多媒體程式設計
音訊資料處理

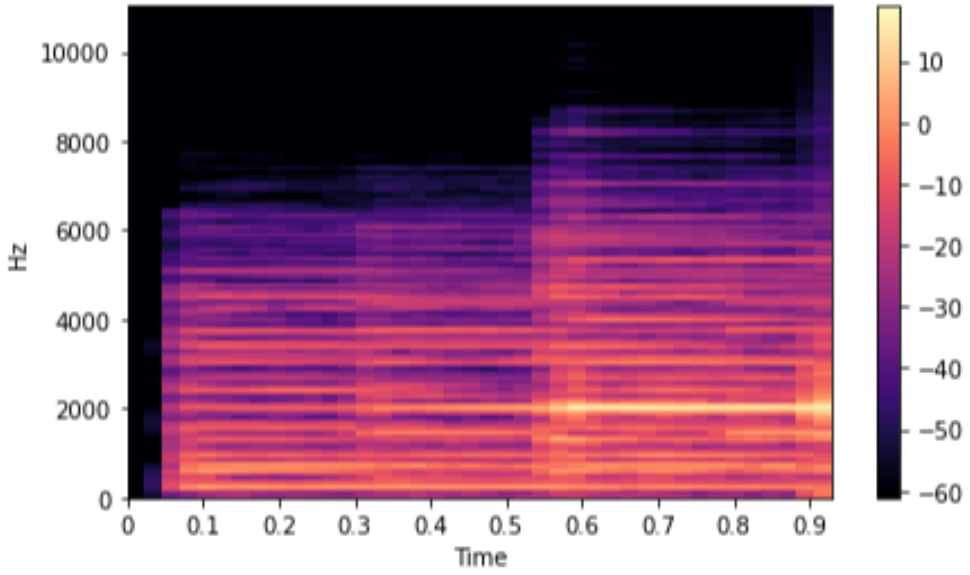
Instructor: 馬豪尚

音訊中的雜訊(噪音)

- 音訊中的雜訊(噪音)雖作為一個隨機訊號，仍然具有統計學上的特徵屬性
- 頻譜圖上的波型和能量分布即是雜訊的特徵之一，我們可以透過它來區分不同類型的雜訊
- 不同的雜訊可能會需要不同的處理方式
  - 疊加在時域上的噪音
    - 高頻與低頻噪音
    - 穩定/不穩定噪音
  - 疊加在頻域上的噪音
    - 混疊卷積噪音

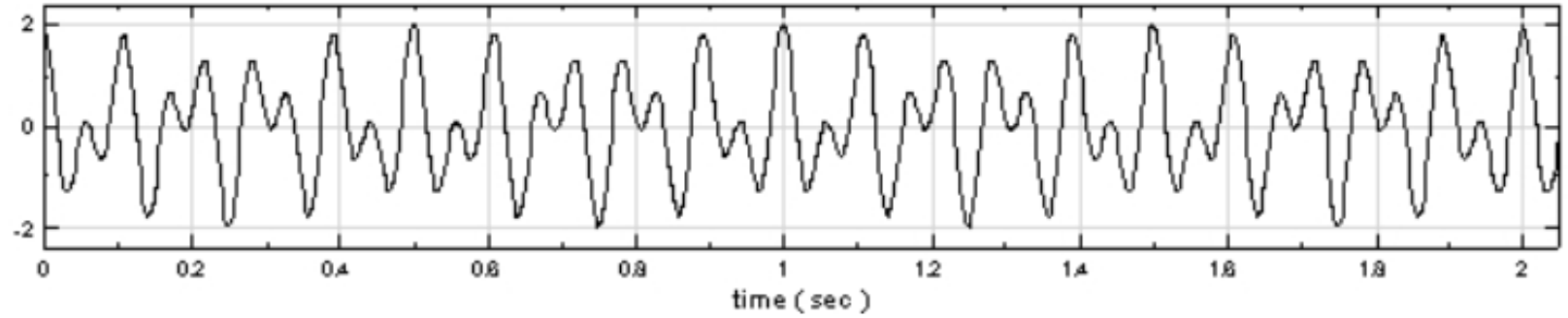
高頻與低頻噪音

- 高頻噪音通常指的是超過某個頻率以上的雜訊
- 低頻噪音通常指的是低於某個頻率以上的雜訊
- 這兩種雜訊大部分不是人類關注的聲音
- 我們可以透過濾波器來濾除高頻或低頻雜訊

