

Original Research / Artículo original - Tipo 1

Emotiv EPOC BCI with Python on a Raspberry pi

José Salgado Patrón / josesalgadop@usco.edu.co / <http://orcid.org/0000-0003-2382-3774>

Cristian Raúl Barrera Monje / u2005100288@usco.edu.co

Universidad Surcolombiana, Neiva, Colombia

ABSTRACT The hybrid Brain-Computer Interface [BCI] system gives an insight on the development of useful interfaces for users with different backgrounds, from medical applications to video games, where standalone and wearable means accessibility for the user. Systems such as EPOC offers a simple solution for acquiring electroencephalography and electromyography signals with low price and fast setup, compared to high tech medical equipment. From the processing point of view, a computer always offers the main foundation for solving any issue, as the Raspberry Pi [RPI] does, which provides the sufficient computational power for a BCI to be implemented and an open source operating system such as Raspbian. Certainly a wireless communication is a must between the robot and the RPi, where an Xbee module gives a simple bidirectional connection. Python is the principal tool used in the project with multiple libraries for the processing of brain and muscular signals not only for the preparation of them but classification as well, from multithreading functions, feature extraction such as power spectral density and Hjorth parameters, and a support vector machine classifier.

KEYWORDS BCI; EEG; EPOC; Python; Raspberry Pi; support vector machine.

Emotiv EPOC BCI con Python en una Raspberry pi

RESUMEN El sistema de Interfaz Cerebro-Computador [BCI, Brain-Computer Interface] brinda una percepción en el desarrollo de interfaces aplicables para los usuarios con diferentes aproximaciones, desde aplicaciones médicas hasta videojuegos, donde lo autónomo y lo wearable (utilizable en el cuerpo humano) hacen referencia a accesibilidad para los usuarios. Sistemas como los EPOC ofrecen una solución simple para la adquisición de señales de electroencefalografía y electromiografía, a bajo costo y con una rápida configuración, si se comparan con el equipamiento médico de alta tecnología. Desde el punto de vista del procesamiento, un computador siempre ofrecerá la mejor solución para resolver cualquier problema, tal como lo hace la Raspberry Pi [RPI], la cual provee suficiente potencia computacional para que una BCI sea implementada, además de un sistema operativo open source (Raspbian). Una comunicación inalámbrica entre el robot y la RPi es necesaria, un módulo XBee ofrece una comunicación bidireccional simple. Python es la principal herramienta utilizada en este proyecto, con sus múltiples librerías para el procesamiento de señales musculares y cerebrales, se enfoca, tanto en la preparación de ellas, como en su clasificación, desde funciones multi-hilo y extracción de características –como densidad espectral de potencia y parámetros de Hjorth– a clasificadores de máquinas de soporte vectorial.

PALABRAS CLAVE BCI; EEG; EPOC; máquinas de soporte vectorial, Python; Raspberry Pi.

Emotiv EPOC BCI com Python em uma Raspberry Pi

RESUMO O sistema BCI híbrido dá uma visão sobre o desenvolvimento de interfaces úteis para usuários com diferentes formações, desde aplicações médicas até jogos de vídeo, onde autônomo e portátil significam acessibilidade para o usuário. Sistemas como EPOC oferecem uma solução simples para a aquisição de sinais de EEG e EMG com preço baixo e configuração rápida, em comparação com equipamentos médicos de alta tecnologia. Do ponto de vista do processamento, um computador oferece sempre a fonte principal para resolver qualquer problema, tal como o Raspberry Pi [RPI] faz, que fornece o suficiente poder computacional para implementar uma BCI e um sistema operacional de código aberto, como Raspbian. Certamente uma comunicação sem fio é uma obrigação entre o robô e o RPi, onde um módulo XBee permite uma conexão bidireccional simples. Python é a principal ferramenta usada no projeto com múltiplas bibliotecas para o processamento de sinais cerebrais e musculares, não só para a sua preparação, mas também para a sua classificação, a partir de funções multithreading, extração de características, tais como Densidade de Potência Espectral [PSD] e Parâmetros Hjorth, e uma Máquina de Vetores de Suporte [SVM] classificadora.

PALAVRAS-CHAVE BCI; EEG; EPOC; Python; Raspberry Pi; máquina de vetores de suporte.

I. Introduction

Hybrid BCI systems have different types of signals, all put together to offer different degrees of solutions (Lin, Chen, Huang, Ding, & Gao, 2015); these include EEG, EMG and body movement, using relaxation and concentration, winking action and head movement respectively; these types of signals are active, where the user has control over them at any time without depending on an external stimulant.

The purpose of the project is to develop a light installation and portable system where, as the main aspect, the Emotiv EPOC (Emotiv Systems, 2014) gives a straightforward result on the preparation by simply wearing the wireless system with a USB dongle. The EPOC delivers a wide range of magnitudes, from microvolts for the EEG waves to millivolts for the EMG aspects; it has a gyrosopic sensor of two axes especially for the head movement.

The system needs sufficient memory and processing power to handle the features extraction and classification of multiple biological signals, the manipulation of the robot and a graphical interface, where the Raspberry Pi [RPi] (Upton, 2015) performs well enough with a focus on real-time implementation and no hint of any type of delay, which is ideal for a human-machine application.

The chosen operating system is the Raspbian; it offers a faster performance with heavy operations on floating point arithmetic procedures; it works especially with the ARM CPU from the RPi (Upton, 2015).

The programming language selected must be free, light and relatively new, and not only suitable to work with EPOC and RPi, but have a wide range of callable libraries; Python is an open language which offers different aspects for the processing of brain and muscular signals, from pre-processing, extraction of features such as power spectral density and Hjorth parameters, to classification [support vector machines] and execution of commands; it offers multiple functions related to the creation of a BCI.

The multiple end-user applications for the BCI have specific functions, where some have the aim of rehabilitating the patient, and others of providing entertainment with video games or virtual reality by offering the same perspective to the user.

The purpose of the project is to balance the consumption of the processor with the portability of the system;

I. Introducción

Los sistemas híbridos de interfaz cerebro-computadora [BCI, Brain-computer interface] presentan diversos tipos de señales, con el fin de ofrecer diferentes grados de solución (Lin, Chen, Huang, Ding, & Gao, 2015); en este caso, las señales a estudiar son EEG, EMG y movimiento corporal. Esto se hace utilizando relajación y concentración, parpadeo y movimiento de la cabeza, respectivamente. Este tipo de señales son clasificadas como activas, cuando el usuario tiene el control sobre ellas en cualquier instante de tiempo y sin la dependencia de estimulantes externos.

