

## HƯỚNG DẪN LAB 4B

### Bài 1: Load 2 file nhạc vào Matlab.

- Mix (Trộn) 2 bài nhạc lại với nhau và dùng hàm sound để nghe lại tín hiệu vừa tạo ở tần số lấy mẫu 44100 Hz.
- Để tạo tín hiệu echo sau tín hiệu gốc 1 giây trong Matlab ta có thể làm như sau:

```
[x,fs]=wavread('road.wav');
echo1s_x=x;
for i=fs:length(x)
    echo1s_x(i,:)=x(i,:) + x(i-fs+1,:);
end
```

Tương tự, hãy tạo tín hiệu echo sau tín hiệu gốc 0.5 giây và 2 giây mà **không sử dụng vòng for**. Nhận xét về điểm giống nhau của hiệu ứng mix và echo tín hiệu:

Giải :

- Để trộn 2 bài nhạc với nhau thì chúng ta cộng 2 mảng dữ liệu của 2 bài lại. File nhạc mà mấy anh gửi là âm thanh 2 kênh nên mảng dữ liệu là mảng 2 chiều có 2 cột. Chiều dài 2 mảng có thể khác nhau nên khi cộng lại ta cần thêm số 0 vào chỗ trống của mảng có ít phần tử hơn.

Code như sau :

```
[x1, fx1, n1] = wavread('hootie.wav');
[x2, fx2, n2] = wavread('road.wav');
[m,n] = size(x1);
[m1,n1] = size(x2);
data(1:m,1:n) = x1 + x2(1:m,1:n);
data(m+1:m1,1:n) = x2(m+1:m1,1:n);
sound(data,fx1)
```

Ở đây là mình đã biết mảng của file road.wav lớn hơn nên dữ từ đầu cho đến cuối mảng thứ nhất được cộng bình thường.

Dữ liệu từ chỉ số chiều dài mảng x1 trở đi bằng mảng x2.

Phần "1:n" các bạn có thể ghi ":" thôi cũng được.

- Tạo tín hiệu echo sau tín hiệu gốc 1 giây :

```
[x,fx]= wavread('road.wav');
echo1s_x = x;
for i = fx : length(x)
    echo1s_x(i,:)=x(i,:) + x(i-(fx/2)+1,:);
end
sound(echo1s_x,fx)
```

Tần số lấy mẫu là số mẫu trong 1s.

Echo 1s là sau khi bài nhạc phát được 1 giây thì nó cộng thêm tín hiệu của bài nhạc đó.

Bài nhạc phát được 1s tức là nó sẽ phát số mẫu bằng đúng với tần số lấy mẫu, nên khi cộng thêm tín hiệu ta cho biến chạy từ fs.

Nếu không sử dụng vòng lặp for, ta thực hiện tín hiệu echo như sau :

```
[x,fs]= wavread('road.wav');
x1(1:fs) = 0;
x1(fs:length(x),:) = x(1:length(x)-fs+1,:);
echo1s_x = x + x1;
% for i = fs : length(x)
%     echo1s_x(i,:)=x(i,:) + x(i-(fs)+1,:);
% end
sound(echo1s_x, fs);
```

câu lệnh :

```
x1(fs:length(x),:) = x(1:length(x)-fs+1,:);
```

hàng thứ fs trở đi của ma trận x1 sẽ bằng hàng thứ 1 trở đi của ma trận x. Chúng ta tạo một ma trận x1 bị trì hoãn 1s, có nghĩa là số mẫu tương ứng với 1s ban đầu = 0, các phần tử tiếp theo được lấy từ các phần tử đầu của mảng x.

Để tạo tín hiệu echo 0.5s hoặc 2s thì ta chỉ đổi fs thành fs/2 hoặc 2\*fs

**Bài 2:** Load file road.wav vào Matlab và cho phát lại bằng hàm sound của Matlab lần lượt với tốc độ nhanh hơn và chậm hơn tốc độ lấy mẫu 2 lần bằng 2 cách khác nhau.

**Cách 1 :**

Để phát tín hiệu nhanh hơn hoặc chậm hơn 2 lần chúng ta chỉ việc tăng hay giảm tần số lấy mẫu lên xuống 2 lần.

```
[x,fs] = wavread('road.wav');
sound(x,fs/2); % chậm đi 2 lần
sound(x,fs*2); % nhanh lên 2 lần
```

**Cách 2 :**

Để làm tín hiệu nhanh hơn 2 lần ta sẽ cắt bỏ đi 1 nửa tín hiệu, tức là lấy 1 mẫu bỏ 1 mẫu code như sau :

```
[x,fx] = wavread('road.wav');
% sound(x,fx/2);
% sound(x,fx*2);
xn = zeros(length(x)/2,2);
for j = 2:2:length(x)
    xn(j/2,:) = x(j,:);
end
sound(xn, fx);
```

Để làm tín hiệu chậm đi 2 lần thì ta thêm mẫu 0 xen vào giữa mỗi mẫu trong mảng dữ liệu, code như sau :

```
[x,fx] = wavread('road.wav');
xc = zeros(2*length(x),2);
for i = 2:2:2*length(x)
    xc(i,:) = x(i/2,:);
end
sound(xc, fx);
```

**Bài 3:** Thiết kế mạch lọc cao qua chebyshev có  $W_p = 1,2\text{KHz}$ ,  $W_s = 1\text{KHz}$ ,  $R_p = 0.5\text{dB}$ ,  $R_s = 20\text{dB}$  ứng với tần số lấy mẫu của file hootie.wave.

