

Práctica 4 – Procesado de potenciales evocados auditivos reales

Procesamiento de Señales Biomédicas

Universidad de Granada

Joaquín T. Valderrama*

Curso 2024–2025

Resumen de la práctica

Esta práctica consiste en el procesamiento de una base de datos de señales de electroencefalografía (EEG) reales, en el que se han registrado potenciales evocados auditivos del tronco cerebral (ABR, por sus siglas en inglés: *auditory brainstem response*) a diferentes tasas de estimulación. Se propone implementar un flujo completo de procesamiento de señales EEG: desde la lectura y limpieza de datos, hasta la estimación, caracterización morfológica y análisis estadístico de las señales ABR. Estos procedimientos se implementarán en los 5 pasos siguientes.

1. Desarrollo de un *script* (S1) que lea una señal EEG, rechace el 20 % de las tramas más ruidosas, y estime la señal ABR.
2. Conversión del script S1 a una *función* (F1) que reciba como parámetros de entrada la identificación del participante y la tasa de estimulación, y devuelva la señal ABR y el tiempo de procesamiento.
3. Desarrollo de un *script* (S2) que procese toda la base de datos de manera automática utilizando la función F1, y almacene los ABR procesados en un archivo.
4. Desarrollo de un *script* (S3) que lea el archivo con los ABR procesados, y caracterice las principales componentes en términos de latencias y amplitudes, almacenando las medidas en un archivo.
5. Desarrollo de un *script* (S4) que lea el archivo con las medidas, y realice un análisis estadístico de latencias y amplitudes.

Objetivos de aprendizaje

- Implementar una técnica de mejora de la calidad basada en el rechazo de trama.
- Estimar señales ABR mediante deconvolución matricial.
- Procesar una base de datos de manera automática.
- Caracterizar la morfología de señales ABR en términos de amplitudes y latencias.
- Consolidar técnicas avanzadas de generación de figuras con MATLAB.
- Redactar un documento científico con rigor y claridad.

*Email: jvalderrama@ugr.es — [Página web](#)

Descripción de la base de datos

- La base de datos consta de 4 sujetos normoyentes (1 mujer, 26–36 años), seleccionados del estudio realizado por [Valderrama et al. \(2012\)](#) con la metodología *Randomised Stimulation and Averaging* (RSA).
- En este estudio se obtuvieron señales ABR a 8 tasas de estimulación, en el que el intervalo entre-estímulos (ISI, por sus siglas en inglés: *inter-stimulus interval*) varió de manera aleatoria entre dos valores dados conforme a una distribución uniforme de probabilidad. Por ejemplo, la secuencia de estimulación ISI_{20-24} indica que el ISI varió de manera aleatoria según una distribución uniforme de probabilidad entre 20 y 24 ms. Las secuencias de estimulación utilizadas fueron: ISI_{20-24} , ISI_{16-20} , ISI_{12-16} , ISI_{10-14} , ISI_{8-12} , ISI_{6-10} , ISI_{4-8} y ISI_{2-6} , correspondientes con tasas de estimulación promedio de 50.0, 55.6, 71.4, 83.3, 100.0, 125.0, 166.7 y 250.0 Hz, respectivamente.
- El estímulo acústico utilizado fueron clics bifásicos de 100 μs de duración a 70 dB por encima del umbral de audición (dB nHL, por sus siglas en inglés: *dB normal hearing level*).
- El número de estímulos utilizados fue alrededor de 20,000 estímulos en todos los escenarios, excepto en el caso ISI_{2-6} , donde se utilizaron el doble de estímulos con el fin de compensar la disminución de la amplitud de las componentes con una mayor calidad de la señal ABR.
- Las señales EEG de la base de datos se registraron a una tasa de muestreo de 25 kHz, y han sido filtradas mediante un filtro Butterworth de 2° orden con frecuencias de corte [300–3000] Hz, por lo que en esta práctica no es necesario implementar ningún filtrado digital adicional.
- La base de datos de esta práctica puede descargarse del siguiente [Link de consigna](#) [464.4 MB].
- Una vez descargado el archivo “Database.zip”, crea una estructura de carpetas similar a la expuesta en la Figura 1 en el directorio que utilices para esta práctica. Descomprime el archivo “Database.zip” en la carpeta “Database”. Comprueba que en esta carpeta obtienes 4 carpetas (una para cada participante, con los nombres ‘P01’ hasta ‘P04’) y que en la carpeta de cada participante dispones de 8 archivos indicados con diferentes ISIs (por ejemplo, ‘ISI_20-24.mat’).
- En la carpeta ‘Results’ se almacenarán las figuras y archivos generados. El resto de archivos *.m del directorio corresponden con los scripts y funciones a desarrollar en la práctica.

