Real-Time Classification of Electromyographic Signals for Robotic Control

Beau Crawford **Kai Miller** **Pradeep Shenoy** **Rajesh Rao**
Dept. of Computer Science
University of Washington
{beau,kjmiller,pshenoy,rao}@cs.washington.edu

Abstract

Advances in bioengineering have led to increasingly sophisticated prosthetic devices for amputees and paralyzed individuals. Control of such devices necessitates real-time classification of biosignals, e.g., electromyographic (EMG) signals recorded from intact muscles. In this paper, we show that a 4-degrees-of-freedom robotic arm can be controlled in real-time using non-invasive surface EMG signals recorded from the forearm.

The innovative features of our system include a physiologically-informed selection of forearm muscles for recording EMG signals, intelligent choice of hand gestures for easy classification, and fast, simple feature extraction from EMG signals. Our selection of gestures is meant to intuitively map to appropriate degrees of freedom in the robotic arm. These design decisions allow us to build fast accurate classifiers online, and control a 4-DOF robotic arm in real-time.

In a study involving 3 subjects, we achieved accuracies of 92-98% on an 8-class classification problem using linear SVMs. These classifiers can be learned on-line in under 10 minutes, including data collection and training. Our study also analyzes the issues and tradeoffs involved in designing schemes for robotic control using EMG. Finally, we present details of online experiments where subjects successfully solved tasks of varying complexity using EMG to control the robotic arm.

Introduction

Electromyographic (EMG) signals provide an extremely useful non-invasive measure of ongoing muscle activity. They could thus be potentially used for controlling devices such as robotic prosthetics that can restore some or all of the lost motor functions of amputees and disabled individuals. Most commercially available prosthetic devices have limited control (e.g., one degree-of-freedom in the case of a prosthetic gripper), nowhere near the original levels of flexibility of the organ they are intended to replace.

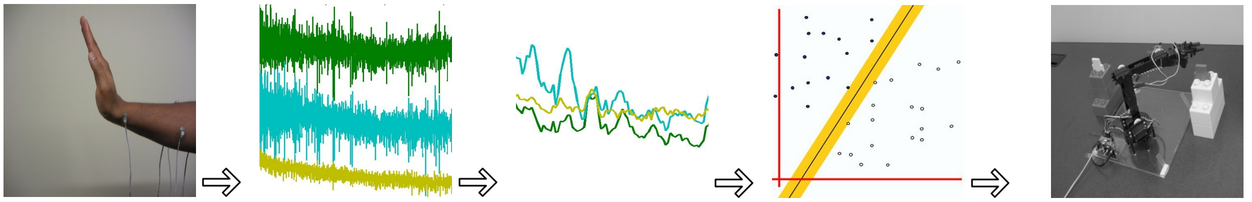
Amputees and partially paralyzed individuals typically have intact muscles that they can exercise varying degrees of control over. In this paper, we address the issue of whether signals from these intact muscles can be used to control robotic devices such as prosthetic hands and limbs with multiple degrees of freedom. We demonstrate that this is indeed possible by showing that activation patterns recorded from muscles in the forearm can be classified in real-time to control a 4 degrees-of-freedom robotic arm-and-gripper. We demonstrate the robustness of our method in a variety of reasonably complex online robotic control tasks involving 3D goal-directed movements, obstacle avoidance, and picking up and accurate placement of objects. The success of our system is based on a combination of several factors: (1) a careful choice of actions to classify, chosen for ease of classification and their intuitiveness for the control task, (2) selection of muscle activity recording sites that are relevant to these actions, (3) the use of a simple, sparse feature representation that can be computed in real-time, and (4) a state-of-the-art classification method based on linear Support Vector Machines (SVMs).

Our results demonstrate that classification accuracies of over 90% can be achieved for an 8-class classification problem. We leverage the information gained from our offline study of action classification to build a real-time online system and show that subjects can successfully and efficiently perform reasonably complex tasks involving all four degrees of freedom of a robotic arm-and-gripper. Our results provide a basis for building not only complex prosthetic devices for disabled individuals, but also novel user interfaces based on EMG signals for human-computer interaction and activity recognition.

