



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco
Departamento de Computación

**Sistema para la detección de oscilaciones y el análisis
multifrecuencia en registros EEG del sueño humano**

Tesis que presenta

Imanti Méndez Fontus

para obtener el Grado de

Maestro en Ciencias
en Computación

Directores

Amilcar Meneses Viveros

José Luis Cantero Lorente

México, D.F.

6 de octubre de 2014

Resumen

En este trabajo se presenta un sistema integral diseñado para automatizar el procesamiento de datos requerido para detectar ondas lentas y husos de sueño, y analizar las relaciones temporales y fase-amplitud entre estas oscilaciones en registros EEG del sueño humano. Investigaciones recientes relacionan a las ondas lentas, oscilaciones con frecuencias menores a 1 Hz presentes en las fases 3 y 4 del sueño, y a los husos de sueño, ráfagas de actividad entre 11 Hz y 15 Hz con duración de 0,5 a 3 segundos, con el proceso de consolidación de memoria durante el sueño humano. Para comprobar estas hipótesis, los neurocientíficos requieren herramientas de cómputo que automaticen las tareas de procesamiento de datos sobre los registros de electroencefalografía. El sistema presentado ofrece herramientas para reportar la estructura del sueño, detectar ondas lentas y husos del sueño durante las etapas NREM del sueño y para analizar la relación temporal mediante histogramas de correlación y la relación fase-amplitud usando una métrica de acoplamiento multifrecuencia. Todos los resultados son validados estadísticamente para evaluar la confiabilidad de los datos al ser comparados entre grupos o condiciones experimentales. Las herramientas, desarrolladas en lenguaje Matlab®, pueden ser utilizadas por medio de interfaces intuitivas en línea de comandos diseñadas para permitir al usuario configurar y ejecutar todas las etapas de procesamiento y análisis de datos. Finalmente, el conjunto de herramientas que componen al sistema, al cual hemos llamado EEGTools, se entrega como una caja de herramientas con licencia GNU pública para uso no comercial acompañado de un manual de usuario y documentación técnica.

Abstract

In this thesis we present a comprehensive set of tools designed to automate the processes required to detect slow waves and sleep spindles, and to analyze temporal and phase-amplitude relations between these oscillations in human sleep EEG recordings. Recent studies postulate the existence of a relation between slow waves, oscillations with frequencies below 1 Hz that appear in sleep stages 3 and 4, and sleep spindles, burst of activity between 11 Hz and 15 Hz with 0,5 to 3 seconds length, with the process known as memory consolidation during human sleep. In order to test these hypothesis, neuroscientists require computational tools to automate processing tasks on electroencephalography recordings. The system described here offers tools to report sleep structure, to detect slow waves and sleep spindles during NREM sleep stages and to analyze temporal relations through correlation histograms and phase-amplitude relations using cross-frequency coupling. The results of each task, after being compared between groups or experimental conditions, are validated to find statistical significance. All tools, written in Matlab®, offer intuitive command line interfaces designed to allow the user configure and execute every step of the workflow. Finally, all the tools that compose the system, called EEGTools, are packed as a Matlab® toolbox available with a GNU public license for noncommercial use, and accompanied with documentation and a user manual.

Agradecimientos

Quisiera agradecer a mis directores de tesis, los doctores José Luis Cantero y Amílcar Meneses, por su gran disposición y la confianza que tuvieron en mi trabajo, a la Dra. Mercedes Atienza por su guía y enseñanzas, y en especial a todos los integrantes del Laboratorio de Neurociencia Funcional por su colaboración, buena disposición y simpatía. Me gustaría también dedicar esta tesis a todos mis amigos, compañeros y familia por su apoyo incondicional. En especial a mi padre, a Michele y a Martina por acompañarme y soportarme durante esta etapa.

Finalmente, agradezco al Consejo de Ciencia y Tecnología (CONACYT) y al Centro de Investigación y de Estudios Avanzados (CINVESTAV) por su apoyo y patrocinio para la realización de este proyecto de tesis.

Índice general

Resumen	III
Abstract	V
Agradecimientos	VII
Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
1.1. Estado del Arte	3
1.1.1. Cajas de herramientas para neurocientíficos	4
1.1.2. EEGLab	4
1.1.3. FieldTrip	5
1.1.4. BioSig Project	5
1.2. Objetivos	6
1.2.1. General	6
1.2.2. Específicos	6
1.3. Justificación	7
1.4. Contribuciones	7
1.5. Estructura de la tesis	8
2. Marco Teórico	11
2.1. Fundamentos neurofisiológicos	11
2.1.1. Conceptos	11
2.1.2. Electroencefalografía	17
2.1.3. Características del sueño	20
2.2. Tratamiento de señales EEG	22
2.2.1. Tipos de señales	23
2.2.2. Operaciones básicas	24
2.2.3. Transformada discreta de Fourier	25
2.2.4. Espectro de potencia	26
2.2.5. Transformada rápida de Fourier	27

2.2.6.	Transformada Wavelet	27
2.2.7.	Transformada de Hilbert	29
3.	Diseño del sistema	33
3.1.	Metodología experimental y requerimientos del sistema	33
3.2.	Detección de ondas lentas	37
3.2.1.	Detección por percentiles	38
3.2.2.	Estadísticas	39
3.3.	Detección de husos del sueño	39
3.3.1.	Modificaciones al algoritmo de detección	40
3.3.2.	Estadísticas	43
3.4.	Histograma de correlación	43
3.4.1.	Estadísticas	44
3.5.	Métrica de acoplamiento multifrecuencia	46
3.5.1.	Índice de modulación de Tort	47
3.5.2.	Estadísticas	48
4.	Desarrollo del sistema	51
4.1.	Estructura del sistema	51
4.2.	Herramientas básicas	52
4.2.1.	Lectura y almacenamiento temporal	52
4.2.2.	Filtros digitales	53
4.2.3.	Lectura y procesamiento de marcas	54
4.2.4.	Hipnograma y estructura del sueño	55
4.3.	Detección de ondas lentas	56
4.3.1.	Detección	57
4.3.2.	Promedio de sujetos	58
4.3.3.	Estadísticas	58
4.4.	Detección de husos del sueño	60
4.4.1.	Detección	60
4.4.2.	Promedio de sujetos	62
4.4.3.	Estadísticas	62
4.5.	Histogramas de correlación	63
4.5.1.	Cálculo del histograma	63
4.5.2.	Promedio de sujetos	64
4.5.3.	Figura por grupo o condición	64
4.5.4.	Estadísticas	65
4.6.	Análisis de frecuencia	67
4.6.1.	Gráficas de tiempo-frecuencia	68
4.6.2.	Acoplamiento multifrecuencia	70
5.	Pruebas	73

6. Conclusiones **79**
6.1. Trabajo futuro 80

Índice de figuras

2.1. Anatomía de la neurona	12
2.2. Anatomía del sistema nervioso central	15
2.3. Lóbulos del cerebro	16
2.4. Ejes y planos de medición del cerebro	16
2.5. Sistema Internacional (SI) 10-20	18
2.6. Actividad eléctrica durante las fases del sueño	21
2.7. Ciclo de retroalimentación tálamo-cortical y de generación de husos del sueño	22
2.8. Espectro de potencia	27
2.9. Función Wavelet	28
2.10. Envoltente de amplitud	30
3.1. Flujo de procesamiento.	34
3.2. Protocolo de registro EEG	35
3.3. Detección ondas lentas	37
3.4. Detección de husos del sueño	41
3.5. Ejemplo histograma de correlación	43
3.6. Tipos de sincronización multifrecuencia	46
3.7. Proceso para calcular el MI	48
4.1. Ejemplo de hipnograma	56
4.2. Promedio de ondas lentas por grupo o condición	59
4.3. Comparación de histogramas por grupo o condición	65
4.4. Estadísticas de histograma por grupo o condición.	66
4.5. Gráficas de tiempo-frecuencia.	69
4.6. Estadísticas acoplamiento multifrecuencia por grupo o condición.	71
5.1. Sincronía temporal entre los husos y las ondas lentas	77
5.2. Sincronía temporal entre los husos y las ondas lentas	77

Índice de cuadros

2.1. Bandas de ritmo EEG	19
4.1. Identificadores para fases de sueño	55
5.1. Macro estructura del sueño	75
5.2. Husos del sueño	76

Capítulo 1

Introducción

En este proyecto de tesis se presenta un sistema que automatiza las tareas de procesamiento y análisis de registros electroencefalográficos (EEG) durante el sueño humano. El sistema incluye herramientas para la detección de patrones oscilatorios dentro de las señales eléctricas y el posterior análisis de sincronización entre los patrones detectados. El diseño del sistema está motivado por los requerimientos de la investigación que se lleva a cabo en el Laboratorio de Neurociencia Funcional, Universidad Pablo de Olavide (Sevilla, España). En particular, el sistema ofrece herramientas que ayudan a los investigadores con el estudio de la consolidación de la memoria durante el sueño humano.

Existen muchas las evidencias que apoyan el papel del sueño en los procesos de memoria, sobre todo en los procesos de consolidación [1, 2, 3, 4]. Según la hipótesis dominante en neurociencias, los procesos de consolidación hacen referencia a los procesos cerebrales que, por una parte, facilitan la transformación de las representaciones neuronales iniciales (codificación) en representaciones más resistentes a posibles interferencias (estabilización); y que, por otra, promueven la integración de dichas representaciones con representaciones preexistentes [5].

Aunque los mecanismos neuronales que subyacen a este proceso de consolidación no se conocen bien, algunos de los eventos electrofisiológicos que ocurren durante el SOL (Sueño de Ondas Lentas; ver sección 2.1.3), como la actividad de husos y de ondas lentas generadas en circuitos tálamo-corticales, se postulan como posibles correlatos neurales[6, 7].

La actividad de husos oscila entre 11 Hz y 15 Hz , y es un fenómeno característico de la fase 2 del sueño aunque persiste durante el SOL [8]. En los registros EEG los husos pueden ser identificados como ráfagas de actividad con una duración aproximada de 0,5 a 3 segundos. Los husos de sueño están a su vez relacionados temporalmente con las oscilaciones lentas [9]. Estas oscilaciones con una frecuencia inferior a 1 Hz alternan estados de hiperpolarización (down-state, parte negativa de la onda) asociados a silencio neuronal con estados de despolarización (up-state, parte positiva de la onda) caracterizados por un incremento en la excitabilidad neuronal[10].

Basado en el modelo de memoria en dos etapas (codificación y estabilización), la hipótesis dominante propone que la consolidación de memoria durante el SOL es un proceso orquestado por las ondas lentas y que involucra la reactivación de representaciones

recientemente codificadas (memoria transitoria dependiente del hipocampo) y su subsecuente redistribución desde el hipocampo a sitios de almacenamiento persistente en la neocorteza. La consolidación se traduciría en un aumento de la sincronía entre las ondas lentas y la actividad de husos.

Los niveles de sincronización entre ondas lentas y husos en registros EEG del sueño humano han sido ampliamente estudiados, y se han relacionado fundamentalmente con los procesos de consolidación de la memoria [11]. Este tipo de análisis involucra el cálculo de las coincidencias temporales entre ambos patrones. El presente trabajo pretende dar un paso más estudiando no solo la sincronía temporal sino también la posible sincronía de fase-amplitud que pueda existir entre los dos tipos de frecuencia que caracterizan a estos dos eventos electrofisiológicos. En particular, se pretende evaluar si la fase de las frecuencias más lentas (ondas lentas) modula la amplitud de las frecuencias más rápidas (huso del sueño).

A las técnicas matemáticas diseñadas para el estudio de la sincronización entre bandas de frecuencias se les conoce como acoplamiento multifrecuencia [12]. Existen diversos estudios que han mostrado la importancia de la acoplamiento multifrecuencia en procesos funcionales. Por ejemplo, el acoplamiento entre las bandas teta ($[4\ 8]Hz$) y gamma ($[25\ 100]Hz$) parece estar relacionada con la representación de múltiples objetos en la memoria de trabajo [13], lo cual sugiere la existencia de un mecanismo capaz de controlar el procesamiento serial de memorias a corto plazo [12]. Sin embargo, a nuestro conocimiento, en ningún caso se han aplicado algoritmos de acoplamiento multifrecuencia para analizar la relación entre ondas lentas y husos de sueño.

Dada la gran efectividad de los algoritmos de acoplamiento multifrecuencia para encontrar relaciones entre la fase y amplitud de distintas bandas de frecuencia, en este trabajo se plantea adaptar uno de estos algoritmos para ser usado en registros EEG (electroencefalografía) de sueño humano, con el fin de extender el conocimiento que se tiene sobre la relación entre ondas lentas y husos del sueño.

El sistema que se propone en este trabajo tiene como propósito automatizar las tareas requeridas para hacer el análisis de sincronización entre ondas lentas y husos de sueño. Muchas de estas tareas se llevan a cabo con paquetes de software diversos o *scripts* de Matlab® aislados. Es por eso que se encuentra la necesidad de diseñar e implementar un sistema que incorpore todas estas tareas en un paquete de herramientas único, que ofrezca a los investigadores un modo conveniente de configurar y ejecutar cada experimento.

Visto de manera general, las herramientas que debe incluir el sistema pueden ser catalogadas en tres etapas: herramientas para detección de patrones, para análisis de sincronización temporal y para análisis de acoplamiento multifrecuencia.

Para realizar la detección de ondas lentas se hizo uso de un algoritmo estándar que ha sido ampliamente usado en trabajos previos [14, 15]. Este algoritmo utiliza la definición electrofisiológica de las ondas lentas para establecer criterios de aceptación de las oscilaciones presentes durante la fase SOL. Por otro lado, para la detección de husos de sueño se usó el algoritmo propuesto por Mölle et al. 2009 [11], con algunas modificaciones.

En el caso del análisis de sincronización temporal se decidió implementar la técnica de histogramas de correlación descrita por Mölle et al. 2009 ([11]). Utilizando estas figuras

es posible visualizar la ocurrencia simultánea de los husos de sueño y las ondas lentas.

El análisis de acoplamiento multifrecuencia se hizo mediante el índice de modulación (MI por sus siglas en inglés) propuesto por Tort [16]. Este algoritmo ha sido utilizado en diversos estudios con el fin de analizar las relaciones fase-amplitud entre oscilaciones en distintas bandas de frecuencia, y para este trabajo, se adaptará para el análisis de la relación entre ondas lentas y husos del sueño en registros EEG del sueño humano.

Finalmente, los algoritmos y herramientas implementados fueron organizados en una aplicación capaz de automatizar el proceso de análisis de sincronización entre ondas lentas y husos del sueño en registros EEG del sueño humano.

1.1. Estado del Arte

Los algoritmos elegidos para realizar la detección de oscilaciones (ondas lentas y husos de sueño) son estándares en el procesamiento de registros EEG. Ambos utilizan la descripción electrofisiológica de dichos eventos para llevar a cabo la detección, por lo tanto imitan el análisis que lleva a cabo un experto al marcar estos eventos manualmente.

Como es de esperar, existen propuestas de algoritmos más eficientes y precisos para detectar tanto ondas lentas [17, 18] como husos del sueño [19, 20, 21, 22]. Sin embargo, ninguna de estas propuestas ha logrado desplazar a los algoritmos estándar. Es posible que esto se deba a dos razones:

- Para diseñar un nuevo algoritmo es necesario crear una metodología estandarizada para medir su efectividad y precisión. A nuestro conocimiento, existe una sola base de datos ¹ de registros EEG para entrenar o probar nuevas propuestas de algoritmos. Esto hace que la validez de los resultados de nuevas propuestas disminuya considerablemente.
- Las características de la actividad eléctrica cerebral pueden variar enormemente según condiciones como la edad o el estado de salud de los pacientes. Gracias a esto, un algoritmo que funciona adecuadamente con un paciente puede requerir cambios significativos para poder ser aplicado con otro, y los neurocientíficos optan por diseñar algoritmos a la medida de cada situación.

Por otro lado, existen diversos métodos para el cálculo del acoplamiento fase-amplitud (PAC por sus siglas en inglés) y ninguno de ellos ha sido establecido como un estándar. Estas métricas ofrecen ventajas y desventajas una con respecto a la otra y pueden ser utilizadas según el problema que se desee estudiar [23, 24, 25].

En este trabajo se ha decidido utilizar el algoritmo de índice de modulación propuesto por Tort et al. [16] (que llamaremos algoritmo de Tort). Este algoritmo no solo es uno de los más referenciado en la literatura si no que ofrece las siguientes ventajas con respecto a los demás:

¹<http://www.tcts.fpms.ac.be/devuyst/Databases/DatabaseSpindles/>

- El algoritmo de Tort arroja mejores resultados [16] en segmentos de señal más pequeños comparado con el algoritmo de índice de modulación propuesto por Canolty et al. [26].
- Para señales con alta relación señal-ruido, el algoritmo de Tort arroja resultados más certeros [25] que otros algoritmos como la métrica señal-envolvente [27] y la coherencia multifrecuencia [28].

1.1.1. Cajas de herramientas para neurocientíficos

Las herramientas que se revisan a continuación nacieron por un requerimiento similar al sistema que se presenta en este proyecto. Cuando los investigadores de neurociencias encuentran la necesidad de procesar registros automáticamente suelen crear herramientas aisladas que facilitan tareas como lectura, escritura, filtrado, análisis o despliegue de datos. Con el paso del tiempo estas herramientas son mejoradas e integradas entre sí hasta formar cajas de herramientas que pueden ser reutilizadas en estudios posteriores.

Fue importante analizar estas herramientas por dos motivos:

- Es posible estudiar la organización de estas cajas de herramientas para evitar repetir errores en el momento del diseño de nuestro sistema. Todas estas cajas de herramientas han sufrido considerables cambios en un afán de encontrar la estructura que mejor se acomode al trabajo de los neurocientíficos. Siguiendo estos patrones de diseño podemos minimizar los periodos de re-diseño relacionados con, por ejemplo, interfaces poco intuitivas para el usuario o estructuras de datos poco versátiles en el ambiente de programación de Matlab®.
- Gracias a que estas herramientas fueron diseñadas modularmente y se ofrecen al público con licencias de código abierto, es posible reutilizar algunas de sus características. En muchos casos, las operaciones frecuentes como el filtrado de las señales o la lectura y escritura de datos se pueden importar directamente de estas herramientas y así evitar no solamente la inversión en tiempo que significaría desarrollarlas desde el inicio sino también la aparición de errores comunes que hayan sido previamente resueltos por los desarrolladores del software.

Finalmente, cabe señalar que se han escogido las herramientas más utilizadas por neurocientíficos en la actualidad. Existen muchas más herramientas de este tipo que abordan problemas muy específicos del área, sin embargo, al estudiar estas tres herramientas, es posible hacerse una idea clara del modelo de ingeniería de software requerido.

1.1.2. EEGLab

*EEGLab*² [29] es una caja de herramientas para lenguaje Matlab® que permite analizar registros EEG y MEG tanto continuos como relacionados a eventos. Esta herramienta

²<http://sccn.ucsd.edu/eeglab/>

incorpora funciones para realizar análisis de componentes independientes (ICA por su nombre en inglés) que se utiliza extensamente para realizar análisis espacio-temporal de registros EEG.

Entre las características principales de este software están:

- Importa y exporta diversos formatos de registro EEG y MEG
- Ofrece gráficas interactivas para registros de gran tamaño
- Ofrece un mecanismo para agregar componentes al software base
- Ofrece un mecanismo semi-automático para remoción de artefactos
- Ofrece funciones para realizar descomposiciones ICA y tiempo/frecuencia
- Cuenta con una interfaz de usuario eficiente y amigable

Dentro del sistema presentado en este proyecto se hace uso del *script* encargado de aplicar filtros de impulso finito (FIR) basados en la transformada de Fourier. Este *script* tiene las ventajas de ser robusto y optimizado.

1.1.3. FieldTrip

*FieldTrip*³ [30] es también una caja de herramientas para lenguaje Matlab® que ofrece distintos tipos de análisis para registros EEG y MEG. Esta herramienta, aunque carece de una interfaz gráfica de usuario, posee una estructura modularizada que permite al usuario configurar y ejecutar una gran variedad de operaciones.

Los análisis más importantes que ofrece son:

- Importa y exporta diversos formatos de registro EEG y MEG
- Ofrece técnicas para análisis tiempo-frecuencia
- Incluye algoritmos para calcular la reconstrucción de fuentes
- Ofrece pruebas estadísticas no paramétricas a nivel de electrodo o fuente
- Incluye algoritmos para el análisis de conectividad funcional y efectiva⁴

FieldTrip es un paquete de software maduro que ofrece una gran variedad de herramientas. En nuestro caso, si bien no se logró completar este modulo, utilizamos las herramientas que ofrece para el cálculo de la reconstrucción de fuentes; este módulo tendría como objetivo descartar la ocurrencia simultanea en diversos canales de las ondas lentas detectadas.

1.1.4. BioSig Project

*BioSig Project*⁵ es un software de código libre para procesamiento de señales biomédicas. Esta programado en lenguaje C++ aunque ofrece traducciones tanto a Matlab® como

³<http://fieldtrip.fcdonders.nl/>

⁴La conectividad cerebral se refiere a los patrones generados por conexiones anatómicas (conectividad anatómica), dependencias estadísticas (conectividad funcional) o interacciones de causalidad (conectividad efectiva) entre distintas unidades dentro del sistema nervioso. Estas unidades pueden ser neuronas, poblaciones de neuronas o regiones cerebrales anatómicamente segregadas [31].

⁵<http://biosig.sourceforge.net/>

a Simulink®. Este software ofrece muchas de las herramientas mencionadas tanto en *EEGLab* como en *Fieldtrip*, sin embargo, debido a sus largos tiempos de desarrollo, no es usada con tanta frecuencia como las anteriores.

Se ha decidido mencionar BioSig en esta sección porque ofrece funciones robustas y optimizadas para importar y exportar formatos de registros EEG y MEG. Tanto *EEGLab* como *Fieldtrip*, así como muchos otros paquetes de software, utilizan BioSig para llevar a cabo la lectura y escritura de datos, y nuestro caso no es la excepción. El módulo de adquisición de datos que ofrece BioSig se encuentra incluido en nuestro sistema y es el encargado de leer los registros originales antes de ser convertidos a un formato más adecuado para los propósitos de nuestro sistema.

1.2. Objetivos

1.2.1. General

Implementar un sistema para automatizar la detección y análisis de relaciones en tiempo y frecuencia entre ondas lentas y husos del sueño en registros de electroencefalografía de sueño humano.

1.2.2. Específicos

1. Implementar el algoritmo estándar [14, 15] para la detección de ondas lentas.
2. Implementar el algoritmo para la detección de husos de sueño basado en la propuesta de Mölle et al. 2009 [11].
3. Implementar un algoritmo para analizar la relación temporal entre patrones basado en la propuesta de Mölle et al. 2009 [11].
4. Implementar un algoritmo para analizar el acoplamiento multifrecuencia basado en la propuesta de Tort et al. 2010 [16].
5. Diseñar un conjunto de herramientas que facilite las tareas de lectura, pre y post procesamiento de señales EEG, requeridas por los algoritmos de detección de patrones y análisis de sincronización.
6. Diseñar pruebas estadísticas para validar los resultados arrojados por los algoritmos de detección de patrones y análisis de sincronización.
7. Integrar los algoritmos y herramientas estadísticas diseñadas en un sistema robusto y versátil que ofrezca la posibilidad de configurar y ejecutar experimentos de manera automática.

1.3. Justificación

En neurociencias el análisis de datos es una tarea de vital importancia para la comprobación de sus hipótesis experimentales. Comparado con otras áreas, el procesamiento y análisis de datos en neurociencias muestra tres peculiaridades:

- Los algoritmos y técnicas utilizadas para extraer información relevante de los datos pueden variar con mucha frecuencia debido a la constante innovación en el área.
- El volumen de los datos que componen los registros EEG durante el sueño, que generalmente incluyen 59 canales con una tasa de muestreo de al menos 250 Hz durante aproximadamente 8 horas, hace que la tarea de almacenamiento, procesamiento y análisis de los datos sea computacionalmente costosa.
- Para el procesamiento de datos se hace uso de diversas herramientas con características y requerimientos variados. En algunos casos esto obliga a los investigadores a cambiar constantemente de ambiente de trabajo (i.e. sistema operativo o configuración de *hardware*) para completar el procesamiento.

Estas condiciones en la metodología experimental de neurociencias abren un amplio espectro de oportunidades a la aplicación de nuevas y mejores tecnologías de cómputo. Las áreas de *big data*, procesamiento de señales y algoritmia son las que mayor aplicación pueden encontrar en los sistemas desarrollados para neurocientíficos, sin embargo, incluso áreas como inteligencia artificial o cómputo paralelo pueden encontrar nichos de aplicación en este tipo de sistemas.

En vista de los requerimientos descritos anteriormente, durante el diseño e implementación de la herramienta que se presenta en este proyecto se hará especial énfasis en dos aspectos fundamentales:

- El sistema que se desarrolle debe ser modular y altamente configurable. Los constantes cambios en las técnicas y algoritmos obligan a que el sistema pueda ser modificado sin requerir cambios mayores que obligarían a re-diseñarlo.
- Se requieren mecanismos optimizados de lectura y almacenamiento de datos, capaces de disminuir considerablemente los tiempos de respuesta del sistema en vista del volumen de datos que se maneja.
- El sistema debe automatizar el flujo de trabajo requerido en un único ambiente de trabajo. Es importante que la interfaz de usuario sea amigable e imite, en la medida de lo posible, los procesos manuales utilizados con anterioridad.

1.4. Contribuciones

El sistema que se presenta en este proyecto de tesis fue diseñado para automatizar las tareas requeridas para el análisis de las relaciones existentes entre ondas lentas y husos del

sueño en registros EEG humanos. Estas tareas pueden ser divididas en tres módulos generales: detección de patrones (ondas lentas y husos de sueño), análisis de la sincronización temporal entre patrones y análisis de la sincronización fase-amplitud entre patrones.

Existen diversas herramientas que, a partir de las necesidades de la investigación neurocientífica, están diseñadas para facilitar tareas como lectura, escritura, filtrado, análisis o despliegue de datos de registros EEG. Como ya se explicó anteriormente, las necesidades en la investigación crecen constantemente, lo cual obliga a estas herramientas a ser actualizadas o rediseñadas. Dadas estas condiciones encontramos que nuestro sistema ofrece las siguientes contribuciones o mejoras con respecto a otras herramientas:

- Se diseñó una estructura base para ejecutar y reportar los resultados de algoritmos de detección de patrones. Esta estructura permite la inclusión de nuevos algoritmos o modificaciones de los anteriores con el fin de encontrar aquel que cumpla con los requisitos del tipo de análisis que se realice. Si, por ejemplo, es necesario detectar los husos de sueño en pacientes con algún tipo de enfermedad, para los cuales las características de los husos varían significativamente con respecto a pacientes sanos, es posible implementar e incluir nuevos algoritmos en el sistema de manera rápida y sencilla.
- Tanto en la caja de herramientas fieldtrip ⁶ [30] como en el estudio presentado por Onslow et al. ⁷ [25] se encuentran implementaciones del algoritmo de índice de modulación [26], predecesor del algoritmo de Tort. Sin embargo, a nuestro conocimiento, no existe una implementación pública del algoritmo de Tort. En el sistema presentado aquí se ofrece dicha implementación acompañada de documentación que facilita la aplicación del algoritmo.

