Proceedings of the 2009 IEEE
International Conference on Mechatronics and Automation
August 9 - 12, Changchun, China

Adaptive Myoelectric Human-Machine Interface for Video Games

Mohammadreza Asghari Oskoei and Huosheng Hu

School of Computer Science and Electronic Engineering, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, Essex, United Kingdom

Email: masgha@essex.ac.uk; hhu@essex.ac.uk

***Abstract* – This paper proposes adaptive schemes for myoelectric based human-machine interface (HMI) applied to a video game. Adaptive schemes modify the classification criteria to keep a stable performance in long-term operations. Online support vector machine (SVM) is used as the core of classification to facilitate incremental training during run-time. Supervised and unsupervised methods are individually employed to update online training data set. The experimental results show that the proposed adaptive schemes increase the achieved scores and make a stable performance for myoelectric HMI.**

Index Terms – Myoelectric HMI, Adaptive Schemes, Video Game, Rehabilitation

I. INTRODUCTION

The myoelectric human-machine interfaces (HMI) could be employed as an alternative interface in powered wheelchairs and video games for the disabled people [1][2]. Moreover, the manifestation of fatigue in myoelectric signals is perceivable during long-term muscular activities [2]. However, reliability and robustness of a myoelectric HMI is still an open question particularly in a long-term operation. Myoelectric signal (MES) inherently has a complex stochastic structure and its characteristics are intensively dependent to subject's physical and physiological conditions, muscular activities, and data collection conditions. Existence of these intense dependencies has led us to employ machine learning approaches, such as support vector machine (SVM), in developing myoelectric HMI [3] since they are capable to adapt themselves with signal characteristics using real samples produced before and during the run-time.

Training a classifier (e.g. SVM) is an adaption process, in which its parameters are being adjusted using samples generated by a subject that is going to use HMIs and in conditions that are supposed to be for HMI's work. This is the reason that the training before application, known as offline training, is required. Off-line training discriminates MES patterns corresponding to muscular activities and keeps them in the form of parameters that construct the boundaries between classes. Its period should be adequately enough to collect comprehensive samples that represent different states of muscular activities and the influencing factors on them.

In spite of comprehensive offline training, existence of novel samples for myoelectric HMI during run-time is inevitable. This is due to stochastic process of MES generation, slowly growing phenomena having impact on MES such as fatigue, and external factors such as electrode displacement. Hence, myoelectric HMI needs schemes that maintain its performance robust and reliable.

A closed loop control system can provide a stable performance using feedbacks. As we know, visual feedback and stimulated sensory signals (toward the body) are two feedbacks that can be adopted in myoelectric HMIs. However, visual feedbacks that continuously involve the mind are not convenient for long-term applications. Meanwhile, the stimulated sensory signal is not always cost effective and practical. For example, when we grab an egg, we do not think about how hard we should grasp after we have gained such experience. Instead, our nervous system automatically takes care of grabbing an egg without breaking or dropping [4].

Adaptive control that involves modifying the control criteria to cope with parameter changes is another option to keep a stable performance for myoelectric HMI. Having a proper model of deviations in MES patterns is a key issue to stabilize the accuracy of manipulating commands. The model has to distinguish regular changes that represent various commands from unwanted changes (i.e. deviation) resulted in accuracy decline. It should differentiate transient states as well, which are highly unpredictable and even contradictory [5], from steady states that carry useful information.

Changes in MES patterns are either gradual or significant. The gradual changes can be resolved by adaptive schemes otherwise the system would need a re-configuration. Fatigue is a time-related factor that leads to gradual performance variations. It can be named as the dominant factor that affects steady states of MES in a long-term operation [6]. Online training is the core of adaptive schemes for pattern recognition based HMIs. It rebuilds the boundaries between classes using updated training data set (TDS) during run-time operations. There are two challenges for online training: (i) updating TDS to distinguishing deviations in MES patterns that lead to HMI performance decline, and (ii) online training algorithms that modify the classifier's parameters in run-time [7].

