◎ 2-J Relations among Filter Length N, Transition Band, and Accuracy

♦ Suppose that we want:

① passband ripple $\leq \delta_1$,

② stopband ripple $\leq \delta_2$,

③ width of transition band $\leq \Delta F$ (expressed by normalized frequency)

$$\Delta F = (f_1 - f_2)/f_s = (f_1 - f_2)T \quad (f_s\text{: sampling frequency, } T\text{: sampling interval})$$

Then, the estimated length N of the digital filter is:

$$N = \frac{2}{3}\frac{1}{\Delta F}\log_{10}\left(\frac{1}{10\delta_1\delta_2}\right)$$

.

- When there are two transition bands, $\Delta F = \min(\Delta F_1, \Delta F_2)$
- 犧牲 transition band 的 frequency response, 換取較高的 passband and stopband accuracies

$$N = \frac{2}{3}\frac{1}{\Delta F}\log_{10}\left(\frac{1}{10\delta_1\delta_2}\right) \qquad \frac{3}{2}N\Delta F = \log_{10}\left(\frac{1}{10\delta_1\delta_2}\right)$$

$$-\frac{3}{2}N\Delta F = \log_{10}(10\delta_1\delta_2)$$

$$10\delta_1\delta_2 = 10^{-\frac{3}{2}N\Delta F}$$

$$\delta_1\delta_2 = 10^{-3N\Delta F/2-1}$$

if $\delta_1 = \delta_2 = \delta$, $\delta^2 = 10^{-3N\Delta F/2-1}$

[Ref] F. Mintzer and L. Bede, "Practical design rules for optimum FIR bandpass digital filter", *IEEE Trans. ASSP*, vol. 27, no. 2, pp. 204-206, Apr. 1979.

問題：假設 $\sqrt{10}\delta_1 = \sqrt{10}\delta_2 = \delta$，$N$ 為固定，

當 ΔF 變為 A 倍時，δ 變為多少？

If δ is the original error

δ_0 is the new error

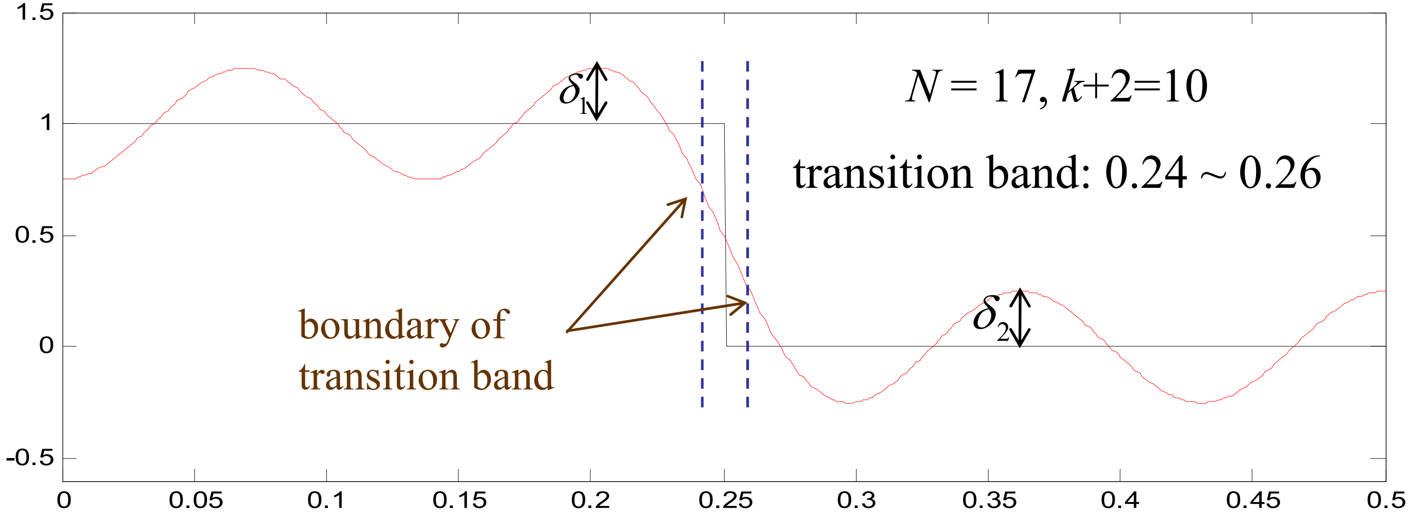
$$10\delta^2 = 10^{-\frac{3}{2}N\Delta F}$$

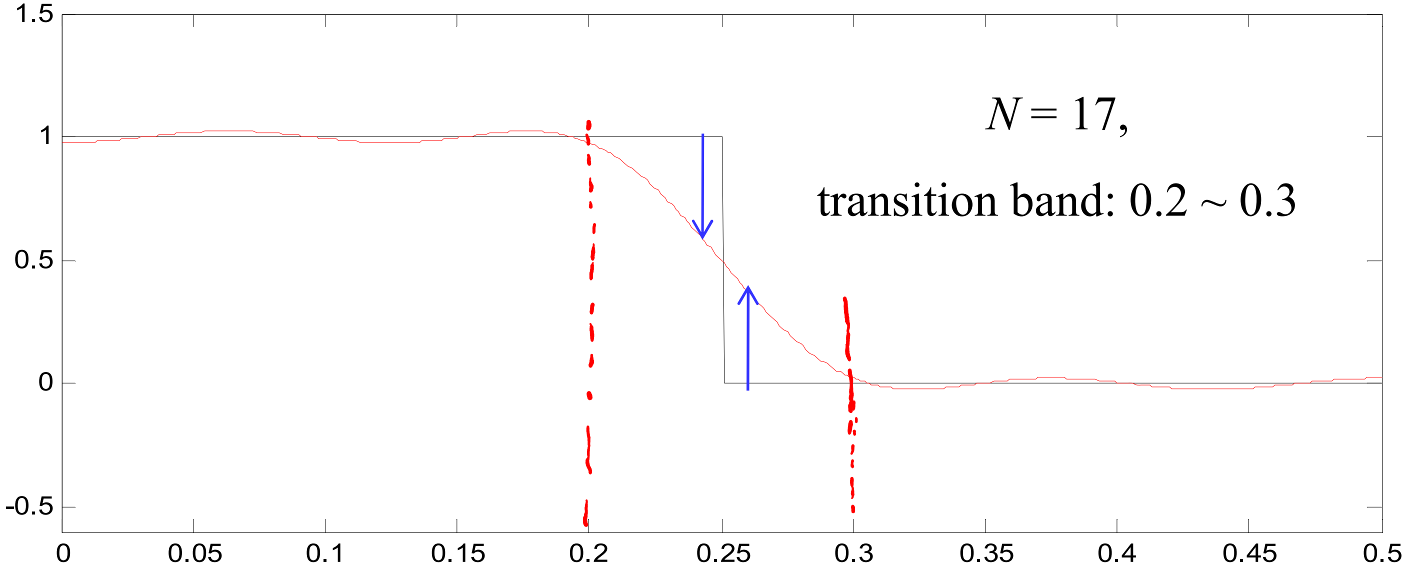
$$10\delta_0^2 = 10^{(-\frac{3}{2}N\Delta F)A} = (10\delta^2)^A = 10^A\delta^{2A}$$

$$\delta_0 = 10^{\frac{A-1}{2}}\delta^A$$

If $\delta = 0.1$, $A = 5$

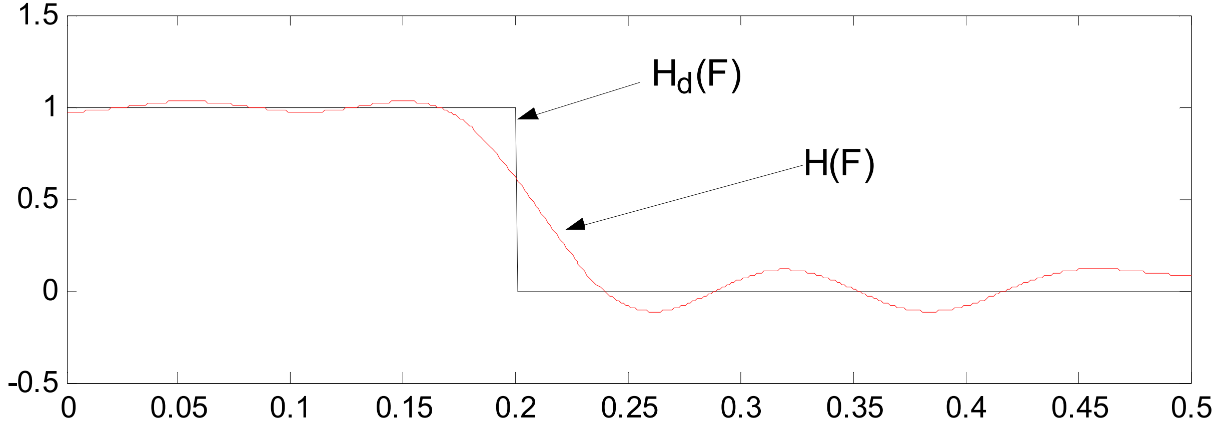
$$\delta_0 = 100(0.1)^5 = 10^{-3}$$


$N = 17$, $k+2=10$
transition band: 0.24 ~ 0.26
δ_1
δ_2
boundary of
transition band
1.5
1
0.5
0
-0.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

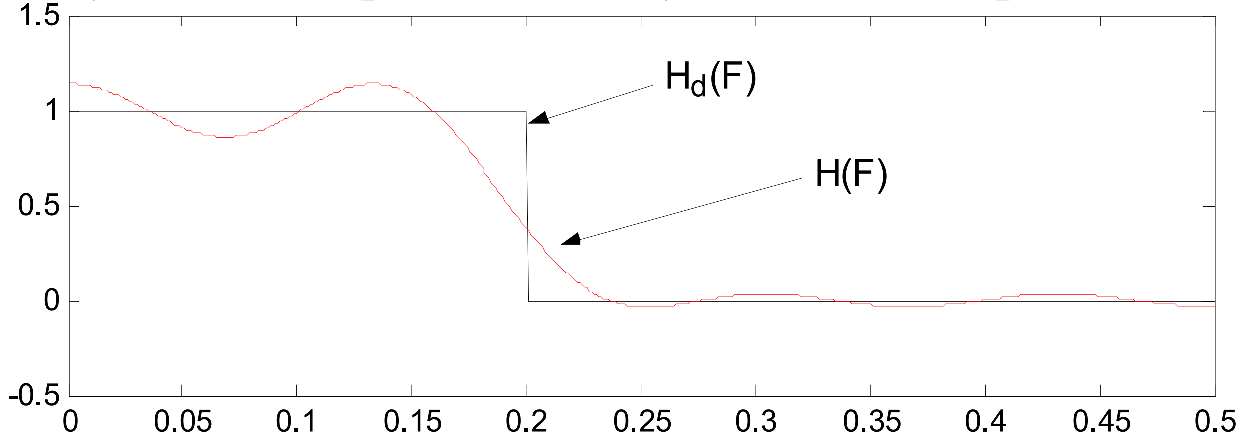

$N = 17$,
transition band: 0.2 ~ 0.3
1.5
1
0.5
0
-0.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

◉ 2-K Relations between Weight Functions and Accuracy

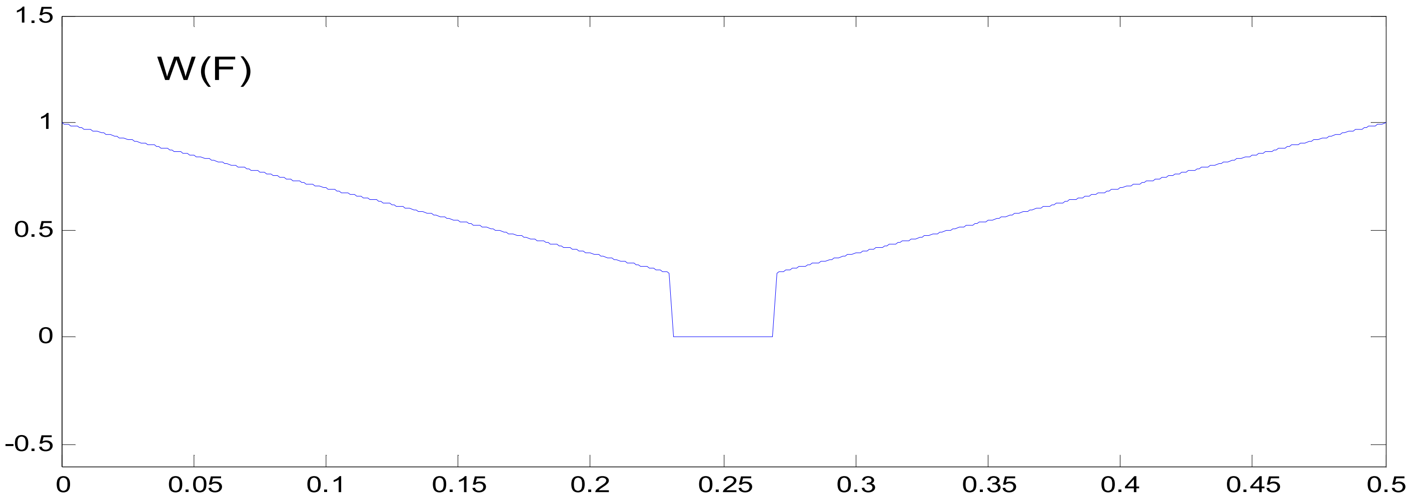
If we treat the passband more important than the stop band
$W(f)$ = 1 in the passband, $0 < W(f) < 1$ in the stopband

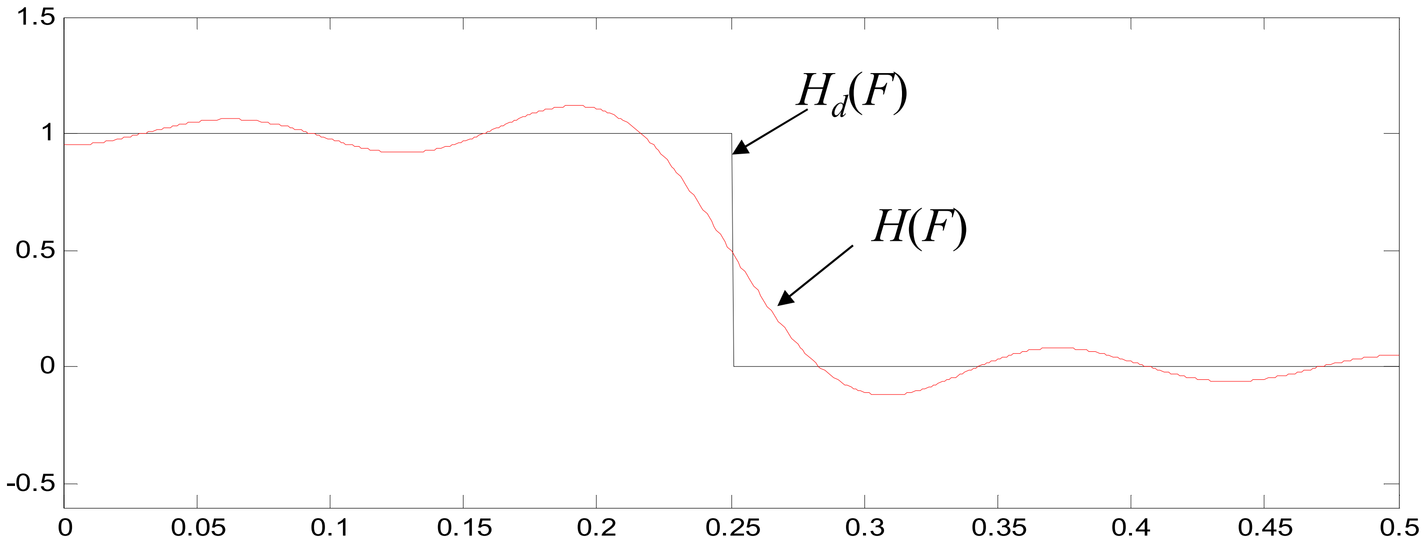


If we treat the stop band more important than the pass band
$0 < W(f) < 1$ in the passband, $W(f)$ = 1 in the stopband



Larger error near the transition band





◉ 2-L FIR Filter in MSE Sense with Weight Functions

$$R(F)=\sum_{n=0}^{k} s[n]\cos(2\pi nF)$$

可對照 pages 49~51

$$MSE=\int_{-1/2}^{1/2} W(F)\left|R(F)-H_d(F)\right|^2 dF$$

$F=f/f_s$

$$=\int_{-1/2}^{1/2} W(F)\left(\sum_{\tau=0}^{k} s[\tau]\cos(2\pi\tau F)-H_d(F)\right)^2 dF$$

$$\frac{\partial MSE}{\partial s[n]}=2\int_{-1/2}^{1/2} W(F)\cos(2\pi nF)\left(\sum_{\tau=0}^{k} s[\tau]\cos(2\pi\tau F)-H_d(F)\right)dF=0$$

$n=0,1,\dots k$

Compared to page 49

$$2\sum_{\tau=0}^{k} s[\tau]\int_{-1/2}^{1/2} W(F)\cos(2\pi nF)\cos(2\pi\tau F)dF-2\int_{-1/2}^{1/2} W(F)H_d(F)\cos(2\pi nF)dF=0$$

$$n=0\sim k$$

問題： $\int_{-1/2}^{1/2} W(F)\cos(2\pi nF)\cos(2\pi\tau F)dF\neq 0$ when $n\neq\tau$

