

4. Some Popular Filters

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

highpass

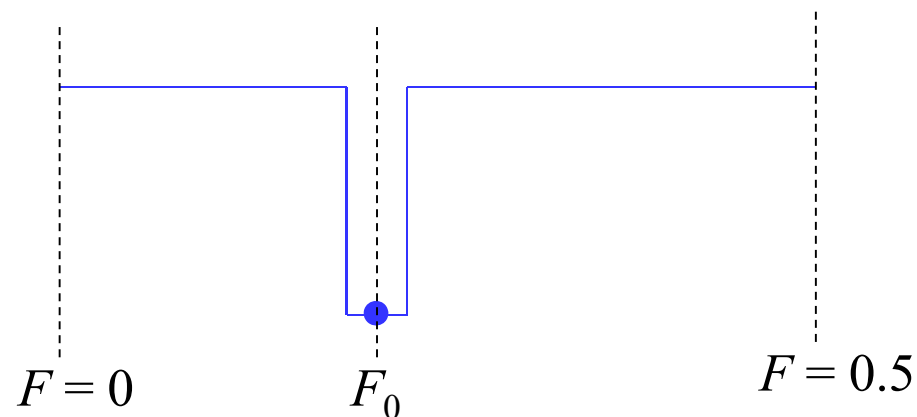
bandpass

lowpass

allpass

bandstop

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



思考: 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)

最簡單的 smoother:

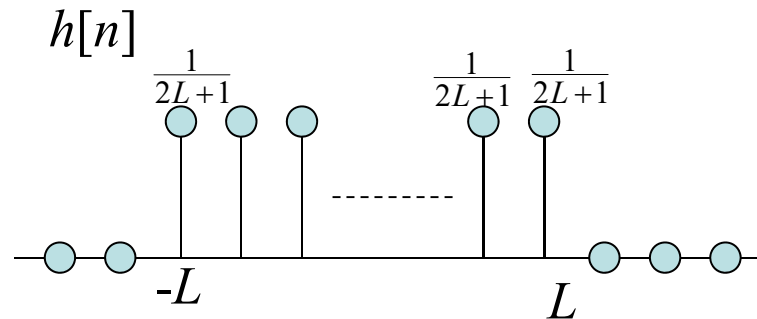
find the average
$$y[n] = \frac{1}{2L+1} \sum_{\tau=n-L}^{n+L} x[\tau]$$

近似 low-pass filter

可改寫成

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

$h[n]$ 如右圖



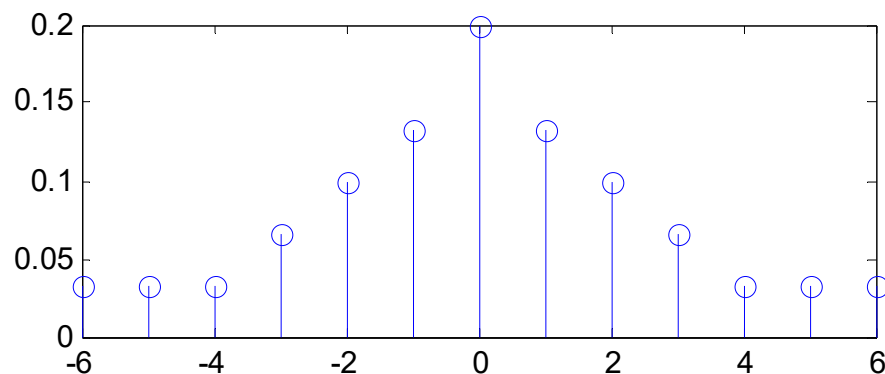
$$y[n] = \sum_{\tau} x[n-\tau]h[\tau] = \sum_{\tau=-L}^L x[n-\tau] \frac{1}{2L+1} = \frac{1}{2L+1} \sum_{\tau=-L}^L x[n+\tau]$$