This paper investigates supervised and unsupervised methods to update TDS, employs online SVM to handle online training smoothly, and evaluates adaptive schemes by applying them to a video game. The rest of this paper is organised as follows. Section II introduces online SVM to manage incremental training in adaptive schemes. Section III describes the proposed supervised and unsupervised adaptive schemes for myoelectric HMI. The experiments conducted to evaluate the adaptive schemes are presented in Section IV, and finally, a conclusion and future work are given in Section V.

II. ONLINE SVM

Updating TDS inserts new samples to it and requires repeating training process during run-time. Using whole TDS for run-time training process is computationally expensive and can't satisfy the real-time constraint. Hence, we need an online algorithm that uses recently arrived (fresh) samples for the training process and meanwhile keeps the old trained patterns.

SVM is a kernel-based approach with a strong theoretical background, which has become popular tool for machine learning tasks involving classification and regression. It has been successfully applied to several applications, such as face identification, text categorization, bioinformatics and database mining [3]. Recently, Bordes et al. [8] have presented an iterative implementation of SVM that suits for online applications.

In standard definition, for a binary data set $(x_1, y_1),\ldots,(x_n, y_n)$, $x_i \in R^d$, and $y_i \in \{\pm 1\}$, separating hyperplanes between two classes in a feature space mapped by $\varphi(x)$ are defined as:

$$w.\varphi(x) + b = 0 \qquad w \in R^d,\ b \in R \tag{1}$$

A unique hyperplane that yields the maximum margin of separation between two classes and tolerates misplaced samples with distance ($\xi_i \geq 0$) is constructed by solving the following quadratic programming (QP) problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad , \quad \forall i \ \ y_i(w.x_i + b) \geq 1 - \xi_i \tag{2}$$

The constant $C \in [0,\infty]$ is an upper bound for samples that lie on the wrong side of the hyperplane, and it creates a trade-off between the capacity of the classifier and error in TDS. Given kernel $K(x_i,x_j)=\varphi(x_i).\varphi(x_j)$ and weights $w=\sum\alpha_i\varphi(x_i)$, a way to solve (2) is via its Lagrangian dual that has been simplified to find the multipliers αi:

$$\max W(\alpha) = \sum_i \alpha_i y_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j K(x_i, x_j) \tag{3}$$

$$\sum_i \alpha_i = 0 \ \ , A_i \leq \alpha_i \leq B_i \ \ , A_i = \min(0, Cy_i) \ \ , B_i = \max(0, Cy_i)$$

The equation (3) slightly deviates from the standard formulation because it makes the coefficients αi positive when $y_i = +1$ and negative when $y_i = -1$. Solving (3) helps to construct optimal hyperplane (1) and build decision function:

$$f(x) = \sum_{i=1}^{n}\alpha_i K(x, x_i) + b \tag{4}$$

The decision function (4) addresses the feature space using samples such that $\alpha_i \neq 0$, i.e. support vectors (SVs). SVM has been very successful and widely used because it reliably delivers state-of-the-art classifier with minimal tweaking.

Sequential minimal optimization (SMO) is one of the efficient numerical algorithms developed to solve (3). It works by making searches along the direction u starting from vector α that yields new vector $\alpha+\lambda*u$, where

$$\lambda^* = \arg\ \max W(\alpha + \lambda u) \ \ , \ \ 0 \leq \lambda \leq \phi(\alpha, u) \tag{5}$$

It was observed that the direction search is much faster when its coefficients are mostly zero, hence, SMO uses search directions whose coefficients are all zero except for single $y_i=+1$ and single $y_j=-1$. Practical implementations of SMO, such as LIBSVM [9], rely on a small positive tolerance $\tau>0$, to select a suitable pair (i,j), called 'τ-violating pair', that $\alpha_i<B_i$, $\alpha_j>A_j$, and $g_i-g_j>\tau$, where g_k is the gradient of $W(\alpha)$ and defined as

$$g_k = \frac{\partial W(\alpha)}{\partial \alpha_k} = y_k - \sum_i \alpha_i K(x_i, x_k) \tag{6}$$

