35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Design and Implementation of a Low Power Mobile CPU Based Embedded System for Artificial Leg Control

Robert Hernandez, *Senior member, IEEE*, Qing Yang, *Fellow, IEEE,* He Huang, *Senior member IEEE*
Fan Zhang, *Student member, IEEE*, Xiaorong Zhang, *Student member, IEEE*

Abstract—**This paper presents the design and implementation of a new neural-machine-interface (NMI) for control of artificial legs. The requirements of high accuracy, real-time processing, low power consumption, and mobility of the NMI place great challenges on the computation engine of the system. By utilizing the architectural features of a mobile embedded CPU, we are able to implement our decision-making algorithm, based on neuromuscular phase-dependant support vector machines (SVM), with exceptional accuracy and processing speed. To demonstrate the superiority of our NMI, real-time experiments were performed on an able bodied subject with a 20ms window increment. The 20ms testing yielded accuracies of 99.94% while executing our algorithm efficiently with less than 11% processor loads.**

Keywords—neural machine interface; leg control; support vector machines; mobile CPU.

I. Introduction

A pattern recognition (PR) strategy based on phase dependant and neuromuscular-mechanical fusion support vector machines (SVM) has been successfully developed in our research group to identify user intent in real-time to allow neural control of artificial legs [1, 2]. To make this strategy a feasible reality, a real-time neural machine interface (NMI) that is small, low cost, low power and capable of executing this computationally intensive algorithm needs to be developed. In our previous study we utilized FPGA technology to meet all of the NMI constraints with excellent results when executing a linear discriminant analysis (LDA) based classifier [3]. A non-linear SVM based algorithm was shown to provide increased accuracy over LDA [1], but is much more computationally intensive, which increases the complexity of an FPGA based design. This complexity exposes challenges such as language syntax, design environments, and toolsets during the design, implementation and troubleshooting phases of FPGA based systems [4].

Commodity mobile processors, such as the Intel Atom™ Z530, are low power (2.2 watts [5]), low cost, and portable. Our prior offline study developed a prototype mobile processor based NMI to execute our complex PR algorithm and performed an offline study [6]. The study showed that a mobile processor based NMI had great promise in control of artificial legs [6]. However, in order to meet the special requirement of high accuracy and real time processing, tailoring our SVM based NMI software to this mobile PC architecture is desirable and challenging. We have developed fully functional software based on the SVM classifier on the mobile PC with all necessary interfaces for a data acquisition system with the capability to acquire real-time electromyographic (EMG), mechanical force and moment data from human subjects. This newly developed NMI was combined with a Measurement Computing's USB-1616HS-BNC DAQ [7] to facilitate the collection of the real-time EMG and 6 degrees-of-freedom (DOF) mechanical data. This final NMI design was utilized to execute and test the performance of our phase dependant SVM based PR algorithm at a 20ms window increment during real-time experiments on an able bodied human subject.

This paper makes the following contributions:

- Design and implementation of a real-time capable NMI for artificial leg control based on a mobile processor;
- The first NMI embedded system to execute our phase dependant SVM based PR algorithm at 20ms window increments;
- A real time experiment that evaluates the potential use of mobile processors for real-time embedded implementation for neural control of powered lower limb prosthesis.

II. Software Design and Implementation

This study is based on a previously developed PR algorithm that identifies the user's locomotion mode based on electromyographic (EMG) signals acquired in real-time from thigh muscles and mechanical forces/moments signals acquired from 6 DOF load cell mounted on the prosthetic pylon [1,2]. The EMG and mechanical data are segmented by sliding analysis windows. Features data are extracted from raw EMG and mechanical signals in each analysis window and fused into a single feature vector. The feature vector is sent to a phase-dependant pattern classifier for determination of user intent. The phase-dependant pattern classifier consists of four sub-classifiers, one for each individually defined gait phase. A gait phase detector identifies the current gait phase in real-time and selects the corresponding sub-classifier for final determination of user intent. A detailed description of this previously designed PR algorithm can be found in [1] and [2].

A. Feature Extraction

In this study, four time-domain (TD) features (the mean absolute value, the number of zero crossings, the waveform length, and the number of slope sign changes) were extracted from EMG signals in each analysis window. For mechanical data, the mean, minimum, and maximum values in each

*This research was supported in part by the Department of the Navy (Naval Undersea Warfare Center, Newport, Rhode Island) and NSF/CPS #0931820, NSF#1149385, NIH #RHD064968A, NSF/CCF #0811333 and NSF/CCF #1017177.

