



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

2^{do} Cuatrimestre 2019

SEÑALES Y SISTEMAS (66.74)

Trabajo Práctico Especial

TEMA: DETECCIÓN AUTOMÁTICA DEL COMPLEJO QRS

FECHA: 4 de diciembre de 2019

Apellido y Nombre	Padrón	Correo Electrónico
Ferres, Julian	101483	julianferres@gmail.com

Índice

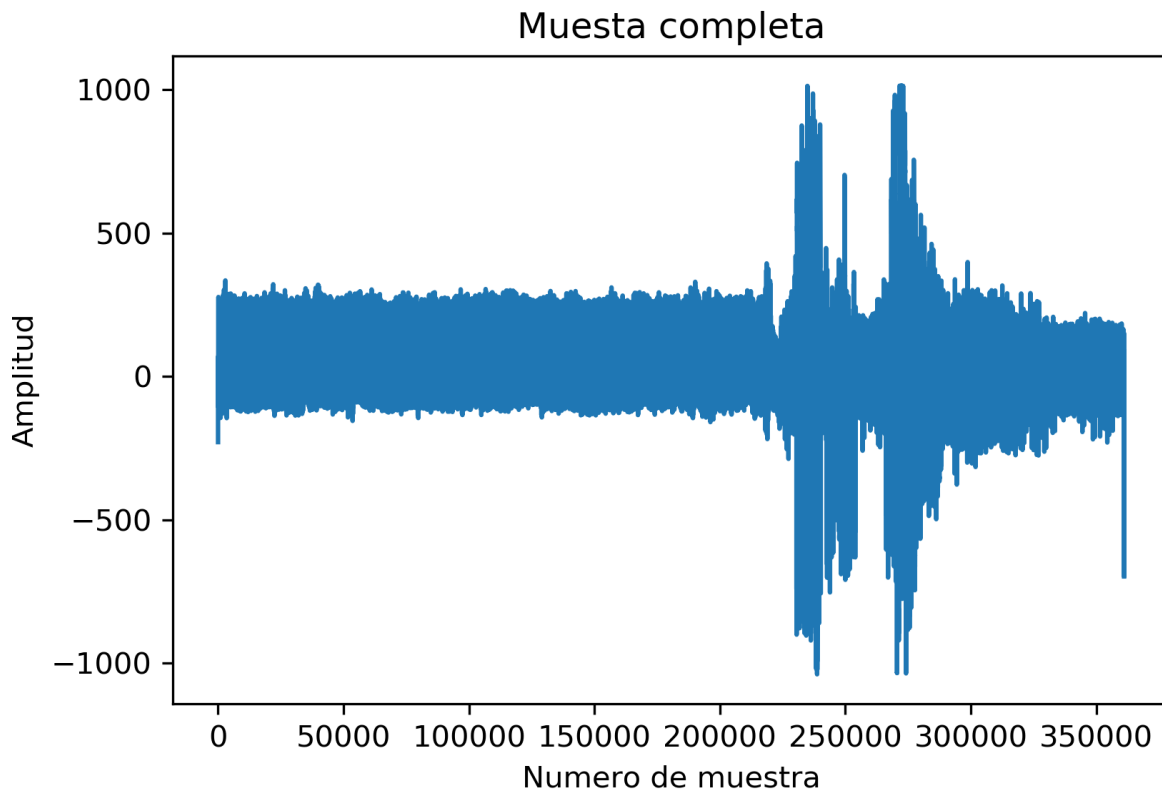
1. Ejercicio 1	4
2. Ejercicio 2	7
3. Ejercicio 3	10
4. Ejercicio 4	14
5. Ejercicio 5	16
5.1. Polos y Ceros de H_L	16
5.2. Respuesta en Frecuencia de H_L	17
5.3. Respuesta al impulso de HL	18
5.3.1. Retardo HL	18
5.4. Respuesta al escalón de HL	19
5.5. Ecuacion en diferencias	19
6. Ejercicio 6	21
6.1. Polos y Ceros de H_H	21
6.2. Respuesta en Frecuencia de H_H	22
6.3. Respuesta al impulso de HH	23
6.3.1. Retardo HH	23
6.4. Respuesta al escalón de HH	24
6.5. Ecuacion en diferencias	24
6.6. Filtros aplicados a la señal	25
7. Ejercicio 7	26
7.1. Ventajas y desventajas	26
8. Ejercicio 8	27
9. Ejercicio 9	31
10. Ejercicio 10	33
11. Ejercicio 11 (opcional)	36
12. Ejercicio 12	37
13. Ejercicio 13	38
13.1. Calculo del desvio necesario segun SNR	38
13.2. Resultados obtenidos	38
13.2.1. $SNR_{dB} = 10dB$	39
13.2.2. $SNR_{dB} = 20dB$	40
13.2.3. $SNR_{dB} = 30dB$	40
13.3. Conclusiones	41

14.Ejercicio 14	43
14.1. Evaluación de desempeño	46
14.2. Conclusiones	46
15.Conclusión	47

1. Ejercicio 1

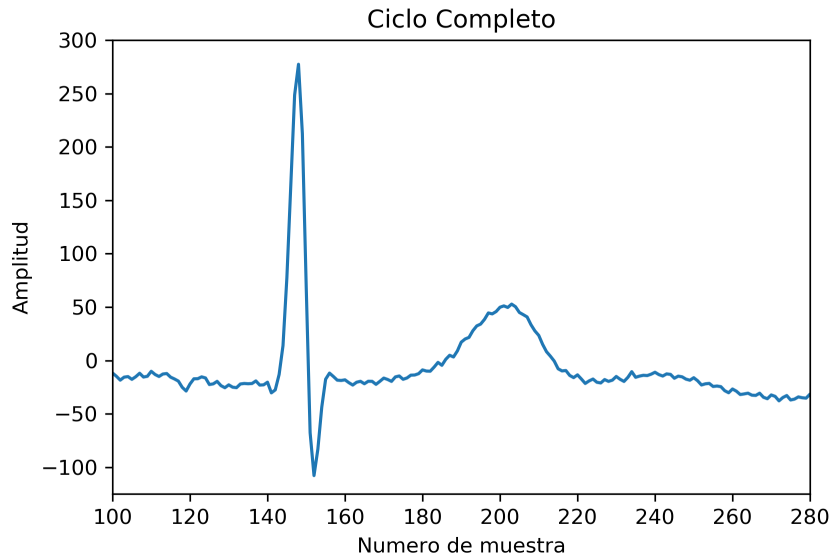
” Ubicar en forma manual los complejos QRS, determinando los segmentos que componen la onda.”

Solución: Antes de comenzar adjunto la imagen de la medición completa:



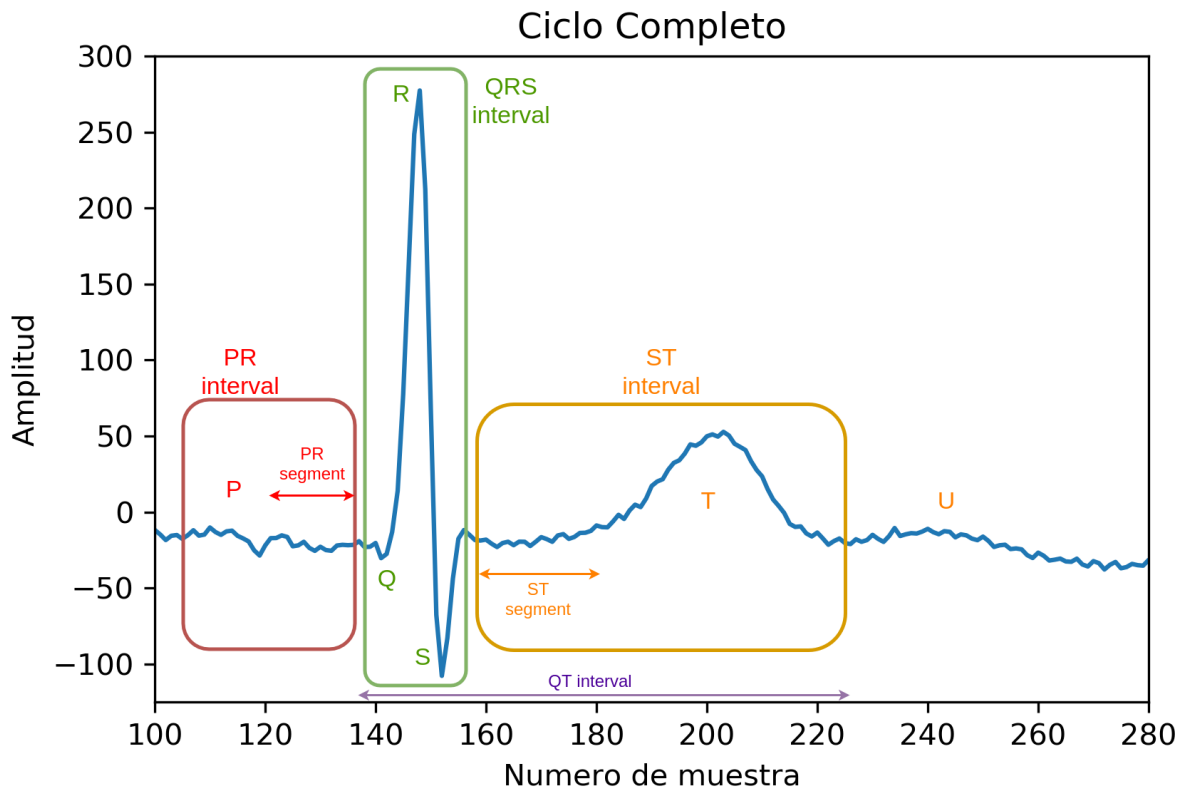
Para ubicar de forma manual los complejos QRS, utilice las muestras desde la 100 hasta la 280, con el fin de mostrar un 'ciclo' completo de la misma.

Se utilizara como frecuencia de muestreo a $f_s = 200Hz$, es decir, un periodo de muestreo $t_s = \frac{1}{200Hz} = 5ms$.



Se puede observar que el QRS equivale aproximadamente a 20 muestras (desde 140 a 160), por lo que teniendo en cuenta que la frecuencia de muestreo es de 200Hz, obtenemos que el mismo tiene una duración (aproximada) de $t_{QRS} = \frac{\#muestras}{f_s} = \frac{20Muestras}{200Muestras/s} = 100ms$.

Procedo entonces a ubicar manualmente las etapas del mismo:



Cabe aclarar, en especial en el PR interval, que no pude percibir visualmente un pico pronunciado en el mismo, estimo que se debe a que la imagen adjunta que use como referencia

(del enunciado), era una representación ideal de las etapas, mientras que en la muestra extraída intervienen factores de todo tipo que podrían distorsionar la medición.

No obstante, tanto en el QRS, como en el ST interval, los picos y depresiones Q,R,S y T se pueden apreciar claramente.

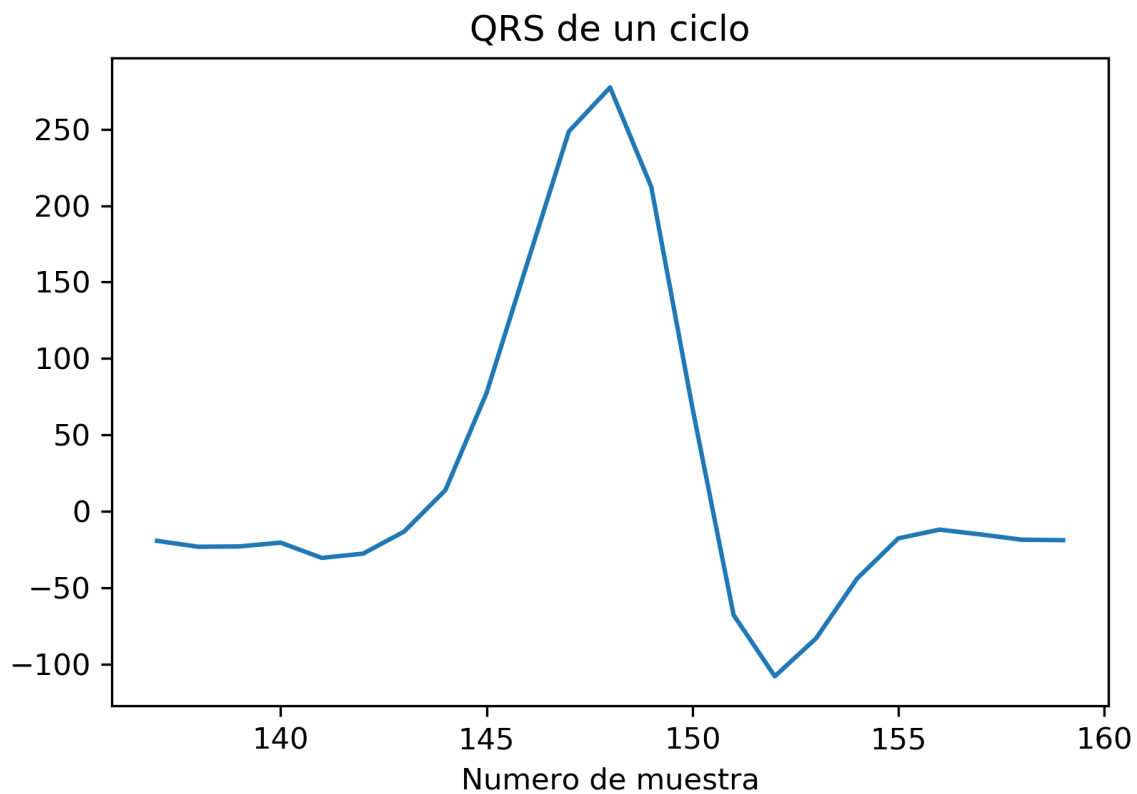
Finalmente, cabe aclarar que a la misma se le retiro la continua, por lo que se ve que el valor medio de la misma es cercano a 0 en la unidad correspondiente al eje de las ordenadas.

