IMPORTING AND ANALYSING LIVE ELECTROENCEPHALOGRAM DATA IN MATLAB/SIMULINK ENVIRONMENT

***József Sárosi*[1] *- Roland Elek*[2]**
[1]Technical Department, Faculty of Engineering, University of Szeged,
H-6724 Szeged, Mars tér 7., Hungary
e-mail: sarosi@mk.u-szeged.hu
[2]Faculty of Science and Informatics, University of Szeged,
H-6720 Szeged, Árpád tér 2., Hungary
e-mail: Elek.Roland@stud.u-szeged.hu

ABSTRACT

The utilization of non-invasive human-computer interfaces makes it possible to develop limb prostheses that recognize the cognitive intentions of the user, and act accordingly. Our goal is creating such a prosthesis that is actuated by pneumatic artificial muscles (PAM). One of the possible approaches to accomplishing this is electroencephalography (EEG). In this paper we describe a method of adapting EEG signals acquired with a concrete EEG device, the Emotiv EPOC, to the MATLAB/Simulink environment, and recommend the next step to using them.

1. INTRODUCTION

The use and development of non-invasive human-computer interfaces is a new and promising area. Research results can be used in a wide range of areas, including rehabilitation of motion-impaired people with limb prostheses that require no special control methods from the user's point of view, but detect and react to the intentions of the user in a natural way. Our goal is developing a prosthesis that accomplishes this goal by using pneumatic artificial muscles as actuators and an electroencephalography-based brain-computer interface as its data source.

Electroencephalography (EEG) is the field of measuring, processing, and analysing small electric voltage fluctuations with a frequency between DC and approximately 100 Hz, generated by neurons firing in the brain, that can be measured using electrodes attached to the scalp, commonly known as brainwaves. Patterns in these voltage fluctuations reflect specific activities in the brain. Depending on the number and parameters of sensors used, recognizable activities range from simple emotions (e.g. alertness) to cognitive concepts, such as „ left" or „disappear".

The most important property of EEG is its non-invasiveness, which means it is not needed to break the integrity of the human body in any way (e.g. needles) to use the technique. Multiple types of sensors exist. Most medical devices require a conductive, adhesive substance to attach to the scalp. Other designs use a headset form factor that supports the sensors in their intended positions, eliminating the need that the conductive substance be adhesive. Some devices also do not need any separate conductive substance for proper operation.

A similar technology to EEG is electromyography (EMG). EMG operates by the same general principle as EEG, and it is generally used to sense and analyse electrical activity of skeletal

muscles. Some EMG data can be present in EEG signals, especially activity resulting from ocular and facial muscle actuation in data acquired from sensors close to the forehead, shown in Figure 4. This can be used to detect other states of the user, e. g. blinks, direction of looking, and facial expressions.
The aforementioned states of the user are reflected in the spectrum and spatial distribution of electrical fluctuations measured by the sensors. These patterns can be recognized using data classification algorithms.

A good environment to experiment in with different processing and classification methods is vital to obtain the best possible results. We chose Simulink, a part of the well-known MATLAB environment, as it also provides tools to simulate the mechanical system of the prosthesis we are developing.

2. MATERIALS AND METHODS

The concrete EEG device we are using is an EPOC neuroheadset by Emotiv Systems. It is a wireless PC peripheral that features 14 channels, 2 reference electrodes, and a 2-axis (yaw and pitch) gyroscope. Its sensing elements are wet electrodes, which mean they have to be moistened with a saline solution to maintain proper electrical contact with the user's scalp [1]. The positions of the electrodes are in accordance with the international 10-20 system. The EPOC is a commercially available product that is sold with different types of licensing. We chose the Research Edition, because it gives access to raw EEG and gyroscope data. The EPOC set can be seen in Figure 1. On the left is the hydrator pack for moistening the felt pads of the sensors, which are currently inserted into the hydrator. On the right is the wireless receiver that can be connected to a PC. The next item from the right is the saline solution used for moistening the felt pads to ensure high contact quality and good electrical conductivity. On the bottom is a standard mini-USB cable for charging the battery of the headset. In the centre is the headset itself.