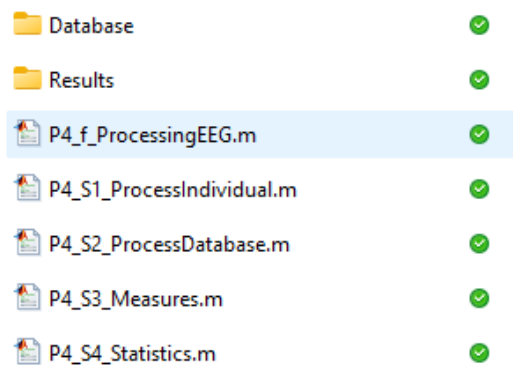


Figura 1: Estructura de carpetas propuesta.

Criterios de corrección

La Tabla 1 presenta la puntuación de cada sección. Secciones correctamente resueltas y descritas en la memoria de forma clara puntuarán con el 100 % de la calificación máxima; secciones con errores menores pero con descripciones claras puntuarán con el 75 %; errores menores con descripciones insuficientes puntuarán con el 50 %; y errores significativos o descripciones confusas ponderarán por debajo del 50 %.

Sección	Puntuación máxima
1. [S1] Script que procesa un archivo individual	3 puntos
2. [F1] Función que procesa un archivo indicado	1.5 puntos
3. [S2] Script que procesa toda la base de datos	1.5 puntos
4. [S3] Medidas manuales de amplitudes y latencias	2 puntos
5. [S4] Análisis de latencias y amplitudes	2 puntos

Tabla 1: Puntuación de cada sección.

Estructura cada sección (1) haciendo un repaso de lo que se pide en la actividad, (2) describiendo los pasos realizados, y (3) describiendo en detalle figuras y tablas. Finalmente, como apéndice al final de todas las secciones, incluye el código de MATLAB utilizado.

IMPORTANTE: Sigue las recomendaciones sobre redacción de documentos científicos y generación de figuras con MATLAB descritas en prácticas anteriores.

1. [S1] Script que procesa un archivo individual

Objetivo: Estimar la señal ABR para un participante y una tasa concreta.

- Crea un script con el nombre 'P4_S1_ProcessIndividual' o cualquier otro nombre representativo.
- Borra el espacio de trabajo, limpia la ventana de comandos, y cierra todas las figuras mediante `close all`.
- Crea una primera sección en el script mediante el comando `%%` para inicializar las variables que definan el archivo a procesar, en particular: (1) el participante y (2) la tasa de estimulación.
- En la segunda sección:
 - Utiliza el comando `sprintf` para almacenar en una variable el nombre del archivo junto con su directorio. Por ejemplo, `FileName = sprintf('Database/P0%s/ISI_%s.mat',P,ISI);` almacena en la variable `FileName` el directorio y el nombre del archivo como una variable tipo *string* (cadena de caracteres). El directorio del archivo puede especificarse desde la carpeta de trabajo de MATLAB, es decir, donde se encuentra almacenado el script. Recuerda que en las funciones `fprintf` y `sprintf`, '`\n`' implementa un salto de línea, '`\t`' implementa una tabulación, '`%s`' se utiliza para referenciar cadenas de caracteres, '`%d`' para referenciar números enteros, y '`%1.2f`' para referenciar números con 2 decimales.
 - Lee el archivo especificado, y define el valor de las siguientes variables: EEG (señal EEG), m (vector de posiciones de los estímulos), fs (frecuencia de muestreo), N (número de muestras del EEG), J (número de muestras del ABR correspondiente a 15 ms), K (número de eventos).
- Caracteriza el EEG registrado en los dominios del tiempo y la frecuencia mediante figuras.
- Caracteriza la distribución del ISI mediante un histograma.

