

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**ADQUISICIÓN, ANÁLISIS Y PROCESAMIENTO DIGITAL DE
SEÑALES DEL CEREBRO (EEG)**

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRONICO

PRESENTADO POR:

MIGUEL ANGEL SOTA ZUÑIGA

**PROMOCIÓN
1986 - I**

**LIMA – PERÚ
2006**

**ADQUISICIÓN, ANÁLISIS Y PROCESAMIENTO DIGITAL DE SEÑALES DEL
CEREBRO (EEG)**

Dedico este trabajo a:
Mis padres, inspiración plena de lucha y
sacrificio,
Mi esposa e hijos, por su paciencia y
comprensión,
Mis hermanos, por el apoyo incondicional en mi
carrera,
Y mis sobrinos esperanza de superación.

SUMARIO

El presente trabajo pretende describir la adquisición, análisis y procesamiento digital de señales del cerebro (EEG), la cual está siendo utilizada por una gran diversidad de compañías, universidades e instituciones en el estudio e investigación de las enfermedades del cerebro, perturbaciones del sueño, desarrollo de equipos médicos de diagnóstico, así como dispositivos que puedan ser gobernado por estas ondas.

En el capítulo I se ofrece una descripción de las señales EEG, una reseña histórica de la evolución de los conceptos y del procesamiento de estas señales, los métodos de análisis actualmente utilizados y sus aplicaciones.

El capítulo II se refiere a las metodologías utilizadas en la adquisición de las señales EEG y sus alternativas de almacenamiento, los estándares utilizados en la nomenclatura de los electrodos y sus coordenadas, así como también las técnicas de análisis de estas señales.

El capítulo III se trata de las herramientas existentes en el mercado que permiten el análisis y procesamiento de estas señales, una descripción detallada de la herramienta EEGLAB, herramienta muy potente para ser utilizada en el ambiente de MatLab, que permite el análisis de muestras EEG grandes, en gráficos de 2D y 3D.

INDICE

PRÓLOGO	1
CAPÍTULO I	
DESCRIPCIÓN DE LAS SEÑALES EEG	2
1.1. Introducción	2
1.2. Antecedentes de estudios de Señales EEG en el Perú	3
1.3. Antecedentes de estudios de Señales EEG en el Mundo	3
1.4. Bases teóricas	5
CAPÍTULO II	
ADQUISICION Y ANALISIS DE LAS SEÑALES EEG	33
2.1. Adquisición de las señales electroencefalográfica	33
2.2. Análisis de las señales EEG	45
CAPÍTULO III	
HERRAMIENTAS DE PROCESAMIENTO	57
3.1. Herramientas de procesamiento EEG	57
3.2. Uso de Matlab en análisis de EEG	58
3.3. La nueva herramienta EEGLAB	59
CONCLUSIONES Y RECOMENDACIONES	193
CONCLUSIONES	193
RECOMENDACIONES	194
ANEXO A	198
BIBLIOGRAFÍA	277

PRÓLOGO

Desde que el hombre tiene uso de razón se ha interesado en conocerse a si mismo y conocer los problemas asociados a su comportamiento y enfermedades, realizando estudios e investigaciones al respecto, dentro de una de las mas importantes tenemos el estudio del funcionamiento del cerebro, utilizando para ello varias formas de mediciones, una de las cuales son las señales Electro Encéfalo Grama (EEG).

Entrando en mas detalle en este tema, podemos ver que, la primera grabación de un campo eléctrico de un cerebro humano fue realizado por el siquiatra alemán Hans Berger en 1924 en Jena (1). El dio a esta grabación el nombre de electroencefalograma (EEG).

Por otro lado, también podemos ver que la historia de la aplicación de la tecnología EEG se remonta por los años 60's, cuando aproximadamente el 60% de personas con incapacidades de aprendizaje reflejaron tener anomalías en la forma de onda de las señales EEG (2). Desde aquel tiempo, numerosos estudios han mostrado que la tecnología EEG es una medida precisa de lectura a tareas relativas al funcionamiento del cerebro. Es importante resaltar, que también se está estudiando las señales EEG para descubrir el pensamiento humano, reto en el que ya se han logrado algunos avances, beneficios de los cuales muy pronto estaremos disfrutando.

CAPÍTULO I

DESCRIPCIÓN DE LAS SEÑALES EEG

1.1. Introducción

El EEG (acrónimo de **E**lectro**E**ncefalo**G**rama), es una grabación (grama) de una señal eléctrica (electro) proveniente del cerebro (encéfalo). Son efectuados en gráficos sobre papel por lápices conectados a galvanómetros tales que leen las señales desde electrodos conectados al cuero cabelludo, estos electrodos no envían electricidad a la persona, por el contrario reciben señales eléctricas naturales generadas por el cerebro (1).

La actividad eléctrica cerebral genera señales eléctricas muy pequeñas que pueden ser amplificadas, y actualmente con el avance de la tecnología, se pueden representar en una computadora o registrar en una gráfica en papel en detalle.

La ubicación de los electrodos, generalmente unos 15 pares, están repartidos de forma estandarizada por todo el cuero cabelludo y siguen una denominación internacional para que los registros obtenidos sean comparables y reproducibles, independientemente del lugar y el momento. La comparación de las señales eléctricas entre distintos electrodos permite obtener numerosos trazados distintos, lo que va a permitir explorar la actividad cerebral desde distintos lugares, que va a servir incluso para estudiar determinadas zonas cerebrales específicas.

El electroencefalograma, representa una curva continua con oscilaciones continuas en los cuales uno puede distinguir ondas largas de primer orden con un promedio de duración de 90 milisegundos y ondas pequeñas de segundo orden de un promedio de duración de 35 milisegundos. La medición de desviación mayor esta entre 150 y 200 micro voltios (1).

Se observa en el electroencefalograma una concurrencia de fenómenos de continuos procesos de los nervios los cuales son efectuados en el cerebro, exactamente igual como

el electrocardiograma representa una concurrencia de fenómenos de la contracción de los segmentos individuales del corazón.

El electroencefalograma es una técnica de diagnóstico fácil de realizar, barata y no es intrusiva; ya que los electrodos se colocan sobre el cuero cabelludo. Si bien hace años su uso era enfocado en el diagnóstico de enfermedades del cerebro, actualmente se siguen utilizando en el estudio de enfermedades y su diagnóstico, en las que sirve también para el control de su evolución así como para la valoración de la eficacia del tratamiento, como por ejemplo: La epilepsia y el estado de coma.

1.2. Antecedentes de estudios de Señales EEG en el Perú

En el Perú, principalmente en la Pontifica Universidad Católica, consideran dentro de sus laboratorios un capítulo en el que tratan la Electroencefalografía, definiciones de electroencefalografía (EEG), tipos de ondas EEG, también modelos de EEG Pruebas con el BIOPAC (Sistema de Adquisición de señales biomédicas). Así mismo, también se están efectuando varios estudios, los cuales se están enfocando principalmente en:

- Resumir las técnicas existentes para el procesamiento digital de señales e imágenes vinculadas a la neuro-física y la neuro-ingeniería del EEG y los potenciales relacionados con eventos.
- Implementar mediante software dichas técnicas tomando en cuenta sus bondades conceptuales y de rendimiento en términos de velocidad y capacidad de muestreo, conversión analógico-digital, almacenamiento, recuperación y análisis de la información con software ya sea en línea o mediante post-procesamiento.

Por otro lado, también hay instituciones que están efectuando investigaciones sobre las señales EEG, por ejemplo el Instituto de Sueños – Hipnos, quienes conjuntamente con el apoyo de estudiantes de la Universidad Nacional de Ingeniería están efectuando investigaciones y desarrollos sobre estas señales; convirtiéndose en aportes importantes que permitirán ampliar el conocimiento y mejora en los diagnósticos de sus pacientes, así como también en su cura.

1.3. Antecedentes de estudios de Señales EEG en el Mundo

En el mundo, el estudio de las señales EEG esta en completa actividad, una diversidad de compañías, universidades e instituciones están involucrados en el desarrollo y estudio de estas señales, debido a que ayudará en diversos campos, tales como el diagnóstico y/o tratamiento de enfermedades como la esquizofrenia, la epilepsia, perturbaciones del

sueño, los cuales permitirán el mejoramiento en el tratamiento de estas enfermedades. Dentro de las muchas empresas dedicadas a estos desarrollos podemos citar a las empresas "Datex-Ohmeda", con sus estudios: "Entropía una aportación Revolucionaria a la anestesia adecuada", "Nociones Basicas sobre EEG y Potenciales de Evocados", etc. La empresa Bionic, Iberica S. A., empresa dedicada a la investigación de alta tecnología para el sector hospitalario y universitario, así como la industria farmacéutica., el cual ha desarrollado el producto software SCAN 4.0, cuyas funciones sirven como una Estacion de Trabajo para EEG y Potencial Evocados utilizado por numerosas universidades y laboratorios medicos de investigación y también en aplicaciones clinicas en hospitales y centros privados como Harvard, NIH, Baylor, Albert Einsten, Stanford, Instituto Beckman y el Instituto Max Planck de Alemania.

Por otro lado, también algunos estudios están orientados a la investigación del comportamiento de las señales del cerebro, las cuales han permitido descubrir determinados patrones, como por ejemplo la ejecución de determinada acción o comando que la persona ha efectuado, mover una mano, un dedo, de tal forma que en el futuro permitirá el desarrollo de elementos o equipos que permitan reemplazar alguna parte del cuerpo humano, o también puedan ejecutar alguna función. Dentro de las empresas que estan desarrollando estas investigaciones podemos citar al Instituto Tecnológico de Informatica – Universidad Politecnica de Valencia – España

Proyecto: Interfaz Cerebro Computadora, sistema que permitira a un usuario controlar un ordenador (u otro dispositivo) simplemente pensando en ello.

Otra institucion importante de mencionar en la investigación y desarrollo de las señales EEG es "Ragnar Granit Institute" la cual tiene una serie de proyecto como por ejemplo: Proyecto: NeuroAtlas, es el desarrollo de un paquete de software que contiene herramientas de procesamiento de imágenes para mostrar un modelo de toda la cabeza en imágenes digitales tridimensionales, en el cual se puede visualizar las señales del cerebro. Proyecto: "Laplacian brain surface potencial distribution", La superficie Laplaciana provee un estimado de la densidad de corriente radial a través del craneo en la superficie del cuero cabelludo. Proyecto: "Modeling of the human body as a volume conductor", efectuan el modelamiento con diferentes metodos, como segmentacion de imagenes, diferencia finita, con los cuales se grafican las diferentes señales EEG que se producen en el cerebro.

El mundo de investigación de las señales EEG esta en apogeo por lo que la lista de empresas e instituciones en el mundo que se estan dedicando es enorme.

1.4. Bases teóricas

1.4.1. Definición de términos

A continuación se presenta la terminología utilizada y una breve definición, las cuales nos ayudarán a entender los conceptos y temas tratados en el presente estudio.

- **ANALISIS BIESPECTRAL:** El análisis bi-espectral es un método de procesamiento de señales de alta complejidad. En la señal del EEG este método cuantifica el acoplamiento entre las fases de las distintas armónicas de frecuencia, es decir se correlacionan las fases o puntos de partida de los distintos componentes de frecuencia y se cuantifica su bi-coherencia y magnitud.
- **ANALISIS ESPECTRAL:** El análisis espectral de una señal consiste en estudiar en detalle el comportamiento de sus componentes armónicas en el dominio de la frecuencia.
- **ARCO REFLEJO:** Es el sustrato básico del Sistema Nervioso, es la respuesta inmediata que presenta un órgano (efector) ante un estímulo recibido por un receptor, generalmente, sólo actúa, si el estímulo está presente.
- **ARTIFACTS:** Contaminación de los datos EEG que ocurren durante el proceso de grabación.
- **AXON:** Es una parte de la neurona, la cual consiste en una fibra larga de nervio y su función es la de transferir las señales desde el cuerpo de la célula a otro nervio o a otro músculo de la célula.
- **BICOHERENCIA:** Bioespectro normalizado por el producto de tres espectros de segundo orden, siendo el bioespectro la transformada de Fourier bidimensional de la función acumulante.
- **CONTAMINACION ELECTRICA:** Disturbios ocasionados por señales eléctricas producidas interna o externamente, como por ejemplo el ruido que genera un bisturí o el disturbio que ocasiona las señales producidas por el ritmo cardiaco.
- **EEG:** El ElectroEncefaloGramma (EEG) es el registro de la actividad eléctrica de las neuronas que componen la capa superficial o capa de células piramidales de la corteza cerebral. Esta actividad eléctrica se genera por el intercambio iónico entre el citoplasma y el medio extracelular en las neuronas que constituyen esa zona del sistema nervioso central. Representa los potenciales eléctricos post sinápticos sincronizados, en las dendritas que están orientadas en la misma dirección, perpendicular a la superficie cortical (1). El voltaje producido debe atravesar las meninges y el líquido cefalorraquídeo, la calota craneal, tejido celular subcutáneo y

piel hasta llegar a los electrodos metálicos que captan la señal. Desde allí se filtra, amplifica y recoge para su análisis, en tiempo real o a posteriori.

- **ELECTRODO:** Un electrodo es un conductor utilizado para hacer contacto con una parte no metálica de un circuito, por ejemplo un semiconductor, un electrolito, el vacío (en una válvula termoiónica), un gas (en una lámpara de neón), etc.
La palabra fue establecida por el científico Michael Faraday y procede de las voces griegas *elektron*, que significa ámbar y de la que proviene la palabra electricidad y *hodos*, que significa camino. Por tanto, de una manera general, podemos decir que el electrodo es el extremo de un conductor en contacto con un medio, al que lleva o del que recibe una corriente eléctrica.
- **ELECTROENCEFALOGRAFIA:** Similar a EEG.
- **ENTROPIA:** La entropía puede ser entendida como un concepto físico que se relaciona con el grado de desorden de un sistema. En el análisis de señales la entropía cuantifica el grado de irregularidad presente en la señal. En un ejemplo simple, si una señal es perfectamente regular o predecible, su entropía es mínima. En caso opuesto, si la señal es extremadamente compleja, irregular o impredecible, su entropía es máxima. En general la señal del EEG cambia de patrones irregulares a más regulares cuando el paciente pierde conciencia tras el uso de drogas anestésicas.
- **EPOCH:** Fragmentos o segmentos correlativos de señales EEG, que pueden descomponerse en trenes de ondas que son los componentes básicos o fundamentales de la señal original.
- **EVENTOS BIOELECTRICOS:** Procesos electroquímicos del cerebro que generan impulsos bioeléctricos, reflejados en impulsos eléctricos que emite el cerebro a diferentes frecuencias y amplitudes.
- **FFT: Fast Fourier Transforms,** es la Transformada Rápida de Fourier (TRF).
- **ISOELECTRICO:** Forma de onda tendiente a ser lineal. Generalmente presente en el estado de coma o inerte.
- **MORFOLOGIAS DEL EEG:** Cambios de forma de las señales EEG debido a muchos factores. Los cambios que ocurren se deben naturalmente a cambios del estado de la conciencia, por ejemplo, despierto, dormido, etc.
- **NEURONA:** La neurona es la célula fundamental y básica del sistema nervioso, es una célula alargada, especializada en conducir impulsos nerviosos y esta formado por soma o cuerpo celular, dendritas y axón.
- **POTENCIAL DE EVOCADOS:** Técnicas de investigación neurofisiológica aplicada actualmente a muchos estudios del comportamiento humano, en la que se registra la

actividad eléctrica cerebral a través de los potenciales cerebrales asociados a estímulos específicos, más conocidos como potenciales evocados cognitivos o de larga latencia, provocados por la elaboración sensorial del individuo frente al estímulo, cuyas respuestas se relacionan con funciones cognitivas superiores, especialmente la atención.

- **POTENCIALES AUDITIVOS DE EVOCADOS (PAE):** Los PAE son señales electroencefalográficas producidas por estimulación auditiva controlada. Estos potenciales representan el paso de actividad eléctrica desde la cóclea a la corteza cerebral
- **SOMA:** Parte de la neurona que contiene el núcleo y los nucleolos, también se encuentran: los cuerpos de *Nissl*, que son aglomeraciones de retículo endoplasmático rugoso (responsable de la síntesis proteica), un aparato de Golgi prominente que empaqueta material en vesículas para su transporte a distintos lugares de la célula); numerosas mitocondrias y elementos cito esqueléticos (micro túbulos y micro filamentos).
- **SINAPSIS:** La relación existente entre el axón de una neurona y las dendritas de otra se llama "sinapsis". A través de la sinapsis, una neurona envía los impulsos de un mensaje desde su axón hasta las dendritas o un cuerpo de otra, transmitiéndole así la información nerviosa.
- **SINAPSIS EXITATORIA:** Potencial excitatorio post sináptico que ha producido la transmisión de un impulso por el axón, que es un cambio de permeabilidad, solo al sodio. En este caso el neurotransmisor se llama excitatorio.
- **SINAPSIS INHIBITORIA:** La inhibición sináptica ocurre en la unión nervio-nervio (neuro-neuro) cuando la actividad pre-sináptica libera un transmisor que hiper-polariza la membrana post-sináptica (ejemplo: hace el voltaje de la membrana más negativo).
- **TEORIA DEL CAOS:** es el estudio cualitativo del comportamiento dinámico periódico mostrado por sistemas deterministas no lineales.

1.4.2. Definición de conceptos

Para comprender el objetivo que se quiere alcanzar en el presente estudio es necesario tener claro los conceptos utilizados en cada uno de los capítulos, los cuales se presentan a continuación.

A) ESTRUCTURA Y FUNCIONAMIENTO DEL CEREBRO

a) ESTRUCTURA DE LA NEURONA

La neurona puede ser dividida en base a su estructura y funciones en tres partes principales (2):

- (1) El cuerpo de la neurona, también llamado el "Soma"
- (2) Numerosas prolongaciones cortas del soma, llamadas "Dendritas"
- (3) Una fibra de nervio largo, llamado "Axón".

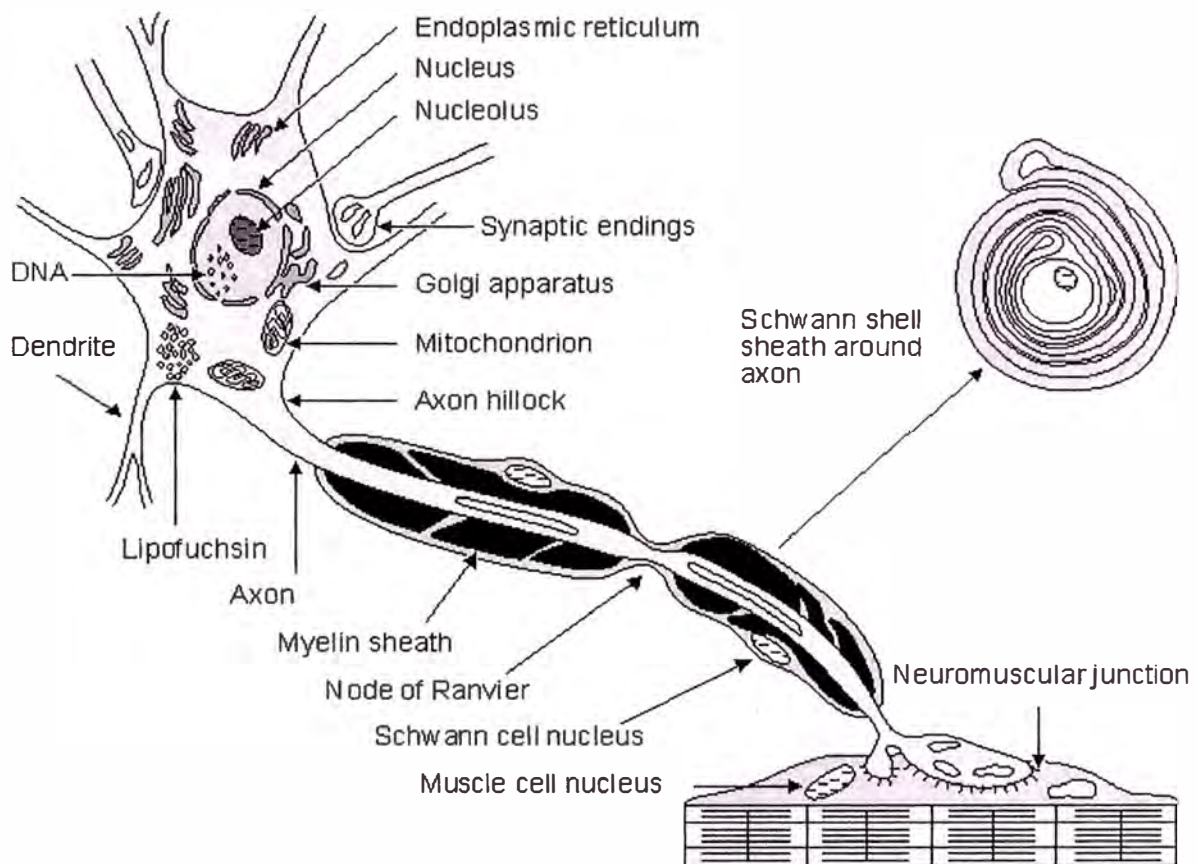


Fig. 1. 1 Detalles de Neurona

Fuente: *Oxford University Press* - 1995

El cuerpo de la neurona generalmente incluye el núcleo, mitocondria, retículo endoplasmático, ribosoma y otras estructuras, tal como se muestra en la Fig. 1.1. Las neuronas están compuestas por alrededor de 70 - 80% de agua, el resto de material sólido tiene cerca del 80% de proteínas y 20% de grasas, y tienen una variación en su volumen entre 600 y 70,000 μm^3

Las dendritas reciben impulsos desde otras células y transfieren estos impulsos hacia el cuerpo de la célula (señales que vienen de la parte externa de la neurona y van hacia el

centro de la misma). Los efectos de estos impulsos pueden ser excitatorios o inhibitorios. Una neurona cortical puede recibir impulsos de decenas o cientos de miles de neuronas (ver Fig. 1.2).

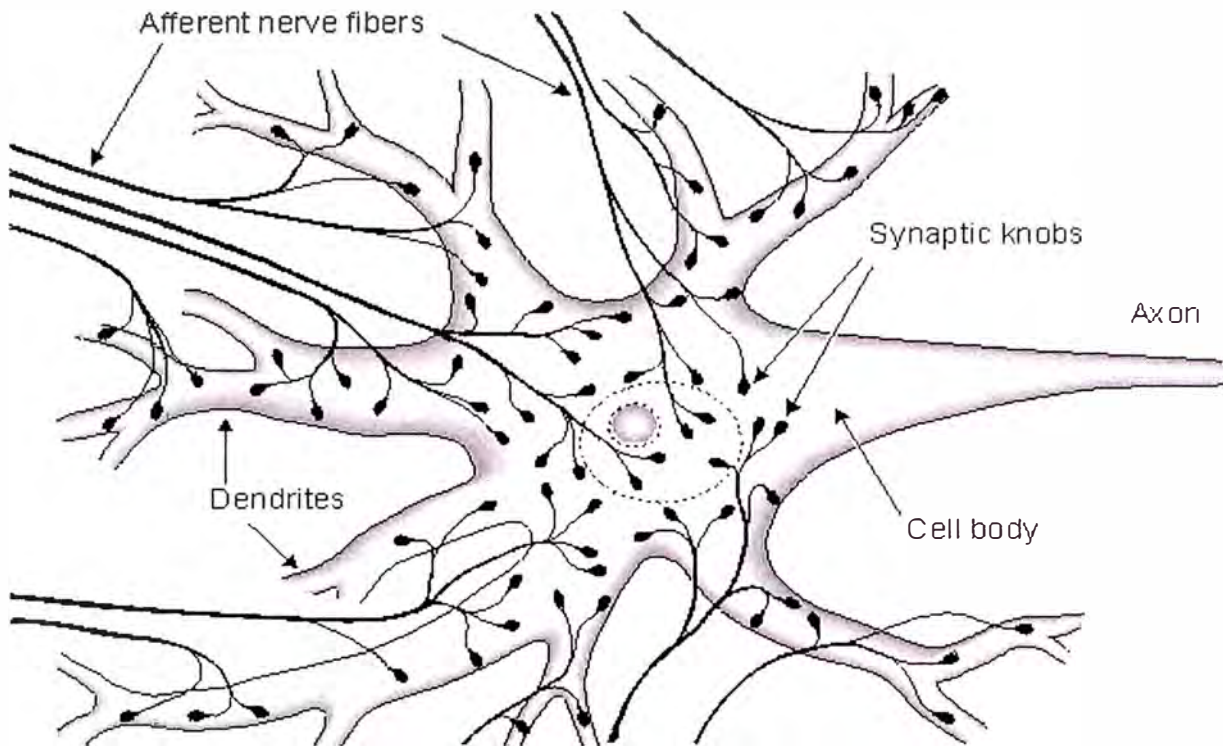


Fig. 1. 2 Célula de nervio cortical y sus conexiones con otras células

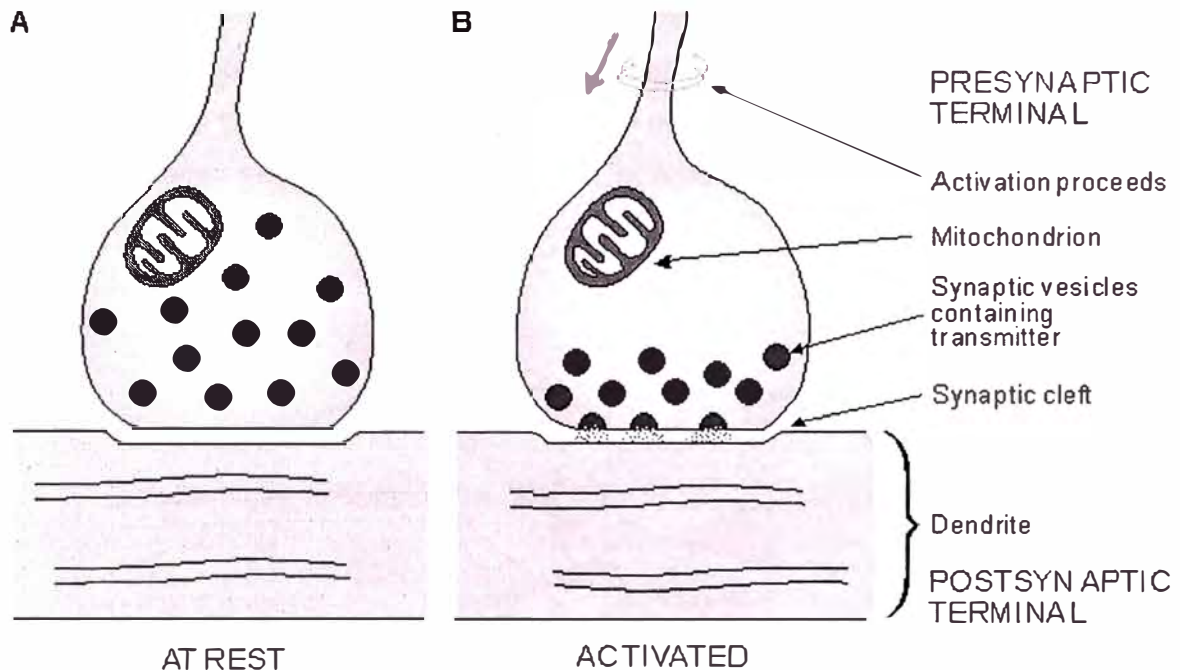
Fuente: *Oxford University Press* - 1995

El axón transfiere la señal desde el cuerpo de la célula a otro nervio o a otro músculo de la célula. En cuanto a su tamaño, en el caso del axón de los mamíferos esta en el orden de 1 - 20 μm de diámetro y en algunos animales grandes, puede ser de varios metros de longitud. Así mismo, el axón está protegido por una capa de aislamiento llamado el Sello de Myelin.

b) LA SINAPSIS

La unión entre un axón y la próxima célula con la cual se comunica es llamada sinapsis. La información proviene desde el cuerpo de la célula unidireccionalmente sobre la sinapsis, primero a lo largo del axón, y luego a través de la sinapsis al próximo nervio o célula muscular. La parte de la sinapsis que está en el lado del axón es llamada el extremo pre-sináptico (*presynaptic Terminal*), la parte del lado de la célula adyacente es llamado extremo post-sináptico (*postsynaptic terminal*). Entre estos extremos, existe una brecha, la hendidura sináptica (*synaptic cleft*), con un espesor de 10 a 50 nm. El hecho

que la transmisión del impulso a través de la sinapsis en una sola dirección, desde el extremo pre-sináptico al extremo post-sináptico, es debido a la liberación de un transmisor químico por la célula pre-sináptica. Este transmisor, cuando es liberado, activa el extremo pre-sináptico tal como se muestra en la Fig. 1.3. La sinapsis que estimula entre un nervio motor y el músculo es llamado la unión neuromuscular (2).



(A) La vejiga sináptica contiene un transmisor químico

(B) Cuando la activación alcanza al Terminal presináptico el transmisor es liberado y es difundido a través de la grieta sináptica al activar la membrana post sináptica.

Fig. 1. 3 Ilustración simplificada de la anatomía de la sinapsis

Fuente: Oxford University Press - 1995

• ESTRUCTURA Y FUNCION DE LA SINAPSIS

La función de la sinapsis es transferir actividad eléctrica (información) desde una célula hacia otra. La transferencia puede ser de nervio a nervio (*neuro-neuro*), o nervio a músculo (*neuro-myo*). La región entre la membrana pre- y post sináptica es muy angosta, solo 30-50 nm y es llamada la hendidura sináptica (*synaptic cleft* o *synaptic gap*). No ocurre comunicación eléctrica directa entre pre- y post-cruce de células, se emplea un mediador químico (ver Fig. 1.4). La secuencia de eventos es como sigue (3):

- 1) Un pulso llega al extremo final de la célula pre-sináptica.
- 2) Un neuro-transmisor es liberado, el cual se propaga a través de la hendidura sináptica uniéndose al receptor dentro de membranas especializadas de células post-sinápticas

3) El transmisor actúa abriendo canales de una o varias especies de iones, dando lugar a un cambio en el potencial de la trans-membrana. Si es despolarizando, es un potencial post-sináptico excitativo (EPSP), y si es hiper-polarizando, es un potencial post-sináptico inhibitor (IPSP).

En promedio, cada neurona realiza probablemente 1000 sinapsis. Por otro lado, una simple neurona motor vertebral puede tener un promedio de 10,000 entradas sinápticas. Basado en estos datos, no es sorpresa que el índice de sinápsis a neuronas en la parte frontal del cerebro humano es aproximadamente de 4×10^4 . En la sinapsis neuro-neuro, la post-unión puede ser una dendrita o el cuerpo de una célula, pero de forma predominante.

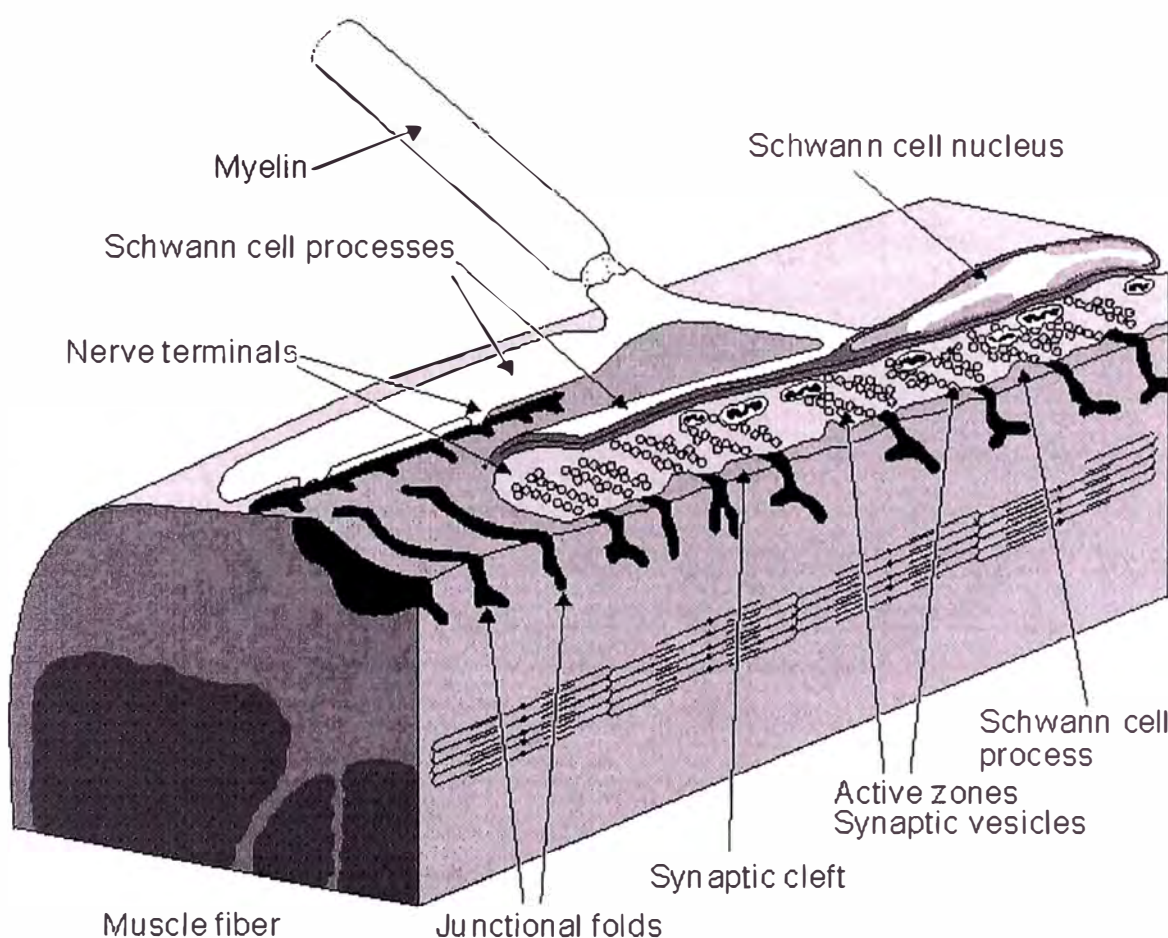


Fig. 1. 4 Unión sináptica neuro muscular

Fuente: *Oxford University Press* - 1995

Muchas características de estas uniones son también vistas en la sinapsis nervio-nervio. La finalización del Terminal de la celda pre-unión contiene muchas vejigas, las cuales son empaquetadas de *acetylcholine* (ACh) neurotransmisor. La abertura entre la membrana de pre y post unión esta en el orden de 15-30 nm. El transmisor es liberado por el arribo

de un impulso del nervio, difundida y unida a receptores en la membrana del músculo de post unión, brindando acerca de un EPSP y la iniciación del potencial de una acción muscular.

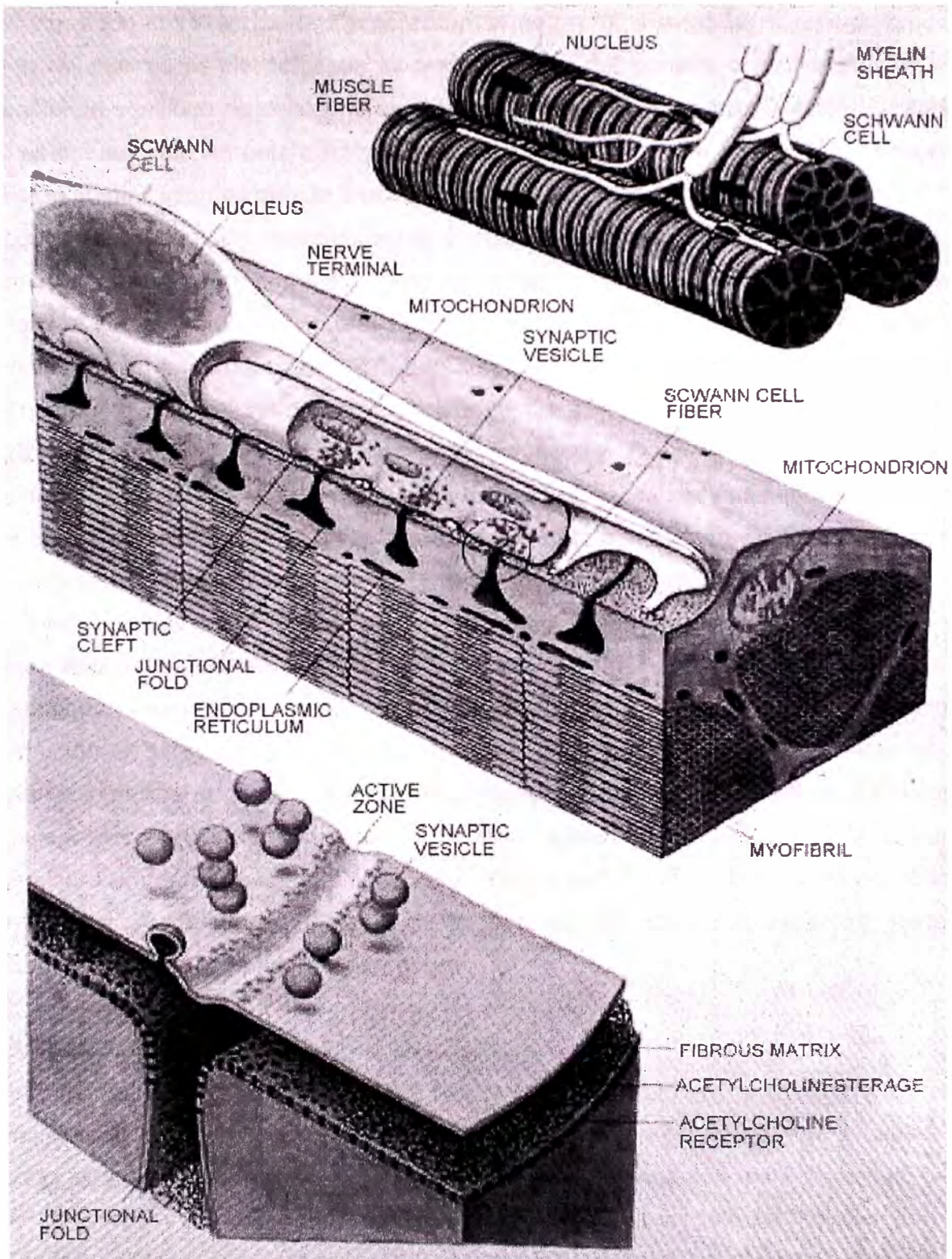


Fig. 1. 5 Corteza cerebral

Fuente: *Oxford University Press* – 1995

• SINAPSIS EXITATORIA Y INHIBITORIA

La inhibición sináptica ocurre en la unión nervio-nervio (*neuro-neuro*) cuando la actividad pre-sináptica libera un transmisor que hiper-polariza la membrana post-sináptica (ejemplo: hace el voltaje de la membrana más negativo). En teoría, la hiperpolarización puede resultar de la elevación de la permeabilidad del potasio o el cloruro porque el potencial de equilibrio de cada uno es más negativo que el potencial normal del resto (el cual es influenciado en polarización positiva por la presencia de sodio). En realidad, sin embargo, la inhibición es debido a la elevada permeabilidad del cloruro.

En contraste con la unión *neuro-muscular* (*neuro-myo*), una sola entrada excitatoria a una sinapsis *neuro-neuro* es completamente inadecuada para despolarizar la membrana de post-unión hasta el umbral. En realidad, con quizá miles de ambas entradas excitatorias e inhibitoras en la célula post-unión, una suma espacial y temporal esta tomando lugar continuamente, y el voltaje de la membrana fluctuará. Cuando finalmente, un umbral de 10-15 mV es alcanzada, resulta una potencial acción. De esta forma, tiene lugar un importante proceso integrativo a la entrada a cada célula nerviosa. Los lectores con experiencia en ciencias de la computación pueden apreciar las tremendas posibilidades para procesar información que puede (y hacen) tener lugar, particularmente cuando uno considera que hay por lo menos 10^{12} neuronas y 10^{15} sinapsis en el cerebro humano. Esto en realidad es una red neural (2).

La transmisión sináptica ha sido comparada a un circuito eléctrico de transferencia de información en la siguiente forma: En el nervio axón la información es transferida por medio de impulsos nerviosos en "digital" o, mas precisamente, en forma de "Modulación de pulsos codificados". En la sinapsis, la información es conducida con la sustancia transmisora en forma análoga, siendo convertida nuevamente en la próxima neurona en forma digital. Aunque esta analogía no es correcta en todos sus aspectos, ilustra el carácter de la cadena de información neural.

c) EL ARCO REFLEJO

El arco reflejo es considerado la unidad básica de la actividad neural integrada. Consiste esencialmente de un receptor sensorial, una neurona aferente, una o más sinapsis, una neurona eferente, y un músculo u otro ejecutor. La conexión entre el camino de un aferente y un eferente es encontrada, generalmente, en la médula espinal o el cerebro. Los reflejos simples involucra solamente a una sinapsis entre las neuronas aferente y el eferente (un reflejo monosináptico), un ejemplo familiar es el reflexivo movimiento automático del pie resultado de golpear la rodilla.

d) MODELO ELECTRICO DE LA SINAPSIS

En la unión neuromuscular, se muestra que el acetilcolino incrementa significativamente la permeabilidad de la membrana de la célula con pequeños iones, en la cual la conductividad del cloruro no fue afectada. ¿Qué pasa si la membrana se convierte igualmente permeable a los iones de sodio y potasio? Tal condición podría alterar el potencial de la membrana cerca al potencial *Nernst* de potasio que aproxima el promedio de equilibrio de potenciales de sodio y potasio (el potencial es cercano a un voltaje de cero en la transmembrana y es completamente adecuado para inicial una activación). Si la región *postsináptica* es un voltaje mantenido, el valor que reduce la corriente de la membrana a cero durante el flujo del transmisor es llamado voltaje de reversa V_r . Se puede mostrar que es igual al promedio de potencial *Nernst* de sodio y potasio, como se menciona líneas arriba. En la unión neuromuscular del músculo del esqueleto, el voltaje reversa es cerca de 15 mV.

El comportamiento eléctrico en una sinapsis puede ser estimado por el análisis de un circuito equivalente de la membrana **postsináptica**, tal como se muestra en la Fig. 1.6. Dos regiones son identificadas: Una representa la membrana asociada con los receptores sensitivos al transmisor, y el otro la membrana excitable normal de la celda. En la figura estas dos regiones son representadas por un elemento discreto, pero en realidad estos son distribuidos a lo largo de la estructura que constituye la celda actual. Esta figura representa una unión neuromuscular, donde la liberación de *acetylcholine* resulta en la elevación de la conductancia del sodio y potasio en la región objetivo, el cual es representado por el *switch Ach* cerrado. En el estado cerrado de este *switch* se tienen las siguientes ecuaciones (2):

$$\square I_{Na} = \square G_{Na}(V_m - V_{Na}) \quad (1.1)$$

$$\square I_K = \square G_K(V_m - V_K) \quad (1.2)$$

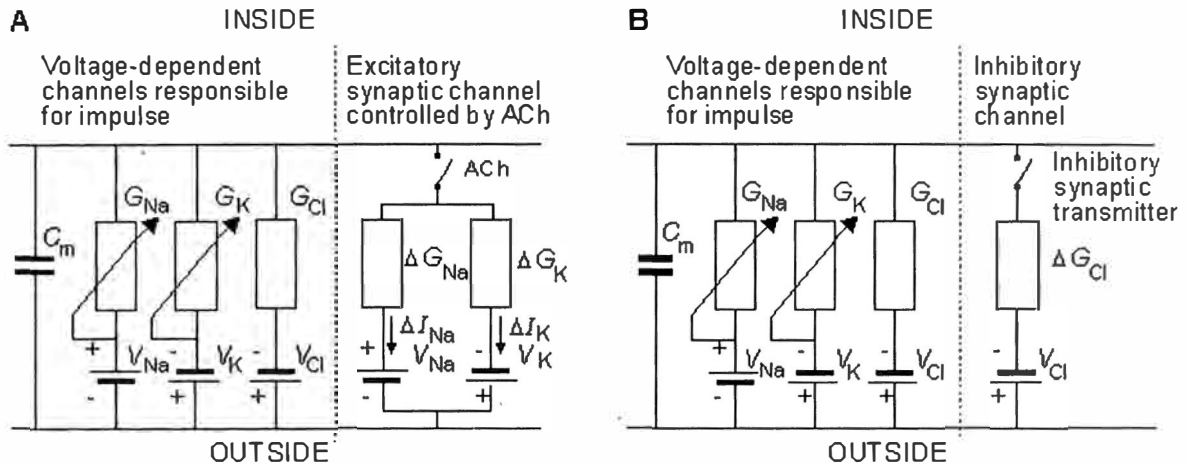
Donde:

I_{Na}, I_K = corriente de iones de sodio y potasio [$\mu A/cm^2$]

$\square G_{Na}, \square G_K$ = Conductancia adicional de sodio y potasio seguido por la activación por ACh (ejemplo: cercanamente igual a una alta conductancia) [mS/cm^2]

V_{Na}, V_K = Voltajes *Nernst* correspondientes a la concentración de sodio y potasio [mV]

= Voltaje de la membrana [mV]



Lado A: Modelo eléctrico de la celda post sináptica con sinapsis excitatoria (una unión neuromuscular es específicamente representada). La mayoría de las celdas esta limitada por una membrana excitable normal, como se describe en la izquierda. Además, una región sináptica especializada (final del plato) existe que es sensitiva a la transmisión química *ACh*. Cuando la *ACh* es liberada, difunde al lado del receptor en la membrana post unión, resultando en la apertura de las puertas del potasio y sodio. Este efecto es imitado en el modelo a través del cierre del switch, por tanto introduciendo la transmembrana alta de conductancia de potasio y sodio (ΔG_{Na} y ΔG_K).

Lado B: Modelo correspondiente a una sinapsis inhibitoria.

Fig. 1. 6 Modelo eléctrico de la celda post sináptica

Fuente: *Oxford University Press* – 1995

Si ahora introducimos y mantenemos el voltaje inverso (V_R) a través de la membrana post sináptica con un voltaje fijo, las ecuaciones (1.1) y (1.2) son reemplazadas por:

$$\square I_{Na} = \square G_{Na}(V_R - V_{Na}) \quad (1.3)$$

$$\square I_K = \square G_K(V_R - V_K) \quad (1.4)$$

Desde que el voltaje de la transmembrana V_m toma el valor V_R , el voltaje reversa.

Para las condiciones descritas por las ecuaciones (1.3) y (1.4), desde que la corriente total en el voltaje reversa es cero, se tiene que la corriente iónica del sodio y potasio son iguales y opuestos en signo (ejemplo: $\square I_{Na} = -\square I_K$). Consecuentemente aplicando esta condición la ecuación (1.3) y (1.4) se obtiene las ecuaciones siguientes:

$$\square G_{Na}(V_R - V_{Na}) = -\square G_K(V_R - V_K) \quad (1.5)$$

Recogiendo términos en la ecuación (1.5) tenemos

$$(\square G_{Na} + \square G_K) V_R = \square G_{Na} V_{Na} - \square G_K V_K \quad (1.6)$$

Y resolviendo para el voltaje reversa resulta en:

$$V_R = \frac{\Delta G_{Na} V_{Na} + \Delta G_K V_K}{\Delta G_{Na} + \Delta G_K} = \frac{\frac{\Delta G_{Na}}{\Delta G_K} V_{Na} + V_K}{\frac{\Delta G_{Na}}{\Delta G_K} + 1} \quad (1.7)$$

De la ecuación (1.7) es fácil observar que si la introducción de *ACh* causa un incremento igual en la conductancia del sodio y potasio, esto es, si:

$$\frac{\Delta G_{Na}}{\Delta G_K} = 1 \quad (1.8)$$

Entonces

$$V_R = \frac{V_{Na} + V_K}{2} \quad (1.9)$$

Como previamente se ha notado, para la unión neuro muscular el voltaje reversa tiene alrededor de -25 mV. En la practica, el voltaje reversa es un valor cercano a cero, lo cual significa que *ACh* incrementa la conductancia del sodio un poco mas que la conductancia del potasio. Es claro que el incremento de la conductancia del sodio y potasio puede ocurrir simultáneamente.

Retornando a la Fig. 1.6, y aplicando el teorema *Thevenin*, podemos simplificar el circuito receptor para consistenciarlo en una sola batería cuyo *emf* es el promedio de V_{Na} y V_K (de aquí V_R), y con una conductividad $g_R = g_{Na} + g_K$. Su efecto en la membrana normal de la celda post sináptica puede ser calculada desde que la corriente total en cualquier nodo es cero necesariamente – que es, hay corriente no aplicada. Consecuentemente:

$$G_R (V_m - V_R) + G_K (V_m - V_K) + G_{Na} (V_m - V_{Na}) = 0 \quad (1.10)$$

El camino del cloruro en la Fig. 1.6 no esta incluida en la ecuación (1.10), desde que $g_{Cl} \approx 0$, como notamos arriba, Resolviendo para el potencial post sináptico V_m resulta en:

$$V_m = \frac{G_R V_R + G_K V_K + G_{Na} V_{Na}}{G_R + G_K + G_{Na}} \quad (1.11)$$

La expresión es solamente aproximada desde que la membrana distribuida es representada por una membrana discreta. Además, si la membrana es llevada a un umbral cercano, entonces la representación del circuito lineal de la Fig. 1.6 se vuelve inválido. A pesar de todo, la ecuación (1.11) debería ser una medición útil si el potencial post sináptico es probablemente resultado de la excitación de la celda post sináptica.

B) OBTENCIÓN Y MEDICIONES DEL FUNCIONAMIENTO DEL CEREBRO

a) ELECTROENCELOGRAFIA

Las actividades eléctricas producidas en el cerebro y grabadas mediante electrodos son llamadas señales EEG, las cuales podemos clasificarlas por su estudio en las siguientes (4):

- (1) Actividades espontáneas
- (2) Potencial de evocados, y
- (3) Eventos bioeléctricos producidos por una neurona.

b) ACTIVIDAD EXPONTANEA

Es una medición efectuada en el cuero cabelludo o en el cerebro y es llamado el electroencefalograma. La amplitud de la señal EEG es de 100 μ V cuando se mide en el cuero cabelludo y alrededor de 1–2 mV cuando se mide en la superficie del cerebro. El ancho de banda de esta señal es desde 1 Hz hasta cerca de 50 Hz como se muestra en la Fig. 1.7. La frase “*actividad espontánea*” implica, que esta actividad es continua en la vida del individuo.

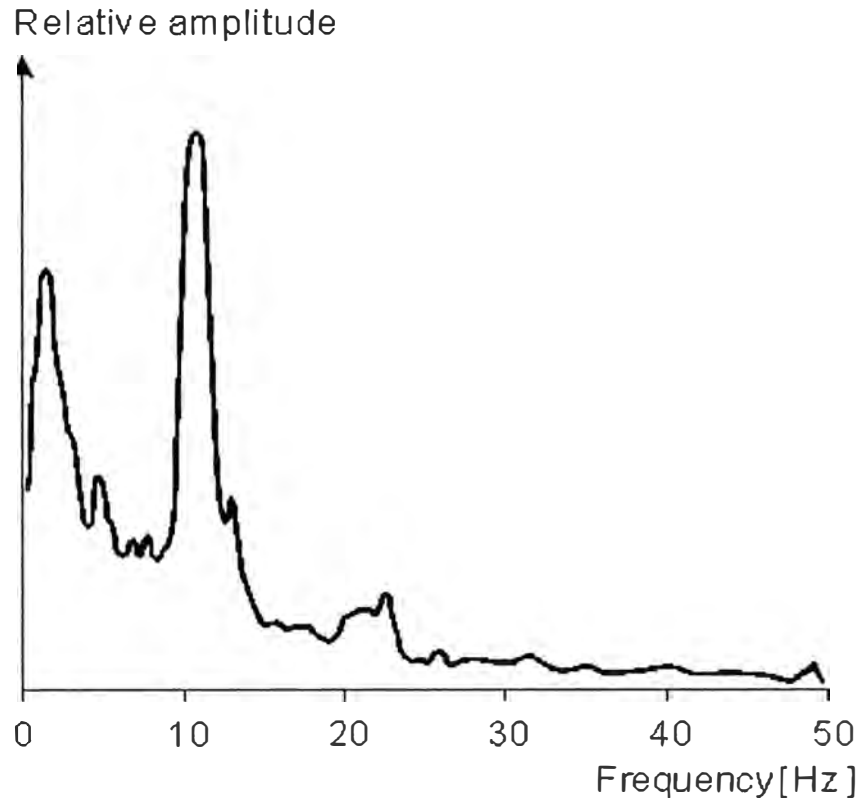


Fig. 1. 7 Espectro de frecuencia de un EEG normal

Fuente: *Oxford University Press – New York Oxford -1995*

c) POTENCIAL DE EVOCADOS

Son componentes de las señales EEG que se incrementan en respuesta a los estímulos (eléctricos, auditivos, visuales, etc.) Tales señales están generalmente debajo del nivel de ruido y además no son fácilmente distinguibles, y para ello uno puede usar una serie de estímulos y señales promedios para mejorar el índice de señal a ruido (6).

d) MEDIO DE ANALISIS DEL COMPORTAMIENTO DE UNA NEURONA

El comportamiento de una sola neurona puede ser analizado a través del uso de micro electrodos los cuales se pueden conectar a una sola célula de interés. A través de estudios de células individuales, uno espera construir modelos de redes de células que reflejen las propiedades de la red celular real.

e) SISTEMA PRINCIPAL DE LOCALIZACION DE ELECTRODO

Para grabar las señales EEG espontáneas se emplea generalmente el sistema internacional estandarizado 10-20. En este sistema 21 electrodos son localizados en la superficie del cuero cabelludo, como se muestra en la Fig. 1.8. Las posiciones son determinadas como sigue: Los puntos de referencia son la nariz, el cual es explorado en

su parte superior a nivel de los ojos, y el inion, el cual es un bulto de huesos en la base del cráneo en el medio de la parte posterior de la cabeza. Desde estos puntos, el perímetro del cráneo son medidos en planos transversos y medios.

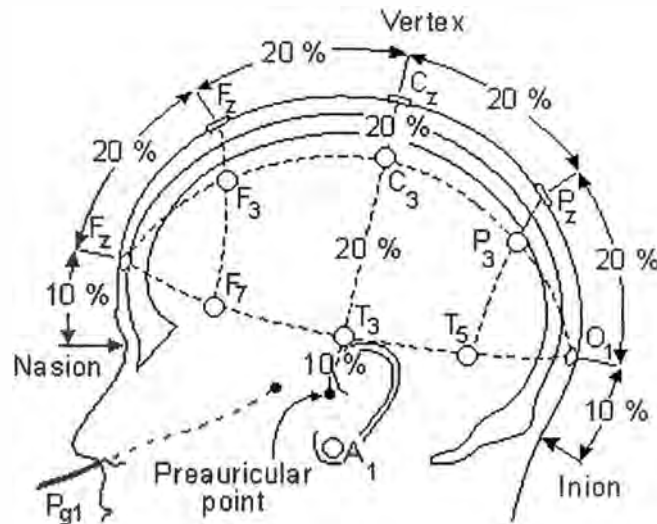


Fig. 1. 8 Sistema Internacional de electrodos 10-20: Lado Izquierdo

Fuente: *Oxford University Press – New York Oxford -1995*

La localización de los electrodos son determinados por la división de estos perímetros en intervalos de 10% y 20%. Otros tres electrodos son ubicados en cada lado equidistante desde los puntos adyacentes, tal como se muestra en la Fig. 1.9.

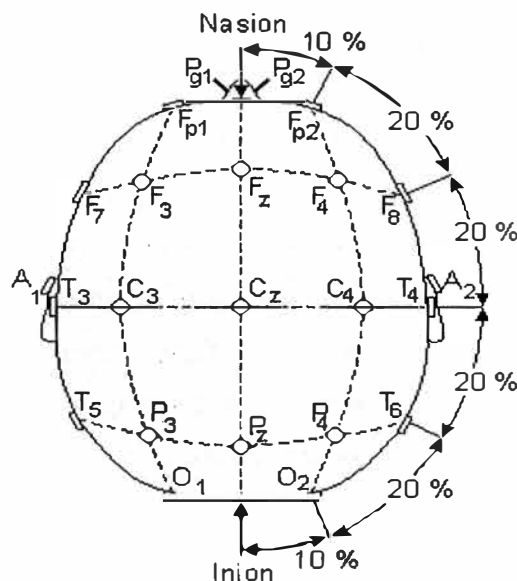


Fig. 1. 9 Sistema Internacional de electrodos 10-20 Encima de la Cabeza

Fuente: *Oxford University Press – New York Oxford -1995*

Además de los 21 electrodos del sistema internacional 10-20, 10% de las posiciones intermedias son también utilizadas. La localización y nomenclatura de estos electrodos son estandarizadas por la Sociedad Americana de Electroencefalografía. En esta recomendación, 4 electrodos tienen diferentes nombres comparados con el sistema 10-20; estos son: T7, T8, P7, y P8. Estos electrodos están dibujados en color negro con texto blanco en la Fig. 1.10.

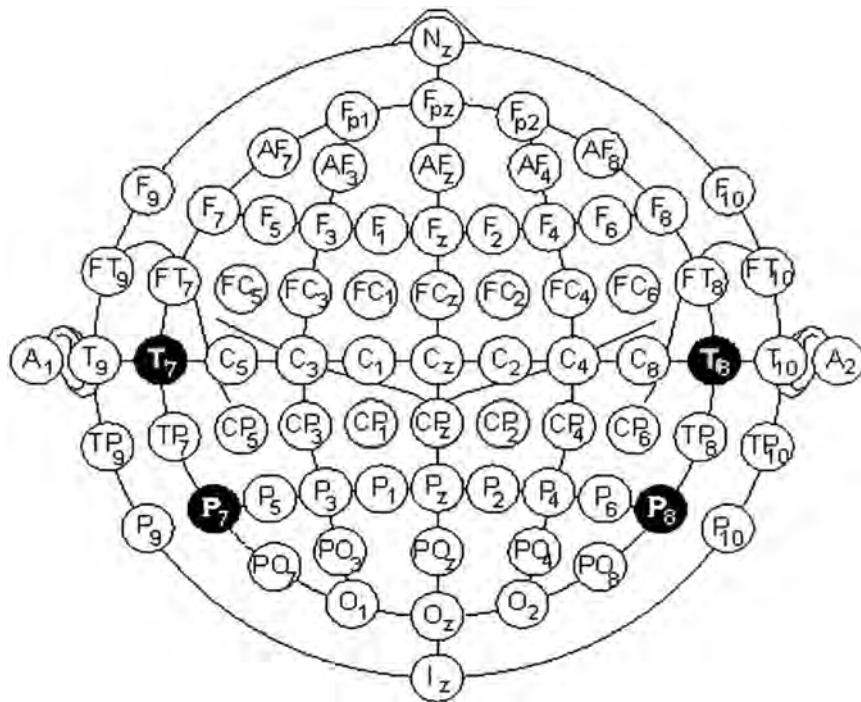


Fig. 1. 10 Localización y nomenclatura de los electrodos intermedios

Fuente: *Oxford University Press – New York Oxford -1995*

Además del sistema internacional 10-20, existe otros sistemas de electrodos para grabación del potencial eléctrico en el cuero cabelludo. El sistema "Queen Square" de ubicación de electrodos ha sido propuesto como un *Standard* en grabaciones de patrones de potenciales de evocados en pruebas clínicas.

Electrodos bipolares o unipolares pueden ser utilizados en mediciones de las señales EEG (ver Fig. 1.11). En el primer método la diferencia de potencial es medido entre un par de electrodos y en el segundo método, el potencial de cada electrodo es comparado con un electrodo neutral o con el promedio de todos los electrodos.

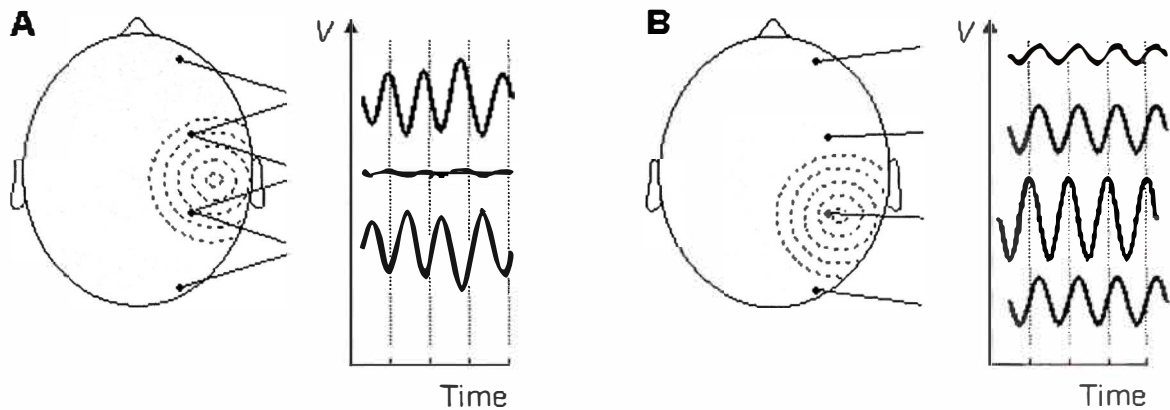


Fig. 1. 11 Mediciones A) Bipolar y B) Unipolar: Notar que las formas de ondas EEG depende de la posición de la medición

Fuente: *Oxford University Press – New York Oxford -1995*

f) TIPOS DE ELECTRODOS

Según las aplicaciones, existen diversos tipos de electrodos, según la forma (aguja, semicúpula, autoadhesivos) o el metal que lo compone (oro, cloruro de plata). Recientemente se ha desarrollado una nueva tecnología, los llamados Zipprep®, electrodos autoadhesivos similares a los empleados rutinariamente para electrocardiografía, capaces de mejorar la impedancia mediante la liberación de pasta conductora encapsulada dentro de micro vesículas que quedan en contacto con la piel, tras aplicar presión sobre el electrodo. La desventaja de los electrodos autoadhesivos es que sólo pueden colocarse en zonas donde no haya pelo, siendo la frente el área escogida en la práctica en la totalidad de los casos.

g) COMPORTAMIENTO DE LAS SEÑALES EEG

De la información que se recolecta en las señales EEG es posible diferenciar ondas tales como alfa (a), beta (b), delta (d), y teta (Q), un ejemplo de cada una de estas ondas podemos observar en el siguiente figura Fig. 1.12.

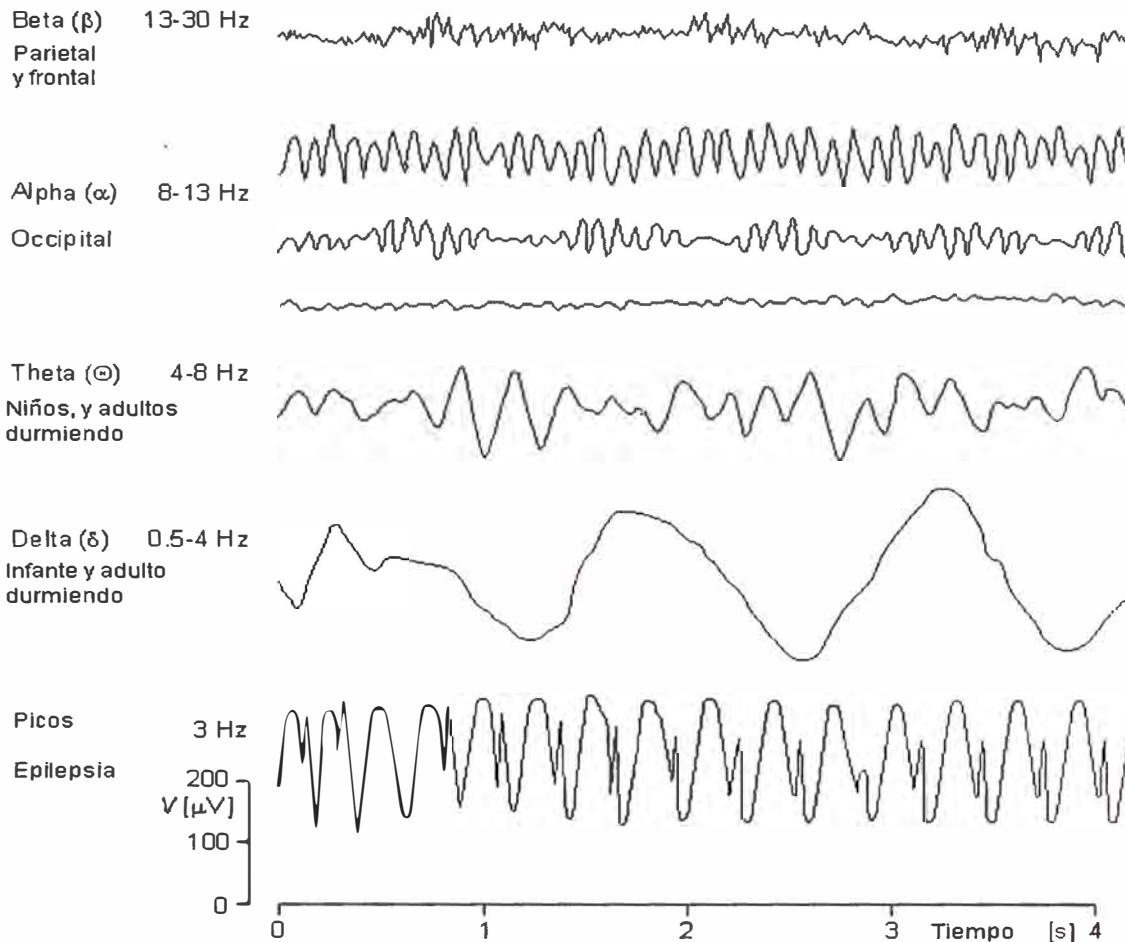


Fig. 1. 12 Algunos ejemplos de señales EEG

Fuente: *Oxford University Press – New York Oxford -1995*

Las ondas alfa (alpha) tienen el espectro de frecuencia de 8-13 Hz y pueden ser medidos en la región occipital en una persona despierta y con los ojos cerrados. La banda de frecuencia de las ondas beta es de 13-30 Hz; son detectables sobre los lóbulos parietal y frontal. Las ondas delta tienen el rango de frecuencia de 5-4 Hz y son detectables en infantes y en adultos dormidos. Las ondas theta tienen el rango de frecuencia de 4-8 Hz y son obtenidos de niños y de adultos dormidos.

h) LOS PRINCIPIOS BASICOS DE DIAGNOSIS CON SEÑALES EEG

La señal EEG es estrechamente relacionado al nivel de conciencia de la persona. Como la actividad incrementa, la señal EEG cambia a frecuencia dominante alta y amplitud baja. Cuando los ojos son cerrados, las ondas alfa empiezan a ser dominantes en las señales EEG. Cuando la persona cae dormida, decrementa la frecuencia EEG dominante. En una cierta fase de dormido, los movimientos rápidos de los ojos (REM) llama a dormir, la persona sueña y tiene movimientos activos de los ojos, los cuales pueden ser vistos como una característica de la señal EEG. En sueño profundo, las

señales EEG tiene grandes y pequeñas desviaciones llamados ondas delta. Puede ser detectada una completa ausencia de actividad cerebral de un paciente con muerte cerebral completa. Un ejemplo de las ondas indicadas podemos verlo en la Fig. 1.13.

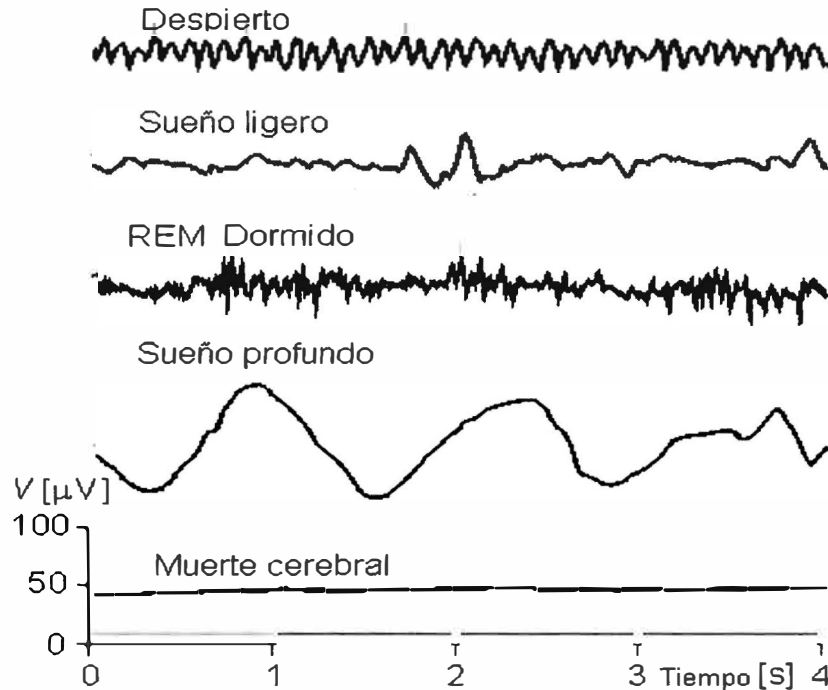


Fig. 1. 13 La actividad EEG depende del nivel de conciencia
Fuente: Oxford University Press – New York Oxford -1995

i) PRINCIPIO DE MEDICION

El EEG se mide básicamente como una diferencia de voltaje entre dos electrodos (7), detalle del cual podemos observar en la Fig. 1.14.

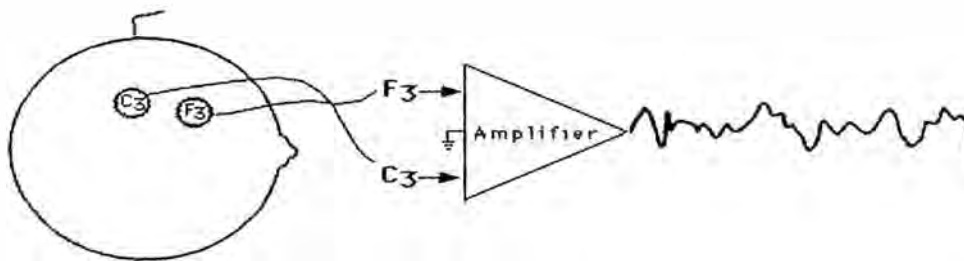


Fig. 1. 14 Medición del EEG: Amplificador diferencial
Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

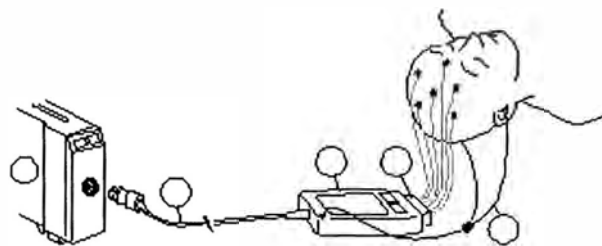


Fig. 1. 15 Medición del EEG

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

La amplitud de las señales EEG registrada desde la superficie del cerebro (cuero cabelludo) (ver Fig. 1.15), comparativamente, están alrededor de una décima de la amplitud de las señales del ElectroCardioGramma (ECG), dificultando su medición en un ambiente ruidoso como un quirófano o una Unidad de Cuidados Intensivos. A continuación en las figuras Fig. 1.16 al Fig. 1.20 se observa las formas de adquisición de las señales EEG (13).

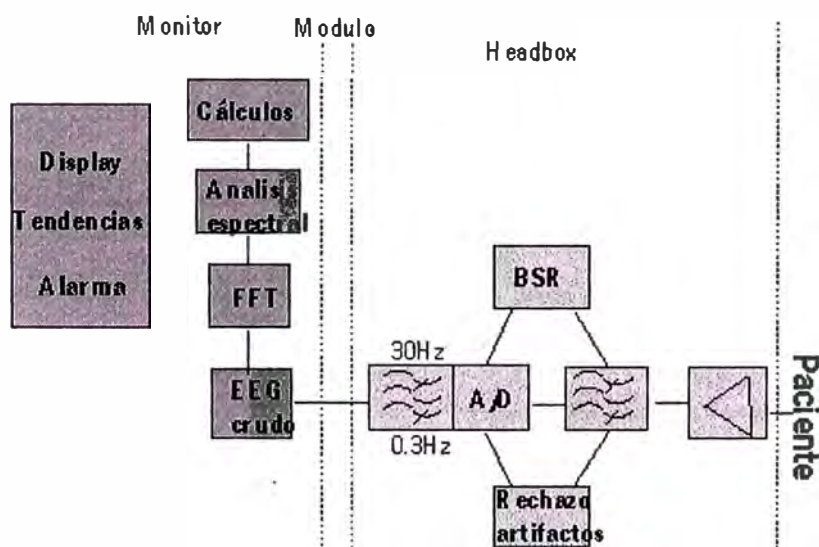


Fig. 1. 16 Diagrama general de bloques de equipo de adquisición de señales EEG (Tecnología Datex – Ohmeda)

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

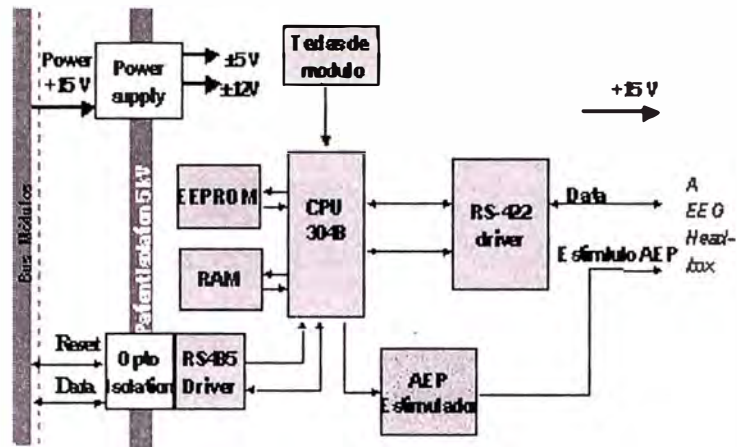


Fig. 1. 17 Estructura de módulos

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

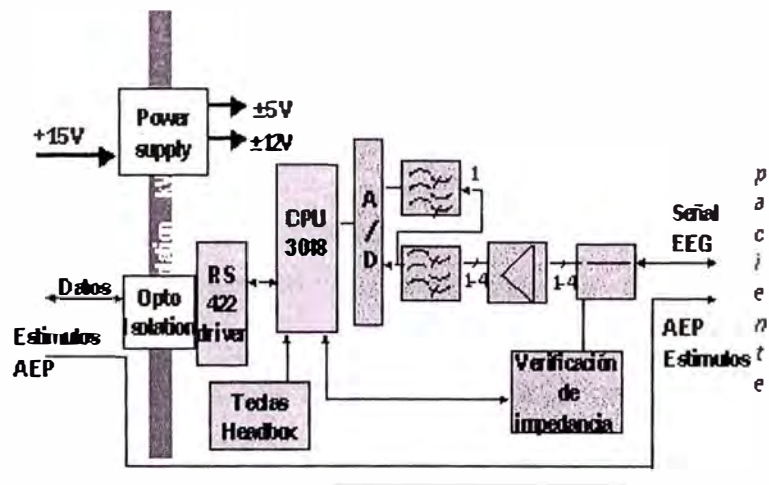


Fig. 1. 18 Arquitectura

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

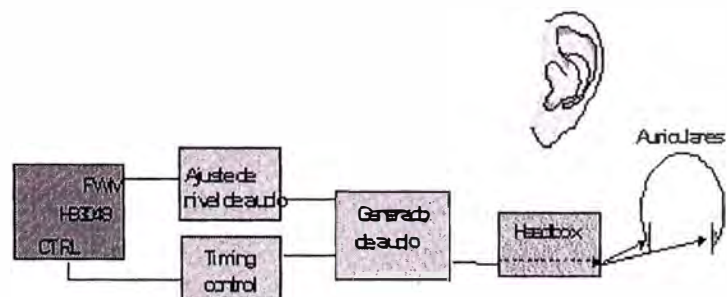


Fig. 1. 19 Circuito de estimulación de los potencial de evocados

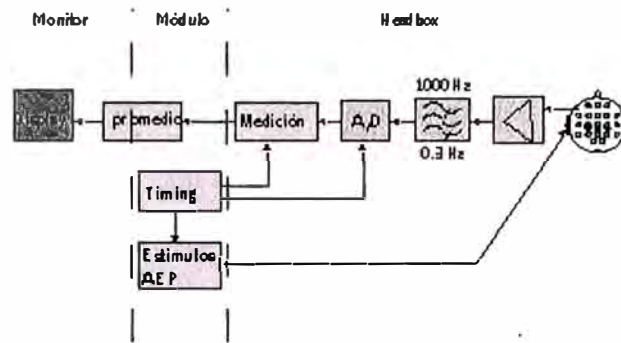


Fig. 1. 20 Diagrama de bloques de medición de los Potenciales de Evocados (AEP)

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

j) CONSIDERACIONES PARA UNA BUENA MEDICIÓN DE SEÑALES EEG

Para poder medir una señal EEG, la cual generalmente tiene valores muy pequeños ($100 \mu V$), hay que asegurar que el contacto entre el electrodo y la piel sea muy bueno, para lo cual, se efectúa un tratamiento preparatorio a la piel, que consiste en eliminar de su superficie la grasa y las células muertas. Así mismo, se puede utilizar un gel o una pasta conductora para mejorar dicho contacto.

La evaluación del contacto, se efectúa midiendo la impedancia entre el electrodo y la piel, teniendo en cuenta que para obtener buenos resultados, la impedancia no debe ser superior a 5 kilo-Ohmios.

Luego de efectuado y verificado que se tiene un buen contacto, las señales analógicas son recogidas desde el paciente para ser amplificadas, filtradas, digitalizadas, procesadas y enviadas al dispositivo de salida (monitor) donde se puede visualizar.

k) INTERPRETACIÓN DEL EEG

De lo experimentado se puede observar que el registro en tiempo real de la actividad eléctrica cerebral, a simple vista, carece de patrón reproducible, asemejando el registro a una señal "caótica". Para una mejor interpretación podemos comparar el proceso efectuado a la señal de EEG en tiempo real la cual es sometida a un análisis espectral utilizando el algoritmo conocido por Transformada Rápida de Fourier (FFT), con el proceso similar al que sufre la luz cuando se la hace pasar a través de un prisma (ver Fig. 1.21), descomponiéndose esta en los colores fundamentales (7).

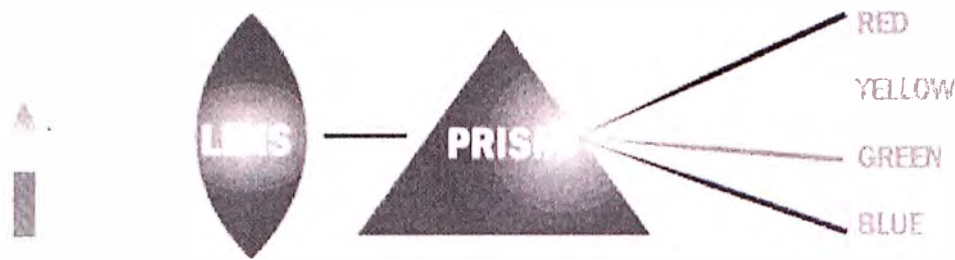


Fig. 1. 21 Proceso de descomposición de los colores

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

El análisis espectral de las señales EEG, las descompone en sus componentes fundamentales. Estos componentes, son señales senoidales puras, en frecuencias de una banda entre 0.5 Hz. y 30 Hz., así mismo se tiene también, sus correspondientes valores de Amplitud (o Potencia). Estos componentes, se agrupan en cuatro bandas (7), según su valor (ver Fig.1.22).



Fig. 1. 22 Señales EEG luego de un análisis espectral FFT

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

En la práctica, el EEG en tiempo real se digitaliza, a continuación se divide en trozos o "segmentos" de 2 segundos de duración y se aplica a cada "segmento" el análisis espectral. De esta forma se van obteniendo sucesivos espectros de potencia, que no son más que la representación de las frecuencias de los armónicos frente a sus amplitudes (o potencias) (ver Fig.1.23).

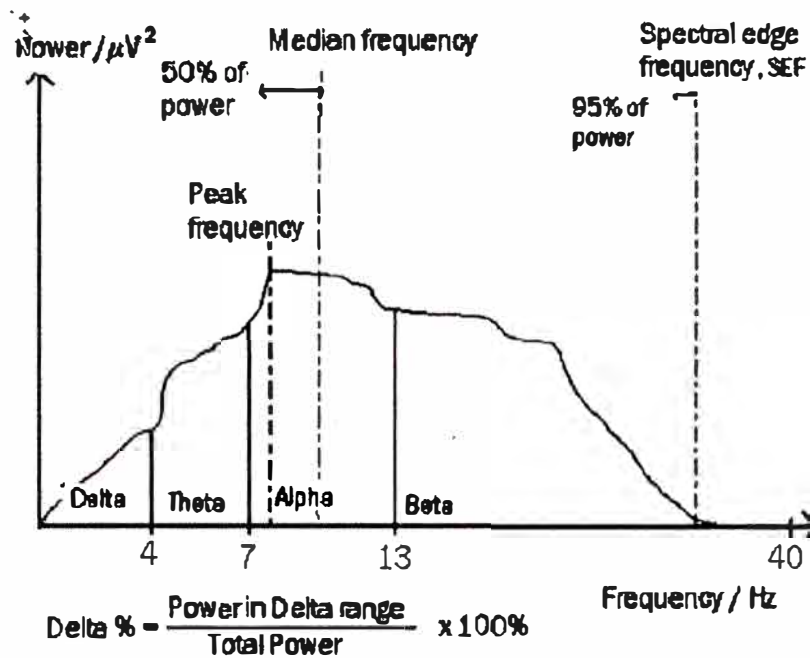


Fig. 1. 23 Espectro de frecuencia y parámetros espectrales

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

El análisis espectral proporciona además otros parámetros, como la Frecuencia Media (FM), Frecuencia de Corte (SEF 95) y Pico de Frecuencia, cuantificando de esta forma esa señal caótica que es el EEG y facilitando por consiguiente su interpretación (7).

I) DIFERENTES MORFOLOGÍAS DEL EEG

- **EEG en individuo despierto y activo**

Las señales EEG cambian de forma debido a muchos factores, principalmente debido a los cambios del estado de conciencia del cerebro: por ejemplo en el ser humano totalmente despierto que está concentrado en una tarea, el EEG tiene frecuencias más altas y el análisis espectral tiene un contenido en el rango BETA (13 Hz – 30 Hz) muy elevado (ver Fig 1.24).

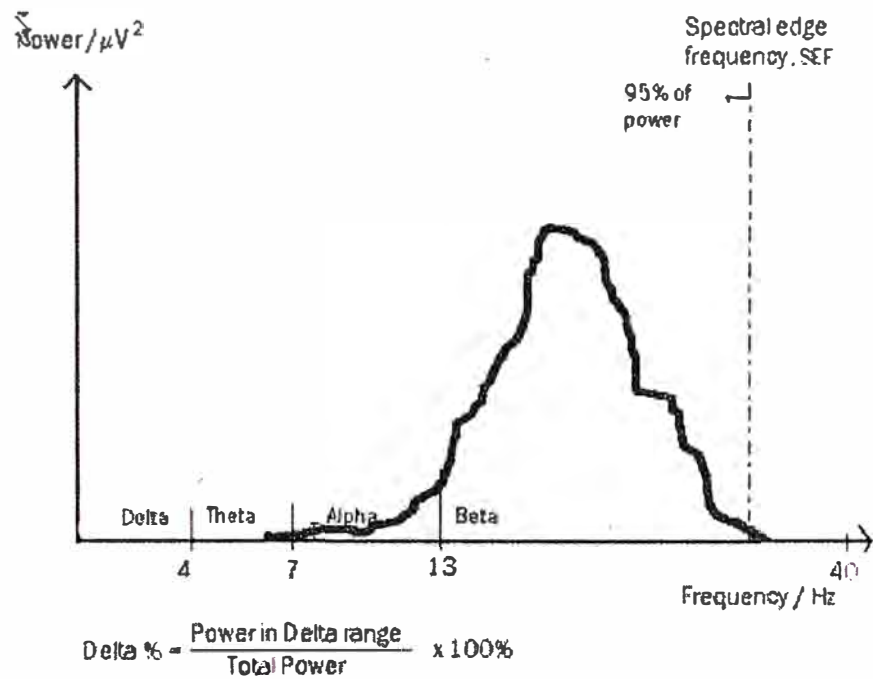
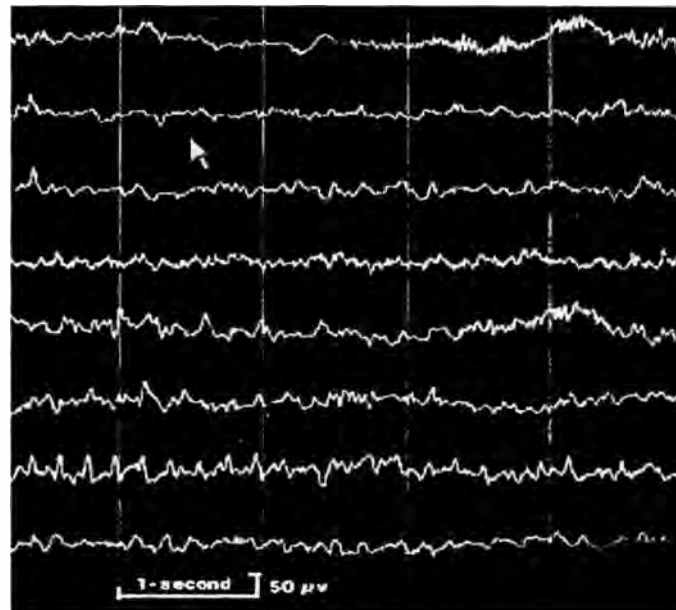


Fig. 1. 24 Registro de EEG en individuo despierto y activo
Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

- **EEG durante el sueño**

Por el contrario una persona dormida tiene unas señales EEG con un contenido en frecuencias muy bajo y su análisis espectral reflejaría una agrupación de frecuencias en el rango DELTA (0.5 Hz – 4Hz) (ver Fig. 1.25).

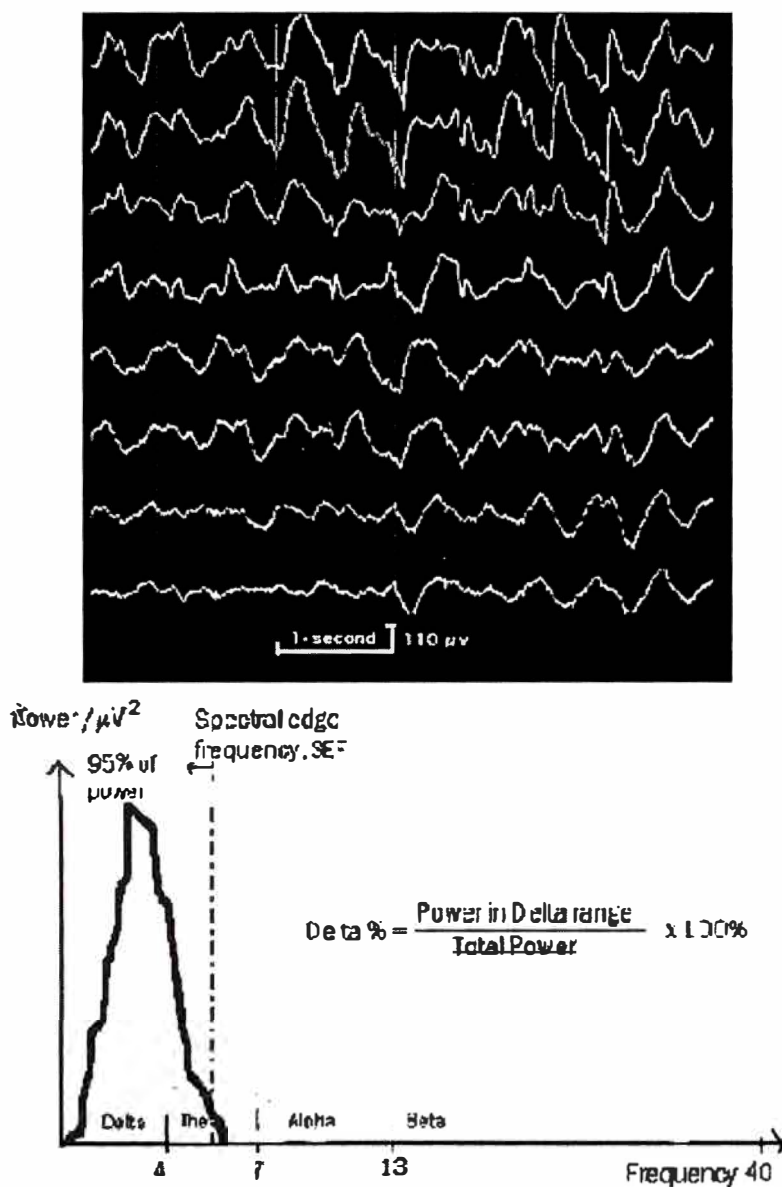


Fig. 1. 25 Registro de EEG durante el sueño

Fuente: Aula de Bioingeniería – Datex Ohmeda – España, 2001

m) EEG en procedimientos clínicos

Durante la cirugía, la mayoría de los cambios en el EEG son inducidos por la anestesia de forma similar al sueño natural, de esta forma cuanto más profunda es la anestesia, más lento es el EEG, pudiendo llegar en casos extremos, a un trazado de EEG plano (línea isoelectrica) si se aumenta el nivel de anestesia lo suficientemente. Dicho EEG plano indica ausencia total de actividad sináptica neuronal, recuperando la normalidad si disminuimos el nivel hipnótico de la anestesia (7).

Cambios similares son causados también por una oxigenación cerebral insuficiente (situación de hipoxia que puede derivar en isquemia). Si el flujo de sangre al cerebro disminuye o se detiene por alguna razón, la consecuencia es una rápida disminución de

la frecuencia del EEG, con un contenido en frecuencias de baja magnitud (ondas DELTA).

Un tercer factor que puede disminuir la frecuencia de las señales EEG es la temperatura corporal. En la cirugía cardíaca, el EEG del paciente es llevado intencionadamente a un estado plano isoelectrico, mediante enfriamiento (hipotermia provocada). Esto se hace porque cuando el EEG es plano, situación que representa ausencia de actividad sináptica neuronal, el cerebro también consume muy poco oxígeno, por lo tanto, se puede cortar la circulación al cerebro por un momento.

Una razón para admitir al paciente en la Unidad de Cuidados Intensivos son los ataques epilépticos, que aparecen como picos u ondas de gran amplitud en el EEG sin signos clínicos externos. Los ataques continuos dañan el cerebro, y es por esta razón que se suprimen con fármacos. Sin una adecuada monitorización del EEG, estos ataques o episodios epilépticos pasarían desapercibidos.