一般型態的 smoother

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

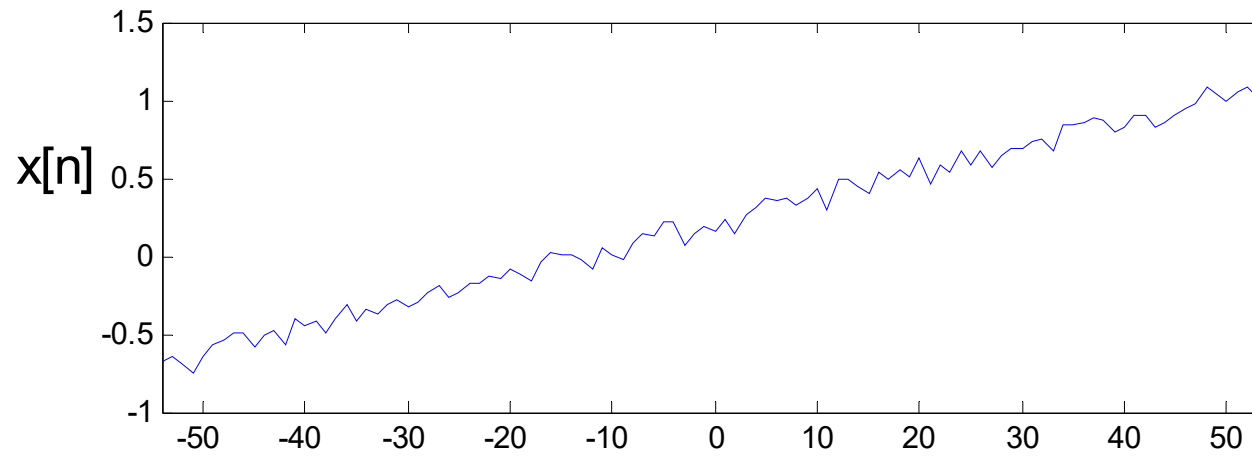
Choose (1) $h[n] = h[-n]$

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

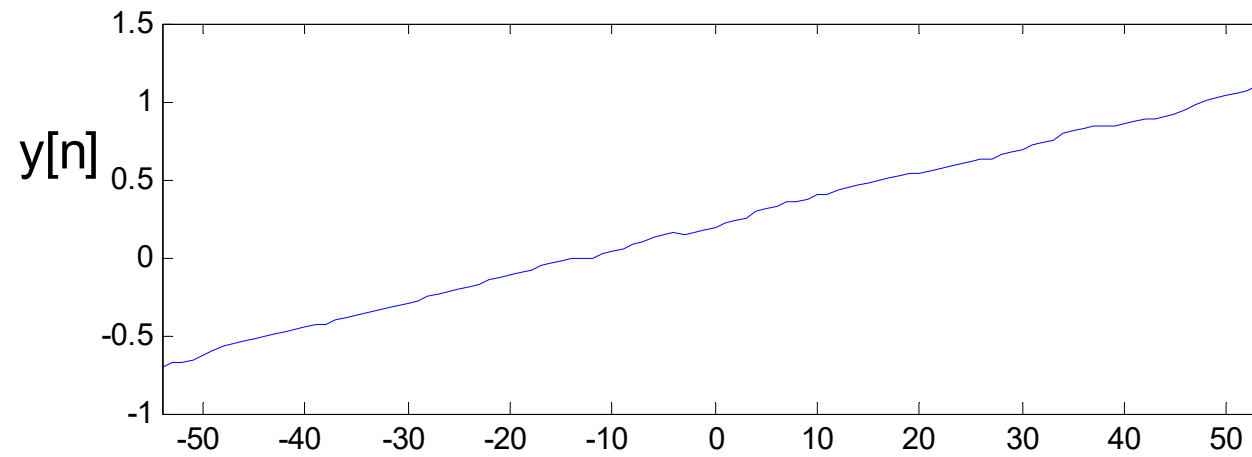


任何能量隨著 $|n|$ 遞減的 even function，都可以當成 smoother filter

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 $-f_s/2 < f < f_s/2$,

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

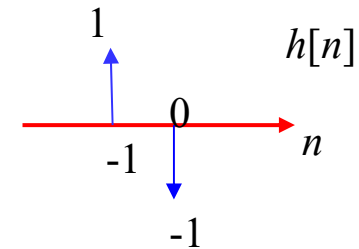
(b) Difference (一個簡單取代 differentiation 的方法)

