III. Gabor Transform

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

III-A Definition

Standard Definition:

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

Alternative Definitions:

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

$$\int_{-\infty}^{\infty} \left(2^{\frac{1}{4}} e^{-\pi\tau^2}\right)^2 d\tau = 1$$

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

normalization

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

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

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

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.

Note：

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

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

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

the Gabor transform can be simplified as:

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

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

$$e^{-\pi\sigma a^2} < 10^{-3} \quad \text{for} \quad |a| > \frac{1.9143}{\sqrt{\sigma}}$$

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)

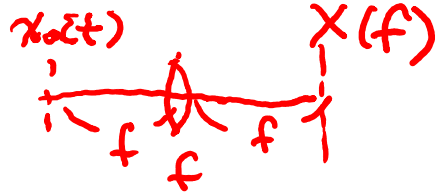
$w(t)$ 太寬 $\rightarrow$ time domain 的解析度較差

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

(2) Special relation between the Gaussian function and the rectangular function

(Note): 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

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



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

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

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

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

(will be illustrated in the 8th week)

Uncertainty Principle (Heisenberg, 1927)

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

$$\boxed{\sigma_t\,\sigma_f \geq 1/4\pi}$$

σ_t^2: variance
σ_t: standard deviation

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

$= E((t-\mu_t)^2)$

$\mu_t = \int t\,P_x(t)\,dt$, $= E(t)$ $\qquad$ $\mu_f = \int f\,P_X(f)\,df$

expected value

$$P_x(t) = \frac{|x(t)|^2}{\int |x(t)|^2\,dt}, \qquad P_X(f) = \frac{|X(f)|^2}{\int |X(f)|^2\,df},$$

$X(f) = FT(x(t))$

(Proof of Henseinberg's uncertainty principle):

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

Then, use Parseval's theorem

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

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

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

$$\int t^2 \,|\, x(t)\,|^2 \, dt \int |\, x'(t)\,|^2 \, dt \geq \left(\left| \int t x^*(t) \frac{d}{dt} x(t) dt \right|^2 + \left| \int t x(t) \frac{d}{dt} x^*(t) dt \right|^2 \right) / 2$$

$$\geq \left| \int \left(t x^*(t) \frac{d}{dt} x(t) + t x(t) \frac{d}{dt} x^*(t) \right) dt \right|^2 / 4 \quad \text{(using } |a+b|^2 + |a-b|^2 \geq 2|a|^2 \text{)}$$

$$= \left| \int t \frac{d}{dt} \left[x(t) x^*(t) \right] dt \right|^2 / 4 = \left| t x(t) x^*(t) \Big|_{-\infty}^{\infty} - \int x^*(t) x(t) \, dt \right|^2 / 4$$

$$= \left| \left[t x(t) x^*(t) \Big|_{t \to \infty} - t x(t) x^*(t) \Big|_{t \to -\infty} \right] - \int x^*(t) x(t) \, dt \right|^2 / 4$$

$$= \left| \int |x(t)|^2 \, dt \right|^2 / 4$$

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

For Gaussian function

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

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

$P_x(t)$

$$= \frac{1/2^{\frac{5}{2}}\pi}{1/2^{\frac{1}{2}}} = \frac{1}{2^2\pi} = \frac{1}{4\pi}$$

Since $\mu_t = 0$ $\quad \mu_f = 0$

$$\int_{-\infty}^{\infty} |x(t)|^2\,dt = \int_{-\infty}^{\infty} e^{-2\pi t^2}\,dt = ? \quad \sqrt{\frac{1}{2}}$$

$a = 2\pi,\ b = 0$

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

$m = 2,\ a = 2\pi$

$$\int_{-\infty}^{\infty} t^2 |x(t)|^2\,dt = \int_{-\infty}^{\infty} t^2 e^{-2\pi t^2}\,dt = 2\int_0^{\infty} t^2 e^{-2\pi t^2}\,dt = 2\frac{\Gamma(3/2)}{2(2\pi)^{3/2}} = \frac{1}{2^{5/2}\pi}$$

Gamma function

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

$$\Gamma(1/2) = \sqrt{\pi} \qquad \Gamma(n+1) = n\Gamma(n), \qquad \Gamma(3/2) = \sqrt{\pi}/2$$

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

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

所以對 Gaussian function 而言，

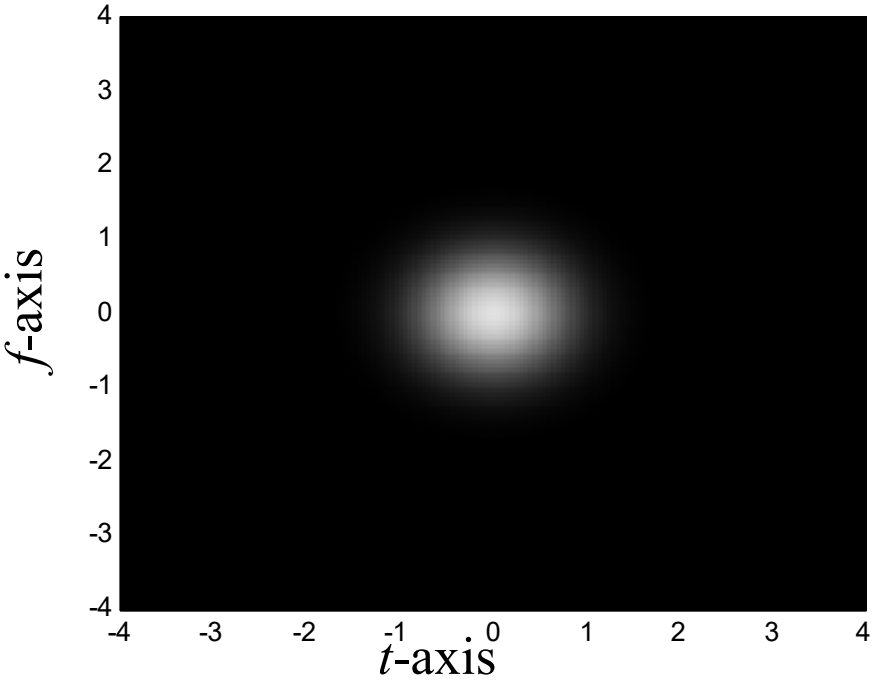
$$\sigma_t \sigma_f = \frac{1}{4\pi}$$

滿足下限

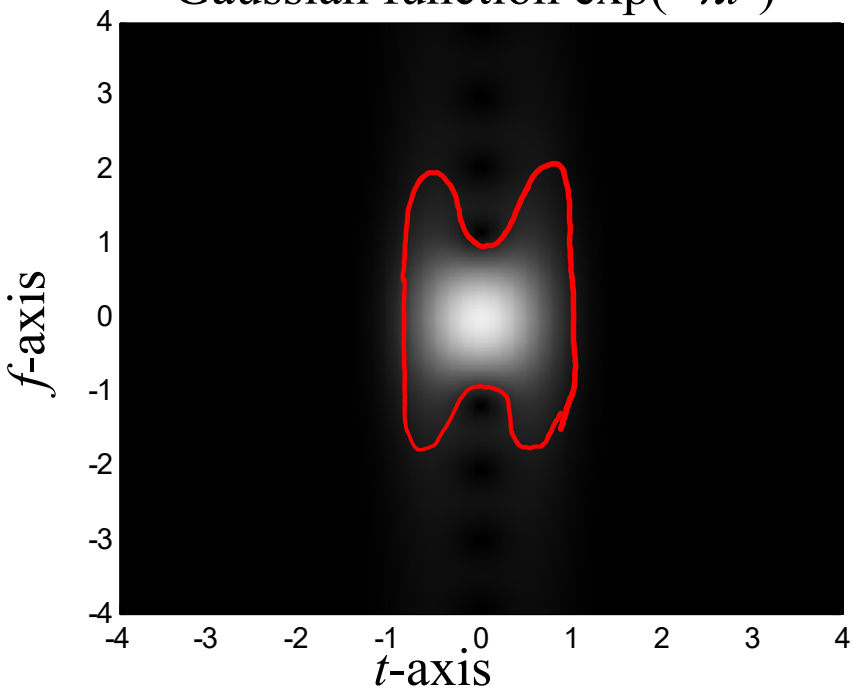
[工具書] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 3rd Ed., 2009.

