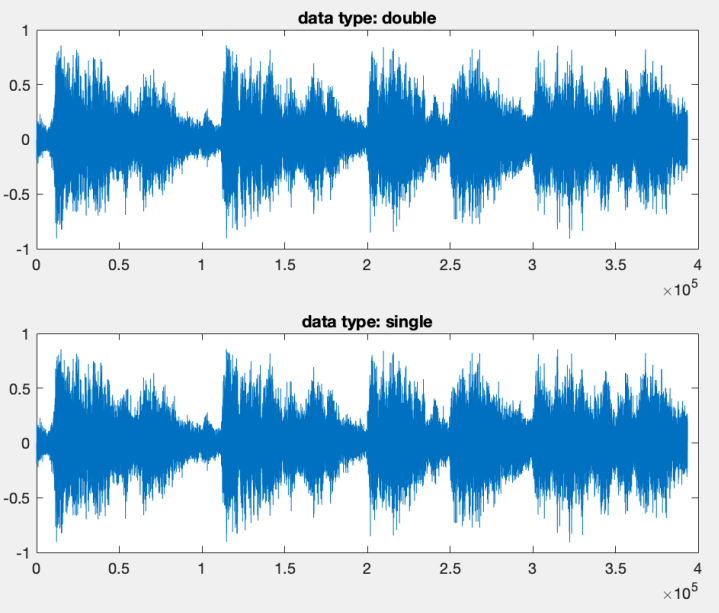
Lab 1: Audio Signal Processing using Matlab

1.

(a) 用原本的fs，聽起來就是正常的歌聲；用fs/2聽起來像是慢速播放，外加聲音低八度；用2*fs則是快速播放，外加聲音高八度。

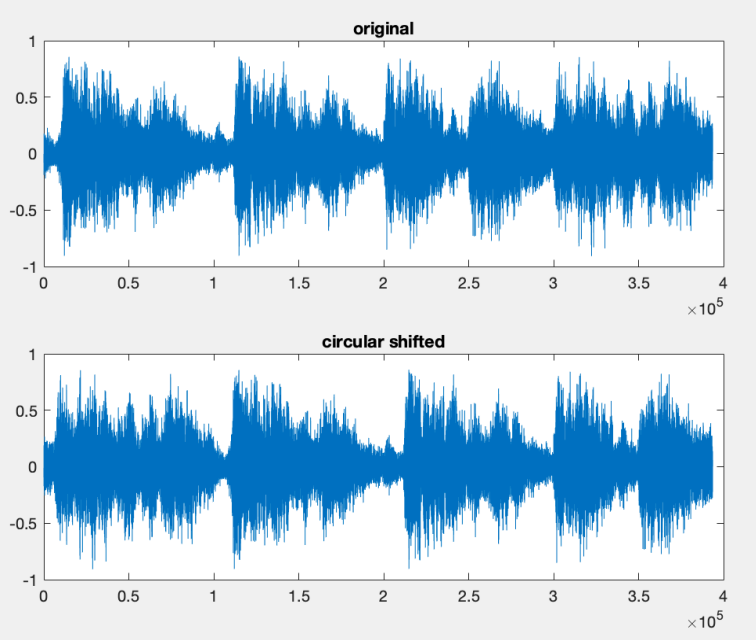
(b) 我用 y=(x, single) 把x轉成single（原本是double），再用sound(y, fs)播放，但我完全聽不出來差異...

(c)

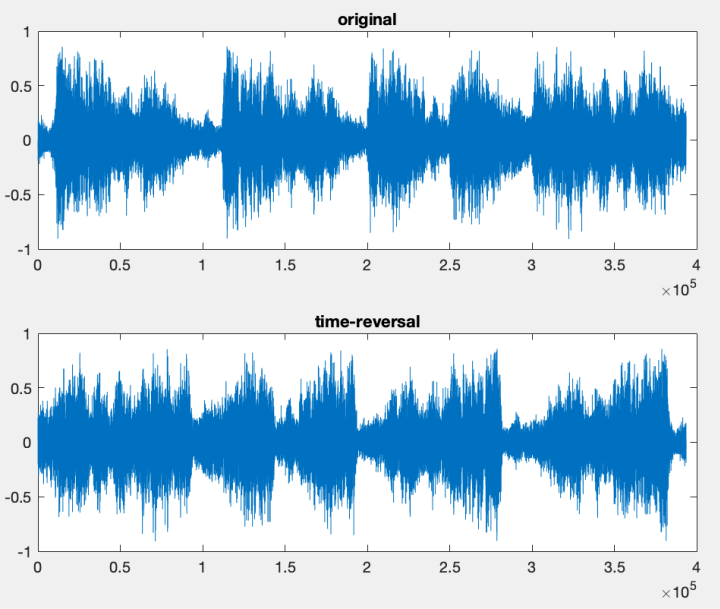


2.

