◎ 2-J FIR Filter in MSE Sense with Weight Functions

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

可對照 pages 49~51

If the transition band is from F_0 to F_1

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

$F=f/f_s$

$$MSE=\int_{-1/2}^{-F_1} w(F)\left|R(F)-H_d(F)\right|^2 dF+\int_{-F_0}^{F_0} w(F)\left|R(F)-H_d(F)\right|^2 dF+\int_{F_1}^{1/2} w(F)\left|R(F)-H_d(F)\right|^2 dF$$

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

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

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

$$n=0\sim k$$

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

(not orthogonal)

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

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

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

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

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

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

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

with transition band $\int_{-1/2}^{1/2} \xRightarrow{\text{modified}} \int_{-1/2}^{-F_1} + \int_{-F_0}^{F_0} + \int_{F_1}^{1/2}$

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

$$MSE = ?$$

$$B[n,\tau] = ?$$

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

  $\underline{h[n]=h[N-1-n]}$ (even symmetric) and $\underline{N \text{ is odd.}}$

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

  $\underline{h[n]=h[N-1-n]}$ (even symmetric) and $\underline{N \text{ is even.}}$

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

  $\underline{h[n]=-h[N-1-n]}$ (odd symmetric) and $\underline{N \text{ is odd.}}$

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

  $\underline{h[n]=-h[N-1-n]}$ (odd symmetric) and $\underline{N \text{ is even.}}$

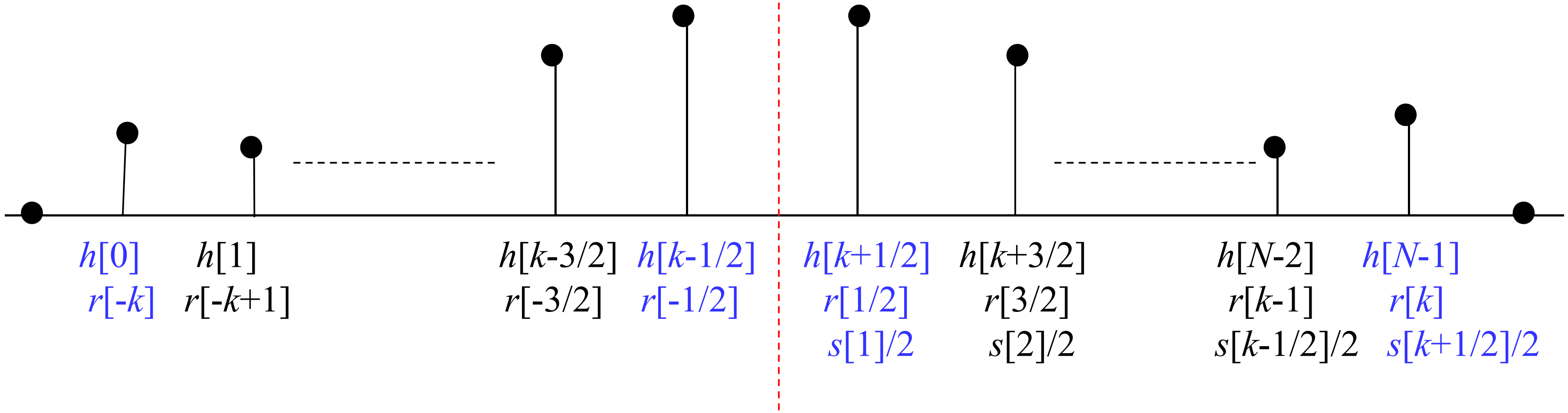
$k=(N-1)/2$

- 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)

$h[k + \frac{2n-1}{2}] = h[k - \frac{2n-1}{2}]$ $n = 1, 2, 3 \cdots$ [illegible]

(比較 page 45)



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

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

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

$n=1/2,\ 3/2\ \cdots,k$

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

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

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

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

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

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

$\cos(\alpha-\beta)+\cos(\alpha+\beta)=2\cos\alpha\cos\beta$

$\alpha=2\pi nF,\ \beta=\pi F$

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

when $n=k+1/2$

$\cos(2\pi(n-\frac{1}{2})F)=\cos(2\pi kF)$

由於 n 和 $n+1$ 兩項相加可得

$$\cos\left(2\pi(n-1/2)F\right)+\cos\left(2\pi(n+1/2)F\right)=2\cos\left(\pi F\right)\cos\left(2\pi nF\right)$$

所以可以「判斷」$R(F)$ 能被改寫成

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

求 $s_1[n]$ 和 $s[n]$ 之間的關係

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

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

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

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

n=0 n=1~k₁ n=1~k₁ n=k₁+1

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

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

$\frac{1}{2}(s_1[1]+s_1[0])\cos(\pi F)$ n=1 n=2~k₁

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

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

(令 $k_1+1/2=k$) ⇒ $k_1=k-1/2$

1

比較係數可得

$$\begin{cases} s[1]=s_1[0]+\frac{1}{2}s_1[1] \\ s[n]=\frac{1}{2}\left(s_1[n]+s_1[n-1]\right) & \text{for } n=2,\ 3,\ \ldots,\ k-1/2 \\ s[k+1/2]=\frac{1}{2}s_1[k-1/2] \end{cases}$$

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

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

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

注意 $s_1[n]$ 和 $s[n]$ 之間的關係即可

sin(α+β) − sin(α−β) = 2 sinα cosβ
α = 2πF, β = 2πnF

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

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

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

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

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

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

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

令 $k_1=k-1$, 比較係數可得

$$\begin{cases} s[1]=s_1[0]-\frac{1}{2}s_1[2] \\ s[n]=\frac{1}{2}s_1[n-1]-\frac{1}{2}s_1[n+1] \qquad \text{for } n=2,\ 3,\ \ldots,\ k-2 \\ s[k-1]=\frac{1}{2}s_1[k-2] \\ s[k]=\frac{1}{2}s_1[k-1] \end{cases}$$

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

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

~ (n^{th} term) and the $(n+1)^{th}$ term

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

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

$$-\sin(\alpha-\beta) + \sin(\alpha+\beta)$$
$$= 2\sin\beta\cos\alpha$$

$\beta = \pi F$, $\alpha = 2\pi nF$

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

causal

$r_1[n]$: output of Step 2
$r[n]$: output of Step3

Proof:

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

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

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

when $F=m/N$

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

$FN=\frac{m}{N}N$
$=m$

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

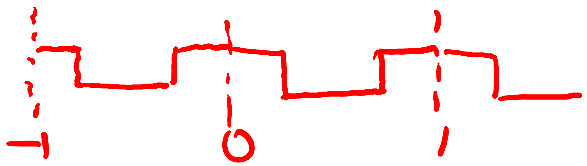
又由於 $r_1[n]$ 是 $H_d(m/N)$ 的 inverse discrete Fourier transform (IDFT)

