

● Lab Objective

在前一個 Lab 中，我們使用一個 Analog 的 notch filter 來處理 noise。在這個 Lab 中，我們改採用 DSP 的方式，去實作各種 Digital filter 來解決 ECG 量測時 noise 的問題，並對結果進行分析比較。除此之外，我們也設計一個演算法來處理降低 sampling 位元數的問題

● Design Implementation

在量測 ECG 訊號時，常見的 noise 有以下四種：

1. Power line noise
2. Baseline wander
3. EMG noise
4. Motion Artifact

在這個 Lab 中我們要設計數位的 filter 來處理前兩種 noise。

首先是 60 Hz powerline noise。要壓抑這個 noise 我們可以設計一個 moving average filter 或是一個 digital notch filter。下面我會介紹這些 filter 是如何設計的：

1. Moving Average Filter

一個 L-sample 的 moving average filter 是一個 FIR filter，他的 impulse response 如下：

$$h[n] = \begin{cases} \frac{1}{L}, & \text{for } n = 0, 1, 2, \dots, n-1 \\ 0, & \text{otherwise} \end{cases}$$

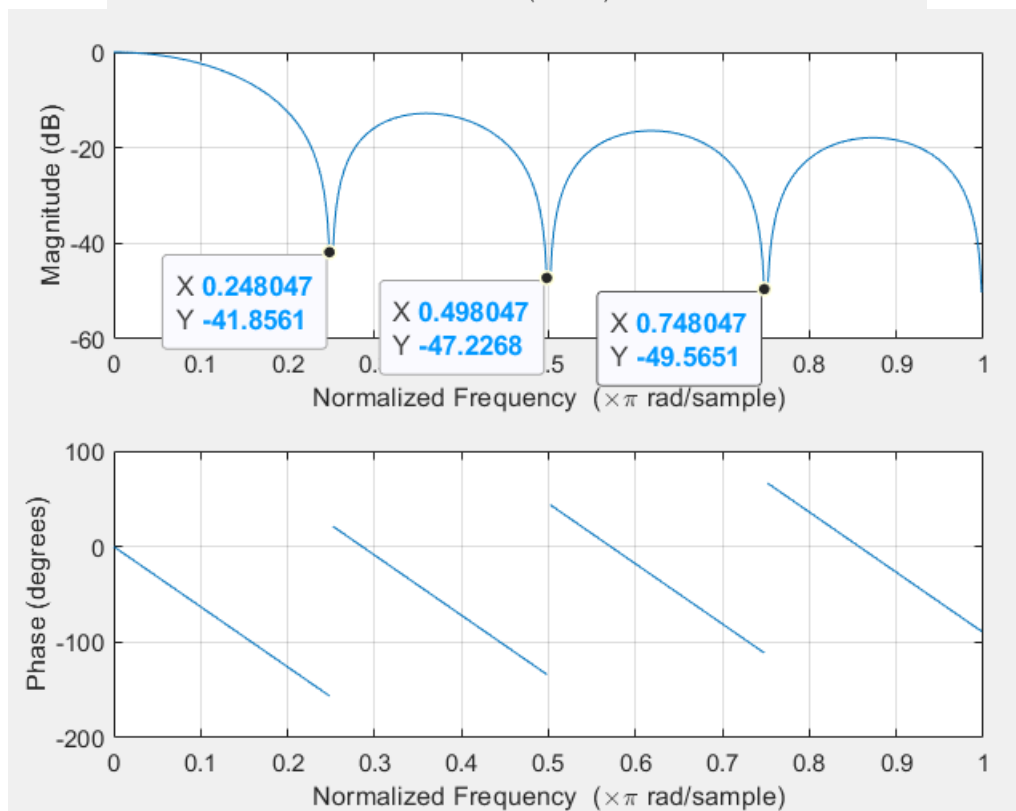
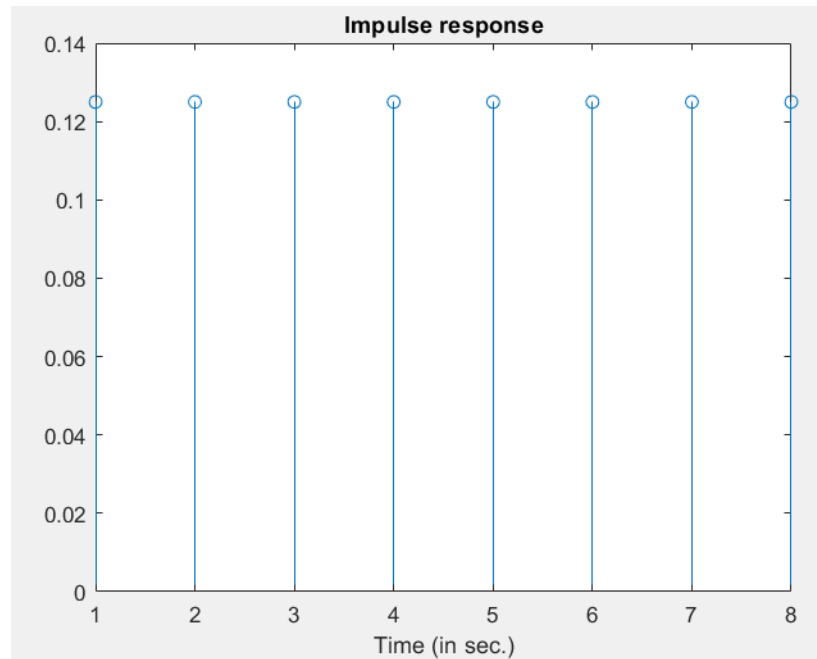
其 frequency response 則是

$$H(e^{j\omega}) = \frac{1}{L} \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}}$$

可以看到當 $\omega = \frac{2n\pi}{L}$ 時，frequency response 會是 0。所以我們的目標就是透過設計 L 讓這個 0 點掉在 60Hz 和其 harmonic 的位置。

$$\frac{2\pi}{L} \cdot \frac{f_s}{2\pi} = 60 \rightarrow L = \frac{f_s}{60}$$

根據前一個 Lab 的經驗， f_s 取 400~500Hz 就相當足夠，因為大多數 ECG 訊號的 frequency component 就掉在 200Hz 之內。因此我們取 $f_s = 500\text{Hz}$ ，這樣計算出來的 L 為 8。我們運用 MATLAB 畫圖，其 impulse response 和 frequency response 如下，可以看到他在 60Hz 整數倍頻率附近有零點：



