

<https://doi.org/10.23913/ride.v15i30.2267>

Artículos científicos

FIMALOF, una interfaz gráfica de usuario para el tratamiento de señales cerebrales y localización de fuentes

FIMALOF, a graphical user interface for brain signal treatment and source localization

FIMALOF, uma interface gráfica de usuário para processamento de sinais cerebrais e localização de fontes

Alina Santillán-Guzmán

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica,
México

alina.santillan@correo.buap.mx

<https://orcid.org/0000-0001-5165-4889>

María Monserrat Morín-Castillo

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica,
México

maria.morin@correo.buap.mx

<https://orcid.org/0000-0002-9121-5917>

José Jacobo Oliveros-Oliveros

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica,
México

jose.oliveros@correo.buap.mx

<https://orcid.org/0000-0002-1715-3631>

Jesús Rubén Vargas-García

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica,
México

actitudvgjr@gmail.com

<https://orcid.org/0000-0001-7084-6421>

Resumen

Las señales electroencefalográficas (*EEG*) necesitan ser tratadas y procesadas debido a que contienen diferentes tipos de distorsiones. Generalmente, se suelen emplear softwares especializados para tal propósito. En el presente trabajo se describe una nueva interfaz gráfica de usuario, FIMALOF (Filtrado, Mapeo y Localización de Fuentes), que consiste en un conjunto de programas que realizan tareas específicas para el tratamiento de señales cerebrales, incluyendo filtros digitales. También incluye técnicas de descomposición de señales, como el análisis de componentes independientes (*ICA*, Independent Component Analysis en inglés), *ICA*-filtros pasa-baja, descomposición empírica de modos. Asimismo, permite generar señales electroencefalográficas (*EEG*) sintéticas mediante una relación matemática basada en las características eléctricas del cerebro. Con esta señal *EEG* sintética se pueden probar y validar los filtros implementados para verificar su correcto funcionamiento. La interfaz también es capaz de procesar señales *EEG* reales. De esta manera se presenta una herramienta, alternativa a las existentes, para poder llevar a cabo el procesamiento de señales *EEG*.

Keywords: interfaz gráfica de usuario; filtrado; *ICA*; *EMD*.

Abstract

The electroencephalographic (*EEG*) signals need to be treated and processed due to the fact that they contain different artifacts. Generally, specialized software tools are used for that purpose. In the present work, a graphical user interface (GUI), FIMALOF (Filtering, Mapping and Source Localization) is introduced, which consists of a group of programs that do specific tasks for brain signals treatment, including digital filters, and decomposition signals techniques such as Independent Component Analysis (*ICA*), *ICA*-Low-pass filters, and Empirical Mode Decomposition (*EMD*). Moreover, it is possible to generate *EEG* synthetic signals through a mathematical relationship based on the brain's electrical characteristics. The synthetic *EEG* signals allow validating the proper functioning of the implemented filters. The interface is able to process real *EEG* signals. Thus, the software tool is introduced as an alternative to existing solutions, in order to process the *EEG* signals.

Keywords: graphical user interface; filtering; *ICA*; *EMD*.

Resumo

Os sinais eletroencefalográficos (EEG) precisam ser tratados e processados porque contêm diferentes tipos de distorções. Geralmente, software especializado é usado para essa finalidade. Este artigo descreve uma nova interface gráfica de usuário, FIMALOF (Source Filtering, Mapping and Localization), que consiste em um conjunto de programas que executam tarefas específicas para o processamento de sinais cerebrais, incluindo filtros digitais. Também inclui técnicas de decomposição de sinais, como análise de componentes independentes (ICA), filtros passa-baixa ICA e decomposição de modo empírico. Ele também permite a geração de sinais eletroencefalográficos (EEG) sintéticos por meio de uma relação matemática baseada nas características elétricas do cérebro. Com este sinal de EEG sintético, os filtros implementados podem ser testados e validados para verificar seu correto funcionamento. A interface também é capaz de processar sinais reais de EEG. Desta forma, é apresentada uma ferramenta, alternativa às existentes, para poder realizar o processamento de sinais de EEG.

Palavras-chave: interface gráfica do usuário; filtrado; ACI; EMD.

Fecha Recepción: Agosto 2024

Fecha Aceptación: Enero 2025

Introducción

La electroencefalografía (*EEG*) es una técnica ampliamente utilizada para el estudio del cerebro. Consiste en el registro de la actividad cerebral, tanto normal como anormal. Sin embargo, las señales adquiridas suelen estar contaminadas con diversos tipos de ruido o distorsiones, denominados artefactos. Estos artefactos pueden clasificarse en fisiológicos, como movimientos oculares, musculares, frecuencia cardíaca o sudoración, entre otros; y técnicos, como la desconexión de electrodos o interferencias de la línea de alimentación (Nayak y Anilkumar, 2022). Todas estas distorsiones necesitan ser eliminadas, dado que pueden estar ocultando información fisiológica de importancia. Por ello, para eliminarlas se usan diferentes técnicas de filtrado, que van desde las más simples como los filtros digitales, ya sean de paso-bajo, paso-alto, pasa-banda o banda eliminada, hasta las más complejas, como técnicas de descomposición de señales.

Una de las técnicas para suprimir diferentes tipos de distorsiones (artefactos), es la de análisis de componentes independientes (*ICA*). Con ella, es posible separar las señales *EEG* en diferentes componentes, en los que idealmente unos tendrán solamente la señal cerebral y

las distorsiones o artefactos de distintos tipos (Jiang et al, 2019). Sin embargo, existen algunos artefactos que no son tan simples de identificar, tal es el caso de los artefactos musculares, que pueden aparecer en muchos componentes y además suelen estar mezclados con señales fisiológicas en un mismo componente, por lo que, para su eliminación (Santillán et al., 2017), se propone una combinación de *ICA* con filtros pasa-baja y con modelos de espacio de estados. Otra técnica de reciente uso, es la de *Empirical Mode Decomposition* (*EMD*, en español: descomposición empírica de modos) (Huang y Attoh-Okine, 2005), que consiste en la eliminación del ruido y la tendencia (*trend*) de las señales a las que se aplica. En *EMD*, una sola señal de entrada se descompone en *N* modos (*Intrinsic Mode Functions*, *IMF*), donde el primero contiene las frecuencias más altas y el último contiene las más bajas.

Actualmente, existe una variedad de interfaces gráficas de usuario capaces de analizar, filtrar, y calcular distintas mediciones de similitud en tiempo y frecuencia entre señales. Una de ellas es *EEGLAB* (Swartz Center for Computational Neuroscience, 2022; Delorme y Makeig, 2004). Esta herramienta permite importar datos, aplicar filtros *FIR* (finite impulse response, filtros de respuesta finita) pasa-baja, pasa-banda o banda eliminada, dependiendo de la(s) frecuencia(s) a eliminar o analizar. La herramienta de *EEGLAB* resulta ser muy útil para quienes ya conocen cómo trabajar con ella. Sin embargo, pueden existir problemas cuando no se está familiarizado con la herramienta y cuando hay que hacer conversión de formatos de las señales de entrada para que puedan ser leídos con el software. Por otra parte, también existen otras interfaces gráficas de usuario como son: *Brain Visualizer*, la cual provee una visualización en tiempo real y en tercera dimensión del cerebro (Excellent brain EEG visualizar, 2022) ofreciendo con ello algo adicional al público; el software *Curry 8 SBR*, presenta un paquete de procesamiento de señales e imágenes así como el análisis de fuentes para potenciales relacionados con eventos o ERP (por sus siglas en inglés *event related potential*) (Compumedics Neuroscan, 2022) adicionalmente está *LORETA KEY* que permite, a partir de datos de señales EEG, realizar una reconstrucción de fuente distribuida, utilizando un modelo de cabeza esférica basada en tres capas (Fieldtrip, 2022). Adicionalmente, Niso et al. (2013) crearon *HERMES*, una interfaz gráfica utilizada para determinar las conexiones entre las señales y zonas activadas en el cerebro, así como para el cálculo de correlación, coherencia, causalidad de Granger, entre otros parámetros destacados. Estas interfaces, aunque útiles para aplicaciones específicas, presentan inconvenientes como altos costos, suelen ser poco intuitivas para el usuario, o en algunas ocasiones se debe pagar una licencia extra para poder usarlas. Por este motivo, y con el

