THE UNIVERSITY OF RHODE ISLAND

Department of Electrical, Computer, and Biomedical Engineering Faculty Publications

University of Rhode Island
DigitalCommons@URI

Department of Electrical, Computer, and Biomedical Engineering

2011

On Design and Implementation of Neural-Machine Interface for Artificial Legs

Xiaorong Zhang
University of Rhode Island

Yuhong Liu
University of Rhode Island

See next page for additional authors

Follow this and additional works at: https://digitalcommons.uri.edu/ele_facpubs

Authors

Xiaorong Zhang, Yuhong Liu, Fan Zhang, Jin Ren, Yan (Lindsay) Sun, Qing Yang, and He Huang

On Design and Implementation of Neural-Machine Interface for Artificial Legs

Xiaorong Zhang,
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

Yuhong Liu,
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

Fan Zhang,
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

Jin Ren[Student Member, IEEE],
VeloBit Inc, 80 Central St. Boxborough, MA 01719

Yan (Lindsay) Sun[Member, IEEE],
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

Qing Yang[Fellow, IEEE], and
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

He Huang[Member, IEEE]
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881

Xiaorong Zhang: zxiaorong@ele.uri.edu; Yuhong Liu: yuhong@ele.uri.edu; Fan Zhang: fzhang@ele.uri.edu; Yan (Lindsay) Sun: yansun@ele.uri.edu; Qing Yang: qyang@ele.uri.edu; He Huang: huang@ele.uri.edu

Abstract

The quality of life of leg amputees can be improved dramatically by using a cyber physical system (CPS) that controls artificial legs based on neural signals representing amputees' intended movements. The key to the CPS is the neural-machine interface (NMI) that senses electromyographic (EMG) signals to make control decisions. This paper presents a design and implementation of a novel NMI using an embedded computer system to collect neural signals from a physical system - a leg amputee, provide adequate computational capability to interpret such signals, and make decisions to identify user's intent for prostheses control in real time. A new deciphering algorithm, composed of an EMG pattern classifier and a post-processing scheme, was developed to identify the user's intended lower limb movements. To deal with environmental uncertainty, a trust management mechanism was designed to handle unexpected sensor failures and signal disturbances. Integrating the neural deciphering algorithm with the trust management mechanism resulted in a highly accurate and reliable software system for neural control of

artificial legs. The software was then embedded in a newly designed hardware platform based on an embedded microcontroller and a graphic processing unit (GPU) to form a complete NMI for real time testing. Real time experiments on a leg amputee subject and an able-bodied subject have been carried out to test the control accuracy of the new NMI. Our extensive experiments have shown promising results on both subjects, paving the way for clinical feasibility of neural controlled artificial legs.

Index Terms

Neural-machine interface; prosthetics; high performance computer; trust management

I. Introduction

There are over 32 million amputees worldwide whose lives are severely impacted by their condition. This number is growing as the population ages and as the incidence of dysvascular disease increases. Over 75% of major amputations were lower-limb, with nearly 17% of lower-limb amputees suffering bilateral amputations [1]. Therefore, there is a continued need to provide this large and growing population of amputees with the best care and return of function possible.

With the rapid advances of cyber system technologies, it has in recent years become possible for high speed, low cost, and real time embedded computers to be widely applied in biomedical systems. The computerized prosthetic leg is one prominent example, in which motion and force sensors and a microcontroller embedded in the prosthesis form a close loop control and allow the user to produce natural gait patterns [2–3]. However, the function of such a computerized prosthesis is still limited. The primitive prosthesis control is based entirely on mechanical sensing without knowledge of user intent. Users have to "tell" the prostheses their intended activities manually or using body motion, which is cumbersome and does not allow smooth task transitions. The fundamental limitation on all existing prosthetic legs is lack of neural control that would allow the artificial legs to move naturally as if they were the patient's own limb.

This paper presents a novel neural-machine interface (NMI) that makes neural controlled artificial legs possible. The new NMI is a cyber physical system (CPS), in which a complex physical system (i.e. neuromuscular control system of a leg amputee) is monitored and deciphered in real time by a cyber system. It senses neural control signals from leg amputees, interprets such signals, and makes accurate decisions for prostheses control. The neural signals that our NMI senses and collects from leg amputees are Electromyographic (EMG) signals that represent neuromuscular activity and are effective biological signals for expressing movement intent [4]. EMG signals have been used in many engineering applications, such as EMG-based power-assisted wheelchair [5], biofeedback therapeutic manipulator for lower limb rehabilitation [6], neuro-fuzzy interference system for identifying hand motion commands [7], and neural controlled artificial arms [8–9]. Previous research has shown that EMG was effective and clinically successful for artificial upper limbs [8–9]. However, no EMG-controlled lower limb prosthesis is currently available, and published studies in this area are very limited because of the following technical challenges.

First of all, in human physiological systems, EMG signals recorded from leg muscles during dynamic movements are highly non-stationary. Dynamic signal processing strategies [10] are required for accurate decoding of user intent from such signals. In addition, patients with leg amputations may not have enough EMG recording sites available for neuromuscular

information extraction due to the muscle loss [10]. Maximally extracting neural information from such limited signal sources is necessary and challenging.

The second important challenge is that the accuracy in identifying the user's intent for artificial legs is more critical than that for upper limb prostheses. A 90% accuracy rate might be acceptable for control of artificial arms, but it may result in one stumble out of ten steps, which is clearly inadequate for safe use of artificial legs. Achieving high accuracy is further complicated by environmental uncertainty, such as perspiration, temperature change, and movement between the residual limb and prosthetic socket may cause unexpected sensor failure, influence the recorded EMG signals, and reduce the trustworthiness of the NMI [11]. It is therefore critical to develop a reliable and trustworthy NMI for safe use of prosthetic legs.

The third challenge is the compact and efficient integration of software and hardware in an embedded computer system in order to make the EMG-based NMIs practical and available to patients with leg amputations. Such an embedded system must provide high speed and real time computation of neural deciphering algorithm because any delayed decision-making from the NMI also introduces instability and unsafe use of prostheses. Streaming and storing multiple sensor data, deciphering user intent, and running sensor monitoring algorithms at the same time superimpose a great challenge to the design of an embedded system for the NMI of artificial legs.

To tackle these challenges, a neural interfacing algorithm has been developed that takes EMG inputs from multiple EMG electrodes mounted on a user's lower limb, decodes the user's intended lower limb movements, and monitors sensor behaviors based on trust models. Our EMG pattern recognition (PR) algorithm, together with a post-processing scheme, effectively process non-stationary EMG signals of leg muscles so as to accurately decipher the user's intent. The neural deciphering algorithm consists of two phases: offline training and online testing. To ensure the trustworthiness of NMI in an uncertain environment, a real time trust management (TM) module was designed and implemented to examine the changes of the EMG signals and estimate the trust level of individual sensors. The trust information can be used to reduce the impact of untrustworthy sensors on the system performance.

The new deciphering algorithm was implemented on an embedded hardware architecture as an integrated NMI to be carried by leg amputees. The two key requirements for the hardware architecture were high speed processing of training process and real time processing of the interfacing algorithm. To meet these requirements, the newly designed embedded architecture consists of an embedded microcontroller, a flash memory, and a graphic processing unit (GPU). The embedded microcontroller provided necessary interfaces for AD/DA signal conversion and processing and computation power needed for real time control. The control algorithm was implemented on the bare machine with our own memory and IO managements without using existing OS to avoid any unpredictability and variable delays. The flash memory was used to store training data. EMG PR training process involved intensive signal processing and numerical computations, which needs to be done periodically when the system trust value is low. Such computations can be done efficiently using modern GPUs that provide supercomputing performance with very low cost. New parallel algorithms specifically tailored to the multi-core GPU were developed exploiting memory hierarchy and multithreading of the GPU. Substantial speedups of the GPU for training process were achieved, making the classifier training time tolerable in practice.

