Electromyography Gesture Identification Using CNN-RNN Neural Network for Controlling Quadcopters

Ray Antonius, Hendra Tjahyadi
Faculty of Computer Science, Universitas Pelita Harapan, Tangerang, Indonesia
E-mail: hendra.tjahyadi@uph.edu

Abstract. 

Purpose: This paper presents a CNN-RNN neural network approach towards electromyography. Here the neural network is employed to identify different gestures from the reading of signals from human’s hand muscles. The identified gestures are then utilized to control a drone. 

Methodology: Implementation is made by using Myo, an 8-channel Electromyography (EMG) data acquisition device. 14000 datasets of 9 different gestures are collected and used to train the CNN-RNN models. The trained models are then tested on a drone using dronekit to translate the gestures to drone commands and send them from Python to the drone. 

Results: The results show that a CNN-RNN approach is very effective in identifying gestures from raw muscle data, resulting in an average of 96.60% positive identification for each gesture. The identified gesture shows an effective drone control from the simulated drone. 

Applications/Originality/Value: The high rate of positive identification opens the possibility to use a wearable device that is able to give the current state of the hand’s muscle tensions, translating them into a specific command to control devices or machines. With a rich information and real-time human control, it is expected that humans will experience a more intuitive approach towards devices or machines control.

1. Introduction

Just like televisions and smartphones, drone technology will soon be the next craze of humanity as it is projected to grow into 2.4 million users in 2022. People have seen lots of things that are being integrated with the automation of that drone. Drones will soon be irreplaceable to human daily lives as if they are an extension of their own.

Current approach on controlling drones has been rather unnatural. People currently need to use a radio remote controller to command drones and many preparations to setup before any launchings. Looking at the trend of drone usage, drones are likely to be used by the majority of people in the future, therefore a more “human” like intuitive control is required.

Human muscles are very rich in information, every movement gives a different set of muscle tension characteristics. These different set of muscle tension combinations can be interpreted as data to see what movement/gesture the muscles is trying to achieve. One method for acquiring the muscle information is called Electromyography (EMG). The idea is to use a wearable device that is able to give the current state of the hand’s muscle tensions, translating them into a specific command to control drones. With a rich information and real-time human control, it is hoped that humans will experience a more intuitive approach towards drone control.
One problem with that idea is, however, the difficulty of interpreting muscle information. The ever-growing field of deep learning gives us a new opportunity to tackle this problem. Neural networks are able to translate a set of states into an accurate output, given enough training data.

2. Underlying Theory

2.1. Anatomy of Human Hands
The upper limb consists of three regions: the arm, the forearm, and the hand [10]. The muscles that allow the human hand to move can be classified into two separate groups, intrinsic and extrinsic. Intrinsic muscles are muscles located inside the hand, such as the muscles used for controlling the thumb and little finger. Extrinsic muscles are muscles that are located in the lower arm, and they connect to the rest of the fingers through tendons [12].

Muscles tissues contains cells that are excitable, meaning that they respond to a stimulus. Muscles are contractile, meaning they can shorten and generate a pulling force. Mahmoud [2] explains the procedures of hand locomotion. The brain starts the journey by sending a series of action potentials called neural activity signals to the spinal cord from the motor cortex of the brain. The signal contains information about the movement will be sent to the relevant muscle via motor neurons. This begins with upper motor neurons, that carry the signal to lower motor neurons.

The thumb, which is often ignored in most models of hand movement, is capable of abduction from the palm, rotation through to opposition, and flexion or extension of the metacarpophalangeal and interphalangeal joints [13].

2.2. Surface Electromyography
Surface Electromyography is a method in obtaining raw muscle signals by sampling the depolarization (a change in the electrochemical gradient) using an electrode. The signal obtained from the electrodes contains the information of the sampled muscle’s activity and contraction status. De Luca [13] states that acquiring EMG signals is a complicated process because of the anatomical and physiological characteristics of the muscles. EMG signal is stochastic in nature. The usable energy is between 0 to 500 Hz frequencies and is dominant in 50 to 150 Hz range [1].

