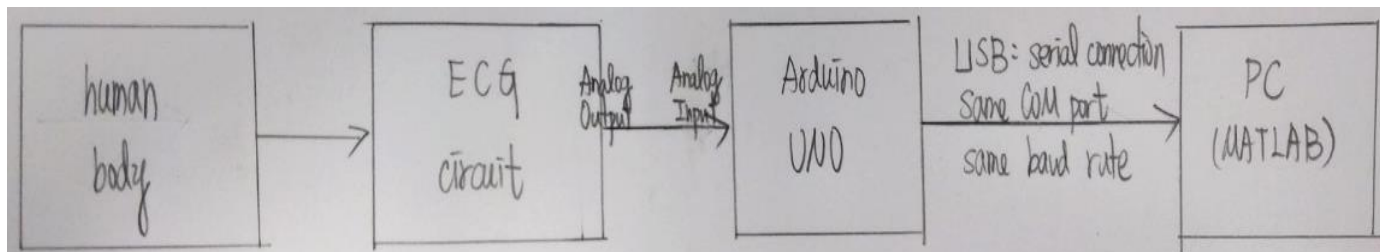


105061254 林士平 數位訊號處理實驗報告 Lab4

1.

(1) Design Specification :



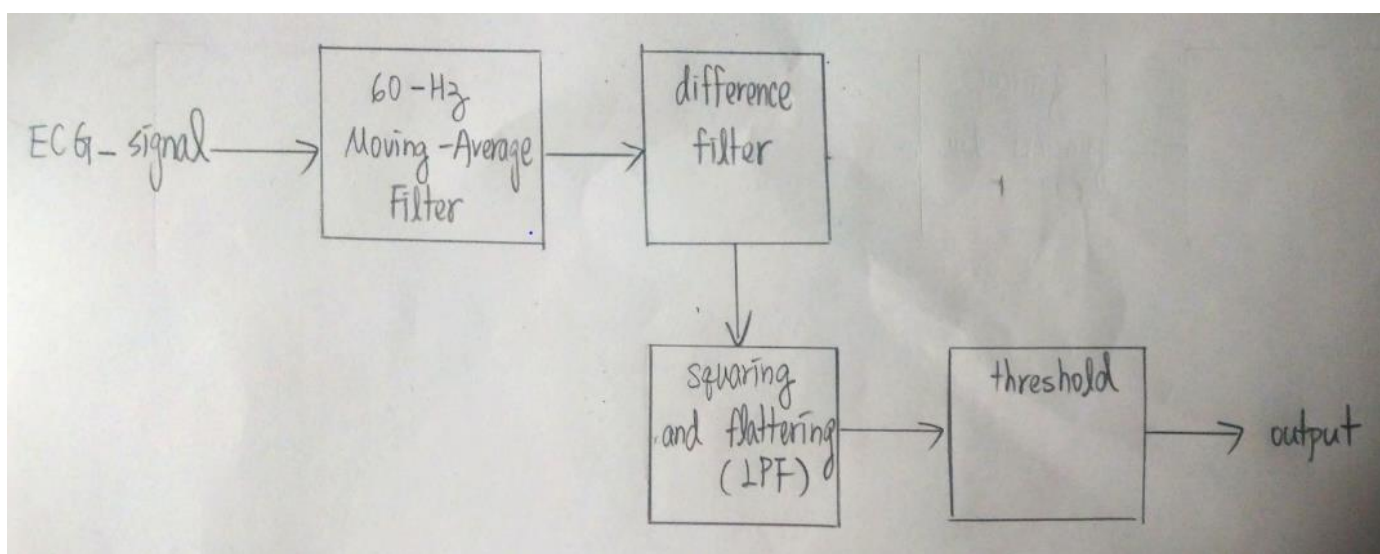
↑ 本實驗的 block diagram，其中 ECG 電路沒有使用 analog notch filter

(2) Experiment Objective :

- 找出我的 ECG 訊號的 R-peak。

(3) Results and analysis :

