V. Homomorphic Signal Processing

同態

◉ 5-A Homomorphism

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

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

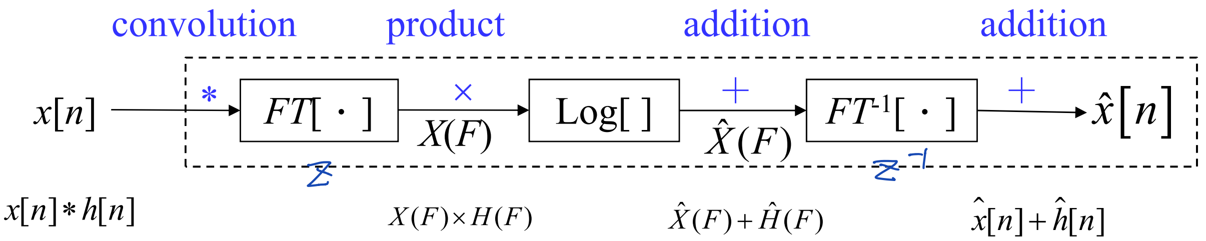
把複雜的運算，變成效能相同但較簡單的運算，取 log 是讓 × → + 比較 easy

◉ 5-B Cepstrum

倒頻譜

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

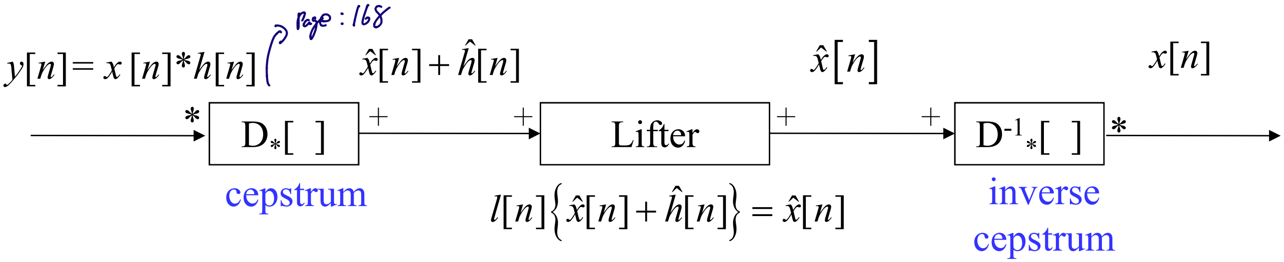
For the process of cepstrum (denoted by $D_*[\ \cdot\ ]$)



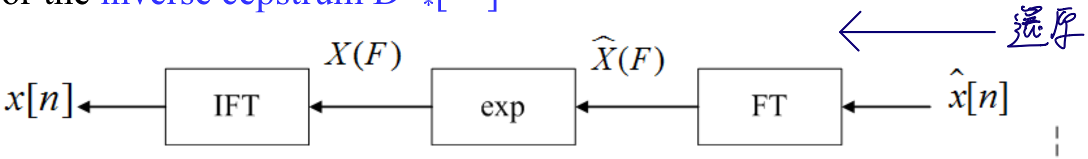
$\hat{x}(F) = \log(X(F))$, $\hat{H}(F) = \log(H(F))$, $\hat{x}[n] = FT^{-1}(\hat{X}(F))$, $\hat{h}[n] = FT^{-1}(\hat{H}(F))$

FT: discrete-time Fourier transform

- 由 $y[n]=x[n]*h[n]$ 重建 $x[n]$ ，$x[n]$ 和 $h[n]$ 位置要不同

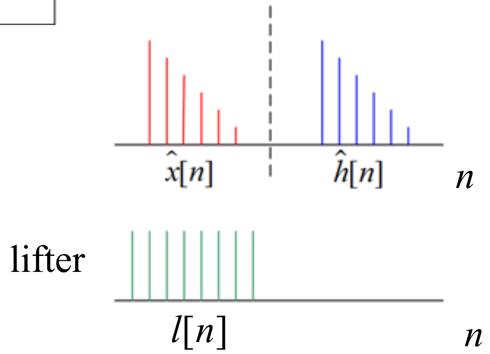


For the inverse cepstrum $D^{-1}_*[\cdot]$



- 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



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

◉ 5-C Methods for Computing the Cepstrum

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

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

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

ambiguity for phase

Problems: (1) $\log|X(F)| \to \infty$ when $X(F) = 0$, 看 paper 需考慮 log 是否是 stable