2. Digital Notch Filter

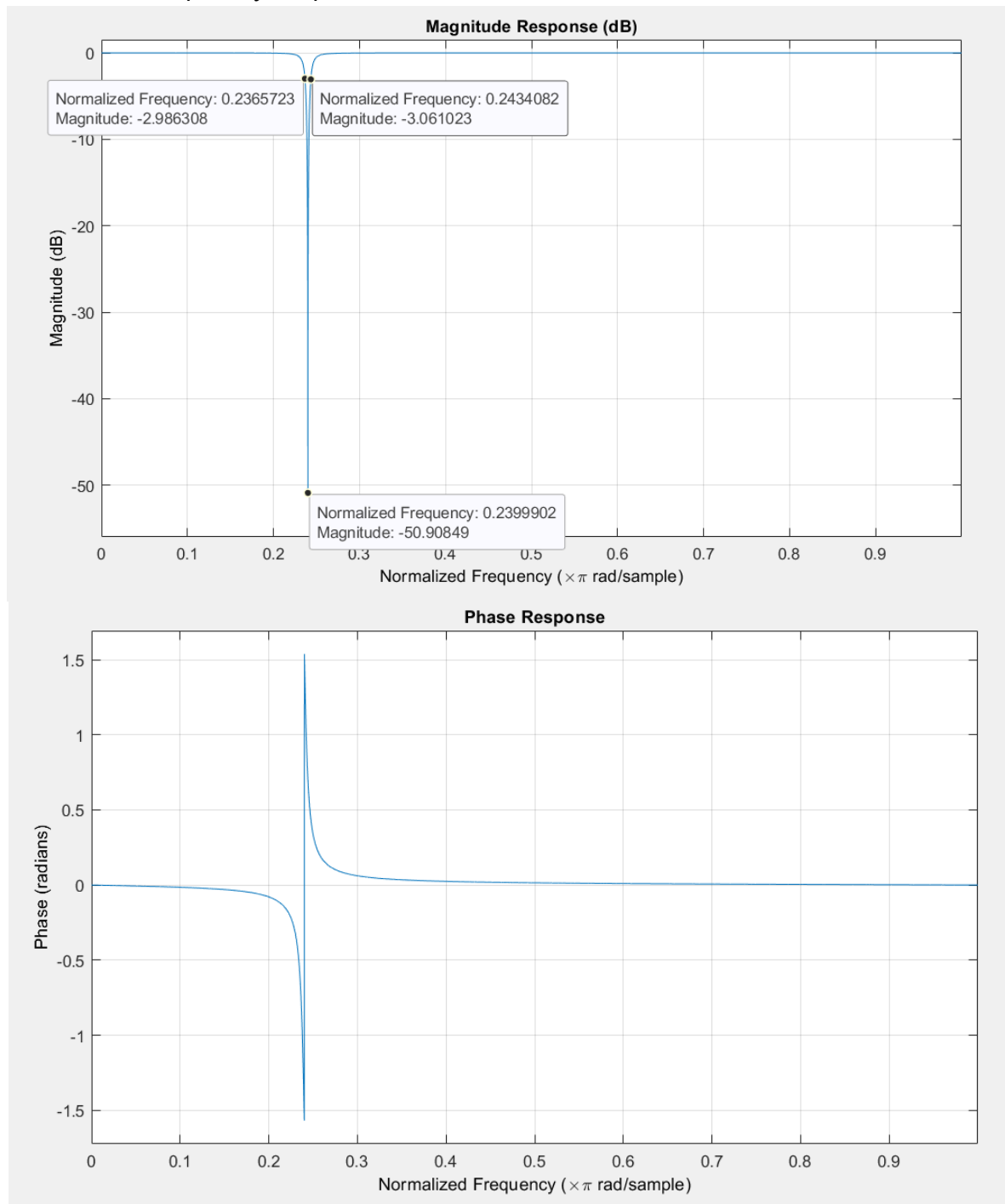
在上一個 Lab 中，我們使用了 Analog notch filter 來壓制 60Hz noise。要在數位系統中實做一個 Digital notch filter 我們得先從 Analog notch filter 的 transfer function 著手，透過一些推導找到 Digital notch filter 的 transfer function，才能進行實作。

Digital notch filter 根據設計方式不同，可以是 FIR filter 或 IIR filter。由於各自的方式都有很多種，也會造成不同的效果以及 error，下面我就對 FIR 和 IIR 各闡述一種我們在這個 Lab 中實作的方法，並進行分析比較。

