MyoHMI: A Low-Cost and Flexible Platform for Developing Real-Time Human Machine Interface for Myoelectric Controlled Applications

Ian Donovan*, Kevin Valenzuela*, Alejandro Ortiz*, Sergey Dusheyko*,

Hao Jiang*, Kazunori Okada**, Xiaorong Zhang*

* School of Engineering, San Francisco State University, San Francisco, USA

** Department of Computer Science, San Francisco State University, San Francisco, USA

xrzhang@sfsu.edu

***Abstract*—EMG pattern recognition has been studied for control of prostheses and rehabilitation systems for decades. Existing research platforms for developing EMG pattern recognition algorithms are typically based on MATLAB and the collection of EMG signals is often done by expensive, non-portable data acquisition systems. The requirement of these resources usually limits the use of these platforms in the lab environments and prohibits their widespread to other fields and applications. To address this limitation, this paper presents a low-cost, easy to use, and flexible platform called MyoHMI for developing real-time human machine interfaces for myoelectric controlled applications. MyoHMI facilitates the interface with a commercial EMG-based armband Myo, which costs less than $200 and can be easily worn by the user without the need of special preparation. MyoHMI also provides a highly modular and customizable C/C++ based software engine which seamlessly integrates a variety of interfacing and signal processing modules, from data acquisition through signal processing and pattern recognition, to real-time evaluation and control. The experimental results on able-bodied human subjects for controlling two evaluation platforms in real time verified the merit of the MyoHMI platform and demonstrated the feasibility of a low-cost solution for the development of myoelectric controlled applications.**

I. INTRODUCTION

Surface electromyographic (sEMG) signal is a bio-signal that measures muscle activities by using electrodes placed on the skin and has been found effective for representing movement intentions [1]. Within the last decade, the processing and pattern recognition of sEMG signals have been widely studied for myoelectric control of prosthetic and rehabilitation applications, such as EMG-based power-assisted wheelchair [2], human-assisting robotic treatment for stroke rehabilitation [3], and myoelectric controlled prosthetic limbs [4-7]. Most of these researches have been limited in the lab environments and the measurement of EMG signals has typically been done by experts in the field using expensive non-portable data acquisition systems. With the advances of embedded computer systems and electronics technology, low-cost, wearable myoelectric control systems which seamlessly integrate EMG sensing, embedded processing, and wireless interfacing in a stand-alone portable device have become available. One commercially available solution is the Myo gesture control armband (Thalmic Labs) which was released in 2014 with the price of less than 200 dollars. Recently, the Johns Hopkins University employed two Myo armbands to control a powered prosthetic arm and tested it on an amputee subject for performing a few hand motions successfully [8]. Moreover, the release of such inexpensive, easy-to-wear device has attracted great attentions and unlocked tremendous possibilities in many other fields for advanced human machine interface (HMI) applications, such as smart control of home appliances [9], gesture-based music interaction [10], mobile device interface [11], and control of virtual reality (VR) applications [12-15]. However, without the expertise in EMG signal processing and pattern recognition, most of these systems were constrained by only using the five default gestures (fist, fingers spread, wave left, wave right, and double tap) provided by the manufacturer without any application specific adaptation. These default gestures are neither re-trainable nor customizable. The gesture recognition performance on users varies a lot from person to person. These factors greatly limit the systems' functionality and usefulness.

To address these limitations, this project aims to provide a low-cost easy-to-use and flexible platform called MyoHMI for developing real-time HMIs for myoelectric controlled systems. In literature for prosthetic control research, similar software platforms have been proposed by a few institutions for developing and evaluating EMG pattern recognition algorithms. Such platforms include the Acquisition and Control Environment (ACE) from the University of New Brunswick [16], the Virtual Integration Environment (VIE) developed by the Johns Hopkins University Applied Physics Laboratory [17], the Control Algorithms for Prosthetic Systems (CAPS) from the Rehabilitation Institute of Chicago [5], and the open source research platform BioPatRec provided by Chalmers University of Technology [18]. All these platforms are developed based on MATLAB and usually require the installation of several MATLAB

This work is partly supported by the San Francisco State University (SFSU) Ken Fong Translational Research Fund and the SFSU Center for Computing for Life Sciences (CCLS) Mini Grant. Corresponding author: Xiaorong Zhang (email: xrzhang@sfsu.edu).