A complete prototype has been built implementing all the software and hardware functionalities. The prototype was used to carry out real time testing on human subjects. A

male patient with unilateral transfemoral amputations was recruited in our experiments for evaluation of the user intent identification module. The goal of our experiments is to use the newly designed NMI prototype to sense, collect, and decode neural muscular signals of the human subject. Based on the neural signals, the NMI tries to interpret the subject's intent for sitting and standing, two basic but difficult tasks for patients with transfemoral amputations due to the lack of power from the knee joint. The trust management module was also tested on a male able-bodied subject by introducing motion artifacts during the subject's normal sitting and standing task transitions. The detection rate and false alarm rate for distribution detection was evaluated.

Extensive experiments of our NMI on the human subjects have shown promising results. Among the 30 sitting-to-standing transitions and the 30 standing-to-sitting transitions of the amputee subject, our NMI recognized all the intended transitions correctly with a maximum decision delay of 400ms. Our algorithm can also filter out occasional signal disturbances and motion artifacts with 99.37% detection rate and 0% false alarm rate. The videos of our experiments can be found at http://www.youtube.com/watch?v=H3VrdqXfcm8 and http://www.youtube.com/watch?v=6NwtMOw0YS0.

The paper is organized as follows: The next section presents the system architecture and design of the algorithms and embedded system. Section III describes the experimental settings for our real time testing of the NMI prototype on the amputee and able-bodied subjects. The results of the study are demonstrated in section IV, followed by related work in section V, and a conclusion in section VI.

II. System Architectures

A. System Architecture

The architecture of neural-machine interface is shown in Fig. 1. Multiple channels of EMG signals are the system inputs. EMG signals are preprocessed and segmented by sliding analysis windows. EMG features that characterize individual EMG signals are extracted for each analysis window. The system consists of two major pathways: one path for classifying user movement intent and the other for sensor trust evaluation (the dashed blocks in Fig. 1). To identify user intent, EMG features of individual channels are concatenated into one feature vector. The goal of pattern recognition is to discriminate among desired classes of limb movement based on the assumption that patterns of EMG features at each location is repeatable for a given motion but different between motions [9]. The output decision stream of EMG pattern classifier is further processed to eliminate erroneous task transitions. In the path for sensor trust evaluation, the behaviors of individual sensors are closely monitored by abnormal detectors. A trust manager evaluates the trust level of each sensor and then adjusts the operation of the classifier for reliable EMG pattern recognition.

The hardware architecture of the NMI (Fig. 2) for artificial legs consists of seven components: EMG electrodes, amplifier circuits, analog-to-digital converters (ADCs), flash memory, RAM, GPU and an embedded controller. Multiple channels of EMG signals are collected from different muscles on the patient's residual limb using EMG electrodes. The amplifier circuits are built to make signal polarity, amplitude range, and signal type (differential or single-ended) compatible with the input requirements of ADCs. The outputs of the amplifier circuits are converted to digital format by the ADCs and then stored in a flash memory or a RAM. The embedded hardware works in two modes: training mode and real time testing mode. In the training mode, a large amount of EMG data are collected and stored in the flash memory. These data are then processed to train the EMG pattern classifier. The PR algorithm for the training phase includes complex signal processing and numerical computations, which are done efficiently in a high performance GPU. The

parameters of the trained classifier are stored in the flash memory upon completion of the training phase. The real time testing phase is implemented on the embedded microcontroller, including both the PR algorithm and the TM algorithm. In the real time testing mode, the EMG signals are sampled continuously and stored in the RAM of the embedded controller. The EMG data are then sent to the trained classifier for a decision to identify the user's intended movement and at the same time each EMG sensor is monitored by an abnormal detector. The trust value of each sensor is evaluated by a trust manager.

B. Identification of User Intent

A dynamic EMG pattern classification strategy and post-processing methods were developed in this study for high decision accuracy.

EMG Signals—EMG signals recorded from gluteal and thigh muscles of residual limb were considered.

EMG Features—Four time-domain (TD) features [12] (the mean absolute value, the number of zero-crossings, the waveform length, and the number of slope sign changes) were selected for real-time operation because of their low computational complexity [9] compared to frequency or time-frequency domain features. The detailed equation and description of these four TD features can be found in [12].

EMG Pattern Classification—Various classification methods, such as linear discriminant analysis (LDA) [12], multilayer perceptron [13], Fuzzy logic [14], and artificial neural network [10, 15], have been applied to EMG PR. The simple LDA classifier was used in this study because of the comparable classification accuracy to more complex classifiers [9, 16–18] and the computation efficiency for real-time prosthesis control [9].

The idea of discriminant analysis is to classify the observed data to the movement class in which the posteriori probability $P(C_g \mid \bar{f})$ can be maximized. Let Cg ($g \in [1, G]$) denote the movement classes and $\bar{f}$ be the feature vector in one analysis window. The posteriori probability is the probability of class Cg given the observed feature vector $\bar{f}$ and can be expressed as

$$P(C_g \mid \overline{f}) = \frac{P(\overline{f} \mid C_g)P(C_g)}{P(\overline{f})}, \tag{1}$$

where $P(C_g)$ is the priori possibility, $P(\bar{f} \mid C_g)$ is the likelihood, and $P(\bar{f})$ is the possibility of observed feature vector $\bar{f}$. Given the movement class C_g, the observed feature vectors have a multivariate normal (MVN) distribution. In addition, the priori possibility is assumed to be equivalent for each movement class, and every class shares a common covariance. Hence, the maximization of posteriori possibility in (1) becomes

$$\tilde{C}_g = \arg \max_{C_g} \{\overline{f}^T \Sigma^{-1} \mu_g - \frac{1}{2}\mu_g{}^T \Sigma^{-1} \mu_g\}. \tag{2}$$

The following expression,

$$d_{C_g} = \overline{f}^T \Sigma^{-1} \mu_g - \frac{1}{2}\mu_g{}^T \Sigma^{-1} \mu_g, \tag{3}$$

is defined as the linear discriminant function, where μ_g is the mean vector and Σ is the common covariance matrix.

During the offline training, μ_g and $\sum$ were estimated by feature vectors calculated from a large amount of training data and were stored in the flash memory.

Let

$$\tilde{\mu}_g = \frac{1}{K_g} \sum_{k=1}^{K_g} \overline{f}_{C_{g,k}}$$

and

$$\tilde{\Sigma} = \frac{1}{G} \sum_{g=1}^{G} \frac{1}{K_g - 1} (F_g - M_g)(F_g - M_g)^T$$

where Kg is the number of observations in class C_g; $\overline{f}_{C_g, k}$, is the k observed feature vector in class Cg; Fg is the feature matrix $F_g = [\overline{f}_{C_g,1}, \overline{f}_{C_g,2}, \ldots \overline{f}_{C_g,k}, \ldots, \overline{f}_{C_g,K_g}]$; Mg is the mean matrix $M_g = [\tilde{\mu}_g, \tilde{\mu}_g, \ldots \tilde{\mu}_g]$ that has the same number of columns as in Fg. Then, the parameters in the linear discriminant function (3) were known, i.e.

$$\tilde{d}_{C_g} = \overline{f}^T \tilde{\Sigma}^{-1} \tilde{\mu}_g - \frac{1}{2} \tilde{\mu}_g{}^T \tilde{\Sigma}^{-1} \tilde{\mu}_g. \tag{4}$$

In the real time testing, the observed feature $\overline{f}$ derived from each analysis window was fed to the classifier to calculate $\tilde{d}_{C_g}$ in (4) for each movement class and was classified into a specific class $\tilde{C}_g$ that satisfied

$$\tilde{C}_g = \arg \max_{C_g} \{\tilde{d}_{C_g}\}, C_g \in \{C_1, C_2, \ldots, C_G\}.$$

Dynamic Pattern Classification Strategy—When EMG signals are non-stationary, the EMG features across time show large variation within the same task mode, which results in overlaps of features among classes and therefore low accuracy for PR [10]. By assuming that the pattern of non-stationary EMGs has small variation in a short-time window and EMG patterns are repeatable for each defined short-time phase, a phase-dependent EMG classifier was designed, which was successfully applied to accurately and responsively recognize the user's locomotion modes [10]. For non-locomotion modes such as sitting and standing, the classifier can be built in the movement initiation phase by the same design concept. The structure of such a dynamic design of the classifier can be found elsewhere [10].