1. IIR

實作上我們可以直接使用 MATLAB 中的 `iirnotch` 來實作。在呼叫 `iirnotch` 時，我們需要設

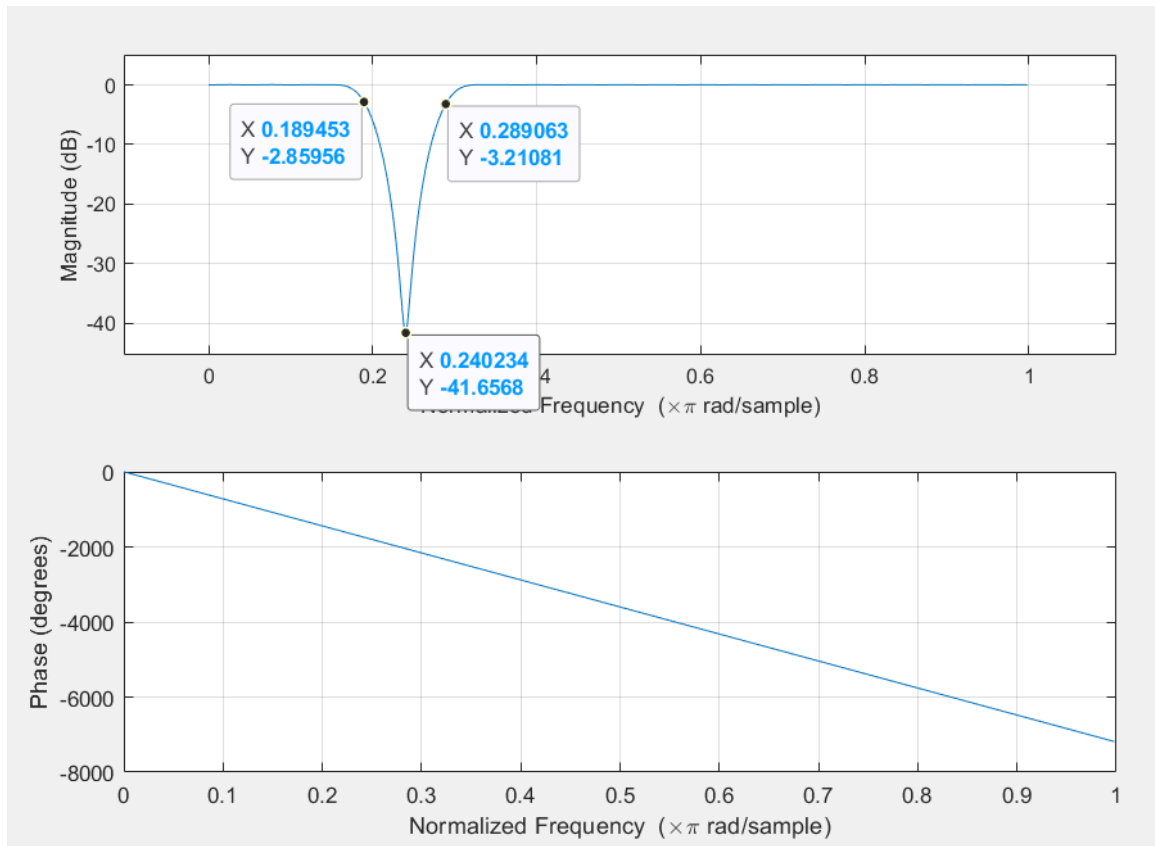
定兩個參數：notch 的位置以及這個 filter 的 Q factor。另外，要注意的是 iirnotch 回傳的東西是這個 filter 的 transfer function 中分子分母的係數，因此在後續要做 convolution 時還需要進行一些額外的步驟，要使用 filter() function 而非 convolution。我們得到的 IIR notch filter 的 frequency response 如下：



可以看到這個 filter 在 $250 \times 0.24 = 60\text{Hz}$ 附近有一個很窄的 stop band。另外，從 phase 的角度來看這不是一個 linear phase filter，所以會有 phase distortion 的現象。

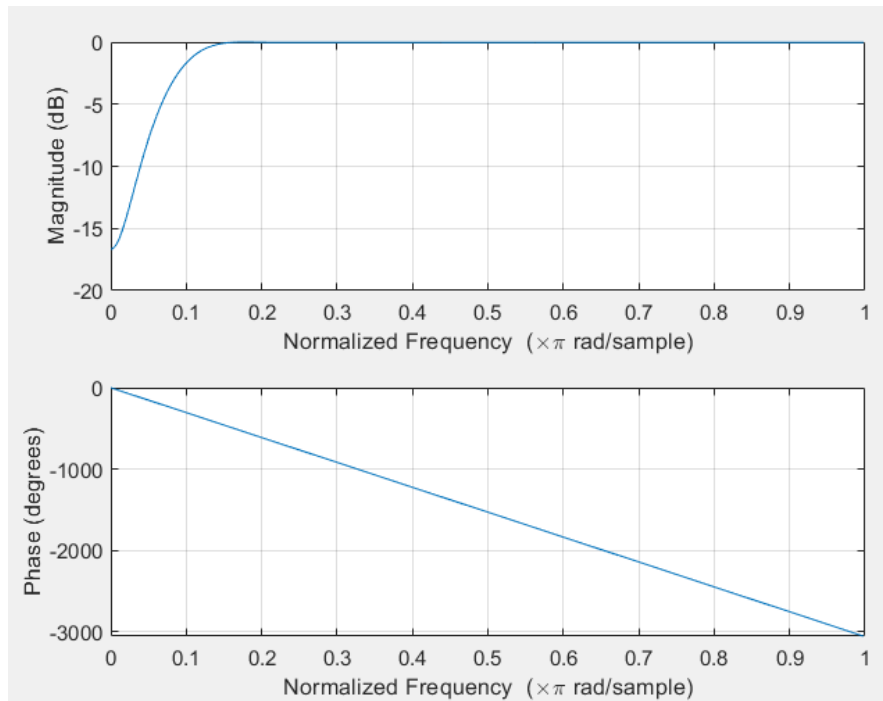
2. FIR

實作上，我們可以使用 MATLAB 中的 fir1() function，設定其 passband 以及 filter 的種類，我採用的 passband 為 $[0.2, 0.28]$ ，filter 的 order 為 48，filter 的種類是一個 bandstop filter。他的 frequency response 如下：

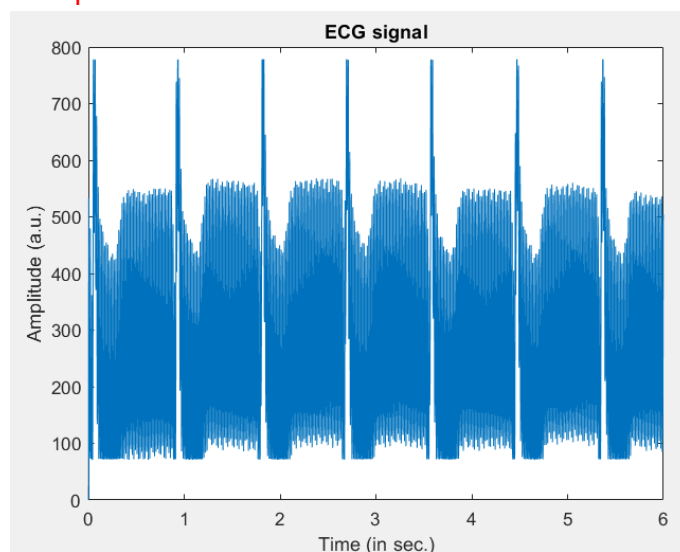


可以看到他在 60Hz($250 * 0.240234 \approx 60$)處被壓抑許多，他的 stopband 位在 $250 * 0.18 = 45\text{Hz}$ 到 $250 * 0.28 = 70$ 之間，相較於前面設計的 IIR notch filter，FIR 的 stop band 比較寬，我認為原因來自於我們在呼叫 `fir1()` 時使用的 filter type 是 bandstop filter，他設計的 Q factor 不夠大，因此 passband 才比較寬。另一方面，可以發現這是一個 linear phase 的 filter 所以不會發生 phase distortion 的情況。

