

Práctica 2 – Fundamentos de Procesado de Señal

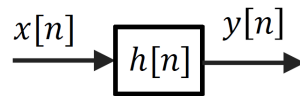
Procesamiento de Señales Biomédicas
Universidad de Granada

Joaquín T. Valderrama*

Curso 2024–2025

1. Convolución de dos señales

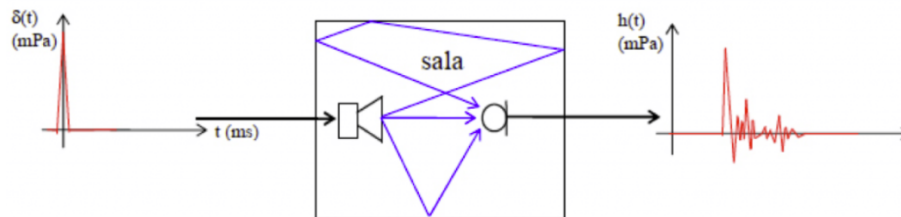
La convolución de dos señales permite estimar la salida de un sistema lineal e invariante en el tiempo (LTI, por sus siglas en inglés *linear and time-invariant*), conocida la señal de entrada y teniendo caracterizado el sistema LTI a partir de su respuesta al impulso.



En este ejercicio vamos a recrear cómo se escucharía una canción en un determinado escenario acústico, en particular, en el aula donde nos encontramos. Para realizar esta actividad, primero deberás caracterizar el sistema ($h[n]$: el aula) a través de su respuesta al impulso, para posteriormente estimar la señal de salida ($y[n]$: la canción recreada) a través de la convolución de la señal de entrada ($x[n]$: la canción original) con $h[n]$.

1.1. Caracterización del sistema $h[n]$

Un sistema LTI queda perfectamente caracterizado por su respuesta al impulso. Por tanto, considerando el aula como un sistema LTI, podemos caracterizar dicho sistema registrando su salida cuando introducimos un pulso unitario. Una técnica habitualmente utilizada para caracterizar la acústica de una habitación consiste en presentar un estímulo breve de alta intensidad ($\delta[n]$: explotar un globo o dar una palmada) y en registrar el sonido generado por el estímulo $y[n]$. De este modo, la señal registrada $y[n]$ corresponde con la respuesta al impulso de la habitación, es decir, $h(t)$. La figura adjunta ilustra este proceso.



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

Sigue los siguientes pasos para caracterizar el aula mediante su respuesta al impulso, es decir, para estimar $h(t)$.

1. Pide silencio en la habitación.
2. Pon a grabar la grabadora de tu teléfono móvil.
3. Da una palmada fuerte.
4. Detén la grabadora, y envíate a ti mismo por email el fichero de audio.
5. Abre el fichero de audio en MATLAB mediante la función *audioread*.
6. Almacena la señal de audio en la variable h y la frecuencia de muestreo en la variable fs .
7. Actualiza la variable h seleccionando el segmento relevante.
8. Genera una figura en formato .png mediante la función *print* que represente $h[n]$ en función del tiempo.
9. Guarda la señal procesada como fichero de audio en el archivo 'h_aula.wav' mediante la función *audiowrite*.

1.2. Recreación de una canción en el aula

Una vez caracterizado el aula a partir de su respuesta al impulso, estamos en condiciones de recrear cómo se escucharía una determinada señal de audio en dicho ambiente acústico. Para esto, crea un script de MATLAB que realice los siguientes pasos.

1. Carga el fichero 'Song.wav'. Almacena la canción en la variable x , y la frecuencia de muestreo en la variable fs .
2. Escucha la canción utilizando la función *soundsc*.
3. Carga el fichero 'h_aula.wav'. Almacena la señal en la variable h , y la frecuencia de muestreo en la variable fs_h .
4. En caso de que fs_h sea diferente a fs , remuestrea la variable h a la frecuencia fs utilizando la función *resample*. Este paso es importante para asegurarte que las señales x y h están muestreadas a la misma frecuencia.
5. Almacena en la variable y la convolución de x con h utilizando la función *conv*. Este proceso recrea cómo se escucharía la canción x en la habitación caracterizada por la respuesta al impulso h .
6. Compara cómo se escucha la canción recreada y , en comparación con la original x .
7. Repite el proceso para un ambiente acústico altamente reverberante (registrado en una cueva) utilizando el fichero 'h_cueva.wav'.

Contenido de la memoria

- Describe el proceso de caracterización acústica del aula mediante el registro de la respuesta al impulso.
- Presenta una figura que compare las $h[n]$ del aula y de la cueva en el dominio del tiempo. ¿Qué diferencias se perciben? ¿A qué se deben?
- Representa la señal de audio en función del tiempo.
- Describe el proceso de la convolución para recrear un sonido en un ambiente acústico.
- Describe la percepción del audio de la señal original, la señal recreada en el aula, y la señal recreada en una cueva.
- Incluye apéndices que incluyan los scripts de MATLAB utilizados.