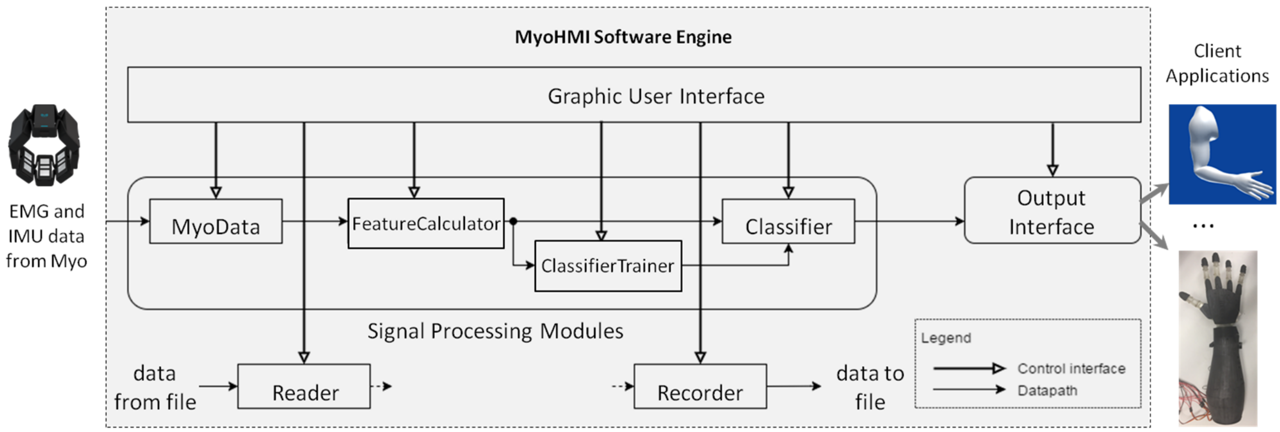

Figure 1. Overall Architecture of the MyoHMI platform.

toolboxes for data acquisition, data analysis, etc. In addition, the data acquisition systems interfaced with these software platforms are typically expensive, professional, non-portable EMG systems that cost tens of thousands of dollars. The requirement of these resources may prohibit the widespread of these platforms to other HMI fields and applications.

In this work, we develop MyoHMI with the aim of providing a low-cost solution to facilitate collaborations in the HMI community, and accelerate the development of more and better myoelectric controlled systems that can potentially benefit our society and improve the quality of life. MyoHMI is a highly modular and customizable platform that takes input from up to two Myo armbands and seamlessly integrates a variety of interfacing and signal processing modules, from data acquisition through signal processing and pattern recognition, to real-time evaluation and control. A friendly graphic user interface (GUI) is developed for the users to easily access all the functionalities provided by the platform, as well as visualize the input signals, intermediate features, and output results in real time. MyoHMI is developed based on C/C++ and can be easily installed on any computers without the requirement of any licensed software or toolboxes. The total cost of using MyoHMI for real-time HMI development is just the cost of the Myo armbands. In addition, we developed two evaluation platforms: a 3D-printed prosthetic hand and a VR system that displays a virtual human hand as part of the MyoHMI package to verify the functionality of the MyoHMI software for real-time myoelectric control. Currently efforts are underway to make the MyoHMI system as well as the evaluation platforms open source.