Background and Related Work

Muscle contraction is the result of activation of a number of muscle fibers. This process of activation generates a change in electrical potential and can be measured in sum at the surface of the skin as an electromyogram (EMG). The EMG signal is a measure of muscle activity, and its properties have been studied extensively (Deluca 1997). The amplitude of the EMG signal is directly correlated with the force generated by the muscle; however, estimating this force in general is a hard problem due to difficulties in activating a single muscle in isolation, isolating the signal generated by a muscle from that of its neighbors, and other associated problems.

In the field of prosthetics, the EMG signal has been used in one of two ways: The first approach is to use the amplitude of the *steady state* EMG signal where the subject exerts constant effort with a chosen muscle. A single recording site or channel on this muscle is then used as a binary switch, or to control a degree of freedom in a manner proportional



(a) static gesture (b) steady-state EMG (c) extracted features (d) SVM classification (e) robot arm control

Figure 1: Schematic for EMG-based robotic control.

to the amplitude. Many commercially available prosthetics fall into this category and afford the use of a single degree of freedom such as a gripper for grasping and releasing objects. The second approach uses discrete actions, i.e., where the user performs a quick gesture with the hand, wrist or fingers and the temporal structure of the transient EMG activity during action initiation is exploited for classification.

Graupe and Cline (Graupe & Cline 1975) were the first to classify EMG signals. They obtained 85% accuracy in classifying data from a single channel with autoregressive coefficients as features. Engelhart et al. (Engelhart *et al.* 1999) classify four discrete elbow and forearm movements and capture the transient structure in the EMG using various time-frequency representations such as wavelets. They achieve accuracy upto 93.7% with four channels of EMG data from the biceps and triceps. Nishikawa et al. (Nishikawa *et al.* 1999) classify 10 discrete movements of the wrist and fingers using four electrodes placed on the forearm. They propose an online learning scheme, and obtain an average accuracy of 91.5%. Ju et al. (Ju, Kaelbling, & Singer 2000) address applications in user interfaces and consumer electronics. They achieve 85% accuracy in classifying four finger movements with the aid of two electrodes placed close to the wrist. The electrode locations are suboptimal but chosen for appropriateness in the chosen applications.

Thus, much of the recent research has focused on classifying discrete actions, and significant effort has been directed towards design of new and powerful feature representation for the EMG signal. The problem with this approach is that each action initiation is interpreted as one command, leading to a very slow rate of control. In contrast, we work with *steady state* EMG signals, where the user maintains one of a set of hand poses, and windows of EMG signals are then classified continuously as one of these poses. In fact, we generate 16 commands/second for use in controlling a prosthetic device, a rate that is not possible in the other paradigms. We show that in combination with our choice of recording sites, steady-state gestures and powerful classification techniques, we can get high accuracy in an 8-class classification problem, and achieve continuous control of a robotic arm.

System Design

Figure 1 shows the overall design of our system. The user maintains a static hand pose that corresponds to one of a selected set of gestures. We record EMG activity from carefully chosen locations on the user's forearm. This stream of data is transformed into feature vectors over windows of the EMG signals, and classified by a linear SVM classifier. The classifier output serves as a command that moves the robotic arm by a small fixed amount in the desired direction. Thus, maintaining a hand gesture will make the arm move continuously in a chosen direction. In the following sections we describe each of these components in greater detail.

Gestures for Robot Arm Control

Figure 2 shows a list of the actions we use to control each degree of freedom in the robotic arm. Our goal is to map gross actions (distinguishable via forearm musculature) at the wrist to commands for the robotic arm.

The actions we have chosen are appropriate metaphors for the corresponding actions we wish to control in the robotic arm. Our study demonstrates that forearm muscles contain enough information to reliably distinguish between a large number of actions; the interpretation of these actions is left to the user. Thus, for control of other prosthetic devices, one has the option of customizing the actions to better suit the device in question and the desired control. This customization has to be done on a case-by-case basis. Finally, we mention that although our study used healthy subjects, there is evidence (Eriksson, Sebelius, & Balkenius 1998) that amputees who have lost their hand are able to generate signals in the forearm muscles that are very similar to those generated by healthy subjects. We are currently in the process of working with amputees to explore the customization issues involved, and to validate our system in a real-life setting.

