Discrete Classification of Upper Limb Motions Using Myoelectric Interface

Chris Wilson Antuvan[1], Federica Bisio[2], Erik Cambria[3] and Lorenzo Masia[1]

Abstract— **Electromyografic signals offer insights into understanding the intent and extent of motion of the musculoskeletal system. This information could be utilized in developing controllers for applications such as prostheses and orthosis, and in general assistive technology. This paper presents a myoelectric based interface to control five discrete upper limb motions involving the shoulder and elbow joint. Four subjects performed the experiment, which consisted of two separate phases: the training and testing phase. Extreme Learning Machine algorithm is used to classify the myoelectric signals to the control motions. The data collected during the training phase is used to train the parameters of the decoder, and the data from the testing phase is used to quantify the performance of the decoder. The muscle activations of each subject are used to manipulate a virtual human avatar. The graphical visualization serves to provide real-time feedback of the motions generated. The performance of the decoder for both offline and online classification are evaluated. Results indicate an overall classification accuracy for online control being $78.96 \pm 23.02\%$. The rate of transition from rest phase to the desired motion phase, on an average is 0.25 ± 0.10 seconds.**

I. Introduction

There has been an increasing popularity in recent years towards myoelectric based interfaces, especially in the field of Prostheses [1], [2], Orthotics [3], [4] and Tele-manipulation [5]–[8], offering recovery of lost functionality, in providing assistance or in augmenting capabilities. With recent technological advancements, it has been possible to use surface electrodes to gather useful information from the muscle. Ever since its inception in the application of human-machine interaction, there has been a huge amount of research contribution utilizing this medium for human embedded control [9].

Electromyogram (EMG) signals are considered a proxy to the signals sent by the brain and spinal cord. These signals are non-repetitive, and changes with time due to electrode shift, fatigue or sweat. As such mapping the myoelectric signals to control outputs, using machine learning algorithms is challenging. The accuracy depends highly on the features derived from the EMG signals, and the variety of the dataset used for training [10], [11]. Increasing the number of data leads to an increase in the time required for training and initial calibration. The traditional learning algorithms are slow in terms of converging to the nominal solution, and the accuracy is not so high for real-time performance. Using regression based decoder for continuous decoding of joint angles or torques gets highly complicated and the performance degrades considerably, as the number of degrees of freedom increases. For that reason, we would like to focus on discrete classification, which has been widely applied especially in assistive technologies. The learning algorithm used for classification is extreme learning machine (ELM), which has the capability for faster training, less degree of intervention and ease of implementation [12]–[14].