II. System Design and Implementation

A. MyoHMI Software Engine

Figure 1 shows the overall architecture of the MyoHMI platform. The MyoHMI software engine consists of four major components: *signal processing modules*, *output interface*, *file interaction modules*, and *GUI*. The *signal processing modules* take input signals from the Myo armbands and process the signals to identify the user's motions using pattern recognition methods. The *output interface* converts the output decisions into control signals for controlling external applications such as prostheses and VR systems. The *file interaction modules* include a *Reader* and a *Recorder*, which assist with the HMI experimentation procedure by communicating with files. A layer of *GUI* separates the end-user from the backend signal processing and output communication. It enables easy access and flexible configuration to all the functionalities provided by other components of the MyoHMI software engine.

Signal Processing Modules

The signal processing modules are implemented as a modular system responsible for the collection, processing, and classification of the Myo data. An object-oriented approach is used in consideration of variability in the data sources, the output clients, and the classification algorithms. All the modules are subclasses of an abstract class and inherit the ability to receive incoming data and output data to another module. The complex task of pattern recognition is broken down as a series of transforms. Each module is tasked with a specific transform.

Data Collection Module – MyoData: The *MyoData* module manages the connection with and the data collection from Myo armbands. Specifically, each Myo armband has eight EMG sensors, each of which streams a unique signal at 200 Hz. It also has an integrated inertial measurement unit (IMU) including three-axis accelerometer, three-axis gyroscope, and three-axis magnetometer with each channel streaming data at 50 Hz. The *MyoData* module identifies available Myo devices through Bluetooth Low Energy (BLE). Upon successful connection, the multi-channel EMG and IMU signals are streamed to the *MyoData* module wirelessly and can then be forwarded to any modules connected to it.

Feature Calculation Module – FeatureCalculator: The *FeatureCalculator* module receives raw EMG and IMU data from the MyoData module, segments the input signals into overlapped sliding analysis windows, and then extracts a variety of features that represent the signal characteristics from each window. To recognize the user's motion, features of individual input channels are concatenated into one feature vector and then sent to the classification modules for motion recognition. Currently four time-domain features have been implemented in MyoHMI, including mean absolute value (MAV), waveform length (WL), number of zero crossings (ZC), and number of slope sign changes (TURN) [19]. Additional time-domain, frequency-domain, and time-frequency-domain features are under development and will be added to the system in future. The length and increment of the analysis window, as well as the selection of