(2) $\arg[X(F)]$ has infinite solutions.
$(+2\pi N)$

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)

result不變, Z^r可忽略

time delay

$Z^{-r} X(Z)$

$x[n-r]$

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

$= 0$ $= 0$

zeros: a_k, b_k^{-1}, poles: c_k, d_k^{-1}

where

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

$|a_k| \le 1$, $|b_k^{-1}| \ge 1$

Poles & zeros inside unit circle	Poles & zeros outside unit circle

$\prod$: continued multiplication $\prod_{k=1}^{2} (1 - a_k z^{-1}) = (1 - a_1 z^{-1})(1 - a_2 z^{-1})$

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

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

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

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

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

?

Taylor series

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

$z = (re^{jw})$

$$Z^{-1}(\log(A)) = \begin{cases} \log(A) & \text{for } n=0 \\ 0 & \text{otherwise} \end{cases}$$

$t_0 = 0$, $\quad \frac{d^n}{dt^n} \log(1-t) = \frac{(-1)^{n-1}(n-1)!}{(t-1)^n}$

$$\log(1-t) = 0 + \sum_{n=1}^{\infty} \frac{(-1)^{n-1} \cdot (n-1)!}{n!(t-1)^n}\Bigg|_{t=0} t^n = \sum_{n=1}^{\infty} \frac{-1 \cdot t^n}{n}$$

$$\Rightarrow \log(1 - a_k z^{-1}) = -\sum_{n=1}^{\infty} \frac{a_k^n}{n} z^{-n}, \quad \log(1 - b_k z) = -\sum_{n=1}^{\infty} \frac{b_k^n}{n} z^n = \sum_{n=-\infty}^{-1} \frac{b_k^{-n}}{n} z^{-n}$$

$(n_{new} = -n_{old})$

$$\because \; x(z) = \sum_{n=1}^{\infty} x[n] \cdot z^{-n}$$

$$\Rightarrow Z^{-1}(\log(1 - a_k z^{-1})) = \begin{cases} -\frac{a_k^n}{n} & n \geq 1 \\ 0 & n \leq 0 \end{cases}$$

$$Z^{-1}(\log(1 - b_k z)) = \begin{cases} \frac{b_k^{-n}}{n} & n \leq -1 \\ 0 & n \geq 0 \end{cases}$$

常用, HW

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

ex1:

$x[n]$ 0.5

0 1

$x(z) = 1 + 0.5z^{-1}$, $|n|$

$= z^{-1}(z+0.5)$

$a_1 = -0.5$

$\hat{x}[n] = \begin{cases} -\frac{(-0.5)^n}{n}, & n>0 \\ 0, & n \le 0 \end{cases}$

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

ex2:

$x[n]$ 2 3 1

0 1 2

$x(z) = 2 + 3z^{-1} + z^{-2}$

$= 2(1+z^{-1})(1+0.5z^{-1})$, $a_1 = -1$, $a_2 = -0.5$

$\hat{x}[n]: \begin{cases} \log 2 \\ -\frac{(-1)^n}{n} - \frac{(-0.5)^n}{n}, & n>0 \\ 0, & \text{otherwise} \end{cases}$

Poles & zeros inside unit circle, right-sided sequence

Poles & zeros outside unit circle, left-sided sequence

$(0.6^n z^{-n} - \cdots)$

$\delta[n]$ FIR, cesptrum 無限

HW

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

$x[n] = \begin{cases} 0.6^n, & \text{for } n \ge 0 \\ 0, & n < 0 \end{cases}$ # 0 ф 5

