AN EEG MOBILE DEVICE AS A GAME CONTROLLER

ALEKSANDER DAWID* AND PAWEŁ BUCHWALD

Department of Transport and Computer Science
WSB University, Cieplaka 1c
41-300 Dąbrowa Górnicza, Poland
*adawid@wsb.edu.pl

(received: 04.08.2021; revised: 18.10.2021;
accepted: 21.10.2021; published online: 30.12.2021)

Abstract: In this work the real-time control of computer games was explored by a single electrode mobile electroencephalography (EEG) device with a Bluetooth interface. The amplitude variation in the two frequency bands of 4-12 Hz and 60-200 Hz was selected as the real-time control parameter. The frequency-domain of a raw EEG signal was calculated using the discrete Fourier transform. The time-dependent signal samples equal to 512, 1024, and 2048 time points in size were used in our research. The well-known classic Pong game was used to try out our controller. The developed software handles communication with the device and real-time game rendering. The .NET Framework with the C# programming language was used as a development tool. 50 gameplay trials were made for each controller setup. The obtained results are promising for the possible use of the device in real-time communication with computer devices for people with hand disabilities.

Keywords: EEG, EEG signal disturbance, signal processing, amplitude driven control, game controller

DOI: https://doi.org/10.34808/tq2021/25.4/b

1. Introduction

The hands have always been a crucial part of the human-computer interface. The concept of using computers by typing, clicking, touching and swiping implies the use of hands and fingers. Humans with dysfunctional hands feel lost in the modern world full of computers. Nevertheless, the hands are not moving on their own but it is the brain that controls them. For many years researchers have focused on how to transfer the commands directly from the brain to the computer. There have been many approaches to make a brain-computer interface (BCI). It all began in 1929 when Hans Berger, for the first time, described EEG by the frequency bands [1]. Since that time, scientists have believed that someday people will communicate without using words, just by sending thoughts to each other. The first applications in this area showed up in the 1970s. It was a time of rapid development in the computer science. In 1977 Vidal showed that humans could control a two-dimensional cursor inside a computer-generated maze [2]. The perfect condition for testing BCI is in the case of a paralyzed human. He or she is unable to move any muscle, so there is less disturbance in the EEG signal. EEG devices usually read a mix of EEG signals and the disturbance signal from scalp muscles in the case of a healthy person. One of the first successes in decoding EEG signals was recognition of the states of attention and relaxation [3]. These two states were the basis for Durka and his team to construct a device for completely paralyzed persons. The hardware of this solution consisted of a stationary medical class EEG device connected to a computer. The operation of this device assumes that a disabled person is not blind and understands the alphabet. The patient communicates with the outer world by watching letters or phrases displayed on the computer screen. When the patient wants to choose a letter, or a phrase, the level of attention on the EEG readings is rising [4]. This method needs some training from the disabled person. Another approach to this problem is to force the machine to understand our intentions. The computer has to learn the state of the EEG signal. There are some research papers about people's identification by EEG signal patterns [5–8]. These solutions led to the creation of the first event control systems for computer games. One of the most popular solutions is the use of the steady-state visually evoked potential (SSVEP) [9]. This method consists in displaying flashing characters with a constant frequency. If we have the desired sign, the reaction to it changes the phase of the EEG signal. This change could serve as a command for a computer program. This method of control has found application in many computer games [10–14]. However, it cannot control game situations that require reaction times of less than one second. On the other hand, cheap EEG devices on the market that could potentially be used as game controllers can only sample the signal at 512 samples per second. This result imposes a one-second delay in controlling an object in a computer game.

In this project, we want to show that we can create a controller capable of controlling a dynamic computer game, like Pong, using EEG signals from a one-electrode device [15]. To move the object, we suggest using EEG signals that come from the facial muscles (band 60-200 Hz) and the tongue movement (band 4-12 Hz). In our approach, we will not try to recognize the signal by any inference method. We will mimic changes of the signal amplitude to the movement of the player's object. As far as we know, there is no report on using this direct approach. This solution can find application in controlling medical robots [16] as extra support for hands.

