

逢甲大學學生報告 ePaper

音訊信號的隨機性程度測試

Testing the degree of randomness of audio signals

作者：蔡和佑

系級：自控三甲

學號：D0713967

開課老師：林育德

課程名稱：生醫信號處理

開課系所：自動控制工程學系

開課學年：109 學年度 第二學期



中文摘要

本篇報告主要在於驗證音訊信號是否為隨機性訊號，音訊信號主要分為三種：voiced、unvoiced、plosives。那麼判斷音訊信號隨機性程度的標準是基於此信號的轉折點是否大於某個數值。而在這篇報告中，驗證以及測試的方面是使用 MATLAB(2020a, MathWorks®, Inc., USA)。



關鍵字：峰谷值、窗框、轉折點、隨機性訊號

Abstract

This report is mainly to verify whether the audio signal is a random signal. There are three main types of audio signals: voiced, unvoiced, and plosives. Then the standard for judging the degree of randomness of the audio signal is based on whether the turning point of the signal is greater than a certain value. In this report, MATLAB (2020a, MathWorks®, Inc., USA) is used for verification and testing.



Keyword : Random signal, Peak and Trough, Turning Point ,Window

目 次

中文摘要	1
英文摘要	2
目次	3
一、研究目的	4
二、程式碼流程圖	5
三、測試訊號（一）	10
四、測試訊號（二）	12
五、Window 的影響	14
六、參考文獻	18



一、研究目的：

測試訊號的隨機性，首先要將語音訊號的峰谷值 (Peak and Trough) 找出來，接著找出轉折點 (Turning Point)，轉折點即是峰谷值數目的總和。

再來依照自己所設定的窗框 (Window) 將轉折點對於時間的圖畫出來，其中語音訊號大致上能分成三種：voiced、unvoiced、plosives。

至於隨機訊號的定義為

$$\text{Turning Point} > \frac{2}{3} \times (N - 2)$$

其中 N 為一個 Window 有多少筆資料。

舉例來說：取樣頻率 (fs) 為 8 KHz，而 Window 的寬度為 50 ms，則

$$N = 8000 \times 0.05 = 400$$

意思是說輸入的訊號每 0.000125 秒為一個點，那麼在 0.05 秒的區間裡面會有幾個點，因此 $0.05 / 0.000125$ ，得到 N 為 400。

二、程式碼流程圖：

```
1 - [y, fs]=audioread('safety.mp3');
2 - %sound(y, fs); % 播放此音訊
3 - time=(1:length(y))/fs; % 時間軸的向量
4 - %plot(time, y); % 畫出時間軸上的波形
5 - %xlabel("time(s)")
6 - %ylabel("Amplitude")
7
8 - fileName='safety.mp3';
9 - info=audioinfo(fileName);
10 - fprintf('檔案名稱 = %s\n', info.FileName);
11 - fprintf('壓縮方式 = %s\n', info.CompressionMethod);
12 - fprintf('通道個數 = %g 個\n', info.NumChannels);
13 - fprintf('取樣頻率 = %g Hz\n', info.SampleRate);
14 - fprintf('取樣點總個數 = %g 個\n', info.TotalSamples);
15 - fprintf('音訊長度 = %g 秒\n', info.Duration);
16 - Max=[];
17 - MaxPlace=[];
18 - Min=[];
19 - MinPlace=[];
20 - TurningPoint=[];
21 - i=2;
22 - j=1;
23 - k=1;
24 - h=0;
25 - MaxPoint=0;
26 - MinPoint=0;
27 - while i<length(y)
28 -     if y(i) > y(i+1) & y(i) > y(i-1)
29 -         Max(j) = y(i);
30 -         MaxPlace(j) = i;
31 -         j = j + 1;
32 -     end
33 -     if y(i) < y(i+1) & y(i) < y(i-1)
34 -         Min(k) = y(i);
35 -         MinPlace(k) = i;
36 -         k = k + 1;
37 -     end
38 -     if i == (fs*0.05)
39 -         for z = 1:(fs*0.05)
40 -             TurningPoint(z) = length(Max)+length(Min);
41 -         end
```

```
42 -     end
43 -     if i > (fs*0.05)
44 -         for c = 0:(fs*0.05)-1
45 -             h = i-c;
46 -             if y(h) > y(h+1) & y(h) > y(h-1)
47 -                 MaxPoint = MaxPoint + 1;
48 -             end
49 -             if y(h) < y(h+1) & y(h) < y(h-1)
50 -                 MinPoint = MinPoint + 1;
51 -             end
52 -             TurningPoint(i) = MaxPoint + MinPoint;
53 -         end
54 -         MaxPoint = 0;
55 -         MinPoint = 0;
56 -     end
57 -     i=i+1;
58 - end
59 - TurningPoint(i) = TurningPoint(i-1);
60 - plot(time,TurningPoint)
61 - xlabel("time(s)")
62 - ylabel("Turning point")
```

Step1:

先把訊號讀近來，並令它等於某個變數，我用 y 表示。

Step2:

接著處理時間軸，將它轉換成實際的時間。

Step3:

找出此訊號的基本資訊 (8~15 行)。

輸出範例結果為

```
檔案名稱 = C:\Users\User\Desktop\safety.mp3
壓縮方式 = MP3
通道個數 = 1 個
取樣頻率 = 44100 Hz
取樣點總個數 = 135933 個
音訊長度 = 3.08237 秒
```

Step4:

接著宣告可能使用到的陣列以及變數。

- ✓ Max[] 為峰的數值
- ✓ Min[] 為谷的數值
- ✓ MaxPlace[] 為峰值的位置
- ✓ MinPlace[] 為谷值的位置
- ✓ TurningPoint[] 為每個 Window 的轉折點個數
- ✓ MaxPoint 為峰值的數量
- ✓ MinPoint 為谷值的數量

其餘的為跑函式時需要用到的代數。

Step5:

先設立一個 while 函式，讓他從 2 跑到(訊號的長度-1)的地方，
扣掉前後兩點的原因是，訊號的頭尾無法確定是否為峰谷值。

Step6:

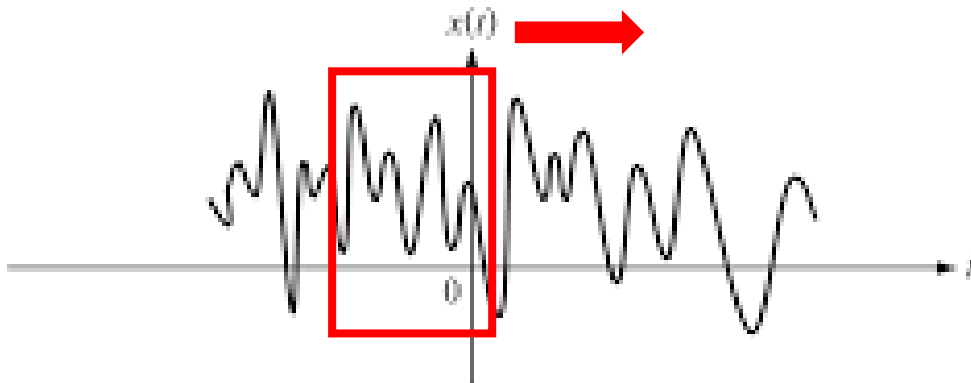
而我判斷峰谷值的依據為，與前後的值互相比較，如果此點比前後點都大，那它必為峰值；反之，此點比前後點都還小，那它必為谷值。

Step7:

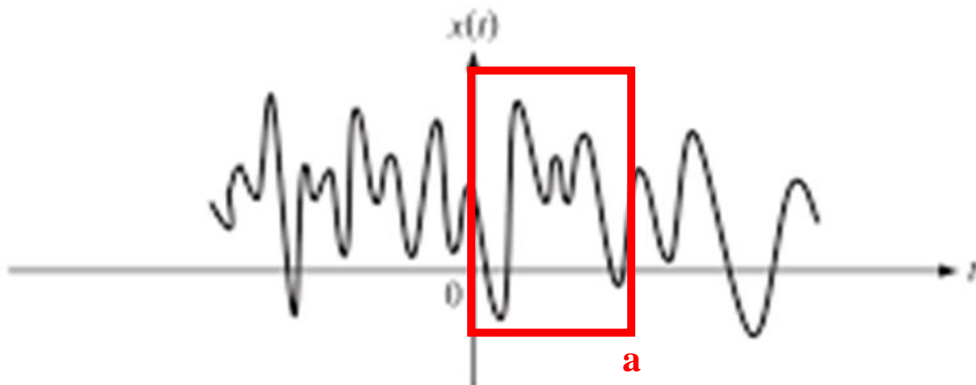
將找出來的結果用陣列存起來，我分別存了值以及位置，以便後續的處理。

Step8:

While 裡的第三個 if 用意在於，當參考點之前未滿 window 的長度時，我都要令它等於第一個 window 的結果。



(圖一)

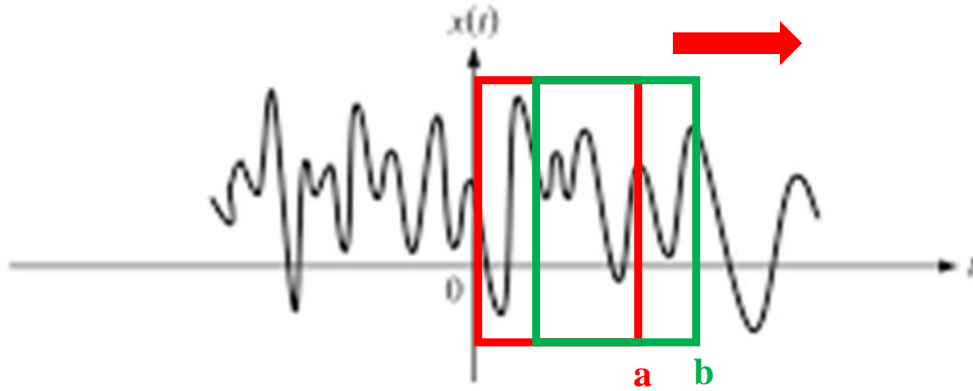


(圖二)