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

$$h[n] = 1 \text{ when } n = -1, \quad h[n] = -1 \text{ when } n = 0,$$

$$h[n] = 0 \text{ otherwise}$$

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



These two filters are equivalent only at low frequencies

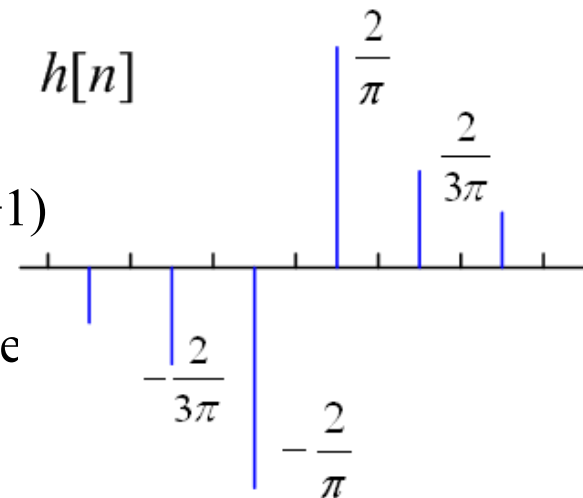
(C) Discrete Hilbert Transform

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

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

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

$$H(F) = H(F+1)$$



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

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

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

(D) Edge Detection

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

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

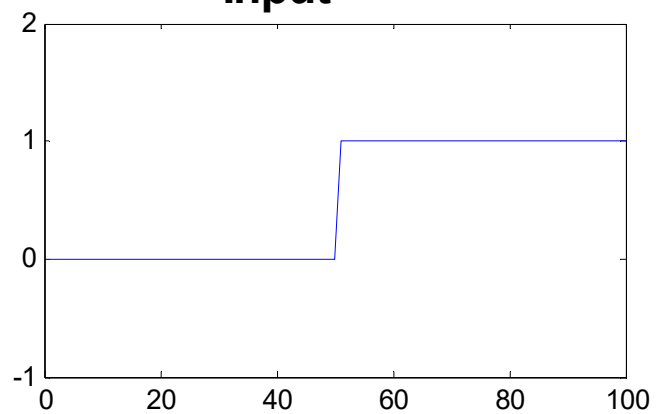
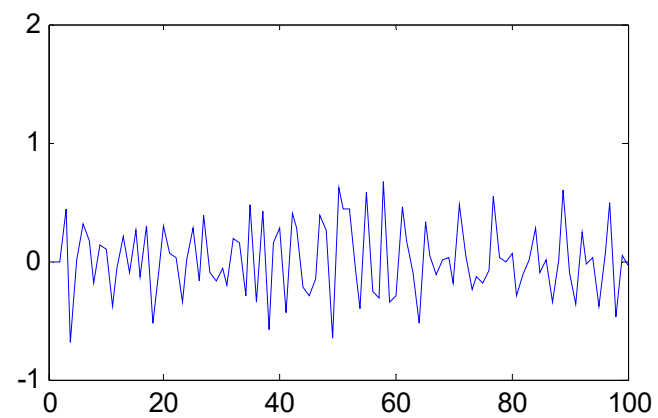
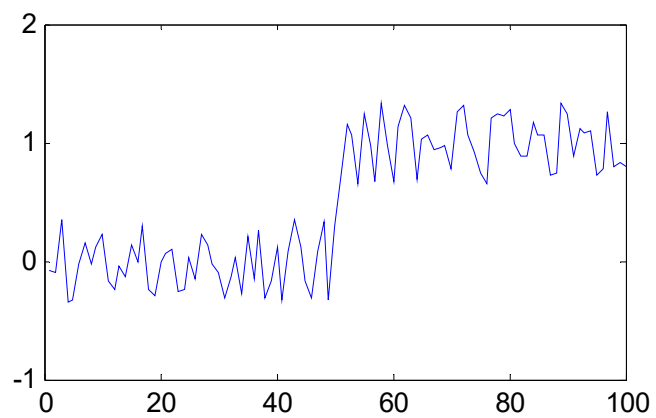
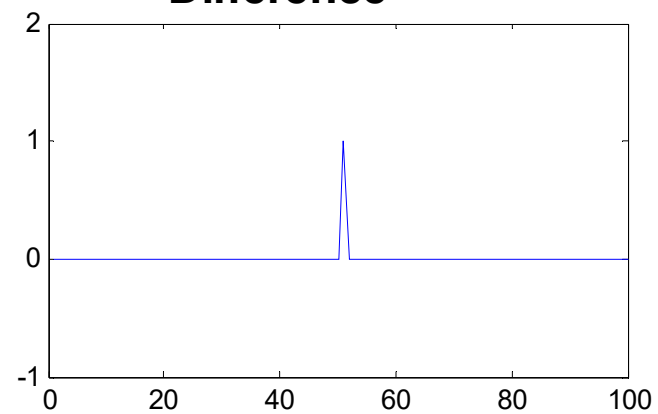
Difference 和 discrete Hilbert transform 都可用作 edge detection

