
Filtros 1D en frecuencia

APELLIDOS, NOMBRE: QUEJO SEOANE, DANIEL

Objetivo

El objetivo de esta práctica es iniciarse en el empleo de técnicas de filtrado de señales en el dominio de la frecuencia.

La transformada discreta de Fourier (DFT) permite observar las componentes frecuencias de una señal. A partir de esa representación, es posible realizar el filtrado realizando la multiplicación de la DFT de la señal y la DFT del filtro.

EJERCICIO 1: DFT de señales con y sin ruido

En este ejercicio consideraremos la señal vista en la PRÁCTICA 1 definida como sigue:

$$x(n) = \cos \cos(2\pi f n) + \sum_{k=1}^{Na} \cos(2\pi(k+1)fn)$$

Además, necesitará generar ruido blanco gaussiano

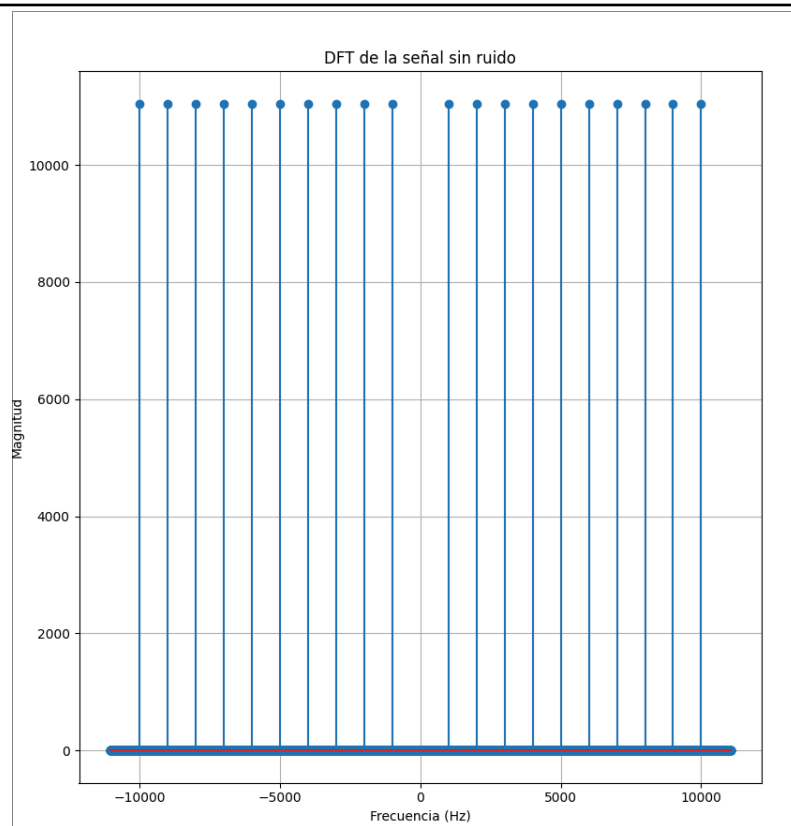
```
xn = np.random.normal(media, sigma, lx)
```

y sumarlo a la señal original

```
x = xcos + xn
```

Haga un programa que calcule y represente la DFT de la señal $x(n)$ SIN sumarle ruido utilizando como frecuencia fundamental $f = 1000$ Hz. **Tenga cuidado al crear el vector de tiempo ($f_s > 2 f$).**