III-D Simulations

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



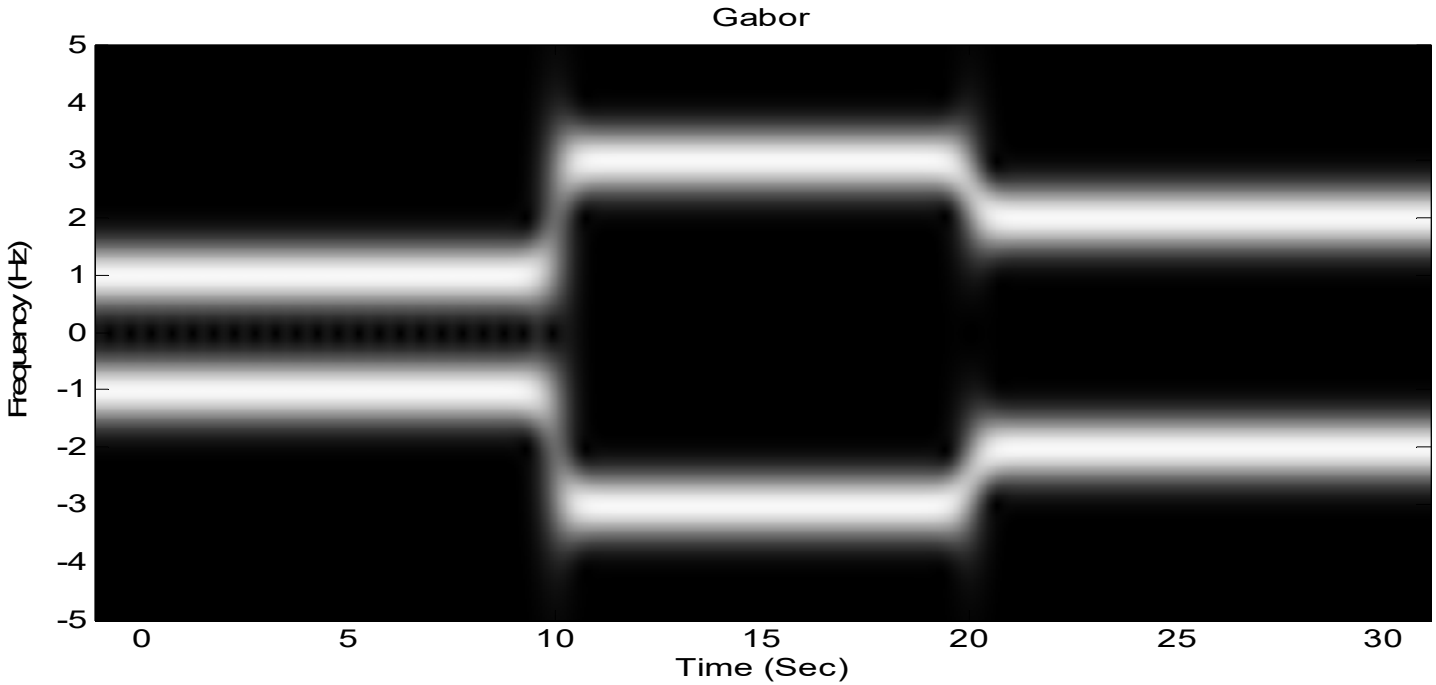
rec-STFT, $B = 0.5$ for
Gaussian function $\exp(-\pi t^2)$



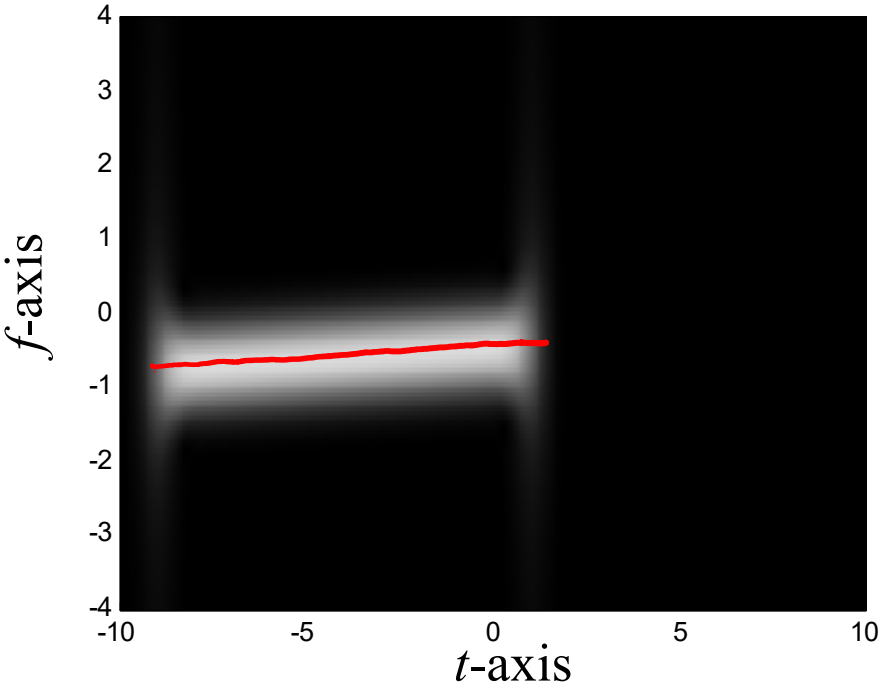
$x(t) = \cos(2\pi t)$ when $t < 10$,

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

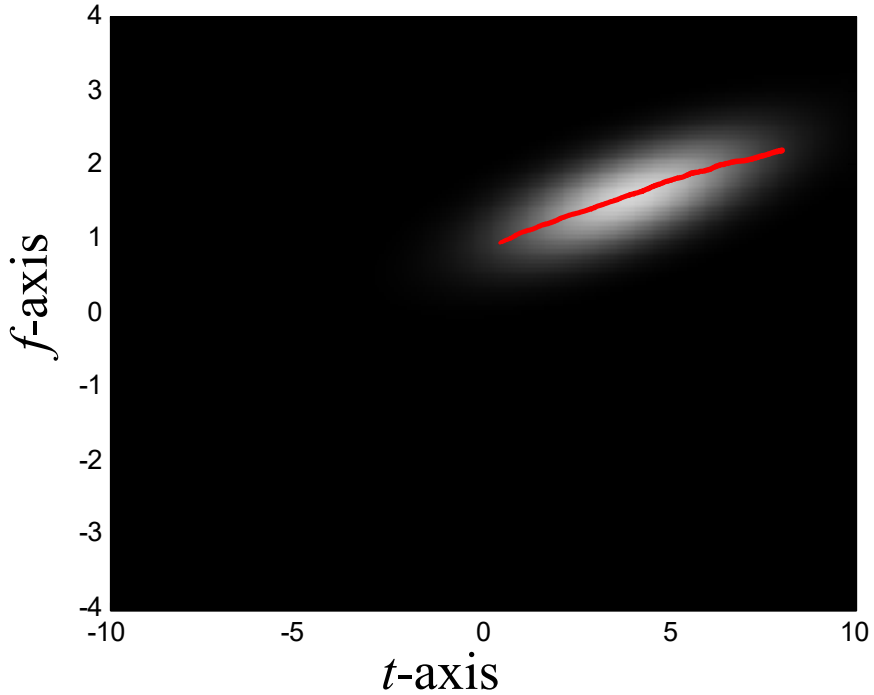
$x(t) = \cos(4\pi t)$ when $t \ge 20$



Gabor transform of $s(t)$



Gabor transform of $r(t)$



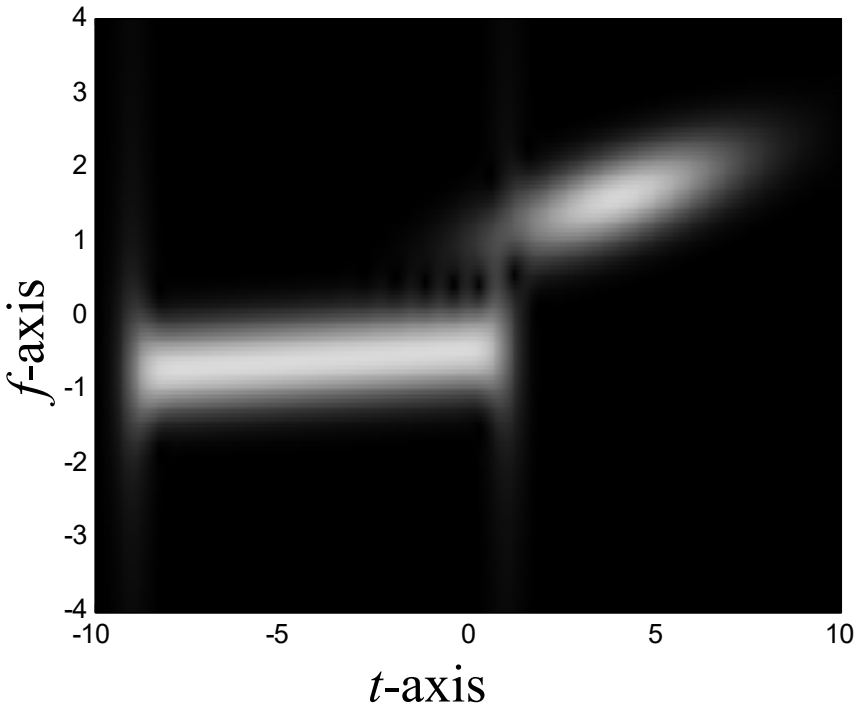
$$s(t) = \exp\left(jt^2/10 - j3t\right) \text{ for } -9 \le t \le 1,$$

