V. Homomorphic Signal Processing

同質

◉ 5-A Homomorphism

Homomorphism is a way of "carrying over" operations from one algebra system into another.

Ex. convulution $\xrightarrow[\textit{Fourier}]{}$ multiplication $\xrightarrow[\log]{}$ addition

把複雜的運算，變成效能相同但較簡單的運算

◉ 5-B Cepstrum 倒頻譜

$$\hat{X}(Z)\Big|_{z=e^{i2\pi F}} = \log X(Z)\Big|_{z=e^{i2\pi F}} = \log\left|X(Z)\right|_{z=e^{i2\pi F}} + j\arg[X(e^{i2\pi F})]$$

For the process of cepstrum (denoted by $D_*[\ \cdot\ ]$)

convolution　　product　　addition　　addition

$$x[n] \xrightarrow{*} \underset{Z}{\boxed{FT[\ \cdot\ ]}} \xrightarrow[X(F)]{\times} \boxed{\text{Log}[\ ]} \xrightarrow[\hat{X}(F)]{+} \underset{Z^{-1}}{\boxed{FT^{-1}[\ \cdot\ ]}} \xrightarrow{+} \hat{x}[n]$$

$x[n]*h[n]$　　$X(F)\times H(F)$　　$\hat{X}(F)+\hat{H}(F)$　　$\hat{x}[n]+\hat{h}[n]$

$\hat{X}(F)=\log(X(F))$　$\hat{H}(F)=\log(H(F))$

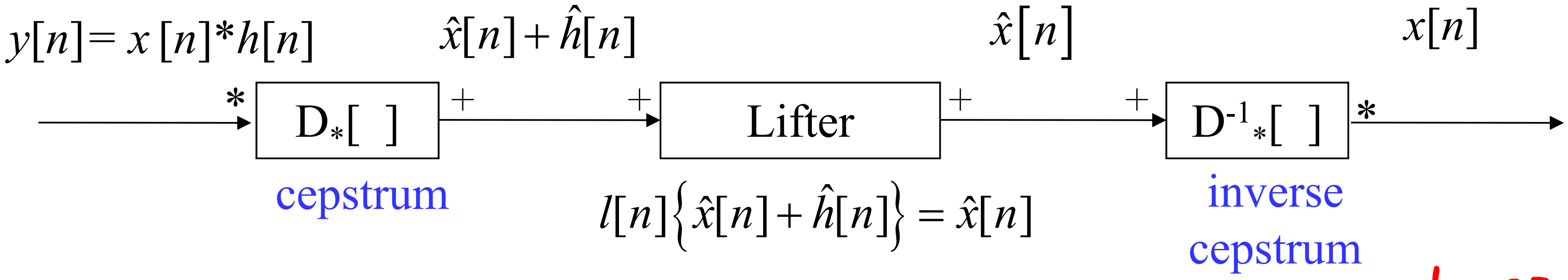
FT: discrete-time Fourier transform

$\hat{x}[n]=IFT(\log(X(F)))$

$\hat{h}[n]=IFT(\log(H(F)))$

- 由 $y[n]=x[n]*h[n]$ 重建 $x[n]$

$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$

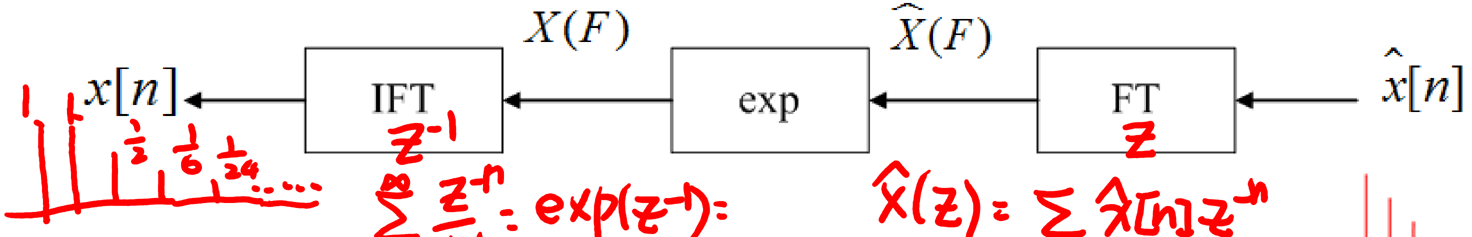


For the process of the inverse cepstrum $D^{-1}_*[\cdot]$

if $\hat{x}[n] = \delta[n-1]$

1

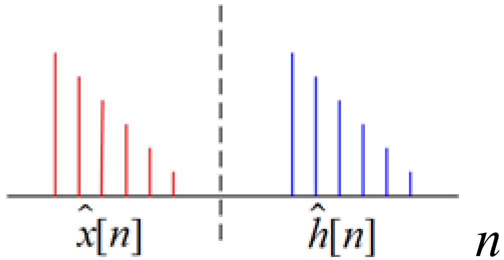
$n=1$



$\sum_{n=0}^{\infty} \frac{z^{-n}}{n!} = \exp(z^{-1}) =$

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

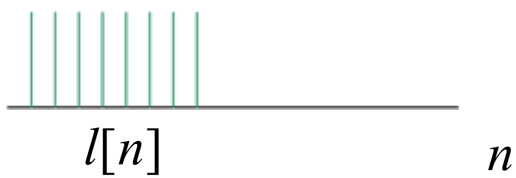
z^{-1}



- 有趣的名詞

$\hat{x}[n]$	cepstrum	spectrum	倒頻譜
n	quefrency	frequency	倒頻率
$l[n]$	lifter	filter	倒濾波器

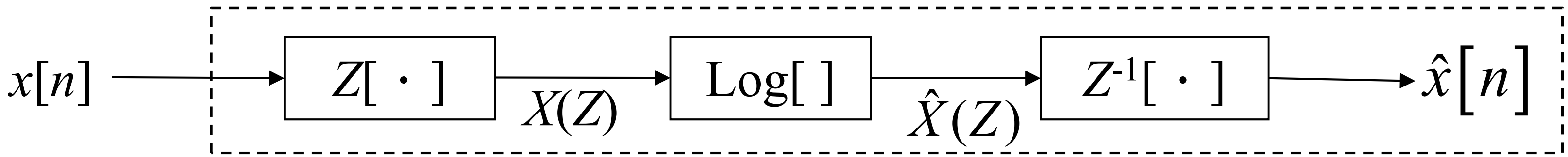
lifter



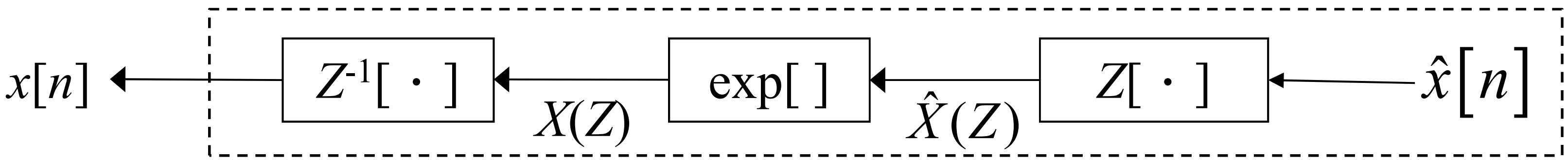
$l[n]\{\hat{x}[n]+\hat{h}[n]\}=\hat{x}[n]$

Using the Z transforms instead of the Fourier transforms:

For the process of cepstrum



For the process of the inverse cepstrum



◎ 5-C Methods for Computing the Cepstrum

- **Method 1**: Compute the inverse discrete time Fourier transform:

$$\hat{x}[n] = \int_{-1/2}^{1/2} \hat{X}(F) e^{i2\pi nF} dF \qquad \text{: inverse F.T}$$

$\hat{X}(F) = \log X(F)$

where $\hat{X}(F) = \log|X(F)| + j\arg[X(F)]$

$X(F) = |X(F)| e^{j\arg X(F)}$

ex: $X(F) = j$ 通解?

$|X(F)| = 1$, $\arg(X(F)) = \frac{\pi}{2}, \frac{-3\pi}{2}, \ldots$ ambiguity for phase

Problems: (1) $\log|X(F)| \to -\infty$ if $|X(F)| \to 0$

(2) $\arg(X(F))$ has infinite number of solutions.

Actually, the COMPLEX Cepstrum is REAL for real input

實際上計算
cepstrum的方法

• Method 2 (From Poles and Zeros of the Z Transform)

time delay

$$X(Z) = \frac{A \cancel{Z^{r}} \prod_{k=1}^{m_i}(1-a_k Z^{-1}) \prod_{k=1}^{m_0}(1-b_k Z)}{\prod_{k=1}^{P_i}(1-c_k Z^{-1}) \prod_{k=1}^{P_0}(1-d_k Z)}$$

where

$$|a_k| \ , |b_k| \ , |c_k| \ , |d_k| \le 1$$

a_k: zeros inside unit circle
c_k: poles inside unit circle

b_k^{-1}: zeros outside unit circle
d_k^{-1}: poles outside unit circle

∏: 連乘
Σ: 連加

$$\therefore \hat{X}(Z) = \log X(Z) = \log A + \cancel{r \cdot \log Z} + \sum_{k=1}^{m_i} \log(1-a_k Z^{-1}) + \sum_{k=1}^{m_0} \log(1-b_k Z)$$

$$-\sum_{k=1}^{P_i} \log(1-c_k Z^{-1}) - \sum_{k=1}^{P_0} \log(1-d_k Z)$$

ex: $z^2 - 0.7z + 0.1$
$= (z-0.5)(z-0.2)$
$= z^2 (1-0.5z^{-1})(1-0.2z^{-1})$

$$\therefore \hat{X}(Z) = \log X(Z) = \log A + \cancel{r \cdot \log Z} + \sum_{k=1}^{m_i} \log(1 - a_k Z^{-1}) + \sum_{k=1}^{m_0} \log(1 - b_k Z)$$

$$-\sum_{k=1}^{P_i} \log(1 - c_k Z^{-1}) - \sum_{k=1}^{P_0} \log(1 - d_k Z)$$

$$\downarrow Z^{-1} \text{ (inverse Z transform)}$$

?

Taylor series

$$f(t) = f(t_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(t_0)}{n!}(t - t_0)^n$$

$$\frac{d^n}{dt^n}\log(1+t) = \frac{(-1)^{n-1}(n-1)!}{(1+t)^n}$$

$$\log(1+t) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}(n-1)!}{n!} t^n \quad (t_0 = 0)$$

$$\log(1+t) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} t^n$$

$$\log(1 - a_k Z^{-1}) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}(-1)^n}{n} a_k^n Z^{-n}$$

$$= \sum_{n=1}^{\infty} \frac{-1}{n} a_k^n Z^{-n}$$

$$\log(1 - b_k Z) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}(-1)^n b_k^n Z^n$$

$$= \sum_{n=1}^{\infty} \frac{-1}{n} b_k^n Z^n \quad n \to -n$$

$$= \sum_{n=-\infty}^{-1} \frac{1}{n} b_k^{-n} Z^{-n}$$

$\boxed{\text{Taylor series expansion}}$ $\Big\downarrow Z^{-1}$ (Suppose that $r = 0$)

$$\hat{x}[n] = \begin{cases} \log(A) & ,n = 0 \\ -\sum_{k=1}^{m_i} \frac{a_k{}^n}{n} + \sum_{k=1}^{P_i} \frac{c_k{}^n}{n} & ,n > 0 \\ \sum_{k=1}^{m_0} \frac{b_k{}^{-n}}{n} - \sum_{k=1}^{P_0} \frac{d_k{}^{-n}}{n} & ,n < 0 \end{cases}$$

Poles & zeros inside unit circle, right-sided sequence

Poles & zeros outside unit circle, left-sided sequence

Note:

(1) $\hat{x}[n]$ always decays with $|n|$.

(2) 在 complex cepstrum domain

Minimum phase 及 maximum phase 之貢獻以 $n = 0$ 為分界切開

(3) For FIR case, there is no c_k and d_k

(4) The complex cepstrum is unique and of infinite duration for both positive & negative n, even though $x[n]$ is causal & of finite durations

$\boxed{\hat{x}[n] \text{ is always IIR}}$

- **Method 3**

$$Z \cdot \hat{X}'(Z) = Z \cdot \frac{X'(Z)}{X(Z)}$$

$$\therefore ZX'(Z) = Z\hat{X}'(Z) \cdot X(Z)$$

$$\Big\downarrow \; Z^{-1}$$

$$n\,x[n] = \sum_{k=-\infty}^{\infty} k\,\hat{x}[k]\,x[n-k]$$

$$\therefore x[n] = \sum_{k=-\infty}^{\infty} \frac{k}{n}\hat{x}[k]\,x[n-k] \qquad for\ n \neq 0$$

Suppose that $x[n]$ is causal and has minimum phase, i.e. $x[n] = \hat{x}[n] = 0,\ n < 0$

$$x[n] = \sum_{k=-\infty}^{\infty} \frac{k}{n} \hat{x}[k] x[n-k] \qquad for\ n \neq 0$$

$$\Rightarrow x[n] = \sum_{k=0}^{n} \frac{k}{n} \hat{x}[k] x[n-k] \qquad for\ n > 0 \qquad (causal\ sequence)$$

$$x[n] = \hat{x}[n] x[0] + \sum_{k=0}^{n-1} \frac{k}{n} \hat{x}[k] x[n-k]$$

For a minimum phase sequence $x[n]$

$$\hat{x}[n] = \begin{cases} 0 & ,n < 0 \\ \dfrac{x[n]}{x[0]} - \displaystyle\sum_{k=0}^{n-1} \left(\frac{k}{n}\right) \hat{x}[k] \frac{x[n-k]}{x[0]} & ,n > 0 \\ \log A & ,n = 0 \end{cases} \qquad \text{recursive method}$$

Determining $\hat{x}[n]$ from $\hat{x}[0], \hat{x}[1], \cdots, \hat{x}[n-1]$

For anti-causal and maximum phase sequence, $x[n] = \hat{x}[n] = 0,\ n > 0$

$$x[n] = \sum_{k=n}^{0} \frac{k}{n} \hat{x}[k] x[n-k] \qquad ,n < 0$$

$$= \hat{x}[n] x[0] + \sum_{k=n+1}^{0} \frac{k}{n} \hat{x}[k] x[n-k]$$

For maximum phase sequence,

$$\hat{x}[n] = \begin{cases} 0 & ,n > 0 \\ \log A & ,n = 0 \\ \dfrac{x[n]}{x[0]} - \displaystyle\sum_{k=n+1}^{0} \left(\frac{k}{n}\right) \hat{x}[k] \frac{x[n-k]}{x[0]} & ,n < 0 \end{cases}$$

◉ 5-D Properties

P.1) The complex cepstrum decays at least as fast as $\frac{1}{n}$

$$|\hat{x}[n]| < c\left|\frac{\alpha^n}{n}\right| \qquad -\infty < n < \infty$$

$$\alpha = \max(|a_k|, |b_k|, |c_k|, |d_k|)$$

P.2) If $X(Z)$ has no poles and zeros outside the unit circle, i.e. $x[n]$ is minimum phase, then

$$\hat{x}[n] = 0 \qquad \textit{for all } n < 0$$

because of no b_k, d_k

P.3) If $X(Z)$ has no poles and zeros inside the unit circle, i.e. $x[n]$ is maximum phase, then

$$\hat{x}[n] = 0 \qquad \textit{for all } n > 0$$

because of no a_k, c_k

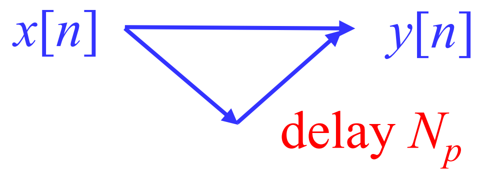
P.4) If $x[n]$ is of finite duration, then

