

## 3-1 Introduction to Audio Signals (音訊基本介紹)

[[english](#)][[all](#)]

(請注意：中文版本並未隨英文版本同步更新！)

### Slides

所謂「聲音訊號」(Audio Signals)簡稱「音訊」，泛指由人耳聽到的各種聲音的訊號。一般來說，發音體會產生震動，此震動會對空氣產生壓縮與伸張的效果，形成聲波，以每秒大約 340 公尺的速度在空氣中傳播，當此聲波傳遞到人耳，耳膜會感覺到一伸一壓的壓力訊號，內耳神經再將此訊號傳遞到大腦，並由大腦解析與判讀，來分辨此訊號的意義。

音訊可以有很多不同的分類方式，例如，若以發音的來源，可以大概分類如下：

- 生物音：人聲、狗聲、貓聲等。
- 非生物音：引擎聲、關門聲、打雷聲、樂器聲等。

若以訊號的規律性，又可以分類如下：

- 準週期音：波形具有規律性，可以看出週期的重複性，人耳可以感覺其穩定音高的存在，例如單音絃樂器、人聲清唱等。
- 非週期音：波形不具規律性，看不出明顯的週期，人耳無法感覺出穩定音高的存在，例如打雷聲、拍手聲、敲鑼打鼓聲、人聲中的氣音等。

以人聲而言，我們可以根據其是否具有音高而分為兩類，如下：

- **Voiced sound**: 由聲帶振動所發出的聲音，例如一般的母音等。由於聲帶振動，造成規律性的變化，所以我們可以感覺到音高的存在。
- **Unvoiced sound**: 由嘴唇所發出的氣音，並不牽涉聲帶的震動。由於波形沒有規律性，所以我們通常無法感受到穩定音高的存在。

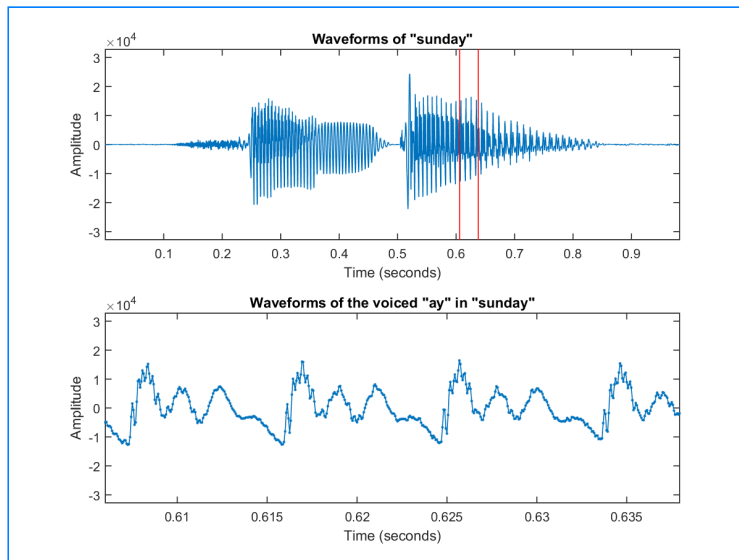
要分辨這兩種聲音，其實很簡單，你只要在發音時，將手按在喉嚨上，若有感到震動，就是 **voiced sound**，如果沒有感到震動，那就是 **unvoiced sound**。

下圖顯示在 "sunday" 發音中的 "ay" 部分波形，這是一個 **voiced sound**。

#### Example 1: [voicedFrame01.m](#)

```
figure;
waveFile='sunday.wav';
au=myAudioRead(waveFile); y=au.signal; fs=au.fs; nbits=au.nbits;
y=y*2^nbits/2;
subplot(2,1,1)
time=(1:length(y))/fs;
plot(time, y); axis([min(time), max(time), -2^nbits/2, 2^nbits/2]);
xlabel('Time (seconds)'); ylabel('Amplitude'); title('Waveforms of "sunday"');

frameSize=512;
index1=0.606*fs;
index2=index1+frameSize-1;
line(time(index1)*[1, 1], 2^nbits/2*[-1 1], 'color', 'r');
line(time(index2)*[1, 1], 2^nbits/2*[-1 1], 'color', 'r');
subplot(2,1,2);
time2=time(index1:index2);
y2=y(index1:index2);
plot(time2, y2, '-'); axis([min(time2), max(time2), -2^nbits/2, 2^nbits/2]);
xlabel('Time (seconds)'); ylabel('Amplitude'); title('Waveforms of the voiced "ay" in "sunday"');
```



