Congr. Int. en Ing. Electrónica. Mem. ELECTRO, Vol. 39, pp. 77-82, Oct 2017, Chihuahua, Chih. , México
http://electro.itchihuahua.edu.mx/memorias_electro/MemoriaElectro2017.zip
ISSN 1405-2172

DISEÑO DE UN SISTEMA ALTERNATIVO DE COMUNICACIÓN UTILIZANDO ARTEFACTOS DE PARPADEO CODIFICADOS DENTRO DE UNA SEÑAL EEG Y UN SISTEMA MODULAR DE REDES NEURONALES

Carlos Eduardo Cañedo Figueroa[1], Mario Ignacio Chacón Murguía[1], Juan Alberto Ramírez Quintana[1], María del Pilar Gómez Gil[2]

[1]Instituto Tecnológico de Chihuahua, Laboratorio de Percepción Visual con Aplicaciones Robóticas,
Ave. Tecnológico #2909, Chihuahua, Chih. México, CP 31310
[2]Instituto Nacional de Astrofísica, Óptica y Electrónica.
Luis Enrique Erro # 1, Tonantzintla, Puebla, México C.P. 72840
cecanedo, mchacon, jaramirez{@itchihuahua.edu.mx}, pgomez@ccc.inaoep.mx

Resumen

En este trabajo se presenta el diseño de un sistema alternativo de comunicación (SAC) basado en una interfaz cerebro-computadora, mediante análisis de parpadeo para generación de expresiones lingüísticas. El sistema se basa en información de señales de artefacto incluidas en señales EEG al generar parpadeos codificados. Estas señales son procesadas para hacerlas invariantes al usuario, después se realiza un proceso de extracción de características estadísticas y de magnitud, las cuales son utilizadas para diseñar un clasificador modular basado en redes neuronales tipo perceptron. Finalmente el sistema es implementado en un sistema embebido. Los resultados del desempeño del sistema en su diseño y verificación son mayores al 90% de clasificación correcta y fueron comprobados en pruebas experimentales con varios usuarios.

Abstract

This paper presents the design of an alternative communication system based on a brain computer interface by analyzing blinking signals and producing voice expressions. The system is based on artifact signal information included in EEG signals. These signals are processed to make them user independent. Then, statistical and amplitude feature extraction techniques are use on them to determine the feature vectors. These features vectors are employed to design a modular perceptron artificial neural network classifier. Finally, the system is implemented on a embedded system. The performance of the system during the design and validation are above 90% of correct classification, and these results were proved with experimental test on several users.

1. Introducción

La comunicación es una habilidad básica y necesaria para los seres humanos, la cual se desarrolla con la capacidad del habla y el lenguaje. Sin embargo, existen personas que por diversos motivos no cuentan con dicha habilidad. Actualmente existen diversas técnicas y métodos que permiten la comunicación de dichas personas, los cuales son conocidos como sistemas alternativos de comunicación (SAC) [1] [2].

Los SAC han tenido cambios importantes debido al desarrollo de la tecnología, utilizando diversas señales biológicas en donde destacan las señales del parpadeo debido a que es una de las últimas funciones del cuerpo humano en perderse [3] [4] [5].

Las señales del parpadeo en algunas áreas de estudio como en los sistemas interfaz cerebro computadora (BCI por sus siglas en inglés) son denominadas como artefactos y en la mayoría de las aplicaciones desarrollados dichos artefactos tienden a ser eliminados [6] [7]. Sin embargo en este trabajo estas señales son utilizadas como fuente de información para codificar mensajes.

En el presente trabajo se realiza la detección, análisis y clasificación de señales de artefactos codificados, generados por la acción de parpadeo, para el diseño de un sistema alternativo de comunicación. Dichas señales se obtienen por medio del dispositivo BCI MindWave Mobile y la plataforma de desarrollo Raspberry Pi 3. Los artefactos de parpadeo fueron codificados y asociados a una expresión lingüística de acuerdo a lo que se muestra en la Tabla 1, sin embargo puede ser codificadas a otras expresiones según sea de mayor conveniencia para la comunicación.

