◎ 2-H 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)

$\Delta F = (f_1 - f_2)/f_s = (f_1 - f_2)T$ (f_s: sampling frequency, T: 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

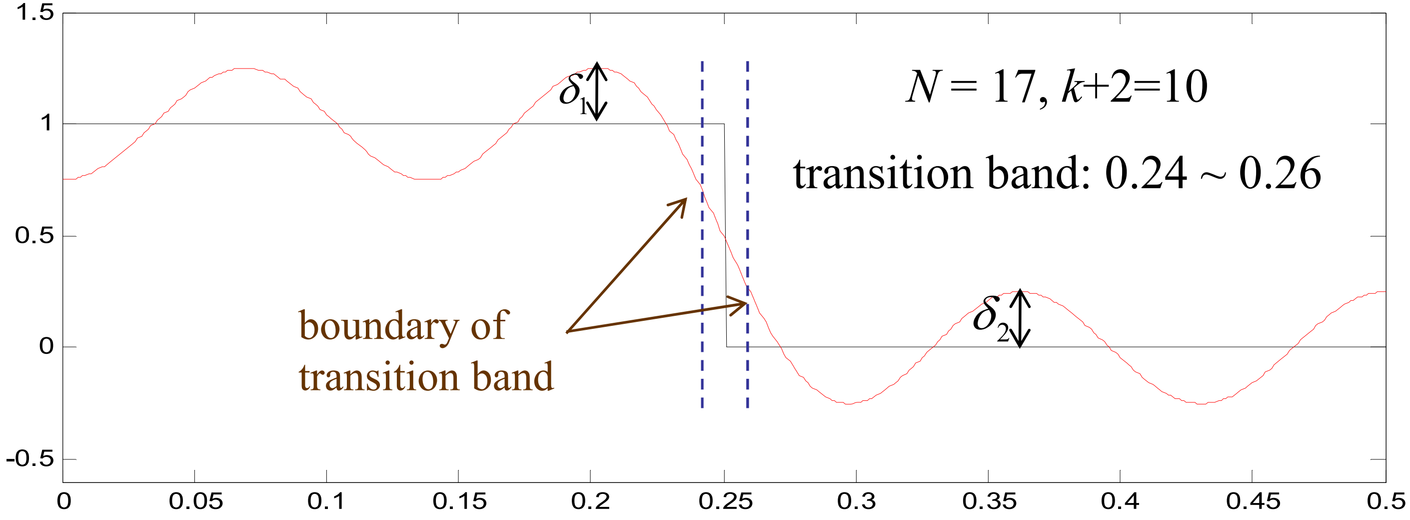
$$N = \frac{2}{3}\frac{1}{\Delta F}\log_{10}\left(\frac{1}{10\delta_1\delta_2}\right) \qquad \frac{3}{2}N\Delta F = \log_{10}\left(\frac{1}{10\delta_1\delta_2}\right)$$

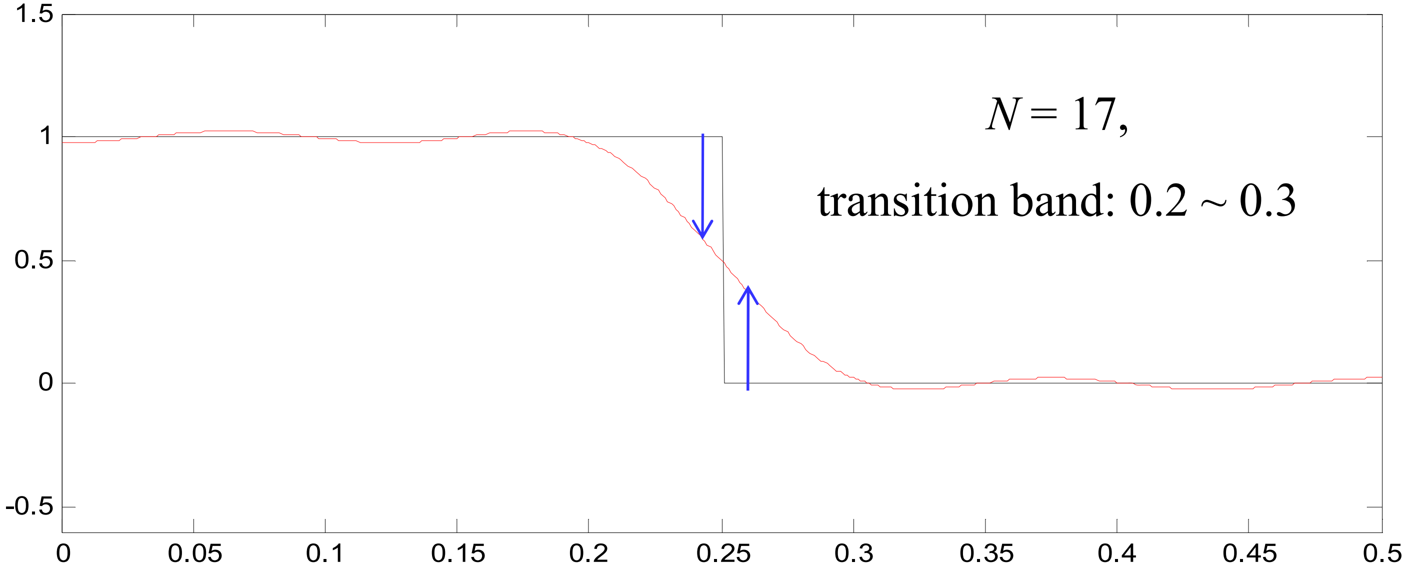
$$\delta_1\delta_2 = 10^{-3N\Delta F/2-1}$$