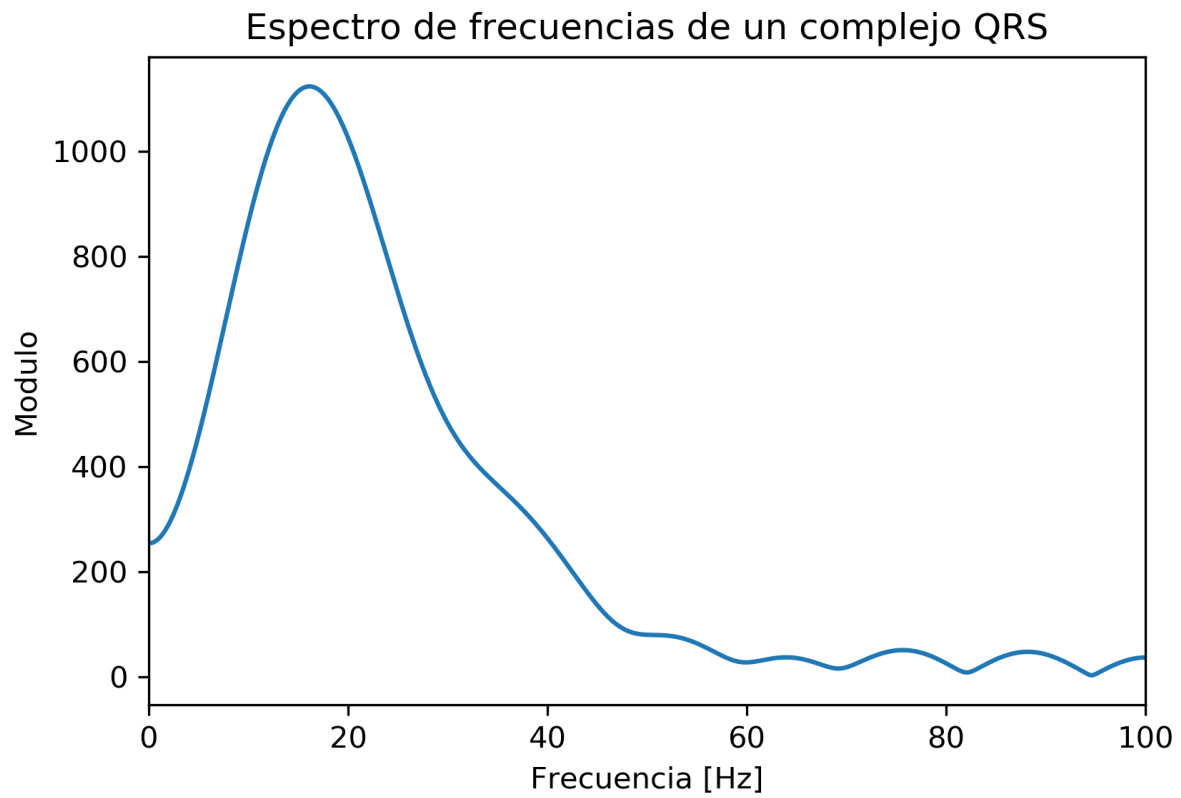
2. Ejercicio 2

”Muestre las características en frecuencia del complejo QRS.”

Solución: Para analizar las características en frecuencia del complejo QRS se transformo mediante la DFT (Discrete Fourier Transform) con el algoritmo de 2048 puntos.

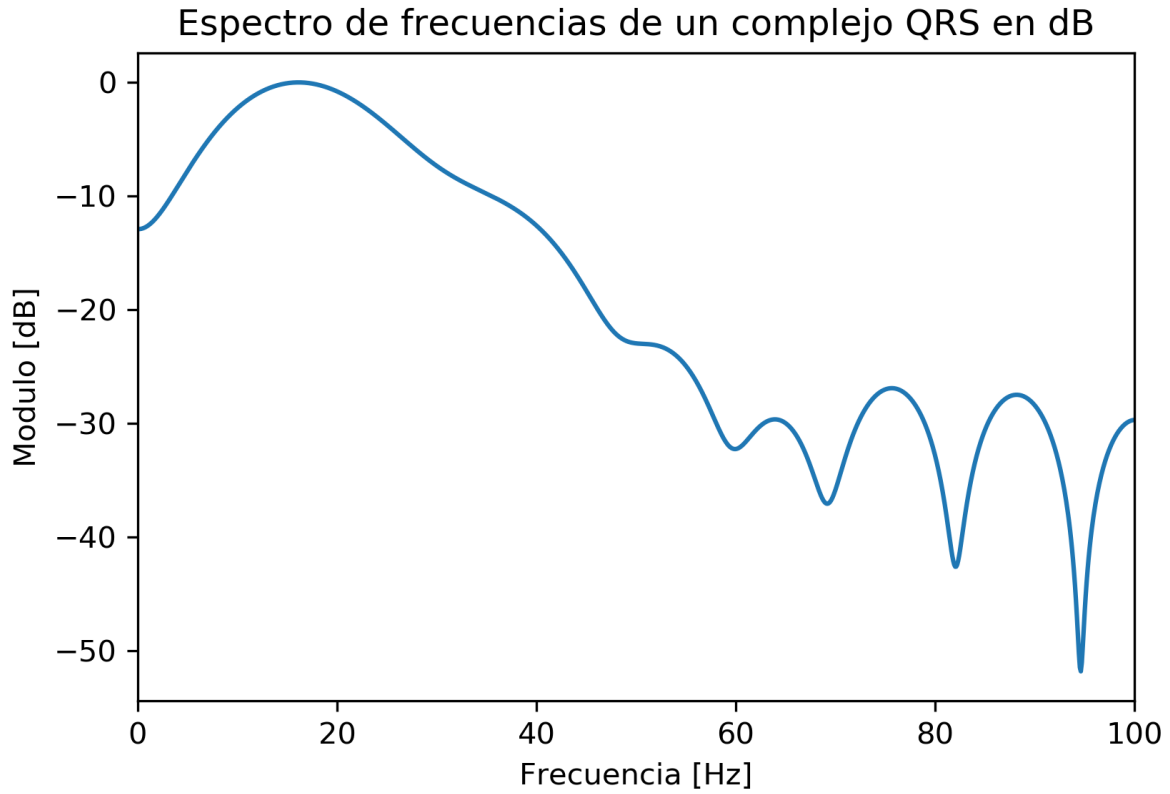
Se puede apreciar el rango de la señal al que se le realizó la transformada, y posteriormente la transformada de la misma.





Se puede apreciar en la misma una continua de aproximadamente 200.

Ahora veamos el eje de las ordenadas en dB:



Cabe aclarar que en el eje x de los graficos, que representa a $[\Omega]$, se muestra el intervalo desde $-\pi$ a π , que se corresponden con $-f_s/2 = -100Hz$ y $f_s/2 = 100Hz$, respectivamente.

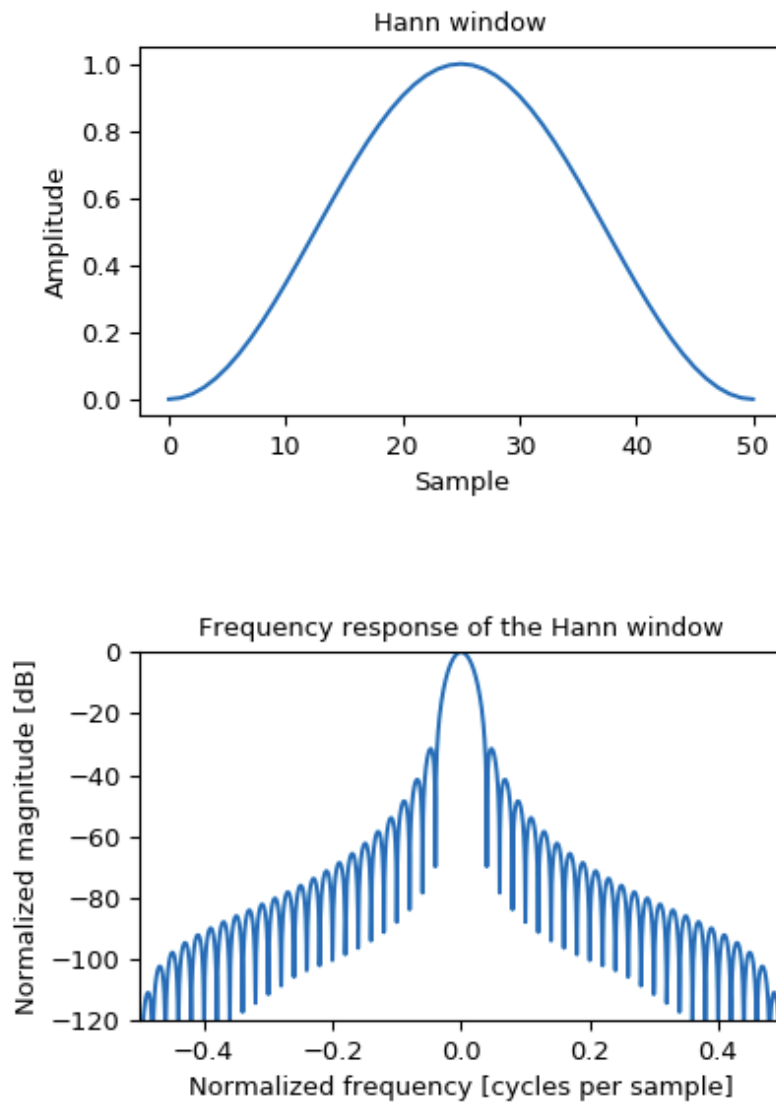
Es clara una elevación de la intensidad en el rango de frecuencias entre $0,25 * \frac{100Hz}{\pi} \approx 8Hz$ y $1 * \frac{100Hz}{\pi} \approx 32Hz$.

3. Ejercicio 3

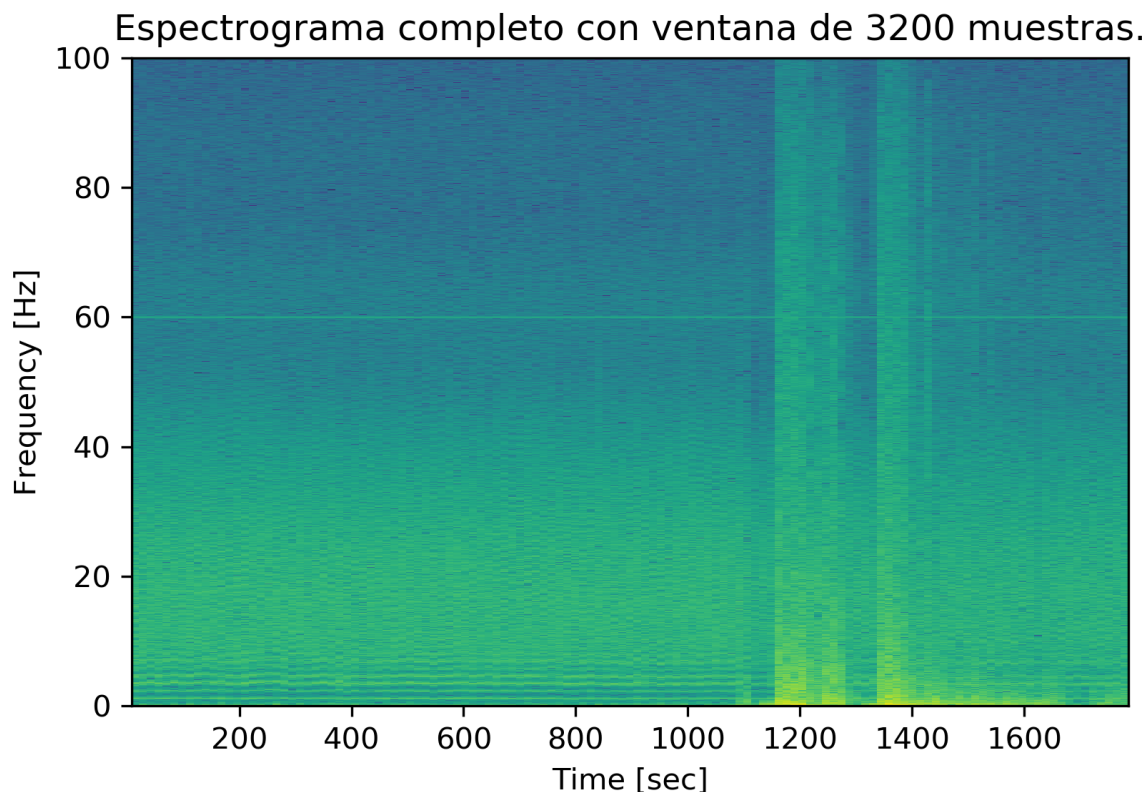
”Realice un espectrograma de la señal de electrocardiograma que permita visualizar las características en frecuencia de la onda QRS. Implementar el espectrograma de modo de observar las características de frecuencia descritas en el punto anterior.”

Solución: Se realizó el espectrograma de la señal completa. En este caso utilicé una ventana Hann de 3200 muestras mediante el comando (del modulo `scipy.signal`): `window = signal.hann(3200)`.

A continuación muestro un gráfico temporal y las características en frecuencia de la ventana utilizada:



El espectrograma de la señal se muestra a continuación:



Se puede observar esta analogía de 'principio de incertidumbre' entre la resolución en frecuencia y la resolución en tiempo: Cuanto mayor es la cantidad de puntos en la ventana, se puede observar una mayor resolución en frecuencia, y menor resolución en tiempo, debido a la superposición de muchos intervalos no disjuntos al realizar las transformadas sucesivas.

Veamos además que la ventana elegida (Hann) tiene características de leves altas frecuencias (suave). Ventanas mas suaves tienen un lóbulo principal mas 'ancho', ya que se pueden pensar como varias convoluciones de cuadradas (en frecuencia es elevar a alguna potencia la *sinc*) con lo que perderían resolución en frecuencia, pero tienen la ventaja de tener lóbulos secundarios que son de menor preponderancia, reduciendo el ruido.

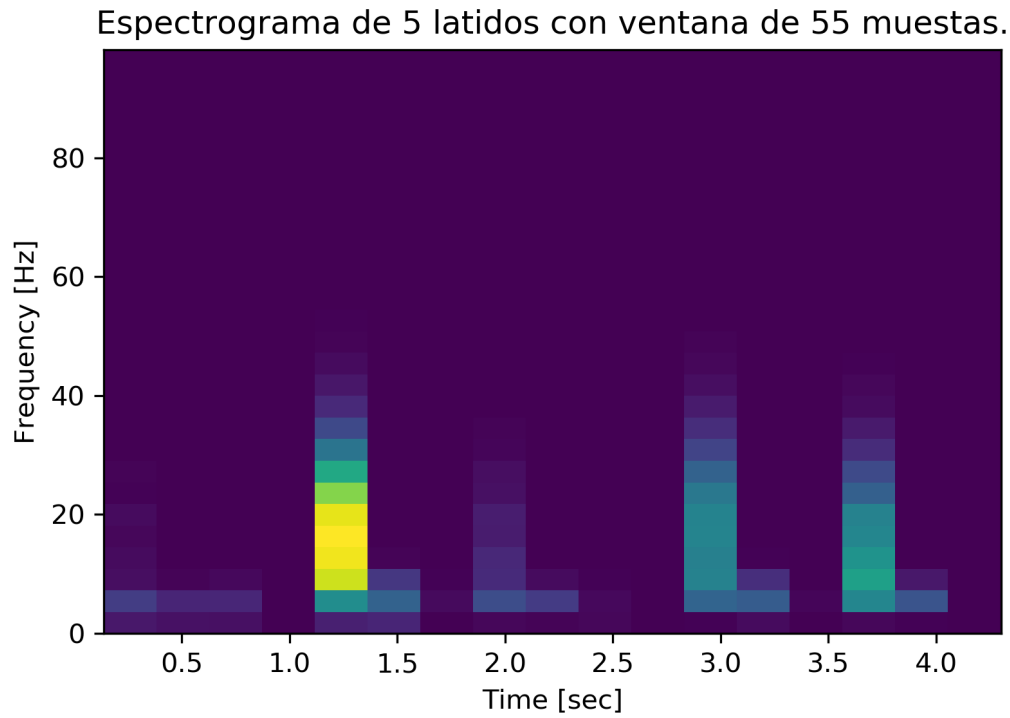
En caso de ventanas similares a la cuadrada (por ejemplo la Tukey que fue brindada como ejemplo, sucede lo contrario) tendrá mejor resolución en frecuencia, a costo de introducir ruido. Esto se debe a los lóbulos secundarios de la *sinc*, que en este caso serían mas significativos que en otras ventanas.