(a)聲音聽起來一樣，只是順序調換了。



(b)聽起來就是音樂倒帶播放的聲音



(c)這樣用

```
clear x fs
[x, fs] = audioread('handel.ogg');
y = flipud(x);
size(x)
z = [x;y]
sound(z, fs)
audiowrite('output_file.wav', z, fs)
```

(d)這樣用

```
clear x fs
[x, fs] = audioread('handel.ogg');
x_up = upsample(x,2)
x_down = downsample(x,2)
sound(x_up, fs)
sound(x_down, fs)
```

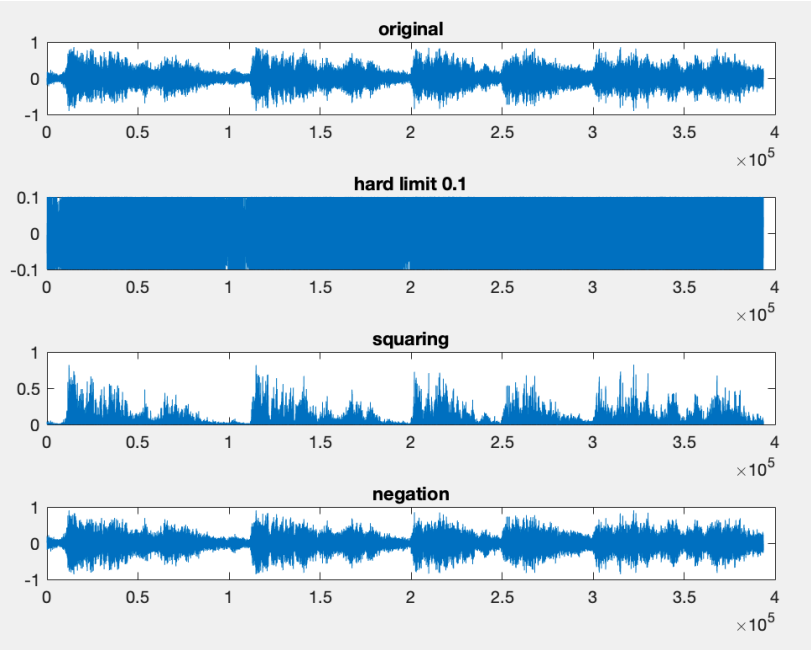
upsample聽起來跟1(a)裡用fs/2播放一樣，downsample聽起來跟1(a)裡用2*fs播放一樣。

3.

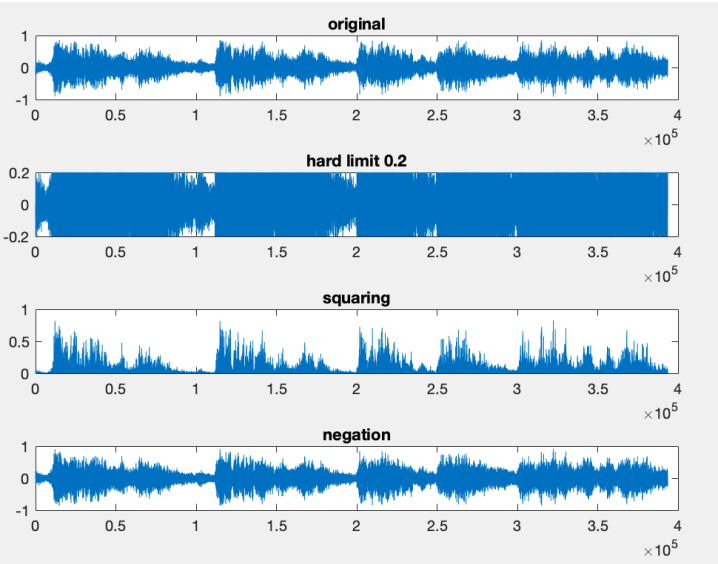
(a)hard limit聽起來像是喇叭開太大聲，聲音破掉，但值得注意的是，聲音大小聽起來差不多，因為大部分的數據點的絕對值都沒有超過0.1，都沒被truncade掉，因此聲波的總能量差不多。

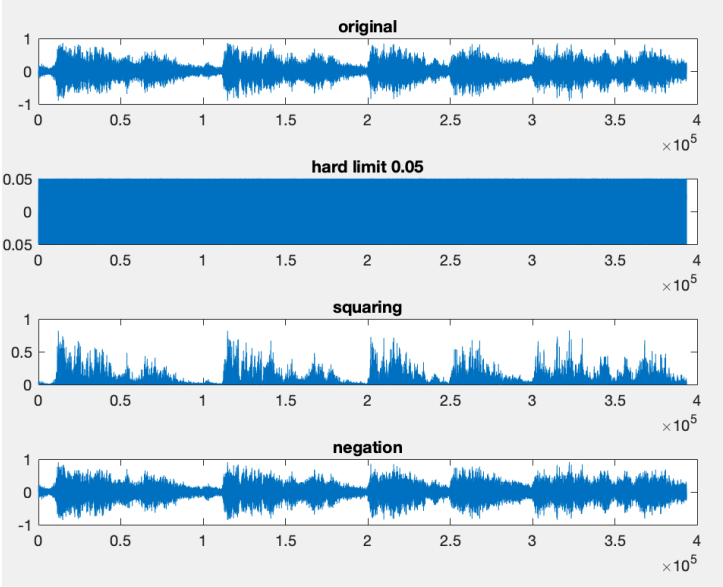
squaring聽起來很奇怪，免強聽得出音高和聲音強弱，但音色已經改變，聽不出字句。