Post-processing of Decision Stream—Majority vote was used to eliminate erroneous decisions from the classifier. Majority vote [9] simply removes the decision error by smoothing the decision output. Note that this method can further increase the accuracy of NMI, but may sacrifice the system response time.

C. Trustworthy sensor Interface

The NMI for artificial legs must be reliable and trusted by the prosthesis users. The design goals of trustworthy sensor are (1) prompt and accurate detection of disturbances in real time applications, and (2) assessment of reliability of a sensor/system with potential

disturbances. To achieve these goals, a trust management module that contains three parts: abnormal detection, trust manager, and decision support was designed.

Abnormal Detection—For each EMG channel, an abnormal detector is applied to detect disturbances occurring in the EMG signal. Disturbances that cause sensor malfunctions can be diverse and unexpected. Among all these disturbances, motion artifacts can cause large damage and are extremely difficult to totally remove. Motion artifacts are also fairly common in both laboratory environment and in real systems. Therefore, in this paper, the focus was on the detection of motion artifacts.

To detect abnormality in EMG signals, a change detector that identifies changes in the statistics of EMG signals was proposed. During preliminary study, it was found that motion artifacts can lead to changes in two time-domain (TD) features: mean absolute value (increase) and the number of slope sign changes (decrease). Let Fe_{mean} and Fe_{slope} denote these two features, respectively. Positive change in Fe_{mean} and negative change in Fe_{slope} are used as indicators of the presence of motion artifacts. Moreover, since the changes are in two directions, a two-sided change detector, which can detect both positive change and negative change, is required.

Many statistical methods can be used to build the change detector. In this work, the Cumulative Sum (CUSUM) algorithm was chosen because it is reliable for detecting small changes, insensitive to the probabilistic distribution of the underlying signal, and optimal in terms of reducing the detection delay [19]. Particularly, the two-sided CUSUM detector was adopted [20].

$$S_{hi}(i)=\max(0, S_{hi}(i-1)+x_i - \widehat{\mu}_0 - k) \tag{5}$$

$$S_{lo}(i)=\max(0, S_{lo}(i-1)+\widehat{\mu}_0 - k - x_i) \tag{6}$$

where x_i represents the ith data sample, $\hat{\mu}_0$ is the mean value of data without changes, and k is CUSUM sensitivity parameter. The smaller the k is, the more sensitive the CUSUM detector is to small changes. In (5) and (6), S_{hi} and S_{lo} are used for detecting the positive and negative changes, respectively. If S_{hi} (or S_{lo}) exceeds a certain threshold (Th), a positive (or negative) change is detected. The initial values of S_{hi} and S_{lo} were set to 0. In the real time testing, once CUSUM detector detects a change, it will raise an alarm and restart by setting S_{hi} and S_{lo} as 0 in order to detect the next change. By doing so, it can respond sensitively and promptly to multiple changes in the EMG signal.

The presence of a positive change in Fe_{mean} and a negative change in Fe_{slope} at the same time can serve as the indicator of a motion artifact. Therefore, S_{hi} is applied to detect positive changes in Fe_{mean} and S_{lo} is applied to detect negative changes in Fe_{slope}. When S_{hi} and S_{lo} exceed their corresponding thresholds at the same time, a motion artifact is detected.

In (5), x_i denotes the ith sample of Fe_{mean}, and is calculated as mean of the absolute value of EMG signal within the ith window. In (6), x_i denotes the ith sample of Fe_{slope}, and is calculated as number of the slope sign changes within the ith window. The value $\hat{\mu}_0$ in (5) and (6) is computed as the average of x_i before any changes were detected. The sensitivity parameter, k, is set as 0.05, and the threshold Th is set as 0.1 for both (5) and (6).

Notice that, to promptly respond to disturbances, CUSUM detector restarts for the next round of disturbance detection right after it detects a disturbance. However, there may be a disturbance lasting for some time and CUSUM detector would detect it for more than once.

This may lead to an inaccurate trust calculation. To avoid this problem, a post processing scheme is proposed to stabilize the detection result. Disturbances that are very close to each other are combined (i.e. within L continuous windows) as one disturbance. In our real time testing, L is set as 3, which represents 240ms. That is, if the detector is triggered repeatedly within 240ms, we consider this as one disturbance.

Trust manager—After the abnormal detector detects the disturbance in an EMG signal, the EMG sensor is either permanently damaged or perfectly recoverable. To evaluate the trust level of the sensor, let p_1 denotes the probability that a sensor behaves normally after one disturbance is detected.

Assume all disturbances are independent. The probability that a sensor is still normal after i disturbances, denoted by p_i, is $p_i = p_1^i$. The trust value is computed from the probability value by the entropy-based trust quantification method [21], as

$$T = \begin{cases} 1 - H(p_i), if 0.5 \leq p_i \leq 1 \\ H(p_i) - 1, if 0 \leq p_i \leq 0.5 \end{cases}$$

where T is the trust value and $H(p_i)$ is the entropy calculated as

$$H(p_i) = -p_i \log_2(p_i) - (1 - p_i) \log_2(1 - p_i) \qquad (7)$$

Different p_1 values should be set according to the nature of the disturbance. The larger the p_1 value, the less likely the disturbance can damage the sensor. The calculation of trust is extendable to the case that different disturbances are detected for one sensor. For example, if two disturbances, whose p_1 values are 0.8 and 0.9, respectively, are detected for a sensor, the p_i value in (7) can be replaced by 0.8×0.9. In this paper, only one type of disturbance (i.e. motion artifact) was tested. The p_1 value for motion artifact is set as 0.9.

Decision Making and Report—The trust information is provided to the user intent identification (UII) module to assist trust-based decisions. There are two levels of decisions.

1. Sensor level: When the sensor's trust value drops below a threshold, this sensor is considered as damaged, and its reading is removed from the UII module. The classifier needs to be re-trained without the damaged sensor.

2. System level: After removing the damaged sensors, the system trust can be calculated by the summation of trust values of the remaining sensors. If the system trust is lower than a threshold, this entire UII model is not trustworthy, and actions for system recovery must be taken. One possible action is to re-train the classifier. Another possible action is to instruct the patient to manually examine the artificial leg system.

D. Hardware Design

Technical challenges in hardware design are twofold. First of all, in order to increase the decision accuracy, frequent training computations may often be required, especially in uncertain environment, where the appearance of disturbances can be unpredictable and frequent. A training computation needs to be done not only whenever the user puts on the prosthesis but also whenever the system trust level goes below the predetermined threshold. Training data need to be recollected in these two cases. In addition, when a sensor's trust value drops below a threshold, the classifier also needs to be re-trained using existing training data in the flash memory such that the classifier can make decisions based on the

remaining undisturbed sensors. The training algorithms require intensive numerical computations that take a significantly long time, in the range of a few minutes to hours on a general purpose computer system [22]. It is very important to substantially speed up this training computation to make the training time of our NMI tolerable and practical. The second challenge is the real time processing of decision making in order to have smooth control of the artificial legs. Such real time processing includes signal sampling, AD/DA conversion, storing digital information in memory, executing PR algorithms, periodical trust management, and decision outputs. To meet these technical challenges, a new hardware design incorporating a multi-core GPU and an embedded system with a built-in flash memory was presented.