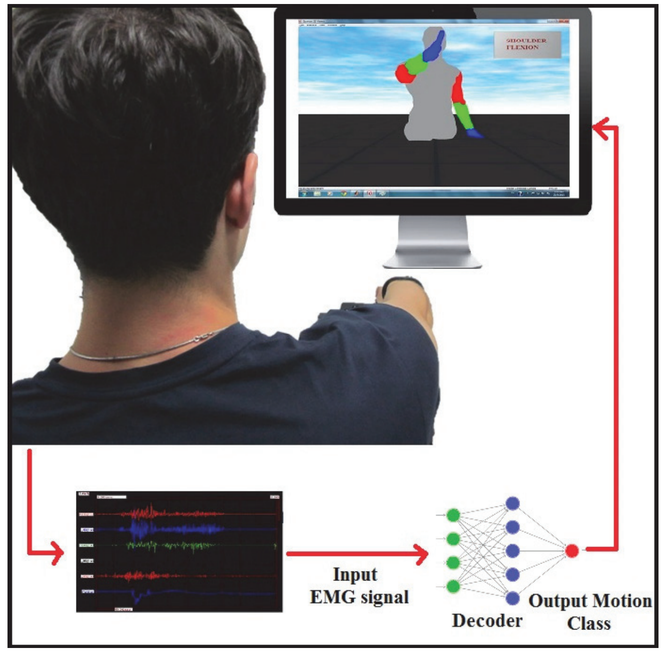

Fig. 1. Experimental setup showing the Real-Time Testing phase

This paper focuses on trying to decode discrete upper arm motions, especially the shoulder and elbow joint motions using EMG signals. The main aim is to quantify the performance of this particular machine-learning algorithm. A training phase is used to evaluate the parameters of the decoder, which is used by the succeeding testing phase for real-time motion prediction. The trajectory-motion for the virtual arm is characterized by minimizing the Jerk function, for smooth human-like movement. The rest of the paper is organized as follows. Section II explains the experimental methodology, the instrumentation used, the concept of ELM and Minimum Jerk Trajectory, the feature set used for the EMG signals, and the metrics used for quantifying the performance. Section III presents the results of the experiments,

[1]Chris Wilson Antuvan and Lorenzo Masia are with the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore Email: chriswil001@e.ntu.edu.sg, lorenzo.masia@ntu.edu.sg

[2]Federica Bisio is with the Department of Naval, Electrical, Electronic and Telecommunications Engineering at University of Genoa, Genoa, Italy Email: federica.bisio@edu.unige.it

[3]Erik Cambria is with the School of Computer Engineering at Nanyang Technological University, Singapore Email: cambria@ntu.edu.sg

978-1-4799-1808-9/15/$31.00

TABLE I
THE SEVEN MUSCLES USED AND THE OUTPUT MOTION CLASSES

MUSCLES
Biceps Brachii (Long head)
Triceps Brachii (Lateral head)
Anterior Deltoid
Posterior Deltoid
Pectoralis Major
Infraspinatus
Trapezius

MOTION CLASS
Elbow Flexion
Elbow Extension
Shoulder Protraction
Shoulder Retraction
Shoulder Flexion

and Section IV presents the observations and conclusions regarding the experiments, and the future implementation of the algorithm for assistive technology.

II. METHODS

A. Experimental Protocol

Myoelectric signals from seven muscles of the shoulder and upper arm, are utilized for classifying the five different classes of motion. The seven muscles used and the respective motions are as shown in Table I and Figure 2. The whole experiment is done on the same day and was split into two phases:
1) Training phase: It involves the acquisition of EMG signals for calibrating the parameters of the decoder. A custom made Matlab graphical user interface (GUI) was designed to aid the acquisition procedure. Instructions are displayed on the GUI, requesting the subject to perform specific motions for a certain period of time. Each of the five different *motion class* is performed 10 times, alternating between 3 seconds of Motion phase and 2 seconds of Rest phase. Also there is a 10 second Rest period between the end of a *motion class* and start of a new *motion class*, in order to prevent muscular fatigue for the subject.

2) Testing phase: This follows subsequent to the Training phase, after the parameters of the decoder are calibrated. The muscular activity generated by the movement of the subject is translated through the decoder, to equivalent movements of the virtual arm. The acquisition of EMG signals, its processing, the classification algorithm, and the trajectory motion of the virtual arm are all done in real-time. Instructions, prompting the subject to perform a specific *motion class* are displayed on the screen in a random fashion. Each motion is performed 10 times, always alternating between motion phase and three seconds of rest phase. The motion phase ends when the subject moves the virtual arm to the designated target position and maintains the position for at least 0.5 seconds.

A total of 4 healthy subjects participated in the experiment, and they all gave informed consents. And the procedures were approved by the Institution Review Board.

B. Experimental Setup

The setup includes Wireless Electromyogram system (Trigno wireless, Delsys Inc.), which was used to record the surface EMG signals. Seven wireless electrodes were placed on the respective electrode sites on the surface of the shoulder and upper arm. A real-time Data acquisition board (Quanser QPIDe) is used to acquire the EMG signals as analog inputs, and the acquisiton frequency is 1 kHz. A MATLAB Simulink based custom program is then used to interface the myoelectric signals. The processing of the signals and pattern classification algorithm are implemented in real-time, and is then used to manipulate the object in the virtual environment, thereby providing visual feedback to the subject.

