

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！

- 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)，每回答一次(無論答對否)，只要不離題，都加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

方式有四種，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

(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，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

1分
4

(3) 口頭報告

限四個名額，每個人 40分鐘，題目可選擇和課程有關的任何一個主題。口頭報告將於最後一週 (1月13日)進行。有意願的同學，請儘早告知，以先登記的同學為優先。

口頭報告時，鼓勵大家提問（包括口頭報告的同學，也可針對其他同學的報告內容提問）。曾經提問的同學，期末報告皆加 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，請於明年1月13日(本學期最後一次上課)前，向我登記並告知我要編輯的條目(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輯的條目

編輯完成之後，要將連結寄給老師

書面報告、Tutorial、和編輯 Wikipedia，期限是 1月20日

以上若有做實驗模擬，請附上程式碼，會有額外的加分 (鼓勵不強制)

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

3n 出作業
3n+2 交作業

上課時間：15 週

9/23,

11/18, 出 HW3

9/30,

11/25,

10/7, 出 HW1

12/2, 交 HW3

10/14,

12/9, 出 HW4

10/21, 交 HW1

12/16,

10/28, 出 HW2

12/23, 交 HW4,

11/4,

12/30, 出 HW5

11/11, 交 HW2

1/13, Oral

1/20, 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

(續) → Time - Frequency Analysis

課程大綱：

黃鶴院士

- (11) Hilbert Huang Transform
- (12) From Haar Transforms to Wavelet Transforms
- (13) Continuous Wavelet Transforms
- (14) Continuous Wavelet Transforms with Discrete Coefficients
- (15) Discrete Wavelet Transform
- (16) 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) Cochleagram (for Acoustic Feature Extraction)
- ✓ (2) Seismic Wave Analysis Using Time-frequency Analysis
- ✓ (3) Time-Frequency Analysis for Tremor Analysis in Parkinson's Disease
- ✓ (4) Time-frequency Analysis of Musical Instruments
- ✓ (5) Learning-Based Speech Analysis in the Time-Frequency Domain
- (6) Learning-Based Bearing Fault Diagnosis in the Time-Frequency Domain
- ✓ (7) Computational Perception Auditory Structure
- ✓ (8) Dual Tree Complex Wavelet Transforms
- ✓ (9) Wavelet Pooling in Convolutional Neural Networks
- ✓ (10) Contourlet Convolutional Neural Networks
- ✓ (11) Fresnelet
- (12) Prediction Models Using Wavelet Transforms

I. Introduction

Fourier transform (FT)

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

↑ 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.

$$\cdot \cos(440\pi t) = \frac{1}{2}(e^{j440\pi t} + e^{-j440\pi t})$$

13

Example 1: $x(t) = \cos(440\pi t)$ when $t < 0.5$,

$x(t) = \cos(660\pi t)$ when $0.5 \leq t < 1$,

$x(t) = \cos(524\pi t)$ when $t \geq 1$



220Hz 低 La

330Hz Mi

262Hz Do

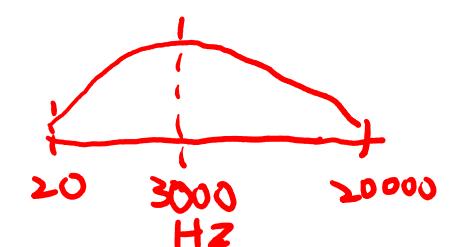
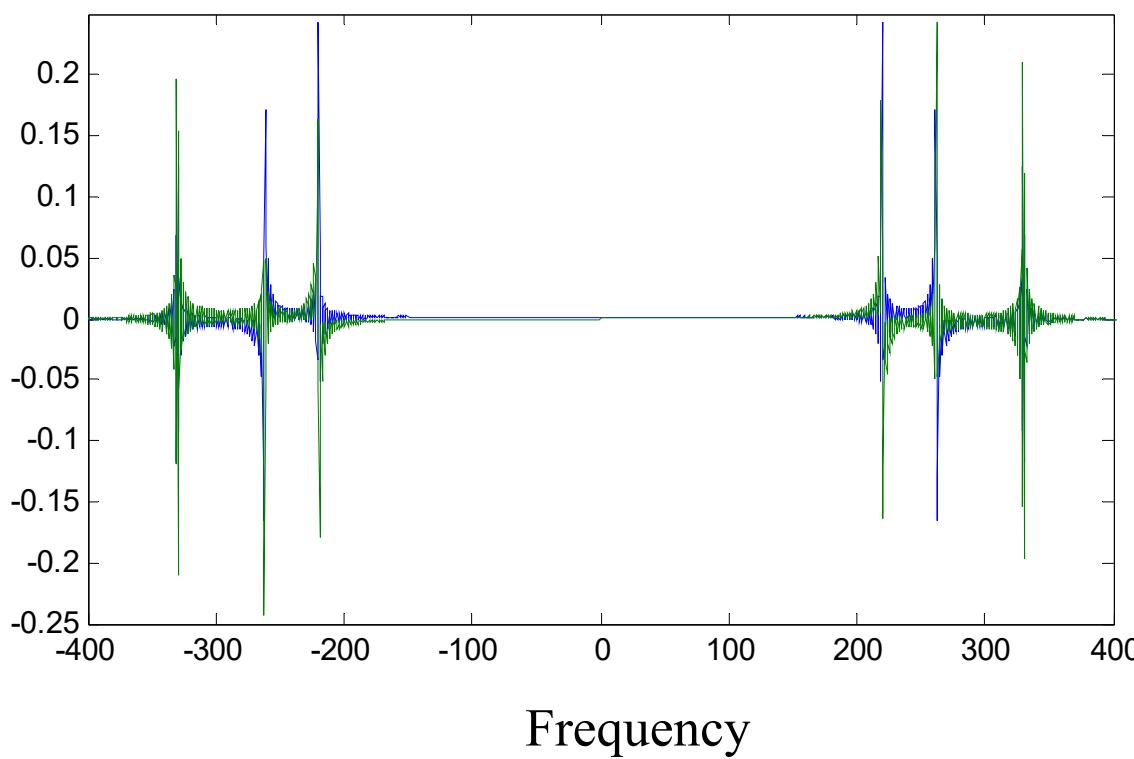
Do 261.63 Hz

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

低 La $261.63 \times 2^{\frac{-3}{12}}$

The Fourier transform of $x(t)$

20~20000 Hz



(A) Finite-Supporting Fourier Transform

$$(-\infty, \infty) \Rightarrow (t_0-B, t_0+B)$$

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

(B) Short-Time Fourier Transform (STFT)

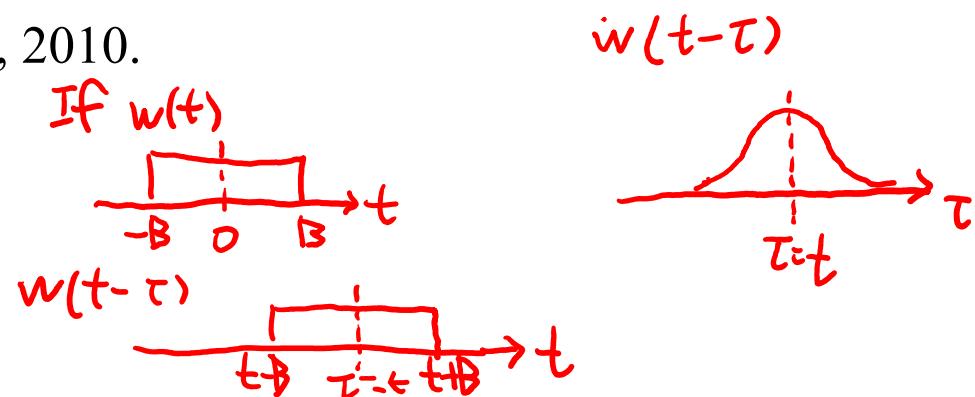
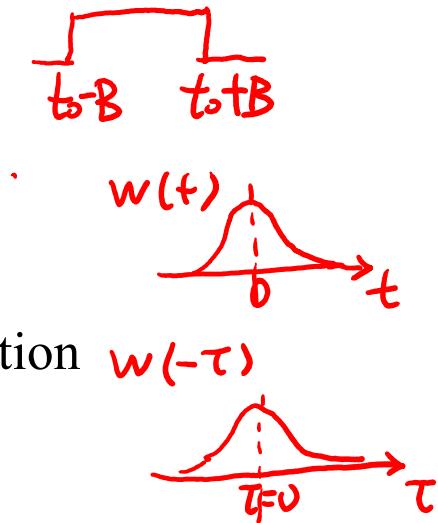
$$\underline{X(t, f)} = \int_{-\infty}^{\infty} w(t-\tau) x(\tau) e^{-j2\pi f \tau} d\tau$$

$w(t)$: 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.

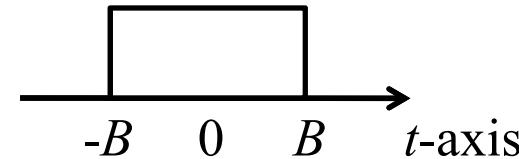
[Ref] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*,
London: Prentice-Hall, 3rd ed., 2010.



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

最簡單的例子： $w(t) = 1$ for $|t| \leq B$,

$$w(t) = 0 \quad \text{otherwise}$$

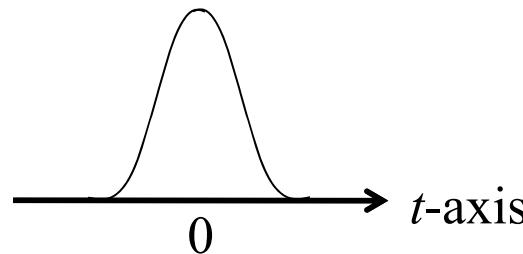


此時 Short-time Fourier transform 可以改寫

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

其他的例子：

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



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

此時的 Short-Time Fourier Transform 亦稱作 Gabor Transform

(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 $0.5 \leq t < 1$, 330Hz

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

$$B=330$$

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

of sampling points
 $= 660 \times 1.5 = \underline{\underline{990}}$

modified

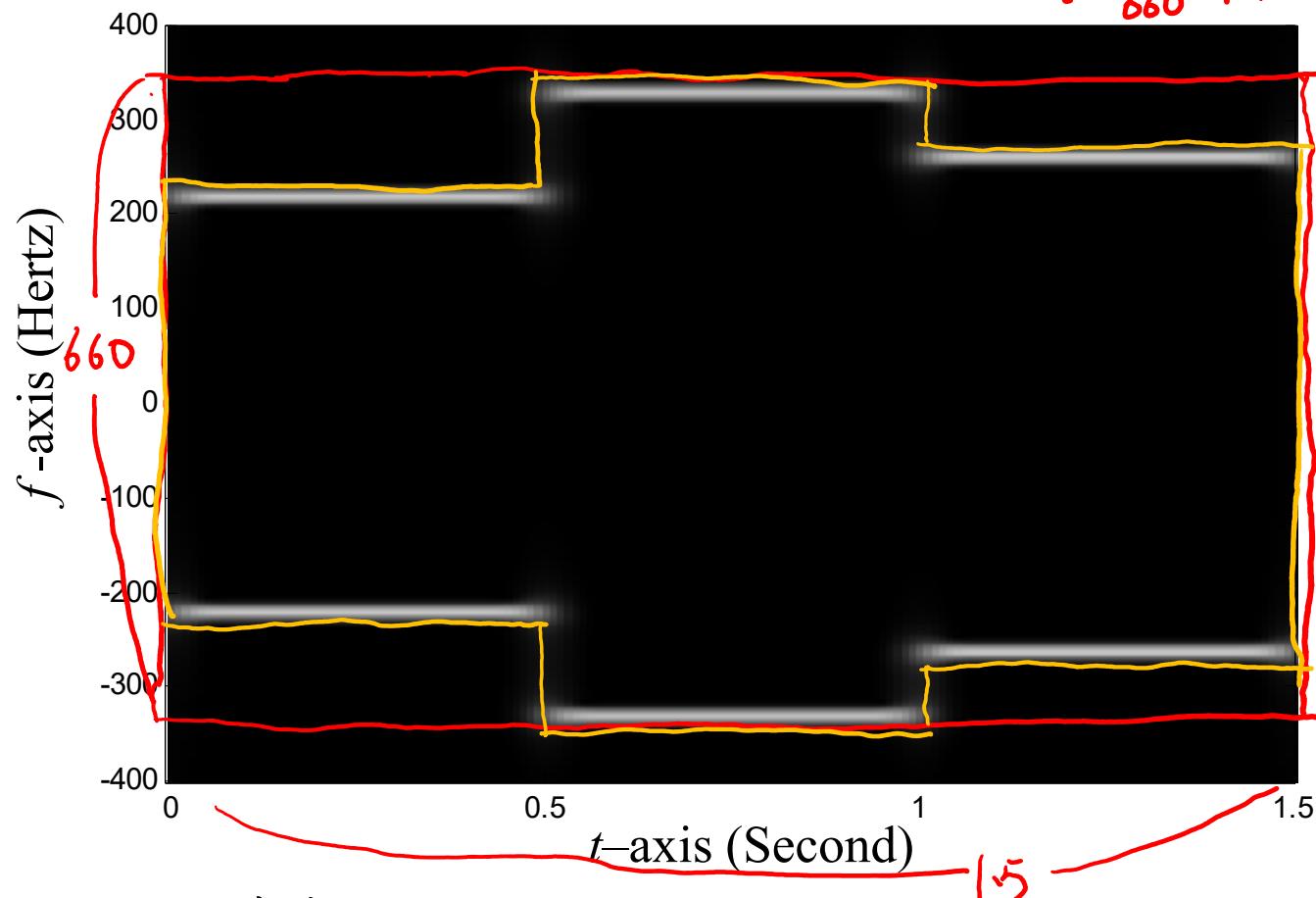
$$\Delta t < \frac{1}{440} \text{ for } 0 < t < 0.5$$

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

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

of sampling points

$$440 \times 0.5 + \\ 660 \times 0.5 + \\ 524 \times 0.5 \\ = \underline{\underline{812}}$$



用 Gray level 来表示 $X(t, f)$ 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

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

$$\phi_1'(t_0), \phi_2'(t_0), \phi_3'(t_0), \dots, \phi_N'(t_0) \quad (\text{以角頻率 angular frequency 表示}) :$$

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

$$\text{ex: if } \phi_1(t) = 440\pi t$$

$$\phi_1'(t) = 440\pi$$

$$\frac{\phi_1'(t)}{2\pi} = 220$$

自然界中，頻率會隨著時間而改變的例子

Frequency Modulation

Music

Speech

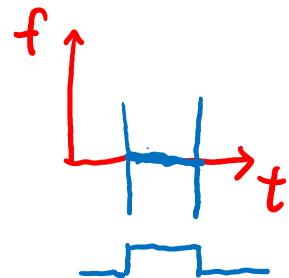
地震波

光學

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

$$\text{矩形波} \rightarrow \text{sinc}(f) = \frac{\sin(\pi f)}{\pi f}$$

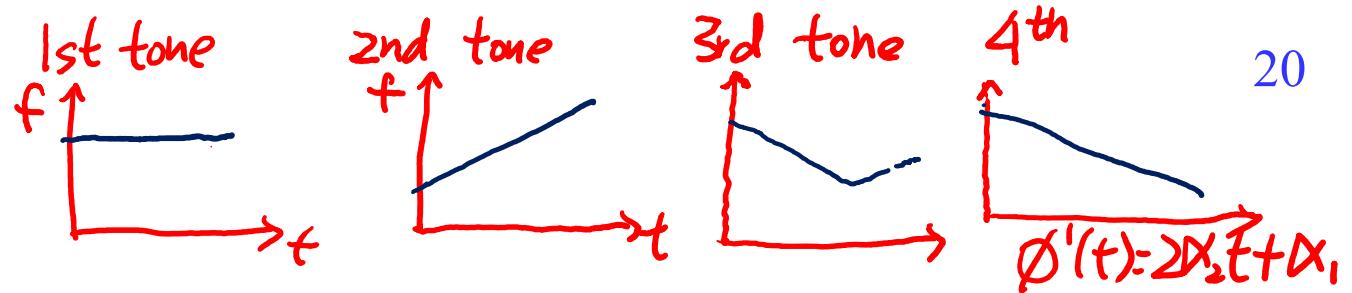
In fact, in addition to **sinusoid-like functions**, the instantaneous frequencies of other functions will inevitably vary with time.



$$\cos(2\pi f t + \phi) \quad \text{for } t \in (-\infty, \infty)$$

20

- Sinusoid Function



唱歌聲

- Chirp function *phase is a 2nd order polynomial*

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

acoustics, wireless communication, radar system, optics

例：Y ($F_1 = 900\text{Hz}$, $F_2 = 1200\text{Hz}$), — ($F_1 = 300\text{Hz}$, $F_2 = 2300\text{Hz}$)

F_1 由嘴唇的大小決定, $F_2 - F_1$ 由如面的高低決定

- Higher order exponential function

：

Example 2

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

$$\phi(t) = \pm(6400\pi t - 600\pi t^2)$$

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

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

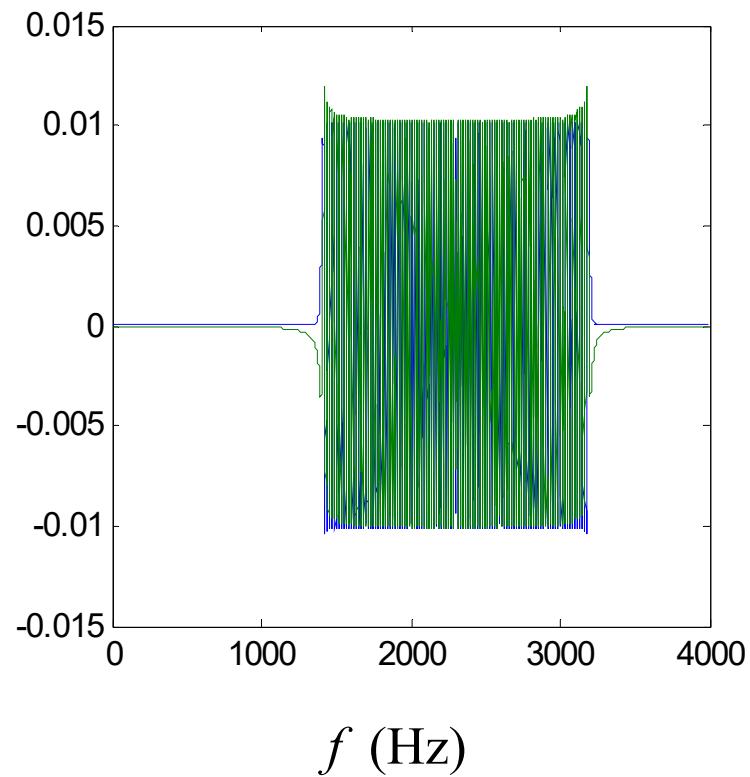
$$\phi(t) = \pm(600\pi t^3 - 2700\pi t^2 + 5050\pi t)$$