Electrode Placement

Our choice of electrode positions was designed to make the interpretation of the signal as intuitive as possible and as reproducible from subject to subject as possible. While no electrode position will isolate a single muscle, placing a given electrode on the skin above a given superficial muscle should ensure that the largest contribution to the signal from that location is from the desired muscle. The muscles of the deep layer will contribute to the signal, as will other surrounding muscles. Since our goal is classification into a discrete set of actions, and not the study of individual muscles, we rely on the classifier to extract the important components for each class from this mixture of information in each electrode channel.

The muscles we chose, and their relevant functions are listed below: (1) Brachioradialis- flexion of the forearm, (2) Extensor carpi ulnaris- extension and adduction of hand

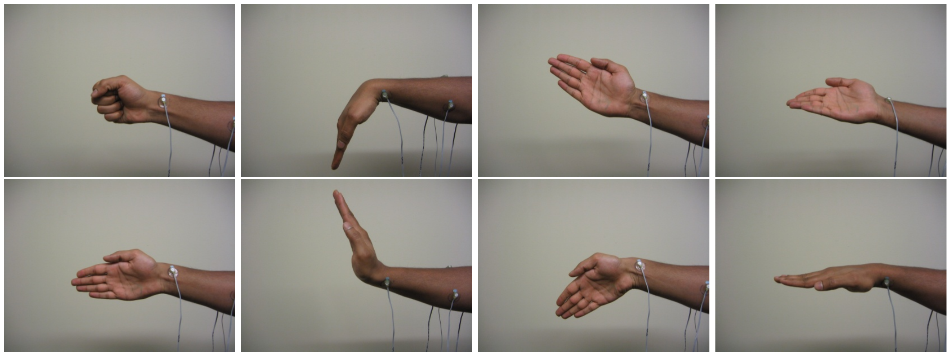


Figure 2: Static hand gestures chosen for controlling the robotic arm. Each column shows the pair of gestures that control a degree of freedom in the robotic arm. These are, in order, grasp and release of the gripper, left-right movement of the arm (top view), up-down arm movement, and rotating the gripper.

at the wrist, (3) Pronator Teres- pronation and elbow flexion (which was not a movement we incorporated), (4) Extensor Communis Digitorum- extension of fingers at the metacarpo-phalangeal joints and extension of wrist at forearm, (5) Flexor Carpi Radialis- flexion and abduction of hand at wrist, (6) Anconeus- antagonistic activity during pronation of the forearm, (7) Pronator Quadratus- initiates pronation.

The combination of these muscles in coordination span the movements that we classify and map to arm control. Because of redundancy amongst the actions of these muscles as well as the redundancy amongst deeper muscles whose signals are measured in conjunction, our channels may contain correlations. This redundancy of information leads to a more robust classifier that can handle the unavoidable variations in electrode placement and quality of EMG recordings across subjects and sessions.

We use single electrodes, in contrast to the differential pair at each recording site traditionally used in the literature (Deluca 1997). Instead we use an eighth electrode on the upper arm as a reference for the other electrodes.

Feature Extraction

We record EMG signals at a sampling rate of 2048Hz. Since we classify steady-state signals from hand poses, we need to classify windows of EMG data from all channels. Our feature representation is simply the amplitude of the steady-state EMG signal from each channel. The feature extraction procedure is thus greatly simplified: we take 128-sample windows from each channel, and compute the amplitude of each channel over this window. The resulting feature vector consists of only 7 values, one per channel. This feature vector serves as the input to our classifier. The choice of 128-sample windows is empirically motivated, and results in 16 commands/second, thus allowing for very fine control of the robotic arm.

Classification with Linear SVMs

We use linear Support Vector Machines for classifying the feature vectors generated from the EMG data into the respective classes for the gestures. SVMs have proved to be a remarkably robust classification method across a wide variety of applications.

We first consider a two-class classification problem. Essentially, the SVM attempts to find a hyperplane of maximum "thickness" or *margin* that separates the data points of the two classes. This hyperplane then forms the decision boundary for classifying new data points. Let $\mathbf{w}$ be the normal to the chosen hyperplane. Then, the classifier will label a data point $\mathbf{x}$ as $+1$ or -1, based on whether $\mathbf{w} \cdot \mathbf{x} + b$ is greater than 1, or less than -1. Here, b is chosen to maximize the margin of the decision boundary while still classifying the data points correctly.