- MATLAB 程式的 block diagram



↑ MATLAB 程式的 block diagram

b. MATLAB 程式

本題我使用了兩種不同的寫法來找出 R-peak：

方法一：全部使用 for 迴圈做處理

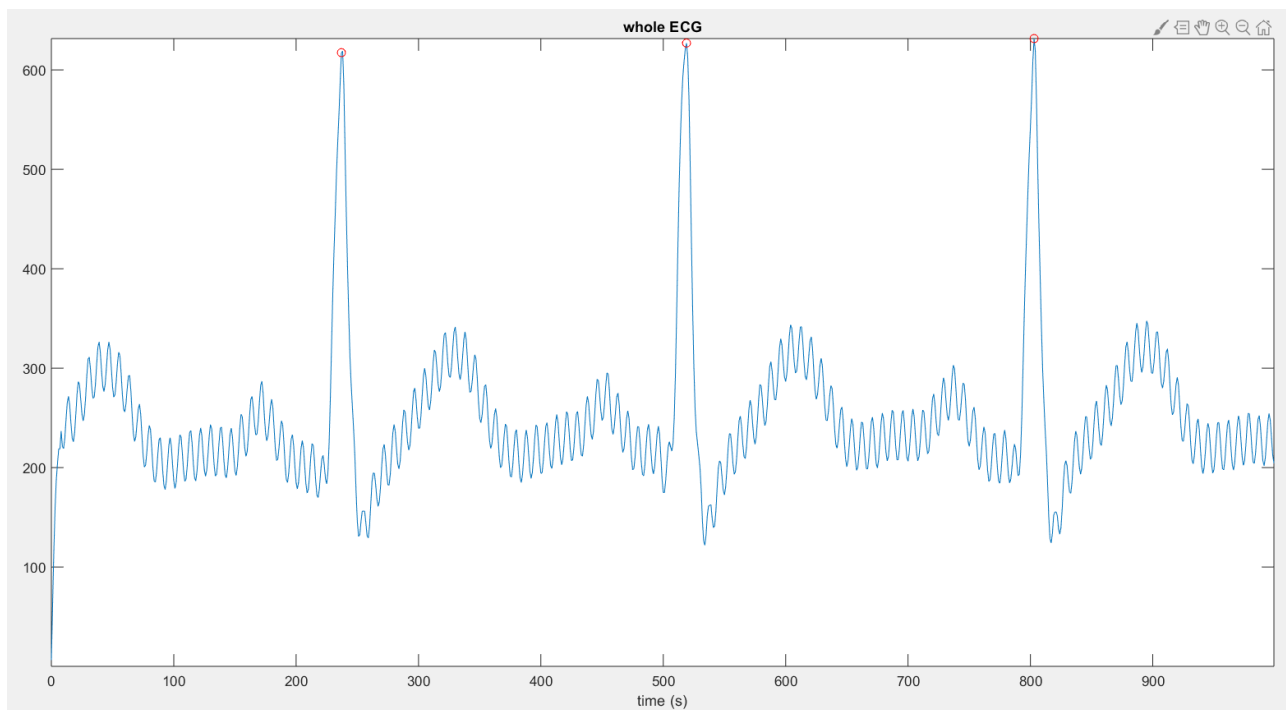
60-Hz moving average filter	<pre>% 60-Hz moving average filter (8-pt) pt = 8; N_new = 0; y2 = zeros(1, N); buffer = zeros(1,pt); for i = 1:(pt-1) buffer(i) = y(i); y2(i) = sum(buffer) / pt; end for i = pt:N for j = 1:(pt-1) y2(i) = y2(i) + y(i - j); end y2(i) = y2(i) / pt; end</pre>
Difference filter	<pre>% difference filter(slope calculation) y3 = zeros(1, N); y3(1) = y2(1); y3(2) = y2(2); y3(3) = y2(3); y3(4) = y2(4); y3(5) = y2(5); for i = 6:N y3(i) = y2(i) - y2(i - 1) + y2(i - 2) - y2(i - 3) + y2(i - 4) - y2(i - 5); end</pre>
Squaring	<pre>% squaring y4 = zeros(1, N); for i = 1:N y4(i) = y3(i) * y3(i); end</pre>
flattering	<pre>% flattering(moving average LPF) (-pt) pt = 15; N_new = 0;</pre>