CAPÍTULO II

ADQUISICION Y ANALISIS DE LAS SEÑALES EEG

2.1. Adquisición de las señales electroencefalográfica

2.1.1. Recolección y almacenamiento de la señal

Las señales EEG, tal como se ha mencionado, se recoge y registra mediante la colocación de electrodos en ciertos puntos de la superficie del cuero cabelludo pre-establecidos según las necesidades y de acuerdo con el sistema Internacional 10/20. Según donde se sitúen los electrodos, el trazado electroencefalográfico puede variar, por tanto, es importante que para poder establecer comparaciones, las posiciones empleadas deberán ser las mismas.

El electrodo debe ser estable tanto mecánica como eléctricamente, presentar distorsión mínima y una impedancia de contacto lo más baja posible. Para su colocación debe tenerse en cuenta el objetivo de la monitorización, considerando que los nuevos sistemas de electrodos autoadhesivos con baja impedancia han facilitado la extensión del empleo del EEG, pues permiten la recogida de la señal desde áreas de simple colocación como la frontal con máximas garantías, en contraposición a la complejidad de colocar electrodos en las zonas occipital y parietal. La señal que se recoge en forma de voltaje a nivel de los electrodos, es transmitida bien sea en forma analógica como voltaje, tal es el caso de los monitores clásicos de EEG, o se digitaliza mediante sistemas de transformación analógica a digital, y en forma digital se transmite al monitor o medio de monitoreo (6).

Es indispensable realizar una calibración de la señal, que suele realizarse automáticamente por la mayoría de sistemas de EEG, para contrastar el voltaje proveniente del córtex con una fuente de voltaje eléctrico conocido. La amplitud de la señal del EEG es escasa, aproximadamente 100 veces menor que, por ejemplo, la señal del electrocardiograma. Este dato es relevante pues la contaminación eléctrica desde el

propio organismo o externa a éste, puede afectar a la pureza de la señal del EEG. En consecuencia, los cálculos que se hagan sobre esta señal, no serán fiables. La Tabla 2.1 muestra los principales tipos de *artifacts* que pueden alterar el trazado de la señal del EEG (11).

Tabla 2. 1 Principales tipos de *artifacts* que pueden alterar el trazado de las señales EEG

<i>Tipo</i>	<i>Definición</i>
Electromiograma	Ondas de frecuencia superior a 30 Hz debidas a movimientos de las fibras musculares cercanas
Electrocardiograma	Presencia de picos periódicos en el trazado de EEG no procesado con una frecuencia idéntica a la de los complejos del EKG
Deglución o parpadeo	Provoca oscilaciones de la línea de base del EEG
Movimiento	Provoca oscilaciones de la línea de base del EEG con una amplitud que excede el rango del sistema
Actividad eléctrica externa	Descargas periódicas que se relacionan con el empleo de bisturi eléctrico
Toma de corriente	La onda de EEG presenta un trazo grueso uniforme
Triboeléctrico	Por presencia de algún sistema eléctrico al producirse deformación del aislante que recubre al conductor. Es típico cuando se emplea bombas rotativas para la circulación extracorpórea

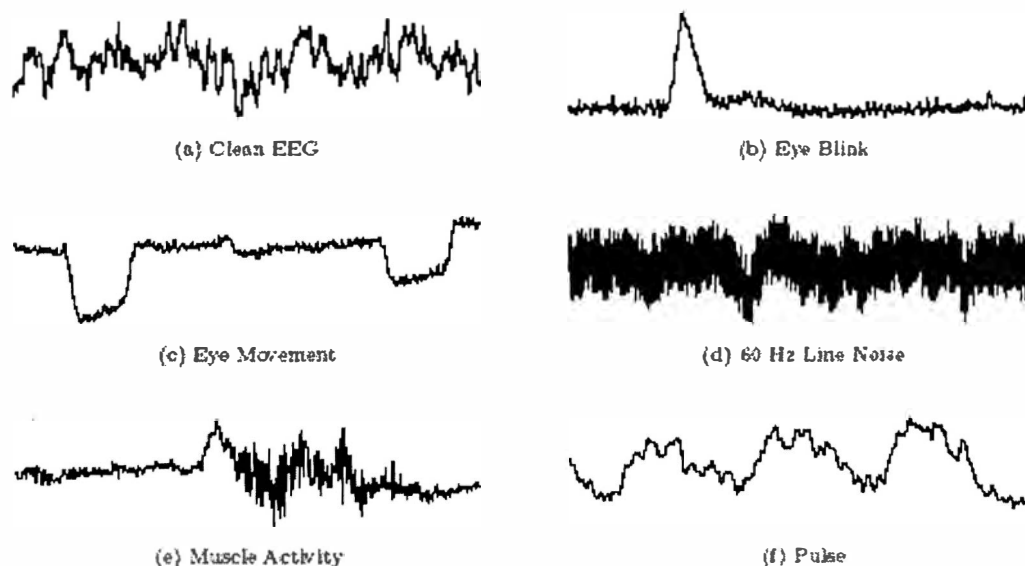


Fig. 2. 1 Señales Artifacts

Fuente: Colorado State University, Fort Collins, Colorado Fall, 2003

2.1.2. Formatos de datos EEG

Luego de adquiridas las señales EEG, es necesario almacenarlos en medios informáticos masivos y de alta capacidad (por el volumen de esta información), y la principal característica para su desarrollo entre muchos fabricantes de equipamientos clínicos y/o de laboratorio, desarrolladores e investigadores, es tener formatos estándares, con la finalidad de interrelacionar la adquisición de datos con los procesamientos.

a) Formato de archivos

Dentro de los formatos estándares de archivos de señales EEG (16) tenemos los siguientes:

Archivos texto.- Pueden ser descritos como cualquier archivo que contiene texto y es almacenado en el código estándar "*American Standards Code for Information Interchange*" (*ASCII*) (17). La información contenida puede ser colocada en cualquier formato, pero en términos de colección de señales, los datos son frecuentemente configurados en columnas, con la primera columna que representa el intervalo de tiempo. Otras columnas de datos encontrados en archivos texto frecuentemente son lecturas de voltaje de varias señales. Estas lecturas y mediciones pueden variar de señal a señal, pero de forma general son generalmente consistentes. Los archivos textos usualmente son mostrados a través del uso de programas simples tales como el *Notepad*, o el *Wordpad*. La extensión comúnmente asociado con estos archivos es *.txt*, sin embargo, no esta limitado a esta extensión, utilizan también la extensión *.doc*.

Formato EDF.- Este formato fue publicado en 1992 como "Un formato simple para intercambio de grabaciones poligráficas digitalizadas" en Electroencefalografía y Neurofisiología Clínica. Este formato es bastante parecido al archivo de texto estándar, excepto que para aprovechar su potencial con un programa conversor o visor es necesario leer la información. Sobretudo este formato ha sido utilizado para almacenar múltiples señales que tienen un elemento de tiempo común. La extensión de estos archivos es *.edf*.

Formato binario o .dat.- Este formato es uno de los más comúnmente utilizados en "Physionet", sin embargo, hay otras bases de datos que incorporan estos formatos. El archivo esta escrito en "formato 212". Este formato permite dos muestras de 12-bit para ser empaquetado en bits dentro de tres bytes de 8 bits. En muchos casos, para determinar si el formato del archivo es binario, hay un archivo del encabezado (*.hea*) que

describe los detalles del archivo. Con la finalidad de visualizar estos archivos eficientemente, pueden usarse programas C, C++ o fortran. Hay también en la página web de Physionet un programa que lee y convierte archivos binarios en texto.

Formato C3D.- Este formato es un dato estándar de 3D biomechanic. Este formato almacena coordenadas 3-D y datos numéricos para cualquier medición, con todos sus parámetros que describe los datos, en un archivo simple. Este estándar fue creado para estandarizar la forma de datos que pueden ser accedidos en una gran variedad de productos fabricados. Este formato incluye dos clases de información, mediciones físicas (coordenadas 3-D o data numérica) e información acerca de parámetros de mediciones. El archivo C3D generalmente almacena un número pequeño de parámetros comunes que describen los datos 3D (18) y permite entonces la definición por los usuarios, generar y almacenar dentro del archivo cualquier número de datos definido por los usuarios de tal manera que cualquiera puede abrir el archivo y accederlos. Para mayor información acceder a la siguiente dirección <http://www.c3d.org/> o a la dirección <http://www.emgsrus.com/>.

Formato .dst.- El significado del formato de estos archivos es “Data Storage and Transfer” (Almacenamiento y transferencia de datos), utilizado para el almacenamiento e intercambio de datos y no está enfocado en un tipo de datos en particular, equipamiento o bio-mecánico. Este formato no es utilizado en el caso de grandes lotes de información por que los archivos no son muy compactos.

Archivos Zip o .Gz.- Son archivos de formato simple que han sido creados utilizando un utilitario de compresión. Muchos archivos .zip y .gz contienen varios archivos dentro de ellos. El propósito de estos archivos es comprimir los datos y hacerlos más manejables en términos de almacenamiento y descarga. Hay muchos utilitarios que pueden ser utilizados para descomprimirlos. Algunos de los comúnmente usados son el WinZip y el Pkzip.

Matlab o archivos (.m).- Archivos propuestos por “Mathworks program Matlab”. Estos archivos son archivos de datos que frecuentemente contienen una gran cantidad de información en forma de matriz. Son utilizados comúnmente para procesamiento de señales digitales.

b) Ejemplos de Formatos

Formato ASCII para señales EEG (escrito por Paul Bourke – Setiembre 1997).- A

continuación describimos un formato de archivo ASCII para almacenamiento de señales EEG multicanal. Esto no significa que sea un formato ideal para archivar datos pero preferentemente es más conveniente un formato independiente de los equipos para intercambio de análisis de investigación y datos (17).

El archivo consiste de dos líneas de encabezado seguido por los datos. La primera línea de encabezado tiene 4 constantes, el número de canal, la frecuencia de muestreo (Hz), el número de eventos, y el número de muestras. La segunda línea de encabezado consiste en el nombre de las columnas de la sección de datos los cuales son predefinidos, estos son el número de muestras y el código del evento. Las columnas remanentes contienen las muestras de cada canal (electrodo).

Notas:

- La primera columna, el número de la muestra, empieza por "1". Por ejemplo, puede ser utilizado para identificar en que parte de una grabación muy extensa un subconjunto de datos fue adquirido.
- Cómo el código del evento es codificado no es especificado excepto que el "0" representa la ausencia de un evento.
- Las unidades y tipo de datos numéricos de las muestras actuales no son especificadas. En grabaciones en vivo ellos pueden ser números directos desde el ADC, para datos procesados ellos pueden ser números de punto flotante en micro voltios.
- Cada valor es separado por espacios en blanco estándares, por ejemplo, un simple espacio, múltiples espacios, un carácter tab.

Ejemplo:

Las primeras líneas representativas de un archivo pueden apreciarse como se muestra a continuación en la Tabla 2.2:

Tabla 2. 2 Formato ASCII para señales EEG

```

62 500 0 1000
sample event FP1 FPZ FP2 F3A F4A F7 F5 F3 F1 FZ F2 F4 F6 F8 C5A C3A ...
0 0 -1746 -1717 -1716 -1710 -1676 -1528 -1738 -1493 -1736 -1324 -1432 ...
1 0 -1962 -1891 -1988 -1864 -1717 -1663 -1853 -1583 -1746 -1378 -1505 ...
2 0 -2112 -1952 -2128 -1936 -1688 -1748 -1867 -1612 -1786 -1394 -1548 ...
3 0 -2063 -1857 -2032 -1853 -1600 -1694 -1774 -1547 -1802 -1373 -1535 ...
4 0 -1767 -1595 -1697 -1551 -1438 -1433 -1552 -1406 -1688 -1296 -1437 ...
5 0 -1315 -1222 -1229 -1128 -1219 -1032 -1196 -1176 -1430 -1172 -1281 ...
6 0 -898 -887 -822 -802 -1062 -683 -838 -976 -1144 -1067 -1148 -1139 ...
7 0 -757 -799 -692 -772 -1042 -571 -676 -940 -1051 -1079 -1143 -1147 ...
: : : : : : : : : : : : : : :

```

Formato de archivo NeuroScan EEG – EEG Continuos.- Los siguientes documentos de la estructura de archivos para archivos EEG continuos fueron creados para SCAN versión 4 (ver Fig. 2.2). La experiencia ha mostrado que muchos de los campos no son llenados correctamente por los programas de software. En particular, la mejor forma para calcular el número de muestras es:

$$\text{nsamples} = \text{SETUP.EventTablePos} - (900 + 75 * \text{nchannels}) / (2 * \text{nchannels})$$

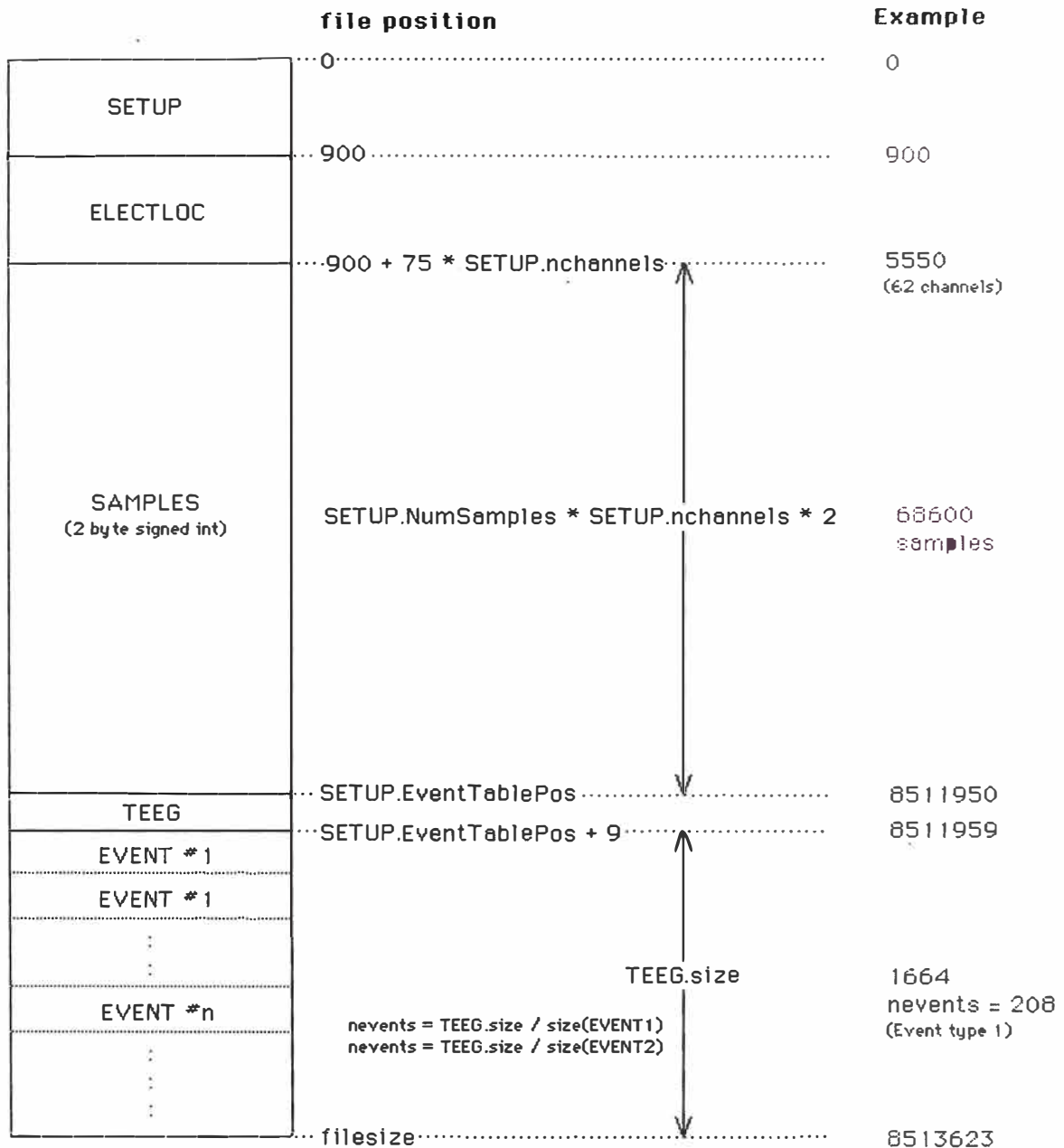


Fig. 2. 2 Estructura de datos para archivos continuos Neuroscan EEG

Ver en el Anexo I, un “script” “Convertor de archivos EEG de datos continuos NeuroScan a ASCII”, que sirve para visualizar los datos más amigablemente.

- **Promedio de señales EEG**

Calcular el número de muestras esperadas desde el tamaño del archivo con la siguiente fórmula:

$$\text{nsamples} = ((\text{filesize} - 900 - \text{nchannels} * 75) / \text{nchannels} - 5) / 4$$

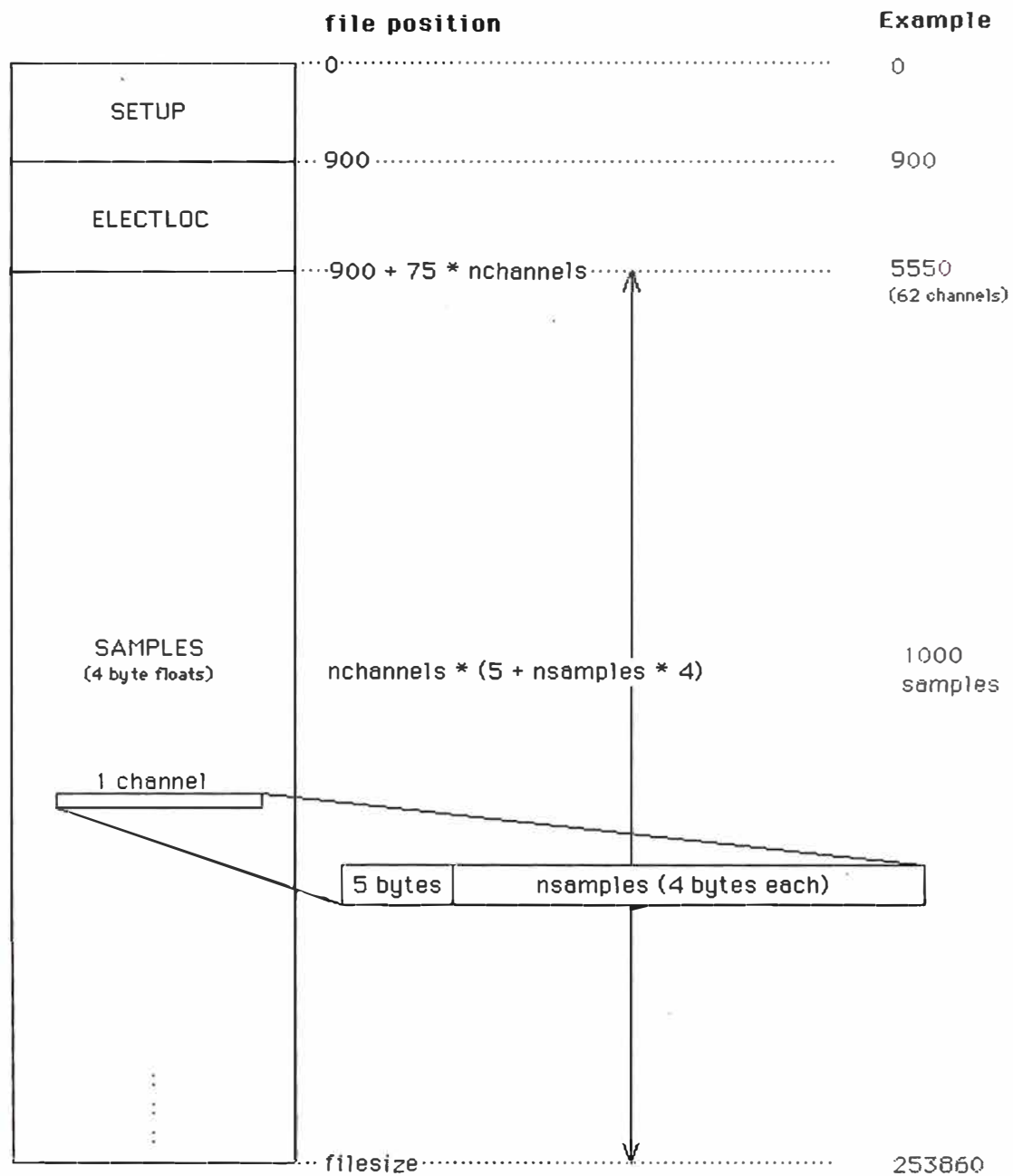


Fig. 2. 3 Estructura de datos para archivos EEG promedios Neuroscan

Ver en el Anexo I un "script" "Convertor de archivos Promedios de EEG NeuroScan a ASCII", que sirve para visualizar los datos más amigablemente.

Así mismo, en el Anexo I, tenemos una estructura "Sethead.h para Archivos EEG NeuroScan", el cual facilita implementaciones con muchas marcas de máquinas.

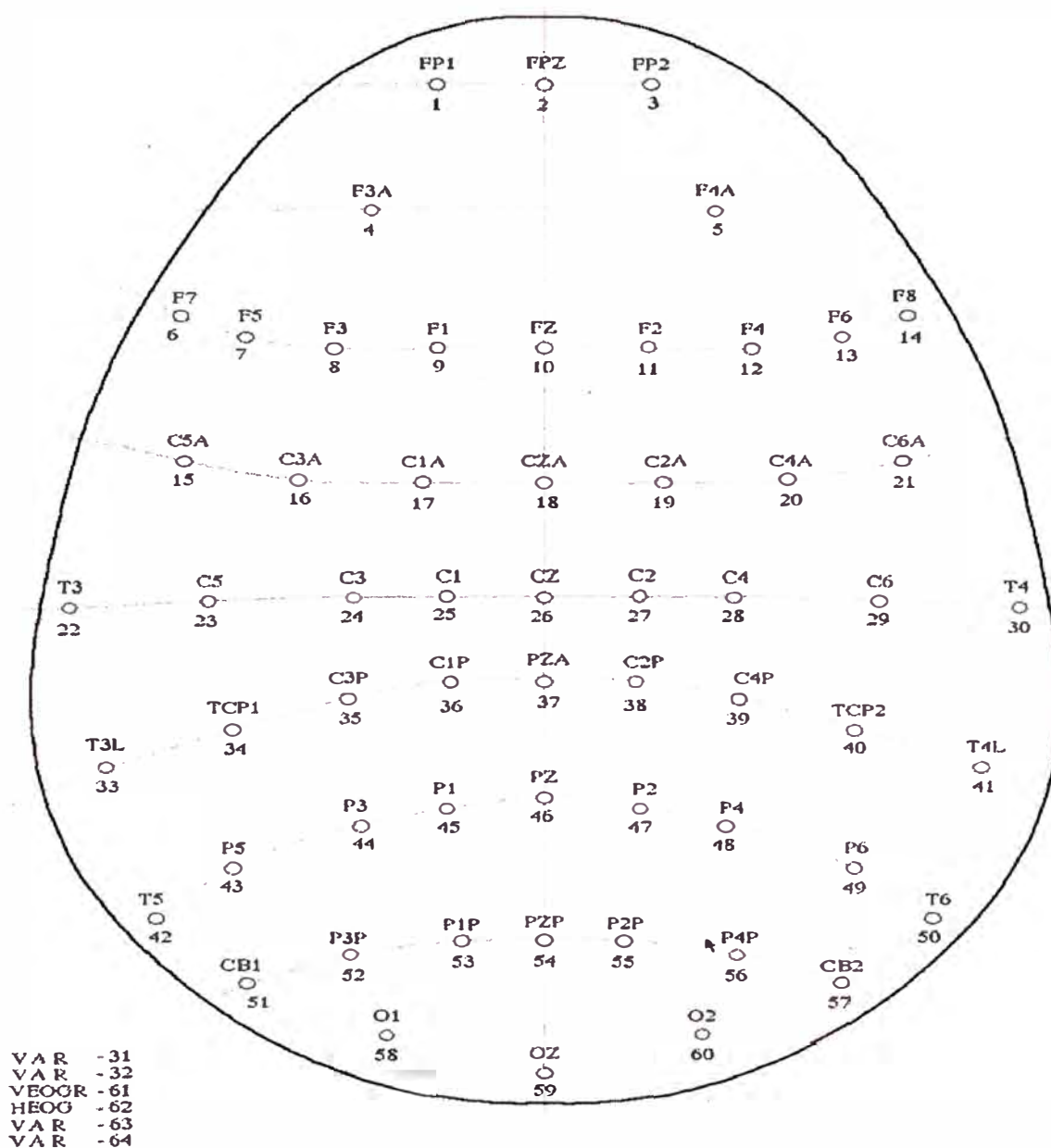


Fig. 2. 4 Posición de electrodos como fue utilizado por el laboratorio "Brain Dynamics Laboratory", 1996/1997

Fuente: *Data Formats*, Paul Bourke, Australia 1997

Formato BKR V 2.07.- Un archivo BKR empieza con un encabezado conteniendo información como frecuencia de muestreo, número de canales, etc. La segunda parte de los archivos es el block de datos con valores de muestras por cada canal. El tamaño del block de datos esta dado por el número de experimentos (ntr), canales (nch), número de muestras por experimento (nsp) y el tamaño de los datos de un valor de muestra (2 bytes para SSHORT), si el archivo no tiene un punto de referencia del final del block de datos con el archivo BKR.

Ver en Anexo I, la tabla “**Formato BKR V 2.07**” en el cual se puede observar la estructura de este formato.

Formato de archivo Westmead EEG.- A continuación describimos los aspectos importantes de los archivos de datos EEG del “Department of Medical Physics and Cognitive Neuroscience Unit, Westmead Hospital, Westmead, NSW 2145, Australia” (ver).

Estos archivos consisten en una línea de Identificación (ID) seguido por un encabezado y luego la sección de datos. El ID y header son textos ASCII/legibles, mientras que los datos son binarios, simplemente una serie de 2 bytes enteros son signo, usualmente referido a un C como un “short int”). Notar que mientras los archivos tienen un encabezado ASCII ellos pueden ser transferidos por FTP o en modo binario para preservar la parte de datos EEG. La naturaleza mezclada ASCII/BINARIO hace complejo a estos archivos, siendo conveniente mirar el encabezado usando un editor de texto.

En el Anexo I, tenemos un código fuente en lenguaje C y que con un “Convertor de formato de archivo Westmead EEG a ASCII” puede ser fácilmente modificado para utilizarlo de acuerdo a sus necesidades.

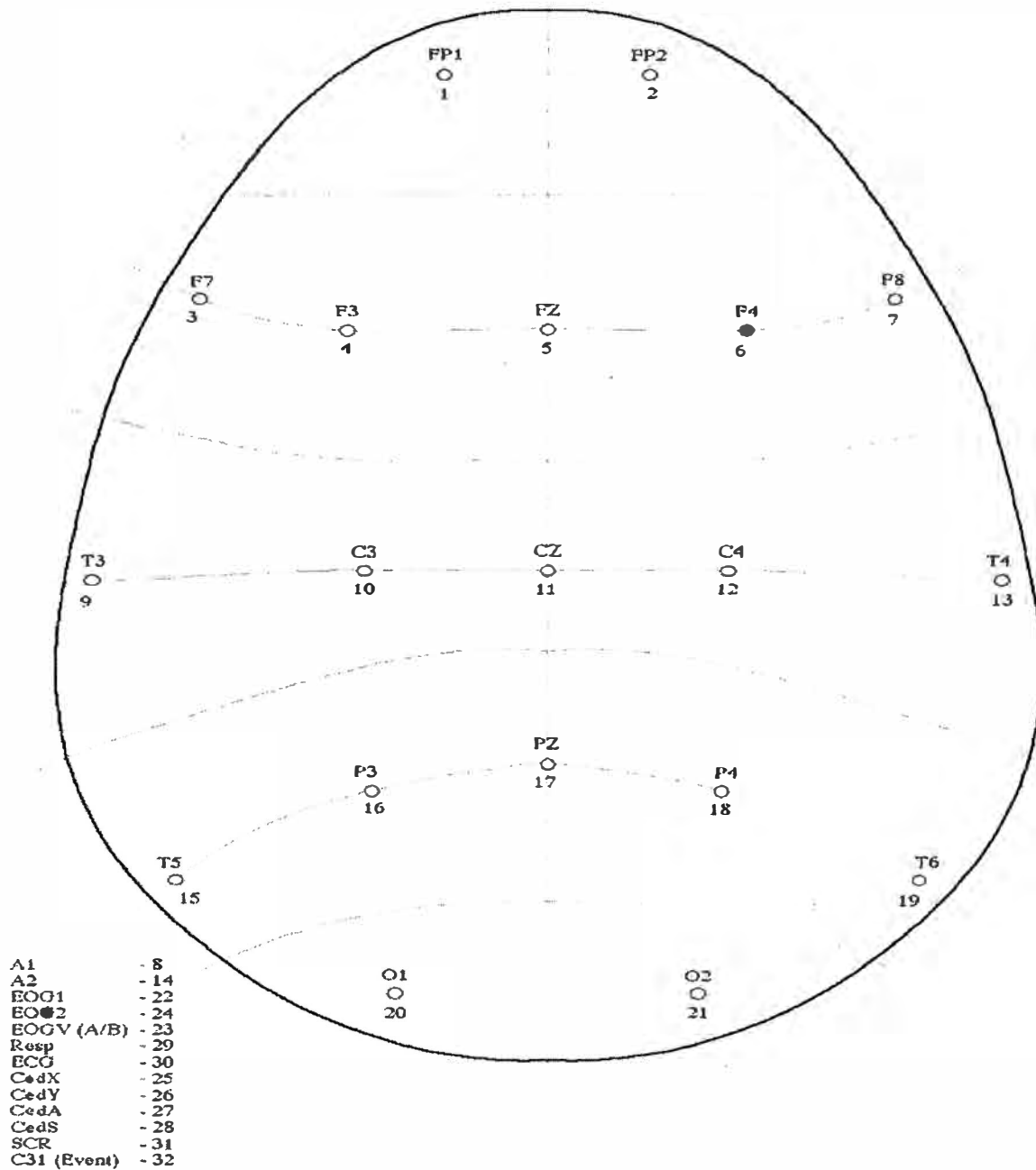


Fig. 2. 5 Posición de electrodos usado por Westmead Hospital

Fuente: *Data Formats*, Paul Bourke, Australia 1997

2.1.3. Coordenadas de localización de los Electrodos en EEG

Existen una variedad de tipos de coordenadas de localización de electrodos, los cuales son utilizados en diferentes formatos estándar de archivos (varios de los cuales menciono en el capítulo anterior). Dentro de estos tipos de coordenadas, tenemos los siguientes:

a) **Coordenadas Polares.**- Coordenadas de 2-D, definidas por ángulo y radio.

Angulo en grados: oreja derecha es 90°; oreja izquierda es -90°;

Radio del disco de la cabeza es 0.5

Campos: N ángulo radio etiqueta

Ejemplo: 1 -18 .511 Fp1

2 18 .511 Fp2

3 -90 .256 C3

4 90 .256 C4

.....

.....

Nota.- En versiones anteriores, la etiqueta tenía que estar definida por 4 caracteres y en caso que nos los tuviere se completaba con ".", esto todavía es utilizado pero ya no es necesario.

- b) Coordenadas esféricas Matlab.-** usadas por las funciones de Matlab y son diferentes de las coordenadas esféricas utilizadas por BESA. El ángulo theta representa la azimuth/horizontal y está en grados: 0° es hacia la nariz, 90° rotado a la oreja izquierda. Luego, representamos la elevación (phi), ángulos en grados (10).

Campos: N theta phi etiqueta

Ejemplo: 1 18 -2 Fp1

2 -18 -2 Fp2

3 90 44 C3

4 -90 44 C4

.....

- c) Coordenadas esféricas BESA.-** En estas coordenadas el ángulo de elevación (phi) es medido del eje vertical. La rotación positiva es hacia la oreja izquierda. Luego, la descripción del ángulo de rotación (theta) azimuth/horizontal es: 0° es hacia la oreja derecha, 90° es hacia la nariz, -90° hacia el occipital. Si las etiquetas están ausentes o son muy extensas son colocados en la última columna.

Las etiquetas por defecto son E1, E2.

Los campos son: etiqueta phi theta

Ejemplo: Fp1 -92 -72

Fp2 92 72

C3 -46 0

C4 46 0

....

- d) **Coordenadas Cartesianas Matlab/EEGLAB.-** En esta opción las referencias son: "x" es hacia la nariz, "y" es hacia la oreja izquierda, y "z" hacia el vértice.

Los campos son: Número_de_Canal x y x etiqueta

Ejemplo: 1 .950 .308 -.035 Fp1
 2 .950 -.308 -.035 Fp2
 3 0 .719 .695 C3
 4 0 -.719 .695 C4

- e) **Coordenadas polares Cartesianas - NeuroScan.-** Esta opción esta soportada en archivos en texto y se puede ver con mas detalle en el punto anterior.
- f) **Coordenadas Cartesianas BESA/EGI-xyz.-** Para EGI, "x" está hacia la oreja derecha, "y" está hacia la nariz, "z" está hacia el vértice.

Sus campos son: etiqueta x y z

Ejemplo: Fp1 -.308 .950 -.035
 Fp2 .308 .950 -.035
 C3 -.719 0 .695
 C4 .719 0 .695

2.2. Análisis de las señales EEG

2.2.1. Cuantificación y parámetros

El objetivo del análisis de una señal EEG es obtener una medida extraída del efecto electroencefalográfico que relacione el EEG con otra señal de interés, sea ésta por efecto de algún movimiento efectuado por el cuerpo, concentración de algún fármaco, probabilidad de conciencia, algún recuerdo u otro. Es decir, conseguir que un parámetro derivado del EEG refleje de forma sencilla los cambios o la presencia de algún componente que deseamos estudiar, parámetro que debería contener exclusivamente información del evento en estudio, siendo los métodos de análisis más complejos. El estudio y análisis de los diversos parámetros que puedan extraerse del EEG requiere de nuevos métodos de análisis estadístico para optimizar su correlación con el evento en estudio, métodos que permitan incorporar la máxima cantidad de información que permita identificarlos. Por ejemplo, en el caso de análisis de efectos farmacológicos, realizar el

análisis de los cambios del parámetro asociado, dará una perspectiva clara de la evolución del efecto farmacológico.

2.2.2. Métodos de análisis de la señal

Existen diferentes métodos con los cuales se puede analizar las señales EEG. Por ejemplo, analizando y evaluando los cambios en los componentes de las señales, este método aplica al análisis espectral y bioespectral (6). Por otro lado, también tenemos el método de análisis y evaluación de cambios que ocurren en forma cronológica, este análisis se efectúa en el dominio del tiempo o en análisis aperiódicos. Así mismo, tenemos el método mediante el cual se analiza el grado de "desorden" o caos (8) existente en las señales EEG (análisis de entropía).

Los métodos anteriormente mencionados ofrecen la posibilidad de obtener parámetros que ayuden a identificar situaciones clínicas en pacientes o para definir las propiedades de los agentes anestésicos y su profundidad anestésica alcanzada, o para identificar acciones efectuadas por un individuo, su estado de ánimo u otros.

2.2.3. Análisis en el dominio de la frecuencia

Las señales de ondas EEG pueden descomponerse en sus componentes fundamentales de tipo sinusoidal que al interrelacionarse dan como resultado la señal EEG. Los métodos descritos a continuación, análisis espectral y análisis bioespectral, descomponen las señales EEG en sus componentes esenciales.

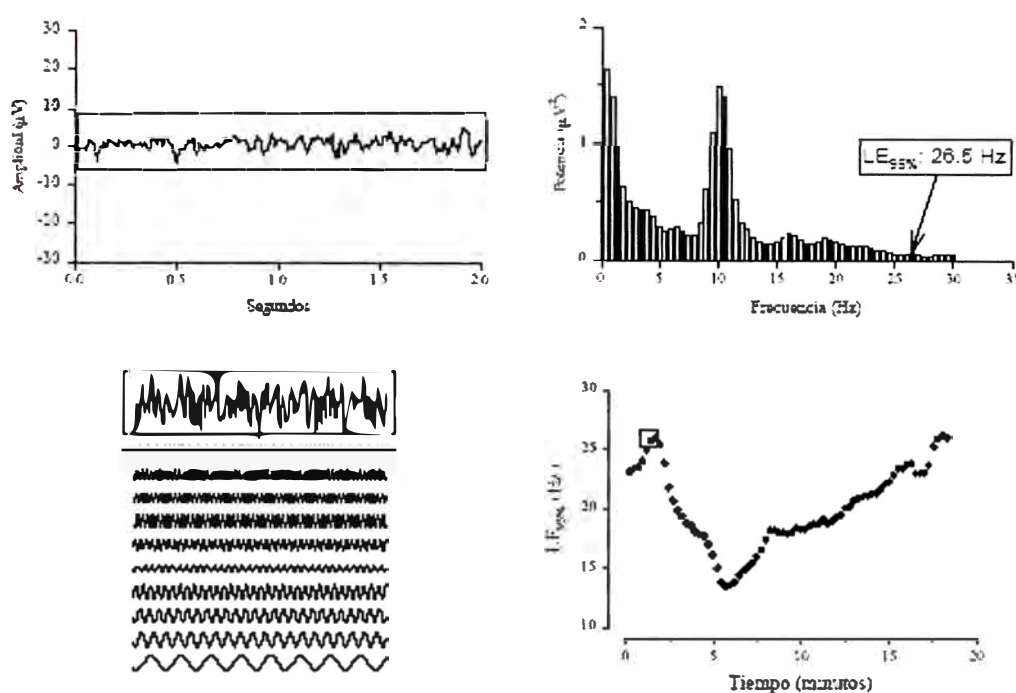
a) Análisis espectral

El análisis espectral se aplica a señales periódicas y estacionarias, es decir señales con patrones que se repiten con cierta periodicidad y cuando la distribución de las frecuencias del espectro no cambia, estas condiciones son asumidas para su análisis en las señales EEG.

Así mismo, el análisis espectral consiste también en el análisis de pequeños fragmentos o segmentos correlativos de las señales EEG, los cuales son denominados "epochs", segmentos que contienen los componentes básicos o fundamentales de la señal original y cuya duración más utilizada es de dos segundos y variable según algunos autores. Los componentes básicos de cada "epoch" son ondas sinusoidales, ondas que pueden ser definidas por su frecuencia, amplitud, y fase, componentes que pueden utilizarse para su análisis, como el caso de la amplitud que sirve para calcular la potencia, el cual se calcula

como el cuadrado de la amplitud de cada una de las frecuencias componentes de la "epoch" de la señal EEG, valor que puede ser asignado a un parámetro con las características que se desean estudiar, observándose principalmente la tendencia del cambio que induce cierto evento en estudio, como por ejemplo el estado de sueño de una persona.

A continuación vemos en la Fig. 2.6 un ejemplo de análisis espectral, utilizando para ello un fragmento de registro electroencefalográfico de un sujeto en reposo (6).



El panel superior izquierdo recoge 5 segundos de EEG basal. En el panel inferior izquierdo se detalla la descomposición de los dos primeros seg en sus sinusoides componentes fundamentales. El panel superior derecho muestra la misma descomposición en forma de histograma y calcula el LE95%. El panel inferior derecho representa el curso temporal del LE95% durante la administración de una infusión de propofol, así como la situación (enmarcada) del punto correspondiente al segmento de EEG analizado.

Fig. 2. 6 Análisis espectral de un fragmento de señal EEG.

Fuente: División de ciencias de la salud Universidad de Barcelona, España 2003

Como se observa, la descomposición en las diferentes ondas fundamentales que constituye el espectro de frecuencias está en función de la frecuencia, amplitud y fase. También se puede observar en forma de histograma el espectro de frecuencias, el cual es la descomposición de la señal en su "espectro" de frecuencias lo cual se conoce como análisis espectral. El análisis de esta descomposición espectral inicialmente se basó en los principios desarrollados por *Fourier*, mediante el cual se puede detectar diferencias en el ritmo y armónicos entre "epochs" de sujetos sanos y de pacientes con alteraciones

mentales, posteriormente, se desarrollaron estudios utilizando para ello una aproximación matemática, denominada *Transformada Rápida de Fourier* (TRF) que permite calcular de forma rápida (tiempo real) y obtener en estos tiempos los análisis espectrales, tanto en la distribución de las frecuencias como también en la potencia de cada una de las frecuencias componentes del espectro, en un rango de frecuencias que van desde 0.5 Hz hasta 30 Hz. Los filtros FFT son diseñados para rechazar las frecuencias superiores a 30 Hz porque se considera que no existe información importante en ese rango y también por que hay más probabilidad de que se mezclen con señales parásitas tales como: señales de la actividad muscular o sistemas que funcionan con electricidad,

b) Análisis biespectral: Índice Biespectral

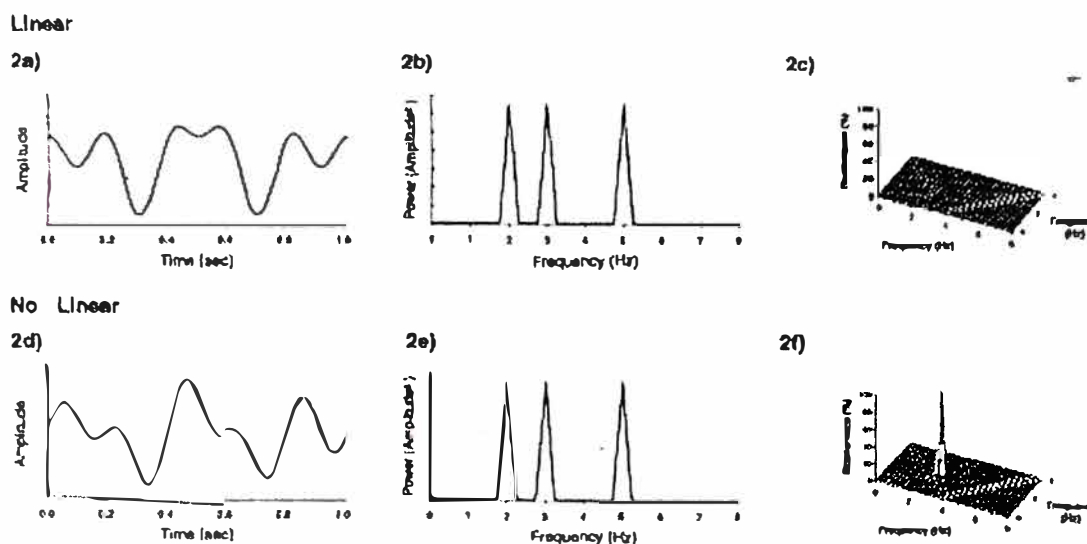
En el análisis espectral de las señales del EEG ya mencionado, en el cual se utiliza las TRF, se obtiene información analizando la potencia, amplitud o frecuencias componentes de las señales EEG. Sin embargo, otros métodos utilizan la característica de la fase de las señales sinusoidales, analizando e integrándolo, es posible conocer el grado de armonización que puede haber entre dos tramos de señales y, por tanto, definir relaciones incluso no lineares, permitiendo inclusive valorar la coherencia entre tramos de estas señales, entendiendo que las relaciones entre los impulsos neuronales probablemente no son simples y lineales.

Cabe mencionar también que la distribución de las amplitudes en el espectro no establece una distribución normal y tiene la probabilidad que, al igual que existen relaciones no lineares entre las neuronas, éstas se traduzcan en relaciones no lineares entre los diferentes tramos (*epochs*) de señales EEG.

El método de análisis *biespectral* permite cuantificar las desviaciones de la normalidad así como las relaciones no lineares de tipo cuadrático, es decir, cuando una variable dependiente es el resultado de elevar al cuadrado cada uno de los valores de la variable independiente.

Para ello, el análisis biespectral permite cuantificar la interacción entre todos los componentes que forman la señal EEG incluida la fase de la sinusoide.

En la figura Fig. 2.7 continuación recogemos un ejemplo de cómo la aplicación de análisis biespectral permite obtener más información sobre el tren de ondas que el análisis espectral.



Las gráficas 2b y 2e muestran dos espectros de frecuencias idénticos. Sin embargo, proceden de dos trazados de EEG diferentes. Las ondas componentes de 2a y 2b son idénticas en frecuencia y amplitud por lo que su espectro es idéntico, pero la fase de la onda de 3 Hz es diferente en 2a de 2d. Por ello la interacción de las tres es detectable en el segundo caso al aplicar análisis biespectral cuyo resultado gráfico es visible en el gráfico 2f.

Fig. 2. 7 Comparación entre análisis espectral simple y análisis biespectral.

Fuente: División de ciencias de la salud Universidad de Barcelona, España 2003

Este análisis se lleva a cabo mediante una expansión de la TRF. Mientras la TRF descompone la señal en sus componentes fundamentales y asume que éstos son independientes, el análisis biespectral permite medir las potenciales interacciones entre las sinusoides componentes de los trenes de ondas. Los componentes que dependen entre sí se denominan "armónicos". Así como el análisis espectral permite identificar amplitud, fase y frecuencia, empleando el análisis biespectral es posible identificar otros parámetros. Uno de ellos es la bicoherencia, que representa el grado de acople existente entre las fases de los posibles componentes. Se trata de una variable que define también el grado de no linealidad, expresada en forma de porcentaje, de forma que la bicoherencia de un sistema en que las relaciones son enteramente lineales sería del 0%, mientras que los sistemas no lineales presentan grados variables de bicoherencia. El otro cálculo básico en el análisis biespectral del EEG es el del Triple Producto Real (*Real Triple Product*). Este valor se obtiene a partir de la valoración conjunta de dos frecuencias primarias cualquiera, f_1 y f_2 , y de su componente de modulación $f_1 + f_2$, constituyendo el trío f_1 , f_2 , y $f_1 + f_2$. Por tanto para cada par de frecuencias existirá la Bicoherencia de ambas frecuencias y la magnitud conjunta de los componentes del trío que en este caso sería el Triple Producto Real. Bajo estas premisas es posible afirmar que el análisis biespectral permite incrementar la información que conocemos sobre los cambios que aparecen en el trazado del EEG pues integra la posibilidad de caracterizar las

interdependencias que probablemente existan entre las ondas del EEG. Por tanto cualquier parámetro que se obtenga del EEG mediante análisis biespectral, al menos en teoría, será capaz de reflejar el efecto que los diferentes fármacos anestésicos hayan inducido mejor que los parámetros obtenidos mediante análisis espectral simple, al incorporar más información. Sigl y Chamoun desarrollaron la aplicación de análisis biespectral al estudio del trazado del EEG, con el objetivo de conseguir un indicador que permitiera monitorizar la profundidad anestésica. Desde los inicios de la Anestesiología ha existido interés por encontrar un sistema que permitiese cuantificar el nivel alcanzado. En 1920 *Guedel* describió los cuatro estados diferentes de la anestesia conseguida con la administración de éter, basándose en parámetros clínicos: nivel de consciencia, patrón respiratorio, presencia de movimientos oculares, tamaño pupilar, reflejo del vómito, reflejo palpebral, reflejo conjuntival, reflejo corneal y respuesta refleja al estímulo lumínico (6). La validez de estos signos quedó en entredicho desde la introducción en la práctica anestésica de los fármacos bloqueantes neuromusculares ya que sus efectos impiden que se puedan valorar la respuesta a la mayoría de estos estímulos. El poder sintetizar en un sólo indicador la profundidad alcanzada de los diversos efectos farmacológicos que componen el estado anestésico, permitiría ajustar los requerimientos a las necesidades individuales de cada paciente anestesiado. De ahí la posible relevancia de un parámetro como éste, el Índice Biespectral comercializado como Bis®. El Índice Biespectral en su versión actual es un conglomerado de sub parámetros extraídos del EEG tras la aplicación de análisis en el dominio temporal, análisis espectral y análisis biespectral, modulados para correlacionar de forma óptima con la concentración de hipnótico administrado y también con las respuestas clínicas asociadas con hipnosis. Estas respuestas son la presencia o ausencia de consciencia, los diferentes niveles de sedación o la capacidad o no de desarrollar memoria a diferentes concentraciones de fármacos hipnóticos. Se trata de un indicador cuyo valor oscila entre 100 y 0. Valores alrededor de 100 se asocian con consciencia, ausencia de efecto hipnótico, valores alrededor de 0 indicarían la máxima depresión inducida por hipnóticos, siendo independiente su valor del fármaco hipnótico empleado. Se considera que por debajo de 50 la probabilidad de que el paciente esté consciente es mínima (6).

Las versiones iniciales del algoritmo de cálculo del Índice Biespectral contemplaban la posibilidad de predicción de movimiento del paciente como respuesta a un estímulo doloroso. Sin embargo se demostró que dicha capacidad de predicción de movimiento quedaba alterada en el caso de que se hubiera administrado opiáceos al paciente, sin que la presencia de opiáceos afectara al valor del Índice Biespectral. La versión inicialmente comercializada e incorporada en el monitor A 1000 salido al mercado en

1994, Bis versión 2.5, ya había sido actualizada para presentar máxima 26 correlación con respuestas relacionadas con el componente hipnótico de la anestesia. El algoritmo de cálculo y por tanto el Índice Biespectral ha pasado por sucesivas versiones capaces de detectar y mejorar la calidad de la señal frente a artefactos fisiológicos o externos al paciente y por tanto la fiabilidad del valor del Índice Biespectral.

2.2.4. Análisis en el dominio temporal

Los métodos de análisis efectuados en el dominio del tiempo son los que se basan en el análisis y **evaluación de los cambios** de las señales EEG en cada *epochs* y en **función del tiempo transcurrido**.

Dentro de estos métodos, utilizados en muchos trabajos, tenemos por ejemplo el de la **determinación de la amplitud media de la señal por unidad de tiempo** (evaluación de los cambios), analizada en cada *epoch* y asociada a su correspondiente espectro resultante, esta información da la característica del evento en estudio y para su análisis es necesario que la amplitud del espectro tenga una distribución normal o *gaussiana*, por lo que para cumplir con esta afirmación es necesario asumir que la amplitud del espectro sigue esta distribución y de no ser el caso, es recomendable realizar transformaciones que normalicen dicha distribución, por ejemplo la transformación logarítmica de las amplitudes.

Así mismo, tenemos también el método "**zero crossing frequency**" (**función del tiempo transcurrido**), el cual es derivado del análisis en el dominio del tiempo y fundamentado en el estudio de las frecuencias y cómo se produce el cruce de la línea de voltaje por cero, teniendo en cuenta que la mayoría de las señales EEG en un tramo en concreto del EEG, cruzan la línea de cero voltios. De lo cual, se puede distinguir tramos de señales "*positivos*" y tramos "*negativos*", información que servirá para cuantificar el número o la magnitud de los tramos positivos respecto al número o magnitud de tramos negativos por segmento de EEG. Este método tiene el inconveniente que en el caso de ser señales de magnitud pequeña, pueden estar moduladas por una señal de mayor amplitud y con la posible consecuencia de no cruzar la línea de voltaje cero, razón por la cual en algunos casos no se le emplea.

a) Tasa de supresión

Por otro lado, también tenemos el método "*Burst Supresión Ratio*" (BSR), o llamado también, "*Tasa de Supresión*" (TS), el cual es un parámetro que compara cambios que suceden en el comportamiento de las señales en función del tiempo, debidos sobre todo

al empleo de fármacos hipnóticos potentes. Este método permite calcular la relación entre los períodos con presencia de señal electroencefalográfica y los períodos en que aparece situación de *isoelectricidad* (línea eléctrica neutra o plana) debida al efecto farmacológico. La TS es uno de los sub parámetros en los que se basa el cálculo del Índice Biespectral, indicado anteriormente.

Otros métodos como el llamado *Monitor de Función Cerebral*, basado en un complejo análisis acompañado de filtrado y varias transformaciones de la señal, han quedado obsoletos o demostrado situaciones de incoherencia desde el punto de vista estadístico o clínico, lo que ha llevado a abandonar su uso.

Concluyendo, y por las referencias encontradas, se puede mencionar que de todos los sistemas que analizan el EEG en el dominio del tiempo, el análisis aperiódico de las señales electroencefalográficas es el método que ha sido empleado más extensamente y sobre todo en aplicaciones relacionadas con la descripción del efecto de los fármacos.

2.2.5. Entropía en las señales EEG

Los cambios en el patrón del EEG de irregular a más regular, cuando el paciente pierde conciencia, se entienden como una disminución de la Entropía del EEG, en el análisis de las señales EEG sirve para describir las características de irregularidad (complejidad, impredecibilidad, etc.) de estas señales.

Teniendo claro que la complejidad del EEG se correlaciona con el nivel de actividad cerebral, se pueden aplicar varios algoritmos basados en el concepto de Entropía (en sus diversas definiciones) como descriptores de la regularidad de la señal del EEG.

Existen varios métodos que han sido utilizados con resultados consistentes:

a) Entropía espectral

Desorden en el espacio de la frecuencia. Analiza la señal del EEG en el dominio de la frecuencia, para lo cual se necesita saber el contenido en frecuencias del EEG.

b) Entropía aproximada Kolmogorov-Sinai

Desorden en el espacio del tiempo. Analiza la señal del EEG en el dominio del tiempo. Se basa en la posibilidad de predecir la señal de EEG conociendo la señal previa.

Todos estos métodos se basan en la evidencia de que a mayor actividad cerebral, mayor complejidad del EEG.

2.2.6. Técnicas actuales

a) Análisis de Componentes Independientes (ICA)

ICA es una técnica que trata acerca de la factorización de la distribución de probabilidad, haciendo la separación desde una fuente de datos oscura (conjunto de datos mezclados). Está relacionado a muchas cosas tales como, entropía y maximizando la información, maximizando la probabilidad de estimación de densidad (MLE), EM (Maximización de la expectativa, el cual es MLE con variables ocultas) y búsqueda de proyección. Es básicamente un camino de búsqueda especial de sistema de co-coordenadas (no ortogonales) lineares en datos multi-variables, usando estadísticas de alto orden en varias formas.

Su aplicación se utiliza en donde quiera que se tiene un conjunto de datos (información) multi-variada, ejemplo: en cualquier sitio donde se puede usar PCA (Análisis de Componentes Principales). Los ejemplos incluyen separación de los datos oscuros (ejemplo: de señales de voz mezcladas), procesamiento de datos biomédicos (ejemplo: de datos EEG), búsqueda de "características" en los datos (ejemplo: aprendiendo detectores de cambios para conjunto de imágenes naturales) (20).

El crecimiento del interés en la clasificación del Electroencefalograma (EEG) ofrece una necesidad para el desarrollo de técnicas de análisis y procesamiento apropiados. Uno de los más significativos puntos asociados con análisis de EEG es la alta contaminación de la señal grabada con varias señales contaminantes, que pueden ser de los sujetos objeto del análisis y de la interferencia de los equipos.

La grabación externa de las señales EEG es altamente atenuada y mezclada desde sus orígenes debido a la actividad de miles de neuronas, los cuales pasan a través de diferentes capas finas antes de ser alcanzadas por la grabación de los electrodos. Un importante problema en el análisis de EEG es la contaminación de la señal con varios efectos perturbadores, tales como el parpadeo de los ojos o movimiento de los músculos de la cara. Estas perturbaciones pueden ser removidas previamente a cualquier análisis. Un método para alcanzar este objetivo es fijando umbrales, donde cualquier dato cuya amplitud excede un umbral definido es descartado. Una consecuencia de fijar umbrales es la pérdida de la actividad EEG valuable, el cual ha sido enmascarado por la gran amplitud del disturbio.

Un método alternativo es la aplicación de la técnica "Blind Source Separation" (BSS), el algoritmo BSS más comúnmente utilizado es la técnica "Independent Component Analysis" (ICA). ICA es una técnica estadística para obtener fuentes independientes, "s", desde sus misturas lineales, "x", donde ni la fuente original ni la matriz mezclada actual,

“A”, son conocidos. Esto es logrado por la explotación de orden alta de la estadística de las señales y técnicas de optimización.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.1)$$

Con la finalidad de una aplicación exitosa de ICA para análisis de EEG, la suposición en el cual es basada en la función de ICA debe ser cumplida. En el estándar ICA es asumida que las señales grabadas son un resultado de mezclas lineares; las perturbaciones son estadísticamente independientes de las señales EEG y el número de fuentes es el mismo que el número de mezclas.

Hay un número de temas asociados con la aplicación de ICA para análisis EEG:

- Las señales EEG y perturbaciones tienen una estructura temporal. Esto está mostrado en la Fig. 2.8, el cual muestra la correlación de matrices de las señales EEG y perturbaciones para los diferentes retrasos de tiempo. Las matrices de correlación, C_{τ} , son definidas como la ecuación:

$$C_{\tau} = \left\langle s_i(t) \cdot s_j(t + \tau) \right\rangle \quad (2.2)$$

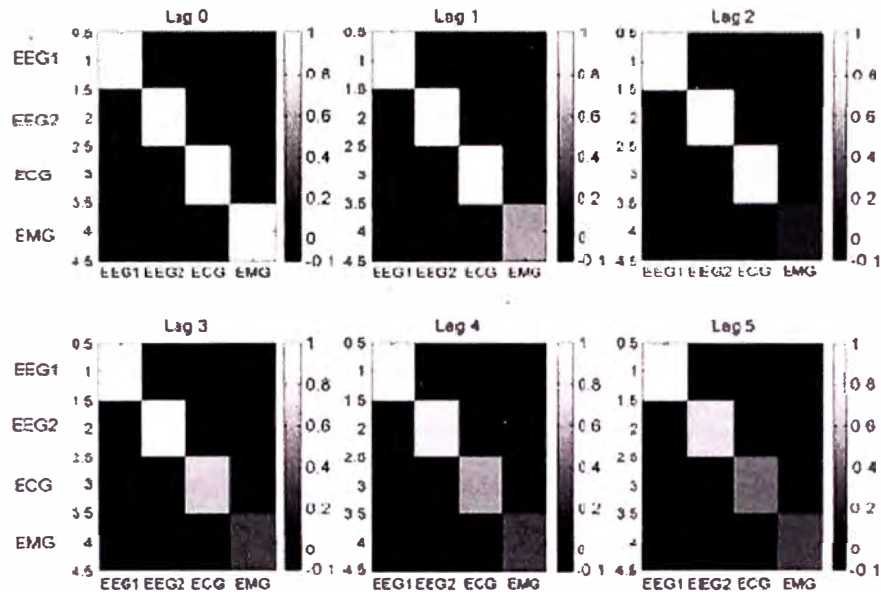
Donde:

- S_i Las señales
- $\langle .. \rangle$ Expectación
- τ retardo de tiempo; $\tau = 0, 1, \dots$

La estructura diagonal de la matriz de correlación para retardo de tiempos diferentes es claramente visible. Esto significa que las señales, S_i , son no correlacionados mutuamente. La presencia de elementos diagonales no nulos indica que todas las señales muestran alguna dependencia en sus valores anteriores, por lo tanto tener una clara estructura temporal. Sin embargo, ICA estándar, usa solamente mayores órdenes estadísticos información efectivamente descartada contenida en la estructura temporal de las señales.

- La distribución de amplitud de la EEG, al menos en regiones cuasi-estacionarias, es Gaussiana, como puede ser visto en la Fig. 2.9. Esto es también soportado por el uso extensivo de modelos de Auto regresión (AR) con distribución de ruido Gaussiano para modelamiento de las EEG en varios estudios. Sin embargo, fallas ICA estándares a separar la mixtura si más de uno de las fuentes tiene una distribución de amplitud Gaussiana. En consecuencia, empleando información de la estructura de tiempo en

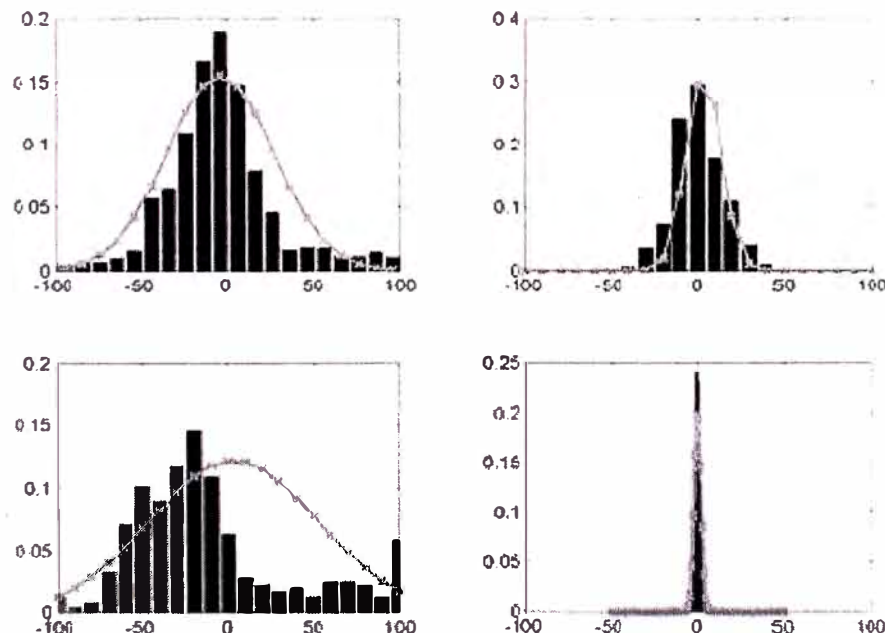
cálculos de ICA puede potencialmente mejorar la extracción de objetos perturbadores y realiza la clasificación en conjunto.



Matrices de correlación para dos señales EEG y dos señales perturbadoras (ECG y EMG) para un retardo de tiempo de 0 (gráfico de arriba izquierda) hasta a un retardo de tiempo de 5 (gráfico de abajo derecha). Los nombres de las columnas y filas son los mismos.

Fig. 2. 8 Matrices de correlación

Fuente: *Department of Cybernetics, University of Reading, UK 2004*



Histograma de la distribución de la amplitud de dos señales EEG (fila de arriba) y dos señales perturbadoras, ECG y EMG (fila de abajo). Se puede ver que la señal EEG muestra aproximadamente una distribución Gaussiana, como contrapuesto a la señal perturbadora.

Fig. 2. 9 Histograma de la distribución de la amplitud de señales EEG

Fuente: *Department of Cybernetics, University of Reading, UK 2004*

b) Potenciales de eventos relacionados (ERP)

Los potenciales de eventos relacionados (ERP) son pequeñas fluctuaciones de voltaje resultado de la actividad neural evocada (12). Estos cambios eléctricos son extraídos desde las grabaciones de la superficie de la cabeza por promedios de cómputo de periodos (epochs) de tiempo fijo de grabación de señales EEG a ocurrencias repetidas de eventos sensoriales, cognoscitivos, o eventos motores. La espontánea fluctuación del entorno EEG, los cuales son relativamente aleatorias a cuando el estímulo ocurrió, son fuera de lo promediado, dejando los potenciales de eventos relacionados del cerebro (21). Estas señales eléctricas refleja solamente que la actividad es consistentemente asociada con el procesamiento del estímulo en una forma de tiempo fijo. La ERP además refleja, la alta resolución en el tiempo, el patrón de la actividad neuronal evocada por un estímulo.

Debido a su alta resolución en el tiempo, los ERPs proveen únicas e importantes informaciones en el tiempo acerca del procesamiento del cerebro. Operaciones mentales, tales como los involucrados en la percepción, atención selectiva, procesamiento de lenguaje, y memoria, proceder sobre rangos de tiempos en el orden de 10 milisegundos. Muchas técnicas funcionales de graficar requieren la integración de la actividad del cerebro evocado sobre muchos segundos y de esta manera no es posible capturar estas señales en el curso del tiempo (o secuencia) de estas operaciones. Las grabaciones de los ERP, sin embargo, proveen reflexión de un milisegundo por milisegundo de la actividad del cerebro evocado. Por esta razón, los ERPs son una metodología ideal para el estudio de los aspectos en el tiempo de los procesos cognoscitivos normales y anormales. Por otro lado, la información ERP provee menor precisión de información espacial que la tomografía por emisión de positrón (PET) o los gráficos de la resonancia magnética (fMRI), el cual tiene falta de resolución fina temporal. Como consecuencia, los ERPs representa el complemento natural del PET y fMRI en el estudio del conocimiento humano. Mientras que el PET y el fMRI pueden localizar regiones de activación durante una tarea mental dada, los ERPs pueden ayudar en la definición del curso en el tiempo de estas activaciones (14).

En una definición clínica, la resolución en el tiempo de los ERPs son útiles en la identificación de en cual nivel a lo largo del camino sensorial esta localizado. Las ERPs visuales ayudan el diagnóstico temprano de la esclerosis múltiple antes que sea detectada una anomalía estructural. Los ERPs son también usados para monitorear pacientes comatosos para evaluar la funcionalidad en el centro vital del tronco cerebral.

CAPÍTULO III

HERRAMIENTAS DE PROCESAMIENTO

3.1. Herramientas de procesamiento EEG

Actualmente el desarrollo del análisis de las señales EEG esta en auge, interpretar las diferentes características y el gran potencial de información que se tiene en estas señales, ha creado una gran expectativa, motivo por el cual, instituciones de investigaciones, universidades, clínicas y otros, alrededor del mundo, están desarrollando diferentes herramientas que les permita facilitar su estudio, análisis y aplicaciones, y dentro de estas podemos mencionar las siguientes:

Herramientas de visualización

- EEG ISO surfaces

<http://www.banyantree.org/jsale/isosurfaces/isosurfaces.html>

Equipos

- MICROMED BQ3200

<http://www.micromed-it.com/inglese/brainquick4.htm>

- Bio-Medical Engineering

<http://www.elec.york.ac.uk/bio/welcome.html>

Herramientas

- Biomedical Signal data formats (formato de archivo específico para máquina EEG con rutinas importadas de Matlab)

<http://www.dpmi.tu-graz.ac.at/~schloegl/matlab/eeg/>

- Geodesic System 120

<http://www.egi.com/GSN.shtml>

<http://eeg.sourceforge.net/bioelectromagnetism.html>

Herramientas basadas en MATLAB

- EEGLAB

<http://www.sccn.ucsd.edu/eeglab/index.html>

Las funciones de esta herramienta de estudio se encuentran en:

<http://www.sccn.ucsd.edu/eeglab/allfunctions/index.html>

- La FMRIB Plug-in para EEGLAB

EEGLAB plug-in diseñado por Rami Niazy de la University of Oxford Centre for Functional MRI of the Brain (FMRIB), el cual permite la extracción de FMRI-“artifacts” (señales perturbadoras) relacionados de los datos EEG. Este plug-in está diseñado para trabajar dentro de un entorno de EEGLAB, proveyendo una interface gráfica de usuario (GUI) para eliminar “artifacts” de una gradiente FMRI, detectar complejas QRS de un canal ECG, y eliminar pulsos (ballistocardiographic/BCG) artifacts de las EEG. Todas estas funciones pueden también ser usados desde este plug-in desde la línea de comandos del Matlab, permitiendo a usuarios expertos usarlos en scripts personalizados. Este plug-in, un tutorial y más información pueden encontrarlos en:

<http://www.fmrib.ox.ac.uk/~rami/fmribplugin/>

- BrainStorm 2.0

<http://neuroimage.usc.edu/brainstorm/>

- Delay Vector Variance Method por Temu Gautama

<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3264&objectType=file>

- DTU Sidekick para EEGLAB

<http://www.imm.dtu.dk/~mad/sidekick.htm>

En el presente estudio y por ser una herramienta “Open Source” disponible para cualquier persona que este interesada en utilizarla, analizaremos las bondades de la herramienta EEGLab, desarrollado por el instituto “Swartz Center for Computational Neuroscience Institute for Neural Computation University of California San Diego”, la cual esta implementada para ser utilizada en el ambiente de MatLab.

3.2. Uso de Matlab en análisis de EEG

El MatLab es una herramienta muy poderosa que ha permitido el desarrollo de muchas aplicaciones en el análisis de las señales EEG, principalmente en el desarrollo de emuladores, simuladores que permiten realizar las pruebas y las verificaciones de las

teorías correspondientes a las investigaciones, tales como: “Biomedical Signal data formats”, “The FMRI Plug-in for EEGLAB”, “EEGLab”, “BrainStorm 2.0” y muchas otras. El progreso tecnológico en el mundo, ha permitido el desarrollo del uso de las señales EEG, principalmente enfocados en el campo de las neuro-imágenes, ya sea anatómicas: tomografía axial computarizada (TAC), resonancia nuclear magnética (RNM); como también neuro-imágenes funcionales: tomografía por emisión de positrones (PET), tomografía por emisión de fotón único (SPECT), resonancia magnética funcional (f-RNM), los cuales afianzan nuestros desarrollos en la comprensión del “donde” y del “como”.

Los desarrolladores de equipos utilizan las señales EEG desde hace mucho tiempo, por lo que están compitiendo con otras tecnologías, pero el avance de las velocidades de las computadoras ha permitido el desarrollo de aplicativos que antes se demoraban o era imposible efectuarlos, ahora se lleva a cabo con facilidad y en tiempos óptimos y también en tiempo real.

3.3. La nueva herramienta EEGLAB

La necesidad de una herramienta que facilite el manejo complejo de los procesos en los análisis de las señales EEG, ha determinado que un grupo de estudiosos con la colaboración del instituto “Swartz Center for Computational Neuroscience, Institute” desarrolle el aplicativo “EEGLab” como una herramienta muy potente el cual permite el análisis de muestras EEG en cantidades bastante altas, así como también con facilidades gráficas que facilita su uso en la presentación de formas 2D y 3D, permitiendo la aplicación de nuevos conceptos en la interpretación de las señales EEG.

EEGLab es una herramienta interactiva para procesamiento continuo y eventos relacionados con datos EEG usando Análisis de Componentes Independientes (ICA), análisis de tiempo / frecuencia y otros métodos incluyendo rechazo de objetos.

EEGLab provee una interface de usuario grafica e interactiva permitiendo a los usuarios procesos flexibles y de forma interactiva señales EEG de alta densidad o datos dinámicos del cerebro usando análisis de componentes independientes (ICA) y/o análisis de tiempo / frecuencia (TFA), también los métodos estándares promedios.

El EEGLab es una herramienta que tiene el código abierto para análisis y realización de pruebas EEG dinámicas incluyendo análisis de componentes independientes, el cual incorpora tutoriales y ayudas, más una función que permite ver los históricos de comandos.

EEGLab fue diseñado para usuarios novatos y expertos, dependiendo del nivel de experiencia que tengan en Matlab, los usuarios pueden interactuar con la interface gráfica (GUI) del EEG y también pueden invocar directamente funciones del EEGLab desde la línea de comandos del Matlab o escribir sus propios scripts de Matlab usando funciones y estructuras de EEGLab.

3.3.1. Las funciones eeglab están agrupadas en tres niveles

- a) **Función EEGLab principal y su manejador de menú:** Los usuarios EEGLab típicamente llaman a estas funciones por selección de ítems de menú desde el menú de la ventana principal de EEGLab (ver Anexo I).
- b) **Pop_functions:** Funciones Matlab con sus propias interfaces gráficas. Invocado con pocos argumentos (o ninguno) desde la interface de usuario, estas funciones "pop up" en una ventana solicitada para elegir opciones en parámetros adicionales. Los usuarios generalmente llaman a uno o varias de las funciones de procesamiento de señales de las herramientas del EEGLab. Las "pop_functions" también pueden ser invocados desde la línea de comandos de Matlab o desde scripts Matlab (ver Anexo I).
- c) **Funciones de procesamiento de señales:** Los usuarios experimentados en Matlab pueden invocar directamente las funciones de herramientas de ICA desde la línea de comandos de Matlab o desde sus propios scripts de análisis. Algunas funciones de ayudas están también en este nivel (Ver Anexo I).

En la distribución de esta herramienta, han optado una filosofía similar a la que adoptó Linux, es decir cualquier persona puede contribuir en el desarrollo de EEGLab, para ello se utiliza funciones de procesamiento de datos compatibles, de tal manera que cualquiera puede solucionar sus problemas y/o mejorarlos dirigiéndolo a eeglab@sccn.ucsd.edu. Los desarrolladores considerarán todas las sugerencias e incorporar funciones complementarias de interés general a esta herramienta.

3.3.2. Análisis de datos

En este apartado veremos cómo se usa la herramienta EEGLab a un preproceso interactivamente, analizando y visualizando la dinámica de eventos EEG relacionados,

usando para ello información obtenida del “Equipo electroencefalógrafo del Instituto del Sueño – Hipnos”.

a) Adquisición de las señales del cerebro

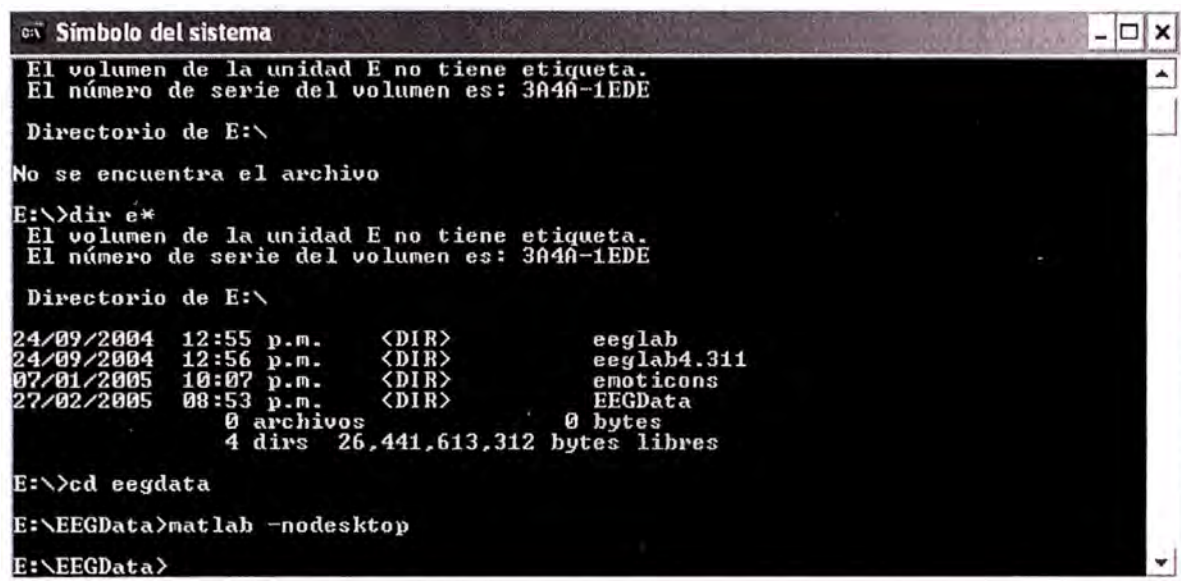
En general, el paciente que va a someterse a esta exploración no necesita una preparación especial. Los electrodos se colocan sobre el cuero cabelludo y se mantienen en el sitio gracias a un gel conductor o a un gorro de plástico. El examen electroencefalográfico generalmente dura una hora, pero en ocasiones donde se efectúa análisis más exhaustivos como en el caso de problemas con el sueño, puede durar todas las horas del sueño (toda la noche). Generalmente se colocan unos quince pares de electrodos, en el lóbulo de la oreja y en las zonas frontal, temporal, parietal y occipital, así como en el polo frontal y superior del cráneo.

Suelen realizarse varios registros electroencefalográficos, uno en condiciones de reposo que se denomina EEG basal; y otro u otros tras el empleo de procedimientos o maniobras que estimulan la actividad cerebral, lo que se denomina EEG tras estimulación. En este caso, lo más frecuentemente empleado es el estímulo producido por la respiración forzada durante varios minutos o hiperventilación, o por la producción de numerosos flashes luminosos rápidos. La estimulación puede desencadenar una alteración de la actividad cerebral y facilitar el diagnóstico de la enfermedad. Asimismo, puede realizarse un registro continuo del EEG durante 24 horas y durante el sueño para ayudar al diagnóstico.

b) Carga y visualización de datos

Lo primero que se hace para empezar es cambiar nuestro directorio de trabajo a uno en el cual se encuentre el conjunto de datos de las señales EEG adquiridas “dataset” que deseamos analizar. Luego empezamos Matlab, y para incrementar la estabilidad de Matlab, aconsejamos usar sin “java desktop”, “>matlab –nodesktop” desde la línea de comandos de DOS o UNIX. Luego ejecutamos EEGLAB, tal como se indica líneas abajo (Fig. 3.1 y Fig. 3.2).

Para el presente estudio, nuestro directorio de trabajo será el \EEGData, en el cual invocamos al Matlab con la opción matlab –nodesktop”.



```

C:\ Símbolo del sistema
El volumen de la unidad E no tiene etiqueta.
El número de serie del volumen es: 3A4A-1EDE

Directorio de E:\

No se encuentra el archivo

E:\>dir e*
El volumen de la unidad E no tiene etiqueta.
El número de serie del volumen es: 3A4A-1EDE

Directorio de E:\

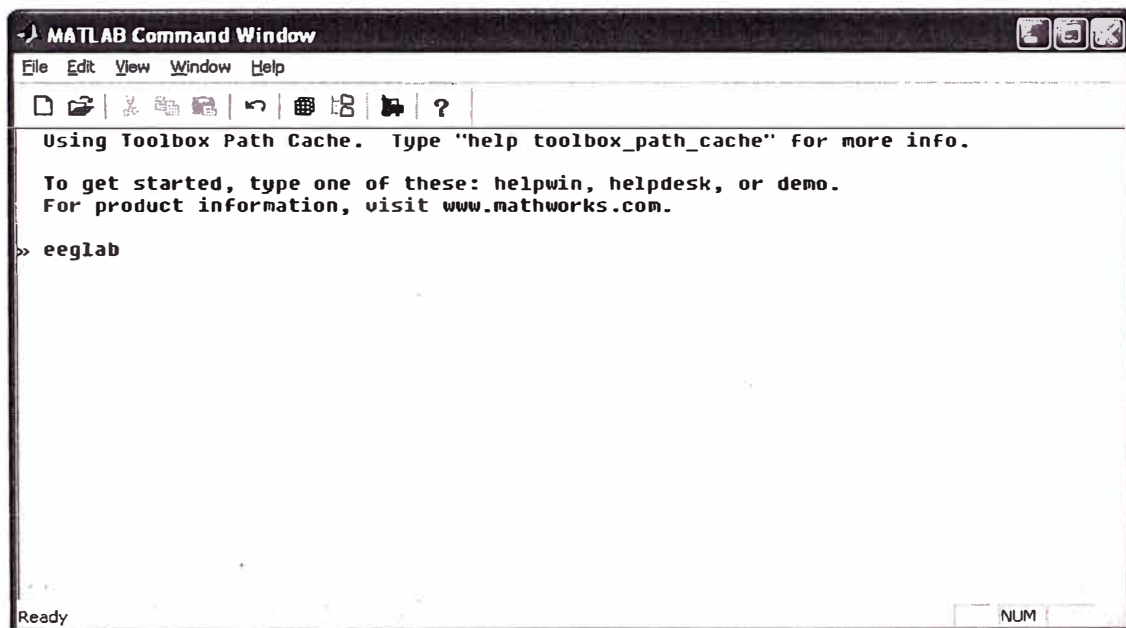
24/09/2004  12:55 p.m.  <DIR>          eeglab
24/09/2004  12:56 p.m.  <DIR>          eeglab4.311
07/01/2005  10:07 p.m.  <DIR>          emoticons
27/02/2005  08:53 p.m.  <DIR>          EEGData
            0 archivos          0 bytes
            4 dirs 26,441,613,312 bytes libres

E:\>cd eegdata
E:\EEGData>matlab -nodesktop
E:\EEGData>

```

Fig. 3. 1 Invocación del MATLAB

Luego invocamos al EEGLab en el Matlab.



```

MATLAB Command Window
File Edit View Window Help
Using Toolbox Path Cache. Type "help toolbox_path_cache" for more info.

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>> eeglab

```

Fig. 3. 2 Invocación del EEGLAB

Luego de este comando emerge la ventana principal del EEGLAB (Fig. 3.3), el cual tiene las siguientes opciones de menú: **File**, **Edit**, **Tools**, **Plot**, **Datasets** y **Help**

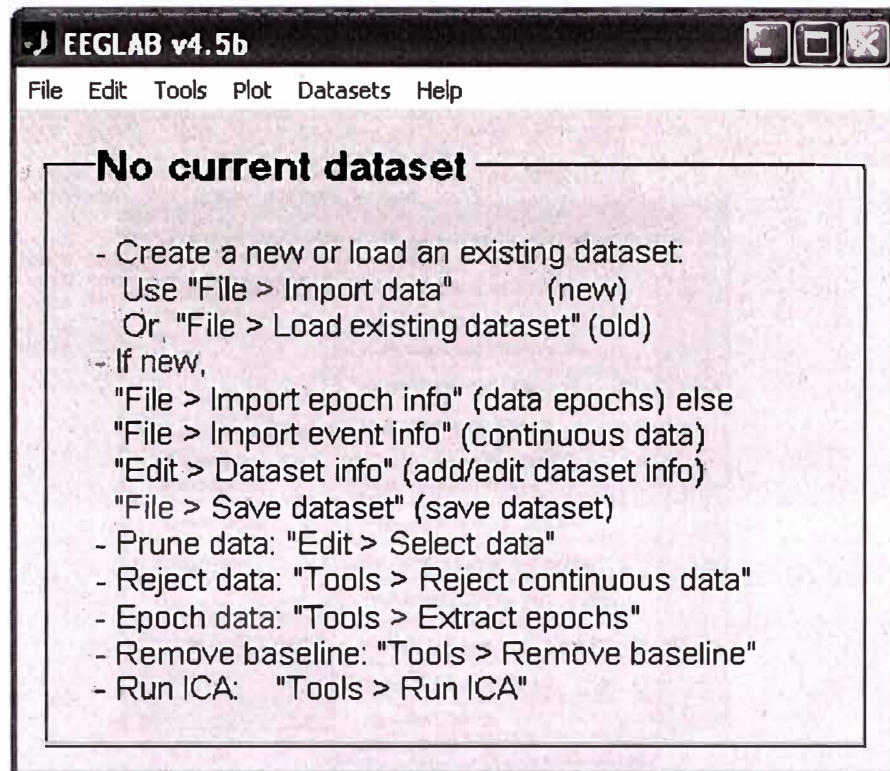


Fig. 3. 3 Ventana de aplicación del EEGLAB

Como se observa en la ventana anterior, el programa EEGLab tiene varias opciones de menú y su estudio se puede efectuar de diferentes modalidades, pero en el presente caso vamos a seguir un procedimiento práctico que nos permita entender fácilmente esta aplicación. En el procedimiento siguiente se tiene la condición previa de tener información adquirida y para ello tenemos la información obtenida del "Equipo electroencefalógrafo del Instituto del Sueño – Hipnos", el cual se encuentra grabado en el directorio "\EEGData" con el nombre de "EEGData_A5r.txt".

- **Importación de datos al eeglab**

Las opciones que dispone el EEGLab para importar datos son muy variadas, soporta formatos ASCII, EGI.RAW, BCI2000, Neuroscan, Biosemi.BDF, etc., tal como se muestra en la ventana de abajo (Fig. 3.4).

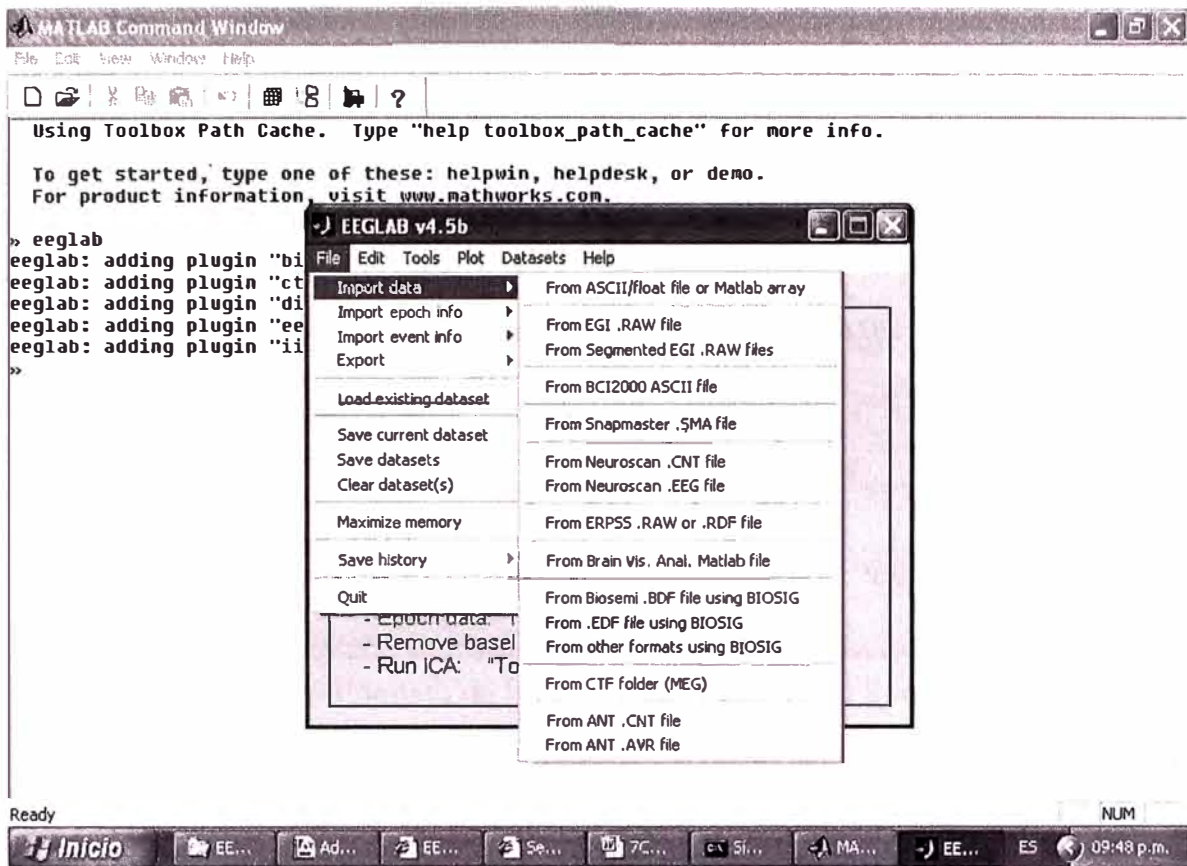


Fig. 3. 4 Menú para importar datos EEG

El procesamiento de los datos EEG a través del EEGLab está preparado para soportar codificaciones estándares, tal como se observó en la ventana anterior, pero el formato ofrecido por el Instituto del Sueño no es soportado, por lo que la información necesita ser adaptada para que puedan ser importados y tratados por este aplicativo.

El formato de las señales EEG que brinda el equipo electroencefalógrafo del Instituto del Sueño es el siguiente (Fig. 3.5):

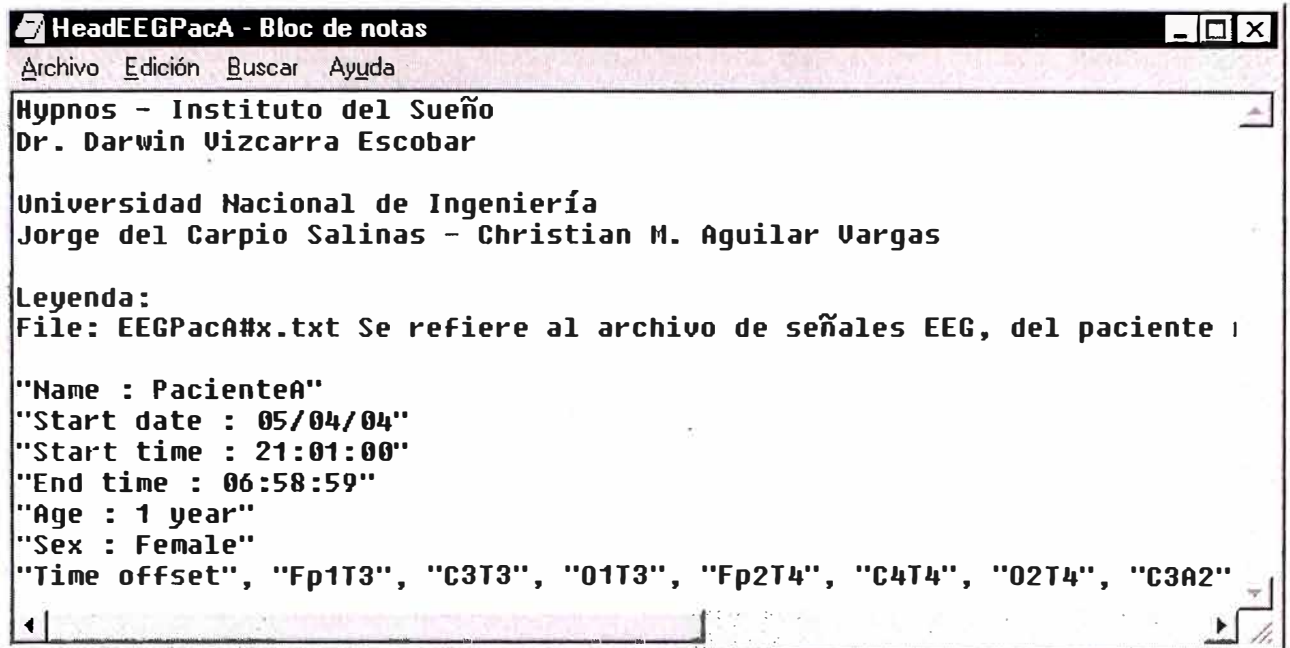


Fig. 3. 5 Formato de los datos del Instituto del Sueño Hypnos

Como se puede observar los datos obtenidos tienen 14 campos los cuales son:

"Time offset", "Fp1T3", "C3T3", "O1T3", "Fp2T4", "C4T4", "O2T4", "C3A2", "C4A1", "Fp1C3", "C3O1", "Fp2C4", "C4O2", "Fp2Fp1d".

El primer campo es el tiempo relativo en el que se tomó la muestra y los 13 campos restantes son los datos correspondientes a electrodos tomados en valores diferenciales entre dos posiciones del cerebro. Según la nomenclatura estándar, por ejemplo el dato de la columna "Fp1T3" corresponde al voltaje diferencial de las posiciones "Frontal Polar 1" (Fp1) y la posición "Temporal 3" (T3)

El procedimiento intermedio que permite adaptar los datos para su análisis, previo a ser importados con el EEGLab, se efectúa directamente con el Matlab aprovechando las herramientas y comandos que dispone este programa, el cual consiste en ejecutar los siguientes pasos: Luego de tener identificado el archivo de los datos EEG (EEGData_A5r.txt), se importa con la opción del Matlab (File -> Show Workspace) y luego en la ventana del WorkSpace con la opción File -> Import Data se selecciona el archivo de los datos EEG, para que sea importado, tal como se muestra en la figura mostrada a continuación (Fig. 3.6).