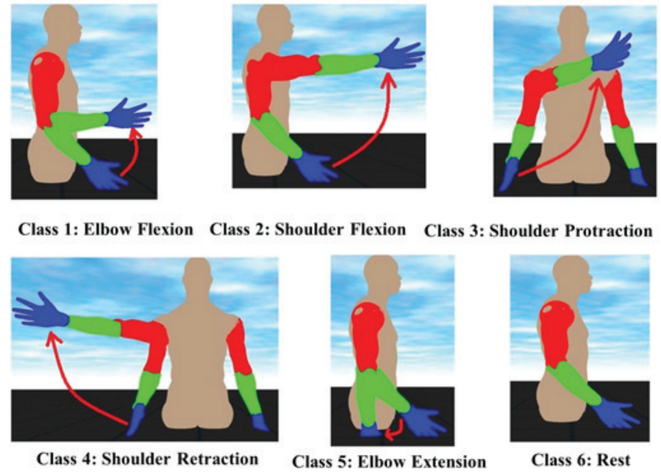

Fig. 2. The five different Motion Class and the Rest pose which represents the sixth class.

C. Machine Learning Algorithm

Extreme learning machine is an emerging learning technique that provides efficient unified solutions to generalized feed-forward networks including (but not limited to) single-/multi-hidden-layer neural networks, radial basis function networks, and kernel learning. ELMs offer significant advantages such as fast learning speed, ease of implementation, and minimal human intervention. They thus have strong potential as a viable alternative technique for large-scale computing and machine learning in many different application fields, including image, text, and speech processing, but also cognitive learning and reasoning. The ELM model [15] implements a single-hidden layer feedforward neural network (SLFN) with N mapping neurons. The neuron's response to an input stimulus, $\mathbf{x}$, is implemented by any nonlinear piecewise continuous function $a(\mathbf{x}, R)$ (activation function), where R denotes the set of parameters of the mapping function. Thus, the function connecting the input layer with the hidden layer can be expressed as follows, for each neuron $j \epsilon \{1, ..., N\}$:

$$h_j(\mathbf{x}) = \mathbf{a}(\mathbf{x}, R_j) \tag{1}$$

The overall output function connecting the hidden and the output layer is expressed as

$$f(\mathbf{x}) = \sum_{\mathbf{j=1}}^{\mathbf{N}} \mathbf{w_j h_j}(\mathbf{x}) \quad (2)$$

,where $\mathbf{w_j}$ denotes the weight that connects the $\mathbf{j^{th}}$ neuron with the output.

The peculiar aspect of ELM, though, is that the parameters R_j are set randomly. Hence, if one uses, for example, classical Radial Basis Functions (RBFs) to implement $\mathbf{a}(.)$:

$$a(\mathbf{x}, R) = exp(-\zeta\|\mathbf{x} - \mathbf{c}\|^2) \quad (3)$$

the parameters to be set randomly are the coordinates of each centroid, $\mathbf{c} \in \mathbb{R}^{\mathbf{Z}}$, and the quantity ζ.

Accordingly, in general, the training process reduces to the adjustment of the output layer, i.e., setting the vector $\mathbf{w} \in \mathbb{R}^{\mathbf{N}}$ in (2). As a result, training ELMs is equivalent to solving a regularized least squares (RLS) problem in a linear space; the eventual minimization problem can be expressed as

$$\underset{w}{min}\{\|\mathbf{y} - \mathbf{Hw}\|^2 + \lambda\|\mathbf{w}\|^2\} \quad (4)$$

The vector of weights $\mathbf{w}$ is then obtained as follows:

$$\mathbf{w} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-\mathbf{1}}\mathbf{H}^T\mathbf{y} \quad (5)$$

Here, H is a $P \times N$ matrix with $h_{ij} = h_j(\mathbf{x_i})$, P is the number of training patterns and λ is a regularization parameter.

