An experimental investigation of manipulation control based on facial EMG signals

By

Gordon R. Daily

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Advisor: Dr. Wyatt S. Newman

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

May 2001

SIGNATURE SHEET

ABSTRACT

This BS/MS project constitutes an initial exploration of control interface options for tetraplegics. Specifically, the use of electromyographic (EMG) signals from facial muscles was investigated for control of a robot, with the expectation of extensions to functional neural stimulation (FNS) control of one's own muscles. This investigation involved design of a suitable testbed, including development of EMG-based control of an articulated robot, innovation of alternative candidate control algorithms, and human-subject evaluations of the alternative algorithms. Through timed dexterity tests performed by nearly 30 volunteers, a preferred control mode was identified. The conclusions will have applications to both FNS restoration of arm functions as well as rehabilitation and service robotics.

DEDICATION

I would like to dedicate this project to my parents who have been the push behind my academic endeavors. They have supported me from the beginning giving me the opportunity to seek my goals. Their guidance has brought me all of the successes in my life.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ACKNOWLEDGMENTS

I would like to acknowledge my advisor, Professor Newman, for an optimistic attitude, gentle guidance, and healthy sense of humor. I also thank Professors Branicky and Kirsch for serving on my MS project committee and taking the time to provide feedback and support.

Those willing to take time to be tested deserve my appreciation for volunteering their time and face for this cause.

1 - Introduction

1.1 Background

1.1.1 Real Life Statistics

In America today there are 250,000 - 400,000 individuals living with Spinal Cord Injury or Spinal Dysfunction. Of these, 34.4% become paraplegic and 24.3% become quadriplegic[1]. This concurs with other estimates that at least 44,000 people are considered quadriplegics[2]. Unfortunately, in the event of these types of injuries, almost all physical interaction with the surrounding world is prevented; the head remains the only controllable part of the body. Fortunately, since facial muscles retain their ability for motion use in speech, chewing, and other facial manipulations, they can be further utilized for increased interaction with the environment. This study aims to find the most effective means of controlling a robotic arm in three-dimensional space by the detection of facial muscle stimulation.

This project is not the first of its kind but follows others that also aim at facilitating more accessible interaction between the handicapped and society, including moving a mouse pointer, controlling a wheelchair, and one or two-degree motion prosthetics attached to the body.

1.1.2 EMG Definition

EMG (electro myo graph) and electro-neuro-myography are diagnostic medical technologies used to measure the attenuation and speed at which brief electrical impulses travel in nerve axons. The surface electrodes used in this experiment are the same high quality type used for EKG’s (electrocardiograms). Surface electrodes simply adhere to the skin (like miniature band-aids). The sensor is made of small amounts of conductive gel packaged within an adhesive tape and snap-on connector.

The EMG activity from the contraction of muscles under the skin is "conducted" from the two electrodes through small wires to a special instrumentation amplifier, integrator, and analog-to-digital converter for sending the EMG to the computer. The electronic instrumentation amplifiers are designed to reduce noise by using differential amplifiers for each input; therefore, two electrodes are required for each EMG channel, and all four signals use a shared ground electrode. Since we are looking at the electrical signals coming from deep inside the muscle, through the skin, the instrumentation amplifiers must be extremely sensitive (high gain). The equipment will "display" the electrical activity from the muscle contraction even before you can palpate the contraction.

1.2 Goals

1.2.1 Feasibility

The first attempt to determine the feasibility of the experiment was to observe and enhance the signals from the amplifier. Since I brought my body with me everywhere I went, I provided the standard experimentation face. Determining if the signals were in fact consistent with the user intent was the goal of the initial testing. To verify the feasibility, it was essential that instant feedback on the produced signals be displayed. After the creation of the display scope software, it was concluded that EMG signals would be a good way to infer a user’s intent. This motivated subsequent development of EMG-based robot control software.

1.2.2 Sensor Placement and Filtering

Determining how to achieve good sensitivity from facial EMG signals was a project in itself. There were many factors that had to be thought through when generating a consistent and reliable signal, let alone three consistent reliable signals. There were two aspects to obtaining quality signals from the user. The first was to determine where the electrodes to detect the electrical impulses from the muscles were to be placed on the face. The second was to determine the best settings for filtering and amplification once these signals were fed into the EMG filter.

1.2.3 Proposition and Evaluation of Alternative Control Schemes

The ultimate goal of this project was to decide how best to manipulate the robot with EMG inputs. Using the data read in and averaged by the computer, this project explored seven different algorithms for mapping EMG signals onto robot motion commands.

Experimental evaluation proved to be a complex undertaking. Testing was designed to determine speed, stability, and how learning curves affect the ability to complete various tasks that disabled people will encounter from day to day. It was unrealistic to find volunteers willing to give enough time for a more involved experiment. Thus, the experiment was designed for a large number of volunteers to perform tests only once. Each test lasted approximately 90 minutes. Initially, the goal was to prepare a very formal and consistent test with many participants doing fairly simple tasks. Once testing began, however, it became apparent that there were large individual differences between volunteers. Each individual needed to be coached in a slightly different manner before he was able to pass what were thought to be simple tests. As a result, experimentation changed over time as we tried to refine how to create a consistent benchmark to be tested by few people over a longer time period.

1.3 Project Constraints

This project was limited by the precision of robot movement. Human subjects were mostly college age white males due to the predominantly white male population on campus. Of course, it was also limited because every individual has a unique face. In a real life application, individuals will have a more focused effort when the system is to be configured over the long term with a patient. The patient's ability to learn and his hand eye coordination also influences his ability to perform these tests.

2.0 System Overview

2.1 EMG Signal Hardware

The device used to receive the signals from the electrodes placed on the face was the Cambridge Electronic Design Limited model 1902 (Cambridge, UK). This device is used for EMG, EEG, ECG, ERG, Evoked response, Skin Conductance, Tremor Measurement, Auditory Brainstem and many more life science and engineering research applications. The 1902 signal conditioner is a versatile modular unit designed to work in harmony with modern computer-controlled data acquisition systems. Developed for a broad range of applications, the 1902 can accept signals from a wide variety of sources.

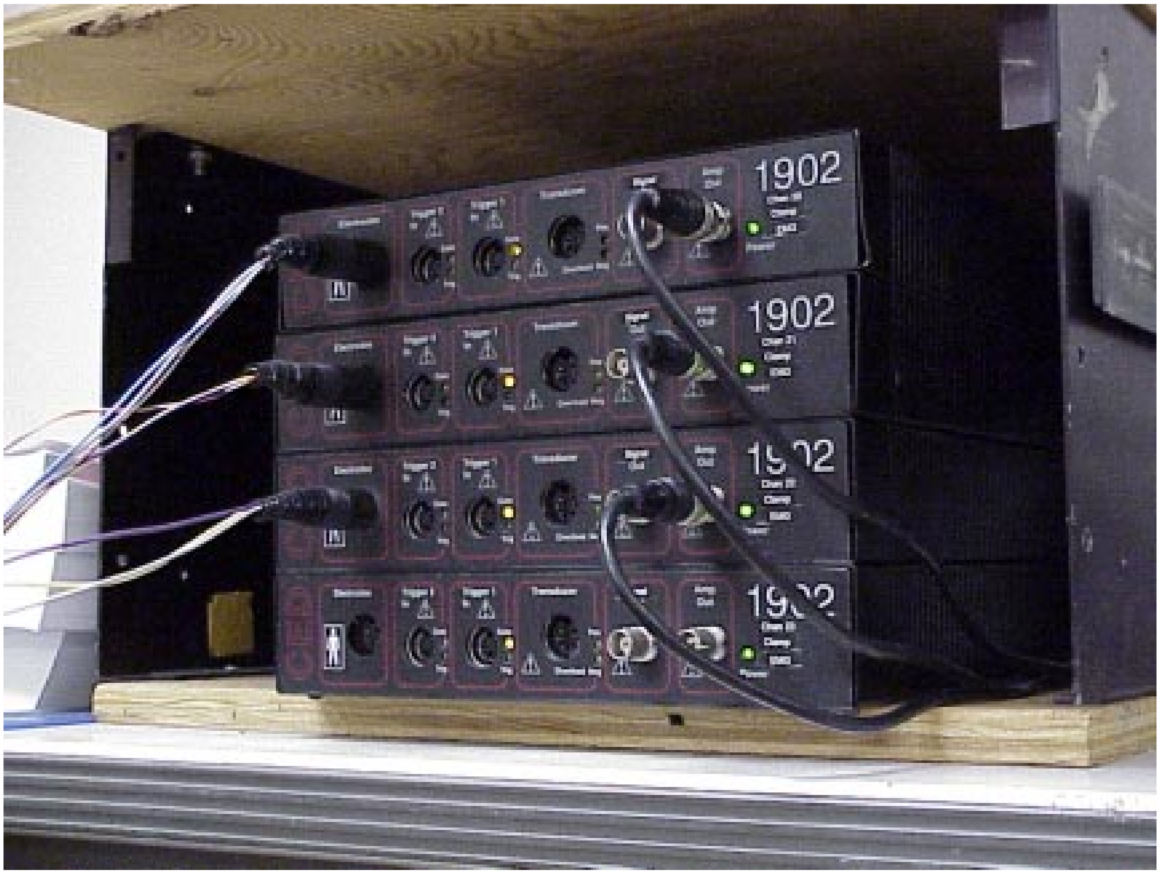


Figure 37 - EMG filter and amplifier (CED 1902)

In our application, we used seven electrodes. The common ground was connected to an electrode on the elbow. There were three sets of two electrodes, each set prescribed as an independent input to the others as discussed earlier in the explanation of how EMG's work.

Options that the 1902 offered:

Programmable gain with readback
Dynamically controlled 12-bit offset
Selectable high and low pass filter settings
Mains notch filter on/off
Menu and Hot Key control
AC/DC coupling

A Windows application called Lab View, by National Laboratories (Austin, TX), was used to configure the amplifier. A project for Lab View was written by the Cleveland FES center configures the above options using a RS232 serial port located on the back of the unit.

2.2 Computer and Interconnect Hardware

Data acquisition of the filtered signals was read in by an I/O board connected to an ISA data acquisition card on the computer. The computer was running QNX, a real time Unix-like operating system using Photon as its window manager. The filtered signals as output from the CED 1902 EMG filter ranged in voltage from zero to five volts. An I/O software process used the I/O board to read the voltages and place the digital equivalents into a shared memory location. Similarly, robot control software placed control commands in shared memory. I/O process polled these values and outputted them using the digital-to-analog converters on the I/O board interfed to the Rhino power amps.

2.3 Robot

2.3.1 Robot Specifications

Introduced in 1982, the Rhino XR-1 Robot is a motor/chain driven robot made by Sandhu Machine Design, Inc (Champaign Ill.). It allows movement along six joints: base, shoulder, elbow, wrist, hand and fingers. For our application, we focused on controlling the robotic hand in three dimensions. Although the software written for this project controlled all six joints, only the first three joints -- base, shoulder and elbow -- were used.

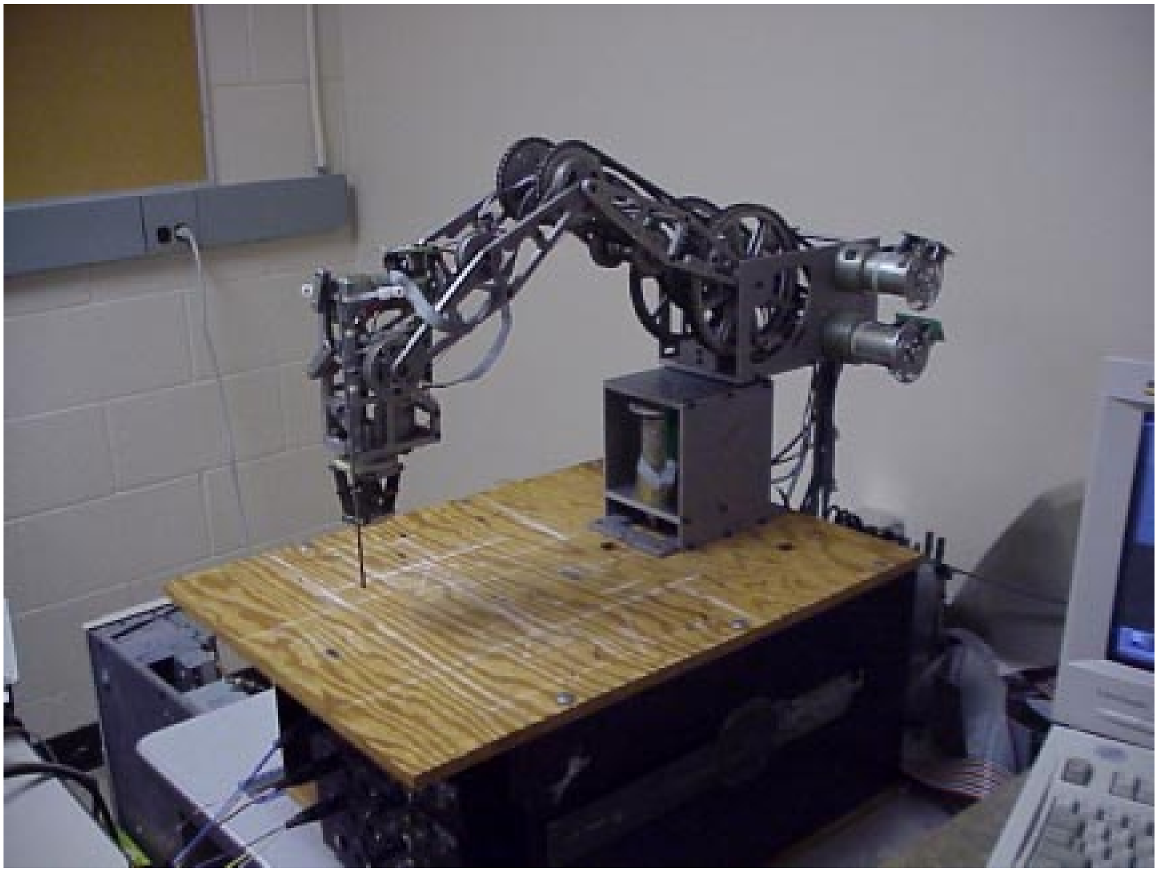


Figure 38 - Rhino Robot

2.3.2 Robot Kinematics

All kinematics were based on joints one, two, and three (i.e. base, shoulder and elbow). In the inverse direction, inverse kinematics took a three-dimensional Cartesian point and converted it into the necessary angles required for the robot to achieve that position. Forward kinematics were calculated by solving for a Cartesian point based on the joint angles. Rick Hudson and Marco Ngai wrote the forward and inverse kinematics, along with a simplified GUI, for a previous experiment. Their software allowed direct switching between both coordinate systems.

2.3.3 Robot Error Significance

There were several shortcomings of the Rhino robot that reflected its age. The accuracy of the Rhino robot was dependant on the calibration mechanism, the precision of measurements made for calculating the kinematics, and the rate at which the joints were updated for the robot. The calibration mechanism required that the robot travel to the extent of its range where a switch was triggered to create a reference point. Over time, the robot lost the ability to manipulate all joints due to a malfunctioning controller. Some of the homing switches also needed to be replaced, thus the previous calibration routine became useless. Instead, the robot was placed in a position with all angles visually estimated at zero. Starting the robot in an initial position gave us a reference point that was easy to keep relatively consistent with a simple line of sight. The measurements for kinematics could be calculated more precisely by fixing the previous homing method, but this improvement would have been marginal.

A more dangerous error occurred when the robot operated at a high speed. When this situation arose, the kinematics calculated a target angle for each joint to instantly attain without waiting for the robot to hit every point in between two points. For example, if commanded to move in a straight line from the negative extreme to the positive extreme along the x-axis at a high speed, joints two and three would not move fast enough to maintain coordination. Thus, the robot swung out far in the y direction even though the user did not intend for this motion to occur. In order to resolve this discrepancy, speed was limited so that the robot could travel steadily along the desired path. We thus picked an end effecter speed that the robot could attain everywhere in the workspace.

The primary function of this robot, however, was to test the effectiveness of facial EMG based control. The person driving the robot used the current position as a reference point, for differential motion commands, and thus it was not imperative that the robot's movements or placement be highly accurate. Over the course of experimentation, it became very apparent that the serious concerns of this project were not the accuracy of the robot but rather the users' consistency to reproduce signals and the computer's ability to infer what those signals meant.