$1 + \left(\frac{0.6z^{-1}}{1-0.6z^{-1}}\right)$

(3) For FIR case, $c_k = 0$, $d_k = 0$

(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**

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

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

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

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

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

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

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

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

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

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 $\dfrac{1}{n}$

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

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

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

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

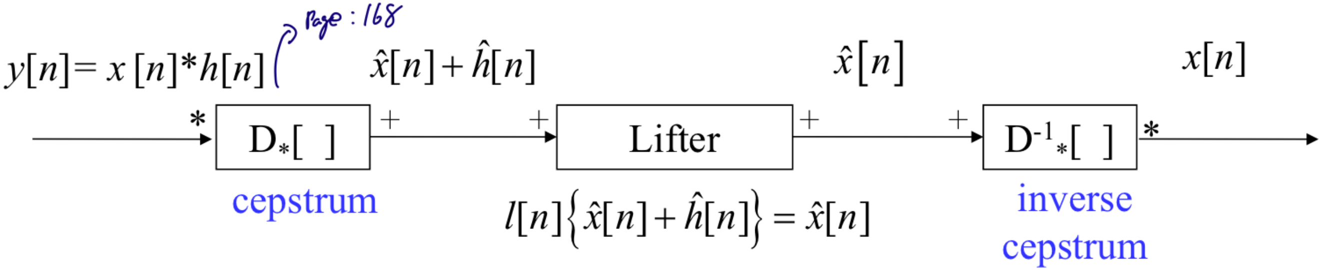
because of no a_k, c_k

178

P.4) If $x[n]$ is of finite duration, then $\hat{x}[n]$ has infinite duration

169

- 由 $y[n]=x[n]*h[n]$ 重建 $x[n]$ ， $x[n]$ 和 $h[n]$ 位置要不同



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

$y[n] = x[n] + a_1 x[n-15] + a_2 x[n-20]$

$P(z) = 1 + a_1 z^{-15} + a_2 z^{-20}$

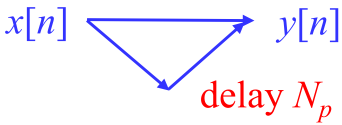
$\hat{P}(z) = \log(1 + a_1 z^{-15} + a_2 z^{-20})$

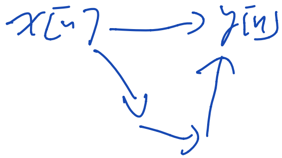
HW3

◉ 5-E Application of Homomorphic Deconvolution

(1) Equalization for Echo

$$y[n] = \overset{1}{x[n]} + \alpha\, x[n - \overset{2}{N_p}]$$





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

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

$y[n] = x[n] + a_1 x[n-15] + a_2 x[n-20]$

$y[n] = x[n] + \alpha x[n-N_p] - \frac{\alpha^2}{2} x[n-2N_p]$

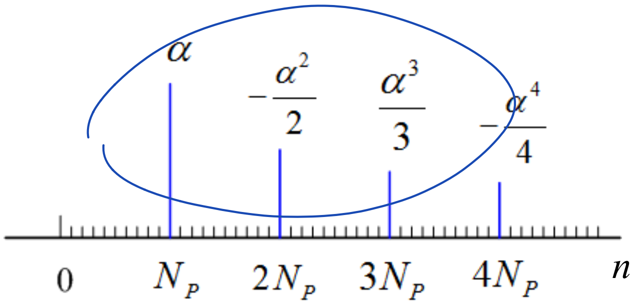
$$P(Z) = 1 + \alpha Z^{-N_p}$$

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

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

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

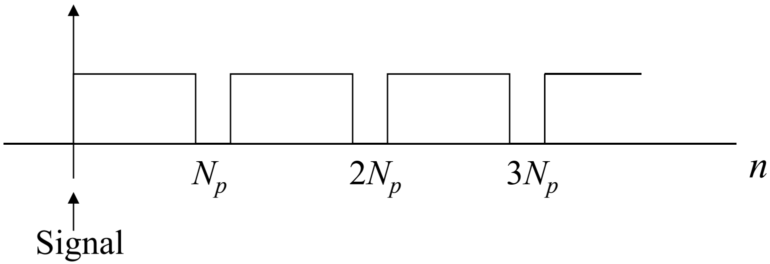
$$\hat{p}[n] = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\alpha^k}{k} \delta(n - k \cdot N_p)$$



$\hat{p}[1] = \frac{\alpha}{1} \delta(n - N_p)$

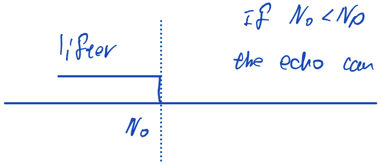
$\hat{p}[2] = -\frac{\alpha^2}{2} \delta[n - 2 \cdot N_p]$

