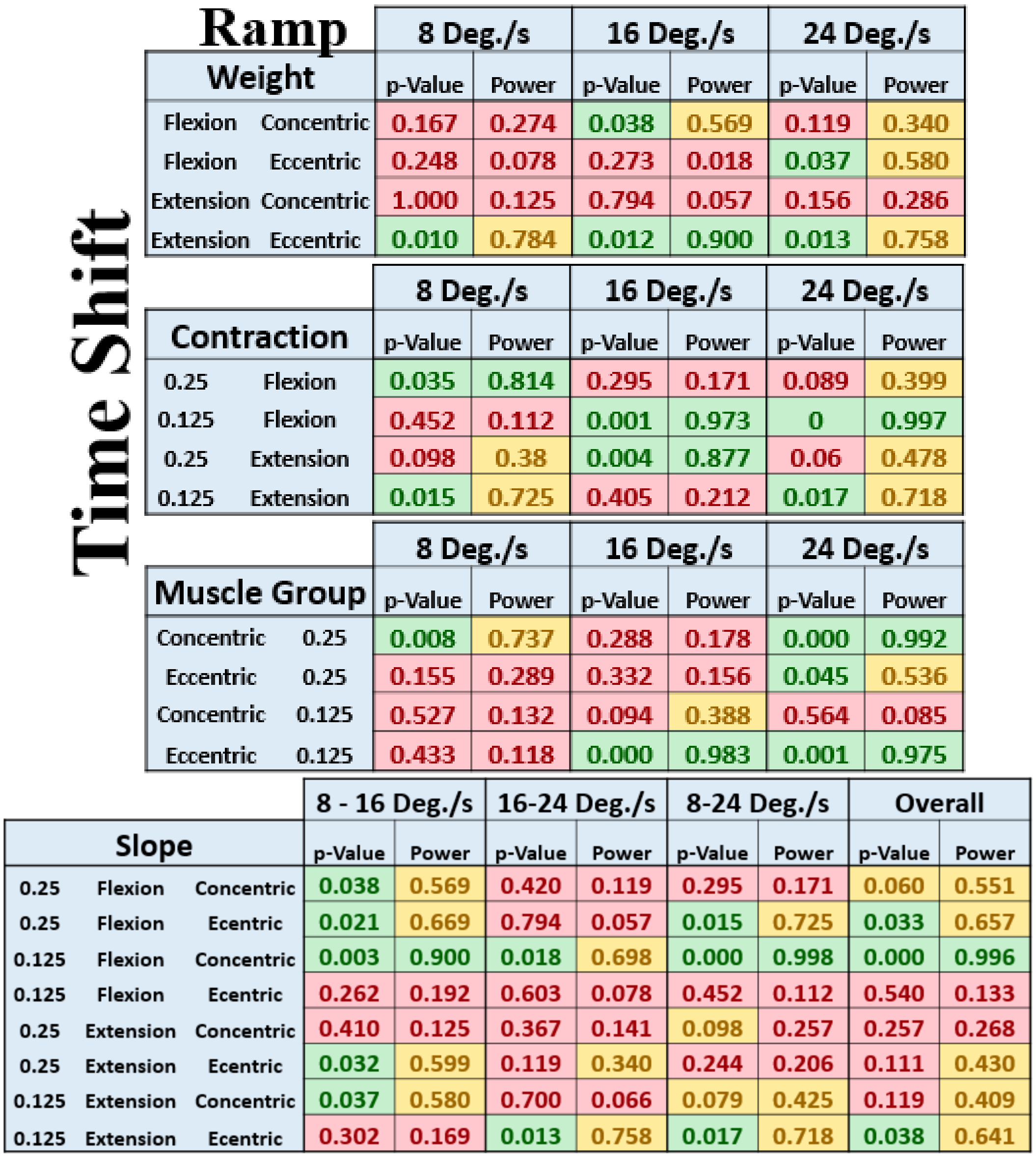
Weight		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.439	0.068	0.071	0.244	0.808	0.167
Flexion	Eccentric	0.593	0.053	0.012	0.619	0.09	0.323
Extension	Concentric	0.467	0.054	0.796	0.05	0.09	0.128
Extension	Eccentric	0.467	0.07	0.796	0.05	0.317	0.163

Muscle Group		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.439	0.147	0.134	0.7	0.467	0.206
Eccentric	0.25	0.617	0.206	0.134	0.166	0.317	0.302
Concentric	0.125	0.317	0.191	0.003	0.955	0.808	0.162
Eccentric	0.125	0.439	0.108	0.197	0.620	0.090	0.321

Contraction		15 Deg.		30 Deg.		45 Deg.	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.197	0.134	0.317	0.184	0.002	0.853
0.125	Flexion	0.439	0.085	0.003	0.96	0.225	0.564
0.25	Extension	0.467	0.128	0.012	0.899	0.317	0.322
0.125	Extension	0.467	0.066	0.050	0.514	0.808	0.063

Angle			15 - 30 Deg.		30 - 45 Deg.		15 - 45 Deg.		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.071	0.153	0.046	0.798	0.001	0.935	0.001	0.986
0.25	Flexion	Ecentric	0.617	0.233	0.467	0.276	0.134	0.649	0.368	0.668
0.125	Flexion	Concentric	0.617	0.081	0.003	0.972	0.003	0.909	0.001	0.935
0.125	Flexion	Ecentric	0.285	0.654	0.046	0.167	0.071	0.458	0.062	0.708
0.25	Extension	Concentric	0.002	0.719	0.467	0.068	0.000	0.793	0.000	0.647
0.25	Extension	Ecentric	0.617	0.068	0.197	0.641	0.000	0.937	0.008	0.898
0.125	Extension	Concentric	0.003	0.889	0.439	0.053	0.000	0.902	0.000	0.825
0.125	Extension	Ecentric	0.317	0.072	0.317	0.778	0.008	0.654	0.047	0.835

Table B.9: ANOVA tests for time shift in ramp function. These analyses evaluated if the different independent variable caused a significant difference in time shift. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The slope comparison was between 8 and 16° per second, 16 and 24° per second, 8 and 24° per second, and an overall comparison of the data sets collected at each of the three slope. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Time Shift

Ramp

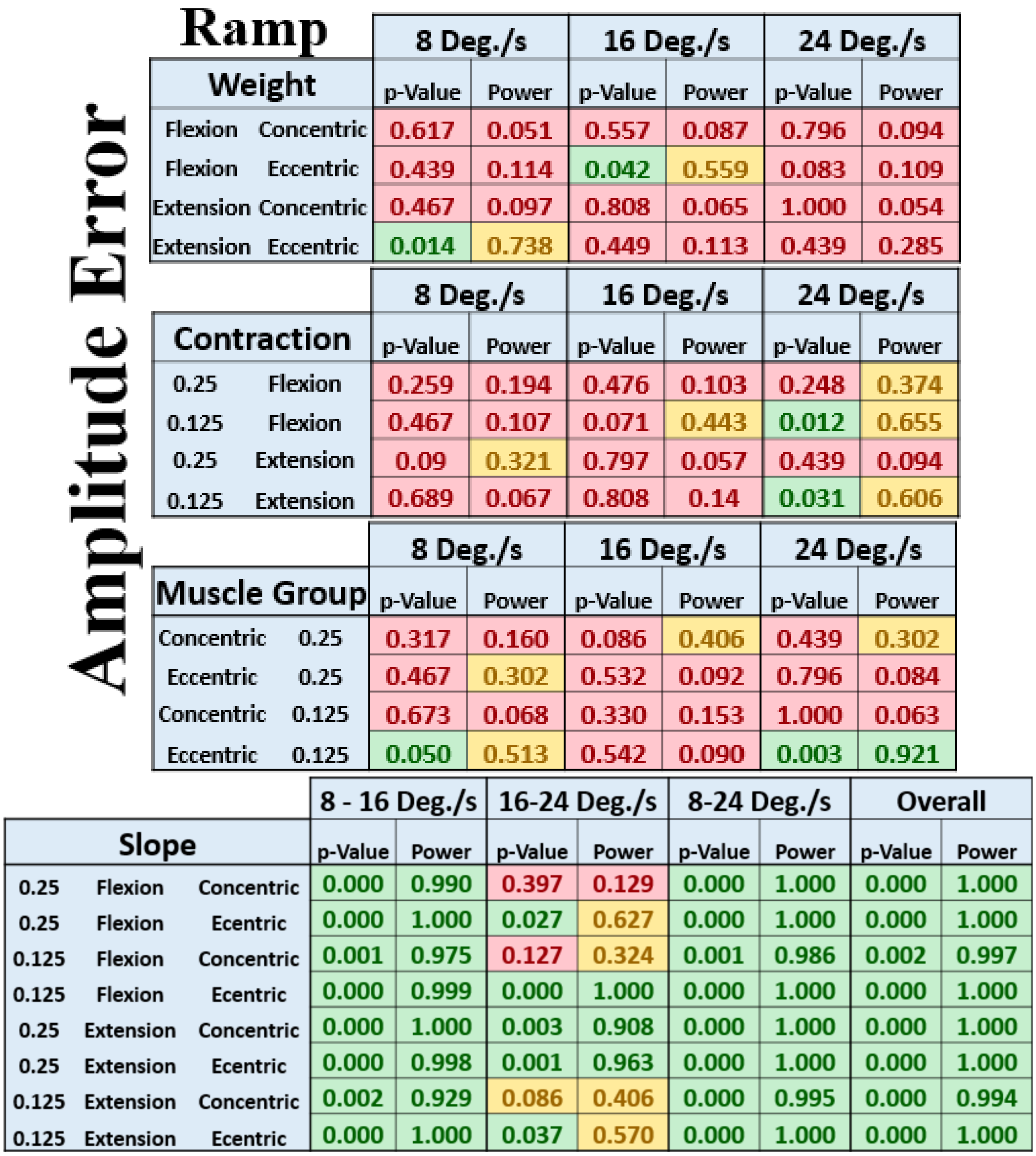
Weight		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.167	0.274	0.038	0.569	0.119	0.340
Flexion	Eccentric	0.248	0.078	0.273	0.018	0.037	0.580
Extension	Concentric	1.000	0.125	0.794	0.057	0.156	0.286
Extension	Eccentric	0.010	0.784	0.012	0.900	0.013	0.758

Contraction		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.035	0.814	0.295	0.171	0.089	0.399
0.125	Flexion	0.452	0.112	0.001	0.973	0	0.997
0.25	Extension	0.098	0.38	0.004	0.877	0.06	0.478
0.125	Extension	0.015	0.725	0.405	0.212	0.017	0.718

Muscle Group		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.008	0.737	0.288	0.178	0.000	0.992
Eccentric	0.25	0.155	0.289	0.332	0.156	0.045	0.536
Concentric	0.125	0.527	0.132	0.094	0.388	0.564	0.085
Eccentric	0.125	0.433	0.118	0.000	0.983	0.001	0.975

