

## 6.- Metodología y protocolo experimental

Aunque en esta sección trataremos en la medida de lo posible de expresar conceptos que se repitan de manera general en todos los experimentos, cabe hacer ciertas suposiciones particulares para poder ilustrarlos de forma adecuada. En aquellos experimentos cuya metodología difiera por alguna razón de la que vamos a exponer aquí, lo indicaremos expresamente haciendo hincapié en las diferencias y explicando los motivos de dichos cambios.

Supongamos, por ejemplo, que queremos llevar a cabo un experimento para recoger simultáneamente los datos provenientes de la actividad cerebral (EEG) y las señales biológicas electrocardiograma (ECG), volumen de pulso sanguíneo (BVP), conductividad de la piel (SC) y respiración (RESP) de una persona, mientras está teleoperando un robot por medio de un BCI, y que el objetivo es, por una parte, determinar los momentos en que el sujeto está relajado, estresado o agotado (para lo cual nos serviremos del EEG y las bioseñales), y por otra, evaluar el nivel de atención y la carga de trabajo a la que está sometida el individuo (para lo cual nos basaremos exclusivamente en las señales EEG).

Veremos cómo se realiza este proceso, con los equipos de que disponemos en el laboratorio, en los siguientes pasos:

### 6.1.- Montaje de los sensores y configuración del sistema

#### 6.1.1.- Amplificadores. Características y conexionado

Para nuestros propósitos tenemos en el área de trabajo dos amplificadores de señales biológicas g.USBamp con conexión USB 2.0 para enviar los datos a un PC, de la compañía g.tec Medical Engineering GmbH (Guger Technologies).



Fig. 6.1.1.1: Amplificador de señales biológicas g.USBamp.

Cada amplificador dispone de 16 canales para EEG o biosensores, cada uno de ellos con 24 bits y frecuencia de muestreo de hasta 38.400 Hz, agrupados en grupos de 4, cada uno de estos grupos con su propia tierra y referencia, para evitar interferencias entre distintos tipos de sensores. El amplificador puede conectarse directamente a un PC mediante USB para la adquisición de datos, sin necesidad de ningún otro dispositivo intermedio. Posee un conversor analógico/digital para cada canal y un DSP que realiza el sobremuestreo y el filtrado en tiempo real de los datos, consiguiendo una alta relación señal ruido (SNR). Además, tiene una unidad interna de calibración y chequeo de impedancias, y la posibilidad de suprimir la interferencia de la red eléctrica. Por último, es configurable mediante software gracias a una aplicación basada en C llamada C API (C language Application Programming Interface), e incluye un programa de adquisición de datos (g.USBamp demo).

En las siguientes figuras podemos ver detalladamente la parte frontal y posterior del equipo:

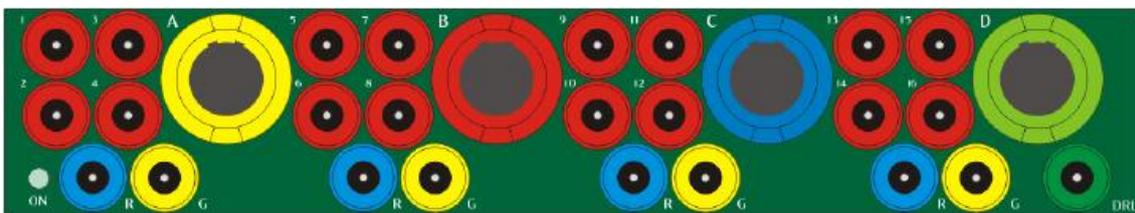


Fig. 6.1.1.2: Conectores de la parte frontal. Canales (rojo), referencias (azul) y tierras (amarillo).



Fig. 6.1.1.3: Parte posterior. Switch de encendido, fusible, alimentación, sincronización con otro amplificador, entradas/salidas digitales y conexión USB.

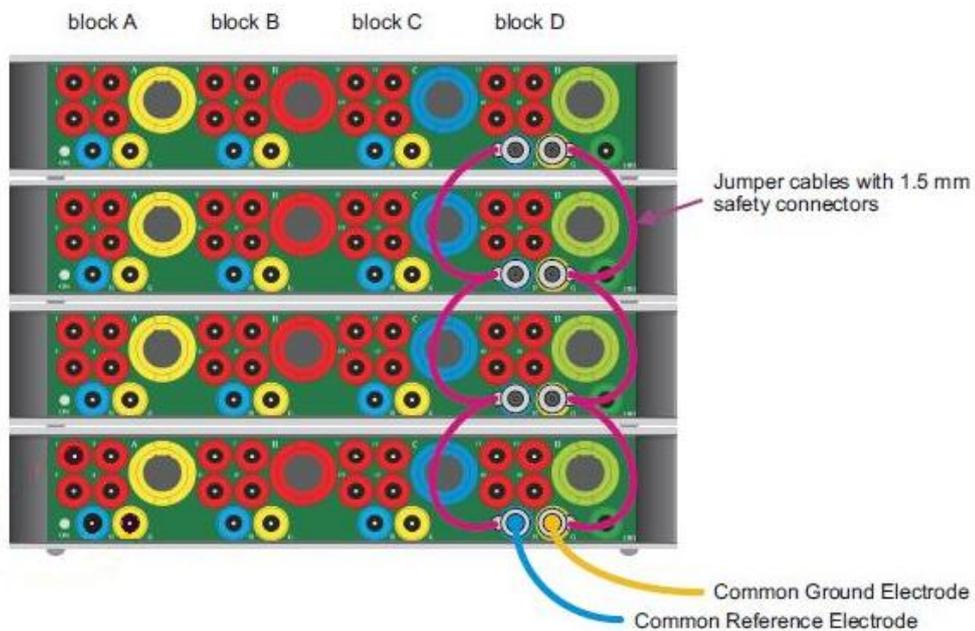


Fig. 6.1.1.4: Conector de la fuente de alimentación (5 V DC).



**Fig. 6.1.1.5: Cable USB y conector.**

Si queremos usar más de 16 canales de EEG, o como haremos habitualmente, adquirir datos de EEG y biosensores de manera simultánea, debemos sincronizar varios amplificadores como se indica en las siguientes figuras:



**Fig. 6.1.1.6: Conexión de varios amplificadores. Vista frontal.**

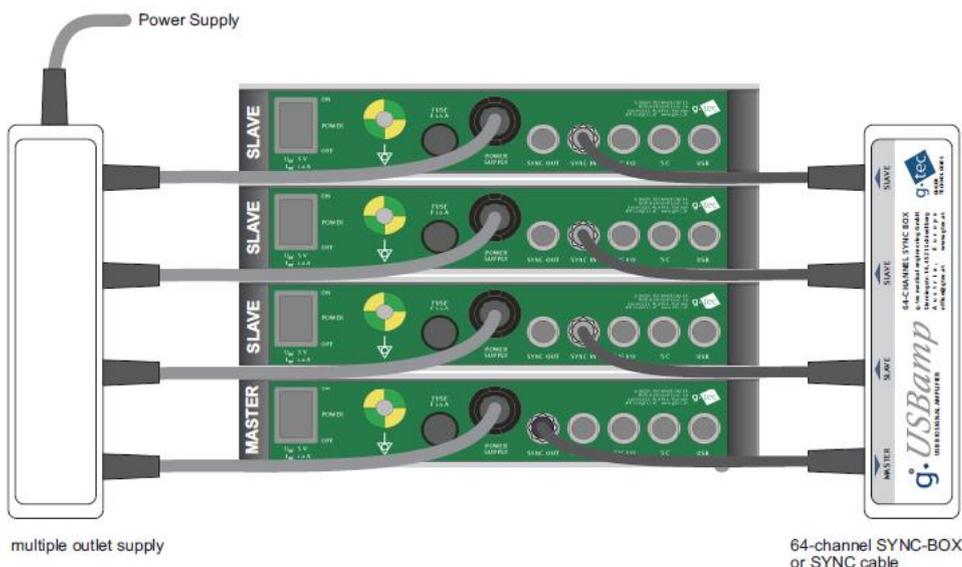


Fig. 6.1.1.7: Conexión de varios amplificadores. Vista posterior.

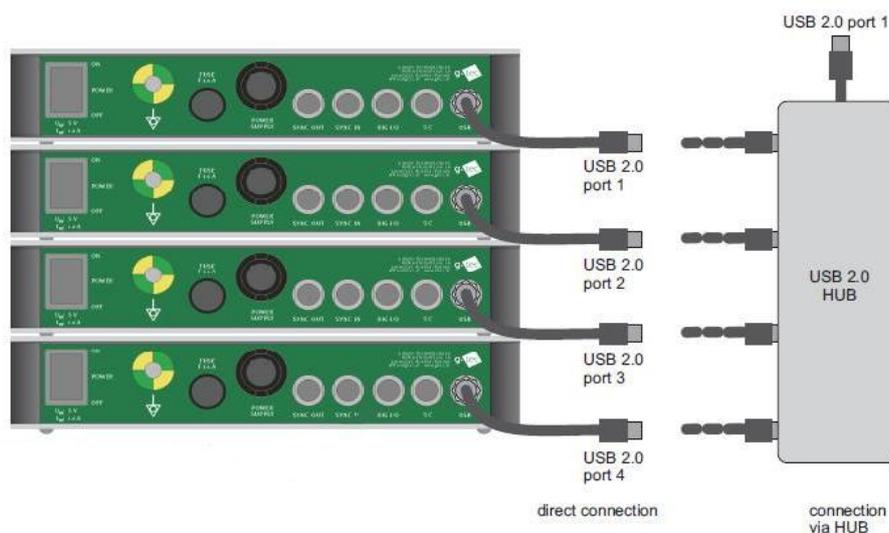


Fig. 6.1.1.8: Conexión de varios amplificadores. Hub USB.

En nuestro caso, habitualmente usaremos 16 ó 24 canales EEG y 4 biosensores, con lo cual será necesario conectar dos amplificadores, y recogeremos los datos en dos PCs, por lo que no necesitaremos el hub y simplemente conectaremos un puerto USB a cada PC.

Para calibrar el equipo, todos los electrodos han de estar desconectados de las entradas del amplificador. Éste generará internamente una señal para proceder a la calibración, que se hará a través del programa g.USBamp demo. Posteriormente se conectan los electrodos, se miden las impedancias y se configura el aparato, como veremos más adelante, y el sistema está listo para funcionar.

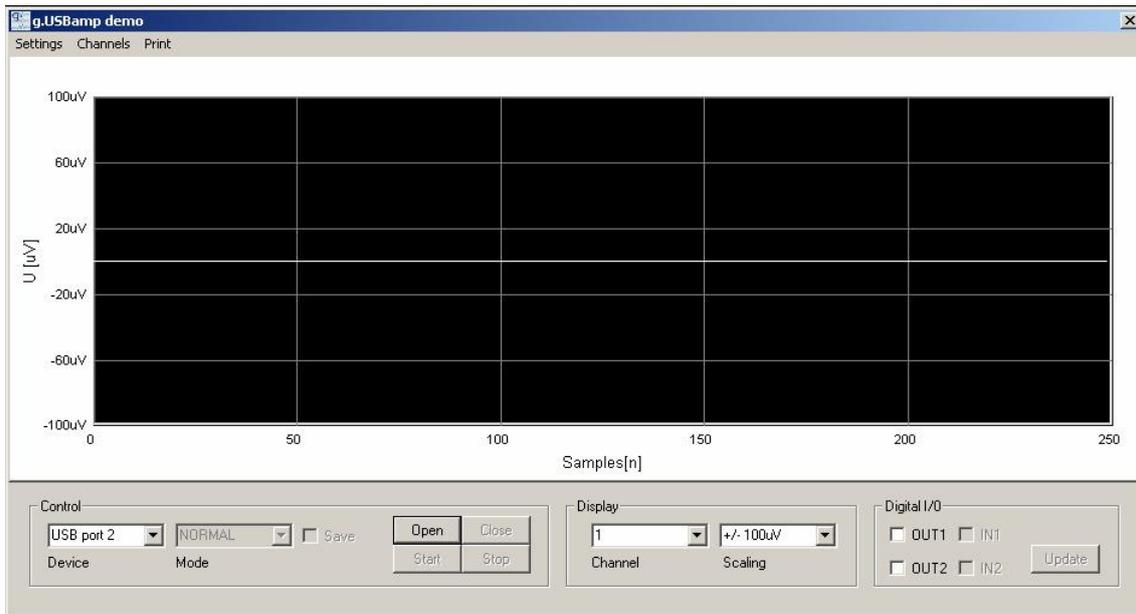


Fig. 6.1.1.9: Software g.USBamp demo.