3.0 Sensor Placement and Filtering

3.1 Electrode Placement

The face has a very complex arrangement of muscles. Most facial movements and expressions are a result of collective muscle activity. Unfortunately for this project, the brain’s inability to control each muscle directly without stimulating other muscles made it difficult to isolate EMG signals. To make matters more complicated, at the beginning of the project, it was decided that mouth muscles were not to be used in order to allow people to talk and chew food without interfering with the signals.

Since the goal of this project was to control movement in three-dimensional space, the control program was designed to utilize three inputs: one for the x-axis, one for the y, and one for the z. Each signal input required two electrodes. Since the area of the skin available for placement of the electrodes was already limited to the area above the mouth and the use of two electrodes per signal further limited the amount of space available (adhering sensors require substantial surface area), deciding where to place the electrodes relative to each other was a complicated task. In order to produce consistent data, all three signals had to be independently controllable.

In order to allow for this independent movement, it was imperative to determine which muscle groups the average human could control directly. Everyone tested was able to raise his or her eyebrows, but a person who could raise one eyebrow independent of the other was rare. The ability to wink the left and right eyes separately was a common feature among most test subjects. There were two people, however, who were unable to squint each upon demand. These people were omitted from testing.

Therefore, only three independent muscle groups were found that could be adequately read by the EMG machine in the majority of test subjects: the forehead and the upper cheek muscles under each eye. When squinting, the muscles below the eyes flex quite strongly. This action is a default body function used to shield the eye from light and incoming objects.

3.2 Filter Configuration

Configuring the filter took customization for each individual. Once the subject was wired to the machine, the QNX application was run to view the signals graphically. The Lab View project was opened and started on the Windows machine. The following diagram shows the layout of the electronic filtering:

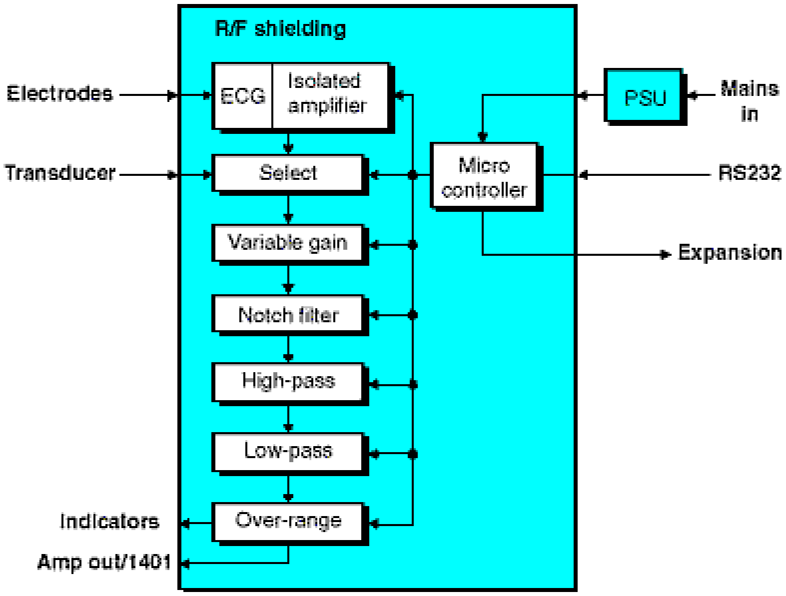
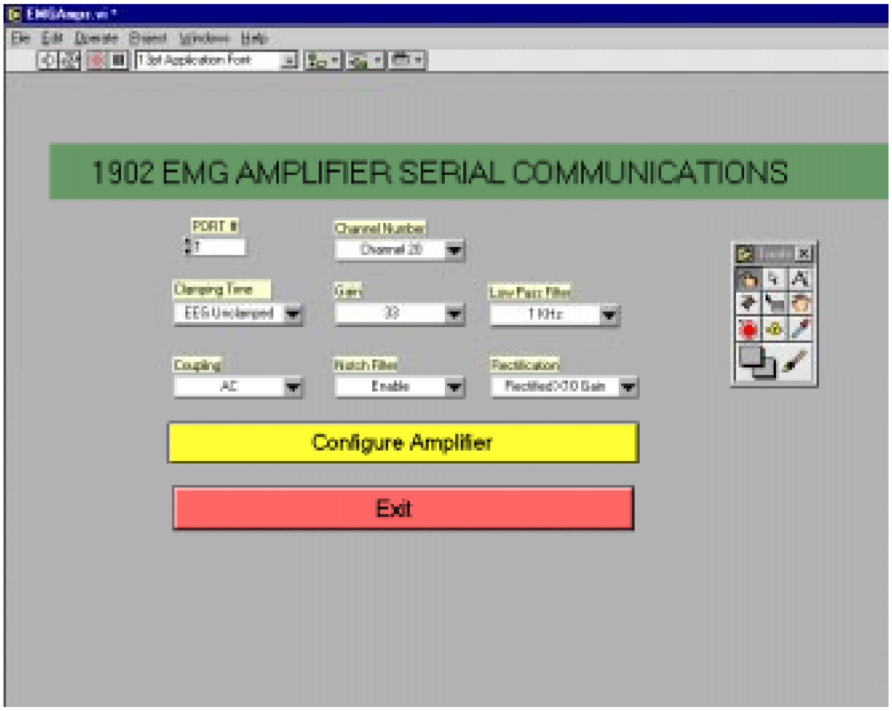


Figure 39 - Signal Filter Flowchart

The following settings were used:

Unclamped EEG
9900/33000 Gain
Low Pass filter at 2Hz
AC coupling
Notch filter enabled
x1 & x10 Rectification

The only settings that varied between users were the gain and rectification. The remaining settings were unchanged throughout the experiments. Until the low-pass filter was set to 2 Hz, the signal was unrecognizable on the oscilloscope and screen. The channel number specified which channel was to be configured, which was set for “All Channels” so that all four amplifiers would share the same settings. Here is a picture of the application used to configure the amplifier:



PORT # 1
Channel Number: Channel 20
Clamping Time: EEG Unclamped
Gain: 33
Low Pass Filter: 1 KHz
Coupling: AC
Notch Filter: Enable
Rectification: RectifiedX10 Gain

Figure 40 - Lab View EMG Configuration Utility

3.3 Electrode Error Significance

There were two important factors that prevented accurate electrode readings. It was very important for the subjects to wash their faces before electrodes were placed. Natural body oils and sweat caused the electrodes to loose their ability to adhere to the skin. Some subjects over time became rather sweaty and their electrodes occasionally needed to be replaced.

The other danger to consistent readings was proper grounding. Although we had no problem with losing the grounded electrode (there is little body oils on the elbow skin), there was a very large disturbance in the ground when the subject made contact with a large metallic surface. Often, the subject would periodically touch the large table adjacent to their seat. Once this problem was discovered, it was easily avoided by quick instruction.

4.0 Software

4.1 Code Flow Layout & General Description

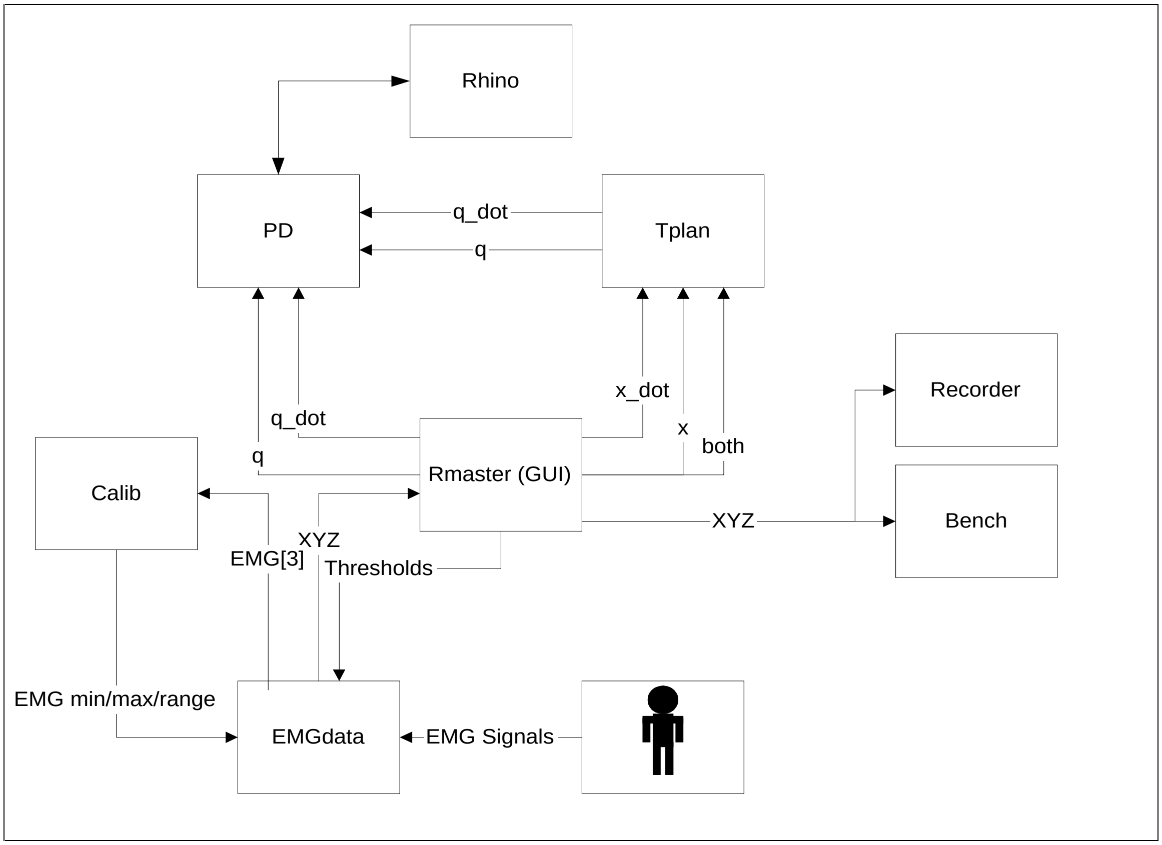


Figure 41 - Software Communication

The processes communicated via data manipulation in shared memory. There was semaphore locking access to the global memory so that only one process at a time was permitted to read or write from the data. Processes PD (positioning and derivation) and Tplan (trajectory planner) had the job of constantly updating the current robot position according to the data in shared memory. If the GUI mode button was in Cartesian mode or XYZ mode, these processes calculated the inverse kinematics and moved the joints to the appropriate angles. If the GUI mode button was in Joint mode, these processes queried the shared memory variables of each joint angle command and moved the robot accordingly.

There were two processes that accessed and updated the actual data for robot motion. The GUI process controlled both XYZ positions and joint angles directly through the interface. Under XYZ mode, there were sliders on the GUI that could be manipulated to directly control each axis. If in joint mode, there were 5 sliders that had direct control over each angle of each corresponding joint. The EMG data process had access to update only the XYZ data based upon the analysis of the EMG readings. A button on the GUI that cycled through all possible EMG control modes determined the analysis applied to the signals by the process EMGdata. The calibration process also affected position changes by process EMGdata since it established the maximum and minimum EMG readings thereby defining the ranges with which the received signals were to be judged.

Once the calibration routine was complete, the benchmarking program was signaled to begin. This process controlled the remaining test sequence and wrote the progress of the test to an output file as it occurred.

The Recorder process kept a constant tally of robot position and time for future data analysis. (See figures 35 and 36)

4.2 Process Control

4.2.1 – The Rmaster GUI & Rhinocb.c (callback functions)

The Rmaster process, formally know as the “Rhino Master” program, was the only executable process that ran from the command line. It accepted no arguments for input and ran with a priority of ten. Initially, it prompted for the first and last names of the test subject, the type of analysis method used, and the number of times the method was attempted successfully. Using this information, the Rmaster process specified the name of the output files as strings in shared memory. Before spawning off all supporting processes, Rmaster initialized all global memory values in one location. Next, it prepared the GUI by instantiating buttons and sliders. It also created a timer function that updated the sliders that needed to be updated. Since these values were being changed by the process Tplan, if in joint mode, this timer used a callback function to examine the XYZ slider positions, compared to the value in memory, and moved the Cartesian sliders accordingly. Likewise, this timer updated the XYZ sliders according to what was specified in memory. Next, Rmaster transferred control to Photon for management of the GUI interface. Upon compilation, Rmaster.c (the source file for Rmaster) was dynamically linked to Rhinocb.c. Rhinocb.c contained the callback functions as listed by callback.h.

The layout of the GUI is shown below in Figure 6.

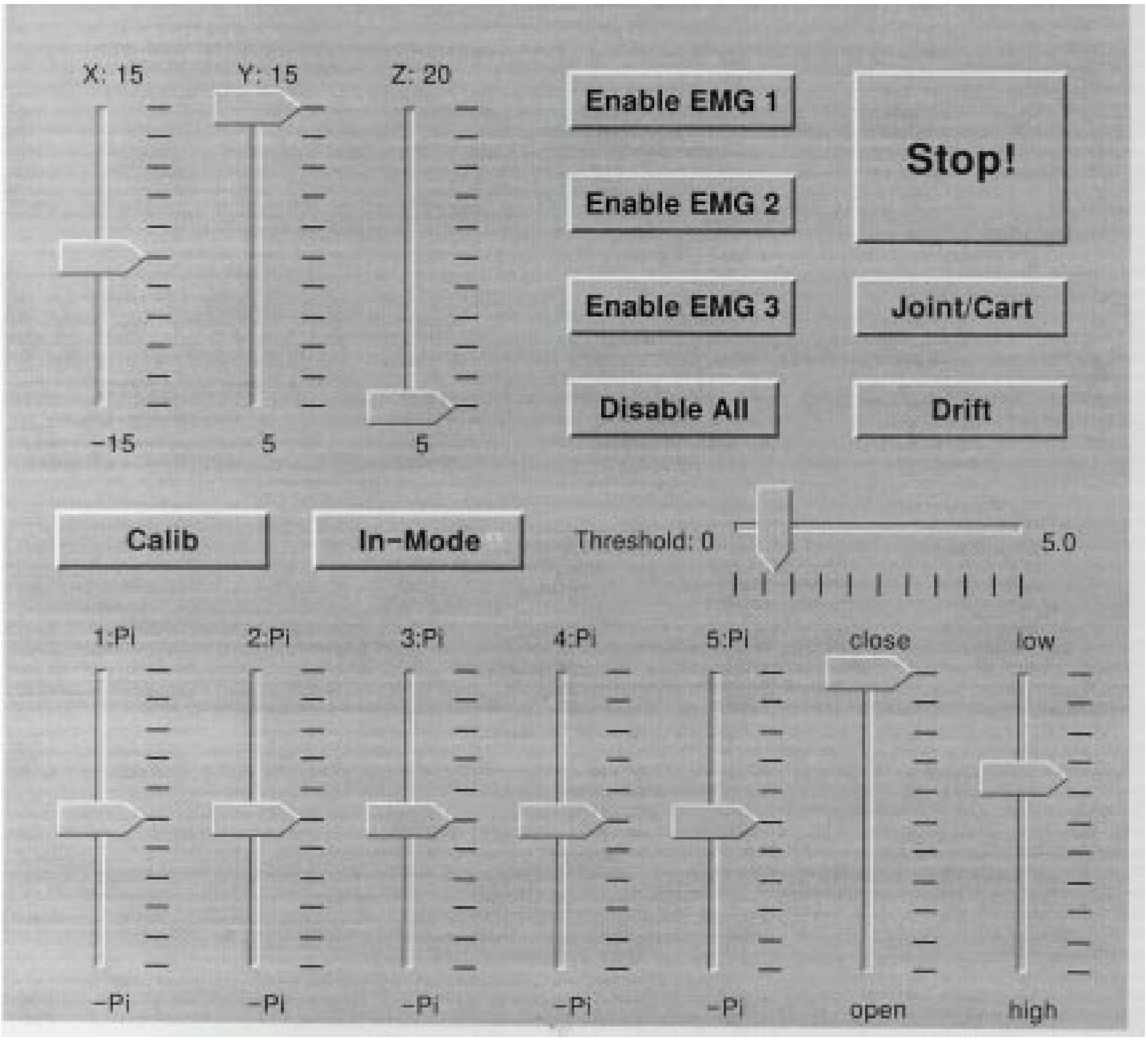


Figure 42 - GUI Layout

The following buttons performed the corresponding functions according to the callback procedures found in Rhinocb.c:

Stop button - This button caused the robot to return to the starting position. Joints one, two, and three returned to the position where their angles were zero. After ample time for this position to be reached had elapsed, the signal was sent for all other process to cease execution.