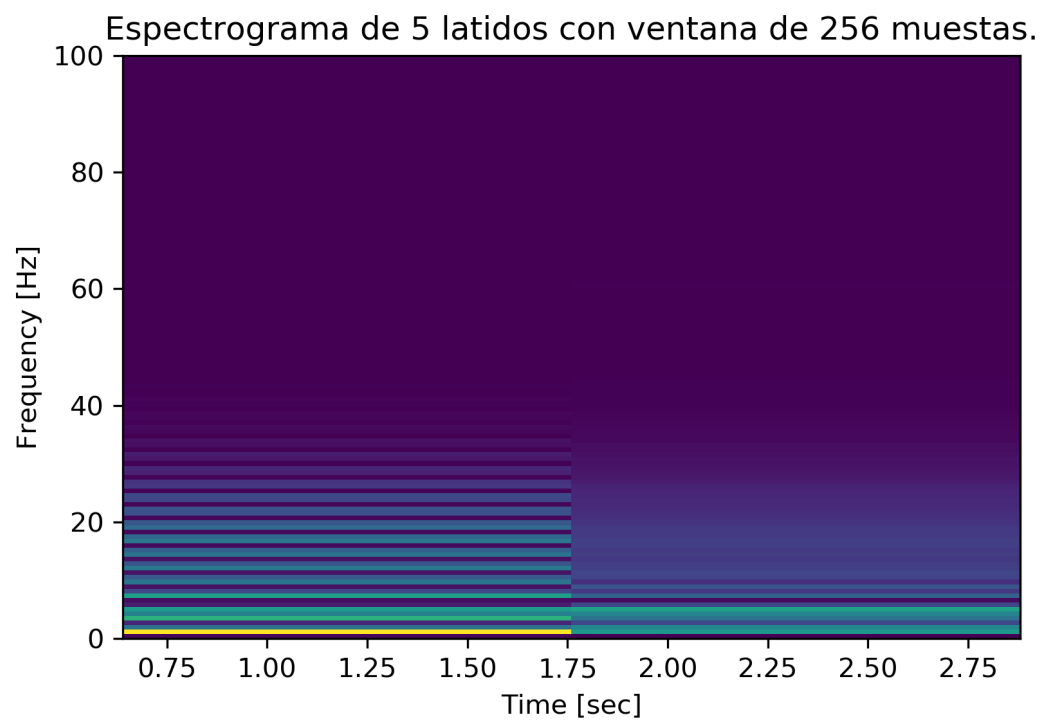
En nuestro caso es de utilidad tener una buena resolución en frecuencia, entonces sería razonable elegir una función suave y con un ancho considerable en tiempo como la Hann, debido a que con ella serán mas notorias los espectros de las señales útiles y los ruidos.

- Dejo a continuación también el espectrograma de 5 latidos, utilizando la misma ventana

(Hann), pero con 55 y 256 muestras respectivamente, pero ya la resolución en frecuencia es pobre.

En estos ejemplos se nota claramente como al utilizar menos puntos (55) , se distinguen los picos de frecuencia de los 5 latidos. En cambio, una subida de los mismos aumenta la resolución en frecuencia pero ya no se distinguen en tiempo los mismos.

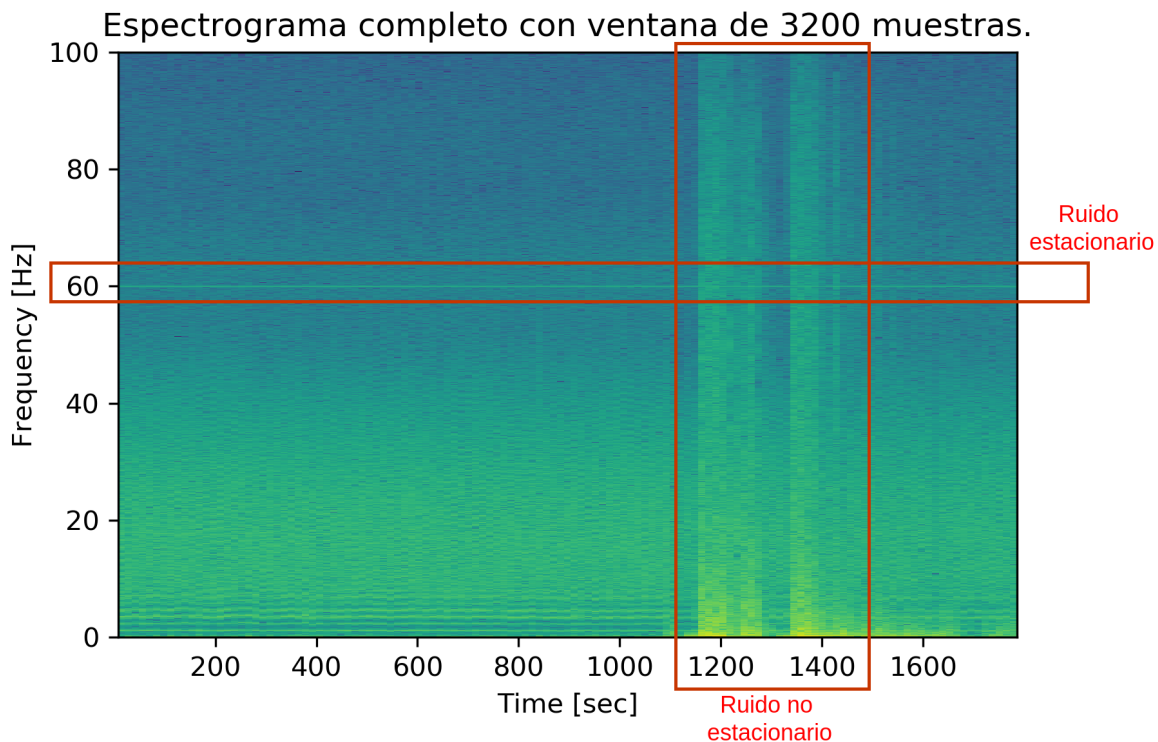




4. Ejercicio 4

”Determine qué tipo de ruidos están presentes en la señal de ECG. En primer lugar describa los ruidos estacionarios que encuentra, utilizando tanto gráficos en tiempo como en frecuencia. Luego determine qué porciones de señal de ECG tienen sumados ruidos no estacionarios. Explique las posibles fuentes de cada ruido.”

Solución: En la figura del ECG completo del Ejercicio 3 pueden observarse:



- Ruidos estacionarios:
 - Componente a 60Hz: Esto se evidencia a lo largo de todo el espectrograma, y estimo que la causa es de la red eléctrica (en este caso de USA) que a diferencia de Argentina (50Hz) utiliza dicha frecuencia. Por supuesto es un ruido indeseado y debería ser eliminado.
- Ruidos no estacionarios:
 - Se puede observar que entre las muestras 1100 y 1400 segundos se encuentran ruidos de altas frecuencias. Estimo que debido a que es de corta duración (aproximadamente 5 min), pudo haber sido causado por algún desperfecto en los materiales o movimiento del paciente.
- Frecuencias útiles:

- Componente de aproximadamente 1Hz: Es la frecuencia fundamental del latido del corazón (60BPM en reposo), que es acompañada razonablemente de algunos de sus armónicos.

5. Ejercicio 5

”Obtenga en forma analítica y utilizando funciones de MATLAB el diagrama de polos y ceros, la respuesta en frecuencia y la respuesta al impulso del filtro $H_L(z)$. Calcule el retardo producido por el filtro. Analice causalidad, estabilidad y el efecto producido sobre la señal de ECG.”

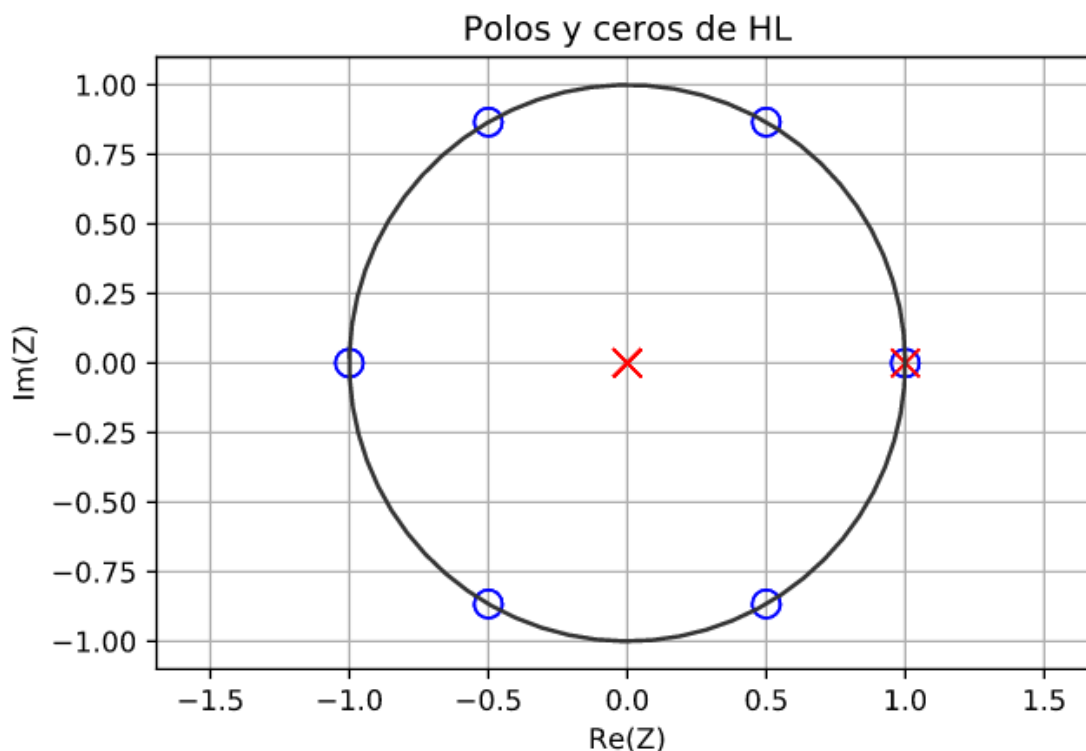
Solución:

Tenemos nuestro filtro pasa bajos:

$$H_L(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2}$$

5.1. Polos y Ceros de H_L

Se muestra a continuación el diagrama de polos y ceros, realizado mediante el comando roots de numpy, y el ploteo correspondiente:



Podemos notar lo siguiente analíticamente:

- Polos: Tenemos un polo doble en 0(cero) y en 1(un) (simples pero al elevar al cuadrado son dobles)
- Ceros: Tenemos un polo doble en cada una de las raíces sextas de la unidad ($z_i = e^{2\pi i/6}$, con $i = 1, 2, \dots, 6$) (simples por $1 - z^{-6}$) pero al elevar al cuadrado son dobles)

Veamos que la transformada de Laplace existe en todo punto (excepto en 0): Podemos ver que en $z = 1$ tenemos la misma cantidad de ceros que de polos, por lo tanto la integral no diverge.

Analizando ahora la distribución de los ceros (donde la transformada tiene atenuación completa) al movernos por el círculo unitario, se puede observar que desde $z = 1$ hasta $z = e^{j\pi/3}$ no se atenúa la señal, es decir, llevándolo al continuo (con $f_s/2 = 100Hz \equiv \pi$) :

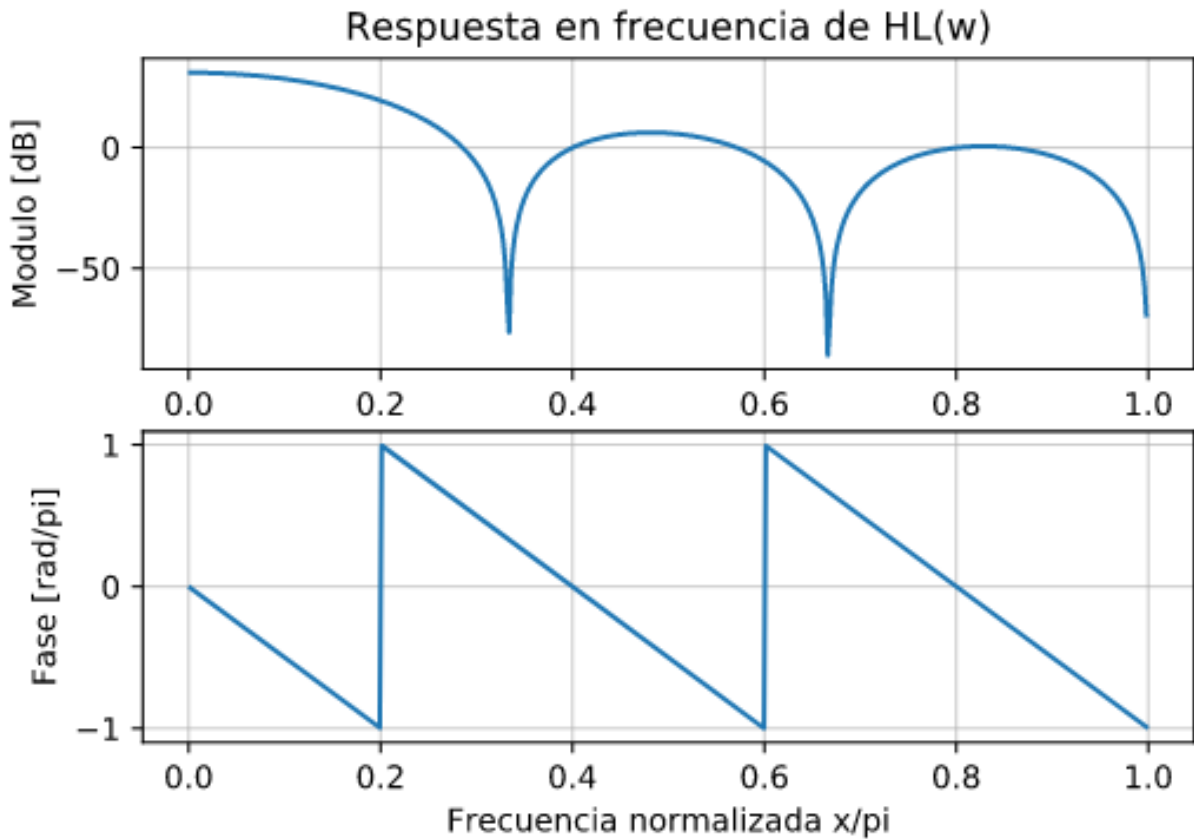
$f_1 = \pm \frac{\pi}{3} \frac{100Hz}{\pi} = \pm 33,3Hz$ es la primer frecuencia en la que tenemos una atenuación completa.