This leads to the following learning algorithm for linear SVMs. For the classifier to correctly classify the training data points $\mathbf{x_1}, ..., \mathbf{x_n}$ with labels $y_1, ..., y_n$ drawn from ± 1, the following constraints must be satisfied:

$$\mathbf{w} \cdot \mathbf{x_i} + b >= 1 \; if \;\; y_i = 1$$
$$\mathbf{w} \cdot \mathbf{x_i} + b <= -1 \;\; if \;\; y_i = -1$$

It can be shown that choosing the hyperplane of maximum margin corresponds to minimizing $\mathbf{w} \cdot \mathbf{w}$ subject to these constraints. Real-life data, however, is noisy and we need to allow for errors in the classifier. This is achieved by relaxing each constraint above with the use of an error term for each point. The total error is then included in the optimization criterion. The distances of misclassified points from the hyperplane are represented by the error variables ξ_i. The set of constraints now reads:

$$\mathbf{w} \cdot \mathbf{x_i} + b + \xi_i \geq 1 \; if \; y_i = 1$$
$$\mathbf{w} \cdot \mathbf{x_i} + b - \xi_i \leq -1 \; if \; y_i = -1$$
$$\xi_i \geq 0 \forall i$$

With these new constraints, the optimization goal for the noisy classification case is to minimize $\frac{1}{2}\mathbf{w}.\mathbf{w} + C\Sigma_i \xi_i$, where C is a user-specified *cost* parameter. Intuitively, the criterion is trading off the margin width with the amount of error incurred. We refer the reader to appropriate texts (Scholkopf & Smola 2002) for more technical details. This is the formulation we use, and in this formulation, the classifier has a single free parameter C that needs to be chosen by model selection.

Multiclass Classification and Probabilities The two-class formulation for the linear SVM can be extended to multiclass problems, e.g., by combining all pairwise-binary classifiers (Hsu & Lin 2002). In our system, we use the LIBSVM (Chang & Lin 2001) package which has support for multiclass classification. There is also support for estimating class-conditional probabilities for a given data point. This can be useful in reducing the number of false classifications due to noisiness in the data. Specifically, the class-conditional probabilities returned can be tested against a threshold, and a "no-operation" command can be executed if the classifier is uncertain about the correct class label for the data point. Indeed, we implement this scheme in our system, for two reasons: firstly, the transitional periods when the user switches between different steady states may generate data that the classifier has not seen, and does not actually correspond to any of the chosen classes. Secondly, although we do not investigate this in our paper, we believe that by using a conservative threshold, the user will adapt to the classifier via feedback and produce gestures that are more easily classifiable.

Offline Experiments

Data Collection

We collected data from 3 subjects over 5 sessions. Each session consisted of the subject maintaining the 8 chosen action states for 10 seconds each, thus providing 160 data points per class. We use 5 sessions in order to prevent overfitting– a given action may be slightly different each time it is performed. All reported errors are from 5-fold leave-session-out crossvalidation, to account for this variation in the actions.

Classification Results

We used the collected data to train a linear SVM classifier, and performed parameter selection using across-session crossvalidation error as measure. Figure 3 shows the SVM classifier error as a function of the cost parameter C. The graph demonstrates two aspects of our system: Firstly, 8-class classification can be performed with accuracy of 92-98% for all three subjects. Secondly, the classification results are stable over a wide range of parameters for all three subjects, indicating that in an online setting, we can use a preselected value for this parameter. It is important to note, however, that this offline preselection is important, as the error can be significant for bad choices of C. We also point out that 10-fold crossvalidation on any one session of data yielded 0-2% errors for all subjects, which is significantly lesser than the across-session error. Since each session is essentially one static hand-pose, this result indicates that overfitting is likely to occur if a single session of data is used for training.

As noted in the previous section, we can threshold the class-conditional probabilities in order to decrease the false classification rate. In an online scenario it will simply add to the time taken for task performance, but will not cause errors.

