Armband Gesture Recognition on Electromyography Signal for Virtual Control

Tanasanee Phienthrakul
Department of Computer Engineering,
Faculty of Engineering, Mahidol University
NakornPahom, Thailand
E-mail: tanasanee.phi@mahidol.ac.th

Abstract— **Many new devices come out with the idea of making more comfortable life. Myo armband is a wireless device for interacting with computer using electromyography (EMG) sensor. To communicate with the computer, the poses of hand and arm are matched with the command to control like a mouse click. Although the standard Myo can be used to communicate with computer, some poses cannot be detected or their results may be wrong. In this paper, the machine learning techniques will be applied to detect the hand gestures or poses. Double-tap, fist, spread finger, wave-in, and wave-out are 5 basic poses. These basic poses and rest will be trained and tested. The experimental results show that RBF network yields the acceptable results when it is compared to the results of many techniques.**

Keywords-Myo armband; electromyography sensor; gesture recognition; radian basis function network; learning algorithm; virtual control

I. INTRODUCTION

Technologies of wireless are applied to every device we daily use, for example, remote control with infrared technology, headphones or speakers with Bluetooth technology, keycard with radio frequency identification detection (RFID) technology, mouse and keyboard with wireless technology. These devices can be wired for stable data communication but people invent the wireless technologies for the easiest life style.

This paper talks about wireless device called Myo Armband [1]; which wear in the forearm of user to interact with computer or any device that support Bluetooth by making a gesture as sending a command to computer [2][3][4]. This device has own processor called ARM Cortex M4 Processor to process the data from electromyography (EMG) sensors. There are 8 sensors around Myo to read the electric signal in the muscle. A figure of Myo armband shows in Figure 1.

Not only EMG sensors, Myo also has a gyroscope, accelerometer, and magnetometer to detect 9-axis of movements. Moreover, it can detect 5 basic poses including double-tap, fist, spread finger, wave-in, and wave-out as showed in the Fig. 2.

Figure 1. Myo armband

Figure 2. Double-tap, fist, spread finger, wave-in, and wave-out (From left to right)

Myo armband was used to control a robot interface in a research of Gabriel Doretto Morais, and others [5]. They discussed the application of myoelectric signals to control electronic devices. They found that Myo is possible to control the movement of a robot and interact with the environment.

Then, Myo was applied on virtual control [6]. Virtual control of a robotic arm was built in Unity 3D [6]. A robotic arm based on EMG was controlled by using the tension and relaxation of muscles [6]. A virtual robotic arm based on EMG was preferred for a hand amputee to a virtual robotic arm based on a gyroscope and an accelerometer [6].

Moreover, Myo was used for musical interation [7]. MuMYO was a prototype instrument. Myo armband was an interface for musical expression [7]. The quality of the motion and muscle sensing data was sufficient for sound production and modification [7]. The weakest part was the limited number of built-in classification actions of standard Myo, and the occasional misclassifications that occur [7].

However, the gesture recognition is not really correct. The efficiency of gesture recognition may be improved using machine learning. Thus, in this paper, 8 learning algorithms are tested to compare their results. The suitable learning algorithm may be used to improve the accuracy of gesture recognition with Myo armband. Moreover, these algorithms may detect new gesture.

978-1-5386-4015-9/18/$31.00 @2018 IEEE

II. Machine Learning Techniques

As we mention in the previous section, in this paper, the results of 8 learning algorithms will be compare on gesture recognition with Myo armband. These algorithms have the different strong points and week points. Some algorithms may be used the same concepts, but difference in details. In this part, these algorithms will be briefly reviewed.

A. Naïve Bayes

Naïve Bayes [8] is an algorithm which adapted from Bayes's Theorem. The conditional probabilities are applied to classification problems that may have many attributes. Naïve Bayes can predict the class of testing data based on the statistic of learning data. In generally, this algorithm is widely used in many applications especially in text mining and natural language processing.

B. Neural Network and RBF Network

Neural Network [9] is a learning technique that imaged from the neuron system. In the computation of neural network, the nodes are divided into three kinds of layers, i.e., input layer, hidden layers, and output layer. All nodes in a layer are linked to the nodes in the next layer with the weight. Within a node, the inputs will be combined and the activation function will produce the output. This output will be used as an input of the nodes in the next layer or it can be the output of neural network in the output layer. Generally, linear, binary, or bipolar functions are used as the activation function of neural network. RBF Network is the neural network that uses radial basis functions (RGB) as the activation functions.

