

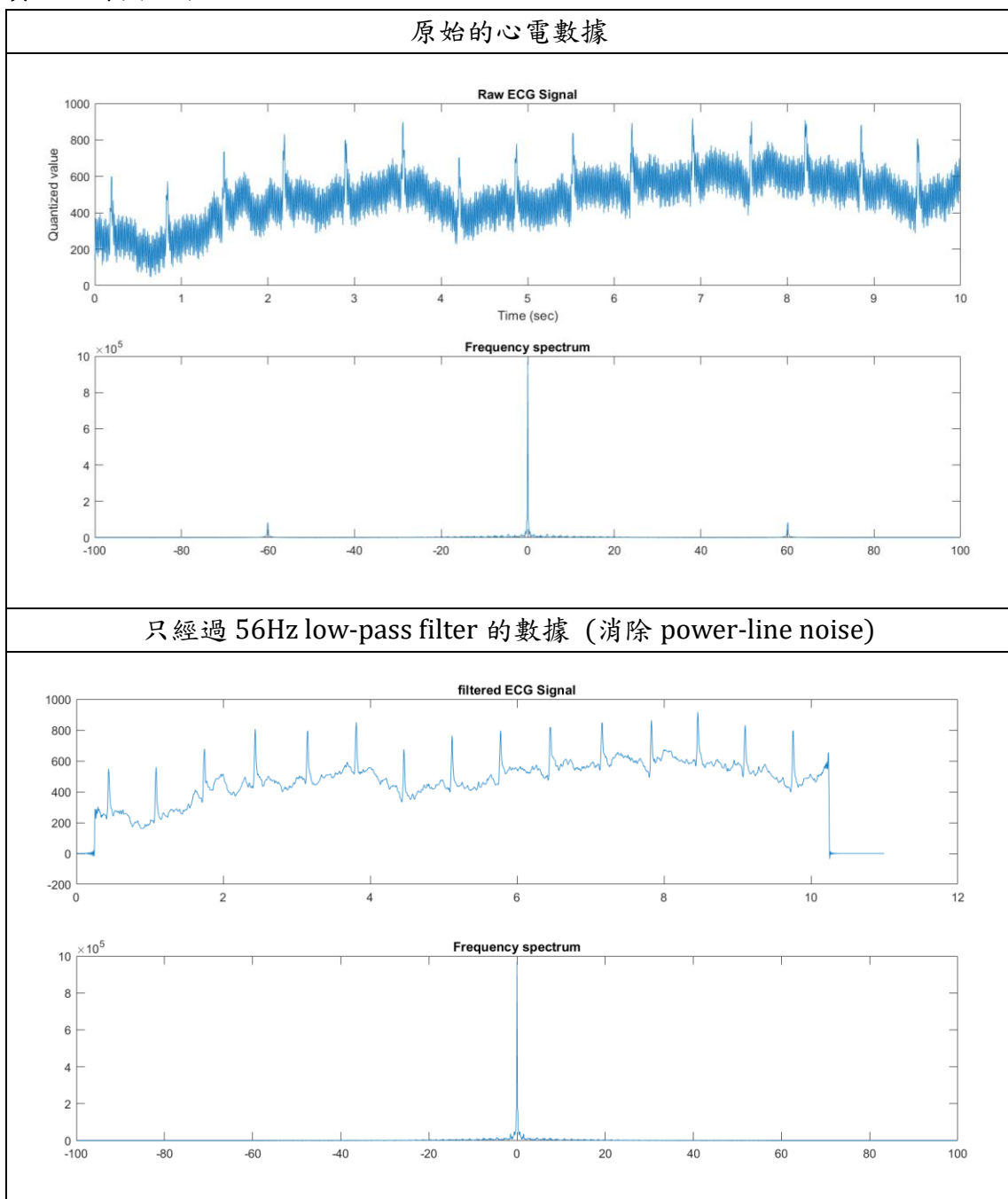
Digital Signal Processing Laboratory

Lab 3 Preprocessing of ECG Signals

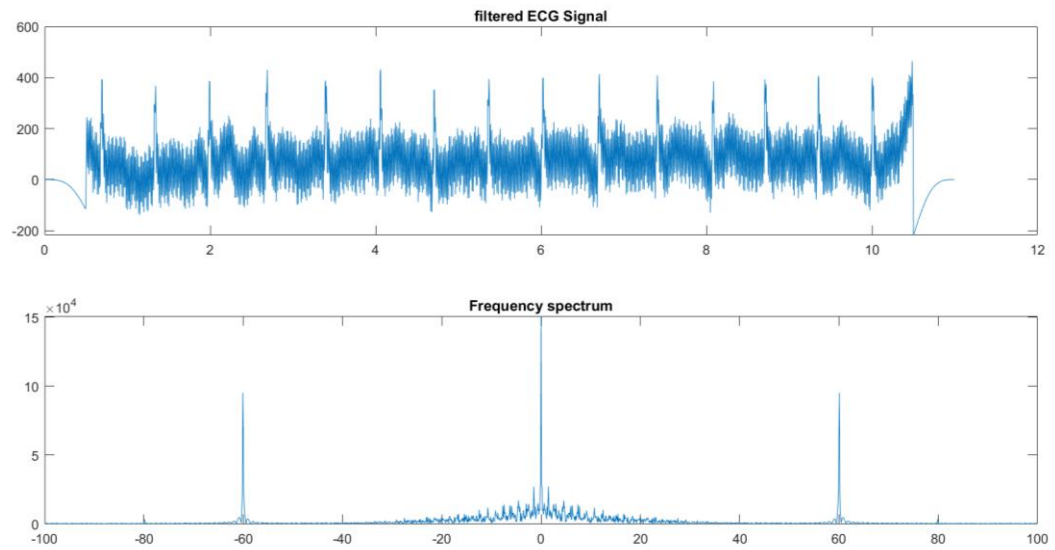
一、實驗介紹與目的：

不同於 Lab2 利用實體電路接出 60Hz notch filter，此實驗利用 Matlab 程式做出兩個數位濾波器，一個是代替 Lab2 