(not orthogonal)

$$\sum_{\tau=0}^{k} s[\tau] \underline{\int_{-1/2}^{1/2} W(F)\cos(2\pi n F)\cos(2\pi \tau F)\,dF} = \int_{-1/2}^{1/2} W(F) H_d(F)\cos(2\pi n F)\,dF$$

$$\underline{\tau = 0 \sim k,} \qquad \underline{n = 0 \sim k}$$

可以表示成 $(k+1) \times (k+1)$ matrix operation

$$\begin{array}{c c}
 & \begin{array}{ccccc} \boldsymbol{\tau = 0} & \boldsymbol{\tau = 1} & \boldsymbol{\tau = 2} & & \boldsymbol{\tau = k} \end{array} \\
\begin{array}{c} \boldsymbol{n = 0} \\ \boldsymbol{n = 1} \\ \boldsymbol{n = 2} \\ \\ \boldsymbol{n = k} \end{array} &
\begin{bmatrix}
B[0,0] & B[0,1] & B[0,2] & \cdots & B[0,k] \\
B[1,0] & B[1,1] & B[1,2] & \cdots & B[1,k] \\
B[2,0] & B[2,1] & B[2,2] & \cdots & B[2,k] \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
B[k,0] & B[k,1] & B[k,2] & \cdots & B[k,k]
\end{bmatrix}
\begin{bmatrix} s[0] \\ s[1] \\ s[2] \\ \vdots \\ s[k] \end{bmatrix}
=
\begin{bmatrix} C[0] \\ C[1] \\ C[2] \\ \vdots \\ C[k] \end{bmatrix}
\end{array}$$

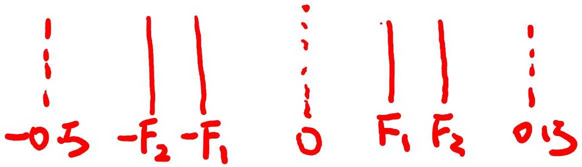
$$\mathbf{B} \qquad \mathbf{S} = \mathbf{C} \qquad \therefore\ \mathbf{S} = \mathbf{B}^{-1}\mathbf{C}$$

$$B[n,\tau] = \int_{-1/2}^{1/2} W(F)\cos(2\pi n F)\cos(2\pi \tau F)\,dF$$

$$C[n] = \int_{-1/2}^{1/2} W(F) H_d(F)\cos(2\pi n F)\,dF$$

When $W(F)=1$

$$B[n,\tau] : \begin{bmatrix} 1 & & & 0 \\ & 1/2 & & \\ & & 1/2 & \\ 0 & & & \ddots \\ & & & & 1/2 \end{bmatrix}$$



Q： Is it possible to apply the transition band to the FIR filter in the MSE sense?

$$MSE = ?\ \int_{-0.5}^{-F_2} W(F)\left|R(F)-H_d(F)\right|^2 dF + \int_{-F_1}^{F_1} W(F)\left|R(F)-H_d(F)\right|^2 dF + \int_{F_2}^{0.5} W(F)\left|R(F)-H_d(F)\right|^2 dF$$

for $B[n,\tau] = ?$ $\quad \int_{-\frac{1}{2}}^{\frac{1}{2}} \cdots dF \Rightarrow \int_{-0.5}^{-F_2} \cdots dF + \int_{-F_1}^{F_1} \cdots dF + \int_{F_2}^{0.5} \cdots dF$

$C[n]$

◉ 2-M Four Types of FIR Filter

$h[n] = 0$ for $n < 0$ and $n \geq N$ 點數為 N

$$H(F) = \sum_{n=0}^{N-1} h[n]\exp(-j2\pi nF)$$

- Type 1 $R(F) = \sum_{n=0}^{k} s[n]\cos(2\pi nF)$ ← 之前的方法只討論到 Type 1

$h[n_1] = h[n_2 - n]$ and N is odd.

(even symmetric)

$k = (N-1)/2$

- Type 1: $$R(F)=\sum_{n=0}^{k} s[n]\cos(2\pi n F)$$

  <u>$h[n]=h[N-1-n]$</u> (even symmetric) and <u>N is odd.</u>

- Type 2: $$R(F)=\sum_{n=1}^{k+1/2} s[n]\cos(2\pi (n-1/2)F)$$

  <u>$h[n]=h[N-1-n]$</u> (even symmetric) and <u>N is even.</u>

- Type 3: $$R(F)=\sum_{n=1}^{k} s[n]\sin(2\pi n F)$$

  <u>$h[n]=-h[N-1-n]$</u> (odd symmetric) and <u>N is odd.</u>

- Type 4: $$R(F)=\sum_{n=1}^{k+1/2} s[n]\sin(2\pi (n-1/2)F)$$

  <u>$h[n]=-h[N-1-n]$</u> (odd symmetric) and <u>N is even.</u>

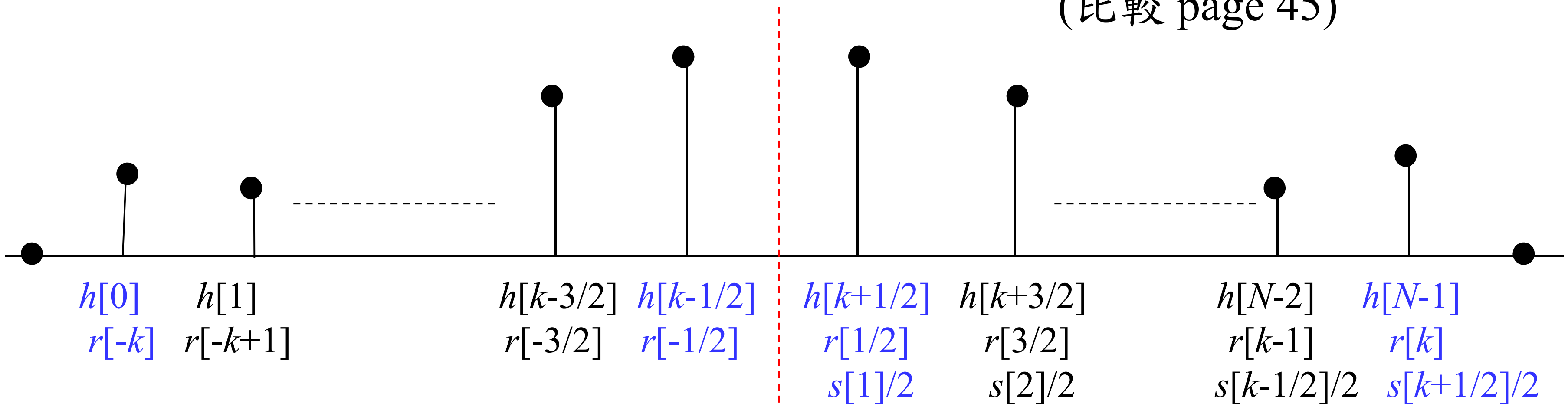
$k=(N-1)/2$

for MSE
substitute R(F) on page 49
by the corresponding functions

- Type 2: When $h[n] = h[N-1-n]$ and N is even:
  (even symmetric)

令 $r[n] = h[n + k]$, where $k = (N-1)/2$ (注意此時 k 不為整數)

(比較 page 45)



k $s[n]/2 = r[n-\frac{1}{2}]$

$= \sum_{n=-k}^{-1/2} r[n] e^{-j2\pi nF} + \sum_{n=1/2}^{k} r[n] e^{-j2\pi nF}$

當 $R(F) = \sum_{n=-k}^{k} r[n]\exp(-j2\pi nF)$

$R(F) = e^{j2\pi Fk} H(F)$

$\downarrow$

$2\sum_{n=1/2}^{k} r[n]\cos(2\pi nF)$

$$R(F) = \sum_{n=1/2}^{k} \{r[n]\exp(-j2\pi nF) + r[-n]\exp(j2\pi nF)\}$$

$$= \sum_{n=1/2}^{k} r[n]\{\exp(-j2\pi nF) + \exp(j2\pi nF)\} = \sum_{n=1/2}^{k} 2r[n]\cos(2\pi nF)$$

$$\boxed{R(F) = \sum_{n=1}^{k+1/2} s[n]\cos(2\pi(n-1/2)F)}$$

$$n_{(new)} = n_{(old)} + \frac{1}{2} \qquad n_{(old)} = n_{(new)} - \frac{1}{2}$$

$s[n] = 2r[n-1/2]$ $\quad n = 1, 2, \ldots, k+1/2$

設計出 $s[n]$ 之後

$r[n] = s[n+1/2]/2, \qquad h[n] = r[n-k],$

Design Method for Type 2

(for minimax)

$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$
$\cos(\alpha-\beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$
$\frac{1}{2}(\cos(\alpha+\beta)+\cos(\alpha-\beta)) = \cos\alpha\cos\beta$

$$R(F) = \sum_{n=1}^{k+1/2} s[n]\cos(2\pi(n-1/2)F)$$

由於 n 和 $n+1$ 兩項相加可得

$$\cos(2\pi(n-1/2)F) + \cos(2\pi(n+1/2)F) = \underline{2\cos(\pi F)}\cos(2\pi nF)$$

所以可以「判斷」$R(F)$ 能被改寫成

$$\boxed{R(F) = \underline{\cos(\pi F)}\sum_{n=0}^{k_1} s_1[n]\cos(2\pi nF)}$$

求 $s_1[n]$ 和 $s[n]$ 之間的關係

$$R(F) = \sum_{n=0}^{k_1} s_1[n]\cos(\pi F)\cos(2\pi nF)$$

$$= \sum_{n=0}^{k_1}\frac{1}{2}s_1[n]\cos(2\pi(n-1/2)F) + \sum_{n=0}^{k_1}\frac{1}{2}s_1[n]\cos(2\pi(n+1/2)F)$$

$$= \sum_{n=0}^{k_1}\frac{1}{2}s_1[n]\cos(2\pi(n-1/2)F) + \sum_{n=1}^{k_1+1}\frac{1}{2}s_1[n-1]\cos(2\pi(n-1/2)F)$$

$n_{new} = n_{old}+1$
$n_{old} = n_{new}-1$

$$R(F)=\sum_{n=0}^{k_1}\frac{1}{2}s_1[n]\cos(2\pi(n-1/2)F)+\sum_{n=1}^{k_1+1}\frac{1}{2}s_1[n-1]\cos(2\pi(n-1/2)F)$$

$n = 0$ $n = 1\sim k_1$ $n = 1\sim k_1$ $n = k_1+1$

$$R(F)=\frac{1}{2}s_1[0]\cos(\pi F)+\sum_{n=1}^{k_1}\frac{1}{2}(s_1[n]+s_1[n-1])\cos(2\pi(n-1/2)F)$$

$$+\frac{1}{2}s_1[k_1]\cos(2\pi(k_1+1/2)F)$$

$n = 1$ $n = 2\sim k_1$

$$R(F)=\left(s_1[0]+\frac{1}{2}s_1[1]\right)\cos(\pi F)+\sum_{n=2}^{k-1/2}\frac{1}{2}(s_1[n]+s_1[n-1])\cos(2\pi(n-1/2)F)$$

$$+\frac{1}{2}s_1[k-1/2]\cos(2\pi(k)F)$$

(令 $k_1 + 1/2 = k$)

比較係數可得

$$s[1]=s_1[0]+\frac{1}{2}s_1[1]$$

$$s[n]=\frac{1}{2}(s_1[n]+s_1[n-1])$$ for $n = 2, 3, \ldots, k-1/2$

$$s[k+1/2]=\frac{1}{2}s_1[k-1/2]$$

$$err(F) = [R(F) - H_d(F)]W(F)$$
$$= \left[\cos(\pi F)\sum_{n=0}^{k-1/2} s_1[n]\cos(2\pi nF) - H_d(F)\right]W(F)$$
$$= \left[\sum_{n=0}^{k-1/2} s_1[n]\cos(2\pi nF) - \sec(\pi F)H_d(F)\right]\cos(\pi F)W(F)$$

只需將 pages 58-61 的方法當中，$H_d(F)$ 換成 $\sec(\pi F)H_d(F)$

$$\left[\sum_{n=0}^{k} s[n]\cos(2\pi nF) - H_d(F)\right]W(F)$$

$W(F)$ 換成 $\cos(\pi F)W(F)$

k 換成 $k - 1/2 = N/2 - 1$

注意 $s_1[n]$ 和 $s[n]$ 之間的關係即可

Design Method for Type 3

$$R(F)=\sum_{n=1}^{k} s[n]\sin(2\pi nF)$$

由於 $n-1$ 和 $n+1$ 兩項相減可得

$$\sin(2\pi(n+1)F)-\sin(2\pi(n-1)F)=2\sin(2\pi F)\cos(2\pi nF)$$

所以「判斷」可將 $R(F)$ 改寫為

$$\boxed{R(F)=\sin(2\pi F)\sum_{n=0}^{k_1} s_1[n]\cos(2\pi nF)}$$

求 $s_1[n]$ 和 $s[n]$ 之間的關係

$$R(F)=\frac{1}{2}\sum_{n=0}^{k_1} s_1[n]\sin(2\pi(n+1)F)-\frac{1}{2}\sum_{n=0}^{k_1} s_1[n]\sin(2\pi(n-1)F)$$

$$=\frac{1}{2}\sum_{n=0}^{k_1} s_1[n]\sin(2\pi(n+1)F)+\frac{s_1[0]}{2}\sin(2\pi F)-\frac{1}{2}\sum_{n=2}^{k_1} s_1[n]\sin(2\pi(n-1)F)$$

$$=\frac{1}{2}\sum_{n=1}^{k_1+1} s_1[n-1]\sin(2\pi nF)+\frac{s_1[0]}{2}\sin(2\pi F)-\frac{1}{2}\sum_{n=1}^{k_1-1} s_1[n+1]\sin(2\pi nF)$$

$\sin(\alpha+\beta)=\sin\alpha\cos\beta+\sin\beta\cos\alpha$

$\sin(\alpha-\beta)=\sin\alpha\cos\beta-\sin\beta\cos\alpha$

$\alpha=2\pi nF$

$\beta=2\pi F$

$$R(F) = \frac{s_1[0]}{2}\sin(2\pi F) + \frac{1}{2}(s_1[0] - s_1[2])\sin(2\pi F)$$
$$+\frac{1}{2}\sum_{n=2}^{k_1-1}(s_1[n-1] - s_1[n+1])\sin(2\pi n F)$$
$$+\frac{1}{2}s_1[k_1-1]\sin(2\pi k_1 F) + \frac{1}{2}s_1[k_1]\sin(2\pi(k_1+1)F)$$

令 $k_1 = k - 1$, 比較係數可得

$$s[1] = s_1[0] - \frac{1}{2}s_1[2]$$

$$s[n] = \frac{1}{2}s_1[n-1] - \frac{1}{2}s_1[n+1] \qquad \text{for } n = 2,\ 3,\ \ldots,\ k-2$$

$$s[k-1] = \frac{1}{2}s_1[k-2]$$

$$s[k] = \frac{1}{2}s_1[k-1]$$

$$err(F) = [R(F) - H_d(F)]W(F)$$
$$= \left[\sin(2\pi F)\sum_{n=0}^{k-1} s_1[n]\cos(2\pi nF) - H_d(F)\right]W(F)$$
$$= \left[\sum_{n=0}^{k-1} s_1[n]\cos(2\pi nF) - \csc(2\pi F)H_d(F)\right]\sin(2\pi F)W(F)$$

將 pages 58-61 的方法當中，$H_d(F)$ 換成 $\csc(2\pi F)H_d(F)$

$W(F)$ 換成 $\sin(2\pi F)W(F)$

k 換成 $k-1$

注意$s_1[n]$ 和 $s[n]$ 之間的關係即可

(Think)：Design the Method for Type 4

附錄三：寫 Matlab / Python 程式需注意的地方

一、各種程式語言寫程式共通的原則

(1) 能夠不在迴圈內做的運算，則移到迴圈外，以節省運算時間

(2) 寫一部分即測試，不要全部寫完再測試 (縮小範圍比較容易 debug)

(3) 先測試簡單的例子，成功後再測試複雜的例子

二、Matlab 寫程式特有的技巧

(1) 迴圈能避免就儘量避免

(2) 儘可能使用 Matrix 及 Vector operation

Example: 由 1 加 到100，用 Matlab 一行就可以了

```
sum([1:100])
```

完全不需迴圈

三、一些重要的 Matlab 指令

(1) function: 放在第一行，可以將整個程式函式化

(2) tic, toc: 計算時間

tic 為開始計時，toc 為顯示時間

(3) find: 找尋一個 vector 當中不等於 0 的entry 的位置

範例：find([1 0 0 1]) = [1, 4]

find(abs([-5:5])<=2) = [4, 5, 6, 7, 8]

(因為 abs([-5:5])<=2 = [0 0 0 1 1 1 1 1 0 0 0])

(4) ′ : Hermitian (transpose + conjugation)， . ′ : transpose

(5) imread: 讀圖， image, imshow, imagesc: 將圖顯示出來，

(註： 較老的 Matlab 版本 imread 要和 double 並用

A=double(imread(‘Lena.bmp’));

(6) imwrite: 製做圖檔

(7) xlsread: 由 Excel 檔讀取資料

A = xlsread(‘檔名’, ‘工作表名’, 範圍);

例如

A = xlsread(‘test. xlsx’, ‘工作表1’, A1:D50);

(8) xlswrite: 將資料寫成 Excel 檔

(9) aviread: 讀取 video 檔

(10) dlmread: 讀取 *.txt 或其他類型檔案的資料

(11) dlmwrite: 將資料寫成 *.txt 或其他類型檔案

四、寫 Python 版本程式可能會用到的重要指令

建議必安裝模組

```
pip install numpy
pip install scipy
pip install opencv-python
pip install openpyxl   # for Excel files
```

(1) 定義函式：使用 def

(2) 計算時間

```
import time
start_time = time.time()   #獲取當前時間
end_time = time.time()
total_time = end_time - start_time   #計算時間差來得到總執行時間
```

感謝2021年擔任助教的蔡昌廷同學

(3) 讀取圖檔、輸出圖檔(建議使用opencv)

```
import cv2
image = cv2.imread(file_name)  #預設color channel為BGR
cv2.imwrite(file_name, image)   #需將color channel轉為BGR
```

(4) 尋找array中滿足特定條件的值的位置

(相當於 Matlab 的 find 指令)

```
import numpy as np
a = np.array([0, 1, 2, 3, 4, 5])
index = np.where(a > 3) # 回傳array([4, 5])
print(index)
    (array([4, 5], dtype=int64),)
index[0][0]
    4
index[0][1]
   5
```

$$\mathbf{A_1} = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 4 & 5 \end{bmatrix}$$

```
A1= np.array([[1,3,6],[2,4,5]])
index = np.where(A1 > 3)
print(index)
    (array([0, 1, 1], dtype=int64), array([2, 1, 2], dtype=int64))
```

(代表滿足 A1 > 3 的點的位置座標為 [0, 2], [1, 1], [1, 2]

```
[index[0][0], index[1][0]]
    [0, 2]
[index[0][1], index[1][1]]
    [1, 1]
[index[0][2], index[1][2]]
    [1, 2]
```

(5) Hermitian、transpose

```
import numpy as np
result = np.conj(matrix.T)     # Hermitian
result = matrix.T    # transpose
```

(6) 在 Python 當中讀取 Matlab 當中的 mat 檔

```
data = scipy.io.loadmat('***.mat')
y = np.array(data['y'])  # 假設 y 是 ***.mat 當中儲存的資料
```

(7) 在 Python 當中讀取 Excel 檔

```
import openpyxl
data = openpyxl.load_workbook(‘filename’)
data1 = data[‘工作表名’]
A  = [row for row in data1.values]
A1 = np.array(A)
A1 = np.double(A1)  # 資料數值化
```

◉ 2-M Frequency Sampling Method

假設 designed filter $h[n]$ 的區間為 $n \in [0 , N-1]$

filter 的點數為 N, $k = (N - 1)/2$

remember:

$H_d(f) = H_d(f+f_s)$

- **Frequency Sampling 基本精神：**

若 $H_d(f)$ 是 desired filter 的 discrete-time Fourier transform

$R(f)$ 是 $r[n] = h[n+P]$ 的 discrete-time Fourier transform

要求 $\boxed{R\left(\frac{m}{N}f_s\right) = H_d\left(\frac{m}{N}f_s\right)}$ for $m = 0, 1, 2, 3, \ldots\ldots, N - 1$

f_s : sampling frequency

若以 normalized frequency $F = f/f_s$ 表示

$\boxed{R\left(\frac{m}{N}\right) = H_d\left(\frac{m}{N}\right)}$ for $m = 0, 1, 2, 3, \ldots\ldots, N - 1$

(see page 110)

References：

- L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, N. J., 1975.
- B. Gold and K. Jordan, “A note on digital filter synthesis,” *Proc. IEEE*, vol. 56, no. 10, pp. 1717-1718, 1969.
- L. R. Rabiner and R. W. Schafer, “Recursive and nonrecursive realizations of digital filters designed by frequency sampling techniques,” *IEEE Trans. Audio and Electroacoust.*, vol. 19, no. 3, pp. 200-207. Sept. 1971.