Bordes et al [8] presents a novel online SVM called as LASVM that reorganizes SMO direction searches, as such, converges to the solution of (3). It is built on alternating two kinds of direction searches named PROCESS and REPROCESS. PROCESS involves at least one sample that is not already a SV, and potentially can change its multiplier (α_i) as such make it a SV. This enables LASVM to update SVs with merely using new TDS. REPROCESS involves two samples that already are SVs, and potentially can zero their multipliers to remove one or both of them from current SVs. LASVM, at first, initializes state variables, and then runs online iterations (e.g. PROCESS and REPROCESS) that sequentially visit all the randomly shuffled TDS samples (this may occur in epochs), and finally performs finishing step which is only useful when one limits the number of online iterations.

LASVM handles gracefully noisy data, converges to the known SVM methods (e.g. LIBSVM) solutions, and brings the computational benefits and the flexibility of online learning algorithms. Experimental evidence indicates that LASVM matches the SVM accuracy after a single sequential pass over TDS [8]. LASVM can be used in the online setup where one is given a continuous stream of fresh random samples. The online iterations process fresh training samples as they come and update existing SVs without referring to pervious samples. This is called incremental training, and it is a vital requirement to implement the online training for huge data sets, such as MES data.

III. ADAPTIVE SCHEMES

An adaptive scheme rebuilds the boundaries between classes during a real-time operation. It updates TDS from fresh data (i.e. have not been used for training yet) and applies them for online training (i.e. LASVM online iterations) periodically. SVM was initially trained (i.e. offline training) using pre-collected labelled data (supervised method). However, supplying online TDS whilst run-time is a challenge, and it can be conducted either supervised or unsupervised.

Although supervised methods, in which labelled data are used for training a classifier, can be a safe and protected option, in real-time applications, it often either imposes lots of expenses or entirely is impossible. For instance, providing true labels for continuous stream of MES data provided by a subject with amputation is nearly impossible. Unsupervised methods employ data samples without knowledge about their true labels. In these methods, the fresh data along with their predicted labels are adopted to update TDS.

To prevent performance decline in myoelectric HMI and provide a stable performance, adaptive schemes need to determine fresh samples that cause error in HMI outputs and add them to TDS during run-time. Outputs that do not match

