FCU

ePaper

逢甲大學學生報告 ePaper

# 凹陷濾波器與梳形濾波器的設計與驗證

# Design and verification of notch filter and comb filter


作者：吳宜峰

系級：資訊四丙

學號：D0683410


開課老師：林育德

課程名稱：生醫信號處理

開課系所：自動控制工程學系生醫學程

開課學年：109 學年度 第二學期

# 摘要

本次實驗的目的是要設計出可降低某一特定頻率干擾的濾波器，也就是凹陷濾波器，但通常干擾會形成諧波，所以干擾在基頻的奇數倍數的頻率也會有能量，且會持續遞減，此時就要設計可同時降低某一特定頻率與它的奇數倍數頻率的濾波器，也就是梳形濾波器。

簡單設計出來的濾波器除了會降低目標頻率的能量，也會影響到附近頻率的能量，可能會將重要的信號也一併濾掉，所以還會利用 Shank's recursive 的觀念去改良，將濾波器的影響範圍縮小，就會變成 IIR notch/comb filter。

最後再去比對凹陷濾波器與梳形濾波器改良前後的濾波效果。

本次實驗是根據課本第 147~150 頁與第 163 頁的觀念來設計四個濾波器，先計算要過濾的頻率所對應的角度，再假設轉移函數 H(z)並將算式推導成只含有實數的部分，再將 H(z)正規化，最後將函式 Z 轉換就可得到濾波器的使用參數。

本次實驗是使用 Python 撰寫，將計算出的參數代入 signal.freqz()就可得到濾波器的幅度響應與相位響應，將計算出的參數與未濾波的信號代入 signal.lfilter()就可得到濾波後的信號。

觀察結果可發現只過濾目標頻率的效果會比有同時過濾奇數倍頻率的效果還要差，而有調整成 IIR filter 的濾波也會比較好。


**關鍵字：**Z 轉換、凹陷濾波器、信號處理、梳形濾波器、無限脈衝響應濾波器

# Abstract

The purpose of this experiment is to design a filter that can reduce the interference of a certain frequency, that is, the notch filter, but usually the interference will form harmonics, so the interference will also have energy at odd multiples of the fundamental frequency, and It will continue to decrease. At this time, it is necessary to design a filter that can simultaneously reduce a certain frequency and its odd multiples, that is, a comb filter.

The simply designed filter will not only reduce the energy of the target frequency, but also affect the energy of nearby frequencies. It may also filter out important signals. Therefore, the concept of Shank's recursive will be used to improve the filter. When the scope of influence is reduced, it will become an IIR notch/comb filter.

Finally, compare the filtering effects of the notch filter and the comb filter before and after the improvement.

In this experiment, four filters are designed according to the concepts on pages 147 to 150 and 163 of the textbook. First, calculate the angle corresponding to the frequency to be filtered, and then assume the transfer function H(z) and derive the formula as only For the part containing real numbers, normalize H(z), and finally perform Z-transformation of the function to obtain the filter parameters.

This experiment is written in Python. Substitute the calculated parameters into signal.freqz() to get the magnitude response and phase response of the filter, and substitute the calculated parameters and unfiltered signal into signal.lfilter() to get The filtered signal.

After observing the results, it can be found that the effect of filtering only the target frequency will be worse than the effect of filtering odd multiples at the same time, and the filtering adjusted to the IIR filter will be better.


**Keyword**：comb filter, infinite impulse response(IIR) filter, notch filter, signal processing, Z-transformation

# 目　次

# 一、說　明

1. 本次實驗的詳細內容：
   一. 根據講義第 147~150 頁與第 163 頁的觀念來設計四個濾波器。
      (1) 過濾 頻率 為 60-Hz 的 notch filter
      (2) 過濾 頻率 為 60-Hz 的 Shank's recursive (IIR) notch filter
      (3) 過濾 基率 為 60-Hz 的 comb filter
      (4) 過濾 基頻 為 60-Hz 的 Shank's recursive (IIR) comb filter
   二. 利用前面設計的四個濾波器對受 60-Hz 電源線干擾的心電信號進行降噪，並呈現濾波前後的結果。

2. 設計濾波器的步驟：
   一. 畫一個單位圓，並在(1, 0)與(-1, 0)的地方分別標記 0 與 $\frac{fs}{2}$。
   二. 計算過濾的頻率的正負值在單位圓上的位置與角度。
   三. 將 zeros 放在要過濾的頻率的位置上，若要設計 comb filter 則還需要將 poles 放在與 zeors 相同角度且在單位圓內的位置上。
   四. 假設 H(z)，並將算式推導成只含實數部分(將 $e^{j\theta}$ 與 $e^{-j\theta}$ 消掉)。
      假設要過濾的頻率為 $hz_1, hz_2, hz_3, \ldots$。
      計算它們在單位圓上的角度為 $\theta_1, \theta_2, \theta_3, \ldots$。
      則

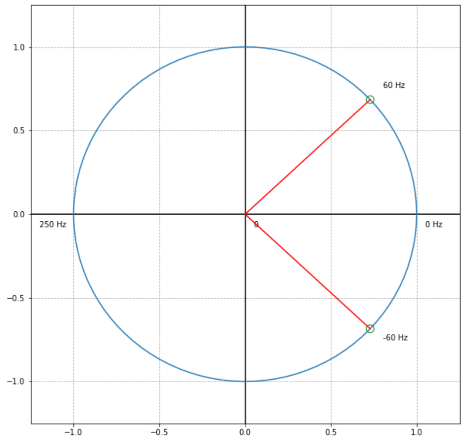
$$\mathrm{H(z)} = \frac{\left(1 - e^{j\theta1} \cdot z^{-1}\right)\left(1 - e^{-j\theta1} \cdot z^{-1}\right)\left(1 - e^{j\theta2} \cdot z^{-1}\right)\left(1 - e^{-j\theta2} \cdot z^{-1}\right)\ldots}{\left(1 - r \cdot e^{j\theta1} \cdot z^{-1}\right)\left(1 - r \cdot e^{-j\theta1} \cdot z^{-1}\right)\left(1 - r \cdot e^{j\theta2} \cdot z^{-1}\right)\left(1 - r \cdot e^{-j\theta2} \cdot z^{-1}\right)\ldots}$$