neg聽起來完全一樣，因為加負號只是相位改變180度，而人耳聽不出聲波的相位。



(b)我把T改成0.2，聲音效果聽起來差不多，聽起來像開大聲因而破音，但沒有上一題嚴重。
T改成0.05，聲音效果也差不多，但更嚴重。





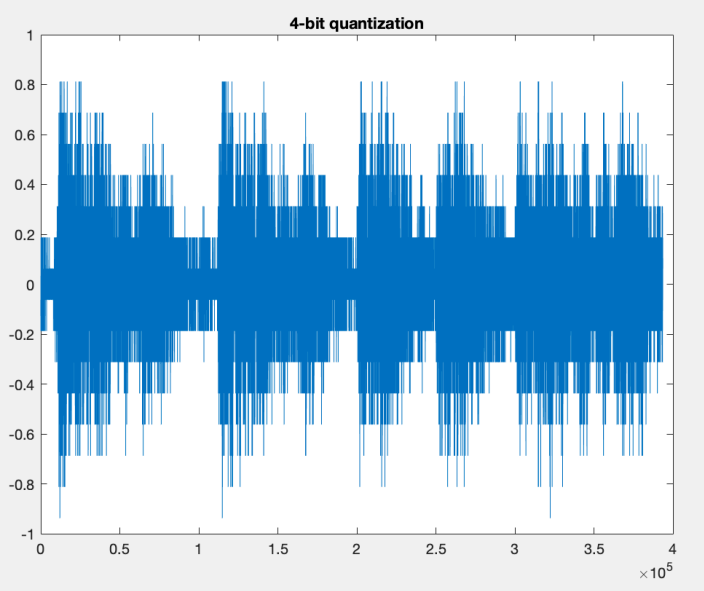
4.
(a)

$$d(u)=\left(\left\lfloor \frac{u}{\Delta} \right\rfloor+0.5\right)\cdot\Delta,\ \Delta=\frac{2x_{\max}}{L}$$

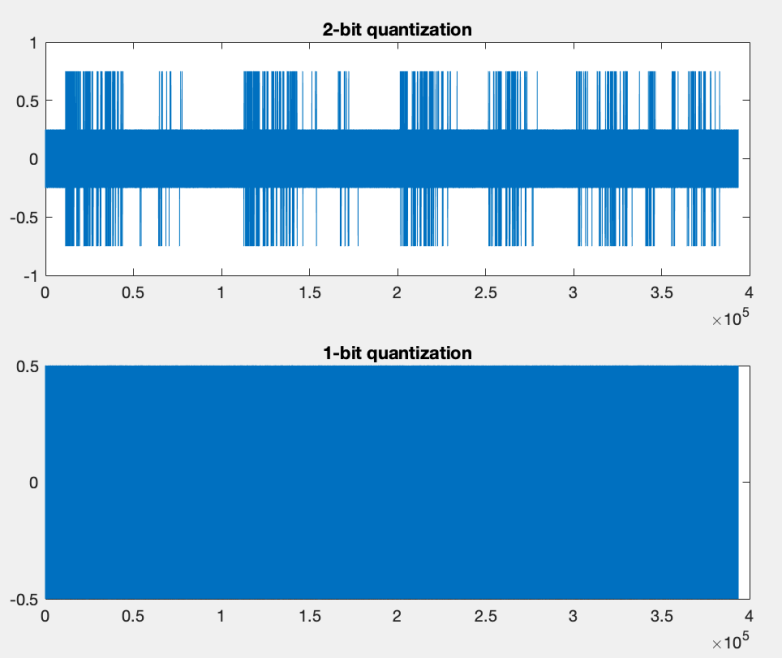
(b)

```
function y = quantizer_L_level(x, x_max, L)
Delta = 2*x_max/L;
y=(floor(x/Delta)+0.5)*Delta;
end
```