Default: It was the initially active button.

Joint/Cart button - This button acted as a toggle switch between joint control and Cartesian axis control. When it was first pressed, it switched to XYZ mode displaying “XYZ” on the button itself. With the next press, it changed to joint mode and the button displayed "Joint".

Default: Joint mode until the calibration was complete. Cartesian mode thereafter.

Home button - This button was disabled because the home routine Tplan was not running correctly due to the disabled joints and switches. When Rmaster was originally written, this button was used to recalibrate the robot if it was lost.

Default: none.

In-Mode button - This button cycled through the different analysis methods. The active analysis method title was displayed on the button to give the proctor feedback.

Default: Analysis method was specified during the startup script in Rmaster before the GUI was activated.

Drift button - Activated drift correction on the robot by toggling between drift on and drift off. Drift correction caused the robot to move slowly back to the center of its 3D space, for the ease of the user and the safety of the robot staying within safe bounds.

Default: Drift was off.

Calib button – This button was used to recalibrate the EMG signal range.

Default: none.

Enable EMG 1 - Enabled signal from EMG 1 to affect robot movement (toggled between enabled and disabled).

Default: none.

Enable EMG 2 - Enabled signal from EMG 2 to affect robot movement (toggled between enabled and disabled).

Default: none.

Enable EMG 3 - Enabled signal from EMG 3 to affect robot movement (toggled between enabled and disabled).

Default: none.

Disable All - Disabled all of the input signals when pushed.

Default: none.

The following slider bars performed the corresponding functions:

X slider - Directed control of the X position (enabled when joint/cart button was in XYZ mode).

Default: 0

Y slider - Directed control of the Y position (enabled when joint/cart button was in XYZ mode).

Default: 15

Z slider - Directed control of the Z position (enabled when joint/cart button was in XYZ mode).

Default: 5

Joint 1 slider - Directed control of the angle for joint 1 (enabled when joint/cart button was in joint mode).

Default: 0

Joint 2 slider - Directed control of the angle for joint 2 (enabled when joint/cart button was in joint mode).

Default: 0

Joint 3 slider - Directed control of the angle for joint 3 (enabled when joint/cart button was in joint mode).

Default: 0

Joint 4 slider - Directed control of the angle for joint 4 (enabled when joint/cart button was in joint mode).

Default: 0

Joint 5 slider - Directed control of the angle for joint 5 (enabled when joint/cart button was in joint mode).

Default: 0

Activation threshold slider - Controlled the size of activation threshold distinguishing a relaxed EMG position from an excited one.

Default: 0.5

EMG scale slider - Controlled how much the EMG signals affected the robot movement.

Default: 5.0

4.2.2 – The Trajectory Planner Process

Running with a priority of 12 at 10ms, this process simply cycled through a while loop that evaluated whether the parent process, Rmaster, had killed the semaphore. When the semaphore was dead, the trajectory planner understood this as a signal to also cease execution. In the main while loop, one of four function calls were followed depending on which mode was selected. If the mode was joint mode, tplan calculated the forward kinematics and placed the newly calculated values of Cartesian points into memory. If in Cartesian mode, the inverse kinematics were calculated and the desired joint angles were updated by writing the values into memory. Home mode, although never used anymore, triggers the homing routine. If none of these modes was specified by the global variable MODE, nothing happened. Each time through the loop, the time was recorded to allow for the PD process to calculate velocities.

4.2.3 – The PD Control Process

With an execution time of 1ms, this process ran continuously as fast as possible. It looped, waiting for the Rmaster semaphore to die. The first job of PID was to read in data from the IO board. Next it interpolated the correct angle set points based on current velocity and angle. The required voltages were calculated and sent back to the board as output.

When the data was read in from the I/O board, it immediately updated the actual angles and velocities in shared memory. If this was the first time the process was run, the angles and velocities were initiated, and the time was reset. It regularly determined the values coming in from the EMG filters as voltages. Next, it calculated the elapsed time and change in angles. If enough time had elapsed, the actual velocities were extrapolated, and the time was recorded for the next time interval, thus preventing the robot from moving too quickly. The next point was calculated with linear interpolation based on angle set-points between inverse kinematics computations. Finally, voltage outputs were set on the I/O board to the desired levels.

4.2.4 – The EMGdata Process

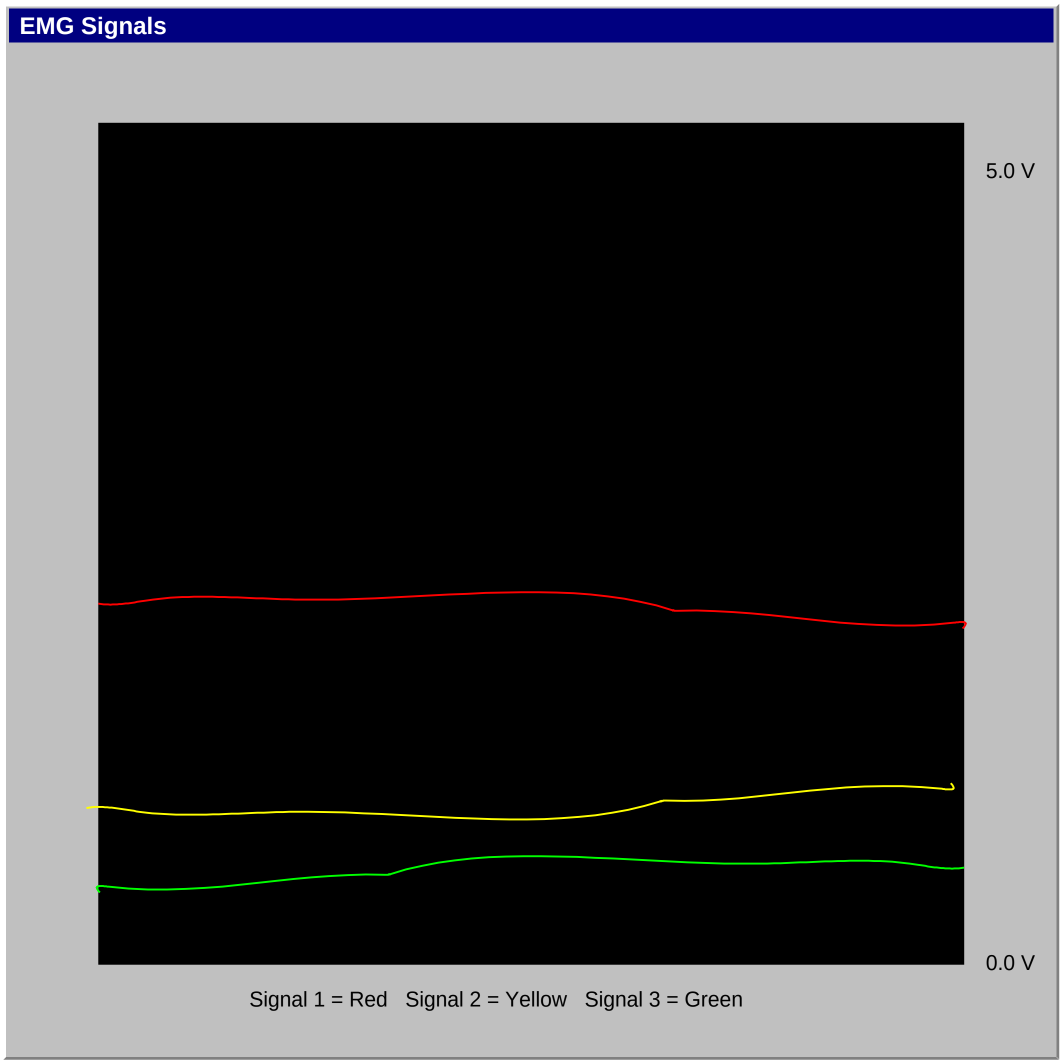


Figure 43 - EMG Signal Display

The EMGdata process was spawned by Rmaster. Like all processes spawned in this system, it accepted the process name, its semaphore number, the priority of the process, and the execution period. EMGdata ran with a priority of 12 at an execution period of 100ms. This process served two main purposes. First, it acquired the data from the EMG signals, averaged it, and continuously displayed these values graphically on the screen (like an oscilloscope). Second, this process performed the transition from EMG signals to robot motion commands, per the selected control mode. The control mode employed depended on which method via the GUI. How the command interpretation was executed will be discussed in the next section. This process also checked resulting position commands and verified that they were within safe bounds.

Other maintenance work performed by this process included implementing the drift correction function. If the drift button was set to enable drift, the robot slowly returned to the center of its allowed space.

4.2.5 – The Calibration Process

The calibration process was also spawned from Rmaster. Rmaster specified a priority of 17 and an execution period of 100ms. This simple process waited for a key to be pressed before it took each of its six readings. First it read in the minimums of each signal when the user was in a relaxed state. When a key was pressed, it took 25 readings, each separated by 100ms. The calibration process then determined the minimum value and wrote it into global memory. Next it read the maximum signal. The user was instructed to first squint hard with the right eye. A key was pressed and the process chose the maximum value of 25 100ms readings. The process was repeated to obtain maximum left eye and forehead readings. Once the maximum and minimum values for all three signals were known, the range between each was calculated so analysis could be made based on a percentage of the range according to his or her maximum and minimum values. Before exiting, it quickly displayed the determined values for the individual and set the mode to Cartesian mode. Upon changing the mode, control was transfered to the benchmarking process to begin an experiment.

4.2.6 – The Position Recording Process

This simple process wrote to an output file the current time and current position of the robot. It ran with a low priority of 15 and an execution period of 50ms. Recorder ceased execution when Rmaster killed the semaphore once the stop button was pressed.

4.2.7 – The Benchmarking Process

This process guided the user through the testing protocol. It continuously looped through the main procedure as it gained access to the global memory with the semaphore key. As it looped, it checked for the signal from Rmaster to cease execution. First the control was suspended until calibration was complete. The proctor was given the option to enter a 100 second warm-up by pressing the 'y' key at the console. If warm-up was selected, the program displayed a count down while the user practiced the method to be used for the test. Originally there were several tests, but over time two tests were left. The results of these tests were recorded in the test data directory under the file name that depicted the tester's name, test number, and analysis type.

The first of these dexterity tests simulated dialing a telephone on an enlarged touch-tone keypad. The user was given a predetermined seven digit random number that he or she was expected to dial. Once the volunteer was ready to begin the test, the proctor pressed a key and the program, by default, brought the robot up away from the table to allow for the placement of the numbered pad. The test began when the proctor pressed a key and the first number was displayed on the console. The subject was told the number verbally and they were to bring the pointer on the robot down to the targeted number.

Since the robot was controlled in software, its position could be determined in software by viewing the global memory values for position. The computer beeped when a certain elevation threshold was reached, signaling to the user that they had pressed a button. Originally, this test would have been restarted every time a mistake was made, because a real phone dialer must hang up and try again. This was frustrating for novice users, and many tests were not completed because the subjects were unable to pass this test. The output file recorded a success or a failure each time the pointer was brought down onto the numbered paper.

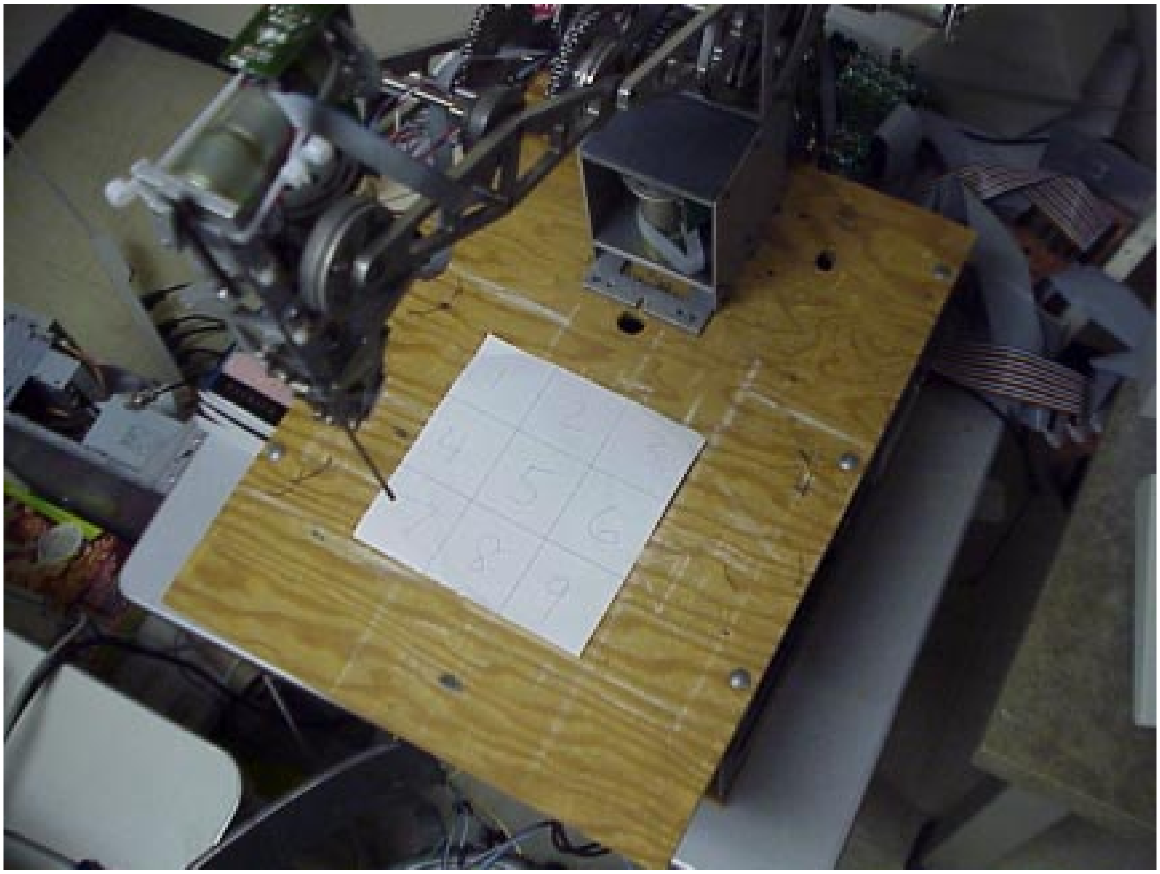


Figure 44 - Dial Test

The second test required the user to weave figure eights around two plastic cups that are placed on the table. The robot was positioned in the upper left corner when a button was pressed. A few seconds of time elapsed before the user was allowed to continue on with the test so the robot could successfully reach its starting destination. The next key press initiated the test as the user was permitted to begin weaving around the cups. If a cup was struck, the computer beeped, informing the user that he would have to begin the test again. Failure or success was recorded in the output file. When this test completed successfully, the process exited cleanly.

Figure 45 - Trace Test

Two additional tests that were proposed but not conducted were axis extremity and point stabilization. The preliminary experiments showed that the proposed tests would consume large amounts of time, and thus the number of test was reducd. The extremity test required that the user move the pointer of the robot from the center of space to an extreme point and back to the origin. All 6 directions would be tested to ensure the user would have the ability to pass the more complicated tests. This test was subsumed by the dialing and figure-eight test.

The other discarded test required that the user travel between three points. At each point, the user concentrated on holding the pointer still for 10 seconds before racing to the next point. This test was considered valuable because it would show how stable the user could keep the position of the robot. Since ultimately all methods tested could have the robot stationary in a relaxed state, point stabilization was not worth measuring.

5.0 EMG Signal Modes

The software remained relatively unchanged as the testing progressed. New EMG control methods were developed over time, but initially, velocity mapped methods were used to update each axis based its corresponding EMG signal input. Methods that utilized memory hysteresis were developed next to increase the dynamic range of velocity mapping to encourage velocity control. The final control method merged the benefits from the previous methods tested by hysteresis (mode switching) in order to control the active axis from one of the inputs. A velocity map of the remaining two inputs signals controlled the movement of the active axis. The order of the following control methods reflects the order of their creation and explains the reason for the creation of each mode.

5.1 Velocity Mapped EMG Control Methods

The benefit to velocity control is an increased smoothness in motion and added safety. For each signal received, the computer immediately determined a specific velocity from a designated function within the process EMGdata.

