MATLAB for SMS

Problem Set 4

Wavelets and fft

1. Generate a normally distributed signal with mean 100, variance 15, and 450 elements.
2. Pad the signal with its mean value (using `mean`) so that its number of elements is a power of 2 (try using `nextpow2`).
3. Compute the 1024-point Fourier transform of the signal.
4. Use `wfilters` and the string `'haar'` to get the wavelet and scaling filters. Alternatively, construct them manually. $\phi = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right] \quad \psi = \left[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$.
5. Pad the filter vectors with zeros so that they are the same length as the padded signal.
6. Compute the 1024-point Fourier transform of ϕ and ψ.
7. Elementwise multiply to obtain $\hat{a}_1(\omega) = \hat{\phi}(\omega)\,\hat{s}(\omega)$ and $\hat{d}_1(\omega) = \hat{\psi}(\omega)\,\hat{s}(\omega)$
8. Take the 1024-point inverse Fourier transform of $\hat{a}_1(\omega)$ and $\hat{d}_1(\omega)$. Truncate so they have 1024/2 = 512 elements. (Take the first 512 elements)
9. Dyadically downsample a_1 and d_1 so that they contain 512/2 = 256 elements.
10. Repeat steps 3-9 with $s = a_1$ to calculate a_2 and d_2. Note that a_2 will have 256/2 = 128 elements.
11. Repeat step 10 with a_2 to obtain a_3 and d_3 (64 elements).
12. Create time vectors for resolution levels 0:3. Notice you can simply downsample the time vector for the zeroth resolution level.
13. Make an 8.5x11 figure with 4 rows and 2 columns. The padded signal should be in row 1, and should span the two columns. The approximation at each resolution level should be in the left column, and the details should be in the right column. Print this figure as a pdf.
14. Calculate the threshold using $\tau = \sigma\sqrt{2\,log(n)}$, with `σ = mean(signal)`, and `n = numel(signal).`
15. Hard threshold the detail coefficients from each resolution level. Remember these are zero if less than or equal to the threshold, and that they remain unchanged if greater than the threshold. Use find or logical indexing, whichever of them suits your tastes.
16. Soft threshold the detail coefficients. Something like this:
    ```
    det1_soft = sign(det1).*max(abs(det1)-thresh,0);
    ```
17. Plot all three <u>sets</u> of detail coefficients – unaltered, hard thresholded, and soft thresholded – on a single page for comparison. Print this as a pdf. It may also be useful to plot the threshold values (positive and negative, as a dotted line maybe) so you can see them in comparison to the detail coefficients.
18. Using the hard thresholded detail coefficients, reconstruct the signal. Do this:
    a. Reverse the order of (2-element) phi and psi. Notice this has no effect on phi.
    b. Upsample a3 and d3_hard so that they have 128 elements. Pad phi and psi so that they also have 128 elements.
    c. Convolve (with fft and ifft…) the upsampled a3 with phi and d3_hard with psi. Add the resulting vectors to obtain r2.

d. Upsample r2 and d2_hard so that they have 256 elements. Convolve them with padded (to 256) phi and psi. Add to obtain r1.
e. Upsample r1 and d1_hard so that they have 512 elements. Convolve. Add to obtain r0. Truncate r0 so that it has 450 elements. This is your denoised signal.

19. Repeat 18 for the soft thresholded detail coefficients.
20. Construct a plot that overlays soft and hard thresholded reconstructions on the signal. Display each vector's mean and standard deviation on the plot.
21. Send me what you have (hopefully an m-file and/or the plots) by Monday August 1, 10 am.

Optional because it requires use of a wavelet toolbox function:

Plot the Daubechies family (8 wavelet and 8 scaling functions) of wavelets in the Fourier domain. Plot the amplitude of its Fourier transform as in the demo m-file. Note that `'db1' = 'haar'`, and that `'dbn'` represents the Daubechies wavelet having n vanishing moments over its support (interval on which it's nonzero). (The number of vanishing moments is the number of zero-crossings in the wavelet's support.) Put all the scaling functions on one plot and the wavelets on the other for 2 total subplots. Create some arbitrary sampling frequency and label x in terms of f_s. Constructing this plot should provide some insight as to the intention of the Daubechies family of wavelets. Plot them in the wavelet domain (no fft, just the filter) if you like. It won't be as informative, but will help familiarize.