1.5. Estructura de la tesis

El presente documento se encuentra organizado en seis capítulos. El primer capítulo, como se ha podido leer, incluye el flujo de procesos requerido para analizar un estudio del sueño, la motivación fisiológica y computacional detrás del desarrollo del sistema y los objetivos del presente trabajo.

En el capítulo 2 se introduce la anatomía del sistema nervioso haciendo énfasis en la organización del cerebro y la morfología de las neuronas. Posteriormente se describen las características más importantes de las mediciones EEG; el montaje estándar, la polisomnografía y el almacenamiento y digitalización de las señales. Finalmente se da una explicación detallada sobre las etapas del sueño y los distintos patrones de actividad eléctrica presentes en los registros EEG del sueño humano.

Posteriormente, el capítulo 3 presenta una revisión de las herramientas más comunes en el tratamiento de imágenes. Se inicia listando algunos operadores básicos. A continuación

⁶<http://fieldtrip.fcdonders.nl/>

⁷<http://www.cs.bris.ac.uk/Research/MachineLearning/pac/>

se detalla el origen y aplicación de las transformadas *Wavelet*, de Fourier y de Hilbert, que serán ampliamente utilizadas en los algoritmos del presente trabajo.

En el cuarto capítulo se explica con mayor detalle los procesos involucrados en el análisis de un estudio del sueño. Después de delinear el flujo de procesos, se detalla la organización modular del sistema desarrollado y finalmente se explica en detalle los algoritmos para detección de patrones oscilatorios así como los que serán utilizados para analizar la relación temporal y fase-amplitud entre los patrones detectados.

Los pormenores de la implementación del sistema se explican en el quinto capítulo. En él se listan las herramientas más importantes que componen cada módulo, cómo pueden ser ejecutadas y qué tipo de resultados arrojan. Se aprovecha este capítulo para indicar los aspectos técnicos más significativos como funciones de Matlab® utilizadas para filtrar señales, los formatos de los archivos de registro y reporte.

Finalmente, en el capítulo 6 se realiza un análisis y discusión sobre los resultados. Al término de éste capítulo se incluyen las conclusiones, valoración de objetivos iniciales y los futuros trabajos de investigación que pueden derivarse de esta tesis.

Capítulo 2

Marco Teórico

2.1. Fundamentos neurofisiológicos

La neurofisiología es una rama de la fisiología que estudia el funcionamiento del sistema nervioso. En primer lugar se revisarán los elementos más importantes que integran el sistema nervioso para posteriormente pasar a explicar como puede registrarse de manera no invasiva la actividad eléctrica producida por miles de neuronas que disparan de forma sincrónica con el fin de realizar una función de manera coordinada. Finalmente se describirá el protocolo que ha de seguirse para registrar esta actividad durante un estado fisiológico tan particular como el sueño. Con esta introducción a la neurofisiología se pretende dar un contexto del área en la cual está enmarcado este proyecto.

2.1.1. Conceptos

Neurona

La neurona es la unidad fundamental del sistema nervioso de los vertebrados. Su capacidad de transferir información de una parte del organismo a otra en forma de impulsos electroquímicos la distingue del resto de las células.

Las neuronas tienen tres regiones morfológicas definidas:

- **Cuerpo celular.** También conocido como soma. Es la parte más gruesa, contiene todos los componentes necesarios de la célula, como el núcleo (contiene el ADN), retículo endoplasmático y ribosomas (para sintetizar proteínas), y la mitocondria (para la transformación de energía). Si el núcleo muere, la neurona muere.
- **Axón.** Proyección alargada en forma de cable, se encarga de transportar el mensaje electroquímico a lo largo de toda la longitud de la célula. Dependiendo del tipo de neurona, los axones pueden estar recubiertos por una fina capa de mielina, a modo de aislante. La mielina está constituida principalmente por lípidos o grasas, facilitando la rápida transmisión del impulso nervioso a lo largo del axón. Neuronas mielinizadas se encuentran principalmente en los nervios periféricos, tanto sensores

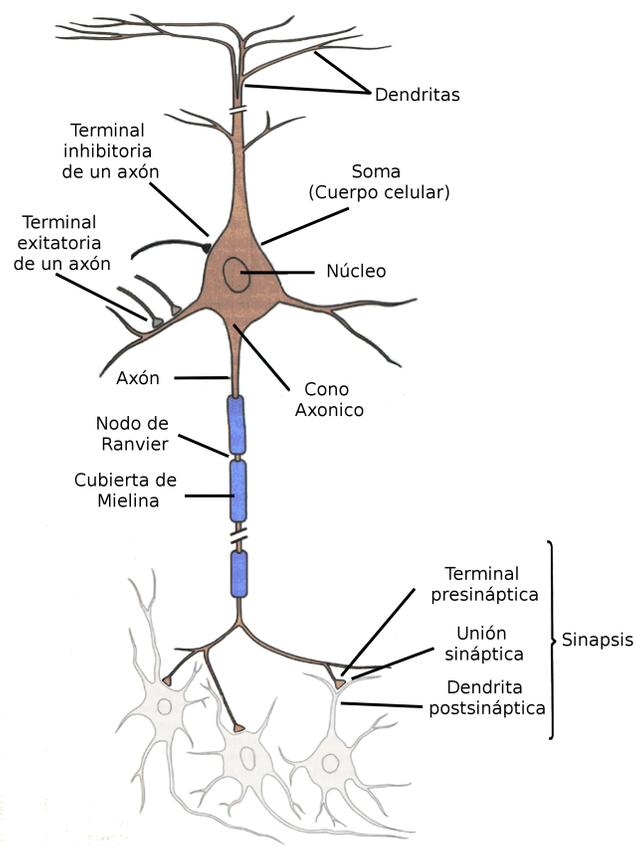


Figura 2.1: Una neurona tiene tres regiones morfológicas: soma, axón y dendritas. El axón, recubierto por mielina, se extiende alejándose del soma hasta las terminales sinápticas. Un potencial de acción inicia en el cono axónico y viaja a través del axón regenerándose en cada nodo de Ranvier, hasta alcanzar alguna de estas terminales. Imagen modificada del libro *Principles of Neural Science* [32].

como motores, mientras que neuronas no mielinizadas se encuentran en el cerebro y la médula espinal, constituyendo lo que se denomina sustancia gris.

- Dendritas. Son pequeñas prolongaciones arborescentes de la célula encargadas de realizar las conexiones con otras neuronas, permitiendo la comunicación entre las mismas o la percepción del entorno exterior.

La longitud de las neuronas es muy variable, puede ir desde unos pocos milímetros hasta algunos decímetros, como en el caso de neuronas cerebrales y táctiles respectivamente. De igual modo la forma también es muy variable dependiendo de la actividad que lleven a cabo, las neuronas motoras que controlan la contracción muscular tienen el cuerpo de la célula en un extremo, un axón largo en el medio y dendritas en el otro extremo; mientras que las neuronas sensoras tienen dendritas en ambos extremos, conectados por largos axones con el cuerpo en el medio.

De forma muy general, las células nerviosas se dividen en neuronas de proyección, cuyos axones abandonan la corteza cerebral para adentrarse en la sustancia blanca, y las interneuronas, con axones confinados a la sustancia gris. Las principales neuronas de

proyección son las neuronas piramidales, que representan el 80 % de todas las neuronas de la corteza cerebral. Los axones de estas neuronas siguen una trayectoria descendente en la que primero emiten colaterales y después envían proyecciones hacia regiones subcorticales, otras áreas corticales del mismo hemisferio, o hacia el hemisferio contralateral. Estas conexiones de largo alcance suelen ser bidireccionales y, por lo tanto, dan lugar a circuitos que se retroalimentan de forma recíproca. Por su parte, las interneuronas forman circuitos locales dentro de la propia corteza, no llegando sus axones a alcanzar la sustancia blanca cerebral. El 80 % de estas neuronas son excitadoras y la mayoría son células estrelladas que reciben proyecciones talámicas, mientras el 20 % restante son inhibitoras y muestran una morfología más diversa que las excitadoras. Esta diversidad no solo es crucial para el procesamiento global de la información, sino que además es responsable de las oscilaciones cerebrales que posteriormente vamos a registrar con el EEG.

El punto de contacto entre dos neuronas, denominado sinapsis, es de especial importancia en la transmisión de información del sistema nervioso, ya que la información de una neurona pasa a otra a través de los mismos. Este pequeño espacio que separa ambas neuronas consta de:

- Terminación presináptica, la cual contiene neurotransmisores, mitocondria y orgánulos celulares.
- Terminación postsináptica, formado por los lugares de recepción de los neurotransmisores.
- Hendidura sináptica, espacio submicroscópico entre ambas terminaciones.

Mientras que en el interior de la neurona la transmisión de información a lo largo del axón se realiza en forma de impulso eléctrico, entre neuronas es un proceso químico, llevado a cabo en la sinapsis, el que transmite la información de una neurona a otra, o bien a una célula muscular.

Transmisión eléctrica. Se dice que una neurona está polarizada cuando tiene una carga eléctrica negativa en el interior de la membrana celular respecto al exterior. Este efecto de polarización se debe a la libre circulación de iones K^+ a través de la membrana celular, a la vez que moléculas grandes con carga negativa son retenidas dentro de la célula. Los iones Na^+ se mantienen en el exterior de la célula mediante un proceso activo. Aunque en principio todas las células del cuerpo presentan esta diferencia de potencial, las neuronas se distinguen en que al aplicar una corriente estimuladora, se provoca que los iones potasio penetren dentro de la célula despolarizándola. Tras un pequeño intervalo de tiempo las propiedades de la membrana cambian haciendo que la célula sea permeable al Na^+ , que entra en ella con rapidez y origina una carga neta positiva en el interior de la neurona. A este efecto se le denomina potencial de acción. Una vez alcanzado éste potencial en una zona de la neurona, éste se propaga a lo largo del axón mediante un intercambio de iones en puntos específicos denominados nodos de Ranvier. La amplitud del potencial de acción es auto limitada, debido a que una concentración elevada de Na^+ en el interior origina la expulsión de la célula en primer lugar de iones K^+ y después de Na^+ , restableciendo la

carga negativa en el interior de la membrana celular, re-polarizando la célula. El proceso completo dura menos de una milésima de segundo. Después de un breve lapso, llamado periodo refractario, la neurona se encuentra en condiciones de repetir este proceso.

Transmisión química. Cuando el potencial de acción alcanza el extremo del axón, éste estimula en la célula unas pequeñas vesículas pre-sinápticas que contienen neurotransmisores, los cuales son liberados en la hendidura sináptica. El neurotransmisor se une a receptores especializados sobre la superficie de la neurona adyacente, provocando su despolarización y la propagación de su propio potencial de acción. La duración de un estímulo procedente de un neurotransmisor está limitada por su degradación en la hendidura sináptica y su recaptación por la neurona que lo había elaborado.

Cuando un neurotransmisor se enlaza en un receptor de la terminación postsináptica, se produce un cambio en la excitabilidad de la célula haciendo más o menos probable que se desencadene un potencial de acción. Si el número de eventos excitatorios postsinápticos es lo suficientemente grande, se provocará un potencial de acción en la célula y el mensaje se transmitirá.

Los potenciales postsinápticos de conjuntos de neuronas sincronizadas constituyen la mayor parte de la actividad eléctrica que la electroencefalografía registra. En particular, se sabe que las neuronas piramidales de la corteza cerebral, al integrar la información proveniente de neuronas del núcleo reticular del tálamo, son las que contribuyen con la porción más significativa de la señal electroencefalografía [33].

Sistema Nervioso

El sistema nervioso es una red de células especializadas que coordina las acciones de un animal y envía señales desde una parte del cuerpo hasta otra. Su estudio se restringe en la mayoría de los casos al análisis de las propiedades de las células excitables que hemos mencionado en el apartado anterior, a pesar de que las células gliales son bastante más numerosas. Esto se debe a que tradicionalmente se ha considerado que la función de estas células está limitada al mantenimiento del tejido neuronal, dada su incapacidad para generar potenciales de acción. Sin embargo, estudios recientes han mostrado que una sola célula glial (concretamente un astrocito) puede realizar decenas de miles de contactos con neuronas cercanas [34, 35].

Desde un punto de vista anatómico, en los vertebrados el sistema nervioso está compuesto por el sistema nervioso central y el sistema nervioso periférico. El primero está compuesto por el encéfalo y la médula espinal, mientras que el segundo por nervios craneales y espinales.

El sistema nervioso central está inmerso en líquido cefalorraquídeo, envuelto dentro de una triple estructura de tejido fino a modo de bolsas, que previenen el contacto directo del encéfalo y médula espinal con las estructuras óseas de protección como el cráneo y la espina dorsal.

El encéfalo a su vez está formado por el cerebro, el cerebelo y el tronco encefálico.

- El cerebro es la parte más voluminosa, ocupa casi la totalidad de la caja craneal. Tiene forma ovoide, con la extremidad posterior más ancha. Su peso en el hombre

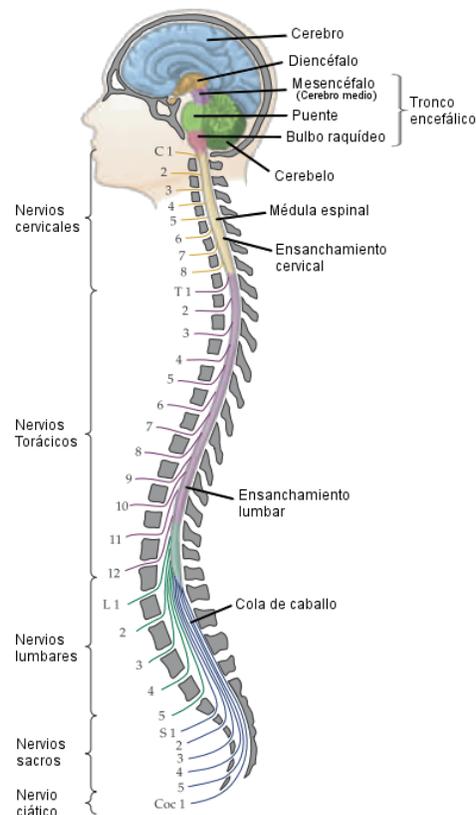


Figura 2.2: Anatomía del sistema nervioso central en el humano. Para su estudio, se divide en siete secciones o subsistemas: médula espinal, bulbo raquídeo, puente o tronco encefálico, cerebelo, mesencéfalo, diencéfalo y hemisferios cerebrales. Imagen modificada del libro Neuroscience [36].

es de aproximadamente 1 Kg.

- El cerebelo es la parte inferior y posterior del encéfalo, alojado en la fosa cerebral posterior junto al tronco encefálico.
- El tronco encefálico está compuesto por el diencéfalo, el mesencéfalo, el puente troncoencefálico y el bulbo raquídeo.
- La médula espinal es la prolongación del encéfalo, extendiéndose por la columna vertebral; la misma sustancia gris que forma el cerebro se encuentra en el interior y la blanca en el exterior.

Los nervios del sistema nervioso periférico emergen del sistema nervioso central y recorren todo el cuerpo, conteniendo axones de vías neurales con distintas funciones y ganglios periféricos, que se encuentran en el trayecto de los nervios y que contienen cuerpos neuronales, los únicos fuera del sistema nervioso central. Los 12 nervios craneales transmiten información sensorial procedente del cuello y la cabeza hacia el sistema nervioso central, y reciben órdenes motoras para el control de la musculatura esquelética de los mismos. Los

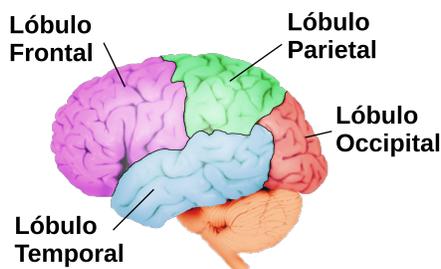


Figura 2.3: Lóbulos del cerebro. Cada uno es nombrado según el hueso craneal en que se ubica.

31 pares de nervios espinales (ver Figura 2.2) salen desde las vértebras y se distribuyen por las regiones del tronco y las extremidades. Se encargan de enviar información sensorial del tronco y las extremidades hacia el sistema nervioso, y reciben órdenes motoras para el control de la musculatura esquelética de los mismos.

Cerebro

El cerebro es el órgano más grande del sistema nervioso central. Está situado en el interior del cráneo, consta de 100 000 millones de neuronas y llega a pesar aproximadamente 1500 gramos.

Existen diversas subdivisiones anatómicas del cerebro. La más notoria es la división formada por una fisura longitudinal que lo divide en dos hemisferios (izquierdo y derecho) de aproximadamente igual tamaño y forma. En los humanos, los hemisferios cerebrales son proporcionalmente más grandes que en cualquier otro animal, y están caracterizados por circunvoluciones (o crestas) de tejido cortical separadas por cisuras (o surcos). Aunque los patrones generados por las circunvoluciones o cisuras varían de un individuo a otro, existen marcas consistentes que ayudan a dividir los hemisferios en cuatro lóbulos. Los lóbulos, ver Figura 2.3, son nombrados según el hueso del cráneo en que se ubican: frontal, temporal, parietal y occipital.

Para referenciar posiciones o direcciones en el sistema nervioso, y en el cerebro, se

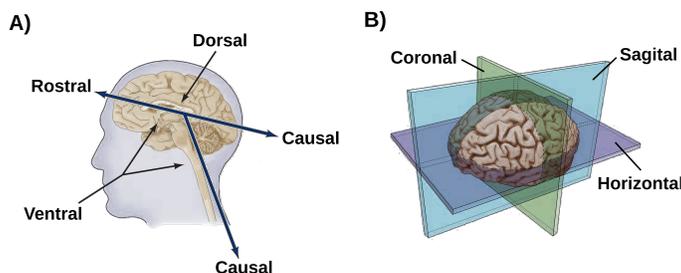


Figura 2.4: Ejes y planos de medición del cerebro. A) Se utilizan las direcciones rostro-caudal (anterior-posterior) y ventro-dorsal (inferior-superior) para referenciar puntos en el cuerpo humano. B) A partir de las direcciones se generan planos de medición utilizados en imagenología o para realizar cortes histológicos.

utilizan los mismos términos que en la anatomía del cuerpo completo, ver Figura 2.4A. Los términos rostral y caudal se utilizan para referirse a las partes delantera y trasera del cuerpo respectivamente (anterior y posterior; hacia la cabeza y la cola). Los términos ventral y dorsal se utilizan para referirse al vientre y el dorso respectivamente (inferior y superior; hacia abajo y arriba del cuerpo). En el caso de la mayoría de los animales, estas direcciones (junto con la dirección lateral) forman un espacio de tres ejes que puede describir la posición de cualquier parte del cuerpo. En el caso del encéfalo de los animales bípedos, el eje rostrocaudal sufre una inclinación debido a la posición de la masa encefálica en la cavidad craneal: la dirección rostrocaudal está definida tanto por la línea formada desde el lóbulo frontal hasta el lóbulo occipital, como por la línea marcada por la médula espinal.

Estos ejes generan planos estándar que son utilizados para realizar cortes histológicos o para llevar a cabo imagenología cerebral, ver Figura 2.4B. El plano horizontal corre paralelo al eje rostrocaudal, desde el lóbulo frontal hasta el lóbulo occipital. El plano sagital es el que divide a los dos hemisferios del cerebro. Finalmente, el plano coronal (o frontal) tiene dirección dorsoventral y es ortogonal a los planos horizontal y sagital. Cabe señalar que estos planos son utilizados únicamente en el encéfalo, en la médula espinal o en otras partes del cuerpo estos planos sufren algunas variaciones.

2.1.2. Electroencefalografía

Para el registro de la actividad eléctrica del cerebro se pueden utilizar tres tipos de electrodos: superficiales, corticales e intracorticales. Cuando se emplean electrodos superficiales el registro se denomina electroencefalograma (EEG), si los electrodos se disponen sobre la superficie expuesta del cerebro al registro se denomina electrocorticograma (ECoG), y si se utilizan electrodos intracorticales se emplea el término registro profundo para denominar a esta técnica [33]. Las variaciones de potencial registradas representan los potenciales de acción y fluctuaciones de membrana de las neuronas próximas al electrodo.

Montaje

Los electrodos utilizados en las mediciones de EEG registran la diferencia de voltaje entre sí mismo y una referencia que puede ser un electrodo adyacente, una referencia fija o el promedio ponderado de los electrodos adyacentes. Si se pretende evaluar relaciones de fase o de fase-amplitud se utiliza la referencia promedio, para evitar la inclusión de posibles relaciones espurias. Posteriormente, el valor registrado por el electrodo es amplificado para resultar en una señal con amplitud en el intervalo $[10\ 100]\ \mu V$.

El montaje de los electrodos en la superficie del cuero cabelludo se hace conforme a estándares internacionales. En la Figura 2.5 se muestra la disposición del Sistema Internacional (SI) 10-20 [37]. En este montaje se posiciona cada electrodo a 10% o 20% de distancia del tamaño del cráneo. Aunque si se eleva el número de electrodos se disminuyen estas distancias proporcionalmente. Los canales se nombran utilizando las etiquetas Fp (fronto-polar), F (frontal), T (temporal), O (occipital), C (central), y P (parietal) para

indicar su posición con respecto al eje sagital, y los números pares para el hemisferio derecho e impares para el izquierdo. La letra z se usa para identificar todos los canales en la línea central en el eje sagital.

Polisomnografía

Las mediciones EEG durante el periodo del sueño son conocidas como polisomnografía. Para poder clasificar las diferentes fases que componen el sueño se requiere registrar además de la actividad eléctrica cerebral otras medidas extracerebrales. Al conjunto de todas estas medidas se le conoce como polisomnografía. Por lo regular, el montaje de electrodos durante un estudio del sueño incluye:

- Electroencefalografía: Usando el montaje basado en el SI donde los electrodos son posicionados sobre el cuero cabelludo a distancias de 10 % y 20 % entre si, teniendo como punto de partida de los cuatro puntos fiduciales: nasion, inion y preauriculares derecho e izquierdo.
- Electrooculograma (EOG): Se utilizan dos electrodos posicionados un centímetro por encima de la línea central del ojo derecho y un centímetro por debajo de la línea central del ojo izquierdo respectivamente. Estos electrodos registran la actividad del movimiento de los ojos. Ayuda a diferenciar las etapas del sueño y eliminar artefactos en el registro EEG causados el movimiento ocular.
- Electromiograma (EMG): Se utilizan electrodos para medir la actividad muscular. Dos electrodos son posicionados en el mentón, justo en la línea con la comisura de los labios, y guardando una distancia de 3 cm entre ellos. Al igual que el EOG, estos registros ayudan a diferenciar etapas del sueño y eliminar artefactos introducidos al registro EEG por el movimiento del paciente. En estudios clínicos, también es frecuente colocar dos electrodos en el músculo tibial anterior de ambas piernas.
- Electrocardiograma (ECG o EKG): Se utilizan dos o tres electrodos para medir la actividad del corazón. Los electrodos son posicionados en el pecho por debajo de la

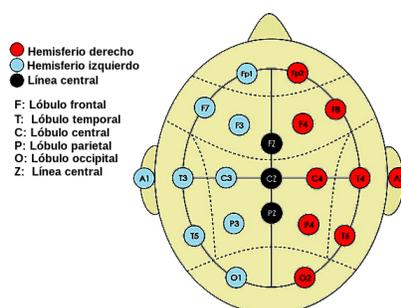


Figura 2.5: Sistema Internacional (SI) 10-20 para montaje de electrodos en mediciones EEG. Contiene 19 electrodos posicionados entre si a 10 % o 20 % del tamaño del cráneo.

clavícula. Este tipo de electrodos suele utilizarse únicamente en la polisomnografía clínica, con el fin de detectar alguna anomalía en el ritmo cardíaco del paciente.

En algunos casos se incluyen también medidores de la respiración y grabaciones del sonido de los ronquidos.

Digitalización de señales

Para almacenar un registro EEG (conjunto de señales de cada electrodo) es necesario digitalizar las señales. Por lo general los fabricantes del equipo de medición marcan los estándares para los parámetros de digitalización, sin embargo es útil saber cuáles son los parámetros mínimos requeridos dependiendo del tipo de estudio que se realizará.

Según el teorema de Nyquist, la frecuencia de muestreo debe ser al menos dos veces mayor que la mayor frecuencia en la señal que se desea analizar. Tomando en cuenta este requisito, se suele sugerir que la frecuencia de muestreo sea tres veces mayor que la mayor frecuencia del análisis [33].

Clasificación de señales

Las señales EEG son divididas regularmente en cinco ritmos distintos. La Tabla 2.1 describe las bandas de frecuencia que caracterizan a cada uno de estos ritmos. Las características de estos ritmos varían con el estado cerebral, y por lo tanto con la fase del sueño y con la edad.

El ritmo delta se observa principalmente durante el sueño de ondas lentas, y muestra amplitudes relativamente grandes. Adicionalmente, el ritmo delta puede ocurrir en casos de lesiones cerebrales, estado de coma y durante anestesia. El ritmo teta puede observarse en estados de vigilia asociados al movimiento y a tareas de memoria así como en diferentes fases del sueño. Casi todo el conocimiento adquirido en relación con este ritmo se deriva de estudios en roedores, pero es importante señalar que el ritmo teta generado en el hipocampo de roedores no guarda ninguna relación con el ritmo teta que se genera en los humanos [38]. El ritmo alfa domina durante los periodos de relajación con los ojos cerrados, antes de que el sujeto entre en algún estado del sueño. El ritmo alfa tiene mayores amplitudes en la región occipital. El ritmo beta está asociado con el estado de vigilia y concentración. El ritmo gamma está relacionado con la percepción y la memoria [33].

Cuadro 2.1: Definición de las bandas de frecuencia para los distintos ritmos en registros EEG [33].

Ritmo EEG	Banda de frecuencias (Hz)
Delta (δ)	< 4
Teta (θ)	$4 - 7$
Alfa (α)	$8 - 13$
Beta (β)	$13 - 30$
Gama (γ)	> 30

Cabe señalar que no existe una definición universal para los rangos de cada banda de frecuencia; los valores presentados aquí son mayormente aceptados por la comunidad científica, aunque pueden llegar a variar ligeramente según cada autor.

