ELEC9123: Spectrum Analyser Project

Digital Signal Processing Design

Introduction

According to Fourier theory, any time-domain signal can be represented as a sum of one or more sine waves of appropriate frequency, phase, and amplitude. In other words, any time-domain signal can be transformed into its frequency-domain equivalent. Measurements in the frequency domain will give us information concerning the amount of energy in a given signal at particular frequencies.

Spectrum analysers are used to present frequency-domain information of input signals, i.e. to analyse the frequency components of a given time-domain signal. Such analysers are useful in a wide range of applications, from lab equipment to data coding schemes to noise cancellation and speech enhancement.

![Figure 1: Time-domain and Frequency-domain representations of a signal](image)

Project Definition

You are required to design a spectrum analyser that will take an audio input signal $x[n]$ and decompose it into $N$ frequency bands $x_1[n] \ldots x_N[n]$, as specified by Figure 2 below. The audio signal will be sampled at a rate of $f_S$; the centre frequencies and bandwidths of the band-pass filters are detailed in an Appendix A to this document. The output of this system will be a measure of the energy in each frequency band, sampled every $T_p$ seconds (a different rate to the initial sampling frequency $f_S$). This design should be carried out in the MATLAB environment.
**Sampling**

Examine the following snippets of MATLAB code:

```matlab
Fs = 32000;
t = 0:1/Fs:2; % t in seconds
y = 0.4*sin(2*pi*720*t) + 0.75*sin(2*pi*1020*t) + 1.1*sin(2*pi*2210*t);
k=0.02*Fs;
plot(t(1:k), y(1:k))
soundsc(y,Fs);
```

```matlab
Fs = 32000;fa1 = 720;ffa2 = 1020;ffa3 = 2210;
ffl = 2*pi*ffa1/Fs;ff2 = 2*pi*ffa2/Fs;ff3 = 2*pi*ffa3/Fs;
n = 1:1000;
x = 0.4*sin(n*ffl) + 0.75*sin(n*ff2) + 1.1*sin(n*ff3);
plot(n,x)
```

1. What is the sampling period of these signals?

2. What difference is there between the first and second portions of code?

3. What frequencies are present in the input signals x and y?

4. Try changing the sampling rate Fs to other frequencies (try 1.5kHz, 3kHz and 8kHz to start with). What changes do you notice? Plot the output signal; try listening to it too.
ELEC9123: Spectrum Analyser Project

Now type `load handel` into MATLAB (you might like to try loading ‘chirp’ and ‘splat’ too). Re-sample this signal at the half the sampling frequency $F_s$, and again at $F_s/4$.

5. What changes do you notice? Listen to the output signal.

6. Examine the frequency spectrum using the MATLAB tool `specgram`. Explain what you can see.

**Filter design**

There are a number of ways that filter requirements can be specified. Below are a few examples. Implement each of these as digital filters for a specific frequency range (i.e. one or more of the band-pass filters specified in Appendix A) and compare their effectiveness. Use any of the signals from part 1 to test your system.

Some aspects you might like to consider include:

- Impulse response
- Computational requirements
- Poles/zeros
- Stability
- Phase response
- FIR/IIR
- Amplitude response
- Any other relevant data

**Analogue Filter types:** (a) Butterworth, (b) Elliptic, (c) Chebyshev I, (d) Chebyshev II – State the main characteristics of the four analogue filters

**Transfer function**

A band-pass filter is described by the transfer function $H(s)$:

$$H(s) = \frac{b_p s}{s^2 + b_p s + \omega_p^2}$$

where $\omega_p$ and $b_p$ are the centre frequency and bandwidth of the filter respectively, both expressed in rad/s.

1. What kind of transformation, if any, is needed to digitally implement this filter?

2. Implement this filter in MATLAB, demonstrating any analytical theory work you used to arrive at your solution.
Digital Filter structure

The diagram below represents a digital filter, with the multiplication factors \( a_n \) and \( b_n \) given in the following table.

<table>
<thead>
<tr>
<th>( a_1 )</th>
<th>( b_1 )</th>
<th>( a_2 )</th>
<th>( b_2 )</th>
<th>( a_3 )</th>
<th>( b_3 )</th>
<th>( a_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7821</td>
<td>0.0061</td>
<td>-2.5640</td>
<td>-0.0122</td>
<td>1.5853</td>
<td>0.0061</td>
<td>-0.7920</td>
</tr>
</tbody>
</table>

1. Implement this filter in MATLAB.
2. Where are the poles and zeros?
3. Can you evaluate the impulse response \( h[n] \)?
4. How does this system respond to an impulse input? How about a square wave? An arbitrary input?
5. Comment on its utility for this project.

Poles/Zeros Placement

The poles and zeros for three different filters are given in the following table.

