

# Filtrado de señales ECG

Juan Casado Ballesteros

UAH - 19 Abril, 2020



Universidad  
de Alcalá

## Introducción

La presencia de ruido en las señales capturadas es una cuestión prácticamente inevitable. En el caso de la captura de los ECG conocemos algunas de las fuentes de ruido de modo que podremos tomar medidas muy concretas para limitarlo o eliminarlo. Existen tres fuentes de ruido principales a considerar en la captura de un ECG.

**Ruido procedente de la corriente alterna:** este ruido se caracteriza por estar entorno a los 50Hz de frecuencia o por estar entorno a los 60Hz. Estar entorno a una u otra depende de cual sea la frecuencia a la que la electricidad alterna con la que se alimente al aparato de medida se alimente. Visualizando el espectro de frecuencias o contando la cantidad de picos de ruido en un periodo suficientemente grande podremos averiguar de forma sencilla si nuestra señal contiene este tipo de ruido.

Para eliminar este ruido será necesario reducir la banda de frecuencias entorno a los 50Hz o los 60Hz según corresponda.

**Ruido procedente de las corrientes que surgen entre los electrodos y la piel:** para medir el ECG deberemos utilizar electrodos conectados en puntos estratégicos que nos permitan capturarlos de forma adecuada. Estos electrodos en contacto con la piel producen interferencias de baja frecuencia, por debajo de los 0.05 Hz. Adicionalmente la corriente continua (DC) producirá ruido situada en estas bajas frecuencias.

Una consecuencia de eliminar estas bajas frecuencias es el aplanamiento del ECG el cual pudo contener oscilaciones periódicas entorno a su línea base las cuales ahora desaparecerían.

Para eliminar este ruido debemos utilizar un filtro paso alto con un corte de frecuencia muy bajo, entorno a 1 Hz. Este tipo de filtros eliminarán las frecuencias bajas dejando solo las altas.

**Ruido de fondo o de alta frecuencia:** este es el ruido más habitual al realizar cualquier medición, pues puede surgir por muchos y muy diversos factores. Si conocemos la señal que estamos capturando y sabemos de antemano cual es el espectro de frecuencias en el que se debe manifestar podremos eliminarlo. En el caso de ECG sabemos que las frecuencias más altas se localizan en el QRS cuyas frecuencias están entre los 60 y 100 Hz.

Podremos ver este ruido como oscilaciones mínimas a lo largo de las curvas reconocibles del ECG que entorpecen su estudio, identificación y comprensión.

Podremos eliminarlo con un filtro pasa bajos que deje pasar las frecuencias por debajo de los 100Hz en adelante (podemos poner el límite en 110 o 120 Hz si queremos conservar más parte del QRS).

Mediante la aplicación de estos tres filtro para la eliminación de estos tres ruidos principales que afectan a los ECG obtendremos señales más claras y fáciles de interpretar.

## Visualización de la señal original

Medir digitalmente una señal, en este caso un ECG, consiste en asignar a intervalos regulares de tiempo una representación binaria proporcional a la magnitud registrada que se esté midiendo.

Partiendo de esta definición podremos cara caracterizar a nuestro aparato de medida por una frecuencia de muestreo y una cantidad de bits utilizada para almacenar la información. Esta cantidad de bits será la resolución binaria del aparato de medida. Si adicionalmente conocemos el valor máximo y mínimo de la magnitud que se está midiendo podremos conocer la resolución en las unidades de medida. La resolución es la menor cantidad de magnitud que el aparato puede medir.

Adicionalmente es común utilizar un amplificador de la señal, el cual multiplica su amplitud por un factor conocido.

Para un aparato de medida definido como:

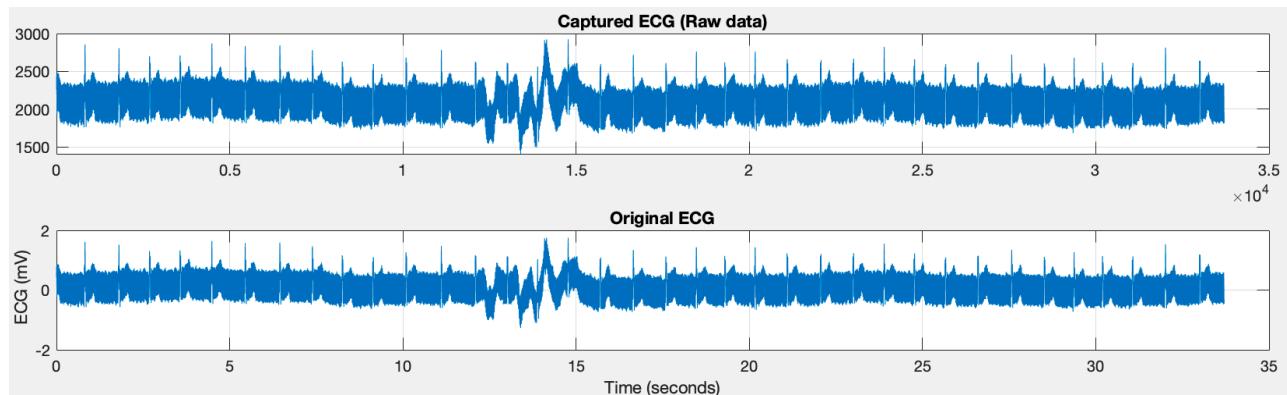
```
% Measuring device parameters
sampling_rate = 1000; % Hertz
sampling_period = 1/sampling_rate; % Seconds
adc_binary_resolution = 12; % Binary digits
v_min = -4.096; % Volts
v_max = 4.096; % Volts
gain = 1000; % How much the ECG was amplified
adc_resolution = (v_max-v_min)/(2^adc_binary_resolution); % Volts

>> adc_resolution = 0.0020V
```

Podremos transformar las señales que capture a señales en unidades de tiempo y voltaje estandarizadas mediante las siguientes transformaciones:

```
% The original ECG is reconstructed using the measuring device parameters
ecg = (ecg1ms(:,2).*adc_resolution+v_min)./gain.*1000; % in mV
% The ideal time stamps based on the sampling period are also reconstructed
elapsed_time = (ecg1ms(end,1) - ecg1ms(1,1))*sampling_period;
time_vector = 0:sampling_period:elapsed_time;
```

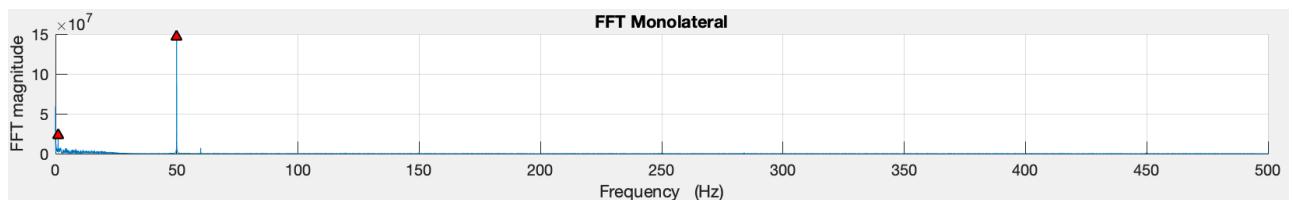
Podemos ver que estas transformaciones no afectan a la forma de la señal capturada, tan solo transforman sus unidades.



## Filtros FIR

Utilizaremos filtros FIR para eliminar las frecuencias donde se concentra el ruido en los ECG, tal y como explicamos en la introducción estas son las frecuencias muy bajas, inferiores a 1 Hz, las muy altas, superiores a 100 Hz y las que se sitúan entorno a los 50 o 60 Hz dependiendo de la corriente alterna con la que se alimente el aparato de medida.

Para analizar la señal del ECG según la estamos filtrando de modo que podamos comprobar que estamos actuando de forma adecuada es recomendable visualizarla en forma de FFT Monolateral. Esta visualización nos dice la magnitud de cada frecuencia presente en la señal.



Podemos ver que para nuestro ECG hay dos grandes picos de frecuencias, uno entorno a los 50 Hz y otro mucho menor entorno a 1 Hz. Adicionalmente podemos ver que de 100 a 500 Hz hay un continuo de pequeñas magnitudes de frecuencia que son ruido en su mayoría.

### Eliminación de ruido procedente de la corriente alterna

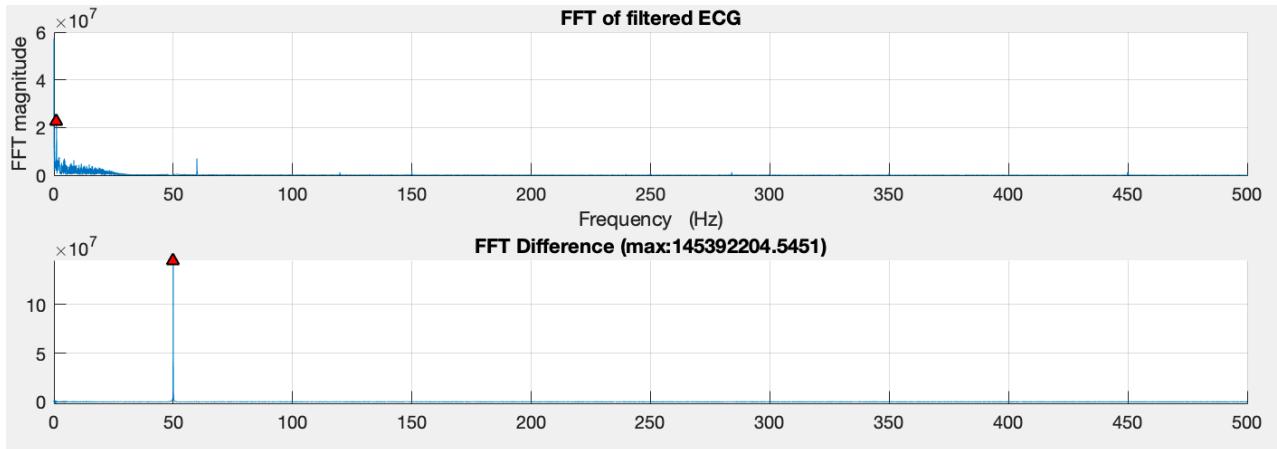
Crearemos un filtro FIR que pare la manda de frecuencias entre los 48 y los 51 Hz. La banda debe tener una anchura mínima para que el filtro pueda actuar correctamente, además esta debe cubrir el entorno de los 50Hz que es la frecuencia que deseamos eliminar.

```

%% STOP 50Hz FIR filter
half_sampling_rate = sampling_rate/2;
wl = 48/half_sampling_rate;
wh = 51/half_sampling_rate;
stop50 = fir1 (1536, [wl wh], 'stop');
ecgno50 = filter(stop50, 1, ecg);

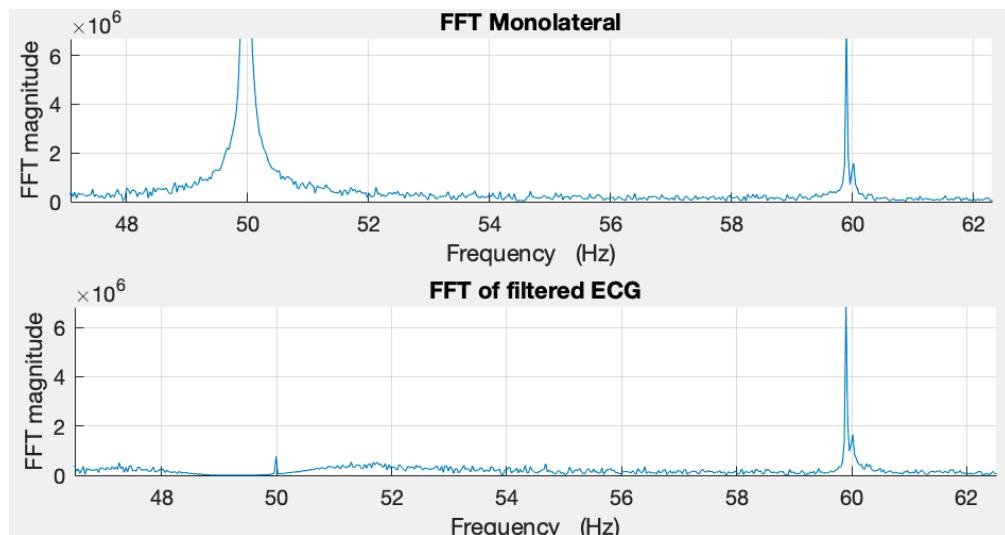
```

Podemos ver los resultados del filtro sobre la FFT de la señal. Mostramos primero la FFT de la señal actual en la que como vemos el pico de los 50Hz ha desaparecido. En la segunda gráfica hemos calculado la diferencia entre ambas FFT de modo que a partir de ella podemos ver las frecuencias afectadas y su magnitud.