設計方法：

Step 1 Sampling $H_d\left(\frac{m}{N}\right)$ for $m = 0, 1, 2, 3, \ldots\ldots, N-1$

Step 2 $$\boxed{r_1[n] = \frac{1}{N}\sum_{m=0}^{N-1} H_d\left(\frac{m}{N}\right)\exp\left(j\frac{2\pi m}{N}n\right)}$$ $n = 0, 1, \ldots., N-1$

換句話說，$r_1[n]$ 是 $H_d(m/N)$ 的 inverse discrete Fourier transform (IDFT)

Step 3 When N is odd

$r[n] = r_1[n]$ for $n = 0, 1, \ldots., k$ $k = (N-1)/2$

$r[n] = r_1[n+N]$ for $n = -k, -k+1, \ldots., -1$

注意：$r[n]$ 的區間為 $n \in [-(N-1)/2, (N-1)/2]$

Step 4 $h[n] = r[n-k]$ $k = (N-1)/2$

Proof:

注意，若 $R(F)$是 $r[n]$ 的 discrete-time Fourier transform

$$R(F) = \sum_{n=-\infty}^{\infty} r[n] e^{-j2\pi Fn} = \sum_{n=-k}^{k} r[n] e^{-j2\pi Fn} = \sum_{n=0}^{k} r[n] e^{-j2\pi Fn} + \sum_{n=-k}^{-1} r[n] e^{-j2\pi Fn}$$

$$= \sum_{n=0}^{k} r[n] e^{-j2\pi Fn} + \sum_{n=-k}^{-1} r_1[n+N] e^{-j2\pi F(n+N)} = \sum_{n=0}^{N-1} r_1[n] e^{-j2\pi Fn}$$

when $F = m/N$

(We apply the fact where $e^{-j2\pi Fn} = e^{-j2\pi F(n+N)}$ when $F = m/N$)

$$R(m/N) = \sum_{n=0}^{N-1} r_1[n] \exp\left(-j\frac{2\pi m}{N} n\right)$$

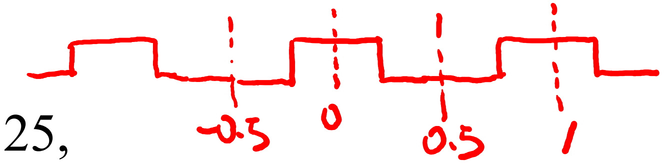
又由於 $r_1[n]$ 是 $H_d(m/N)$ 的 inverse discrete Fourier transform (IDFT)

$$H_d\left(\frac{m}{N}\right) = DFT\{r_1[n]\} = \sum_{m=0}^{N-1} r_1[n] \exp\left(-j\frac{2\pi m}{N} n\right)$$

所以 $R\left(\frac{m}{N}\right) = H_d\left(\frac{m}{N}\right)$

Example: $N = 17$

$H_d(F) = H_d(F+1)$

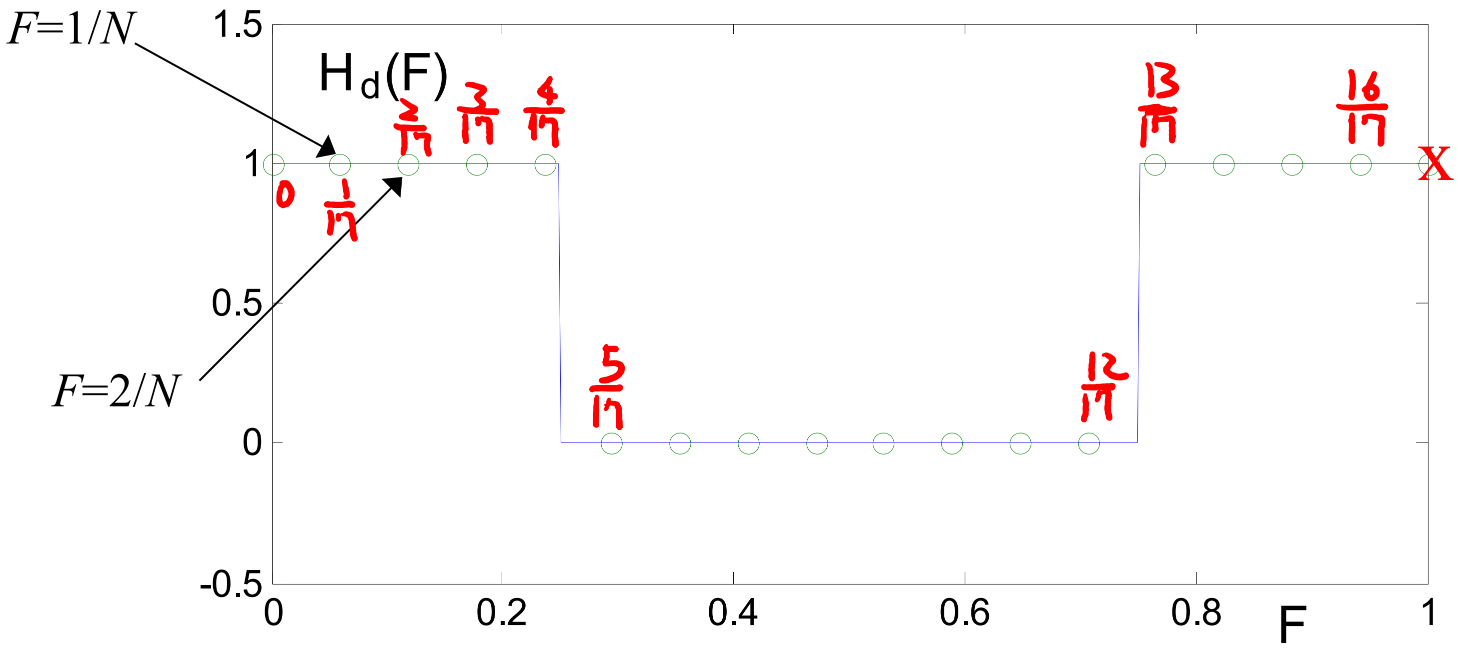


$H_d(F) = 1$ for $-0.25 < F < 0.25$,

$H_d(F) = 0$ for $-0.5 < F < -0.25$, $0.25 < F < 0.5$

(Step 1)

[1 , 1 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1]



(Step 2)

$r_1[n] = \text{ifft}([1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1]$

$= [0.529\ \ 0.319\ \ \text{-}0.030\ \ \text{-}0.107\ \ 0.032\ \ 0.066\ \ \text{-}0.035\ \ \text{-}0.049\ \ 0.040$
$0.040\ \ \text{-}0.049\ \ \text{-}0.035\ \ 0.066\ \ 0.032\ \ \text{-}0.107\ \ \text{-}0.030\ \ 0.319]$ $\quad n = 0 \sim 16$

(Step 3)

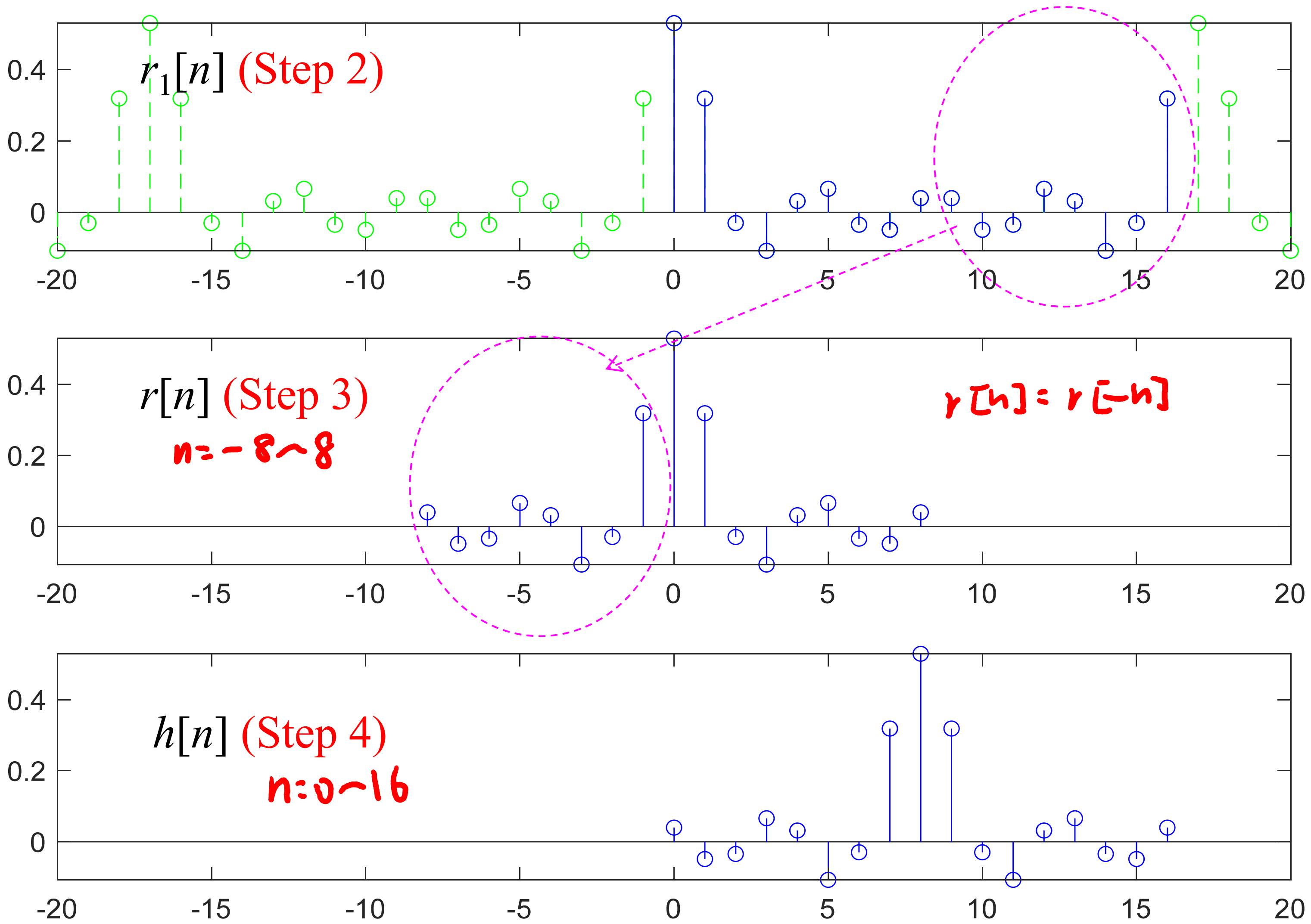
$r[n] = [0.040\ \ \text{-}0.049\ \ \text{-}0.035\ \ 0.066\ \ 0.032\ \ \text{-}0.107\ \ \text{-}0.030\ \ 0.319\ \ 0.529$
$0.319\ \ \text{-}0.030\ \ \text{-}0.107\ \ 0.032\ \ 0.066\ \ \text{-}0.035\ \ \text{-}0.049\ \ 0.040]$ $\quad n = -8 \sim 8$

(Step 4)

若我們希望所設計出來的 filter $h[n]$ 有值的區域為 $n \in [0, 16]$

$h[n] = r[n-8]$

$= [0.040\ \ \text{-}0.049\ \ \text{-}0.035\ \ 0.066\ \ 0.032\ \ \text{-}0.107\ \ \text{-}0.030\ \ 0.319\ \ 0.529$
$0.319\ \ \text{-}0.030\ \ \text{-}0.107\ \ 0.032\ \ 0.066\ \ \text{-}0.035\ \ \text{-}0.049\ \ 0.040]$ $\quad n = 0 \sim 16$


$r_1[n]$ (Step 2)
$r[n]$ (Step 3)
n = -8 ~ 8
r[n] = r[-n]
$h[n]$ (Step 4)
n: 0 ~ 16

Frequency Response in terms of $R(F)$

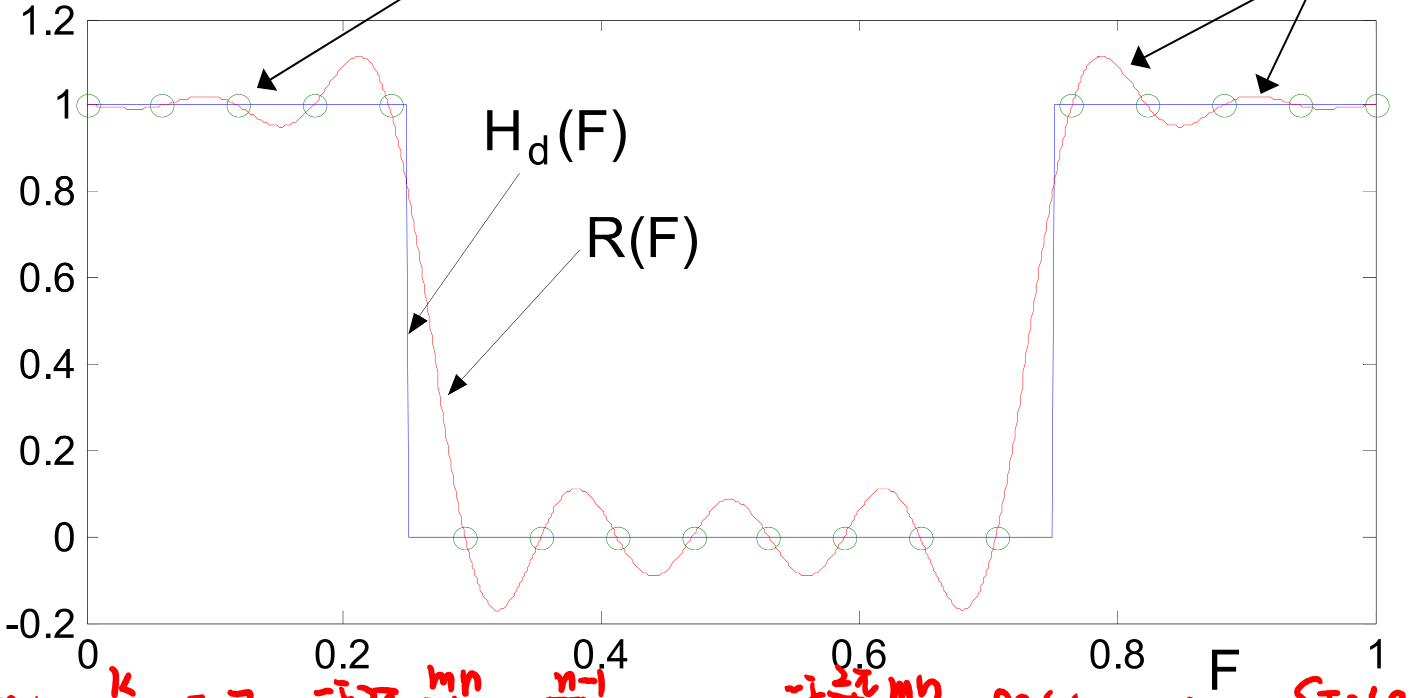
$$R\left(\frac{m}{N}\right) = H_d\left(\frac{m}{N}\right)$$

$$R(F) = \sum_{n=-\infty}^{\infty} r[n] e^{-j2\pi Fn}$$

($r[n]$ is from Step 3)

$R(F)$在sample frequency 等於$H_d(F)$

Error 非 equal-ripple



When $F=\frac{m}{N}$

$$R\left(\frac{m}{N}\right) = \sum_{n=-k}^{k} r[n] e^{-j2\pi \frac{mn}{N}} = \sum_{n=0}^{N-1} r_1[n] e^{-j\frac{2\pi}{N}mn} = fft(r_1[n])$$