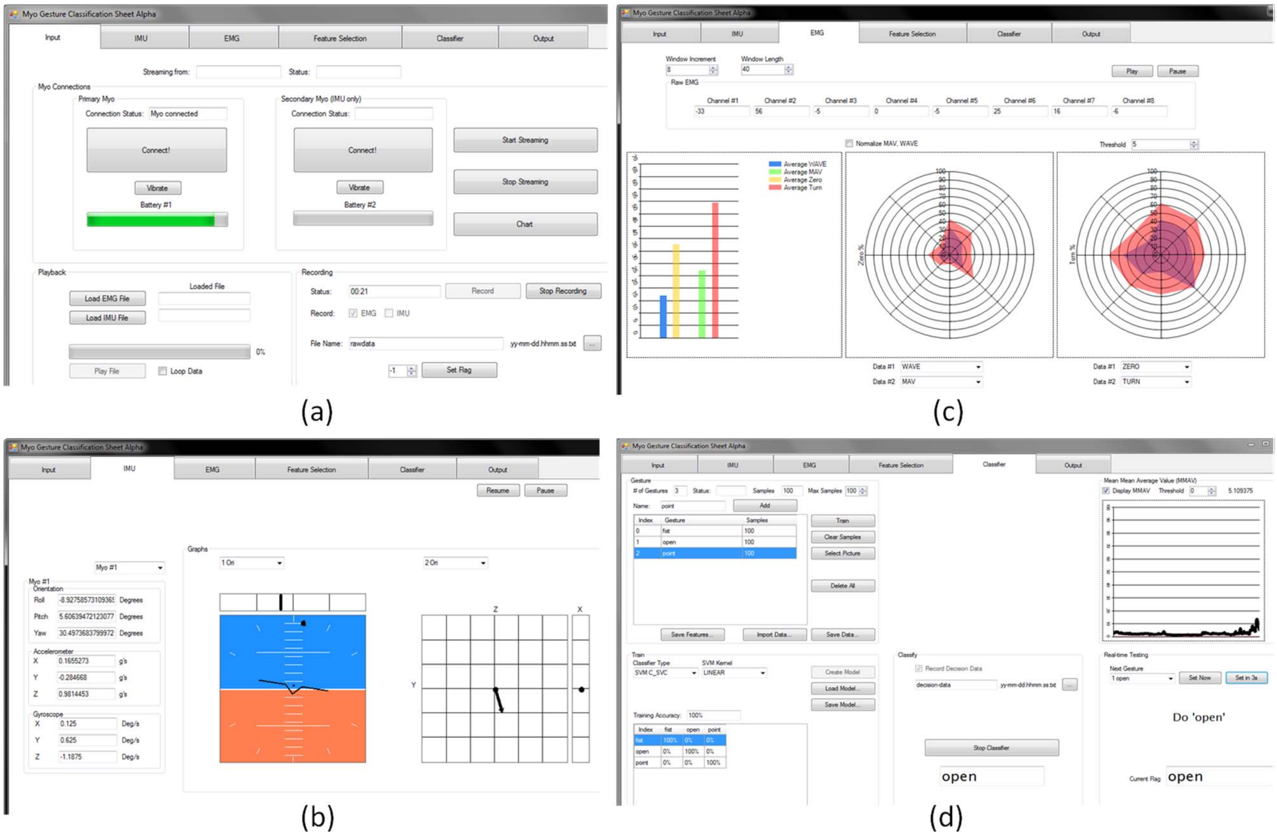

Figure 2. The graphic user interface of the MyoHMI platform. (a) Input tab; (b) IMU tab; (c) EMG tab; (d) Classifier tab.

features are fully configurable by the user through the GUI.

Classification Modules – ClassifierTrainer and Classifier: Pattern classification algorithms generally consists of two phases: training and testing. During the training phase, a set of labeled feature vectors from each investigated motion are trained to create a classification model that maximally distinguishes the patterns of different classes. In the testing phase, the feature vector extracted from new incoming signals is sent to the trained classification model for a decision of the identified class, one decision per analysis window The *Classifier* module provides an abstract class which defines common interface methods for training ("*train*") and testing ("*predict*"). *ClassifierTrainer* is a simple module that collects and labels feature vectors from the *FeatureCalculator* module and organizes the training data set to be sent to the *Classifier* module. This module is only active during the training phase and will be bypassed in the testing phase. Using the same interface standard, a variety of classification algorithms can then be implemented and easily managed. At the current stage, two classification algorithms have been implemented in MyoHMI, including linear discriminant analysis (LDA) and support vector machine (SVM) because of their previous success in real-time EMG pattern recognition [6, 20]. The implementation of the SVM algorithm was based on LIBSVM, an open-source SVM-based library [21].

Output Interface

The *output interface* receives classification decisions from the *Classifier* module and broadcasts the decisions to any external client applications such as prostheses and VR systems. It also provides functions to convert the motion decisions into emulated peripheral actions, such as keyboard press for controlling external devices.

File Interaction Modules – Reader and Recorder

The *file interaction modules* include a *Reader* and a *Recorder*, which assist with the HMI experimentation procedure by communicating with files. Objects of these two modules can be easily connected to any signal processing module. The *Recorder* writes incoming data of a module to a file and the *Reader* acts as a source by reading from a file. The data flow at every stage of the system can be viewed in real-time or played back anytime afterwards for further analysis.

GUI

The friendly GUI is implemented based on C++/CLR and Microsoft .NET Framework. The designed GUI organizes its features into six tabs: *Input*, *IMU*, *EMG*, *Feature Selection*, *Classifier*, and *Output*. The software was written with flexibility in mind, so that additional methods of feature extraction, classification can be easily integrated in the future.

The *Input* tab, as shown in Figure 2a, provides information relating to the Myo device configuration and connection options. The system can connect to up to two Myo devices. Information such as the battery level and connection status of the devices is displayed. This tab also provides functions to start/stop real-time data collection, save the data into files, and playback existing data files.