[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？

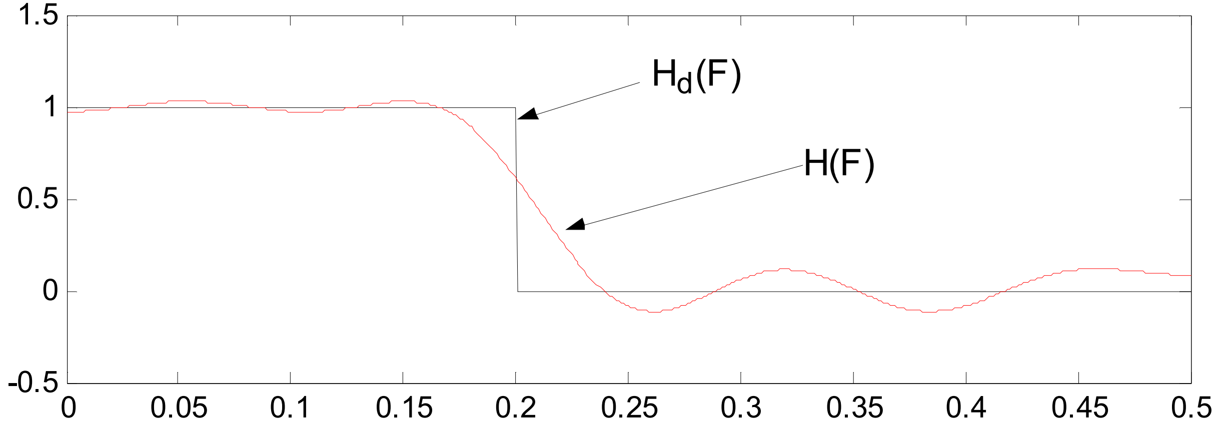

1.5
1
0.5
0
-0.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
δ_1
$N = 17$, k+2=10
transition band: 0.24 ~ 0.26
boundary of transition band
δ_2


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

◉ 2-I Relations between Weight Functions and Accuracy

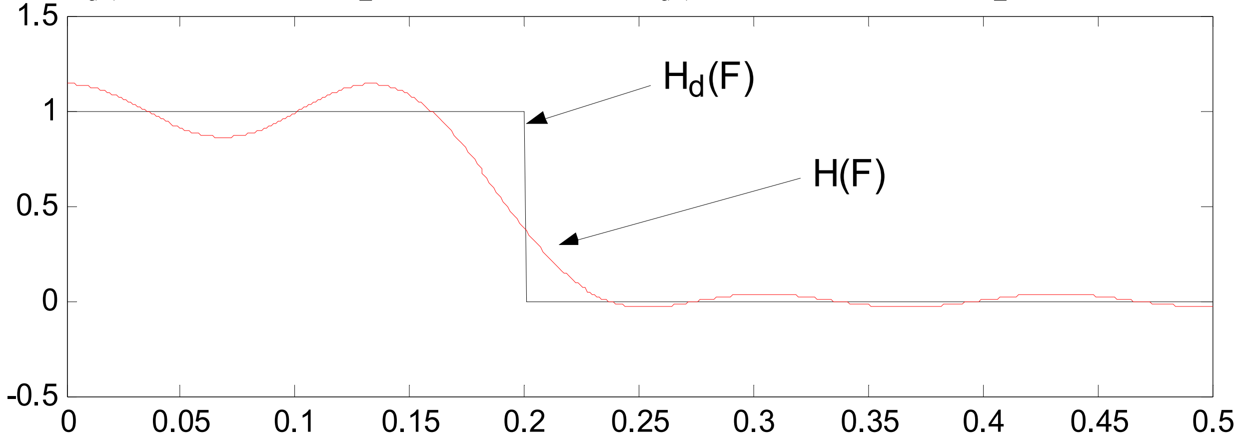
If we treat the passband more important than the stop band