Tabla 1. Códigos asociados a expresiones lingüísticas.

Expresión	Código de parpadeos
Si	5 parpadeos seguidos
No	3 parpadeos seguidos
Más o menos	4 parpadeos seguidos
No sé	3 parpadeos seguidos, 1 espacio con ojos abiertos y 2 parpadeos seguidos

2. Sistema Alternativo de Comunicación

2.1 Descripción general del sistema

El funcionamiento del sistema desarrollado se describe mediante la Figura 1. El proceso realizado por el sistema inicia con la lectura de señales EEG del dispositivo MindWave Mobile a través de su único electrodo localizado en la posición FPZ, de acuerdo al estándar internacional 10-

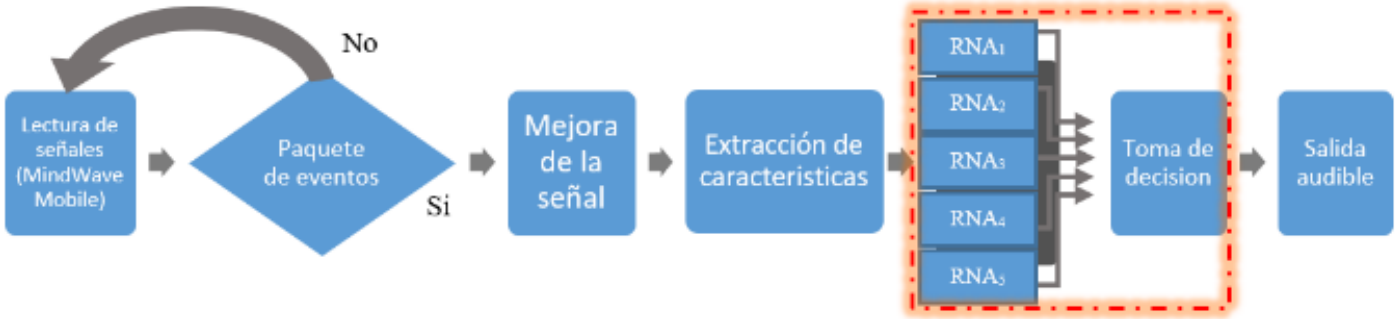

Figura 1 Esquema del sistema.

20, por un tiempo de dos segundos para la creación de un paquete de datos que posteriormente será analizado para determinar la existencia o inexistencia de algún evento. En caso de determinarse la existencia de un evento, se procederá a analizar el siguiente paquete de datos y si éste también presenta señales de eventos, se creará un paquete de mayor tamaño que contenga los datos de ambos paquetes analizados, al cual se le denominará "paquete de eventos". Éste proceso se detendrá una vez que se detecte en algún paquete de datos la inexistencia de eventos. En ese momento, el "paquete de eventos" creado previamente, pasará a la etapa de mejora de la señal, en donde se le aplicarán un procesamiento para eliminar posible ruido, realizar una normalización de los datos, creando así una señal mejorada y normalizada. Esta señal es luego procesada para extraer características, las cuales son utilizadas para un clasificador modular de 5 redes neuronales. Al tener una clasificación definida, el sistema enviará una señal que puede ser audible a través de una bocina en donde se dará a conocer algunas de las expresiones mostradas en la Tabla 1.