The authors are with the Electrical, Computer and Biomedical Engineering Department, University of Rhode Island, Kingston, RI 02881, USA.

analysis window were extracted as the features. Further details on the feature extraction can be found in [1].

B. Phase Dependant Pattern Recognition

To accurately determine user intent, an SVM based classification architecture utilizing a Radial Basis Function (RBF) kernel and an SVM gamma parameter of 0.015 was used [1,2]. The phase-dependant classifier is composed of four sub-classifiers corresponding to one of the following four gait phases: initial double limb stance (phase 1), single limb stance (phase 2), terminal double limb stance (phase 3), and swing (phase 4) [8]. Throughout this paper, inclusive of the figures, we utilize the following gait phase definitions: 1 - Initial Double Limb Stance, 2 - Single Limb Stance, 3 - Terminal Double Limb Stance and 4 - Swing. The gait phase detector uses the real-time vertical Ground Reaction Force (GRF) to determine the gait phases. In order to build the SVM sub-classifier models, a training procedure is conducted on all the acquired training data sets. During training phase, the output of the phase detector is used to label the training data with its corresponding gait phase. Each sub-classifier is trained only with the data pertinent for its gait phase. During the real-time testing phase, the gait phase detector determines which sub-classifier is responsible for the determination of user intent. The gait phase detector's determination is used to select the appropriate sub-classifier to act upon the feature vector composed of fused EMG and mechanical data. The algorithmic data flow of the phase-dependant pattern recognition is shown in Fig. 1.

C. Software Architecture

We implemented the NMI software as shown in Fig. 1 in the C programming language. To meet real-time constraints, while executing on an Atom™ CPU, we implemented various performance enhancements techniques to the program. We took advantage of reduced dynamic memory management [9], loop unwinding [10] and inline function expansion [11].

To minimize the impacts of the real-time data logging on the application, a statically allocated and statically defined Random Access Memory (RAM) buffer was implemented that stored all the raw EMG, mechanical, classification and application performance data. The RAM buffer eliminated the need to write to the hard drive during time critical operations. Furthermore, it took advantage of the RAM's superior speed for storage. The real-time data logging for each classification was performed after all time-critical functions were completed (i.e., at the end of each classification). Lastly, the RAM buffer's contents were written to the hard drive for post analysis after the experiment was completed, such that no further time critical functions were being executed.

The final result is an embedded application specifically designed to minimize pipeline stalls, minimize OS impacts, minimize cost of memory allocation, minimize the impacts of real-time data logging and take advantage of the Intel Atom™ Z530 Processor hardware architecture. These enhancements provided the basis for the speed performance introduced by this embedded application.

As in our previous study [5], LIBSVM [12] was chosen as the open source library to utilize as the open source SVM libraries for our embedded application. This decision was based on LIBSVM's proven accuracy. Also, the analysis of LIBSVM's source code showed that it would be possible to modify the libraries for real-time use.

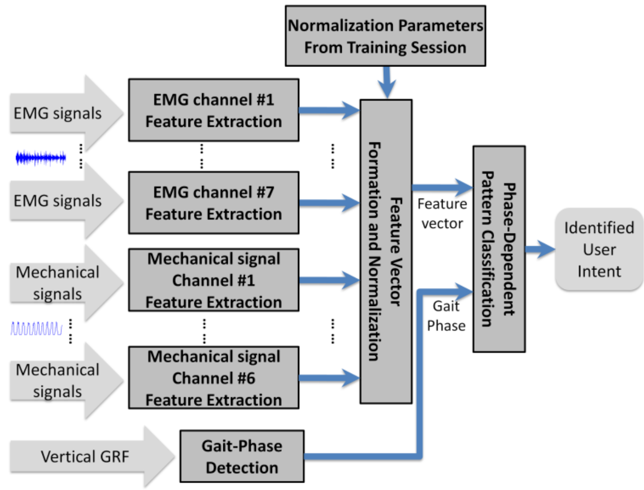

Figure 1. Phase-Dependant PR Algorithmic Data Flow

D. Software Implementation

To implement the Phase-Dependant PR algorithm, four applications were developed: a real-time training data capture application, a feature extraction & normalization application, a SVM training application and a Neuromuscular-Mechanical Fusion PR application. The real-time training data application captures training data for all the various human locomotion tasks. The feature extraction & normalization application accepts as input the real-time training data, performs the EMG and mechanical feature extraction and normalization, and then finally fuses the features into vectors. The feature vectors are then separated into their corresponding gait phases and provided to the training application. This application is also responsible for generating the normalization parameters required by the real-time PR application to normalize the real-time testing data, when determining user intent. The SVM training application accepts the four sets of training vectors and generates four SVM models, one model for each gait phase. The real-time PR application is used during the real-time testing phase. It accepts as input: raw real-time testing data, the four gait phase SVM models, and the normalization parameters. The real-time PR application extracts EMG and mechanical features from the raw testing data acquired in real-time. The features are then fused and normalized, with the provided normalization parameters and formed into a vector. Finally, the application determines the current gait phase, and forwards the test vector to the respective phase based classifier for determination of user intent. The software implementation data flow is shown in Fig. 2.

