A. CHEBYSHEV FILTER DESIGN

Before starting this assignment, please set up a suitable Word document to form your report answering the questions below and assessing the results. For each exercise please capture the plot (using eg the Snipping tool or any other convenient approach) and include in your report. The requested Matlab scripts should each be included in the report as an appendix. There are **5 marks** for the overall presentation and organization of your report.

This assignment is an opportunity to explore a number of new MATLAB functions, in particular those connected with filter design.

The *cheb1ord* function takes a set of filter design requirements and uses this to calculate the minimum order of the Chebyshev filter which would satisfy these requirements. The *cheby1* function uses this together with the design requirements to calculate the numerator and denominator coefficients of the filter.

The *freqs* function uses the numerator and denominator coefficients together with a specified frequency range to calculate the frequency response of the filter; the linear magnitude response can then be plotted using the *abs* function to calculate the magnitude.

The *bode* function is used to produce Bode plots from the transfer function, which itself can be calculated from the numerator and denominator coefficients using the *tf* function.

The *roots* function can be used to find the zeros (roots of the polynomial corresponding to the numerator coefficients) and/or the poles (roots of the polynomial corresponding to the denominator coefficients). These can then be plotted as complex numbers using the basic *plot* function.

The *step* function can be used to plot the step response of a filter or other LTIC system from the transfer function, obtained as described above.

All these functions assume that the numerator and/or denominator coefficients are expressed as a series of real numbers and arranged in descending order ie the coeeficient of the highest order element in the corresponding polynomial is given first.

Exercise A

Part 1

Locate slides 13-15 from Lecture 9.

Following the steps on slide 14, use MATLAB to design a Chebyshev filter according to the specifications on slide 13, with a linear magnitude plot.

The axes of this plot are to be labelled "Frequency (linear)" and "Magnitude (linear)". The plot should be titled "Linear Magnitude Plot". Save this program as a script called "Chebychev1.m". Verify that the plot is the same as the plot on slide 15.

(10 marks)

Part 2

An example of the use of the bode and tf functions to produce Bode plots from the numerator and denominator coefficients is as follows:

```
>> num = [1 2]; den = [1 20];
>> Ht =tf(num,den);
>> bode(Ht,'k-',{0.1 100});
```

where the figures in curly brackets represent the range of frequencies required.

Type this program into MATLAB and get it working. The plot should appear the same as the first plot in the answers to Tutorial 4 (answer to section 2).

Modify "Chebyshev 1" to allow two subplots, one above the other. The first subplot to show the linear magnitude plot already implemented. The second subplot to show a pair of Bode plots (log magnitude and phase against log of frequency), to be implemented using the bode function. The bode function will label the axes and produce a title automatically but you will need to specify an appropriate frequency range for the bode function to use.

Save this program as a script called "Chebychev2.m". Save the plot to include in your report.

(5 marks)

Part 3

The roots function takes one parameter which is a set of coefficients of eg a Laplace polynomial. The following MATLAB code fragment:

```
>> r = roots (set_of_coefficients);
>> plot(real(r),imag(r),'x');
```

would work out the roots corresponding to the coefficients given in the vector "set_of_coefficients" and plot them as a set of complex numbers represented by points marked by an "x".

Now use the roots function to derive the poles of the transfer function from the denominator coefficients already calculated and plot them on a third subplot below the Bode plots. Use the *axes* function to set the axes on this plot so that the divisions are of equal length (to prevent distortion of the shape of the plot of the filter poles). Label the axes of this subplot as "Real" and "Imaginary". The plot should be titled "Poles".

Save this program as a script called "Chebychev3.m". Save the plot to include in your report.

(5 marks)

Part 4

What is the order of the filter you have designed?

What would you estimate to be the approximate rolloff rate above 150 radians per second in dB per decade of frequency to the nearest 10dB (you may have to adjust the Bode plot upper frequency to get a better view of this).

Based on your Bode plot, what do you think the phase response at zero frequency will be? What would you estimate the phase response at infinite frequency to be?

Using Chebyshev3, now change the design parameters as follows:

(i) Ws = 60, all other parameters unchanged

(ii) Ws = 55, all other parameters unchanged

Record the order of the filter, your approximate estimate of the rolloff rate and of the phase response at infinite frequency in each case.