5.1.1 Velocity Map 1

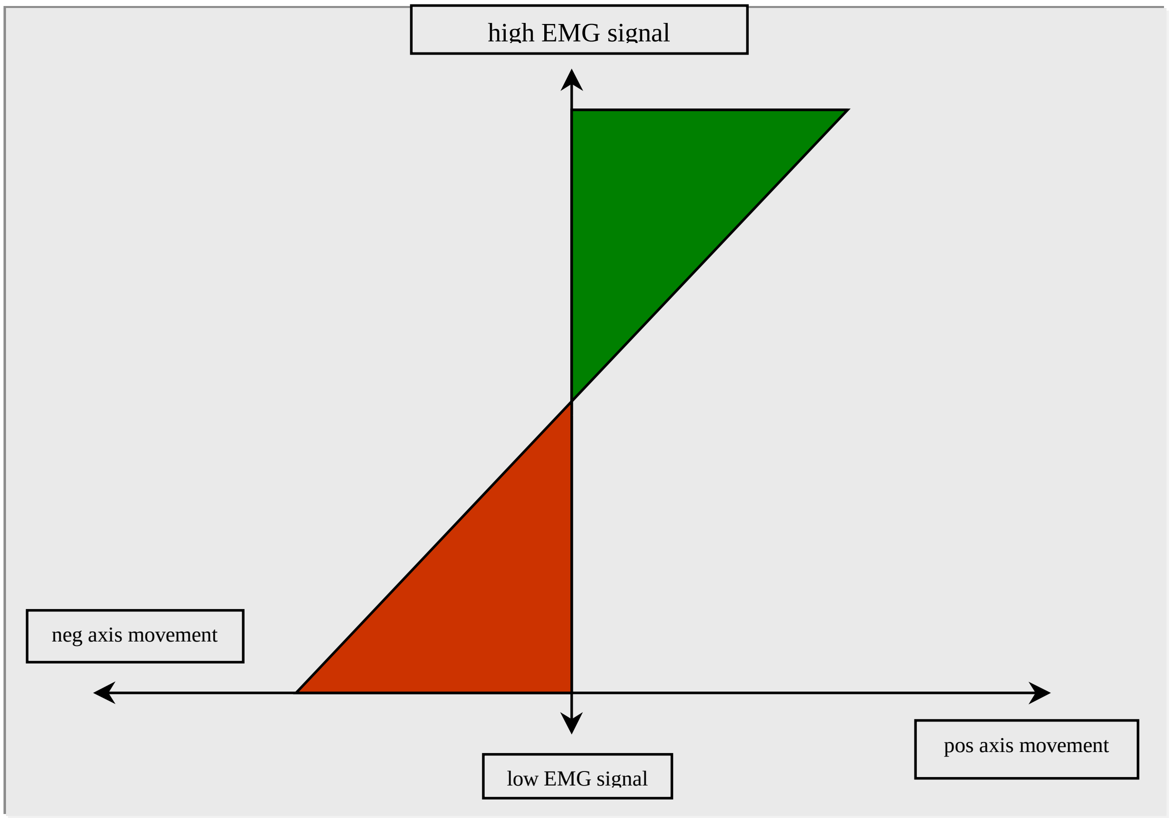


Figure 46 - Velocity Map 1

The first method, named offset proportional, reflected the simplest way to program the computer in order to incorporate velocity control with all three axes. Each signal was used to control both positive and negative movements. A relaxed state moved the arm in a negative direction and a fully excited state moved the arm in the positive direction.

This method examined each signal, one at a time. If the corresponding button was enabled for the signal, the mapping function was permitted to cause the robot to move. EMG signal value was first offset by a constant corresponding to the middle of the EMG range. The result was scaled to a reasonable value by multiplying a constant scale converter. The value was then added to the previous position before being written into the shared memory as the new target position. Since updates were computed at a fixed rate, this had the effect of velocity control.

The only adjustment that could be made was to scale the responsiveness of the robot to the input EMG signals. A graphical slider was implemented to adjust the equivalent velocity scale factor.

This mode was very unstable since it was nearly impossible to keep the robot stationary, which required keeping all three inputs effortlessly at exactly half of the EMG range. For any mode, achieving zero motion control effortlessly was essential. A threshold zone or dead band for future methods would provide a relaxed state.

5.1.2 Velocity Map 2 (Scaled Velocity Mode)

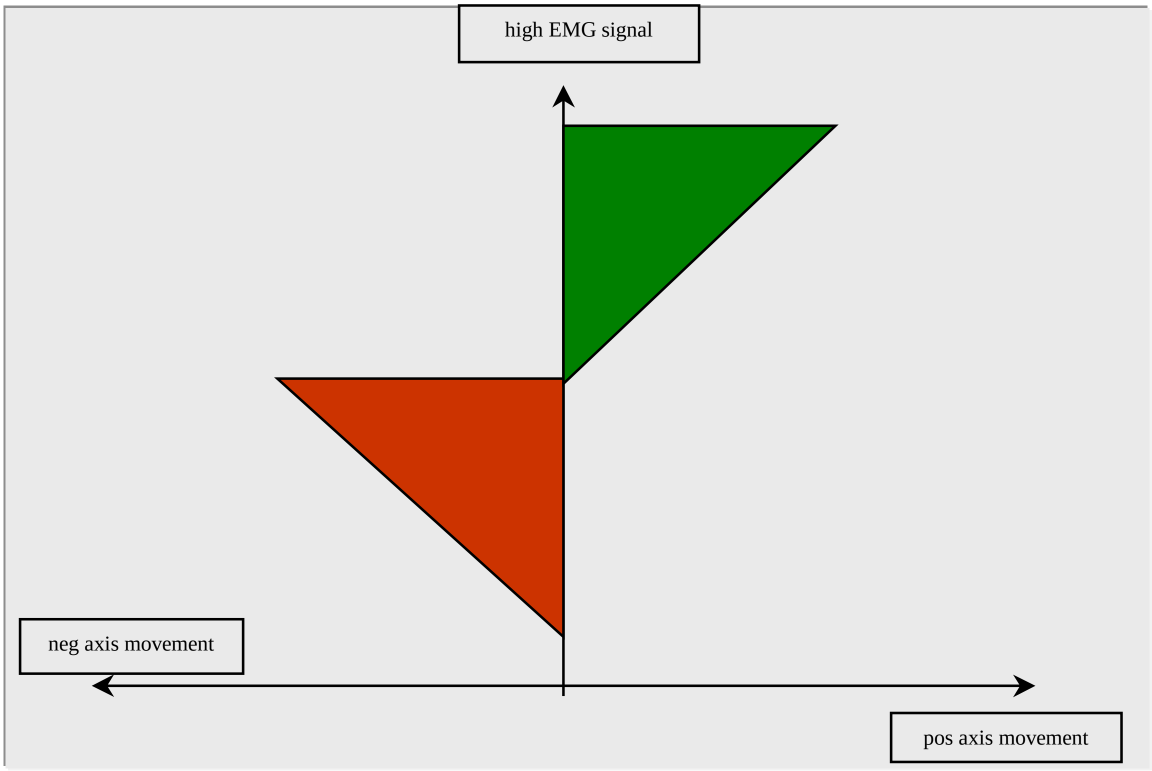


Figure 47 - Velocity Map 2 (Scale Velocity Mode)

The second control mode, “velocity map 2”, improved upon the previous mode by introducing an activation threshold. The robot position remained unchanged if the signal strength remained below the desired threshold, as specified by a slider on the GUI. If, however, the signal strength exceeded this threshold, then all signals less than the middle of the EMG range moved the robot in a negative direction according to its magnitude. For a signal greater than the middle position the robot was moved in a positive direction. A scale-factor slider controlled the sensitivity of the robot velocity command vs. EMG signal.

This method increased the stability of robot motion because it gave the user the ability to rest physically, and the luxury of little concentration in order to stay stationary. Fatigue was also an important factor in these evaluations. EMG’s do not detect the intent of the user or how hard somebody is trying to flex a muscle. Fatigue reduces the EMG signal in spide of intensity of intent. Once the user has rested, signal strength is recovered.

Since this method was successful it was named ‘scaled velocity control’ and used in the formal testing.

5.1.3 Velocity Map 3

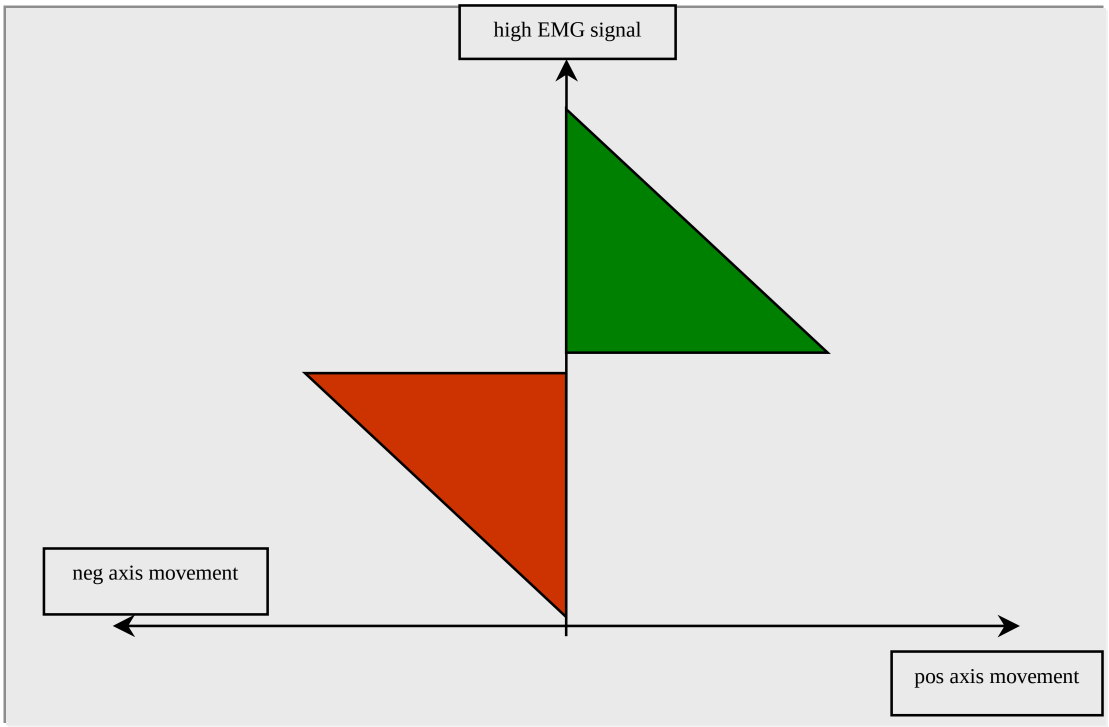


Figure 48 - Velocity Map 3

A fallback to the previous method occurred when the user attempted to move the robot in the positive direction at a high speed. To reach this voltage, the user had to produce a strong signal and sustain it for an extended period of time. When the muscle was first stimulated, the type of signal used to send information to and along the output segments of the cell, often over long distances, is called the action potential[3, pg.145]. Eckert's Fourth Edition Animal Physiology textbook explains

> "Action potentials are large, brief changes in Vm (membrane voltage) that are propagated along axons without decrement"[3, pg.145]

Since the intent of the strong squint was to produce a high voltage from the amplifier, these action potentials caused the neural transmitters to spike the EMG signal voltage.

Unfortunately, over a short period of time, the signal began to declined due to what we nicknamed fatigue. This made maintaining a maximum voltage even more difficult. We leaned from this physiology text book that this was to be expected.

> “If a neuron is stimulated by a series of subthreshold depolarizations, a time-dependent decrease in excitability occurs.”(146)

To overcome this problem, the gain was increased to roof the spike from the signal at the filter output limit of 5 volts.

Since the middle emg range maps the largest movements, the method was named center biased scaled velocity control mode.

The cost to increasing the gain, was a loss in the range of detection. This made it more difficult to move the robot in the negative direction because there is less dynamic range than before. This analysis method should have avoided this problem by inverting the responsiveness of the positive movement. Now, a signal just above half way mapped the maximum movement and the largest signal no movement. The user can give a hard flex and steadily release tension, or simply wait for their cell membrane’s to depolarize as the signal begins to fall It should also be noted that humans can produce steady nonspiking signals. This steady voltage signal is what we have been successfully detecting and filtering.

> "Some neurons, in both vertebrates and invertebrates have been shown to release neurotransmitter from their terminals even in the absence of APs, which is called nonspiking release. At least some of these neurons are unusual in that they may never carry AP's; all information transfer is accomplished by electrotonically conducted voltage signals." ... "when the cells are more strongly depolarized,

they release more transmitter; when they are less strongly depolarized, less transmitter is released. As a result, the amount of transmitter released is a direct function of the membrane potential, Vm, of the neurons, which can indicate how strongly the neurons themselves were stimulated." (194)

5.1.4 Velocity Map 4 (Jog Mode)

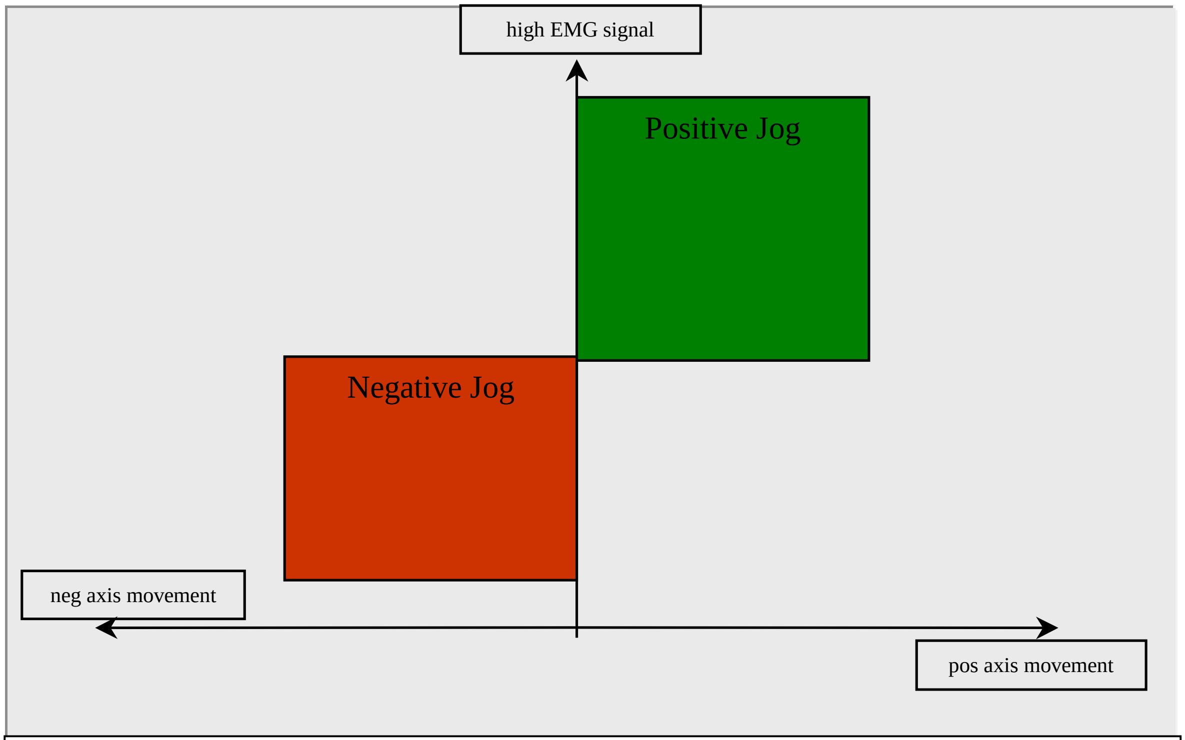


Figure 49 - Velocity Map 4 (Jog Mode)

A jog mode velocity map was tested to get a realistic comparison of continuous velocity control versus on/off control of a constant velocity. Like the previous velocity mode, the jog method commands positive and negative velocities based on the EMG intensity. However the jog-mode velocity can only be on of three values – a fixed positive velocity, a fixed negative velocity or zero. The negative zone existed between an activation threshold and the middle of the EMG range. All signals above the threshold caused the robotic arm to move with a constant velocity in the positive direction. Originally, there was a space between positive and negative zones aimed at preventing the user from producing a signal in the opposite direction of intention. This proved to be counter productive. Preliminary subjects did not have a problem keeping between positive and negative zones and more difficulty reaching either. This dead band only caused confusion since there was no movement with an intended muscle stimulus.