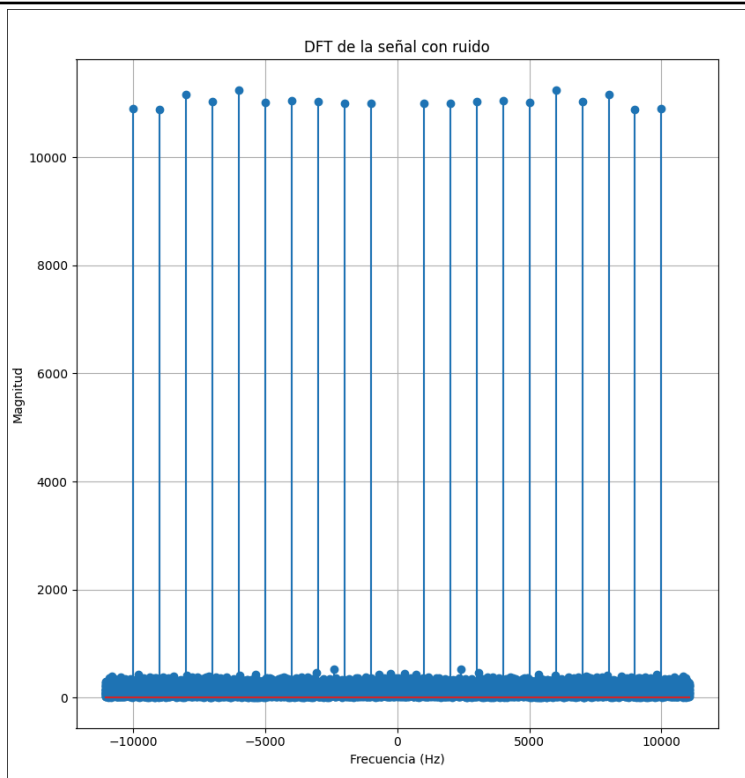
| |
|---------------------------|
| Copie la figura de la DFT |
|---------------------------|



¿Cuántas señales “delta” aparecen en frecuencia? ¿En qué frecuencias están?
Aparecen 10 ya que tengo configurado $N_a = 10$

Utilizando ese código, calcule y represente la DFT de la señal $x(n)$ CON ruido utilizando como frecuencia fundamental **$f = 1000$ Hz.**

Copie la figura de la DFT



¿Cuántas señales “delta” aparecen en frecuencia? ¿En qué frecuencias están?
 10, igual que antes. Alrededor del 0, de $-f_s/2$ a $f_s/2$, porque están centradas en el 0

¿Qué efecto tiene el ruido?
 El ruido aparece baja altura alrededor de las deltas

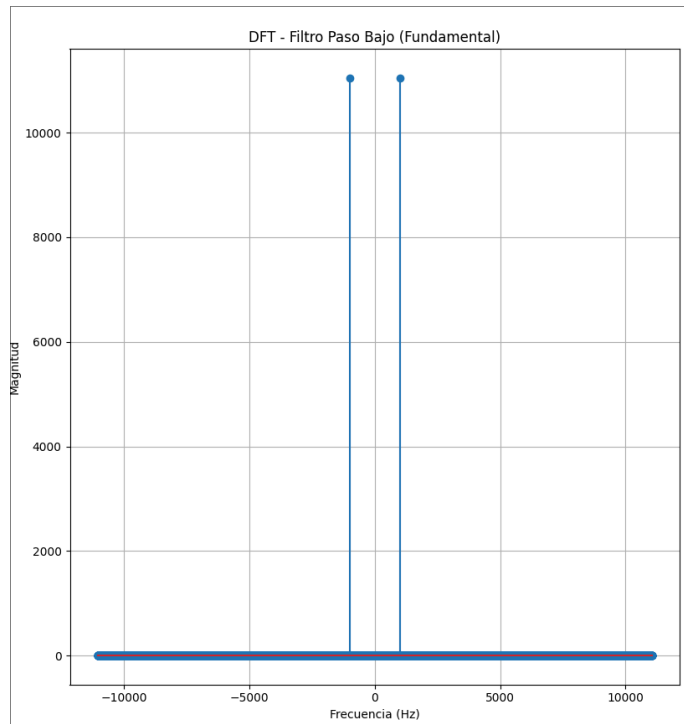
EJERCICIO 2: Filtrado en frecuencia

Para realizar el filtrado en frecuencia, puede emplearse un filtro ideal que tiene 1s en las frecuencias deseadas y 0s en el resto. El siguiente código permite generar un filtro paso bajo ideal que deja pasar solamente la componente de la frecuencia fundamental:

```
fcorde = 1500
HPB = (frec > -fcorde) & (frec < fcorde)
```