2.3. Signal Processing
The received EMG signals are considerably weak (in orders of tens to thousands of µV). To effectively process the signal, operational amplifiers are used. Operational amplifiers (op amp) is a voltage controlled voltage source that takes two different voltage inputs and outputs an amplified the difference of the two voltage inputs.

After amplification has been done, the signal then gets processed to remove the unnecessary noise. Noise figure increases with each cascaded amplifier stage [2–4]. Looking at the nature of EMG, there will be noise based on the constant change of electrode position during their use [1], hence it is recommended to also estimate the noise as the moving average of the signal.

2.4. Myo
The myo armband is a wearable device designed by Thalmic Labs Inc that implements EMG in its data acquisition. Not only is Myo equipped with eight EMG electrodes, it also has a 9-axis inertial measurement unit (IMU) and a transmission module (Abduo and Galster, 2015). The physical appearance of the Myo armband is shown in Figure 1.

A Myo armband is equipped with several pre-built gestures for immediate use as seen in Figure 2. However, these pre-built gestures have really poor accuracies. Phienthrakul [3] have been tried many algorithms to tackle the problem of gesture recognition accuracy level with little
success as shown in Table 1. The table shows that these algorithms still give poor accuracy with the best accuracy is obtained using RBF Network algorithm with 62.78% of accuracy. All of the identification algorithms listed in the table are not designed to process a sequence of inputs, therefore their outputs are considerably low in the level of accuracy and precision.

Table 1: Current Identification Accuracies

| Identification Algorithm | 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  | 0.62      |
| K-Nearest Neighbor       | 52.2727  | 0.512     | 0.53    | 0.515     |
| J48                      | 53.125   | 0.532     | 0.531  | 0.53      |
| NB Tree                  | 51.8295  | 0.535     | 0.548  | 0.537     |
| Decision Table           | 53.9773  | 0.522     | 0.54   | 0.517     |
| Random Forest            | 59.375   | 0.568     | 0.594  | 0.57      |

2.5. Neural Network
Artificial Neural Networks (ANN) can be described as computational tools which are composed of neurons [4]. These neurons are interconnected to each other and pass information from one neuron to the other. This architecture was inspired by its biological counterpart, though it is still quite dissimilar but it contains the same concept [5]. Similar to a biological neuron, there
are a set of inputs (synapses) with different weights (dendrites) that get summed together. After the summing the weights, the result is passed through an activation function, thereby exciting the neuron (firing an action potential through the axon).

Figure 3: Multi-Layer Perceptron Architecture Illustration

Little notable research was done on Artificial Neural Networks until the backpropagation algorithm was developed in the mid-1980s, which permitted to train networks with more than one hidden layer. Hence, the Multi-Layer Perceptron (MLP) was developed. Figure 3. illustrates the way each perceptron are connected.

There have been many applications and attempts in improving and developing the Neural Network Model as seen in [4–12].

2.5.1. **Convolutional Neural Network**  Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other [13]. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics [4].

A CNN can be broken down into four regions based on its functionalities. The input layer, the convolutional layer, the pooling layer, and the fully connected layers. The input layer is the most common layer that can be seen across many Artificial Neural Networks. The convolutional layer will determine the output of neurons which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The pooling layer will perform a simple down-sampling along the spatial dimensionality of the given input. After the pooling layer, normal Artificial Neural Network procedures are done using the fully-connected layers [5].

A good example of CNN being used can be seen from the MNIST handwritten digits classification [6] where most of the CNN models give a relatively good result compared to the other classifiers.

2.6. **Recurrent Neural Network**  Recurrent Neural Networks (RNNs) are an extension of feed-forward networks especially designed to handle sequential data [7]. The fundamental feature of a RNN is that the network contains at least one feed-back connection, therefore, the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences, e.g., perform sequence recognition/reproduction or temporal association/prediction.
Recurrent neural network architectures can have many different forms. One common type consists of a standard Multi-Layer Perceptron (MLP) plus added loops. These can exploit the powerful non-linear mapping capabilities of the MLP, and also have some form of memory. The name Long Short-Term Memory (LSTM) arise from its ability to remember the preceding states not only for the short time (short-term), but also preceding states for far much longer time (long term) as the commonly used recurrent neural networks at that time [8]. Others have more uniform structures, potentially with every neuron connected to all the others, and may also have stochastic activation functions. Figure 4. illustrates a Recurrent Neural Network architecture.