5.2 Mode-Switching (Hysteretic) Controllers

Another category of controllers considered involved the use of memory. In order to have more effective command of velocity, this method utilized the entire dynamic range of the EMG's for both positive and negative velocity commands. This requires a hysteric function to switch between positive and negative modes. When a signal reaches a smaller specified voltage range, the polarity switches. This leaves a larger dynamic range for velocity control.

5.2.1 Hysteresis Control Mode One (Single Zone)

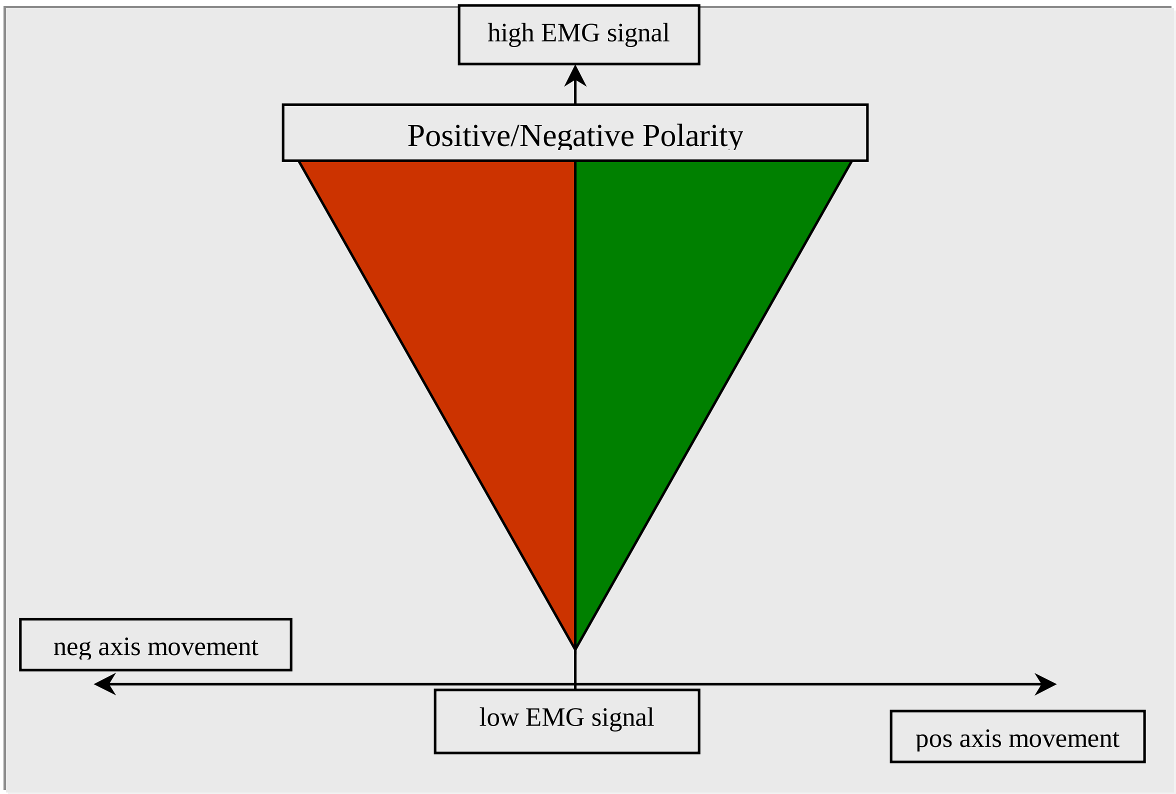


Figure 50 – Single Zone Hysteresis Control Mode

This mode requires that any axis that is to be controlled is enabled. First, the polarity is determined from the signal produced by the subject. If the voltage zone is from maximum reading to maximum minus a MAX_POLARITY_THRESHOLD constant, the polarity is reversed. Values greater than the activation threshold and less than the polarity switching zone cause the robot to move proportional to signal strength in the direction specified by the current polarity proportional to signal strength.

This method was designed to utilize the spike detected by the activation potential. The upper range should have been easy to reach with an AP, creating a large dynamic range with which to more accurately control velocity in the specified direction.

A problem often found with this method was its inconsistent orientation. The user did not know which mode they were in or had a hard time switching modes. Orientation problems were more common when the user was at an axis limit. They felt that something was broken because there was not response with muscle stimulus. In order to move back to the center, the user should have produced a very strong signal to switch modes. Then with a weaker signal, the robot would begin movement back to the middle. The user also had trouble switching modes because they found it difficult to keep a strong signal in the polarity switching zone long enough for the software to detect and register a polarity switch. Simply taking more frequent samples could solve this problem.

5.2.2 Hysteresis Control Mode Two (Dual Zone)

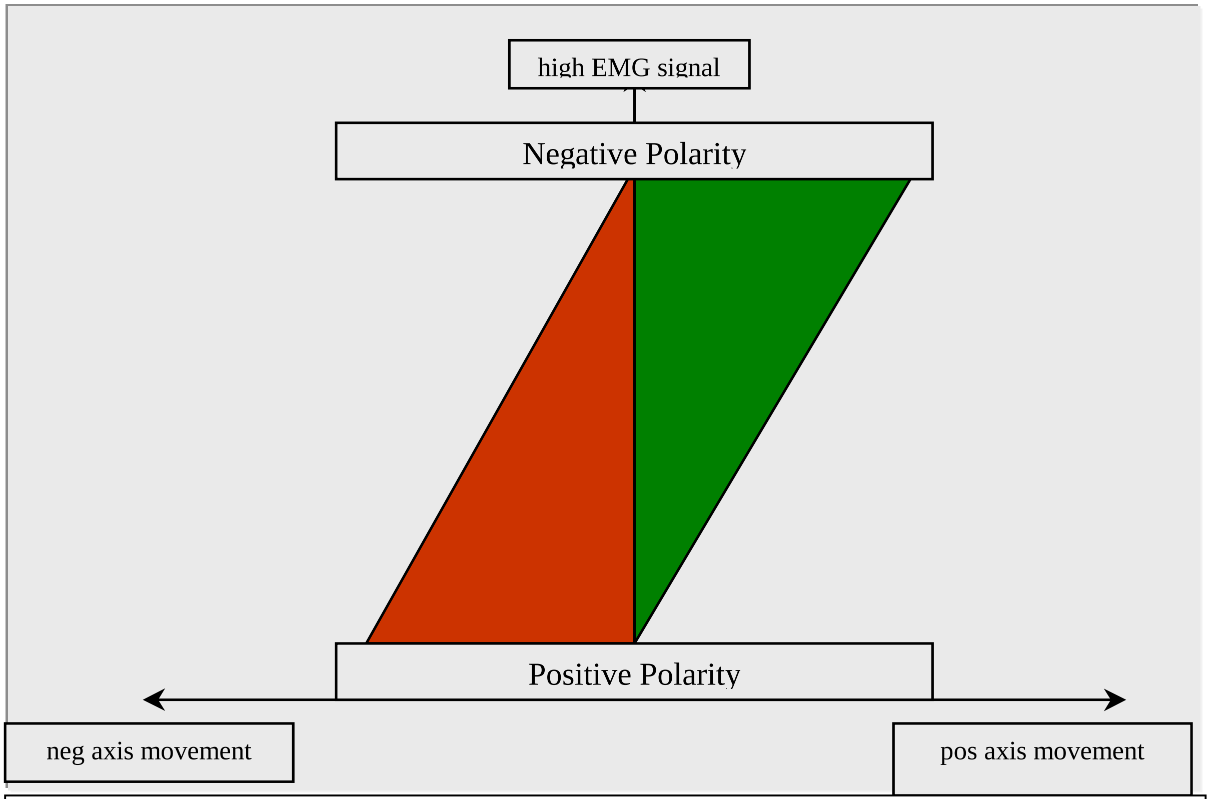


Figure 51 - Dual Zone Hysteresis Control Mode

The second switching mode method improved on the first because it gave the user a default orientation for polarity. The same dynamic range was used, but there were two differences. First, negative velocity commands were inversely proportional to signal strength. Secondly, the polarity-switching zone from the previous method was used to only activate the negative velocity mode. As the user brings down the signal, the negative velocity command increases in magnitude until the signal voltage falls below the activation threshold. For all positive movements, the user had to first relax, activating a positive polarity. As signal strength increased, so did the positive velocity until the polarity threshold was reached. Like all other methods, the EMG scale slider controlled the responsiveness of slope of robot movement over time.

It was very intuitive for eyebrows to control the direction of movement in the z direction. When the user wished to move the arm up vertically along the positive z-axis, he or she raised their eyebrows. To bring it down, the user visualized first going up first in order to push or pull the arm down when they brought their eyebrows down by relaxing those muscles.

5.3 Axis Selection Method (Axis Select)

Experimentation showed that both velocity mapping methods and hysteresis methods had faults. The axis selection method described here was a hybrid method that took the best from both from the velocity mapping and hysteresis mode switching methods.

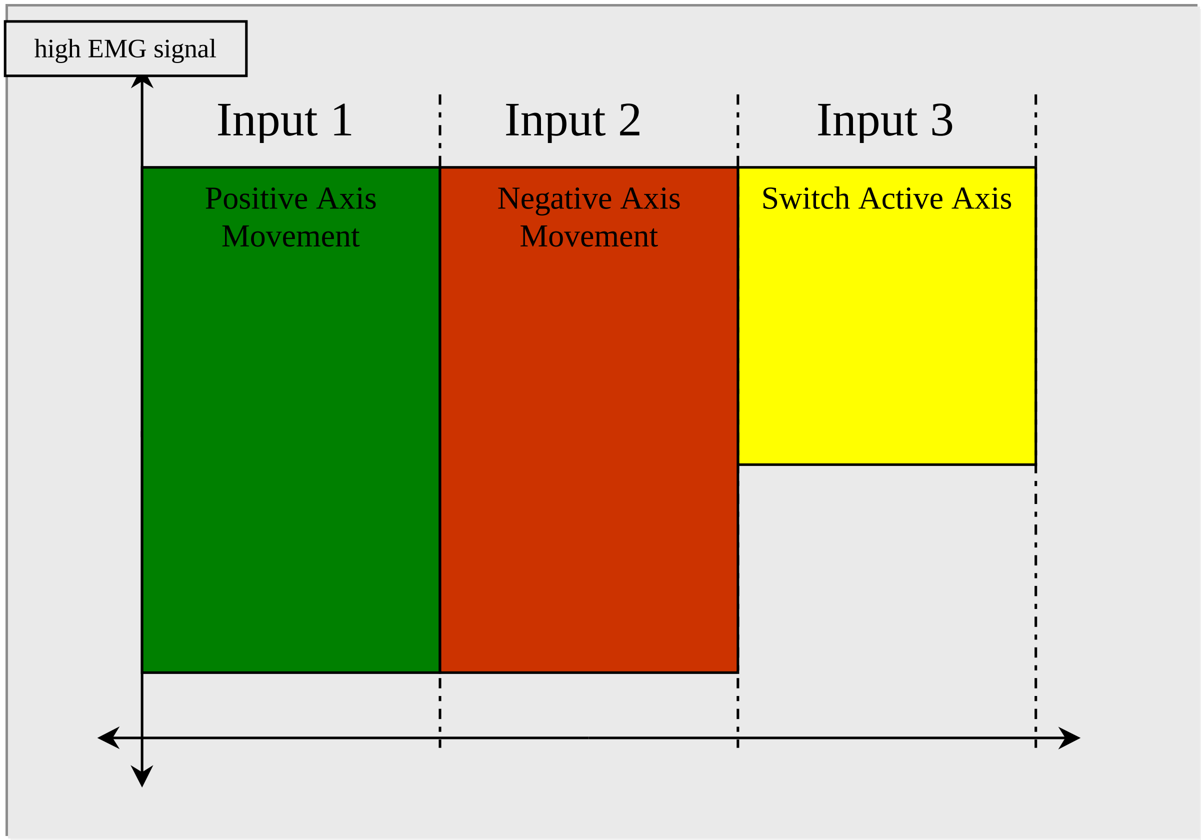


Figure 52 - Axis Selection Method (Axis Select)

The forehead was used to select which axis is to be controlled, one axis at a time. With each forehead reading above the middle point of the range, the active axis cycled through X, Y, and Z then switched back to X. The right eye moved the robot in the positive direction, and the left eye moved the robot in the negative direction. The user was given feedback by the computer beeping to signal an axis change. Further, the axis label displayed on the InMode button of the Rmaster GUI changed when the selected axis was changed.

In order to maintain a position, preventing any movement, the user simply remained in a relaxed state. Once a signal from an eye exceeded the activation threshold, the robot began to move at a fixed velocity. If both eye signals were above this threshold, the movements canceled each other out. This was useful if the subject had trouble isolating the forehead. By tensing all of his or her face, he may have more easily switched the activation axis and would not be as concerned with effect on movement since both eye signals excited in tandem resulted in no movement. The velocities were mapped to a constant multiplied by the scaling multiplier controlled by the EMG scale slider.

6 - Experimental Setup and Results

6.1 Preliminary Testing

The preliminary testing was used to test and develop new analysis methods. This testing helped to determine which methods were easy to use and which ones should be left out of the formal experiments. Each method developed was first tested using a simple variable DC voltage power supply because the voltage read by the computer could be set using the dial. This would have been be much more difficult to test by directly controlling with an EMG signal detected by muscle movement.

The first method evaluated was velocity map 1 (offset proportional mode) and it was nearly impossible to keep stationary. Simply moving the robot to different locations in space was somewhat reasonable, but it was obvious that this method was not worth testing. Our initial impressions from the preliminary testing favored the velocity map 2 (scaled velocity control mode) because it was much more intuitive with the increase of speed with an increase of muscle tension.

In order to find a contrast with and without velocity control, velocity map 2, was chosen to be tested with velocity map 4 (jog mode) because jog mode shares similar thresholds, except is maps a constant velocity for the active regions. Two hysteresis modes were also important because of the increased dynamic range, consequently both were chosen to represent this type of method because the differed in orientation.

6.2 Formal Test Setup

6.2.1 Agenda

Largely displayed on the chalkboard of the lab where this experiment was conducted was written the agenda of the protocol for the experiment. It gave a step-by-step description of what was to be expected.

1) watch video
2) wash face
3) configure amplifier
4) view signals
5) practice method for test
6) perform test

6.2.2 Video

The video was created to provide each tester a uniform introduction to the test. Since each volunteer had no experience with this type of experiment, there were many interactions between the volunteer and the proctor. In an attempt to eliminate error resulting from inconsistent coaching, if each subject was exposed to the same information, they would not have an unfair advantage. The short five-minute video began with a warm welcome and thanks for participation. A quick explanation of the practical explanation of this study gave the subjects important awareness of why they were being tested. Next, it showed what the robot would look like and how the electrodes would be placed on their faces. The format of the experiment was explained along with helpful tactics used in prior experiments. It was instructed that they were to focus on producing independent signals while they had a few minutes to view the oscilloscope. Next a description of each test was given before an actual demonstration of the analysis methods to be evaluated.

6.3 Phase 1- Mode Comparison Jog, Scale Velocity, Dual, and Single Hysteresis

The first formal test of 10 people was performed on these moes. To prevent an unfair advantage, it was required that no volunteers perform the test multiple times or have been tested before in a similar way. Each test subject was given the tests in random order to remove learning experience's effect on the data. The figure 17 shows the test subjects' response to which method was preferred for them to perform. It is not surprising that most volunteers chose the method that they performed well.