objetivo de ofrecer una herramienta más intuitiva y didáctica, se desarrolló una mejora de la interfaz gráfica de usuario FIMALOF (Filtrado, Mapeo y Localización de Fuentes), como se describe en Santillán et al. (2018), cuyo objetivo es proveer una herramienta que no solo filtre las señales usando filtros digitales o las técnicas de descomposición de señales descritas anteriormente (*ICA*, *ICA* con filtro pasa-baja, *EMD*), sino que también filtra señales sintéticas creadas con un modelo matemático, con la cual se conocen los parámetros que caracterizan la señal y por lo tanto se pueden validar los algoritmos de identificación de la fuente y alterarlas de manera consciente para probar y validar los filtros. La señal creada puede ingresarse a la misma interfaz para ser filtrada, además, sienta las bases para que en un futuro se pueda usar como una herramienta en el campo de trabajo de los médicos, en particular a los que realizan esta actividad en zonas alejadas que no cuentan con dispositivos tecnológicos para el registro y análisis de señales *EEG* y poder detectar a tiempo alguna alteración o daño en el cerebro.

Materiales y métodos para el desarrollo de la interfaz gráfica

Se comenzará describiendo el funcionamiento general de cada elemento involucrado en el desarrollo de la interfaz FIMALOF.

Filtros digitales

Para poder llevar a cabo el filtrado de las señales cerebrales, para posteriormente hacer la localización de fuentes, se implementaron diversos filtros digitales. Primeramente, se desarrolló un filtro pasa-baja, con el cual se podrán eliminar las frecuencias por encima de una frecuencia de corte (f_c) especificada, y se preservarán aquellas por debajo de ésta (Malvino, 2000). Al rango de 0 a f_c se le llama banda pasante. Posteriormente se implementaron los filtros pasa-alta, los cuales actúan de manera inversa que los pasa-baja. Es decir, conservarán las señales cuyas frecuencias estén por encima de f_c mientras que se eliminarán aquellas por debajo de ésta. Otro filtro que es muy comúnmente usado en el procesamiento de señales cerebrales, es el filtro pasa-banda, el cual consiste en dejar pasar las señales cuyas frecuencias estén entre una frecuencia de corte inferior (f_{c1}) y una frecuencia de corte superior (f_{c2}), atenuando el resto. La diferencia entre ambas frecuencias de corte se conoce como ancho de banda (B) y es inversamente proporcional al factor Q , el cual, junto con el ancho de banda definen la selectividad del filtro. Así también, existe una

frecuencia central (f_0), la cual corresponde a la frecuencia que tiene la máxima amplitud. Finalmente, se desarrollaron filtros de banda-eliminada (de ranura o Notch), frecuentemente empleados para suprimir distorsiones provenientes de la línea de alimentación (50 o 60Hz). En general, al igual que los filtros pasa-banda, los filtros Notch cuentan con dos frecuencias (f_{c1} y f_{c2}), pero al contrario de éstos, los Notch eliminarán las señales comprendidas entre ambas frecuencias. El factor Q se calcula de la misma manera que en el pasa-banda, y la frecuencia central es la frecuencia principal que necesita ser suprimida. Todos estos filtros pueden ser diseñados en diferentes aproximaciones. Cuatro de esas aproximaciones se describen brevemente a continuación. La primera de ellas es la aproximación Butterworth, la cual se caracteriza por tener una respuesta plana en la banda pasante y una caída monótona en la banda eliminada, la pendiente decae a razón de $20 n$ dB/década o $6 n$ dB/octava, donde n el orden del filtro (Malvino, 2000). La siguiente aproximación es la *Chebyshev I*, presenta rizados en la banda pasante (A_p), cuya amplitud puede ser seleccionada por el usuario y generalmente está dada en dB. Este filtro se caracteriza porque su caída es más rápida que la de los filtros Butterworth sin embargo, presenta rizados en la banda pasante, reflejándose atenuaciones diferentes de cero en la banda de frecuencias donde dichas atenuaciones deberían ser cero (Malvino, 2000). Dependiendo de las necesidades del usuario, se elegirá este tipo de aproximación, haciendo un compromiso entre el decaimiento rápido en la banda eliminada y la atenuación debida a los rizados en la banda pasante. La siguiente aproximación desarrollada fue la de *Chebyshev II* o *Chebyshev inverso*, la cual presenta rizados en la banda eliminada, denotados por A_s (definido por el usuario, también dado en dB y menor que A_p) y tiene una banda pasante plana. La ventaja de este filtro es la banda pasante no presentará disminución en amplitud, mientras que su desventaja es que en la banda eliminada habrá menos atenuaciones. Finalmente, la aproximación Elíptica se caracteriza por la presencia de rizados en ambas bandas, tanto pasante como eliminada. Esto implica que se producirán atenuaciones en las frecuencias de la banda pasante y que parte de la señal permanecerá en las frecuencias de la banda eliminada. La mayor ventaja de esta aproximación es que con un menor orden, se puede lograr una respuesta similar a la que se obtendría con otra aproximación de mayor orden.

Análisis de componentes independientes

El objetivo de *ICA* es estimar los componentes independientes a partir del siguiente modelo (Santillán et al., 2017):

$$\mathbf{y}(k) = \sum_{l=1}^m c_l s_l(k) = \mathbf{C}\mathbf{s}(k), \quad (1)$$

Donde, $\mathbf{y}(k) = [y_1(k), \dots, y_n(k)]^T$, corresponde a las señales adquiridas por el sistema de medición, k indica el tiempo; n es el número de electrodos *EEG*; $\mathbf{s}(k) = [s_1(k), \dots, s_m(k)]^T$, denotan los m componentes independientes; y $\mathbf{C} = [c_1, \dots, c_m]$ es la matriz de mezclas de tamaño $n \times m$. De acuerdo a (1), se tienen dos incógnitas y solamente una variable conocida ($\mathbf{y}(k)$). Por ello, es necesario tomar en cuenta las suposiciones de *ICA*, descritas en (Santillán et al., 2017).

Uno de los algoritmos para el cálculo de los componentes independientes y que se usan más comúnmente en *MATLAB* debido a su rápida convergencia y a su facilidad de implementación es *FastICA* (Santillán et al., 2017). En la presente interfaz, *FastICA* es el algoritmo usado para obtener los componentes independientes. Los componentes con artefactos son eliminados y las señales limpias son reconstruidas usando la nueva serie de componentes sin artefactos y la matriz de mezclas.

Descomposición empírica de modos

Esta técnica: descomposición empírica de modos permite la entrada de una señal univariable, $y(k)$, para descomponerla en una serie de funciones intrínsecas de modos (intrinsic mode functions, IMF), $IMF_i(k)$, y un residuo den $r_N(k)$, el cual puede ser una constante o considerarse como la tendencia ('trend') de la señal de entrada. Matemáticamente, se expresa como (Huang y Attoh-Okine, 2005):

$$y(k) = \sum_{i=1}^N IMF_i(k) + r_N(k). \quad (2)$$

Al procedimiento para identificar y extraer los IMFs se denomina *sifting*. Cada IMF debe cumplir dos condiciones fundamentales. La primera es que el número de extremos y el número de cruces entre la envolvente superior y la envolvente inferior (definidas por los máximos y mínimos locales, respectivamente) debe ser menos que dos.

Para ello, se identifican los extremos locales (máximos y mínimos), los cuales determinan las envolventes superior e inferior. Posteriormente, se calcula el valor medio, $m_1(k)$, entre ambas curvas. Este valor medio se resta a la señal de entrada para extraer al primer candidato IMF, conocido como proto-IMF, denotado $pIMF_1(k) = y(k) - m_1(k)$. El

proto-IMF es la nueva señal de entrada y el proceso de *sifting* es aplicado iterativamente a éste. Para evitar un número elevado de iteraciones, así como perder el sentido físico de los IMFs, dos criterios se emplean para parar el proceso, los cuales se detallan en (Huang y Attoh-Okine, 2005). Cuando ambos criterios se satisfacen, después de i iteraciones, el primer IMF ($IMF_1(k)$, el cual contiene la frecuencia más alta de la señal) es obtenido:

$$IMF_1(k) = pIMF_{(i-1)}(k) - m_i(k). \quad (3)$$

El primer IMF se separa del resto de la señal de entrada de la siguiente manera:

$$r_1(k) = yk - IMF_1(k). \quad (4)$$

Con ello, $r_1(k)$ es la nueva entrada al proceso de *sifting*. El algoritmo se detiene cuando el residuo es una constante o corresponde a la tendencia de la señal.