C. K-Nearest Neighbor Algorithm

This algorithm is using the distance between sets of training data and the testing data to find out which class the testing data is in by comparing the distance value of each sets of data including testing data then following the K value, get K training data sets that their distance values are nearest the testing data set, count up to get which class of training data sets is the highest [10] as the answer of the class that testing data set is. So, this algorithm is fast if the sets of data are small.

D. Decision Tree Algorithm

Decision Tree is an algorithm that splits the sets of data into groups depends on condition we give them. It including a root which is a starter node, branches which are the condition for split the sets of data into groups then a branch has only one group connects to which is called a node [11], a node can be split more with branches into groups indefinitely until they cannot be split more the last node on each branches are called leaf node which are the answer of decision tree. So, decision tree can be small or large depends on the sets of data can be split more or less. There are many variations of decision trees, such as J48, NB Tree, and Random Forest. These variations try to improve the efficiency of basic decision tree technique.

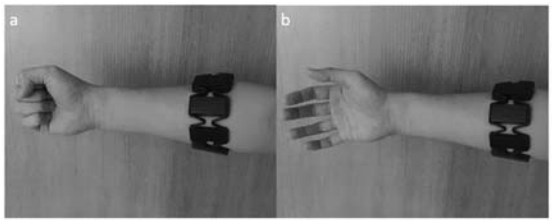
III. Myo Virtual Control

To simulation in using Myo armband as a mouse to control the cursor in computer, an application is created to receive data of 8 EMG sensors from Myo armband and decide the pose results by using decision tree. To do this, we got to know the pattern of fist pose and rest pose first by recording the datasets of 8 EMG sensors when doing a fist pose and a rest pose as we summarized in Table I. For this, 562 samples are used for making a decision tree.

TABLE I. Number of Samples in each Class of EMG Data

Pose	Number of Samples
Fist	280
Rest	282

Fist is a pose that we will use as left-click command and Rest means idle state for no click as shown in Figure 3. All of these data are recorded during fist and rest poses are done to use as training datasets in order to make a decision tree. Figure 4 demonstrates the method for deciding a pose from decision tree.



(a) (b)

Figure 3. Hand poses: (a) Fist pose, (b) Rest pose

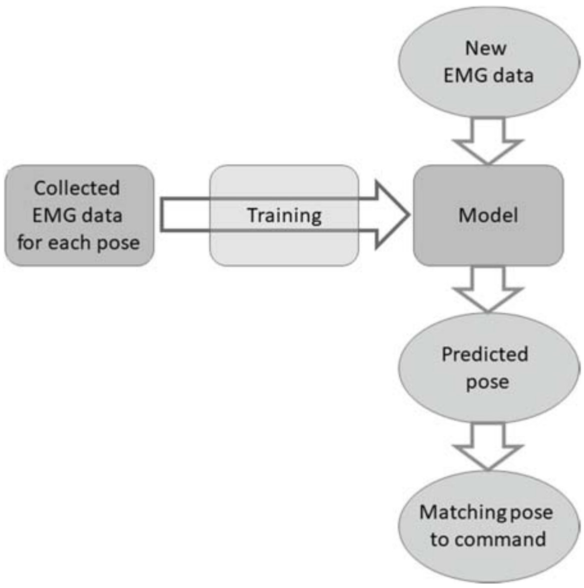


Figure 4. Method for deciding pose

To prove the result of human-computer interaction, Fitts' law [12] is used to test the proposed system. Throughput (TP) is calculated by dividing effective index of difficulty (ID_e) by the mean movement time (MT). ID_e is a Shannon formulation of the index of difficulty with the effective target width (W) and the effective distance (D) as showed in (1).

$$ID_e = \log_2\left(\frac{D}{W}+1\right), \quad (1)$$

where the units of ID_e are bits. Distance (D) and width of the target object (W) are shown in Figure 5.