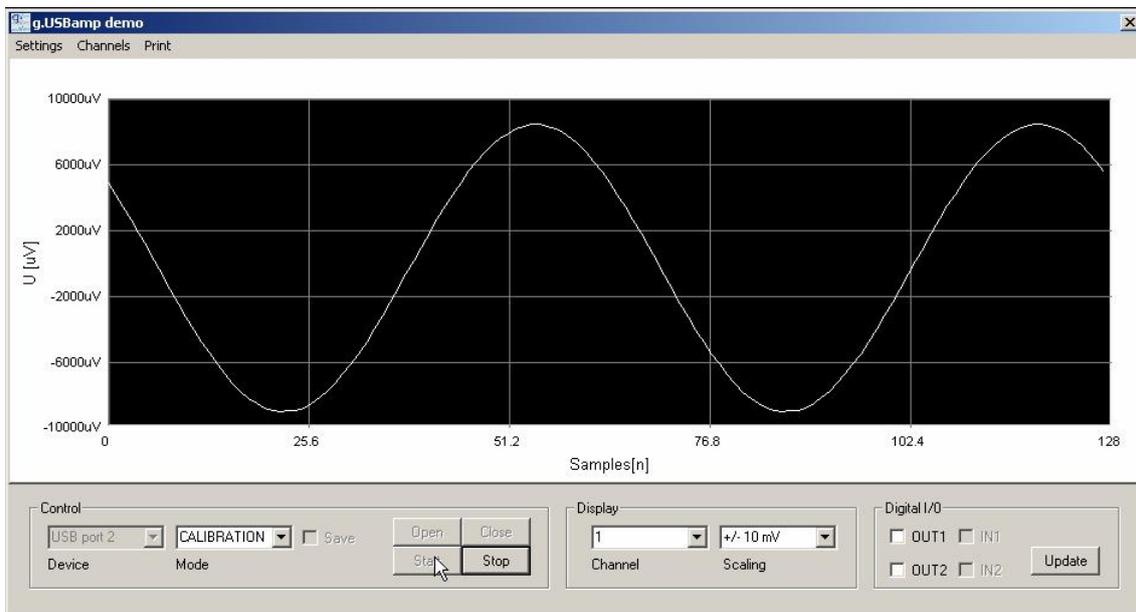
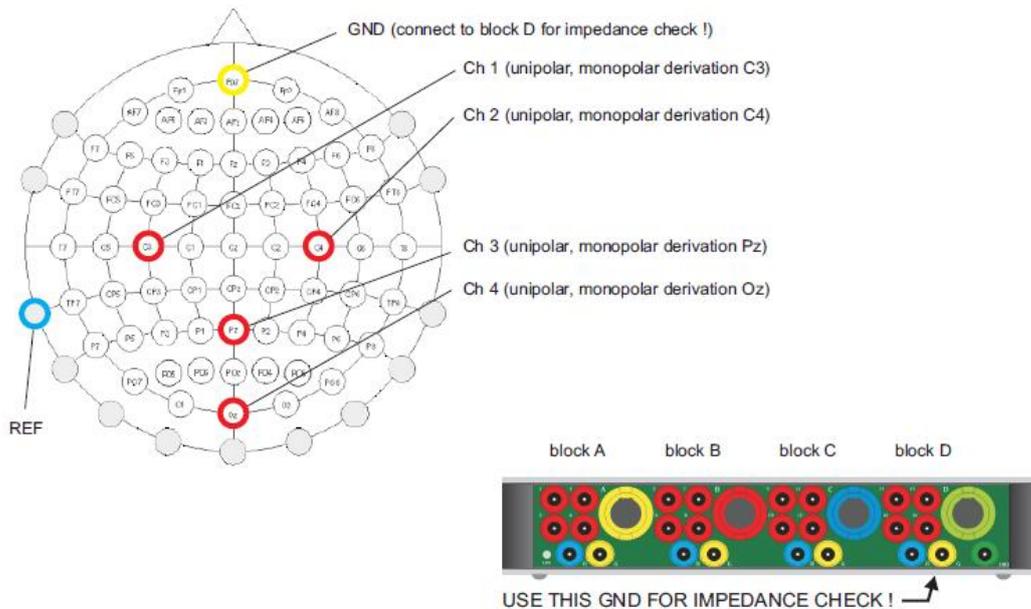


Fig. 6.1.1.10: Calibración del equipo mediante software.

Podemos modificar la señal de calibración pulsando el botón 'Settings', en nuestro caso pondremos una onda senoidal de  $\pm 10$  mV de amplitud y 10 Hz.

## 6.1.2.- Colocación de los electrodos de EEG

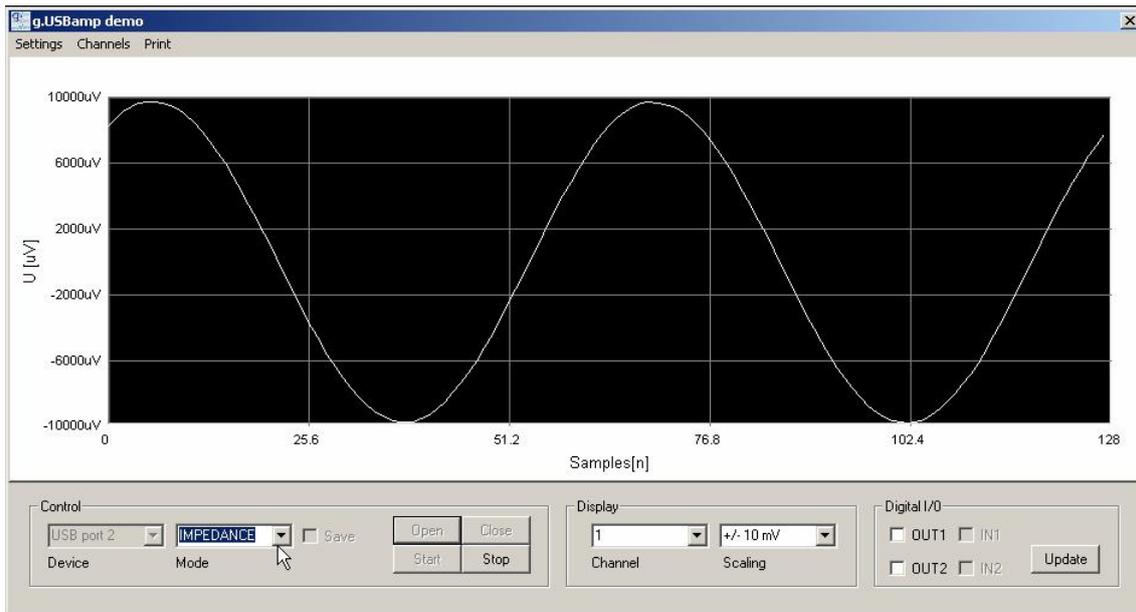
A continuación, emplazamos los electrodos con gel conductor en un casco de malla en la cabeza del sujeto, según el sistema internacional 10-20, que pudimos ver en la figura 1.1.1 de la introducción, y los conectamos a las entradas del amplificador:



**Fig. 6.1.2.1: Colocación de los electrodos EEG.**

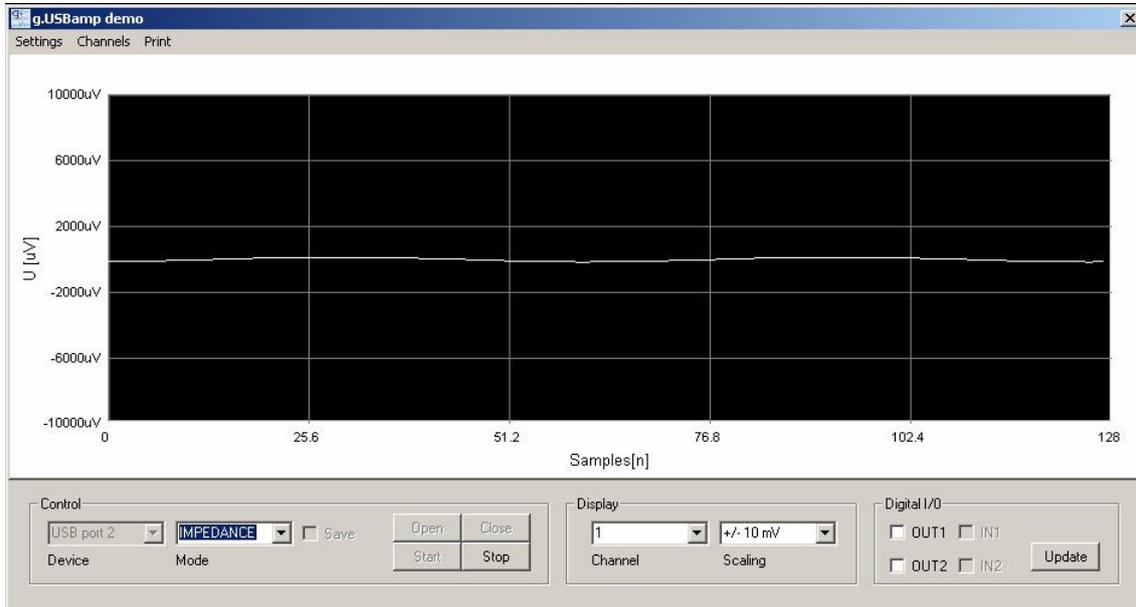
Para la medida de impedancias, el electrodo de tierra debe estar conectado en la entrada amarilla del bloque D, como se indica en la figura anterior. El equipo genera internamente una onda senoidal de  $\pm 10$  mV y 10 Hz (la señal de calibración) en cada electrodo, mide la caída de tensión en cada uno de ellos y la transmite a través del USB al PC para su monitorización en la interfaz software.

En el modo de medición de impedancias, si seleccionamos por ejemplo el canal 1 y no tenemos conectado un electrodo, lo que vemos es la señal interna:



**Fig. 6.1.2.2: Medición de impedancias en g.USBamp con el electrodo desconectado.**

Si conectamos el electrodo, para una impedancia de unos 10 kΩ, la señal que se vería en pantalla sería más pequeña:



**Fig. 6.1.2.3: Medición de impedancias en g.USBamp con una impedancia de 10 kΩ.**

La impedancia se puede calcular con la siguiente fórmula:

$$Z_x = U_1 * 106 / (U_{cal} - U_1) - 10^3$$

Donde  $U_1$  es la amplitud de la señal medida en el canal 1 y  $U_{cal}$  la amplitud de la señal generada internamente por el dispositivo (10 mV). La impedancia de cada electrodo debe ser menor de 5 k $\Omega$  para una adecuada adquisición de datos. En caso contrario, habría que rehacer el montaje aplicando de nuevo el gel conductor, hasta que el resultado sea el correcto.