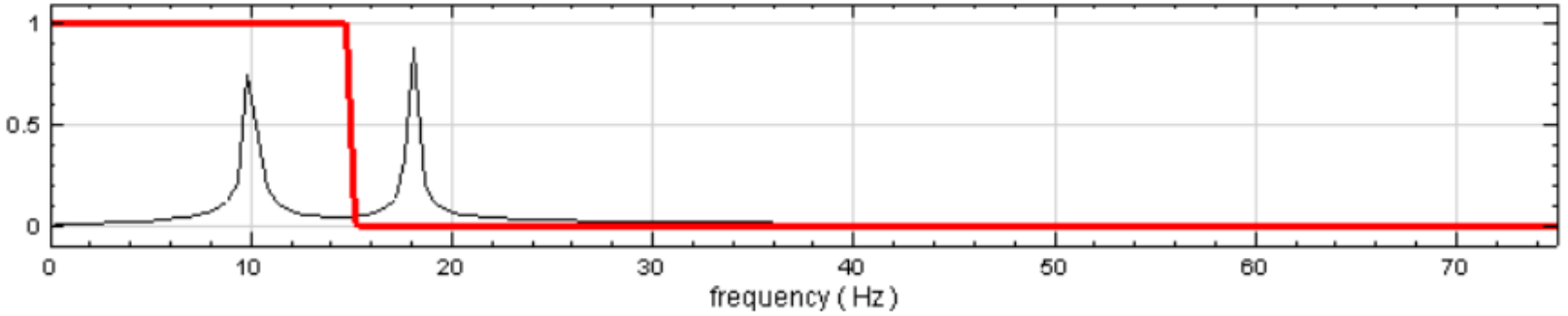


低通濾波器降低噪音

- 音訊波形(空間域)

