

Design, implementation and evaluation for a high precision prosthetic hand using MyoBand and Random Forest algorithm

Duc Nguyen^{1,*}, Thien Pham², Tho Quan¹



Use your smartphone to scan this QR code and download this article

ABSTRACT

A prosthesis is an equipment provided to people who lost one or some parts of their limbs to help them having almost normal behaviors in daily or hard activities. The convenience and intelligence of devices should create easiness and flexibility for users. Artificial devices require interdisciplinary collaboration from neurosurgeons, surgical surgeons, physiotherapists and equipment development. Computer engineering plays a crucial role in the design step, supporting manufacturing, training and recognition to match the desirability of customers. Moreover, users need a wide range of different options such as an aesthetic functional material, a myoelectric mechanism, a body-powered appliance or an activity specified device. Thus, the flexible configuration, the proper features and the cost are some important factors that drive user's selection to the prosthesis. In this article, we describe an effective and powerful solution for analyzing, designing hardware and implementing software to train and recognize hand gestures for prosthetic arms. Moreover, we provide evaluation data of the method compared with similar approaches to support our design and implementation. This is fairly a complete system, making it a convenient solution for hand-cutoff people to control prosthetic hands using their electromyography signals. Statistical results with evaluations show that the device can respond correspondingly and the method creates promisingly recognition data after correct training processes. The prosthetic hardware implementation has also been simulated using a Light-emitting diode (LED) hand model with a high accuracy result.

Key words: Electromyography (EMG), MyoBand, Prosthetic hand, Random Forest Algorithms

INTRODUCTION

Prosthesis history

By the recent cutting-edge technologies, prostheses are developed to be as convenient as real body parts. From ancient Greece (210 BC), a Roman named Marcus Sergius made a prosthetic arm to fight when he lost his part previously. In 1579, a doctor, Ambroise Paré, recorded the prosthetic limb literature which emphasized mechanical support for the prosthetic hand, the concept which is being used until nowadays. Prosthetic technical knowledge was developed greatly during and after the war to achieve continuous progress over time.

The state-of-the-art prostheses are lightweight because of such advanced materials including plastic, aluminum and synthetic fibers. Besides, the appearance design is customized based on the user's convenience, flexibility and durability. Furthermore, the colors and shapes of the artificial parts are also manufactured to resemble human skin color and increase aesthetics.

Prosthesis manufacture classification

There are two main approaches in manufacturing a prosthesis, cost-oriented (mainly targets customers in developing countries) and quality-oriented (specifically dedicated for customers in developed countries).

Cost-oriented approach:

- Low-cost: Prosthetic limbs can be fabricated simply using a 3D printer. This type of material is suitable for children because the usage duration is relatively short as the children grow up quickly and the old prosthesis could not fit them anymore.
- Fast creation: Prosthetic limbs can be designed and manufactured within 24 hours.
- Adjustable: Devices are designed and customized using computer software, which is convenient for anybody.

There are lots of charity organizations providing prostheses for poor amputees using this method of manufacturing.

¹Ho Chi Minh University of Technology - Ho Chi Minh City National University, Vietnam

²Terralogic Vietnam Inc., Vietnam

Correspondence

Duc Nguyen, Ho Chi Minh University of Technology - Ho Chi Minh City National University, Vietnam

Email: duc.nguyenquang@hcmut.edu.vn

History

- Received: 6-8-2019
- Accepted: 21-8-2019
- Published: 17-10-2020

DOI : 10.32508/stdjet.v3iS11.536



Copyright

© VNU-HCM Press. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International license.



Cite this article : Nguyen D, Pham T, Quan T. **Design, implementation and evaluation for a high precision prosthetic hand using MyoBand and Random Forest algorithm.** *Sci. Tech. Dev. J. – Engineering and Technology*; 3(S1):SI28-SI39.

Quality-oriented approach:

- Prostheses that are controllable by thinking, for example: Targeted Muscle Reinnervation¹ is used to receive control signals from the brain following the nerve to the muscle, thereby can rule the limbs. When users think of moving their arms or legs, the signals from the brain direct the movement of the artificial parts. This paper uses electromyography in (remaining part of) the arm to transmit control commands to the prosthesis, which is an example of this manufacturing direction. Of course, such devices are more supreme, intelligent and pricey compared with the above type of devices.
- Prostheses that are able to sense and react instantly: The prosthetic limbs normally only follow the commands and therefore have limited abilities. People develop the parts so that they can recognize heat, cold, pain... to respond immediately to the environment². This is a progressive development trend of the prostheses nowadays.

EMG transmitted from the muscle

The nervous system generates a signal called the action potential to describe the desire to control a body part³. As a result, the limb control system needs to recognize the signal of the action through this electrical impulse. However, an action will go through some steps (Figure 1) to fully represent the whole cycle of it⁴. According to the timeline (ms) of the horizontal axis:

1. Taking a break.
2. Receiving stimulation, the cell receives ions that increase the voltage. This is the start-up phase for the electrical impulse.
3. At the top of the electrical impulse, the amount of ions begins to decrease.
4. The ion decreases and leaves the muscle cell, causing it to move toward the preparation period of the rest. This is the downward phase of electrical impulse.
5. The rest period is ready for another operation cycle.