El propósito de este proyecto es ofrecer una instalación ligera y un sistema portable donde su principal aspecto, el Emotiv EPOC (Emotiv Systems, 2014) presente un resultado directo en la preparación, al simplificar el montaje del sistema con un dongle USB. El EPOC entrega un amplio rango de magnitudes –desde los μV para ondas de electroencefalografía [EEG], hasta varios mV para aspectos relacionados con electromiografía [EMG]–. Además, presenta un sensor giroscópico de dos ejes diseñado especialmente para el movimiento de la cabeza.

El sistema requiere suficiente memoria y potencia de procesamiento para manejar las características de extracción y clasificación de múltiples señales biológicas y para la manipulación del robot. Además, se requiere una interfaz gráfica, donde la Raspberry Pi [RPi] (Upton, 2015) se desempeñe lo suficientemente bien y se enfoque en implementaciones en tiempo real. Por ende, se desea que no haya ningún tipo de retraso o latencia, ideal para aplicaciones humano-máquina.

El sistema operativo elegido es Raspbian, el cual ofrece un buen desempeño bajo operaciones basadas en procedimientos aritméticos de punto flotante; además, es compatible con la arquitectura ARM de la CPU de la RPi (Upton, 2015).

El lenguaje de programación seleccionado debe ser libre, liviano y relativamente nuevo; además, no sólo debe ser compatible para el trabajo con EPOC y la RPi, sino que también debe proveer un amplio rango de librerías utilizables. Python es un lenguaje abierto que ofrece diferentes aspectos para el procesamiento de señales musculares y cerebrales, desde el pre-procesamiento hasta la extracción de características como la densidad espectral de potencia y los parámetros de Hjorth, hasta la clasificación a través de máquinas de soporte vectorial [SVM, Support Vector Machines] y la ejecución de comandos. Además, ofrece funciones relacionadas con la creación de una BCI.

Múltiples aplicaciones de las BCI para el usuario final presentan una función específica, donde algunas buscan rehabilitar al paciente, mientras que otras entretenerlo con videojuegos o realidad virtual, ofreciéndoles la misma perspectiva al usuario.

El propósito de este proyecto es lograr un balance en la utilización del procesador con la portabilidad del sistema. Con base en dichos aspectos, todos los segmentos del procesamiento se mantuvieron al mínimo. Se utilizó un brazo robótico como aplicación final con el fin de llevar a cabo una tarea específica con la combinación de comandos BCI.

II. Metodología

BCI

La Figura 1 presenta el diagrama del proyecto, donde el usuario utiliza el EPOC y tres señales son extraídas de él (EEG, EMG y giroscópica). La principal aplicación ocurre offline, donde se realizan las pruebas, se toman las muestras y se almacenan los respectivos clasificadores. El proceso en tiempo real, básicamente utiliza algunos de los aspectos offline, pero ajustados hacia un trabajo a corto plazo y cíclico, proporcionando al usuario la habilidad de controlar el robot y visualizarlo en una interfaz gráfica.

Muchas BCI están enfocadas en aspectos visuales del cerebro utilizando electroencefalografía, tal como SSVEP (Guneyisu & Akin, 2013) y P300 (Tahmasebzadeh, Bahrani, & Setarehdan, 2013). Por otra parte, en otros trabajos se presentan estímulos visuales sin la necesidad de una pantalla y utilizando aspectos como imágenes motoras (Yao, Meng, Zhang, Sheng, & Zhu, 2013) y estados mentales (Grude, Freeland, Yang, & Ma, 2013). El concepto de híbrido es para adaptar diferentes aspectos en el mismo enfoque, mientras que la electromiografía es para proveer un mayor rango a las funciones de las BCI, como expresiones faciales (Sinyukov, Li, Otero, Gao, & Padir, 2014). En este caso, se hace referencia puntual a los estados mentales con funciones de relajación y concentración y a la función de parpadear desde la expresión facial. El Emotiv EPOC ofrece la posibilidad de usar sensores giroscópicos para ser adaptados a comandos del movimiento de la cabeza (Rechy-Ramirez, Hu, & McDonald-Maier, 2012), de donde únicamente se consideró el movimiento horizontal.

based on these aspects, all the segments of the processing were kept to a minimum. A robotic arm was used as the end-application, using it to perform a specific task with the combination of the BCI commands.

II. Methodology

BCI

Figure 1 shows the diagram of the project, where the user wears the EPOC and the three respective signals are extracted (EEG, EMG, gyroscope). The main application occurs offline from doing the trials, taking the samples and saving the respective classifiers. The process in real time basically uses some of the offline aspects but fitted to work in a shorter time and cyclically, delivering to the user the ability to control the robot and visualizing it on a graphical interface.

Many BCIs are focused on the visual aspect of the brain using electroencephalography (EEG), such as SSVEP (Guneyisu & Akin, 2013) and P300 (Tahmasebzadeh, Bahrani, & Setarehdan, 2013), while others without the need for a screen to present a visual stimulus, use aspects like the motor imagery (Yao, Meng, Zhang, Sheng, & Zhu, 2013) and mental states (Grude, Freeland, Yang, & Ma, 2013). The concept of a hybrid is to adapt different aspects within the same scope, some involving electromyography [EMG] to give a greater range of functions to the BCI, such as facial expressions (Sinyukov, Li, Otero, Gao, & Padir, 2014). In this case, the mental states include relaxation and concentration functions, and a winking function from the facial expression. The Emotiv EPOC offers the possibility of using gyroscope sensors to be adapted for head movement commands (Rechy-Ramirez, Hu, & McDonald-Maier, 2012), in which only the horizontal one is used.