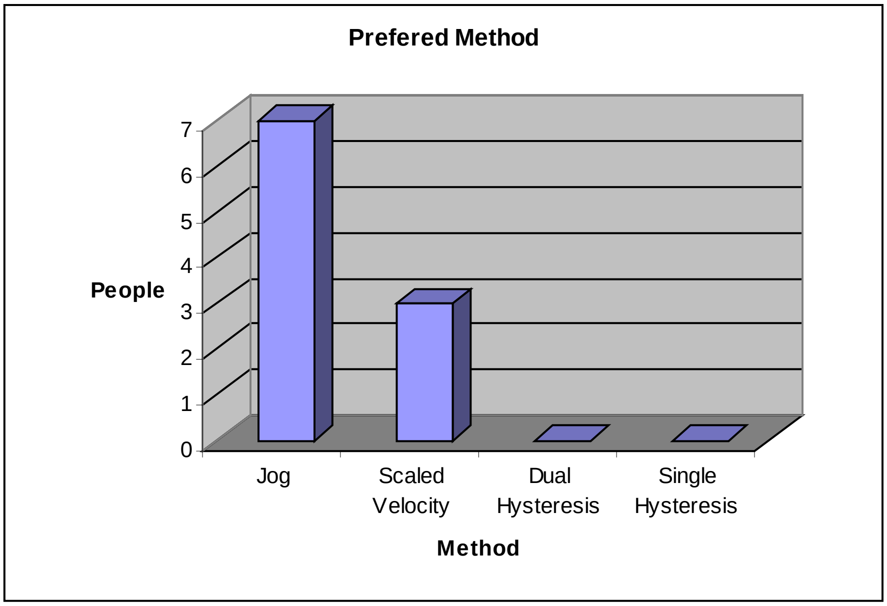


Figure 53 - Phase I Preferred Method

6.3.1 Phase 1 Testing Success & Failure

As discussed before, during preliminary testing, I could easily perform both tests with all analysis methods. To my surprise, when it came time to test others however, many people were unable to pass all of the tests. Figure 18 shows the failure/success rate for the four methods during the dial test.

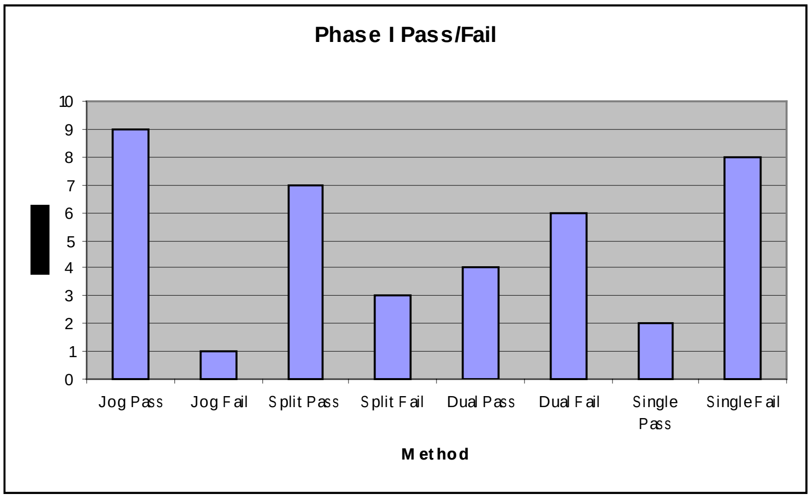


Figure 54 - Phase I Success/Failure Rate

The failure rate would improve with experience, but, unfortunately, test subjects were unwilling to spend substantial time learning the interface. This prevented more conclusive experimentation on long term, high experience test performance.

6.3.2 Dial Test

The first of the two tests, the dial test was successfully completed between 40 and 150 seconds. Here is a graph of how each person performed each test (test failures are omitted as zero time). Test subject number 10 failed all three tests given as shown by the large gap on the right.

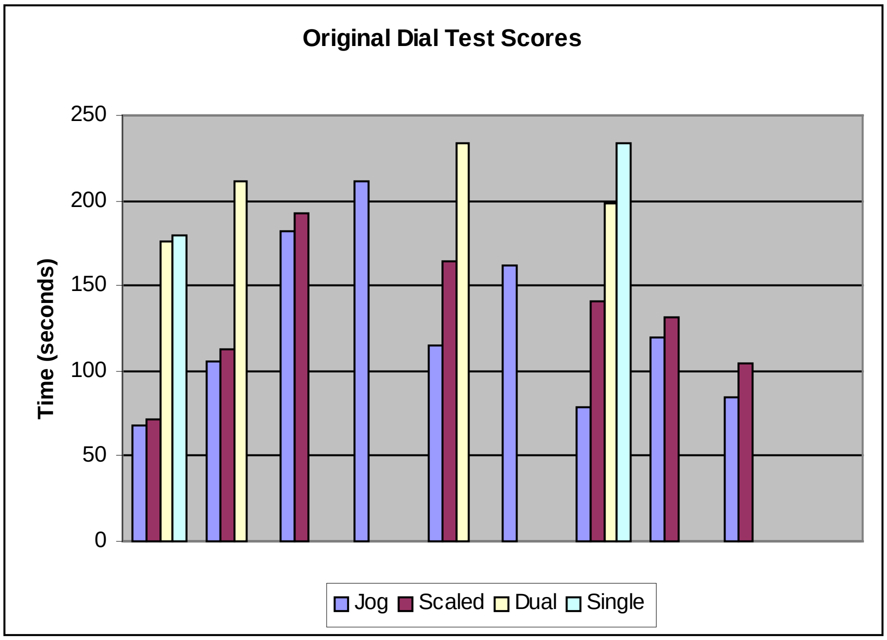


Figure 55 - Phase I Original Dial Test Scores

The average time of each test, shown in the next figure, 20, give a better indication as to how quickly the group as a whole performed the test. The simple jog mode was the fastest of the four methods. Close behind was scaled, the velocity control version of the jog mode. The dual and single hysteresis methods did not perform well on the dial dexterity test. Users had trouble moving the robot in the orientation opposite of the current one that they were using.

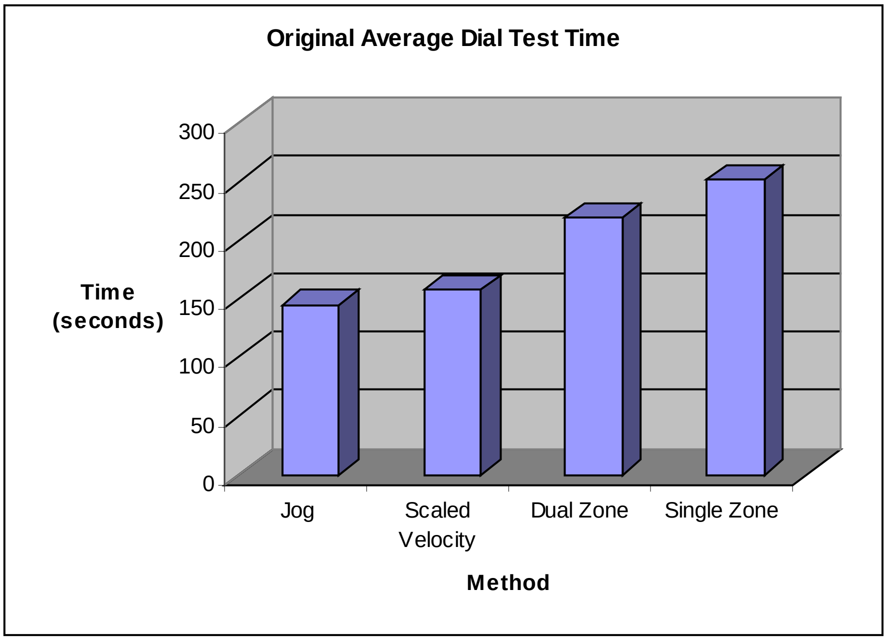


Figure 56 – Phase I Original Average Dial Test Time

6.3.3 Adjusted Dial Test

Next the data needed to justly reflect the erroneous button presses as failures. In real life, most operations performed wrong needed to be corrected. The formula for calculating a time penalty for each mistake reflected being penalized the average amount of time it would take to doing the task over. In this simulation, the penalty is the average time of all users to press the button. Thus the formula used was equal to the original time to complete the task plus the average time to complete each button press. This error penalty is generous considering that when a mistake dialing a phone requires the entire number to be repeated.

Jog Mode	Scaled Velocity	Dual Zone	Single Zone
17.89	18.76	29.25	29.57

Notice in the following figure, 21, the adjusted scores ranged from sixty to almost three hundred seconds. Since the original time of completion for subject seven was 234 seconds, two mistakes caused an adjusted score of 293.

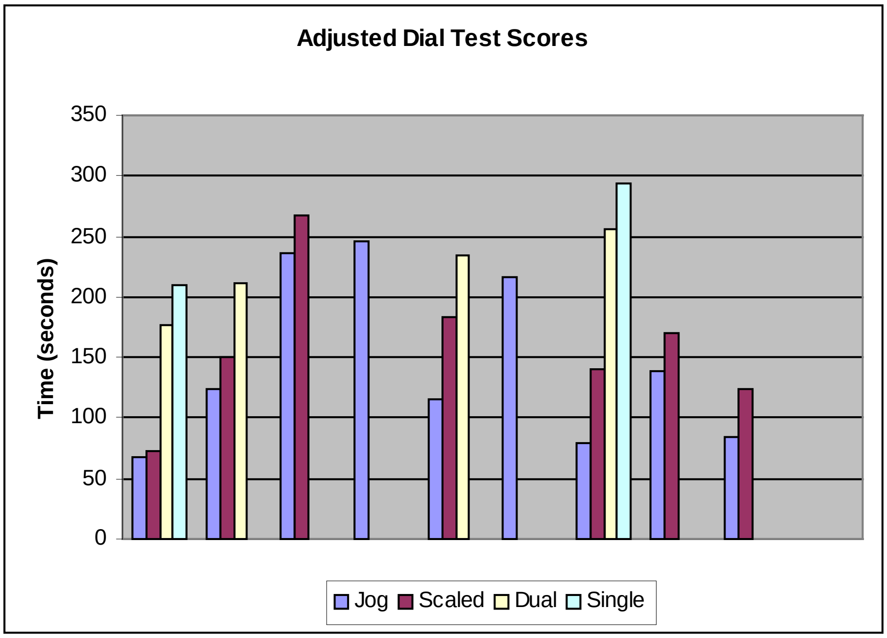


Figure 57 - Phase I Adjusted Dial Test Scores

The next figure shows how the original scores stack up against the adjusted scores. The difference shows the influence of the error in the adjusted scores especially in single zone mode. The jog method showed to achieve the fastest time in the dial test showing that the most simplistic method was the most successful.

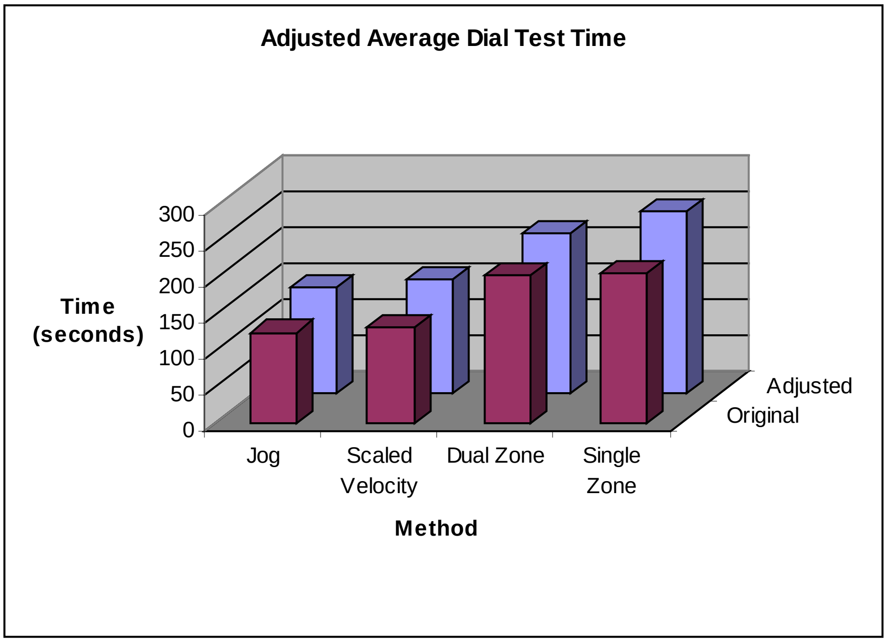


Figure 58 – Phase I Adjusted Average Dial Test Time

The data showed that the first two methods devised were much more effective. These tests did not evaluate any fine movements, which may give a strong advantage to the velocity control enabled methods executed with the same precision. Modes jog and scaled were very similar in nature and their scores represent a similarity. Jog mode was slightly superior in speed because all movement was at a constant speed. With mode B, fast consistent movement was tough to control and hence consistent fast robotic movement was very difficult.

The second two modes were less successful because users had a difficult time controlling direction of movement. The volunteers that performed the last two modes agreed that method dual zone was superior because it had a built in orientation of movement. In the single mode hysteresis, it was difficult to switch polarity of motion and no feedback was given. This difficulty caused the users to be confused quickly, increasing their frustration.

6.3.4 Trace Test

The trace test proved to show very similar results as the dial test. Only those who passed the dial test got the opportunity to go on to the Trace Test. Here in figure 23 are the times, ranging from 38 seconds to 147, for the ten test subjects.

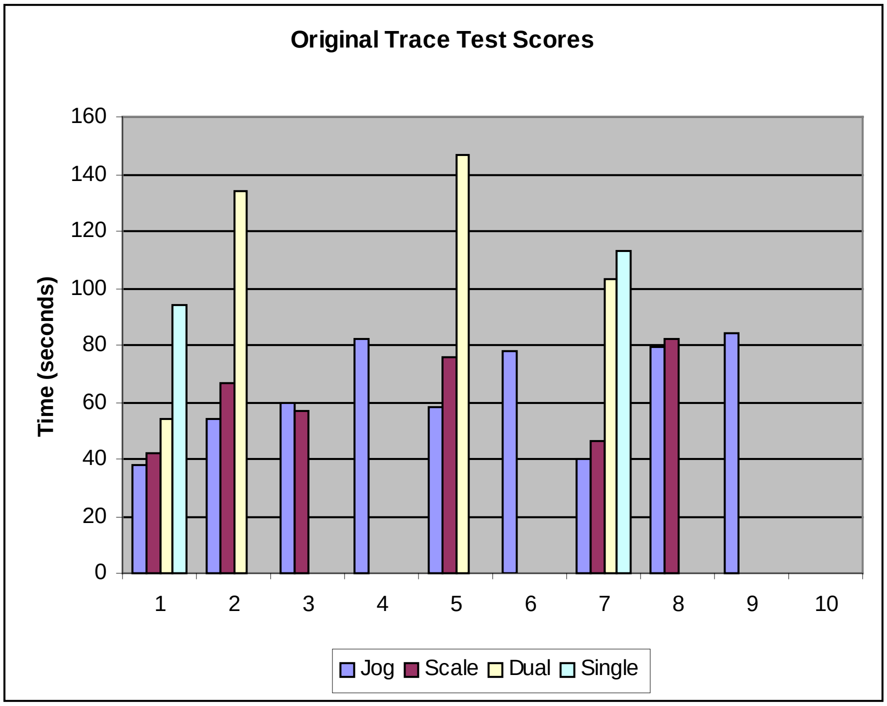


Figure 59 – Phase I Original Trace Test Scores

These times in figure 23 represent a successful run of the trace test, but do not include any information about unsuccessful runs. If the user struck a cup, the test would be restarted. The average values of successful runs are shown in figure 24. Method scaled velocity was completed the quickest, followed by methods jog, single and dual. Test subjects 4, 6, and 9 however heavily influence these averages significantly. They struggled with all 4 methods, successfully finishing the trace test very slowly (82, 78, and 84 seconds respectfully). If you remove these values from the statistics and consider only the tests where both jog and scaled were successfully completed, the jog method proves to be superior. The hysteresis methods again were similar in performance, but far less effective.

