Exercise 3 - Sound Compression with wavelets

In this exercise we will see how wavelets can be used to compress audio signals to a fraction of their original size. First a few words on the digital representation of audio signals. An analog sound is first *sampled*, it's intensity is measured at fixed intervals. The *sample rate*, or *sample frequency* is the number of samples taken per second, measured in Hertz (Hz). For the representation of a sample one uses a fixed number of bits, typically eight or sixteen, the intensity is scaled to the interval $[-1, 1]$.

You can start the Matlab code using `exercise3`. A figure window will appear with the following elements

- **Sample axes**, this shows the sample.
- **Wavelet axes**, this shows the wavelet coefficients in logarithmic scale. The coefficients of the different detail levels are separated by blue lines.
- **Load Sample button**, opens a Sun `.au` audio sample.
- **Play Sample button**, plays the sample.
- **Play Original button**, plays the original, unthresholded sample.
- **Wavelet drop down menu**, choose a wavelet for the analysis here. The coefficients will be drawn after selection. First set the number of decomposition levels.
- **Threshold button**, removes wavelet coefficients below the threshold. Set the threshold first, in the edit box (note that the exponent notation does not work, use 0.05 instead of $5e-2$). Clicking once marks the coefficients to be removed by red crosses, clicking a second time removes the unmarked coefficients from the plot.
- **Noise button**, adds some random noise to the sample. Set the noise level first in the edit box.
- **Remove button**, removes the wavelet coefficients of one level of the decomposition. Set the level to remove first in the edit box.

Try how high you can set the threshold without audibly distorting the sound. Use 'play sample' and 'play original' to compare. Try some different files. What kind of error is there in the result?

In view of the previous exercises it would be interesting to see if wavelets are suited for noise filtering. Use the edit box and button in the bottom row to add some noise to the signal and see if you can remove it using some different wavelets.

There is another way to filter the coefficients, setting the coefficients of a certain level to zero. Try removing coefficients from samples in this way. Do you notice that different levels roughly correspond to frequency bands? For example, if you do a seven level decomposition and remove the last two for 'door.au' you will notice that you have eliminated the whistling sound. The file 'sine.au' is the sum of two pure sine functions. Is it possible to remove the high frequency sine?