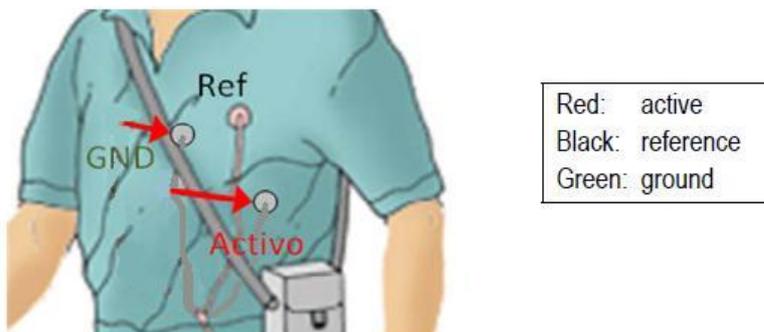
### 6.1.3.- Colocación de los biosensores

El montaje de los sensores electrocardiograma, volumen de pulso sanguíneo, conductividad o respuesta galvánica de la piel y respiración se hará de la siguiente manera:

#### ELECTROKARDIOGRAM: ECG

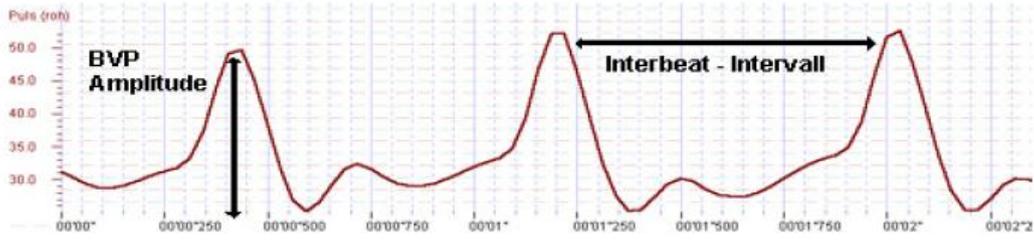


Sensors placement:



**Fig. 6.1.3.1: Colocación electrodos ECG.**

**BLOOD VOLUME PULSE: BVP**



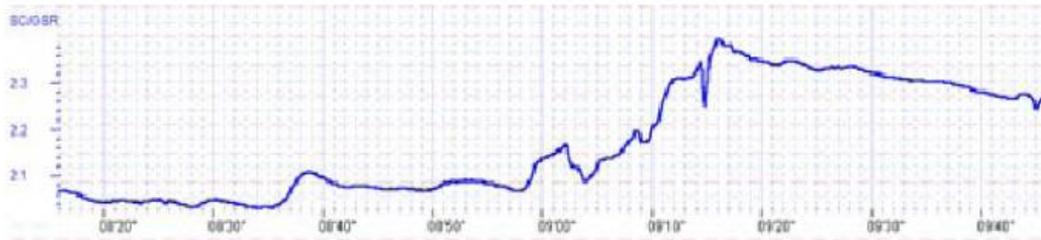
Sensors placement:



Red:	active
Black:	reference
Yellow:	ground

**Fig. 6.1.3.2: Colocación sensor BVP.**

**SKIN CONDUCTIVITY: SC**

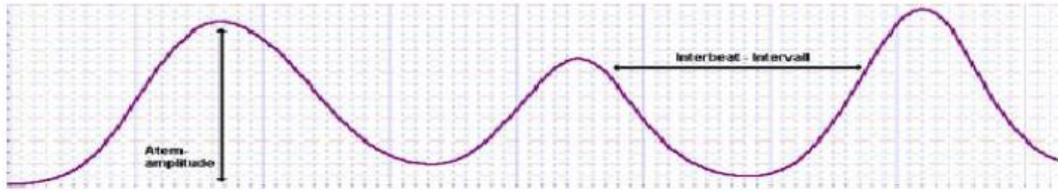


Sensors placement:

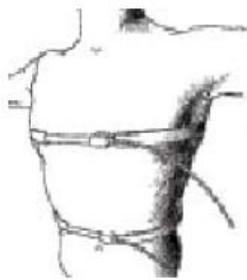


Red:	active
Black:	ground

**Fig. 6.1.3.3: Colocación sensor SC.**

**RESPIRATION: RESP**

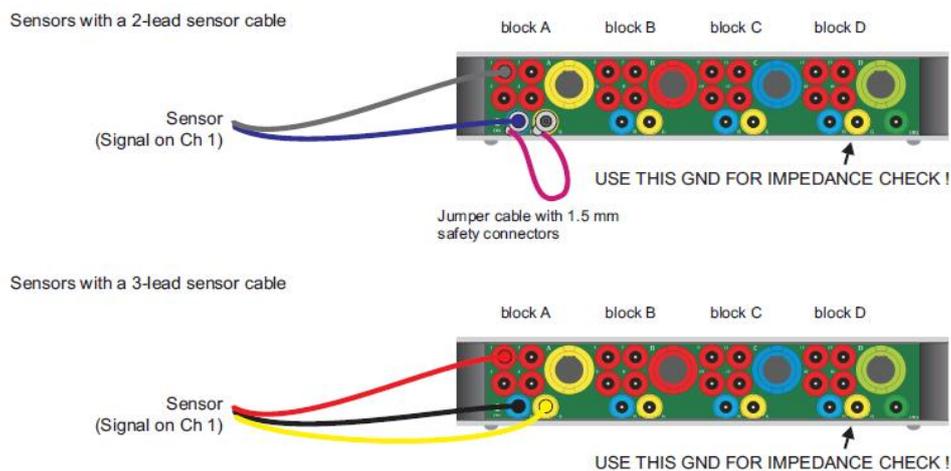
Sensors placement:



Blue:	active
White:	ground

**Fig. 6.1.3.4: Colocación sensor RESP.**

Posteriormente, se conectarán los sensores al amplificador (en bloques en los que no haya ningún electrodo de EEG). La calibración y la medida de impedancias se efectuarán tal como se explicó en el apartado anterior.

**Fig. 6.1.3.5: Conexión de los sensores al amplificador (abajo ECG y BVP, arriba SC y RESP).**

## 6.2.- Adquisición de datos. Protocolo de estímulos/tareas

Una vez tengamos todo el montaje previo, podemos comenzar el experimento. Usaremos el software g.USBamp para configurar los parámetros de adquisición (frecuencia de muestreo, filtros, etc.). Si abrimos el programa y seleccionamos el modo normal y el puerto USB de entrada, pulsando el botón 'Open' estableceremos la conexión con el dispositivo. Si la conexión se realiza correctamente, se hará visible el botón 'Start', que podremos pulsar para ver los canales en pantalla y comenzar la adquisición:

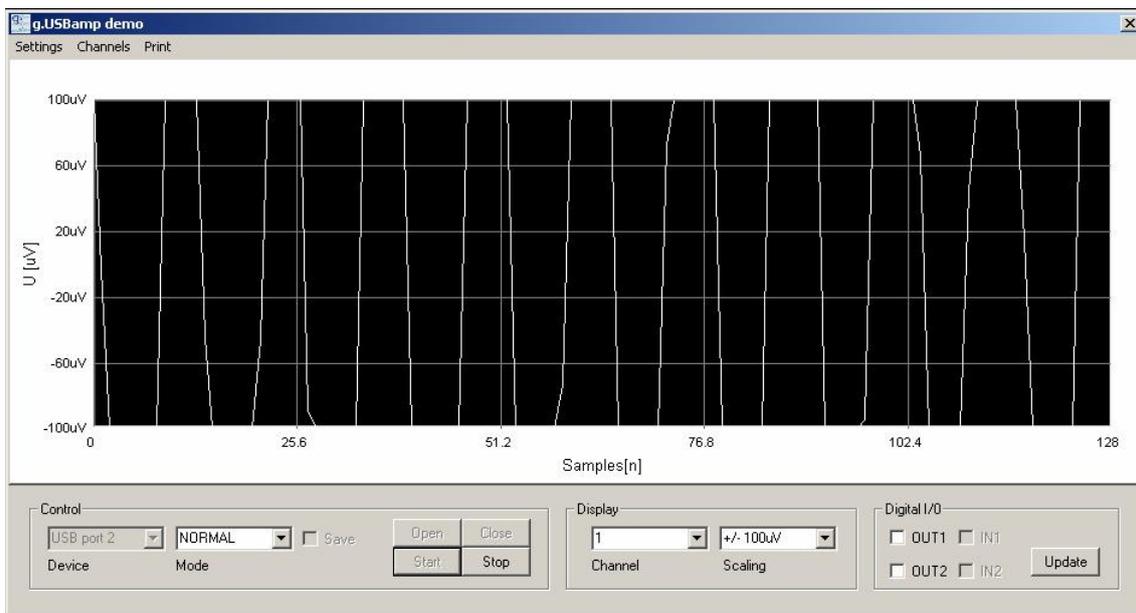


Fig. 6.2.1: Adquisición de datos en g.USBamp.

Podemos cambiar el canal en la pestaña 'Channel' y la escala en la pestaña 'Scaling'. Para cambiar la tasa de muestreo tendremos que pulsar 'Stop' y abrir la ventana 'Filter' del menú 'Settings':

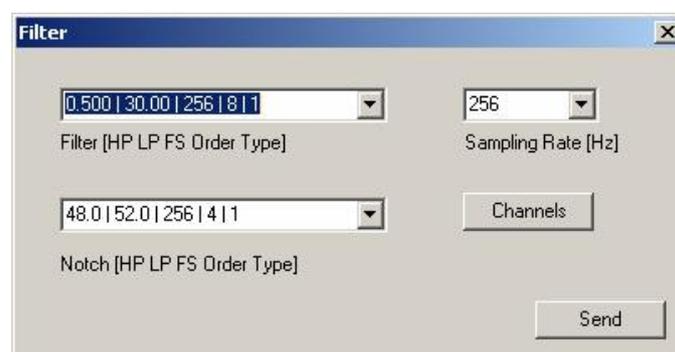


Fig. 6.2.2: Configuración de la frecuencia de muestreo y el filtrado.

Seleccionaremos la frecuencia 256 Hz, y pondremos los parámetros de filtrado tal como vemos en la figura, esto es, un filtro pasobanda entre 0.5 y 30 Hz (Butterworth de orden 8) para eliminar las interferencias a altas frecuencias, y un filtro notch o rechazo de banda entre 48 y 52 Hz (Butterworth de orden 4) para suprimir la interferencia de la red eléctrica. En el botón 'Channels' podemos seleccionar los canales a los que se aplicará esta configuración, pulsando 'Send' enviamos los datos al amplificador.

También podemos conectar vía software las tierras de los distintos bloques (en el caso de bloques de electrodos EEG los conectamos todos), abriendo la ventana 'Common ground' del menú settings. Lo mismo podemos hacer con las referencias en la ventana 'Common reference'.

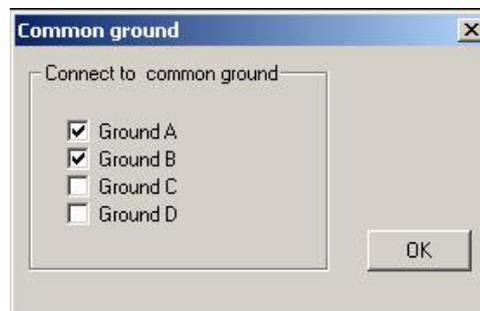


Fig. 6.2.3: Conexión de las tierras entre bloques.

Para guardar los datos en el disco, en la ventana principal señalamos la opción 'Save' y presionamos 'Start' para comenzar. Aparecerá una ventana para que escribamos el nombre del archivo. El programa comenzará la adquisición de los datos e irá guardando éstos en dicho archivo. Para parar la adquisición y cerrar el programa, bastará con pulsar 'Stop' y 'Close' para desconectar el amplificador, y cerrar la ventana.

