Lab 7

Filter banks

0. Abstract

In signal processing, a filter bankis an array of bandpass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal. In this lab, we will first start with a easy filter bank and know the property of it. Then we will use a simulation of cochlea to explore the frequency response find the application of it.

1. Sections

1.1. Plot the magnitude response of each of these filters

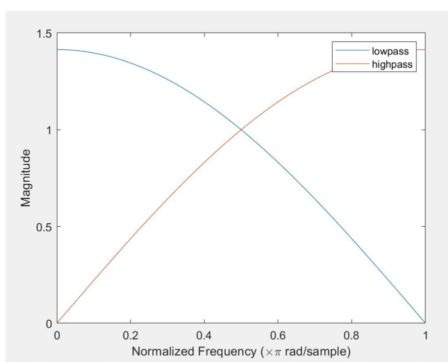

Figure 1: The magnitude response of each of these filters

1.2. Plot the random signal

The code is as follows:

```
function exe1_2()
x = zeros(1,512);
for i = 1 : 512
for j = 0 : 9
x(i) = x(i) + rand(1) * cos(2* pi * rand(1) * i + 2 * pi *rand(1));
```

```
end
end
b1 = (2 ^ (-0.5)) .* [1 1];
b2 = (2 ^ (-0.5)) .* [1 -1];
a = [1];
Ndata = 1024;
fs = 2 * pi;
f=(0:Ndata-1)*fs/Ndata;
han = hann(512);
for i = 1:512
han_win(i) = han(i) * x(i);
end
FT=fft(han_win,Ndata);
FT_dB = 20 * log10(abs(FT)./max(abs(FT)));
t = 0 : 1/1024 : 1;
subplot(2,1,1);plot(x);xlabel('Time(n)');ylabel('Amplitude');
subplot(2,1,2);plot(t(1:Ndata/2),FT_dB(1:Ndata/2));xlabel('Normalized
Frequency');ylabel('Amplitude');axis([0 0.5 -60 0]);
```

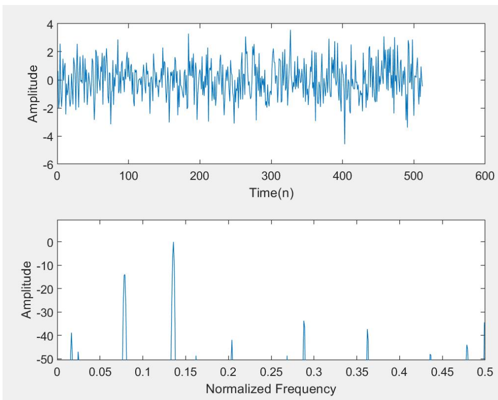

Figure 2: The waveform of the signal and frequency response

1.3. Answer to the question Pass the signal through filter bank

The code is as follows:

```
x = zeros(1,512)
for i = 1 : 512
for j = 0 : 9
x(i) = x(i) + rand(1) * cos(2* pi * rand(1) * i + 2 * pi *rand(1));
end
end
b1 = (2 ^ (-0.5)) .* [1 1];
b2 = (2 ^ (-0.5)) .* [1 -1];
a = [1];
Ndata = 1024;
fs = 2 * pi;
```

```
han = hann(512);
for i = 1:512
han_win(i) = han(i) * x(i);
end
FT=fft(han_win,Ndata);
f=(0:Ndata-1)*fs/Ndata;
ydb = 20 * log10(abs(FT)./max(abs(FT)));
x1 = filter(b1,a,x);
x2 = filter(b2,a,x);
scal = (x1 + x2) ./ x;
subplot(311);plot(x);
subplot(312);plot(x1);ylabel('Amplitude');
subplot(313);plot(x2);xlabel('Time(s)');
```