Slope			8 - 16 Deg./s		16-24 Deg./s		8-24 Deg./s		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.038	0.569	0.420	0.119	0.295	0.171	0.060	0.551
0.25	Flexion	Ecentric	0.021	0.669	0.794	0.057	0.015	0.725	0.033	0.657
0.125	Flexion	Concentric	0.003	0.900	0.018	0.698	0.000	0.998	0.000	0.996
0.125	Flexion	Ecentric	0.262	0.192	0.603	0.078	0.452	0.112	0.540	0.133
0.25	Extension	Concentric	0.410	0.125	0.367	0.141	0.098	0.257	0.257	0.268
0.25	Extension	Ecentric	0.032	0.599	0.119	0.340	0.244	0.206	0.111	0.430
0.125	Extension	Concentric	0.037	0.580	0.700	0.066	0.079	0.425	0.119	0.409
0.125	Extension	Ecentric	0.302	0.169	0.013	0.758	0.017	0.718	0.038	0.641

Table B.10: ANOVA tests for amplitude error in ramp function. These analyses evaluated if the different independent variable caused a significant difference in amplitude error. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The slope comparison was between 8 and 16° per second, 16 and 24° per second, 8 and 24° per second, and an overall comparison of the data sets collected at each of the three slope. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Amplitude Error

Ramp

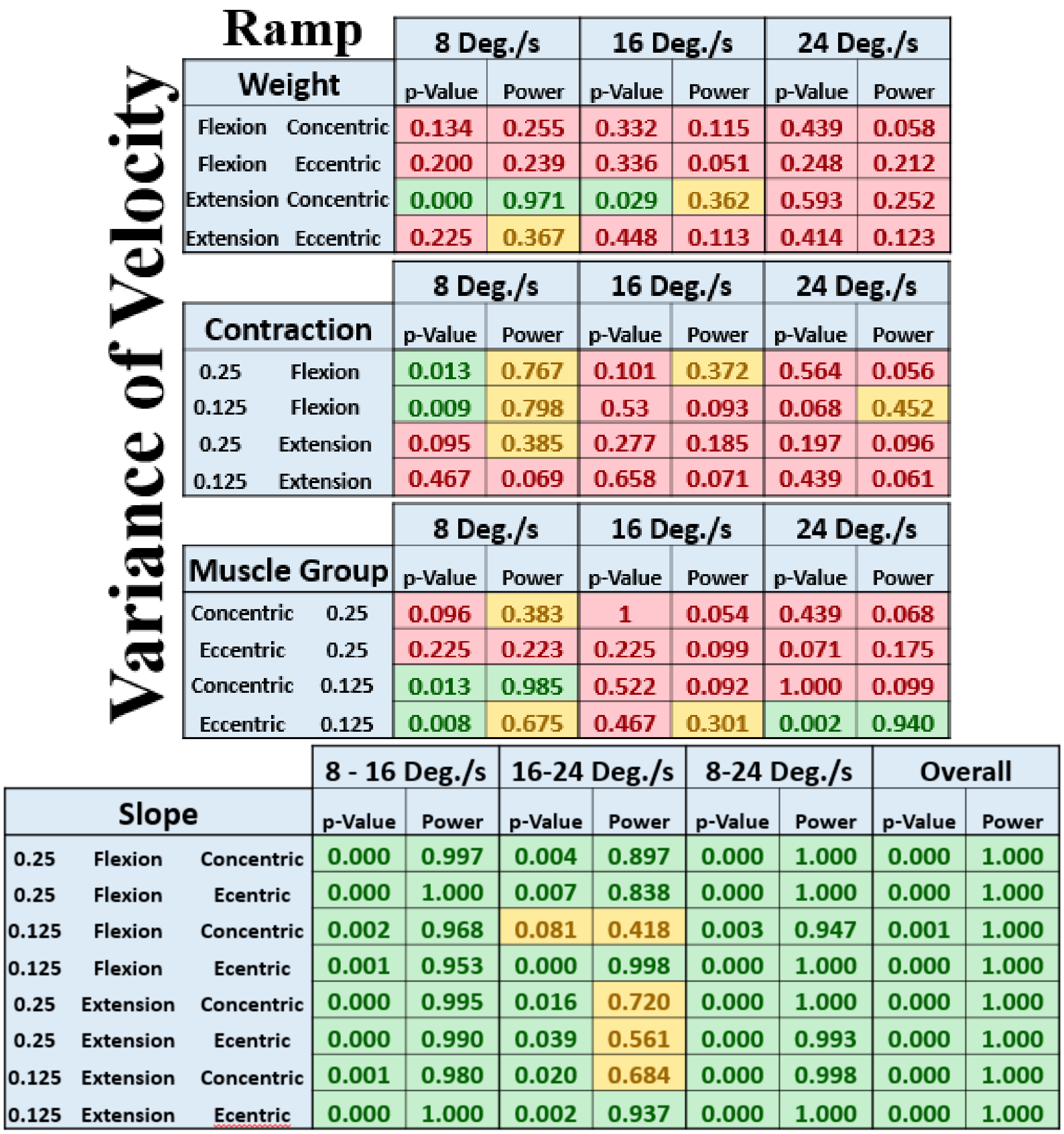
Weight		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.617	0.051	0.557	0.087	0.796	0.094
Flexion	Eccentric	0.439	0.114	0.042	0.559	0.083	0.109
Extension	Concentric	0.467	0.097	0.808	0.065	1.000	0.054
Extension	Eccentric	0.014	0.738	0.449	0.113	0.439	0.285

Contraction		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.259	0.194	0.476	0.103	0.248	0.374
0.125	Flexion	0.467	0.107	0.071	0.443	0.012	0.655
0.25	Extension	0.09	0.321	0.797	0.057	0.439	0.094
0.125	Extension	0.689	0.067	0.808	0.14	0.031	0.606

Muscle Group		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.317	0.160	0.086	0.406	0.439	0.302
Eccentric	0.25	0.467	0.302	0.532	0.092	0.796	0.084
Concentric	0.125	0.673	0.068	0.330	0.153	1.000	0.063
Eccentric	0.125	0.050	0.513	0.542	0.090	0.003	0.921

Slope			8 - 16 Deg./s		16-24 Deg./s		8-24 Deg./s		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.000	0.990	0.397	0.129	0.000	1.000	0.000	1.000
0.25	Flexion	Ecentric	0.000	1.000	0.027	0.627	0.000	1.000	0.000	1.000
0.125	Flexion	Concentric	0.001	0.975	0.127	0.324	0.001	0.986	0.002	0.997
0.125	Flexion	Ecentric	0.000	0.999	0.000	1.000	0.000	1.000	0.000	1.000
0.25	Extension	Concentric	0.000	1.000	0.003	0.908	0.000	1.000	0.000	1.000
0.25	Extension	Ecentric	0.000	0.998	0.001	0.963	0.000	1.000	0.000	1.000
0.125	Extension	Concentric	0.002	0.929	0.086	0.406	0.000	0.995	0.000	0.994
0.125	Extension	Ecentric	0.000	1.000	0.037	0.570	0.000	1.000	0.000	1.000

Table B.11: ANOVA tests for variance of velocity in ramp function. These analyses evaluated if the different independent variable caused a significant difference in variance of velocity. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The slope comparison was between 8 and 16° per second, 16 and 24° per second, 8 and 24° per second, and an overall comparison of the data sets collected at each of the three slope. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Variance of Velocity

Ramp

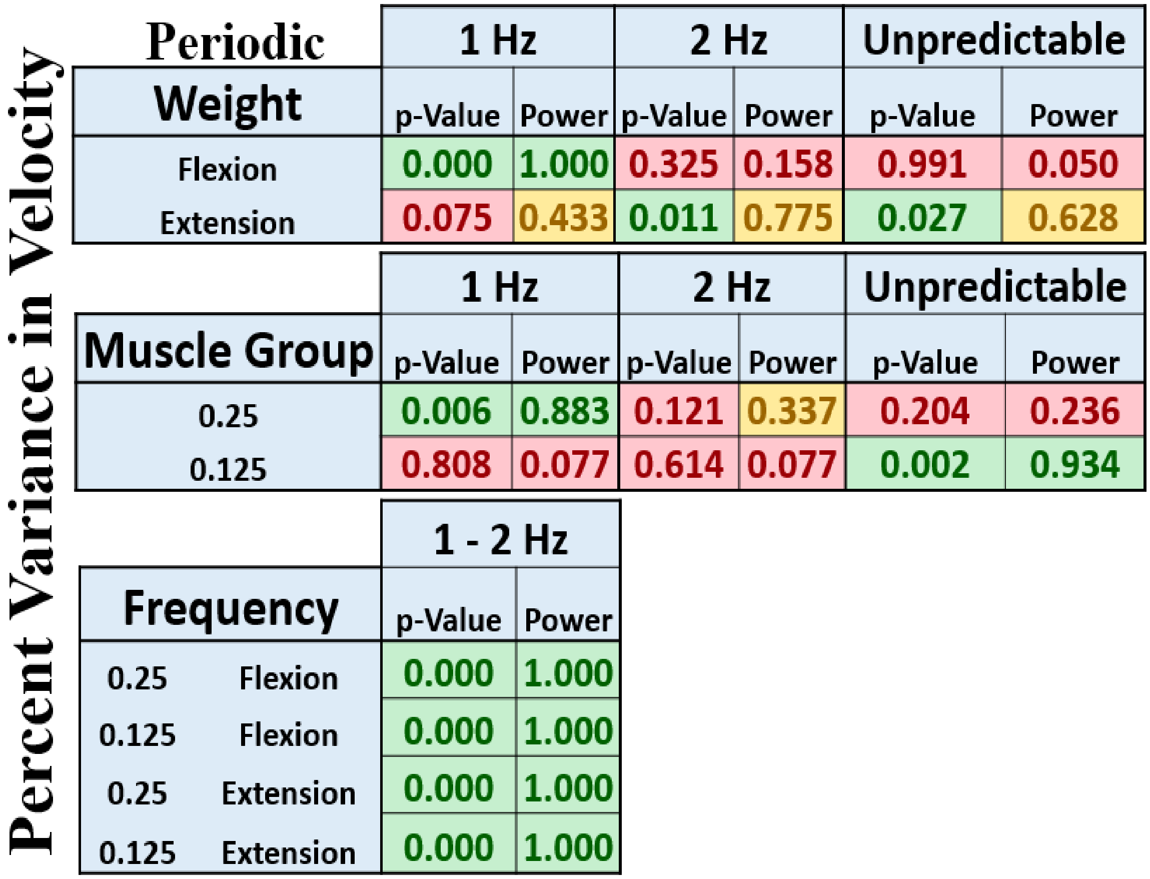
Weight		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Flexion	Concentric	0.134	0.255	0.332	0.115	0.439	0.058
Flexion	Eccentric	0.200	0.239	0.336	0.051	0.248	0.212
Extension	Concentric	0.000	0.971	0.029	0.362	0.593	0.252
Extension	Eccentric	0.225	0.367	0.448	0.113	0.414	0.123