$W(f) = 1$ in the passband, $0 < W(f) < 1$ in the stopband

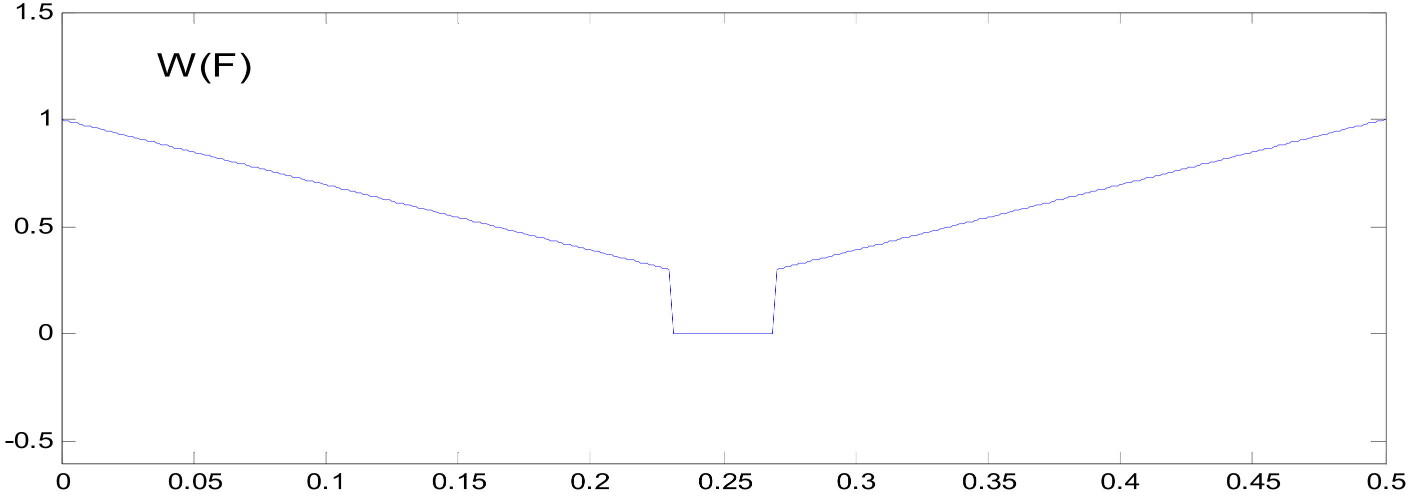


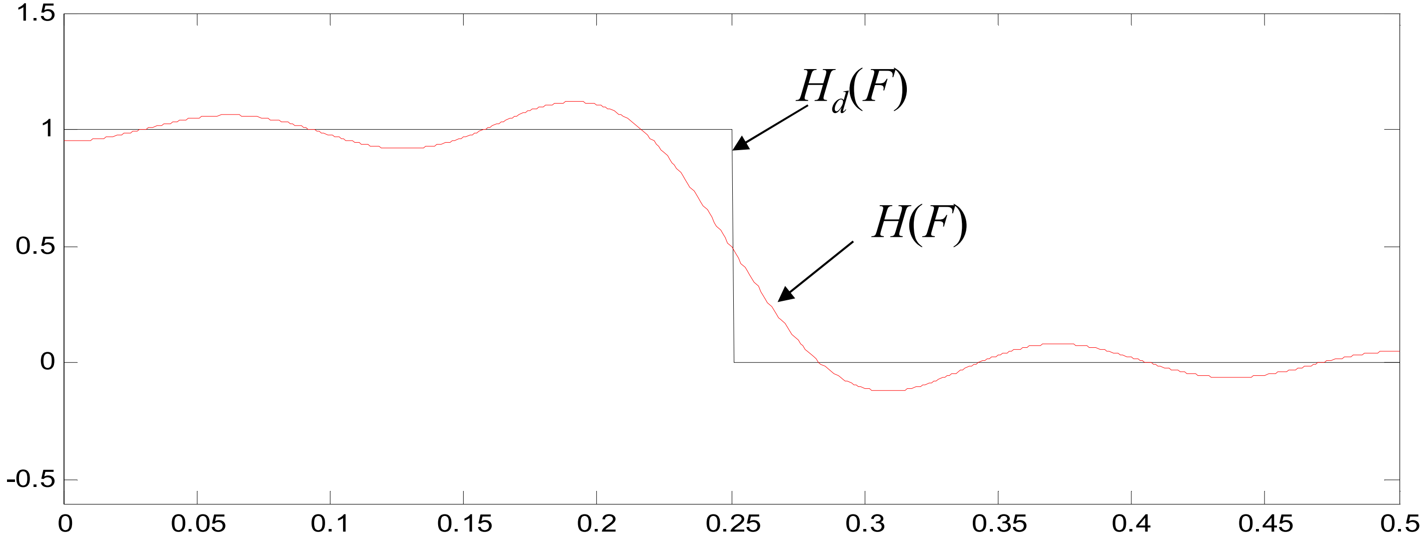
If we treat the stop band more important than the pass band