Combinación de ICA y filtros pasa-baja

Debido a que los artefactos musculares no pueden ser completamente eliminados mediante el uso exclusivo de filtros pasa-baja o ICA, en la presente interfaz se cuenta con una combinación de ICA con filtros pasa-baja, tal como se describe en (Santillán et al., 2017). Esta combinación comienza con la aplicación de ICA a las señales de entrada. Se obtienen los componentes independientes y de éstos se seleccionan por inspección visual aquellos que contengan información fisiológica mezclada con artefactos musculares o que contengan puramente artefactos. A los componentes seleccionados se les aplica un filtro pasa-baja para evitar que los componentes independientes contengan una mezcla de señales fisiológicas con artefactos. Si existen componentes independientes que, a pesar de ser filtrados, siguen conteniendo residuos de artefactos, estos serán eliminados. Las señales EEG filtradas se reconstruyen mediante la nueva serie de componentes independientes libres de artefactos, junto con la matriz de mezclas C previamente definida.

Método de identificación de fuentes

Uno de los métodos utilizados comúnmente para validar la identificación de una fuente es a través de la generación de la misma en forma de fuente sintética. Para ello, se resuelve el problema directo de identificación de fuentes, el cual se basa en un modelo de medio conductor ya estudiado en Morín et al. (2019).

Una representación de este modelo, considera a la cabeza en dos capas conductoras, donde la más interna, Ω_1 representa al cerebro y la siguiente Ω_2 representa la otra capa conductora que debe agrupar a líquido intracraneal, cráneo y cuero cabelludo, S_1 representa la superficie de separación entre Ω_1 y Ω_2 , S_2 la superficie del cuero cabelludo. A partir de ciertas condiciones y dado que se consideran capas conductoras, es posible aplicar, para la modelación algunas de las leyes de Maxwell, esta condición y algunas restricciones hacen posible encontrar que el potencial electrostático u satisface el problema de contorno siguiente, cuando la fuente se encuentra en la corteza cerebral:

$$\begin{aligned} \Delta u &= 0 && \text{en } \Omega_1 \\ \Delta u &= 0 && \text{en } \Omega_2 \end{aligned}$$

Con las condiciones de contorno:

$$\begin{aligned} u_1 &= u_2 && \text{en } S_1 \\ \sigma_i \frac{\partial u_1}{\partial n_1} &= \sigma_2 \frac{\partial u_2}{\partial n_2} + j \cdot n_1 && \text{en } S_1 \\ \frac{\partial u_2}{\partial n_2} &= 0 && \text{en } S_2 \end{aligned}$$

Usando estos planteamientos, se pueden plantear y resolver dos problemas principales: El problemas directo e inverso (Morín et al., 2019). Para el problema directo en 2 dimensiones, se supondrá que se conoce la fuente $g = j \cdot n_1$, la cual restringida a S_1 , denotada como $g(\theta)$, cuando está descrita por series de Fourier y se calcula el potencial (medición) producida por dicha fuente (Kirsch, 1996). Para ello, se propone: $g(\theta) = \sum_{k=1}^{\infty} g_k^1 \cos k\theta + g_k^2 \sen k\theta$. Donde los coeficientes de Fourier están dados en la forma $g_k^1 = R_1 \int_{-\pi}^{\pi} g(\theta) \cos k\theta d\theta$ y $g_k^2 = R_1 \int_{-\pi}^{\pi} g(\theta) \sen k\theta d\theta$

Los potenciales se buscan en la siguiente forma:

$$\begin{aligned} u_1(r, \theta) &= \sum_{k=1}^{\infty} a_k^1 r^k \cos k\theta + b_k^1 r^k \sen k\theta \\ u_2(r, \theta) &= \sum_{k=1}^{\infty} (a_k^2 r^k + b_k^2 r^{-k}) \cos k\theta + \sum_{k=1}^{\infty} (c_k^2 r^k + d_k^2 r^{-k}) \sen k\theta \end{aligned}$$

De acuerdo a las condiciones de contorno, al desarrollar y simplificar, se obtiene el potencial generado por la expresión $g(\theta) = \sum_{k=1}^{\infty} (g_k^1 \cos k\theta + g_k^2 \sen k\theta)$, se expresa en la forma:

$$V(\theta) = u_2|_{s_2} = u_2(r, \theta)$$

$$= \sum_{k=1}^{\infty} \left(\frac{2g_k^1(R_1^{k+1} + R_2^k)}{k[(\sigma_1 - \sigma_2)R_1^{2k} + (\sigma_1 + \sigma_2)R_2^{2k}]} \right) \cos k\theta$$

$$+ \sum_{k=1}^{\infty} \left(\frac{2g_k^2(R_1^{k+1} + R_2^k)}{k[(\sigma_1 - \sigma_2)R_1^{2k} + (\sigma_1 + \sigma_2)R_2^{2k}]} \right) \sin k\theta$$

donde $a_k^1 = \frac{g_k^1(R_1^{2k} + R_2^{2k})}{k[(\sigma_1 - \sigma_2)R_1^{2k} + (\sigma_1 + \sigma_2)R_2^{2k}]R_1^{k-1}}$, $b_k^1 = \frac{g_k^2(R_1^{2k} + R_2^{2k})}{k[(\sigma_1 - \sigma_2)R_1^{2k} + (\sigma_1 + \sigma_2)R_2^{2k}]R_1^{k-1}}$

Esta expresión puede ser considerada como la medición (electroencefalográfica) producida por la fuente, donde: g_k^1 y g_k^2 son los coeficientes de Fourier de g y dado que se consideró desde un inicio que la cabeza sería representada por esferas concéntricas, podemos convenir en que el radio del cerebro es R_1 , R_2 el radio de la cabeza, σ_1 y σ_2 son las conductividades promedio del cerebro y cabeza respectivamente, θ indica la posición de cada electrodo ubicado sobre el cuero cabelludo. Con esta relación se genera la señal EEG sintética mencionada, con el cual es posible validar la interfaz implementada.

Resultados

La interfaz gráfica fue implementada en *MATLAB*, la cual se muestra en la figura 1.

La interfaz FIMALOF incluye diversos botones, distribuidos según su funcionalidad: Cuatro para filtros digitales (pasa-baja, pasa-alta, pasa-banda y rechaza-banda), tres para técnicas de descomposición de señales (*ICA*, *ICA*-filtro pasa-baja y *EMD*) y uno adicional para realizar conversión de formatos. También es posible seleccionar elegir el idioma, ya sea español o inglés, además de contar con un botón que permite generar una señal *EEG* sintética, creado a partir del modelo matemático descrito. Si se selecciona una fuente a ser filtrada es una generada a partir del problema de contorno, será necesario generar una medición V (medición sintética), la cual será expresada por series de Fourier en el modelo de medio conductor. Por la estructura, se generará una señal *EEG* sintética que podrá ser filtrado mediante esta interfaz. En el caso en que el usuario ingrese datos fuera de rango, la interfaz gráfica desplegará una alerta, informándole el rango o límite del dato que ingresó erróneamente.

Figura 1. GUI principal FIMALOF



Fuente: Elaboración propia.