	<pre> y5 = zeros(1, N); buffer = zeros(1,pt); for i = 1:(pt-1) buffer(i) = y4(i); y5(i) = sum(buffer) / pt; end for i = pt:N for j = 1:(pt-1) y5(i) = y5(i) + y4(i - j); end y5(i) = y5(i) / pt; end </pre>
threshold	<pre> % threshold thd = 10000; y6 = zeros(1, N); for i = 1:N if y5(i) > thd y6(i) = y5(i); else y6(i) = 0; end end </pre>
Find local maximum (R-peak) (利用前項-後項來判斷 local maximum)	<pre> % local maximum label = 1; delay = 9; %total group delay y7 = zeros(1, N); %final result for i = 2:N if label == 1 && (y6(i)-y6(i-1)<0) y7(i - 1) = y2(i - 1 - delay); label = 0; elseif label == 0 && (y6(i)-y6(i-1)==0) label = 1; end end </pre>

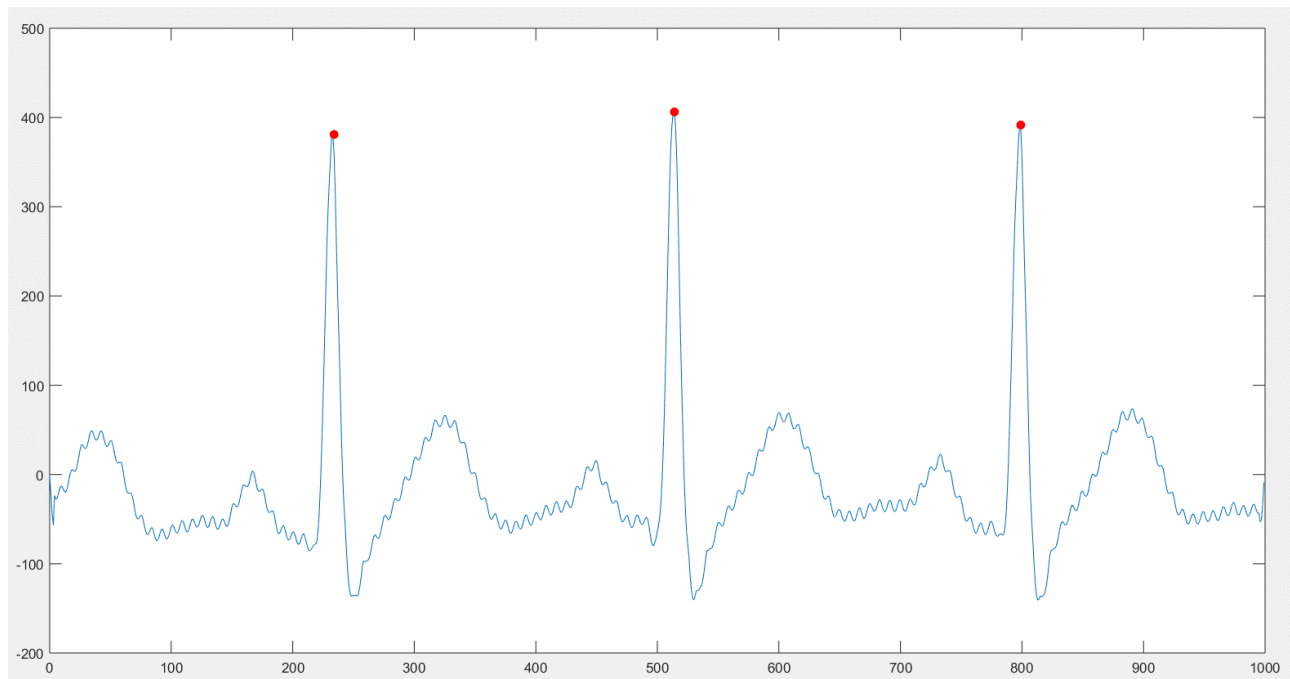
方法二：全部使用 MATLAB 函數做處理

60-Hz moving average filter	<pre>% 60-Hz moving average filter (8-pt) pt = 8; f1 = ones(1, pt) / pt; y2 = conv(y2, f1, 'same');</pre>
Difference filter	<pre>% difference filter(slope calculation) f2 = [1 -1]; y3 = conv(y2, f2, 'same');</pre>
Squaring	<pre>% squaring y4 = y3 .* y3;</pre>
flattering	<pre>% low pass filter pt2 = 23; f3 = ones(1, pt2) / pt2; y5 = conv(y4, f3, 'same');</pre>
Threshold and find local maximum (R-peak) (利用 findpeaks)	<pre>% threshold [~,locs_Rwave] = findpeaks(y5,'MinPeakHeight',1500,'MinPeakDistance',100);</pre>

