INTRODUCCIÓN A MATLAB: SÍNTESIS DE SEÑALES CONTINUAS, MUESTREO, ANÁLISIS MEDIANTE TRANSFORMADA DE FOURIER.

Prac1_1.m: En este fichero realizaremos una serie de tareas básicas para la síntesis y manejo de señales en MATLAB, así como introduciremos el análisis en frecuencia para poder representar nuestras señales en dicho dominio mediante el uso de la Transformada Rápida de Fourier (FFT).

1.- Señal continua senoidal.

En este primer apartado generamos una señal senoidal que, supuestamente, es continua. En MATLAB no podremos generar señales realmente continuas debido a que el tratamiento que hace de ellas es discreto en todo momento, es decir, todas las señales se tratan como vectores o matrices con diferentes posiciones donde se van guardando los valores de dicha señal cada cierto intervalo de tiempo. Dicho intervalo o incremento de tiempo de muestra a muestra lo podemos hacer tan pequeño como queramos, pero siempre será un tiempo discreto, no continuo.

En el ejemplo mostrado se define inicialmente la frecuencia de muestreo para la señal original, Fs (por defecto en 10 Hz en este caso) y la resolución de la señal, Ts, como la inversa de la frecuencia Fs (100 ms en este caso). También se definen un par de constantes: TMax, que indica el tiempo máximo de la señal a generar, y Frec_senal_1, que indicará la frecuencia que poseerá la primera señal senoidal a generar (0.21 Hz por defecto). Una vez definidas todas estas constantes, estamos en condiciones de definir el intervalo de representación en el tiempo para nuestra señal, indicado por la variable “t”, que oscilará entre 0 y (TMax-Ts), en incrementos de Ts tal y como hemos indicado anteriormente. A partir de ahí generamos la señal senoidal “senal”, y la representamos gráficamente.

➡ Prueba a cambiar los diferentes parámetros de Fs y Frec_senal_1 para ver y comprender cómo afectan al resultado que se observa en pantalla en la figura 1. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

2.- Tren de pulsos para muestreo.

Tras generar una primera señal “continua” (ya sabemos que no lo es, pero la tomaremos como la original), procedemos a realizar un muestreo sobre ella. Para ello definimos una nueva variable que será la frecuencia de muestreo a utilizar (Fm, de 1 Hz por defecto), y el periodo de muestreo como la inversa de la anterior (Tm, 1 segundo por defecto).

Para simular el efecto del muestreo, crearemos una variable llamada “tren” que representará un tren de pulsos, es decir, una señal que valdrá 0 todo el rato salvo en los puntos donde se vaya a tomar el valor de la señal original, en cuyo caso valdrá 1. Representamos gráficamente dicha función en la figura 2, observando cómo queda.

➡ Prueba de nuevo a variar el parámetro Fs para ver el efecto que tiene sobre el propio tren de pulsos, ya que su “definición” o precisión será mayor o menor dependiendo de la misma. Varía también el valor de la frecuencia de muestreo Fm para observar el cambio en el tren de pulsos. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

3.- Señal muestreada.

En este apartado simplemente calculamos la señal muestreada a partir de multiplicar el tren de pulsos generado en la etapa anterior por la señal original, pero esta multiplicación se hará punto a punto (observar el comando ".*" de MATLAB, no "*" a secas) sobre todo el vector de datos completo. En la figura 3 representamos la señal original, el tren de pulsos y la señal ya muestreada (en rojo).

Prueba de nuevo a variar los parámetros Fs y Fm para ver el efecto que tienen sobre el tren de pulsos y la señal muestreada, de la misma manera que se hizo en el apartado anterior. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

4.- Análisis en frecuencia (Completo).

Una vez generadas las señales en el tiempo pasamos a calcular y representar gráficamente dichas señales pero en frecuencia, mediante el uso de la Transformada Rápida de Fourier (FFT, *Fast Fourier Transform*). Para ello debemos realizar el siguiente proceso:

- Calcular el número de muestras que tenemos (por defecto, 100) en la señal a representar, para lo que nos valemos de la función de MATLAB "length", y guardamos el resultado en la variable "longitud".
- Calcular la FFT mediante la función de MATLAB con el mismo nombre, dejando el resultado en la variable "senal_fft".
- Calculamos la magnitud y la fase de la transformada; la magnitud nos representará la potencia de la señal en cada punto de frecuencia, definida como el valor absoluto de cada uno de los números complejos que forman el resultado: Si el número es $a + jb$, el valor absoluto es $\sqrt{a^2 + b^2}$. La fase se corresponderá con el ángulo formado por la representación polar de dicho número complejo.
- Se genera un vector que contendrá los puntos de frecuencia que representa cada valor de la FFT, es decir, generamos los valores para el eje X de la representación gráfica de los valores en frecuencia, que tendrá tantos como indique "longitud", y lo guardamos en la variable "EjeX". Estos valores comienzan en la frecuencia 0 y terminan en la frecuencia de muestreo Fs (sin incluirla), por lo que irá desde "0" hasta "Fs-(Fs/longitud)").
- Representamos gráficamente la magnitud y la fase de la FFT en los 2 primeros "subplots" de la figura 4.

Observa que el espectro en frecuencia forma un "espejo" al superar la frecuencia Fs/2, es decir, se confirma lo visto en teoría en el teorema del muestreo. Puedes probar a variar nuevamente el parámetro Fs para ver el efecto que tiene sobre la representación gráfica de la FFT. ¿Qué diferencias observas al aumentar, por ejemplo, la Fs a 100? ¿Y si pones 1000? ¿Estás visualizando todo el tiempo el mismo análisis en frecuencia o hay algo que cambia? Explica lo que sucede a tu profesor. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

5.- Análisis en frecuencia (Zoom).

