Force-Based Controller for Myoelectric Prostheses

Kaitlyn Ayers[1], Rebecca Jones[1], Allyson King[1,2], Shaan Ramaprasad[1], Christian Stano[1]
Advisor: Emily Graczyk, PhD[3,4]

1 Department of Biomedical Engineering, Vanderbilt University, Nashville, TN. **2.** Department of Chemical and Biomolecular Engineering, Vanderbilt University, Nashville, TN **3.** Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH **4.** Louis Stokes Cleveland Veterans Affairs Medical Center, Cleveland, OH

Background

Current upper limb myoelectric prostheses are controlled by the communication of EMG signals from intact musculature to the prosthesis motor to provide limb functionality. However, these lack natural hand control schemes, which contributes to a pattern of frustration and abandonment of the device among users [2]. The Functional Neural Interface Lab of Case Western Reserve University developed a neural-connected sensory prosthesis system to restore sensation and an element of natural hand control to myoelectric prostheses. With this sensory feedback restored, further development is required to allow for a more accurate and natural user experience.

Figure 1: Example of a neural-connected prosthesis[1]

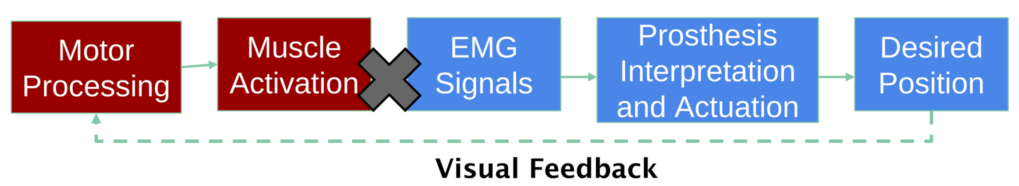


Figure 2: Demonstration of current control schemes for prosthesis, showing the required break in muscle activation for posture maintenance. (Red = User action, Blue = Prosthesis Action)

Project Objective

The objective of this senior design project was to convert a commercially available velocity-based prosthetic controller to a force-based controller through the manipulation of both software and hardware to more accurately represent intact muscle control and movement during hand posturing. This, in tandem with the restoration of sensory feedback, could allow for improved prosthesis user experience.

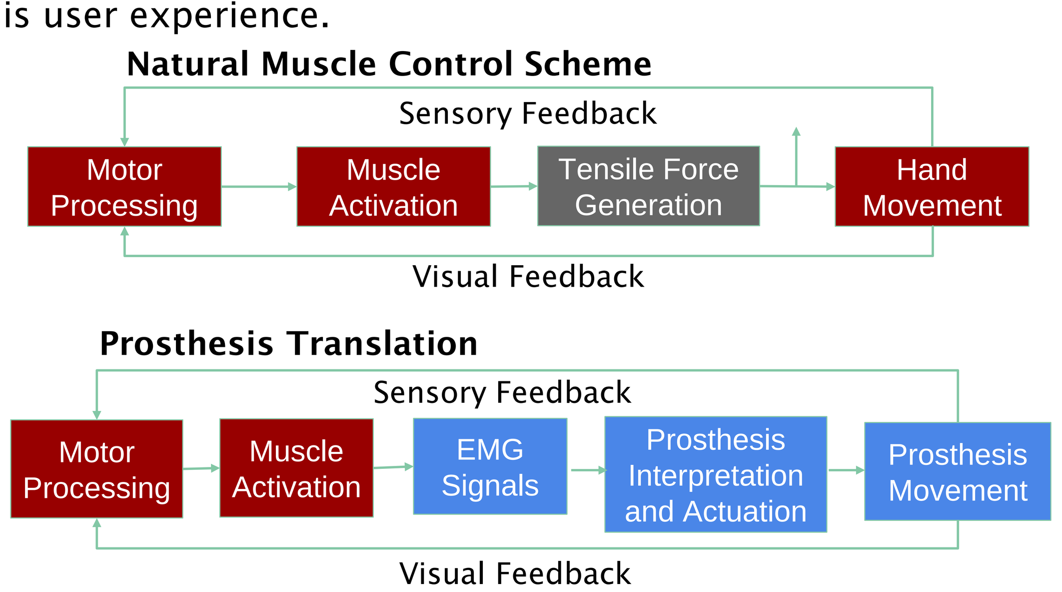


Figure 3: Natural muscle control schemes integrate both sensory and visual feedback to create a continuous feedback loop which is being translated into the prosthetic environment

Needs Assessment and Design Overview

- Wearable
- Durable
- Safe and Easy to Use
- Increased Intuitiveness
- Cost Effective
- Robust, Quick, Real-Time EMG to prosthesis actuation
- Continuous Feedback for Control of Prosthetic Hand