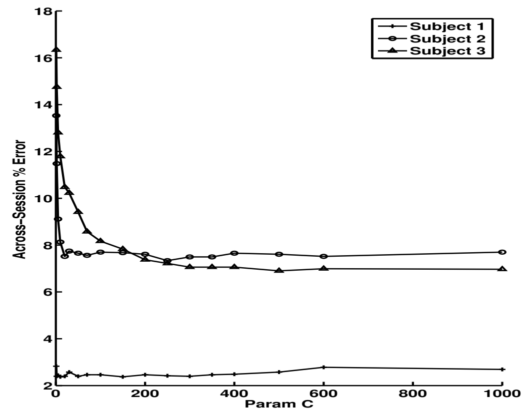


Figure 3: Classifier error on the 8-gesture classification problem as a function of the SVM cost parameter C

Evaluating Choice of Recording Locations

In earlier sections, we noted that our choice of recording sites for muscle activity is motivated by the relevance of the chosen muscles to the gestures we wish to classify. We now address the following questions pertaining to this selection: Firstly, do all the channels of EMG data contribute to the classification accuracy–i.e., is there redundancy in our measurements? Secondly, how many channels or electrodes would we need for controlling a given number of degrees of freedom with high accuracy?

Figure 4 addresses these two questions. We study performance of the linear SVM classifier on problems involving various subsets of classes, as we drop the number of channels used for classification. For any single classification problem, the channel to drop at each step was chosen using a greedy heuristic. That is, at each step, the channel dropped was the one that least increased the cross-validation error of the classifier trained on the remaining channels. This feature-selection procedure was carried out for the following classification problems: (1) grasp-release, (2) left-right, (3) left-right-up-down, (4) left-right-up-down-grasp-release, and (5) all 8 classes. These choices represent control of an increasing number of degrees of freedom.

The figure clearly illustrates the following two points: Firstly, as expected, more degrees of freedom require more channels of information for accurate classification. For example, the 2-class classification problems need only one or two channels of information, but the 6-class problem requires 3 or more channels for a low error rate. Secondly, the full 8-class classification problem can be accurately solved with fewer than 7 electrodes. However, the order in which the channels were dropped was very different for each subject. We ascribe this to the variation in the performance of actions by different individuals and in the recordings obtained from these individuals. Thus our results do not favor dropping of any one channel; however, the redundancy in information contained in the channels makes our classifiers more robust across subjects.

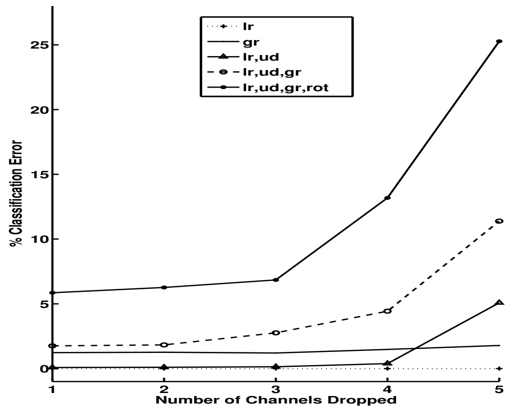


Figure 4: Classifier Error as a function of number of channels dropped, from an initial set of 7 channels. The legend describes the degrees of freedom included in the classification problem: lr=left-right, ud=up-down, gr=grasp-release, rot=rotate.

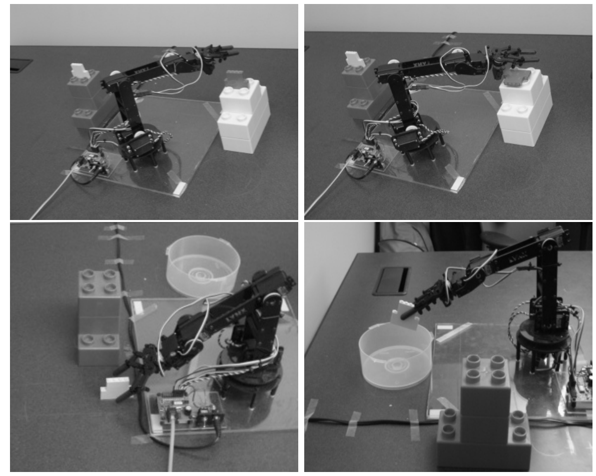


Figure 5: The simple and intermediate online tasks. The first row shows the simple task, where the robot arm starts in the middle, and the goal is to topple the two objects placed on either side. The second row shows the intermediate task, where the goal is to pick up a designated object, carry it over an obstacle, and drop it in the bin.