2.2 Lectura de señales EEG con artefactos

Para el diseño del sistema se tomaron muestras a tres hombres y una mujer de edades entre 23 y 25 años en una habitación con una iluminación aproximadamente de 600 luxes, ruidos de pláticas de personas dentro de la misma habitación y el sujeto de pruebas sentado en una silla con respaldo y descansa brazos observado el monitor del equipo de cómputo con el que se tomaron las señales a una distancia aproximadamente de un metro con un ángulo de inclinación de la cabeza de alrededor de 20° tal como se indica en la Figura 2

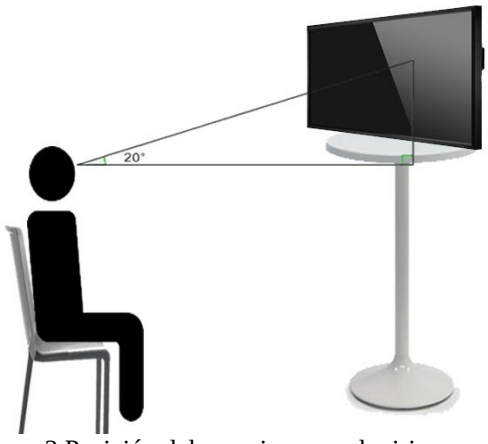

Figura 2 Posición del usuario para adquirir muestras.

La lectura de las señales EEG fue realizada por medio de la librería NeuroPi 0.1.0 [8], la cual está basada en el protocolo de comunicación Mindset [9], en donde se explica la composición de paquetes que MindWave Mobile transmite.

Los datos fueron recibidos y guardados en forma de paquetes en un vector s_i de 1024 muestras. Dicho vector se compone de 512 muestras leídas una iteración previa y 512 muestras leídas en el momento. Debido a que el dispositivo BCI MindWave Mobile tiene una frecuencia de muestreo de 512 Hz, se estará creando un vector equivalente a dos segundos con un segundo de traslape. De esta manera, se tendrán ventanas que se pueden analizar sin tener pérdida de información como se muestra en el esquema de la Figura 3 en donde se puede observar que los vectores s, contienen los datos que serán analizados y se encuentran traslapados con la ventana anterior por un segundo. Cada bloque de color rojo indica 512 muestras nuevas, mientras que los bloques de color purpura indican 512 muestras de la ventana anterior.

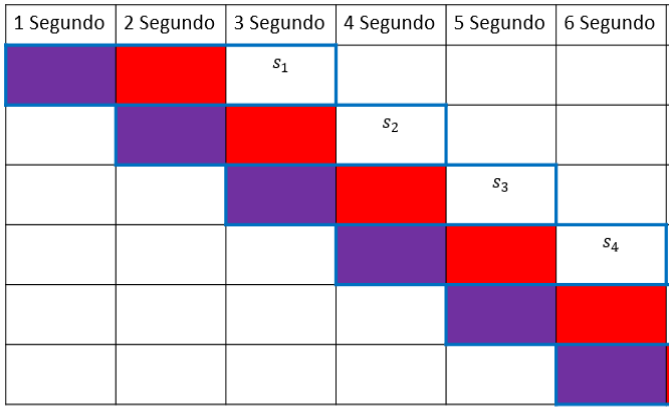

Figura 3 Ilustración de señales por bloques traslapados.

2.3 Paquetes de eventos

Para la detección de señales correspondientes a eventos cada vector s_i fue analizado por medio de la varianza (1) en donde K corresponde al número de muestras de s_i y μ_{s_i} (2) representa la media del vector s_i

$$\sigma^2_{s_i} = \frac{1}{K-1}\sum_{j=1}^{K}\left(s_{ij} - \mu_{s_i}\right)^2 \qquad (1)$$

$$\mu_{s_i} = \frac{1}{K-1}\sum_{j=1}^{K} s_{ij} \quad (2)$$

Con lo que se determinó que una señal de evento E_{s_i} será definida de acuerdo a (3).

$$E_{s_i} = s_i \ si \ \sigma_{s_i}^2 \geq 5{,}000 \quad (3)$$

Una señal de evento de parpadeos codificados está formada por un conjunto de varías señales E_{s_i}. Por tal motivo se realizó la concatenación de dichas señales, bajo la condición de que si las señales son continuas una de la otra, estas se juntarían formando una sola señal (paquete de evento), dicho proceso de definió con (4).

$$E_{s_i} = \begin{cases} E_{s_i} & si \quad E_{s_i} \ no\ es\ continua \\ E_{s_{i-1}} \ || \ E_{s_i}[512:1024] & si \quad E_{s_i} \ es\ continua \end{cases} \quad (4)$$

En donde || indica una concatenación.