Analogamente tenemos:

$f_2 = \pm \frac{2\pi}{3} \frac{100Hz}{\pi} = \pm 66,6Hz$ es la segunda frecuencia en la que tenemos una atenuación completa.

$f_3 = \pi \frac{100Hz}{\pi} = 100Hz$ es la tercer frecuencia en la que tenemos una atenuación completa.

5.2. Respuesta en Frecuencia de H_L



Modulo: Se puede observar en la figura que los f_1 , f_2 y f_3 respectivamente a $\frac{1}{3} \equiv 33,3Hz$, $\frac{2}{3} \equiv 66,6Hz$ y $1 \equiv 100Hz$ atenuados completamente (orden de -80dB), mientras que las frecuencias bajas (entre 0Hz y aprox. 25Hz) son amplificadas.

Fase: A la luz de los gráficos, se observa una fase lineal, por lo tanto por lo visto en clase, debería provenir de una respuesta al impulso simétrica.

5.3. Respuesta al impulso de HL



5.3.1. Retardo HL

De las clases prácticas se conocía que:

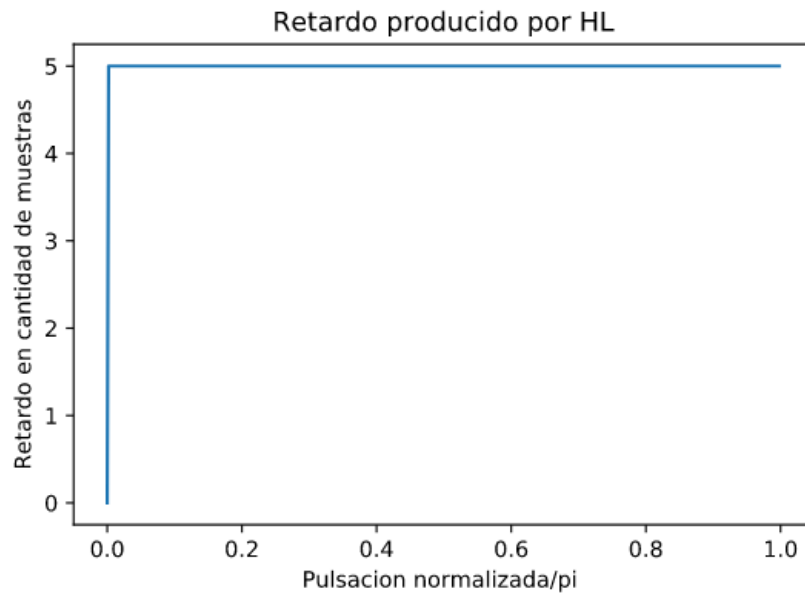
' Si el filtro es simétrico, cuando la fase es plotada obtienes una recta , y además'

$$retardo(\tau) = -\frac{dH(\Omega)}{d(\Omega)}$$

Es decir el negativo de la pendiente de la recta. En este caso, en el gráfico de la fase en frecuencia, la misma tiene pendiente $(-1 - 1)/(0,6 - 0,4) = -5$, entonces esperaríamos un retardo de 5.

Además, cabe notar que nuestra respuesta al impulso está centrada en 5, como debería ocurrir para que la fase sea lineal, y el retardo es el centro.

Confirmamos (nuevamente) mediante el plot del retardo, que nuestro retardo es de 5 muestras (función `signal.group_delay` de scipy):



5.4. Respuesta al escalón de HL

Adjunto la respuesta al escalon de H_L (notemos que su 'derivada' discreta coincide con la respuesta al impulso) :



5.5. Ecuacion en diferencias

La ecuación en diferencias que puede despejarse de la transferencia misma (utilizando cuadrado de un binomio y dejando convenientemente expresados el numerador y denominador:

$$H_L(z) = \frac{1 - 2z^{-6} + z^{-12}}{1 - 2z^{-1} + z^{-2}}$$

Por lo tanto los coeficientes coinciden con los de la ecuación en diferencias, y los 'offset' de los términos con la potencias de la expresión del filtro:

$$y(n) - 2y(n-1) + y(n-2) = x(n) - 2x(n-6) + x(n-12)$$

- Se ve que claramente el filtro es causal: La ecuacion no depende de valores posteriores a n , sino exclusivamente de anteriores.
- Viendo que la ROC termina en infinito: Condiciones iniciales de Reposo.
- Como en cero hay singularidad, no puede afirmarse que hay condiciones finales de reposo.

6. Ejercicio 6

”Obtenga en forma analítica y utilizando funciones de MATLAB el diagrama de polos y ceros, la respuesta en frecuencia y la respuesta al impulso del filtro $H_H(z)$. Calcule el retardo producido por el filtro. Analice causalidad, estabilidad y el efecto producido sobre la señal de salida de la etapa anterior. Obtenga además la ecuación en diferencias que caracteriza al filtro.”

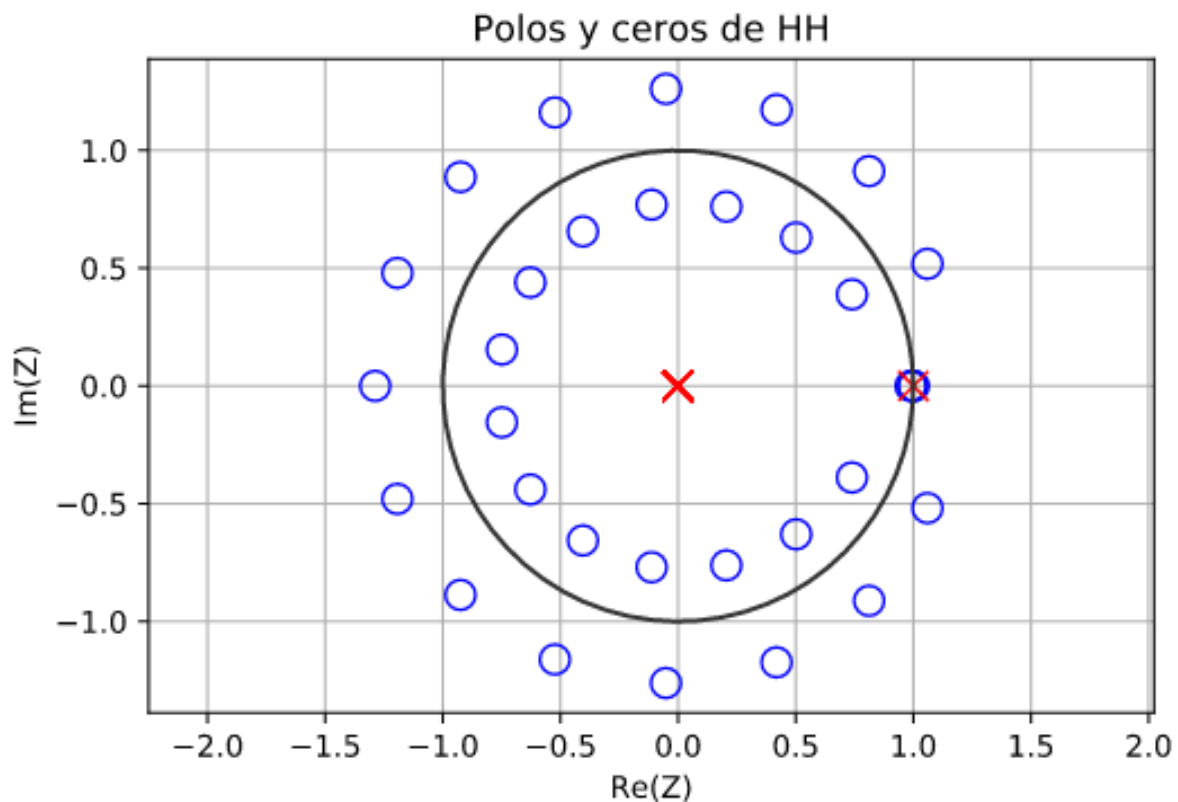
Solución:

Tenemos nuestro filtro pasa altos:

$$H_H(z) = \frac{-\frac{1}{32} + z^{-16} - z^{-17} + \frac{z^{-32}}{32}}{1 - z^{-1}}$$

6.1. Polos y Ceros de H_H

Se muestra a continuación el diagrama de polos y ceros, realizado mediante el comando roots de numpy, y el ploteo correspondiente:



Podemos notar lo siguiente analíticamente:

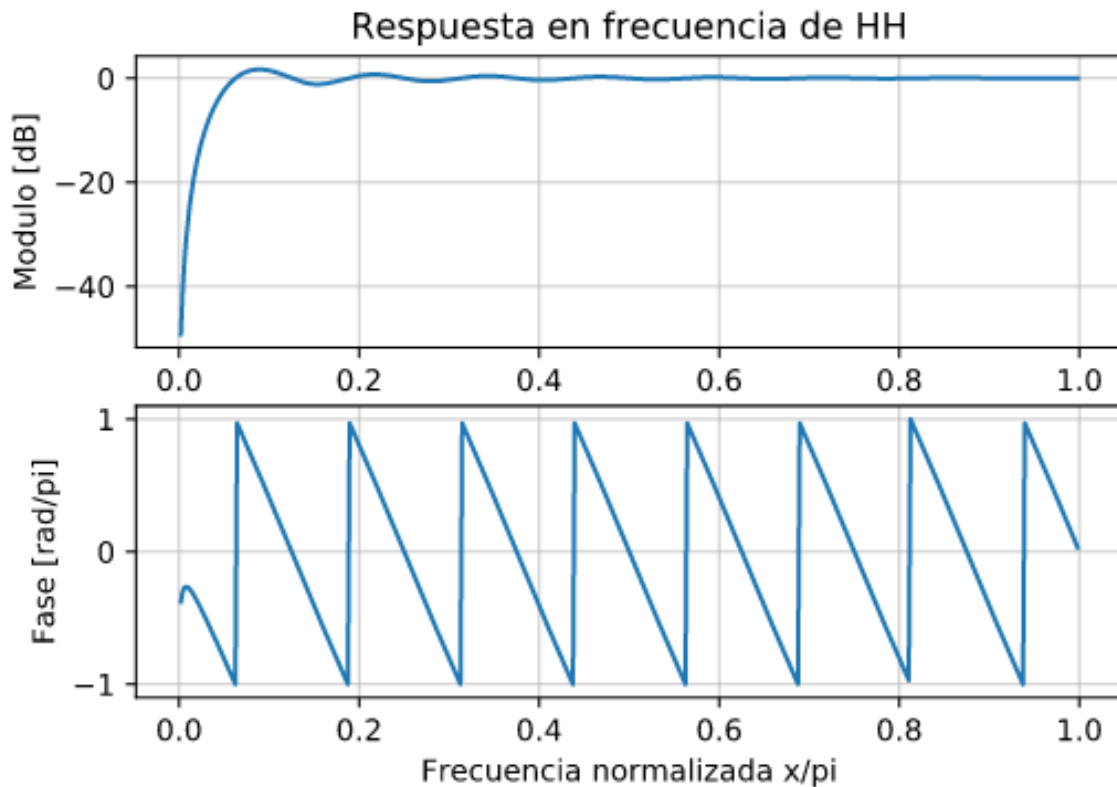
- Polos: Polo en 0 y polo en 1 (simples).

- Ceros: Se pueden observar en la figura (obs: Hay varios ceros en $z = 1$).

Veamos que la transformada de Laplace existe en todo punto (excepto en 0): Podemos ver que en $z = 1$ tenemos mas cantidad de ceros que de polos, por lo tanto la integral no diverge.

Analizando ahora la distribución de los ceros (donde la transformada tiene atenuación completa) al movernos por el círculo unitario, se puede observar que en $z = 1$ hay ceros, por lo que la continua es filtrada completamente (ese Ω se corresponde con 0Hz).

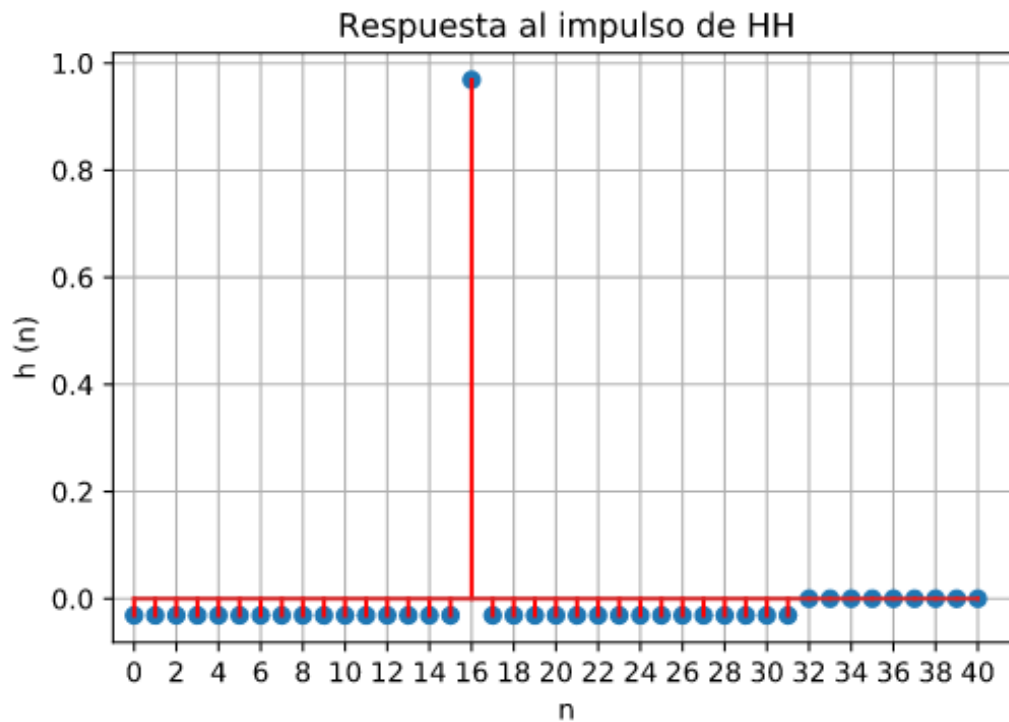
6.2. Respuesta en Frecuencia de H_H



Modulo: Se puede observar en la figura que las frecuencias hasta $0,05\pi \equiv 5Hz$ son atenuadas, y luego la atenuación es prácticamente nula (oscila entre 0dB).

Fase: A la luz de los gráficos, se observa una fase lineal, por lo tanto por lo visto en clase, debería provenir de una respuesta al impulso simétrica.

