DSP Lab 3
104061171 紀伯翰

1a.Specification

Remove the notch filter circuits in lab 2. Connect the ECG amplification output to the analog input of Arduino board directly. Construct a digital filter to remove the 60-Hz power noise.

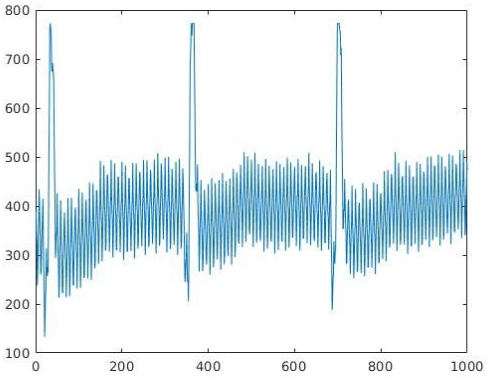
1b. Implement

1 Solution:

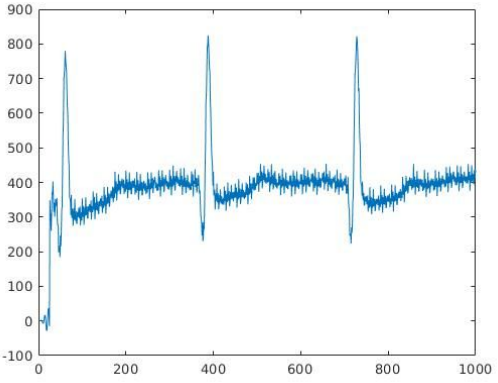
Pass through a band reject filter [remove 60hz]

Time domain

Original

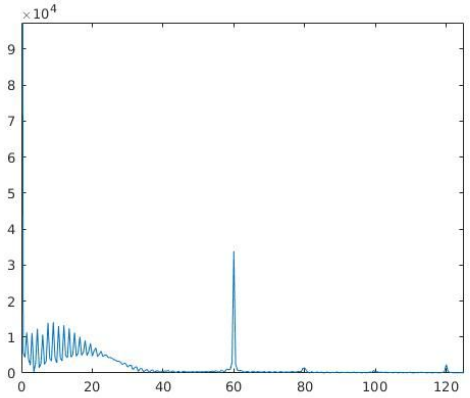


FIR filter(band reject on 60hz)



Frequency domain

Original



FIR filter(band reject on 60hz)

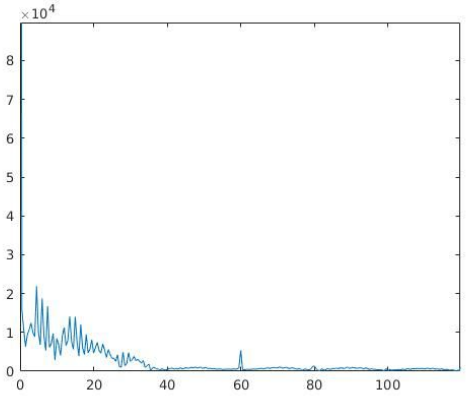
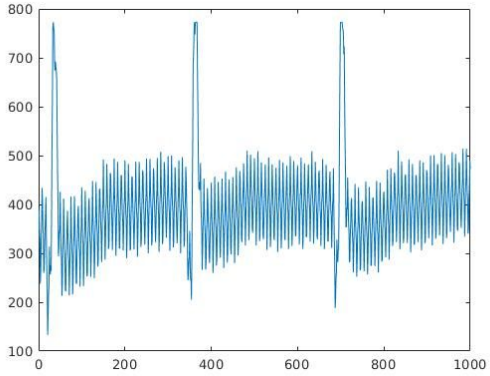


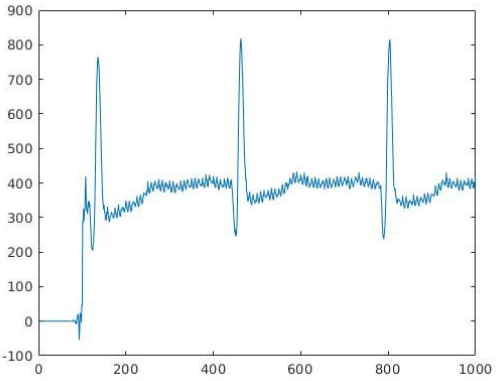
Figure 1(a,b,c,d)