Los estímulos y las tareas a realizar por parte de los usuarios variarán en cada experimento (usaremos el 'Speller' o la tele-operación de un robot, los cuales explicaremos más adelante), pero generalmente utilizaremos el mismo patrón, las tareas serán de la siguiente forma:

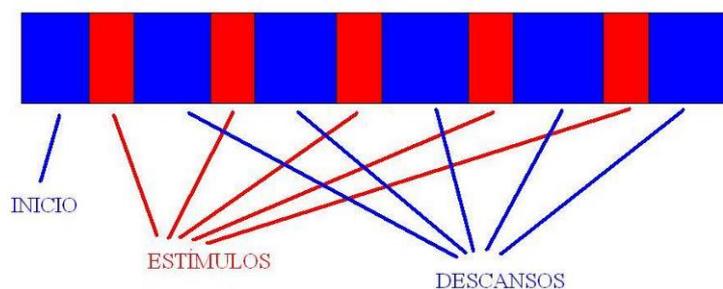


Fig. 6.2.4: Protocolo de tareas a desarrollar por el usuario de BCI.

Habrà una fase inicial, lo suficientemente larga para que el usuario se relaje, y que servirà para adquirir el ‘baseline’ o paràmetros de relajaci3n. Posteriormente, el experimento se dividirà en fases de trabajo, ya sea entrenamiento (machine learning) u operaci3n (dependerà del experimento en concreto), y tiempos de descanso, algo màs duraderos que los anteriores y en los que deja de haber estimulaci3n (por ejemplo, en el caso de la tele-operaci3n del robot, coincidiràn con el perìodo en que el robot navega segùn la orden que ha producido el sujeto). La idea es que, por un lado, podamos obtener características de los dos bloques (que se corresponderàn con los estados de relajaci3n y estr3s) y comprobar en cuàles de ellas se encuentran las diferencias màs significativas entre dichos estados, y por otra parte, ver c3mo evolucionan estas características a medida que avanza el experimento para poder estudiar el agotamiento (la hip3tesis es que, cuando se han realizado muchas misiones, el usuario se agota, cambian algunos paràmetros de estr3s y le cuesta màs alcanzar el estado de relajaci3n).

En todos los casos, sea cual sea la tarea a realizar, tanto la presentaci3n de los estìmulos como las acciones producidas por los usuarios seràn llevadas a cabo por medio del software BCI2000.

El sistema BCI2000 consta de cuatro m3dulos que son programas independientemente ejecutables: Operador, adquisici3n y almacenamiento de seàales EEG, procesamiento de seàal y aplicaci3n.

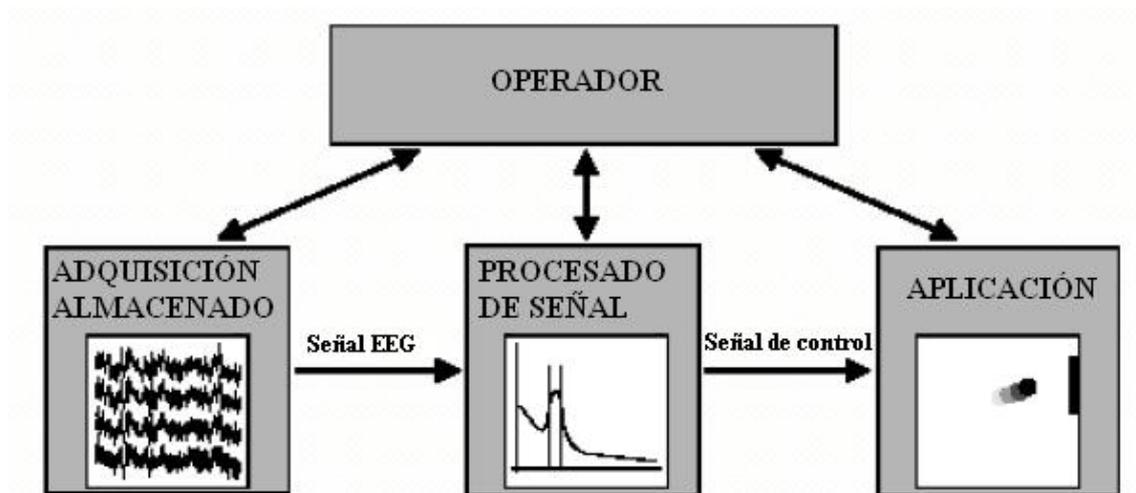


Fig. 6.2.5: M3dulos autoejecutables de BCI2000.

El m3dulo de adquisici3n y almacenado va tomando series de datos del amplificador, los guarda en memoria y los pasa al m3dulo de procesamiento de seàal, que implementa una serie de filtros en cascada y envía la seàal resultante al m3dulo de aplicaci3n. Éste controla la tarea del usuario y la apariencia de la ventana donde se muestra la interfaz del usuario. El operador sirve como interfaz entre el operador humano y el sistema BCI2000, recolectando paràmetros y representando datos. Cada

módulo dispone de un menú de configuración en el operador, siendo especialmente importantes el de procesamiento de señal y el de aplicación, porque variarán en función de la aplicación que se use (por ejemplo, se puede hacer entrenamiento con ritmos *mu* o con P300). Existen ficheros de parámetros disponibles para cada aplicación, de modo que basta con cargarlos en el operador para configurar todos los módulos, o bien se puede realizar la configuración manualmente. Para el caso de aplicaciones nuevas, también se permite guardar la configuración en un archivo para ser utilizado en ocasiones posteriores.

Aunque BCI2000 está formado por los cuatro programas que deben estar corriendo simultáneamente para que el sistema pueda funcionar, habitualmente se lanzará el programa desde un fichero 'batch' para la aplicación en cuestión de manera que se ejecutarán todos los módulos pero únicamente se hará visible el operador, como podemos ver en las siguientes figuras:

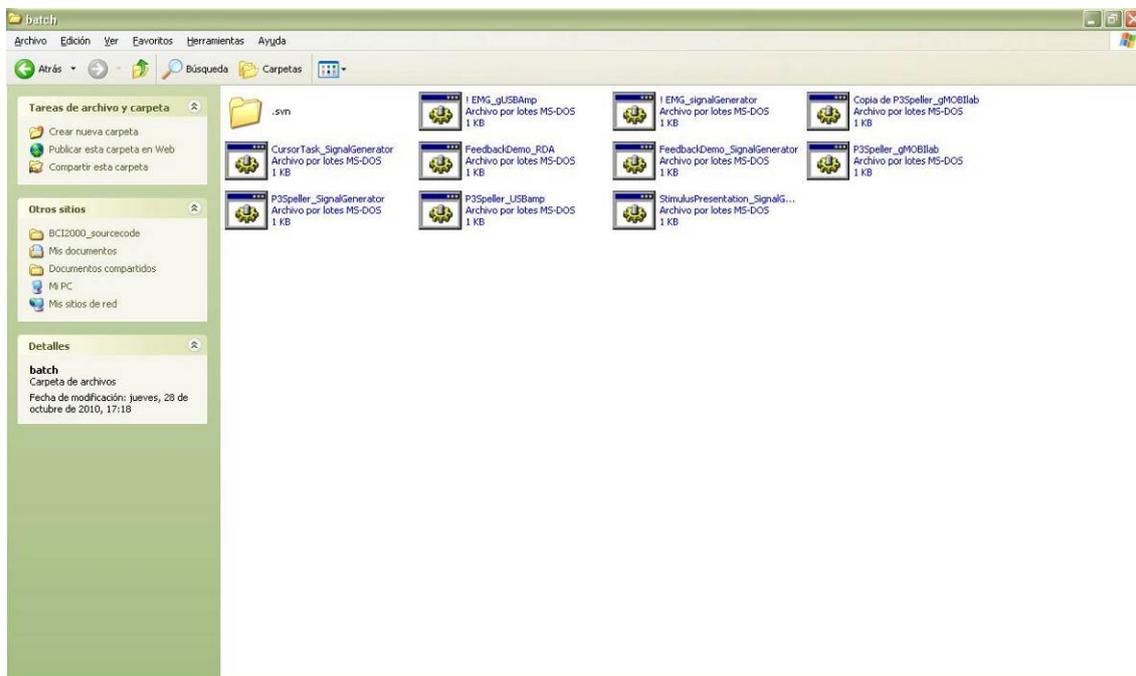


Fig. 6.2.6: Archivos 'batch' para lanzar la aplicación deseada en BCI2000.

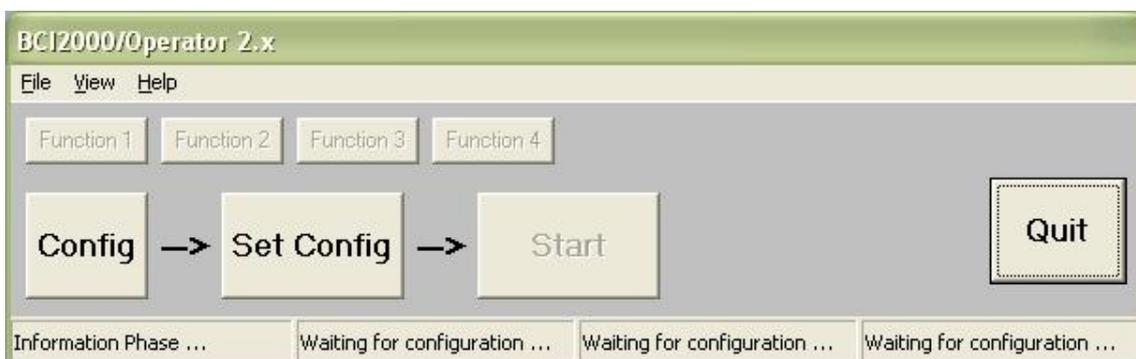


Fig. 6.2.7: Módulo operador de BCI2000.