The ELM model can be conveniently described as a 2-stage learning machine. In the first stage, the data originally lying in the Z-dimensional space are remapped into a new N-dimensional space (ELM feature space) by exploiting as many random neurons. Then, an RLS problem is solved for learning the linear classifier in the N-dimensional space.

For the binary classification applications, the final decision function of ELM is

$$f_L(\mathbf{x}) = \mathbf{sign}(\mathbf{f}(\mathbf{x})) \quad (6)$$

For multiclass problems, one can set multi-output nodes instead of a single-output node: m-class classifiers have m output nodes. If the original class label is c, the expected output vector of the m output nodes is $y_i = [0, ..., 0, 1, 0, ..., 0]^T$, with the c^{th} element of $y_i = [y_{i1}, ..., y_{im}]^T$ set to one, while the rest of the elements are set to zero. The classification problem for ELM with multi-output nodes can be formulated as:

$$\underset{w}{min}\{\frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{2}\sum_{i=1}^{P}\|\zeta_i\|^2\} \quad (7)$$

Subject to: $h(\mathbf{x_i})\mathbf{w} = \mathbf{y_i^T} - \zeta_\mathbf{i}^\mathbf{T}\ \mathbf{i = 1, ..., P}$

Where $\zeta_i = [\zeta_{i1}, ..., \zeta_{im}]^T$ is the training error vector of the m output nodes with respect to the training sample $\mathbf{x_i}$.

Therefore, the single-output node case can be considered a specific case of multi-output nodes when the number of output nodes is set to one: $m = 1$. For both cases, the hidden layer matrix H remains the same, and the size of H is only decided by the number of training samples P and the number of hidden nodes N, which is irrelevant to the number of output nodes (number of classes) m.

D. Motion Planning

The Central Nervous System (CNS) performs a smooth motion when moving the arm (for this specific case, and in general other body parts) from one point to another. The CNS does this smooth motion by minimizing the sum of the squared jerk along its trajectory. Hogan et al. [16] found that smoothness can be quantified as a function of jerk, which is the time derivative of acceleration, $\dddot{x}(t)$. The Jerk cost can be expressed as:

$$H(x(t)) = \frac{1}{2}\int_{t=0}^{0.5} \dddot{x}^2 dt \quad (8)$$

The trajectory having the lowest jerk cost will have the smoothest trajectory. A function $x(t)$ having its sixth derivative equal to zero is considered to minimize the jerk function. The general solution to the function having the sixth derivative to be zero is given by:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (9)$$

The six unknown parameters $a = a_0, ..., a_5$ can be solved by using the six boundary conditions involving position, velocity and acceleration for both the initial and final conditions. The movement of the arm in the virtual environment is based on the minimum jerk motion concept. This ensures that a smooth motion is developed even when there is rapid fluctuation between different Motion classes, and also gives a realistic feeling of the movement to the subject.

E. Preprocessing and Feature Extraction

The myoelectric signals are acquired at 1 KHz sampling rate, and they are rectified and low-pass filtered with a cut-off frequency of 10 Hz in order to obtain the envelope of the signal. The transient part of the myoelectric activity in each motion is not included for training the decoder. The work by Hargrove et al. [17] shows better decoding accuracy while using only the steady state part of the muscle activity, rather than the entire part consisting of the transient and the steady state activity. Two time-domain features were used to extract meaningful information out of the EMG signal. A sliding window of 100 ms (100 samples) was used to calculate the feature vectors at each time step. The features used are:

Mean Absolute Value (MAV): It is calculated by taking the average value of the EMG signals in the window frame.

$$MAV = \frac{1}{N}\sum_{n=1}^{N}|x_n| \quad (10)$$

Variance (VAR): It is the mean value of the square of the standard deviation of the EMG signals.

$$VAR = \frac{1}{N-1}\sum_{n=1}^{N}|{x_n}^2| \tag{11}$$

F. Performance Metrics

Quantifying the performance of the learning algorithm can be broadly grouped into two groups:

Offline Performance: This involves calculating the decoding accuracy of the learning algorithm during the training phase. The data collected during the Training phase is split into a training and a testing data-set. The decoder is trained using the training set and the classification accuracy is calculated using the testing data-set. Classification accuracy is calculated as:

$$\frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ testing\ samples} \times 100(\%) \tag{12}$$

Online performance: This is used to quantify the performance during the online testing phase, where the subject is performing the task in the virtual environment. These performance metrics are based on the Li et al. [18], where they used similar measures to quantify decoding performance.