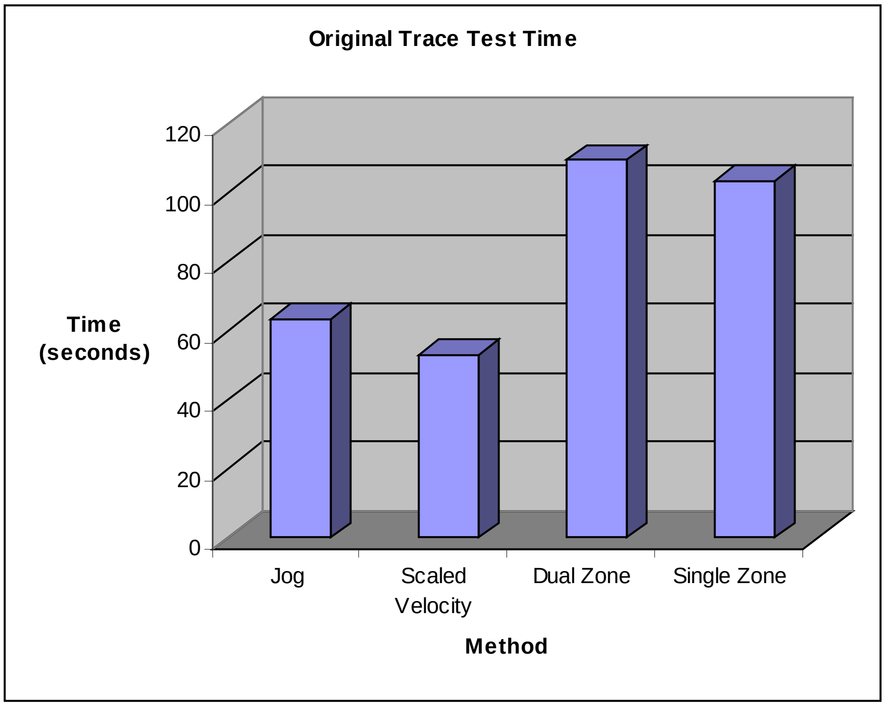


Figure 60 – Phase I Original Trace Test Time

6.3.5 Adjusted Trace Test

The adjusted scores in figure 25 show a more accurate representation of the trace race since it was the actual time required for each person to successfully perform the test. If the user was unsuccessful, time continued once again when they began their retry.

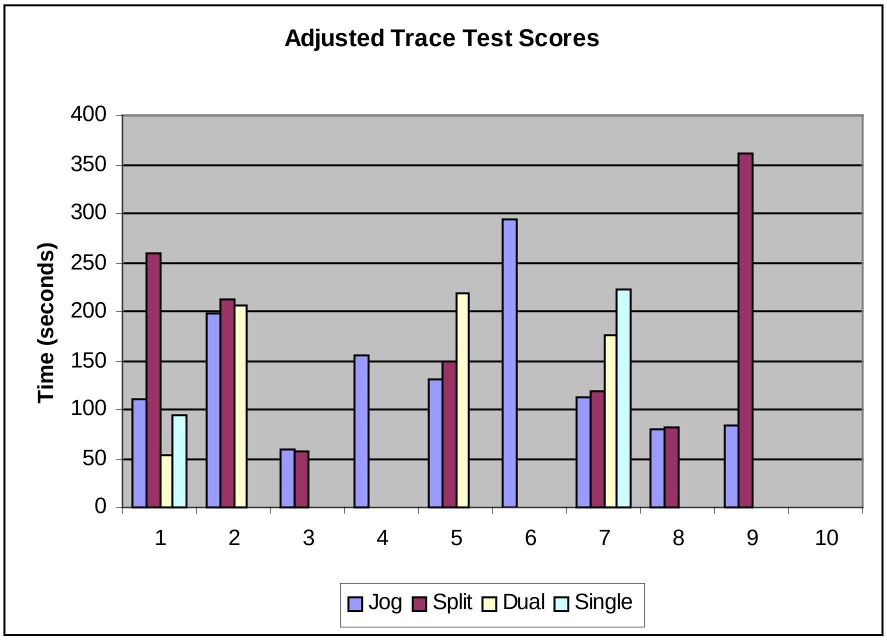


Figure 61 - Phase I Adjusted Trace Test Scores

The next figure shows the average of the adjusted times versus the original trace times. Method A had the best adjusted time followed by D, C, and B. Since the simple it jog mode was the superior method to use in this phase since it had the most successful experiments, it was the most favorable to use, and it scored the best overall times in both tests.

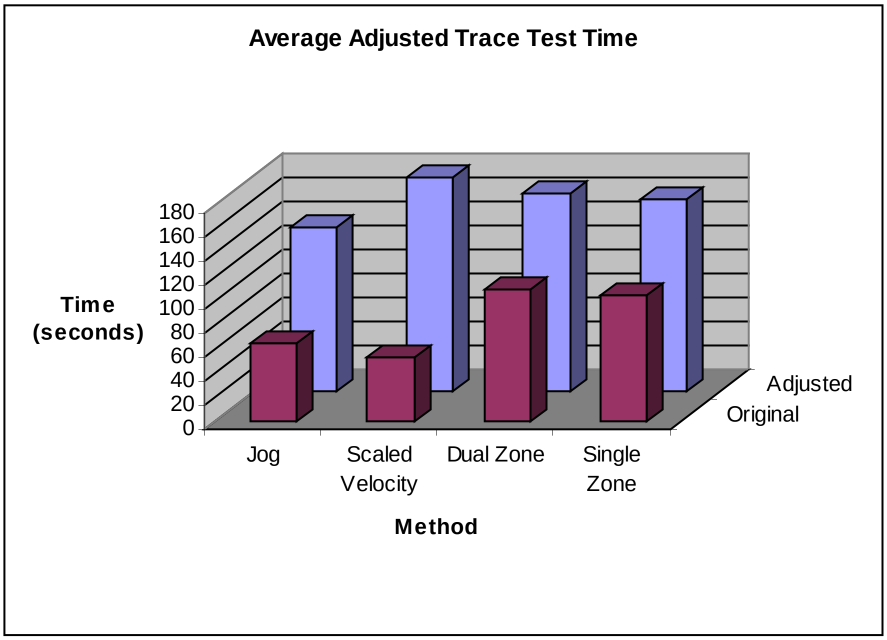


Figure 62 - Phase I Average Adjusted Trace Test Time

The scores for methods D, C, and B in these experiments however did not necessarily reflect the effectiveness of the methods. Those who had good motor skills and or could easily reproduce signals were the only ones who attempted methods C and D. A better way to look at the data is to compare how each volunteer scored progressively on each test in figure. How each subject performed relative to the other methods agrees with what the failure tests show. Methods A and B are superior to C and D. And method C appeared to be slightly more effective than method D both in success rate and dial test times.

In this phase, we sought performance in dexterity with all methods. For the next experiment, the focus will be on evaluating methods moving the position of the robot since the most straightforward method was the most effective.

6.4 Phase 2 - Jog Mode Versus Mode Switching

The next experiment proved to be much more rewarding and all of the tests given were performed successfully. Since the simple jog mode proved to be the superior method, another simple to learn method was tested for comparison.

The new method tested was named axis select, because it used the forehead signal to switch the current active axis. The user would have to deliberately trigger an axis switch giving them direct control along only one axis at a time. The user could use each eye to control a direction of movement, either positive or negative. When asked which method was preferred, the axis selection mode was chosen.

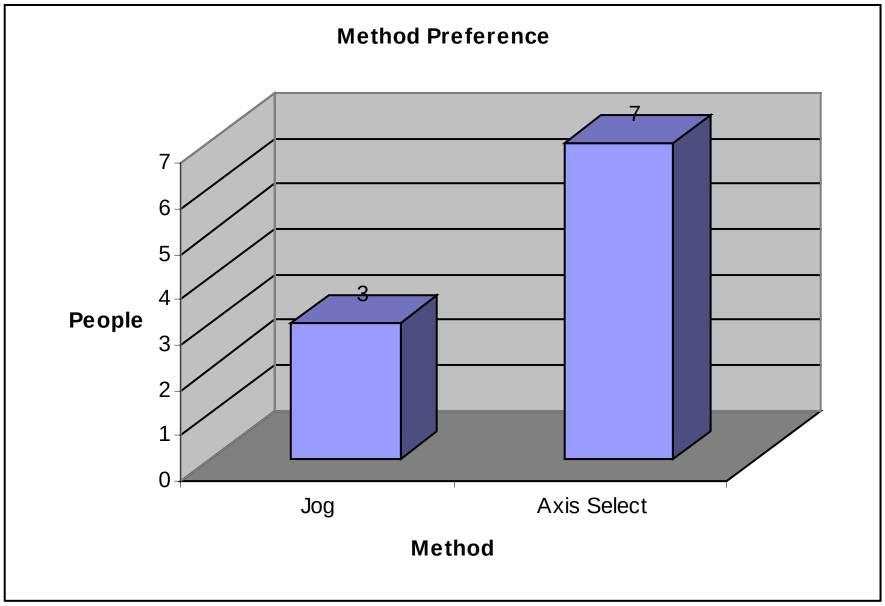


Figure 63 - Phase II Method Preference

6.4.1 Dial

Here are the results from the first test.

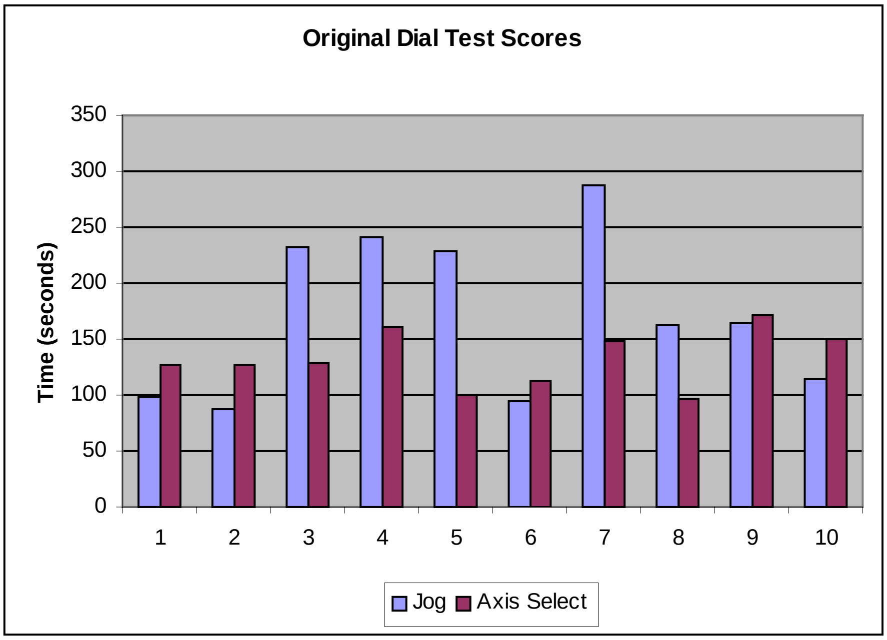


Figure 64 - Phase II Original Dial Test Scores

Half of the people experimented performed better on the jog mode, half on axis selection. The difference in time completion of the test favor axis switching as seen in the next figure of the averages.

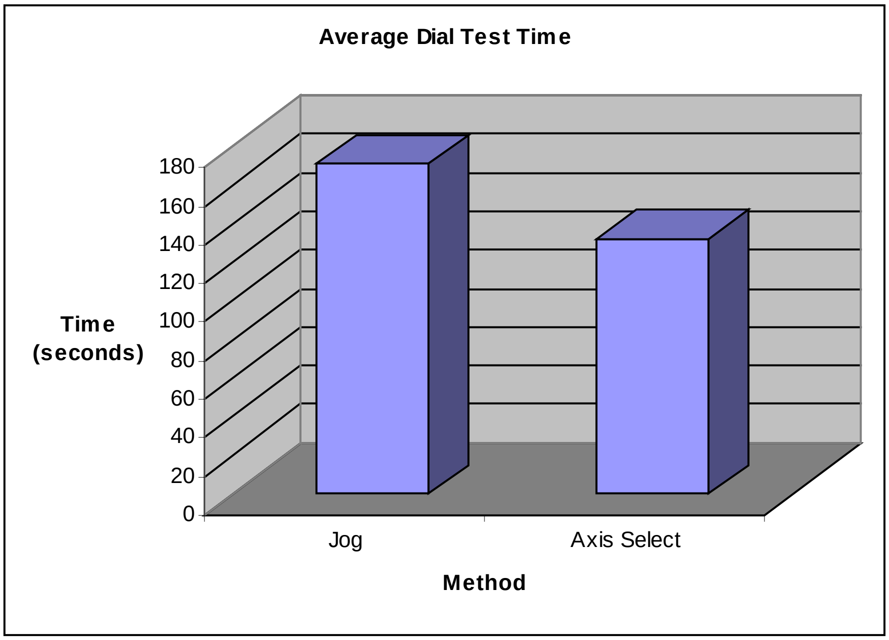


Figure 65 - Phase II Average Dial Test Times

6.4.2 Adjusted Dial

When the error adjustment is applied, the performance gap broadened. The adjusted times show that only two people performed better with the jog mode.

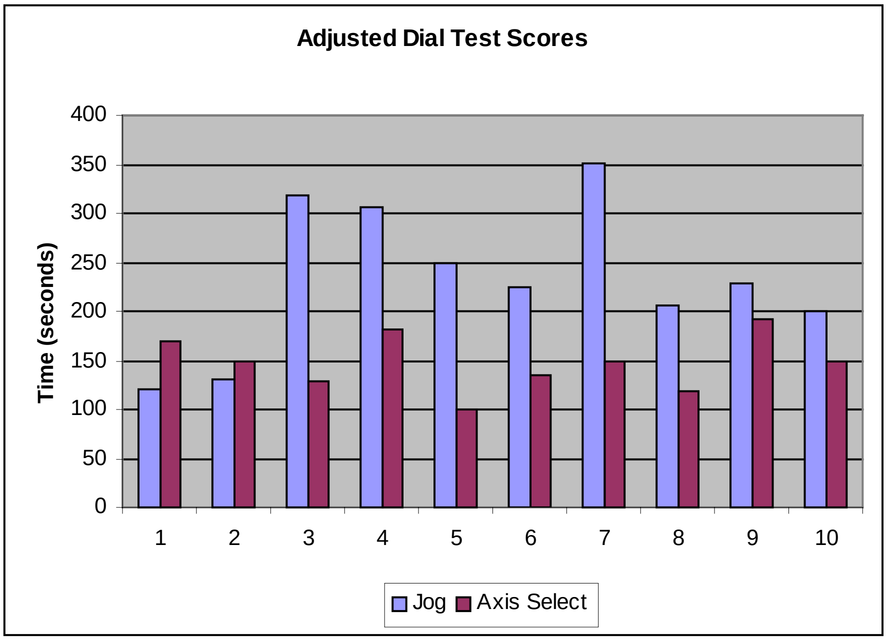


Figure 66 - Phase II Adjusted Dial Test Scores

This gives a clear advantage to the axis select mode. The next figure shows the performance difference displaying the adjusted and original relative scores for each method.

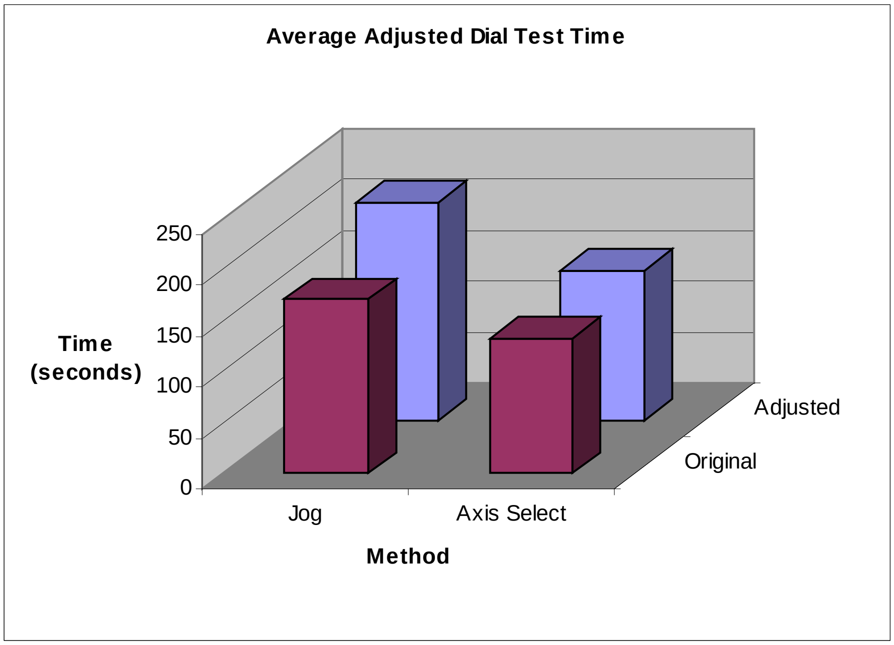


Figure 67 - Phase II Average Adjusted Dial Test Time

6.4.3 Trace Test

The trace test data is surprisingly different in that the jog mode at first sight appears to be more effective for six of the ten participants. Two of the ten failed the jog test, and these times were consequently not included in calculating the averages. Figure 32 shows the scores of each volunteer. Figure 33 shows the averages between jog mode and Axis switching mode for comparison. The average times favor the jog mode. Next we will look at the adjusted times to include the effect of error significance in the next section.