Contraction		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	0.013	0.767	0.101	0.372	0.564	0.056
0.125	Flexion	0.009	0.798	0.53	0.093	0.068	0.452
0.25	Extension	0.095	0.385	0.277	0.185	0.197	0.096
0.125	Extension	0.467	0.069	0.658	0.071	0.439	0.061

Muscle Group		8 Deg./s		16 Deg./s		24 Deg./s	
		p-Value	Power	p-Value	Power	p-Value	Power
Concentric	0.25	0.096	0.383	1	0.054	0.439	0.068
Eccentric	0.25	0.225	0.223	0.225	0.099	0.071	0.175
Concentric	0.125	0.013	0.985	0.522	0.092	1.000	0.099
Eccentric	0.125	0.008	0.675	0.467	0.301	0.002	0.940

Slope			8 - 16 Deg./s		16-24 Deg./s		8-24 Deg./s		Overall	
			p-Value	Power	p-Value	Power	p-Value	Power	p-Value	Power
0.25	Flexion	Concentric	0.000	0.997	0.004	0.897	0.000	1.000	0.000	1.000
0.25	Flexion	Ecentric	0.000	1.000	0.007	0.838	0.000	1.000	0.000	1.000
0.125	Flexion	Concentric	0.002	0.968	0.081	0.418	0.003	0.947	0.001	1.000
0.125	Flexion	Ecentric	0.001	0.953	0.000	0.998	0.000	1.000	0.000	1.000
0.25	Extension	Concentric	0.000	0.995	0.016	0.720	0.000	1.000	0.000	1.000
0.25	Extension	Ecentric	0.000	0.990	0.039	0.561	0.000	0.993	0.000	1.000
0.125	Extension	Concentric	0.001	0.980	0.020	0.684	0.000	0.998	0.000	1.000
0.125	Extension	Ecentric	0.000	1.000	0.002	0.937	0.000	1.000	0.000	1.000

Table B.12: ANOVA tests for percent variance of velocity in periodic function. These analyses evaluated if the different independent variable caused a significant difference in percent variance of velocity. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The frequency comparison was between 1 and 2 Hertz. The unpredictable function was left out of this analysis. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



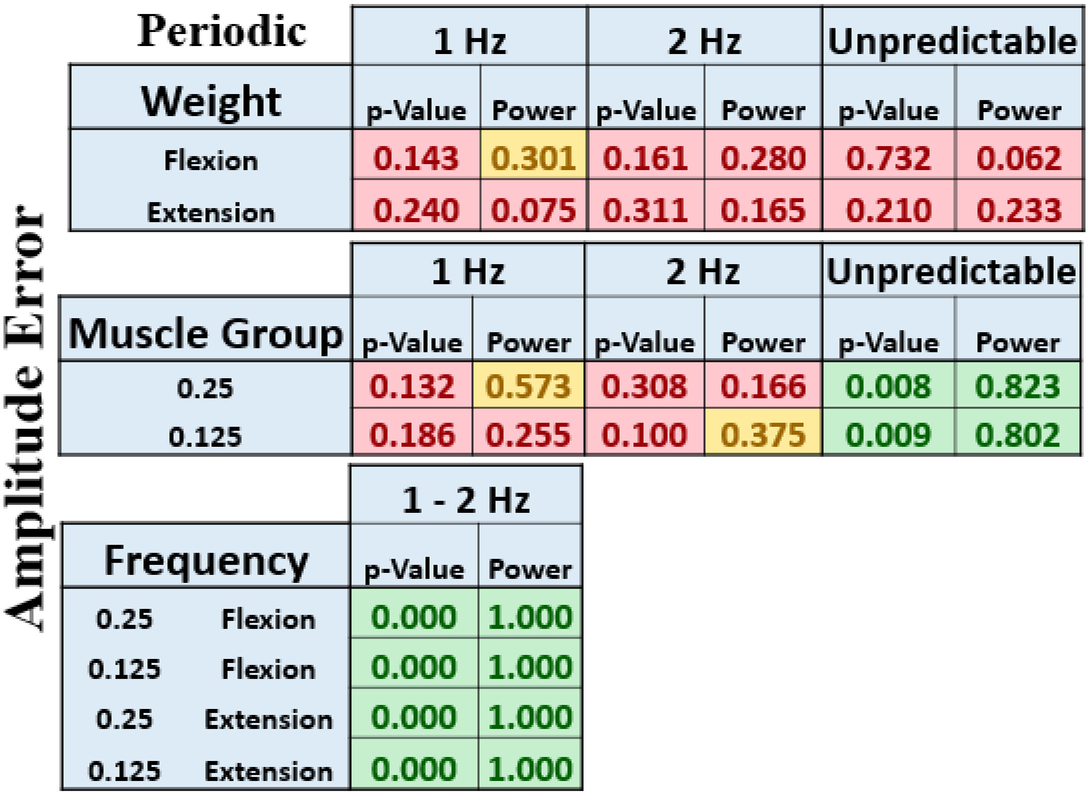
Percent Variance in Velocity

Periodic	1 Hz		2 Hz		Unpredictable	
Weight	p-Value	Power	p-Value	Power	p-Value	Power
Flexion	0.000	1.000	0.325	0.158	0.991	0.050
Extension	0.075	0.433	0.011	0.775	0.027	0.628

	1 Hz		2 Hz		Unpredictable	
Muscle Group	p-Value	Power	p-Value	Power	p-Value	Power
0.25	0.006	0.883	0.121	0.337	0.204	0.236
0.125	0.808	0.077	0.614	0.077	0.002	0.934

		1 - 2 Hz	
Frequency		p-Value	Power
0.25	Flexion	0.000	1.000
0.125	Flexion	0.000	1.000
0.25	Extension	0.000	1.000
0.125	Extension	0.000	1.000

Table B.13: ANOVA tests for amplitude error in periodic function. These analyses evaluated if the different independent variable caused a significant difference in amplitude error. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The frequency comparison was between 1 and 2 Hertz. The unpredictable function was left out of this analysis. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



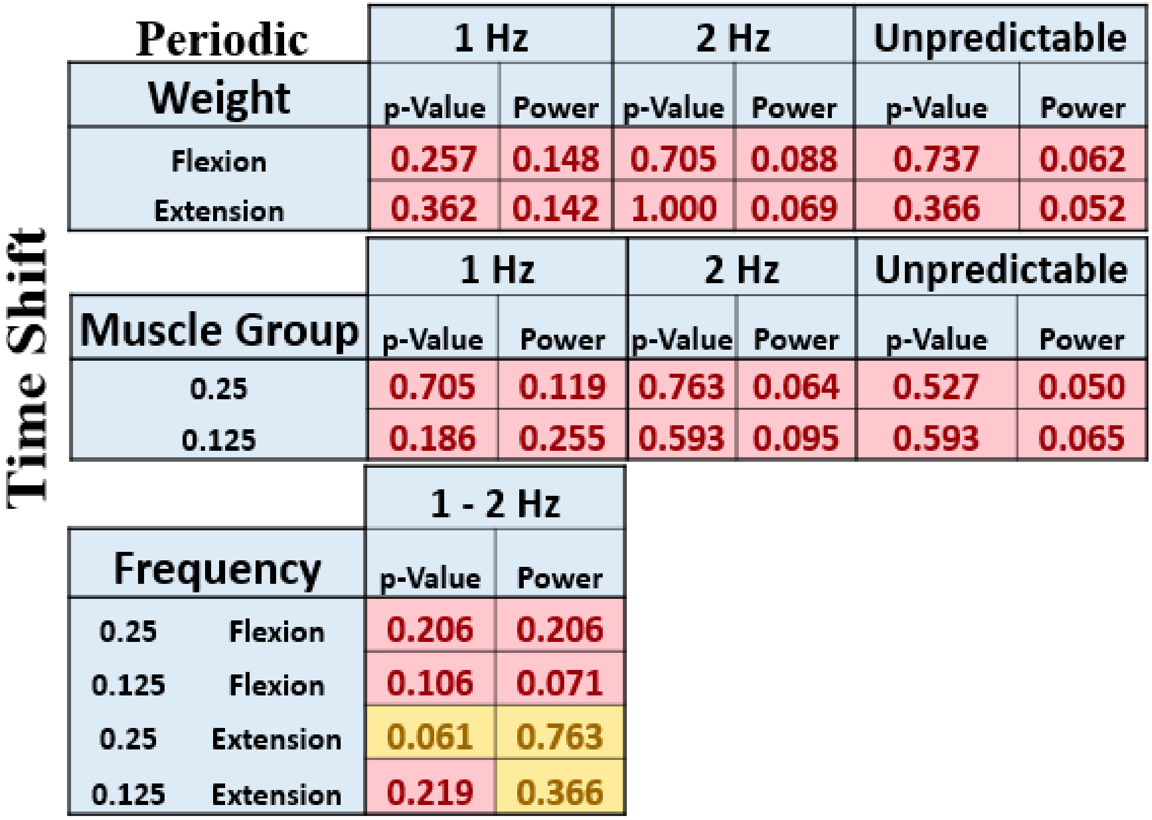
Amplitude Error

Periodic	1 Hz		2 Hz		Unpredictable	
Weight	p-Value	Power	p-Value	Power	p-Value	Power
Flexion	0.143	0.301	0.161	0.280	0.732	0.062
Extension	0.240	0.075	0.311	0.165	0.210	0.233

	1 Hz		2 Hz		Unpredictable	
Muscle Group	p-Value	Power	p-Value	Power	p-Value	Power
0.25	0.132	0.573	0.308	0.166	0.008	0.823
0.125	0.186	0.255	0.100	0.375	0.009	0.802

		1 - 2 Hz	
Frequency		p-Value	Power
0.25	Flexion	0.000	1.000
0.125	Flexion	0.000	1.000
0.25	Extension	0.000	1.000
0.125	Extension	0.000	1.000

Table B.14: ANOVA tests for time shift in periodic function. These analyses evaluated if the different independent variable caused a significant difference in time shift. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The frequency comparison was between 1 and 2 Hertz. The unpredictable function was left out of this analysis. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