The process of reading these activity cycles is used in EMG to identify muscle fiber movements in controlling prosthetic limbs.⁵

EMG prostheses

In general, prostheses can use EMG to turn nerve signals into desired actions on the limbs⁶. Signals from muscle are transmitted via sensors and converted into digital signals through the decoder. Data is processed and delivered to a processor for a recognition mechanism. Most of these prostheses use sensors attached to the rest of the user's limb or the head... to receive signals from users. Indicators from humans also carry information about strength and speed corresponding to the transmitted voltage and create natural feedback of the action⁷.

Signal processing algorithms require some step of noise reduction, normalization and feature extraction to filter all important information. One common technique is Root Mean Square (RMS), which relies on the average value of the signal and creates reliable data. As a result, the input is gathered for the training process.

The following sections of this paper are organized in the following order. Chapter *Related technologies* describes a number of studies related to the techniques used in this paper. Chapter *Hardware design* illustrates the technical details about hardware using MyoBand. Chapter *Method* explains software modules designed and implemented on the basis of Random Forest algorithm. Chapter *Results and discussion* provides the experimental results of the statistical evaluation compared with some similar methods. And the last Chapter *Conclusion* concludes the paper and shares some future intentions.

RELATED TECHNOLOGIES

Hardware

Sensors

Sensors are the devices contacting the user's remaining parts of the limbs, so that they play the most important role in data accuracy. Sensitive sensors reflect every small change in data to create precise pieces of signal reported. There are a huge number of devices designed for this purpose and the following options are the most popular ones:

- Myoware Muscle Sensor (Figure 2)

This sensor costs about \$34 and can be attached to the limb. Similar to other same type devices, the sensor requires EMG Electrodes (cost \$23 per set of 10, Figure 3) to work. The manufacturer recommends using two sensors to ensure the dual-channel to increase the accuracy of controlling output. However, a single channel is still basically acceptable for prostheses in

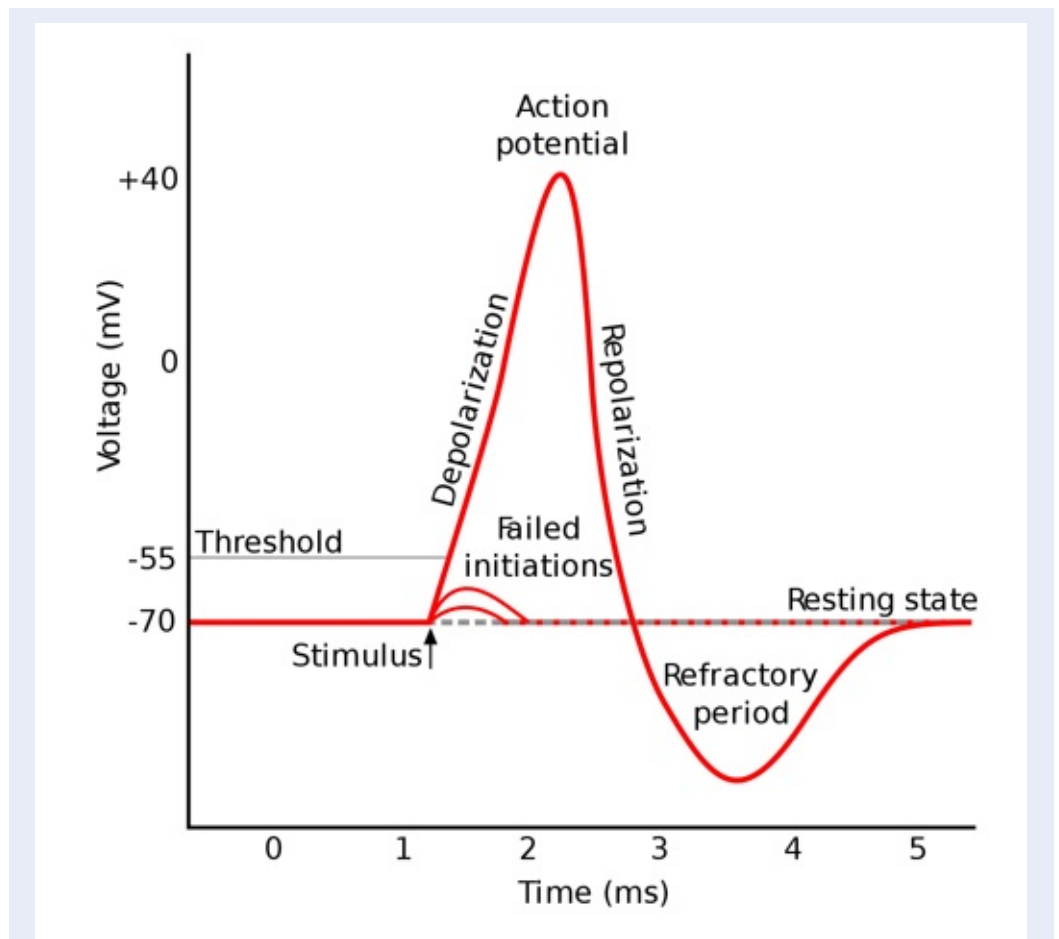


Figure 1: Different states of electrical impulses (https://commons.wikimedia.org/wiki/File:Action_potential.svg)

common testing. The limitation of this sensor is its precision is only adaptable for basic and general applications. Besides, the sensor is not flexible and convenient if users need an online sensor removal. As a result, this device is not a medical compatible material and is more applicable for research and academic purposes.

