Using Force Sensors to Effectively Control a Below-Elbow Intelligent Prosthetic Device

Jeremy Evan Blum

Byram Hills High School

Armonk, NY

"The industry's aim of a thousand years ago endures—foremost not to rebuild the human part, rather to offer its basic function"

-Meier *et al*., 2004

Review of Literature

I. *Introduction*

The human body is an ingenious result of evolution. Intelligent prosthetic devices, those utilizing computerized systems and minimal user input, cannot yet mimic the human range of motion; however, important new technologies are making it increasingly possible to restore practical function.

State-of-the-art prosthetic devices are often prohibitively expensive (~$18,000) (Touch Bionics, Inc., Livingston, UK) and may require the surgical implantation of electrodes and sensors, something that many people with limb deficiencies will not tolerate [1]. Those in need of prosthetics usually prefer a device that is easy to maintain, equip and learn, goals on which this research focuses.

II. *Methods of Prosthetic Control*

Current principle control methods include passive, cable, experimental neural control, and myoelectric control. In passive control, a prosthetic hand is essentially locked into one of a limited number of chosen positions. Passive models are generally used as strictly cosmetic devices, with limited manipulative abilities [2].

Cable, or body powered control, allows for the simple direct control of a prosthetic device. They make use of cables connected to existing residual limbs in order to control the hand [3]. The OttoBock hand, for example, uses deliberate pulls on a cable to control movement.

Neural control is a potential future control method that is in its infancy. It works by using electrodes on the brain's surface to intercept limb control signals in the form of goal and trajectory signals. These are then translated into movement using a microcontroller. Neural control is not yet fully practical and does not work efficiently for a single limb below-elbow amputation [4].

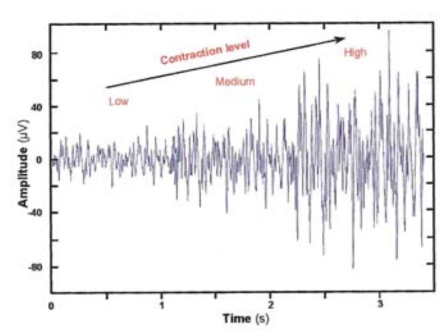


Figure 1: Variation of Myoelectric Signal with Contraction Level [6].

Presently, the most effective and accurate type of prosthetic control is myoelectric control, first envisioned in 1945 by Reinhold Reiter of Munich University [5]. All muscles generate natural electrochemical potential when they contract (Figure 1); these myoelectric signals (MES) can then be read by myoelectrodes and amplified to measure a muscle's naturally generated electricity. After myoelectric processing via a microprocessor, these signals can be designated to control a particular degree of freedom in the prosthesis [6].

If a MES is picked up by an electrode, it indicates that the muscle is being contracted. This signal can then be processed and evaluated to determine whether the signal is active enough to indicate that the muscle has been intentionally tensed. If the signal is active enough, the controller will instruct the prosthesis to operate based upon the signal's amplitude.

III. *Limitations of Current Prosthetic Control Methods*

Though the most sophisticated current control method is advanced pattern-recognition myoelectric control, it still has a number of disadvantages. Primarily, it remains relatively inaccurate, with advanced models correctly determining muscle activation approximately 95% of the time when using four input channels. Therefore, one out of 20 times, the hand will operate in an undesirable manner [7] (figures are not available for conventional myoelectrodes that do not use pattern recognition) [8]. Pattern recognition, a control method under development, requires the surgical implantation of electrodes, which run the risk of becoming infected or falling out of position. Finally, extensive signal processing must be performed to interpret the signal and remove excess noise before it can control a myoelectric device. Various processing techniques, including time-frequency analysis, wavelet analysis, neural network, and fuzzy classifications, have been developed, but none exist that work without flaws [9]. The need for customized signal processing makes it difficult to customize one design for different users with different needs and disabilities [10].

Creating an improved control method that can overcome these obstacles is essential to patients. This research focuses on using force sensors in place of myoelectrodes as a new intelligent prosthetic control method.

IV. *Creating a More Accurate Control Method*

Force sensors are piezoelectric, meaning that their output voltages can be manipulated based on the amount of pressure applied to a resistive ink. They can be used to measure the contraction of muscles, and the resulting voltage measurement can be compared with the electrical signal generated when measured by conventional myoelectrodes. While others have used devices to measure muscle bulge, it has never been measured using force sensors, nor has it been implemented in a multi-sensor, pattern recognition setup with the purpose of controlling a prosthesis [11]. By investigating this unexplored control method, this research has the potential to make some applications of multifunction prostheses less expensive and less invasive with the potential to eliminate signal processing.

Hypothesis

If direct measurement via force sensors accurately predicts muscle bulge comparably to indirect measurement by surface myoelectrodes, while effectively demonstrating a sufficient resolution between various forearm muscles, it will offer an alternative form of prosthetic control. This may provide a simple yet superior low-cost method to accurately detect muscle movement without invasive surgery, while appealing to a much broader socioeconomic group.

Objectives

I. *Objective 1*

A proof-of-concept hand, capable of using a single force sensor and a microprocessor to control hand movement, will be developed to respond directly to force sensor input in order to demonstrate the practicality of this control scheme.

II. *Objective 2*

A computer interface will be constructed that will allow readings to be taken from multiple channels of force sensors and myoelectrodes so that the inputs from multiple muscles may be compared for each method.

III. *Objective 3*

An input analysis program will be written to compare the accuracy of the force sensor inputs to the myoelectric inputs.

Methods and Materials

I. *Prosthetic Prototype: Objective 1*

I built the prosthetic prototype (Figure 2a, and 2b) as a proof-of-concept to demonstrate that force sensor control of a grasping mechanism could be accomplished.