Pulsando el botón ‘Config’ accederemos a los menús comentados anteriormente, de manera que podremos, por ejemplo, cargar un archivo de parámetros para la aplicación que vayamos a usar (en este caso, a modo ilustrativo, el P300Speller):

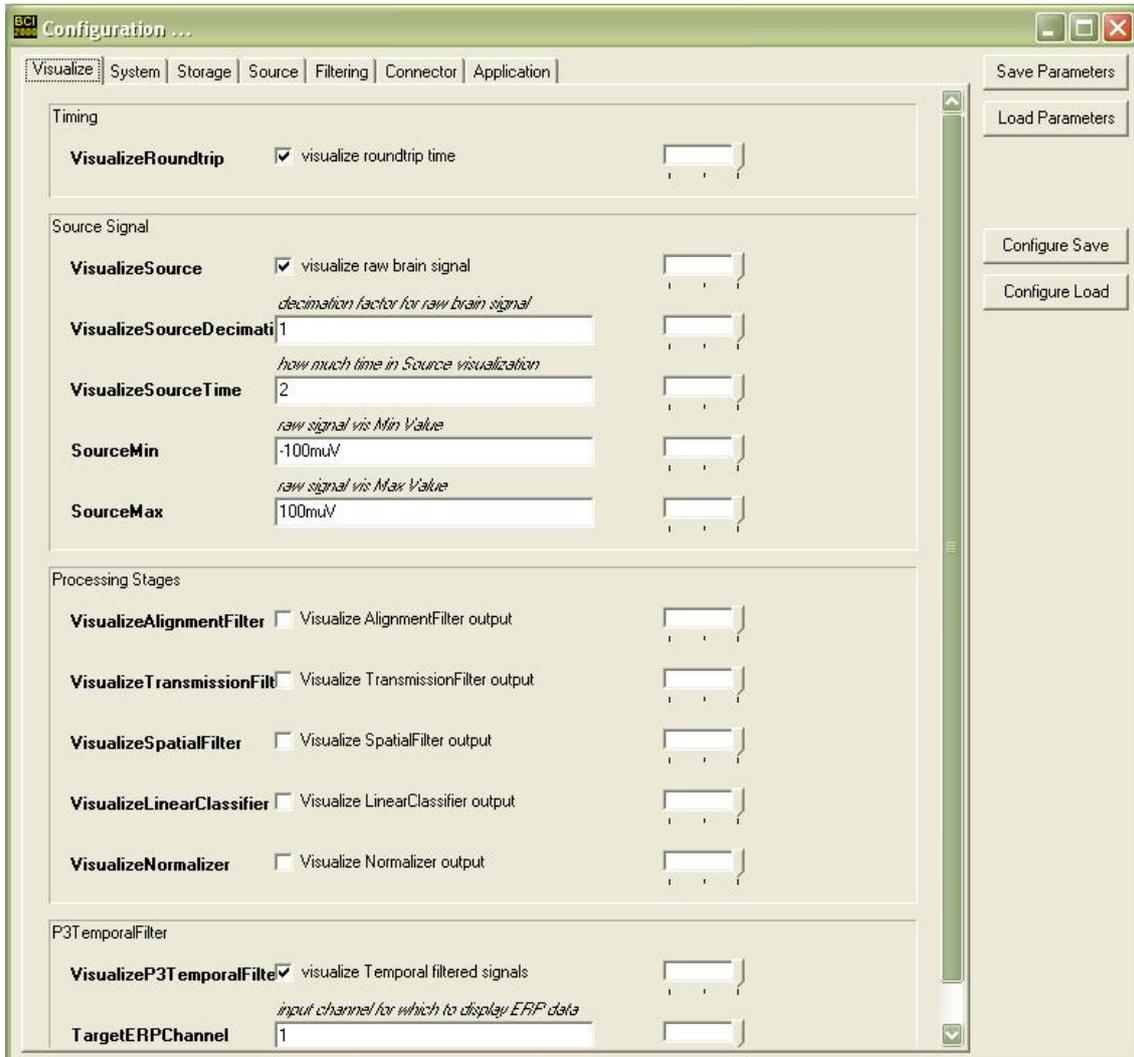


Fig. 6.2.8: Menú de configuración del operador de BCI2000.

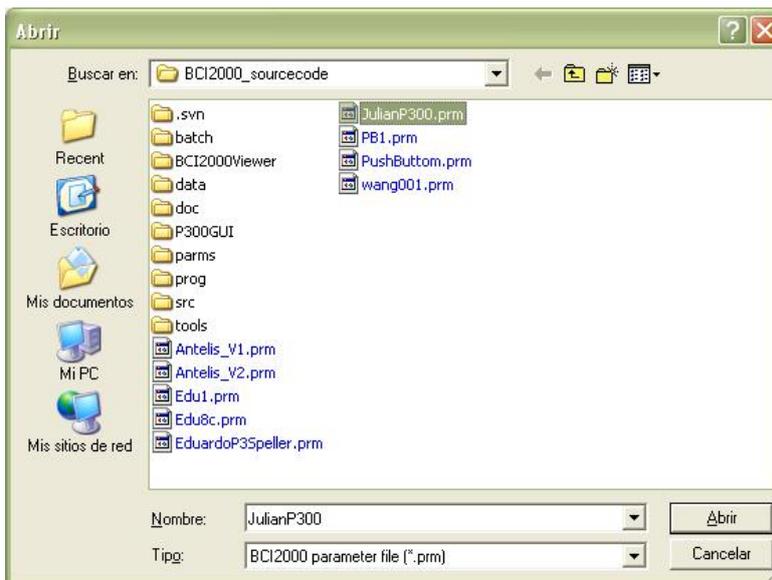


Fig. 6.2.9: Carga de fichero de parámetros de configuración.

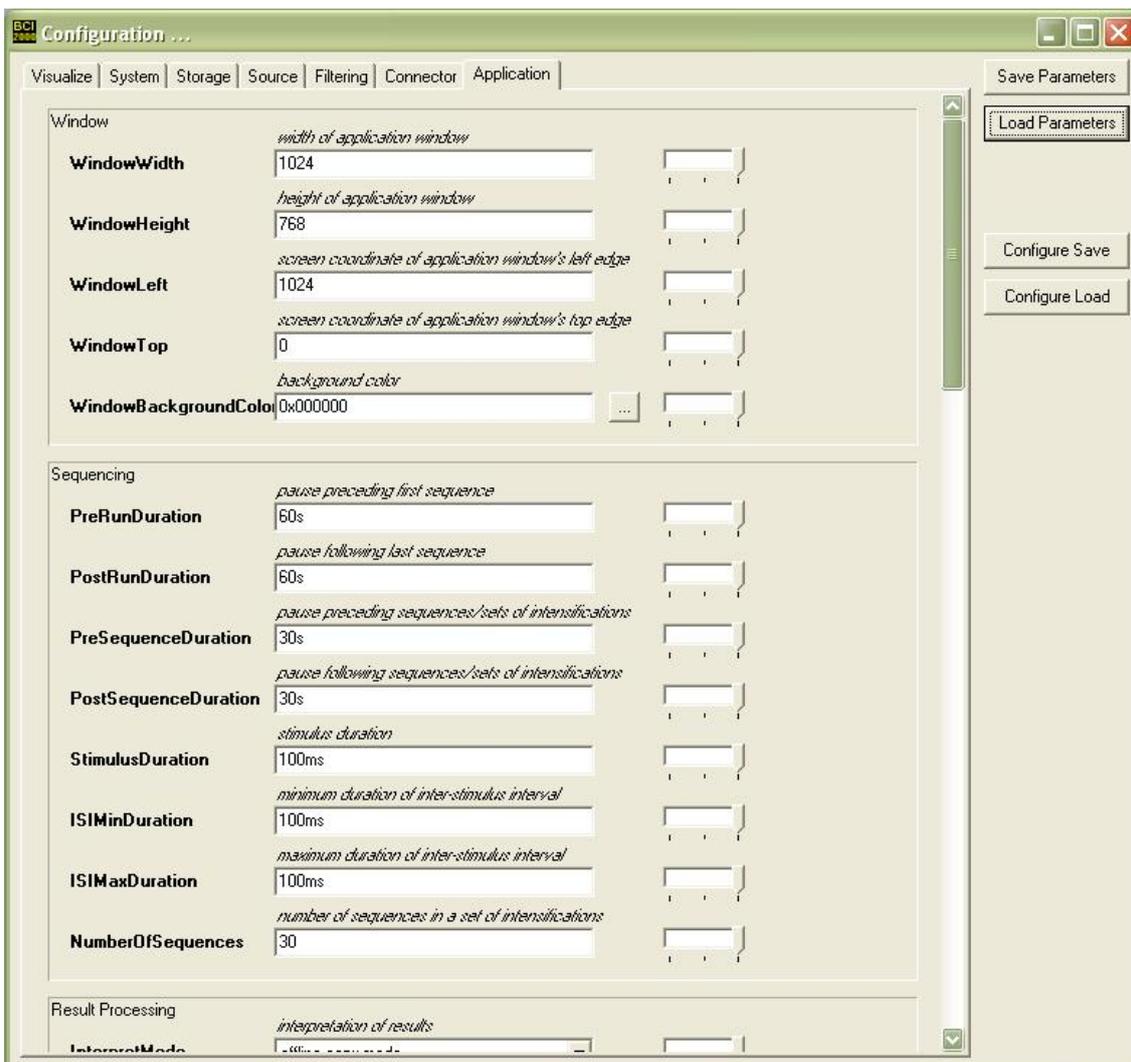


Fig. 6.2.10: Menú de configuración con los parámetros deseados para la aplicación.

Ahora sólo quedaría cerrar la ventana de configuración, pulsar el botón ‘Set Config’ y el sistema estaría listo para comenzar el experimento, para ello extendemos la ventana de interfaz de usuario a un monitor externo conectado al PC y pulsamos el botón ‘Start’:

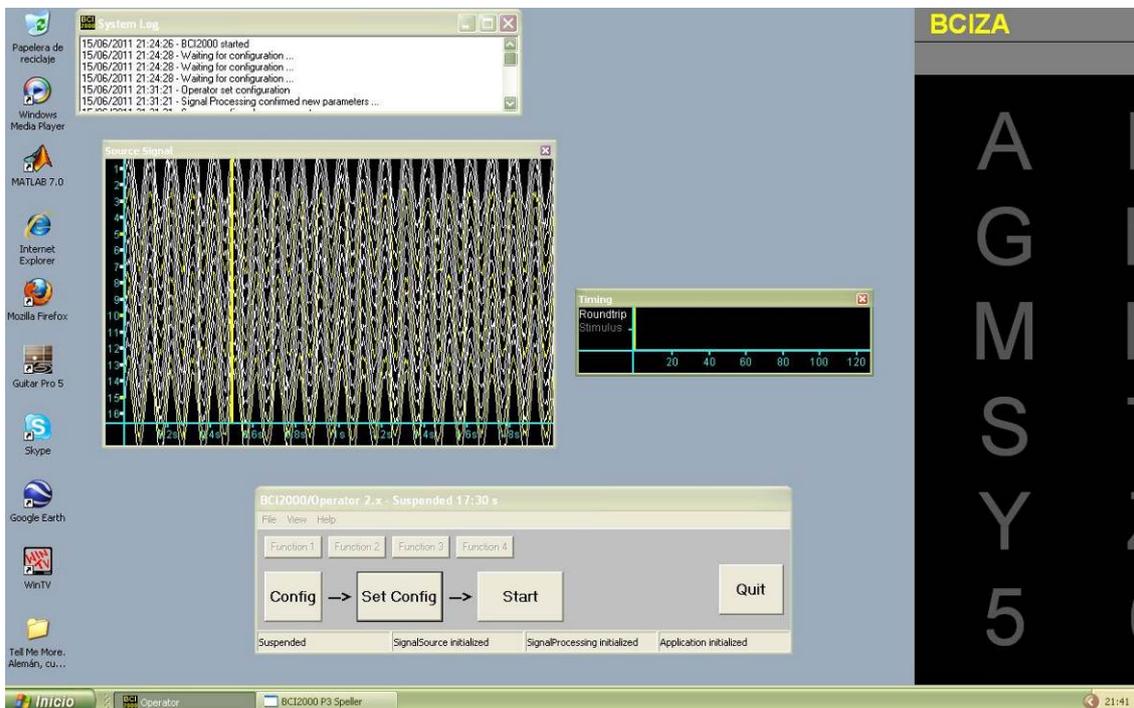


Fig. 6.2.11: Sistema BCI2000 con la aplicación (derecha) y los parámetros configurados.

### 6.3.- Procesamiento de señales y extracción de características

Explicaremos aquí en detalle el tratamiento de las señales obtenidas a partir de los experimentos, la metodología en general será la siguiente:

- Convertiremos los datos adquiridos mediante g.USBamp demo o el propio BCI2000 (habitualmente éste último) en variables del espacio de trabajo de MATLAB mediante el programa ‘load\_bcidat.m’, que forma parte del software BCI2000. Llamaremos a este programa desde un programa propio que se llama ‘datosbci2matlab.m’, que, además, adecuará estas variables para nuestro uso posterior, cambiando por ejemplo la notación o la estructura de los datos, y haciendo representaciones para que podamos tener una vista preliminar de las señales de interés.