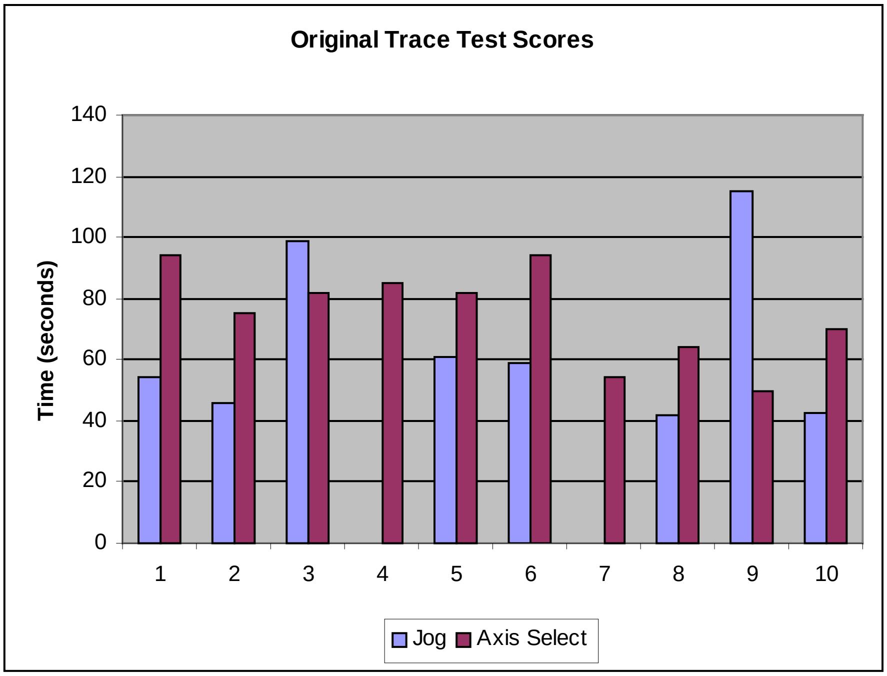


Figure 68 - Phase II Original Trace Test Scores

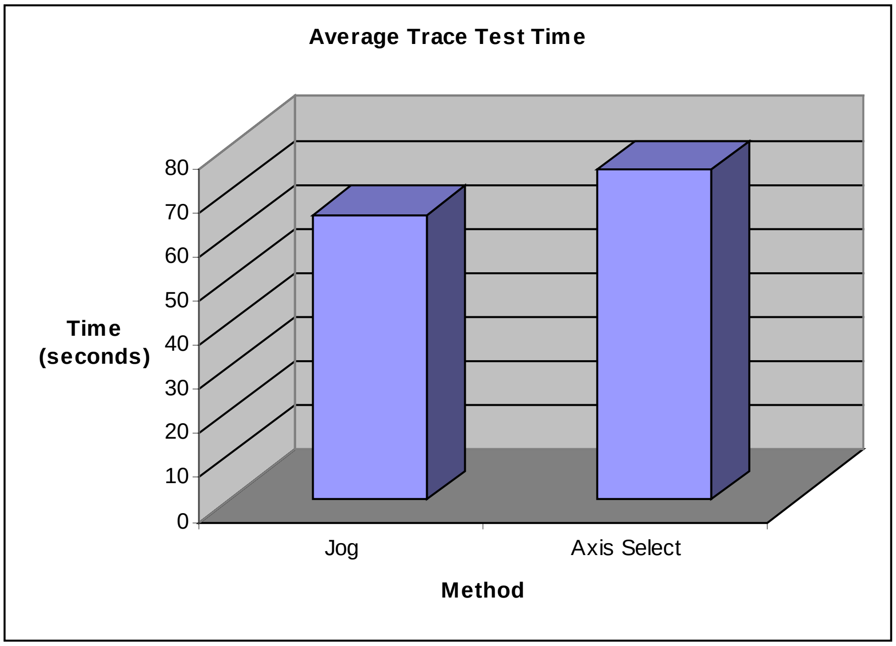


Figure 69 - Phase II Average Trace Test Time

6.4.4 Adjusted Trace Test

The error significance of the trace test shows the weakness of the jog mode. Two people were unsuccessful after five tries because they were unable to avoid the cups when weaving figure eight’s. The Axis Switching mode provided more controlled movement giving the clear advantage in the following graph.

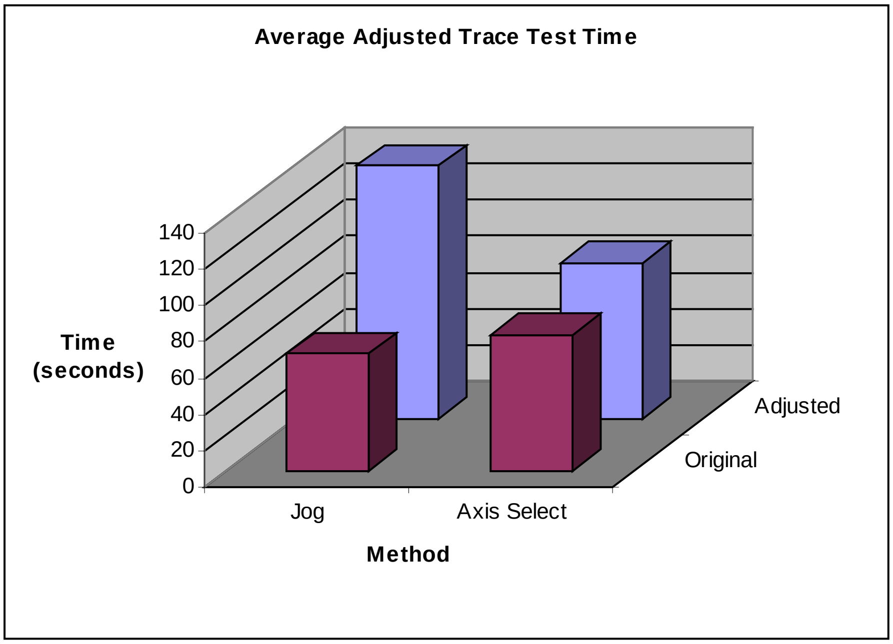


Figure 70 - Phase II Average Adjusted Trace Time

6.5 Position Recording Data

The recorder process kept an accurate record of position and time for all tests. The next two figures show a plot of the x and y position during the trace test. In the first figure, you can see that every movement is in a straight line because movement is only allowed in one direction at a time. The second figure shows the points covered during the jog mode for the same volunteer. There are diagonal movements, but since all velocity is constant, all angles traveled are at a 45-degree angle.

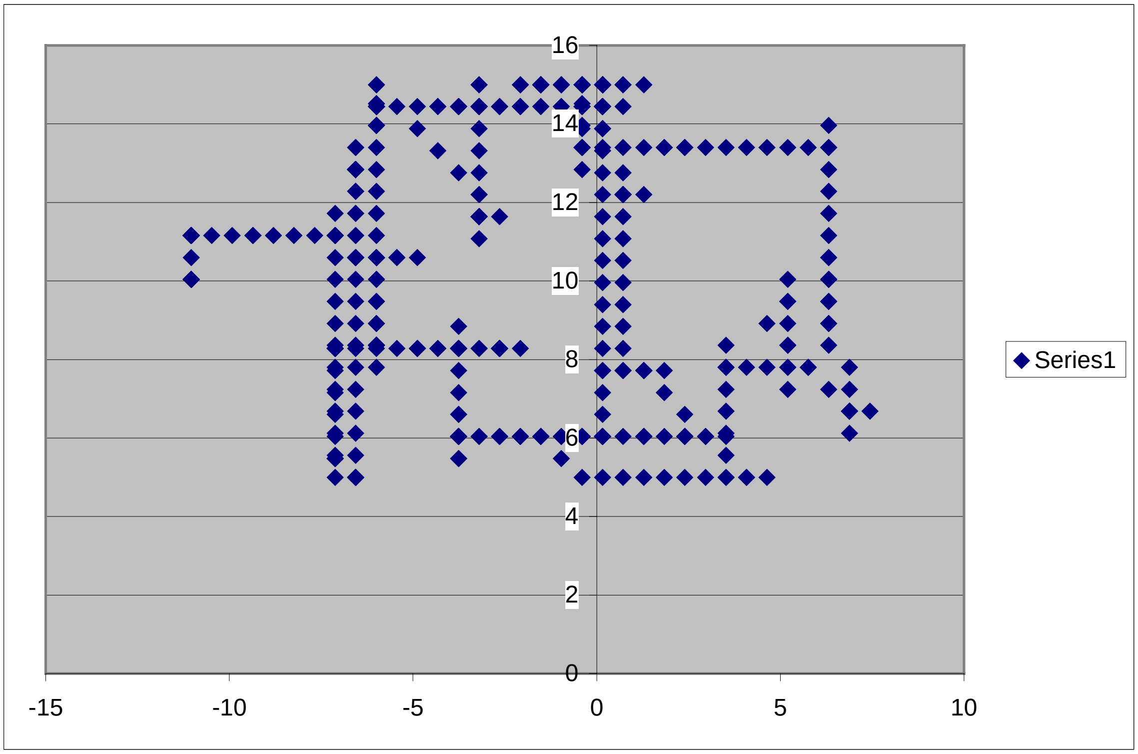


Figure 71 – Data From A Subject Performing Simple Jog Mode

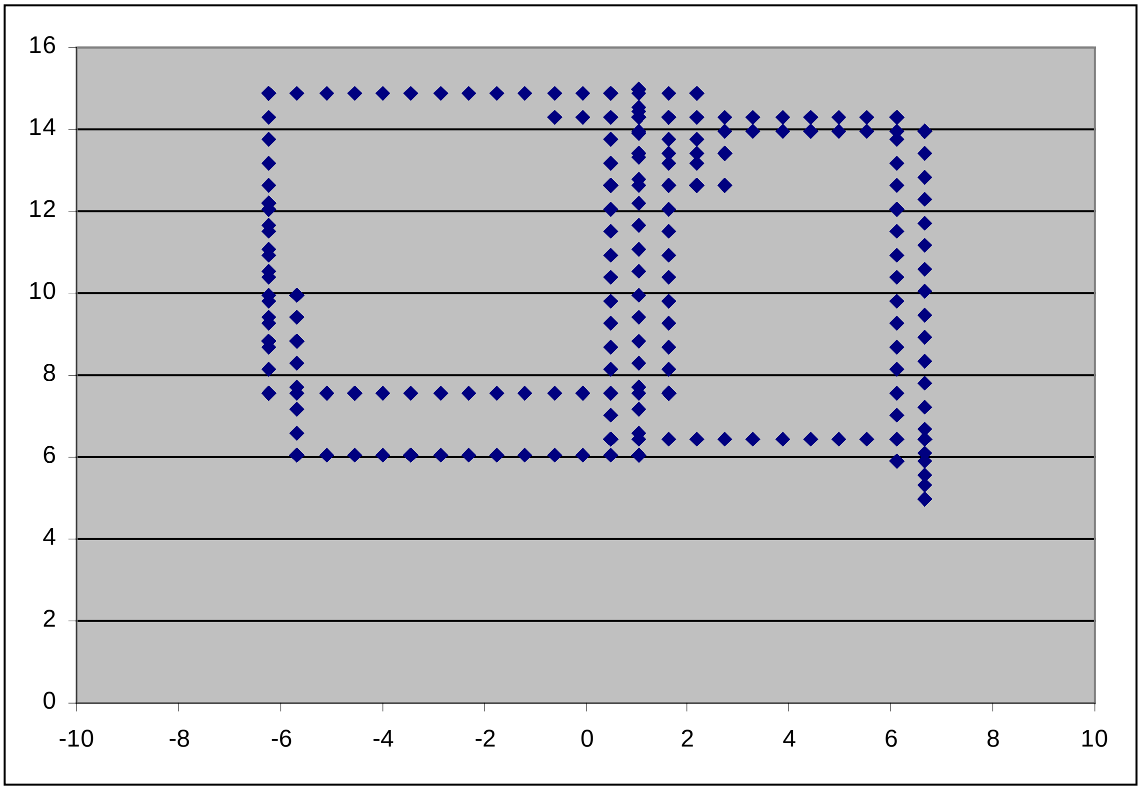


Figure 72 – Data From A Subject Performing Bipolar Jog with Axis Select Mode

The jog mode, which allows for movement in three directions at one time, has the distinct advantage of allowing for more efficient motion over methods that only allow for single axis movement because it may allow for more efficient movement. The shortest distance between two points is a straight line, which requires multi-dimensional movement if the points lie on different axes. It is very possible to learn multi-dimensional movement because there are a limited number of combinations between the three inputs.

Test subject, Matthew Shelly, confirmed:

> “The first one is better for people with good coordination with their motor skills or that have done, practice, this because they can take advantage of

> 3d moving, the only problem with that is it is easier to have one going unintentionally. For the second method, much simpler, therefore you can do things more quickly without unintentionally mishaps slowing you down and for people that are beginners or poor motor skills, much simpler. (It is) Much easier to think in only one direction at a time - easily correctable if a mistake is made, but if you are able to use the first way, you will be better if you have practiced enough."[4]

Matt's analysis agrees with the results from the tests.

7 – Conclusion and Future Work

7.1 Final Remarks

The purpose of this project was to explore control interface options for people able to interact using EMG's as inputs to a computer. The different modes developed and tested favor simplistic and straightforward methods when based on novice users. It was strongly shown by nearly 30 volunteers that an axis selecting bipolar method was the most effective method for three-dimensional robotic motion.

The dexterity tests for the different methods seemed very, almost overly, complicated. Ironically, when the test subjects were first given the opportunity to tryout the movements almost everyone was successful at moving in all six directions. It is when the test began that failure and frustration arose. The three most prominent factors were people attempt to coordinate more than one movement at a time, confusion when trying to move to a new direction, and competitive nervousness.

Some contestants felt they could utilize more complicated facial movements as a mechanism to help accelerate their performance. This included smiles, frowns and other multiple muscle movements, which clouded their perception of a simple straightforward approach to controlling proper muscle stimuli. When smiling, signals were highly coupled, causing unintentional movement. These poor mechanics led to bad habits, which in the short run are hard to over come. It may be true that once mastered, these types of macro style movements may be more effective in the long term, but for this experiment, they were very counterproductive.

Familiarity increases performance, as orientation and coordination learning will eliminate the confusion factor. For each every person the learning curves are different. Those that adapted well were the same type of people that succeed at video games, ping-pong, and other coordination efforts.

Nervous testees sometimes panic when exposed to testing. They tend to over exaggerate movements and frequently try too hard. Their behavior is cyclic as poor performance leads to trying even harder which leads to more exaggerated movements which leads to even poorer behavior.

7.2 Further research

Other velocity controlling concepts that were not tested include multiple speed control & continuous use based acceleration because the experiments showed variable control inhibited the completion of the task. Over time with learning, a modified jog mode with multiple speed ranges may prove to be beneficial.

The forward and reverse kinematics between Cartesian coordinates and spherical coordinates were used by default, but other coordinate schemes such as spherical and cylindrical should also be investigated.

Signal coupling was the largest inhibitor to success. A simple transform could be used to remove coupling from the data during the calibration routine if all signals were recorded during the maximum and minimum readings of the EMG. This has the potential to be instantly beneficial to the inexperienced users, but will not give good feedback to the user of effective signal production. This could lead to bad habits prevent users from learning how to maximize each signals precision and dynamic range.

The outdated Rhino robot was sluggish, unreliable and hard to fix. In the future, a better robot will lead to more efficient testing and effective tools useful for completing tasks. A larger robot would be more useful since the rhino’s range of 3 feet is rather restricting. If a robot were force sensitive, this would increase the potential for safety, durability, and accommodating kinematic constraints in manipulation.

References:

[1] LaPlante Ph. D., Mitchell P. and Carlson Ph D., Dawn "Disability in the United States; Prevalence and Causes." Report 7 Disability Statistics Center at UCSF, (1996)

[2] **http://www.crosswinds.net/~missq/stat.htm**

(The following information relates to traumatic spinal cord injury. It was compiled primarily by researchers at the University of Alabama using data from the regional SCI Centers funded by NIDRR. For more information on spinal cord injury statistics call 205-934-3320, the National Spinal Cord Injury Statistical Center, Birmingham, Alabama.)

[3] Shelley, Matthew. Personal Inteview. 17 March, 2001.

[4] Animal Physiology 4th Ed. (Eckert)