2. System design

We used the Neurosky Mindwave mobile device to research the controller [17]. The device is a low-cost headset that can read the EEG signal with a single electrode. The advantage of such a device is its ease of use. Putting such a device on the head does not cause any problems, even for inexperienced users. The active electrode in the device should be two inches above the user's left eyebrow. According to the international 10-20 EEG Electrode Placement Standard [18], this location has the symbol fp1. The second electrode, known as the reference electrode, is fastened to the user's ear with a clip. The device for communication with a computer or mobile devices uses the Bluetooth wireless communication (BT). This device can read the states of concentration and meditation, measured in percentage. Besides, this EEG set allows reading individual bands described by Hans Berger. The functionality that is most interesting for this project is the raw signal transmission. As we know, the device samples the signal at 512 samples per second. Each element is a 32-bit integer. The next part of this system is a computer that receives the signal from the EEG headset. The software task is to receive data from the EEG device and transform it into a form suitable for use as a controller. This software is a critical part of this entire system. The comfort of using the system by the player depends on it. In Figure 1, we can see the system flowchart. The raw signal created, for example, during the process of contracting the facial muscles, goes to the computer via the BT interface and then it is processed by the software. In this processing, the Fourier transform converts a time-dependent signal into a frequency-dependent signal. We use the amplitude over a given frequency range to control the player's motion.

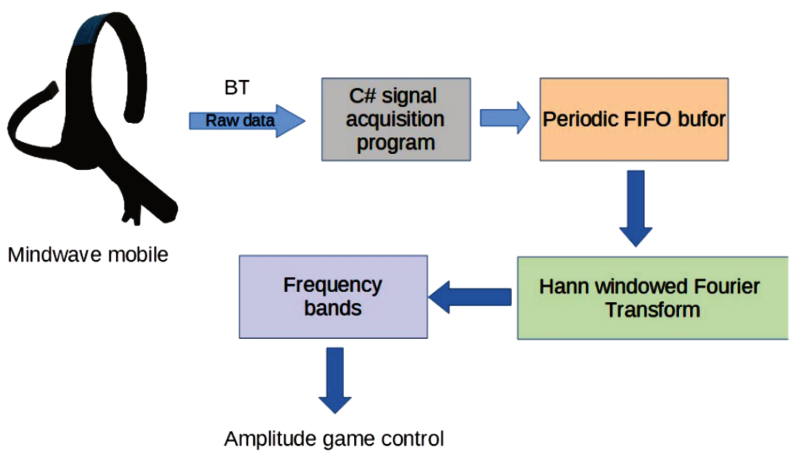

Figure 1. Flowchart of the EEG signal processing system

3. Signal processing and control

Before the raw signal from the EEG device can be used to control a computer game, it has to be processed. The raw signal from the EEG device is a time-dependent function. The time-dependent signal is shown in Figure 2. It is hard to draw features that would allow distinguishing two different time-dependent items of data from the EEG data. For example, two phase-shifted waveforms represent the same signal in the frequency domain. The transformation from the time domain $f(t)$ to the frequency domain $I(\nu)$ most often used in the analysis of signals is the Fourier transform.

$$I(\nu) = \int_{-\infty}^{+\infty} f(t)\, e^{-i2\pi\nu t} dt \tag{1}$$

As a result of such a transformation, the dependence of the amplitude on frequency is obtained. The results of the transform are in the domain of complex numbers. We have a real part of a transform and an imaginary part. What was used in this work was the Fourier transform module, also known as the signal power spectrum. We used the Fast Fourier Transform (FFT) algorithm [19] in this work. The amplitude of the power spectrum was used to control the player in the frequency limit $\nu \in [a,b]$. For this, we will calculate the integral

$$I_{ab} = \int_{a}^{b} I(\nu) d\nu \tag{2}$$

within these limits for the function $I(\nu)$. The value of I_{ab} will be an indicator of the frequency activation in the range of $[a,b]$. In this solution, the signal from the EEG device flows into a FIFO buffer. Data periodically moves inside a buffer of a chosen size. We can imagine this buffer as a window through which a time-dependent signal moves. Now, the Fourier transform is only the data lying in this FIFO window.