La señal de 50Hz se ha reducido desde  $1.4616e+08$  hasta  $7.6819e+05$  lo que produce un factor de reducción de 190. Si eliminamos ese pico de reducción la media de reducción de la señal es de  $1.2249e+04$  valor que es extremadamente bajo.

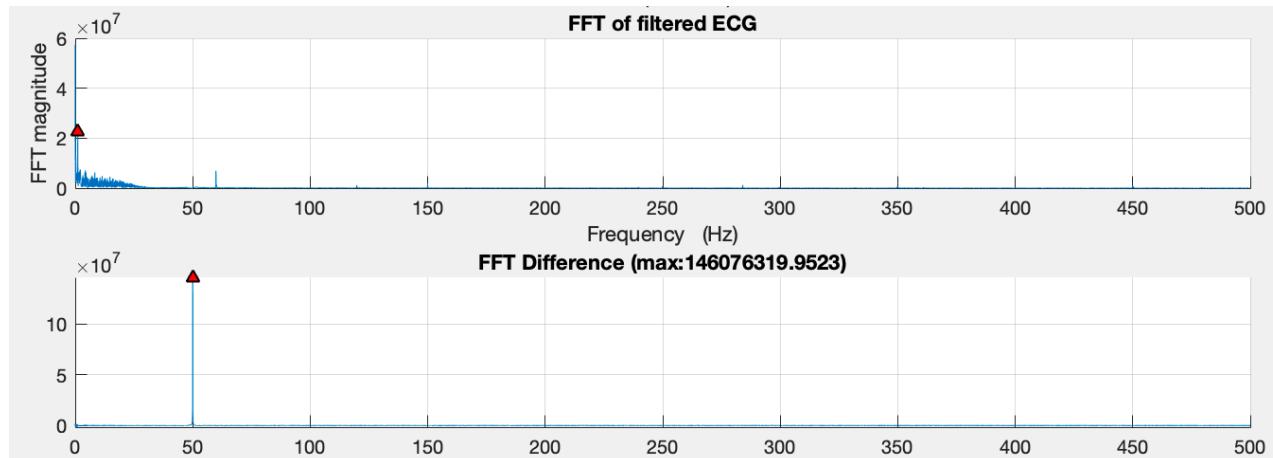
Podemos ver el entorno de los 50 Hz en más detalle para ver como estaba y como ha quedado tras la aplicación del filtro.



Vemos que donde antes había un gran pico de frecuencia ahora existe un valle y una mínima elevación. Podemos ver que en los 60Hz existe también una acumulación de frecuencias probablemente debida al QRS.

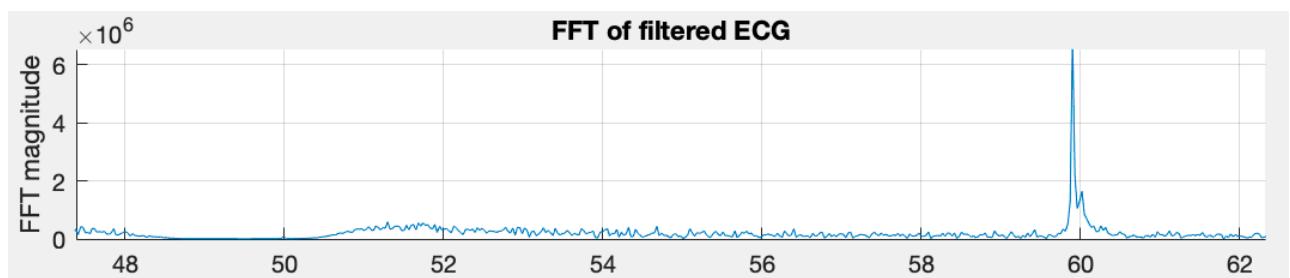
En esta sección se nos sugiere también que exploremos qué sucede cuando la cantidad de coeficientes elegidos para el filtro es demasiado alta. Elegiremos una cantidad de coeficientes de 2048 en lugar de 1536 que fue la elegida para el caso anterior.

Si hacemos esto obtendremos los siguientes resultados en la FFT y en su diferencia con respecto a la FFT de la señal sin filtrar.



A simple vista no parece que se hayan producido grandes cambios. Tan solo que la cantidad de señal en los 50Hz de frecuencia se ha reducido todavía más como era de esperar. En este caso la señal se ha reducido en un factor de 1738.5 lo cual es un orden de magnitud superior al factor con el filtro anterior.

Visualizando la FFT más de cerca podemos ver que la señal ha sido filtrada en exceso.

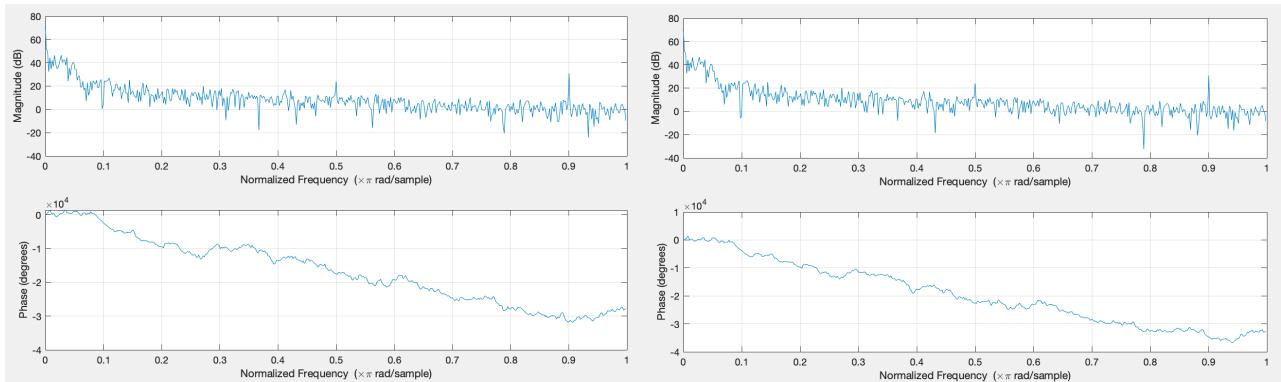


El pico en 50Hz y en general el resto de frecuencias en la banda están muy por debajo de lo que es habitual dentro de las frecuencias que forman parte del ECG. Además el filtro está empezando a afectar a las frecuencias colindantes con la banda que deseábamos eliminar.

También podemos explorar cómo la señal es afectada por cada uno de los filtros haciendo uso del comando `freqz`. Este comando nos muestra dos gráficas, para ambas el eje horizontal representa las frecuencias normalizadas entre [0,1]. Para la primera gráfica el eje vertical es la amplitud en decibelios y para la segunda la fase.

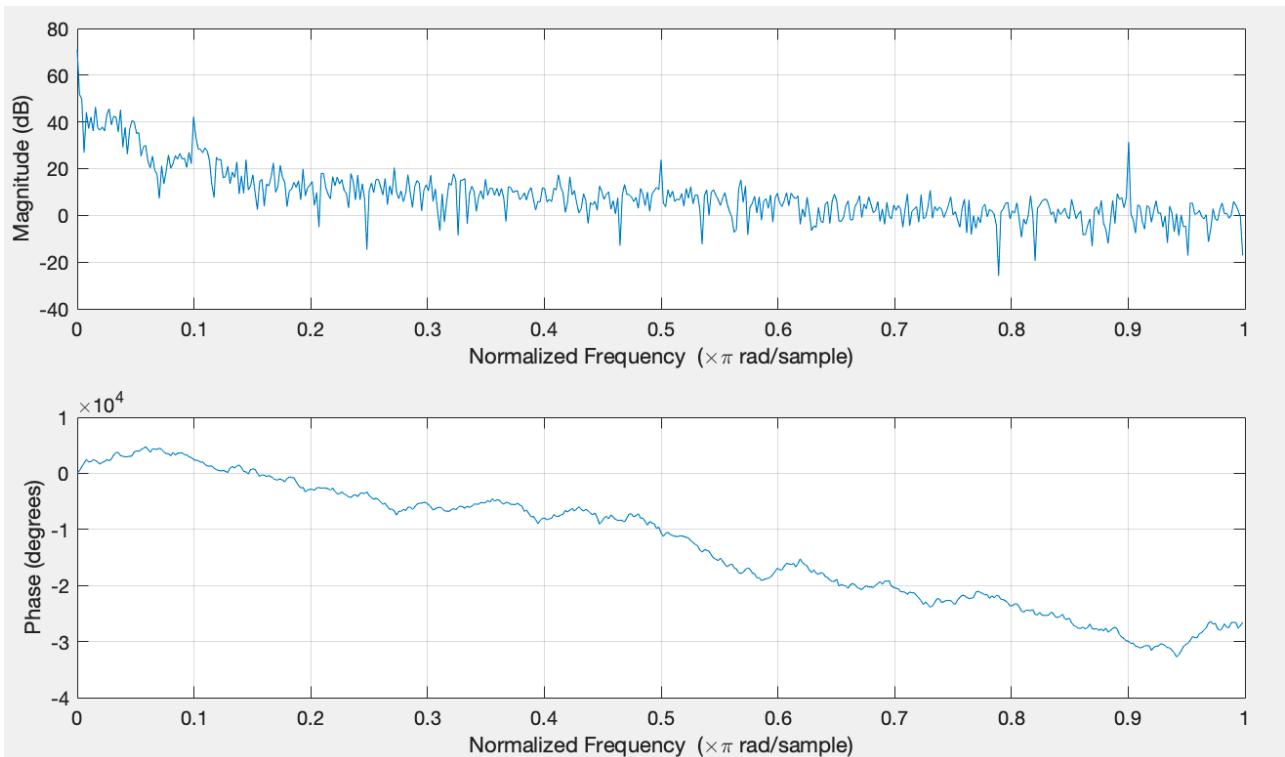
Analizando la segunda gráfica podríamos saber cuánto el filtro desfasa las señales que se le pasen. No obstante nos centraremos solo en el análisis de la amplitud.