的 60Hz notch filter、另一個是 1Hz 的 high pass filter，用來消除 Baseline noise。另外，我們也透過更改 Arduino 的解析度與取樣電壓範圍來最大化輸入訊號的動態範圍，以增加取樣效率並減少所需記憶體大小。

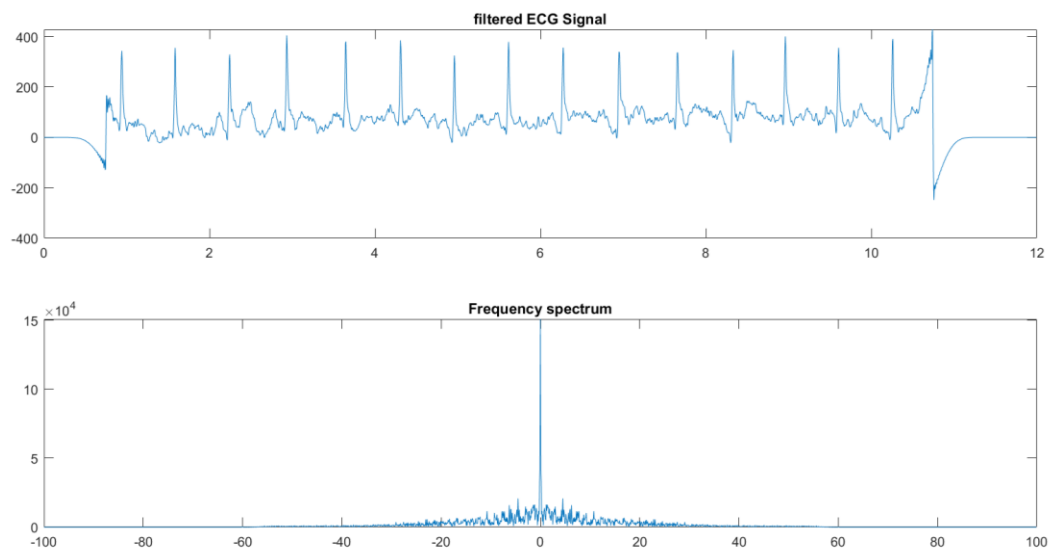
二、實驗數據與結果：



只經過 1Hz high-pass filter 的數據 (消除 base-line noise)