Online Experiments

Online Procedure: We had the 3 subjects return for a second study and perform various tasks online. We studied the performance of the subjects on various tasks of increasing complexity. The subjects were once again connected to the EMG recording system, 5 sessions of training data were recorded and the SVM classifier was trained online with these 5 sessions and a parameter value that was recommended by the offline study. The process of collecting training data took 10 minutes, and the classifier was trained in less than a minute.

Task Selection: We chose 3 different tasks for our study; simple, intermediate and complex. The simple task is a gross-movement task that only requires the subject to move the robotic arm to two specific locations in succession and knock off objects placed there. Here the goal is to test basic reach ability where fine control is not necessary. The intermediate task involves moving the robotic arm to a designated object, picking it up, and carrying it over an obstacle to a bin where it is then dropped. In this task, accurate positioning of the arm is important, and additional commands are used for grasp/release. Figure 5 describes the first two tasks in more detail. The third, complex, task involves picking up a number of pegs placed at chosen locations, and stacking them in order at a designated place. This requires very fine control of the robotic arm both for reaching and picking up objects, and also for placing them carefully on the stack. Figure 6 illustrates the complex task.

Measure and Baseline: The measure used is the time taken to successfully complete a task. Each subject performed each task 3 times, and the average time across trials was recorded. For the third task, only two repetitions were used since the task is composed of four similar components. We use two baselines to compare against: firstly, we measured the theoretical time needed to perform these tasks by counting the number of commands needed for the robotic arm to perform a perfect sequence of operations, and assuming that the task is accomplished at the rate of 16 commands/s. Secondly, we had a fourth person perform these same tasks with a keyboard-based controller for the robotic arm. The second baseline is used to estimate the cognitive delays in planning various stages of a task as well as the time spent in making fine adjustments to the arm position for picking and placing pegs. Naturally, there are differences in the skill of any given subject at performing the task, as well as a learning component wherein the same subject can improve their performance at the task. These issues are, however, beyond the scope of this paper; here we wish to demonstrate that the EMG-to-command mapping is robust and provides a good degree of control over the robotic arm.

Online Task Performance

Figure 7 shows the performance of the three subjects and the baselines on the three online tasks. For the simple task, involving gross movements, all subjects take time close to the theoretical time required. Interestingly, the keyboard-based control takes less time, since keeping a key pressed continuously can generate more than our EMG-controller's control rate of 16 cmds/sec. For the intermediate task, since a moderate amount of precision is required, the keyboard baseline and the three subjects take more time than the theoretical time needed. Finally, for the complex task, there is a clear difference between the theoretical time and the time taken by the three subjects. It is interesting to note that the keyboard-based control also takes a comparable amount of time, thus showing that the bottleneck is not the control scheme (keyboard or EMG-classification based control), but the task complexity.

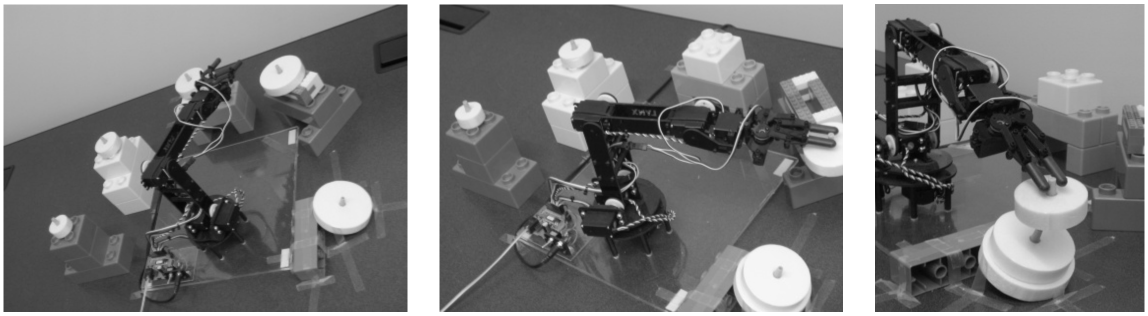


Figure 6: The complex online task: Five pegs are placed at various fixed locations, and the goal is to stack them according to their size. The pictures show, in order, the initial layout, an intermediate step in moving a peg to the target location, and the action of stacking the peg.