This approach has the advantage of being able to analyze changes in signals that are shorter than one second. For example, a buffer consisting of 512 elements corresponds to one-second sampling in this device. We can refresh its content with a maximum frequency of 512 Hz, which corresponds to a rounded refresh every 0.002 s = 2 ms. The human reaction time to external visual stimuli is 190 ms on average [20, 21]. With such frequent refreshing, the changes in the signal would be small. The amplitude in the given frequency band would change quite gradually. At the same time, the number of Fourier transform calculations would be quite large. It would put a heavy load on the processor and slow down the gameplay experience. The refresh rate that reconciles both the response time and the calculation intensity lies between 62.5 and 125 ms, which corresponds to the FIFO window refresh 16 and 8 times per second. We used these refresh rates of amplitude to control the object moving along the X-axis on the monitor screen. The calibration process matches the amplitude value to the window width. The process is to store the lowest amplitude and then subtract that value from the highest amplitude. The maximum difference in amplitudes should be the same as the game window width. In practice, this is hard to achieve, and usually, we have higher amplitude difference values. The quotient of the amplitude difference and the window width w give us the scaling factor $S_f = \left(I_{ab}^{high} - I_{ab}^{low}\right)/w$, used

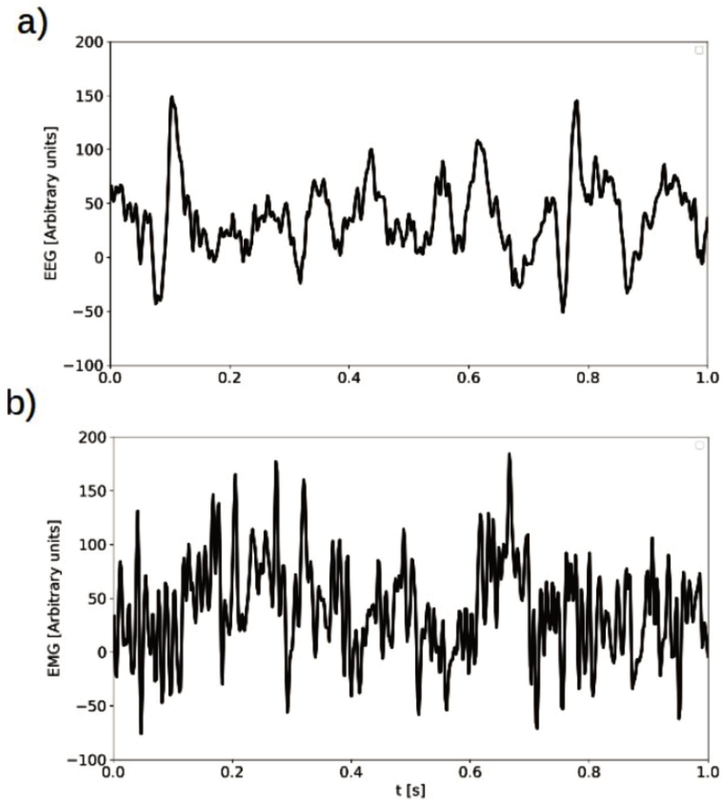

Figure 2. One-second time relationship for a) EEG and b) EEG + EMG face muscle disturbance signal

to scale the current amplitude $I_{scaled} = I_{ab} \cdot S_f$. This scaled amplitude was used to control the game described in the next chapter.

4. Application

An EEG device alone is not sufficient to control a computer game. Software is also needed to convert the EEG signals into the horizontal displacements of the player.

The software presented here is based on a simple game similar to games such as Pong and Arkanoid. This game consists of bouncing the ball off the walls in two dimensions with a movable pong racket moving horizontally along the X-axis (Fig. 3). The rule of the game is to hit the ball with the pong racket. The user controls the game by the measured disturbance of the EEG signal generated by the face or tongue muscles. For this purpose, the signal processing described in the previous section is performed. In Figure 4, we can see the pattern of the FIFO queue. The head pointer points the position in the buffer from where the data is read. New EEG readings are added to the buffer by the tail pointer. The numbers shown in this figure are the sample data. In the first phase (Fig. 4a), the incoming