Se repite el proceso del apartado anterior, pero a la hora de representar gráficamente los resultados nos quedaremos solamente con la primera mitad del espectro, desde la frecuencia de 0 Hz hasta la Fs/2 Hz (inclusive), ya que el resto de valores hasta Fs son los mismos que los anteriores pero en forma de "espejo".

Para hacer esta representación el proceso será exactamente igual al apartado anterior, pero a la hora de hacer el comando “plot” indicaremos tanto para el eje X como para el Y que solamente queremos representar los valores desde “1” (frecuencia de 0 Hz) hasta “longitud/2+1” (frecuencia de Fs/2). El resultado aparece en los dos “subplots” inferiores de la figura 4.

Observa de nuevo el espectro en frecuencia representado y varía nuevamente el parámetro Fs para ver el efecto que tiene sobre la representación gráfica de la FFT. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

6.- Mezcla de 2 señales senoidales.

En este apartado se define una segunda señal senoidal con una frecuencia distinta a la definida en el apartado 1 de esta práctica, que vendrá indicada en la variable “Frec_senal_2” con valor 1 por defecto, y que sirve para general la “senal2”. A continuación se genera la suma de las señales “senal” y “senal2”, dejando el resultado en “senal_mezclada”. Finalmente, se representan todas ellas en 3 “subplots” la figura 5.

Prueba a cambiar los diferentes parámetros de Frec_senal_1 y Frec_senal_2 para observar los distintos resultados de sumar ondas senoidales de diferentes frecuencias. Modifica posteriormente el valor de Fs aumentándolo. ¿Qué diferencias observas en las señales? Coméntalo con tu profesor. Tras finalizar, restaura los valores originales para continuar con el siguiente apartado.

7.- Análisis en frecuencia de la señal mezclada.

Finalmente, realizamos una representación en frecuencia de la señal compuesta por la suma de las dos señales senoidales de la misma forma que se hizo en el apartado 5, es decir, representando la magnitud y la fase, pero solamente para la primera mitad del espectro desde 0 Hz hasta Fs/2 (inclusive).

Prueba a cambiar los diferentes parámetros de Frec_senal_1 y Frec_senal_2 para observar los cambios en frecuencia. Modifica posteriormente el valor de Fs aumentándolo. ¿Qué diferencias observas en las señales? Coméntalo con tu profesor. Tras finalizar, restaura los valores originales.

INTRODUCCIÓN A MATLAB: CARGA DE SEÑALES DE FICHEROS EXTERNOS; REPRESENTACIÓN EN TIEMPO Y EN FRECUENCIA DE UN ECG.

Prac1_2.m: En este fichero realizaremos la carga y representación, en el dominio del tiempo y de la frecuencia, de una señal de un electrocardiograma (ECG).

1.- Señal de ECG.

Realizamos la carga un fichero que contiene una señal de un ECG mediante el comando de MATLAB “load”, asignando dicha señal a la variable “ECG_1”. Posteriormente calculamos la longitud de la misma, creamos un intervalo de representación de esta señal en la variable “m”, con tantas muestras como posea la señal de entrada, y representamos gráficamente el ECG en la figura 1.

Observar que esta representación se realiza solamente en función del número de muestras introducidas, en el siguiente apartado se pondrá en función del tiempo realmente.

2.- Señal de ECG (en función del tiempo).

Para representar el ECG en función del tiempo debemos conocer la frecuencia de muestreo del mismo, que en este caso es de 500 Hz, introduciendo ese valor en la variable Fs de la cual derivamos el valor de Ts (periodo de muestreo) como inversa de la anterior. A continuación creamos un intervalo de representación en tiempo para nuestra señal, haciendo que “t” sea un vector con tantos puntos como la longitud del ECG, con incrementos de Ts (periodo de muestreo). El tiempo máximo de la señal vendrá dado por “(longitud-1)/Fs”.

Observar que en esta nueva representación sí se ha dibujado la señal del ECG en función del tiempo, indicando en el comando “plot” de la figura 2 (primer “subplot”) que en el eje X tendremos la variable “t” en lugar de “m”, como pasaba en la figura 1.

3.- Señal de ECG (en función del tiempo, sin DC, en mV).

En primer lugar, antes de procesar un ECG, procederemos a eliminar la componente de continua que posee para que así esté centrada en 0 mV. Para ello calculamos la señal “ECG_sin_DC” restando a la señal original la media de ella misma, calculada con la función “mean” del MATLAB. Representamos esta señal en otro “subplot” de la figura 2, indicando que la escala para el eje Y ahora está en milivoltios (mV).

Observar que la escala vertical de los 2 subplots de la figura 2 ha cambiado, ya que en el segundo caso está centrada en 0 mV.

4.- Análisis en frecuencia.

Finalmente, realizamos el cálculo de la Transformada de Fourier para toda la señal del ECG y la representamos gráficamente utilizando el proceso descrito en el fichero “Prac1_1.m”, visto anteriormente en la presente práctica. Indicar que representaremos en la figura 3 la primera mitad del espectro (la que contiene información única), y posteriormente, en las figuras 4 y 5, realizaremos un “zoom” a la parte que más nos interesa del ECG que son las frecuencias entre 0 y 60 Hz, y entre 0 y 20 Hz, respectivamente.

Observar cómo se puede apreciar con más detalle los picos importantes de la representación en frecuencia del ECG conforme más nos centramos en la zona típica de presencia para las ondas P, T y complejo QRS.