Controlling hand robot using pattern recognition of finger movement

S K Risandriya[1], D S Pamungkas[1]

[1] Department of Mechatronics, Politeknik Negeri Batam, Jalan Ahmad Yani, Riau, Indonesia

E-mail: daniel@polibatam.ac.id

Abstract. The system enables to mimic the movement of the subject up to 87.5 percent accuracy. Electromyography (EMG) is one of the methods to control a robot's hand. This paper discusses the utilization of the signal EMG with a neural network algorithm to activate the fingers of the robot hand. The statistical analysis, which is the root mean square method, is used to train the patterns of the motion of the fingers of the user. Moreover, to sense the user's EMG signal, Myo armband is used. The results obtained reach 92.68 percent using 0.9 learning rate, 0.00001 error tolerance, one hidden layer with five nodes.

1. Introduction

Applications of robot hands are wide. One of the applications becomes a prosthesis hand. This hand is to help disabled people to do their daily activity. One of the questions of the prosthesis hand is to control this device. Some researchers utilize a user's brain signal/ electroencephalography (EEG) to control the artificial hand [1]. Other researchers exploit the user voice [2]. While the signal of the user's muscle or Electromyography (EMG) is used to control the hand [3]. However, the method using the EMG signal is the favoured method compared to others.

The EMG signal needs to be processed to be known by the system. There are several methods to process the EMG signal, namely, time domain, frequency domain, and time-frequency domain. Root mean square (RMS), mean absolute value (MAV), slope sign changes (SSC), waveform length (WL), Willison amplitude (WAMP), and zero crossings (ZC) are the methods which use time domain method [4, 5]. While Mean Power Frequency (MPF) and Fast Fourier Transforms (FFT) are the frequency domain method [6]. Wavelets and Wavelets Packet Transforms (WPT) is the method from the combination between time and frequency signal [7]. Moreover, the placement sensor can be divided into two, namely in and on the skin of the user.

To control the hand robot, there are some technics for example Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) [8], and artificial neural network (ANN) Algorithms [9]. The proposed method is using the sensor which is placed in the skin/surface EMG. To sense the movement of the sensor, Myo Armband is applied. Moreover, to process the signal time domain (RMS) is used. The RMS of the EMG signal is trained by an artificial neural network (ANN) algorithms. To deliver a complete explanation, this paper is organized as follows: Section 2 aims to provide the proposed system. To control the robot and using the ANN algorithm. Then proceed with next, Section 3 which presents the experiments on the proposed method by describing the potentials of the system in allowing to identify gestures of the hand. Followed by section 4, which provides the concluding remarks and the future work of this study.

2. Methodology

The overall system used in this system consists of Myo Armband, then Laptop or PC as data acquisition and programs that contain filters and neural networks. Then Arduino Uno becomes an interface between a PC and a robotic hand. The ANN algorithm technique requires two phases, namely the learning phase and application phase. The overall system can be seen in Figure 1 below.

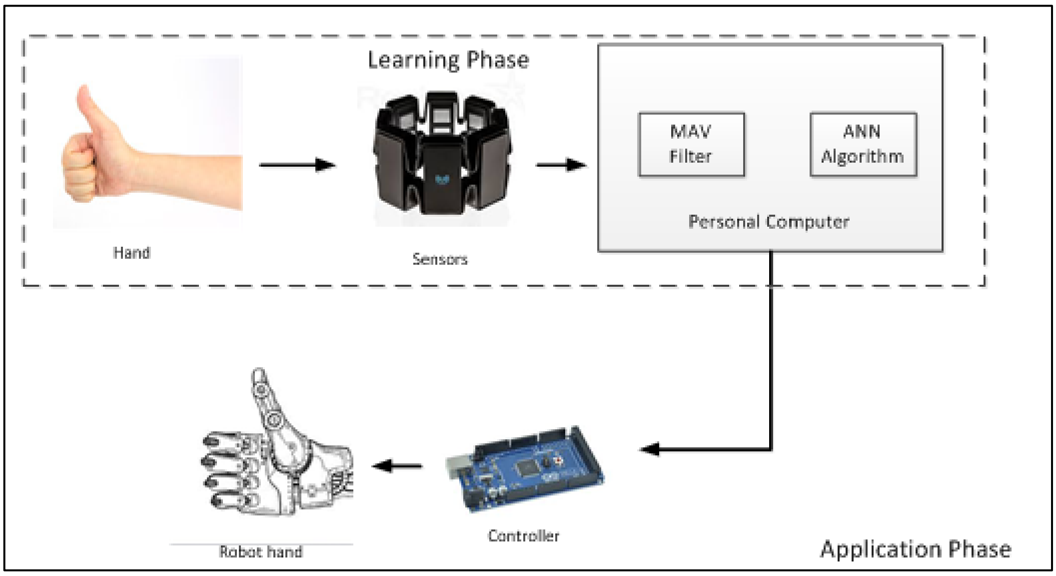

Figure 1. Diagram block of the system.