A continuación, se describen algunas funciones especiales usadas en la interfaz

a) Sub-interfaz: conversión de formatos

La pantalla principal de la interfaz contiene un botón para la conversión de formatos, que permite convertir archivos de los formatos .edf, .cnt y .mat al formato .mat compatible con FIMALOF. Esta conversión es necesaria para que FIMALOF pueda cargar y leer las señales. Es importante mencionar que dentro de la interfaz existen dos tipos de conversión de .edf a .mat. Esto es debido a que, por un lado, es posible determinar los canales que provienen del *EMOTIV Epoc +* (sistema de mediciones *EEG* de 14 canales más dos de referencia, (*EMOTIV*, 2024)), mientras que, por otro, existen otros sistemas cuyos canales con señales *EEG* pueden variar, pero siguen teniendo el formato.edf originalmente.

b) Sub-Interfaz: Filtros

En el caso que el usuario selecciona aplicar un filtro pasa-baja de la interfaz principal, el usuario deberá ingresar los parámetros necesarios para cada tipo de aproximación, como son: la frecuencia de muestreo (F_s), la de corte (F_c), el orden (Orden), A_p , para el caso de filtros tipo *Chebyshev I* y el elíptico, A_s , para el caso de filtros tipo *Chebyshev II* y el elíptico, el número de muestra de inicio de las señales a procesar (Punto de Inicio) y la cantidad de segundos a analizar (Segundos). Posteriormente, se deberán seleccionar los canales a filtrar,

los cuales pueden ser parte de cuatro posibles configuraciones que, actualmente, se están considerando en la interfaz: Configuración 1, correspondiente al *EMOTIV* (14 canales); Configuración 2, de 19 canales, correspondiente al sistema 10-20; Configuración 3, de 32 canales, correspondiente al sistema 10-20; Configuración 4, correspondiente a 128 canales. En este caso, el usuario podrá elegir aquellos que requiera para analizar y procesar.

Una vez que el usuario seleccionó el (los) canal(es) a filtrar, las señales EEG se cargan en la interfaz. Como se mencionó anteriormente, los datos deberán estar en formato .mat para poder utilizarlos en FIMALOF. Si están en otro formato, se hace la conversión con la sub-interfaz “Convertir Data”, con la cual los datos son convertidos a .mat, para que puedan ser procesados. Inmediatamente después de ser cargada la información a filtrar, se grafican los canales seleccionados en la parte superior derecha, y se hace una estandarización (varianza unitaria y valor medio de cero). Posteriormente, la(s) aproximación(es) de filtro(s) se selecciona(n). En la parte inferior izquierda, se grafica la respuesta en frecuencia de cada selección en el color correspondiente a la aproximación. Se aplica(n) los filtro(s) seleccionado(s) y se grafican las señales filtradas en la parte superior derecha. En la parte inferior derecha, se grafica el espectro de potencia de un canal a elegir.

Así, una vez aplicadas las distintas aproximaciones, el usuario podrá determinar cuál de ellas otorga mejores resultados de filtrado de las señales *EEG*, de acuerdo a la banda de frecuencias que desee analizar posteriormente. Lo anterior se hace de manera análoga para todos filtros, los cuales han sido mejorados de la versión previa de la interfaz FIMALOF. En la presente versión, se agregó el guardado de las señales, además de que se generaron mensajes de advertencia al usuario cuando los datos ingresados por este no se realizaron adecuadamente, informándole cómo pueden ser escritos. Para más detalles sobre el funcionamiento de la parte de filtrado, referirse a (Santillán et al., 2018).

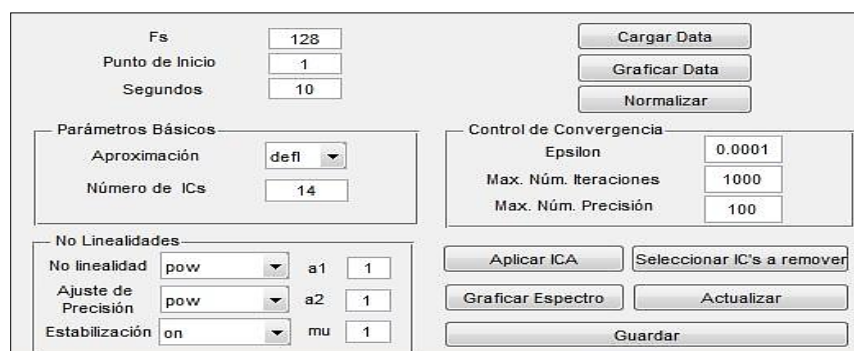
c) Sub-Interfaz: ICA

Como se mencionó anteriormente, con ICA es posible descomponer los n canales de entrada en m componentes independientes de salida. De éstos, se seleccionan aquel(los) que contenga(n) artefactos para eliminarlos, y de este modo, reconstruir las señales limpias.

La interfaz gráfica correspondiente a ICA, se muestra en la figura 2. En este caso, el usuario deberá ingresar el valor de la frecuencia de muestreo, de inicio de la señal, además del número en segundos de las señales a limpiar. Para este caso particular, no se seleccionan los canales a ser filtrados, ya que ICA trabaja con todos los canales de entrada para producir

los m componentes independientes. Una vez ingresados los datos, se cargan las señales a ser filtradas y se elige la forma de cómputo del algoritmo *fastICA*. En el panel “Parámetros Básicos” se encuentran: “defl” con el que los componentes independientes se calcularán uno por uno; y “symm”, que calculará todos los componentes en forma paralela. El valor por defecto es “defl”. Se puede escribir el número de componentes que se quieren tomar. Por defecto, aparecerá el mismo número de canales, es decir, no se hace una reducción de la dimensión de las señales.

Figura 2. Interfaz correspondiente a ICA



The screenshot shows a software interface for Independent Component Analysis (ICA). It is divided into several sections:

- Top Left:** Input fields for 'Fs' (128), 'Punto de Inicio' (1), and 'Segundos' (10).
- Top Right:** Three buttons: 'Cargar Data', 'Graficar Data', and 'Normalizar'.
- Middle Left (Parámetros Básicos):** A dropdown menu for 'Aproximación' set to 'defl' and an input field for 'Número de ICs' set to 14.
- Middle Right (Control de Convergencia):** Input fields for 'Epsilon' (0.0001), 'Max. Núm. Iteraciones' (1000), and 'Max. Núm. Precisión' (100).
- Bottom Left (No Linealidades):** Three rows of controls: 'No linealidad' (pow, a1: 1), 'Ajuste de Precisión' (pow, a2: 1), and 'Estabilización' (on, mu: 1).
- Bottom Right:** Buttons for 'Aplicar ICA', 'Seleccionar IC's a remover', 'Graficar Espectro', 'Actualizar', and 'Guardar'.

Fuente: Elaboración propia

En el panel “No linealidades” están las funciones no-lineales que se usan para obtener C (matriz de mezclas); tales como “pow3”, “tanh”, “gauss” y “skew” (Hyvarinen y Oja, 2000). Dichas funciones también son empleadas para tener un ‘Ajuste de precisión’. La ‘Estabilización’ puede o no estar habilitada. El valor por defecto es ‘on’, indicando que se utilizará una versión estable para obtener en forma más efectiva los componentes independientes. Los coeficientes ‘a1’ y ‘a2’; y ‘mu’, que corresponde al tamaño de paso también deben ser ingresados. Dichos parámetros tienen el valor por defecto igual a 1. Los parámetros para controlar la convergencia, se muestran en la parte derecha. Los valores por defecto son “Epsilon”=0.0001, con el cual se detiene el algoritmo; “Max. Núm. Iteraciones” = 1000, parámetro relacionado con el número máximo de iteraciones para hacer la estimación de cada componente independiente; y “Max. Núm. Precisión” igual a 100, el cual tiene que ver con el número máximo de iteraciones para ajustar al algoritmo.

Una vez ingresados todos los datos correspondientes, se aplica el algoritmo *fastica*, se obtienen los componentes y de entre ellos se eligen por inspección visual aquellos que tanto en el dominio del tiempo como en la frecuencia sean considerados como contaminados

con artefactos. Una vez seleccionados, se eliminan para poder suprimir los distintos tipos de artefactos que contengan. Todos los datos de entrada previamente descritos, pueden ser actualizados tantas veces lo considere necesario el usuario. Finalmente, se podrán guardar los resultados para su futura comparación y análisis.