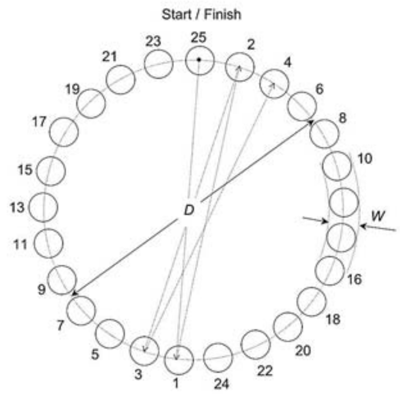


Figure 5. Multidirectional tapping task [9]

Movement time (MT) refers to the time that spend in moving from object to object. The unit is milli-seconds. Thoughput (TP) is an index of performance which are used for comparison. Equation (2) shows the calculation of the throughput.

$$TP = \frac{1}{y}\sum_{i=1}^{y}\left(\frac{1}{x}\sum_{j=1}^{x}\frac{IDe_{ij}}{MT_{ij}}\right), \quad (2)$$

where y is the number of test and x is the number of movement conditions. The units of throughput are bits per second.

The pose results are matched to commands and the click events are sent to the system. Figure 6 shows the sample of application for testing with sequence of red button appearance from 1 to 9. In this Fitts' law test application, the red button will randomly be appeared for clicking. After clicking, the red button will be disappeared and another button will be appeared in the opposite side from previous one. The next red button will be appeared in clockwise direction and this process are repeated.

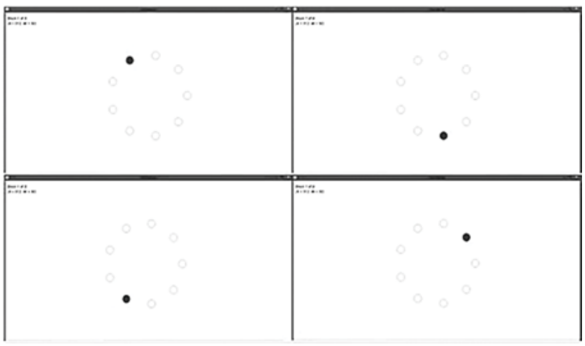


Figure 6. Multidirectional tapping task [9]

The distances between buttons for testing are 128, 256, and 512 bits. The size of buttons for each distance are 30, 40, and 50 bits, respectively. Each setting is tested 5 times for standard Myo and trained Myo. The average reaction times of cursor moving to the target button until left clicking are recorded. The experimental results are showed in Table II.

TABLE II. FITTS' LAW RESULT

Pattern		Standard Myo		Trained Myo	
D (pixels)	**W (pixels)**	**MT (ms)**	**TP (bits/s)**	**MT (ms)**	**TP (bits/s)**
128	30	1848.30	1.43	**1217.60**	2.68
256	40	2011.30	1.63	**1650.90**	2.02
512	50	1864.30	2.09	**1346.00**	1.91
Average		1907.97	1.72	**1404.83**	2.20

From Table II, the experiment results show that the trained Myo can be used to control the cursor. The average movement times are reduced about 26%. This makes the throughputs are also increased. However, these results are not better than mouse usage in term of precision. Thus, the other learning algorithms are explored.

IV. IMPROVING ARMNBAND GESTURE RECOGNITION

To improve the efficiency of gesture recognition with Myo armband, learning techniques are applied to the data from EMG sensors. Myo armband has 8 built in EMG sensors to detected electric signals in the muscles and these data are used to detect the gestures. EMG data were collected from six poses, i.e., double-tap, fist, spread finger, wave-in, wave-out, and rest. Rest will be occurred when the poses are changing; it is an idle state that Myo armband detected that we did not do any gestures. The numbers of sample data for each pose are showed in Table III.

TABLE III. NUMBER OF SAMPLES IN EACH CLASS

Poses	Number of Samples
Double-Tap	21
Fist	79
Rest	60
Spread	63
Wave-In	73
Wave-Out	56
Total	352

The total number of sample is 352 records. Each record has 8 features and 1 class that is the pose that we have done. The characteristic of each feature or each sensor on these poses are showed in Table IV.

