BRACHIOPLEXUS: MYOELECTRIC TRAINING SOFTWARE FOR CLINICAL AND RESEARCH APPLICATIONS

Michael R. Dawson[1,4], Heather E. Williams[2], Glyn S. Murgatroyd[3], Jacqueline S. Hebert[1,3], Patrick M. Pilarski[1,4]

[1]Division of Physical Medicine and Rehabilitation, University of Alberta, Canada; [2]Department of Biomedical Engineering, University of Alberta, Canada; [3]Glenrose Rehabilitation Hospital, Alberta Health Services, Canada; [4]Alberta Machine Intelligence Institute, Canada

ABSTRACT

Various control strategies are now available for myoelectric devices. The selection of the most appropriate strategy for an individual patient and training to improve their skills are important components to optimize user function with their myoelectric prosthesis. Existing myoelectric training software is often limited by not providing enough features to allow prosthesis users to try the multiple options for prosthetic hands, wrists, and elbows and the various control strategies used to modulate or switch between them. To address this gap, we developed an open-source training software for clinical and research applications called brachI/Oplexus that aims to provide a wider breadth of options and also be easy to use by non-technical users. The software supports several input devices (EMG systems), output devices (robotic arms), and methods for mapping between them (conventional and machine learning controllers). A comparison was performed between brachI/Oplexus and two commercial myoelectric software programs. Results from the testing showed that brachI/Oplexus had similar or slightly improved EMG signal separation and delay when compared to the commercial software. Several research labs and hospitals are already using this software, and by releasing it open source, we hope to lower the barrier of entry and encourage other clinicians and researchers to explore this area.

INTRODUCTION

Training prior to provision of a definitive device plays an important role during the fitting process for an upper limb myoelectric prosthesis. Not only does it allow a person with amputation to improve their skills at using the technology, but it allows them try various options for prosthetic devices and control strategies. One of the main limitations of the existing training systems is that they typically only allow for training of a single prosthetic device with a limited set of mapping options [1]. To allow a person with amputation to try the full breadth of devices and mapping options, it is necessary to borrow training systems from multiple manufacturers, which can be time consuming and logistically difficult. In order to solve this issue, we previously developed our own custom training system, called the Myoelectric Training Tool (MTT) [2], which included an electromyography (EMG) acquisition system, control software, and a desktop robotic arm with 5 degrees of freedom (DoFs) including shoulder rotation, elbow flexion/extension, wrist rotation, wrist flexion/extension, and hand open/close. This system has been successfully used for research at the University of Alberta since 2011 and for clinical training at the Glenrose Rehabilitation Hospital since 2015.

An upgraded version of the robotic arm, called the Bento Arm, was developed in 2013 [3,4] with stronger servos that included integrated sensors for measuring position, velocity, temperature, voltage and load. The Bento Arm also was designed to have a more anatomical appearance and could be mounted to a desktop stand or worn as a prosthesis by attaching it to a socket. In 2017, we released a compatible sensorized, multi-articulated hand, called the HANDi Hand [5,6] which included flexion of all digits as well as thumb rotation. Initial studies employing these devices used software based out of Robot Operating system (ROS) or MATLAB’s Simulink Realtime (SLRT) Operating System. While these software platforms worked well in a research environment we found they were difficult to translate into clinical environments, as they could take several hours to install on new computers and were not easy to operate by researchers or clinicians coming from non-technical backgrounds.

In order to improve the accessibility of our software and its applicability to both clinical and research environments, we developed a new version based on lessons learned from our prior research and clinical deployments. The main objectives of the new software were that it be easy to install and use, and allow for mapping between a wide array of input devices and robots. We also decided to make the software open source, to facilitate use by other research groups and hospitals.

Taking inspiration from the anatomical term ‘brachial plexus’, which is the main network of nerves that connects the brain and spinal cord to the arm, we named the software brachI/Oplexus (pronounced “brak-I-O-PLEX-us”), with the goal of providing an improved digital nerve center for connecting input devices to robotic arms.

