

# Entrega de la Práctica 1: Señales en MATLAB y SIMULINK

## CONTENIDOS OBLIGATORIOS (8 puntos)

El alumno deberá subir en la actividad de Moodle un fichero comprimido (formatos .rar o .zip) que contenga los siguientes ficheros .m de MATLAB:

- a) Seis funciones que permitan generar las señales de entrada discretas del enunciado de la práctica: escalón, rampa, pulso, pulso triangular, seno cardinal (*sinc*) e impulso. Sus cabeceras deben coincidir exactamente en nombre, argumentos y orden de los mismos, con las siguientes<sup>1</sup>:

- o `function [u,n] = escalon(ni, nf, n0, A)`
- o `function [d,n] = impulso(ni, nf, n0, A)`
- o `function [p,n] = pulso(ni, nf, n0, n1, A)`
- o `function [r,n] = rampa(ni, nf, n0, m)`
- o `function [sc,n] = senoc(N, p, A)`
- o `function [tri,n] = triangular(ni, nf, n0, n1, m)`

Las funciones **NO** deben realizar representaciones gráficas, sólo calcular las señales.

- b) Cuatro funciones para realizar operaciones con **señales discretas** (el eje temporal se indica mediante la posición de  $n=0$ )<sup>2</sup>:

- o `function [xdes,nout] = desplazamiento(xin, n0, a)`
- o `function [xcom,nout] = compresion(xin, n0, a)`
- o `function [xdil,nout] = dilatacion(xin, n0, a)`
- o `function [xref,nout] = reflexion(xin, n0)`

Se debe garantizar que los tiempos proporcionados se corresponden con una señal muestreada o dar un error informando de que no es así.

- c) Una de las funciones que aparecen a continuación (trenes de ondas), según la siguiente asignación:

- o **Grupo 1:** Jaime y Luis Miguel:
  - `function [x,n] = pulsossimetrico(A, w, n)` <sup>3</sup>

El tren de pulsos debe ser simétrico respecto de  $n=0$ .

<sup>1</sup> Los nombres de los argumentos de entrada representan

- *nf*: muestra final de la señal
- *ni*: muestra inicial
- *n0*: muestra de comienzo de escalón/rampa/pulso/triangulo/impulso
- *n1*: muestra de final del pulso/señal triangular
- *A*: amplitud de la señal
- *m*: pendiente de la rampa
- *N*: periodo del seno cardinal
- *p*: número de periodos del seno cardinal que se deben representar

<sup>2</sup> Los nombres de los argumentos de entrada representan

- *xin*: señal discretizada
- *x0*: indice de *xin* que corresponde al instante 0
- *a*: Valor de la amplitud de desplazamiento/compresión/dilatación

<sup>3</sup> Los nombres de los argumentos de entrada de la señal representan

- *A*: amplitud del pulso
- *w*: ancho del pulso
- *n*: número de pulsos

- Grupo 2: Elena y Francisco José:
  - `function [x,n] = dientes(A, w, n)`  
La media de la señal DEBE ser cero.
- Grupo 3: Rafael y Ángel:
  - `function [x,n] = rectificadacompleta(A, w, n)`  
Señal senoidal rectificada de onda completa.
- Grupo 4: Jesús y Carlos:
  - `function [x,n] = pulsos(A, w, n)`  
Tren de pulsos de simetría impar (el que aparece en las diapositivas).
- Grupo 5: Adrián y Rodrigo:
  - `function [x,n] = rectificadamedia(A, w, n)`  
Señal senoidal rectificada de onda completa.
- d) La función de convolución de dos señales entre dos índices dados:
  - `function [cn] = convolucion(x, h, n0, n1)`  
Suponer que ambas señales empiezan en  $n=0$ .
- e) Un *fichero de comandos* que ejecute un ejemplo de cada una de las funciones de los apartados anteriores y muestre los resultados gráficamente. En aquellos casos donde sea necesario representar más de un gráfico se recomienda el empleo de la función `subplot` de MATLAB.

## CONTENIDOS OPCIONALES (2 puntos)

De manera opcional el alumno entregará:

- a) Demostrar con un ejemplo que las operaciones sobre la variable independiente no son conmutativas. Utilizar para ello las funciones del apartado b) de los contenidos obligatorios.
- b) El resto de trenes de onda del apartado c)
- c) Mejorar el algoritmo de la convolución para que calcule de forma automática los valores no triviales (no haga multiplicaciones de ceros por ceros) de la salida. En este caso, la cabecera será la siguiente:
  - `function [cn, cn0] = convolucion_opt(x, h, x0, h0)`  
Los argumentos de entrada serán las señales  $x$  y  $h$ . y los índices de la posición  $n=0$  de cada secuencia,  $x0$  y  $h0$ . Tendremos dos argumentos de salida: la convolución con los valores no triviales,  $cn$ , y el índice del elemento cero de la convolución,  $cn0$ .
- d) Auralización del fichero de sonido `bach.mp3` mediante la respuesta a impulso de una catedral recogida en `cathIR.wav`. El sonido resultante se debe guardar en formato `mp3`.
- e) Función que calcule un sonido sintético a partir de una secuencia aleatoria de  $M$  puntos (timbre del sonido) mediante el algoritmo de Karplus-Strong:
  - `function [y] = KS(M, alpha, N)`
    - $M$  número de puntos de la secuencia finita aleatoria.
    - $\alpha$  es la atenuación de la señal entre un periodo y el siguiente.
    - $N$  es el número de periodos que queremos que se repita la secuencia.
- f) Generar un sonido de timbre aleatorio de 2 segundos mediante el algoritmo de Karplus-Strong y guardarlo como `KS.mp3`.

## FECHA DE ENTREGA

La fecha límite de la entrega de los **contenidos obligatorios** será el **viernes 30 de septiembre a las 11:30h**. Cada día de retraso en la entrega penalizará un punto de la nota.

Los contenidos opcionales se pueden entregar hasta el día del examen final de la asignatura.