接下來是處理 Baseline wander 的方法。Baseline wander 可以想成是 ECG 訊號搭載一個低頻 (below 1Hz) 的載波，因此我們可以嘗試使用一個 high pass filter 去處理這個問題。在實作上，我們可以直接使用 MATLAB 中的 `fir1()`，並將其 filter type 設為 high-pass，cutoff frequency 設在 10Hz 左右(如前面的 FIR notch filter 的 case，若我們把 cutoff frequency 設在 1Hz 附近，baseline wander 被壓抑的程度就不太夠，最後仍看得出有 baseline wander noise)。下面是我設計的 filter 的 frequency response：

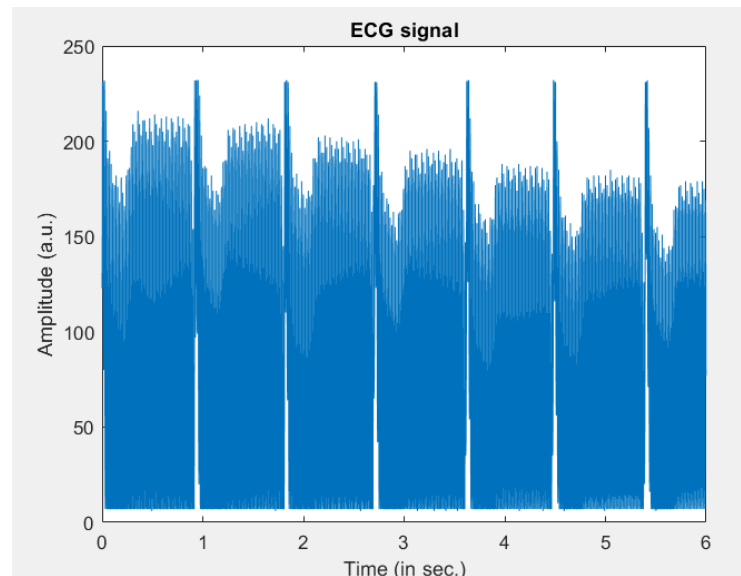


另外，我們還需要處理一個問題也就是改變 **sampling 位元數** 的問題。這個問題會發生在若我們希望將處理完的訊號透過一個 DAC 傳送出去，但是 DAC 所支援的位元數不及我們的數位示波器取樣的位元數(ex. Arduino 取樣 10bit 但下一級 DAC 只支援 8bit)。在第一個 Lab 中，我們透過 Arduino 的 **map function** 將 10 位元(0-1023)線性轉換到 8 位元(0-255)，這麼做確實能解決這個問題，但是這種作法的缺點在於：量測到的 ECG 訊號並不會完全佔滿整個 10 位元的 **range**，以剛剛量測到的訊號為例，量測到的訊號的 range 掉在 70-800 之間，依此事實上我們並沒有完全利用 0-1023 這個 range，若我們直接將他 map 到 8bit，我們也會同時在 8 bit 浪費掉一些空間，導致 **quantization error** 變多。



面對這個問題，我的解決方法為：我們可以**多使用兩個參數 bottom 和 top**，**bottom** 代表 **mapping** 的最低值，**top** 代表 **mapping** 的最大值，以前面的量測為例，我取 **bottom = 50**, **top = 850**，這樣就能用一個比較小的 range 完全包住整個量測訊號，但如果訊號因 **baseline wander** 或其他 **noise** 上下波動太大導致掉出這個 range 怎麼辦呢？我們可以**動態調整這個**

range，如果量測到的 signal 掉到這個 range，我們就把整個 range 向下平移，反之則向上平移，這樣就能在用比較少的取樣位元下，仍保持 maximum dynamic range。下面是我們用比較少的取樣位元下的演算法成果，可以發現我們有效的運用 0-255 這個 range，而不會浪費太多。



● Discussion

下面對於量測到的結果以及一些發現的問題進行討論：

Q1)IIR filter 和 FIR filter 在實作上和使用上有什麼區別，各有什麼優缺點？

FIR filter 以及 IIR filter 的差別在於 FIR 的 impulse response 只包含有限多項，從 transfer function 的角度來看，兩者的 transfer function 來表示成以下形式：

$$H(z) = \sum_{i=0}^M b_i z^{-i} \quad H(z) \equiv \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

左邊是 FIR 的 transfer function 而右邊則是 IIR 的 transfer function (a_k 不全為 0)，可以看到兩者的差異在於有沒有 non-trivial denominator term，這些 non-trivial denominator term 的影響是什麼呢？從 z transform 的角度來看，non-trivial term 的影響就是有 feedback，因為我們能將 FIR 和 IIR 的 transfer function 做 inverse z transform 得到：

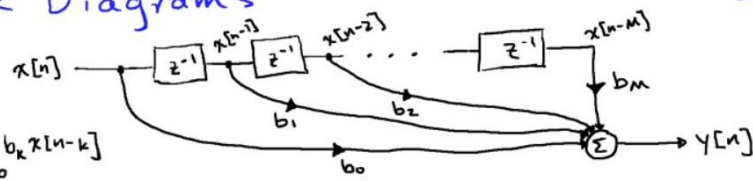
$$\begin{aligned} \text{FIR} : y[n] &= \sum_{i=0}^M b_i x[n-i] \\ \text{IIR} : y[n] &= \sum_{i=0}^M b_i x[n-i] - \sum_{k=1}^N a_k y[n-k] \end{aligned}$$

畫成 block diagram 如下，可以看到 IIR 的 output 處有很多的 feedback 這就是 non-trivial denominator term 的影響。

