

# MATLAB: Taller 3

Prof. Mario Enrique Duarte González  
Señales de Tiempo Discreto (Grupo 2)

5 de abril de 2019

## Taller de filtrado

Estimados ingenieros, visiten el siguiente vínculo para mayor información sobre Matlab:  
<https://la.mathworks.com/help/matlab/getting-started-with-matlab.html>

Para entregar de manera individual, impreso por ambas caras en blanco y negro el día de su parcial Vale por 30% de la nota del segundo parcial. Imprimir con Figuras pequeñas y letra tamaño 8. Marque claramente el punto, no es necesario repetir en enunciado.

Para esta actividad considere el siguiente script y su resultado (Figura 1):

```
1 Ts=1/44100;  
2 Fs=1/Ts;  
3 t=0:Ts:3-(Ts);  
4 y=sin(2*pi*440*t); % 440=La.  
5 sound(y,Fs);  
6 pause(3)  
7 ynoise=y+(0.1).*randn(size(y));  
8 sound(ynoise,Fs);  
9 ynoise(randi(length(y),1,ceil(0.01*length(y))))=randi(3)-2; % toma un ...  
    0.01 del vector y lo reemplaza por valores fuera de lo normal ...  
    "outliers" [-1, 0, 1]  
10 pause(3)  
11 figure(1)  
12 subplot(2,1,1),plot(t(1:300),y(1:300)),title("La")  
13 subplot(2,1,2),plot(t(1:300),ynoise(1:300)),title("La + ruido. -ruido ...  
    de media 0 y desviacion estandar 0.1-")
```

**Actividad 1. Filtro media.** Una primera elección simple para filtrar una señal es generar una ventana y tomar un valor calculado a partir de la ventana, como el valor central de una muestra de valores. A partir de la señal ynoise (Figura 1), filtre usando diferentes tamaños de ventana promediadora hasta encontrar un tamaño para el cual crea usted se está removiendo de manera efectiva el ruido. Grafique, y compare usando la fft como estimación de la transformada de fourier de la señal continua.

**Entregar**

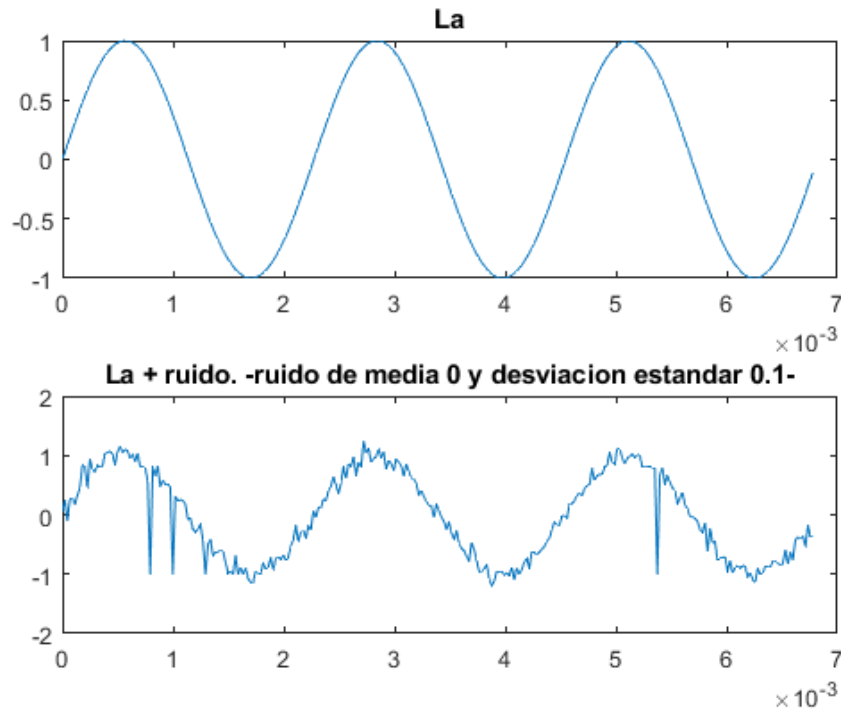


Figura 1: Figura generada por el script

1. Función que entrega el filtrado por media de tamaño  $N$  de un arreglo de entrada.
2. Gráficas en tiempo y magnitud de la frecuencia de las señales antes y después de ser filtradas.
3. Conclusiones.

**Actividad 2. Filtro mediana.** Averigüe que es la mediana y explique. repita el punto 1 usando una ventana de mediana

**Entregar**

1. Función que entrega el filtrado por mediana de tamaño  $N$  de un arreglo de entrada.
2. Gráficas en tiempo y magnitud de la frecuencia de las señales antes y después de ser filtradas.
3. Conclusiones.

**Actividad 3. Convolución.** Implemente el ejercicio del primer punto de este taller empleando una convolución.

**Entregar**

1. Función que entregue el filtrado usando convolución de un arreglo de entrada con un arreglo  $h$  arbitrario.

2. Gráficas en tiempo y magnitud de la frecuencia de las señales antes y después de ser filtradas.
3. Conclusiones comparando los resultados con los de la actividad 1, use las funciones tic toc para comparar tiempos.

**Actividad 4. Convolución + FFT.** Implemente el ejercicio del primer punto de este taller empleando una convolución pero utilizando solamente las funciones *fft* y *ifft*.

**Entregar**

1. Función que entregue el filtrado usando convolución y los algoritmos FFT de un arreglo de entrada con un arreglo *h* arbitrario.
2. Gráficas en tiempo y magnitud de la frecuencia de las señales antes y después de ser filtradas.
3. Conclusiones comparando los resultados con los de la actividad 1, use las funciones tic toc para comparar tiempos.

**Actividad 5. Derivadores.** Existen varias implementaciones de derivadores discretos:

- Referencia a: <https://www.dsprelated.com/showarticle/35.php>
- Referencia b: <https://www.dsprelated.com/showarticle/814.php>

De la referencia a, implemente a su gusto el derivador dado por la ecuación 5 a partir de convolución.

**Entregar**

1. Función diferenciadora, probada usando el arreglo *y* y *y<sub>noise</sub>*, derive y compruebe si la salida es parecida a:

$$2\pi \frac{440}{44100} \cos(2\pi 440t)$$

2. Gráficas en tiempo y frecuencia de las señales *y* y *y<sub>noise</sub>* antes y después de ser derivadas.
3. Conclusiones y breve explicación del diferenciador implementado.

**Actividad 6. Algoritmo FFT propio y el de Matlab.** Implemente un algoritmo propio para el cálculo de la FFT sin utilizar los comandos: `for` o `while` o `fft` o `ifft`. **Pista:** Observe que retorna el comando `exp(randi(6,3,3))`.

**Entregar**

1. Función con el nombre *mi\_fft* que realice la Transformada de Fourier Discreta (DFT) de una señal discreta arbitraria.
2. Ejemplos usados para verificar el correcto funcionamiento de su función.
3. Conclusiones comparando las funciones *mi\_fft* y *fft*, use las funciones tic toc para comparar tiempos.

**Actividad 7. Algoritmo IFFT.** Investigue y realice el cómputo de la IFFT (Inverse Fourier Transform) usando el algoritmo `fft` y, logicamente, sin utilizar la función `ifft`.

**Entregar**

1. Función con el nombre *mi\_ifft* que realice la Transformada de Fourier Discreta (DFT) de una señal discreta arbitraria usando la función `fft`.
2. Ejemplos usados para verificar el correcto funcionamiento de su función.
3. Ecuación matemática usada en la implementación y conclusiones. Explicar la ecuación.