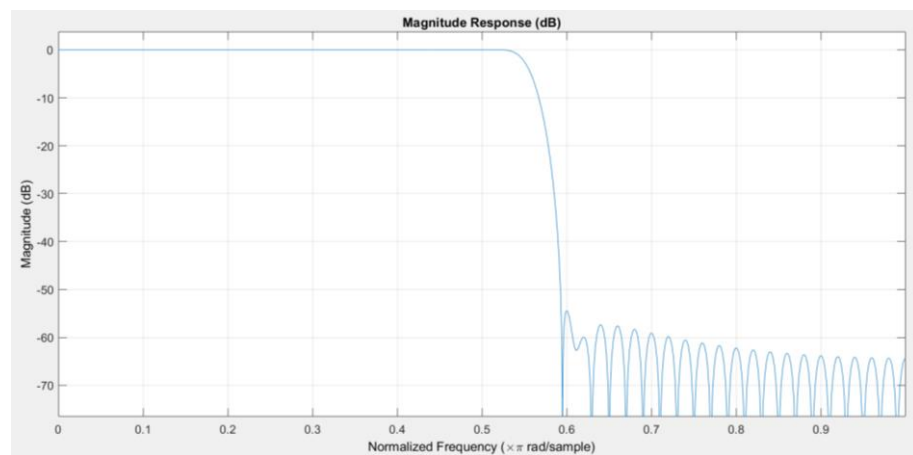


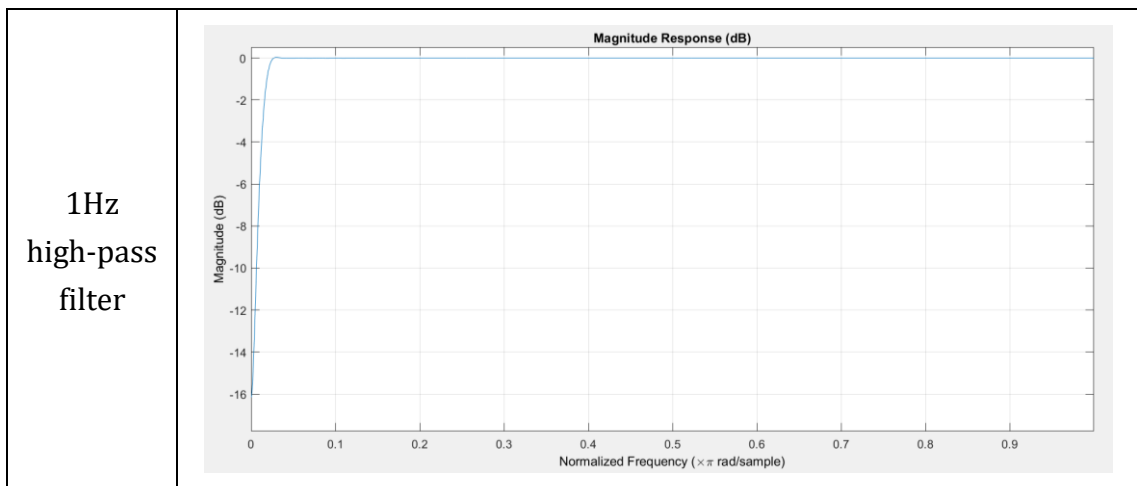
經過 low-pass 與 high-pass filter 後的最終數據



Frequency Response

56Hz
low-pass
filter





三、問題與討論：

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

此 ECG signals 在 time domain 可以看出其基準線有高低起伏，代表有受到 base-line noise 的影響；另外在每個 R peak 之間也有許多頻率較高的雜訊，使得波形中完全看不到 P 波與 T 波。在經過 Fourier transform 之後，從 frequency domain 可看到在 0Hz 附近的地方有很高 magnitude，此為 DC 直流成分以及 base-line noise；另外在正負 60Hz 的地方也有突起，代表有 power-line noise，而因為此 data 的取樣頻率為 200Hz，所以在頻率響應中只會顯示到正負 100Hz 的頻率成分，故看不到 120Hz, 180Hz 之後的 power-line harmonic noise。

2. Please elaborate what kind of digital filters you implemented and how you design your digital filters to remove noises in your report.