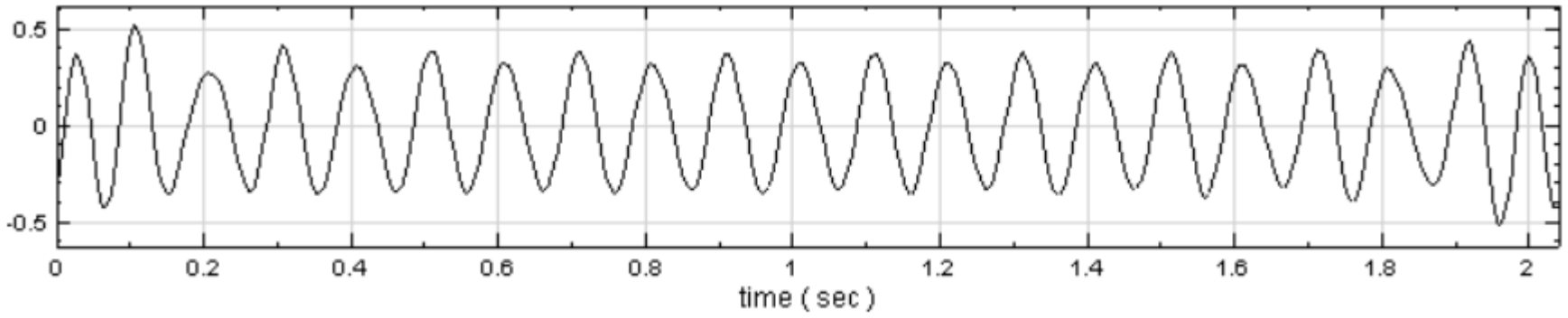


- 傅立葉轉換後的頻譜(頻率域)
  - 在傅立葉空間定義出上述紅線的函數
  - 並將此函數與原本訊號的傅立葉轉換相乘



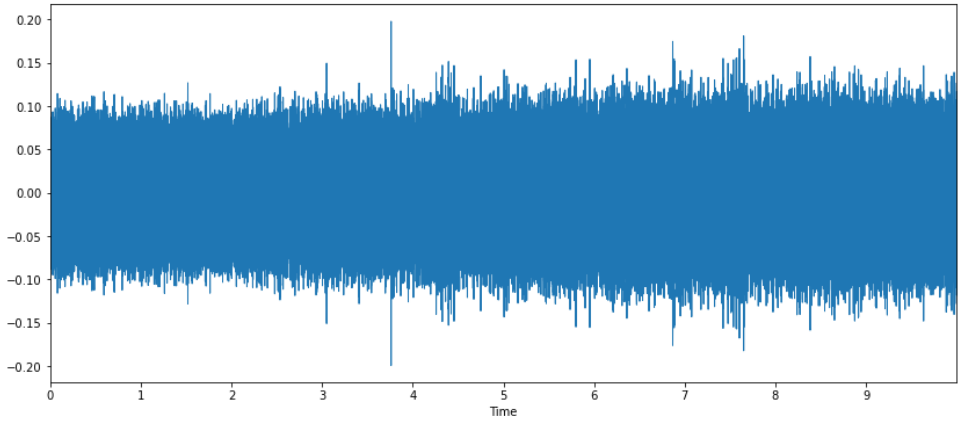
低通濾波器降低噪音

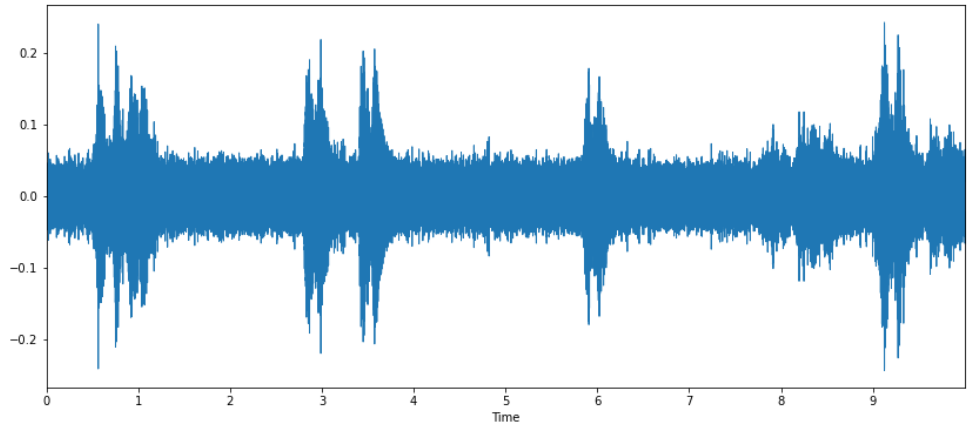
› 再做傅立葉逆轉換，就可得到濾除高頻部份的圖型



高頻或低頻噪音應用

- 應用情境
  - 假設原始音訊和噪音的頻率相差比較大
- 時域上的特徵為震幅
  - 情況1: 因為噪音的震幅比較大
  - 情況2: 噪音的震幅比原始音訊小