你可以輕易地由目視來看出在放大波形中的基本頻率。

此外，你也可以觀察在發音 "sunday" 中的 unvoiced sound "s", 如以下範例所示：

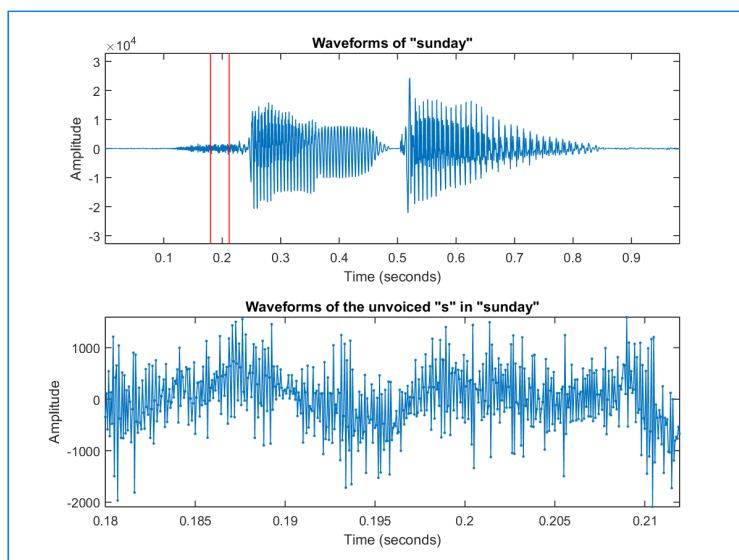
#### Example 2: [unvoicedFrame01.m](#)

```

waveFile='sunday.wav';
au=myAudioRead(waveFile); y=au.signal; fs=au.fs; nbits=au.nbits;
y=y*2^nbits/2;
subplot(2,1,1)
time=(1:length(y))/fs;
plot(time, y); axis([min(time), max(time), -2^nbits/2, 2^nbits/2]);
xlabel('Time (seconds)'); ylabel('Amplitude'); title('Waveforms of "sunday"');

frameSize=512;
index1=0.18*fs;
index2=index1+frameSize-1;
line(time(index1)*[1, 1], 2^nbits/2*[-1 1], 'color', 'r');
line(time(index2)*[1, 1], 2^nbits/2*[-1 1], 'color', 'r');
subplot(2,1,2);
time2=time(index1:index2);
y2=y(index1:index2);
plot(time2, y2, '-'); axis([min(time2), max(time2), -inf inf]);
xlabel('Time (seconds)'); ylabel('Amplitude'); title('Waveforms of the unvoiced "s" in "sunday"');

```



我們在其放大波形中並無法觀察到基本週期的存在，其波形比較像是雜訊，並無週期性。

#### Hint

You can also use CoolEdit for simple recording, replay and observation of audio signals.  
若要對聲音進行簡單的錄音、播放、觀察及處理，可以使用 CoolEdit 軟體。

聲音代表了空氣的密度隨時間的變化，基本上是一個連續的函數，但是若要將此訊號儲存在電腦裡，就必須先將此訊號數位化。一般而言，當我們將聲音儲存到電腦時，有下列幾個參數需要考慮：

- 取樣頻率（**sample Rate**）：每秒鐘所取得的聲音資料點數，以 **Hertz**（簡寫 **Hz**）為單位。點數越高，聲音品質越好，但是資料量越大，常用的取樣頻率如下：
  1. 8 kHz：電話的音質、一般玩具內語音 IC 的音質
  2. 16 KHz：一般語音辨識所採用
  3. 44.1 KHz：CD 音質
- 取樣解析度（**Bit Resolution**）：每個聲音資料點所用的位元數，常用的數值如下：
  1. 8-bit：可表示的數值範圍為 0~255 或 -128~127
  2. 16-bit：可表示的數值範圍為 -32768~32767換句話說，每個取樣點的數值都是整數，以方便儲存。但是在 **MATLAB** 的表示法，通常把音訊的值正規化到 **[-1, 1]** 範圍內的浮點數，因此若要轉回原先的整數值，就必須再乘上  $2^{nbits}/2$ ，其中  $nbits$  是取樣解析度。
- 聲道：一般只分單聲道（**Mono**）或立體聲（**Stereo**），立體音即是雙聲道。

以我所錄的「[sunday](#)」來說，這是單聲道的聲音，取樣頻率是 16000（16 KHz），解析度是 16 Bits（2 Byte），總共包含了 15716 點（等於  $15716/16000 = 0.98$  秒），所以檔案大小就是  $15716 \times 2 = 31432 \text{ bytes} = 31.4 \text{ KB}$  左右。由此可以看出聲音資料的龐大，例如：

- 如果我以相同的參數來進行錄音一分鐘，所得到的檔案大小大約就是 60 秒 x 16 KHz x 2 Byte = 1920 KB 或將近 2 MB。
- 以一般音樂 CD 來說，大部分是立體聲，取樣頻率是 44.1 KHz，解析度是 16 Bits，所以一首三分鐘的音樂，資料量的大小就是 180 秒 x 44.1 KHz x 2 Byte x 2 = 31752 KB = 32 MB。（由此可知，MP3 的壓縮率大概是10倍左右。）

---

### Audio Signal Processing and Recognition (音訊處理與辨識)