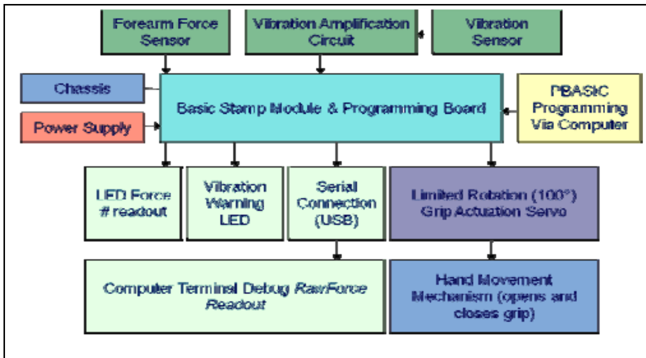


Figure 2a: Prosthetic Prototype Schematic (Blum, 2007).

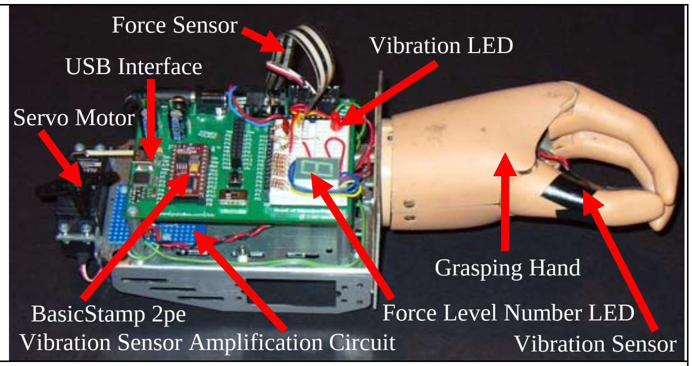


Figure 2b: Prosthetic Prototype (Blum, 2007)

The BasicStamp 2pe microprocessor (Parallax, Inc., Rocklin, CA) was run on PBASIC language, a modified version of the popular BASIC programming language. It was chosen due to low power consumption (5V at 15mA current draw), processor speed (8MHz), and large read-only memory (ROM) for downloading programs (16 x 2000 Bytes). It was chosen over BasicStamps with slightly faster processors due to its fairly quick processing speed, but very low power consumption, which resulted in less battery usage.

To control the hand’s grasping mechanism, a 100º limited rotation grip actuation servo (Parallax, Inc.) was used for movement because of its design as a PBASIC-controlled motor. Its movement was controlled by square-wave pulse-width modulation determined in PBASIC.

A FlexiForce model A201 force sensor (Tekscan, Inc., South Boston, MA) was utilized as the main activator for hand movement because of its low price and durability. Via PBASIC programming, levels of force measured by capacitor charge time (0.1 μF capacitor) were assigned to different motor degrees. This resulted in a semi-proportional control scheme in which a range of raw forces were assigned to a single activation magnitude to achieve as much proportional control as was possible with a BasicStamp. To debug the force level, a numerical readout LED was programmed to display the force level on an arbitrary scale from zero to six. Figures 3a and 3b show voltage levels produced by the sensors at various levels of force, and the values for R (ohms) and C (μF).

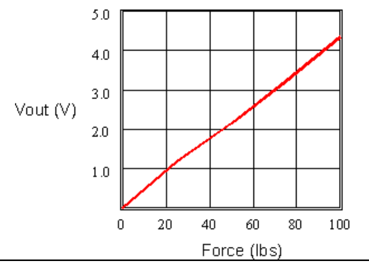


Figure 3a: Force inputs and resulting voltages (tekscan.com, 2006)

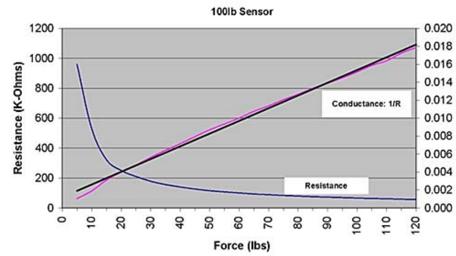


Figure 3b: Resistances as a function of force (tekscan.com, 2006)

A vibration-activated slip sensor that could sense and then arrest a slipping object was installed. Its successful operation demonstrated that a force-activated prosthesis could be successfully interfaced with technology that is currently used on myoelectric arms. A non-inverting voltage amplifier circuit, using a current to voltage converter, amplified the charge created by the piezo sensor. The final amplification was equivalent to 1+(12M/12), or 100001 (Figure 4). This circuit was designed using a TLC2272 dual channel operational amplifier (Texas Instruments, Inc., Dallas, TX) to increase the signal level from the vibration sensor mounted in the grip. The operational amplifier could take two input signals and amplify the difference, providing a voltage gain. A voltage amplifier was used instead of a charge amplifier because it exhibits less temperature dependence, a potential problem when using piezo film sensors (Measurement Specialties, Inc., Hampton, VA). A red warning LED was installed and instructed by the PBASIC program to illuminate every time the vibration sensor circuit detected slip with the hand automatically arresting it.

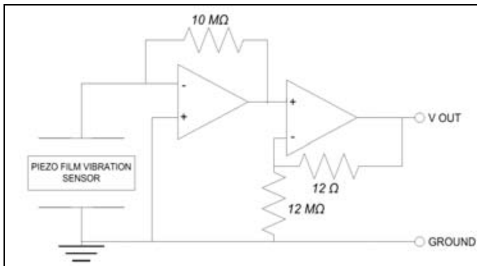


Figure 4: Vibration Sensor Circuit (Kyberd/Blum, 2007)

II. *Computer Interfaced Force Sensor Circuit: Objective 2*

A computer interface board was developed to allow the force sensors to communicate with MATLAB, a computing program and language used for acquiring and analyzing data. A

USB Data Acquisition Device (DAQ) from National Instruments was interfaced with a circuit board to amplify the signal from the four force sensors so that the resulting voltages could be fed into the computer. The main amplification unit, a TLC2274 quad channel operational amplifier (Texas Instruments, Inc., Dallas, TX), was used because of its ability to simultaneously amplify up to four channels (Figure 5).

