Time Frequency Analysis and Wavelet Transforms

時頻分析與小波轉換

授課者： 丁 建 均

Office：明達館723室， TEL： 33669652

E-mail: jjding@ntu.edu.tw

課程網頁：http://djj.ee.ntu.edu.tw/TFW.htm

歡迎大家來修課，也歡迎有問題時隨時聯絡！

上課方式

(1) 錄影 ，影片將藉由 NTU Cool 下載　http://cool.ntu.edu.tw

(2) 現場 (明達館231室)

作業和報告繳交方式

用 NTU Cool 來繳交作業與報告的電子檔　http://cool.ntu.edu.tw

注意，Tutorial 一定要交Word 或 Latex 原始碼
Wiki 要寄編輯條目的連結給老師

- 評分方式：

平時分數: 15 scores

基本分 12 分，各位同學皆可拿到

另外，每週都將會在課堂上問一個問題，請特定學號尾數的同學，在作業上回答 (每位同學每學期會輪到四次左右)，每正確回答一次加 0.8 分

將來再視疫情決定是否要在上課時問答

Homework: 60 scores

5 times, 每 3 週一次

請自己寫，和同學內容相同 ，將扣 60% 的分數，就算寫錯但好好寫也會給 40~95% 的分數，遲交分數打 8 折，不交不給分。

不知道如何寫，可用 E-mail 和我聯絡，或於下課時和老師討論

Term paper 25 scores

Term paper 25 scores

方式有五種，可任選其中一種

(1) 書面報告

(10頁以上(不含封面)，中英文皆可，11或12的字體，題目可選擇和課程有關的任何一個主題。

格式不限，但儘量和一般寫期刊論文或碩博士論文相同，包括 abstract, conclusion, 及 references，並且要分 sections，必要時有subsections。鼓勵多做實驗及模擬。

嚴禁剪刀漿糊 (Ctrl-C , Ctrl-V) 的情形，否則扣 60% 的分數

pw： gice1997

(2) Tutorial

限十二個名額，和書面報告格式相同，但 17頁以上(若為加強前人的 tutorial，則頁數為 (2/3)N + 12 以上，N 為前人 tutorial 之頁數)，題目由老師指定 (see page 12)，以清楚且有系統的介紹一個主題的基本概念和應用為要求，為上課內容的進一步探討和補充，**交Word 檔**。

選擇這個項目的同學，學期成績加 4分

(3) 口頭報告 full

限五個名額，每個人 30分鐘，題目可選擇和課程有關的任何一個主題。口頭報告的同學，請在 12月21日之前將報告的影片寄給老師。有意願的同學，請儘早告知，以先登記的同學為優先。

口頭報告鼓勵同學們參與聽講和票選，有參與票選的同學期末報告成績加2分。

選擇這個項目的同學，學期成績加2分

(4) 編輯 Wikipedia

中文或英文網頁皆可，至少 2 個條目，但不可同一個條目翻成中文和英文。總計80行以上。限和課程相關者，自由發揮，越有條理、有系統的越好

選擇編輯 Wikipedia 的同學，請於12月14日(本學期最後一次上課)前，向我登記並告知我要編緝的條目(2 個以上)，若有和其他同學選擇相同條目的情形，則較晚向我登記的同學將更換要編緝的條目

編輯完成之後，要將連結寄給老師

(1)~(4) 若有做實驗模擬，請附上程式碼，會有額外的加分 (鼓勵不強制)

(5) 程式編寫，協助信號處理程式資料庫的建立

py Mat full

選擇第13頁的其中一組題目，來編寫相關的程式，程式用 Matlab 或 Python或 C 編寫皆可，共5個題目15個名額

唯，選擇程式編寫的同學，請於名額額滿之前，向我登記並告知我要編寫的程式 (2 個題目以上)，以避免和其他同學重覆。

選擇這個題目的同學，期末要用 NTUCool 交程式，並另外寫一個 ReadMe 的檔案(Word 檔)，說明程式該如何執行，並舉例子顯示程式執行結果。

書面報告、Tutorial、編輯 Wikipedia、和程式編寫的期限是 12月28日

(6) full

9:10 ~10:00
10:10~11:00
11:10~12:00

上課時間：15 週

9/7,

9/14,

9/21, 出 HW1

9/28,

10/5, 交 HW1

10/12, 出 HW2

10/19,

10/26, 交 HW2

3n 出 HW

3n+2 交 HW

11/2, 出 HW3

11/9,

11/16, 交 HW3

11/23, 出 HW4

11/30,

12/7, 交 HW4,

12/14, 出 HW5

12/21, 口頭報告的同學交影片

12/28, 交 HW5 及 term paper

課程大綱：

(1) Introduction

(2) Short-Time Fourier Transform

(3) Gabor Transform

(4) Implementation of Time-Frequency Analysis

(5) Wigner Distribution Function

(6) Cohen's Class Time-Frequency Distribution

(7) S Transforms, Gabor-Wigner Transforms, Matching Pursuit, and Other Time Frequency Analysis Methods

(8) Movement in the Time-Frequency Plane and Fractional Fourier Transforms

(9) Filter Design by Time-Frequency Analysis

(10) Modulation, Multiplexing, Sampling, and Other Applications

time frequency analysis

(續)

課程大綱：

(11) Hilbert Huang Transform 黃鍔院士

(12) From Haar Transforms to Wavelet Transforms

(13) Continuous Wavelet Transforms with Discrete Coefficients

(14) Discrete Wavelet Transform

(15) Applications of the Wavelet Transform

wavelet transform

- 上課資料：

(1) 講義 (將放在網頁上，請大家每次上課前先印好)

(2) S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, 3rd ed., 2009.

(3) S. Qian and D. Chen, *Joint Time-Frequency Analysis: Methods and Applications*, Prentice-Hall, 1996.

(4) P. Flandrin, *Time-frequency / Time Scale Analysis*, translated by J. Stöckler, Academic Press, San Diego, 1999.

(5) K. Grochenig, *Foundations of Time-Frequency Analysis*, Birkhauser, Boston, 2001.

(6) L. Debnath, *Wavelet Transforms and Time-Frequency Signal Analysis*, Birkhäuser, Boston, 2001.

(7) F. Hlawatsch and F. Auger, *Time-frequency Analysis Concepts and Methods*, Wiley, London, 2008.

Matlab Program

Download: 請洽台大各系所

參考書目

洪維恩，Matlab 程式設計，旗標，台北市，2013 .　(合適的入門書)

張智星，Matlab 程式設計入門篇，第四版，碁峰，2016.

預計看書學習所花時間： 3~5 天

Python Program

Download: https://www.python.org/

參考書目

葉難， Python程式設計入門，博碩，2015

黃健庭， Python程式設計：從入門到進階應用，全華，2020

The Python Tutorial　　https://docs.python.org/3/tutorial/index.html

Tutorial 可供選擇的題目(可以略做修改)

(1) Higher-Order Synchrosqueezing Transform

✓(2) Chirp Radar for Distance Measurement

✓(3) Denoising Using Time-Frequency Analysis

✓(4) Chromagram for Music Signal Analysis

✓(5) Speaker Identification Using Time-Frequency Analysis

(6) Gabor Network

✓(7) Chirplet Transform

✓(8) Applications of Time-Frequency Analysis in Financial Market

(9) Legendre Wavelet and Hermite Wavelet

(10) Shanoon Wavelet

✓(11) Enhanced Compression Wavelet (ECG) Image Format

✓(12) Wavelet Transform for Video Analysis

程式編寫可供選擇的題目

(有意願的同學可選擇其中一組，用 Matlab, Python，或 C++皆可，要加說明檔 ReadMe，不可 copy 網路程式)

(1) (i) Polynomial Wigner Distribution ($q = 4$)

(ii) Modified Wigner Distribution Function (Modified Form II)

(2) (i) Matching Pursuit

(ii) S Transform for Any Window $s(f)$

(3) (i) Using the Hilbert Transform to determine the instantaneous Frequency

(ii) Hilbert-Huang Transform with pre-Denoising

(4) (i) Generating the Daubechies Wavelet for $2p$ Point

(ii) The Curvelet

(5) (i) The Bandlet

(ii) The Contourlet

I. Introduction

Fourier transform (FT)

$$X(f)=\int_{-\infty}^{\infty} x(t)e^{-j2\pi f t}dt$$ Time-Domain $\rightarrow$ Frequency Domain

$\uparrow$ t varies from $-\infty\sim\infty$

Laplace Transform $X(s)=\int_{-\infty}^{\infty} x(t)e^{-st}dt$

Cosine Transform, Sine Transform, Z Transform.

Some things make the FT not practical:

(1) It less happens that a signal has the interval of $(-\infty\sim\infty)$

Even for a signal with infinite length, we are only interested in the characteristics in a finite interval.

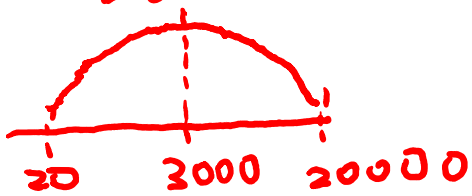
(2) It is hard to observe the variation of spectrum with time by the FT.

Do: 261.63 Hz $261.63 \cdot 2^{\frac{k}{12}}$

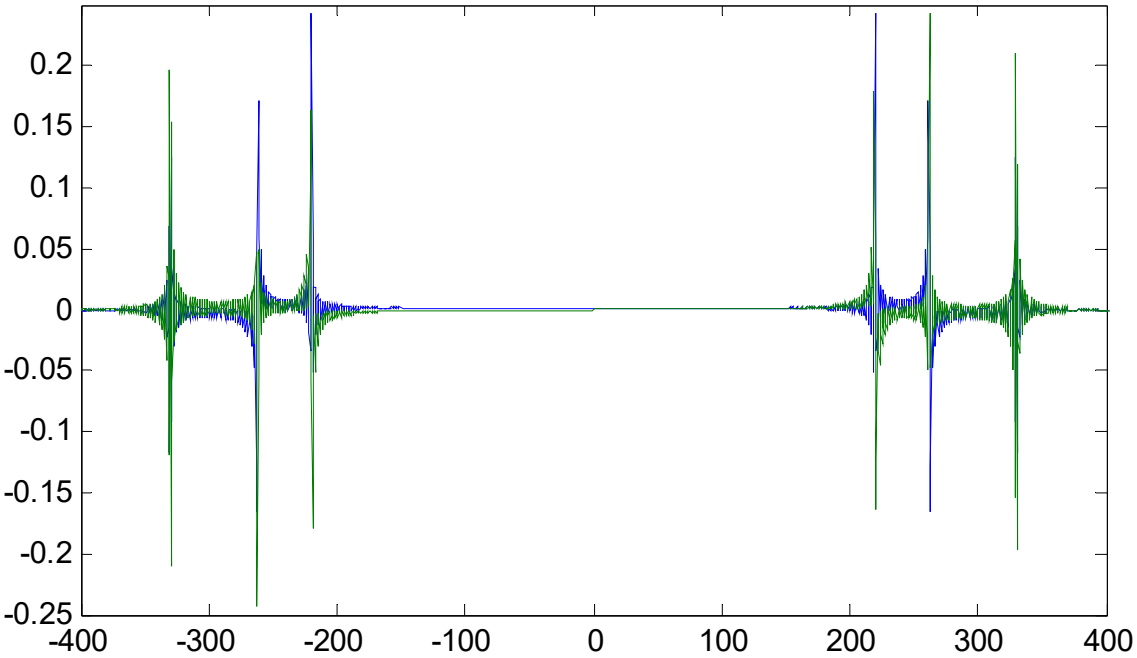
Example 1: $x(t) = \cos(440\pi t)$ when $t < 0.5$, ±220 Hz 低 La

$x(t) = \cos(660\pi t)$ when $0.5 \le t < 1$, ±330 Hz Mi

$x(t) = \cos(524\pi t)$ when $t \ge 1$ ±262 Hz Do



The Fourier transform of $x(t)$



Frequency

(A) Finite-Supporting Fourier Transform

$$X(f)=\int_{t_0-B}^{t_0+B} x(t)e^{-j2\pi f t}dt = \int_{-\infty}^{\infty} x(\tau)\Pi\left(\frac{\tau-t_0}{2B}\right)e^{-j2\pi f\tau}d\tau$$

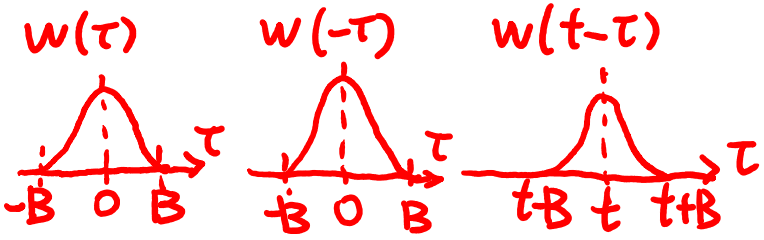
(B) Short-Time Fourier Transform (STFT)

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

$w(t)$: window function 或 mask function

STFT 也稱作 windowed Fourier transform 或

time-dependent Fourier transform



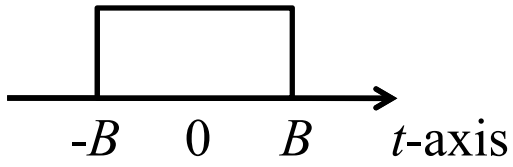
[Ref] L. Cohen, *Time-Frequency Analysis*, Prentice-Hall, New York, 1995.

[Ref] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, London: Prentice-Hall, 3rd ed., 2010.

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

最簡單的例子： $w(t) = 1$ for $|t| \le B$,

$w(t) = 0$ otherwise



rectangular function $\Pi(t) = \begin{cases} 1 & |t| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$

$\Pi\left(\frac{t-t_0}{2B}\right)$

$t = t_0 - B$, $\frac{t-t_0}{2B} = -\frac{1}{2}$

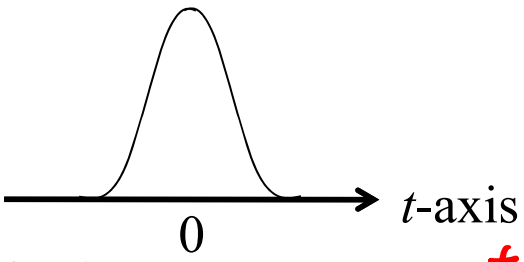
$t = t_0 + B$, $\frac{t-t_0}{2B} = \frac{1}{2}$

此時 Short-time Fourier transform 可以改寫

$$X(t,f)=\int_{t-B}^{t+B} x(\tau)e^{-j2\pi f\tau}d\tau$$

其他的例子：

$$w(t)=\exp\left(-\pi\sigma t^2\right)$$



$e^{-\pi t^2} < 10^{-5}$ for $|t| > 1.9143$

一般我們把 exp($-\pi\sigma t^2$) 稱作為 Gaussian function 或 Gabor function

此時的 Short-Time Fourier Transform 亦稱作 Gabor Transform

$e^{-\pi 200 t^2} < 10^{-5}$ for $|t| > \frac{1.9143}{\sqrt{200}} \cong 0.136$

(C) Gabor Transform

$$G_x(t,f)=\int_{-\infty}^{\infty} e^{-\pi\sigma(\tau-t)^2} e^{-j2\pi f\tau} x(\tau)\,d\tau$$

$$G_x(t,\omega)=\sqrt{\frac{1}{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{\sigma(\tau-t)^2}{2}} e^{-j\omega\tau} x(\tau)\,d\tau$$

• S. Qian and D. Chen, *Joint Time-Frequency Analysis: Methods and Applications*, Prentice Hall, N.J., 1996.