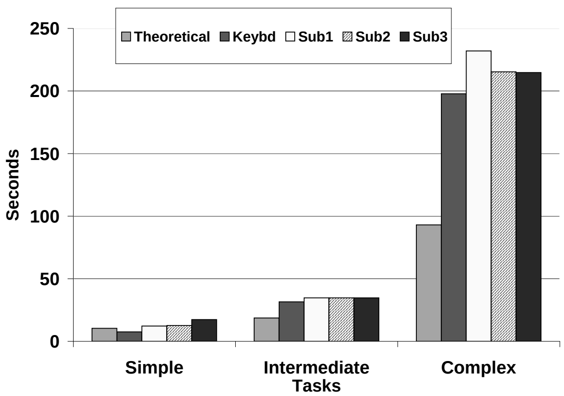


Figure 7: Performance of three subjects using the EMG-based robotic arm control for 3 online tasks. The graph includes the baselines of theoretical time required, and time taken with a keyboard-based controller.

Conclusions and Future Work

We have shown that EMG signals can be classified in real-time with an extremely high degree of accuracy for controlling a robotic arm-and-gripper. We presented a careful offline analysis of an 8-class action classification problem based on EMG signals for three subjects as a function of the number of recording sites (electrodes) used for classification. Classification accuracies of over 90% were obtained using a linear SVM-based classifier and a sparse feature representation of the EMG signal. We then demonstrated that the proposed method allows subjects to use EMG signals to efficiently solve several reasonably complex real-time motor tasks involving 3D movement, obstacle avoidance, and pick-and-drop movements using a 4 degrees-of-freedom robotic arm.

Our ongoing work is focused on extending our results to other types of movements, e.g, discriminating finger movements. A separate effort is targeted towards replicating the results presented in this paper with actual amputees in collaboration with the Rehabilitation department at our university. A parallel study involves combining EEG signals from the scalp (reflecting underlying brain activity) with EMG signals for more accurate classification of motor patterns, with potential applications in brain-computer interfaces (BCIs). An interesting theoretical question that we are beginning to study is whether the EMG-based control system can be adapted online rather than only at the start of an experiment. This is a difficult problem since the subject is also presumably adapting on-line to generate the best muscle activation patterns possible for control and to compensate for changes in electrode conductivity with the passage of time. We intend to explore variations of our SVM-based classification technique to tackle this challenging non-stationary learning problem.

References

Chang, C.-C., and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Deluca, C. 1997. Surface electromyography detection and recording. In *Neuromuscular Research Center, Boston University*.

Engelhart, K.; Hudgins, B.; Parker, P.; and Stevenson, M. 1999. Classification of the myoelectric signal using time-frequency based representations. In *Medical Engg. and Physics*, volume 21, 431–438.

Eriksson, L.; Sebelius, F.; and Balkenius, C. 1998. Neural control of a virtual prosthesis. In *ICANN*.

Graupe, D., and Cline, W. 1975. Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes. In *IEEE Trans. Systems Man and Cybernetics*, volume 5, 252–259.

Hsu, C.-W., and Lin, C.-J. 2002. A comparison of methods for multi-class support vector machines. In *IEEE Trans. Neural Networks.*, volume 13, 415–425.

Ju, P.; Kaelbling, L.; and Singer, Y. 2000. State-based classification of finger gestures from electromyographic signals. In *ICML*.

Nishikawa, D.; Yu, W.; Yokoi, H.; and Kakazu, Y. 1999. Emg prosthetic hand controller discriminating ten motions using real-time learning method. In *IEEE/RSJ IROS*.

Scholkopf, B., and Smola, A. 2002. *Learning with Kernels: Support vector machines, regularization, optimization and beyond*. MIT Press, Cambridge MA.