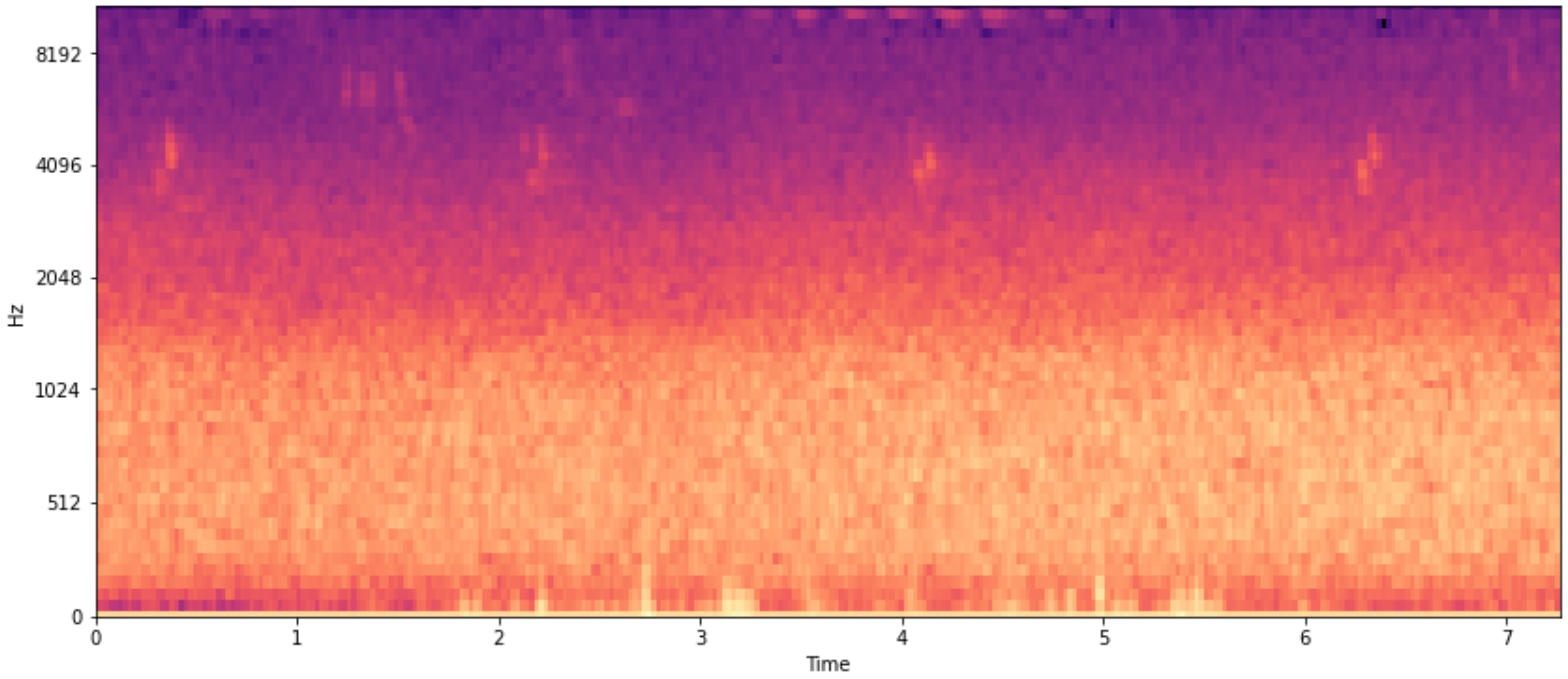




轉換到頻譜圖上

- 頻率域上的特徵為頻率與能量
- 將原始音訊轉換到頻率域上可觀察到頻率與能量的分布
  - 低頻的部分能量非常大

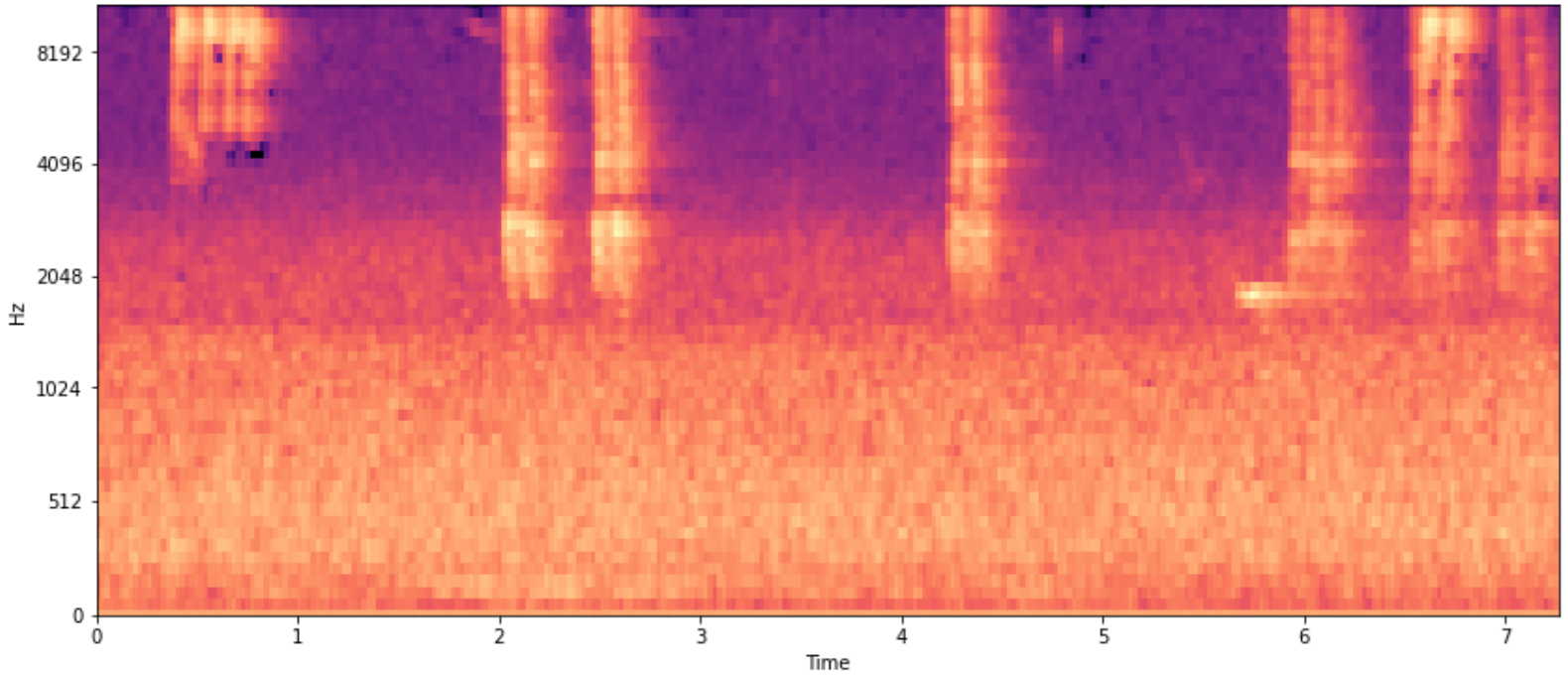