No interesa conocer cómo se ha modificado la frecuencia de 50Hz. Debido a que nuestro rango de frecuencias en la señal está entre 0 y 500 Hz podremos encontrar en la columna 0.1 del eje horizontal el valor de estudio.

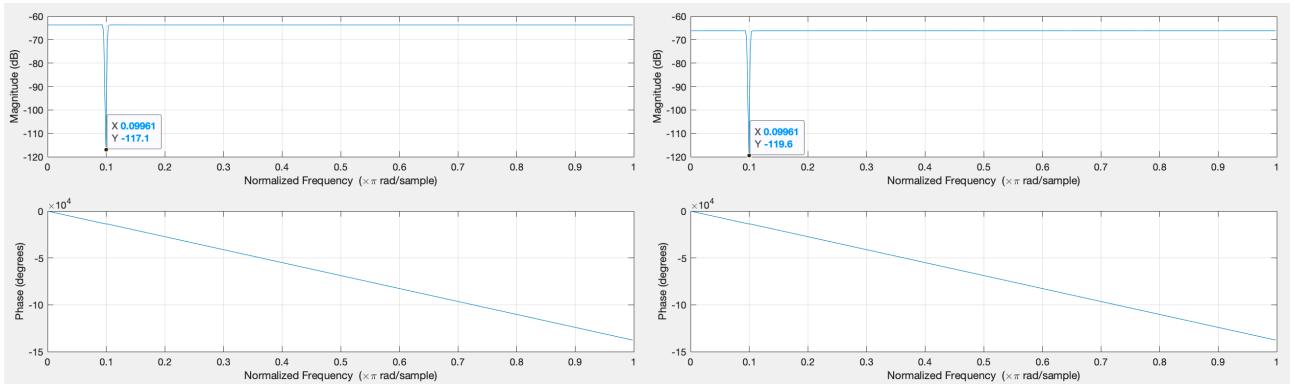


Obteniendo las gráficas para la señal filtrada con ambos filtros no es demasiado descriptivo ya que al ser los decibelios una magnitud en escala logarítmica no se aprecia demasiado la diferencia.

El valor con el primer filtro es de 0.55db y con el segundo de -4.678db de modo que la reducción con el segundo filtro es mayor además de negativa. En la señal original el valor era de 42.24db, podemos ver que en 0.1 del eje horizontal encontramos un pico en los valores del eje vertical en la primera gráfica. Por el contrario en las anteriores existe un valle muy pronunciado.

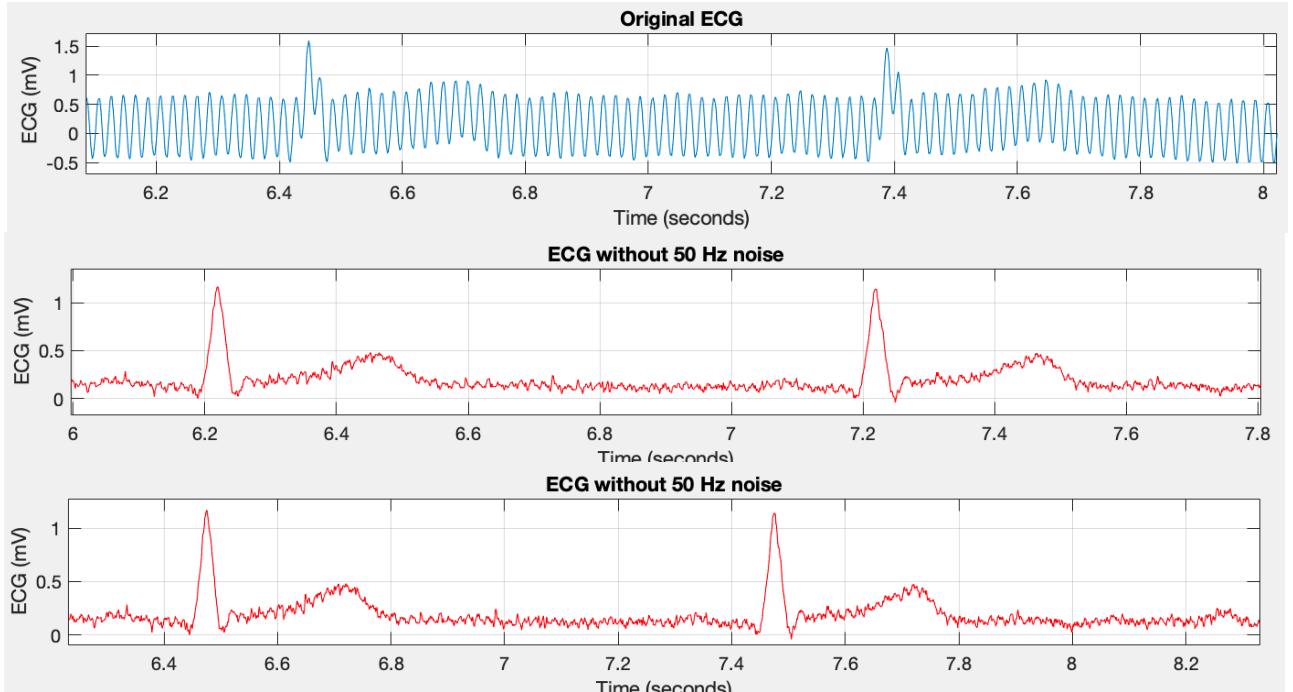


Este comando **freqz** podríamos haberlo utilizado también sobre los filtros directamente y no solo sobre la señal. Esto nos permitirá ver la respuesta del filtro.



Podemos ver a la izquierda el primer filtro y a la derecha el segundo. Podemos ver también que la respuesta de ambos se aplica sobre el intervalo indicado, entorno a los 50Hz. También se ve que el primer filtro tiene una respuesta menor, de -177.1db mientras que el segundo tiene una mayor de -119.6db.

Finalmente exploraremos la forma de la señal ECG para cada uno de los filtros. En general esta será de mayor calidad que la señal ECG original en la que no se podían apreciar demasiado bien las ondas típicas esperadas. No obstante con el ECG demasiado filtrado estaremos perdiendo información relevante y sin ni siquiera poderlo detectar a simple vista.



Podemos ver el ECG original seguido del ECG obtenido con el primer filtro y finalmente el obtenido con el segundo. Se aprecian grandes mejoras al aplicar cualquiera de los dos filtros, también que la señal ha sido desplazada cantidades distintas con cada uno a lo

largo del eje horizontal. No obstante sabemos por la evaluación anterior que la señal producida por el segundo filtro es peor pues carece de una cantidad irreal de la frecuencia de 50Hz.

Sobre la señal a la que le hemos eliminado la frecuencia de 50Hz con el filtro FIR con 1536 coeficientes continuaremos aplicando filtros hasta haber eliminado todos los ruidos descritos en la introducción. Para hacerlo utilizaremos distintos filtros FIR que no analizaremos con tanto detalle como con el filtro anterior por aligerar la explicación.

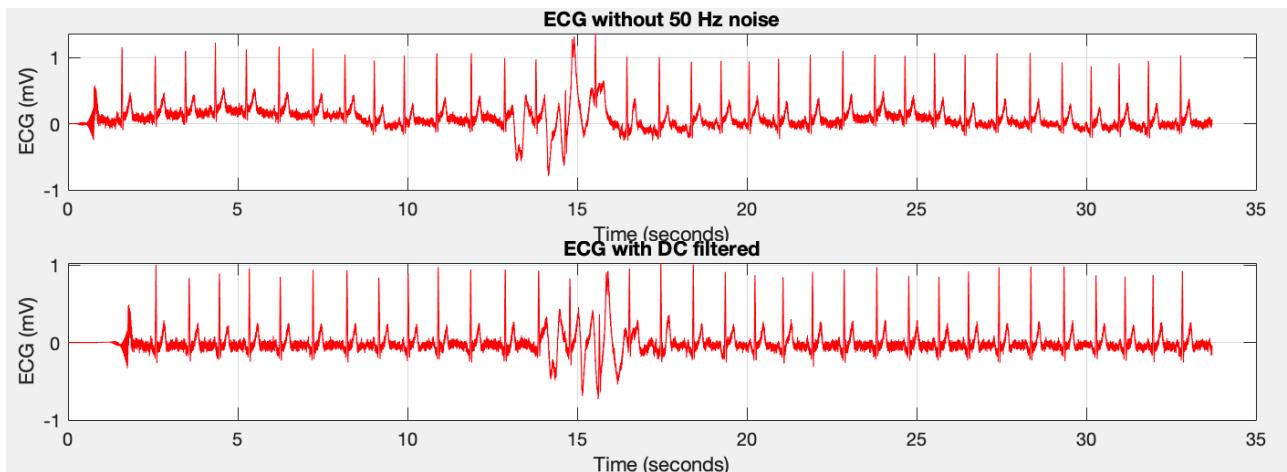
### Eliminación de baja frecuencia

Procedemos ahora a aplicar un filtro FIR que elimine las bajas frecuencias. En estas bajas frecuencias encontramos ruido proveniente de la componente DC de la señal y de las corrientes que se producen entre los electrodos y la piel al capturar el ECG.

Diseñamos un filtro que elimine las frecuencias inferiores a 1Hz y se lo aplicamos a la señal. Este será un filtro paso alto que eliminará todas las frecuencias superiores a 1Hz.

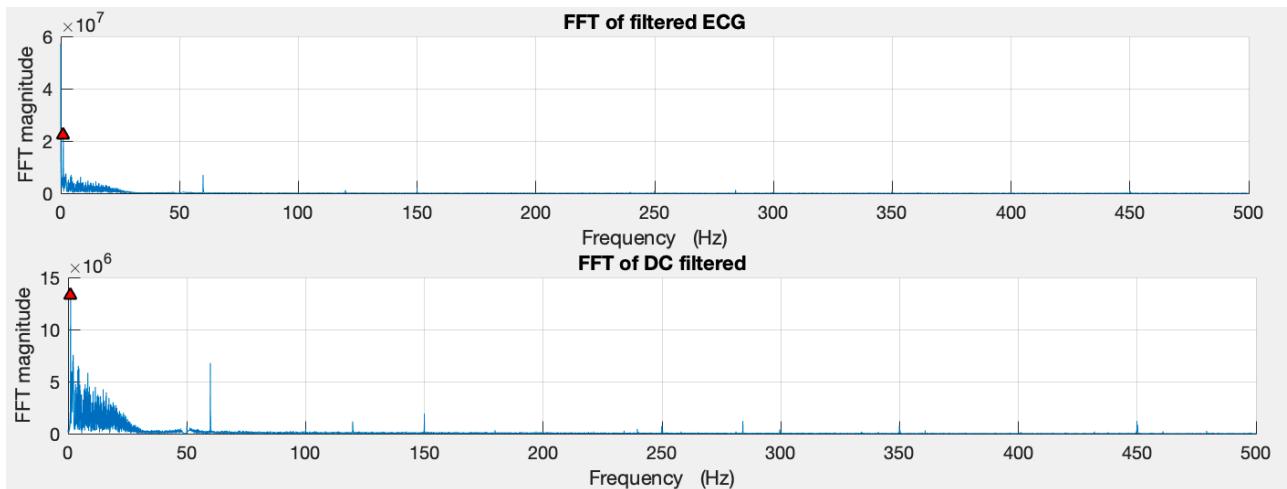
```
%% Delete DC with highpass filter  
highpass = fir1(2000, 1/half_sampling_rate, 'high');  
ecgnoDC = filter(highpass, 1, ecgno50);
```

Como vemos en el ECG resultante la eliminación de las frecuencias por debajo del 1Hz se aprecia al ver el ECG en escalas de tiempo mayores que cuando eliminamos la banda de frecuencia de 50Hz. Para ver correctamente la eliminación de esta banda tuvimos que hacer zoom al ECG para poder apreciar con detalle suficiente las modificaciones realizadas sobre la señal. En este caso, no será necesario.