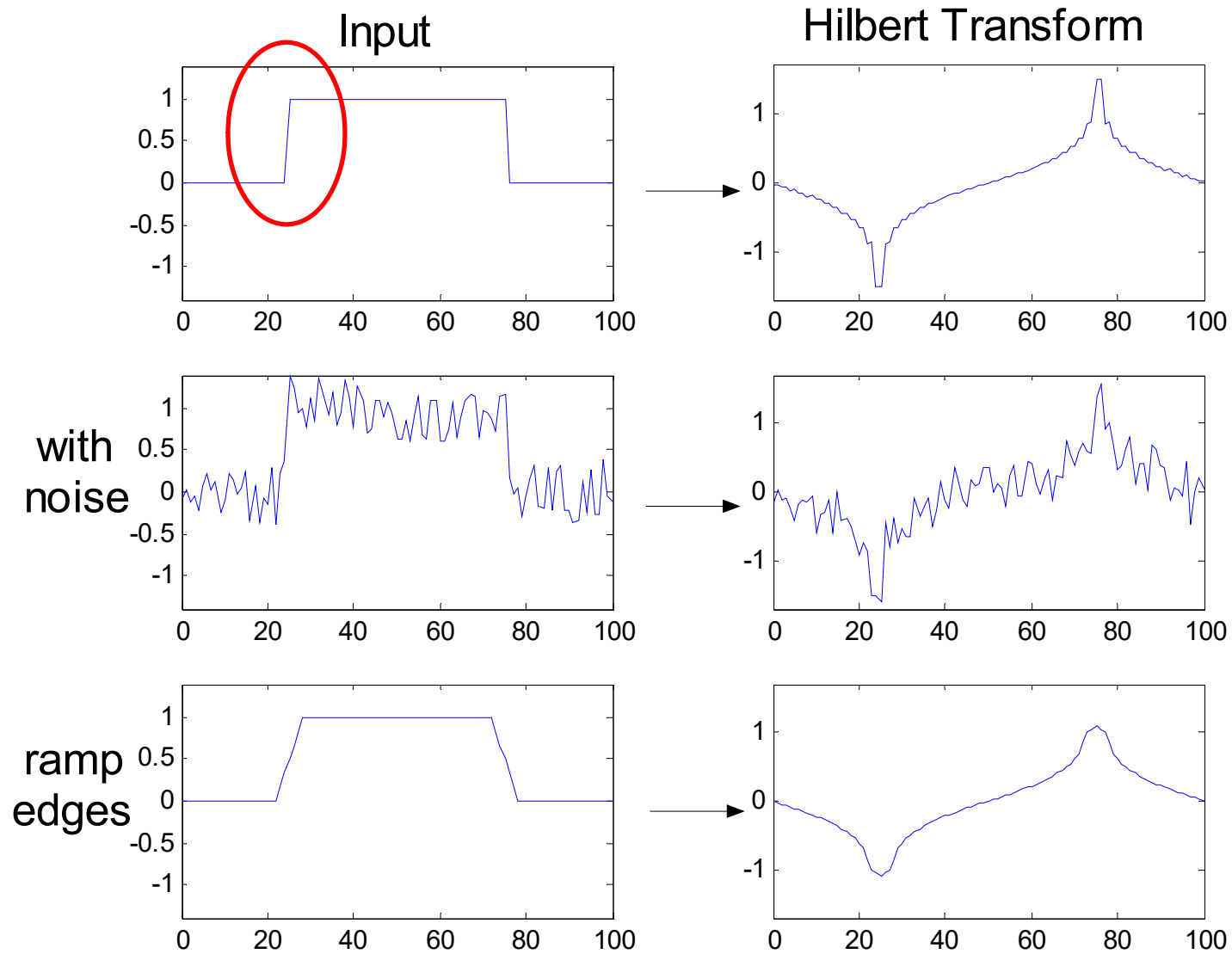
(1) 任何能量隨著 $|n|$ 遞減的 odd function，都可以當成 edge detection filter