The sensor used is the Myo Armband sensor produced by Thalmic Lab in 2014. This device has 8 EEG sensors. Moreover, this armband has IMU sensors consisting of Accelerometers, Gyroscopes, and Magnetometers, each of which is divided into X, Y, and Z axes. Myo Armband uses a connection Bluetooth. The frequency of the armband is up to 200Hz frequency [10]. The subject wears the armband in the user's right arm, as shown in Figure 2.

Figure 2. A Myo Armband in the user's arm.

The signal from the armband will be transmitted to the PC. The computer will filter the signal and do the ANN algorithm. In the application phase, the gestures are recognized by the ANN algorithm. PC will pass the information to the robot via the controller. Arduino Uno is used as the controller. The custom robot is used for these experiments. This robot enables to move its fingers. The robot is shown in Figure 3.

Figure 3. A robot hand.

The gestures of the subject which are learned are the finger posture, namely thumb, index finger, middle finger, ring finger, and pinkie finger. These postures are for the right hand of the subject. These gestures are shown in Figure 4.

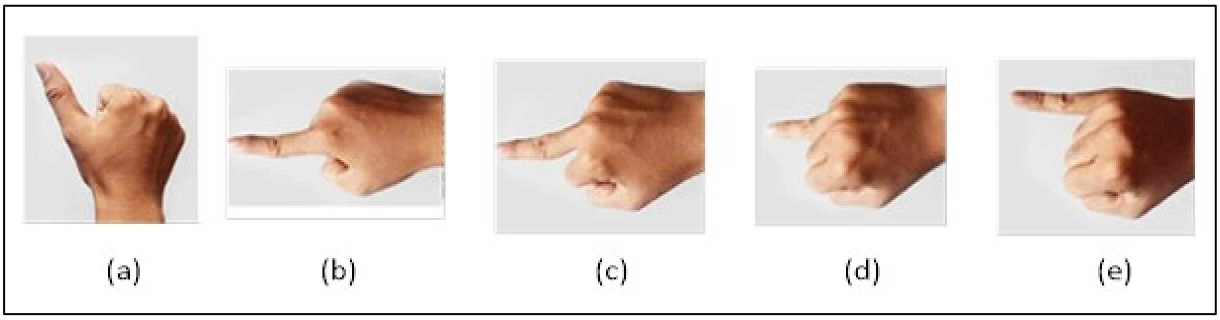

Figure 4. Postures of the fingers.

The signal the sensor must be learned before the ANN algorithm enables to recognize the subject's gestures. This learning process in ANN is known as Backpropagation ANN. The flowchart of the Backpropagation is shown in Figure 5(a). After the Backpropagation obtains the weights of the algorithm, then the forward ANN algorithm is applied. This phase is called the application phase. The flowchart of the application phase is shown in Figure 5(b).