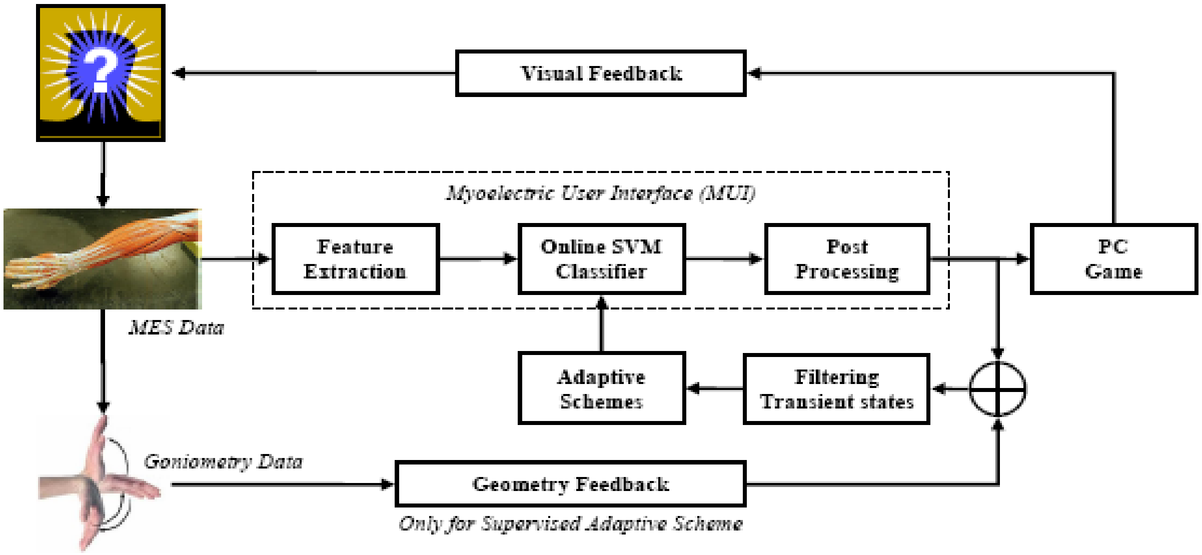

Figure 1 - A user, with MES electrodes on his forearm, drives a car in the computer based game using myoelectric HMI

the user's desired commands are counted as erroneous commands, and their frequent occurrences can represent the change in MES patterns.

To distinguish erroneous commands, we can rely on either outputs' features or external feedbacks. The former is categorised as an unsupervised method and the later is categorised as a supervised method. In the proposed unsupervised method, the correctness of the output commands were evaluated using their statistical features (i.e. continuousness and entropy). The samples that had entropy [9] less than a threshold and continuousness [11] more than a threshold were eligible to insert into the online TDS. This was based on a fact that a sample generating an output command with high certainty and continuity can be considered as a desired one.

The supervised method employs the geometry data to update TDS. It uses a goniometry sensor as a feedback to detect the deviation between HMI's estimated and real hand state. This method is not practical for amputees, but it can be used as a reference to evaluate other methods. In addition, preliminary experiments had revealed that the online training couldn't always improve the HMI performance, and it even sometimes made it deteriorated. Studying details showed that some of newly updated samples had made huge changes over the initial boundaries between classes. Moreover, the most of ambiguous samples that mislead the classifier were located in transient states. To this end, we adopted a filter that rejects transient states before updating online TDS.

The steady state, in this work, was defined in such way that the output command (i.e. hand state) has not been changed in the last three consequent segments (i.e. last 300ms) and the maximum recent changes in bending angles of the hand (in two directions) were less than five degrees. The first part of definition ensures equal levels of contractions of the involved muscles corresponding to hand states, and the second part (exclusive to the supervised method) provides fixed length of muscles. Excluding chaotic samples generated by dynamic conditions and physical movements leads us to study impact of fatigue in constrained contractions in course of time. This means that it helps us to study the impact of fatigue on HMI performance. Ultimately, the online training adapts the classifier based on the recent changes in steady state MES patterns presented by the online TDS samples. Figure 1 depicts a schematic diagram of the proposed adaptive myoelectric HMI applied to a video game.

IV. Experiments and Results

A. Experiments

The game and myoelectric HMI were both developed under Java Applet and run in two parallel threads. The game was graphically making a virtual driving environment, and enforcing the user to drive a car in a route with randomly appearing obstacles, having five ordinary manipulating commands: Go Forward, Backward, Right, Left, and Stop. The commands were being produced by a myoelectric HMI using MES corresponding to five hand motions: hand flexion, extension, abduction, adduction, and keeping straight. For subjects with amputation or deficient limb, the commands were being recognized through the muscular contractions depending to their ability and convenience.

Comparison of the achieved scores by a subject who had got matured training and exercises, during several sessions of playing game could be a measurable index to evaluate the HMI performance. The game scoring policy was oriented towards a safe and quick driving. Every forward step was gaining a positive score and every wrong step, in which car hit to an obstacle or a border, was gaining ten negative scores.