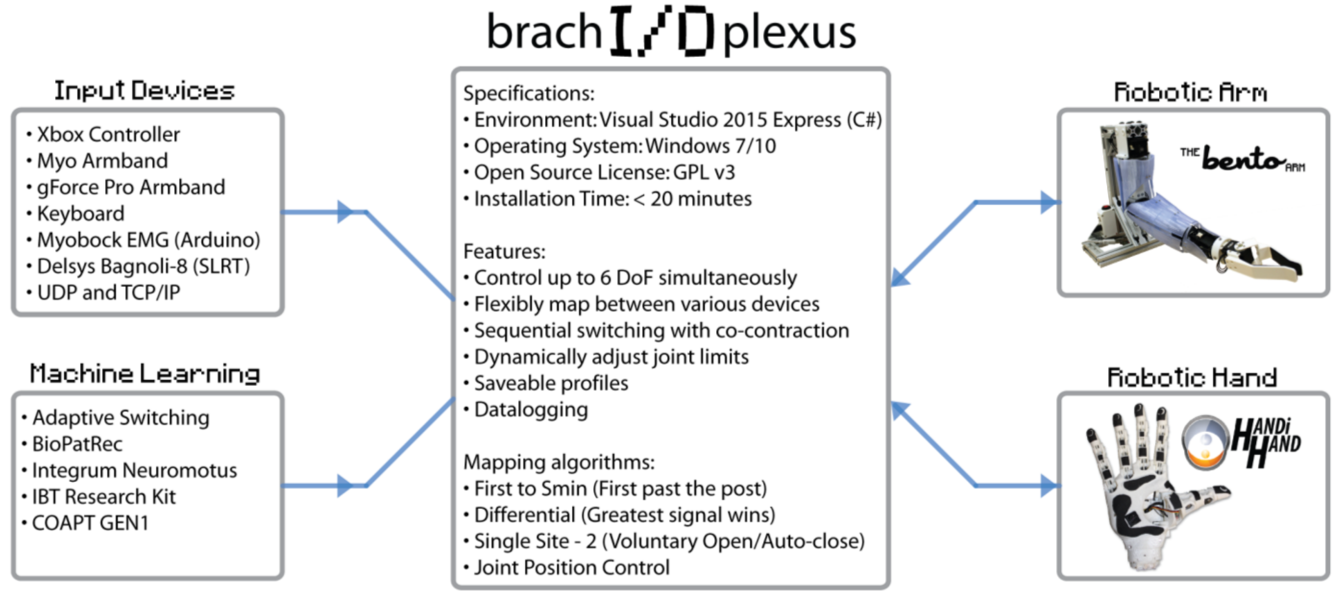

Figure 1: A high level software diagram showing how brachI/Oplexus takes in signals from input devices or external machine learning software and then maps them to joint positions or velocities on robotic devices.

SOFTWARE DESIGN

The main specifications and features of the brachI/Oplexus software are summarized in Figure 1. Initial requirements were determined by gathering feedback from clinical stakeholders, including clinicians, researchers, and persons with amputation. As part of this process, the developers of the software also gained valuable insight into possible improvements by directly observing over 50 myoelectric training sessions with clinicians and patients with amputation.

The development environment was selected based on criteria including timing performance, ease of use for both programmers and non-programmers, availability of libraries/interfaces, compatibility with existing software/hardware, and cost. Visual Studio 2015 Express using the C# language was chosen as it is free to use by researchers, students, and hobbyists and provides advanced tools for building graphical user interfaces and easy to use installer packages. Using the installer package combined with automatic driver installations from Windows Update, the entire software can be installed on a new computer and operational in less than 20 minutes. Initial performance testing in C# showed that we could achieve the desired step times of 200 Hz or faster for acquiring sensor data and sending motor commands to the Bento Arm.