High performance and low cost multi-core GPUs [23] have traditionally been thought of as commodity chips to drive consumer video games. However, the push for realism in such games along with the rapid development of semiconductor technologies has made GPUs capable of supercomputing performance for many applications at very low cost. There are many low-end to medium GPU controller cards available on the market for under $50. However they deliver extraordinary computation power in the range of several hundreds of GFLOPS. Besides high performance and low cost, there has also been a technology drive for reliable and low power GPUs alongside FPGAs and CPUs for embedded applications such as military systems. For example, an embedded system using the ATI Radeon HD 3650 GPU draws very little power but delivers performance levels of hundreds of GFLOPS. The next-generation mobile GPUs are expected to nearly double this performance with a similar power envelope. Our NMI makes the first attempt to exploit such high speed and low cost GPU for the purpose of speeding up complex PR training computations. Our design for the training of the classifier used a NVIDIA 9500GT graphic card that has four multiprocessors with 32 cores working at the clock rate of 1.4 GHz. Each multiprocessor supports 768 active threads giving rise to a total of 3072 threads that can execute in parallel. These threads are managed in blocks. The maximum number of threads per block is 512. The size of the global memory is 1 GB with bandwidth of 25.6 GB/s. 64 KB of the global memory is read-only constant memory. The threads in each block have 16 KB shared memory which is much faster than the global memory because it is cached. In this study, this GPU card was connected using the ×16 PCI Express bus. Whenever the training computation was triggered, the GPU was called in to perform the training process and store the parameters of trained classifier in the flash memory to be used for real time decision-making.

The second part of the hardware design is based on Freescale's MPC5566 132 MHz 32 bits microcontroller unit (MCU) with the Power Architecture as shown in Fig. 3. The MCU has 40 channels of ADCs with up to 12 bit resolution and two levels of memory hierarchy. The fastest memory is 32KB unified cache. The lower level memories include 128KB SRAM and 3MB flash memory. The default system clock of the MCU is 12 MHz. The frequency modulated phase locked loop (FMPLL) generates high speed system clocks of 128 MHz from an 8 MHz crystal oscillator. The direct memory access (DMA) engine transfers the commands and data between SRAM and ADC without direct involvement of the CPU. Minimizing the intervention from CPU is important for achieving optimal system response. The device system integration unit (SIU) configures and initializes the control of general-purpose I/Os (GPIOs). The real-time results of the embedded system, including the identified user intent, individual sensor status and trust value, are sent to the GPIO pins and displayed by multiple LEDs on MPC5566 EVB.

III. Experiments and Prototype

A. Evaluation of Designed Algorithm

Assigned Tasks—To prove the design concept, the NMI system was designed to decipher the task transitions between sitting and standing. These tasks are the basic activities of daily living but difficult for patients with transfemoral amputations due to the lack of knee power. During the transition phase, EMG signals are non-stationary. The classifier was designed in the short transition phase. Although it is possible to activate the knee joint directly based on the magnitude of one EMG signal or force data recorded from the prosthetic pylon, unintentional movements of the residual limb in the sitting or standing position may accidently activate the knee, which in turn may cause a fall in leg amputees. Hence, intuitive activation of a powered artificial knee joint for mode transitions requires accurate decoding of EMG signals for identifying the user's intent from the brain.

Data Collection—This study was conducted with Institutional Review Board (IRB) approval at the University of Rhode Island and informed consent of subjects. For the real time evaluation of the designed pattern recognition algorithm, one male patient with a unilateral transfemoral amputation was recruited. To evaluate the sensor trust algorithm, one male able-bodied subject, free from orthopedic or neurological pathologies, was recruited. Seven surface EMG electrodes (MA-420-002, Motion Lab System Inc., Baton Rouge, LA) were used to record signals from gluteal and thigh muscles in one side of both subjects. The EMG electrodes contained a pre-amplifier which band-pass filtered the EMG signals between 10 Hz and 3,500 Hz with a pass-band gain of 20. For the able-bodied subject, the gluteal and thigh muscles on the dominant leg were monitored. After the skin was shaved and cleaned with alcohol pads, the EMG electrodes were placed on the anatomical locations described in [24]. For the amputee subject, the muscles surrounding the residual limb and the ipsilateral gluteal muscles were monitored. The subject was instructed to perform hip movements and to imagine and execute knee flexion and extension. We placed EMG electrodes at the locations, where strong EMG signals can be recorded. EMG electrodes were embedded into a customized gel-liner system (Ohio Willow Wood, US) for reliable electrode-skin contact. A ground electrode was placed near the anterior iliac spine for both able-bodied and amputee subjects. An MA-300 system (Motion Lab System Inc., Baton Rouge, LA) collected 7 channels of EMG data. The cut-off frequency of the anti-aliasing filter was 500 Hz for EMG channels. All the signals were digitally sampled at a rate of 1000 Hz and synchronized.

The states of sitting and standing were indicated by a pressure measuring mat. The sensors were attached to the gluteal region of the subject. During the weight bearing standing, the recordings of the pressure sensors were zero; during the non-weight bearing sitting, the sensors gave nonzero readings.

Experiment Protocol—To evaluate the pattern recognition algorithm, before the real-time system testing, a training session was required in order to collect the training data for the classifier. During the training session, the subject was instructed to perform four tasks (sitting, sit-to-stand, standing, and stand-to-sit) on a chair (50 cm high). For sitting or standing task, the subject was required to keep the position for at least 10 sec. In the sitting or standing position, the subject was allowed to move the legs and shift the body weight. For two types of transitions, the subject performed the transitions without any assistance at least 5 times. During the real-time system evaluation testing, the subject was asked to sit and stand continuously. A total of 5 trials were conducted. In each trial, the subject was required to sit and stand at least five times, respectively. Rest periods were allowed between trials in order to avoid fatigue.

To evaluate the sensor trust algorithm, 13 trials of real-time disturbance detection testing were performed on the able-bodied subject. In each trial, motion artifacts were introduced randomly on one EMG electrode in each task phase for four times. To add motion artifacts, the experimenter tapped an EMG electrode with roughly same strength. Motion artifacts were introduced 159 times in the entire experiment.

Real-time Evaluation of EMG Pattern Recognition—Four classes during the movement initiation phase were considered: sitting, sit-to-stand transition, standing, and stand-to-sit transition. Note that the classes of sitting and standing were not stationary because the subject was instructed to move the legs and shift the body weight in these positions. The output of the classifier was further combined into two classes (class 1: sitting and stand-to-sit transition; class 2: standing and sit-to-stand transition). Four TD features defined in [12] and LDA-based classifier were used. Overlapped analysis windows were used in order to achieve prompt system response. For the real-time algorithm evaluation, 140ms window length and 80ms window increment were chosen. Two indicators were used to evaluate the real-time performance of EMG pattern classifier: classification accuracy and classification response time. Two types of classification response time were defined: the time delay (RT1) between the moment that the classification decision switched from sitting (0) and standing (1) and the moment that the gluteal region pressure changed from nonzero value (non-weight bearing sitting) to zero value (weight-bearing standing); the time delay (RT2) between the moment that the classification decision switched from standing (1) to sitting (0) and the moment that the gluteal region pressure changed from zero value (weight-bearing standing) to non-zero value (non-weight bearing sitting).

Real-Time Evaluation of Abnormal Detection and Trust Management—EMG electrodes recorded EMG signals under the task transitions, unintentional leg movements, and sensor disturbances. There were two different states: (1) normal movements (N), including unintentional leg movements and transitions between sitting and standing, the total number of which were 364, and (2) disturbances (D), the total number of which were 159. The detectors detected two types of results: normal (N) or disturbance (D).

For the data sets with motion artifacts, the data in each trial were divided into analysis windows. A state (N or D) was assigned to each window. There were four detection results: (1) Hit (H): Truth = 'D', Detection = 'D'; (2) False Alarm (F): Truth = 'N', Detection = 'D'; (3) Miss Detection (M): Truth = 'D', Detection = 'N'; and (4) Correct no detection (Z): Truth = 'N', Detection = 'N'. The performance of designed detector were evaluated by

Probability of detection: $PD = \dfrac{H}{H+M}$

Probability of false alarm: $PFA = \dfrac{F}{F+Z}$

The trust value of sensors will also be shown.