- Implementa una técnica de mejora de la calidad basada en el rechazo del 20% de los eventos cuyas respuestas sean más ruidosas. Para este paso, puedes estimar la energía de cada trama en términos del valor RMS (*root mean square*), establecer un umbral correspondiente con el percentil 80 mediante la función `prctile`, establecer la condición de aceptación de trama para los casos en los que el valor RMS sea inferior al umbral, y actualizar las variables `m` y `K` de manera que incluyan exclusivamente las tramas aceptadas.
- Estima la señal ABR mediante el algoritmo de deconvolución matricial (de la Torre *et al.*, 2019), descrito en el Tema 2 – Sección 5 del temario. En concreto, sigue estos pasos:
 1. Construye la matriz S . Puesto que la matriz S está formada principalmente por ceros, inicialízala como tipo ‘sparse’. Esto reducirá de manera significativa el uso de memoria.
 2. Obtén la matriz de autocorrelación $S^T S$.
 3. Obtén el vector suma simple de segmentos $S^T y$.
 4. Obtén el ABR promediado mediante $S^T y / K$.
 5. Obtén el ABR deconvolucionado mediante $(S^T S)^{-1} (S^T y)$.
- Compara la morfología del ABR deconvolucionado con respecto al ABR promediado en un caso sin solapamiento de respuestas (donde el ISI min es mayor que J, por ejemplo en el caso ISI_{20–24}) y en un caso con solapamiento de respuestas (por ejemplo, en el caso ISI_{8–12}). Discute las posibles diferencias entre el ABR promediado y deconvolucionado en ambos escenarios.
- ¿Cómo son las matrices de autocorrelación en cada caso? Puedes representar estas matrices de manera visual con el comando `imagesc`. También puedes identificar los valores únicos de las matrices con el comando `unique`.

2. [F1] Función que procesa un archivo indicado

Objetivo: Reutilizar el código del paso anterior como una función modular.

- Crea una función con el nombre ‘P4_f1_ProcessEEG.m’ o cualquier otro nombre que sea significativo e indique que el archivo es una función.
- La función debe (1) recibir como variables de entrada el participante y el ISI; y (2) devolver como variables de salida el ABR procesado y el tiempo que ha tardado la función en procesar el archivo.
- Para estimar el ABR, reutiliza las instrucciones programadas en el script S1, eliminando código redundante, como por ejemplo la generación de figuras.
- El tiempo de procesado puedes estimarlo mediante las funciones `tic` y `toc`. La función `tic` registra el tiempo actual y la función `toc` utiliza el valor registrado para calcular el tiempo transcurrido. De este modo, (1) utiliza el comando `tic` en la primera línea de código de la función para registrar el tiempo en el comienzo de la función, y (2) utiliza el comando `ProcTime = toc;` en la última línea del código de la función para almacenar en la variable `ProcTime` el tiempo transcurrido en segundos en ejecutar todos los comandos de la función. Finalmente, devuelve la variable `ProcTime` como parámetro de salida de la función.
- Comprueba el buen funcionamiento de la función llamándola desde la ventana de comandos con diferentes participantes y tasas de estimulación.

3. [S2] Script que procesa toda la base de datos

Objetivo: Estimar de forma automática los ABR de los 4 sujetos para las 8 tasas.

- Crea un nuevo script (S2).
- Borra el espacio de trabajo, limpia la ventana de comandos, y cierra todas las figuras mediante `close all`.
- En este script, crea una primera sección sobre “Inicialización de variables”, donde definas las siguientes variables:
 - Define la variable `J` correspondiente al número de muestras de un ABR de 15 ms de duración.
 - Define la variable `ISI` como un array de celdas con todos los ISIs posibles, por ejemplo: `ISI = {'20-24', '16-20', ...}`; . Posteriormente puedes acceder a las celdas de esta variable indicando el índice mediante llaves. Por ejemplo, `ISI{1}` devolvería '20-24'.
 - Inicializa la variable `ABRs` como una matriz de NaN (*not a number*) de 3 dimensiones, en donde la primera dimensión indique el participante, la segunda dimensión indique la tasa, y la tercera dimensión almacene el ABR. Este paso puedes implementarlo mediante el comando `ABRs = nan(4,8,J)`; considerando que hay 4 participantes, 8 tasas de estimulación, y cada ABR tiene longitud `J`.
- Posteriormente, crea una segunda sección que procese la base de datos. Sigue estos pasos:
 - Accede a los diferentes participantes y tasas mediante dos bucles *for*.
 - En cada caso analizado, llama a la función `F1` pasándole como parámetros de entrada el sujeto y la tasa de estimulación, y almacenando en dos variables el ABR procesado y el tiempo de procesado.
 - Almacena el ABR procesado por la función en la variable 'ABRs' de tres dimensiones previamente definida. Por ejemplo, el comando `ABRs(1,3,:) = ABR_Procesado`; almacenaría el ABR procesado para el sujeto 1 en la tasa 3 (correspondiente a un ISI [12–16] ms).
 - Utiliza los comandos `fprintf` y `sprintf` para mostrar por pantalla (1) el sujeto, (2) la tasa, y (3) el tiempo de procesado. Por ejemplo, el comando `fprintf(sprintf('Sujeto %d procesado en %1.2f segundos\n', Subject, ProcTime))` muestra por pantalla 'Sujeto 1 procesado en 1.23 segundos' cuando las variables `Subject` y `ProcTime` tienen el valor de 1 y 1.23, respectivamente. El comando `\n` implementa un retorno de carro, moviendo el cursor de la ventana de comandos a la línea siguiente. En este paso de monitorización, incluye la tasa de estimulación analizada. Las cadenas de caracteres se referencian mediante '`%s`' en la función `sprintf`.
- Finalmente, Almacena la variable 'ABRs' en el archivo 'Results/ABRs' mediante el comando `save`.