- Se programará, para cada señal, una serie de filtros y algoritmos para obtener características que puedan ser de interés para buscar las diferencias entre estados, según los estudios teóricos en los que nos basamos, y que puedan ser utilizadas como entradas directas a un clasificador. A continuación explicaremos en detalle esta fase para cada biosensor y para los canales de EEG.

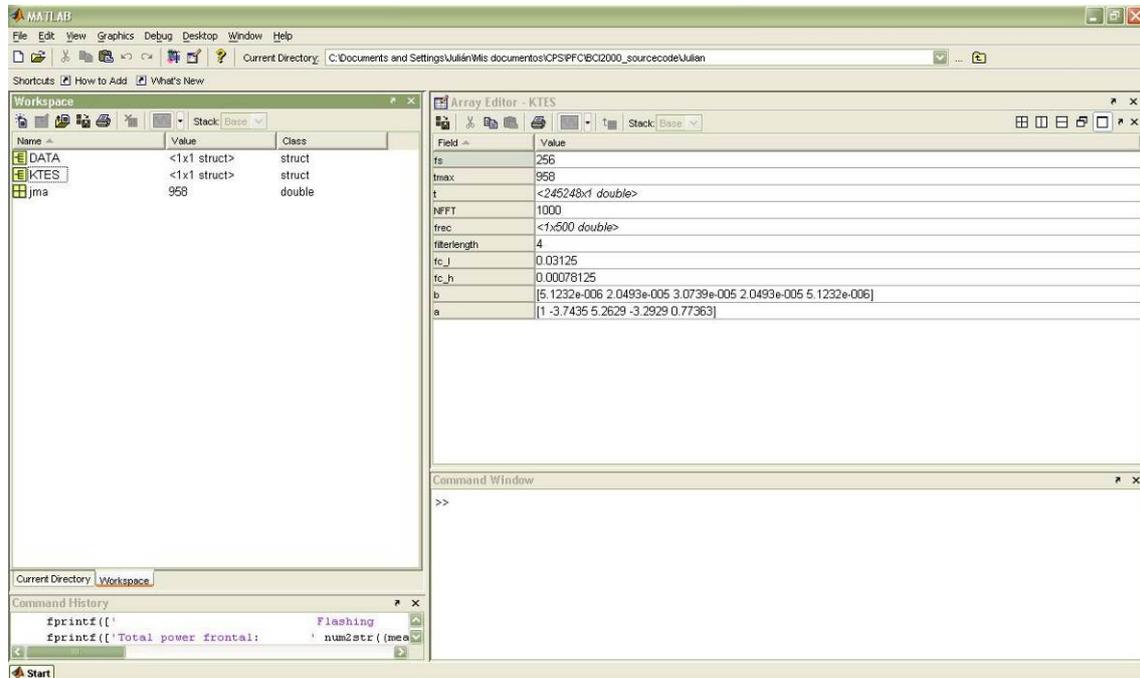


Fig. 6.3.1: Variables de MATLAB obtenidas a partir de un fichero ‘.dat’ generado por BCI2000.

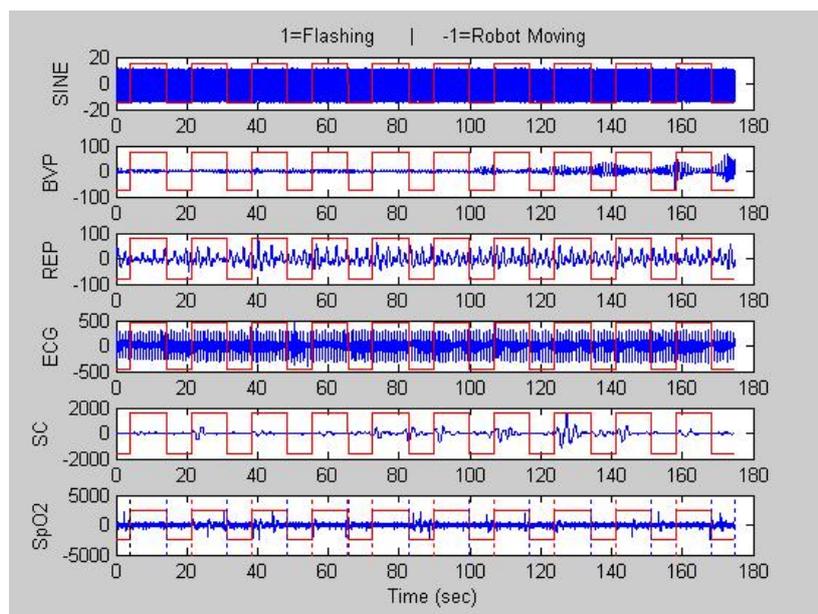


Fig. 6.3.2: Vista preliminar de las señales adquiridas, biosensores en este ejemplo.

### 6.3.1.- Procesamiento de la señal ECG

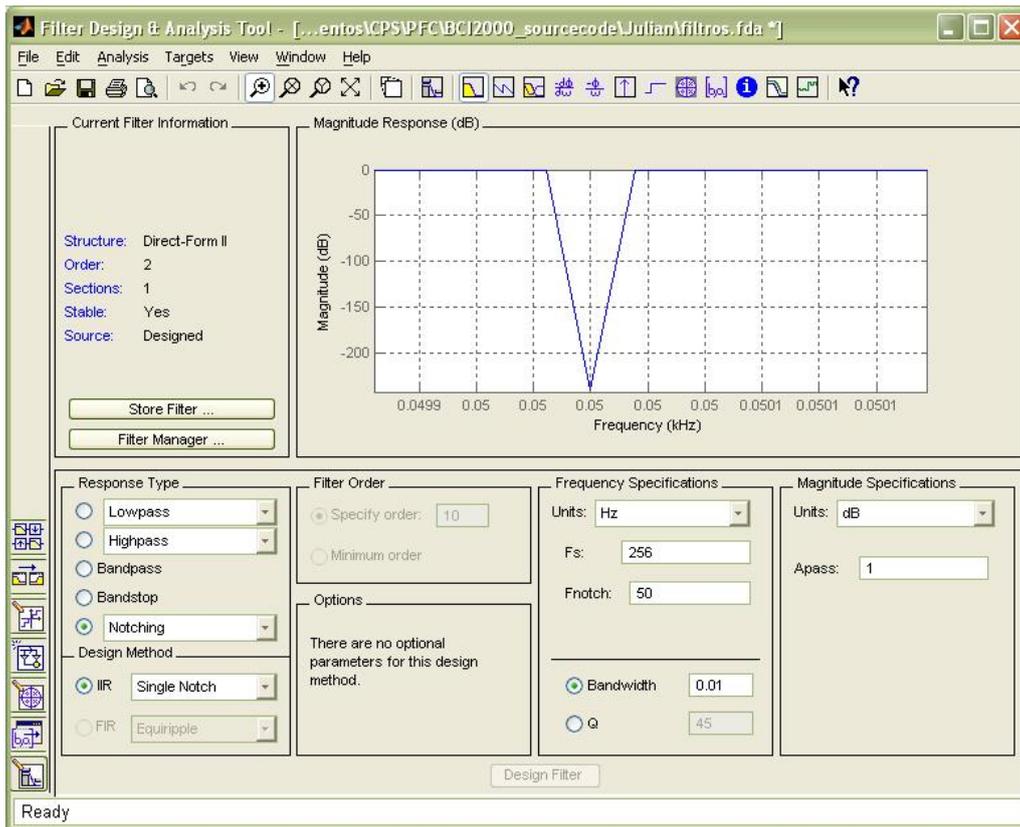
Tomando los datos del biosensor ECG, trataremos de obtener los siguientes parámetros:

- Heart Rate o tasa de latidos (HR) en media y varianza.
- Heart Rate Variability (HRV) o variabilidad de la tasa de latidos, que no es igual a la varianza estadística del HR, como veremos más adelante.

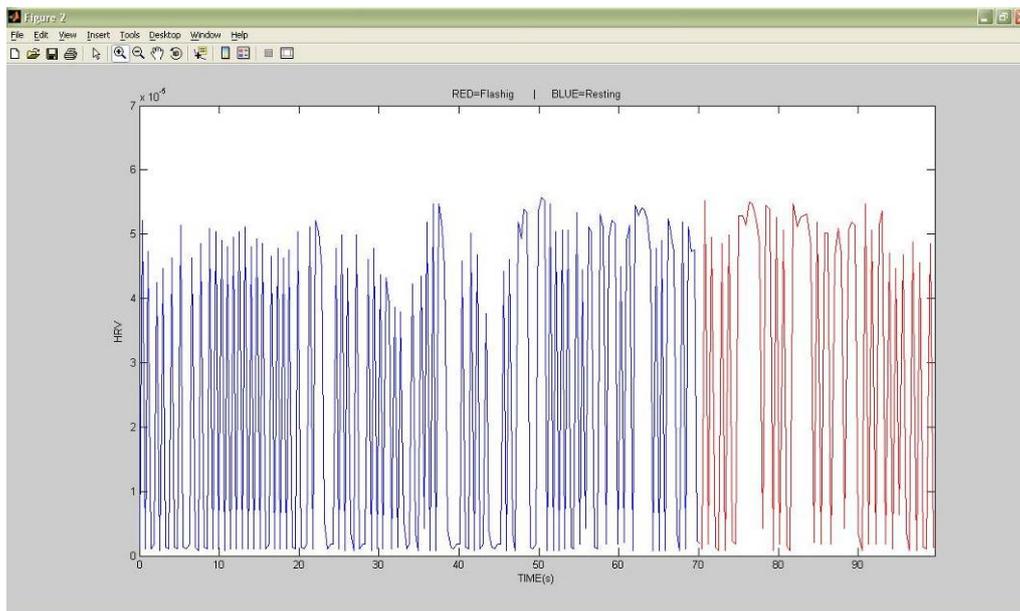
Para este cometido seguiremos la línea marcada por los investigadores del Departamento de Ingeniería Eléctrica de la Universidad de Brasilia, que desarrolló una herramienta para el análisis del HRV en MATLAB llamada ECGlab. Nos basaremos principalmente en su algoritmo para la detección de la onda QRS:

- Filtro pasabajo en 30 Hz para ruido en altas frecuencias.
- Filtro pasobanda en 17 Hz con factor  $Q = 3$ , que optimiza la relación señal ruido (SNR).
- Calculamos la derivada de la señal, para estrechar los picos (ondas R).
- Elevamos la señal al cuadrado, para hacer los picos más altos.
- Filtro de media móvil con ventana de tamaño 150 ms, para eliminar los picos que no sean ondas R.
- Detectamos la amplitud y localización de las ondas R, con un umbral dinámico (obtenemos una lista).
- Cálculo del Heart Rate instantáneo mediante la diferencia entre puntos consecutivos de la lista anterior.
- Cálculo de la media y varianza del HR en cada intervalo.
- Representación del HR instantáneo a lo largo del vector de tiempo: Heart Rate Variability (HRV).

Diseñaremos los filtros, tanto para este biosensor como para el resto, en la 'toolbox' de MATLAB 'Filter Design and Analysis Tool', y después los guardaremos como variables del 'workspace' para poder aplicarlos a las señales en los programas que realicemos. Para detectar los picos llamaremos a una función de MATLAB llamada 'pkpicker.m'. El resto de los cálculos están implementados en el código.