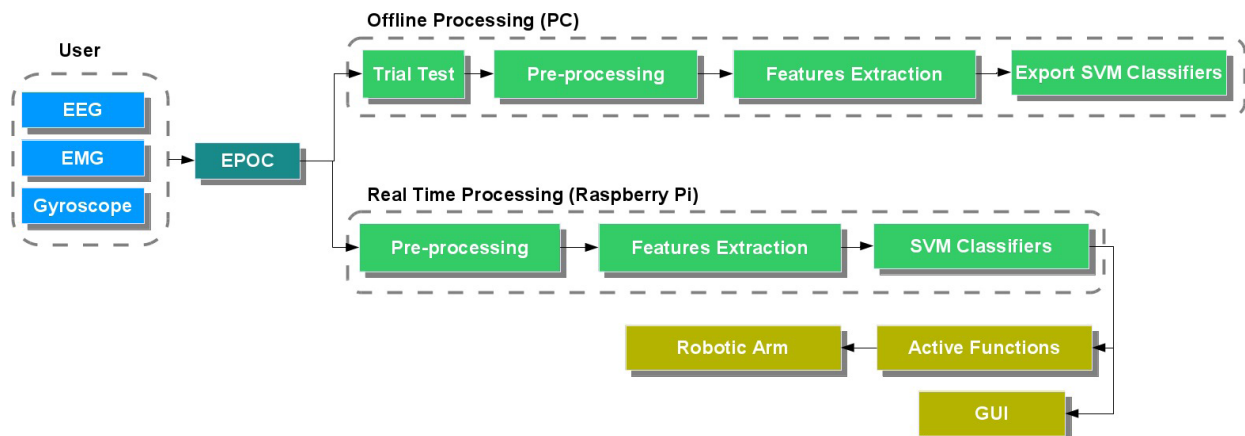


Figure 1. Diagram of the BCI / Diagrama de la BCI

Connectivity / <i>Conectividad</i>	Casco EEG Banda de 2.4 GHz
Channel names / <i>Nombres de los canales</i>	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T7, FC6, F4, F8, AF4
Resolution / <i>Resolución</i>	14 bits con 1 LSB = 0.51 μ V
Bandwidth / <i>Ancho de banda</i>	0.2 – 45 Hz

Table 1. Specifications of Emotiv EPOC / Especificaciones del Emotiv EPOC (Emotiv, 2011)

Some aspects of the Emotiv EPOC highlight the need for equipment that offers the necessary tools for a BCI to work. It possesses the channels needed to obtain the action used in the project, with a resolution allowing for brain and muscular waves to be measured correctly, as can be seen in **TABLE 1**.

For the relaxation, the test consists of closing the eyes and relaxing for a short period of 8 seconds, showing a strong alpha wave (8–12 Hz) on channel O2 of the EPOC, this being one of the most visible brain waves, according to (Liu, Chiang, & Chu, 2013).

The concentration task requires that the user performs a continuous subtraction of three from a random number of higher value, for example, 300, 297, 294, 291 and so on, for a longer period of 100 seconds, in order to keep the user concentrated; channel F8 is used, showing strong beta and gamma values during the test, as analyzed by (Wang & Sourina, 2013).

The winking action, which is basically alternating left and right winking for two seconds each, is related to the EMG aspect. It is mostly recorded as noise, but in this case is taken with its high value of magnitude and lower frequency as an active action, as in (Sinyukov et al., 2014) as a type of human–machine interface, by recording the electrodes closer to the eyebrows. In this case channel F8 is shared between the EMG and EEG action. The winking action offers a different aspect as well as the head movements in which the gyroscopic sensor of the horizontal axis is used; in this case, the sensor's deflection provides straightforward information, so there is no need for extra processing.

TABLE 2 shows the different aspects of the actions used for the hybrid BCI, specifying the typical frequencies and magnitudes of some of the brain waves, as depicted before by Liu et al. (2013) and Wang and Sourina (2013).

Algunos de los aspectos del Emotiv EPOC resaltan la necesidad de equipamiento que ofrezca las herramientas necesarias para que la BCI opere. Éste posee los canales necesarios para obtener la acción utilizada en el proyecto con una resolución adecuada para la medición de las ondas musculares y cerebrales, como puede verse en la Tabla 1.

Para la relajación, el test consiste en cerrar los ojos y relajarse por un corto periodo de tiempo –ocho segundos–, mostrando una significativa onda alfa (8-12 Hz) en el canal O2 del EPOC. Esta onda es una de las más visibles del subconjunto de ondas cerebrales, tal como describen Liu, Chiang, y Chu (2013).

La concentración requiere que el usuario realice una resta entre el dígito 3 y un valor más alto. Por ejemplo, 3-300, 3-297, 3-294, 3-291..., y así sucesivamente, durante un largo periodo de tiempo –cine segundos–, con el fin de mantener al usuario concentrado. El canal F8 se utiliza con beta y gamma mostrando un valor elevado durante el test, como se analiza en el trabajo de Wang y Sourina (2013).

La acción de parpadear, la cual es básicamente alternar el pestañeo izquierdo y derecho cada dos segundos, está relacionada con el aspecto de la EMG y es grabada principalmente como ruido, pero en este caso se toma con su valor más alto de magnitud y más bajo de frecuencia, como una acción activa, tal como muestran Sinyukov et al., (2014). Estos autores presentan dicha acción como un tipo de interfaz humano-máquina al ubicar los electrodos cerca de las cejas. En este caso, el canal F8 es compartido entre la EMG y la EEG. La acción de parpadear ofrece un aspecto diferente, tal como los movimientos de la cabeza, donde el sensor giroscópico del eje horizontal es adaptado. En este caso, la deflexión del sensor provee información confiable, por ende, no hay necesidad de procesamiento adicional.

En la **TABLA 2** se aprecian los diferentes aspectos de las acciones utilizadas por la BCI híbrida, especificando las frecuencias típicas y magnitudes de algunas ondas cerebrales, como las descritas por Liu et al., (2013) y Wang y Sourina (2013).

Las **FIGURAS 2A, 2B Y 2C** muestran el espectrograma de la relajación, concentración y pruebas de parpadeo, donde cada uno presenta una duración en el cual los segmentos

Action/ <i>Acción</i>	Frequency range/ <i>Rango de frecuencias</i>	Typical magnitude/ <i>Magnitudes típicas</i>	EEG channel/ <i>Canal EEG</i>
Relaxation/ <i>Relajación</i>	Alfa (8-12 Hz)	30–50 μ V	O2
	Beta (14-30 Hz)	5-20 μ V	
Concentration/ <i>Concentración</i>	Gamma (31-50 Hz)	5-10 μ V	F8
Ancho de banda/ <i>Bandwidth</i>	0-10 Hz	0-10m V	AF3 (izq.) F8 (der.)