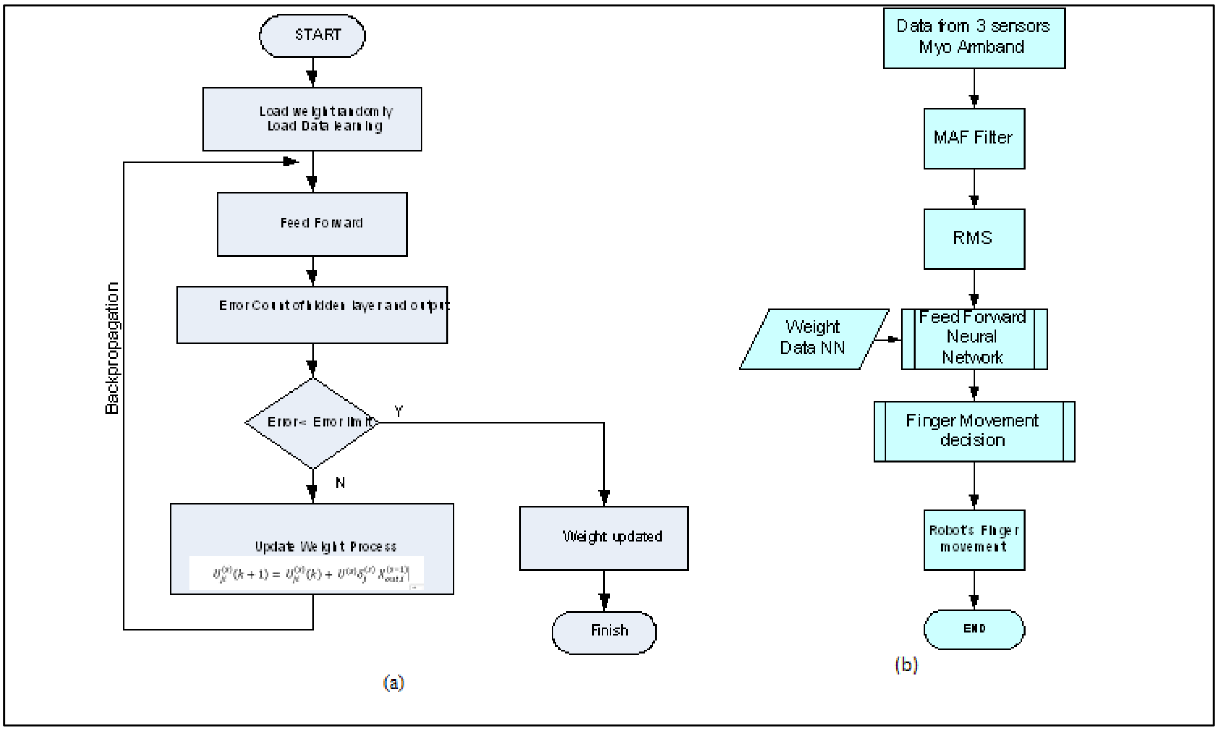

Figure 5. Flowchart of (a) Backpropagation (b) Forward NN.

The reading method uses the RMS method of the signal produced by Myo Armband, which was previously filtered using the Moving Average Filter (MAF) method. The signal produced is still an irregular EMG signal so that the RMS (Root Mean Square) value is taken. The results of the EMG signal from Myoarmband are as many as eight output signals that have been filtered, but from the eight output signals, not all signals are significant so that three significant input sensors are determined. The Equation (1) used for this filter is:

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[i+j] \tag{1}$$

Where: M is the number of windowing. The RMS outputs are shown Equation (2).

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} xi^2} \tag{2}$$

Data from 3 sensors are then filtered using the Moving Average Filter (MAF) filter, and the filter results are then taken the RMS (Root Mean Square) value, then studied using the Neural Network (NN), where 3 of these RMS inputs will be input for NN. The Artificial Neural network used is the standard Backpropagation type by using one hidden layer and five nodes. Figure 6 shows the architecture of the ANN.

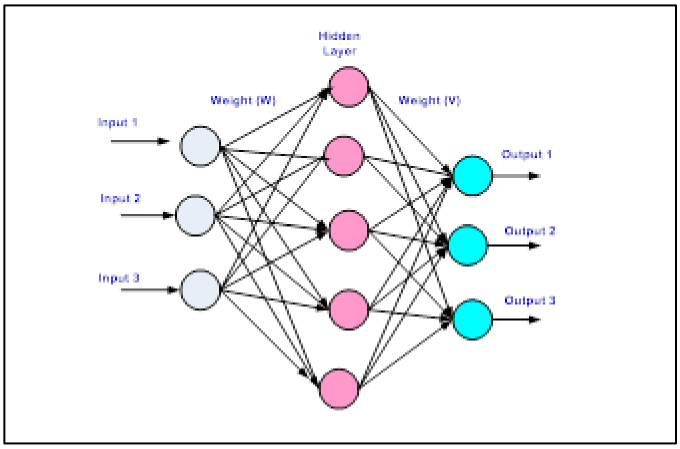

Figure 6. Structure ANN.

The activation function used is the sigmoid function, with an output between 0 and 1 (0.1). For the feedforward and backpropagation learning process, the order below can be considered. In the beginning, there is an initialization step. All weight values and threshold levels from random numbers are set. Those are distributed in a variety of small intervals. Afterward, the activation process is performed. Activates the backpropagation neural network with the input $\mathrm{X_1(P), X_2(P), \ldots, X_n(P)}$ and the desired output $\mathrm{X_{d.1}(P), X_{d.2}(P), \ldots, X_{d.n}(P)}$. Then, calculate the actual output of neurons in the hidden layer using Equation (1) and the equation to calculate the actual output from the output layer is Equation (2).