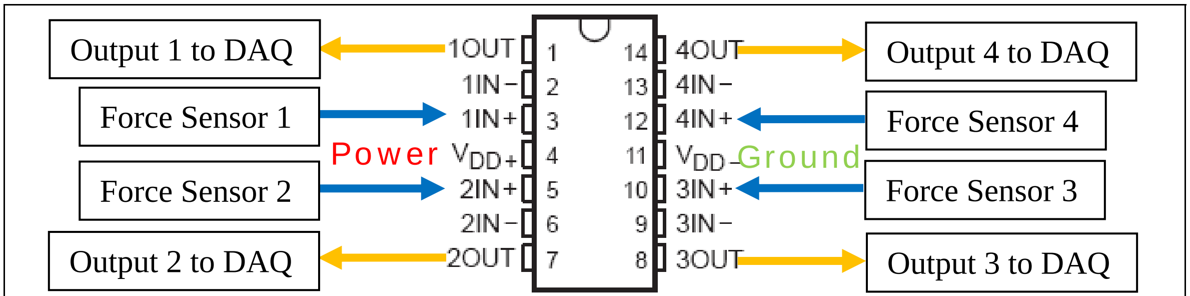


Figure 5: TLC2274 Operational Amplifier (adapted from Texas Instruments, Blum, 2007)

This complete setup (shown below in Figure 6) allowed four force sensors to have their voltages accurately measured on a USB-enabled computer.

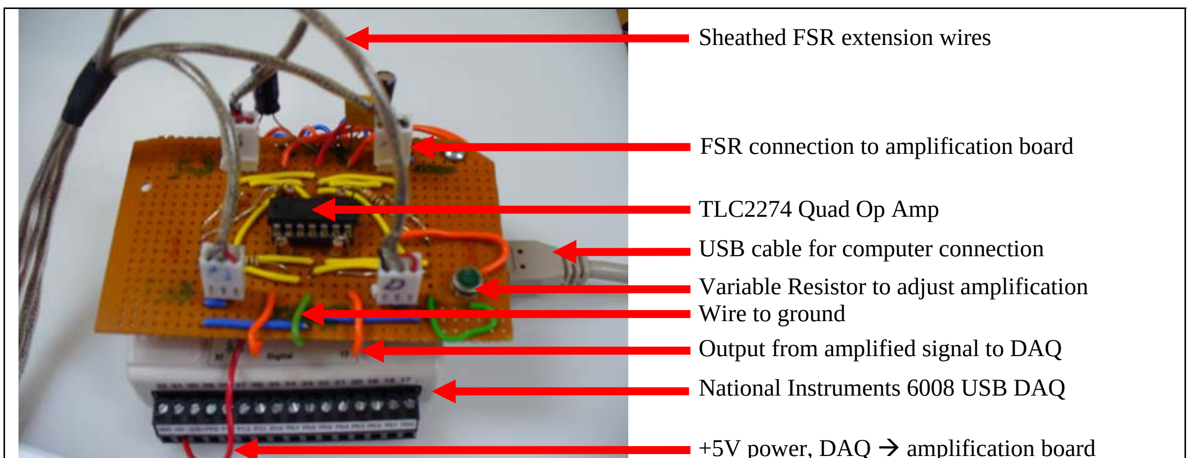


Figure 6: DAQ and amplification circuit board with four force sensors connected (Blum, 2007)

The last device needed to capture muscle bulge levels was a thermoplastic forearm splint cast equipped with force sensors. The cast was molded to my arm, fitted with force sensors, and retained with Velcro® straps. It was fabricated for the author by a prosthetist at the University of New Brunswick's (UNB) Institute of Biomedical Engineering using a sheet of low-temperature molding plastic. The sheet was heated and fitted around the arm, and a thumb hole was cut, preventing the cast from rotating once on. On a prosthesis user, the socket would be locked around the elbow for the same reason. The cast was cut down the middle for ease of removal and was fitted with Velcro to allow tension adjustment. Next, the author, with the assistance of Dr. Peter Kyberd of UNB's Institute of Biomedical Engineering, laid out the force sensor locations. Due to limitations in the available number of simultaneous input channels, four force sensors were added for this proof-of-concept. The four muscles used were associated with four different actions:

Table 1: Four Forearm Force Sensor Locations and Associated Actions (Blum, 2007)

Flexor Digitorum Superficialis - finger flexion	**Extensor Carpi Ulnaris** - finger extension
Flexor Carpi Ulnaris - wrist flexion	**Extensor Carpi Radialis Brevis** - wrist extension

These four forearm muscles were chosen because they are generally still accessible on an individual with a partial limb deficiency. Pronation and supination were dependant on Pronator Teres and the Supinator, respectively, and could not be used. The areas of largest muscle bulge difference were marked on my arm by Dr. Peter Kyberd (Figures 7 and 8).

Figure 7: Bulge areas for muscle *#2 (Extensor Carpi Ulnaris), #4 (Supinator), and #6 (Extensor Carpi Radialis Brevis)* (Blum, 2007)

Figure 8: Bulge areas for muscle *#1 (Flexor Digitorum Superficialis), #3 (Pronator Teres), and #5 (Flexor Carpi Ulnaris)* (Blum, 2007)

Force sensors were mounted inside the cast using standard double-sided tape (Figure 9). Sheathed extension wires were soldered to the force sensors to reduce the level of external electrostatic noise pickup from local sources such as power line frequencies and transformers.

Molex® connectors were utilized for easy connection with the amplification board (Figure 10). Finally, the cast was fitted, and the force sensors were tested (Figure 11).

