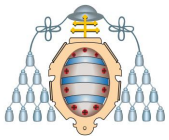


La Convolución de Señales

1. Objetivos

En esta práctica de laboratorio se busca afianzar los conocimientos que se han visto en las clases expositivas con respecto a la convolución. El objetivo de la práctica es doble, en primer lugar se persigue implementar una función en Matlab que calcule la convolución de dos señales y, además, muestre el proceso de manera animada. En segundo lugar, se mostrará de manera cualitativa el efecto de aplicar la convolución de una señal cualquiera con una función gaussiana, dado que este es un efecto que aparece a menudo en muchos problemas físicos.

2. Convolución de señales aperiódicas

En este primer apartado, el objetivo es implementar una rutina en Matlab que realice la convolución de dos funciones definidas en Matlab. Para tal fin, se describen en primer lugar las herramientas que le serán de utilidad y el pseudocódigo que le permitirá realizar la implementación.

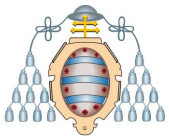
2.1. Herramientas

Para desarrollar la práctica necesitaremos de diferentes herramientas con las que puede que no esté familiarizado. En primer lugar, la convolución de señales de variable continua viene dada por una integral, que en general no tiene solución analítica y, por ello, debe evaluarse de manera numérica. Si bien una descripción detallada del cálculo de integrales numéricas queda fuera de los objetivos de esta asignatura, el lector debe conocer que una función $f(x)$ evaluada en un vector `f` en los puntos del vector `x`, puede calcularse mediante el comando `trapz(x,f)`.

Cuando se realicen este tipo de integrales numéricas debe tenerse en cuenta que se están realizando dos aproximaciones: i) sólo se dispone de información en un conjunto discreto de puntos y no en un continuo; ii) incluso aunque la integral sea impropia, la señal sólo puede ser evaluada en un rango finito de valores $[a, b]$, típicamente correspondientes a los puntos `a=x(1)` y `b=x(end)`. La discretización de la señal y el posible truncamiento del intervalo de integración resultarán en *errores numéricos*. Sin embargo, estos errores pueden ser asumibles siempre y cuando se usen suficientes puntos para que el modelado de la señal sea preciso y el intervalo abarque la mayor parte de la energía de la señal.

$\int_{-\infty}^{\infty} f(x)dx \cong \int_{a}^{b} f(x)dx \Rightarrow$

```
1 x = linspace(a,b,N); %N is the number of points
2 integral = trapz(x,f);
```



La segunda de las herramientas son las *anonymous functions*. A efectos prácticos funcionan de manera similar a las funciones que puede definir en un fichero *.m pero con una definición que se puede incluir a lo largo de un fichero. De este modo, se puede evaluar la función en un vector de puntos de la manera habitual con `f(x)`. Su principal ventaja es que se pueden pasar como argumentos a otras funciones. A continuación se muestra un ejemplo con una señal a trozos:

$$f(x) = \begin{cases} 1 & x < -1 \\ x^2 & -1 \leq x \leq 1 \\ 2 & x > 1 \end{cases} \Rightarrow$$

```
f = @(x) 1      * (x<-1)           + ...
         x.^2 .* (x>=-1 & x<=1)    + ...
         2      * (x>1);
```

Observe como el uso de operadores lógicos (`<`, `>`, `&`, etc.) nos permite definir con facilidad esta función a trozos.

Finalmente, la tercera herramienta que nos será de utilidad está relacionada con la posibilidad de realizar animaciones en Matlab. Al igual que sucede con la salida por pantalla cuando se crean programas en consola con C u otros lenguajes, Matlab, tras dibujar con el comando plot, no refresca la pantalla automáticamente sino que es necesario forzar la actualización. Si bien este efecto es inapreciable cuando se dibuja una única gráfica, es necesario cuando se realiza una animación. El comando que fuerza este refresco, jugando un papel similar a *flush* en otros lenguajes, es:

```
>> drawnow;
```

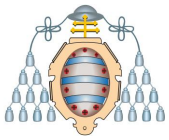
El lector interesado puede consultar la ayuda de este comando si desea utilizar las opciones más avanzadas.

2.2. Pseudocódigo de convolución

El pseudocódigo que realiza la convolución se muestra en el Algoritmo 1. Tenga en cuenta que cada línea de este pseudocódigo se traduce, en general, en una o dos instrucciones en Matlab, lo cual debería servirle de orientación para comprobar si se está complicando más de la cuenta.

Además de esta función, se recomienda que genere un segundo fichero desde el cual se llamará a la función. En dicho fichero deberá definir los vectores que contengan las muestras de x y x', así como las *anonymous functions* correspondientes a $f(x)$ y $g(x)$.