2.1.3. Características del sueño

La actividad eléctrica del sueño puede ser catalogada según los componentes de frecuencia y amplitud que presentan las señales registradas. Esta actividad nos revela el patrón de campos eléctricos corticales que caracteriza a cada estado funcional del cerebro [39], facilitando así el estudio de la arquitectura del sueño y la distinción de otros estados alterados de conciencia como el coma y la anestesia.

Rechtschaffen y Kales (R&K) fueron los primeros en describir los estados de vigilia en humanos y propusieron un sistema de clasificación utilizando dos fases principales [8]: fase con movimientos oculares rápidos (REM por sus siglas en inglés), y fase sin movimientos oculares rápidos (NREM por sus siglas en inglés). La última la dividieron a su vez en sueño ligero (fases 1 y 2) y sueño profundo (fases 3 y 4).

La guía propuesta por R&K ha sido ampliamente utilizada para realizar la detección visual de las fases del sueño en registros de polisomnografía. Sin embargo, en años recientes, la Academia Americana de la Medicina del Sueño (AASM) realizó una actualización a la guía propuesta por R&K, con el propósito de clarificar el estudio de las transiciones entre cada fase [40]. Aunque esta actualización deja intacta la mayoría de la clasificación de R&K, la AASM incluye algunas recomendaciones que simplifican la detección de patrones en cada fase; por ejemplo, en la nueva guía se fusionan las fases tres y cuatro para crear una sola fase conocida como sueño de ondas lentas. A continuación se describen las características principales que presentan los registros EEG de cada fase del sueño.

La actividad EEG que caracteriza a la vigilia, ver Figura 2.6, presenta una composición de altas frecuencias (ritmo beta: [15 35] Hz) mezclada con ritmo teta ([4 7] Hz) y un ritmo alfa prominente ([8 12] Hz) si el sujeto está relajado y con los ojos cerrados. La transición de la vigilia al sueño viene acompañada de actividad alfa intermitente en regiones occipitales y oscilaciones teta en regiones de la corteza frontal y sensoriomotora, junto con una actividad electrooculográfica de movimientos oculares lentos. A este estado cerebral se le denomina Fase 1 del sueño.

La Fase 2 del sueño, ver Figura 2.6, se caracteriza fundamentalmente por la presencia de dos eventos fásicos, los husos de sueño (oscilaciones de [10 15] Hz con amplitudes superiores a $15 \mu V$) y los complejos K (inflexiones de voltaje con amplitudes comprendidas en [75 150] μV y frecuencias dominantes de aproximadamente 1 Hz).

Los husos de sueño suelen aparecer bilateralmente en diversas regiones de la corteza cerebral, presentando una distribución topográfica frontocentral en las primeras horas de sueño y desplazándose hacia regiones parietales a medida que transcurre la noche. La distribución topográfica de los husos también depende de su frecuencia dominante [41]. Los husos son fundamentalmente inducidos por despolarizaciones de las neuronas GABAérgicas del núcleo reticular del tálamo, y pueden ser registradas tanto en la corteza cerebral (en células individuales y en potenciales de campo) como en neuronas del núcleo

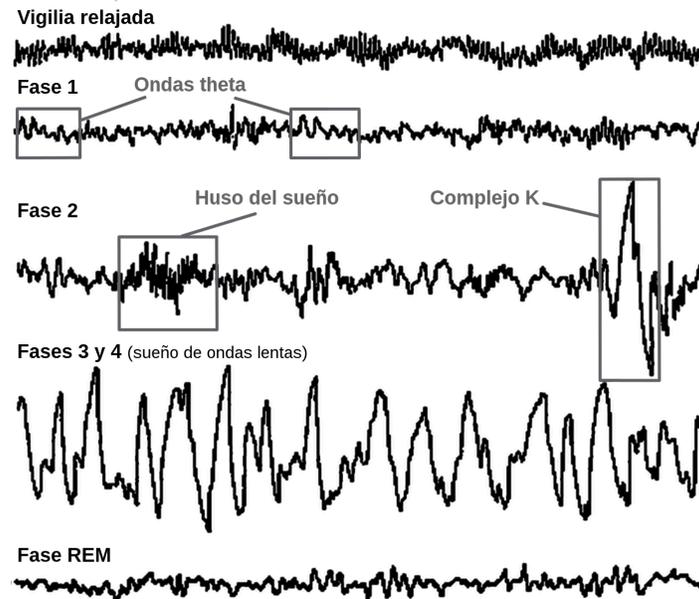


Figura 2.6: Actividad eléctrica durante las fases del sueño. La vigilia relajada está compuesta en su mayoría por actividad en la banda beta. La fase 1 corresponde a la transición entre vigilia y sueño; se caracteriza por una combinación de oscilaciones en las bandas alfa y teta, y movimientos oculares lentos. La fase 2 presenta dos eventos fásicos: husos de sueño y complejos K. Las fases 3 y 4, conocidas en conjunto como sueño de ondas lentas (SOL), están caracterizadas por ondas de gran amplitud y frecuencias en la banda delta. La fase REM se compone en su mayoría por actividad en la banda beta; en esta etapa aparecen movimientos oculares rápidos y se produce una atonía de la musculatura esquelética.

reticular del tálamo [42]. La Figura 2.7 muestra el circuito neuronal que participa en la generación de husos del sueño, junto con registros de EEG y celulares en corteza y en el núcleo reticular del tálamo.

Por otra parte, los complejos K presentan una topografía que se extiende por regiones frontocentrales de la corteza cerebral [43]. Mientras que los husos del sueño resultan de complejas interacciones entre el núcleo reticular del tálamo y la corteza siendo su función principal inhibir el procesamiento sensorial que procede del medio externo [44, 45], los complejos K participan en el procesamiento sensorial residual que se produce mientras dormimos evitándose así una excesiva fragmentación del sueño derivada de los cambios externos [46, 43].

Las Fases 3 y 4 del sueño, ver Figura 2.6, integran el denominado SOL o sueño profundo, caracterizado por ondas de gran amplitud ($[75\ 300]\ \mu V$) en el rango de frecuencias del ritmo delta ($[0.5\ 3\ \text{Hz}]$). Estas ondas lentas del sueño consisten en despolarizaciones rítmicas seguidas por hiperpolarizaciones prolongadas ($[0.2\ 0.8]\ \text{s.}$) de neuronas piramidales tálamo-corticales [47]. Expandir un poco; en Niedermeyer's mencionan que hay dos orígenes propuestos de esta actividad.

Al conjunto de estados cerebrales, que comienzan en la Fase 1 y acaban en la Fase 4, se les conoce como sueño NREM. El patrón EEG que caracteriza a la fase REM, ver Figura 2.6, se asemeja al de vigilia en tanto que presenta un patrón de EEG desincronizado (be-

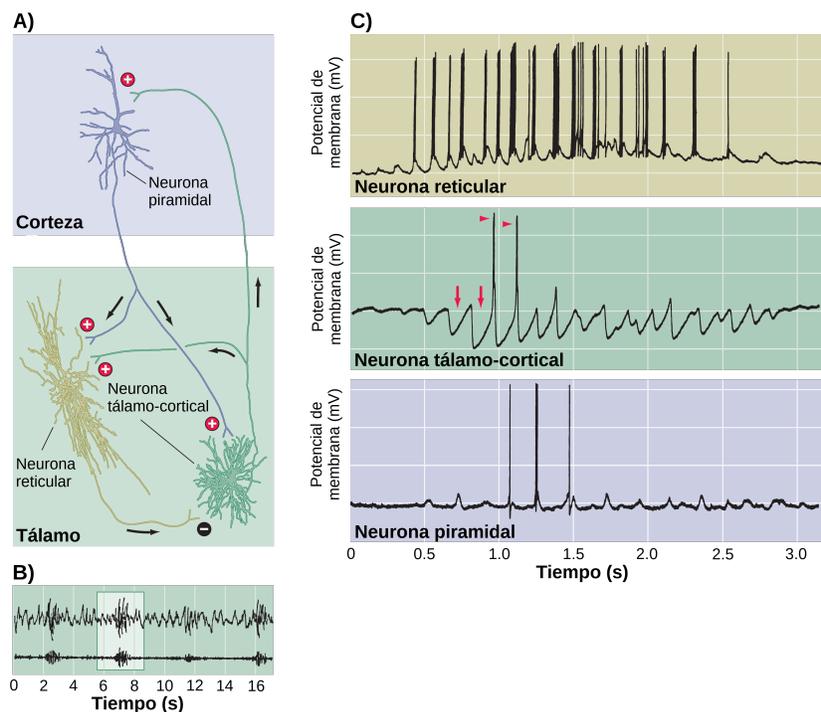


Figura 2.7: Ciclo de retroalimentación tálamo-cortical y de generación de husos del sueño. A) Muestra las conexiones excitatorias (+) e inhibitorias (-) entre neuronas tálamo-corticales, reticulares y piramidales. Estas conexiones son las causantes la aparición de husos del sueño. B) Registro EEG de los husos del sueño; la señal inferior fue filtrada para mostrar con nitidez los husos del sueño. C) El potencial de membrana de una neurona tálamo-cortical, reticular y piramidal respectivamente. Las descargas de la neurona tálamo-cortical dan pie a un tren de actividad en la neurona piramidal, lo cual genera un huso del sueño en el registro EEG. Imagen modificada del libro Neuroscience [36].

ta: 15-35 Hz) mezclado con oscilaciones teta (4-7 Hz) y brotes espontáneos de actividad alfa [48]. Sin embargo, y en claro contraste con la vigilia, durante la fase REM aparecen movimientos oculares rápidos y se produce una atonía casi total de la musculatura esquelética (a excepción de músculos extraoculares, músculos del oído medio y musculatura respiratoria) [49].

2.2. Tratamiento de señales EEG

En esta sección se explican algunas de las herramientas matemáticas básicas en el área de tratamiento de señales. En las siguientes secciones del documento se describirán los algoritmos utilizados durante el proyecto y se hará referencia a estas herramientas. Cabe señalar que se describirán solo aquellas que están relacionadas con los algoritmos utilizados en este proyecto.

En la primera sección del capítulo se presenta una introducción al tratamiento de señales así como las características principales de las señales en registros EEG (Electroencefalografía).

La segunda sección describe algunas de las operaciones más básicas en el tratamiento de señales. La mayoría de estas técnicas son también utilizadas en áreas como estadística o teoría de la información [50].

Finalmente, las últimas tres secciones explican los fundamentos matemáticos y aplicaciones de las tres transformadas más comunes en el tratamiento de señales: transformada de Fourier, transformada de Hilbert y transformada *Wavelet*. Cualquiera de estas tres herramientas resulta indispensable en el análisis de señales que provienen de procesos complejos o incluyen demasiado ruido.

2.2.1. Tipos de señales

El tratamiento o procesamiento de señales es la técnica que estudia la representación, transformación y manipulación de señales. Una señal es la cuantificación de cualquier proceso físico que varía en el tiempo, espacio o cualquier otra variable independiente. Las señales pueden pertenecer a una o más de las siguientes categorías [50]:

- **Analógica, discreta o digital:** una señal analógica es representada como una función continua del tiempo; en cualquier punto en el tiempo puede tomar valores continuos. Una señal discreta está representada en instantes discretos en el tiempo; en cada instancia del tiempo puede tomar valores continuos. Una señal digital es una señal discreta en la que, por cada instancia del tiempo, puede tomar un valor discreto.
- **Señales periódicas y no periódicas:** una señal periódica repite sus valores en instantes de tiempo que son múltiplos del periodo de la señal.
- **Señales determinísticas y aleatorias:** una señal determinística es una función en el tiempo con una estructura conocida y predecible. Por el contrario, una señal aleatoria no puede ser predicha con exactitud; únicamente se pueden estimar sus valores a través de propiedades estadísticas.
- **Señales estacionarias y no-estacionarias:** las señales estacionarias aun siendo aleatorias y no periódicas presentan propiedades estadísticas relativamente constantes en el tiempo; por el contrario, para las señales no estacionarias, es imposible predecir sus propiedades estadísticas. En concreto, se usa una función de distribución acumulativa calculada en ventanas de tiempo para decidir si una función es o no estacionaria.
- **Señales mono-canal y multicanal:** una señal mono canal está asociada con una sola fuente, mientras señales multicanal están compuestas de la agregación de múltiples procesos físicos.

Las señales en registros EEG (Electroencefalografía) son aleatorias y multicanal. Es posible encontrar señales EEG periódicas y no periódicas. En la actualidad, la mayoría del procesamiento de señales EEG se lleva a cabo con algoritmos digitales, por lo que

en la mayor parte del tiempo se tiene que trabajar con señales digitalizadas. Previo al proceso de digitalización, los fabricantes de equipo para mediciones EEG suelen convenir en el uso de algunos filtros estandarizados para reducir componentes de poco interés.

2.2.2. Operaciones básicas

Los operadores que se muestran a continuación tienen diversas aplicaciones tanto por sí solos como combinados con diversas transformadas. Todos ellos pueden ser calculados por medio de ventanas (intervalos de tiempo) en lugar de usar la señal completa. La técnica de ventaneo se utiliza primordialmente en los algoritmos digitales, por ejemplo, la convolución con ventaneo es la operación fundamental de cualquier filtro digital.

Convolución

La **convolución** es una operación usada con frecuencia en el tratamiento de señales. Dadas dos señales x y h la convolución está definida como:

$$y(n) = h(n) * x(n) = \sum_{k=0}^{N-1} h(k)x(n-k). \quad (2.1)$$

Si las señales tienen longitud N_h y N_x respectivamente, la señal resultante tendrá longitud $N_y = (N_h + N_x) - 1$.

Energía y Poder

El poder instantáneo de una señal x en el tiempo t está definido como $x^2(x)$. La **energía** de la misma señal está definida por

$$E_x = \sum_{n=0}^{N-1} x^2(n). \quad (2.2)$$

Debido a que el poder está definido como energía por unidad de tiempo, el **poder promedio** de una señal se calcula mediante:

$$P_x = \frac{E_x}{N} = \frac{1}{N} \sum_{k=0}^{N-1} x^2(k). \quad (2.3)$$

En el caso de señales periódicas, es posible sustituir el número de muestras de la señal (N) por el valor de su periodo fundamental (K_0), ya que el poder promedio de un solo periodo es igual al de la señal completa.

Autocorrelación y correlación cruzada

La correlación cruzada es una medida de similitud entre dos señales, que se utiliza frecuentemente para encontrar características de una señal desconocida por medio de otra conocida. Para dos señales x y y , la correlación cruzada R_{xy} está definida como:

$$R_{xy}(k) = \sum_{n=0}^{N-1} x(n)y(n-k). \quad (2.4)$$

La autocorrelación está definida como la correlación aplicada sobre una sola señal:

$$R_{xx}(k) = \sum_{n=0}^{N-1} x(n)x(n-k). \quad (2.5)$$

Raíz media cuadrática

La raíz media cuadrática (RMS por sus siglas en inglés) es una medida estadística del tamaño de la variación entre las muestras de una señal. Se calcula con la fórmula:

$$x_{rms} = \sqrt{\frac{1}{N}(x_1^2 + x_2^2 + \dots + x_N^2)}. \quad (2.6)$$

Si se calcula utilizando ventaneo, es decir dividiendo la señal en intervalos de tiempo y calculando la RMS de cada uno de ellos, la RMS puede ser un sustituto económico (en términos computacionales) para el cálculo de la envolvente de amplitud de una señal.

Entropía

La entropía es una medida de desorganización de un sistema; mientras mayor sea la entropía, mayor desorganización presenta el sistema. Para calcular la entropía de una señal x , primero se debe definir una función de probabilidad $P(x(i))$ que asocie a cada posible valor de x una probabilidad de ocurrencia, entonces la entropía H de x está definida por:

$$H(x) = \sum_i P(x(i)) \log_b P(x(i)) \quad (2.7)$$

donde b regularmente toma alguno de los valores 2, e o 10.

2.2.3. Transformada discreta de Fourier

La transformada discreta de Fourier (DFT por sus siglas en inglés) es una de las herramientas más comunes y poderosas en el campo del tratamiento de señales. Nos permite analizar los componentes de frecuencia (armónicos) en una señal discreta o secuencia de datos.

La DFT está definida por la ecuación [51]:

$$H(\omega) = \sum_{n=0}^{N-1} x(n)[\cos(an) - j\sin(an)] \quad (2.8)$$

donde x es una señal discreta, $a = 2\pi\omega/N$, N es el número de puntos para el cálculo de la DFT, $j = \sqrt{-1}$ y X es la DFT de la señal x . La ecuación más común para describir la DFT se deriva utilizando la relación de Euler $e^{-j\phi} = \cos(\phi) - j\sin(\phi)$; y la ecuación 2.8 queda como:

$$H(\omega) = \sum_{n=0}^{N-1} x(n)e^{-jan}. \quad (2.9)$$

En ocasiones se suele denominar "par de transformada de Fourier" al conjunto compuesto por una señal y su DFT; esto es debido a que, por medio de la transformada inversa de Fourier, es posible reconstruir en su totalidad una señal utilizando su transformada.

La DFT de una señal real es una secuencia de números complejos que representan la magnitud y el ángulo de fase para cada componente de frecuencia de la señal. En el caso de la DFT discreta (ecuación 2.8), las frecuencias presentes en la secuencia resultante dependerán tanto de la frecuencia de muestreo f_s de la señal como del número de puntos N utilizados para calcular la DFT; de este modo, el valor $X(m)$ corresponde al componente de la frecuencia:

$$m = \frac{(\omega f_s)}{N} \quad (2.10)$$

Y los valores de la magnitud y el ángulo de fase para el componente de frecuencia m se obtienen respectivamente con las siguientes formulas:

$$X_{mag}(m) = |X(m)| = \sqrt{(X_{real}^2(m) + X_{imag}^2(m))} \quad (2.11)$$

y

$$X_{\phi}(m) = \angle(X(m)) = \tan\left(\frac{X_{imag}}{X_{real}}\right)^{-1}. \quad (2.12)$$

2.2.4. Espectro de potencia

La transformada de Fourier es utilizada frecuentemente para el cálculo del espectro de potencia de una señal (PSD por sus siglas en inglés). El PSD describe la distribución de poder de la señal sobre distintas frecuencias, por lo que se usa regularmente para encontrar los componentes de frecuencia más significativos de la señal.

Existen muchos algoritmos para estimar el espectro de potencia de una señal digital a través de la DFT. En particular, el algoritmo no paramétrico con ventaneo y traslapes propuesto por Welch (1967) es uno de los más populares debido a que reduce los efectos del ruido en la señal [52]. En la Figura 2.8 se muestra un ejemplo del espectro de potencia calculado con el algoritmo de Welch. La señal original (izquierda), de un segundo de

duración y una frecuencia de muestreo de 500 Hz, contiene una senoide a 10 Hz. Se le ha añadido ruido aleatorio con una distribución normal con media 0 y desviación estándar 1. El espectro de potencia (derecha) muestra los componentes de frecuencia en un rango de $[0,100] Hz$. La unidad de medida de la potencia son los decibeles (dB), que se calculan tomando el logaritmo base 10 de la magnitud de cada componente de frecuencia ($10 * \log_{10}|X(m)|$). Las frecuencias son calculadas utilizando las ecuaciones 2.11 y 2.12. El espectro de potencia muestra un incremento en la magnitud de la frecuencia alrededor de los $10 Hz$.

2.2.5. Transformada rápida de Fourier

El algoritmo para calcular la TDF utilizando la ecuación 2.9 es de orden $O(n^2)$. Si bien un algoritmo de esta complejidad no es malo, la DFT fue originalmente diseñada para ser utilizada en tiempo real, ya que es utilizada en muchas aplicaciones prácticas. En la actualidad el algoritmo más usado para convertir una señal al espacio de frecuencias es el conocido como la transformada rápida de Fourier (FFT por sus siglas en inglés) [53]. La FFT, en contraste con los algoritmos anteriores, es un algoritmo de orden $O(N \log N)$ que además reduce considerablemente el almacenamiento requerido para el cálculo.

2.2.6. Transformada Wavelet

Una onda (*wave*) es usualmente definida como una función oscilatoria en el tiempo o el espacio. El análisis de Fourier hace uso de ondas, expandiendo a la función original en términos de sinusoides. Por otro lado, una *wavelet* (a veces traducido como “onda pequeña”) es una señal oscilatoria de corta duración cuya energía (ver ecuación 2.2) es finita y se encuentra concentrada en un determinado intervalo de tiempo [54]. La diferencia entre éstas dos se muestra en la Figura 2.9. Desde el punto de vista gráfico, las ondas son suaves, predecibles y por lo general se encuentran en un dominio infinito $(-\infty, \infty)$. Por otro lado, las *wavelets* están limitadas en dominio, son irregulares y pueden ser asimétricas.

Para que una función sea utilizada como una *wavelet* $\psi(t)$, ésta debe cumplir con los siguientes requerimientos [55]:

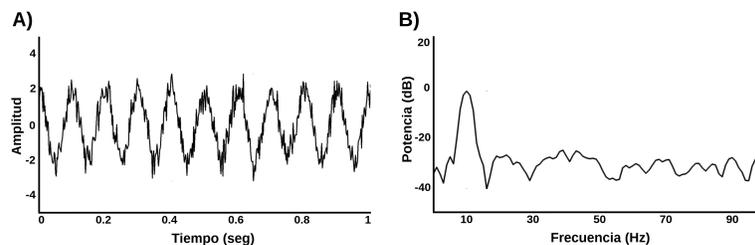


Figura 2.8: Ejemplo del espectro de potencia. A) Una senoide de 10 Hz calculada con frecuencia de muestreo de 500 Hz y duración de 1 segundo. Se ha agregado ruido con una distribución $N(0,1)$. B) El espectro de potencia calculado con el algoritmo de Welch. En la gráfica solo se muestran frecuencias en $[0, 100] Hz$.

- La integral de $\psi(t)$ debe ser cero:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0. \quad (2.13)$$

Esto quiere decir que el valor medio de $\psi(t)$ con respecto al tiempo debe ser cero. O dicho de otra manera la función $\psi(t)$ debe ser oscilatoria.

- La integral de $\psi^2(t)$ debe ser igual a la unidad:

$$\int_{-\infty}^{\infty} \psi^2(t) dt = 1. \quad (2.14)$$

Esta última ecuación corresponde a la energía para la función $\psi(t)$ que se presenta en su versión para señales discretas en la ecuación 2.2.

Utilizando las transformaciones *wavelet* se puede generar una familia de *wavelets* a partir de una wavelet base. Existen dos transformaciones *wavelet*:

1. Traslación o desplazamiento:

$$\psi(t) \rightarrow \psi(t + k). \quad (2.15)$$

2. Dilatación o escalamiento:

$$\psi(t) \rightarrow \psi(mt). \quad (2.16)$$

donde k es el factor de traslación y m es el factor de dilatación. A partir de estas dos operaciones se pueden generar una infinidad de variantes para una *wavelet* madre cualquiera; estas variantes cumplen la siguiente propiedad:

$$\int \psi_j(t) \psi_i(t) dt = \begin{cases} n & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (2.17)$$

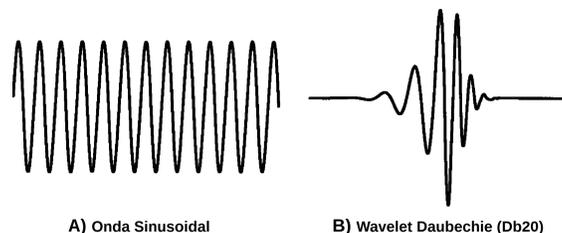


Figura 2.9: Representación de A) una onda sinusoidal de longitud infinita y B) un wavelet de Daubichie (Psi_{D20}). La onda sinusoidal tiene una frecuencia bien definida mientras que el wavelet tiene una pseudo-frecuencia que varía ligeramente con el tiempo.

para dos *wavelets* $\psi_i(t)$ y $\psi_j(t)$ que pertenecen a la familia de la *wavelet* madre $\psi(t)$, y n que es un factor de escalamiento. En otras palabras, dos *wavelets* generadas a partir de la misma *wavelet* madre deben ser ortogonales.

2.2.7. Transformada de Hilbert

La transformada de Hilbert H de una señal x está definida como:

$$H_x(t) = \frac{1}{\pi} p.v. \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \quad (2.18)$$

donde *p.v.* son valores principales de Cauchy, usados debido a que la transformada de Hilbert (HT por sus siglas en inglés) es una integral impropia. Por simplicidad, en algunos lugares la HT se puede encontrar expresada como la siguiente convolución:

$$H_x(t) = \left(\frac{1}{\pi t} p.v. \right) * x(t). \quad (2.19)$$

La señal H_x se encuentra en el mismo dominio que la señal de entrada x . La relación entre estas señales tiene las siguientes propiedades:

- Todos los componentes de frecuencia positiva en H_x son iguales a los componentes de frecuencia positiva en x desplazados $\frac{-\pi}{2}$ (-90°).
- Todos los componentes de frecuencia negativa en H_x son iguales a los componentes de frecuencia negativa en x desplazados $\frac{\pi}{2}$ (90°).
- Tanto el espectro de potencia (PSD) como la auto-correlación de x y de H_x son iguales.

Es posible además crear la señal analítica de x usando la notación:

$$x_a(t) = x(t) + jH_x(t) \quad (2.20)$$

donde $j = \sqrt{-1}$.

Transformada discreta de Hilbert

La transformada discreta de Hilbert (DHT por sus siglas en inglés) puede ser calculada directamente o por medio de filtros digitales. El algoritmo más utilizado para el cálculo directo de la DHT utiliza la siguiente fórmula [56]:

$$h_x(t) = \frac{1}{N} \sum_{v=0}^{N-1} x(v) [1 - (-1)^{v-t}] \cot(v - t) \frac{\pi}{N} \quad (2.21)$$

en donde N es el total de muestras en la señal x .

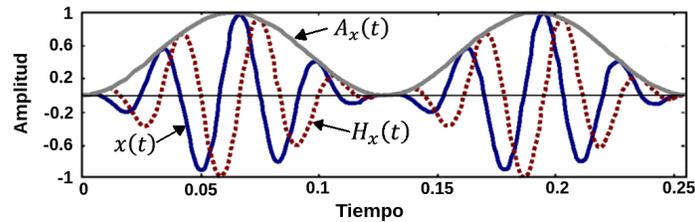


Figura 2.10: Ejemplo del cálculo de la envolvente de amplitud. Se muestra la señal original $x(t)$ en azul con línea continua, la transformada de Hilbert $H_x(t)$ en rojo con línea punteada y la envolvente de amplitud $A_x(t)$ en gris con línea continua.

En el caso del cálculo por medio de filtros digitales, es posible crear un *kernel* h de convolución utilizando la siguiente función [51]:

$$h(n) = \frac{2\text{sen}^2(\pi n/2)}{\pi n}. \quad (2.22)$$

Una vez teniendo el *kernel*, se hace uso de la siguiente propiedad:

$$h_x(n) = h(t) * x(t) \quad (2.23)$$

para encontrar la HDT utilizando la convolución entre el *kernel* h y la señal original. Es posible comprobar que a partir de la propiedad en la ecuación 2.19 se puede derivar la fórmula directa de la ecuación 2.21 [57].