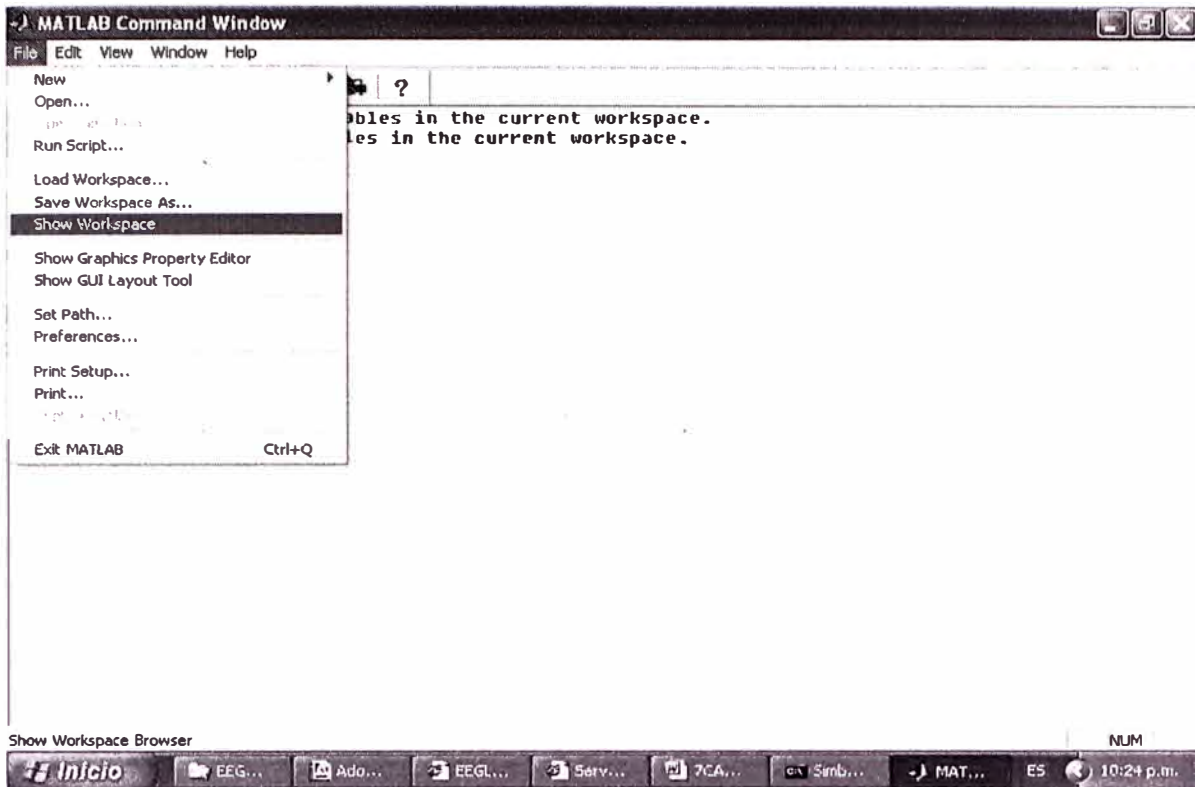


Fig. 3. 6 Menú para inicio del área de trabajo en Matlab

Luego se elige la opción de importar datos, que los efectuamos desde el archivo EEGData_A5r.txt, tal como se muestra a continuación (Fig. 3.7 y Fig. 3.8).

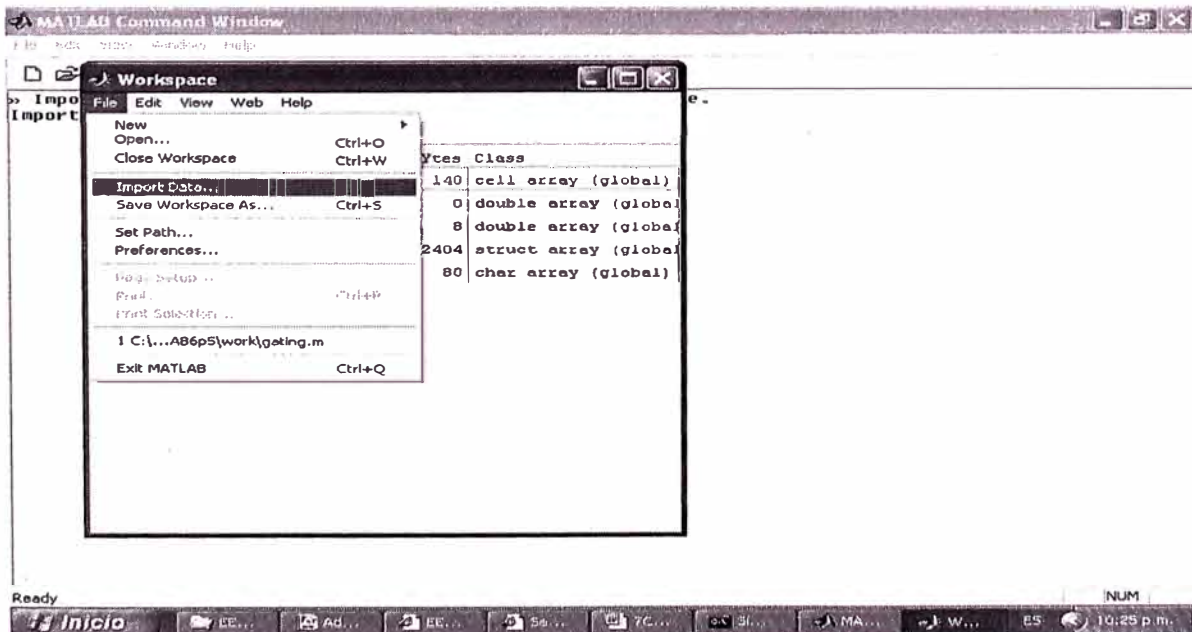


Fig. 3. 7 Opción de menú de importar datos EEG en el Matlab

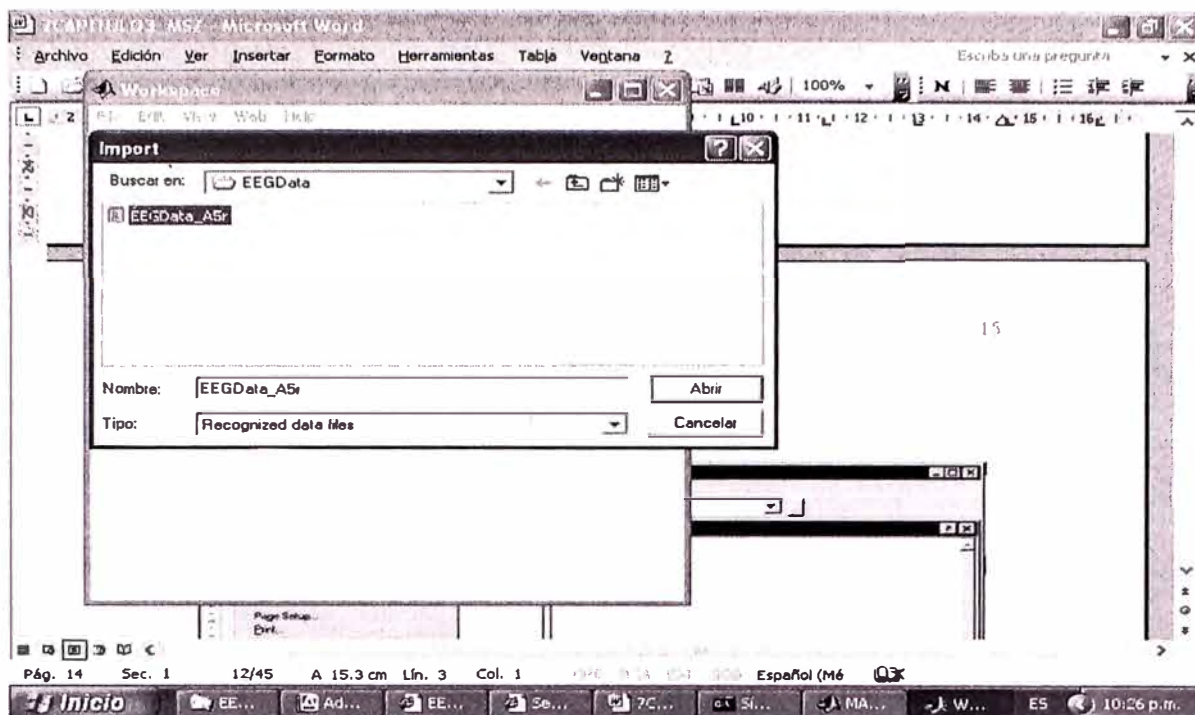


Fig. 3. 8 Selección del archivo de datos EEG

Como se observa, tenemos que la primera columna es la hora de adquisición de los datos y el resto de columnas son los datos, una columna por canal (Fig. 3.9).

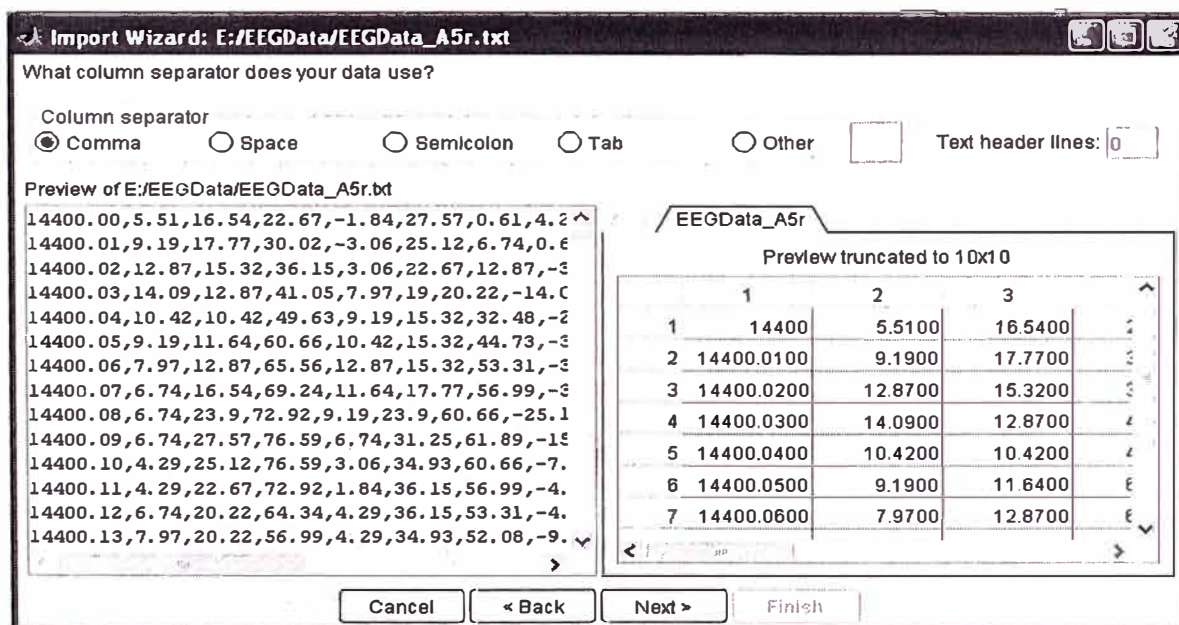


Fig. 3. 9 Datos del archivo EEGData_A5r

Se puede observar (Fig. 3.10) también que el Matlab detecta las columnas que están separadas por comas y crea una variable en el workspace (en el ejemplo lo llamaremos EEGData_A5r). Esta variable es una matriz que tiene 14 columnas (la primera con el tiempo y las otras con los electrodos), formato de información ofrecida por el equipo del Instituto del Sueño - Hypnos.

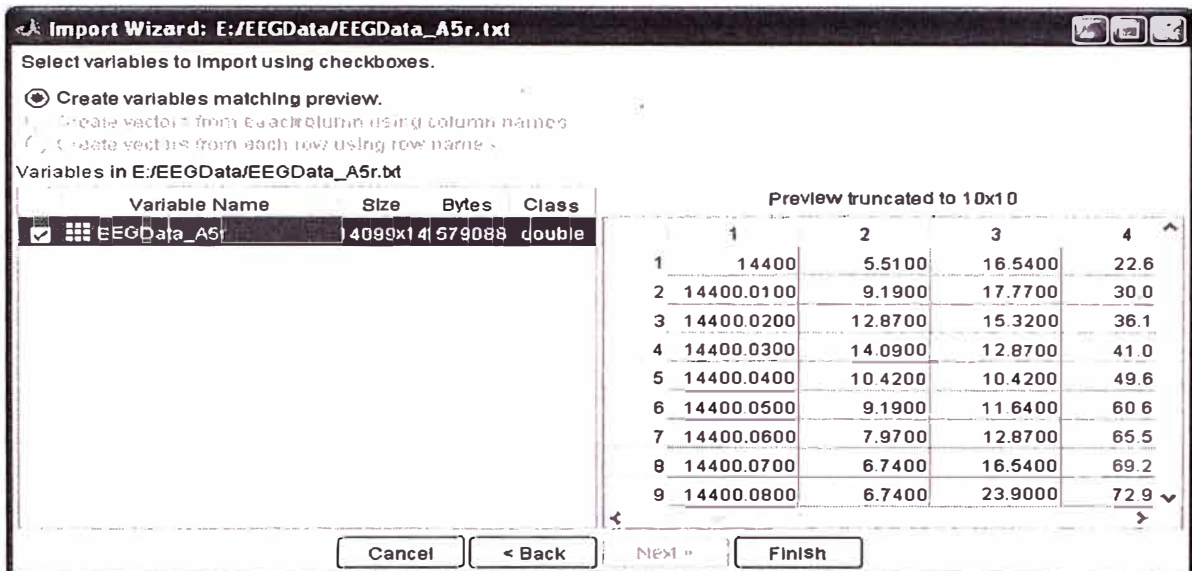


Fig. 3. 10 Matriz de datos EEG cargados en la variable EEGData_A5r del Matlab

La variable asociada del Matlab, se podrá utilizar en este entorno para cualquier cálculo y/u operación, en este caso (Fig. 3.11) vemos que se asoció a la variable EEGData_A5r con los datos 14099x14 (muestras x columnas).

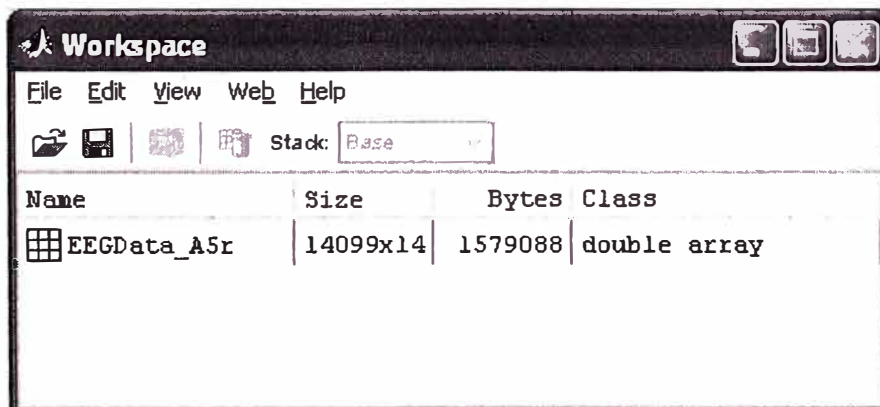


Fig. 3. 11 Características de la variable EEGData_A5r

Cómo la información de la primera columna correspondiente al tiempo es correlativa, y vemos que la del periodo entre cada muestra es de **0.01 segundo**, obtenemos que la frecuencia de muestreo es de **100 Hz**.

Para el análisis la columna del tiempo, ya no es necesaria, por lo que es necesario eliminarla de la variable EEGData_A5r, y solo nos quedamos con la información de los electrodos (para nuestro caso 13 señales), por lo que necesitamos importar solo estas 13 señales. Para conseguir este objetivo efectuamos los siguientes pasos:

Necesitamos que la variable importada, este en el formato que utiliza el EEGLab, por lo que convertiremos los datos en su transpuesta (cambiamos filas por columnas) con el siguiente comando Matlab (Fig. 3.12):

```
EEGData_A5r= EEGData_A5r';
```

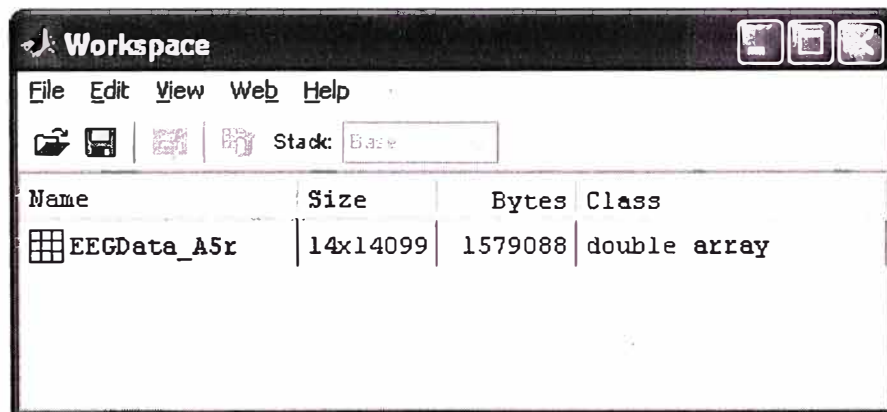


Fig. 3. 12 Variable EEGData_A5r transpuesta

Vemos ahora en el Workspace que la variable EEGData_A5r, se ha transpuesto en una matriz de 14x14099 (columnas x muestras).

Luego hacemos una copia de esta variable en una nueva que contenga solamente las columnas de los electrodos (desde la columna 2 hasta la 14), que para el presente ejemplo son las 13 columnas, con el siguiente comando Matlab (Fig. 3.13):

```
EEGdata= EEGData_A5r (2:14,:);
```


Luego en la ventana "Importdata" desplegada, en el campo "Data file/Array" se pone el nombre de la variable (EEGdata en nuestro caso) y marcando el "sampling rate" con el valor de 100, que es la frecuencia con la que se muestreo estas señales (Fig. 3.15).

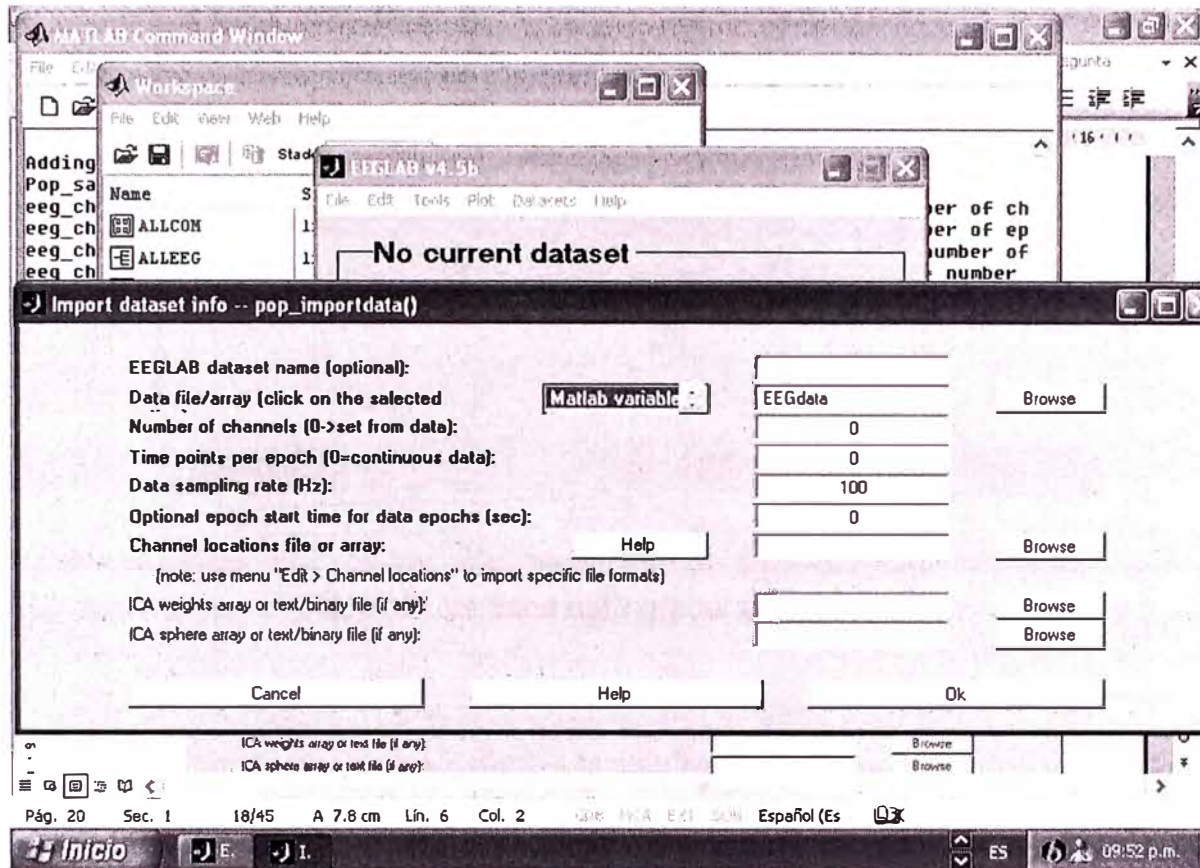


Fig. 3. 15 Formulario de importación de datos EEG

Luego de aceptar estos datos eligiendo la opción "Ok" en la ventana emergente que aparece a continuación, se indica el nombre de la base de datos con la cual se grabará, así como el nombre del archivo con el que se grabará (Fig. 3.16).

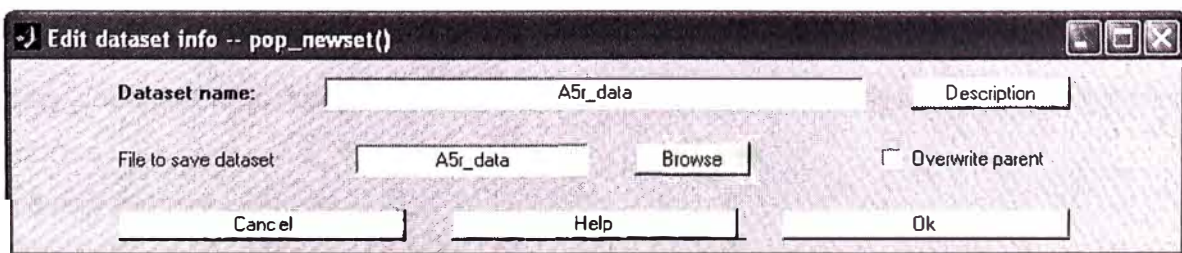


Fig. 3. 16 Asignación del nombre de Base de datos y archivo

También, opcionalmente podemos detallar una descripción que permita recordar que tipo de información estamos guardando, paciente, fecha, etc. (Fig. 3.17).

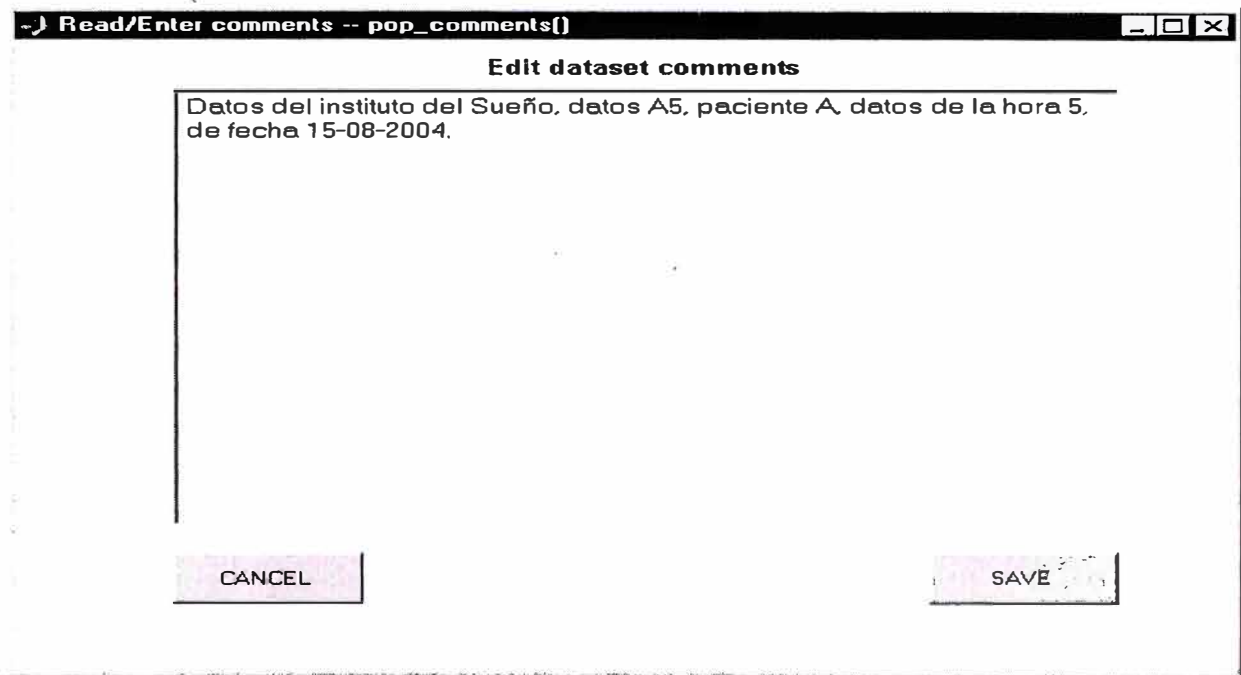


Fig. 3. 17 Ventana para grabar comentarios

Al procesar la importación de los datos, el Matlab reporta las verificaciones de los datos y corrige y/o asume lo que no se ingresó, tal como se muestra en el siguiente reporte:

```
Pop_saveset: Performing extended dataset syntax check...
eeg_checkset warning: number of columns in data (13) does not match the number of
channels (0): corrected
eeg_checkset warning: 3rd dimension size of data (1) does not match the number of
epochs (0), corrected
eeg_checkset warning: number of columns in data (14099) does not match the number of
points (0): corrected
eeg_checkset note: upper time limit (xmax) adjusted so (xmax-xmin)*srate+1 = number of
frames
Saving dataset...
Creating a new dataset with index 1
Done.
```

Finalmente, los datos importados son mostrados en la siguiente figura, en la cual se puede observar los 13 canales importados, la frecuencia de muestreo, el nombre del archivo en el cual se almacena, tramas por *epoch*, Nro de *Epochs*, etc. (Fig. 3.18).

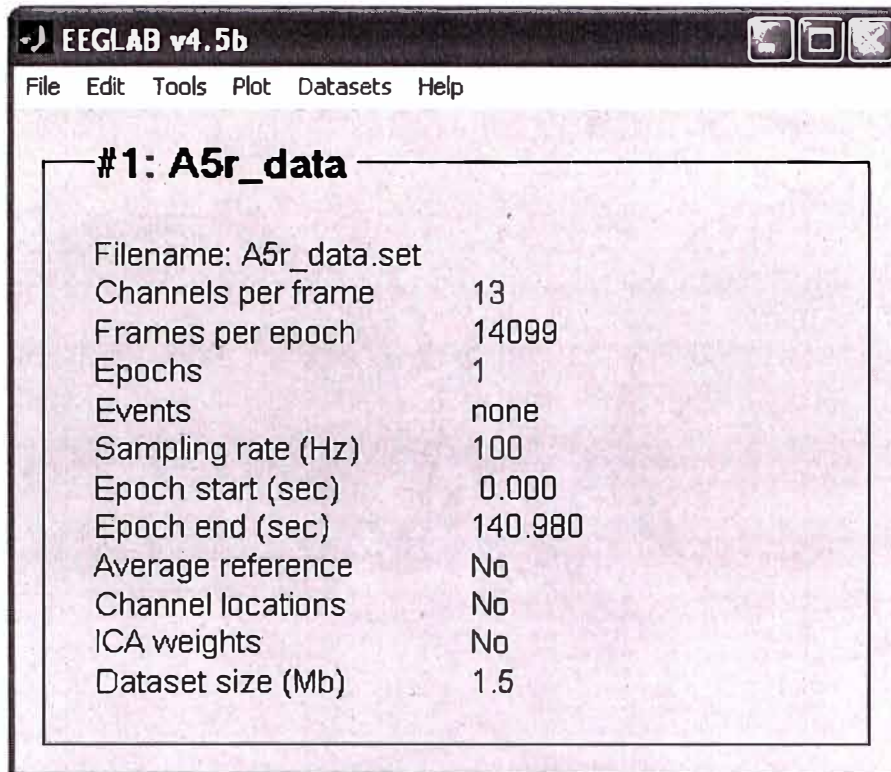


Fig. 3. 18 Ventana con la información de la carga de los datos EEG

Luego de haber importado los datos del paciente, necesitamos ingresar la información de las coordenadas correspondientes a cada electrodo cargado en los trece canales, pero para ello necesitamos entender el formato que utiliza el EEGLAB para identificar estos parámetros, así cuales son las formas de las ubicaciones de cada electrodo.

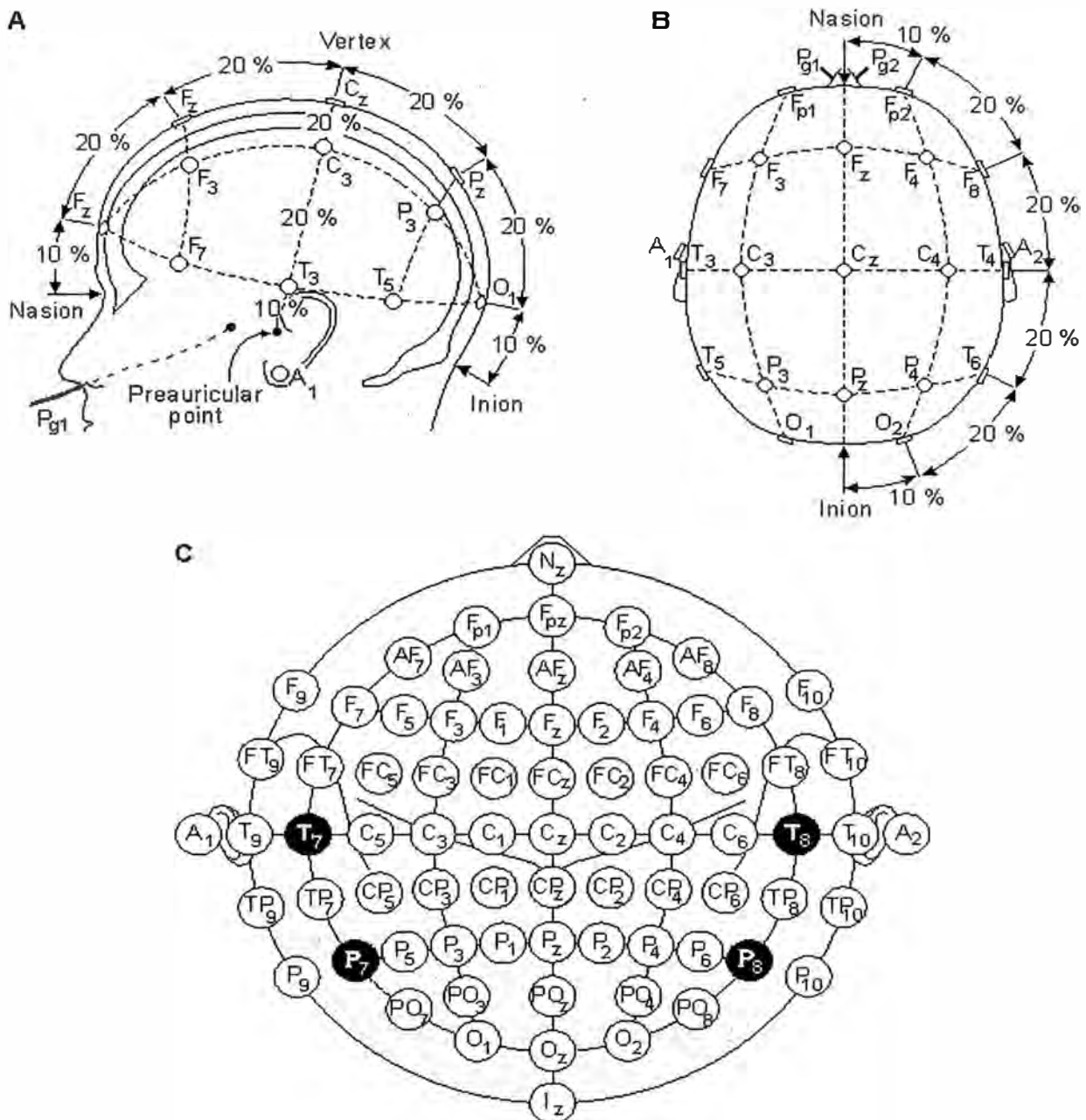
- **Los electrodos en el EEGLAB**

Las señales de los electrodos ya importados, son vistas en el EEGLab como canales, y cada uno de ellos debe de ser identificado de acuerdo a la nomenclatura estándar de nombres de acuerdo a su posición de adquisición, esta nomenclatura y forma se muestra en la figura que viene a continuación, así como las formas de conexión que se pueden realizar y las coordenadas que se pueden elegir.

Esto nos servirá para comprender y crear nuestro archivo de coordenadas que nos permitirá analizar las muestras importadas.

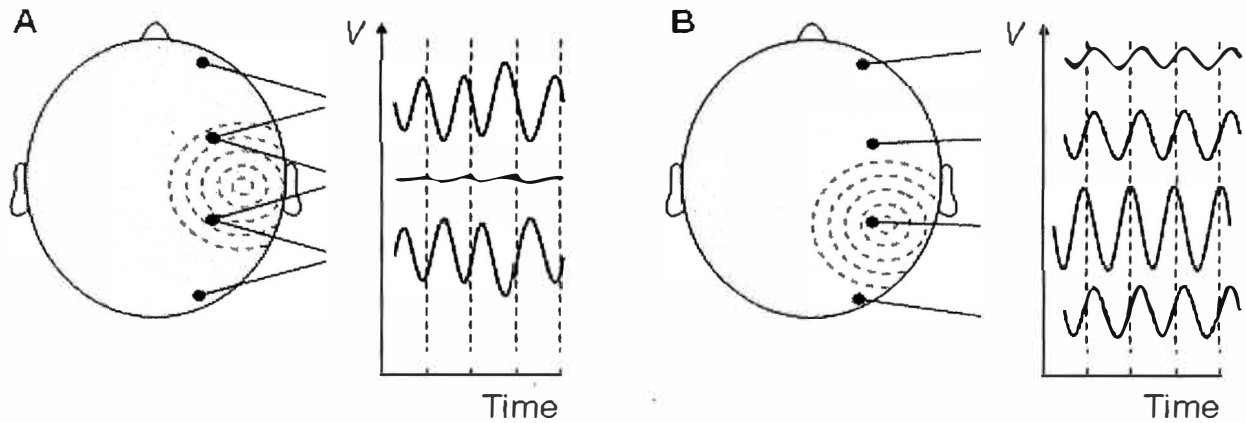
En la figura mostrada a continuación (Fig. 3.19 y Fig. 3.20), vemos las ubicaciones de los electrodos y los nombres de cada uno de ellos. Estos electrodos están mostrados en una

vista lateral (visto del lado izquierdo), vista de arriba de la cabeza y también se muestra una vista del 10% de las localizaciones intermedias estandarizadas por la American Electroencephalographic Society. (Redrawn from Sharbrough, 1991). Según el propósito de la adquisición de las muestras (análisis del sueño), los electrodos utilizados en nuestro análisis son solo 13 de las mostradas.



El Sistema Internacional 10-20 visto desde la izquierda (A) y arriba de la cabeza (B). A=lóbulo de la oreja, C=central, Pg = nasofaríngeal, P = parietal, F = frontal, Fp = frontal polar, O = occipital. (C) Localización y nomenclatura de los 10% de los electrodos intermedios, el cual ha sido estandarizado por la American Electroencephalographic Society. (Redrawn from Sharbrough, 1991).

Fig. 3. 19 Electrodes en el EEG LAB – Sistema Internacional



(A) Bipolar (B) Medida unipolar. Notar que la forma de onda del EEG depende de la localización de las mediciones

Fig. 3. 20 Electrodo en el EEGLAB – Formas de Onda

Los siguientes son ejemplos de formatos de archivos de localización de los canales "channel location":

Coordenadas polares.- El siguiente es un ejemplo de un archivo de coordenadas polares, el archivo tiene una extensión locs. (Tabla 3.1).

Tabla 3. 1 Localizaciones en coordenadas polares

#	Deg.	Radius	Label
1	-18	.352	Fp1
2	18	.352	Fp2
3	-90	.181	C3
4	90	.181	C4

Coordenadas esféricas.- Las mismas localizaciones pero en formato de coordenadas Esféricas, el archivo tiene una extensión .sph. (Tabla 3.2).

Tabla 3. 2 Localizaciones en coordenadas esféricas

#	Azimet	Horiz.	Label
1	-63.36	-72	Fp1
2	63.36	72	Fp2
3	32.58	0	C3
4	32.58	0	C4

Coordenadas cartesianas.- Las mismas localizaciones pero en formato de coordenadas Cartesianas, el archivo tiene una extensión .xyz. (Tabla 3.3).

Tabla 3. 3 Localizaciones en coordenadas cartesianas

#X	Y	Z	Label
1	-0.8355	-0.2192	Fp1
2	-0.8355	0.2192	Fp2
3	0.3956	0	C3
4	0.3956	0	C4

- **Preparación del archivo de coordenadas**

Para importar las coordenadas de los electrodos, primero vamos a preparar un archivo de acuerdo a los archivos estándares utilizados, adaptando a la cantidad de canales que tenemos adquiridos, que para este caso es de 13 canales.

Del estándar de coordenadas polares "**Standard-10-20-Cap19.locs**", adaptamos a 13 coordenadas de acuerdo a la localización creando el archivo "**prueba_13elecs.locs**", cuyo contenido es el siguiente (Tabla 3.4):

Tabla 3. 4 Coordenadas del archivo Prueba_13elecs.locs

1	-18	0.51111	Fp1
2	18	0.51111	Fp2
3	-54	0.51111	F7
4	-39	0.33333	F3
5	0	0.25556	Fz
6	39	0.33333	F4
7	54	0.51111	F8
8	-90	0.51111	T7
9	-90	0.25556	C3
10	90	0	Cz
11	90	0.25556	C4
12	90	0.51111	T8
13	-126	0.51111	P7

- **Importar las coordenadas de los electrodos**

Luego procedemos a importar las coordenadas desde el archivo "**prueba_13elecs.locs**", siguiendo los siguientes pasos (Fig. 3.21): elegimos el menú "**edit**" y submenú "**channel locations**" y elegimos la opción de "**Read locations**".

Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")

Polar angle ("theta")

Polar radius ("radius")

Cartesian X ("X")

Cartesian Y ("Y")

Cartesian Z ("Z")

Spherical horiz. angle ("sph_theta")

Spherical azimuth angle ("sph_phi")

Spherical radius ("sph_radius")

Opt. 3d center

Rotate axis

Transform axes

XYZ -> polar && sph.

Sph. -> polar && xyz

Polar -> sph. && xyz

Delete chan

Inset chan

Channel number (of 1)

Append chan

Plot 2d

Plot radius (0.2-1, []=auto)

Shrink to center

Plot 3d (xyz)

Read locations

Read help

Save .ced

Save others

Cancel

Help

Ok

Fig. 3. 21 Formulario para importar coordenadas

En esta ventana, elegimos la opción **"Read locations"**, la cual nos muestra una ventana para elegir el archivo de localizaciones y en esta opción elegiremos el archivo **"prueba__13elecs.locs"** que preparamos (Fig. 3.22).

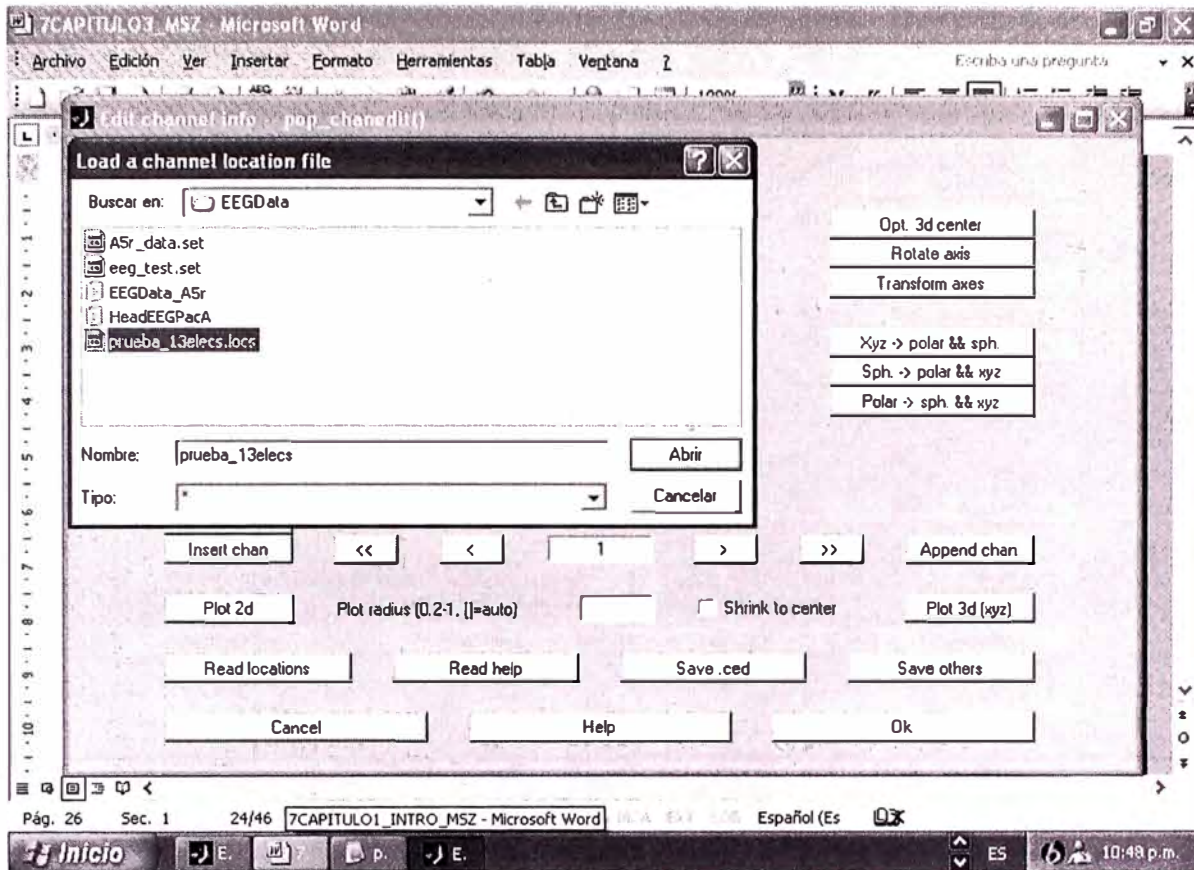


Fig. 3. 22 Importación de coordenadas desde el archivo prueba_13elecs.locs

Luego de elegir el archivo, elegimos la opción “loc eeglab”, en la ventana “Read electrode file”, tal como se muestra a continuación (Fig. 3.23):

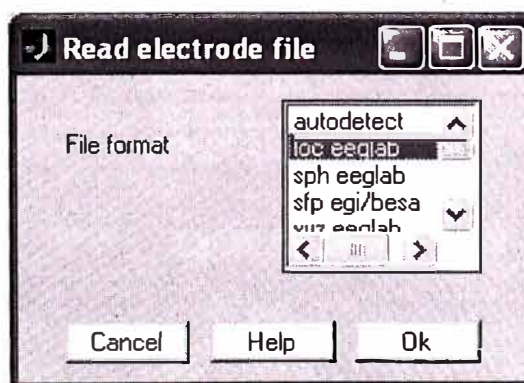


Fig. 3. 23 Ventana de elección del tipo de coordenadas

Con lo cual obtenemos los canales con sus respectivos valores de coordenadas, tal como se muestra para los canales 1, 2, 3 y 13, a continuación en las siguientes figuras (Fig. 3.24, Fig. 3.25, Fig. 3.26 y Fig. 3.27).

Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")	Fp1	Opt. 3d center
Polar angle ("theta")	-18	Rotate axis
Polar radius ("radius")	0.51111	Transform axes
Cartesian X ("X")	0.95048	XYZ -> polar && sph.
Cartesian Y ("Y")	0.30883	Sph. -> polar && xyz
Cartesian Z ("Z")	-0.034896	Polar -> sph. && xyz
Spherical horiz. angle ("sph_theta")	18	
Spherical azimuth angle ("sph_phi")	-1.9998	
Spherical radius ("sph_radius")	1	

Delete chan | Channel number (of 13) | Append chan

Insert chan | << | < | 1 | > | >>

Plot 2d | Plot radius (0.2-1, []=auto) | Shrink to center | Plot 3d (xyz)

Read locations | Read help | Save .ced | Save others

Cancel | Help | Ok

Fig. 3. 24 Coordenadas del Canal 1 importadas

Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")	Fp2	Opt. 3d center
Polar angle ("theta")	18	Rotate axis
Polar radius ("radius")	0.51111	Transform axes
Cartesian X ("X")	0.95048	XYZ -> polar && sph.
Cartesian Y ("Y")	-0.30883	Sph. -> polar && xyz
Cartesian Z ("Z")	-0.034896	Polar -> sph. && xyz
Spherical horiz. angle ("sph_theta")	-18	
Spherical azimuth angle ("sph_phi")	-1.9998	
Spherical radius ("sph_radius")	1	

Delete chan | Channel number (of 13) | Append chan

Insert chan | << | < | 2 | > | >>

Plot 2d | Plot radius (0.2-1, []=auto) | Shrink to center | Plot 3d (xyz)

Read locations | Read help | Save .ced | Save others

Cancel | Help | Ok

Fig. 3. 25 Coordenadas del Canal 2 importadas

Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")	F7	Opt. 3d center
Polar angle ("theta")	-54	Rotate axis
Polar radius ("radius")	0.51111	Transform axes
Cartesian X ("X")	0.58743	
Cartesian Y ("Y")	0.80852	Xyz -> polar && sph.
Cartesian Z ("Z")	-0.034896	Sph. -> polar && xyz
Spherical horiz. angle ("sph_theta")	54	Polar -> sph. && xyz
Spherical azimuth angle ("sph_phi")	-1.9998	
Spherical radius ("sph_radius")	1	

Delete chan Channel number (of 13)

Insert chan << < 3 > >> Append chan

Plot 2d Plot radius (0.2-1, []=auto) Shrink to center Plot 3d (xyz)

Read locations Read help Save .ced Save others

Cancel Help Ok

Fig. 3. 26 Coordenadas del Canal 3 importadas

Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label")	P7	Opt. 3d center
Polar angle ("theta")	-126	Rotate axis
Polar radius ("radius")	0.51111	Transform axes
Cartesian X ("X")	-0.58743	
Cartesian Y ("Y")	0.80852	Xyz -> polar && sph.
Cartesian Z ("Z")	-0.034896	Sph. -> polar && xyz
Spherical horiz. angle ("sph_theta")	126	Polar -> sph. && xyz
Spherical azimuth angle ("sph_phi")	-1.9998	
Spherical radius ("sph_radius")	1	

Delete chan Channel number (of 13)

Insert chan << < 13 > >> Append chan

Plot 2d Plot radius (0.2-1, []=auto) Shrink to center Plot 3d (xyz)

Read locations Read help Save .ced Save others

Cancel Help Ok

Fig. 3. 27 Coordenadas del Canal 13 importadas

Luego de importado los datos de localización de los canales, es decir las coordenadas polares del archivo que preparamos, vemos ahora que el EEGLAB, los ha recalculado también en coordenadas cartesianas y esféricas.

Con la opción del menú “**Plot>Channel locations>By name**” podemos ver la gráfica de la ubicación de los 13 electrodos (Fig. 3.28), solo que como varias de las muestras se han tomado relacionadas con puntos comunes, esto se refleja en la gráfica superponiendo varios indicadores.

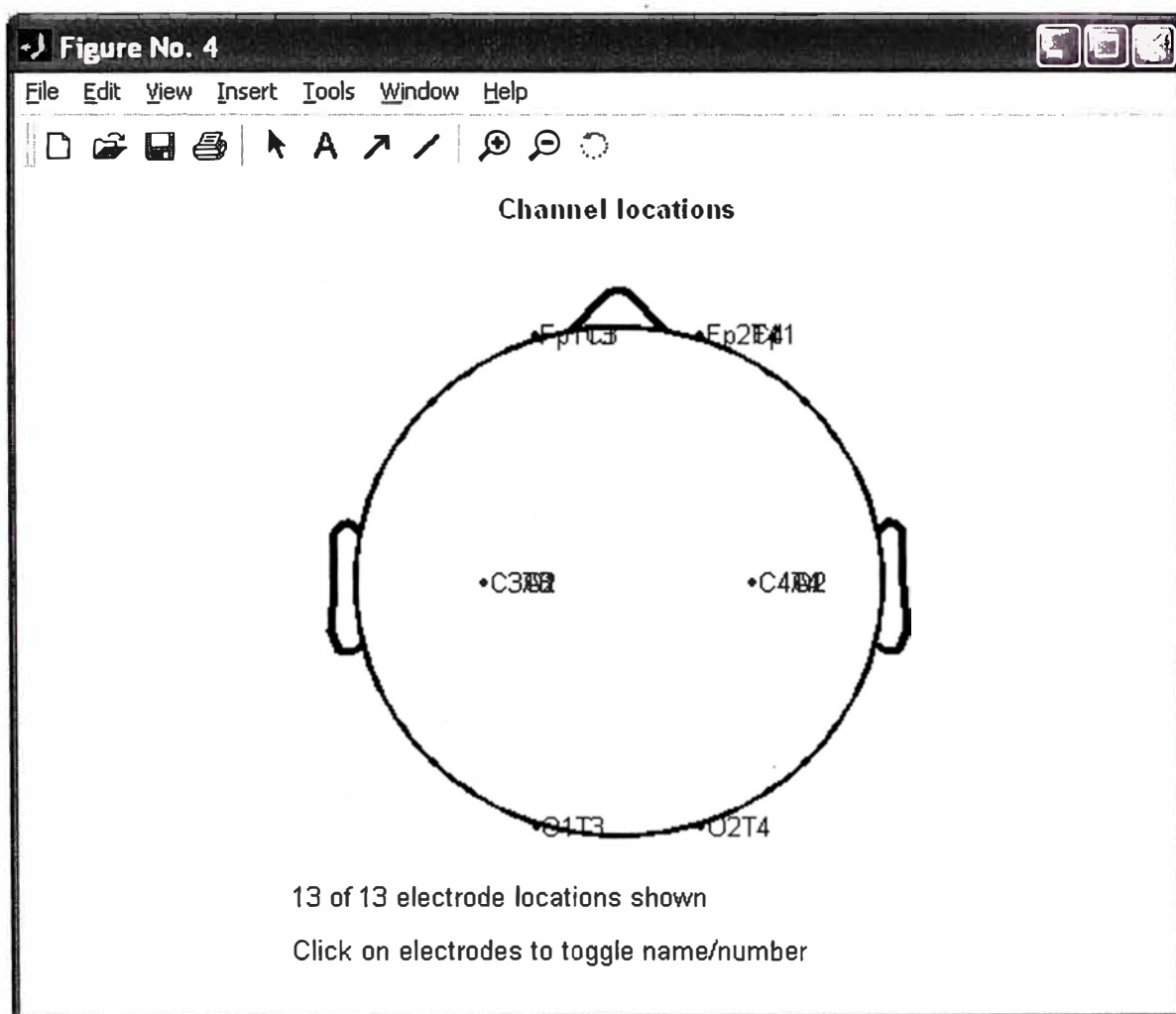


Fig. 3. 28 Ubicación de los 13 electrodos en la cabeza

Así mismo, podemos ya observar las señales correspondientes a los 13 electrodos (Fig. 3.29), visualizándolos con la opción “**Plot>Channel data (scroll)**”.

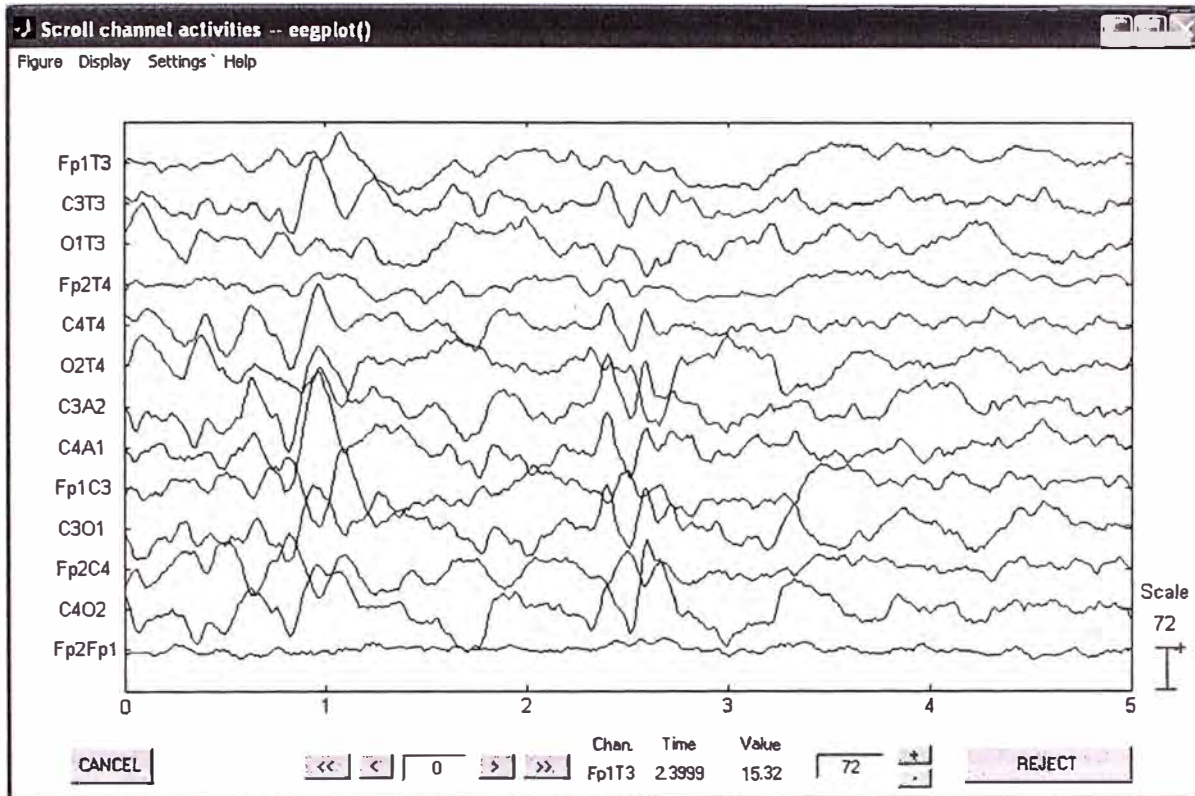


Fig. 3. 29 Gráfica de las 13 señales EEG

- **Filtrado de datos**

Con la finalidad de eliminar las tendencias lineares de los datos, pasamos los datos por un filtro pasa alto, con la frecuencia de corte inferior de 1 Hz. Para ello elegimos el menú **“Tools > Filter the data”** desplegándose la siguiente Fig. 3.30 y Fig. 3.31:

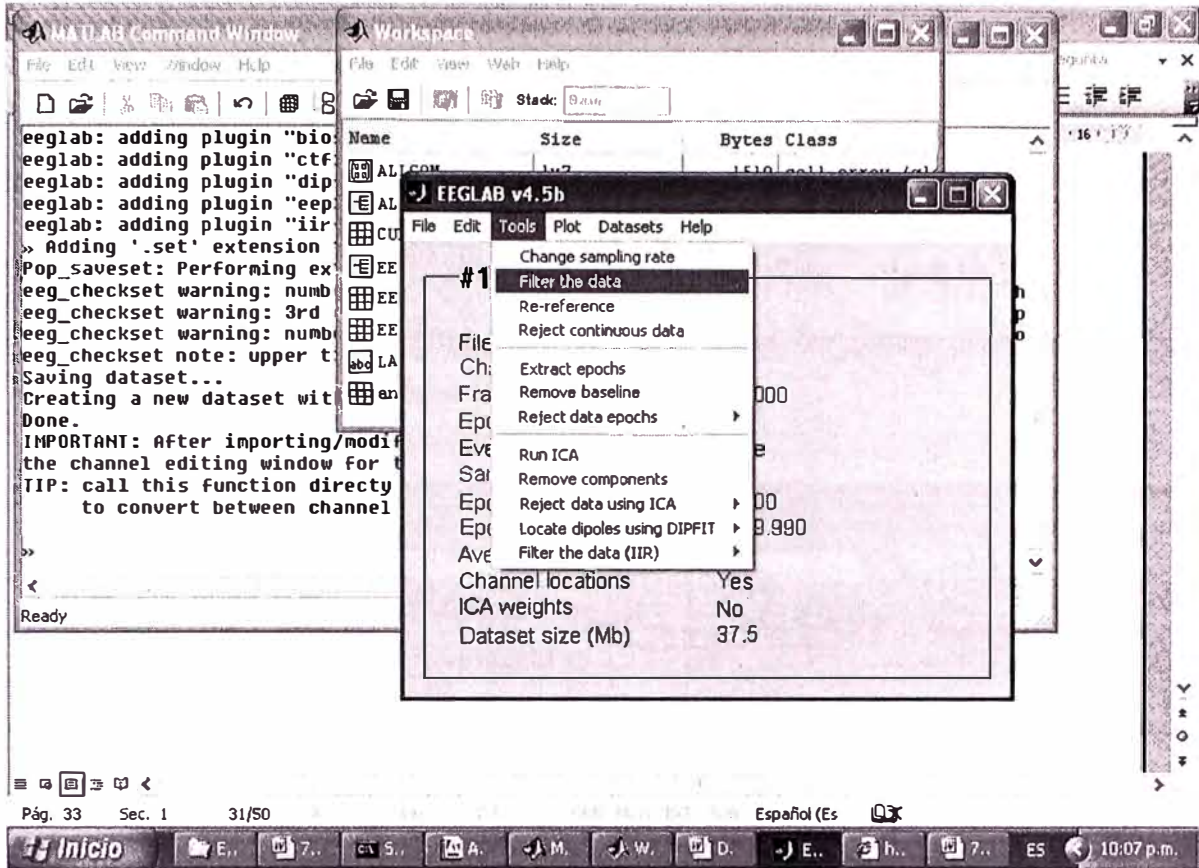


Fig. 3. 30 Opción de menú de filtrado

The dialog box titled "Filter the data -- pop_eegfilt()" contains the following fields and buttons:

- Highpass: lower edge of the frequency pass band (Hz) (0 -> lowpass)
Input field: 1
- Lowpass: higher edge of the frequency pass band (Hz) (0 -> highpass)
Input field: 0
- Notch filter the data. Give the notch range, i.e. [45 55] for 50 Hz
(this option overwrites the low and high edge limits given above)
Input field: []
- Filter length in points (default: see >> help pop_eegfilt)
Input field: []
- Buttons: Cancel, Help, Ok

Fig. 3. 31 Formulario del filtro Pasa alto / Pasa bajo

En el matlab vemos el siguiente reporte de la ejecución del filtro.

```
eegfilt() - performing 300-point highpass filtering.
eegfilt() - highpass transition band width is 0.1 Hz.
```

Se tiene la posibilidad de almacenar en una base de datos diferente que permite salvaguardar los datos originales (Fig. 3.32).

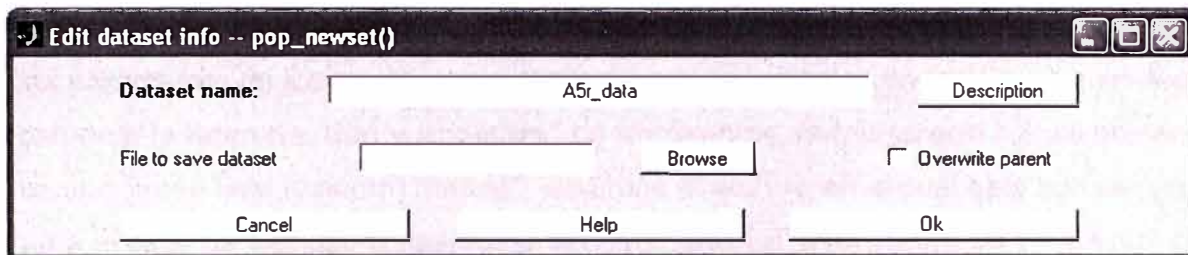


Fig. 3. 32 Formulario para definir el nombre de la base de datos

Cabe mencionar que esta función de filtrado también tiene la posibilidad de filtrar los 60 Hz de la corriente eléctrica.

- **Estructura de Eventos**

Con la finalidad de facilitar el análisis de las señales EEG, EEGLab ha diseñado una estructura de datos que maneja los eventos que se van presentando en el momento de la adquisición de los datos de los electrodos. La información de los eventos para datos EEG continuos es almacenado en campos que dan información relevante para cada evento. Para usar eventos en una base de datos EEGLAB, tiene definido dos campos obligatorios: **“type”** y **“latency”**, y se puede utilizar uno o más campos con cualquier información adicional del usuario, los cuales pueden ser números.

En el caso del análisis que estamos efectuando, generalmente tenemos tres campos llamados **“type”**, **“position”**, y **“latency”**, estos tres datos son adjuntados a cada evento.

El campo **“type”** indica siempre el tipo del evento.

El campo **“latency”** indica siempre la latencia del evento en segundos desde el inicio de los datos continuos.

Es importante entender que los nombres de los campos son definidos por el usuario al crear la base de datos **“dataset”**, y es posible crear, guardar, y cargar tantos campos eventos como se desee.

Se debe tener presente que “type” y “latency” (minúsculas) son dos palabras claves que el EEGLAB reconoce y estos campos tienen que estar definidos por el usuario, excepto si se importa información de eventos de “epochs” (nota: si solamente se define el campo “latency”, el campo “type” puede ser definido como una constante con el valor por defecto de “1” para cada evento). A menos que no sean definidos estos dos campos, EEGLAB no estará habilitado para manejar apropiadamente los eventos al extraer las “epochs”, dibujar los tiempos de reacción, etc.

En el caso de la asociación de los eventos con los datos importados, tenemos varias formas de definir los eventos, uno de los cuales es, importando el archivo que contiene los parámetros de los eventos que para el caso del análisis contiene los siguientes campos “la latencia, tipo y posición” de los eventos, nombres que colocamos en el campo “Input field (column) names”, elegimos el archivo en el cual está sus valores, que para el caso de ejemplo utilizamos el archivo “tutorial_eventtable.txt”, eligiendo para ello el campo “Event file or array” de la siguiente opción “file > Import event info” ventana (Fig. 3.33).

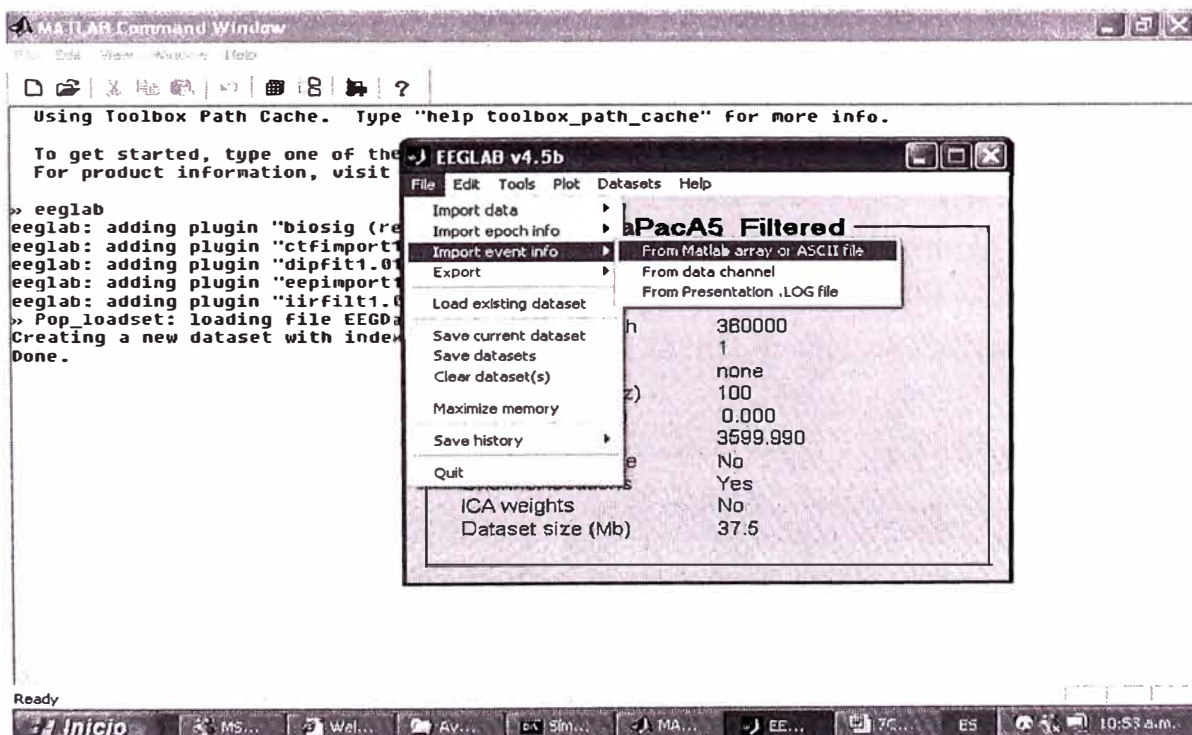


Fig. 3. 33 Menú de importación de información de eventos

Luego, haciendo click en la opción “Browse” seleccionamos el archivo “tutorial_eventtable.txt” (Fig. 3.34).

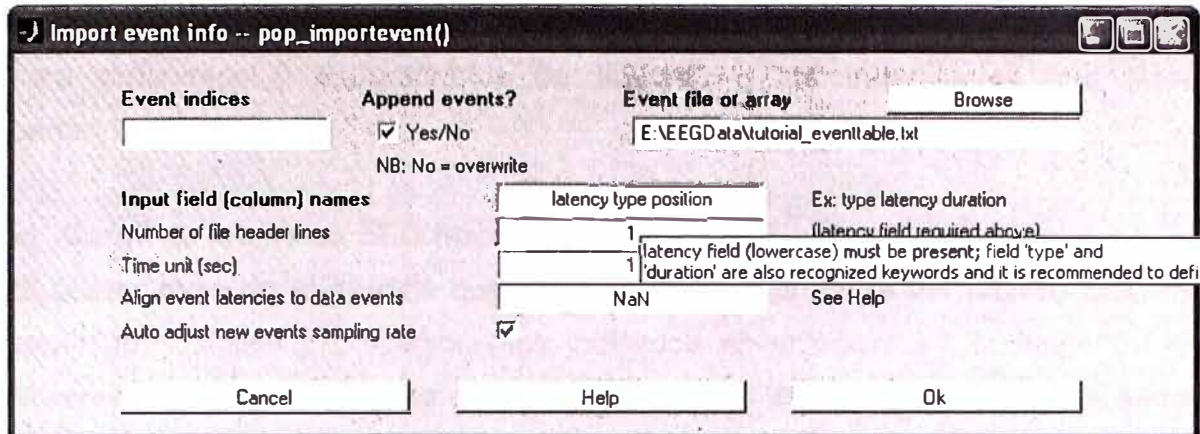


Fig. 3. 34 Formulario de importación de información de eventos

El Matlab reporta al crear estos eventos:

Reading file (lines): 10 20 30 40 50 60 70 80 90 100 100

eeg_checkset note: creating the original event table (EEG.urevent)

Done.

Ahora podemos ver los campos de las señales EEG con el campo “Events” con el valor de 100, tal como se observa en la Fig. 3.35.

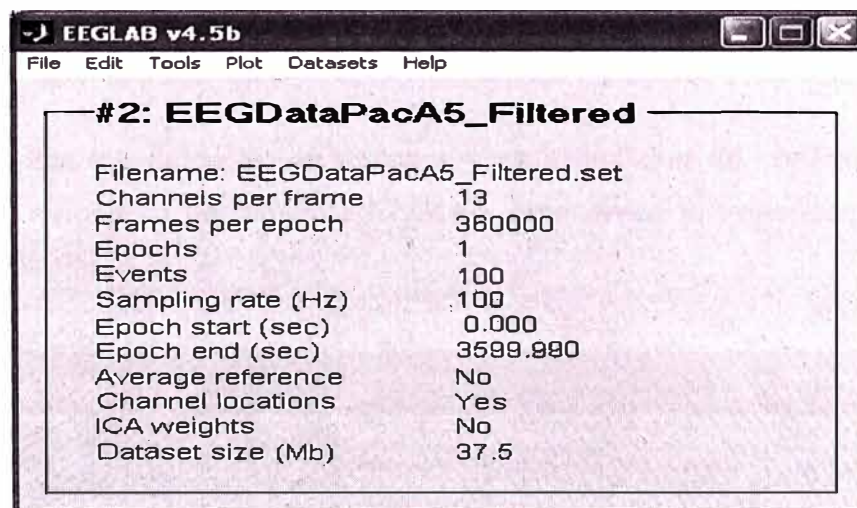


Fig. 3. 35 Reporte de los eventos importados

3.3.3. Análisis y procesamiento con Herramienta EEGLAB

En el presente caso, y debido a que tenemos solamente la disponibilidad de la información del Instituto del Sueño “Hypnos”, realizaremos el análisis y procesamiento de los datos de información adquirida en dicho instituto, datos que tiene un tamaño de 31Mbytes y están almacenados en el archivo **EEGPacA5.txt**, estos datos han sido adquiridos de un paciente al que denominaremos “A5”. El formato con el cual se ha

almacenado no es estándar, por lo que nos ayudaremos con instrucciones del Matlab para ordenarlos y estructurarlos de acuerdo a los tratamientos que dispone la herramienta EEGLab.

a) Carga de los datos EEG hacia la aplicación EEGLab

El primer paso en el análisis que efectuaremos es la carga del archivo EEGPacA5.txt según los detalles y procedimientos indicados en el punto 3.3.2., luego de la carga observamos en el Workspace del Matlab la variable de nombre EEGPacA5, tal como se puede observar en la Fig. 3.36.

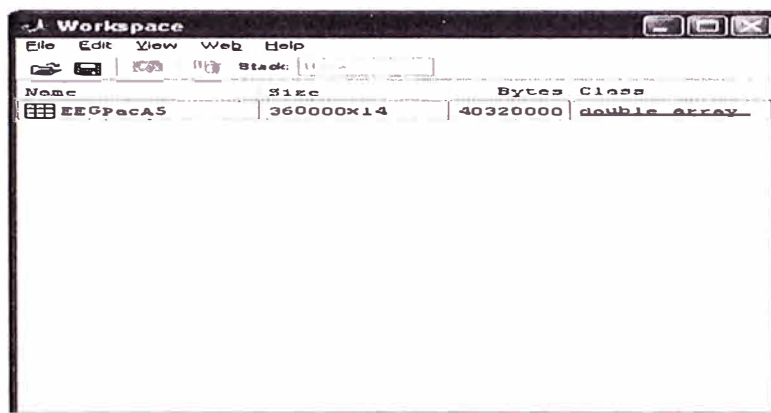


Fig. 3. 36 Datos de paciente EEGPacA5 cargados al Matlab

Luego de cargado los datos en el Matlab y con la finalidad de configurar estos en la estructura que maneja la herramienta EEGLab, obtenemos la transpuesta, cuyo detalle se muestra en la Fig. 3.37

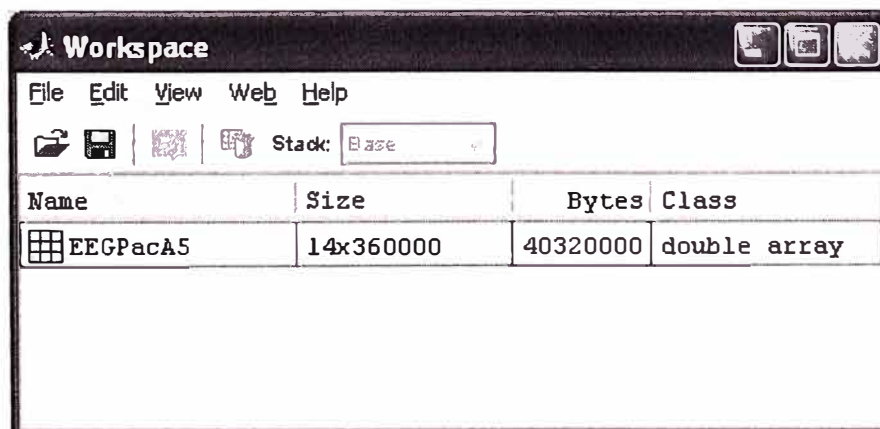


Fig. 3. 37 Transpuesta de los datos EEGPacA5

Luego de efectuado la transpuesta, obtenemos una nueva variable de estos datos eliminando la columna 1, columna que contiene el número secuencial de los datos, el

cual solo muestra como información la frecuencia de muestreo de 100 mseg, valor que utilizaremos posteriormente.

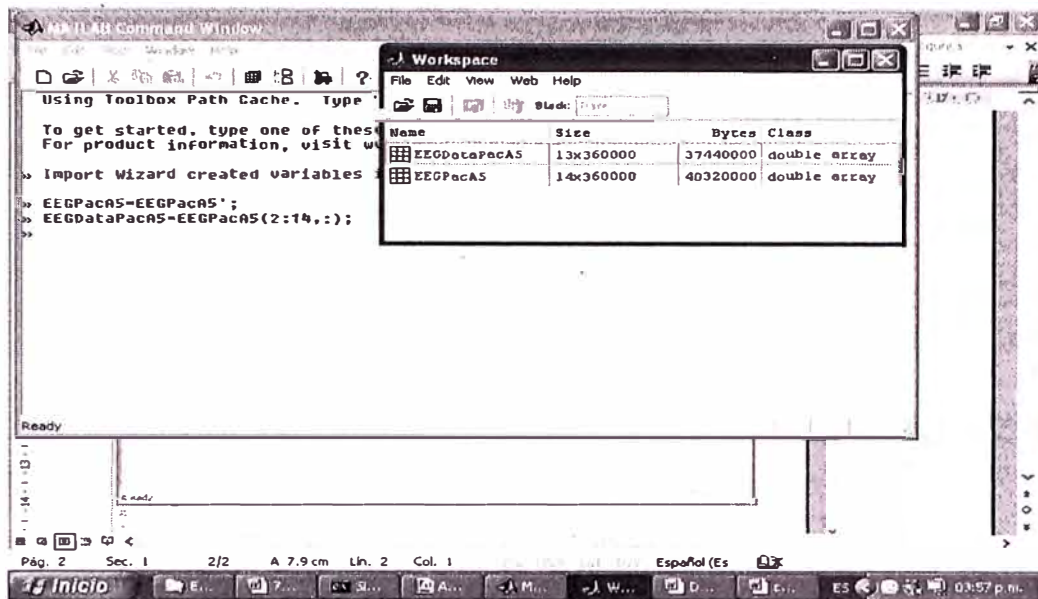


Fig. 3. 38 Nueva variable EEGDataPacA5 con canal 1 eliminado

Luego se ejecuta el aplicativo "eeglab" dentro del Matlab, observando en el Workspace los datos y estructuras correspondientes a este aplicativo, tal como se muestra en la Fig. 3.39.

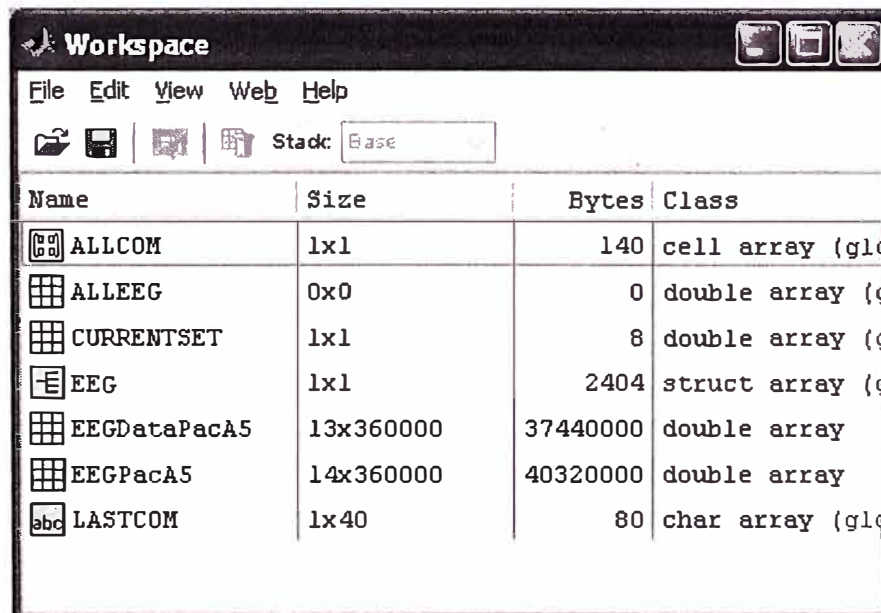


Fig. 3. 39 Variables luego de ejecutado el EEGLAB

Continuando con el proceso de carga de datos, se importa desde la variable EEGDataPacA5 los datos cargados por el Matlab hacia el "eeglab" desde el menú File >

Import data > From ASCII/float file or Matlab array, tal como se muestra en la Fig. 3.40.

EEGLAB dataset name (optional):

Data file/array (click on the selected **Matlab variable**): EEGDataPecA5

Number of channels (0->set from data): 0

Time points per epoch (0=continuous data): 0

Data sampling rate (Hz): 100

Optional epoch start time for data epochs (sec): 0

Channel locations file or array:

(note: use menu "Edit > Channel locations" to import specific file formats)

ICA weights array or text/binary file (if any):

ICA sphere array or text/binary file (if any):

Fig. 3. 40 Formulario de importación de datos EEG desde variable matlab

El siguiente paso es cargar el archivo de localización de los electrodos utilizados en la adquisición de datos del paciente "A5", este archivo lo configuramos de acuerdo a la nomenclatura de Electrodos estándar 10-20 y para ello utilizamos el menú **Edit > Channel locations** y luego se elige la opción de **"Read locations"**, tal como se muestra en las figuras Fig. 3.41 y Fig. 3.42 mostradas a continuación.

Channel information ("field_name"):

Channel label ("label")

Polar angle ("theta")

Polar radius ("radius")

Cartesian X ("X")

Cartesian Y ("Y")

Cartesian Z ("Z")

Spherical horiz. angle ("sph_theta")

Spherical azimuth angle ("sph_phi")

Spherical radius ("sph_radius")

Fig. 3. 41 Formulario de importación de localizaciones de electrodos

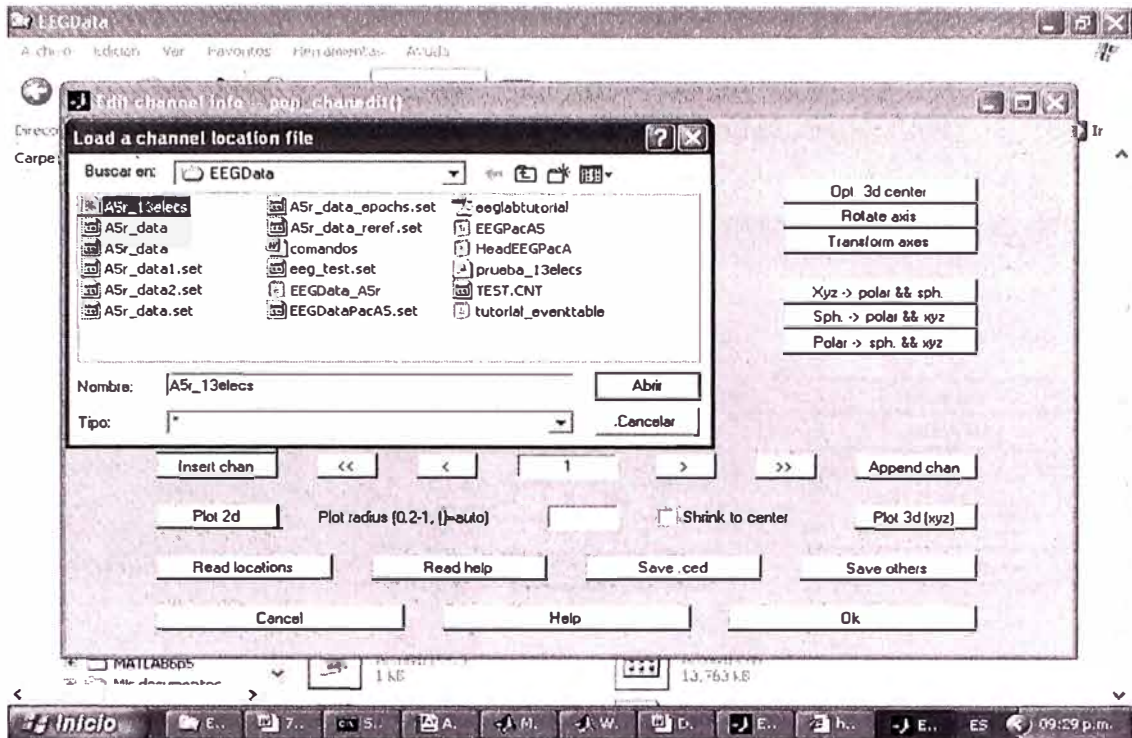


Fig. 3. 42 Ventana de elección del archivo de localización de los electrodos

Después de haber elegido el archivo correspondiente a los electrodos, obtenemos la estructura que servirá para los análisis posteriores, detalles de estos datos se pueden observar en la Fig. 3.43.

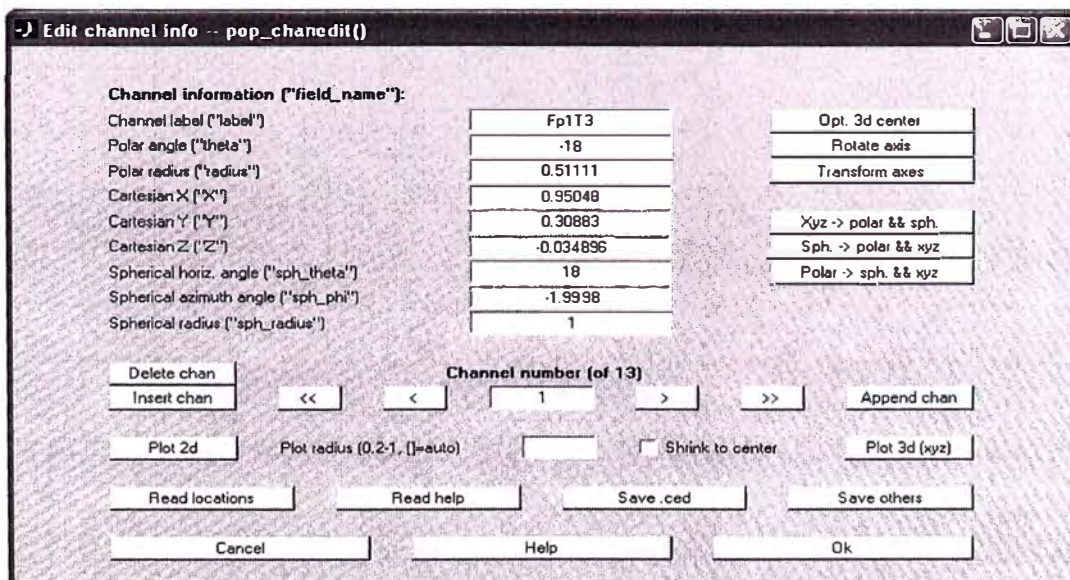


Fig. 3. 43 Detalle de las coordenadas del electrodo 1 (Fp1T3)

Como se puede observar en las figuras Fig. 3.44 y Fig. 3.45, los campos correspondientes a las coordenadas se muestran en todas las posibilidades que pueden procesarse, tales como cartesianas, polares y/o esféricas.

También es importante resaltar que se puede observar todos los canales de los electrodos utilizados (en este caso 13) con sus coordenadas respectivas.

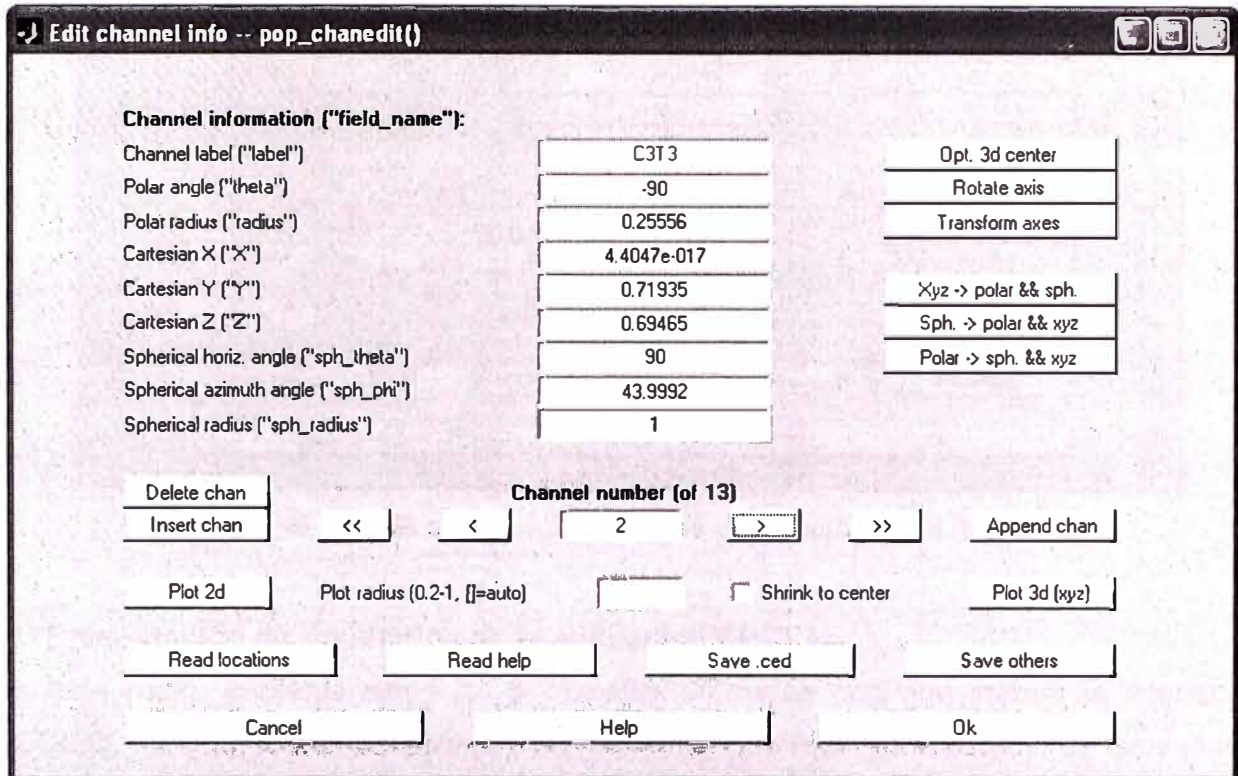


Fig. 3. 44 Detalle de las coordenadas del electrodo 2 (C3T3)

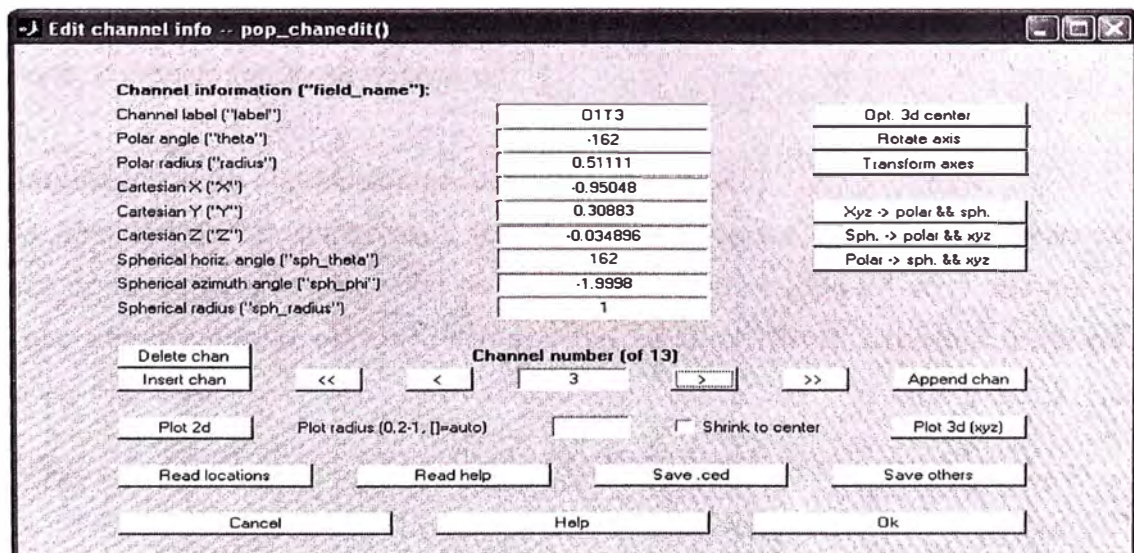


Fig. 3. 45 Detalle de las coordenadas del electrodo 3 (O1T3)

En el presente caso presentamos los datos de los canales 1, 2, 3 y 13, para que sirva como una referencia específica (Fig. 3.46).