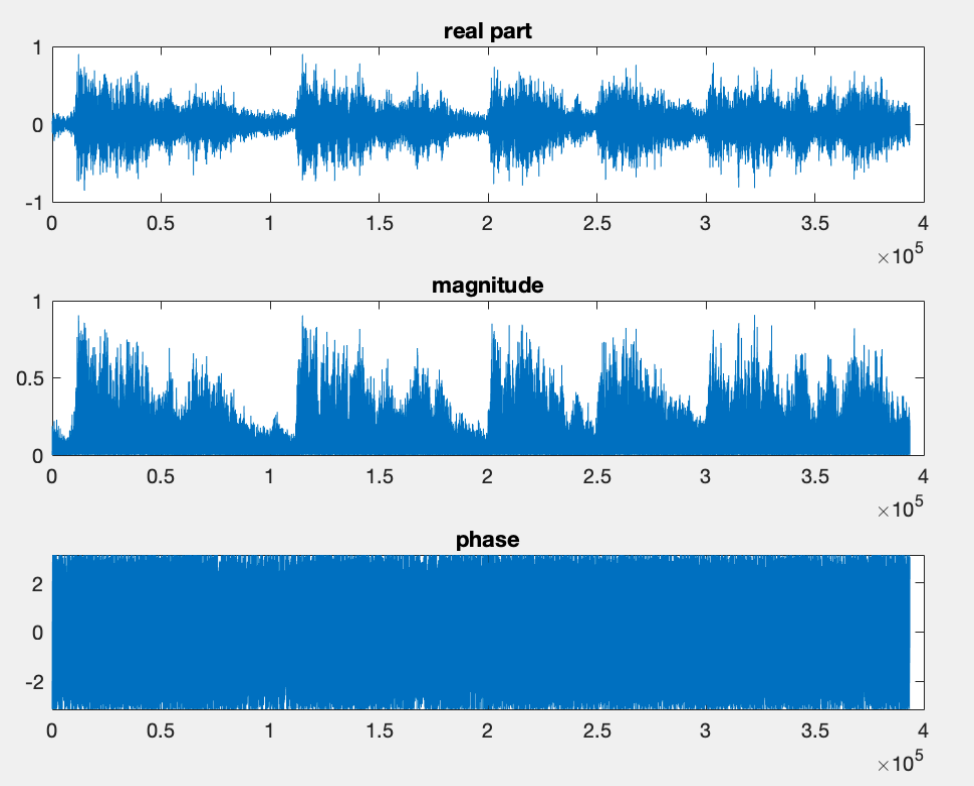
(c)出乎意料得，聲音聽起來其實還不差，跟原本的音樂很接近，只是背景有沙沙聲。



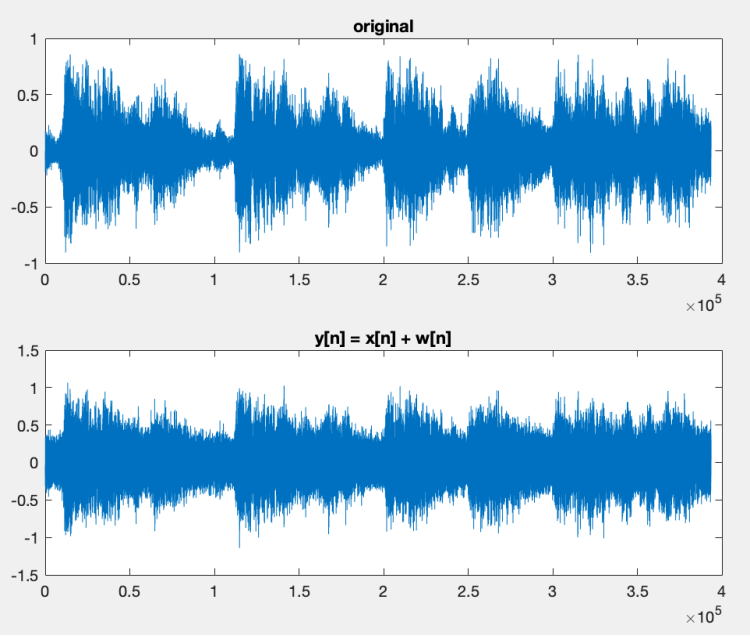
(d)我進一步把信號壓縮成2-bit和1-bit(數值剩下正負0.5)。
bit數越少，背景沙沙聲越重，但即便降到了1-bit，還是可以清楚聽到歌詞，甚至樂器的種類都還可以分辨，我想這是因為人類是靠辨識聲音頻率來接收聲音的，即便壓縮成1-bit，依然可以呈現原本聲波的頻率，所以還是可以聽得到音樂。