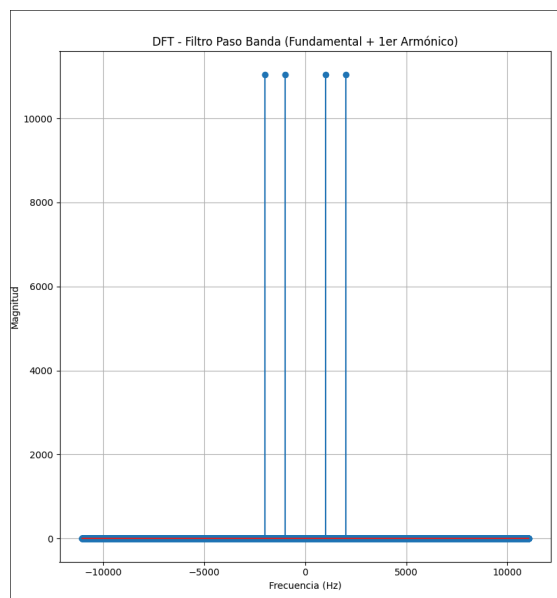
Realice el código para realizar el filtrado y la recuperación de la señal.
 Ejecute el código para dejar pasar solamente la señal con la frecuencia fundamental **f = 1000 Hz.**

Copie la figura de la DFT de la salida



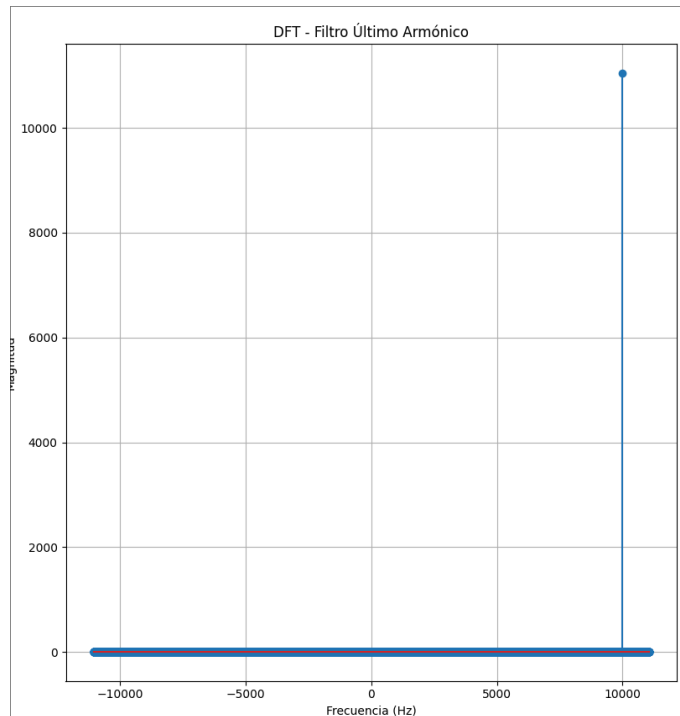
Cambie el filtro para que deje pasar el coseno de la frecuencia fundamental y el primer armónico.

Copie la figura de la DFT de la salida



Cambie el filtro para que deje pasar SOLAMENTE el último armónico. Tenga en cuenta que ahora no puede utilizar un filtro paso bajo.

Copie la figura de la DFT de la salida



EJERCICIO 3: Filtrado en frecuencia de audio

Partiendo del fichero `prog_filtrado_audio.py`, se realizará el filtrado de una señal de audio. Realice varias pruebas:

- Prueba con filtro paso bajo con enventanado. Ponga una frecuencia donde se pueda escuchar con bastante calidad y otra donde no.
- Prueba con filtro paso bajo sin enventanado. ¿Hay diferencia con lo anterior?
- Prueba con filtro paso alto con eventanado. ¿Qué sucede?

Puede probar a ajustar el solapamiento entre ventanas. Cuanto menor solapamientos, menora carga computacional tendrá.

Ponga las figuras más ilustrativas y una explicación de los resultados.