Filtering out the echo by the following "lifter": Independent of α



Q: For the case where N_p is unknown

N_p 夠大，就對原 signal 影響不大



If $N_0 < N_p$ the echo can be deleted

(2) Representation of acoustic engineering

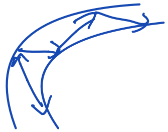
$$y[n] \quad = \quad x[n] \quad * \quad h[n]$$

<u>Synthesized music</u>   <u>music</u>   <u>building effect：e.g. 羅馬大教堂的 impulse response</u>

(3) Speech analysis

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

Speech wave	Global wave shape	聲帶 Vocal tract impulse	Pitch



Vocal path (multipath)

They can be separated by filtering in the complex cepstrum domain

(4) Seismic Signals

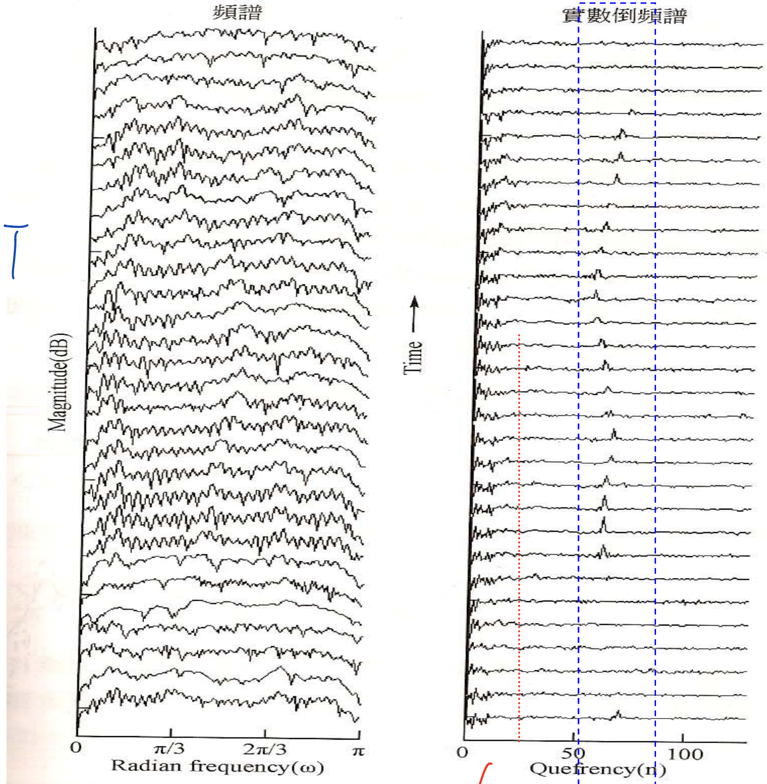
地震波

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

用 cepstrum 將 multipath的影響去除

From 王小川，"語音訊號處理"，全華出版，台北，民國94年。

FT



cepstrum

圖 7-5 元音頻譜與倒頻譜

multipath

前13個值去分析，就夠用了 by mel cepstrum

From 王小川，"語音訊號處理"，全華出版，台北，民國94年。

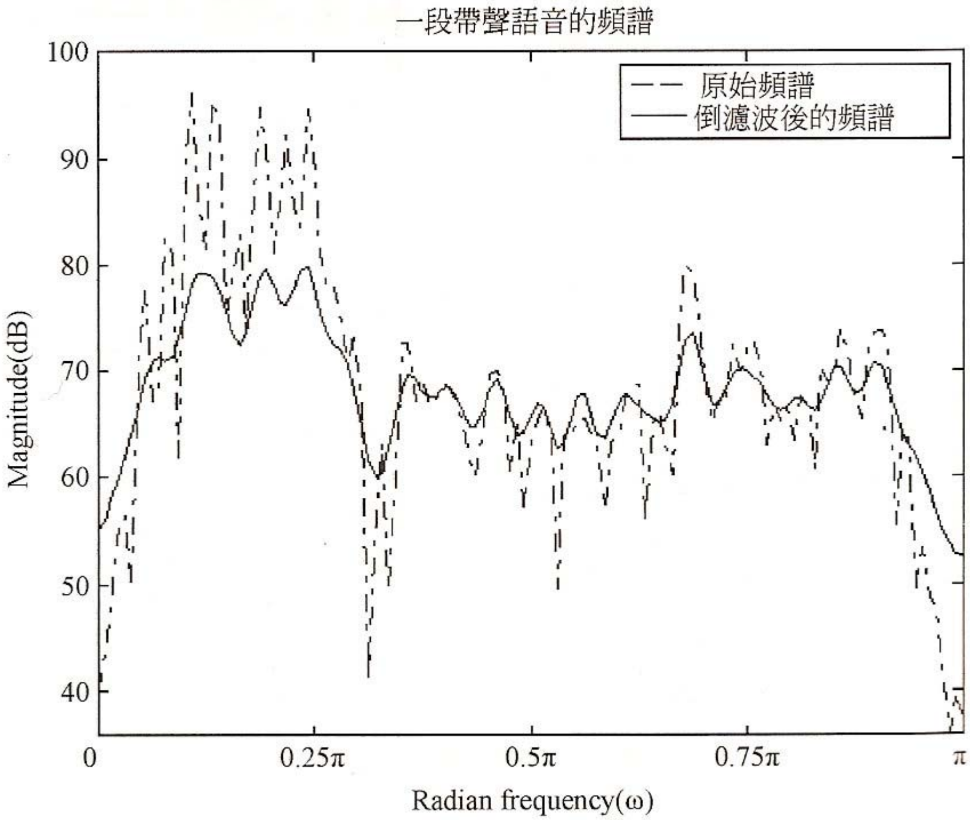


圖 7-6　經過倒濾波器作平滑處理的頻譜

◉ 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

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

inverse Z transform

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

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

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

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

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

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

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

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