Table 2. Description of the actions used / Descripción de las acciones utilizadas

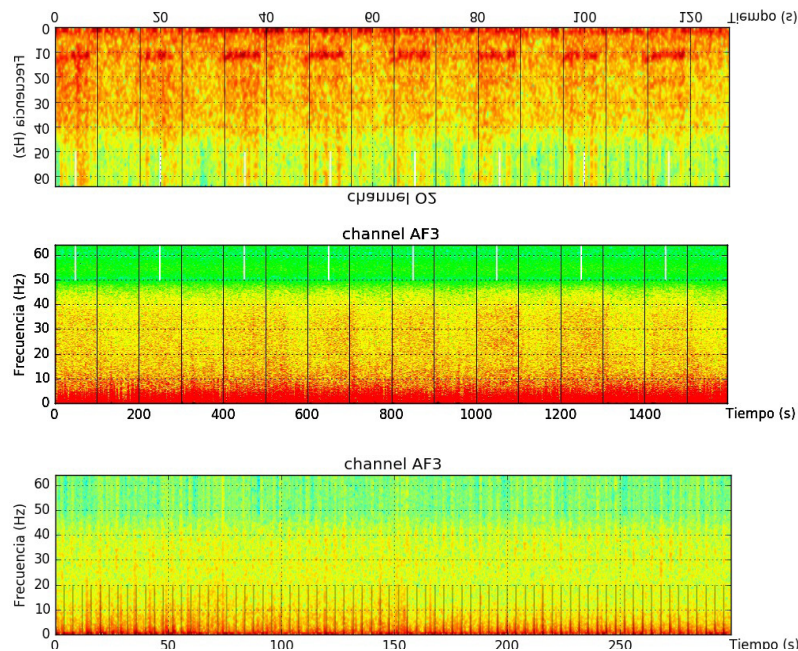


Figure 2. Spectrograms of the actions / Espectrograma de las acciones

activos para la relajación son resaltados por los segmentos rojos en la frecuencia de 10-12 Hz. Para la concentración, los puntos rojos dispersados en cada segmento del rango de frecuencias de 10-40 Hz representan la parte activa. La acción de parpadeo está alrededor del rango de 0-10 Hz, el cual sería considerado generalmente como ruido para la relajación y concentración, pero como se dijo, en este caso se toma como muestras activas. El eje vertical representa la frecuencia en Hz, mientras que el eje horizontal en las figuras representa el tiempo para cada prueba en segundos, donde la concentración es la más duradera de todas, debido a la dificultad de detectar y visualizar dicha característica en cortos periodos de tiempo.

Pre-procesamiento

El uso de las librerías Numpy (Van-Der-Walt, Colbert, & Varoquaux, 2011) es esencial para la mayoría del procesamiento básico. Con el fin de preparar las muestras tomadas para la correspondiente extracción, debe aplicarse un pre-procesamiento. Como primera instancia, las pruebas son convertidas de decimal a voltaje para una mejor representación de los datos, restando el valor promedio de la señal y multiplicándolo por el LSB, el cual, para este caso es de 0.51 μV .

El filtrado se realiza a la prueba entera, conociendo que el EPOC presenta un filtro interno pasa banda de 0.2-45 Hz. Por ende, con el fin de remover el ruido remanente, un filtro paso alto Butterworth de quinto orden en 2 Hz es utilizado junto con el módulo `filtfilt` (SciPy, 2016). Las muestras activas son divididas en épocas con tamaños de ventana más pequeños. Esto se hace utilizando la función `split` de Numpy. Para las tres acciones, el tamaño de la época

FIGURES 2A, 2B AND 2C show the spectrogram of the relaxation, concentration and winking trials. Each trial is for a different length of time, where the active segments for relaxation are highlighted as the red segments on the 10–12 Hz frequency range; for concentration, the red dots dispersed on each segment in the frequency range of 10–40 Hz represent the active part; the winking action is around the 0–10 Hz frequency range, which would usually be considered noise for relaxation and concentration, but in this case is taken as active samples. The vertical axis is the frequency in Hz; the horizontal axes represent the time for each trial in seconds, where the concentration is the longest of all, due to the difficulty of

detecting and visualising concentration over shorter periods of time.

Pre-processing

The use of Numpy libraries (van der Walt, Colbert, & Varoquaux, 2011) is essential for most of the basic processing. In order to prepare the samples taken for the subsequent features extraction, pre-processing is necessary. First of all, the trials are converted from decimal to voltage for a better representation of the data, subtracting the average signal value and multiplying it by the LSB, which in this case is 0.51 μV .

Filtering is done for the whole trial as well, knowing that the EPOC has an internal band pass filter of 0.2–45 Hz. Then, in order to remove some of the remaining noise, a high pass filter [HPF] at 2 Hz, Butterworth 5th order type, is applied, using the `filtfilt` module (SciPy, 2016). The active samples are divided into epochs of small window sizes; this is done by using the `split` function from Numpy. For the three actions, the size of the active epoch is 4 seconds. In the case of the relaxation and concentration, both extremes of the samples (the beginning and end of the trial) are deleted to avoid possible errors from the user.

Scikit-learn (Pedregosa et al., 2011) offers a wide range of libraries related to the processing of biological signals. In this case, the SVM is used, so it is necessary to stan-

standardize the data to avoid misclassification of samples. Using the 'preprocessing.scale' library from scikit-learn, the samples are centered along the axis to the mean and scaled to unit variance.

Features extraction

For the segment of extracting the parameters of the samples, different functions based on Python PyEEG (Bao, Liu, & Zhang, 2011) are used with simple scalar outputs. In the case of the relaxation, Hjorth Mobility–Hjorth Parameters– (Oh, Lee, & Kim, 2014) and the power spectral density ratio [PSD] with the Welch method (Kaysa & Widyotriatmo, 2013) are used, where the ratio indicates the relationship between the alpha's frequency band and the total sample band.

The concentration has three extracting functions, due to the greater difficulty in detecting it owing to the low magnitude on the beta and gamma waves: Hjorth Mobility, Petrosian fractal dimension [PFD] (Goh, Hamadicharef, Henderson, & Ifeachor, 2005) and PSD ratio. These functions have linear, quasi-linear and non-linear characteristics that are useful for brain signals. In the case of winking, from Numpy, the Frobenius Norm is used with the Hjorth Complexity.

Classifiers

The main library for the classifier is the scikit-learn, using the C-support vector classification 'sklearn.svm.SVC' function (scikit-learn, 2014a), based on the library libsvm, using the one vs. one system and kernel function 'RBF' with C and gamma variables to adjust.

The one vs. one system basically puts into one class the active samples and into the other class the non-active samples, such as artefacts, noise and data opposite to the active parameter; this is also known as binary classification, where it simply identifies if the sample is active or not. From the active data, a certain percentage is separated for a later test in order to prove the classifier. A specific standardization is applied due to the necessity for the samples to be within the same unit variance and Gaussian with zero mean; the library used was the 'StandardScaler' (scikit-learn, 2014b) and it is saved to be applied in real time. The chosen RBF parameters were the same for the three actions: $C = 100$ and $\text{gamma} = 0.1$.