Without cross term, poor clarity

(D) Wigner Distribution Function

$$G_x(t,f)=\int_{-\infty}^{\infty} e^{-j2\pi f\tau} x(t+\tau/2)\,x^*(t-\tau/2)\,d\tau$$

$$G_x(t,\omega)=\int_{-\infty}^{\infty} e^{-j\omega\tau} x(t+\tau/2)\,x^*(t-\tau/2)\,d\tau$$

With cross term, high clarity

Example: $x(t) = \cos(440\pi t)$ when $t < 0.5$, 220Hz

$x(t) = \cos(660\pi t)$ when $0.5 \le t < 1$, 330Hz

$x(t) = \cos(524\pi t)$ when $t \ge 1$ 262Hz

B=330, F=660

T=1.5 TF=990 points

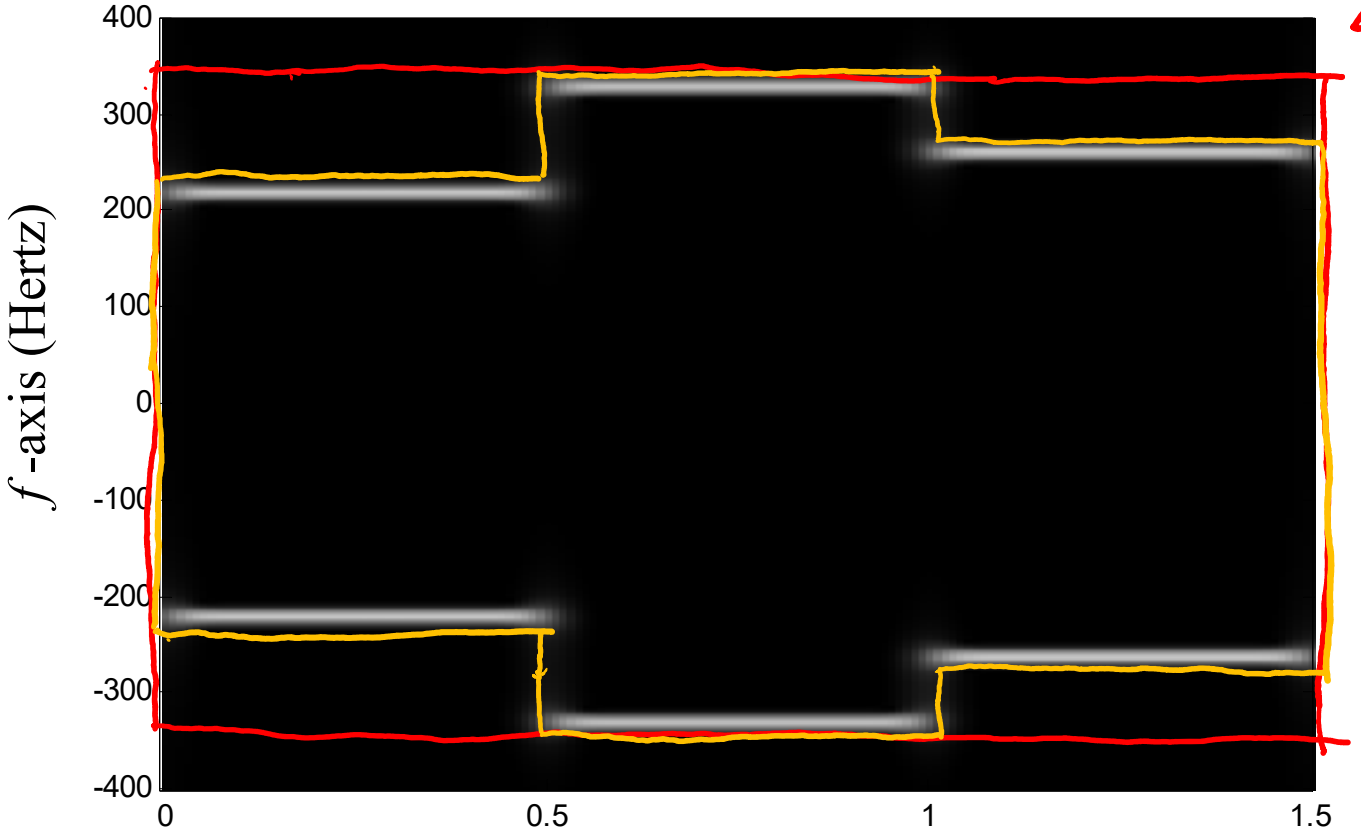
Adaptive sampling

$\Delta t < \frac{1}{440}$ $t<0.5$

$\Delta t < \frac{1}{660}$ $0.5<t<1$

$\Delta t < \frac{1}{524}$ $1<t<1.5$

The Gabor transform of $x(t)$ ($\sigma = 200$)



440×0.5+
660×0.5+
524×0.5
=812 points

|X(t,f)|

用 Gray level 來表示 $X(t, f)$ 的 amplitude

★ Instantaneous Frequency 瞬時頻率

If $x(t) = \sum_{k=1}^{N} a_k \cdot \exp(j \cdot \phi_k(t))$ around t_0

then the instantaneous frequency of $x(t)$ at t_0 are

$$\frac{\phi_1'(t_0)}{2\pi}, \frac{\phi_2'(t_0)}{2\pi}, \frac{\phi_3'(t_0)}{2\pi}, \cdots, \frac{\phi_N'(t_0)}{2\pi}$$ (以頻率 frequency 表示)

$$\phi_1'(t_0), \phi_2'(t_0), \phi_3'(t_0), \cdots, \phi_N'(t_0)$$ (以角頻率 angular frequency 表示)：

If the order of $\phi_k(t)$ > 1, then instantaneous frequency varies with time

自然界中，頻率會隨著時間而改變的例子

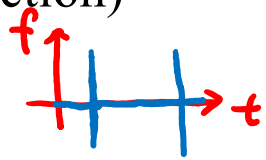
Frequency Modulation

Music

Speech

Others (Animal voice, Doppler effect, seismic waves, radar system, optics, rectangular function)

地震



In fact, in addition to **sinusoid-like functions**, the instantaneous frequencies of other functions will inevitably vary with time.

- Sinusoid Function

$\exp(j\phi(t))$

$\phi(t) = \alpha_1 t + \alpha_0$: sinusoid

$\phi(t) = \alpha_2 t^2 + \alpha_1 t + \alpha_0$: chirp

啁啾聲

- Chirp function

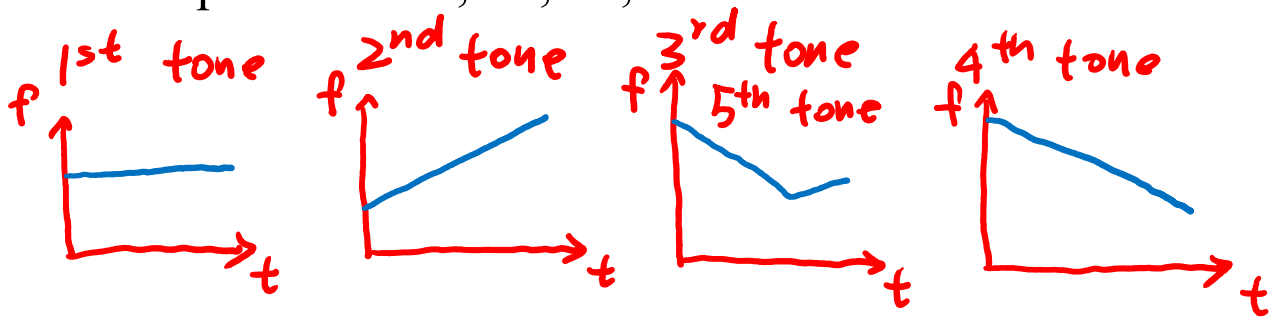
$$\exp\left[j(\alpha_2 t^2 + \alpha_1 t + \alpha_0)\right] \qquad \text{Instantaneous frequency} = \frac{\alpha_2}{\pi} t + \frac{\alpha_1}{2\pi}$$

$\phi'(t) = 2\alpha_2 t + \alpha_1$

$\frac{\phi'(t)}{2\pi}$

acoustics, wireless communication, radar system, optics

Example：the 2nd, 3rd, 4th, and 5th tones



- Higher order exponential function

Example 2

(1) $x(t) = 0.5\cos\left(6400\pi t - 600\pi t^2\right)$ $\quad t \in [0, 3]$

$= 0.25\exp(j(6400\pi t - 600\pi t^2)) + 0.25(\exp(j(-6400\pi t + 600\pi t^2)))$

$\frac{\phi'(t)}{2\pi} = \pm(3200 - 600t)$

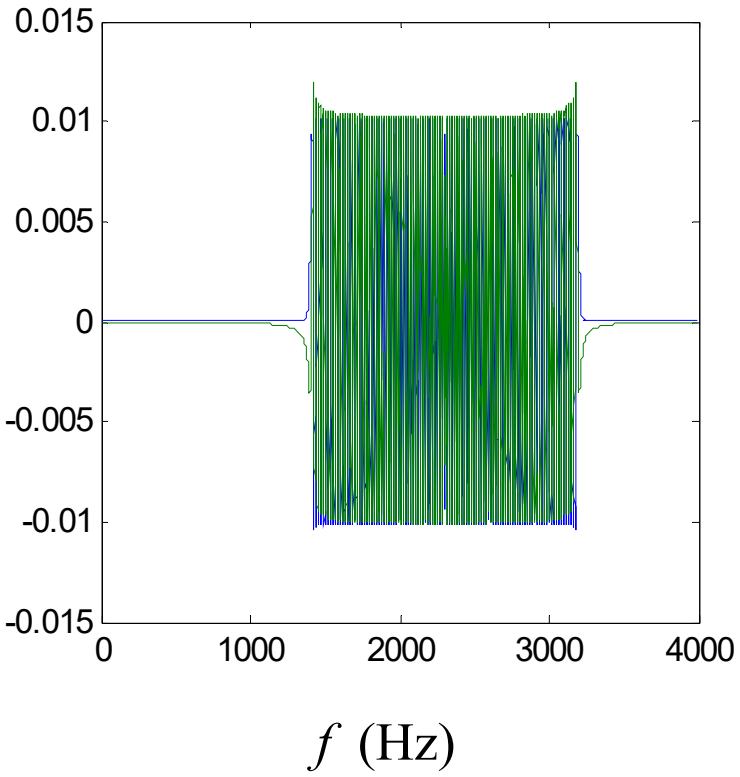
(2) $x(t) = 0.5\cos\left(600\pi t^3 - 2700\pi t^2 + 5050\pi t\right)$ $\quad t \in [0, 3]$

$\frac{\phi'(t)}{2\pi} = \pm(900t^2 - 2700t + 2525)$

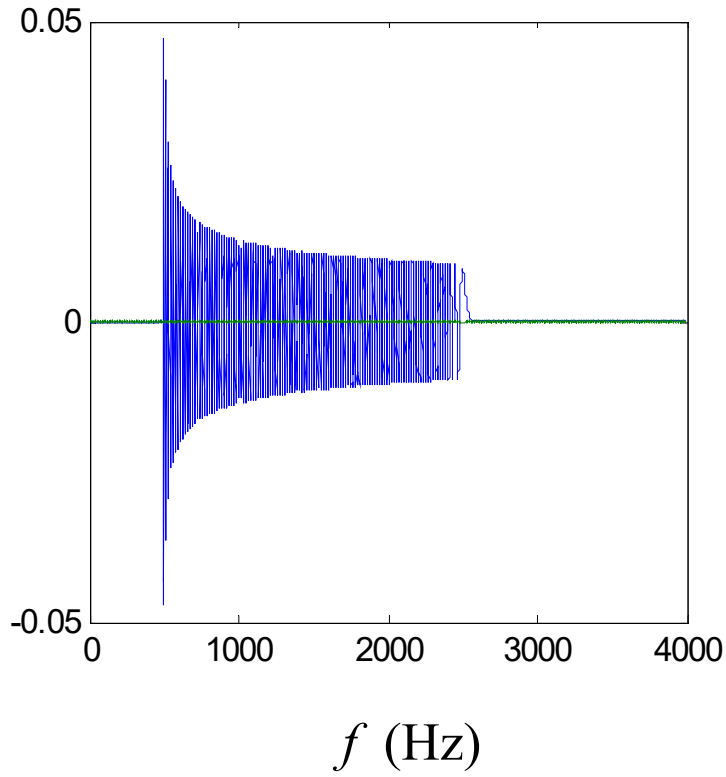
$= \pm(900(t - \frac{3}{2})^2 + 500)$

Fourier transform

$$x(t) = 0.5\cos\left(6400\pi t - 600\pi t^2\right)$$

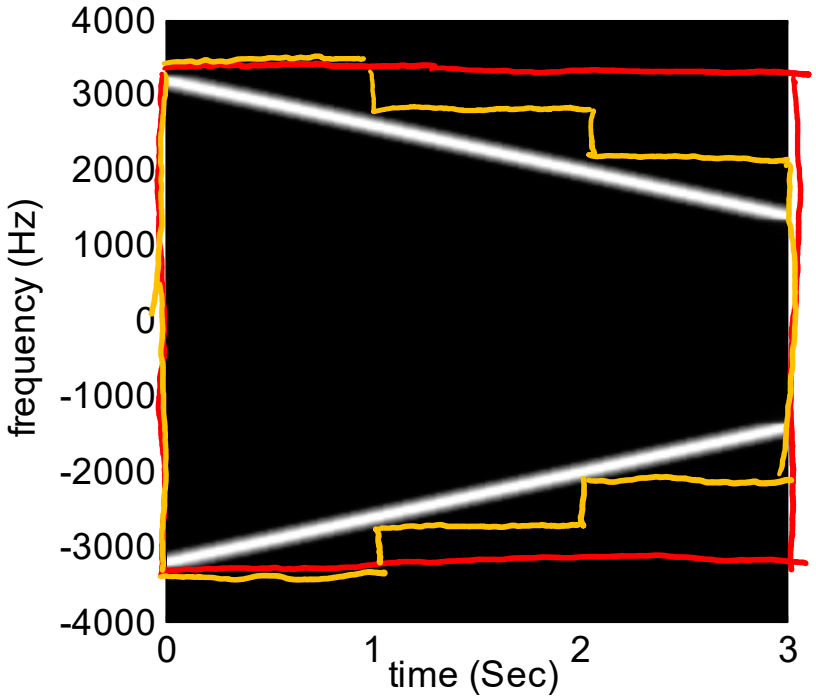


$$x(t) = 0.5\cos\left(600\pi t^3 - 2700\pi t^2 + 5050\pi t\right)$$



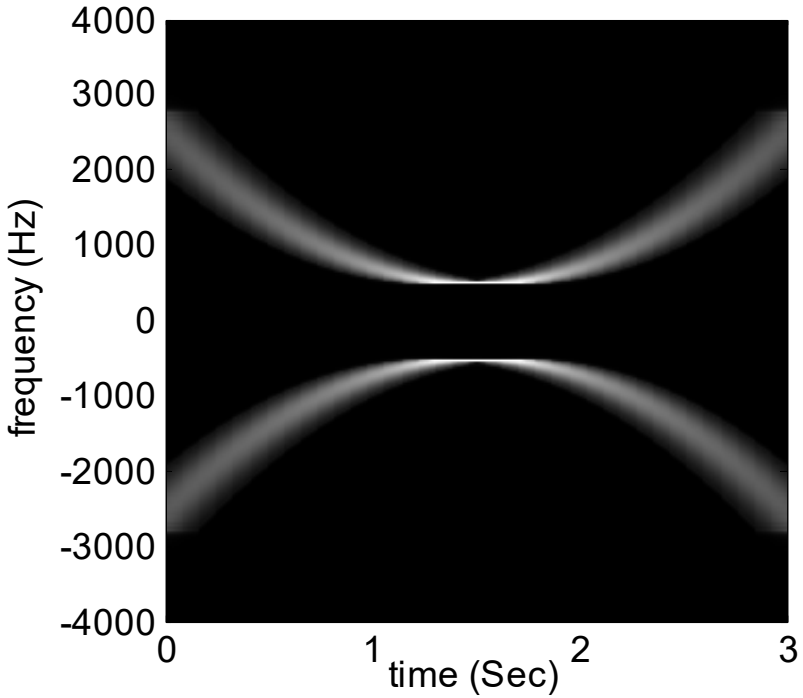
(1)