$$Y_i(P) = sigmoid\left[\sum_{i=1}^{n} X_i(P)xW_{ij}(P) - \theta_i\right] \quad (3)$$

$$Y_i(P) = sigmoid\left[\sum_{j=1}^{m} X_{jk}(P)xW_{jk}(P) - \theta_k\right] \quad (4)$$

After that, the weights of the algorithm are trained. Update the weight of the error value propagated backward corresponding to the output neuron. Calculate the error gradient from the neurons in the output layer using Equation (5) and (6).

$$\delta_k(P) = Y_k(P)\, x\, [1 - Y_k(P)]\, x\, e_k(P) \quad (5)$$

$$e_k(P) = Y_{d.k}(P) - Y_k(P) \quad (6)$$

Update the weights of output neurons:

$$U_{ji}^{(s)}(k+1) = U_{ji}^{(s)}(k) + U^{(s)}\delta_j^{(s)}\, X_{out,i}^{(s-1)} \quad (7)$$

Calculate the error gradient from neurons in the hidden layer:

$$\delta_j(P) = Y_j(P)\, x\, \left[1 - Y_j(P)\right] x\, \sum_{k=1}^{l} \delta_k(P)\, x\, w_{jk}(P) \quad (8)$$

Update the weights of hidden neurons:

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + U^{(s)}\delta_j^{(s)}\, X_{out,i}^{(s-1)} \quad (9)$$

In its application, the value of the node in the hidden can be modified (customized) to obtain the optimum iteration (epoch) value. Moreover, the learning rate value that can be changed to determine the accuracy of the NN. By varying the value of learning rates and errors as well as the value of nodes from the hidden layer, the NN optimization will be obtained.

3. Results and discussion

To demonstrate the potency of the proposed system a 25 years old male is held the trials. The sensors are placed on his arm. For the learning process, the subject must move each finger. As explained in the previous section, three sensors are used to read EMG signals. The filter results for five fingers can be seen as shown in Figure 7. Those data are taken with 200Hz sampling frequency.