為了消除 power-line noise 造成的 60Hz 雜訊，我用了一個 56Hz 的 low-pass filter 來達成。雖然老師上課時是希望我們使用 notch filter 來濾掉 60Hz 的訊號，但由於 ECG 訊號中有用的訊號(我們所在意的訊號)幾乎都低於 50Hz，所以我認為沒有必要做出一個較複雜的 notch filter 而只是濾掉 60Hz 的 power-line noise；若是使用低於 60Hz 一些的 low-pass filter，不但能濾掉 60Hz 的 power-line noise，還能濾掉其他沒用且高於 60Hz 的雜訊。

我透過 Matlab 中的 `filter1=fir1(100, 56/fs*2, 'low')` 來做出用來消除 power-line noise 的低通濾波器，為了讓它的 frequency response 更像理想的低通濾波器，但又不希望 order 數太高增加濾波器複雜度以及訊號 delay 程度，因此經過調整後選擇 order=100。另外，雖然理論上要消除 60Hz 以上的雜訊只要設定截止頻率為 60Hz 就好，但由於濾波器的不理想性，讓他在 60Hz 的地方只有不到 -3dB 的 magnitude，再加上原本訊號在 60Hz 的頻率成分很高，因此 -3dB 的 magnitude 無法濾掉大部分的 power-line noise，也因此我將截止頻率往前移一些到 56Hz，如此便可以讓 60Hz 以上的訊號有著超過 -50dB 的 magnitude，以最大化此低通濾波器的特性，當然，這樣也會影響到 55~59Hz

範圍內的訊號成分，不過這段區間的頻率並不重要。

為了消除在 0Hz 附近的 base-line noise，我用 1Hz 的 high-pass filter 來達成，透過 `filter2=fir1(200, 1/fs*2, 'high')` 來建立。由於 high-pass filter 的複雜度相較於 low-pass filter 高，需要較高的 order 才能讓截止頻率之前的訊號有足夠低的 magnitude，且從原本的 ECG 資料的頻譜圖也可以看出，在 0Hz 附近的頻率成分遠比真正的 ECG 訊號頻率、甚至是 60Hz 的 power-line noise 高出許多，所以高通濾波器會需要更高的 order 來濾掉大部分的 base-line noise，因此這裡設定 `order=200`。另外，因為 `fir1()` 中的橫軸是 normalize frequency，單位是 $(*\pi \text{ rad/sample})$ ，所以頻率需要除以取樣頻率的一半。

3. Please elaborate what algorithm you implement in Arduino to maintain the maximum dynamic range of the input ECG signal with 8-bit sampling.

將 10bit 的解析度改為 8bit 並根據 ECG 訊號選定輸入電壓的範圍可以降低資料傳輸所需的速率以提升傳輸效率，因此，我利用 Arduino 中的 `map()` 語法來達成此演算法。有別於之前降低解析度的方式，是將輸入值的範圍從原本的 0~1023 轉換成 0~255，這次是先觀察接收到的 ECG 訊號範圍大約落在 150~750 之間(電壓範圍 5V，解析度 10bit，因此 ECG 訊號電壓範圍大約為 0.73V~3.66V)，而因為在 100 以下和 800 以上的區域都沒有用到，所以將 mapping 的輸入設為 100~800；接著因為我們要將解析度降為 8bit，所以在 mapping 的輸出端設定範圍為 0~255，如此一來，就能達成最大化 ECG 資料範圍的功能了。

四、結論與心得：

此實驗用軟體程式實作出各種濾波器，以過濾掉各種不同的雜訊，讓接收到的 ECG 訊號能夠更清晰更乾淨，且最令我驚豔的是上次實驗花了一個小時才接完濾波器電路，這次實驗只花了十分鐘幾行程式碼就搞定，可見軟體程式的強大。