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

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

參考資料

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

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

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

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

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

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

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

For the alternative definition, the inverse transform is:

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

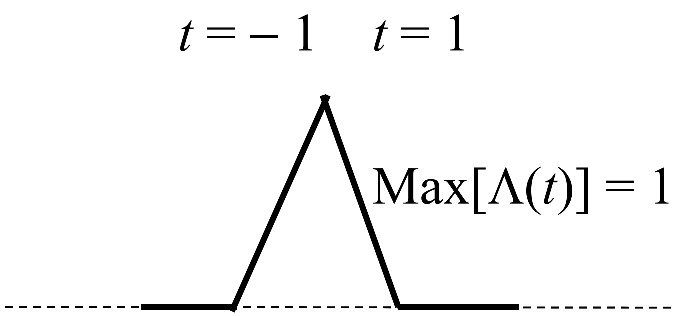
The mask function $w(t)$ always has the property of

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

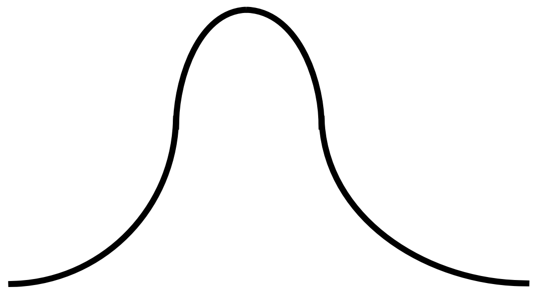
(b) $\max(w(t)) = w(0)$, $w(t_1) \geq w(t_2)$ if $|t_2| > |t_1|$

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

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



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

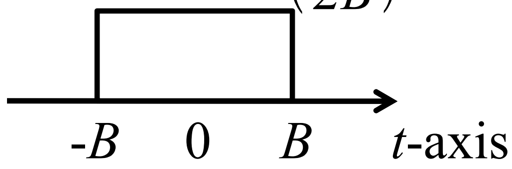


II-B Rec-STFT

Rectangular mask STFT (rec-STFT)

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

$$w(t)=rect\left(\frac{1}{2B}\right)$$



Inverse of the rec-STFT

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

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

The simplest form of the STFT

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

II-C Properties of the Rec-STFT

(1) Integration (recovery):

(a)

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

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

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

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

$= 0$ otherwise

(2) Shifting property (橫的方向移動)

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

(3) Modulation property (縱的方向移動)

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

(4) Special inputs:

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

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

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

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

思考： ***B*** 值的大小，對解析度的影響是什麼？

(5) Linearity property

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

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

(6) Power integration property

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

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

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

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

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

思考:

(1) 哪些性質 Fourier transform 也有？

(2) 其他型態的 STFT 是否有類似的性質？

Shifting

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

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

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

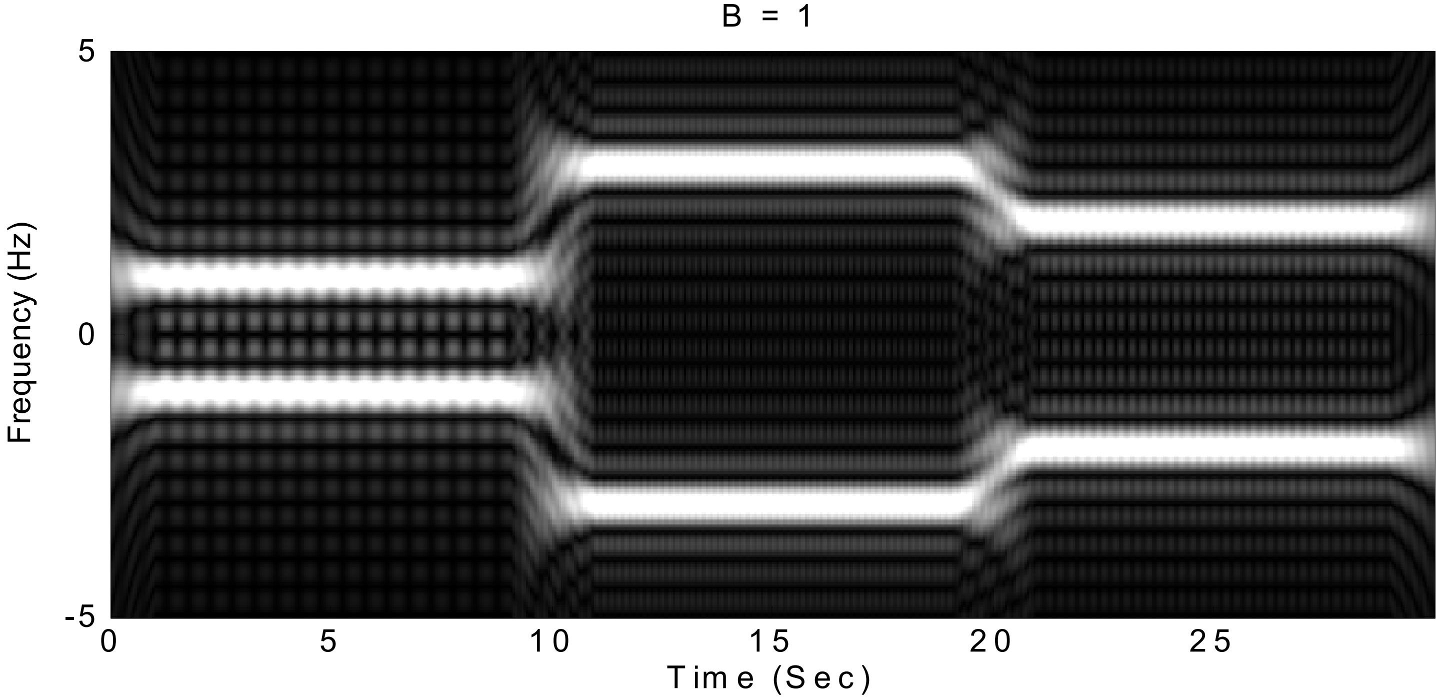
Modulation

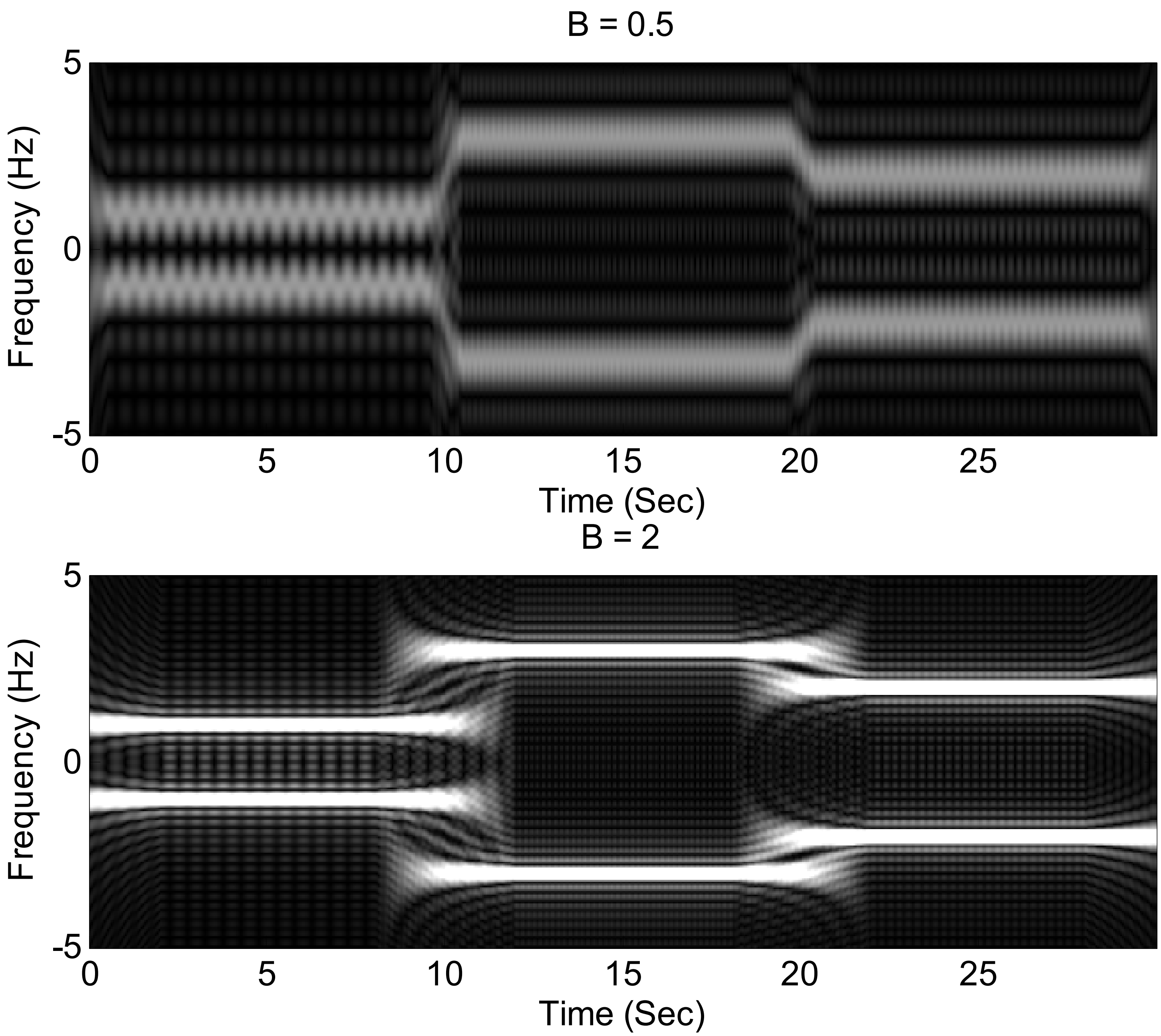
$$\int_{-\infty}^{\infty} w(t-\tau)[x(\tau)e^{j2\pi f_0\tau}]e^{-j2\pi f\tau}d\tau = X(t,f-f_0)$$