$$x(t) = 0.5\cos\left(6400\pi t - 600\pi t^2\right)$$



(2)

$$x(t) = 0.5\cos\left(600\pi t^3 - 2700\pi t^2 + 5050\pi t\right)$$



原: 6400x3=19200 points

Adaptive sampling

$\Delta t < \frac{1}{6400}$ $0<t<1$ $\Delta t < \frac{1}{4000}$ $2<t<3$

$\Delta t < \frac{1}{5200}$ $1<t<2$

Example 3

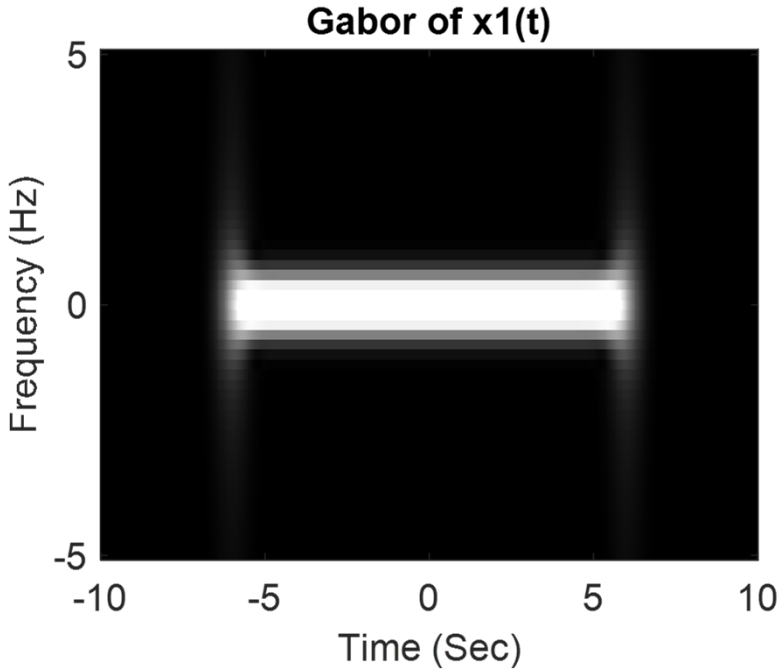
$\Pi\left(\frac{t-t_0}{2B}\right)$ t_0 : center
$2B$: width

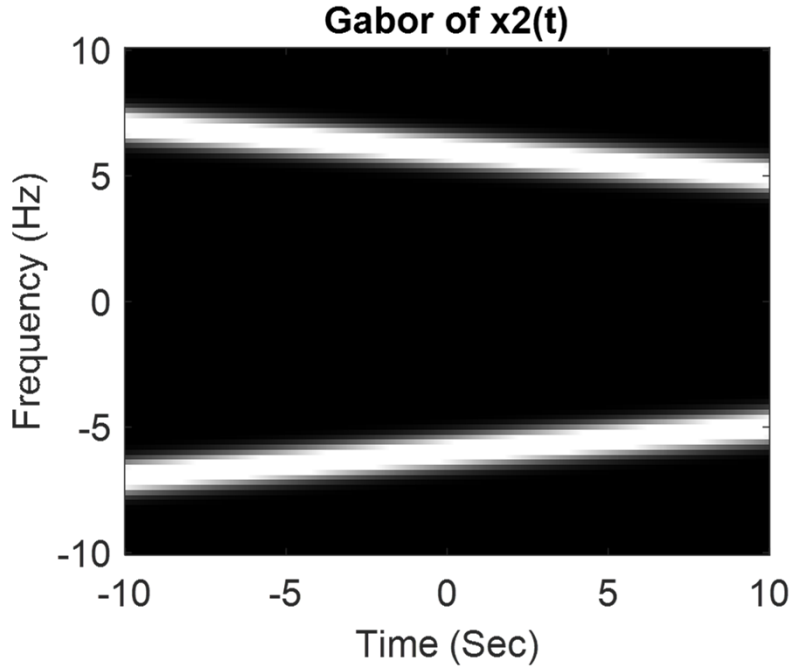
$\Pi\left(\frac{t}{12}\right)$

left: $x_1(t) = 1$ for $|t| \le 6$, $x_1(t) = 0$ otherwise, right: $x_2(t) = \cos(12\pi t - 0.1\pi t^2)$

instantaneous freq. ?

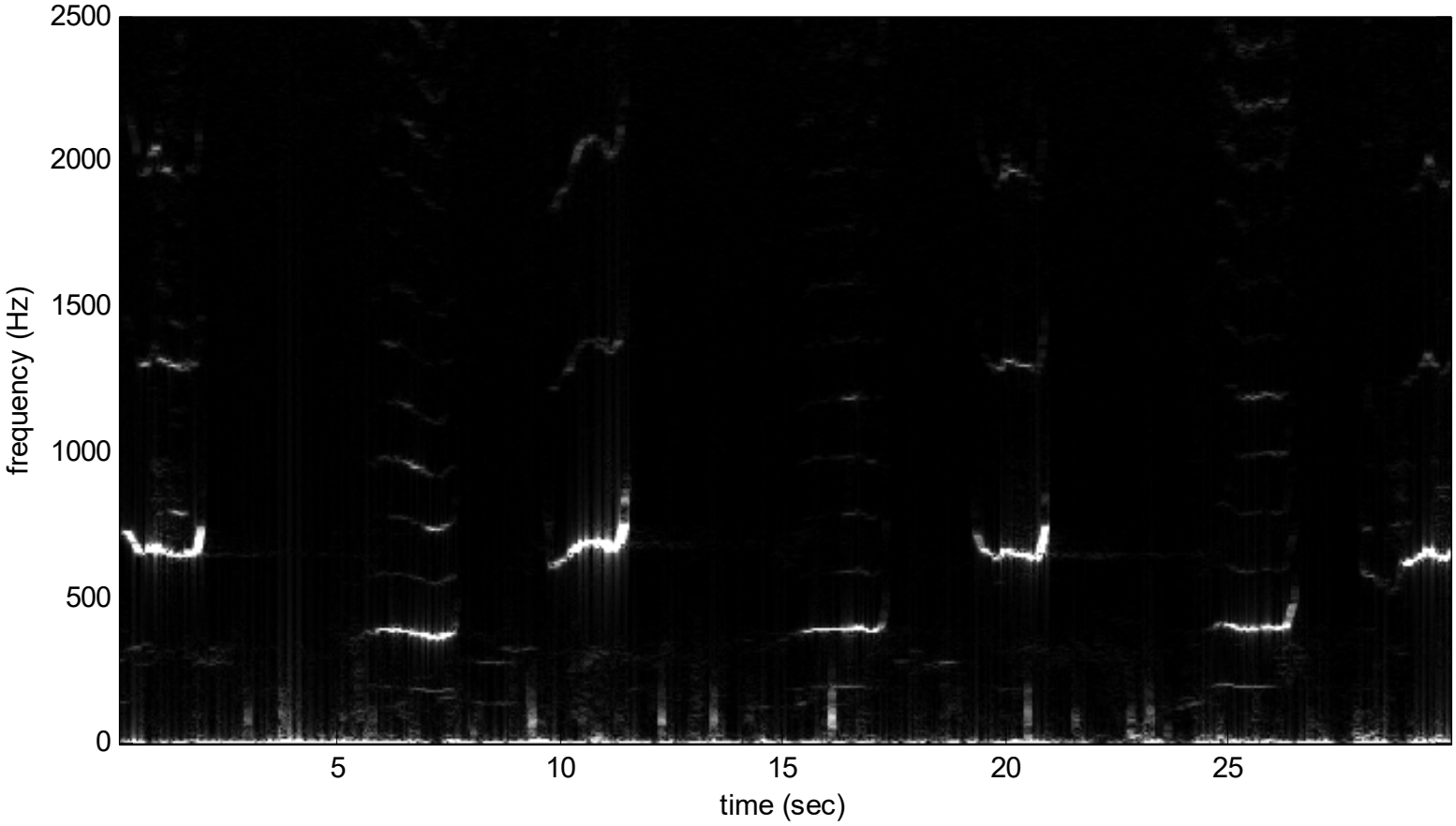
Gabor transform





Example 4

Data source: http://oalib.hlsresearch.com/Whales/index.html



Why Time-Frequency Analysis is Important?

replaced by time-frequency analysis

- Many digital signal processing applications are related to the spectrum or the bandwidth of a signal.

applications of the Fourier transform — spectrum analysis — calculate the convolution

- If the spectrum and the bandwidth can be determined adaptive, the performance can be improved.

- modulation,
- multiplexing,
- filter design,
- data compression,
- signal analysis,
- signal identification,
- acoustics,
- system modeling,
- radar system analysis
- sampling

100 50000

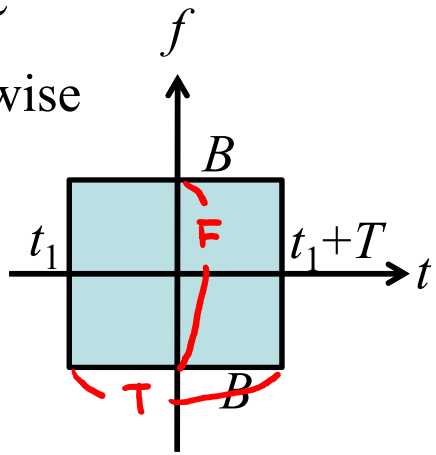
10^{-4} 0.05

Example: Generalization for sampling theory

假設有一個信號，

① The supporting of $x(t)$ is $t_1 \le t \le t_1 + T$, $x(t) \approx 0$ otherwise

② The supporting of $X(f) \ne 0$ is $-B \le f \le B$, $X(f) \approx 0$ otherwise



根據取樣定理， $\Delta_t \le 1/F$, $F = 2B$, B:頻寬

所以，取樣點數 N 的範圍是

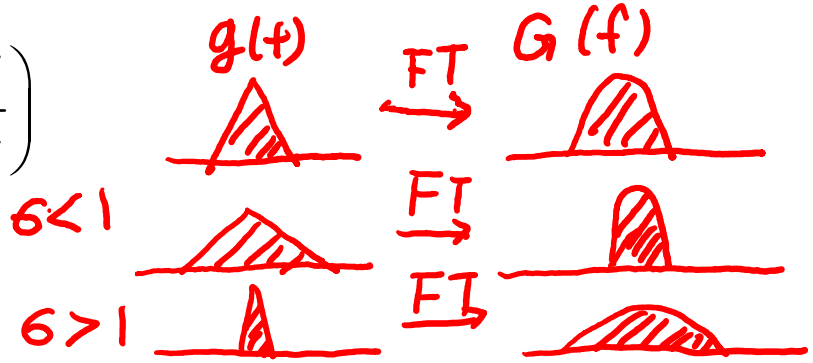
$$N = T/\Delta_t \ge TF$$

重要定理：一個信號所需要的取樣點數的下限，等於它時頻分佈的面積

Q1：Scaling 對於一個信號的取樣點數有沒有影響？

Hint:

$$g(\sigma t) \xrightarrow{FT} \frac{1}{|\sigma|} G\left(\frac{f}{\sigma}\right)$$



Q2: How to use time-frequency analysis to reduce the number of sampling points?

Time-frequency analysis is an efficient tool for adaptive signal processing.