如圖一以及圖二所示。我要令圖一的结果等於圖二。

Step9:

While 裡的第四個 if 用意在於，當我的 window 開始符合我所要求的長度時，每經過一個點，就確認此 window 有幾個峰谷值。



(圖三)

如圖三所示，當我從 a 跑到 b 時，我就必須把 b 以前的峰谷值並且包含在綠框裡面的峰谷值統計出來，然後記錄在 TurningPoint 的陣列裡。

Step10:

當我記錄完一次 window 的轉折點有多少個時，必須將峰谷值的數目歸 0，進而作下次 window 轉折點個數的統計。

Step11:

最後，因為最後一點無法判別峰谷值，所以令它等於倒數第二個 window 的值。

三、測試訊號 (一)

Safety : _____

取樣頻率 = 44100 Hz

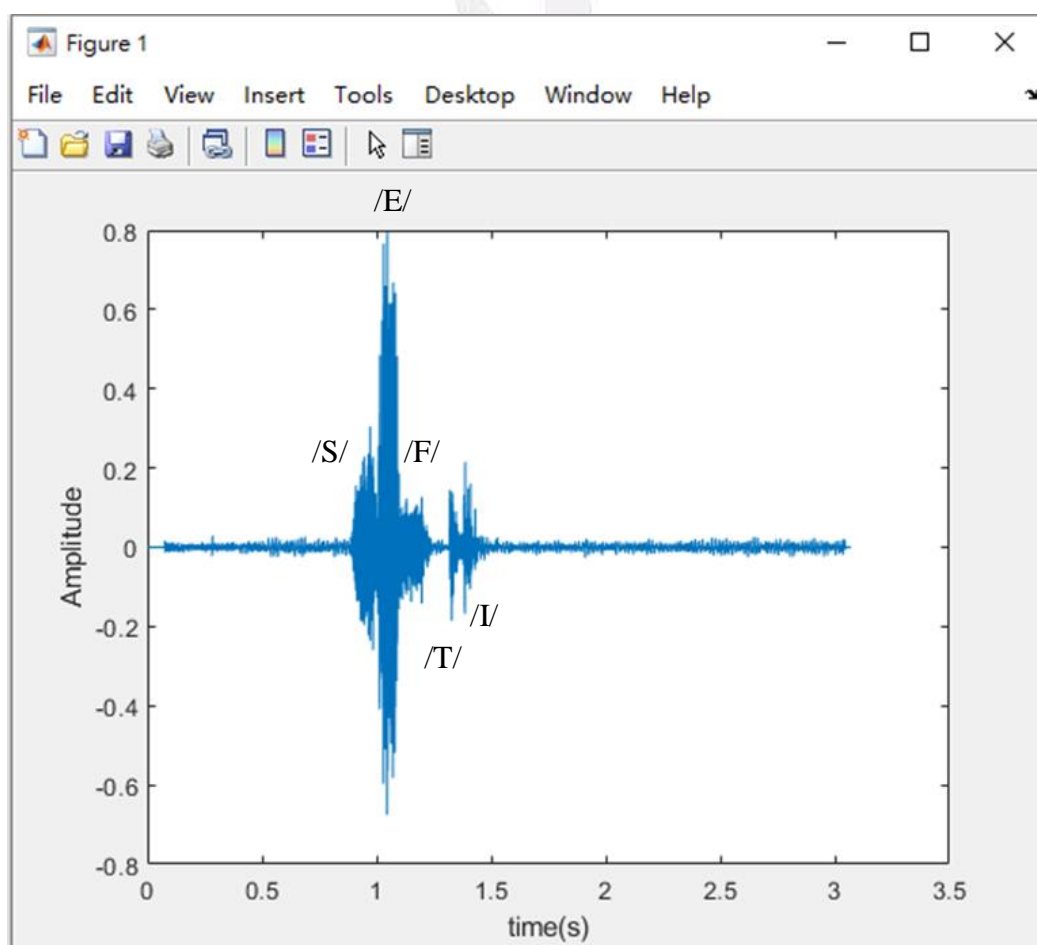
取樣點總個數 = 135933 個

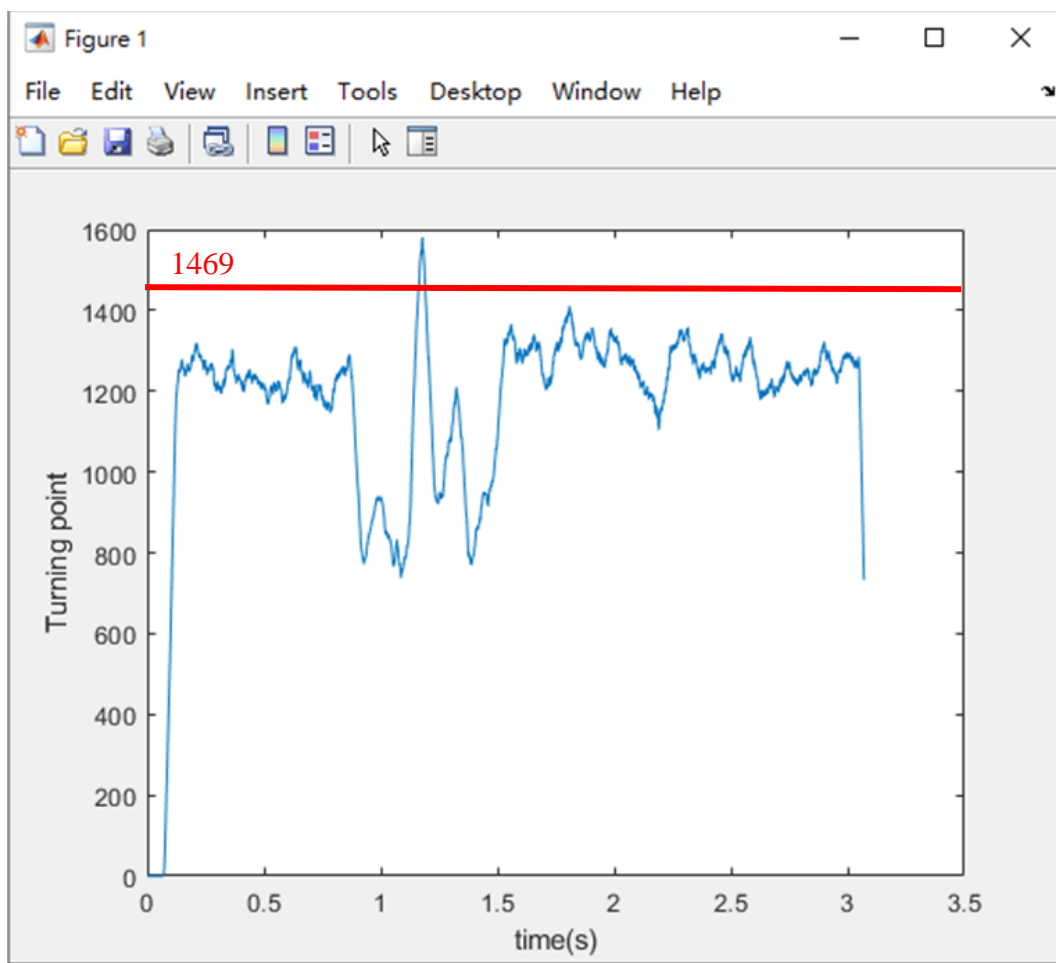
音訊長度 = 3.08237 秒

Window = 0.05 秒

N = 2205

隨機訊號的界線 = 1469





四、測試訊號 (二)