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

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

Example: $N = 17$

lowpass

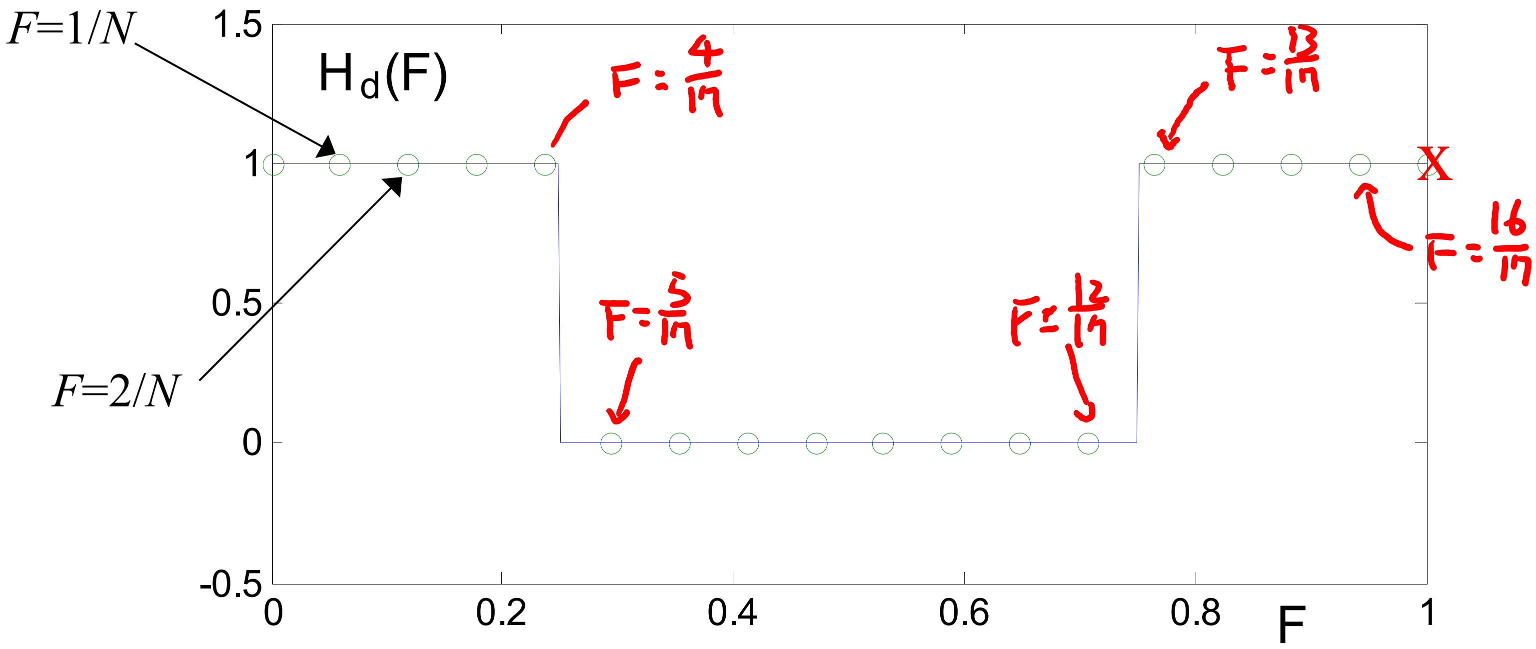


$$H_d(F) = 1 \text{ for } -0.25 < F < 0.25,$$

$$H_d(F) = 0 \text{ for } -0.5 < F < -0.25, \; 0.25 < F < 0.5$$

(Step 1) m=0 1 2 3 4 5 12 13 14 15 16

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



(Step 2) inverse discrete Fourier transform

$r_1[n]$ = ifft([1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1])

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

(Step 3)

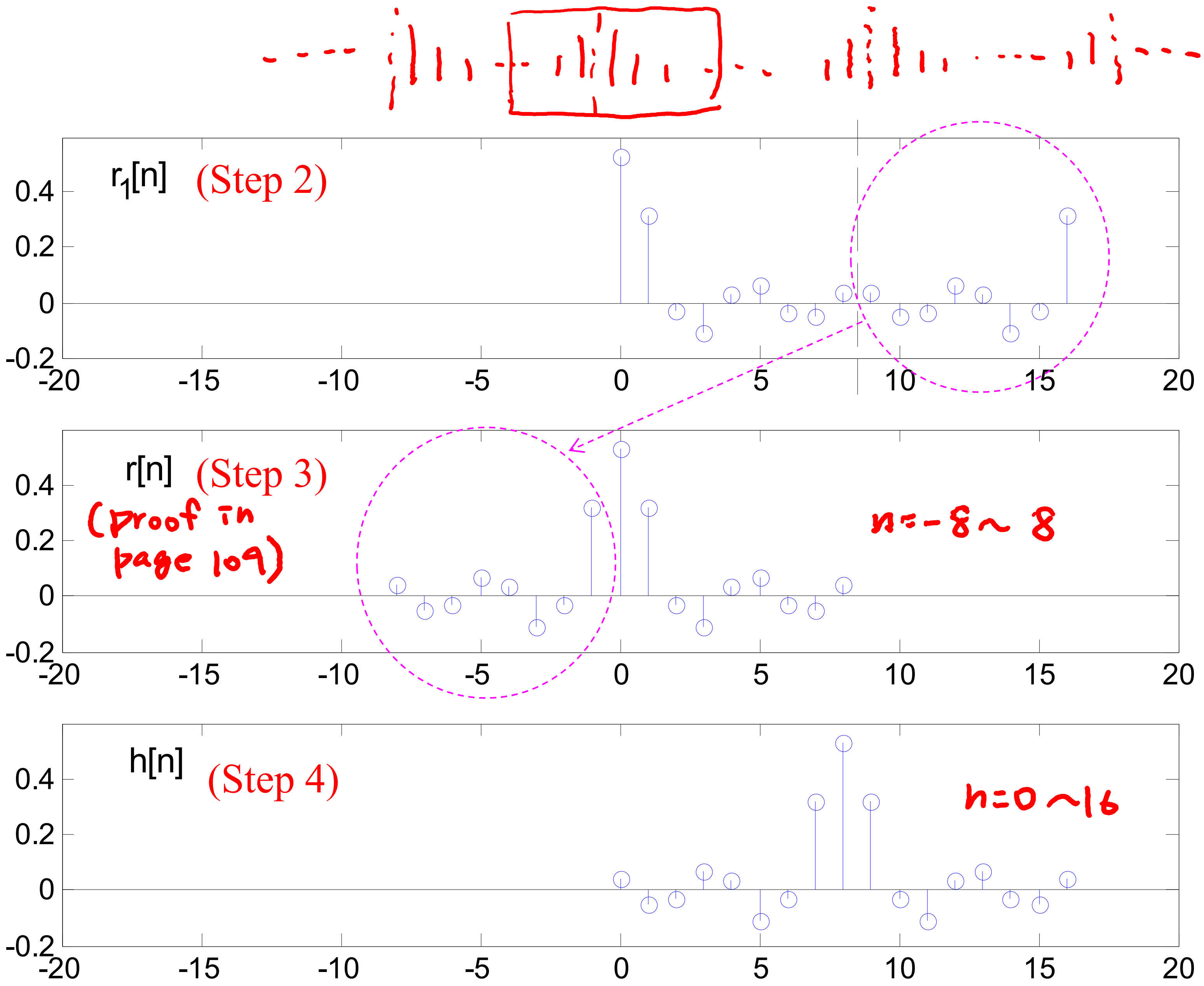
$r[n]$ = [0.040 -0.049 -0.035 0.066 0.032 -0.107 -0.030 0.319 0.529
0.319 -0.030 -0.107 0.032 0.066 -0.035 -0.049 0.040] $n = -8 \sim 8$