時頻分析大家族

- (1) Short-time Fourier transform (STFT) 1946
  (rec-STFT, Gabor, …)
  - square → spectrogram → generalized spectrogram
  - improve → Asymmetric STFT
  - improve → S transform
  - combine → **Gabor-Wigner Transform**
- (2) Wigner distribution function (WDF) 1932
  - improve → windowed WDF
  - improve → Cohen's Class Distribution (Choi-Williams, Cone-Shape, Page, Levin, Kirkwood, Born-Jordan, …)
  - improve → Polynomial Wigner Distribution
  - improve → Pseudo L-Wigner Distribution
- (3) Time-Variant Basis Expansion
  - → Matching Pursuit and Compressive Sensing
  - → Prolate Spheroidal Wave Function
- (4) Wavelet transform
  - → Haar and Daubechies
  - → Coiflet, Morlet
  - → Directional Wavelet Transform
- (5) Hilbert-Huang Transform (唯一跳脫 Fourier transform 的架構)

小波

• Continuous Wavelet Transform

forward wavelet transform:

ψ psi /saɪ/

$$X(a,b)=\frac{1}{\sqrt{b}}\int_{-\infty}^{\infty}x(t)\psi\left(\frac{t-a}{b}\right)dt$$

$\psi(t)$: mother wavelet, a: location, b: scaling,

some high frequency function

inverse wavelet transform:

large b ↔ lower f
small b ↔ higher f

$$x(t)=\sum_{a}\sum_{b}X(a,b)\varphi_{a,b}(t)$$

$\varphi_{a,b}(t)$ is dual orthogonal to $\psi(t)$.

	output
Fourier transform	$X(f)$, f: frequency
time-frequency analysis	$X(t, f)$, t: time, f: frequency
wavelet transform	$X(a, b)$, a: time, b: scaling

限制：

(1) $$\frac{1}{\sqrt{b}}\int_{-\infty}^{\infty}\varphi_{a_1,b_1}(t)\psi\left(\frac{t-a}{b}\right)dt=1 \qquad \text{when } a_1=a \text{ and } b_1=b,$$

$$\frac{1}{\sqrt{b}}\int_{-\infty}^{\infty}\varphi_{a_1,b_1}(t)\psi\left(\frac{t-a}{b}\right)dt=0 \qquad \text{otherwise}$$

(2) $\psi(t)$ has a finite time interval

Two parameters, a: 調整位置， b: 調整寬度

應用： adaptive signal analysis

思考：需要較高解析度的地方，b 的值應該如何？

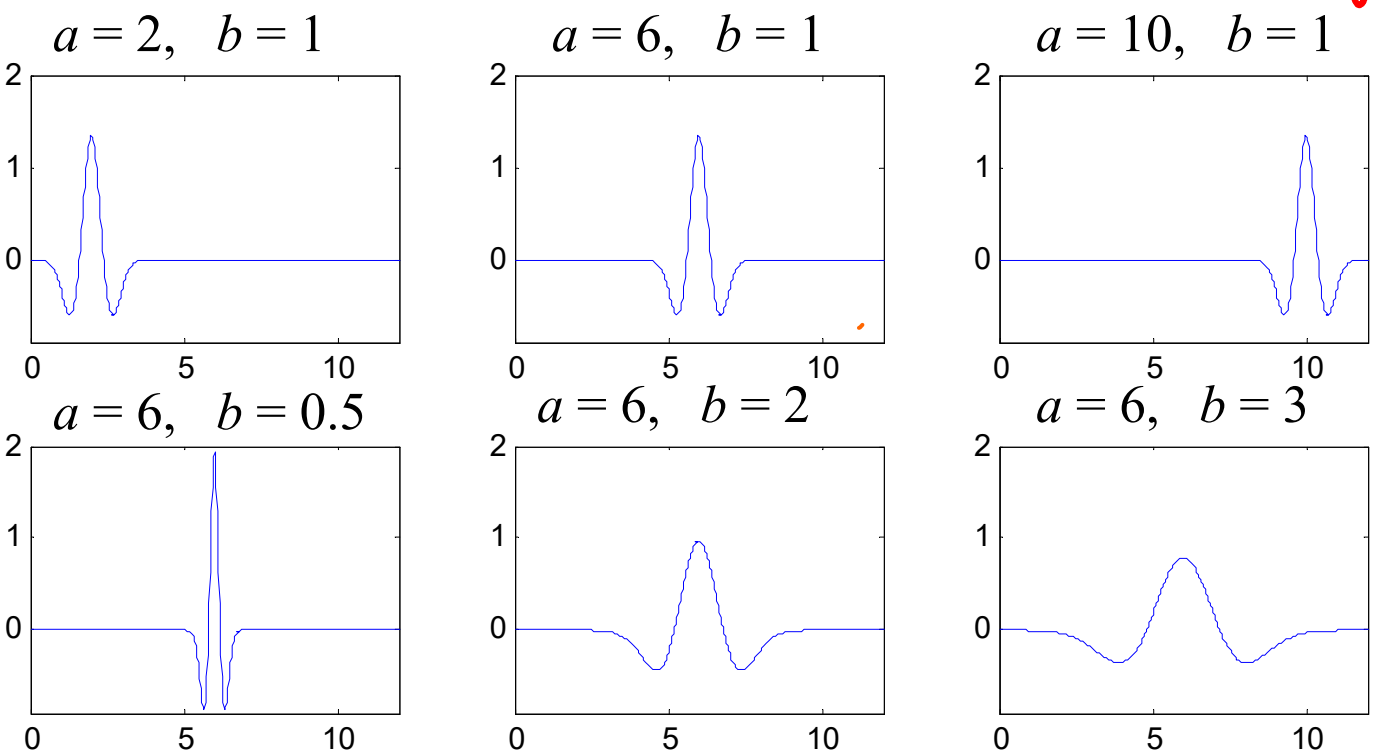
Wavelet 的種類甚多

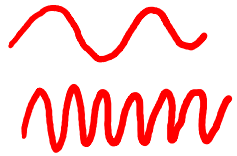
Mexican hat wavelet, Haar Wavelet, Daubechies wavelet, triangular wavelet, ……

例子： Mexican hat wavelet (Laplacian of Gaussian) 隨 a and b 變化之情形

$$\psi(t) = \frac{2^{5/4}}{\sqrt{3}}\left(1-2\pi t^2\right)e^{-\pi t^2} \qquad \frac{1}{\sqrt{b}}\psi\left(\frac{t-a}{b}\right)$$

$$\psi(t) = \left(\frac{d^2}{dt^2} e^{-\pi t^2}\right) \times \frac{2^{\frac{1}{4}}}{\sqrt{3}\pi}$$

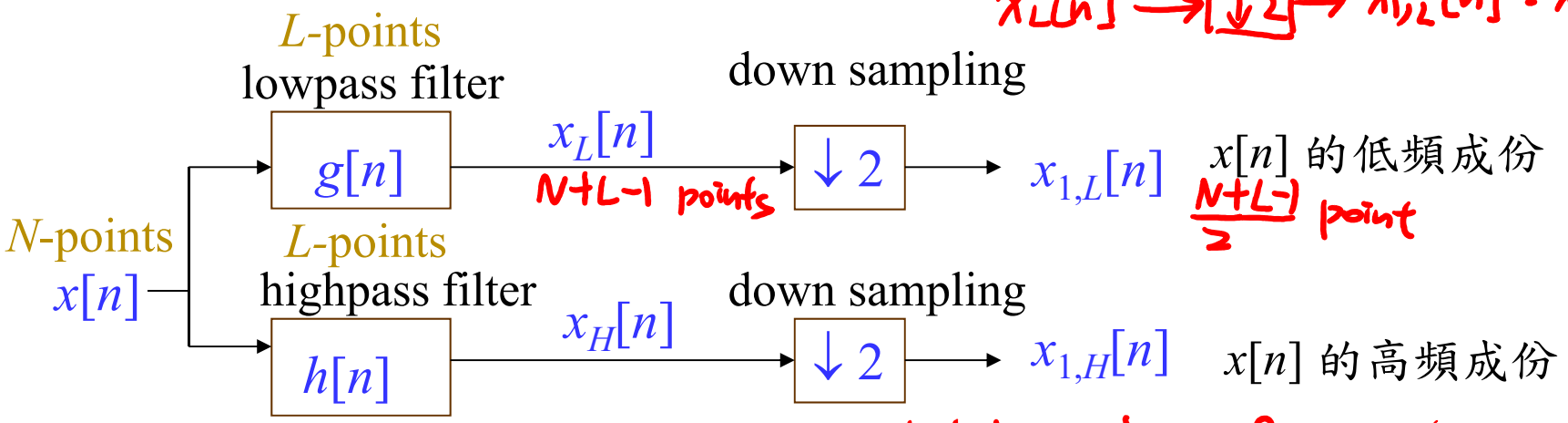




• Discrete Wavelet Transform (DWT)

The discrete wavelet transform is very different from the continuous wavelet transform. It is simpler and more useful than the continuous one.

$x_L[n] \rightarrow \boxed{\downarrow 2} \rightarrow x_{1,L}[n] = x_L[2n]$



total number of points

$\frac{N+L-1}{2} \times 2 = N+L-1$ points $\cong N$ points

$$x_L[n] = \sum_k x[n-k]g[k] \qquad x_{1,L}[n] = \sum_k x[2n-k]g[k]$$

$$x_H[n] = \sum_k x[n-k]h[k] \qquad x_{1,H}[n] = \sum_k x[2n-k]h[k]$$

$$x_{1,L}[n] = \sum_k x[2n-k]g[k] \qquad x_{1,H}[n] = \sum_k x[2n-k]h[k]$$

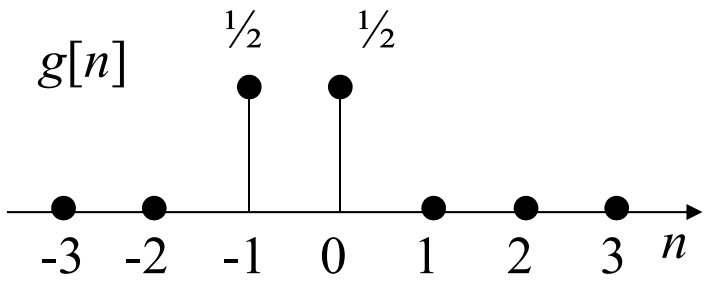
例子：2-point Haar wavelet

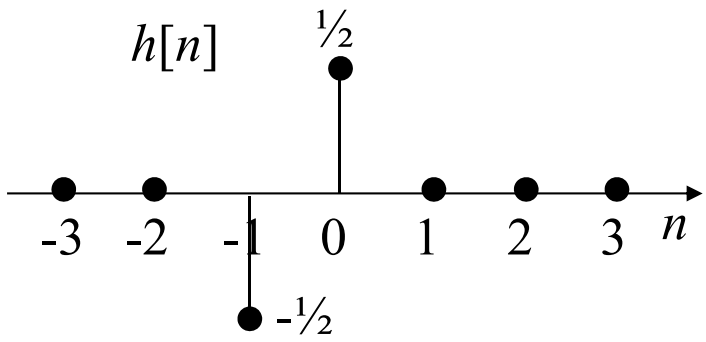
$g[n] = 1/2$ for $n = -1, 0$

$g[n] = 0$ otherwise

$h[0] = 1/2, \quad h[-1] = -1/2,$

$h[n] = 0$ otherwise





then

$$x_{1,L}[n] = \frac{x[2n] + x[2n+1]}{2}$$

（兩點平均）

$$x_{1,H}[n] = \frac{x[2n] - x[2n+1]}{2}$$

（兩點之差）

Remember: For the 2-point DFT $X[m]=\sum_k e^{-j\frac{2\pi mn}{2}}x[n]=\sum_k(-1)^{mn}x[n]$

$$\begin{bmatrix} X[0] \\ X[1] \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} x[0] \\ x[1] \end{bmatrix}$$

一般的 wavelet, $g[n]$ 和 $h[n]$ 點數會多於 2 點

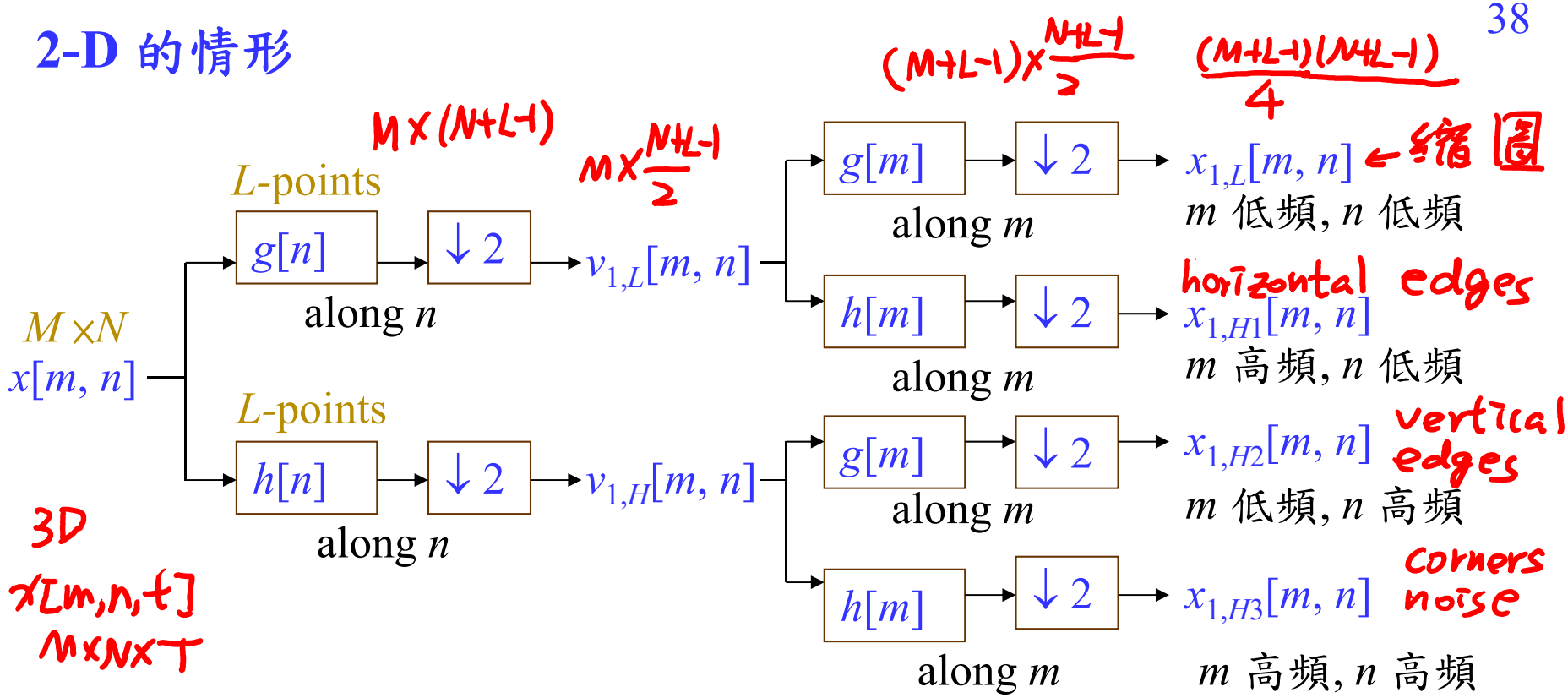
但是 $g[n]$ 通常都是 lowpass filter 的型態

$h[n]$ 通常都是 highpass filter 的型態

Discrete wavelet transform 有很多種

(discrete Haar wavelet, discrete Daubechies wavelet, B-spline DWT, symlet, coilet, ……..)