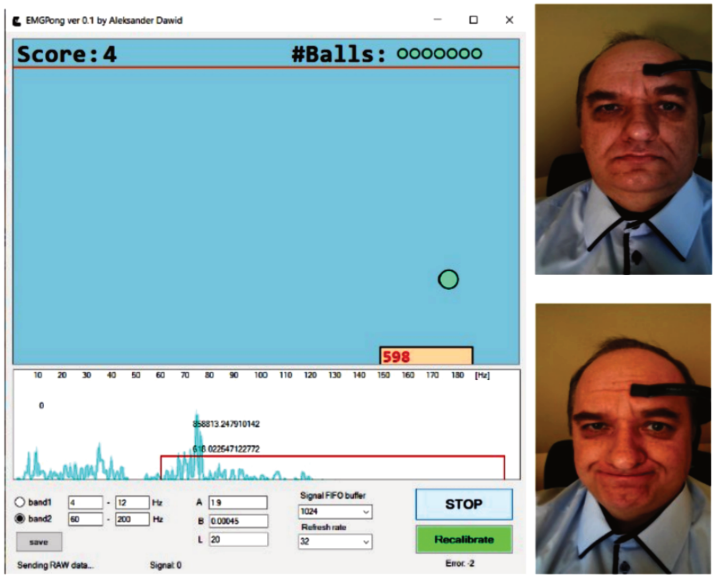

Figure 3. Pong game controlled by measured EEG signal disturbance

EEG data fills the entire buffer of size N. After the buffer is full, the incoming data is recorded from the beginning of the queue, deleting the old data (Fig. 4b). The size of the buffer can be changed in the program startup parameters. In this article, integer size buffers equal to 512, 1024, and 2048 elements are analyzed, which, in turn, correspond to a one, two, and four second EEG signal, respectively. The raw signal from the device is received, the data is processed, and the object on the screen is controlled in separate threads. The libStreamSDK library for the .NET framework is responsible for the communication between the program and the EEG device. It allows connection with the ThinkGear driver provided by the device manufacturer. This driver deals with the BT communication and data streaming. In our program, we have to handle the connection, disconnection, and data transfer events. These events come from the EEG device. Such a design pattern, called MVC, allows separating the data from its processing and presentation.

The MathNet.Numerics library, version 4.7.0, created by Rüegg et al. to calculate the Fourier transform was used [22]. And for the phase invariance, we tried the symmetrical filter invented by the Austrian meteorologist von Hann [23]. First, we have to multiply the raw signal by the *Hann* function.

$$SignalToTransform(i) = Signal(i) \cdot Hann(i) \tag{3}$$

where the *Hann* function has the following form

$$Hann(i) = 0.5 \cdot \left(1 - \cos\left(\frac{2\pi i}{N}\right)\right) \tag{4}$$

The variable i enumerates successive elements in the FIFO buffer, and the variable N is the buffer size. The *SignalToTransform* buffer has a complex type. It contains

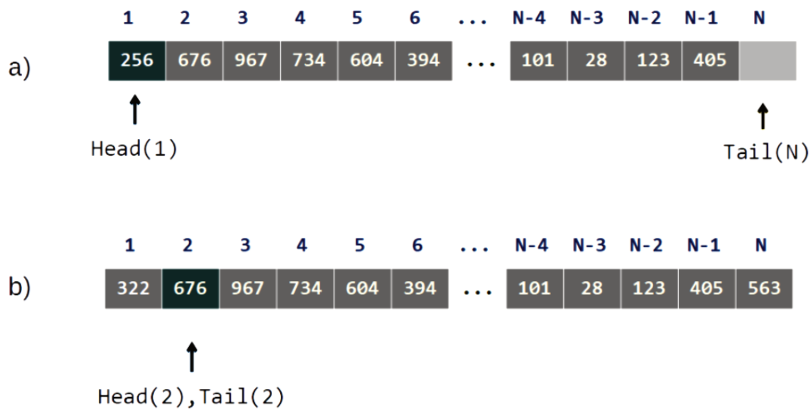

Figure 4. FIFO buffer of EEG data analysis

the real and imaginary parts of the Fourier transform. To deal with it, we calculated the power spectrum using the following equation.

$$I(\nu) = \int_{-\infty}^{+\infty} |f(t)|^2 dt \tag{5}$$

The program allows the user to observe the EEG signal power spectrum change in the graphical form (Fig. 3, the window below the game). Such a visual inspection allows better adjustment of individual parameters defining the range of the analyzed frequency. The game itself runs in a separate thread. The most important part of the game code is the connection between the processed EEG signal and the moving object. We did this by calculating the integral within the given limits. The numerical procedure of integration uses the so-called trapezoidal algorithm [24]. We know that more accurate results could be received if Simpson's numerical method were used [25]. Nevertheless, in this particular application, we care more about the speed of computation than the increased accuracy. The controller calibration process starts when the "calibrate" button is pressed. At the moment of pressing this button, the integrated minimum amplitude is saved. Pressing this button again saves the maximum amplitudes from the given frequency range and calculates the difference between these two amplitudes. Next, we have to scale this value to the width of the game window. We must go through this process before we can start the game.