What do you think is the relationship between filter order and rolloff rate in dB per decade of frequency? Similarly what is the relationship between filter order and phase response at high frequencies?

(10 marks)

B. LTIC FILTER TIME AND FREQUENCY RESPONSE

The following MATLAB code takes a set of numerator coefficients and a set of denominator coefficients (those given are just examples) and uses the step function to calculate the step response of an LTIC filter or system.

```
>> num = [1]; den = [1 1 25];
>> Ht = tf(num, den);
>> step(Ht,'k');
```

Type this program into MATLAB and get it working. You should see an oscillatory response.

Exercise B

Part 1

Reusing as much as possible of the work you've already done, produce a MATLAB program to achieve the following (you should be able to do this from the work you did above to implement Chebyshev3 supplemented by using/adapting the code fragments in this document). Save your script as LTIC1.m and include it in an appendix to your report.

Specification:

1. Define vectors num and den to represent the numerator and denominator coefficients

2. Calculate the transfer function from these (you may wish to leave the semicolon out of the line which calculates this – this should have the effect of displaying the transfer function in the command window)
3. Produce three subplots as follows (presented as one below the other in this order):
    a. Step response
    b. Bode plot
    c. Poles and zeros

The poles should be represented in the same way as in the poles plot in the previous exercise, with the same labels for the axes. You will need to extend this subplot to include the zeros, which should be represented with the lower case letter "o" (not the zero character which would appear elliptical rather than round). The plot should be titled "Poles and Zeros".
The step response plot should be titled "Step Response" and the axes should be labelled "Time" and "Amplitude".

Set num to [1 2] and den to [1 20] and run the program. Capture the three subplots and include them in your report. [Note at this stage the Bode plots should be the same as you had in the bode function practice exercise above].

(15 marks)

Part 2

Now use your program to produce plots for the following transfer functions:

(i) $25/(s^2 + 5s + 25)$

(ii) $(s^2 + 5s + 25)/(s^2+2.5s+25)$

For each transfer function:

- capture the three subplots and include them in your report
- identify what type of filter this is (low pass, high pass, bandpass, bandstop)
- explain the key points of the Bode magnitude and phase plots in relation to the positions of the poles and zeros
- explain the step response in relation to the position of the poles

(10 marks)

C. LTIC NOTCH FILTER DESIGN

Exercise C

An item of audio equipment needs a notch filter to eliminate mains hum. It is to be designed to operate in the UK where the mains frequency is nominally 50Hz but may be assumed to vary between 48Hz and 52Hz.

The filter transfer function can be assumed to be:

$(s^2 + w_n^2)/(s^2+2dw_n+w_n^2)$

As well as producing Bode plots, the bode function can be used to calculate magnitude and phase at specific frequencies (in radians per second). The magnitude is provided in linear form and needs to be converted if the dB value is needed. The following code fragment takes a transfer function Ht in the same form as above and produces the log magnitude and the phase value for a specified radian frequency W:

```
>> [mag, phase]= bode(Ht, W);
>> logmag=20*log10(mag);
```

Set up a small program based on the work you've already done which does the following:

1. defines w_n and d
2. uses these values to calculate values for num and den based on the function given above
3. calculates the transfer function (suggest leave the semicolon out so the values of the coefficients in the transfer function are shown in the command window)
4. calculates the log magnitude and phase at a given radian frequency

Save this program in Notch1.m and include it in an appendix to your report. Now use it to answer the following questions.

What should the value of w_n be? Considering values of d in the range 0.1 to 1.0 (to one decimal place only), what minimum value of d would you suggest in order to ensure that the gains at 48Hz and 52Hz should each be at least 20dB less than at audio frequencies (for this design assume the gain at audio frequencies is effectively 0dB).

List the values of gain at 48 and 52Hz this setting for d would achieve.

Enter the values for “num” and “den” calculated by the above program into LTIC1. Use this to sanity check your answer, revisiting your program and/or calculations if necessary.

Capture the LTIC1 subplots and include them in your report.

(15 marks)