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

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

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




B = 0.5
Frequency (Hz)
5
0
-5
0
5
10
15
20
25
Time (Sec)
B = 2
Frequency (Hz)
5
0
-5
0
5
10
15
20
25
Time (Sec)

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

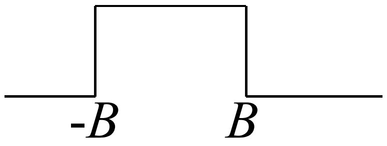
• Compared with other types of time-frequency analysis:

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

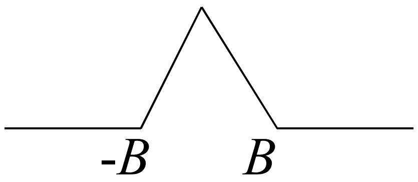
but its performance is worse than other types of time-frequency analysis.

II-E STFT with Other Windows

(1) Rectangle



(2) Triangle



(3) Hanning

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

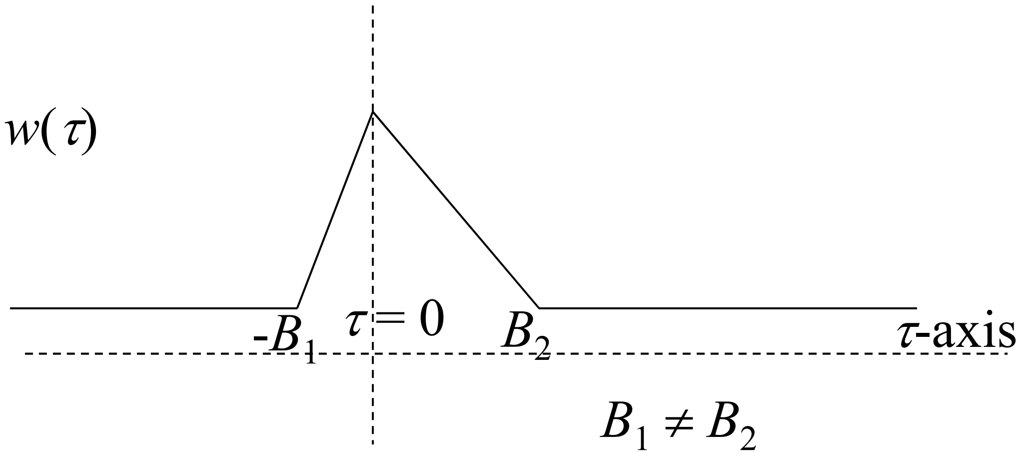
(4) Hamming

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

(5) Gaussian

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

(6) Asymmetric window



$w(t-\tau) = ?$

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

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

比較： spectrum 為 Fourier transform 的絕對值平方