c. 最終結果



↑ 利用方法一偵測 R-peak 的結果



↑ 利用方法二偵測 R-peak 的結果

d. 結果分析：

方法一程式較複雜，且處理 group delay 的方式較為麻煩(方法二使用 `conv('same')`，所以較沒有 group delay 的問題)，濾出來的訊號也沒有方法二乾淨，所以後面第二和第三題均會使用方法二。

2.

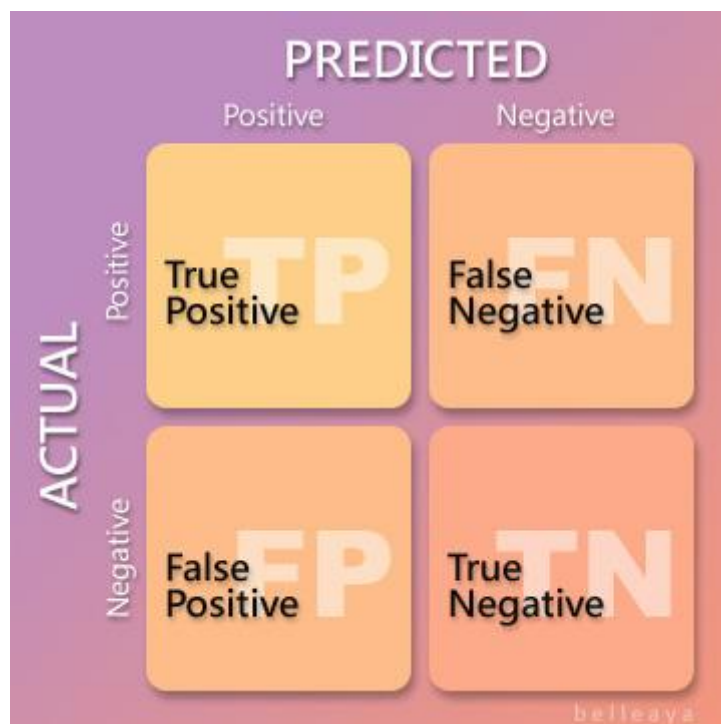
(1) Experiment Objective：

- 使用 1.設計出的演算法來找到 MIT-BIH database 的 R-peaks。
- 分析 TP, TF, FN, FP。

(2) Results and analysis：

a. TP, TF, FN, FP 表格：

	TP	FN	FP	TN
100m	945	1328	1329	646398
103m	662	1422	1424	646492
112m	1489	1050	1052	646409
117m	125	1410	1412	647053
122m	363	2113	2116	645408



