EECS 250

MATLAB Project 1

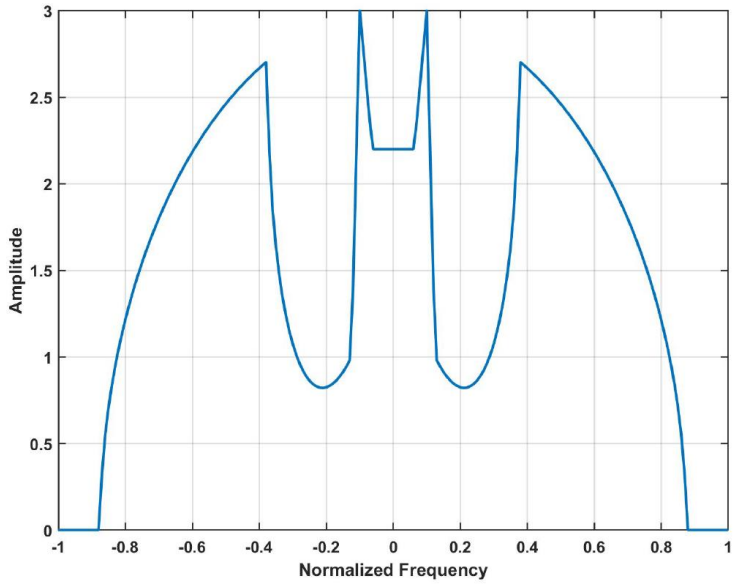
The report for this project needs to be typed and cannot be handwritten. You should add your MATLAB commands and the plots. The file should be in pdf format.

1. Use frequency sampling to design a filter b[n] that approximates the "batman" frequency response $B(e^{j\omega})$ in the figure below. What are the fewest number of samples needed to get a good representation of the filter? The value of $B(e^{j\omega})$ for any ω can be found using the function ***batman.m*** on the Canvas homepage. (Note that ***batman.m*** provides the actual gain, not the gain in dB). Don't forget to use fftshift.

   Use the dtft.m function to plot the spectrum of your filter for different number of samples. There is no specific criteria for choosing the smallest number of samples, just choose it visually, while comparing it to the batman shape. I am looking for an approximate value.

   Show the effect of the number of samples using three plots
   a) The spectrum of the filter when a number much higher than the minimum acceptable is used.
   b) The spectrum of the filter when the minimum acceptable number of samples are used.
   c) The spectrum of the filter when a number lower than the minimum acceptable is used.

   The value of $B(e^{j\omega})$ for any ω can be found using the function ***batman.m*** on the Canvas homepage. (Note that ***batman.m*** provides the actual gain, not the gain in dB).



In the following problems, if you are told to "upsample" a signal by a factor N, it means the combined operation of inserting N − 1 zeros between each existing sample and then interpolating using a low-pass filter. If you are told to "decimate" or "downsample" a signal by a factor of N, it means (unless you are told otherwise) the combined operation of passing the signal through a low-pass anti-aliasing filter, and then keeping only every N-th sample. In all cases below, when needed, use low-pass filters of length 101 generated using the *firpm* command.

2. Show how to generate samples of a 1000 Hz sinewave that is sampled at 20 kHz. Play the tone in MATLAB using the sound command to get an idea of what it sounds like.

3. Upsample the sinewave to 100 kHz. Listen to the signal after you have inserted the zeros and then after you perform the interpolation, and describe the results. Plot samples of the waveform before and after upsampling, ensuring that the time axis corresponds to real time (e.g., seconds) and not sample time. Are the waveforms the same? Explain.

4. Downsample the original sinewave to 4 kHz, with and without the anti-aliasing filter. Does the antialiasing filter make a significant difference in this case? Why or why not? Again plot the original and downsampled signal using real time on the horizontal axis, and explain any differences.

5. Predict what would happen if you kept only one of every 12 samples of the original sinewave, without using any low-pass filter. Be as precise as possible. Then listen to the result and see if you were correct.

6. Load the file tchaikovsky.mat from the MATLAB Files module on Canvas, which contains a portion of Tchaikovsky's famous "Dance of the Sugar Plum Fairy" sampled at 44.1 kHz. Downsample the file so the new sampling rates are $\frac{5}{6}, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{6}$ of the original rate respectively. Describe how the quality of the music clip changes. For the cases of $\frac{1}{2}, \frac{1}{3}\ and\ \frac{1}{6}$, downsample without using the low-pass filter. How important is the anti-aliasing filter? [You can write a function that can change the sampling rate by any factor and then test it by just changing the variables of the function]