$\hat{x}[n]$ has infinite duration

◉ 5-E Application of Homomorphic Deconvolution

(1) Equalization for Echo

$$y[n] = x[n] + \alpha\, x[n - N_p]$$



Let $p[n]$ be $p[n] = \delta[n] + \alpha\delta[n-N_p]$

$$y[n] = x[n] + \alpha\, x[n-N_p] = x[n] * p[n]$$

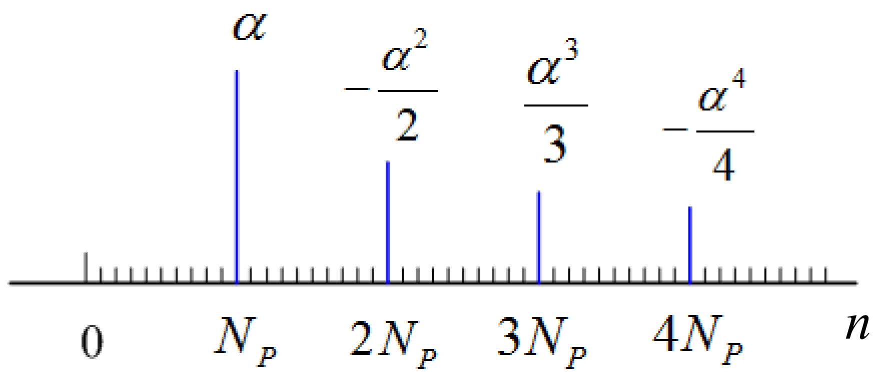
$$\log(1+t) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} t^k$$

$$P(Z) = 1 + \alpha Z^{-N_p}$$

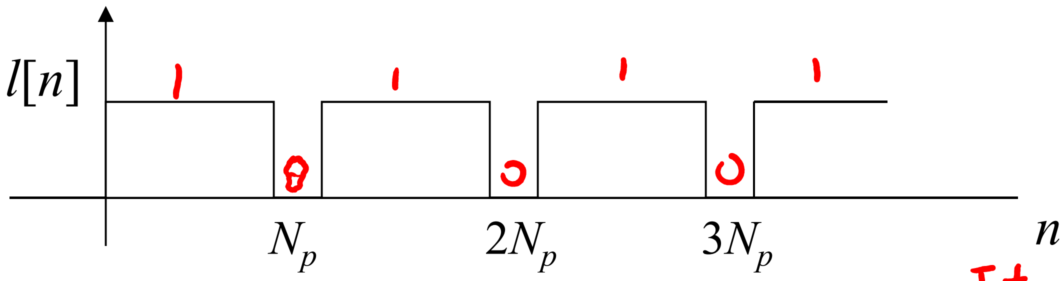
$$\hat{P}(Z) = \log\,(1 + \alpha Z^{-N_p}) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\alpha^k}{k} Z^{-kN_p}$$

$$\downarrow Z^{-1}$$

$$\hat{p}[n] = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\alpha^k}{k} \delta(n - k \cdot N_p)$$

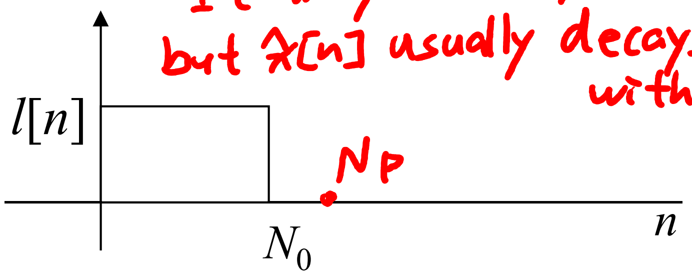


Filtering out the echo by the following "lifter":



or

It may destroy $\hat{x}[n]$ but $\hat{x}[n]$ usually decays fast with n.



$Np > N_0$

Q: For the case where N_p is unknown

(2) Representation of acoustic engineering

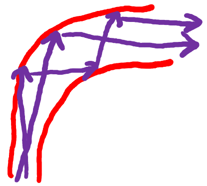
$$y[n] \;=\; x[n] \quad * \quad h[n]$$

Synthesized music — music — building effect：e.g. 羅馬大教堂的 impulse response

(3) Speech analysis

$$s[n] \quad = \quad g[n] \quad * \quad v[n] \quad * \quad p[n]$$

Speech wave	Global wave shape	聲道 Vocal tract impulse	Pitch



They can be separated by filtering in the complex cepstrum domain

(4) Seismic Signals
地震波

(5) Multiple-path analysis for any wave-propagation problem

用 cepstrum 將 multipath的影響去除

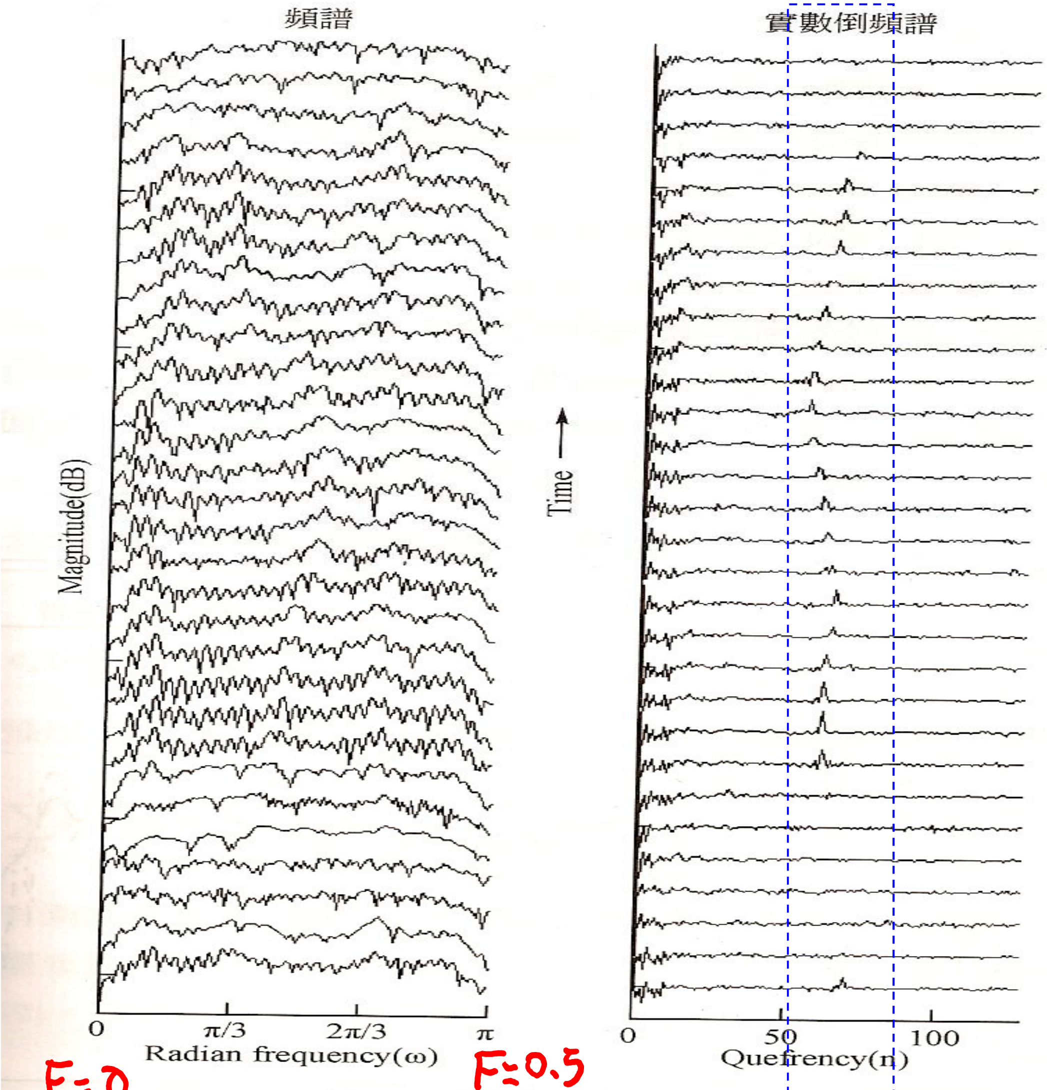


圖 7-5 元音頻譜與倒頻譜

From 王小川，“語音訊號處理”，全華出版，台北，民國94年。

From 王小川，“語音訊號處理”，全華出版，台北，民國94年。

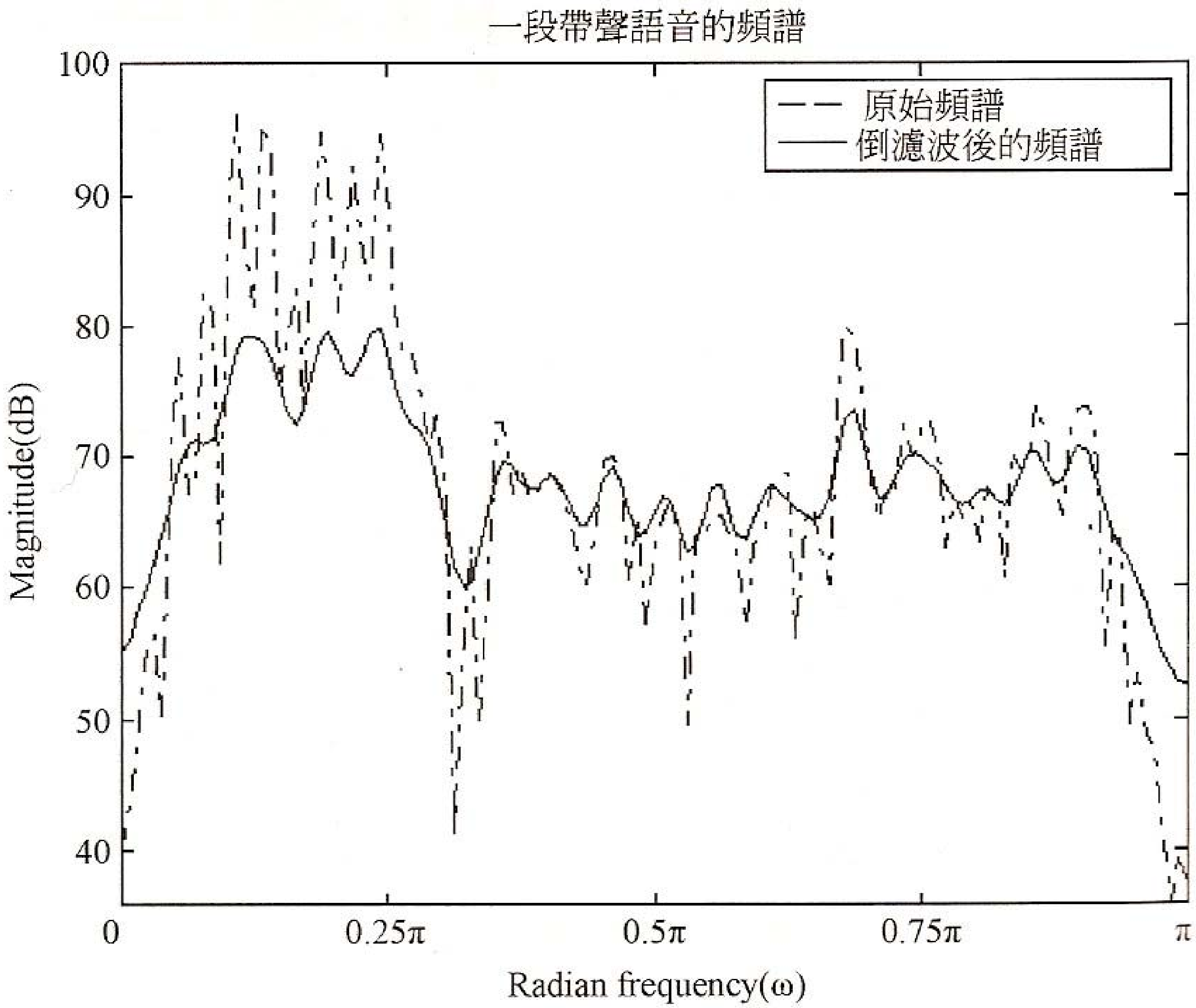


圖 7-6　經過倒濾波器作平滑處理的頻譜

◉ 5-F Problems of Cepstrum

(1) $|\log(X(Z))|$

(2) Phase

(3) Delay Z^{-k}

(4) Only suitable for the multiple-path-like problem

◉ 5-G Differential Cepstrum

$$\hat{x}_d(n) = Z^{-1}\left[\frac{X'(Z)}{X(Z)}\right] \qquad 或 \qquad \hat{x}_d[n] = \int_{-1/2}^{1/2} \frac{X'(F)}{X(F)} e^{i2\pi F}\, dF$$

inverse Z transform

Note: $$\frac{d}{dZ}\hat{X}(Z) = \frac{d}{dZ}\log(X(Z)) = \frac{X'(Z)}{X(Z)}$$

If $x(n) = x_1(n) * x_2(n)$

$$X(Z) = X_1(Z)\cdot X_2(Z)$$

$$X'(Z) = X_1'(Z)\cdot X_2(Z) + X_1(Z)\cdot X_2'(Z)$$

$$\frac{X'(Z)}{X(Z)} = \frac{X_1'(Z)}{X_1(Z)} + \frac{X_2'(Z)}{X_2(Z)} \qquad \therefore\ \hat{x}_d(n) = \hat{x}_{1d}(n) + \hat{x}_{2d}(n)$$

$x_1[n]$ and $Ax_1[n]$ has the same differential cepstrum

Advantages: no phase ambiguity

able to deal with the delay problem

• Properties of Differential Cepstrum

(1) The differential Cepstrum is shift & scaling invariant

不只適用於 multi-path-like problem

也適用於 pattern recognition

If $y[n] = A\, X[n - r]$

$$\Rightarrow \quad \hat{y}_d(n) = \begin{cases} \hat{x}_d(n) & , \quad n \neq 1 \\ -r + \hat{x}_d(1) & , \quad n = 1 \end{cases}$$

(Proof): $Y(z) = Az^{-r}X(z)$

$$Y'(z) = Az^{-r}X'(z) - rAz^{-r-1}X(z)$$

$$\frac{Y'(z)}{Y(z)} = \frac{X'(z)}{X(z)} - rz^{-1}$$

$$\hat{y}_d(n) = \hat{x}_d(n) - r\delta(n-1)$$

(2) The complex cepstrum $\hat{C}[n]$ is closely related to its differential cepstrum $\hat{x}_d[n]$ and the signal original sequence $x[n]$

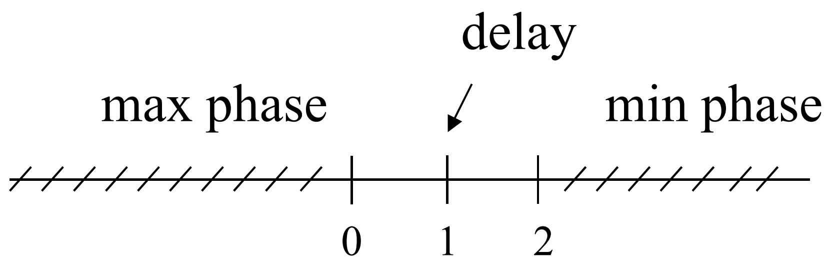
$$\hat{C}(n) = \frac{-\hat{x}_d(n+1)}{n} \qquad n \neq 0 \qquad \textit{diff cepstrum}$$

$$\textit{and} \quad -(n-1)\,x(n-1) = \sum_{k=-\infty}^{\infty} \hat{x}_d(n)\,x(n-k) \qquad \textit{recursive formula}$$

Complex cepstrum 做得到的事情, differential cepstrum 也做得到！

(3) If $x[n]$ is minimum phase (no poles & zeros outside the unit circle), then $\hat{x}_d[n] = 0$ for $n \le 0$

(4) If $x[n]$ is maximum phase (no poles & zeros inside the unit circle) , then $\hat{x}_d[n] = 0$ for $n \ge 2$



