30th Annual International IEEE EMBS Conference
Vancouver, British Columbia, Canada, August 20-24, 2008

An Embedded Controller for a 7-Degree of Freedom Prosthetic Arm

Francesco Tenore, Robert S. Armiger, R. Jacob Vogelstein, Douglas S. Wenstrand, and Stuart D. Harshbarger
The Johns Hopkins University Applied Physics Laboratory
Laurel, MD USA
Email: {francesco.tenore, jacob.vogelstein, robert.armiger, douglas.wenstrand, stuart.harshbarger}@jhuapl.edu

Kevin Englehart
Department of Electrical Engineering
University of New Brunswick
Fredericton, NB, Canada
Email: kengleha@unb.ca

***Abstract*—We present results from an embedded real-time hardware system capable of decoding surface myoelectric signals (sMES) to control a seven degree of freedom upper limb prosthesis. This is one of the first hardware implementations of sMES decoding algorithms and the most advanced controller to-date. We compare decoding results from the device to simulation results from a real-time PC-based operating system. Performance of both systems is shown to be similar, with decoding accuracy greater than 90% for the floating point software simulation and 80% for fixed point hardware and software implementations.**

I. Introduction

Current prosthetic limbs are limited to a few degrees of freedom (DOF) and rely on fairly simple mechanical or myoelectric control schemes [1]. The Defense Advanced Research Projects Agency (DARPA) seeks to change this state of affairs with the Revolutionizing Prosthetics program, which aims to develop a dexterous, high-DOF prosthetic arm that can be controlled with motor signals from the residual limb and/or neural activity in the central and peripheral nervous systems [2], [3]. In support of this goal, we have recently developed two upper limb prostheses, called Proto 1 (Fig. 1) and Proto 2, with 7 and 22 DOF, respectively. While the Proto 2 limb will likely require implanted electrodes to achieve independent control of all 22 DOF, Proto 1 is designed to be controlled non-invasively with surface myoelectric signals (sMES).

Myoelectric signals have been used for prosthesis control for over 50 years (reviewed in ref. 1). Although myoelectric recordings can theoretically support a large number of independently controlled channels, myoelectric prostheses have historically been limited to a few DOF [1]. This is partly due to the lack of multiple-DOF prosthetic limbs, and partly due to the difficulty of extracting multiple channels from the noisy signals.

In this paper, we present an embedded real-time hardware system (called the Biointerface Board, or BIB) aimed at recording and decoding sMES to control the seven DOF in the Proto 1 limb. Although sMES has been used to decode upper limb movements in the past [4]–[7], there are relatively few embedded hardware implmentations of sMES decoding algorithms [8], [9], and we believe the BIB+Proto 1 to be the most advanced embedded sMES controller and limb solution to be developed to-date. For evaluation purposes, the performance of the embedded algorithms is compared to a software implementation of the same algorithms running in a real-time PC-based operating system, using both fixed and floating point processing.

Fig. 1. Proto 1 limb system (from [10]).

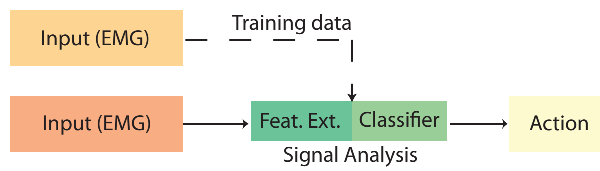

Fig. 2. Algorithm processing block diagram. The dashed line represents offline processing (training).

II. Methods

Electromyography (EMG) data were collected from the forearm of a male able-bodied subject who gave informed consent. The experiments were approved by the Johns Hopkins Institutional Review Board.