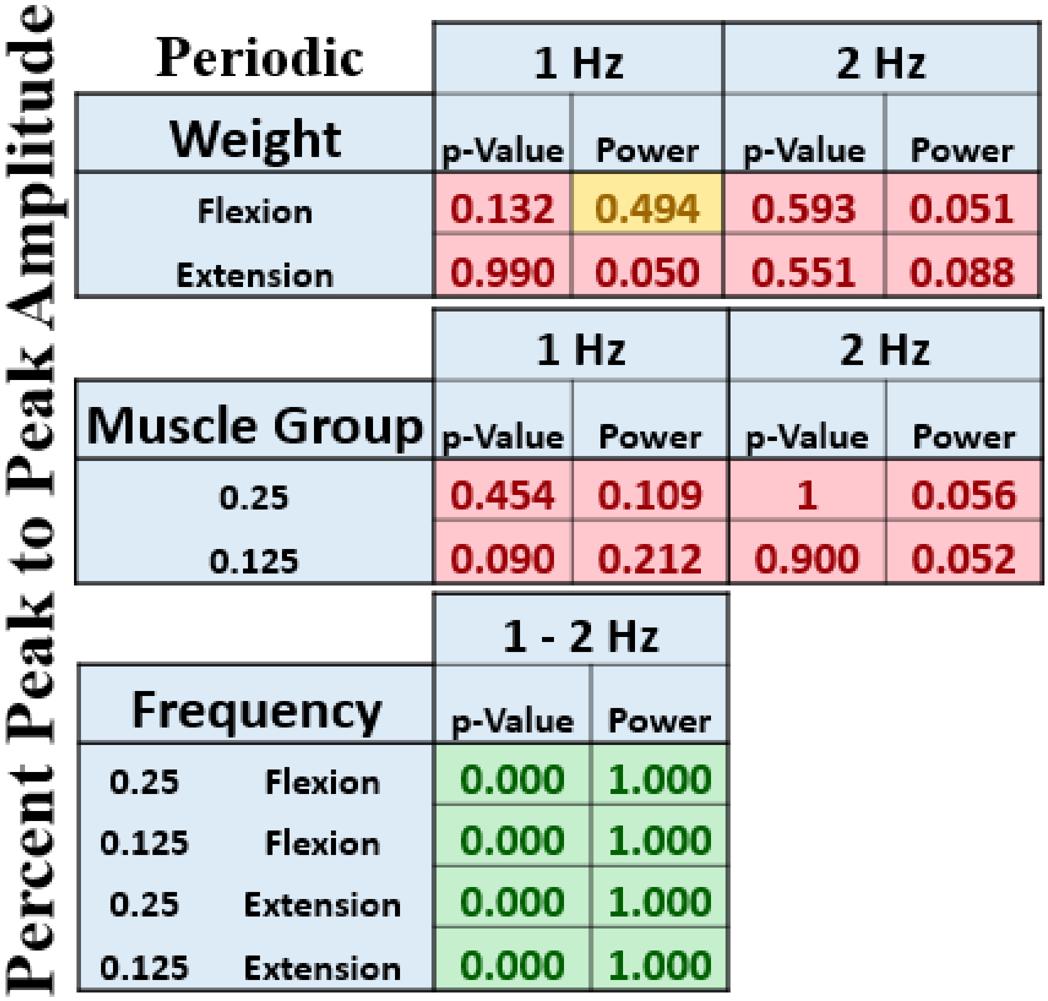
Time Shift

Periodic	1 Hz		2 Hz		Unpredictable	
Weight	p-Value	Power	p-Value	Power	p-Value	Power
Flexion	0.257	0.148	0.705	0.088	0.737	0.062
Extension	0.362	0.142	1.000	0.069	0.366	0.052

	1 Hz		2 Hz		Unpredictable	
Muscle Group	p-Value	Power	p-Value	Power	p-Value	Power
0.25	0.705	0.119	0.763	0.064	0.527	0.050
0.125	0.186	0.255	0.593	0.095	0.593	0.065

		1 - 2 Hz	
Frequency		p-Value	Power
0.25	Flexion	0.206	0.206
0.125	Flexion	0.106	0.071
0.25	Extension	0.061	0.763
0.125	Extension	0.219	0.366

Table B.15: ANOVA tests for percent peak to peak amplitude in periodic function. These analyses evaluated if the different independent variable caused a significant difference in percent peak to peak response amplitude. Each variable was compared while holding all of the other independent variables constant. The independent variables being compared are shown in the top left corner of each table. The weight comparison was between 0.125 MVF and 0.25 MVF. The muscle group comparison was between flexion and extension. The contraction comparison was between concentric and eccentric. The frequency comparison was between 1 and 2 Hertz. The unpredictable function was left out of this analysis. A red value means that it was not significant, a green value means that it was, and a yellow value means that it was close to significance.



Percent Peak to Peak Amplitude

Periodic	1 Hz		2 Hz	
Weight	p-Value	Power	p-Value	Power
Flexion	0.132	0.494	0.593	0.051
Extension	0.990	0.050	0.551	0.088

	1 Hz		2 Hz	
Muscle Group	p-Value	Power	p-Value	Power
0.25	0.454	0.109	1	0.056
0.125	0.090	0.212	0.900	0.052

		1 - 2 Hz	
Frequency		p-Value	Power
0.25	Flexion	0.000	1.000
0.125	Flexion	0.000	1.000
0.25	Extension	0.000	1.000
0.125	Extension	0.000	1.000