2.4 Mejora de la señal

Con la finalidad de que el sistema de comunicación sea independiente del usuario, la información de las señales E_{s_i} deben ser procesada. Este procesamiento se describe a continuación. Primero la señal E_{s_i} es filtrada para acotarla en frecuencia a la banda de interés mediante un filtro Butterworth pasa bajas de primer orden con frecuencia de corte *wc=16Hz* definido por (5).

$$H(w): E_{s_i} \rightarrow E_{fs_i} \quad (5)$$

En donde E_{sf_i} corresponde a la señal filtrada de cada canal y $H(w)$ a la respuesta del filtro definida en (6).

$$|H(w)| = \frac{1}{\sqrt{1+(w/wc)^{2n}}} \quad (6)$$

Siendo *wc* la frecuencia de corte (16 Hz), *w* las componentes de frecuencia de la señal E_{s_i} y *n* el orden del filtro.

Posterior al filtrado se normalizo la señal E_{sf_i} con el algoritmo mapminmax [10] expresada en (7).

$$y = \frac{(y_{max} - y_{min})*(E_{sf_i} - \min(E_{sf_i}))}{\max(E_{sf_i}) - \min(E_{sf_i})} + y_{min} \quad (7)$$

En donde y_{min} y y_{max} toman valores de -1 y 1 respectivamente, por lo que (7) se puede definir como (8).

$$E_{snorm_i} = \frac{(2)*(E_{sf_i} - \min(E_{sf_i}))}{\max(E_{sf_i}) - \min(E_{sf_i})} - 1 \quad (8)$$

2.5 Extracción de características

Una vez teniendo las señales E_{snorm_i}, se extraen características que representaban grandes cambios entre un evento y otro, por lo que se utilizó: Número de picos $M_{E_{snorm_i}}$ definida en (9), en donde *p* corresponde al índice donde se localiza un máximo pico y *Dis(p-1,1)* corresponde a la distancia entre un pico y otro [11] [12].

$$M_{E_{snorm_i}} = \{p : p \text{ es un máximo de } E_{snorm_i} \geq 0.8 \text{ y } Dis(p-1,p) \geq 100\} \quad (9)$$

Distancia entre los picos 3 y 4 $Dis_{E_{snorm_i}}$ definida con (10).

$$Dis_{E_{snorm_i}} = \begin{cases} Dis(p_4 - p_3) & if \quad M_{E_{snorm_i}} \geq 4 \\ 0 & if \quad M_{E_{snorm_i}} < 4 \end{cases} \quad (10)$$

Y la energía de la señal E_{snorm_i}, $E_{E_{snorm_i}}$ definida con (11).

$$E_{E_{snorm_i}} = \sum_{n=0}^{N} \left| E_{snorm_{i_n}} \right|^2 \quad (11)$$

Con tales características se creó un vector característico X_{Es} tal como se muestra en (12).

$$X_{Es} = \begin{bmatrix} M_{E_{snorm_i}} \\ Dis_{E_{snorm_i}} \\ E_{E_{snorm_i}} \end{bmatrix} \quad (12)$$

Previo al análisis de clasificación, se realizó una normalización de las características que componen a los vectores X_{Es}, ya que dichas características pueden presentar magnitudes muy dispersas. Tal normalización se llevó a cabo utilizando el algoritmo mapminmax [10] para obtener una salida en intervalos de [-1, 1] en todas sus características.