The classifier is fitted with the two classes, knowing which one is active and which is not, also called supervised learning on labeled training data. The 'joblib.dump'

ca activa es de cuatro segundos. En el caso de la relajación y la concentración, ambos extremos de las muestras (inicio y final de las pruebas) son suprimidas para evitar posibles errores para el usuario.

El paquete scikit-Learn (Pedregosa et al., 2011) ofrece un amplio rango de librerías relacionadas con el procesamiento de señales biológicas. En este caso se utilizan SVM; por ende, es necesario estandarizar los datos, evitando clasificaciones incorrectas de las muestras. Utilizando la librería preprocessing.scale de Scikit Learn, las muestras son centradas junto con el eje hacia la media y escaladas hacia la varianza unitaria.

Extracción de características

Para el segmento de extraer los parámetros de las muestras se utilizaron diferentes funciones basadas en Python PyEEG (Bao, Liu, & Zhang, 2011) con salidas escalares simples. Se utilizó, para el caso de la relajación, la movilidad Hjorth –junto con sus correspondientes parámetros– (Oh, Lee, & Kim, 2014), y para la densidad espectral de potencia, el método Welch (Kaysa & Widyotriatmo, 2013). Aquí, el radio indica la relación entre la banda de frecuencias de alfa y el total de la banda en la prueba.

La concentración presenta tres funciones de extracción, puesto que ostenta una mayor dificultad en la detección por manifestar una baja magnitud en las ondas beta y gamma. La movilidad Hjorth, la dimensión fractal Petrosiana (Goh, Hamadicharef, Henderson, & Ifeachor, 2005) y el radio de ésta son las tres funciones de extracción. Estas funciones presentan características lineales, cuasi-lineales y no lineales, respectivamente, y son útiles para señales cerebrales. En el caso del parpadeo, la norma de Frobenius se utiliza con la complejidad de Hjorth desde Numpy.

Clasificadores

La librería principal para el clasificador es la Scikit Learn, utilizando la clasificación de vectores support sklearn.svm.SVC en lenguaje C (Scikit Learn, 2014a). Esta librería está basada en la denominada libsim, pero se diferencia en que utiliza el sistema uno versus todos y la función de base radial [RBF, Radial Basis Function] del kernel con C y variables gama a ajustar.

El sistema uno versus uno básicamente pone en una clase las muestras activas, mientras que en la otra pone las no activas –como artefactos, ruido y datos contrarios al parámetro activo–. Es conocida también como clasificación binaria, donde simplemente se identifica si la muestra es activa o no. Para los datos activos se separa un porcentaje dado para un test posterior con el fin de probar el clasificador. Una estandarización específica se aplica debido a la necesidad de que las muestras estén sin la misma varianza unitaria y gaussiana con media cero. La librería utilizada fue la StandardScaler (Scikit Learn, 2014b), la que fue almacenada para ser aplicada en tiempo real. Los parámetros RBF elegidos fueron los mismos para las tres acciones, en los cuales.

El clasificador fue ajustado con las dos clases conociendo cuál es la activa y cuál la no activa. Esto se denomina aprendizaje supervisado. El `joblib.dump` (Joblib, 2009) ayuda a almacenar el clasificador entrenado en un archivo para su uso posterior. Existen cuatro clasificadores SVM, uno para cada acción (relajación, concentración, parpadeo izquierdo y parpadeo derecho) donde se aplica la misma escala estándar. El archivo “SVC.fit” es utilizado como método para clasificar las muestras. En este caso se usa primero para identificar si el clasificador está operando correctamente al ajustarse a las muestras de prueba; de aquí, la salida puede ser 0 o 1 para cada muestra, dependiendo de si pertenece a la clase activa o no activa.

Procesamiento en tiempo real

En este caso, la adaptación de las funciones se restringe al tamaño de la ventana con el fin de evitar cualquier tipo de retraso/latencia y mantener un correcto procesamiento de las señales. Aquí, para dos de tres acciones, el tamaño de la ventana se mantiene en dos segundos, lo suficiente para la acumulación de características de hasta 128 muestras por segundo.

El proceso *offline* es similar, se utilizan buffers para la acumulación de las muestras, obteniendo 128 muestras por segundo durante cuatro segundos. Lo anterior se resume en un total de 512 muestras o un búfer de longitud 512. Las librerías utilizadas fueron `threading.Thread`, `multiprocessing.Process` y `multiprocessing.Pipe`, en ellas, el uso de hilos ofrece la posibilidad de procesar múltiples señales al mismo tiempo, además de compartir el mismo origen, gestionando los datos de los tres canales. De estos canales, uno se comparte entre las acciones de concentración y parpadeo derecho. Los canales ayudan a controlar el envío y recepción de buffers en sincronía y el proceso es exclusivo para lidiar con la extracción de los datos del EPOC y la entrega de éstos a los hilos.

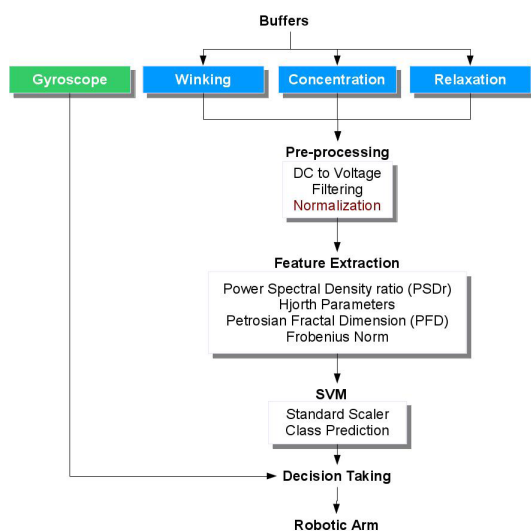


Figure 3. Diagram of the main process /
Diagrama del proceso principal

(joblib, 2009) helps to save the trained classifier into a file for further use. There are four SVM classifiers, one for each action (relaxation, concentration, left winking and right winking), where the same is applied to the standard scaler. ‘SVC.fit’ is used as the method for classifying the samples, in this case using it first to identify if the classifier is working properly by fitting it to the testing samples, where the output would be 0 or 1 for each sample depending on whether it belongs to the active or non-active class.