APPENDIX C

MAIN EXPERIMENT CODE

```
%This code generate a graphic user interface in Matlab and uses event
%listeners to collect position and torque data in real time. This data
is
%saved and displayed for the user. This is an expansion of the code
%developed by using the GUIDE tool in Matlab.

function varargout = MastersGUI(varargin)
% MASTERSGUI MATLAB code for MastersGUI.fig
%      MASTERSGUI, by itself, creates a new MASTERSGUI or raises the
existing
%      singleton*.
%
%      H = MASTERSGUI returns the handle to a new MASTERSGUI or the
handle to
%      the existing singleton*.
%
%      MASTERSGUI('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in MASTERSGUI.M with the given input
arguments.
%
%      MASTERSGUI('Property','Value',...) creates a new MASTERSGUI or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before MastersGUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to MastersGUI_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MastersGUI

% Last Modified by GUIDE v2.5 31-May-2014 14:09:03
addpath(genpath('G:\Masters\Code\ExperimentTry2\'));
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MastersGUI_OpeningFcn, ...
                   'gui_OutputFcn',  @MastersGUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before MastersGUI is made visible.

%This function sets up all of the initial variable and builds the
signals
%the participants will track.
function MastersGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MastersGUI (see VARARGIN)
global countCase
countCase=0;

% Choose default command line output for MastersGUI
handles.output = hObject;

handles.modelName ='MastersGUI_model';
handles.modelName2='Practice_model';
handles.modelName3='Weight_model';
handles.modelName4='ROM_model';
handles.Case='MastersGUI_model/Case';
handles.TorqueIn='MastersGUI_model/TorqueIn';
handles.PracticeTorque='Practice_model/PracticeTorque';
handles.RandomNumber=[];

%This is to update the structure name being output

for i=1:36
handles.CaseNum= ['Case',num2str(i)];
handles.(handles.CaseNum).Signal=[];
handles.(handles.CaseNum).SignalTime=[];
handles.(handles.CaseNum).Speed=[];
handles.(handles.CaseNum).Human=[];
handles.(handles.CaseNum).Torque =[];
handles.(handles.CaseNum).Time=[];
handles.(handles.CaseNum).CaseValue=0;
end
```

```
%________________________SIGNAL
GENERATION_________________________________
% All of the STEP Signals
for j=1:2601
    if j<601
        handles.(['Signal',num2str(1)])(j)=0;
        handles.(['Signal',num2str(2)])(j)=0;
        handles.(['Signal',num2str(3)])(j)=0;
        handles.timeArray(j)=-0.005+j*0.005;
    elseif j<1801
        handles.(['Signal',num2str(1)])(j)=0.667*45;
        handles.(['Signal',num2str(2)])(j)=0.333*45;
        handles.(['Signal',num2str(3)])(j)=-1*45;
        handles.timeArray(j)=j*0.005;
    else
        handles.(['Signal',num2str(1)])(j)=0;
        handles.(['Signal',num2str(2)])(j)=0;
        handles.(['Signal',num2str(3)])(j)=0;
        handles.timeArray(j)=j*0.005; %-(601*0.005);
    end
end
% All of the Ramp Signals
for k=1:2601
    if k<601
        handles.(['Signal',num2str(4)])(k)=0;
        handles.(['Signal',num2str(5)])(k)=0;
        handles.(['Signal',num2str(6)])(k)=0;
    elseif k<951
        handles.(['Signal',num2str(4)])(k)=(handles.timeArray(k-
600))*16;
        handles.(['Signal',num2str(5)])(k)=handles.timeArray(k-600)*8;
        handles.(['Signal',num2str(6)])(k)=handles.timeArray(k-600)*(-
24);

    elseif k<1201
        handles.(['Signal',num2str(4)])(k)=28;
        handles.(['Signal',num2str(5)])(k)=14;
        handles.(['Signal',num2str(6)])(k)=-42;
    elseif k<1501
        handles.(['Signal',num2str(4)])(k)=28;
        handles.(['Signal',num2str(5)])(k)=14;
        handles.(['Signal',num2str(6)])(k)=-42;
    elseif k<1851
        handles.(['Signal',num2str(4)])(k)=(handles.timeArray(k-
1500))*(-16)+28;
        handles.(['Signal',num2str(5)])(k)=(handles.timeArray(k-
1500))*(-8)+14;
        handles.(['Signal',num2str(6)])(k)=handles.timeArray(k-
1500)*(24)-42;
    elseif k<2101
        handles.(['Signal',num2str(4)])(k)=0;
        handles.(['Signal',num2str(5)])(k)=0;
        handles.(['Signal',num2str(6)])(k)=0;
    else
        handles.(['Signal',num2str(4)])(k)=0;
        handles.(['Signal',num2str(5)])(k)=0;
        handles.(['Signal',num2str(6)])(k)=0;
    end
end
% Periodic Signals
for l=1:2601
    if l<601;
        handles.(['Signal',num2str(7)])(l)=0;
        handles.(['Signal',num2str(8)])(l)=0;
        handles.(['Signal',num2str(9)])(l)=0;
    else
        handles.(['Signal',num2str(7)])(l)=0.4*sin(handles.timeArray(l-
600))*45;

handles.(['Signal',num2str(8)])(l)=0.6*sin(2*handles.timeArray(l-
600))*45;

handles.(['Signal',num2str(9)])(l)=0.4*sin(1.1*handles.timeArray(l-
600))*45+...
        0.1*sin(2*handles.timeArray(l-
600))*45+0.25*sin(3*handles.timeArray(l-600))*45;
    end
end
%_______________________________________________
handles.DisplayTime=[];
handles.Clock=0;
handles.DesiredWeight=0;
handles.DesiredWeight_Flexion=0;
handles.DesiredWeight_Extension=0;
handles.pressbutton = false;
handles.set = true;
handles.datetime = datestr(now,30);

%Which buttons start enabled and disabled
set(handles.Load_Model,'Enable','off');
set(handles.Start_Model,'Enable','off');
set(handles.Stop_Model,'Enable','off');
set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Start_Weight_Test,'Enable','off');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Set_Weight,'Enable','on');
set(handles.StartROM,'Enable','on');
set(handles.StopROM,'Enable','on');

rover.yPosition=0;
rover.start_pos=[0 rover.yPosition];
rover.position=rover.start_pos;
rover.Random=handles.RandomNumber;
assignin('base','rover',rover);


set(handles.figure1,'Tag',mfilename);
set(handles.figure1,'CloseRequestFcn',@localCloseRequestFcn);

%This puts up the initial graph on the participants screen.
```

```
rover.hrover1=line('Parent',handles.Graph,'Xdata',handles.timeArray,'Yd
ata',handles.(['Signal',num2str(1)]),'LineWidth',3,'Color',[0.7 0.7
0.7]);
%rover.hrover3=line('Parent',handles.Graph,'Xdata',0,'Ydata',rover.yPos
ition,...
 %   'Marker','o','MarkerSize',12,'MarkerEdgeColor',[0 0
1],'MarkerFaceColor',[0 0 1]);
rover.hrover2=line('Parent',handles.Graph,'Xdata',0,'Ydata',rover.yPosi
tion,...
    'Marker','o','MarkerSize',10,'MarkerEdgeColor',[1 0
0],'MarkerFaceColor',[1 0 0]);
set(handles.Graph,'Xlim',[-0.5 4.5]);
handles.rover = rover;

guidata(hObject, handles);
movegui(hObject,'center')
% Update handles structure
set(hObject,'Visible','on')



% --- Outputs from this function are returned to the command line.
function varargout = MastersGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Load_Model.
function Load_Model_Callback(hObject, eventdata, handles)
% hObject    handle to Load_Model (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
wStr = sprintf('%s\n%s',...
    'Please wait while the model loads.');
hw = waitbar(0.5,wStr);
global countCase;
countCase=1;
%Create a new folder for this participant

ParticipantNum=get(handles.Pnumber,'String');

mkdir(ParticipantNum);
handles.foldername =
strcat('G:\Masters\Code\ExperimentTry2\',ParticipantNum,'\');
handles.RandomNumber=[1 2 3 4 5 6 7 8 9];
handles.(handles.CaseNum).CaseValue=handles.RandomNumber(countCase);
%Opens (invisibly) the simulink model
% if ~modelIsLoaded(handles.modelName)
load_system(handles.modelName);
% end

handles.viewing = struct(...
    'blockName','',...
    'blockHandle',[],...
    'blockEvent','',...
    'blockFcn',[]);


%Either we want to make this a variable or just always use global
position
handles.viewing.blockName = sprintf('%s/Graph',handles.modelName);
handles.viewing.blockHandle =
get_param(handles.viewing.blockName,'Handle');

%sets up the event listener
handles.viewing.blockEvent = 'PostOutputs';
handles.viewing.blockFcn = @localEventListener;

handles.originalStopTime = get_param(handles.modelName,'Stoptime');
handles.originalMode =  get_param(handles.modelName,'SimulationMode');
handles.originalStartFcn = get_param(handles.modelName,'StartFcn');
handles.modelAlreadyBuilt = false;

% set the simulation mode to external
set_param(handles.modelName,'SimulationMode','external');

TorqueVal=str2double(handles.DesiredWeight_Flexion);
set_param(handles.TorqueIn,'Value',num2str(TorqueVal));
handles.CaseNum=['Case',num2str(1)];
%Build the model as an executable
%rtwbuild(handles.modelName);
%set_param(handles.modelName,'SimulationMode',handles.originalMode);
% destroy the waitbar
delete(hw);
% Toggle the state of the buttons
% Turn off the build_button button
set(handles.Load_Model,'Enable','off');
% Turn on the start_button button
set(handles.Start_Model,'Enable','on');
set(handles.Stop_Model,'Enable','on');
set(handles.Practice,'Enable','off');



% Set the already built flag so we don't build_button again
% ad.modelAlreadyBuilt = true;
% Flush the graphics buffer
 %save(strcat(handles.foldername,(handles.CaseNum),'.mat'),'-
struct','handles',(handles.CaseNum));

%Save the random number to the file
%save(handles.foldername,'-struct','rover',Random)
drawnow
% store the changed app data
guidata(hObject,handles);

% --- Executes on button press in Start_Model.
function Start_Model_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to Start_Model (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global countCase
set(handles.Start_Model,'Enable','off');
set(handles.Stop_Model,'Enable','on');
set(handles.Next,'Enable','on');
set(handles.Practice,'Enable','off')
rover=handles.rover;

set_param(handles.Case,'Value',num2str(handles.RandomNumber(countCase))
);
handles.(handles.CaseNum).Signal=45.*handles.(['Signal',num2str(countCa
se)]);
handles.(handles.CaseNum).SignalTime=handles.timeArray;
set(rover.hrover1,'Xdata',handles.timeArray,'Ydata',handles.(['Signal',
num2str(countCase)]));
set_param(handles.modelName,'StopTime','16');
%setting up  the model

% Connect to the code

set_param(handles.modelName,'SimulationCommand','connect');
% start_button the model
set_param(handles.modelName,'SimulationCommand','start');
 guidata(hObject,handles);
% Attach the listener
drawnow
localAddEventListener
%set_param(handles.modelName,'StopFcn',set_param(handles.TorqueIn,'Valu
e','0'));

% --- Executes on button press in Next.
function Next_Callback(hObject, eventdata, handles)
% hObject    handle to Next (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

assignin('base',(handles.CaseNum),handles.(handles.CaseNum));
%csvwrite(strcat(handles.foldername,(handles.CaseNum),'.csv'),(handles.
CaseNum));
save(strcat(handles.foldername,(handles.CaseNum),'.mat'),'-
struct','handles',(handles.CaseNum));

global countCase;
countCase= countCase+1;
Number=num2str(countCase);
rover=handles.rover;
handles.CaseNum= ['Case',Number];
guidata(hObject,handles);

if countCase<=9
   handles.(handles.CaseNum).CaseValue=handles.RandomNumber(countCase);
set(rover.hrover1,'Xdata',handles.timeArray,'Ydata',handles.(['Signal',
num2str(countCase)]));
```

```
   set_param(handles.Case,
'Value',num2str(handles.RandomNumber(countCase)));

handles.(handles.CaseNum).Signal=45.*handles.(['Signal',num2str(countCa
se)]);
   handles.(handles.CaseNum).SignalTime=handles.timeArray;
elseif countCase<=18

set(rover.hrover1,'Xdata',handles.timeArray,'Ydata',handles.(['Signal',
num2str(countCase-9)]));
   set_param(handles.Case,
'Value',num2str(handles.RandomNumber(countCase-9)));
   handles.(handles.CaseNum).CaseValue=handles.RandomNumber(countCase-
9);

handles.(handles.CaseNum).Signal=45.*handles.(['Signal',num2str(countCa
se-9)]);
   handles.(handles.CaseNum).SignalTime=handles.timeArray;
   handles.DesiredWeight=handles.DesiredWeight_Flexion;
   TorqueVal=str2double(handles.DesiredWeight)*(0.5);
   set_param(handles.TorqueIn, 'Value',num2str(TorqueVal));

elseif countCase<=27
   set_param(handles.Case,
'Value',num2str(handles.RandomNumber(countCase-18)));

set(rover.hrover1,'Xdata',handles.timeArray,'Ydata',handles.(['Signal',
num2str(countCase-18)]));
   handles.(handles.CaseNum).CaseValue=handles.RandomNumber(countCase-
18);

handles.(handles.CaseNum).Signal=45.*handles.(['Signal',num2str(countCa
se-18)]);
    handles.(handles.CaseNum).SignalTime=handles.timeArray;
   handles.DesiredWeight=handles.DesiredWeight_Extension;
   TorqueVal=str2double(handles.DesiredWeight)*(-0.5);
   set_param(handles.TorqueIn, 'Value',num2str(TorqueVal));
else
   set_param(handles.Case,
'Value',num2str(handles.RandomNumber(countCase-27)));

set(rover.hrover1,'Xdata',handles.timeArray,'Ydata',handles.(['Signal',
num2str(countCase-27)]));
   handles.(handles.CaseNum).CaseValue=handles.RandomNumber(countCase-
27);

handles.(handles.CaseNum).Signal=45.*handles.(['Signal',num2str(countCa
se-27)]);
   handles.(handles.CaseNum).SignalTime=handles.timeArray;
   handles.DesiredWeight=handles.DesiredWeight_Extension;
   TorqueVal=str2double(handles.DesiredWeight)*(-1);
   set_param(handles.TorqueIn, 'Value',num2str(TorqueVal));
end


set_param(handles.modelName,'StopTime','16');
set_param(handles.modelName,'SimulationMode','external');
% Connect to the code
set_param(handles.modelName,'SimulationCommand','connect');
% start_button the model
set_param(handles.modelName,'SimulationCommand','start');

% Attach the listener
      if countCase==36
          set(handles.Start_Model,'Enable','off');
          set(handles.Stop_Model,'Enable','on');
          set(handles.Next,'Enable','off');
          set(handles.Practice,'Enable','off')

      end
guidata(hObject,handles);
drawnow
localAddEventListener;



% --- Executes on button press in Stop_Model.
function Stop_Model_Callback(hObject, eventdata, handles)
% hObject    handle to Stop_Model (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global countCase
 assignin('base',(handles.CaseNum),handles.(handles.CaseNum));

%csvwrite(strcat(handles.foldername,(handles.CaseNum),'.csv'),(handles.
CaseNum));
 save(strcat(handles.foldername,(handles.CaseNum),'.mat'),'-
struct','handles',(handles.CaseNum));

countCase=0;
set_param(handles.modelName,'SimulationCommand','stop');
% disconnect from the code
set_param(handles.modelName,'SimulationCommand','disconnect');

% set model properties back to their original values
set_param(handles.modelName,'Stoptime',handles.originalStopTime);
set_param(handles.modelName,'SimulationMode',handles.originalMode);
set_param(handles.TorqueIn,'Value','0')
% toggle the buttons
set(handles.Start_Model,'Enable','off');
set(handles.Set_Weight,'Enable','on');
set(handles.Stop_Model,'Enable','off');
set(handles.Load_Model,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Practice,'Enable','off');
% handles.Time=toc;
guidata(hObject,handles);
localRemoveEventListener;

%___________________________Practice _________________________________
% --- Executes on button press in Practice.
function Practice_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to Practice (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
wStr = sprintf('%s\n%s','Please wait while the model loads.');
hw = waitbar(0.5,wStr);
load_system(handles.modelName2)
handles.viewing = struct(...
    'blockName','',...
    'blockHandle',[],...
    'blockEvent','',...
    'blockFcn',[]);
global countCase
countCase=1;
%Either we want to make this a variable or just always use global
position
handles.viewing.blockName = sprintf('%s/Graph',handles.modelName2);
handles.viewing.blockHandle =
get_param(handles.viewing.blockName,'Handle');

%sets up the event listener
handles.viewing.blockEvent = 'PostOutputs';
handles.viewing.blockFcn = @localEventListener;



handles.originalStopTime = get_param(handles.modelName2,'Stoptime');
handles.originalMode =  get_param(handles.modelName2,'SimulationMode');
%handles.originalStartFcn = get_param(handles.modelName2,'StartFcn');
handles.modelAlreadyBuilt = false;
set_param(handles.modelName2,'SimulationMode','normal');
TorqueVal=str2double(handles.DesiredWeight_Flexion);
set_param(handles.PracticeTorque, 'Value',num2str(TorqueVal));
set(handles.Practice,'Enable','on');
set(handles.Stop_Practice,'Enable','on');
set(handles.Load_Model,'Enable','off');

% build_button the model
rtwbuild(handles.modelName2);
% destroy the waitbar

delete(hw);

%setting up  the model
set_param(handles.modelName2,'SimulationMode','external');
set_param(handles.modelName2,'StopTime','inf');
rtwbuild(handles.modelName2);
% Connect to the code
set_param(handles.modelName2,'SimulationCommand','connect');
% start_button the model
set_param(handles.modelName2,'SimulationCommand','start');
% Attach the listener
set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','on');
set(handles.Load_Model,'Enable','off');

guidata(hObject,handles);
```

```
drawnow
localAddEventListener;

% --- Executes on button press in Stop_Practice.
function Stop_Practice_Callback(hObject, eventdata, handles)
% hObject    handle to Stop_Practice (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param(handles.PracticeTorque,'Value','0')
set_param(handles.modelName2,'SimulationCommand','stop');
% disconnect from the code
set_param(handles.modelName2,'SimulationCommand','disconnect');

% set model properties back to their original values
% set_param(handles.modelName2,'Stoptime',handles.originalStopTime);
% set_param(handles.modelName2,'SimulationMode',handles.originalMode);

% toggle the buttons
set(handles.Practice,'Enable','on');
set(handles.Stop_Practice,'Enable','off');
set(handles.Load_Model,'Enable','on');
set(handles.Set_Weight,'Enable','on');
guidata(hObject,handles);
localRemoveEventListener

%_________________________Weight
Test___________________________________
% --- Executes on button press in Start_Weight_Test.
function Start_Weight_Test_Callback(hObject, eventdata, handles)
% hObject    handle to Start_Weight_Test (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global countCase
countCase=0;
handles.(handles.CaseNum).Torque=0;
wStr = sprintf('%s\n%s','Please wait while the model loads.');
load_system(handles.modelName3)
handles.viewing = struct(...
    'blockName','',...
    'blockHandle',[],...
    'blockEvent','',...
    'blockFcn',[]);

%Either we want to make this a variable or just always use global
position
handles.viewing.blockName = sprintf('%s/Graph',handles.modelName3);
handles.viewing.blockHandle =
get_param(handles.viewing.blockName,'Handle');

% %sets up the event listener
handles.viewing.blockEvent = 'PostOutputs';
handles.viewing.blockFcn = @localEventListener;

handles.originalStopTime = get_param(handles.modelName3,'Stoptime');
handles.originalMode =  get_param(handles.modelName3,'SimulationMode');
```

```
%handles.originalStartFcn = get_param(handles.modelName3,'StartFcn');
handles.modelAlreadyBuilt = false;
set_param(handles.modelName3,'SimulationMode','external');

set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Load_Model,'Enable','off');
set(handles.Stop_Weight_Test,'Enable','on');
set(handles.Start_Weight_Test,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Set_Weight,'Enable','off');
set(handles.StartROM,'Enable','off');
set(handles.StopROM,'Enable','off');
% build_button the model
   %rtwbuild(handles.modelName3);
% destroy the waitbar
hw = waitbar(0.5,wStr);
delete(hw);
guidata(hObject,handles);
set_param(handles.modelName3,'StopTime','inf');

% Connect to the code
set_param(handles.modelName3,'SimulationCommand','connect');
% start_button the model
set_param(handles.modelName3,'SimulationCommand','start');
% Attach the listener

drawnow
localAddEventListener;

% --- Executes on button press in Stop_Weight_Test.
function Stop_Weight_Test_Callback(hObject, eventdata, handles)
% hObject    handle to Stop_Weight_Test (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param(handles.modelName3,'SimulationCommand','stop');
% disconnect from the code
set_param(handles.modelName3,'SimulationCommand','disconnect');

% set model properties back to their original values
%set_param(handles.modelName3,'Stoptime',handles.originalStopTime);
%set_param(handles.modelName3,'SimulationMode',handles.originalMode);

% toggle the buttons
set(handles.Load_Model,'Enable','off');
set(handles.Start_Weight_Test,'Enable','on');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Start_Model,'Enable','off');
set(handles.Stop_Model,'Enable','off');
set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Start_Weight_Test,'Enable','on');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Set_Weight,'Enable','on');
set(handles.StartROM,'Enable','on');
set(handles.StopROM,'Enable','off');

MaxWeight=max(abs(handles.(handles.CaseNum).Torque));
set(handles.Max_Weight,'String',MaxWeight);
guidata(hObject,handles);
localRemoveEventListener;

%_______________________Count Down Edit________________________________
function Count_Down_Callback(hObject, ~, handles)
% hObject    handle to Count_Down (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Count_Down as text
%        str2double(get(hObject,'String')) returns contents of
Count_Down as a double


% --- Executes during object creation, after setting all properties.
function Count_Down_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Count_Down (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%_______________________________________________________________________
_



% --- Executes on button press in Change_Weight.
function Change_Weight_Callback(hObject, eventdata, handles)
% hObject    handle to Change_Weight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Not used
% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2


% --- Executes on button press in Set_Weight.
function Set_Weight_Callback(hObject, eventdata, handles)
% hObject    handle to Set_Weight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Load_Model,'Enable','on');
set(handles.Start_Model,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Practice,'Enable','on');
set(handles.Stop_Model,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Start_Weight_Test,'Enable','on');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Set_Weight,'Enable','off');
set(handles.StartROM,'Enable','on');
set(handles.StopROM,'Enable','off');

handles.DesiredWeight_Flexion=get(handles.Weight_Applied_Flexion,'Strin
g');
handles.DesiredWeight_Extension=get(handles.Weight_Applied_Extension,'S
tring');
guidata(hObject,handles);



%__________________Max Weight
Display____________________________________
function Max_Weight_Callback(hObject, eventdata, handles)
% hObject    handle to Max_Weight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Max_Weight as text
%        str2double(get(hObject,'String')) returns contents of
Max_Weight as a double

% --- Executes during object creation, after setting all properties.
function Max_Weight_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Max_Weight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%________________________Weight
(MVF)_____________________________________
%Takes entered weight and saves it into the simulink model
function Weight_Applied_Flexion_Callback(hObject, eventdata, handles)
% hObject    handle to Weight_Applied_Flexion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
Weight_Applied_Flexion as text
```

```
%        str2double(get(hObject,'String')) returns contents of
Weight_Applied_Flexion as a double


% --- Executes during object creation, after setting all properties.
function Weight_Applied_Flexion_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Weight_Applied_Flexion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Weight_Applied_Extension_Callback(hObject, eventdata, handles)
% hObject    handle to Weight_Applied_Extension (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
Weight_Applied_Extension as text
%        str2double(get(hObject,'String')) returns contents of
Weight_Applied_Extension as a double


% --- Executes during object creation, after setting all properties.
function Weight_Applied_Extension_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to Weight_Applied_Extension (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%__________________________Range of
Motion_________________________________

% --- Executes on button press in StartROM.
function StartROM_Callback(hObject, eventdata, handles)
% hObject    handle to StartROM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global countCase
countCase=0;
```

```
wStr = sprintf('%s\n%s','Please wait while the model loads.');
hw = waitbar(0.5,wStr);
load_system(handles.modelName4)
handles.viewing = struct(...
    'blockName','',...
    'blockHandle',[],...
    'blockEvent','',...
    'blockFcn',[]);

%Either we want to make this a variable or just always use global
position
handles.viewing.blockName = sprintf('%s/Graph',handles.modelName4);
handles.viewing.blockHandle =
get_param(handles.viewing.blockName,'Handle');

%sets up the event listener
handles.viewing.blockEvent = 'PostOutputs';
handles.viewing.blockFcn = @localEventListener;



handles.originalStopTime = get_param(handles.modelName4,'Stoptime');
handles.originalMode =  get_param(handles.modelName4,'SimulationMode');
%handles.originalStartFcn = get_param(handles.modelName2,'StartFcn');
handles.modelAlreadyBuilt = false;
set_param(handles.modelName4,'SimulationMode','normal');



set(handles.Practice,'Enable','on');
set(handles.Stop_Practice,'Enable','on');
set(handles.Load_Model,'Enable','off');

% build_button the model
rtwbuild(handles.modelName4);
% destroy the waitbar

delete(hw);

%setting up  the model
set_param(handles.modelName4,'SimulationMode','external');
set_param(handles.modelName4,'StopTime','inf');
rtwbuild(handles.modelName4);
% Connect to the code
set_param(handles.modelName4,'SimulationCommand','connect');
% start_button the model
set_param(handles.modelName4,'SimulationCommand','start');
% Attach the listener
set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Load_Model,'Enable','off');
set(handles.Start_Weight_Test,'Enable','off');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Set_Weight,'Enable','off');
set(handles.StopROM,'Enable','on');

guidata(hObject,handles);
```

```
drawnow
localAddEventListener;

% --- Executes on button press in StopROM.
function StopROM_Callback(hObject, eventdata, handles)
% hObject    handle to StopROM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set_param(handles.modelName4,'SimulationCommand','stop');
% disconnect from the code
set_param(handles.modelName4,'SimulationCommand','disconnect');

set(handles.RangeMax,'String',max((handles.(handles.CaseNum).Human)));
set(handles.RangeMin,'String',min((handles.(handles.CaseNum).Human)));

set(handles.Load_Model,'Enable','off');
set(handles.Start_Model,'Enable','off');
set(handles.Stop_Model,'Enable','off');
set(handles.Practice,'Enable','off');
set(handles.Stop_Practice,'Enable','off');
set(handles.Next,'Enable','off');
set(handles.Start_Weight_Test,'Enable','on');
set(handles.Stop_Weight_Test,'Enable','off');
set(handles.Set_Weight,'Enable','off');
set(handles.StartROM,'Enable','on');
set(handles.StopROM,'Enable','off');

guidata(hObject,handles);
localRemoveEventListener;

function RangeMin_Callback(hObject, eventdata, handles)
% hObject    handle to RangeMin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RangeMin as text
%        str2double(get(hObject,'String')) returns contents of RangeMin
as a double

% --- Executes during object creation, after setting all properties.
function RangeMin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RangeMin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function RangeMax_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to RangeMax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RangeMax as text
%        str2double(get(hObject,'String')) returns contents of RangeMax
as a double

% --- Executes during object creation, after setting all properties.
function RangeMax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RangeMax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%______________________________________________________________________
_

function Pnumber_Callback(hObject, eventdata, handles)
% hObject    handle to Pnumber (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Pnumber as text
%        str2double(get(hObject,'String')) returns contents of Pnumber
as a double


% --- Executes during object creation, after setting all properties.
function Pnumber_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Pnumber (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%____________________EVENT
LISTENERS______________________________________
%Adding the event listener when start button is pushed
function localAddEventListener
% get the application data
handles = guidata(gcbo); %Pull the current handles data.
```