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

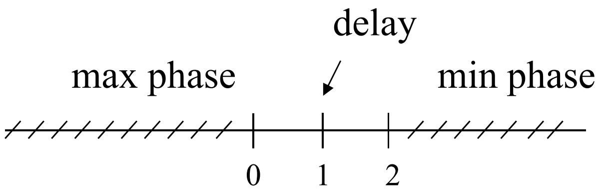
$$\hat{C}(n) = \frac{-\hat{x}_d(n+1)}{n} \qquad n \neq 0 \qquad \textit{diff cepstrum}$$

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

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, $\because \quad \hat{x}_d(n+1) = n \cdot \hat{c}(n) \propto n \cdot \dfrac{1}{n} = 1$

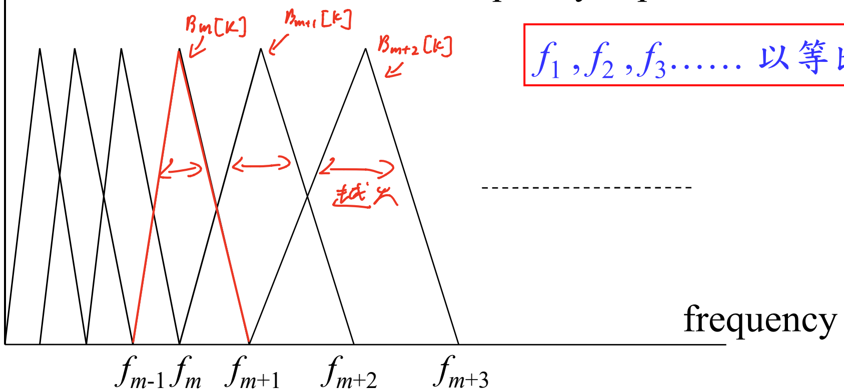
◉ 5-H Mel-Frequency Cepstrum (梅爾頻率倒頻譜)

orientation of vocal signal analysis

Take log in the frequency mask

gain

mask of Mel-frequency cepstrum



$\boxed{f_1, f_2, f_3 \ldots\ldots \text{ 以等比級數增加}}$

✓ 100Hz, 200Hz

1000Hz, 1100Hz

聲音是要以"頻率比"為決定因素

$B_m[k] = 0$ 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-Cepstrum

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

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

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

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

$x[n]$ is real

$X[k] = X^*[-k]$

$|X[k]| = |X[-k]|$

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

Advantages :

(1) $B_m[k]$ 為等比 series，符合人類聽覺特性

(2) The probability that $\sum |X[k]|^2 B_m[k] = 0$ is much less

(3) No phase ambiguity since $\sum |X[k]|^2 B_m[k]$ is real

(4) For the inverse, the DCT (discrete cosine transform) is applied to reduce the computation loading

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

+ Machine Learning

◉ 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 ringin.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

- 思考: 所以，取樣間隔多大?