Block Diagrams

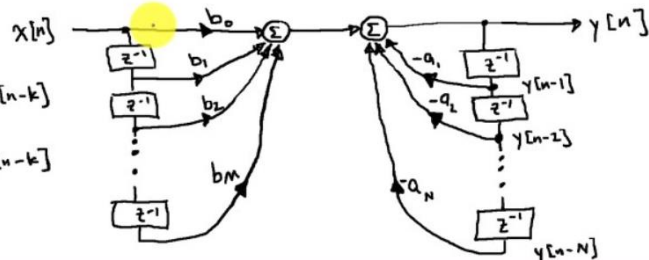
FIR

$$y[n] = \sum_{k=0}^M b_k x[n-k]$$



IIR

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

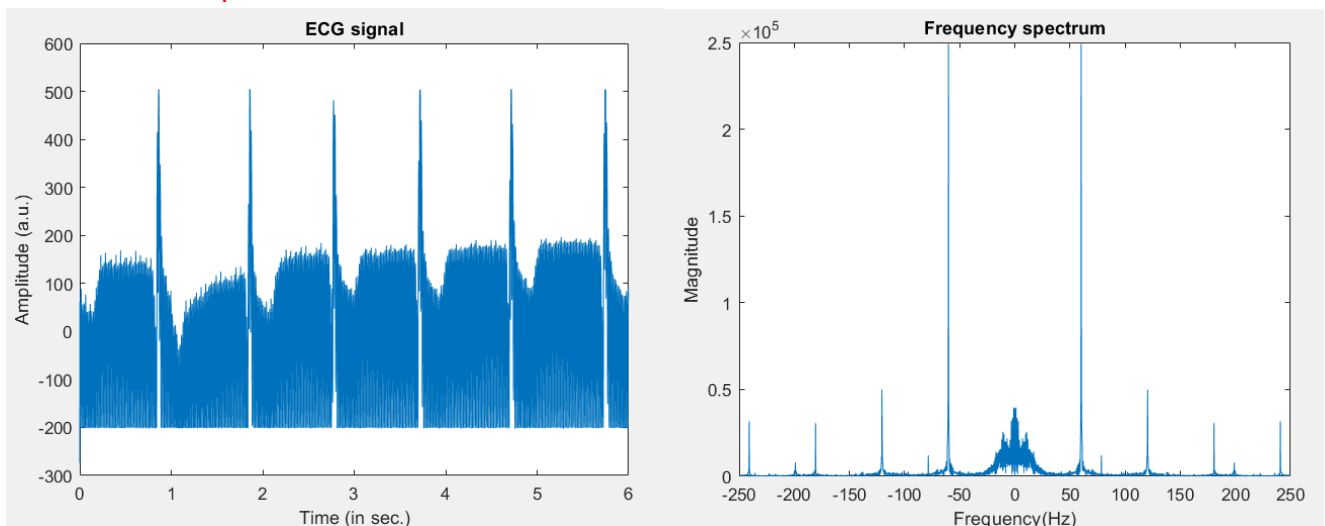


在實作上，FIR filter 可以透過 convolution 去實現，因為他的 impulse response 只有 finite term；相較之下，IIR filter 則無法透過 convolution 去實現，我們只能透過 Linear Constant Coefficient Differential Equation(LCCDE)去實作。

IIR filter 的優點在於設計較為直觀，我們可以透過 impulse invariant transform 或 bilinear transform(Discuss in Q4)來將 Analog filter 的 transfer function 轉換成 Digital IIR filter 的 transfer function，在達成相同的 spec 之下，IIR filter 的 order 通常比 FIR filter 的 order 低。FIR filter 的優點則在於我們能確保這是一個 BIBO 的系統，並且能讓這個 filter 是一個 linear phase 的系統，避免 phase distortion。

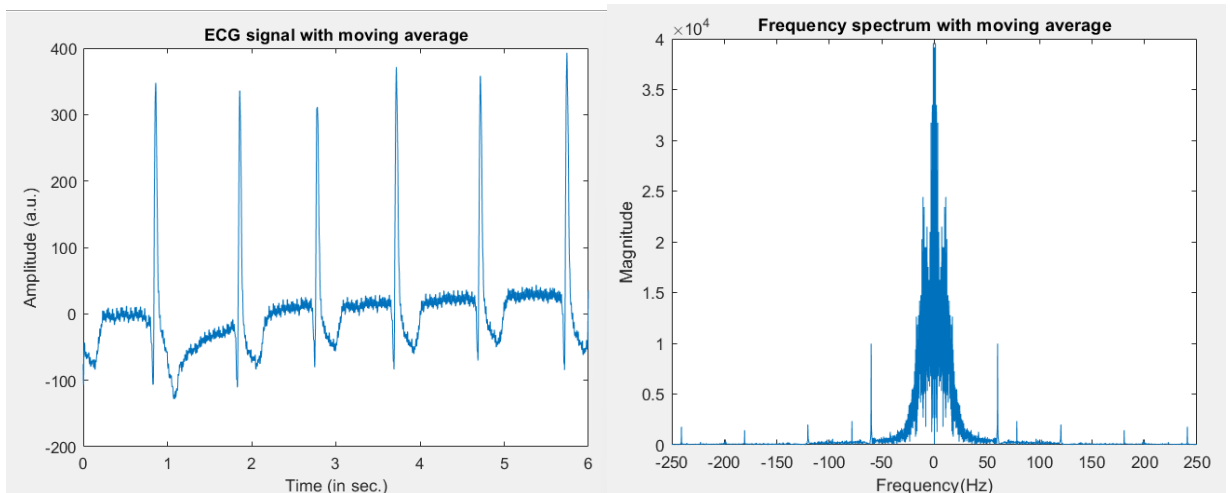
Q2) 我們量測到的 ECG 受那些 noise 影響？不同 filter 對於 noise 的處理結果為何？各自有什麼優缺點？

首先，我們先觀察量測到的 ECG 訊號(沒經過 DSP，但有扣掉 DC signal)。扣掉 DC 的原因在於這個 DC 訊號的來源是我們量測的類比電路中 Driving buffer 提供的 1.5V。可以發現在 P 和 T 兩個 interval 的雜訊很嚴重，從 frequency domain 來看可以發現 noise 主要來自 60Hz power line noise。