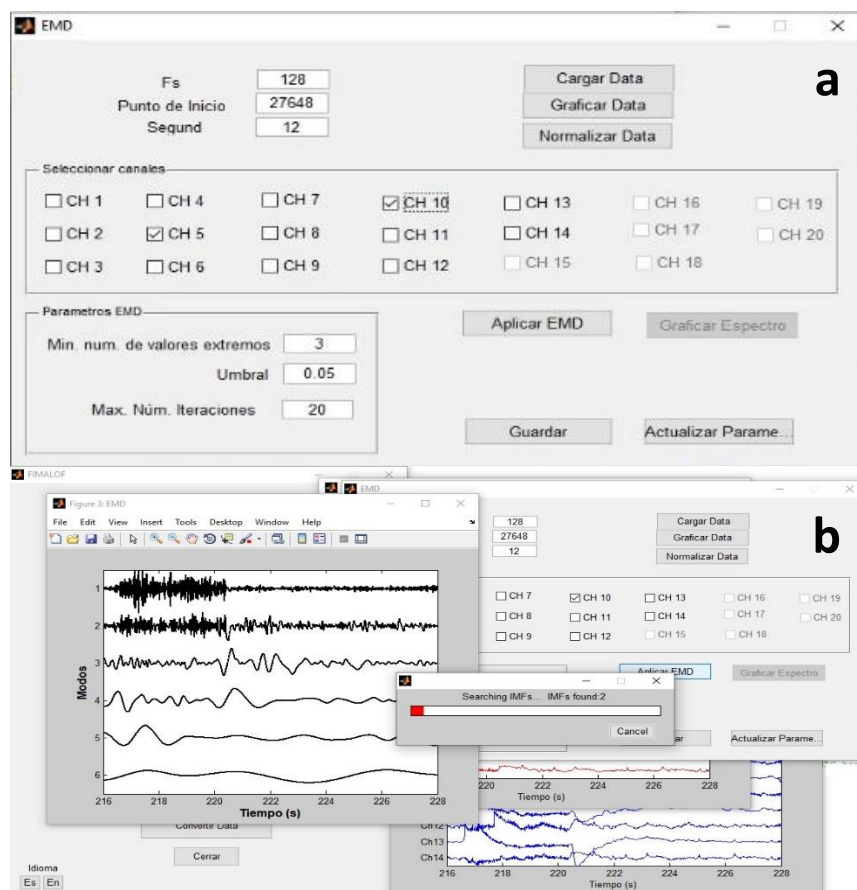
d) Sub-Interfaz: ICA-LPF

La interfaz de la combinación de ICA-LPF es similar a la correspondiente a ICA, en la cual se le agrega un botón que se usa para seleccionar los componentes con artefactos musculares y de esta manera aplicarles un filtro pasa-baja. Los componentes seleccionados son filtrados con un filtro pasa-baja *Butterworth* de sexto orden, con una $f_c=30\text{Hz}$ y que está previamente determinado internamente. La nueva serie de componentes, que contiene a los filtrados y a los no filtrados, se muestran en una figura para seleccionar de entre ellos los que aún sigan teniendo artefactos musculares o de otro tipo. La combinación *ICA-LPF* permite, de manera más eficiente que usando únicamente *ICA*, la eliminación de artefactos musculares.

e) Sub-Interfaz: EMD

En la Figura 3a se muestra la sub-interfaz correspondiente a *EMD*. Los parámetros a introducir por el usuario, además de la frecuencia de muestreo, el punto de inicio y el número de segundos son: El número máximo de iteraciones para obtener los *IMFs*, el valor umbral que sirve para detener el algoritmo, como se describió en secciones anteriores y el tipo de interpolación a usar (cúbica o lineal). Asimismo, es posible elegir el canal a filtrar con *EMD*, uno a la vez, eligiendo primero la configuración, tal como en el caso de los filtros digitales. Cada parámetro básico cuenta con un valor predeterminado. Una vez ingresando los valores, se puede aplicar *EMD* al canal seleccionado, graficar su espectro y seleccionar aquellos *IMFs* que son considerados como contaminados con algún tipo de artefacto. Así como en las interfaces anteriores, se pueden guardar los resultados. En la figura 3b se presenta un ejemplo de cómo se pueden ver los *IMFs* de uno de los canales elegidos.

Figura 3. a) Interfaz correspondiente a EMD, mostrando los parámetros a elegir; b) ejemplo de cómo se observan los IMFs (modos) de uno de los canales elegidos.



Fuente: Elaboración propia.

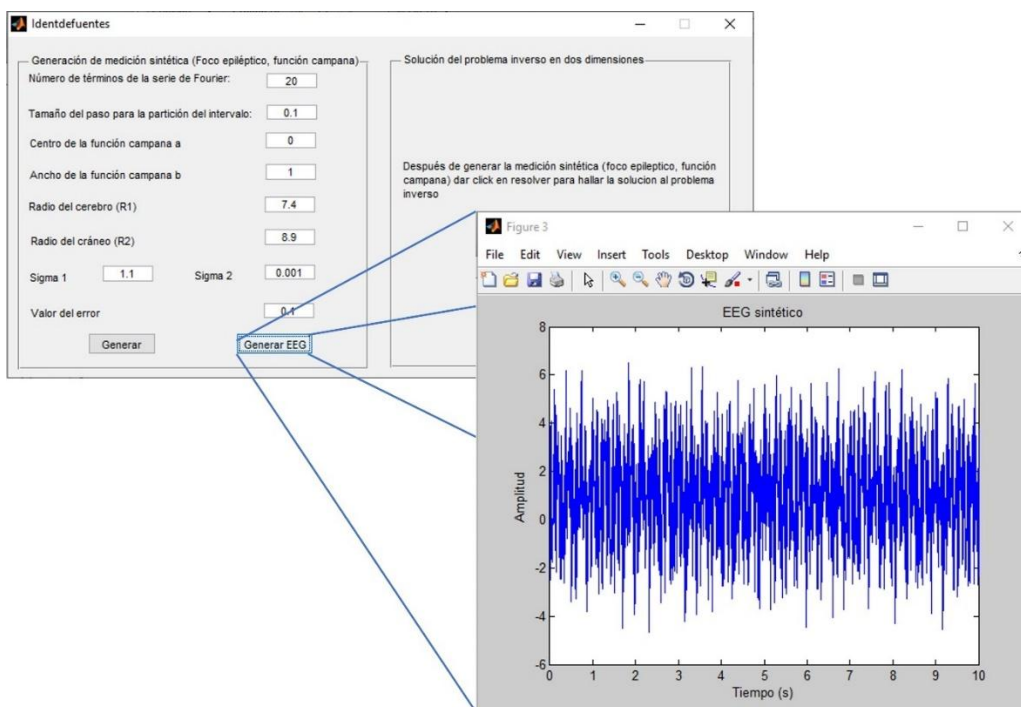
f) Subinterfaz: Identificación de fuentes

Finalmente, son implementados los modelos matemáticos utilizados para la subinterfaz *Identdefuentes* que es usada para la localización de fuentes. Esta consiste en generar una señal *EEG* a través del modelo matemático de medio conductor, descrito anteriormente (Santillán et al., 2021). En la interfaz de identificación de fuentes se piden los datos como son: número de términos de la serie de Fourier, el valor del tamaño del paso para la partición del intervalo, el punto central y el ancho de la función campana, el radio del cerebro y del cráneo, sigma 1, sigma 2 y error. Esta información es para generar una fuente de tipo epiléptica donde la cabeza está representada por dos círculos concéntricos; sigma1 y sigma2 son las conductividades de los medios. Los datos correspondientes a la función campana se deben a la representación del foco epiléptico. Después, al dar click en *Generar*,

la gráfica de la fuente y de su correspondiente serie de Fourier, es mostrada. Al mismo tiempo, también se genera la gráfica de la medición exacta y de la medición con error producidas por la fuente; ambas mediciones obtenidas mediante la solución del problema de contorno electroencefalográfico. Posteriormente, los botones *Generar señales EEG* y *Resolver* son habilitados, y al dar click en *Generar EEG* se produce un *EEG* sintético a partir de la medición obtenida anteriormente. Dicho *EEG* sintético contendrá frecuencias de 3.5Hz, 11Hz y 60Hz, emulando un evento epiléptico de ausencia, ondas alfa y el artefacto técnico de la línea de alimentación, respectivamente.