Envolvente de amplitud

La función envolvente es una curva suave que delinea todos los valores de amplitud de una señal. Utilizando la HT de una función es posible derivar su envolvente de amplitud usando:

$$A_x(t) = |x_a(t)| = \sqrt{x^2(t) + H_x^2(t)}. \quad (2.24)$$

La envolvente es útil para llevar a cabo la demodulación de una señal. En la Figura 2.10 se puede ver cómo la envolvente es capaz de encontrar la senoide de baja frecuencia que modula la amplitud de la senoide original.

Frecuencia instantánea

Otra función de gran importancia, que se deriva de la HT, es la frecuencia instantánea. Este valor está definido por el ángulo del vector formado por un número complejo con respecto al eje positivo real. Es posible calcular éste ángulo usando:

$$\phi(t) = \text{arg}A_x(t) \quad (2.25)$$

en donde:

$$argz = \{\theta | z = |z|cos\theta + j|z|sen\theta\}. \quad (2.26)$$

El uso de esta medida se vuelve especialmente importante cuando se analizan señales no-estacionarias, las cuales tienen variaciones de frecuencia (y posiblemente amplitud) durante el tiempo, lo cual hace necesario tener una medida de su frecuencia en un periodo específico del tiempo [58].

Capítulo 3

Diseño del sistema

El diseño del sistema requirió un levantamiento constante de requisitos de los integrantes de Laboratorio de Neurociencia Funcional. En cada uno de estos pasos el diseño sufrió modificación en pos de mejorar la organización de los módulos y las interfaces (línea de comando) para configurar cada ejecución.

En este capítulo se describe la organización final del sistema haciendo énfasis en los algoritmos más importantes. En la primera sección se hace una explicación general de los pasos requeridos en el proceso completo, cada uno de estos pasos constituye en si mismo un módulo que tuvo que ser diseñado para coordinarse con los demás módulos del sistema. La secciones siguientes explican con mayor detalle los algoritmos que se implementaron en cada paso.

3.1. Metodología experimental y requerimientos del sistema

En esta sección se explicará a grandes rasgos el flujo de procesos que deben ser aplicados en los registros EEG para encontrar las relaciones de sincronización entre ondas lentas y husos de sueño. En la figura 3.1 se muestran gráficamente cada uno de los pasos y su interdependencia. Todos estos pasos son los que el sistema implementado automatiza.

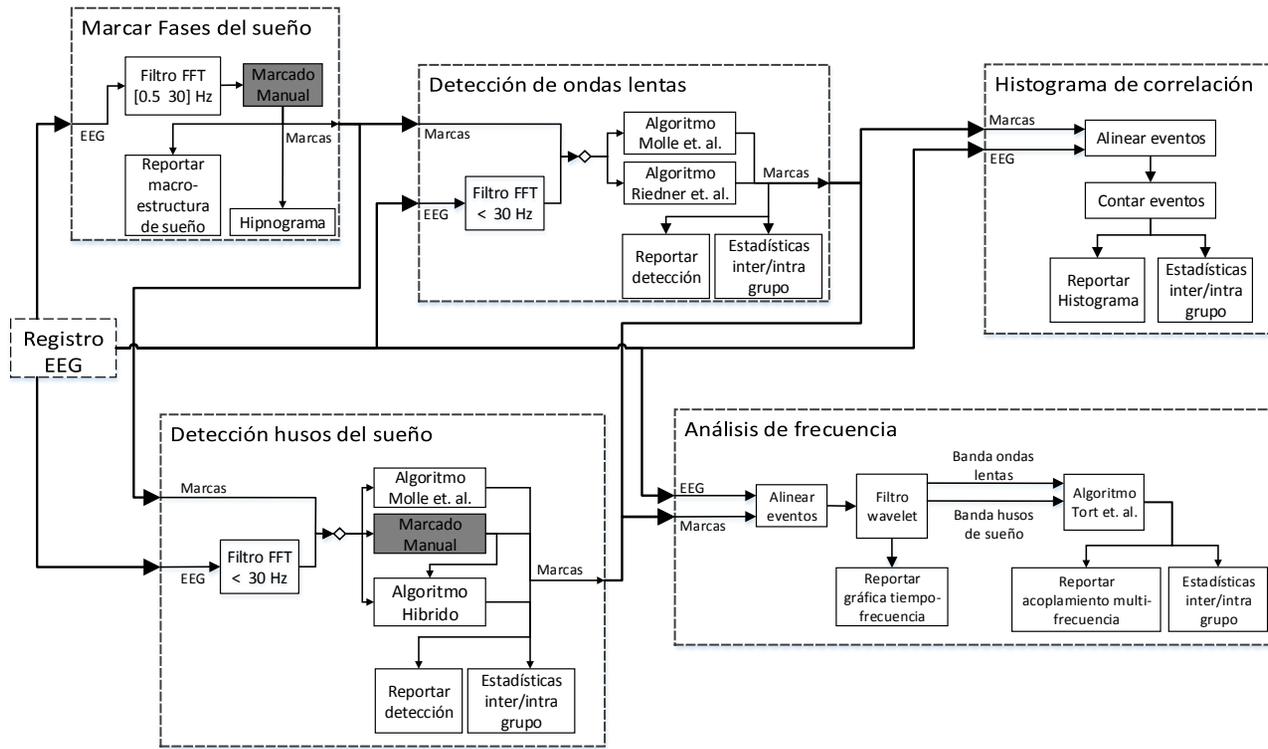


Figura 3.1: El flujo requerido para procesar los registros de sueño puede ser dividido en módulos encargados de operaciones similares. El primer módulo en el flujo de trabajo es el de fases del sueño (esquina superior izquierda) ya que todos los demás módulos dependen de él. Los módulos de detección de patrones (ondas lentas y husos del sueño) tienen las mismas entradas y salidas. Estos dos módulos pueden ser ejecutados simultáneamente. Los módulos de análisis de sincronización también tienen las mismas entradas. El módulo de análisis de frecuencia es el de mayor exigencia, por lo cual es recomendado que su ejecución se realice secuencialmente. Los bloques con fondo oscuro representan tareas que se llevan a cabo manualmente, es decir, sin usar alguna herramienta del sistema.

Registros polisomnográficos

El montaje de polisomnografía (ver sección 2.1.2) incluye la colocación de 59 electrodos de electroencefalografía (EEG) (Fp1, Fp2, AF7, AF3, AFz, AF4, AF8, F7, F5, F3, F1, Fz, F2, F4, F6, F8, FT7, FC5, FC3, FC1, FCz, FC2, FC4, FC6, FT8, T7, C5, C3, C1, Cz, C2, C4, C6, T8, TP7, CP5, CP3, CP1, CPz, CP2, CP4, CP6, TP8, P7, P5, P3, P1, Pz, P2, P4, P6, P8, PO7, PO3, POz, PO4, PO8, O1 y O2), electrooculograma (EOG) de movimientos oculares horizontales y verticales, y electromiograma (EMG) de músculos submentonianos. Estas tres medidas electrofisiológicas son necesarias para clasificar las diferentes fases del sueño, así como los arousals y despertares que ocurren durante el periodo de sueño. Todos los registros electrofisiológicos son realizados con electrodos de superficie de 10 *mm* de diámetro con una aleación de oro (Grass, Inc.) rellenos con crema electrolítica, y adheridos con cinta quirúrgica extra-adhesiva (derivaciones colocadas en la cara) y colodión (derivaciones colocadas en el cuero cabelludo). La colocación de los electrodos EEG se realiza siguiendo las coordenadas del Sistema Internacional 10-20 [37], y las impedancias de cada electrodo se mantienen por debajo de 5k Ω . Las señales neurofisiológicas son amplificadas (amplificadores BrainAmp MR, Brain Vision [®]), filtradas (filtro pasa banda de [0,1 100]Hz), digitalizadas (250 Hz, resolución 16 bits) y almacenadas para su posterior análisis. La muestra una visión general del protocolo de registro.

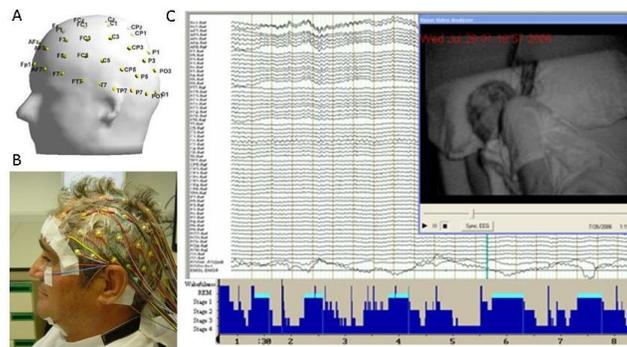


Figura 3.2: **A**, se utilizó un montaje de 59 electrodos (SI 10-20) más electrodos EOG y EMG para poder realizar la clasificación de las fases del sueño. **B**, los registros fueron realizados con electrodos de superficie (10 *mm* de diámetro) con una aleación de oro rellenos con crema electrolítica, y adheridos con cinta quirúrgica extra-adhesiva y colodión. **C**, las señales fueron amplificadas, filtradas (filtro pasa banda [0,1 100]Hz), digitalizadas (250 Hz, resolución 16 bits) y almacenadas para su posterior análisis.

El formato en el cual se almacenan los registros es propietario de la empresa Brain Vision [®]. Este formato incluye un archivo de datos (con extensión *.*eeg*) y un archivo de metadatos (con extensión *.*vhdr*) que contiene información como la tasa de muestreo, los filtros aplicados sobre los datos o los electrodos registrados. Para abrir estos archivos es necesario utilizar el programa Brain Analyzer (Brain Vision [®]). Este programa también es usado para realizar el marcado manual de las fases del sueño y los husos del sueño. Es posible también exportar los registros tanto a archivos de texto como a un formato de datos binarios.

Clasificación de fases de sueño

El primer procesamiento sobre los registros se lleva a cabo para clasificar las fases del sueño en cada sujeto. Para esto se aplica un filtro pasa banda de $[0,5 \ 30]Hz$ sobre las señales de cada electrodo y se clasifican las fases del sueño manualmente sobre épocas de EEG de 20 segundos siguiendo criterios estandarizados [8].

Una vez marcadas las fases del sueño es posible reportar estadísticas descriptivas sobre cada fase en cada sujeto. Algunos de los parámetros que se pueden reportar son la latencia al comienzo del sueño, la latencia al sueño SOL y al sueño REM, la cantidad total de sueño en minutos, y la cantidad de tiempo pasado en vigilia, fase 1, fase 2, y SOL (suma fases 3 y 4) en minutos.

Detección de patrones

Tanto la detección de ondas lentas como la de husos de sueño pueden llevarse a cabo simultáneamente, ninguna de las dos depende de otros datos más que las marcas de las fases de sueño. En ambos casos la detección se lleva a cabo aplicando un algoritmo de umbral en donde se usa la definición electrofisiológica del evento (ver sección 2.1.3) para determinar qué oscilaciones presentan las características adecuadas.

Ambos eventos pueden ser detectados manualmente. Si este es el caso, primero se deben filtrar las señales en las frecuencias en donde está definido el evento. Posteriormente un experto debe de revisar los registros y marcar los eventos según su definición electrofisiológica. Por lo general esta tarea es bastante demandante e incluso en el caso de las ondas lentas son muy pocos los casos en donde se lleva a cabo la detección manualmente.

La salida de estos dos procesos es una lista de marcas que hacen referencia a los eventos detectados. En el caso de las ondas lentas las marcas indican el pico mínimo de la onda, en el caso de los husos de sueño las marcas indican el pico máximo de la señal en el periodo del huso. Estos resultados pueden servir para generar estadísticas descriptivas de la detección o para generar índices estadísticos que comparen un grupo o condición con otro.

Los algoritmos automáticos para detección están descritos en las secciones 3.2 y 3.3.

Relación temporal entre ondas lentas y husos

El siguiente paso es analizar la relación temporal que existe entre los patrones detectados. Una técnica para llevar a cabo este análisis son los histogramas de correlación temporal. Para hacer estos histogramas se debe segmentar la señal en ventanas alineadas a un evento de referencia (ondas lentas en nuestro caso) y contar la cantidad de ocurrencias de el evento analizado (husos de sueño en nuestro caso) dentro de estas ventanas. Una vez creado el conteo se genera un histograma que muestra la distribución temporal de un evento con respecto al otro.

Con esta técnica es posible En la sección 3.4 se describe con detalle los pasos necesarios para generar los histogramas.

Algorithm 1 Detección de ondas lentas

Require: Señal S correspondiente a un canal del registro EEG**Ensure:** Lista de ondas lentas L marcadas como sub-intervalos S Aplicar filtros pasa baja de $30Hz$ y $3,5Hz$ a la señal S Encontrar cruces negativos por cero, Z , en la señal S Colectar los intervalos $I(j) = [Z(j)..Z(j + 1)]$ para $j = 1..(|Z| - 1)$ **for all** $i \in I$ **do****if** $min(S_i) < -40\mu V$ **and** $|min(S_i)| + |max(S_i)| \geq 90\mu V$ **and** $900ms \leq |i| \leq 2000ms$ **then** $L \leftarrow L \cup i$

Primero se aplica un filtro anti-plegamiento (*antialiasing* en inglés) equivalente a $30 Hz$. Posteriormente, se aplica un filtro pasa baja de $3,5 Hz$ para conservar únicamente las frecuencias correspondientes a ondas lentas del sueño. Para detectar las ondas lentas en la señal filtrada primero se encuentran todos los cruces negativos por cero (de positivo a negativo) y se considera, como una potencial onda lenta, a la señal contenida en el intervalo entre cada cruce. Para discriminar entre ondas lentas y otro tipo de oscilaciones, se aplican los criterios que se muestran en la Figura 3.3: 1) el pico mínimo (mínimo valor de la señal) debe ser menor a $-40 \mu V$, 2) la amplitud (medida desde el pico mínimo hasta el pico máximo) de la señal es mayor o igual a $90 \mu V$ y la duración del intervalo comprendido por los dos cruces por cero es menor o igual a $2000 ms$ y mayor o igual a $900 ms$. La referencia de cada evento de onda lenta será la posición del pico mínimo.

3.2.1. Detección por percentiles

Esta modificación del algoritmo tiene como propósito analizar las ondas lentas (y su relación con los husos de sueño) en un espectro de amplitud más amplio que el utilizado por el algoritmo original. Esta modificación fue propuesta por Riedner et. al. 2007 [60].

En general, la detección por percentiles se lleva a cabo de la misma manera que la detección normal. El único paso que cambia es la aplicación del criterio de amplitud para rechazar o aceptar una onda lenta. Es decir, en lugar de aceptar las ondas lentas cuyo pico mínimo sea menor a $-40 \mu V$ y que su amplitud pico a pico sea mayor a $90 \mu V$, se aceptan todas las ondas lentas y se catalogan según el valor de su pico mínimo. Los valores pre-establecidos para catalogar a las ondas lentas son $[0 5]$, $[5 10]$, $[10 20]$, $[20 40]$, $[40 80]$, $[80 160]$ y $[160 300] \mu V$

Los resultados de este algoritmo se presentan igual que la detección normal para cada uno de los percentiles. Es decir, en lugar de generar una imagen y un conjunto de estadísticas para cada sujeto, se generan siete, que corresponden a cada uno de los intervalos usados para catalogar las ondas lentas.

3.2.2. Estadísticas

Para evaluar un cambio significativo entre grupo o condición se utiliza el estadístico t-Student. En el caso de comparar condiciones de los mismos sujetos (i.e. noche antes y después del entrenamiento) se usa el estadístico con muestras relacionadas y por el contrario, para comparar grupos de sujetos distintos (i.e. grupos de control y experimental) se usa el estadístico con muestras no relacionadas.

Los parámetros de interés al comparar las ondas lentas son: densidad (husos por minuto), duración (segundos), frecuencia dominante (Hz), pico mínimo (mV), duración up-state (segundos) y duración down-state (segundos). En el caso de la detección con el método por percentiles se añaden los parámetros: pendiente promedio del up-state, pendiente promedio del down-state, pendiente máxima del up-state y pendiente máxima del down-state.

Para llevar a cabo el estadístico, como el mismo estadístico se aplica sobre diferentes electrodos, se corrigen los falsos positivos que se derivan de las múltiples comparaciones mediante el método de asignación aleatoria basado en el estadístico máximo. Los siguientes pasos se deben llevar a cabo:

1. Se calcula el valor del estadístico t-Student al cuadrado para cada parámetro y para cada electrodo.
2. Para cada electrodo y para cada canal, se intercambia aleatoriamente la posición de los sujetos ya sea por grupo (usando muestras no relacionadas) o por condición (usando muestras relacionadas), se calcula nuevamente el estadístico t-Student al cuadrado y se conserva el mayor.
3. El paso 2 se repite 10000 veces y entonces se conserva el valor que se encuentre en el percentil 95, considerado como el umbral para corregir el error.
4. Todos los resultados del paso uno son comparados con el umbral, solo aquellos que lo sobrepasen son considerados datos significativos.

3.3. Detección de husos del sueño

Existe un gran variedad de algoritmos para la detección de husos en registros EEG humanos [19, 20]. Dados los buenos resultados que arroja, en este trabajo utilizaremos una versión del algoritmo basado en espectros de potencia utilizado en otros trabajos [14]. A continuación, se muestran sus pasos generales, ver Algoritmo 2:

Primero se encuentra el promedio de frecuencias dominantes en todos los canales en el intervalo de frecuencias que corresponde a los husos lentos ($[9\ 12]Hz$) o a los husos rápidos ($[13\ 15]Hz$). Esto se hace calculando el espectro de potencia, por medio del método de Welch, y encontrando el valor máximo dentro del intervalo de frecuencias que se esté analizando. Posteriormente se aplican filtros a la señal para remover ruido y conservar las frecuencias de interés: el primer filtro es de anti-plegamiento equivalente a

Algorithm 2 Detección de husos del sueño

Require: Señal S correspondiente a un canal del registro EEG**Require:** Promedio de frecuencias dominantes \bar{f} sobre todos los canales**Ensure:** Lista de husos L marcados como sub-intervalos de S Aplicar filtro pasa baja de $30Hz$ y pasa banda de $\bar{f} \pm 1,5Hz$ a la señal S Calcular el valor medio cuadrático S_{RMS} de la señal S Calcular el valor del umbral $h = 1,5 * stdev(S_{RMS})$ Encontrar cruces negativos Z^- y positivos Z^+ por el umbral h Colectar los intervalos $I(j) = [Z^+(j)..Z^-(j)]$ para $j = 1..|Z^+|$ **for all** $i \in I$ **do** **if** $500ms <= |i| \geq 3000ms$ **then** $L \leftarrow L \cup i$

$30 Hz$ y el segundo es un filtro pasa banda con un intervalo de frecuencias de $\pm 1,5 Hz$ a partir de la frecuencia dominante media calculada anteriormente. Para llevar a cabo el algoritmo de umbral primero se calcula el valor medio cuadrático (RMS) de la señal filtrada en cada punto y con una ventana de $200 ms$. El valor del umbral se calcula multiplicando la desviación estándar de la señal filtrada por una constante que por lo general toma el valor de $1,5$. A partir de la señal RMS entonces se aplica el criterio de detección de husos: para marcar algún evento como huso, la señal RMS se debe mantener por encima del umbral calculado por un periodo mayor a $500 ms$ y menor a $3000 ms$. El inicio de un evento de huso se marca como el tiempo en el que ocurre el cruce positivo del umbral, mientras el final se marca en el cruce negativo del umbral. La referencia del evento de huso será el pico máximo (posición del valor máximo de la señal RMS) en el intervalo de tiempo comprendido desde el inicio hasta el final del huso. La Figura 3.4 muestra un ejemplo de los criterios de detección de husos del sueño aplicados a la señal RMS y las marcas de inicio y fin del huso de sueño detectado posicionadas sobre la señal filtrada y la señal cruda (sin filtros).

3.3.1. Modificaciones al algoritmo de detección

El algoritmo descrito para detección de husos de sueño puede ser mejorado utilizando tanto información de la detección manual como estrategias adicionales de rechazo de husos. En el primer caso, es necesario contar con marcas colocadas por un experto en los registros. Siempre y cuando se cuente con esas marcas, el algoritmo de detección puede mejorar dramáticamente, sin embargo, debido al tiempo y esfuerzo requeridos en el marcado manual, esta opción puede llegar a ser inviable. Por otro lado, las estrategias de rechazo son fácilmente aplicables después de cualquier algoritmo de detección y están basadas en la propia definición electrofisiológica de los husos de sueño.

En las siguiente secciones se explica con detalle la aplicación de las mejoras al algoritmo de detección de husos.

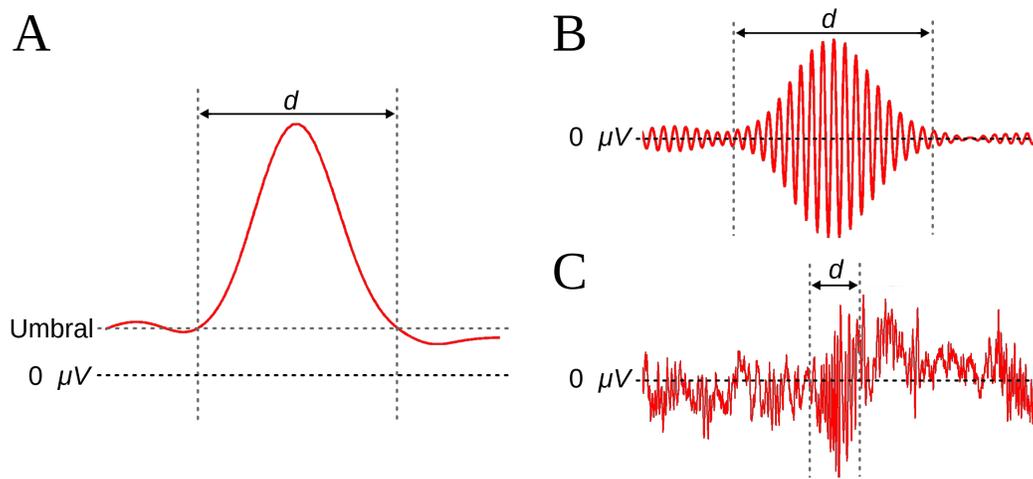


Figura 3.4: Criterio de detección de husos del sueño. En el panel **A** se encuentra el valor medio cuadrático de la señal (RMS) calculado a partir de la señal filtrada. Una vez identificados los cruces positivos y negativos por el umbral, se calcula el tiempo d que la señal RMS se mantiene por encima del umbral; el criterio de aceptación es $500\text{ ms} < d < 3000\text{ ms}$. En los paneles **B** y **C** se muestra el intervalo de tiempo del huso del sueño detectado en la señal RMS, marcado en la señal filtrada y la señal original respectivamente.

Complementos de la detección manual

Las marcas manuales son colocadas por los expertos tanto por canal como para todos los canales. Esto quiere decir que, en el caso de las marcas por canal, el experto estuvo seguro de que el huso ocurrió en un canal en particular. Cuando la marca no tiene canal quiere decir que el huso pudo haber ocurrido (o tener mayor actividad) en cualquier canal del registro.

En el caso de que las marcas hayan sido por canal, es posible incrementar la cantidad de husos detectados analizando los demás canales en el mismo intervalo de tiempo. Para decidir si la actividad en los demás canales es en realidad un huso de deben aplicar los mismos criterios de amplitud y frecuencia aplicados por el algoritmo automático.

Por el contrario, si las marcas manuales no tienen un canal asociado será necesario aplicar esos mismo criterios de huso a la actividad en todos los canales durante el intervalo marcado. Se puede elegir si seleccionar únicamente un canal por intervalo (escogiendo el de mayor amplitud) o si se marcaran todos los canales que cumplan con los criterios de amplitud y frecuencia.

Detección híbrida

Otro enfoque para complementar las detecciones manual y automática es extraer información de la detección manual que después puede aprovechar para calibrar los criterios de la detección automática. De este modo, una estrategia que se puede aplicar es la de calcular el umbral de amplitud utilizando el promedio de amplitud de la actividad marcada por la detección manual.

Por medio de este complemento es posible individualizar los criterios de detección para cada sujeto.

Rechazo de artefactos de electromiograma

En algunas situaciones es útil rechazar los usos detectados que coinciden con movimientos musculares. Para este se usa la señal de los canales EMGL y EMGR (ver sección 2.1.2) que miden movimientos de músculos esqueléticos en el cuello y cabeza.

Dadas la raíz media cuadrática de las señales $RMS_i \forall i \in N$ incluyendo todos los canales del análisis más los dos electrodos del electromiograma, se normaliza cada una por medio de su Z - *score* con:

$$Z_i = \frac{RMS_i - \mu(RMS_i)}{\sigma(RMS_i)}. \quad (3.1)$$

Después se promedian todos los Z - *score* para obtener una señal única normalizada $Z = \mu(Z_i)$. Finalmente el umbral T para el rechazo se calcula con:

$$T = \mu(Z_n) + \sigma(Z_n). \quad (3.2)$$

La señal RMS del intervalo de cada huso marcado por el algoritmo se compara con el umbral T calculado, si la señal excede el umbral entonces es rechazada.

Rechazo por actividad occipital

A veces los husos pueden ser confundidos con oscilaciones características de actividad en otras zonas del cerebro. En este caso, es necesario comparar la actividad marcada como huso con la actividad en electrodos occipitales. Si la actividad occipital tiene mayor amplitud que el huso marcado entonces ese huso se puede rechazar. Para hacer esto se debe calcular la amplitud de la señal de los electrodos O1 y O2 (ver sección 2.1.2) en el mismo intervalo de tiempo donde fue marcado un huso, si la actividad en alguno de estos dos electrodos es mayor a la amplitud del huso entonces se rechaza.

Rechazo de husos vecinos o solapados

Es posible también rechazar los husos que se encuentren muy cerca uno de otro o que se solapen temporalmente de un canal a otro.

En el caso de los husos cercanos, es necesario establecer una separación mínima requerida, si el intervalo entre el final de una marca de huso y el inicio de la siguiente es menor al este valor, entonces se eliminará el huso que tenga menor amplitud.

En el caso de los husos solapados temporalmente, se debe revisar los todos los canales para cotejar que no exista ningún huso en el mismo intervalo de tiempo. Si existen, se debe conservar únicamente aquel con mayor amplitud.

3.3.2. Estadísticas

El cálculo de estadísticos para los parámetros de la detección de husos de sueño es exactamente igual al usado para los parámetros de la detección de ondas lentas (ver sección 3.2.2). En este caso los parámetros seleccionados son: duración (segundos), amplitud (mV), potencia (suma del espectro de potencia), frecuencia dominante (Hz) e intensidad (duración entre amplitud).