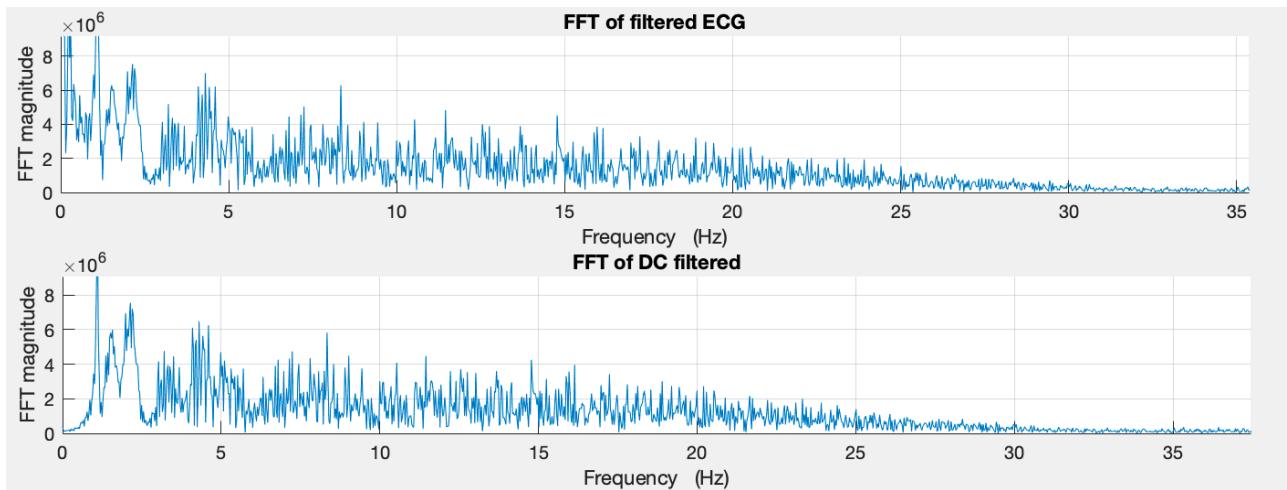


Mostramos primero la señal con la banda de 50Hz eliminada y debajo esa misma señal pero eliminando las frecuencias por debajo de 1Hz. Podemos ver que estas frecuencias perturban la línea base del ECG capturado, perturbación que desaparece al eliminarlas.

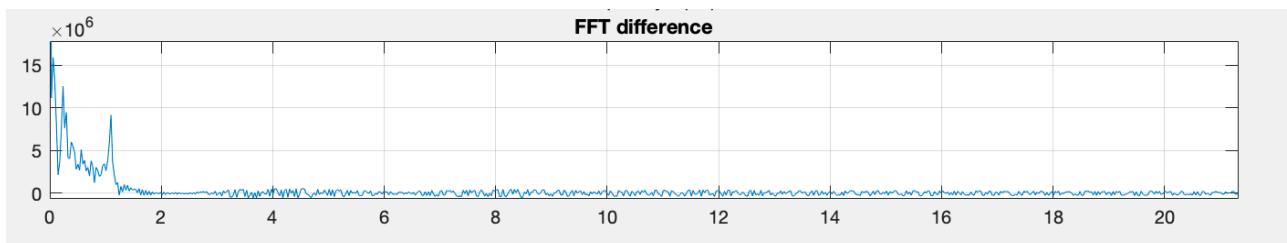
Si nos centramos en el FFT de las señal podemos ver que la acumulación de frecuencias al inicio del ECG se ha reducido en un orden de magnitud.



Explorando esta reducción más de cerca podremos ver exactamente lo que ha ocurrido. Cerca de los 0 Hz existían valores muy altos que el filtro ha eliminado. Por encima de 1Hz existe también un pico de valores que es el que se mide con el comando **findpeaks**. El primero no se mide pues realmente no es un pico ya que a su izquierda no hay valores más bajos ya que no hay ningún valor.



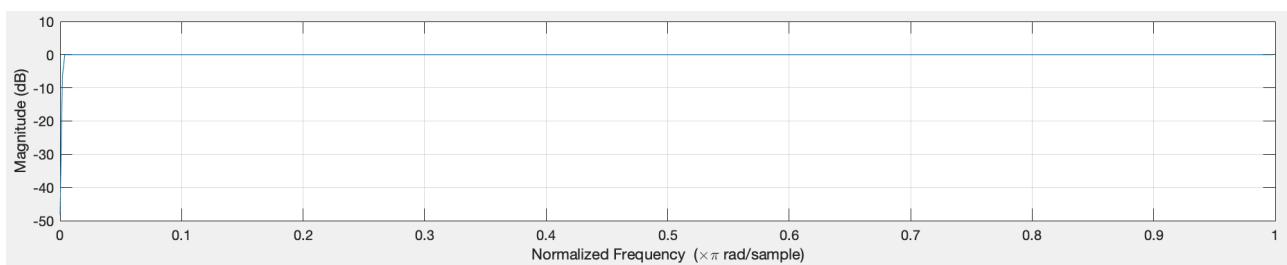
No obstante si vemos la gráfica de la diferencia de las dos anteriores FFT vemos que en este pico así como en todas las frecuencias el filtro ha eliminado valores también. Aunque por supuesto muchos menos en comparación con los que elimina por debajo de 1Hz.



Con todo esto podemos saber que la frecuencia situada en valores de 0 Hz se ha atenuado con un factor de 149.4 pasando de un valor de  $5.7401e+07$  a  $3.8410e+05$ . Por el contrario el pico que está justo por encima de 1Hz quedando por tanto fuera de las frecuencias que el filtro eliminaría se ha atenuado en un factor de 1.69 pasando de un valor de  $2.2442e+07$  a uno de  $1.3265e+07$ .

Una vez más podemos ver que el filtro aunque afecta en un factor mucho mayor a la zona que deseamos afecta también a las frecuencias colindantes aunque queden fuera de su zona de acción.

Esto mismo podría haberse visto con el comando **frqz** aplicado directamente sobre el filtro.



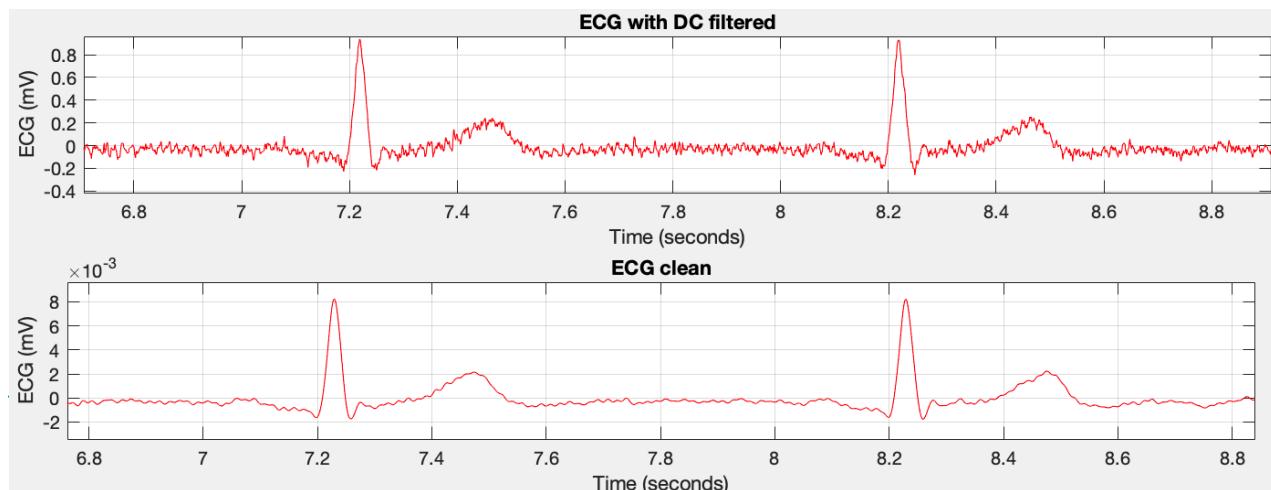
La gráfica nos muestra que las señales más bajas serán eliminadas y la atenuación que se producirá sobre estas.

### Eliminación del ruido de alta frecuencia

Finalmente para eliminar las frecuencias excesivamente altas y que sabemos que forman parte del ruido pues el ECG no se manisera en ellas utilizaremos el filtro opuesto, un filtro pasa bajos. El filtro pasa bajos dejará pasar todas las frecuencias por debajo de la que le indiquemos, en este caso dejaremos las frecuencias menores a 100Hz.

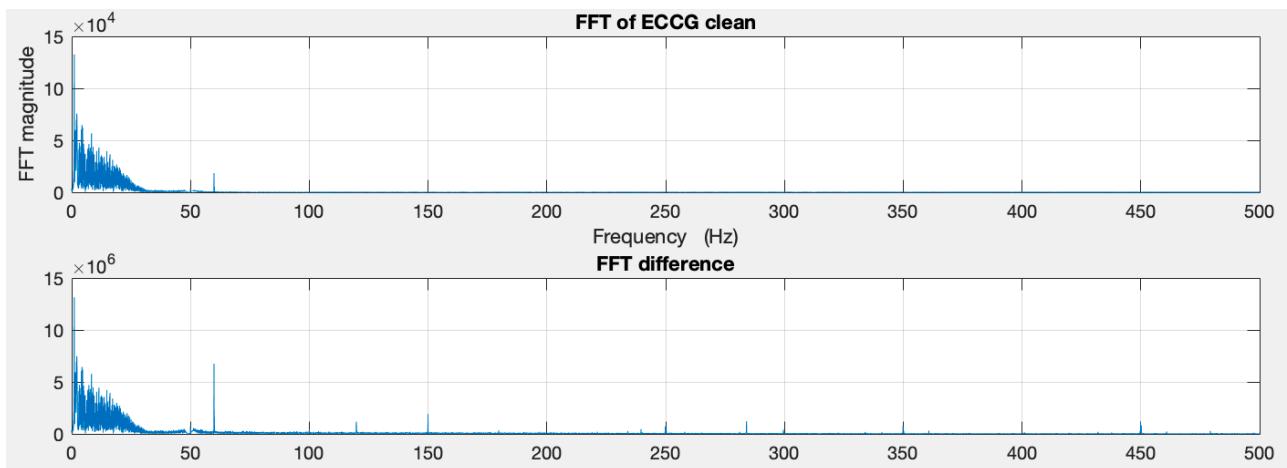
```
%>> Delete high freq with lowpass filter
lowpass = fir1(20, 1/half_sampling_rate, 'low');
ecgClean = filter(lowpass, 100, ecgnoDC);
```

Para evaluar los resultados de la aplicación de este filtro sobre la señal deberemos hacer zoom en ella. La comparemos con la señal que se obtuvo en la sección anterior.