1) Motion Selection Time: It is the time taken by the subject to correctly select the target motion class. It gives a measure of how fast the motor commands sent by the brain and consequently to muscular activity, could be translated to the target motion class. The time is calculated as the period between the first transition from rest motion class to any movement class, and the first time the decoder produces the target motion class.
2) Motion Completion Time: This is the time period between the movement onset class and the time taken to reach the target position of the motion class and hold position for 0.5 seconds. There was no time limit set, in order to complete the task.
3) Learning Trend: This factor is used to identify if there is any improvement in performance, as the number of attempts in a trial progresses. It is quantified by fitting a linear line to the log value of the completion time in each trial. The slope of the line indicates whether there is a learning trend or not. Learning trend generally follows an exponential decay curve which is given by:

$$y(x) = a \times 10^{-bx} + c \tag{13}$$

where $x \,\epsilon\, \{1, 2, ..., 20\}$ is the number of repetitions in a trial, y is the time taken to complete the task corresponding to x. a, b, c represents the initial performance, learning rate, and the steady state value.
4) Online Classification Accuracy: This metric is used to calculate how precise the subjects were in completing a particular task. It calculates the number of times (in samples) the decoder outputs the desired motion class, to the total number of samples in that particular motion trial.

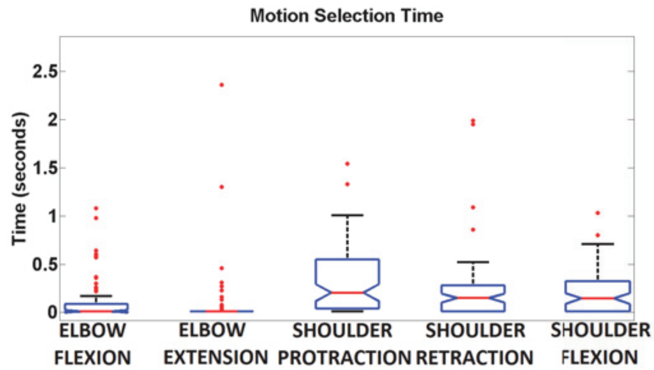

Fig. 3. Plot shows the average Motion Selection Time for each Motion Class across all subjects

III. Results

A. Offline Performance

Classification accuracy has been calculated for each subject across all the five different motion classes. The accuracy for all the four subjects is always above 99%. The average accuracy across all the subjects is found to be $99.81\% \pm 0.0726\%$. A validation data-set was used to decide the number of hidden neurons and the type of activation function to be used. The number of hidden neurons to be used was decided to be 5000. Radial basis function was the activation function used in the network model.

B. Online Performance

The *Motion Selection Time*, *Motion Completion Time*, and *Online Classification Accuracy* values for each subject and each *motion class* across all the attempts are as shown in Table II.

1) Motion Selection Time: The average time taken to correctly select the desired motion class, across all subjects and class is around 0.2593 ± 0.1067 seconds. The selection time for each motion task across all the subjects is shown in Fig. 3. The average time taken for the shoulder movements is found to be more compared to the motions just involving the elbow. A t-test was performed and it was determined that there was significant difference in the selection time between the shoulder and elbow motions ($p = 4.0942e^{-4}$).
2) Motion Completion Time: A time limit in completing each motion task was not set, so the subjects had to figure out how to complete each motion before they could move on to the next one. The average time taken to complete all the task across all the subjects is around 6.9325 ± 9.9052 seconds. There seems to be a lot of variation in the time taken, mainly in the shoulder motions, and especially the Shoulder Retraction, where most of the subjects struggled. The shoulder motions are the complicated motions and involve the coordination of more than one muscle, except for the Shoulder Flexion motion, which was the easiest shoulder motion