Eight bipolar Ag/AgCl Duotrode electrodes (Myotronics-Noromed) were placed around the proximal part of the forearm. Data were acquired off-line using a 12-bit Data Acquisition card (PCI 6024, National Instruments) and sampled at 1 kHz. The subject was asked to perform eight movements, holding each contraction for 3 s and then resting for 3 s. The movement classes were: elbow flexion and extension (ef, ee: isometric contractions); wrist pronation, supination, flexion, and extension (wp, ws, wf, we); and hand open and close (ho, hc). Data were also collected for a rest state, in order to discern movements from rest. Twelve repetitions of each movement and the rest state were recorded, eight of which

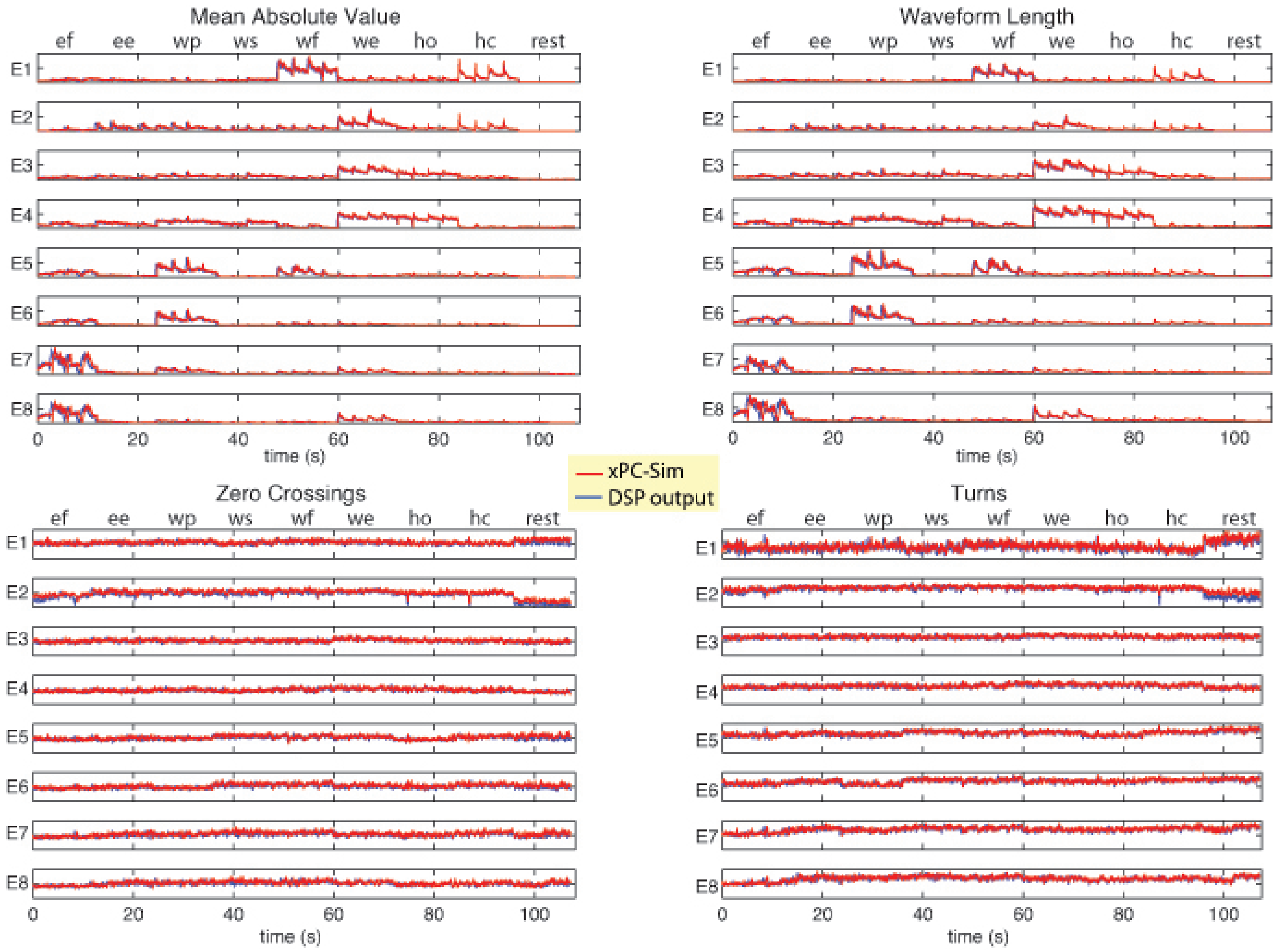

Fig. 3. Features extracted with software simulations (red) and the Bio-interface board (blue) for each electrode. Each of the nine movements described in Section II is performed four times, with each trial lasting 3 s, for a total of 12 s per movement and 108 s overall.

were used for training, and the remaining four were separately and subsequently recorded for testing (approximately two hours after the training data). The testing data were played back in real-time through a digital-to-analog converter (DAC) card (PCI 6733, National Instruments) equipped with eight analog output channels.