$$= \frac{Y(z)}{X(z)}$$

   五. 對 H(z)作正規化，即將 H(z) / H(1)，並將 X(z)與 Y(z)分別放在等號左右。
   六. 將結果做 Z 轉換。

## 二、濾波器設計流程

(一)過濾 頻率 為 60-Hz 的 notch filter




Fs = 500

$$\theta = \frac{60 \cdot 180}{250} = 43.2$$

$$H(z) = \left(1 - e^{j\theta} \cdot z^{-1}\right)\left(1 - e^{-j\theta} \cdot z^{-1}\right)$$

$$= 1 - \left(e^{j\theta} + e^{-j\theta}\right) \cdot z^{-1} + z^{-2}$$

$$= 1 - 2 \cdot \cos\theta \cdot z^{-1} + z^{-2}$$

**• θ = 43.2 代入:**

$$H(z) = 1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}$$

**• z 代 1:**

$$H(1) = 1 - 2 \cdot \cos(43.2) \cdot 1 + 1$$

$$= 2(1 - \cos(43.2))$$

**•正規化 :**

$$H(z) = \frac{1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}}{H(1)} = \frac{Y(z)}{X(z)}$$

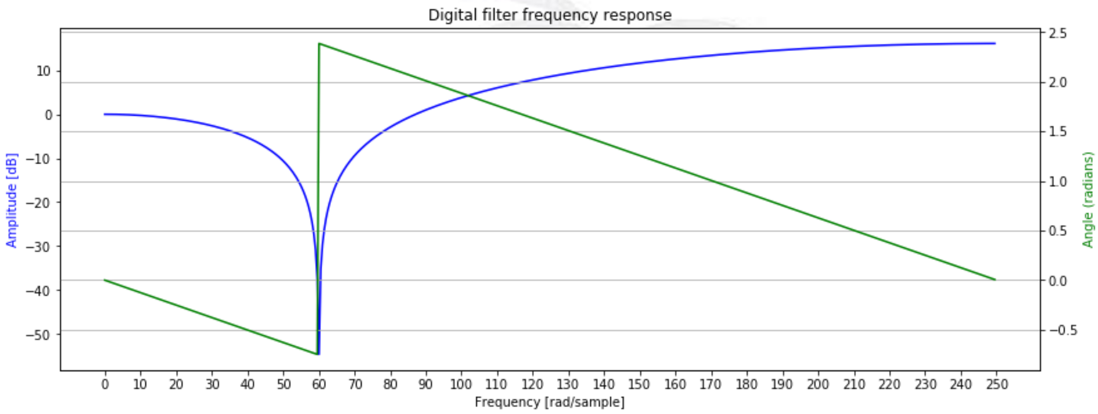
$$=> H(1) \cdot Y(z) = (1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}) \cdot X(z)$$

$$=> H(1) \cdot Y(z) = X(z) - 2 \cdot \cos(43.2) \cdot z^{-1} \cdot X(z) + z^{-2} \cdot X(z)$$

**• Z 轉換 :**

$$H(1) \cdot y[n] = x[n] - 2 \cdot \cos(43.2) \cdot x[n-1] + x[n-2]$$

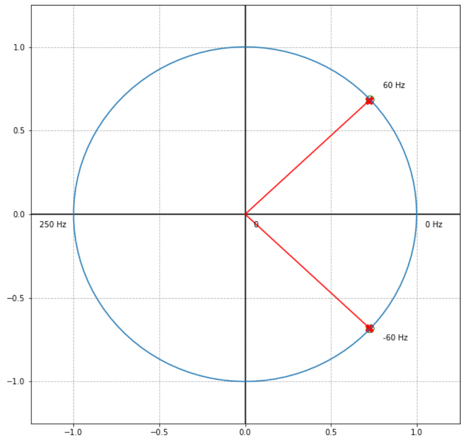
**• magnitude response 以及 phase response :**



(二)過濾 頻率 為 60-Hz 的 Shank's recursive (IIR) notch filter

Fs = 500

$$\theta = \frac{60 \cdot 180}{250} = 43.2$$




$$H(z) = \frac{\left(1 - e^{j\theta} \cdot z^{-1}\right)\left(1 - e^{-j\theta} \cdot z^{-1}\right)}{\left(1 - r \cdot e^{j\theta} \cdot z^{-1}\right)\left(1 - r \cdot e^{-j\theta} \cdot z^{-1}\right)}$$

$$= \frac{1 - \left(e^{j\theta} + e^{-j\theta}\right) \cdot z^{-1} + z^{-2}}{1 - r \cdot \left(e^{j\theta} + e^{-j\theta}\right) \cdot z^{-1} + r^2 \cdot z^{-2}}$$

$$= \frac{1 - 2 \cdot \cos\theta \cdot z^{-1} + z^{-2}}{1 - 2 \cdot r \cdot \cos\theta \cdot z^{-1} + r^2 \cdot z^{-2}}$$

• **θ = 43.2 代入:**

$$H(z) = \frac{1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}}{1 - 2 \cdot r \cdot \cos(43.2) \cdot z^{-1} + r^2 \cdot z^{-2}}$$

• **z 代 1:**

$$H(1) = \frac{1 - 2 \cdot \cos(43.2) \cdot 1 + 1}{1 - 2 \cdot r \cdot \cos(43.2) \cdot 1 + r^2 \cdot 1}$$

$$= \frac{2(1 - \cos(43.2))}{1 - 2 \cdot r \cdot \cos(43.2) + r^2}$$

•**正規化 :**

$$H(z) = \frac{1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}}{(1 - 2r \cdot \cos(43.2) \cdot z^{-1} + r^2 \cdot z^{-2}) \cdot H(1)} = \frac{Y(z)}{X(z)}$$