TABLE II

RESULTS

	Motion Selection Time (sec)				Motion Completion Time (sec)				Online Classification Accuracy (%)			
Motion Type	Sub 1	Sub 2	Sub 3	Sub 4	Sub 1	Sub 2	Sub 3	Sub 4	Sub 1	Sub 2	Sub 3	Sub 4
ELbow Flexion	0.014	0.370	0.061	0.010	6.004	3.895	2.788	3.536	80.160	92.669	98.046	90.788
ELbow Extension	0.053	0.010	0.066	0.393	3.192	5.739	4.578	4.938	86.525	76.146	81.022	72.827
Shoulder Protraction	0.383	0.072	0.368	0.330	9.932	7.917	4.856	7.208	62.887	84.328	84.247	66.937
Shoulder Retraction	0.144	0.023	0.346	0.972	16.740	5.865	22.712	10.179	73.746	81.207	36.121	67.544
Shoulder Flexion	0.091	0.948	0.278	0.249	4.393	5.412	4.146	4.615	87.093	88.975	83.862	84.208

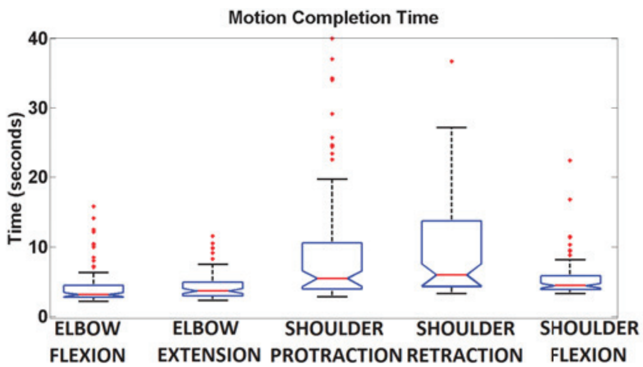

Fig. 4. Plot shows the average Motion Completion Time for each Motion Class across all subjects

TABLE III

LEARNING RATE VALUES FOR ALL THE SUBJECTS FOR EACH OF THE MOTION CLASS

Subject	Elbow Flexion	Elbow Extension	Shoulder Protraction	Shoulder Retraction	Shpulder Flexion
Subject 1	-0.0327	-0.0148	-0.0274	-0.0428	-0.0123
Subject 2	-0.0050	0.0171	-0.0154	0.0192	-0.0115
Subject 3	-0.0035	0.0030	0.0023	0.0290	0.0080
Subject 4	-0.0041	0.0067	-0.0041	-0.0085	0.0048

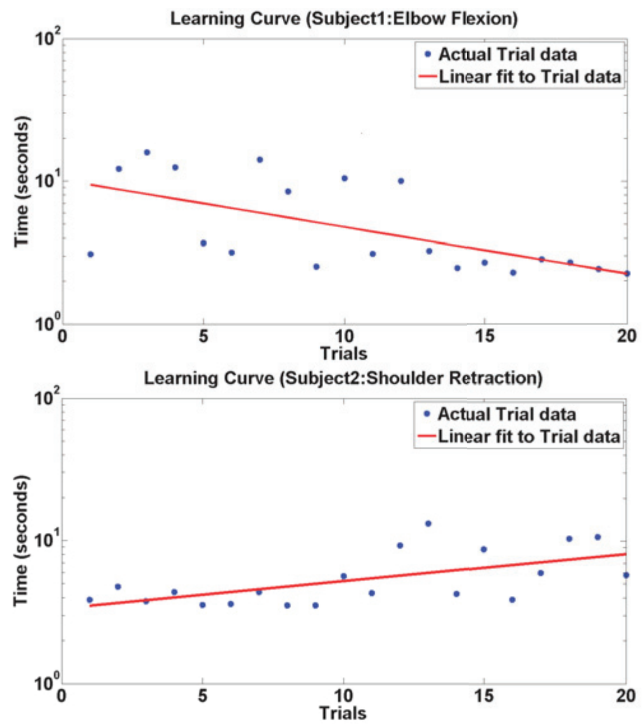

Fig. 5. Learning trend shown in semilog scale: The plot on the top shows positive learning, whereas the plot on the bottom shows negative learning