**Fig. 6.3.1.1: Diseño y exportación de filtros en la herramienta FDA de MATLAB.**



**Fig. 6.3.1.2: HRV: Representación en el tiempo del HR instantáneo.**

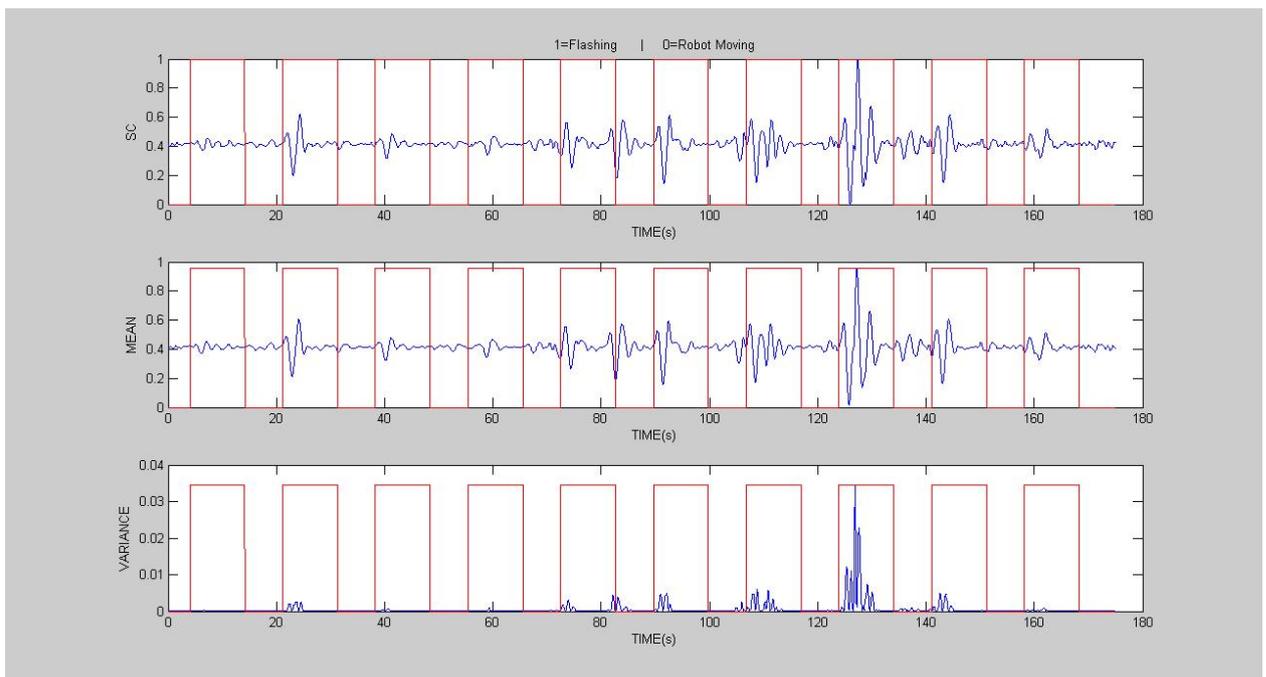
Vemos como el HRV no es más que una gráfica del Heart Rate en función del tiempo, de manera que se puede observar a simple vista cómo varía y diferenciar entre estados de relajación y estrés. Del mismo modo, un clasificador basado en redes neuronales puede analizar estos datos y efectuar una separación de las clases, que será mejor cuanto más variación haya en el HR entre los distintos estados.

### 6.3.2.- Procesamiento de la señal SC

Para el tratamiento de esta señal nos basaremos en los trabajos sobre reconocimiento de emociones llevados a cabo en los laboratorios Sony Corporate Europe en Stuttgart, Alemania, así como los métodos de extracción de características y clasificación de emociones utilizados por el Instituto de Ciencias Computacionales de la Universidad de Augsburg, Alemania. Por tratarse de una señal de baja frecuencia, implementaremos un algoritmo que haga lo siguiente:

- Filtro pasabajo a 4 Hz.
- Normalización (valores entre 0 y 1), ya que sólo queremos cálculos relativos.
- Cálculo de media y varianza móviles mediante una ventana de tamaño  $N=100$  muestras (unos 390 ms).

Nos quedaremos, finalmente, con la propia señal SC filtrada y normalizada y la media y varianza móviles como características para las entradas del clasificador.



**Fig. 6.3.2.1: Representación de las características extraídas de la señal SC.**

### 6.3.3.- Procesamiento de la señal RSP

En este caso, al igual que en el apartado anterior, seguiremos la metodología de los estudios realizados por Haag, Goronzy, Schaich y Williams en los laboratorios Sony Corporate Europe en Stuttgart, y por Wagner, Kim y André en el Instituto de Ciencias Computacionales de la Universidad de Augsburg, ambos en Alemania, para extraer nuestras características a partir de las señales recibidas:

- Eliminamos nivel de continua, substrayendo a las señales su media global.
- Filtro pasabajo a 10 Hz.
- Filtro de media móvil, con ventana de 150 ms.
- Encontramos los cruces por cero (los cruces demasiado cercanos se consideran artefactos y se eliminan).
- Cálculo de las respiraciones por minuto, de una forma similar a como hicimos en el ECG.
- Cálculo de la media y varianza de los ciclos por minuto en cada intervalo de actividad y de descanso.
- Cálculo de la profundidad de la respiración, como la amplitud media del vector de datos en cada intervalo. Calcularemos también su varianza.

De este modo, obtendremos cinco características: respiraciones por minuto (RPM), media y varianza de las mismas en cada intervalo, y profundidad de la respiración (PROF), en media y varianza en cada intervalo.

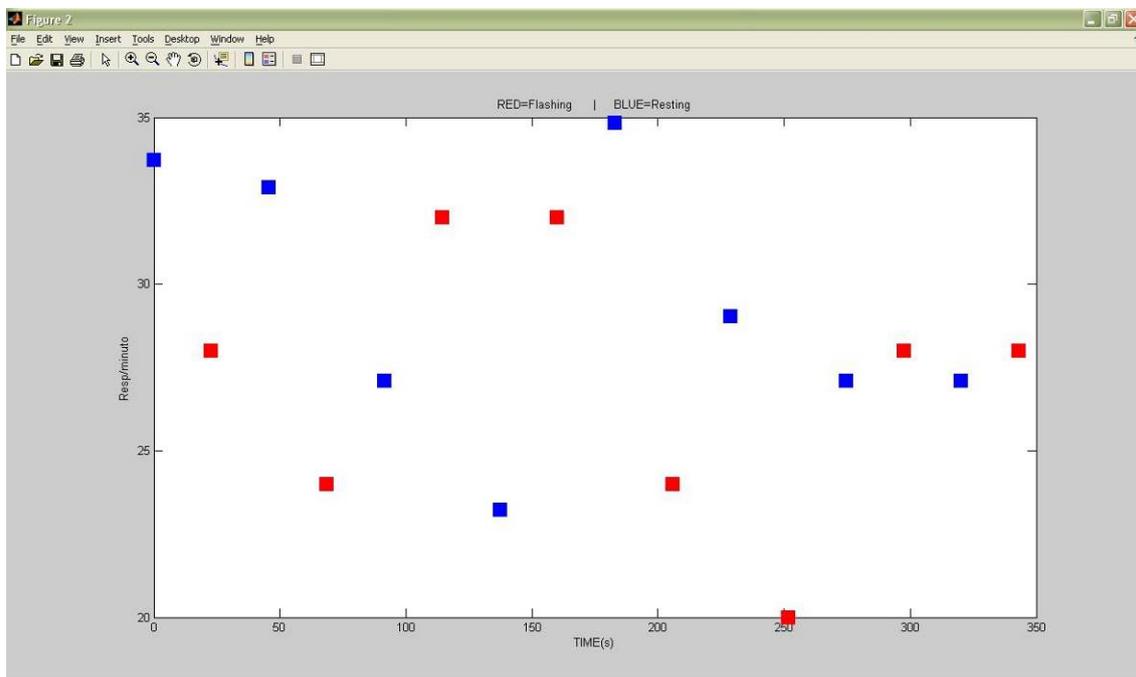


Fig. 6.3.3.1: Respiraciones por minuto (RPM) en media.

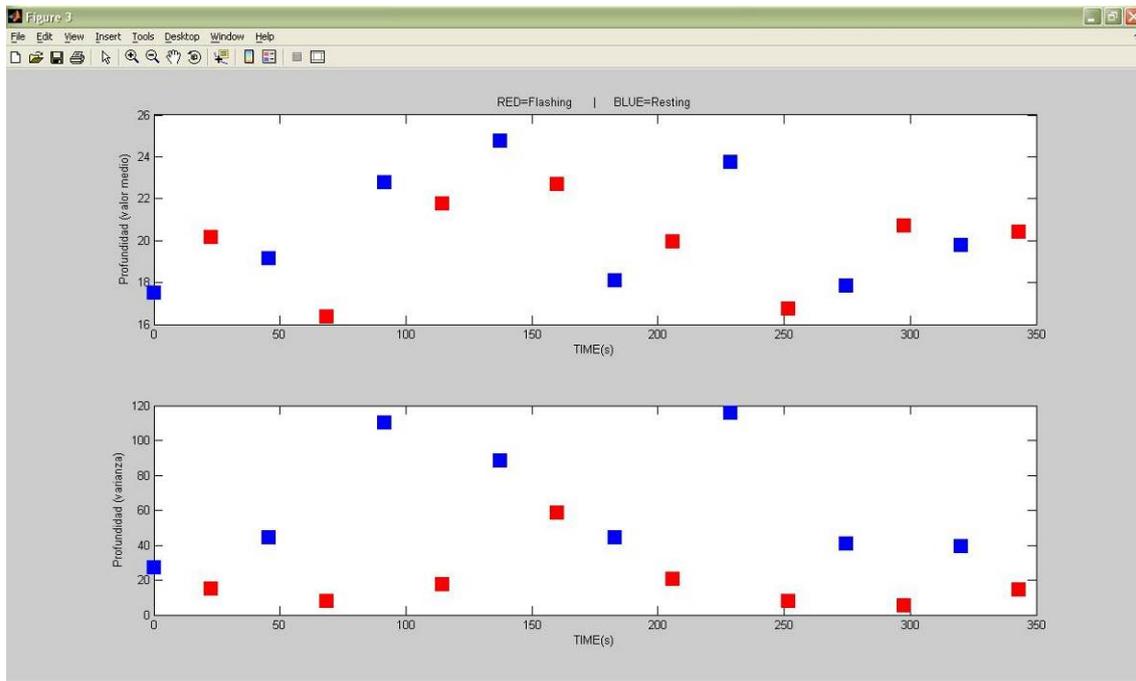


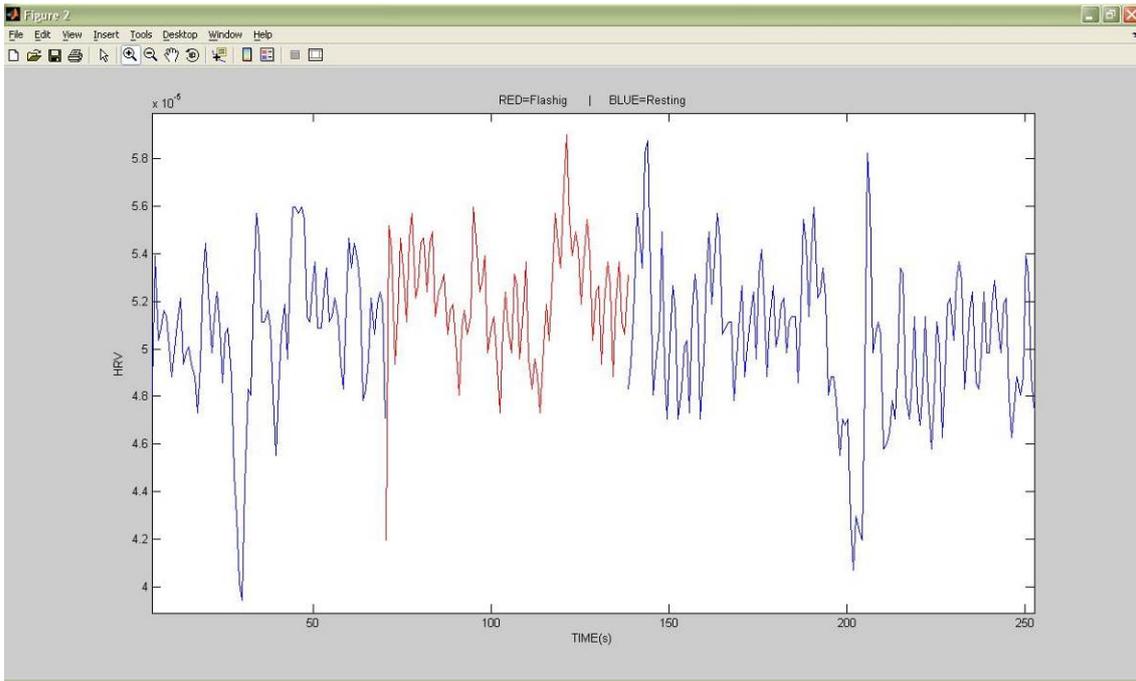
Fig. 6.3.3.2: Profundidad de la respiración (PROF), en media y varianza.