Para la normalización de las características de picos $M_{E_{snorm_i}}$ se tomó el número máximo y mínimo de picos encontrados en el conjunto de muestras. Tomando un número máximo de picos de 12 y un número mínimo de 1 los cuales fueron sustituidos en la función del mapminmax representando dicha normalización con (13).

$$M_{normE_{snorm_i}} = 2\frac{M_{E_{snorm_i}} - 1}{11} - 1 \quad (13)$$

La normalización de las características de la distancia entre los picos 3 y 4 fue dada por (14), debido a que dentro de los conjuntos de entrenamiento se detectó un valor mínimo de distancia de 0 y un máximo de 2028.

$$Dis_{normE_{snorm_i}} = 2\frac{Dis_{E_{snorm_i}} - 0}{2028} - 1 \quad (14)$$

Finalmente la normalización de las características de energía $E_{E_{snorm_i}}$ fue dada por (15), la cual quedó definida de tal forma debido a que el valor mínimo de energía dentro del conjunto de entrenamiento fue de 71.3122 y el valor máximo fue de 989.2077.

$$E_{normE_{snorm_i}} = 2\frac{E_{E_{snorm_i}} - 71.3122}{917.8954} - 1 \quad (15)$$

Al realizar la normalización de las características, se procedió a crear un nuevo vector característico como el que se muestra en (16)

$$X_{normEs} = \begin{bmatrix} M_{normE_{snorm_i}} \\ Dis_{normE_{snorm_i}} \\ E_{normE_{snorm_i}} \end{bmatrix} \quad (16)$$

2.6 Diseño del clasificador

Los vectores X_{normEs} son analizados en esta etapa por un conjunto de 5 redes neuronales modulares, 4 de las 5 fueron diseñadas para poder reconocer códigos correspondientes a las expresiones “si”, “no”,” no sé” y “más o menos” y la última red fue diseñada para reconocer expresiones de ruido.

Cada red fue diseñada con 3 entradas, 4 neuronas en su capa intermedia con función de activación tangencial sigmoidea y una neurona en su capa de salida con una función lineal.
El entrenamiento de cada red se llevó a cabo con el algoritmo de retropropagación de Levenberg-Marquardt [13] en el software MATLAB para su posterior implementación por medio del lenguaje Python. Para dicho entrenamiento se consideraron un número máximo de 1000 épocas, un error mínimo de 1e-7, pesos iniciales igual a 1 en todas las sinapsis, un índice de verificación de disminución de error de 6 iteraciones y un conjunto de 305 vectores X_{normEs} compuestos por 61 vectores de las clases “si”, “no”, “no sé” y “más o menos” y 61 vectores más catalogados como “ruido”.

De los 305 vectores X_{normEs}, 48 de cada clase fueron utilizados como conjunto de entrenamiento, un conjunto de 13 vectores de cada clase fueron utilizados para validación y 13 más de cada clase para prueba, con lo que se obtuvo los resultados de eficiencia mostrados en la Tabla 2.

Tabla 2 Eficiencia de redes neuronales

Red de expresión	Entrenamiento	Validación	Prueba	Total
“si”	100%	100%	100%	100%
“no”	99.0%	98.5%	100%	99.1%
“no sé”	99.0%	100%	98.5%	99.1%
“más o menos”	100%	100%	100%	100%
“ruido”	98.4%	96.9%	100%	98.4%

Dichas redes neuronales, fueron definidas por (17).

$$\psi : X_{normEs} \rightarrow C = \{c_{si}, c_{no}, c_{ns}, c_{ms}, c_r\} \quad (17)$$

Siendo

$$\psi = D_{\varphi}(\varphi_{si}(X_{Es}), \varphi_{no}(X_{Es}), \varphi_{ns}(X_{Es}), \varphi_{ms}(X_{Es}), \varphi_r(X_{Es})) \quad (18)$$

D_{φ}, la salida del sistema modular que será analizado para tomar una decisión y $\varphi(\bullet)$ las salidas de cada red experta $\varphi_{si}(\bullet)$, $\varphi_{no}(\bullet)$, $\varphi_{ns}(\bullet)$, $\varphi_{ms}(\bullet)$ y $\varphi_r(\bullet)$. Cada salida obtenida es normalizada por medio de la función mapminmax inversa aplicada con (19)

$$O_i = \frac{1}{2}(D_{\varphi_i}(\bullet) + 1) \quad \forall \quad i = \{si, no, ns, ms, r\} \quad (19)$$