The *IMU* tab (Figure 2b) displays data relating to the IMU measurements in real-time. It visualizes the orientation measurements using an attitude and heading indicator and the accelerations in a vector graph. In addition to the graphs, there is a column of boxes that displays raw IMU data. The IMU tab is most helpful when performing a dynamic gesture that is dependent on the movement of the Myo armband.

The *EMG* tab (Figure 2c) is used to examine EMG data and features in real-time and provides visual feedback to potentially help the user adjust the way of performing gestures, as well as the parameters for feature calculation. The raw EMG measurements from each individual channel are displayed in text boxes. The calculated EMG features are visualized in real-time using radar and bar graphs. The radar format provides an easy way of analyzing eight channels of EMG features relative to the location of the sensors on the arm. The length and increment of the analysis windows can be adjusted by the user to manipulate feature calculation.

The *Feature Selection* tab and the *Classifier* tab (Figure 2d) allow the user to select features and classification methods for motion recognition. The *Classifier* tab can be used to record a custom set of gestures and use the selected features and classification methods to train the classifier and do motion recognition in real time. The user can easily name each gesture and choose where to save the data. This tab also contains a live monitor of the averaged MAVs of the eight EMG channels to provide visual feedback for the user to better perform the gestures.

B. Design of the 3D Printed Prosthetic Hand

In this work, two evaluation platforms have been developed as part of the MyoHMI package to verify the functionality of the MyoHMI software engine for real-time myoelectric control: a 3D printed prosthetic hand and a VR system. The design of the 3D printed prosthetic hand, as shown in Figure 3, was taken from the open source InMoov robot project [22] because of its many degrees-of-freedom (wrist rotation and five independent finger motions), low development overhead, and low cost. The device features five HK15298B Servo motors for finger actuation and one MG996r Servo motor for wrist actuation. The motors are driven by pulse width modulation commands generated from a microcontroller, which receives motion commands from the *output interface* of the MyoHMI software engine. Pulleys affixed to finger motors, which are housed in the forearm, drive artificial tendons connected to the finger tips. The wrist compartment contains a gear set coupled to the 3D printed hand on one side and the MG996r servo on the other. The process of printing the arm mechanism spanned totally 50 hours across several 3D printing machines. A total of 60 parts and sub assemblies were printed. Once all parts were printed they were processed and assembled. Post processing consisted of gluing, deburring, sanding, drilling, and fit adjustment.

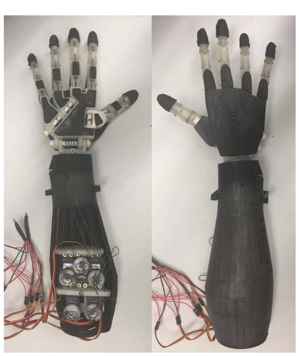

Figure 3. The 3D printed prosthetic hand.

C. Design and Implementation of the VR system

The VR system was implemented using the game development software Unity and the 3D graphics as well as animation software Blender. Specifically, a virtual human arm was developed to perform hand and arm motions received from the MyoHMI software engine as shown in Figure 4. A skeletal system with armature bones was developed using Blender and embedded into the virtual arm to perform various motions by rotating the armature bones. As shown in Figure 5, the armature bones were placed in a formation resembling a human arm skeleton. Each single bone can perform 3-axis rotations. The middle and distal phalanges of the virtual arm's individual fingers can have local rotations to the proximal phalanx, which allows more natural hand motions. Each defined motion has a specific animation file, which will be played when the corresponding motion command is received from the MyoHMI output interface. A finite state machine (FSM) was developed in Unity to enable smooth transitions between different motions. If a new motion command is received while the animation of the previous motion is still not finished, the previous animation will be interrupted, and the animation of the new motion will begin.

Currently only discrete motion commands from the MyoHMI software engine are received by the VR system. Developments are underway to integrate speed control, IMU measurements, and feedback mechanism, which would enhance the VR system to perform more complex tasks (Figure 6).