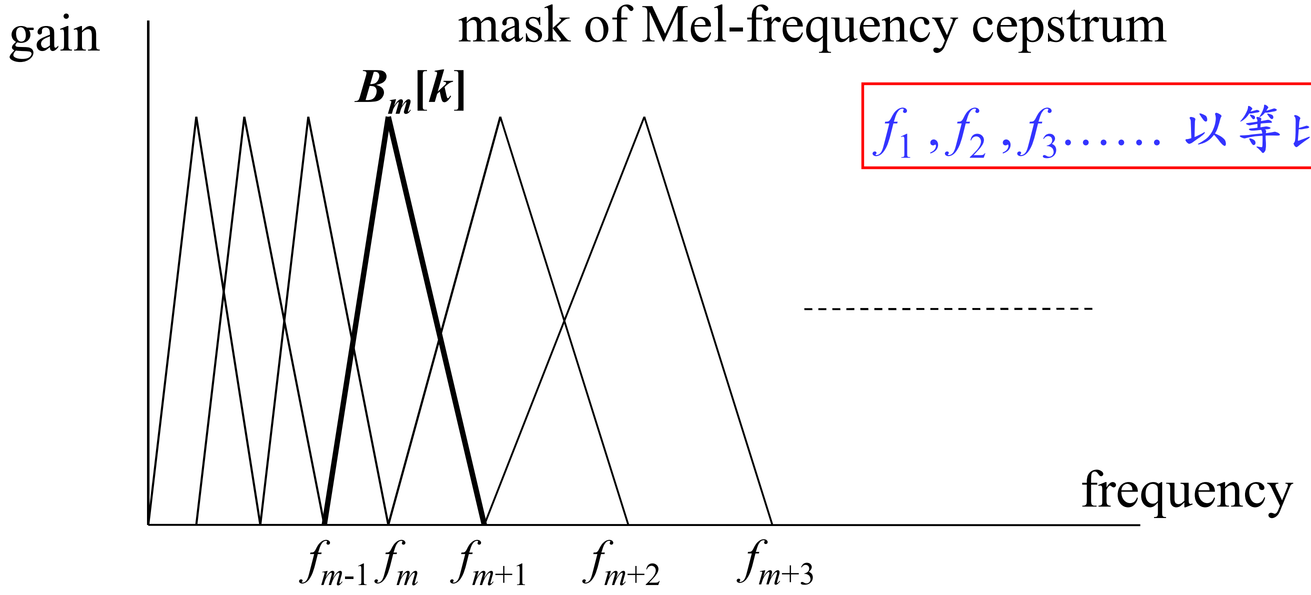
(5) If $x(n)$ is of finite duration, $\hat{x}_d[n]$ has infinite duration

Complex cepstrum decay rate $\propto \dfrac{1}{n}$

Differential Cepstrum decay rate 變慢了，$\because$ $\hat{x}_d(n+1) = n \cdot \hat{c}(n) \propto n \cdot \dfrac{1}{n} = 1$

◎ 5-H Mel-Frequency Cepstrum (梅爾頻率倒頻譜)

Take log in the frequency mask



$f_1, f_2, f_3 \ldots\ldots$ 以等比級數增加

$f_m = \alpha^m f_0$

$B_m[k] = 0$ for $f < f_{m-1}$ and $f > f_{m+1}$

$B_m[k] = (k - f_{m-1})/(f_m - f_{m-1})$ for $f_{m-1} \le f \le f_m$

$B_m[k] = (f_{m+1} - k)/(f_{m+1} - f_m)$ for $f_m \le f \le f_{m+1}$

$f = k f_s / N$

Process of the Mel-Frequency Cepstrum

summation of the effect inside the m^{th} mask

(1) $x[n] \xrightarrow{FT} X[k]$

(2) $$Y[m] = \log\left\{ \sum_{k=f_{m-1}}^{f_{m+1}} |X[k]|^2 B_m[k] \right\}$$

(3) $$c_x[n] = \frac{1}{M} \sum_{m=1}^{M} Y[m] \cos\left(\frac{\pi n(m-1/2)}{M} \right)$$

Q: What are the difference between the Mel-frequency cepstrum and the original cepstrum?

Advantages : (i) $\sum_{k=f_{m-1}}^{f_{m+1}} |X[k]|^2 B_m[k]$ has much smaller probability to be 0

(ii) $|X[k]|^2$ is real and positive, which avoids phase ambiguity

(iii) cutoff frequencies $f_m = \alpha^m f_0$ match the human hearing system

(iv) Using the cosine transform instead of the IFT to reduce the complexity

Mel-frequency cepstrum 更接近人耳對語音的區別性

用 $c_x[1], c_x[2], c_x[3], \ldots\ldots, c_x[13]$ 即足以描述語音特徵

◉ 5-I References

- R. B. Randall and J. Hee, "Cepstrum analysis," *Wireless World*, vol. 88, pp. 77-80. Feb. 1982
- 王小川，"語音訊號處理"，全華出版，台北，民國94年。
- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, London: Prentice-Hall, 3rd ed., 2010.
- S. C. Pei and S. T. Lu, "Design of minimum phase and FIR digital filters by differential cepstrum," *IEEE Trans. Circuits Syst. I*, vol. 33, no. 5, pp. 570-576, May 1986.
- S. Imai, "Cepstrum analysis synthesis on the Mel-frequency scale," *ICASSP*, vol. 8, pp. 93-96, Apr. 1983.

附錄六：聲音檔和影像檔的處理 (by Matlab)

A. 讀取聲音檔

- 電腦中，沒有經過壓縮的聲音檔都是 *.wav 的型態
  有經過壓縮的聲音檔是*.mp3的型態

- 讀取： audioread
  註：2015版本以後的 Matlab，**wavread** 將改為 **audioread**

- 例： [**x**, **fs**] = audioread(C:\WINDOWS\Media\Alarm01.wav');
  可以將 Alarm01.wav 以數字向量 **x** 來呈現。 **fs**: sampling frequency
  這個例子當中 size(**x**) = 122868 2 fs = 22050

standard:
$f_s = 44100$ Hz

- 思考: 所以，取樣間隔多大?

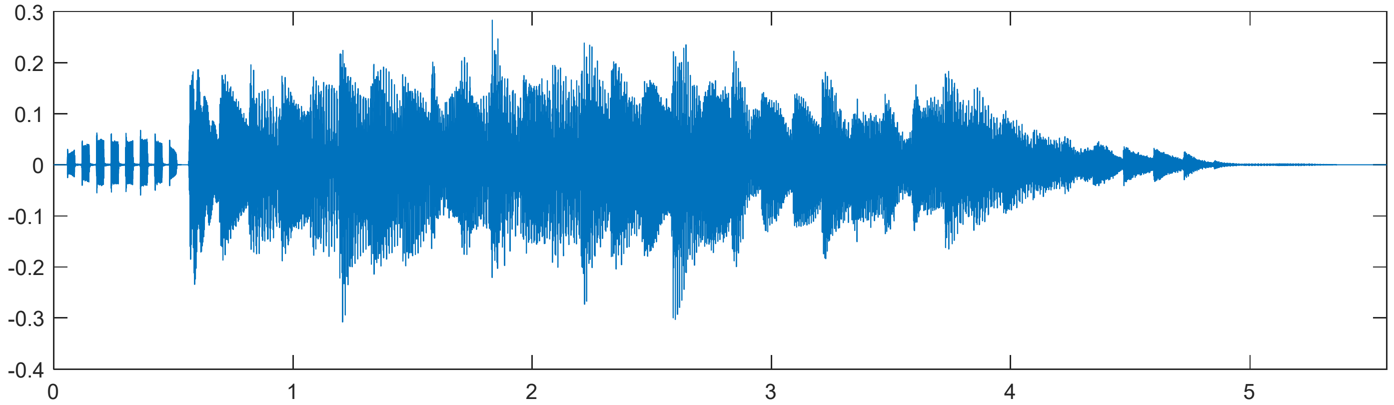
- 這個聲音檔有多少秒？

雙聲道 (**Stereo**，俗稱立體聲)

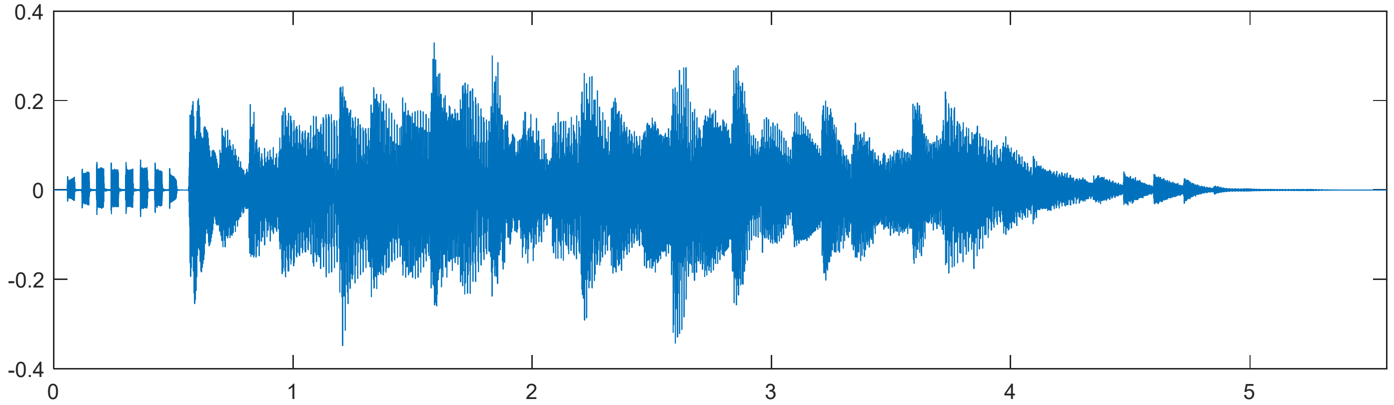
畫出聲音的波型

time = [0:size(x,1)-1]/fs;　 % x 是前頁用 audioread 所讀出的向量

subplot(2,1,1); **plot(time, x(:,1));** xlim([time(1),time(end)])



subplot(2,1,2); **plot(time, x(:,2));** xlim([time(1),time(end)])



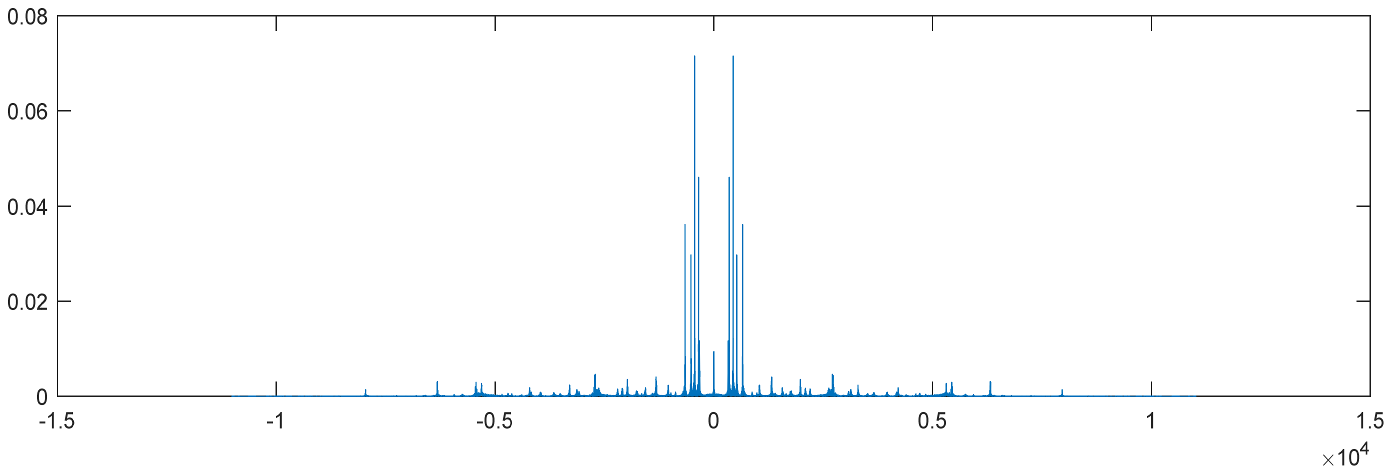
注意： *.wav 檔中所讀取的資料，值都在 −1 和 +1 之間

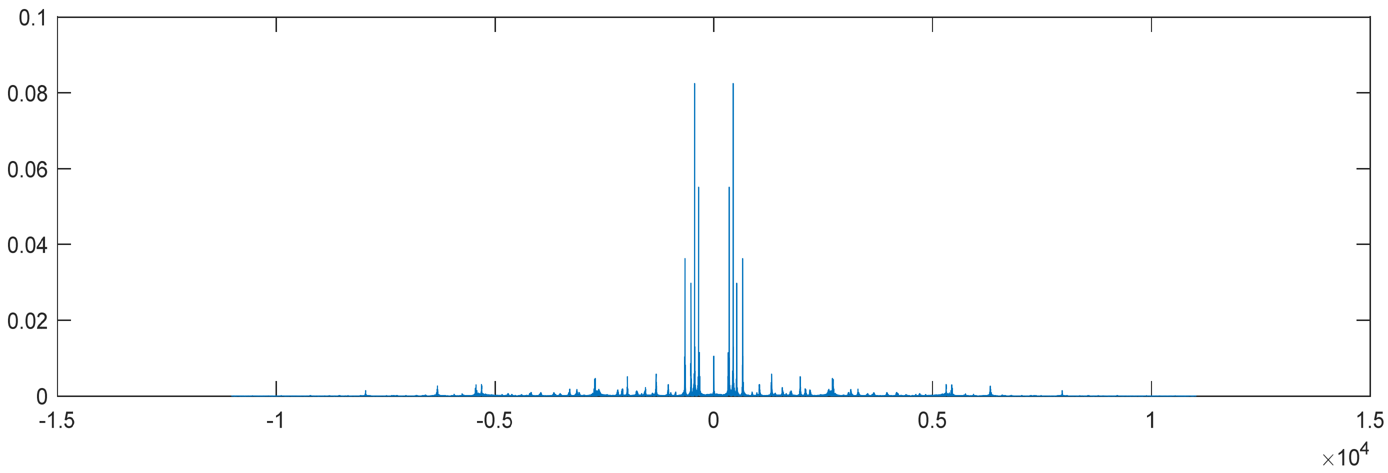
B. 繪出頻譜 (詳細方法請參考附錄二)

X = fft(x(:,1)); % 只做這一步無法得出正確的頻譜

```
X=X.';
N=length(X); N1=round(N/2);
dt=1/fs;
X1=[X(N1+1:N),X(1:N1)]*dt;  % shifting for spectrum
f=[[N1:N-1]-N,0:N1-1]/N*fs;    % valid f
plot(f, abs(X1));
```