Four-channel MES collected from forearm muscles was applied to HMI. Its time domain features, including MAV, WL, and ZC, and extracted from disjoint segments with length of 100 ms, was classified by a SVM-based classifier with RBF kernel that its parameters had been adjusted by a grid search method applied on training data set [3]. The output stream of

Figure 2 - A user, with MES electrodes on his forearm, drives a car in the computer based game using myoelectric HMI

the classifier was post-processed by majority voting (MV) to reduce the error of the transient states and provide a smooth stream of commands. MV was designed to output the most occurred commands in the last three segments, otherwise STOP. Meanwhile, the post-processing module had an accelerating option that made acceleration by doubling the output command when the last three recent HMI outputs were identical. Figure 2 show a user playing game using a myoelectric HMI.

Meanwhile, we had added a goniometry sensor to data collection. The goniometry sensor (Biometrics Ltd SG65) was employed to label real-time stream of myoelectric signals. It shows the bending angle of hand in wrist junction in two directions: horizontal and vertical. Thus, states of hand (i.e. flexion, extension, abduction, adduction, and keeping straight) were automatically and simultaneously (sampling rate 1000Hz) inserted as a label to the recorded MES data. The selected threshold angles to distinguish different hand states, in both directions, were 20 degrees.

The experiment includes playing the game in three independent sessions for each subject. In the first and second sessions, myoelectric HMI with supervised and unsupervised adaptive schemes, respectively, were used to interact with the game, and in the third one, myoelectric HMI without any adaptive scheme was used to play the game. Enough time was considered between sessions to re-settle the muscular conditions into the same situation. Five healthy subjects (3 male and 2 female) with an average age of 24 years participated in the experiment, and repeated the experiment three times (with different order of sessions). The subjects were briefed and got the required skills by enough practicing before experiment. The sessions had to carry on gaining high scores as much as possible, and to make comparable figures, the time of sessions were taken almost equal in each experiment. The average time for sessions was about 23 minutes.

Technically, the game and myoelectric HMI were both developed in two Java threads: a thread to collect MES and goniometry data from the sensors, and a thread to process the collected data. Due to the segment length of MES for feature extraction, the former thread took about 100ms (i.e. segment length). The later that handled MES feed forward process (i.e. feature extraction, SVM classification, and post-processing), game process (i.e. refreshing screen) and online training process (i.e. updating TDS and retraining SVM) was fortunately took no longer than 100ms. Hence, the response time of HMI was about 100ms (i.e. 10Hz manipulating commands for the game).

The parameters were adjusted through the preliminary experiments. In the supervised scheme, samples that cause change in HMI outputs in the last three segments (i.e. 300 ms) and change more than five degree in hand bending angles were considered transient data and rejected to be a potential TDS sample. In unsupervised method, samples that were producing identical outputs in the five recent segments (i.e. 500 ms) with entropy less than 0.15 were selected as the online TDS samples. Online training was conducted at every 10 second. By the way, the first 40 seconds of each session was used for offline training. The labelled MES data produced in this period was applied to manipulate the car and train the SVM classifier.

For a skilled subject with mature training and exercise, achieved scores in each session can be a measure of performance of the applied schemes in real-time myoelectric HMI. Then, in each session, the MES features, the states of hand (via goniometry), the trace of output commands, and the scores were recorded for comparison. Furthermore, the subjects were asked to declare their feeling about the level of controllability of the car in each session. They had to select one of the three options: improvement, degradation or no-change for each adaptive scheme (i.e. supervised and unsupervised) comparing with non-adaptive scheme.

Error in MES classifications is inevitable, and its range for pre-collected data is between 3 and 10% [3]. Misclassified MES data that result in wrong commands to drive the car were often negligible and they were often compensated by immediate proper commands. However, some erroneous commands, named as the high-risk errors, made the car to hit to an obstacle or a border. The evaluation method should be sensitive to high-risk errors rather than the whole errors. Hence, the performance of the applied adaptive schemes was evaluated using three criteria: scored points, the number of high-risk errors, HMI output errors. Meanwhile, the user's satisfaction in controlling the car was recorded through the questionnaire after the experiment.