Real-time processing

In this case, the adaptation of the functions is restricted to the window size, in order to avoid any delay and maintain a correct processing of the signals, where for the three actions, the window length is kept to 2 seconds, enough for the accumulation of features for 128 samples per second.

The process is similar to offline, where buffers are used for the accumulation of the samples, being 128 per second for 4 seconds; this is a total of 512 samples or a buffer of length 512. The ‘`threading.Thread`’, ‘`multiprocessing.Process`’ and ‘`multiprocessing.Pipe`’ (Python Software Foundation, 2016a; 2016b) libraries were used, where Thread offers the possibility to process multiple signals at the same time, sharing the same source, and handling the data from the three channels, one of which is shared between concentration and the right winking action. Pipelines help to control the sending and receiving of the buffers in synchrony. The Process was dedicated to deal with extracting the data from the EPOC and delivering it to the threads.

Each of the actions is processed in a separate file, where, as said before, the concentration and right winking share the same channel. **FIGURE 3** is a diagram of the process of the actions, from the preparation of the buffers to the classification of the extracted features.

The steps followed are similar for all the actions except the muscular action, where the normalization is not applied to avoid removing non-complex information found in EMG signals; a prediction stage is used after the features are extracted, and indicates if the current buffer is active or not. To avoid constant extraction of the same buffer plus one sample, a delay of 64 samples is used, meaning that the buffer waits to add 64 new samples each time they are sent to their respective threads, allowing 0.5 of a second to have meaningful new data.

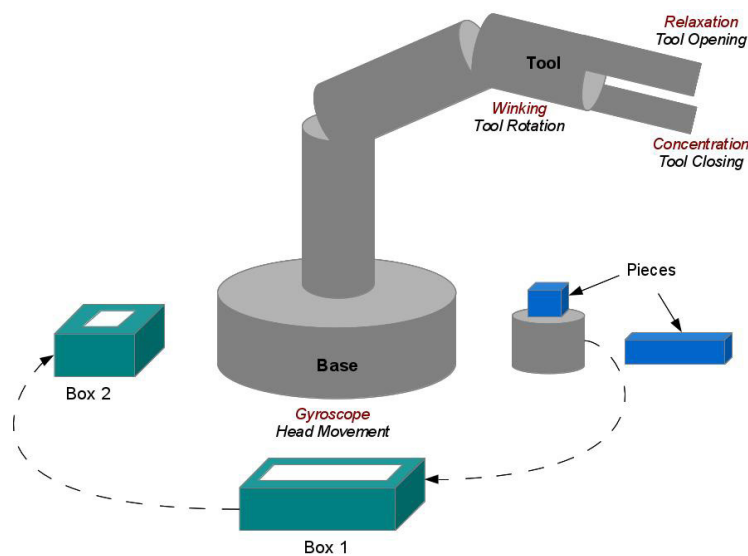


Figure 4. Representative user actions for the robotic arm /
Las acciones representativas del usuario para el brazo robótico

A decision is taken once the prediction indicates if the current sample belongs to class 0 (active) or class 1 (non-active), by an indicator on the GUI and the subsequent movement of the respective robotic arm. Figure 4 indicates the robotic movement with the user's action; for relaxation and concentration the arm opens and closes the tool respectively, for left and right winking the arm rotates the tool and for the head movement, the base rotates to the left or to the right.

Connection between RPi and the robotic arm

The adaptation between the robotic arm and the Raspberry Pi was due to the use of the Xbee modules. In the Python environment the 'serial.Serial' was used in order for the USB Xbee module to work without the need for extra libraries. The microcontroller for the robotic arm was the Maestro Servo Controller, using the Pololu serial protocol, and the serial connection was made at 9600 bauds (Pololu, 2014).

The command for each action represents a single movement action, which means that when relaxation is detected the tool opens at once and not gradually as the head movement does, with the positions depending on where the head stops rotating; this type of ON/OFF control is used for the relaxation, concentration and winking actions, allowing the user to perform more accurately.

Cada una de las acciones son procesadas en archivos separados, donde –como se dijo– la concentración y el parpadeo derecho comparten el mismo canal. La Figura 3 representa el diagrama del proceso de las acciones, desde la preparación de los buffers hasta la clasificación de las características extraídas.

Los pasos siguientes son similares a todas las acciones excepto la muscular, puesto que la normalización no se aplica para evitar la remoción de información no compleja encontrada en las señales EMG. Una etapa de predicción se utiliza después de que las características son extraídas, donde se indica si el búfer actual está activo o no. Para evitar una constante extracción del mismo búfer y muestra, se utiliza una latencia de 64 muestras cada vez que se envían a los respectivos hilos, dando un intervalo de 0.5 segundos para la obtención de datos significativos.

Una vez que la predicción indica si la muestra actual pertenece a la clase 0 (activa) o a la clase 1 (inactiva), se toma una decisión. Esto a través de un indicador en la interfaz gráfica de usuario [GUI, *Graphical User Interface*] y el movimiento relacionado del brazo robótico. La Figura 4 indica el movimiento robótico con la acción del usuario donde, para relajación y concentración, el brazo abre y cierra la herramienta, respectivamente. Por otra parte, para el parpadeo izquierdo y derecho, el brazo rota la herramienta, y para el movimiento de la cabeza, el brazo rota a la izquierda o a la derecha.

Conexión entre la RPi y el brazo robótico

La adaptación entre el brazo robótico y la Raspberry Pi se hizo utilizando módulos Xbee. En el ambiente Python se utilizó el elemento *serial.Serial* con el fin de que el módulo Xbee USB operara sin necesidad de librerías extra. El microcontrolador para el brazo robótico actúa como servo control maestro, utilizando el protocolo serial Pololu (Pololu, 2014). Esta conexión se realizó a 9600 baudios.

Los comandos para cada acción representan un movimiento simple, lo cual indica que cuando se detecta relajación, la herramienta se abre una vez y no gradualmente, tal como el movimiento de la cabeza. Para este último, sus posiciones dependen de dónde se detenga la rotación de la cabeza. Este tipo de control *ON/OFF* se utiliza para las acciones de relajación, concentración y parpadeo, lo que permiten una interacción del usuario más precisa.

Adaptación de la Raspberry Pi

El sistema operativo Raspbian ofrece la posibilidad de que la RPi trabaje con Python sin mayores inconvenientes. La mayoría de los repositorios necesarios pueden instalarse en el paquete Anaconda para RPi. Los restantes pueden instalarse manualmente, dependiendo de las dependencias



Figure 5. Real time graphical interface /
Interfaz gráfica de la operación en tiempo real robótico

en las librerías. Para este caso, se instalaron las librerías necesarias para la interconexión del EPOC con el módulo XBee.