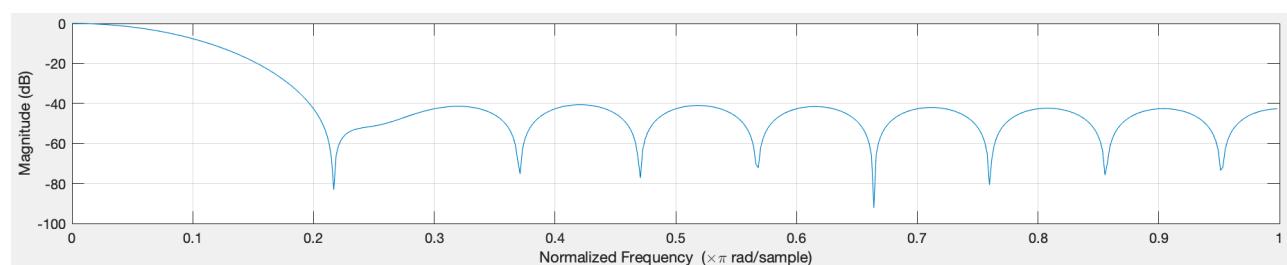
Podemos ver que la nueva señal, en la parte inferior, es mucho más limpia que la anterior pues le hemos eliminado todas las impurezas. Vemos también que la amplitud del QRS no se ha modificado ni se han perdido las ondas Q y S que son más débiles. La onda P (antes del QRS) puede verse muy sutilmente, la onda U (entre la P y la T) es inapreciable lo cual suele ser habitual y la onda T se aprecia sin ningún problema.

Analizando lo que ha sucedido en el FFT monolateral podemos ver que en general la magnitud para todas las frecuencias se ha reducido considerablemente. Adicionalmente se aprecia que a partir de los 100Hz es reducción ha sido especialmente notable.



Mostramos el FFT de la nueva señal en la parte superior y la diferencia de este con el de la señal anterior. Podría ser una preocupación justificada haber perdido información al haber aplicado este filtrado. Como vemos se han ido más frecuencias de las que han quedado. No obstante en el análisis visual de la señal esto no parece ser así. Se siguen apreciando todas las ondas, incluso mejor que antes, sin distorsiones aparentes ni un desplazamiento en el tiempo excesivo.

Aplicando el comando `freqz` sobre el filtro podemos ver la misma información. A partir de los 100Hz se producirá una gran atenuación sobre la señal aunque para lograr esta atenuación estaremos también afectando a las frecuencias inferiores a los 100Hz lo cual es un pérdida de información que podría ser relevante.



La señal finalmente obtenida tras la aplicación de todos los filtros resulta mucho más limpia y clara que la original y nos permite visualizar de una forma mejor las señales que se nos proporcionó.

## ELIMINAR EL RUIDO BORRANDO LOS COEFICIENTES NO DESEADOS DE LA FFT DE LA SEÑAL ORIGINAL Y RECONSTRUYÉNDOLA MEDIANTE UNA IFFT

La idea de esta técnica de filtrado es convertir la señal en una FFT y eliminar de ella aquellas partes que cuando se aplique la Transformada Inversa de Fourier desaparezcan las frecuencias causantes del ruido.

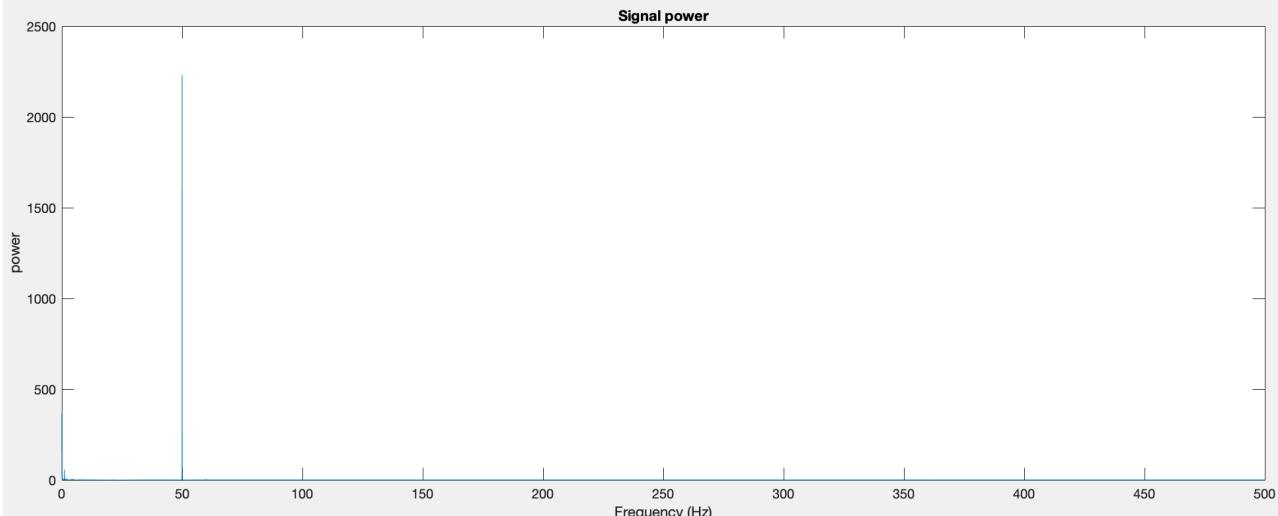
Los pasos que se deben seguir son detectar aquellas partes de la FFT en las que se encuentra el ruido, intervalo entorno a los 50Hz y valores por debajo de 1Hz. También podrían eliminarse los valores por encima de 100Hz.

Estos valores se pondrán a 0 en la FFT de modo que al reconstruirla desaparezcan dejando una señal limpia.

Partiendo del ECG en la escala adecuada calculamos su potencia y espectro de la FFT.

```
len = length(time_vector);
frequency=fft(ecg)/sqrt(len);
range=(1:floor(len/2)+1);
power=abs(frequency(range)).^2;
spectrum=frequency.*conj(frequency);
```

Partiendo de esta FFT podemos identificar los valores que deseamos eliminar, los menores de 1Hz, los mayores de 100Hz y el pico entorno a los 50Hz.



Los valores menores de 0.8Hz los podemos encontrar en índices del espectro por debajo del valor de la variable low.

```
spectrum_len=length(spectrum);
hz=ceil(1/(500/spectrum_len));
low=floor(hz*0.8);
```

Los valores mayores de 100Hz los podemos encontrar en índices del espectro por encima del valor de la variable high.

```
high=hz*440;
```

Finalmente esto nos da el siguiente vector de valores a eliminar del espectro.

```
diff=spectrum_len-(low+high);
delete=[zeros(low,1); ones(diff,1); zeros(high,1)];
```

Para detectar la ubicación del ruido de 50 Hz de forma exacta calculamos los índices que superan cuyo valor está por encima de un valor tal que solo el ruido de 50Hz llegue a él, adicionalmente dejamos un margen antes y después de esta región detectada.

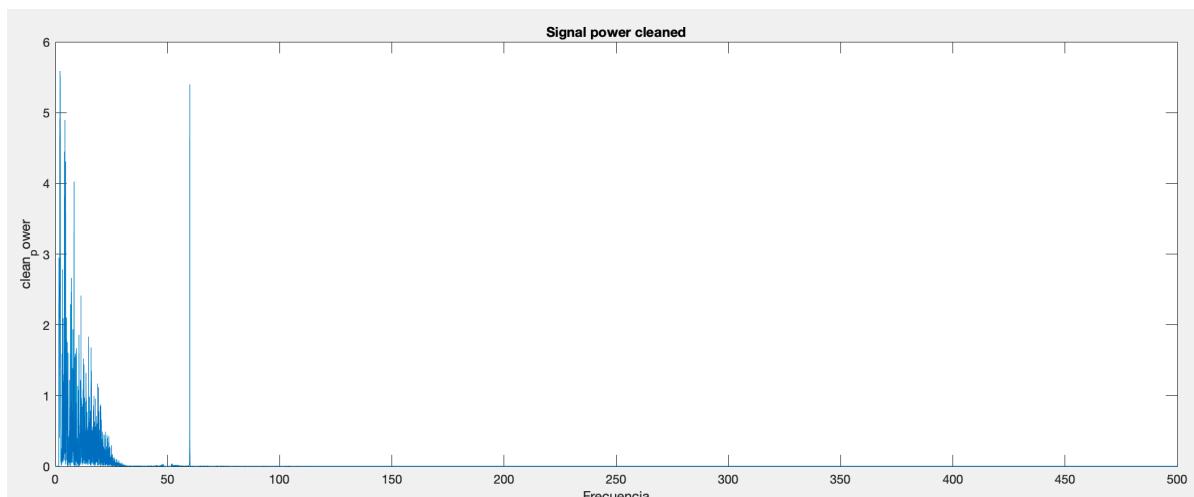
La región la detectamos mediante la siguiente comparación que generará un vector con 0s en la región donde se encuentra el ruido, para ver cómo se expande el margen referirse al código pues son muchas líneas.

```
noise=(spectrum<5);
```

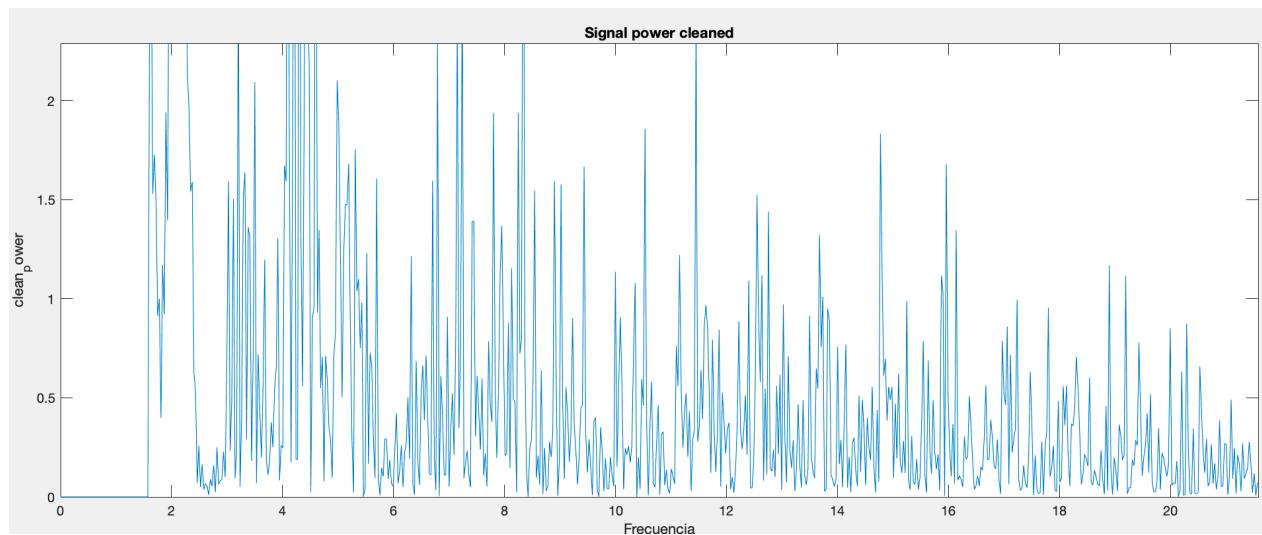
Finalmente borramos del espectro los valores marcados en el vector delete.

```
clean_frecuency=frequency.*(delete);
clean_power=abs(clean_frecuency(range)).^2;
clean_ecg=ifft(clean_frecuency)*sqrt(len);
```