LSTM is a novel recurrent network architecture in conjunction with an appropriate gradient-based learning algorithm. LSTM is designed to overcome the error backflow problems. It can learn to bridge time intervals in excess of 1000 steps even in case of noisy, incompressible input sequences, without loss of short-time lag capabilities. This is achieved by an efficient, gradient-based algorithm for an architecture enforcing constant (thus, neither exploding nor vanishing) error flow through internal states of special units [9].

Figure 4: Recurrent Neural Network Illustration

3. System Design
3.1. Hardware
From the hardware side, EMG data are acquired through Myo, send through the Neural Network which in turn will send drone commands through the mavproxy, giving the commands to the drone while also showing the drone’s current location using mission planner. This flow is illustrated in Figure 5.

Figure 5: Block Diagram Flow
For a more accurate and reliable source of data, a commercial surface electromyography sensor (Myo) is used in this research. Muscle signals (8 channels) from the forearm are extracted and processed into a more refined data stream, which will then be sent through Bluetooth Low Energy to the receiving computer [10].

A simple quadrotor will be assembled to display the effectiveness of the gesture control. The quadcopter will be designed using a ZMR250 quad frame, R2206 Brushless Motors, BLHeli 20A ESCs, Three-Blade Propellers, and a Pixhawk Microcontroller. A pixhawk microcontroller is chosen because of the massive development support and its open-source capabilities. In addition to the great precision and active development, there are many open-source projects supporting this microcontroller, one significant example would be dronekit-python and pymavlink.

3.2. Software

Figure 6 is a flowchart describing the process of how data are acquired, trained on a model, and tested on a drone. Model is developed and trained until its outputs satisfying accuracy level of around 90%. If it is lower than the target accuracy, model will be redesigned to further optimize its results.

Data will firstly be acquired from the Myo Armband. Figure 7, shows an example of the data acquired from Myo. The Myo armband streams at a fixed rate of 200 Hz and output signals from the eight electrodes with amplitudes ranging from -100 to 100. The amplitudes are acquired from the digitalized voltage difference between the respective channel’s reference and ground electrodes.

For the gesture classification, the training data are acquired from multiple people of various genders and age to avoid a biased classification towards a single person. A more varying data will translate into a more robust model that is able to make predictions more accurately in a consistent manner.

At least 8 gestures are required to fly a functional drone and 1 additional gesture for an idle state. In this paper 9 hand gestures are classified as command signals for drone control or command, namely: rest for rest command, fist for decrease altitude, hold right for tilt right, hold left for tilt left, fingers spread for increase altitude, metal for tilt forward, flower for tilt backwards, thumbs up for arms and take off, and peace for return to launch as shown in Table 2. The 9 gestures were chosen carefully knowing that the forearm muscles acting upon them will output a distinguishable pattern.

Figure 8. illustrates the data format in a graphical manner. The training data input will be a 3-dimensional data, consisting of a fixed 32-length array of 8-channel signal amplitudes sampled with a frequency of 200 Hz from the Myo Armband. Signal amplitudes that ranged from -100 to 100 will be normalized into a 0 to 1 floating point. A third of the data are split to be classified as validation data. This is done to avoid biased high output accuracy based on only the training data.

Eleven distinct models are tested against the same training data to find the difference between a single layered CNN (Model 1 – Model 3), double layered CNN (Model 4), a single layered LSTM (Model 5 – Model 7), and a CNN-LSTM combination neural network model (Model 8 – Model 11). Table 3 shows the list of models that are tested.

To ensure the accuracy of the models, each model is manually tested per gesture accuracy. A total size of 1000 sample data from each gesture are propagated through each of the model to measure the accuracy of each gesture. Accuracy of each gesture will be calculated by averaging the positive predictions from the respective gesture input data.