蔡和佑：

取樣頻率 = 44100 Hz

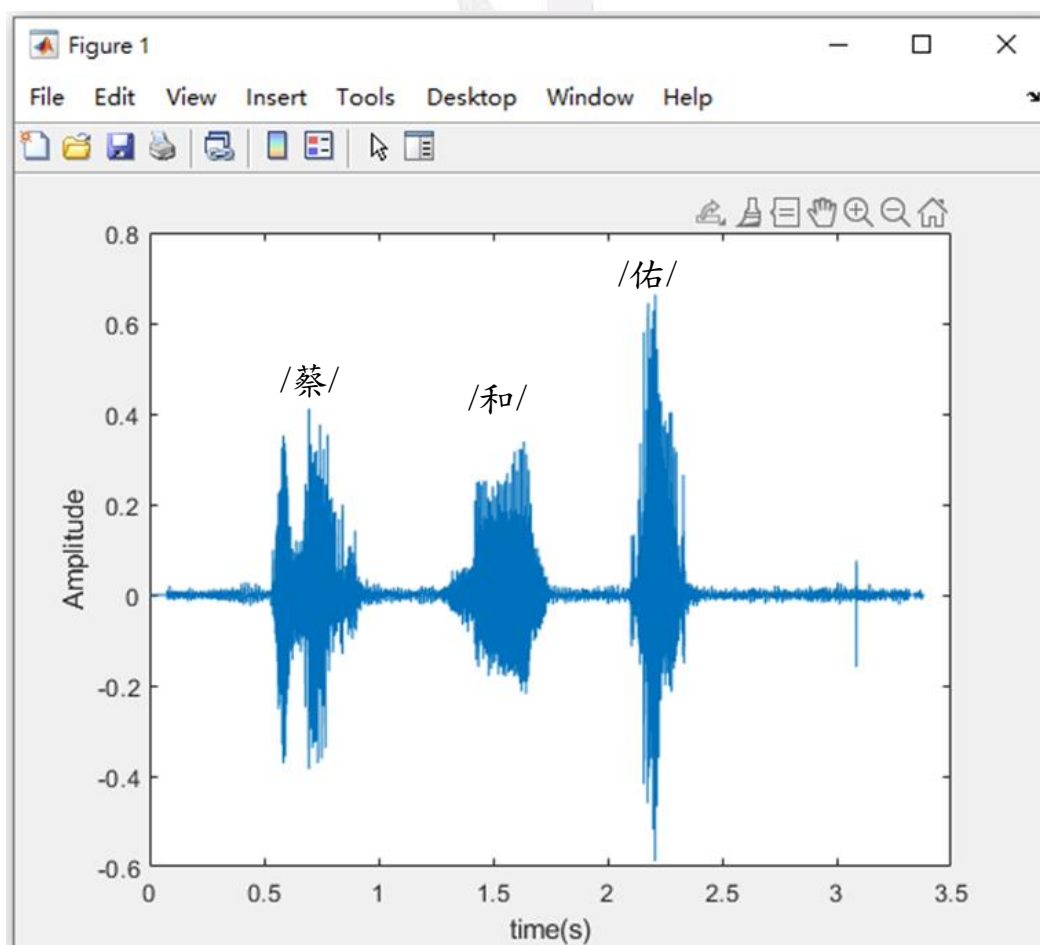
取樣點總個數 = 149755 個

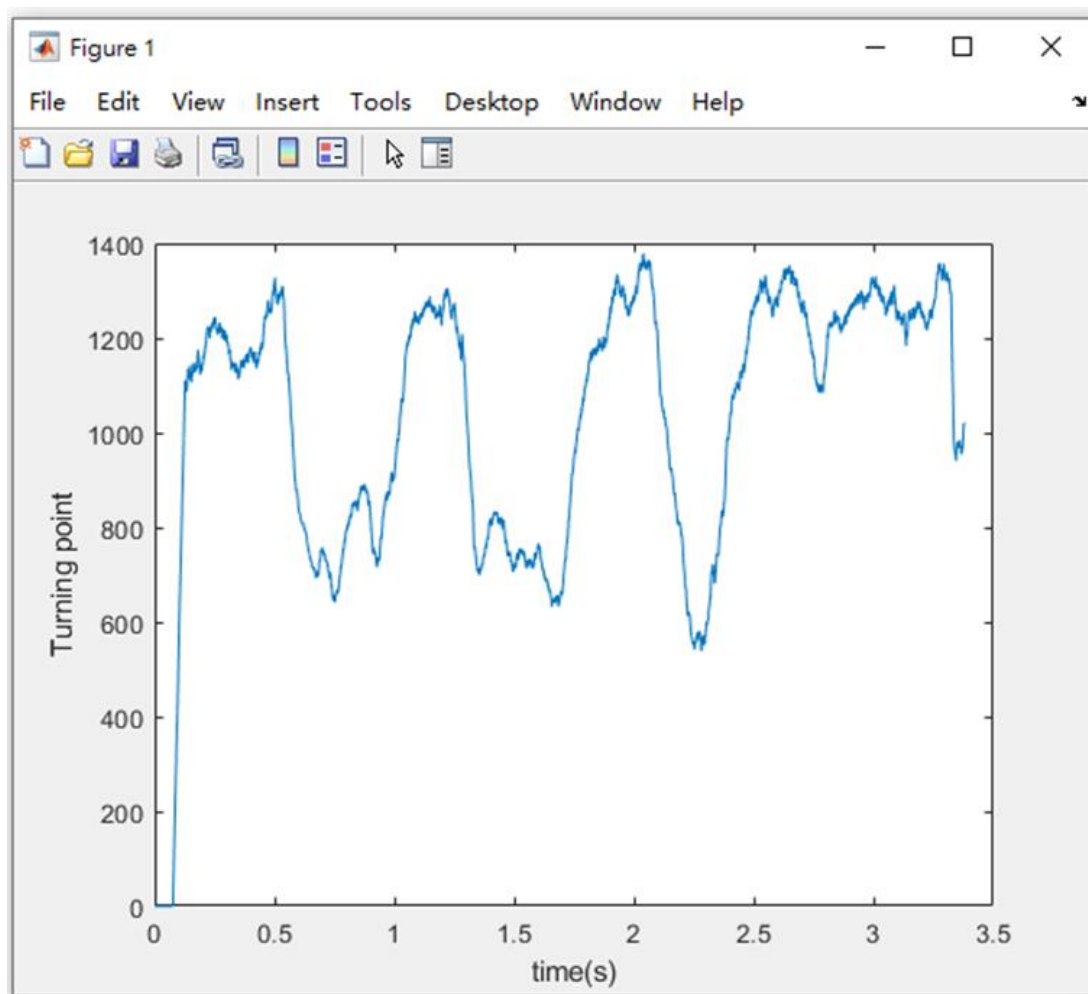
音訊長度 = 3.39581 秒

Window = 0.05 秒

N = 2205

隨機訊號的界線 = 1469



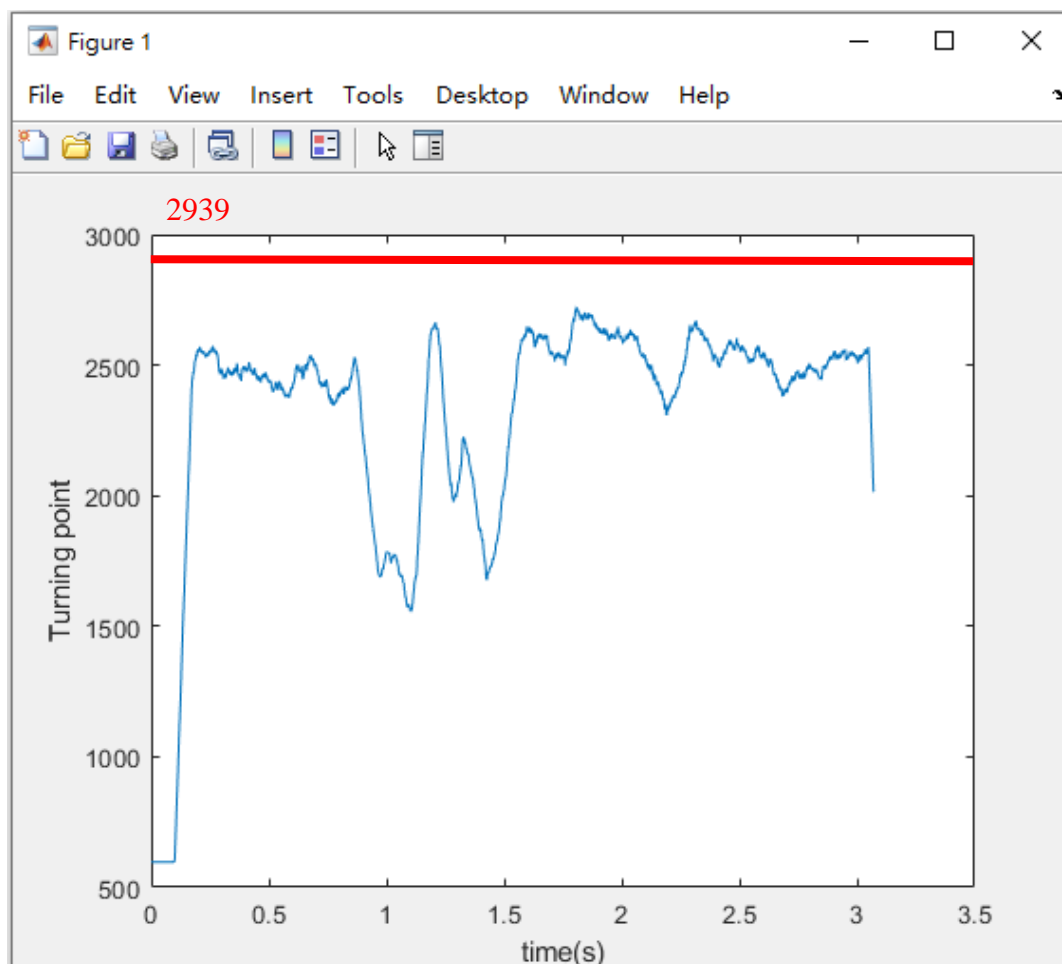