To make brachI/Oplexus more accessible to external users, we released it open-source under a General Public License (GPL) v3 license, which allows the software to be freely used for both commercial and non-commercial purposes. As part of the open-source release, we developed extensive documentation explaining how to install and operate the software [7].

One of the key features of the software is the ability to control up to 6 DoFs simultaneously. This is especially useful when evaluating whether persons with transhumeral amputation are able to operate a multifunctional prosthesis. Additional DoFs that are not being evaluated for possible use in the definitive prosthesis, but that are useful for completing an engaging training task (i.e. shoulder rotation), can be controlled by the intact hand with the Xbox controller/keyboard or automated by the clinician.

Another key feature is the ability to flexibly adjust the mapping while the software is running without having to recompile the code or change settings in configuration files. All the required controls for selecting an input device, output device, and mapping algorithm are available through the graphical user interface as seen in Figure 2. Any combination of input device signals can be used to create multi-device mappings, and the signal settings including the minimum and maximum thresholds and gains are adjustable. The joint limits of the Bento Arm, including joint positions, velocities, and load can also be modified. These adjustments can be used to create movement envelopes to adjust the difficulty of tasks, and to improve the safety of the robot by limiting the torque and movement when in the same workspace as the operator (i.e. when the devices are being worn as prostheses). All of these settings are saveable as profiles, so that they can be easily reloaded at a later time. We find this feature is useful when seeing patients over multiple sessions in order to track their EMG settings and provide starting points for future sessions.

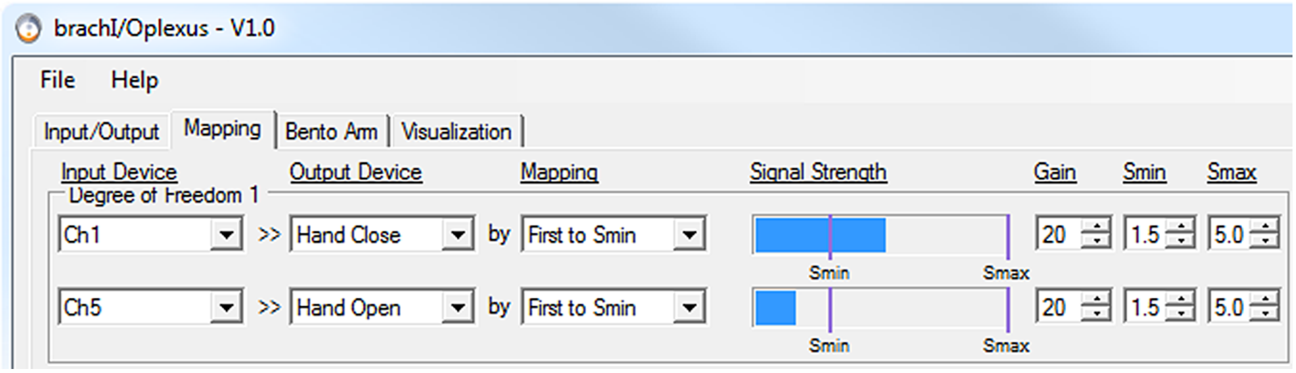

Figure 2: A screenshot of the mapping tab in brachI/Oplexus showing the adjustable settings for the 1st Degree of Freedom.

A data logging module was designed that can be used to log time series data from the input signals or feedback from the robots to a text file. In addition to specifying which signals to log, the sample rate can also be set to as fast as 200 Hz or as slow as desired. The data logging functionality can be used as part of research studies to record study variables and also as part of clinical training for logging training metrics.