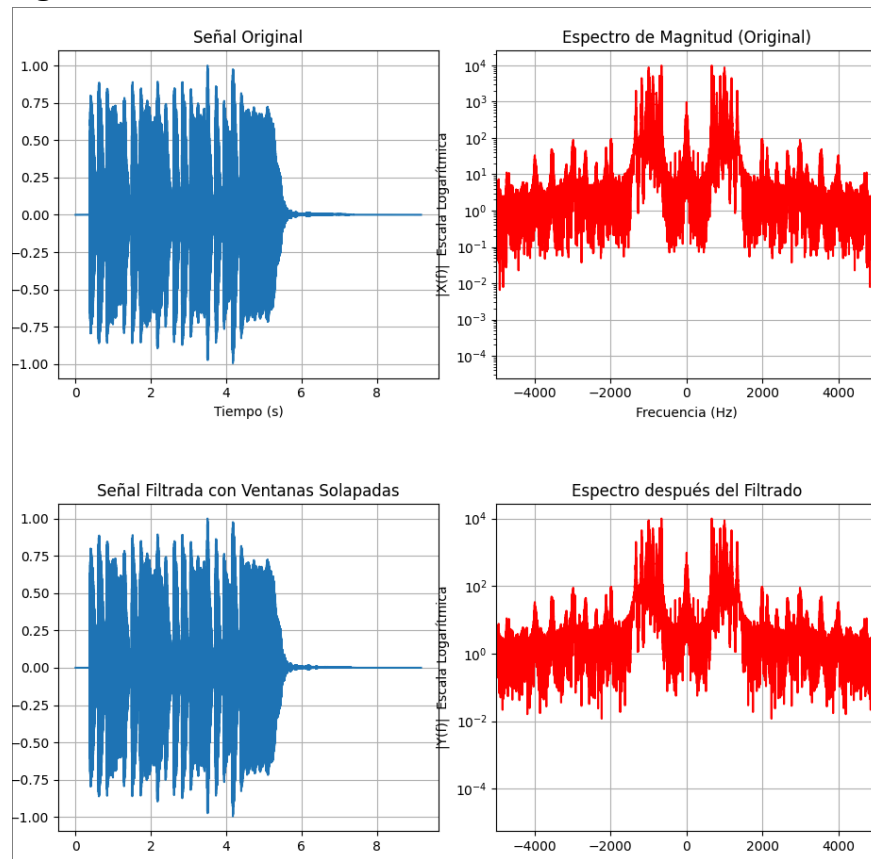
Prueba 1

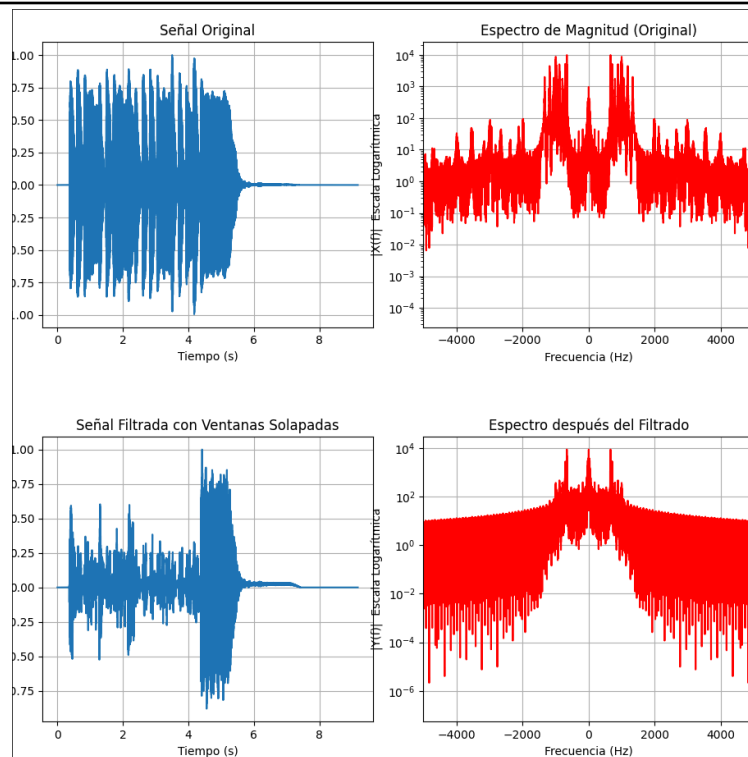
Parámetros:

$f_{\text{corte}} = 8000$,

$f_{\text{corte}} = 500$

Figuras:





Explicación:

cambiando la frecuencia de corte perdemos gran parte del contenido en frecuencia, empeorando la calidad.