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

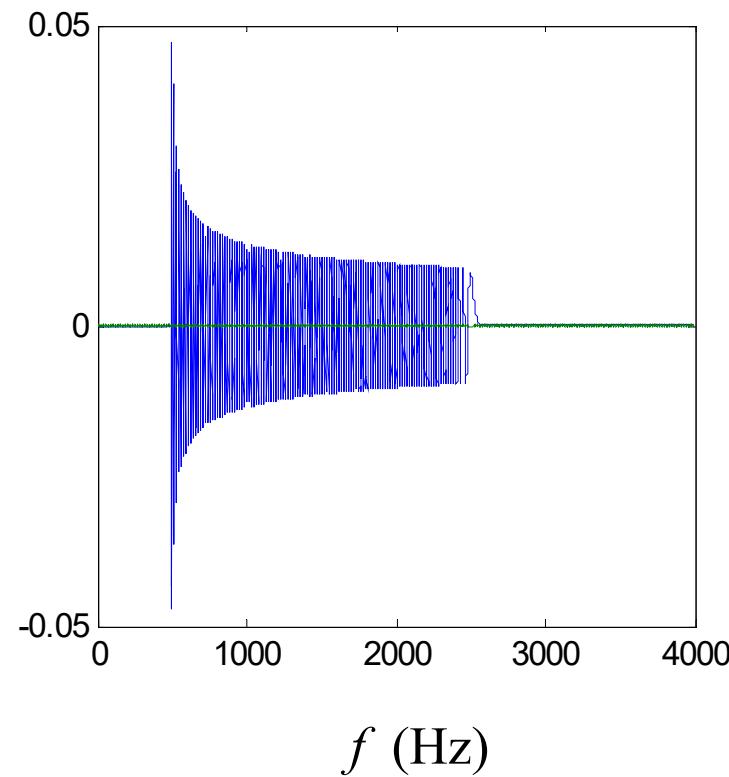
$$= \pm(900(t - \frac{3}{2})^2 + 500)$$

Fourier transform

$$y(t) = 0.5 \cos(6400\pi t - 600\pi t^2)$$

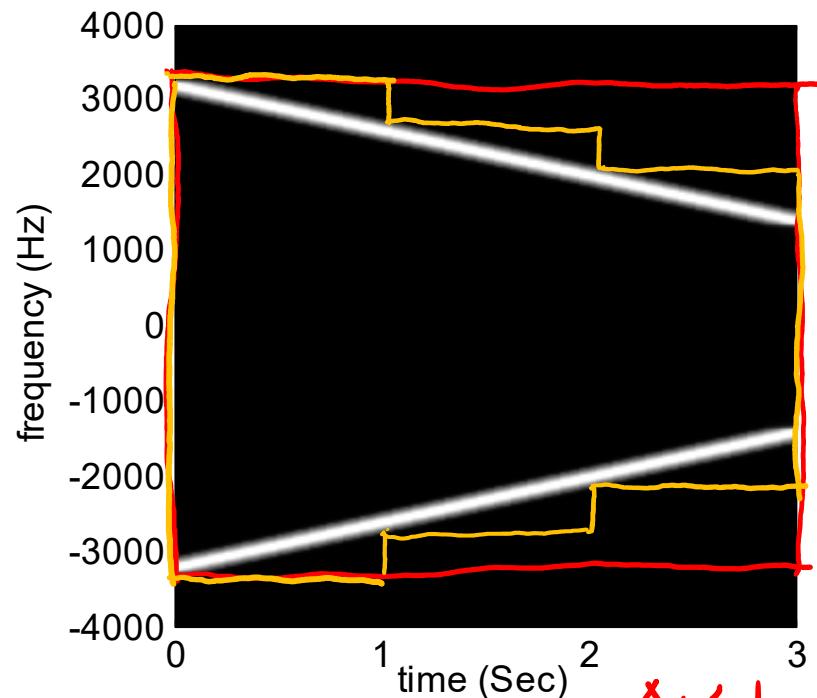


$$x(t) = 0.5 \cos(600\pi t^3 - 2700\pi t^2 + 5050\pi t)$$



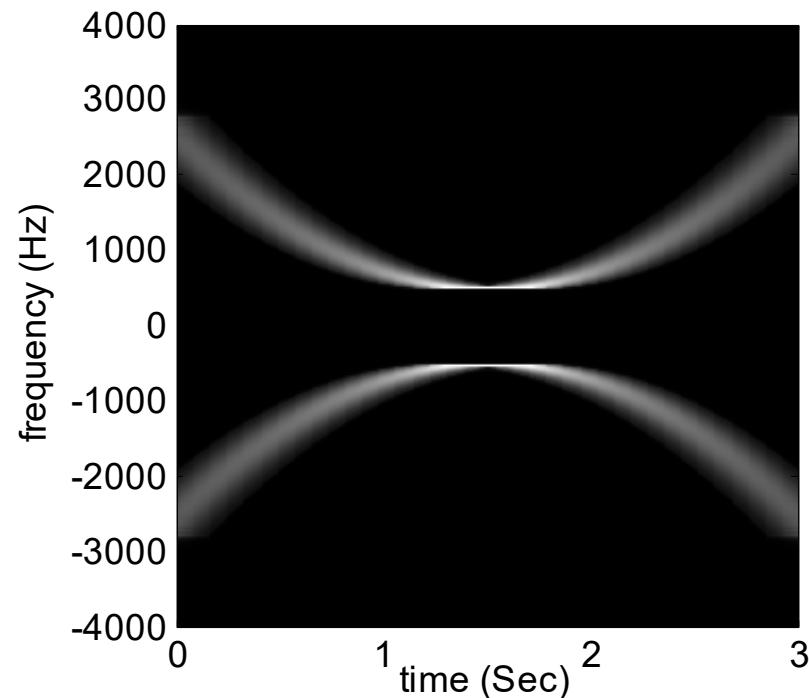
(1)

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



(2)

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



of sampling points: $6400 \times 3 = 19200$
modified
 $\Delta t < \frac{1}{2 \times 3200}$

$\Delta t < \frac{1}{6400}$ for $0 < t < 1$, $\Delta t < \frac{1}{2 \times 2600}$ for $1 < t < 2$, $\Delta t < \frac{1}{2 \times 2000}$ for $2 < t < 3$

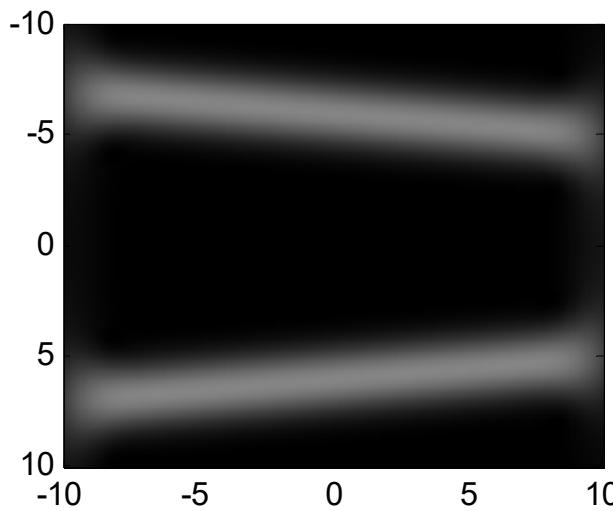
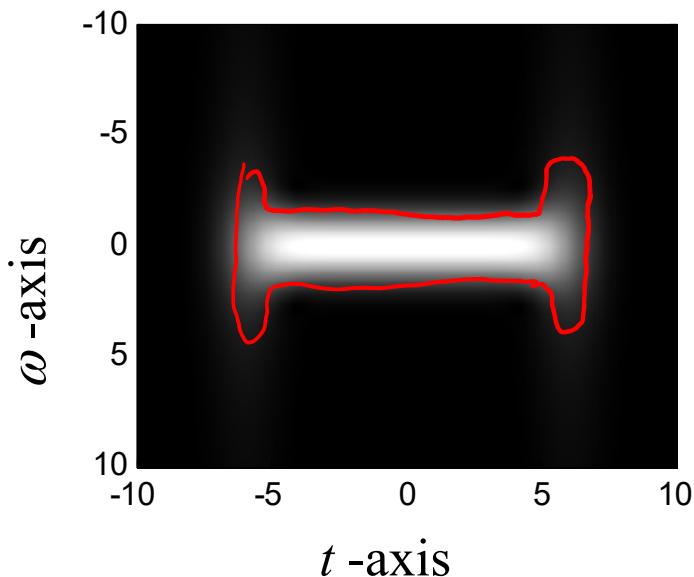
of sampling points: $6400 \times 1 + 5200 \times 1 + 4000 \times 1 = 15600$

Example 3



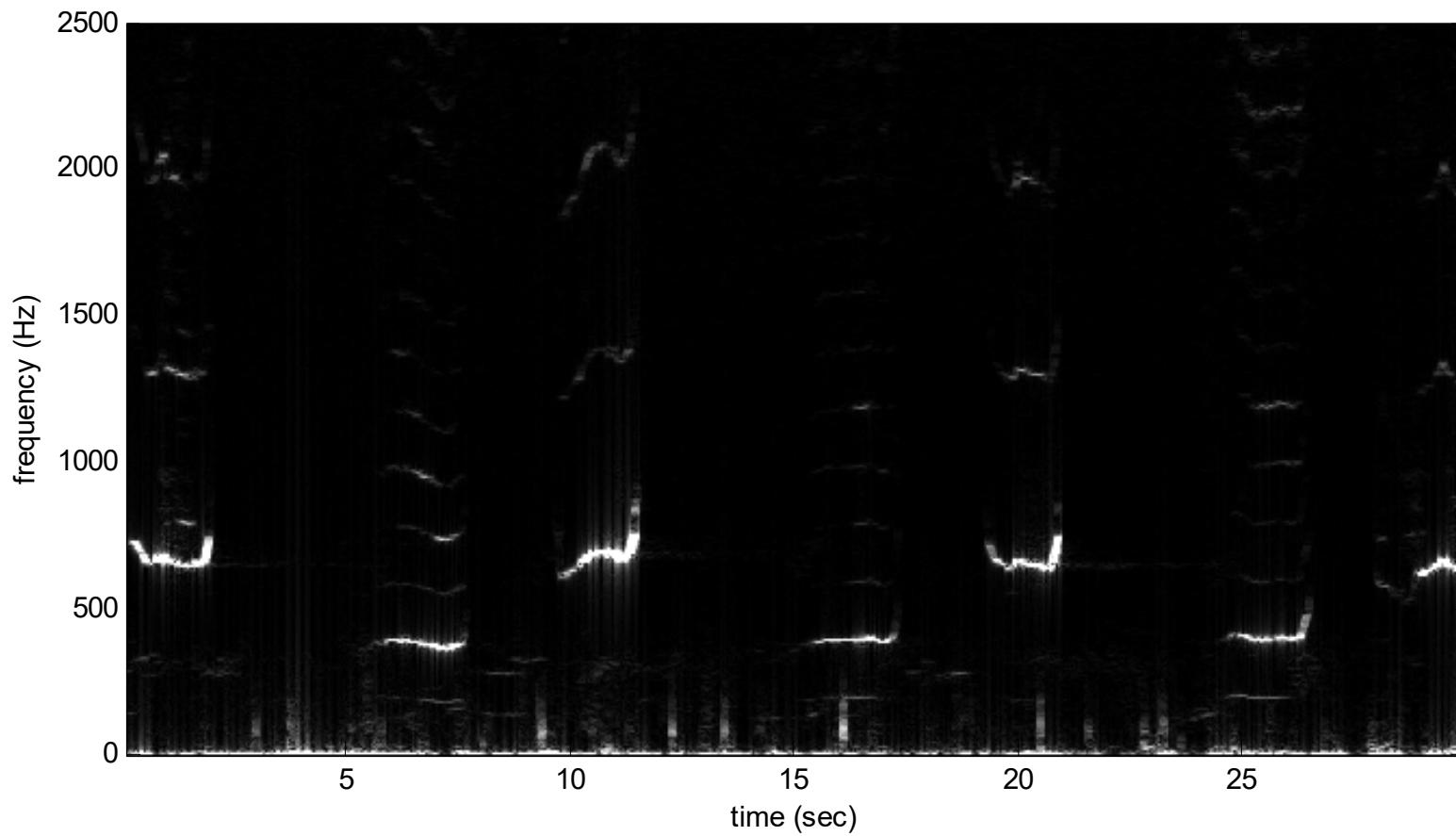
left: $x_1(t) = 1$ for $|t| \leq 6$, $x_1(t) = 0$ otherwise, right: $x_2(t) = \cos(6t - 0.05t^2)$

Gabor transform



Example 4

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



Why Time-Frequency Analysis is Important?

- Many digital signal processing applications are related to the spectrum or the bandwidth of a signal.
 - 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
- FT applications -*
- spectrum analysis
(replaced by time-frequency)
calculate convolution*
- cannot be replaced
by time-frequency*
- $\Delta t < \frac{1}{2B}$.

Example: Generalization for sampling theory

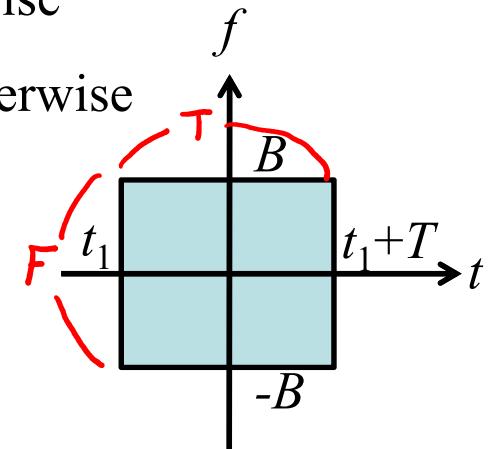
假設有一個信號，

- ① The supporting of $x(t)$ is $t_1 \leq t \leq t_1 + T$, $x(t) \approx 0$ otherwise
- ② The supporting of $X(f) \neq 0$ is $-B \leq f \leq B$, $X(f) \approx 0$ otherwise

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

所以，取樣點數 N 的範圍是

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



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

$$|G(f) e^{-j2\pi f t_0}| = |G(f)|$$

Q1 : Scaling 對於一個信號的取樣點數有沒有影響？

Hint: $(g(t), g(t-t_0)) \xrightarrow{\text{FT}} G(f) e^{-j2\pi f t_0}$

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

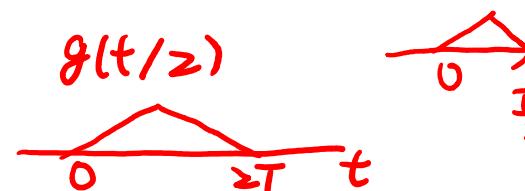
$\sigma > 1$: narrower

$\sigma < 1$: wider

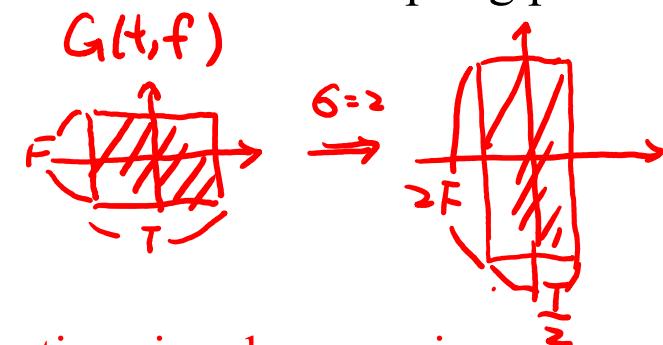
ex: $g(4t)$



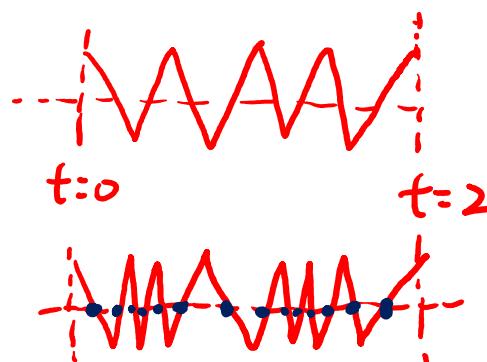
$g(2t)$



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.



$$\text{frequency} = \frac{\text{number of periods}}{\text{time duration}} = \frac{4}{2} = 2$$

2 zero crossing = 1 period

$$\text{frequency} = \frac{(\text{number of zero crossings})/2}{\text{time duration}} = 3.5$$

時頻分析大家族

page 14, 1946

(1) Short-time Fourier transform (STFT)

(rec-STFT, Gabor, ...)

page 16, 1932

(2) Wigner distribution function (WDF)

1981

(3) Wavelet transform

(4) Time-Variant Basis Expansion

1996

黃金等

(5) Hilbert-Huang Transform (唯一跳脫 Fourier transform 的架構)

square

spectrogram

generalized spectrogram

improve

Asymmetric STFT

combine

S transform

Gabor-Wigner Transform

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

Haar and Daubechies

Coiflet, Morlet

Directional Wavelet Transform

Matching Pursuit and 1993

Compressive Sensing 2006

Prolate Spheroidal Wave Function

- **Continuous Wavelet Transform**

ψ : ψsy /saɪ/

forward wavelet transform:

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

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

inverse wavelet transform:

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

$b \downarrow, f \uparrow$
 $b \uparrow, f \downarrow$

$\varphi_{a,b}(t)$ 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：

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

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

(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,

Laplacian of Gaussian (LoG)

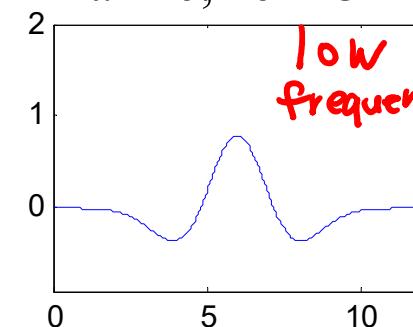
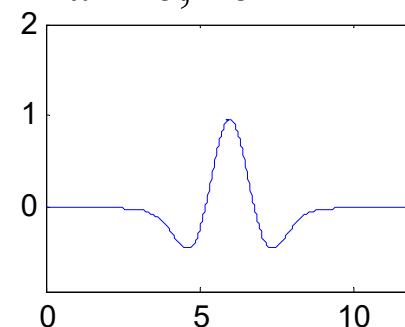
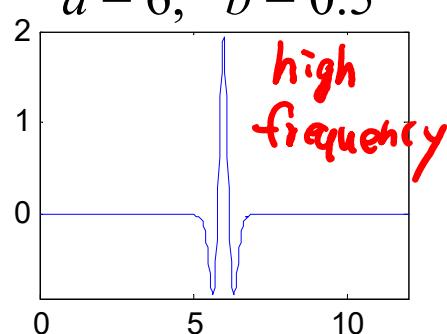
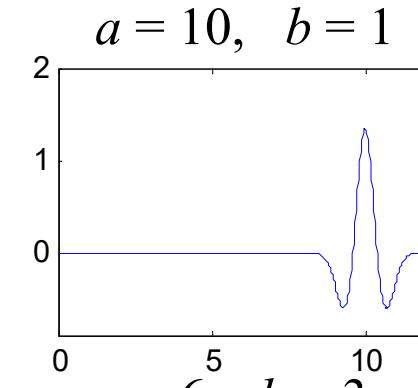
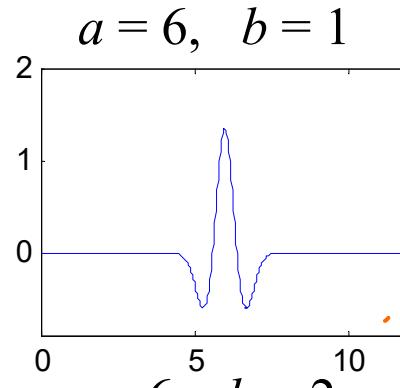
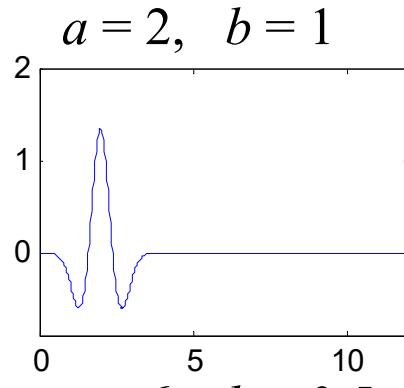
$$\frac{d^2}{dt^2} e^{-\pi t^2}$$

$$= (4\pi^2 t^2 - 2\pi) e^{-\pi t^2}$$

例子：Mexican hat wavelet 隨 a and b 變化之情形

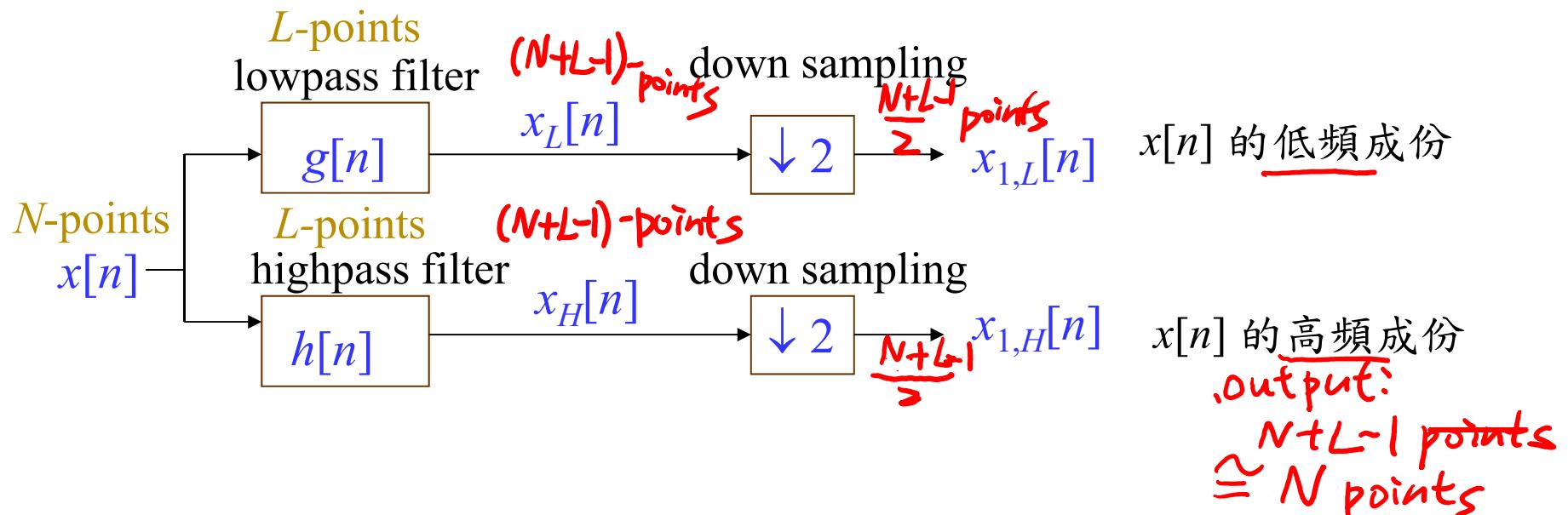
$$\psi(t) = \frac{2^{5/4}}{\sqrt{3}} (1 - 2\pi t^2) e^{-\pi t^2}$$

$$\frac{1}{\sqrt{b}} \psi\left(\frac{t-a}{b}\right)$$



• 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.



$$x_L[n] = \sum_k x[n-k]g[k]$$

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

$$x_H[n] = \sum_k x[n-k]h[k]$$

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

$$x_i[n] \rightarrow \boxed{\downarrow 2} \rightarrow x_o[n] \quad x_o[n] = x_i[2n]$$

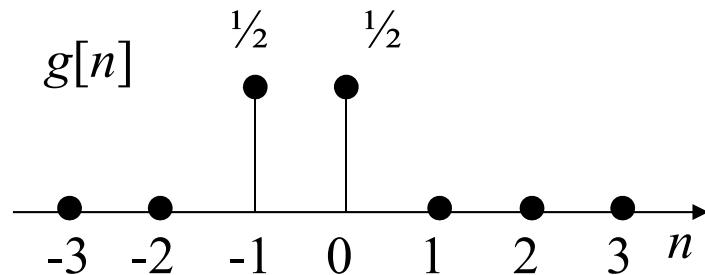
$$x_{1,L}[n] = \sum_k x[2n-k]g[k]$$

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

例子 : 2-point Haar wavelet

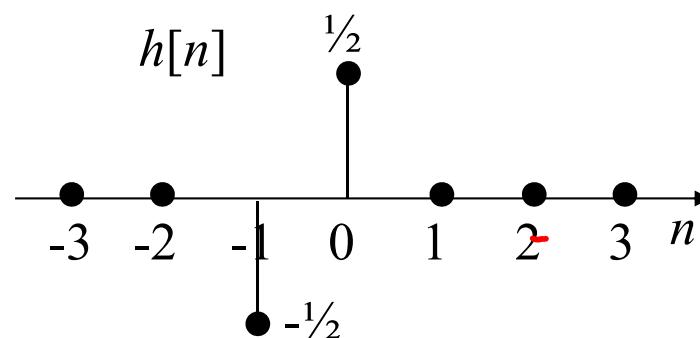
$$g[n] = 1/2 \text{ for } n = -1, 0$$

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



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

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



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)

Discrete wavelet transform 有很多種

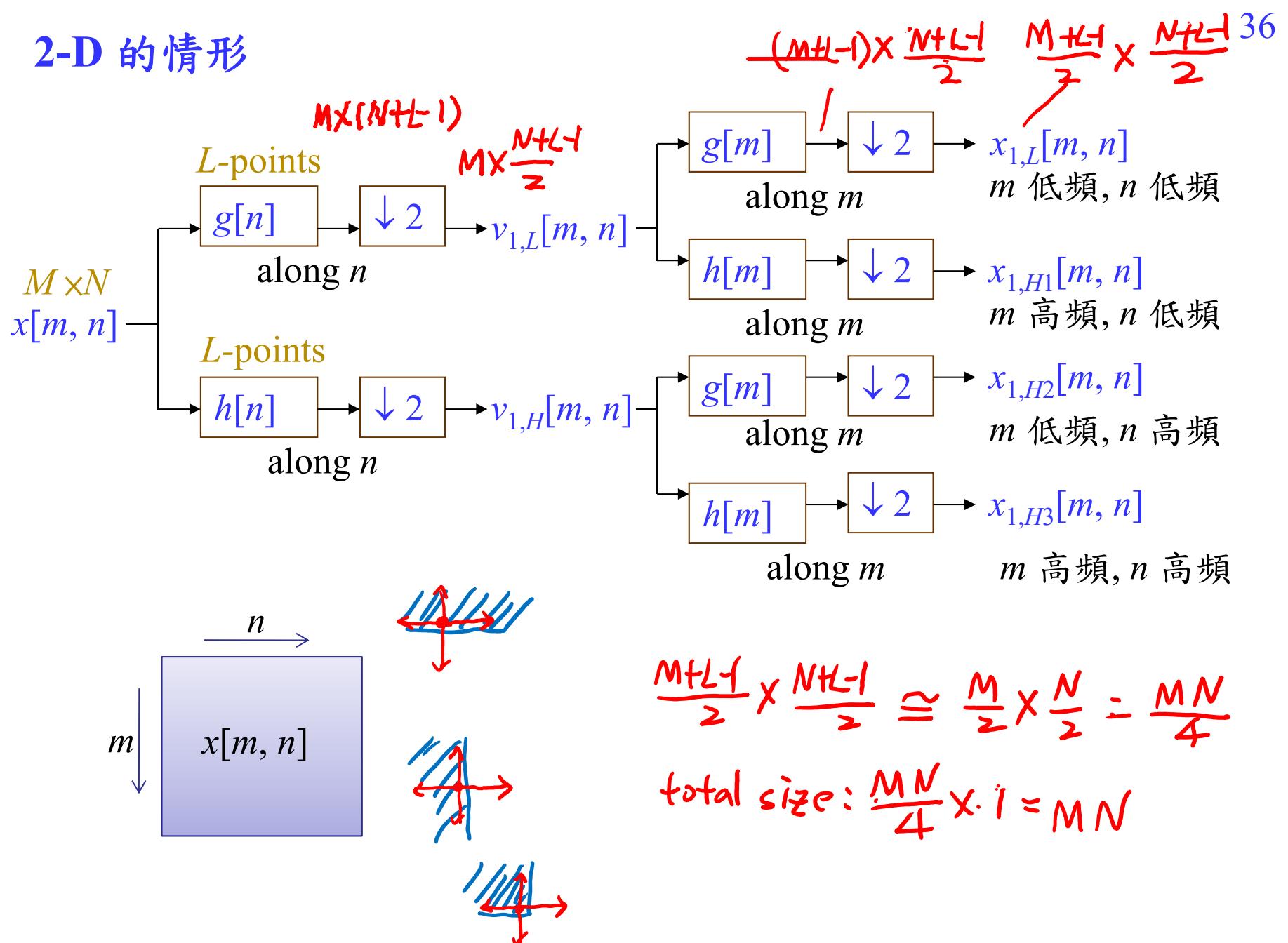
(discrete Haar wavelet, discrete Daubechies wavelet, B-spline DWT,
symlet, coilet,)

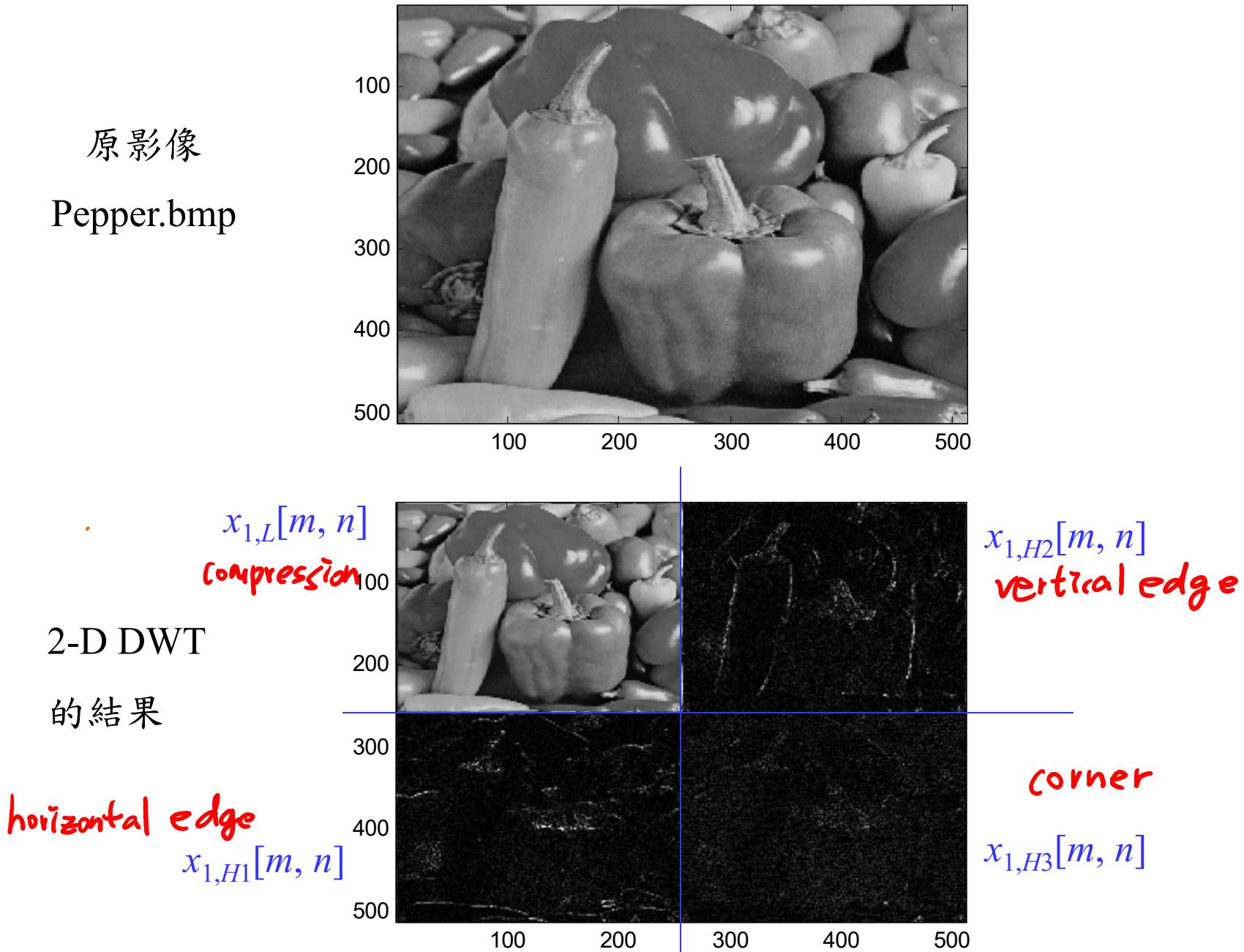
一般的 wavelet, $g[n]$ 和 $h[n]$ 點數會多於 2 點

但是 $g[n]$ 通常都是 lowpass filter 的型態

$h[n]$ 通常都是 highpass filter 的型態

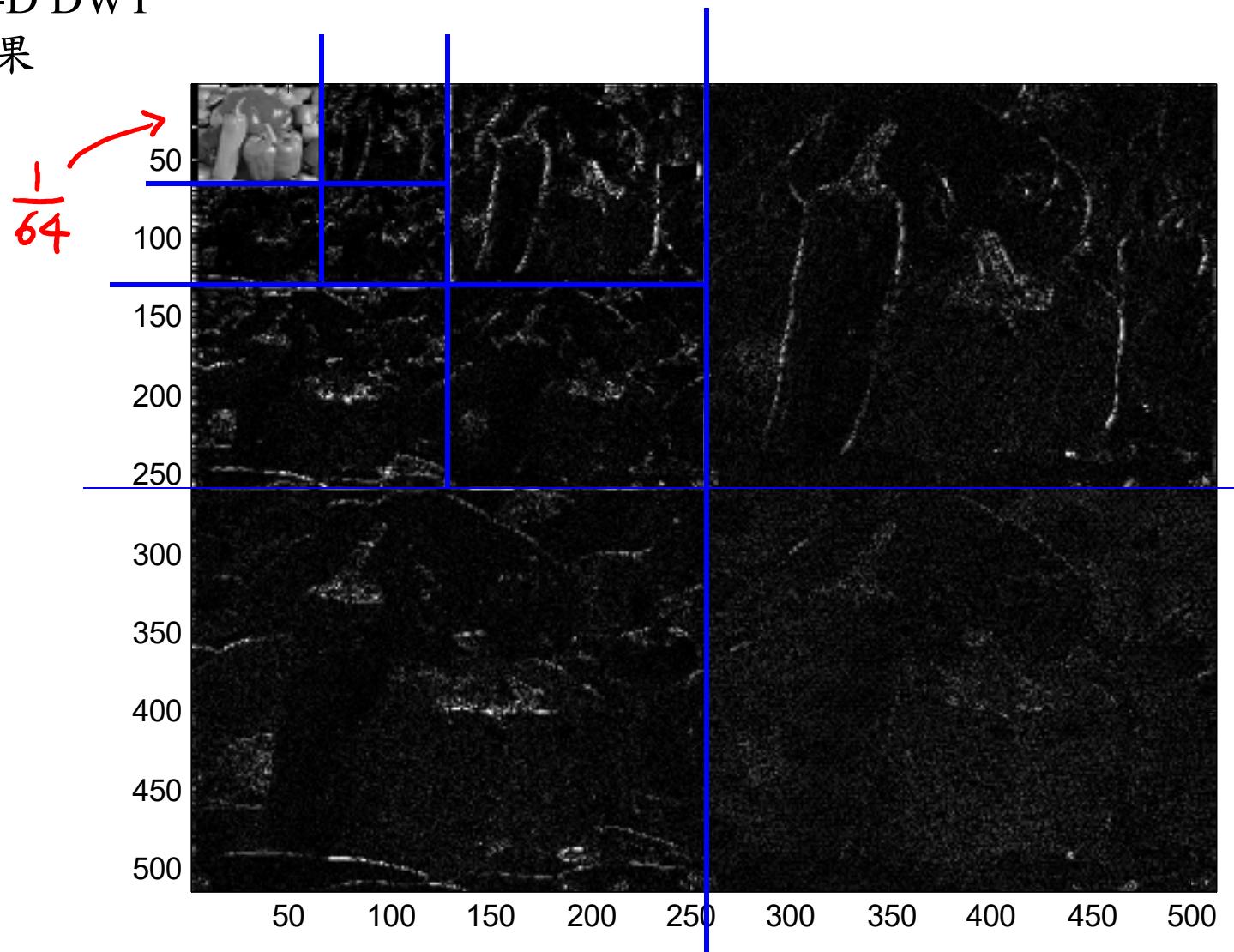
2-D 的情形





3次2-D DWT

的結果



比較：現有最流行壓縮技術 JPEG

應用：影像壓縮 (JPEG 2000) $\frac{1}{50}$ $\frac{1}{100}$

$\frac{1}{10}$ $\frac{1}{20}$

其他應用：edge detection

corner detection

filter design

pattern recognition

music signal processing

economical data

temperature analysis

feature extraction

biomedical signal processing

附錄一：聲音檔的處理

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
- 讀取：**audioread**
- 例：`[x, fs] = audioread('C:\WINDOWS\Media\ringin.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) = 9981 1` `fs = 11025`
- 思考：所以，取樣間隔多大？
- 這個聲音檔有多少秒？