B. Algorithm Implementation on NMI Hardware System

The offline PR training algorithm, the real time PR testing algorithm, and the real time TM algorithm were all implemented on the NMI hardware described in the previous section. The window length and the window increment were set to 140ms and 80ms, respectively. This is because the computation speed of MPC5566 is limited. It takes approximate 80ms to compute the EMG PR algorithm and to run the abnormal detection/trust evaluation algorithm on the data collected in a 140ms window. Therefore, the window increment should be no less than 80ms. It was observed in our experiments that enlarging the window

length exceeding 120ms does not affect the classification performance [10] but increases the decision-making time, which causes delayed system response.

A parallel algorithm specially tailored to the GPU architecture for the computation intensive part of the PR training algorithm was designed using CUDA: Compute Unified Device Architecture, which is a parallel computing engine developed by NVIDIA. At the time of this experiment, our GPU was not directly connected to the embedded MCU. Rather, NVIDIA 9500GT graphics card plugged into the PCI-Express slot of the PC server was used to do the training computation. The training results were then manually loaded into the flash memory of the embedded system board for real time testing. The GPU took inputs from 7 EMG channels, each of which had about 10,000 data points. The EMG data were segmented into analysis windows with 140ms in length. As a result, each window contained a 140×7 matrix. The training algorithm first extracted 4 TD features from each channel, producing a 28×1 feature vector for each window. Our parallel algorithm on the CUDA spawned 7 threads for each window resulting in totally 2,800 threads for 400 windows. All these threads were executed in parallel on the GPU to speed up the process. The resultant features were stored in a 28×W matrix, where W is the number of windows. The algorithm then set up K thread blocks, where K is the number of observed motions of the user. Each one of the K thread blocks had 28×14 threads, and a total of K×28×14 threads could execute simultaneously in parallel on the GPU architecture.

To demonstrate the speedup provided by our parallel implementation on the GPU, an experiment that compared the computation times of our training algorithm on both the GPU system and the fully equipped 3 GHz Pentium 4 PC server was conducted. (The results will be shown in Section IV-C.)

The real time testing algorithm was implemented on Freescale's MPC5566 evaluation board, integrating both the PR algorithm for user intent identification and the TM algorithm for sensor trust evaluation. The parameters of the trained PR classifier, a 28×4 matrix and a 1×4 matrix, calculated during the training phase by GPU were stored in the built-in flash memory on the MPC5566 EVB in advance. The ADCs sampled raw EMG data of 7 channels at the sampling rate of 1000 Hz continuously. Same as in the training phase, the EMG data were divided into windows of length 140ms and increment 80ms. In every analysis window, 4 TD features were extracted for each individual channel. During the user intent identification process, a 28×1 feature vector was derived from each window and then fed to the trained classifier. After the EMG pattern classification, one movement class out of four was identified. The result was post-processed by the majority vote algorithm to produce a final decision – sitting or standing. During the sensor trust evaluation process, each EMG sensor was monitored by an individual abnormal detector. Only two of the four TD features (the mean absolute value and the number of slope sign changes) were used to detect motion artifacts (algorithm details in Section II-C). Each abnormal detector monitored the changes of these two TD features to produce a status output for its corresponding sensor: normal or disturbed. A trust level manager then evaluated the trust level of individual sensor based on accumulated disturbance information.

In the real time embedded system design, to ensure smooth control of the artificial legs, precise timing control and efficient memory management are two challenges due to the speed and memory limitations of the embedded controller. We developed our own hardware management mechanism on the bare machine of the MPC5566 EVB without depending on any real time OS to avoid unpredictability and delay variations. A circular buffer was designed to allow simultaneous data sampling and decision making. The circular buffer consisted of three memory blocks $B1$, $B2$ and $B3$ that were used to store the ADC sampling data. Each block stored the data sampled in one window increment (80 samples in this

experiment). An additional memory block, *B4*, was used as a temporary storage during the computation of PR algorithm and TM algorithm.

Fig. 4 shows the timing diagram of the control algorithm during the real time testing process. In Fig. 4, Δt equals the window increment, t_{PR} is the execution time of PR algorithm, and t_{TM} is the execution time of TM algorithm. Two conditions need to be satisfied to ensure the smooth control of decision making without delay: (1) $t_{TM} + t_{PR} \leq \Delta t$ and (2) $t_w \leq 2\Delta t$, where t_w is the window length. At point t_0, the ADCs begin to sample EMG signals continuously and the digital data are stored in *B1*. From point t_1, *B1* is filled up and the in-coming data are stored in *B2*. At t_2, the data for the first window *W1* are available (stored in *B1* and *B2*), and an interrupt request is generated to notify the CPU that the data is ready for computation and trigger the computation to start. At the same time, new data keep coming in to be stored in *B3*. After the time interval $t_{TM} + t_{PR}$, at point t_3, the PR computation and the sensor trust computation of *W1* complete. The first decision *D1* is made, identifying the user's intent of window *W1* whether to sit or stand, and also reporting the status and the trust value of each sensor. At time t_4, *B3* is filled up and data for *W2* as a result of sliding 80ms are ready for the computation again, using data partly in B2 and partly in B3. At this time, *B1* is no longer in use so it can be replaced by new sampling data. At time t_5, the decision *D2* of window *W2* is made. At time t_6, data for *W3* (stored in *B3* and *B1*) are available, the algorithm computation for *W3* begins. At time t_7, *D3* is done and *B2* can be reused.

C. Real-Time Testing of the NMI Prototype

Using the NMI prototype described above, real time tests were carried out as described in Section III-A. At the time of this experiment, our trust model focused on abnormal detection and the trust was evaluated at the sensor level. The communication between the trust manager and the classifier was not fully considered. Therefore, to better evaluate our system performance, a two-phase experiment was set up to evaluate the performance of pattern recognition and that of sensor trust management separately. For both phases, the subjects performed transitions between sitting and standing continuously. During *the phase of PR evaluation*, there was no motion artifact manually added. However, the subject's unintentional movements and the movements between the residual limb and prosthetic socket were still a factor. The movement decisions made by the classification system were displayed on a LED light and a computer monitor in real time. In our experiment, a 5-window majority vote was applied to the decision stream to further eliminate the classification errors. During *the phase of sensor trust evaluation*, motion artifacts were manually introduced by randomly tapping an EMG electrode with roughly equal strength. The sensor status and the sensor trust value were monitored and displayed on a computer monitor. The user intent classification results were ignored during this phase.

IV. Results and Discussions

A. Real-Time Performance of Pattern Recognition

During the continuous real-time testing (more than 30 times sit-to-stand transitions and 30 times stand-to-sit transitions), all of the transitions between sitting and standing were accurately recognized. Although the subject moved the legs during the sitting position and shifted the body weight in the standing position, no classification error was observed.

The system classification response time (RT1 and RT2) was calculated by using the pressure data under the gluteal region and shown in Table 1.

The real-time performance of the designed NMI prototype in one representative trial is shown in Fig.5. Due to a 5-window majority vote method applied, around 400ms decision

delay for the sit-to-stand transitions were observed in Fig. 5, comparing to the falling edges of pressure data. It can be clearly seen that the majority vote post-processing method significantly improved system accuracy but sacrificed system response time. The video of real-time system performance can be found at http://www.youtube.com/watch?v=H3VrdqXfcm8.

Compared to the real-time testing results on one able-bodied subject (upper two photos in Fig. 6) in our experiments [25], a similarly high classification accuracy and reasonable system response time were achieved on the patient with transfemoral amputation (lower two photos in Fig. 6). The promising real-time performance of our designed NMI prototype demonstrates a great potential to allow the amputee patients to intuitively and efficiently control the prosthetic legs.