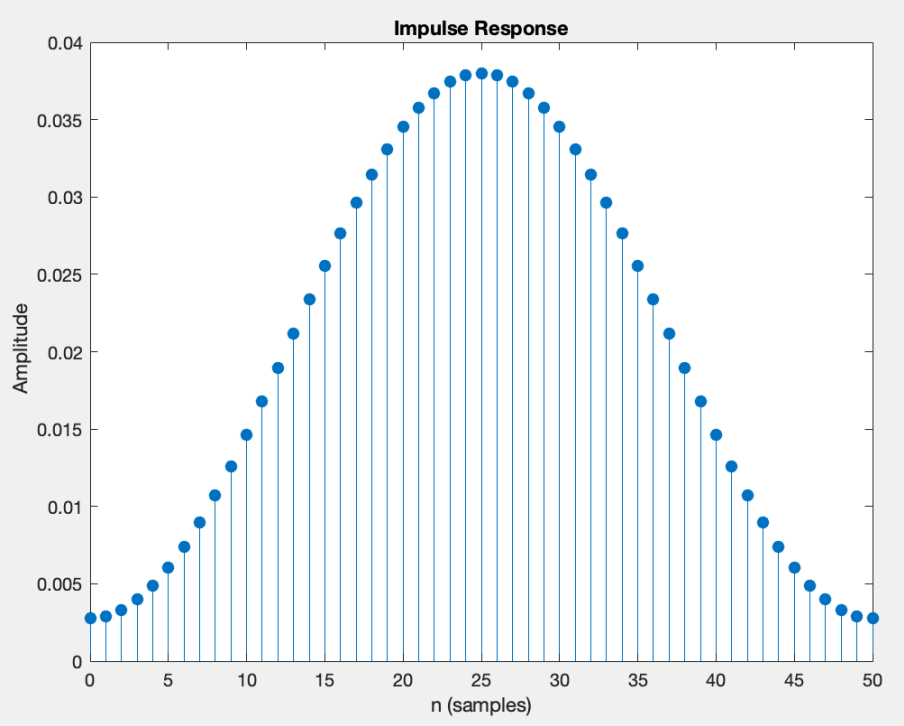
5.
改變後的聲音（real part），還聽得出來是原本的歌，但是(1)聲音變低沉 (2)聽起來像是對著塑膠水管唱歌一樣（有很類似的回音效果）。
這樣操作的效果，等價於把訊號乘以一個100Hz的sine波，原本的訊號和100Hz的波調和，因此音調變低沉。

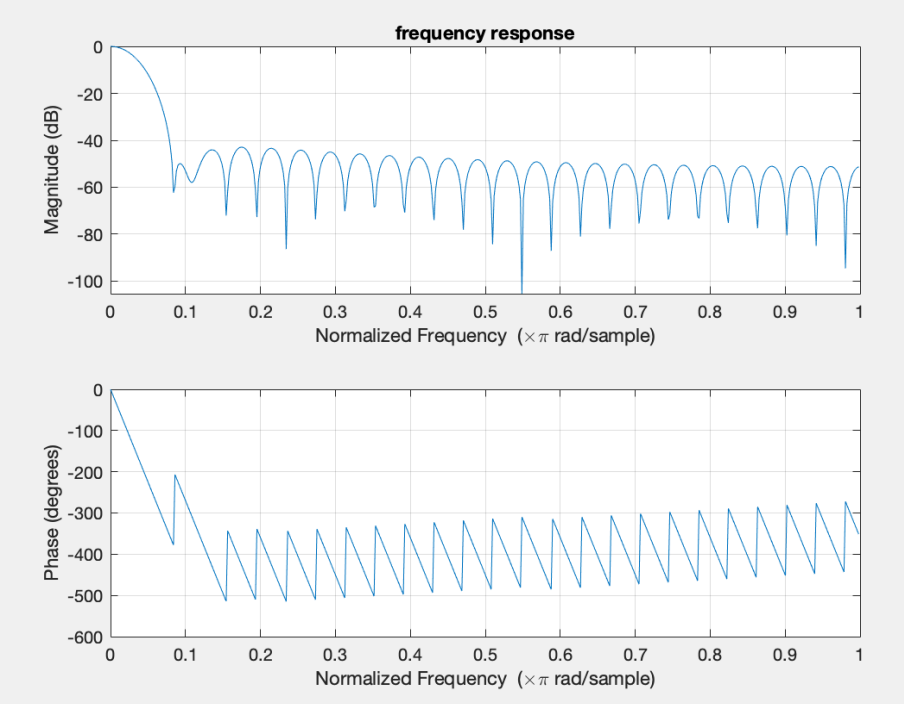


6.處理過後的聲音，背景出現了沙沙聲，好像有人在旁邊淋浴一樣。其實這樣就等價於原本的音樂跟WGN同時播放。
由圖表可以發現，(1)由於Gaussian noise的平均為0，聲波的包絡線大致沒變 (2)由於Gaussian noise偶爾會產生絕對值很大的數據點，改變後聲波的extreme value較原本的大，因此在圖表中可看見，y[n]那張圖的縱軸scale比較大。

