INTRODUCTION

In this laboratory, you will design and evaluate a number of digital filters using the signal processing functions in MATLAB. This will allow you to directly compare the analogue elliptic filter available on the TIMS panel with a digital implementation and to investigate the comparative advantages of digital IIR and FIR filters. To conclude our laboratory module on digital filter designs, we will consider how to compute, using primitive functions, the output from a filter with a given input sequence and a given filter impulse response.

Before attending this laboratory, you should understand the following concepts:

- Differences between FIR and IIR Filters.
- Linear phase filters and their importance.
- Convolution.

LABORATORY PREPARATION

The frequency response of the TIMS Tuneable Lowpass (Elliptic) Filter is shown in Figure 2. The cutoff frequency (-3dB frequency) was set to 4kHz for this measurement. An equivalent digital filter will be designed. The first step is to assign values to the filter specification shown in Figure 1.

Find values of \( A_p \), \( A_s \), \( f_1 \), and \( f_2 \) from the analog filter frequency response shown on Figure 2. You will not be able to read the passband ripple level, but assume it is 0.5dB. Draw a diagram in your laboratory book of the frequency response specification.
LABORATORY WORK

Section I

Digital (Elliptic) Lowpass Filter

The elliptic filter order $N$ and natural frequency $\omega_n$ are calculated from the specification found in laboratory preparation question 4 by the MATLAB function `ellipord()`:

$$[N, \omega_n] = \text{ellipord}(f_1/(fs/2), f_2/(fs/2), A_p, A_s);$$

where $fs$ is the sampling frequency, which equals 22050Hz.

**Question 1:** What is the order and natural frequency of the digital elliptic filter? Check the order is the same as that specified for the TIMS Tuneable Lowpass Filter. Should $A_p$ and $A_s$ be entered as positive or negative numbers?

The next stage in the design process is to calculate the coefficients $B$ and $A$ for the digital filter. This calculation is performed by the MATLAB function `ellip()`.
[B,A] = ellip(N,Ap,As,wn);

where the arguments are defined as above.

**Question 2:** What are the coefficients B and A for the digital elliptic filter?

Once B and A are known, the design is complete. To confirm that the filter achieves the specification, obtain the frequency response using [h,w]=freqz() and plot the amplitude and phase responses using the statements:

```matlab
plot(11025*w/pi,abs(h)),grid   % Amplitude (linear scale)
plot(11025*w/pi,20*log10(abs(h))),grid % Amplitude (dB scale)
plot(11025*w/pi,unwrap(angle(h))),grid % Phase (unwrap scale)
```

Compare the digital filter responses with the analog filter responses shown in Figure 2 and with the specifications.

**Question 3:** Explain any differences in the amplitude and phase plots. What is the purpose of the MATLAB function `unwrap()`?

---

**Section I Checkpoint**  
**Marks: 2 out of 10**

**Section II**

(a) **Impulse Response of Elliptic Filter**

Set the TIMS *Tuneable Lowpass Filter* bandwidth to 4kHz and its (passband) gain to 2 (refer to Lab 1). Using the *VCO* module and the *Twin Pulse Generator* module, generate a rectangular pulse of width between 20µs and 50µs at a repetition frequency of below 250Hz. This can be achieved by connecting the output of *Variable DC* to the input of *VCO*. Refer to the VCO manual for more details. Note that: you may not be able to achieve this pulse width range and repetition frequency exactly, but aim for narrow pulses at the lowest repetition rate. You are trying to generate an impulse.

Connect this signal to the lowpass filter input via an *Adder* module. Adjust the adder gain until the filter's input pulse is 2V amplitude. Record the response seen on the CRO in your notebook. Obtain a record of the lowpass filter response, sampled at 22050Hz. Keep this record for later comparison.

The unit sample response of the Digital (Elliptic) Filter is computed as follows:

```matlab
delta = [1 zeros(1,99)]; %Unit sample signal (100 samples long)
n = 1:length(delta); % Sample counter
 t = (n-1)/22050 % Time scale (use 22050 sampling)
y = filter(B,A,delta); % unit sample response
 stem(t,y) % Plot of unit sample response
```

Display and obtain a record of the unit sample response of the elliptic lowpass filter.
**Question 4:** Compare the impulse response of the digital (elliptic) filter with the analogue impulse response of the TIMS tuneable filter. Explain any differences.

**(b) Linear Phase FIR Digital Lowpass Filter**

In the preceding section, a lowpass filter was designed with an IIR digital filter based on an elliptic prototype. This allowed direct comparisons to be made between the analogue and digital filters. In this section the same specification is achieved by an FIR digital filter designed using the Remez exchange algorithm. This algorithm is a very powerful tool and is in widespread use. Some properties of filters designed using this algorithm is:

1. Optimal equiripple FIR design.
2. Multiple passbands and stopbands can be specified.
3. The filters are of linear phase type.
4. Weights can be assigned to passband and stopband regions.