一個聲音檔如果太大，我們也可以只讀取它部分的點

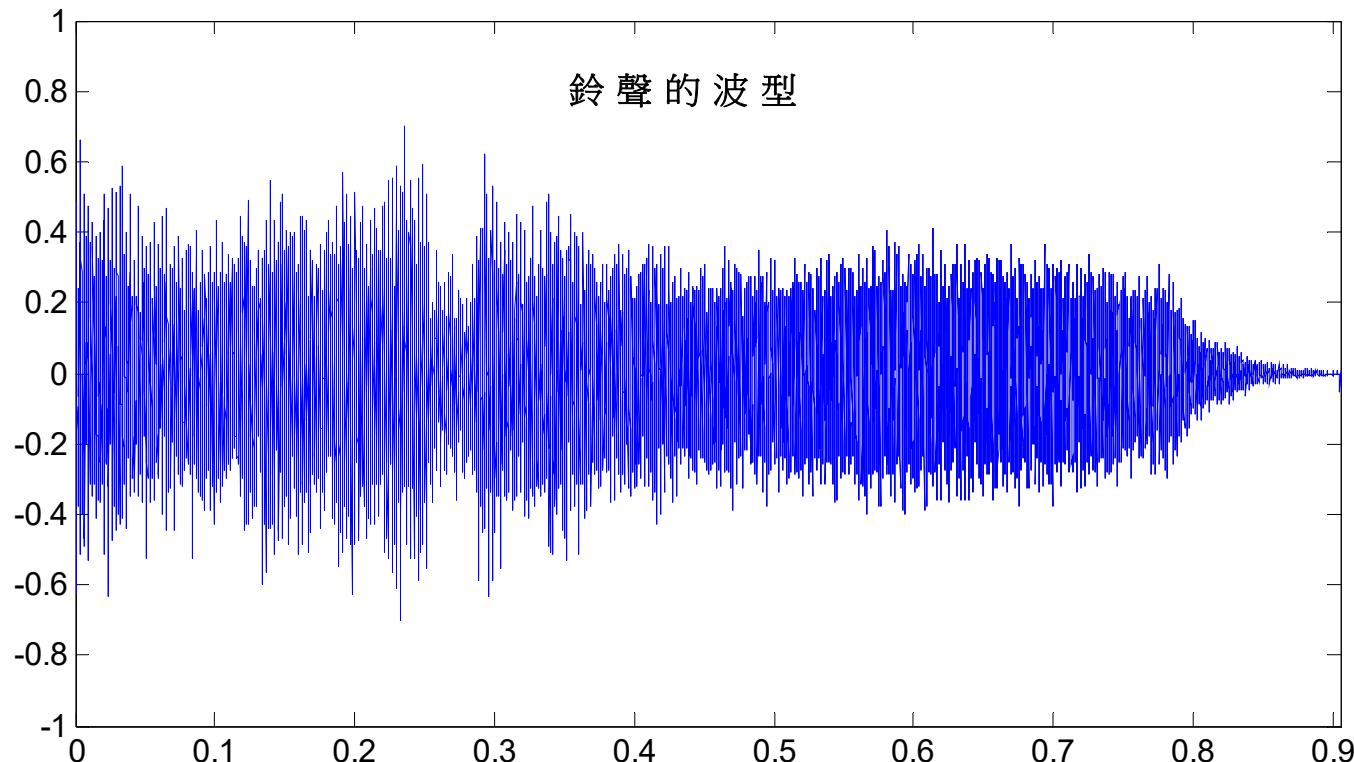
`[x, fs]=audioread('C:\WINDOWS\Media\ringin.wav', [4001 5000]);`

% 讀取第4001至5000點

畫出聲音的波型

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

plot(time, x)

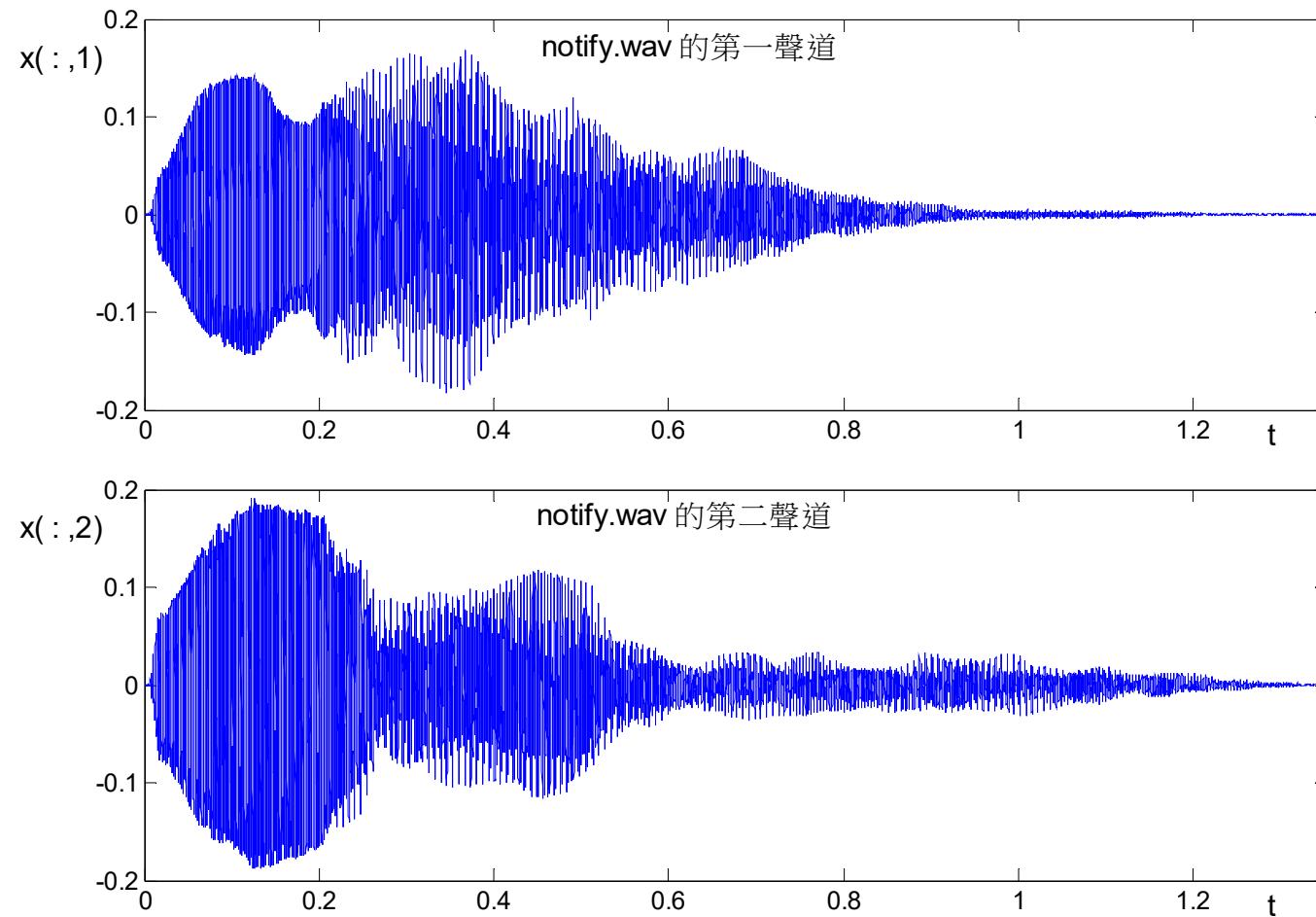


注意： *.wav 檔中所讀取的資料，值都在 -1 和 +1 之間

- 有些聲音檔是雙聲道（Stereo）的型態（俗稱立體聲）

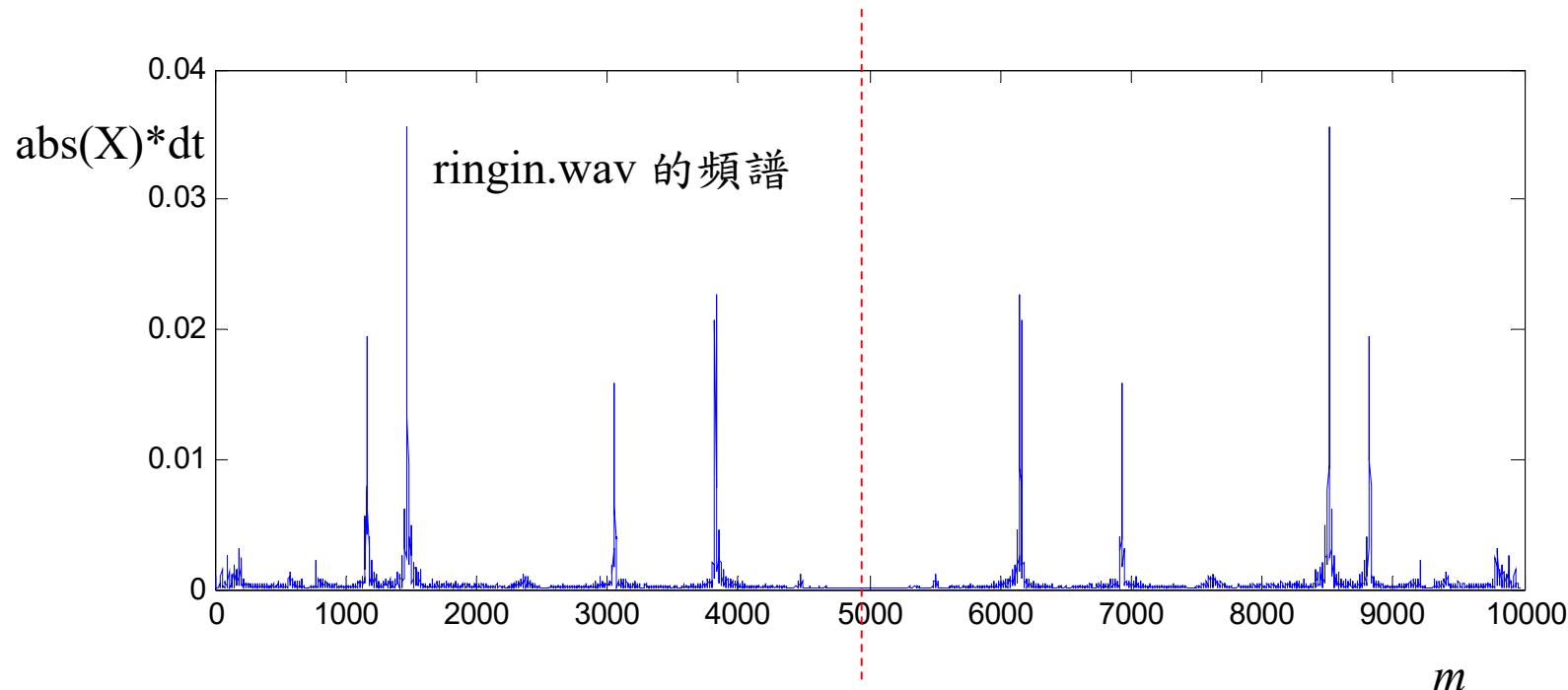
例：[x, fs]=audioread('C:\WINDOWS\Media\notify.wav');

$\text{size}(x) = 29823 \quad 2 \quad \text{fs} = 22050$



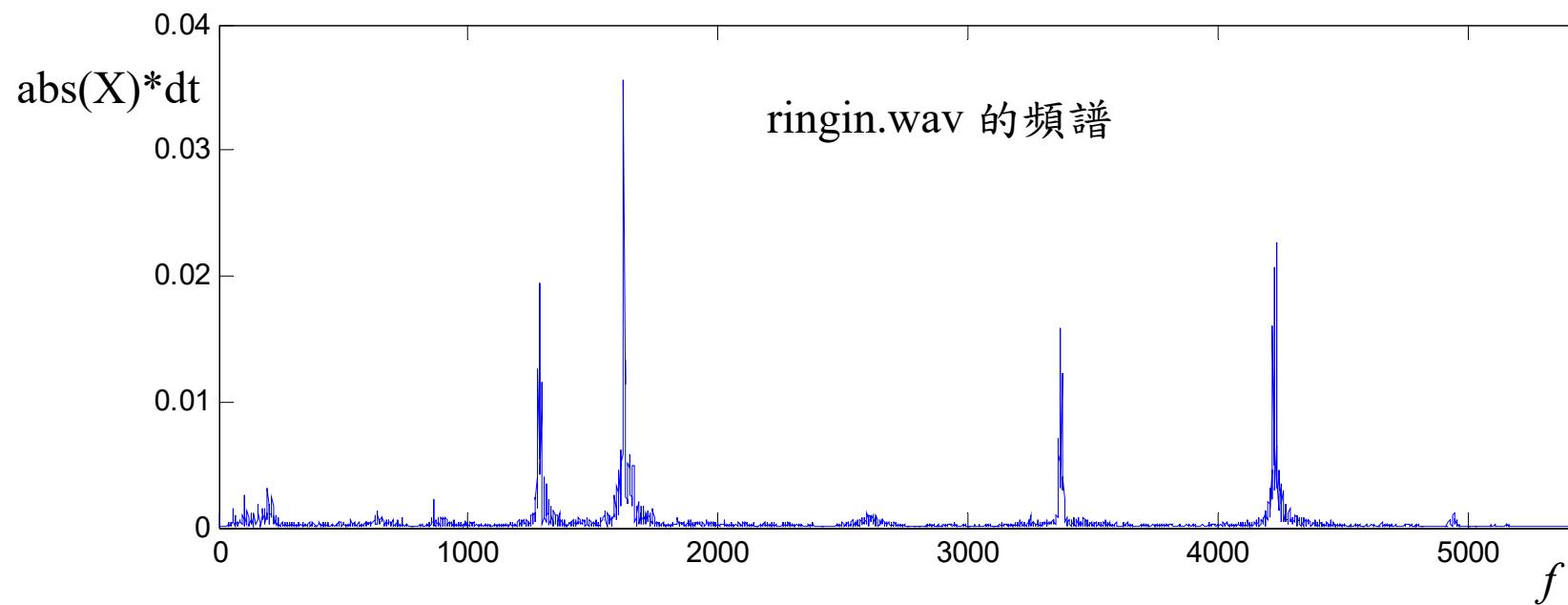
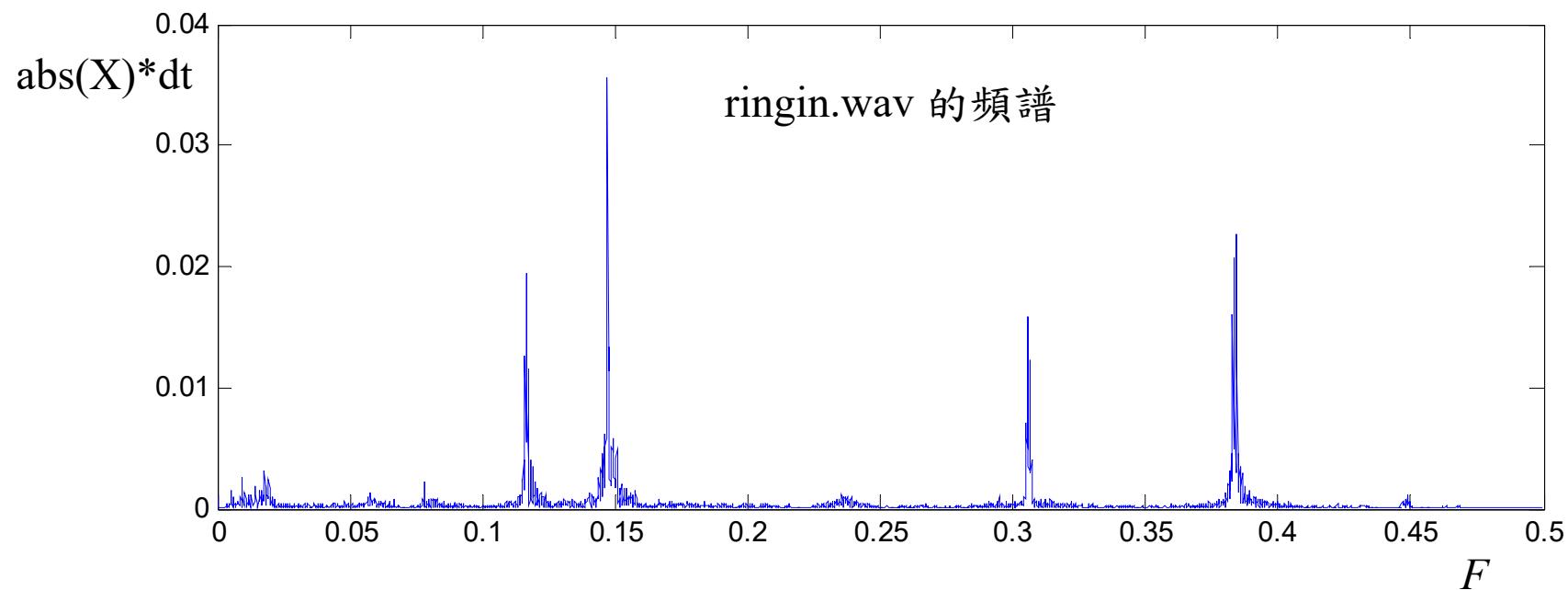
B. 繪出頻譜

X = fft(x); **plot(abs(X)*dt); % dt = 1/fs**



fft 橫軸 轉換的方法

- (1) Using normalized frequency F : $F = m / N$.
- (2) Using frequency f , $f = F \times f_s = m \times (f_s / N)$.