$s(t) = 0$ otherwise,

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

$$r(t) = \underbrace{\exp\left(jt^2/2 + j6t\right)}_{\frac{t}{2\pi} + \frac{3}{\pi}} \underbrace{\exp\left[-(t-4)^2/10\right]}_{\text{amplitude } A(t)}$$

Gabor transform for $s(t) + r(t)$



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

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 $G_y(t,f) = G_x(t, f - f_0)$

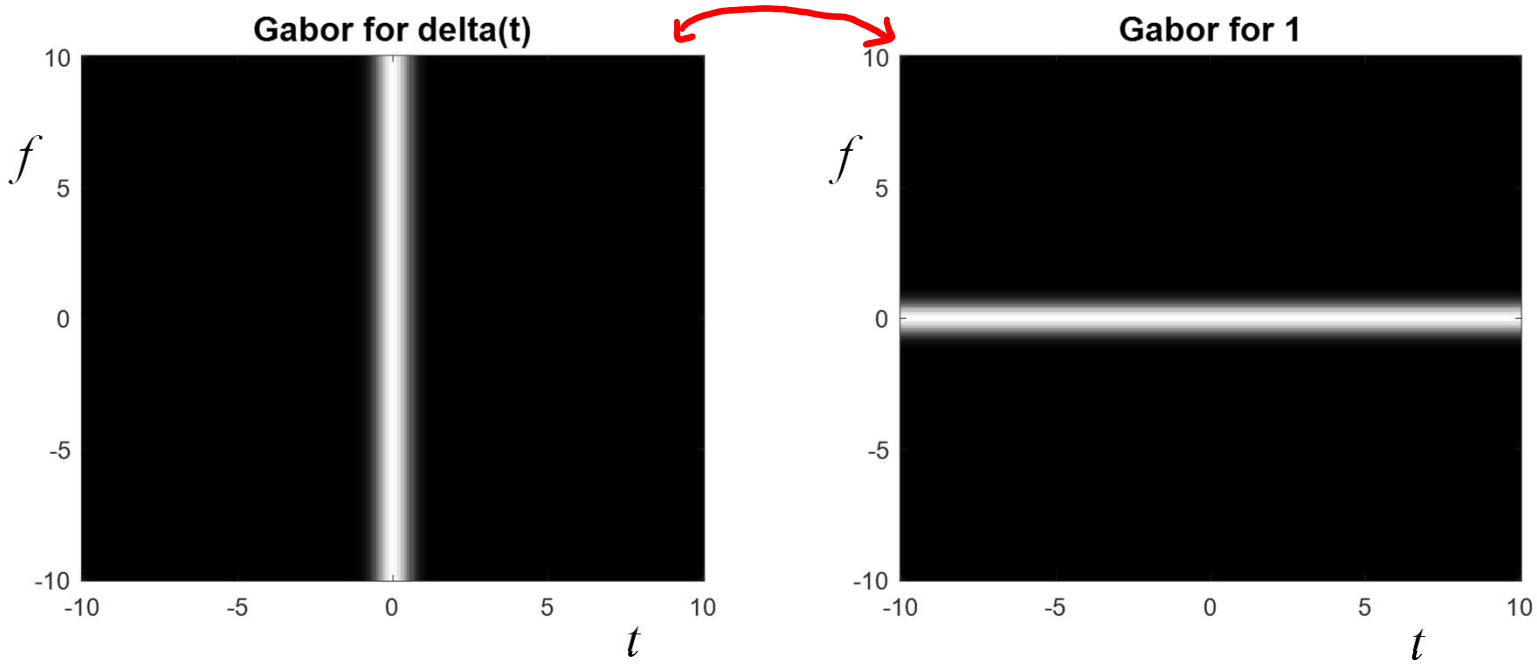
(4) Special inputs:

(a) When $x(\tau) = \delta(\tau)$, $\quad G_x(t,f) = e^{-\pi t^2}$

(b) When $x(\tau) = 1$, $\quad G_x(t,f) = e^{-j2\pi f t} e^{-\pi f^2}$ $\quad |G_x(t,f)| = e^{-\pi f^2}$

(symmetric for the time and frequency domains)

rotated by 90 degree



(5) Power decayed property

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

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

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

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

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

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

(6) Linearity property

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

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

(7) Power integration property:

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

(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

$w(t) = \sqrt[4]{\sigma}\, e^{-\sigma\pi t^2}$

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

(finite interval form)

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

$w(\tau) = \sqrt[4]{\sigma}\, e^{-\sigma\pi\tau^2}$

$$\int_{-\infty}^{\infty} |w(\tau)|^2 d\tau = \int_{-\infty}^{\infty} \sqrt{\sigma}\, e^{-2\sigma\pi\tau^2} d\tau = \int_{-\infty}^{\infty} e^{-2\pi\tau_1^2} d\tau_1 = \int_{-\infty}^{\infty} \left|e^{-\pi\tau^2}\right|^2 d\tau$$

$(\tau_1 = \sqrt{\sigma}\,\tau)$

larger σ: higher resolution in the time domain

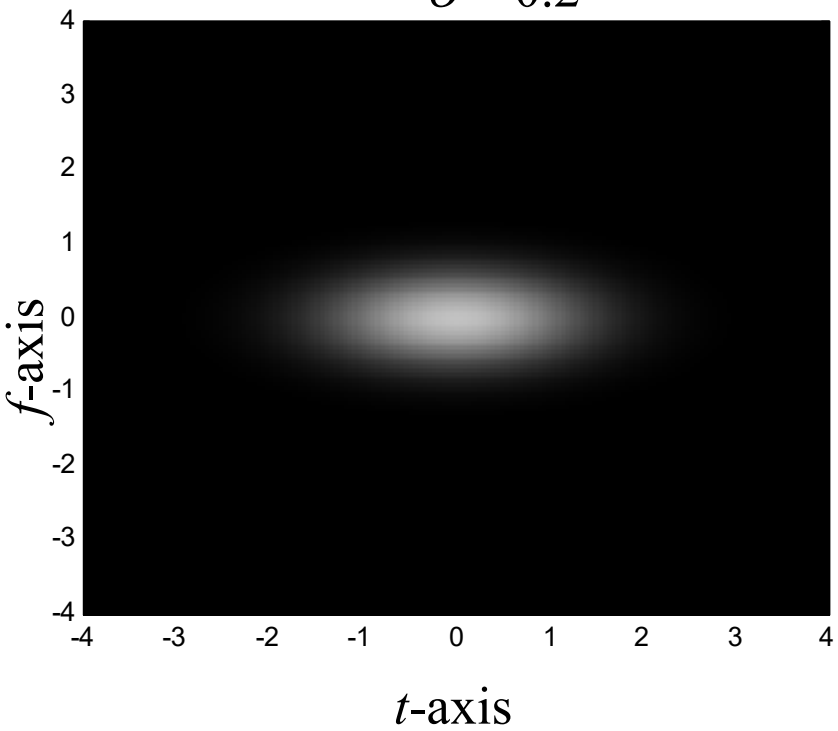
(narrow window) lower resolution in the frequency domain