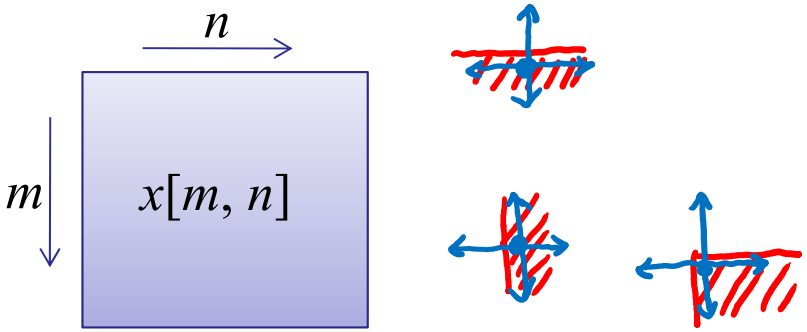
2-D 的情形



3D

$x[m,n,t]$

$M \times N \times T$



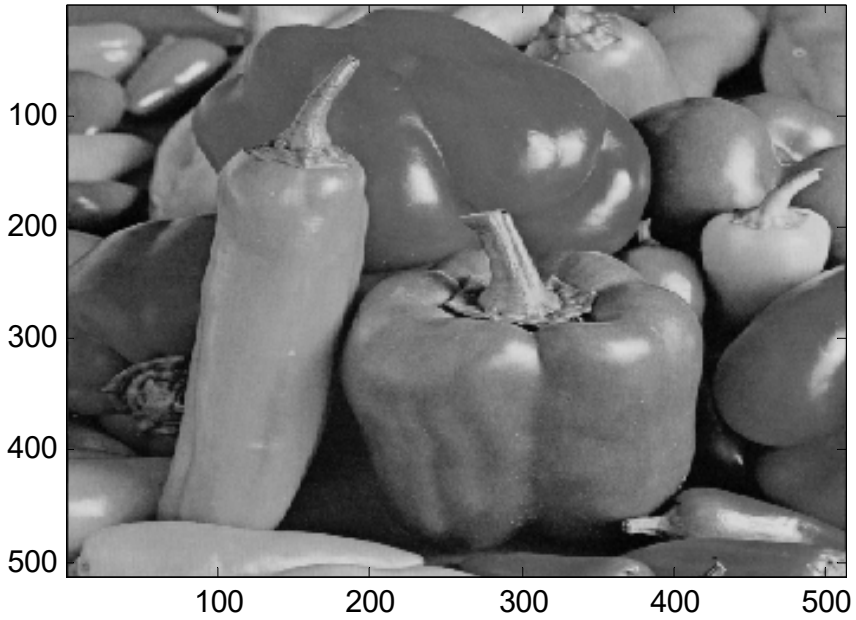
total number of outputs

$\frac{(M+L-1)(N+L-1)}{4} \times 4 = (M+L-1)(N+L-1)$

$\cong MN$

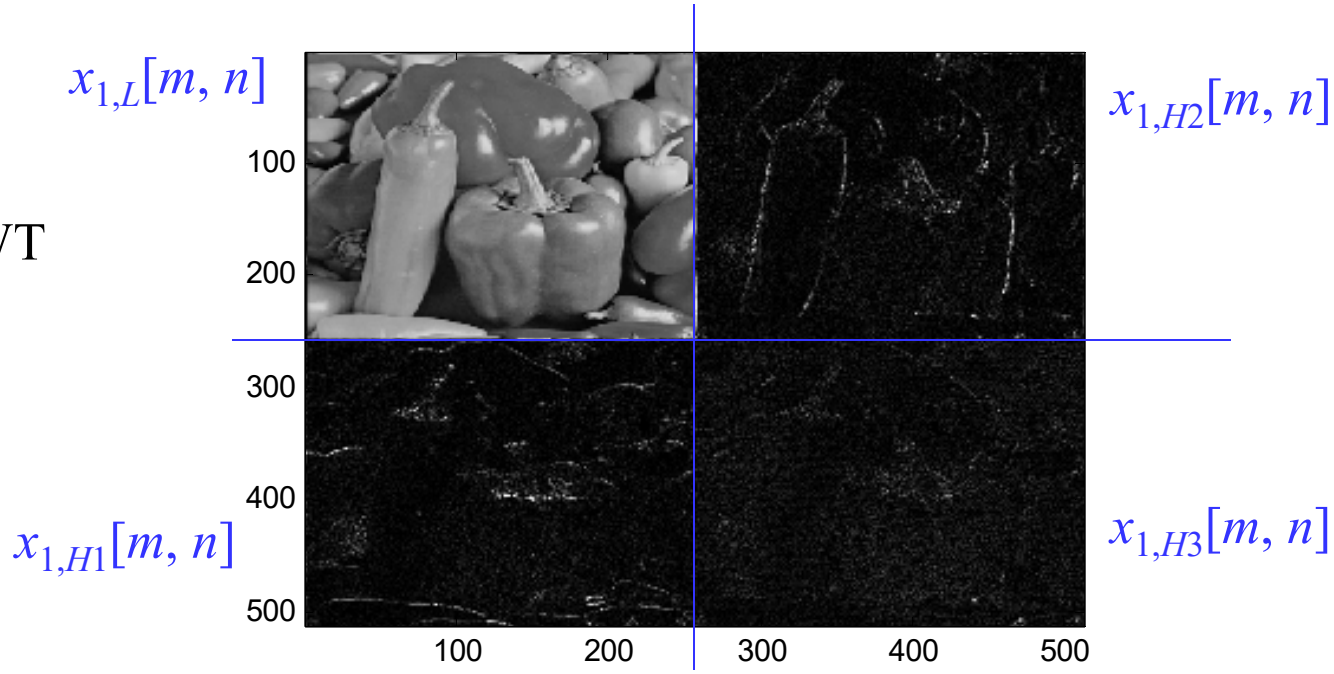
原影像

peppers.bmp

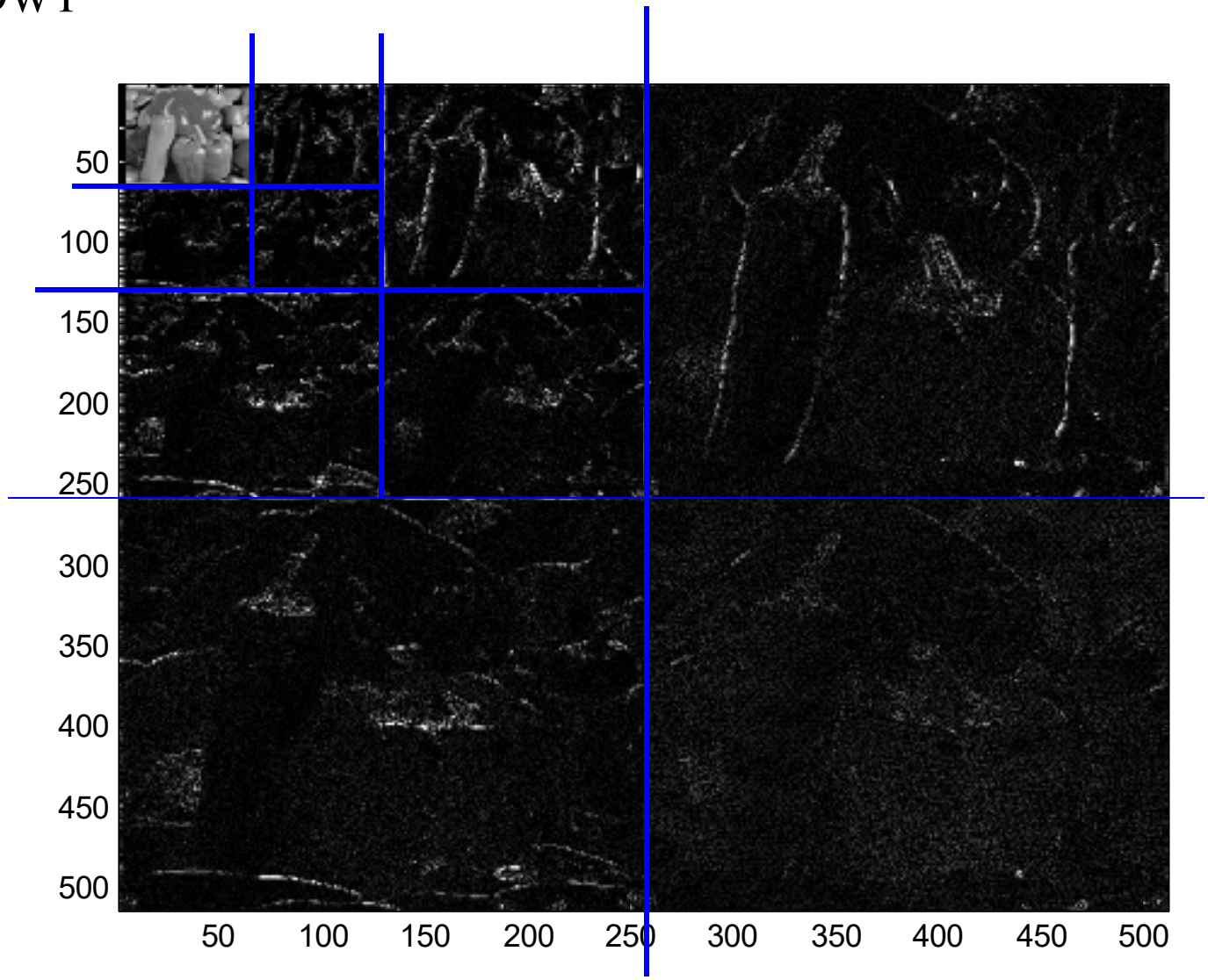


2-D DWT

的結果



3次2-D DWT
的結果


50
100
150
200
250
300
350
400
450
500
50
100
150
200
250
300
350
400
450
500

	gray	color
應用： 影像壓縮 (JPEG 2000) based on wavelet	$\frac{1}{50}$	$\frac{1}{100}$
JPEG based on DCT (discrete cosine transform)	$\frac{1}{15}$	$\frac{1}{30}$

其他應用：edge detection

corner detection

filter design

pattern recognition

music signal processing

economical data

temperature analysis

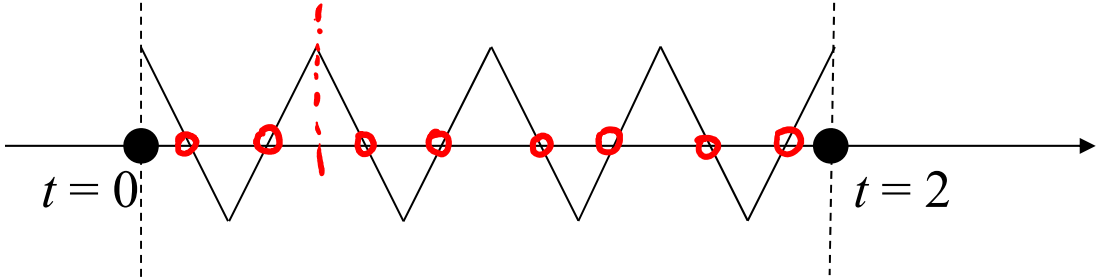
feature extraction

biomedical signal processing

• Hilbert Huang Transform

Another way to determine the frequency (without the FT)

frequency = 2



1 period = 2 zero crossing

$$frequency = \frac{number\ of\ zero\ crossings}{2 \times (time\ duration)} = \frac{8}{2 \times 2} = 2$$

附錄一：聲音檔的處理 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

- 思考: 所以，取樣間隔多大?

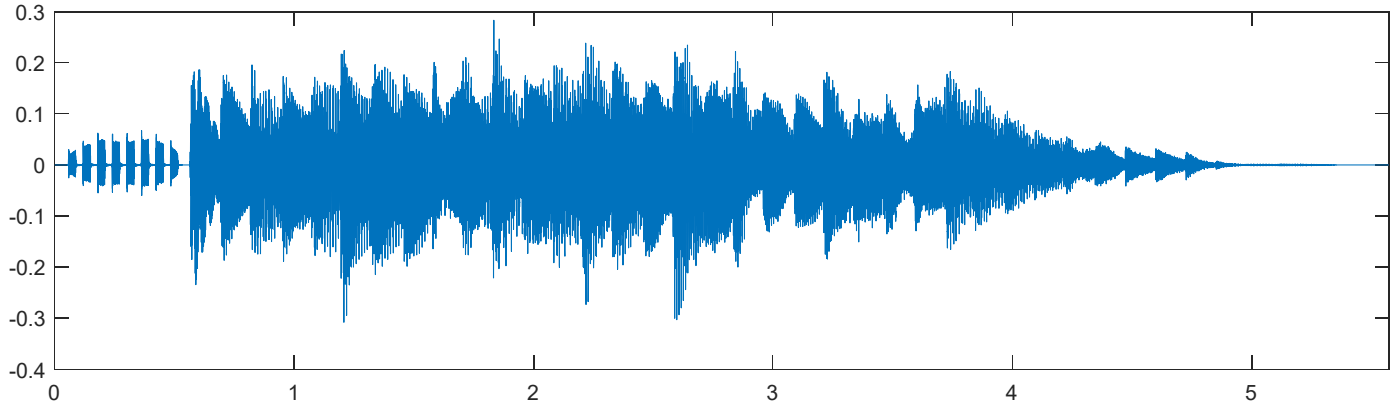
- 這個聲音檔有多少秒？

雙聲道 (**Stereo**，俗稱立體聲)

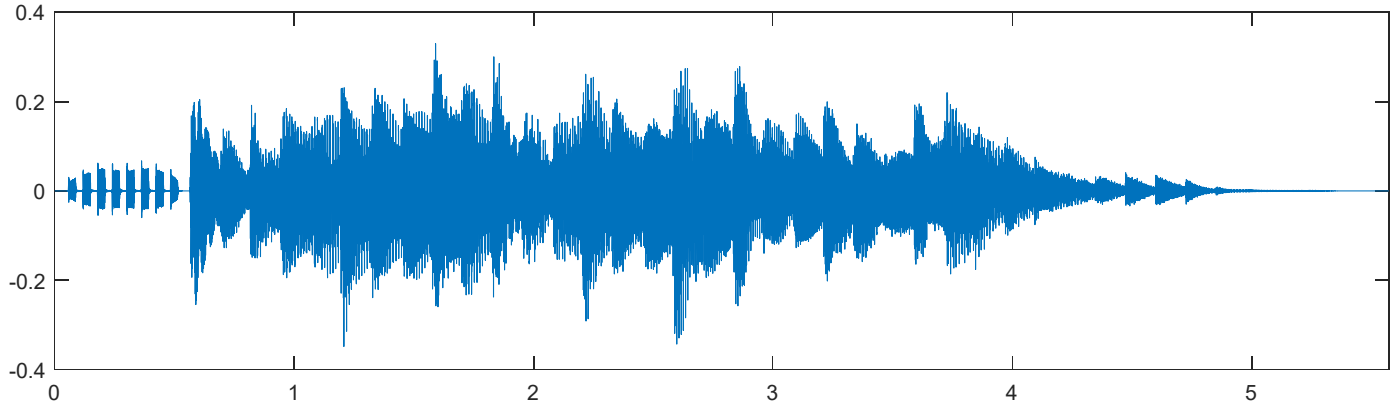
畫出聲音的波型

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)])