5. Control experiment and discussion

This experiment demonstrates how conscious the control over an object on the computer screen is. We have compared the game result when the user does not look at the screen to the game results in which the user consciously looks at the screen and tries to force the signal from the EEG device to change in a given frequency range. The output of the experiment is the number of points scored in the game. One point is scored each time when the ball hits the opposite wall. The software allows us to set any frequency range on the power spectrum of the recorded EEG signal. The EEG signal recording equipment can test signals with an oscillation frequency higher than 60 Hz. These signals are treated as noises caused by the muscles of the face and the neck. The pong racket can be controlled by tightening the muscles of the left and right cheeks if we want the racket to go right. The racket returns left if the muscles are relaxed. In this way, the amplitude of the power spectrum is changed in the frequency band higher than 60 Hz. In our first experiment, we choose the frequency band between 60 and 200 Hz. In Figure 5, we can see the cumulative distribution function (CDF) of different game results. One cumulative distribution function consists of 50 Pong games. The authors of this paper were the only participants in the experiment due to the pandemic. To compare the results, the gameplay was recorded with closed eyes and labeled as 'no control' in Figure 5. The highest scores were obtained for a buffer of the 512 and 1024 integer sizes, except that for the buffer size 512, strong oscillating movements of the played object could be observed. On average, the best results were achieved for the refresh rate equal to 62.5 ms. At a given refresh rate, the control over the moving object was very smooth.

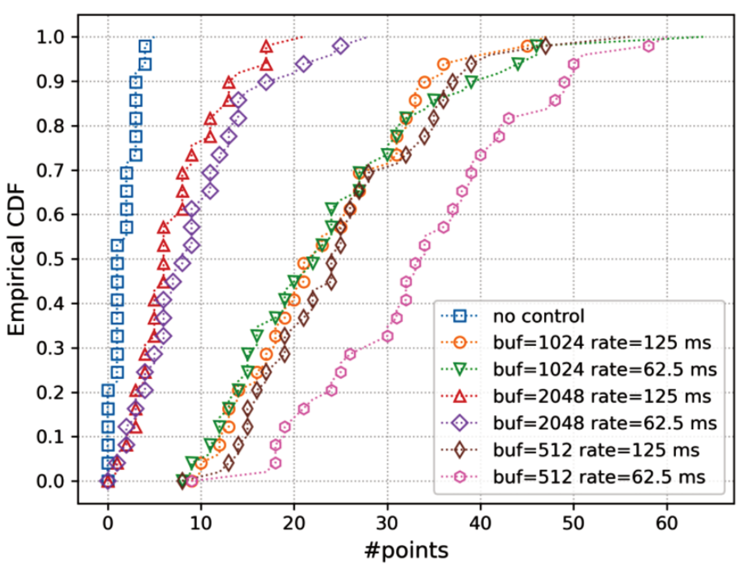

Figure 5. Cumulative distribution of game results for different buffer sizes and refresh rates. The range of used measured disturbance of EEG frequencies is 60-200 Hz

The next frequency range analyzed was 4-12 Hz. In this range, the amplitude was controlled by employing the oscillating movements of the tongue. In our experiment the racket went right when the oscillations speeded up, and left when the oscillations slowed down. This movement produced a very pronounced signal and could be used to adjust the EEG amplitude. Figure 6 shows the performance of a Pong game for different buffer settings and refresh rates. We can see that the results are the worst for a large buffer of 2048 integers. The delay was so huge that it did not allow reacting quickly to the changing game conditions. The best results were received with buffers 512 and 1024, but with a refresh of 62.5 ms. Here, the refreshing was critical to control the game. The maximum score achieved here was half as good as that of the first range of 60-200 Hz.