smaller σ: higher resolution in the frequency domain

(wider window) lower resolution in the time domain

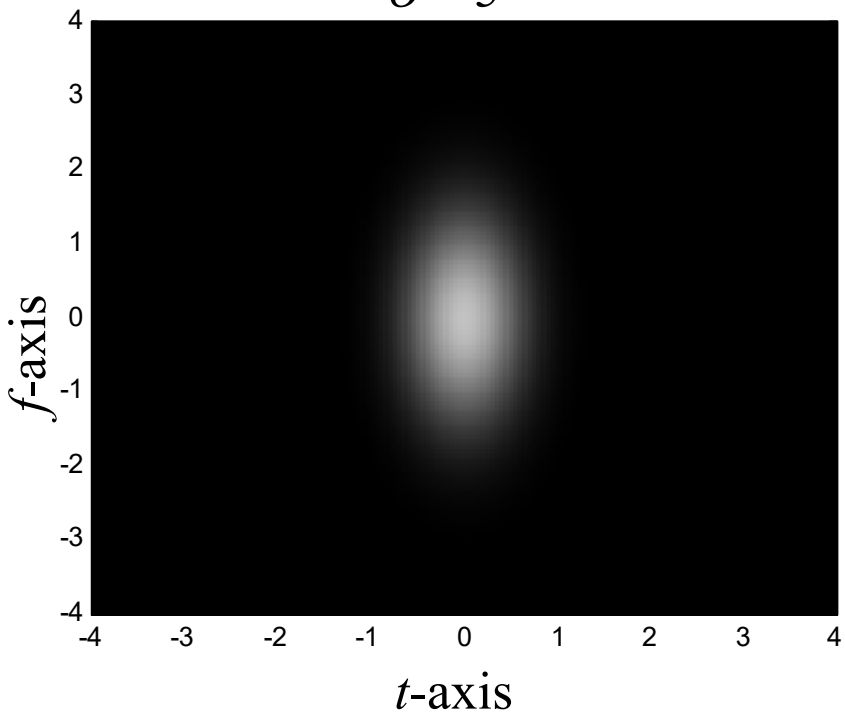
Gabor transform for
Gaussian function $\exp(-\pi t^2)$

$\sigma = 0.2$



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

$\sigma = 5$



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

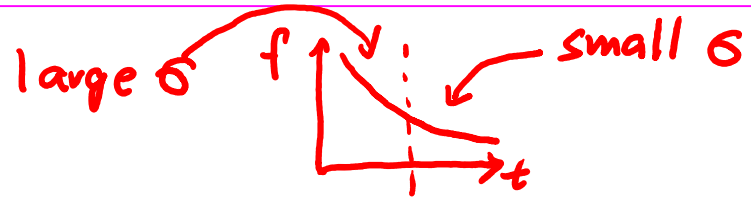
(2) Using other time unit instead of second

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

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

$\sigma(t)$ 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.

附錄五：Matlab 寫程式的原則以及部分常用的指令

(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：find([1 0 0 1]) = [1, 4]

find(abs([-5:5])<=2) = [4, 5, 6, 7, 8]

(因為 abs([-5:5])<=2 = [0 0 0 1 1 1 1 1 0 0 0])

(4) ′ : Hermitian (transpose + conjugation)， .′ : transpose

(5) imread: 讀圖

(註： 較老的 Matlab 版本 imread 要和 double 並用

A=double(imread(‘Lena.bmp’));

(6) image: 將圖顯示出來，

(i) 顯示灰階圖

image(A) % A has the size of *M*x*N*x1

colormap(gray(256)

(ii) 顯示彩色圖，整數的情形

image(A) % A has the size of *M*x*N*x3 and the entries are integer

(iii) 顯示彩色圖，非整數的情形

image(A) % A has the size of *M*x*N*x3 and the entries are non-integer

(7) imshow, imagesc : 也可用來顯示圖

(8) imwrite: 製做圖檔

(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) xlsread: 由 Excel 檔讀取資料

(13) xlswrite: 將資料寫成 Excel 檔

(14) dlmread: 讀取 *.txt 或其他類型檔案的資料

(15) dlmwrite: 將資料寫成 *.txt 或其他類型檔案

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 opencv-python
pip install matplotlib
```

(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、顯示圖檔、輸出圖檔

(方法一)

```
import cv2
image = cv2.imread(file_name)  #預設color channel為BGR
cv2.imshow('test', image)
# 若 image 的值非整數，要改成 cv2.imshow('test', image/255)
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imwrite(file_name, image)   #需將color channel轉為BGR
```

(方法二)

```
import matplotlib.pyplot as plt
image = plt.imread(file_name)  #預設color channel為RGB
plt.imshow(image)
# 若 image 的值非整數，要改成 plt.imshow(image/255)
plt.show()
plt.imsave(file_name, image)   #需將color channel轉為RGB
```

(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
```

$$\mathbf{A_1} = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 4 & 5 \end{bmatrix}$$

```
A1= np.array([[1,3,6],[2,4,5]])
index = np.where(A1 > 3)
print(index)
    (array([0, 1, 1], dtype=int64), array([2, 1, 2], dtype=int64))
    (代表滿足 A1 > 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) 在 Python 當中讀取 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
```

IV. Implementation

IV-A Method 1: Direct Implementation

以 STFT 為例

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

$\int \to \sum$

$d\tau \to \Delta_t$

Converting into the Discrete Form

$$t=n\Delta_t,\quad f=m\Delta_f,\quad \tau=p\Delta_t$$

$$X(n\Delta_t,m\Delta_f)=\sum_{p=-\infty}^{\infty} w((n-p)\Delta_t)x(p\Delta_t)e^{-j2\pi pm\Delta_t\Delta_f}\Delta_t$$

direct implementation method

Suppose that $w(t)\cong 0$ for $|t|>B$, $\quad B/\Delta_t=Q$

$w((n-p)\Delta_t)\cong 0$ for $|n-p|\Delta_t>B$

$|n-p|>\frac{B}{\Delta_t}:Q$

$$X(n\Delta_t,m\Delta_f)=\sum_{p=n-Q}^{n+Q} w((n-p)\Delta_t)x(p\Delta_t)e^{-j2\pi pm\Delta_t\Delta_f}\Delta_t$$

T F

$TF(2Q+1)\cong 2TFQ$

$O(TFQ)$

Problem：對 scaled Gabor transform 而言，$Q=?$

T: number of n ; F: number of m

$\frac{1.9143}{\Delta_t\sqrt{\sigma}}$

- **Constraint for Δ_t** (The only constraint for the direct implementation method)

$y(-\tau) \rightarrow Y(-f)$

$y(-(\tau-t)) \rightarrow e^{-j2\pi f t} Y(-f)$

To avoid the aliasing effect,

$\Delta_t < 1/2\Omega$, Ω is the bandwidth of ? $w(t-\tau)x(\tau)$

Suppose that the bandwidth of $x(\tau)$ is Ω_x

$w(\tau)$ is Ω_w

$w(t-\tau)$ Ω_w

$\Delta_t < \frac{1}{2(\Omega_x+\Omega_w)}$

$\Omega = \Omega_x + \Omega_w$

There is no constraint for Δ_f when using the direct implementation method.

Four Implementation Methods

page 19

$\Delta t = \frac{1}{40000}$, $T = \frac{1.5}{\Delta t} = 60000$

$\Delta f = 1$, $f = -20000 \sim 20000$ $F = 40000$

scaled Gabor $Q = \frac{1.9143}{\Delta t \sqrt{200}} \cong 5400$

(1) Direct implementation

Complexity: $\Theta(TFQ)$

假設 t-axis 有 T 個 sampling points, f-axis 有 F 個 sampling points

(2) FFT-based method

Complexity: $\Theta(TN\log N)$

unbalanced form

$\Theta(\frac{T}{S} N\log N)$

(3) FFT-based method with recursive formula

Complexity: $\Theta(TF)$

(4) Chirp-Z transform method

Complexity: $\Theta(TN\log N)$

complexity

(1) > (4) > (2) > (3)

(A) Direct Implementation

Advantage： simple, flexible

Disadvantage ： higher complexity

(B) DFT-Based Method

Advantage ： lower complexity

Disadvantage ： with some constraints

(i) $\Delta t < \frac{1}{2(\Omega_x + \Omega_w)}$

(ii) $\Delta t \Delta f = \frac{1}{N}$ (iii) $N \geq 2Q+1$

(C) Recursive Method

Advantage ： least complexity

Disadvantage ： more constraints error propogation

(iv) rectangular window

(D) Chirp Z Transform

Advantage ： flexible (only constraints are $\Delta t < \frac{1}{2(\Omega_x + \Omega_w)}$, $N > 2Q+1$)

Disadvantage ： higher complexity than (B)(C)

IV-B Method 2: FFT-Based Method

(i) $\Delta t < \frac{1}{2(\Omega_x + \Omega_N)}$

Constraints：(ii) $\Delta_t \Delta_f = 1/N$, (N is some integer)