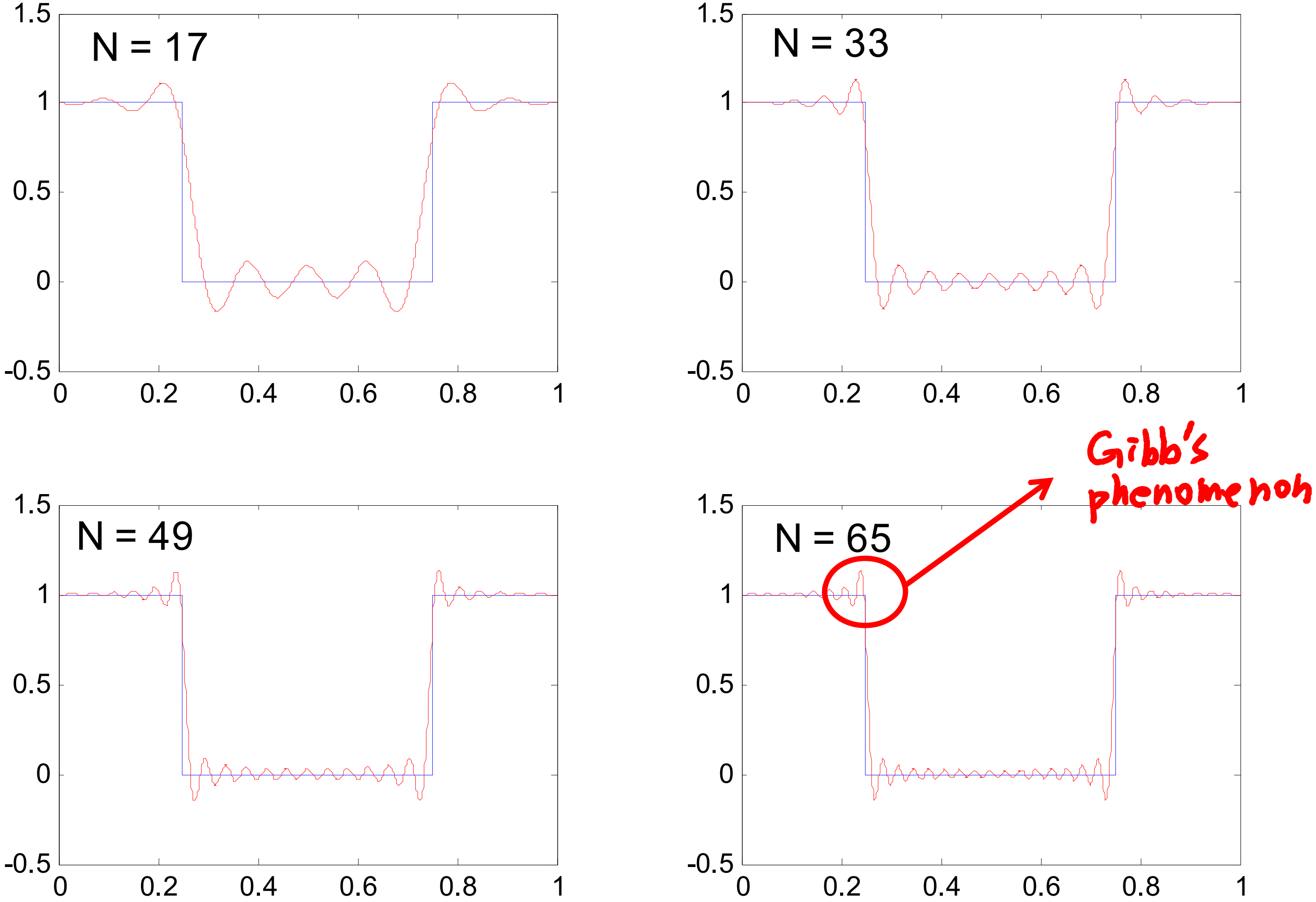
$k=\frac{N-1}{2}$

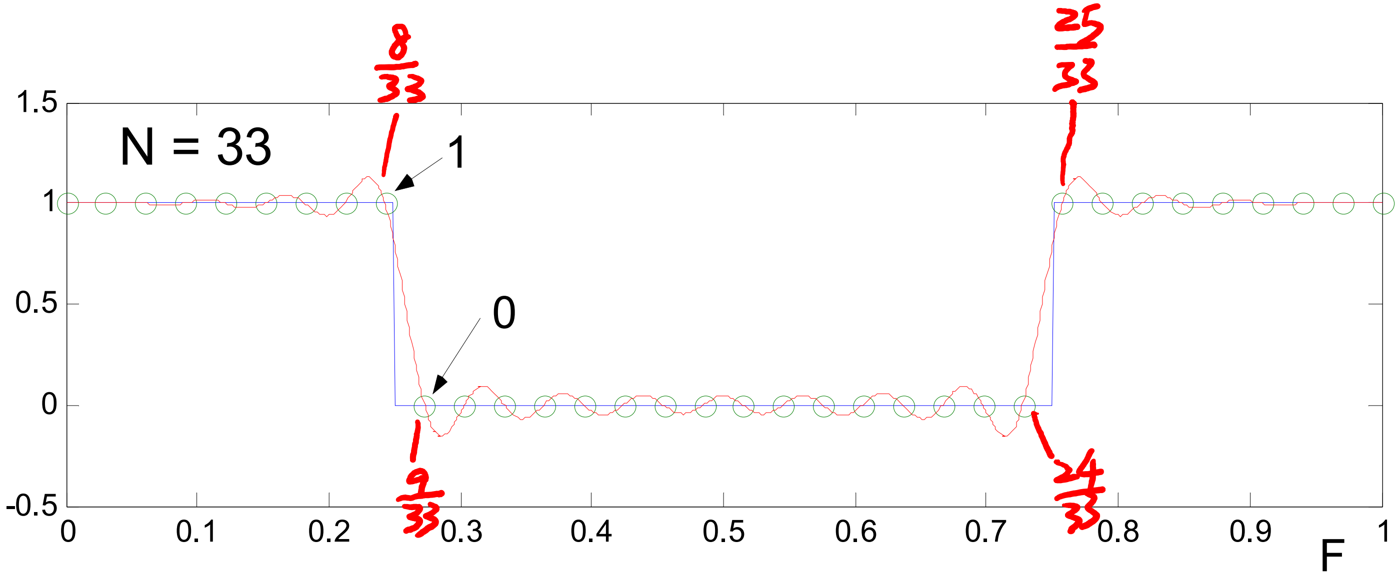
Since $r_1[n] = ifft\left(H_d\left(\frac{m}{N}\right)\right)$

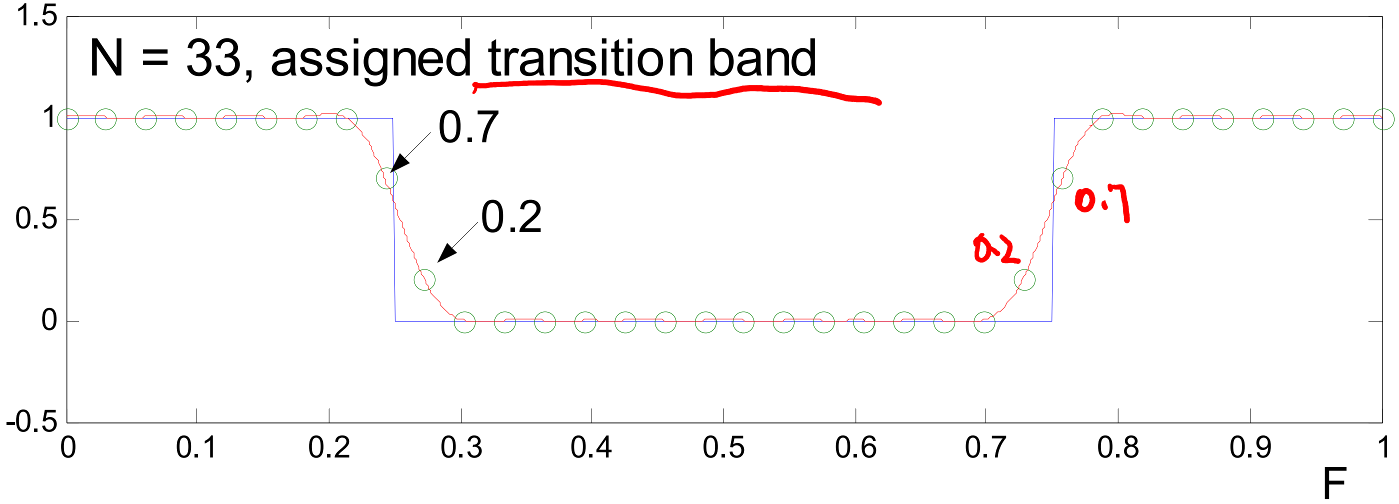
- The approximation error tends to be highest around the transition band and smaller in the passband and stopband regions.

$\therefore H_d\left(\frac{m}{N}\right) = R\left(\frac{m}{N}\right)$

Error is larger at the edge




N = 33
1
0
F
8/33
9/33
24/33
25/33


N = 33, assigned transition band
0.7
0.2
F
0.2
0.7

討論：

(1) Frequency sampling 的方法頗為簡單且直觀，

但得出來的 filter 不為 optimal

(2) Ripple 大小變化的情形，介於 MSE 和 Minimax 之間

(3) 可以用設定 transition band 的方式，來減少 passband 和 stopband 的 ripple。(In transition band, $R(m/N) \neq H_d(m/N)$).

然而，如何設定 transition band $R(m/N)$ 的值，讓 passband 和 stopband 的 ripple 變為最小 ……… 需要作 linear programming。

(運算時間不少)

◎ 2-N 三種 FIR Digital Filter 設計方法的比較

- 以設計方法而論

  MSE : integrals, matrix

  Minimax : most complicated (recursive)

  frequency sampling : simplest (ifft)

- 以方法的限制而論

  MSE : no constraint

  Minimax : often used in pass-stop band filters; transition band is necessary

  frequency sampling : weight function cannot be applied

- 以效果而論

  MSE : minimize the mean square error

  Minimax : minimize the maximal error

  frequency sampling : not optimal

The 4th Method for the FIR Filter Design?

$$x[n] \xrightarrow{\text{DFT}} X[m] \longrightarrow Y[m] = X[m]H[m] \xrightarrow{\text{IDFT}} y[n]$$

$$H[m] = 1 \text{ for passband}$$

$$H[m] = 0 \text{ for stopband}$$

complexity of DFT : $O(N\log_2 N)$

complexity of other FIR methods $O(N)$

N: length(x)   if $N \gg$ filter length

Q: Why do we not apply the method?

◉ 2-O Implementation of the FIR Filter

$$y[n] = x[n] * h[n]$$

convolution

(1) 使用 FFT

$$y[n] = IFFT[\ FFT\{x[n]\} \times FFT\{h[n]\}\ ]$$

(2) 直接作 summation 即可

(3) Sectioned FFT

$$y[n] = x[n] * h[n]$$

(2) 直接作 summation

假設 $h[n] = 0$ for $n < 0$ and $n \geq N$

$$y[n] = h[0]x[n] + h[1]x[n-1] + \ldots.. + h[N-2]x[n-N+2] + h[N-1]x[n-N+1]$$

- 若 $h[n] = h[N-1-n]$ (even symmetric), N 為odd

$$y[n] = h[0](x[n] + x[n-N+1]) + h[1](x[n-1] + x[n-N+2])$$
$$+ \ \ldots.. + h[k-1](x[n-k+1] + x[n-N+k]) + h[k]\, x[n-k]$$

$$k = (N-1)/2$$

3. Theories about IIR Filters

◉ 3-A Minimum-Phase Filter

• FIR filter: The length of the impulse response is **finite**

usually **linear phase** (i.e., even or odd impulse response)

always stable

• IIR filter: (i) May be unstable

(ii) The length of the impulse response is **infinite**.

(Question): Is the implementation also a problem?

Advantages of the IIR filter:

References

- A. Antoniou, *Digital Filters: Analysis and Design*, McGraw-Hill, New York, 1979.
- T. W. Parks and C. S. Burrus, *Digital Filter Design*, John Wiley, New York, 1989.
- O. Herrmann and W. Schussler, 'Design of nonrecursive digital filters with minimum phase,' *Elec. Lett.*, vol. 6, no. 11, pp. 329-330, 1970.
- C. M. Rader and B. Gold, 'Digital filter design techniques in the frequency domain,' *Proc. IEEE*, vol. 55, pp. 149-171, Feb. 1967.
- R. W. Hamming, *Digital Filters*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- F. W. Isen, *DSP for MATLAB and LabVIEW*, Morgan & Claypool Publishers, 2009.

• IIR filter: The length of the impulse response is **infinite**.

→ try to make the energy concentrating on the region near to $n = 0$

.

→ try to make both the forward and the inverse transforms stable

using the **minimum phase filter**.

Advantages of the minimum phase filter

① (All the poles and all the zeros are within the unit circle.)

It makes both the forward and the inverse transforms stable

② It makes the impluse response concentrated around 0.

***Z* transform** $H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n}$

***H*(*z*) can be expressed as** $C \dfrac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$

$$= C\, z^{R-S} \frac{\left(1-z_1 z^{-1}\right)\left(1-z_2 z^{-1}\right)\left(1-z_3 z^{-1}\right)\cdots\cdots\left(1-z_R z^{-1}\right)}{\left(1-p_1 z^{-1}\right)\left(1-p_2 z^{-1}\right)\left(1-p_3 z^{-1}\right)\cdots\cdots\left(1-p_S z^{-1}\right)}$$

$p_1, p_2, p_3, \ldots\ldots, p_S$: **poles** $z_1, z_2, z_3, \ldots\ldots, z_R$: **zeros**

- **Stable filter**: All the poles are within the unit circle.
- **Minimum phase filter**: All the poles and all the zeros are within the unit circle.

  i.e., $|p_s| \leq 1$ and $|z_r| \leq 1$

If any pole falls outside the unit circle ($|p_s| > 1$), then the impulse response of the filter is not convergent.

$$H(z) = C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$$

$$= C\,z^{R-S}\,\frac{(1-z_1z^{-1})(1-z_2z^{-1})(1-z_3z^{-1})\cdots\cdots(1-z_Rz^{-1})}{(1-p_1z^{-1})(1-p_2z^{-1})(1-p_3z^{-1})\cdots\cdots(1-p_Sz^{-1})}$$

$$= C\,z^{R-S}\left(Q(z^{-1}) + \frac{A_1}{1-p_1z^{-1}} + \frac{A_2}{1-p_2z^{-1}} + \cdots\cdots + \frac{A_S}{1-p_Sz^{-1}}\right)$$

If $R \geq S$, $Q(z^{-1})$ is a polynomial of z^{-1} with degree R–S.

$$Q(z^{-1}) = q_0 + q_1z^{-1} + \cdots + q_{R-S}z^{-(R-S)}$$

$$Z\left(A_s p_s^n u[n]\right) = \sum_{n=0}^{\infty} A_s p_s^n z^{-n} = A_s \sum_{n=0}^{\infty} (p_s z^{-1})^n = A_s \frac{1}{1-p_s z^{-1}}$$

If $R < S$, $Q(z^{-1}) = 0$.

$$h_s[n] = Z^{-1}\left(\frac{A_s}{1-p_sz^{-1}}\right) = A_s p_s^n u[n]$$

Z^{-1} : inverse Z transform

~~for $n = 1, 2, 3, 4,$~~ .....

If $|p_s| < 1$, $\quad \lim_{n\to\infty} h_s[n] = 0$

If $|p_s| > 1$, $\quad \lim_{n\to\infty} h_s[n] \to \pm\infty$

$u[n] = 1$ *for* $n \geq 0$,

$u[n] = 0$ *otherwise*.

Therefore,

$$h[n] = C\left(q[\underbrace{n + R - S}_{\text{FIR filter}}] + \sum_{s=1}^{S} h_s[\underbrace{n + R - S}_{\text{geometric series}}] \right)$$

where

$$q[n] = q_n \quad for\ n = 1, 2, \cdots R - S$$

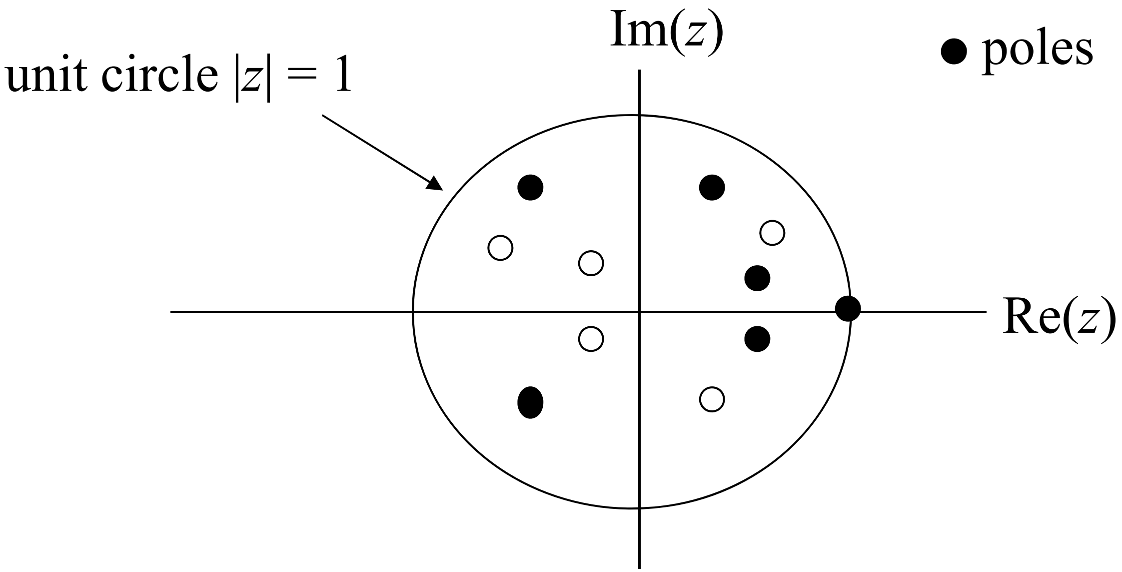
$$h_s[n] = A_s p_s^n u[n] \qquad \text{for } s = 1, 2, \ldots.., S$$

Thus, **the minimum phase filter is stable and causal**.