- Vẽ lại đáp ứng biên độ và pha của mạch lọc vừa tạo.
- Sử dụng mạch lọc trên lọc tín hiệu từ file hootie.wav và dùng hàm sound để nghe lại tín hiệu ra. Dùng hàm plotfft để xem phổ ra của tín hiệu trước và sau khi lọc (lần lượt ở 2 kênh trái và phải). Nhận xét và giải thích:

```
function []=plotfft(wave,fs)
N = 10240; % Number of points
l=length(wave);
waves(1,:)=wave(1,(l-N+1):l);
win=hanning(N);
waves=win.*waves';
wavefft=fft(waves,N);
wave_db=20*log10(abs(wavefft));
figure;
plot([0:N/2-1].*fs/N,wave_db(1:N/2));
grid on;
title('FFT PLOT');
xlabel('ANALOG INPUT FREQUENCY (MHz)');
ylabel('AMPLITUDE (dB)');
end
```

Thiết kế mạch lọc cao qua chebyshev ta thực hiện như sau :

Đầu tiên chuyển tần số tương tự sang tần số số :

```
fx = 44100;
Wp = 2*1.2e3/fx;
Ws = 2*1e3/fx;
```

Kể đến dùng các hàm thiết kế mạch lọc, các bạn có thể thiết kế mạch lọc bậc 1 hoặc bậc 2, ở đây mình dùng bậc 2 :

```
% thiết kế lọc chebyshev bậc 2
[N,Ws] = cheb2ord(Wp,Ws,Rp,Rs);
[num,den] = cheby2(N,Rs,Ws,'high');
```

Phần vẽ thì các bạn vẽ bình thường, tuy nhiên phần vẽ pha thì nhớ dùng hàm unwrap để tránh chồng lấn pha.

Dùng hàm đã thiết kế để lọc tín hiệu :

```
[x,fx] = wavread('hootie.wav');
sound(x,fx);
A = filter(num,den,in);
sound(A,fx);
```

Mình có sound(in,fx) để phát tín hiệu trước lọc để dễ so sánh sự khác nhau. ^^

Hàm `plotfft` chỉ vẽ đồ thị cho từng cột nên khi vẽ các bạn chỉ lấy dữ liệu của 1 cột để vẽ thôi. Trong code mình có viết :

```
% plotfft((x(:,1))',fx); % Tín hiệu fft của cột 1 trong mảng dữ liệu gốc
% plotfft((A(:,1))',fx); % tín hiệu fft của cột 1 trong mảng sau lọc
% plotfft((x(:,2))',fx); % Tín hiệu fft của cột 2 trong mảng dữ liệu gốc
% plotfft((A(:,2))',fx); % tín hiệu fft của cột 2 trong mảng sau lọc
```

Các bạn muốn xem tín hiệu nào thì bỏ dấu `%` khỏi dòng đó là được.

- c) **Biết giọng người (vocal) tập trung ở tần số từ 1 KHz – 3 KHz, thiết kế mạch lọc chebyshev để lọc tần số vocal ở file road.wav ( $R_p = 0.5\text{dB}$ ,  $R_s = 20\text{dB}$ ).** Cũng từ file road.wav, tạo thêm một tín hiệu khác bằng cách lấy tín hiệu ở kênh trái trừ cho tín hiệu ở kênh phải. Phát lại bằng hàm `sound` để so sánh 2 tín hiệu này. Nhận xét và giải thích:

Code thiết kế mạch lọc và thực hiện câu c như sau :

```
clf;
[in,fs] = wavread('road.wav');
W1 = 2*0.6e3/fs; % tần số pass qua bên trái
W2 = 2*1e3/fs;   % tần số stop bên trái
W3 = 2*3e3/fs;   % tần số stop bên phải
W4 = 2*3.4e3/fs; % tần số pass bên phải
Ws = [W2 W3];
Wp = [W1 W4];
Rp = 0.5;
Rs = 20;
% Thiết kế bộ lọc stopband chebyshev bậc 1
[N,Wp] = cheblord(Wp,Ws,Rp,Rs);
[num,den] = cheby1(N,Rp,Wp,'stop');
A = filter(num,den,in);
sound(in,fs);
sound(A,fs);
% Tạo tín hiệu bằng cách lấy kênh trái trừ kênh phải
T = zeros(length(in),1);
T(:,1) = in(:,1) - in(:,2);
sound(T,fs);
% vẽ đáp ứng của bộ lọc
w = 0 : 0.001 : pi;
H = freqz(num,den,w);
subplot(1,2,1)
plot(w/pi,20*log10(abs(H))); grid
title('đáp ứng biên độ lọc stop band');
subplot(1,2,2);
plot(w/pi,unwrap(angle(H)));
title('đáp ứng pha lọc stop band');
```

Các chú thích mình đã chú thích trong code hết rồi nên không phải giải thích nữa nhé.

**KẾT THÚC LAB4B Ở ĐÂY NHÉ ^^**