A schematic overview of the decoding algorithm is depicted in Fig. 2. First, features extracted from the (offline) training data are used to generate parameters—sample mean vectors μ and sample covariance C—for a simple linear discriminant analysis (LDA) classifier. (We chose to use LDA because Englehart *et al*. have shown that EMG-based pattern recognition is almost independent of the choice of classifier [11].) These parameters are then input to the classifier component of the Signal Analysis (SA) block. During testing, the SA block extracts features from incoming data and classifies them via LDA in real-time. For both training and testing, four features are extracted every 10 ms using data from within a 150 ms time window:

- Mean absolute value ($|\overline{X}|$): this feature displays a large increase in value at onset and maintains fairly high values during contraction.
- Waveform length (WL):

$$WL = \sum_{i=1}^{N} |x_i - x_{i-1}|$$

  The waveform length of the signal provides indicators for signal amplitude and frequency.
- Zero-crossings (ZC): this is the number of times the EMG waveform changes sign within a time window. It is correlated with the signal's frequency.
- Turns: this feature indicates the number of times the waveform's derivative changes sign within a time window.

More details on these features can be found in refs. 6 and 12. The implementation of these algorithms in software and hardware are described in the next sections.

A. *Software implementation*

A software implementation of the sMES decoder was implemented in our real-time Virtual Integration Environment (VIE) [13], developed as part of the Revolutionizing Prosthetics program to efficiently and realistically evaluate prototype limbs and associated control algorithms. Algorithms on the

VIE are developed in Simulink (The Mathworks, Inc.) and compiled for the real-time xPC Target (The Mathworks, Inc.) operating system.

The four EMG classification features are extracted in real-time using the formulas described above. Although the extracted features are all fixed-point integer calculations, and should therefore yield similar results in both a software and hardware implementation, the classifier's parameters (i.e., the mean μ and the covariance C) are calculated from the training set in a double-precision floating-point representation. Because the digital signal processor described in the next section has no floating point unit, we rescaled the parameters as follows:

$$\mu_{fp} = \lfloor(\kappa\mu) \quad (1)$$

$$C_{fp} = \lfloor(\kappa C) \quad (2)$$

where the operator $\lfloor$ denotes the extraction of the whole part of the multiplication and κ is the scaling factor:

$$\kappa = \frac{2^{15}}{\max(\max(\mu), \max(C))} \quad (3)$$

because the mapping is on a 16-bit integer space.

B. Hardware implementation

The BIB is implemented on a small (9.5 cm $\times$ 2.5 cm) printed circuit board. At its core, it consists of an embedded fixed-point digital signal processor (TMS320VC5509A, Texas Instruments, Inc.) that includes an analog-to-digital converter running at 16,000 samples/sec. Because the BIB contains no non-volatile memory, it cannot store any configuration data, and instead receives configuration parameters (including μ and C) from the prosthetic limb controller via shared memory. The configuration can subsequently be modified using a Bluetooth link. During operation, the BIB executes code in real-time and performs:

- high-pass filtering of the EMG activity at 80 Hz cutoff to remove cardiac influence as well as 60 Hz noise;
- extraction of the four features described in Section II;
- multiplication of the features by the μ_{fp} parameter as calculated in the previous section;
- addition of the C_{fp} parameter to the result;
- application of the $\max$ function to this result to output the predicted class value, i.e. the prothetic limb's control signal.

III. Results

Figs. 3a–d show the differences in computing the same four features in software (red) and hardware (blue) during the nine movements. It is evident that the integrated approach approximates the simulation results quite closely. However, there is a slight mismatch in sample frequency between the software and harware implementations, resulting in a non-constant offset. The discussion section examines this discrepancy in some detail.