The **inverse** of the **minimum phase filter is stable and causal**.

$$H(z) = C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$$

$$H^{-1}(z) = C^{-1}\, z^{S-R}\frac{\left(1-p_1z^{-1}\right)\left(1-p_2z^{-1}\right)\left(1-p_3z^{-1}\right)\cdots\cdots\left(1-p_Sz^{-1}\right)}{\left(1-z_1z^{-1}\right)\left(1-z_2z^{-1}\right)\left(1-z_3z^{-1}\right)\cdots\cdots\left(1-z_Rz^{-1}\right)}$$



◉ 3-B Converting an IIR Filter into a Minimum Phase Filter

$$H(z)=C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$$

不影響
amplitude

Suppose that z_2 is not within the unit circle, $|z_2|>1$

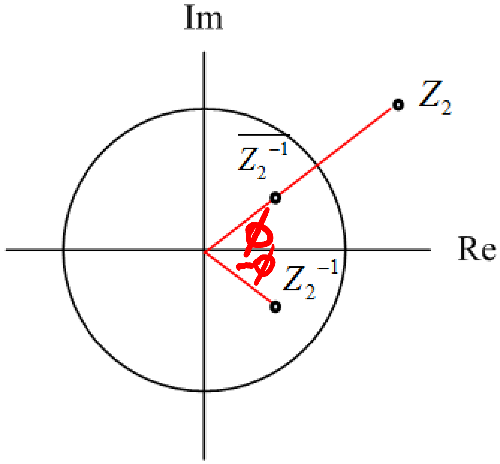
$$H_1(z)=C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}\times\boxed{z_2\frac{z-\overline{(z_2^{-1})}}{z-z_2}}$$

replace z_2 by $\overline{z_2^{-1}}$

$$=z_2C\frac{(z-z_1)\left(z-\overline{(z_2^{-1})}\right)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$$

The upper bar means conjugation.

$$\begin{cases} Z_2 = Ae^{j\phi} \\ \overline{Z_2^{-1}} = \dfrac{1}{A}e^{j\phi} \\ Z_2^{-1} = \dfrac{1}{A}e^{-j\phi} \end{cases}$$



In fact, if $z = e^{j2\pi F}$ (see page 29), then $H(z)$ and $H_1(z)$ only differ in phase,

$|H_1(F)| = |H(F)|$

(proof):

$z - \overline{(z_2^{-1})} = z\left(1 - \overline{(z_2^{-1})} z^{-1}\right) = z\left(\overline{z_2^{-1}}\right)\left(\overline{z_2} - z^{-1}\right)$

amplitudes are 1

$$\left| z_2 \frac{z - \overline{(z_2^{-1})}}{z - z_2} \right| = \left| z_2 \overline{(z_2^{-1})}\, z \frac{\overline{z_2} - z^{-1}}{z - z_2} \right| = \left| z_2 \overline{(z_2^{-1})}\, z \frac{\overline{z_2 - z}}{z - z_2} \right| = 1$$

when $z = e^{j2\pi F}$, $z^{-1} = \overline{z}$ when $z = e^{j2\pi F}$

(單位圓上)

- We call the filter whose amplitude response is always 1 as the **all-pass filter**.

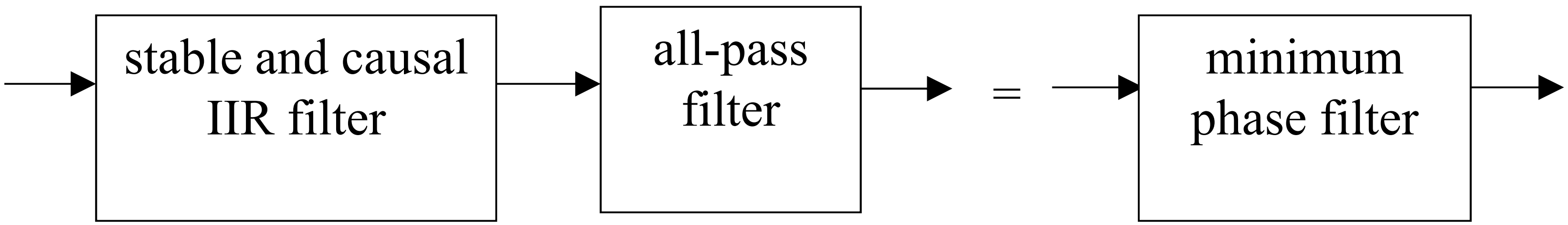
$$z_2 \frac{z - \overline{(z_2^{-1})}}{z - z_2}$$ **is an all-pass filter**

- One can also use the similar way to move poles from the outside of the unit circle into the inside of the unit circle.

Any stable IIR filter can be expressed as a cascade of the **minimum phase filter** and an **all-pass filter**.

$H(z)$:IIR filter, $H_{mp}(z)$: minimum phase filter, $H_{ap}(z)$: allpass filter

$$H(z)H_{ap}(z) = H_{mp}(z)$$



Example:

stable (|pole| = 0.9 < 1
but not minimum phase
|1.6+1.2j| = 2 > 1

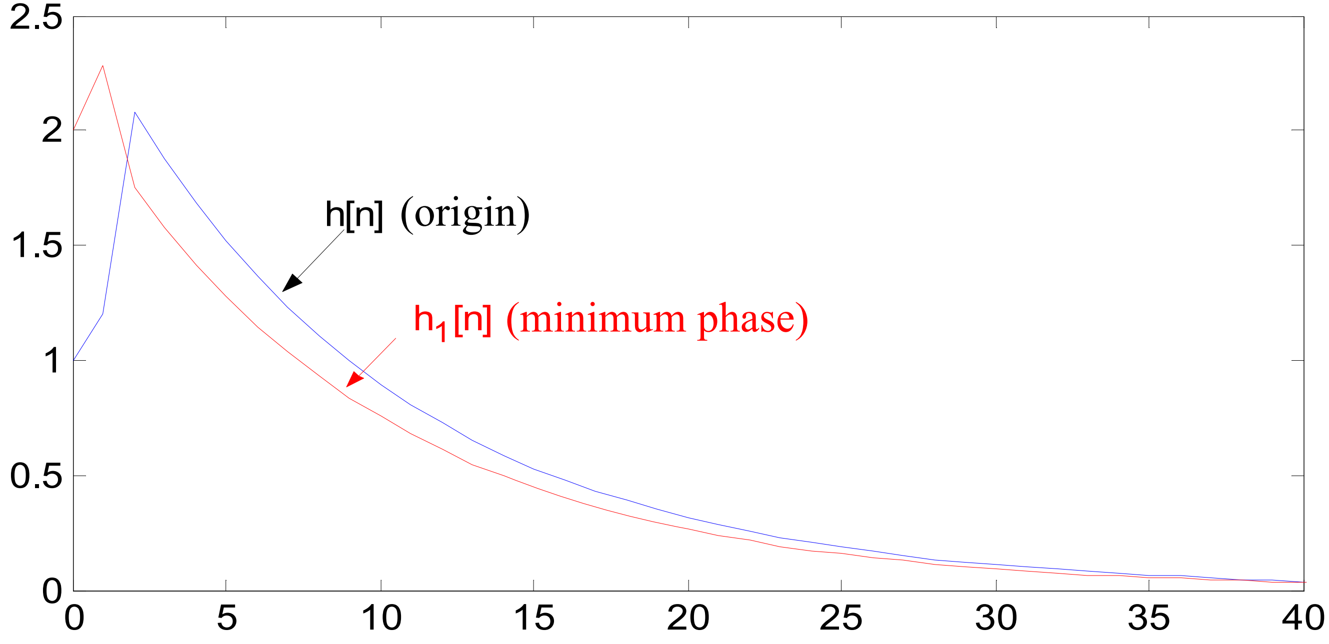
$$H(z) = \frac{(z+0.6)[z-(1.6+1.2j)]}{z-0.9}$$

$$\frac{1}{1.6+1.2j} = 0.4-0.3j \text{ conjugates with } 0.4+0.3j$$

$z_2 = 1.6 + 1.2j$
$= 2(0.8 + 0.6j)$
$z_2^{-1} = \frac{1}{2}(0.8 - 0.6j)$
$\overline{z_2^{-1}} = \frac{1}{2}(0.8 + 0.6j)$
$= 0.4 + j0.3$

$$H_1(z) = (1.6+1.2j)\frac{(z+0.6)[z-(0.4+0.3j)]}{z-0.9}$$

$h[n]$, $h_1[n]$ are the impulse response of the two filters $H(z)$ and $H_1(z)$



◉ 3-C The Meaning of Minimum Phase

Another important advantage of the minimum phase filter：

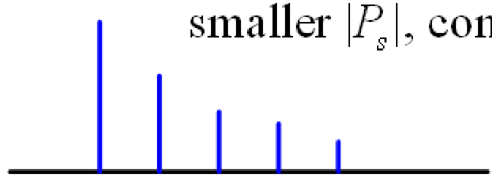
The energy concentrating on the region near to *n* = 0.

$$H(z)=C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)}$$

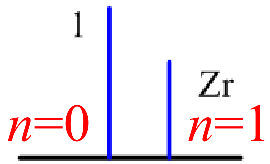
$$=Cz^{R-S}\frac{(1-z_1z^{-1})(1-z_2z^{-1})(1-z_3z^{-1})\cdots\cdots(1-z_Rz^{-1})}{(1-p_1z^{-1})(1-p_2z^{-1})(1-p_3z^{-1})\cdots\cdots(1-p_Sz^{-1})}$$

$Z^{-1}\left[\dfrac{1}{1-p_sz^{-1}}\right]=a_s[n]$ $\quad a_s[n]=0$ when $n<0$ $\quad a_s[n]=p_s^n$ when $n\geq 0$

smaller $|P_s|$, converge faster



$Z^{-1}\left[1-z_rz^{-1}\right]=b_r[n]$ $\quad b_r[0]=1,\ b_r[1]=-z_r,\ b_r[n]=0$ otherwise



Phase is related to delay

$$x[n-\tau] \xrightarrow[]{\text{discrete time Fourier transform}} e^{-j2\pi f\tau\Delta_t}X(f)$$

Minimum phase $\rightarrow$ Minimum delay

$$H(z) = Cz^{R-S}\frac{\left(1-z_1z^{-1}\right)\left(1-z_2z^{-1}\right)\left(1-z_3z^{-1}\right)\cdots\cdots\left(1-z_Rz^{-1}\right)}{\left(1-p_1z^{-1}\right)\left(1-p_2z^{-1}\right)\left(1-p_3z^{-1}\right)\cdots\cdots\left(1-p_Sz^{-1}\right)}$$

The multiplications in the Z domain (frequency domain) are equivalent to the convolutions in the time domain, so we could analyze each term individually in the previous page!!

(Question): How about the case of $|p_n| = 1$ or $|z_n| = 1$?

Note:

$$Z^{-1}\left[\frac{1}{1-p_s z^{-1}}\right] = a_s[n] \qquad a_s[n] = 0 \text{ when } n < 0 \qquad a_s[n] = p_s^n \text{ when } n \geq 0$$

When $|p_n| = 1$, the response is finite but the energy is infinite.

附錄四：查資料的方法

(1) Google 學術搜尋 (不可以不知道)

網址： http://scholar.google.com.tw/

(太重要了，不可以不知道) 只要任何的書籍或論文，在網路上有電子版，都可以用這個功能查得到

輸入關鍵字，或標題、作者

再按「搜尋」，就可找到想要的資料

註：由於版權，大部分的論文必需要在學校上網才可以下載

按搜尋之後將出現相關文章

點選後，可找到該學術文章的原始出處和相關的電子檔

可限定要找的文章的刊登時間

若要引用這篇論文，可點選此按鈕，會出現三種不同格式的引用方式

(2) 尋找 IEEE 的論文

http://ieeexplore.ieee.org/Xplore/guesthome.jsp

註：除非你是 IEEE Member，否則必需要在學校上網，才可以下載到 IEEE 論文的電子檔

(3) Google

(4) Wikipedia

(5) ChatGPT

(6) 數學的百科網站

http://eqworld.ipmnet.ru/index.htm

有多個 tables，以及對數學定理的介紹

(7) 傳統方法：去圖書館找資料

台大圖書館首頁 http://www.lib.ntu.edu.tw/

或者去 http://www.lib.ntu.edu.tw/tulips

(8) 查詢其他圖書館有沒有我要找的期刊

台大圖書館首頁 ⟶ 其他聯合目錄 ⟶ 全國期刊聯合目錄資料庫

如果發現其他圖書館有想要找的期刊，可以申請「館際合作」，請台大圖書館幫忙獲取所需要的論文的影印版

台大圖書館首頁 ⟶ 館際合作

(9) 查詢其他圖書館有沒有我要找的書

「台大圖書館首頁」⟶「其他圖書館」

(10) 找尋電子書

「台大圖書館首頁」⟶「電子書」或「免費電子書」

(11) 中文電子學位論文服務

http://www.cetd.com.tw/ec/index.aspx

可以查到多個碩博士論文 (尤其是 2006年以後的碩博士論文) 的電子版

(12) 想要對一個東西作入門但較深入的了解:

看書會比看 journal papers 或 Wikipedia 適宜

如果實在沒有適合的書籍，可以看 “review”， “survey” ，或 “tutorial” 性質的論文

(13) 有了相當基礎之後，再閱讀 journal papers

(以 Paper Title， Abstract， 以及其他 Papers 對這篇文章的描述，

來判斷這篇 journal papers 應該詳讀或大略了解即可)

(14) 積分查詢網站：http://integrals.wolfram.com/index.jsp

(15) 可以查詢數學公式的工具書 (Handbooks)

M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 3rd Ed., New York, 2009.（已經有電子版）

M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions, with Formula, Graphs and Mathematical Tables*, Dover Publication, New York, 1965.

A. Jeffrey, *Handbook of Mathematical Formulas and Integrals*, Academic Press, San Diego, 2000.

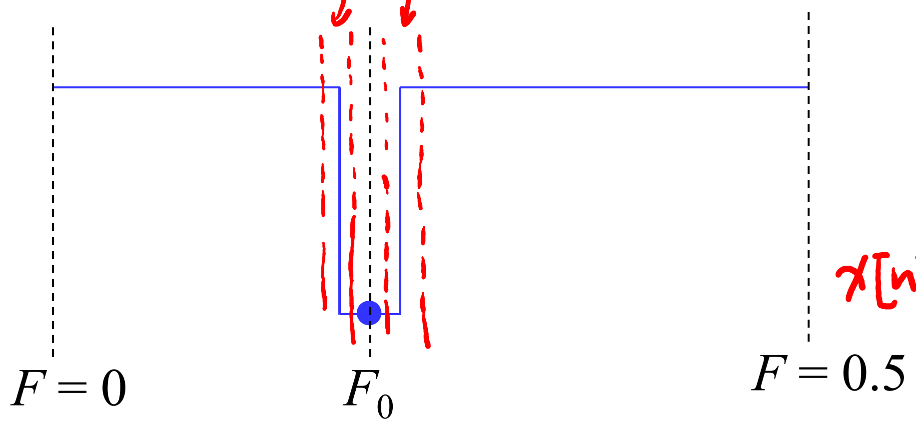
4. Some Popular Filters

◉ 4-A Popular Filters (1): Pass-Stop Band Filters

highpass bandpass lowpass allpass bandstop

very narrow transition band

notch filter: 想濾掉 $F = F_0$ 的 noise，但 stop band 越小越好



$x[n] — C_{F_0} e^{j2\pi F_0 n}$

Question: Why the notch filter is hard to design?

References

[1] K. Hirano, S. Nishimura, and S. K. Mitra, “Design of digital notch filters,” *IEEE Trans. Commun.*, vol. 22, no. 7, pp. 964-970, Jul. 1974.

[2] T. H. Yu, S. K. Mitra and H. Babic, “Design of linear phase FIR notch filters,” in *Sadhana*, Springer, vol. 15, issue 3, pp. 133-155, Nov. 1990.

[3] S. C. D. Roy, S. B. Jain, and B. Kumar, "Design of digital FIR notch filters," *Vision, Image and Signal Processing, IEE Proceedings*, vol.141, no. 5, pp.334-338, Oct. 1994.

[4] S. C. Pei and C. C. Tseng, “IIR multiple notch filter design based on allpass filter,” *IEEE Trans. Circuits Syst. II*, vol. 44, no.2, pp. 133-136, Feb. 1997.

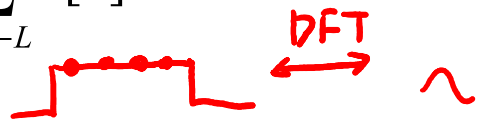
[5] C. C. Tseng and S. C. Pei, “Stable IIR notch filter design with optimal pole placement,” *IEEE Trans. Signal Processing*, vol. 49, issue 11, pp. 2673-2681, Nov. 2001.