(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 -0.049 -0.035 0.066 0.032 -0.107 -0.030 0.319 0.529
0.319 -0.030 -0.107 0.032 0.066 -0.035 -0.049 0.040] $n = 0 \sim 16$


r1[n] (Step 2)
r[n] (Step 3)
(proof in page 109)
n=-8~8
h[n] (Step 4)
n=0~16

Frequency Response in terms of $R(F)$

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

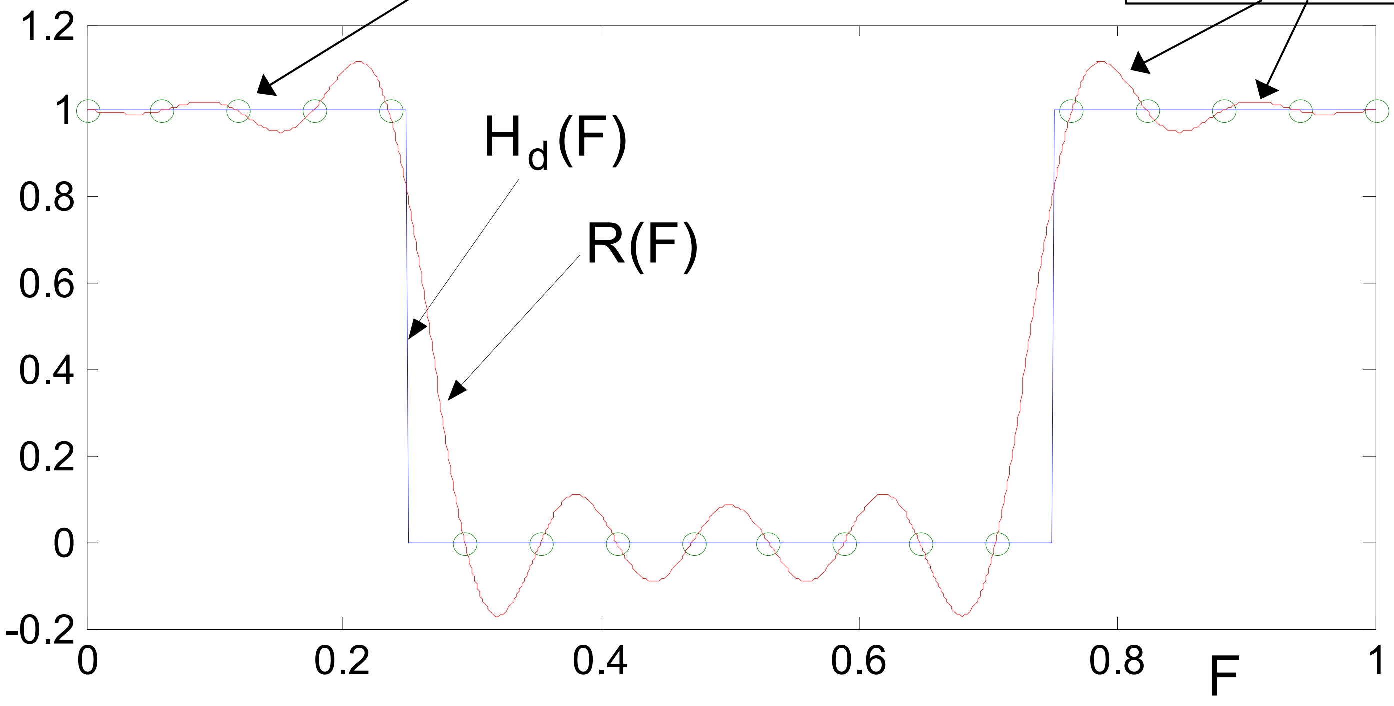
Step 3

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

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

$R(F)$在sample frequency 等於$H_d(F)$

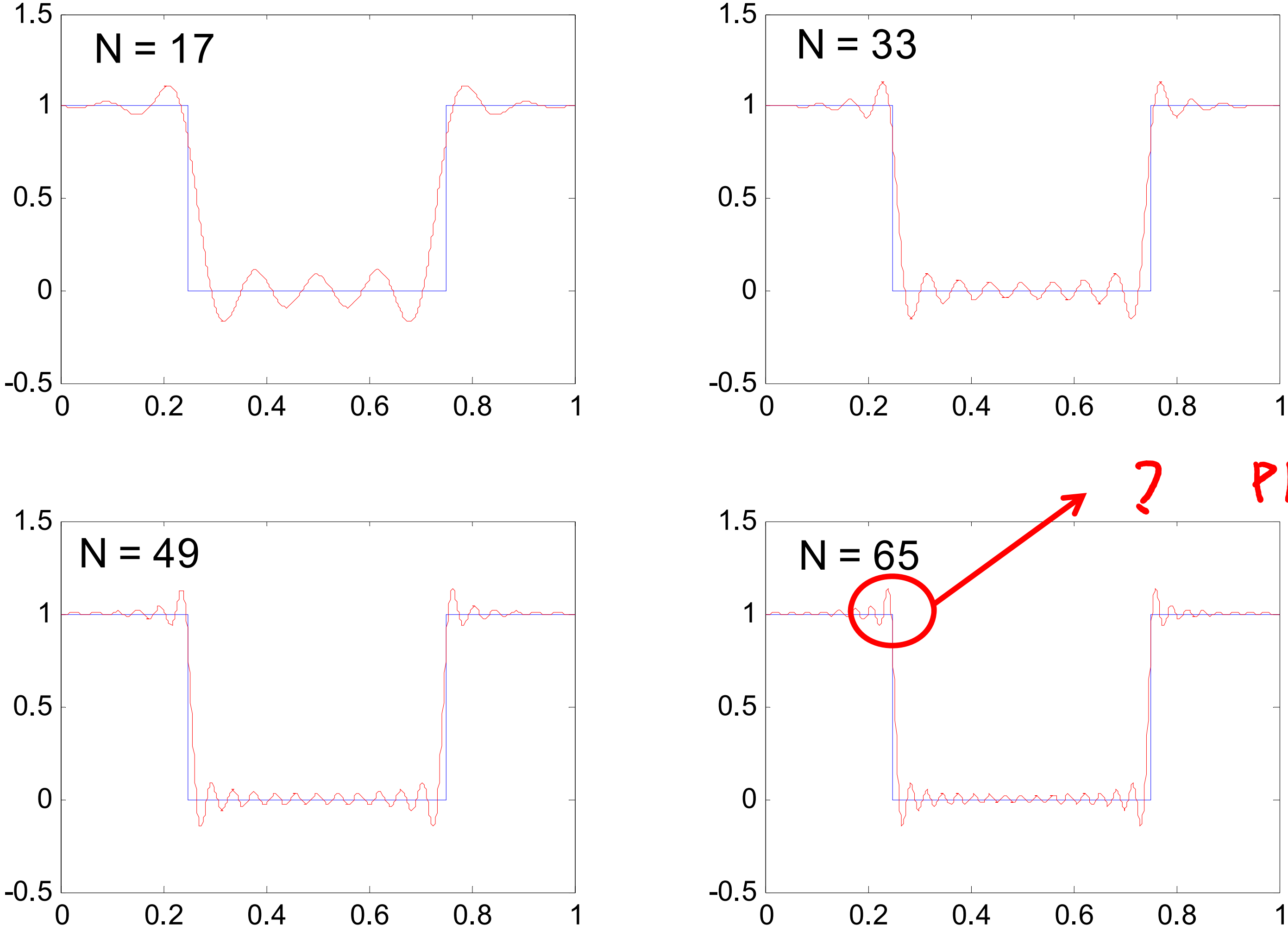
Error 非 equal-ripple



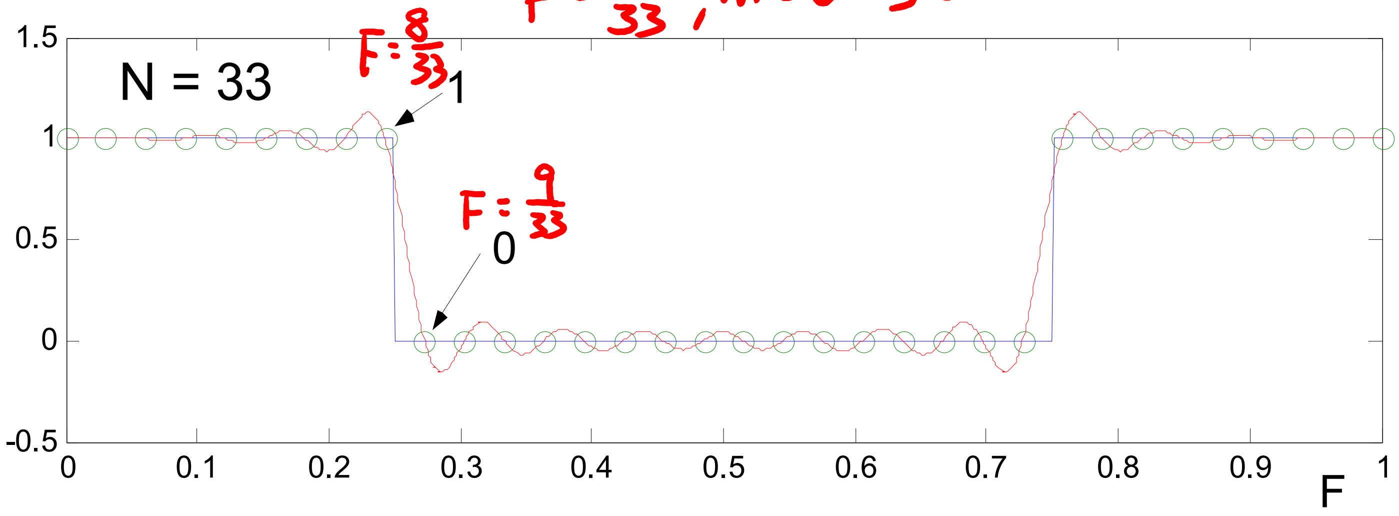
- The approximation error tends to be highest around the transition band and smaller in the passband and stopband regions.

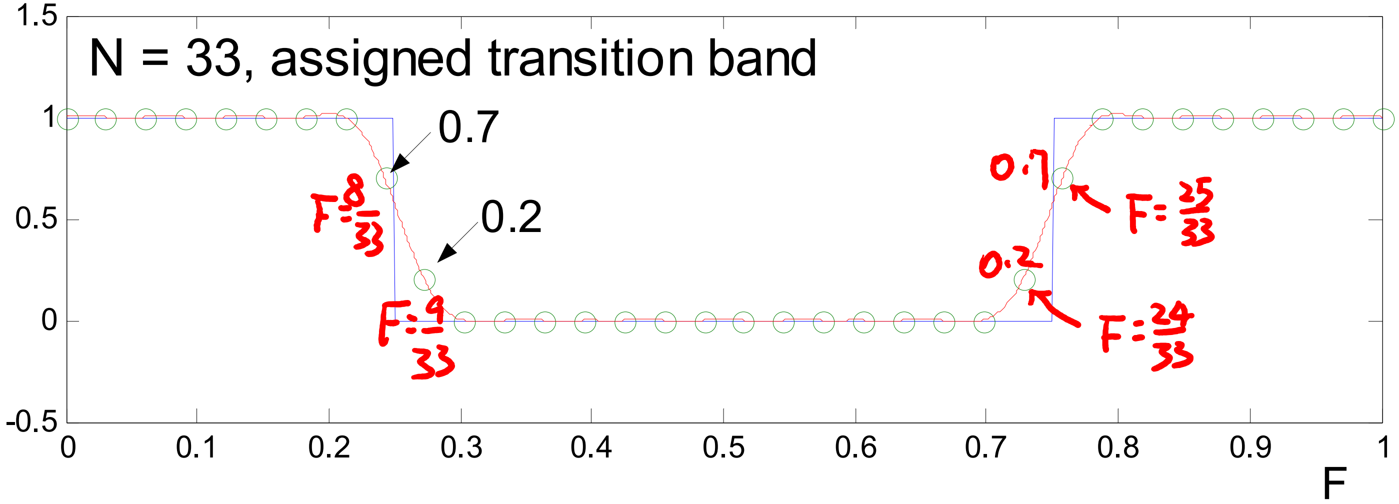
Error is larger at the edge

? Phenomenon



$F = \frac{m}{33}$, $m: 0 \sim 32$





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-M 三種 FIR Digital Filter 設計方法的比較

- 以設計方法而論

  MSE : middle (integral or matrix inverse)

  Minimax : complicated (recursive)

  frequency sampling : simplest (IDFT)

- 以方法的限制而論

  MSE : less constraint

  Minimax : (i) transition band is necessary
  (ii) suitable for pass-stop band filter.

  frequency sampling : hard to apply the weight function

- 以效果而論

  MSE : optimal in mean square error

  Minimax : optimal in maximal error

  frequency sampling : not optimal

The 4th Method for the FIR Filter Design?

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

$H[m] = 1$ for passband

$H[m] = 0$ for stopband

ex: lowpass filter

passband : $|f| \le 10000$

$f_s = 40{,}000$

$N = 10 f_s = 400{,}000$

$m_0 = 10000 \frac{N}{f_s}$ (page 75)

$= 100{,}000$

$H(m) = 1$ for $0 \le m \le 100{,}000$

or $? \le m \le 399{,}999$

$H(m) = 0$ otherwise

When $f = -10000$

$m = ?$

large computation loading

$O(N \log N)$

Q: Why do we not apply the method?

◉ 2-N 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 $O(N)$

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

(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**.

$\Downarrow$

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

forward filter $H(z)$

inverse filter $\frac{1}{H(z)}$

$x[n-n_0] \xrightarrow{DTFT} e^{-j2\pi F n_0} X(F)$

delay ↔ phase

maximal phase filters: all poles and zeros are outside the unit circle

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

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

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

$p_1, p_2, p_3, \ldots.., p_S$: **poles** $z_1, z_2, z_3, \ldots.., z_R$: **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., $|p_s| \le 1$ and $|z_r| \le 1$

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 $R–S$.