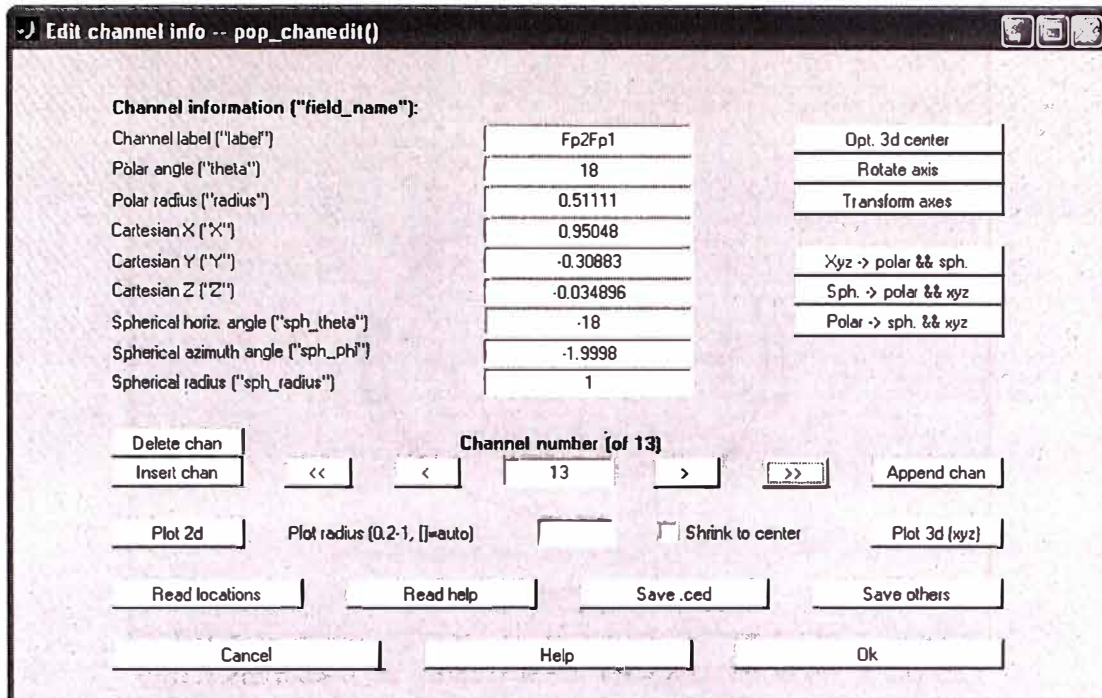


Fig. 3. 46 Detalle de las coordenadas del electrodo 13 (Fp2Fp1)

b) Presentación de resultados de la aplicación EEGLab

En esta parte, presentaremos las principales utilidades con que cuenta la aplicación EEGLab, mostrando sus bondades y herramientas que facilitan el estudio de las señales EEG. Estas bondades se caracterizan principalmente por que pueden ser utilizadas por personas expertas en temas de programación así como también en personas novatas, las cuales podrán ayudarse de las funciones con las que cuenta y poder realizar estudios de diferente índole, según su necesidad.

- **Visualización de electrodos en un mapa cerebral.**

Con la finalidad de familiarizarse y tener confianza de los datos que se han cargado, empezaremos con ver la ubicación de los electrodos utilizados en un mapa del cerebro, para ello utilizaremos la opción del menú del EEGLab **Plot> Channel location > By name** ver Fig. 3.47.

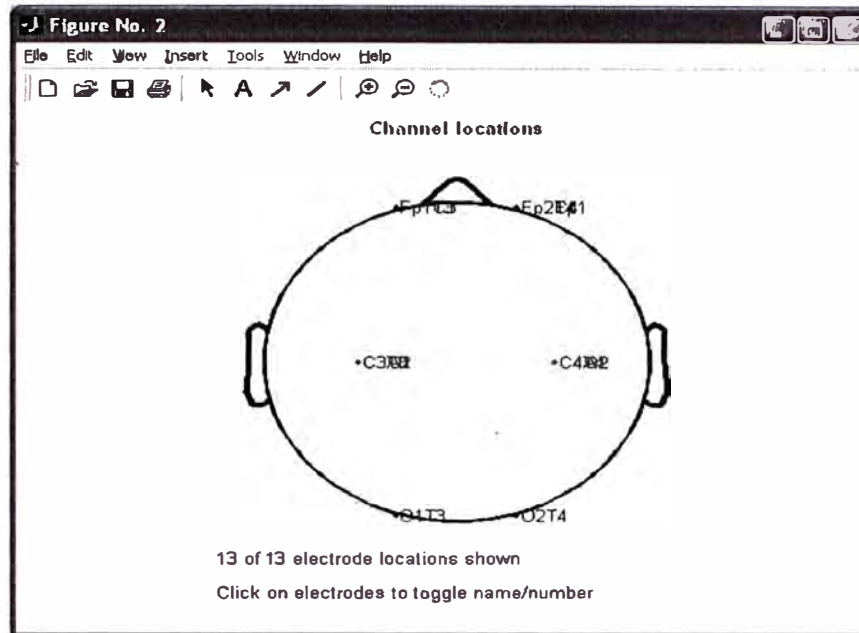


Fig. 3. 47 Mapa de ubicación de los electrodos en la cabeza

- **Visualización de las señales EEG.**

Las 13 señales del paciente "A5" podemos visualizarlas con la opción del menú del EEGLab Plot> Channel data (scroll) ver en las figuras Fig. 3.48 y Fig. 3.49.

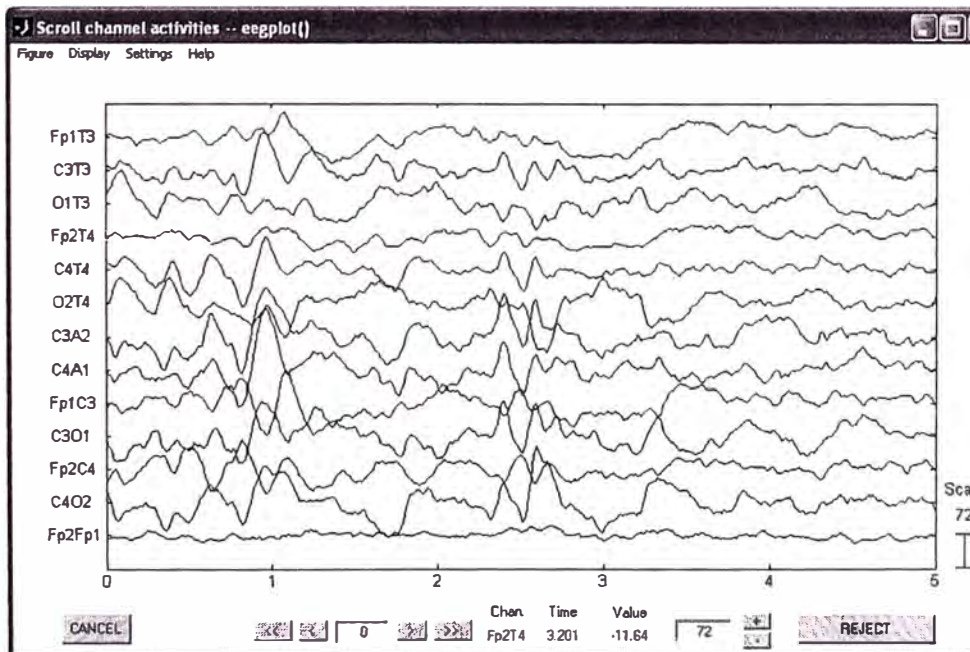


Fig. 3. 48 Gráfico de la primera trama de señales EEG

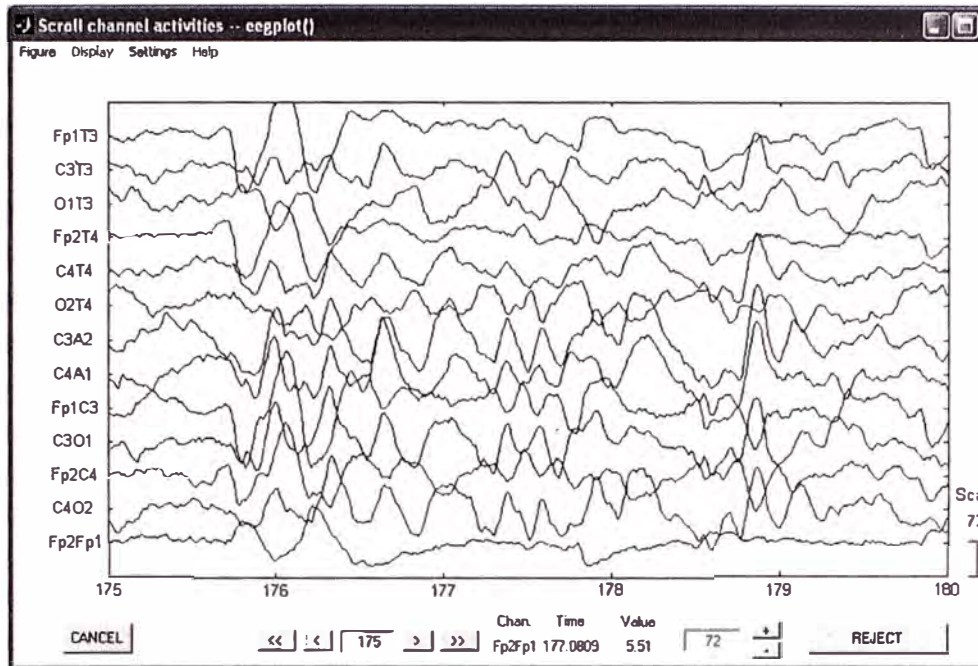


Fig. 3. 49 Gráfico de una trama intermedia de señales EEG

Desplazando la señal obtenida, se puede visualizar la señal en su totalidad, observando que para este caso hemos encontrado tramos de señales no válidas, señales que se encuentran distorsionadas por efecto de algún movimiento físico del paciente, tal como se puede apreciar en las figuras Fig. 3.50 y Fig. 3.51.

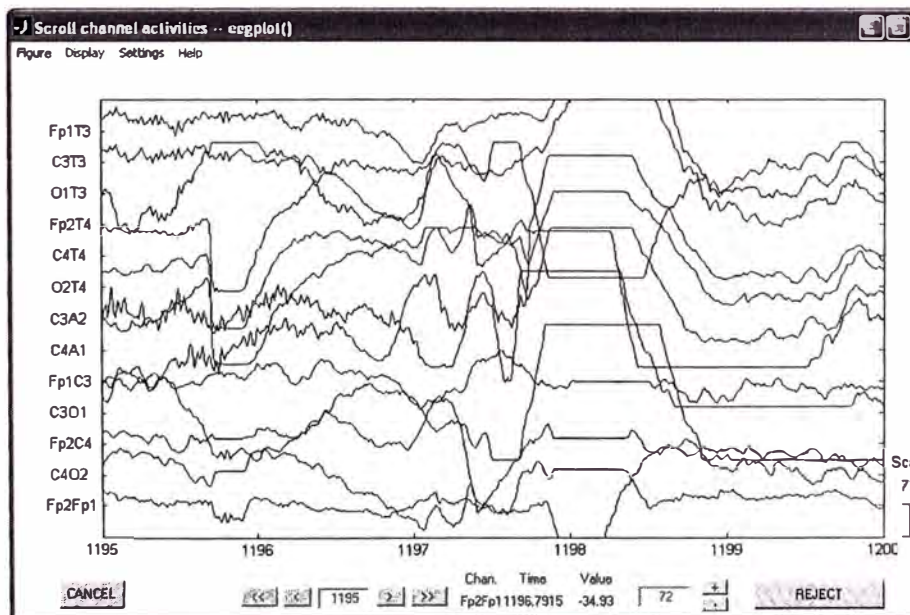


Fig. 3. 50 Señales no válidas encontradas en EEGPacA5

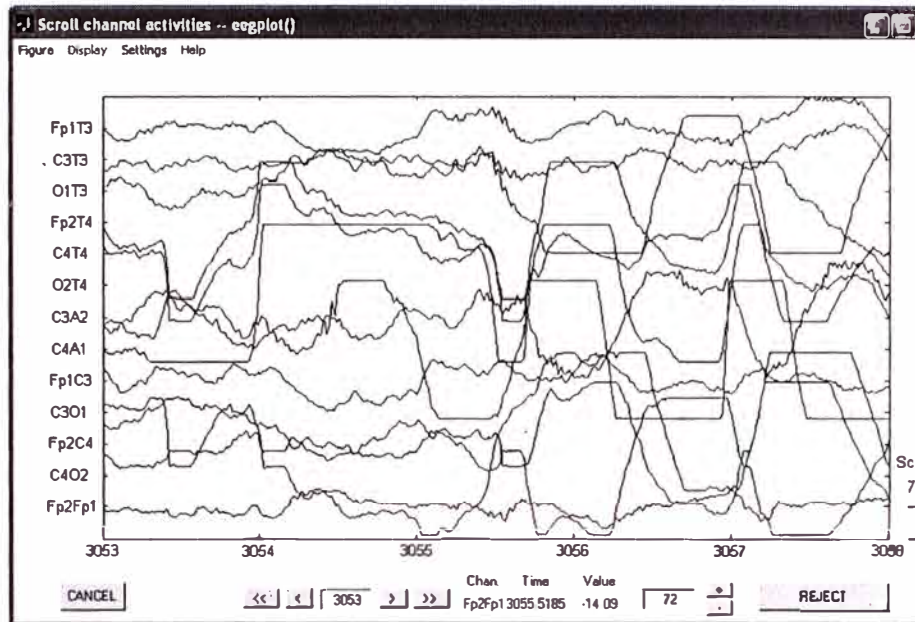


Fig. 3. 51 Señales no válidas encontradas en EEGPacA5

- **Rechazo de muestras inválidas**

Luego de realizar una inspección de las señales, hemos encontrado tramos de señales inválidas, y con la finalidad de realizar análisis con el EEGLab, es necesario que las señales estén limpias sin errores, motivo por el cual el EEGLab cuenta con una diversidad de funciones que permiten eliminar estos tramos, una de las formas es efectuarlas manualmente con la opción del menú **Plot> Channel data (scroll)**, en la cual se puede seleccionar estos tramos con defectos y marcarlos, para luego con la opción "REJECT" eliminarlos, tal como se muestra en la Fig. 3.52.

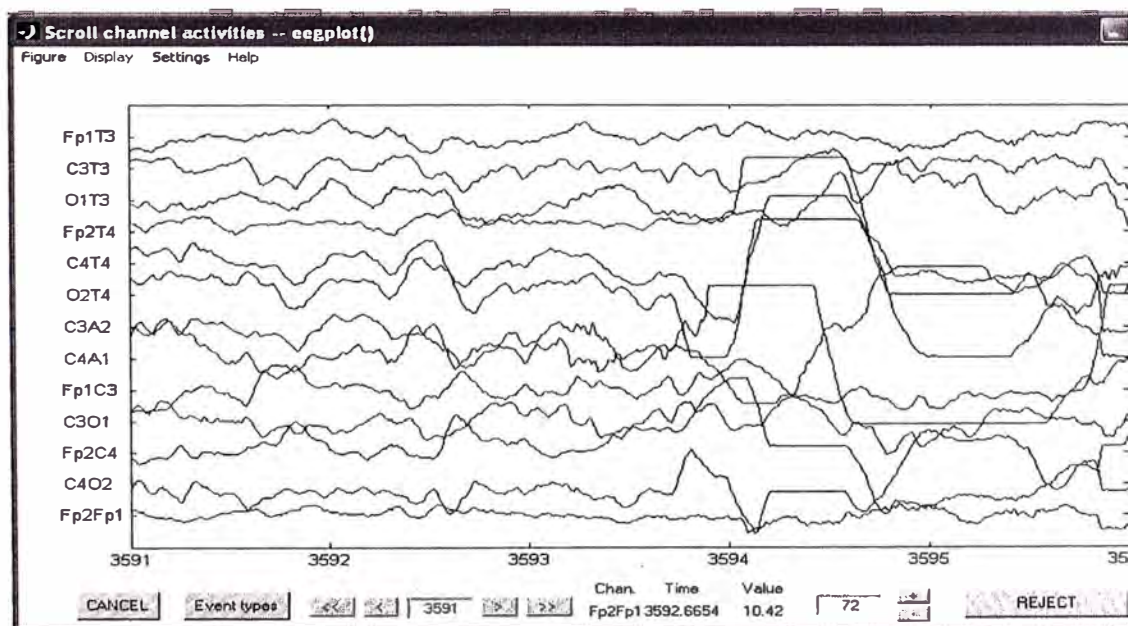


Fig. 3. 52 Selección de datos inválidos para su rechazo

- **Graficando el espectro de las señales de cada canal y su representación como mapas en el cerebro**

Para iniciar el procesamiento de datos, se recomienda primero visualizar todo los datos adquiridos en la opción Plot > Channel data (scroll) como anteriormente ya mostramos (obviamente desechando las porciones o “*epochs*” de datos inválidos), luego estudiando el espectro de potencia estar seguros que los datos cargados son cualificados para análisis posteriores.

Para graficar el espectro de un canal y asociado a un mapa topográfico, seleccionar **Plot > Channel spectra and maps**. Esto desplegará la ventana `pop_spectopo()` mostrado en el siguiente grafico (Fig. 3.53). Dejar los valores de los campos que aparecen por defecto y presione “OK”.

Field	Value
Epoch time range to analyze [min_ms max_ms]:	-1000 1990
Percent data to sample (1 to 100):	15
Frequencies to plot as scalp maps (Hz):	6 10 22
Apply to EEG ERP BOTH:	EEG
Plotting frequency range [lo_Hz hi_Hz]:	2 25
Spectral and scalp map options (see topoplot):	'electrodes','off'

Buttons: Cancel, Help, Ok

Fig. 3. 53 Formulario para graficar el espectro de un canal con data al 15%

La función deberá retornar un gráfico `spectopo()`, tal como se muestra en la siguiente figura (Fig. 3.54 y Fig. 3.55). Notar que la grafica se mostrará con mayor resolución de acuerdo a los datos de la prueba si en el campo “Percent data to simple (1 to 100)” se pone el valor de 100% (por defecto está en 15%).

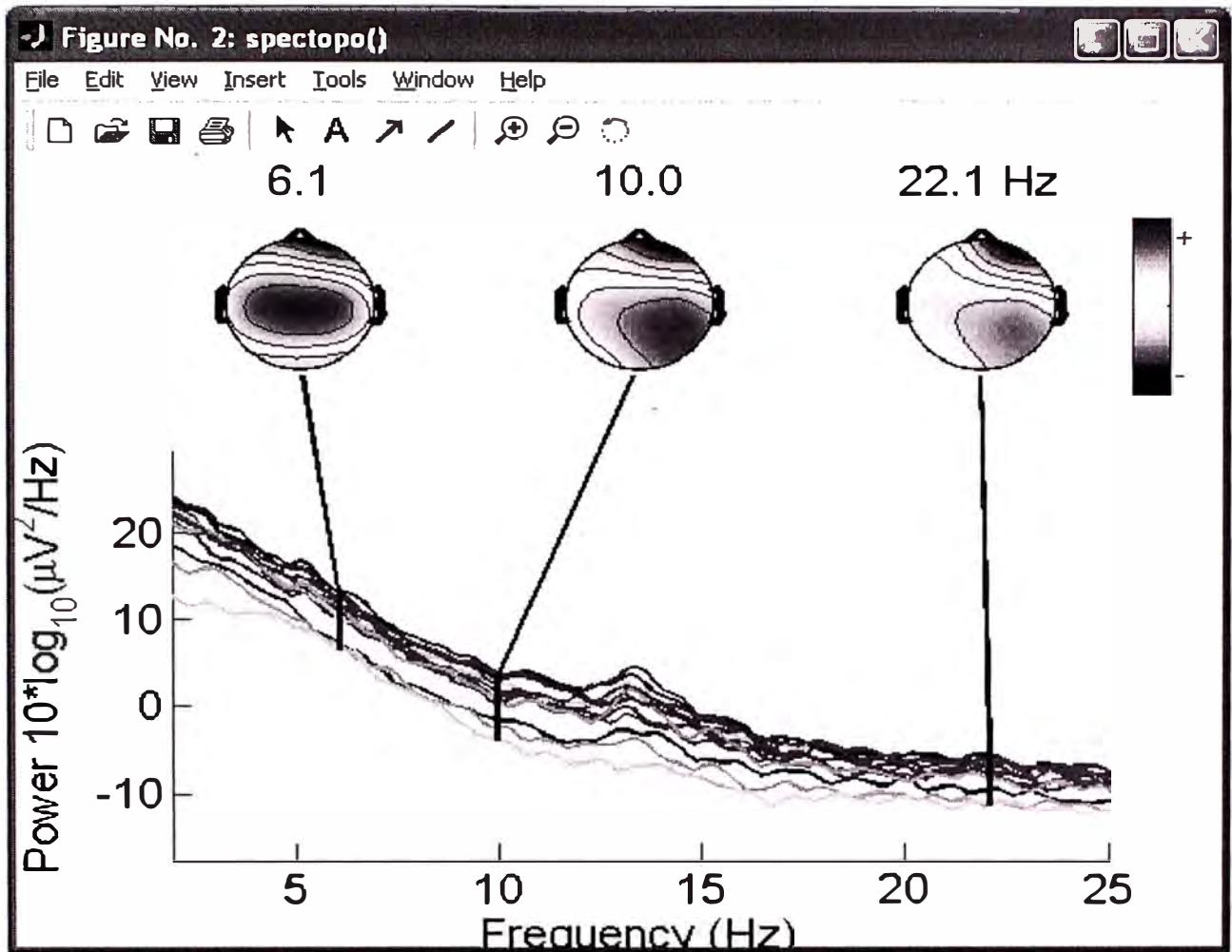


Fig. 3. 54 Gráfico pop_spectopo() al 15%

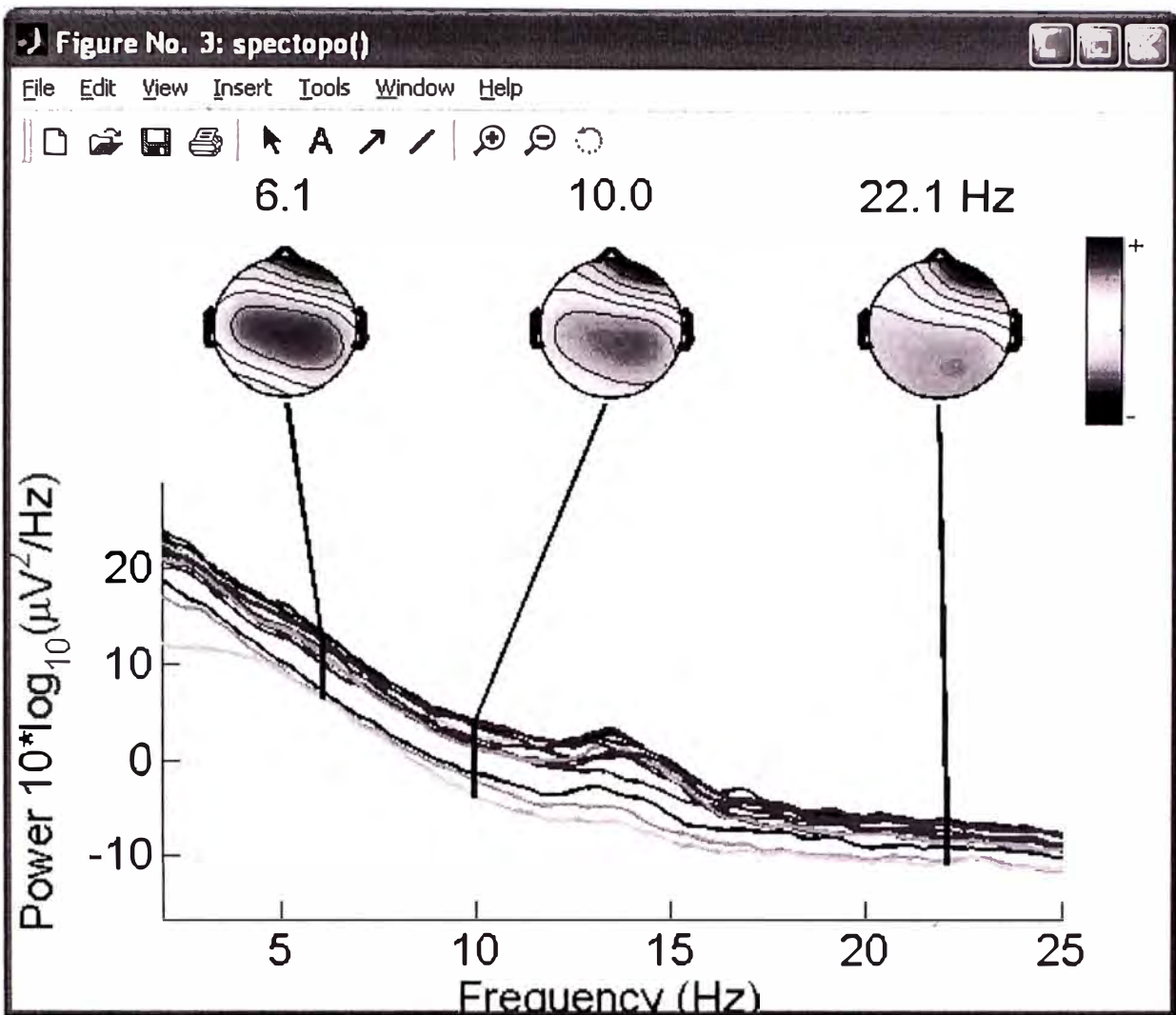


Fig. 3. 55 Gráfico pop_spectopo() al 100%

Cada trazo coloreado representa el espectro de un electrodo o canal de datos. El mapa del área de la cabeza del extremo izquierdo muestra la distribución en esta área, la potencia de 6 Hz que está concentrado en estos datos. Los otros mapas indican la potencia en 10 Hz y 22 Hz.

El menú de la ventana pop_spectopo() permite al usuario calcular y graficar el espectro en ventanas de tiempos (rangos) específicos (Fig. 3.56).

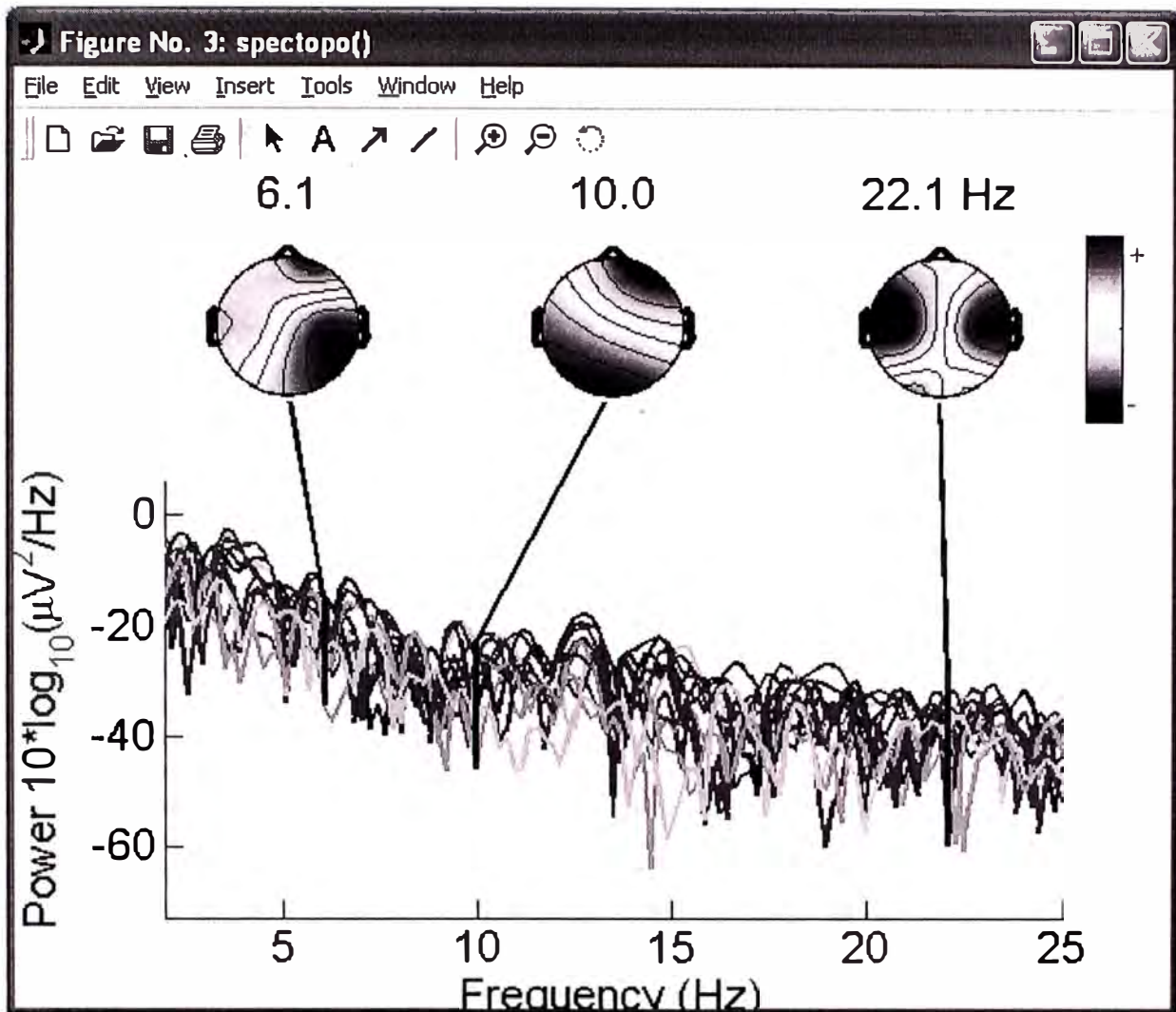


Fig. 3.56 Gráfico pop_spectopo() con la opción ERP

- **Promediando datos ERP (Data averaging)**

En estudios recientes se ha discutido que los Eventos Relacionados EEG dinámicos que ocurren en un conjunto de periodos de datos (*epochs*) no deberán ser confundidos con las características de su promedio visto en el tiempo, los Potenciales Relacionados de Eventos (ERP).

EEGLAB tiene varias funciones para ploteo 1-D de promedios de ERP de dataset. Actualmente se prefiere expandir el estudio de ERPs 1-D (potenciales a través del tiempo) de un conjunto de periodos de Eventos Relacionados EEG (*epochs*) por, primero, estudiando transformadas de ERP-image 2-D (potencial en tiempos por periodo) de datos agrupados en periodos (*epoched*), en los cuales los periodos de potencial de los datos son primeramente ordenados a lo largo de algunas dimensiones relevantes (por ejemplo, tiempo de reacción de un sujeto, potencia del ángulo theta, amplitud, fase alpha, el comienzo de los estímulos, etc.), luego (opcionalmente) suavizado con filtros y finalmente codificado con colores y visualizado como una imagen 2-D rectangular.

- **Graficando datos ERP en un eje simple con mapas de la cabeza**

Usaremos el conjunto de datos el cual no fue re-referenciado, pero fue filtrado, fraccionado en periodos (*epoched*) y con la “baseline” removido. Graficar el promedio (ERP) de toda la base de datos con mapas del cerebro seleccionados, seleccionando con la opción **Plot > Channel ERP > With scalp maps**. Obtendremos un grafico simple manteniendo los valores que se visualizan por defecto en la ventana `pop_timtopo()`, ingresando solo el título y presionando **OK** (Fig. 3.57).

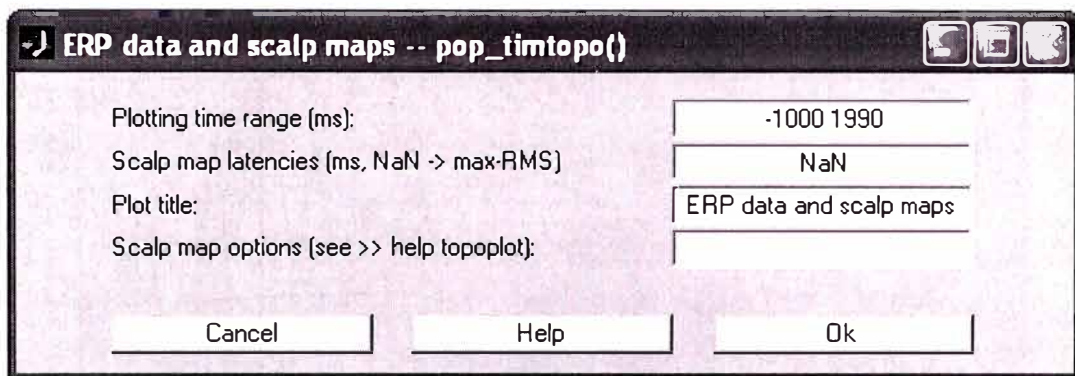


Fig. 3. 57 Formulario de gráfico `pop_timtopo()` – Promedio ERP

Una figura `timtopo()` aparece luego de presionar “OK”, cada traza grafica el promedio ERP de un canal. El mapa de la cabeza (Fig. 3.58) muestra la distribución topográfica del potencial promedio a una latencia de -420 para nuestro caso (la latencia con variación máxima en la ERP). Una o más latencias pueden ser graficadas en los mapas de la cabeza el cual puede ser especificado en la ventana anterior.

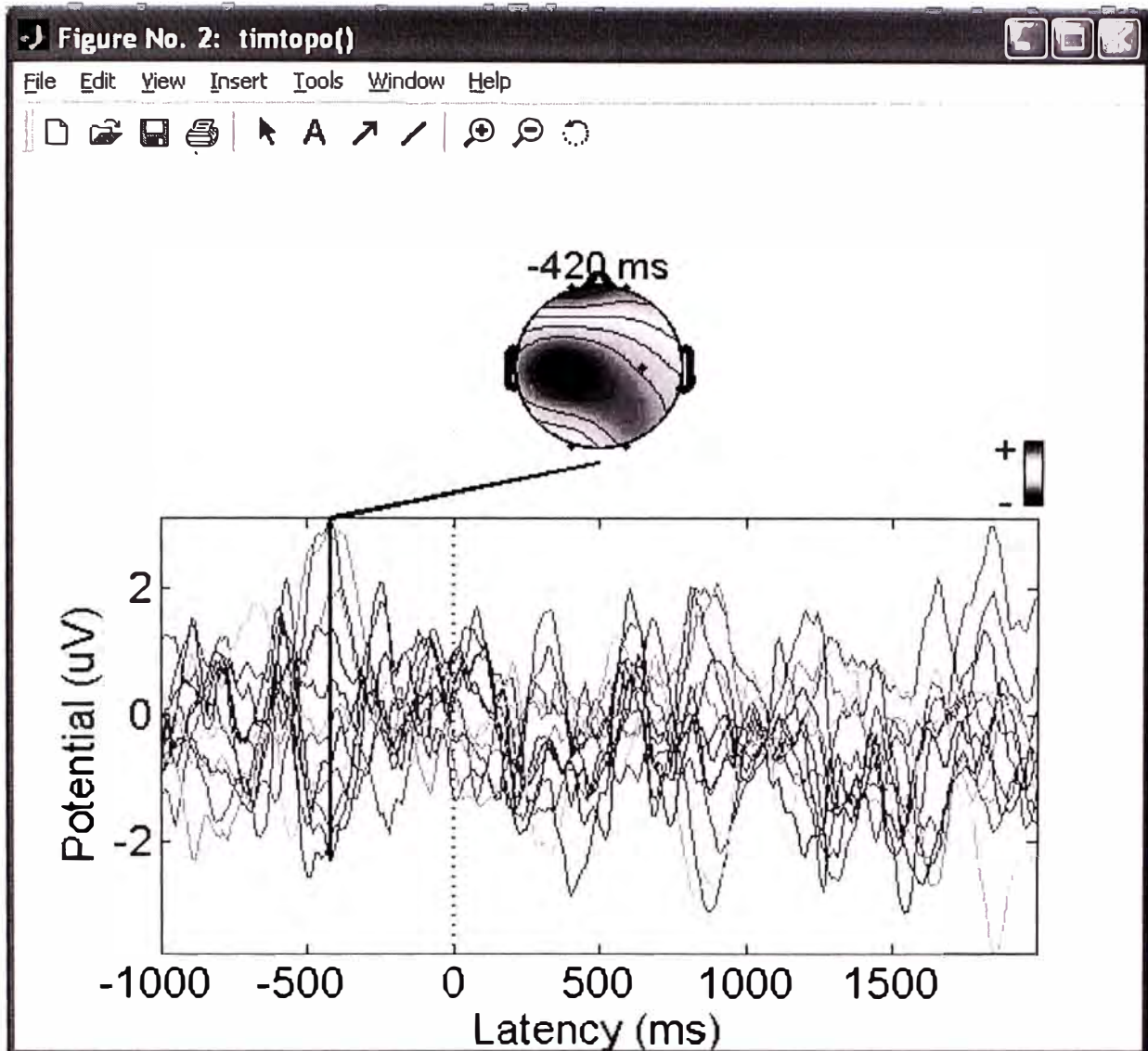


Fig. 3. 58 Gráfica del promedio ERP de un canal

Las gráficas de la función `timtopo()` muestra el curso de los tiempos relativos del promedio ERP en todos los canales, mas una muestra instantánea de los cambios de la distribución del potencial durante el periodo del ERP. Visualizar todos los scalp-map instantáneos como una "Película ERP" (ejemplo en una forma ERP nativa), llamando la función `eegmovie()` desde la línea de comandos.

- **Graficando trazas ERP en un arreglo topográfico**

Graficar el promedio de los ERP de un conjunto de datos (Fig. 3.59) formados en periodos (*epoched*) como trazas de un canal-simple en un arreglo topográfico, seleccionar `Plot > Channel ERPs > In scalp array`. Utilizando los valores por defecto, como se muestra en la siguiente imagen, adicionando un título, y presionando "OK" en la ventana `pop_plottopo()`.

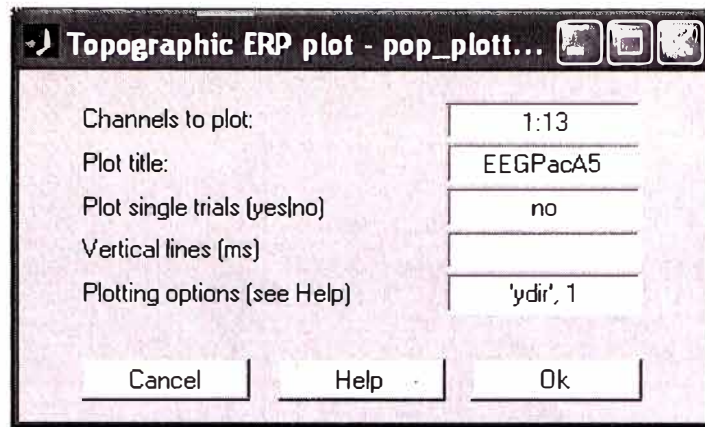


Fig. 3. 59 Formulario de gráfica de promedio ERP de un conjunto de datos

Produce la siguiente figura (Fig. 3.60) de la función `plottopo()`.

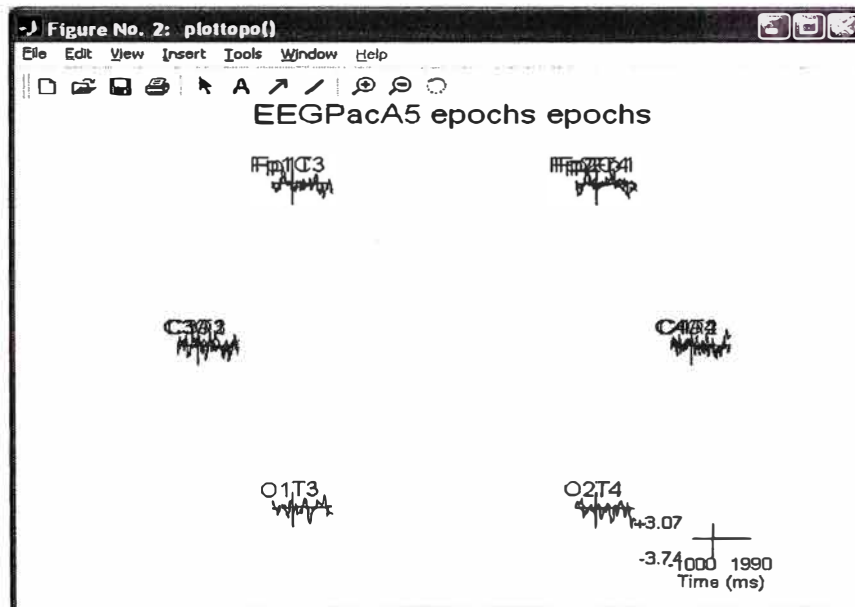


Fig. 3. 60 Gráfica de promedio ERP de un conjunto de datos

Se puede visualizar un canal específico pulsando encima de su traza produciendo un grafico tal como se muestra en la ventana de más abajo (Fig. 3.61).

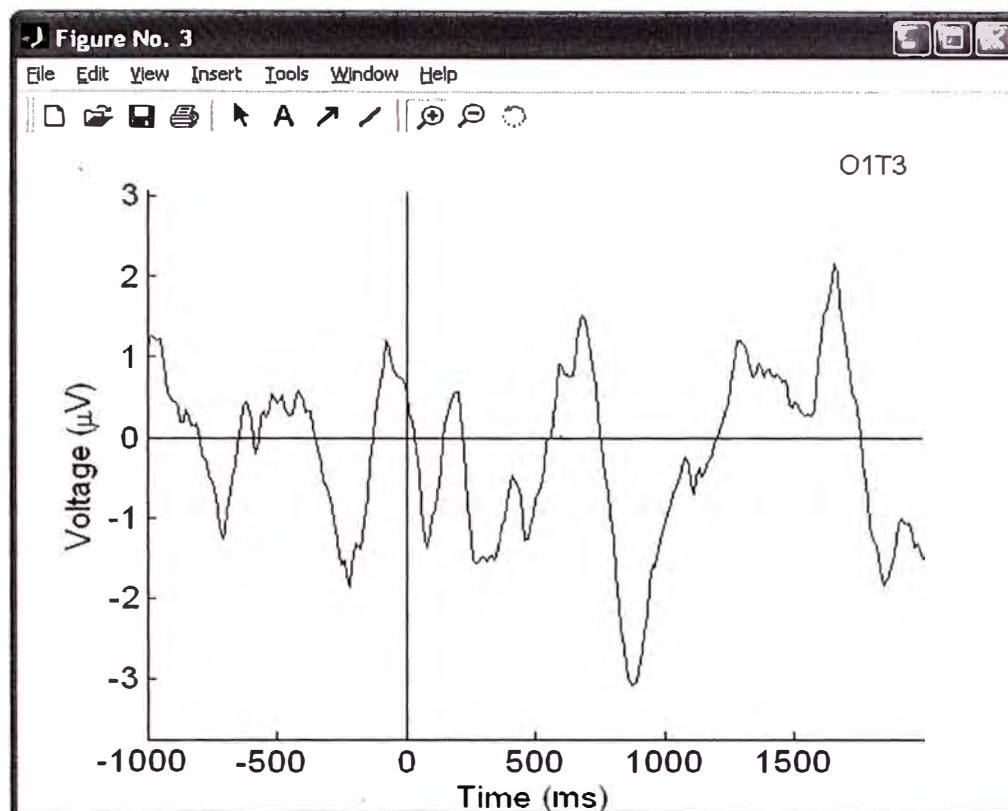


Fig. 3. 61 Visualización de ERP de un canal específico

- **Graficando ERPs en un arreglo de dos columnas**

Para graficar uno o más promedios de trazas de datos ERP en arreglos de dos columnas (Fig. 3.62), ir al menú Plot >Channel ERPs > In rect. array. Utilice las opciones por defecto que aparecen en la ventana pop_plotdata() y presione "OK".

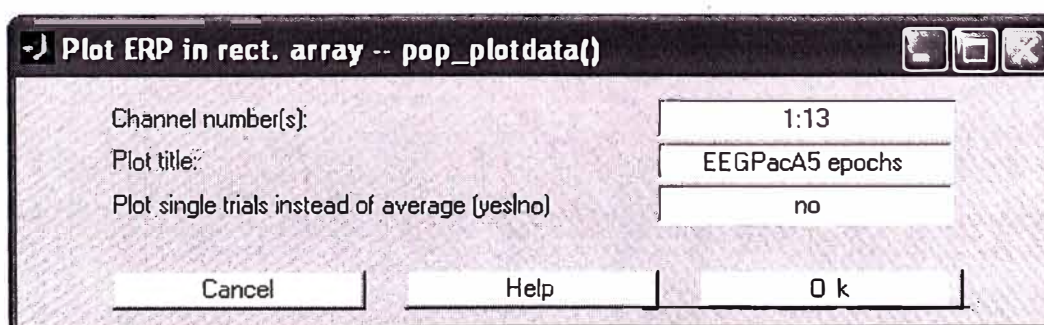


Fig. 3. 62 Formulario de gráfica ERPs en arreglo de 2 columnas

La figura resultante de la función plotdata() se muestra a continuación (Fig. 3.63):

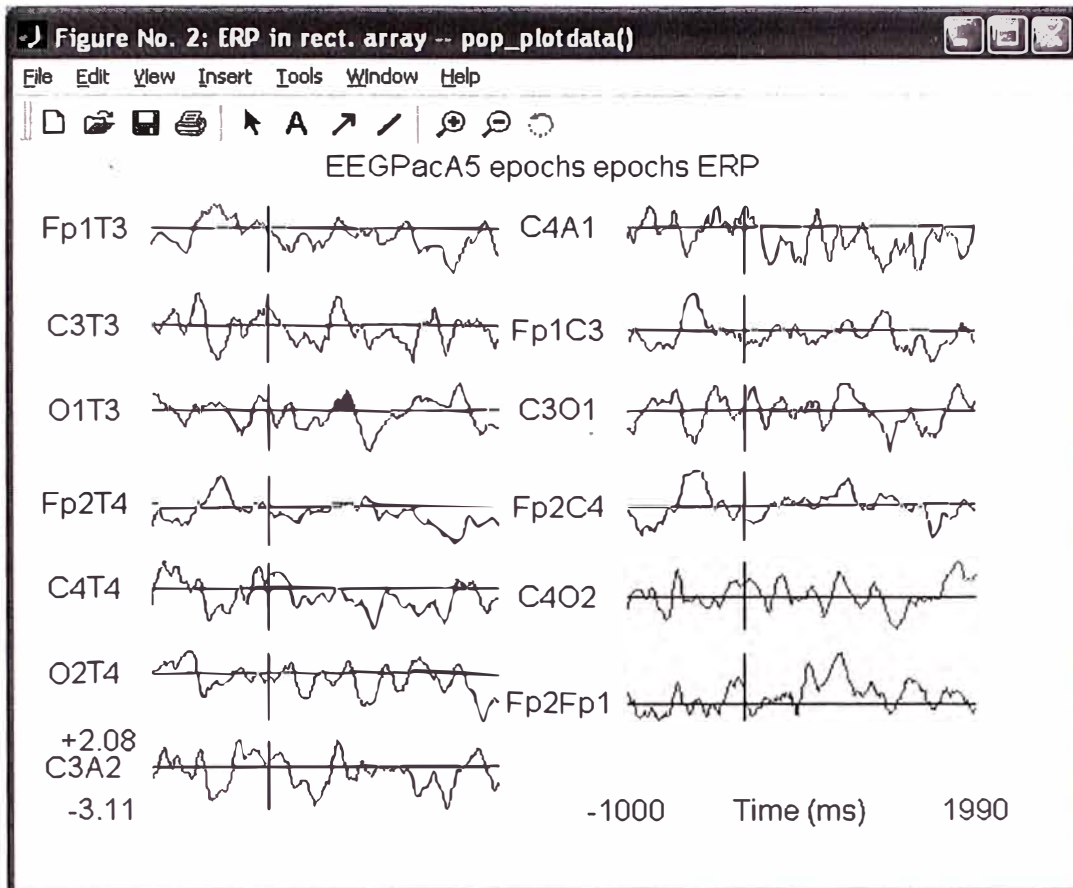


Fig. 3. 63 Gráfica de ERPs en arreglo de 2 columnas

También al hacer click en cualquiera de las trazas de la ventana anterior podemos obtener el gráfico individual, con lo cual podemos observar con más detalle la gráfica (Fig.3.64).

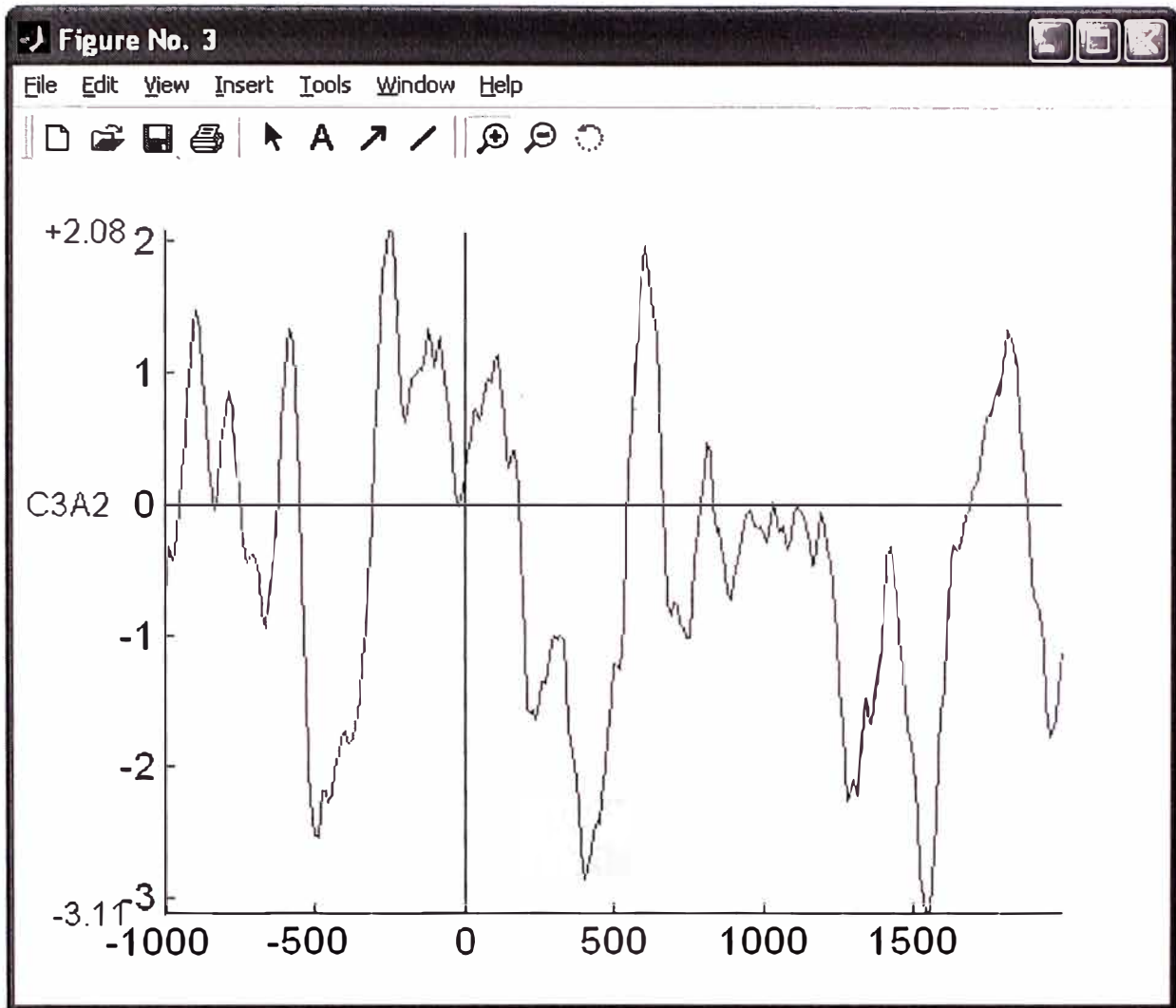


Fig. 3. 64 Gráfica de ERP de un canal específico

- **Graficando un ERP como una serie de mapas.**

Para graficar datos ERP en una serie de mapas ERP (Fig. 3.65 y Fig. 3.66) en 2-D que representan la distribución del potencial en una serie de tiempos durante periodos, seleccionar **Plot > ERP map series > In 2-D**. En la opción de la parte superior de la ventana `pop_topoplot()`, poner la latencia deseada para el mapa de los ERP del cerebro. Nota: En esta opción se puede ingresar cualquier expresión numérica de Matlab, por ejemplo, en vez de poner "0 100 200 300 400 500", intente con "0:100:500". Expresiones tales como "-6000+3*(0:20:120)" son también aceptadas.

ERP scalp map(s) -- pop_topoplot()

Plotting ERP scalp maps at these latencies
(range: -1000 to 1990 ms, NaN -> empty):

Plot title: EEGPacA5 epochs epochs

Plot geometry (rows,col.): [] -> near square

-> Additional scalp map plotting options (see help):
'electrodes', 'on'

Cancel Help Ok

Fig. 3. 65 Formulario para gráficar series de ERP

ERP scalp map(s) -- pop_topoplot()

Plotting ERP scalp maps at these latencies
(range: -1000 to 1990 ms, NaN -> empty): 500:10:1990

Plot title: EEGPacA5 epochs epochs

Plot geometry (rows,col.): [] -> near square

-> Additional scalp map plotting options (see help):
'electrodes', 'on'

Cancel Help Ok

Fig. 3. 66 Formulario de series ERP con latencias entre 500 ms y 1990 ms

La ventana topoplot() muestra mapas del cerebro conteniendo ERP en las latencias especificadas (Fig. 3.67, Fig. 3.68 y Fig. 3.69). En estas graficas se puede seleccionar la forma como se presenta las gráficas en filas por columnas a elección por el usuario.

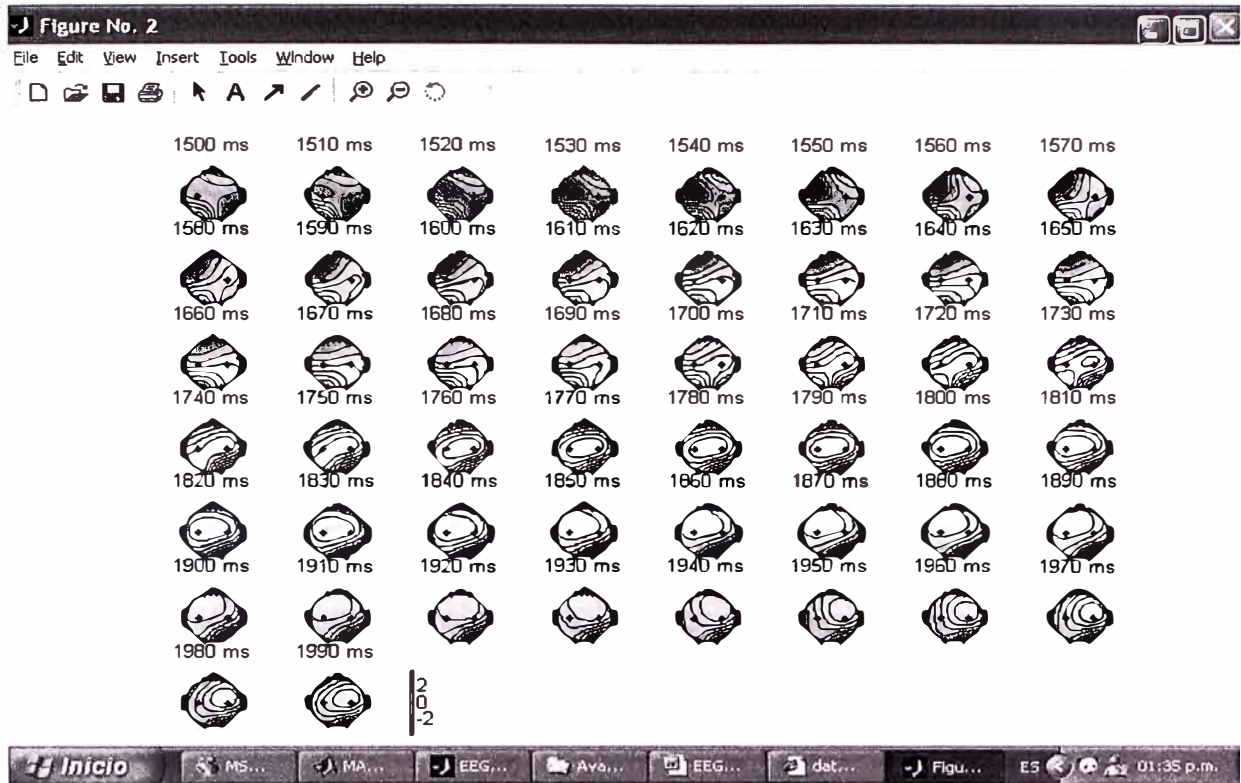


Fig. 3. 67 Gráfica de ERPs en series

Fig. 3. 68 Formulario de series ERP con latencias entre -1000 ms y 500 ms

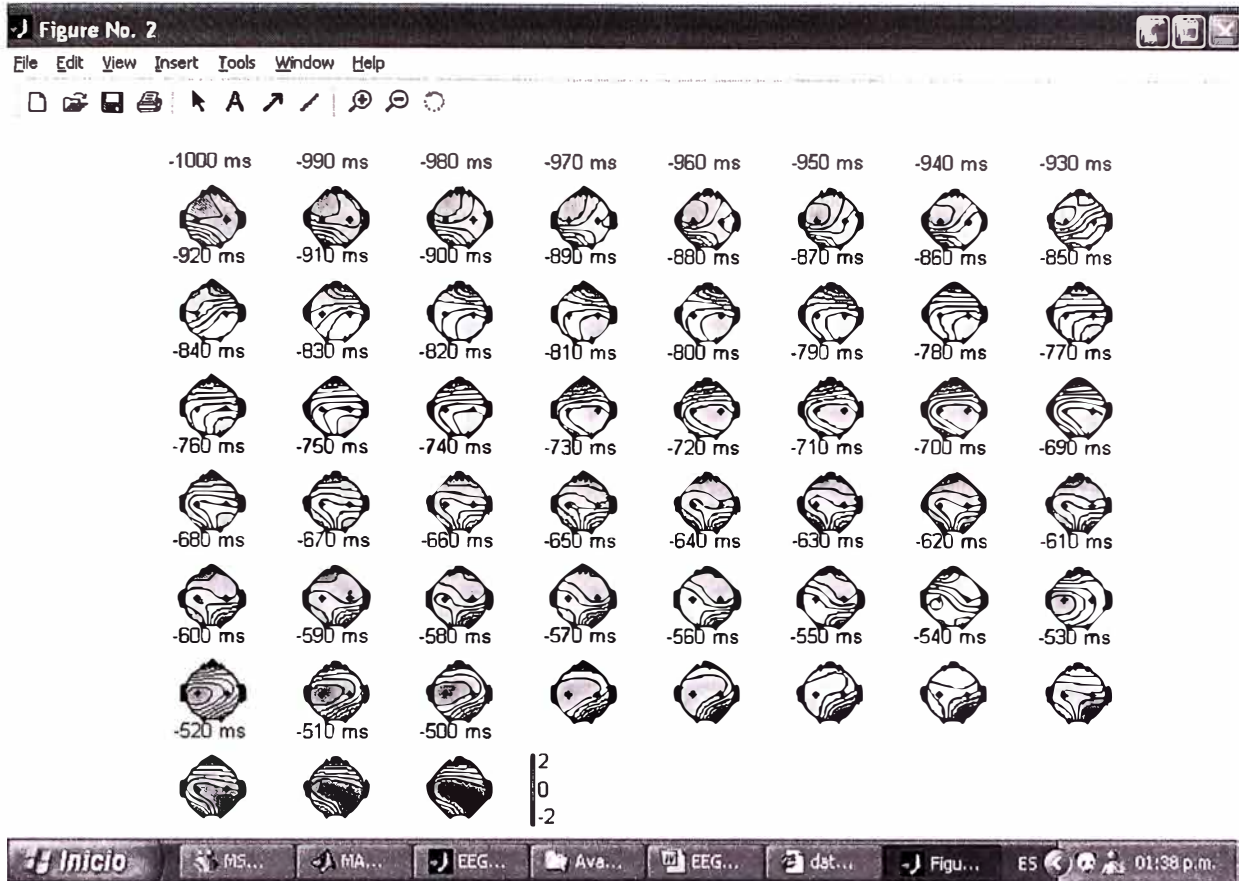


Fig. 3. 69 Gráfica de serie de ERPs entre -1000 ms y 500 ms

- **Graficando datos ERP como una serie de mapas 3-D.**

Para graficar datos ERP como una serie de mapas de la cabeza 3-D, ir al menú **Plot > ERP map series > In 3-D**. El requerimiento de este menú aparecerá preguntando si desea crear y guardar un nuevo archivo “**spline 3-D**”. Este proceso deberá ser hecho solamente una vez por cada conjunto de datos. Presionar “**YES**” para empezar este proceso (Fig. 3.70).

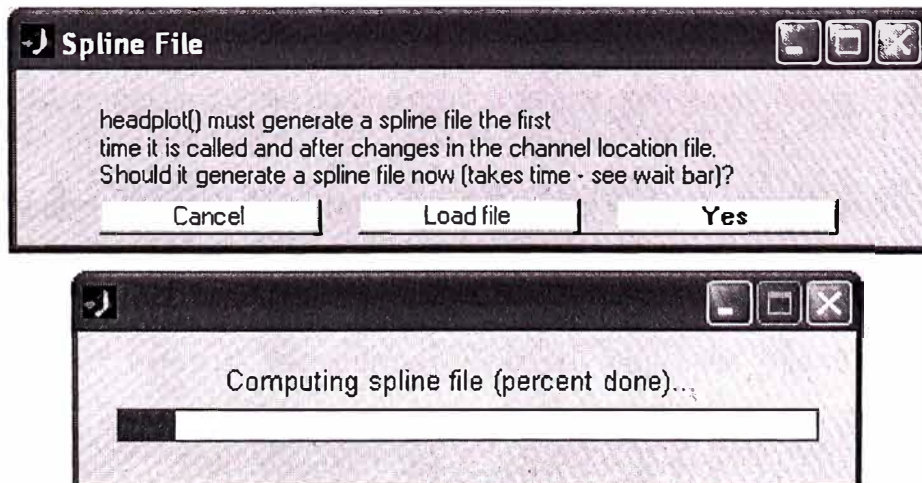


Fig. 3. 70 Reporte del proceso de generación del archivo Spline

Ahora la función `pop_headplot()`, función que grafica en 3-D (Fig. 3.71), creará el archivo "spline". Este proceso puede requerir varios minutos o más. Una barra de progreso se desplegará indicando cuando haya culminado la creación. Cuando el montaje del archivo spline ha sido generado, seleccionar **Plot > ERP map series > In 3-D** y ahora que el archivo spline ha sido ya generado, una ventana diferente e interactiva se desplegará. Tal como los mapas del cerebro 2-D, en la primera fila ponga la latencia deseada y presione **OK**.

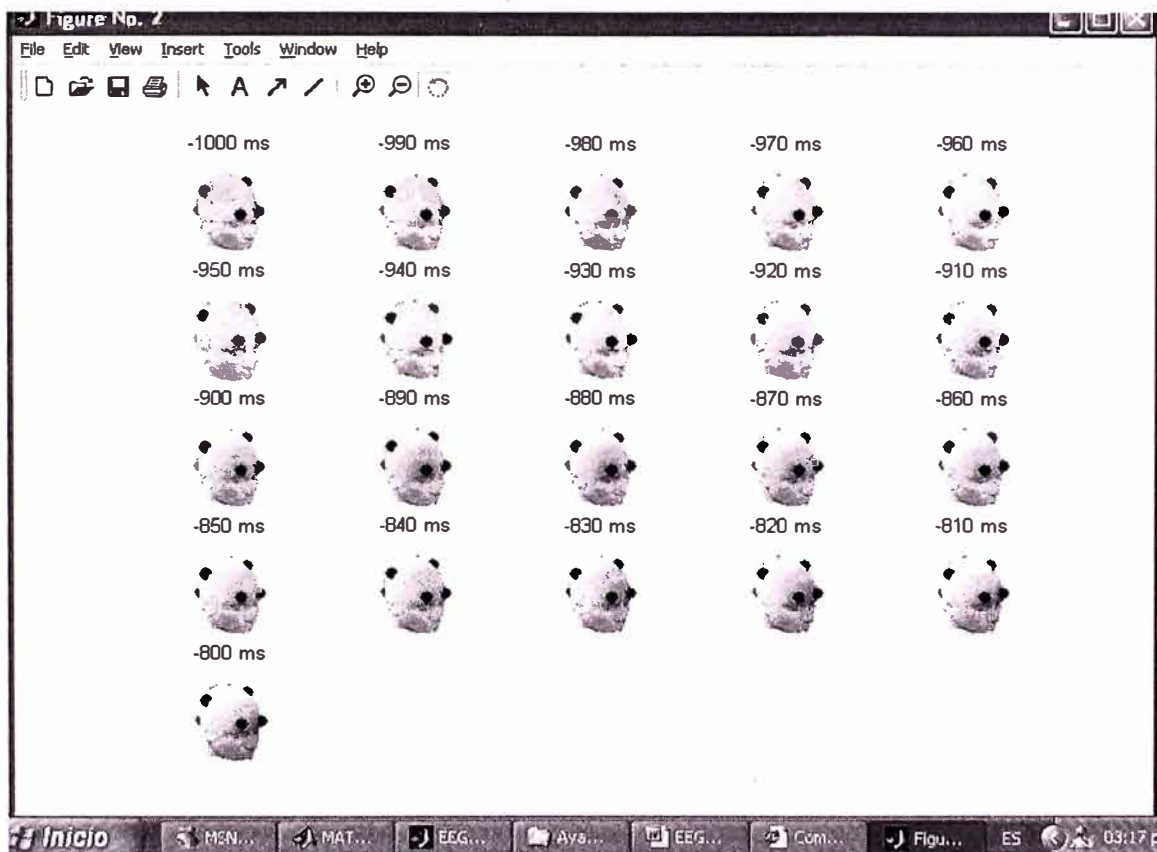


Fig. 3. 71 Gráfica de series de ERPs en 3D

Como es ya usual, haciendo clic en una grafica puede ser rotado usando el Mouse.

- **Seleccionando periodos de datos y graficando datos promedios**

Seleccionando eventos

Para seleccionar eventos específicos, usar el ítem del menú **Edit > Select epochs/events** (Fig. 3.72). Esto puede ser explotado para comparar diferentes tipos de periodos de datos "data epochs", seleccionando eventos y periodos para dos condiciones, los cuales describen unos procesos básicos de selección de eventos. También se puede especificar combinaciones más complejas de selección de campos de

eventos y rangos como criterios para selección de pruebas. Por ejemplo, la siguiente ventana podría seleccionar periodos conteniendo eventos de "Type" etapa1 (de acuerdo a nuestras pruebas), en el cual la latencia esta entre 0 ms y 400 ms, de entre los 40 primeros periodos de la dataset padre. Notar que la opción "Cancel": "Remove *epochs* not referenced by any selected event" (Extraer periodos no referenciados por cualquier evento seleccionado). Si esta opción fue dejada no activada y la opción "Keep only selected events and remove all other events" (Mantener solamente eventos seleccionados y extraer todos los otros eventos), la función podría simplemente seleccionar los eventos especificados pero podría no extraer periodos que no contienen estos eventos. Nota: Para seleccionar eventos fuera del rango dado, seleccionar la opción a la derecha del campo de entrada de rango "Select events NOT selected above" (Seleccionar eventos No seleccionados mas arriba). Es posible de renombrar el tipo de los eventos seleccionados (mientras (opcionalmente) manteniendo el nombre del tipo de evento antiguo como un nuevo campo) usando las dos últimas opciones de campos editables.

Fig. 3. 72 Formulario de selección de eventos y periodos

Ingrese el nuevo nombre del "Dataset" (Fig. 3.73 y Fig. 3.74).

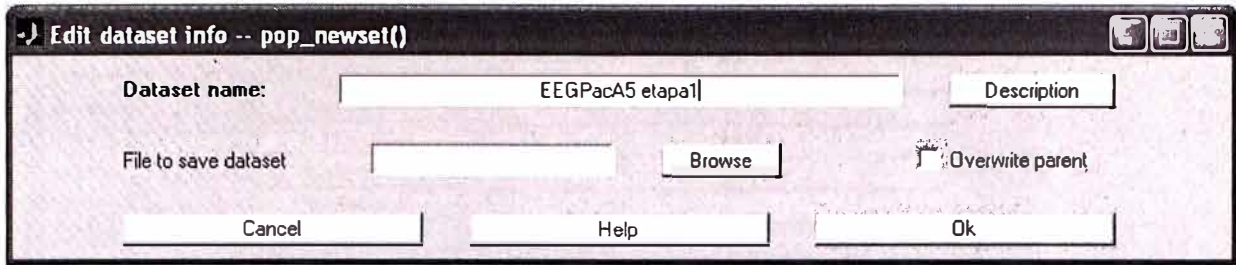


Fig. 3. 73 Formulario de definición del nombre del Dataset

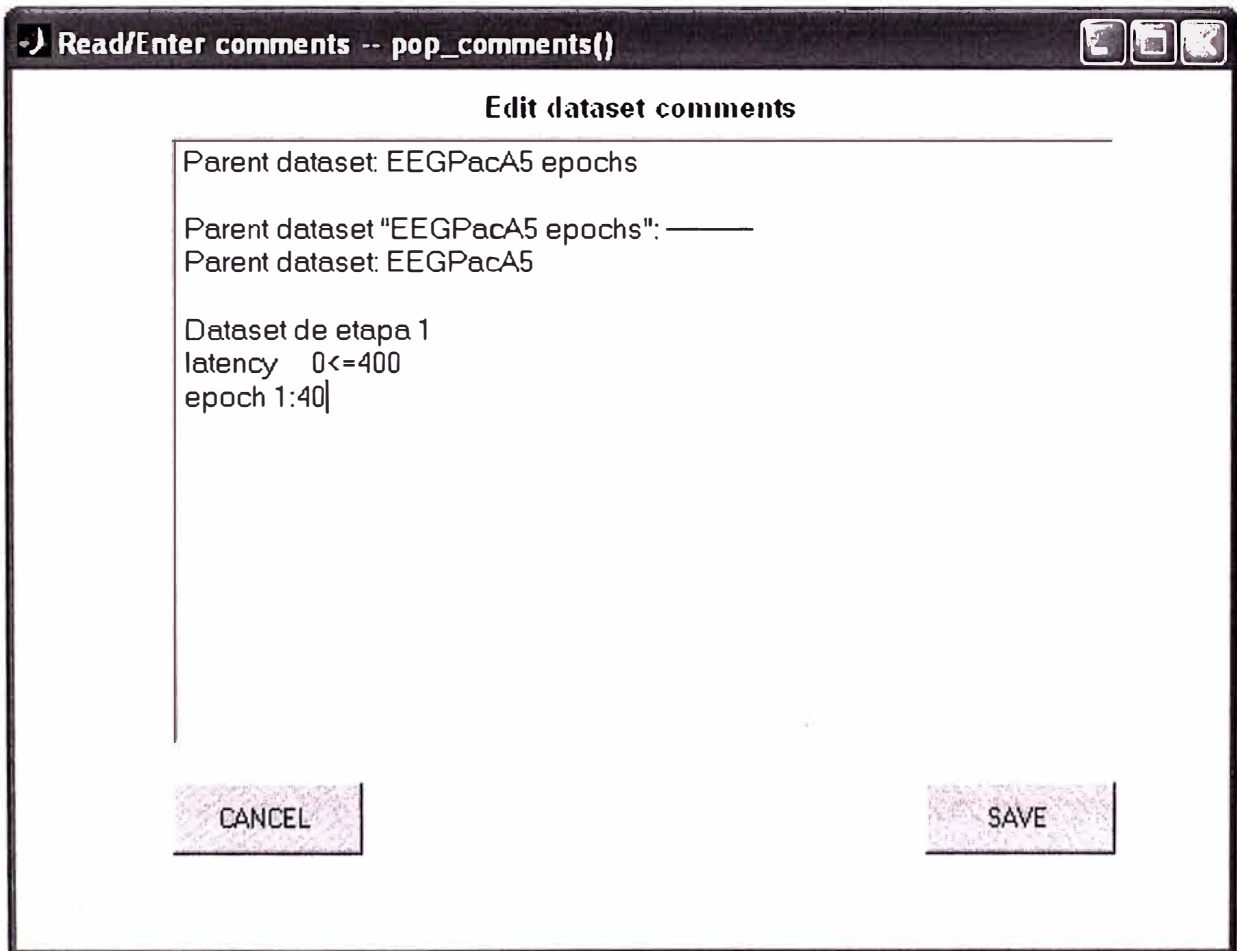


Fig. 3. 74 Formulario de comentario referencial

Seleccionando eventos y periodos por dos condiciones

Para comparar eventos relacionados EEG dinámicos para dos o más condiciones desde un mismo experimento, primero es necesario crear "datasets" conteniendo periodos "epochs" para cada condición. Seleccionar **Edit > Select events**. La ventana pop_selectevent() aparece. Seleccione todos los periodos (epochs) en el cual el objetivo apareció en el tipo "etapa1" de acuerdo a nuestro ejemplo de datos (Fig. 3.75).

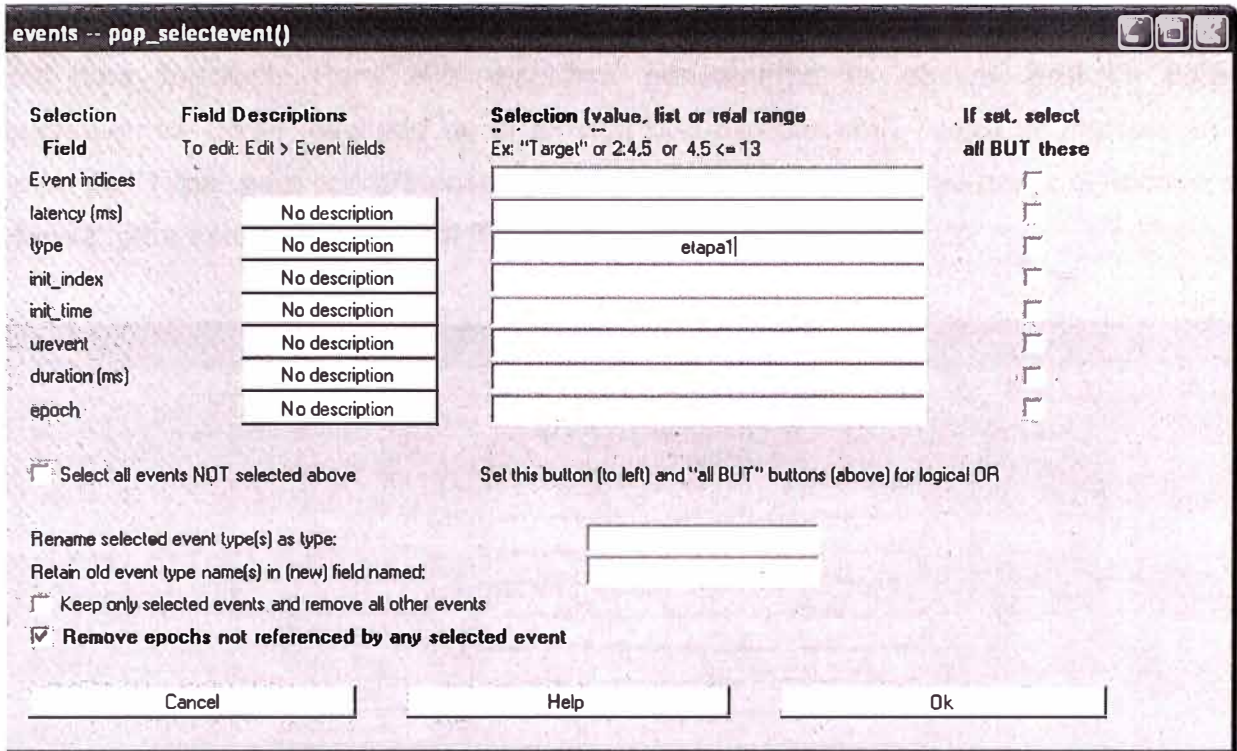


Fig. 3. 75 Formulario de selección del evento y periodo etapa1

Presionando “**Yes**” en la ventana siguiente (Fig. 3.76):

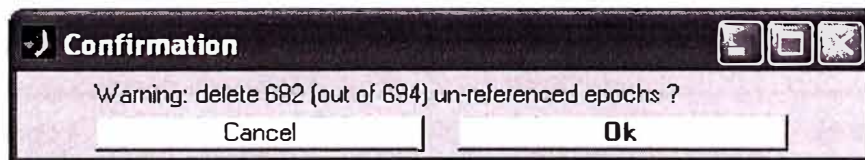


Fig. 3. 76 Confirmación de selección de eventos y periodos etapa1

Ahora en la ventana `pop_newset()` guardamos este nuevo “**dataset**” con el nombre de “**etapa1**” y presionamos “**OK**” (Fig. 3.77).

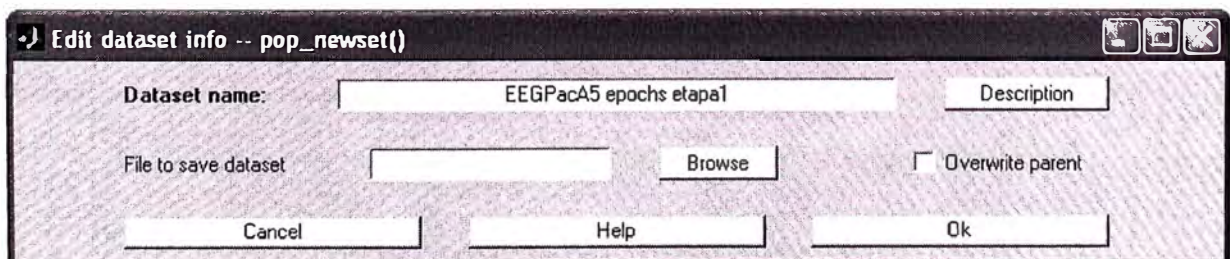


Fig. 3. 77 Grabando los eventos de etapa1 en un nuevo Dataset

Luego, repetimos el proceso de crear un segundo “**dataset**” consistiendo en periodos “**epochs**” con el objetivo de los datos de la etapa2. Primero, “recuperamos el dataset

original” en el menú de datasets. Seguidamente, seleccionamos el segundo “dataset” (“eeg_data_epochs”). Para ello elegimos nuevamente la opción **Edit > Select epoch/events**. En el resultado de la función `pop_selectevent()`, luego se ingresa en el campo de “Type” para nuestro caso “**etapa2**”. Presionar “OK”, luego poner el nombre de “**etapa2**” para este nuevo dataset (Fig. 3.78, Fig. 3.79 y Fig. 3.80).

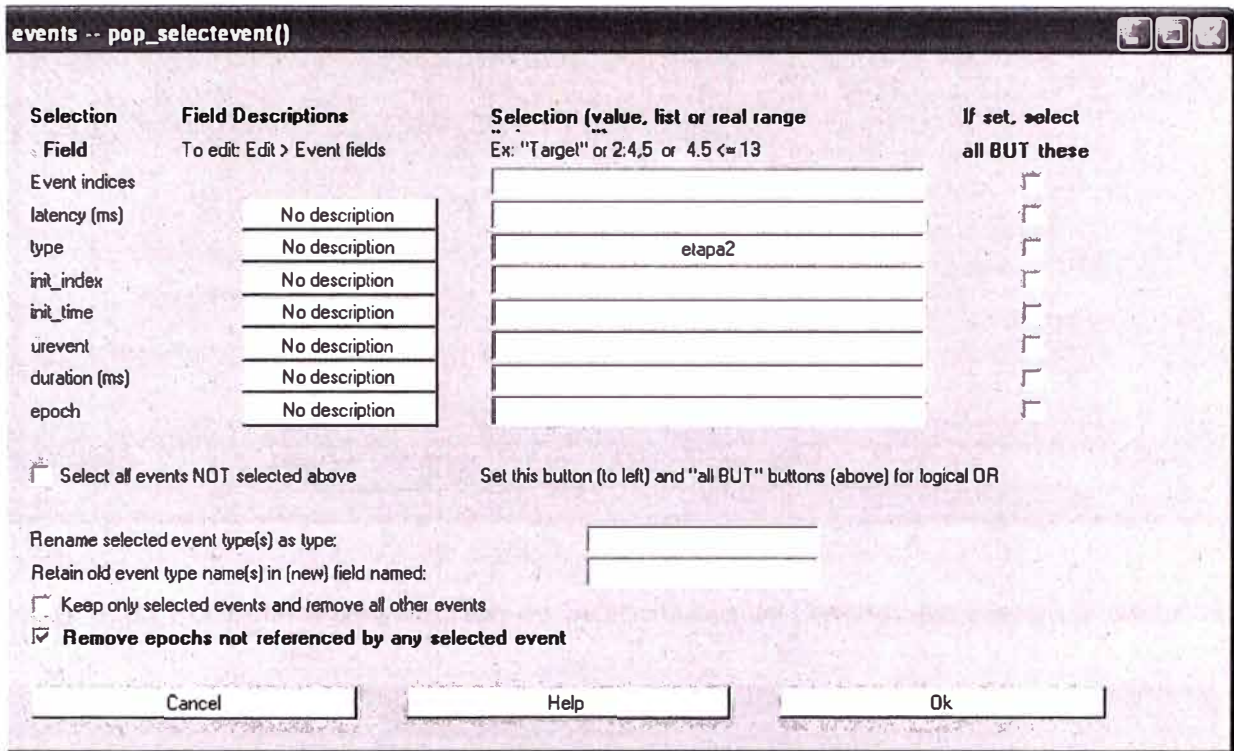


Fig. 3. 78 Formulario de selección del evento y periodo etapa2

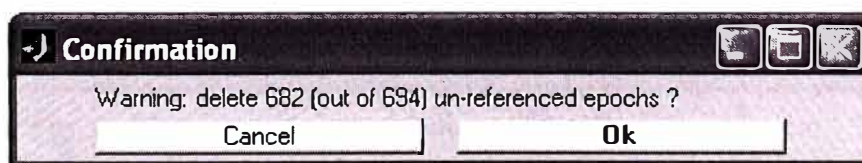


Fig. 3. 79 Confirmación de selección de eventos y periodos etapa2

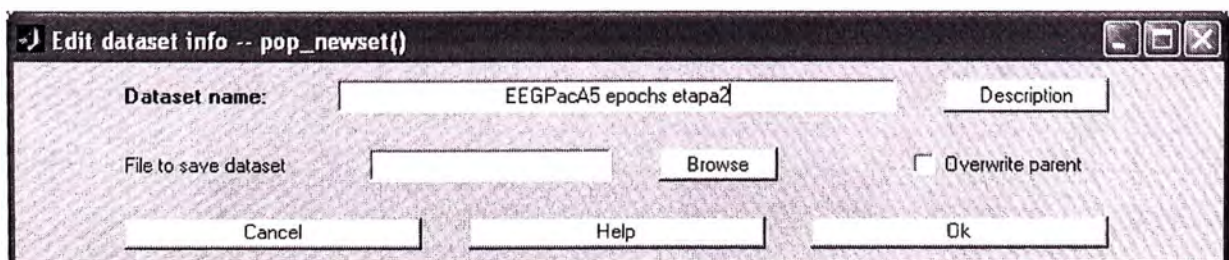


Fig. 3. 80 Grabando los eventos de etapa2 en un nuevo Dataset

Otra función que puede ser útil para seleccionar un subconjunto de un "dataset" es la función `pop_select()` llamado por la selección de la opción **Edit > Select data**. El ejemplo de la figura de abajo (Fig. 3.81, Fig. 3.82 y Fig. 3.83) podría seleccionar datos de sub periodos (sub-epochs) con el rango de tiempo de periodo de -500 ms a 1000 ms. Podría, mas aún, eliminar periodos 2, 3 y 4 del dataset y eliminar completamente el canal 13.

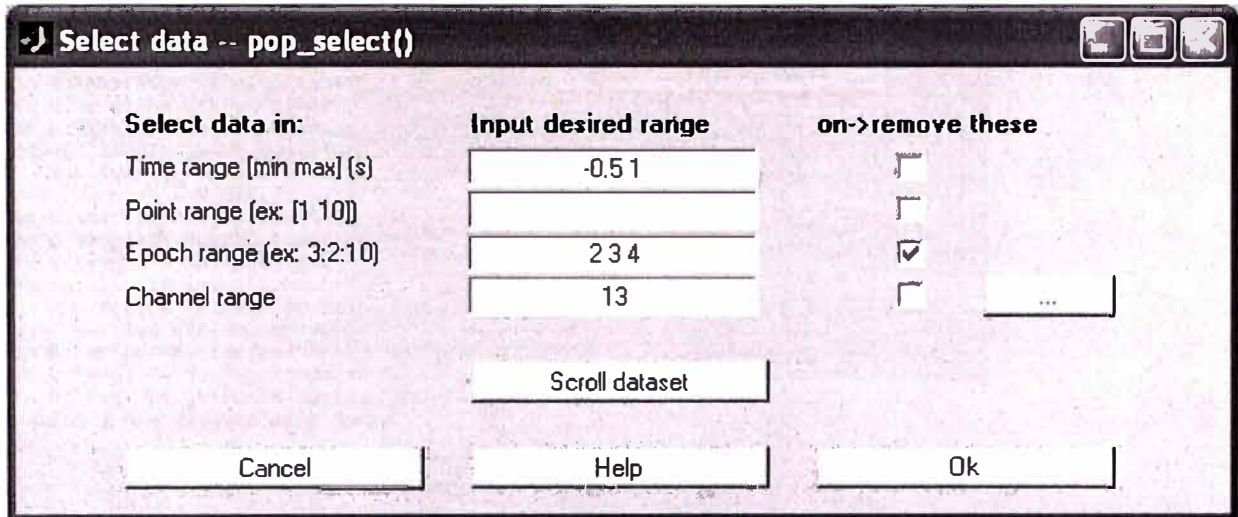


Fig. 3. 81 Formulario de selección de Subconjunto de Dataset eliminando el canal 13

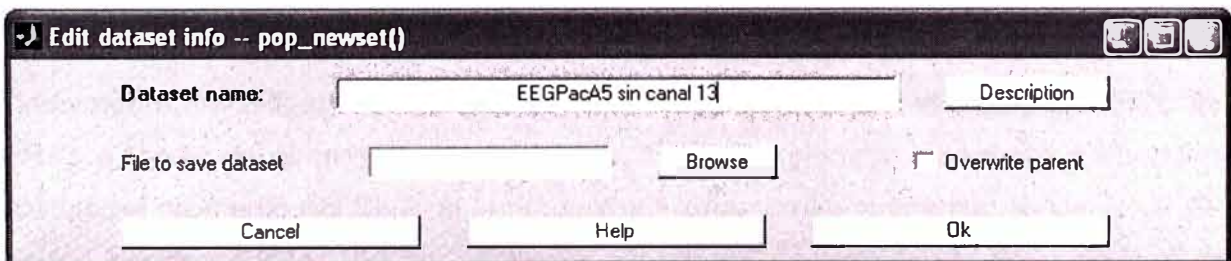


Fig. 3. 82 Grabación en el Dataset EEGPacA5 sin canal 13

ERP grand average/RMS - pop_comperp()

Datasets to add (ex: 1 3 4):

Datasets to subtract (ex: 5 6 7):

Plot difference

	avg.	std.	all ERPs
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Channels subset ([]=all):

Highlight significant regions (.01 -> p=.01)

Use RMS instead of average (check):

Low pass (Hz) (for display only)

Plottopo options ('key', 'val'):

Fig. 3. 84 Formulario para calcular promedios de ERPs

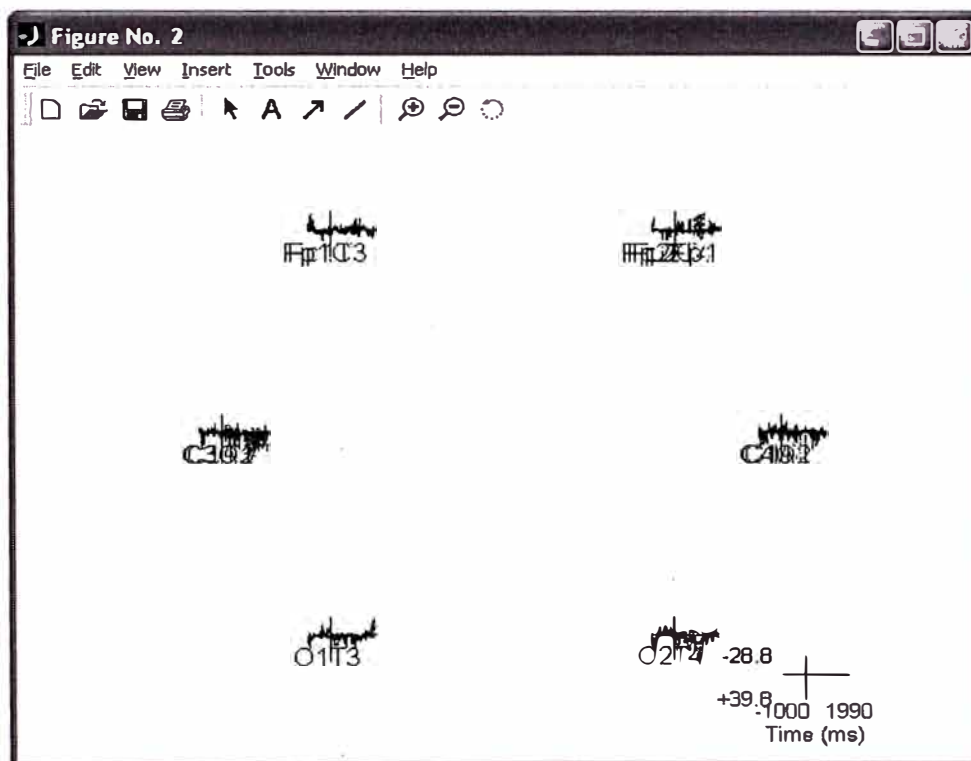


Fig. 3. 85 Promedios de todos los ERPs

Ahora, haga click sobre la traza del electrodo de posición **Fp1C3** y se mostrará el gráfico individual (Fig. 3.86). Es posible también visualizar en grafico negativo para la escala “y” ponga “ydir’, -1” en el campo “Plottopo options”.