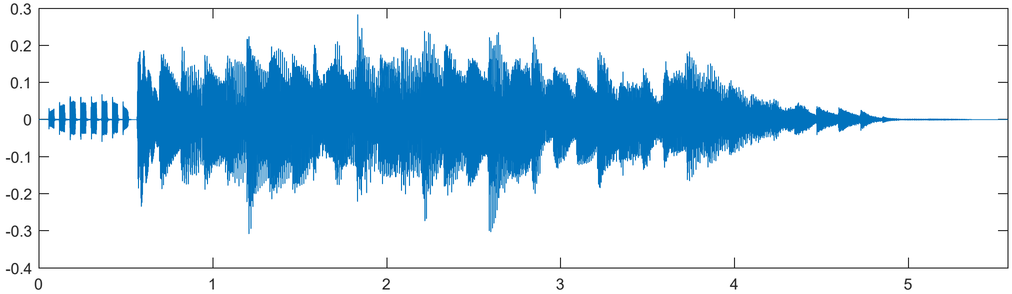
- 這個聲音檔有多少秒？

雙聲道 (**Stereo**，俗稱立體聲)

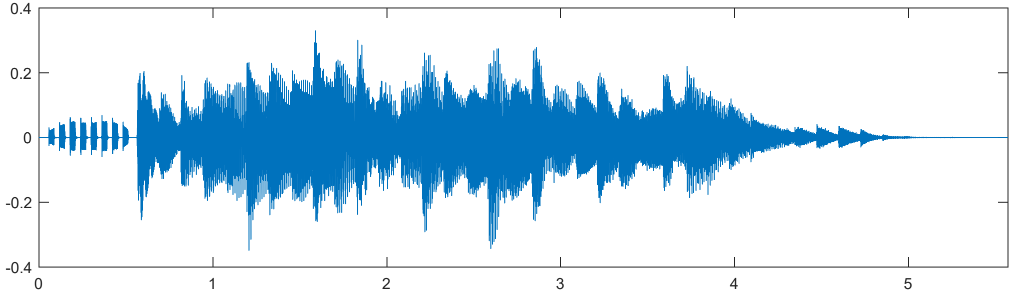
畫出聲音的波型

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)])