Al elegir el botón de “*Generación de una medición sintética*”, la subinterfaz mostrada en la Figura 4, se abre. Tal como se observa, se ingresan valores como el número de términos de la serie de Fourier, el valor del tamaño del paso para la partición del intervalo de θ , que va de $-\pi$ a π , el valor de la posición del centro de la función campana (que es la que se eligió por emular un foco epiléptico) que aproxima a la fuente, el valor del ancho de la función campana que aproxima a la fuente (Morín et al., 2019), el valor de R1, R2, de las conductividades (“Sigma 1”, “Sigma 2”) y el valor del error. Esta subinterfaz muestra dos botones, el de *Generar* que es usado para crear la fuente previamente establecida dentro del código y con la cual se podrá generar el *EEG* sintético, usando el botón “*Generar señales EEG*”. La interfaz proporciona la señal *EEG* sintético requerido, lo guarda y aparece en la pantalla, de esta manera ya es posible aplicar el filtrado que se desee, como se muestra en la figura 4.

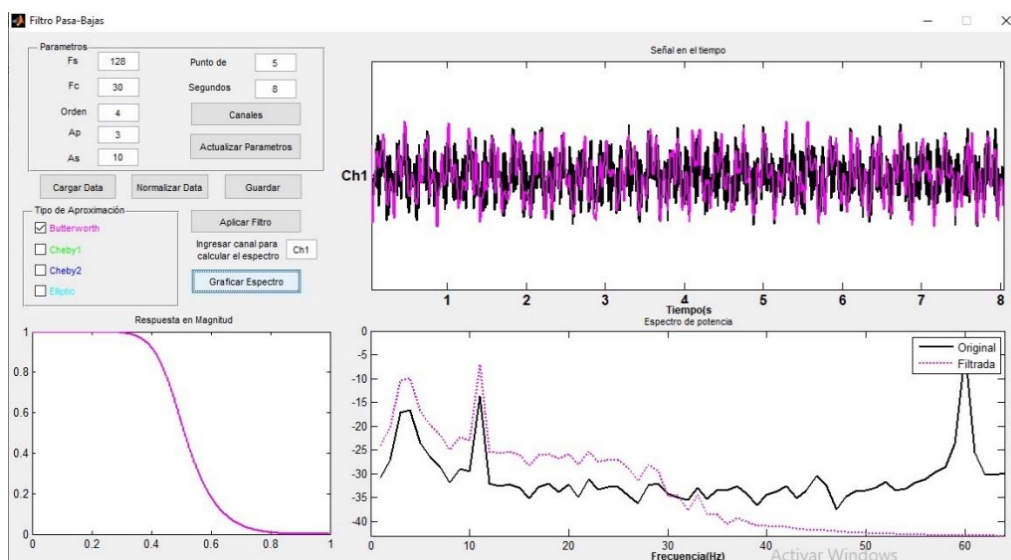
Figura 4. Subinterfaz correspondiente a la generación de una medición sintética. Aquí se muestra la información solicitada al usuario para generar la señal EEG.



Fuente: Elaboración propia.

En la figura 5 se muestra un ejemplo del uso de un filtro pasa-baja aplicado a una EEG sintético de señales que contienen una frecuencia de 3.5Hz (Panayiotopoulos et al., 2010), 10Hz (señal alfa) y 60 Hz (artefacto de la línea de alimentación). A esta señal se añadió ruido aleatorio para simular condiciones más realistas. La señal generada se almacena automáticamente, permitiendo su reutilización en la interfaz para procesos de filtrado adicionales. Como se observa en la Figura 5, se aplicó un filtro Butterworth de cuarto orden y con una $f_c=30\text{Hz}$. Las señales en color negro son las originales y en magenta se presentan las filtradas, tanto en tiempo como en frecuencia. En ambos casos es posible observar una diferencia, notándose que las frecuencias por arriba de 30Hz han sido eliminadas.

Figura 5. Ejemplo del uso de un filtro pasa-baja aplicando a un EEG sintético.



Fuente: Elaboración propia

Discusión

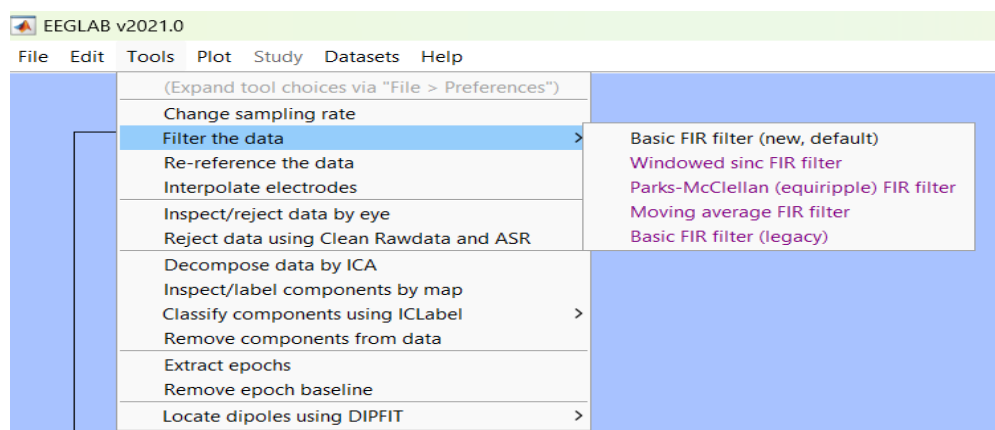
Ventajas: Esta interfaz ofrece fácil accesibilidad y manejo para el usuario debido a que no es necesario contar con conocimientos especializados para su uso. Por otra parte, la generación de señales EEG sintéticas a través del modelo matemático de medio conductor, en el que conocemos las fuentes que generan dichas señales sintéticas, permite con señales EEG ideales totalmente caracterizadas, que pueden ser contaminadas con ruido aleatorio asociado a diferentes artefactos. Estas señales contaminadas pueden ser filtradas desde los diferentes filtros implementados en la interfaz para obtener una señal limpia desde la cual se puedan identificar las fuentes generadoras de señales *EEG* ideales. Esto representa una ventaja para los usuarios y modeladores que, hasta el conocimiento de los autores, no ofrecen otras interfaces. Así, con esta interfaz es posible generar datos de señales que representen alteraciones en el EEG que, en general, puede ser complicado conseguir, pero con esta metodología se puede contribuir a la identificación de alteraciones neuronales como epilepsia, daño cerebral o tumoraciones. Se destaca que este tipo de implementaciones, que utilizan modelos matemáticos basados en leyes de la Física y matemática avanzada que respaldan el marco teórico de este trabajo, son desarrollos tecnológicos robustos que se requieren para que se utilicen por una comunidad más amplia, con lo que permitirá avanzar más eficazmente en el análisis, estudio e investigación de problemas en el cerebro. Cabe destacar que la interfaz puede tener un uso didáctico que impactará en las comunidades de

estudiantes y docentes de medicina, psicología, ingeniería y ciencias. La interfaz está desarrollada en español e inglés. El idioma por defecto es el español. Incluir este idioma la hace accesible a los nativos de esta lengua que es la segunda más hablada del mundo.

Comparaciones: Se muestran comparaciones entre *EEGLAB* y la interfaz propuesta, FIMALOF con el objetivo de mostrar las similitudes y diferencias que nos permitan concluir que FIMALOF es comparable con *EEGLAB* que es ampliamente utilizada para el procesamiento de señales. Así, la interfaz descrita en este trabajo, podría considerarse una alternativa de fácil uso para el tratamiento de señales, generación de señales sintéticas, así como la localización de fuentes.

En la Figura 6 se muestran los filtros que son posibles utilizar con *EEGLAB*, los cuales corresponden a filtros de respuesta finita (*FIR*) e *ICA*, en comparación con la interfaz FIMALOF, que puede realizar filtros de respuesta infinita (*IIR*), además de *ICA* e *ICA* combinada con filtros pasa-baja (ver Fig. 1).