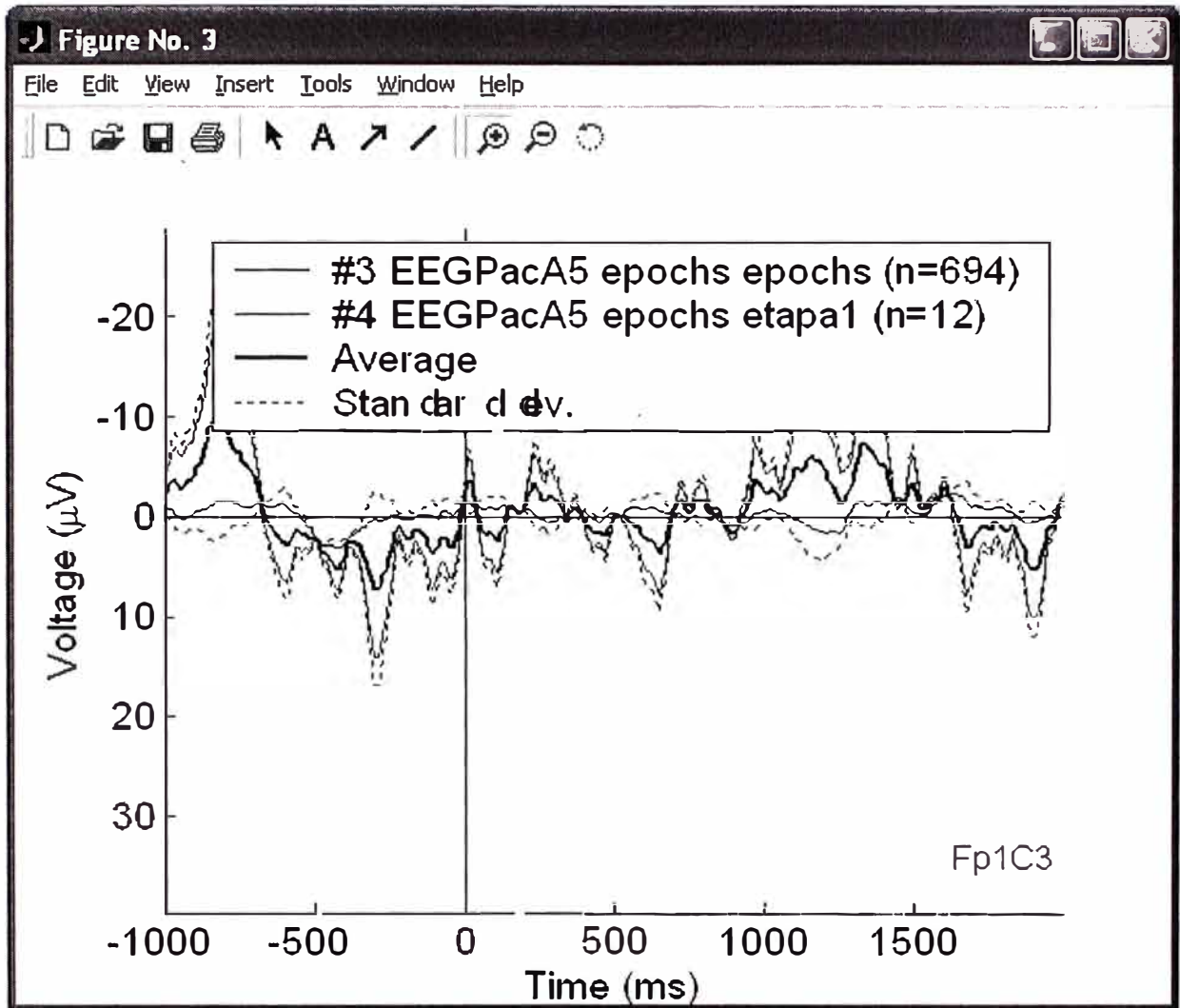


Fig. 3. 86 Gráfica individual de un ERP

Los ERPs para el conjunto de datos 3 y 4 son mostrados en azul y rojo. El promedio ERP para las dos condiciones es mostrado en “**negro resaltado**”. Regiones en los cuales son significativamente diferentes de “0” son resaltados usando dos pruebas en cada punto del tiempo de las señal. Estas pruebas comparan el valor de la distribución actual ERP con una distribución de la misma varianza y un significado “0”. Notar que estas pruebas no corrigen múltiples comparaciones. Los valores “p” en cada punto de tiempo puede ser obtenida con una llamada a la función `pop_comperp()` desde la línea de comandos.

- **Buscando latencias picos de ERP**

Aunque actualmente no se tiene herramientas de EEGLAB para determinar la amplitud pico de ERP y su latencia, uno puede utilizar la facilidad de ampliación para determinar manualmente la latencia de un pico en cualquier figura Matlab. Por ejemplo, en la figura previa, se puede seleccionar el icono “+”. Luego se amplía el pico máximo de la curva, tal como se muestra en la figura de abajo (Fig. 3.87), para ello, se presiona el click izquierdo

del Mouse para reducir y el derecho para ampliar. Con ello se puede ver la latencia y la amplitud con cualquier precisión deseada de la escala de los ejes.

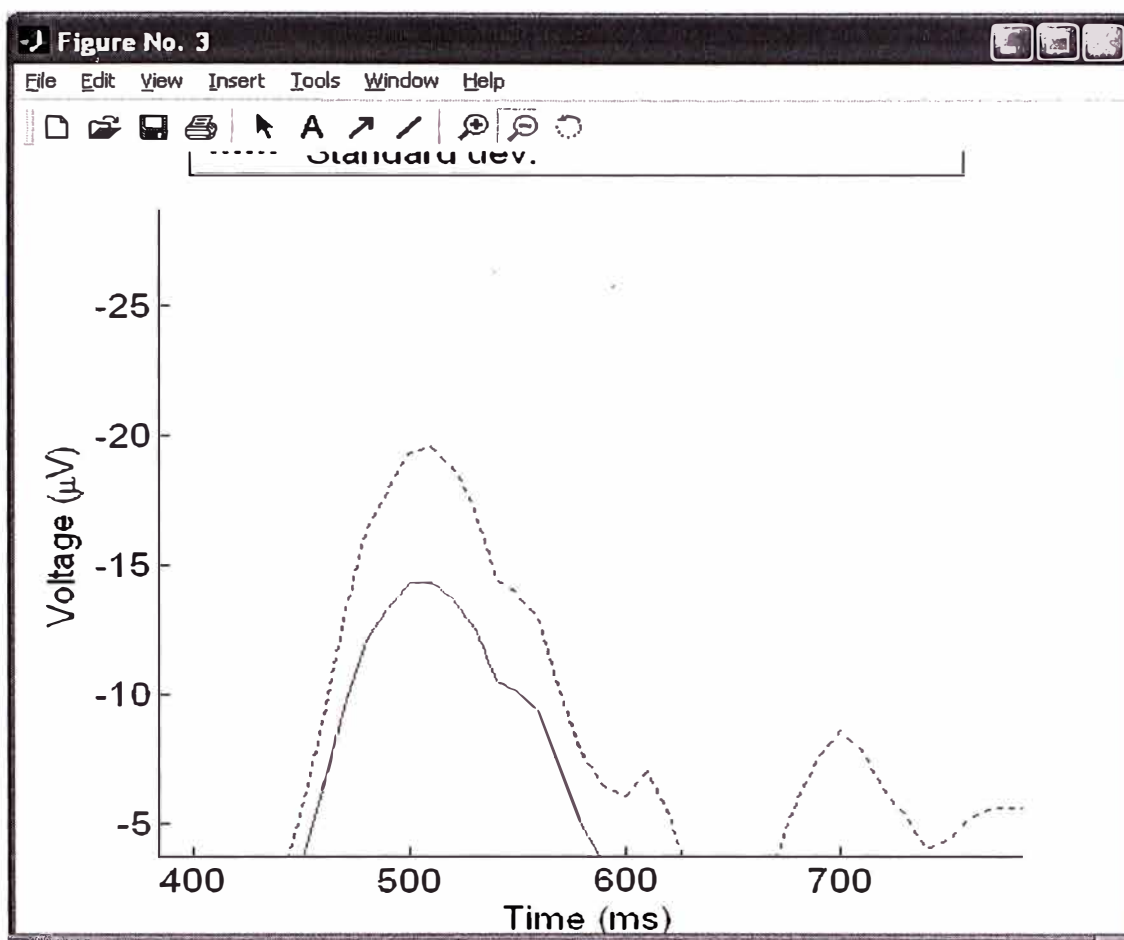


Fig. 3. 87 Amplitud pico y latencia de ERP

- **Comparando ERPs en dos condiciones**

Comparar promedios de ERP para las dos condiciones (objetivos presentados en la etapa1 y etapa2), seleccionamos **Plot > Sum/Compare ERPs**. En el primer campo de ingreso de datos de la ventana `pop_comperp()` ingrese el índice de los conjuntos de datos a comparar (datasets). Marque todos los casilleros de la columna **“avg.”**, ingrese 30 como la frecuencia de pasa bajo y **“title’, ‘Posición etapa1-etapa2”** en la opción de `topoplot()`, luego presione **OK** (Fig. 3.88).

ERP grand average/RMS - pop_comperp()

	avg.	std.	all ERPs
Datasets to add (ex: 1 3 4):	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Datasets to subtract (ex: 5 6 7):	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plot difference	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Channels subset ([]=all):

Highlight significant regions (.01 -> p=.01)

Use RMS instead of average (check):

Low pass (Hz) (for display only): 30

Plottopoptions ('key', 'val'): Help 'title','Posicion etapa1-etapa2'

Cancel Help Ok

Fig. 3. 88 Formulario para comparar dos datasets

Luego aparece la siguiente figura (Fig. 3.89).

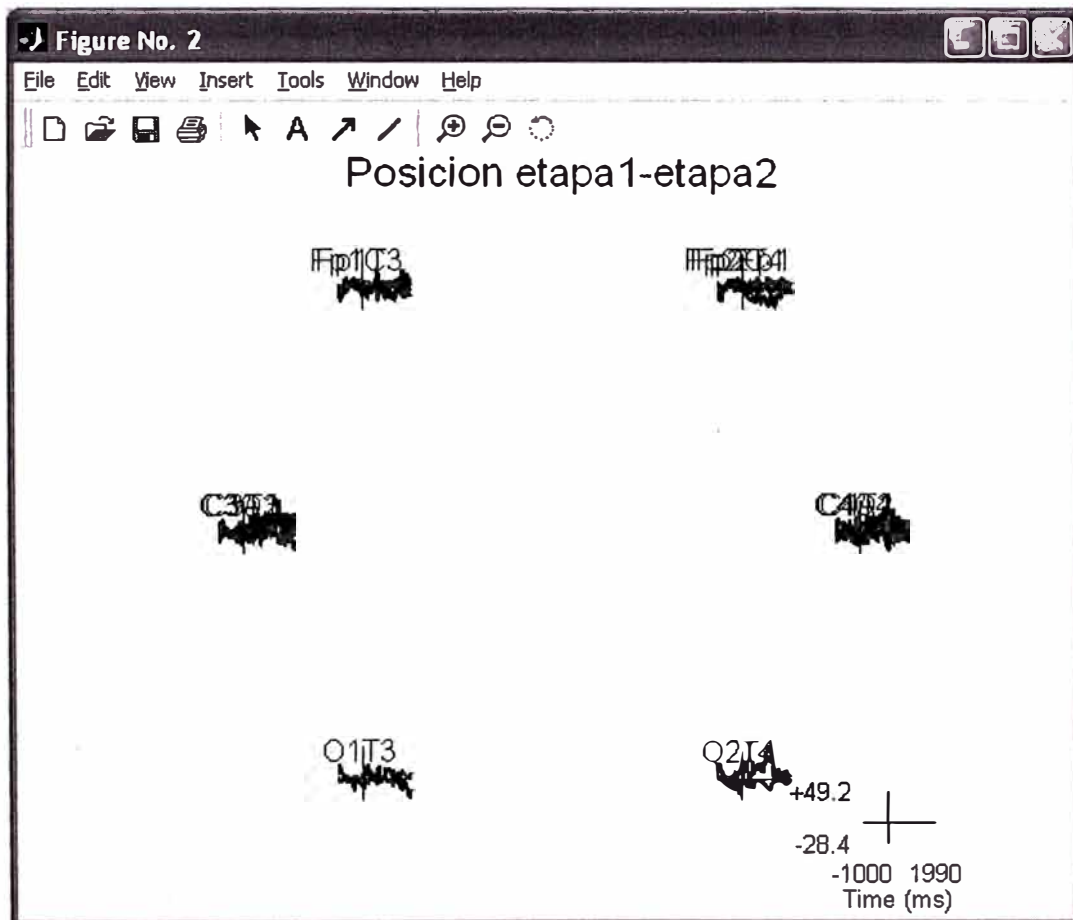


Fig. 3. 89 Grafica resultado de comparación de datasets de etapa1 y etapa2

Nuevamente, las trazas de los electrodos individuales pueden ser graficados en ventanas separadas haciendo clic en localizaciones de interés de la figura de abajo (Fig. 3.90).

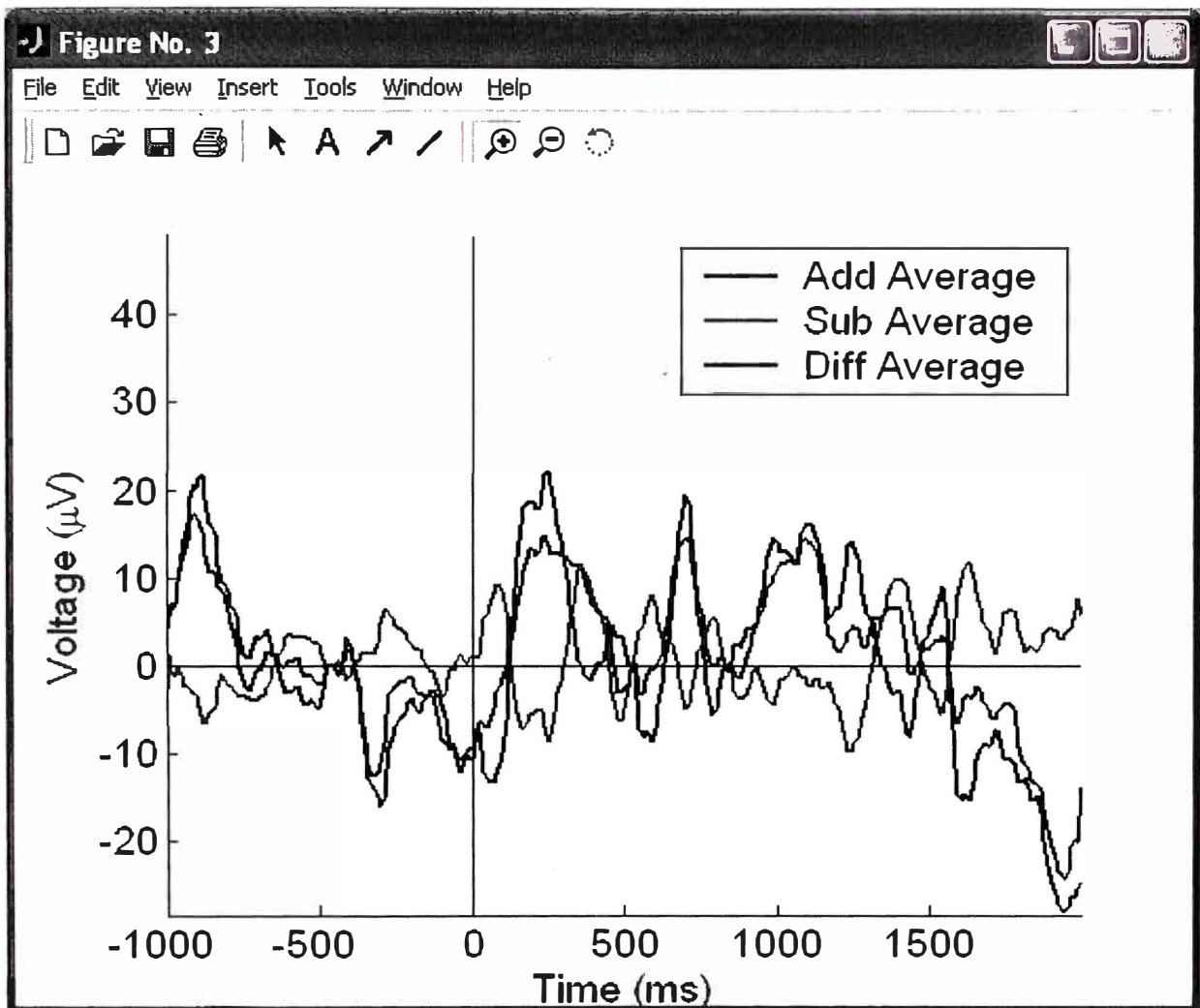


Fig. 3. 90 Gráfica individual de promedios de comparaciones de etapa1 y etapa2

Esta función puede también ser calculado y las diferencias ERP de gran significado entre condiciones a través de varios sujetos, y puede evaluarse diferencias significativas entre dos condiciones utilizando un t-test emparejado. Para realizar esto, para cada sujeto dentro de EEGLAB e ingresar los índices apropiados en la ventana de la función `pop_comperp()`.

- **Análisis gráfico de señales EEG**

Graficando imágenes ERP

Los campos de análisis de datos electrofisiológicos han sido dominados por el análisis de promedios de potencial de eventos relacionados (ERP) de una dimensión. Las imágenes ERP son afines, pero generalmente mas bidimensional 2-D (valores en tiempo por periodos (*epochs*)) vista de los datos. 'ERP images' son transformadas de imágenes 2-D

de los datos de los periodos, en los cuales los periodos de datos son primeramente ordenados a lo largo de algunas dimensiones relevantes, luego opcionalmente suavizar las señales y finalmente codificar con colores y graficarlo.

Como contrapuesto a los ERP, el cual generalmente existe en solamente una forma (promedio de tiempo fijo), el número de posibles imágenes ERP de un grupo de pruebas unitarias es bastante, los datos de pruebas pueden ser ordenados y graficados a lo largo de todas las muestras obtenidas. Sin embargo, no todo el ordenamiento de órdenes da igual entendimiento dentro de la dinámica del cerebro expresado en los datos. Esto está a la altura del usuario decidir que imagen ERP se va a estudiar (por defecto, las pruebas son organizadas en el orden en que aparecen en el experimento).

Se tiene que tener presente también que es fácil interpretar incorrectamente o sobre interpretar una imagen ERP. Por ejemplo, utilizando “ordenamiento de fase” en una frecuencia puede cegar al usuario la presencia de otro fenómeno oscilatorio en algunos datos en diferentes frecuencias. Nuevamente, es de responsabilidad de los usuarios la correcta interpretación de la evidencia que una imagen ERP 2-D muestra relativamente en la hipótesis de interés, igualmente es de responsabilidad del usuario la interpretación correcta de una serie de tiempos ERP 1-D.

Seleccionando un canal a graficar

Para graficar una imagen de una actividad ERP de un canal de datos en una sola prueba de nuestro conjunto de datos (dataset), podríamos primeramente elegir un canal a graficar. Por ejemplo, elegir un canal con alta potencia de banda alpha (cerca de 10 Hz). En el menú **Plot > Channel spectra and maps** graficamos la **etapa1** tal como se muestra a continuación (Fig. 3.91):

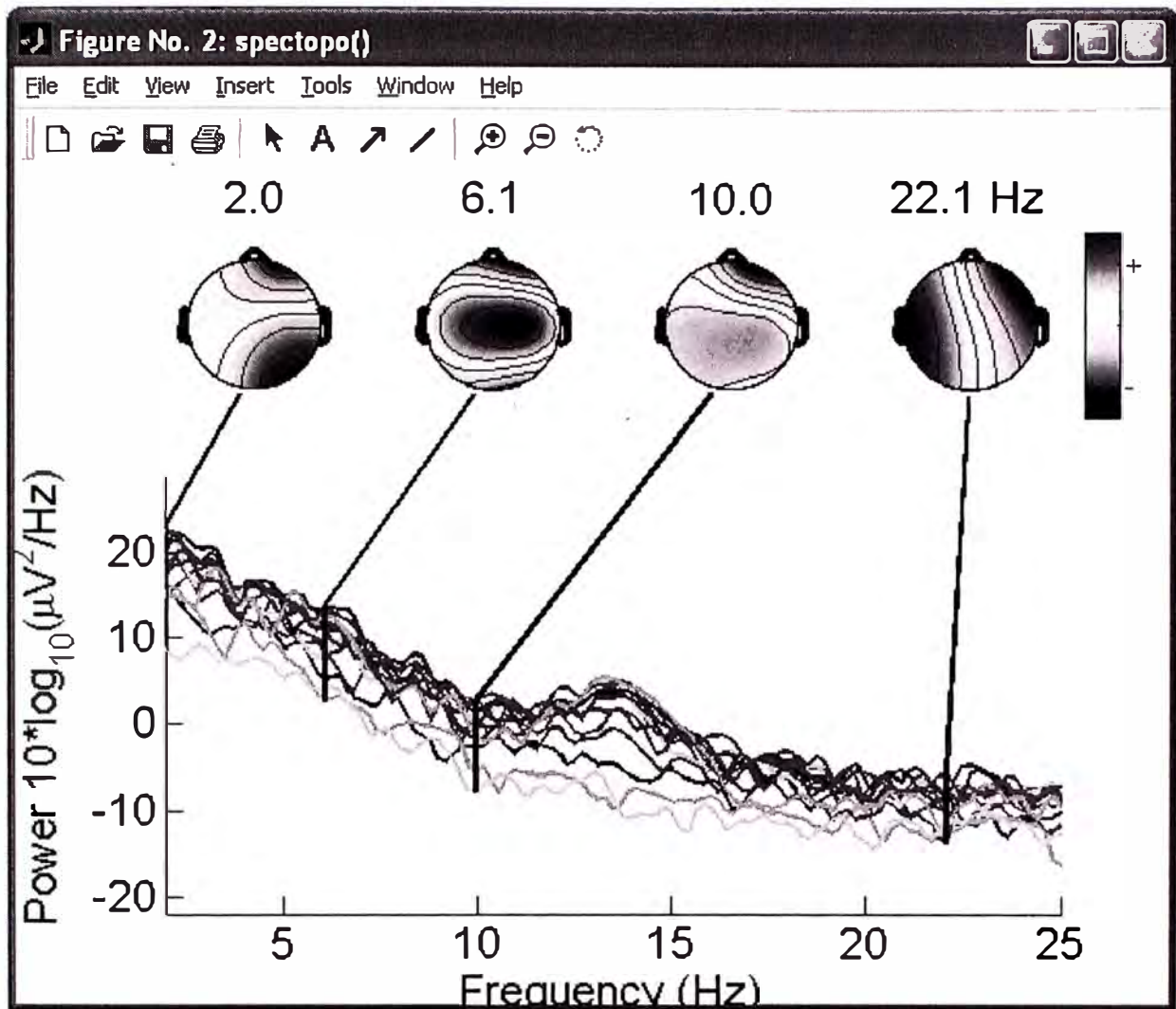


Fig. 3.91 Gráfica espectral y mapas de etapa1

En el gráfico de arriba podemos ver que las señales de potencia de la banda Beta (del ejemplo 22 Hz) están concentradas en la zona parietal. Para encontrar que electrodos están localizados en dicha zona, podemos graficar los nombres de los electrodos y localizaciones seleccionando la opción **Plot > Channel locaitons > By name** generando la siguiente figura (Fig. 3.92).

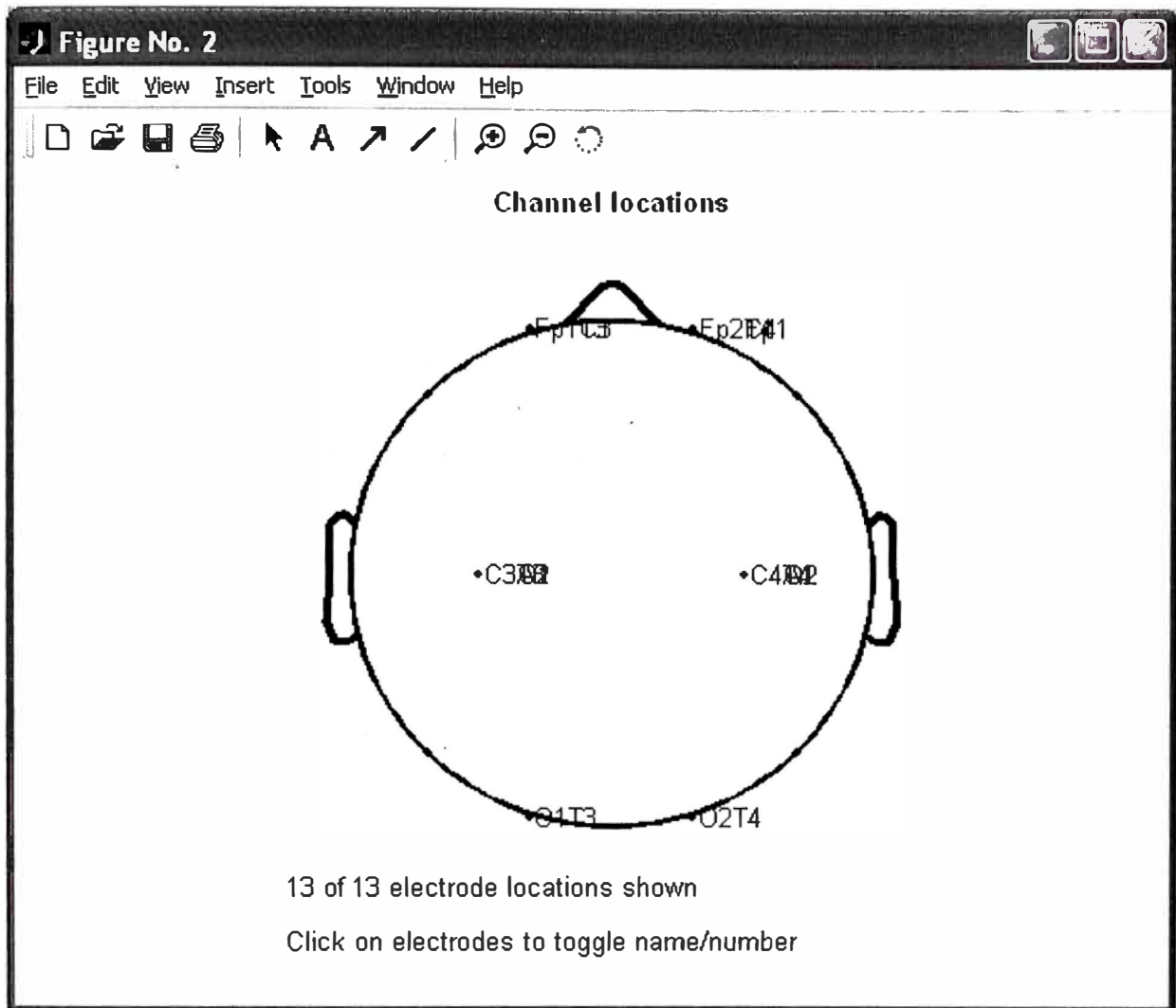


Fig. 3. 92 Localización de los electrodos identificado por nombres

Graficando ERP usando pop_erpimage()

Para el caso de un estudio de pruebas individuales de la actividad de algún canal en la forma de un gráfico ERP-Image, podemos realizarlo seleccionando la opción **Plot > Channel ERP image**. Esto muestra la siguiente pantalla (Fig. 3.93):

Channel ERP image -- pop_erpimage()

Channel: 11
Smoothing: 10
Downsampling: 1
Time limits (ms):

Figure title:
 Plot scalp map
 Plot ERP
 Plot colorbar
ERP limits (uV):
Color limits (see Help):

Sort/align trials by epoch event values
Epoch-sorting field: latency
Event type(s): etapa55
Event time range:
Rescale: no
Align:
 Don't sort by value
 Don't plot values

Sort trials by phase
Frequency (Hz | minHz maxHz): 9 11
Percent low-amp. trials to ignore: 0
Window center (ms): 0
Wavelet cycles: 3

Inter-trial coherence options
Frequency (Hz | minHz maxHz): 9 11
Signif. level (<0.20): 0.01
Amplitude limits (dB): -3 3
Coher limits (<=1):
 Image amps (Requires signif.)

Other options
Plot spectrum (minHz maxHz): 8 50
Baseline ampl. (dB):
Mark times (ms): 500
More options (see >> help erpimage):

Cancel Help Ok

Fig. 3. 93 Formulario para graficar con pop_erpimage()

Una imagen ERP es una imagen rectangular coloreada, en la cual cada línea horizontal representa la actividad que está ocurriendo en una prueba experimental individual (o un promedio del movimiento vertical o una prueba individual adyacente). La figura de abajo (no es una imagen ERP) explica el proceso de construcción de la gráfica de ERP-Image. El cambio de graficar pruebas individuales, tales como trazos de izquierda-a-derecha, en los cuales el potencial está codificado por la altura de la traza, codificamos sus valores a través de códigos de colores en las líneas rectas de izquierda-a-derecha, indicando el valor del potencial cambiando el color en cada punto de tiempo presentado durante la prueba. Por ejemplo en la siguiente imagen, tres diferentes periodos de pruebas (trazos azules) son codificados como tres diferentes líneas de colores, ver en la figura de abajo (Fig. 3.94).

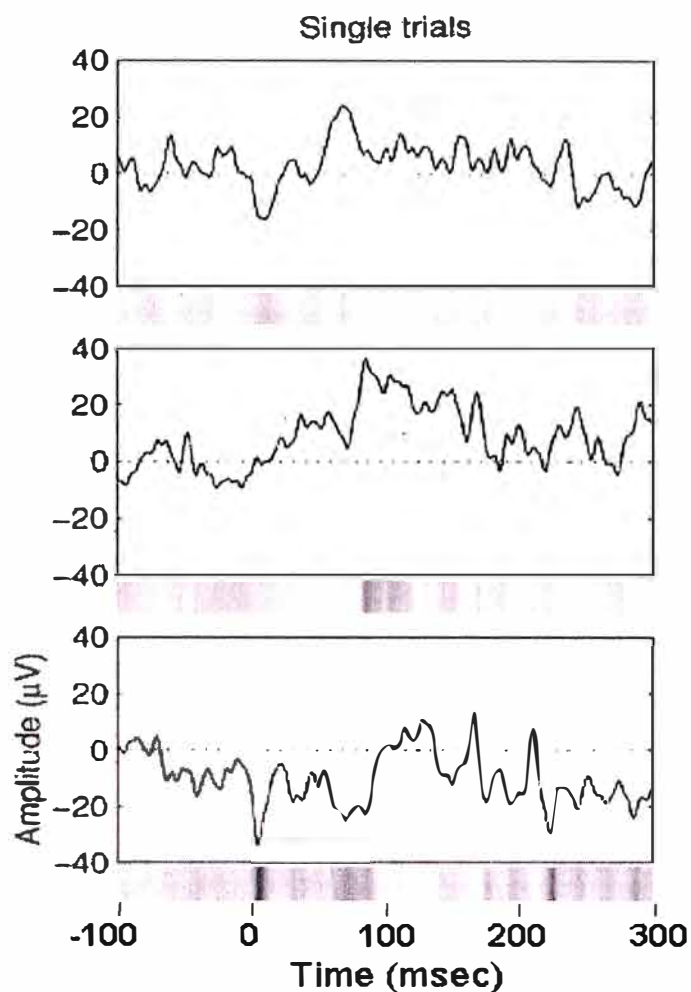


Fig. 3. 94 Gráfica de imagen ERP

En el gráfico de arriba, la secuencia de las líneas de colores que representa el conjunto de datos de todas las pruebas, representa una imagen ERP. En la representación estándar de la función `erpimage()` de la figura de abajo, la traza de la imagen ERP muestra el promedio de la actividad de pruebas individuales, ejemplo, el promedio ERP de la imagen de los periodos de datos. La gráfica de la cabeza (arriba a la izquierda) muestra un punto rojo que indica la posición del canal seleccionado (Fig. 3.95).

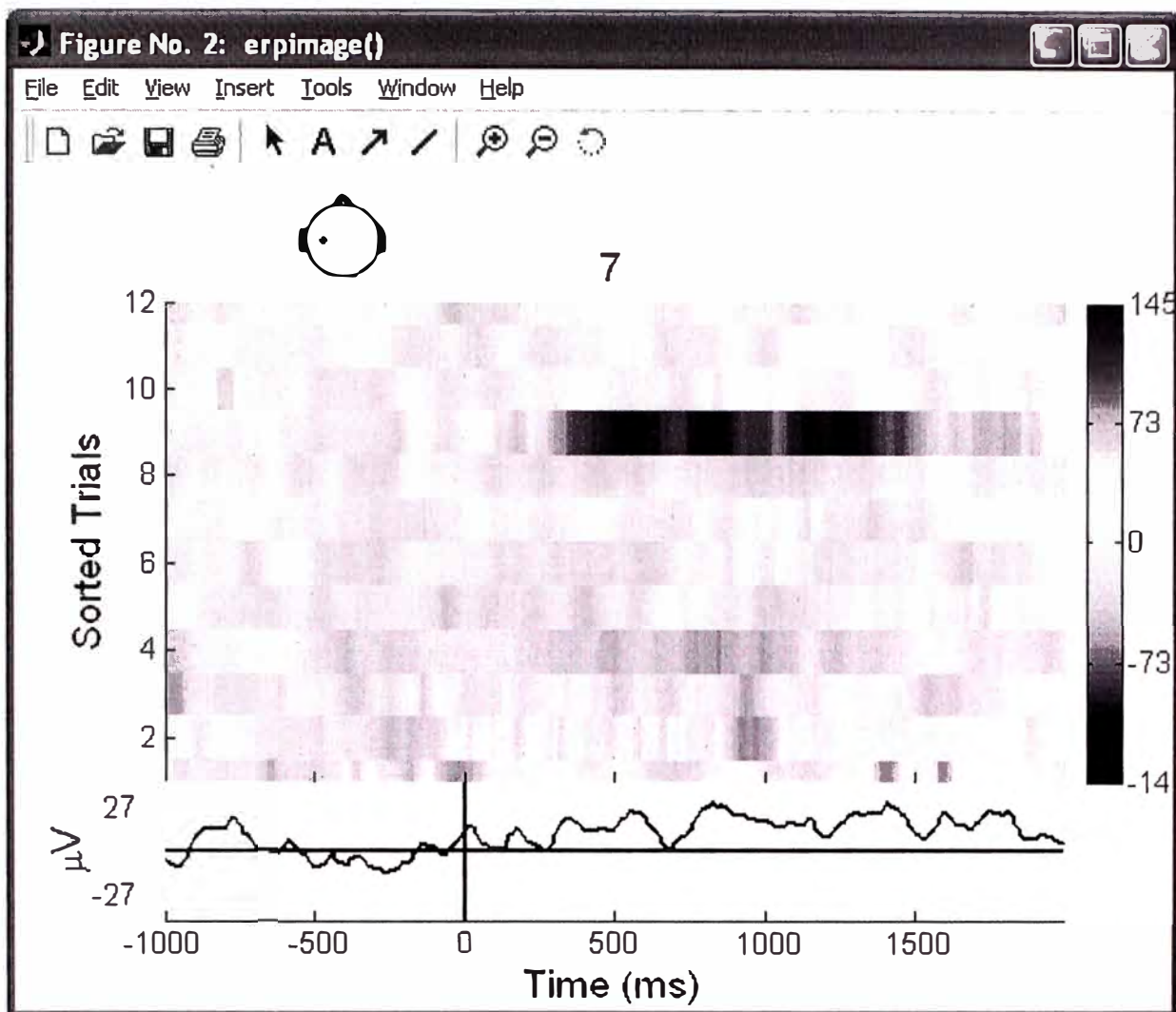


Fig. 3.95 Grafica erpimage() de un canal

Desde que las actividades en pruebas individuales tienen muchas variaciones, puede ser útil atenuar-afinar (*smoothing*) la actividad (verticalmente) sobre las pruebas usando un promedio modificable. Nuevamente invocando la función `pop_erpimage()` (ventana interactiva) y definiendo el ancho de atenuación (*smoothing*) en 10 en vez de 1. Ahora en el gráfico de abajo (Fig. 3.96) es fácil ver la banda dominante de oscilación.

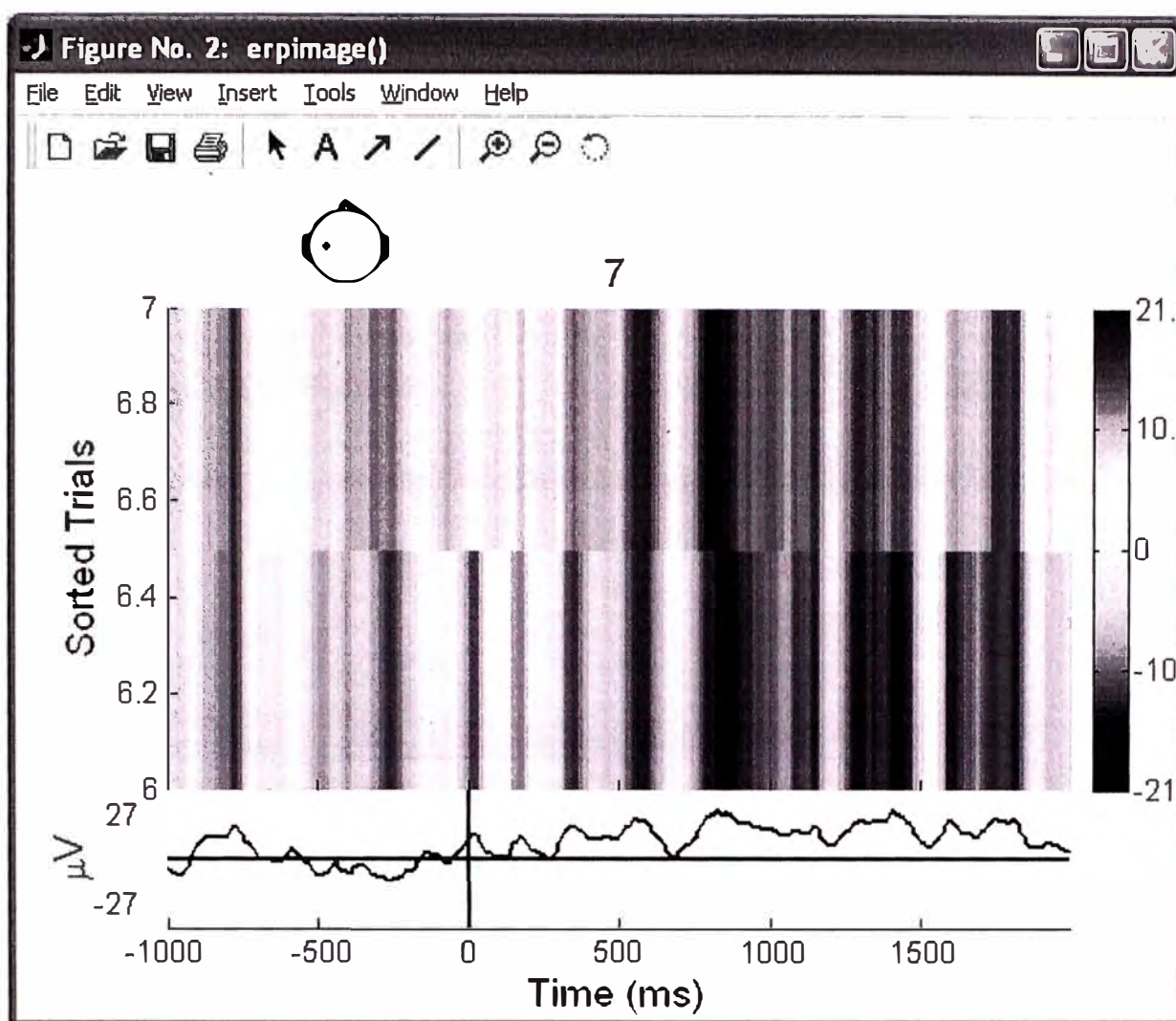


Fig. 3. 96 Gráfica de imagen erpimage() con atenuación de 10

Cuando estamos graficando un gran número de datos, no es necesario graficar cada prueba (atenuada o *smoothed*) como una línea horizontal (la resolución de la pantalla y/o impresora puede ser insuficiente para mostrar todos ellos). Para reducir el retardo de la imagen (y decrementar el tamaño del archivo para guardar), se puede eliminar algunas líneas de imágenes ERP (*smoothed*). Ingresando 4 en el campo "Downsampling" en la ventana de la función `pop_erpimage()` podría reducirse el número de líneas en la imagen ERP por un factor de 4. Si el ancho del "Smoothing" es mayor que $2 \times 4 = 8$, ninguna información será perdida en la imagen.

Clasificando pruebas en imágenes ERP

En las gráficas anteriores, las pruebas fueron graficadas en orden de ocurrencia en que se presentaron durante el experimento. Es también posible clasificar por cualquier otra variable la cual es codificada como un campo del evento perteneciente a cada prueba en

el conjunto de datos (dataset). En la figura de abajo demostraremos la clasificación de algunas pruebas con la finalidad de ver la latencia en el evento.

En la ventana `pop_erpimage()` nuevamente, primero presionamos el boton “**Epoch sorting field**”, y seleccionamos “**Latency**”. Luego, presionamos el boton “**Event type**”, y seleccionamos “**etapa1**” como se muestra abajo. En la consiguiente imagen ERP, las pruebas serán clasificadas por la latencia del evento “**etapa1**” (datos de nuestro ejemplo). Ingrese en el campo “**Event time range**” de “**-200 800**” ms a graficar la actividad siguiente (Fig. 3.97 y Fig. 3.98).

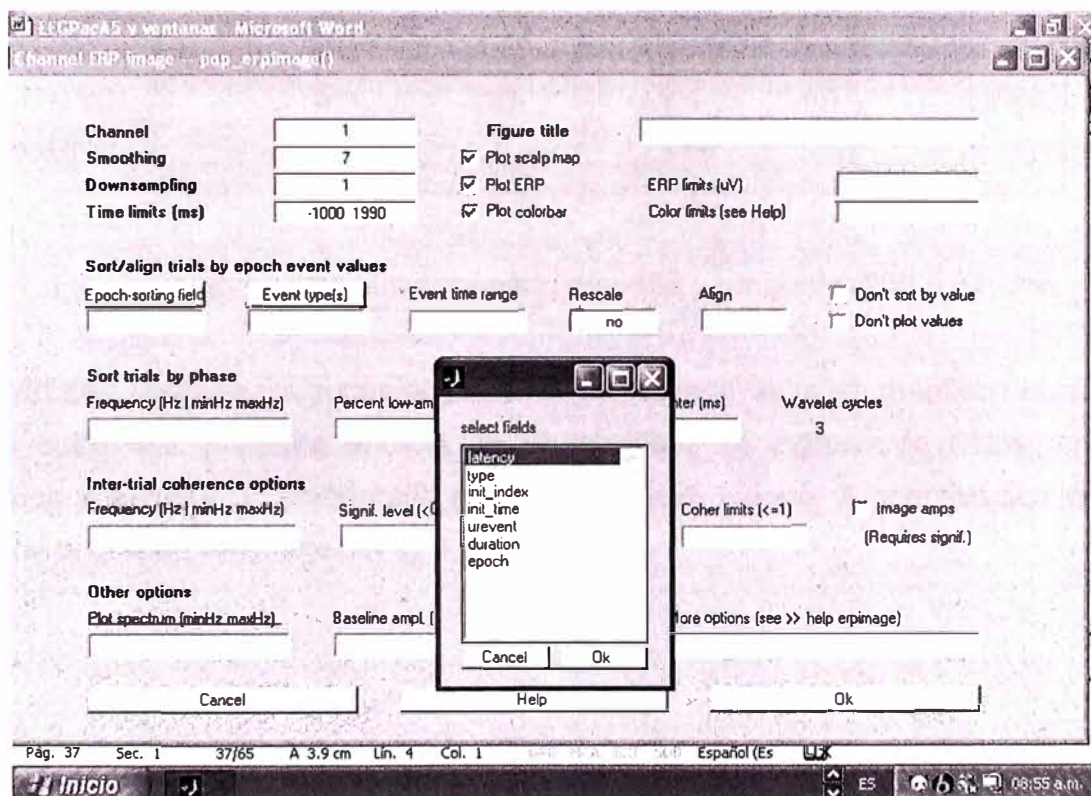


Fig. 3. 97 Formulario para graficar clasificando por latencia

Channel ERP image -- pop_erpimage()

Channel: 1
Smoothing: 7
Downsampling: 1
Time limits (ms): -1000 1990

Figure title:
 Plot scalp map
 Plot ERP
 Plot colorbar
ERP limits (uV):
Color limits (see Help):

Sort/align trials by epoch event values
Epoch-sorting field: latency
Event type(s): etapa52
Event time range: -200 800
Rescale: no
Align:
 Don't sort by value
 Don't plot values

Sort trials by phase
Frequency (Hz | minHz maxHz):
Percent low-amp. trials to ignore:
Window center (ms):
Wavelet cycles: 3

Inter-trial coherence options
Frequency (Hz | minHz maxHz):
Signal level (<0.20):
Amplitude limits (dB):
Coher limits (<=1):
 Image amps (Requires signif.)

Other options
Plot spectrum (minHz maxHz):
Baseline ampl. (dB):
Mark times (ms):
More options (see >> help erpimage):

Cancel Help Ok

Fig. 3. 98 Formulario seleccionando etapa52 y rango de -200 a 800 ms.

Nota: En esta ventana y algunas otras ventanas interactivas, si se mantiene el cursor del mouse sobre las etiquetas encima de un casillero de ingreso de datos, en pocos segundos aparecerá un comentario que explica dicho campo. A continuación vemos la figura de la función erpimage() (Fig. 3.99).

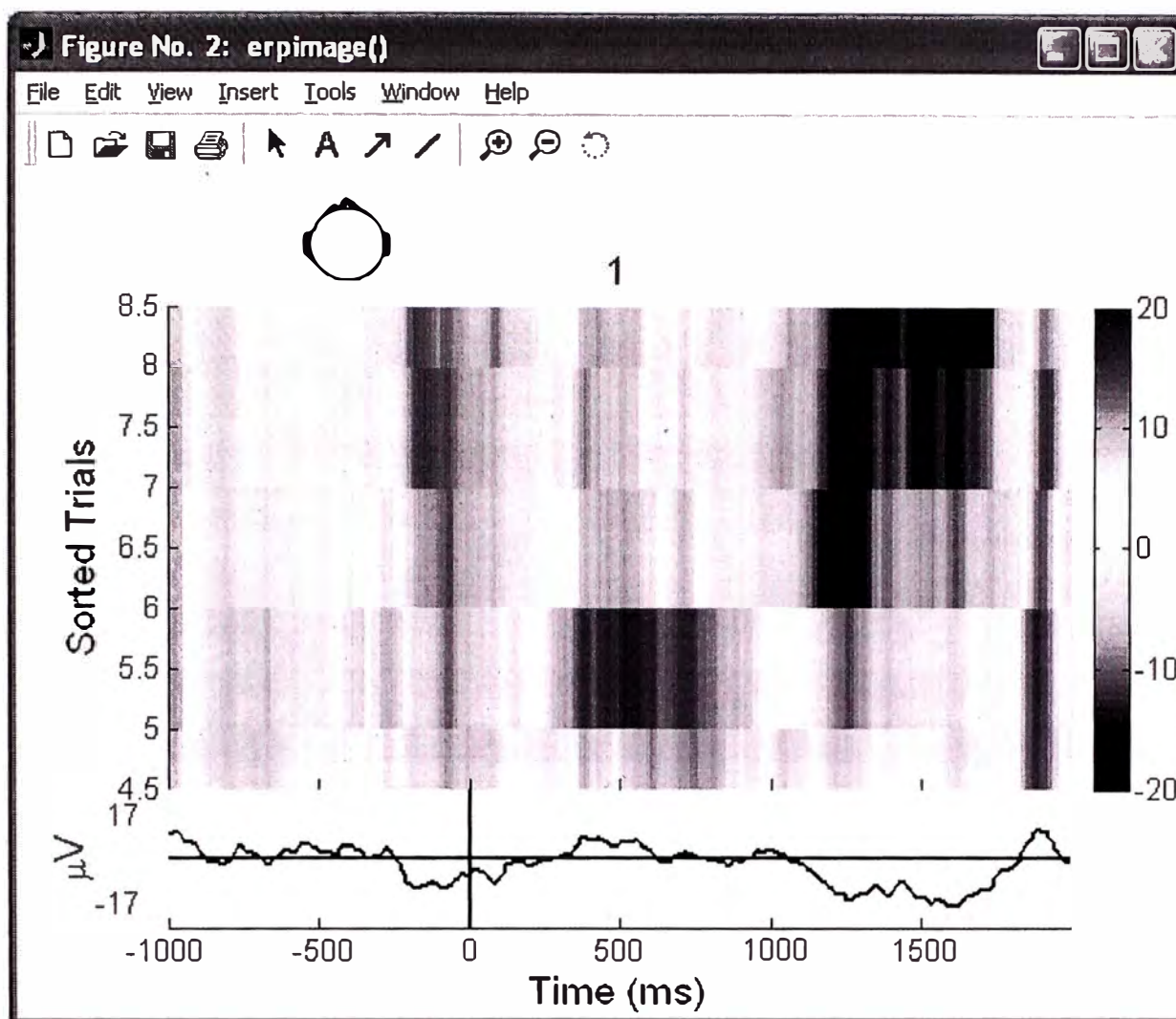


Fig. 3. 99 Gráfica clasificando por latencia tipo de evento y rango de tiempo

En general, el usuario puede clasificar con cualquier valor del campo "event". Por ejemplo, volver a invocar la ventana `pop_erpimage()`, presionar el botón "**Epoch-sorting Field**", y seleccionar "**epoch**" en vez de "**latency**". Limpiar en el campo "**Event type**". Finalmente ingrese "**yes**" en el campo de "**Rescale**". Luego presione **OK**. En la gráfica resultante de `erpimage()` (Fig. 3.100 y Fig. 3.101), las pruebas son clasificadas por el inicio de los periodos (*epochs*) los cuales obedecen a una señal preconcebida de adquisición de datos (algún estímulo ofrecido al sujeto), normalizando automáticamente los valores, acomodando el área de pos-estímulo. Notar que el ancho de afinamiento (*smoothing*) es aplicado a los datos individuales y a las variables de clasificación.

Channel ERP image -- pop_erpimage()

Channel	1	Figure title	
Smoothing	10	<input checked="" type="checkbox"/> Plot scalp map	
Downsampling	1	<input checked="" type="checkbox"/> Plot ERP	ERP limits (uV)
Time limits (ms)	-1000 1990	<input checked="" type="checkbox"/> Plot colorbar	Color limits (see Help)

Sort/align trials by epoch event values

Epoch-sorting field	Event type(s)	Event time range	Rescale	Align	<input type="checkbox"/> Don't sort by value
epoch			yes		<input type="checkbox"/> Don't plot values

Sort trials by phase

Frequency (Hz minHz maxHz)	Percent low-amp. trials to ignore	Window center (ms)	Wavelet cycles
			3

Inter-trial coherence options

Frequency (Hz minHz maxHz)	Signif. level (<0.20)	Amplitude limits (dB)	Coher limits (<=1)	<input type="checkbox"/> Image amps (Requires signif.)

Other options

Plot spectrum (minHz maxHz)	Baseline ampl. (dB)	Mark times (ms)	More options (see >> help erpimage)

Cancel Help Ok

Fig. 3. 100 Formulario eligiendo epoch y Rescale

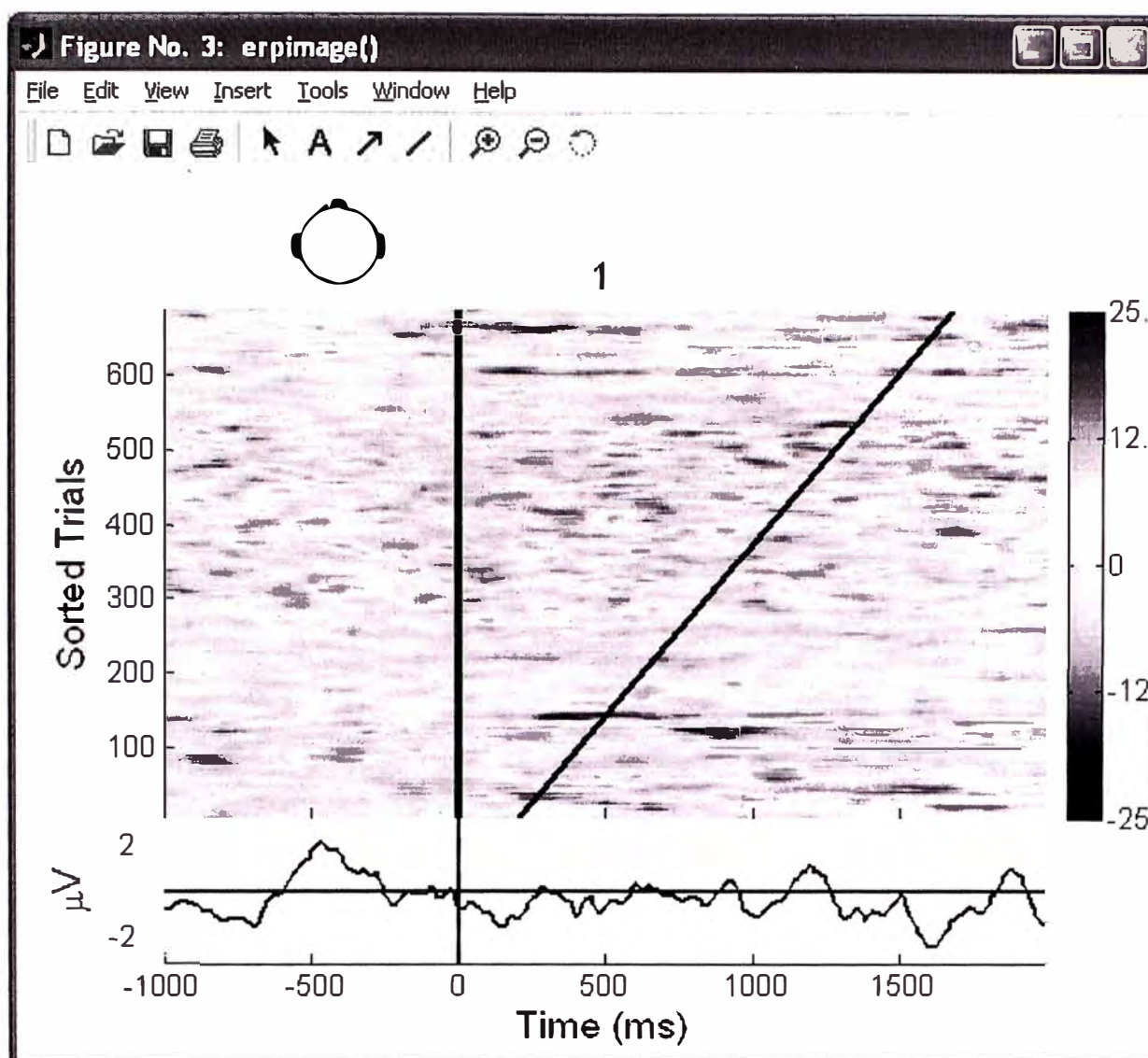


Fig. 3. 101 Grafica clasificado por epoch y rescale

Ahora, volviendo a seleccionar la latencia del evento etapa1 como una variable de ordenamiento de pruebas (presionar el botón "**Epoch-sorting field**" y seleccione "**latency**" y presione el botón "**Event Type**" para seleccionar etapa1). Ingrese "**no**" en el campo de "**Rescale**" (otro tiempo de reacción podría ser automáticamente normalizado). Usar la entrada de **Align** para realinear los datos de prueba en la variable de clasificación (aquí es el tiempo de reacción) y el cambio de límites de tiempo. El valor de latencia dada en **Align** será usado para especificar el tiempo 0. Para seleccionar la media de los valores de clasificación de prueba, para especificar el nuevo tiempo en cero, nuestra convención es usar "**Inf**" (el símbolo de Matlab) para el valor de infinito. Si desea definir con un valor diferente (por ejemplo, mientras graficamos una imagen ERP para un sujeto, se puede desear usar la media de uno de los tipos utilizados en el cálculo de todos los sujetos), simplemente ingrese el valor en **ms** en el campo "**Align**". Nota: El re-

alineamiento temporal de periodos de datos (*epochs*), relativas uno a otro, resultará en pérdida de datos en las esquinas de la parte izquierda-baja y derecha-alta de la imagen ERP. La función ERP-Image muestra en verde (0) y retorna estos valores como “NaN”s (notación Matlab que significa not-a-number) (Fig. 3.102).

Fig. 3. 102 Formulario para gráfica con latency y Align

Ya que nuestras señales han sido tomadas a un paciente que estaba durmiendo, no tenemos un estímulo específico para que la señal de respuesta pueda ser identificada, por lo que el gráfico resultante que se muestra a continuación no muestra otra línea que identifique el evento resultante (Fig. 3.103).

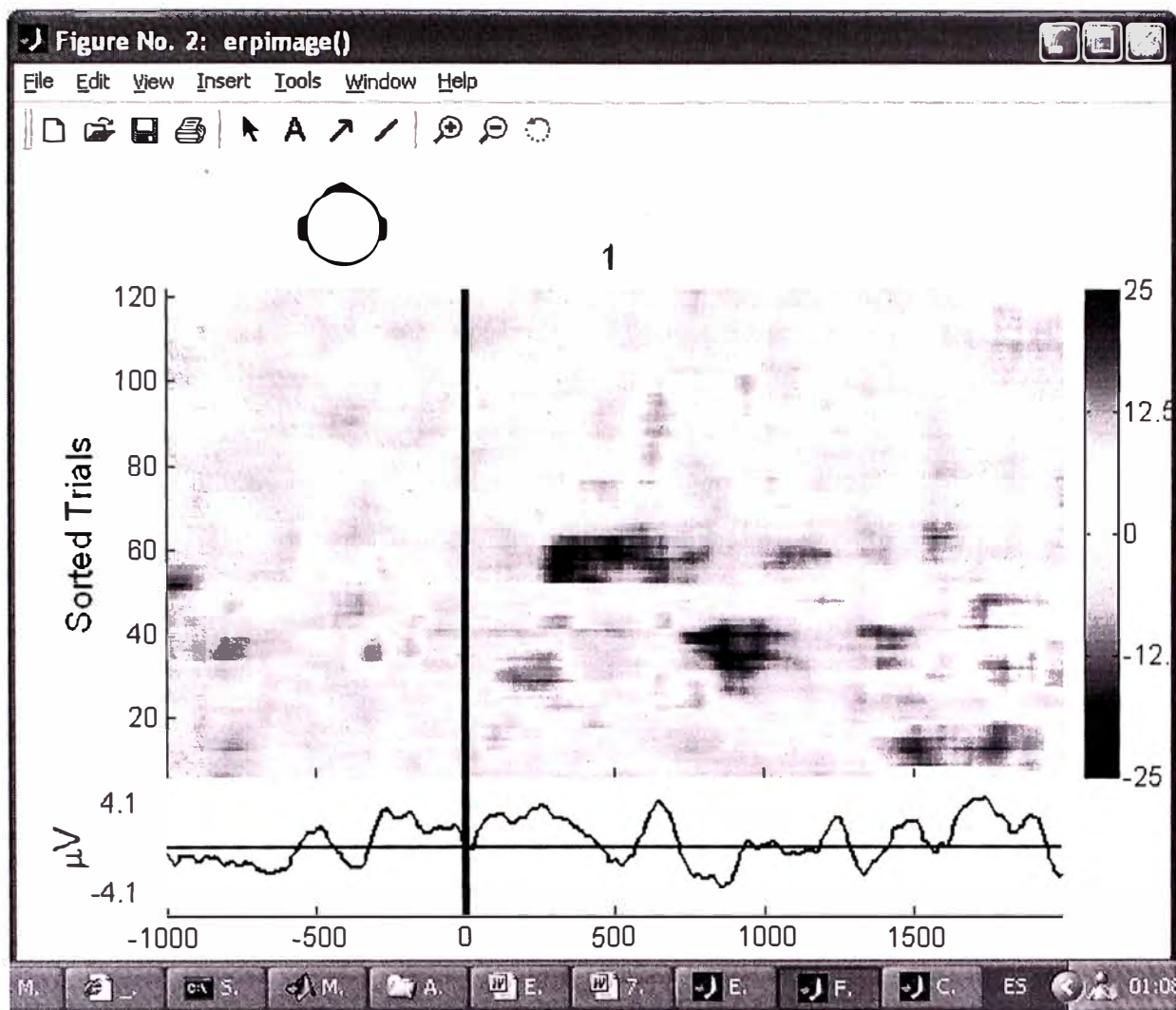


Fig. 3. 103 Gráfica con latency y Align

Graficando imagines ERP con opciones espectrales

Ahora vamos a experimentar clasificando pruebas por el valor de la fase EEG en una ventana de tiempo / frecuencia especificada. Aunque los valores de los tipos de eventos pueden ser mostrados en figuras ERP-Image clasificada en fase, omitiremos estos por simplicidad.

Para hacer esto, retornamos a la ventana `pop_erpimage()` del menú. Limpiar los contenidos previos de los campos "**Epoch- sorting field**", "**Event type**" y "**Align**". Luego en la sección "**Sort trials by phase**", ingrese **10 (Hz)** en el campo de "**Frequency**" y **0 (ms)** en el campo "**Window Center**". Ingrese en el campo "**Time limits (ms)**" **-200 800** cerca de los límites (ms) de inicio de los eventos en los periodos utilizados, esta opción se puede observar en la parte superior de la ventana (Fig. 3.104).

Fig. 3. 104 Formulario utilizando campos para gráficas de opciones espectrales.

Luego, se obtiene la figura ERP-image de abajo (Fig. 3.105).

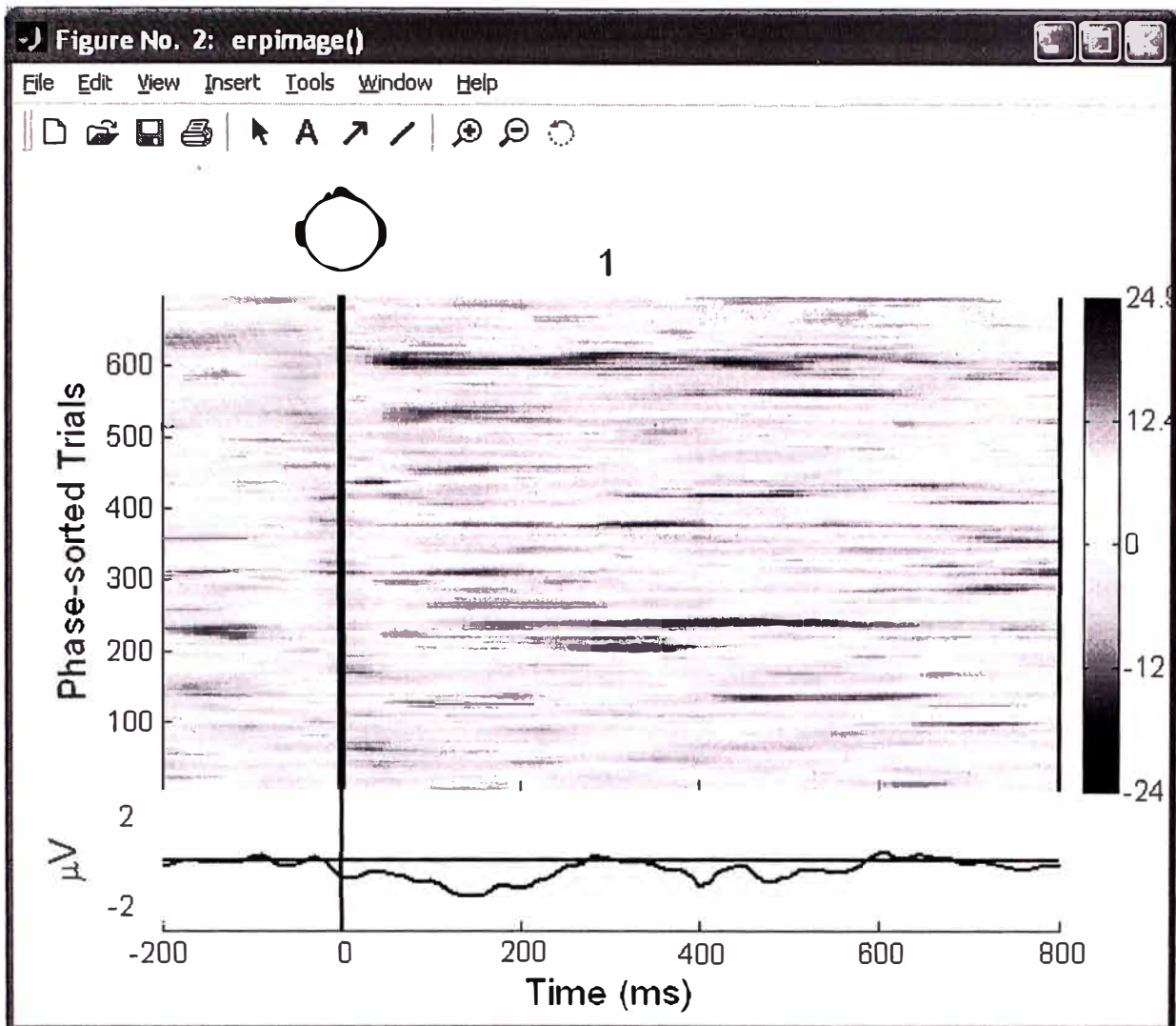


Fig. 3. 105 Gráfica clasificando por el valor de la fase EEG

Notar que justo la presencia de la ocurrencia de un evento al inicio de la banda oblicua roja, esto es producido por el ordenamiento de la fase: la fase (ejemplo, la latencia de los picos de las ondas) es uniformemente distribuido a través del re-ordenamiento de las pruebas.

En este cálculo, tres ciclos de una pequeña onda de **10 Hz** fueron aplicados a la ventana en cada prueba centrada en el **tiempo 0**. El ancho de la onda fue **300 ms** (tres ciclos de **10 Hz de 100 ms**). Por consiguiente, es extendido de -150 ms a 150 ms. Después la pequeña onda fue aplicada a cada prueba, las pruebas fueron clasificadas con la función en el orden de valores de fase ($-\pi$ a π) y mostradas en imágenes ERP en ese orden (de abajo hacia arriba). La dominación de la actividad alrededor de 10 Hz en las pruebas, conjuntamente con el valor de prueba de 10 del campo "**smoothing**" aplicadas, hace que la coherencia de fase entre las pruebas adyacentes sea obvia en esta vista.

Notar que podemos tener aplicado clasificación de fase de las pruebas utilizando cualquier ventana de tiempo / frecuencia. El resultado podría depender de la longitud de las frecuencias seleccionadas en los datos, particularmente en su grado de movimiento (ejemplo, gran impulso de esta frecuencia exhibida por estos datos), y su cierre de fase (o no) en los eventos experimentales. Las imágenes ERP ordenadas por fase usando diferentes ventanas de tiempo y frecuencia representa diferentes caminos para “transportar a través” (*fly through*) datos EEG complejos (canales únicos).

Para ver el ordenamiento por fase mas claramente, mantenemos las mismas definiciones de la grafica anterior, pero ahora ingresamos el valor de **50** en el campo “**Percent low-amp. Trials to ignore**”. Aquí el 50% de las pruebas con potencia menores a 10 Hz (alpha) en la ventana de tiempo seleccionada serán rechazadas, solamente los otros 40 (50% alpha mayores) serán graficados. En la grafica de abajo (Fig. 3.106), podemos ver mejor como la onda alpha parece re-sincronizar siguiendo la señal de inicio del periodo (estimulo). Antes del tiempo cero, la fase alpha es mas o menos aleatorias (uniformemente distribuida) y hay una pequeña actividad en el promedio ERP. Alrededor de 200 ms, la actividad alpha parece (parcialmente) re-sincronizar con la señal estimulo.

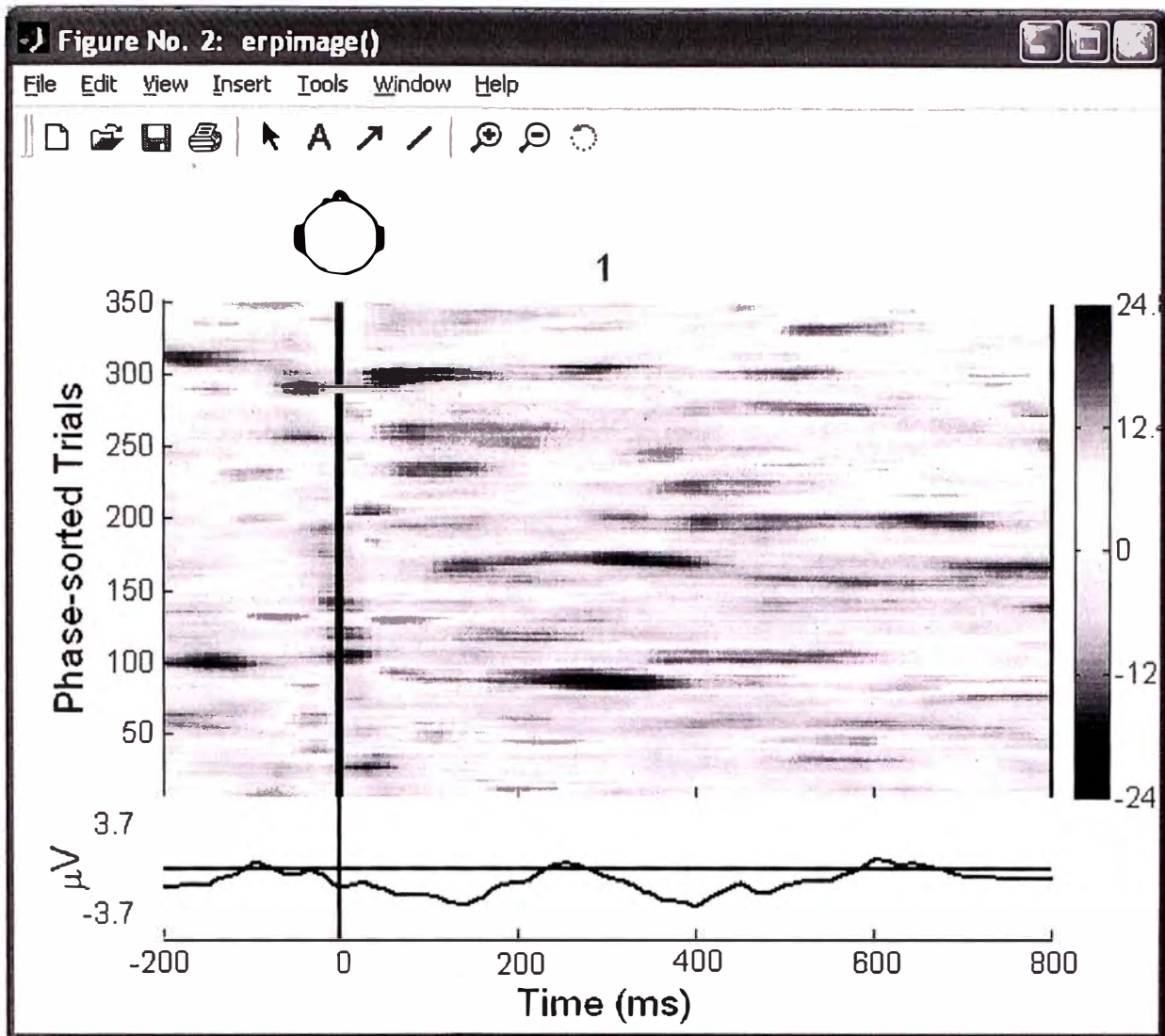


Fig. 3. 106 Gráfica con el campo "Percent low-amp.Trial to ignore" = 50

La interpretación de estas pruebas como la representación de la sincronización de la fase necesita no solamente ser basadas en una impresión visual. Para una evaluación estadística, si la actividad alpha es parcialmente re-sincronizada (es en parte fase reiniciada) por el estímulo (señal inicio del evento), necesitamos graficar la coherencia de fase (o factor de bloqueo de fase) entre la secuencia del estímulo y los datos de post-estímulo. Esta medición, la coherencia de Inter-pruebas ITC (Inter-Trial Coherence) en esta terminología, toma valores entre 0 y 1. Un valor de 1 para la ventana de tiempo frecuencia de interés indica que la fase alpha (en esta ventana de latencia) es constante en cada prueba. Un valor de cero ocurre cuando los valores de la fase en todas las pruebas son uniformemente distribuidos alrededor de un círculo unitario. En la práctica, valores algo así como cero son esperados para cualquier número finito de pruebas con fase distribuida aleatoriamente.

Para graficar los ITC en una imagen ERP (Fig. 3.107 y Fig. 3.108), elegimos ingresar los siguientes parámetros en la ventana `pop_erpimage()`: omitimos el valor del campo “Percent low-amp. of Trials to ignore” e ingresamos los valores “9 11” debajo de la sección “Sort trials by phase” en el campo “Frequency” y también los valores “9 11” en la sección “Inter-Trial Coherent” en el campo “Frequency”. Notar que estos datos ingresados tienen que ser iguales. Ingresando un rango de frecuencias en vez de una sola frecuencia (como este caso 10 Hz.) se le indica a la función `erpimage()` buscar la frecuencia con máximo potencia dentro de los datos de las frecuencias dadas (aquí por ejemplo entre 9 y 11 Hz.). Finalmente ingresar el valor “0.01” en el campo “Signif. Level” y presionar “OK”.

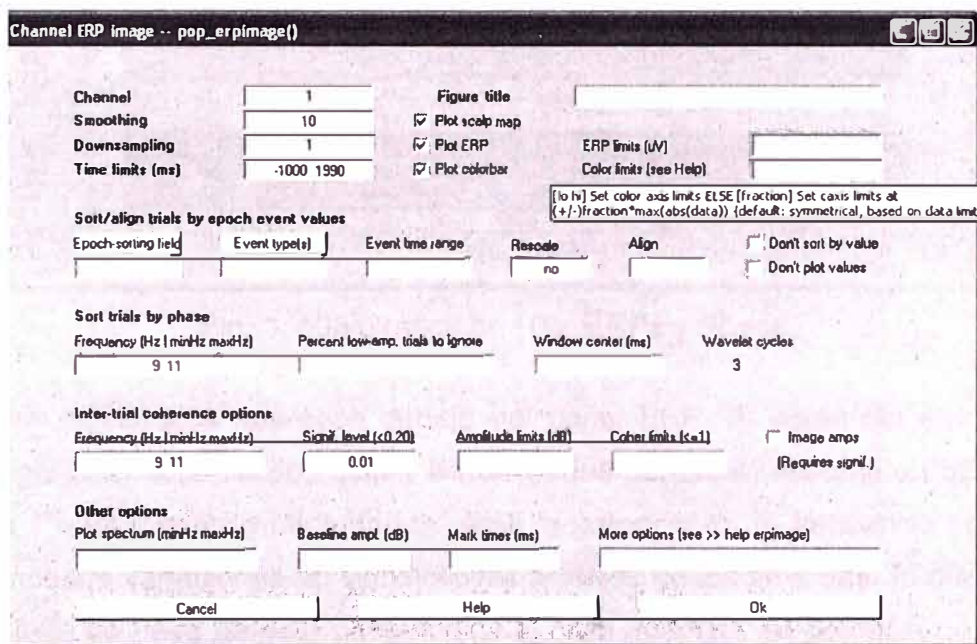


Fig. 3. 107 Forma para graficar los ITC en una imagen ERP

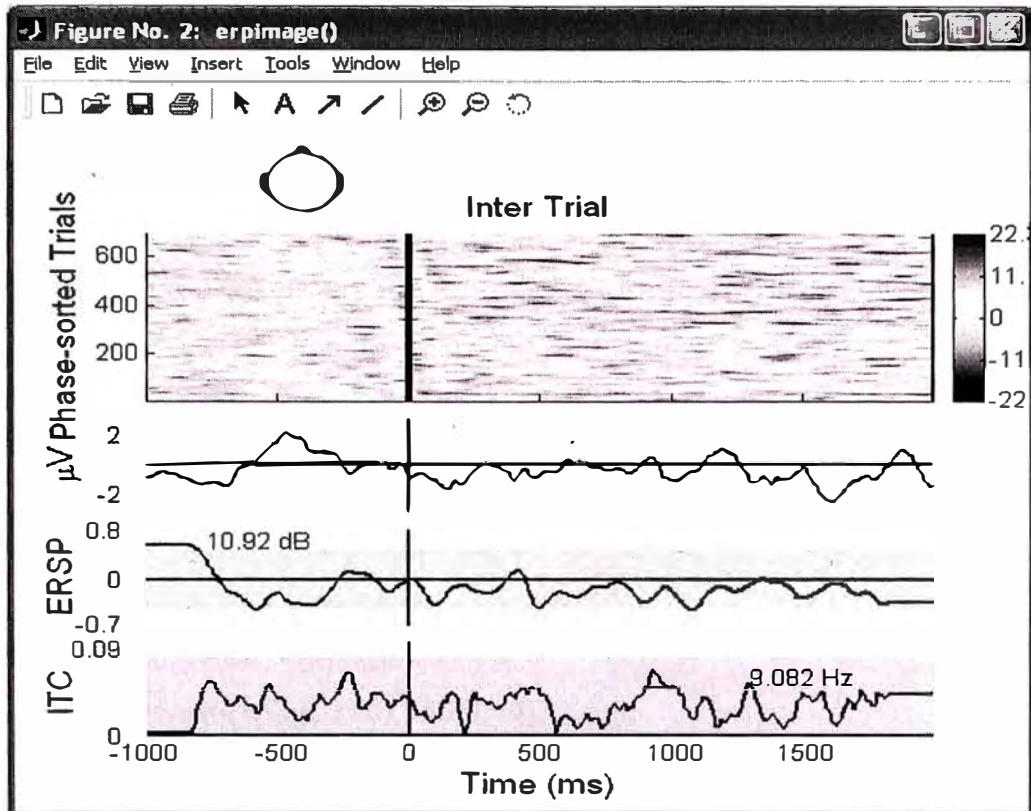


Fig. 3. 108 Gráfica de ITC, ERPs y Phase.

Dos gráficos adicionales aparecen debajo del panel ERP. El panel del medio muestra cambios significativos en la potencia a través de los periodos (*epochs*) en dB. La región azul indica 1% de límite de certidumbre. Aquí la potencia en la frecuencia seleccionada (9.82 hz) muestra variaciones no significativas a través de los periodos. El número "10.92 dB" en la línea de base de este panel indica el nivel absoluto de potencia de la línea de base. Nota: Comparar resultados, es a veces útil para definir estos valores manualmente en la ventana principal de ERP-Image.

Los ERP en el ejemplo, ¿son originados a través de una sincronización parcial de fase o re inicializados siguiendo el inicio del estímulo?

En un caso puro de (parcial) sincronización de fase: La potencia EEG (en las frecuencias relevantes) constantemente reserva en el intervalo de post-estímulo.

El valor de ITC es significativo durante los ERP, pero menor que 1 (bloqueo completo de fase). En nuestro caso, la figura de arriba, muestra un incremento del post-estímulo significativo en el ITC alpha acompañado por un pequeño (aunque no significativo) en la potencia alpha. En general, un ERP podría generarse desde una sincronización parcial de fase de una actividad en curso combinada con un incremento de estímulo relacionado en una potencia de EEG.

Es importante no sobre interpretar los resultados del ordenamiento de las fases en graficas ERP. Por ejemplo, la siguiente invocación desde la línea de comandos de Matlab simula 256 1-s periodos de datos usando Ruido Blanco Gaussiano, y el filtro pasabajo es menor (simulado) a 12 Hz, y la gráfica muestra la fase de 10 Hz ordenada del gráfico ERP resultante de los datos. La figura (Fig. 3.109) parece identificar temporalmente la actividad coherente de 10 Hz en el ruido actual. El panel de la amplitud media ubicado debajo de la Imagen ERP muestra, sin embargo, que la amplitud a 10 Hz. no cambia significativamente a través de los periodos (*epochs*). Y el panel de mas abajo muestra la coherencia "Inter-trial" que es también no significativa (como se confirma visualmente por la diagonal recta de la onda de 10 Hz en el frente y en el centro de la imagen ERP).

```
% Simulate 256 1-s epochs with Gaussian noise
% at 256-Hz sampling rate; lowpass < 12 Hz
>> data = eegfilt(randn(1,256*256),256,0,15);

% Plot ERP image, phase sorted at 10 Hz
>> figure;
>> erpimage(data,zeros(1,256),1:256,'Phase-sorted Noise',1,1,...
    'phasesort',[128 0 10], 'srate',256,...
    'coher',[10 10 .01], 'erp','caxis',0.9);
```

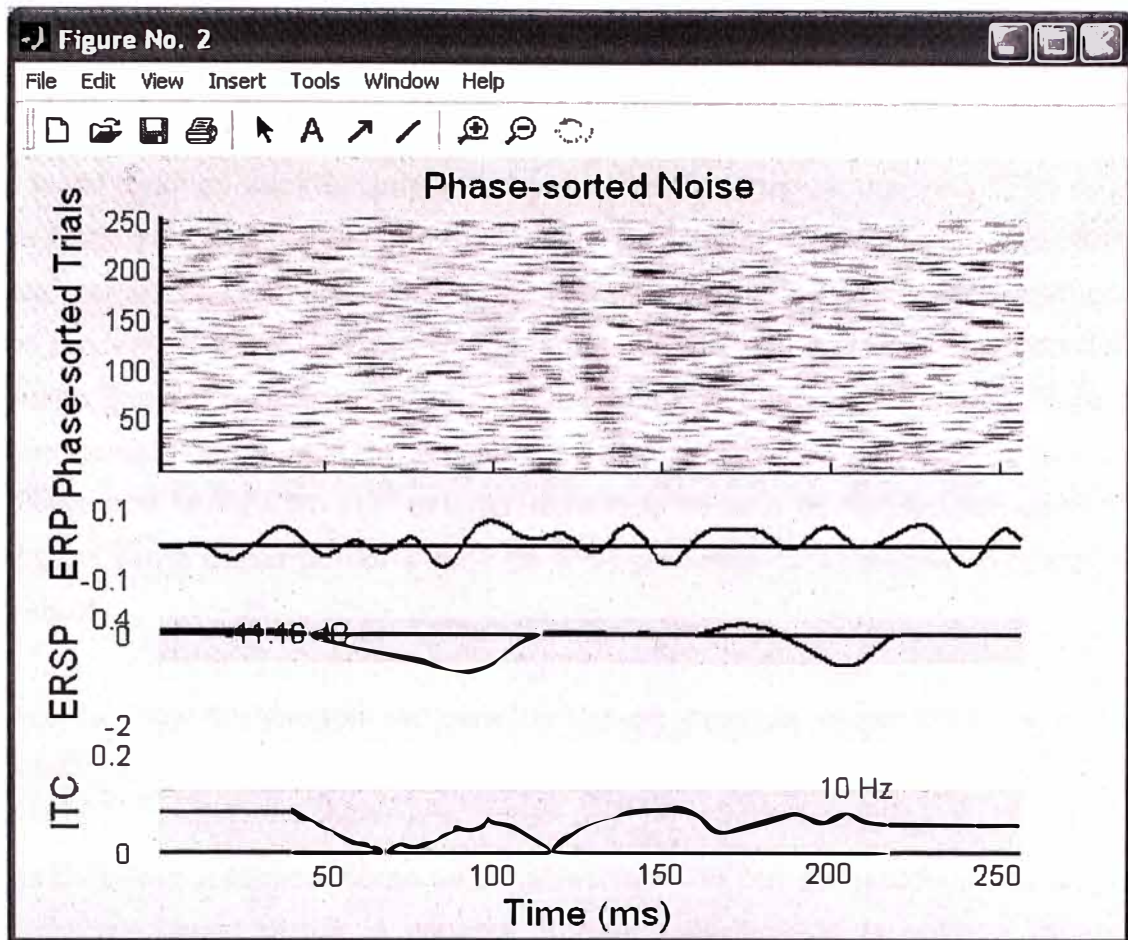


Fig. 3.109 Gráfica de fase ordenada en una imagen ERP

Tomando periodos de ruido blanco (como arriba) y adicionando un estricto bloqueo transitorio de tiempo "ERP-like" a cada prueba, se obtendrá una fase ordenada de una imagen ERP mostrando una sigmoideal, no una línea diagonal de un frente de onda. ¿Cómo puede diferenciarse entre dos interpretaciones de los mismos datos (EEG aleatorios mas ERP versus EEG reiniciado parcialmente en fase)? Para datos de un canal simulado, no hay camino para hacerlo de esta manera, desde que ambos son igualmente caminos válidos de búsqueda en los mismos datos (simulado). Después de todo, la data simulada por ellos mismos no hace retener una impresión de cómo ellos fueron creados aún así como un resto de impresión en la mente del experimentador. Para datos reales, debemos usar una evidencia convergente para influenciar nuestra interpretación hacia una u otras (o a ambas) interpretaciones. El modelo de "reinicio de fase parcial" empieza con los conceptos que las fuentes físicas de la señal EEG (campos locales sincronizados parcialmente) pueden también ser la fuente de contribuidores de características de promedios ERP. Esta suposición puede ser fortalecida o debilitada por el análisis de la distribución espacial en el cerebro de las características de las ERP y de la actividad EEG. Sin embargo, nuevamente aquí, una simple prueba puede no ser

suficiente desde que las fuentes corticales probablemente contribuyen a ambos EEG y ERPs promediados, grabados como un único par de electrodos. Una característica ERP puede resultar desde un reinicio de fase parcial de solamente una de las fuentes EEG, o puede tener muchas contribuciones incluyendo verdaderas excursiones "ERP-like" con latencia fijada y pruebas a través de polaridad. Excursiones "ERP-like" monopolar cuyas latencias varían de acuerdo a las pruebas, y/o reiniciando la fase parcialmente de muchos procesos EEG. Modelos espacio temporal detallado de la colección de datos de una prueba única son requerida a un contexto fuera de estas posibilidades. Para mayor discusión de estas preguntas en el contexto de un conjunto de datos actuales (dataset), ver Makeig et al. (2002). En este estudio, el reinicio de fase de frecuencias alpha y theta fue indicado ser la causa predominante de ERP grabadas (al menos en la posición POZ del cerebro).

En conjunto, hay 5 métodos de clasificación de pruebas disponibles en la función `erpimage()`:

- Clasificación por ordenamiento de variables (**opción por defecto**) – datos de entrada clasificados (*epochs*) por la variable "sortear", clasificando la entrada variable por cada periodo de ingreso de datos.
- Clasificación por el valor ("**valsort**"). Aquí, las pruebas son ordenadas en orden del valor de su significado en una ventana de tiempo dada. Esta opción se utiliza para ordenar por tamaño de ERP (opción no disponible en ventanas interactivas).
- Clasificación por amplitud ("**ampsort**"). Las pruebas son ordenadas en orden de la fase del espectro en una ventana específica de tiempo / frecuencia.
- Clasificación por fase ("**phasesort**"). Las pruebas son clasificadas en orden de fase espectral en una ventana de tiempo/frecuencia específica.
- No clasificado ("**nosort**"). Pruebas de entrada mostrada en el mismo orden que son ingresadas.

Graficando amplitudes espectrales en pruebas individuales y opciones adicionales

Hay variedad de otras opciones `erpimage()` que serán brevemente ilustradas en el siguiente ejemplo. La entrada "Image amps" en la ventana `pop_erpimage()` nos permite graficar amplitudes de la señal (a la frecuencia de interés) en las pruebas individuales, en lugar de las señales originales, para ello active esta opción. La opción "**Plot spectrum (minHz maxHz)**" adiciona una pequeña gráfica de potencia de espectro a la parte derecha y superior de la figura. Ingrese el valor de "**2 50**" para especificar las frecuencias límites para este gráfico.

Retornar el valor del campo "**Epoch-sorting field**" a "**latency**" y retornar "**event type**" a "**etapa1**". Luego ingrese en el campo "**Mark times**" el valor de "**500**" para graficar una marca vertical en **500 ms** (solo es para propósitos ilustrativos). Finalmente ingrese "**-500 1500**" en el campo "**Time limits**" para poder reducir / ampliar en una ventana de tiempo específico, y "**-3 3**" en el campo "**Amplitude limits (dB)**" (Fig. 3.110).

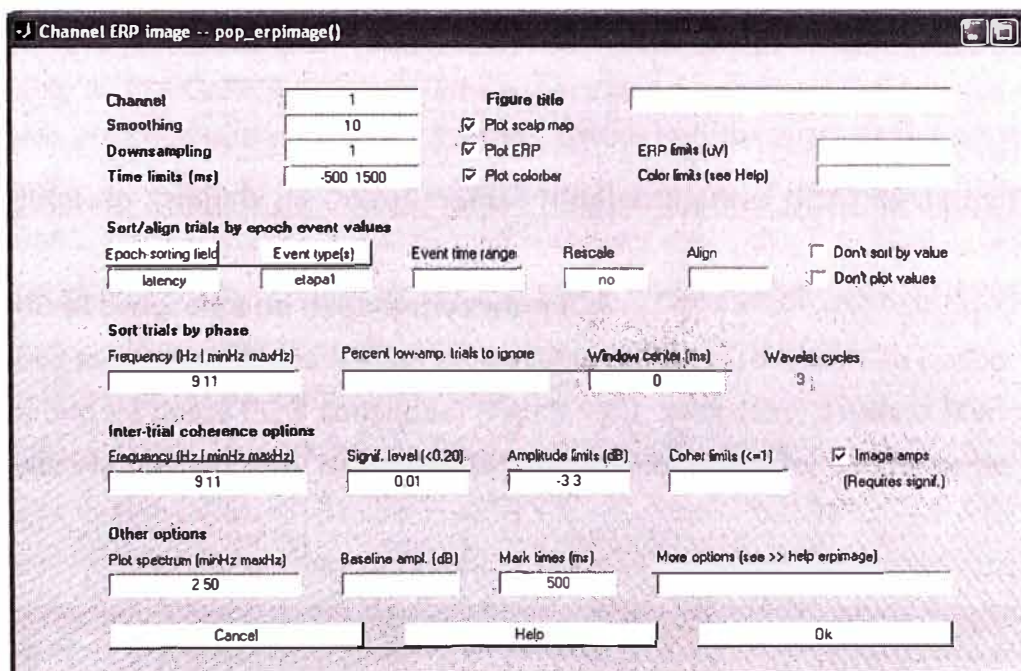


Fig. 3. 110 Forma para graficar amplitudes espectrales

El resultado de los datos ingresados muestra la siguiente ventana (Fig. 3.111).

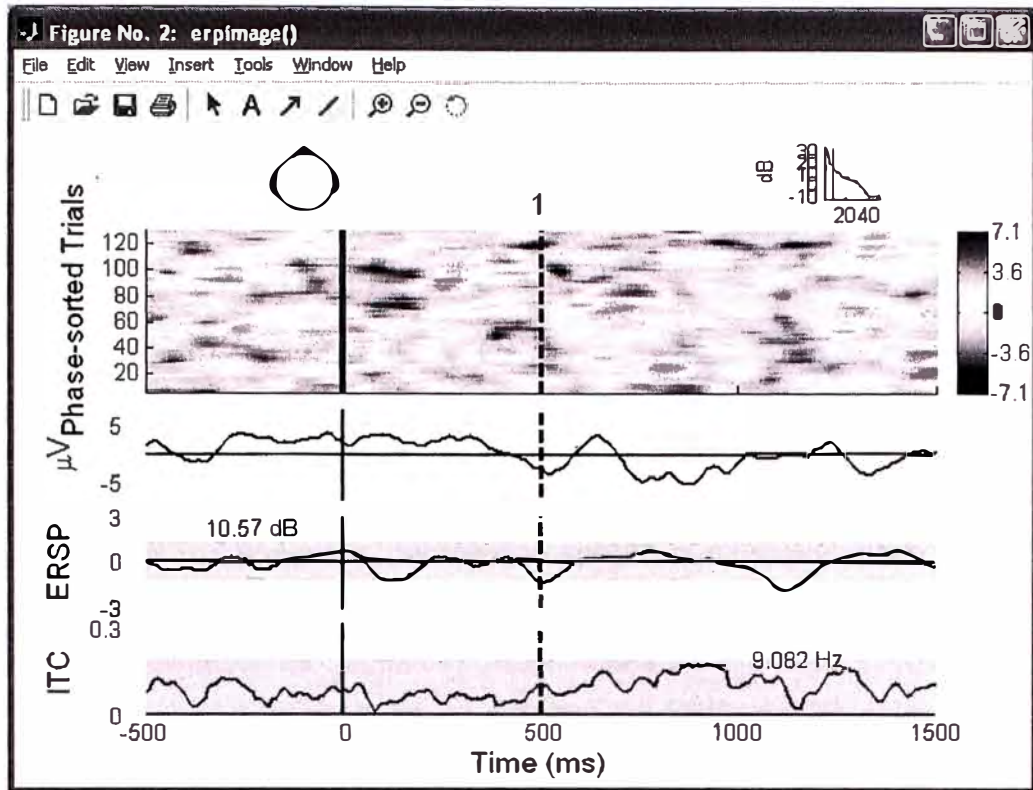


Fig. 3. 111 Gráfica de amplitudes espectrales en pruebas individuales.