3.3. Implementation, testing and system evaluation

The implemented system is successful in collecting EMG signal data. Figure 9 shows one of the data samples acquired when doing a gesture. This acquisition data is done on 10 different
Figure 6: Software Flowchart

Figure 7: Acquired Data Stream from Myo

Figure 8: Data Format Illustration
Table 2: Gesture Command Planning

| Gesture Image | Gesture | Intended Command | Gesture Image | Gesture | Intended Command |
|---------------|---------|------------------|---------------|---------|------------------|
|               | Rest    | Rest             | Metal         | Tilt Forward |
|               | Fist    | Decrease Altitude| Flower        | Tilt Backwards |
| Hold Right    | Tilt Right |                  | Thumbs Up  | Arm and Take Off |
| Hold Left     | Tilt Left |                  | Peace        | Return to Launch |
| Fingers Spread | Increase Altitude |                  |              |                   |

Table 3: Planned Neural Network Models

| Model No. | CNN Layer 1 Size | CNN Layer 2 Size | LSTM Layer Size |
|-----------|------------------|------------------|-----------------|
| 1         | 32               | 0                | 0               |
| 2         | 64               | 0                | 0               |
| 3         | 128              | 0                | 0               |
| 4         | 32               | 32               | 0               |
| 5         | 0                | 0                | 16              |
| 6         | 0                | 0                | 32              |
| 7         | 0                | 0                | 64              |
| 8         | 32               | 32               | 16              |
| 9         | 64               | 32               | 16              |
| 10        | 64               | 64               | 32              |
| 11        | 128              | 64               | 32              |

people, with their gender and age shown in Table 4. A total of 14000 dataset comprised of the 9 gestures are collected and used for the training & testing of all the neural network models.

Figure 10 to Figure 20, visualizes the results from each model’s training. They show the accuracy and loss of the eleven different models. It can be seen from Figure 10 to Figure 12, that a single CNN layered model can be considered as a good model with no overfitting nor underfitting, while a single LSTM layered model has many overfitting problems as shown in Figure 14 to Figure 16. A double layered CNN model also shows a rather poor accuracy as shown in Figure 13. On the contrary, a combined CNN-LSTM layer proves to be a highly
Figure 9: Gesture Data Acquisition Results

Table 4: EMG Data Source

| No. | Age | Gender |
|-----|-----|--------|
| 1.  | 21  | Male   |
| 2.  | 20  | Male   |
| 3.  | 21  | Male   |
| 4.  | 20  | Female |
| 5.  | 21  | Male   |
| 6.  | 17  | Male   |
| 7.  | 19  | Female |
| 8.  | 20  | Male   |
| 9.  | 21  | Male   |
| 10. | 20  | Female |

accurate and reliable model with high accuracy as shown in Figure 17 to Figure 20. CNN layers are designed to process the correlation inside a data sequence at a given time, while LSTM layers are designed to process the correlation of a data with its past input data.

From the training results, seven models, Model 1 to Model 3 and Model 8 to Model 11 are chosen to identify gestures. Two sets of experiments are implemented and measured. In the first set of experiments, the seven models are compared in recognizing the true gestures from the nine different of gestures, and in the second set of experiments, the models are compared in the capability to reject the false gestures from the nine different of gestures.

The results of the first set of experiments are shown in Table 5 with notation $R$: Rest, $F$: Fist, $HL$: Hold Left, $HR$: Hold Right, $ξ$: Flower, $FS$: Finger Spread, $M$: Metal, $TU$: Thumbs Up, $ψ$: Peace. From the table it can be seen that Model 10, the combination of CNN-LSTM 64-64-32 Model gives the best results with the accuracy rate of 96.60%. The results of second experiments are shown in Table 6 with notation $R$: Rest, $F$: Fist, $HL$: Hold Left, $HR$: Hold Right, $ξ$: Flower, $FS$: Finger Spread, $M$: Metal, $TU$: Thumbs Up, $ψ$: Peace. From the table, the Model 10 consistently gives the best results in the capability to reject the false gestures. In average this model only gives 0.4% of falsely recognizing gestures.
The experiments also show that the hardest gesture to recognize correctly is the metal gestures with a success rate of 90.10% and the easiest gesture to recognize is the peace gestures with a rate of success of 97.82%. The metal gestures also consistently give the highest results for falsely recognize with the rate of 1.35%, while the gesture that gives the lowest results of falsely recognize is the hold right gestures with the rate of 0.21%. The metal gesture is hard to identify and also gives highest results of falsely recognize because this gesture is the hardest pose to maintain with a considerable amount of muscle tension. The signals acquired from the pose tends to drift to near zero.