B. *Results*

Because of diversity in personal skills, the experimental results were compared individually for each subject. The performance of real-time HMI in manipulating a game without and with adaptive schemes were compared and the degradation or improvement resulted by the proposed adaptive schemes were figured out. The performance of the offline training process of sessions was assumed identical for each subject, since the distribution of random generator making

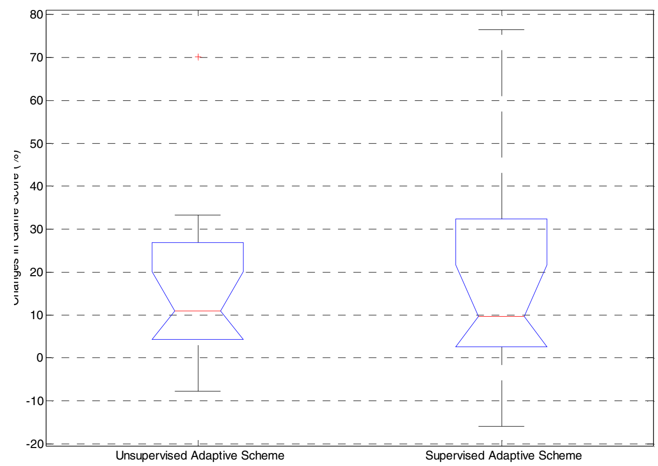

Figure 3 - Changes in the rate of scored points after using adaptive schemes myoelectric HMI

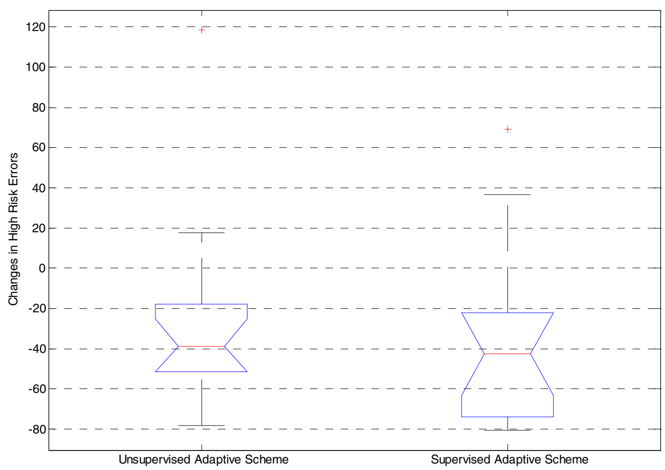

Figure 4 - Changes in the rate of high-risk errors after using adaptive schemes myoelectric HMI

obstacles was identical and the diversity of the activities was same in all sessions.

Statistical analyses were applied to interpret the experimental results. Due to relatively low rate of observations and their unknown distribution, the non-parametric approach (i.e. Kruskal-Wallis) was adopted in this work [12]. The critical p-value, which determines whether a result is judged "statistically significant", was chosen as 0.05. The results are illustrated in box-plots, in which the median and confidence intervals of the observations are shown.

Statistical analysis illustrated in Figure 3 depicts the rate of achieved scores rises by 10% in both unsupervised and supervised adaptive schemes. Figure 4 shows that the rate of high-risk errors in manipulating the car decreases about 40% in both applied adaptive schemes. These confirm that the both adaptive schemes improve myoelectric HMI performance. The diversity in the results is because of the human factors and differences among the performance of offline training of the sessions.

The experimental results suggest that in average, both the adaptive schemes improve the quality of control by increasing the maximum-scored points and decreasing the rate of high-risk and output errors. By the way, according to questionnaire, the subjects believed that the supervised adaptive scheme improves the controllability of the car during the game sessions. An adaptive scheme works as a complementary training process during real-time long-term operations, it can produce a stable performance for the myoelectric HMI if the updated data were chosen properly. The adaptation is bilateral, and beside the HMI, subjects also need to adapt themselves by adjusting the level of muscles contractions. Muscles adjustment is conducted in run-time using visual feedbacks, and subjects need enough time to adapt themselves. Moreover, they just are able to adapt with limited changes, and big changes require re-training. Hence, the online TDS samples should be selected properly and the changes resulted by online training must be smooth and gradual.