Ejecutando Análisis de Componentes Independientes (ICA) de datos EEG

Ejecutando el programa de descomposición ICA

Para calcular los componentes ICA de un conjunto de datos (dataset) de periodos EEG (o de un conjunto de datos EEG continuos) (Fig. 3.112), seleccione **Tools > Run ICA**. Esta opción invoca la función `pop_runica()`. Para probar esta función, simplemente presionar **OK**.

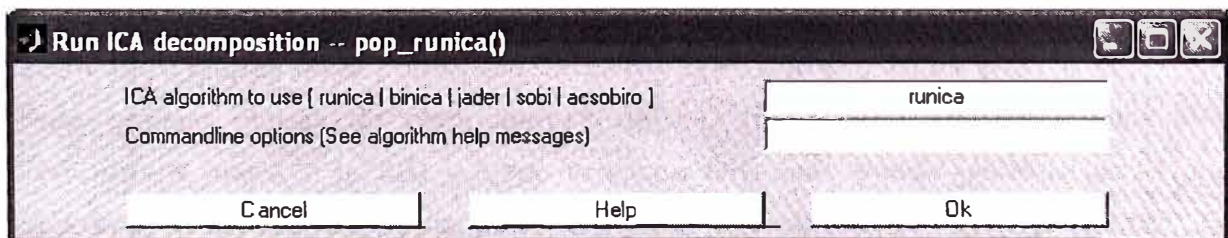


Fig. 3. 112 Ejecutando Análisis de Componentes Independientes (ICA)

Algoritmos ICA: Notar en la ventana de arriba que EEGLAB permite a los usuarios utilizar diferentes algoritmos de descomposición ICA. Solamente "runica", el cual llama a la función `runica()` y "jader", el cual invoca a la función `jader()` son parte de la distribución EEG por defecto.

En general, el trasfondo fisiológico de cualquier diferencia en el resultado o algoritmo diferente (o diferente parámetro elegido en varios de los algoritmos) no ha sido verificado.

Es importante notar como una regla general que buscando N componentes estables (de datos de N -canales) típicamente requieren mas que KN^2 puntos de muestras de datos (en cada canal), donde N^2 es el número de peso en la matriz sin mezcla de datos que ICA esta tratando de aprender y K es un multiplicador. El valor de K incrementa así tan igual como el número de canales se incrementa. Por ejemplo, usando 32 canales, tenemos 30800 puntos de datos, luego $30800/32^2 = 30$ pts/weight points. Sin embargo para encontrar 256 componentes, parece que son 30 puntos, por peso no es suficiente información. En general es importante dar a ICA bastante información como sea posible para un exitoso resultado. Es necesario utilizar bastante información cuando ocurre que la información difiere radicalmente en los estados del EEG, de diferentes ubicaciones de electrodos, o contenido de ruido no estereotípico donde fue concadenado, incrementando el número de mapas de la cabeza asociados con el curso de tiempo independiente y forzando una mixtura de ICA conjuntamente con activaciones desiguales dentro de N componentes de salida. Resulta un mejor trabajo de ICA cuando se da una gran cantidad de información básicamente similares y mayoritariamente clara. Cuando el número de canales (N) es grande ($\gg 32$) puede requerirse una muy grande cantidad de información para encontrar N componentes. Cuando la información disponible no es suficiente, entonces usando la opción "pca" en runica() nos permitirá encontrar pocos componentes de N .

Ejecutando "runica" produce el texto mostrado en el Anexo I (**Reporte en el Matlab a la ejecución del RUNICA en el EEGLab**) en la línea de comandos del Matlab.

Graficando Mapas de la superficie del cerebro en componentes 2-D

Para graficar mapas de componentes de 2-D (Fig. 3.113), seleccionamos **Plot > Component maps > In 2-D** y luego presione **OK**. Esto puede generar varias figuras dependiendo del número de canales y del parámetro del campo "**Plot geometry**". En el gráfico de abajo requerimos para los 13 canales y elegimos la opción '**electrodes**', 'off'.

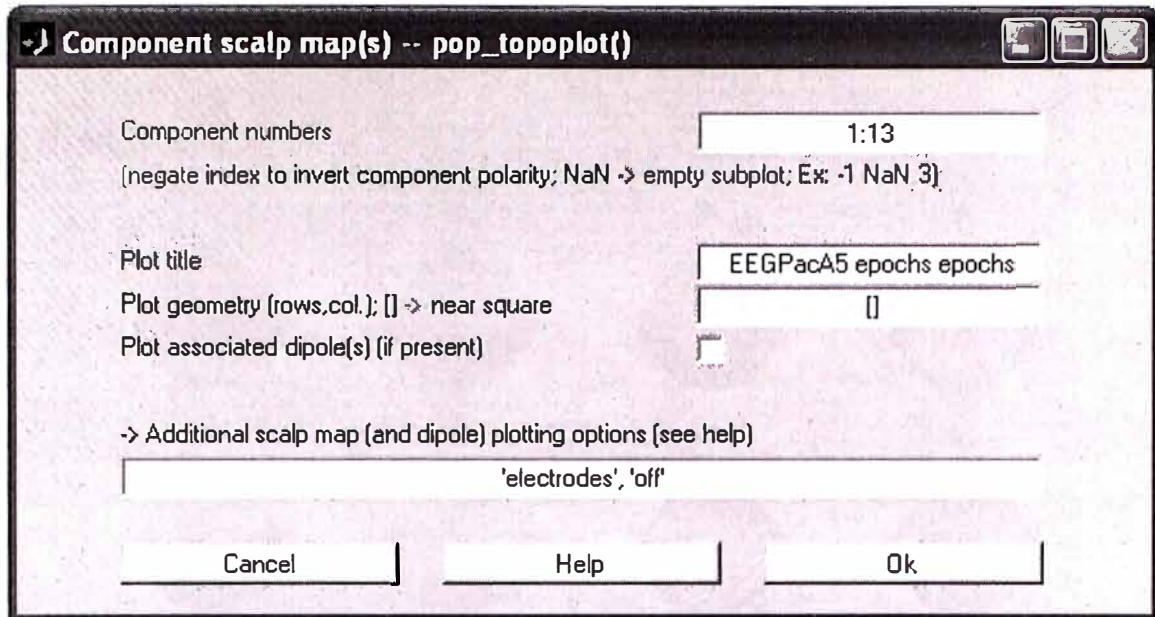


Fig. 3. 113 Formulario para gráficas 2D

Como consecuencia, aparece la siguiente ventana (Fig. 3.114), mostrando las proyecciones de los mapas de la superficie de la cabeza de los componentes seleccionados. Notar que la escala en las siguientes gráficas usan unidades arbitrarias. La escala de la actividad de los componentes del curso del tiempo también usan unidades arbitrarias. Sin embargo, los valores multiplicados de los componentes del cerebro por el componente del tiempo de la actividad están en la misma unidad como los datos.

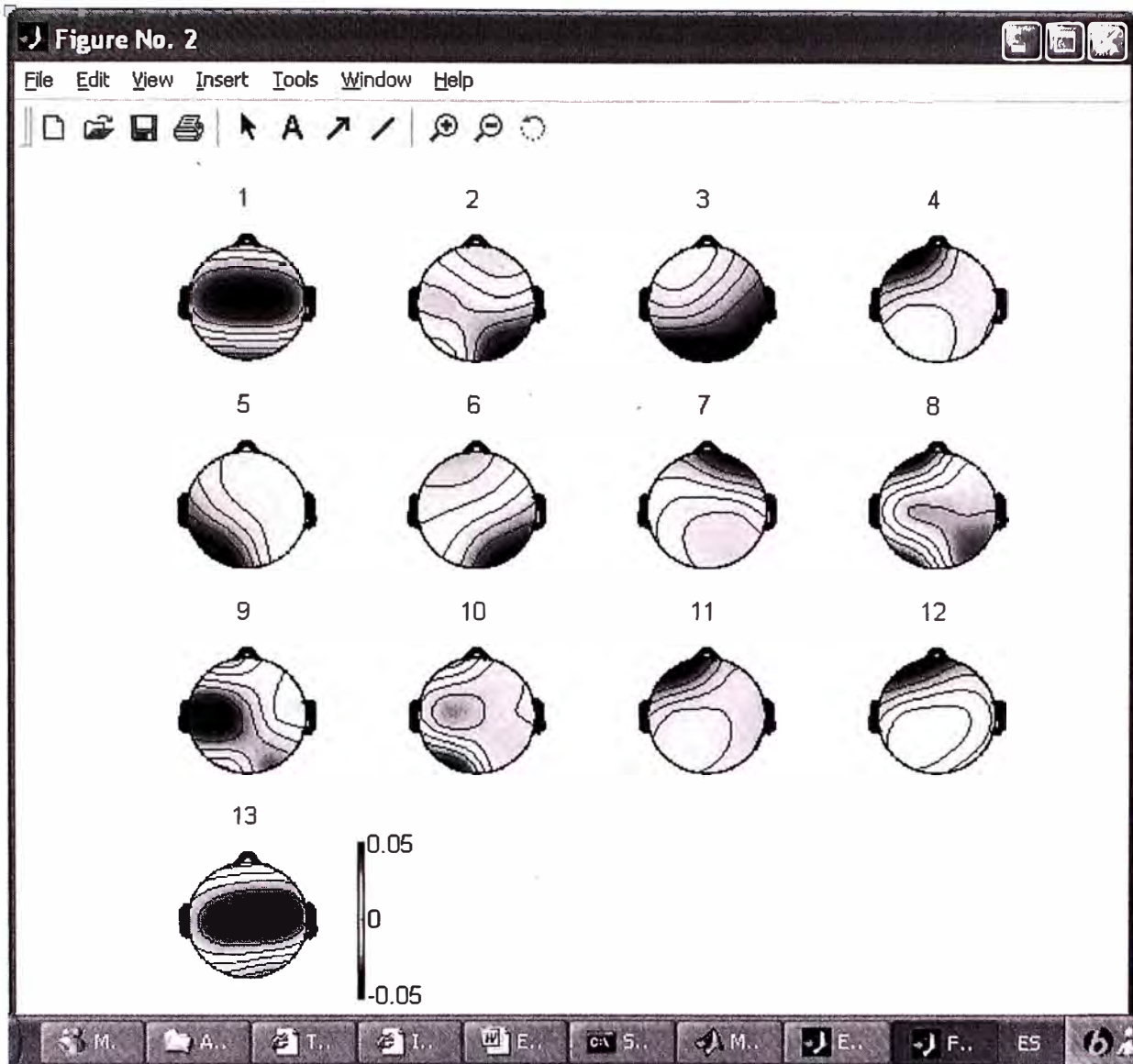


Fig. 3. 114 Gráfica de mapas del cerebro en 2D

Aprendiendo a reconocer los tipos de componentes independientes puede requerir experiencia. El principal criterio para determinar si un componente es **1) cognositivamente relacionado**, **2) una señal perturbadora muscular** ú **3) otros tipos de perturbaciones**, son: primero, el mapa de la superficie cerebral, luego el componente curso de tiempo, luego la actividad del componente de espectro de potencia, y finalmente (dado un "dataset" de periodo de datos de eventos relacionados), la erpimage(). Por ejemplo: un ojo experto podría localizar 2 componentes como un componente de perturbación de los ojos. En la ventana de arriba, seleccionar el mapa numero 2, la cual mostrará una ventana sola, tal como se muestra a continuación (Fig. 3.115).

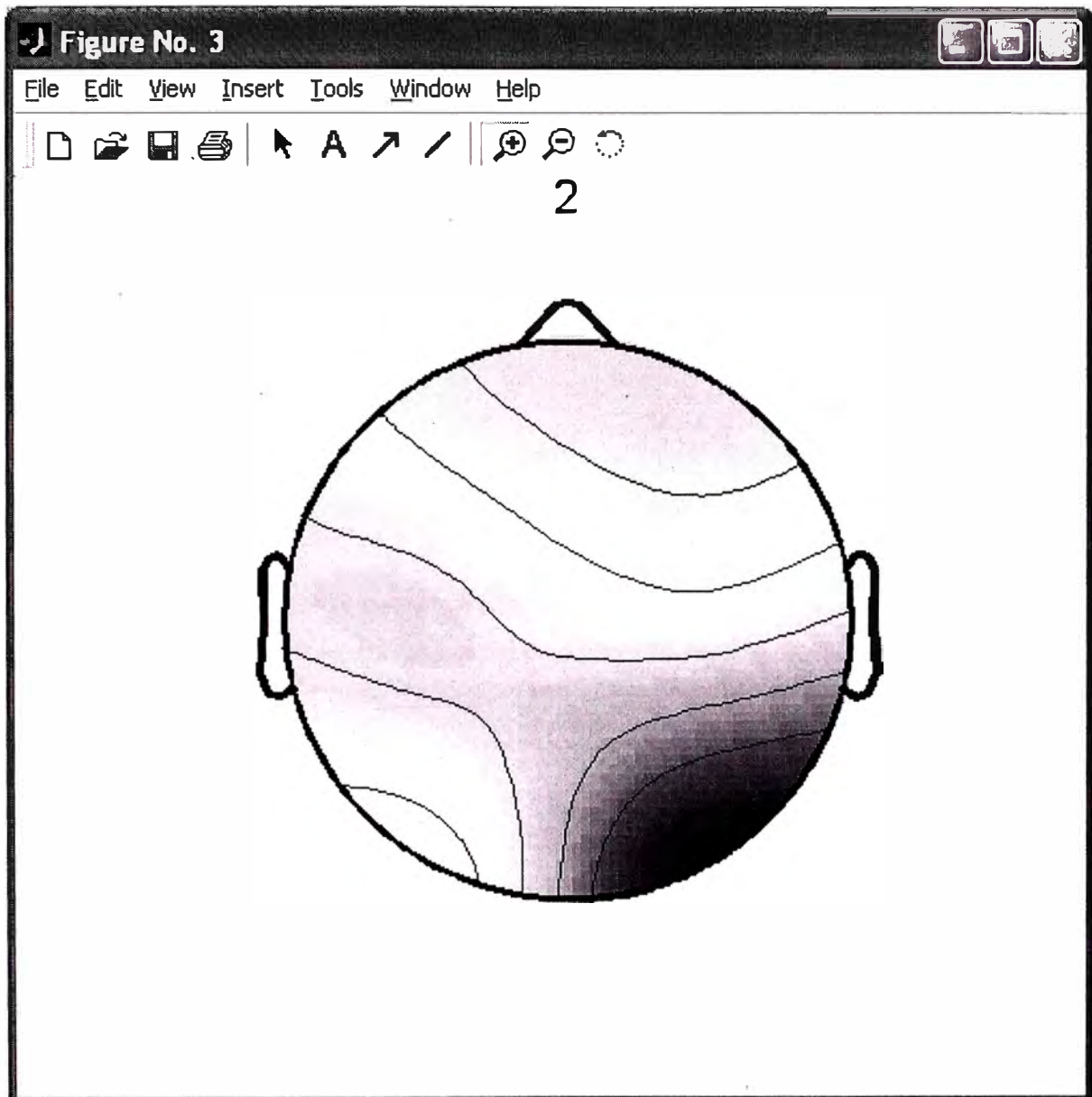


Fig. 3. 115 Gráfica individual con componentes independientes

Graficando componentes de graficas de la cabeza

Usando EEGLAB, podemos graficar una imagen de la cabeza en 3_D en un componente topográfico seleccionando **Plot > Component maps > In 3-D**. Seleccionar uno o más componentes y presionar **OK** (Fig. 3.116).

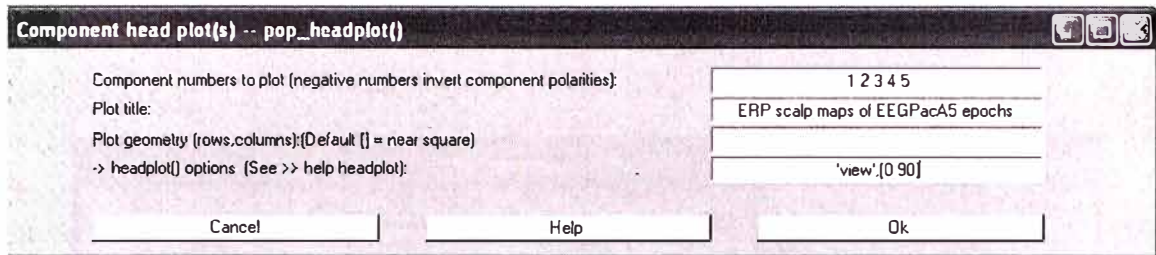


Fig. 3. 116 Formulario para graficar componentes de la cabeza en 3D

La ventana resultante es la que aparece abajo, se puede utilizar la opción de rotar en 3-D de Matlab con el Mouse, o en caso contrario ingrese un ángulo de vista (Fig. 3.117).

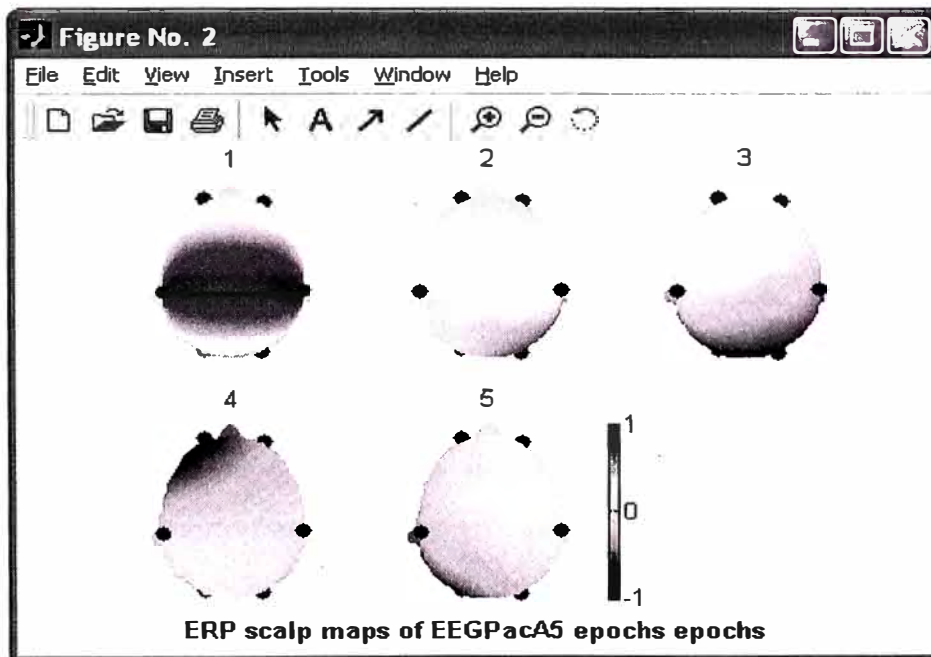


Fig. 3. 117 Gráfica de Mapa de componentes 3D

Estudiando y extrayendo componentes ICA.

Para estudiar las propiedades y nombres de los componentes por rechazo (por ejemplo: identificar componentes a sustraer desde los datos), seleccionar **Tools > Reject data using ICA > Reject components by map**. La diferencia entre las figuras resultantes y la gráfica previa del mapa 2-D de la superficie del cerebro es que uno aquí puede graficar las propiedades de cada componente haciendo click en el botón rectangular que se encuentra encima de cada mapa componente (Fig. 3.118).

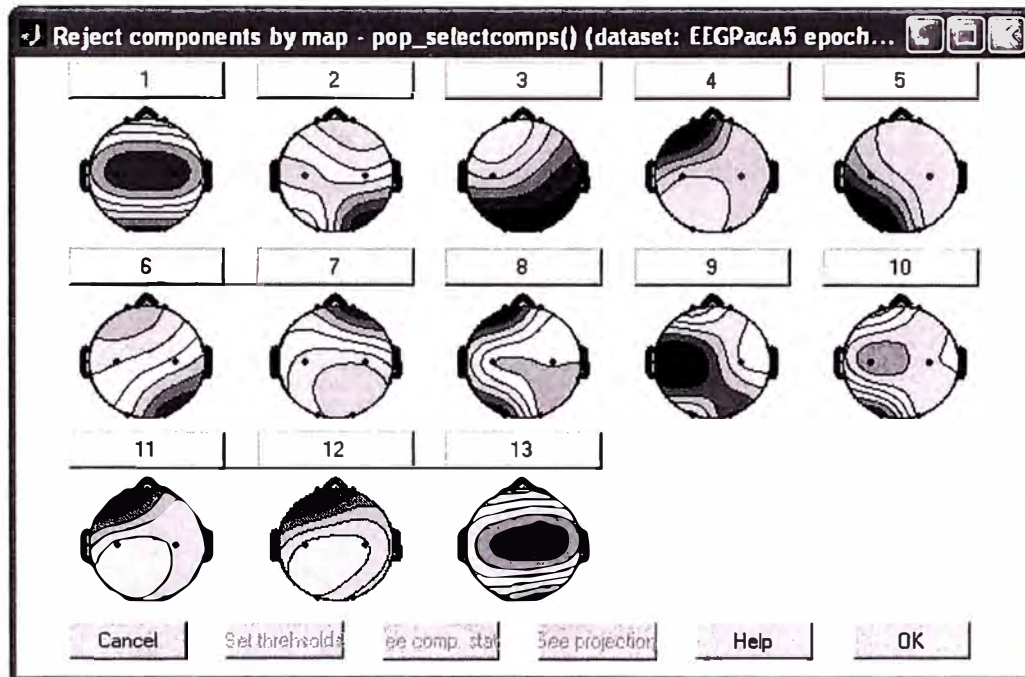
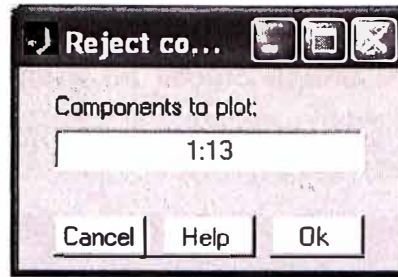


Fig. 3. 118 Gráfica resultado de extraer componentes

Por ejemplo, haciendo click en el botón de **nombre 7**, este componente puede ser identificado como una perturbación del ojo por tres razones: **1)** El decremento suave del espectro EEG (panel de abajo) es típicamente de un disturbio de un ojo; **2)** El mapa del cerebro muestra una fuerte proyección altamente frontal típico de un disturbio de ojos; y **3)** Es posible ver movimientos individuales del ojo en el componente erpimage() (panel arriba-derecho). Las perturbaciones del ojo están siempre presentes en los datos EEG. Por otro lado, el componente 7, muestra una gráfica típica cuando hay presencia de parpadeos del ojo. Hay usualmente, en importantes posiciones de los arreglos de componentes (por ellos tienden a ser grandes) y en sus topografías (si cuentan por lateral movimientos del ojo), muestras parecidas a la gráfica del componente 7. Las gráficas de las propiedades de los componentes pueden ser también accesibles directamente por la selección de la opción **Plot > Component properties** en el menú.

Las componentes perturbadoras son también fáciles para identificar por inspección visual de los componentes en la gráfica de tiempo (menú **Plot > Component activations (scroll)**) (Fig. 3.119).

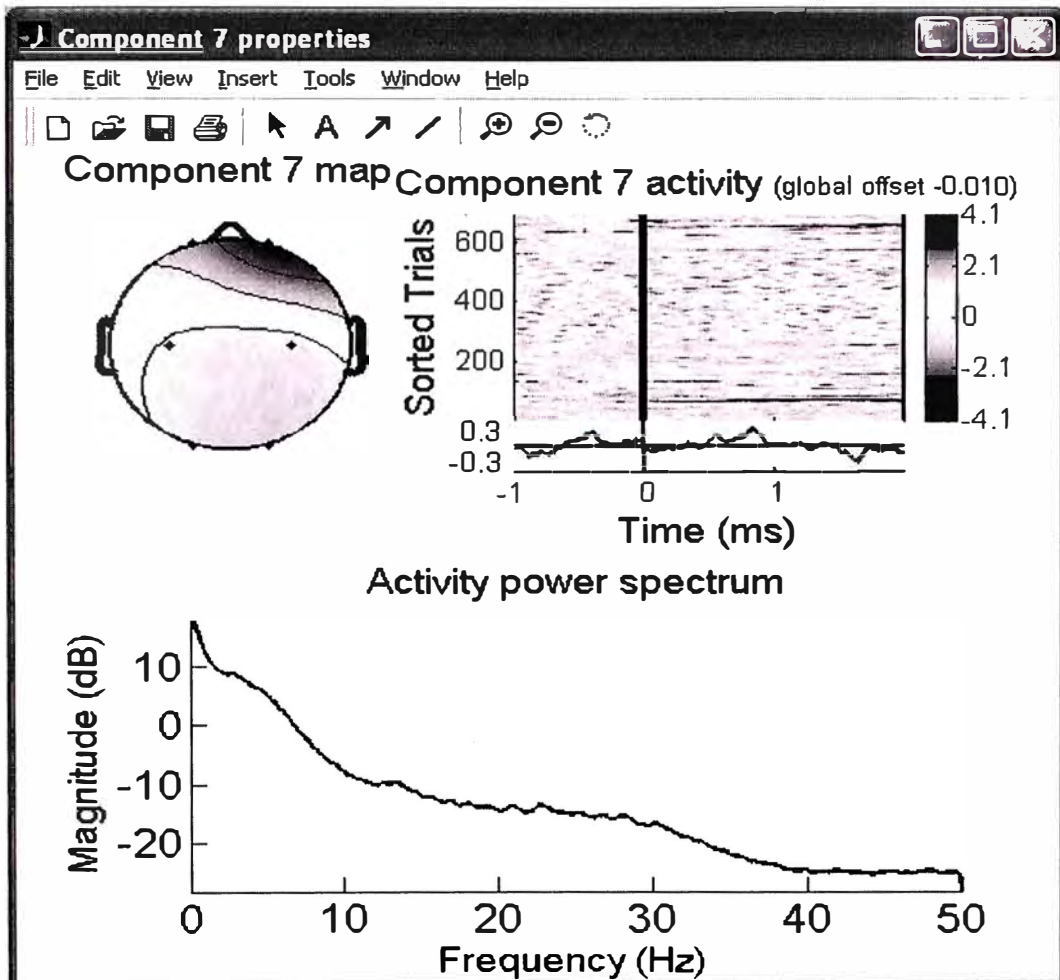


Fig. 3. 119 Gráfica de componente de perturbación del ojo

A continuación se puede ver la ventana del menú **Plot > Component activations (scroll)** (Fig. 3.120).

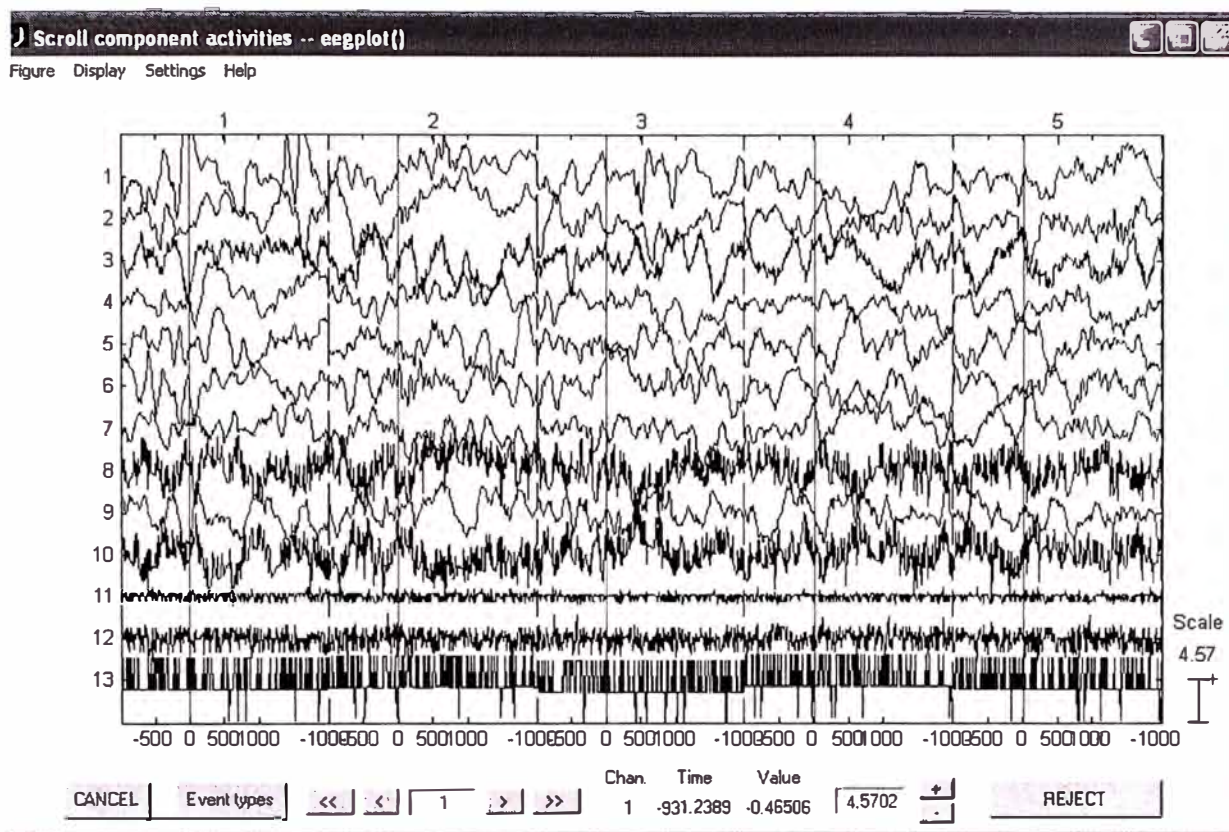


Fig. 3. 120 Señales de componentes extraídos con ICA

Desde que estos componentes cuentan con la actividad de los ojos, se puede sustraerlo de los datos antes de efectuar los análisis y graficas. Si es así, haga click en el botón verde “**Accept**” de abajo y este cambia a un botón rojo con la opción de “**Reject**” (notar que en este punto los componentes son solamente marcados para ser rechazados); para sustraer los componentes marcado, ver la opción de “**Sustraendo componentes ICA de los datos**”, el cual se indicará en el siguiente punto. Ahora presione **OK** para regresar a la ventana de propiedades del componente principal.

Otro ejemplo de disturbio en la descomposición podemos mostrar en el siguiente gráfico (Fig. 3.121) el cual se muestra como un componente de disturbio muscular. Este componente es localizado espacialmente y mostrado con alta potencia en frecuencias altas (20-50 Hz y mayores) como se muestra en la figura.

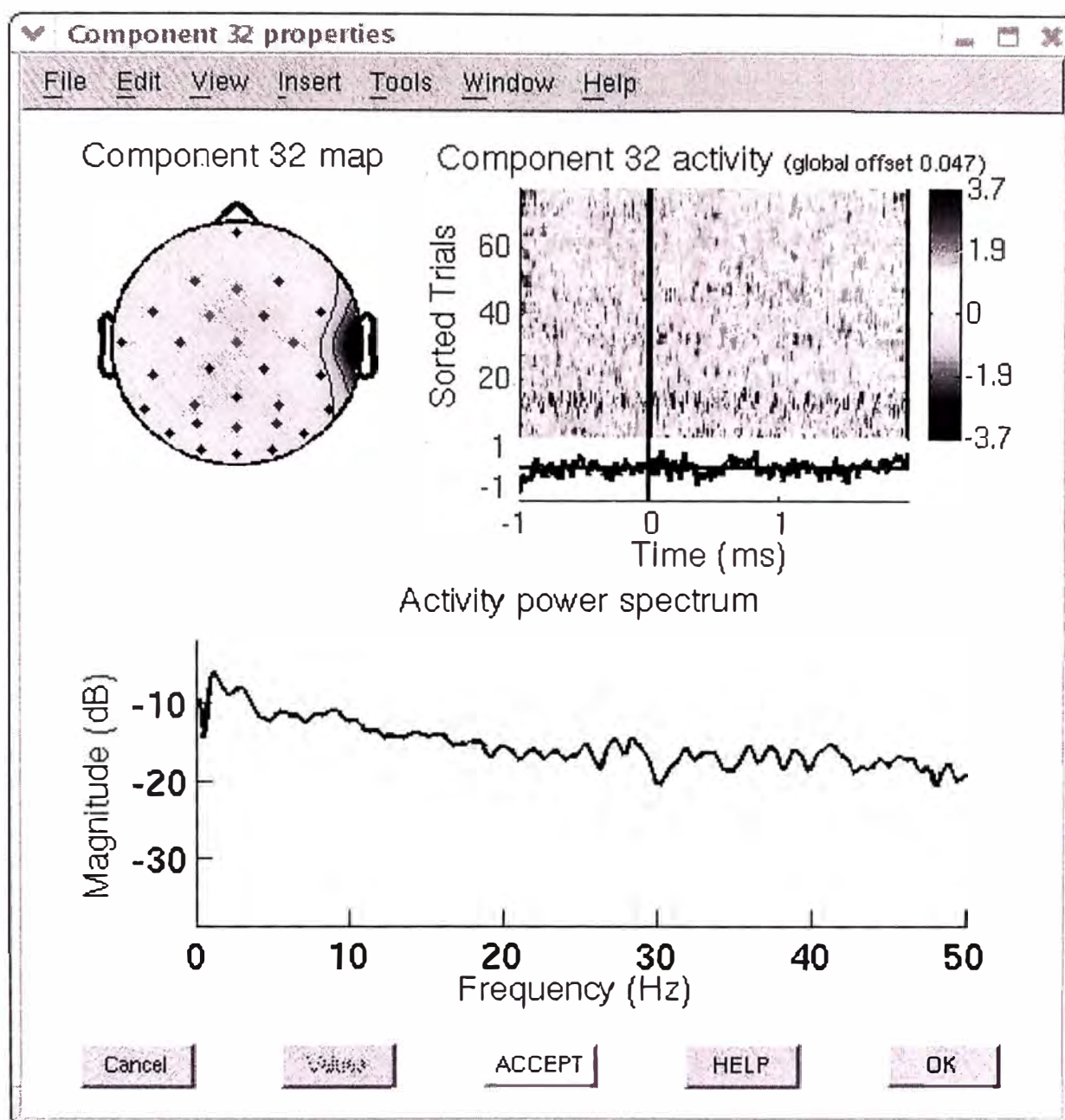


Fig. 3. 121 Componente perturbador de origen muscular

Componentes perturbantes frecuentemente son encontrados (pero no se ha presentado en nuestra prueba) que son canales individuales perturbantes en el cual se pierde la señal o el disturbio es por una línea ruidosa. En la gráfica mostrada a continuación (Fig. 3.122) podemos observar la perturbación que ocasiona una línea ruidosa por la presencia de 60 Hz.

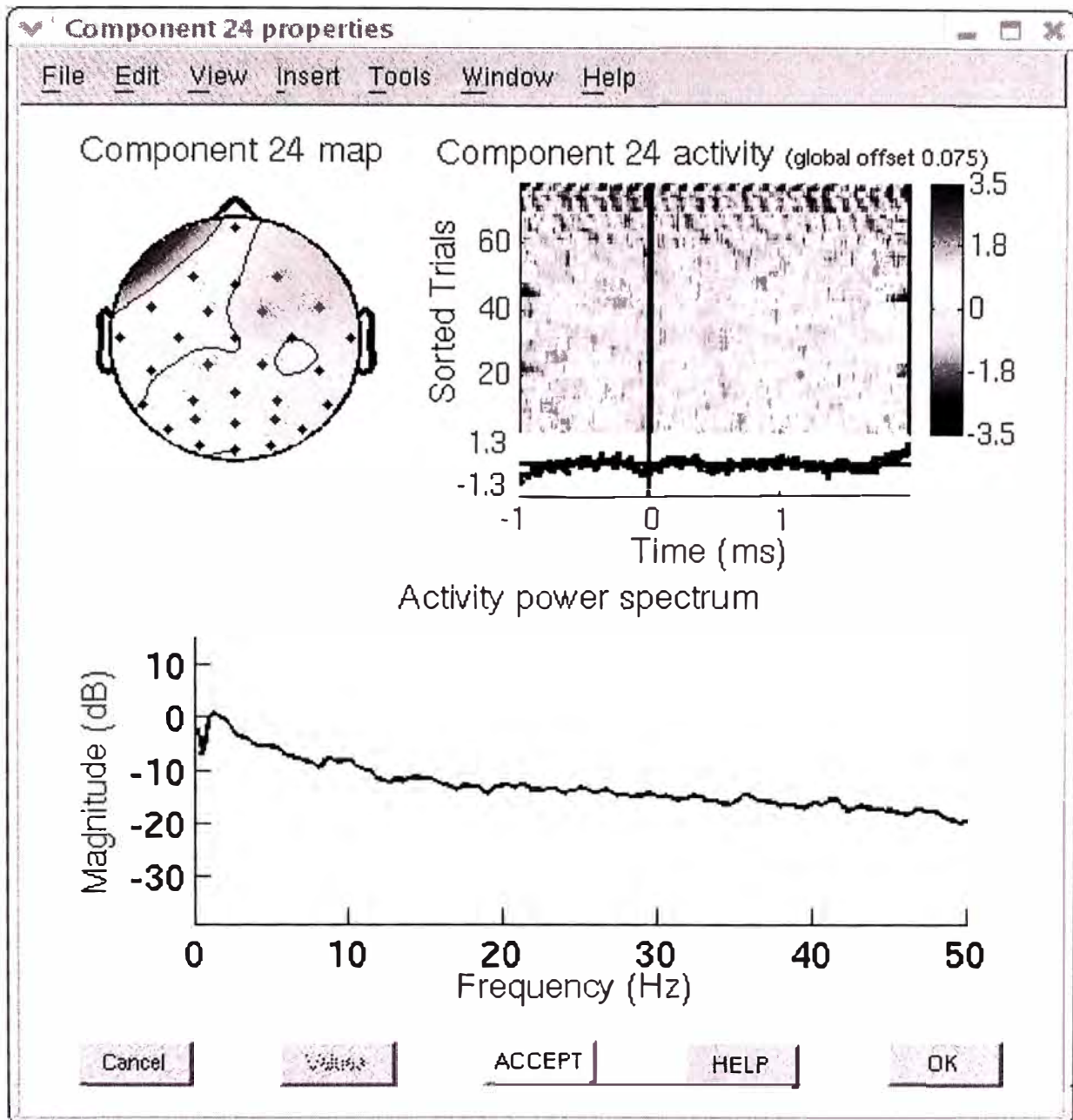


Fig. 3. 122 Perturbación por presencia de 60 Hz.

Hay también muchos otros componentes que son relacionados al cerebro. Los principales criterios para el reconocimiento de componentes relacionados al cerebro son aquellos que tienen: **1)** mapas de dipolos o parecidos del cerebro, **2)** Picos espectrales a las frecuencias EEG típicas y **3)** gráficas de imágenes ERP regulares. La componente de abajo tiene un pico de banda alpha predominante cerca de los 10 Hz y una distribución compatible de mapas del cerebro con una fuente de la corteza del cerebro de la parte occipital izquierda. Cuando se localiza una fuente ICA usando localización de fuente de dipolos simples o pares de dipolos (Fig. 3.123).

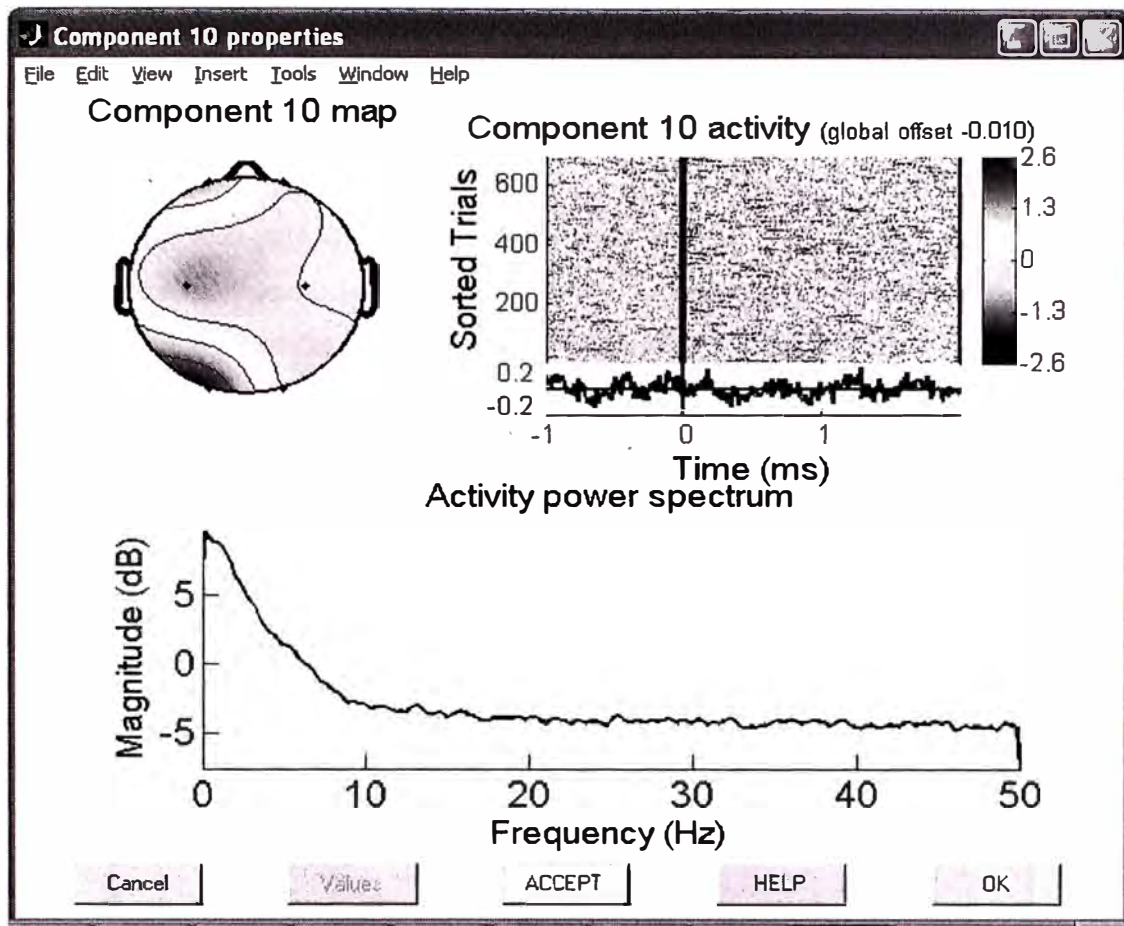


Fig. 3. 123 Componentes bipolares y picos espectrales

¿Que pasa si los componentes parecen ser mitad perturbaciones y mitad relacionados al cerebro? En este caso, podemos ignorarlos o intentar ejecutar una nueva descomposición ICA en un subconjunto de datos mas limpios o usando otros parámetros de ICA. Como una regla de oro, hemos aprendido que removiendo los periodos con perturbaciones conteniendo uno de una clase de perturbación es muy útil par obtener unos componentes ICA limpios.

Nota importante: Una óptima estrategia es: **1)** ejecutar ICA, **2)** rechazar periodos errados. En algunos casos, no vacilar en remover más del 10% de los datos de prueba, aún así se conjunto de datos EEG relativamente limpios. Hemos aprendido que es frecuentemente mejor ejecutar primero la composición ICA en una ventana de muy corto tiempo. **3)** luego ejecute ICA una segunda vez en los datos ya reducidos. **4)** aplicar el peso resultante ICA a los mismos datos o a periodos mayores graficados desde la misma data original (continuos o en conjunto de periodos), por ejemplo copiar pesos de ICA e información esférica desde datos 1 o 2. Primero, invocar el menú **Edit > Dataset info o datos 2.**

Luego ingrese “ALLEEG(1).icaweights” en la “ICA weight array ...” casillero de edición, “ALLEEG(1).icasphere” en la “ICA sphere array ...” casillero de edición, y presione “OK”.

Sustraendo componentes ICA en la data

No siempre sustraemos todos los componentes desde los datos porque se estudia la actividad de los componentes individuales. Sin embargo, cuando deseamos remover componentes, usaremos el menú **Tools > Remove components**, el cual invoca la función `pop_subcomp()`. El número de componentes presentes por defecto en la ventana resultante son aquellos marcados por rechazo en la ventana previa de rechazo de componentes **Tools > Reject using ICA > Reject components by map**, utilizando la opción del boton “Accept/Reject”. Ingrese el número de componentes que desee rechazar y presione “OK” (Fig. 3.124 y Fig. 3.125 y Fig. 3.126).

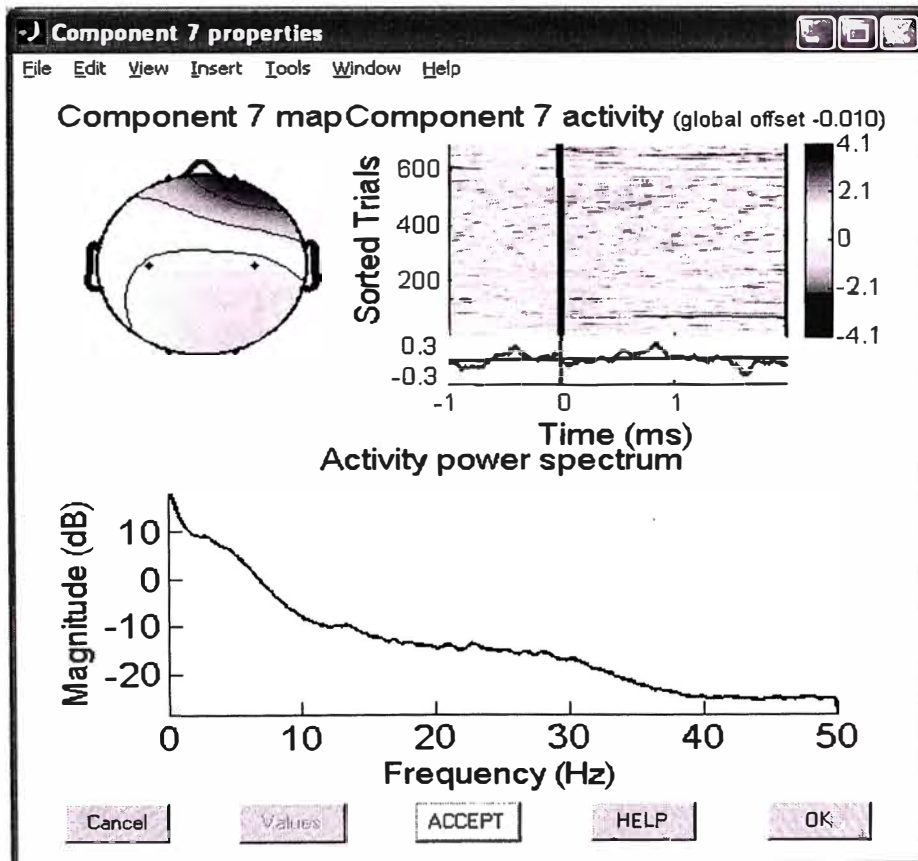


Fig. 3. 124 Gráfica con componentes rechazados

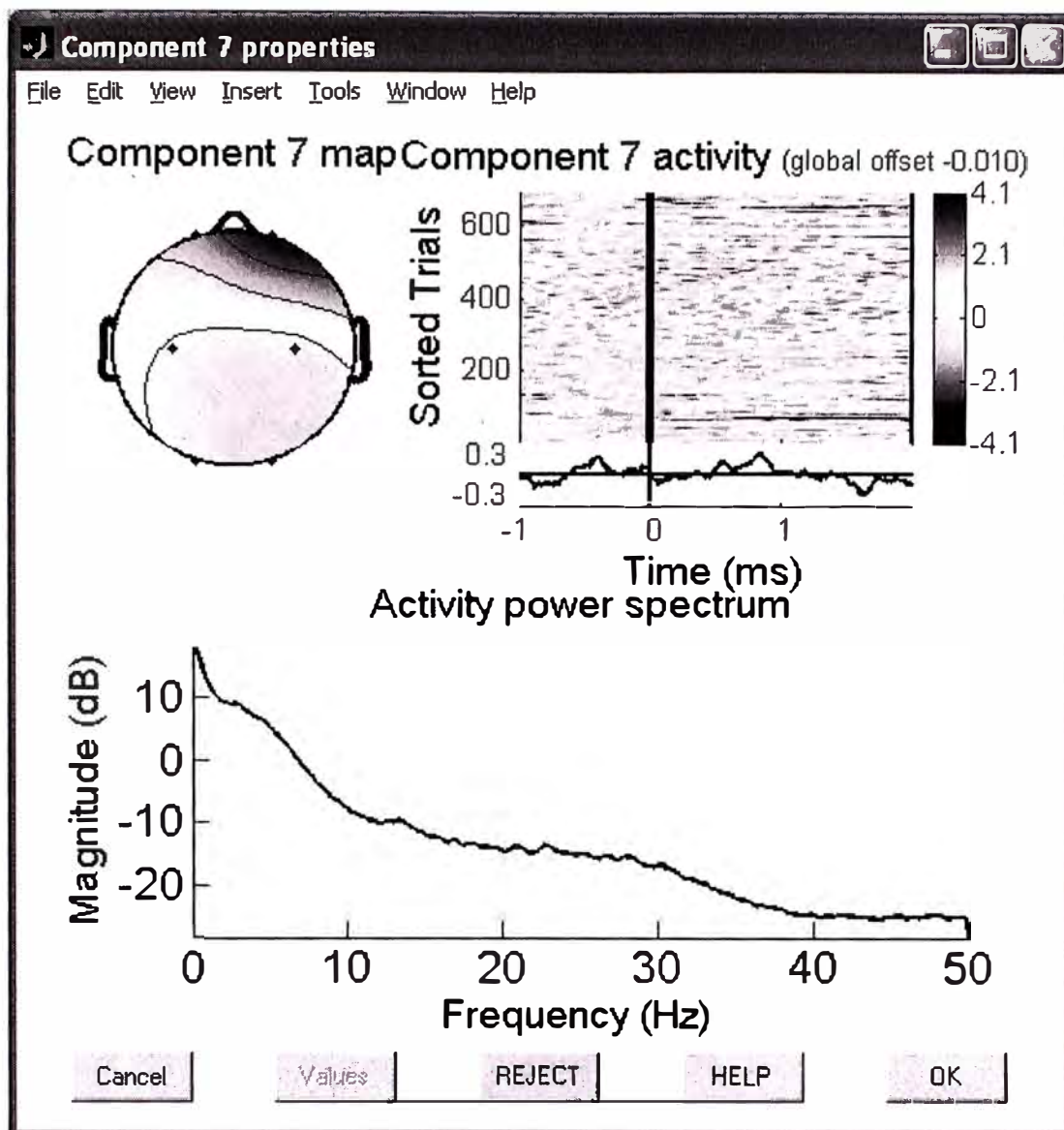


Fig. 3. 125 Componentes ICA sustraídos

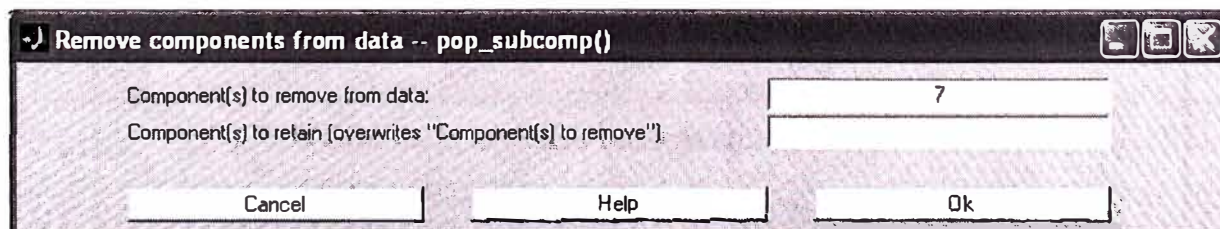


Fig. 3. 126 Removiendo el componente 7

Una ventana se despliega, graficando el canal ERP previo (en azul) y luego (en rojo) la sustracción de componentes y preguntando si esta de acuerdo con la nueva data ERP (Fig. 3.127).

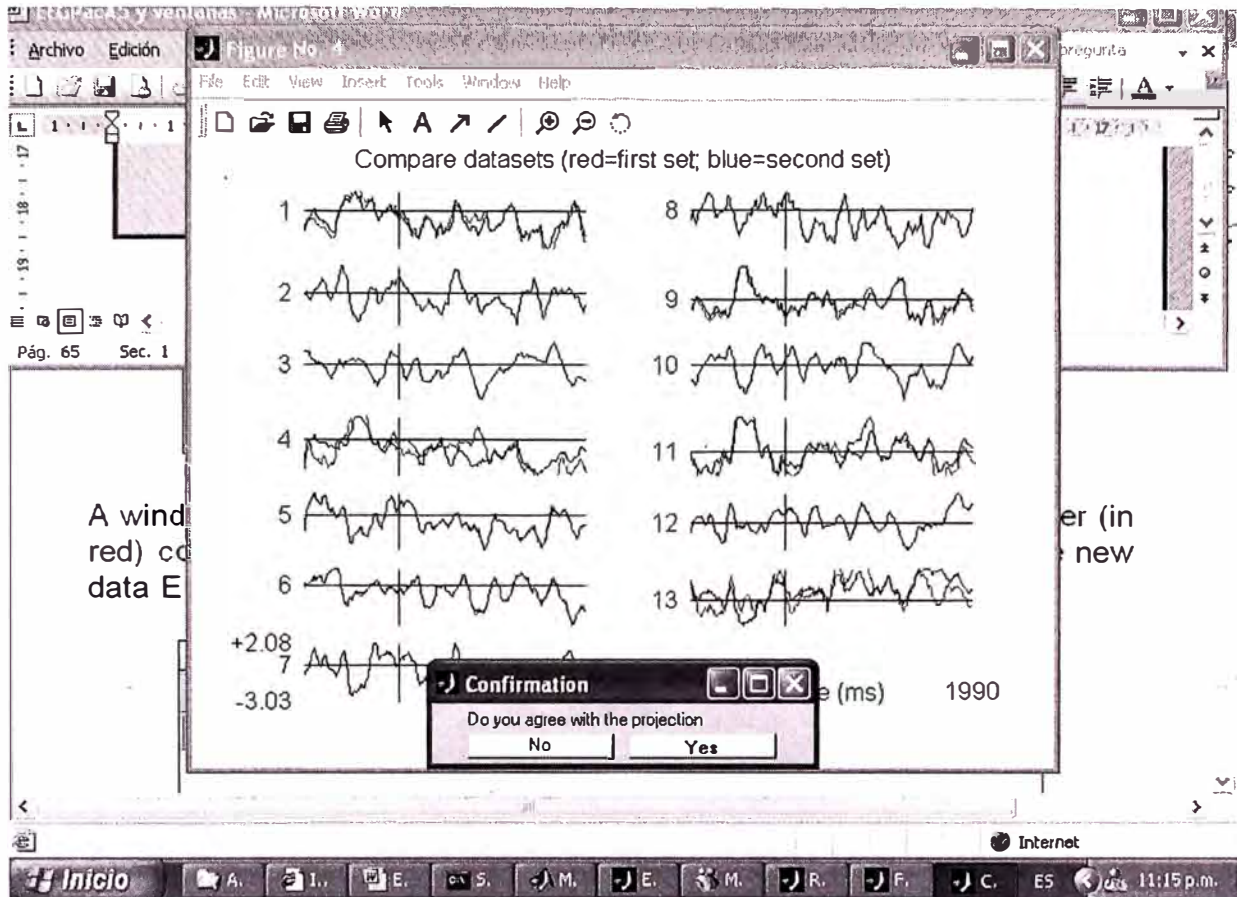


Fig. 3. 127 Gráfica que muestra el canal ERP previo y luego a la sustracción.

Presione **"Yes"**. Una última ventana se desplegará preguntando si desea renombrar el nuevo dataset. De un nombre y presione nuevamente **"OK"** (Fig. 3.128).

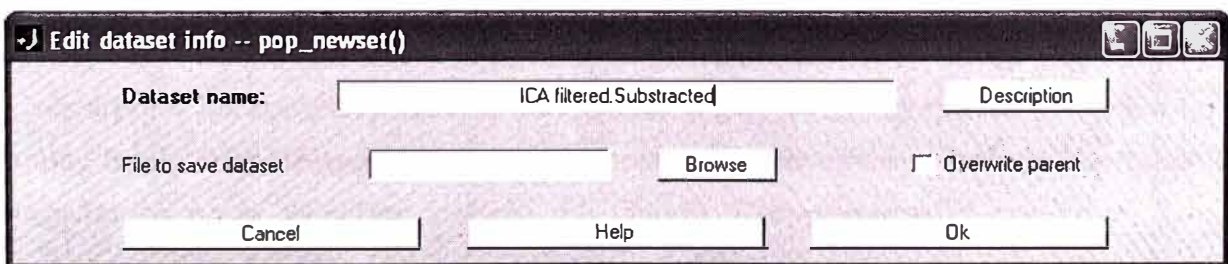


Fig. 3. 128 Grabando la nueva dataset con el componente removido

Luego de esto obtenemos los nuevos datos cuya grafica ICA se observa en el siguiente dibujo (Fig. 3.129).

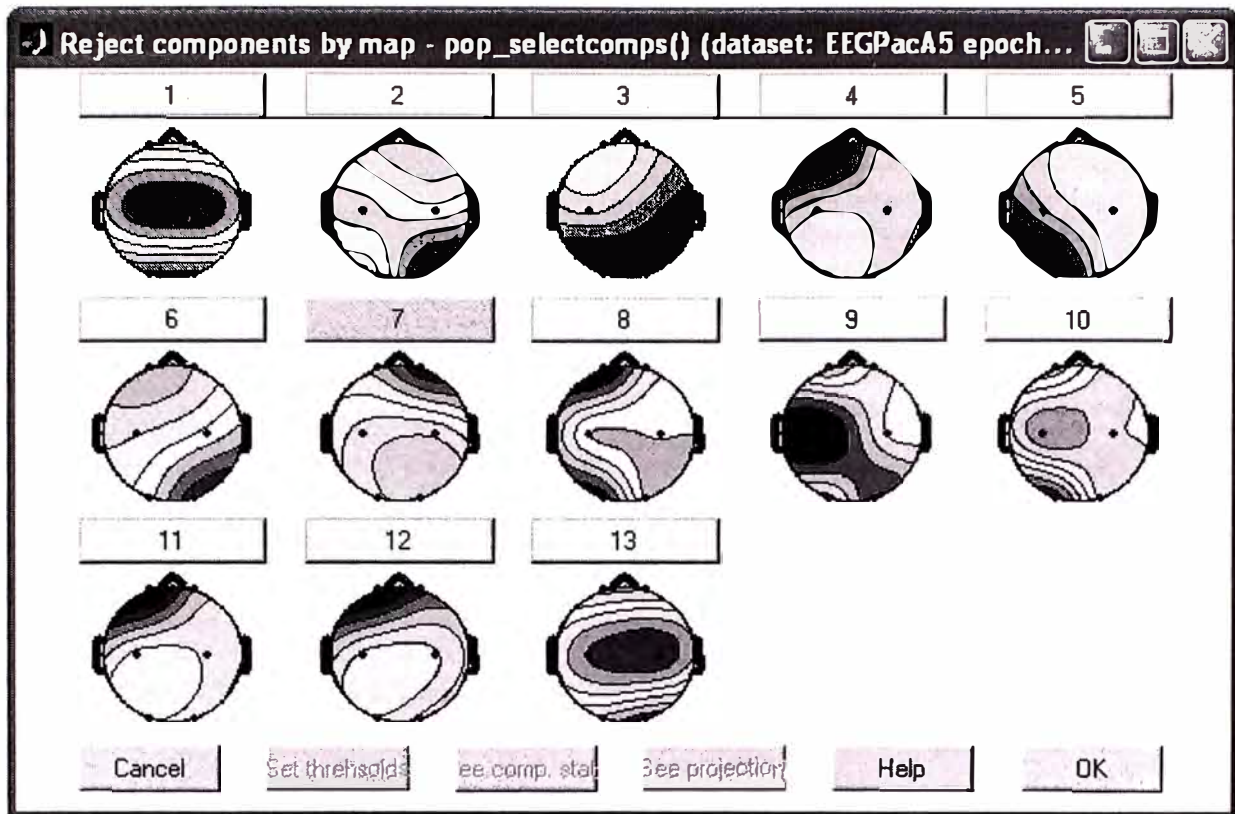


Fig. 3. 129 Gráfica resultante de la remoción de la componente 7

Graficando componentes espectrales y mapas

Es de interés ver que componentes contribuyen más fuertemente a las frecuencias en los datos. Para hacerlo, seleccione **Plot > Component spectra and maps**. La primera entrada a considerar es el rango de tiempo del periodo, en adelante es el porcentaje de datos aleatorios de las muestras (pequeños porcentajes aceleran la velocidad de análisis). En nuestro análisis elegiremos el valor de 100 (todos los datos) y visualizaremos los componentes que contribuyen a los 10 Hz, ingresando este valor en el campo **"Scalp map frequency"**. Luego simplemente hacemos el barrido de todos los componentes en el valor por defecto del campo **"Components to consider"**. Luego presione **OK** (Fig. 3.130).

Component spectra and maps -- pop_spectopo()

Epoch time range to analyze (min_ms max_ms): -1000 1990

Frequency (Hz) to analyze: 10

Electrode number to analyze ([]=elec with max power; 0=whole scalp): 0

Percent data to sample (1 to 100): 100

Components to include in the analysis: 1:13

Number of largest-contributing components to map: 5

Else, map only these component numbers:

[Checked] Compute comp spectra: [Unchecked] (data-comp) spectra:

Plotting frequency range ([min max] Hz): 2 25

Spectral and scalp map options (see topoplot): 'electrodes','off'

Cancel Help Ok

Fig. 3. 130 Formulario para graficar componentes espectrales y mapas

Luego de presionar el botón de OK, se presenta la siguiente figura, en la cual, graficamos el espectro de cada uno de los componentes. Una mayor estrategia de precisión (por razones técnicas) es graficar las señales de los datos menos la actividad componente y estimación del decrecimiento en potencia en comparación a la señal original en un canal (es posible hacerlo en todos los canales pero requiere cómputo del espectro de la proyección de cada componente en cada canal el cual requiere cálculo computable intenso). Para hacer esto, regrese a la ventana anterior, elija explícitamente graficar la contribución de los componentes en el canal 7, donde la potencia parece ser máxima a 10 Hz usando el campo “**Electrode number to analyze ..:**”, el campo “[checked* **compute component spectra..**” sin seleccionar, definir a 100 como antes. Finalmente mostraremos 6 mapas componentes en vez de 5 (valor por defecto) (notar que todos los componentes espectrales serán mostrados) y definiremos la máxima frecuencia a graficar (30 Hz) usando “**Plotting frequency range**” opción que se encuentra en panel de abajo, luego presione **OK** (Fig. 3.131, Fig. 3.132 y Fig. 3.133).

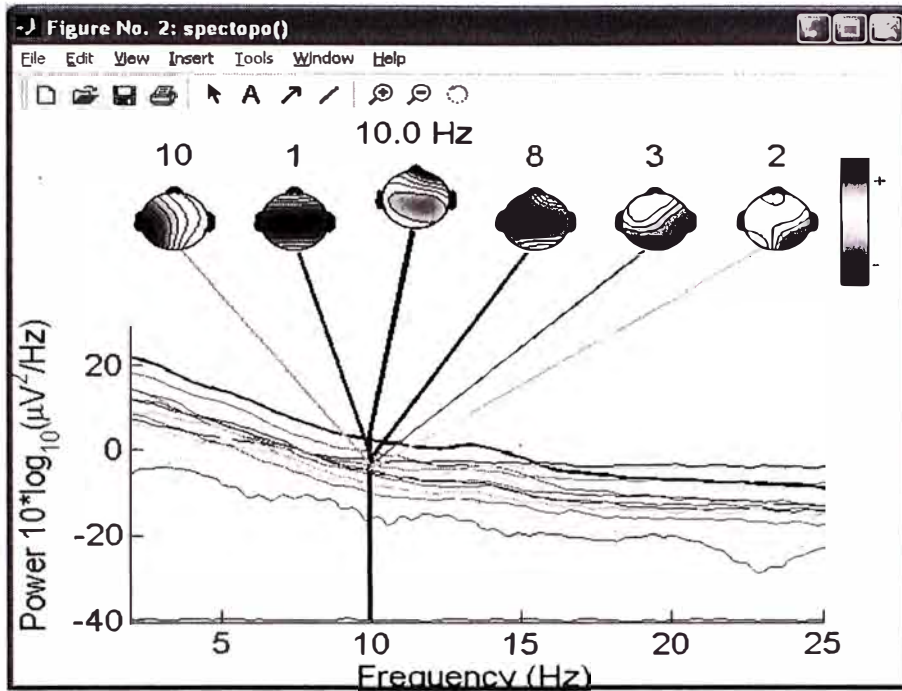


Fig. 3. 131 Gráfica de componentes espectrales y mapas

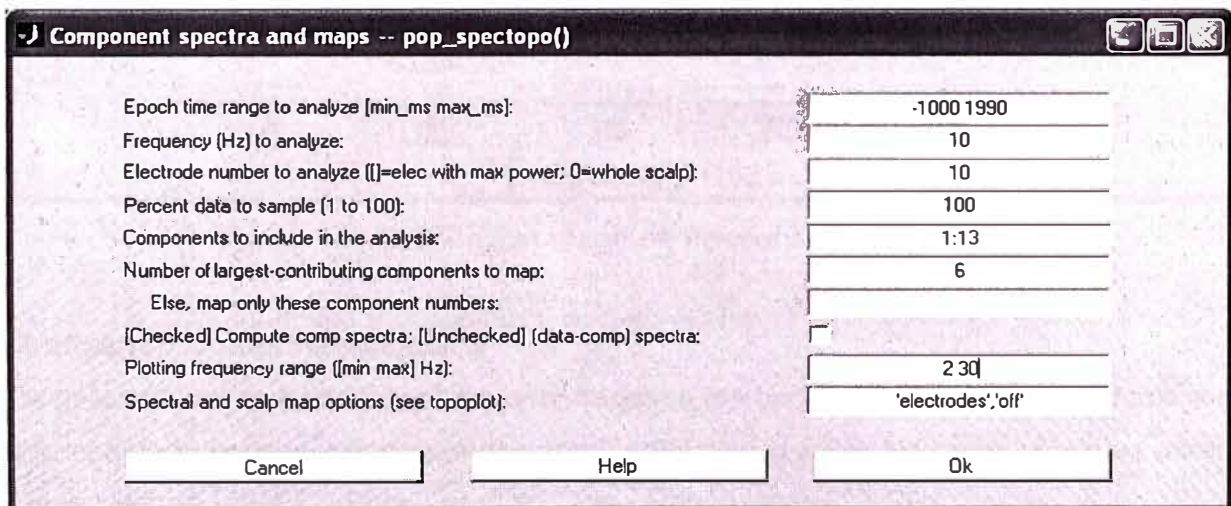


Fig. 3. 132 Seleccionando gráfica con rango de frecuencia min 2, max 30

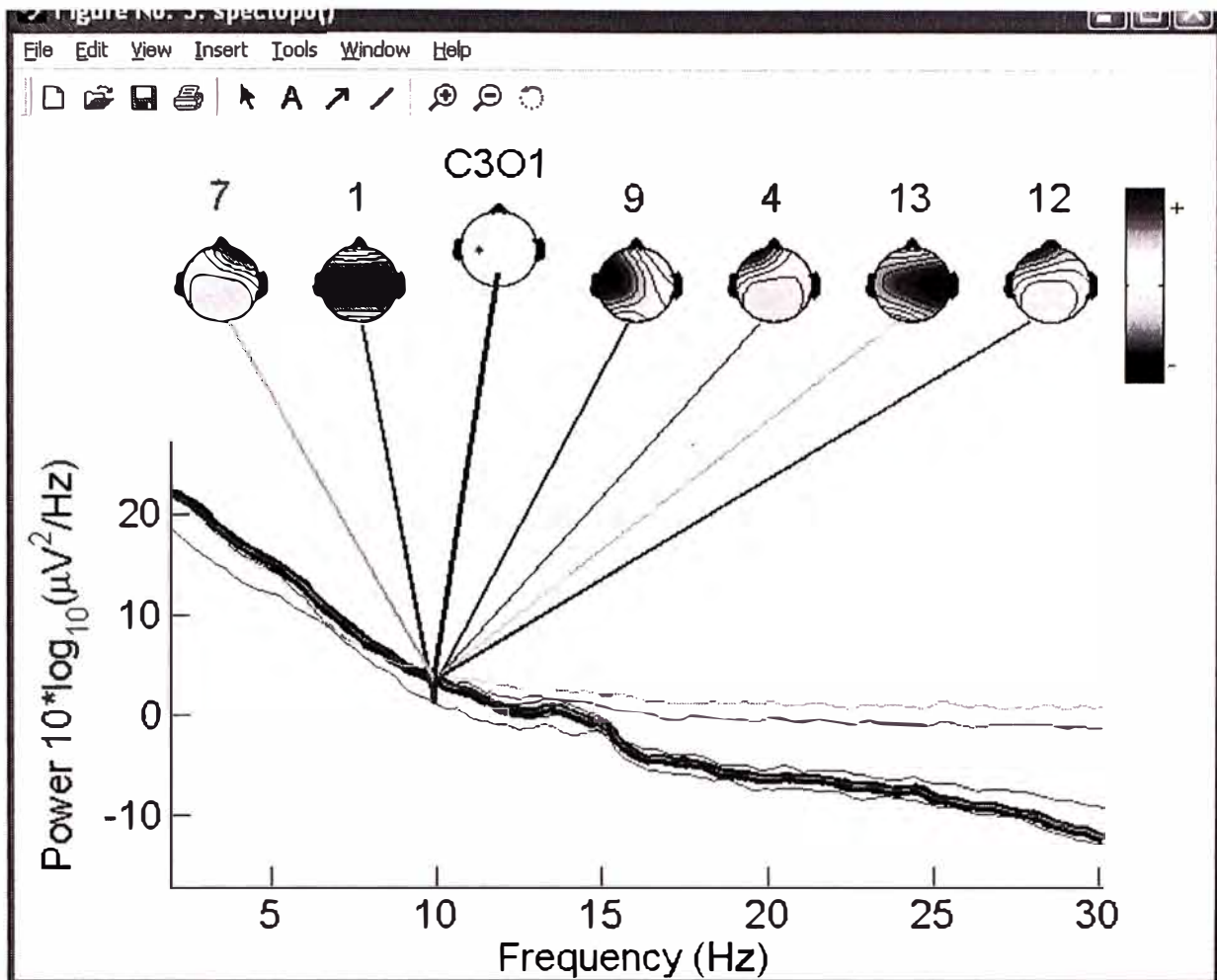


Fig. 3. 133 Gráfica con rango de frecuencia min 20, max 30.

Graficando Componentes ERPs

Después de ver que componentes contribuyen a las bandas de frecuencias de interés, es interesante mirar que componentes contribuyen más al ERP. Un primer paso es mirar los componentes ERPs. Para graficar los componentes ERPs, seleccionar **Plot > Component ERPs > In rectangular array**, el cual invoca a la función `pop_plotdata()`. Luego presione **OK** (Fig. 3.134).

Fig. 3. 134 Formulario invocado con la función `pop_plotdata()`

La ventana que aparece como resultado, se muestra mas abajo en el cual se visualiza el promedio ERP para los 13 componentes de nuestro análisis (Fig. 3.135).

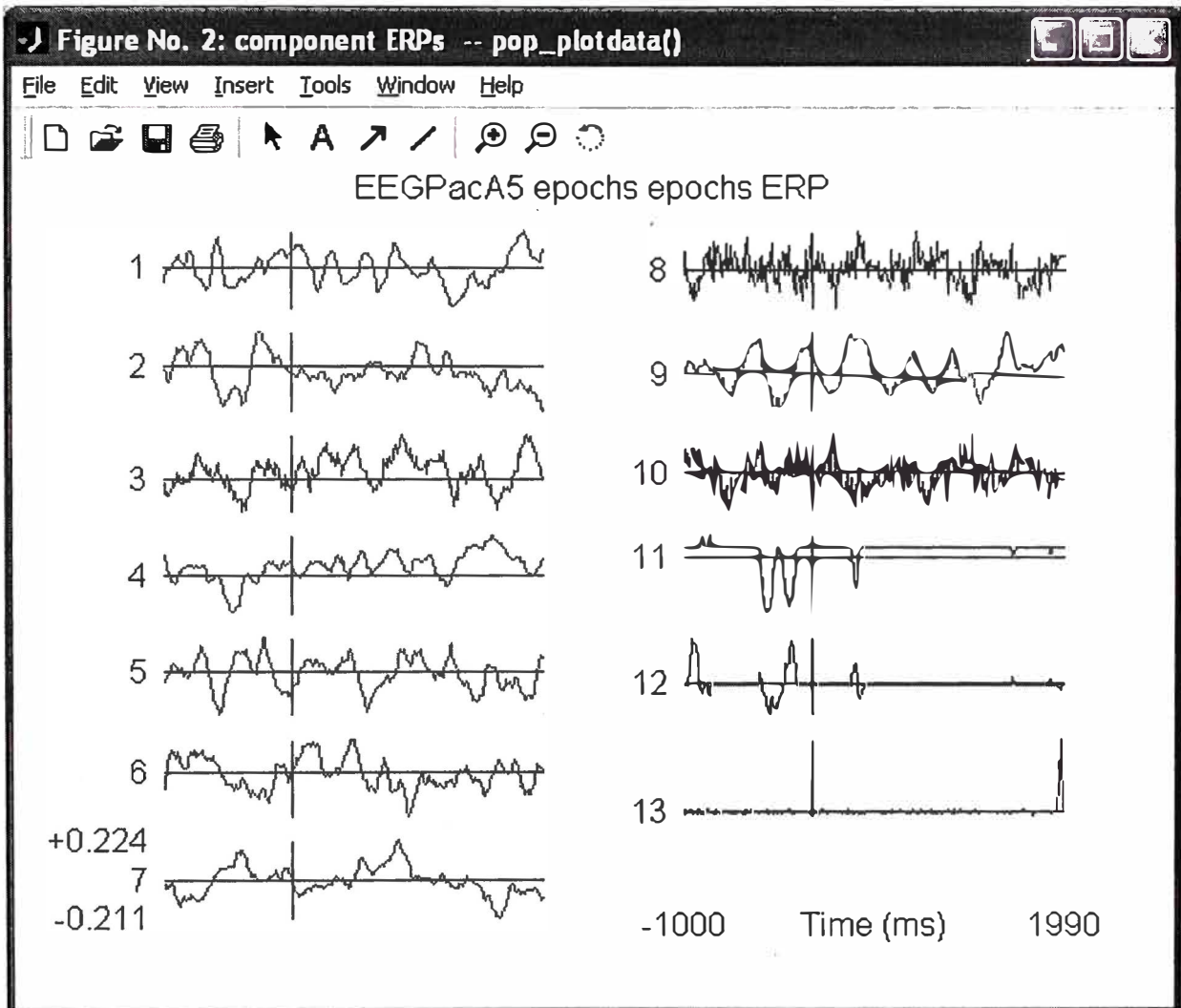


Fig. 3. 135 Grafica de componentes ERP de los 13 canales

Haciendo click en un componente de la gráfica de arriba, aparece una nueva gráfica (Fig. 3.136).

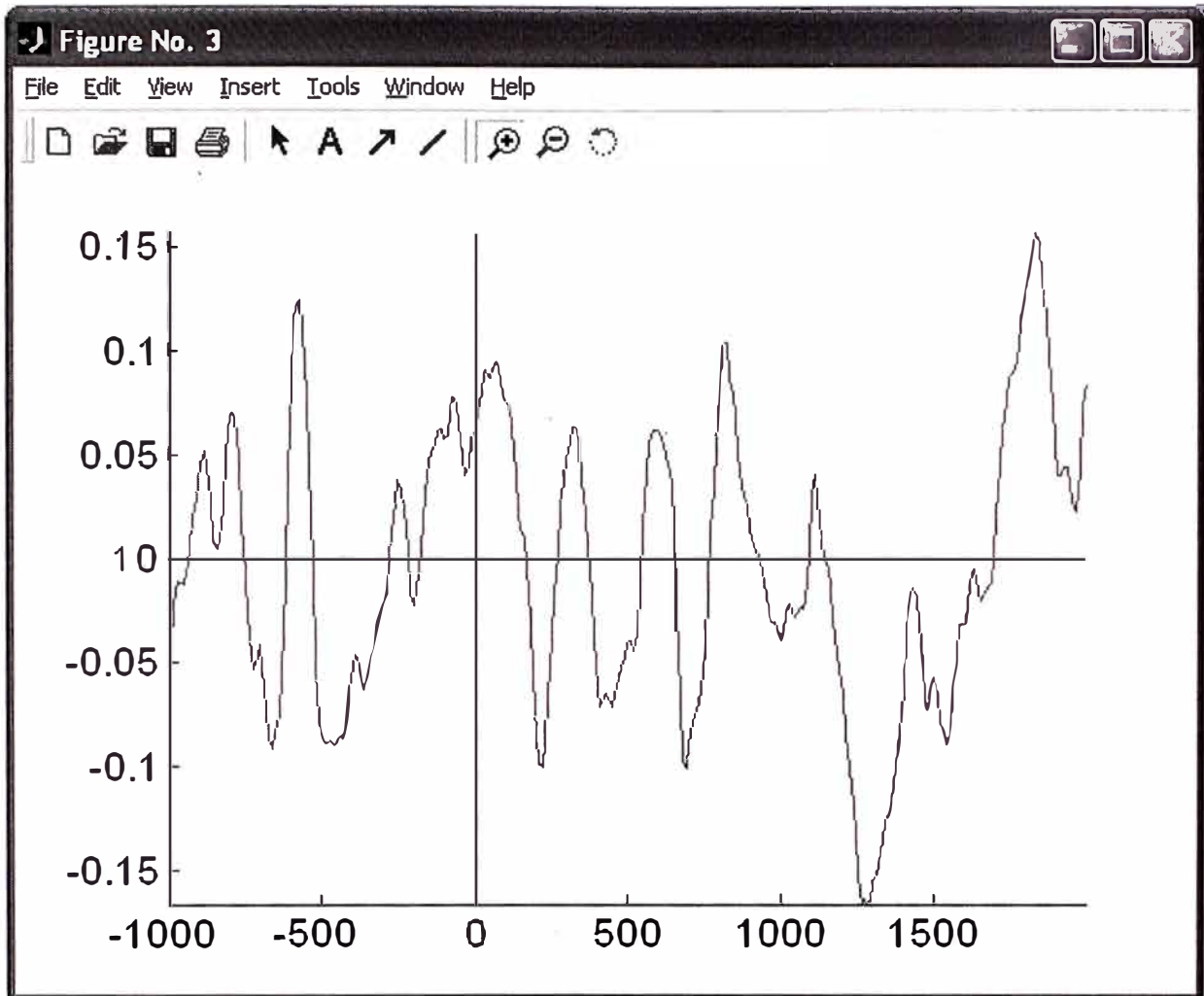


Fig. 3. 136 Gráfica de componentes ERP de un solo canal.

Graficando contribuciones de los componentes ERP

Para graficar la contribución de componentes ERPs a los datos ERP, seleccionar **Plot > Component ERPs > with component maps**, luego presione **OK** para graficar los 7 componentes que contribuye mas al promedio de ERP del conjunto de datos (Fig. 3.137). Notar que los componentes disturbantes pueden ser sustraídos desde los datos previos a graficar la ERP usando **"Indices of component to subtract..."**.

Fig. 3. 137 Formulario para graficar ERPs con mapa de componentes

En la gráfica de abajo, la línea gruesa negra indica la envolvente de la data (mínimo y máximo de todos los canales en cada punto de tiempo) y los colores muestran los componentes ERPs (Fig. 3.138).

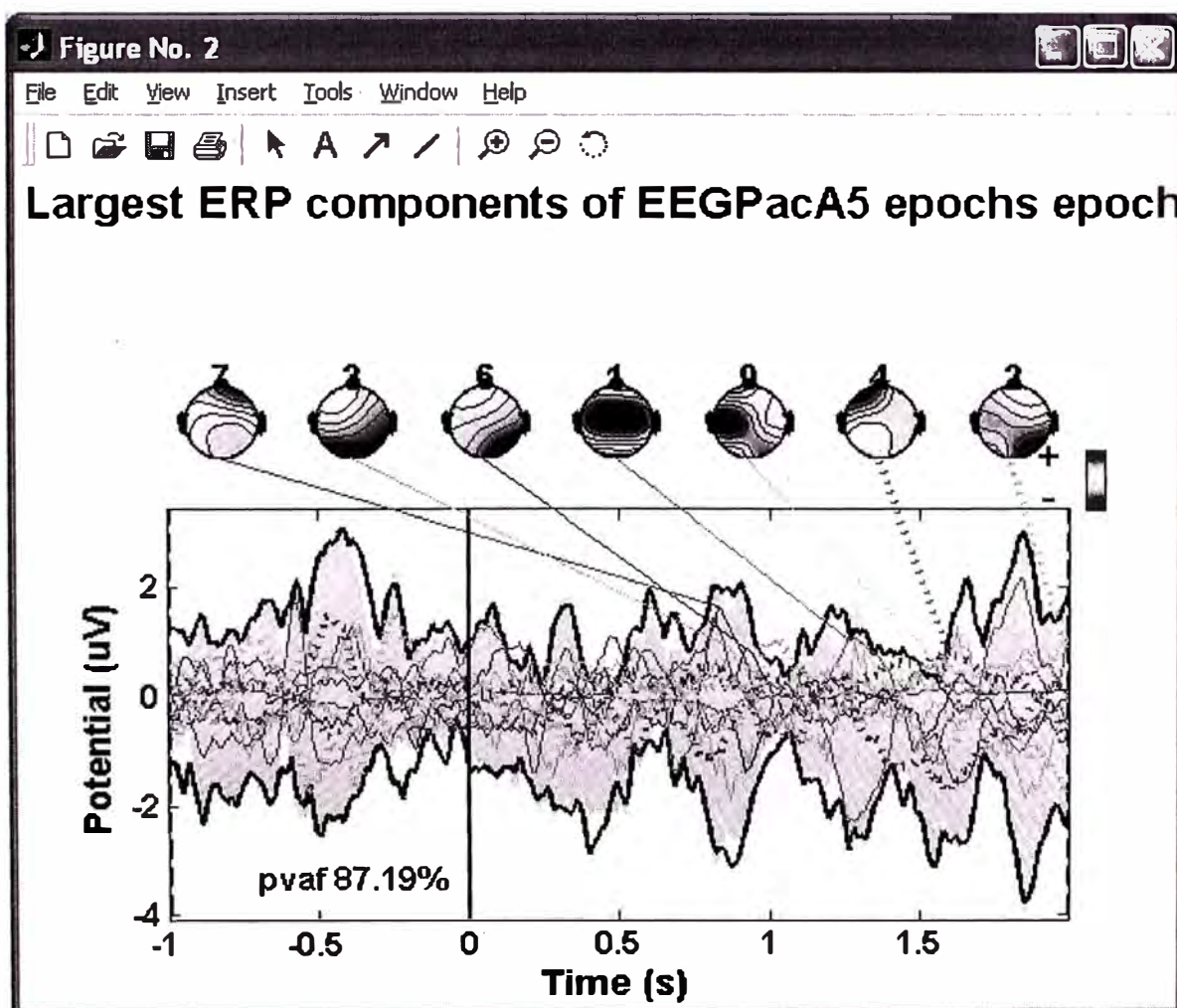


Fig. 3. 138 Grafica de ERPs con mapa de componentes

El dibujo de arriba parece desordenado, por lo que nuevamente graficamos viendo un tramo de un rango menor desde **-600 hasta -300**, tal como se indica abajo (Fig. 3.139).

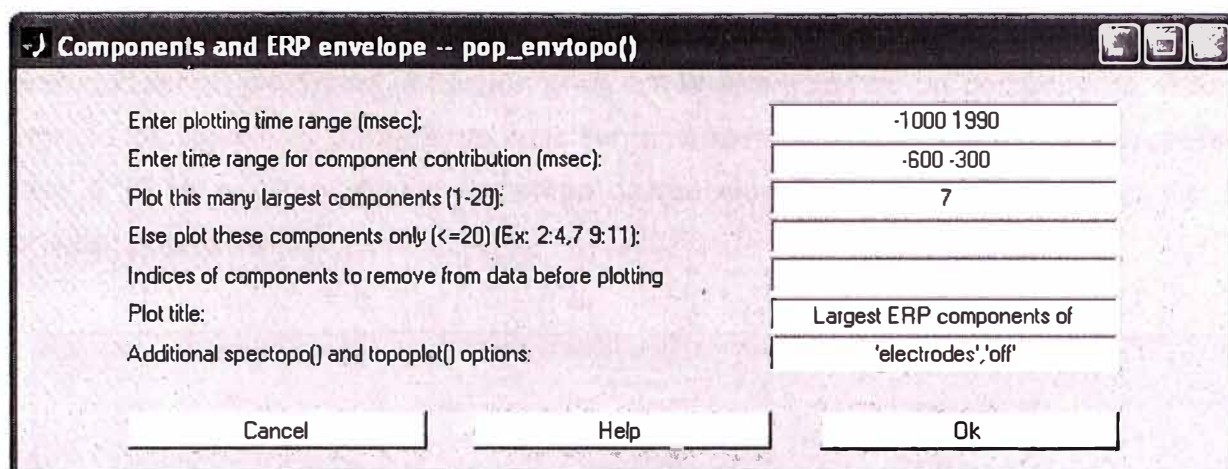


Fig. 3. 139 Eligiendo un rango de -600 hasta 300

Se puede observar en el siguiente gráfico que componentes contribuyen más fuertemente a las ERPs (Fig. 3.140).

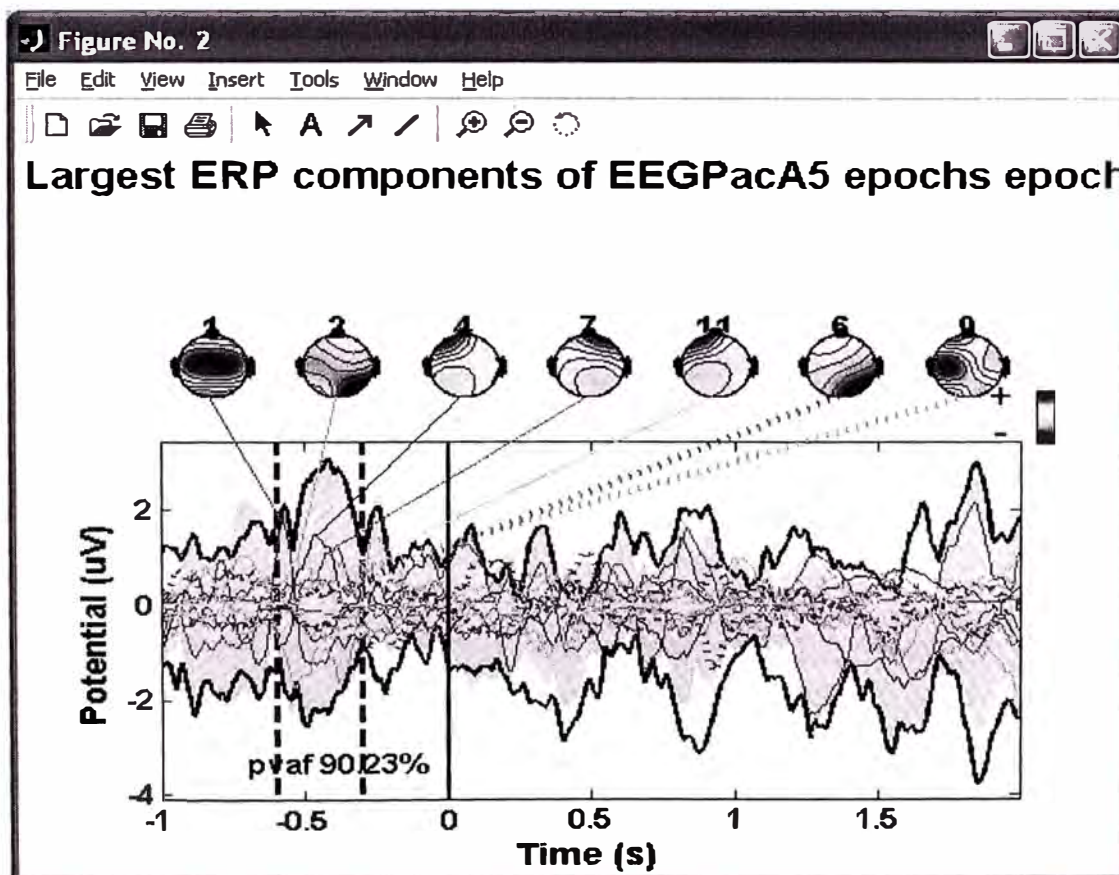


Fig. 3. 140 Gráfica de componentes ERP con rango de -600 a -300

Graficando Componentes ERP-image

Para graficar figuras ERP-image para activación de componentes, seleccionar **Plot > Component ERP image** (llamado `pop_erpimage()`). Esta función trabaja exactamente como usamos para graficar canales de imágenes ERP, pero en vez de visualizar la actividad en un electrodo, la función graficará la activación de un componente. Para ello ingrese los siguientes parámetros en la ventana interactiva para clasificar las pruebas por fase a 10 Hz y 0 ms, dibujar el tiempo de reacción y la potencia de Coherencia inter prueba (Fig. 3.141).

Fig. 3. 141 Formulario de gráfica de Componente ERP-Image

Para la componente 6 (de la figura de abajo) vemos que se ha efectuado para la fase en las frecuencias de análisis de 9Hz a 11Hz, y es eventualmente distribuida en la ventana de tiempo (como se indica en la figura (Fig. 3.142) mostrando la coherencia inter-trial (ITC) o el factor **phase-locking**. Este componente cuenta bastante en la potencia de 10 Hz, pero muy poco en el promedio del ERP. Sobre todo, significa que la potencia a la frecuencia de análisis no cambia a través de los periodos.