Figs. 4–6 present classification results from the simulation and the integrated approaches. Specifically, Figs. 4 and 5 show

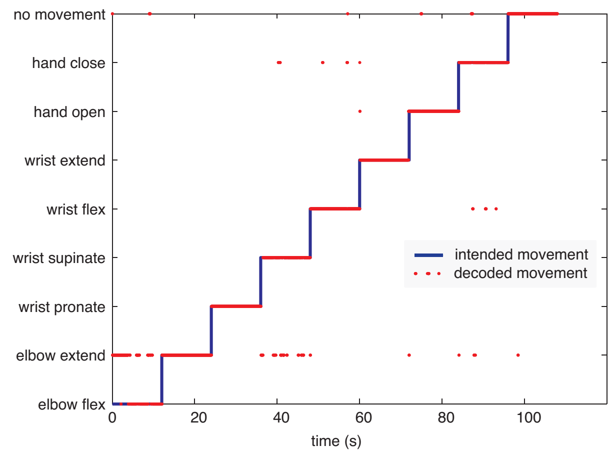

Fig. 4. Simulation results using floating point processing. The average accuracy for the 8+1 states is 91.0%.

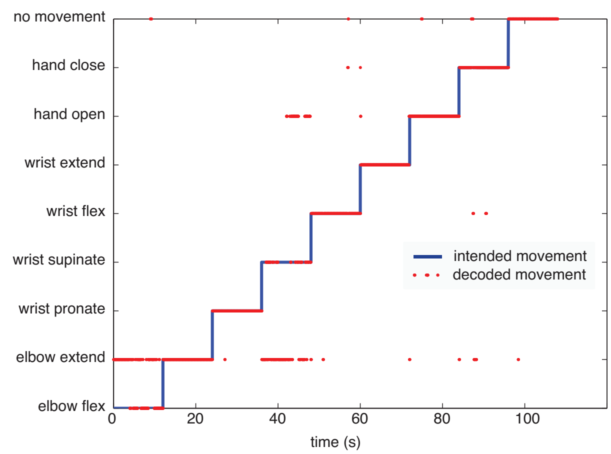

Fig. 5. Simulation results using fixed point processing. The average accuracy for the 8+1 states is 82.1%.

the differences in decoding the movements in software using floating- and fixed-point calculations, respectively. The decoding accuracy drops significantly due to the rescaling, from 91% to 82%. However, the difference in decoding accuracy between the software and hardware fixed-point calculations is not significant and is due in part to the lag between the extracted features. For the nine discrete movement states, the decoding accuracy of the embedded hardware is 79.5%.

IV. Discussion

We have presented experimental results from an embedded upper limb prosthesis controller. Although pattern recognition algorithms for decoding sMES have been studied extensively, to our knowledge, this work represents one of the first implementations of these algorithms in an embedded hardware system compact enough to fit inside an upper limb prosthesis. To validate the performance of the system, we compared its results with those produced by a custom-designed software-based implementation. The primary source of discrepancies between these implementations is caused by the hardware

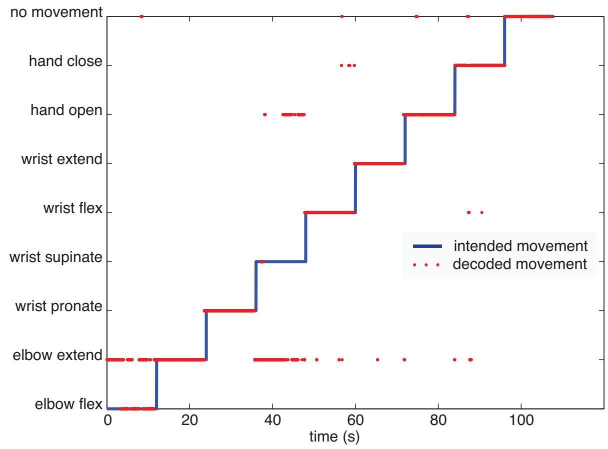

Fig. 6. Hardware results using the bio-interface board. The average accuracy for the 8+1 states is 79.5%.

system's reliance on fixed-point arithmetic as opposed to the higher resolution floating-point numbers used in software. Crucially, this observation implies that either a more tailored conversion from the floating-point to the fixed-point range or the use of a floating point processor is necessary. Currently, the approach depends only indirectly on the extracted training features, through the μ and C parameters. A rescaling based on the maximum values of the products of the μ matrix by the training set would constitute a better mechanism for fixing the output range. This would provide a broader output range thus enabling further separation between the classes. Nevertheless, it is important to note that the decoding accuracy in all cases remains significantly high, and is suitable for control of Proto 1.