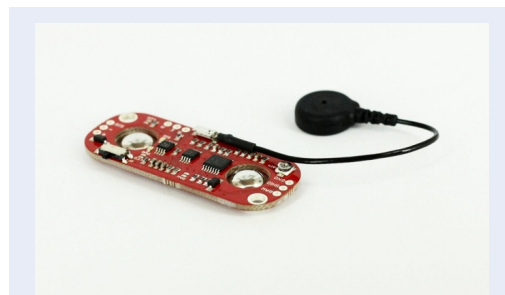


Figure 2: Myoware electromyography sensor



Figure 3: EMG electrodes

- MyoBand

Figure 5 describes a MyoBand sensor that is connected to an arm or a leg. The surrounding sensors increase the accuracy and speed of the output signals. Therefore, this device helps prostheses work closer to the operation of the limb. The price of this MyoBand

is approximately \$200. In this paper, we implement our experiment using this sensor.

The MyoBand has some advantages and disadvantages compared to other options. However, since the requirement of the implementation is limited to an evaluation of accuracy and processing time, we choose MyoBand in our design. Using this sensor, all micro engine vibrations are accurately identified and transmitted. Therefore, the hardware creates advantages for later training and recognition steps.

Motherboard

A compact motherboard with good connectivities has a high priority to be selected. For example, there is a list of common boards such as Arduino Uno, Adafruit ATmega32U4, Arduino Micro, Raspberry Pi...

In this paper, we choose Orange Pi PC Plus (Figure 4) based on its features, size and cost. Besides, this motherboard creates ease in connectivities and programming upon our selected MyoBand via Bluetooth.

Software

The software topic involves both training and recognition algorithms that will be discussed in this section. There are a wide range of different recognition techniques and we have an evaluation of some similar methods in this study. The common similarity of these procedures is to use the machine learning model to train the system through an existing sample dataset, thus reducing noise and increasing the accuracy of the movement. Later on, the recognition process identifies the commands or actions using the trained output. For example, the Least-Squares SVM⁸ is recommended to support the recognition for better speed and higher accuracy (less training process), or a classification model using Deep Learning Convolutional Neural Network⁹ is recommended to increase the accuracy majorly regardless of the simplicity of the neural network architecture. Some other techniques such as Kalman Filter, Random Forest... are proposed and their target are similar. In this paper we use Random Forest for training and recognition because of its flexibility, speed and accuracy.

HARDWARE DESIGN

Motherboard

Figure 4 shows the image of the motherboard we have used to run our proposed system. The motherboard's detail specification is listed in Table 1.

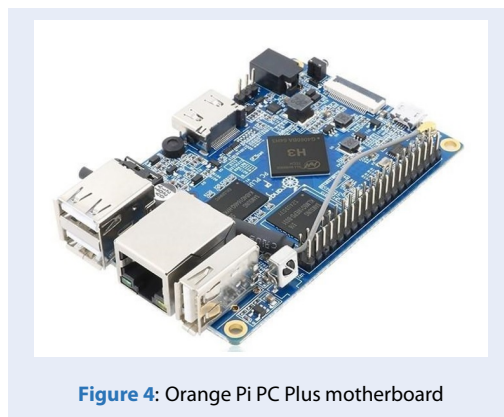


Figure 4: Orange Pi PC Plus motherboard

MyoBand electromyography sensor

The MyoBand sensor (Figure 5) is used because it satisfies our requirements of design and development, creating simplicity of connectivity and programming. The detailed specification of the MyoBand sensor is shown in Table 2. Figure 6 shows how a user wears MyoBand on his arm to archive electromyography data from the remaining part of the hand.

Training

MyoBand is used to get electromyography from the staying arm and transmit them to the motherboard via Bluetooth. On the motherboard, once has received the data, software modules process and normalize the data, providing input for the training module. To ease the training procedure, we develop a user interface for an application to help users interacting with the training component. Later on, this application provides the verification base for our prosthetic hand verification processes.

Figure 7 shows the example of predicting a gesture (users try to do the gesture they want and our system predicts and shows it on the screen).

Prosthetic hand model with LED light

Before designing a complete prosthetic hand, we used an intermediate version of a prosthesis to help users understand how the system works and how to interact with the device. Therefore, we use a LED hand model. Firstly, this model simulates artificial hands by simulating finger moves corresponding to the on/off LED on the model. Next, anytime the user thumb up/down, the corresponding LED is turned on/off respectively. Then, at the wrist, the LED corresponding to the actions of the wrist also turns coincidentally. Finally, this model is connected and fully controlled by a software module running on the motherboard.