Alarm01.wav 的頻譜

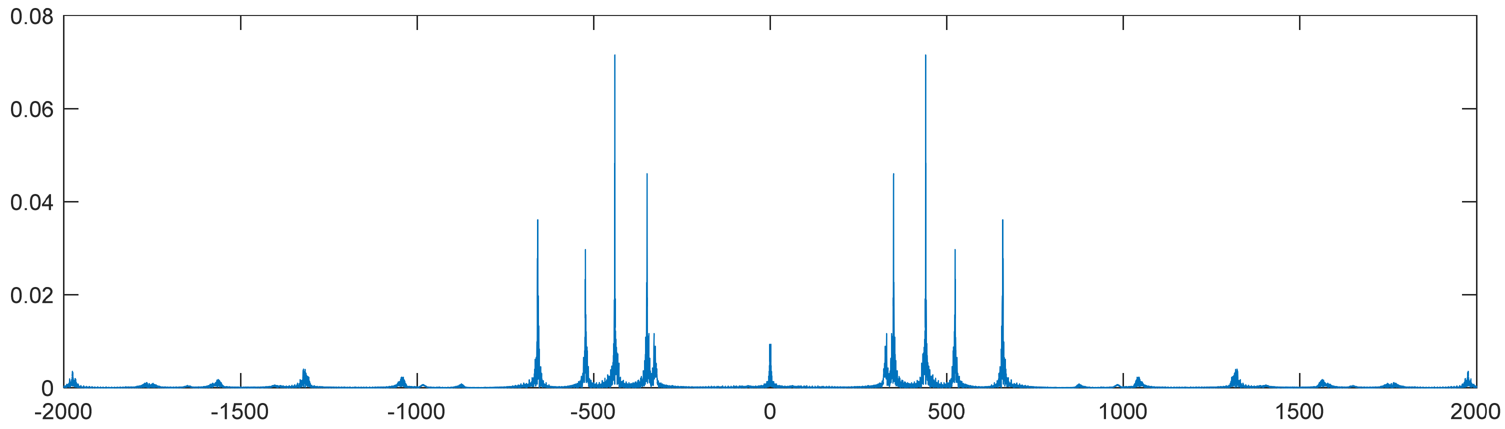


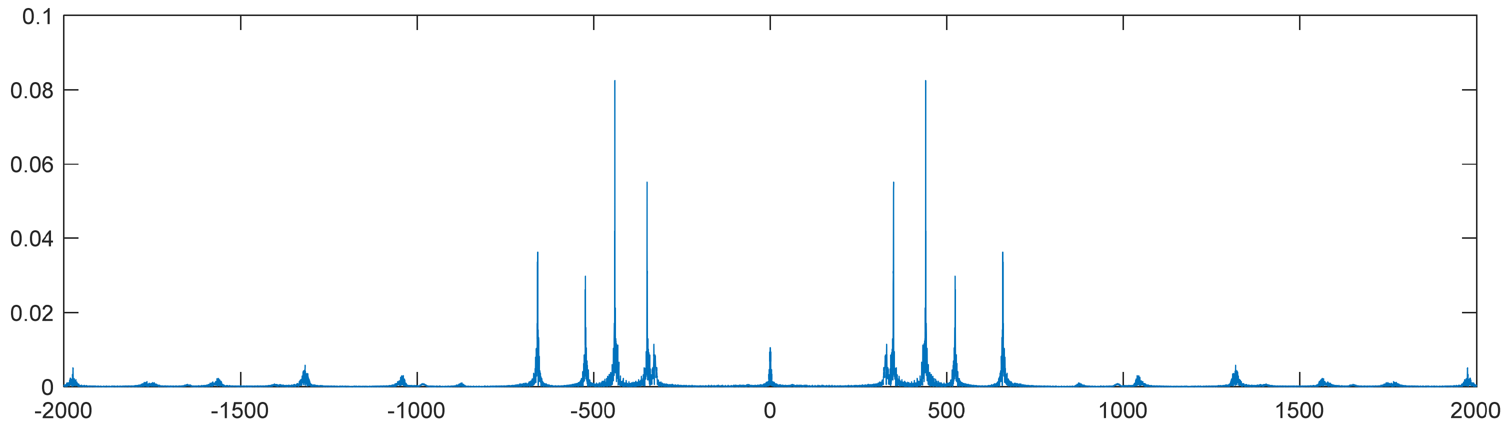


Alarm01.wav 的頻譜

if x[n] is real
then $X(f) = X^*(-f)$

xlim([-2000,2000]) % 只看其中 -2000Hz ~ 2000Hz 的部分





C. 聲音的播放

(1) sound(**x**): 將 **x** 以 8192Hz 的頻率播放

(2) sound(**x, fs**): 將 **x** 以 **fs** Hz 的頻率播放

Note: (1)~(3) 中 **x** 必需是1 個column (或2個 columns)，且 **x** 的值應該 介於 –1 和 +1 之間

(3) soundsc(**x, fs**): 自動把 **x** 的值調到 –1 和 +1 之間 再播放

D. 用 Matlab 製作 *.wav 檔： audiowrite

audiowrite(filename, x, fs)

將數據 **x** 變成一個 *.wav 檔，取樣速率為 **fs** Hz

① **x** 必需是1 個column (或2個 columns) ② **x** 值應該 介於 –1 和 +1

E. 用 Matlab 錄音的方法

錄音之前，要先將電腦接上麥克風，且確定電腦有音效卡

(部分的 notebooks 不需裝麥克風即可錄音)

範例程式：

```
Sec = 3;
Fs = 8000;
recorder = audiorecorder(Fs, 16, 1);
recordblocking(recorder, Sec);
audioarray = getaudiodata(recorder);
```

執行以上的程式，即可錄音。

錄音的時間為三秒，sampling frequency 為 8000 Hz

錄音結果為 audioarray，是一個 column vector (如果是雙聲道，則是兩個 column vectors)

範例程式 (續)：

```
sound(audioarray, Fs);                    % 播放錄音的結果
t = [0:length(audioarray)-1]./Fs;
plot (t, audioarray‘);                    % 將錄音的結果用圖畫出來
xlabel('sec','FontSize',16);
audiowrite(‘test.wav’, audioarray, Fs)    % 將錄音的結果存成 *.wav 檔
```

指令說明：

recorder = audiorecorder(Fs, nb, nch);　(提供錄音相關的參數)

Fs: sampling frequency,

nb: using nb bits to record each data

nch: number of channels (1 or 2)

recordblocking(recorder, Sec);　(錄音的指令)

recorder: the parameters obtained by the command “audiorecorder”

Sec: the time length for recording

audioarray = getaudiodata(recorder);

(將錄音的結果，變成 audioarray 這個 column vector，如果是雙聲道，則 audioarray 是兩個 column vectors)

以上這三個指令，要並用，才可以錄音

F：影像檔的處理

Image 檔讀取: imread

Image 檔顯示: imshow, image, imagesc

Image 檔製作: imwrite

基本概念：灰階影像在 Matlab 當中是一個矩陣

彩色影像在 Matlab 當中是三個矩陣，分別代表 Red, Green, Blue

*.bmp: 沒有經過任何壓縮處理的圖檔

*.jpg: 有經過 JPEG 壓縮的圖檔

Video 檔讀取: aviread

範例一：(黑白影像)

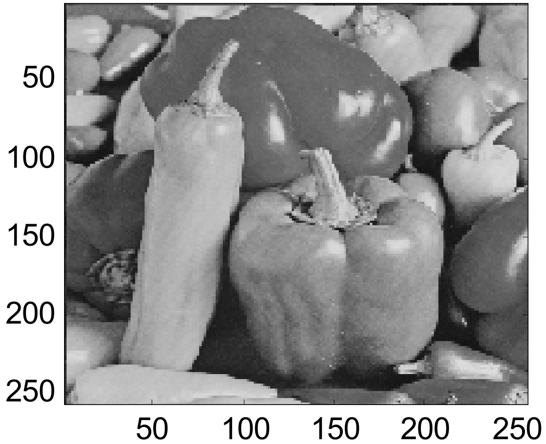
```
im=double(imread('C:\Program Files\MATLAB\pic\Pepper.bmp'));
```

(注意，如果 Pepper.bmp 是個灰階圖，im 將是一個矩陣)

```
size(im)
```
(用 size 這個指令來看 im 這個矩陣的大小)

```
ans =
  256   256

image(im);
colormap(gray(256))
```



範例二：(彩色影像)

```
im2=double(imread('C:\Program Files\MATLAB\pic\Pepper512c.bmp'));

size(im2)
```

(注意，由於這個圖檔是個彩色的，所以 im2 將由三個矩陣複合而成)

```
ans =
  512   512   3
```

```
imshow(im);
```
or
```
image(im/255);
```

注意：要對影像做運算時，<u>要先變成 double 的格式</u>

否則電腦會預設影像為 integer 的格式，在做浮點運算時會產生誤差

例如，若要對影像做 2D Discrete Fourier transform

```
im=imread('C:\Program Files\MATLAB\pic\Pepper.bmp');
im=double(im);
Imf=fft2(im);
```

附錄七 聲音檔和影像檔的處理 (by Python)

可以先安裝幾個模組

```
pip install numpy
pip install scipy
pip install matplotlib  # plot
pip install pipwin
pipwin install simpleaudio  # vocal files
pipwin install pyaudio
```

PS: 謝謝2021年擔任助教的蔡昌廷同學協助製作

A. 讀音訊檔

要先import 相關模組： import scipy.io.wavfile as wavfile

讀取音檔：

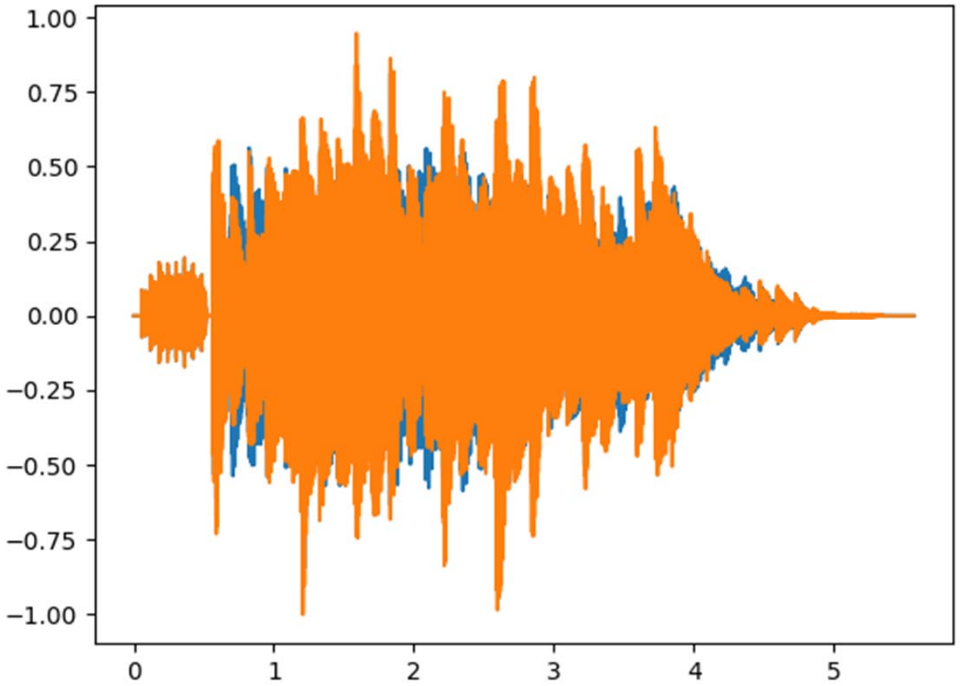
```
fs, wave_data = wavfile.read('C:/WINDOWS/Media/Alarm01.wav')
  # fs: sampling frequency
  # If the audio file has one channel, then wave_data is a column vector
  # If the audio file has two channels, then wave_data has two column
     vectors
num_frame = len(wave_data)   # 音訊長度:
n_channel = int(wave_data.size/ num_frame)   # channels 數量
```

```
>>> fs
22050
>>> num_frame, n_channel
(1150416, 2)
```

畫出音訊波形圖

要先import 相關模組： import matplotlib.pyplot as plt

- time = np.arange(0, num_frame)*1/fs
- plt.plot(time, wave_data)
- plt.show()



B. 畫出頻譜

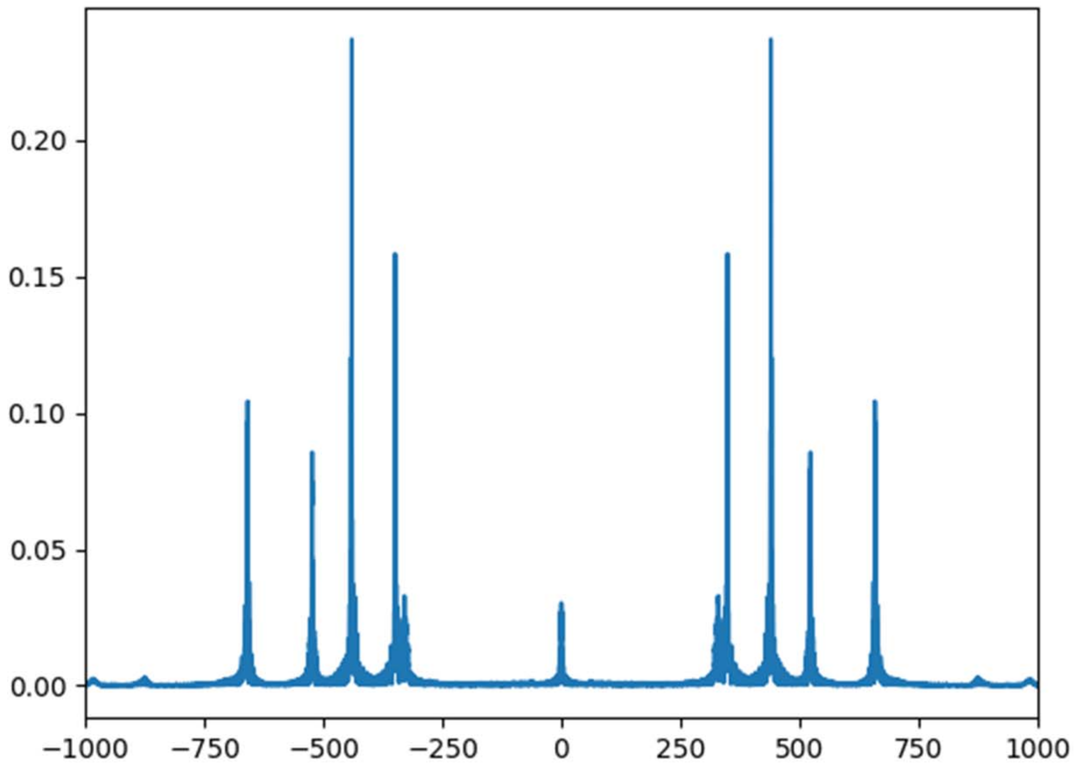
要先import 相關模組： from scipy.fftpack import fft

- fft_data = abs(fft(wave_data[:,1]))/fs   # only choose the 1st channel

  # 注意要乘上 1/fs
- n0=int(np.ceil(num_frame/2))
- fft_data1=np.concatenate([fft_data[n0:num_frame],fft_data[0:n0]])

  # 將頻譜後面一半移到前面
- freq=np.concatenate([range(n0-num_frame,0),range(0,n0)])*fs/num_frame

  # 頻率軸跟著調整
- plt.plot(freq,fft_data1)
- plt.xlim(-1000,1000)   # 限制頻率的顯示範圍
- plt.show() # 如後圖


0.20
0.15
0.10
0.05
0.00
−1000
−750
−500
−250
0
250
500
750
1000

C. 播放聲音

要先import 相關模組： import simpleaudio as sa

- n_bytes =2 # using two bytes to record a data
- wave_data = (2**15-1)* wave_data

  \# change the range to $-2^{15} \sim 2^{15}$
- wave_data = wave_data.astype(np.int16)
- play_obj = sa.play_buffer(wave_data, n_channel, n_bytes, fs)
- play_obj.wait_done()

D. 製作音檔

- wavfile.write(file name, fs, data)

  # fs means the sampling frequency

  # data should be a one-column or two column array

Example:

- wavfile.write('Alarm01_test.wav’, 22050, wave_data)

E. 錄音

要先import 相關模組： import pyaudio

範例程式

```
import pyaudio
pa=pyaudio.PyAudio()
fs = 44100
chunk = 1024
stream = pa.open(format=pyaudio.paInt16, channels=1,
rate=fs, input=True, frames_per_buffer=chunk)

vocal=[]
count=0
```

```
while count<200: #控制錄音時間
    audio = stream.read(chunk)  #一次性錄音取樣位元組大小
    vocal.append(audio)
    count +=1

save_wave_file('testrecord.wav',vocal)
stream.close()
```

參考
https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/491427/

F. 影像檔的處理

可以先安裝幾個模組

pip install numpy

pip install matplotlib

1. 讀取影像檔

```
import cv2
image = cv2.imread('D:/Pic/peppers.bmp’)
```

或

```
import matplotlib.pyplot as plt
image = plt.imread(‘D:/Pic/peppers.bmp’)
```