Transient state data are the consequence of changes in muscles state and/or limb motion, and samples produced by them are most likely to generate erroneous output commands. They start and finish rapidly and cause uneven adaptation if they were chosen as the TDS samples. A quick look on MES data recorded during the experiments and their corresponding HMI outputs suggests that about 65 percent of data are steady and the rest are transient. The error rate of steady data is about 10% while it is about 40% for transient data. This confirms that the steady data are more reliable than the transient data in classification. The proposed supervised adaptive scheme reduces the error rate by 60% and 20% in steady and transient data, respectively. This difference is because the adaptive schemes modify classification criteria based on changes in merely steady data, even though the figures reveal significant improvement in both transient and steady state data.

Figure 5 illustrates the rate of HMI output errors during sixty minutes playing game. As seen, the error rate in non-adaptive HMI rockets noticeably after 40 minutes non-stop playing game and this stops the increasing rate of the scores. This means that the pattern of MES data has remarkably changed in the last 20 minutes of the game. Basically, fatigue in sense of the both psychological and physiological concepts can be named as the main reason of this phenomenon, but it also could be considered as a consequence of it or an intensifying factor. Notwithstanding to its reasons, the experiments reveal that this phenomenon is not always clear and explicit.

Changes in MES patterns often develop slightly and hidden and cause lower efficiency, somehow, after a while make the subject to give up. Figure 5 depicts the impact of adaptive schemes on output errors and achieved scores. It compares the rate of output errors as well as the scores of two sessions: with and without adaptive schemes. It shows that the adaptive scheme makes a stable rate of error and scores in a long-term session.

Though the experimental results show that the both supervised and unsupervised adaptive schemes statistically improve the HMI performance, their approaches are quite

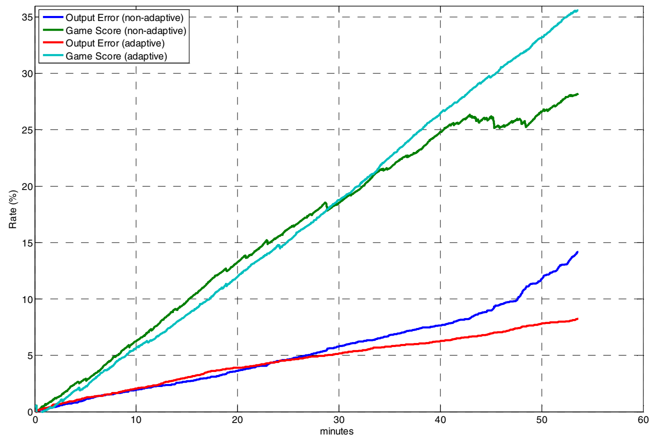

Figure 5 - Rate of output errors rockets noticeably after 40 minutes non-stop playing game and this stops the increasing rate of game scores

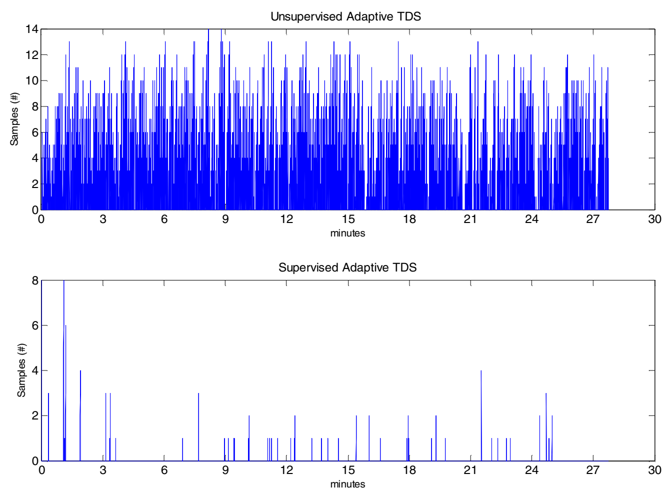

Figure 6 - Rate of new samples added to online TDS during 30 minutes in unsupervised (up) and supervised (down) adaptive HMI

different. Supervised method detects samples that their corresponding outputs didn't match with goniometry data, while the unsupervised method relies on HMI outputs with high certainty and continuity. Goniometry sensors, outputs entropy and continuity show the geometric displacement, classifier's certainty and user's satisfaction, respectively.