Figure 5: MyoBand electromyography sensor



Figure 6: MyoBand sensor on the hand; (<http://irsi.eigsi.fr/wp-content/uploads/2014/11/Myoon-arm-blog-v32.jpg>)

Table 1: Hardware configuration of the motherboard

Hardware specifications	
CPU	H3 Quad-core Cortex-A7 H.265/HEVC 4K
GPU	Mali 400MP2 GPU @600MHz Supports OpenGL ES 2.0
Memory	1GB DDR3 (shared with GPU)
Storage	TF card (Max. 32GB) 8GB EMMC Flash
Network	10/100 Ethernet RJ45 Wifi 2.4GHz b/g/n
USB Ports	Three USB 2.0 HOST + One USB 2.0 OTG
Low-level peripherals	40 Pins Header, Raspberry Pi 3 B+ Compatible
Supported OS	Android Lubuntu, Debian, Raspbian Image

Table 2: Hardware configuration of MyoBand electromyography sensor

Hardware specifications	
Sensors	Medical Grade Stainless Steel EMG sensors Highly sensitive nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer
Processor	ARM Cortex M4
Communication	Bluetooth Smart Wireless Technology
Power and Battery	Built-in rechargeable lithium ion battery
Interface definition	
Arm size	Expandable between 7.5 - 13 inches (19 - 34 cm) forearm circumference



Figure 8: Model is being operated by the actual user

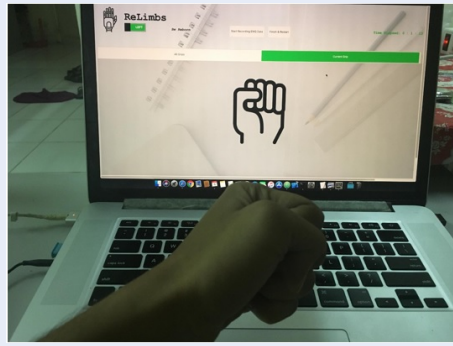


Figure 7: The system interface is displaying the current action

Function block diagram

Figure 9 shows a diagram of the function blocks and data streams. Firstly, the electromyography signals of the hand are transmitted to the EMG sensors in MyoBand. Next, after receiving a raw electrical signal, the sensor conducts pre-processing, noise filtering and transmits the data to the Orange Pi PC Plus motherboard via Bluetooth. Then, on the motherboard, we enable a Web UI to interact with users. Either a smartphone, a laptop or a PC with Wi-Fi connecting to the board can be used for interaction with the Web UI.

Users can start training actions as well as monitor the results of the prediction. By receiving a training order from the user, the motherboard immediately collects and trains data following the Random Forest algorithm. As a result, once the training process is completed, the input signals are automatically used to predict the commands (the predicted outcome is in trained actions). Based on the predicted results, the motherboard will provide appropriate control signals to the prosthetic arm or prosthetic model. Finally, the identified control signals turn on/off the LEDs corresponding to the shrinking fingers.

Advantages and disadvantages of this design

Advantages

Obviously with the above design, several different parts of hardware have been used in our system implementation. During the implementation process, with limited equipment situation, we found that this design has these advantages:

- Easy to implement, can be purchased in the market, not a costly solution.

- Intuitive, convenient for users to be familiar with the system.
- The motherboard is supported for easy connection and programming.

Disadvantages

Some disadvantages of the approach include:

- Unable to perform real actions.
- Can not fully simulate a complete prosthetic arm.

However, these are the foreseen inconveniences and we will consider them in the future plan of this study.