The supported input devices are listed in Figure 1. Whenever possible existing open-source developer kits, libraries, and interfaces were used to connect to external hardware. The Microsoft Xbox controller and keyboard input devices are typically used for testing and demonstration purposes or as mentioned previously when controlling DoFs that are not yet available on commercial prostheses. Two different wireless armbands with EMG capabilities are supported including Thalmic Lab's Myo Armband (now discontinued, but still widely used) and Oymotion's gForce Pro Armband. We have found these armbands useful for demonstration purposes or for use with persons with transradial amputations. The Simulink Realtime interface (SLRT) is used to acquire signals from a Bagnoli-8 EMG Acquisition system with DE 3.1 electrodes (Delsys, Inc.) via a PCI 6259 data acquisition card (National Instruments, Inc.). The Arduino module is used to acquire signals from Myobock 13E200 electrodes (Ottobock, Inc.) via a custom designed wireless and battery-operated Arduino board. Both the SLRT and Arduino modules also allow for external buttons, FSRs, and EMG systems to be connected to the software. The UDP and TCP/IP network interfaces allow for external software to drive the arm. This is most commonly used for controlling the arm using machine learning software such as Adaptive Switching [8], BioPatRec [9], COAPT GEN1 (COAPT, LLC), IBT Research Kit (Infinite Biomedical, LLC), and Neuromotus (Integrum, Inc.). Several packet structures have been pre-designed to facilitate developing communication modules between brachI/Oplexus and new software. Another benefit of these network modules is that they can also communicate with Linux or macOS based operating systems.

The supported output devices include the Bento Arm and HANDi Hand which can be used together or separately in the software. For the Bento Arm, we communicate with the sensorized Dynamixels servos using the Dynamixel SDK, and for the HANDi Hand, we use an Arduino Mega to control its RC servos. There is also an option to directly control commercial myoelectric prehensors via the analog output on the Arduino or SLRT computer.

Several different mapping algorithms have been developed for brachI/Oplexus based on common control strategies available in commercial myoelectric software. Thus far, we have implemented control strategies for first to smin (also known as first pass the post), differential (also known as greatest signal wins), and single site-2 (voluntary open/automatic close). These are meant to serve not only as possible options for clinicians/patients using the software for clinical training, but also in research studies to provide a baseline to compare conventional controllers to machine learning controllers.

Also included in the software is a sequential switching module that allows for switching between up to 5 DoFs on the Bento Arm using a single pair of input signals. The switching can be triggered by exceeding a threshold with a dedicated signal or by co-contracting two signals together within a specified time frame. Several different feedback options are available to indicate to the user when they have switched and what DoFs they have switched to including visual, auditory, and vibratory feedback.

SOFTWARE COMPARISON RESULTS

We compared the signal processing and mapping algorithms in brachI/Oplexus to two commercial software programs. The comparison software included biosim v2.11.0.38 used with the i-Limb ultra revolution hand (Ossur, Inc.), and bebalance Version 3.5c used with the bebionic hand (Ottobock, Inc.). This study was approved by the Research Ethics Board of the University of Alberta (Pro00077893).

Table 1: EMG ratios and delays.

	Flexion agonist-to-antagonist	Extension agonist-to-antagonist	Delay (ms)
biosim + i-Limb + Myobock electrodes	4.0 ± 0.9	4.9 ± 1.4	151.7 ± 22.8
bebalance + bebionic + Myobock electrodes	3.2 ± 1.0	4.0 ± 0.6	150.1 ± 31.2
brachI/Oplexus + Bento Arm + Delsys Bagnoli electrodes	10.7 ± 5.9	12.8 ± 4.8	71.7 ± 13.2
brachI/Oplexus + Bento Arm + Myo Armband	6.3 ± 1.6	3.9 ± 0.6	142.6 ± 21.0

To compare the EMG performance across software, the rectified and averaged signals of 10 wrist flexion movements and 10 wrist extension movements were recorded from an able-bodied participant for each condition as specified in Table 1. The agonist-to-antagonist ratios were calculated for each of these movements, and average ratios and respective standard deviations are shown in Table 1. Larger agonist-antagonist ratios were observed with brachI/Oplexus with the Delsys Bagnoli electrodes, indicating that the signal separation might be a bit higher compared to the commercial myoelectric software.