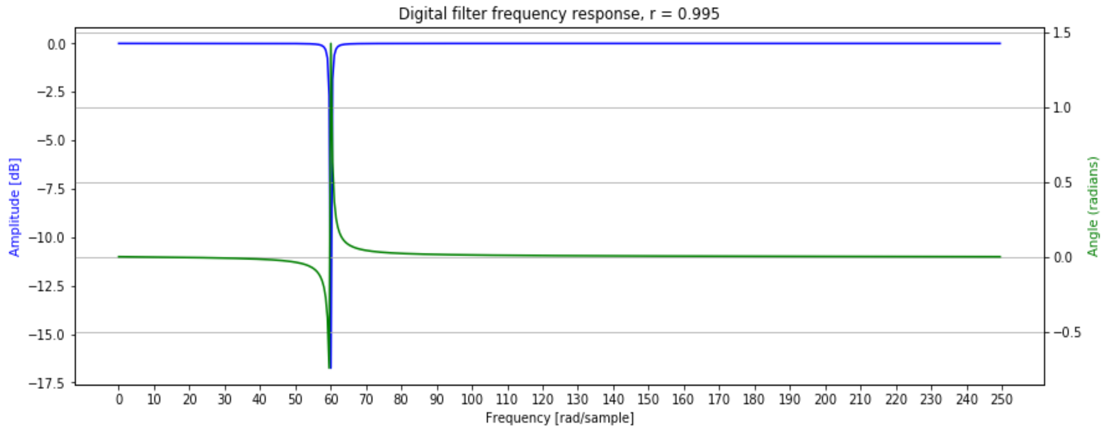
$$=> H(1) \cdot (1 - 2r \cdot \cos(43.2) \cdot z^{-1} + r^2 \cdot z^{-2}) \cdot Y(z)$$
$$= (1 - 2 \cdot \cos(43.2) \cdot z^{-1} + z^{-2}) \cdot X(z)$$

$$=> H(1) \cdot (Y(z) - 2r \cdot \cos(43.2) \cdot Y(z) \cdot z^{-1} + r^2 \cdot Y(z) \cdot z^{-2})$$
$$= (X(z) - 2 \cdot \cos(43.2) \cdot X(z) \cdot z^{-1} + X(z) \cdot z^{-2})$$

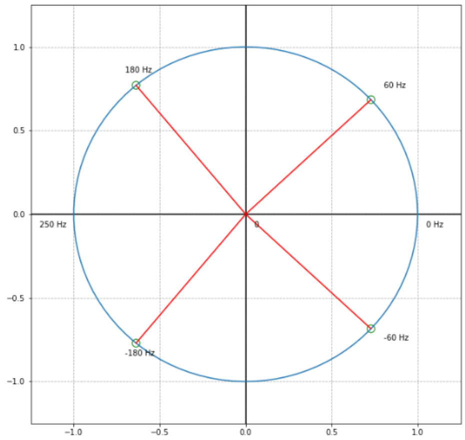
• **Z 轉換 :**

$$H(1) \cdot (y[n] - 2r \cdot \cos(43.2) \cdot y[n-1] + r^2 \cdot y[n-2])$$
$$= (x[n] - 2 \cdot \cos(43.2) \cdot x[n-1] + x[n-2])$$

• **magnitude response 以及 phase response :**




(三) 過濾 基頻 為 60-Hz 的 comb filter




Fs = 500

$$\theta a = \frac{60 \cdot 180}{250} = 43.2$$

$$\theta b = \frac{180 \cdot 180}{250} = 129.6$$

$$H(z) = \left(1 - e^{j\theta a} \cdot z^{-1}\right)\left(1 - e^{-j\theta a} \cdot z^{-1}\right) \cdot \left(1 - e^{j\theta b} \cdot z^{-1}\right)\left(1 - e^{-j\theta b} \cdot z^{-1}\right)$$

$$= \left(1 - \left(e^{j\theta a} + e^{-j\theta a}\right) \cdot z^{-1} + z^{-2}\right)\left(1 - \left(e^{j\theta b} + e^{-j\theta b}\right) \cdot z^{-1} + z^{-2}\right)$$

$$= (1 - 2 \cdot \cos\theta a \cdot z^{-1} + z^{-2})(1 - 2 \cdot \cos\theta b \cdot z^{-1} + z^{-2})$$

| A = -2 · cosθa , B = -2 · cosθb 取代 |
|---|

$$= (1 + A \cdot z^{-1} + z^{-2})(1 + B \cdot z^{-1} + z^{-2})$$

$$= 1 + B \cdot z^{-1} + z^{-2} + A \cdot z^{-1} + AB \cdot z^{-2} + A \cdot z^{-3} + z^{-2} + B \cdot z^{-3} + z^{-4}$$

$$= 1 + (A + B) \cdot z^{-1} + (2 + AB) \cdot z^{-2} + (A + B) \cdot z^{-3} + z^{-4}$$

| A = -2 · cosθa , B = -2 · cosθb 帶回 |
|---|

$$= 1 - 2(\cos\theta a + \cos\theta b) \cdot z^{-1} + 2(1 + 2 \cdot \cos\theta a \cdot \cos\theta b) \cdot z^{-2} - 2 \cdot (\cos\theta a + \cos\theta b) \cdot z^{-3} + z^{-4}$$

- **θa = 43.2, θb = 129.6 代入:**

$$H(z) = 1 - 2(\cos(43.2) + \cos(129.6)) \cdot z^{-1} + 2(1 + 2 \cdot \cos(43.2) \cdot \cos(129.6)) \cdot z^{-2} - 2(\cos(43.2) + \cos(129.6)) \cdot z^{-3} + z^{-4}$$