Figure 4. Hand motions played by the VR system. From left to right, top to bottom: Rest, Fist, Point, Open-Hand, Supination, and Pronation.

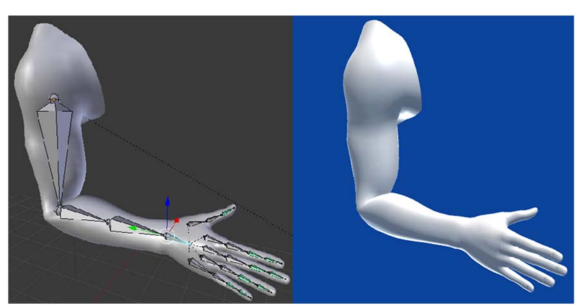

Figure 5. Left: the virtual arm skeleton system with embedded armature bones (Blender). Right: the virtual arm mesh in Unity.

III. Experimental Results and Discussion

This study is conducted with Institutional Review Board (IRB) approval at San Francisco State University and informed consent of subjects. To verify the functionality of the MyoHMI platform, real-time experiments were conducted on four able-bodied human subjects for hand motion recognition. The output decisions from the MyoHMI software engine were then used to control the 3D printed prosthetic hand and the virtual arm in real time.

A. Experimental Protocol

Five classes of hand motion were investigated in the experiments including *open*, *fist*, *point*, *supination*, and *pronation*. This set of classes was selected mainly because of the gesture performing capability of the 3D printed prosthetic hand. The *no motion* class was excluded because the intent was to use the average of the mean absolute values (MMAV) across all channels as a value for representing intensity for speed control of the prosthetic hand. Therefore if the MMAV is below a threshold the output decision will be *no motion*. During the experiment, the subject wore one Myo armband on his/her dominate arm. Eight EMG signals were collected from the armband at the sampling rate of 200 Hz and segmented into overlapped analysis windows. The window length and increment were set to 200ms and 40ms, respectively, so one classification decision was made every 40ms. Four time domain features were extracted from each analysis window. Two classification algorithms were evaluated including LDA and C-SVM with the linear kernel. Due to weak EMG outputs from the Myo armband for *supination* and *pronation* gestures, these two gestures were performed by combining the wrist motion with some finger motion. Specifically, the subject was instructed to perform the five gestures as followed:

- *Open*: fingers fully extended with slight spread, wrist loose.
- *Fist*: fingers flexed, thumb over the top, wrist loose.
- *Point*: index finger extended with radial deviation, other fingers loosely flexed.
- *Supination*: wrist supination with thumb extended, pinky and ring flexed into palm, index and middle relaxed.
- *Pronation*: wrist pronation with index and thumb flexed into each other, pinky extended.

The experiment consisted of two sessions: training and testing. In the training session, the subject was instructed to perform each motion and hold it for two seconds. The rest period between successive motions was three seconds. The five investigated motions were performed in order for three times. A researcher manipulated the MyoHMI GUI to start and stop the data collection as well as save the data into files. Following the initial training data collection, the motion classifier was created by using the selected classification method. Throughout the procedure, the MyoHMI GUI provided various visual feedbacks such as the confusion matrix of the training results and real-time visualization of the EMG signals and features, which helped the researcher and the subject adjust the way of performing motions and the collection of additional training data if needed to get better results.

In the testing session, the subject was instructed to perform the investigated motions first in the same order as in the training session, then in a random order as cued by the researcher in one testing trial. Totally five trials were conducted for each subject. There was a three-second countdown display in the GUI to cue the subject before switching to the next motion. The MyoHMI platform made classification decisions using the trained classifier and displayed the decisions in the GUI in real-time. Meanwhile, the decisions were sent out by the output interface to control the prosthetic hand and the virtual arm.