情況1的頻譜圖



轉換到頻譜圖上

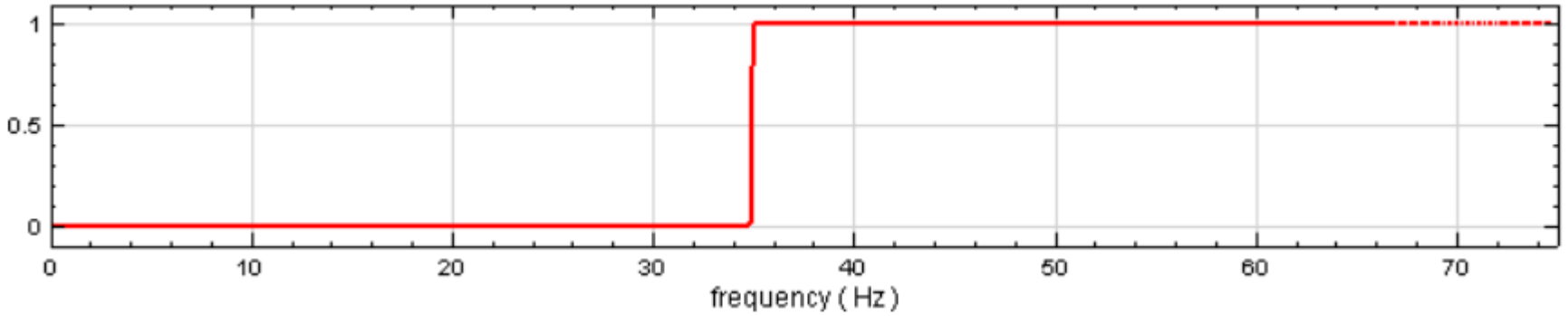
- 將原始音訊轉換到頻率域上可觀察到頻率與能量的分布
  - 低頻的部分能量雖然大，但高頻的部分有些地方的能量更大(我們想要的音訊)