This algorithm is implemented in MATLAB with the command `remez`. It requires the following inputs:

1. A vector $F$ that defines the boundaries between passbands and stopbands. The frequencies must be in ascending order, starting with 0 and finishing with 1, where 1 corresponds to half the sampling rate.
2. A vector $M$ that defines the levels of the passband and stopband regions. The elements of $M$ must appear in equal valued pairs.
3. The filter order $N$.

For example, the coefficients $B$ of an order $N = 10$ FIR lowpass filter having a passband equal to 1 and extending to 4000 Hz and a stopband equal to 0 between 4400Hz and the Nyquist frequency 11025 Hz is designed using `remez()` as follows:

```
F = [0 4000/11025 4400/11025 1]; % Pass- and stop-band frequencies
M = [1 1 0 0]; % Pass- and stop-band levels
N = 10; % Filter order
B = remez(N,F,M); % Calculate the FIR coefficients
A = 1; % Specify 1 for A coefficients
[h,w] = freqz(B,A,200); % Calculate frequency response
plot(11025*w/pi,20*log10(abs(h))); % Display amplitude response
```

The specification used here is the same as that for the elliptic filter designed in the earlier section, but the order $N = 10$ is not large enough to achieve the specification (there is insufficient stopband attenuation). Calculate the coefficients $B$ using `remez()` for a range of filter orders. Hints: try $N>100$.

**Question 5:** What is the lowest order filter that achieves the required specification?

Obtain the unit sample response of the filter and record the result.

**Question 6:** Does this FIR filter require more or less computation and storage than the IIR filter designed previously?
Section III   Convolution and Filter Response

Consider the following filter:

\[ H(z) = \frac{z^8 - 0.9z^7 - 1.375z^6 - 0.975z^5 + 4.8125z^4 - 0.975z^3 - 1.375z^2 - 0.9z + 1}{z^8} \]

**Question 7:** Where are the zeros of this system on the z-plane?

Plot a pole-zero diagram for this filter. Confirm that the zeros are reciprocal of each other.

**Question 8:** From the last laboratory, what special property does this filter have with the zeros being reciprocal of each other?

**Question 9:** Where are the poles of this system on the z-plane?

**Question 10:** By taking the inverse Z-Transform, or by deriving from the definition of the Z-Transform, determine its impulse response, h[n].

**Question 11:** Is this an FIR or an IIR filter? Explain.

**Question 12:** A sequence, x[n], is being applied to this filter (with an impulse filter h[n]), its output sequence, y[n], can be determined as follow:

\[ y[n] = x[n] * h[n] = \sum_{k=0}^{\infty} x[n]h[k - n] \]

* denotes convolution

where \( x[n] = \{1, 3, 5, 7, 9, 11, 9, 7, 5, 3, 1\}, n = 0, 1, \ldots 10 \)

Determine y[n] by writing a simple MATLAB program to perform the required convolution.

**Section III Checkpoint**

**Marks:** 2 out of 10

**Bonus 1 mark:**

Write MATLAB code to perform convolution without using built-in function \texttt{conv()}. 

---

ELEC 3104 Session 1, 2013  Page 5 of 6
Section IV  Design a 4th order bandpass filter

The design of a bandpass filter can be carried out using MATLAB. For example, for the design of an elliptic filter using MATLAB, one might use the sequence of commands

\[
\begin{align*}
Fs &= \ \; \% \text{ sampling rate} \\
N &= \ \; \% \text{ order is } 2*N \\
Rp &= \ \; \% \text{ passband ripple in dB} \\
Rs &= \ \; \% \text{ stopband ripple in dB} \\
Wn &= \quad [\text{Flo}/(Fs/2) \text{ Fhi}/(Fs/2)]; \\
&\quad \% \text{ passband } [\text{Flo}, \text{ Fhi}] \text{ normalized for } \pi \\
[B,A] &= \text{ellip}(N, \text{ Rp}, \text{ Rs}, \text{ Wn})
\end{align*}
\]

The (partial) specification for the required IIR bandpass filter is:

- Sampling Frequency: 8 kHz
- Lower Stopband: 0 - 0.7 kHz
- Passband: 1.1 - 2 kHz
- Higher Stopband: 3.1 - 4 kHz
- Passband ripple/attenuation: 3 dB

Firstly, note that it may not be possible to obtain this specification with a 4th order filter. You need to choose both stopband ripples as small as possible, subject to the constraint of a 4th order filter. (Remember that this is a bandpass filter.) Choose which IIR filter option (Butterworth, Chebychev I, Chebychev II, or Elliptic) you think is the most suitable and explain your reasoning (with plots of magnitude responds of all the above four options).

---

**Module Outcomes:**

At the end of the laboratory, students should have a thorough understanding of the following concepts:

- Main differences between FIR and IIR Filters.
- Convolution.