B. Results and Discussion

Table I shows the classification accuracy of the real-time experiments for hand motion recognition. The classification accuracy was calculated as the number of correct classification decisions divided by the total number of classification decisions in the testing data. The classification accuracy averaged across four subjects was 91.25%±2.38% for using LDA and 92.33%±3.39% for C-SVM with linear kernel. The classification decisions were sent to the two evaluations platforms seamlessly to drive the prosthetic hand and the virtual arm in real-time as shown in Figure 7. The experimental results verified the correctness of functionality of the MyoHMI platform. A video of our real-time experiment can be viewed at:

https://drive.google.com/open?id=0B3naPAeRD3-jNWhnMW9JTUppOWs.

One thing to be noted is that the experimental procedure only took less than 30 minutes for each subject. This includes the preparation and setup time for the experiments as well as the time for the actual training and testing sessions.

TABLE I. Classification Accuracy of Hand Motion Recognition

Subject	1	2	3	4	Average
LDA (%)	90.40	92.30	93.90	88.40	91.25±2.38
SVM (%)	95.10	93.70	93.10	87.40	92.33±3.39

Figure 6. An example showing the more enhanced virtual hand assessment environment currently under development.

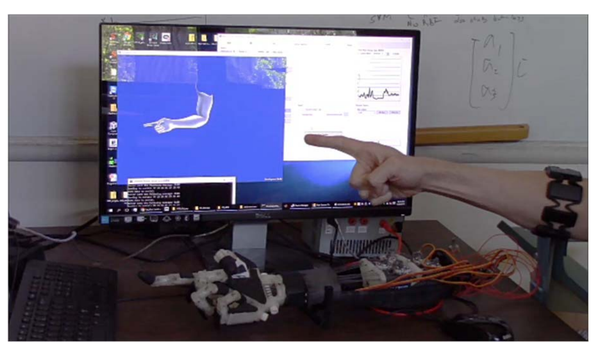
Figure 7. Real-time testing of the MyoHMI platform for controlling the 3D printed prosthetic hand and the virtual arm simultaneously.

TABLE II. COMPARISON BETWEEN THE MYOHMI PLATFORM AND EXISTING RESEARCH PLATFORMS FOR EMG PATTERN RECOGNITION

	MyoHMI	Existing Platforms
Programming language	C/C++	MATLAB
Installation requirement	Can be easily installed on any computer; no requirement for licensed software	Needs the installation of MATLAB and several toolboxes, which are not free
Total cost	The only cost is the cost of the Myo armbands (< $200 per piece)	Thousands to tens of thousands of dollars for purchasing the EMG acquisition system and installing licensed software
Experiment preparation time	No special preparation needed; the Myo armband can be easily worn on the forearm	A professional experimenter is usually required to place EMG electrodes on the subject and check the signal quality
Potential applications	Broad; the platform can easily be used anywhere	Tailored to prosthetic control research; experiments limited in the lab environments

Compared to traditional myoelectric control research that typically requires at least 30 minutes placing EMG electrodes and checking signal quality, this observation indicates that the use of the MyoHMI platform and the commercial armband can significantly simplify the experimental procedure and thus can be easily applied in broader applications. Table II summarizes the comparison between the MyoHMI platform and the existing research platforms for EMG pattern recognition as discussed in the introduction section.

IV. CONCLUSION

This paper introduced a low-cost, easy to use, and flexible platform MyoHMI for developing real-time HMIs for myoelectric controlled systems. MyoHMI facilitates the interface with a commercial EMG-based armband Myo and provides a highly modular and customizable C/C++ based software engine for developing and evaluating EMG signal processing and pattern recognition algorithms. Real-time experiments on able-bodied subjects for controlling a 3D printed prosthetic hand and a VR system verified the functionality of the MyoHMI platform. Compared with existing research platforms for EMG pattern recognition, MyoHMI can significantly simplify the experimental procedure, and be easily installed on any computers without the requirement of any licensed software tools. The results demonstrated the feasibility of a low-cost solution for the development of myoelectric controlled applications, which has great potential to facilitate collaborations and new development in the community. Future work includes integrating more features, classification methods, and control schemes to the MyoHMI platform. Efforts are underway to make MyoHMI open source.