TABLE IV. CHARACTERISTICS OF EACH EMG DATA

EMG	Min	Max	Mean	StdDev
Sensor01	-54	31	-0.432	7.395
Sensor02	-51	55	-1.005	10.902
Sensor03	-60	103	-0.447	10.605
Sensor04	-87	92	-0.011	17.427
Sensor05	-90	102	1.227	25.409
Sensor06	-63	63	0.797	14.867
Sensor07	-80	57	-0.646	13.242
Sensor08	-40	50	-0.987	8.291

Ranges of EMG data for each sensor are differed. Some sensors have high standard deviation. This depends on the position of the sensor on armband. These data will be tested by 10-fold cross-validation to compare the results on 8 learning algorithms, which are Naïve Bayes, Neural Network, RBF Network, K-Nearest Neighbor, J48, NB Tree, Decision Table, and Random Forest. The results will be report in term of average accuracy, average precision, average recall, and average of F-measure.

V. EXPERIMENTAL RESULTS

In the experiment, EMG data are tested on 8 learning algorithms. The experimental results are showed in Table V.

TABLE V. CLASSIFICATION RESULT OF EMG DATA

Classification Algorithm	Measurement			
	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	61.0795	0.585	0.611	0.584
Neural Network	52.5568	0.475	0.562	0.493
RBF Network	**62.7841**	**0.613**	**0.628**	**0.620**
K-Nearest Neighbor	52.2727	0.512	0.523	0.515
J48	53.1250	0.532	0.531	0.530
NB Tree	51.8295	0.535	0.548	0.537
Decision Table	53.9773	0.522	0.540	0.517
Random Forest	59.3750	0.568	0.594	0.570

The results show that RBF network, Naïve Bayes, and Random Forest yield the good solutions with all measurements, respectively. Overall of average accuracy is around 55.87% which is quite low to be accepted. Then, we try to improve these efficiencies. Boosting and bagging are two popular ensemble methods that will be tested. Adaboost and bagging are applied and the results are illustrated in Table VI and Table VII, respectively.

TABLE VI. CLASSIFICATION RESULT OF EMG DATA WITH ADABOOST

Classification Algorithm	Measurement			
	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	61.0795	0.585	0.611	0.584
Neural Network	53.6932	0.492	0.537	0.509
RBF Network	**64.2045**	**0.635**	**0.642**	**0.637**
K-Nearest Neighbor	52.2727	0.512	0.523	0.515
J48	55.3977	0.539	0.554	0.545
NB Tree	53.9773	0.514	0.540	0.523
Decision Table	53.9773	0.522	0.540	0.517
Random Forest	55.3977	0.537	0.554	0.543

TABLE VII. CLASSIFICATION RESULT OF EMG DATA WITH 10 BAGS

Classification Algorithm	Measurement			
	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	61.9318	0.593	0.613	0.591
Neural Network	*21.8750*	*0.092*	*0.219*	*0.119*
RBF Network	**66.1932**	**0.640**	**0.662**	**0.647**
K-Nearest Neighbor	51.1364	0.496	0.511	0.500
J48	58.2386	0.557	0.582	0.564
NB Tree	56.8182	0.556	0.568	0.544
Decision Table	55.1136	0.494	0.551	0.508
Random Forest	59.6591	0.582	0.597	0.577

The results show the performance of Adaboost that can improve the efficiencies of RBF network as 64.20%. Adaboost is not affect to most of testing algorithm. However, in Table VII, bagging technique can improve the efficiencies of many learning algorithms. We use 10 bags in testing this technique. Neural network produces the un-expectable results; it cannot learn this EMG data when we use 10 bags of bagging. The results of prediction are same in every samples that look like the guessing.

The accuracies of RBF network on different settings are compared in Figure 7. This figure shows accuracies of RBF network on EMG data, normalized EMG data, standardize EMG data, wavelet transformation on EMG data, Adaboost, and bagging. These experiments show that Wavelet transformation should not be performed on EMG data. The bagging is a good choice to improve the efficiency of gesture recognition on EMG data. RBF network yield the best solution when it is combined to bagging method.