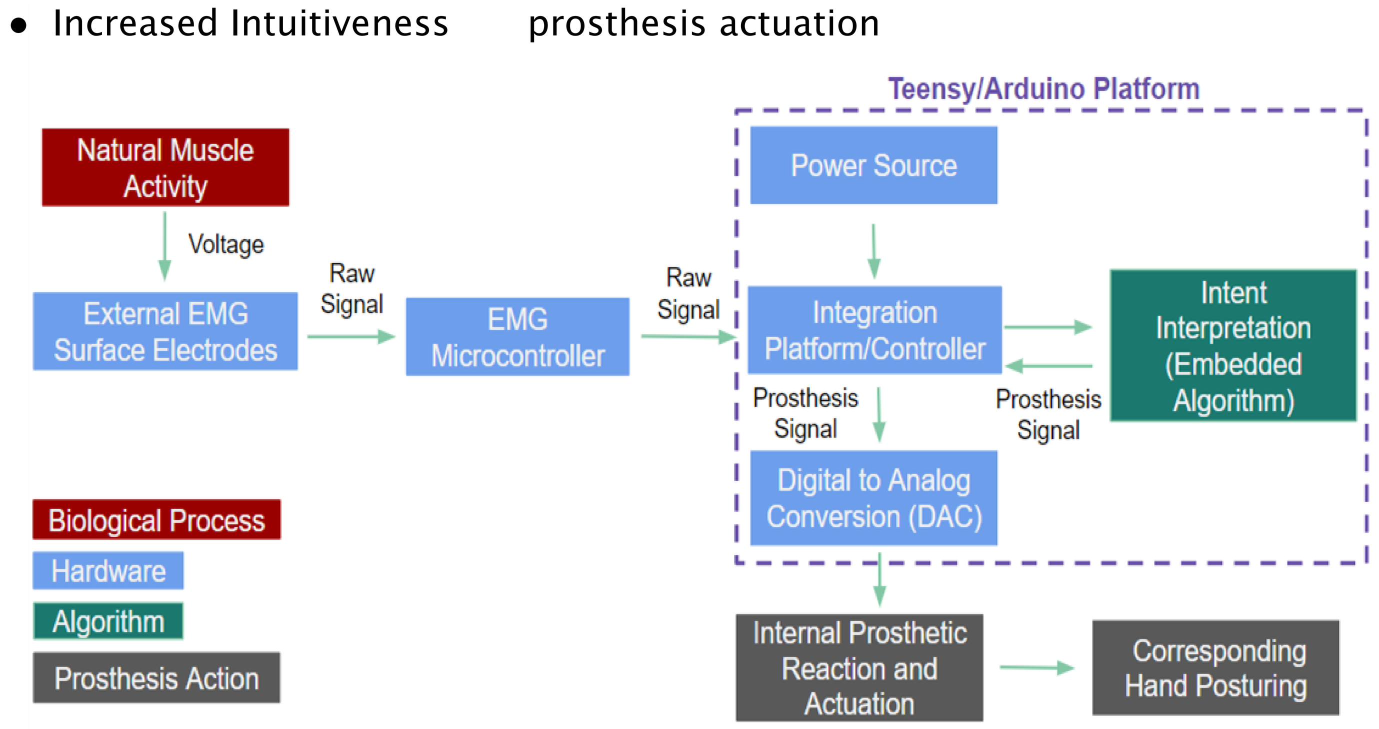


Figure 4: Design process flowchart depicting four major components that defined prototype development: EMG acquisition, motor actuation, external hardware integration, and the development of a novel software platform for alteration of control scheme

Software Development and Validation

Software development occurred in three stages: EMG signal processing and calibration, signal mapping to prosthesis position, and control scheme adaptation. Software was developed in the Platform IO environment which allows for easy implementation within the adapted Arduino/Teensy communication scheme.