↑ TP, TF, FN, FP 示意圖

b. MATLAB 程式：

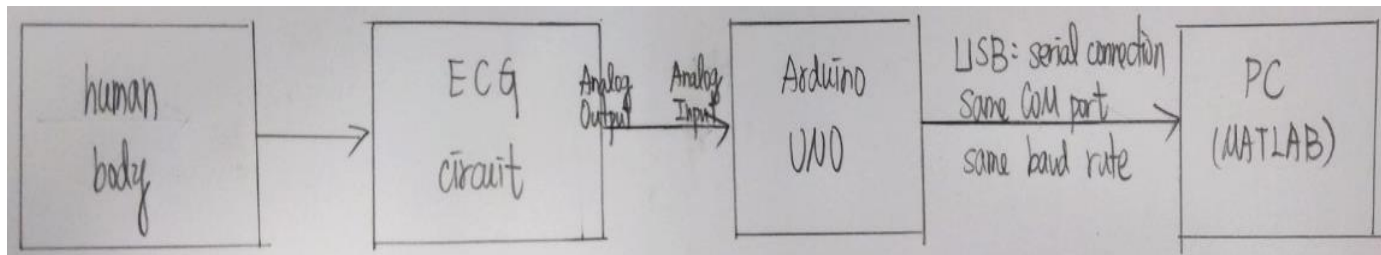
讀取 excel 檔	<pre>%read the data xlsFile = '100.xlsx'; [Data, headerText] = xlsread(xlsFile);</pre>
truth2 為 excel 檔中 R-peak 區間的累積總合，也就是 R-peak 出現的絕對時間	<pre>truth = Data'; truth2 = truth; result = 0; TP = 0; for i = 2:length(truth) truth2(i) = truth2(i) + truth2(i - 1); end</pre>
比較並找出 TP、TN、FN 和 FP	<pre>for i = 1:length(locs_Rwave_final) if truth2(1, i) == locs_Rwave_final(1, i) TP = TP + 1; end end TP FN = length(truth2) - TP FP = length(locs_Rwave_final) - TP TN = N - TP - FN - FP</pre>

c. 結果分析

這些資料中有許多非理想的 ECG 波型(例如：T interval 高到幾乎和 R-peak 一樣高)，因此造成錯誤率偏高的問題，不過我想這些問題如果時間夠多應該可以利用 try and error 來解決，例如：調整 threshold 值、調整最小的 max 值區間、調整 fluttering 的 Moving average filter 的點數等。

3.

(1) Design Specification :



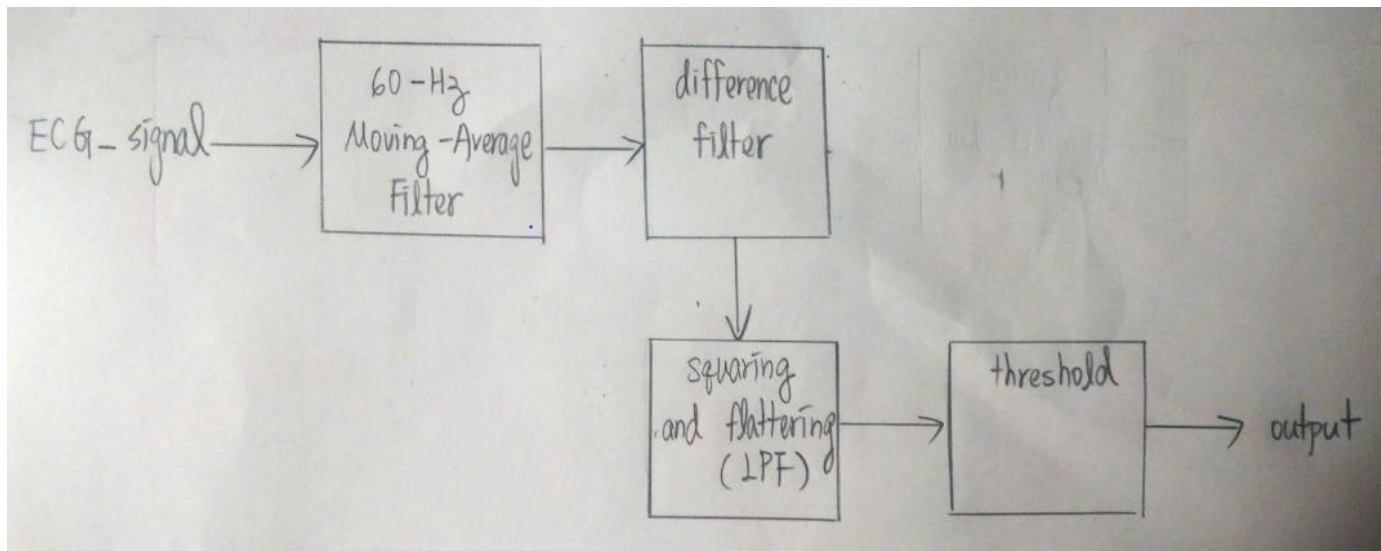
↑ 本實驗的 block diagram，其中 ECG 電路沒有使用 analog notch filter

(2) Experiment Objective :