Table 5: Gesture Accuracy Comparison per Model

| Model                  | True Positive Gesture Samples (%) | Average (%) |
|------------------------|-----------------------------------|-------------|
| 32 CNN Model (Model 1) | 92.40 92.80 93.20 96.90 82.90 85.80 73.70 91.00 97.10 | 89.53       |
| 64 CNN Model (Model 2) | 92.40 94.60 95.30 96.80 86.50 89.40 83.90 84.30 97.90 | 91.23       |
| 128 CNN Model (Model 3)| 94.70 94.30 96.30 98.30 90.40 91.90 88.10 89.10 96.50 | 93.29       |
| 32-32-16 Model (Model 8)| 99.50 97.00 95.40 97.70 93.90 93.10 94.20 95.50 96.70 | 95.89       |
| 64-32-16 Model (Model 9)| 99.60 96.00 91.30 92.50 93.30 97.60 93.40 98.30 | 95.24       |
| 64-64-32 Model (Model 10)| 99.70 95.40 97.60 97.80 95.10 96.40 98.50 98.00 | 96.60       |
| 128-64-32 Model (Model 11)| 99.50 95.10 98.90 97.30 92.20 92.90 94.70 95.00 | 99.30       |
| Average (%)            | 96.83 95.02 95.42 96.75 90.89 90.97 90.10 92.16 | 97.82       |

From the results of the two set experiments, Model 10 is then chosen to control drone. Figure 12 (a) and Figure 12 (b) show a drone being armed and controlled using the Model 10. With a few minor errors in predicting the gesture shown in Figure 22, where the metal gesture was sometimes classified as a thumbs up and flower. Looking at the overall results, this classifier can be used to fly a perfectly balanced drone.

4. Conclusion and Suggestions

From the implementation and experiments, it is concluded that muscle tension acquired by obtaining the raw EMG data can be identified using an identification algorithm and be used as drone commands based on the identification's predictions. There is only limited amount of simple gestures that can be made by identifying the muscle data only. A good gesture identifier
Figure 11: (a) Training Results from a 128 CNN Model (Model 3) (b) Training Results from a 32-32 CNN Model (Model 4) (c) Training Results from a 16 LSTM Model (Model 5) (d) Training Results from a 32 LSTM Model (Model 6) (e) Training Results from a 64 LSTM Model (Model 7) (f) Training Results from a 32-32-16 Model (Model 8) (g) Training Results from a 64-32-16 Model (Model 9) (h) Training Results from a 64-64-32 Model (Model 10) (i) Training Results from a 128-64-32 Model (Model 11)