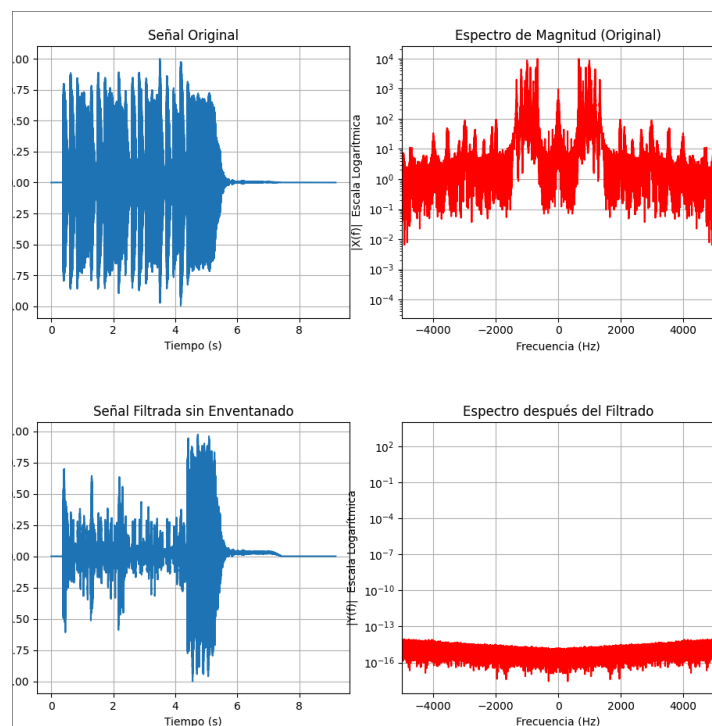
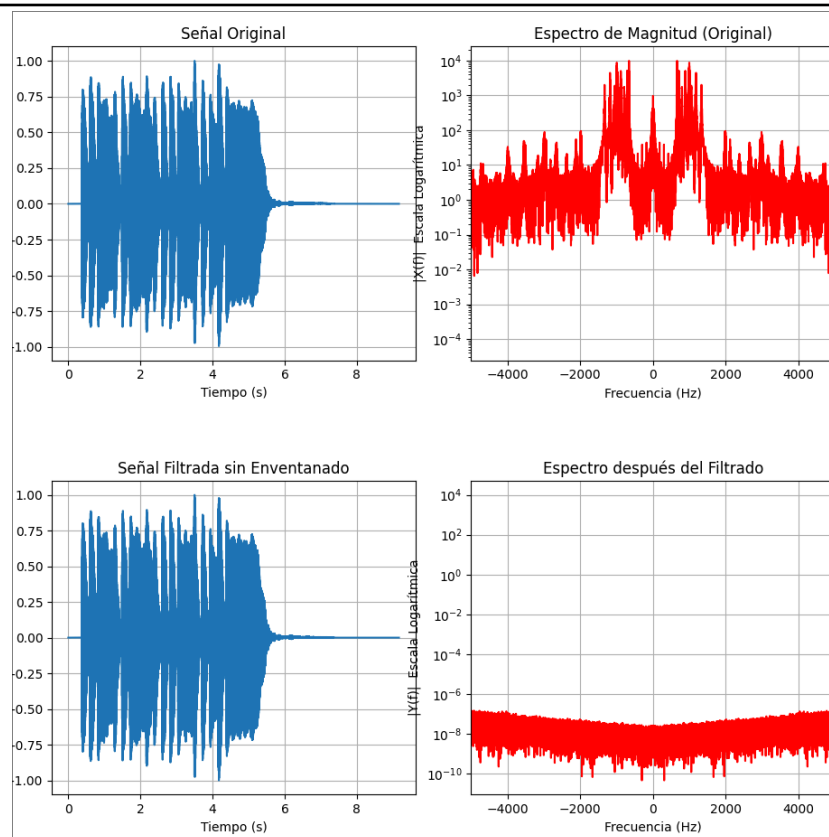
Prueba 2

Parámetros:

$f_{\text{corte}} = 4000$

$f_{\text{corte}} = 500$

Figuras:



Explicación:

al eliminar la ventana, bajar la frecuencia de corte sólo produce un bajada el el volumen del audio