- a. 實現即時的心電圖。

(3) Results and analysis :

- a. MATLAB 程式的 block diagram



↑ MATLAB 程式的 block diagram

b. MATLAB 程式

首先最外層有個大迴圈決定總共要跑幾個點。

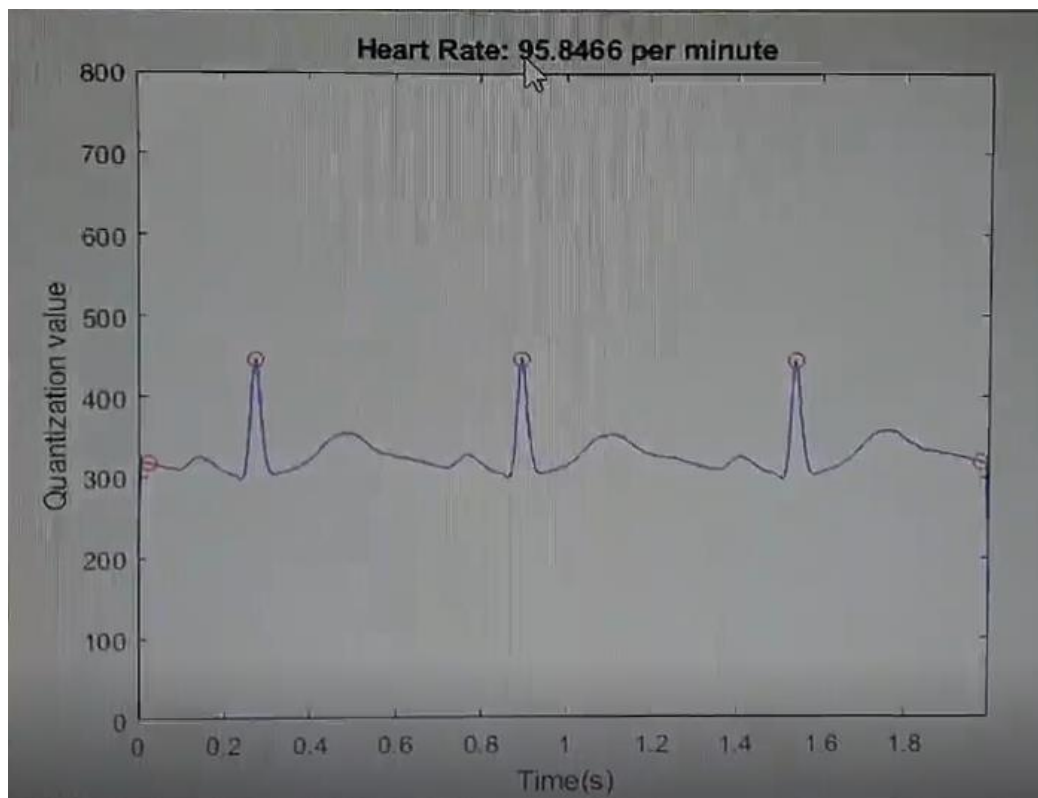
接著裏頭先有 60-Hz moving average filter、Difference filter、Squaring and Flattering 和 Threshold，此流程和第一大題方法二相同。

此部分為本題重點，因為使用 `conv` 函數，所以必須要累積一些點再做處理，累積到一定點數後畫出 ECG 訊號並找出 Heart rate(利用 60/R-peak time interval 找出一分鐘的心跳數)，最後顯示在螢幕上。

```
if i>1 && rem(i,40)==0
    plot(time,y2,'b',locs,y2(locsR),'ro');
    axis([ min(time) max(time) 0 800 ]);
    xlabel('Time(s)');
    ylabel('Quantization value');

    if i>700
        rate=60/(locs(3)-locs(2));
    end
    str=['Heart Rate: ',num2str(rate),' per
minute'];
    title(str);
    drawnow;
end
```