3.4. Histograma de correlación

Los histogramas de correlación indican la frecuencia con la cual dos eventos ocurren dentro del mismo intervalo de tiempo. En este caso, estos histogramas ayudan a determinar si dos eventos de onda distintos (husos y ondas lentas) muestran alguna correlación temporal y de fase. Para el cálculo de estos histogramas, la densidad de eventos en cada barra es convertida a unidades de eventos por segundo para facilitar su visualización: mientras mayor sea el valor de una barra, mayor correlación entre las dos señales existe en un tiempo dado.

En general, el procedimiento para generar los histogramas a partir de los dos eventos en los que este trabajo se enfoca es:

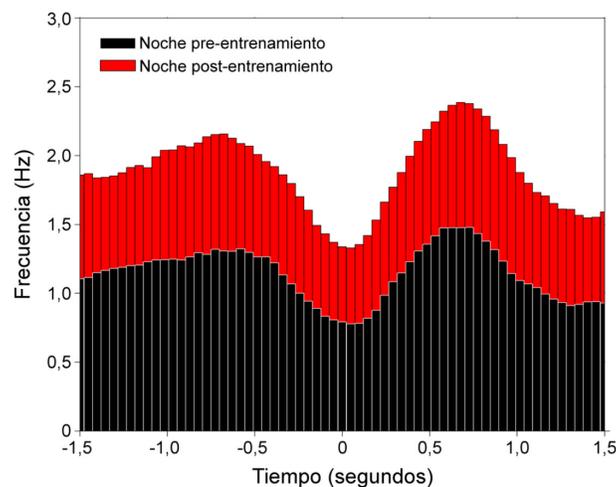


Figura 3.5: Ejemplo de histograma para observar la sincronía temporal entre los husos y las ondas lentas. Se muestran los histogramas para los husos en la noche pre (color oscuro) y post-entrenamiento (color claro).

1. Se hace un conteo de ocurrencia de cada evento en un tiempo dado. En el caso de las de ondas lentas, la referencia del evento es el tiempo en donde ocurre el pico mínimo de la onda. En el caso de los husos se toman como referencias el tiempo en donde ocurrieron todos los picos (negativos y positivos) en la señal.

2. Se crean ventanas de tiempo alrededor del evento de referencia. En este caso, el evento de referencia son las ondas lentas, por lo que se crean intervalos de tiempo de ± 3 segundos alrededor del pico mínimo de la onda.
3. Los intervalos son divididos en sub-intervalos (bin) de tiempo fijo y para cada bin se hace un conteo de los husos que hayan ocurrido en este periodo.
4. Es posible normalizar el conteo en cada bin dividiendo entre el número de eventos de referencia (ondas lentas) y entre el tamaño (en unidades de tiempo) de cada bin.

El método descrito es similar al utilizado por [11]. La Figura 3.5 muestra un ejemplo de este tipo histogramas. En este caso la imagen fue generada para un experimento en el que se miden las diferencias entre la relación de las ondas lentas y husos en dos noches de un protocolo experimental que tiene como objetivo evaluar el efecto del aprendizaje sobre los diferentes parámetros que caracterizan a los husos de sueño, a las ondas lentas y a la sincronía temporal entre estos dos eventos. La escala horizontal en esta figura se mide en ciclos por segundo (Hz) debido a la normalización explicada anteriormente.

Adicionalmente, se aplica el método descrito en [60] para desplegar un gráfico que compare la densidad (normalizada) de eventos entre grupos o condiciones. Para generar esta gráfica es necesario almacenar el valor RMS de la señal de cada electrodo, dividida en los intervalos del paso 2 para generar el histograma ($\pm 3s$ alrededor del pico mínimo de la onda lenta), siempre y cuando un huso haya ocurrido dentro de este intervalo. Esta gráfica entonces despliega el promedio de los intervalos seleccionados para todos los sujetos y electrodos de cada grupo/condición por separado.

3.4.1. Estadísticas

Estadística por bin del histograma

Si bien las estadísticas para los histogramas de correlación utilizan también el estadístico t-Student, en este caso es necesario llevar a cabo algunos pasos para organizar y normalizar los datos. El proceso es el siguiente:

1. Se define un periodo de línea base (los primeros $0,5s$ de la ventana de tiempo del histograma) y se calcula el promedio en de la línea base para cada electrodo.
2. Se calcula el estadístico t-Student al cuadrado para cada bin comparado con el promedio de la línea base. Solo se continua el análisis con aquellos bins para los que el estadístico resulte significativo ($p < 0,05$).
3. Se calcula el valor del estadístico t-Student al cuadrado ahora entre condición o grupo solo para los bins significativos del paso anterior.
4. Para cada electrodo y para cada bin, se intercambia aleatoriamente la posición de los sujetos ya sea por grupo (usando muestras no relacionadas) o por condición (usando muestras relacionadas), se calcula nuevamente el estadístico t-Student al cuadrado y se conserva el valor que se encuentre en el percentil 95.

5. El paso cuatro se repite 10000 veces y entonces se conserva el valor que se encuentre en el percentil 95, considerado como el umbral para corregir el error.
6. Todos los resultados del paso tres son comparados con el umbral, solo aquellos bins que lo sobrepasen son considerados datos significativos.

Esta estadística se aplica para cada canal por separado y para el promedio de los canales.

Estadística para el RMS de la señal

En este caso, como las señales no están separadas en bins, es necesario modificar los pasos 1-3 de las estadísticas por bin para crear primero intervalos que en este caso serán llamados islas. El proceso queda de la siguiente manera:

1. Se define un periodo de línea base (los primeros 0,5s de la ventana de tiempo del histograma) y se calcula el promedio en de la línea base para cada electrodo.
2. Se calcula el estadístico t-Student al cuadrado para cada valor de la señal en el tiempo comparado con el promedio de la línea base.
3. Se calculan los valores para los que el estadístico resulte significativo ($p < 0,05$), y se generan islas de valores significativos consecutivos (más de cinco valores) que cumplen el papel de los bins del histograma.
4. Se calcula el valor del estadístico t-Student al cuadrado ahora entre condición o grupo solo para las islas significativas del paso anterior.
5. Para cada electrodo y para cada isla, se intercambia aleatoriamente la posición de los sujetos ya sea por grupo (usando muestras no relacionadas) o por condición (usando muestras relacionadas), se calcula nuevamente el estadístico t-Student al cuadrado y se conserva el valor que se encuentre en el percentil 95.
6. El paso cuatro se repite 10000 veces y entonces se conserva el valor que se encuentre en el percentil 95, considerado como el umbral para corregir el error.
7. Todos los resultados del paso tres son comparados con el umbral, solo aquellas islas que lo sobrepasen son considerados datos significativos.

Así como en el caso de estadísticas por bin, esta estadística se aplica para cada canal por separado y para el promedio de los canales.

3.5. Métrica de acoplamiento multifrecuencia

La Figura 3.6 muestra las tres formas más comunes de sincronización multifrecuencia en los ritmos del cerebro. Estas formas son: sincronización amplitud-amplitud (PPC por sus siglas en inglés: *Power-Power Coupling*), sincronización fase-frecuencia (PFC por sus siglas en inglés: *Phase-Frequency Coupling*) y sincronización fase-amplitud (PAC por sus siglas en inglés: *Phase-Amplitud Coupling*). El primer tipo muestra una sincronización de fase independiente de la amplitud, haciendo que n ciclos de la frecuencia alta se sincronicen con m ciclos de la frecuencia baja, es por eso que es también llamada sincronización de fase $n : m$ [61]. En el segundo tipo, encontrado con menor frecuencia, la fase de la onda más lenta modula la frecuencia de la onda más rápida sin que la amplitud de ninguna de las dos se vea afectada. En el tercer tipo, conocido también como oscilaciones anidadas, la fase de la frecuencia más lenta influye en la amplitud de la frecuencia más rápida, dando como resultado la sincronización entre la envolvente de amplitud de los ritmos rápidos con la fase de los ritmos lentos.

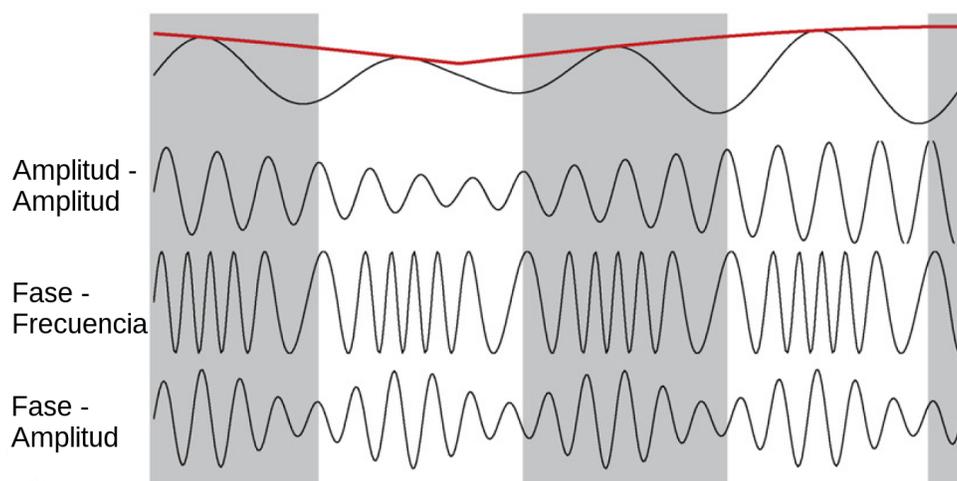


Figura 3.6: Tipos de sincronización multifrecuencia. En la parte superior se presenta una señal con frecuencia que cambia de amplitud durante el tiempo. Las tres señales inferiores describen tipos de sincronización entre de la amplitud o la frecuencia de una señal con frecuencias más rápidas con respecto a los cambios de fase o amplitud de la señal moduladora.

En general, y en este trabajo en particular, la sincronización PAC es la que ha generado mayor interés. Se ha demostrado que tareas de comportamiento pueden generar este tipo de sincronización [62] y es posible que también esté relacionado con integración sensorial y procesos de memoria [63, 12, 64]. Esta modulación ha sido observada en distintas regiones del cerebro incluido el hipocampo, el ganglio basal y la neocorteza; y estas observaciones han sido reportadas en ratas, ratones, ovejas, monos y humanos [65, 23].

El cálculo de la sincronización tipo PAC puede ayudar a encontrar la relación que existe entre actividad en frecuencias altas como pueden ser las entradas sensoriales y motoras, y actividad en frecuencias más bajas relacionadas con cómputos locales [66]. Gracias a esto en los últimos años se han propuesto diversos métodos para calcular la

sincronización PAC, cada uno de los cuales tiene algunas limitaciones y ventajas sobre los otros y pueden ser usados para propósitos distintos [23].

En el presente trabajo se eligió trabajar con el algoritmo propuesto por Tort et al. [16] por ser uno de los más eficientes para trabajar con frecuencias de bandas muy bajas ($< 4Hz$). Este algoritmo calcula un índice de modulación (MI por sus siglas en inglés) que es capaz de medir la modulación de tipo PAC entre dos rangos de frecuencia de interés.

3.5.1. Índice de modulación de Tort

El MI está basado en una medida de entropía normalizada previamente descrita en Hurtado et al. [67]. Para explicar el algoritmo, se denotan los rangos de frecuencia de la amplitud y la frecuencia a analizar como f_A y f_p respectivamente, y la señal original (sin filtros) como $x_{raw}(t)$ (Figura 3.7a). El MI se calcula de la siguiente manera (ver Figura 3.7):

1. Primero, la señal $x_{raw}(t)$ es filtrada en los dos rangos de frecuencias de interés (f_p y f_A). Se denotan como $x_{f_p}(t)$ y $x_{f_A}(t)$ (Figura 3.7b y c).
2. Se calculan las series de tiempo correspondientes a la envolvente de amplitud (transformada de Hilbert) de cada señal filtrada, denotadas por $\phi_{f_p}(t)$ y $A_{f_A}(t)$ (Figura 3.7d y e). Con esto se construye la serie compleja $[\phi_{f_p}(t)A_{f_A}(t)]$ que nos da información acerca de la amplitud de la oscilación f_A en cada fase del ritmo f_p (Figura 3.7f).
3. Las fases de $\phi_{f_p}(t)$ son segmentadas en 18 intervalos de 20° (0° a 360°) y se calcula la media de $A_{f_A}(t)$ para cada intervalo (Figura 3.7g). Se denota $\langle A_{f_A} \rangle_{\phi_{f_p}}(j)$ al valor medio de A_{f_A} en el intervalo j .
4. Entonces se aplica la medida de entropía H definida como:

$$H = - \sum_{j=1}^N p_j \log p_j$$

donde $N = 18$ (el número de intervalos) y p_j se calcula con

$$p_j = \frac{\langle A_{f_A} \rangle_{\phi_{f_p}}(j)}{\sum_{j=1}^N \langle A_{f_A} \rangle_{\phi_{f_p}}(j)}$$

5. El MI se obtiene normalizando H con el máximo valor de entropía posible (H_{max}), que se obtiene para la distribución uniforme $p_j = 1/N$, y por lo tanto $H_{max} = \log N$:

$$MI = \frac{H_{max} - H}{H_{max}}$$

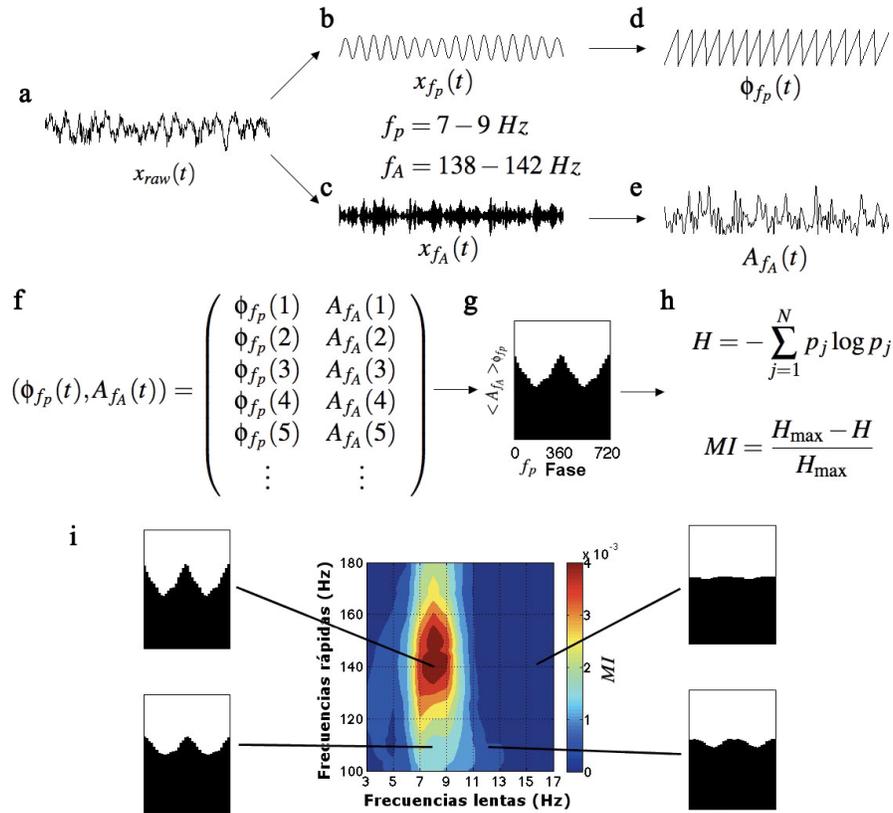


Figura 3.7: Pasos a seguir para calcular el MI y su comodulograma. La señal original (a) se filtra en los rangos de frecuencia de fase (b) y amplitud (c) que se investigan. Por medio de la transformada de Hilbert se calcula las series para la fase (d) y amplitud (e). La composición de las series (f) se usan para calcular la amplitud promedio en cada bin de 20 grados. Finalmente, el MI se calcula aplicando la medida de entropía normalizada (h). El comodulograma (i) se construye graficando los valores MI para cada par de frecuencias de fase y amplitud que se investigan. Imágen extraída de [16]

Al utilizar esta métrica, un valor de $MI = 0$ indica la ausencia de modulación tipo PAC y un valor grande de MI indica una fuerte modulación tipo PAC.

Esta métrica se representa con un gráfico en escalas de pseudocolor (comodulograma) en donde se muestran valores de MI para múltiples pares (f_p, f_A) calculados en diferentes bandas de frecuencia. La figura 3.7i muestra un ejemplo del MI calculado para f_p con una banda de frecuencia de $2Hz$ y tamaño de paso de $1Hz$ y para f_A con una banda de $4Hz$ y tamaño de paso de $2Hz$.

3.5.2. Estadísticas

La primer corrección estadística que se aplica sobre los valores del MI es el cálculo de subrogados. Esta técnica vuelve a calcular el MI para valores de los eventos modificados aleatoriamente. Una vez calculados estos valores subrogados se puede establecer un umbral

de significación para descartar resultados poco consistentes.

Originalmente, el algoritmo de Tort calcula estos subrogados permutando la posición de los eventos, por ejemplo, el primer huso del sueño se intercambia por el quinto huso del sueño. Debido a la naturaleza de la relación entre ondas lentas y husos, se decidió utilizar el método de permutación de fase descrito en Axmacher et al. 2010 [13]. Este método genera números aleatorios con una distribución uniforme dentro del intervalo $[-\phi \phi]$ y los adiciona a la serie de tiempo que corresponde a la fase de la señal de onda lenta. De este modo se está distorsionando la fase de la onda lenta para calcular nuevamente el MI entre ella y los husos.

Para la estadística entre grupos se hace uso tanto de los valores del MI como de sus subrogados. El primer paso es calcular los valores significativos usando los subrogados, después se calcula la estadística por grupo o condición.

Para calcular los valores significativos de los subrogados se aplican los siguientes pasos para cada canal, tiempo, frecuencia lenta y frecuencia rápida:

1. Se ordenan los valores de menor a mayor y se selecciona el que se encuentre en el percentil 95 como umbral de significación.
2. Para todos los sujetos de un grupo o condición, se aplica el estadístico t-Student comparando los valores de MI con los valores de umbral obtenidos en el paso anterior.
3. Se seleccionan solo aquellos en los que el resultado del estadístico haya sido significativo.

Posteriormente, se corrigen los valores significativos utilizando múltiples comparaciones entre grupo o condición. Para esto se llevan a cabo los siguientes pasos para cada canal, tiempo, frecuencia lenta y frecuencia rápida que haya salido significativa en el paso anterior:

1. Se calcula el estadístico t-Student al cuadrado comparando los valores de MI entre los sujetos de un grupo con otro (o condición con otra).
2. Se genera una permutación de sujetos ya sea por grupo (usando muestras no relacionadas) o por condición (usando muestras relacionadas) y se calcula nuevamente el estadístico t-Student al cuadrado.
3. El paso dos se repite 10000 veces y entonces se conserva el valor que se encuentre en el percentil 95, considerado el umbral para corregir el error.
4. Todos los resultados del paso uno son comparados con el umbral, solo aquellos que lo sobrepasen son considerados significativos.

Por cada canal, tiempo, frecuencia lenta y frecuencia rápida se obtendrá un valor que indica si el acoplamiento multifrecuencia fue significativo al comparar grupos o condiciones.

Capítulo 4

Desarrollo del sistema

En el capítulo anterior se delineó el diseño del sistema haciendo énfasis en la estructura modularizada de las herramientas que lo componen. En este capítulo se presentan los detalles de implementación. La primera sección da una mirada general de la organización de los *scripts* y herramientas implementados. Las siguientes secciones explican con mayor detalle cada módulo que compone al sistema y dan un pequeño ejemplo de la configuración y ejecución de cada tarea.

4.1. Estructura del sistema

El conjunto de herramientas que se presenta en este trabajo fue desarrollado en lenguaje Matlab®. Esto consideró muchas ventajas sobre otros lenguajes gracias a la cantidad de operaciones predefinidas que ofrece el ambiente de Matlab®. Dados los requerimientos del sistema se optó por hacer un conjunto de herramientas organizadas por módulos. Cada módulo corresponde a una etapa del procesamiento para la detección y análisis de oscilaciones que se muestra esquematizado en la figura 3.1. Los módulos que componen al sistema son los siguientes:

- *+corr Scripts* para calcular los *textbf*histogramas de correlación
- *+freq Scripts* para calcular el *textbf*acoplamiento multifrecuencia y generar gráficos de frecuencia
- *+SPDetection Scripts* para llevar a cabo la *textbf*detección de husos de sueño
- *+SWDetection Scripts* para llevar a cabo la *textbf*detección de ondas lentas
- *+tools Scripts* que llevan a cabo operaciones requeridas por los demás módulos
- *external* Código de terceros requerido por los demás módulos
- *docs* Manual de usuario y técnico del sistema

Cada uno de estos módulos, excepto el de Herramientas (*+tools*), es independiente de los demás y ofrece un conjunto de operaciones relacionadas con cada paso del análisis. En las siguientes secciones se explicará con detalle las operaciones más importantes que los componen.

4.2. Herramientas básicas

El módulo de herramientas contiene diversos *scripts* utilizados por los demás módulos para realizar tareas repetitivas o que no están directamente relacionadas con la tarea principal del módulo. Este es el módulo de mayor extensión. En este documento se listarán solo algunas de las herramientas que lo componen.

4.2.1. Lectura y almacenamiento temporal

Los registros EEG son archivos que contienen las señales de cada uno de los electrodos registrados junto con metadatos para describir los datos. El formato original de éstos archivos no se conoce ya que pertenece a la empresa (Brain Vision ®) que desarrolla el hardware y software que permite registrar y almacenar las señales eléctricas. Para poder leer estos archivos hemos elegido usar la herramienta BioSig ¹ de código libre. Esta herramienta ofrece diversos paquetes para lectura y procesamiento de señales biomédicas.

Originalmente las señales de todos los electrodos están empaquetadas en el mismo archivo con extensión **.eeg* y los metadatos se encuentran en un archivo con extensión **.vhdr*. Adicionalmente, los archivos **.vmrk* contienen las marcas que se hayan incluido en los registros como pueden ser las fases o husos de sueño marcados manualmente. BioSig es capaz de leer los dos primeros archivos, los datos y metadatos, pero no el archivo de marcas. Para leer las marcas se ha diseñado una función que se puede acceder con el comando *tools.io.loadBAMarks*.

Una vez que los archivos de datos y metadatos son leídos con BioSig, el sistema almacena la señal de cada electrodo por separado en un archivo binario. Posteriormente, cuando el usuario requiera la señal de algún electrodo, es posible leer únicamente el archivo que le corresponde, lo cual mejora considerablemente el tiempo de lectura de los datos. El proceso de separación de los archivos originales y lectura de los archivos de cada canal se oculta al usuario por medio de una clase que representa al registro EEG de un sujeto. Esta clase esta definida en el archivo *tools.eeg.EEG* y sus funciones principales son:

- ***EEG(config : tools.eeg.struct.config)*** Es el constructor de la clase. Recibe una estructura de configuración en la que el usuario deberá introducir al menos los parámetros *dataPath* y *marksPath* que indican las rutas donde se encuentran los archivos de registros y de marcas respectivamente. La función se encargara de leer el archivo de metadatos y asegurarse que todos los archivos necesarios para la consecuente lectura de datos de un electrodo estén listos.
- ***getChannel(channel : [string—int])*** Esta función lee los datos de un canal (electrodo). Si el archivo para ese canal no existe entonces realizará el proceso de separación del archivo **.eeg* original. El canal se puede especificar como cadena de texto o como un número entero. La cadena debe contener el nombre completo del canal e.g. *CP2* o *Pz*. El número indica la posición del canal en la lista original de electrodos.

¹<http://biosig.sourceforge.net/>

4.2.2. Filtros digitales

Todas las funciones relacionadas con filtros digitales se encuentran en el sub-módulo ***tools.filt***. Para ocultar al usuario los detalles de implementación de los filtros se creó la clase ***tools.filt.FilteredEEG***. Por medio de esta es posible configurar qué tipo de filtros se desea aplicar sobre los registros EEG creando un objeto que en realidad contiene a un objeto de tipo ***tools.eeg.EEG***. Las funciones más importantes del objeto ***FilteredEEG*** son las siguientes:

- ***FilteredEEG(eeg: tool.eeg.EEG, config : tools.structs.filterConfig)*** Es el constructor de la clase. Recibe un objeto de tipo ***tool.eeg.EEG*** que contiene la información de los registros que serán filtrados. Adicionalmente, se puede incluir una estructura de configuración que consta de una lista de filtros definidos por medio de la estructura ***tools.structs.filterConfig***.
- ***addFilter(config : tools.structs.filterConfig)*** Por medio de esta función se puede agregar filtros para ser aplicados en los registros. Cada filtro que se agregue será encolado en la lista de filtros definida por el objeto, de este modo el último filtro agregado será el último filtro aplicado en cada señal.
- ***getChannel(channel : [string—int])*** Esta función llama a su vez a la función homónima de la clase EEG contenida. De esta forma, se obtiene la señal original y es posible filtrarla para devolverla al usuario. Al aplicar los filtros se hará un ciclo sobre todas las definiciones de ciclos que contenga la clase aplicando cada uno secuencialmente. Esto quiere decir que la señal devuelta al usuario después de la llamada a esta función habrá sido filtrada tantas veces como filtros se hayan definido para este objeto.

La estructura ***tools.structs.filterConfig*** es usada para definir nuevos filtros que pueden ser agregados en un objeto ***tools.filt.FilteredEEG***. Esta estructura consta de los siguiente campos:

- ***filter*** Nombre del filtro. Los valores pueden ser *'FIR'* (filtro de impulso finito usando transformada de Fourier), *'WAV'* (filtro usando transformada *Wavelet*), *'RMS'* (filtro de raíz media cuadrática con ventaneo) y *'AVG'* (filtro de promedio con ventaneo).
- ***type*** Tipo de filtro. Esta variable solo se aplica a los filtros de tipo *'FIR'* y puede tomar los valores *'LOW'*, *'HIGH'* y *'BAND'*.
- ***freqs*** Frecuencia del filtro. En el caso de filtros pasa banda (*'BAND'*) y *wavelet* (*'WAV'*) entonces esta variable contiene dos valores numéricos. En el caso de filtros pasa baja (*'LOW'*) y pasa alta (*'HIGH'*) esta variable contiene un solo valor numérico.
- ***fstep*** Tamaño de paso para filtros *wavelet* (*'WAV'*).
- ***winSize*** Tamaño de ventana para filtros de raíz media cuadrática (*'RMS'*) o de promedio (*'AVG'*)

Para facilitar la creación de estas estructuras se puede utilizar la función ***tools.structs.createFilter(varargin)*** que únicamente asigna los parámetros que reciba a los campos de una estructura vacía de tipo ***tools.structs.filterConfig***. Por ejemplo, si

se deseara crear un filtro de promedio con ventaneo con ventana de 200 puntos, se debería llamar a esta función de la siguiente manera: `filter = tools.filt.createFilter('filter', 'RMS', 'winSize', 200)`.