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

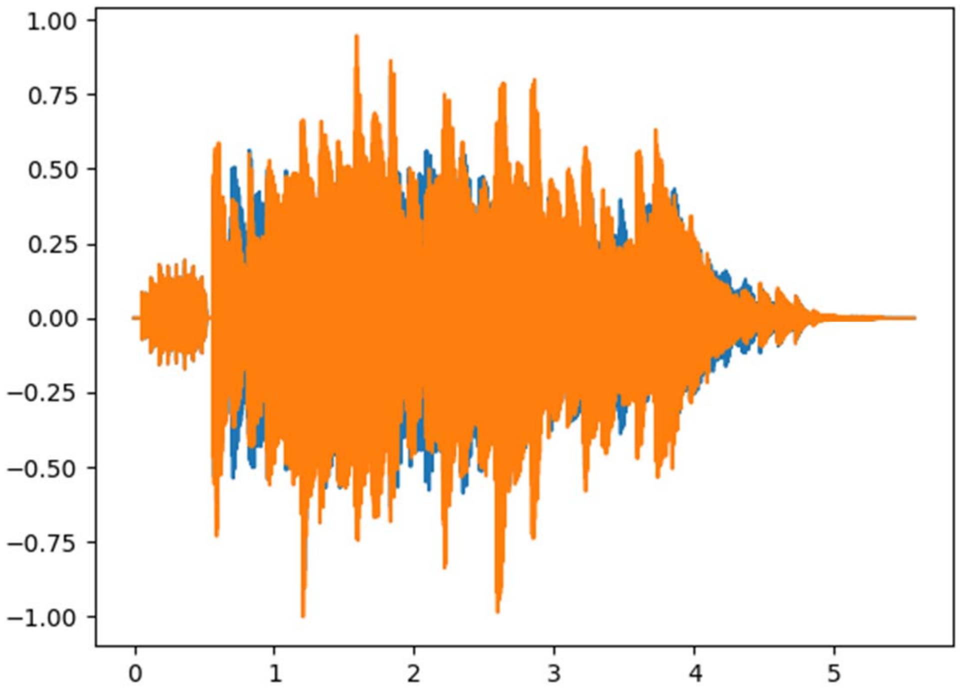
要先import 相關模組： import numpy as np

- 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

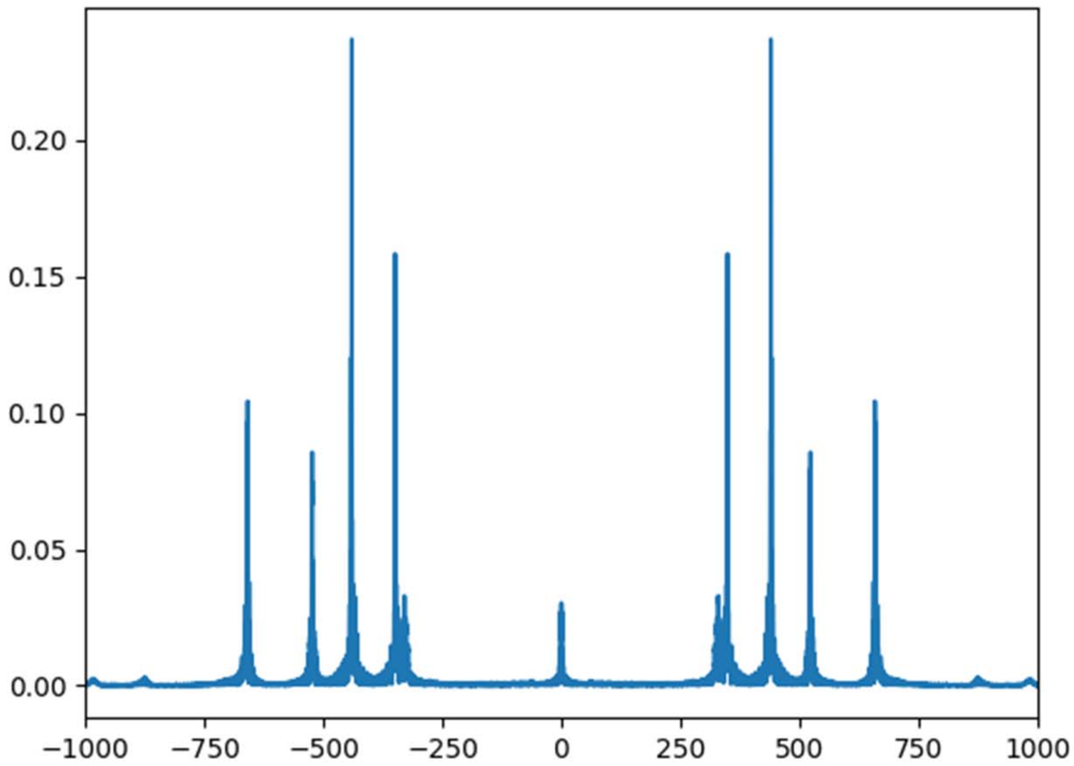
要先import 相關模組： from scipy.fftpack import fft

- 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


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

C. 播放聲音

要先import 相關模組： import simpleaudio as sa

- n_bytes =2 # using two bytes to record a data
- wave_data = (2**15-1)* wave_data
  # change the range to $-2^{15} \sim 2^{15}$
- wave_data = wave_data.astype(np.int16)
- play_obj = sa.play_buffer(wave_data, n_channel, n_bytes, fs)
- play_obj.wait_done()

D. 製作音檔

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

E. 錄音

要先import 相關模組： import pyaudio

範例程式

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

vocal=[]
count=0
```

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

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

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

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