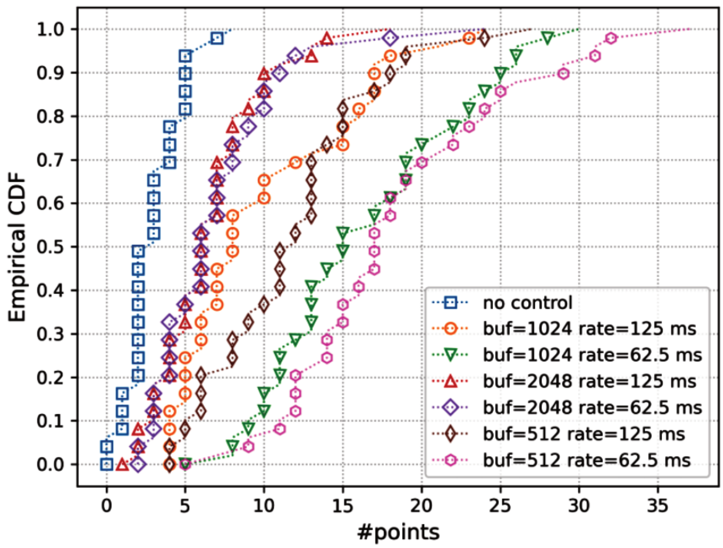

Figure 6. Cumulative distribution of game results for different buffer sizes and refresh rates. The range of used EEG frequencies is 4-12 Hz

6. Conclusions

In conclusion, creating a universal algorithm that would allow controlling a dynamic computer game by an EEG device without training is a very complex task. The method presented here can serve as an example of dynamic game control using face and tongue muscles as the measured disturbance of the EEG signal. Our results showed that direct mapping of the amplitude to the cursor movement could give reasonable control. It should be noted that a better score for the facial muscles is associated with a much wider frequency band and the absence of other interference. This method still needs future improvements to produce smooth and repeatable control of a computer game.

Acknowledgment

The authors would like to thank the WSB University for the funds for supporting this work.

References

[1] Berger H 1929 *Über das Elektrenkephalogramm des Menschen*, Arch. Für Psychiatr. Nervenkrankh **87** 527-570 doi: https://doi.org/10.1007/BF01797193

[2] Vidal J J 1977 *Real-time detection of brain events in EEG*, Proc. IEEE. **65** 633–641 doi: https://doi.org/10.1109/PROC.1977.10542

[3] Liu N H, Chiang C Y and Chu H C 2013 *Recognizing the Degree of Human Attention Using EEG Signals from Mobile Sensors*, Sensors **13** 10273–10286 doi: https://doi.org/10.3390/s130810273

[4] Durka P J, Kuś R, Żygierewicz J, Michalska M, Milanowski P, Łabęcki M, Spustek T, Laszuk D, Duszyk A and Kruszyński M 2012 *User-centered design of brain-computer interfaces: OpenBCI.pl and BCI Appliance*, Bull. Pol. Acad. Sci. Tech. Sci. **60** 427–431 doi: https://doi.org/10.2478/v10175-012-0054-1

[5] Mohammadi G, Shoushtari P, Molaee Ardekani B and Shamsollahi M B 2006 *Person Identification by Using AR Model for EEG Signals*, Proceeding World Acad. Sci. Eng. Technol. **11** 281–285

[6] Poulos M, Rangoussi M, Alexandris N and Evangelou A 2002 *Person identification from the EEG using nonlinear signal classification*, Methods Inf. Med. **41** 64–75

[7] Dawid A 2019 *PSR-based research of feature extraction from one-second EEG signals: a neural network study*, SN Appl. Sci. **1** 1536, doi: https://doi.org/10.1007/s42452-019-1579-9

[8] Alakus T B, Gonen M and Turkoglu I *Database for an emotion recognition system based on EEG signals and various computer games – GAMEEMO*, Biomed. Signal Process. Control. **60** 101951 doi: https://doi.org/10.1016/j.bspc.2020.101951

[9] Parafita R, Pires G, Nunes U and Castelo-Branco M 2013 *A spacecraft game controlled with a brain-computer interface using SSVEP with phase tagging*, IEEE 2nd Int. Conf. Serious Games Appl. Health SeGAH 1-6 doi: https://doi.org/10.1109/SeGAH.2013.6665309

[10] Martinez P, Bakardjian H and Cichocki A 2007 *Fully Online Multicommand Brain-Computer Interface with Visual Neurofeedback Using SSVEP Paradigm*, Comput. Intell. Neurosci. e94561 doi: https://doi.org/10.1155/2007/94561

[11] Chumerin N, Manyakov N V, van Vliet M, Robben A, Combaz A and Hulle M M V 2013 *Steady-State Visual Evoked Potential-Based Computer Gaming on a Consumer-Grade EEG Device*, IEEE Trans. Comput. Intell. AI Games **5** 100–110 doi: https://doi.org/10.1109/ TCIAIG.2012.2225623