(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**



© 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 $\tau_1 \leq n \leq \tau_2$

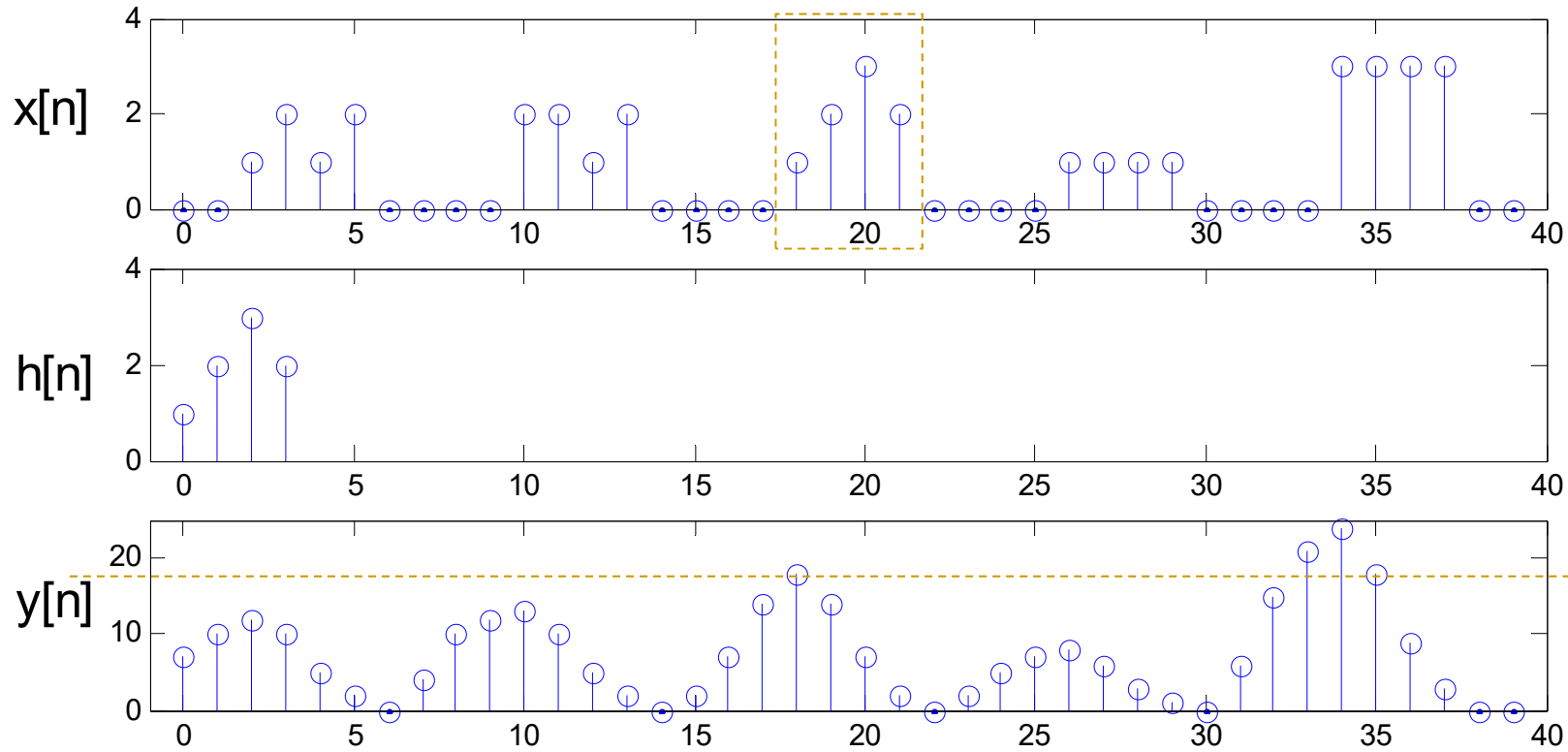
$x[n]$: input pattern, $h[n]$: the desired pattern