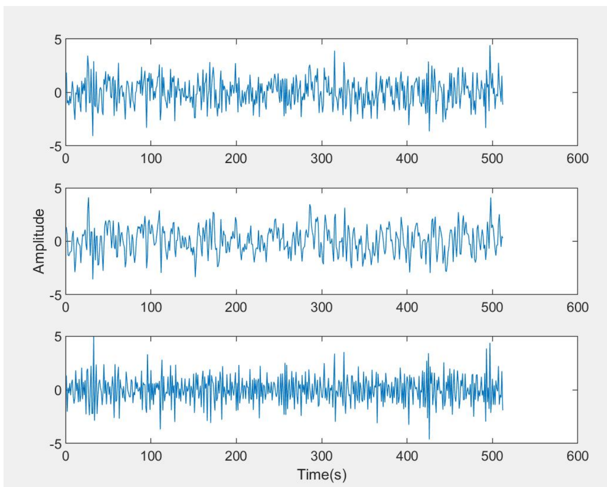

Figure 3: The time-domain signal with different filter

From the time domain plot, we can see that H0 is the high pass filter and H1 is the low pass filter.

1.4. Verify the scaled sum of two channels

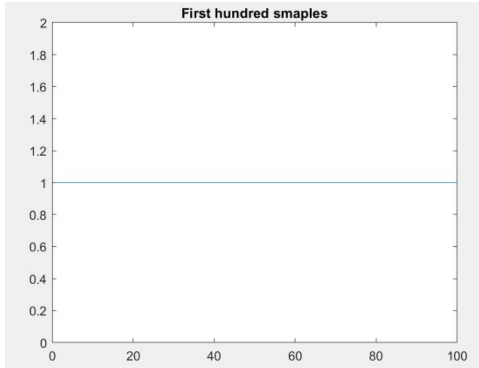

Figure 4: Verify the scaled sum of two channels

1.5. Answer the question

I don’t think we can get the original signal back because when downsampling the data, half of the image is skipped and we we combine the two separate channels together. Although the data-rate is maintained, but some information is still omitted.

1.6 .Use the Hann window for each signal

The code is as follows:

```
function exe1_6()
x = zeros(1,512);
for i = 1 : 512
for j = 0 : 9
x(i) = x(i) + rand(1) * cos(2* pi * rand(1) * i + 2 * pi *rand(1));
end
end
b1 = (2 ^ (-0.5)) .* [1 1];
b2 = (2 ^ (-0.5)) .* [1 -1];
a = [1];
Ndata = 1024;
fs = 2 * pi;
f=(0:Ndata-1)*fs/Ndata;
han = hann(512);
for i = 1:512
han_win(i) = han(i) * x(i);
end
FT=fft(han_win,Ndata);
FT_dB = 20 * log10(abs(FT)./max(abs(FT)));
x1 = filter(b1,a,x);
x2 = filter(b2,a,x);
t = 0 : 1/1024 : 1;
```

```
for i = 1:512
han_win1(i) = han(i) * x1(i);
end
FT1=fft(han_win1,Ndata);
ydb1 = 20 * log10(abs(FT1)./max(abs(FT1)));
for i = 1:512
han_win2(i) = han(i) * x2(i);
end
FT2=fft(han_win2,Ndata);
ydb2 = 20 *log10(abs(FT2)./max(abs(FT2)));
subplot(3,1,1);plot(t(1:Ndata/2),FT_dB(1:Ndata/2));xlabel('Normalized
Frequency');axis([0 0.5 -60 0]);
subplot(3,1,2);plot(t(1:Ndata/2),ydb1(1:Ndata/2));ylabel('Magnitude(dB)');axis
([0 0.5 -60 0])
subplot(3,1,3);plot(t(1:Ndata/2),ydb2(1:Ndata/2));xlabel('Normalized
Frequency');axis([0 0.5 -60 0])
end
```

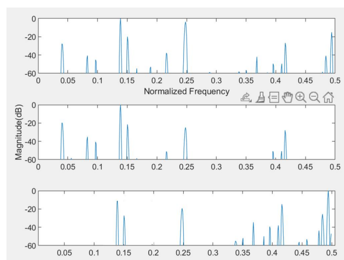

Figure 5: (Top) the frequency content of the random input to the system (Middle) the frequency content of the output from the upper path. (Bottom) the frequency content of the output from the lower path.