<table>
<thead>
<tr>
<th>System A</th>
<th>System B</th>
<th>System C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poles</td>
<td>Zeros</td>
<td>Poles</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0.75 + 0.25j )</td>
<td>( 0.76 + 0.64j )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 0.75 - 0.25j )</td>
<td>( 0.76 - 0.64j )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 1.20 + 0.40j )</td>
<td>( -0.98 + 0.12j )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>( 1.20 - 0.40j )</td>
<td>( -0.98 - 0.12j )</td>
</tr>
</tbody>
</table>

1. Evaluate the filter characteristics. 2. Comment on their uses.
3. How does each system respond to an impulse input? How about a square wave? An arbitrary input?
4. Are any of these suitable for this project?
5. If not, how might you go about choosing poles and zeros to achieve the desired filter characteristics?
Numerical design methods
MATLAB’s `remez()` or `firpm()` algorithm uses the Parks-McLennan method for FIR filter design. An example of its use is given below.

```matlab
% example for filter n = 9
Fs = 8000; f = [0 1000/Fs 1080/Fs 1270/Fs 1370/Fs 1];
m = [0 0 1 1 0 0];
B = remez(50,f,m); A = 1;
zplane(B,A);
```

1. Can you adapt this to be of use?
2. How does it compare to other filter designs?

Frequency Response
The criteria for a low pass filter are described by the following frequency response plot, where $f_P = 1000\text{Hz}$ and $f_S = 1270\text{Hz}$.

1. Implement this filter in MATLAB and check its frequency response against the specification.
2. How would you convert this specification to a band-pass filter required for the spectrum analyser?
3. What aspects will you need to watch out for?
4. How can you implement this form of filter?

Other filter design methods
1. From your research into filter design, what other methods are available to you?
2. Does MATLAB have any tools that will aid you in your design, or in specifying the criteria?
Design Choice
Choose one of the filtering methods you have investigated above, and implement the entire filterbank. Be able to justify your choice.

Frequency analysis
Having implemented a filter bank, you will now have $N$ filtered signals, $x_{1...N}[n]$.

1. What tools do you have to examine what frequencies are present in these filtered signals? What are the relevant theorems?

2. Examine the frequency spectrum for each filtered signal. Is it what you expected/designed for?

3.

Rectification and Filtering
After the band-pass filter block, each of the signals is rectified, and put through a low-pass filter. This low-pass filter is commonly first order, and has a cut-off frequency $f_p$ of 30Hz. The switch on the output of the LPF block samples this resultant data every $T_p = 16ms$.

1. The output of this sub-system is a measure of the energy of the signal at the output of each band-pass filter. Demonstrate why this is the case.

2. Examine the output of your spectrum analyser, and compare it to the data obtained in the last section. Is it what you expected/designed for?

3. What is the best way to communicate this data? Does it accurately reflect the components present in the original audio signal?

4. What effect does sampling rate here have on the information extracted? What would the implications be if the switch period $T_p$ were increased to 125ms?

Multi-rate Processing
Assume that for a specific application, the audio input to this device is speech. It is also required that you change the sampling rate of the input signal from 8192Hz to 10.24kHz.

1. What scaling ratio is required?

2. What filter bandwidths do you need?

3. Implement the rate-change block.

4. Is your implementation efficient? Can it be improved?
Evaluation

1. Using MATLAB, plot the frequency response of each band pass filter on a single plot. The plot should be given with centre frequency of each filter and the magnitude normalised to 0dB.

2. Write a MATLAB program to generate a test signal consisting of sum of many sine waves as an input to the band-pass filters and plot the output of the spectrum analyser to generate a 2D plot (filter number versus magnitude) of the spectrum of the input signal.

Appendix A – Analogue Band-Pass Filter Frequencies

<table>
<thead>
<tr>
<th>Filter No.</th>
<th>Lower Cut-Off Frequency (Hz)</th>
<th>Centre Frequency (Hz)</th>
<th>Upper Cut-Off Frequency (Hz)</th>
<th>Bandwidth (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>450</td>
<td>510</td>
<td>110</td>
</tr>
<tr>
<td>5</td>
<td>510</td>
<td>570</td>
<td>630</td>
<td>120</td>
</tr>
<tr>
<td>6</td>
<td>630</td>
<td>700</td>
<td>770</td>
<td>140</td>
</tr>
<tr>
<td>7</td>
<td>770</td>
<td>840</td>
<td>920</td>
<td>150</td>
</tr>
<tr>
<td>8</td>
<td>920</td>
<td>1000</td>
<td>1080</td>
<td>160</td>
</tr>
<tr>
<td>9</td>
<td>1080</td>
<td>1170</td>
<td>1270</td>
<td>190</td>
</tr>
<tr>
<td>10</td>
<td>1270</td>
<td>1370</td>
<td>1480</td>
<td>210</td>
</tr>
<tr>
<td>11</td>
<td>1480</td>
<td>1600</td>
<td>1720</td>
<td>240</td>
</tr>
<tr>
<td>12</td>
<td>1720</td>
<td>1850</td>
<td>2000</td>
<td>280</td>
</tr>
<tr>
<td>13</td>
<td>2000</td>
<td>2150</td>
<td>2320</td>
<td>320</td>
</tr>
<tr>
<td>14</td>
<td>2320</td>
<td>2500</td>
<td>2700</td>
<td>380</td>
</tr>
<tr>
<td>15</td>
<td>2700</td>
<td>2900</td>
<td>3150</td>
<td>450</td>
</tr>
<tr>
<td>16</td>
<td>3150</td>
<td>3400</td>
<td>3700</td>
<td>550</td>
</tr>
</tbody>
</table>

Table 1: Frequency bands