- **z 代 1:**

$$H(1) = 1 - 2(\cos(43.2) + \cos(129.6)) \cdot 1 + 2(1 + 2 \cdot \cos(43.2) \cdot \cos(129.6)) \cdot 1 - 2(\cos(43.2) + \cos(129.6)) \cdot 1 + 1$$

$$= 2 - 2 \cdot \cos(43.2) - 2 \cdot \cos(129.6) + 2 + 4 \cdot \cos(43.2) \cdot \cos(129.6) - 2 \cdot \cos(43.2) - 2 \cdot \cos(129.6)$$

$$= 4 - 4\cdot\cos(43.2) - 4\cdot\cos(129.6) + 4\cdot\cos(43.2)\cdot\cos(129.6)$$
$$= 4(1 - \cos(43.2) - \cos(129.6) + \cos(43.2)\cdot\cos(129.6))$$

**•正規化 ：**

$$H(z) = \frac{\begin{array}{c}1 - 2(\cos(43.2) + \cos(129.6))\cdot z^{-1} + 2(1 + 2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} \\ -2(\cos(43.2) + \cos(129.6))\cdot z^{-3} + z^{-4}\end{array}}{H(1)} = \frac{Y(z)}{X(z)}$$

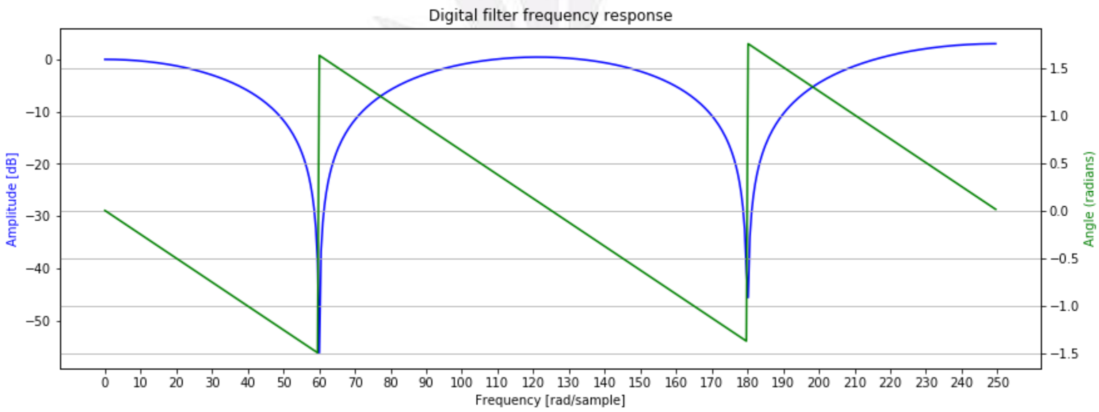
$$\Rightarrow H(1)\cdot\ Y(z)$$
$$= (1 - 2(\cos(43.2) + \cos(129.6))\cdot z^{-1}$$
$$+ 2(1 + 2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2}$$
$$- 2(\cos(43.2) + \cos(129.6))\cdot z^{-3} + z^{-4})\cdot X(z)$$

$$\Rightarrow H(1)\cdot\ Y(z)$$
$$= X(z) - 2(\cos(43.2) + \cos(129.6))\cdot X(z)\cdot z^{-1}$$
$$+ 2(1 + 2\cdot\cos(43.2)\cdot\cos(129.6))\cdot X(z)\cdot z^{-2}$$
$$- 2(\cos(43.2) + \cos(129.6))\cdot X(z)\cdot z^{-3} + X(z)\cdot z^{-4}$$

**• Z 轉換 ：**

$$H(1)\cdot y[n] = x[n] - 2(\cos(43.2) + \cos(129.6))\cdot x[n-1]$$
$$+ 2(1 + 2\cdot\cos(43.2)\cdot\cos(129.6))\cdot x[n-2]$$
$$- 2(\cos(43.2) + \cos(129.6))\cdot x[n-3] + x[n-4]$$

**• magnitude response 以及 phase response :**

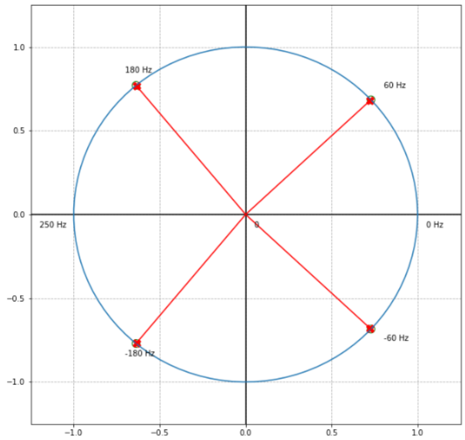


（四）過濾 基頻 為 60-Hz 的 Shank's recursive (IIR) comb filter

Fs = 500

$$\theta a = \frac{60 \cdot 180}{250} = 43.2$$

$$\theta b = \frac{180 \cdot 180}{250} = 129.6$$




H(z)

$$= \frac{(1-e^{j\theta a}\cdot z^{-1})(1-e^{-j\theta a}\cdot z^{-1})(1-e^{j\theta b}\cdot z^{-1})(1-e^{-j\theta b}\cdot z^{-1})}{(1-r\cdot e^{j\theta a}\cdot z^{-1})(1-r\cdot e^{-j\theta a}\cdot z^{-1})(1-r\cdot e^{j\theta b}\cdot z^{-1})(1-r\cdot e^{-j\theta b}\cdot z^{-1})}$$

$$= \frac{(1-(e^{j\theta a}+e^{-j\theta a})\cdot z^{-1}+z^{-2})(1-(e^{j\theta b}+e^{-j\theta b})\cdot z^{-1}+z^{-2})}{(1-r\cdot(e^{j\theta a}+e^{-j\theta a})\cdot z^{-1}+r^2\cdot z^{-2})(1-r\cdot(e^{j\theta b}+e^{-j\theta b})\cdot z^{-1}+r^2\cdot z^{-2})}$$

$$= \frac{(1-2\cdot\cos\theta a\cdot z^{-1}+z^{-2})(1-2\cdot\cos\theta b\cdot z^{-1}+z^{-2})}{(1-2\cdot r\cdot\cos\theta a\cdot z^{-1}+r^2\cdot z^{-2})(1-2\cdot r\cdot\cos\theta b\cdot z^{-1}+r^2\cdot z^{-2})}$$