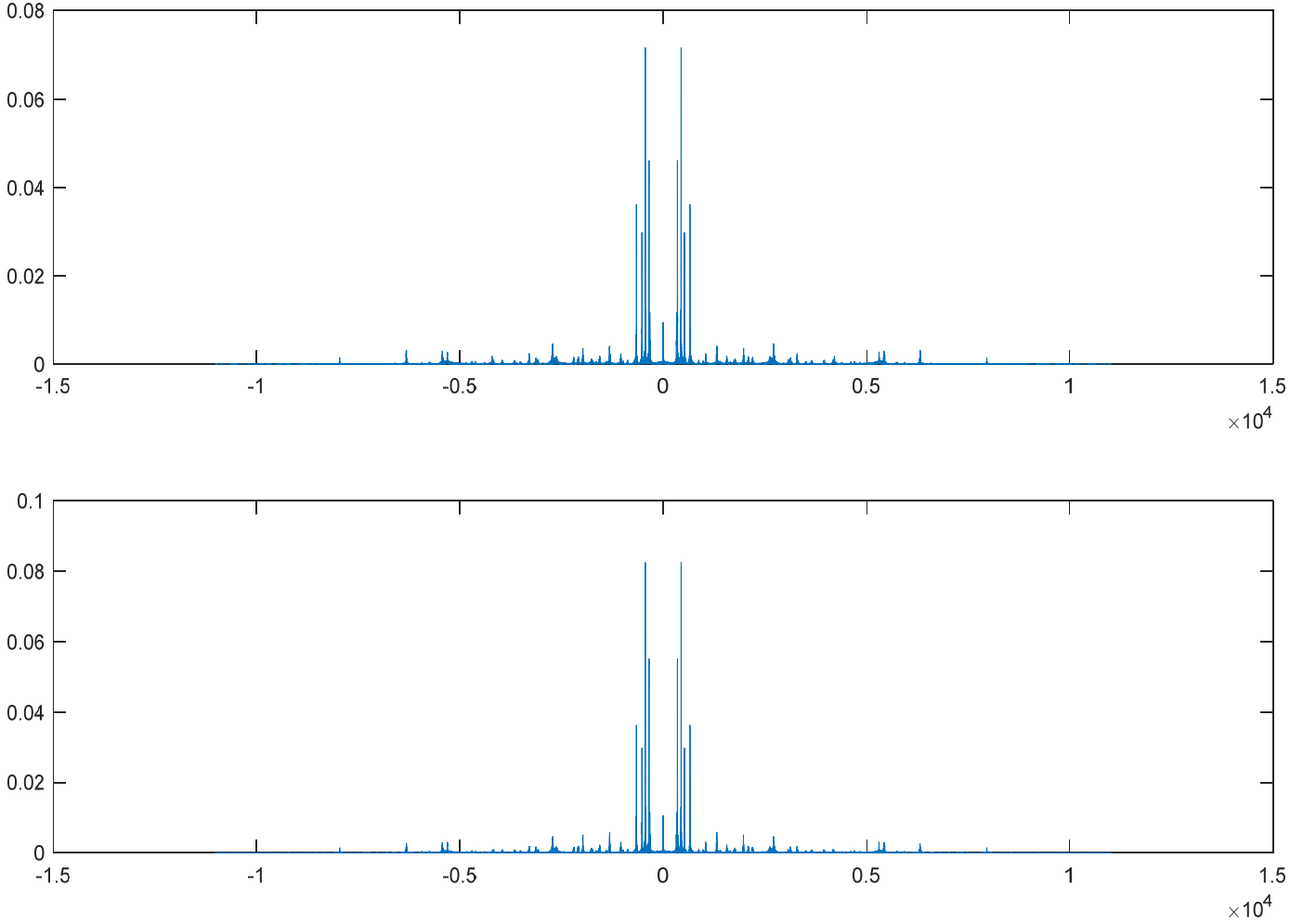
注意： *.wav 檔中所讀取的資料，值都在 –1 和 +1 之間

B. 繪出頻譜

X = fft(x(:,1)); % 只做這一步無法得出正確的頻譜

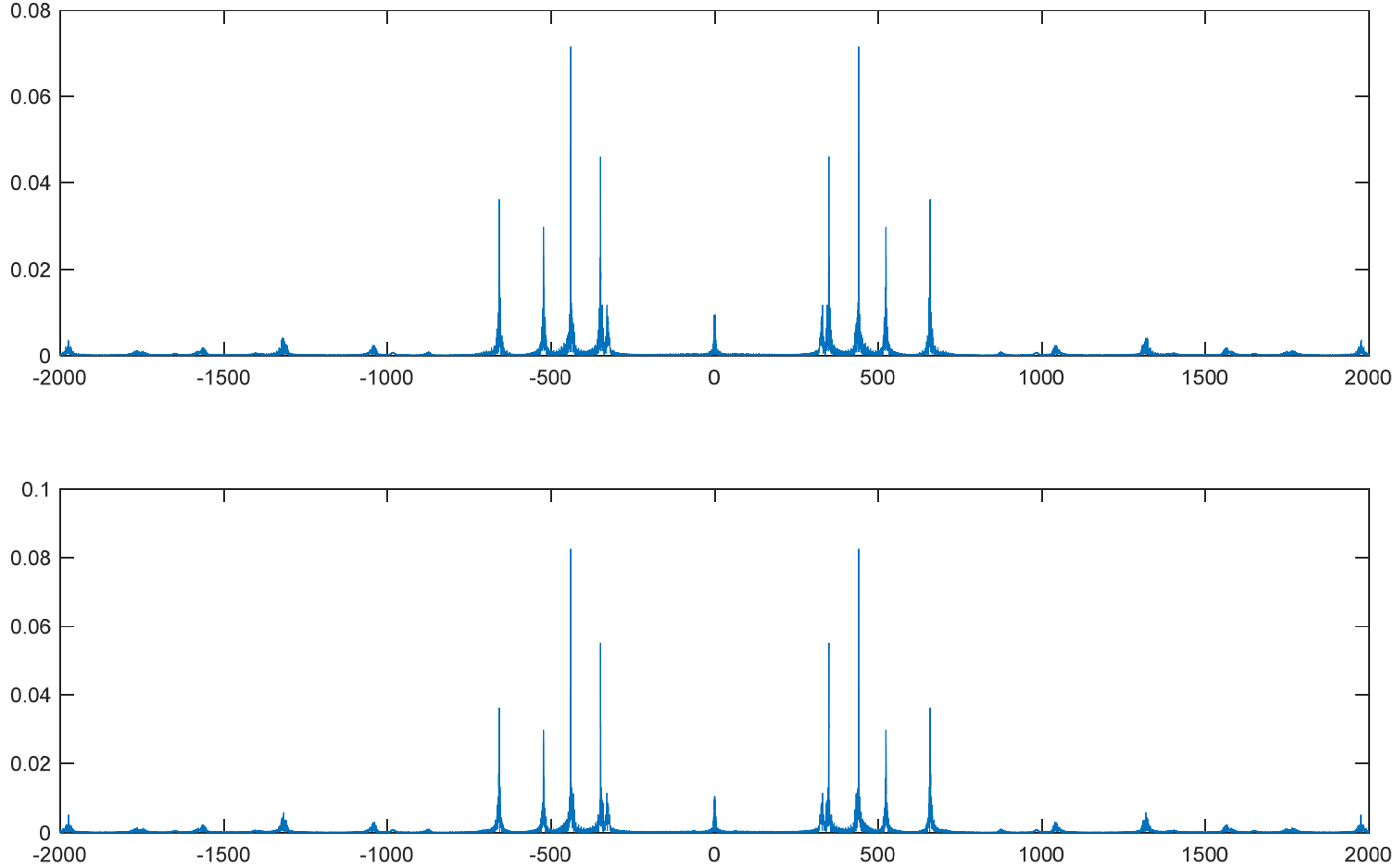
```
X=X.';
N=length(X); N1=round(N/2);
dt=1/fs;
X1=[X(N1+1:N),X(1:N1)]*dt;  % shifting for spectrum
f=[[N1:N-1]-N,0:N1-1]/N*fs;    % valid f
plot(f, abs(X1));
```

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. 製作音檔： 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. 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)：

```
audioplay(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(audioarray, Fs, ‘test.wav’)   % 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

附錄二 使用 Python 處理音訊的方法

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
```

後面將說明使用 Python 讀檔，畫出頻譜，撥放聲音，製作音檔，錄音的方法

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 wave

讀取音檔：

wavefile = wave.open('C:/WINDOWS/Media/Alarm01.wav', 'rb‘)

獲得音檔取樣頻率和音訊長度:

```
fs =wavefile.getframerate()       # sampling frequency
num_frame = wavefile.getnframes()  # length of the vocal signal
```

```
>>> fs
22050

>>> num_frame
122868
```

讀取波形與數據

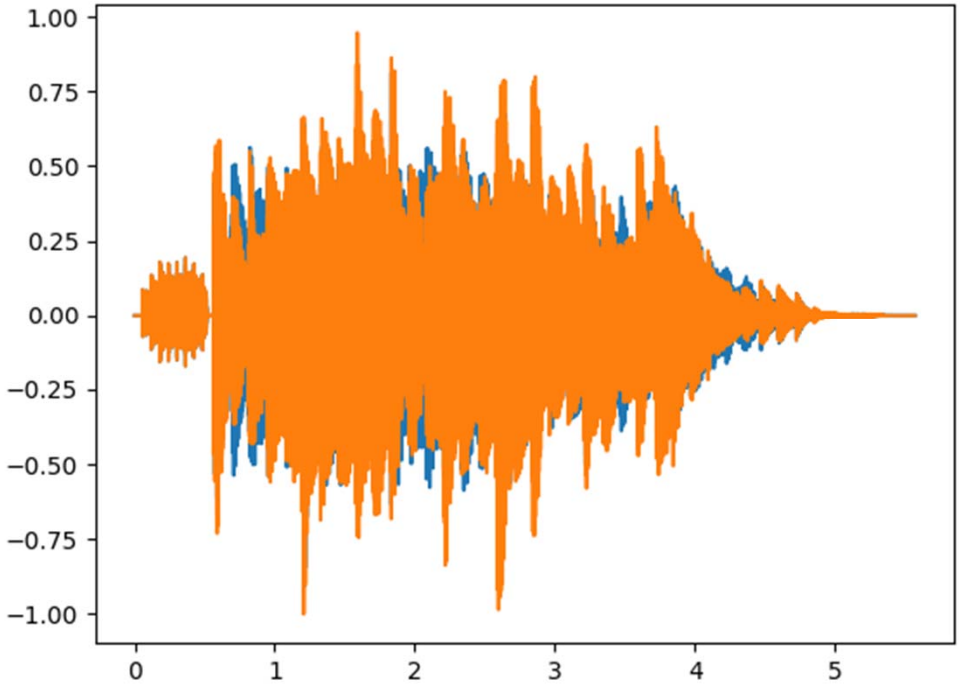
接續前頁，
要先import 相關模組： import numpy as np

- str_data = wavefile.readframes(num_frame)
- wave_data = np.frombuffer(str_data, dtype=np.int16)
  # 轉成整數型態
- wave_data = wave_data / max(abs(wave_data)) # normalization
- n_channel = 2
- wave_data = np.reshape(wave_data, (num_frame, n_channel))
  # 若為雙聲道音檔需要做 reshape

畫出音訊波形圖

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

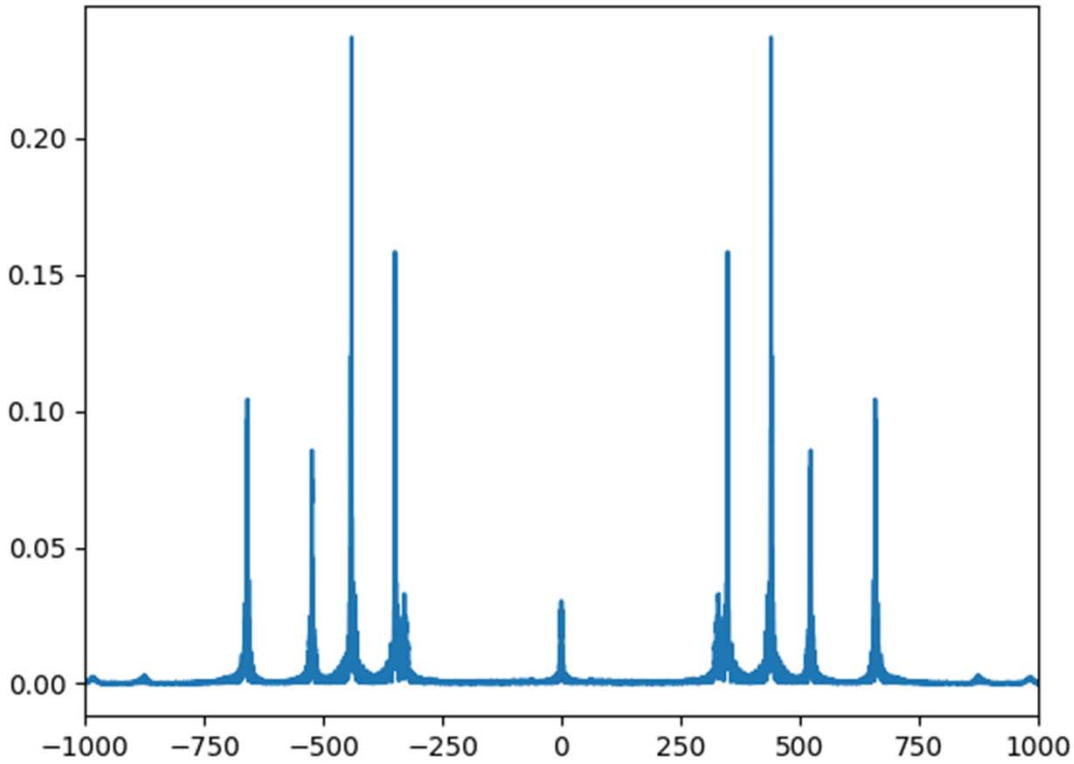
- time = np.arange(0, num_frame)*1/fs
- plt.plot(time, wave_data)
- plt.show()



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

- f = wave.open('testing.wav', 'wb')
- f.setnchannels(2) # 設定聲道數
- f.setsampwidth(2) # 每個 samples 有幾個位元組
- f.setframerate(fs) # 設定取樣頻率
- f.writeframes(wave_data.tobytes())
- f.close()

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/

II. Short-time Fourier Transform

II-A Definition

Short-time Fourier transform (STFT)

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

Alternative definition

$$X(t,\omega)=\int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{-j\omega\tau}d\tau$$

$\omega=2\pi f$

參考資料

[1] S. Qian and D. Chen, Section 3-1 in *Joint Time-Frequency Analysis: Methods and Applications*, Prentice-Hall, 1996.

[2] S. H. Nawab and T. F. Quatieri, "Short time Fourier transform," in *Advanced Topics in Signal Processing*, pp. 289-337, Prentice Hall, 1987.

STFT $$X(t,f)=\int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{-j2\pi f\tau}d\tau$$

$$X(t,\omega)=\int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{-j\omega\tau}d\tau$$

Inverse of the STFT: To recover $x(t)$,

$$x(t)=w^{-1}(t_1-t)\int_{-\infty}^{\infty} X(t_1,f)e^{j2\pi f t}df$$

where $w(t_1 - t) \neq 0$.