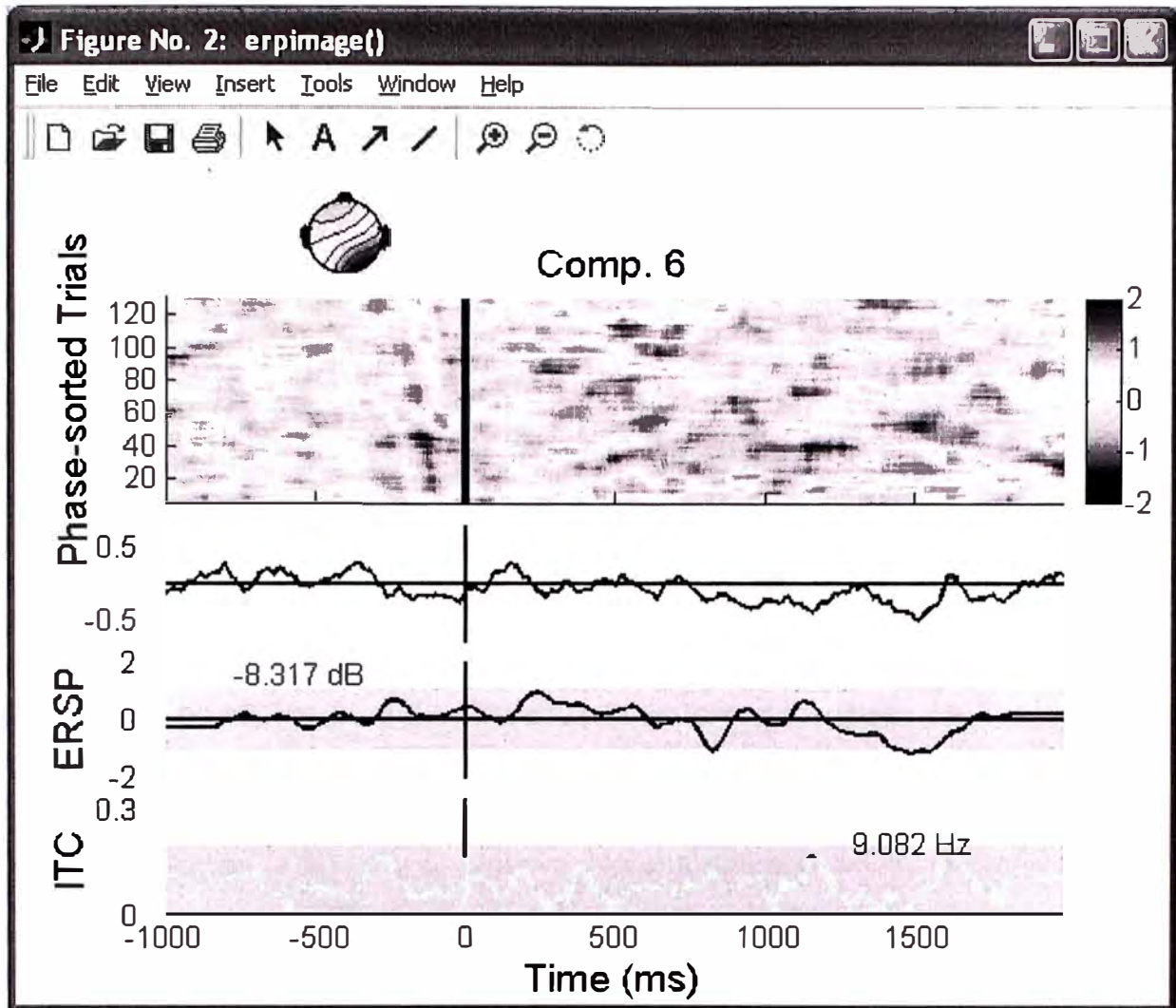


Fig. 3. 142 Gráfica de análisis de fase 9 Hz a 11 Hz.

Descomposición de Tiempo / frecuencia

El análisis tiempo / frecuencia se caracteriza por cambios o perturbaciones en el contenido espectral de los datos considerados como una suma de funciones sinusoidales en ventanas (ejemplo pequeñas ondas sinusoidales). Hay una larga historia y mucho desarrollo reciente de métodos para descomposición de **tiempo / frecuencia**. Los métodos usados en las funciones EEGLAB básicas son simples. Sus detalles matemáticos son dados en una referencia documentada.

Descomponiendo datos de canales

Para detectar perturbaciones espectrales transitorias de eventos relacionados (event-related) (ERSP) (eventos desplazados en el espectro de potencia) y eventos Inter-Trial Coherente (ITC) (event-related phase-locking) en periodos de datasets EEGLAB, seleccionar **Plot > Time frequency transform > Channel time-frequency**. Abajo ingresamos "11" en el campo "Channel number", ".01" en el campo "Bootstrap

significance level", y definir el parámetro opcional "**padratio**" a "**16**" como se indica abajo, dejamos los otros campos en su valor por defecto y presionamos **OK** (Fig. 3.143).

Channel number	11
Epoch time range [min max] (msec)	-1000 1990
Wavelet cycles (0->FFT, see >> help timef)	3 0.5
[set]->Linear coher / [unset]->Phase coher	<input type="checkbox"/>
Bootstrap significance level (Ex: 0.01 -> 1%)	.01
Optional timef() arguments (see Help)	'pdratio', 16, 'plotphase',off
<input checked="" type="checkbox"/> Plot Event Related Spectral Power	<input checked="" type="checkbox"/> Plot Inter Trial Coherence
<input type="checkbox"/> Plot Phase	

Fig. 3. 143 Formulario pop_time()

La ventana **timef()** es la que se muestra abajo como resultado. La imagen de arriba muestra los cambios significativos en la potencia espectral en cada tiempo durante los periodos y en cada frecuencia (<50 Hz). Para analizar estos cambios mas en detalle hacemos click en la figura y una nueva ventana aparece mostrando más de cerca la imagen. La imagen ERSP muestra un extracto pero significativo decremento en potencia. El panel de la izquierda de arriba muestra el espectro de potencia del promedio de línea de base (baseline), y la parte de debajo del panel de arriba, la envolvente de la ERSP (bajo y alto) significa valores dB, relativos a la base de línea (baseline) en cada tiempo en el periodo.

La imagen de abajo (Fig. 3.144) muestra el ITC (Inter-Trial coherente) en todas las frecuencias. Un significativo ITC indica que la actividad EEG en un tiempo y frecuencia dados en una simple prueba empieza en la fase-fija (**phase-locked**) (no fase aleatoria con respecto al evento de tiempo fijo experimental).

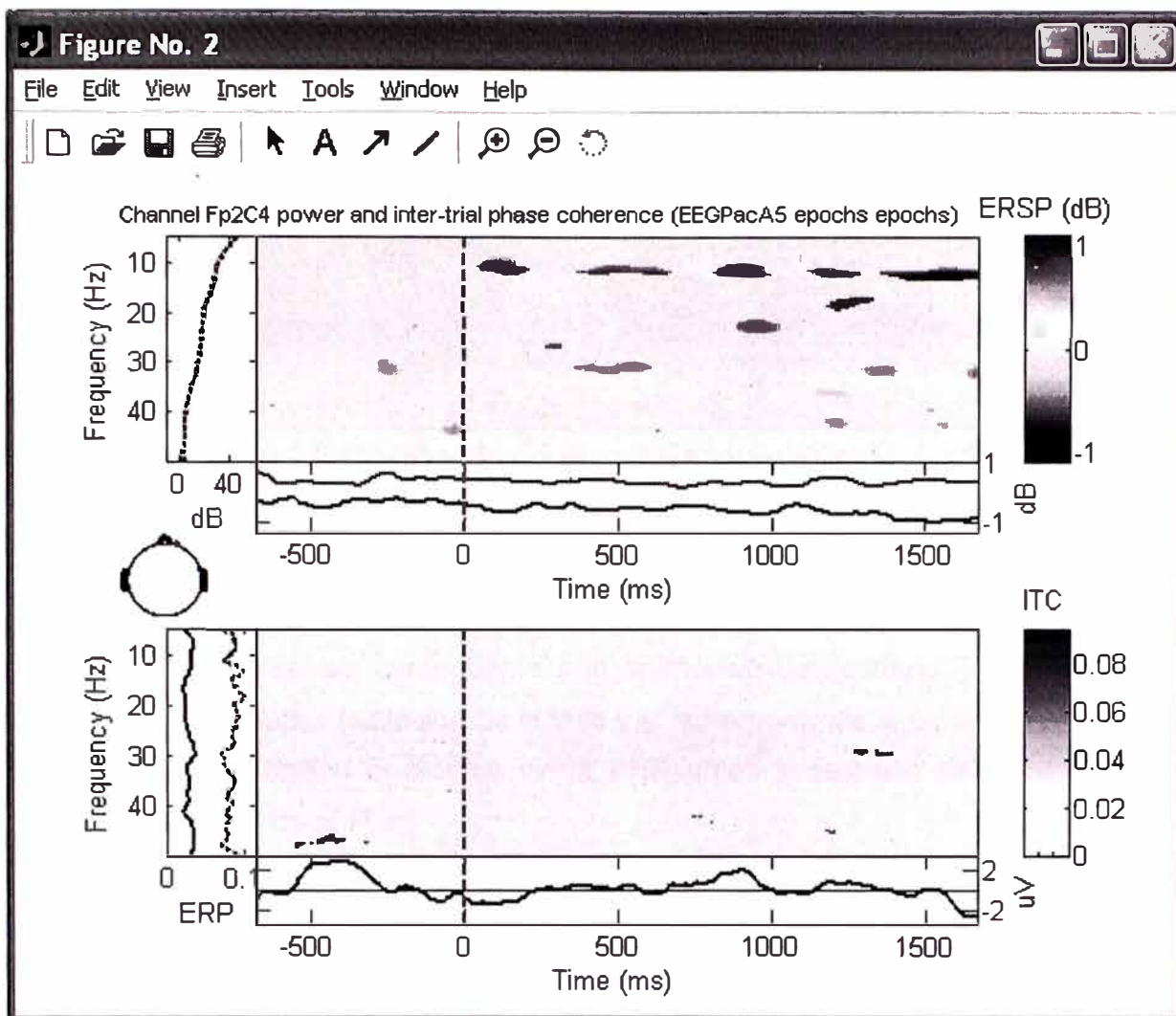


Fig. 3. 144 Gráfica de ERSP, ITC en periodos datasets

Calculando transformadas de componentes tiempo / frecuencia

Es más interesante mirar una descomposición de tiempo / frecuencia de una activación de componente que separar actividades de canales.

Para graficar una transformada de una componente tiempo / frecuencia, seleccionamos **Plot > Time/frequency transform > Component time-frequency**. Para graficar (Fig. 3.145), ingresar **10** para el campo "Component number", "**[-500 100]**" para el campo "Epoch time range", "**0**" (FFT) para el campo "Wavelet cycles", y "**.01**" para el campo "Bootstrap significance level". Cambiar 'padratio' a **16** y adicionar el argumento adicional "'maxfreq','30'", para visualizar solamente frecuencias hasta 30 Hz. Nuevamente presione **OK**. Notar que la función **timef()** descomposición usando FFTs permite cálculos de frecuencias bajas de señales pequeñas.

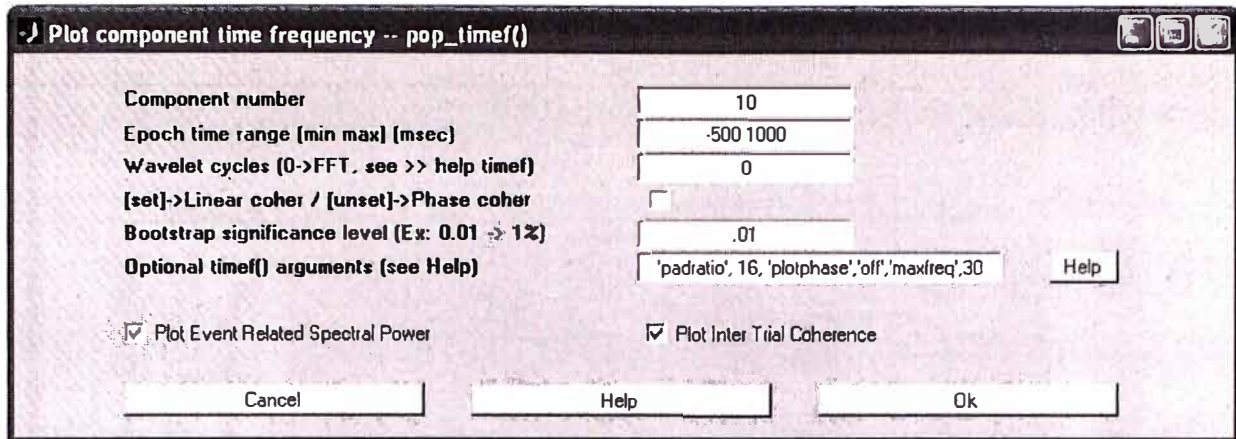


Fig. 3. 145 Ingresando frecuencias hasta 30 Hz en formulario de transformada frecuencia tiempo

En la imagen de abajo (Fig. 3.146), el ITC (panel de abajo) muestra la sincronización marcada entre la actividad componente y la aparición de estímulo. La imagen ERSP (panel de arriba) muestra la fijación de la fase y el incremento de la potencia. Notar que la aparición de la actividad oscilatoria en la ERP antes y después del estímulo no es significativa de acuerdo al ITC.

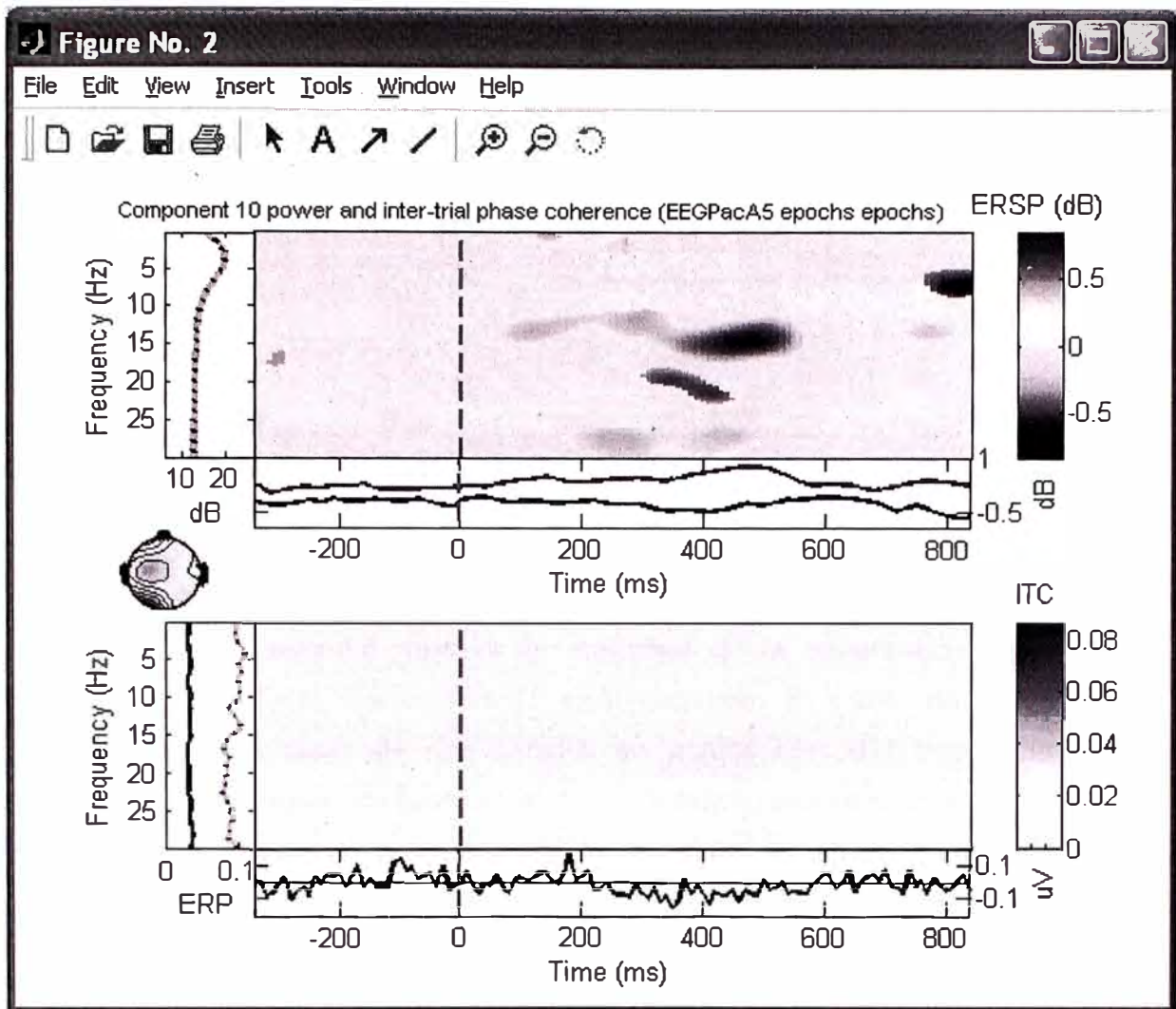


Fig. 3. 146 Gráfica en frecuencias hasta 30 Hz en transformada frecuencia tiempo

Calculando componentes cruzando coherencias

Para calcular el grado de sincronización entre la activación de dos componentes, puede graficarse la coherencia cruzada de sus eventos relacionados (un concepto demostrado primeramente por análisis EEG por Rappelsberger). Aún cuando las componentes independientes son máximamente independientes sobre todo el rango de tiempo de los datos de entrenamiento, pueden empezar sincronizados transitoriamente en bandas de frecuencia específicas. Para graficar la coherencia cruzada de componentes, seleccionar **Plot > Time-frequency transforms > Component cross-coherence**, luego ingresamos los componentes "4" y "9" (puede usar cualquier componente en su descomposición), definir el campo "**Bootstrap significance level**" a "0.01", definir "**padratio**" a "16", luego nuevamente presionamos **OK** (Fig. 3.147).

Fig. 3. 147 Formulario para graficar componentes cruzando coherencias

En la ventana de abajo (Fig. 3.148), los dos componentes empiezan sincronizados (panel superior), el panel superior muestra la magnitud de la coherencia (entre 0 y 1, 1 representa dos señales perfectamente sincronizadas). El panel de abajo indica la diferencia entre las fases de dos señales en puntos tiempo / frecuencia donde la magnitud de cross-coherence (visto en el panel de arriba) es significativa.

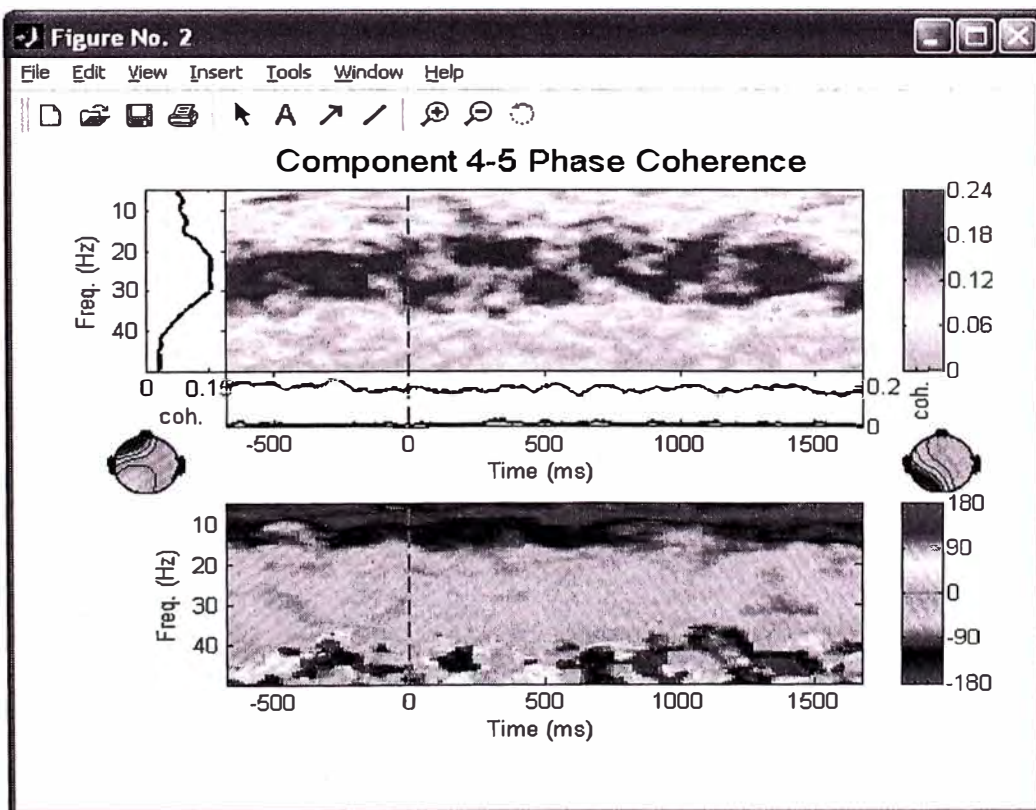


Fig. 3. 148 Componentes cruzando coherencias de fase de canales 4 y 5

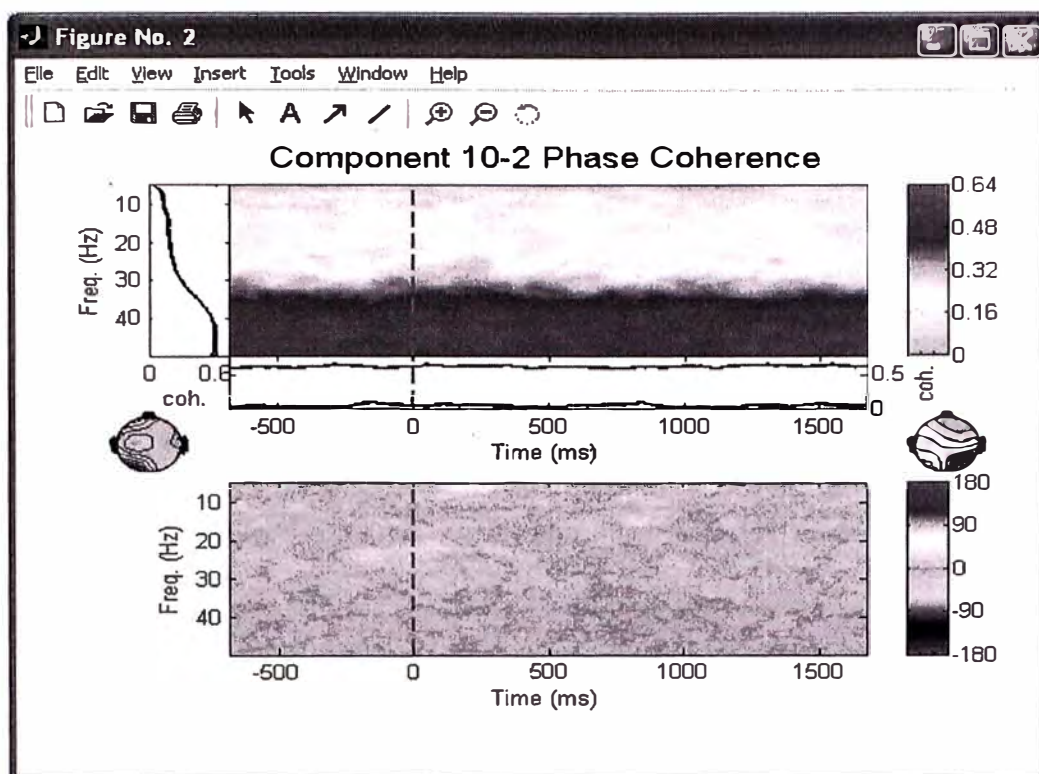


Fig. 3. 149 Componentes cruzando coherencias de fase de canales 10 y 2

Se puede también utilizar la opción **Plot > Time-frequency transforms > Channel cross-coherence** para graficar (Fig. 3.149) a un par de canales la coherencia cruzada en eventos relacionados, pero aquí canales cuyos electrodos están a relativa distancia pueden aparecer sincronizados solamente porque proyecta una fuente grande de datos EEG a ambos de ellos. Otra fuente de confusión puede también afectar la coherencia de canales en caminos intuitivos. Computando coherencias cruzadas en componentes de datos independientes puede indexar sincronización transitoria entre dominios corticales particulares (Fig. 3.150).

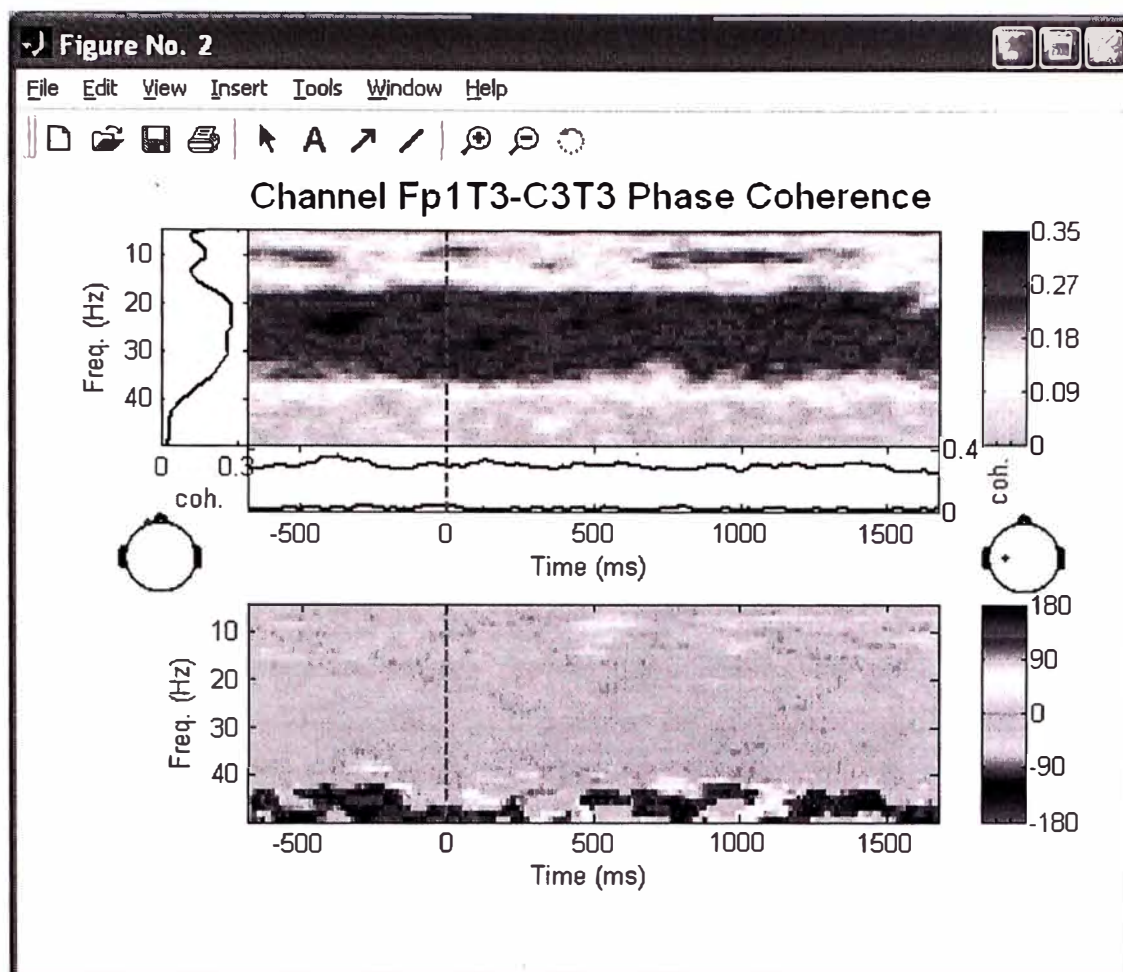


Fig. 3. 150 Componentes de coherencia cruzada en canales de electrodos ubicados a relativa distancia.

Localización de fuentes de dipolos equivalentes de componentes independientes (DIPFIT)

Un mayor obstáculo al usar datos EEG, para visualizar la dinámica del cerebro microscópicamente, es la naturaleza indeterminada del problema inverso: Dado una distribución EEG desde cualquier número de canales del cerebro, cualquier número de distribuciones de fuentes del cerebro puede hallarse lo que lo produce.

El problema de encontrar las localizaciones de muchos dipolos activos simultáneamente, es por propósitos prácticos insolubles, y soluciones mejor dadas diferirán dependiendo en la condición inicial dada a la fuente del algoritmo de inversión. Por esta razón, modelos inversos han tendido a enfocar su atención en casos en los cuales un simple mapa del cerebro bipolar es observado y esperado frecuentemente, muy tempranamente picos en ERP (o ERF magnéticos) responde al crecimiento de promedios en precoz corteza sensorial.

Otros intentos para adecuar múltiples dipolos a grandes porciones de formas de ondas de ERP basadas en “conocimiento previo” (o tanteo) como a donde debe ser localizada la fuente. Esta estrategia frecuentemente ha sido criticada por no ser basado totalmente en la data observada y además sujeto a otros errores.

Usando ICA sin mezclar el EEG promediado es una estrategia radicalmente diferente. ICA identifica temporalmente fuentes de señales independientes en datos EEG multi-canales, y su proyección a la superficie del cerebro. Estos han sido mostrados a ser significativamente mas “parecido al dipolo” (o bipolar) que la EEG sin procesar, aunque sin la localización de los electrodos ni la biofísica de volumen de la figura de propagación en el algoritmo infomax (manuscrito en preparación). Muchos componentes EEG independientes tienen mapas del cerebro que casi perfectamente coinciden la proyección de un simple dipolo del cerebro equivalente. Esta búsqueda es consistente con su generación a través de su sincronía parcial de procesos de potenciales de campos locales (LFP) en un parche conectado o dominio de corteza. El problema de buscar la localización de una simple generación 3-D de un dipolo equivalente de un mapa del cerebro bipolar dado es bien expuesto a través de la localización del dipolo equivalente en general no será en el centro del parche cortical activo (si el parche es radicalmente orientado, el dipolo equivalente será profundo que el).

EEGLAB puede ubicar localizaciones de dipolos equivalentes de mapas del cerebro de componentes independientes usando dos pequeños programas: directamente dentro de Matlab, usando DIPFIT desarrollado por Robert Oostenveld, o por unión a una versión antigua y externa del programa BESA (BESAFIT).

DIPFIT puede ejecutar localización de dipolos usando varios métodos incluyendo minimización de variante residual y varianza mínima forzada lineal usando modelo esférico N-shell.

Adaptación automática de un dipolo

Para ajustar un modelo de dipolo a componentes ICA en datos EEG, primero se necesita realizar la descomposición y luego seleccionar los componentes a ser ajustados.

Para usar DIPFIT, que ajusta mapas de componentes independientes en datos EEGLAB, primero se debe construir o cargar los datos, importar un archivo de localización de canales y calcular una descomposición de los datos.

Para seguir nuestro ejemplo de ajustamiento de un dipolo usado, primero debemos seleccionar el componente a ajustar. Para nuestro caso vamos a utilizar el componente “2” de nuestra muestra de datos, para ello elegimos la opción del menú **Plot > Component maps > In 2-D**, luego ingrese el valor de “2” como el número de componente y poner la opción de ‘**electrodes**’, ‘**pointlabels**’ para obtener la gráfica de abajo. Notar que el archivo para la localización de canales para nuestro caso ya lo cargamos y fue en el proceso inicial de nuestra pruebas en el punto de “**Carga de archivo de Localización de electrodos**”, localización que se muestra en la siguiente gráfica (Fig. 3.151).

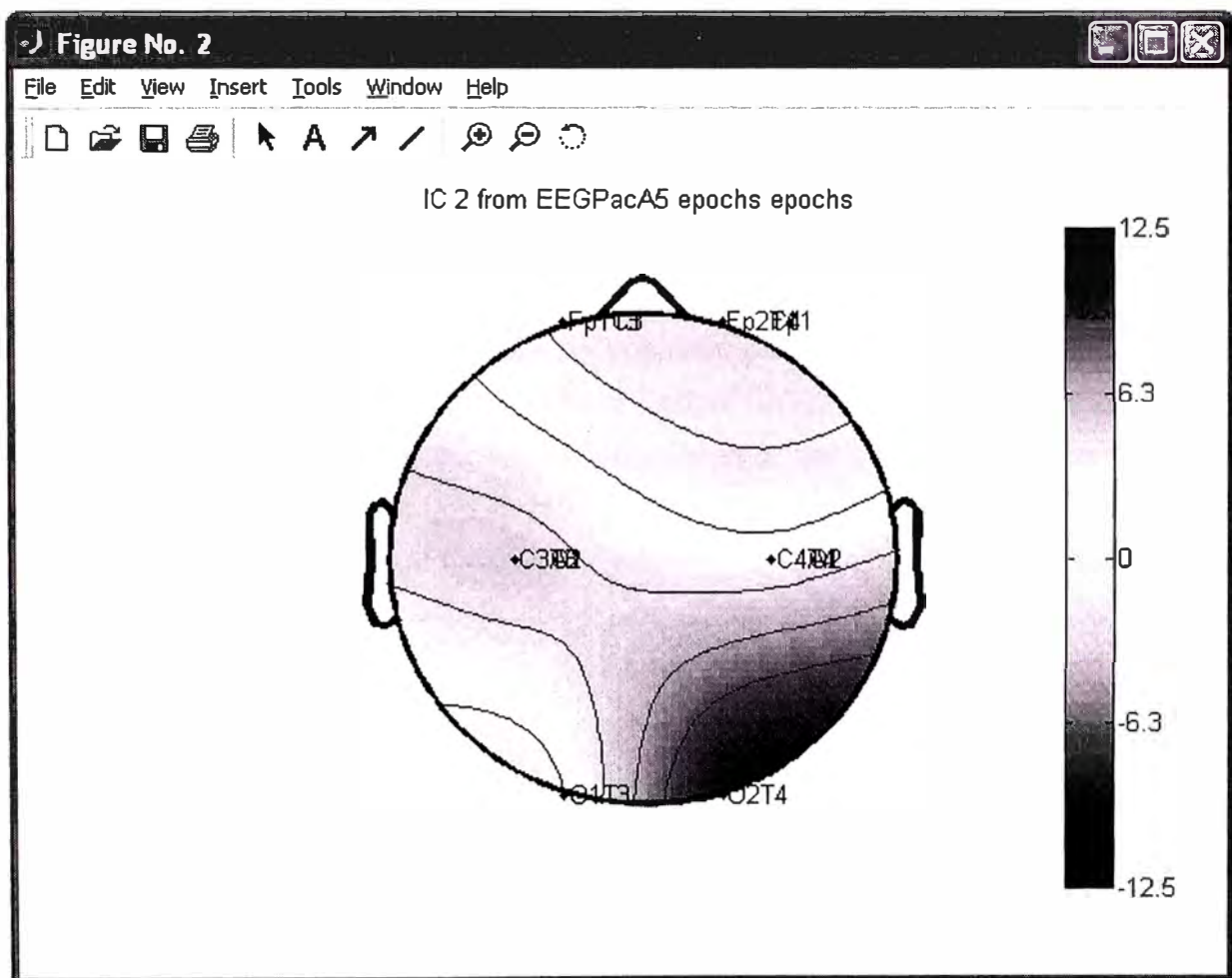


Fig. 3. 151 Identificación del componente 2 para adaptación automática de dipolo

Para encontrar el dipolo equivalente para el componente de arriba, seleccionar el ítem del menú **EEGLAB Tools > Locate dipoles using DIPFIT > Autofit components** para ajustar automáticamente el componente ICA seleccionado. Definir el campo “**Component indices**” al valor de “2”, ingrese el valor de “100” en el campo “**rejection threshold**” (así la solución interactiva es calculada independientemente de la varianza residual del

ajuste rústico), y luego seleccionar la opción de **“Plot resulting dipoles”** para graficar dipolos componentes en el final de la adecuación. Finalmente presione **OK** (Fig. 3.152).

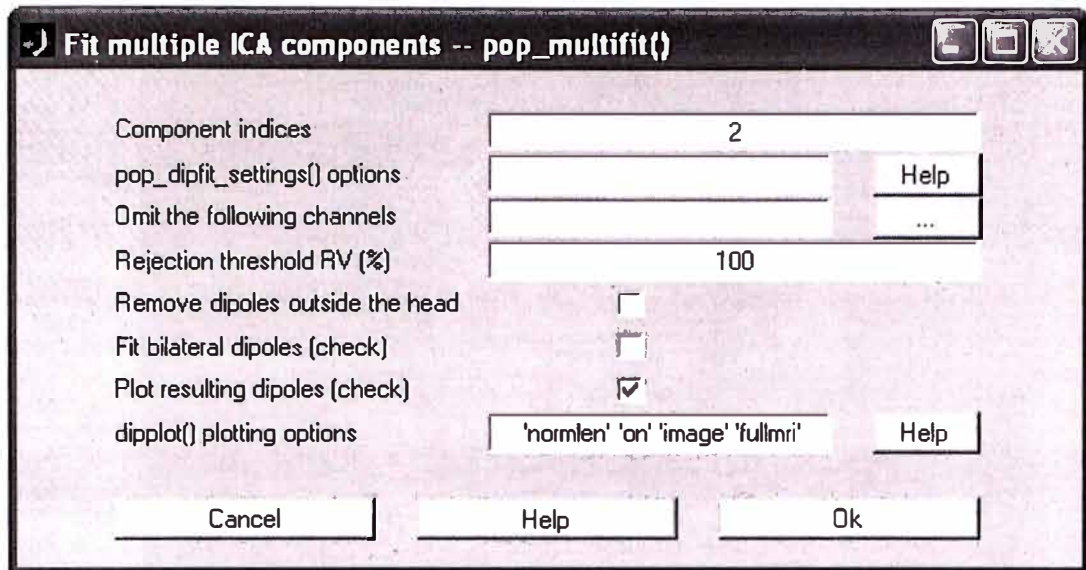


Fig. 3. 152 Formulario para ajustar dipolo con definiciones DIPFIT

La función empieza por la búsqueda de posiciones en una cuadrícula para determinar una posición de inicio aceptable. Luego utiliza un algoritmo de optimización para encontrar la posición exacta de cada dipolo. Al final del procedimiento de adecuación aparece la siguiente ventana (Fig. 3.153).

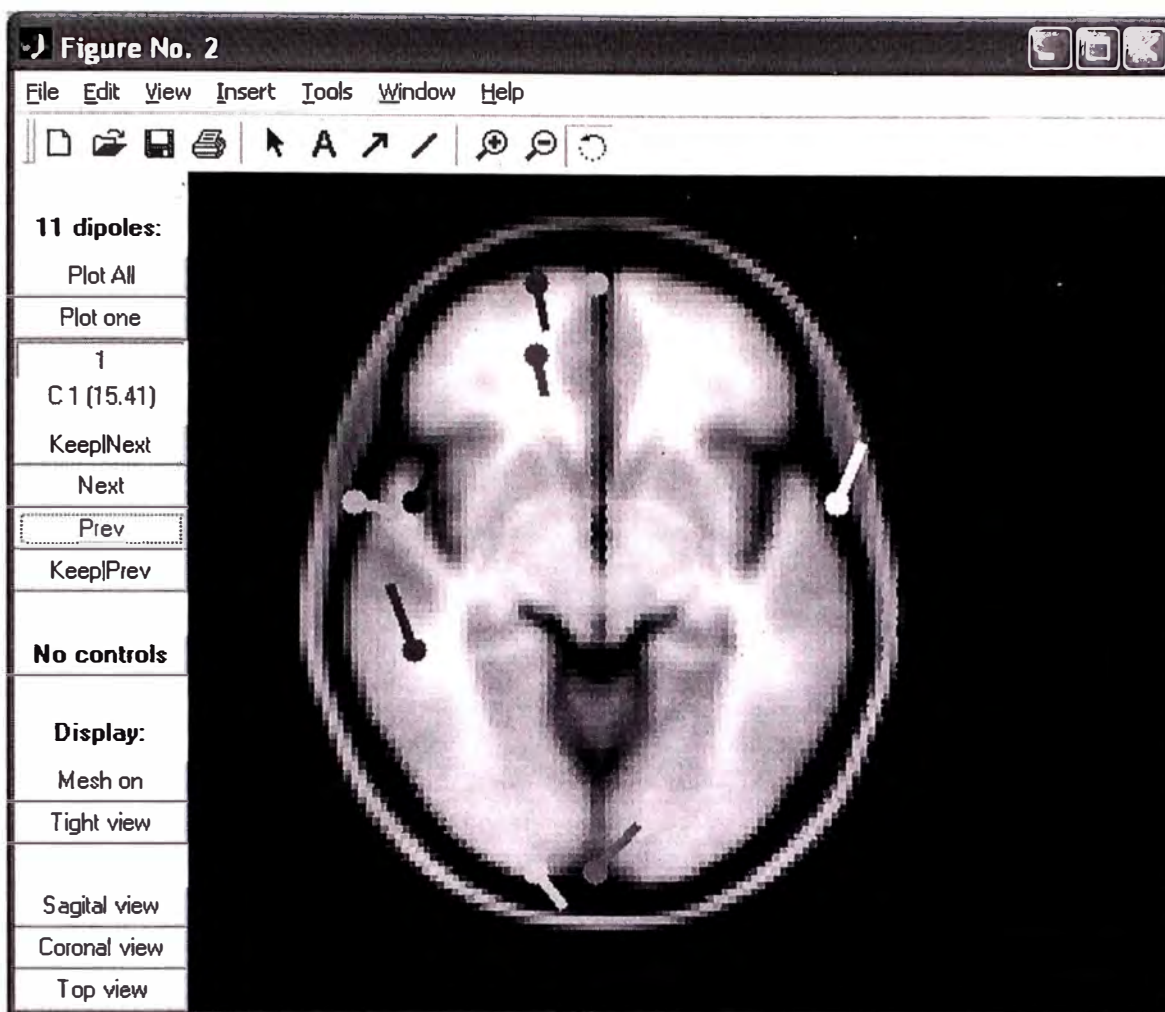


Fig. 3. 153 Dipolos encontrados y su ubicación en el mapa del modelo de la cabeza

La variación residual es mostrada por el número de componente en la esquina superior izquierda. "C 1 (15.41)" indica que el componente que esta siendo graficado es el componente 1 y que la variación residual es 15.41%.

La función de visualización permite rotar el modelo de la cabeza en 3-D con el Mouse, graficando tajadas de MRI muy cercano a los dipolos equivalentes (Fig. 3.154).

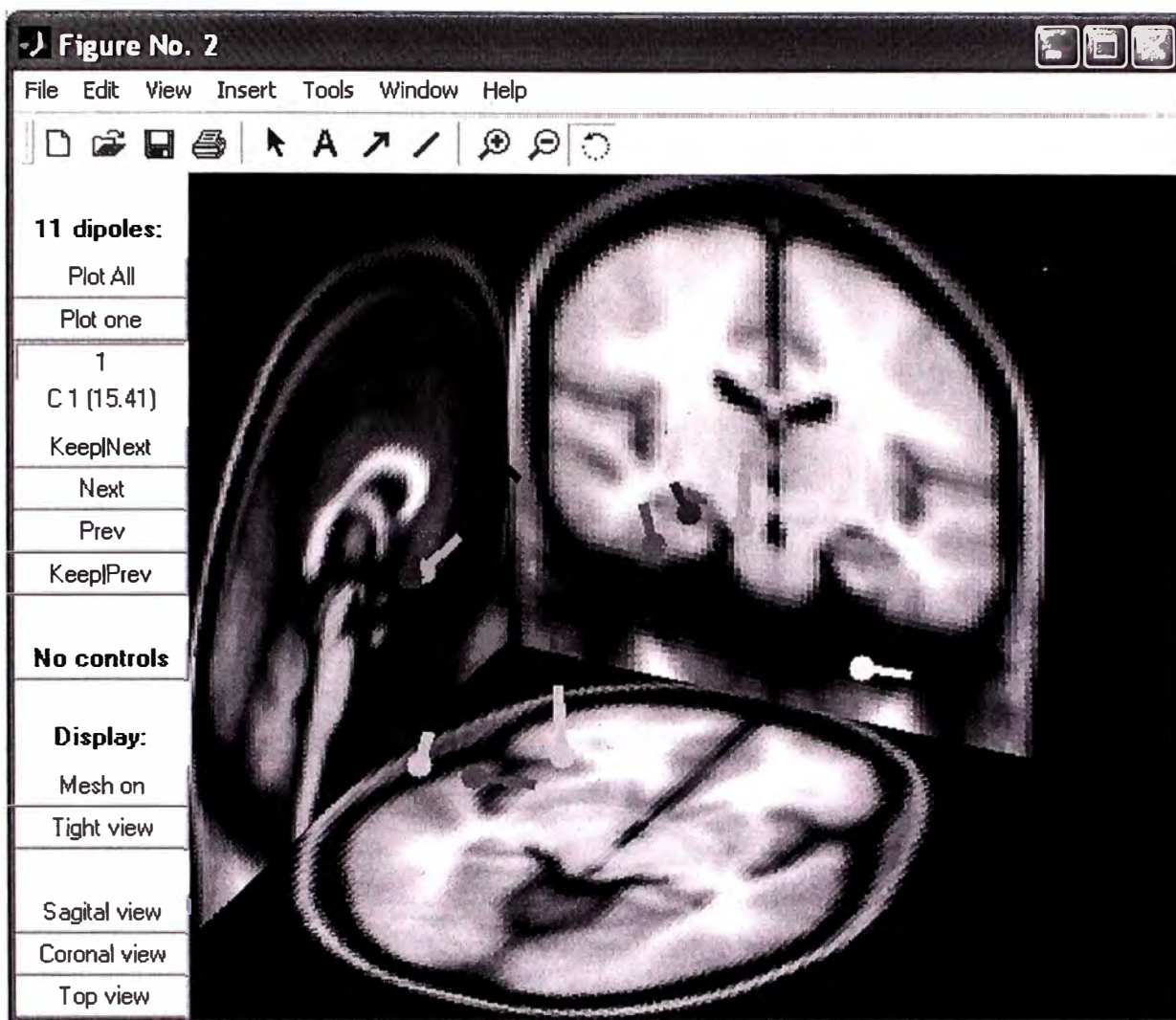


Fig. 3. 154 Visualización y ubicación de los dipolos en 3D en el modelo de la cabeza

En el anterior ejemplo utilizamos definiciones DIPFIT por defecto para la búsqueda de dipolos equivalentes en los componentes del mapa del cerebro. Otros menús permiten a los usuarios parámetros de ajustamiento y sintonía fina. Para ello seleccionar el ítem **Tools > Locate dipoles using DIPFIT > Head model and settings** del menú EEGLAB para modificar definiciones del DIPFIT. Esto hace aparecer la ventana de abajo (Fig. 3.155).

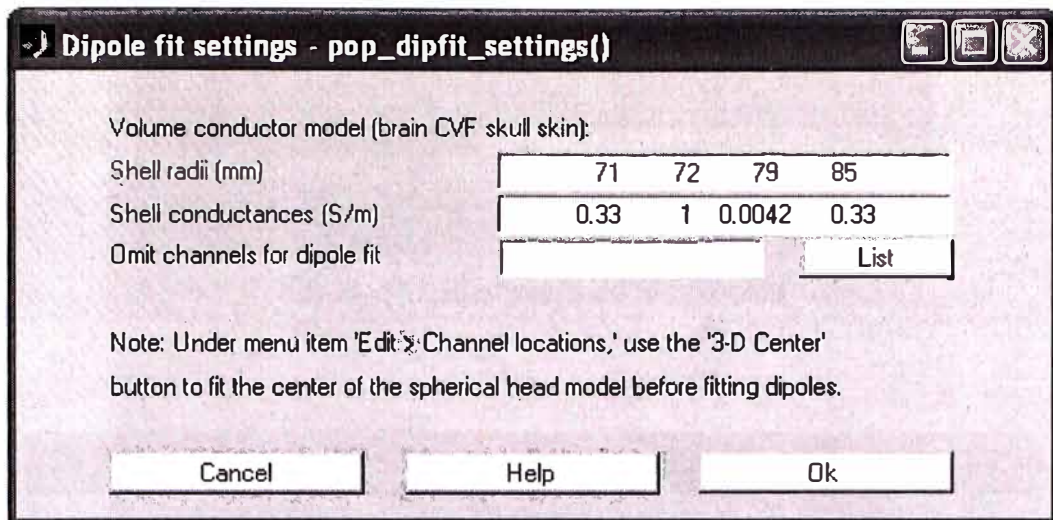


Fig. 3. 155 Formulario para ajuste fino de las definiciones de los dipolos

El primer campo de arriba “**Shell radii (mm),**” especifica el radio esférico del caparazón usado en el modelo. El valor por defecto son cuatro-caparazones modelos simulando el cerebro, meninges, cráneo, y piel. (Removiendo uno o más de estos se puede intentar localizar dipolos usando solamente 3-caparazones o 2-caparazones de modelos.). El segundo campo “**Shell conductances (S/m),**” da modelos de valores de conductancia para cada uno de estos caparazones. Los parámetros por defecto corresponden a un parámetro típico para un modelo de 4-shell, el cual fue también implementado como valor por defecto en BESA 3.0. El último campo “**Omit channels for dipole fit**” permite ajustar dipolos usando un subconjunto de canales de datos (Fig. 3.156, Fig. 2.157, Fig. 2.158, Fig. 2.159, Fig. 2.160 y Fig. 2.161).

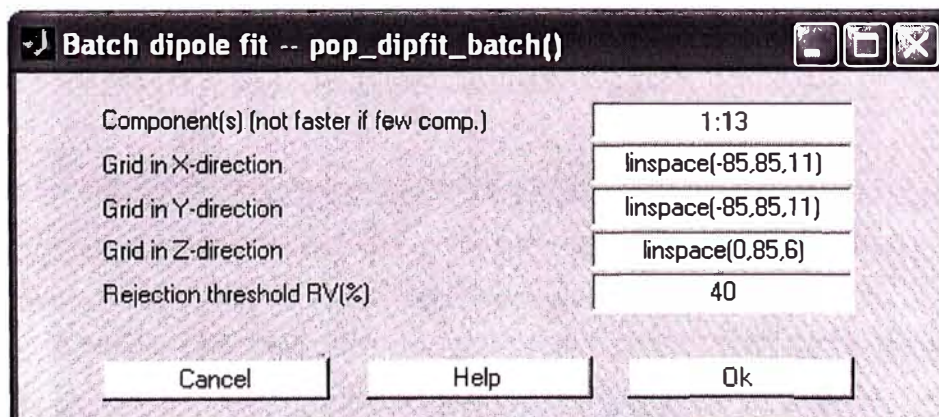


Fig. 3. 156 Ajuste de los dipolos en los 13 canales

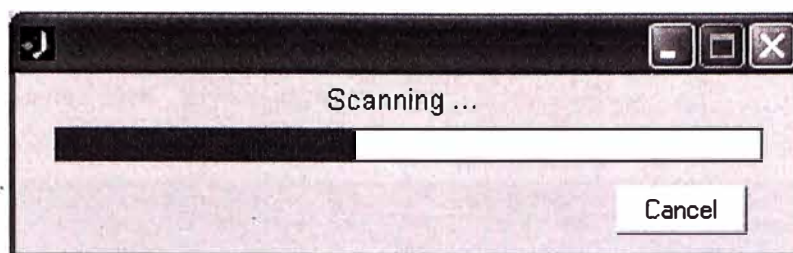


Fig. 3. 157 Búsqueda de los dipolos

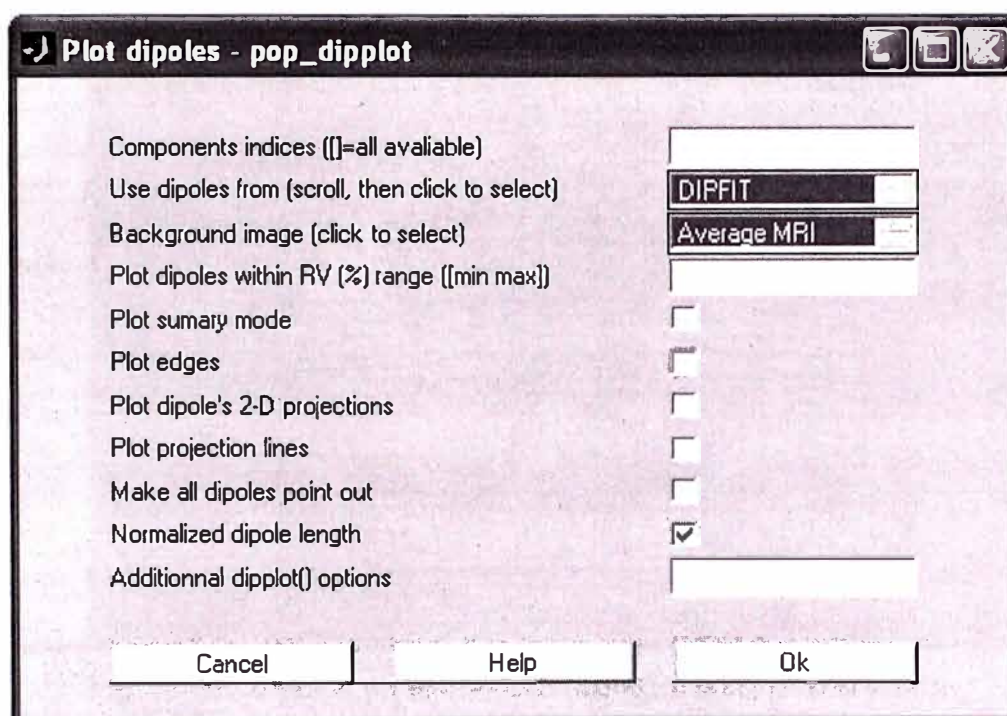


Fig. 3. 158 Formulario para graficar los dipolos

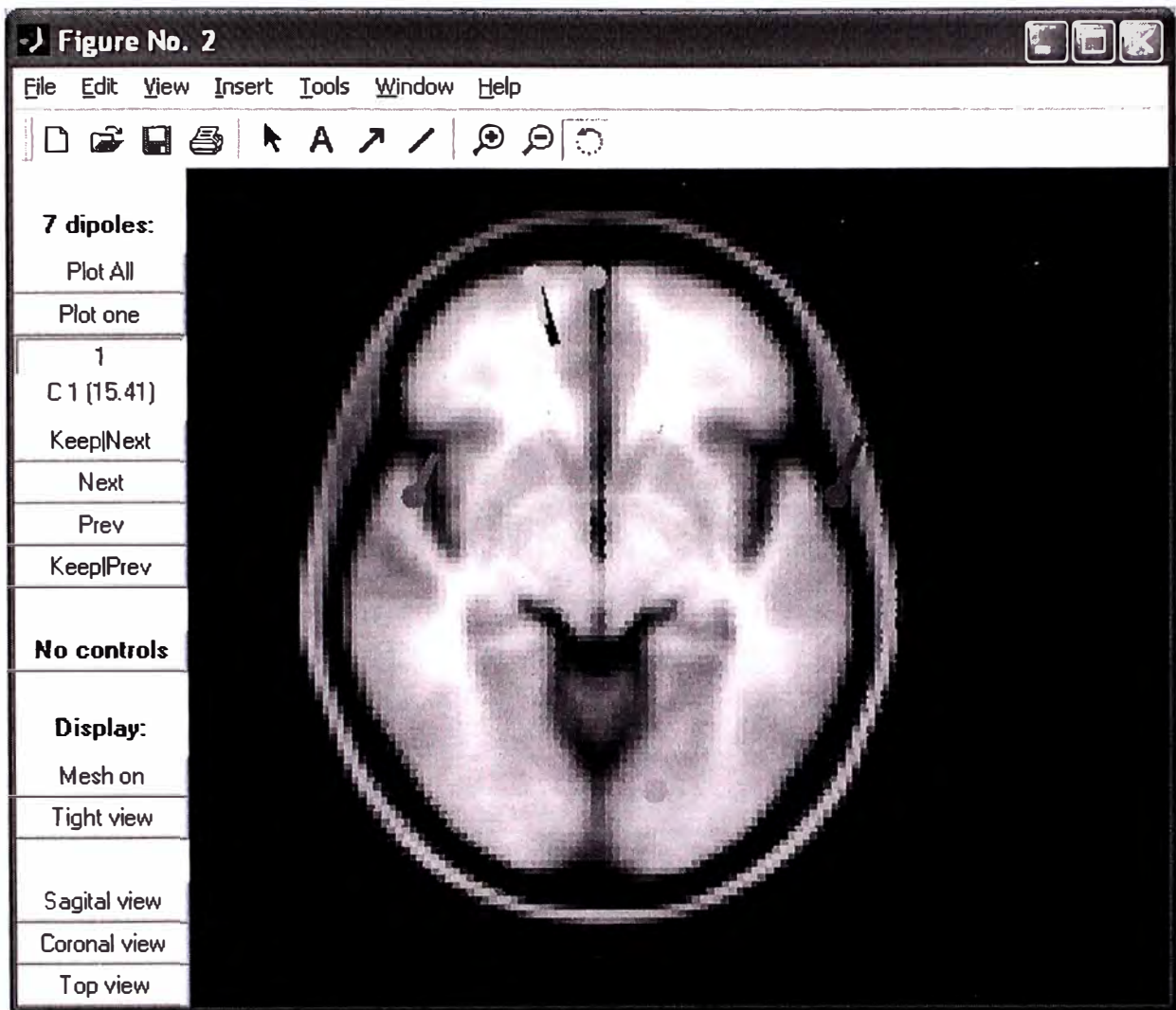


Fig. 3. 159 Dipolos en el mapa del cerebro vista desde arriba

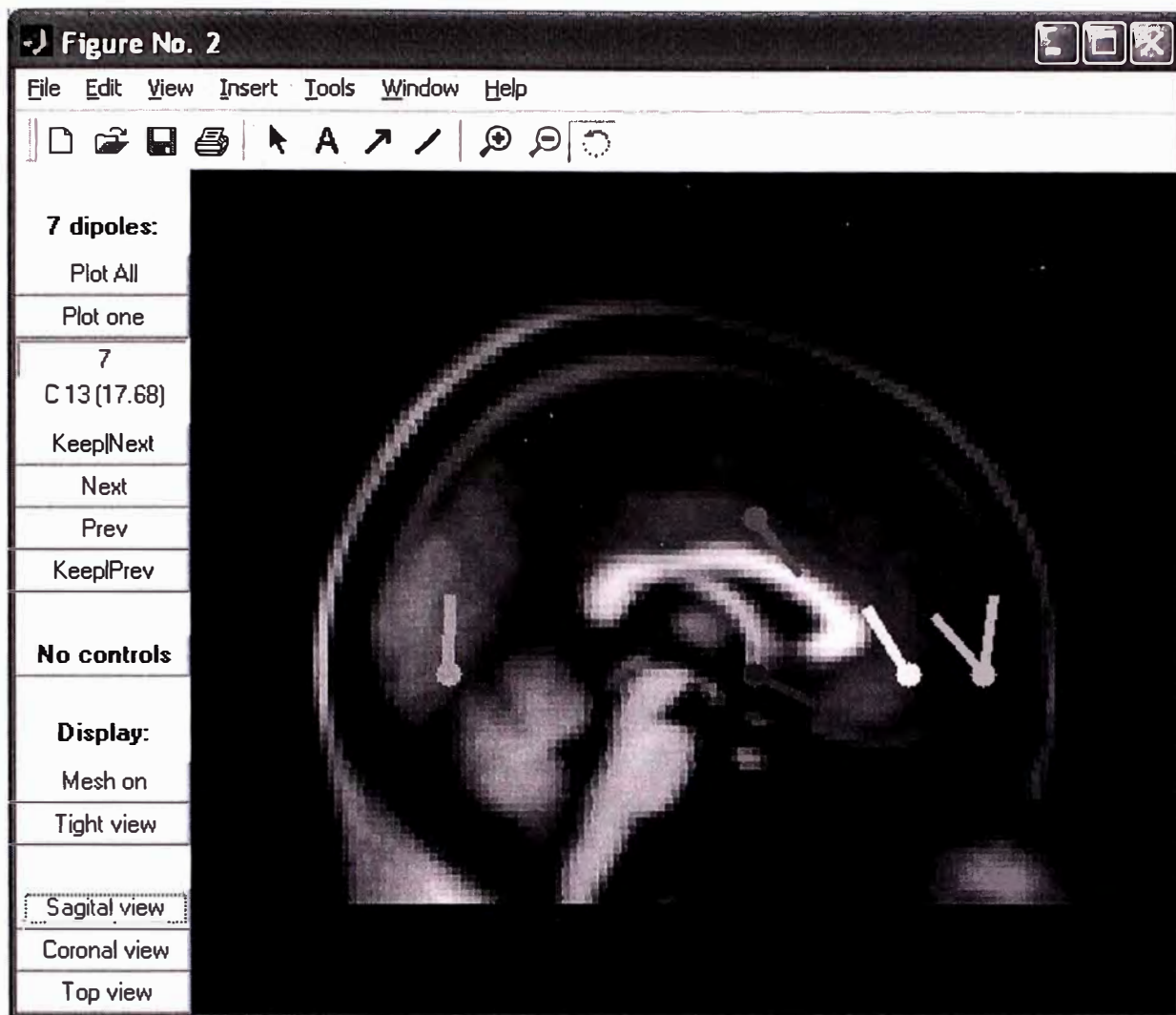


Fig. 3. 160 Dipolos en el mapa del cerebro vista lateral

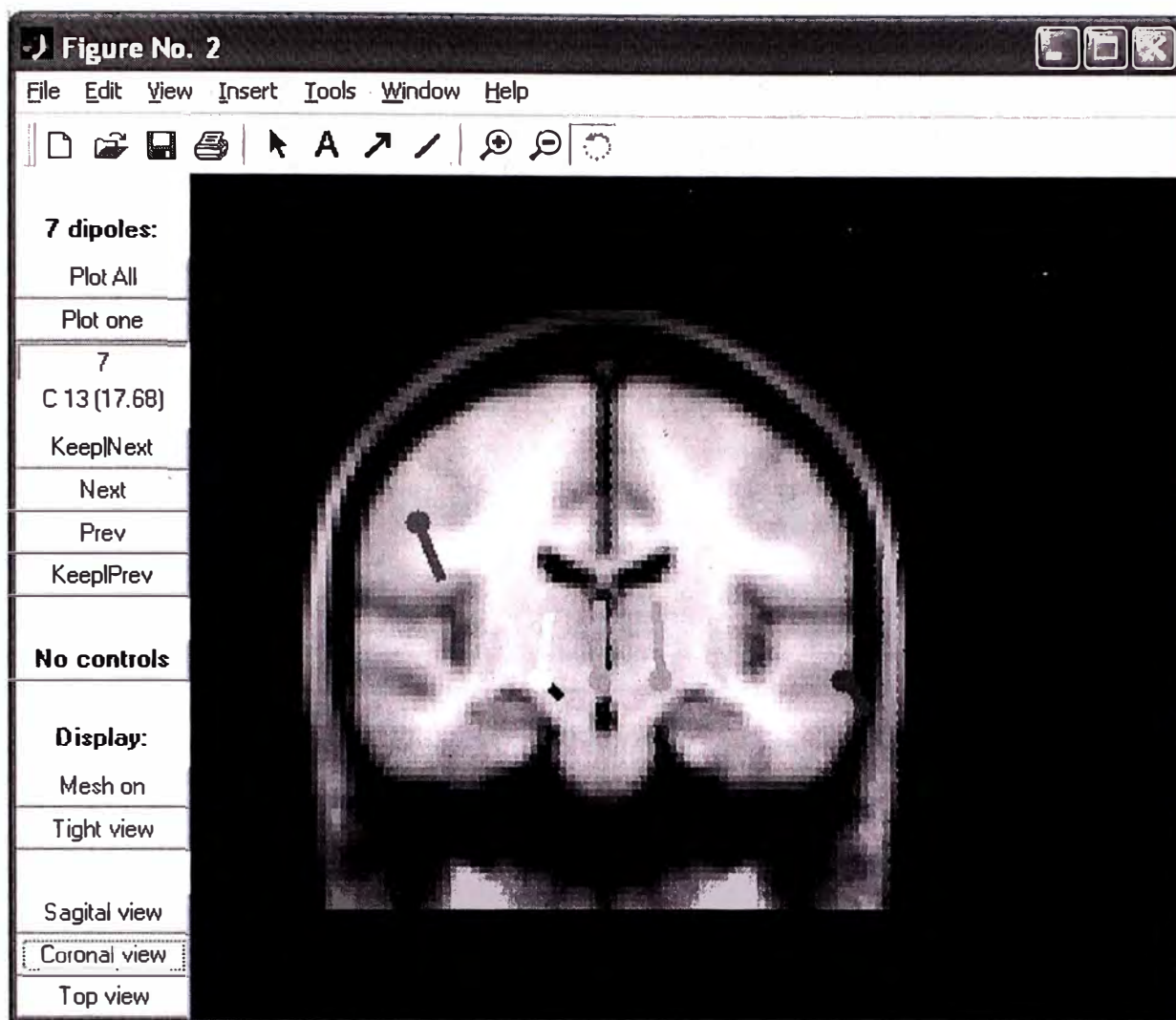


Fig. 3. 161 Dipolos en el mapa del cerebro vista frontal

Visualizando modelos de dipolos

Para visualizar modelos, utilice el menú **Tools > Locate dipoles using DIPFIT > Plot component dipoles** y seleccione los componentes número **3** y **5** para graficar tal como se muestra abajo (asumir que hizo un ajuste fino para ambos componentes). Seleccione todas las opciones excepto el “**Summary mode**”, ingrese ‘**view**’,**[51 18]** en el campo “**Additional dipplot() options**” para definir la vista 3-D inicial, esta función imprimirá los índices de los componentes (Fig. 2.162).

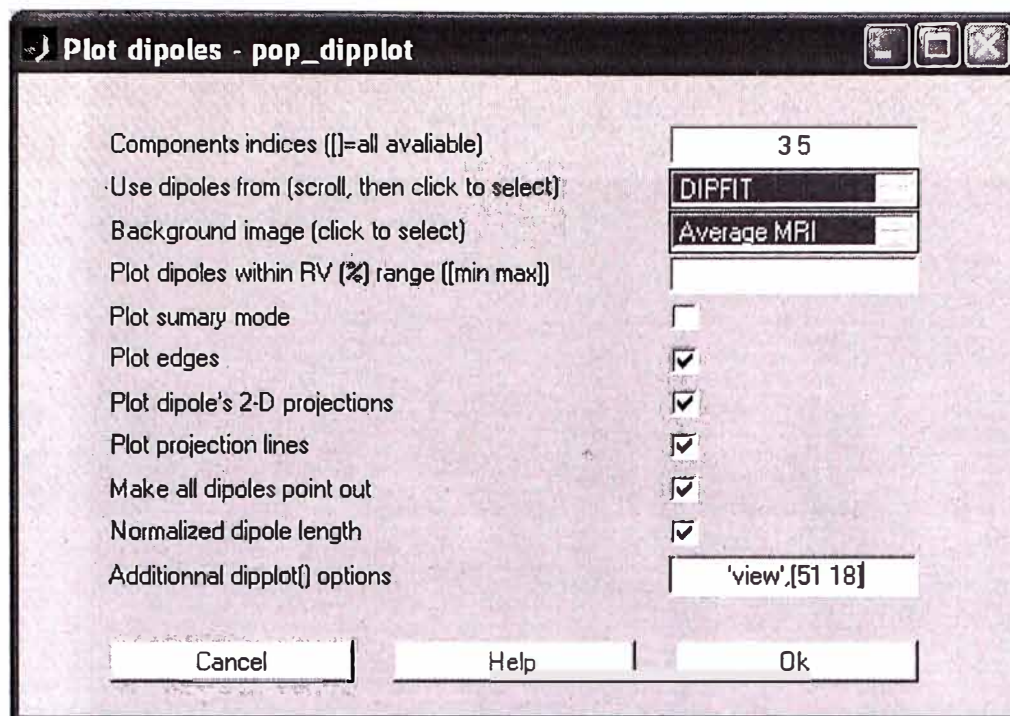


Fig. 3. 162 Selección de componentes 3 y 5 con todas las opciones de gráfica excepto "summary mode"

El siguiente gráfico se visualiza luego de presionar **OK** (Fig. 2.163).

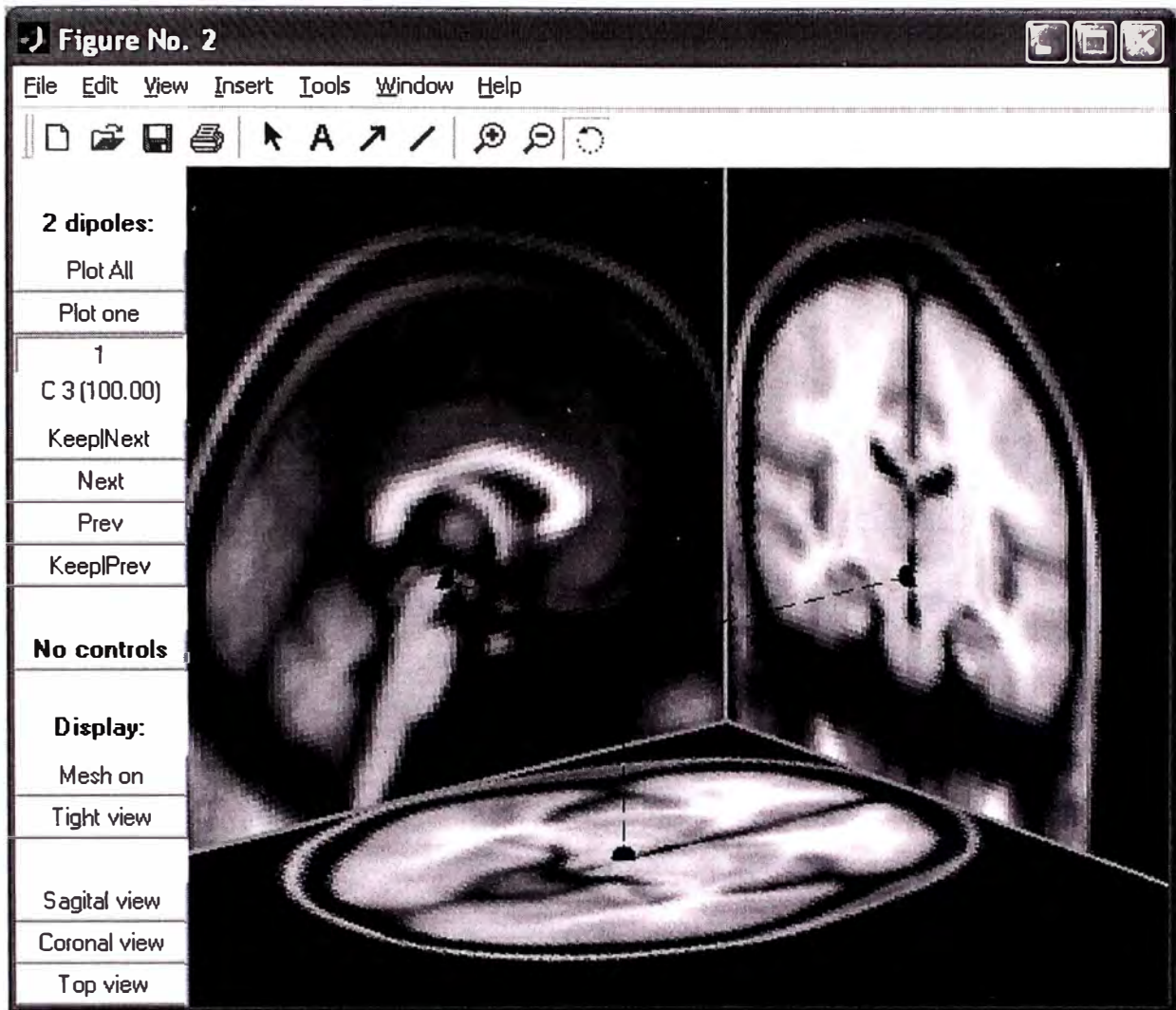


Fig. 3. 163 Dipolos de los componentes 3 y 5

Presione el boton "**Plot one**" y puede ver cada uno de los componentes presionando los botones "**Next/Prev**". Luego presione el botón "**Tight view**" para visualizar el MRI promedio en rodajas cercanas a la localización del dipolo. Intente mover la perspectiva 3-D de la gráfica de la vista integrada con el Mouse y/o seleccione los ángulos de vista cardinales (sagital, coronal o top) usando los botones de control izquierdos de abajo (Fig. 2.164).

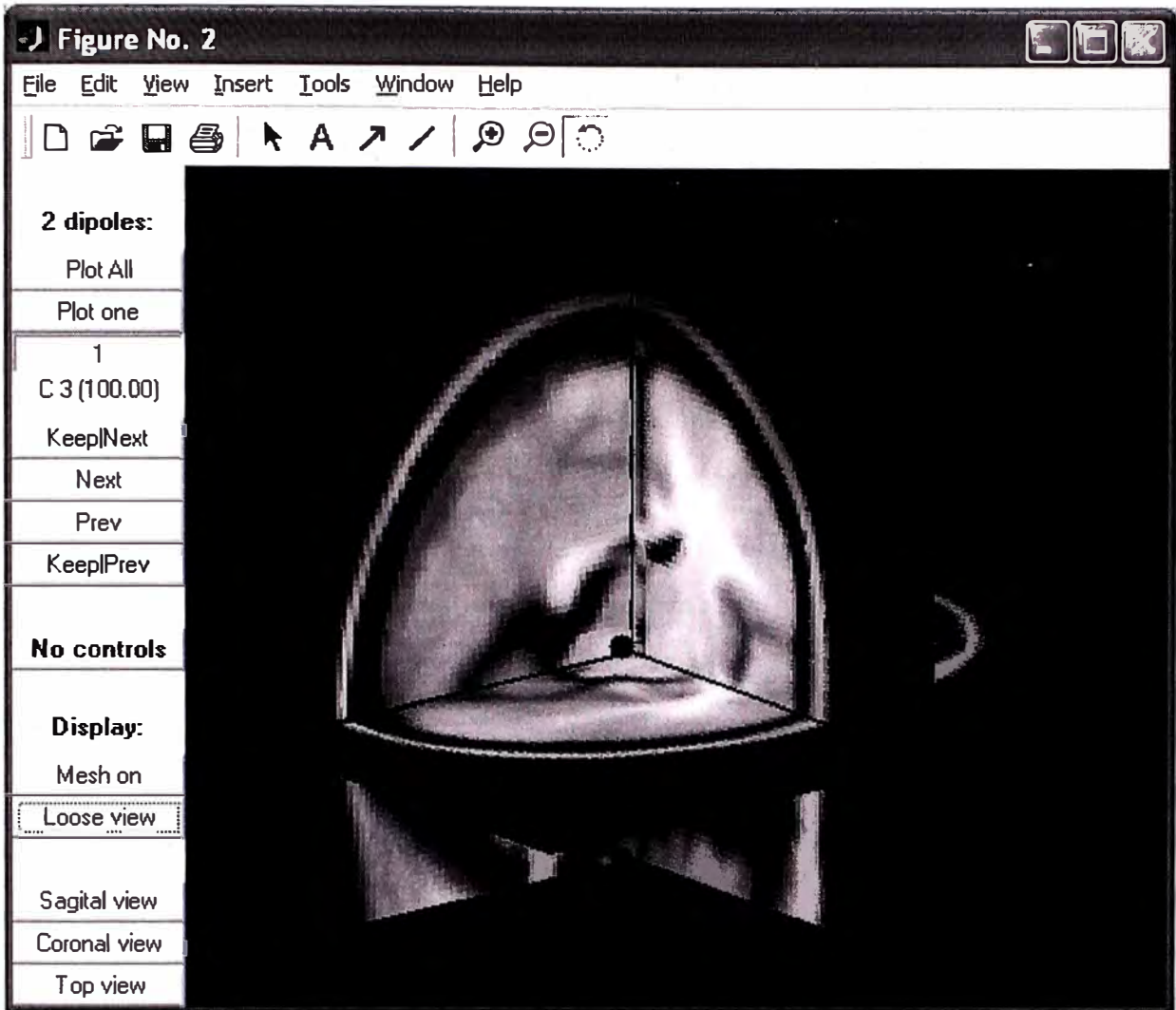


Fig. 3. 164 Grafica de los dipolos con la opción "Plot one"

Finalmente, nuevamente invocar el menú **Tools > Locate dipoles using DIPFIT > Plot component dipoles**. Seleccionar la opción de **"Summary mode"** y deseccione las otras opciones, ingrese 'num', 'on' en el campo **"additional dipplot() options"** para visualizar los números de los componentes (Fig. 2.165).

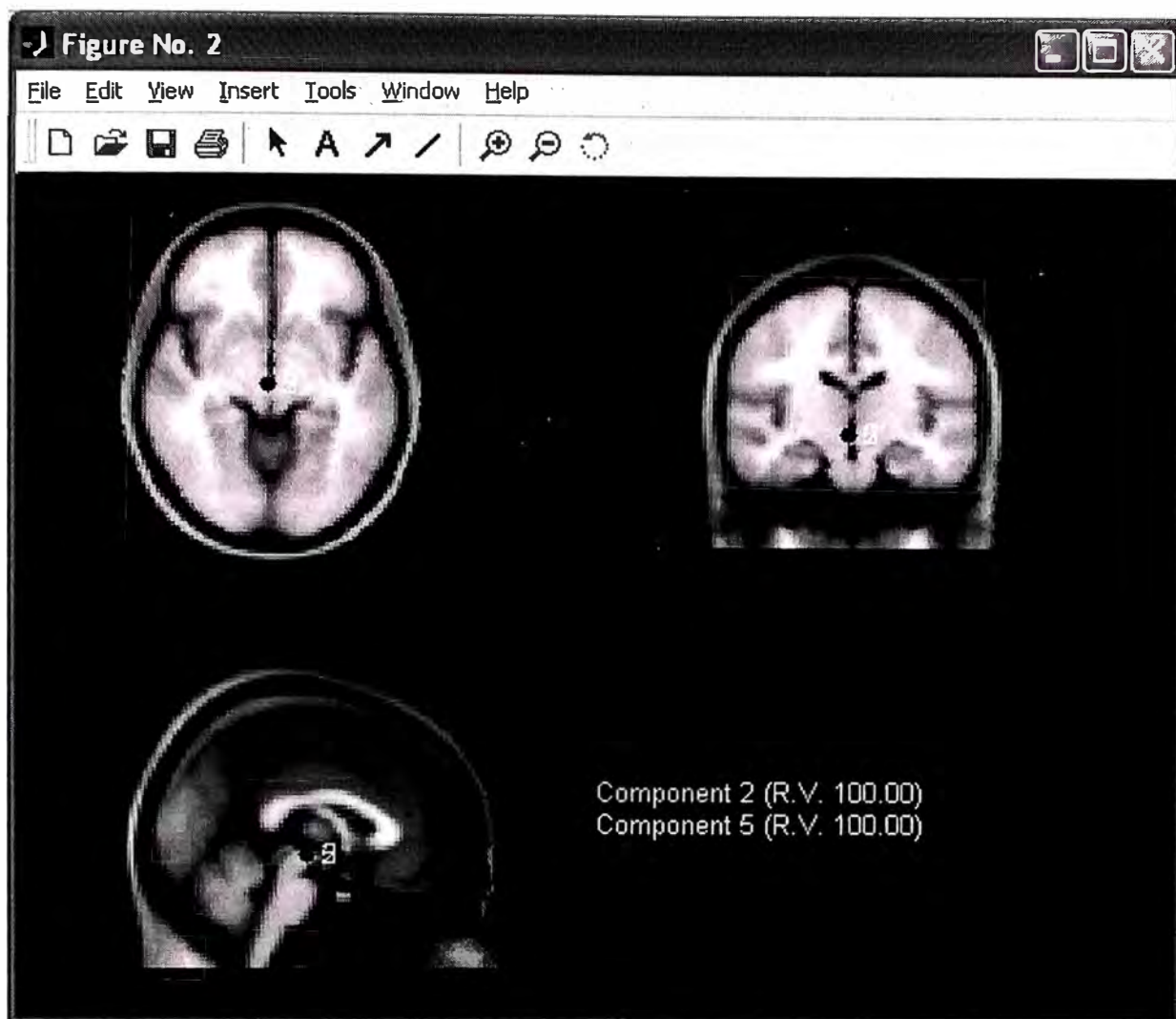


Fig. 3. 165 Gráfica de los dipolos de los componentes 3 y 5 en la opción "Summary mode"

Graficando localización de dipolos en los mapas del cerebro

Usar el menú **Plot > Component maps > In 2-D**, ingrese "1:13" en el índice de componentes (los dipolos componentes son pre-almacenados en la muestra de datos, si se desea procesar otro conjunto de datos, primero usar el menú **Tools > Locate dipoles using DIPFIT > Autofit components** para automáticamente adecuar los componentes ICA de los canales indicados). Seleccionar la opción "**Plot dipole**", como se indica abajo, y luego presione **OK** (Fig. 2.166).

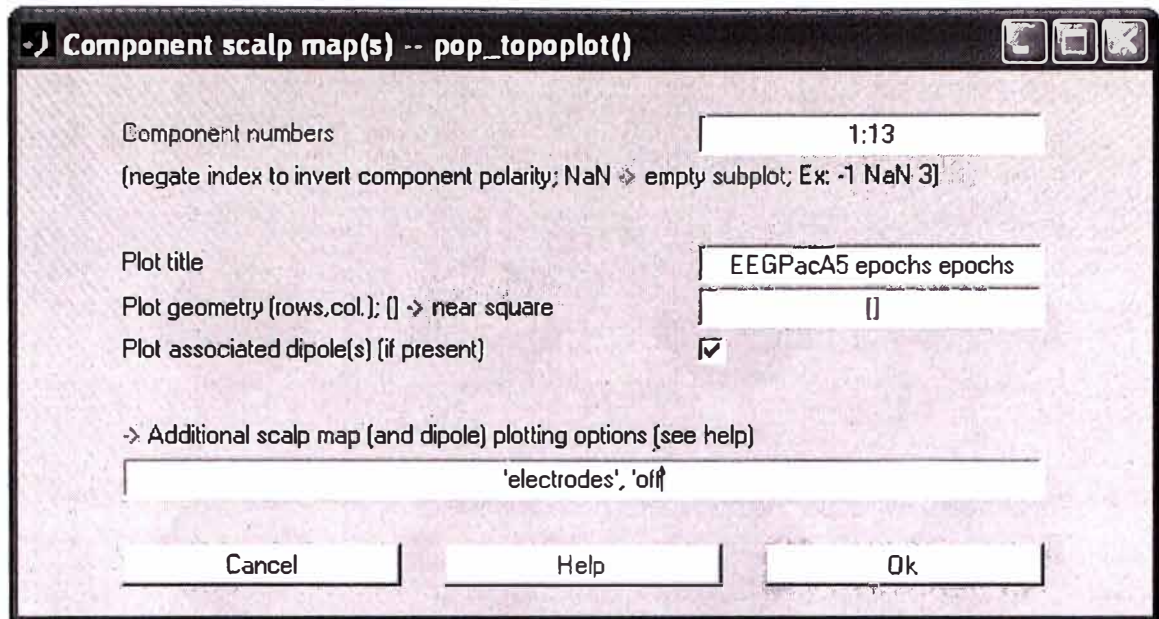


Fig. 3. 166 Formulario para graficar los dipolos de todos los canales de los electrodos

Las opciones elegidas, muestra la siguiente imagen (Fig. 2.167).

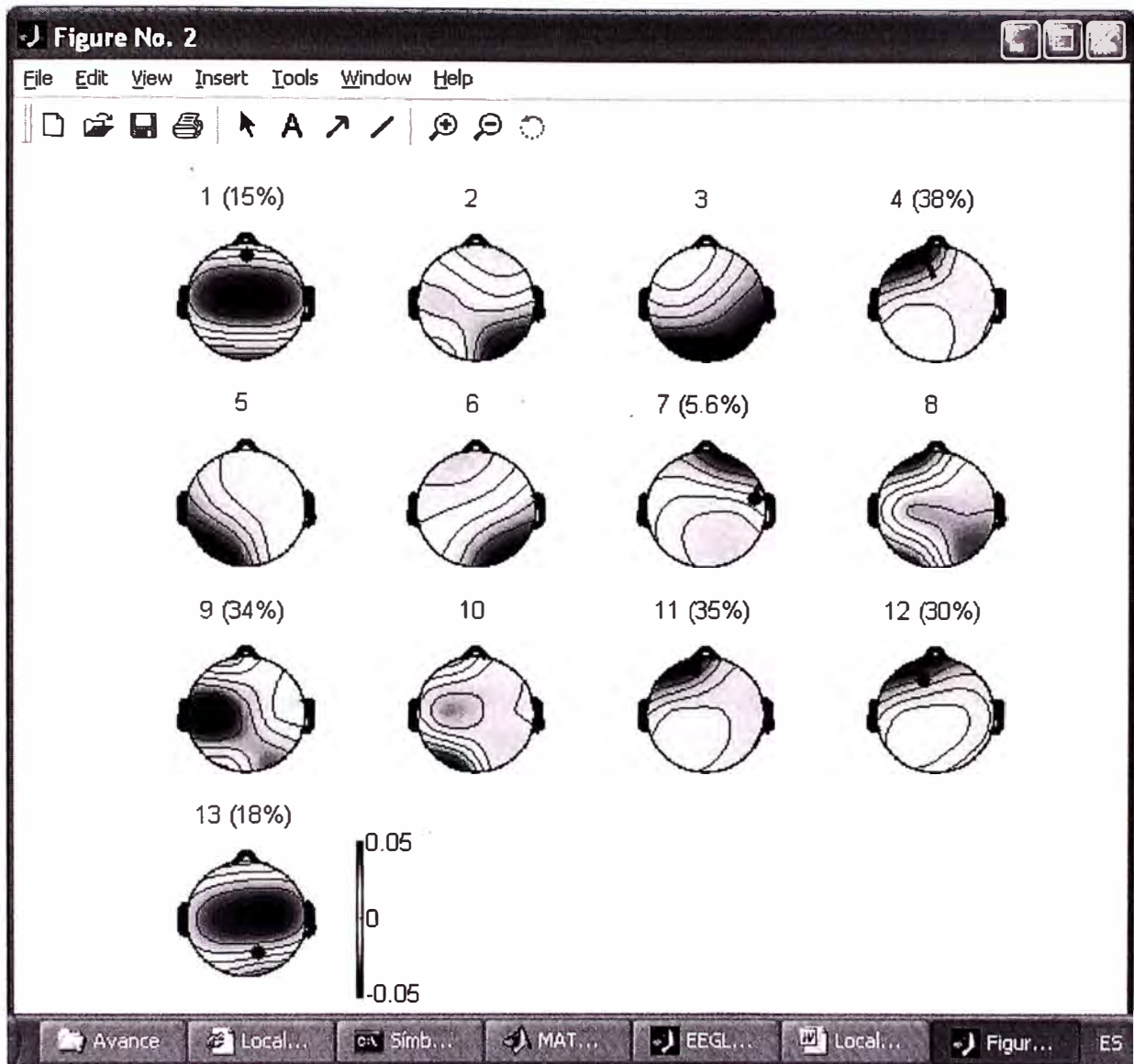


Fig. 3. 167 Gráfica de los dipolos encontrados en los canales de los 13 electrodos

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El avance en el conocimiento de las señales EEG ha incrementado su importancia debido a que se ha descubierto una serie de aplicaciones que permiten su utilización en una amplia gama de la medicina la cual está permitiendo el diagnóstico y cura de enfermedades, así como en descubrir los patrones de onda del pensamiento humano, los cuales permitirán la implementación de dispositivos que puedan ser gobernados con el pensamiento y en algunos casos reemplazar algunos órganos del cuerpo humano. En comparación de las técnicas utilizadas como la resonancia magnética, las señales EEG tienen la ventaja de que capturan los cambios de estados en el tiempo que sufre las señales que el cerebro genera ante la presencia de un estímulo y/o cambio de ánimo.
2. La definición de estándares de las coordenadas de los electrodos, los formatos de los medios de almacenamiento de datos, así como los métodos de análisis de las señales EEG, ha permitido el desarrollo de su conocimiento, con el aporte de muchas personas y diferentes entidades que han y están trabajando siguiendo las definiciones establecidas.
3. En toda investigación y desarrollo es necesario contar con una herramienta que facilite estas tareas, el EEGLAB es una de ellas, es de código abierto (open source) y de libre distribución, sus facilidades permiten procesar datos EEG de diferentes formatos, análisis gráficos 2D y 3D de: espectros, Potenciales relacionados de Eventos (ERP), Análisis de Componentes Independientes (ICA), y otros.

RECOMENDACIONES

1. El desarrollo de aplicaciones asociadas a señales EEG requiere de un conocimiento profundo del tema, por lo que se recomienda el aprendizaje y dominio de herramientas ya existentes como el EEGLab, FMRLab, etc. que facilitan dichos estudios. También es importante la participación en el Forum de EEGLAB de libre asociación que permite compartir conocimientos con especialistas y desarrolladores de alto grado de conocimiento, en el cual se podrá recibir ayuda y consejo sobre el estudio de estas señales.
2. El estudio de las señales EEG esta en apogeo debido a que se están descubriendo una serie de aplicaciones importantes y en otras se avizora como muy prometedora; por lo que su estudio traerá muchas satisfacciones en el conocimiento del pensamiento y el comportamiento del ser humano, así como en el descubrimiento de nuevas aplicaciones, desarrollo de nuevos dispositivos médicos, y muchas otras.