情況2的頻譜圖



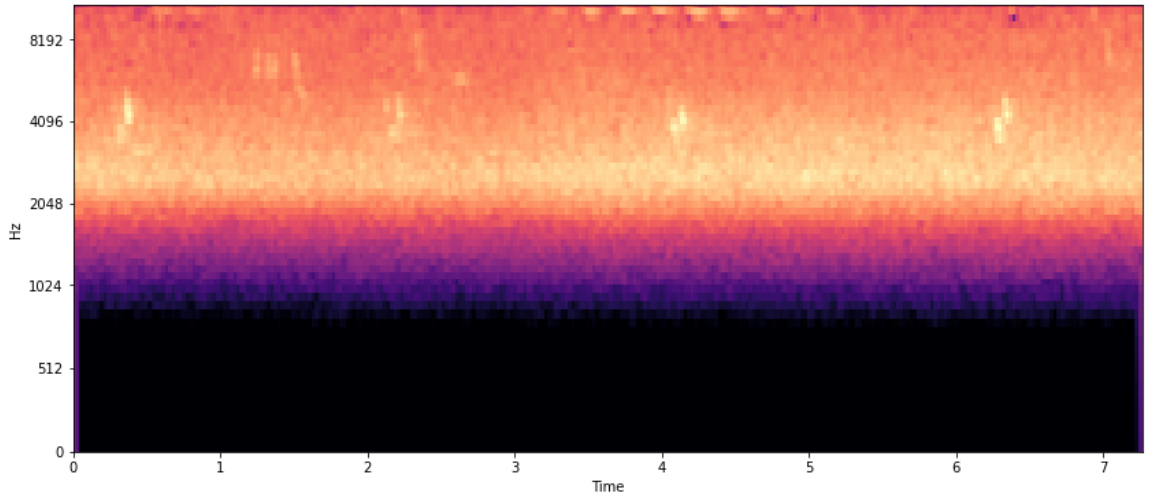
過濾掉低頻的音訊

- 制定一個高通濾波器
- 將頻譜乘上這個高通濾波器

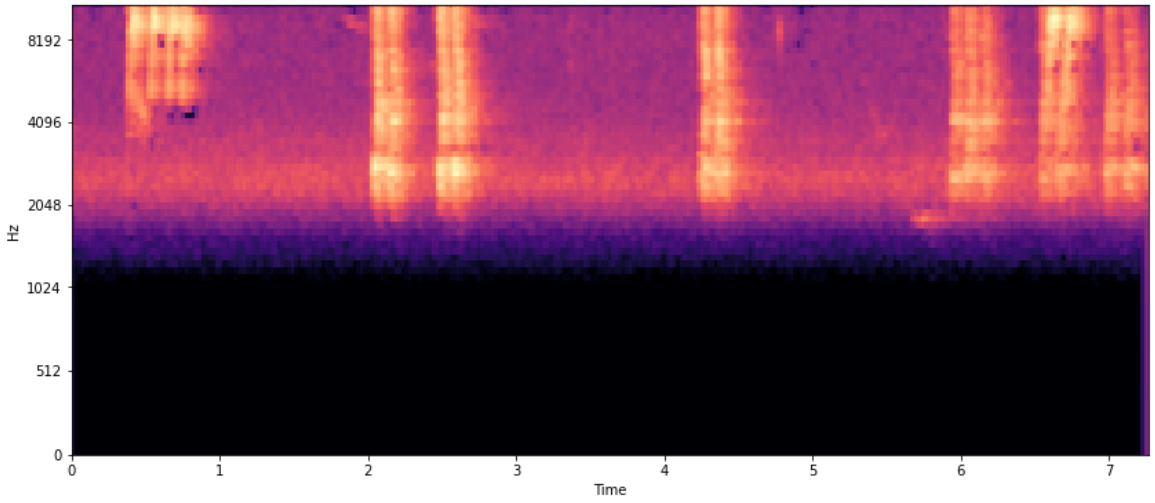


過濾後的頻譜圖

情況1的頻譜圖

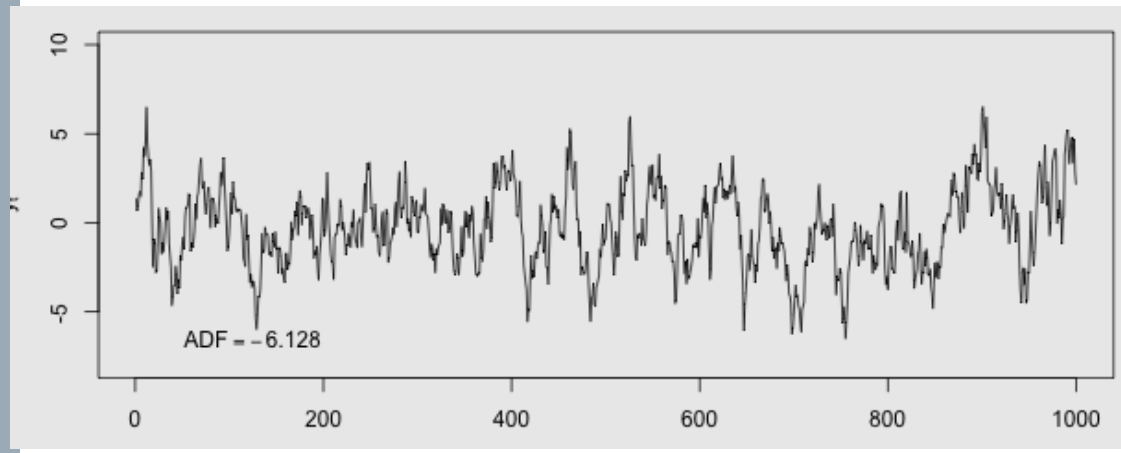


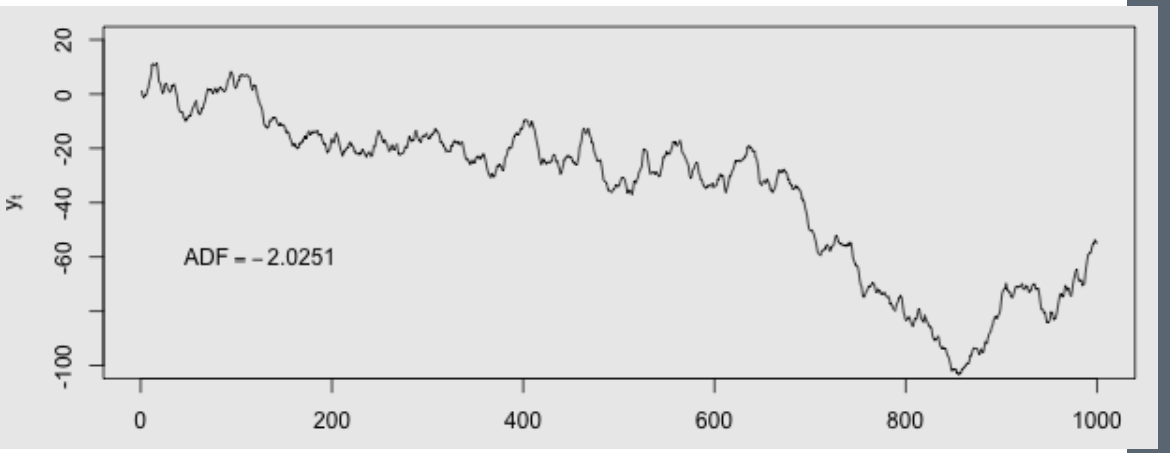
情況2的頻譜圖



穩定與不穩定噪音

- 雜訊的頻譜是否隨著時間改變
  - 不改變為穩定噪音
    - 冷氣、汽車等機械聲音
  - 會隨著時間改變為不穩定噪音
    - 多種噪音混雜或音樂噪音





穩定噪音的降噪方法

- 譜減法(Spectral Subtraction)
  - 假設語音中的雜訊只有疊加噪音(把雜訊當作與正常訊號沒有關聯)
  - 帶噪訊號的頻譜減去噪音訊號的頻譜

$$|X(\omega)|^2 = |Y(\omega)|^2 - |D(\omega)|^2$$

  - 關鍵在於對噪音頻譜的估計
    - 基於頻譜在短時間內是平穩不變的特性來估計
    - 一般認為在一段音訊中的前幾個frame是沒有正常訊號

穩定噪音的降噪方法

- 譜減法實作過程
  - 根據語音訊號前幾個frame計算平均噪音譜作為估計噪音譜
  - 對語音進行切分frame，對每一個frame都做傅立葉轉換
  - 使用帶噪的訊號譜減去估計噪音譜
  - 如果出現負值，將其設定為0
  - 對減去噪音譜後的訊號進行傅立葉反轉換得到時域訊號
  - 根據frame的長度和hop length重組時域訊號