6.3. Respuesta al impulso de HH

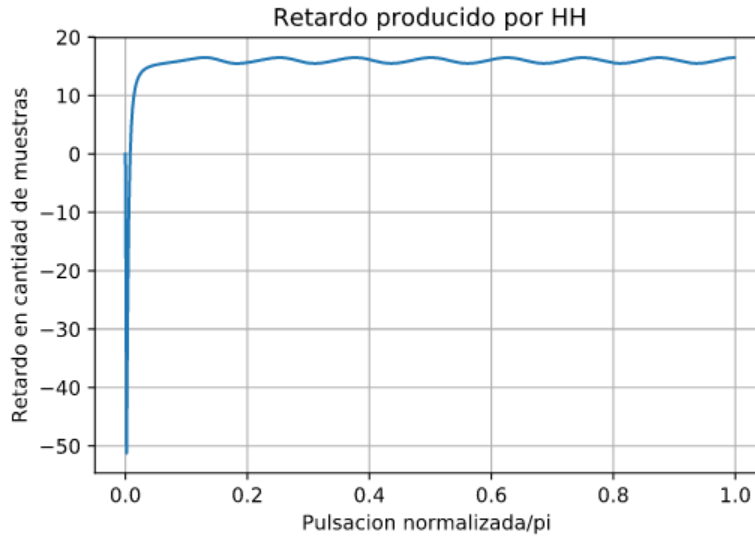


6.3.1. Retardo HH

Utilizando el mismo argumento que en H_L de la pendiente del grafico de la fase:
 $retardo(\tau) \approx -(-1 - 1)/(0,3 - 0,18) = 16,66$ muestras.

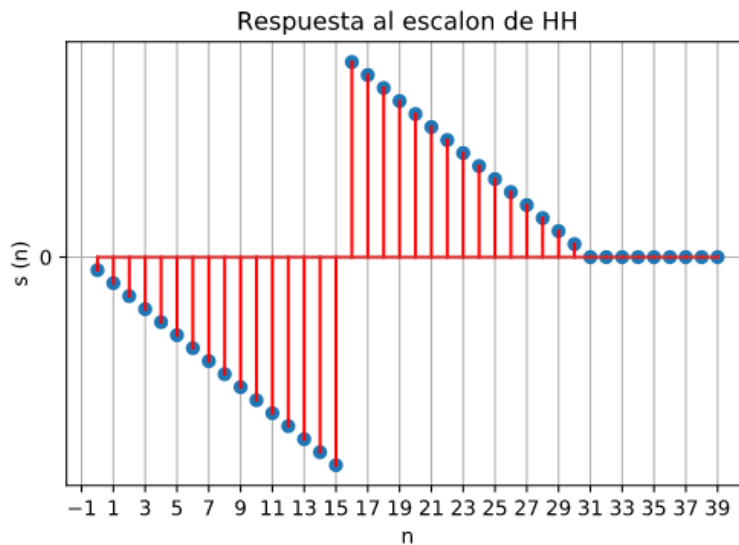
Ademas, cabe notar que nuestra respuesta al impulso está centrada en 16, como debería ocurrir para que la fase sea lineal, y el retardo es el centro.

Confirmamos (nuevamente) mediante el plot del retardo, que nuestro retardo es de 16 muestras (funcion `signal.group_delay` de `scipy`), vemos que oscila levemente en 16, debido a la leve perdida de linealidad del grafico de la fase:



6.4. Respuesta al escalón de HH

Adjunto la respuesta al escalon de H_H (notemos que su 'derivada' discreta coincide con la respuesta al impulso) :



6.5. Ecuacion en diferencias

La ecuación en diferencias que puede despejarse de la transferencia misma:

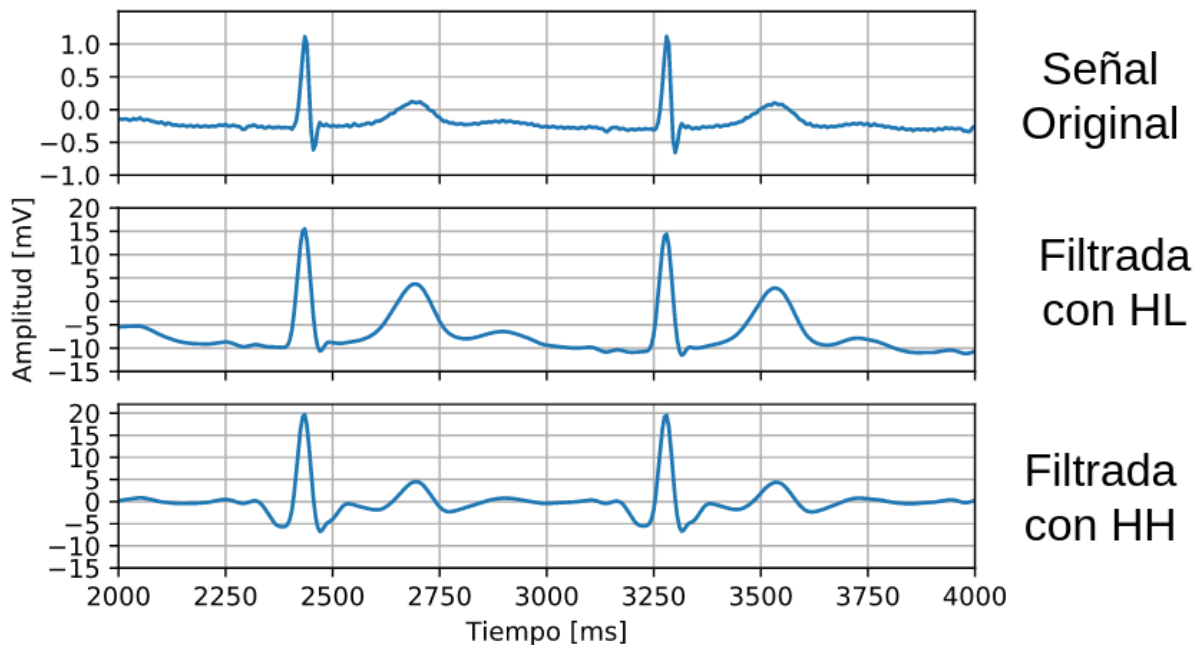
$$H_H(z) = \frac{-\frac{1}{32} + z^{-16} - z^{-17} + \frac{z^{-32}}{32}}{1 - z^{-1}}$$

Por lo tanto los coeficientes coinciden con los de la ecuación en diferencias, y los 'offset' de los términos con la potencias de la expresión del filtro:

$$y(n) - y(n-1) = -\frac{1}{32}x(n) + x(n-16) - x(n-17) + \frac{1}{32}x(n-32)$$

- Se ve que claramente el filtro es causal: La ecuación no depende de valores posteriores a n , sino exclusivamente de anteriores.
- Viendo que la ROC termina en infinito y no hay singularidades ahí: Condiciones iniciales de Reposo.
- Como en cero hay singularidad, no puede afirmarse que hay condiciones finales de reposo.
- Estabilidad: Notemos que finitos valores de la respuesta al impulso son diferentes a 0 (los de valor abs menor a 32), y al efectuar su suma no diverge, por lo que el sistema es estable.

6.6. Filtros aplicados a la señal



Vemos como en la primer pasada, el pasa bajos se ocupa de filtrar las altas frecuencias, suavizando la señal (por pérdida de armónicos altos). En la segunda pasada, como H_H atenuaba principalmente frecuencias en $[0Hz, 5Hz]$, vemos la pérdida de la continua, por lo que nuestra señal debería tener ahora media 0.

7. Ejercicio 7

”Obtenga versiones FIR de los filtros $H_L(z)$ y $H_H(z)$. Discuta ventajas y desventajas de ambas implementaciones.”

Solución: Citando a Wikipedia: *’FIR es un acrónimo en inglés para Finite Impulse Response o Respuesta finita al impulso. Se trata de un tipo de filtros digitales cuya respuesta a una señal impulso como entrada tendrá un número finito de términos no nulos... Estos filtros tienen todos los polos en el origen, por lo que son estables. Los ceros se presentan en pares de recíprocos si el filtro se diseña para tener fase lineal.’*

• Ambos filtros, poseen esta cualidad (ya que en $z = 1$ tenemos mas ceros que polos en ambos casos), por lo tanto existen versiones FIR de los mismos.

La salida puede expresarse como la convolución de la señal de entrada $x(n)$ con la respuesta al impulso $h(n)$:

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k]$$

Y aplicando transformada Z a dicha expresión:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)}$$

En nuestros casos, mirando los graficos de las respuestas al impulso en las figuras correspondientes:

$$H_L(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3} + 5z^{-4} + 6z^{-5} + 5z^{-6} + 4z^{-7} + 3z^{-8} + 2z^{-9} + 1z^{-10}$$

$$H_H(z) = \frac{31}{32}z^{-16} + \sum_{k=0, k \neq 16}^{32} -\frac{1}{32}z^{-k}$$

7.1. Ventajas y desventajas

Los filtros FIR tienen la gran ventaja de que pueden diseñarse para ser de fase lineal, lo cual hace que presenten ciertas propiedades en la simetría de los coeficientes. Este tipo de filtros tiene especial interés en aplicaciones de audio. Además son siempre estables.

Por el contrario también tienen la desventaja de necesitar un orden mayor respecto a los filtros IIR para cumplir las mismas características. Esto se traduce en un mayor gasto computacional.

8. Ejercicio 8

”Obtenga un sistema de tiempo discreto LTI y causal $H_d(z)$ que permita aproximar la derivada de la señal de ECG. Justifique para que frecuencias es válida dicha aproximación basándose en la respuesta en frecuencia del sistema obtenido. Calcule el retardo producido por el filtro y obtenga la derivada de la señal de salida de la etapa anterior.”

Solución:

Notemos primero que de la aproximación que utilizaremos para la derivada:

$$\frac{dx(t)}{dt} \approx \frac{1}{8h}(-x(t-2h) - 2x(t-h) + 2x(t+h) + x(t+2h))$$

Como no tengo costos computacionales, desearía poder obtener la mejor aproximación posible disminuyendo el paso. Es por eso que elegí $h = 1$:

$$\frac{dx(t)}{dt} \approx \frac{1}{8}(-x(t-2) - 2x(t-1) + 2x(t+1) + x(t+2))$$

Aplicando la transformada Z a la aproximación de la derivada (ya con $h=1$):

$$H_D(z) = \frac{1}{8}(-z^{-2} - 2z^{-1} + 2z^1 + z^2)$$

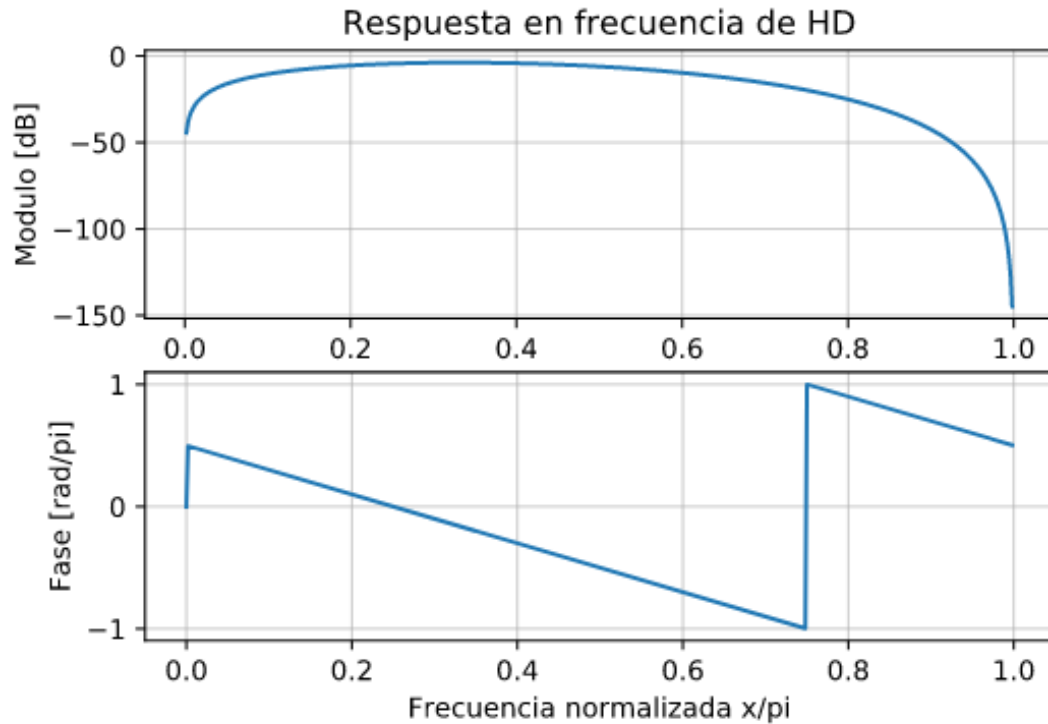
Pero el problema es que no es causal, ya que depende de términos posteriores a n (que vienen dados por los términos $2z^1 + z^2$ de la transformada Z).

Entonces para poder deshacernos de esos términos, sería conveniente multiplicar a la transformada H_D por z^{-2} , ya que con esto nos garantizaríamos la causalidad del mismo.

Multiplicar la transferencia por z^{-2} equivale a agregar un retardo de 2 muestras, que corregiremos una vez aplicado el filtro para realmente calcular la derivada como nos la piden.

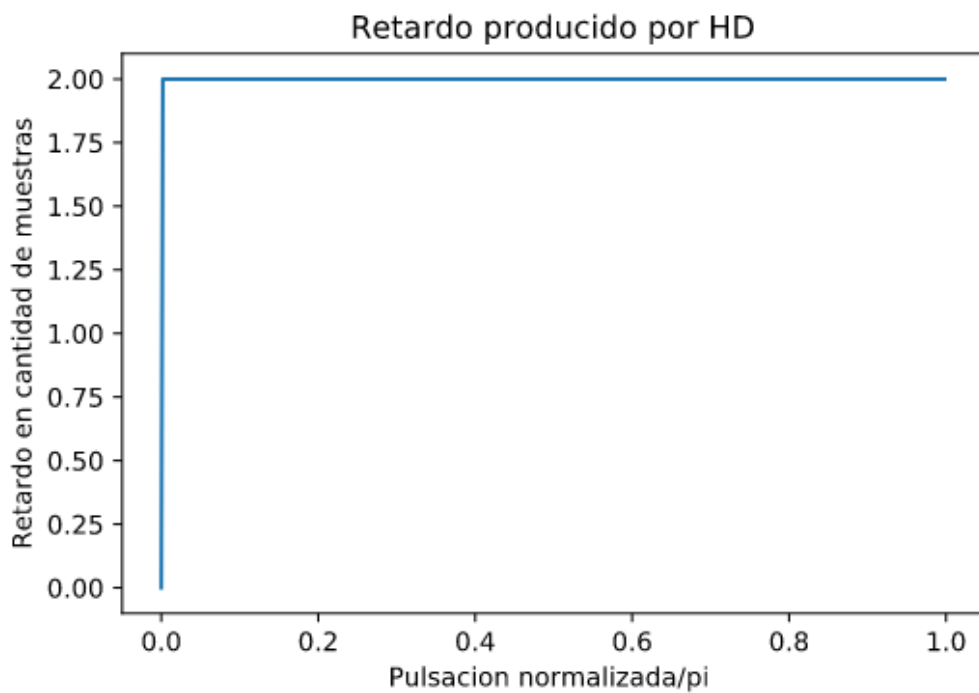
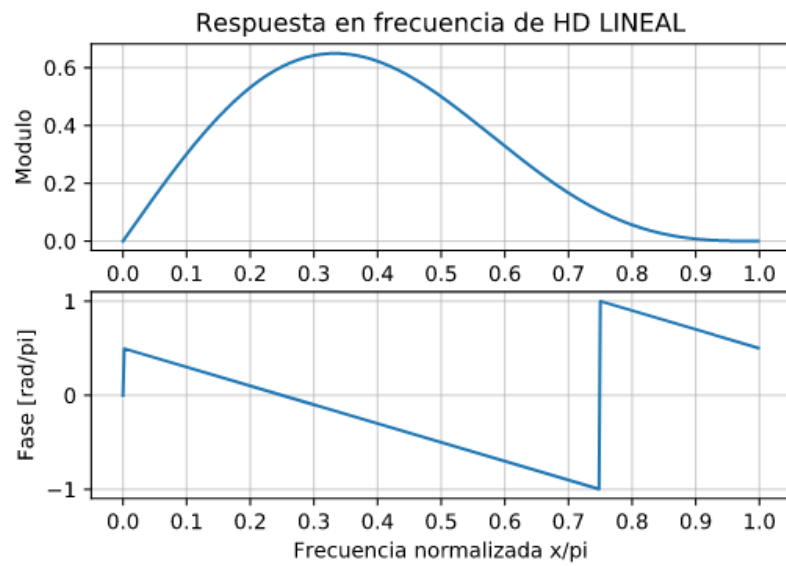
$$H_D(z) = \frac{1}{8}(-z^{-4} - 2z^{-3} + 2z^{-1} + 1)$$