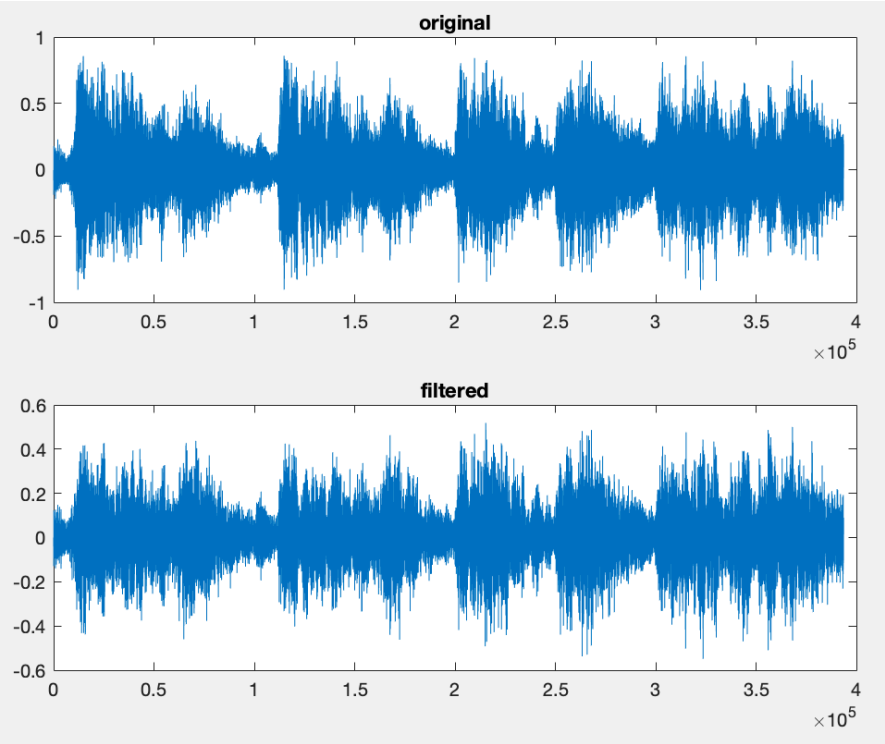


7.
(a)