III. EXPERIMENTAL PROTOCOL

The AxiomTek eBOX530-820-FL1.6G fanless embedded hardware [13] with an Intel Atom™ Z530 Processor [5] was chosen for the prototype design to test real-time feasibility and capability. To sample the raw EMG and mechanical data in real-time a Measurement Computing's USB-1616HS-BNC DAQ [7] system was interfaced with the AxiomTek embedded hardware. The Measurement Computing DAQ

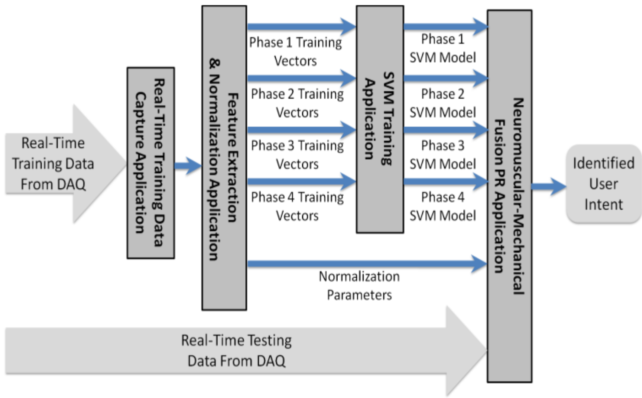

Figure 2. Software Implementation Data Flow

was chosen for its accuracy and capability to sample the data with a skew of 1 microsecond in between channels providing similar performance to that of a simultaneous sampling DAQ system.

A real-time performance evaluation utilizing a 20ms window increment with a window length of 160ms was conducted as part of this study. This experiment was conducted with approval of Institutional Review Board (IRB) at the University of Rhode Island and informed consent of the subject. The evaluations were performed on the data collected from a male able bodied subject. The collected data included the EMG signals from the subject's thigh muscles and mechanical forces/moments measured by a 6 degree-of-freedom load cell mounted on the prosthetic pylon. The monitored muscles included the sartorius (SAR), rectus femoris (RF), vastus medialis (VM), adductor magnus (ADM), biceps femoris short head (BFS), biceps femoris long head (BFL), and semitendinosus (SEM).

The EMG and mechanical forces/moments were sampled at 1 KHz by the Measurement Computing's USB-1616HS-BNC DAQ device. The user intent decisions provided by the embedded hardware were provided via an analog output interface on the DAQ device. The experiment provided real-time gait-phase and user intent decisions to the console screen as a visual cue during the training and testing processes.

For all the experiment performed in this study, the prediction time will be defined as the total time to execute feature extraction, normalization, gait phase detection, majority vote and classification for a single analysis window.

IV. Real-Time Performance Evaluation

The 20 ms window increment embedded software design incorporated a real-time ten point majority vote algorithm as in [8] and the phase detector was tuned to the subject's locomotion patterns during the real-time training phase.

For this experiment, three tasks (level-ground walking (W), stair ascent (SA), and standing (ST)) and two mode transitions (ST→W and ST→SA) were studied. To ensure the subject's safety, the subject was allowed to use hand rails when necessary. To train the gait-phase classifier, the subject was instructed to perform each task for approximately 10 seconds. Two trials of standing data, three trials of walking data, and three trials of stair ascent data were accumulated to train the classifier. For the real-time performance evaluation, 10 trials of each task and mode transitions were conducted (20 trials total). To assess the real-time performance of the NMI, the timing and processor loading of the application's execution on the embedded hardware are provided and the recognition accuracy of the NMI will be evaluated via a comparison with a similar LDA based NMI and the following parameters:

Classification Accuracy in the Static State: The static state is defined as the state where the subject has completed a transition and is continuously performing the same task (W, SA). The classification accuracy in the static state is the total number of correct classifications observed over the total number of classifications observed during the static state.