What appears as a time lag between features extracted in software and hardware (Fig. 3) is mainly an artifact of our experimental setup. Because the actual sample frequency of the BIB differs from that of the xPC (although they are nominally set equal), both systems generate a different number of features during the same 108 s testing period. Here we have chosen to time-align the end of data collection in both systems. Therefore, although the data appears to have an decreasing time lag as the test goes on, this visual effect is caused by the synchronization of the endpoints of data collection and the slightly different number of samples collected by each system in the same time frame. In principle, this should not affect the classification results, because both systems operate in real-time. However, it does have the effect of reducing the decoding accuracy of the BIB when the xPC Target is used as the standard (compare Figs. 5 and 6).

Figs. 4–6 also allow identification of decoding errors due to confusion in the classifier output. The limit of 91.0% accuracy is due to two main sources of confusion. The first, confusion between elbow flexion and extension, is due in part to the fact that no activity from the biceps and triceps (the muscles most involved in these movements) was recorded—as mentioned in Section II, only EMG activity from the forearm was captured. The second, confusion of wrist supination with elbow extension and hand opening, is likely dependent on a characteristic of the movement itself: pronation and supination of the wrist were the only two movements that were not completely isometric and required movements of the hand itself.

V. Conclusion

We have successfully classified upper limb motions (and non-movement) using an embedded hardware controller for a prosthetic upper limb. The results of the decoding tasks were evaluated through comparison with a real-time software-based controller. Though both systems provide a high level of decoding accuracy for the trained movements, we conclude that the major differences between the two approaches lie not in the system used, but in the method to convert a floating point-based space into one provided by an embedded fixed-point digital signal processor. While acknowledging that there is room for improvements in classification, it should be noted that this difference does not prevent a successful deployment of this embedded system in our custom-designed upper limb prostheses.

Acknowledgments

The authors would like to thank the Defense Advanced Research Projects Agency's Revolutionizing Prosthetics 2009 program and NSERC grant 217354.

References

[1] P. Parker, K. Englehart, and B. Hudgins, "Myoelectric signal processing for control of powered limb prostheses," *Journal of Electromyography and Kinesiology*, vol. 16, no. 6, pp. 541–548, 2006.
[2] [Online]. Available: http://spectrum.ieee.org/dec07/5756
[3] [Online]. Available: http://www.darpa.mil/dso/thrusts/bio/restbio_tech/revprost/index.htm
[4] M. Zecca, S. Micera, M. Carrozza, and P. Dario, "Control of multi-functional prosthetic hands by processing the electromyographic signal," *Critical Review in Biomedical Engineering*, vol. 30, pp. 459–485, 2002.
[5] B. Hudgins and P. Parker, "A new strategy for multifunction myoelectric control," *Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
[6] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Medical Engineering & Physics*, pp. 431–438, 1999.
[7] G. Wang, Z. Wang, W. Chen, and J. Zhuang, "Classification of surface EMG signals using optimal wavelet packet method based on davies-bouldin criterion," *Medical and Biological Engineering and Computing*, vol. 44, pp. 865–872, 2006.
[8] G. Prasad, S. Srinivasan, and K. Patil, "A new DSP-based multichannel EMG acquisition and analysis system," *Computers and Biomedical Research*, vol. 29, no. 5, pp. 395–406, 1996.
[9] R. Martínez, R. Munoz, L. Leija, P. Hernandez, and J. Alvarez, "A control unit for myoelectrically controlled prosthesis," in *Proceedings of the Joint IEEE BMES/EMBS Conference*, 1999, p. 641.
[10] [Online]. Available: http://www.jhuapl.edu/newscenter/pressreleases/2007/070426.asp
[11] K. Englehart and B. Hudgins, "A robust, real time control scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, 2003.
[12] F. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Towards the control of individual fingers of a prosthetic hand using surface EMG signals," in *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference*, 2007.
[13] W. Bishop, R. Armiger, J. Burke, M. Bridges, J. Beaty, R. J. Vogelstein, and S. Harshbarger, "Virtual integration environment for the development of neural prostetics," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2008.