Time domain

Original

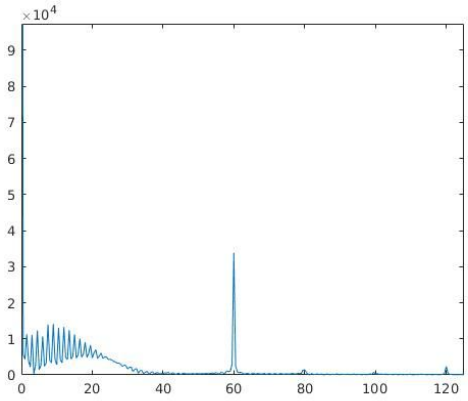


FIR filter(band reject on all noise)



Frequency domain

Original



FIR filter(band reject on all noise)

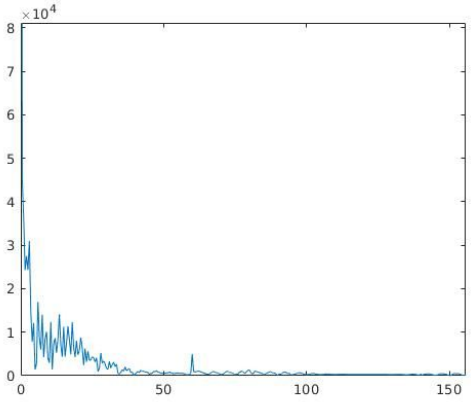


Figure 2(a,b,c,d)

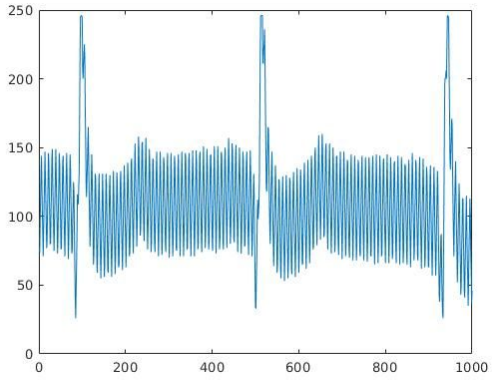
2 solution:

Pass through a low pass filter

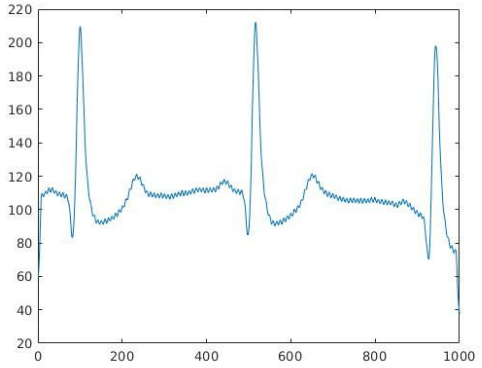
use moving_average_method to_implement

Time domain

Original

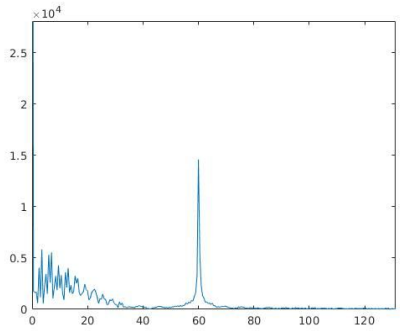


low pass filter

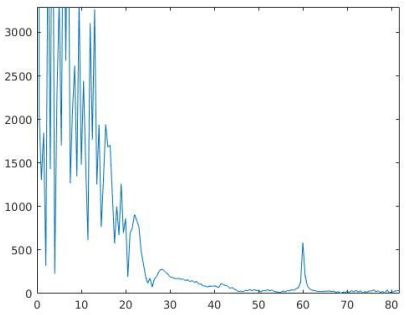


Frequency domain

Original



Low pass filter



1c.Discussion

以第一種解法來說,可以看到經過band reject filter後，frequency domain 位於60hz的signal被有效的壓低, 題目提及60hz的power noise,但我濾去的只是位於60hz的訊號(Figure 1(a,b,c,d)),還有harmonic的雜訊並未去除,所以我 後來也分別對於120,180,240,50hz(300-250=50)做band reject之後(Figure 2(a,b,c,d))得到濾乾淨的訊號，而在公式設計下我設b_k=1,order設計為50,代表其公式是取50個點做sigma.