B. Real-Time Performance of Sensor Trust Algorithm

Fig. 7 shows the performance of the designed trust management method. There are three subfigures. The upper figure shows the EMG signal disturbed by motion artifacts. The middle one shows the CUSUM detection results, the bar represents the period that a motion artifact was detected. As seen in the figure, CUSUM detector was sensitive to motion artifacts, but insensitive to the muscle activity due to the normal leg movements. Additionally, the CUSUM had very small detection delay. The bars were always present immediately after a motion artifact. The lower figure shows the corresponding trust value. The trust value for motion artifacts gradually reduced when consistent disturbances were detected. In the future work, other methods for trust value calculation will be explored. For instance, for sensors with non-perfect trust values, it can be checked whether their future readings are consistent with other sensors that have high trust values. By doing so, the sensors that experienced an occasional disturbance and were not damaged can gradually regain the trust. Furthermore, the performance of CUSUM detector was evaluated by calculating its detection rate and false alarm rate. During the real time testing experiments, **CUSUM detector achieved 99.37% detection rate and 0% false alarm rate.** The undetected disturbances are disturbances with either small amplitude or short duration, so that a small number of such disturbances were not expected to affect the NMI decision significantly. The video of our experiment can be found at http://www.youtube.com/watch?v=6NwtMOw0YS0.

The designed CUSUM detector is accurate and prompt. The limitations of the current study are that only one electrode was disturbed and the trust manager evaluated the trust only at the sensor level. In the next design phase, we will (1) consider the situation with multiple sensor failures, (2) enable the communication between the trust manager and the classifier, and (3) evaluate the system-level trust of the entire NMI.

C. Performance of CPU vs. GPU for Training Procedure

Table 2 shows the measured speedup of our parallel algorithm on the NVIDIA GPU over the PC server for different window sizes. It is clear from this table that our parallel implementation on the GPU gives over an order of magnitude speedup over the PC server. This order of magnitude speedup is practically significant. Consider the case where the training time took half hour on a PC server [22]. The same training algorithm takes less than a minute using our new parallel algorithm on the GPU. From an amputee user point of view, training for less than a minute for the purpose of accurate and smooth neural control of the artificial leg is fairly manageable as compared to training for half an hour every time training is necessary. Furthermore, the speedup increases as the number of windows increases (Table 2). As a result, parallel computation of the training algorithm on GPU helps

greatly in the NMI design since the larger the number of windows, the higher its decision accuracy will be [25].

D. Discussion

While our experiments on two subjects, one able-bodied subject and the other transemoral amputee subject have demonstrated promising results of the new NMI prototype, system performance may vary among different subjects due to the inter-subject variations. One of our future research tasks is to recruit more amputee subjects with diverse age and gender groups to evaluate the performance of the new NMI

In this presented study, only two simple tasks (sitting and standing) were tested. To allow the prosthesis to perform more functions, more locomotion modes during the ambulation such as level ground walking, stair ascent/descent, and ramp ascent/descent should be considered in our future work. A phase-dependent strategy, which was proposed in our previous study [10], may be required for the designed neural-machine interface structure to handle the walking dynamics.

Among all types of disturbances, motion artifact occurs most frequently in practice and therefore is the main disturbance studied in this paper. Besides motion artifact, there are also some other disturbances which should be considered in our future work, such as baseline noise amplitude change, signal saturation, sensor loss of contact, and etc. To handle these diverse disturbances, we may need to extend the work by (1) applying the abnormal detection on more EMG features and (2) modifying the trust manager so that the trust value is determined not only by how many times disturbances occur but also by more complex factors, such as disturbance type, duration, severe level and etc.

V. Related Work

Real-time EMG pattern recognition has been designed to increase the information extracted from EMG signals and improve the dexterity of myoelectric control for upper limb prosthetics [9, 26]. However, no EMG-controlled lower-limb prostheses are currently available. Recently, the need for neural control of prosthetic legs has brought the idea of EMG-based control back to attention. Two previous studies have attempted to use EMG signals to identify locomotion modes for prosthetic leg control [10, 27]. Jin et al. [27] used features extracted from EMG signals from a complete stride cycle. Using such features, the algorithm results in a time delay of one stride cycle in real-time. In practical application, this is inadequate for safe prosthesis use. Our previous study designed a phase-dependent EMG pattern recognition method [10], which is a dynamic classifier over time. The result indicated over 90% classification accuracy, which can be applied for real time NMI. While both studies demonstrated that EMG information recorded from transfemoral amputees is sufficient for accurate identification of user intent, there has been no experimental study on design and implementation of embedded system to realize the NMI for reliable and real time control of prosthesis.

Reliable EMG pattern recognition system for artificial legs has been developed in our previous study [11]. It can enhance the system performance when sudden disturbances were applied to multiple sensors. In the previous work, however, the disturbances were generated through simulations and the algorithms were only tested offline [25]. The proposed algorithms in this paper, which were very different from the previous approaches, focused on real-time design with low detection latency, and were implemented and tested in a real-time embedded system.

There has been extensive research in using GPUs for general purpose computing (GPGPU) to obtain exceptional computation performance for many data parallel applications [23, 28–30]. A good summary of GPGPU can be found in [23, 31–32]. Our prior study made the first attempt to use GPU in EMG-controlled artificial legs and other medical applications [22]. Our results on individual computation components on EMG signal pattern recognition showed good speedups of GPU over CPU for various window sizes. The focus of the work reported in [22] was on parallel implementations of individual algorithms on GPU whereas this paper makes the first attempt to integrate the entire system for neural-machine interfacing (i.e. a CPS) for real time control of artificial legs. Our prior works [22] report offline analysis, while the work presented in this paper implements online decoding method for real-time testing. To the best knowledge of the authors, there has been no existing study on implementing the entire training algorithm on GPU for different numbers of windows and integrating the training algorithm together with real time testing on the same subject.

VI. Conclusions

A new EMG-based neural-machine interface (NMI) for artificial legs was developed and implemented on an embedded system for real time operation. The NMI represents a typical cyber-physical system that tightly integrated cyber and physical systems to achieve high accuracy, reliability, and real-time operation. This cyber-physical system consists of (1) an EMG pattern classifier for decoding the user's intended lower limb movements and (2) a trust management mechanism for handling unexpected sensor failures and signal disturbances. The software was then embedded in a newly designed hardware platform based on an embedded microcontroller and a GPU to form a complete NMI for real time testing. To prove our design concepts, a working prototype was built to conduct experiments on a human subject with transfemoral leg amputations and an able-bodied subject to identify their intent for sitting and standing. We also tested our trust management model on an able-bodied human subject by adding motion artifacts. The results showed high system accuracy, reliability and reasonable time response for real time operation. Our NMI design has a great potential to allow leg amputees to intuitively and efficiently control prosthetic legs, which in turn will improve the function of prosthetic legs and the quality of life for patients with leg amputations. Our future work includes the consideration of other movement tasks such as walking on different terrains, communications between trust models and user intent identification models, and exploring online training algorithms.

Acknowledgments

References

1. Potter B, Scoville C. Amputation is not isolated: an overview of the us army amputee patient care program and associated amputee injuries. Journal of the American Academy of Orthopaedic Surgeons. 2006; vol. 14:S188. [PubMed: 17003197]

2. Herr H, Wilkenfeld A. User-adaptive control of a magnetorheological prosthetic knee. Industrial Robot: An International Journal. 2003; vol. 30:42–55.

3. Psonak R. Transfemoral prosthetics. Orthotics and Prosthetics in Rehabilitation.

4. Basmajian, JV.; De Luca, CJ. Muscles alive : their functions revealed by electromyography. 5th ed.. Baltimore: Williams & Wilkins; 1985.

5. Oonishi Y, Oh S, Hori Y. A New Control Method for Power-Assisted Wheelchair Based on the Surface Myoelectric Signal. Industrial Electronics, IEEE Transactions on. 2010; vol. 57:3191–3196.