穩定噪音的降噪方法

- Berouti譜減法
  - 減小音樂噪聲的方法是對噪聲譜使用過減技術
  - 同時對譜減後的負值設定一個下限，不是將它們設為0

$$|X(\omega)|^2 \begin{cases} |Y(\omega)|^2 - \alpha |D(\omega)|^2, |Y(\omega)|^2 - |D(\omega)|^2 > 0 \\ \beta |D(\omega)|^2, else \end{cases}$$

  - alpha（大於等於1）為過減因子，它主要影響語音譜的失真程度
  - beta（大於0小於1）是譜下限參數，可以控制殘留噪音的多少以及音樂噪音的大小
  - beta為一個固定值，而alpha根據當前frame的訊噪比進行計算

穩定噪音的降噪方法

- Berouti譜減法實作過程
  - 根據語音訊號前幾個frame計算平均噪音譜作為估計噪音譜
  - 對語音進行切分frame，對每一個frame都做傅立葉轉換
  - 計算當前frame的訊噪比，然後求出過減因子alpha
  - 使用帶噪的訊號譜減去估計噪音譜
  - 如果出現負值，將其設定為 $\beta|D(\omega)|^2$
  - 對減去噪音譜後的訊號進行傅立葉反轉換得到時域訊號
  - 根據frame的長度和hop length重組時域訊號

Scipy

- SciPy是一個開源的Python演算法庫和數學工具
- 包含的模組有最佳化、線性代數、積分、插值、特殊函數、快速傅立葉轉換、訊號處理和圖像處理、常微分方程式
- SciPy的基礎資料結構是由NumPy模組提供的多維陣列
- 安裝套件
  - pip install scipy