注意

(1)寫入圖片若為彩色圖片，需要注意 cv2.imread 預設channels 順序為 BGR,

image[:, :, 0] => B, image[:, :, 1] => G, image[:, :, 2] => R

(2) 若使用 plt.imread, 則 3 個 channels 的順序仍為 RGB

image[:, :, 0] => R, image[:, :, 1] => G, image[:, :, 2] => B

(3) 若讀檔讀不出來，有時要將路徑的 \ 改為 /

(4) image.shape 可以看出影像之大小

```
>>> image.shape
(512, 512, 3)
```

(5) 可讀 jpg, bmp, png 檔，但不能讀 gif 檔

2. 顯示影像

Case 1: 圖的值格式為 int

以下的指令要配合使用 (彩色，灰階皆可)

```
cv2.imshow('test', image)  # 以 test 為顯示的圖的名稱
cv2.waitKey(0)
cv2.destroyAllWindows()
```

亦可用以下之指令

```
import matplotlib.pyplot as plt
plt.imshow(image)
plt.show()
```

若一開始用 cv.imread 讀圖但要用 plt.imshow來顯示彩色圖，要先將 BGR 的順序轉回 RGB，將第二行改為

```
plt.imshow(image[:,:,[2,1,0]])
```

Case 2: 圖的值格式為 double (非整數)

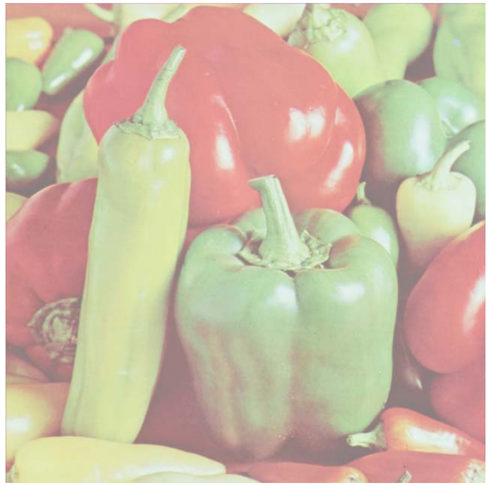
此時，無論用 cv2.imshow或是 plt.imshow，皆要先除以255，再將圖顯示出來

Example 1:

```
image = cv2.imread('D:/Pic/peppers.bmp’)
Image1 = image*0.5 + 127.5  # lighten the image
cv2.imshow(‘test’, image) # int 不用除255
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imshow(‘test’, image1/255) # 非整數要除255
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Example 2:

```
import matplotlib.pyplot as plt
image = plt.imread('D:/Pic/peppers.bmp’)
image1 = image*0.5 + 127.5  # lighten the image
plt.imshow(image) # int 不用除255
plt.show()
plt.imshow(image1/255) # 非整數要除255
plt.show()
```



3. 寫入圖片檔

cv2.imwrite('D:/Pic/jpg', image)

或

plt.imsave('D:/Pic/jpg', image)

注意

(1) 寫入圖片若為彩色圖片，在使用 cv2.imwrite 時需要注意

image[:, :, 0] => B, image[:, :, 1] => G, image[:, :, 2] => R

若用 plt.imsave 則

image[:, :, 0] => R, image[:, :, 1] => G, image[:, :, 2] => B

(2) 若用 cv2.imwrite('D:\Pic\jpg', image) 可能無法存檔，要將 \ 改為 /

(3) 若是使用 plt.imshow 和 plot.show() 來顯示，可以用右下角的 "save the figure" 來存檔

VI. Brief Introduction for Acoustics

(聲學)

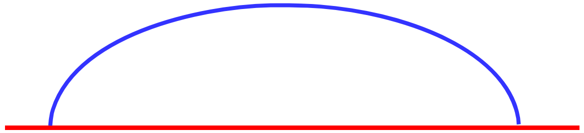
[參考資料]

- 王小川，“語音訊號處理”，第三版，全華出版，台北，民國98年。
- T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principle and Practice*, Pearson Education Taiwan, Taipei, 2005.
- L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.
- P. Filippi, *Acoustics : Basic Physics, Theory, and Methods*, Academic Press, San Diego, 1999.

◉ 6-A 聲音的相關常識

人耳可以辨識頻率：20Hz ~ 20000Hz

說話：150~2000Hz >2000Hz 倍頻 子音 氣音



電話系統頻域：小於 4000Hz

電腦音效卡取樣頻率：44100Hz (最新技術可達192K)

(一般用 22050Hz, 11025Hz 即可)

music file

250 sec × 22050 × 2 × 8/8 bytes

= 11 M

MP3 : 3M ~ 4M

30%

> 20000Hz: 超音波 (ultrasound)

< 20Hz: 次聲波 (infrasound)

波長較長 -> 傳播距離較遠，但容易散射

波長較短 -> 衰減較快，但傳播方向較接近直線

正比於 e^{-kz} $k = \frac{2\pi}{\lambda}$

- 一般聲音檔格式：
  - (1) 取樣頻率 22050Hz
  - (2) 單聲道或雙聲道
  - (3) 每筆資料用8個bit來表示

- 電腦中沒有經過任何壓縮的聲音檔： *.wav

Q: What is the data size of a song without compression?

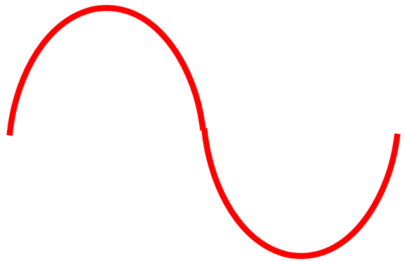
- 數位電話取樣頻率：8000Hz

聲音在空氣中傳播速度 ： 每秒 340 公尺 (15°C 時)
所以，人類對3000Hz 左右頻率的聲音最敏感

$\lambda/4$

(一般人，耳翼到鼓膜之間的距離： 2.7公分)

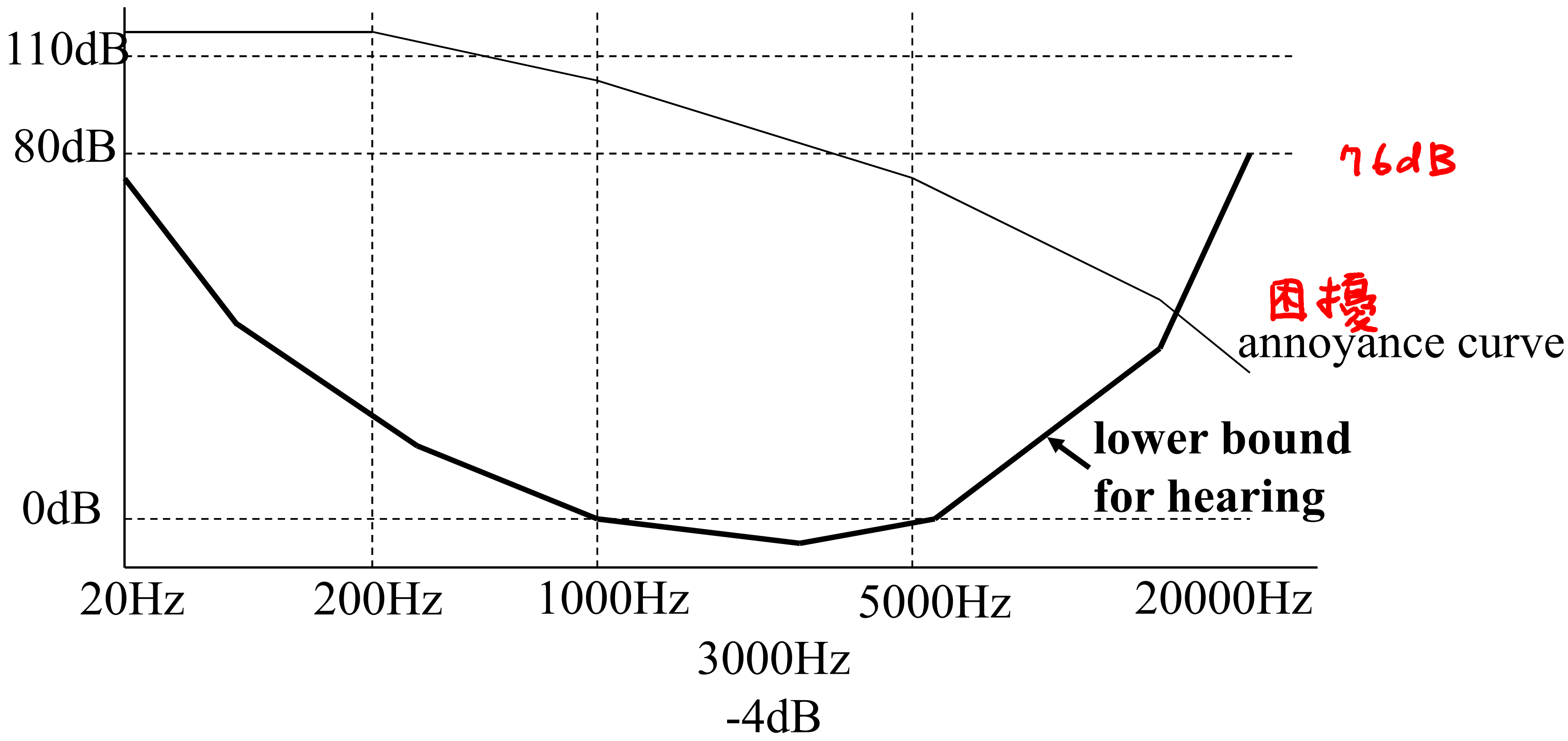
$\lambda : 0.108$ m



附： (1) 每增加 1°C，聲音的速度增加 0.6 m/sec

(2) 聲音在水中的傳播速度是 1500 m/sec

在鋁棒中的傳播速度是 5000 m/sec



110dB
80dB
0dB
76dB
困擾
annoyance curve
lower bound
for hearing
20Hz
200Hz
1000Hz
5000Hz
20000Hz
3000Hz
-4dB

- dB: 分貝　$10\log_{10}(P/C)$， 其中P為音強(正比於振福的平方)；C為0dB時的音強

每增加 10dB，音強增加10倍，振幅增加 $10^{0.5}$ 倍；
每增加3dB ，音強增加2倍，振幅增加 $2^{0.5}$ 倍；
所幸，內耳的振動不會正比於聲壓

- 人對於頻率的分辨能力，是由頻率的「比」決定

對人類而言，300Hz 和 400 Hz 之間的差別，與 3000Hz 和 4000 Hz 之間的差別是相同的

acoustics
- music
- speech
- others

◉ 6-B Music Signal

電子琴 Do 的頻率：低音 Do: 131.32 Hz
中音 Do: 261.63 Hz
高音 Do: 523.26 Hz
更高音 Do: 1046.52 Hz, …….

音樂每增加八度音，頻率變為 2 倍

If Do : 200 Hz
then Re $200 \times 2^{\frac{2}{12}}$ Hz
Mi : $200 \times 2^{\frac{4}{12}}$ Hz

每一音階有12個半音

增加一個半音，頻率增加 $2^{1/12}$ 倍 (1.0595 倍)

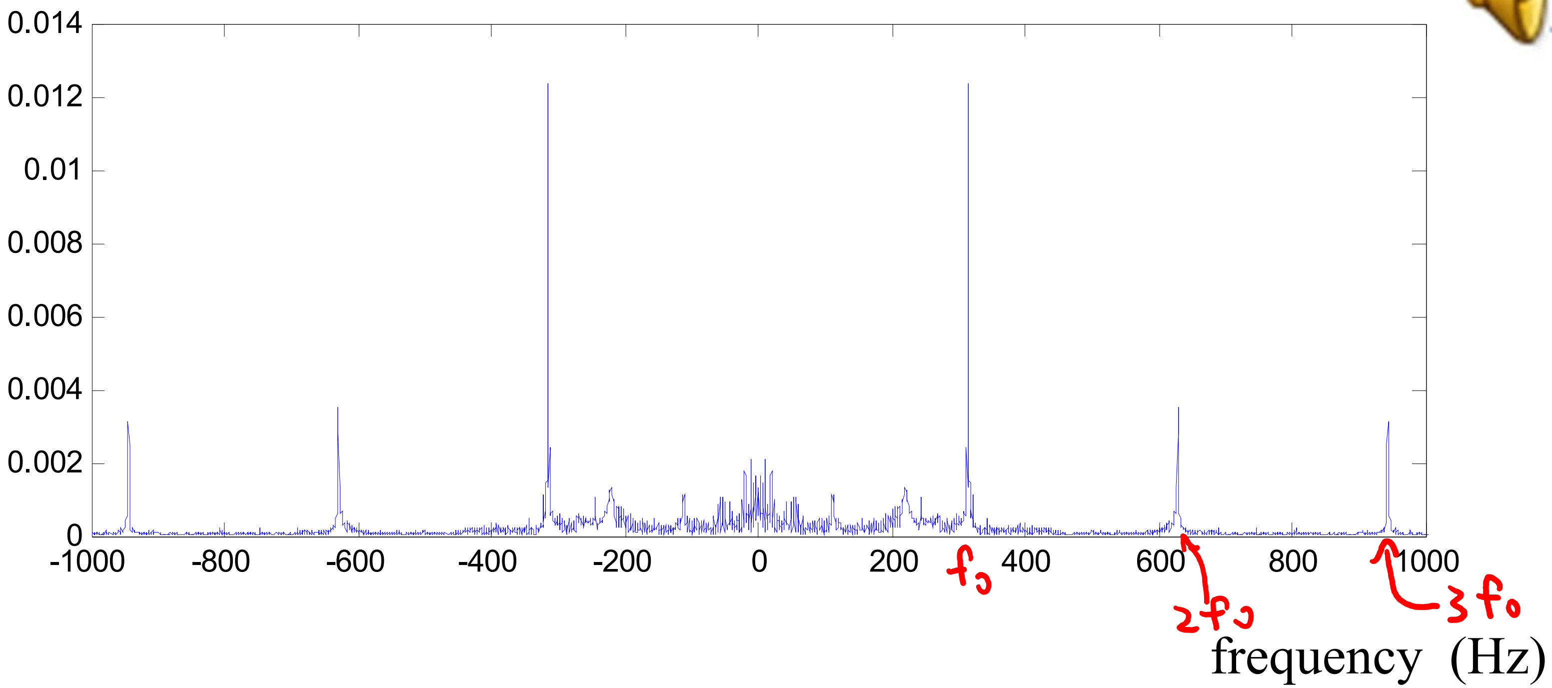
	Do	升Do	Re	升Re	Mi	Fa	升Fa	So	升So	La	升La	Si
Hz	262	277	294	311	330	349	370	392	415	440	466	494

$261.63 \times 2^{\frac{2}{12}}$ $261.63 \times 2^{\frac{4}{12}}$

音樂通常會出現「和弦」(chord) 的現象

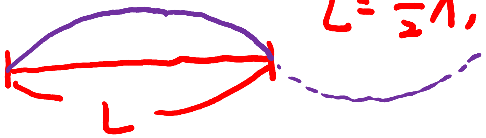
除了基頻 f_0 Hz 之外，也會出現 $2f_0$ Hz, $3f_0$ Hz, $4f_0$ Hz , ….. 的頻率

Mi : 330

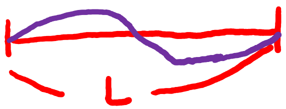


為什麼會產生和弦？

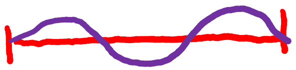
以共振的觀點：



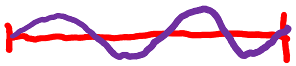
$L=\frac{1}{2}\lambda,\ \lambda=2L$ $\quad f=\frac{340}{2L}$ $\quad$ set $f_0=\frac{340}{2L}$



$L=\lambda$ $\quad f=\frac{340}{L}=2f_0$



$L=\frac{3}{2}\lambda,\ \lambda=\frac{2}{3}L$ $\quad f=\frac{3}{2}\frac{340}{L}=3f_0$



$L=2\lambda,\ \lambda=\frac{L}{2},\ f=2\frac{340}{L}=4f_0$

$\vdots$

$L=\frac{n}{2}\lambda,\ \lambda=\frac{2L}{n},\ f=\frac{n}{2}\frac{340}{L}=nf_0$

聲音信號是一個 <u>periodic signal</u>，但是<u>不一定是 sinusoid</u>

⦿ 6-C 語音處理的工作

(1) 語音編碼 (Speech Coding)

(2) 語音合成 (Speech Synthesis)

(3) 語音增強 (Speech Enhancement)

前三項目前基本上已經很成功

(4) 語音辨認 (Speech Recognition)

音素 → 音節 → 詞 → 句 → 整段話

目前已有很高的辨識率

(5) 說話人辨認 (Speaker Recognition)

(6) 其他：語意，語言，情緒

⦿ 6-D 語音的辨認

音素 → 音節 → 詞 → 句 → 整段話

音素：相當於一個音標

(1) Spectrum Analysis

Time-Frequency Analysis

(2) Cepstrum

(3) Correlation for Words

◉ 6-E 子音和母音

ㄅ ㄆ ㄇ ㄈ ㄉ ㄊ ㄋ ㄌ ㄍ ㄎ ㄏ ㄐ ㄑ ㄒ ㄓ ㄔ ㄕ ㄖ ㄗ ㄘ ㄙ
ㄚ ㄛ ㄜ ㄝ ㄞ ㄟ ㄠ ㄡ ㄢ ㄣ ㄤ ㄥ ㄦ ㄧ ㄨ ㄩ

母音： ㄚ ㄛ ㄜ ㄝ ㄞ ㄟ ㄠ ㄡ ㄢ ㄣ ㄤ ㄥ ㄦ ㄧ ㄨ ㄩ

- 單母音： a, e, i, o, u ㄚ ㄛ ㄜ ㄝ ㄦ ㄧ ㄨ ㄩ
- 雙母音：ㄞ ㄟ ㄠ ㄡ
- 母音＋濁音：ㄢ ㄣ ㄤ ㄥ

子音： ㄅ ㄆ ㄇ ㄈ ㄉ ㄊ ㄋ ㄌ ㄍ ㄎ ㄏ ㄐ ㄑ ㄒ ㄓ ㄔ ㄕ ㄖ ㄗ ㄘ ㄙ

	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ	ㄑ	ㄒ
漢語拚音	b	p	m	f	d	t	n	l	g	k	h	j	q	x
通用拚音	b	p	m	f	d	t	n	l	g	k	h	j	c	s

	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ	ㄠ
漢語拚音	zh	ch	sh	r	z	c	s	a	o	e	e	ai	ei	ao
通用拚音	jh	ch	sh	r	z	c	s	a	o	e	e	ai	ei	ao

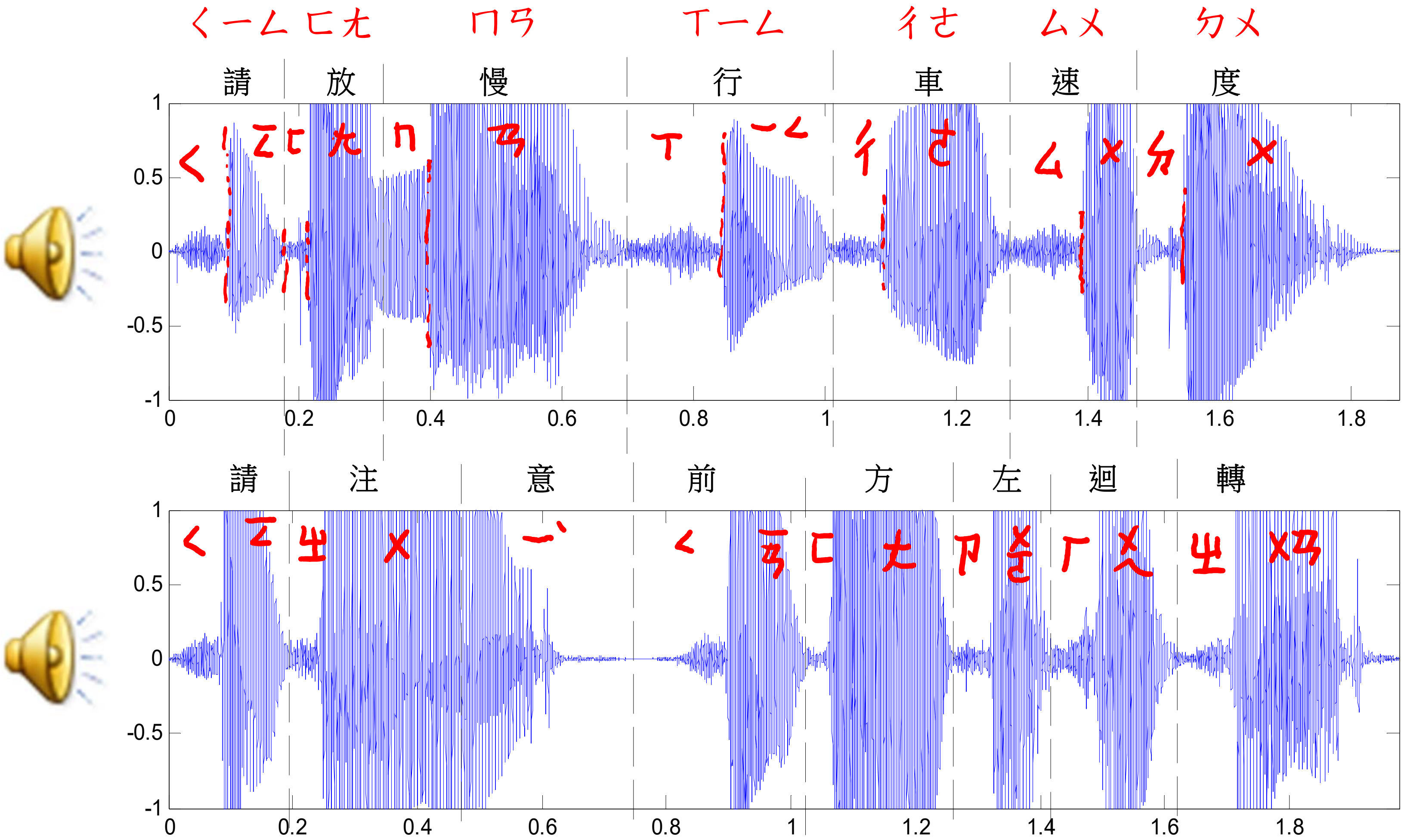
	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ	ㄧ	ㄨ	ㄩ
漢語拚音	ou	an	en	ang	eng	er	i, y	u, w	yu, iu
通用拚音	ou	an	en	ang	eng	er	i, y	u, w	yu, iu

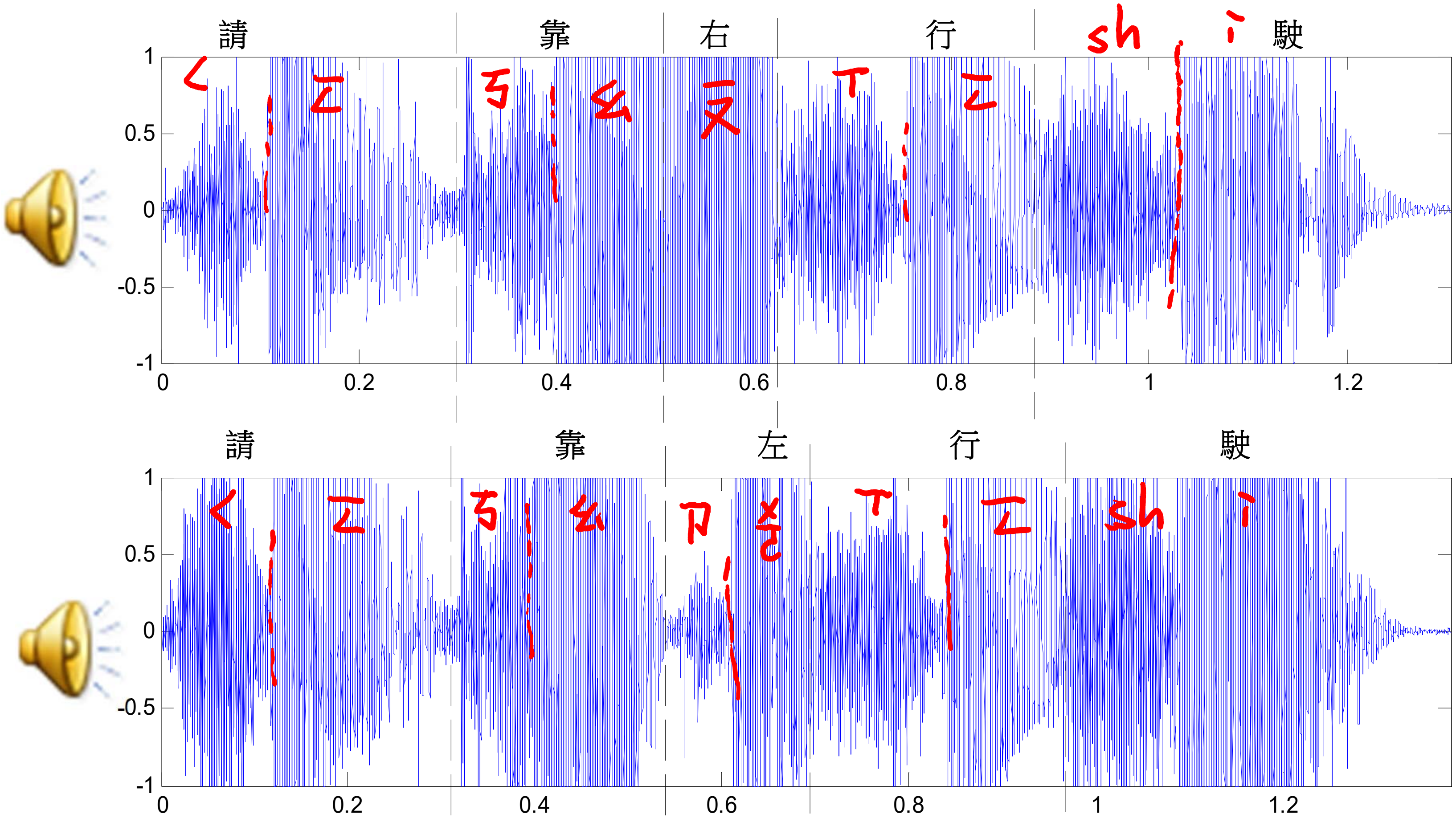
母 音： 依唇型而定

子 音： 在口腔，鼻腔中某些部位將氣流暫時堵住後放開

子音的能量小，頻率偏高，時間較短，出現在母音前

母音的能量大，頻率偏低，時間較長，出現在子音後或獨立出現


ㄑㄧㄥ ㄈㄤ ㄇㄢ ㄒㄧㄥ ㄔㄜ ㄙㄨ ㄉㄨ
請 放 慢 行 車 速 度
1
0.5
0
-0.5
-1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8
請 注 意 前 方 左 迴 轉
1
0.5
0
-0.5
-1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8



請
靠
右
行
駛
請
靠
左
行
駛

發音模型 (線性非時變近似)

$x[n] = e_p[n] * g[n] * h[n] * r[n]$, $*$ means the convolution

$X(z) = E_p(z)\ G(z)\ H(z)\ R(z)$

$r[n]$ ：嘴唇模型， $h[n]$: 口腔模型， $g[n]$ ：聲帶模型

$e_p[n]$ ：輸入(假設為週期脈衝)

音量和 $e_p[n], g[n]$ 有關

頻率和 $g[n]$ 有關

子音和 $h[n], r[n]$有關   de → te → tea → tea

母音和 $r[n]$有關

- 分析一個聲音信號的頻譜：

  用**Windowed Fourier Transform**

  或稱作 **Short-Time Fourier Transform**

- Fourier transform

$$G(f)=\int_{-\infty}^{\infty} g(t)e^{-j2\pi f t}dt$$

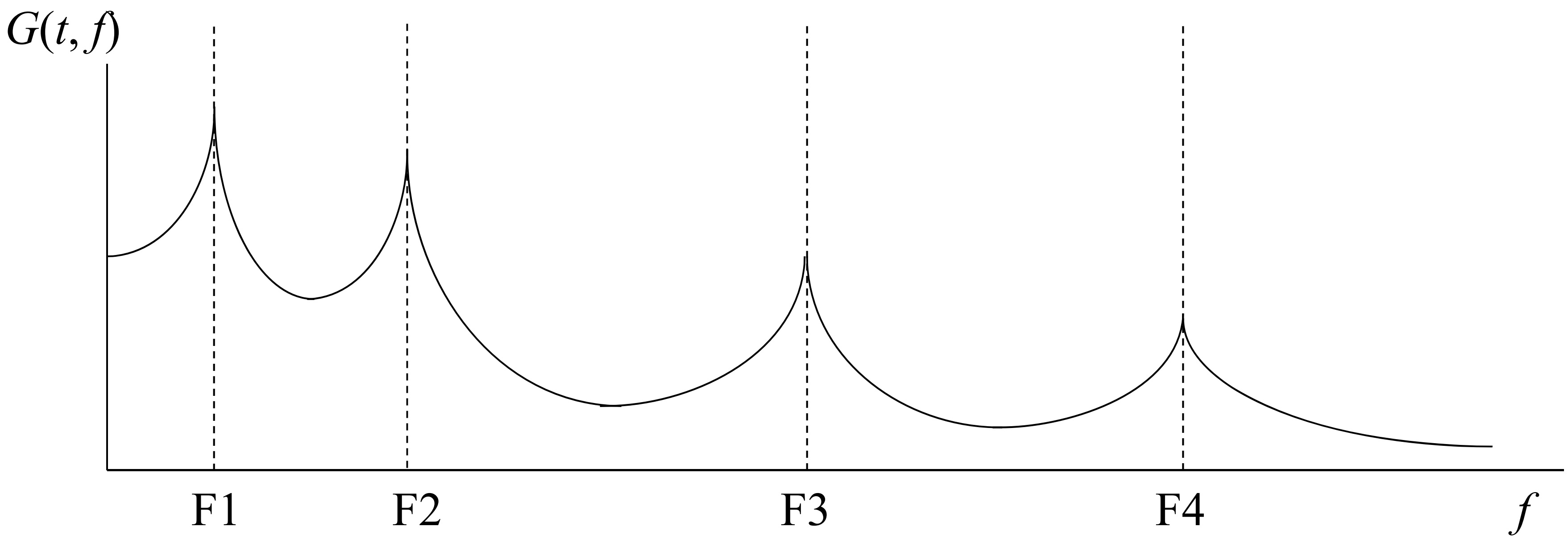
Windowed Fourier transform

$$G(f)=\int_{t_0-B}^{t_0+B} g(t)e^{-j2\pi f t}dt$$

強調 $t=t_0$ 附近的區域

或 $$G(t,f)=\int_{-\infty}^{\infty} w(t-\tau)g(\tau)e^{-j2\pi f \tau}d\tau$$

典型的聲音頻譜(不考慮倍頻)：



頻譜上，大部分的地方都不等於0。

出現幾個 peaks 值

可以依據 peaks 的位置來辨別母音

母音 peaks 處的頻率 (Hz) (不考慮倍頻)：

	男聲			女聲		
	F1	F2	F3	F1	F2	F3
ㄚ	900	1200	2900	1100	1350	3100
ㄛ	560	800	3000	730	1100	3200
ㄜ	560	1090	3000	790	1250	3100
ㄝ	500	2100	3100	600	2400	3300
ㄧ	310	2300	3300	360	3000	3500
ㄨ	370	540	3400	460	820	3700
ㄩ	300	2100	3400	350	2600	3200
ㄦ	580	1500	3200	760	1700	3200

原則上： (1) 嘴唇的大小，決定F1

(2) 舌面的高低，決定 F2 – F1

[Ref] 王小川，“語音訊號處理”，第三版，全華出版，台北，民國98年

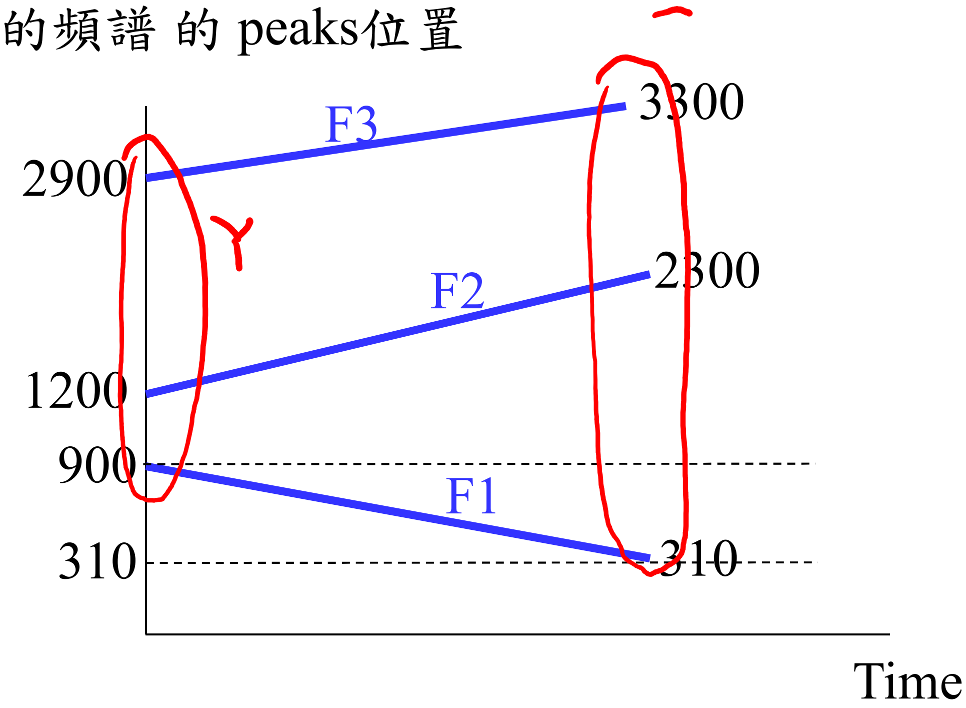
- 雙母音：

ㄚㄧ　ㄝㄧ　ㄚㄛ　ㄛㄨ

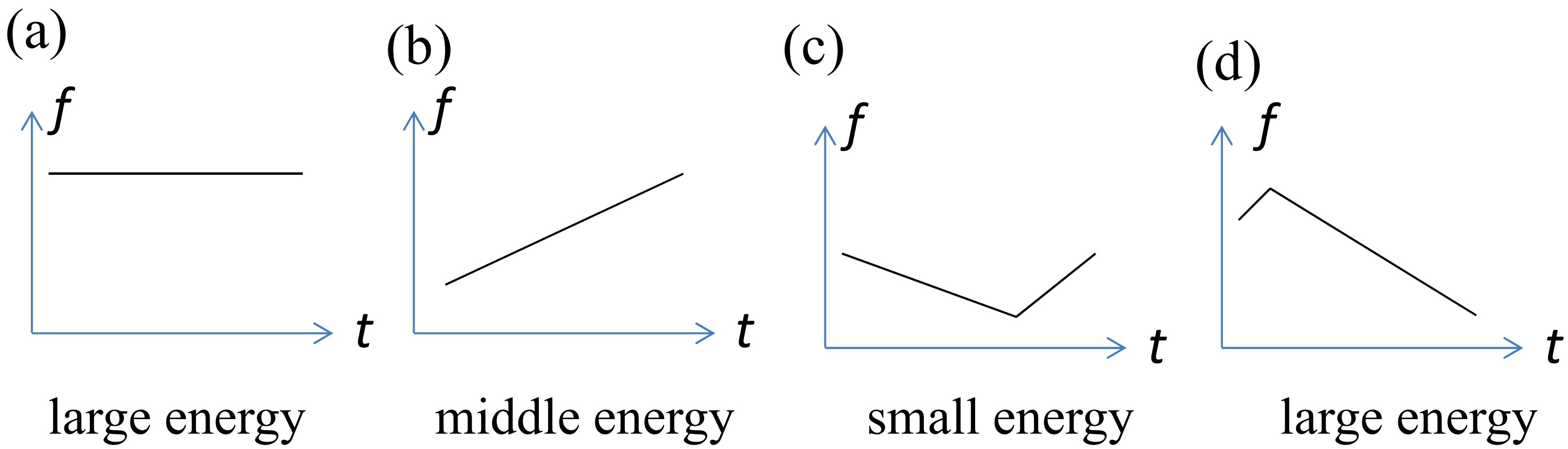
ㄞ (ai)，　ㄟ (ei)，　ㄠ (ao)，　ㄡ (ou)

頻譜隨時間而改變，一開使始像第一個母音，後變得像另一個母音

ㄞ 的頻譜 的 peaks位置



◉ 6-F Tone Analysis



Typical relations between time and the instantaneous frequencies for (a) the 1st tone, (b) the 2nd tone, (c) the 3rd tone, and (d) the 4th tone in Chinese.

X. X. Chen, C. N. Cai, P. Guo, and Y. Sun, "A hidden Markov model applied to Chinese four-tone recognition," *ICASSP*, vol. 12, pp. 797-800, 1987.

◉ 6-G 語意學的角色

以「語意學」或「機率」來補足語音辨識的不足

例如：經過判定，一個聲音可能是

ㄅㄧ ㄖㄢ　　ㄆㄧ ㄖㄢ

ㄅㄧ ㄌㄢ　　ㄆㄧ ㄌㄢ