◎ 4-B Popular Filters (2): Smoother (Weighted Average)

近似 low-pass filter

最簡單的 smoother:

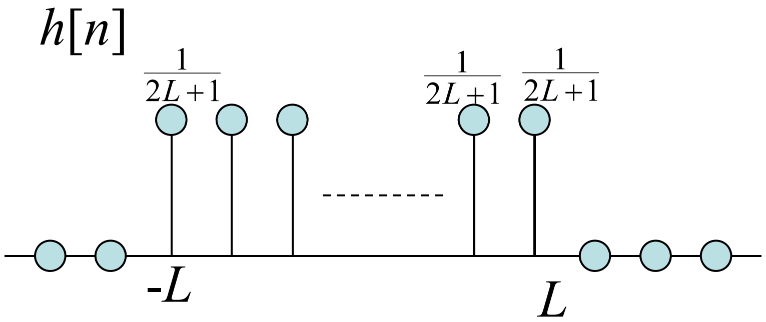
find the average $y[n] = \dfrac{1}{2L+1}\displaystyle\sum_{\tau=n-L}^{n+L} x[\tau]$



可改寫成

$y[n] = x[n] * h[n]$

$h[n]$ 如右圖



$$y[n] = \sum_{\tau} x[n-\tau]h[\tau] = \sum_{\tau=-L}^{L} x[n-\tau]\frac{1}{2L+1} = \frac{1}{2L+1}\sum_{\tau=-L}^{L} x[n+\tau]$$

一般型態的 smoother

$$y[n] = x[n] * h[n] = \sum_{\tau} x[n-\tau] h[\tau]$$

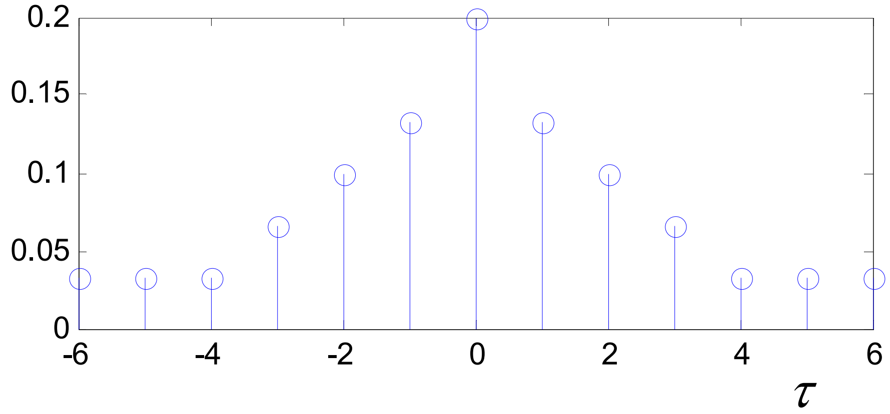
$= x[n]h[0] + (x[n+1] + x[n-1])h[1]$
$+ (x[n+2] + x[n-2])h[2]$
$+ (x[n+3] + x[n-3]h[3]$
$+ \cdots\cdots$

Choose (1) $h[n] = h[-n]$

(2) $|h[n_1]| \le |h[n_2]| \quad \text{if } |n_1| > |n_2|$

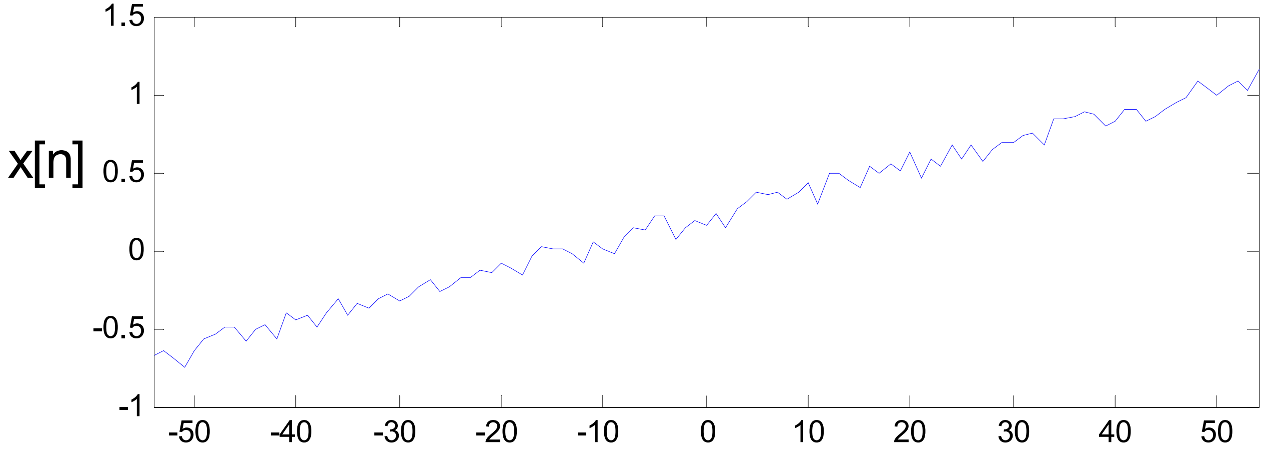
(3) $h[n] \ge 0$ for all n

(4) $\sum_{\tau} h[\tau] = 1$

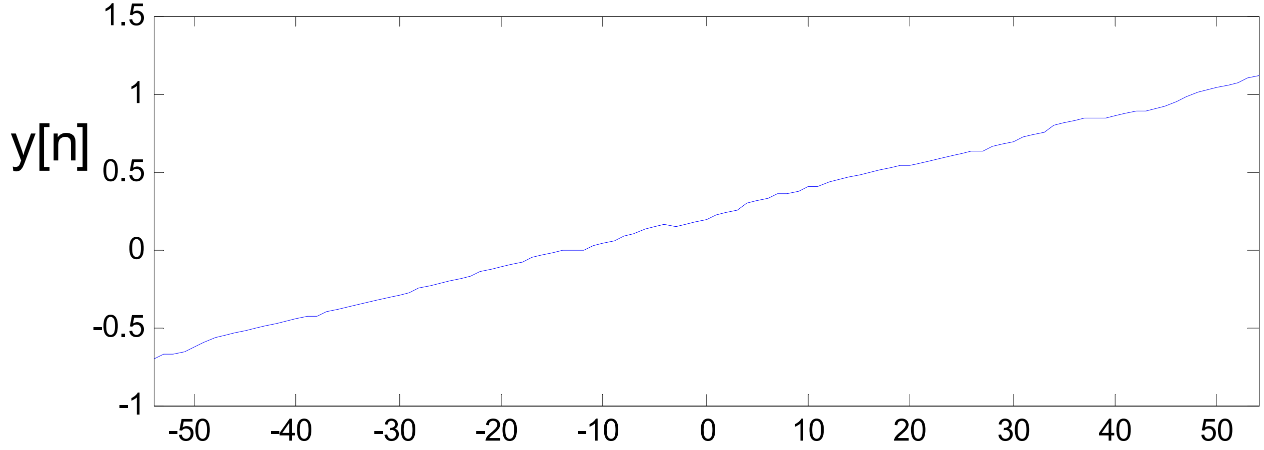


任何能量隨著 $|n|$ 遞減的 even function，都可以當成 smoother filter

Example:



After applying the smoother filter



Smoother 是一種 lowpass filter (但不為 pass-stop band filter)

思考: smoother 在信號處理上有哪些功用？

(i) extract trend

(ii) extract large-scaled features

(iii) noise removal

◉ 4-C Popular Filters (3): Family of Odd Symmetric Filters

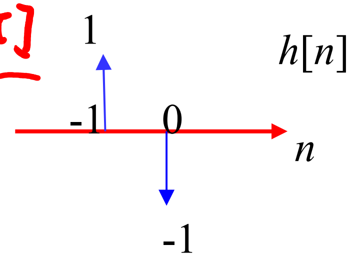
(a) Differentiation $H(f) = j2\pi f$ when $-f_s/2 < f < f_s/2$,

$$H(f) = H(f + f_s)$$

(b) Difference (一個簡單取代 differentiation 的方法)

$$x_1[n] = x[n] * h[n] = x[n+1] - x[n]$$

$\sum_{\tau} x[n-\tau]\, h[\tau]$



$h[n] = 1$ when $n = -1$, $h[n] = -1$ when $n = 0$,

$h[n] = 0$ otherwise

$$H(F) = j2e^{j\pi F}\sin(\pi F)$$

These two filters are equivalent only at low frequencies

$x[n] \xrightarrow{DTFT} X(F) \to X(F)H(F) \xrightarrow{IDTFT} x_H[n]$

(C) Discrete Hilbert Transform (IIR filter)

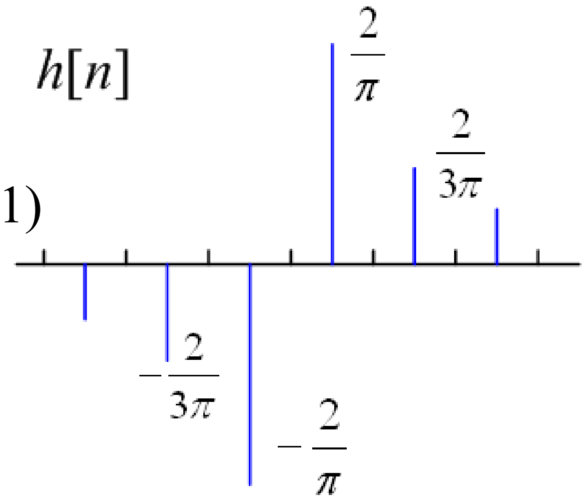
$H(F) = -j$ for $0 < F < 0.5$

$H(F) = j$ for $-0.5 < F < 0$ $\qquad H(F) = H(F+1)$

$H(0) = H(0.5) = 0$

$h[n] = \dfrac{2}{\pi n}$ when n is odd, $\quad h[n] = 0$ otherwise

single-sided band



Applications: (1) analytic function, (2) instantaneous frequency, (3) edge detection

for real $x[n]$, $X(F) = X^*(-F)$

Analytic function: $\quad x_a[n] = x[n] + jx_H[n]$

where $\quad x_H[n] = x[n] * h[n]$

$x[n] = \mathrm{Re}(x_a[n])$

$X_a(F) = X(F) + jX_H(F)$
$= X(F) + jH(F)X(F)$
$= (1 + jH(F))X(F)$

$$1 + jH(F) = \begin{cases} 2 & \text{if } F > 0 \\ 1 & \text{if } F = 0 \\ 0 & \text{if } F < 0 \end{cases}$$

$y[n] = \sum_{\tau} x[n-\tau]h[\tau]$

if $h[n] = -h[-n]$

$y[n] = h[1](x[n-1] - x[n+1])$
$+ h[2](x[n-2] - x[n+2])$
$+ h[3](x[n-3] - x[n+3])$
$+ \cdots$

(D) Edge Detection ← 近似 high-pass filter

(1) $h[n] = -h[-n]$

(2) $|h[n_1]| \le |h[n_2]|$ if $|n_1| > |n_2|$

or the shifted version of $h[n]$ satisfies the above two constraints.

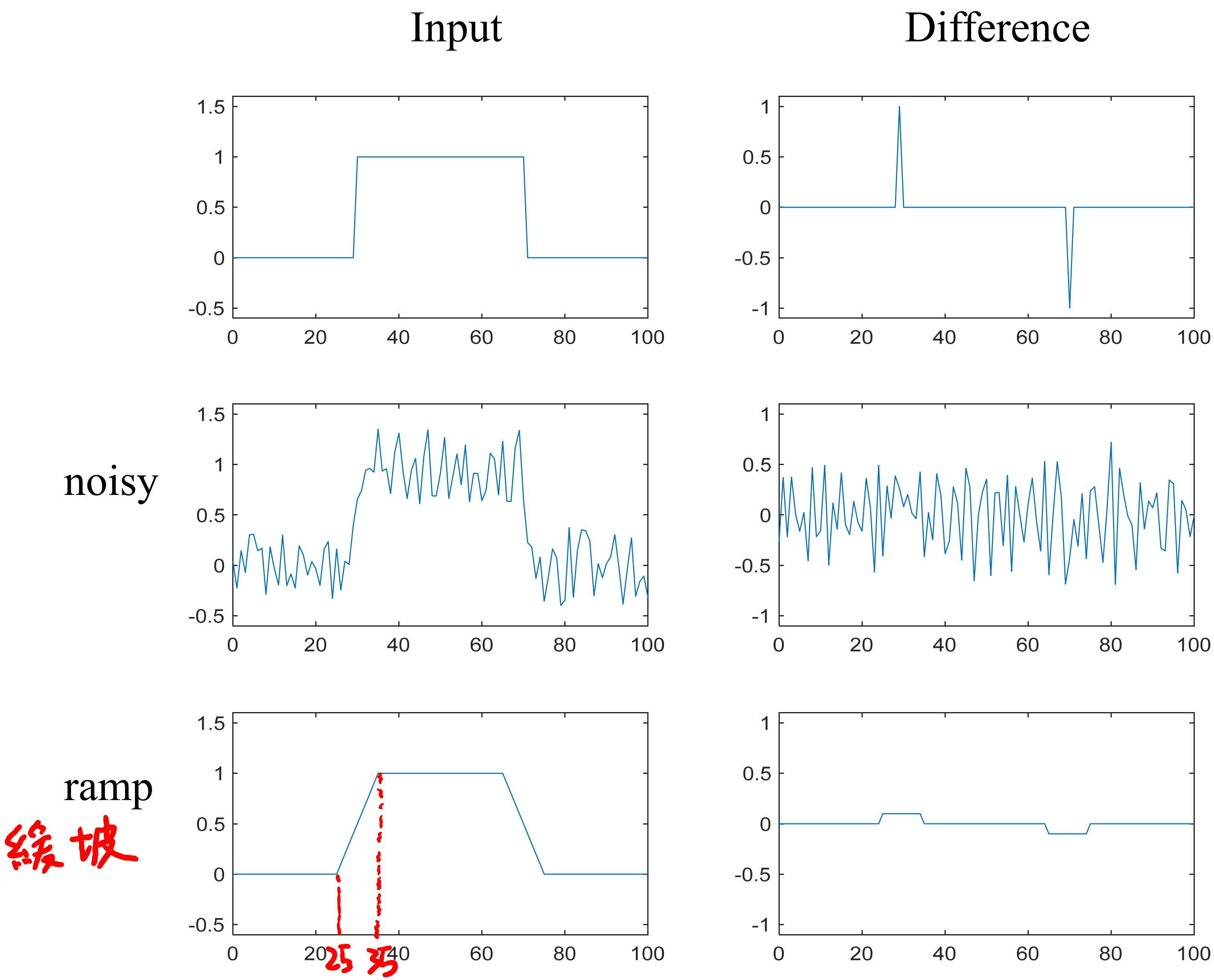
Difference 和 discrete Hilbert transform 都可用作 edge detection

(1) 任何能量隨著 $|n|$ 遞減的 odd function，都可以當成 edge detection filter

(2) The edge detection filter is in fact a matched filter.

Reference

S. C. Pei and J. J. Ding, "Short response Hilbert transform for edge detection," *IEEE Asia Pacific Conference on Circuits and Systems*, Macao, China, pp. 340-343, Dec. 2008.



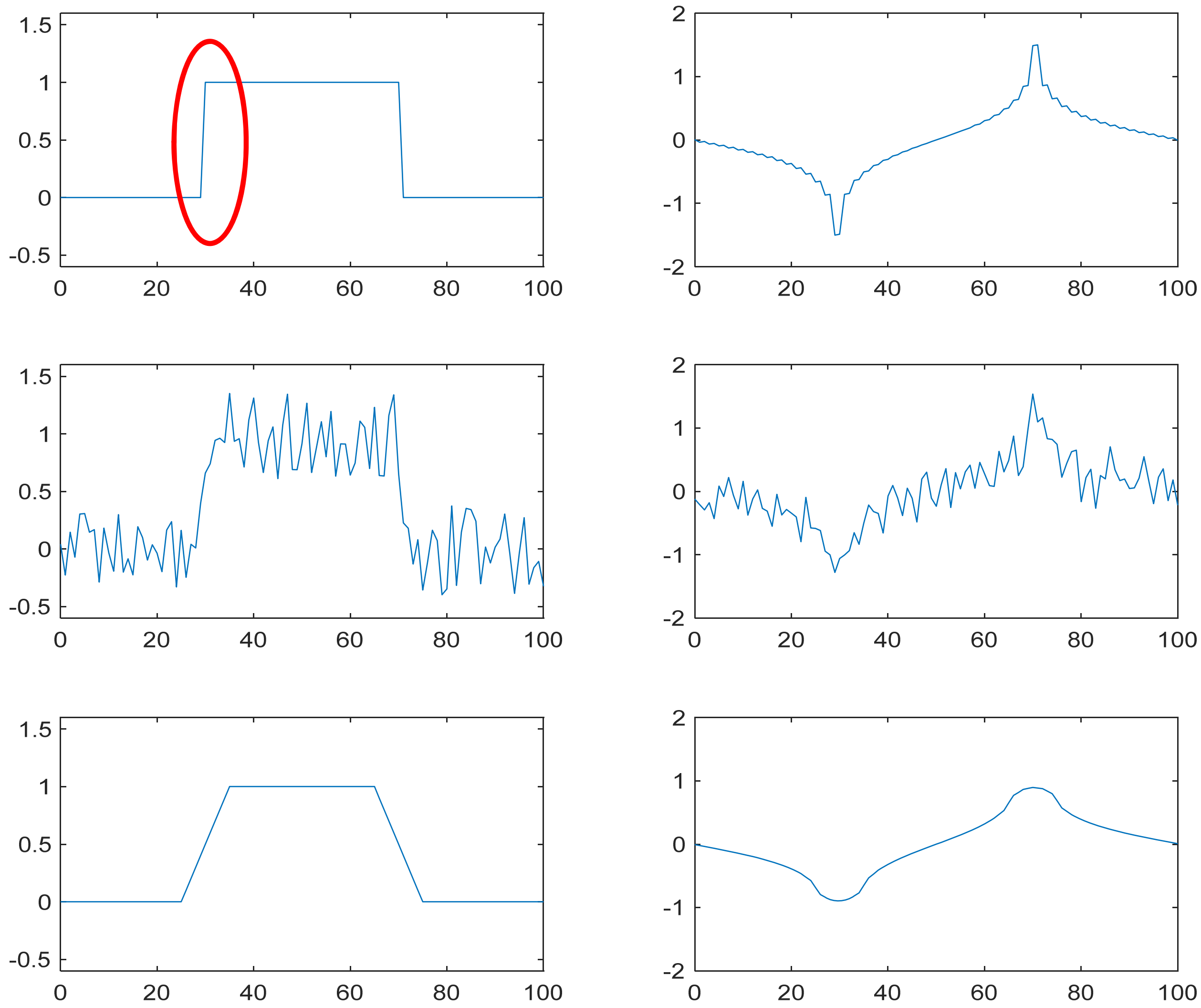
Input
Difference
noisy
ramp
緩坡
25
35

Input

Discrete Hilbert Transform

noisy

ramp



Other Well-know Edge Detection Filters:

Canny's Filter

L. Ding and A, Goshtasby. "On the Canny edge detector," *Pattern Recognition*, vol. 34, issue 3, pp. 721-725, 2001.

Sobel filter (A 2D Edge Detection Filter)

$$Im * M = 2Im(m,n+1) - 2Im(m,n-1) + Im(m-1,n+1) - Im(m-1,n-1) + Im(m+1,n+1) - Im(m+1,n-1)$$

horizontal $M =$

$$\begin{array}{cc} & \xrightarrow{n} \\ m\downarrow & \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \end{array}$$

vertical

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

45°

$$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

135°

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Sobel Operator (Horizontal)

$$\{2A[m, n+1] - 2A[m, n-1] + A[m+1, n+1] - A[m+1, n-1] + A[m-1, n+1] - A[m-1, n-1]\}/4$$

$$A * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} / 4$$

n →, m ↓

11	10	10	10	12	11	10	9	10
10	10	11	10	10	10	10	11	9
10	10	9	150	150	150	10	10	10
10	10	160	160	155	160	158	10	11
10	10	158	160	161	161	160	150	10
10	155	160	163	164	165	160	151	10
10	148	160	160	162	160	155	10	12
8	10	140	150	152	150	10	11	10
9	12	10	10	10	10	9	10	10