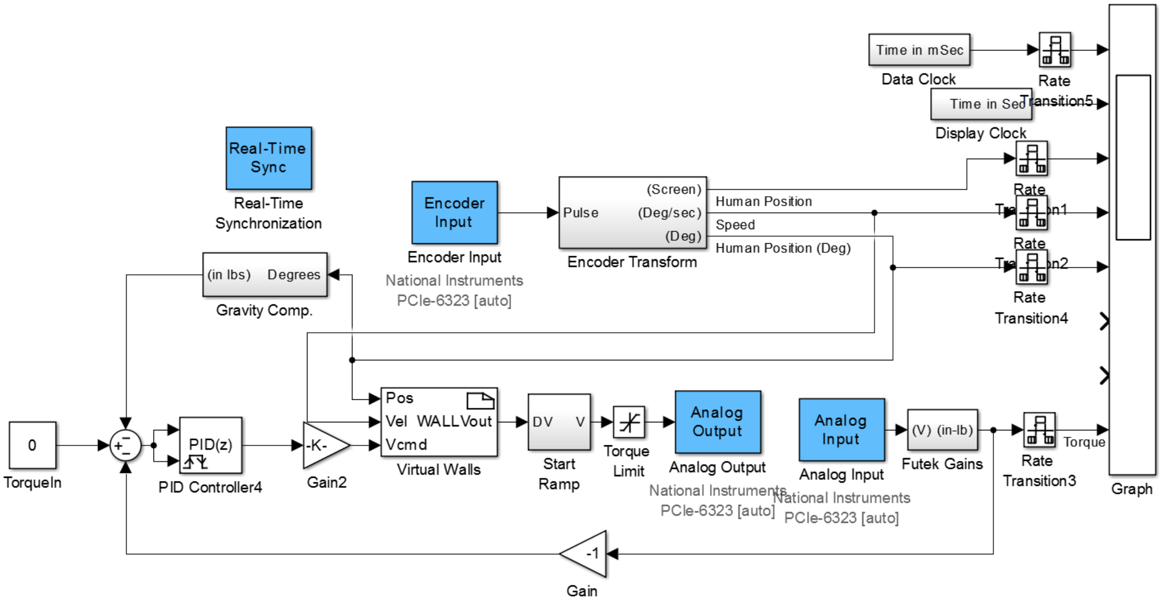
```
handles.eventHandle(1) =
add_exec_event_listener(handles.viewing.blockName,...
    handles.viewing.blockEvent,handles.viewing.blockFcn);
handles.index=0;
% store the changed app data
guidata(gcbo,handles);

%*****Looking at clock,torque and position blocks********
function localEventListener(block, eventdata)
% get the application data this mfilename had to be initialized in the
% code, will not work if not initilized
global countCase
hf = findall(0,'tag',mfilename);
%Event listeners need this stuff drawn in manually like this instead of
%throught the function inputs.
handles = guidata(hf);
rover=handles.rover;
%Read in signals from simulink
DataTimer =       block.InputPort(1).Data;
DisplayTimer =    block.InputPort(2).Data;
DisplayPosition = block.InputPort(3).Data;
Speed =           block.InputPort(4).Data;
rover.yPosition2= block.InputPort(5).Data;
torque=           block.InputPort(8).Data;

%Update the position of the circles on the Gui screen
%Save signals in arrays to be saved in a folder later.
 handles.(handles.CaseNum).Speed(end+1)=Speed;
 handles.(handles.CaseNum).Human(end+1)=rover.yPosition2;
 handles.(handles.CaseNum).Torque(end+1)=torque;
 handles.(handles.CaseNum).Time(end+1)=DataTimer;
if DataTimer<0
    set(handles.Count_Down,'String',DisplayTimer*(-1));
    set(handles.Count_Down,'BackgroundColor',[1,1,1])
elseif DataTimer<12.8
      set(handles.Count_Down,'String','Go')
      set(handles.Count_Down,'BackgroundColor',[0,1,0.1])
else
    set(handles.Count_Down,'String','Stop')
    set(handles.Count_Down,'BackgroundColor',[1,0,0.1])
end
handles.index=handles.index+1;
if DataTimer<0
    set(rover.hrover2,'Xdata',0,'Ydata',DisplayPosition);
else
    set(rover.hrover2,'Xdata',DataTimer,'Ydata',DisplayPosition);
end

% The axes limits may also need changing. This shifts the axis -0.05
every
% time step. This makes it look like you are moving forward.
if DataTimer>0
newXLim = [(DataTimer-0.5) (DataTimer+4.5)];
set(handles.Graph,'Xlim',newXLim);
guidata(hf,handles)
else
```

```
newXLim = [(-0.5) (4.5)];
set(handles.Graph,'Xlim',newXLim);
guidata(hf,handles)
end
%Update GUI screen
drawnow;

function localRemoveEventListener
% get the application data
handles = guidata(gcbo);

% delete the listener(s)
delete(handles.eventHandle(1));
% remove this field from the app data structure
handles = rmfield(handles,'eventHandle');

%save the changes
guidata(gcbo,handles);
%__________________________CLOSING THINGS___________________________
%Closes the model before closing the GUI
function localCloseRequestFcn(hObject,eventdata)

handles = guidata(hObject);
%this just checks if their is a model open or not and stops the model
%before closeing deleting the gui

if
~isempty(find_system('Type','block_diagram','Name',handles.modelName))
    switch get_param(handles.modelName,'SimulationStatus');
        case 'stopped'
            delete(gcbo);
        otherwise
            stop_button_Callback(hObject, eventdata, handles);
            delete(gcbo);
    end
else
    delete(gcbo);
end
```

