ECE-210-B Homework #6

Filters

Cat Van West, Spring 2024

This homework is… filter. All filter. You'll feel like a barnacle afterwards. Especially since you'll need to derive sustenance from the water around you: this homework does not have notes, so you'll have to go to the Mathworks docs yourself! I'll provide some links to documentation pages, and will make myself available as a resource, but this is an exercise in learning on your own. Good luck, and let me know if you have questions!

The following table contains a few filter specifications:

Class	*Type*	R_{pass}	R_{stop}	f_{pass}	f_{stop}
Butterworth	bandpass	1 dB	50 dB	$f_s/6, f_s/3$	$f_s/7, f_s/2.5$
Elliptic	bandstop	1 dB	50 dB	$f_s/7, f_s/2.5$	$f_s/6, f_s/3$
Chebyshev Type I	lowpass	5 dB	40 dB	$f_s/9$	$f_s/8$
Chebyshev Type II	highpass	5 dB	40 dB	$f_s/3$	$f_s/4$

Your task is to design them. Assume f_s is 44.1 kHz. For each filter, do the following:

1. Either
   - **(a)** use `filterDesigner` to generate a function that creates the given filter, then call that function, or
   - **(b)** use `fdesign` and its associated functions to set filter specifications, create a filter object, and apply the filter.

   Use each of these strategies at least once. Since `filterDesigner` generates functions that use `fdesign` internally, it's a useful tool to learn the syntax.
2. Acquire a frequency response plot (both magnitude *and* phase) using `fvtool`.
3. Apply the filter to 2 seconds of Gaussian white noise (samples from a normal distribution, assumed to be sampled at f_s).

4. Plot the Fourier transform of the filtered signal (using `fft`). It should look a lot like the filter response, as white noise has a uniform frequency spectrum. Refer to the notes for proper scaling and use of `fft`.
5. Play back the unfiltered and filtered signals using `soundsc` and *give your impressions*: what, qualitatively, was the change the filter made?