C. 聲音的播放

(1) audioplay(x): 將 x 以 11025Hz 的頻率播放
(時間間隔 = $1/11025 = 9.07 \times 10^{-5}$ 秒)

(2) sound(x): 將 x 以 8192Hz 的頻率播放

(3) sound(x, fs) 或 sound(x, fs) 或 audioplay(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

(4) 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(x, fs, waveFile)`

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 之間
- ③ 若沒有設定 **fs**，則預設的 **fs** 為 8000Hz

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

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

$$X(t, \omega) = \int_{-\infty}^{\infty} w(t - \tau) x(\tau) e^{-j\omega \tau} d\tau \quad \omega = 2\pi f$$

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.

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

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

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

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

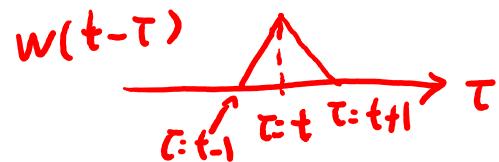
where $w(t_1 - t) \neq 0$.

For the alternative definition, the inverse transform is:

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

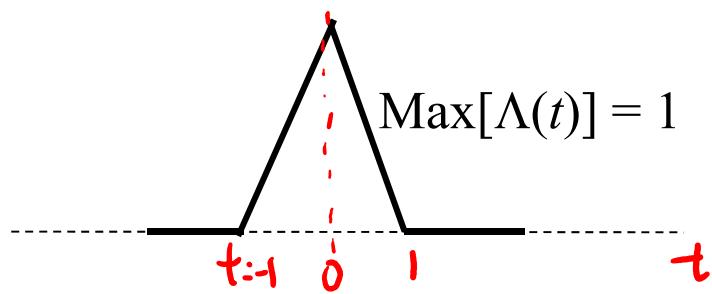
The mask function $w(t)$ always has the property of

- (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



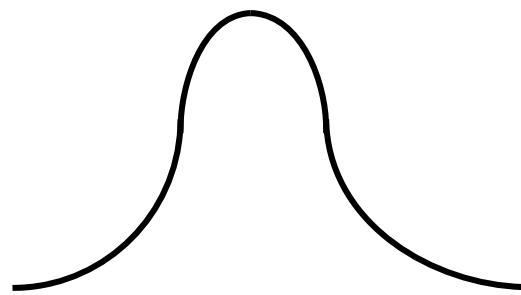
$w(t) = \Lambda(t)$ (triangular function)

$$t = -1 \quad t = 1$$



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

(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

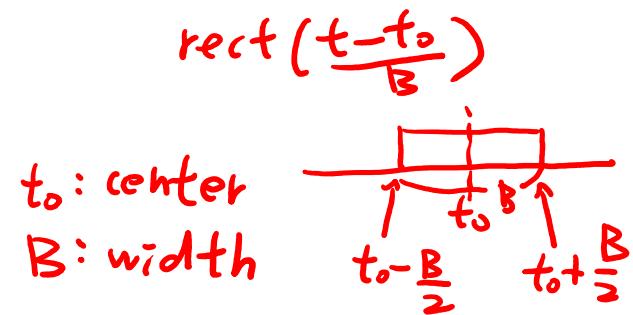
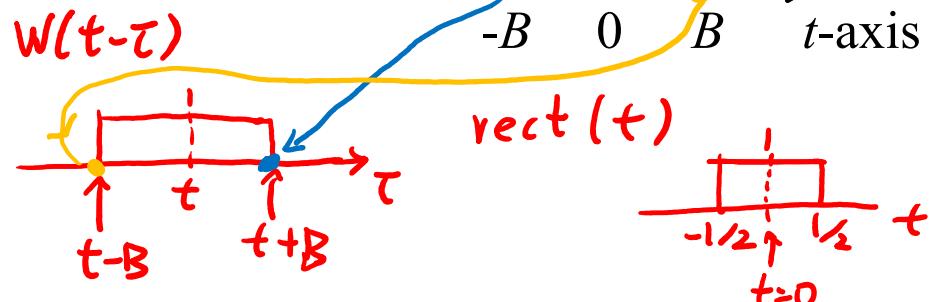
$$x(t) = \int_{-\infty}^{\infty} X(t_1, f) e^{j2\pi f t} df$$

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

$$\frac{1}{w(t_1 - \tau)} \quad \text{when } |t_1 - \tau| < B$$

The simplest form of the STFT

$$w(t) = \text{rect}\left(\frac{t}{2B}\right)$$



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

II-C Properties of the Rec-STFT

(1) Integration (recovery):

$$\begin{aligned}
 (a) \quad \int_{-\infty}^{\infty} X(t, f) df &= \int_{t-B}^{t+B} x(\tau) \int_{-\infty}^{\infty} e^{-j2\pi f \tau} df d\tau \\
 &= \int_{t-B}^{t+B} x(\tau) \delta(\tau) d\tau \\
 &= \begin{cases} x(0) & \text{when } t - B < 0 < t + B, \quad -B < t < B \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

$$\begin{aligned}
 (b) \quad \int_{-\infty}^{\infty} X(t, f) e^{j2\pi f v} df &= x(v) \quad \text{when } v - B < t < v + B, \\
 &= 0 \quad \text{otherwise}
 \end{aligned}$$

tt 較

$$\mathcal{F}(x(t+t_0)) = e^{j2\pi f t_0} X(f)$$

$$\mathcal{F}(x(t) e^{j2\pi f_0 t}) = X(f - f_0)$$

(2) Shifting property (橫的方向移動)

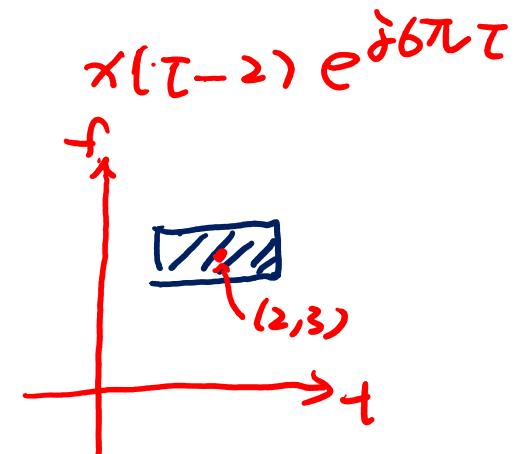
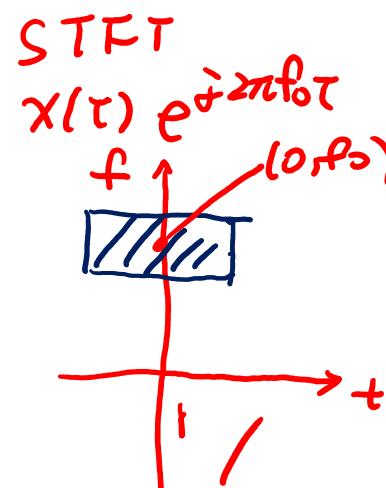
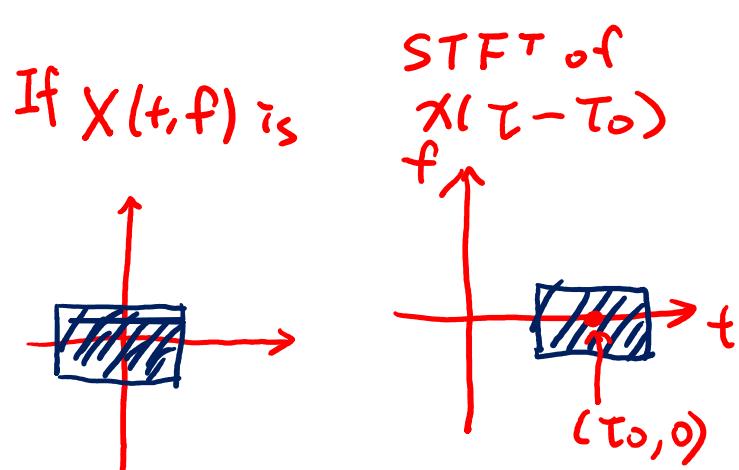
$$\int_{t-B}^{t+B} x(\tau + \tau_0) e^{-j2\pi f \tau} d\tau = X(t + \tau_0, f) e^{j2\pi f \tau_0}$$

$x(\tau - \tau_0)$

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

(3) Modulation property (縱的方向移動)

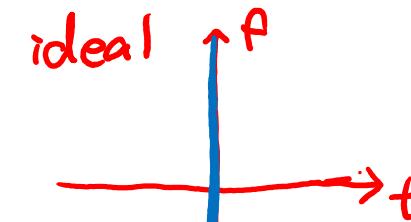
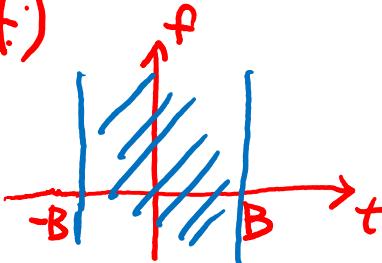
$$\int_{t-B}^{t+B} [x(\tau) e^{j2\pi f_0 \tau}] e^{-j2\pi f \tau} d\tau = X(t, f - f_0)$$



比較: $\mathcal{F}(\delta(t))=1$, $\mathcal{F}(1)=\delta(f)$

(4) Special inputs:

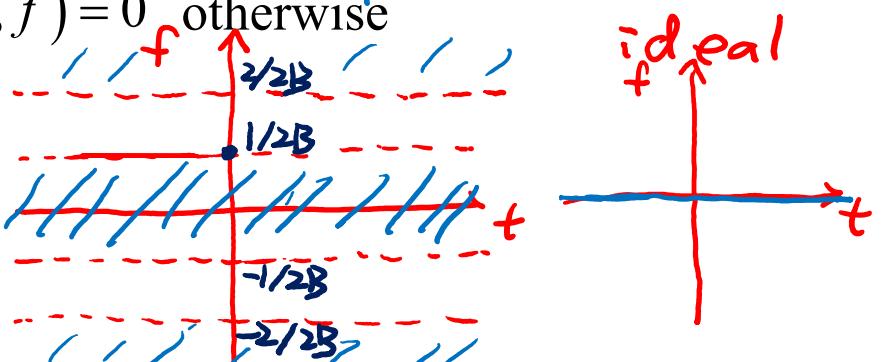
(1) When $x(t) = \delta(t)$,



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

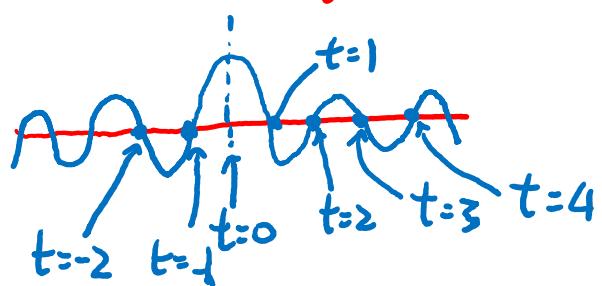
$$X(t, f) = 2B \operatorname{sinc}(2Bf) e^{-j2\pi ft}$$

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



思考：B 值的大小，對解析度的影響是什麼？

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$$



large B better resolution along f-axis poor resolution along t-axis

small B poor resolution along f-axis better resolution along t-axis

uncertainty principle

$$\operatorname{sinc}(2Bf) = 0$$

if $f = \frac{n}{2B}$, n is an integer

(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

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

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

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

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

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

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？

$$\begin{aligned} \text{Shifting} \quad & \int_{-\infty}^{\infty} w(t-\tau)x(\tau-\tau_0)e^{-j2\pi f\tau}d\tau \\ &= \int_{-\infty}^{\infty} w(t-\tau-\tau_0)x(\tau)e^{-j2\pi f\tau}e^{-j2\pi f\tau_0}d\tau \\ &= X(t-\tau_0, f)e^{-j2\pi f\tau_0} \end{aligned}$$