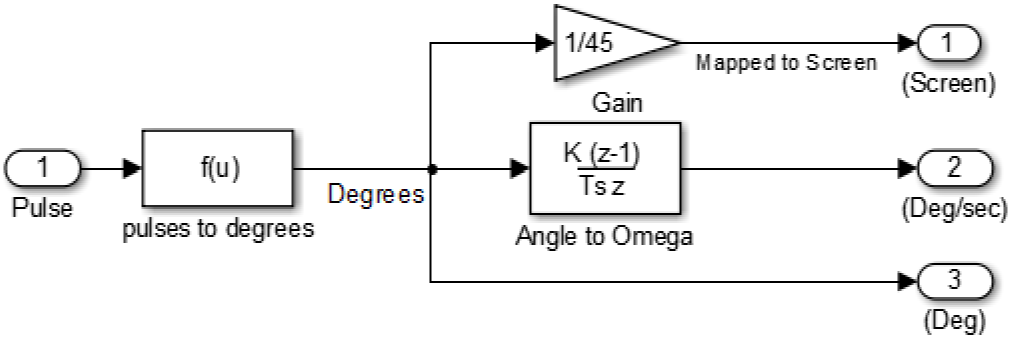
APPENDIX D

SIMULINK MODELS

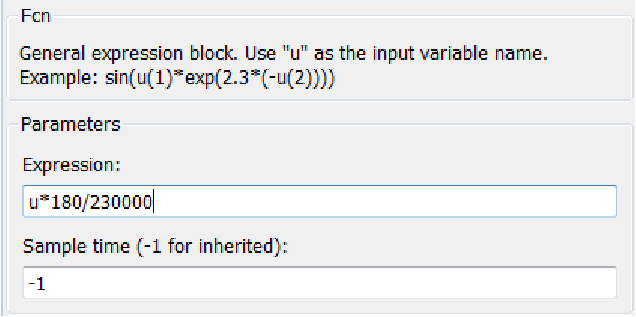
D.1. MastersGUI_model



Encoder Transform



Pulses to degrees



Fcn

General expression block. Use "u" as the input variable name.
Example: sin(u(1)*exp(2.3*(-u(2))))

Parameters

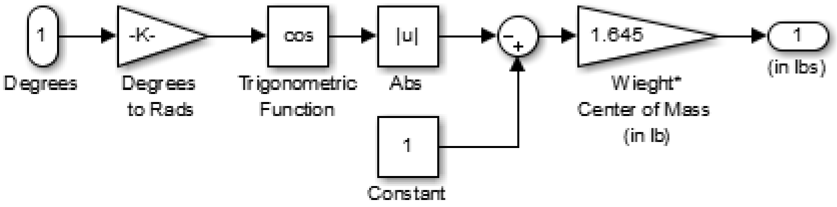
Expression:

u*180/230000

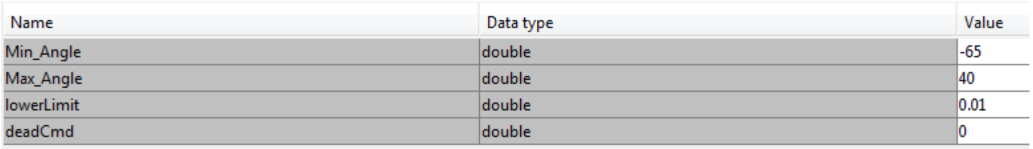
Sample time (-1 for inherited):