(iii) $N = 1/(\Delta_t \Delta_f) \geqq 2Q + 1$: ($\Delta_t \Delta_f$ 是整數的倒數)

Standard form of the DFT $Y[m] = \sum_{n=0}^{N-1} y[n] e^{-j\frac{2\pi mn}{N}}$ $\mathcal{O}(N \log N)$

$$X(n\Delta_t, m\Delta_f) = \sum_{p=n-Q}^{n+Q} w((n-p)\Delta_t)\, x(p\Delta_t)\, e^{-j2\pi pm\Delta_t\Delta_f}\, \Delta_t$$

$\Delta_t \Delta_f = 1/N$

$$X(n\Delta_t, m\Delta_f) = \sum_{p=n-Q}^{n+Q} w((n-p)\Delta_t)\, x(p\Delta_t)\, e^{-j\frac{2\pi pm}{N}}\, \Delta_t$$

$e^{-j\frac{2\pi}{N}pm} = e^{-j\frac{2\pi}{N}qm}\, e^{-j\frac{2\pi}{N}(n-Q)m}$

$p = n-Q \Rightarrow q = 0$

$p = n+Q \Rightarrow q = 2Q$

$q = p-(n-Q) \rightarrow p = (n-Q)+q$

$$X(n\Delta_t, m\Delta_f) = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} \sum_{q=0}^{2Q} w((\cancel{q-Q})\Delta_t)\, x((q+n-Q)\Delta_t)\, e^{-j\frac{2\pi qm}{N}}$$

$Q-q$

$x_1(q)$

Note that the input of the N-point FFT should have N points (others are set to zero).

$$\boxed{X\left(n\Delta_t, m\Delta_f\right) = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} \sum_{q=0}^{N-1} x_1(q) e^{-j\frac{2\pi qm}{N}}} \quad , \quad q = p-(n-Q) \rightarrow p = (n-Q)+q$$

where $x_1(q) = w\left((Q-q)\Delta_t\right) x\left((n-Q+q)\Delta_t\right)$ for $\boxed{0 \le q \le 2Q,}$

$x_1(q) = 0$ for $2Q < q < N$.

$n\text{-}Q \le n\text{-}Q\text{+}q \le n\text{+}Q$

$Q \ge Q - q \ge -Q$

$k = q\text{-}Q$

$$\boxed{X\left(n\Delta_t, m\Delta_f\right) = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} DFT\left(x_1(q)\right)}$$

If $x((n+k)\Delta_t)$ is out of range, set $x((n+k)\Delta_t)=0$

where $x_1(q) = w(k\Delta_t)\, x\left((n+k)\Delta_t\right)$ for $\overset{0}{0} \le q \le \overset{40}{2Q}$, $-Q \le k \le Q$ $(k = q\text{-}Q)$

$x_1(q) = 0$ for $\overset{40}{2Q} < q < \overset{400}{N}$.

(Suppose that $w(t) = w(-t)$)

$w((Q-q)\Delta_t) = w(-k\Delta_t)$

$= w(k\Delta_t)$

a part of $x(n\Delta_t)$

$n-Q \sim n+Q$

$$\boxed{X\left(n\Delta_t, m\Delta_f\right) = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} DFT\left(x_1(q)\right)}$$

注意：

(1) 可以使用 Matlab 的 FFT 指令來計算 $\sum_{q=0}^{N-1} x_1(q) e^{-j\frac{2\pi qm}{N}} = X_1(m)$

since $x_1(q)$ varies with n

(2) 對每一個固定的 n，都要計算一次下方的式子

T F

$$X\left(n\Delta_t, m\Delta_f\right) = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} \sum_{q=0}^{N-1} x_1(q) e^{-j\frac{2\pi qm}{N}} = \Delta_t e^{j\frac{2\pi(Q-n)m}{N}} X_1(m)$$

(fixed n)

DFT multiply $\Delta_t e^{j\frac{2\pi}{N}(Q-n)m}$

$T(N\log N + F)$

$\cong TN\log N$

(3) Complexity = ?

假設 $t = n_0\Delta_t,\ (n_0+1)\,\Delta_t,\ (n_0+2)\,\Delta_t,\ \cdots\cdots,\ (n_0+T\text{-}1)\Delta_t$

$n_0 = 0,\ T = 601$

$f = m_0\,\Delta_f,\ (m_0+1)\,\Delta_f,\ (m_0+2)\,\Delta_f,\ \cdots\cdots,\ (m_0+F\text{-}1)\Delta_f$

Step 1: Calculate n_0, m_0, T, F, N, Q

Step 2: $n = n_0$

Step 3: Determine $x_1(q)$

Step 4: $X_1(m) = \mathrm{FFT}[x_1(q)]$

Step 5: Convert $X_1(m)$ into $X(n\Delta_t, m\Delta_f)$

page 120

when using Matlab

$m = f/\Delta_f$

$m_1 = \mathrm{mod}(m, N)+1$

$N = 400$

$X_1[-100] = X_1[300]$

$X_1[-99] = X_1[301]$

$X_1[-1] = X_1[399]$

$$X\left(n\Delta_t, m\Delta_f\right) = X_1(?)\times ?$$

$= \Delta_t e^{j\frac{2\pi}{N}(\theta - \cancel{\theta_0})m} X_2(m),$

$X_2(m) = X_1(\mathrm{mod}(m,N))$

取餘數

$$X_1[m+N] = \sum_q x_1(q)\, e^{-j\frac{2\pi}{N}qm}\, e^{-j\frac{2\pi}{N}qN} = X_1[m]$$

$$X_1[m] = \sum_{q=0}^{N-1} x_1(q)\, e^{-j\frac{2\pi qm}{N}} \qquad X_1[m] = X_1[m+N]$$

$0 \le q \le N-1$

$0 \le m \le N-1$

Step 6: Set $n = n+1$ and return to Step 3 until $n = n_0+T\text{-}1$.

IV-C Method 3: Recursive Method

- **A very fast way for implementing the rec-STFT**

(n 和 $n-1$ 有 recursive 的關係)

$$\boxed{X\left(n\Delta_t, m\Delta_f\right) = \sum_{p=n-Q}^{n+Q} x\left(p\Delta_t\right) e^{-j\frac{2\pi pm}{N}} \Delta_t}$$

$$X\left((n-1)\Delta_t, m\Delta_f\right) = \sum_{p=n-1-Q}^{n-1+Q} x(p\Delta_t) e^{-j\frac{2\pi}{N}pm} \Delta_t$$

(1) Calculate $X(\min(n)\Delta_t, m\Delta_f)$ by the N-point FFT

$$\boxed{X\left(n_0\Delta_t, m\Delta_f\right) = \Delta_t e^{j\frac{2\pi(Q-n_0)m}{N}} \sum_{q=0}^{N-1} x_1(q) e^{-j\frac{2\pi qm}{N}}}, \quad n_0 = \min(n),$$

$$x_1(q) = x\left((n-Q+q)\Delta_t\right) \quad \text{for } q \le 2Q, \qquad x_1(q) = 0 \quad \text{for } q > 2Q$$

$N\log N + F + (T-1)(2F) \cong N\log N + F + 2TF \cong 2TF$

(2) Applying the recursive formula to calculate $X(n\Delta_t, m\Delta_f)$, $\Theta(TF)$

$n = n_0 + 1 \sim \max(n)$

$p = n-1-Q$

$$\boxed{\begin{aligned} X\left(n\Delta_t, m\Delta_f\right) &= X\left((n-1)\Delta_t, m\Delta_f\right) - x\left((n-Q-1)\Delta_t\right) e^{-j2\pi(n-Q-1)m/N} \Delta_t \\ &\quad + x\left((n+Q)\Delta_t\right) e^{-j2\pi(n+Q)m/N} \Delta_t \end{aligned}}$$

T點 F點

two F-point vectors

$p = n+Q$

IV-D Method 4: Chirp Z Transform

$$\exp\left(-j2\pi\, pm\Delta_t\Delta_f\right)=\exp\left(-j\pi\, p^2\Delta_t\Delta_f\right)\exp\left(j\pi\,(p-m)^2\Delta_t\Delta_f\right)\exp\left(-j\pi m^2\Delta_t\Delta_f\right)$$