Modulation

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

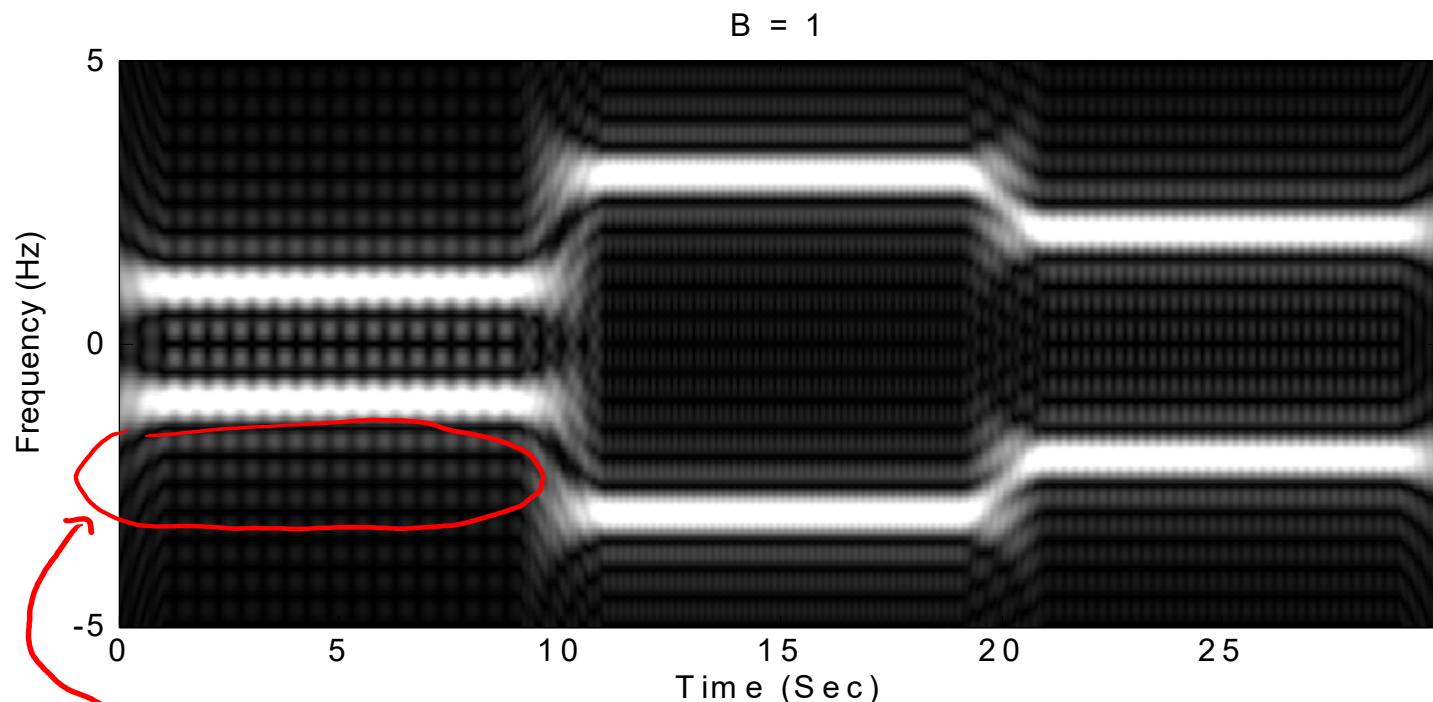
59

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

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

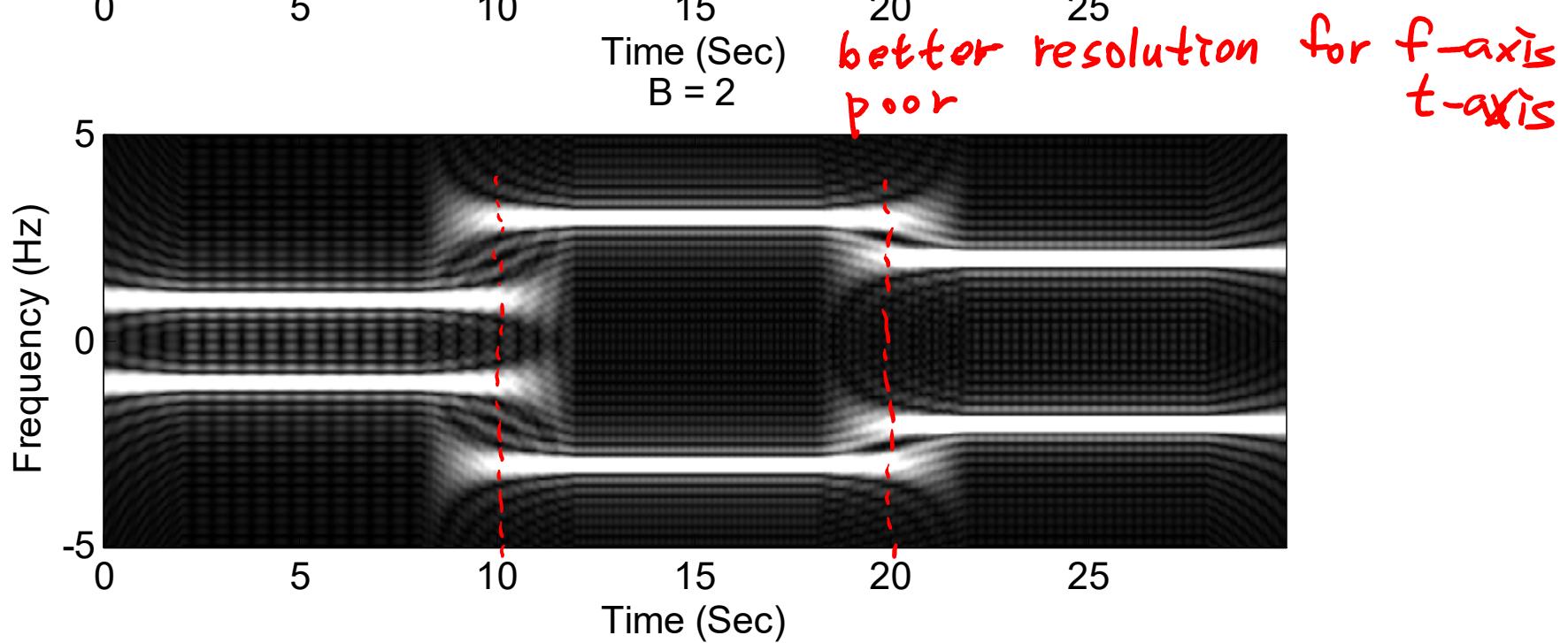
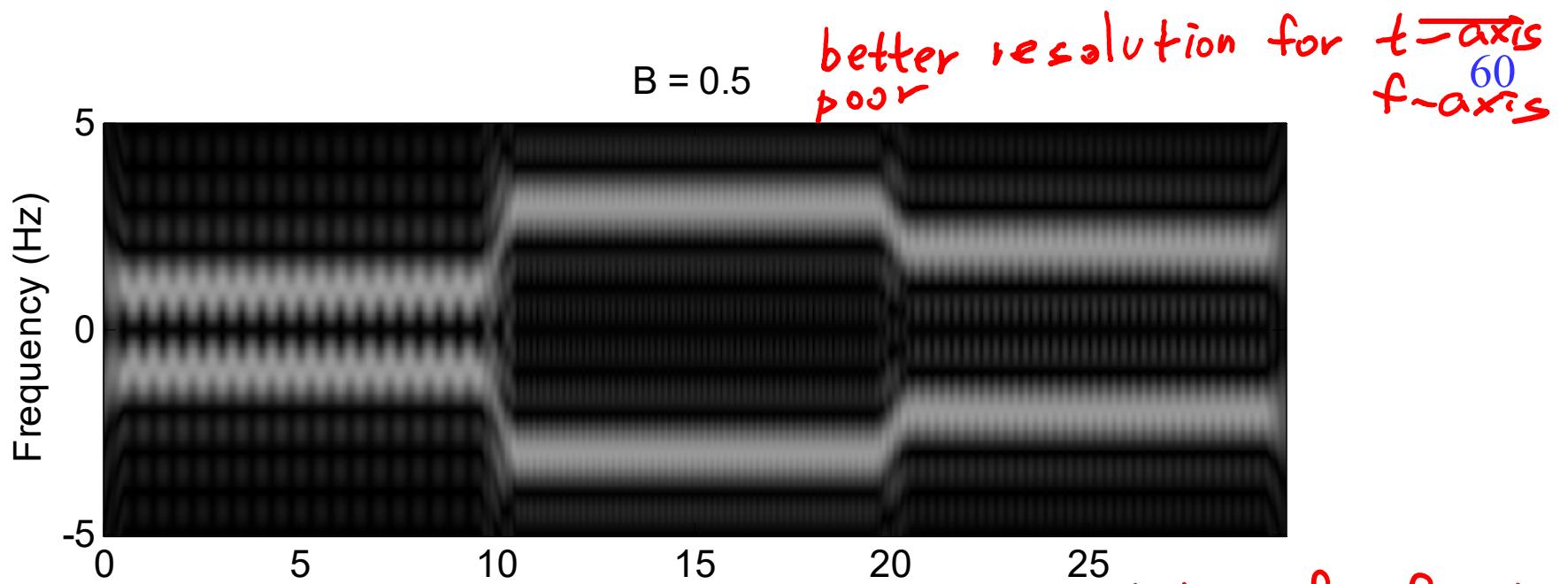
$x(t) = \cos(4\pi t)$ when $t \geq 20$, ± 2 Hz

$$\begin{aligned} & \cos(2\pi t) \\ &= \frac{1}{2}(e^{j2\pi t} + e^{-j2\pi t}) \\ &= \frac{1}{2}(e^{j2\pi t} \cdot 1 + e^{-j2\pi t} \cdot 1) \end{aligned}$$



sidelobe problem

$$\mathcal{F}(x(\tau)w(t-\tau)) = X(f) * \mathcal{F}(w(t-\tau))$$



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

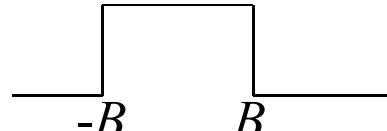
- Compared with other types of time-frequency analysis:

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

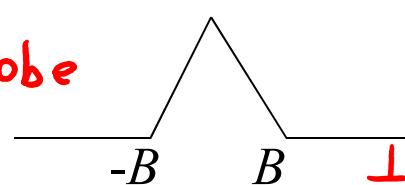
(1) Rectangle



$$\frac{1}{B} B^2 \sin^2(Bf) = \mathcal{F}_L\left(\frac{\text{rect}}{B}\right)$$

(2) Triangle

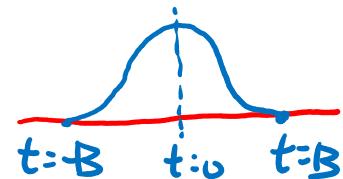
reduce the sidelobe



$$\frac{1}{B} \left(\text{rect}\left(\frac{t}{B}\right) * \text{rect}\left(\frac{t}{B}\right) \right) = \frac{\text{triangular}}{B^2}$$

(3) Hanning

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



(4) Hamming

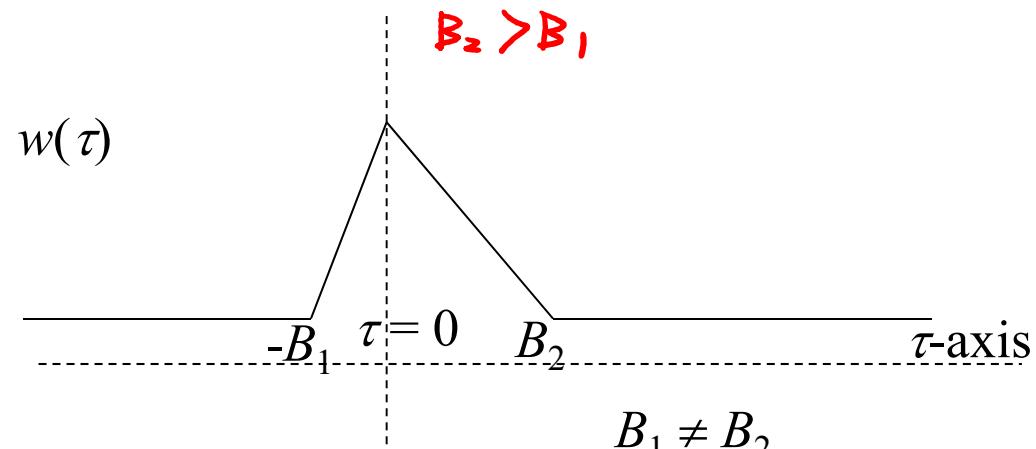
$$w(t) = \begin{cases} 0.54 + 0.46 \cos(\pi t / B) & \text{when } |t| \leq B \\ 0 & \text{otherwise} \end{cases}$$

(5) Gaussian

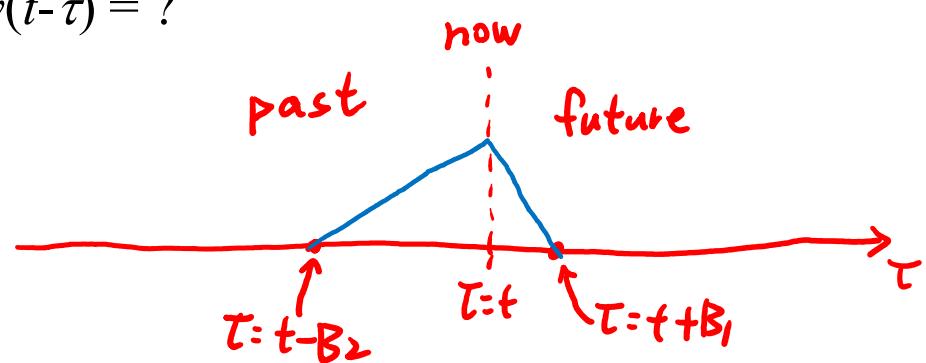
no sidelobe problem $w(t) = \exp(-\pi\sigma t^2)$

無限次 *central limit theorem* $\text{rect} * \text{rect} * \text{rect} \dots = \text{Gaussian}$

(6) Asymmetric window



$$w(t-\tau) = ?$$



smaller $B_1 \rightarrow$ smaller delay
real-time analysis

應用 : seismic wave analysis, collision detection

(The applications that require real-time processing)

onset detection

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

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

STFT

比較：spectrum 為 Fourier transform 的絕對值平方

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

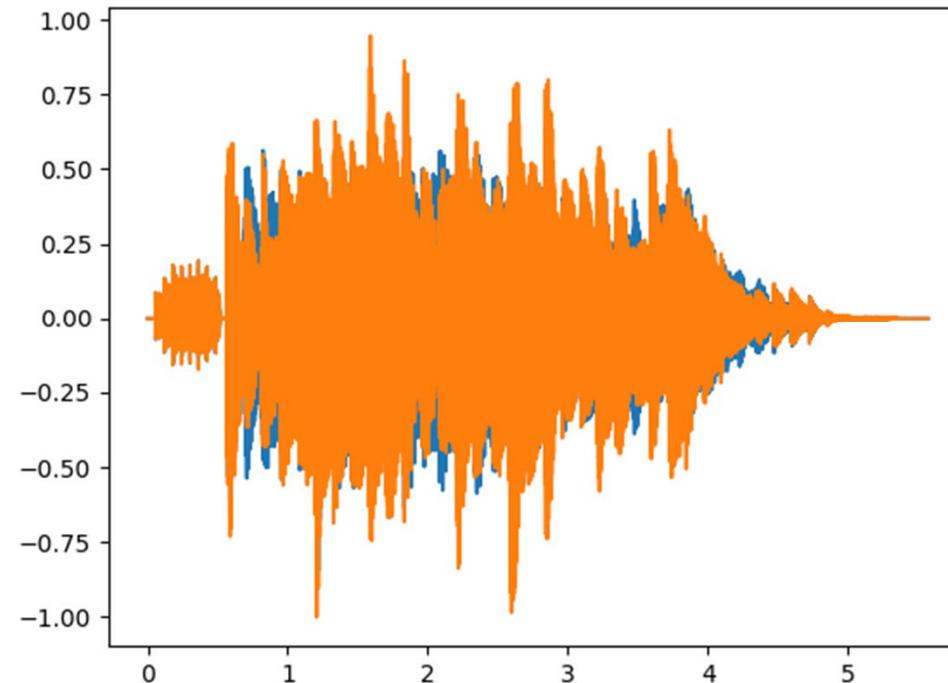
要先import 相關模組： import numpy as np

- 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

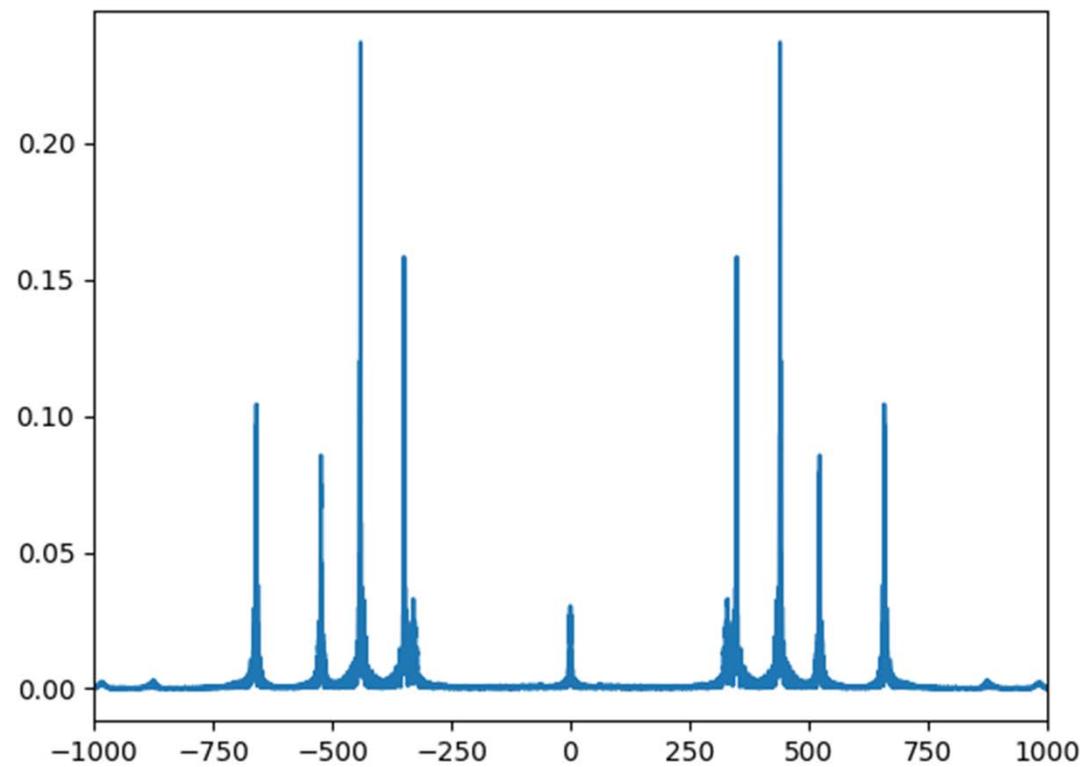
- time = np.arange(0, num_frame)*1/fs
- plt.plot(time, wave_data)
- plt.show()



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`



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 -2¹⁵ ~ 2¹⁵
- 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/>

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

III. Gabor Transform

III-A Definition

Standard Definition:

$$w(t-\tau) \quad w(\tau) = e^{-\pi\tau^2}$$

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

Alternative Definitions:

$$G_{x,1}(t, f) = e^{j\pi ft} G_x(t, f)$$

$$G_{x,1}(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f(\tau-\frac{t}{2})} x(\tau) d\tau$$

$$G_{x,2}(t, f) = \sqrt[4]{2} \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau \quad \text{normalization}$$

$$\int |G_{x,2}(t, f)|^2 dt df = \int |x(t)|^2 dt$$

$$G_{x,3}(t, \omega) = \int_{-\infty}^{\infty} e^{-(\tau-t)^2/2} e^{-j\omega\tau} x(\tau) d\tau$$

$$G_{x,4}(t, \omega) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{(\tau-t)^2}{2}} e^{-j\omega(\tau-\frac{t}{2})} x(\tau) d\tau$$

Main Reference

- S. Qian and D. Chen, [Sections 3-2 ~ 3-6](#) in *Joint Time-Frequency Analysis: Methods and Applications*, Prentice-Hall, 1996.

Other References

- D. Gabor, “Theory of communication”, *J. Inst. Elec. Eng.*, vol. 93, pp. 429-457, Nov. 1946. (最早提出 Gabor transform)
- M. J. Bastiaans, “Gabor’s expansion of a signal into Gaussian elementary signals,” *Proc. IEEE*, vol. 68, pp. 594-598, 1980.
- R. L. Allen and D. W. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*, Wiley- Interscience.
- S. C. Pei and J. J. Ding, “Relations between Gabor transforms and fractional Fourier transforms and their applications for signal processing,” *IEEE Trans. Signal Processing*, vol. 55, no. 10, pp. 4839-4850, Oct. 2007.

註：

許多文獻把 Gabor transform 直接就稱作 short-time Fourier transform (STFT)，實際上，Gabor transform 是 STFT 當中的一個 special case.

III-B Approximation of the Gabor Transform

Although the range of integration is from $-\infty$ to ∞ , due to the fact that

$$e^{-\pi a^2} < 0.00001 \quad \text{when } |a| > 1.9143$$

$$e^{-a^2/2} < 0.00001 \quad \text{when } |a| > 4.7985$$

$$\begin{aligned} e^{-\pi t^2/16} &< 10^{-5} \\ \text{when } |t| &> \frac{1.9143}{\sqrt{16}} \end{aligned}$$

the Gabor transform can be simplified as:

$$G_x(t, f) \approx \int_{\underline{t-1.9143}}^{\underline{t+1.9143}} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

$$G_{x,3}(t, \omega) = \sqrt{\frac{1}{2\pi}} \int_{t-4.7985}^{t+4.7985} e^{-\frac{(\tau-t)^2}{2}} e^{-j\omega(\tau-\frac{t}{2})} x(\tau) d\tau$$

III-C Why Do We Choose the Gaussian Function as a Mask

(1) Among all functions, the Gaussian function has the advantage that the area in time-frequency distribution is minimal.