2-D form:

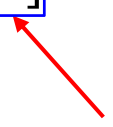
$$y[m, n] = x[m, n] * h^*[-m, -n] = \sum_{\tau=\tau_1}^{\tau_2} \sum_{\rho=\rho_1}^{\rho_2} x[m+\tau, n+\rho] h^*[\tau, \rho]$$

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

Example



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



The result of the convolution should be normalized!

- Normalization Form

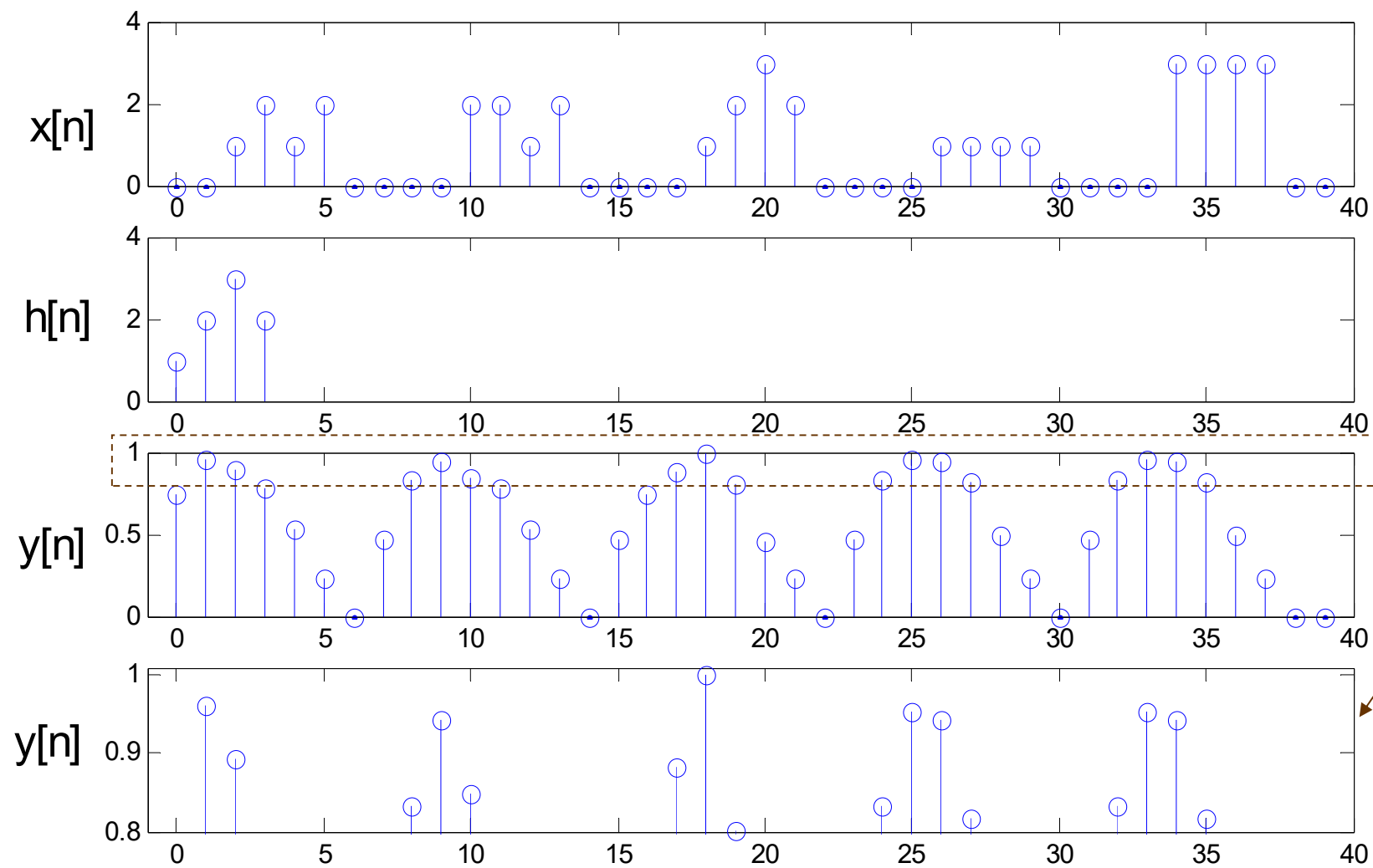
$$y[n] = \frac{\sum_{\tau=\tau_1}^{\tau_2} x[n+\tau] h^*[\tau]}{\sqrt{\sum_{s=n+\tau_1}^{n+\tau_2} |x[s]|^2 \sum_{s=\tau_1}^{\tau_2} |h[s]|^2}} \quad \text{when} \quad \sum_{s=n+\tau_1}^{n+\tau_2} |x[s]|^2 \neq 0$$