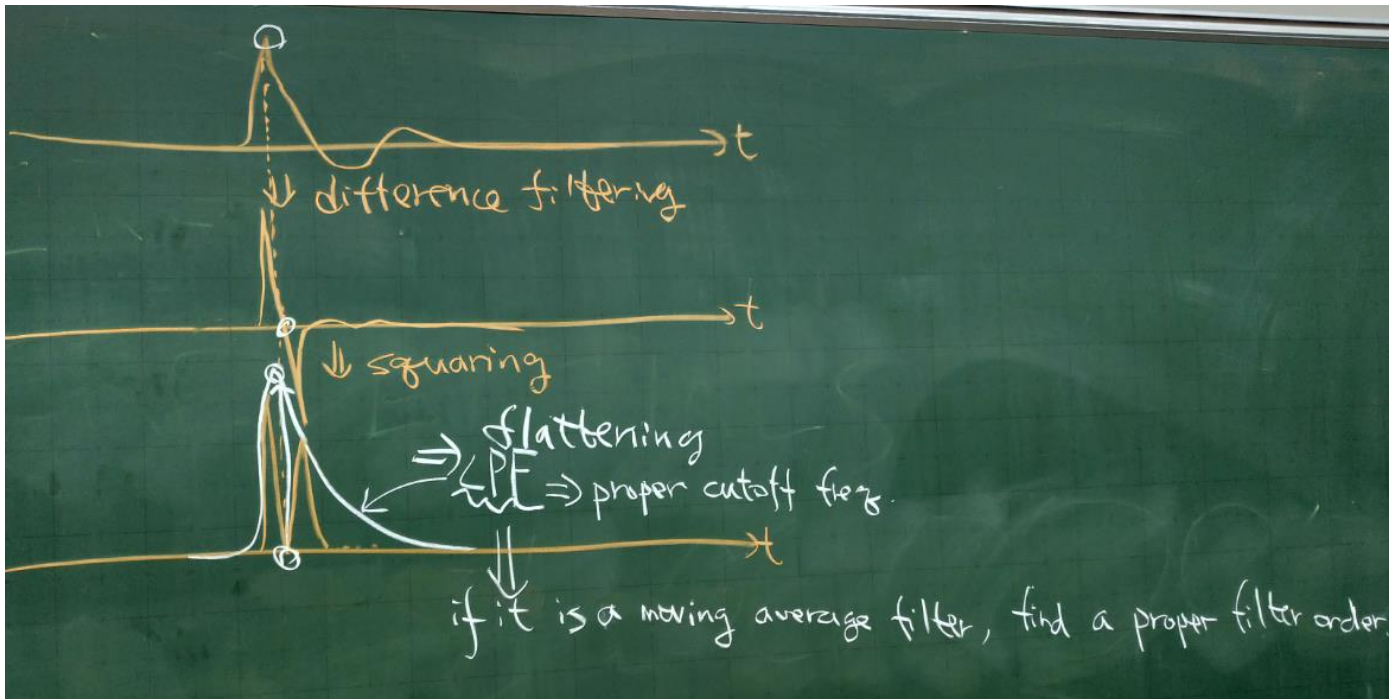
c. 結果分析



↑ ECG Demo 截圖，可以觀察到 R-peak 與心律

4. Discussion :

Why do we need flattening after squaring if you perform difference filtering or high pass filtering to remove baseline wandering?



由上圖可知，經過 difference filter 後的波型會有一個波峰和一個波谷，因此利用 squaring 讓波谷變成另一個波峰，然而我們想要找的 R-peak 實際上在兩個波峰正中間凹陷下去的地方，所以利用一個 LPF 讓凹陷下去的地方突起來，便能利用 threshold 找到最大值，也就是 R-peak 的所在。

5. Extra stuff :

我覺得 MIT 資料庫的資料真的太多，而且裏頭的波型千奇百怪，只有一個禮拜的時間根本不足以把裡面所有的資料給做好，而且資料的 ground truth 是 txt 檔，還要轉成 excel 檔非常麻煩，我認為這一大題可以把它拿掉，因為光是 1.3.題的 loading 已經夠重了。

6. Conclusion :

終於把 ECG 的部分結束了，雖然很累不過看到自己的 Real time 心電訊號還是覺得滿有成就感的，而且從中學到滿多數位訊號處理的實際應用與實際會遇到的問題，更多的是學到如何 debug 電路還有寫 Matlab 程式。