Figure 9: Internally mounted force sensors (Blum, 2007)

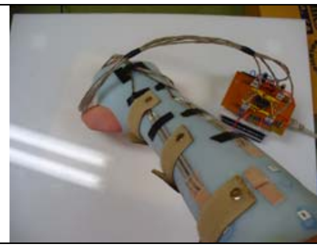


Figure 10: Cast worn by Blum, connected to DAQ (Blum, 2007)

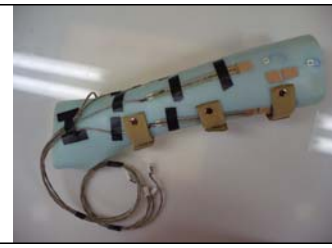


Figure 11: Cast with sensors mounted (Blum, 2007)

The last step of the second objective was completed with multiple MATLAB programs, coded by the author, used to record force sensor data for a fixed period of time (capture time), analyze it, and then export the data to graphs for later comparison[1]. I wrote a command prompt-based user interface to allow control of all aspects of data acquisition and analysis. Three types of muscle bulge were analyzed, each with a capture time of 10 seconds and a sample rate of 1 KHz. First, a resting measurement was taken, where no muscle bulge occurred; this served as a baseline for the active bulge data. Next, calibration data, or a continuous 10 second flex of a single muscle, was acquired for each of the six muscles undergoing testing. Last, action data, or a pause, followed by a flex of a single muscle, followed by a pause, was acquired for each of the six muscles undergoing testing. The muscles tested are shown below in table 2.

Table 2: Six Forearm Muscles Tested (Blum, 2007)

Flexor Digitorum Superficialis - finger flexion	**Extensor Carpi Ulnaris** - finger extension
Flexor Carpi Ulnaris - wrist flexion	**Extensor Carpi Radialis Brevis** - wrist extension
Pronator Teres - wrist pronation	**Supinator** - wrist supination

III. *Computer Input Analysis Program: Objective 3*

In order to show that a force sensor-controlled hand would only move based on desired input, I created an analysis program to convert raw data into a single control signal that would vary based on the various inputs. For this algorithm, a linear discriminant analysis method that

[1] *See appendix A for complete description of acquisition and analysis process.*

had been previously employed by the SVEN myoelectric hand was adapted [12]. The SVEN Function is defined as follows:

$$F(x)=Wx+w_0$$

where an F(x) value greater than zero indicated activation, and a value less than or equal to zero indicated no activation; Wx and W_0 were found using the formulas shown below. If the force sensors were accurate, they should only result in a value above zero when the calibration muscle matched the action muscle. To keep the formula simple, it was assumed that the covariance of x was the same whether or not the function F was on.

The data was then acquired in this method: 10,000 samples of data were acquired over a 10 second period while the user was soliciting function F (a given muscle bulge). This data was called $x_1(t)$. Next, the same type of data was acquired while the user did not solicit function F. This data was called $x_2(t)$. MATLAB was used to calculate the mean signal vector of each set:

$$m_1=F(x_1) \qquad m_2=F(x_2)$$

MATLAB was also employed to calculate the covariance matrix of the data using the code:

$$C=cov(x_1)$$

Next, W and w_0 were calculated so that F could be determined[2]:

$$W=inv(C)*(m_1+m_2) \qquad w_0=(-.5)*((m_1+m_2)')*W$$

This process was carried out for all six input movements mentioned previously in table 2.

No sensors were placed to measure pronation or supination, but these were tested to ascertain if they could be separated using only four sensors. Finally, the action data was acquired in the method described earlier, and it was combined with the first SVEN Function into a new one, resulting in a graph showing how well each calibration predicted an action. The MATLAB code to do so was as follows:

[2] This is using MATLAB notation, where apostrophe (') means transpose the matrix, and inv() means matrix inverse

```
for i=1:number_in_col
    F(i)= ((W')*((x3(i,:))'))+w0;
end
```

A FOR statement was used to make sure this was calculated for all 10,000 samples. This resulting graph was then smoothed using another MATLAB program to remove noise resulting from the previous calculations.

With a final MATLAB program, this was converted to a digital signal where anything above zero was converted to a one (hand activated), and everything equal to or below zero was converted to a zero (hand not activated). This process was repeated with every combination of action data and calibration data to derive which force sensors picked up on which muscle bulges.

Results and Discussion

I. *Prosthetic Prototype: Objective 1*

Results were measured qualitatively and were used to demonstrate that a prosthesis could be controlled both accurately and efficiently using a force sensor-controlled program and accompanying circuitry. The novel BASIC program functioned properly, assigning a value between zero and six depending on the approximate force applied. The slip sensing circuit was also successful, alerting the processor of vibration and working to arrest it. While the prototype was bulky, this was a limitation of prototyping and not of the fundamental design. The cost of the prototype was less than that of an ordinary intelligent prosthetic device: the force sensors alone cost 100 times less than myoelectrodes (without the additional cost of surgical implantation). The potential for low-cost-of-manufacture indicates that force sensor-based control methods could be applied to create more accessible prosthetic options for a wider socioeconomic group.

II. *MATLAB Simulations of Force Sensors' Accuracy: Objectives 2 and 3*

Inactive calibration data was the first to be acquired, as it was necessary to act as a baseline for later comparison by the analysis program. It consisted of keeping the entire forearm relaxed. The voltage graph showed very low levels of noise or activation, indicating that interference would not be a concern for force sensor control (Figure 12).

All Muscles Relaxed (Force Sensors)

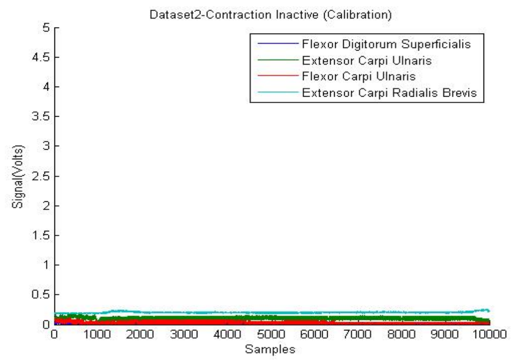


Figure 12: Inactive Calibration Data (force sensors)

The following raw calibration data shows that the active muscle was the one that produced the highest voltage level in each case, indicating accurate prediction. The resolution between the voltages from different muscles indicates that force sensors differentiated correctly. The following graphs are digitized values of the amplified sensor inputs (Figure 13).