這個聲音是「必然」的機率比較大。

ㄅㄛ ㄅㄛ　　ㄆㄛ ㄆㄛ

可能是「伯伯」，也可能是「婆婆」，看上下文

儲存詞庫

- 當前主流的語音辨識技術：

Mel-Frequency Cepstrum + Tone Analysis + 語意分析 + Machine Learning

附錄八：線性代數觀念補充

(1) $\mathbf{x}$ 和 $\mathbf{y}$ 兩個向量的內積可表示成 $<\mathbf{x}|\mathbf{y}>$

(2) 兩個互相正交(orthogonal)或垂直(perpendicular)的向量，其內積為0。
可表示成:$<\mathbf{x}|\mathbf{y}>=0$ 或 $<\mathbf{x},\mathbf{y}>=0$

(3) 令 S 為內積空間V的一組正交集合(set)且由非零向量構成，

其中 $$\mathbf{x}=\sum_{\mathbf{y}\in S} a_y \mathbf{y}, \quad a_y=\frac{<\mathbf{x}|\mathbf{y}>}{<\mathbf{y}|\mathbf{y}>}$$

如果 S 是由一組正規集合(orthonormal set)構成，那麼 $a_y=<\mathbf{x}|\mathbf{y}>$

(4) Gram-Schmidt algorithm: 對於內積空間V的任意一組基底 $<\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_n}>$

，我們可以透過這演算法找到一組正交基底 $<\mathbf{y_1},\mathbf{y_2},\ldots,\mathbf{y_n}>$

$$\mathbf{y_j}=\mathbf{x_j}-\sum_{i=1}^{j-1}\frac{<\mathbf{x_j}\,|\,\mathbf{y_i}>}{<\mathbf{y_i}\,|\,\mathbf{y_i}>}\mathbf{y_i}\quad for\ \ each\ j=2,\ldots,n$$

幾何意義:把 $\mathbf{x_j}$ 在 $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_{j-1}}$上面的分向量全都從向量 $\mathbf{x_j}$ 身上扣掉之後，剩下的向量 $\mathbf{y_j}$自然就會跟 $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_{j-1}}$ 垂直。

(5) Solving $\mathbf{Ax}=\mathbf{b}$ but $size(\mathbf{A})=m\times n\ \ and\ \ \mathbf{b}\in F^m$, $m>n$

Interpolation Theorem (插值定理)

1. For any inner-product function of F^m , there exists a vector $\mathbf{z}$ that minimizes $\|\mathbf{Az}-\mathbf{b}\|$ where $\mathbf{z}\in F^n$

2. If rank($\mathbf{A}$) = n, then $\mathbf{z}=\left(\mathbf{A}^H\mathbf{A}\right)^{-1}\mathbf{A}^H\mathbf{b}$ is the unique minimizer of $\|\mathbf{Az}-\mathbf{b}\|$

附錄九：PCA and SVD

PCA (principal component analysis) 是資料分析和影像處理當中常用到的數學方法，用來分析資料的「主要成分」或是影像中物體的「主軸」。

它其實和各位同學在高中和大一線代所學的回歸線 (regressive line) 很類似。回歸線是用一條一維 (one-dimensional) 的直線來近似二維 (two-dimensional) 的資料，而 PCA 則是用 M-dimensional data 來近似 N-dimensional data ，其中 M 小於等於 N

在講解 PCA 之前，先介紹什麼是 SVD (singular value decomposition)

我們在大一的時候，都已經學到該如何對於 N x N 的矩陣做 eigenvector -eigenvalue decomposition

那麼………

<u>當一個矩陣的 size 為 M x N，且 M 和 N 不相等時，我們該如何對它來做 eigenvector-eigenvalue decomposition?</u>

SVD 的流程：

假設 **A** 是一個 $M \times N$ 的矩陣。

(Step 1) 計算

$$\mathbf{B} = \mathbf{A}^{H}\mathbf{A} \qquad \mathbf{C} = \mathbf{A}\mathbf{A}^{H}$$

注意，**B** 是 $N \times N$ 的矩陣，而 **C** 是 $M \times M$ 的矩陣。上標H代表 Hermitian matrix，相當於做共軛轉置。

(Step 2) 接著，對 **B** 和 **C** 做 eigenvector-eigenvalue decomposition

$$\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \qquad \mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

其中 **V** 的每一個 column 是 **B** 的 eigenvector (with normalization)，**U** 的每一個 column 是 **C** 的 eigenvector (with normalization)，**Λ** 和 **D** 都是對角矩陣，**Λ** 和 **D** 對角線上的 entries 是 **B** 和 **C** 的 eigenvalues。並假設 eigenvectors 根據 eigenvalues 的大小排序 (由大到小)

Note: 值得注意的是，由於 $\mathbf{B} = \mathbf{B}^{H}$ 且 $\mathbf{C} = \mathbf{C}^{H}$，所以 **B** 和 **C** 的 eigenvectors 皆各自形成一個 orthogonal set。經過適當的 normalization 使得 **U** 和 **V** 的 column 自己和自己的內積為 1 之後，$\mathbf{U}^{-1} = \mathbf{U}^{H}$ 和 $\mathbf{V}^{-1} = \mathbf{V}^{H}$將滿足。因此，**B** 和 **C** 可以表示成

$$\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^{H} \qquad \mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{H}$$

注意，**V**和**U**是unitary matrix

(Step 3) 計算

$$\mathbf{S_1} = \mathbf{U^H A V}$$

$\mathbf{S_1}$ 是一個 M x N 的矩陣，只有在 $\mathbf{S_1}[n, n]$ (n = 1, 2, …, min(M, N)) 的地方不為 0

(Step 4) $\mathbf{S} = |\mathbf{S_1}|$ 取絕對值

若 $\boldsymbol{S}_1[n, n] < 0$，改變 **U** 第 n 個 column 的正負號

即完成 SVD

$$\mathbf{A} = \mathbf{USV^H}$$

Note: Since **V** is bound to be real,

$$\mathbf{A} = \mathbf{USV^T}$$

A 也可以表示為

$$\mathbf{A} = \lambda_1 \mathbf{u_1 v_1^T} + \lambda_2 \mathbf{u_2 v_2^T} + \cdots\cdots + \lambda_k \mathbf{u_k v_k^T}$$

其中 $\lambda_n = \boldsymbol{S}[n, n]$, $k = \min(M, N)$)

註： Matlab 有內建的 svd 指令可以計算 SVD

從 SVD 到 PCA (principal component analysis ，主成份分析)

$$\mathbf{A} = \lambda_1 \mathbf{u_1} \mathbf{v_1^T} + \lambda_2 \mathbf{u_2} \mathbf{v_2^T} + \cdots\cdots + \lambda_k \mathbf{u_k} \mathbf{v_k^T} \qquad k = \min(M, N))$$

若 $\lambda_1 \geqq \lambda_2 \geqq \lambda_3 \geqq \ldots.. \geqq \lambda_k$

$\lambda_1 \mathbf{u_1} \mathbf{v_1^T}$ 是 A 矩陣的最主要的成份

$\lambda_2 \mathbf{u_2} \mathbf{v_2^T}$ 是 A 矩陣的第二主要的成份

:

:

$\lambda_k \mathbf{u_k} \mathbf{v_k^T}$ 是 A 矩陣的最不重要的成份

若為了壓縮或是去除雜訊的考量，可以選擇 $h < k$，使得 **A** 可以近似成

$$\mathbf{A} \cong \lambda_1 \mathbf{u_1} \mathbf{v_1^T} + \lambda_2 \mathbf{u_2} \mathbf{v_2^T} + \cdots\cdots + \lambda_h \mathbf{u_h} \mathbf{v_h^T}$$

PCA 的流程

假設現在有 M 筆資料，每一筆資料 為 N dimension

$$\mathbf{g_1} = [f_{1,1}\ f_{1,2},\ \ldots, f_{1,N}]$$
$$\mathbf{g_2} = [f_{2,1}\ f_{2,2},\ \ldots, f_{2,N}]$$
$$\vdots$$
$$\mathbf{g_M} = [f_{M,1}\ f_{M,2},\ \ldots, f_{M,N}]$$

(Step 1) 扣掉平均值，形成新的 data

$$\mathbf{d_m} = \begin{bmatrix} e_{m,1} & e_{m,2} & \cdots\cdots & e_{m,N} \end{bmatrix} \qquad m = 1, 2, \ldots, M$$

其中 $e_{m,n} = f_{m,n} - \tilde{f}_n, \qquad \tilde{f}_n = \dfrac{1}{M}\sum_{m=1}^{M} f_{m,n}$

(Step 2) 形成 M x N 的矩陣 **A**

A 的第 m 個 row 為 d_m , $\quad m = 1, 2, \ldots, M$

(Step 3) 對 A 做 SVD 分解

$$\mathbf{A} = \mathbf{USV}^{\mathbf{H}}$$
$$= \lambda_1 \mathbf{u_1 v_1^T} + \lambda_2 \mathbf{u_2 v_2^T} + \cdots\cdots + \lambda_k \mathbf{u_k v_k^T} \qquad k = \min(M, N))$$

$$\lambda_1 \geqq \lambda_2 \geqq \lambda_3 \geqq \ldots\ldots \geqq \lambda_k$$

(Step 4) 將 A 近似成

$$\mathbf{A} \cong \lambda_1 \mathbf{u_1 v_1^T} + \lambda_2 \mathbf{u_2 v_2^T} + \cdots\cdots + \lambda_h \mathbf{u_h v_h^T}$$

則每一筆資料可以近似為

$$g_{\mathbf{m}} \cong \lambda_1 u_1[m]\mathbf{v_1^T} + \lambda_2 u_2[m]\mathbf{v_2^T} + \cdots\cdots + \lambda_h u_h[m]\mathbf{v_h^T} + \begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \cdots & \tilde{f}_N \end{bmatrix}$$

除了平均值 $\begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \cdots & \tilde{f}_N \end{bmatrix}$ 之外

$\mathbf{v_1}^{\mathbf{T}}$ 是資料的最主要成分，$\mathbf{v_2}^{\mathbf{T}}$ 是資料的次主要成分，
$\mathbf{v_3}^{\mathbf{T}}$ 是資料的第三主要成分，以此類推

Example of PCA

3. 在處理二維數據時，有種方法是將數據垂直投影到某一直線，並以該直線為數線，進而了解投影點所成一維數據的變異。下圖的一組二維數據，試問投影到哪一選項的直線，所得之一維投影數據的變異數會是最小？

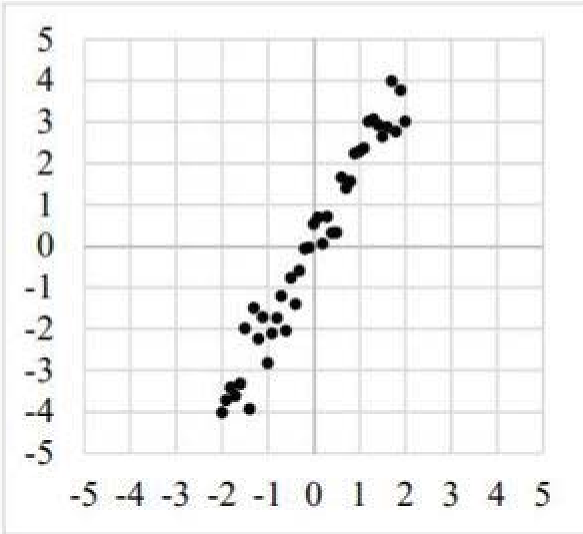
(1) $y = 2x$

(2) $y = -2x$

(3) $y = -x$

(4) $y = \dfrac{x}{2}$

(5) $y = -\dfrac{x}{2}$



From 2022 大考中心官網

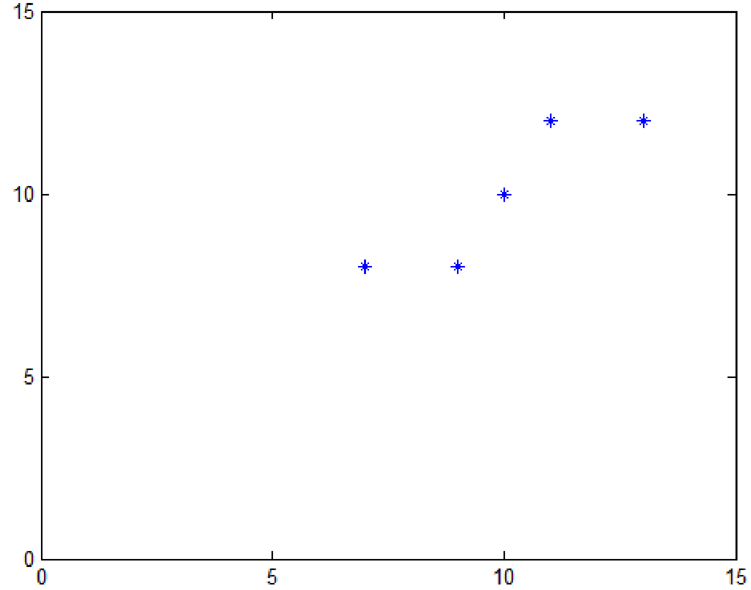
Example of PCA

假設在一個二維的空間中，有5個點，座標分別是

(7,8), (9,8), (10, 10), (11,12), (13,12)

$M = 5, N = 2$

試求這五個點的 PCA (即回歸線)



(Step 1) 將這五個座標點減去平均值 (10, 10)

(-3, -2), (-1 -2), (0, 0), (1, 2), (3, 2)

(Step 2) 形成 5x2 的 matrix

$$\mathbf{A} = \begin{bmatrix} -3 & -2 \\ -1 & -2 \\ 0 & 0 \\ 1 & 2 \\ 3 & 2 \end{bmatrix}$$

(Step 3) 計算 SVD