Prueba 3

Parámetros:

fcorte = 500

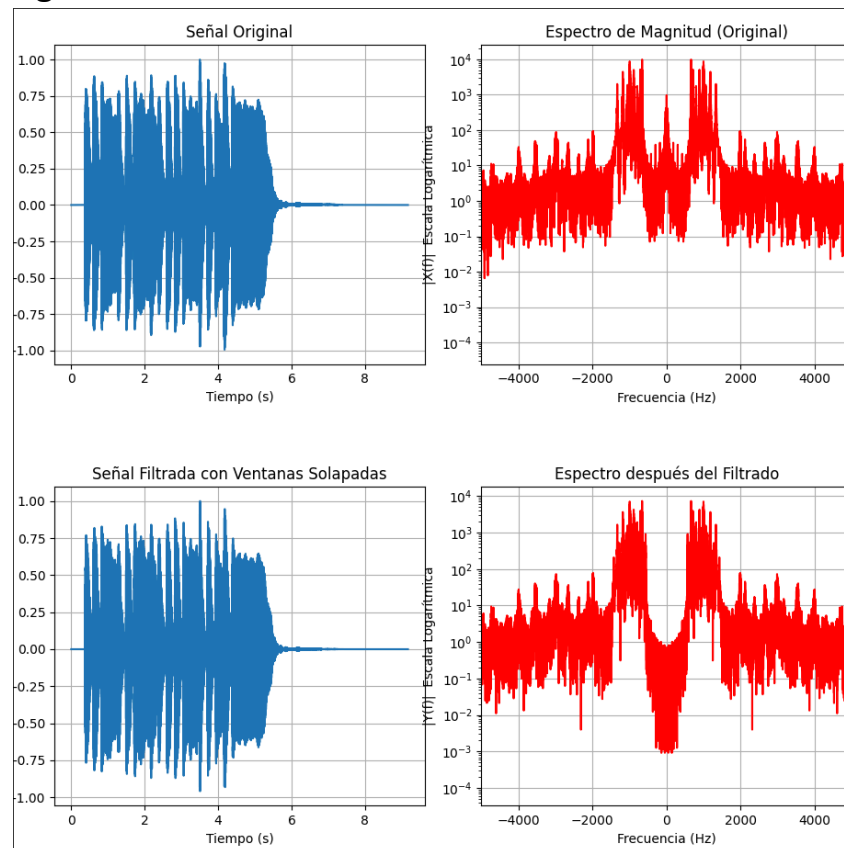
fcorte = 8000

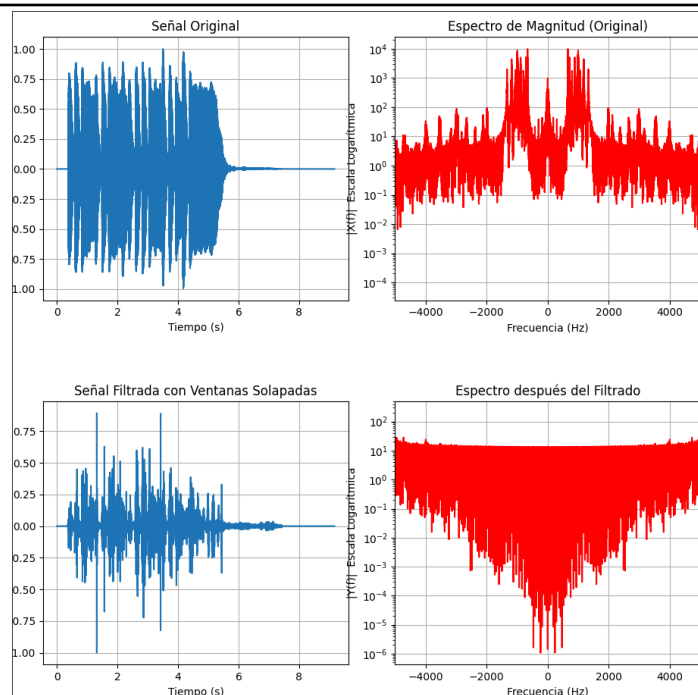
```
H_bajo = 1 / (1 + (freqs_shifted / fcorte)**(2 * norden))
```

```
H_alto = 1 - H_bajo
```

```
H = H_alto
```

Figuras:





En este caso sucede al contrario, con frecuencias altas antes tenemos mala calidad de audio, y con 500 de fcorde el audio permanece prácticamente intacto