Scipy 讀取音訊

- 載入scipy讀取模組
  - import scipy.io.wavfile
- 讀取wav
  - scipy.io.wavfile.read('filename')
    - filename為檔案名稱
    - 會回傳兩個值
      - 音訊的sample rate
      - 音訊本體資料(numpy array)

Scipy 讀取音訊

- 儲存音訊的Numpy array為ndarray型態
- 根據音訊本身的位元深度來決定numpy array裡面儲存的資料型態為int16、int32、float32等
- 第一個維度的shape表示音訊的取樣點數
  - data.shape[0]
- 第二個維度的shpae表示音訊的聲道數
  - data.shape[1]
- 元素的值代表取樣點的振幅
  - 經過量化後的值

Librosa讀取音訊

- 儲存音訊的Numpy array為ndarray型態
- ndarray維度為(n,) 或 (…, n)
- 元素的值代表取樣點的振幅
  - 經過量化後的值

data單聲道(n,)

data雙聲道 (…, n)

Scipy 寫入音訊

- 寫入WAV
  - scipy.io.wavfile.write(filename, rate, data)
    - filename為要寫入的檔案名稱
    - rate為音訊的samplerate
    - data為音訊資料
      - 資料格式為numpy ndarray(1D/2D)
      - array元素的資料型態可為float32、int16、int32、unit8等

濾波器降噪

- 讀取音訊
  - y, sr = librosa.load(filename, mono=True, sr=sr)
- 將音訊切成frame並做傅立葉轉換
  - dataFFT = librosa.stft(data, n_fft=2048)
- 制定一個濾波器
  - 高通濾波器
  - 低通濾波器
- 將訊號通過濾波器做降噪

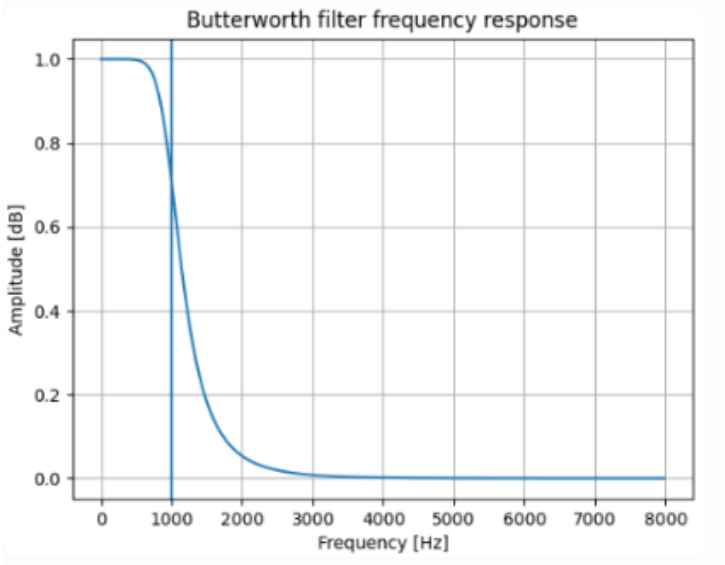
Scipy.signal模組制定濾波器

- b, a = signal.butter(N, Wn, btype='low', analog=False)
  - N：濾波器的階數，越高階會讓頻率衰減越快
  - Wn：臨界頻率
    - 對於低通和高通濾波器，Wn 是一個頻率值
    - 對於帶通(bandpass)和帶阻(bandstop)濾波器，Wn 是一個長度為 2 的序列
  - Btype：濾波器的類型
    {'lowpass'，'highpass'，'bandpass'，'bandstop'}
  - analog：如果為True，則返回模擬濾波器，否則返回數字濾波器

- 回傳兩個值為濾波器的參數

Scipy.signal模組制定濾波器

- 低通濾波器
  - b, a = signal.butter(4, 1000, btype='lowpass', analog=False)



- 高通濾波器
  - b, a = signal.butter(4, 3000, btype='highpass', analog=False)

Scipy.signal 濾波函數

- scipy.signal.filtfilt(b, a, data)
  - 輸入濾波器參數b和a
  - data為輸入帶有噪音的音訊
  - 輸出經過濾波器降噪後的音訊

譜減法實作

- 根據語音訊號前幾個frame計算平均噪音譜作為估計噪音譜
- 對語音進行切分frame，對每一個frame都做傅立葉轉換
- 使用帶噪的訊號譜減去估計噪音譜
- 如果出現負值，將其設定為0
- 對減去噪音譜後的訊號進行傅立葉反轉換得到時域訊號
- 根據frame的長度和hop length重組時域訊號