2. Filtrado paso banda de una señal de electrocardiografía (ECG)

Familiarizarte con el proceso de filtrado digital de señales es fundamental para procesar adecuadamente señales de carácter biomédico. En esta sección, tomaremos una señal de 10 segundos de ECG de referencia, le añadiremos ruido de banda ancha, y estimaremos la mejora de calidad en términos de relación señal ruido (SNR, por sus siglas en inglés *signal-to-noise ratio*). El ECG de referencia se ha obtenido de la base de datos pública proporcionada por [Zheng et al. \(2022\)](#), disponible en PhysioNet ([Golderberger et al., 2000](#)).

2.1. Caracterización de una señal de ECG

1. Carga el fichero 'ECG_Example.mat' mediante la función *load*. Observa las variables *data* y *fs*.
2. Crea un vector de tiempos, y representa en la Figura 1 el ECG en función del tiempo. La escala de amplitud está en μV .
3. Crea un vector de frecuencias, y en la Figura 2 representa el módulo de la Transformada de Fourier del ECG en decibelios. Nota: $X_{dB} = 20 * \log_{10}(X_{lineal})$.
4. Identifica la banda de frecuencias en la que el ECG presenta su mayor parte de energía. Un criterio razonable es considerar un umbral de 30 dB.

2.2. Contaminación de un ECG con ruido blanco

Una vez caracterizada nuestra señal de interés tanto en el dominio del tiempo como de la frecuencia, añadiremos ruido aditivo blanco Gaussiano (con espectro plano en toda la banda de frecuencias) a una determinada SNR.

1. Almacena en la variable *Noise* un segmento de ruido aditivo blanco Gaussiano de la misma dimensión que la señal ECG, generado mediante la función *randn*.
2. Escala la amplitud del segmento de ruido para que tenga 10 dB menos de SNR que la señal ECG. Puedes calcular la energía de un vector x mediante su varianza $var(x)$, siendo su valor en decibelios: $10 * \log_{10}(var(x))$; o de manera análoga, también puedes utilizar la desviación estándar: $20 * \log_{10}(std(x))$.
3. Verifica que el escalado es correcto confirmando que $20 * \log_{10} \frac{std(data)}{std(Noise)} = +10$, es decir, que la energía del ECG es 10 dB superior a la energía del ruido.
4. En la Figura 1, sobre el ECG anteriormente dibujado, representa el segmento de ruido escalado. Puedes utilizar *hold on* para representar una nueva función en una gráfica sin borrar las representaciones ya existentes.
5. Del mismo modo, representa en la Figura 2 el módulo del espectro de amplitud del segmento de ruido en decibelios, sin borrar el espectro del ECG.
6. Almacena en la variable *ECG_Noise* la suma del ECG original junto con el segmento de ruido escalado.
7. En dos nuevas figuras (Figuras 3 y 4), representa el ECG contaminado con ruido en el dominio del tiempo y de la frecuencia.

2.3. Filtrado del ECG contaminado por ruido

1. Estudia la ayuda de MATLAB sobre la función *butter*, y diseña un filtro Butterworth de orden 2 paso bajo, con una frecuencia de corte apropiada para preservar las componentes más relevantes de la señal de interés (ver paso 2.1.4).

2. Filtra el ECG contaminado con ruido utilizando el filtro diseñado, y almacena la señal filtrada en la variable *ECG_filtered*. Para filtrar la señal, utiliza la función *filtfilt*. Estudia lo que hace esta función mediante la ayuda de MATLAB, y describe en la memoria la diferencia de utilizar esta función frente a la función *filter*.
3. Representa en la Figura 3 el ECG filtrado, y en la Figura 4, el espectro de la señal filtrada, en ambos casos sin borrar las representaciones anteriores.
4. Calcula la mejora de calidad entre el ECG contaminado por ruido y el ECG filtrado como la diferencia de energía entre ambas señales.

Contenido de la memoria

1. Representa el ECG en el tiempo y en la frecuencia, identificando la banda de frecuencias con mayor contribución a la señal de interés.
2. Describe el proceso de estimación de la calidad de una señal en términos de SNR.
3. Representa el ECG contaminado en el tiempo y la frecuencia, comparándolo con las representaciones de la señal original sin ruido.
4. ¿Qué ventajas ofrece la función *filtfilt* sobre *filter*? ¿Qué consideraciones hay que tener en cuenta sobre el orden del filtro al aplicar *filtfilt*?
5. ¿Qué efecto tiene el filtrado sobre el ruido de fondo? ¿Observas algún efecto del filtrado sobre la señal de interés?
6. Incluye apéndices con el código de MATLAB utilizado.

Criterios de evaluación de la memoria

- **Contenido.** ¿Es correcto? ¿Está completo?
- **Claridad.** ¿Está el contenido bien estructurado? ¿Fácil de seguir por alguien con formación técnica?
- **Redacción.** ¿Es la expresión escrita adecuada? ¿Utiliza lenguaje inclusivo?

Referencias

- Zheng, J., Guo, H., Chu, H. (2022). "A large scale 12-lead electrocardiogram database for arrhythmia study (version 1.0.0)," PhysioNet. <https://doi.org/10.13026/wgex-er52>
- Golderberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P.C., Mark, R., ..., Stanley, H.E. (2000). "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," Circulation 101, e215–e220. <https://doi.org/10.1161/01.CIR.101.23.e215>