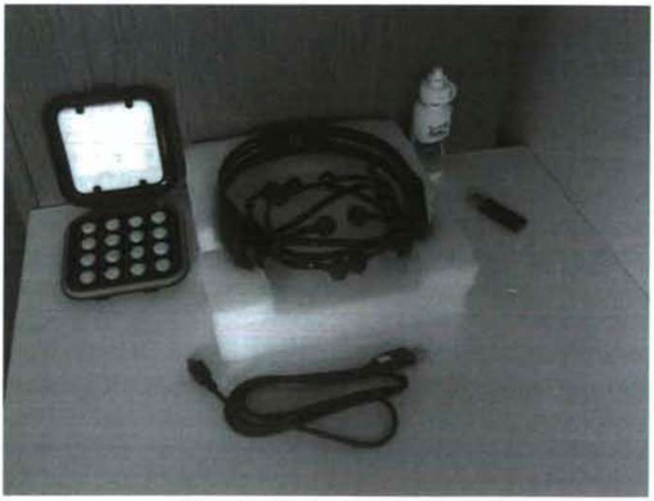

Figure 1. The Emotiv EPOC set

The Simulink environment currently lacks support for the EPOC. To be able to conduct real-time experiments, we need to be able to directly import live streams of data from the headset. To this end, we have created two Simulink blocks that accomplish this task. One of these blocks import the 14 EEG channels, and the other adapts the gyroscope data. The latter was created because the method required to acquire gyroscope data from the headset is almost the same as for EEG data, so it was straightforward to implement, and could be used to test the implementation controllably, as head movements are easily reproduced multiple times and provide more predictable output than EEG patterns.

3. RESULTS AND DISCUSSION

Multiple methods exist that enable the use of the collected data in simulations. One of these is recording the data to a file, and then using the file as the data source. A disadvantage of this method is that it permits only off-line processing of the data. Another method is using C MEX S-functions. A C MEX S-function is an extension to the MATLAB environment written in the C programming language that can be used as an optionally multi-input, multi-output system in Simulink. The Emotiv EPOC Research Edition Software Development Kit (SDK) provides ANSI C libraries to facilitate communication with the headset. A C MEX S-function, being a C program, can thus access the data provided by the SDK, and make it available on its output ports in Simulink. As mentioned above, two Simulink blocks were created this way, one for gyroscope data, and one for the 14 EEG channels. These two blocks can be seen in Figure 2.

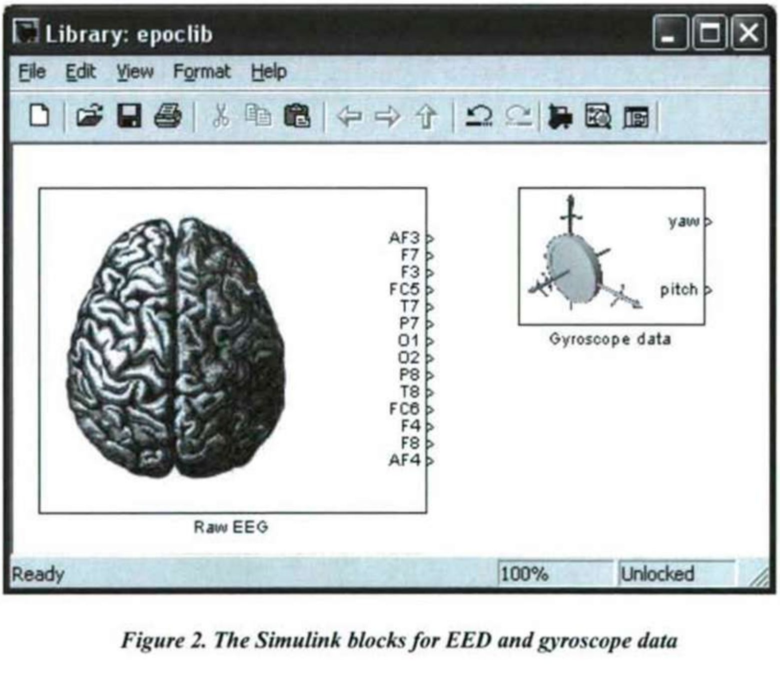

Figure 2. The Simulink blocks for EED and gyroscope data

The process of reading live data from the headset can be divided into three distinct major stages: initialization, acquisition, and clean-up. The first task of the initialization stage is to establish a connection to the Emotiv headset software called the EmoEngine. The available headsets are then discovered. The next step is to set up buffers for data acquisition, and to start the process itself. In the acquisition stage, measurements are taken periodically, and the results are stored in an internal data buffer in EmoEngine. This buffer must be read to a local buffer at a rate that does not permit the buffer in EmoEngine to overflow. As our Simulink block always reads the whole EmoEngine data buffer, and the smallest local buffer that can be set up holds one second of data, an overflow condition is theoretically possible only if the sampling time is longer than one second. The outputs of the aforementioned Simulink blocks can be updated from the local buffer. In the clean-up stage, which in the case of Simulink blocks, is executed on stopping the simulation, first the data acquisition is disabled using an explicit function call. The local buffers are then freed. As a last step, the clean-up stage closes the connection to EmoEngine. For testing purposes, we visualized EEG channels F3 and F7 on a scope in real time. As the raw signal acquired from the headset is discrete-valued and discrete-time, there are sharp jumps on the original graph of the channels at time values that are multiples of the sampling time. This can be solved by using a low-pass filter on the data. On Figure 4, the signals were filtered using low-pass filters with a cut-off frequency of 20 Hz. Real-time operation was achieved using a third-party extension called RTBlock, shown in Figure 3. The artefacts circled in red in Figure 4 are electromyographic signals denoting blinks. The yellow graph corresponds to channel F3, and the purple one to F7.

