

Señales y Sistemas

Práctica 2

Señales en tiempo continuo

ÍNDICE

1. Introducción teórica
2. Señales básicas
 - 2.1. Escalón
 - 2.2. Delta de Dirac
 - 2.3. Pulso cuadrado
 - 2.4. Pulso triangular
3. Transformaciones
 - 3.1. Simetría
 - 3.2. Desplazamiento

1. Introducción teórica

La telecomunicación consiste en la transmisión de información a distancia. Esta transmisión de información se hace a través de señales que viajan por un canal de transmisión. A lo largo de esta práctica se pretende que el alumno analice los tipos de señales básicas estudiados en teoría y que se utilizan para transmitir información. Las señales deterministas se describen mediante funciones matemáticas dependiente de la variable t , $x(t)$.

Se pretende por tanto, hacer un estudio de estas señales **analógicas** en el *dominio del tiempo* utilizando la herramienta matemática MATLAB; sin embargo MATLAB utiliza valores discretos para su representación (un PC trabaja siempre con señales **digitales** y no analógicas), de manera que al hacer una representación gráfica (como hemos estudiado en la práctica 1) lo que hace realmente es calcular los valores de la función que va a representar para unos instantes discretos de tiempo (no para un eje continuo de tiempos que es lo que caracteriza a las señales analógicas). A pesar de ello, si elegimos los incrementos de tiempo suficientemente pequeños y en la representación se unen los valores de puntos consecutivos mediante líneas, tendremos la sensación de estar representando una señal continua; si por el contrario elegimos incrementos del eje temporal mayores, la representación que obtendremos no será adecuada. Como podemos suponer, la elección de este incremento temporal será importante para la correcta representación de las señales en general.

A lo largo de esta práctica se pretende que el alumno pueda afianzar los conocimientos adquiridos en las clases teóricas de la asignatura.

2. Señales básicas

Se trata de programar un conjunto de funciones en MATLAB que generen las señales básicas definidas a continuación. El parámetro de entrada en todas ellas será un vector t que define el intervalo de tiempos a considerar. Cada una de las funciones debe devolver un vector cuyas componentes sean los valores de la función para diferentes tiempos. A continuación **se representarán** estas señales para comprobar que se ha obtenido un resultado correcto.

2.1. Escalón

Programa una función para simular la representación de un escalón unidad $x(t) = u(t)$. A la función le pasamos como parámetro el vector de tiempos y nos devuelve en una variable vector el valor de dicha función para cada tiempo. La llamada será:

```
>>x=escalon(t);
```

Dibuje la señal en Matlab, comprobando con un ejemplo que el resultado obtenido es correcto.

2.2. Delta de Dirac

Repita el apartado anterior con una señal delta de Dirac $x(t) = A \delta(t)$. En este caso la llamada a la función será:

```
>>x=delta(t,A);
```

Como, lógicamente, no podemos obtener un valor infinito en $t = 0$, se escogerá el

valor de la amplitud A para la representación en el instante cero. Dibuje la señal en Matlab, comprobando con un ejemplo el resultado obtenido.

2.3. Pulso cuadrado

Programe ahora la función pulso cuadrado. En este caso los parámetros de entrada serán el vector de tiempos, la **anchura total** del pulso τ y su **amplitud** A . La llamada a la función será:

```
>>x=pulsocua(t,tau,A);
```

Dibuje la señal en Matlab, comprobando con un ejemplo el resultado obtenido.

2.4. Pulso triangular

Repita el apartado anterior para un pulso triangular. En este caso los parámetros de entrada serán el vector de tiempos, la **mitad del ancho** del pulso τ y su **amplitud** A . La llamada a la función será:

```
>>x=pulsotri(t,tau,A);
```

Dibuje la señal en Matlab, comprobando con un ejemplo el resultado obtenido.

3. Transformaciones

3.1. Inversión temporal (simetría)

Programe una función que represente la señal invertida en el tiempo de la que se le pasa como parámetro. La llamada a la función será $y=\text{invertir}(x,t)$ donde,

x contendrá los valores de la señal que queremos invertir,
 t es el vector de tiempos, y debe corresponder a un intervalo simétrico respecto a $t=0$,
 y contendrá los valores de la señal invertida temporalmente, esto, es $y(t) = x(-t)$.

Se pide generar una señal e invertirla con la función programada. Después represente la señal original y la invertida superpuestas en el mismo intervalo de tiempo con diferente color, y utilice `legend` para identificar cada una de ellas con una leyenda.

3.2. Desplazamiento temporal

Programe una función que desplace una señal que se le pasa como parámetro. El valor del desplazamiento se debe pasar como un parámetro más a la función. Así, la llamada a la función será: $y=\text{desplazar}(x,t,t_0)$ donde

x contendrá los valores de la señal que queremos invertir,
 t es el vector de tiempos, y debe corresponder a un intervalo simétrico respecto a $t=0$,
 y contendrá los valores de la señal desplazada temporalmente, $y(t) = x(t - t_0)$.

Se pide generar una señal y su desplazada con la función programada. Después represente la señal original y la desplazada en ejes diferentes, es decir, no superpuestas, sino

en subventanas dentro de una única ventana gráfica. Utilice `title` para explicitar qué señal está dibujada en cada caso.