[12] van Vliet M, Robben A, Chumerin N, Manyakov N V, Combaz A and Hulle M MV 2012 *Designing a brain-computer interface controlled video-game using consumer grade EEG hardware*, ISSNIP Biosignals Biorobotics Conf. Biosignals Robot. Better Safer Living BRC 1–6 doi: https://doi.org/10.1109/BRC.2012.6222186

[13] Finke A, Lenhardt A and Ritter H 2009 *The MindGame: A P300-based brain–computer interface game*, Neural Netw. **22** 1329–1333 doi: https://doi.org/10.1016/j.neunet.2009.07.003

[14] Cabañero L, Hervás R, González I, Fontecha J, Mondéjar T and Bravo J 2020 *Characterisation of mobile-device tasks by their associated cognitive load through EEG data processing*, Future Gener. Comput. Syst. **113** 380–390 doi: https://doi.org/10.1016/j.future.2020.07.013

[15] *Pong Game, (n.d.) (accessed December 24, 2020)* doi: https://www.ponggame.org/

[16] Rostański M, Buchwald P, Mączka K, Kostka P and Nawrat Z 2015 *The development of internetwork channel emulation platform for surgical robot telemanipulation control system (INSeRT)*, Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 1081–1085 doi: https://doi.org/10.15439/2015F239

[17] NeuroSky, EEG Sensors - EEG Headsets | NeuroSky, (n.d.) (accessed September 7, 2018) doi: http://neurosky.com/biosensors/eeg-sensor/biosensors/

[18] Oostenveld R and Praamstra P 2001 *The five percent electrode system for high-resolution EEG and ERP measurements*, Clin. Neurophysiol. **112** 713–719 doi: https://doi.org/10.1016/S1388-2457(00)00527-7

[19] Duhamel P 1990 *Algorithms meeting the lower bounds on the multiplicative complexity of length-2/sup n/ DFTs and their connection with practical algorithms*, IEEE Trans. Acoust. Speech Signal Process. **38** 1504–1511 doi: https://doi.org/10.1109/29.60070

[20] Jain A, Bansal R, Kumar A and Singh K 2015 *A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students*, Int. J. Appl. Basic Med. Res. **5** 124–127 doi: https://doi.org/10.4103/2229-516X.157168

[21] Thompson P D, Colebatch J G, Brown P, Rothwell J C, Day B L, Obeso J A and Marsden C D 1992 *Voluntary stimulus-sensitive jerks and jumps mimicking myoclonus or pathological startle syndromes*, Mov. Disord. Off. J. Mov. Disord. Soc. **7** 257–262 doi: https://doi.org/10.1002/mds.870070312

[22] Math.NET Numerics, (n.d.) (accessed December 27, 2020) doi: https://numerics.mathdotnet.com/

[23] Cattani C and Rushchitsky J 2007 *Wavelet and Wave Analysis as Applied to Materials with Micro or Nanostructure*, WORLD SCIENTIFIC doi: https://doi.org/10.1142/6497

[24] Pang T 2006, An Introduction to Computational Physics 2 edition, Cambridge University Press, Cambridge; New York

[25] Atkinson K 1989, An Introduction to Numerical Analysis, 2nd edition, Wiley, New York

Aleksander Dawid Received his M.Sc and Ph.D. degrees in a computer simulation in molecular physics at the University of Silesia, Poland, in 1995 and 2000, respectively. Worked at the University until 2017. In the same year, joined the computer science department of the WSB University in Dąbrowa Górnicza. Currently, Professor at the WSB University. His research interests include molecular dynamic simulation, molecular physics and chemistry, programming, computational intelligence, parallel processing, machine learning, signal processing, and brain-computer interface. Published over 50 articles in refereed journals in the areas of computational physics, algorithms, and signal processing.

Paweł Buchwald PhD in computer science, database specialties. Scientifically interested in data processing systems, mobile applications and Internet of Things solutions. A research and teaching worker at the WSB Academy in Dąbrowa Górnicza and the Silesian University of Technology in Gliwice. Also professionally involved in software architecture and designing IT systems for industry and production management. Implementer of scientific projects in the areas of ICT security, artificial intelligence, virtual reality and augmented reality.