此訊號轉折點皆無超過 1469，故沒有隨機訊號。



五、Window 的影響：

當 window 的長度增加時，Safety 的訊號變成



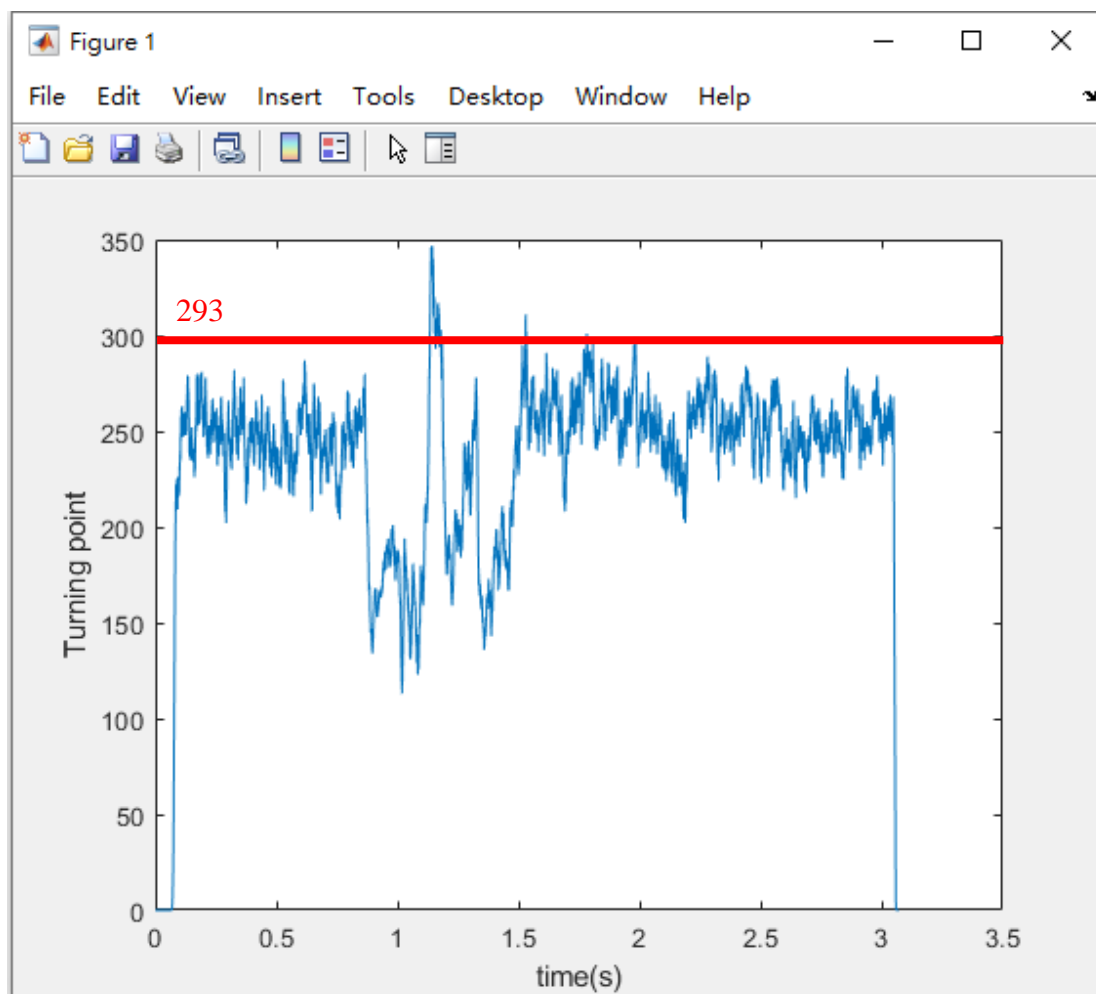
Window = 0.1 秒

N = 4410

隨機訊號的界線 = 2939

取樣頻率 = 44100 Hz

當 window 的長度縮短時，Safety 的訊號變成



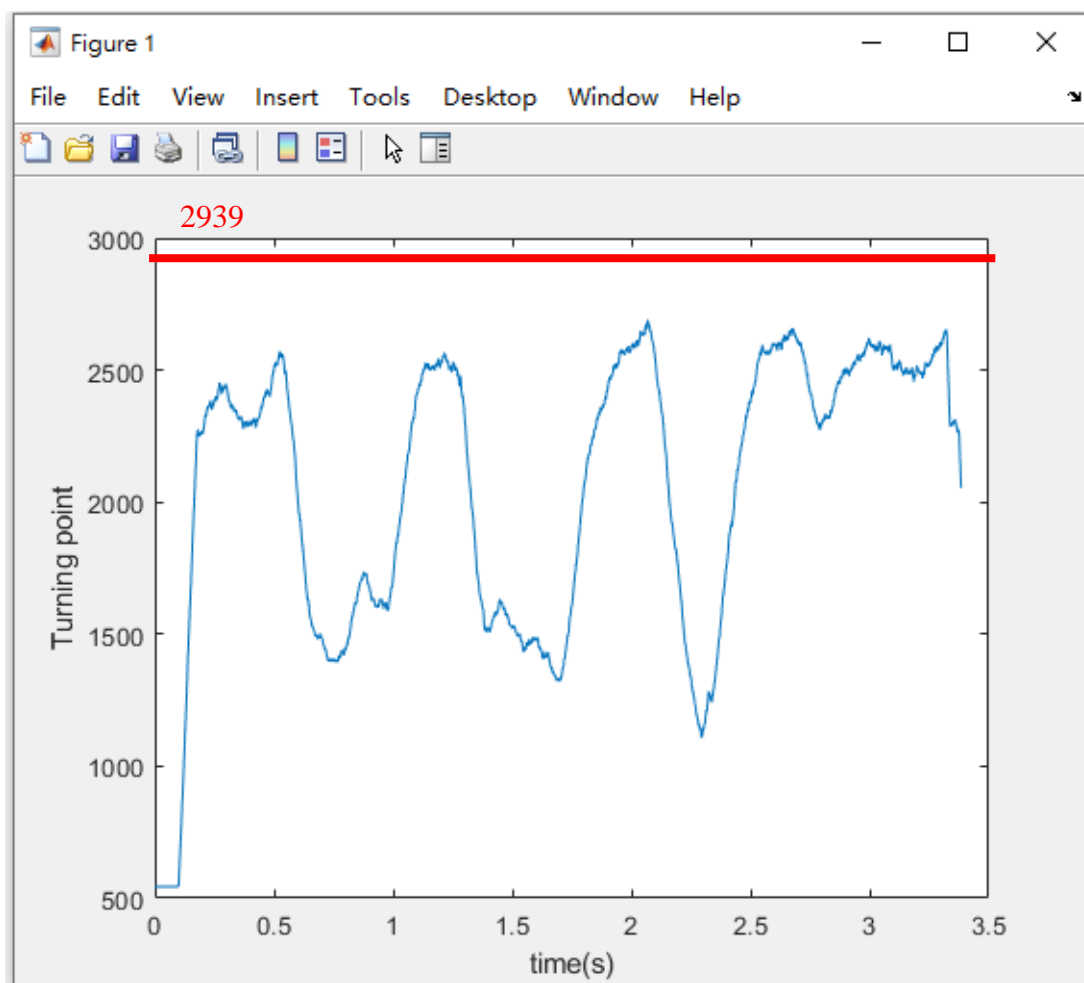