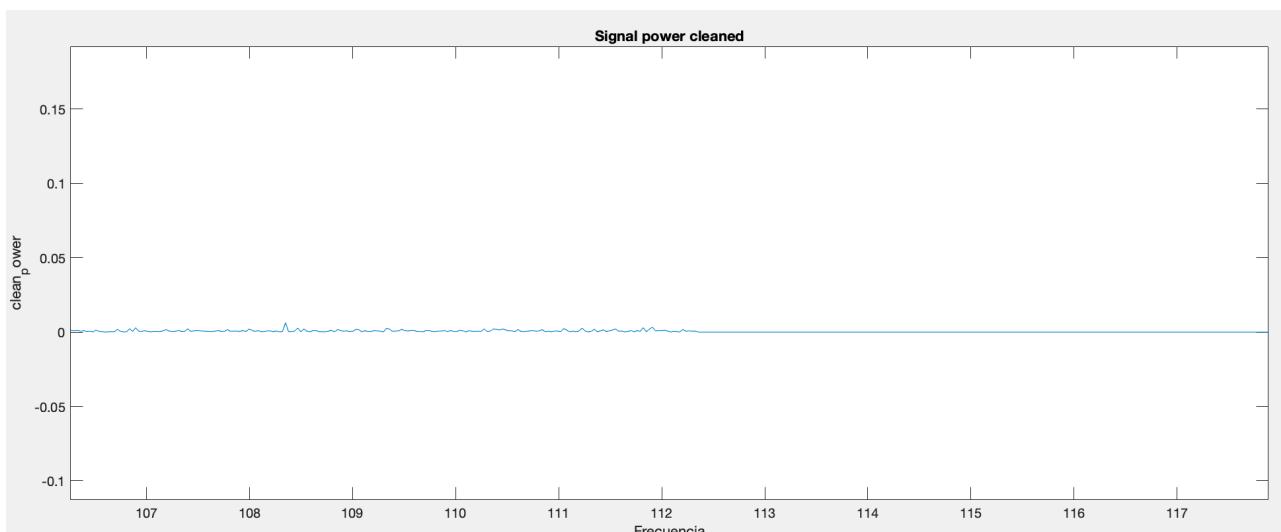
Los que nos da la siguiente FFT



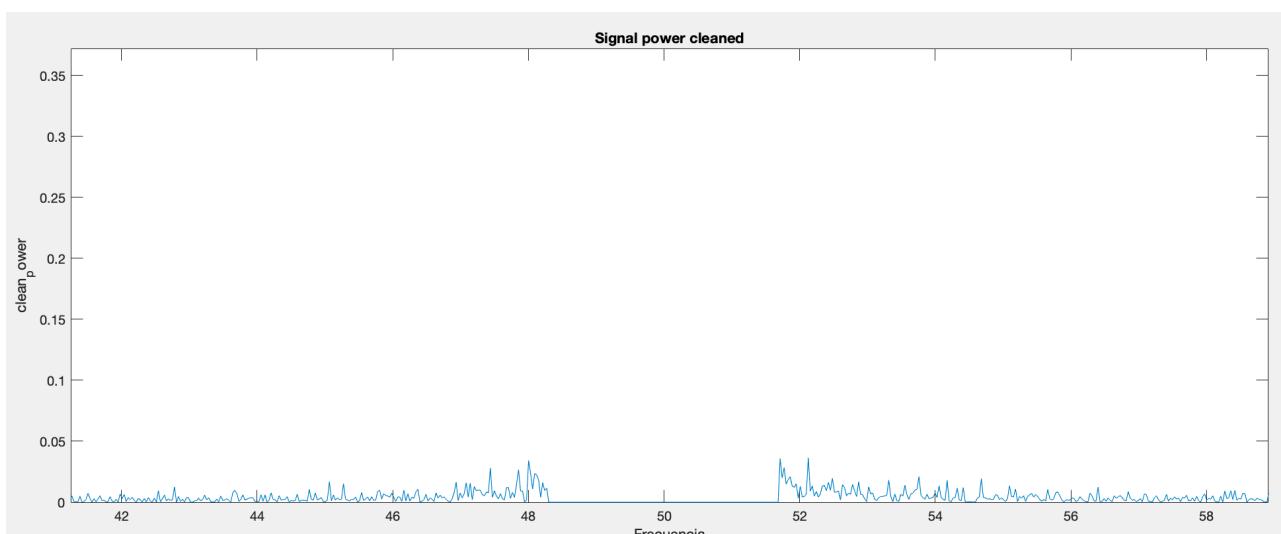
Podemos ver el corte por debajo de 1 Hz que imitaría el efecto del filtro pasa altos.



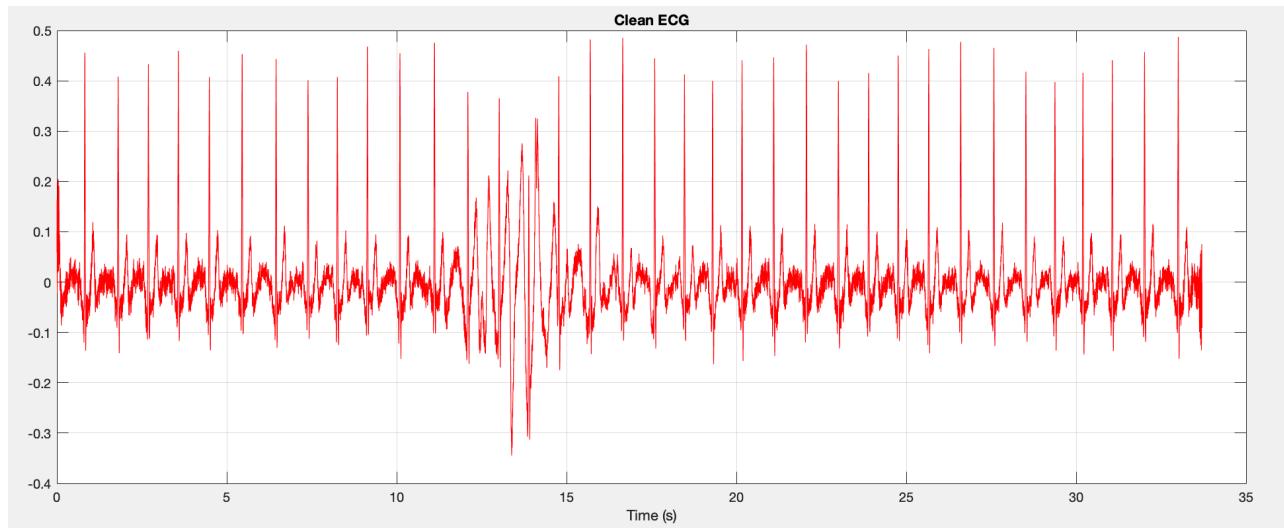
El corte por aproximadamente por encima de los 100 Hz que imitaría al filtro pasa bajos.



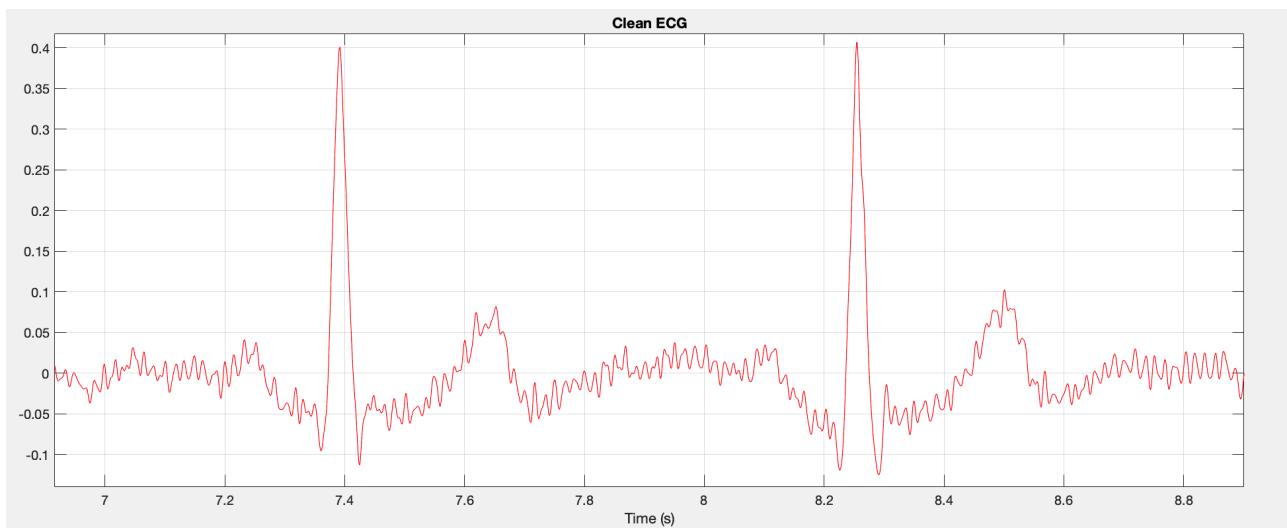
El corte entorno a los 50 Hz que imitaría al filtro para banda.



Finalmente inspeccionamos el ECG reconstruido. A simple vista podemos ver que las bajas frecuencias han sido eliminadas pues la señal está recta, es decir, no tiene ondulaciones en su línea base.



Haciendo zoom podemos ver que el ruido de alta frecuencia así como el de 50Hz han sido eliminados también. Todas las ondas del ECG pueden distinguirse claramente, la onda P al inicio, el QRS, la onda T incluso puede atisbarse la onda U.



Los resultados de utilizar esta técnica son muy similares a los de usar filtros FIR, la señal obtenida es muy limpia. Adicionalmente nos permite seleccionar de forma muy clara las frecuencias a eliminar.

El mayor problema de esta técnica es que la eliminación de las frecuencias deja cortes muy pronunciados en la FFT y no cortes suaves como cuando se usa un filtro lo cual podría dar problemas al reconstruir el ECG.

## Filtrado de la señal mediante filtros IIR

Se ha examinado el código utilizado por los compañeros de otros años con la intención de aprender lo que ellos realizaron para conseguir completar este apartado.