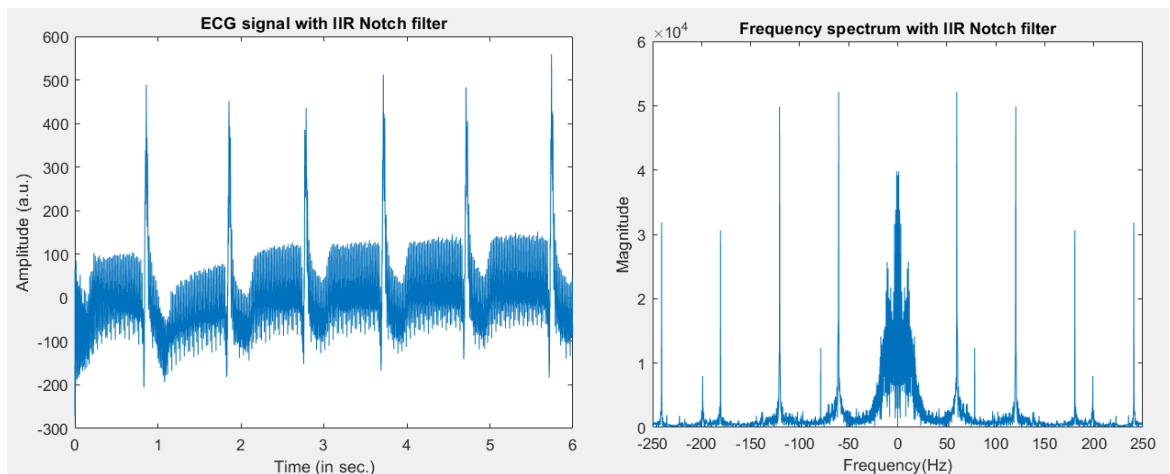


接下來，我分別將這個訊號通過前面所設計的 4 種 filter，並觀察他們的效果：

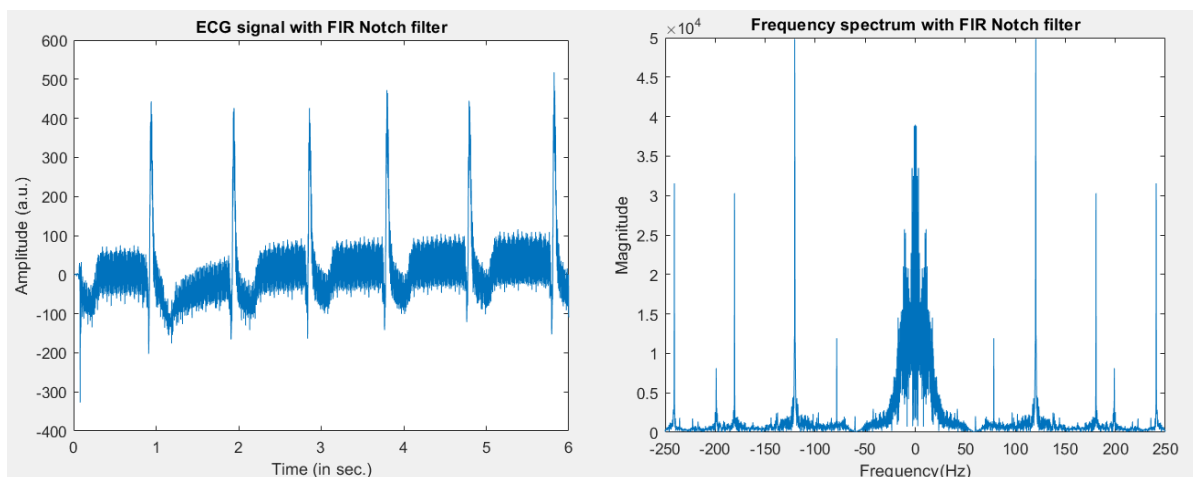
1. Moving Average filter



2. IIR notch filter

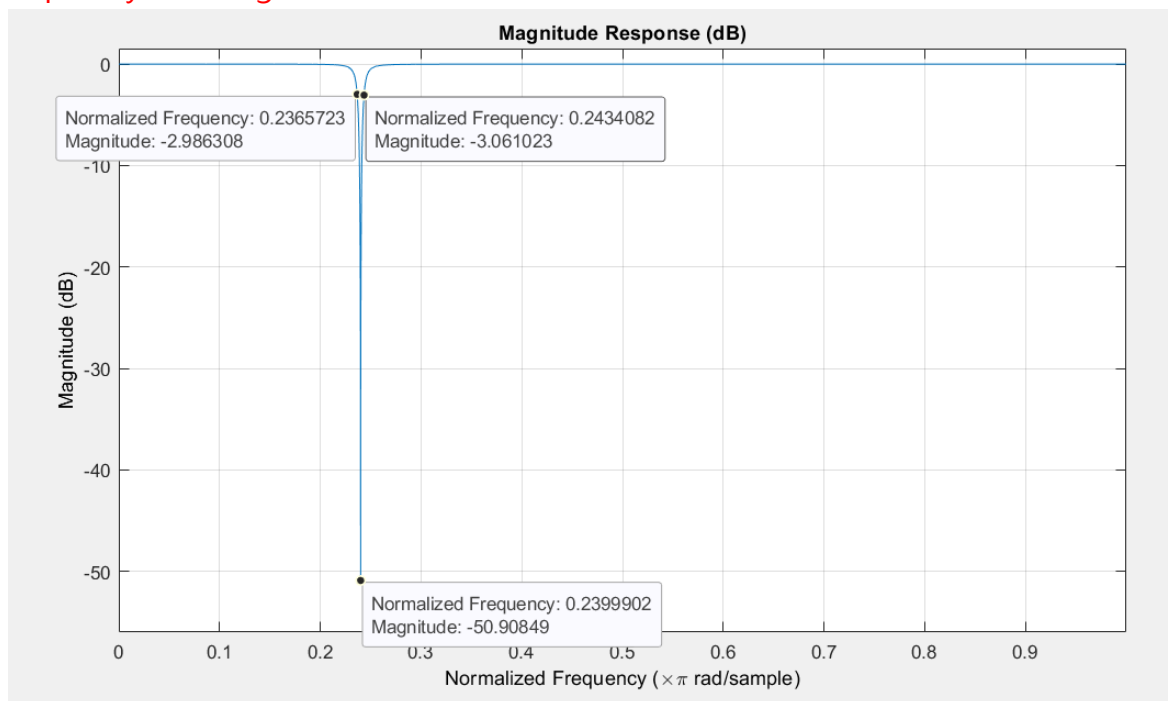


3. FIR notch filter

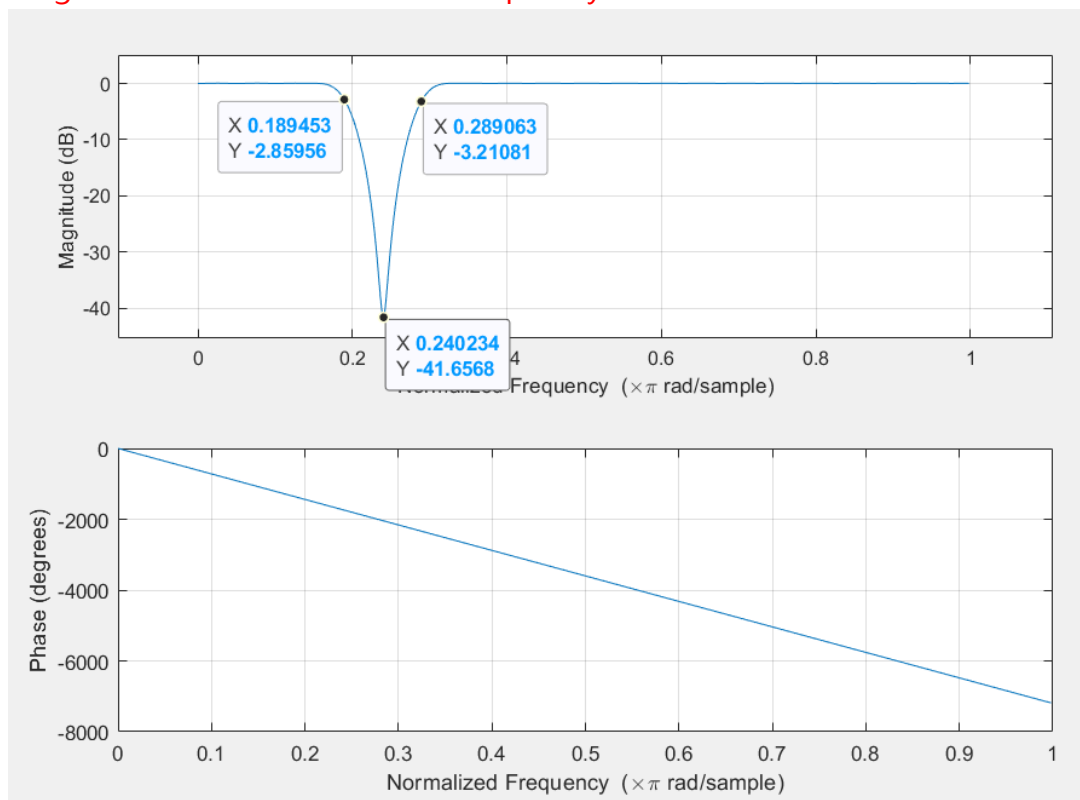


從時域就能看出三者對於 power line noise 都有達到壓抑的效果，其中又以 moving average filter 的效果最好。原因可以從頻域中發現，moving average filter 不只壓抑了 60Hz 的 power line noise，連同他的 harmonic 也一起壓抑了。