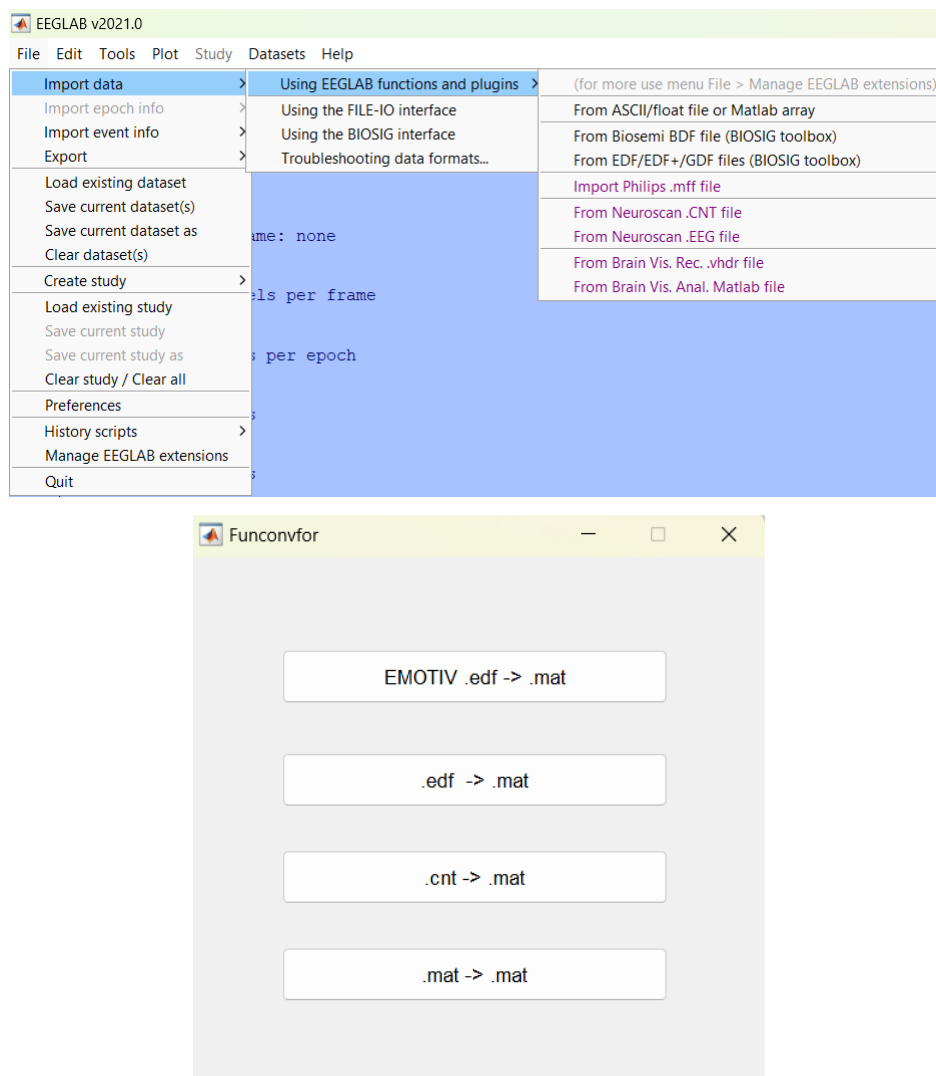
Figura 6. Parte de la herramienta *EEGLAB*, mostrando los filtros disponibles a aplicar a señales EEG.



Fuente: Elaboración propia, basada en *EEGLAB*.

Uno de los problemas a los que el usuario se enfrenta normalmente cuando utiliza interfaces gráficas consiste en importar los datos. En la figura 7 se muestra los distintos formatos que pueden ser cargados con *EEGLAB*, además de que permite procesar señales previamente localizadas en el espacio de trabajo de *MATLAB*. Comparando con la interfaz propuesta, FIMALOF, también es posible importar datos en distintos tipos de formatos, provenientes de diversos sistemas, tal como está descrito en la sección de resultados en la sub-interfaz conversión de formatos. Si bien es cierto que en *EEGLAB* es posible usar distintos formatos, en FIMALOF resulta más sencillo realizar dicha carga.

Figura 7. Formatos que pueden ser cargados con *EEGLAB* (arriba) y con *FIMALOF* (abajo).



Fuente: Elaboración propia, basada en *EEGLAB*.

A manera de ilustrar el potencial de la interfaz, en la figura 8 se muestra una comparación de señales antes y después de filtrar. En la parte de arriba se muestran los resultados usando *EEGLAB*; en la parte de abajo, los resultados usando *FIMALOF*. En ambos casos se utilizaron 4 señales EEG correspondientes a los canales FC5, T7, T8 y FC6, del segundo 6 al 8, donde se observaron artefactos musculares. En el caso de *EEGLAB* se utilizó un filtro *FIR* de 0.1 a 30Hz. En el caso de *FIMALOF*, se eligió un filtro Butterworth, de cuarto orden, con una frecuencia de corte en 30Hz.

Mientras que en *EEGLAB* únicamente se visualizan las señales filtradas, *FIMALOF* permite observar, además, la respuesta en frecuencia del filtro aplicado y la densidad

espectral de potencia del canal seleccionado, proporcionando una visión más completa del proceso.

Conclusiones

Se desarrolló una interfaz gráfica de usuario para uso académico e investigativo implementada en *MATLAB*, intuitiva y de fácil manejo. Esta herramienta permite el filtrado de señales *EEG*, así como la identificación de fuentes bioeléctricas sintéticas que emulan a las fuentes reales. La generación de señales *EEG* sintéticas, obtenidas mediante un modelo matemático basado en un problema de contorno, representa una ventaja significativa, ya que permite validar tanto los filtros implementados como el algoritmo de identificación de fuentes, garantizando un control completo sobre las señales generadas. Además, la interfaz está disponible en español e inglés, lo que la hace accesible para un público más amplio. Incluye mensajes de ayuda en caso de que el usuario al ingresar parámetros, y su capacidad de combinar *ICA* con un filtro pasa-baja, mejora significativamente la eliminación de artefactos musculares en comparación con el uso exclusivo de *ICA*. Las comparaciones realizadas demuestran la funcionalidad y viabilidad de FIMALOF como una alternativa altamente comparable a *EEGLAB*.

Futuras Líneas de Investigación

Entre los desarrollos futuros se plantea la automatización en la eliminación de artefactos mediante *ICA* e *ICA-LPF*, identificando componentes contaminados de manera automática. Además, se espera implementar mapeos topográficos y cálculos estadísticos de señales EEG.