Finalmente, se tomara una decisión apoyado en [14], a través de (20)

$$C = \begin{cases} c_i & si \quad (O_i > (0.2 + O_j)) \wedge (O_i > 0.6) \\ c_r & si \quad Otro\ caso \end{cases} \quad \forall\, i, j = \{si, no, ns, ms, r\} \quad i \neq j \quad (20)$$

3. Resultados

Dicho sistema, fue programado a través del lenguaje Python [15] en una plataforma Raspberry Pi 3 [16]. En donde por medio de la librería eSpek y una bocina conectada a dicha plataforma, se reproduce una voz sintética indicando la expresión que fue detectada. Si no hay detección de expresión, no se produce ningún audio.

Para un correcto funcionamiento del sistema, se colocó un LED en los pines GPIO de la plataforma Raspberry Pi 3 tal como se muestra en la Figura 4.

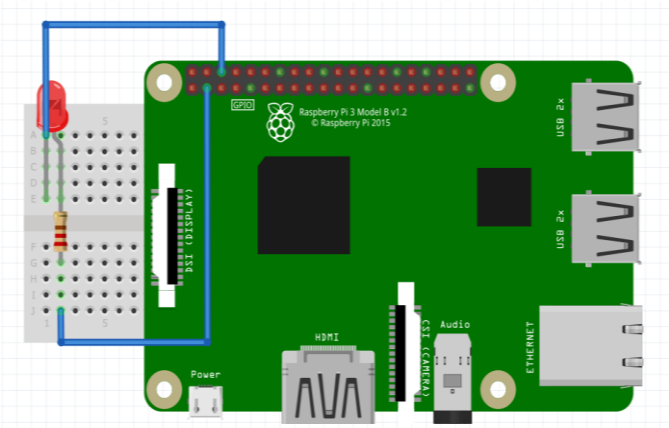

Figura 4 Colocación de un LED en los pines GPIO de la Raspberry Pi 3.

Este LED tiene la función de estar encendido siempre y cuando el sistema este detectando señales E_{s_i}, con la finalidad de que el usuario reciba una retroalimentación que le indique el momento en el que se puede realizar códigos de parpadeo, ya que si se realiza algún código inmediato a la realización de movimientos de extremidades, de cuello, o bien del sensor EEG, dichas señales sean clasificadas dentro de la clase "ruido". Por lo que el usuario debe realizar los códigos de parpadeo siempre y cuando el LED se encuentre apagado. Cabe mencionar que dicho LED se encenderá una vez iniciado cualquier código de parpadeo y se volverá a apagar cuando se termine de ejecutar.

El sistema fue puesto a prueba en 5 usuarios que no fueron consideras para su diseño. A estos usuarios se les realizaron preguntas aleatorias las cuales respondieron por medio del sistema, dando a conocer las respuestas que el mismo usuario quería expresar. El desempeño del sistema en estos experimentos fue similar a los desempeños reportados en el diseño del clasificador, Tabla 2.

4. Conclusiones

El sistema, fue implementado y probado en la plataforma de desarrollo Raspberry Pi 3 con la ayuda del lenguaje de programación Python. Con tal implementación se consiguió un sistema embebido para la detección de artefactos codificados utilizando análisis de varianzas, redes neuronales y una diadema EEG con un solo electrodo en la posición FPZ.

Con el desarrollo de este sistema embebido se demostró que el uso de sistemas BCI es una opción viable para el diseño de sistemas alternativos de comunicación a través del parpadeo, ya que estos últimos, muestran un comportamiento determinístico dentro de las señales EEG recibidas por el dispositivo BCI. Además por ser un primer prototipo se deja un campo abierto para el diseño de un sistema más robusto y con un número más amplio de expresiones.