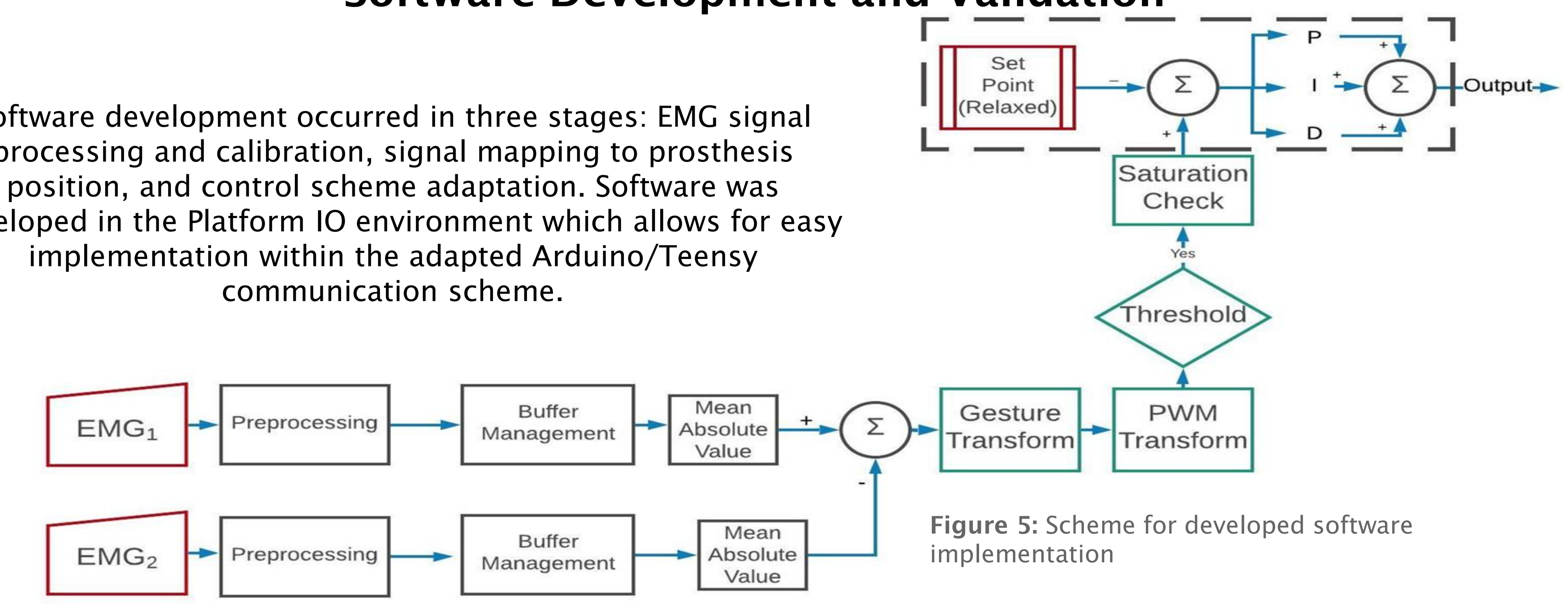


Figure 5: Scheme for developed software implementation

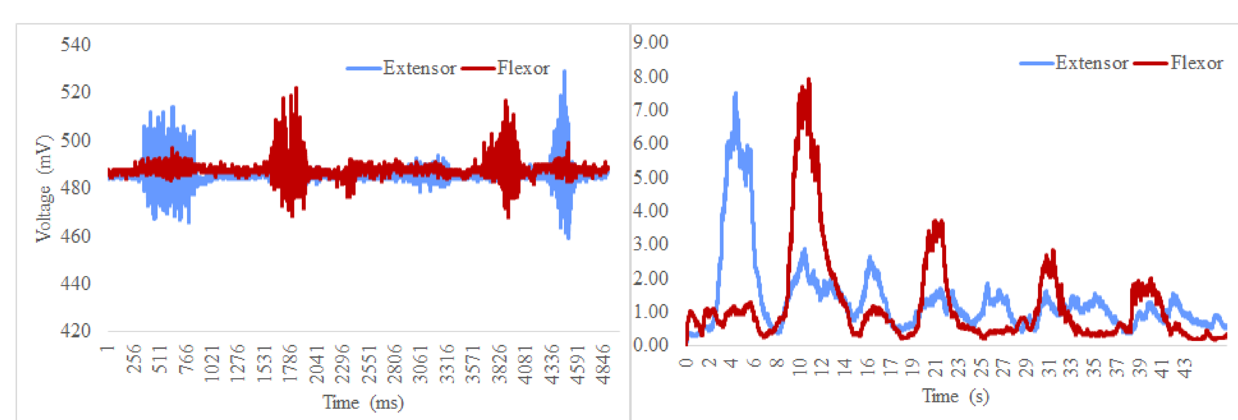


Figure 6: (A) Example raw signal output from EMG hardware (B) Example of preprocessed and rectified waveform obtained from developed software

Signal preprocessing was integrated into the developed software to allow for signal manipulation downstream. SNR was maintained at approximately 46 dB for extension and 37 dB for flexion with a large decrease in variance across subjects following developed preprocessing.