1.7 Provide some interpretation

We can notice that some peaks in the orginal signal with hann window disappear in the second and third plot. Since H0 is a low pass filter so the frequency more than 0.4 normalized frequency is filters so it can not be shown in the second plot in 1.6. Since H1 is a high pass filter so the frequency less than 0.1 normalized frequency is filters so it can not be shown in the third plot in 1.6. Because of the hanning window there is also some small peak near the real peak.

1.8 Plot the frequency content of this system

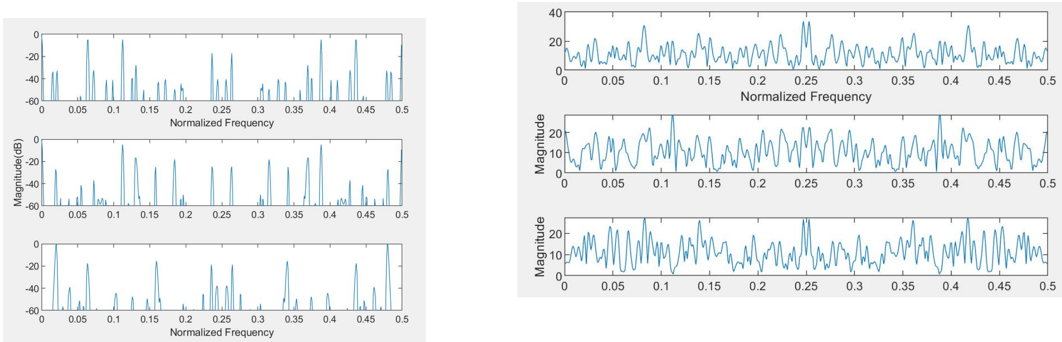

Figure 6: The frequency content of this system for random signal (in dB and magnitude)

1.9 Plot the frequency content of the input as well as the output of two channels in the system

The commented code is as follows:

```
function exe1_6()
x = zeros(1,512);
for i = 1 : 512
for j = 0 : 9
x(i) = x(i) + rand(1) * cos(2* pi * rand(1) * i + 2 * pi *rand(1));
end
end
b1 = (2 ^ (-0.5)) .* [1 1];
b2 = (2 ^ (-0.5)) .* [1 -1];
a = [1];
Ndata = 1024;
fs = 2 * pi;
f=(0:Ndata-1)*fs/Ndata;
han = hann(512);
for i = 1:512
han_win(i) = han(i) * x(i);
end
han_win = upsample(han_win(1:2:end),2);
FT=fft(han_win,Ndata);
FT_dB = 20 * log10(abs(FT)./max(abs(FT)));
x1 = filter(b1,a,x);
x2 = filter(b2,a,x);
t = 0 : 1/1024 : 1;
for i = 1:512
han_win1(i) = han(i) * x1(i);
end
han_win1 = upsample(han_win1(1:2:end),2);
FT1=fft(han_win1,Ndata);
ydb1 = 20 * log10(abs(FT1)./max(abs(FT1)));
for i = 1:512
```

```
han_win2(i) = han(i) * x2(i);
end
han_win2 = upsample(han_win2(1:2:end),2);
FT2=fft(han_win2,Ndata);
ydb2 = 20 * log10(abs(FT2)./max(abs(FT2)));
for i = 1:512
han_win3(i) = han(i) .* (x2(i)+x1(i));
end
han_win3 = upsample(han_win3(1:2:end),2);
FT3=fft(han_win3,Ndata);
ydb3 = 20 * log10(abs(FT3)./max(abs(FT3)));
subplot(4,1,1);plot(t(1:Ndata/2),abs(FT(1:Ndata/2)));xlabel('Normalized 
Frequency');
subplot(4,1,2);plot(t(1:Ndata/2),abs(FT1(1:Ndata/2)));ylabel('Magnitude');
subplot(4,1,3);plot(t(1:Ndata/2),abs(FT2(1:Ndata/2)));ylabel('Magnitude');
subplot(4,1,4);plot(t(1:Ndata/2),abs(FT3(1:Ndata/2)));xlabel('Normalized 
Frequency');
```