ACKNOWLEDGMENT

The authors thank Kartik Bholla at San Francisco State University for his assistance in this study. The authors gratefully acknowledge the reviewers' comments.

REFERENCES

[1] J. V. Basmajian and C. De Luca, "Muscles Alive: Their Functions Revealed by Electromyography, 5th ed," vol. 278, ed: Williams & Wilkins, 1985, p. 126.

[2] Y. Oonishi, *et al.*, "A new control method for power-assisted wheelchair based on the surface myoelectric signal," *Industrial Electronics, IEEE Transactions on,* vol. 57, pp. 3191-3196, 2010.

[3] L. Dipietro, *et al.*, "Customized interactive robotic treatment for stroke: EMG-triggered therapy," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on,* vol. 13, pp. 325-334, 2005.

[4] P. Parker, *et al.*, "Myoelectric signal processing for control of powered limb prostheses," *Journal of electromyography and kinesiology: official journal of the International Society of Electrophysiological Kinesiology,* vol. 16, p. 541, 2006.

[5] T. A. Kuiken, *et al.*, "Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms," *Jama,* vol. 301, pp. 619-628, 2009.

[6] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *Biomedical Engineering, IEEE Transactions on,* vol. 50, pp. 848-854, 2003.ss

[7] X. Zhang, *et al.*, "On Design and Implementation of Neural-Machine Interface for Artificial Legs," *Industrial Informatics, IEEE Transactions on,* vol. 8, pp. 418-429, 2012.

[8] (2015, *Meet the Man with a Myo-Controlled Robotic Arm*. Available: http://developerblog.myo.com/meet-the-man-with-a-myo-controlled-robotic-arm/

[9] A. M. Qamar, *et al.*, "A Multi-Sensory Gesture-Based Occupational Therapy Environment for Controlling Home Appliances," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 671-674.

[10] K. Nymoen, *et al.*, "MuMYO–Evaluating and Exploring the MYO Armband for Musical Interaction," 2015.

[11] Y. Yang, *et al.*, "EMG Sensor-based Two-Hand Smart Watch Interaction," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 73-74.

[12] R. Lipovský and H. A. Ferreira, "Self hand-rehabilitation system based on wearable technology," in *Proceedings of the 3rd 2015 Workshop on ICTs for improving Patients Rehabilitation Research Techniques*, 2015, pp. 93-95.

[13] J. Orlosky, *et al.*, "An Interactive Pedestrian Environment Simulator for Cognitive Monitoring and Evaluation," in *IUI Companion*, 2015, pp. 57-60.

[14] M. McCullough, *et al.*, "Myo arm: swinging to explore a VE," in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, 2015, pp. 107-113.

[15] I. Phelan, *et al.*, "Exploring virtual reality and prosthetic training," in *Virtual Reality (VR), 2015 IEEE*, 2015, pp. 353-354.

[16] E. Scheme and K. Englehart, "A flexible user interface for rapid prototyping of advanced real-time myoelectric control schemes," 2008.

[17] W. Bishop, *et al.*, "A real-time virtual integration environment for the design and development of neural prosthetic systems," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2008, pp. 615-619.

[18] M. Ortiz-Catalan, *et al.*, "BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms," *Source code for biology and medicine,* vol. 8, p. 1, 2013.

[19] B. Hudgins, *et al.*, "A new strategy for multifunction myoelectric control," *Biomedical Engineering, IEEE Transactions on,* vol. 40, pp. 82-94, 1993.

[20] H. Huang, *et al.*, "Continuous locomotion-mode identification for prosthetic legs based on neuromuscular-mechanical fusion," *IEEE Trans Biomed Eng,* vol. 58, pp. 2867-75, Oct 2011.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

[22] G. Langevin, "InMoov-Open Source 3D printed life-size robot," *pp. URL: http://inmoov. fr, License: http://creativecommons. org/licenses/by–nc/3.0/legalcode,* 2014.