Adjunto la respuesta en frecuencia del mismo:

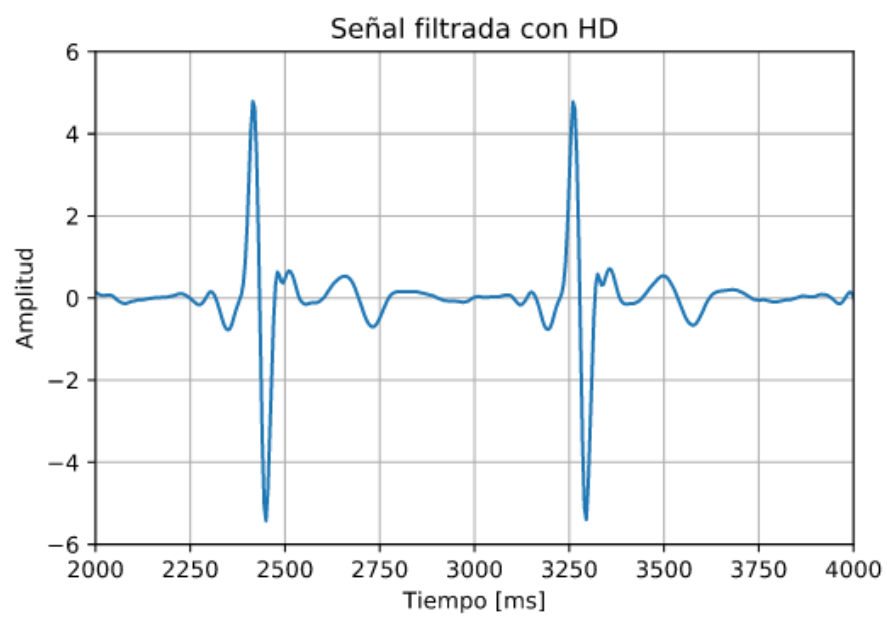


Modulo: Veamos que el mismo atenúa (maximo en -3.7481 dB, que se da en 0.3 de la frecuencia normalizada, es decir 30Hz), por fuera de estos valores, la señal se ve fuertemente atenuada (del orden de los -50db a -150dB). Veamos en la escala lineal que nuestro diferenciador se parece a una recta (diferenciador ideal), para los valores hasta 30Hz. Luego de esa frecuencia, no se obtienen las características deseadas

Fase: A la luz de los gráficos, se observa una fase lineal, por lo tanto por lo visto en clase, debería provenir de una respuesta al impulso simétrica.



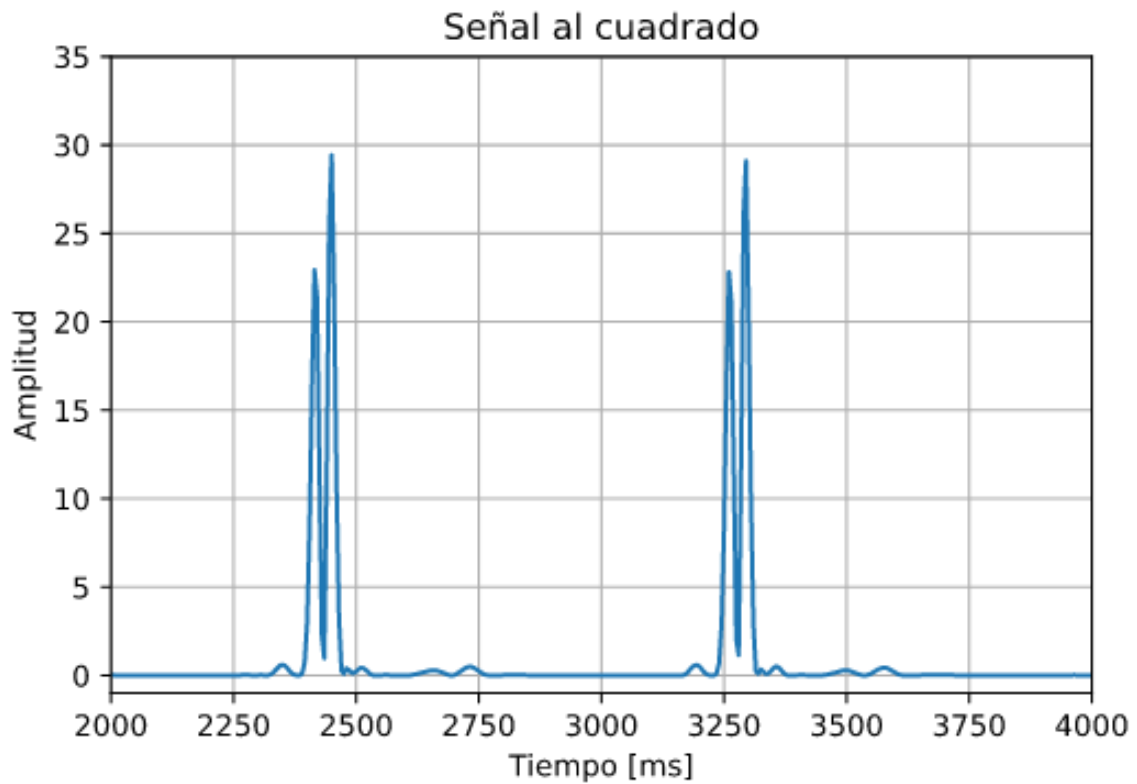
Se ve claramente, como suponíamos teóricamente, que el retardo es de 2. También se puede ver como el negativo de la pendiente en el gráfico de la fase de la respuesta en frecuencia : $(-1-0.5)/(0.75-0) = -2$.



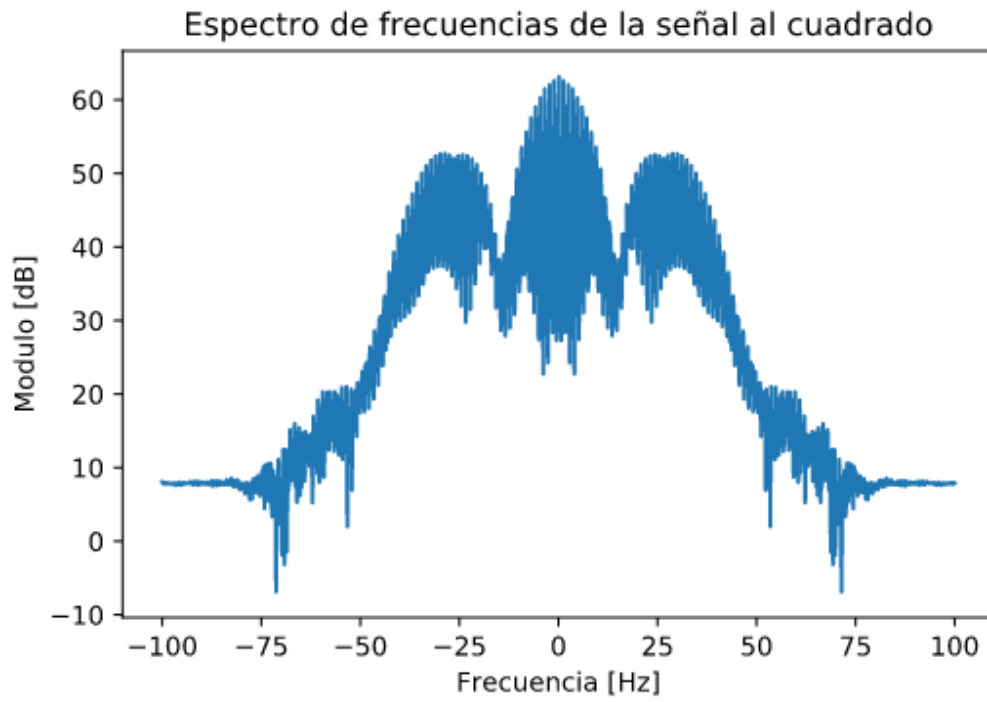
9. Ejercicio 9

”Obtenga el cuadrado de la señal de salida de la etapa de diferenciación y muestre el efecto producido en tiempo y en frecuencia.”

Solución:



Claramente elevar al cuadrado la señal enaltece los complejos QRS, y atenúa el resto de la señal, por lo que ahora los mismos pueden ser mas faciles de identificar.



El producto de las señales en tiempo (en este caso la que venimos filtrando consigo misma) se podría interpretar como una convolución consigo misma en frecuencia.

10. Ejercicio 10

” Obtenga la respuesta en frecuencia del sistema integrador y analice el retardo generado por el mismo. Ajuste el valor de N de modo tal de lograr resaltar la posición temporal de los complejos QRS. Recuerde que valores pequeños de N resultan en una gran cantidad de máximos pero valores demasiado grandes pueden producir la fusión del complejo QRS con el resto de las ondas existentes en el periodo.”

Solución:

Escribimos primero la expresión del integrador:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k]$$

Si aplicamos transformada Z a ambos lados:

$$Y[z] = \frac{1}{N} \sum_{k=0}^{N-1} X[z]z^{-k}$$

Sacamos factor comun $X[z]$ y usamos serie geo:

$$Y[z] = \frac{1}{N} X[z] \frac{1 - z^{-N}}{1 - z^{-1}}$$

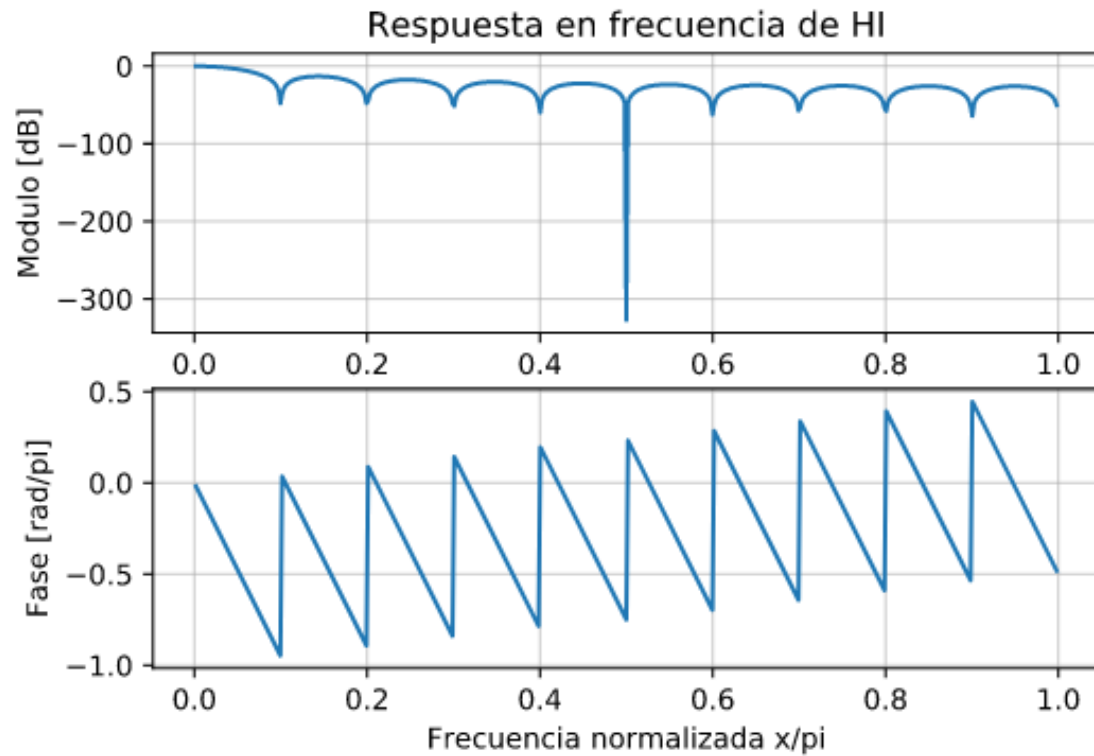
Por ende:

$$H[z] = \frac{Y[z]}{X[z]} = \frac{1}{N} \frac{1 - z^{-N}}{1 - z^{-1}}$$

Veamos que el N tiene que cumplir que nos cubra el ancho de los complejos QRS (para no sumar parcialmente los mismos y llevar a errores).

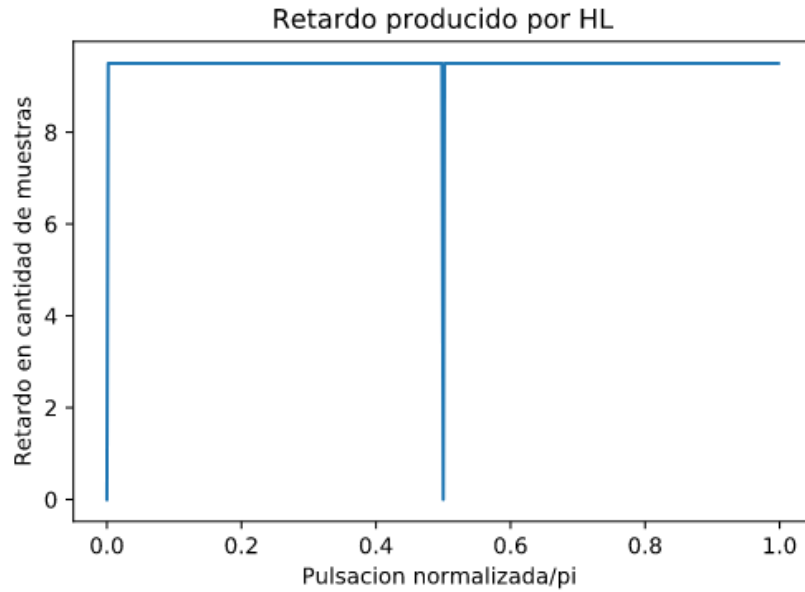
Del ejercicio 1 notamos que estos tienen aproximadamente un ancho de 20muestras (por entonces de 140 a 160). Suponiendo que los filtros de preprocesamiento no afectan demasiado este ancho, deberíamos garantizar por seguridad que al menos, $N \geq 20$. Como no queremos acumular demasiadas muestras y perder referencia temporal de las mismas, usaré $N = 20$.