FIR filter 設計底層概念:

以其中一種 MSE 方法 做解釋，參考wiki上的解釋方可得到答案。

（1）基本思想

$$MSE = f_s^{-1} \int_{-f_s/2}^{f_s/2} |H(f) - H_d(f)|^2 \, df$$

其中，f_s 为采样频率（sampling frequency），$H(f)$ 为设计出的滤波器之频谱，$H_d(f)$ 为欲设计的滤波器（desired filter）之频谱。

基本做法如上，在透過微分求取極小值，方得到欲設計的濾波器。

经过数学推导，我们可将MSE表示为

$$MSE = s^2[0] + \frac{1}{2}\sum_{n=1}^{k} -2\int_{-1/2}^{1/2} \sum_{n=0}^{k} s[n]\cos(2\pi nF)H_d(F)dF + \int_{-1/2}^{1/2} {H_d}^2(F)$$

其中的 F代表的是normalize的訊號 所以在參數設計上也有這個層份在其中。

以第二種解法來說,我把moving_average當作一種low pass filter，而我選擇16當作我unit_signal的長度(一次選取16個點),我在原signal訊號前補上(16/2-1)跟在訊號後補上(16/2）個零以便我做完的signal，長度一樣是1000，唯一要注意的點是當我在做moving_average的時候左右兩側的訊號會因為補0取平均被極端值干擾，面對這個問題有兩個解決方法，第一種先把bias 1.5伏特扣除，讓訊號在0上下，第二點只保留其中（n/2:1000-(n/2))個訊號點，減少訊號受到0干擾。

Moving_average = low_filter？:

如果將moving_average的運算轉換到frequency domain來看，就是一個sinc function而他的主頻帶即可以粗略當成一種low pass filter,而他的寬度取決於你的方波長度給多少，決定你大致上保留多少頻寬下來，但那些其他週期的小頻帶也有但值微小所以同時也造成了將原訊號其他的週期的範圍，壓低的作用。

ex:以我現在實作使用長度16個點的方波所對應到的sinc函數，他的low pass頻段大約是左右各f=15.625，小於這個值的signal才可通過但它其實還有其他範圍的小波段，雖說值很小，但在以15.625為週期的f(f=31.25,62.5)中會把signal屬於那些f的值壓低到0。

2a.Specification

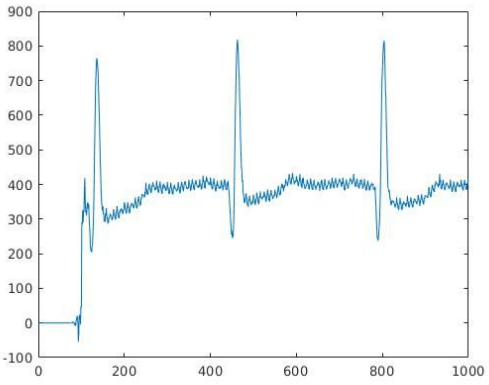
There exist a baseline wander noise in the recorded ECG signal. Try to use some filters to remove this low frequency noise.

2b.implement

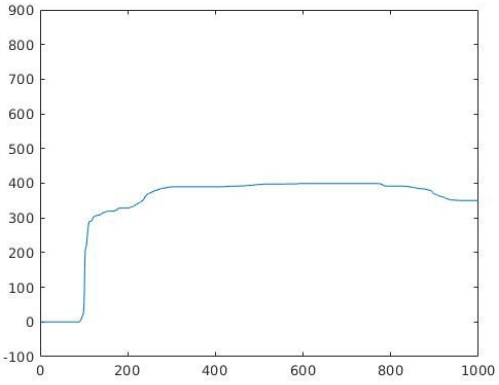
1 solution:

After remove 60hz power noise,the original signal pass through 200ms & 600ms median filter generate a baseline signal.Then take original and baseline signal to do differential will remove baseline problem.

Original Signal



Baseline Signal



Differential Signal(往上加180)[方便在freq看]

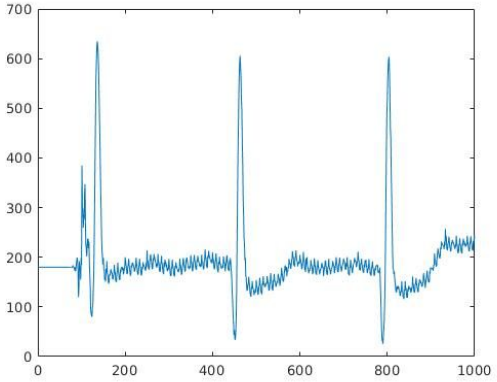
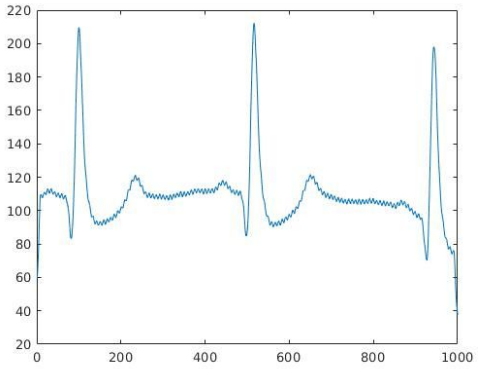