Table 6: False Prediction Comparison per Model

| Model                | $R$  | $F$  | HL  | HR  | $\xi$ | $FS$ | $M$  | $TU$ | $\psi$ | Average (%) |
|----------------------|------|------|-----|-----|-------|------|------|------|--------|-------------|
| 32 CNN Model (Model 1)| 92.40| 92.80| 93.20| 96.90| 82.90 | 85.80| 73.70| 91.00| 97.10  | 89.53       |
| 64 CNN Model (Model 2)| 92.40| 94.60| 95.30| 96.80| 86.50 | 89.40| 83.90| 84.30| 97.90  | 91.23       |
| 128 CNN Model (Model 3)| 94.70| 94.30| 96.30| 98.30| 90.40 | 91.90| 88.10| 89.10| 96.50  | 93.29       |
| 32-32-16 Model (Model 8)| 99.50| 97.00| 95.40| 97.70| 93.10 | 94.20| 95.50| 96.70| 98.30  | 95.89       |
| 64-32-16 Model (Model 9)| 99.60| 96.00| 91.30| 92.50| 95.20 | 93.30| 97.60| 93.40| 98.30  | 95.24       |
| 64-64-32 Model (Model 10)| 99.70| 95.40| 97.60| 97.80| 95.10 | 90.40| 98.50| 95.90| 99.00  | 96.60       |
| 128-64-32 Model (Model 11)| 99.50| 95.10| 98.90| 97.30| 92.20 | 92.90| 94.70| 95.90| 99.30  | 96.20       |
| Average (%)          | 0.57 | 0.47 | 0.57 | 0.21 | 1.34  | 0.63 | 1.35 | 0.83 | 0.94   |             |
needs to also consider the orientation of the hand to make different gestures. With an average result of 96.6% accuracy, it is concluded that a CNN-RNN Neural Network Model approach is very effective in identifying a sequential data such as the muscle tension data stream. A drone can be effectively controlled with an EMG based controller, looking at the accuracy of the model developed in this research. From the conclusions stated above, there are a few suggestions on improving this research. More training data should be collected from a more varying range of age and muscle mass. In this research the use dataset only contains 14000 samples of varying gestures from a limited range of age (17-21). Providing a more varying age range would help improve the robustness of the models. An important thing to note is that the muscles of teens and adults have mostly developed and have the characteristics. However, the muscles of children are still in their development stage and might give a more varying EMG patterns and output. To increase the variety and accuracy of gesture classification, the motion of the hand should also be tracked. This research only limited the use of muscle sensors and it was proven to be not sufficient for a variative gesture identification. The drone was just used to prove the effectiveness of an embedded system implementation. Lian et al. (2017) have tried implementing EMG gesture identification on internet of things (IoT) with only 89% accuracy using a tree-KNN algorithm. With a 96.60% accuracy provided by this research, it is more viable to control IoT devices using the model that this research has produced.

Acknowledgments
This work is supported by Research Grant No. 20/AKM/MONOPNT/2019 of the Ministry of Research, Technology and Higher Education of the Republic of Indonesia, managed by LPPM of Universitas Pelita Harapan, Research Contract No. 193/LPPM-UPH/V/2019.

References
[1] A. Nagar and X. Zhu, “Gesture control by wrist surface electromyography,” 2015 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2015, pp. 556–561, 2015, doi: 10.1109/PER-COMW.2015.7134098.
[2] Mahmoud Abduo and M. Galster, “Myo Gesture Control Armband for Medical Applications,” Dep. Comput. Sci. Softw. Eng. Univ. Canterbury, no. October, pp. 4–23, 2015.
[3] T. Phienthrakul, “Armband Gesture Recognition on Electromyography Signal for Virtual Control,” 2018
10th Int. Conf. Knowl. Smart Technol. Cybern. Next Decad. KST 2018, pp. 149–153, 2018, doi: 10.1109/KST.2018.8426118.

[4] A. Olsson, “sEMG Classification with Convolutional Neural Networks,” 2018.

[5] R. Laezza, “Deep Neural Networks for Myoelectric Pattern Recognition An Implementation for Multifunctional Control,” 2018.

[6] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” no. November, 2015.

[7] Y. LeCun, C. Cortes, and C. J. C. Burges, “The MNIST Database of Handwritten Digits.”

[8] S. Hochreiter, “Long Short-Term Memory,” vol. 1780, pp. 1735–1780, 1997.

[9] Abduo, M., & Galster, M. (2015). Myo Gesture Control Armband for Medical Applications. Dep. Comput. Sci. Softw. Eng. Univ. Canterbury, October, pp. 4–23, 2015.

[10] Betts, J. G., Desaix, P., Johnson, E., Johnson, J.E., Korol, O., Kruse, D., Poe, B., Wise, J.A., Womble, M., and Young, K.A. (2017). Anatomy and Physiology. Openstax.

[11] Bilbao, I. & Bilbao, J. (2017). Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks. 2017 IEEE 8th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2017, 2018-January, 173–177, https://doi.org/10.1109/INTELCIS.2017.8260032.

[12] Borglin, J. (2011). Classification of hand movements using multi-channel EMG.

[13] De Luca, C. J. (1979). Physiology and Mathematics of Myoelectric Signals. IEEE Transaction on Biomed. Eng.,6, 313–325. DOI: 10.1109/TBME.1979.326594