(a) Finger Flexion (Force Sensors) **(b) Finger Extension (Force Sensors)**

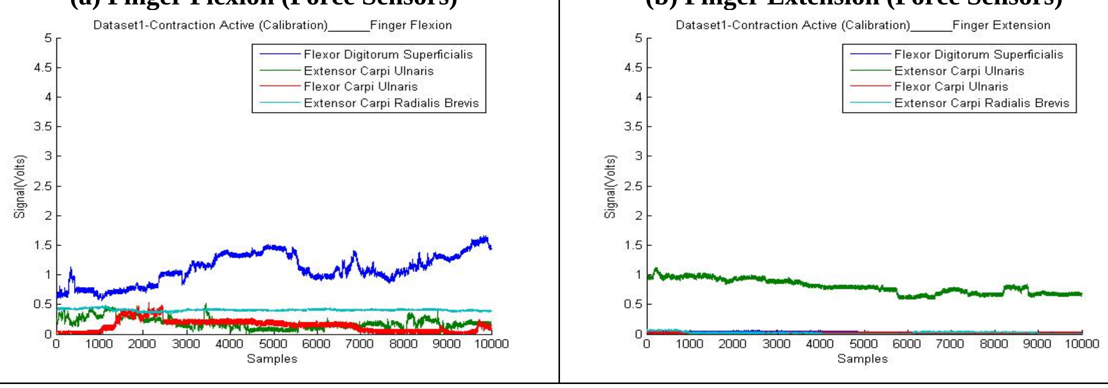
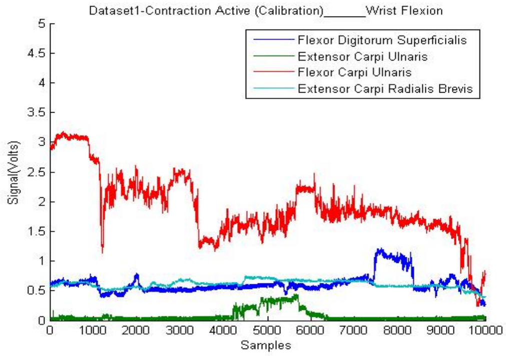


Figure 13 (a-b): Active Calibration Data (force sensors)

(c) Wrist Flexion (Force Sensors)



(d) Wrist Extension (Force Sensors)

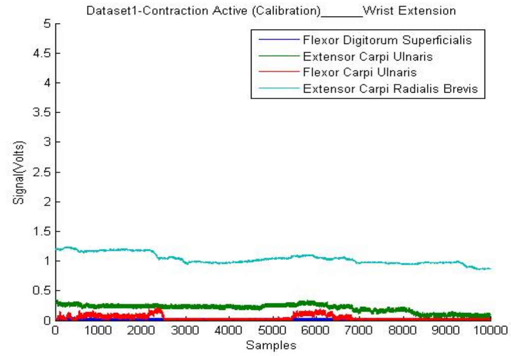


Figure 13 (c-d): Active Calibration Data (force sensors)

In figure 13a, when I flexed my fingers, the Flexor Digitorum Superficialis showed the greatest voltage, indicating that the force sensors had accurately predicted the correct muscle. This held true for the active calibration data for Extensor Carpi Ulnaris, Flexor Carpi Ulnaris, and Extensor Carpi Radialis Brevis as well, showing that all force sensor locations resolved the correct muscle.

Pronation and supination were also tested, despite that no force sensors were placed with the intention to resolve them. Pronation elicited a response similar to wrist extension, indicating that if the pronator teres muscle were to be used as a control muscle on a prosthesis user, the placement of the sensor would be critical to ensure minimal cross talk. Wrist supination, as expected, showed no significant response.

Matching raw data, including inactive and active calibration, was also acquired for the myoelectrodes. Inactive calibration data, as with the force sensors, was acquired to act as a baseline for later comparison using the analysis program. The inactive myoelectrode data showed levels of interference equally as low as the force sensors (Figure 14).

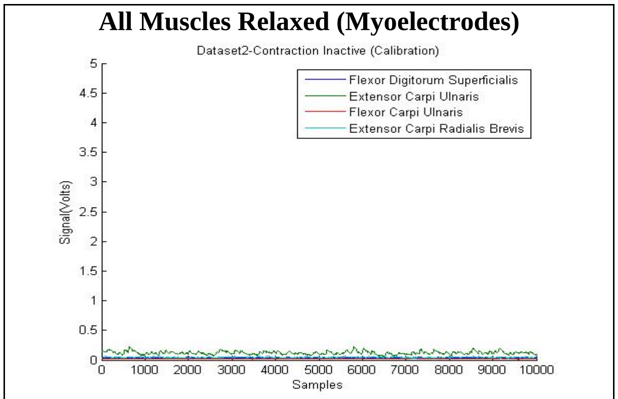


Figure 14: Inactive Calibration Data (myoelectrodes)

The raw active calibration data acquired from the myoelectrodes showed far more crosstalk and noise. The following is raw data from the four main myoelectrodes (Figure 15):