(和其他的 STFT 相比，比較能夠同時讓 time-domain 和 frequency domain 擁有較好的清晰度)

$w(t)$ 太寬 → time domain 的解析度較差

$w(t)$ 太窄 → $W(f) = FT[w(t)]$ 太寬 → frequency domain 的解析度較差

(2) 由於 Gaussian function 是 FT 的 eigenfunction，因此 Gabor transform 在 time domain 和 frequency domain 的性質將互相對稱

$$\int_{-\infty}^{\infty} \underline{e^{-\pi t^2}} \underline{e^{-j2\pi f t}} dt = \underline{e^{-\pi f^2}}$$

$$\mathcal{Y}(e^{-\pi t^2}) = e^{-\pi f^2}$$

$$\int_{-\infty}^{\infty} e^{-t^2/2} e^{-j\omega t} dt = e^{-f^2/2}$$

Uncertainty Principle (Heisenberg, 1927)

For a signal $x(t)$, if $\sqrt{t} x(t) = 0$ when $|t| \rightarrow \infty$, then

$$\sigma_t \sigma_f \geq 1/4\pi$$

standard deviations in the t and f domains

where $\sigma_t^2 = \int (t - \mu_t)^2 P_x(t) dt$ $\sigma_f^2 = \int (f - \mu_f)^2 P_X(f) df,$
variance

$$\mu_t = \int t P_x(t) dt,$$

$$\mu_f = \int f P_X(f) df$$

$$P_x(t) = \frac{|x(t)|^2}{\int |x(t)|^2 dt},$$

$$P_X(f) = \frac{|X(f)|^2}{\int |X(f)|^2 df},$$



(Proof of Henseinberg's uncertainty principle):

From simplification, we consider the case where $\mu_t = \mu_f = 0$

Then, use Parseval's theorem

$$\sigma_t^2 \sigma_f^2 = \frac{1}{4\pi^2} \frac{\int t^2 |x(t)|^2 dt}{\int |x(t)|^2 dt} \frac{\int |x'(t)|^2 dt}{\int |x(t)|^2 dt}$$

$$\int |x(t)|^2 dt = \int |X(f)|^2 df \quad \text{if } X(f) = FT[x(t)]$$

From Schwarz's inequality $\langle x(t), x(t) \rangle \langle y(t), y(t) \rangle \geq |\langle x(t), y(t) \rangle|^2$

$$\begin{aligned}
& \int t^2 |x(t)|^2 dt \int |x'(t)|^2 dt \geq \left(\left| \int tx^*(t) \frac{d}{dt} x(t) dt \right|^2 + \left| \int tx(t) \frac{d}{dt} x^*(t) dt \right|^2 \right) / 2 \\
& \geq \left| \int \left(tx^*(t) \frac{d}{dt} x(t) + tx(t) \frac{d}{dt} x^*(t) \right) dt \right|^2 / 4 \quad (\text{using } |a+b|^2 + |a-b|^2 \geq 2|a|^2) \\
& = \left| \int t \frac{d}{dt} [x(t)x^*(t)] dt \right|^2 / 4 = \left| tx(t)x^*(t) \Big|_{-\infty}^{\infty} - \int x^*(t)x(t) dt \right|^2 / 4 \\
& = \left[\left. tx(t)x^*(t) \right|_{t \rightarrow \infty} - \left. tx(t)x^*(t) \right|_{t \rightarrow -\infty} \right] - \left| \int x^*(t)x(t) dt \right|^2 / 4 \\
& = \left| \int |x(t)|^2 dt \right|^2 / 4
\end{aligned}$$

$$\sigma_t^2 \sigma_f^2 \geq \frac{1}{16\pi^2} \implies \sigma_t \sigma_f \geq \frac{1}{4\pi}$$

For Gaussian function

$$x(t) = e^{-\pi t^2} \quad X(f) = e^{-\pi f^2}$$

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} e^{-2\pi t^2} dt = ?$$

use $\int_{-\infty}^{\infty} e^{-(at^2+bt)} dt = \sqrt{\pi/a} \cdot e^{b^2/4a}$



$$\int_{-\infty}^{\infty} t^2 |x(t)|^2 dt = \int_{-\infty}^{\infty} t^2 e^{-2\pi t^2} dt = 2 \int_0^{\infty} t^2 e^{-2\pi t^2} dt = ?$$

use $\int_0^{\infty} t^m e^{-at^2} dt = \frac{\Gamma[(m+1)/2]}{2a^{(m+1)/2}}$

$$\Gamma(1/2) = \sqrt{\pi} \quad \Gamma(n+1) = n\Gamma(n)$$

[工具書] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 3rd Ed., 2009.

$$\sigma_t^2 = \frac{\int t^2 |x(t)|^2 dt}{\int |x(t)|^2 dt} = \frac{1}{4\pi},$$

同理, $\sigma_f = \sqrt{\frac{1}{4\pi}}$

所以對 Gaussian function 而言,

$$\sigma_t \sigma_f = \frac{1}{4\pi}$$

滿足下限

$$\sigma_t = \sqrt{\frac{1}{4\pi}}$$

scaled Gaussian function

$$g_0(t) = e^{-\pi \sigma_t^2 t^2}$$

$$g(t) = e^{-\pi t^2}$$

$$g_\sigma(t) = g(\sqrt{\sigma}t) \xrightarrow{\text{def}} G_\sigma(f) = \frac{1}{\sqrt{\sigma}} G\left(\frac{f}{\sqrt{\sigma}}\right)$$

$$\sigma_t = \frac{1}{\sqrt{4\pi\sigma}}$$

$$G_\sigma(f) = \frac{1}{\sqrt{\sigma}} e^{-\pi \frac{f^2}{\sigma}}$$

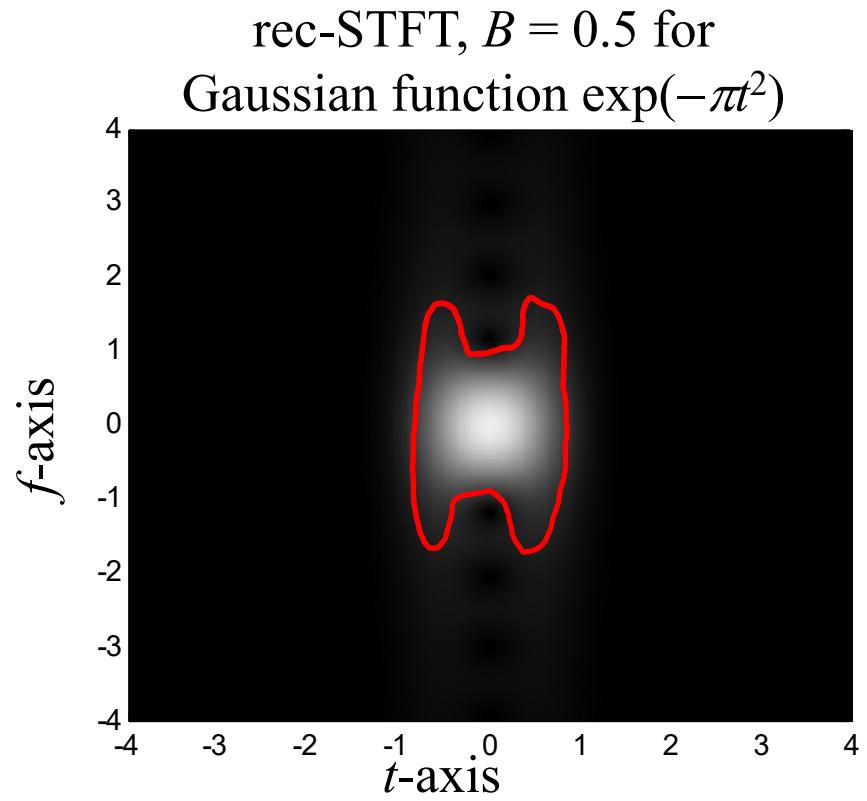
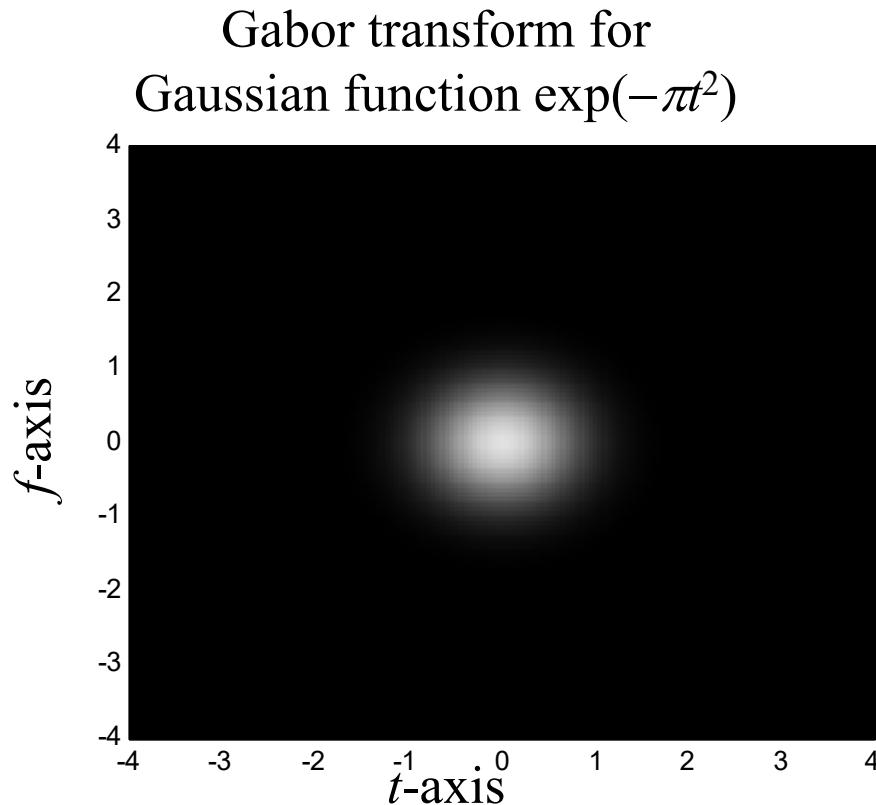
$$\sigma_t \sigma_f = \frac{1}{4\pi} \text{ (unchanged)}$$

Special relation between the Gaussian function and the rectangular function

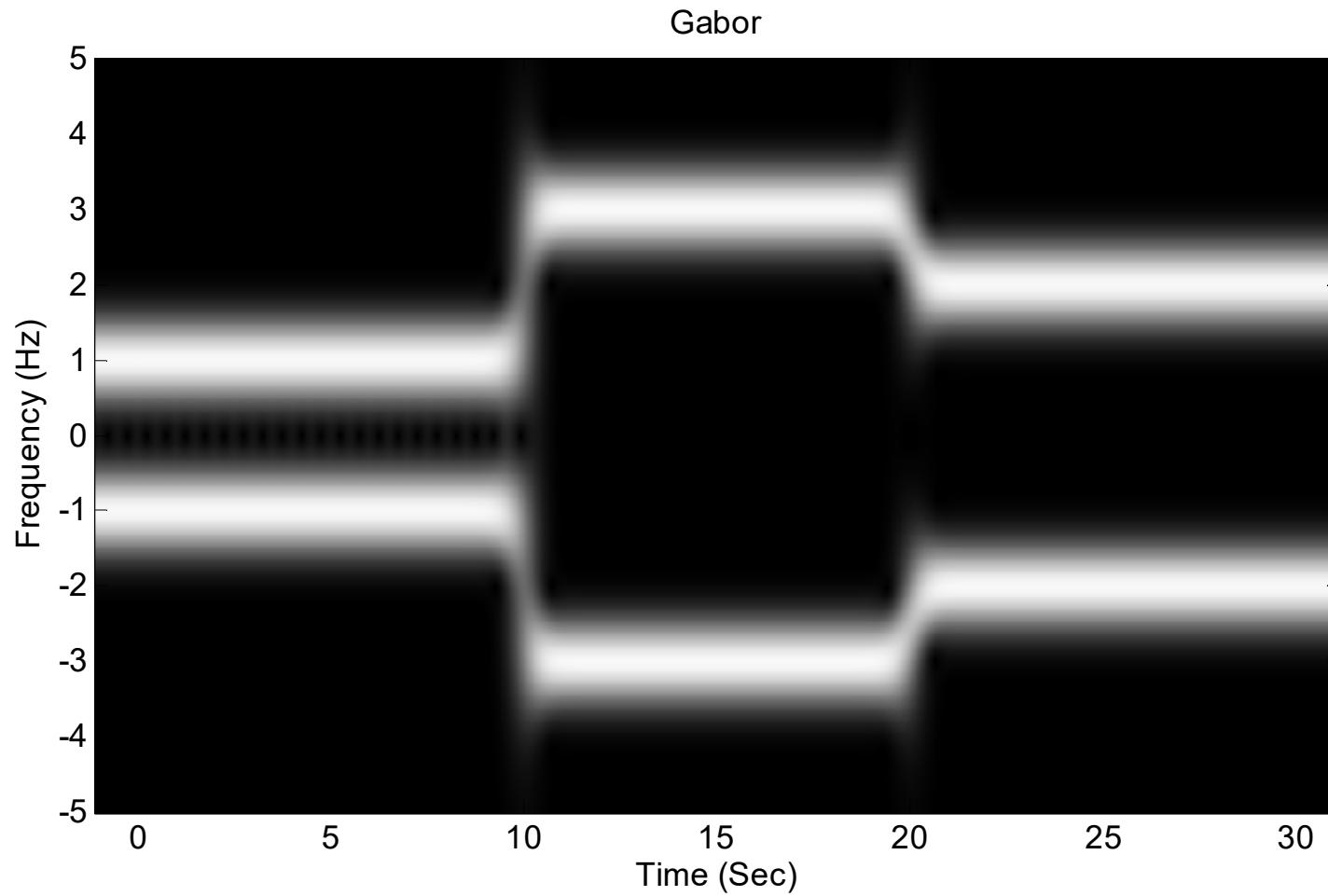
Gaussian function is also an eigenmode in optics, radar system, and other electromagnetic wave systems.

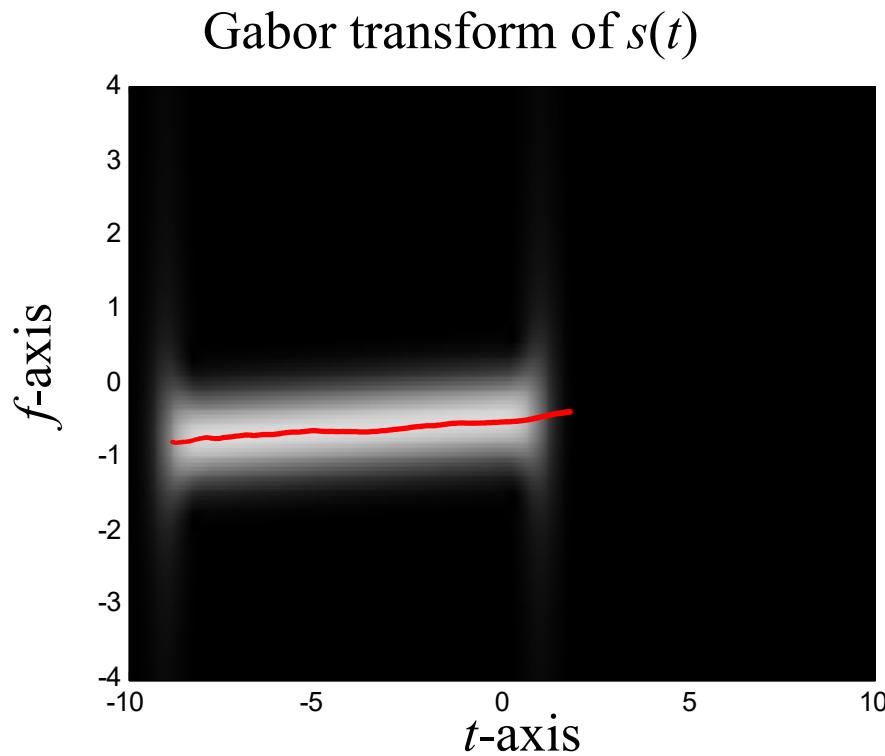
(will be illustrated in the 8th week)

III-D Simulations



$x(t) = \cos(2\pi t)$ when $t < 10$,
 $x(t) = \cos(6\pi t)$ when $10 \leq t < 20$,
 $x(t) = \cos(4\pi t)$ when $t \geq 20$