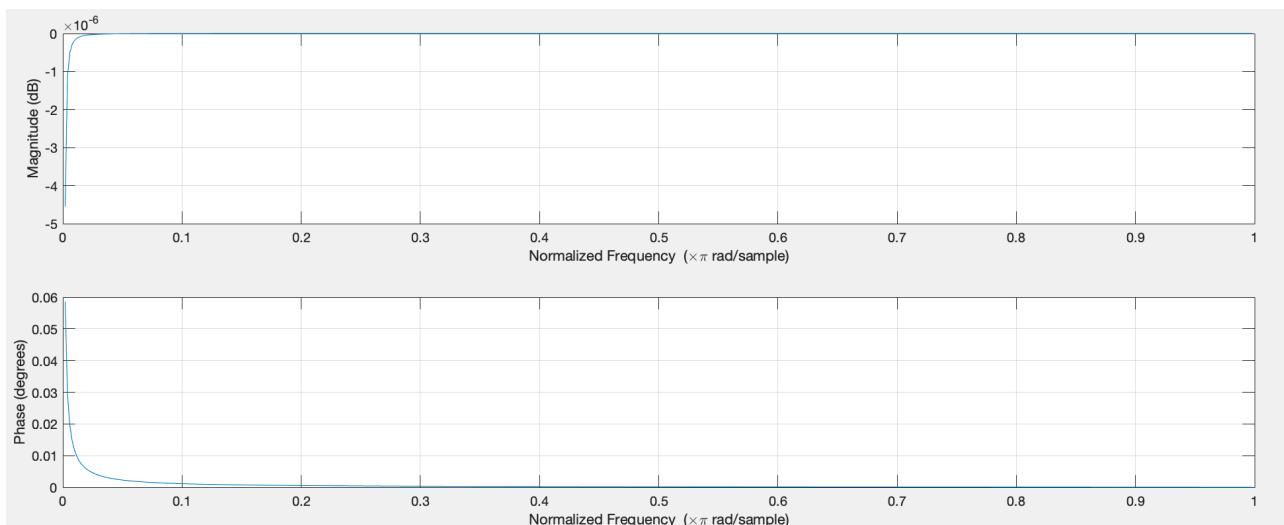
Los filtros IIR son filtros más potentes que los filtros FIR en el sentido de que configurados con menores coeficientes logran obtener atenuaciones mayores sobre las frecuencias para las que hayan sido configurados que los filtros FIR. En los apartados anteriores hemos utilizado los siguientes coeficientes: 1536, 2000 y 20; para los filtros de eliminación de los 50Hz, eliminación de las frecuencias por debajo de 1Hz y eliminación de las frecuencias por encima de 1000Hz respectivamente. Al utilizar filtros IIR estos coeficientes serán mucho menores.

El problema con los filtros IIR y la razón por la que no sean utilizados en todos los casos el lugar de los FIR es que su zona de actuación no es tan concentrada como en los otros. Al aplicar los filtros IIR afectaremos también a las frecuencias próximas a aquellas que realmente queramos eliminar de modo que no siempre serán de utilidad pues distorsionarán la señal de entrada en lugar de limpiarla.

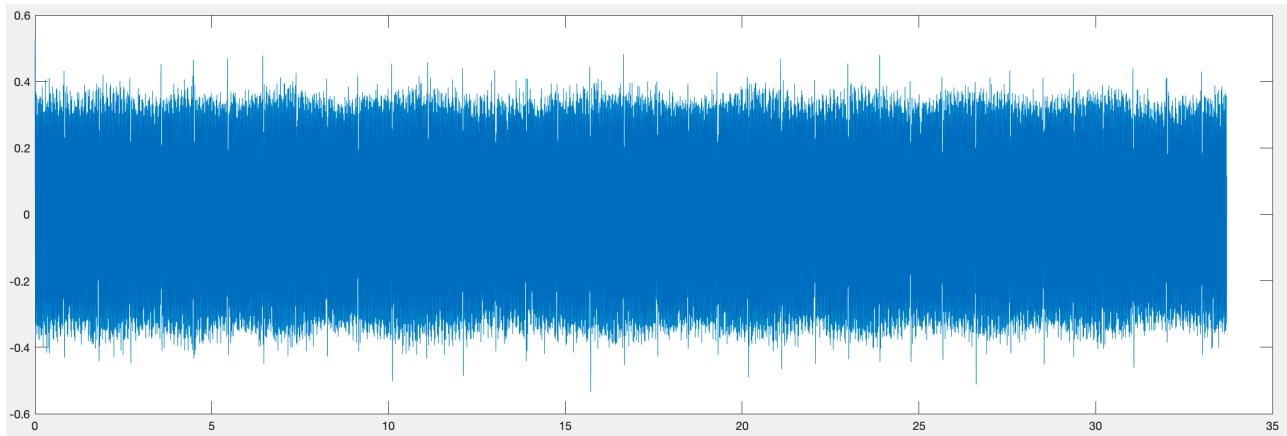
Diseñamos un filtro pasa altos que deje pasar todas las frecuencias por encima de 0.8 Hz.

```
hig_pass=0.001/half_sampling_rate;  
N1=1;  
[a1,b1] = butter(N1, hig_pass,'high');  
ecg_1=filter([a1,b1],1,ecg);
```

En la FFT del filtro comprobamos que en efecto eliminará frecuencias que sean bajas.



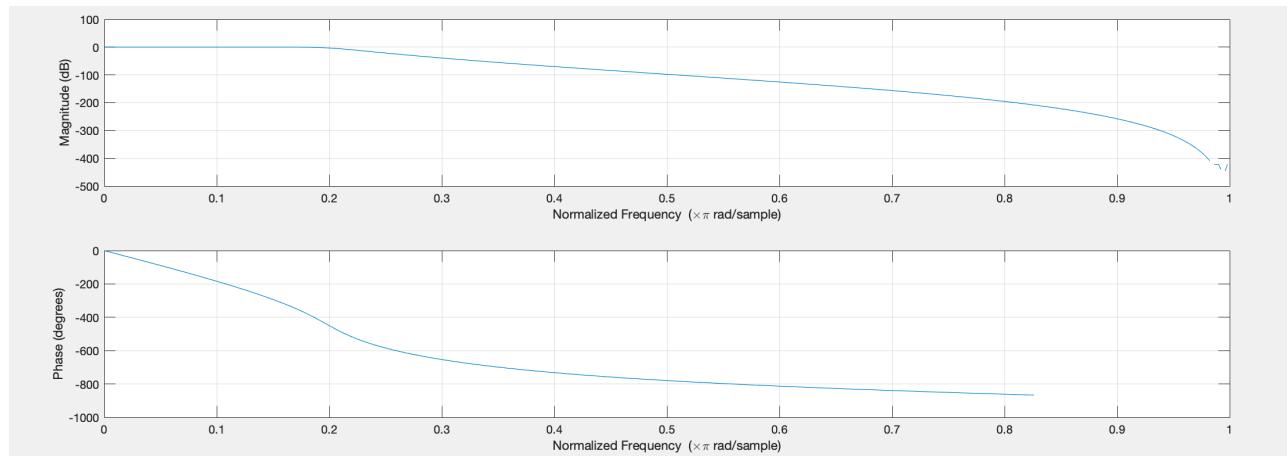
No obstante aunque las intenciones fueran buenas los resultados obtenidos son pésimos. El ECG resultante está excesivamente distorsionado a pesar de haber utilizado el menor coeficiente posible para aplicar el filtro. Este borra frecuencias más altas de lo que debería.



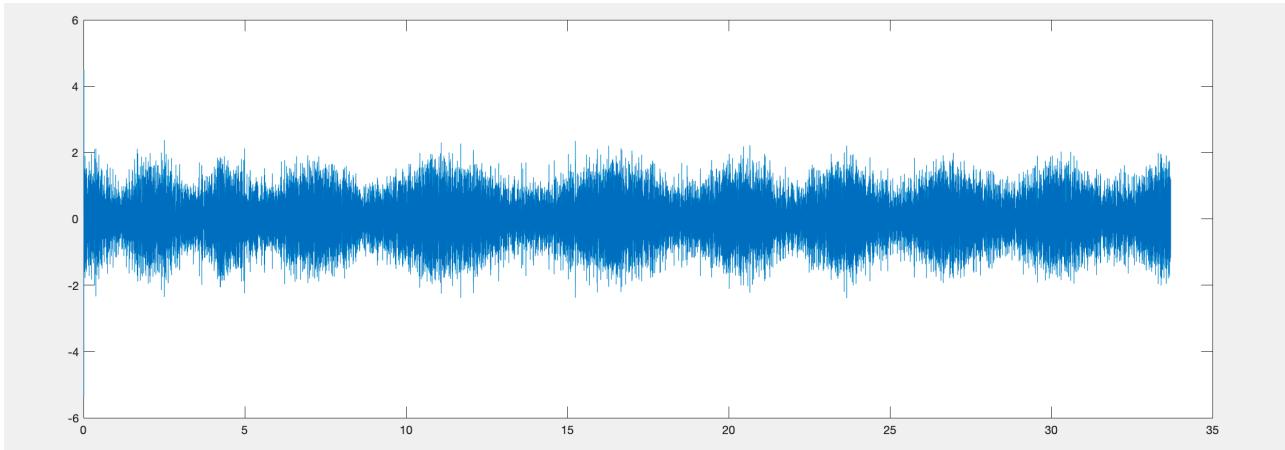
Diseñamos ahora un filtro pasa bajos que elimine las frecuencias por encima de los 100Hz.

```
low_pass=100/half_sampling_rate;  
N2=10;  
[a2,b2]=butter(N2,low_pass);  
ecg_2=filter([a2,b2],1,ecg);
```

En la FFT del filtro comprobamos que en efecto este parece que va a hacer lo que deseamos.



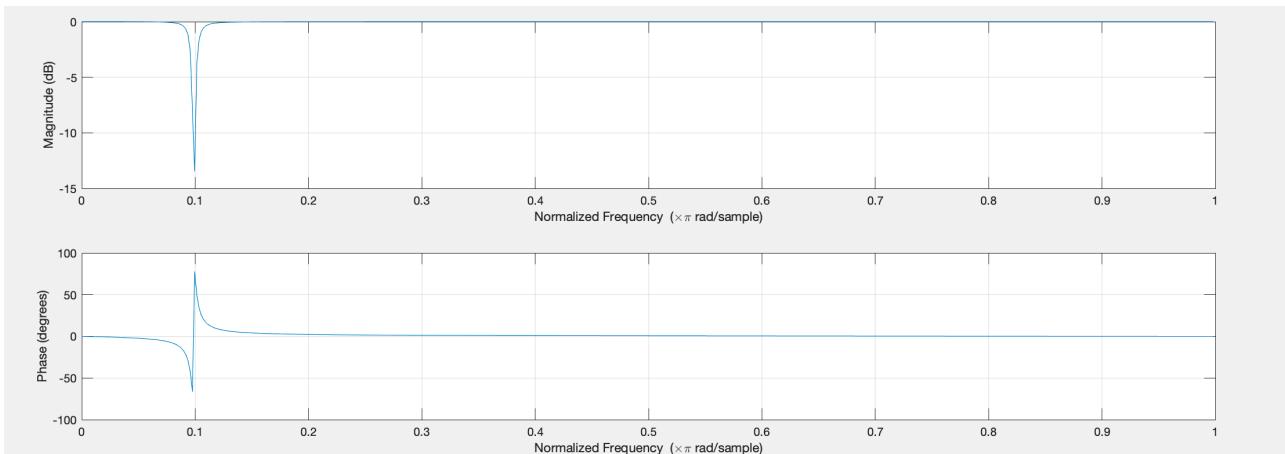
No obstante al aplicarlo los resultados son poco satisfactorios.