⇨

	0	35	35.25	0.25	-35.5	-35	-0.5	
	37.25	107.5	69	0	-69.25	-107.25	-37	
	111.75	147.5	33.5	0.25	-33.75	-112.75	-111	
	149	114.5	1.25	1	-0.75	-46.5	-149.25	
	149.5	44.5	3.25	1.25	-4	-47.25	-148.25	
	145.5	43	5	0.5	-40	-113.25	-109	
	103.75	72.5	6.5	0	-73	-107	-35.5	

Sobel Operator (Vertical)

$$\{2A[m+1, n] - 2A[m-1, n] + A[m+1, n+1] - A[m-1, n+1] + A[m+1, n-1] - A[m-1, n-1]\}/4$$

$$A * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} / 4$$

n →, m ↓

11	10	10	10	12	11	10	9	10
10	10	11	10	10	10	10	11	9
10	10	9	150	150	150	10	10	10
10	10	160	160	155	160	158	10	11
10	10	158	160	161	161	160	150	10
10	155	160	163	164	165	160	151	10
10	148	160	160	162	160	155	10	12
8	10	140	150	152	150	10	11	10
9	12	10	10	10	10	9	10	10

⇒

	-0.5	34.5	104.25	138.75	104	35	0.5	
	37.25	112	148.5	147.5	148.25	111.25	37	
	37.25	77	45	10.75	45.75	112.75	107.5	
	72.5	37	3.75	6.5	5.25	37.5	70.75	
	69.5	35.5	0.75	0.25	-1.5	-37.75	-70.75	
	-78	-49.5	-14.5	-13	-48	-113.75	-107.5	
	-105.75	-146.5	-150.5	-151	-149.5	-110.5	-37	

Sobel Operator (45°)

$$\{2A[m\text{-}1, n\text{+}1] - 2A[m\text{+}1, n\text{-}1] + A[m\text{-}1, n] - A[m\text{+}1, n] + A[m, n\text{+}1] - A[m, n\text{-}1]\}/4$$

$$A * \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} / 4$$

$\longrightarrow n$

$\downarrow m$

11	10	10	10	12	11	10	9	10
10	10	11	10	10	10	10	11	9
10	10	9	150	150	150	10	10	10
10	10	160	160	155	160	158	10	11
10	10	158	160	161	161	160	150	10
10	155	160	163	164	165	160	151	10
10	148	160	160	162	160	155	10	12
8	10	140	150	152	150	10	11	10
9	12	10	10	10	10	9	10	10

	0.25	0.25	-33.75	-104	-104.75	-70.25	-0.5	
	0.25	-2.25	-77.25	-111.25	-145	-146.5	-74.25	
	37	70.25	-7.75	-7.75	-77.5	-150.5	-146.75	
	75.75	40	-2.5	-3.5	-4.5	-80.75	-147.25	
	77	7.5	1.5	0.75	-1.75	-7.25	-75	
	149.75	84.5	15.75	10.5	6	0.5	-0.75	
	142.5	146.5	116.5	113	74.5	1.75	1.5	

◉ 4-D Popular Filters (4): Matched Filter

Used for demodulation, similarity measurement, and pattern recognition

"Edge and corner detections" are special cases of pattern recognition.

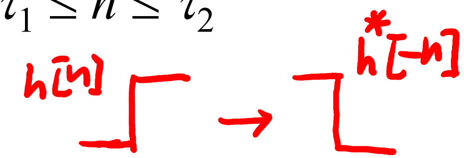
To detect a pattern $h[n]$, we use its time-reverse and conjugation form as the filter

(correlation)

$$y[n] = x[n] * h^*[-n] = \sum_{\tau=-\tau_1}^{-\tau_2} x[n-\tau] h^*[-\tau] = \sum_{\tau=\tau_1}^{\tau_2} x[n+\tau] h^*[\tau]$$

if $h[n] \neq 0$ for $\tau_1 \leq n \leq \tau_2$

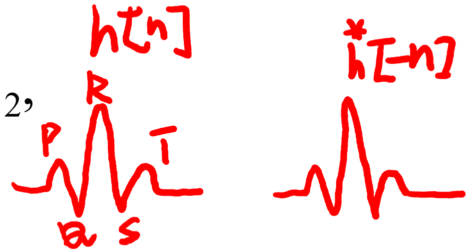
$x[n]$: input pattern, $h[n]$: the desired pattern



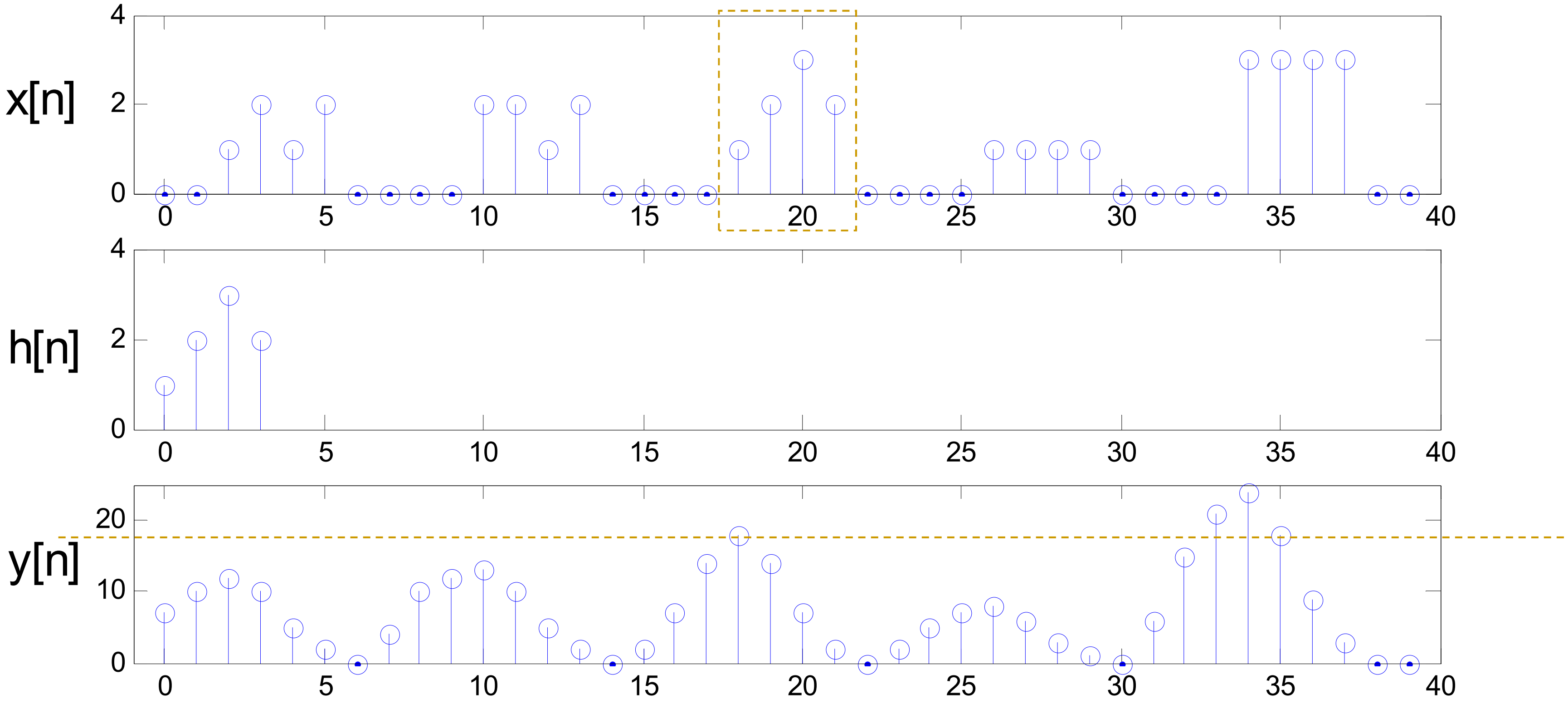
2-D form: $x * h = \sum_{\tau} x[n-\tau] h[\tau]$

$$y[m,n] = x[m,n] * h^*[-m,-n] = \sum_{\tau=\tau_1}^{\tau_2} \sum_{\rho=\rho_1}^{\rho_2} x[m+\tau, n+\rho] h^*[\tau,\rho]$$

if $h[m, n] \neq 0$ for $\tau_1 \leq m \leq \tau_2$, $\rho_1 \leq n \leq \rho_2$,



Example



$$y[n] = x[n] * \boxed{h^*[-n]}$$

The result of the convolution should be normalized!

• Normalization Form

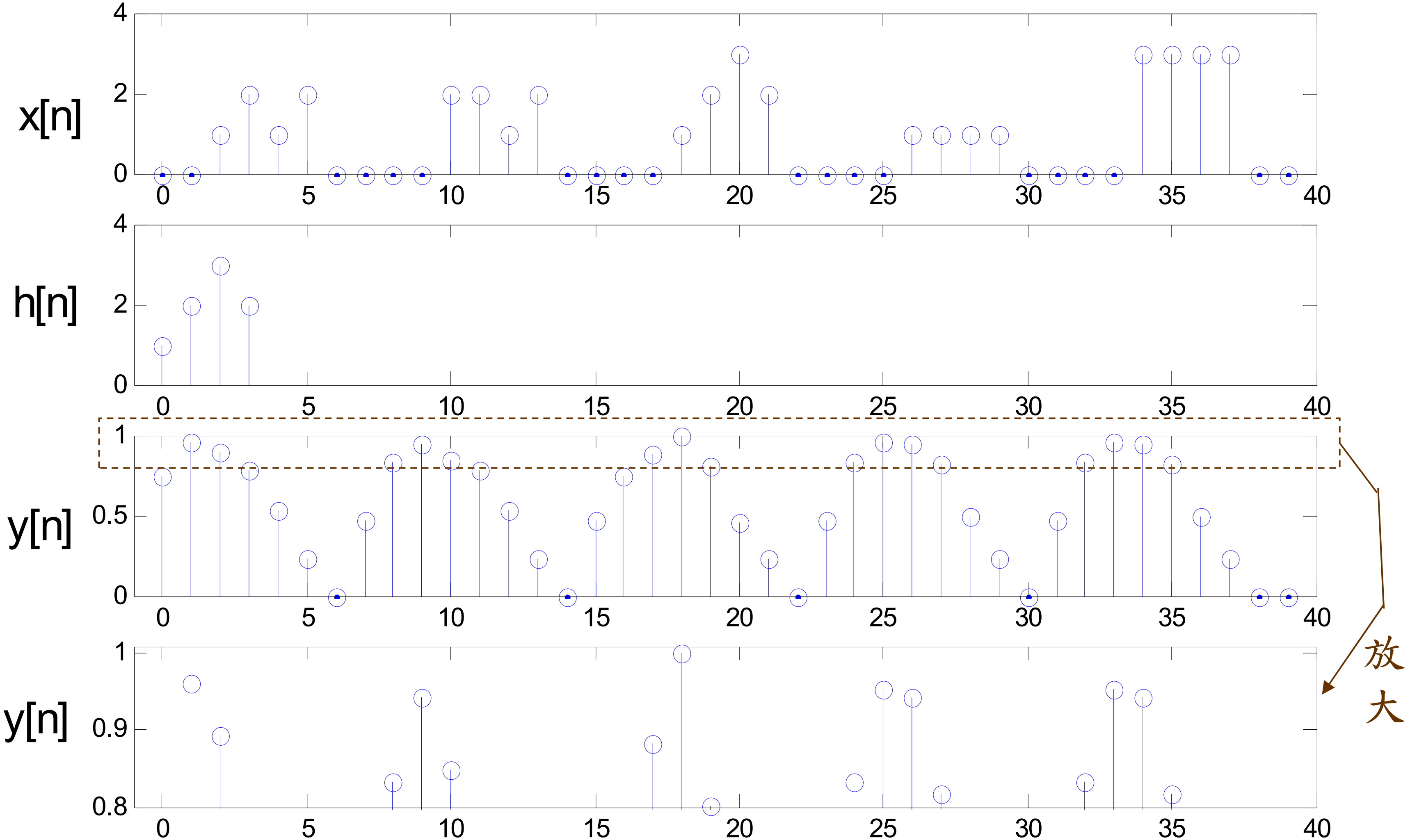
$$y[n] = \frac{\sum_{\tau=\tau_1}^{\tau_2} x[n+\tau]h^*[\tau]}{\sqrt{\sum_{s=n+\tau_1}^{n+\tau_2} |x[s]|^2 \sum_{s=\tau_1}^{\tau_2} |h[s]|^2}} \qquad \text{when} \quad \sum_{s=n+\tau_1}^{n+\tau_2} |x[s]|^2 \neq 0$$

$$y[n] = 0 \qquad \text{when} \quad \sum_{s=n+\tau_1}^{n+\tau_2} |x[s]|^2 = 0$$

2-D Case

$$y[m,n] = \frac{\sum_{\tau=\tau_1}^{\tau_2} \sum_{\rho=\rho_1}^{\rho_2} x[m+\tau, n+\rho]h^*[\tau,\rho]}{\sqrt{\sum_{s=m+\tau_1}^{m+\tau_2} \sum_{v=n+\rho_1}^{n+\rho_2} |x[s,v]|^2 \sum_{s=\tau_1}^{\tau_2} \sum_{v=\rho_1}^{\rho_2} |h[s,v]|^2}} \quad \text{when} \quad \sum_{s=m+\tau_1}^{m+\tau_2} \sum_{v=n+\rho_1}^{n+\rho_2} |x[s,v]|^2 \neq 0$$

$$y[m,n] = 0 \quad \text{when} \quad \sum_{s=m+\tau_1}^{m+\tau_2} \sum_{v=n+\rho_1}^{n+\rho_2} |x[s,v]|^2 = 0$$



x[n]
h[n]
y[n]
y[n]
4
2
0
1
0.5
0.9
0.8
0
5
10
15
20
25
30
35
40
放大

- Normalization and Offset Form

$$y[n]=\frac{\sum_{\tau=\tau_1}^{\tau_2}\left[x[n+\tau]-x_0[s]\right]h_1^*[\tau]}{\sqrt{\sum_{s=n+\tau_1}^{n+\tau_2}\left|x[s]-x_0[s]\right|^2\sum_{s=\tau_1}^{\tau_2}\left|h_1[s]\right|^2}} \qquad \text{when} \quad \sum_{s=n+\tau_1}^{n+\tau_2}\left|x[s]-x_0[s]\right|^2\neq 0$$

correlation coefficient

$$=\frac{E((X-\bar{X})(Y-\bar{Y}))}{\sigma_X\,\sigma_Y}$$

$\bar{X}=E(X)$, $\sigma_X=\sqrt{E((X-\bar{X})^2)}$

standard deviation of X

$$y[n]=0 \qquad \text{when} \quad \sum_{s=n+\tau_1}^{n+\tau_2}\left|x[s]-x_0[s]\right|^2=0$$

where

$$h_1[s]=h[s]-\frac{1}{\tau_2-\tau_1+1}\sum_{s=\tau_1}^{\tau_2}h[s]=h[s]-mean\left(h[s]\right)$$

$$x_0[s]=\frac{1}{\tau_2-\tau_1+1}\sum_{s=n+\tau_1}^{n+\tau_2}x[s] \qquad \text{(local mean)}$$

Comparison:

Correlation in Probability

$$corr(g,h)=\frac{\sigma_{g,h}}{\sigma_g\sigma_h}=\frac{\sum_n\left(g[n]-g_0\right)\left(h[n]-h_0\right)}{\sqrt{\sum_n\left(g[n]-g_0\right)^2\sum_n\left(h[n]-h_0\right)^2}}$$

$$g_0=\frac{1}{N}\sum_n g[n] \qquad\qquad h_0=\frac{1}{N}\sum_n h[n]$$

N: length of the sequences

- Normalization and Offset Form for the 2D Case

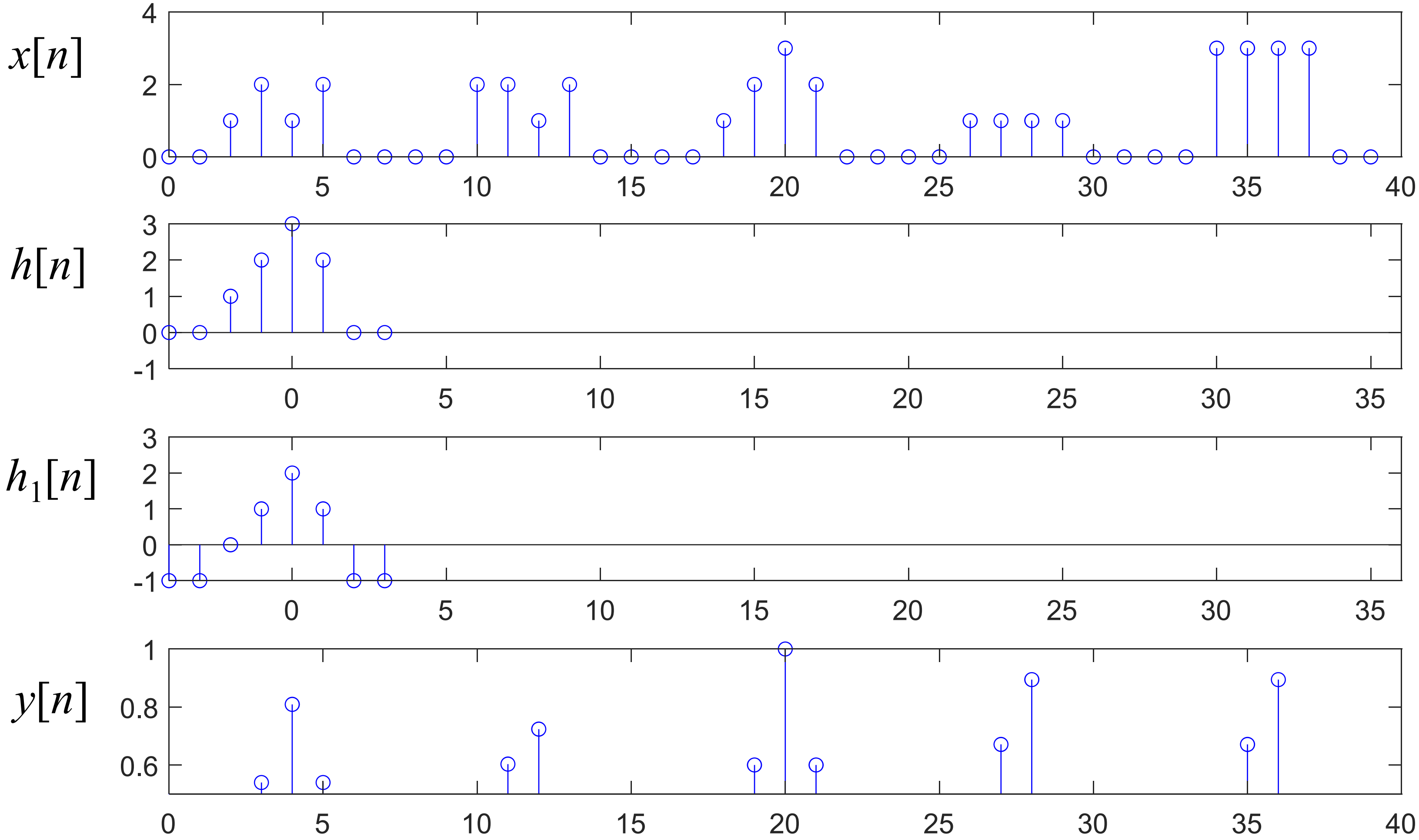
$$y[m,n]=\frac{\sum_{\tau=\tau_1}^{\tau_2}\sum_{\rho=\rho_1}^{\rho_2}x[m+\tau,n+\rho]h_1^*[\tau,\rho]}{\sqrt{\sum_{s=m+\tau_1}^{m+\tau_2}\sum_{v=n+\rho_1}^{n+\rho_2}\left|x[s,v]-x_0[s,v]\right|^2\sum_{s=\tau_1}^{\tau_2}\sum_{v=\rho_1}^{\rho_2}\left|h_1[s,v]\right|^2}}$$