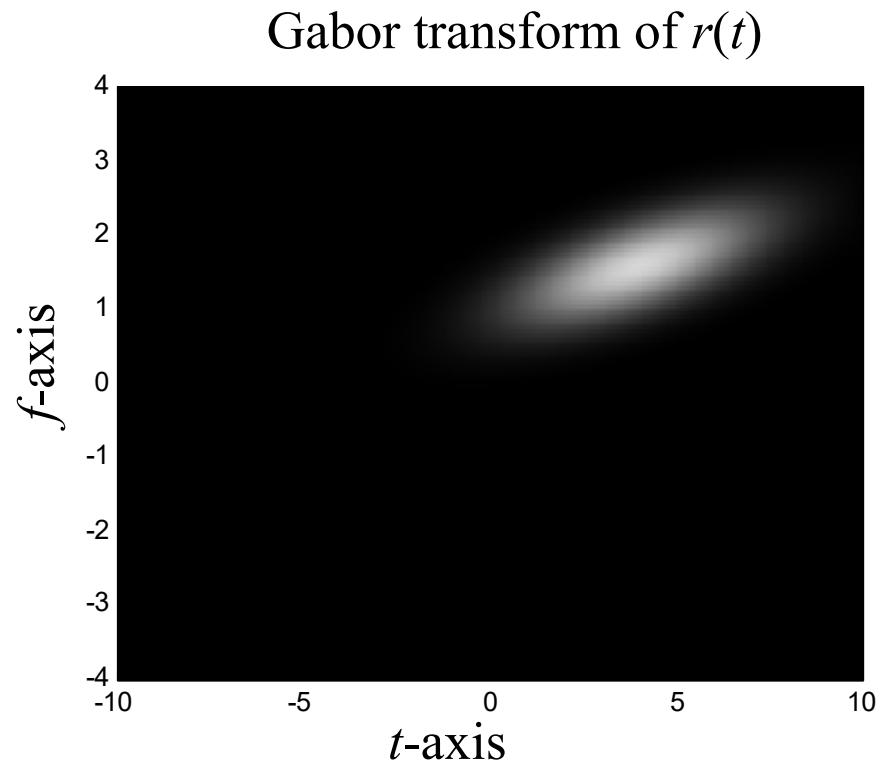




$$s(t) = \exp\left(jt^2/10 - j3t\right) \text{ for } -9 \leq t \leq 1,$$

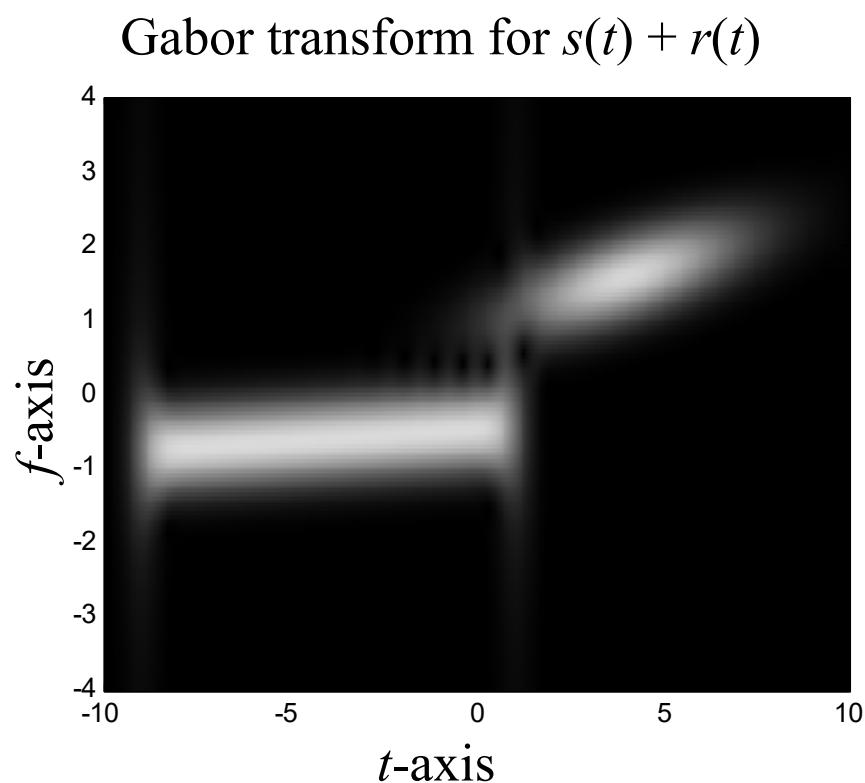
$$s(t) = 0 \text{ otherwise,}$$

$$\frac{t}{10\pi} - \frac{3}{2\pi}$$



$$r(t) = \exp\left(jt^2/2 + j6t\right) \exp\left[-(t-4)^2/10\right]$$

$A(t) e^{j\phi(t)}$
↑
 amplitude
 $A(t)$



III-E Properties of Gabor Transforms

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

(1) Integration property

When $k \neq 0$, $\int_{-\infty}^{\infty} G_x(t, f) e^{j2\pi kt f} df = e^{-\pi(k-1)^2 t^2} x(kt)$

When $k = 0$, $\int_{-\infty}^{\infty} G_x(t, f) df = e^{-\pi t^2} x(0)$

When $k = 1$, $\int_{-\infty}^{\infty} G_x(t, f) e^{j2\pi t f} df = x(t)$ (recovery property)

(2) Shifting property

If $y(t) = x(t - t_0)$, then $G_y(t, f) = G_x(t - t_0, f) e^{-j2\pi f t_0}$.

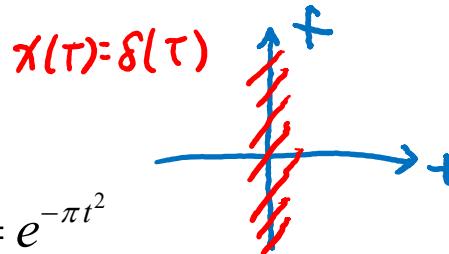
(3) Modulation property

If $y(t) = x(t) \exp(j2\pi f_0 t)$, then $G_y(t, f) = G_x(t, f - f_0)$

(4) Special inputs:

(a) When $x(\tau) = \delta(\tau)$,

$$G_x(t, f) = e^{-\pi t^2}$$



(b) When $x(\tau) = 1$,

$$G_x(t, f) = e^{-j2\pi f t} e^{-\pi f^2} \quad |G_x(t, f)| = e^{-\pi f^2}$$

~~If $x(\tau) = \delta(\tau)$~~ (symmetric for the time and frequency domains)

$$\begin{aligned} G_x(t, f) &= \int x(\tau) e^{-\pi(t-\tau)^2} e^{-j2\pi f t} d\tau \\ &= \int \delta(\tau) e^{-\pi(t-\tau)^2} e^{-j2\pi f t} d\tau \\ &= e^{-\pi(t-\tau)^2} e^{-j2\pi f t} \Big|_{\tau=0} = e^{-\pi t^2} \end{aligned}$$

If $z(\tau) = \alpha x(\tau) + \beta y(\tau)$ and $G_z(t, f)$, $G_x(t, f)$ and $G_y(t, f)$ are their Gabor transforms, then

$$G_z(t, f) = \alpha G_x(t, f) + \beta G_y(t, f)$$

(5) Linearity property

$$\begin{aligned} \text{If } x(\tau) &= 1 \\ G_x(t, f) &= \int e^{-\pi(t-\tau)^2} e^{-j2\pi f t} d\tau \\ &= e^{-\pi t^2} \int e^{-\pi \tau^2 - \tau(-2\pi t - j2\pi f)} d\tau \\ &\quad (\alpha = \pi, b = -2\pi t - j2\pi f) \\ &= e^{-\pi t^2} e^{\frac{4\pi t^2 - 4\pi^2 f^2 + 8\pi^2 t f}{4\pi}} \end{aligned}$$

$$\int_{-\infty}^{\infty} |G_x(t, f)|^2 df = \int_{-\infty}^{\infty} e^{-2\pi(\tau-t)^2} |x(\tau)|^2 d\tau \approx \int_{u-1.9143}^{u+1.9143} e^{-2\pi(\tau-u)^2} |x(\tau)|^2 d\tau$$

$$= e^{-\pi t^2} e^{\frac{4\pi t^2 - 4\pi^2 f^2 + 8\pi^2 t f}{4\pi}}$$

(7) Power decayed property

- If $x(t) = 0$ for $t > t_0$, then

$$\int_{-\infty}^{\infty} |G_x(t, f)|^2 df < e^{-2\pi(t-t_0)^2} \int_{-\infty}^{\infty} |G_x(t_0, f)|^2 df$$

i.e., average of $|G_x(t, f)|^2$ $\underset{(\text{fix } t, \text{ vary } f)}{<} e^{-2\pi(t-t_0)^2}$ \times average of $|G_x(t_0, f)|^2$ $\underset{(\text{fix } t_0, \text{ vary } f)}{<}$ for $t > t_0$.

Proof:

$$G_x(t, f) = \int_{-\infty}^{t_0} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau \quad G_x(t_0, f) = \int_{-\infty}^{t_0} e^{-\pi(\tau-t_0)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

Since $(\tau - t)^2 > (\tau - t_0)^2 + (t_0 - t)^2$ $e^{-\pi(t-\tau)^2} < e^{-\pi(t-t_0)^2} e^{-\pi(t_0-\tau)^2}$

$$G_x(t, f) < e^{-\pi(t-t_0)^2} G_x(t_0, f)$$

- If $X(f) = FT[x(t)] = 0$ for $f > f_0$, then

average of $|G_x(t, f)|^2$ $\underset{(\text{fix } f, \text{ vary } t)}{<} e^{-2\pi(f-f_0)^2}$ \times average of $|G_x(t, f_0)|^2$ for $f > f_0$.

(8) Energy sum property

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G_x(t, f) G_y^*(t, f) df dt = \int_{-\infty}^{\infty} x(\tau) y^*(\tau) d\tau$$

where $G_x(t, f)$ and $G_y(t, f)$ are the Gabor transforms of $x(\tau)$ and $y(\tau)$, respectively.

III-F Scaled Gabor Transforms

$$w(\tau) = e^{-6\pi\tau^2}$$

$$G_x(t, f) = \sqrt[4]{\sigma} \int_{-\infty}^{\infty} e^{-\sigma\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

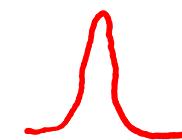


(finite interval form)

$$G_x(t, f) = \sqrt[4]{6} \int_{t - 1.9143/\sqrt{6}}^{t + 1.9143/\sqrt{6}} e^{-6\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

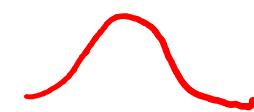
larger σ : higher resolution in the time domain

lower resolution in the frequency domain

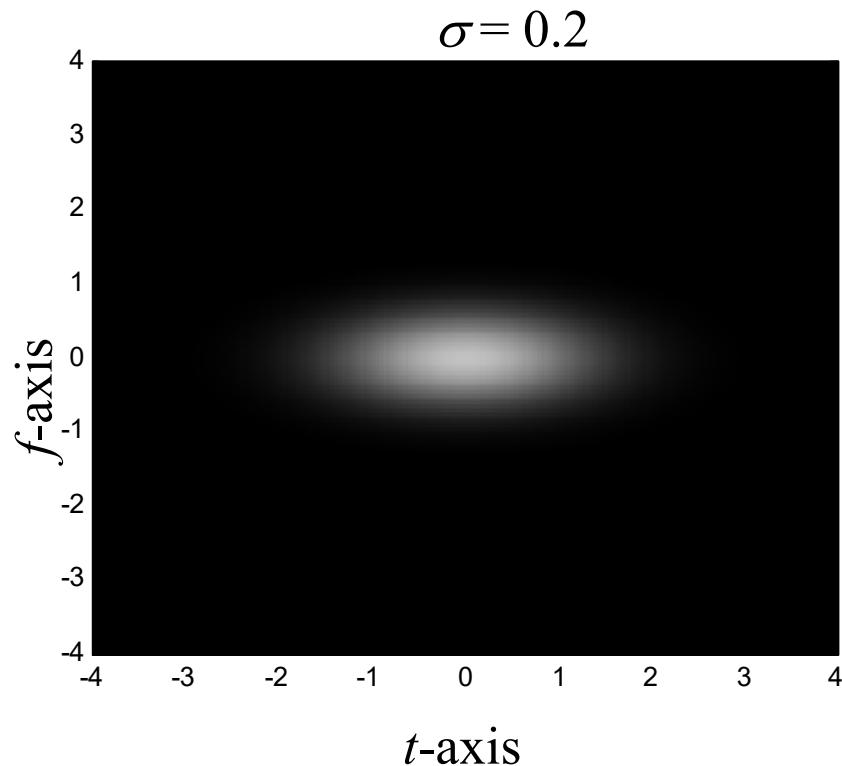


smaller σ : higher resolution in the frequency domain

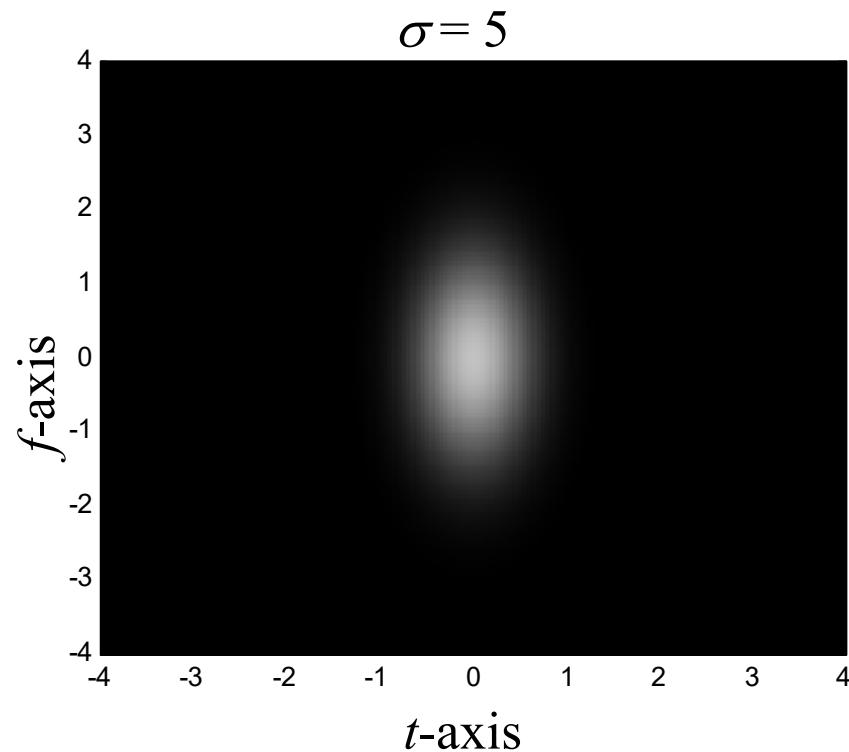
lower resolution in the time domain



Gabor transform for
Gaussian function $\exp(-\pi t^2)$



Gabor transform for
Gaussian function $\exp(-\pi t^2)$



處理對 time resolution 相對上比 frequency resolution 敏感的信號

- (1) Using the generalized Gabor transform with larger σ
- (2) Using other time unit instead of second

例如，原本 t (單位 : sec) f (單位 : Hz)

對聲音信號可以改成

t (單位 : 0.1 sec) f (單位 : 10 Hz)

III-G Gabor Transforms with Adaptive Window Width

For a signal,

when the instantaneous frequency varies fast \rightarrow larger σ

when instantaneous frequency varies slowly \rightarrow smaller σ

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

$\sigma(t)$ is a function of t

S. C. Pei and S. G. Huang, “STFT with adaptive window width based on the chirp rate,” *IEEE Trans. Signal Processing*, vol. 60, issue 8, pp. 4065-4080, 2012.

附錄四：Matlab 寫程式的原则

- (1) 迴圈能避免就儘量避免
- (2) 儘可能使用 Matrix 及 Vector operation
- (3) 能夠不在迴圈內做的運算，則移到迴圈外
- (4) 寫一部分即測試，不要全部寫完再測試（縮小範圍比較容易 debug）
- (5) 先測試簡單的例子，成功後再測試複雜的例子

註：作業 Matlab Program (or Python program) 鼓勵各位同學儘量用精簡而快速的方式寫。Program 執行速度越快，分數就越高。

一些重要的 Matlab 指令

(1) **function**: 放在第一行，可以將整個程式函式化

(2) **tic, toc**: 計算時間

tic 為開始計時，toc 為顯示時間

(3) **find**: 找尋一個 vector 當中不等於 0 的 entry 的位置

範例： $\text{find}([1\ 0\ 0\ 1]) = [1, 4]$

$\text{find}(\text{abs}([-5:5]) \leq 2) = [4, 5, 6, 7, 8]$

(因為 $\text{abs}([-5:5]) \leq 2 = [0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0]$)

(4) **'** : Hermitian (transpose + conjugation) , **.'** : transpose

(5) **imread**: 讀圖 , **image, imshow, imagesc**: 將圖顯示出來 ,

(註：較老的 Matlab 版本 imread 要和 double 並用

```
A=double(imread('Lena.bmp'));
```

(6) **imwrite**: 製做圖檔

- (7) `xlsread`: 由 Excel 檔讀取資料
- (8) `xlswrite`: 將資料寫成 Excel 檔
- (9) `aviread`: 讀取 video 檔，限副檔名為 avi
- (10) `VideoReader`: 讀取 video 檔
- (11) `VideoWriter`: 製作 video 檔
- (12) `dlmread`: 讀取 *.txt 或其他類型檔案的資料
- (13) `dlmwrite`: 將資料寫成 *.txt 或其他類型檔案

附錄五：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

附錄六：寫 Python 版本程式可能會用到的重要指令

建議必安裝模組

pip install numpy

pip install scipy

pip install scipy

pip install opencv-python

(1) 定義函式：使用def

(2) 計算時間

import time

start_time = time.time() #獲取當前時間

end_time = time.time()

total_time = end_time - start_time #計算時間差來得到總執行時間

感謝2021年擔任助教的蔡昌廷同學

(3) 讀取圖檔、輸出圖檔(建議使用opencv)

```
import cv2  
  
image = cv2.imread(file_name) #預設color channel為BGR  
cv2.imwrite(file_name, image) #需將color channel轉為BGR
```

(4) 尋找array中滿足特定條件的值的位置

(相當於 Matlab 的 find 指令)

```
import numpy as np  
  
a = np.array([0, 1, 2, 3, 4, 5])  
index = np.where(a > 3) #回傳array([4, 5])  
print(index)  
    (array([4, 5], dtype=int64),)
```

```
index[0][0]
```

4

```
index[0][1]
```

5

```
A1= np.array([[1,3,6],[2,4,5]])  
index = np.where(A1 > 3)  
print(index)
```

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 4 & 5 \end{bmatrix}$$

(array([0, 1, 1], dtype=int64), array([2, 1, 2], dtype=int64))

(代表滿足 $\mathbf{A}_1 > 3$ 的點的位置座標為 [0, 2], [1, 1], [1, 2]

```
[index[0][0], index[1][0]]
```

[0, 2]

```
[index[0][1], index[1][1]]
```

[1, 1]

```
[index[0][2], index[1][2]]
```

[1, 2]

(5) Hermitian、transpose

```
import numpy as np  
result = np.conj(matrix.T)    # Hermitian  
result = matrix.T    # transpose
```

(6) 讀取 Matlab 當中的 mat 檔

```
data = scipy.io.loadmat('***.mat')  
y = np.array(data['y'])    # 假設 y 是 ***.mat 當中儲存的資料
```