6. Komada S, Hashimoto Y, Okuyama N, Hisada T, Hirai J. Development of a biofeedback therapeutic-exercise-supporting manipulator. Industrial Electronics, IEEE Transactions on. 2009; vol. 56:3914–3920.

7. Khezri M, Jahed M. A Neuro-Fuzzy Inference System for SEMG Based Identification of Hand Motion Commands. Industrial Electronics, IEEE Transactions on. :1–1.

8. Parker P, Scott R. Myoelectric control of prostheses. Critical reviews in biomedical engineering. 1986; vol. 13:283. [PubMed: 3512166]

9. Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. IEEE Trans Biomed Eng. 2003 Jul.vol. 50:848–854. [PubMed: 12848352]

10. Huang H, Kuiken TA, Lipschutz RD. A strategy for identifying locomotion modes using surface electromyography. IEEE Trans Biomed Eng. 2009 Jan.vol. 56:65–73. [PubMed: 19224720]

11. Huang H, Zhang F, Sun YL, He H. Design of a robust EMG sensing interface for pattern classification. J Neural Eng. 2010 Oct..vol. 7:056005. [PubMed: 20811091]

12. Hudgins B, Parker P, Scott RN. A new strategy for multifunction myoelectric control. IEEE Trans Biomed Eng. 1993 Jan.vol. 40:82–94. [PubMed: 8468080]

13. Guler NF, Kocer S. Classification of EMG signals using PCA and FFT. J Med Syst. 2005 Jun.vol. 29:241–250. [PubMed: 16050079]

14. Ajiboye AB, Weir RF. A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control. IEEE Trans Neural Syst Rehabil Eng. 2005 Sep.vol. 13:280–291. [PubMed: 16200752]

15. Wilamowski BM. Neural network architectures and learning algorithms. Industrial Electronics Magazine, IEEE. 2009; vol. 3:56–63.

16. Huang Y, Englehart KB, Hudgins B, Chan AD. A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. IEEE Trans Biomed Eng. 2005 Nov.vol. 52:1801–1811. [PubMed: 16285383]

17. Englehart K, Hudgins B, Parker PA, Stevenson M. Classification of the myoelectric signal using time-frequency based representations. Med Eng Phys. 1999 Jul–Sep.vol. 21:431–438. [PubMed: 10624739]

18. Hargrove LJ, Englehart K, Hudgins B. A comparison of surface and intramuscular myoelectric signal classification. IEEE Trans Biomed Eng. 2007 May.vol. 54:847–853. [PubMed: 17518281]

19. Philips T, Yashchin E, Stein D. Using Statistical Process Control to Monitor Active Managers. Journal of Portfolio Management. 2003; vol. 30:186–191.

20. Page ES. Continuous Inspection Scheme. Biometrika. 1954; vol. 41:100–115.

21. Sun Y, Yu W, Han Z, Liu R. Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks. IEEE JSAC special issue on security in wireless ad hoc networks. 2006; vol. 24

22. Xiao, W.; Huang, H.; Sun, Y.; Yang, Q. Promise of embedded system with GPU in artificial leg control: Enabling time-frequency feature extraction from electromyography; Conf Proc IEEE Eng Med Biol Soc; 2009. p. 6926-6929.

23. Owens DJ, Luebke D, Govindaraju N, Harris M, Krüger J, Lefohn AE, Purcell TJ. A Survey of General-Purpose Computation on Graphics Hardware. Computer Graphics Forum. 2007; vol. 26:80–113.

24. Perotto, A.; Delagi, EF.; Iazzetti, J.; Morrison, D. Anatomical Guide For The Electromyographer: The Limbs And Trunk. Charles C Thomas Publisher Ltd; 2005.

25. Huang H, Sun Y, Yang Q, Zhang F, Zhang X, Liu Y, Ren J, Sierra F. Integrating neuromuscular and cyber systems for neural control of artificial legs. 2010:129–138.

26. Kuiken TA, Li G, Lock BA, Lipschutz RD, Miller LA, Stubblefield KA, Englehart KB. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. Jama. 2009 Feb 11.vol. 301:619–628. [PubMed: 19211469]

27. Jin D, Yang J, Zhang R, Wang R, Zhang J. Terrain Identification for Prosthetic Knees Based on Electromyographic Signal Features*. Tsinghua Science & Technology. 2006; vol. 11:74–79.

28. Bayoumi A, Chu M, Hanafy Y, Harrell P, Refai-Ahmed G. Scientific and engineering computing using ati stream technology. Computing in Science & Engineering. 2009:92–97.

29. Montrym J, Moreton H. The geforce 6800. Micro, IEEE. 2005; vol. 25:41–51.

30. Garland M, Le Grand S, Nickolls J, Anderson J, Hardwick J, Morton S, Phillips E, Zhang Y, Volkov V. Parallel computing experiences with CUDA. Micro, IEEE. 2008; vol. 28:13–27.

31. Owens JD, Houston M, Luebke D, Green S, Stone JE, Phillips JC. GPU computing. Proceedings of the IEEE. 2008; vol. 96:879–899.

32. Nickolls J, Dally WJ. The GPU computing era. Micro, IEEE. 2010; vol. 30:56–69.

Biographies
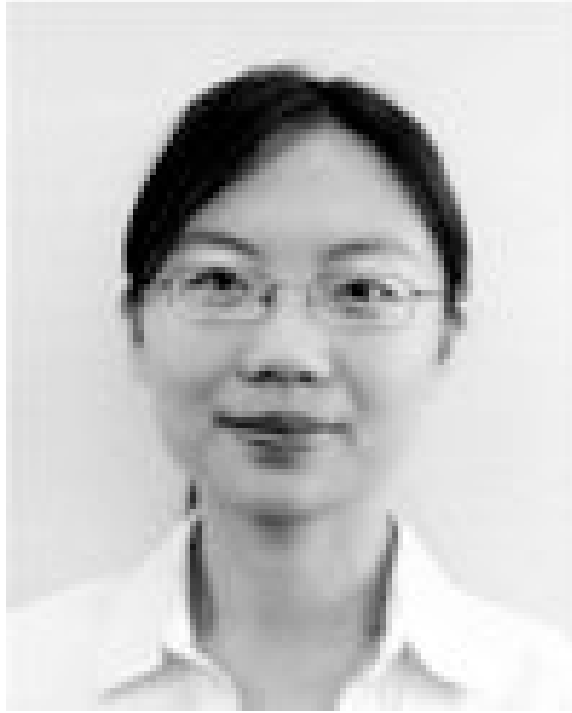
Xiaorong Zhang (S'10) received the bachelor's degree in computer science from Huazhong University of Science and Technology, China, in 2006, the master's degree in computer engineering from University of Rhode Island, Kingston, in 2009.

She is currently a Ph.D. student in the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island. Her research interests include embedded system, neural-machine interface, parallel and distributed computing, and computer architecture.

Yuhong Liu (S'10) is a PhD candidate in the Electrical, Computer and Biomedical Engineering Department at the University of Rhode Island (URI). She received her B.S. degree in information engineering in 2004 and M.S. degree in signal processing in 2007, both from Beijing University of Posts and Telecommunications.

Her primary research interests include trustworthy social computing, network security and trust in biomedical systems. Her work on detecting dishonest ratings/feedbacks and

malicious users in online rating systems received the best paper award at the IEEE International Conference on Social Computing (SocialCom'10, acceptance ratio <13%). She has applied trust in biomedical systems, in which her research goal is to improve reliability and security of the neutral control of artificial leg and ensuring safety of amputees.

Fan Zhang (S'11) received his B.S. degree in biomedical engineering from Tianjin Medical University, Tianjin, China, in 2006, and his M.S. degree in biomedical engineering from Tianjin University, Tianjin, China, in 2008.

He is currently pursuing the Ph.D. degree in biomedical engineering at the University of Rhode Island, Kingston, RI, USA. His research interests include signal processing, pattern recognition, and human motion analysis.