The Number of Missed Mode Transitions: The mode transition period starts from the beginning of gait phase 2 (single limb stance) and terminates at the beginning of gait phase 4 (swing). A mode transition is declared to have been missed, if no correct transition decision is made during this defined period.

Mode Transition Prediction Time: The mode transition prediction time is defined and the amount of time prior to the critical timing, during which the classifier user intent decision has stabilized and is no longer changing, such that safe switching of the prosthesis device is made possible. For this experiment, the critical timing is defined as the termination of the mode transition (i.e. - just prior to the start of the swing gait phase).

A. Recognition Accuracy of NMI and LDA Comparison

The overall classification accuracy of the NMI in the static states for all the predictions performed during the 20 trials inclusive of all tasks (W, SA, and ST) was 99.94%. No missed mode transitions were observed during the defined mode transition period. The mean mode transition prediction time for ST→SA was 658.0ms with a standard deviation of 155.6ms. The mean mode transition prediction time for ST→W was 534.0ms with a standard deviation of 103.3ms. The mode transition performance implies that user intent classification during transitions can be accurately determined, on the average, 514ms prior to the critical timing and be used for safe switching and control of the prosthesis. Representative trials, acquired during real-time testing, depicting the user intent classifications prior and during the ST→SA and ST→W transitions are provided in Fig. 3 and Fig. 4, respectively. As can be seen, the system is highly accurate and responsive. Furthermore, it can be seen that the transitions were correctly predicted prior to the critical timing and the static state accuracy was 100% during these two trials.

In comparison, a LDA based neuromuscular-mechanical fusion, phase-dependent pattern recognition NMI provided 97.41% accuracy in the static states [3]. Similarly, the LDA study was based on the same three tasks (W, SA, and ST), utilized the same window increment of 20ms, the same window length of 160ms, and performed same number of trials as well

B. Execution Timing and Processor Loading on the Embedded Hardware

A total of 14276 predictions were produced by the Intel Atom™ based embedded hardware during the trials. The mean prediction time per trial was 0.721ms with a

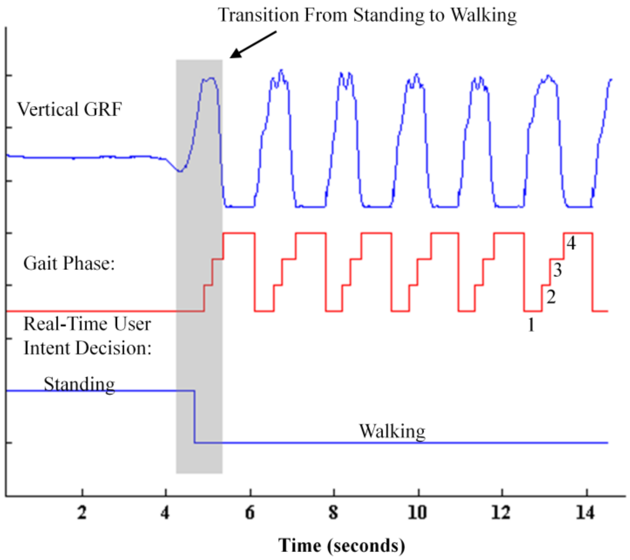

Figure 3. Real-Time Performance of a Standing to Walking Trial

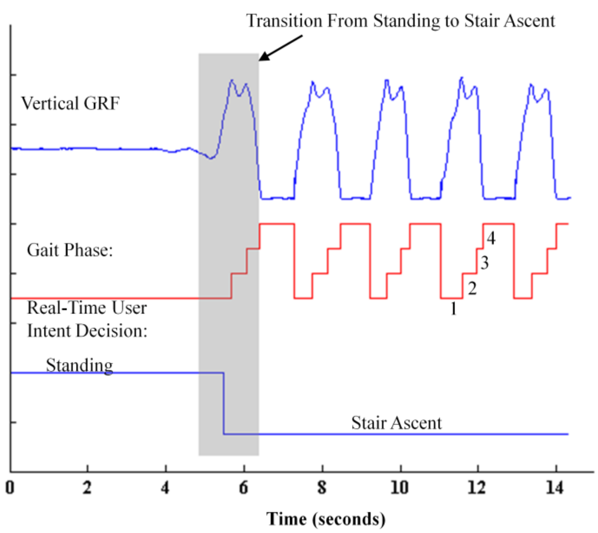

Figure 4. Real-Time performance of a Standing to Stair Ascent Trial

standard deviation of 0.0754ms. The worst case prediction executed in 2.124ms.