ANEXO A

A.1 Conversor de archivos EEG Continuos NeuroScan a ASCII

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"
#include "time.h"
#include "sethead.h"
#include "paulslib.h"

/*
  Minimal conversion program for continuous EEG files
  The output file is as follows
  First line:   nchannels adcrate nevents nsamples
  Second line:  "sample event elec_1 elec_2 .... elec_nchannels"
  Rest of file: sample event# sample_1 sample_2 .... sample_nchannels

  Be warned: check whether the machine being used is big or small endian
  Also check the size of short and int, should be 2 and 4 bytes respectively
*/

#define TRUE 1
#define FALSE 0
#define SWAPENDIAN FALSE

#define MAXEVENT 1000
int event[MAXEVENT];
int nevent = 0;

int main(argc,argv)
int argc;
char **argv;
{

```



```

int i,j,readok,foundevent,showheader = TRUE;
short sample;
short unsigned nchannels;
short unsigned adc;
int c,filesize=0;
int nsamples;
int eventoffset,offset,eventsize;
char ename[16];
int eventtype;
FILE *fptr;

/* Do we have enough arguments */
if (argc < 2) {
    fprintf(stderr,"Usage: %s cntfile [-h]\n",argv[0]);
    exit(0);
}

/* Do we want to show the header ? */
if (argc >= 3) {
    if (strcmp(argv[2],"-h") == 0)
        showheader = FALSE;
}

/* Attempt to open the continuous EEG file */
if ((fptr = fopen(argv[1],"r")) == NULL) {
    fprintf(stderr,"Unable to open file <%s>\n",argv[1]);
    exit(0);
}

/* Calculate the file size */
while ((c = fgetc(fptr)) != EOF)
    filesize++;
fseek(fptr,0L,SEEK_SET);

/* Read the number of channels */
fseek(fptr,370L,SEEK_SET);

```

```

readok = ReadUShort(fptr,&nchannels,SWAPENDIAN);
if (!readok || nchannels < 1 || nchannels > 64) {
    fprintf(stderr,"Unexpected number of channels <%d>\n",nchannels);
    exit(0);
}
if (showheader)
    printf("%d ",nchannels);

/*Read the ADC rate */
fseek(fptr,376L,SEEK_SET);
readok = ReadUShort(fptr,&adc,SWAPENDIAN);
if (!readok || adc < 1) {
    fprintf(stderr,"Unexpected error reading adc rate\n",adc);
    exit(0);
}
if (showheader)
    printf("%d ",adc);

/*Read the event table offset */
fseek(fptr,886L,SEEK_SET);
readok = ReadInt(fptr,&eventoffset,SWAPENDIAN);
if (!readok || eventoffset > filesize) {
    fprintf(stderr,"Event offset %ld is greater than the file size\n",
        eventoffset);
    exit(0);
}

/*Read the events */
fseek(fptr,eventoffset,SEEK_SET);
eventtype = fgetc(fptr);
if (eventtype < 1 || eventtype > 2) {
    fprintf(stderr,"Unexpected eventtype <%d>\n",eventtype);
    exit(0);
}
readok = ReadInt(fptr,&eventsized,SWAPENDIAN);
if (!readok || eventsized < 0) {

```

```

    fprintf(stderr,"Unexpected event size <%ld>\n",eventsize);
    exit(0);
}
if (eventtype == 1)
    nevent = eventsize / 8;
else
    nevent = eventsize / 19;
for (i=0;i<nevent;i++) {
    if (eventtype == 1)
        fseek(fp,offset + 9 + i * 8 + 4,SEEK_SET);
    else
        fseek(fp,offset + 9 + i * 19 + 4,SEEK_SET);
    readok = ReadInt(fp,&offset,SWAPENDIAN);
    if (!readok || offset < 0) {
        fprintf(stderr,"Unexpected event offset <%ld>\n",offset);
        exit(0);
    }
    event[i] = (offset - 900 - 75 * nchannels) / (nchannels * 2);
    if (event[i] < 0) {
        fprintf(stderr,"Unexpected event sample <%ld>\n",event[i]);
        exit(0);
    }
}
if (showheader)
    printf("%d ",nevent);

/* Calculate the number of expected samples */
nsamples = (eventoffset - (900L + 75 * nchannels)) / (2 * nchannels);
if (showheader)
    printf("%ld\n",nsamples);
if (nsamples < 1) {
    fprintf(stderr,"Unexpected number of samples <%ld>\n",nsamples);
    exit(0);
}

/* Read the electrode names */

```

```

if (showheader)
    printf("sample event ");
for (i=0;i<nchannels;i++) {
    fseek(fp,900L + i * 75,SEEK_SET);
    readok = ReadString(fp,ename,10);
    if (showheader)
        printf("%s ",ename);
}
if (showheader)
    printf("\n");

/*Read the samples and write results */
fseek(fp,900 + 75 * nchannels,SEEK_SET);
for (i=0;i<nsamples;i++) {

    /*The first column is the sample counter */
    printf("%ld ",i);

    /*The second column is the event flag */
    foundevent = 0;
    for (j=0;j<nevent;j++) {
        if (event[j] == i) {
            foundevent = j + 1;
            break;
        }
    }
    printf("%d ",foundevent);

    /*The remaining channels are the samples */
    for (j=0;j<nchannels;j++) {
        if (!ReadShort(fp,&sample,SWAPENDIAN)) {
            fprintf(stderr,"Unexpected end of samples\n");
            exit(0);
        }
        printf("%d ",sample);
    }
}

```

```
    printf("\n");  
}  
  
fclose(fptr);  
}  
  
#include "paulslib.c"
```

A.2 Conversor de archivos Promedios de EEG NeuroScan a ASCII

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"
#include "sethead.h"
#include <sys/types.h>
#include <time.h>
#include "/usr/people/pdb/include/paulslib.h"
#include "/usr/people/pdb/include/paulslib.proto"

/*
    Minimal conversion program for Neuroscan average files
    The output file is as follows
    First line:   nchannels adcrate nevents nsamples
    Second line:  "sample event elec_1 elec_2 .... elec_nchannels"
    Rest of file: sample event# sample_1 sample_2 .... sample_nchannels
*/

int main(argc,argv)
int argc;
char **argv;
{
    int i,j,readok,foundevent;
    float sample,scale;
    float themin=1e32,themax=-1e32;
    short unsigned nchannels;
    short unsigned adc;
    short unsigned nsamples;
    long c,filesize=0;
    char ename[16];
    FILE *fptr;
    float **data;

```

```

/*Do we have enough arguments */
if (argc < 2) {
    fprintf(stderr,"Usage: %s avgfile scale\n",argv[0]);
    exit(0);
}

/*Attempt to open the average EEG file */
if ((fptr = fopen(argv[1],"r")) == NULL) {
    fprintf(stderr,"Unable to open file <%s>\n",argv[1]);
    exit(0);
}

/*Calculate the file size */
while ((c = fgetc(fptr)) != EOF)
    filesize++;
fseek(fptr,0L,SEEK_SET);

/*Read the number of channels */
fseek(fptr,370L,SEEK_SET);
readok = ReadUShort(fptr,&nchannels,TRUE);
if (!readok || nchannels < 1 || nchannels > 64) {
    fprintf(stderr,"Unexpected number of channels <%d>\n",nchannels);
    exit(0);
}
printf("%d ",nchannels);

/*Read the ADC rate */
fseek(fptr,376L,SEEK_SET);
readok = ReadUShort(fptr,&adc,TRUE);
if (!readok || adc < 1) {
    fprintf(stderr,"Unexpected error reading adc rate\n",adc);
    exit(0);
}
printf("%d 0 ",adc);

/*Calculate the number of expected samples */

```

```

nsamples = ((filesize - 900 - nchannels*75) / nchannels - 5) / 4;
    if (nsamples < 1) {
        fprintf(stderr,"Unexpected number of samples <%ld>\n",nsamples);
        exit(0);
    }
    printf("%ld\n",(long)nsamples);

    /* Read the electrode names */
    printf("sample event ");
    for (i=0;i<nchannels;i++) {
        fseek(fp,900L + i * 75,SEEK_SET);
        readok = ReadString(fp,ename,10);
        printf("%s ",ename);
    }
    printf("\n");

    /* Allocate memory for the samples */
    data = (float **)malloc(nchannels*sizeof(float *));
    for (i=0;i<nchannels;i++)
        data[i] = (float *)malloc(nsamples*sizeof(float));

    /* Read the samples */
    fseek(fp,900 + 75 * nchannels,SEEK_SET);
    for (i=0;i<nchannels;i++) {

        /* Read past the 5 byte header */
        for (j=0;j<5;j++)
            c = fgetc(fp);

        /* Read nsamples for this channel */
        for (j=0;j<nsamples;j++) {
            if (!ReadFloat(fp,&(data[i][j]),TRUE)) {
                fprintf(stderr,"Unexpected end of samples\n");
                exit(0);
            }
        }

        themin = MIN(themin,data[i][j]);
    }

```



```

        themax = MAX(themax,data[i][j]);
    }
}

/* Write the data as ascii */
for (j=0;j<nsamples;j++) {
    printf("%ld 0 ",j);
    for (i=0;i<nchannels;i++)
        printf("%d ",(int)(10000*(data[i][j]-themin)/(themax-themin)-5000));
    printf("\n");
}

fclose(fp);
}

#include "/usr/people/pdb/include/paulslib.c"

```

A.3 Sethead.h para Archivos EEG NeuroScan

sethead.h

La siguiente estructura es una version de archive de "sethead.h", diseñado para hacer implementaciones en otras máquinas mas facilmente, particularmente para máquinas "Otros-Endian". Notar que los campos carácter es un simple byte, longs son 4 bytes, float son formato IEEE de 4 bytes y short ints son 2 bytes.

```

typedef struct {          /* Electrode structure ----- */
    char lab[10];        /* Electrode label - last bye contains NULL */
    char reference;     /* Reference electrode number */
    char skip;          /* Skip electrode flag ON=1 OFF=0 */
    char reject;        /* Artifact reject flag */
    char display;       /* Display flag for 'STACK' display */
}

```

```

char bad;          /* Bad electrode flag          */
unsigned short int n; /* Number of observations          */
char avg_reference; /* Average reference status        */
char ClipAdd;     /* Automatically add to clipboard  */
float x_coord;    /* X screen coord. for 'TOP' display */
float y_coord;    /* Y screen coord. for 'TOP' display */
float veog_wt;    /* VEOG correction weight          */
float veog_std;   /* VEOG std dev. for weight        */
float snr;        /* signal-to-noise statistic       */
float heog_wt;    /* HEOG Correction weight          */
float heog_std;   /* HEOG Std dev. for weight        */
short int baseline; /* Baseline correction value in raw ad units*/
char Filtered;   /* Toggel indicating file has be filtered */
char Fsp;        /* Extra data                       */
float aux1_wt;    /* AUX1 Correction weight          */
float aux1_std;   /* AUX1 Std dev. for weight        */
float sensitivity; /* electrode sensitivity            */
char Gain;        /* Amplifier gain                   */
char HiPass;     /* Hi Pass value                    */
char LoPass;     /* Lo Pass value                    */
unsigned char Page; /* Display page                     */
unsigned char Size; /* Electrode window display size    */
unsigned char Impedance; /* Impedance test                  */
unsigned char PhysicalChnl; /* Physical channel used          */
char Rectify;    /* Free space                       */
float calib;     /* Calibration factor               */
} ELECTLOC;

typedef struct{
char rev[20];    /* Revision string                  */
char type;      /* File type AVG=1, EEG=0          */
char id[20];    /* Patient ID                       */
char oper[20];  /* Operator ID                      */
char doctor[20]; /* Doctor ID                       */
char referral[20]; /* Referral ID                     */
char hospital[20]; /* Hospital ID                     */

```

```

char patient[20];      /* Patient name          */
short int age;        /* Patient Age           */
char sex;             /* Patient Sex Male='M', Female='F' */
char hand;           /* Handedness Mixed='M', Rt='R', lft='L' */
char med[20];        /* Medications           */
char classif[20];    /* Classification        */
char state[20];      /* Patient wakefulness   */
char label[20];      /* Session label         */
char date[10];       /* Session date string   */
char time[12];       /* Session time strin    */
float mean_age;      /* Mean age (Group files only) */
float stdev;        /* Std dev of age (Group files only) */
short int n;        /* Number in group file  */
char compfile[38];   /* Path and name of comparison file */
float SpectWinComp; /* Spectral window compensation factor */
float MeanAccuracy; /* Average respose accuracy */
float MeanLatency;  /* Average response latency */
char sortfile[46];  /* Path and name of sort file */
long NumEvents;     /* Number of events in eventable */
char compoper;      /* Operation used in comparison */
char avgmode;       /* Set during online averaging */
char review;        /* Set during review of EEG data */
short unsigned nsweeps; /* Number of expected sweeps */
short unsigned compsweeps; /* Number of actual sweeps */
short unsigned acceptcnt; /* Number of accepted sweeps */
short unsigned rejectcnt; /* Number of rejected sweeps */
short unsigned pnts; /* Number of points per waveform */
short unsigned nchannels; /* Number of active channels */
short unsigned avgupdate; /* Frequency of average update */
char domain;        /* Acquisition domain TIME=0, FREQ=1 */
char variance;      /* Variance data included flag */
unsigned short rate; /* D-to-A rate */
double scale;       /* scale factor for calibration */
char veogcorrect;   /* VEOG corrected flag */
char heogcorrect;   /* HEOG corrected flag */
char aux1correct;   /* AUX1 corrected flag */

```

```

char aux2correct;      /* AUX2 corrected flag          */
float veogtrig;       /* VEOG trigger percentage      */
float heogtrig;       /* HEOG trigger percentage      */
float aux1trig;       /* AUX1 trigger percentage      */
float aux2trig;       /* AUX2 trigger percentage      */
short int heogchnl;   /* HEOG channel number         */
short int veogchnl;   /* VEOG channel number         */
short int aux1chnl;   /* AUX1 channel number         */
short int aux2chnl;   /* AUX2 channel number         */
char veogdir;        /* VEOG trigger direction flag  */
char heogdir;        /* HEOG trigger direction flag  */
char aux1dir;        /* AUX1 trigger direction flag  */
char aux2dir;        /* AUX2 trigger direction flag  */
short int veog_n;     /* Number of points per VEOG waveform */
short int heog_n;     /* Number of points per HEOG waveform */
short int aux1_n;     /* Number of points per AUX1 waveform */
short int aux2_n;     /* Number of points per AUX2 waveform */
short int veogmaxcnt; /* Number of observations per point - VEOG */
short int heogmaxcnt; /* Number of observations per point - HEOG */
short int aux1maxcnt; /* Number of observations per point - AUX1 */
short int aux2maxcnt; /* Number of observations per point - AUX2 */
char veogmethod;     /* Method used to correct VEOG   */
char heogmethod;     /* Method used to correct HEOG   */
char aux1method;     /* Method used to correct AUX1   */
char aux2method;     /* Method used to correct AUX2   */
float AmpSensitivity; /* External Amplifier gain      */
char LowPass;        /* Toggle for Amp Low pass filter */
char HighPass;       /* Toggle for Amp High pass filter */
char Notch;          /* Toggle for Amp Notch state    */
char AutoClipAdd;    /* AutoAdd on clip               */
char baseline;       /* Baseline correct flag         */
float offstart;      /* Start point for baseline correction */
float offstop;       /* Stop point for baseline correction */
char reject;         /* Auto reject flag              */
float rejstart;      /* Auto reject start point       */
float rejstop;       /* Auto reject stop point        */

```

```

float rejmin;      /* Auto reject minimum value      */
float rejmax;     /* Auto reject maximum value      */
char  trigtype;   /* Trigger type                    */
float trigval;    /* Trigger value                   */
char  trigchnl;   /* Trigger channel                 */
short int trigmask; /* Wait value for LPT port        */
float trigisi;    /* Interstimulus interval (INT trigger) */
float trigmin;    /* Min trigger out voltage (start of pulse)*/
float trigmax;    /* Max trigger out voltage (during pulse) */
char  trigdir;   /* Duration of trigger out pulse   */
char  Autoscale; /* Autoscale on average           */
short int n2;    /* Number in group 2 (MANOVA)     */
char  dir;       /* Negative display up or down    */
float dispmin;   /* Display minimum (Yaxis)        */
float dispmax;   /* Display maximum (Yaxis)        */
float xmin;     /* X axis minimum (epoch start in sec) */
float xmax;     /* X axis maximum (epoch stop in sec) */
float AutoMin;   /* Autoscale minimum             */
float AutoMax;   /* Autoscale maximum             */
float zmin;     /* Z axis minimum - Not currently used */
float zmax;     /* Z axis maximum - Not currently used */
float lowcut;   /* Archival value - low cut on external amp*/
float highcut;  /* Archival value - Hi cut on external amp */
char  common;   /* Common mode rejection flag     */
char  savemode; /* Save mode EEG AVG or BOTH     */
char  manmode;  /* Manual rejection of incomming data */
char  ref[10];  /* Label for reference electrode   */
char  Rectify;  /* Rectification on external channel */
float DisplayXmin; /* Minimun for X-axis display    */
float DisplayXmax; /* Maximum for X-axis display    */
char  phase;    /* flag for phase computation     */
char  screen[16]; /* Screen overlay path name     */
short int CalMode; /* Calibration mode              */
short int CalMethod; /* Calibration method           */
short int CalUpdate; /* Calibration update rate      */
short int CalBaseline; /* Baseline correction during cal */

```

```

short int CalSweeps;      /* Number of calibration sweeps      */
float CalAttenuator;     /* Attenuator value for calibration    */
float CalPulseVolt;     /* Voltage for calibration pulse      */
float CalPulseStart;    /* Start time for pulse               */
float CalPulseStop;     /* Stop time for pulse               */
float CalFreq;         /* Sweep frequency                   */
char taskfile[34];     /* Task file name                    */
char seqfile[34];     /* Sequence file path name           */
char SpectMethod;     /* Spectral method                   */
char SpectScaling;    /* Scaling employed                  */
char SpectWindow;     /* Window employed                   */
float SpectWinLength; /* Length of window %                */
char SpectOrder;     /* Order of Filter for Max Entropy method */
char NotchFilter;    /* Notch Filter in or out           */
char unused[11];     /* Free space                        */
short FspStopMethod; /* FSP - Stopping mode              */
short FspStopMode;  /* FSP - Stopping mode              */
float FspFValue;    /* FSP - F value to stop terminate  */
short int FspPoint; /* FSP - Single point location      */
short int FspBlockSize; /* FSP - block size for averaging */
unsigned short FspP1; /* FSP - Start of window           */
unsigned short FspP2; /* FSP - Stop of window           */
float FspAlpha;     /* FSP - Alpha value                */
float FspNoise;     /* FSP - Signal to ratio value      */
short int FspV1;    /* FSP - degrees of freedom        */
char montage[40];   /* Montage file path name           */
char EventFile[40]; /* Event file path name            */
float fratio;       /* Correction factor for spectral array */
char minor_rev;     /* Current minor revision           */
short int eegupdate; /* How often incoming eeg is refreshed */
char compressed;    /* Data compression flag           */
float xscale;       /* X position for scale box - Not used */
float yscale;       /* Y position for scale box - Not used */
float xsize;        /* Waveform size X direction        */
float ysize;        /* Waveform size Y direction        */
char ACmode;        /* Set SYNAP into AC mode           */

```

```

unsigned char CommonChnl; /* Channel for common waveform */
char Xllcs; /* Scale tool- 'llc' flag in X direction */
char Xrange; /* Scale tool- range (ms,sec,Hz) flag X dir*/
char Yllcs; /* Scale tool- 'llc' flag in Y direction */
char Yrange; /* Scale tool- range (uV, V) flag Y dir */
float XScaleValue; /* Scale tool- value for X dir */
float XScaleInterval; /* Scale tool- interval between llcs X dir */
float YScaleValue; /* Scale tool- value for Y dir */
float YScaleInterval; /* Scale tool- interval between llcs Y dir */
float ScaleToolX1; /* Scale tool- upper left hand screen pos */
float ScaleToolY1; /* Scale tool- upper left hand screen pos */
float ScaleToolX2; /* Scale tool- lower right hand screen pos */
float ScaleToolY2; /* Scale tool- lower right hand screen pos */
short int port; /* Port address for external triggering */
long NumSamples; /* Number of samples in continuous file */
char FilterFlag; /* Indicates that file has been filtered */
float LowCutoff; /* Low frequency cutoff */
short int LowPoles; /* Number of poles */
float HighCutoff; /* High frequency cutoff */
short int HighPoles; /* High cutoff number of poles */
char FilterType; /* Bandpass=0 Notch=1 Highpass=2 Lowpass=3 */
char FilterDomain; /* Frequency=0 Time=1 */
char SnrFlag; /* SNR computation flag */
char CoherenceFlag; /* Coherence has been computed */
char ContinuousType; /* Method used to capture events in *.cnt */
long EventTablePos; /* Position of event table */
float ContinuousSeconds; /* Number of seconds to displayed per page */
long ChannelOffset; /* Block size of one channel in SYNAMIPS */
char AutoCorrectFlag; /* Autocorrect of DC values */
unsigned char DCThreshold; /* Auto correct of DC level */
} SETUP;

typedef struct {
char Teog; /* Either 1 or 2 */
long Size; /* Total length of all the events */
long Offset; /* Hopefully always 0 */

```

```
} TEEG;
```

```
typedef struct {
    unsigned short StimType; /* range 0-65535 */
    unsigned char Keyboard; /* range 0-11 corresponding to fcn keys +1 */
    char KeyPad_Accept; /* 0->3 range 0-15 bit coded response pad */
                        /* 4->7 values 0xd=Accept 0xc=Reject */
    long Offset; /* file offset of event */
} EVENT1;
```

```
typedef struct{
    unsigned short StimType; /* range 0-65535 */
    unsigned char Keyboard; /* range 0-11 corresponding to fcn keys +1 */
    char KeyPad_Accept; /* 0->3 range 0-15 bit coded response pad */
                        /* 4->7 values 0xd=Accept 0xc=Reject */
    long Offset; /* file offset of event */
    short Type;
    short Code;
    float Latency;
    char EpochEvent;
    char Accept2;
    char Accuracy;
} EVENT2;
```


A.4 Formato BKR V 2.07

Información de Encabezado (1024 Bytes)

Valores muestras (data)

Datablock: untriggered file

value 1 to nch	channel 1 sample 1	channel 2 sample 1	channel 3 sample 1	channel nch-1 sample 1	channel nch sample 1
value nch-1 to 2nch	channel 1 sample 2	channel 2 sample 2	channel 3 sample 2	channel nch-1 sample 2	channel nch sample 2
...
...
values (n-2)nch+1 to...	channel 1 sample n-1	channel 2 sample n-1	channel 3 sample n-1	channel nch-1 sample n-1	channel nch sample n-1
values (n-1)nch to n*nch	channel 1 sample n	channel 2 sample n	channel 3 sample n	channel nch-1 sample n	channel nch sample n
value n*nch+1	EOF						

Datablock: triggered file

	channel 1	channel 2	channel 3	channel nch-1	channel nch
	sample nsp	sample nsp	sample nsp	sample nsp	sample nsp
	EOF						

Header in Bytes

<u>Name</u>	<u>Datatype</u>	<u>Size</u>	<u>Offset</u>	<u>Meaning</u>
ver	USHORT	2 Byte	0	Version
nch	USHORT	2 Byte	2	Number of channels
nhz	USHORT	2 Byte	4	Sampling frequency
ntr	ULONG	4 Byte	6	Number of trials
nsp	ULONG	4 Byte	10	Number of samples per trial
cvlt	USHORT	2 Byte	14	Calibration voltage
cval	USHORT	2 Byte	16	Calibration value
code	char[4]	4 Byte	18	Electrode code
lcf	float	4 Byte	22	Lower cut-off frequency
ucf	float	4 Byte	26	Upper cut-off frequency
sref	ULONG	4 Byte	30	Start of reference period in samples
eref	ULONG	4 Byte	34	End of reference period in samples

sact	ULONG	4 Byte	38	Start of action period in samples
eact	ULONG	4 Byte	42	End of action period in samples
trg	BOOL	2 Byte	46	Flag if file is triggered
pre	ULONG	4 Byte	48	Number of samples before trigger
pst	ULONG	4 Byte	52	Number of samples after trigger
hav	BOOL	2 Byte	56	Flag if file is "horizontally" averaged
nah	ULONG	4 Byte	58	Number of averaged trials
vav	BOOL	2 Byte	62	Flag if file is "vertically" averaged
nav	USHORT	2 Byte	64	Number of averaged trials
cav	BOOL	2 Byte	66	Flag for data compression
nac	ULONG	4 Byte	68	Number of compressed samples
com	BOOL	2 Byte	72	flag: Common Average Reference
loc	BOOL	2 Byte	74	flag: Local Average Reference
lap	BOOL	2 Byte	76	flag: Laplace
wgt	BOOL	2 Byte	78	flag: Weighted Average Reference
pwr	BOOL	2 Byte	80	flag: Power
avr	BOOL	2 Byte	82	flag: Average
std	BOOL	2 Byte	84	flag: Standard deviation
bps	BOOL	2 Byte	86	flag: Bandpaß filtered
erd	BOOL	2 Byte	88	flag: ERD

sig	BOOL	2 Byte	90	flag: Significance
coh	BOOL	2 Byte	92	flag: Coherence
spc	BOOL	2 Byte	94	flag: Spectrum
conf	float	4 Byte	96	Confidence
csp	BOOL	2 Byte	100	flag: Coherence Power Spectrum
erc	BOOL	2 Byte	102	flag: ERC
ham	BOOL	2 Byte	104	flag: Hanning smoothed
ann	BOOL	2 Byte	106	flag: art. Neuronal. NW. Filter (ANN)
niu	USHORT	2 Byte	108	ANN: Number of input units
nhu	USHORT	2 Byte	110	ANN: Number of hidden units
nlc	ULONG	4 Byte	112	ANN: Number of learn cycles
reg	float	4 Byte	116	ANN: Regression
lco	float	4 Byte	120	ANN: Learn coefficient
epo	USHORT	2 Byte	124	ANN: Epoch
rel	BOOL	2 Byte	126	flag: ERC in relative values
wnd	USHORT	2 Byte	128	Window type
kal	BOOL	2 Byte	130	flag: Kalman filtered
cwt	BOOL	2 Byte	132	flag: cont. Wavelet transformed
cwt_fmin	float	4 Byte	134	lower frequency, Wavelet
cwt_fmax	float	4 Byte	138	upper frequency, Wavelet

scales	USHORT	2 Byte	142	Number of frequency bands, Wavelet
cwt_fe	float	4 Byte	146	freq. for $\Delta t = \Delta f = 1/2 (\Delta)^{1/2}$ Wavelet
cwt_start	ULONG	4 Byte	150	Start sample for Wavelet
--NULL--	----	----	154	---- Offset to 512 Byte ----
eletype[1]	UCHAR	1 Byte	512	Electrode 1: Signal type (e.f. EEG)
elenum[1]	UCHAR	1 Byte	513	Electrode 1: Channel nr. for same signal
ref[1]	float	4 Byte	514	Reference value for Channel 1
...	etc.
eletyp[nch]	UCHAR	1 Byte	1018	Electrode nch: Signal type (e.f. EEG)
elenum[nch]	UCHAR	1 Byte	1019	Electrode nch: Channel nr. for same signal
ref[nch]	float	4 Byte	1020	Reference value for Channel nch

A.5 Formato de archivo Westmead EEG

Description	Example	Source example
<p>First line</p> <p>It consists of two items, a version number and offset. The offset is the number of 512 bytes chunks in the header.</p> <p>The current format descriptor is EEG2.</p> <p>The previous EEG1 format differs only in that the binary data was stored in a 1's complement offset format.</p>	<p>EEG2 15</p>	<pre>char ident[32]; int offset; FILE *fptr; if (fscanf(fptr,"%s %ld",ident,&offset) != 2) { report unexpected result }</pre>
<p>Header</p> <p>"offset" * 512 in length</p> <p>This consist of a number of variables providing additional quantitative and qualitative values. Most are obvious from their</p>	<p>[ProgramType] desc=Identifies acquisition program dimension=1 1 weight+value= 1 aep</p> <p>[PatName] desc=Subject's name dimension=1 1</p>	<p>There are many ways to parse the desired fields from this header. For example to extract the number of channels.</p> <pre>int found; int nchannels; char s[64]; rewind(fptr); found = FALSE; while (fscanf(fptr,"%s",s) == 1) { if (strstr(s,"[NumChans]") != NULL) {</pre>

<p>descriptors in square brackets [].</p>	<p>weight+value= 1 _____</p>	<pre>SkipUntil(fptr, '\n'); SkipUntil(fptr, '\n'); SkipUntil(fptr, '\n'); SkipUntil(fptr, '\n');</pre>
<p>The ones of particular interest are:</p>	<p>[PatID] desc=Subject's ID</p>	<pre>if ((fscanf(fptr, "%s %d", s, &nchannels) != 2) { Handle read error }</pre>
<p>[NumChans] [Label] [SamRate]</p>	<p>dimension=1 1 weight+value= 1 MS044</p>	<pre>found = TRUE; break; }</pre>
<p>Lines are terminated with <CR><LF></p>	<p>[StartTime] desc=Start time of study</p>	<pre>if (!found) { Handle case where the number of channels wasn't found }</pre>
<p>In detail, each variable is represented in the header by:</p>	<p>dimension=1 1 weight+value= 1 17 Apr 96</p>	<pre>if (nchannels < 1 nchannels > 64) { Handle unexpected number of channels }</pre>
<p>1</p>	<p>13:16:51</p>	
<p>The variable's identifying name. This consists of as many as 11 characters enclosed by square brackets. The left bracket must be in the first column. In reading the variable's name, leading and trailing blanks are removed, case is ignored, and control characters are removed.</p>	<p>[NumChans] desc=Number of data channels dimension=1 1 weight+value= 1 32 [NumSegs] desc=Number of data segments dimension=1 1 weight+value= 1 1</p>	
<p>2 A description of the</p>	<p>[DataType] desc=Identifies</p>	

likely correct		
defaults set by any program that reads these data files.	[Label] desc=Identifies channel dimension=32 1	
5 An introduction to the array list: "weight+value=3D".	weight+value= 1 Fp1 1 Fp2 1 F7 1 F3	
6 The array of weights and values. There should be dim1 x dim2 lines. The values for segment 0 are listed first, then for segment 1, and so on. Each line consists of a weight and a value. The weight may be 0, in which case, no value is present or necessary. Otherwise the weight is >0 and a value is specified. The value may be of any of the various types recognised by this set of programs - see item 4. There is a function decode() that decodes any of	1 Fz 1 F4 1 F8 1 A1 1 T3 1 C3 1 Cz 1 C4 1 T4 1 A2 1 T5 1 P3 1 Pz 1 P4 1 T6 1 O1 1 O2 1 EOG1 1 EOGv 1 EOG2 1 CedX 1 CedY 1 CedA 1 CedS 1 Resp	

these standard	1 ECG	
formats. Concerning	1 SCR	
weights, for string	1 C31	
and boolean		
variables, there is	[HWGain]	
no useful distinction	desc=Total HW	
between weights of	gain	
1 and >1. However	dimension=32 1	
for numeric	weight+value=	
variables, the value	1 200	
of weight is	1 200	
important when	1 200	
combining data sets.	1 200	
Then it is necessary	1 200	
to form weighted	1 200	
averages, hence the	1 200	
importance of	1 200	
including a weight	1 200	
with each value.	1 200	
	1 200	
The variables may	1 200	
be arranged in any	1 200	
order within the	1 200	
header, with the	1 200	
exception of the two	1 200	
variables	1 200	
NumChans and	1 200	
NumSegs that	1 200	
MUST precede any	1 200	
multidimensional	1 200	
variable. This is	1 200	
because with such	1 200	
array-type variables,	1 200	
one or both of the	1 1	
two dimensions	1 1	
specified in item 3		

1 DC

0

0

0

0

1 DC

1 DC

1 DC

0

[HCFilt]

desc=High Cut

filter setting

dimension=32 1

weight+value=

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000

1 50.0000
1 50.0000
1 50.0000
1 50.0000
1 50.0000
1 50.0000
1 50.0000
1 250.0000
1 50.0000
1 50.0000
1 50.0000
1 250.0000

[HWUnit]

desc=Unit

associated with

raw signal

dimension=32 1

weight+value=

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV
1 uV
1 uV
1 uV
1 uV
1 uV
1 uV
1 uV
1 CU
1 CU
1 CU
1
1 uV
1 uV
1 mV
1

[PlotRow]
desc=Row in
which to plot data
dimension=32 1
weight+value=
1 0
1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
1 11
1 12

1 13
1 14
1 15
1 16
1 17
1 18
1 19
1 20
1 21
1 22
1 23
1 24
1 25
1 26
1 27
1 28
1 29
1 30
1 31

[PlotCol]
desc=Colour with
which to plot data
dimension=32 1
weight+value=
1 10
1 10
1 14
1 10
1 12
1 10
1 14
1 10
1 14
1 10

1 12

1 10

1 14

1 10

1 14

1 10

1 12

1 10

1 14

1 10

1 10

1 2

1 2

1 2

1 11

1 11

1 11

1 11

1 2

1 4

1 10

1 15

[PlotGain]

desc=Plot scale,

as a power of 2

dimension=32 1

weight+value=

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 14

1 10

0

1 9

1 10

1 12

0

[Positive]

desc=Up, if data
is to be plotted

+ve up

dimension=32 1

weight+value=

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Down

1 Up

1 Up

1 Up

0

1 Up

1 Up

1 Up

0

[Clipping]

desc=Yes, if plot

clipping required

dimension=32 1

weight+value=

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

1 No

0

1 No

1 No

1 No

0

[CalFac]

desc=Scale

0

[CalUnit]

desc=Unit

associated with

scaled data

dimension=32 1

weight+value=

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 uV

1 deg

1 deg

1 deg

0
1 uV
1 uV
1 ms
0

[SamRate]
desc=Sampling
rate in Hz
dimension=1 1
weight+value=
1 250.00000000

[RefSite]
desc=Reference
site for EEG
channels
dimension=1 1
weight+value=
1 LEL

[PatComments]
desc=Any
comments
dimension=1 1
weight+value=
1

[TagF1]
desc=Meaning of
F1 marker
dimension=1 1
weight+value=
1

	<p>[TagF2] desc=Meaning of F2 marker dimension=1 1 weight+value= 1</p> <p>[TagF3] desc=Meaning of F3 marker dimension=1 1 weight+value= 1</p> <p>[TagF4] desc=Meaning of F4 marker dimension=1 1 weight+value= 1</p>	
<p>Samples - Binary data</p> <p>Each sample is stored as a 2 byte signed integers (2's complement) in "little-endian" order (Byte swapping may be required). To calculate the total number of samples use:</p> <p><code>nsamples = (size - 512*offset) /</code></p>	<p>Example inappropriate</p>	<pre>fseek(fptr,512L*offset,SEEK_SET); for (i=0;i<nsamples;i++) { for (j=0;j<nchannels;j++) { if (!ReadShort(fptr,&samples[i].electrode[j],TRUE)) { Handle read error } } } /* Read a possibly byte swapped short */ int ReadShort(FILE *fptr,short n,int swap) {</pre>

<p>(2*nchannels)</p> <p>where "size" is the total file size. ie: the file is made up of "offset" blocks, each 512 bytes in length. This is followed by "nsamples", each one made up of "nchannels" 2 byte samples.</p> <p>Typically, the last channel (C31) is for the event markers.</p>		<pre> unsigned char *cptr,tmp; if (fread(n,2,1,fptr) != 1) return(FALSE); if (swap) { cptr = (unsigned char *)n; tmp = cptr[0]; cptr[0] = cptr[1]; cptr[1] =tmp; } return(TRUE); } </pre>
---	--	--

A.6 Conversor de formato de archivo Westmead EEG a ASCII

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

/*
Minimal conversion program of Westmead EEG files to ascii
The output file is as follows
First line:   nchannels adcrate nevents nsamples
Second line:  "sample event elec_1 elec_2 .... elec_nchannels"
Rest of file: sample event# sample_1 sample_2 .... sample_nchannels

```

Note: this program is more complicated than it would need to be if it was running from a PeeCee (or at least from a machine with the same byte ordering. In that case the binary data could be read with a single "fread()". This code is primarily for UNIX machines with the opposite byte ordering to PeeCees, for example, an SGI.

```

*/

/* Prototypes */
int ReadShort(FILE *, short int *, int);
void SkipLine(FILE *);

#define TRUE 1
#define FALSE 0

int main(int argc, char **argv)
{
    int c,i,j,readok,offset,nevent=0,nchannels,adc;
    short event,channels[200];
    long filesize=0;
    long nsamples;
    char s[128],label[128];
    FILE *fptr;

    /* Do we have enough arguments */
    if (argc < 2) {
        fprintf(stderr,"Usage: %s filename\n",argv[0]);
        exit(0);
    }

    /* Attempt to open the file */
    if ((fptr = fopen(argv[1],"r")) == NULL) {
        fprintf(stderr,"Unable to open file <%s>\n",argv[1]);
        exit(0);
    }

    /*
    Calculate the file size.
    There are better "non portable" ways of doing this
    */
    while ((c = fgetc(fptr)) != EOF)
        filesize++;

```

```

/*
    Check the header
    label should be "EEG2"
*/
rewind(fp_ptr);
for (i=0;i<32;i++) {
    s[i] = fgetc(fp_ptr);
    s[i+1] = '\0';
}
if (sscanf(s,"%s %d",label,&offset) != 2) {
    fprintf(stderr,"Failed to read identifier and offset\n");
    exit(0);
}

/* Read the number of channels */
rewind(fp_ptr);
nchannels = 0;
while (fscanf(fp_ptr,"%s",s) == 1) {
    if (strstr(s,"NumChans") != NULL) {
        SkipLine(fp_ptr);
        SkipLine(fp_ptr);
        SkipLine(fp_ptr);
        SkipLine(fp_ptr);
        fscanf(fp_ptr,"%s %d",s,&nchannels);
        break;
    }
}
nchannels--; /* Last channel holds the events */
if (nchannels < 10) {
    fprintf(stderr,"Unexpected number of channels <%d>\n",nchannels);
    exit(0);
}
printf("%d ",nchannels);

/* Read the ADC rate */

```

```

rewind(fptr);
adc = 0;
while (fscanf(fptr,"%s",s) == 1) {
    if (strstr(s,"[SamRate]") != NULL) {
        SkipLine(fptr);
        SkipLine(fptr);
        SkipLine(fptr);
        SkipLine(fptr);
        fscanf(fptr,"%s %d",s,&adc);
        break;
    }
}
if (adc < 10) {
    fprintf(stderr,"Unexpected ADC rate <%d>\n",adc);
    exit(0);
}
printf("%d ",adc);

/* Determine the number of events, not done correctly! */
printf("%d ",nevent);

/* Calculate the number of expected samples */
nsamples = (filesize - 512*offset) / (2*(nchannels + 1));
printf("%ld\n",nsamples);
if (nsamples < 1) {
    fprintf(stderr,"Unexpected number of samples <%ld>\n",nsamples);
    exit(0);
}

/* Read the electrode names */
printf("sample event ");
rewind(fptr);
while (fscanf(fptr,"%s",s) == 1) {
    if (strstr(s,"[Label]") != NULL) {
        SkipLine(fptr);
        SkipLine(fptr);

```

```

SkipLine(fptr);
SkipLine(fptr);
for (i=0;i<nchannels;i++) {
    if (fscanf(fptr,"%s %s",s,label) != 2) {
        fprintf(stderr,"Error reading channel names\n");
        exit(0);
    }
    printf("%s ",label);
}
break;
}
}
printf("\n");

/* Read past the header */
rewind(fptr);
for (i=0;i<512*offset;i++)
    c = fgetc(fptr);

/* Read the samples and write results */
for (i=0;i<nsamples;i++) {

    /* The first column is the sample counter */
    printf("%ld ",i);

    for (j=0;j<nchannels;j++) {
        if (!ReadShort(fptr,&channels[j],TRUE)) {
            fprintf(stderr,"Error reading the %ldth sample\n",i);
            exit(0);
        }
    }
    if (!ReadShort(fptr,&event,TRUE)) {
        fprintf(stderr,"Error reading the %ldth event\n",i);
        exit(0);
    }
}

```

```

/* The second column is the event flag */
printf("%d ",event);

/* The remaining channels are the samples */
for (j=0;j<nchannels;j++)
    printf("%d ",channels[j]);

printf("\n");
}

fclose(fp);
}

void SkipLine(FILE *fp)
{
    int c;

    while ((c = fgetc(fp)) != '\n' && c != EOF)
        ;
}

/*
Change this depending on whether you need byte swapping or not.
The following work on SGI and machines with the opposite byte
ordering to a PeeCee.
*/
int ReadShort(FILE * fp,short int *n,int swap)
{
    unsigned char *cptr,tmp;

    if (fread(n,2,1,fp) != 1)
        return(FALSE);
    if (swap) {
        cptr = (unsigned char *)n;
        tmp = cptr[0];
        cptr[0] = cptr[1];
    }
}

```

```
    cptr[1] = tmp;  
}  
return(TRUE);  
}
```

A.7 Reporte en el Matlab a la ejecución del RUNICA en el EEGLab

Input data size [13,208200] = 13 channels, 208200 frames/nFinding 13 ICA components using logistic ICA.

Decomposing 1231 frames per ICA weight $((169)^2 = 208200$ weights, Initial learning rate will be 0.001, block size 61.

Learning rate will be multiplied by 0.9 whenever angledelta \geq 60 deg.

More than 32 channels: default stopping weight change 1E-7

Training will end when wchange $<$ 1e-006 or after 512 steps.

Online bias adjustment will be used.

Removing mean of each channel ...

Final training data range: -217.309 to 212.692

Computing the sphering matrix...

Starting weights are the identity matrix ...

Sphering the data ...

Beginning ICA training ...

Lowering learning rate to 0.0009 and starting again.

Lowering learning rate to 0.00081 and starting again.

Lowering learning rate to 0.000729 and starting again.

Lowering learning rate to 0.0006561 and starting again.

Lowering learning rate to 0.00059049 and starting again.

Lowering learning rate to 0.000531441 and starting again.

Lowering learning rate to 0.000478297 and starting again.

Lowering learning rate to 0.000430467 and starting again.

Lowering learning rate to 0.00038742 and starting again.

Lowering learning rate to 0.000348678 and starting again.

Lowering learning rate to 0.000313811 and starting again.

step 1 - lrate 0.000314, wchange 486.24936858, angledelta 0.0 deg

Lowering learning rate to 0.00028243 and starting again.

step 1 - lrate 0.000282, wchange 316.85717655, angledelta 0.0 deg

step 2 - lrate 0.000282, wchange 639.93446462, angledelta 0.0 deg

step 3 - lrate 0.000282, wchange 728.19869296, angledelta 93.9 deg

step 4 - lrate 0.000254, wchange 552.88136218, angledelta 131.9 deg

step 5 - lrate 0.000229, wchange 251.28154924, angledelta 81.3 deg

step 6 - lrate 0.000206, wchange 506.38496638, angledelta 141.7 deg

step 7 - lrate 0.000185, wchange 256.08288022, angledelta 116.6 deg
step 8 - lrate 0.000167, wchange 639.22176699, angledelta 128.8 deg
step 9 - lrate 0.000150, wchange 999.50836668, angledelta 131.8 deg
step 10 - lrate 0.000135, wchange 780.38004168, angledelta 138.9 deg
step 11 - lrate 0.000122, wchange 54.77617439, angledelta 106.3 deg
step 12 - lrate 0.000109, wchange 298.71023433, angledelta 104.2 deg
step 13 - lrate 0.000098, wchange 71.83108039, angledelta 96.9 deg
step 14 - lrate 0.000089, wchange 83.84975920, angledelta 121.4 deg
step 15 - lrate 0.000080, wchange 50.78324363, angledelta 134.4 deg
step 16 - lrate 0.000072, wchange 19.03450751, angledelta 107.0 deg
step 17 - lrate 0.000065, wchange 24.18945306, angledelta 117.5 deg
step 18 - lrate 0.000058, wchange 10.36536226, angledelta 150.4 deg
step 19 - lrate 0.000052, wchange 69.13648145, angledelta 95.5 deg
step 20 - lrate 0.000047, wchange 43.20779162, angledelta 145.8 deg
step 21 - lrate 0.000042, wchange 47.37857568, angledelta 97.5 deg
step 22 - lrate 0.000038, wchange 26.92098149, angledelta 139.1 deg
step 23 - lrate 0.000034, wchange 16.36933658, angledelta 92.6 deg
step 24 - lrate 0.000031, wchange 6.06034009, angledelta 141.3 deg
step 25 - lrate 0.000028, wchange 7.82763234, angledelta 54.9 deg
step 26 - lrate 0.000028, wchange 17.01222387, angledelta 135.7 deg
step 27 - lrate 0.000025, wchange 2.03034074, angledelta 135.1 deg
step 28 - lrate 0.000023, wchange 8.05168224, angledelta 57.2 deg
step 29 - lrate 0.000023, wchange 9.23579371, angledelta 68.5 deg
step 30 - lrate 0.000020, wchange 8.13656572, angledelta 157.8 deg
step 31 - lrate 0.000018, wchange 13.23783488, angledelta 70.9 deg
step 32 - lrate 0.000016, wchange 12.98013609, angledelta 166.0 deg
step 33 - lrate 0.000015, wchange 5.15577280, angledelta 119.0 deg
step 34 - lrate 0.000013, wchange 0.36003688, angledelta 94.2 deg
step 35 - lrate 0.000012, wchange 2.21042923, angledelta 92.1 deg
step 36 - lrate 0.000011, wchange 2.90907187, angledelta 133.6 deg
step 37 - lrate 0.000010, wchange 0.72904709, angledelta 124.1 deg
step 38 - lrate 0.000009, wchange 0.59837893, angledelta 111.0 deg
step 39 - lrate 0.000008, wchange 0.72611366, angledelta 132.2 deg
step 40 - lrate 0.000007, wchange 0.86194650, angledelta 106.7 deg
step 41 - lrate 0.000006, wchange 1.03204502, angledelta 151.1 deg
step 42 - lrate 0.000006, wchange 0.57871484, angledelta 92.3 deg

step 43 - lrate 0.000005, wchange 0.64385801, angledelta 145.0 deg
step 44 - lrate 0.000005, wchange 0.27693213, angledelta 87.7 deg
step 45 - lrate 0.000004, wchange 0.17540071, angledelta 138.9 deg
step 46 - lrate 0.000004, wchange 0.40757143, angledelta 92.3 deg
step 47 - lrate 0.000003, wchange 0.54609572, angledelta 140.0 deg
step 48 - lrate 0.000003, wchange 0.26605505, angledelta 120.6 deg
step 49 - lrate 0.000003, wchange 0.19055358, angledelta 130.5 deg
step 50 - lrate 0.000002, wchange 0.05288135, angledelta 111.4 deg
step 51 - lrate 0.000002, wchange 0.02413290, angledelta 114.0 deg
step 52 - lrate 0.000002, wchange 0.07666706, angledelta 114.9 deg
step 53 - lrate 0.000002, wchange 0.09468198, angledelta 136.2 deg
step 54 - lrate 0.000002, wchange 0.02560958, angledelta 110.0 deg
step 55 - lrate 0.000001, wchange 0.02128924, angledelta 107.7 deg
step 56 - lrate 0.000001, wchange 0.02782287, angledelta 96.1 deg
step 57 - lrate 0.000001, wchange 0.01922948, angledelta 128.8 deg
step 58 - lrate 0.000001, wchange 0.03023375, angledelta 80.5 deg
step 59 - lrate 0.000001, wchange 0.01584976, angledelta 83.8 deg
step 60 - lrate 0.000001, wchange 0.01982266, angledelta 120.0 deg
step 61 - lrate 0.000001, wchange 0.02174509, angledelta 100.1 deg
step 62 - lrate 0.000001, wchange 0.00732907, angledelta 112.1 deg
step 63 - lrate 0.000001, wchange 0.01736306, angledelta 142.9 deg
step 64 - lrate 0.000001, wchange 0.02468640, angledelta 144.8 deg
step 65 - lrate 0.000001, wchange 0.01184110, angledelta 131.2 deg
step 66 - lrate 0.000000, wchange 0.02218807, angledelta 139.9 deg
step 67 - lrate 0.000000, wchange 0.01034890, angledelta 144.7 deg
step 68 - lrate 0.000000, wchange 0.00444673, angledelta 144.0 deg
step 69 - lrate 0.000000, wchange 0.00482514, angledelta 146.7 deg
step 70 - lrate 0.000000, wchange 0.00208246, angledelta 112.2 deg
step 71 - lrate 0.000000, wchange 0.00157887, angledelta 117.4 deg
step 72 - lrate 0.000000, wchange 0.00199462, angledelta 124.9 deg
step 73 - lrate 0.000000, wchange 0.00166656, angledelta 83.3 deg
step 74 - lrate 0.000000, wchange 0.00128046, angledelta 107.7 deg
step 75 - lrate 0.000000, wchange 0.00056990, angledelta 117.9 deg
step 76 - lrate 0.000000, wchange 0.00075802, angledelta 123.5 deg
step 77 - lrate 0.000000, wchange 0.00130704, angledelta 137.7 deg
step 78 - lrate 0.000000, wchange 0.00129985, angledelta 125.9 deg

step 79 - lrate 0.000000, wchange 0.00068295, angledelta 134.5 deg
 step 80 - lrate 0.000000, wchange 0.00041333, angledelta 103.0 deg
 step 81 - lrate 0.000000, wchange 0.00057732, angledelta 129.9 deg
 step 82 - lrate 0.000000, wchange 0.00026404, angledelta 136.3 deg
 step 83 - lrate 0.000000, wchange 0.00018106, angledelta 112.5 deg
 step 84 - lrate 0.000000, wchange 0.00019216, angledelta 116.8 deg
 step 85 - lrate 0.000000, wchange 0.00012319, angledelta 116.4 deg
 step 86 - lrate 0.000000, wchange 0.00006517, angledelta 129.5 deg
 step 87 - lrate 0.000000, wchange 0.00004882, angledelta 119.1 deg
 step 88 - lrate 0.000000, wchange 0.00005947, angledelta 100.4 deg
 step 89 - lrate 0.000000, wchange 0.00004194, angledelta 97.0 deg
 step 90 - lrate 0.000000, wchange 0.00003886, angledelta 121.6 deg
 step 91 - lrate 0.000000, wchange 0.00002573, angledelta 136.8 deg
 step 92 - lrate 0.000000, wchange 0.00002522, angledelta 128.4 deg
 step 93 - lrate 0.000000, wchange 0.00000819, angledelta 128.6 deg
 step 94 - lrate 0.000000, wchange 0.00000941, angledelta 126.6 deg
 step 95 - lrate 0.000000, wchange 0.00001131, angledelta 110.8 deg
 step 96 - lrate 0.000000, wchange 0.00000543, angledelta 123.3 deg
 step 97 - lrate 0.000000, wchange 0.00000284, angledelta 78.2 deg
 step 98 - lrate 0.000000, wchange 0.00000198, angledelta 88.2 deg
 step 99 - lrate 0.000000, wchange 0.00000113, angledelta 127.8 deg
 step 100 - lrate 0.000000, wchange 0.00000086, angledelta 77.1 deg
 Inverting negative activations: -1 2 -3 -4 5 6 7 -8 9 10 11 -12 13

Sorting components in descending order of mean projected variance ...

1 2 3 4 5 6 7 8 9 10 11 12 13

Use menu "Display > Hide event duration" to hide colored regions representing event duration

A.8 Funciones de los menus de EEGLAB

EEGLAB menus

En EEGLAB, todas las opciones del menú pueden ser invocados como funciones independientes, a continuación se indica la correspondencia con la función respectiva:

File

Import data

From ASCII/float file or Matlab array -- pop_editset()

From EGI .RAW file -- pop_readegi()

From segmented EGI .RAW file -- pop_readegi()

From BCI2000 ASCII file -- pop_loadbci()

From Snapmaster .SMA file -- pop_snapread()

From .BDF and Biosemi .EDF file -- pop_readedf()

From Neuroscan .CNT file -- pop_loadcnt()

From Neuroscan .EEG file -- pop_loadeeg()

From ERPSS .RAW or .RDF file -- pop_read_erpss()

Import epoch info

From Matlab array or ASCII file -- pop_importepoch()

From Neuroscan .DAT file -- pop_loaddat()

Import event info

From Matlab array or ASCII file -- pop_importevent()

From data channel -- pop_chanevent()

From Presentation .LOG file -- pop_importpres()

Export

Data and ICA activity to text file -- pop_export()

Weight matrix to text file -- pop_expica()

Inverse weight matrix to text file -- pop_expica()

Load existing dataset -- pop_loadset()

Save current dataset -- pop_saveset()

Save datasets -- pop_saveset()

Clear dataset(s) -- pop_delset()

Maximize memory -- pop_editoptions()

Save history -- pop_saveh()

Quit

Edit

Dataset info -- [pop_editset\(\)](#)
 Event fields -- [pop_editeventfield\(\)](#)
 Event values -- [pop_editeventvals\(\)](#)
 About this dataset -- [pop_comments\(\)](#)
 Channel locations -- [pop_chanedit\(\)](#)
 Select data -- [pop_select\(\)](#)
 Select epoch/events -- [pop_selectevent\(\)](#)
 Copy current dataset -- [pop_copyset\(\)](#)
 Append datasets -- [pop_mergeset\(\)](#)
 Delete dataset(s) -- [pop_delset\(\)](#)

Tools

Change sampling rate -- [pop_resample\(\)](#)
 Filter the data -- [pop_eegfilt\(\)](#)
 Re-reference -- [pop_reref\(\)](#)
 Reject continuous data -- [pop_eegplot\(\)](#)
 Extract epochs -- [pop_epoch\(\)](#)
 Remove baseline -- [pop_rmbase\(\)](#)

Reject data epochs

Reject data (all methods) -- [pop_rejmenu\(\)](#)
 Reject by inspection -- [pop_eegplot\(\)](#)
 Reject extreme values -- [pop_eegthresh\(\)](#)
 Reject flat line data -- [pop_rejtrend\(\)](#)
 Reject by probability -- [pop_jointprob\(\)](#)
 Reject by kurtosis -- [pop_rejkurt\(\)](#)
 Reject by spectra -- [pop_rejspec\(\)](#)
 Export marks to ICA reject
 Reject marked epochs -- [pop_rejepoch\(\)](#)

 Run ICA -- [pop_runica\(\)](#)
 Remove components -- [pop_subcomp\(\)](#)

Reject using ICA

Reject components by map -- pop_selectcomps()

Reject data (all methods) -- pop_rejmenu()

Reject by inspection -- pop_eegplot()

Reject extreme values -- pop_eegthresh()

Reject flat line activity -- pop_rejtrend()

Reject by probability -- pop_jointprob()

Reject by kurtosis -- pop_rejkurt()

Reject by spectra -- pop_rejspec()

Export marks to data reject

Reject marked epochs -- pop_rejepoch()

... (Custom plugin menus)

Plot

Channel locations

By name -- topoplot()

By number -- topoplot()

EEG data (scroll) -- pop_eegplot()

Channel spectra and maps -- pop_spectopo()

Component properties -- pop_prop()

Channel ERP image -- pop_erpimage()

Channel ERPs

With scalp maps -- pop_timtopo()

In scalp array -- pop_plottopo()

In rect. array -- pop_plotdata()

ERP maps series

In 2-D -- pop_topoplot()

In 3-D -- pop_headplot()

Plot/Compare ERPs -- pop_compareerps()

Component activations (scroll) -- pop_eegplot()

Component spectra and maps -- pop_spectopo()

Component maps

In 2-D -- pop_topoplot()

In 3-D -- pop_headplot()

Component properties -- pop_prop()

Component ERP image -- pop_erpimage()

Component ERPs

With component maps -- pop_envtopo()

With comp. maps (compare) -- pop_envtopo()

In rectangular array -- pop_plotdata()

Data statistics

Channel statistics -- pop_signalstat()

Component statistics -- pop_signalstat()

Event statistics -- pop_eventstat()

Time-frequency transforms

Channel time-frequency -- pop_timef()

Channel cross-coherence -- pop_crossf()

Component time-frequency -- pop_timef()

Component cross-coherence -- pop_crossf()

A.9 Funciones de EEGLAB en orden alfabético**EEGLAB functions in alphabetical order**

Las funciones son organizadas en tres grupos:

Admin functions: eeglab() y sus funciones de menú.

Interactive pop-functions: Funciones Matlab con ventanas de interface gráfica de funciones emergentes llamadas desde los menus EEGLAB. Estas funciones pueden ser utilizadas desde la linea de comandos del Matlab o en scripts o funciones macros de Matlab. Estos proveen al EEGLAB una interface gráfica para las funciones de procesamiento de señales

Signal-processingfunctions: Funciones Matlab invocados por la interface gráfica EEGLAB. Usuarios experimentados en Matlab pueden invocarlos directamente desde sus scripts o desde Matlab.

Admin functions

eeg_checkset check the consistency of fields of an EEG dataset

eeg_global declare global eeglab variables. These variables are used only as global by the main function eeglab, the function pop_rejmenu and the history h function.

eeg_helpadmin Help file for eeglab

eeg_helphelp How to use eeglab help.

eeg_helpmenu Call the help file for eeglab menus

eeg_helppop Help file for eeglab

eeg_helpsigproc Help file for eeglab

eeg_hist history for eeglab dataset.

eeg_options eeglab option script

eeg_retrieve Retrieve an EEG dataset from the variable containing all datasets (standard:ALLEEG).

eeg_store store a dataset into the variable containing all datasets

errordlg2 same as errordlg for eeglab

finputcheck check matlab function { 'key', 'val' } inputs

gethelpvar convert a Matlab m-file help-message header into out variables

getkeyval get variable value from a 'key', 'val' sequence string.

gettext This function prints a dialog box on screen and waits for the user to enter a string. There is a cancel button which returns a value of [].

h history function.

inputdlg2 inputdlg function clone with coloring and help for eeglab.

inputgui A comprehensive gui automatic builder. This function helps to create GUI very quickly without bothering about the positions of the elements. After creating a geometry, elements just place themselves in the predefined locations. It is especially useful for figures in which you intend to put text buttons and descriptions.

is_sccn returns 1 if computer is located at SCCN (Swartz Center for computational Neuroscience) and 0 otherwise

listdlg2 listdlg function clone with coloring and help for eeglab.

pop_delset Delete a dataset from the variable containing all datasets.

pop_editoptions Edit memory-saving eeglab options stored in a file 'eeg_options.m'. With no argument, pop up a window to allow the user to set/unset these options. Store user choices in a new 'eeg_options.m' file in the working directory.

pop_rejmenu Main menu for rejecting trials in an EEG dataset

pophelp Same as matlab hthelp but does not crash under windows.

questdlg2 questdlg function clone with coloring and help for eeglab.

supergui a comprehensive gui automatic builder. This function help to create GUI very fast without bothering about the positions of the elements. After creating a geometry, elements just place themselves into the predefined locations. It is especially usefull for figure where you intend to put text button and descriptions.

vararg2str transform arguments into string for evaluation using the eval() command

warndlg2 same as warndlg for eeglab

Interactive pop_functions

eeg_context returns a matrix giving, for each event of specified ("target") type(s), the latency (in ms) to the Nth preceding and/or following urevents (if any) of specified ("neighbor") type(s). Can return the target event and urevent numbers, the neighbor urevent numbers, and the values of a

specified urevent field for each of the neighbor urevents. Uses the EEG.urevent structure, plus EEG.event().urevent pointers to it. For use in event handling scripts.

eeg` eegrej reject portion of continuous data in an eeglab dataset

eeg_emptyset Initialize an EEG dataset structure with default values.

eeg_epochformat Convert the epoch information of a dataset from struct to array or vice versa.

eeg_eventformat Convert the event information of a dataset from struct to array or vice versa.

eeg_eventhist return or plot histogram of event or urevent field values. If NO output args, plots the histogram. If the field values are not numbers or strings, no histogram is computed.

eeg_eventtypes find closest channels in a larger eeglab chanlocs structure to channels in a smaller chanlocs structure

eeg_getepochevent Return dataset event field values for all events of one or more specified types

eeg_insertbound insert boundary event in eeglab event structure.

eeg_lat2point convert latencies in time units relative to the time locking event of an eeglab data epoch to latencies in data points (assuming concatenated epochs).

eeg_matchchans find closest channels in a larger eeglab chanlocs structure to channels in a smaller chanlocs structure

eeg_multieegplot Produce an eegplot of a the average of an epoched dataset (with optional pre-labelling of specific trials).

eeg_point2lat convert latency in data points to latency in ms relative to the time locking. Used in eeglab.

eeg_pvaf Compute EEG.data 'percent variance accounted for' (pvaf) by specified components. Can omit specified components and channels from the computation. Can draw a plot of the scalp distribution of pvaf, or progressively compute the pvaf for comps 1:k, where k = 1 -> the total number of components. Note: pvaf's of spatially non-orthogonal independent components may not add to 100%, and

individual component pvaf could be $< 0\%$.

eeg_rejmacro Internal eeglab macro for all pop_ functions that perform data rejection.

eeg_rejsuperpose superpose rejections of a EEG dataset.

eeg_urlatency find the real latency of an event in the continuous data.

pop_averref Convert an EEG dataset to average reference. This function is obsolete. See pop_reref instead.

pop_chancenter recenter cartesian X,Y,Z channel coordinates

pop_chanedit Edit channel locations (chanlocs) structure of an eeglab dataset. For channel location structure and file format, see `>> help readlocs %` help message of readlocs

pop_chanevent import event latencies from 'edge' values of a specified EEG.data channel

pop_chanse select channel graphical interface

pop_comments edit comments

pop_compareerps Compare the (ERP) averages of two datasets.

pop_comperp Compute the grand average ERP waveforms of multiple datasets, with optional ERP subtraction.

pop_copyset Copy the current EEG dataset into another dataset.

pop_crossf Return estimates and plots of event-related spectral coherence

pop_dipplot plot dipoles.

pop_editeventfield Load/remove event fields in a dataset. If the EEG dataset is the only inputs, a window pops up to ask for the relevant parameter values.

pop_editeventvals Edit events contained in an EEG dataset structure. If the dataset is the only input, a window pops up allowing the user to insert the relevant parameter values.

pop_editset Edit EEG dataset structure fields.

pop_eegfilt interactively filter EEG dataset data using eegfilt

pop_eegplot Visually inspect EEG data using a scrolling display. Perform rejection or marking for rejection of visually (and/or previously) selected data portions (i.e., stretches of continuous data or whole data epochs).

- pop_eegthresh reject artifacts by detecting outlier values. This has long been a standard method for selecting data to reject. Applied either for electrode data or component activations.
- pop_envtopo Plot envelope of an averaged EEG epoch, plus scalp maps of largest or specified components reference to their time point maximum amplitude. Calls envtopo.
- pop_epoch Convert a continuous EEG dataset to epoched data by extracting data epochs time locked to the specified events. Calls epoch.
- pop_erpimage draw an ERP-image plot of a given EEG channel or independent component. Uses a pop-up window if less than three (or four in one condition) input arguments are supplied. Calls erpimage. For further details see >> help erpimage
- pop_eventstat Computes and plots statistical characteristics of an EEG event, including the data histogram, a fitted normal distribution, a normal distribution fitted on trimmed data, a boxplot, and the QQ-plot. The estimates value are printed in a panel and can be read as output. NaNs are omitted. See signalstat.
- pop_expica export ica weights or inverse matrix
- pop_export export EEG dataset
- pop_headplot plot a spherically-splined EEG field map on a semi-realistic 3-D head model.
- pop_icathresh main menu for choosing threshold for component rejection in EEGLAB.
- pop_importdata import data from a Matlab variable or disk file by calling importdata().
- pop_importepoch Export epoch and/or epoch event information to the event structure array of an EEG dataset. If the dataset is the only input, a window pops up to ask for the relevant parameter values.
- pop_importevent Import events into an EEG dataset. If the EEG dataset is the only input, a window pops up to ask for the relevant parameter values.
- pop_importpres append Presentation event file information into eeglab
- pop_jointprob reject artifacts in an EEG dataset using joint probability of the recorded electrode or component activities observed at each time

point. e.g., Observing large absolute values at most electrodes or components is improbable and may well mark the presence of artifact.

pop_loadbcj import a BCI2000 ascii file into eeqlab

pop_loadbva import a Matlab file from brain vision analyser software.

pop_loadcnt load a neuroscan CNT file (pop out window if no arguments).

pop_loaddat merge a neuroscan DAT file with input dataset (pop out window if no arguments).

pop_loadeeg load a Neuroscan .EEG file (via a pop-up window if no arguments).
Calls loadeeg.

pop_loadset load a dataset (pop out window if no arguments)

pop_mergeset Merge two datasets. If only one argument is given, a window pops up to ask for other arguments.

pop_newset Edit EEG dataset info.

pop_plotdata Plot average of EEG channels or independent components in a rectangular array. Else, (over)plot single trials.

pop_plottopo plot concatenated multichannel data epochs in a topographic array format using plottopo

pop_prop plot the properties of a channel or of an independent component.

pop_read_erpss interactively import an uncompressed ERPSS-format data file (.RAW or .RDF) using read_erpss

pop_readbdf Read Biosemi 24-bit BDF file

pop_readedf Read a European data format .EDF data file.

pop_readegi load a EGI EEG file (pop out window if no arguments).

pop_readlocs load a EGI-format EEG file (pop up an interactive window if no arguments).

pop_readsegegi load a segmented EGI EEG file. Pop up query window if no arguments.

pop_rejepoch Reject pre-labeled trials in a EEG dataset. Ask for confirmation and accept the rejection

pop_rejkurt rejection of artifact in a dataset using kurtosis of activity (i.e. to detect

- peaky distribution of activity).
- pop_rejspec rejection of artifact in a dataset using thresholding of frequencies in the data.
- pop_rejtrend Measure linear trends in EEG data; reject data epochs containing strong trends.
- pop_reref Convert an EEG dataset to average reference or to a new common reference.
- pop_resample resample dataset (pop up window).
- pop_rmbase remove channel baseline means from an epoched or continuous EEG dataset.
- pop_runica Run an ica decomposition on an EEG dataset using runica, or other ica algorithm.
- pop_saveh save the eeglab session command history stored in ALLCOM
- pop_saveset save one or more EEG dataset structures
- pop_select remove trials and channels from a dataset
- pop_selectcomps Display components with button to visualize their properties and label them for rejection.
- pop_selectevent Find events in an EEG dataset. If the dataset is the only input, a window pops up to ask for the relevant parameter values.
- pop_signalstat Computes and plots statistical characteristics of a signal, including the data histogram, a fitted normal distribution, a normal distribution fitted on trimmed data, a boxplot, and the QQ-plot. The estimates value are printed in a panel and can be read as output. See SIGNALSTAT.
- pop_snapread load an EEG SnapMaster file (pop out window if no arguments).
- pop_spectopo Plot spectra of specified data channels or components. Show scalp maps of power at specified frequencies. Calls spectopo.
- pop_subcomp subtract specified components from an EEG dataset.
- pop_timef Returns estimates and plots of event-related (log) spectral perturbation (ERSP) and inter-trial coherence (ITC) changes timelocked to a set of input events in one data channel.

pop_timtopo call the timtopo function for epoched EEG datasets. Plots the epoch mean for each channel on a single axis, plus scalp maps of the data at specified latencies.

pop_topoplot Plot scalp map(s) in a figure window. If number of input arguments is < 3, pop-up an interactive query window.

pop_writelocs load a EGI EEG file (pop out window if no arguments).

Signal processing functions

acsobiro A.C.'s Robust second-order blind identification (SOBI) by joint

adjustlocs read neuroscan polar location file (.asc)

axcopy Copy a Matlab figure axis and its graphic objects to a new pop-up window using the left mouse button.

binica Run stand-alone binary version of runica from the Matlab command line. Saves time and memory. If stored in a file, data are not read into Matlab.

blockave make block average of concatenated data sets of same size Each data set is assumed to be of size (chans,frames).

cart2topo convert xyz-cartesian channel coordinates to polar topoplot coordinates. Input data are points on a sphere centered at (0,0,0) or at optional input 'center'. This function is now DEPRECATED! See Important warning below.

cbar Display full or partial colorbar, choose numeric range

cell2mat convert cell array to matrix

chancenter recenter cartesian X,Y,Z channel coordinates

changeunits Takes one or more points in one axes and gives its position in another axes. Useful for drawing lines between sbplots (see sbplot).

compvar project selected components and compute the variance of the original signal they account for.

condstat accumulate surrogate data for comparing two data conditions

convertlocs Convert electrode locations between coordinate systems using the EEG.chanlocs structure.

copyaxis helper function for axcopy

crossf Returns estimates and plots event-related coherence (ERCOH) between two input data time series. A lower panel (optionally) shows

dftfilt discrete Fourier filter

dipplot Visualize EEG equivalent-dipole locations and orientations in the BESA spherical head model or in a averaged MRI model.

eegfilt (high|low|band)-pass filter data using two-way least-squares FIR filtering. Multiple data channels and epochs supported. Requires the MATLAB Signal Processing Toolbox.

eegfiltfft (high|low|band)-pass filter data using inverse fft (without using the Matlab signal processing toolbox)

eegplot Scroll (horizontally and/or vertically) through multichannel data. The current version (3) allows vertical scrolling through channels and manual marking/unmarking of data stretches or epochs for rejection.

eegplot2event convert eegplot rejections into events compatible with eeglab format for continuous datasets.

eegplot2trial convert eegplot rejections into trial and electrode rejections compatible with eeglab format.

eegrej reject/excise arbitrary periods from continuous EEG data (e.g., EEG.data).

eegthresh classical trial rejection rejection using a threshold on the raw signal

entropy calculation of entropy of a 1D, 2D or 3D array and rejection of odd last dimension values of the input data array using the discrete entropy of the values in that dimension (and using the probability distribution of all columns).

env return envelope of rows of a data matrix.

envtopo Plot a data epoch envelope with envelopes and scalp maps of specified components. Click on individual plots to examine separately (with zoom feature).

epoch Extract epochs time locked to specified events from continuous EEG data.

erpimage Image a collection of single-trial data epochs, optionally sorted on and/or aligned to an input sorting variable and smoothed across trials with a moving-average. (To return event-aligned data without plotting, use

eventlock. Optionally sort trials on value, amplitude or phase within a specified latency window. Optionally plot the ERP mean and std. dev., and moving-window spectral amplitude and inter-trial coherence at a selected or peak frequency. Click on individual figures parts to examine them separately and zoom (using axcopy).

eventalign function called by pop_importevent to find the best sampling rate ratio to align 2 arrays of data events.

eventlock Time lock data epochs to specified event frames or event x-values. Use to timelock existing data epochs to reaction times or other events

eyelike calculate a permutation matrix P and a scaling (diagonal) maxtrix S such that $S * P * E$ is eyelike (so permutation acts on the rows of E). E must be a square matrix.

fastif fast if function.

floatread Read matrix from float file ssuming four byte floating point number

floatwrite Write data matrix to float file.

forcelocs rotate location in 3-D so specified electrodes match specified locations.

headplot plot a spherically-splined EEG field map on a semi-realistic 3-D head model. Rotate head using left mouse button.

icaact compute ica activation waveforms = weights*sphere*(data-meandata)

icadefs file of default filenames and constants to source in the ica /ERP package functions. Insert local dir reference below.

icaproj project ica component activations through the associated weight matrices to reconstitute the observed data using only the selected ica components.

icavar project ica component activations through the ica weight matrices to reconstitute the observed data using selected ica components. Returns time course of variance on scalp for each component.

imagesctc imagesc in true color. Can help plot different colormap on the same window.

jader blind separation of real signals using JADE (v1.5, Dec. 1997).

jointprob rejection of odd columns of a data array using joint probability of the values in that column (and using the probability distribution of all columns).

kurt return kurtosis of input data distribution

- loadavg loading eeg average data file from Neuroscan into matlab.
- loadcnt Load a Neuroscan continuous signal file.
- loaddat loading neuroscan format data file into matlab.
- loaddeeg load a binary data file in Neuroscan .eeg file format.
- loadtxt load ascii text file into numeric or cell arrays
- mat2cell convert matrix to cell array
- matsetl select rows, columns, and epochs from given multi-epoch data matrix
- movav Perform a moving average of data indexed by xvals. Supports use of a moving non-rectangular window. Can be used to resample a data matrix (see NOTE below).
- moveaxes move, resize, or copy Matlab axes using the mouse
- nan_mean Average, not considering NaN values
- openbdf Opens an BDF File (European Data Format for Biosignals) in MATLAB (R)
- parsetxt parse text input into cell array
- phasecoher Implements inter-trial amp/coherence using Gaussian wavelets. Removes epoch means. Returns same data length as input frames. Plots results when nargin>6. Outputs have flat ends at data indices [1:halfwin] and [frames-halfwin:frames].
- plotchans3d Plots the 3D configuration from a Polhemus ELP file. The axis of the cartesian coordinate system are plotted. The nose is plotted as a bold red segment.
- plotdata plot concatenated multichannel data epochs in two-column format
- ploterp plot a selected multichannel data epoch on a common timebase
- plotsphere only keep the biggest surface
- plottopo plot concatenated multichannel data epochs in a topographic or rectangular array. Uses a channel location file with the same format as topoplot, or else plots data on a rectangular grid. If data are all positive, they are assumed to be spectra.
- posact Make runica activations all RMS-positive. Adjust weights and inverse weight matrix accordingly.
- projtopo plot projections of one or more ica components along with the original data

in a 2-d topographic array. Returns the data plotted. Click on subplot to examine separately.

qqdiagram Empirical quantile-quantile diagram.

quantile computes the quantiles of the data sample from a distribution X

read_erpss read an compressed and uncompressed ERPSS file formats (.RAW or .RDF)

readbdf Loads selected Records of an EDF or BDF File (European Data Format for Biosignals) into MATLAB

readedf read eeg data in EDF format.

readetraklocs read 3-d location files saved using the EETrak digitizing software.

readegi read EGI Simple Binary datafile (versions 2,3,4,5,6,7). Return header info, EEG data, and any event data.

readegihdr read header information from EGI (versions 2,3,4,5,6,7) data file.

readelp read electrode locations from an .elp (electrode positions) file as generated, for example, by a Polhemus tracking device

readlocs read electrode location coordinates and other information from a file. Several standard file formats are supported. Users may also specify a custom column format. Defined format examples are given below (see File Formats).

readneurodat read neuroscan location file (.dat)

readneurolocs read neuroscan polar location file (.asc)

realproba compute the effective probability of the value in the sample.

rejkurt calculation of kurtosis of a 1D, 2D or 3D array and rejection of outliers values of the input data array using the discrete kurtosis of the values in that dimension.

rejstatepoch reject bad eeg trials based a statistical measure. Can be applied either to the raw eeg data or the ica component activity. This is an interactive function.

rejtrend detect linear trends in EEG activity and reject the epoched trials based on the accuracy of the linear fit.

reref convert common reference EEG data to some other common reference or

to average reference

- rmbase subtract basevector channel means from multi-epoch data matrix
- runica Perform Independent Component Analysis (ICA) decomposition of input data using the logistic infomax ica algorithm of Bell & Sejnowski (1995) with the natural gradient feature of Amari, Cichocki & Yang, or optionally the extended-ICA algorithm of Lee, Girolami & Sejnowski, with optional PCA dimension reduction. Annealing based on weight changes is used to automate the separation process.
- subplot create axes in arbitrary subplot grid positions and sizes
- sdfopen Opens EDF/GDF/SDF files for reading and writing. The EDF format is specified in [1], GDF in [2]. SDF is the SIESTA convention (see [3], pp.8-9); SDF uses the EDF format.
- sdfread Reads selected seconds of an EDF File (European Data Format for Biosignals) The EDF data and header format is specified in [1].
- shuffle shuffle a given dimension in an array
- signalstat Computes and plots statistical characteristics of a signal, including the data histogram, a fitted normal distribution, a normal ditribution fitted on trimmed data, a boxplot, and the QQ-diagram. The estimates value are printed in a panel and can be read as output. Optionally, a topographic map (see TOPOPLOT) can be plotted. The boxplot and the Kolmogorov-Smirnov test require the MATLAB Statistics Toolbox.
- slider add slider to a figure
- snapread Read data in Snap-Master Standard Binary Data File Format Reads Snap-Master header and data matrix (nchans,nframes). Ref: Users Guide, Snap-Master for Windows (1997) p. 4-19
- sobi Second Order Blind Identification (SOBI) by joint diagonalization of correlation matrices. THIS CODE ASSUMES TEMPORALLY CORRELATED SIGNALS, and uses correlations across times in performing the signal separation. Thus, estimated time delayed covariance matrices must be nonsingular for at least some time delays.
- spec power spectrum. This function replaces psd() function if the signal processing toolbox is not present. It uses the timef function.
- spectopo Plot the mean log spectrum of a set of data epochs at all channels as a

bundle of traces. At specified frequencies, plot the relative topographic distribution of power. Uses `pwelch()` from the Matlab signal processing toolbox or `eeglab_spec` function.

sph2topo Convert from a 3-column headplot file in spherical coordinates to a 3-column topoplot file in polar (not cylindrical) coords. Used for `topoplot` and other 2-d topographic plotting programs. Assumes a spherical coordinate system in which horizontal angles has a range [-180, 180] with zero pointing to the right ear. In the output polar coordinate system, zero points to the nose.

spher return the sphering matrix for given input data

spherror_chancenter sub function to compute minimum distance of Cartesian coordinates to a sphere

textsc places text in screen coordinates and places a title at the top of the figure.

timef Returns estimates and plots of mean event-related spectral perturbation (ERSP) and inter-trial coherence (ITC) changes across event-related trials (epochs) of a single input time series. * Uses either fixed-window, zero-padded FFTs (fastest), wavelet 0-padded DFTs (both Hanning-tapered), OR multitaper spectra ('mtaper'). * For the wavelet and FFT methods, output frequency spacing is the lowest frequency ('srate'/'winsize') divided by 'padratio'. NaN input values (such as returned by `eventlock`) are ignored. * If 'alpha' is given, then bootstrap statistics are computed (from a distribution of 'naccu' surrogate data trials) and non-significant features of the output plots are zeroed out (i.e., plotted in green). * Given a 'topovec' topo vector and 'elocs' electrode location file, the figure also shows a `topoplot` of the specified scalp map. * Note: Left-click on subplots to view and zoom in separate windows.

timefdetails details of the `timef` function for time/frequency analysis of multiple epochs of single-channel event-related data.

%%
 %%%

Global Description: `timef` performs normalized time/frequency averaging using FFT-, wavelet-, or multitaper DFT estimates. The wavelets are N-cycle Hanning-windowed sinusoids. (Note: To substitute for `hanning()` windowing `gauss` or other windowing, replace the `timef.m` reference to

hanning()). By default, the two image panels of the output plot show, respectively, the event-related spectral perturbation (ERSP) and inter-trial coherence (ITC) of the input data.

%%
 %%%

ERSP (S. Makeig, *Electroencephalogr Clin Neurophysiol* 86:283-93, 1993) shows mean log event-located deviations from epoch-mean (or baseline-mean) power at each frequency. If bootstrap statistics are computed (via the 'alpha' probability input parameter), (time, freq) points with non-significant differences from 0 are colored green in the image (but not in the 'ersp' output variable - use the output 'erspboot' variable to re-mask the 'ersp' output if desired). The baseline mean spectrum removed from each epoch is available as output parameter "powbase". Note that $\log(\text{power})$ differences are equivalent to $\log(\text{power ratios})$ between baseline and observed spectra, meaning the implicit ERSP model is one of (multiplicative) amplitude modulation of the EEG/MEG spectrum by e.g. subcortical and/or intra-cortical influences. In the default view, the thin bottom panel below the upper (ERSP) image shows the ERSP envelope (the most positive and most negative values at each output time point). The thin left panel shows the mean (or baseline) log spectrum (blue trace). When bootstrap statistics are computed (via the "alpha" argument), the left panel (green trace) also shows the bootstrap significant levels (+/-) at each frequency.

%%
 %%%

ITC: (Inter-trial Coherence, cf. Tallon-Baudry et al., "Phase-locking factor") The lower panel shows the degree of tendency for spectral phase at each (time, freq) point to repeat across the data epochs. If bootstrap statistics are computed (as per the 'alpha' input parameter), non-significant points are colored green (but again not in the 'itc' output variable - use the itcboot output to re-mask if desired in later plotting). The lower thin panel shows the time-domain average (ERP) of the input data (blue) plus a zero-line (green). The average (ERP) is created principally by partial phase resetting of the EEG as measured by the ITC. (While phase resetting dominates, event-related spectral power changes (as measured by the

ERSP) may also play a minor role). The thin left panel shows the frequency-mean ITC (blue trace) and, if bootstrap statistics are computed, the ITC significance limits at each frequency (green trace).

timtopo plot a data epoch and map its scalp map(s) at given time(s)

topo2sph convert a topoplot style 2-D polar-coordinates channel locations file to a 3-D spherical-angle file for use with headplot

topoplot plot a topographic map of a scalp data field in a 2-D circular view (looking down at the top of the head) using interpolation on a fine cartesian grid.

transformcoord
s Select nasion and inion in anatomical MRI images.

trial2eegplot convert eeglab format to eeplot format of rejection window

writectnt Write a Neuroscan continuous signal file.

writelocs write a file containing channel location, type and gain information

Other functions not accessible using the GUI

abspeak find peak amps/latencies in each row of a single-epoch data matrix

arrow Draw a line with an arrowhead.

averef convert common-reference EEG data to average reference

caliper Measure a set of spatial components of a given data epoch relative to a reference epoch and decomposition.

chanproj make a detailed plot of data returned from plotproj for given channel.
Returns the data plotted.

compdsp Display standard info figures for a data decomposition Creates four figure windows showing: Component amplitudes, scalp maps, activations and activation spectra.

compheads plot multiple topoplot maps of ica component topographies

compmap Plot multiple topoplot maps of ica component topographies Click on an individual map to view separately.

compplot plot a data epoch and maps its scalp topography at a given time

compsort reorder ica components, first largest to smallest by the size of their

maximum variance in the single-component projections, then (if specified) the nlargest component projections are reordered by the (within-epoch) time point at which they reach their max variance.

convolve convolve two matrices (normalize by the sum of convolved elements to compensate for border effects).

covary For vectors, covary(X) returns the variance of X. For matrices, covary(X) is a row vector containing the variance of each column of X.

crossfold Returns estimates and plot of event-related coherence (ERC) changes between data from two input channels. The lower panel gives the coherent phase difference between the processes. In this panel, for Ex. - 90 degrees (blue) means xdata leads ydata by a quarter cycle. 90 degrees (orange) means ydata leads xdata by a quarter cycle. Click on each subplot to view separately and zoom in/out.

del2map compute the discrete laplacian of an EEG distribution.

eeg_time2prev returns a vector giving, for each event of a specified ("target") type, the delay (in ms) since the past preceding event (if any) of an event of a specified ("previous") type. Requires the EEG.urevent structure, plus EEG.event().urevent pointers to it.

eegdraw subroutine used by eegplotold to plot data.

eegdrawg subroutine used by eegplotgold to plot data.

eegmovie Compile and view a Matlab movie. Uses scripts eegplotold and topoplot. Use seemovie to display the movie.

eegplotgold display EEG data in a clinical format

eegplotold display data in a horizontal scrolling fashion with (optional) gui controls (version 2.3)

eegplotsold display data in a clinical format without scrolling

envproj plot envelopes of projections of selected ica component projections against envelope of the original data

erpave perform epoch selection plus simple artifact rejection, baseline-zeroing, and/or block averaging

erpimages image multiple single-trial ERPs optionally sorted on an input variable and smoothed by moving-average Plot selected single channels in separate

panels.

gabor2d generate a two-dimensional gabor matrice.

gauss return a smooth Gaussian window

gauss2d generate a 2-dimensional gaussian matrix

getallmenus get all submenus of a window or a menu and return a tree.

gradmap compute the gradient of an EEG spatial distribution.

gradplot Compute the gradient of EEG scalp distribution(s) on a square grid

headmovie Record a Matlab movie of scalp data. Use seemovie to display the movie.

help2html Convert a Matlab m-file help-message header into an .html help file

hungarian Solve the assignment problem using the Hungarian method.

ica Signal processing functions of the eeglab toolbox

icademo a sample ica analysis script using the ICA/ERP package of Matlab functions distributed via <http://www.sccn.ucsd.edu/eeglab>

imagescloglog make an imagesc(0) plot with log y-axis and x-axis values

imagesclogy make an imagesc(0) plot with log y-axis values (ala semilogy()).

laplac2d generate a 2 dimensional gaussian matrice

lapplot Compute the discrete laplacian of EEG scalp distribution(s)

loadelec Load electrode names file for eegplot

logimagesc make an imagesc(0) plot with log y-axis values (ala semilogy())

loglike log likelihood function to estimate dependence between components

logspec plot mean log power spectra of submitted data on loglog scale using plotdata or plottopo formats

makeelec subroutine to make electrode file in eegplot

makehtml generate .html function-index page and function help pages composed automatically from formatted Matlab function help messages

matcorr Find matching rows in two matrices and their corrs. Uses the Hungarian (default), VAM, or maxcorr assignment methods. (Follow with matperm to permute and sign x -> y).

matperm transpose and sign rows of x to match y (run after matcorr)

nan_std std, not considering NaN values

- numdim estimate a lower bound on the (minimum) number of discrete sources in the data via their second-order statistics.
- pcexpand expand data using Principal Component Analysis (PCA) returns data expanded from a principal component subspace [compare pcsquash]
- pcsquash compress data using Principal Component Analysis (PCA) into a principal component subspace. To project back into the original channel space, use pcexpand
- perminv returns the inverse permutation vector
- plotproj plot projections of one or more ica components along with the original data (returns the data plotted)
- promax perform Promax oblique rotation after orthogonal Varimax rotation of the rows of the input data. A method for linear decomposition by "rotating to simple structure."
- qrtimax perform Quartimax rotation of rows of a data matrix.
- read_RDF read RDF-formatted EEG files.
- readlocsold Read electrode locations file in style of topoplot or headplot. Output channel information is ordered by channel numbers.
- rmart Remove eye artifacts from EEG data using regression with multiple time lags. Each channel is first made mean-zero. After JL Kenemans et al., *Psychophysiology* 28:114-21, 1991.
- rmsave return the RMS in each channel, epoch
- runicatest Perform Independent Component Analysis (ICA) decomposition using natural-gradient infomax - the infomax ica algorithm of Bell & Sejnowski (1995) with the natural gradient method of Amari, Cichocki & Yang, the extended-ICA algorithm of Lee, Girolami & Sejnowski, PCA dimension reduction, and/or specgram() preprocessing (suggested by M. Zibulevsky).
- runpca perform principal component analysis (PCA) using singular value decomposition (SVD) using Matlab svd() or svds() >> inv(eigvec)*data = pc;
- seemovie see an EEG movie produced by eegmovie
- shortread Read matrix from short file.

testica Test the runica function's ability to separate synthetic sources. Use the input variables to estimate the (best) decomposition accuracy for a given data set size.

textgui make sliding vertical window. This window contain text with optional function calls at each line.

tftopo Generates a figure showing a selected image (e.g., an ERSP or ITC) from a supplied set of images for every scalp channel, plus topoplot scalp maps at specified (x,y) (e.g., time,frequency) points. Else, images the signed (selected) channel std(). Inputs may be outputs of timef; else, e.g., can be used to image a set of smoothed erpimage images.

timefrq progressive Power Spectral Density estimates on a single EEG channel using out-of-bounds and muscle activity rejection tests. Uses Matlab FFT-based psd().

topoimage plot concatenated multichannel time/frequency images in a topographic format Uses a channel location file with the same format as topoplot or else plots data on a rectangular grid of axes. Click on individual images to examine separately.

tree Make a hierarchical (tree-diagram) component plot. Use successive calls to this function to build the full plot.

tutorial Bring up the ica / electrophysiology toolbox tutorial in a browser window (see docopt.m in the toolbox dir). Tutorial URL:
<http://www.sccn.ucsd.edu/tutorial/> Download: See
<http://www.sccn.ucsd.edu/ica.html>

varimax Perform orthogonal Varimax rotation on rows of a data matrix.

varsort reorder ica components, largest to smallest, by the size of their MEAN projected variance across all time points

zica Z-transform of ica activations; useful for studying component SNR

BIBLIOGRAFÍA

1. EEG – ElectroEncephaloGraph, Biocybernaut Institute, California, 2000-2005.
2. Jaakko Malmivuo, Robert Plonsey, "Principles and Applications of Bioelectric and Biomagnetic Fields," Oxford University Press, New York Oxford, 1995.
3. Thomas Berger, Matthew E. Larkum, and Hans-RLuscher, "High Ih Channel Density in the Distal Apical Dendrite of Layer V Piramidal Cells Increases Bidirectional Attenuation of EPSPS", Institute of Physiology, University of Bern, Switzerland; and Abteilung Zellphysiologie, Max-Planck-Institut fur Medizinische Forscghung, Heidelberg, Germany, 2000.
4. Jaakko Malmivuo, Robert Plonsey, "ElectroEncephalography," Oxford University Press, New York Oxford, 1995.
5. MSOE, "ElectroEncephalograms", Milwaukee School Engineering, 2000.
<http://www.msoe.edu/eecs/be/srdesign/biodata/signals/eeg/eeg.html>
6. Pedro Luis Gambús Cerrillo. "Empleo del ElectroEncefalógrama como medida del nivel de profundidad del efecto de los anestésicos". División de Ciencias de la Salud Universidad de Barcelona, España, 2002.
7. Alberto Llorente e Isaias Martinez, "Encefalografía", Aula – Bioingeniería - Datex – Ohmeda, 2001.
8. Nieves Lorenzo, "Sistemas Biológicos – Caos, Ruido", 2002.

9. Martinez, Isaías “Entropía Espectral y Profundidad Anestésica” Aula – Bioingeniería - Datex – Ohmeda, 2003.
10. Eric W. Weisstein. "Coordinates." From MathWorld--A Wolfram Web Resource, 1999 – 2005.
11. James N. Knight, “Thesis, Signal Fraction Analysis and artifact removal in EEG”, Colorado State University Fort Collins, Colorado Fall 2003.
12. “Nociones Básicas sobre EEG y Potenciales Evocados”, Aula – Bioingeniería - Datex – Ohmeda, 2003.
13. Peter Wong, Cheryl Plumier, Donna Gregory, Cathy Cross, “Guidelines for Digital EEG”, OSET Education Committee Commission 1995-1999.
14. A. Labarga, “Análisis de Componentes Independientes en el Estudio de Potenciales Auditivos P300”, Centro de Estudios e Investigaciones Técnicas de Guipúzcoa CEIT, San Sebastián, España, 2001.
15. Germán Gómez Herrero. “Mismatch Negativity Detection in EEG recordings using Wavelets”. Tampere University of Technology, 2003.
<http://sp.cs.tut.fi/publications/theses/msc/GomezHerrero2003.pdf>
16. Paul Bourke, “Data Formats”,
<http://astronomy.swin.edu.au/~pbourke/dataformats/>
17. Paul Bourke, “Ascii Format for EEG Signals”, 1997.
<http://astronomy.swin.edu.au/~pbourke/dataformats/eegascii/>
18. Medical Imaging Solution GmbH. “EEG File Formats”. Berlin Germany, 2004.
<http://www.eemagine.com/index.htm>
19. Arnaud Delorme y Scott Makeig, “EEGLab” Swartz Center for Computational Neuroscience (SCCN), 2004.

20. Nicoletta Nicolaou and Slawomir J. Nasuto. "Comparison of Temporal and Standard Independent Component Analysis (ICA) Algorithms for EEG Analysis". Department of Cybernetics, University of Reading, UK. 2004.
<http://www.cyber.reading.ac.uk/common/publications/01415.pdf>
21. ERP Division, Research Imaging Center. "What are ERPs?", University of Texas Health Science Center at San Antonio, 2004
<http://ric.uthscsa.edu/facts/erp.html>