Figure3(a,b,c)

2 solution:
use [1 -1] do convolution with the signal which removed 60 hz signal to construct a high pass filter to find the peak

Original Signal



After High Pass Filter

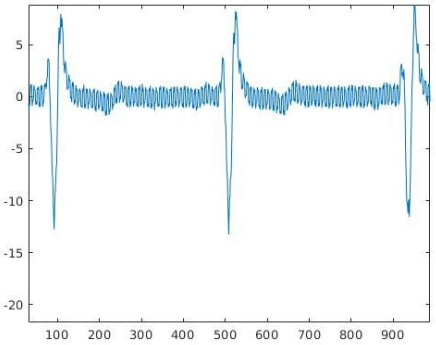


Figure 4(a,b)

2c. Discussion

第一種solution:

通過兩個median filter找出baseline然後再與原訊號相減移除baseline的雜訊。
Median filter :
取樣n個(以當前的點向左取([n/2 -1]個，以及向右取[n/2]個)，然後依照大小排列，找中間兩個值取平均。 由上述可知，我們可以知道此filter是一種nonlinear的low pass filter，因為當你將採樣的訊號按照大小排列，相當於以中間兩項取平均的值當作你此點的值，所以只會保留low pass的訊號。(或是理解為 保留的訊號變化不大)
最後我們可以透過此舉動將baseline濾出來，再和原訊號相減，就得到一個消除baseline機制的訊號了。
以paper中提到，使用200ms median filter可以剔除P-Q-R-S的訊號，以及使用600ms median filter可以剔除T的訊號，那是因為(P-Q-R-S)這些訊號普遍的間隔比200ms還要小所以當我通過median filter的時候還會有旁邊很多比較低的訊號，如此一來，訊號較高的值就不會被選到。而600ms是因為T訊號，分佈的範圍較長,所以用200ms median filter還是不易消除，唯獨用600ms median filter才可以達到移除的效果。

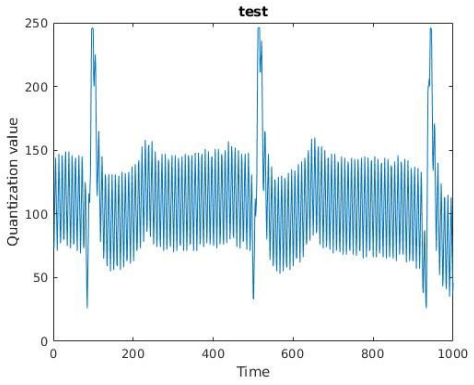
第二種solution:

以[1 -1]此種filter滑過數組做element-wise相乘，此作法相當於對原signal做斜率的運算，每一項後面減前面，所以只會留下兩點之間的差距，即baseline就被減掉了，因為訊號是整個被抬升或是下降，但每兩點的差距在每個週期內並不會差異太多，才會說是high pass filter，即刻的變化(相當於取微分)被保留在後面的 圖中。

3a.Specification

Use 8 bits instead of 10 bits for the digitization of ECG signals. Design an algorithm and implement in Arduino to maintain the maximum dynamic range of the input ECG signal with 8-bit sampling.

3b.Implement



3c Discussion

作法是先量測幾次找出範圍所在，再進行mapping，以我的設計方法是map(50,800,0,255),考慮到訊號會出現baseline shift出現上下移動的問題所以我將範圍取的稍微寬一點，增加noise的容忍度。

Extra Question

1a.Specification

Please analyze the recorded ECG signals in TIME and in FREQUENCY DOMAIN to see by what kind of noises (see the ECG introduction slides) your recorded signals are contaminated.

According to such an analysis, please "elaborate" (explain why) what kind of filter you implemented to remove the noises in your report.