To compare the delays observed with brachI/Oplexus to those observed with commercial devices, an iPhone 11 high speed camera (240 fps) was used to identify the time between when wrist extension was initiated and when the respective robotic hand began to open, for 10 repetitions. The average delays and respective standard deviations are shown in Table 1. The Bento Arm with brachI/Oplexus generally exhibited smaller delays than the i-Limb and bebionic hands, especially when paired with the Delsys Bagnoli electrodes.

Although there were some differences between brachI/Oplexus and the commercial software, in practise all of the systems behaved in a similar manner. If required, brachI/Oplexus also has the flexibility to mimic the commercial systems even more closely by adding a small delay to the motor commands or by using the Myobock electrodes with the Arduino module instead of the alternative EMG systems.

FUTURE WORK & CONCLUSIONS

Thus far, brachI/Oplexus has been used by 3 university research labs and 2 rehabilitation hospitals with other deployments ongoing. In the future we would like to add more input devices and output devices to the software to further improve its capabilities, so that it can be useful to even more clinicians or researchers. For example, we are in the process of creating a virtual reality version of the Bento Arm that would allow for training or research to take place in situations where it may not be practical to deploy physical robots (i.e. take-home training).

In conclusion, brachI/Oplexus is a fully functional myoelectric software with support for many different kinds of input devices, conventional and machine learning based controllers, and robotic arms. The wide array of features, improved accessibility, and dynamic adjustability make it well suited for clinical training and assessment as well as research applications.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support from the Alberta Machine Intelligence Institute (Amii), the Natural Sciences and Engineering Research Council (NSERC), Canada Chairs Research Program, and the Glenrose Rehabilitation Hospital. Thanks as well to Michael Stobbe for his feedback on commercial myoelectric software and to Dylan Brenneis, McNiel-Inyani Keri, James Austin, Ahmed Shehata, and Eric Wells for their contributions to brachI/Oplexus feature development.

REFERENCES

[1] M.R. Dawson, J. P. Carey, F. Fahimi, “Review of Myoelectric Training Systems,” *Expert Reviews of Medical Devices*, Vol.8, No.5, Pages 581-589, 2011.

[2] M.R. Dawson, F. Fahimi, J. P. Carey, “The Development of a Myoelectric Training Tool for Above-Elbow Amputees,” *The Open Biomedical Engineering* Journal, 6:5–15, 2012.

[3] M.R. Dawson, C. Sherstan, J.P. Carey, J.S. Hebert, P.M. Pilarski, “Development of the Bento Arm: An Improved Robotic Arm for Myoelectric Training and Research,” *Proc. of MEC'14*, Fredericton, New Brunswick, August 18-22, 2014, pp. 60-64

[4] BLINCdev, “BLINCdev/Bento-Arm-Hardware,” [Online]. Available: https://github.com/blincdev/Bento-Arm-Hardware [Accessed Feb.18, 2020]

[5] D. J. A. Brenneis, M. R. Dawson, P. M. Pilarski, “Development of the HANDi Hand: An Inexpensive, Multi-Articulating, Sensorized Hand for Machine Learning Research in Myoelectric Control,” *Proc. of MEC'17*, Fredericton, New Brunswick, August 15-18, 2017.

[6] BLINCdev, “BLINCdev/HANDi-Hand,” [Online]. Available: https://github.com/blincdev/HANDi-Hand [Accessed Feb.18, 2020]

[7] BLINCdev, “BLINCdev/brachIOplexus,” [Online]. Available: https://github.com/blincdev/brachIOplexus [Accessed Feb.18, 2020]

[8] A. L. Edwards, M. R. Dawson, J. S. Hebert, C. Sherstan, R. S. Sutton, K. M. Chan, P. M. Pilarski, “Application of Real-time Machine Learning to Myoelectric Prosthesis Control: A Case Series in Adaptive Switching,” Prosthetics & Orthotics International. vol. 40, no. 5, 573–581, 2016.

[9] M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, “BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms,” *Source Code Biol. Med.*, vol. 8, no. 1, p. 11, 2013