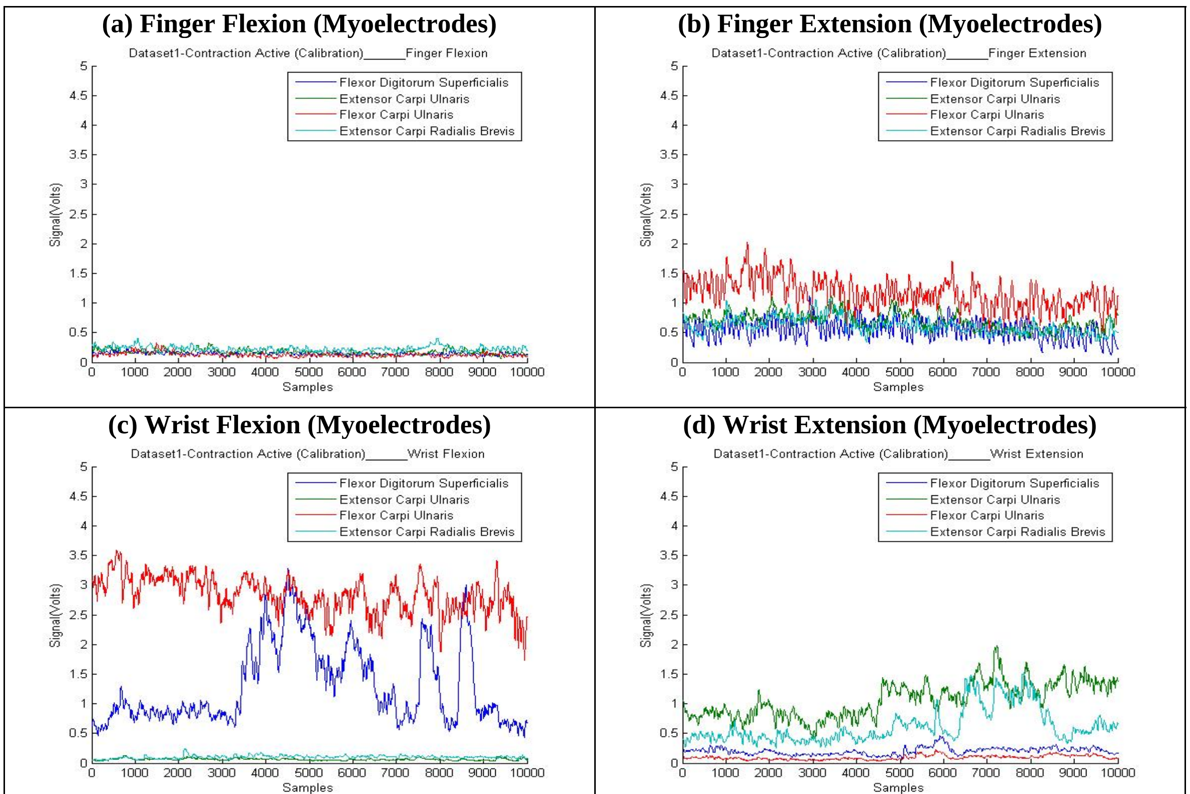


Figure 15 (a-d): Active Calibration Data (myoelectrodes)

Signal variation was significantly higher for the myoelectrodes, with voltages varying greatly between each sample (Figures 15 a-d). The only prediction that showed the correct voltage differences was that of wrist flexion (Figure 15c), compared with correct voltage differences on all four sensor channels.

Pronation and supination were also tested as before. Pronation elicited a response similar to wrist flexion, and supination elicited a response from Extensor Carpi Radial Brevis, the muscle that controls wrist extension, indicating that crosstalk would present a problem should these muscles be designated as control muscles on a myoelectric prosthesis.

Qualitatively, the raw data for the force sensors was accurate, showing high levels of differentiation between the voltages from different force sensors. However, the analysis algorithm, detailed below, did not always accurately represent this performance, often indicating concurrent activation for more than one input. The performance of the SVEN algorithm, when analyzing the myoelectrode data, resulted in the same problem, indicating that the issue was not with the force sensors but with the analysis algorithm.

There are 12 cases showing all the possible combinations between each calibration and action for both myoelectrodes and force sensors; for clarity there is a comparison of only one case: finger extension. Figures 16a-g show the action data for finger extension compared with the active and inactive calibration data sets from the six tested muscles using the SVEN algorithm to determine which ones would elicit a significant response from a prosthesis (an arbitrary value above zero).

(a) Finger Extension (Force Sensor Action Data)

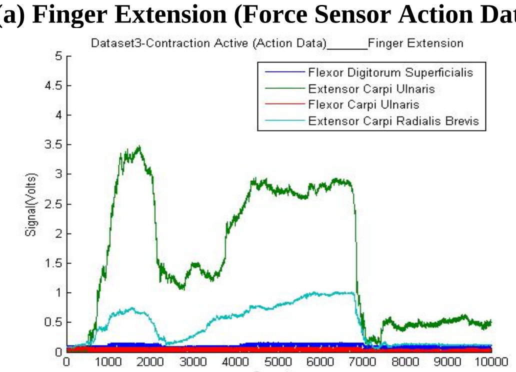


Figure 16 (a): Finger Extension SVEN outputs (force sensors)