$$\text{when}\quad \sum_{s=m+\tau_1}^{m+\tau_2}\sum_{v=n+\rho_1}^{n+\rho_2}\left|x[s,v]-x_0[s,v]\right|^2\neq 0$$

$$y[m,n]=0\quad \text{when}\quad \sum_{s=m+\tau_1}^{m+\tau_2}\sum_{v=n+\rho_1}^{n+\rho_2}\left|x[s,v]-x_0[s,v]\right|^2=0$$

where $h_1[s,v]=h[s,v]-\frac{1}{\tau_2-\tau_1+1}\frac{1}{\rho_2-\rho_1+1}\sum_{s=\tau_1}^{\tau_2}\sum_{v=\rho_1}^{\rho_2}h[s,v]=h[s,v]-mean\left(h[s,v]\right)$

$$x_0[s]=\frac{1}{\tau_2-\tau_1+1}\frac{1}{\rho_2-\rho_1+1}\sum_{s=m+\tau_1}^{m+\tau_2}\sum_{v=n+\rho_1}^{n+\rho_2}x[s,v]\quad \text{(local mean)}$$


x[n]
h[n]
h1[n]
y[n]
0
5
10
15
20
25
30
35
40

◉ 4-E Popular Filters (5): Particle Filter and Kalman Filter

Particle filter:

$$x[n+1] = f\left(x[n], m[n]\right)$$

$$x[n+1] = f\left(x[n], x[n-1], \cdots, x[n-K], m[n]\right)$$

$f(\ )$ is some mapping function and $m[n]$ is the noise
(prediction model) (prediction error)

The goal of the particle filter is not to remove the noise.

It is used for system modeling or prediction.

When (i) $f(\ )$ is a linear function and (ii) $m[n]$ is a Gaussian noise, it becomes the Kalman filter.

Example: $$x[n+1] = \sum_{\tau=0}^{K} c_\tau x[n-\tau] + m[n]$$

◎ 4-F Popular Filters (5): Wiener Filter

(Nobert Wiener 維納, AD 1949)

52 y

- No specific passband and stop band

It is related to random process.

- The filter is designed based on the statistics of signal and noise

Suppose that

(a) The cross-correlation between the original signal $x_s[n]$ and the received signal $y_s[n]$ (s = 1, 2, 3, ….) is $R_{xy}[t, \sigma]$,

$$R_{x,y}[n,\sigma] = E\left[x[n]y^*[\sigma]\right] = \frac{1}{N}\sum_{s=1}^{N} x_s[n]y_s^*[\sigma]$$

$(x_s[n] - x_0)(y_s^*[\sigma] - y_0^*)$

$x_0 = \text{mean}(x[n])$

$y_0 = \text{mean}(y[\sigma])$

$x_s[n], y_s[\sigma]$: the values of $x[n]$ and $y[\sigma]$ measured in the s^{th} trial

There are N times of trials.

(b) The auto-correlation of the received signal (denoted by $R_{yy}[n, \sigma]$).

$$R_{y,y}[n,\sigma] = E\left[y[n]y^*[\sigma]\right] = \frac{1}{N}\sum_{s=1}^{N} y_s[n]y_s^*[\sigma]$$

Then the transfer function of the optimal filter can be designed as

$$\boxed{\bigstar\ H_{opt}(F) = R_{X,Y}(F,F) / R_{Y,Y}(F,F)}$$

where
$$R_{X,Y}(F,F) = \sum_{\sigma}\sum_{n} e^{j2\pi F(\sigma-n)} R_{xy}[n,\sigma]$$

$$R_{Y,Y}(F,F) = \sum_{\sigma}\sum_{n} e^{j2\pi F(\sigma-n)} R_{yy}[n,\sigma]$$

(Proof):

To design the optimal filter $H_{opt}(F)$ that can well reconstruct $y[n, s]$ from $x[n, s]$, we want that

$$Y(F,s)H_{opt}(F) \cong X(F,s)$$

where $X(F, s)$ and $Y(F, s)$ are the discrete-time Fourier transform of $x[n, s]$ and $y[n, s]$, respectively:

$$X(F,s) = \sum_n e^{-j2\pi Fn} x(n,s) \qquad Y(F,s) = \sum_n e^{-j2\pi Fn} y(n,s)$$

We can define the error function as:

$$E = \frac{1}{N}\sum_{s=1}^{N}\int_{-1/2}^{1/2}\left|X(F,s) - Y(F,s)H(F)\right|^2 dF$$

$$= \frac{2}{N}\sum_{s=1}^{N}\int_{0}^{1/2}\left|X(F,s) - Y(F,s)H(F)\right|^2 dF$$

Suppose that

$$X(F,s) = X^*(-F,s)$$

$$Y(F,s) = Y^*(-F,s)$$

$$H(F) = H^*(-F)$$

To find the value of $H(F)$ at $F = F_1$, we can set that

$$\frac{\partial E}{\partial H(F_1)} = \frac{\partial}{\partial H(F_1)}\frac{2}{N}\sum_{s=1}^{N}\left|X(F_1,s) - Y(F_1,s)H(F_1)\right|^2 dF = 0$$

$$\frac{\partial}{\partial H(F_1)}\frac{2}{N}\sum_{s=1}^{N}\left|X(F_1,s)-Y(F_1,s)H(F_1)\right|^2 dF=0$$

$$\frac{\partial}{\partial H(F_1)}\sum_{s=1}^{N}\left(X(F_1,s)-Y(F_1,s)H(F_1)\right)\left(X^*(F_1,s)-Y^*(F_1,s)H^*(F_1)\right)=0$$

$$\sum_{s=1}^{N}\left(Y(F_1,s)X^*(F_1,s)-\left|Y(F_1,s)\right|^2 H^*(F_1)\right)=0$$

$$\sum_{s=1}^{N}\left(X(F_1,s)Y^*(F_1,s)-\left|Y(F_1,s)\right|^2 H(F_1)\right)=0$$

$$H(F_1)=\frac{\sum_{s=1}^{N}X(F_1,s)Y^*(F_1,s)}{\sum_{s=1}^{N}\left|Y(F_1,s)\right|^2}$$

In general, $H(F)=\dfrac{\sum_{s=1}^{N}X(F,s)Y^*(F,s)}{\sum_{s=1}^{N}\left|Y(F,s)\right|^2}$

$$H(F) = \frac{\sum_{s=1}^{N} X(F,s) Y^*(F,s)}{\sum_{s=1}^{N} \left| Y(F,s) \right|^2}$$

Since $\sum_{s=1}^{N} X(F,s) Y^*(F,s) = \sum_{s=1}^{N} \sum_{n} e^{-jFn} x[n,s] \overline{\sum_{\sigma} e^{-jF\sigma} y(\sigma,s)}$

$$= \sum_{n} \sum_{\sigma} e^{-jF(\sigma-n)} \sum_{s=1}^{N} x[n,s] y^*(\sigma,s) = N \sum_{n} \sum_{\sigma} e^{-jF(\sigma-n)} R_{xy}[n,\sigma]$$

$$= N R_{X,Y}[F,F]$$

Similarly, $\sum_{s=1}^{N} \left| Y(F,s) \right|^2 = N \sum_{n} \sum_{\sigma} e^{-jF(\sigma-n)} R_{yy}[n,\sigma] = N R_{Y,Y}[F,F]$

Therefore, $H(F) = \dfrac{R_{X,Y}[F,F]}{R_{Y,Y}[F,F]}$

References

[1] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, M.I.T. Press, Cambridge, Mass. , 1964.

[2] S. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, N.J., 2002.

[3] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine*, vol.14, no. 2, pp. 24-41, Mar. 1997

◉ 4-G Popular Filters (6): Equalizer

Used for compensation (such as the multiple path problem)

$$y[n] = x[n] * k[n]$$

$x[n]$: original signal, $y[n]$: received signal

$k[n]$: effect of the system

比較: noise removal problem

$y[n] = x[n] + noise[n]$

ex: $k[n] = e^{-\frac{\pi n^2}{\sigma^2}}$

$Y = XK$

Equalizer: $X = Y/K$

$$x[n] = y[n] * h[n] \qquad H(F) = \frac{1}{K(F)}$$

或者用 Z transform 表示 $\quad H(z) = \frac{1}{K(z)}$

$$y[n] = x[n] * k[n]$$ Equalizer: $$H(F) = \frac{1}{K(F)}$$

Problem: If the system is interfered by noise $m[n]$

$$y[n] = x[n] * k[n] + m[n]$$

$$Y(F) = X(F)K(F) + M(F)$$

$$H(F)Y(F) = X(F)H(F)K(F) + H(F)M(F)$$

$$= X(F) + \frac{M(F)}{K(F)}$$

If $K(F)$ is near to 0, the effect of the noise is magnified.

Combined with the concept of the Wiener filter, the equalizer is modified as:

$$H(F) = \frac{1}{\frac{1}{K^*(F)} \frac{E\left(|M(F)|^2\right)}{E\left(|X(F)|^2\right)} + K(F)}$$

E: mean

$$H(F) = \frac{1}{\frac{c}{K^*(F)} + K(F)}$$

$c = \frac{1}{SNR}$

c is large when the SNR is small
c is small when the SNR is large

- Equalizer for the Multiple Path Problem

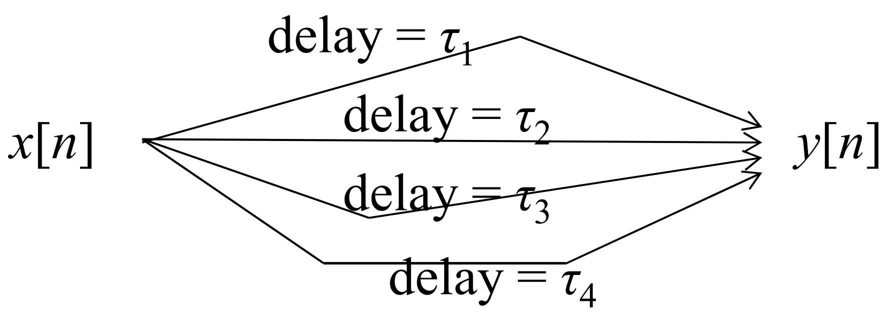
$$k[n]=\alpha_1\delta[n-\tau_1]+\alpha_2\delta[n-\tau_2]+\alpha_3\delta[n-\tau_3]+......$$

$$y[n]=x[n]*k[n]=\alpha_1 x[n-\tau_1]+\alpha_2 x[n-\tau_2]+\alpha_3 x[n-\tau_3]+......$$

$$Y[z]=(\alpha_1 z^{-\tau_1}+\alpha_2 z^{-\tau_2}+\alpha_3 z^{-\tau_3}+.......)X[z]$$

Usually α_k is related to τ_k, so it could be rewritten as $\alpha_k(\tau_k)$

Equalizer: $H(z)=\dfrac{1}{\alpha_1 z^{-\tau_1}+\alpha_2 z^{-\tau_2}+\alpha_3 z^{-\tau_3}+.......}$



- 缺點: (1) $H(z)$可能unstable

  (2) $H(z)$ is usually a dynamic response

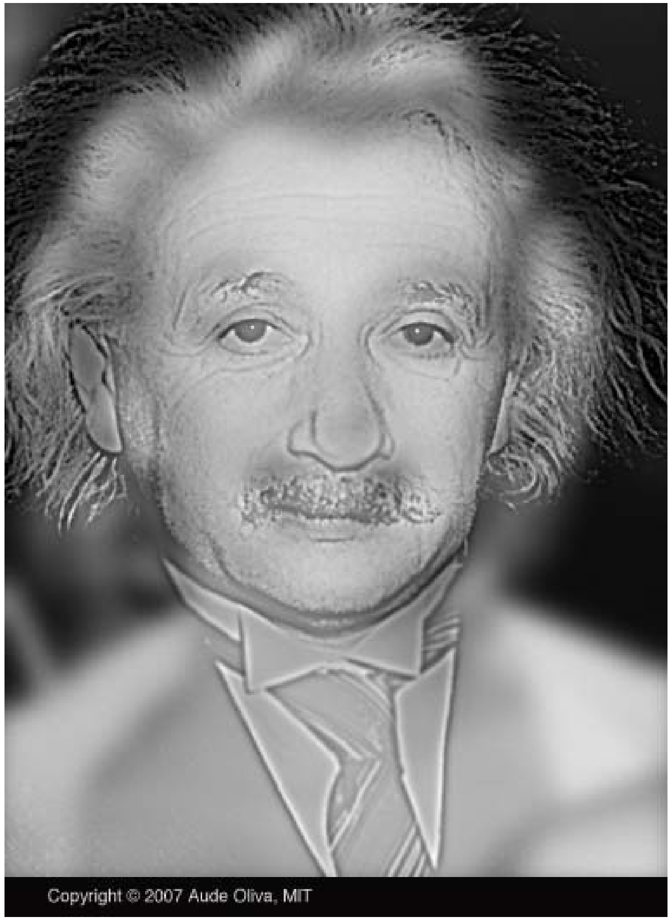
α_k, τ_k vary with the location

- 可以用 homomorphic signal processing 來取代 equalizer 處理 multiple path problem.

References

S. S. Haykin, *Communication Systems*, John Wiley, N.J., 2010

W. D. Chang, J. J. Ding, Y. Chen, C. W. Chang, and C. C. Chang, "Edge-membership based blurred image reconstruction algorithm," *APSIPA Annual Summit and Conference*, Hollywood, USA, Dec. 2012



high frequency

http://cvcl.mit.edu/hybrid_gallery/monroe_einstein.html

low frequency

附錄五 讀論文的方法 (個人心得)

為了做研究和工作的需要，同學們將來都要經常閱讀論文，甚至於，有的時候可能要一週要閱讀三篇以上的論文，而且大部分的論文說得都沒有像大學課本那麼有條理。用大學以前的讀書習慣，恐怕將難以應付。

要如何在短時間之內讀懂那麼多的論文，甚至於發現論文所提的方法可以改良的地方，是上了研究所之後必需學會的能力。

以下是幾點原則 (根據我個人的經驗)：

(A) 先判斷這篇論文是否應該被詳讀

(1) 越是核心，越是最早提出某個理論的論文，越是應該被詳讀

(2) 和自己目前研究密切相關的論文，當然有詳讀的必要

(3) Citation rate (引用次數) 較高的論文，可能也比較重要 (雖然不完全相關)。

至於比較支節的論文，大略讀過即可

(B) 自己動手算

對於該「詳讀」的論文，可以自己動手來計算當中的幾個重要公式。

不是每篇論文都對論文中的理論和公式的來源有清楚的說明。在這個時候，還不如自己拿起筆來，親手證明論文當中的公式和理論。

自己動手算，不只能幫助自己了解論文當中的理論，而且，有時還可以「意外」的發現論文當中的理論可以進一步改良的地方，進而寫出新的論文出來。

(C) 讀過論文之後，問自己一些問題

(1) 這篇論文所提的概念 (Concepts) 是什麼？

(2) 方法的優點何在 (Advantages)？

(3) 可能的應用 (Applications) 在何處？

若能回答這三個問題，表現你大致讀通了這篇論文

若回答不出來，可能要再把論文當中遺漏的地方，再好好看 一看

(D) 進一步的分析

如果你不以讀懂一篇論文為滿足，想要進一步的發明創造之外，可以再問自己幾個問題

(1) Analysis for Advantages: 是什麼原因，造成這個方法有這樣的優點？
類似的概念，是否可以延伸、用在其他地方？

(2) Analysis for Disadvantages: 這方法有什麼問題？
是什麼原因，造成這些問題？
有什麼方法，可以改良這些問題？

(3) Innovations: 綜合以上的分析，再加上個人的靈感，想想這篇論文是否有可以再進一步發明創新的地方？

(E) 註解

我經常看過一篇論文之後，會寫上幾行的文字，來描述這篇論文要點，以及在這個領域當中所扮演的角色。一方面有助於釐清概念，一方面也可以避免日後還要花時間來回憶這篇論文的內容是什麼

(F) 做個整理

可以將多篇論文所提的許多種方法，做一個有系統的整理和比較。

總共有多少種方法被提出來處理這個問題？這些方法的優缺點和適用的地方是什麼？它們之間是否可以歸納成幾大類？這些方法的相似和相異之處是什麼？

有時，把各種不同的方法做個綜合，拮取各方法的優點，將有助出創造出效能更好的新方法