$$\mathbf{A} = \mathbf{USV}^{\mathbf{H}}$$

$$\mathbf{U} = \begin{bmatrix} -0.6116 & 0.3549 & 0 & 0.0393 & 0.7060 \\ -0.3549 & -0.6116 & 0 & 0.7060 & -0.0393 \\ 0 & 0 & 1 & 0 & 0 \\ 0.3549 & 0.6116 & 0 & 0.7060 & -0.0393 \\ 0.6116 & -0.3549 & 0 & 0.0393 & 0.7060 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 5.8416 & 0 \\ 0 & 1.3695 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} 0.7497 & -0.6618 \\ 0.6618 & 0.7497 \end{bmatrix}$$

主成分　　次要成分

$$\mathbf{A} = 5.8416 \begin{bmatrix} -0.6116 \\ -0.3549 \\ 0 \\ 0.3549 \\ 0.6116 \end{bmatrix} \begin{bmatrix} 0.7497 & 0.6618 \end{bmatrix} + 1.3695 \begin{bmatrix} 0.3549 \\ -0.6116 \\ 0 \\ 0.6116 \\ -0.3549 \end{bmatrix} \begin{bmatrix} -0.6618 & 0.7497 \end{bmatrix}$$

(Step 4)

$$\mathbf{A} \cong 5.8416 \begin{bmatrix} -0.6116 \\ -0.3549 \\ 0 \\ 0.3549 \\ 0.6116 \end{bmatrix} \begin{bmatrix} 0.7497 & 0.6618 \end{bmatrix}$$

$$\begin{bmatrix} 7 & 8 \\ 9 & 8 \\ 10 & 10 \\ 11 & 12 \\ 13 & 12 \end{bmatrix} \cong \begin{bmatrix} 10 & 10 \end{bmatrix} + 5.8416 \begin{bmatrix} -0.6116 \\ -0.3549 \\ 0 \\ 0.3549 \\ 0.6116 \end{bmatrix} \begin{bmatrix} 0.7497 & 0.6618 \end{bmatrix}$$

得到主成分 $\begin{bmatrix} 0.7497 & 0.6618 \end{bmatrix}$

這五個座標點可以近似成

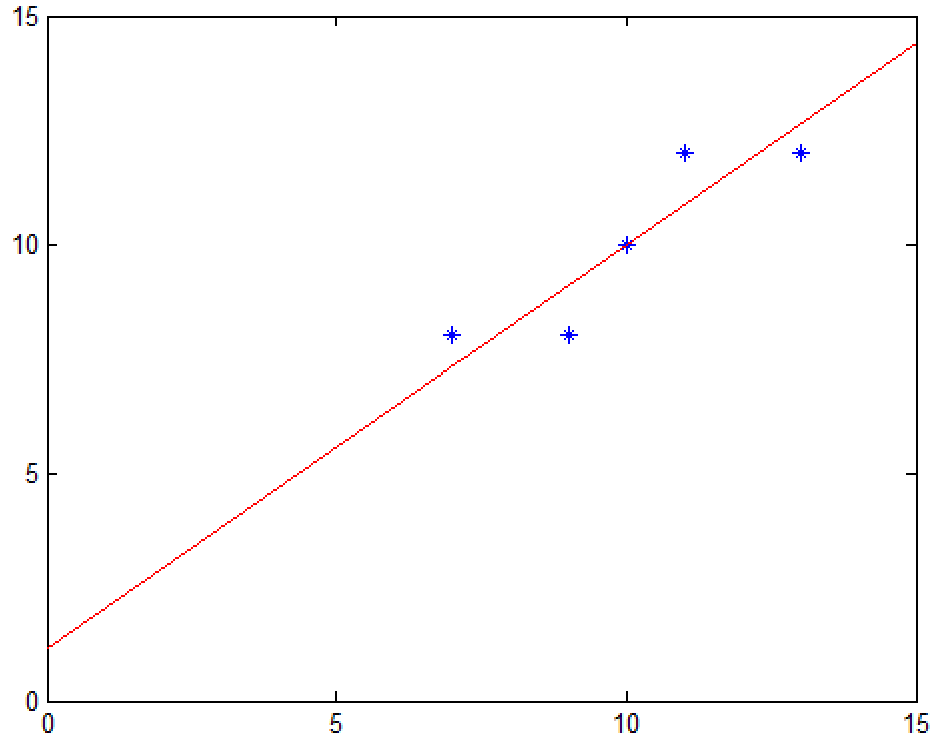
$$5.8416 \cdot u_m \begin{bmatrix} 0.7497 & 0.6618 \end{bmatrix} + \begin{bmatrix} 10 & 10 \end{bmatrix} \qquad m = 1, 2, \ldots, 5$$

$$u_1 = \text{-}0.6116, \quad u_2 = \text{-}0.3549, \quad u_3 = 0, \quad u_4 = 0.3549, \quad u_5 = 0.6116$$

回歸線

$$[10 \quad 10] + c[0.7497 \quad 0.6618]$$

$$c \in (-\infty, \infty)$$



Using the PCA method can obtain the best approximation result.

(Proof):

Without the loss of generalization, we discuss the problem in the 2D case (i.e., $N = 2$). Suppose that the location of the M points are

$$(x_1, y_1),\ (x_2, y_2),\ \ldots\ldots,\ (x_M, y_M)$$

We want to find a line passing through the origin such that the projection of (x_1, y_1), (x_2, y_2), ……, (x_M, y_M) on the line has the maximal sum of the square norm. That is, to find a unit vector

$$\mathbf{e} = (e_1, e_2) \quad \textit{where} \quad \|\mathbf{e}\| = 1$$

(The line passing through the origin is $\alpha\mathbf{e}$.) (1)

such that

$$\left\|\langle(x_1, y_1), \mathbf{e}\rangle\mathbf{e}\right\|^2 + \left\|\langle(x_2, y_2), \mathbf{e}\rangle\mathbf{e}\right\|^2 + \cdots\cdots + \left\|\langle(x_M, y_M), \mathbf{e}\rangle\mathbf{e}\right\|^2 \qquad (2)$$

is maximal. Note that

$$\begin{aligned}&\left\|\langle(x_1, y_1), \mathbf{e}\rangle\mathbf{e}\right\|^2 + \left\|\langle(x_2, y_2), \mathbf{e}\rangle\mathbf{e}\right\|^2 + \cdots\cdots + \left\|\langle(x_M, y_M), \mathbf{e}\rangle\mathbf{e}\right\|^2 \\ &= \left(\langle(x_1, y_1), \mathbf{e}\rangle\right)^2 + \left(\langle(x_2, y_2), \mathbf{e}\rangle\right)^2 + \cdots\cdots + \left(\langle(x_M, y_M), \mathbf{e}\rangle\right)^2\end{aligned} \qquad (3)$$

Suppose that for the matrix

$$\mathbf{A}=\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_M & y_M \end{bmatrix}$$

we have performed the SVD for **A** and decompose it into

$$\mathbf{A}=\mathbf{USV}^T$$

$$\mathbf{U}=\begin{bmatrix}\mathbf{u_1} & \mathbf{u_2} & \cdots & \mathbf{u_M}\end{bmatrix} \qquad \mathbf{V}=\begin{bmatrix}\mathbf{v_1} & \mathbf{v_2}\end{bmatrix} \qquad \mathbf{S}=\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{A}=\lambda_1\mathbf{u_1}\mathbf{v_1}^T+\lambda_2\mathbf{u_2}\mathbf{v_2}^T \tag{4}$$

If $\mathbf{v_1}=\begin{bmatrix} v_{1,1} \\ v_{1,2}\end{bmatrix}$, $\mathbf{v_2}=\begin{bmatrix} v_{2,1} \\ v_{2,2}\end{bmatrix}$ then $\mathbf{v_1}$ and $\mathbf{v_2}$ are orthonormal $\mathbf{v_1}^T\mathbf{v_2}=\mathbf{v_2}^T\mathbf{v_1}=0$

$$\mathbf{v_1}^T\mathbf{v_1}=\mathbf{v_2}^T\mathbf{v_2}=1$$

Therefore,

$$\mathbf{Av_1}=\lambda_1\mathbf{u_1}\mathbf{v_1^H}\mathbf{v_1}+\lambda_2\mathbf{u_2}\mathbf{v_2^H}\mathbf{v_1}=\lambda_1\mathbf{u_1} \qquad \mathbf{Av_2}=\lambda_1\mathbf{u_2} \tag{5}$$

Since $\mathbf{v_1}$ and $\mathbf{v_2}$ are orthonormal, any two-entry vector $\mathbf{e}$ can be expressed as

$$\mathbf{e} = c_1\mathbf{v_1^T} + c_2\mathbf{v_2^T} \qquad \text{where} \qquad c_1^2 + c_2^2 = 1$$

Therefore, from (3),

$$\left\|\left\langle (x_1, y_1), \mathbf{e}\right\rangle \mathbf{e}\right\|^2 + \left\|\left\langle (x_2, y_2), \mathbf{e}\right\rangle \mathbf{e}\right\|^2 + \cdots\cdots + \left\|\left\langle (x_M, y_M), \mathbf{e}\right\rangle \mathbf{e}\right\|^2$$

$$= \left(\left\langle (x_1, y_1), c_1\mathbf{v_1^T} + c_2\mathbf{v_2^T}\right\rangle\right)^2 + \left(\left\langle (x_2, y_2), c_1\mathbf{v_1^T} + c_2\mathbf{v_2^T}\right\rangle\right)^2 + \cdots + \left(\left\langle (x_M, y_M), c_1\mathbf{v_1^T} + c_2\mathbf{v_2^T}\right\rangle\right)^2 \tag{6}$$

Moreover, from (5),

$$\left(\left\langle (x_m, y_m), c_1\mathbf{v_1^T} + c_2\mathbf{v_2^T}\right\rangle\right)^2 = \left(\lambda_1 c_1 u_{1,m} + \lambda_2 c_2 u_{2,m}\right)^2 \tag{7}$$

where $u_{1,\mathrm{m}}$ and $u_{2,\mathrm{m}}$ are the m^{th} entries of $\mathbf{u_1}$ and $\mathbf{u_2}$, respectively. Therefore,

$$\left\|\left\langle (x_1, y_1), \mathbf{e}\right\rangle \mathbf{e}\right\|^2 + \left\|\left\langle (x_2, y_2), \mathbf{e}\right\rangle \mathbf{e}\right\|^2 + \cdots\cdots + \left\|\left\langle (x_M, y_M), \mathbf{e}\right\rangle \mathbf{e}\right\|^2$$

$$= \sum_{m=1}^{M}\left(c_1\lambda_1 u_{1,m} + c_2\lambda_2 u_{2,m}\right)^2 = c_1^2\lambda_1^2\sum_{m=1}^{M}u_{1,m}^2 + c_2^2\lambda_2^2\sum_{m=1}^{M}u_{2,m}^2 + 2c_1\lambda_1 c_2\lambda_2\sum_{m=1}^{M}u_{1,m}u_{2,m}$$

Since $\mathbf{u_1}$ and $\mathbf{u_2}$ are orthonormal,

$$\sum_{m=1}^{M} u_{1,m}^2 = \sum_{m=1}^{M} u_{2,m}^2 = 1, \quad \sum_{m=1}^{M} u_{1,m} u_{2,m} = 0$$

we have

$$\left\| \left\langle (x_1, y_1), \mathbf{e} \right\rangle \mathbf{e} \right\|^2 + \left\| \left\langle (x_2, y_2), \mathbf{e} \right\rangle \mathbf{e} \right\|^2 + \cdots\cdots + \left\| \left\langle (x_M, y_M), \mathbf{e} \right\rangle \mathbf{e} \right\|^2 = c_1^2 \lambda_1^2 + c_2^2 \lambda_2^2$$

Since $c_1^2 + c_2^2 = 1$ and $\lambda_1 > \lambda_2$, the best way to assign c_1 and c_2 is

$$c_1 = 1, \ c_2 = 0$$

That is, we can choose

$$\mathbf{e} = \mathbf{v_1}^T$$

and the projection of (x_m, y_m) on $\mathbf{e}$ is $\lambda_1 u_{1,m} \mathbf{v_1^T}$

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_M & y_M \end{bmatrix} \cong \begin{bmatrix} \lambda_1 u_{1,1} \mathbf{v_1^T} \\ \lambda_1 u_{1,2} \mathbf{v_1^T} \\ \vdots \\ \lambda_1 u_{1,M} \mathbf{v_1^T} \end{bmatrix} = \lambda_1 \mathbf{u_1} \mathbf{v_1^T}$$