4. [S3] Medidas manuales de amplitudes y latencias

Objetivo: Identificar y medir manualmente las latencias y amplitudes de las ondas I, III y V de un participante.

- Crea un nuevo script (S3).
- Borra el espacio de trabajo, limpia la ventana de comandos, y cierra todas las figuras mediante `close all`.
- En este script, crea una primera sección en la que se seleccione un participante (e.g., `Participant = 1`);).

- En una nueva sección:
 - Lee el archivo ABRs generado en el script S2.
 - Define las variables J , f_s , e ISI como array de celdas con todos los posibles ISI.
 - Define un eje de tiempos para representar apropiadamente las señales ABR.
- En una sección aparte, genera una figura similar a la Figura 2 para cada participante, en la que se muestre el participante en el título, se identifiquen las principales componentes, y se indique qué ABR corresponde con cada tasa de estimulación. Puedes utilizar una separación de $0.2 \mu V$ entre cada señal ABR. Con la función `gtext` puedes incluir texto en una posición definida por el ratón (apropiado para las etiquetas de las componentes), mientras que la función `text` te permite incluir un texto en una posición definida mediante comandos (apropiado para las etiquetas de las tasas de estimulación).
- Finalmente, en otra sección caracterizaremos las componentes I, III y V en términos de latencias y amplitudes:
 - Inicializa las variables L y A como matrices de NaN de 8 filas y 3 columnas para almacenar los diferentes valores de latencias y amplitudes, respectivamente. Cada fila representa una tasa de estimulación (siendo la fila 1 para el ISI 20-24 y la fila 8 para el ISI 2-6) y cada columna una componente diferente (columna 1: onda I, columna 2: onda III, columna 3: onda V).
 - Crea un bucle *for* que recorra las diferentes tasas de estimulación. Para cada escenario:
 1. Marca 6 puntos relevantes en la señal ABR para caracterizar latencias y amplitudes de cada componente. Para esto, utiliza el comando `ginput(6)`; y almacena el resultado en la variable M . En la figura, marca los picos y valles de las componentes siguiendo este orden: pico onda I, valle onda I, pico onda III, valle onda III, pico onda V, valle onda V.
 2. Almacena en las posiciones correspondientes de las matrices L y A los valores de latencias y amplitudes de cada componente, obtenido a partir de la matriz de medidas M .
 - Finalmente, almacena las matrices L y A de cada participante en un archivo dentro de la carpeta 'Results'. Por ejemplo, en el archivo 'Results/P_01_Measures'.

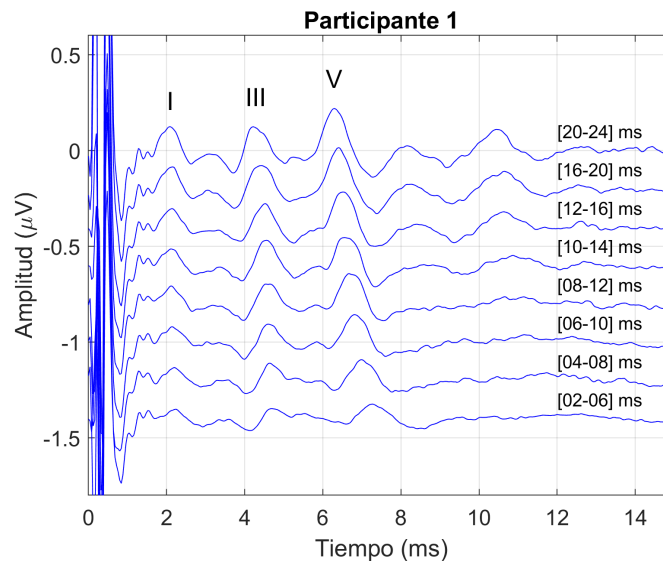
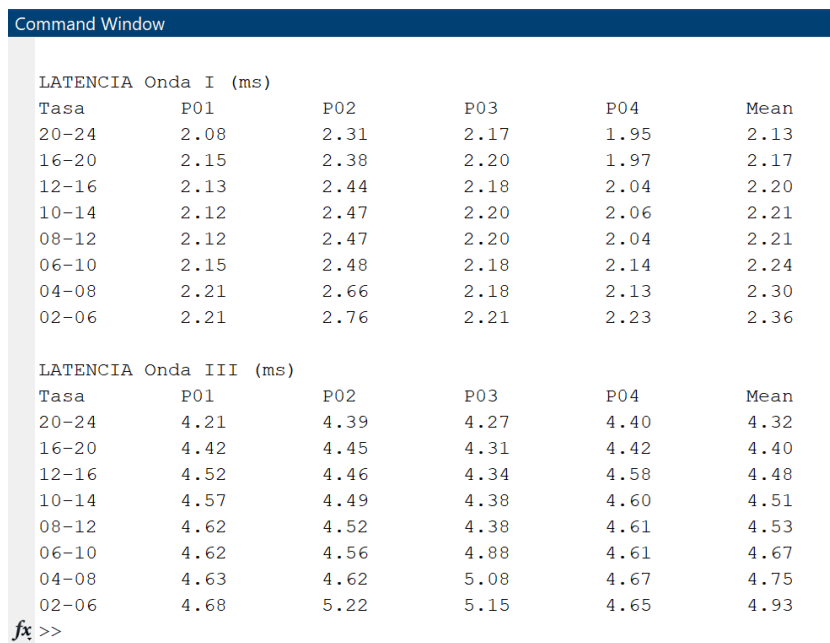