Para calcular los filtros se utilizaron funciones tanto propias como de otros paquetes de software. En el caso de los filtros de promedio y de raíz media cuadrática se programaron funciones propias que están definidas en los archivos `tools.filt.filt_AVG` y `tools.filt.filt_RMS` respectivamente. La función para los filtros *wavlet* hace uso de la transformada y transformada inversa continuas definidas en el paquete *Continuous Wavelet Transform*² de Matlab®. En el caso de los filtros con transformada de Fourier se utilizó una versión modificada de la función `eegFilt()` del software *EEGLab*³ [29].

4.2.3. Lectura y procesamiento de marcas

Los archivos de marcas (`*.vmrk`) contienen etiquetas asociadas con un instante y un canal del registro. Estas marcas son utilizadas para indicar la ocurrencia de algún evento como puede ser el inicio o fin de un huso de sueño, el inicio de una fase de sueño o los periodos en donde el paciente estuvo despierto. Los archivos están compuestos de texto y cada marca es un renglón que especifica entre otras cosas el nombre del evento, el tiempo de ocurrencia y el canal. Para leer estos archivos se creó el *script* `tools.io.loadBAMarks` que procesa cada línea y almacena los eventos en una estructura de datos. La clase `tool.EEG` ejecuta automáticamente este *script* en su constructor, de este modo no es necesario ejecutar implícitamente la lectura de los archivos de marcas cuando se está configurando alguno de los experimentos del sistema.

Una vez cargadas las marcas es posible hacer distintas operaciones con ellas. El paquete `tools.marks` contiene todos los *scripts* relacionados con estas operaciones. Una de las tareas más comunes es la de extraer los instantes de tiempo que corresponden con alguna fase de sueño. Para esta operación se usa el *script* `tools.marks.filterMarks` que recibe los siguientes parámetros:

- **fullMarks** Lista completa de marcas cargadas del archivo `*.vmrk`.
- **stage** Identificador de la fase de sueño que se desea filtrar (ver tabla 4.1).
- **time** Inicio y fin de tiempo de interés de las marcas. Si inicio y fin son ambos cero entonces se considera todo el registro.
- **sampFreq** Frecuencia de muestreo del registro.
- **cycles** Lista de ciclos que se tomarán en cuenta para filtrar las marcas. Si el parámetro es cero entonces se consideran todos los ciclos.
- **merge** Bandera para indicar si las marcas con final cercano al siguiente inicio deben ser consideradas como una sola marca. Los valores son `true` o `false`.

²<http://www.mathworks.es/es/help/wavelet/gs/continuous-wavelet-transform.html>

³<http://scn.ucsd.edu/eeglab/>

Cuadro 4.1: Lista de indicadores para referenciar cada fase de sueño.

Identificador	Descripción
1	SOL (Fases 3 y 4)
2	Fase 2
3	Fase 3
4	Fase 4
5	Toda la noche (Fases 1, 2, 3 y 4)
6	Arousals
7	Husos
8	Fases 2, 3 y 4
9	Vigilia
10	Fase 1
11	REM
12	REM fásico
13	Todo el registro
14	Sueño (Fases 1, 2, 3, 4 y REM)

Otra herramienta de gran importancia por ser usada por muchos otros módulos del sistema es la de convertir estructuras de marcas en índices de marcas. La estructura de datos donde se almacenan las marcas contiene únicamente los instantes de tiempo de inicio y fin de un evento, haciendo imposible usarlas directamente para referenciar índices en los registros. Para convertir las marcas en índices es necesario crear un vector que contenga todas las posiciones que una marca representa, por ejemplo, si la marca inicia en el tiempo t_1 y termina en el tiempo t_2 , los índices de esa marca son todos los valores comprendidos en el intervalo $[t_1 t_2]$. Los *scripts* ***tools.marks.flatten*** y ***tools.marks.flattenWC*** se encargan de realizar esta operación para todos los eventos almacenados en una estructura de marcas; el primero solo toma en cuenta los instantes de tiempo (inicio y fin) mientras que el segundo guarda también el valor del canal donde ocurre el evento en el vector de índices. Para llevar a cabo la operación inversa, es decir, convertir un vector de índices en una estructura de marcas se utilizan los *scripts* ***tools.marks.unflatten*** y ***tools.marks.unflattenWC***.

4.2.4. Hipnograma y estructura del sueño

El módulo ***tools.sleep*** contiene *scripts* que se calculan las estadísticas descriptivas para las fases del sueño y los husos de sueño, y generan hipnogramas que representan gráficamente la macro-estructura del sueño.

Para calcular las estadísticas de la macro-estructura del sueño se ejecuta el comando ***tools.sleep.sleepStats***. Este *script* generará un archivo *Excel* con 131 datos. Los parámetros reportados pueden ser catalogados en:

- Tiempos totales (en minutos) de registro, sueño, latencia de sueño y latencia de sueño de ondas lentas.
- Porcentaje de permanencia en cada fase de sueño.

- Porcentaje de permanencia en cada fase de sueño durante la primera y segunda mitad de la noche.
- Tiempos totales (en minutos) de cada fase de sueño.
- Medidas relacionadas con los arousals (despertares).

Para calcular las estadísticas de los husos de sueño se ejecuta el comando *tools.sleep.spindleStats*. Este *script* generará un archivo *Excel* que reporta la cantidad, duración promedio y densidad (husos por minuto) detectados en un sujeto.

Finalmente, el *script* *tools.sleep.hypnogram* genera una hipnograma que representa la macro-estructura del sueño del sujeto, la Figura 4.1 muestra un ejemplo. Con éste tipo de gráfica se pueden observar los lapsos de tiempo en los que el sujeto permaneció en cada una de las fases del sueño durante las horas de registro.

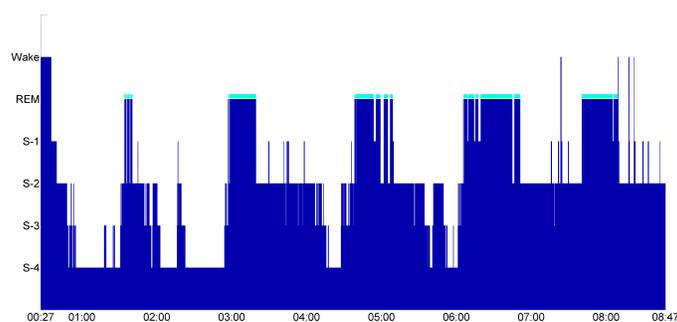


Figura 4.1: El eje vertical indica la fase de sueño en la que el sujeto se encuentra mientras que el eje horizontal indica el tiempo del registro en horas. Por lo general la fase de movimientos oculares Rápidos (REM por sus siglas en inglés) suele aparecer después de un ciclo completo, es decir, después de que el sujeto a pasado desde fase 1 hasta fase 4. Sin embargo, en los hipnogramas la fase REM se suele posicionar después de la vigilia y se suele agregar un bloque de color contrastante para enfatizar la permanencia del sujeto en esta fase.

4.3. Detección de ondas lentas

En el módulo *SWDetection* se encuentran todos los *scripts* necesarios para ejecutar y reportar la detección de ondas lentas. Dentro de la carpeta del módulo se encuentran los siguientes archivos y sub-módulos:

- *+struct Scripts* que contienen la definición de las estructuras de datos usadas en este módulo.
- *+descriptive Scripts* para calcular las estadísticas descriptivas de la detección.
- *+report Scripts* para generar reportes de la detección.
- *+stats Scripts* para calcular las estadísticas por grupo o por condición.
- *main* Configura y ejecuta la detección
- *crossSubjAvg* Configura y ejecuta los promedios de grupos o condiciones de las estadísticas descriptivas de la detección.
- *alg_SWDetection* Ejecuta los algoritmos de detección de ondas lentas.

La ejecución completa de la detección de ondas lentas se lleva a cabo en tres pasos: detección, promedio por grupo o condición y cálculo de estadísticas. Las siguientes tres secciones explican cada uno de estos pasos.

4.3.1. Detección

La detección de ondas lentas se configura y ejecuta a través del *script* ***SWDetection.main***. Este *script* es el punto de entrada para ejecutar tanto el algoritmo descrito por Mölle et al. [11] (que llamaremos detección normal) como el descrito por Riedner et al. [60] (que llamaremos detección por percentiles).

Es posible configurar una gran cantidad de parámetros relacionados con la detección. En el *script* ***SWDetection.struct.config*** se encuentran definidos todos estos parámetros. En el mismo *script* ***SWDetection.main*** se pueden configurar los parámetros más importantes de la detección, en el siguiente fragmento de código se muestra una parte de este *script*:

```
% SWDetection.main

% Variable to configure subject names.
subject.condition = {'Sleep1_' 'Sleep2_'};
subject.group = {'C'};
subject.number = {[2,3,5:12,14:19,21]};

eeg.dataPath = 'E:\Registros'; % Data path should contain *.vhdr and *.dat files
eeg.marksPath = 'E:\Registros'; % Marks path should contain *.vmrk files

% SW detection 3.5 Hz low-pass filter
filt(2) = tools.filt.createFilter('filter', 'FIR', 'type', 'LOW', 'freqs', 3.5);

% Analysis output path. The output path will be created inside this one.
cf.outPath = 'C:\Analysis\SWS';

% Algorithm parameters
cf.stage = 1; % Sleep stage targeted in experimente as in:
            % 1 -> SWS (STAGE 3 + STAGE 4)
            % 2 -> STAGE 2
            % 3 -> STAGE 3
            % 4 -> STAGE 4
            % 5 -> All night
            % For a full list see tools.mark.getStageString

cf.byPercentile = false; % false -> normal detection
                    % true -> percentile detection
```

La primera estructura es ***subject*** que contiene tres campos: Una lista de las condiciones experimentales (***condition***), una lista de los grupos experimentales (***group***) y tantas listas de números de sujetos como grupos experimentales existan (***number***). Posteriormente se debe indicar la ruta de acceso a los registros y los archivos de marcas en las

variables *eeg.dataPath* y *eeg.marksPath* respectivamente.

Las siguientes variables están relacionadas con la detección. Primero se crea un filtro pasa baja de $3,5Hz$ que será aplicado a todas las señales. Después, en la estructura *cf* de tipo *SWDetection.struct.config* se inicializa la ruta de salida (*outPath*), la fase de sueño que se desea analizar (*stage*) y una bandera que indica si se usará el algoritmos normal o por percentiles.

Los demás parámetros de la detección pueden ser inicializados en este *script* y se pueden consultar en el archivo *SWDetection.struct.config*.

Una vez ejecutada la detección, en la ruta de salida se almacenarán dos archivos por cada sujeto: Un archivo con extensión *.mat* que contiene los resultados de la detección en formato de Matlab® y una imagen *.tiff* que muestra el promedio de las ondas lentas detectadas en este sujeto. Adicionalmente, se almacenarán dos archivos por cada ejecución: Un archivo con extensión *.xls* que contiene las estadísticas descriptivas descritas en la sección 3.2.2 y un archivo de texto con la lista de los parámetros de configuración de la detección.

4.3.2. Promedio de sujetos

El siguiente paso en la detección es calcular el promedio de la detección por sujeto. Para esto se usa el *script* *SWDetection.crossSubjectAvg*. En la configuración se debe indicar, entre otras cosas, la ruta donde se encuentran los reportes de la detección, la ruta donde se almacenarán los promedios y el tipo de configuración experimental usada (por grupo o por condición). A continuación se muestra la parte del *script* donde se configuran estos parámetros:

```
% SWDetection.crossSubjectAvg

% Input and output paths
cf.path = 'C:\Analysis\SWS';
cf.outPath = 'C:\Analysis\SWS\Avg_crosssubj';

% Type of comparision: 'group' or 'condition'
cf.compare = 'group';
```

La salida de esta ejecución incluye un archivo de estadísticas (*.xls*), un reporte binario (*.mat*) y tres figuras de promedios (una para cada lóbulo cerebral) para cada uno de los grupos/condiciones.

Adicionalmente, se incluye una figura que despliega los promedios de la onda lenta para cada grupo/condición solapados. En la figura 4.2 se muestra un ejemplo de este tipo de gráficas.

4.3.3. Estadísticas

Las estadísticas entre grupo o condición se ejecutan por medio del *script* *SWDetection.stats.main*. Así como en el *script* para el promedio de las detecciones, se deben

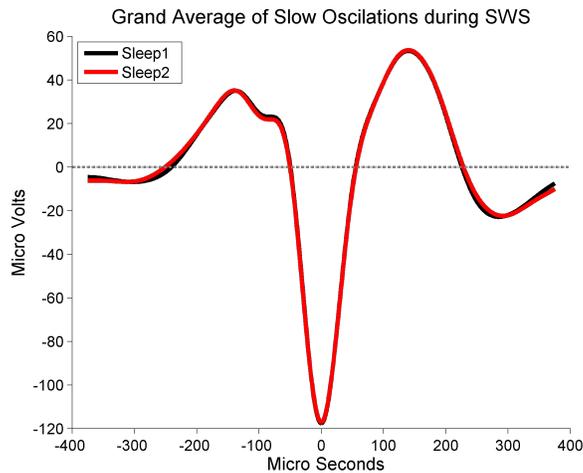


Figura 4.2: El promedio de ondas lentas para cada grupo o condición se hace con una ventana de tiempo de 4 *seg* alrededor del pico mínimo de las ondas detectadas. En la imagen se solapan los promedios para cada grupo o condición.

configurar las rutas de entrada y salida de la ejecución así como indicar si la comparación se realizará por grupo o por condición. Se debe también indicar que tipo de detección se realizó (normal o por percentiles). Finalmente, se pueden configurar tres parámetros relacionados con las pruebas estadísticas (ver sección 3.2.2): el umbral para considerar un resultado significativo, el número de comparaciones para el método de múltiples permutaciones y el porcentaje mínimo de sujetos que deben ser intercambiados en cada permutación. A continuación se muestra un extracto de código del *script SWDetection.stats.main*:

```
% SWDetection.stats.main

% Input and output paths
cf.path = 'C:\Analysis\SWS';
cf.outPath = 'C:\Analysis\SWS\stats_group';

cf.compare = 'condition'; % Type of comparision: 'group' or 'condition'

cf.detType = 'percentile'; % Type of detection to analyze: 'normal' or 'percentile'

% Statistics parameters
cf.sgnf = 0.05; % Significance threshold
cf.nPerm = 1000; % Number of permutations for multiple comparissons
cf.permPerc = 0.3; % Minimum number of permuted subjects. [0 1]
```

El resultado de esta ejecución es un archivo con extensión **.xls* que contiene los resultados del estadístico t-Student.

4.4. Detección de husos del sueño

En el módulo *SPDetection* se encuentran todos los *scripts* necesarios para ejecutar y reportar la detección de husos de sueño. Dentro de la carpeta del módulo se encuentran los siguientes archivos y sub-módulos:

- *+struct Scripts* que contienen la definición de las estructuras de datos usadas en este módulo.
- *+descriptive Scripts* para calcular las estadísticas descriptivas de la detección.
- *+report Scripts* para generar reportes de la detección.
- *+reject Scripts* para complementar la detección por medio de criterios de rechazo de husos detectados.
- *+stats Scripts* para calcular las estadísticas por grupo o por condición.
- *main* Configura y ejecuta la detección
- *crossSubjAvg* Configura y ejecuta los promedios de grupos o condiciones de las estadísticas descriptivas de la detección.
- *alg_Molle* Ejecuta los algoritmos de detección definido en Mölle et al. 2009[11].
- *alg_SPHybrid* Ejecuta el algoritmo de detección híbrido descrito en la sección 3.3.1.
- *alg_SPLoad* Carga marcas de husos puestas manualmente en un solo canal del registro.
- *alg_SPLoadAll* Carga marcas de husos puestas manualmente sin información del canal del registro.

Al igual que la detección de ondas lentas, la ejecución completa de la detección de husos se lleva a cabo en tres pasos: detección, promedio por grupo o condición y cálculo de estadísticas. Las siguientes tres secciones explican cada uno de estos pasos.

Los algoritmos implementados así como las técnicas de rechazo de husos están explicados en la sección 3.3.

4.4.1. Detección

La detección de husos de sueño se configura y ejecuta a través del *script SPDetection.main*. Este *script* es el punto de entrada para ejecutar todos los algoritmos de detección implementados.

Es posible configurar una gran cantidad de parámetros relacionados con la detección. En el *script SPDetection.struct.config* se encuentran definidos todos estos parámetros. En el mismo *script SPDetection.main* se pueden configurar los parámetros más importantes de la detección, en el siguiente fragmento de código se muestra una parte de este *script*:

```
% SPDetection.main

% Anti-aliasing low-pass filter.
filt(1) = tools.filt.createFilter('filter', 'FIR', 'type', 'LOW', 'freqs', 30);

% Algorithm parameters
cf.outPath = 'C:\Analysis\SP_molle'; % Analysis output path.
```

```
cf.band = 1;      % 0 - slow -- 1 - fast -- 2 - both
cf.stage = 4;    % Sleep stage targeted in experiment as in:
                % 1 -> SWS (STAGE 3 + STAGE 4)
                % 2 -> STAGE 2
                % 3 -> STAGE 3
                % 4 -> STAGE 4
                % 5 -> All night

cf.alg = 'molle'; % ['molle'|'load'|'loadall'|'hybrid']

% Parameters for load, loadall and hybrid algorithm. Comment when using molle
% algorithm.
% cf.minWaveTime = 0.1;

% Parameters for load algorithm. Comment when using other algorithms
% cf.semiautomatic = true;
% cf.keepOriginal = true;

% Parameters for loadall algorithm. Comment when using other algoritms
% cf.singleChann = false;
```

Se han omitido los parámetros relacionados con los nombres de sujetos y las rutas de los registros.

Primero se crea un filtro pasa baja de 30Hz que será aplicado a todas las señales. Después, en la estructura *cf* de tipo *SPDetection.struct.config* se inicializa la ruta de salida (*outPath*), la banda de frecuencia deseada (*band*), la fase de sueño que se desea analizar (*stage*) y el tipo de algoritmo que se desea ejecutar (*alg*). Dependiendo del algoritmo seleccionado es necesario configurar algunos parámetros como se describe a continuación:

- Para el algoritmo *molle* no es necesario incluir ningún otro parámetro, sin embargo es necesario comentar la definición de los parámetros necesarios para los otros algoritmos así como se muestra en el extracto de código anterior.
- Si se selecciona cualquier algoritmo excepto *molle* es necesario quitar el comentario del parámetro *minWaveTime* para asignarle un valor de 0,1.
- Para el algoritmo *load* es necesario quitar el comentario y asignar valores a los parámetros *semiautomatic* y *keepOriginal*. El primero, *semiautomatic*, se pone como *true* si se desea buscar husos en los intervalos de tiempo donde hay marcas manuales. El segundo, *keepOriginal*, se pone como *true* si se desea conservar los husos marcados manualmente o *false* en caso contrario.
- El algoritmo *loadall* requiere un parámetro extra llamado *singleChann*. Al poner este parámetro como *true* se indica al algoritmo que busque canales en los intervalos de tiempo donde hay marcas manuales pero que solo seleccione un huso por canal.

Los demás parámetros de la detección pueden ser inicializados en este *script* y se pueden consultar en el archivo *SPDetection.struct.config*.

Una vez ejecutada la detección, en la ruta de salida se almacenarán dos archivos por cada sujeto: Un archivo con extensión **.mat* que contiene los resultados de la detección en formato de Matlab® y una imagen **.tiff* que muestra el promedio de los husos de sueño en este sujeto. Adicionalmente, se almacenarán dos archivos por cada ejecución: Un archivo con extensión **.xls* que contiene las estadísticas descriptivas descritas en la sección 3.3.2 y un archivo de texto con la lista de los parámetros de configuración de la detección.

4.4.2. Promedio de sujetos

El siguiente paso en la detección es calcular el promedio de la detección por sujeto. Para esto se usa el *script* ***SPDetection.crossSubjectAvg***. En la configuración se debe indicar, entre otras cosas, la ruta donde se encuentran los reportes de la detección, la ruta donde se almacenarán los promedios y el tipo de configuración experimental usada (por grupo o por condición). A continuación se muestra la parte del *script* donde se configuran estos parámetros:

```
% SWDetection.crossSubjectAvg

% Input and output paths
cf.path = 'C:\Analysis\SP_molle';
cf.outPath = 'C:\Analysis\SP_molle\Avg_crosssubj';

% Type of comparison: 'group' or 'condition'
cf.compare = 'condition';
```

La salida de esta ejecución incluye un archivo de estadísticas (**.xls*), un reporte binario (**.mat*) y tres figuras de promedios (una para cada lóbulo cerebral) para cada uno de los grupos/condiciones.

4.4.3. Estadísticas

Las estadísticas entre grupo o condición se ejecutan por medio del *script* ***SPDetection.stats.main***. Así como en el *script* para el promedio de las detecciones, se deben configurar las rutas de entrada y salida de la ejecución así como indicar si la comparación se realizará por grupo o por condición. Adicionalmente, se pueden configurar tres parámetros relacionados con las pruebas estadísticas (ver sección 3.3.2): el umbral para considerar un resultado significativo, el número de comparaciones para el método de múltiples permutaciones y el porcentaje mínimo de sujetos que deben ser intercambiados en cada permutación. A continuación se muestra un extracto de código del *script* ***SWDetection.stats.main***:

```
% SWDetection.stats.main

% Input and output paths
cf.path = 'C:\Analysis\SP_molle';
```

```
cf.outPath = 'C:\Analysis\SP_molle\stats_group';

cf.compare = 'condition'; % Type of comparision: 'group' or 'condition'

% Statistics parameters
cf.sgnf = 0.05; % Significance threshold
cf.nPerm = 1000; % Number of permutations for multiple comparissons
cf.permPerc = 0.3; % Minimum number of permuted subjects. [0 1]
```

El resultado de esta ejecución es un archivo con extensión **.xls* que contiene los resultados del estadístico t-Student.

4.5. Histogramas de correlación

Los *scripts* encargados del calculo de los histogramas de correlación se encuentran en el módulo *histogram*. Dentro de la carpeta del módulo se encuentran los siguientes archivos y sub-módulos:

- *+report Scripts* para generar reportes de la detección.
- *+stats Scripts* para calcular las estadísticas por grupo o por condición.
- *main* Configura y ejecuta la detección
- *crossSubjAvg* Configura y ejecuta los promedios de grupos o condiciones de los histogramas.
- *grandAvgFig* Genera una figura para comparar los promedios entre grupos o condiciones.
- *alg_Histogram* Ejecuta el cálculo del histograma de correlación definido en Mölle et al. 2009[11].

En este caso, la ejecución completa del cálculo del histograma se lleva a cabo en cuatro pasos: cálculo de histogramas, promedio por grupo o condición, figura por grupo o por condición y cálculo de estadísticas. Las siguientes tres secciones explican cada uno de estos pasos.

4.5.1. Cálculo del histograma

Los histogramas se configuran y ejecutan utilizando el *script histogram.main*. El algoritmo para calcular el histograma (descrito en la sección 3.4) es el presentado por Mölle et al. [11].

El histograma se calcula el solapamiento temporal entre dos eventos, por lo que para ejecutarlo se deben indicar primero las rutas en donde se encuentran los resultados de la detección de estos eventos. Posteriormente se debe indicar la ruta en donde se almacenarán los resultados de la ejecución y, finalmente, se puede indicar el tamaño (en segundos) de la ventana de tiempo del histograma y el número de barras que lo componen. A continuación se muestra un extracto de código del *script histogram.main* donde se pueden configurar estos parámetros:

```
% histogram.main

% Input and output paths
cf.refPath = 'C:\Analysis\SWS';
cf.spPath = 'C:\Analysis\SP_molle';
cf.outPath = 'C:\Analysis\Hist_SWS-SP_molle';

cf.winSize = 3;    % Histogram window size in seconds around reference event
cf.nBins = 125;   % Number of histogram bins
```

Al final de la ejecución, en la ruta de salida se almacenarán dos archivos por cada sujeto: Un archivo con extensión **.mat* que contiene los resultados de la correlación en formato de Matlab® y una imagen **.tiff* que muestra el histograma de correlación para los dos eventos seleccionados en cada sujeto.

4.5.2. Promedio de sujetos

El siguiente paso en el cálculo del histograma es calcular el promedio por sujeto. Para esto se usa el *script histogram.crossSubjectAvg*. En la configuración se debe indicar, entre otras cosas, la ruta donde se encuentran los reportes del cálculo del histograma, la ruta donde se almacenarán los promedios y el tipo de configuración experimental usada (por grupo o por condición). A continuación se muestra la parte del *script* donde se configuran estos parámetros:

```
% histograma.crossSubjectAvg

% Input and output paths
cf.path = 'C:\Analysis\Hist_SWS-SP_molle';
cf.outPath = 'C:\Analysis\Hist_SWS_perc-S\Avg_crosssubj';

cf.compare = 'group'; % Type of comparision: 'group' or 'condition'
```

Por cada grupo o condición este *script* genera tres archivos: Un archivo con extensión **.mat*, una imagen **.tiff* y un archivo con extensión **.xls* que contienen el promedio de todos los sujetos.

4.5.3. Figura por grupo o condición

En el siguiente paso es posible calcular una figura que solapa el promedio del histograma para cada sujeto por grupo o por condición. El *script* que crea esta figura es *histogram.grandAvgFig*. Además de los parámetros solicitados en el *script* para calcular los promedios, se pueden indicar parámetros relacionados con la creación de la figura. El siguiente extracto de código muestra una configuración típica de estos parámetros:

```
% histograma.grandAvgFig

% Plot variables
```

```

cf.plotOrder = [1 2]; % Group/condition order in the plot
cf.plotColor = {'k', 'r'}; % Color for each group/condition
cf.plotYlim = [0 40]; % Y scale
cf.plotXlim = [-1.5 1.5]; % X scale

```

El primero, *plotOrder*, sirve para indicar cual grupo se debe desplegar los grupos o condiciones. El orden original se indica asignando el valor [1 2], mientras el orden inverso con [2 1]. De igual modo, se pueden configurar los colores para cada grupo o condición en la variable *plotColor*. Finalmente, las variables *plotYlim* y *plotXlim* configuran las escalas de los ejes *y* y *x* respectivamente.

Al ejecutar este *script* se generará un archivo con un gráfico similar al que se presenta en la figura 4.3.