A = -2·cosθa , B = -2·cosθb , C = -2·r·cosθa , D = -2·r·cosθb 取代

$$= \frac{(1+A\cdot z^{-1}+z^{-2})(1+B\cdot z^{-1}+z^{-2})}{(1+C\cdot z^{-1}+r^2\cdot z^{-2})(1+D\cdot z^{-1}+r^2\cdot z^{-2})}$$

$$= \frac{1+B\cdot z^{-1}+z^{-2}+A\cdot z^{-1}+AB\cdot z^{-2}+A\cdot z^{-3}+z^{-2}+B\cdot z^{-3}+z^{-4}}{1+D\cdot z^{-1}+r^2\cdot z^{-2}+C\cdot z^{-1}+CD\cdot z^{-2}+r^2\cdot C\cdot z^{-3}+r^2\cdot z^{-2}+r^2\cdot D\cdot z^{-3}+r^4\cdot z^{-4}}$$

$$= \frac{1+(A+B)\cdot z^{-1}+(2+AB)\cdot z^{-2}+(A+B)\cdot z^{-3}+z^{-4}}{1+(C+D)\cdot z^{-1}+(2r^2+CD)\cdot z^{-2}+r^2(C+D)\cdot z^{-3}+r^4\cdot z^{-4}}$$

A = -2·cosθa , B = -2·cosθb , C = -2·r·cosθa , D = -2·r·cosθb 代回

$$= \frac{1-2(\cos\theta a+\cos\theta b)\cdot z^{-1}+2(1+2\cdot\cos\theta a\cdot\cos\theta b)\cdot z^{-2} -2(\cos\theta a+\cos\theta b)\cdot z^{-3}+z^{-4}}{1-2r(\cos\theta a+\cos\theta b)\cdot z^{-1}+2r^2(1+2\cdot\cos\theta a\cdot\cos\theta b)\cdot z^{-2} -2r^3(\cos\theta a+\cos\theta b)\cdot z^{-3}+r^4\cdot z^{-4}}$$

**• θa = 43.2, θb = 129.6 代入:**

$$H(z) = \frac{1-2(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} -2(\cos(43.2)+\cos(129.6))\cdot z^{-3}+z^{-4}}{1-2r(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} -2r^3(\cos(43.2)+\cos(129.6))\cdot z^{-3}+r^4\cdot z^{-4}}$$

**• z 代 1:**

$$H(1) = \frac{1-2(\cos(43.2)+\cos(129.6))\cdot 1+2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot 1 -2(\cos(43.2)+\cos(129.6))\cdot 1+1}{1-2r(\cos(43.2)+\cos(129.6))\cdot 1+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot 1 -2r^3(\cos(43.2)+\cos(129.6))\cdot 1+r^4\cdot 1}$$

$$= \frac{1-2(\cos(43.2)+\cos(129.6))+2(1+2\cdot\cos(43.2)\cdot\cos(129.6)) -2(\cos(43.2)+\cos(129.6))+1}{1-2r(\cos(43.2)+\cos(129.6))+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6)) -2r^3(\cos(43.2)+\cos(129.6))+r^4}$$

$$= \frac{4-4\cdot\cos(43.2)-4\cdot\cos(129.6)+4\cdot\cos(43.2)\cdot\cos(129.6)}{1+2r^2+r^4-2r(1+r^2)(\cos(43.2)+\cos(129.6))+4r^2\cdot\cos(43.2)\cdot\cos(129.6)}$$

$$= \frac{4(1-\cos(43.2)-\cos(129.6)+\cos(43.2)\cdot\cos(129.6))}{1+2r^2+r^4-2r(1+r^2)(\cos(43.2)+\cos(129.6))+4r^2\cdot\cos(43.2)\cdot\cos(129.6)}$$

**•正規化 ：**

$$H(z) = \frac{1-2(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} -2(\cos(43.2)+\cos(129.6))\cdot z^{-3}+z^{-4}}{1-2r(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} -2r^3(\cos(43.2)+\cos(129.6))\cdot z^{-3}+r^4\cdot z^{-4}} \cdot \frac{1}{H(1)} = \frac{Y(z)}{X(z)}$$

$$\Rightarrow H(1)\cdot\big(1-2r(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} - 2r^3(\cos(43.2)+\cos(129.6))\cdot z^{-3}+r^4\cdot z^{-4}\big)\cdot Y(z)$$
$$= \big(1-2(\cos(43.2)+\cos(129.6))\cdot z^{-1}+2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot z^{-2} - 2(\cos(43.2)+\cos(129.6))\cdot z^{-3}+z^{-4}\big)\cdot X(z)$$

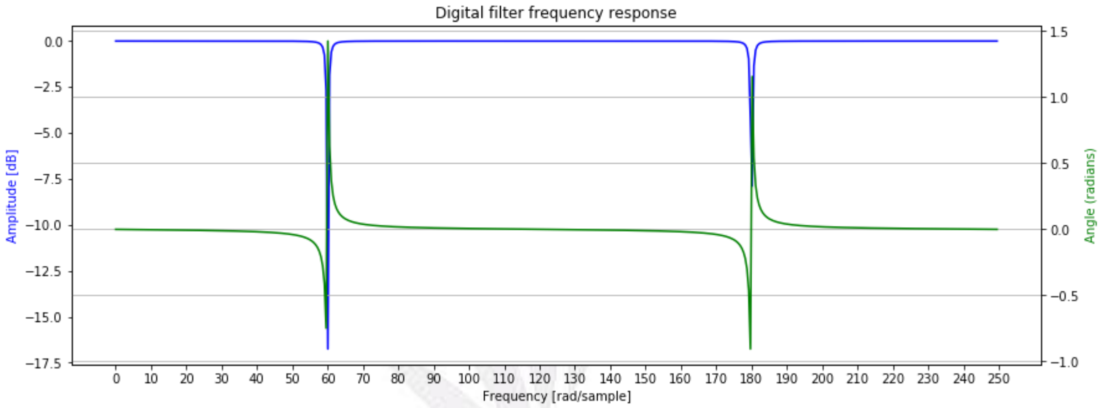
$$\Rightarrow H(1)\cdot\big(Y(z)-2r(\cos(43.2)+\cos(129.6))\cdot Y(z)\cdot z^{-1}+2r^2(1+2\cdot\cos(43.2)\cdot\cos(129.6)) \cdot Y(z)\cdot z^{-2}-2r^3(\cos(43.2)+\cos(129.6))\cdot Y(z)\cdot z^{-3}+r^4\cdot Y(z)\cdot z^{-4}\big)$$
$$= \big(X(z)-2(\cos(43.2)+\cos(129.6))\cdot X(z)\cdot z^{-1} + 2(1+2\cdot\cos(43.2)\cdot\cos(129.6))\cdot X(z)\cdot z^{-2}-2(\cos(43.2)+\cos(129.6)) \cdot X(z)\cdot z^{-3}+X(z)\cdot z^{-4}\big)$$