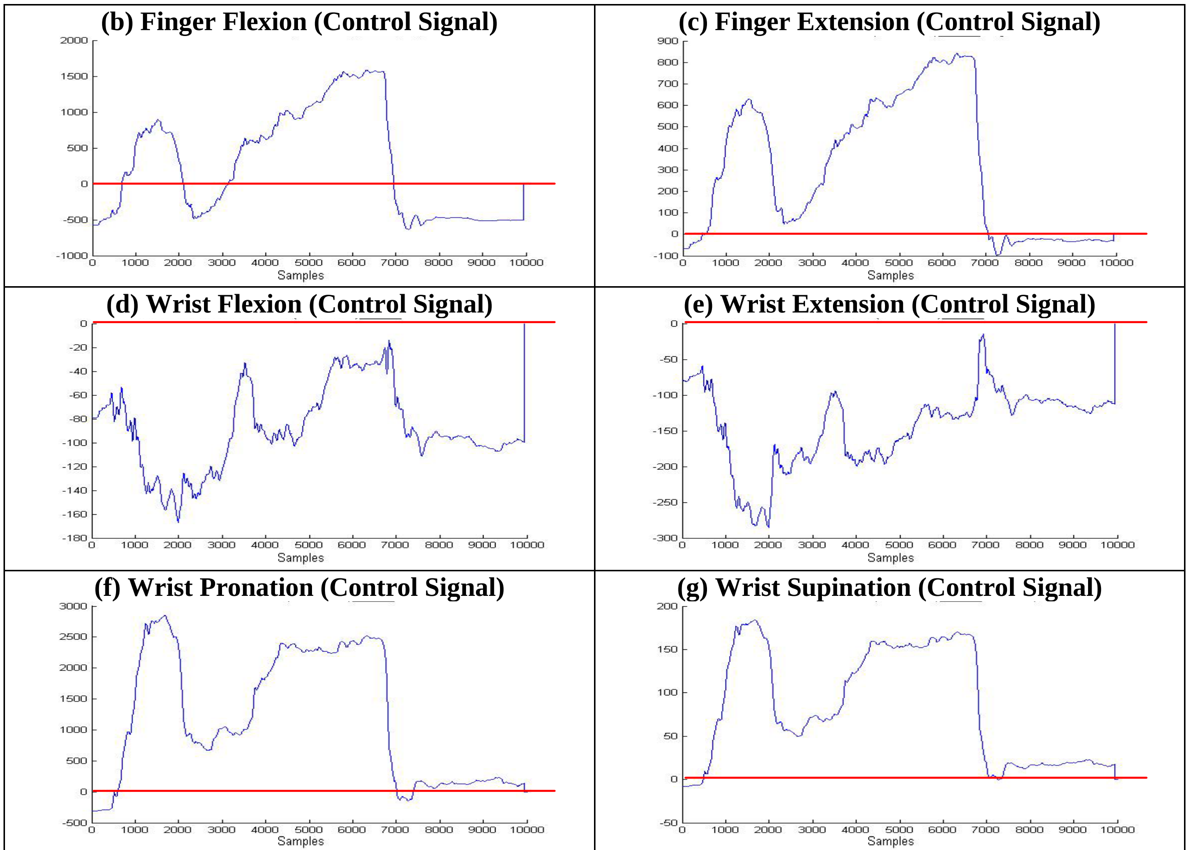


Figure 16 (b-g): Finger Extension SVEN outputs (force sensors)

As would be expected, the finger extension SVEN Function (Figure 20c) peaked at the appropriate times, indicating hand activation. However, finger flexion did as well. Neither wrist flexion nor extension showed any activation, which is good since they were not the muscles being activated; pronation and supination had fairly random outcomes, as would be expected. Should these be controlling a prosthesis, the hand would be activated whenever the control signal was greater than zero; in other words, the SVEN Function acts as a digital signal, with all values above zero indicating activation, and all values less than zero indicating no action. The errors that were present in this processing method could potentially be resolved in the future by performing a more advanced dynamic comparison with the voltages.

For comparison, following is the same data from the myoelectrodes (Figure 17 a-g):

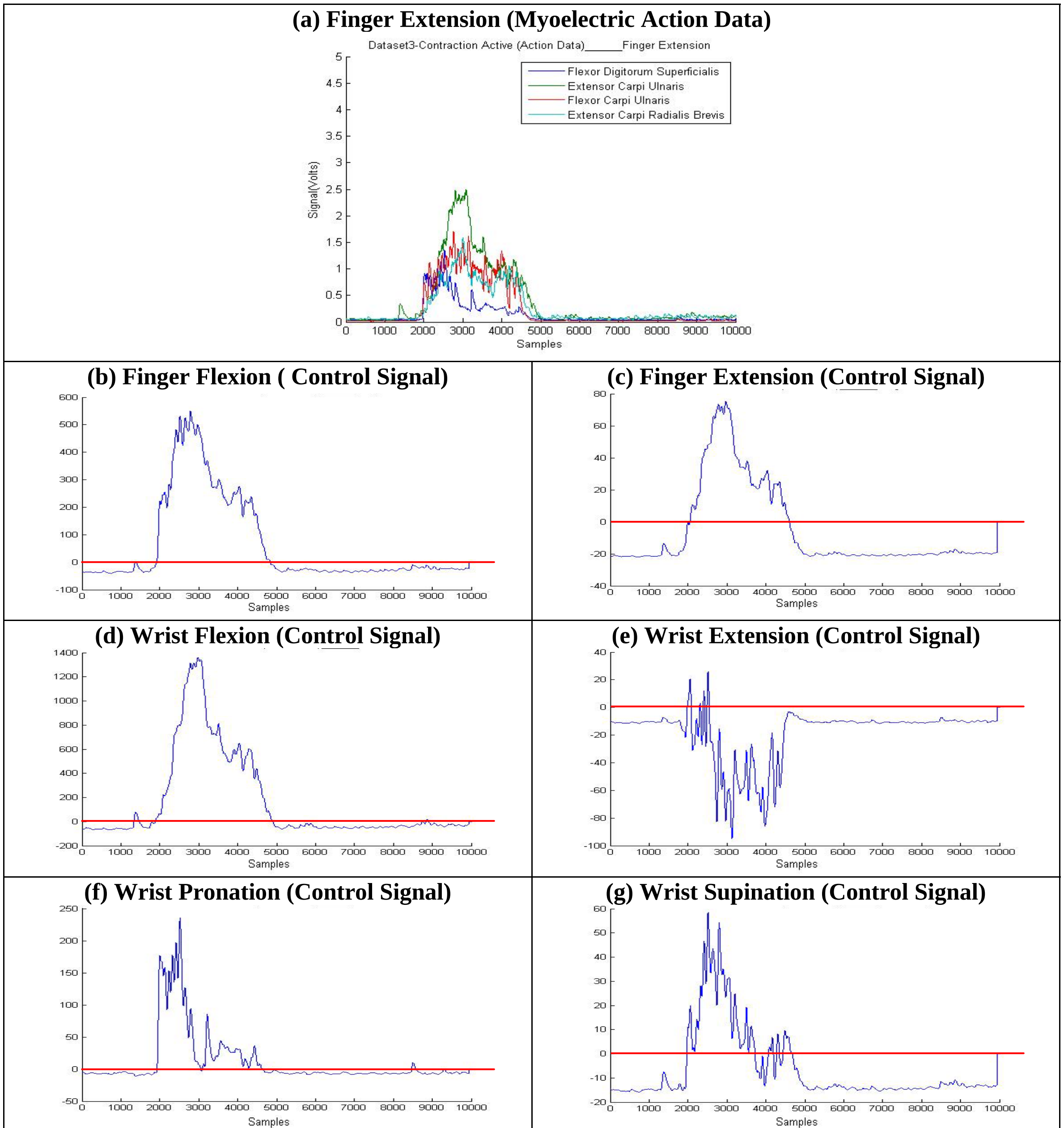


Figure 17 (a-g): Finger Extension Action Data and SVEN outputs (myoelectrodes)