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

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

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

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

Z^{-1} : inverse Z transform

for $n = 1, 2, 3, 4, \ldots..$

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

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

$|p_s| = 1$ $h_s[n]$ is finite
but its energy is infinite

Therefore,

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

where

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

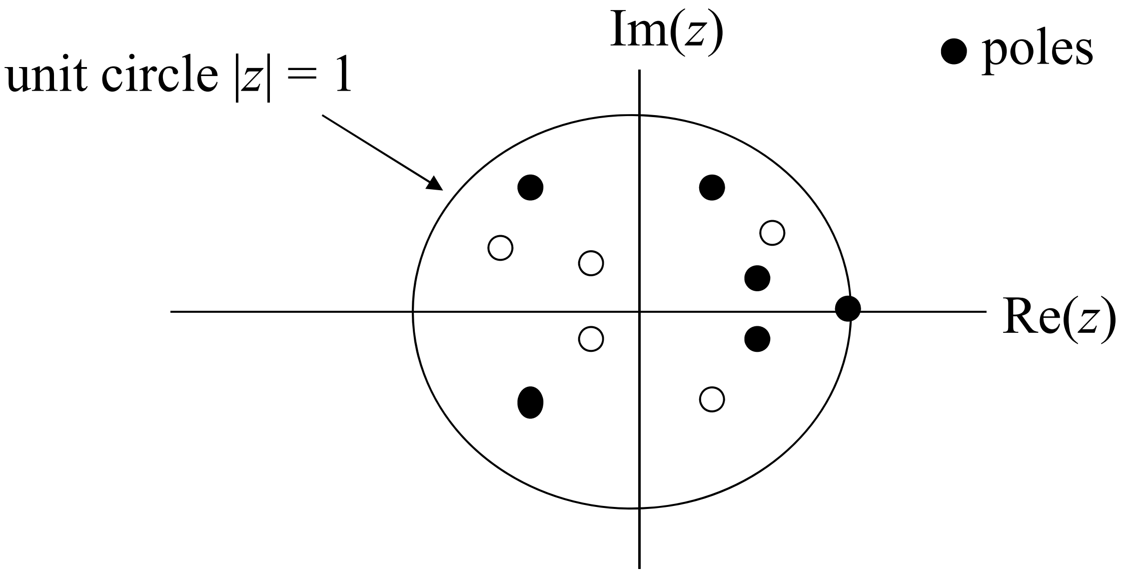
$$h_s[n] = A_s p_s^n u[n] \qquad \text{for } s = 1,\ 2,\ \ldots\ldots,\ S$$

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, $|z_2| > 1$

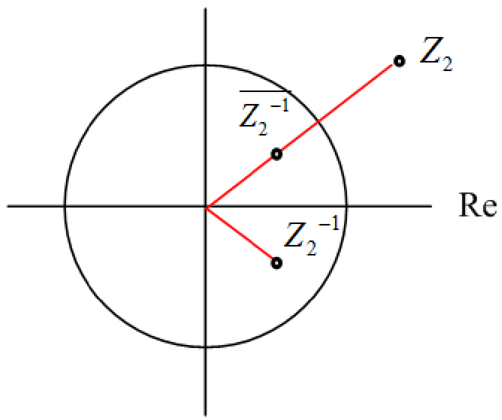
$$H_1(z) = C\frac{(z-z_1)(z-z_2)(z-z_3)\cdots\cdots(z-z_R)}{(z-p_1)(z-p_2)(z-p_3)\cdots\cdots(z-p_S)} \times \boxed{z_2\frac{z-\overline{(z_2^{-1})}}{z-z_2}}$$

allpass filter

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

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 28), then $H(z)$ and $H_1(z)$ only differ in phase,

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

(proof):

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

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

when $z = e^{j2\pi F}$, $z^{-1} = \bar{z}$
(單位圓上) $e^{-j2\pi F}$

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

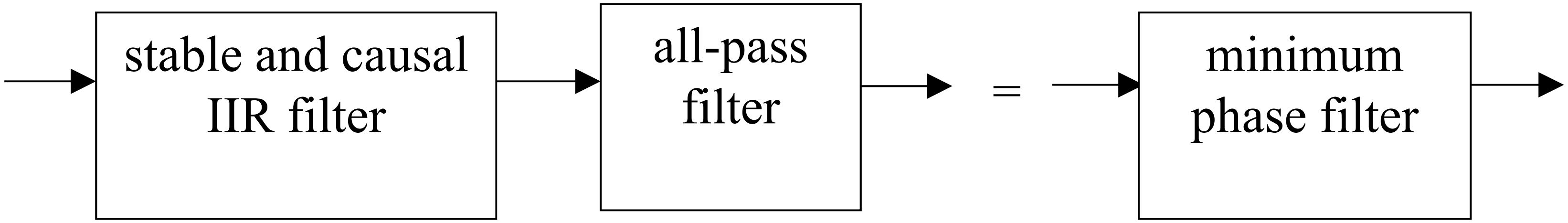
$$z_2 \frac{z - \overline{(z_2^{-1})}}{z - z_2} \quad \textbf{is an all-pass filter}$$

- 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:

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

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

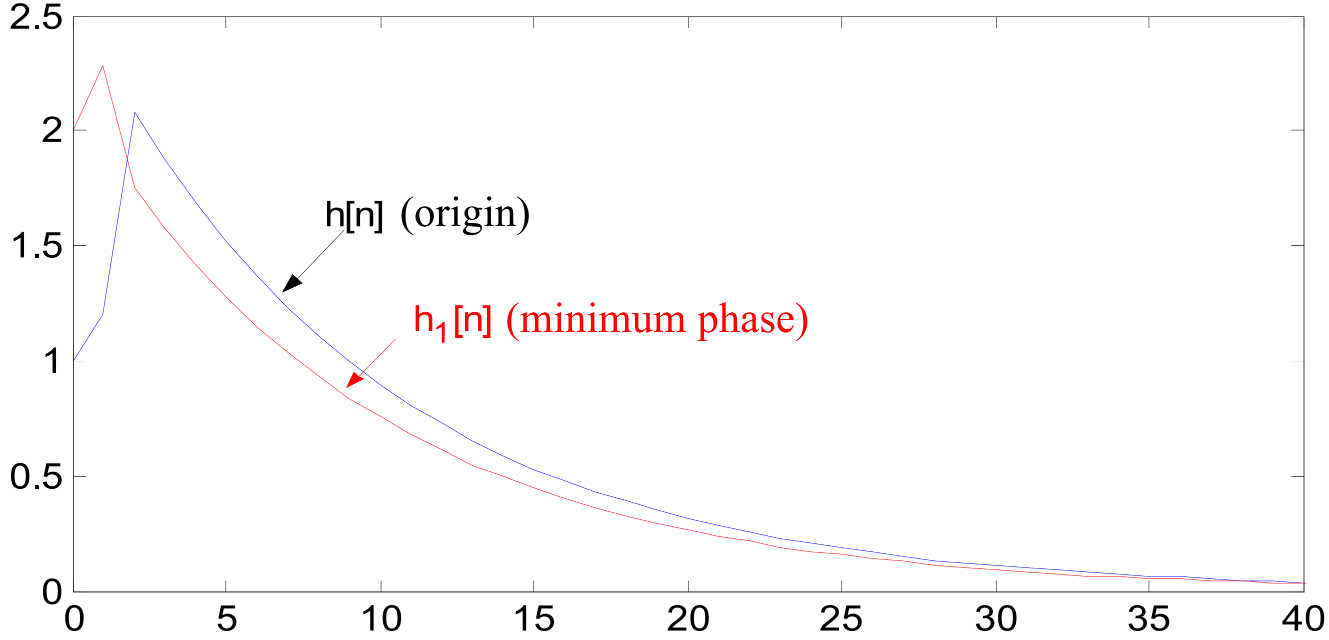
$$H_1(z)=(1.6+1.2j)\frac{(z+0.6)[z-(0.4+0.3j)]}{z-0.9}$$