**• Z 轉換 ：**

$$H(1)\cdot\begin{pmatrix} y[n]-2r\big(\cos(43.2)+\cos(129.6)\big)\cdot y[n-1]+2r^2\big(1+2\cdot\cos(43.2)\cdot\cos(129.6)\big)\cdot y[n-2] \\ -2r^3\big(\cos(43.2)+\cos(129.6)\big)\cdot y[n-3]+r^4\cdot y[n-4] \end{pmatrix}$$

$$=x[n]-2\big(\cos(43.2)+\cos(129.6)\big)\cdot x[n-1]+2\big(1+2\cdot\cos(43.2)\cdot\cos(129.6)\big)$$
$$\cdot x[n-2]-2\big(\cos(43.2)+\cos(129.6)\big)\cdot x[n-3]+x[n-4]$$

**• magnitude response 以及 phase response :**



## 三、程　式　碼

```
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
import os
from scipy import signal

def cos(theta):
    import numpy as np
    return np.cos(theta / 180 * np.pi)

path = ["C:/Users/USER/Desktop/../60_Hz_干擾心電圖/"]

sample_rate = 500

for i in range(len(path)):
    for filename in os.listdir(path[i]):
        if filename[0] in ["A", "D"]:
            print(filename)

            # 讀取 ECG 資料，並擷取前 10 秒
            data = get_ECG_data(path[i], filename)
            data = data[:10 * sample_rate]

            # 輸出原圖
```

```
        plot_fig(data, sample_rate, 10, save_fig = True, fig_save_path = path[i] + "10 秒切割
原圖/",
                 filename = filename, additional_name = "_原圖")

        # ( 一 ) 60-Hz notch filter
        # 設定想要過濾的頻率
        zeros_hz = [60]

        # 換算出頻率的角度，並將單位圓與 zeros 與 poles 畫出
        theta = plot_circle_and_return_theta(sample_rate, zeros_hz, fig_size= 10, plot_fig =
False)[0]

        # 定義 H(z)
        def H(z):
            return (1 - 2 * cos(theta) * z + (z ** 2))

        # 設定正規化後 difference equation X 與 Y 的係數
        x_coeff = np.array([1,
                            -2 * cos(theta),
                            1])
        y_coeff = np.array([1])
        y_coeff = y_coeff * H(1)

        # 進行降噪
        filted_sigs = signal.lfilter(x_coeff, y_coeff, data)

        # 輸出降噪結果圖
        plot_fig(filted_sigs, sample_rate, 10,save_fig = True, fig_save_path = path[i] + "10
秒切割降噪結果圖/",
                 filename = filename, additional_name = "_60Hz_notch")

        # ( 二 ) 60-Hz IIR notch filter
        # 設定想要過濾的頻率與 r 值
        r = 0.995
        zeros_hz = [60]
        poles_hz = [60]

        # 換算出頻率的角度，並將單位圓與 zeros 與 poles 畫出
```

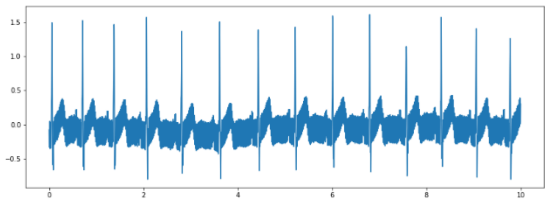
```
        theta = plot_circle_and_return_theta(sample_rate, zeros_hz = zeros_hz, fig_size= 10,
poles_hz = poles_hz,
                                                r = r, plot_fig = False)[0]

        # 定義 H(z)
        def H(z):
            return (1 - 2 * cos(theta) * z + (z ** 2)) / (1 - 2 * r * cos(theta) * z + (r ** 2) * (z
** 2))

        # 設定正規化後 difference equation X 與 Y 的係數
        x_coeff = np.array([1,
                            -2 * cos(theta),
                            1])
        y_coeff = np.array([1,
                            -2 * r * cos(theta),
                            r ** 2])
        y_coeff = y_coeff * H(1)

        # 進行降噪
        filted_sigs = signal.lfilter(x_coeff, y_coeff, data)

        # 輸出降噪結果圖
        plot_fig(filted_sigs, sample_rate, 10,save_fig = True, fig_save_path = path[i] + "10
秒切割降噪結果圖/",
                 filename = filename, additional_name = "_60Hz_IIR_notch")

        # ( 三 ) 60, 180-Hz comb filter
        # 設定想要過濾的頻率
        zeros_hz = [60, 180]

        # 換算出頻率的角度，並將單位圓與 zeros 與 poles 畫出
        theta_1, theta_2 = plot_circle_and_return_theta(sample_rate, zeros_hz = zeros_hz,
fig_size= 10,
                                                        plot_fig = False)

        # 定義 H(z)
        def H(z):
            return (1 - 2 * (cos(theta_1) + cos(theta_2)) * z + 2 * (1 + 2 * cos(theta_1) *
```

```
cos(theta_2)) * (z ** 2) \
                        -2 * (cos(theta_1) + cos(theta_2)) * (z ** 3) + (z ** 4))

        # 設定正規化後 difference equation X 與 Y 的係數
        x_coeff = np.array([1,
                            -2 * (cos(theta_1) + cos(theta_2)),
                             2 * (1 + 2 * cos(theta_1) * cos(theta_2)),
                            -2 * (cos(theta_1) + cos(theta_2)),
                             1])
        y_coeff = np.array([1])
        y_coeff = y_coeff * H(1)

        # 進行降噪
        filted_sigs = signal.lfilter(x_coeff, y_coeff, data)

        # 輸出降噪結果圖
        plot_fig(filted_sigs, sample_rate, 10,save_fig = True, fig_save_path = path[i] + "10
秒切割降噪結果圖/",
                    filename = filename, additional_name = "_60,180Hz_notch")

        # ( 四 ) 60, 180-Hz IIR comb filter
        # 設定想要過濾的頻率與 r 值
        r = 0.995
        zeros_hz = [60, 180]
        poles_hz = [60, 180]

        # 換算出頻率的角度，並將單位圓與 zeros 與 poles 畫出
        theta_1, theta_2 = plot_circle_and_return_theta(sample_rate, zeros_hz = zeros_hz,
fig_size= 10,
                                                            poles_hz = poles_hz, r =
r, plot_fig = False)

        # 定義 H(z)
        def H(z):
            return (1 - 2 * (cos(theta_1) + cos(theta_2)) * z + 2 * (1 + 2 * cos(theta_1) *
cos(theta_2)) * (z ** 2) - \
            2 * (cos(theta_1) + cos(theta_2)) * (z ** 3) + (z ** 4)) / \
            (1 - 2 * r * (cos(theta_1) + cos(theta_2)) * z + 2 * (r ** 2) * (1 + 2 *
```