For the STFT

$$X\left(n\Delta_t,m\Delta_f\right)=\sum_{p=n-Q}^{n+Q} w\left((n-p)\Delta_t\right)x\left(p\Delta_t\right)e^{-j2\pi\, pm\Delta_t\Delta_f}\Delta_t$$

$$X\left(n\Delta_t,m\Delta_f\right)=\Delta_t e^{-j\pi\, m^2\Delta_t\Delta_f}\sum_{p=n-Q}^{n+Q} w\left((n-p)\Delta_t\right)x\left(p\Delta_t\right)e^{-j\pi\, p^2\Delta_t\Delta_f}e^{j\pi\,(p-m)^2\Delta_t\Delta_f}$$

T F

Step 1 multiplication $2Q+1$

Step 2 convolution $2N\log N$

Step 3 multiplication F

$T(2Q+1+2N\log N+F)\cong 2TN\log N$ $\quad \Theta(TN\log N)$

Step 1 $x_1[p] = w((n-p)\Delta_t)\, x(p\Delta_t)\, e^{-j\pi p^2 \Delta_t \Delta_f}$ $\qquad n\text{-}Q \le p \le n+Q$

Step 2 $X_2[n,m] = \sum_{p=n-Q}^{n+Q} x_1[p]\, c[m-p]$ $\qquad c[m] = e^{j\pi m^2 \Delta_t \Delta_f}$ IDFT(DFT(x_1)DFT(c))

Step 3 $X(n\Delta_t, m\Delta_f) = \Delta_t\, e^{-j\pi m^2 \Delta_t \Delta_f}\, X_2[n,m]$

Step 2 在計算上，需要用到 linear convolution 的技巧

Question: <u>Step 2 要用多少點的 DFT</u>?

• Illustration for the Question on Page 124

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

• Case 1

When length($x[n]$) = N, length($h[n]$) = K, N and K are finite,

$\longrightarrow$ length($y[n]$) = $N+K-1$,

Using the ($N+K-1$)-point DFTs (學信號處理的人一定要知道的常識)

• Case 2

$x[n]$ has finite length but $h[n]$ has infinite length ????

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

- Case 2

$x[n]$ has finite length but $h[n]$ has infinite length

$x[n]$ 的範圍為 $n \in [n_1, n_2]$，範圍大小為 $N = n_2 - n_1 + 1$

$h[n]$ 無限長

$y[n]=\sum_k x[n-k]h[k]$ $y[n]$ 每一點都有值 (範圍無限大)

但我們只想求出 $y[n]$ 的其中一段

希望算出的 $y[n]$ 的範圍為 $n \in [m_1, m_2]$，範圍大小為 $M = m_2 - m_1 + 1$

$h[n]$ 的範圍？

要用多少點的 FFT？

$$y[n]=x[n]*h[n]=\sum_{k=-\infty}^{\infty}x[n-k]h[k]$$

改寫成 $y[n]=x[n]*h[n]=\sum_{s=n_1}^{n_2}x[s]h[n-s]$

$$y[n]=x[n_1]h[n-n_1]+x[n_1+1]h[n-n_1-1]+x[n_1+2]h[n-n_1-2]$$
$$+\cdots\cdots+x[n_2]h[n-n_2]$$

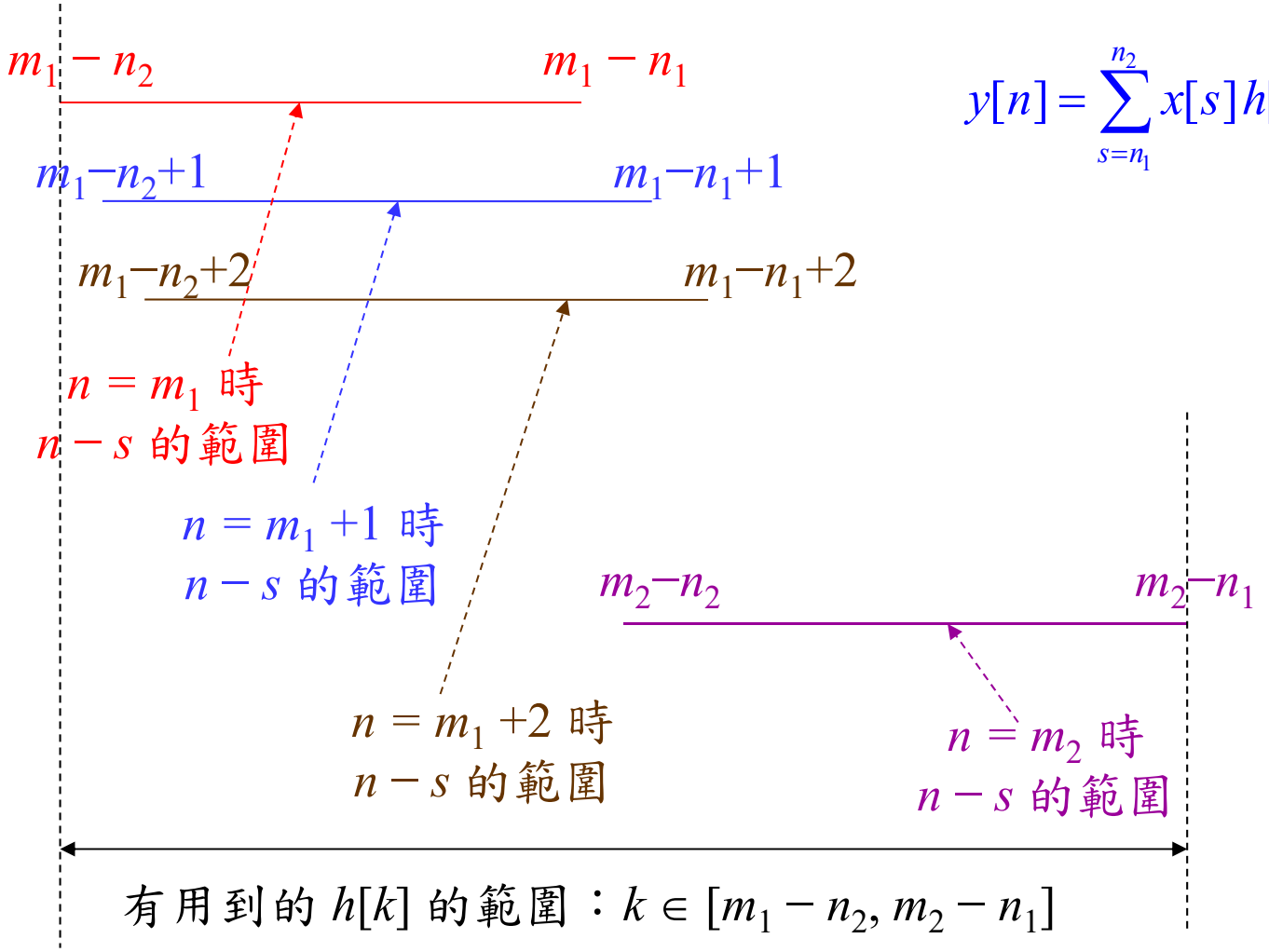
當 $n=m_1$

$$y[m_1]=x[n_1]h[m_1-n_1]+x[n_1+1]h[m_1-n_1-1]+x[n_1+2]h[m_1-n_1-2]$$
$$+\cdots\cdots+x[n_2]h[m_1-n_2]$$

當 $n=m_2$

$$y[m_2]=x[n_1]h[m_2-n_1]+x[n_1+1]h[m_2-n_1-1]+x[n_1+2]h[m_2-n_1-2]$$
$$+\cdots\cdots+x[n_2]h[m_2-n_2]$$

$$y[n] = \sum_{s=n_1}^{n_2} x[s]h[n-s]$$



有用到的 $h[k]$ 的範圍：$k \in [m_1 - n_2, m_2 - n_1]$

所以，有用到的 $h[k]$ 的範圍是 $k \in [m_1 - n_2, m_2 - n_1]$

範圍大小為 $m_2 - n_1 - m_1 + n_2 + 1 = N + M - 1$

FFT implementation for Case 2

$x_1[n] = x[n + n_1]$ for $n = 0, 1, 2, \ldots, N-1$

$x_1[n] = 0$ for $n = N, N+1, N+2, \ldots\ldots, L-1$ $L = N + M - 1$

$h_1[n] = h[n + m_1 - n_2]$ for $n = 0, 1, 2, \ldots, L-1$

$$y_1[n] = IFFT_L\left(FFT_L\{x_1[n]\}\, FFT_L\{h_1[n]\}\right)$$

$y[n] = y_1[n - m_1 + N - 1]$ for $n = m_1, m_1+1, m_1+2, \ldots, m_2$

IV-E Unbalanced Sampling for STFT and WDF

將 pages 115 and 119 的方法作修正

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

$w(n\Delta_t - p\Delta_\tau) = w(nS\Delta_\tau - p\Delta_\tau)$

$$X(n\Delta_t, m\Delta_f)=\sum_{p=nS-Q}^{nS+Q} w((nS-p)\Delta_\tau)x(p\Delta_\tau)e^{-j2\pi pm\Delta_\tau\Delta_f}\Delta_\tau$$

where $\boxed{t=n\Delta_t}$, $f=m\Delta_f$, $\boxed{\tau=p\Delta_\tau}$, $B=Q\Delta_\tau$ (假設 $w(t)\cong 0$ for $|t|>B$),

ex: $\Delta_\tau = \frac{1}{44100}$ $\Delta_t = 0.01$

$S=441$

$S=\Delta_t/\Delta_\tau$ $\Delta_t \neq \Delta_\tau$

註：Δ_τ (sampling interval for the input signal)

Δ_t (sampling interval for the output t-axis) can be different.