另外，我發現 IIR notch filter 對於 60Hz power line noise 的壓抑效果比 FIR notch filter 的效果還差，我認為原因在於如下圖所示，IIR filter 的 notch frequency 沒辦法完美對到 60Hz 上，再加上我們設計的 IIR notch filter 非常窄，因此只要有一些誤差就會導致 notch

frequency 附近的 gain 差異很大，因此才會出現 60Hz 沒被壓抑很多的現象。

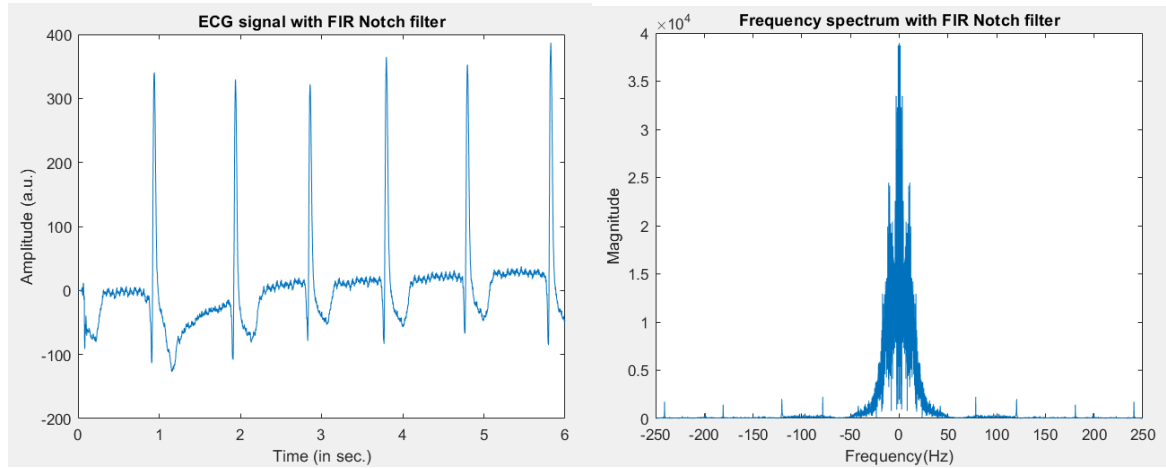


相反的，我們設計的 FIR notch filter 的 passband 較寬(如下圖)，notch frequency 附近的 gain 變化不大，就算 notch frequency 不準確，也能有不錯的壓抑效果。