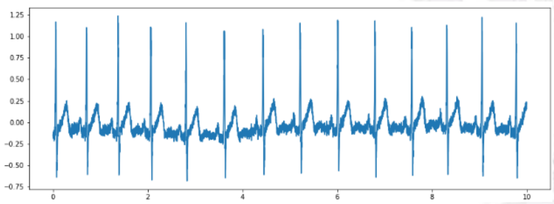
```
cos(theta_1) * cos(theta_2)) * (z ** 2) - \
                2 * (r ** 3) * (cos(theta_1) + cos(theta_2)) * (z ** 3) + (r ** 4) * (z ** 4))

            # 設定正規化後 difference equation X 與 Y 的係數
            x_coeff = np.array([1,
                                -2 * (cos(theta_1) + cos(theta_2)),
                                 2 * (1 + 2 * cos(theta_1) * cos(theta_2)),
                                -2 * (cos(theta_1) + cos(theta_2)),
                                 1])

            y_coeff = np.array([1,
                                -2 * r * (cos(theta_1) + cos(theta_2)),
                                 2 * (r ** 2) * (1 + 2 * cos(theta_1) * cos(theta_2)),
                                -2 * (r ** 3) * (cos(theta_1) + cos(theta_2)),
                                 (r ** 4)])
            y_coeff = y_coeff * H(1)

            # 進行降噪
            filted_sigs = signal.lfilter(x_coeff, y_coeff, data)

            # 輸出降噪結果圖
            plot_fig(filted_sigs, sample_rate, 10,save_fig = True, fig_save_path = path[i] + "10
秒切割降噪結果圖/",
                     filename = filename, additional_name = "_60,180Hz_IIR_notch")
```