$z_2=1.6+1.2j$

$\overline{z_2^{-1}}=\overline{\frac{1}{1.6+1.2j}}=\overline{0.4-0.3j}=0.4+0.3j$

$\frac{1}{(1.6+1.2j)}=\frac{1.6-1.2j}{(1.6+1.2j)(1.6-1.2j)}=\frac{1.6-1.2j}{4}$

$h[n]$, $h_1[n]$ 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：

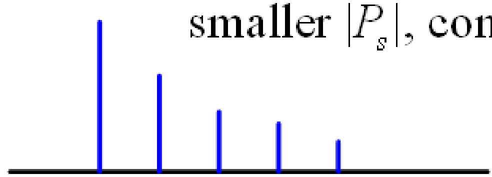
The energy concentrating on the region near to n = 0.

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

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

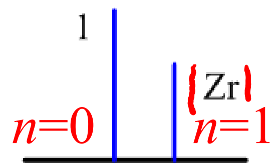
$Z^{-1}\left[\frac{1}{1-p_sz^{-1}}\right]=a_s[n]$ $a_s[n]=0$ when $n<0$ $a_s[n]=p_s^n$ when $n\geq 0$

smaller $|P_s|$, converge faster



$X(z)=\sum_n x[n]z^{-n}$

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

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

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 $|z_n| = 1$?

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) 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

(6) 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

(7) 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

(8) 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」

(9) 找尋電子書

「台大圖書館首頁」→「電子書」或「免費電子書」

(10) 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

(11) 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

(12) 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)

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

(14) 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.

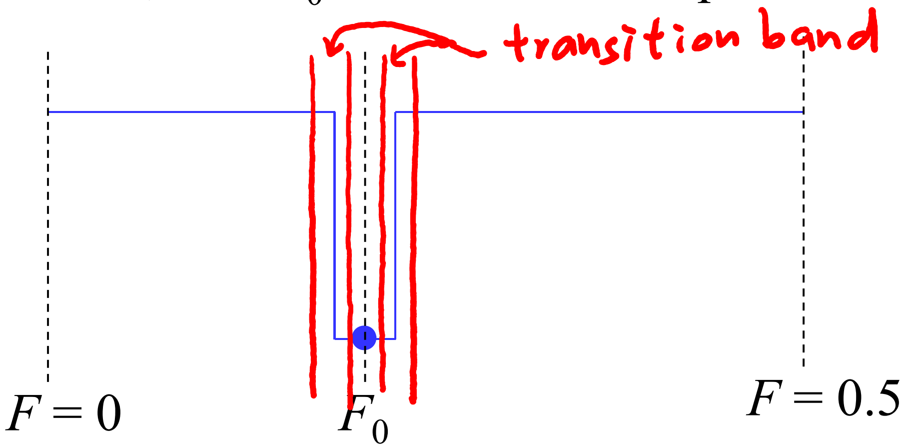
A. Jeffrey, *Handbook of Mathematical Formulas and Integrals*, Academic Press, San Diego, 2000.

4. Some Popular Filters

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

highpass bandpass lowpass allpass bandstop

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



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:

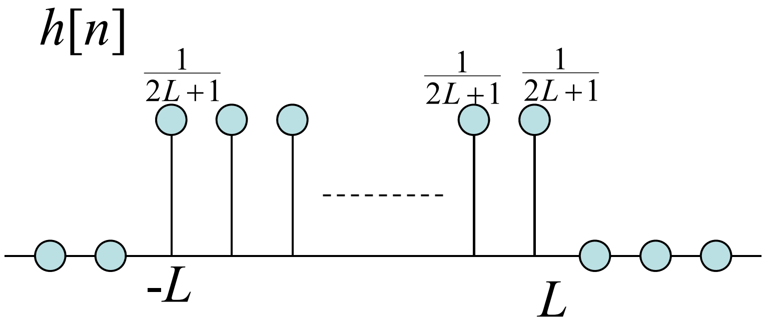
find the average $$y[n] = \frac{1}{2L+1}\sum_{\tau=n-L}^{n+L} x[\tau]$$

sinc $\frac{\sin \pi f}{\pi f}$

可改寫成 $1 \cdot u[n+L] - 1 \cdot u[n-L-1]$

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

$h[n]$ 如右圖



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

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

$$\therefore y[n] = y[n-1] - x_1[n-1-L] + x_1[n+L] \qquad x_1[n] = \frac{1}{2L+1} x[n]$$

for each n, MUL = ? ADD = ?

一般型態的 smoother

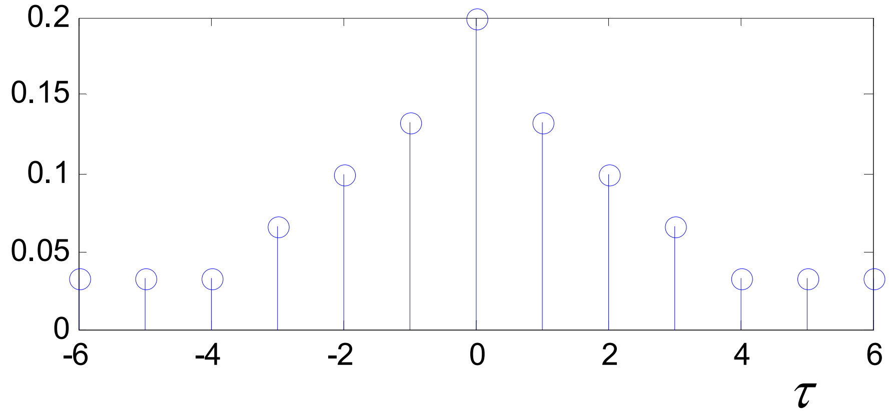
$$y[n] = x[n] * h[n] = \sum_{\tau} x[n-\tau] h[\tau]$$