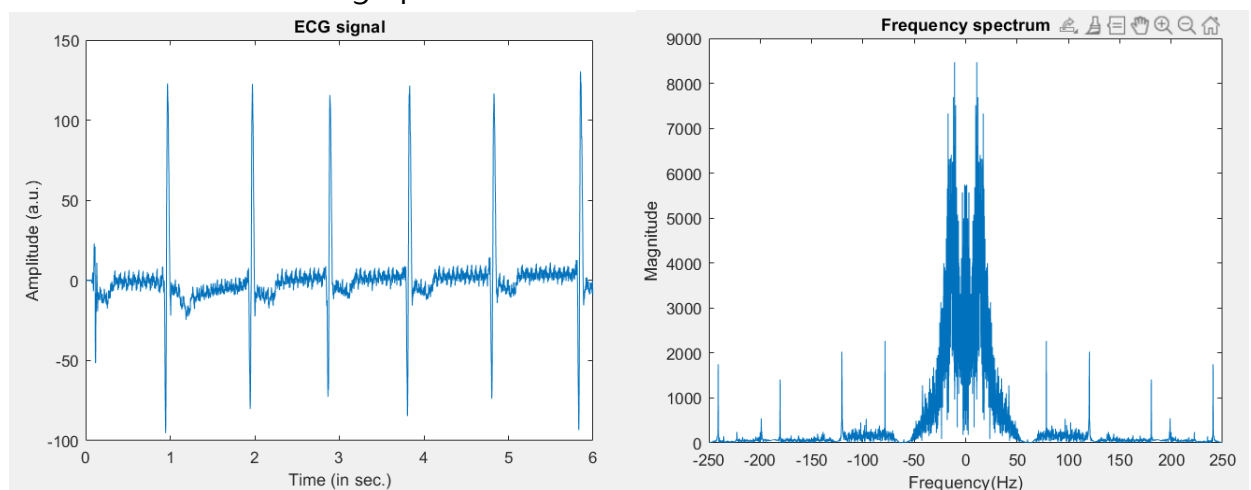
最後，我們也發現就算是 moving average filter 都沒辦法完全將 60Hz power line noise 壓抑掉，因此我們可以考慮串接多個 filter 來強化 filter 的能力。下面我們就嘗試將 moving average filter 與 FIR notch filter 做串聯。

4. Moving average filter + FIR notch filter



可以看到串接後的效果又更好了，從頻域也可看出 60Hz power line noise 被壓抑到很小。在時域上我們也能清楚看出 ECG 各區域的波形。這部分我們也能發現我們量測到的 ECG 訊號有些微的 baseline wander noise，所以接下來我們就來處理 baseline wander noise。

因為 baseline wander noise 是非常低頻的訊號，所以我們就用一個 high pass filter 來過濾掉這個雜訊。具體的 high pass filter 設計如上面所述，結果如下：



可以發現 baseline wander noise 被過濾掉很多，但同時可以發現 P 和 T 兩區域的 noise 多了一些。原因在於 low pass filter 也會同時壓抑部分 ECG 訊號，讓 power line noise 相較於 ECG 又變大了。

Q3) Digital filter 和 Analog filter 的效果有哪些異同？如何選擇要使用 Digital filter 或者是 Analog filter ？

Digital filter 透過數位系統中的位元運算來實現，他的優點在於訊號處理的過程中不太受 noise 的影響，再加上是用位元來儲存數值，做出來的 transfer function 是相當準確，不會因元件的 process variation 而變，精確度提高很多(ex.我們能很精確地做出某個頻率的 notch filter)。此外，在 Digital 之下複雜的演算法也較容易實現。他的缺點則在於受硬體上效能的限制，例如我們很難去處理一些極端的 case(極低頻需要極大的取樣點→超過硬體儲存空間；極高頻的取樣率受系統的 clock cycle 限制可能會有 aliasing 產生→所需取樣週期 < clock per instruction(CPI))，運算速度也比 Analog filter 慢(但目前 VLSI 技術進步

所以差異不大)。在大多數的情況下，我們會 prefer Digital filter 因為他精確度高且抗噪比強，Analog filter 則適合一些 lower-order 的簡易 filter 設計(ex. 通訊中的 envelop detector)，可以用 low cost 達成不錯的效果。另外，在非常高頻的頻段中，Analog filter 也比 Digital filter 合適，原因就來自前面說的系統效能極限。

Q4)能否推導 Digital IIR filter 的 transfer function ?

這部分我介紹一個從 Analog filter 的 transfer function 推導出 Digital IIR filter 的 transfer function 的方法，叫做 Impulse Invariant Transform。

假設我們現在有一個 Analog Filter，他的 transfer function 是 $H(s)$ 。假設 $H(s)$ 如下：

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{i=0}^M b_i s^i}{\sum_{j=0}^N a_j s^j} = \sum_{i=1}^N \frac{K_i}{s - s_i}$$

s_i 是這個 analog system 的 i -th pole，我們假設這是一個 stable system (所有 pole 都在左半平面)，我們就能去做 inverse Laplace transform，我們得到

$$h(t) = \sum_{i=1}^N K_i e^{s_i t} \quad t > 0$$

impulse invariant transform 的想法就是去對 impulse response 做取樣，也就是

$$h[n] = h(nT) = \sum_{i=1}^N K_i e^{s_i nT} \quad n = 0, 1, 2 \dots$$

接著再做 z transform 就能得到 Digital filter 的 transfer function

$$H(z) = \sum_{i=1}^N \frac{K_i}{1 - e^{s_i T} z^{-1}}$$

這等價於將 s-plane 上的 pole s_i map 到 z-plane 上的 pole z_i

$$z_i = e^{s_i T}$$

我們就能得到等效的 Digital filter 的 transfer function。這仍然是一個 stable system 因為 $\text{Re}\{s_i\} \leq 0 \leftrightarrow |z_i| \leq 1$ 。要注意的是對兩個串聯 system 做 impulse invariant transform 並不一定會等於對個字做 impulse invariant transform 再做串聯。

● Conclusion

在這個 Lab 中，我們實作各種不同的 FIR filter 和 IIR filter 來處理 ECG 中的 power line noise 以及 baseline wander noise。過程中我們討論了各種 filter 成果上的異同，比較其優劣，也探討該如何從 Analog filter 推導出 Digital filter 的設計。另外，實作一個演算法讓我們在改變 sampling bit 情況下仍能維持 maximum dynamic range，這個能夠用在後續的系統若不是一個十位元(Arduino 預設的 sampling bit 數量)的系統的情況下，能夠使量測的結果更為精準。

過去在修 DSP 導論時，對於 IIR 和 FIR filter 的感覺並沒有特別深刻，沒想到在實作上兩者有相當大的差異，效果也不太一樣。Analog \rightarrow Digital 的 impulse invariance transform 則是全新的東西，相較於修課時分析 filter，設計 filter 的難度高上許多，相當有趣。

● References

- 教授與助教的講義
- 李祈均教授的DSP講義
- 李夢麟教授的訊號與系統講義
- 維基百科—Impulse invariant method, IIR filter, FIR filter
- Moving Average Filter—[Frequency Response of the Moving Average Filter \(berkeley.edu\)](http://www.berkeley.edu)
- Impulse Invariant Transform : [Impulse Invariant Method \(stanford.edu\)](http://stanford.edu)
- MATLAB : fir1(), iirnotch(), iircomb()