METHODS

Random Forest algorithm

Building a tree using CART

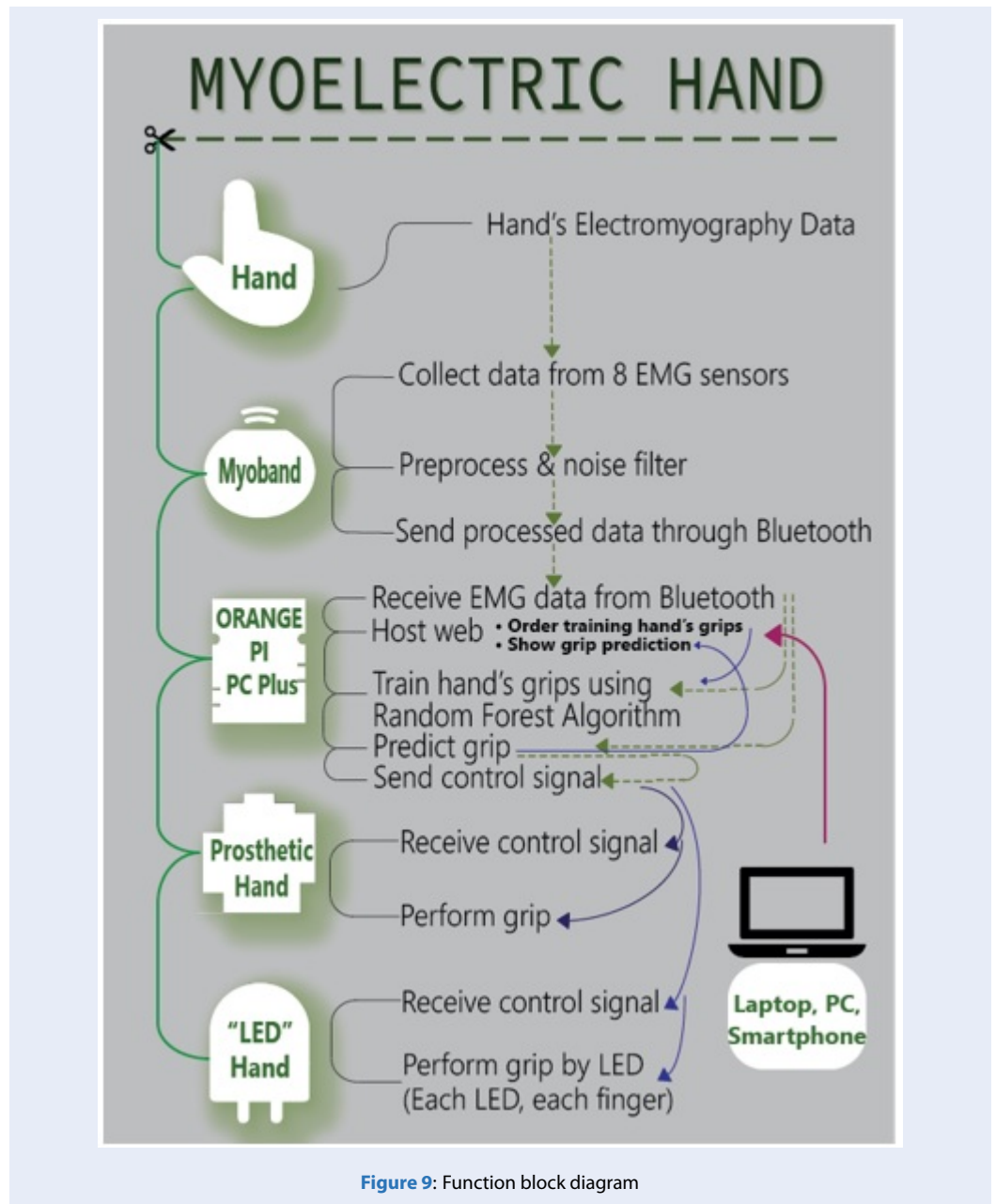
CART (classification and regression tree)¹⁰ is an algorithm used to build a decision tree. We will describe the main flow of the algorithm in the below sections. Firstly, a binary tree is considered for a CART objective. Input data is the attribute dataset of its classes and subclasses. Each set of n-attributes of a class is an n-dimensional vector. Next, at each node, the algorithm tries to find the best split point (greedy splitting approach) by scanning all n-attributes of the training data and calculating the Gini coefficients of the split point. In each attribute, the best split point will be chosen to compare with other attribute's one. The best split point of all n-attributes is considered as the root node. After that, the training dataset is divided into two parts based on the root node's condition. Then, at each node connected from the tree root, the algorithm continues to scan all n-attributes and calculates the Gini coefficients to divide the tree. The process runs until we get the stop condition.

Finally, the stop condition is configured when either all the leaf nodes belong to only one class or the number of samples at a node is lower than a threshold (specified case by case). As a result, we have a classification tree.

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (1)$$

The formula for calculating Gini coefficients at each node D is given in (1). In this equation, p_i is the probability of exporting data with an i label on the total data at a node.

The formula for calculating the Gini coefficient of the



dividing point at node D follows attribute A is given below:

$$Gini_A(D) = 1 - \sum_{j=1}^v \frac{|D_j|}{|D|} Gini(D_j), \quad (2)$$

in this equation, D_j is a child of D after branching.

Random Forest construction

Firstly, to build a software module based on the Random Forest algorithm¹¹ we randomly divide the training dataset into multiple subsets corresponding to the number of trees expected. Next, for each subset,

we generate a CART tree (using the described method with (1) and (2)). The special point of Random Forest is that in a tree, instead of scanning all n -attributes to find the best split point at each node, it limits the number of attributes that CART can scan. Then, the algorithm randomly selects attributes from n -attributes with the number of them is smaller than a predefined number. For the classification problem, the number of attributes is limited to each node is usually equal to \sqrt{n} (where n is the total number of attributes of the training dataset).

Later on, by conducting a classification of the class from the input data sample, Random Forest passes that sample through all the trees in the forest. Finally, each tree creates a subclass correspondingly and the subclass that has the maximum number of trees produced will be selected as the expected result.

Applying Random Forest in the project

Data archived from the MyoBand sensor has a speed of 50Hz (50 samples per second). Each time we transfer data from eight EMG sensors, we have an eight-dimensional vector. As a result, we plan to label the vector with corresponding actions.

Firstly, in order to obtain a training dataset, each action is collected by at least 100 data samples. Next, we train the Random Forest model using 12 trees. The prediction output of a chosen action is the most anticipated action of the 36 most recent data samples.

Web UI

Since the application needs to interact with users easily, we build a web-app for accessible development. The site is hosted on the motherboard itself so that the interconnectivity within different modules is simple. Website data is updated in real-time with the motherboard. Figure 10 shows the prediction's probability corresponding to each gesture. The gesture prediction result is presented in the Prediction tab (Figure 11). Also, the training interface (Figure 12) shows each gesture at a time for collecting the user's myography data corresponding to that gesture.

RESULTS AND DISCUSSION

Initial results

The time from the start of the action to the time when the algorithm can recognize the action is approximately 720ms.

By the comparison during practical runtime, we select 36 recent samples which should create the best accuracy. Otherwise, if less than 30 data samples are collected, the accuracy is relatively reduced (no characteristic of action is found). On the contrary, if more than 40 samples are archived, the action is correspondingly time-consuming (since there is remaining time to transmit control signals to the hardware).