Diseñamos también un filtro que elimine el entorno de los 50 Hz.

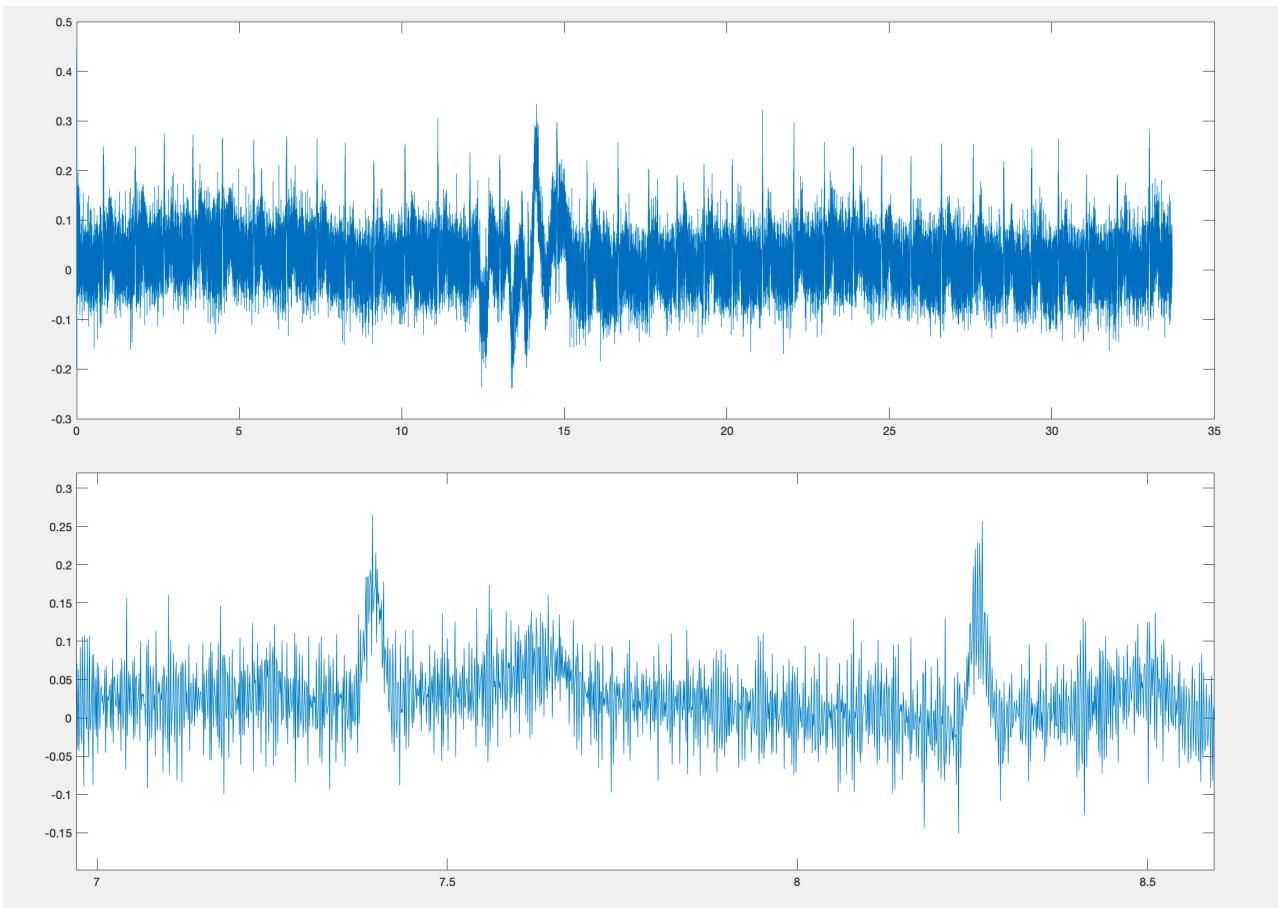
```
start_cut=48/half_sampling_rate;
end_cut=51/fm3e;
N3=1;
[a3,b3]=butter(N3,[start_cut,end_cut],'stop');
ecg_3=filter([a3,b3],1,ecg);
```

Observando la FFT del filtro una vez más podemos ver que este parece ir a hacer lo que deseamos.



---

Y en este caso logramos un resultado mucho más satisfactorio que las veces anteriores.



Podemos finalmente concluir que los filtros IRR, o al menos este tipo en concreto de filtro IRR no es adecuado para eliminar las altas o bajas frecuencias del ECG pues lo distorsiona hasta el punto de no poder reconocer ninguna de sus ondas características.

No instanté si es un filtro adecuado para eliminar el ruido de 50Hz para lo cual necesita tan solo un coeficiente mientras que el filtro FIR utilizaba 2000.

## Promediado coherente de señales

Esta técnica de filtrado o eliminación de ruido nos permite eliminar ruido blanco o de media cero del cual desconozcamos su procedencia. Con desconocer su procedencia nos referimos a que o bien no sabemos exactamente en que rango de frecuencias se encuentra el ruido o bien a que este se ubica sobre rangos de frecuencias en las que por existir contenido que no deseamos eliminar no podemos aplicar los filtros vistos en la parte anterior.

El requisito necesario para poder utilizar este filtro es necesario que las señales a filtrar sean coherentes, es decir, que coincidan en el tiempo. Necesitaremos múltiples registros coherentes de la señal, cuantos más mejor. Calculando la media de todos estos registros el ruido quedará anulado pues como dijimos para poderlo eliminar debe tener media cero.

Simularemos tener registros coherentes de media cero por medio de aplicar ruido blanco a la misma señal. Con esto generaremos múltiples señales ruidosas de las que podremos calcular su media eliminando el ruido.

Haremos uso de un timer de un segundo de periodo como se nos indica en la práctica aunque con ligeras modificaciones. Crearemos un ventana global cuyo contenido sobre escribiremos dentro de la función callback del timer. Cuando la ventana se cierre llamaremos a una función que cerrará el timer de modo que garantizamos tener solo uno por ventana.

```
global t
```

```
% Create the figure where data will be plotted (on window closed timer is cleared)
figure('Name', 'ECG promediado', 'CloseRequestFcn', @clear_timer);
% On timer period window is updated
t = timer('TimerFcn', {@promedio_call_back}, 'Period', 1);
set(t, 'ExecutionMode', 'fixedRate');
start(t)

% Clears timer and window
function clear_timer(~, ~)
    global t
    stop(t)
    delete(t)
    delete(gcf)
end
```

La señal sobre la que vamos a aplicar un ruido blanco distinto a cada llamada del timer la creamos haciendo uso de la función **ecg** y del suavizado **sgolayfilt** como vimos en la práctica anterior.

```
global latido_ecg
```

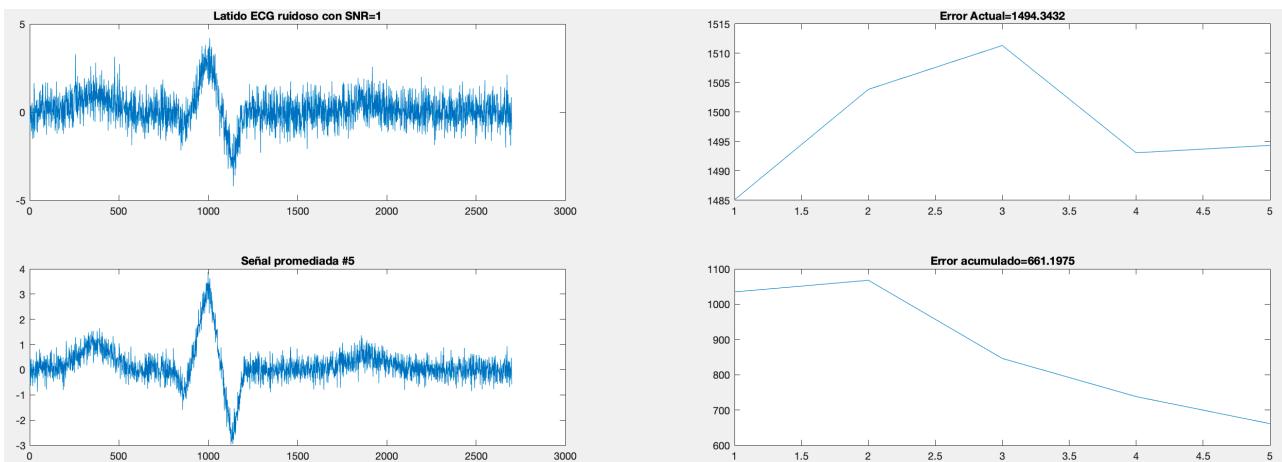
```
% Generamos un latido QRS con 2700 muestras de resolución y amplitud 3.5.  
X = 3.5*ecg(2700);  
% Smooth ECG  
latido_ecg = sgolayfilt(kron(ones(1, 1), X), 0, 31);
```

Dentro de la función a la que el timer llama creamos el ruido para la señal, almacenamos la señal ruidosa y calculamos la media de todas las señales ruidosas. También almacenamos el error absoluto obtenido entre la señal ruidosa y la ideal y entre la señal filtrada y la ideal para mostrar su evolución en el tiempo.

```
global numero_promedios ecg_promediado current_error acc_error  
  
numero_promedios = 0;  
ecg_promediado = [];  
current_error = [];  
acc_error = [];  
  
% Created new ECG and Updates the window with it  
function promedio_call_back(~,~)  
    global numero_promedios ecg_promediado latido_ecg current_error acc_error  
    numero_promedios = numero_promedios + 1;  
  
    SNR=1;  
    ecg_ruidoso = awgn(latido_ecg, SNR,'measured','linear');  
    current_error = [current_error sum(abs(latido_ecg - ecg_ruidoso))];  
  
    %Acumulamos ecg_ruidoso en ecg_promediado:  
    ecg_promediado = [ecg_promediado; ecg_ruidoso];  
    promediado = mean(ecg_promediado);  
    acc_error = [acc_error sum(abs(latido_ecg - promediado))];  
  
    % PLOT ECGs  
    ...  
end
```

Para generar el ruido blanco utilizamos la función awgn con un ratio entre señal y ruido de valor unidad.

Visualizamos la última señal ruidosa generada en la parte superior izquierda. Bajo ella se encuentra el promediado actual. Con solo 5 señales generadas con las que poder promediad podemos ver que el ruido en dicha señal es todavía alto.



En la parte superior izquierda mostramos la gráfica de la evolución del error absoluto medio entre cada señal ruidosa generada y la ideal. Bajo él mostramos el error absoluto medio entre el promedio y la señal ideal.

Al aumentar el número de señales de las que disponemos para realizar el promedio vemos cómo el ruido de la señal se ha reducido considerablemente. Vemos también que el error de las señales generadas es aleatorio, pero que el de la señal promediada siempre disminuye aunque cada vez lo hace más lentamente.

Una de las ventajas de haber creado el timer y la ventana tal y como lo hemos hecho es que una vez obtenido un promedio satisfactorio podremos cerrar la ventana cancelando a la vez el timer sin tener que acordarnos de cancelarlo cada vez.