However, it is better that $S=\Delta_t/\Delta_\tau$ is an integer.

When (1) $\Delta_\tau \Delta_f = 1/N$, (2) $N = 1/(\Delta_\tau \Delta_f) > 2Q+1$: ($\Delta_\tau \Delta_f$只要是整數的倒數即可)

(3) $\Delta_\tau < 1/2\Omega$, Ω is the bandwidth of $w(\tau - t)x(\tau)$

$\Delta_\tau < \frac{1}{2(\Omega_x + \Omega_w)}$

i.e., $|FT\{w(\tau - t)x(\tau)\}| = |X(t,f)| \approx 0$ when $|f| > \Omega$

$$\boxed{X(n\Delta_t, m\Delta_f) = \sum_{p=nS-Q}^{nS+Q} w((nS-p)\Delta_\tau)\, x(p\Delta_\tau)\, e^{-j\frac{2\pi pm}{N}} \Delta_\tau}$$

令 $q = p - (nS-Q) \rightarrow p = (nS-Q) + q$

$$\boxed{X(n\Delta_t, m\Delta_f) = \Delta_\tau \boxed{e^{j\frac{2\pi(Q-nS)m}{N}}} \sum_{q=0}^{N-1} x_1(q)\, e^{-j\frac{2\pi qm}{N}}}$$

$x_1(q) = w((Q-q)\Delta_\tau)\, x((nS-Q+q)\Delta_\tau)$ for $0 \le q \le 2Q$,

$x_1(q) = 0$ for $2Q < q < N$.

不同一: $n+k \Rightarrow nS+k$

不同二: $2\pi(Q-n) \Rightarrow 2\pi(Q-nS)$

比較 page 120

If $w(t) = w(-t)$)

$-Q \le k \le Q$

$x_1(q) = w(k\Delta_\tau)\, x((nS+k)\Delta_\tau)$ for $0 \le q \le 2Q$, $k = q-Q$

$x_1(q) = 0$ for $2Q < q < N$.

假設 $t = c_0\Delta_t,\ (c_0+1)\,\Delta_t,\ (c_0+2)\,\Delta_t,\ \cdots\cdots,\ (c_0+C-1)\,\Delta_t$

$= c_0 S\Delta_\tau,\ (c_0S+S)\,\Delta_\tau,\ (c_0S+2S)\,\Delta_\tau,\ \cdots\cdots,\ [c_0S+(C-1)S]\,\Delta_\tau,$

$f = m_0\,\Delta_f,\ (m_0+1)\,\Delta_f,\ (m_0+2)\,\Delta_f,\ \cdots\cdots,\ (m_0+F-1)\,\Delta_f$

$\tau = n_0\,\Delta_\tau,\ (n_0+1)\,\Delta_\tau,\ (n_0+2)\,\Delta_\tau,\ \cdots\cdots,\ (n_0+T-1)\,\Delta_\tau,$ $\qquad S = \Delta_t/\,\Delta_\tau$

Step 1: Calculate c_0, m_0, n_0, C, F, T, N, Q

Step 2: $n = c_0$

Step 3: Determine $x_1(q)$

Step 4: $X_1(m) = \mathrm{FFT}[x_1(q)]$

Step 5: Convert $X_1(m)$ into $X(\,n\Delta_t,\ m\Delta_f)$

Step 6: Set $n = n+1$ and return to Step 3 until $n = c_0+C-1$.

Complexity = ?

IV-F Non-Uniform Δ_t

(A) 先用較大的 Δ_t

(B) 如果發現 $\left|X\left(n\Delta_t, m\Delta_f\right)\right|$ 和 $\left|X\left((n+1)\Delta_t, m\Delta_f\right)\right|$ 之間有很大的差異

則在 $n\Delta_t$, $(n+1)\,\Delta_t$ 之間選用較小的 sampling interval Δ_{t1}

($\Delta_\tau < \Delta_{t1} < \Delta_t$, Δ_t / Δ_{t1} 和 $\Delta_{t1} / \Delta_\tau$ 皆為整數)

再用 page 131 的方法算出

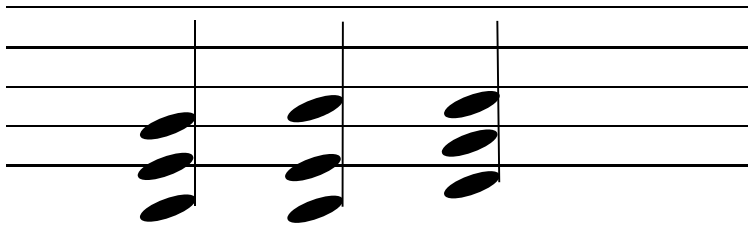
$$X\left(n\Delta_t + \Delta_{t1}, m\Delta_f\right), \quad X\left(n\Delta_t + 2\Delta_{t1}, m\Delta_f\right), \quad \cdots\cdots, \quad X\left((n+1)\Delta_t - \Delta_{t1}, m\Delta_f\right)$$

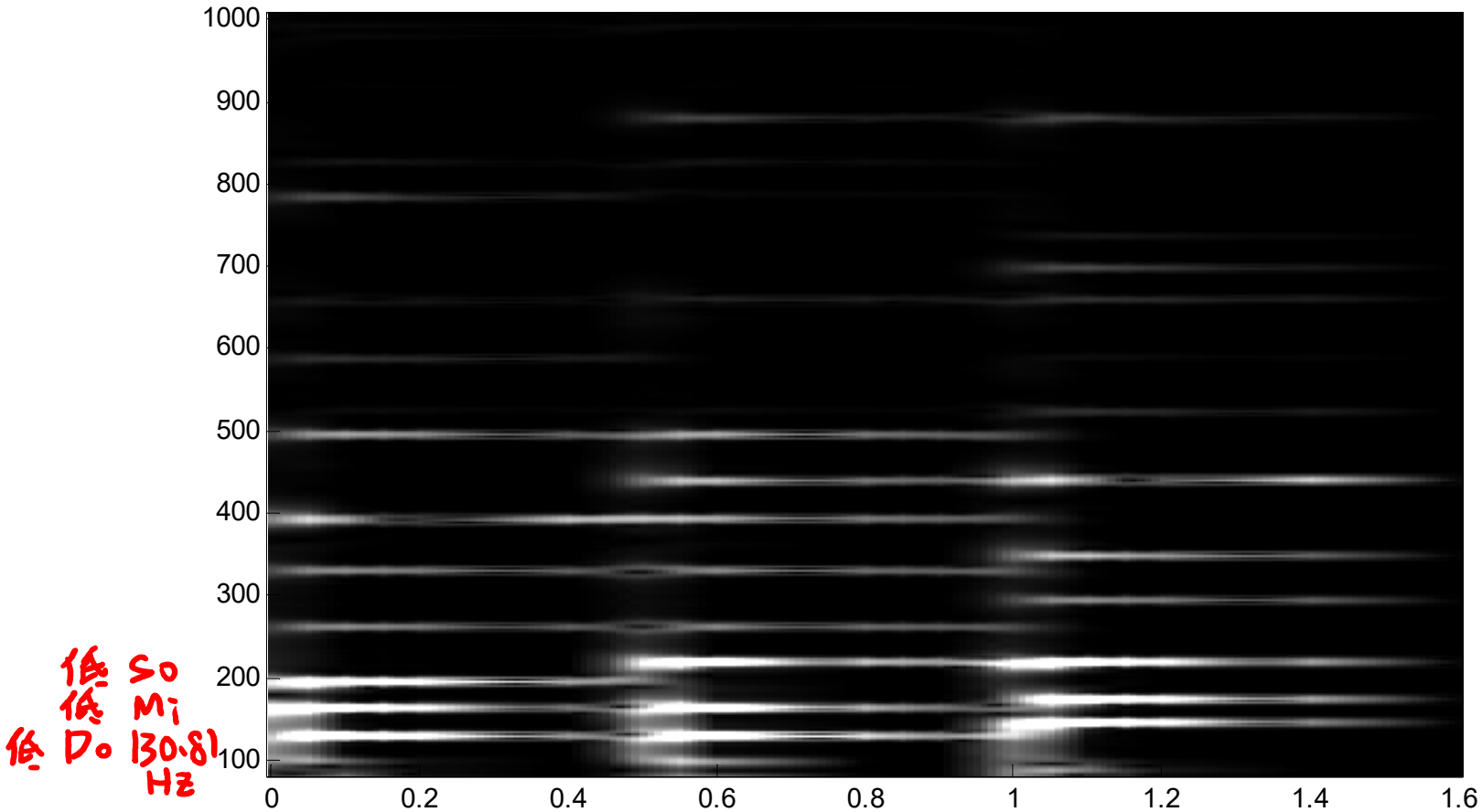
(C) 以此類推，如果 $\left|X\left(n\Delta_t + k\Delta_{t1}, m\Delta_f\right)\right|$, $\left|X\left(n\Delta_t + (k+1)\Delta_{t1}, m\Delta_f\right)\right|$