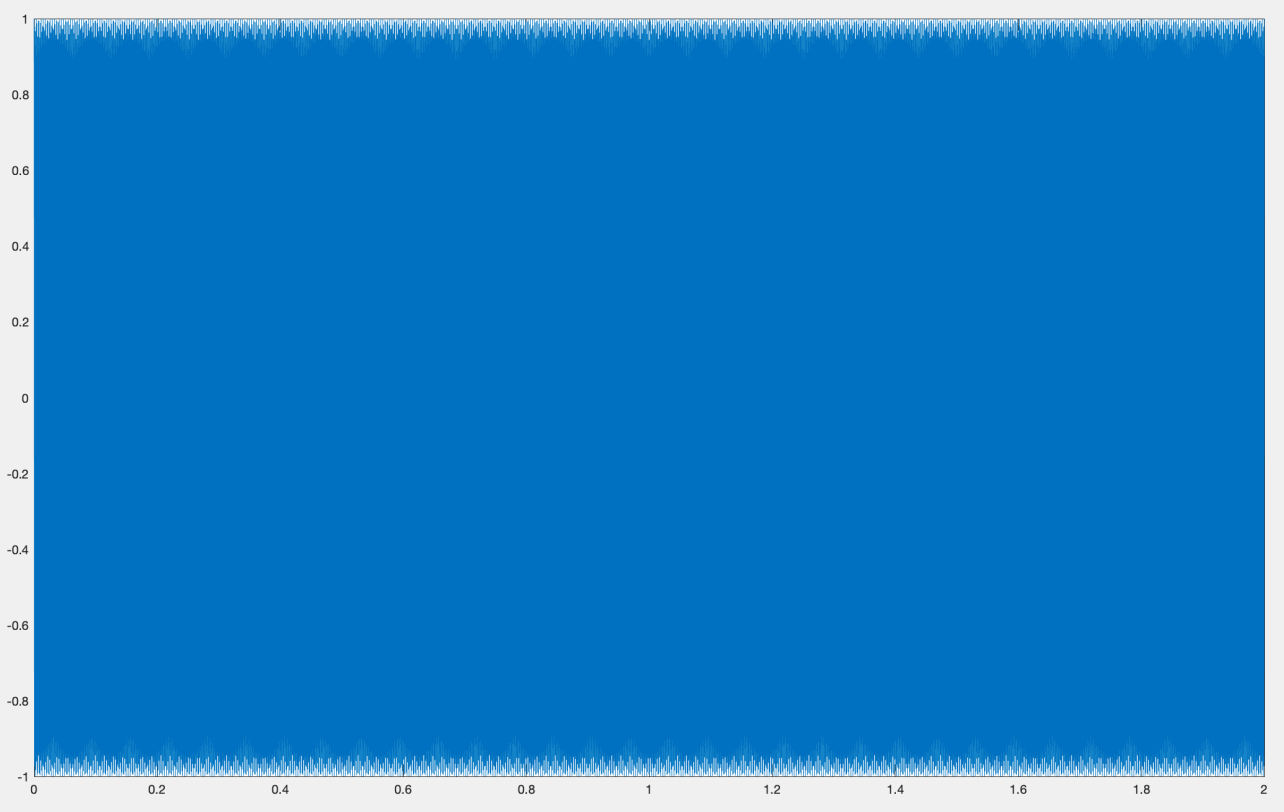


(b)聲音變低沉、變悶。
波形包絡線看起來十分類似。

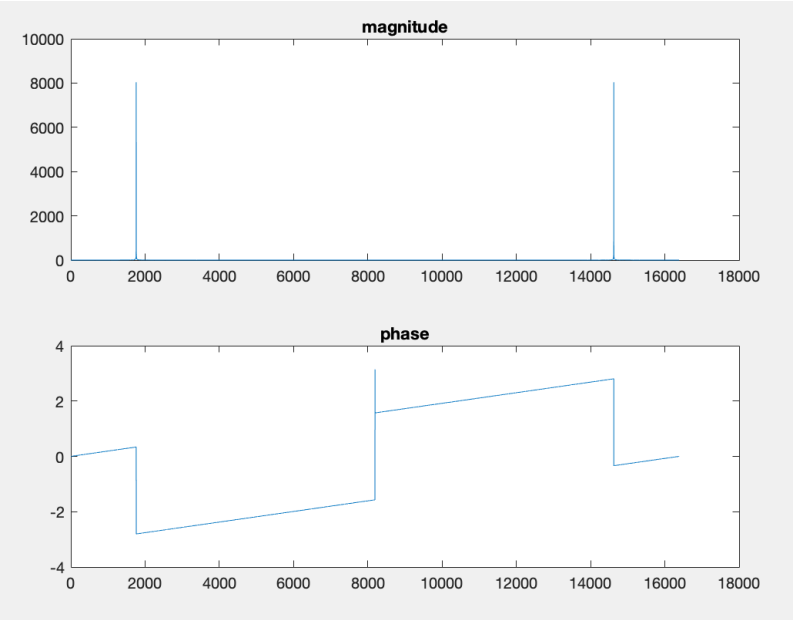


8.

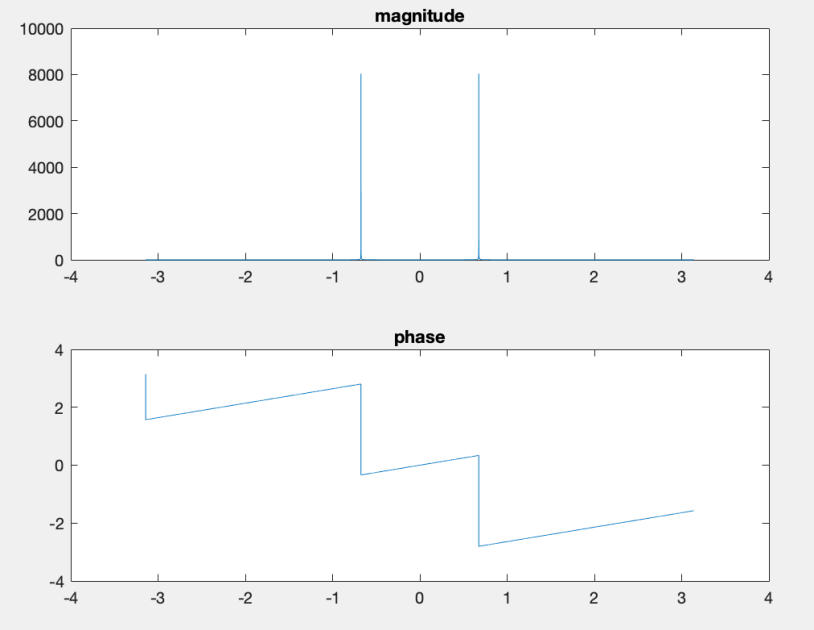
(a)聽起來是個高頻、單調的聲音。



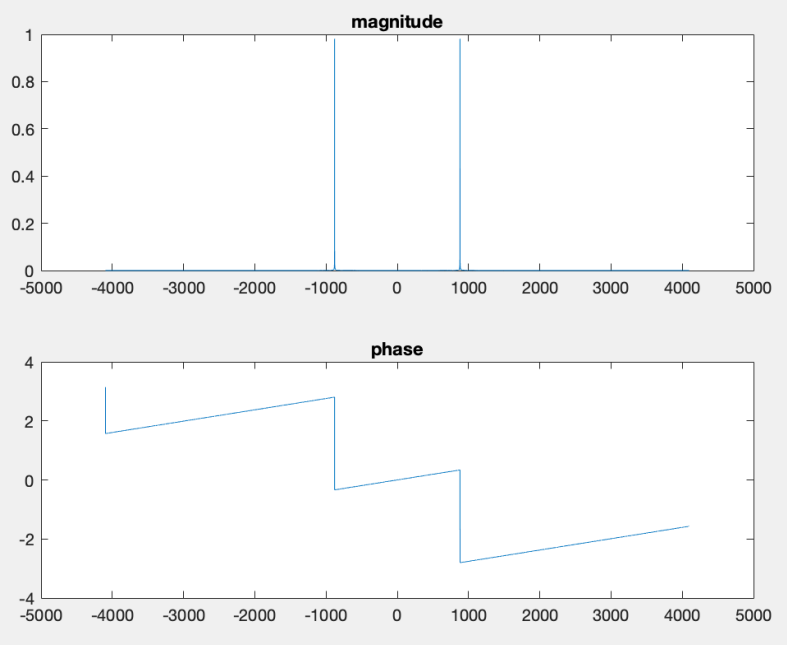
(b)由圖中可以看見，cosine的DFT就是兩根相位相反的delta function。其他地方的振幅是零，相位就沒什麼意義。