1b.Implement:

原第一小題

原第二小題

1c Discussion
可以看到figure 1(a,b,c,d),我把time domain 和 frequency domain show出來分析在60hz的訊號的確有被壓低，但我是使用FIR FILTER做band reject把60hz附近的頻段移除，但這樣詬病的是我harmonic的雜訊反而沒被消到。後來我使用了moving average的手段利用sinc function幾乎只保留主頻段的特性(其他頻段的值很小還有不少zero crossing)，實現了一個low pass filter，很有效的把60hz的市電干擾全部移除。
可以看到第二題，我做了一個differential的filter，透過每兩點相減我把baseline的上下移位濾掉，因為這是在實現一個high pass filter，而baseline wander是屬於低頻的noise就被移除掉了。

2a.Specification
Please "elaborate" what kind of digital filter you implemented to remove the 60-Hz power-line harmonic noises (i.e., 60 Hz, 120 Hz, 180 Hz. ..etc) in your report. You may try to implement a simple MOVING AVERAGE FILTER to remove 60-Hz power-line harmonic noises. Please compare and comment the results from your designed FIR (MATLAB function: fir1()), IIR (MATLAB function: iirnotch) notch filters and moving average filter.

2b Implement

以FIR1實作

```
a = fir1(50,[0.20 0.28],'stop');
y2 = filter(a,1,disbuff);
figure('name','past by FIR Filter');
plot((y2));

figure('name','pass by FIR Filter & fft');
ft_ya2=abs(fft(y2));
plot((0:999)*fs/1000,ft_y2);
###以上濾除60hz的雜訊###

b = fir1(50,[0.4 0.56],'stop');
figure(100)
freqz(b)
y2 = filter(b,1,y2);
c = fir1(50,[0.68 0.76],'stop');
y2 = filter(c,1,y2);
d = fir1(50,[0.76 0.84],'stop');
y2 =filter(d,1,y2);
```

```
figure('name','past by all FIR Filter');
plot((y2));
figure('name','pass by all FIR Filter & fft');
ft_y2=abs(fft(y2));

plot((0:999)*fs/1000,ft_y2);
###以上濾除120,180,240hz的雜訊###
```

以 Moving_Average實作:

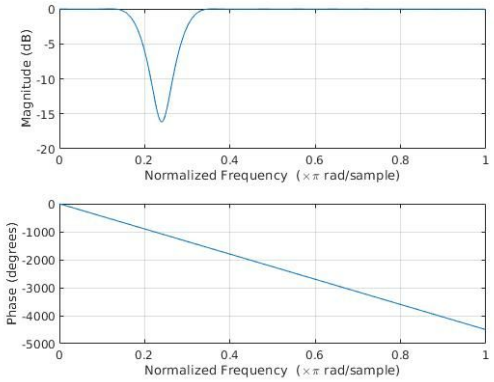
```
n=16;
moving_average = 1:n;
moving_average(moving_average>1)=1;

pad_zero = 1:((n/2)-1);
pad_zero(pad_zero>1)=0;
new_vector=zeros(1,1000);
pad_vector = [pad_zero disbuff pad_zero 0];

for i=1:1000
   new_vector(i)=(sum(pad_vector(i:(i+n-1)).* moving_average))/n;
End
```

以IIR Filter實作:

```
w0 = 60/(500/2);
bw = w0/10;
[b,a] = iirnotch(w0,bw);
freqz(b,a,50);
y2 = filter(b,a,disbuff);
figure('name','low_pass_filter ');
plot(y2);
ft_y2=abs(fft(y2));
figure('name','low_pass_filter fs');
plot((0:999)*fs/1000,ft_y2);
```

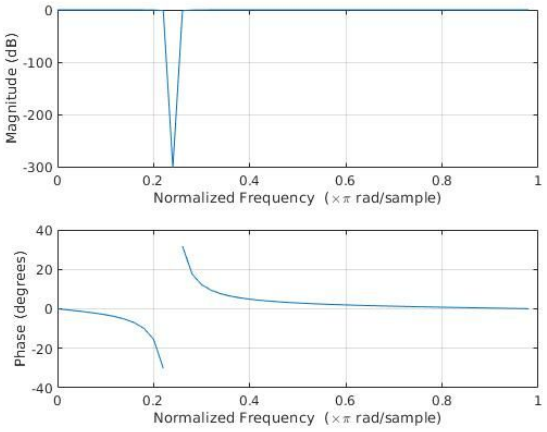
2c.Discussion

FIR 50 Order band reject 60hz



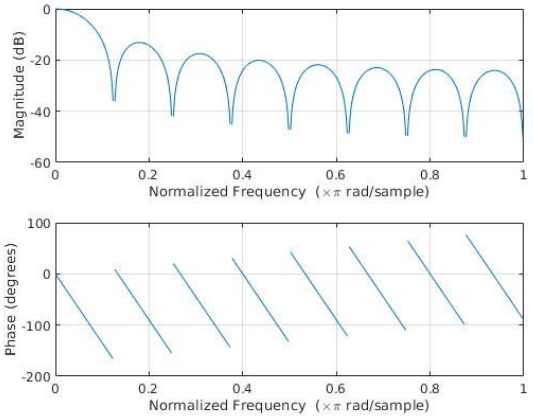
以FIR 50 order的band reject filter 我們可以看到它只是壓低大約 15db在0.24相當於(250hz * 0.24 = 60hz)的地方,可是其實並非讓訊號完全的不通過，但他的phase變化是linear的。