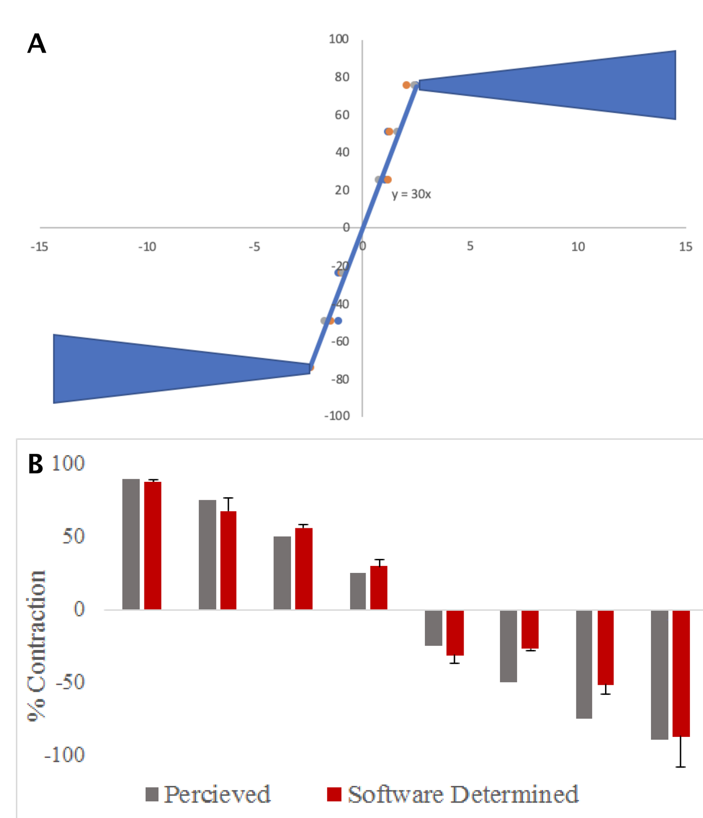


Figure 7: A) Comparison of contraction intent as perceived by the user and determined by the software (n=3) B) Graph of the developed gesture classifier (n=9)

The gesture classification was determined using human test subjects to allow for the translation of percent contraction to mean absolute value of incoming EMG differential. This mapping is constant across the central region, but is calibrated to the patient's maximum contractions for both flexion and extension.

The actuation of the developed software was tested using a servo motor through the entire scheme of the software system; EMG intake and processing, gesture mapping, motor communication.

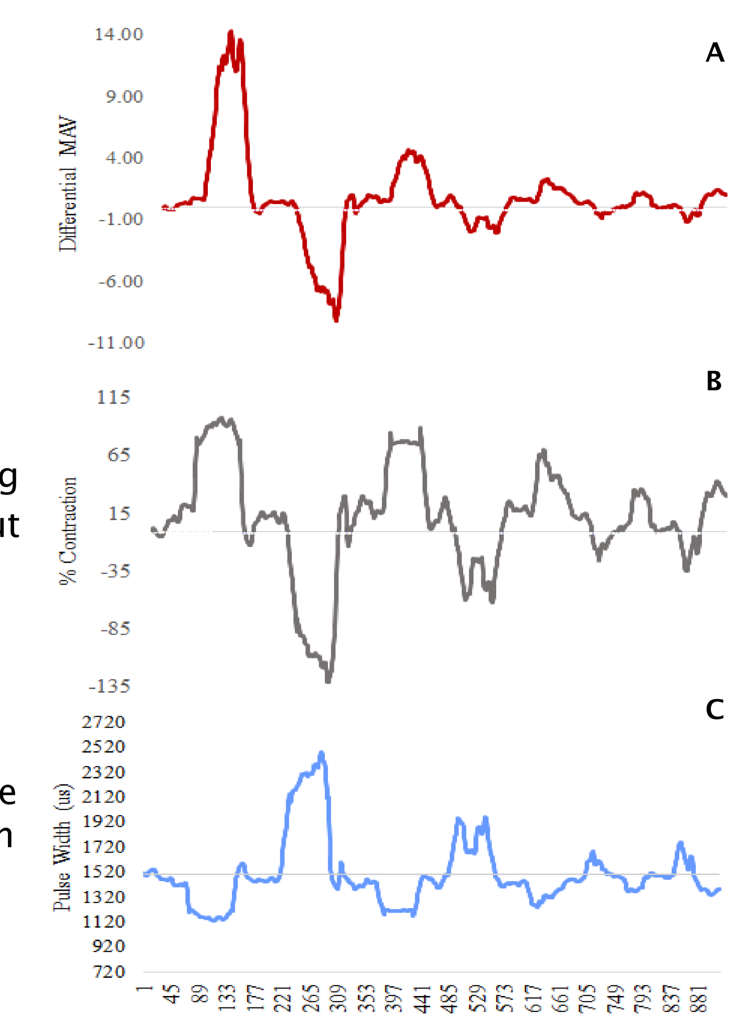


Figure 8: The developed software output for (A) differential MAV, (B) gesture classifier determined percent contraction, and (C) mapped pulse widths for actuation with varying muscle contraction