### 6.3.4.- Procesamiento de la señal BVP

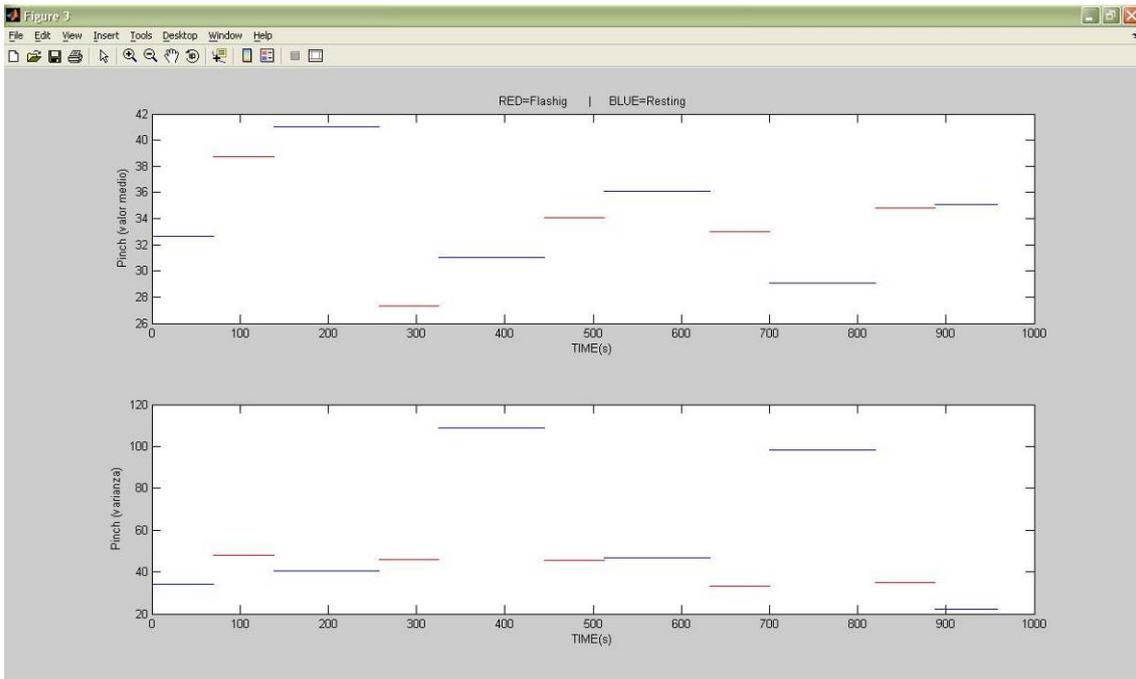
Para este biosensor seguiremos exclusivamente las indicaciones de Haag, Goronzy, Schaich y Williams para extraer el HR y la dilatación vascular o “pinch”. En cuanto al HR, el procedimiento es similar al realizado para el ECG, y la finalidad es tener redundancia en esta importante característica para mejorar la fiabilidad del clasificador y la robustez del sistema. En cuanto a la dilatación vascular en los vasos sanguíneos, esta variable se corresponde con la amplitud de la señal en cada intervalo de actividad o descanso, por lo que se calculará del mismo modo que la profundidad de la respiración. Los pasos a seguir serán:

- Restamos a las señales su media global.
- Filtramos LP en 90 Hz para eliminar el ruido de alta frecuencia.
- Para el cálculo del HR: Hacemos una suma ponderada de las derivadas primera y segunda de la señal.
- Filtro de media móvil con ventana de 150 ms (a la frecuencia de muestreo de 256 Hz, unas 38 muestras).
- Detectamos la localización de los picos del mismo modo que para el ECG.
- Calculamos la dilatación vascular a partir de la señal original filtrada, tal como hicimos con la profundidad de la respiración.

Por lo tanto, las características obtenidas serán: BVPHRV, BVPHR en media y varianza en cada intervalo y dilatación vascular o PINCH, así como su media y varianza en cada intervalo.



**Fig. 6.3.4.1: Representación del HRV obtenido a partir del biosensor BVP.**



**Fig. 6.3.4.2: Dilatación vascular (PINCH) en media y varianza.**

### 6.3.5.- Procesamiento del EEG

En este subcapítulo describiremos el procesado de los canales del EEG de cara a la posterior clasificación que permita diferenciar entre estados de relajación y estrés en operación, de modo que las características que extraigamos formarán parte, junto con las obtenidas a partir de los biosensores, del conjunto de entradas (o “capa” de entrada, para ir introduciendo la terminología usada en redes neuronales) del clasificador. El procesamiento de esta misma señal para el estudio del ADHD y el ‘workload’ será descrito en capítulos posteriores.

Para procesar estos datos, seguiremos la metodología usada anteriormente para otros proyectos por el grupo de trabajo del Área de BCI del Centro Politécnico Superior (con algunos cambios para adaptarla a nuestros requerimientos), ya que en dichos proyectos usaban también un clasificador, aunque de una naturaleza diferente al que implementaremos en este proyecto (se trata de un clasificador de tipo estadístico, no está basado en redes neuronales).

Tenemos, en principio, 16 ó 24 (según el experimento) vectores de MATLAB correspondientes al registro de los canales de EEG durante el tiempo que dura la prueba, cada uno de los cuales dividiremos previamente en intervalos de actividad y descanso, cuyo número dependerá del protocolo del experimento en cuestión. Agruparemos los datos de actividad y los de descanso, de manera que obtendremos, por ejemplo en el caso de los 24 canales, 48 vectores de muestras, dos (actividad y descanso) por cada canal. A cada uno de ellos se le aplicará el siguiente tratamiento de señal:

- Cálculo de las FFT de cada uno con una ventana móvil.
- Implementación del algoritmo r-cuadrado, para identificar los canales más representativos o con más información (aquellos que presenten mayor varianza entre los datos correspondientes a actividad o descanso).
- Representación del espectrograma, para poder observar las diferencias en el r-cuadrado entre los canales y seleccionar los más adecuados.
- Representación topográfica (mapeo) del r-cuadrado para cada ventana.
- Cálculo de la potencia en las bandas de frecuencia de cada canal seleccionado para actividad y descanso.
- Cálculo del ratio THETA/BETA en cada canal seleccionado para actividad y descanso.
- Elección de los parámetros a exportar y colocación de los resultados en una matriz apropiada para el clasificador.

De esta manera obtendremos, finalmente, una matriz, que contendrá las muestras de potencia en las bandas de frecuencias deseadas (además del ratio THETA/BETA si se desea) y en los canales deseados (número de filas = número de canales \* número de bandas; número de columnas = número de muestras).

## 6.4.- Clasificación de estados y análisis de tendencias

Una vez tengamos procesadas todas las señales procedentes de todos los sensores, la siguiente tarea a realizar será buscar los estados de relajación, estrés, y agotamiento o fatiga. Esta detección debe ser lo más fiable y robusta que se pueda, y además, se debe realizar la clasificación de manera automática, en la medida de lo posible.

El procesamiento o extracción de características nos ha revelado qué variables nos pueden ayudar a realizar la clasificación entre estados de relajación y estrés, basándonos en estudios e hipótesis previas. Ahora es el momento de construir el detector de estados. Asimismo, también hemos visto en las numerosas investigaciones consultadas, que para encontrar índices de agotamiento o fatiga, es conveniente observar la tendencia de algunas de estas variables a lo largo del tiempo, con lo que haremos una importante separación aquí: para clasificar estados de relajación y estrés (que son opuestos) utilizaremos un programa automático, mientras que para detectar el agotamiento nos basaremos en análisis cualitativos de tendencias temporales.

En el primer caso, existen varias alternativas para implementar el software. Se puede optar por programas algorítmicos, muy costosos y de una complejidad enorme, que tal vez fueran eficaces para hacer una predicción, pero que serían difícilmente adaptables a cambios, y necesitarían una cantidad exagerada de recursos para llevarla a cabo. Sin embargo, dada la naturaleza del problema, se hace evidente que lo más eficaz para trabajar con estos datos es, o bien un clasificador de tipo estadístico (como por ejemplo ANOVA, muy utilizado en los estudios precedentes), o bien uno basado en redes neuronales artificiales (RNA), como pueden ser los mapas auto-organizados, el perceptrón o la adalina, o las funciones de base radial (RBF).

Por el reto que supone desarrollar una herramienta diferente y novedosa, y para poder comprobar si con ella se consiguen alcanzar resultados de una fiabilidad similar a los de los citados estudios, elegiremos finalmente basar nuestro clasificador en redes neuronales del tipo MLP (perceptrón multicapa), que además de haber sido ratificado como aproximador universal, es muy apropiado para resolver problemas de clasificación fuertemente no lineales, como el que nos podemos encontrar en este caso.

En cuanto al agotamiento o fatiga, utilizaremos herramientas de regresión para realizar un análisis del cual podamos extraer conclusiones que estén en concordancia con las investigaciones previas.

## 6.5.- Presentación de resultados

Respecto a los estados de relajación/estrés, todos los programas que se elaboren en este proyecto generarán a su término gráficos en los que podamos ver de forma cualitativa las diferencias entre estados para todas las variables consideradas de interés. Después de eso, tendremos la ocasión de supervisar las fases de entrenamiento y evaluación de la red neuronal que usaremos como clasificador, y cuando éstas finalicen, obtendremos valores de un índice conocido como índice kappa, que están entre 0 y 1, y que sirven para evaluar la bondad del clasificador, así como gráficas en los que se podrá comprobar su capacidad de predicción. Se omitirán al usuario otros datos estadísticos o técnicos, de cara a una mayor claridad y concisión.

Para el agotamiento, el análisis de tendencia se podrá presentar en varios formatos, bien en EXCEL, trabajando sobre datos exportados de MATLAB, bien elaborando dicho análisis mediante la toolbox “Curve Fitting Tool” de MATLAB, o bien directamente en nuestros programas de MATLAB, haciendo uso de las funciones que este entorno pone a nuestra disposición para ello.

En ambos casos, la idea es implementar un GUI (Graphic User Interface) de MATLAB, de modo que podamos manejar la ejecución de los programas que compondrán el proyecto con comodidad y sencillez, utilizando una interfaz gráfica con menús y botones, como si de una toolbox de MATLAB se tratara.

Esta interfaz se verá en capítulos posteriores, donde se incluirá una especie de pequeño tutorial para que podamos ver con detalle su funcionamiento.