-1

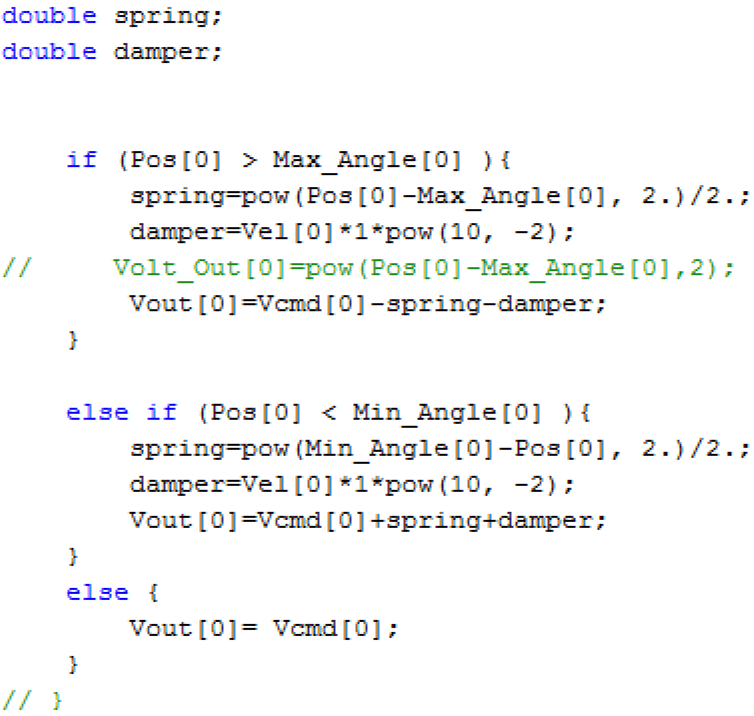
Gravity Compensation



Virtual Walls

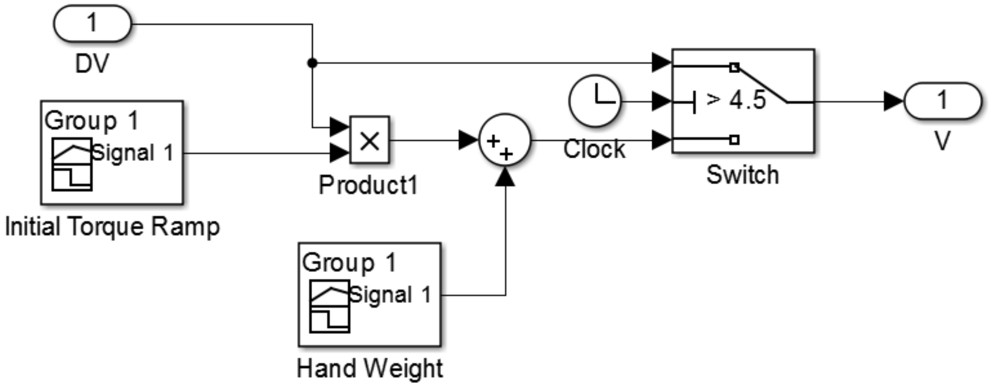


Name	Data type	Value
Min_Angle	double	-65
Max_Angle	double	40
lowerLimit	double	0.01
deadCmd	double	0

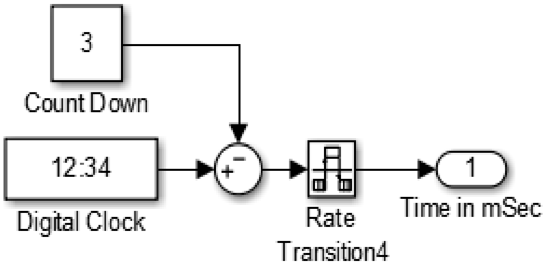


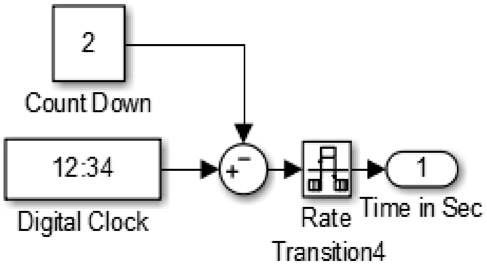
```
double spring;
double damper;


    if (Pos[0] > Max_Angle[0] ){
        spring=pow(Pos[0]-Max_Angle[0], 2.)/2.;
        damper=Vel[0]*1*pow(10, -2);
//      Volt_Out[0]=pow(Pos[0]-Max_Angle[0],2);
        Vout[0]=Vcmd[0]-spring-damper;
    }

    else if (Pos[0] < Min_Angle[0] ){
        spring=pow(Min_Angle[0]-Pos[0], 2.)/2.;
        damper=Vel[0]*1*pow(10, -2);
        Vout[0]=Vcmd[0]+spring+damper;
    }
    else {
        Vout[0]= Vcmd[0];
    }
// }
```

Start Ramp

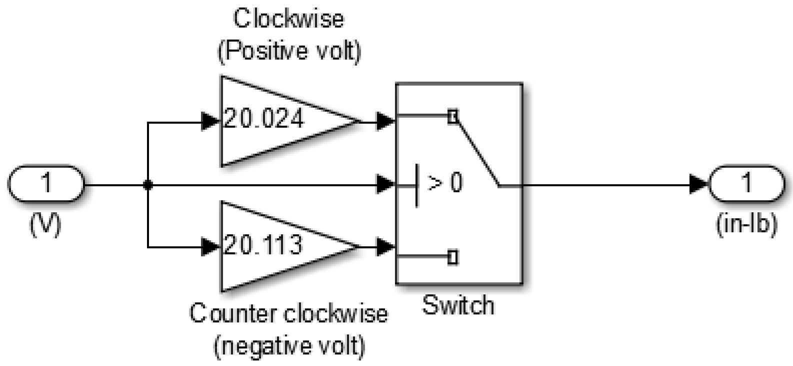


Data Clock/ Display Clock

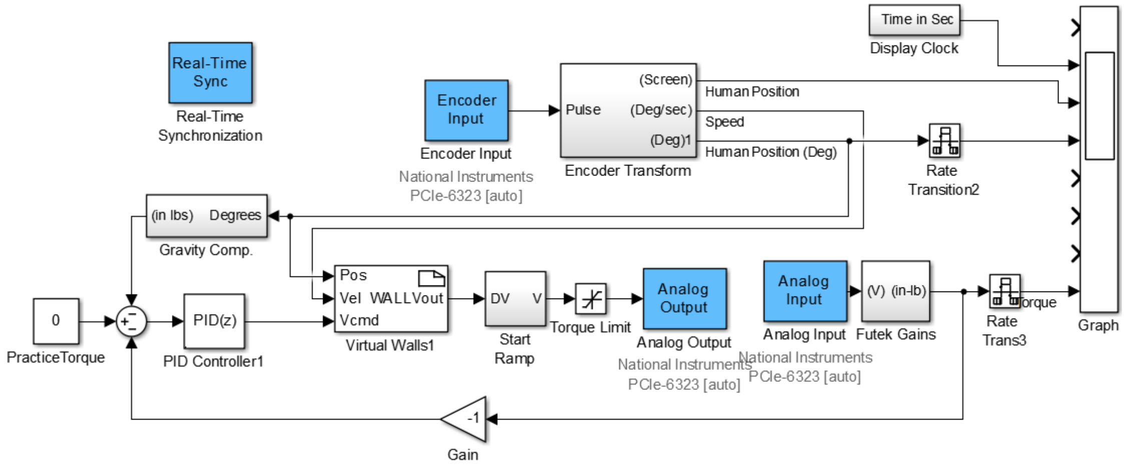




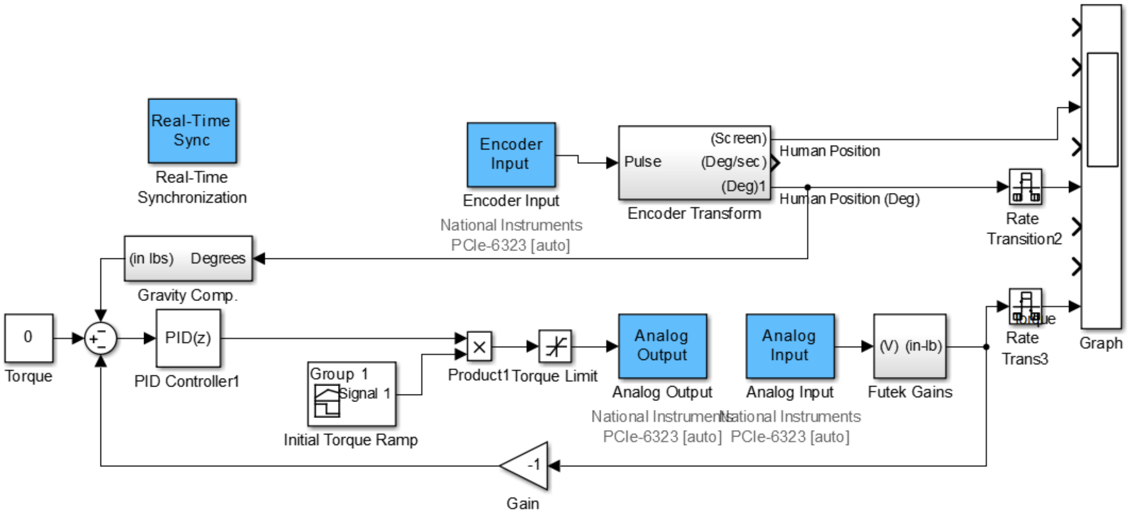
Futek Gains



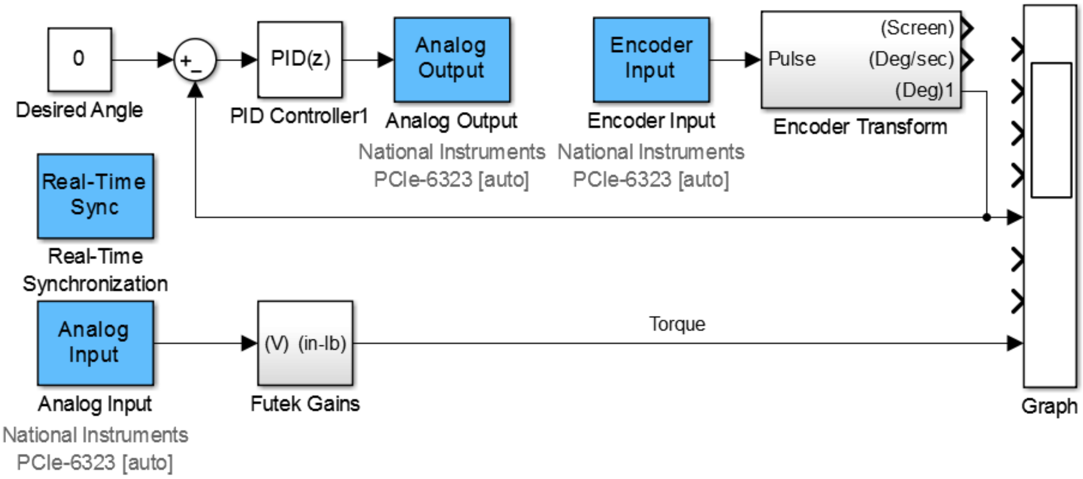
D.2. Practice_model



D.3. ROM_model



D.4. Weight_model


0
Desired Angle
PID(z)
PID Controller1
Analog Output
Analog Output
National Instruments
PCIe-6323 [auto]
Encoder Input
Encoder Input
National Instruments
PCIe-6323 [auto]
Pulse
(Screen)
(Deg/sec)
(Deg)1
Encoder Transform
Real-Time Sync
Real-Time Synchronization
Analog Input
Analog Input
National Instruments
PCIe-6323 [auto]
(V) (in-lb)
Futek Gains
Torque
Graph