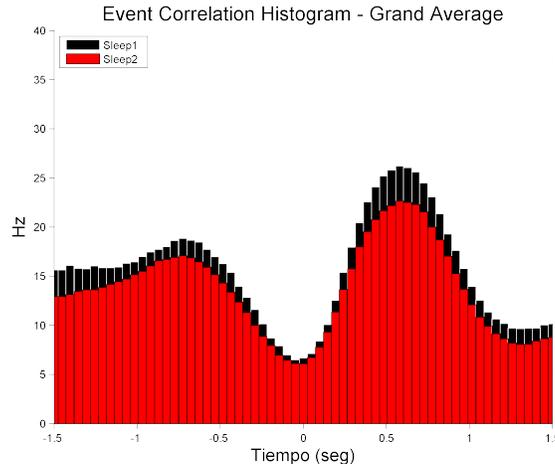


Figura 4.3: La figura muestra el promedio de los valores del histograma para cada grupo o condición. El eje del tiempo se muestra en una ventana de 1,5 s alrededor del pico de las ondas lentas (evento de referencia). El valor de cada barra del histograma está normalizado de tal modo que muestre la tasa de ocurrencia de husos por segundo.

4.5.4. Estadísticas

Para calcular las estadísticas de la comparación por grupo o condición de los resultados del histograma se tienen dos opciones. Es posible calcular las estadísticas utilizando tanto el conteo de los eventos como se muestra en el histograma o el valor RMS de la señal cuando los eventos se solapan así como se describe en la sección 3.4. En ambos casos la configuración de los *scripts* es la misma, sin embargo los archivos para ejecutarlas cambian. En el sub-módulo *histogram.stats* se encuentran los archivos para ejecutar estas estadísticas y se describen a continuación:

- El archivo *histogram.stats* calcula las estadísticas con los valores del histograma para los casos en los que la detección de ondas lentas fue normal (no por percentiles). Al final de la ejecución se generará una imagen por cada electrodo que compara los

valores para cada grupo o condición y un archivo con extensión **.xls* con los resultados del estadístico t-Student (ver sección 3.4.1). El archivo *histogram.stats_perc* ejecuta las mismas estadísticas para los casos en los que la detección de ondas lentas fue por percentiles. La figura 4.4 muestra un ejemplo de los gráficos arrojados por estos *scripts*.

- El archivo *histogram.stats_rms* calcula las estadísticas del valor RMS promedio de las señales EEG en los casos en los que la detección de ondas lenta fue normal. Al igual que en el caso anterior, la ejecución genera una imagen por cada electrodo que compara los valores para cada grupo o condición y un archivo con extensión **.xls* con los resultados del estadístico t-Student (ver sección 3.4.1). El archivo *histogram.stats_rms_perc* ejecuta las mismas estadísticas para los casos en los que la detección de ondas lentas fue por percentiles. La figura 4.4 muestra un ejemplo de los gráficos arrojados por estos *scripts*.

En ambos casos, las figuras muestran los bins (o islas) que resultaron significativos en distintos puntos del análisis estadístico:

- Los bins significativos después de compararlos con la línea base se muestran con una línea horizontal negra.
- Los bins significativos después de hacer el estadístico t-Student comparando grupo o condición se muestran con un asterisco negro.
- Los bins significativos después de hacer la prueba de múltiples comparaciones se muestran con un asterisco rojo.

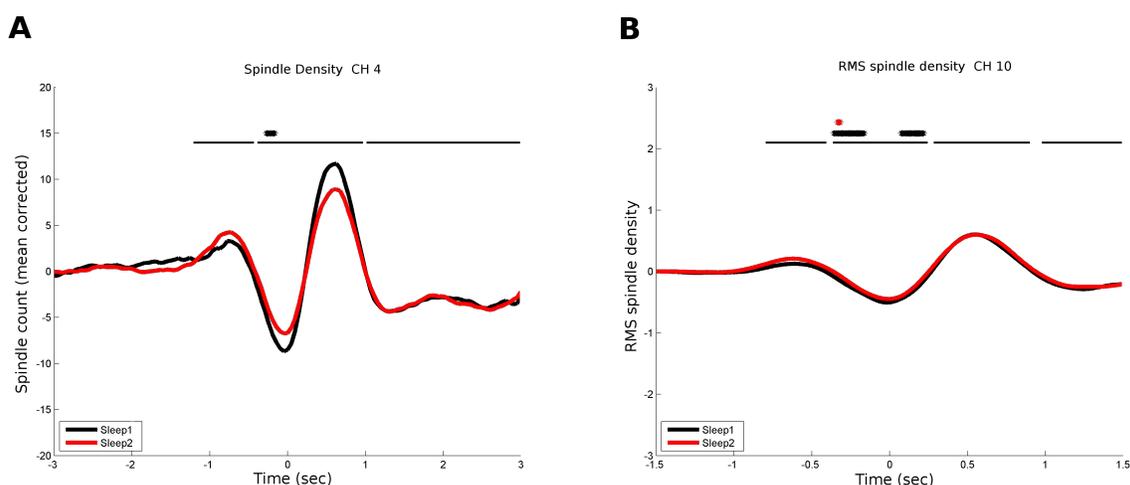


Figura 4.4: Gráfico de estadísticas para la relación temporal entre ondas lentas y husos usando conteo de eventos **A** o valore RMS de la señal **B**. En ambos casos la figura muestra una comparación entre grupo o condición. Las líneas negras en la parte superior de la figura indican que esos bins (o instantes de tiempo en el caso de RMS) son significativos con respecto a la línea base. Para los bins significativos en la comparación entre grupo o condición, se muestra un asterisco negro, y para los que fueron significativos después de la corrección por múltiples comparaciones se muestra un asterisco rojo.

4.6. Análisis de frecuencia

El módulo *+frequency* contiene *scripts* relacionados con el análisis de frecuencia de las señales. El sub-módulos *+tfr* tiene *scripts* que se encargan tanto de calcular espectros de frecuencia como de generar las representaciones gráficas tiempo-frecuencia (TFR por sus siglas en inglés). Por otro lado, el sub-módulo *+crossfreq* aloja los *scripts* para calcular el acoplamiento multifrecuencia y sus estadísticas.

Dentro de la carpeta del módulo se encuentran los siguientes archivos y sub-módulos:

- *+crossfreq* Contiene *Scripts* para el calculo del acoplamiento multifrecuencia.
- *+tfr* Contiene *Scripts* para la generación de las gráficas tiempo-frecuencia.
- *collectAlignedSignals* Recolecta las señales alineadas a un evento.
- *waveletFilter* Aplica un filtro *wavelet* a una lista de señales.

Antes de poder ejecutar los cálculos en los sub-módulos es necesario recolectar las señales deseadas y aplicarles algún filtro según la necesidad del análisis. Para esto se utilizan los *Script collectAlignedSignal* y *waveletFilter* respectivamente.

La recolección se lleva a cabo alineando las ventanas de tiempo al pico mínimo de las ondas lentas detectadas. Para esto es necesario primero seleccionar un canal como fuente generadora de cada onda lenta en un sujeto. Esto se debe hacer debido a que muchas ondas lentas detectadas se solapan temporalmente entre canales y, por definición (ver sección 2.1.3), todas ellas pertenecen a un solo evento electrofisiológico. Al ejecutar el *Scripts collectAlignedSignal*, este elimina los solapamientos seleccionando el canal con las ondas lentas de mayor amplitud. Posteriormente, para cada onda lenta que se solape con un huso, el *Scripts* almacenara una señal de onda lenta (correspondiente al canal con mayor amplitud) y tantas señales de husos como alineados a la onda lenta como canales haya en el registro. De este modo, esta ejecución genera dos archivos por cada sujeto, uno con la lista de ondas lentas generadoras solapadas con un huso y otro con la lista de husos alineados a la onda lenta con la que coinciden. A continuación se muestra un extracto de código correspondiente a la configuración de la ejecución del *Script collectAlignedSignal*:

```
% frequency.collectAlignedSignal

% Input and output paths
cf.swPath = 'C:\Analysis\SWS';
cf.spPath = 'C:\Analysis\SP_load';
cf.outPath = 'C:\Analysis\Frequency\raw_load';

cf.timeWindow = 4000; % ms to each side of negative peak
```

Los primeros parámetros indican las rutas de entrada y salida de la ejecución, mientras el parámetro *timeWindow* indica el tamaño de la ventana de tiempo de interés que será tomado para cada lado de la posición del pico mínimo de las ondas lentas.

Para filtrar las señales recolectadas en el paso anterior se usa el *Script waveletFilter*. En este caso, los parámetros que se deben configurar están relacionados con el filtro *wavelet* que se aplicará. A continuación se muestra un extracto del código donde se pueden

configurar:

```
% frequency.wavletFilter

% Filter parameters
cf.foi = [0.2 1.4]; % Frequency band of interest, in Hz
cf.fstep = 0.2;    % Frequency resolution, in Hz
```

El primero, *foi*, es la frecuencia de interés. Todo filtro *wavelet* debe ser configurado como pasa banda, por lo que este parámetro debe contener dos valores correspondientes a los cortes de frecuencia superior e inferior del filtro. El segundo parámetro indica la resolución del filtro *wavelet* que se aplicará.

Una vez ejecutados estos dos *Scripts* es posible ejecutar cualquiera de las operaciones en los sub-módulos de esta carpeta, que se explican en las siguientes secciones.

4.6.1. Gráficas de tiempo-frecuencia

La carpeta *+frequency.tfr* contiene dos *scripts* para el cálculo de los gráficos de tiempo-frecuencia (TFR por sus siglas en inglés). El primer, *calcSpectrum*, calcula los espectros de potencia utilizando filtros *wavelet* para un rango de frecuencias definido por el usuario. El segundo, *plotSpectrum*, genera los gráficos TFR que se ejemplifican en la figura 4.5. El usuario debe ejecutar ambos scripts de manera secuencial para obtener los resultados deseados.

Para calcular el espectro de frecuencias se deben configurar los siguientes parámetros en el *script calcSpectrum*:

```
% frequency.tfr.calcSpectrum

% Input and output paths
cf.path = 'C:\Analysis\Frequency\raw_load';
cf.outPath = 'C:\Analysis\Frequency\load_spec_0.2-16';

% Spectrum parameters
cf.foi = [0.2 16.2]; % Frequency band of interest, in Hz
cf.fstep = 1; % Frequency resolution, in Hz
cf.toi = [-1000 1000]; % Time window of interest, in ms
cf.base = [-1500 -1000]; % Window of baseline, in ms
```

Los primeros dos parámetros indican la ruta de entrada y salida de la ejecución. La ruta de entrada debe contener la lista de husos alineados a la onda lenta después de la ejecución del *script collectAlignedSignal* explicado al inicio de esta sección. Los siguientes parámetros sirven para configurar el espectro de potencia que se calculará. *foi* y *fstep* configuran el rango de frecuencias de interés y el tamaño de paso que se utilizará para calcular el filtro *wavelet*. El parámetro *toi* indica el intervalo de tiempo de interés en mili-segundos mientras *base* configura el intervalo de tiempo que será considerado como línea base. La línea base será restada al intervalo de tiempo de interés para normalizar los resultados del espectro de potencia.

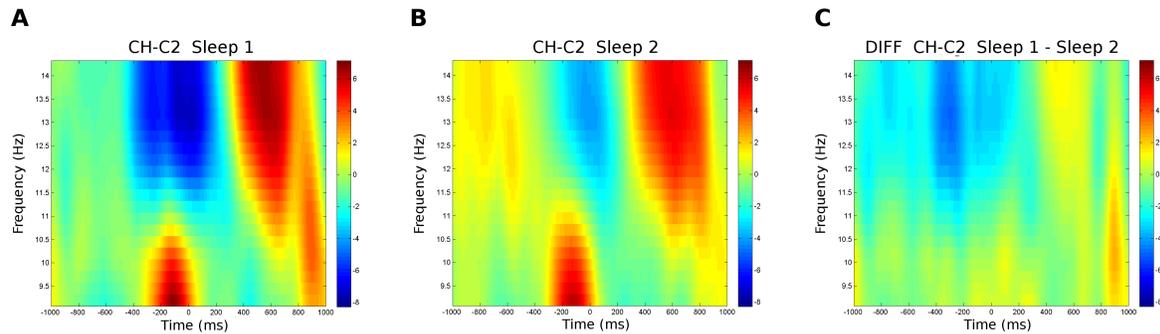


Figura 4.5: Las gráficas de tiempo-frecuencia tienen en el eje horizontal el intervalo de tiempo de interés y en el eje vertical el rango de frecuencias de interés. El color en la gráfica indica la potencia (en unidades arbitrarias) de una frecuencia en un tiempo dado según la escala que se muestra a la derecha. En **A** se encuentra un ejemplo de gráfica TFR para el promedio de sujetos de la condición *Sleep1* en el canal *CH*. En **B** lo mismo para los sujetos de la condición *Sleep2*, y en **C** la diferencia entre ambos promedios.

Al terminar esta ejecución se almacenará en la carpeta de salida un archivo **.mat* por cada sujeto que contendrá el reporte de la ejecución. Este reporte incluye el espectro de frecuencias para el tiempo de interés, el espectro corregido con la línea base, el rango de frecuencias de interés y el intervalo de tiempo de interés.

Después del cálculo del espectro se pueden generar las gráficas TFR. Para hacer esto se ejecuta el *script plotSpectrum*. En este caso, se debe introducir como ruta de entrada la ruta de salida del *script* anterior y se debe indicar si la comparación se realizará por grupo o por condición.

```
% frequency.tfr.plotSpectrum

% Input and output paths
cf.path = 'C:\Analysis\Frequency\load_spec_0.2-16';
cf.outPath = 'C:\Analysis\Frequency\load_spec_0.2-16\plot';
cf.compare = 'condition';

% Plot parameters
cf.foi = [9 15]; % Frequency band of interest, in Hz
cf.toi = [-1000 1000]; % Time window of interest, in ms
```

Adicionalmente, se puede introducir nuevamente un rango de frecuencias de interés y un intervalo de tiempo de interés. Estos valores serán utilizados únicamente para calcular las escalas de la gráfica TFR. Como se puede ver en la figura 4.5, la ejecución de este *script* genera tres imágenes por cada canal del registro. Las primeras dos representan el promedio de los valores del espectro de frecuencias para los sujetos del primero y segundo grupo o condición. La tercera es la resta de estos promedios en cada grupo o condición.

4.6.2. Acoplamiento multifrecuencia

El cálculo del acoplamiento multifrecuencia entre ondas lentas y husos se hace por medio del *script* `frequency.crossfreq.main`. En este caso la configuración ofrece diversos parámetros que pueden ser calibrados para obtener mejores resultados.

```
% frequency.crossfreq.main

% Input and output paths
cf.basicpath = 'C:\Analysis\Old\Frequency';
cf.srcPath = 'load_filt_0.2-1.4';
cf.spPath = 'load_filt_12.8-13.2';
cf.outPath = 'cross_freq';

% Algorithm parameters
cf.toi = [-1000 -500 0 500]; % Time window of interest, in ms
cf.base = [-1500 1000]; % Time window of baseline, in ms
cf.time = [-4000 4000]; % Time window of original data, in ms
cf.fLow = [0.4,0.8,1.2]; % List of low frequencies of interest, in Hz
cf.fHigh = [13,14,15]; % List of high frequencies of interest, in Hz
cf.fLow_pad = 0.4; % Size of low frequencies pad, in Hz
cf.fHigh_pad = 1; % Size of high frequencies pad, in Hz
cf.numSurr = 4; % Number of surrogates
cf.pValue = 0.05; % Significance threshold
```

El primer grupo de parámetros está relacionado con las rutas de entrada y salida de la ejecución. Se debe indicar la ruta tanto de las señales filtradas en la banda de ondas lentas (*srcPath*) como de las señales filtradas en la banda de husos (*spPath*). Posteriormente se pueden configurar los parámetros usados por el algoritmo de acoplamiento multifrecuencia. El tiempo del análisis se debe especificar en tres variables distintas: *toi* contiene los intervalos de tiempo para el análisis, *base* es el intervalo de tiempo de línea base y *time* es el intervalo completo de tiempo de los datos filtrados. Las frecuencias del análisis también se especifican usando diversas variables: *fLow* y *fLow_pad* indican los valores de frecuencias bajas y el tamaño de la ventana alrededor que se usará alrededor de las frecuencias, mientras *fHigh* y *fHigh_pad* representa los mismos datos pero para las frecuencias altas. Finalmente, para el cálculo de los subrogados se necesita especificar el número de permutaciones (*numSurr*) que se desea aplicar y el valor del umbral para considerar a un dato significativo (*pValue*).

Al término de la ejecución se generará un variable de tipo **.mat* por cada sujeto que contiene los valores del *MI* por cada canal, tiempo, frecuencia baja y frecuencia alta, así como los valores de los subrogados, que se explican en la sección 3.5.2, para cada *MI*.

Estadísticas

Después del cálculo de los subrogados, se lleva a cabo la estadística para grupos o condiciones como se explica en la sección 3.5.2. Para esto se ejecuta el *script* `frequency.crossfreq.stats` que generará tres gráficas similares a las de la figura 4.6 por cada canal analizado. Para

ejecutar éste *script* se deben configurar prácticamente los mismos parámetros que para calcular el acoplamiento multifrecuencia:

```
% frequency.crossfreq.stats

% Input and output paths
cf.path = 'C:\Analysis\Frequency\cross_freq';
cf.outPath = 'C:\Analysis\Frequency\cross_freq\stats_group';
cf.compare = 'group';

% Algorithm parameters
cf.toi = [-1000 -500 0 500]; % Time window of interest, in ms
cf.base = [-1500 1000]; % Time window of baseline, in ms
cf.time = [-4000 4000]; % Time window of original data, in ms
cf.fLow = [0.4,0.8,1.2]; % List of low frequencies of interest, in Hz
cf.fHigh = [13,14,15]; % List of high frequencies of interest, in Hz
cf.numSurr = 40; % Number of surrogates
cf.sgnf = 0.05; % Significance threshold
cf.nPerm = 1000; % Number of permutations for multiple comparissons
cf.permPerc = 0.3; % Minimum number of permuted subjects. [0 1]
```

Los únicos parámetros que se incluyen en este *script* que no estaban en el anterior son los relacionados con la prueba estadística, el número de permutaciones para las múltiples comparaciones (*nPerm*) y el mínimo porcentaje de sujetos permutados en cada comparación (*permPerc*).

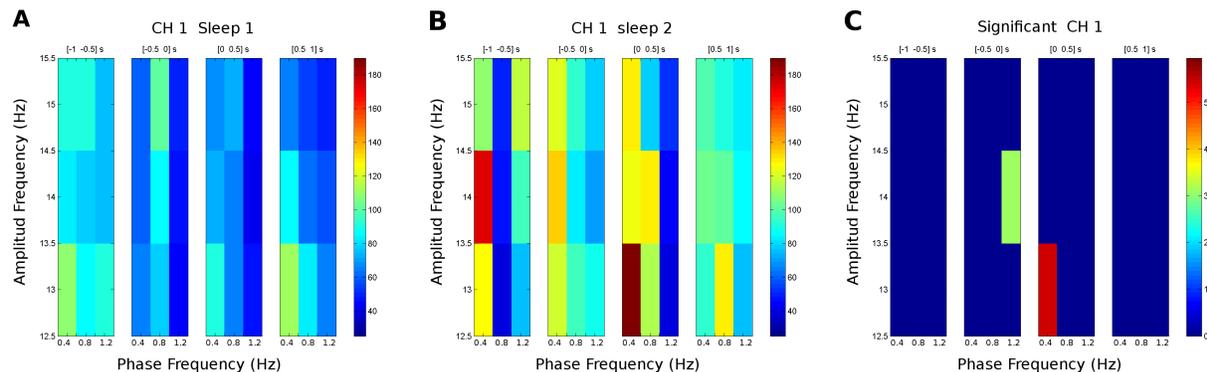


Figura 4.6: Las gráficas generadas tienen 4 dimensiones: rango de frecuencias bajas (eje X), rango de frecuencias altas (eje Y), intervalos de tiempo (bloques horizontales) y valor del dato (escala de color del gráfico). Las primeras dos gráficas muestran los valores que fueron significativos al ser comparados con los subrogados para la condición Sleep 1 (A) y Sleep 2 (B). El gráfico C muestra los valores significativos después de la comparación entre condiciones.

Las gráficas generadas contienen los resultados de la comparación entre grupo o condición para cada frecuencia baja (eje X), frecuencia rápida (eje Y) e intervalo de tiempo (bloque horizontal). Las primeras dos gráficas en la figura 4.6 (A y B) corresponden a los valores que sobrepasaron el umbral de significación al ser comparados con los subrogados. La tercera gráfica (C) muestra los valores significativos después de hacer la comparación por grupo o condición.

Capítulo 5

Pruebas

El sistema que se desarrollo en esta tesis se valido con una serie de experimentos que se presentan en la tesis "Papel de las ondas lentas en los procesos de consolidación de la memoria que operan durante el sueño" de la maestría en neurociencias de la Universidad Pablo de Olavide, llevada a cabo en el Laboratorio de Neurociencia Funcional.

A continuación se presenta un resumen del planteamiento, objetivo, hipótesis, metodología y resultados de dicho proyecto de tesis. Se hace énfasis en las partes del proyecto que hicieron uso de *scripts* que pertenecen al sistema desarrollado.

Planteamiento, objetivos e hipótesis

Como se explicó en la introducción de este documento, resultados procedentes de diferentes líneas de investigación sugieren que el sueño juega un papel importante en los procesos de consolidación de la memoria. Aunque se desconocen los mecanismos responsables, algunos de los eventos electrofisiológicos que se postulan como posibles correlatos neurales son los husos de sueño y a las ondas lentas. Evidencias recientes sugieren que el aprendizaje asociativo produce un aumento en la densidad de estos dos eventos y sobre todo en su sincronía temporal, y que dicho incremento está relacionado con la mejora en el recuerdo. Sin embargo, otras investigaciones sugieren que el papel del sueño sería preservar las memorias más fuertes en detrimento de las más débiles. Esta última hipótesis es la que se intentó comprobar en el trabajo.

El objetivo de este estudio fue determinar cuál o cuáles de los parámetros que definen a los husos y ondas lentas se ven afectados por el aprendizaje. Adicionalmente, el segundo objetivo del trabajo fue identificar cuáles de estos parámetros correlacionan con la memoria en función de que la información a recuperar se haya codificado en un contexto semánticamente congruente (i.e., memoria más fuerte) o incongruente (i.e, memoria más débil).

Metodología

Para llevar a cabo el experimento, durante tres días consecutivos los participantes fueron sometidos a una sesión de entrenamiento y a dos sesiones de evaluación de la memoria. La

fase de entrenamiento consistió en mostrar a los sujetos pares de caras famosas que podían o no compartir la misma profesión (memoria congruente e incongruente respectivamente). En la fase de pruebas los sujetos hicieron un reconocimiento de los pares de caras, lo cual arrojó índices de retención de memoria que después fueron comparados con los distintos parámetros de husos y ondas lentas.

Para analizar los efectos del aprendizaje se registró la actividad eléctrica cerebral durante el sueño a 16 pacientes durante dos noches consecutivas, antes y después de la fase de entrenamiento. Posteriormente, se compararon los parámetros que definen a las ondas lentas y los husos extraídos en cada noche.

El montaje de polisomnografía para los registros fue el mismo que se describió en la sección 3.1.

La clasificación de las fases del sueño se realizó manualmente sobre épocas de EEG de 20 segundos siguiendo criterios estandarizados [8]. Como se explicó en la sección 3.1, antes de realizar el marcado se filtró la señal con un filtro pasa banda de $[0,5\ 30]Hz$.

La detección de ondas lentas se llevó a cabo con el algoritmo descrito en la sección 3.2 utilizando los parámetros por defecto. También se aplicó la modificación del algoritmo descrita en la sección 3.2.1 aunque estos resultados no se incluyeron en el reporte del proyecto.

La detección de husos se hizo tanto manual como automática. Para la detección manual se aplicó un filtro pasa banda de $[0,5\ 30]Hz$ y se tomaron en cuenta únicamente los canales C3, C1, Cz, C2, C4, CP3, CP1, CP2, CP4, P3, Pz y P4. Los husos se detectaron solo en el canal que presentará la mayor amplitud. Para la detección automática se utilizó la mejor al algoritmo de detección presentada en la sección 3.3.1. Se utilizaron los mismos canales que para la detección manual y se optó por restringir la detección a la banda de husos rápidos ($[13\ 15]Hz$). Los demás parámetros del algoritmo quedaron con su valor por omisión.

Los histogramas de correlación, descritos en la sección 3.4, se aplicaron también utilizando los parámetros por omisión.

Los índices de reconocimiento de memoria, así como los estadísticas que relacionan estos índices con los parámetros de los husos y ondas lentas, fueron calculados con el programa SPSS v.15 (SPSS Inc., Chicago, IL). Los resultados relacionados pueden ser consultados en el documento del proyecto de tesis, sin embargo, se ha decidido omitirlos en este trabajo ya que fueron calculados utilizando otros paquetes de software.

Resultados

Macro-estructura del sueño

Al comparar las dos noches de sueño se encontraron diferencias en el tiempo pasado en vigilia y por consiguiente en el tiempo total de sueño. Sin embargo, el SOL no se vio afectado como consecuencia de la posible falta de sueño en la primera noche. Es más, fue la fase REM la que se vio aumentada en la noche post-entrenamiento. Los estadísticos descriptivos correspondientes a la macro-estructura del sueño se muestran en la Tabla 5.1.

Cuadro 5.1: Parámetros relacionados con la macro-estructura del sueño en la noche pre y post-entrenamiento.

Parámetro	Minutos		
	Pre-entrenamiento	Post-entrenamiento	$p <$
Vigilia	49,30±35,55	16,47±10,57	0,003
Fase 1	28,10±19,45	28,60±12,98	0,93
Fase 2	185,34±55,41	185,51±24,92	0,99
SOL	123,43±34,01	132,71±22,74	0,37
REM	109,06±21,31	127,89±25,01	0,03
Tiempo total de sueño	445,93±40,37	474,72±21,71	0,01
Latencia al comienzo del sueño	16,57±18,02	10,32±8,81	0,25
Latencia al SOL	31,98±25,94	25,28±12,20	0,38
Latencia al REM	108,36±34,40	84,61±34,12	0,10
	Porcentaje		
Vigilia	9,98±7,30	3,35±2,12	0,004
Fase 1	5,70±3,95	5,82±2,63	0,91
Fase 2	37,36±11,13	37,82±5,23	0,90
SOL	24,95±6,93	27,05±4,75	0,31
REM	22,02±4,25	25,97±4,75	0,02

Micro-estructura del sueño: Husos de sueño y ondas lentas

Al comparar los diferentes parámetros que caracterizan a los husos de sueño ($[13\ 15]Hz$) entre el SOL de la noche pre y post-entrenamiento para cada electrodo en los que estos fueron identificados, primero de forma manual y luego de forma automática, encontramos diferencias significativas en el parámetro densidad (número de husos/min) en 6 de estos electrodos (C1, Cz, C2, CP1, CP2, Pz).