Jin Ren (S'09) received the bachelor's degree in material science and engineering and the master's degree in computer science from Huazhong University of Science and Technology, China, in 1999 and 2002, respectively. He received his Ph.D. degree in Electrical, Computer, and Biomedical Engineering at the University of Rhode Island, Kingston, in 2011.

He is currently a Research Scientist at VeloBit Inc. His research interests include data storage and recovery, GPU high performance computing, and virtual machine technology.

Yan (Lindsay) Sun (M'04) received her B.S. degree with the highest honor from Peking University in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Maryland in 2004. She joined the University of Rhode Island in 2004, where she is currently an associate professor in the department of Electrical, Computer and Biomedical Engineering. Dr. Sun is the Director of Network Security and Trust (NEST) Laboratories. Dr. Sun's research interests include trustworthy social computing, network security and wireless networking. She is an associate editor of IEEE Signal Processing Magazine since 2010. She is the recipient of NSF CAREER Award.

Qing Yang (M'88-SM'96-F'09) received the MS degree from University of Toronto and the PhD degrees from University of Louisiana, Lafayette, in 1985 and 1988 respectively. He is currently a Distinguished Engineering Professor in the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island where he has been a faculty member since 1988. His research interests include Computer Architectures, Parallel and Distributed Computing, I/O architectures and Data Storage, Storage Networking, Embedded Computer Systems and Applications, Neural-Machine Interface, Computer application in biomedical engineering. He is an IEEE fellow and a member of the ACM SIGARCH.

He (Helen) Huang (S'03-M'06) received a B.S. from the School of Electronics and Information Engineering at Xi'an Jiao-Tong University, China and a M.S. and Ph.D. degree from the Harrington Department of Bioengineering, Arizona State University. She was a research associate in the Neural Engineering Center for Artificial Limbs at the Rehabilitation Institute of Chicago. Currently she is an Assistant Professor at the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island.

Her primary research interests include neural-machine interface, modeling and analysis of neuromuscular control of movement in normal and neurologically disordered humans, virtual reality in neuromotor rehabilitation, and design and control of therapeutic robots, orthoses, and prostheses. She is a member of the IEEE Engineering in Medicine and Biology Society and Society of Neuroscience.
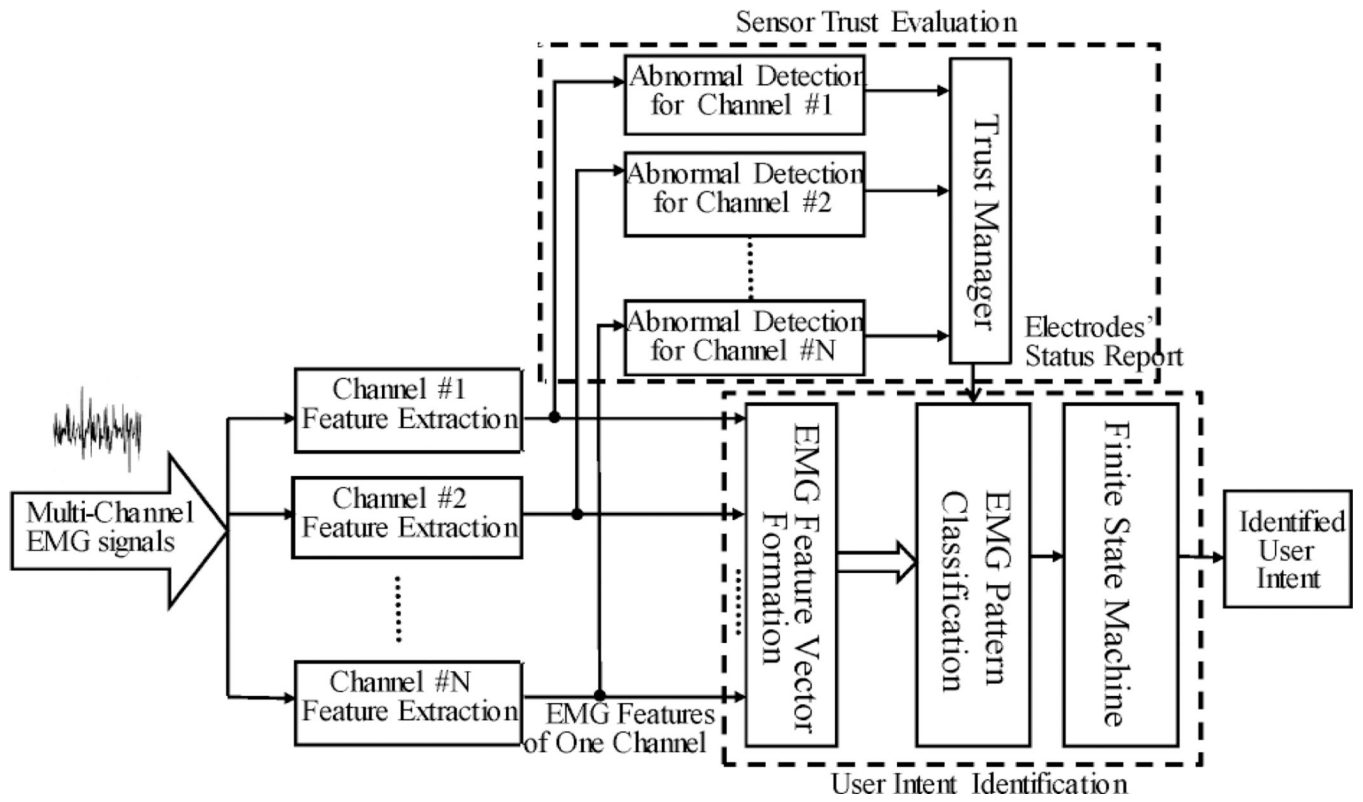
Fig. 1.
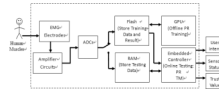Software architecture of EMG-based neural-machine interface for artificial legs.

Fig. 2.
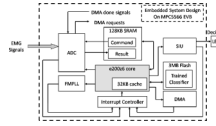Hardware architecture of designed neural-machine interface.

Fig. 3.
Block diagram of embedded system design on MPC5566 EVB for real-time testing.
MPC5566: device modules; ADC: analog-to-digital converter; FMPLL: frequency
modulated phase-locked loop; SRAM: internal static RAM; SIU: system integration unit;
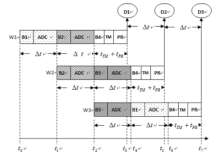DMA: direct memory access.

Fig. 4.
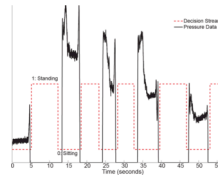Timing control of real-time decision making

Fig. 5.
Real-time performance of the designed NMI system. The decision stream (0: sitting, 1: standing) is aligned with the pressure data (black solid line) measured under the gluteal region of the subject.

Fig. 6.
Real-time testing of the designed NMI prototype on one patient with transfemoral amputation (lower two photos). Upper two photos show our experiments on an able-bodied subject.
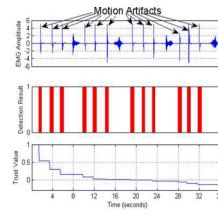
Fig. 7.
Real time performance demonstration of the abnormal detector under motion artifacts. The representative EMG signals(upper panel), the detection results of CUSUM (middle panel), and the trust value (lower panel) are demonstrated

Table 1

System classification response time

RT1	RT2
+(364±38) ms	−(875±27) ms

Note: '+' represents the classification decision was made after the event (non-weight bearing sitting to weight-bearing standing); '−' means the classification decision was made before the event (weight-bearing standing to non-weight bearing sitting)

Table 2

Speedups of our GPU parallel training algorithm over the 3GHz PC server.

Window size	100	200	400	600	800
Speedup	22.98	29.50	35.94	37.16	39.21