Therefore, the embedded board running Linux operating system creates advantages for our development. Besides, we have conducted several tests on a practical handicapped user using a LED hand model (Figure 8). After having trained the model and been familiar with the system, users can control the scheme and rule the hardware with accuracy up to 47/50 actions (over 90%).

Advantages and disadvantages of the method

Advantages

- Hi-speed training and prediction.
- Inexpensive hardware resources, suitable for deployment on embedded boards.
- High-accuracy (90%+).

Disadvantages

- Users need to study the behavior of how the system works and get used to its processing approaches.
- A stable embedded Linux board is required.

Compare with other algorithms

The side-by-side review data on Table 3 with related methods shows that our approach ensures the best output based on two main-factors (processing time for training and recognition, as well as accuracy). Specifically, our system has real-time processing for a sample of approximately 20ms (because of the limitation of 20ms sampling period from hardware devices), but still reaches an average accuracy of 47/50 (94%). In the same scenarios, the K-Nearest Neighbors algorithm costs the same running time but can only achieve lower accuracy of 38/50 (76%). Another example, the Multilayer Perceptron algorithm provides the same accuracy, but its execution time takes nearly five times higher than our approach. In summary, the Random Forest algorithm outperforms other methods by the combination of processing time and accuracy, thus we select it in our implementation.

The feasibility of upgrading

Based on the above promising results, it is applicable to create prosthetic limbs using artificial intelligence. In terms of cost, it will be a bit higher than other existing designs since we use modern sensors and embedded boards. However, as the motherboard is running embedded Linux, the program is upgradable and the algorithm is configurable. Besides, our hardware design creates much ease in interface connectivities, program maintenance, and software enhancement.

CONCLUSION

Current solution

By using the Random Forest algorithm, the system is able to be trained and recognize the actions from limbs successfully. With the above results, the solution can be effectively applied in the practical creation of prostheses for real patients.