Adjunto la respuesta en frecuencia del filtro H_I :



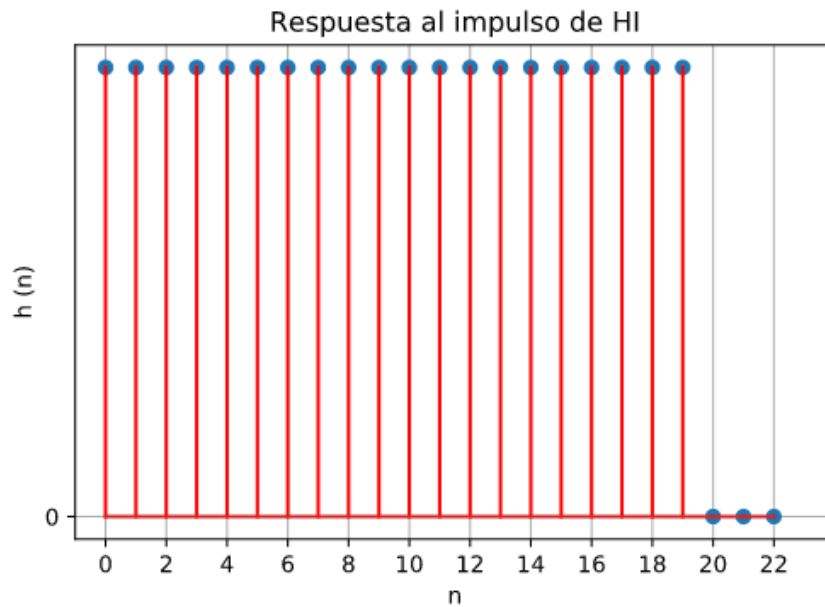
Modulo: Se puede observar en la figura que los f_1 , f_2 y f_3 respectivamente a $\frac{1}{3} \equiv 33,3Hz$, $\frac{2}{3} \equiv 66,6Hz$ y $1 \equiv 100Hz$ atenuados completamente (orden de -80dB), mientras que las frecuencias bajas (entre 0Hz y aprox. 25Hz) son amplificadas.

Fase: A la luz de los gráficos, se observa una fase lineal, por lo tanto por lo visto en clase, debería provenir de una respuesta al impulso simétrica.

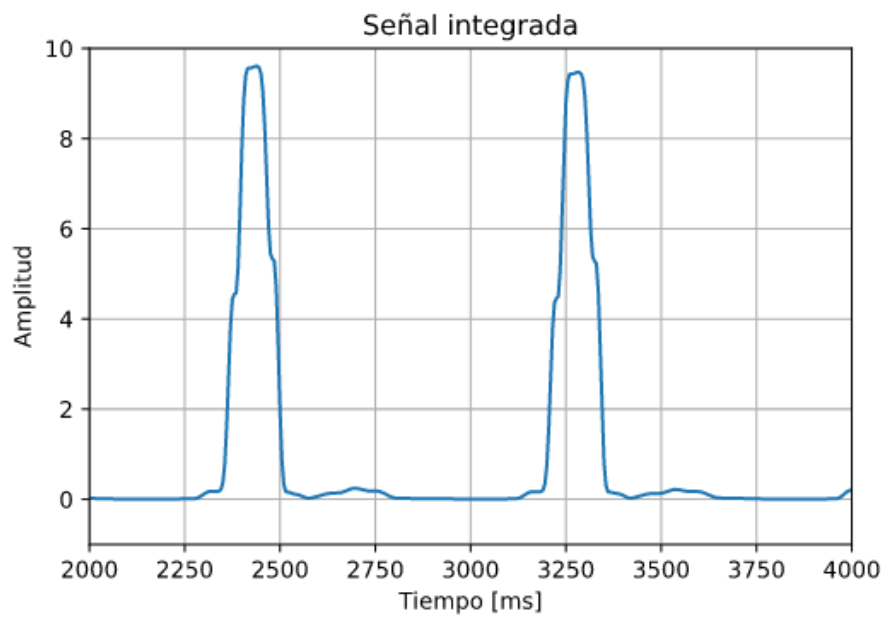


Utilizando el mismo argumento que en H_L de la pendiente del gráfico de la fase:
 $retardo(\tau) \approx -(-0,5 - 0,5)/(1 - 0,9) = 10$ muestras.

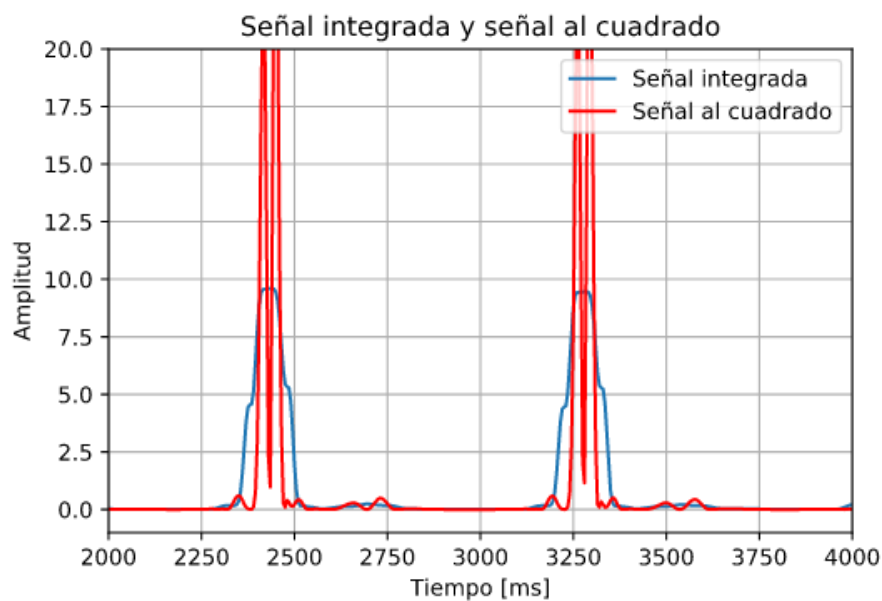
Estimo que esto sucede en base a la simetria de la respuesta al impulso del filtro en $\frac{N}{2}$, con todas las amplitudes iguales a $0,05 = \frac{1}{20}$



Por ultimo agrego adjunto el gráfico de la respuesta luego de la integración, y a continuación una comparación entre la señal antes (cuando sale de ser elevada al cuadrado) y luego de HI :



Se puede ver con claridad los complejos QRS, y se unifico el 'doble pico' apreciable luego de la salida de elevar al cuadrado la derivada.



11. Ejercicio 11 (opcional)

Implemente un algoritmo de detección automática de los complejos QRS utilizando la señal de salida de la etapa de preprocesamiento. Puede utilizar el primer minuto de la señal para ajustar los parámetros de dicho algoritmo.

12. Ejercicio 12

” Utilice el algoritmo Gold Standard, cuya implementación se encuentra en el archivo *deteccion_qrs.m*, para realizar la detección automática de los complejos QRS a partir de la señal de salida de la etapa de preprocesamiento. Evaluar la performance contando la cantidad de falsos positivos y falsos negativos. Un falso positivo es una marca que no se corresponde con un complejo QRS. Esto también puede tomarse como la inserción errónea de una marca de complejo QRS. Un falso negativo es no marcar un complejo QRS. Utilice como tolerancia para el cálculo del error de localización el ancho aproximado de los complejos QRS.”

Solución: Para detectar los complejos QRS, los picos locales de la señal integrada son encontrados. Un pico se define como el punto en donde la señal cambia el signo de la derivada (de positivo o creciente a negativa o decreciente). Después de cada pico, ningún pico puede ser detectado en los siguientes 200ms. Esto es una restricción fisiológica debido al periodo refractario durante el cual la despolarización ventricular no puede ocurrir incluso ante la presencia de un estímulo.

Performance:

Aciertos	Falsos Positivos	Falsos Negativos	Eficiencia
1801	143	289	86.17224880 %

Cantidad Esperada	Cantidad Obtenida
2090	1944

Donde la eficiencia porcentual se calcula como: $Eficiencia \% = \frac{Aciertos}{Cantidad_Esperada} \cdot 100 \%$

13. Ejercicio 13

”Adicionar ruido blanco a la señal de ECG para obtener señales con una relación señal a ruido de 30dB, 20dB y 10dB, respectivamente. Evalúe el desempeño del detector con estas nuevas señales. La relación señal a ruido puede obtenerse como: ”

$$SNR_{dB} = 10 * \log_{10} \left(\frac{\sum_{n=1}^N \text{señal}(n)^2}{\sum_{n=1}^N \text{ruido}(n)^2} \right)$$

Solución:

Para generar el ruido blanco utilizamos la funcion

`np.random.normal(scale=np.sqrt(varianza), size=cant_muestras)` del modulo numpy, que genera una muestra de longitud `cant_muestras`, utilizaremos el largo de nuestra muestra inicial de electrocardiograma. Esto nos permite sumar las muestras con el ruido uno a uno.

13.1. Calculo del desvio necesario segun SNR

Para la varianza a utilizar, podemos usar que como la media del mismo es 0 (por como lo generamos), entonces:

$$\frac{1}{N} \sum_{n=1}^N \text{ruido}(n)^2 = \text{Var}(\text{ruido})$$

Por lo que, por ejemplo, si deseamos un $SNR_{dB} = 10dB$, podriamos despejar la varianza de la ecuacion que define al SNR:

$$\text{Var}(\text{ruido}) = \frac{1}{N} \frac{\sum_{n=1}^N \text{señal}(n)^2}{10^{\frac{SNR_{dB}}{10}}}$$

Donde rápidamente se puede calcular la suma de los cuadrados de la señal (que además no depende del valor de SNR elegido).

Esto me permitio definir el desvio estandar ($\sqrt{\text{Var}(\text{ruido})}$) de mi ruido blanco en cada caso, el unico parametro que restaba definir.

13.2. Resultados obtenidos

Implemente la funcion `calcular_ruido`, que recibe la señal y el SNR deseado y calcula el ruido a agregar a la señal:

```
def calcular_ruido(y,SNR):
    cant_muestras = len(y)
    varianza = np.var(y)/(10**((SNR/10)))
    return np.random.normal(0,np.sqrt(varianza), cant_muestras)
```

Luego del agregado del ruido a la señal original, se realizó nuevamente todo el pre-procesamiento (pasabanda+derivador+cuadrado+integrador).

Con el fin de obtener nuevamente la performance a la salida de nuestro ultimo filtro se utilizó la misma función (`find_peaks` y el algoritmo `Gold Standard`) para nuestras 'nuevas' señales.

```
#Preprocesamiento de las señales
def preprocesar(y):
    y_out_hl = filter_signal(num_hl,den_hl,y)
    y_out_hl = compensar_retardo(5,y_out_hl)
    y_out_hh = filter_signal(num_hh,den_hh,y_out_hl)
    y_out_hh = compensar_retardo(16,y_out_hh)
    y_out_hd = filter_signal(num_hd,den_hd,y_out_hh)
    y_out_hd = compensar_retardo(2,y_out_hd)
    y_out_cuad = np.power(y_out_hd,2)
    y_out_hi = filter_signal(num_hi,den_hi,y_out_cuad)
    y_out_hi = compensar_retardo( N//2, y_out_hi)

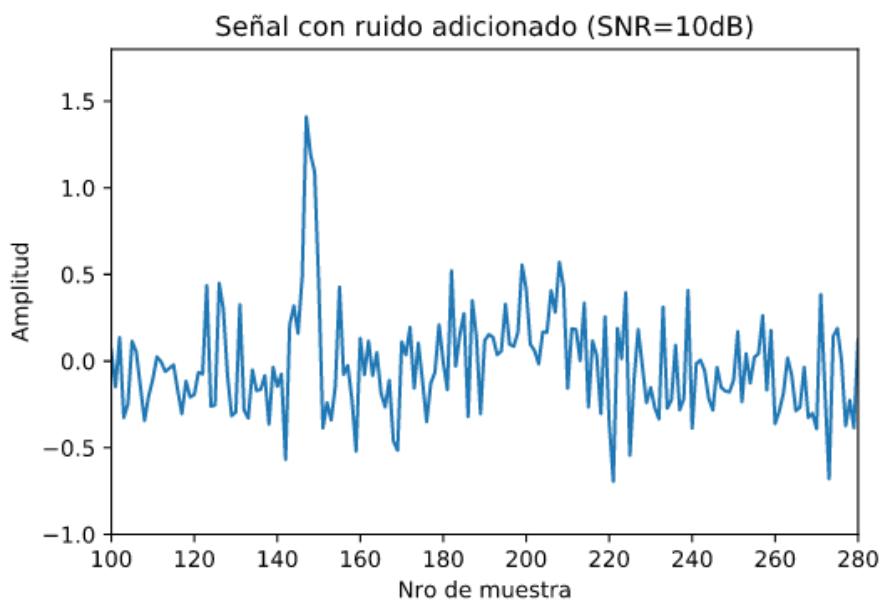
    return y_out_hi
```

Esta función que se implementó comprime todos los filtros en uno, para hacer mas compacto el código. En los puntos previos no se utilizó debido a que necesitaba analizar los estados parciales de la señal.

Luego de obtener las 3 señales preprocesadas, se evalua la performance.