for all the subjects. The plot in Fig. 4 shows the mean and spread of the time taken to complete the task. The significance in the completion time between the shoulder and elbow motions is determined by performing a t-test ($p = 1.5160e^{-5}$).

3) Learning Trend: The learning rate values for each of the trials for all the subjects are as shown in Table III. A negative value indicates the occurrence of learning, leading to an improvement in performance, and contrary for the case when the value is positive. There is no consistent indication for learning happening across all the subjects for each Motion Tasks, except for the Elbow Flexion motion. It shows that the subjects had difficulty figuring out the activation patterns used during the Training phase. Figure Fig. 5shows the learning curve for two subjects: one performing Elbow Flexion motion and the other Shoulder Retraction motion.
4) Online Classification Accuracy: The performance of the decoder for real-time motion control is found to be good. The average classification performance of the decoder for all the subjects, and across all the Motion classes is $78.9673 \pm 23.0261\%$. The decoding accuracy is the highest in the case of Elbow Flexion motion,followed by Shoulder Flexion motion. The performance is the worst for Shoulder Retraction motion, mainly due to the intrinsic complexity of activation of muscles for this motion. Figure Fig. 6 shows the performance trends across all subjects for each *motion class*.

Something, that is interesting to find from the myoelectric signal patterns recorded during the training phase is that, subjects tend to co-contract muscles while performing the experiments. For example, Elbow Extension motion could be performed by just activating the Triceps Brachii muscle, or also by co-contracting both the Biceps Brachii and the Triceps Brachii (with more activation for the Tricep muscle). This is some-

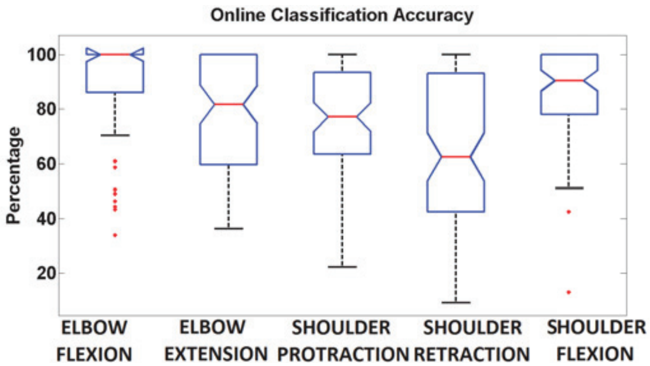

Fig. 6. Plot indicates the classification accuracy for each motion class across all the subjects.

thing that the subjects do not realize, while performing the calibration experiments. The degree of complexity, in terms of characterizing motion intention increases as we progress from a more distal motion (such as hand movements) to a proximal motion (such as shoulder movements).

IV. Conclusion and Future Directions

The classification accuracy is better for motions involving the elbow joint (Elbow Flexion and Elbow Extension), and the Shoulder Flexion motion, compared to the other Shoulder motions (Shoulder Protraction and Shoulder Retraction. The Motion Selection time is instantaneous, reiterating the responsiveness, and accuracy of initial classification of the decoder. The learning rates do not indicate any trend of learning as the number of attempts progress. It just depends on whether the subject was able to recollect and reproduce the same activation patterns performed while training. The redundancy involved in performing the shoulder motions, especially Shoulder Protraction and Shoulder Retraction, is a major reason for the degradation of the performance in these cases. Overall the results indicate a good performance, comparable with other classification strategies.