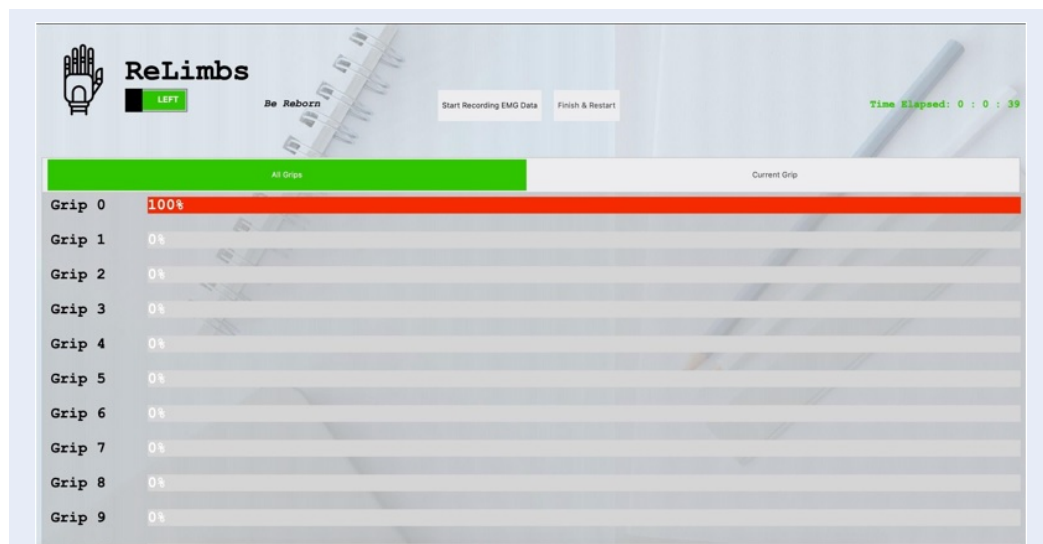


Figure 10: Interface for evaluating the ability to perform actions

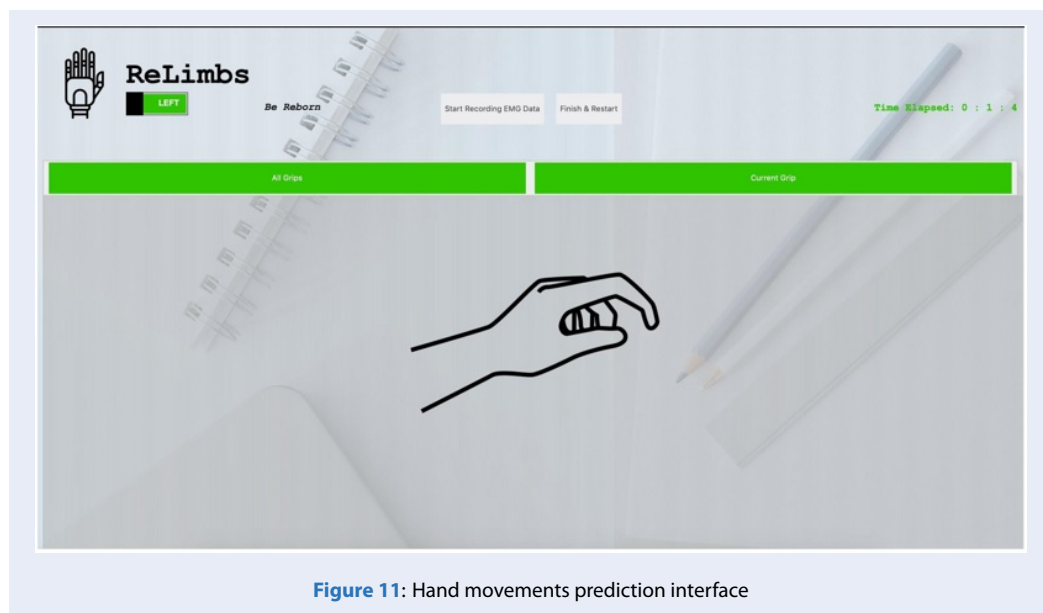


Figure 11: Hand movements prediction interface

Table 3: Comparison of Random Forest with some similar classification algorithms

Algorithm	Predict time (s)	Training 1 sample time (s)	Accuracy (50 times)
Random Forest	0.002 – 0.02	0.016 – 0.027	47/50
K-Nearest Neighbors	0.002 – 0.020	0.003 – 0.022	38/50
Multi-layer Perceptron	0.0005 – 0.020	0.0002 – 0.1175	48/50

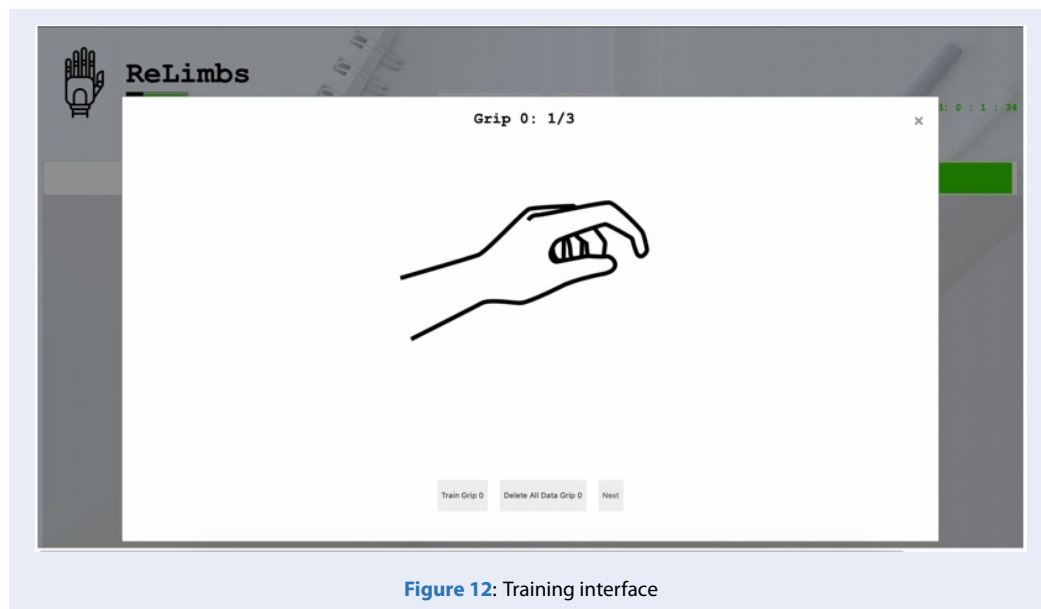


Figure 12: Training interface

Future development

We are investigating some approaches, so that the next phase of this study is applicable, such as:

- Use 3D printing technology or specialized hand-held devices to complete the final stage of the system to help hand-cutoff people use it in their daily activities.
- Apply Deep Learning¹² models to increase the accuracy and speed of prediction.
- Combine a number of other tools to help prosthetic hands respond to external conditions.
- Research to apply this model for prosthetic legs.

CONFLICT OF INTEREST

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

AUTHOR'S CONTRIBUTIONS

This work was written through the contributions of all authors.