For the alternative definition, the inverse transform is:

$$x(t)=\frac{1}{2\pi}w^{-1}(t_1-t)\int_{-\infty}^{\infty} X(t_1,\omega)e^{j\omega t}d\omega$$

The mask function $w(t)$ always has the property of

(a) even: $w(t) = w(-t)$, (通常要求這個條件要滿足)

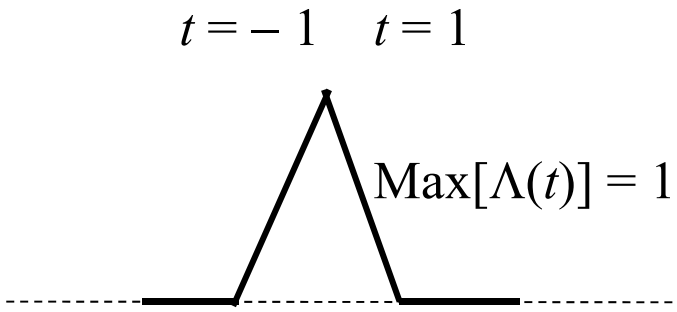
(b) $\max(w(t)) = w(0)$, $w(t_1) \geq w(t_2)$ if $|t_2| > |t_1|$

(c) $w(t) \approx 0$ when $|t|$ is large

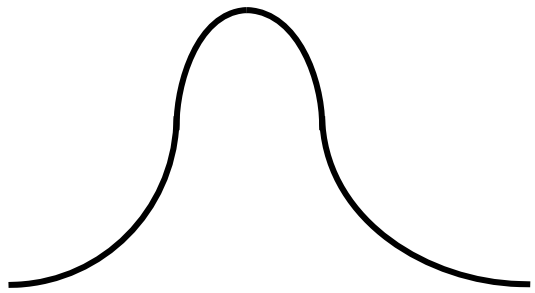
$e^{-\pi(1.9143)^2} = 10^{-5}$

when $|t| > ?$
$w(t) < 10^{-5}$

$w(t) = \Lambda(t)$ (triangular function)



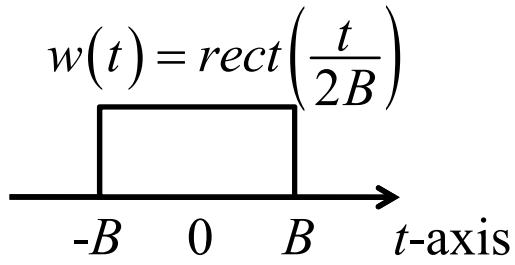
$w(t) = \exp(-a|t|^b)$
(hyper-Laplacian function)



II-B Rec-STFT

Rectangular mask STFT (rec-STFT)

$$X(t,f)=\int_{t-B}^{t+B} x(\tau)e^{-j2\pi f\tau}d\tau$$



Inverse of the rec-STFT

$$x(t)=\int_{-\infty}^{\infty} X(t_1,f)e^{j2\pi f t}df$$

where $t-B<t_1<t+B$

The simplest form of the STFT

Other types of the STFT may require more computation time than the rec-STFT.

II-C Properties of the Rec-STFT

(1) Integration (recovery):

(a) $\int_{-\infty}^{\infty} X(t,f)e^{j2\pi f v}\,df = x(v)$ when $v - B < t < v + B$,

$= 0$ otherwise

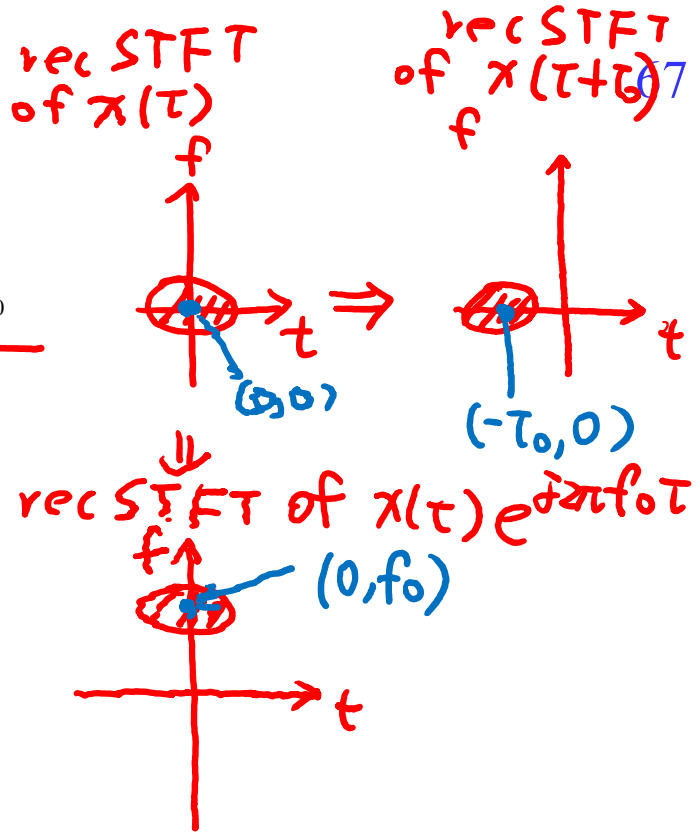
(b) $$\int_{-\infty}^{\infty} X(t,f)\,df = \int_{t-B}^{t+B} x(\tau)\int_{-\infty}^{\infty} e^{-j2\pi f \tau}df\,d\tau$$

$$= \int_{t-B}^{t+B} x(\tau)\delta(\tau)\,d\tau$$

$$= \begin{cases} x(0) & \text{when } t - B < 0 < t + B, \quad -B < t < B \\ 0 & \text{otherwise} \end{cases}$$

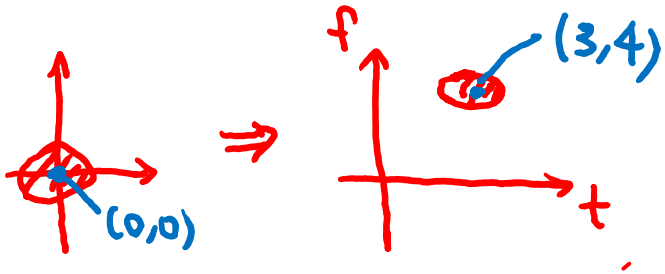
(2) Shifting property (橫的方向移動)

$$\int_{t-B}^{t+B} x(\tau+\tau_0)e^{-j2\pi f\tau}d\tau = X(t+\tau_0, f)e^{j2\pi f\tau_0}$$



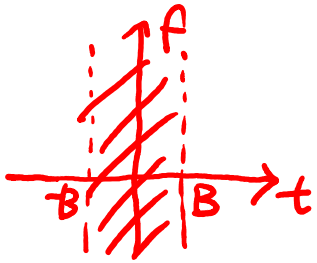
(3) Modulation property (縱的方向移動)

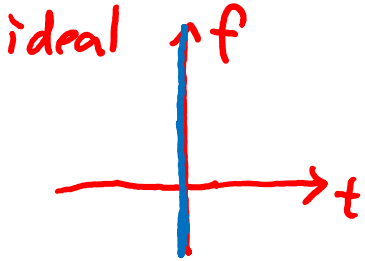
$$\int_{t-B}^{t+B} [x(\tau)e^{j2\pi f_0\tau}]e^{-j2\pi f\tau}d\tau = X(t, f-f_0)$$



$x(\tau-3)e^{j8\pi\tau}$

(4) Special inputs:





better result for small B

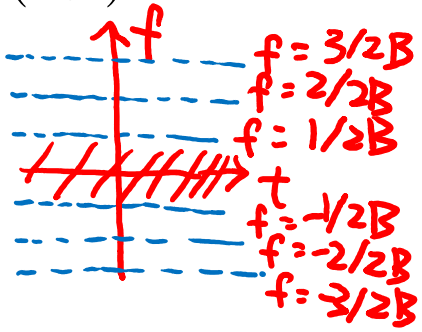
(1) When $x(t) = \delta(t)$,

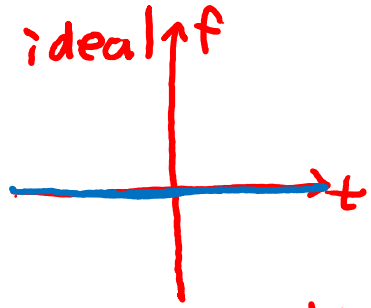
$$X(t,f) = 1 \text{ when } -B < t < B, \quad X(t,f) = 0 \text{ otherwise}$$

(2) When $x(t) = 1$

$$X(t,f) = 2B\,\mathrm{sinc}(2Bf)e^{-j2\pi f t}$$

$$|X(t,f)| = 2B\,|\mathrm{sinc}(2Bf)|$$





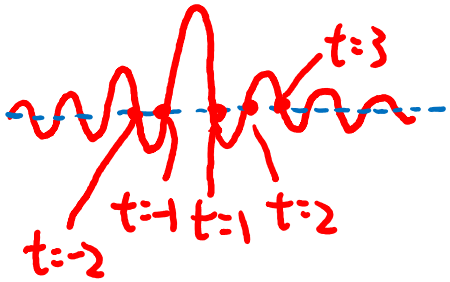
better result for large B

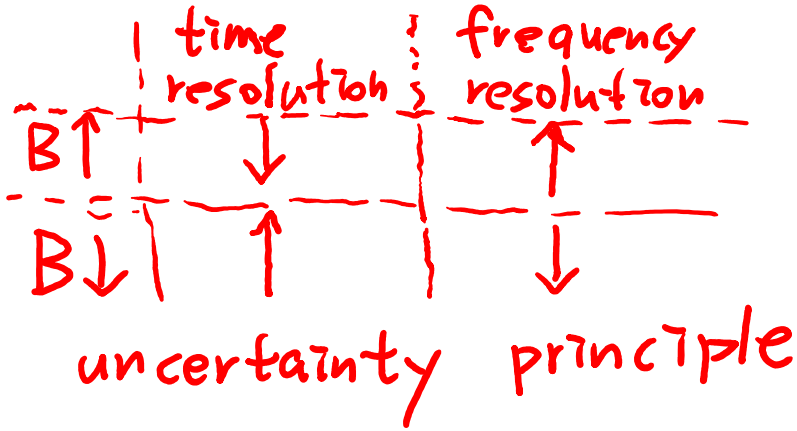
思考： B 值的大小，對解析度的影響是什麼？

$$\mathrm{sinc}\,t = \frac{\sin \pi t}{\pi t}$$

$\mathrm{sinc}\,t = 0$ when $t = \pm 1, \pm 2, \pm 3 \cdots$

$$\mathrm{sinc}\,0 = \lim_{t \to 0} \frac{\sin \pi t}{\pi t} = \lim_{t \to 0} \frac{\pi \cos \pi t}{\pi} = 1$$





(5) Linearity property

If $h(t) = \alpha\, x(t) + \beta\, y(t)$ and $H(t, f)$, $X(t, f)$ and $Y(t, f)$ are their rec-STFTs, then

$$H(t, f) = \alpha\, X(t, f) + \beta\, Y(t, f).$$

(6) Power integration property

$$\int_{-\infty}^{\infty} \left| X(t, f) \right|^2 df = \int_{t-B}^{t+B} \left| x(\tau) \right|^2 d\tau$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| X(t, f) \right|^2 df\, dt = 2B \int_{-\infty}^{\infty} \left| x(\tau) \right|^2 d\tau$$

(7) Energy sum property (Parseval's theorem)

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(t, f) Y^*(t, f) df\, dt = 2B \int_{-\infty}^{\infty} x(\tau) y^*(\tau) d\tau$$

$$\int_{-\infty}^{\infty} X(t, f) Y^*(t, f) df = \int_{t-B}^{t+B} x(\tau) y^*(\tau) d\tau$$

思考:

(1) 哪些性質 Fourier transform 也有？

(2) 其他型態的 STFT 是否有類似的性質？

Shifting

$$\int_{-\infty}^{\infty} w(t-\tau)x(\tau-\tau_0)e^{-j2\pi f\tau}d\tau$$

$$=\int_{-\infty}^{\infty} w(t-\tau-\tau_0)x(\tau)e^{-j2\pi f\tau}e^{-j2\pi f\tau_0}d\tau$$

$$=X(t-\tau_0,f)e^{-j2\pi f\tau_0}$$

Modulation

$$\int_{-\infty}^{\infty} w(t-\tau)[x(\tau)e^{j2\pi f_0\tau}]e^{-j2\pi f\tau}d\tau = X(t,f-f_0)$$

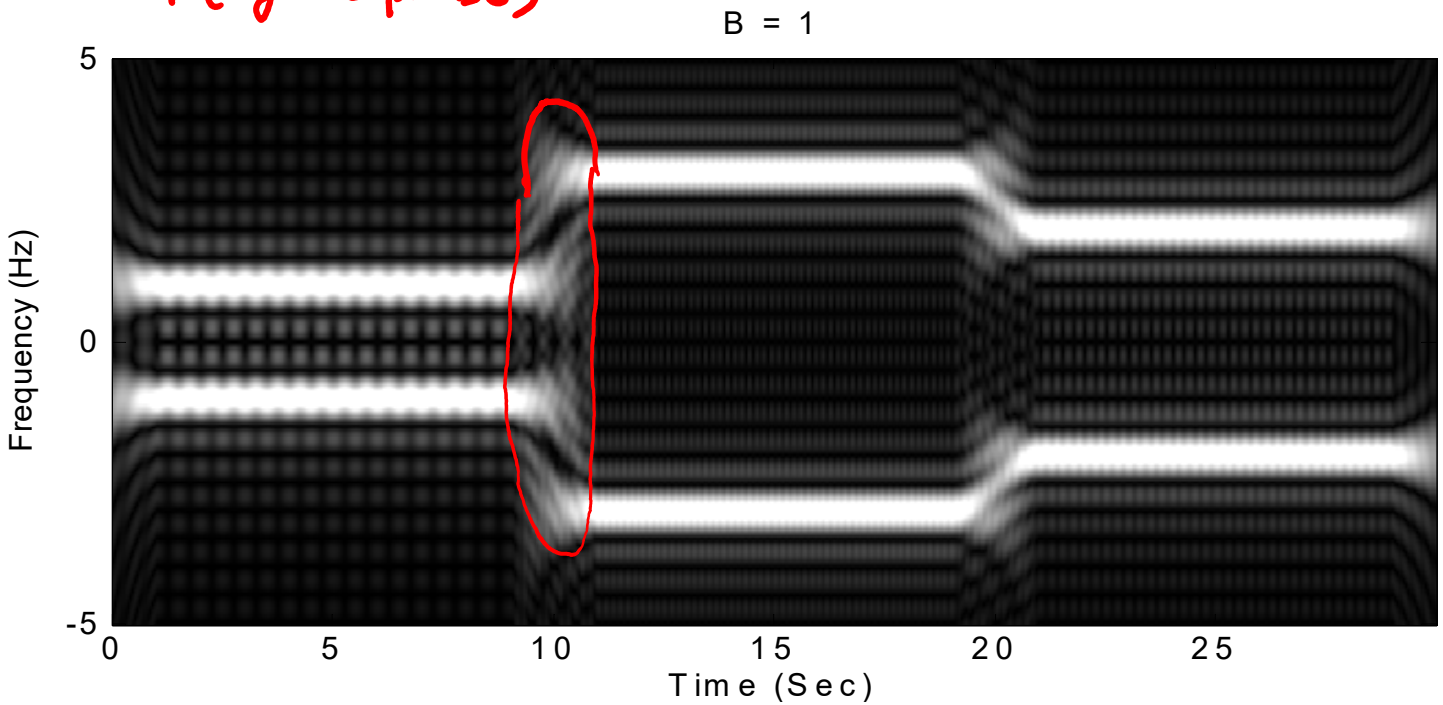
$\frac{1}{2}e^{j2\pi t} + \frac{1}{2}e^{-j2\pi t}$