The unsupervised method, in spite of its difference with supervised method, is also depicting deviation in MES patterns. This can be studied by comparing online TDS samples updated through the two methods. A major difference between two methods is in their online TDS. The number of samples of online TDS in the unsupervised method is much higher than the supervised method. Figure 6 shows the number of new samples inserted to online TDS in the supervised and unsupervised methods. Large TDS as well as extra calculation to compute the output's entropy and continuity in the unsupervised method, impose high load of computation that necessitates online SVM. This can be named as the main setback of the unsupervised scheme.

V. Conclusion and Future Works

In this paper, myoelectric signals have been applied to the control of a video game based on user's muscular activities. Adaptive schemes, in real-time operation, modify the classification criteria to keep a stable performance in long-term operations. Unsupervised and supervised methods were implemented and examined in this work. Online SVM was employed to facilitate incremental training of the classifier.

The conducted experiments compared the performance of adaptive schemes with a non-adaptive HMI during playing a game. The experimental results show that the proposed adaptive schemes increase the achieved scores and make a stable performance for a myoelectric HMI. The unsupervised method imposed high computation load due to the calculation of entropy and high rate of updated samples in online TDS.

Reference

[1] M. Asghari Oskoei and H. Hu, "Myoelectric based Virtual Joystick Applied to Electric Powered Wheelchair", Accepted to IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS'08), France, September 2008

[2] M. Asghari Oskoei and H. Hu, "Manifestation of Fatigue in Myoelectric Signals of Dynamic Contractions Produced During Playing PC Games", Proc. of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EBMC'08), Canada, August 2008

[3] M. Asghari Oskoei and H. Hu, "Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb," IEEE Transactions on Biomedical Engineering, vol. 55, no. 8, pp. 1956-1965, Aug. 2008

[4] D. MacIsaac and K.B. Englehart, "The Science in Science Fiction Artificial Men", CROSSTALK The Journal of Defence Software Engineering, Vol. 19, No. 10, pp 4-8, October 2006

[5] K. Englehart, and B. Hudgins, "A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control", IEEE Transactions on Biomedical Engineering, Vol. 50, No. 7, July 2003, pp. 848-854

[6] M. J. Zwarts, G. Bleijenberg, and B.G.M. Engelen, "Clinical Neurophysiology of Fatigue", Clinical Neurophysiology (Elsevier), Vol. 119, No. 1, pp 2-10, January 2008

[7] M. Asghari Oskoei and H. Hu, Myoelectric Control Systems-A Survey, Journal of Biomedical Signal Processing and Control (Elsevier), Vol. 2, Issue 4, October 2007, pp. 275-294

[8] S. Bordes, J. Ertekin, J. Weston and L. Bottou: Fast Kernel Classifiers with Online and Active Learning, Journal of Machine Learning Research, 6:1579-1619, Sept. 2005.

[9] C-C. Chang and C-J Lin, LIBSVM: a library for support vector machines, at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

[10] Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm Motions", IEEE Transactions on Robotics And Automation, Vol. 19, No. 2, April 2003, Page 210-222

[11] R. Kato, H. Yokoi, and T. Arai, "Real-time Learning Method for Adaptable Motion-Discrimination using Surface EMG Signal", Proceedings of IEEE/RSJ International Conference on intelligent Robots and Systems, China, 2006, Page 2127-2132

[12] M. H. DeGroot and M. J. Schervish, "Probability and Statistics", Addison Wesley, 2002, pp. 589-694