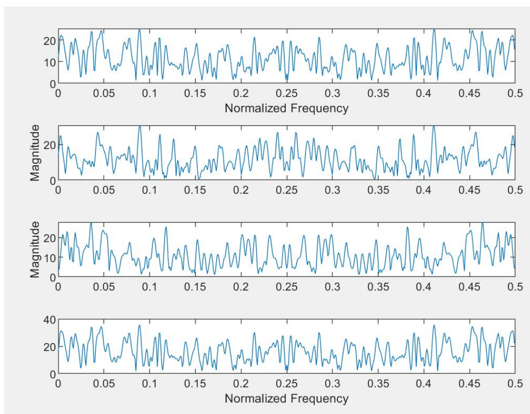

Figure 7: Plot the frequency content of the input as well as the output of two channels in the system

2.1 Running the simulation of a cochlea

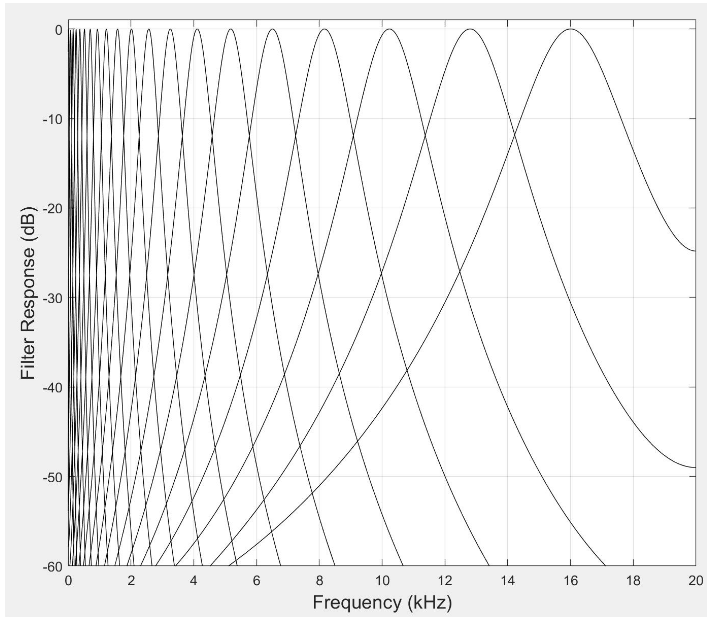

Figure 8: The resulting plot of simulating a cochlea

2.2 Answer a question

Since each filter in this filter bank that has a sharp resonance about some frequency has a linear phase. The filter is FIR.