Due to the fact that there is additional loading on the CPU to execute the data logging for post analysis, the CPU loading provided by the operating system may be inaccurate; therefore the mean and maximum values of CPU loading were calculated using (1), which were 3.61% and 10.62% respectively.

$$CPU\ Loading = \frac{Prediction\ Time}{Window\ Increment\ (20ms)} * 100 \qquad (1)$$

V. Conclusion

This paper presented the design and implementation of a mobile CPU based neural machine interface for artificial legs. The designed NMI prototype was tested on an able-bodied subject for classifying multiple movement tasks (level-ground walking, stair ascent and standing) in real-time. In the 20ms real-time window increment experiments, the system achieved 99.94% classification accuracy in static states, while utilizing less than 10.62% of the Intel Atom™ CPU. The experiment showed fast response time for predicting the mode transitions. Lastly, this mobile CPU based design utilizes less power than other systems designed for similar applications [6], while still providing nearly 90% reserve to provide additional expansion capability of our NMI. The results demonstrated the feasibility of a mobile CPU based real-time NMI for control of artificial legs.

Our future work includes utilizing the reserve capacity provided by this efficient implementation to provide real-time impedance based leg control, real-time EMG motion artifact detection, and a real-time EMG signal trust assessments; thereby creating a single processor based NMI embedded solution that performs all these functions.

References

[1] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart, “Continous locomotion-mode indentification for prosthetic legs based on neuromuscular-mechanical fusion,” IEEE Trans Biomed Eng, vol 58, pp. 2867-75, 2011.

[2] H. Huang, T. A. Kuiken, and R. D. Lipshutz, “A strategy for identifying locomotion mode using surface electromyography,” *IEEE Trans Biomed Eng*, vol 56, pp. 67-73, 2009.

[3] X. Zhang, Q. Yang and H. Huang, “A Neural-Controlled Cyber Physical System for Intent Recognition for Artificial Legs,” presented at Design Automation Conference, San Francisco, 2012 (Accepted).

[4] I. Gonzalez, E. El-Araby, P. Saha, T. El-Ghazawi, H. Simmler, S. Merchant, B. Holland, C. Reardon, A. George, H. Lam, G. Stitt, N. Alam, M. Smith, “Classification of application development for FPGA-based systems,” *Conf Proc National Aerospace Electronics Conference*, 2008.

[5] Intel Corporation. (2010, June). “*Intel® Atom™ Processor Z5xx Series Datasheet*” [online]. Available: http://www.intel.com/content/www/us/en/processors/atom/atom-z540-z530-z520-z510-z500-45-nm-technology-datasheet.html [March 19, 2012]

[6] R. Hernandez, F. Zhang, X. Zhang, H. Huang and Q. Yang, “Promise of a Low Power Mobile CPU based Embedded System in Artificial Leg Control,” *Conf Proc IEEE Engineering in Medicine and Biology (EMBC) 2012*.

[7] Measurement Computing Corporation. (2008). “*USB-1616HS-BNC User's Guide*” [online]. Available: http://www.microdaq.com/measurement_computing/documents/usb-1616hs-bnc-user-manual.pdf [May 21, 2012]

[8] F. Zhang, W. DiSanto, J. Ren, Z. Dou, Q. Yang, H. Huang, “A Novel CPS System for Evaluating a Neural-Machine Interface for Artificial Legs,” *Proceeding of 2nd ACM/IEEE International Conference on Cyber-Physical Systems,* pp. 67-76, 2011.

[9] D. Tiwari, S. Lee, J. Tuck, Y. Solihin. “Exploiting fine-grained parallelism in dynamic memory management,” *IPDPS,* 2010.

[10] A. Nicolau, “Loop quantization: unwinding for fine-grain parallelism exploitation,” Cornell University, 1985, Avaliable: http://ecommons.library.cornell.edu/bitstream/1813/6549/1/85-709.pdf [March 19, 2012]

[11] W. W. Hwu, P. P. Chang, “Inline function expansion for compiling C programs,” *ACM SIGPLAN '89 Conference on Programming Language Design and Implentation*, Portland, Oregon, June 1989.

[12] C. C. Chang, C. J. Lin, “LIBSVM: a library for support vector machnies,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2 issue 3, pp. 27:1-27:27, 2011.

[13] AxiomTek Corporation. (2012). “*Fanless Embedded System with Intel® Atom™ Processor*” [online]. Available: http://axiomtek.com/Download/Spec/ebox530-820-fl.pdf [March 19, 2012]