5. Trabajo Futuro

Se deja como trabajo futuro la prueba de este sistema en personas que presenten discapacidades de comunicación, para realizar en caso de ser necesario, modificaciones en los parámetros del sistema descrito.

6. Agradecimientos

Se agradece al Tecnológico Nacional de México por el apoyo brindado para la realización de este trabajo bajo el proyecto 5684.16-P

7. Referencias

[1] C. Larraz Istúriz, «dossier: Comunicación aumentativa y alternativa,» MinusVal, 2006.

[2] D. A. Abadín, C. I. Delgado Santos y Á. Vigara Cerrato, «Guía de referencia,» *Comunicación aumentativa y alternativa*, vol. 1, nº 3, pp. 4-15, 2010.

[3] K. Mukherjee y D. Chatterjee, «Augmentative and Alternative Communication Device Based on Eye-Blink Detection and Conversion to Morse-Code to Aid Paralyzed Individuals,» de *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, India, 2015.

[4] J. Hori, K. Sakano y Y. Saitoh, «Development of Communication Supporting Device Controlled by Eye Movements and Voluntary Eye Blink,» de *Proceedings of the 26th Annual International Conference of the IEEE EMBS* , San Francisco, CA, USA, 2004.

[5] T. Bastos-Filho, A. Ferreira, D. Cavalieri, R. Silva, S. Muller y E. Pérez, «Multi-Modal Interface for Communication Operated by Eye Blinks, Eye Movements, Head Movements, Blowing/Sucking and Brain Waves,» de *2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, Rio de Janeiro, Brasil, 2013.

[6] L. F. Nicolas Alonso y J. Gomez Gil, «Brain Computer Interfaces, a Review,» *Sensors*, pp. 1211-1279, 31 Enero 2012.

[7] M. Ahn, M. Lee, J. Choi y S. C. Jun, «A Review of Brain-Computer Interface Games and an Opinion Survey from Researchers, Developers and Users,» *Sensors*, pp. 14601-14633, 11 Agosto 2014.

[8] V. Magri, «NeuroPi 0.1.0,» python, 2017. [En línea]. Available: https://pypi.python.org/pypi/NeuroPi/0.1.0. [Último acceso: 16 Febrero 2017].

[9] NeuroSky, «MindSet Communications Protocol,» NeuroSky inc, 2010.

[10] MathWorks, «mapminmax,» MathWorks, 2006. [En línea]. Available: https://www.mathworks.com/help/nnet/ref/mapminmax.html. [Último acceso: 22 Febrero 2017].

[11] J. Escolá, «Análisis e implementación de algoritmo de detección de complejos QRS en tiempo real,» Universidad Tecnológica Nacional (Facultad Regional de Buenos Aires), Buenos Aires, 2009.

[12] N. Yoder, «peakfinder(x0, sel, thresh, extrema, includeEndpoints, interpolate),» MathWorks, 06 Octubre 2009. [En línea]. Available: https://www.mathworks.com/matlabcentral/fileexchange/25500-peakfinder-x0--sel--thresh--extrema--includeendpoints--interpolate-. [Último acceso: 09 Diciembre 2016].

[13] MathWorks, «trainlm,» MathWorks, 2006. [En línea]. Available: https://www.mathworks.com/help/nnet/ref/trainlm.html. [Último acceso: 24 Febrero 2017].

[14] M. I. Chacon Murguía, Y. Quezada Holguín, P. Rivas Perea y S. Cabrera, «Dust Storm Detection Using a Neural Network with Uncertainty and Ambiguity Output Analysis,» de *Pattern Recognition:Third Mexican Conference, MCPR 2011*, Cancun, México, 2011.

[15] Python, «Python,» Python Software Foundation, 2017. [En línea]. Available: https://www.python.org/. [Último acceso: 15 Febrero 2017].

[16] RaspberryPi, «RaspberryPi,» RaspberryPi, 2016. [En línea]. Available: https://www.raspberrypi.org/. [Último acceso: 18 Abril 2016].