$0 < W(f) < 1$ in the passband, $W(f) = 1$ in the stopband



Larger error near the transition band





◎ 2-J 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

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

$F = f/f_s$

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

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

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

$$n = 0 \sim k$$

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)

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

$$\tau = 0 \sim k,\ n = 0 \sim k$$

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

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

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

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

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

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

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

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

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

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

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

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

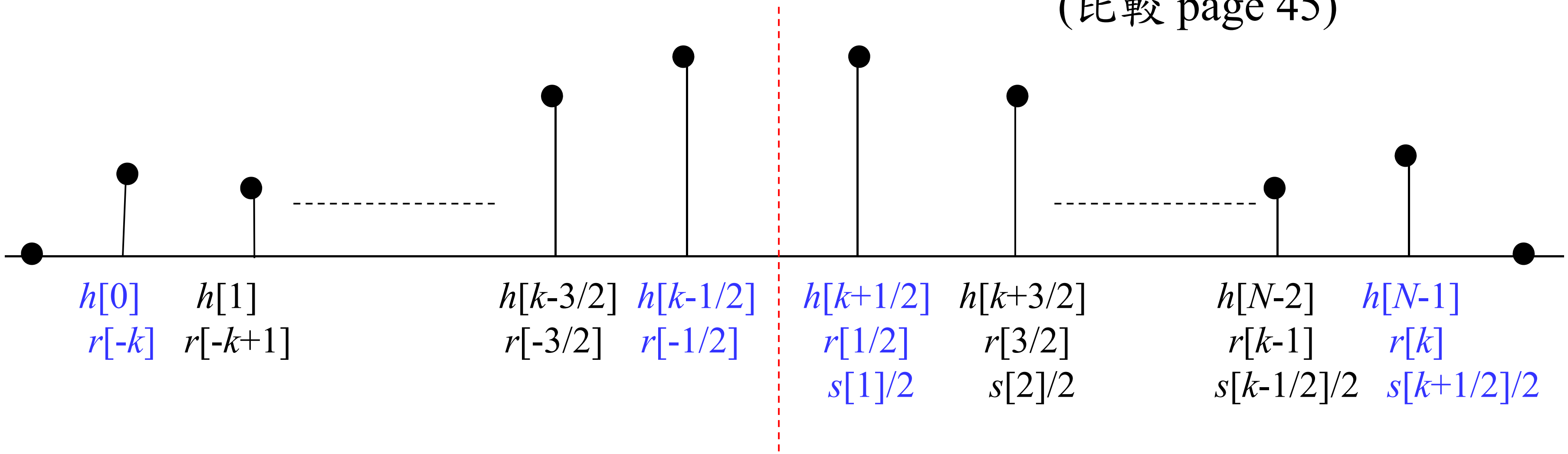
  $h[n]=-h[N-1-n]$ (odd symmetric) and N is even.

$k=(N-1)/2$

- 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)

(比較 page 45)



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

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

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

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

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

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

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

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

由於 n 和 $n+1$ 兩項相加可得

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

所以可以「判斷」$R(F)$ 能被改寫成

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

求 $s_1[n]$ 和 $s[n]$ 之間的關係

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

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

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

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

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

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

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

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

(令 $k_1 + 1/2 = k$)

比較係數可得

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

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

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

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

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

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

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

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

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

令 $k_1 = k - 1$, 比較係數可得

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

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

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

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

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

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
```