13.2.1. $SNR_{dB} = 10dB$

Al agregar el ruido blanco correspondiente se obtuvo:



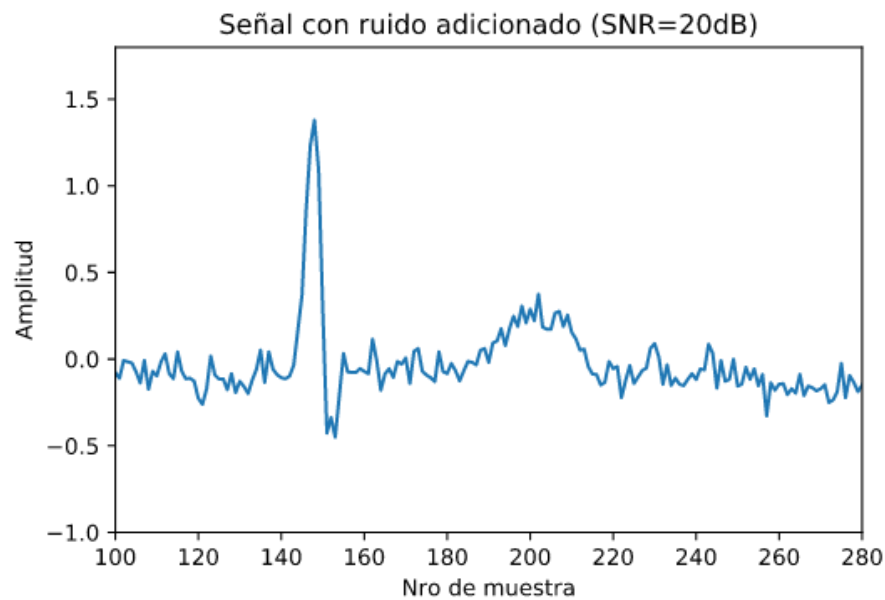
Performance:

Aciertos	Falsos Positivos	Falsos Negativos	Eficiencia
1772	425	318	84.78468899 %

Cantidad Esperada	Cantidad Obtenida
2090	2197

13.2.2. $SNR_{dB} = 20dB$

Al agregar el ruido blanco correspondiente se obtuvo:

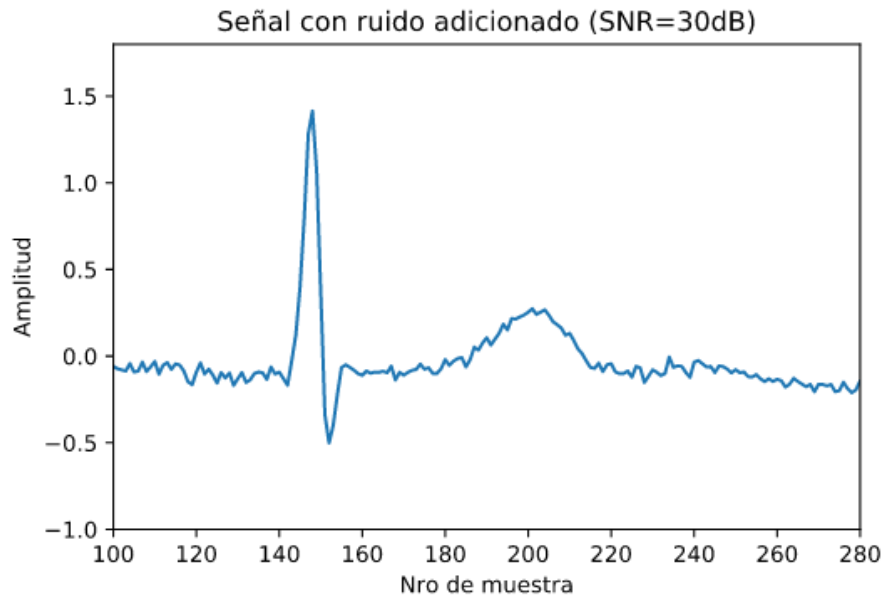
**Performance:**

Aciertos	Falsos Positivos	Falsos Negativos	Eficiencia
1796	141	294	85.93301435 %

13.2.3. $SNR_{dB} = 30dB$

Al agregar el ruido blanco correspondiente se obtuvo:

Cantidad Esperada	Cantidad Obtenida
2090	1937



Performance:

Aciertos	Falsos Positivos	Falsos Negativos	Eficiencia
1803	143	287	86.2679425 %

Cantidad Esperada	Cantidad Obtenida
2090	1946

13.3. Conclusiones

Es evidente que a medida que el ruido adicionado es mas predominante en la señal, menor es la eficiencia del algoritmo obtenida.

No obstante, es relevante aclarar que en el caso de $SNR = 30dB$, se obtiene una mejor Eficiencia que sin ruido (en el ejercicio 12).

Con una breve inspección al algoritmo utilizado, estimo que el mismo está mejor preparado para trabajar con señales con este nivel de SNR, por lo que la eficiencia en la misma se vería afectada levemente de forma positiva.

Viendo los falsos positivos y falsos negativos, es notable la diferencia entre $SNR = 10dB$ y el resto, y atribuyo la misma a la inmensa cantidad de picos nuevos que se generan al agregar

el ruido correspondiente.

14. Ejercicio 14

”Realice todo el procesamiento anterior pero con las señales re-muestreadas a $360Hz$ en lugar de $200Hz$. Describa el procesamiento discreto necesario para cambiar la frecuencia de muestreo y determine la nueva forma que deben tener los filtros $H_L(z)$, $H_H(z)$, $H_d(z)$ y $H_i(z)$ (integrador) para cumplir con idénticos requerimientos de procesamiento que en el caso de la frecuencia de muestreo original.”

Solución: En primer lugar, comenzaré realizando el razonamiento para poder cambiar la frecuencia de muestreo de $200Hz$ a $360Hz$.

Se puede lograr lo buscado mediante un resampleo de la misma mediante una expansión, un filtro pasa bajos y un decimador.

Se puede apreciar a continuación un esquema de lo descrito anteriormente.

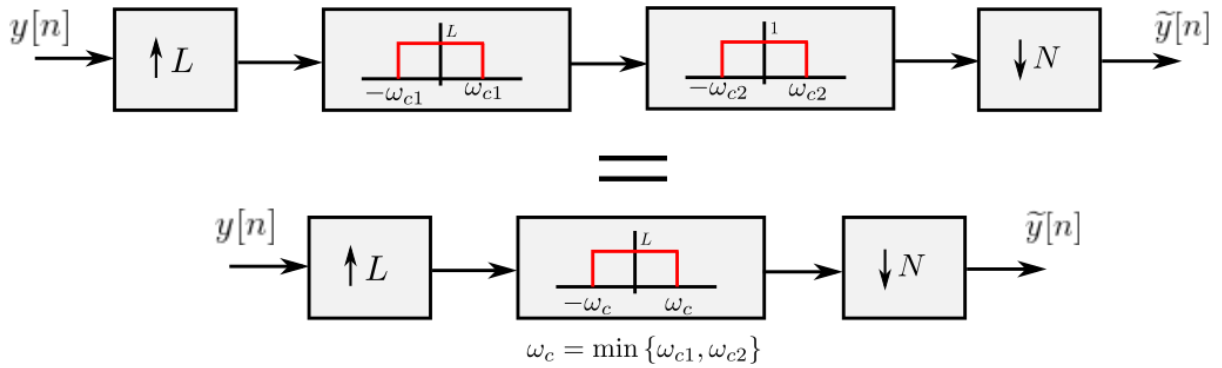


Figura 1: Esquema del resampleo

Si la señal en tiempo real es $y_c(t)$ y su muestreo a $200Hz$, se puede expresar:

$$y[n] = y_c\left(\frac{n}{200}\right)$$

La idea entonces es generar $\tilde{y}[n] = y_c\left(\frac{n}{360}\right)$, que representa la señal muestreada a $360Hz$.

Para obtener el objetivo utilizamos el esquema de expansor, pasabajos y decimador propuesto.

Notemos antes que ω_c debería cumplir:

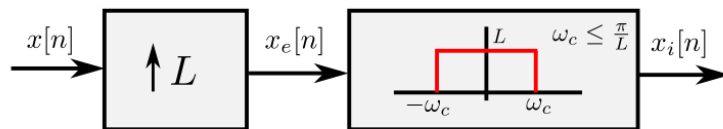


Figura 2: Esquema del expansor

Entonces como la configuración a utilizar nos garantiza:

$$\tilde{y}[n] = y_c \left(n \cdot \frac{N}{200L} \right) = y_c \left(\frac{n}{360} \right)$$

Llegamos a la conclusion de que:

$$\boxed{\frac{N}{L} = \frac{5}{9}}$$

Esto no nos limita $N=5$ y $L=9$ necesariamente, ya que por ejemplo otro par que cumple la ecuacion es $N=10$ y $L=18$.

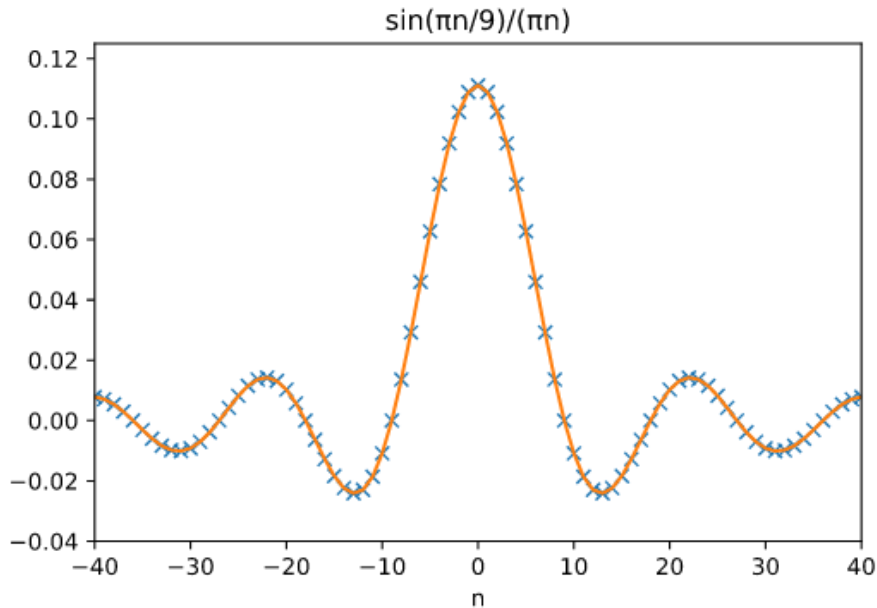
Sin embargo utilizo $\boxed{N = 5}$ y $\boxed{L = 9}$, ya que la frecuencia de corte del filtro pasabajos debe cumplir $\omega_c \leq \frac{\pi}{9}$. Elegir un mayor L nos restringiría mas la frecuencia de corte del mismo.

Para implementar el filtro deseado, necesitamos que

$$h_l[n] = \frac{\sin\left(\frac{\pi n}{9}\right)}{\pi n}$$

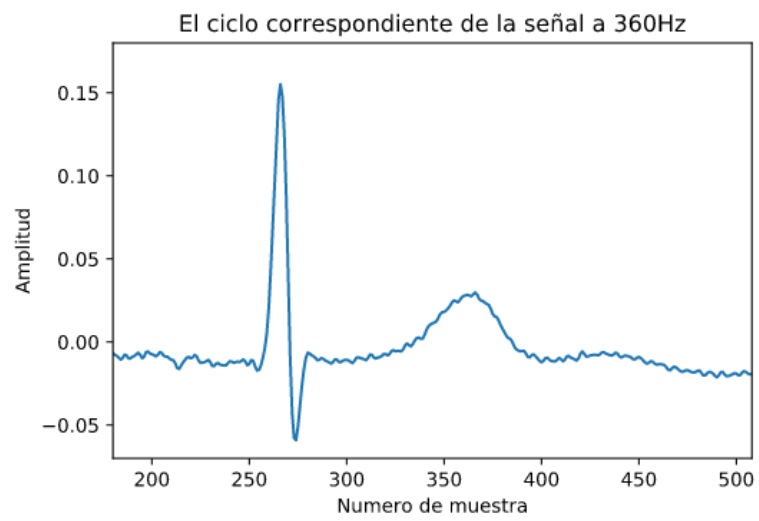
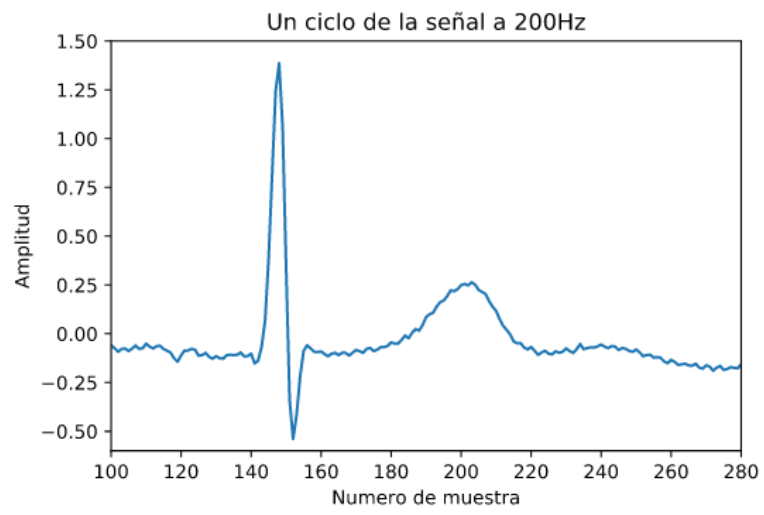
La truncaremos con fines prácticos en 15000 muestras (teniendo en cuenta la cantidad de muestras iniciales).

Muestro entonces un grafico de la funcion $h_l[n]$ utilizada:



Cabe aclarar que la misma tiene una ganancia de $1/9$, por lo que, como se muestra en las siguientes imágenes, las amplitudes se verán alteradas por una constante.

Decidí no reescalarla por el simple hecho de que no afectará la evaluación de la performance a la hora de encontrar los complejos QRS.



14.1. Evaluación de desempeño

SNR	Aciertos	Falsos Positivos	Falsos Negativos	Eficiencia
-	1975	192	115	94.49760765 %
10dB	1964	253	126	93.97129186 %
20dB	1974	207	116	94.44976076 %
30dB	1978	191	112	94.64114832 %

SNR	Cantidad Esperada	Cantidad Obtenida
-	2090	2167
10dB	2090	2217
20dB	2090	2181
30dB	2090	2169

14.2. Conclusiones

Se puede apreciar una gran mejora en la calidad de la Eficiencia frente a la señal muestreada a $200Hz$. Además, cabe destacar el caso en el que $SNR = 30dB$, que ayuda a aumentar aún mas la eficiencia del algoritmo. Este ultimo parece estar mejor adaptado para dichos niveles de ruido.

15. Conclusión

Finalizando todos los ejercicios del trabajo practico especial, cabe realizar un breve resumen de los aspectos más importantes del mismo.

Cabe destacar que el hecho de asentar en un caso práctico, la mayoría de los conceptos vistos en la materia, fue de gran ayuda para poder terminar de afianzar la teoría y ejercicios estudiados a lo largo del cuatrimestre en la misma.

En segunda instancia, quiero mencionar que la utilización de Python en la realización del mismo pudo terminar de consolidar mi manejo de las librerías `numpy`, `scipy` y `matplotlib`. Estoy mas que satisfecho con los conceptos y herramientas incorporadas, que serán de gran ayuda no solo en las materias a cursar en el futuro, sino también en mis futuros emprendimientos como Informático utilizando Python.

En cuanto al contenido del trabajo practico en particular, cabe aclarar que los últimos ejercicios fueron en los que encontré más dificultad, en especial aquellos en los que utilizaban el algoritmo Gold Standard. Este ultimo era sencillo en su utilización, pero fue complejo preparar la muestra desde la salida del integrador hasta que ingresaba al algoritmo.

No obstante, el algoritmo presento resultados mas que satisfactorios, incluso en los momentos en que trataba con ruidos no estacionarios.

Referencias

- [1] Repositorio del proyecto: <https://github.com/julianferres/TP-SyS>
- [2] Version .tex del Informe: <https://www.overleaf.com/read/rykgzmgptzxk>
- [3] Documentacion de numpy: <https://numpy.org>
- [4] Documentacion de matplotlib: <https://matplotlib.org/>
- [5] Documentacion de scipy: <https://www.scipy.org/>
- [6] Documentacion de scipy.signal: <https://docs.scipy.org/doc/scipy/reference/signal.html>