La Raspberry Pi tiene cuatro puertos USB, de los cuales dos fueron utilizados durante el presente trabajo, así: uno para el EPOC, otro para el módulo XBee. El dongle EPOC es detectado por el sistema operativo Raspbian utilizando la llave de archivo Emotiv (ofrecida por el sistema Linux a nivel de EPOC) o la herramienta Emokit. Esta última fue creada como un método alternativo para sistemas Linux. Sea cual sea la decisión de uso, ambas opciones ofrecen los datos *en crudo* del EPOC. Únicamente con importar la librería y con los repositorios en Python necesarios, el sistema podría funcionar correctamente.

Interfaz gráfica en la RPi

La interfaz gráfica para el usuario se desarrolló utilizando el software *Qt designer* (Qt Company, 2016), el cual requiere una conversión final de la extensión de archivo “.ui” a la extensión “.py” para ser utilizado. La interfaz presenta indicadores para la acción cuando éstos son activados para procesamiento en tiempo real. Por otra parte, para el procesamiento *offline*, se utilizó una GUI más sencilla, la cual se construyó con el fin de crear las muestras para los clasificadores y la observación del usuario. En las Figuras 5a, 5b y 6 se pueden observar las dos interfaces, respectivamente.

III. Resultados

Se utilizó un simple experimento para probar el proyecto. En este test, el usuario mueve dos piezas de formas específicas en cajas de diferentes formas. Por ende, la idea principal es probar la exactitud de las acciones al no cometer muchos errores. Las mejoras se incrementaron en la segunda sesión de diez ensayos cada una.

El ajuste se realizó principalmente debido a la acción del giroscopio, puesto que, al no proporcionar un mayor desplazamiento al movimiento de la cabeza, ésta permitía al usuario mantener los ojos en el brazo robótico todo el tiempo.

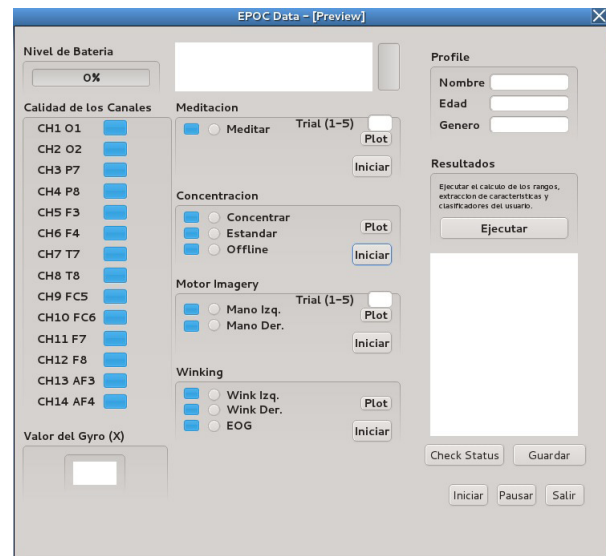


Figure 6. Offline graphical interface /
Interfaz gráfica de la operación offline

Raspberry Pi adaptation

OS Raspbian offers great possibilities for RPi to work with Python. Most of the repositories needed for it can be installed with the Anaconda Package for RPi. Others can be installed manually depending on the dependencies of libraries; in this case, the only libraries needed to work are the ones for interconnecting the EPOC and the Xbee module.

Raspberry Pi has 4 USB ports, of which two are used during the process, one for the EPOC and another for the Xbee. The EPOC dongle is detected by the Raspbian using either the Emotiv key file which is offered for the Linux system at the research EPOC level, or the Emokit, a tool created as a workaround for systems to perform on Linux, which both offer the raw data of the EPOC. By just importing the library and having the necessary Python repositories, the system will work correctly.

Graphical interface on the RPi

The graphical interface for the user was developed using the *Qt designer* (Qt Company, 2016), and simply needed a final conversion from ‘.ui’ to ‘.py’ format to be used. It has indicators for the action when these are activated for the real time processing, and for the offline process a more condensed GUI was used, which was created in order to create the samples for the classifiers and observation of the user. Figures 5a, 5b and 6 show the two interfaces respectively.

III. Results

A simple experiment was performed in order to test the project. In this test, the user had to move two pieces of specific shapes into boxes of different shapes, thus, testing the accuracy of the actions by seeking not to commit any errors. The improvement increased in the second session of 10 trials each.

The adjustment was done mainly to the gyroscopic action, after not initially providing the head movement with a major displacement to allow the user to keep the eyes on the robotic arm all the time.

The routine of the trial is for the user to use the relaxation and concentration to open and close the tool for grabbing the piece; then use the head movement to position it over the respective box of the same shape as the piece; followed by the winking action to rotate the piece to fit correctly into the box.

IV. Conclusions

The adaptation of the project into the Raspberry Pi was possible as a result of the inclusion of the Raspbian operating system and the necessary Python repositories, not just for the processing but also for the EPOC and Xbee connection.

From the EPOC, only three channels were used, as two actions shared one of them, avoiding heavy processing and delays in the real-time processing.

The use of SVM with the same RBF values for all the classifier's actions indicates the adaptability of the classifier for signals, as their features were extracted by the same extractors, where Hjorth Parameters fulfil the function perfectly for EEG and EMG signals.

It is possible to create a wearable end-user BCI with the RPi and EPOC, for not just portability but also wireless manipulation of robotic systems. \square

La rutina de la prueba es para que el usuario utilice la relajación y concentración para abrir y cerrar la herramienta para sostener la pieza. Por consiguiente, el uso del movimiento de la cabeza es útil para posicionarla sobre la respectiva caja de la misma forma que la pieza, seguido de la acción de parpadeo para rotar la pieza y que encaje correctamente en la caja.

IV. Conclusiones

La adaptación de este proyecto a la Raspberry Pi fue posible gracias al resultado de la inclusión del sistema operativo Raspbian junto con sus repositorios en Python necesarios, no solo para el procesamiento, sino también para la conexión entre el EPOC y el módulo Xbee.

Desde el EPOC únicamente se utilizaron tres canales, dos para acciones compartidas entre ellos, evitando procesamientos adicionales y potenciales retardos en la utilización en tiempo real.