En la Tabla 5.2 se muestran la media y desviación estándar en ambas noches para cada uno de los parámetros en los que hubo diferencias significativas en al menos dos de los electrodos analizados. Puede verse que además de un aumento en la densidad, se produjo un aumento significativo de la amplitud en Cz y CP1 y un aumento de la duración en electrodos parietales. Cuando se corrigió por múltiples comparaciones en el dominio espacial, únicamente la densidad en C2 superó el umbral de significación estadística derivado del procedimiento de permutaciones.

En el caso de las ondas lentas, además de la densidad se evaluó la duración y la amplitud de la fase despolarizante (up-state) e hiperpolarizante (down-state). Únicamente se encontraron diferencias para la amplitud del up-state. Aunque debido a la gran consistencia de este parámetro dentro de un mismo sujeto las diferencias fueron muy pequeñas (ver Figura 5.1), fueron evidentes en todos los electrodos utilizados para su detección automática ($p < 10^{-10}$).

Sincronía temporal entre los husos y las ondas lentas

	Noche 1	Noche 2	$p <$
Densidad (husos/min)			
C1	0,35±0,25	0,49±0,30	0,04
Cz	0,50±0,35	0,72±0,44	0,01
C2	0,39±0,24	0,59±0,38	0,01
CP1	0,63±0,51	0,98±0,82	0,02
CP2	0,47±0,38	0,68±0,51	0,01
Pz	0,55±0,48	0,84±0,71	0,02
Duración (s)			
P3	0,64±0,26	0,85±0,15	0,03
Pz	0,74±0,69	0,79±0,09	0,04
Amplitud (μV)			
Cz	7,44±3,14	6,95±2,95	0,03
CP1	6,82±2,94	6,43±2,86	0,04

Cuadro 5.2: Parámetros relacionados con los husos de sueño. Únicamente se muestran aquellos electrodos donde se han encontrado diferencias significativas en alguno de los parámetros.

Al evaluar la sincronía entre los husos y las ondas lentas puede observarse que los husos se concentran en mayor medida en el up-state de las ondas lentas y en menor medida en el down-state. Este hecho se ilustra en la Figura 5.1, donde en el panel superior muestra el número de husos detectados a lo largo de la onda lenta para cada una de las noches, mientras que el panel inferior muestra las ondas lentas obtenidas en las noches pre y post-entrenamiento superpuestas con el valor cuadrático medio de los husos alineados con respecto al pico mínimo de la onda lenta. De esta forma es posible observar además que la máxima amplitud de los husos se produce también en el up-state de la onda lenta.

No obstante, el resultado más novedoso es que esta distribución de los husos en relación con las fases hiperpolarizante y despolarizante de la onda lenta cambió significativamente de la noche pre a la noche post-entrenamiento, como puede verse en la Figura 5.2. Esta figura muestra el número de husos a lo largo de la onda lenta tras restarle la media en los primeros 500ms en los 3s previos al pico negativo de la onda lenta. Las líneas horizontales que aparecen en la parte superior son los intervalos en los que hubo un aumento o decremento de la señal con respecto a la línea base en una o ambas noches, mientras que los asteriscos hacen referencia a los cambios significativos que se produjeron de la noche pre a la noche postentrenamiento. Como puede verse, en la noche post-entrenamiento hubo una disminución del número de husos en el down-state y un incremento en el up-state.

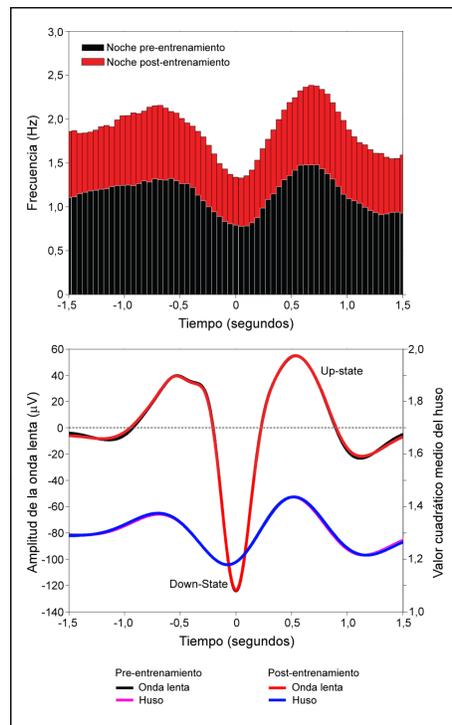


Figura 5.1: El panel superior muestra los histogramas para los husos en la noche pre (negro) y post-entrenamiento (rojo). El panel inferior muestra las ondas lentas promediadas en todos los electrodos y sujetos en la noche pre (rojo) y post-entrenamiento (negro) superpuesta por el valor cuadrático medio de los husos de sueño alineados con respecto al pico mínimo de la onda lenta en la noche pre (rosa) y post-entrenamiento (azul).

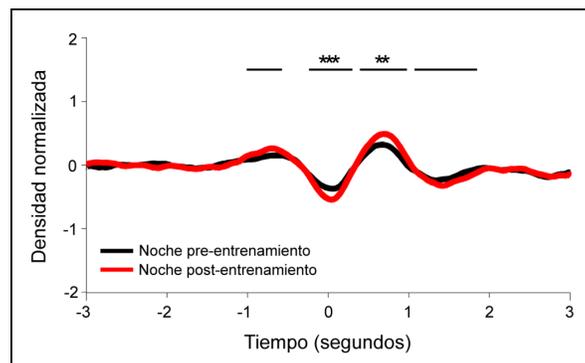


Figura 5.2: Densidad de husos normalizada en la noche pre (negro) y post-entrenamiento (rojo) tras promediar todos los electrodos y sujetos. El conteo de husos se ha realizado tomando como referencia el pico mínimo de la onda lenta. Las líneas negras horizontales en la parte superior indican los intervalos de tiempo en los que la densidad en la noche pre y/o post-entrenamiento fueron significativamente diferentes de la línea base (tras corregir por múltiples comparaciones en el dominio del tiempo). Los asteriscos indican los bins de frecuencia en donde la densidad fue significativamente diferente al comparar la noche pre y post-entrenamiento.

Capítulo 6

Conclusiones

El sistema que se presenta en este proyecto fue diseñado para automatizar las tareas requeridas para el análisis de las relaciones existentes entre ondas lentas y husos del sueño en registros EEG humanos. Como se explicó en secciones anteriores, los requerimientos del sistema obedecen a las necesidades, relacionadas con el procesamiento de los registros del sueño, de los integrantes del Laboratorio de Neurociencia Funcional, Universidad Pablo de Olavide (Sevilla, España). Esto quiere decir que cada módulo del sistema se relaciona con una tarea que se hacía previamente por medio de software especializado o que se incluye como nueva técnica de análisis.

El módulo dedicado a la detección de ondas lentas gira en torno al algoritmo estándar [14, 15] utilizado ampliamente por neurocientíficos. El *script* encargado de esta ejecución reporta las características más importantes de las ondas lentas.

Para llevar a cabo la detección de husos se implementó el algoritmo basado en la propuesta de Mölle et al. 2009 [11] así como diversas mejoras para intentar aumentar su precisión. Las mejoras incluyeron un enfoque híbrido que extrae información de la detección manual para calibrar los parámetros de la detección automática, y una serie de mejoras a la detección manual que ayudan a detectar mayor cantidad de husos. También se implementaron técnicas de rechazo de husos detectados que utilizan información de otros electrodos del registro para asegurarse que la detección no incluyera actividad que no corresponde a los husos.

El análisis temporal entre las ondas lentas y los husos del sueño se incluyó en el sistema con la implementación del algoritmo de histogramas de correlación propuesto por Mölle et al. 2009. [11]. El resultado de esta herramienta es, como su nombre lo dice, un histograma que muestra la co-ocurrencia de los dos eventos en el tiempo. Además, dado que las ondas lentas presentan mucho menor frecuencia que los husos del sueño, estos histogramas le sirven a los investigadores para observar en qué partes de la onda lenta es más común la ocurrencia de un huso.

En este trabajo se decidió incluir también una herramienta para el análisis del acoplamiento multifrecuencia entre ondas lentas y husos. Este tipo de herramientas no habían sido aplicadas anteriormente al estudio de estos dos patrones relacionados con la consolidación de la memoria durante el sueño. Para añadir esta funcionalidad en el sistema

se optó por implementar el algoritmo para el cálculo del índice de modulación (IM por sus siglas en inglés) propuesto por Tort et al. 2010 [16]. Los resultados que arroje este algoritmo al ser aplicado a los dos eventos de interés servirán para que los investigadores puedan reforzar o plantear nuevas hipótesis sobre la relación de las ondas lentas con los husos del sueño.

En el módulo de herramientas del sistema se incluyeron scripts para realizar tareas de lectura y escritura de datos, filtrado de señales y otras operaciones utilizadas comúnmente por los demás módulos. En este caso se hizo énfasis en optimizar en la medida de lo posible la implementación de las herramientas. El diseño modular del sistema en general hace que cualquier herramienta implementada pueda ser ejecutada en otros paquetes de software sin implicar modificaciones significativas.

Todos los algoritmos tanto de detección de oscilaciones y de análisis de sincronización contienen herramientas para calcular estadísticas que comparan los resultados entre grupos o condiciones experimentales. En general, estas herramientas hacen uso del estadístico t-Student y de alguna corrección por múltiples comparaciones. En el caso de los histogramas de correlación y el acoplamiento multifrecuencia, las estadísticas además generan gráficos que facilitan la visualización y reporte de los resultados.

Como se puede ver en el capítulo 4, cada módulo del sistema encapsula alguno de los pasos requeridos para procesar registros EEG. Las funciones que incluye cada módulo se organizaron para imitar en la medida de lo posible las tareas que se realizaban previamente (de manera manual o con herramientas aisladas) para procesar los registros. Cada módulo incluye los *scripts* para ejecutar y reportar un algoritmo (detección o análisis de sincronización) así como un *script* diseñado para hacer la comparación estadística entre los resultados arrojados por grupo o por condición experimental.

Finalmente, el sistema presentado en este trabajo es una caja de herramientas para lenguaje Matlab® que integra los pasos requeridos para procesar y analizar registros EEG del sueño. Cada módulo está aislado de los demás para facilitar su modificación o la posible inclusión de nuevas funcionalidades. Los *scripts* están documentados con el formato de ayuda del ambiente Matlab® y se entrega un manual de usuario que explica el modo de configurar y ejecutar el sistema.

6.1. Trabajo futuro

Durante el diseño y desarrollo del sistema se encontraron algunos problemas que merecían dedicar tiempo y esfuerzo para ser solucionados. Los siguientes tres puntos resumen propuestas de trabajo futuro en tres áreas principales: mejora del algoritmo de detección de husos de sueño, implementación de más algoritmos para el cálculo del acoplamiento multifrecuencia y mejoras en el diseño del sistema presentado.

- La detección automática de husos del sueño sigue siendo un problema abierto en el análisis de registros EEG. El problema resulta ser especialmente complejo debido a la alta variabilidad que se puede encontrar en los registros entre un paciente y otro. Aunado a esto, existen situaciones en las que incluso para los expertos es difícil llegar

al consenso de marcar cierta actividad como un huso. En los últimos años se han hecho muchos esfuerzos por mejorar estos algoritmos utilizando técnicas de distintas áreas de la computación. Entre los trabajos más importantes se puede encontrar el uso de técnicas de inteligencia artificial como redes neuronales [20] y arboles de decisiones [68], operaciones avanzadas de procesamiento de señales como la energía de Teager [22, 69] o la transformada Hilbert-Huang [70], y operadores estadísticos como los modelos estado-espacio (state-space models) [71]. A pesar de todos estos esfuerzos, la comunidad neurocientífica sigue requiriendo algoritmos más confiables para la detección de husos.

- Los algoritmos de acoplamiento multifrecuencia son relativamente nuevos en el área de neurociencias, y constantemente se proponen mejoras o modificaciones a los ya existentes. Algunos paquetes de software científico incluyen implementaciones de estos algoritmos, por ejemplo, *Brainstorm*¹ ofrece análisis de acoplamiento tipo PAC (fase-amplitud) y *FieldTrip*² ofrece análisis tanto PAC como PPC (amplitud-amplitud). Sin embargo, es importante que se hagan más implementaciones públicas y se ofrezcan a los investigadores como herramientas de análisis. Gracias a su diseño modular, el sistema presentado en este proyecto podría incluir la implementación de otros algoritmos sin requerir cambios significativos.
- Para facilitar el uso del sistema presentado es posible diseñar una interfaz gráfica de usuario (GUI por sus siglas en inglés) que ayude a configurar y ejecutar los diversos pasos del proceso de análisis de registros EEG. En su estado actual, el sistema ofrece interfaces intuitivas y documentadas en línea de comandos, sin embargo, incluir una GUI puede disminuir la curva de aprendizaje para usuarios que no tienen experiencia con el ambiente de trabajo de Matlab®.

¹<http://neuroimage.usc.edu/brainstorm/Tutorials/TutPac>

²http://fieldtrip.fcdonders.nl/example/cross_frequency_analysis

Bibliografía

- [1] Paul W Frankland and Bruno Bontempi. The organization of recent and remote memories. *Nature reviews. Neuroscience*, 6(2):119–30, February 2005.
- [2] Susanne Diekelmann and Jan Born. The memory function of sleep. *Nature reviews. Neuroscience*, 11(2):114–26, February 2010.
- [3] Marion Inostroza and Jan Born. Sleep for preserving and transforming episodic memory. *Annual review of neuroscience*, 36:79–102, July 2013.
- [4] Björn Rasch and Jan Born. About sleep’s role in memory. *Physiological reviews*, 93(2):681–766, April 2013.
- [5] L. R. Squire, N. J. Cohen, and L. Nadel. The medial temporal region and memory consolidation: A new hypothesis. pages 185–210, 1984.
- [6] M Steriade and F Amzica. Coalescence of sleep rhythms and their chronology in corticothalamic networks. *Sleep research online : SRO*, 1(1):1–10, January 1998.
- [7] György Buzsáki and Xiao-Jing Wang. Mechanisms of gamma oscillations. *Annual review of neuroscience*, 35:203–25, January 2012.
- [8] A Rechtschaffen and A Kales. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *Electroencephalography and Clinical Neurophysiology*, 26(6):644, June 1968.
- [9] Mircea Steriade and Igor Timofeev. Neuronal plasticity in thalamocortical networks during sleep and waking oscillations. *Neuron*, 37(4):563–76, February 2003.
- [10] M Steriade, D Contreras, R Curro Dossi, and A Nunez. The slow (1 Hz) oscillation in reticular thalamic and thalamocortical neurons: scenario of sleep rhythm generation in interacting thalamic and neocortical networks. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 13(8):3284–99, August 1993.
- [11] Matthias Mölle, Oxana Eschenko, Steffen Gais, Susan J Sara, and Jan Born. The influence of learning on sleep slow oscillations and associated spindles and ripples in humans and rats. *The European journal of neuroscience*, 29(5):1071–81, March 2009.

- [12] John E Lisman and Ole Jensen. The θ - γ neural code. *Neuron*, 77(6):1002–16, March 2013.
- [13] Nikolai Axmacher, Melanie M Henseler, Ole Jensen, Ilona Weinreich, Christian E Elger, and Juergen Fell. Cross-frequency coupling supports multi-item working memory in the human hippocampus. *Proceedings of the National Academy of Sciences of the United States of America*, 107(7):3228–33, February 2010.
- [14] Matthias Mölle, Lisa Marshall, Steffen Gais, and Jan Born. Grouping of spindle activity during slow oscillations in human non-rapid eye movement sleep. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 22(24):10941–7, December 2002.
- [15] Marcello Massimini, Reto Huber, Fabio Ferrarelli, Sean Hill, and Giulio Tononi. The sleep slow oscillation as a traveling wave. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 24(31):6862–70, August 2004.
- [16] Adriano B L Tort, Mark A Kramer, Catherine Thorn, Daniel J Gibson, Yasuo Kubota, Ann M Graybiel, and Nancy J Kopell. Dynamic cross-frequency couplings of local field potential oscillations in rat striatum and hippocampus during performance of a T-maze task. *Proceedings of the National Academy of Sciences of the United States of America*, 105(51):20517–22, December 2008.
- [17] Andreas M Koupparis, Vasileios Kokkinos, and George K Kostopoulos. Semi-automatic sleep EEG scoring based on the hypnospectrogram. *Journal of neuroscience methods*, 221:189–95, January 2014.
- [18] Antoine Picot, Harry Whitmore, and Florian Chapotot. Automated detection of sleep EEG slow waves based on matching pursuit using a restricted dictionary. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011:4824–7, January 2011.
- [19] Fazil Duman, Aykut Erdamar, Osman Eroglu, Ziya Telatar, and Sinan Yetkin. Efficient sleep spindle detection algorithm with decision tree. *Expert Systems with Applications*, 36(6):9980–9985, August 2009.
- [20] Errikos M Ventouras, Efstratia A Monoyiou, Periklis Y Ktonas, Thomas Paparrigopoulos, Dimitris G Dikeos, Nikos K Uzunoglu, and Constantin R Soldatos. Sleep spindle detection using artificial neural networks trained with filtered time-domain EEG: a feasibility study. *Computer methods and programs in biomedicine*, 78(3):191–207, June 2005.
- [21] E. Huupponen, A. Värri, S.-L. Himanen, J. Hasan, M. Lehtokangas, and J. Saarinen. Optimization of sigma amplitude threshold in sleep spindle detection. *Journal of Sleep Research*, 9(4):327–334, 2000.

- [22] Beena Ahmed, Amira Redissi, and Reza Tafreshi. An automatic sleep spindle detector based on wavelets and the teager energy operator. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2009:2596–9, January 2009.
- [23] Adriano B L Tort, Robert Komorowski, Howard Eichenbaum, and Nancy Kopell. Measuring phase-amplitude coupling between neuronal oscillations of different frequencies. *Journal of neurophysiology*, 104(2):1195–210, August 2010.
- [24] Michael X Cohen. Assessing transient cross-frequency coupling in EEG data. *Journal of neuroscience methods*, 168(2):494–9, March 2008.
- [25] Angela C E Onslow, Rafal Bogacz, and Matthew W Jones. Quantifying phase-amplitude coupling in neuronal network oscillations. *Progress in biophysics and molecular biology*, 105(1-2):49–57, March 2011.
- [26] R T Canolty, E Edwards, S S Dalal, M Soltani, S S Nagarajan, H E Kirsch, M S Berger, N M Barbaro, and R T Knight. High gamma power is phase-locked to theta oscillations in human neocortex. *Science (New York, N.Y.)*, 313(5793):1626–8, September 2006.
- [27] Andreas Bruns and Reinhard Eckhorn. Task-related coupling from high- to low-frequency signals among visual cortical areas in human subdural recordings. *International Journal of Psychophysiology*, 51(2):97–116, January 2004.
- [28] Daria Osipova, Dora Hermes, and Ole Jensen. Gamma power is phase-locked to posterior alpha activity. *PloS one*, 3(12):e3990, January 2008.
- [29] Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1):9–21, March 2004.
- [30] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational intelligence and neuroscience*, 2011:156869, January 2011.
- [31] O. Sporns. Brain connectivity. *Scholarpedia*, 2(10):4695, 2007. revision 91083.
- [32] Eric Kandel, James JH Schwartz, and TM Thomas Jessell. *Principles of Neural Science*. McGraw-Hill Medical, fourth edition, 2000.
- [33] DL Donald L. Schomer and FL Da Fernando Lopes da Silva. *Niedermeyer’s electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 6 edition, 2010.

- [34] Michael M Halassa, Tommaso Fellin, Hajime Takano, Jing-Hui Dong, and Philip G Haydon. Synaptic islands defined by the territory of a single astrocyte. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(24):6473–7, June 2007.
- [35] Andrea Volterra and Jacopo Meldolesi. Astrocytes, from brain glue to communication elements: the revolution continues. *Nature reviews. Neuroscience*, 6(8):626–40, August 2005.
- [36] Purves. *Neuroscience*. Third edition, 2007.
- [37] HH Jasper. The ten-twenty system of the international federation. *Electroencephalography and Clinical Neurophysiology*, 10:371–375, 1958.
- [38] Jose L Cantero, Mercedes Atienza, Robert Stickgold, Michael J Kahana, Joseph R Madsen, and Bernat Kocsis. Sleep-dependent theta oscillations in the human hippocampus and neocortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 23(34):10897–903, November 2003.
- [39] Hans Berger. Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, 87(1):527–570, December 1929.
- [40] Michael H Silber, Sonia Ancoli-Israel, Michael H Bonnet, Sudhansu Chokroverty, Madeleine M Grigg-Damberger, Max Hirshkowitz, Sheldon Kapen, Sharon A Keenan, Meir H Kryger, Thomas Penzel, Mark R Pressman, and Conrad Iber. The visual scoring of sleep in adults. *Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine*, 3(2):121–31, March 2007.
- [41] J Zeitlhofer, G Gruber, P Anderer, S Asenbaum, P Schimicek, and B Saletu. Topographic distribution of sleep spindles in young healthy subjects. *Journal of sleep research*, 6(3):149–55, September 1997.
- [42] M Steriade, M Deschênes, L Domich, and C Mulle. Abolition of spindle oscillations in thalamic neurons disconnected from nucleus reticularis thalami. *Journal of neurophysiology*, 54(6):1473–97, December 1985.
- [43] Péter Halász. K-complex, a reactive EEG graphoelement of NREM sleep: an old chap in a new garment. *Sleep medicine reviews*, 9(5):391–412, October 2005.
- [44] Kimberly A. Cote, TinA. M. Epps, and Kenneth B. Campbell. The role of the spindle in human information processing of high-intensity stimuli during sleep. *Journal of sleep research*, 9(1):19–26, March 2000.
- [45] Péter Halász. Hierarchy of micro-arousals and the microstructure of sleep. *Neurophysiologie Clinique/Clinical Neurophysiology*, 28(August 1997):24–30, 1998.

- [46] Florin Amzica and Mircea Steriade. The functional significance of K-complexes. *Sleep Medicine Reviews*, 6(2):139–149, 2002.
- [47] M Steriade, A Nuiiez, Laboratoire De Neurophysiologie, Faculté De Médecine, Université Laval, Canada Glk, A Nunez, and F Amzica. A Novel Slow (≈ 1 Hz) Oscillation Depolarizing and Hyperpolarizing of Neocortical Neurons. *Journal of Neuroscience*, 73(August):3252–3265, August 1993.
- [48] José Luis Cantero and Mercedes Atienza. Alpha burst activity during human REM sleep: descriptive study and functional hypotheses. *Clinical Neurophysiology*, 111(5):909–915, 2000.
- [49] M H Chase and F R Morales. The atonia and myoclonia of active (REM) sleep. *Annual review of psychology*, 41:557–84, January 1990.
- [50] Zahir M. ZM Hussain, Amin Z. AZ Sadik, and Peter O’Shea. *Digital signal processing: an introduction with MATLAB and applications*. Springer, 2011.
- [51] Richard G. Lyons. *Understanding Digital Signal Processing (3rd Edition)*. Prentice Hall, 2010.
- [52] John G. Proakis and Dimitris K Manolakis. *Digital Signal Processing*. Prentice Hall, 4 edition, 2006.
- [53] James W Cooley and John W Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90):297–301 CR – Copyright © 1965 American Mathem, April 1965.
- [54] C S Burrus. *Introduction to wavelets and wavelet transforms : a primer*. Prentice Hall, Upper Saddle River, N.J, 1998.
- [55] M.N.O. Sadiku, C. Akujuobi, and R.C. Garcia. An introduction to wavelets in electromagnetics. *IEEE Microwave Magazine*, 6(2):63–72, June 2005.
- [56] V Cizek. Discrete Hilbert transform, 1970.
- [57] Jr. Marple S.L. Computing the discrete-time ‘analytic’ signal via FFT. In *Signals, Systems and Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 2, pages 1322–1325 vol.2, November 1997.
- [58] B Boashash. Time Frequency Signal Analysis and Processing: A Comprehensive Reference. 2003. *Elsevier*.
- [59] Conrad Iber, Sonia Ancoli-Israel, Andrew L. Chesson Jr., and Stuart F. Quan. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. Westchester, Illinois, 1 edition, 2007.

- [60] Sleep homeostasis and cortical synchronization: III. A high-density EEG study of sleep slow waves in humans. *Sleep*, 30(12):1643–57, December 2007.
- [61] J Matias Palva, Satu Palva, and Kai Kaila. Phase synchrony among neuronal oscillations in the human cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 25(15):3962–72, April 2005.
- [62] Bradley Voytek, Ryan T Canolty, Avgusta Shestyuk, Nathan E Crone, Josef Parvizi, and Robert T Knight. Shifts in gamma phase-amplitude coupling frequency from theta to alpha over posterior cortex during visual tasks. *Frontiers in human neuroscience*, 4:191, January 2010.
- [63] J E Lisman and M A Idiart. Storage of 7 ± 2 short-term memories in oscillatory subcycles. *Science (New York, N.Y.)*, 267(5203):1512–5, March 1995.
- [64] Charles E Schroeder and Peter Lakatos. Low-frequency neuronal oscillations as instruments of sensory selection. *Trends in neurosciences*, 32(1):9–18, January 2009.
- [65] Keith M Kendrick, Yang Zhan, Hanno Fischer, Alister U Nicol, Xuejuan Zhang, and Jianfeng Feng. Learning alters theta amplitude, theta-gamma coupling and neuronal synchronization in inferotemporal cortex. *BMC neuroscience*, 12(1):55, January 2011.
- [66] Ryan T Canolty and Robert T Knight. The functional role of cross-frequency coupling. *Trends in cognitive sciences*, 14(11):506–15, November 2010.
- [67] Jose M Hurtado, Leonid L Rubchinsky, and Karen A Sigvardt. Statistical method for detection of phase-locking episodes in neural oscillations. *Journal of neurophysiology*, 91(4):1883–98, April 2004.
- [68] Fazil Duman, Aykut Erdamar, Osman Eroglu, Ziya Telatar, and Sinan Yetkin. Efficient sleep spindle detection algorithm with decision tree. *Expert Systems with Applications*, 36(6):9980–9985, August 2009.
- [69] Syed Anas Imtiaz, Siavash Saremi-Yarahmadi, and Esther Rodriguez-Villegas. Automatic detection of sleep spindles using Teager energy and spectral edge frequency. In *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 262–265. IEEE, October 2013.
- [70] Leonardo Causa, Claudio M Held, Javier Causa, Pablo A Estévez, Claudio A Perez, Rodrigo Chamorro, Marcelo Garrido, Cecilia Algarín, and Patricio Peirano. Automated sleep-spindle detection in healthy children polysomnograms. *IEEE transactions on bio-medical engineering*, 57(9):2135–46, September 2010.
- [71] Tracey A. Camilleri, Kenneth P. Camilleri, and Simon G. Fabri. Automatic detection of spindles and K-complexes in sleep EEG using switching multiple models. *Biomedical Signal Processing and Control*, 10:117–127, March 2014.