Hardware

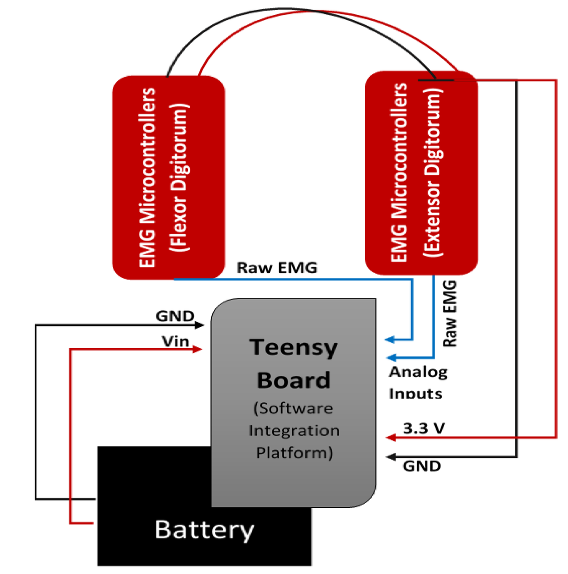
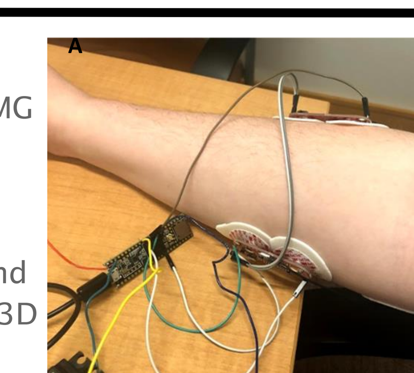


Figure 9: This scheme demonstrates the hardware implementation independent from LED shields as will be seen on a phantom subject

The hardware implemented included two EMG microcontrollers (MyoWare), a 32 bit 180 MHz ARM Cortex-M4 processor Teensy 3.6, a 4.5 V battery pack, and plastic cage. Each component was chosen and designed to minimize cost and space while maximizing wearability and durability. The total cost of hardware was $115.

Figure 10: A) Example of EMG electrode placement B) Image of prosthetic hand and attached 3D printed cage



Clinical Integration

Figure 11: Sample PID modulation of pulse width modulation servo adaptation

The developed PID controller allows for the maintenance of contraction to result in a pulse width output of zero. This would maintain position within the current prosthesis scheme. The PID was manipulated to allow validation with the servo motor and will require calibration once integrated with working prosthesis.

This system was designed for easy integration with current prostheses. It is intended to communicate directly to the prosthesis microcontroller using pulse width modulation.

Figure 12: Image of Ottobock myoelectric prosthetic hand with integration points highlighted

Conclusions

- Software that converts traditional velocity-based prosthetic devices to force-based operation has been successfully developed.
- Software and hardware components necessary for proof of concept were combined and integrated.
- The software was successfully implemented on a servo motor with the ability to be translated to a myoelectric prosthetic hand.
- System has been proven to accurately extract the intent of the user's muscle activity and translate it to pulse widths.
- The project is being given to the FNI lab for implementation in-home trials.

Next Steps

- Calibrate PID for prosthetic hand integration and implement software packages with prosthesis.
- Adjust thresholding and advisory controls in future iterations.
- Verify and use system to complete quantitative in-home trials of the neural-connected sensory prosthesis system.

Acknowledgements

We would like to acknowledge the Functional Neural Interface Lab and Case Western Reserve University for allowing us to join their team, specifically our advisor Dr. Emily Grazyck. Special thank you to Dr. Thomas Withrow, Dr. Michael Goldfarb, and Dr. Matthew Walker III for support during the design process and to Alvaro Villaslada for granting permission for adaptation of Myoware communication source code.

References

[1] Research. (n.d.). Retrieved from http://engineering.case.edu/ebme/tyler/research

[2] Chadwell, Alix, et al. "The Reality of Myoelectric Prostheses: Understanding What Makes These Devices Difficult for Some Users to Control." *Frontiers in Neurorobotics*, vol. 10, 22 Aug. 2016, doi:10.3389/fnbot.2018.00015.

Force- Based Microcontroller Design and Documentation Website