IIR notch band reject 60hz



可以看到，我是以相同的階數(50)去實現IIR notch filter，可以看到他在一樣60hz的地方，但卻壓低了300db的訊號，可以看到雖然階數相同，但IIR的效率比FIR好很多，不過在phase的部份卻是非線性的狀況。

Moving_average



可以看到moving_average除了主頻段的地方其他都壓得很低，跟我們所說lowpass filter非常符合。

4a. Specification:

Can you derive the system transfer function or frequency response of the analog notch filter? The system transfer function or frequency response can be used to derive the corresponding digital notch filter via impulse invariant transformation or Bilinear transformation. You may need it in Lab 3

4c.Discussion:

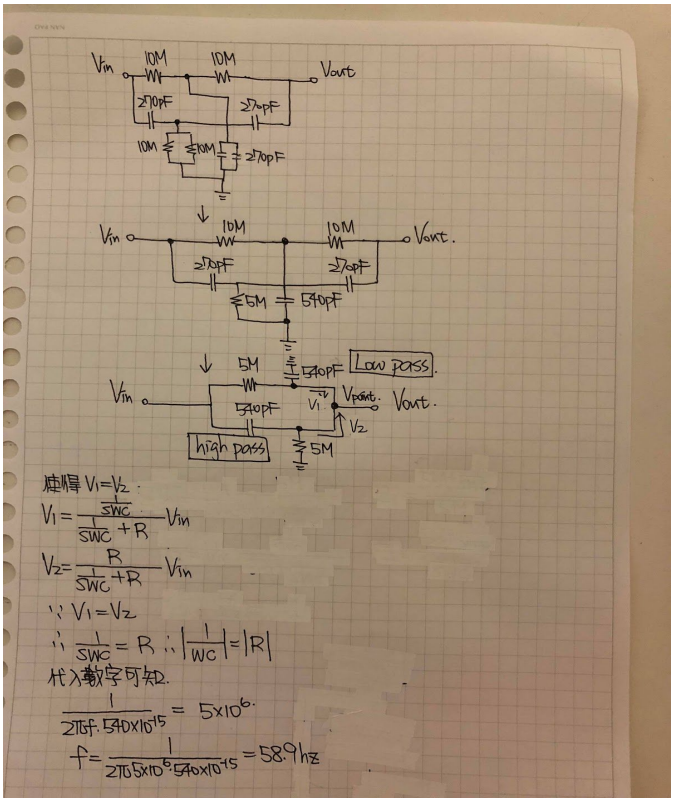


Fig.7(a)

相關公式推導可以參考這個網址:

http://fourier.eng.hmc.edu/e84/lectures/TwinT/node1.html

$$H(j\omega) = \frac{v_{out}}{v_{in}} = \frac{1 + (j\omega\tau)^2}{1 + 4j\omega\tau + (j\omega\tau)^2} = \frac{(j\omega)^2 + \omega_n^2}{(j\omega)^2 + 4j\omega\omega_n + \omega_n^2}$$

Fig 7(b)

4.Conclusion

這次實驗，接觸了不少有關數位訊號濾波器的實作，也讓我覺得其實這些底層的東西，實作起來，難度並不是那麼難，反而在現實的世界中，這些東西誤差也不會差很多，卻也達到相同的效果。雖然稍覺粗糙，不過卻可以達到不錯的效果與效率。

5.Reference

Wiki:
https://zh.wikipedia.org/wiki/%E6%9C%89%E9%99%90%E5%86%B2%E6%BF%80%E5%93%8D%E5%BA%94
Median_filter_to_implement_ECG:
https://arxiv.org/ftp/arxiv/papers/1408/1408.0453.pdf
Median_filter:
https://www.mathworks.com/matlabcentral/answers/47316-how-to-apply-200-ms-median-filter-in-matlab
Notch_filter:
http://fourier.eng.hmc.edu/e84/lectures/TwinT/node1.html