# 四、實 驗 結 果

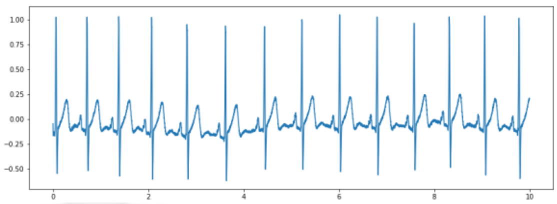
A1a_原圖 ▼



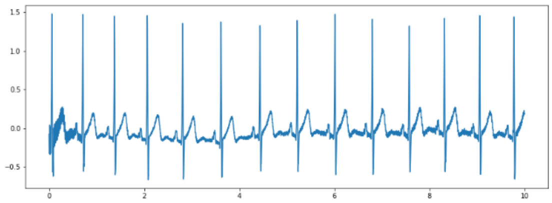
A1a_60Hz_notch_filter ▼



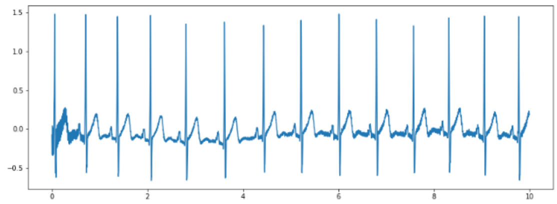
A1a_60, 180Hz_ comb _filter ▼



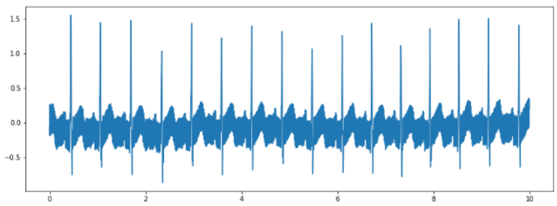
A1a_60Hz_IIR_notch_filter ▼



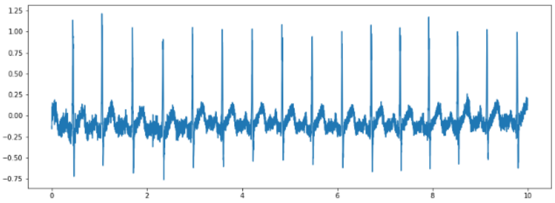
A1a_60, 180Hz_IIR_ comb _filter ▼



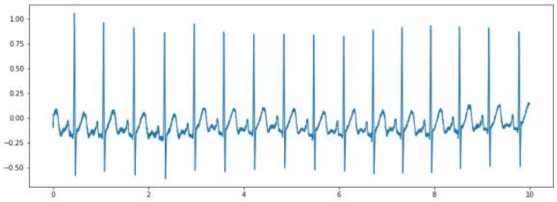
A2a_原圖 ▼



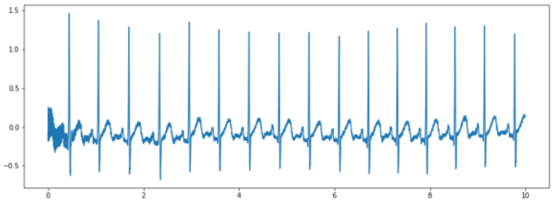
A2a_60Hz_notch_filter ▼



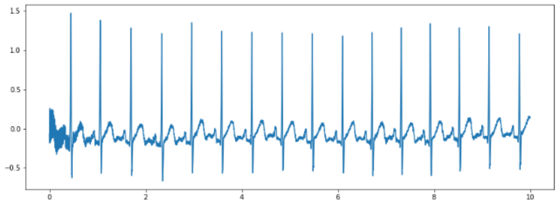
A2a_60, 180Hz_ comb _filter ▼



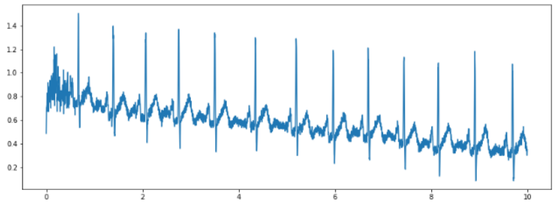
A2a_60Hz_IIR_notch_filter ▼



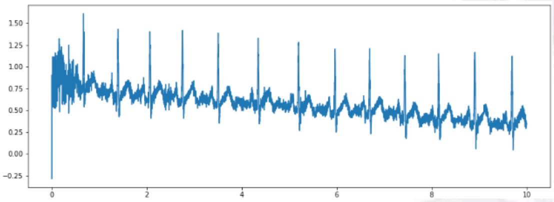
A2a_60, 180Hz_IIR_ comb _filter ▼



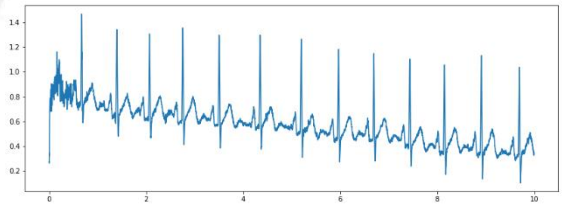
D1a_原圖 ▼



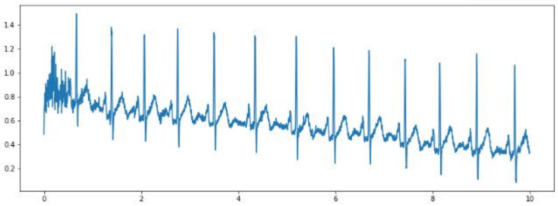
D1a_60Hz_notch_filter ▼



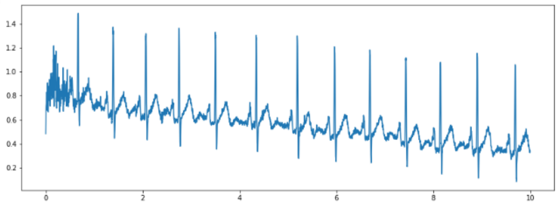
D1a_60, 180Hz_ comb _filter ▼



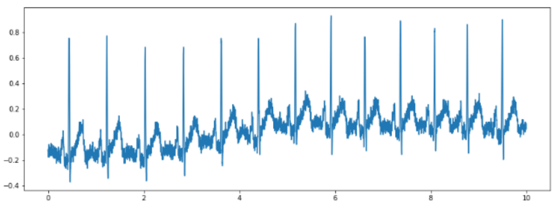
D1a_60Hz_IIR_notch_filter ▼



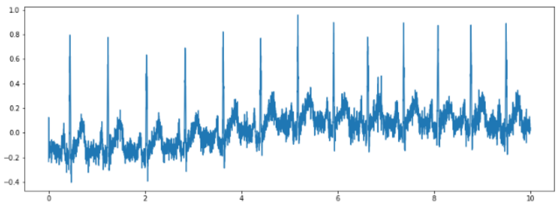
D1a_60, 180Hz_IIR_ comb _filter ▼



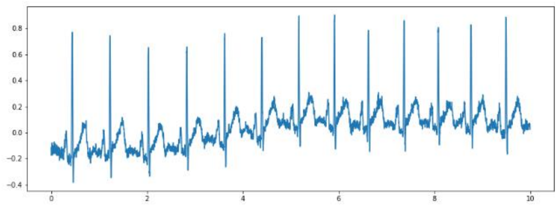
D2a_原圖 ▼



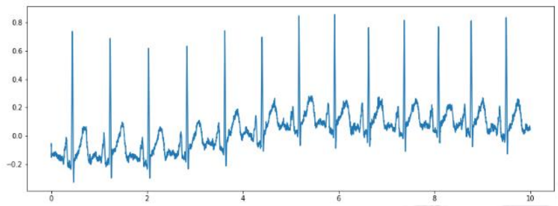
D2a_60Hz_notch_filter ▼



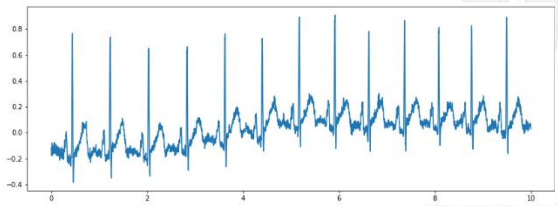
D2a_60Hz_IIR_notch_filter ▼



D2a_60, 180Hz_comb_filter ▼



D2a_60, 180Hz_IIR_ comb _filter ▼



# 五、參 考 文 獻

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