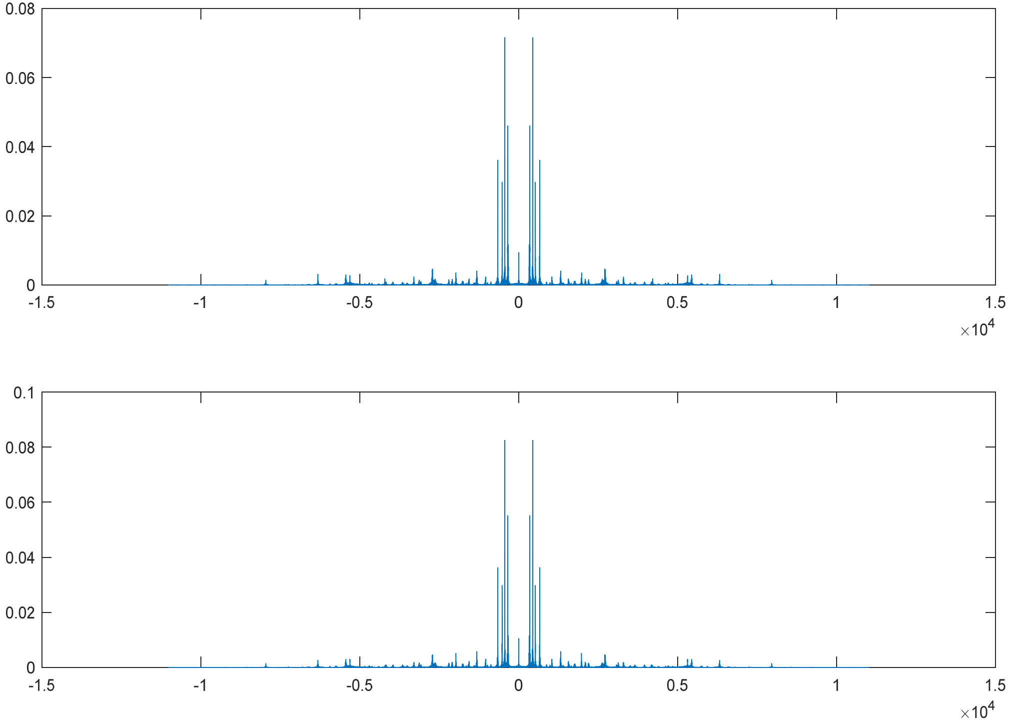
注意： *.wav 檔中所讀取的資料，值都在 −1 和 +1 之間

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

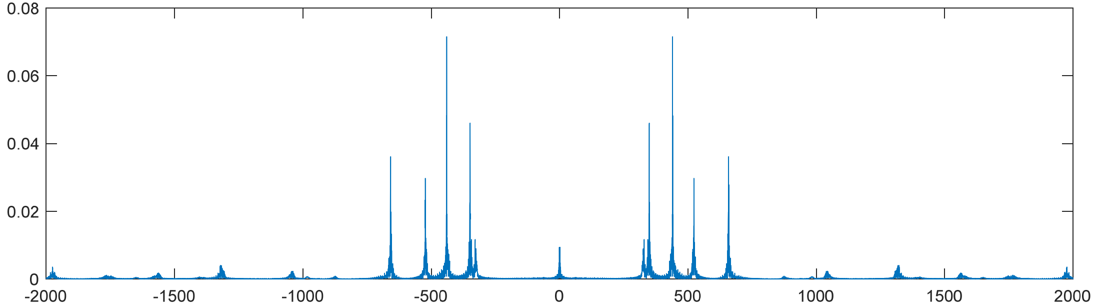
```
X=X.';
N=length(X); N1=round(N/2);
dt=1/fs;
X1=[X(N1+1:N),X(1:N1)]*dt;  % shifting for spectrum
f=[[N1:N-1]-N,0:N1-1]/N*fs;    % valid f
plot(f, abs(X1));
```

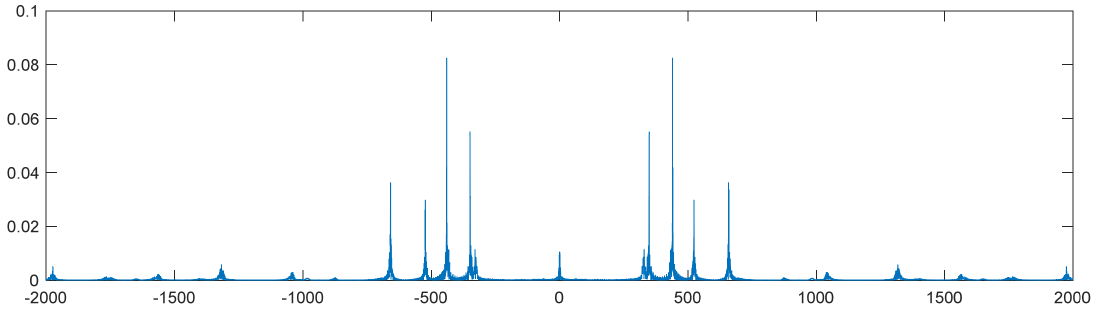
Alarm01.wav 的頻譜



Alarm01.wav 的頻譜

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'));
```

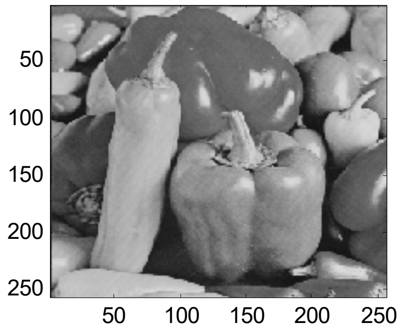
(注意，如果 Pepper.bmp 是個灰階圖，im 將是一個矩陣)

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);
```