2.3 Run the code provided and explain

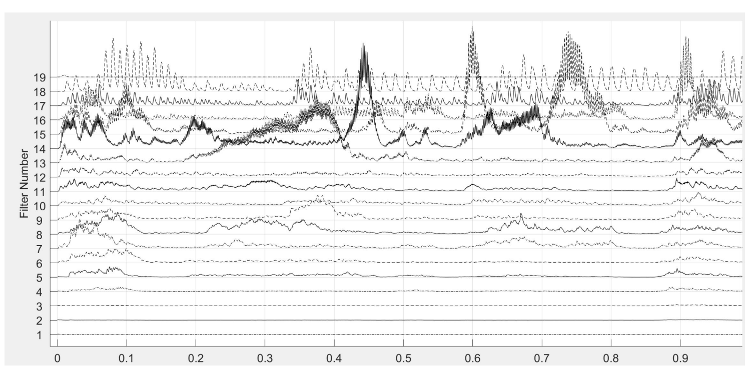

Figure 9: The resulting plot of the code

The fcoefs in the line 7,which is the output of the MakeERBFilters, is the coefficients of the filter.And the xout is the output of the filter. A loop is used in line 20 which uses the length of the xout as an argument. It first chooses the max number in a raw, then use it as the input data. The filter function filters the input data c using a rational transfer function defined by the numerator and denominator coefficients b =[1] and a = [1 -0.99]. And at last in order to maintain the data rate, c is downsampled.

2.4 Increase the number of channels to 100

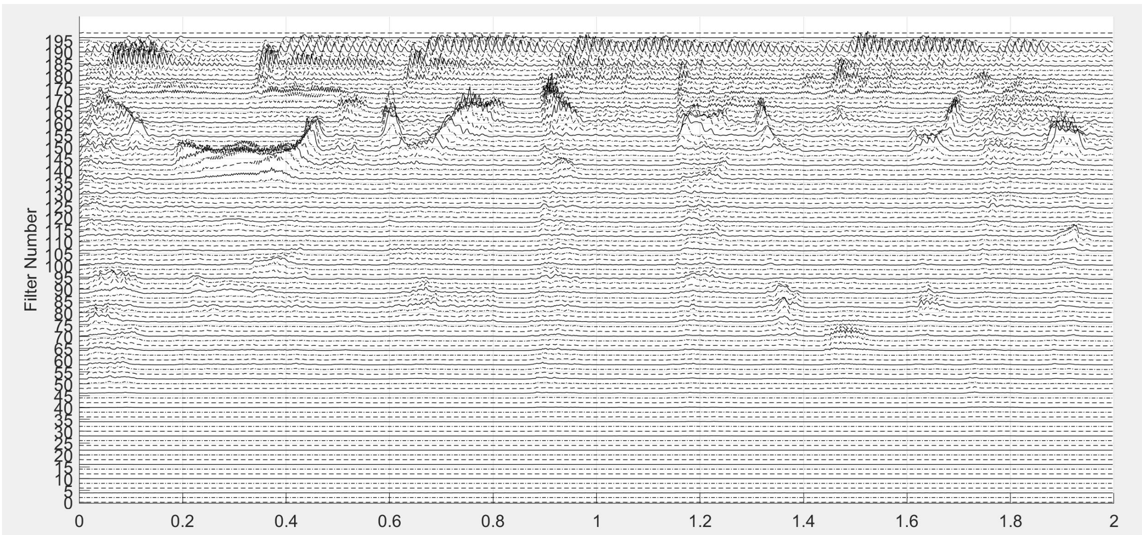

Figure 10: The resulting plot of the code(when increasing the channels to 100)

2.5 Discuss the sound

The above code is using a loop and sound the x line by line, starting with the 50th line, with the frequency Fs = 44100. Basically, it is repeating the same part in the original sound. But the sound on each line is not exactly the same because it is output of a filter. The sound is blurred and more hard to recognize as the j increases.

2.6 Discuss one way for audio compression

The application of filter banks is signal compression when some frequencies are more important than others. After decomposition, the important frequencies can be coded with a fine resolution. Like in 2.5, when the j is over 80, xout(j,:) is no longer important, since the signal can be well reconstructed by the previous important frequencies.

2. Conclusion

In lab7, we first create a random signal and analyze the frequency response of it. Then a simple filter bank is provided to filter the signal, we have to downsample it in order to keep the data rate. Then we explore a simulation of cochlea, and have a basic idea how the cochlea works. At last, we find out that audio compression is one of the application of this model.

3. Acknowledgments

Thanks for the TA gives us the illustration explain the basic principle of Filter Bank.