REFERENCES

1. Cheesborough JE, Smith LH, Kuiken TA, Dumanian GA. Targeted muscle reinnervation and advanced prosthetic arms. *Seminars in plastic surgery*, Thieme Medical Publishers. 2015;29(62). PMID: 25685105. Available from: <https://doi.org/10.1055/s-0035-1544166>.
2. Nistler A, Downey C, Currie CT, McDonald D, Port G, Sabatino J, et al. Design and development of a myoelectric transradial prosthesis. 2017;.
3. Talbot K. Using arduino to design a myoelectric prosthetic. 2014;.
4. Cipriani C, Zaccone F, Micera S, Carrozza MC. On the shared control of an emg-controlled prosthetic hand: analysis of user-prosthesis interaction. *IEEE Transactions on Robotics*. 2008;24(1):170–184. Available from: <https://doi.org/10.1109/TRO.2007.910708>.
5. Liarakapis MV, Artemiadis PK, Kyriakopoulos KJ, Manolakos ES. A learning scheme for reach to grasp movements: On emg-based interfaces using task specific motion decoding models. *IEEE journal of biomedical and health informatics*. 2013;17(5):915–921. PMID: 25055370. Available from: <https://doi.org/10.1109/JBHI.2013.2259594>.
6. Zecca M, Micera S, Carrozza MC, Dario P. Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical ReviewsTM in Biomedical Engineering*. 2002;30(46). PMID: 12739757. Available from: <https://doi.org/10.1615/CritRevBiomedEng.v30.i456.80>.
7. Atzori M, Gijssberts A, Castellini C, Caputo B, Hager AGM, Elsig S, et al. Effect of clinical parameters on the control of myoelectric robotic prosthetic hands. *Journal of Rehabilitation Research & Development*. 2016;53(3). PMID: 27272750. Available from: <https://doi.org/10.1682/JRRD.2014.09.0218>.
8. Orabona F, Castellini C, Caputo B, Fiorilla AE, Sandini G. Model adaptation with least-squares svm for adaptive hand prosthetics. 2009;p. 2897–2903. Available from: <https://doi.org/10.1109/ROBOT.2009.5152247>.
9. Atzori M, Cognolato M, Iler HM. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in neurobotics*. 2016;10(9). PMID: 27656140. Available from: <https://doi.org/10.3389/fnbot.2016.00009>.
10. Liaw A, Wiener M, et al. Classification and regression by randomforest. *R news*. 2002;2(3):18–22.
11. Denil M, Matheson D, Freitas ND. Narrowing the gap: Random forests in theory and in practice. *International conference on machine learning*. 2014;p. 665–673.
12. Ghazaei G, Alameer A, Degenaar P, Morgan G, Nazarpour K. Deep learning-based artificial vision for grasp classification in myoelectric hands. *Journal of neural engineering*. 2017;14(3):036025. PMID: 28467317. Available from: <https://doi.org/10.1088/1741-2552/aa6802>.

Thiết kế và hiện thực cánh tay giả độ chính xác cao sử dụng MyoBand và thuật toán Rừng ngẫu nhiên

Nguyễn Quang Đức^{1,*}, Phạm Công Thiện², Quán Thành Thơ¹



Use your smartphone to scan this QR code and download this article

TÓM TẮT

Chi giả (tay hoặc chân giả) là thiết bị được cung cấp cho người khuyết tật bị mất một phần chi, giúp họ có được hoạt động gần như bình thường qua hoạt động hằng ngày hoặc các hoạt động gắng sức. Chi giả càng được cải tiến tiện lợi và thông minh thì con người càng dễ điều khiển và hoạt động của họ càng linh hoạt. Việc sản xuất và phát triển chi giả là công việc liên ngành của các bác sĩ thần kinh, bác sĩ phẫu thuật ngoại khoa, bác sĩ vật lý trị liệu và các kỹ sư chế tạo phát triển thiết bị. Khoa học máy tính đóng vai trò lớn trong bước thiết kế, hỗ trợ chế tạo, huấn luyện và nhận dạng để phù hợp với ngoại hình và chức năng cần thiết của khách hàng. Người sử dụng cũng cần có nhiều lựa chọn chẳng hạn là một thiết bị chức năng thẩm mỹ (aesthetic functional device), một thiết bị điện cơ (myoelectric), một thiết bị chạy bằng cơ thể (body-powered device) hoặc là thiết bị dành cho hoạt động nhất định (activity specified device). Càng có nhiều lựa chọn và càng phù hợp về chức năng, giá thành, thì việc sử dụng của họ càng thuận lợi. Trong bài báo này, chúng tôi đưa ra một giải pháp hữu hiệu và mạnh mẽ trong việc phân tích, thiết kế phần cứng và hiện thực phần mềm để huấn luyện và nhận dạng cử chỉ hoạt động tay cho cánh tay giả. Đây là một hệ thống khá đầy đủ, thuận tiện làm giải pháp cho người khuyết tật tay có thể điều khiển tay giả thông qua chuyển động cơ điều khiển của họ. Kết quả thống kê cho thấy khả năng đáp ứng tốt và rất triển vọng của bộ huấn luyện (training) và nhận dạng (recognition). Việc hiện thực ra phần cứng tay giả cũng đã được mô phỏng qua một mô hình tay bằng đèn LED cho độ chính xác cao và sẽ được hiện thực trong tương lai.

Từ khóa: Điện cơ, Cảm biến MyoBand, Tay giả, Rừng ngẫu nhiên

¹Trường Đại học Bách khoa - Đại học Quốc gia Thành phố Hồ Chí Minh, Việt Nam

²Công ty Terralogic Việt Nam

Liên hệ

Nguyễn Quang Đức, Trường Đại học Bách khoa - Đại học Quốc gia Thành phố Hồ Chí Minh, Việt Nam

Email: duc.nguyenquang@hcmut.edu.vn

Lịch sử

- Ngày nhận: 6-8-2019
- Ngày chấp nhận: 21-8-2019
- Ngày đăng: 17-10-2020

DOI :10.32508/stdjet.v3iS1.536



Bản quyền

© ĐHQG Tp.HCM. Đây là bài báo công bố mở được phát hành theo các điều khoản của the Creative Commons Attribution 4.0 International license.



Trích dẫn bài báo này: Đức N Q, Thiện P C, Thơ Q T. **Thiết kế và hiện thực cánh tay giả độ chính xác cao sử dụng MyoBand và thuật toán Rừng ngẫu nhiên.** *Sci. Tech. Dev. J. - Eng. Tech.*; 3(S1):SI28-SI39.