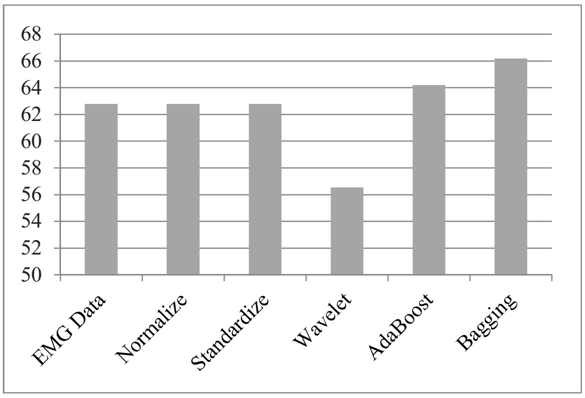


Figure 7. Graph of the accuracies on RBF network with the different additional techniques

VI. Conclusion and Future Work

Myo armband is an equipment that can detect arm gesture using electromyography Signal. A benefit of Myo is the convenience; users can move freely and the signal will be send to computer via Bluetooth. This paper proposed to improve Myo armband using machine learning technique. Myo can be used as mouse for controlling the cursor. Users can use Myo as a part of the virtual control. The first experimental results show that the trained Myo can improve the movement time. Then, to increase the precision of Myo armband, 8 learning algorithms are compared on gesture recognition problem.

EMG data from 8 sensors are used as the features of training. The experimental results showed that RBF network yielded the good results when it is compared to the other algorithms. Moreover, the accuracy of gesture recognition can be improved by the ensemble methods such as bagging and boosting. From the experiments, all measurements of RBF network can be improved by bagging method. Thus, RBF network and bagging techniques are another technique that may be applied to Myo armband. However, the other powerful classification methods may be applied in the future.

Myo armband is good enough to be used in not seriously works. Now, Myo is not smooth to control all objects. However, if the gesture recognition in Myo armband can be improved, the new application may be applied. The users may touch on small objects on screen. Also, the Myo armband can be used as a game controller. For the next testing, after we improve the gesture recognition, the Myo armband will be tested in game controlling or other applications.

Acknowledgment

I would like to thank to Mr.Yunyong Plerngpit. He is an expert on Myo device. Moreover, he dedicated himself for collecting all data.

References

[1] Myo, https://www.myo.com/

[2] Arthur, T.C., Faustina, H.: An analysis of mid-air gestures used across three platforms. In: British HCI ’15 Proceedings of the 2015 British HCI Conference, pp. 257–258. ACM, New York (2015)

[3] Tobias, M., Mithileysh, S.: Characteristics of Hand Gesture Navigation: a case study using a wearable device (MYO). In: British HCI ’15 Proceedings of the 2015 British HCI Conference, pp. 283–284. ACM, New York (2015)

[4] Guan-Chun, L., Heng-An, L., Yi-Hsiang, M., Chien-Jung, Y.: Intuitive Muscle-Gesture Based Robot Navigation Control Using Wearable Gesture Armband. In: 2015 International Conference on Machine Learning and Cybermetics (ICMLC), vol. 1, pp. 389–395. IEEE Conference Publications, (2015)

[5] Gabriel, D.M., Leonardo, C.N., Andrey, A.M., Maria, C.F.: Application of Myo Armband System to Control a Robot Interface. In: The 9th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2016), vol. 4, pp. 227-231, 2016.

[6] Ganiev, A., Shin, H.S., Lee, K.H.: Study on Virtual Control of a Robotic Arm via a Myo Armband for the Self-Manipulation of a Hand Amputee. Internaltional Journal of Applied Engineering Research, vol. 11, no. 2, pp. 755-782, 2016.

[7] Nymoen, K., Haugen, M.R., Jensenius, A.R.: MuMYO- Evaluating and Exploring the MYO Armband for Musical Interaction. NIME’15, May 31-June 3, 2015, Louisiana State Univ., Baton Rouge, LA.

[8] Mohammed, J.Z., Wagner, M.JR.: Data mining and analysis: fundamental concepts and algorithms. Cambridge university press, 467-473 (2014)

[9] Michael, A.N.: Neural networks and deep learning. Determination press (2015)

[10] Kai, Y., Liang, J., Xuegong, Z.: Kernel nearest-neighbor algorithm. Neural processing letters 15, pp. 147–156. Kluwer academic publishers, Netherlands (2002)

[11] Tom, M.M.: Machine learning. McGraw Hill, 52-53 (1997)

[12] Soukoreff, R.W., MacKenzie, I.S.: Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts’ law research in HCI. International Journal of Human-Computer Studies, vol. 61, pp. 751-789, December 2004.