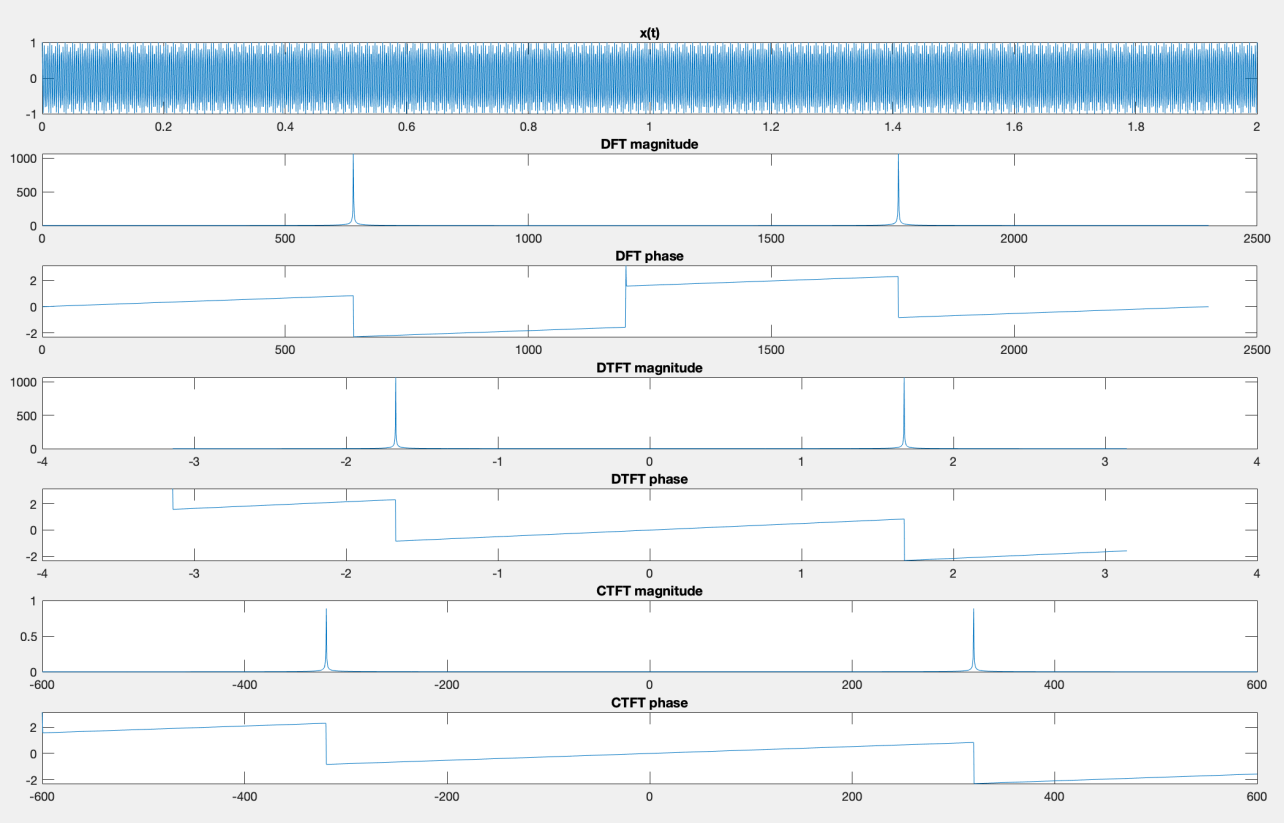
(c)由圖中可見，DTFT的結果，就是DFT的結果，經過circular shift後，再縮放到正負pi之間。



(d)由圖可見，兩根delta的大小變成1，位置也在正負fc的地方。



(e)由下圖可見，在CTFT中，由於fc>fs/2，delta出現的位置並不是在fc，而是在fs-fc的位置。



9.(additional)

(a)平移、縮放

方法：把第一題的聲音訊號x，乘上一個倍數再加上一個bias（x變成ax+b）。

結果：1.若a>1則聲音變大聲，若a<1則變小聲 2.當產生絕對值大於一的數據點時，會有類似3(a)加hard limit的效果。

討論：

原本的訊號x，絕對值大小在正負一之間，matlab在播放聲音時，絕對值大於1的數值也會被當作正負一處理，從實驗中可知，這樣的聲音效果聽起來很像喇叭開太大聲產生的破音。在1.中可觀測

到，藉由調整a的大小，我們可以控制聲音的強弱，但當a>1時，部分數據點的絕對值會大於一，這些數據會被truncade掉，進而產生破音的效果。在2.中，大致上平移縮放對於聲音的影響不大，但如果產生絕對值大於一的數據點（例如0.8x+0.5，有可能會有絕對值大於一的數據點，因為x在正負一之間），會有一樣的破音效果。

APPENDIX

All the simulation programs are stored in the following github link.
https://github.com/Andy19961017/Communication_System_Lab/tree/master/Lab1