Results were more variable in the myoelectrodes, most likely due to greater crosstalk in the raw data. All of the muscles except for wrist extension showed SVEN activation, indicating that the hand would activate far too often using myoelectrodes with this algorithm.

The results indicate that while this analysis algorithm is not yet perfected, the raw data from the force sensors suggests clear promise in their future use for prosthetic control. In general, they differentiated well between muscles, without undue crosstalk on other force sensor channels.

Conclusions

The qualitative results of the first objective demonstrate that force sensors hold a promising future. Force sensor control provides a low-cost solution, opening advanced prosthetic control to a larger socioeconomic group. They do not require implantation, thus eliminating the risk of infection or sensor movement, while allowing for the easy removal of a prosthesis, an advantage for many single-arm amputees. The low levels of electrical interference indicate that an amputee could be easily trained to make use of force sensors and that force sensor control may prove more effective than myoelectrodes in some situations. The raw force sensor data shows high levels of separation between signals, indicating their possible use as a control method without any post processing; a simple voltage boundary may be sufficient to determine muscle activation. The SVEN Function algorithm shows promise as a potential post-processing method, should it prove necessary, and requires perfecting before it can be used reliably. Low cost, durability, and the ability to avoid crosstalk, all indicate that future prosthetic devices could utilize force sensors either independently or in a hybrid control method.

Appendix A: Data Acquisition and Analysis System

1. Raw voltage signals from the force sensors were tested to ensure that there was minimum interference using an oscilloscope, MATLAB, and National Instruments LabView software.
2. The first program prompts the user to enter the desired sample rate and acquisition duration.
3. Calibration data is acquired for each of the six muscles. The program asks for the muscle the user is attempting to activate, so that the program may save the data appropriately.
4. Resting data is acquired once as a comparison point, and the data is saved to a file.

5. Activation data is acquired for each of the six muscles. The program asks for the muscle the user is attempting to activate, so that the program may save the data appropriately.
6. Once all acquisitions have been completed, user can re-import saved data files to perform analysis. First, resting data is imported, stored in memory, and a graph is exported. Second, activation data is imported, stored in memory, and a graph is exported.
7. Each of the six calibration data sets is compared to each of the six action data sets, resulting in a total of 36 outcomes. First, all 6 calibration data sets and the resting data are compared using the SVEN Function. The result of that is then compared via the SVEN Function again to the activation data to determine if activation has occurred.
8. A SVEN graph, a smoothed SVEN graph, and a Digital On/Off graph is drawn and exported.
9. Calibration graphs are visually compared with their associated action data to determine if muscle differentiation occurred.
10. The Digital On/Off signal can be used to activate a prosthesis.

References

[1] Kyberd, P. J. (2007, June). Personal Interview. (J. E. Blum, Interviewer)

[2] Meier, R. H. (2004). History of Arm Amputation, Prosthetic Resotration, and Arm Amputation Rehabilitation. In R. H. Meier, & D. J. Atkins, *Functional Restoration of Adults and Children with Upper Extremity Amputation* (pp. 1-7). New York: Demos Medical.

[3] Furlong, A. M. (2004). Aesthetic Restoration for the Upper Limb Amputee. In R. H. Meier, & D. J. Atkins, *Functional Restoration of Adults and Children with Upper Extremity Amputation* (pp. 165-173). New York: Demos Medical.

[4] Musallam, S., Corneil, B. D., Greger, B., Scherberger, H., & Anderson, R. A. (2004). Cognitive Control Signals for Neural Prosthetics. *Science, 305* (5681), 258-262.

[5] McLean, L., & Scott, R. N. (2004). The Early History of Myoelectric Control of Prosthetic Limbs (1945-1970). In M. Ashok, *Powered Upper Limb Prostheses* (pp. 1-15). Berlin: Springer.

[6] Lovely, D. F. (2004). Signals and Signal Processing for Myoelectric Control. In A. Muzumdar, *Powered Upper Limb Prostheses* (pp. 35-54). Berlin: Springer.

[7] Hargrove, L. J., Englehart, K., & Hudgins, B. (2007). A Comparison of Surface and Intramuscular Myoelectric Signal Classification. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 54* (5), 847-853.

[8] Kyberd, P. J. (2006, May). Online Interview. (J. E. Blum, Interviewer)

[9] Zheng, Y. P., Chan, M. M., Shi, J., Chen, X., & Huang, Q. H. (2005, July 11). Sonomyography: Monitoring morphological changes of forearm muscles in actions with the feasibility for the control of a powered prosthesis. *Medical Engineering and Physics*, 1-11.

[10] Poulton, A. S., Kyberd, P. J., & Gow, D. (2002). Progress of a Modular Prosthetic Arm. *Universal Access and Assistive Technology* .

[11] Kenney, L. J., Lisitsa, I., Bowker, P., Heath, G. H., & Howard, D. (1999, September). Dimensional change in muscle as a control signal for powered upper limb prostheses: a pilot study. *Medical Engineering and Physics*, 589-597.

[12] Midtgaard, T. M. (2006). Advanced Myoelectric Control of a Multifunction Upper-Limb Prosthesis. *Master's Thesis*.