> **!** En esta sesión se recomienda que fije como intervalo de integración $x' \in [-10, 10]$ y como intervalo de observación $x \in [-3, 3]$. Puede utilizar como paso $\Delta x' = 0{,}01$ para el paso al muestrear x' y $\Delta x = 0{,}1$ para el paso al muestrear el dominio de observación x.



input : Handle a anonymous function `f`
input : Handle a anonymous function `g`
input : Vector `x` con los valores de x en lo cuales se evalúa $c(x)$
input : Vector `x_` con los valores de x' en los cuales se calcula la integral de convolución
output: Valores de $c(x)$ evaluados en `x`

`f_evaluated` ← Valores de `f` en `x_`;
Inicializa `c` con ceros `%c = zeros(1, length(x));`
foreach *elemento `x0` del vector `x`* **do**
    `g_evaluated` ← Valores de `g` en `x0-x_`;
    `c(x0)` ← Resultado de la integral de `f_evaluated.*g_evaluated` along `x_`;
    Dibujar $f(x')$ y $g(x_0 - x')$;
    Dibujar los datos disponibles de $c(x)$;
    `drawnow`;
end

Algoritmo 1: Pseudocódigo para la convolución de señales aperiódicas.

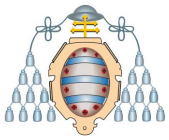
TAREA

1. Implemente una función en Matlab para calcular la convolución entre dos funciones siguiendo el Algoritmo 1.
2. Haciendo uso de la función anterior, calcule la convolución de las señales $f(x) = \Pi(\frac{x}{2})$ y $g(x) = e^{-x}u(x)$.
3. Compare el resultado anterior con la solución analítica:

$$c(x) = \Pi(\frac{x}{2}) * \left[e^{-x}u(x)\right] = \begin{cases} 0 & x \leq -1 \\ 1 - e^{-x-1} & -1 \leq x \leq 1 \\ 2e^{-x}\sinh 1 & x \geq 1 \end{cases}$$

OPCIONAL

1. Haciendo uso del código de la anterior tarea, calcule numéricamente y represente la señal $c(x) = f(x) * g(x)$ donde $f(x) = x\Pi(\frac{x}{2})$ y $g(x) = e^{-x}$. Puede utilizarse los valores de intervalos y pasos de la tarea anterior.
2. Compare el resultado anterior con el resultado analítico $c(x) = 2e^{-(x+1)}$. ¿Observa alguna diferencia? ¿A qué cree que pueden deberse estas discrepancias?



OPCIONAL

1. Realice una función que realice la convolución de dos señales periódicas de variable continua. Observe que la definición de la convolución es idéntica al caso aperiódico y que tan sólo es necesario cambiar los límites de integración.
2. Realice una función que realice la convolución de dos señales aperiódicas de variable discreta. Observe que la definición de la convolución es idéntica al caso de variable discreta pero que en lugar de integrar (función `trapz`) debe sumar (función `sum`)

3. Efectos de la convolución

3.1. Efecto sobre señales unidimensionales

Para poder comprender la operación que realiza la convolución, es importante observar que para cada punto x_0, el valor de la convolución viene dado por:

$$c(x_0) = \int_{-\infty}^{\infty} f(x')g(x_0 - x')dx'$$

la operación anterior puede verse como un promediado de los valores de la señal $f(x')$ en torno al punto x_0, usando como pesos los valores de la señal $g(-x')$ convenientemente desplazada al punto x_0.

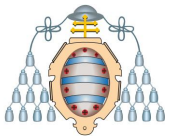
TAREA

1. Calcule la convolución entre un pulso $f(x) = \Pi(\frac{x}{2})$ y una función gaussiana $g(x) = e^{-\frac{x^2}{2\sigma}}$ con $\sigma = 0{,}04$.
2. Represente, bien con lápiz y papel bien con Matlab, las funciones $f(x')$, $g(x_0 - x')$ y $f(x')g(x_0 - x')$ y reflexione sobre la operación de promediado que se hace.
3. Compare la forma del pulso original y del pulso tras realizar la convolución. Señale las diferencias que observa.

3.2. Efecto sobre señales bidimensionales

El efecto anterior se puede observar en muchas señales de uso cotidiano como por ejemplo fotografías. Si bien el estudio de señales bidimensionales queda fuera de los objetivos de la asignatura, a continuación se muestra un pequeño código con el antes y el después de una imagen a la cual se aplica una convolución con una gaussiana para que pueda observar, de manera cualitativa, el impacto de esta operación. Esta operación es realizada de manera habitual por programas de retoque fotográfico para diversos fines (p.ej., Photoshop con su comando de "desenfoque gaussiano")

```
1 I = imread('cameraman.tif'); %Carga una imagen predefinida de Matlab
2 figure; imshow(I); title('Original image'); %Muestra la imagen original
3 PSF = fspecial('gaussian',7,10);
```



```
4 I_blurred = conv2(double(I),double(PSF)); %Realza una convolución 2D (los ...
      valores se convierten de unit8 a double)
5 figure; imshow(uint8(I_blurred)); title('Blurred image'); %Muestra la ...
      imagen tras la convolución (requiere convertir a unit8)
```

TAREA

Ejecute el código anterior (requiere la *Image Processing Toolbox*) y compruebe las diferencias entre la imagen normal y la imagen tras aplicar una convolución con una gaussiana.