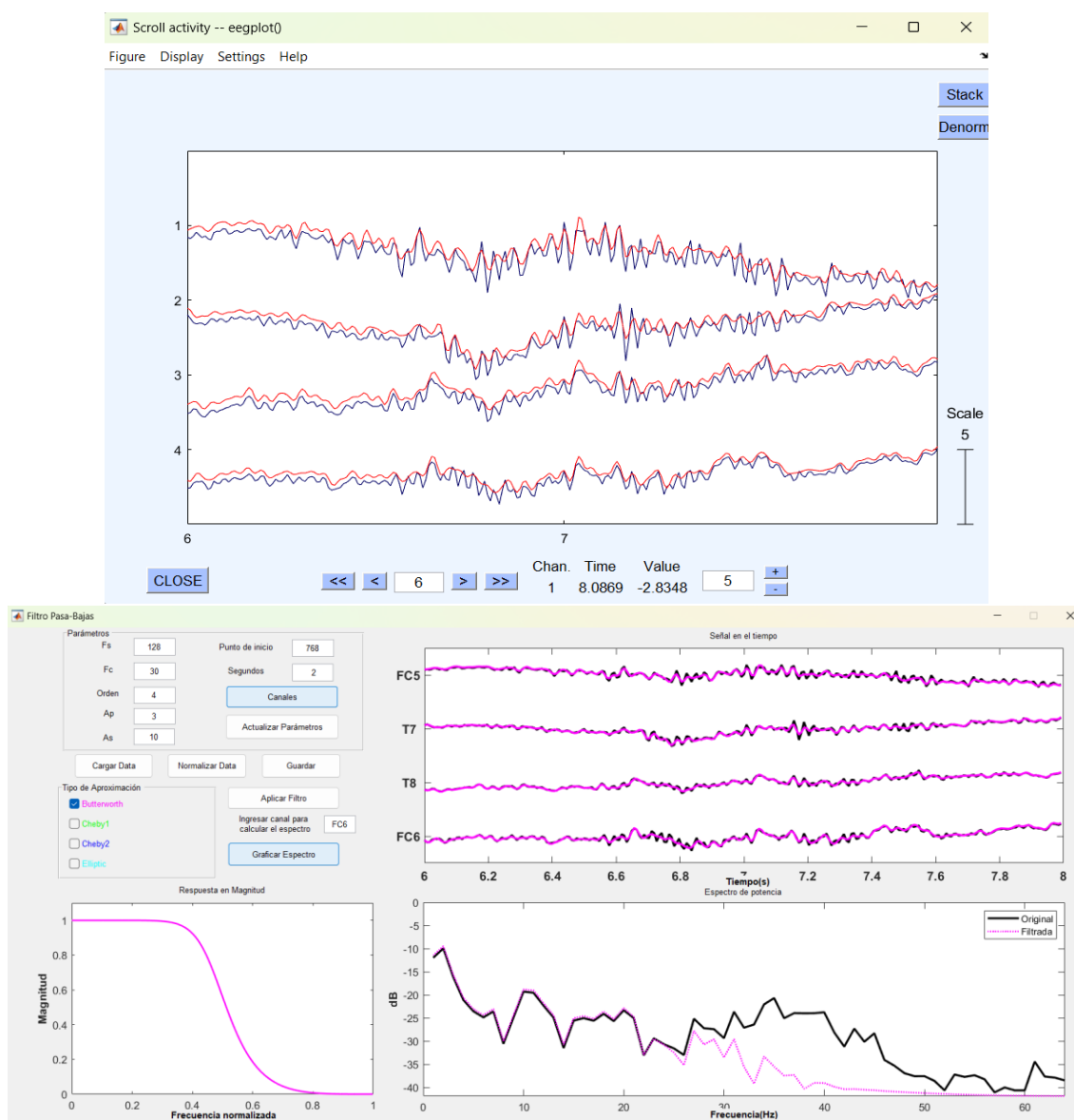
En cuanto a los modelos matemáticos, se busca emular otras enfermedades neurodegenerativas, generando bases de datos sintéticas para análisis y filtrado. Esto permitirá preservar la información completa de las señales EEG, incluyendo el ruido suprimido, útil para aplicaciones como el desarrollo de interfaces cerebro-computadora.

Entre las posibles mejoras, se encuentra la ampliación del modelo de identificación de fuentes, que actualmente opera en dos dimensiones. En futuros desarrollos, se planea incorporar un modelo basado en esferas concéntricas para mayor precisión.

Por otro lado, se planea desarrollar una versión ejecutable de la interfaz que pueda ser utilizada por médicos y especialistas biomédicos sin necesidad de conocimientos en

MATLAB. Asimismo, se está trabajando en la incorporación de técnicas de filtrado basadas en *wavelets*, que ofrecen una mejor resolución en tiempo-frecuencia y mejoran el análisis de señales *EEG*.

Figura 8. Arriba: Resultados de filtrar 4 señales *EEG* usando *EEGLAB*. Abajo: Resultados de filtrar 4 señales *EEG* usando *FIMALOF*. En ambos casos se utilizaron 4 señales *EEG* correspondientes a los canales FC5, T7, T8 y FC6, del segundo 6 al 8, donde se observaron artefactos musculares.



Fuente: Elaboración propia, basada en *EEGLAB*.

Referencias

- Compumedics Neuroscan, CURRY 9 X – Data Acquisition and Online Processing. Consultado el 30 de Septiembre de 2024. <https://compumedicsneuroscan.com/product/curry-9-x-data-acquisition-and-online-processing/>.
- Delorme, A y Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, *Journal of neuroscience methods*, 134(1), 9–21.
- EMOTIV, Epoc +. Consultado el 30 de Septiembre de 2024. <https://www.emotiv.com/epoc/>.
- Excellent Brain, Excellent brain EEG visualizer. Consultado el 30 de Septiembre de 2024. <https://excellent-brain.com/eeg-visualizer/>
- FieldTrip, Getting started with Loreta. Consultado el 30 de Septiembre de 2024. https://www.fieldtriptoolbox.org/getting_started/loreta/
- Huang, NE, Attoh-Okine, NO. (2005) *The Hilbert-Huang transform in engineering*. CrC Press.
- Hyvarinen, A, Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 12, 411-430
- Jiang, X., Bian, G. Bin, & Tian, Z. (2019). Removal of Artifacts from EEG Signals: A Review. *Sensors*, 19(5): 987, <https://doi.org/10.3390/S19050987>
- Kirsch A. (1996). *An introduction to the mathematical theory of inverse problems*. Springer Verlag
- Malvino A. P. (2000) *Principios de Electrónica*, Mc Graw Hill, 747 España.
- Morín M., Netzahualcóyotl C., Oliveros J., Conde J., & Santillán A. (2019). Identificación Estable de Fuentes asociadas a Focos Epilépticos ubicadas sobre la Corteza. *Revista Mexicana de Ingeniería Biomédica*, 40, 1-14, DOI: [dx.doi.org/10.17488/RMIB.40.3.5](https://doi.org/10.17488/RMIB.40.3.5).
- Nayak CS, Anilkumar AC (2024, 30 de Septiembre). EEG Normal Waveforms. In: *StatPearls. Treasure Island (FL): StatPearls Publishing*. <https://www.ncbi.nlm.nih.gov/books/NBK539805/>.
- Panayiotopoulos, CP, Benbadis, SR, Beran, RG, Berg, AT, Engel, J, Galanopoulou, Jr. AS, Kaplan, PW, Koutroumanidis, M, Moshe, SL, Nordli, DR, Serratosa, Jr., J.M., Sisodiya, S.M., Tatum, W.O. Valeta, T., Wilner, A. (2010). *Atlas of Epilepsies*. Springer-Verlag London, pp. 225-234.

- Santillán, A., Heute U., Stephani U., Galka A. (2017). Comparison of different methods to suppress muscle artifacts in EEG signals. *Signal, image, and Video Proc Journal*, 11, 761-593
- Santillán A., Ramírez H., Oliveros J., Morín M. & Ramírez-Díaz H. (2018). Interfaz Gráfica Intuitiva para el Procesamiento de Señales EEG. *Memorias Del Congreso Nacional De Ingeniería Biomédica*, 1, 146-149. <https://memorias.somib.org.mx/index.php/memorias/article/view/623>
- Santillán Guzmán, A., Morín Castillo MM., Rodríguez Domínguez, AM., Vargas García, JR., Oliveros Oliveros, JJ. (2021). Interfaz Gráfica de Usuario para la Generación de un EEG Sintético. *Memorias de la Vigésima Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI*, Orlando, Florida, USA.
- Swartz Center for Computational Neuroscience, What is eeglab? Consultado el 30 de Septiembre de 2024. <https://sccn.ucsd.edu/eeglab/index.php>.

Rol de Contribución	Autor (es)
Conceptualización	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual)
Metodología	Alina Santillán, Monserrat Morín, Jacobo Oliveros, Jesús Vargas (igual)
Software	Jacobo Oliveros, Jesús Vargas (igual).
Validación	Alina Santillán, Monserrat Morín, Jacobo Oliveros, Jesús Vargas (igual).
Análisis Formal	Monserrat Morín, Jacobo Oliveros (igual).
Investigación	Alina Santillán, Jacobo Oliveros (igual).
Recursos	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual).
Curación de datos	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual).
Escritura - Preparación del borrador original	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual).
Escritura - Revisión y edición	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual).
Visualización	Alina Santillán, Monserrat Morín, Jacobo Oliveros , Jesús Vargas (apoya).
Supervisión	Alina Santillán, Monserrat Morín, Jacobo Oliveros (igual).
Administración de Proyectos	Alina Santillán, Monserrat Morín, Jacobo Oliveros, Jesús Vargas (apoya).
Adquisición de fondos	Alina Santillán, Monserrat Morín, Jacobo Oliveros, Jesús Vargas (apoya).