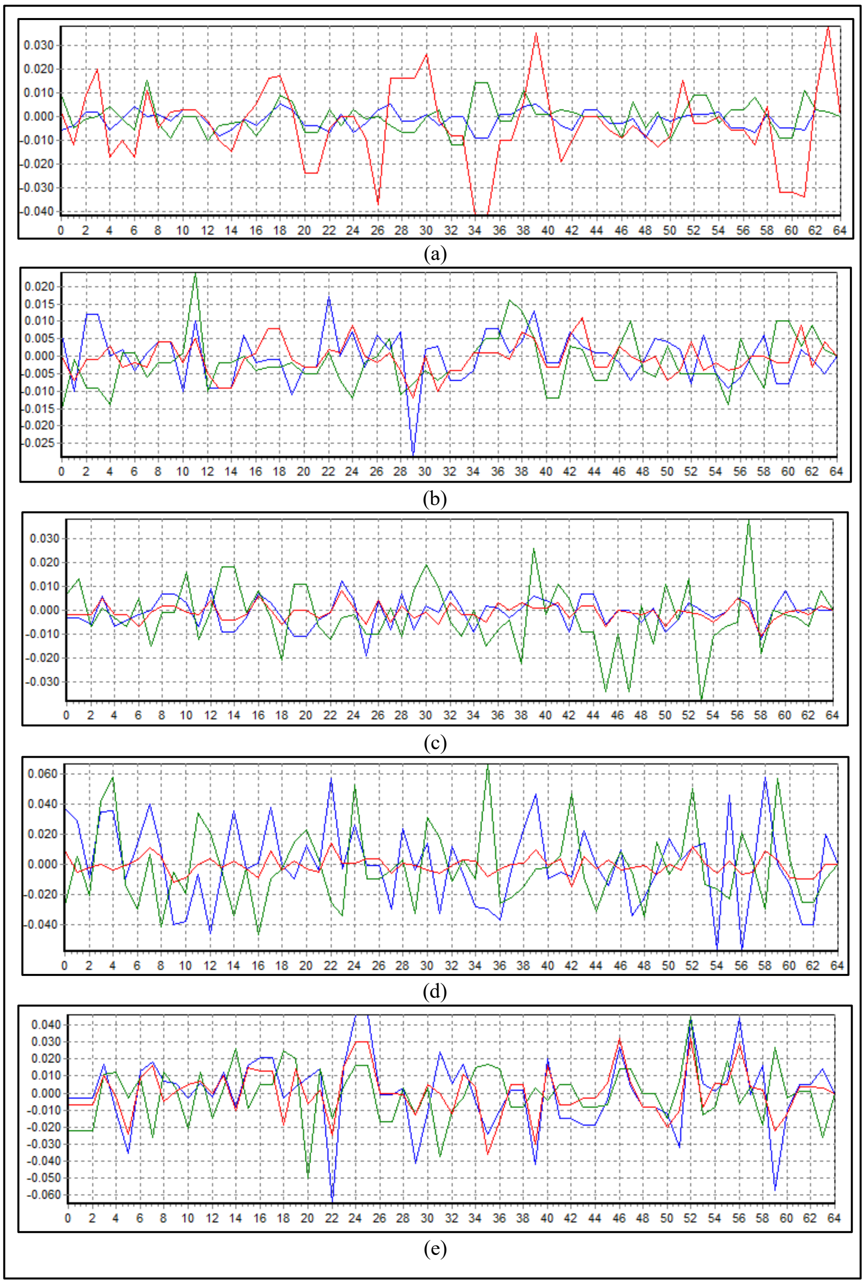

Figure 7. Signal of (a) thumb (b) index (c) middle (d) ring (e) pinkie finger.

The subject must move twenty times each finger to obtain the RMS of the signal. The RMS of these signals is learned using backpropagation neural network algorithm. For the learning process, the backpropagation NN is used one hidden layer with three nodes input, five nodes output, and three nodes output. The tolerance error is set 0.00001. Figure 8 shows the graph of the epoch the signal for learning rate 0.9.

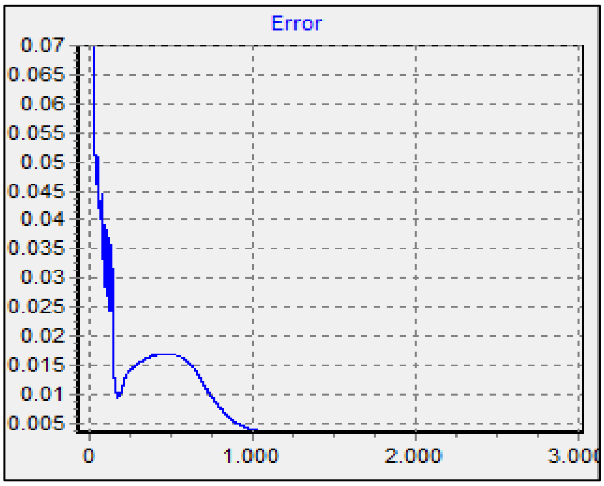

Figure 8. Epoch of the signal for learning rate 0.9.

Using the weights which obtain from back propagation NN, the forward NN are performed. To verify, the trials are done thirty times, with varying the learning rate. Table 1 shows the accuracy of various learning rates.

Table 1. The epoch and the accuracy of the system.

Learn rate	Epoch	% Accuracy
0.9	90679	92.68%
0.8	2196	92.56%
0.7	1112	89.76%
0.6	1092	88.87%

4. Conclusions
This paper provides the system which adequate to imitate the user's finger motion. The system consists of Myo armband, a PC for MAV filtering and NN algorithm also a robot hand. This proposed system shows that this method has the potential to recognize the user's motion up to 92.68% accuracy.

5. References
[1] Amlie A K *et al.* 2017 *Procedia Computer Science* **105**
[2] Asyali M H, Yilmaz M, Tokmakci M, Sedef K, Aksebzeci B H and Mittal R 2011 *Journal of Electrical Engineering and Computer Sciences* **19** 33-46
[3] Shunzhan H, Chenguang Y, Wang M, Cheng L and Zedong H 2017 *Chinese Automation Congress (CAC)* **1**
[4] Phinyomark A, Quaine F, Charbonnier S, Serviere C, Tarpin B F and Yann L 2014 *Computer Methods and Programs in Biomedicine* **117** 2
[5] Han-Pang H, Yi-Hung L and Chun-Shin W 2003 *IEEE International Conference on Robotics and*

Automation **11**
[6] Sadikoglu F, Kavalcioglu C and Dagman B 2017 *Procedia Computer Science* **120** 422-429
[7] Xin G, Peng Y, Ying L and Wei-Li Y 2004 *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 341-344
[8] Zhang D, Xiong A, Zhao X and Han J 2012 *IEEE International Conference on Information and Automation ICIA* **1**
[9] Kurniawan S R and Pamungkas D *2018 International Conference on Applied Engineering (ICAE)* 1-6
[10] Rawat S, Vats S and Kumar P 2016 *International Conference System Modelling & Advancement in Research Trends (SMART)* 115-120

Acknowledgments
This research is supported by Politeknik Negeri Batam.