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

2-D Case

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

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



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

Particle filter:

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

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

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

It is used for [system modeling](#) or [prediction](#).

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

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

(Nobert Wiener 維納, AD 1949)


- 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 $s[n]$ and the received signal $x_i[n]$ is $R_{si}[n, \sigma]$,
- (b) the auto-correlation of the received signal (denoted by $R_{ii}[n, \sigma]$), then the transfer function of the optimal filter can be designed as

where  $H_{opt}(F) = R_{S,I}(F, F) / R_{I,I}(F, F)$

$$R_{S,I}(F, F) = \sum_{\sigma} \sum_n e^{j2\pi(F\sigma - Fn)} R_{si}[n, \sigma]$$

$$R_{I,I}(F, F) = \sum_{\sigma} \sum_n e^{j2\pi(F\sigma - Fn)} R_{ii}[n, \sigma]$$

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](#))

$$y[n] = x[n] * k[n] \quad \begin{array}{l} x[n]: \text{original signal, } y[n]: \text{received signal} \\ k[n]: \text{effect of the system} \end{array}$$

Equalizer:

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

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

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

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

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

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

$$\begin{aligned} H(F)Y(F) &= X(F)H(F)K(F) + H(F)M(F) \\ &= X(F) + \frac{M(F)}{K(F)} \end{aligned}$$

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:

$$H(F) = \frac{1}{\frac{1}{K^*(F)} \frac{E(|M(F)|^2)}{E(|X(F)|^2)} + K(F)} \quad E: \text{ mean}$$

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

c is large when the SNR is small

c is small when the SNR is large

- Equalizer for the Multiple Path Problem

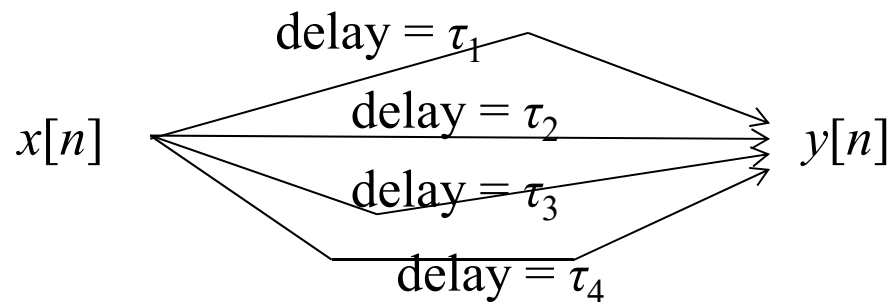
$$k[n] = \alpha_1 \delta[n - \tau_1] + \alpha_2 \delta[n - \tau_2] + \alpha_3 \delta[n - \tau_3] + \dots$$

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

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

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

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



- 缺點: (1) $H(z)$ 可能 unstable
(2) $H(z)$ is usually a dynamic response
- 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

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