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

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

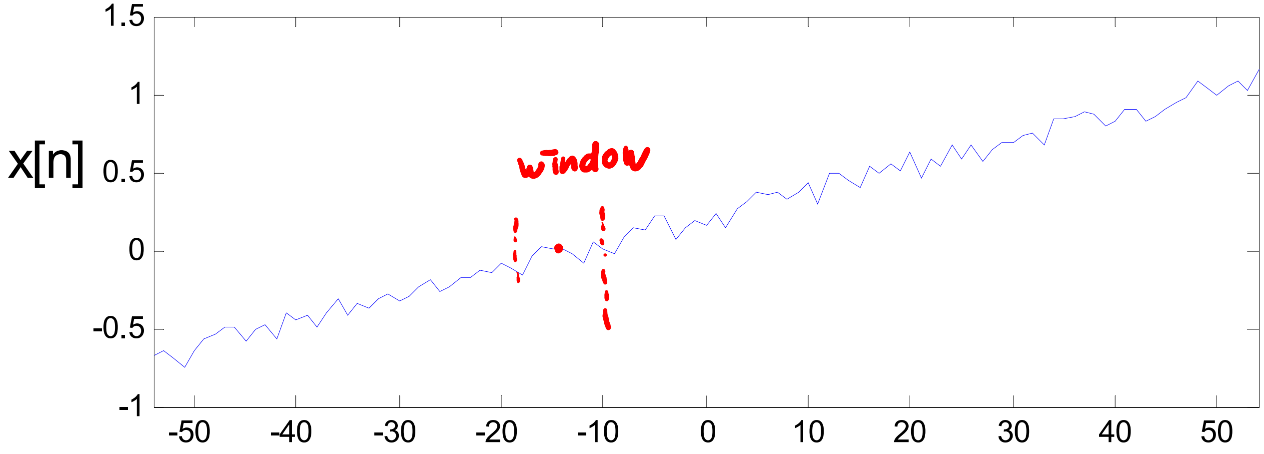
(3) $h[n] \ge 0$ for all n

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

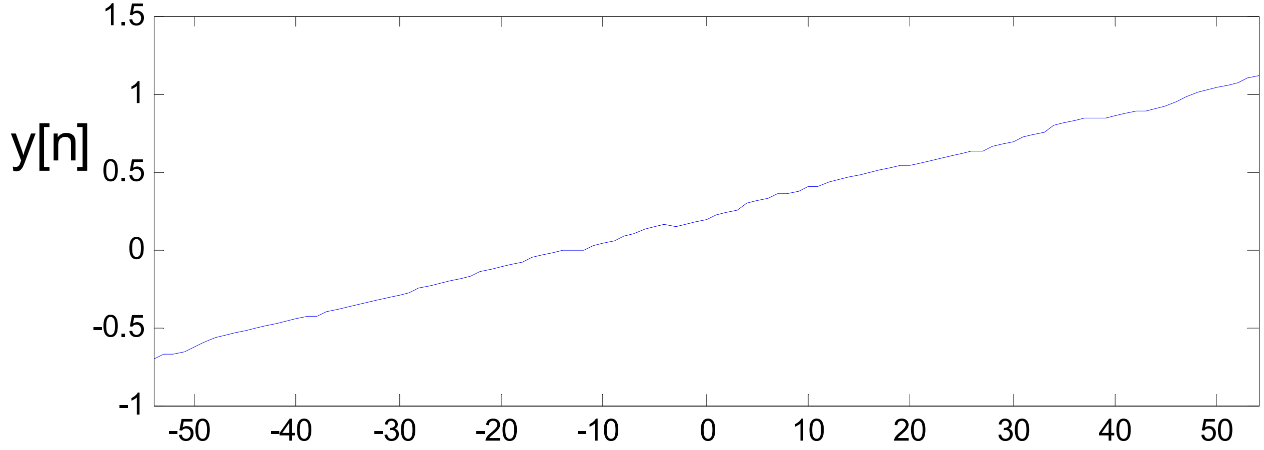


任何能量隨著 $|n|$ 遞減的 even function，都可以當成 smoother filter

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？

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

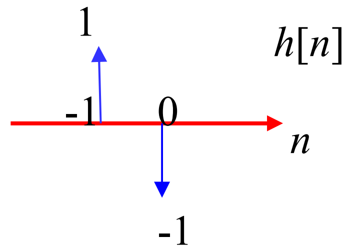
(a) Differentiation $H(f) = j2\pi f$ when $-f_s/2 < f < f_s/2$,

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

$$x_1[n] = \sum_m x[n-m]\,h[m]$$

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

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



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

$h[n] = 0$ otherwise

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

$$h[-\tfrac{1}{2}+n] = -h[-\tfrac{1}{2}+n]$$

$$h_1[n] = h[n-\tfrac{1}{2}]$$

$$h_1[n] = -h_1[-n]$$

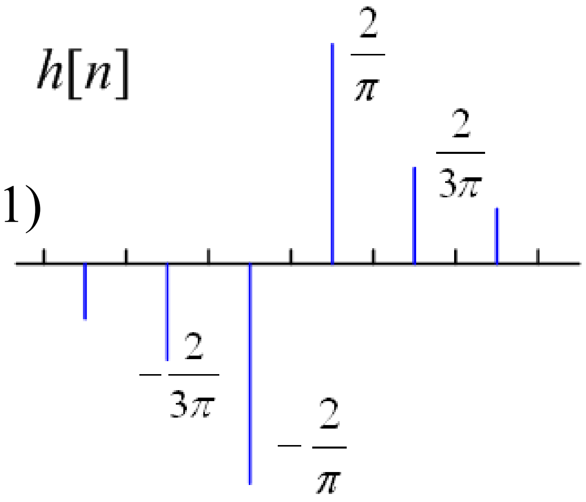
These two filters are equivalent only at low frequencies

(C) Discrete Hilbert Transform

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

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

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



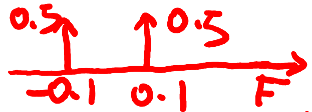
IIR $\quad h[n] = \dfrac{2}{\pi n}$ when n is odd, $\quad h[n] = 0$ otherwise

single-sided band

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

Ex: $x[n] = \cos(0.2\pi n)$

$X(F) = \frac{1}{2}\delta(F-0.1) + \frac{1}{2}\delta(F+0.1)$



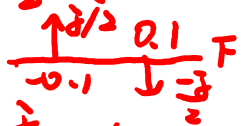
Analytic function: $\quad x_a[n] = x[n] + jx_H[n]$

$X_H(F) = H(F)X(F)$ where $\quad x_H[n] = x[n] * h[n]$

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

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

$X_H(F) = \frac{-j}{2}\delta(F-0.1) + \frac{j}{2}\delta(F+0.1)$



$$X_a(F) = \begin{cases} 2X(F) & 0 < F < 0.5 \\ 0 & -0.5 < F < 0 \\ X(F) & F = 0, 0.5 \end{cases}$$

$x_H[n] = \frac{-j}{2}\exp(j(0.2\pi n)) + \frac{j}{2}\exp(-j0.2\pi n)$
$= \sin(0.2\pi n)$