El uso de SVM con los valores RBF similares para todas las acciones del clasificador indica la adaptabilidad de éste a señales cuyas características han sido extraídas por los mismos extractores. Por lo anterior, los parámetros Hjorth cumplen con la función adecuadamente para señales EEG y EMG.

Es posible crear un dispositivo BCI usable (*wearable*) con la RPi y utilizando EPOC, no solo para portabilidad, sino también para manipulación inalámbrica de sistemas robóticos. \square

References / Referencias

- Bao, F., Liu, X., Zhang, C. (2011). PyEEG: An Open Source Python Module for EEG / MEG Feature Extraction. *Computational Intelligence and Neuroscience*. 2001(Art. 406391). doi: 10.1155/2011/406391.
- Emotiv (2014). *Emotiv EPOC: Brain Computer Interface & Scientific contextual EEG* [blog]. Retrieved from: <https://emotiv.com/product-specs/Emotiv%20EPOC%20Specifications%202014.pdf>
- Goh, C., Hamadicharef, B., Henderson, G., & Ifeachor, E. (2005). Comparison of fractal dimension algorithms for the computation of EEG biomarkers for dementia. In *2nd International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2005)*, (pp.464-471). Retrieved from: <https://hal.inria.fr/inria-00442374>
- Grude, S., Freeland, M., Yang, C., & Ma, H. (2013). Controlling mobile Spykee robot using Emotiv neuro headset. In *2013 32nd Chinese Control Conference (CCC)*, (pp. 5927-5932). IEEE.
- Guneyusu, A., & Akin, H. (2013). An SSVEP based BCI to control a humanoid robot by using portable EEG device. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (pp. 6905-6908). IEEE. doi: 10.1109/EMBC.2013.6611145.
- Joblib (2009). *Joblib: running Python functions as pipeline jobs* [blog]. Retrieved from: <https://pythonhosted.org/joblib/generated/joblib.dump.html>
- Kaysa, W. A. & Widyotriatmo, A. (2013). Design of Brain-computer interface platform for semi real-time commanding electrical wheelchair simulator movement. In *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*, (pp. 39-44). IEEE. doi: 10.1109/ICA.2013.6734043.
- Lin, K., Chen, X., Huang, X., Ding, Q., & Gao, X. (2015). A Hybrid BCI speller based on the combination of EMG envelopes and SSVEP. *Applied Informatics*, 2(1). doi: 10.1186/s40535-014-0004-0
- Liu, N., Chiang, C., & Chu, H. (2013). *Recognizing the degree of human attention using EEG signals from mobile sensors*. *Sensors*, 13(8). 10273-10286. doi: 10.3390/s130810273.
- Oh, S. H., Lee, Y. R., & Kim, H. N. (2014). A novel EEG feature extraction method using Hjorth parameter. *International Journal of Electronics and Electrical Engineering*, 2(2), 106-110. doi: 10.12720/ijeee.2.2.106-110.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825-2830.
- Pololu Corporation (2014). *Pololu maestro servo controller: user's guide* [on line] Retrieved from: <https://www.pololu.com/docs/pdf/OJ40/maestro.pdf>
- Qt Company. (2016). *Qt Designer Manual* [blog]. Retrieved from: <http://doc.qt.io/qt-4.8/designer-manual.html>
- Rechy-Ramirez, E. J., Hu, H., & McDonald-Maier, K. (2012). Head movements based control of an intelligent wheelchair in an indoor environment. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (pp. 1464-1469). IEEE. doi: 10.1109/ROBIO.2012.6491175.
- Scikit Learn (2014b). *Standardize features by removing the mean and scaling to unit variance* [blog]. Retrieved from: <http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
- Scikit Learn. (2014a). *C-Support vector classification* [blog]. Retrieved from: <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- Sinyukov, D. A., Li, R., Otero, N. W., Gao, R., & Padir, T. (2014). Augmenting a voice and facial expression control of a robotic wheelchair with assistive navigation. In *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, (pp. 1088-1094). IEEE. doi:10.1109/SMC.2014.6974059.
- Tahmasebzadeh, A., Bahrani, M., & Setarehdan, S. K. (2013). Development of a robust method for an online P300 speller brain computer interface. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, (pp. 1070-1075). IEEE. doi: 10.1109/NER.2013.6696122.
- The Python Software Foundation (2016a). 16.2. threading — Higher-level threading interface. In *The Python Standard Library* [blog]. Retrieved from: <https://docs.python.org/2/library/threading.html>
- The Python Software Foundation (2016b). 16.6. multiprocessing — Process-based “threading” interface. In *The Python Standard Library* [blog]. Retrieved from: <https://docs.python.org/2/library/multiprocessing.html>
- Upton, L. (2015). *Benchmarking raspberry Pi 2* [blog]. Retrieved from: <https://www.raspberrypi.org/blog/benchmarking-raspberry-pi-2/>
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2), 22-30. doi:10.1109/MCSE.2011.37.
- Wang, Q., & Sourina, O. (2013). Real-time mental arithmetic task recognition from EEG signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 21(2), 225-232. doi: 10.1109/TNSRE.2012.2236576.
- Yao, L., Meng, J., Zhang, D., Sheng, X., & Zhu, X. (2014). Combining motor imagery with selective sensation toward a hybrid-modality BCI. *IEEE Transactions on Biomedical Engineering*, 61(8), 2304-2312. doi: 10.1109/TBME.2013.2287245.

CURRICULUM VITAE

José de Jesús Salgado Patrón Electronic Engineer; Master in Computing and Electronic Engineering; professor at the Universidad Surcolombiana (Neiva, Colombia): Electronic Engineering Program. His professional interest areas are: biomedical instrumentation, biomedical signal processing, robotics, and computational vision. / Ingeniero Electrónico, Magister en Ingeniería Electrónica y de Computadores, docente de planta del Programa de Ingeniería Electrónica de la Universidad Surcolombiana (Neiva). Sus áreas de interés profesional son: la instrumentación biomédica, el procesamiento de señales biomédicas, la robótica y la visión computacional.

Cristian Raúl Barrera Monje Student of Electronic Engineering at the Universidad Surcolombiana (Neiva, Colombia). His professional interest areas are: biomedical signal processing [EEG], learning machine, embedded systems and brain computer interfaces. / Estudiante de Ingeniería Electrónica de la Universidad Surcolombiana (Neiva). Sus áreas de interés profesional son: el procesamiento de señales biomédicas - EEG, el aprendizaje de máquina, los sistemas embebidos y las interfaces cerebro computadora - BCI.