的差距還是太大，則再選用更小的 sampling interval Δ_{t2}

($\Delta\tau < \Delta t_2 < \Delta t_1$, $\Delta t_1 / \Delta t_2$ 和 $\Delta t_2 / \Delta\tau$ 皆為整數)

Gabor transform of a music signal





$\Delta_\tau = 1/44100$ (總共有 44100×1.6077 sec $+ 1 = 70902$ 點)

(A) Choose $\Delta_t = \Delta_\tau$

running time = out of memory (2008年) 15.262140 sec (2022年)

(B) Choose $\Delta_t = 0.01 = 441\Delta_\tau$ (1.6/0.01 + 1 = 161 points)

running time = 1.0940 sec (2008年) 0.041053 sec (2022年)

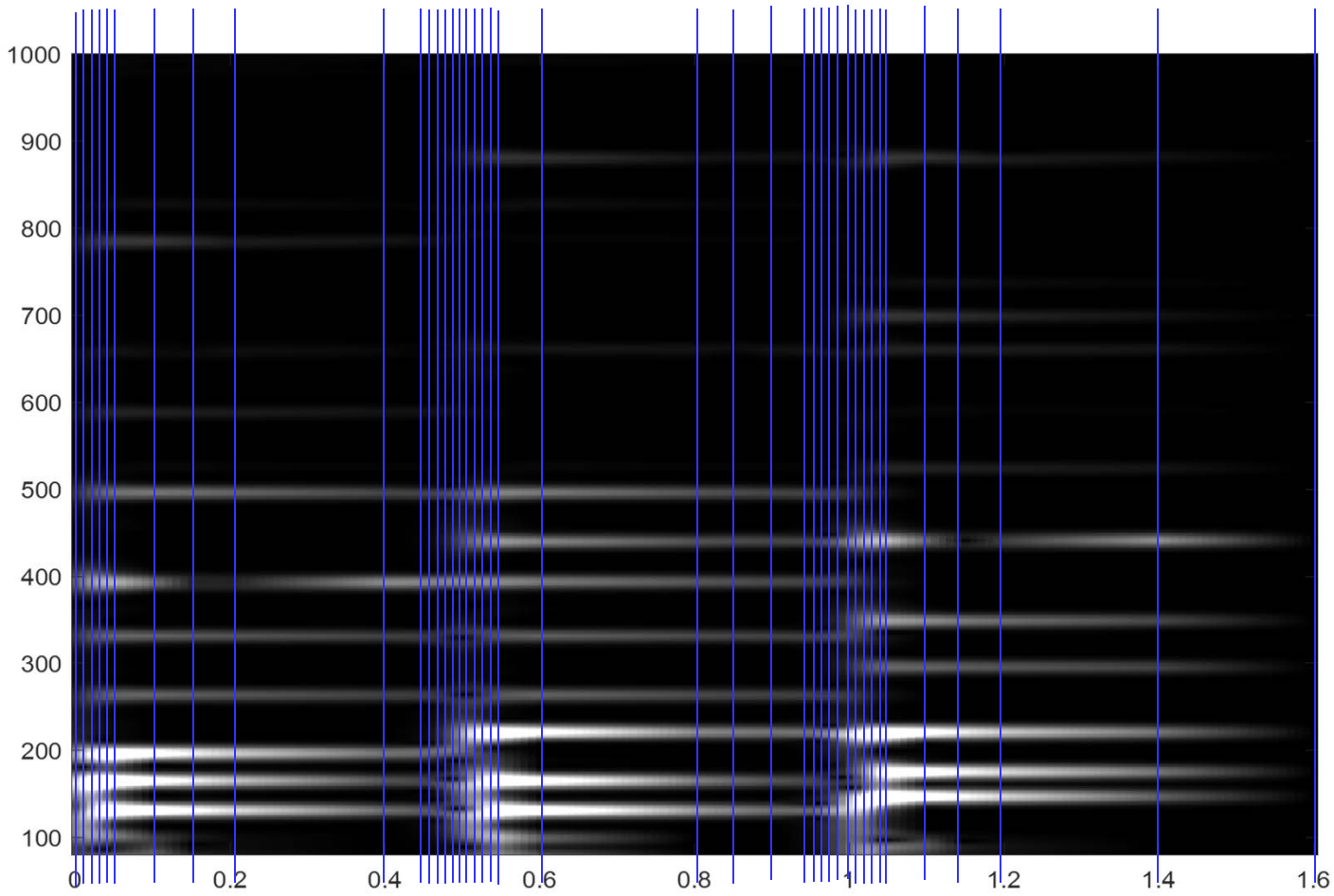
(C) Choose the sampling points on the t-axis as nonuniform

t = 0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.15, 0.2, 0.4, 0.45, 046, 0.47, 0.48, 0.49, 0.5, 0.51, 0.52, 0.53, 0.54, 0.55, 0.6, 0.8, 0.85, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 1, 1.01, 1.02, 1.03, 1.04, 1.05, 1.1, 1.15, 1.2, 1.4, 1.6

(41 points) $\Delta t = 0.2 \rightarrow 0.05 \rightarrow 0.01$

running time = 0.2970 sec (2008年) 0.010594 sec (2022年)

with adaptive output sampling intervals

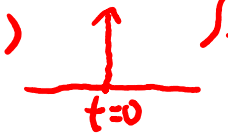


附錄七 和 Dirac Delta Function 相關的常用公式

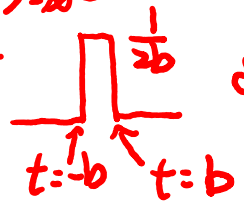
FT(1)

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

$\delta(t)$



$\int_{-\infty}^{\infty} \delta(t)\,dt = 1$

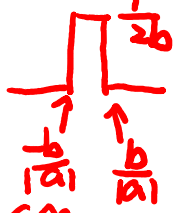


$\delta(t)$

$b \to 0$

(2) $\delta(t) = |a|\,\delta(at)$ (scaling property)

$\delta(at)$



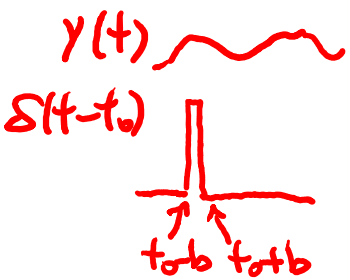
$\int_{-\infty}^{\infty} \delta(at)\,dt = \frac{1}{|a|}$

(3) $\int_{-\infty}^{\infty} e^{-j2\pi t g(f)} dt = \delta(g(f)) = \sum_n |g'(f_n)|^{-1} \delta(f - f_n)$

where f_n are the zeros of $g(f)$

(4) $\int_{-\infty}^{\infty} \delta(t - t_0)\, y(t, \cdots\cdots)\, dt = y(t_0, \cdots\cdots)$ (sifting property I)

(5) $\delta(t - t_0)\, y(t, \cdots\cdots) = \delta(t - t_0)\, y(t_0, \cdots\cdots)$ (sifting property II)



$y(t)\delta(t-t_0)$

$= y(t_0)\delta(t-t_0)$

when $b \to 0$