Example: $x(t) = \cos(2\pi t)$ when $t < 10$, ± 1Hz

$x(t) = \cos(6\pi t)$ when $10 \le t < 20$, ± 3Hz

$x(t) = \cos(4\pi t)$ when $t \ge 20$ ± 2Hz

$|X(t,f)|$ (ignore phase)



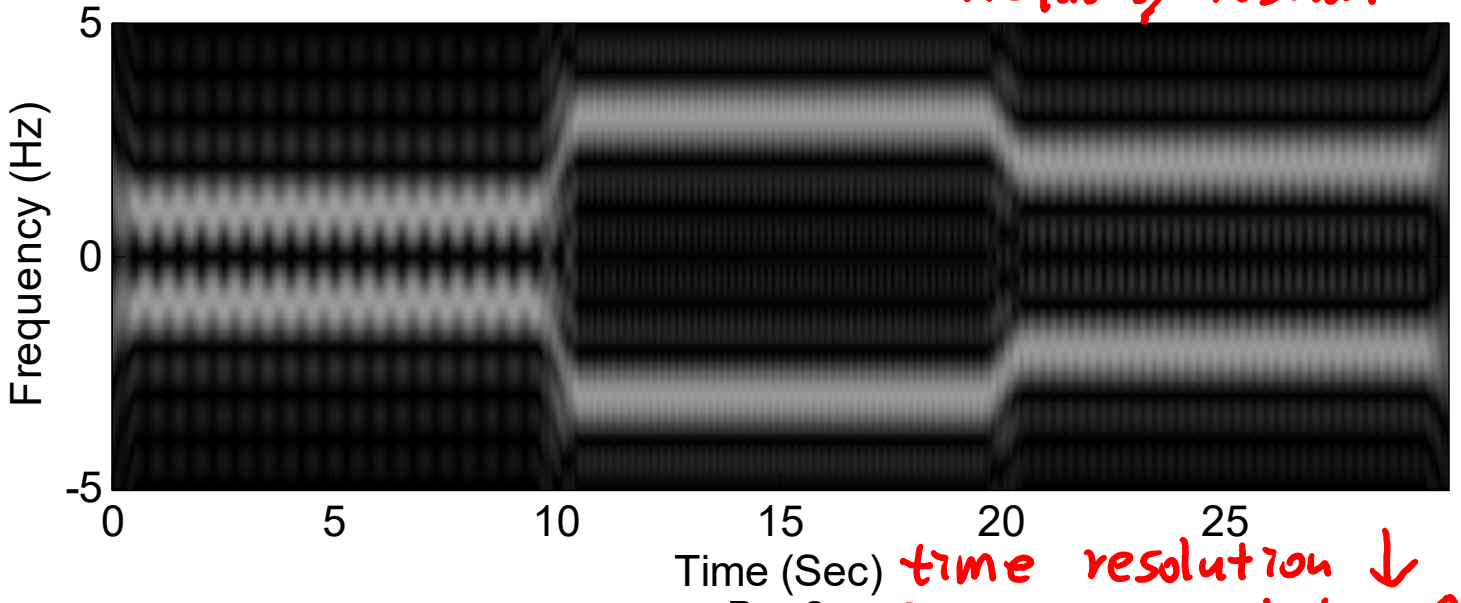
side lobe effect

$\Pi\left(\frac{t}{2B}\right) \xrightarrow{FT} 2B\,\mathrm{sinc}(2Bf)$

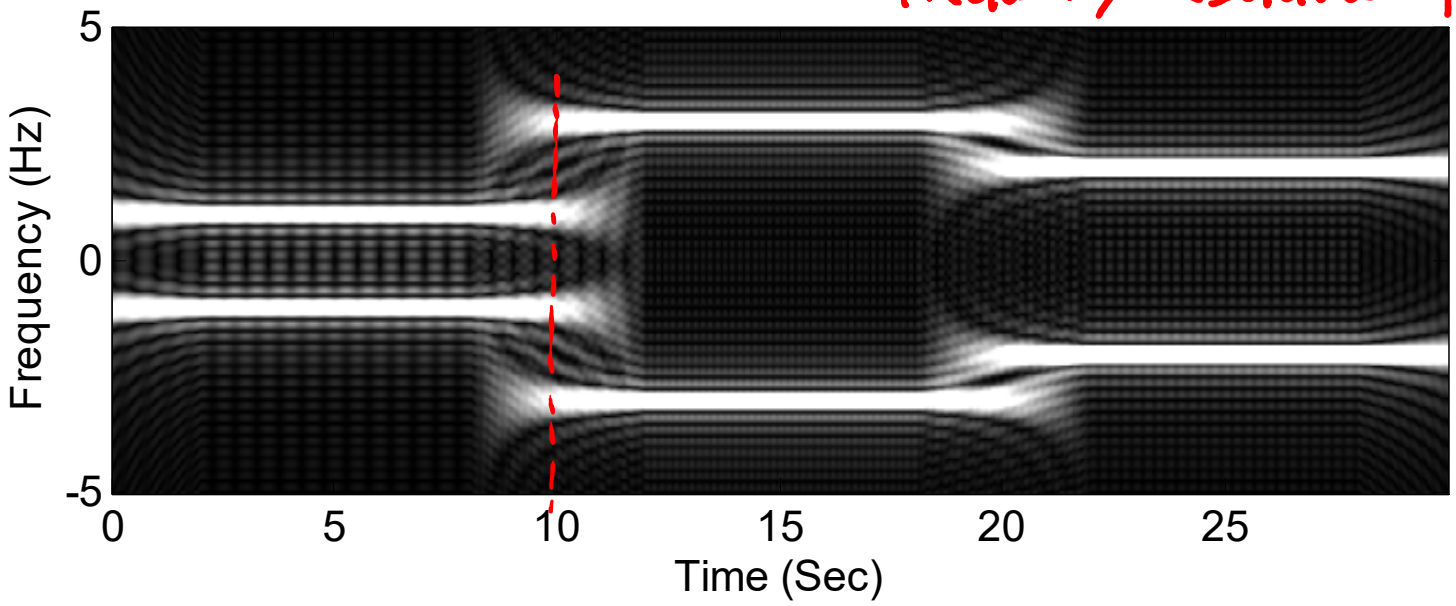
(page 30)

$\Pi(t) \xrightarrow{FT} \mathrm{sinc}(f)$

time resolution ↑
frequency resolution ↓



time resolution ↓
frequency resolution ↑



II-D Advantage and Disadvantage

- Compared with the Fourier transform:

All the time-frequency analysis methods has the advantage of:

The instantaneous frequency can be observed.

All the time-frequency analysis methods has the disadvantage of:

Higher complexity for computation

- Compared with other types of time-frequency analysis:

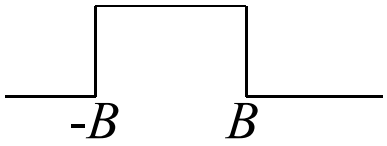
The rec-STFT has an advantage of the least computation time for digital implementation

but its performance is worse than other types of time-frequency analysis.

II-E STFT with Other Windows

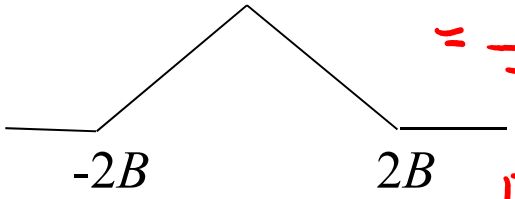
simplest

(1) Rectangle



$\xrightarrow{FT} 2B\,\mathrm{sinc}(2Bf)$

(2) Triangle

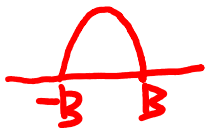


$= \sqcap_{-B\ B} * \sqcap_{-B\ B} \xrightarrow{FT} 4B^2\,\mathrm{sinc}^2(2Bf)$

(3) Hanning

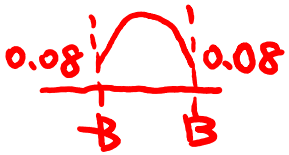
$\sqcap * \sqcap * \sqcap \xrightarrow{FT} 8B^3\,\mathrm{sinc}^3(2Bf)$

$$w(t) = \begin{cases} 0.5 + 0.5\cos(\pi t / B) & \text{when } |t| \le B \\ 0 & \text{otherwise} \end{cases}$$



(4) Hamming

$$w(t) = \begin{cases} 0.54 + 0.46\cos(\pi t / B) & \text{when } |t| \le B \\ 0 & \text{otherwise} \end{cases}$$



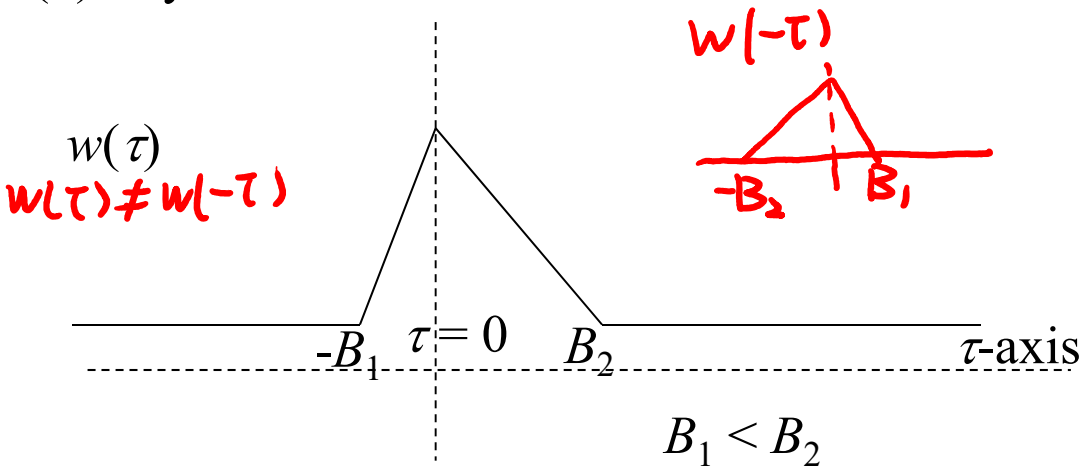
(5) Gaussian $\sqcap * \sqcap * \sqcap * \cdots\cdots$ (infinite times of convolution)

best result

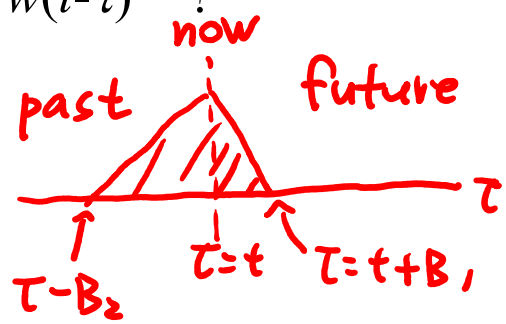
$$w(t) = \exp(-\pi\sigma t^2)$$

no sidelobe effect

(6) Asymmetric window



$w(t-\tau) = ?$



應用： seismic wave analysis, collision detection

(The applications that require real-time processing)

動腦思考：

(1) Are there other ways to choose the mask of the STFT?

(2) Which mask is better?

沒有一定的答案

II-F Spectrogram

STFT 的絕對值平方，被稱作 Spectrogram

$$SP_x(t,f)=\left|X(t,f)\right|^2=\left|\int_{-\infty}^{\infty} w(t-\tau)e^{-j2\pi f\tau}x(\tau)d\tau\right|^2$$

比較： spectrum 為 Fourier transform 的絕對值平方

$|X(f)|^2$

文獻上，spectrogram 這個名詞出現的頻率多於 STFT

但實際上， spectrogram 和 STFT 的本質是相同的

附錄三：使用 Matlab 將時頻分析結果 Show 出來

可採行兩種方式：

(1) 使用 mesh 指令畫出立體圖

(但結果不一定清楚，且執行時間較久)

(2) 將 amplitude 變為 gray-level，用顯示灰階圖的方法將結果表現出來

假設 y 是時頻分析計算的結果

```
image(abs(y)/max(max(abs(y)))*C)   % C 是一個常數，我習慣選 C=400
或 image(t, f, abs(y)/max(max(abs(y)))*C)
colormap(gray(256))          % 變成 gray-level 的圖
set(gca,'Ydir','normal')     % 若沒這一行, y-axis 的方向是倒過來的
```

```
set(gca,‘Fontsize’,12)        % 改變橫縱軸數值的 font sizes
xlabel('Time (Sec)','Fontsize',12)          % x-axis
ylabel('Frequency (Hz)','Fontsize',12)      % y-axis
title(‘STFT of x(t)','Fontsize',12)         % title
```

計算程式執行時間的指令：

tic　(這指令如同按下碼錶)

toc　(show 出碼錶按下後已經執行了多少時間)

註：通常程式執行第一次時，由於要做程式的編譯，所得出的執行時間會比較長

程式執行第二次以後所得出的執行時間，是較為正確的結果

附錄四：使用 Python 將時頻分析的圖畫出來

事前安裝模組

```
pip install numpy
pip install matplotlib
```

假設y為時頻分析結果(應為二維的矩陣數列)，將 y 以灰階方式畫出來

```
import numpy as np
import matplotlib.pyplot as plt
C = 400
y = np.abs(y) / np.max(np.abs(y)) * C
plt.imshow(y, cmap='gray', origin='lower')
# 加上 origin='lower' 避免上下相反
plt.xlabel('Time (Sec)')
plt.ylabel('Frequency (Hz)')
plt.show()
```

感謝2021年擔任助教的蔡昌廷同學

若要加上座標軸數值(在plt.show()之前加上以下程式碼)

```
x_label = ['0', '10', '20', '30']   # 橫軸座標值
y_label = ['-5', '0', '5']          # 縱軸座標值
plt.xticks(np.arange(0, x_max, step=int(x_max/(len(x_label)-1)), x_label)
plt.yticks(np.arange(0, y_max, step=int(y_max/(len(y_label)-1)), y_label)
```

Reference :
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xticks.html