$x_a[n] = \cos(0.2\pi n) + j\sin(0.2\pi n)$
$= \exp(j0.2\pi n)$

(D) Edge Detection ← 近似 high-pass filter

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

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

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

Difference 和 discrete Hilbert transform 都可用作 edge detection

(1) 任何<u>能量隨著 $|n|$ 遞減</u>的 <u>odd function</u>，都可以當成 edge detection filter

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

ex: $h[n] = \exp(-kn)$ $0 < n \le L$

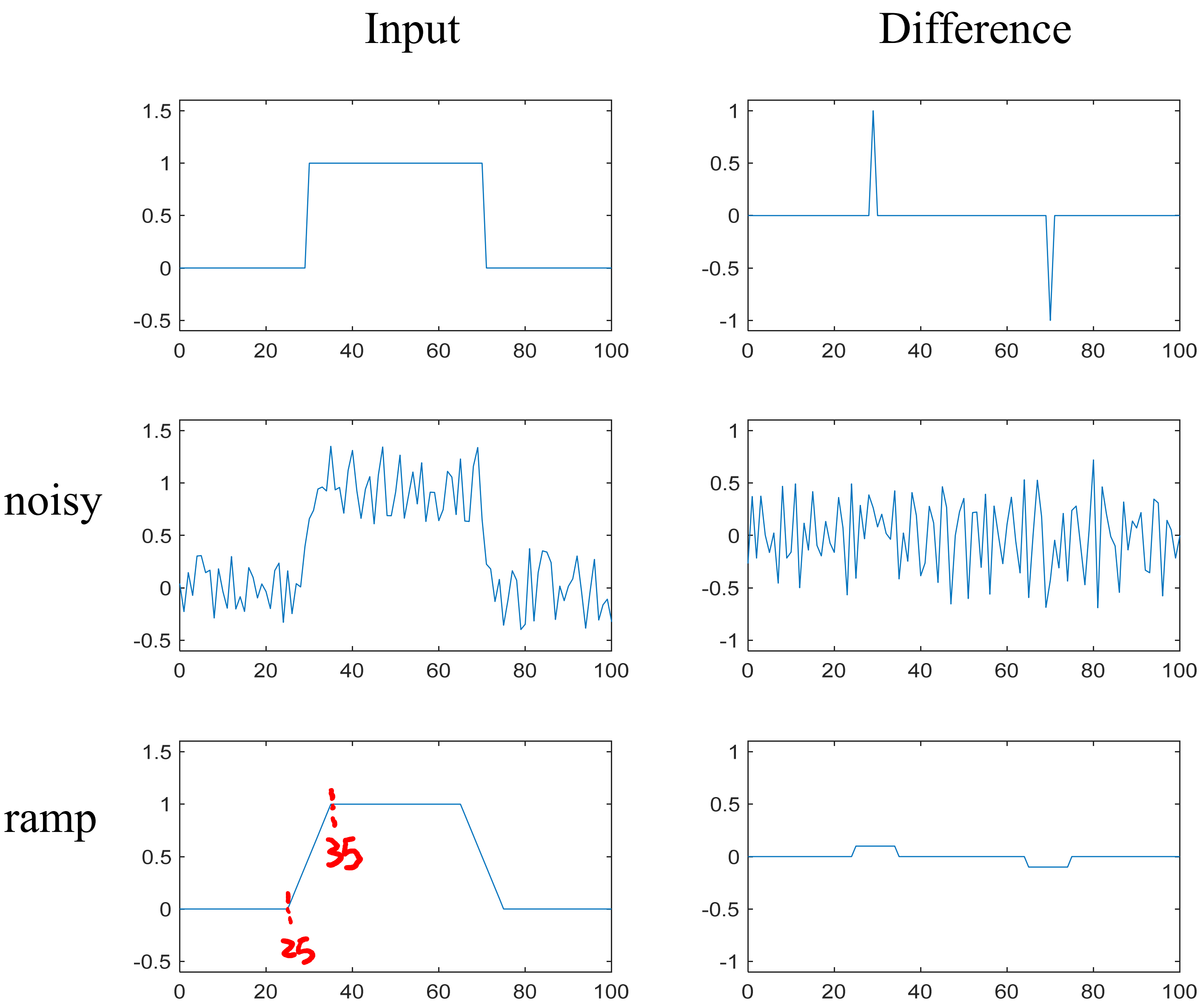
$h[0] = 0$

$h[n] = -\exp(-k|n|)$ $-L \le n < 0$

$h[n] = 0$ otherwise $k > 0$

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
1.5
1
0.5
0
-0.5
-1
0
20
40
60
80
100
25
35

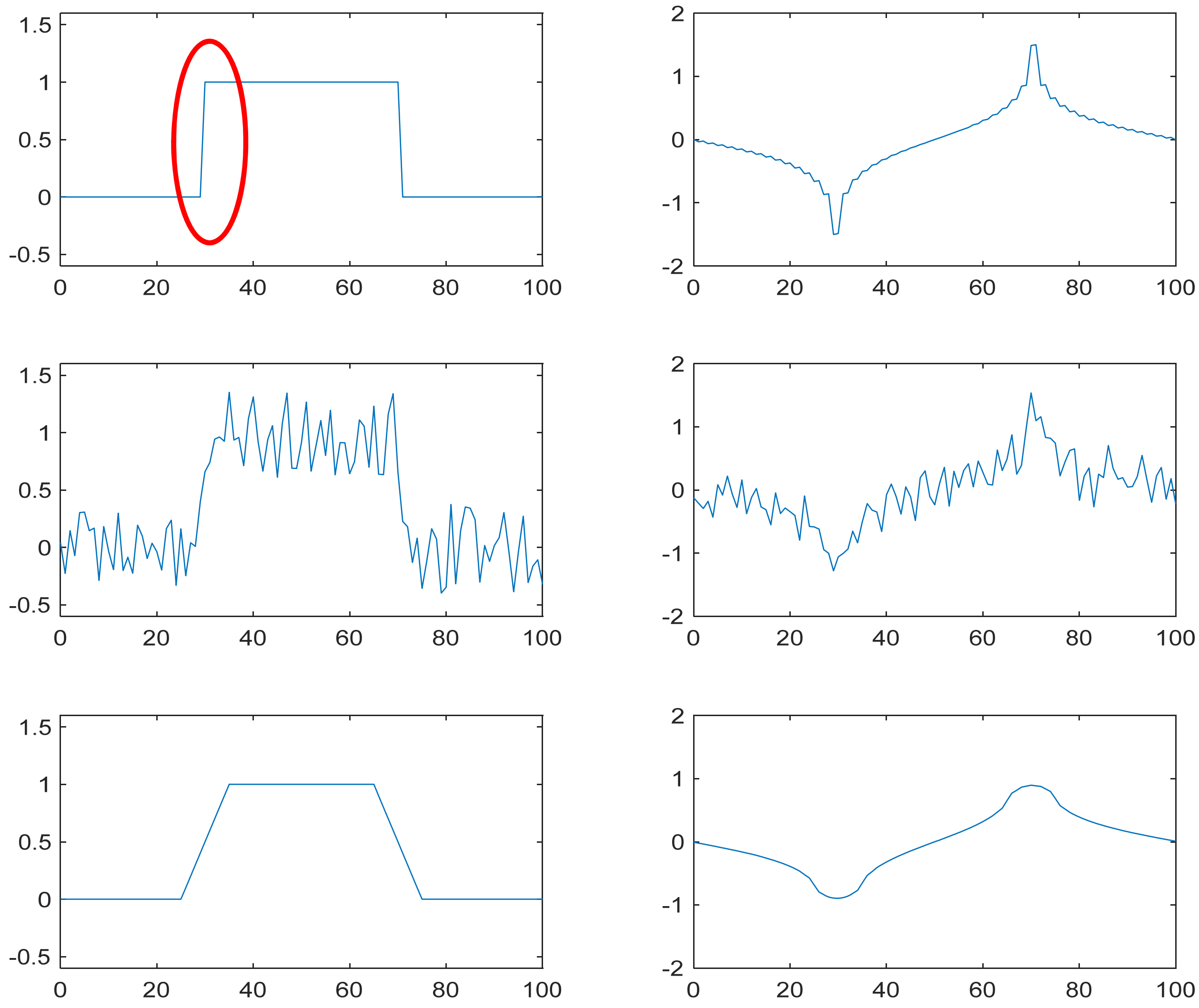
discrete Hilbert transform

Input

~~Difference~~

noisy

ramp



Other well-know edge detection filter:

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)

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

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

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

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

⇨

	0.25	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	