In this paper we propose a methodology for real-time discrete control of the five different upper arm motions. The main purpose of this study is to characterize the reliability and the accuracy of ELM in classifying motion. This approach would result extremely beneficial for those application in assistive technology where the motion is decided with regards of the muscular activation of the subject. Currently, the classification is sequential, meaning that only one particular motion is done at a time. But in future, we would like to extend it to simultaneous classification of motions. Also, we would like to explore the role of muscle synergies in decoding shoulder motions, where coordinated muscle activation is involved.

References

[1] S. Bitzer and P. van der Smagt, "Learning EMG control of a robotic hand: towards active prostheses," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, may 2006, pp. 2819–2823.

[2] C. Castellini, A. E. Fiorilla, and G. Sandini, "Multi-subject / Daily-Life Activity EMG-based control of mechanical hands," *Journal of Neuroengineering and Rehabilitation*, vol. 6, no. 41, 2009.

[3] P. K. Artemiadis and K. J. Kyriakopoulos, "EMG-Based Position and Force Estimates in Coupled Human-Robot Systems: Towards EMG-Controlled Exoskeletons," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Springer Berlin Heidelberg, 2009, vol. 54, pp. 241–250.

[4] C. Zhu, S. Shimazu, M. Yoshioka, and T. Nishikawa, "Power assistance for human elbow motion support using minimal EMG signals with admittance control," in *Mechatronics and Automation (ICMA), 2011 International Conference on*, aug. 2011, pp. 276–281.

[5] T. S. Saponas, D. S. Tan, D. Morris, and R. Balakrishnan, "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 515–524.

[6] P. K. Artemiadis and K. J. Kyriakopoulos, "A Switching Regime Model for the EMG-Based Control of a Robot Arm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 53–63, 2011.

[7] M. Erkilinc and F. Sahin, "Camera control with EMG signals using Principal Component Analysis and support vector machines," in *Systems Conference (SysCon), 2011 IEEE International*, april 2011, pp. 417–421.

[8] J. Vogel, C. Castellini, and P. van der Smagt, "EMG-based teleoperation and manipulation with the DLR LWR-III," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 672–678.

[9] P. Parker, K. Englehart, and B. Hudgins, "Myoelectric signal processing for control of powered limb prostheses." *Journal of electromyography and kinesiology*, vol. 16, no. 6, pp. 541–548, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.jelekin.2006.08.006

[10] P. Z. Geng, Yanjuan and G. Li, "toward attenuating the impact of arm positions on electromyography pattern-recognition based motion classification in transradial amputees," *ournal of NeuroEngineering and Rehabilitation*, 2012.

[11] G. B. Jiang Ning, Muceli Silvia and F. Dario, "Effect of arm position on the prediction of kinematics from emg in amputees," *Medical and Biological Engineering and Computing*, vol. 51, no. 1-2, pp. 143–151, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11517-012-0979-4

[12] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2005.12.126

[13] E. Cambria and G.-B. Huang et al., "Extreme learning machines [trends controversies]," *Intelligent Systems, IEEE*, vol. 28, no. 6, pp. 30–59, Nov 2013.

[14] G. Huang, E. Cambria, K. Toh, B. Widrow, and Z. Xu, "New trends of learning in computational intelligence," *Computational Intelligence Magazine, IEEE*, vol. 10, no. 2, pp. 16–17, May 2015.

[15] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, April 2012.

[16] N. Hogan and T. Flash, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, 1985.

[17] L. Hargrove, Y. Losier, B. Lock, K. Englehart, and B. Hudgins, "A real-time pattern recognition based myoelectric control usability study implemented in a virtual environment," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Aug 2007, pp. 4842–4845.

[18] G. Li, A. Schultz, and T. Kuiken, "Quantifying pattern recognition - based myoelectric control of multifunctional transradial prostheses," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 2, pp. 185–192, April 2010.