Window = 0.001 秒

N = 441

隨機訊號的界線 = 293

取樣頻率 = 44100 Hz

當 window 的長度增加時，蔡和佑的訊號變成



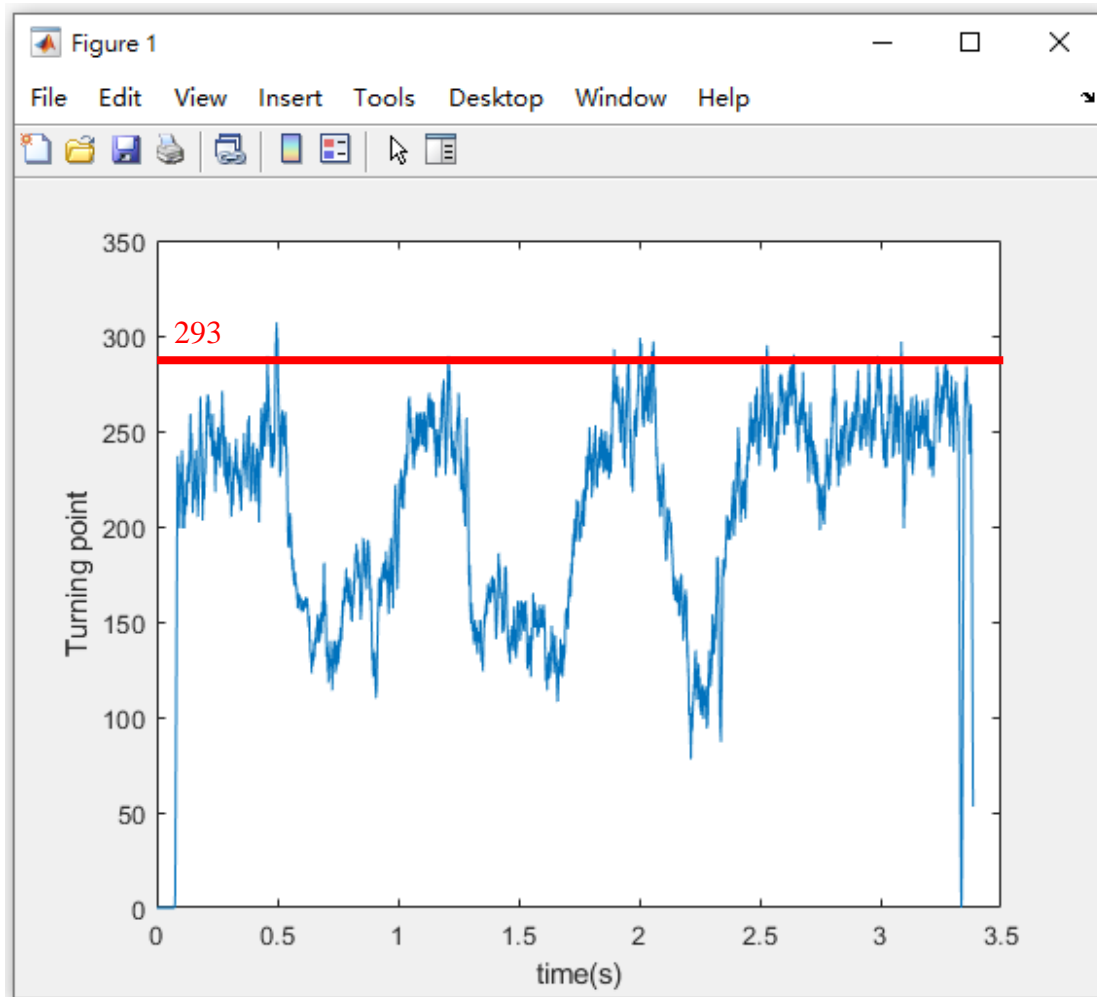
Window = 0.1 秒

N = 4410

隨機訊號的界線 = 2939

取樣頻率 = 44100 Hz

當 window 的長度縮短時，蔡和佑的訊號變成



Window = 0.001 秒

N = 441

隨機訊號的界線 = 293

取樣頻率 = 44100 Hz

從上述兩個例子，可以看出改變 window 的長度，會改變訊號的隨機性。當 window 長度增加時，原本為隨機訊號的地方不見了；反之當 window 長度縮短時，原本不為隨機訊號的地方，開始有了隨機的趨勢。

六、參考文獻

R. M. Rangayyan, Biomedical Signal Analysis, 2nd Ed., Wiley, 2015.