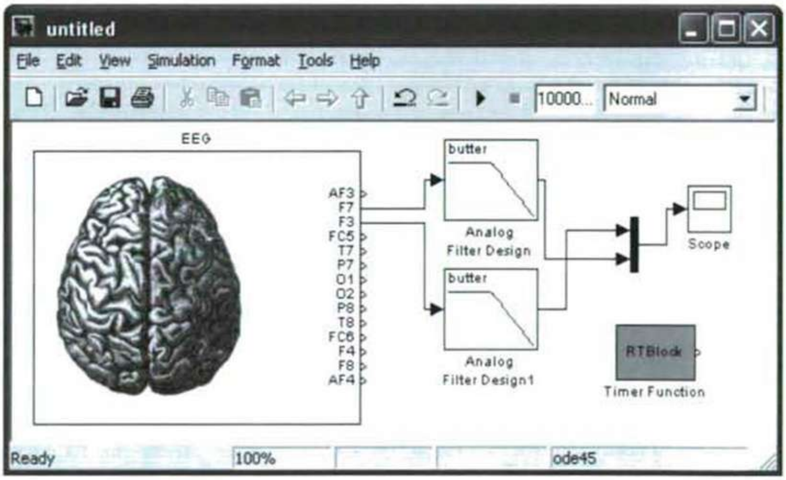

Figure 3. The Simulink model for experimental filtering

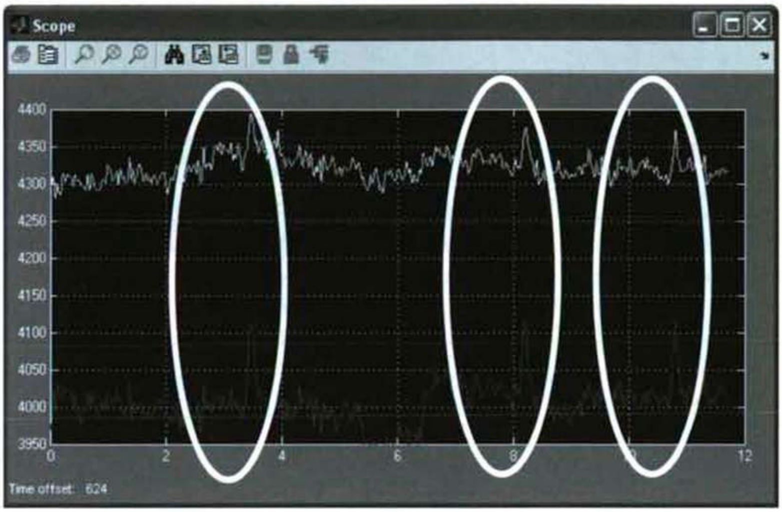

Figure 4. Measurements taken on channels F3 (yellow) and F7 (purple), filtered with Butterworth low-pass filters with 20 Hz cut-off frequency

Having imported the signals, it is possible to experiment with various signal processing and data classification tools. For example, as relevant information is carried by the spectrum and spatial distribution of the signals, it is useful to carry out frequency domain analysis on the data. Simulink provides various tools to support this, such as Fourier and wavelet transforms, filters, and other signal processing blocks.

For our task of controlling a prosthetic limb, a major challenge is to reliably detect signatures of the user's cognitive intentions in the data. Multiple present EEG-based brain-computer interfaces, such as the ones described in [2] and [3], utilize a type of supervised learning classifier to accomplish this. There has been research on using neural networks to classify the EEG data that shows promising results, described in [4].

4. CONCLUSION

In this paper, we described the method of importing live EEG data to the Simulink environment, and discussed the possible approaches to using them as a data source for controlling a pneumatic artificial muscle actuated limb prosthesis. It was also stated that, besides experimenting with signal processing and data classification methods, it is possible to simulate the mechanical hardware of the prosthesis, and control the simulated hardware with the outputs of the classifier, making it possible to test the whole system in a virtual environment, facilitating changes, and accelerating research.

REFERENCES

1. Official website of Emotiv (http://emotiv.com/)
2. Müller, K. R., Krauledat, M., Dornhege, G., Curio, G. and Blankertz, B. (2004): Machine Learning Techniques for Brain-computer Interfaces. Biomedizinische Technik, 2004:49 (1), 11-22. p.
3. Palaniappan, R., Paramesran, R., Nishida, S. and Saiwaki, N. (2003): A new Brain-computer Interface Design Using Fuzzy ARTMAP. IEEE Transactions on Neural Systems and Rehabilitation Engineering , 2003:10 (3), 140-148. p.
4. Haselsteiner, E. and Pfurtscheller, G. (2000): Using Time-dependent Neural Networks for EEG Classification. IEEE Transactions on Rehabilitation Engineering, 2000:8 (4), 457-463. p.