Figura 2: Señales ABR del participante 1.

5. [S4] Análisis de latencias y amplitudes

Objetivo: Representar en formato de tabla los valores de latencias y amplitudes individuales y promedio, así como evaluar mediante pruebas estadísticas posibles diferencias entre condiciones de estimulación.

- Crea un nuevo script (S3).
- Borra el espacio de trabajo, limpia la ventana de comandos, y cierra todas las figuras mediante `close all`.
- Lee los archivos que almacenan las medidas de latencias y amplitudes de los 4 participantes.
- Representa en la ventana de comandos los valores de amplitudes y latencias individuales y promedio de las tres componentes siguiendo el formato especificado en la Figura 3. Para este paso, puedes utilizar un bucle `for` que recorra las diferentes tasas de estimulación. En cada iteración utiliza las funciones `fprintf` y `sprintf` para mostrar en la ventana de comandos una cadena de caracteres. Recuerda que en estas funciones ‘\t’ inserta una tabulación.



Command Window					
LATENCIA Onda I (ms)					
Tasa	P01	P02	P03	P04	Mean
20-24	2.08	2.31	2.17	1.95	2.13
16-20	2.15	2.38	2.20	1.97	2.17
12-16	2.13	2.44	2.18	2.04	2.20
10-14	2.12	2.47	2.20	2.06	2.21
08-12	2.12	2.47	2.20	2.04	2.21
06-10	2.15	2.48	2.18	2.14	2.24
04-08	2.21	2.66	2.18	2.13	2.30
02-06	2.21	2.76	2.21	2.23	2.36

LATENCIA Onda III (ms)					
Tasa	P01	P02	P03	P04	Mean
20-24	4.21	4.39	4.27	4.40	4.32
16-20	4.42	4.45	4.31	4.42	4.40
12-16	4.52	4.46	4.34	4.58	4.48
10-14	4.57	4.49	4.38	4.60	4.51
08-12	4.62	4.52	4.38	4.61	4.53
06-10	4.62	4.56	4.88	4.61	4.67
04-08	4.63	4.62	5.08	4.67	4.75
02-06	4.68	5.22	5.15	4.65	4.93

Figura 3: Tablas de medidas de latencia individuales y promedio de las ondas I y III a modo de ejemplo.

- Finalmente, evalúa mediante el test estadístico apropiado si existen diferencias estadísticamente significativas entre las latencias de la onda V a las tasas [20–24] ms y [4–8] ms.
- ¿Y en las amplitudes? Si las diferencias no fuesen estadísticamente significativas, ¿a qué puede deberse?

Referencias

- de la Torre, A., Valderrama, J.T., Segura, J.C., Alvarez I.M. (2019). “Matrix-based formulation of the iterative randomized stimulation and averaging method for recording evoked potentials,” *The Journal of the Acoustical Society of America* 146, 4545–4556. <https://doi.org/10.1121/1.5139639>.
- Valderrama, J.T., de la Torre, A., Alvarez I.M., Segura, J.C., Sainz, M., Vargas, J.L. (2012). “Recording of auditory brainstem responses at high stimulation rates using randomized stimulation and averaging,” *The Journal of the Acoustical Society of America* 132, 3856–3865. <https://doi.org/10.1121/1.4764511>.